File Number S360-37
GY24-5027-0

Program Logic

IBM System/360 Disk Operating System Autotest

Program Number 360N-PT-459

This publication describes the internal logic of the IBM System/360 Disk Operating System Autotest program. It is intended for use by persons involved in program maintenance and system programmers who are altering the program design. Program logic information is not necessary for the operation of the program. Therefore, distribution of this publication is limited to those with maintenance and alteration requirements.

First Edition, March 1967

Specifications contained herein are subject to change from time to time. Any such change will be reported in subsequent revisions or Technical Newsletters.

Requests 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 IBM Corporation, Programming Publications, Endicott, New York 13760.

This publication is a detailed and comprehensive guide to the internal structure and functions of the IBM System/360 Disk Operating System Autotest program. It is a supplement to the program listing.

Effective use of this publication is based on an understanding of the IBM System/360 Operation, Disk Operating System (DOS) Assembler, DOS Supervisor and Control Program and DOS Autotest. This information can be found in the following publications:

- IBM System/360 Principles of Operation, Form A22-6821.
- IBM System/360 Disk and Tape Operating Systems, Assembler Specifications, Form C24-3414.
- IBM System/360 Disk Operating System, System Control and System Service Programs, Form C24-5036.
- IBM System/360 Disk Operating System, Supervisor and Input/Output Macros, Form C24-5037.
- IBM System/360 Disk Operating System, Operating Guide, Form C24-5022.
- IBM System/360 Disk and Tape Operating Systems, Utility Programs Specifications, Form C24-3465.
- IBM System/360 Disk Operating System, System Generation and Maintenance, Form C24-5033.
- IBM System/360 Disk Operating System, Autotest Specifications, Form C24-5062.
- IBM System/360 Disk Operating System, System Control Program Logic Manual, Form Y24-5017.

For information regarding other related publications, see IBM System/360 Bibliography, Form A22-6822.

This Program Logic Manual (PLM) consists of the following eight sections:

1. Introduction
2. Autotest Linkage Editor
3. Control Card Analysis
4. User Program Execution
5. Post-Processing
6. Disaster Continue
7. Card-to-Tape (Variable) Utility
8. Label List
9. Detail Level Flowcharts
10. Appendixes
    a. Diagnostic Messages
    b. Phase Name Identification
    c. Flowchart Symbols
    d. Flowchart Abbreviations

The Autotest program is divided into the following four major components:

- Autotest Linkage Editing
- Control Card Analysis
- User Program Execution
- Post-Processing

A general description, a high-level flowchart, storage allocation maps, record formats, and I/O flow are provided for each major component. The disaster-continue routine and the Autotest card-to-tape (variable) utility program are also described in detail. The high-level flowcharts have numeric designations, such as 00, for the program level flowchart; while the detail level flowcharts for each phase of Autotest have alphabetic designations, AA through ZZ.

This manual provides quick access to the detailed information on the internal logic of the DOS Autotest program. Cross-referencing is:

1. Program level chart refers to major component level flowcharts.
2. Major component level flowcharts refer to detail (phase) level flowcharts.
3. Label list--provides a cross-reference between the listing and the detail level (phase) flowcharts.
4. Diagnostic messages--cross-referenced to the program phase and detail level flowchart.

CONTENTS

CHARTS

The IBM System/360 Disk Operating System (DOS) Autotest is a testing aid designed to assist System/360 users in debugging object programs assembled by the DOS assembler, processed by the linkage editor, and run under the control of the DOS supervisor. Autotest can be used to test program modules that are in punched-card format or are in the relocatable program library on the system disk. Autotest can run in the background area under control of a multiprogramming supervisor, but concurrent allocations of storage to either foreground area are not permitted.

Autotest provides dynamic testing services by monitoring the execution of the user's object program. If a program being tested destroys some vital portion of the supervisor or Autotest control program, Autotest provides a special operating procedure (disaster continue) that finishes that test-job and continues the run. When an abnormal end-of-job condition arises, Autotest automatically gives a storage dump according to the specifications previously given by the user, and continues to the next job.

Autotest permits the user (by means of autopatch) to make changes to his program without a reassembly. Debugging of the program is aided by test requests, which allow repeated displays of main storage during user program execution.

The user identifies his program and communicates with Autotest by means of control cards. The control cards provide two advantages:

1. Remote testing is facilitated in that a minimum of operator intervention is necessary.
2. The program can be returned to the pre-Autotest state without a reassembly merely by removing the control cards.

Also, any utility programs (IBM or user-supplied) can be used to build input files and/or process output files.

Autotest provides the following program features that may be selected by the user through control cards:

Autopatch

Facilitates changes and corrections to the assembled object program without reassembly. Provisions are made to add and replace instructions or constants. Autotest computes all necessary linkages and inserts patches into the Autotest table area in main storage. The three functions of autopatch are: exchange (EXC), replace constants (CON), and add (ADD).

Display

Data from selected areas of main storage can be printed at specified times during execution of the object program. Because Autotest provides a variety of formats for the printout, the user can select the one most meaningful for that area of storage. For additional information see the Autotest Specifications publication listed in the Preface.

Panel

General registers, main-storage positions 24-127, and floating-point registers, in any combination, can be printed at specified times during execution of the object program.

Occurrence of Panel and Display

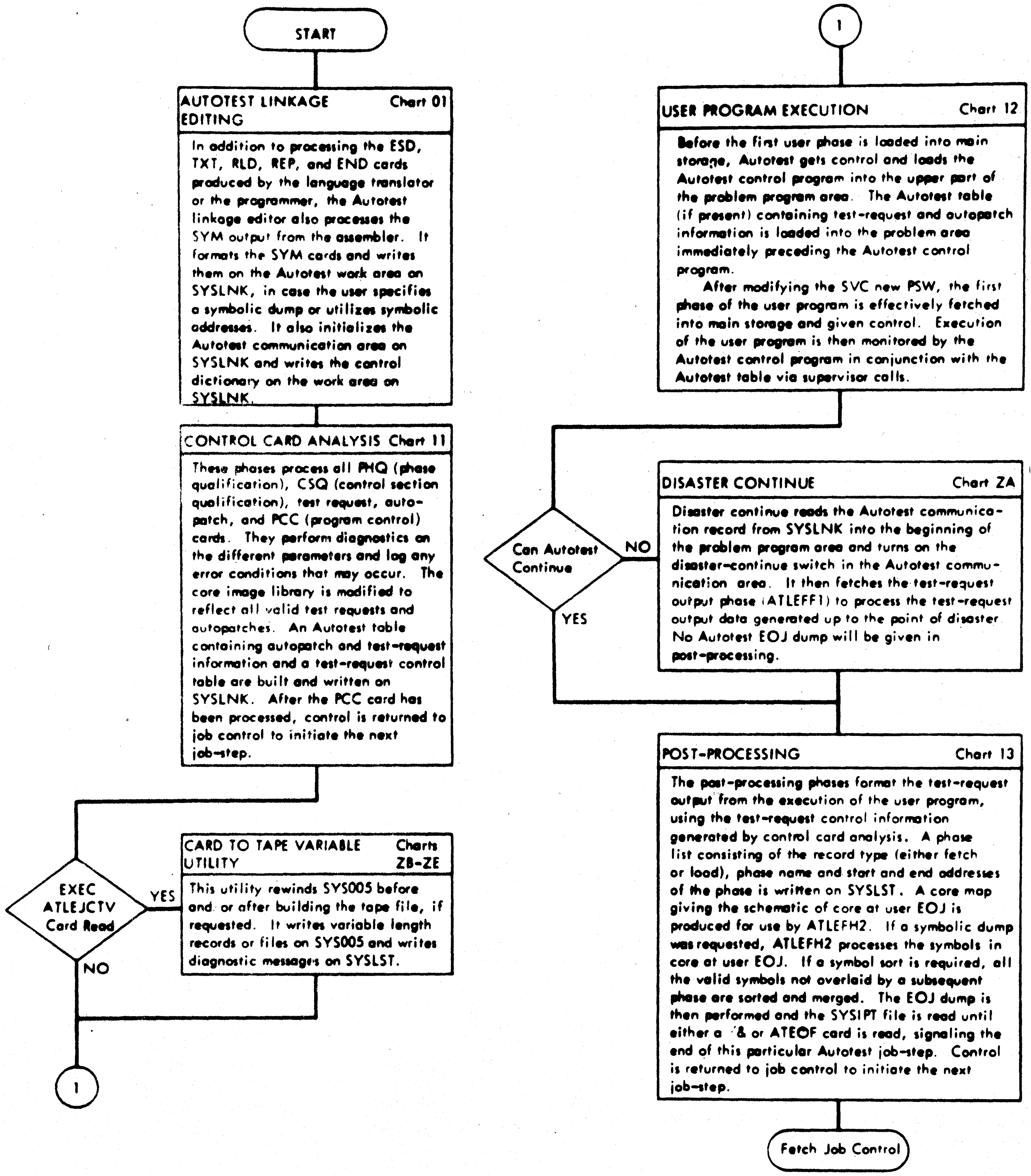
The user can control the conditions under which a panel and/or display occur in a loop. The conditional function is called ON. An ON request is a conditional regulation of Autotest actions according to the program flow counted and tested in the Autotest control program. A request is executed if the conditions of the ON field are met when the test-point address specified on the control card is reached during execution of the object program.

The ON condition is specified by three numbers (BEGIN, END, and EVERY), which serve as comparison values for a counter that Autotest sets up. Each time the test-point address is reached, the counter is decreased by one and then compared with the ON values. For example, ON 3, 7, 2 results in action taking place on the third time (BEGIN), every second time (EVERY) ending not greater than seven (END). Hence, action takes place on the 3rd, 5th, and 7th times the test point is reached.

Main-Storage Printout

The contents of main storage are dumped on a normal end-of-job, unless the dump is requested to be suppressed. On an abnormal

Chart 00. Disk Operating System Autotest Program



end-of-job, this dump is always provided and cannot be suppressed. The user can specify one of the following formats:

1. Hexadecimal only or

2. Any combination of hexadecimal and:

    a. character (alphameric)

    b. mnemonic

    c. symbolic

If a symbolic dump is requested, an assembler symbol table must be supplied to Autotest.

Phase List

Provides a list of the phases with their start and end addresses in the order they are fetched or loaded into main storage.

Card-to-Tape (Variable) Utility

Autotest provides a special utility program for building tape records of any format or length, depending on the size of the machine.

Symbolic Capabilities

These symbolic capabilities are available only if an assembler symbol table is supplied to Autotest.

Autopatch: Allows symbolic reference of an instruction address or constant at which the patch is to take place. The appropriate instruction or constant does not necessarily require a label (symbol). Address increment (also called byte displacement) from a preceding or succeeding labeled instruction can be used to define the appropriate patch-point address.

Test Request: Allows symbolic reference of either the display limits and/or the instruction address at which the test-request action is to occur. Address increment (also called byte displacement) from a preceding or succeeding labeled instruction can be used in the same manner as described for autopatch.

EOJ Dump: Allows the user to include symbols for easy referencing. Only those symbols not overlaid by a subsequent phase are included in the EOJ dump.

Disaster Continue

If the user's program destroys the supervisor or the Autotest control program, or goes into an unending loop and the machine operator's usual job-termination procedure cannot conclude the job, the disaster-continue procedure must be used. After the operator dumps main storage using a utility storage dump program and processing is reinitiated using the system IPL (initial program load) procedure to restore the supervisor, the disaster-continue routine is called from the core image library by the //bEXECbATLECONT control card. This routine in turn calls the post-processing phases to process the phase list and all test-request output data generated up to the point of disaster. No Autotest end-of-job dump is given. After processing of the terminated job is complete, control is returned to job control when the ATEOF card from SYSIPT is read. At this time, processing of the next job step is initiated.

Utilities

Any utility program (IBM or user-supplied) resident in the core image library (CIL) can be executed before or after each user test.

SYSTEM ENVIRONMENT

The minimum machine configuration required for DOS Autotest is:

- 16K bytes of main storage
- Card reader--1442, 2501, 2520, or 2540. A tape or disk unit may be substituted for this device if an IBM 1052 Printer-Keyboard is provided for operator messages.
- Standard Instruction Set
- Printer--1403, 1443, or 1404 (continuous forms only). A tape or disk unit may be substituted for this device if a 1052 printer-keyboard is provided for operator messages.
- 2311 disk storage drive
- 1052 printer-keyboard (limited system operation possible if not available)

Also, DOS Autotest supports these machine features:

- Additional main storage up to 1024K bytes
- Storage-Protection feature
- Timer feature
- Simultaneous Read-While-Write Tape Control (2404 or 2804)

- Any channel configuration up to one multiplexor channel and six selector channels
- Tape Switching Unit (2816).

PROGRAM COMMON DATA

Autotest Work Area on SYSLNK (IJSYSAT)

Used by Autotest for communication between the phases. It contains the following information:

1. Autotest communication area.
2. Small 155-byte EOJ dump record. See Figure 1 for format.
3. SYM records, if the assembler symbol table was provided for the Autotest linkage editor to process.
4. Control dictionary.
5. Test request control information.
6. Test request output, including user phases fetched or loaded into the problem program area.
7. Autotest table, which is built backwards starting at the end of the work area.
8. EOJ dump information (overlays Autotest table).
9. Core map (overlays test request output area).
10. Symbol sort and merge output (if required).

See Figures 1 and 2 for the format of the Autotest work area on SYSLNK.

For a machine size of 16K, the minimum size of the Autotest work area is six tracks. For a machine size of 32K or greater, the minimum size of the Autotest work area is eleven tracks. The minimum size work area can only be used if the symbolic and test request capabilities are not utilized. If there is an insufficient amount of work area assigned, diagnostic messages are written on SYSLST and SYSLOG.

When calculating the required DOS Autotest work area size, space for any or all of the following components must be considered:

SYMBOLIC ADDRESSES: 169 symbols can reside on one track, but 156 symbols is the maximum for the last track of symbols. (The last track of symbols must contain an EOF marker to delimit the symbol file.) For example: 156 symbols require one track, but 157 symbols require two tracks.

Area*	Record(s)	Description	R = Required O = Optional
1	1	116-byte Autotest communication area	R
1	2	155-byte small EOJ dump record**	R
1	3-X	Maintenance area	
2	1-X	SYM1 area containing the SYM records processed by the Autotest linkage editor. The SYM1 area is used only if an assembler symbol table accompanied the user object program.	O
3	1-X	Control dictionary	R
4	1-X	If valid test-request control cards have been processed by control card analysis, this area contains the test-request control information necessary to format any test-request output.	O
5	1-X	Contains test-request output from execution of the user program as well as any phase fetch or load records. The phase fetch or load records are interspersed with the test-request output records.	R
6		Unassigned	
7	1-X	Autotest table built backwards starting at the end of the work area.	O
8		EOJ2 area containing the user EOJ dump information. The EOJ2 area overlays the Autotest table area. Three tracks are reserved if the machine size is 16K. Eight tracks are reserved for EOJ dump information for a machine size of 32K or greater.	R

* Each area (1-8) is defined on a track basis. Each area starts on the next sequential track except the Autotest table area which is built backwards starting at the end of the work area and the EOJ2 area (8) which overlays the Autotest table. An overflow condition occurs whenever areas 2-6 overlay area 8, the EOJ dump area.

** The format of the 155-byte record is:

Bytes	Contents
0-63	General registers (0-15)
64-95	Floating-point registers
96-103	User PSW
104	Cancel code
105-106	Length of user label area
107-154	User communication area

Figure 1. Format of the Autotest Work Area (IJSYSAT) Through Execution of the Test Request Output Phase, ATLEFF1

Area*	Record (s)	Description	R = Required O = Optional
1	1	116-byte Autotest communication area	R
1	2	155-byte small EOJ dump record (See Figure 1 for format.)	R
1	3-X	Maintenance area	
2	1-X	SYM1 area containing the SYM records processed by the Autotest linkage editor. This area is used only if an assembler symbol table accompanied the user object program.	O
3	1-X	Control dictionary**	R
4	1-X	If valid test-request control cards have been processed by control card analysis, this area contains the test-request control information used to format the test-request output.**	O
5	1-X	Core map records	O
6	1-X	SYM2 area containing the symbol output from the symbol organization phase, ATLEFH2. If a symbol sort is required, this area is used to sort the symbols. The sorted symbol strings are written on the SYM1 area to be merged.	O
7		The EOJ2 area containing the EOJ dump information to be processed by the EOJ dump phase, ATLEFH3.	R

*Each area (1-7) is defined on a track basis. Each area starts on the next sequential track except the EOJ2 area (7) which is built starting at the end of the work area. An overflow condition occurs whenever areas 5 and 6 overlay area 7, the EOJ2 area, and whenever the SYM1 area overlays the core map area.
**The SYM1 area can be expanded to include areas 3 and 4 after the phase list and core map phase, ATLEFG1, has been executed. If a symbolic dump is requested and the symbol output from ATLEFH2 needs sorting, the SYM1 is used for merging the symbol output.

Figure 2. Format of the Autotest Work Area (IJSYSAT) After Execution of ATLEFF1

CONTROL DICTIONARY: A maximum of 156 combined phases and control sections can reside on one track. Example: 78 phases and 78 control sections require one track, but 78 phases and 79 control sections require two tracks.

TEST REQUESTS: 234 records maximum per track, but only 216 records permitted on the last track of test-request control information.

END-OF-JOB DUMP: Three tracks are required if less than 32K of main storage, but eight tracks are required if 32K or greater.

AUTOTEST CONTROL PROGRAM: At object time, the Autotest control program writes on disk, recording phases fetched and loaded, and temporarily stores test request output on disk. The number of tracks required for this can be calculated by the following formula.

$$O = \frac{F+L+2(NP_1+NP_2+..NP_n)+(ND_1+ND_2+..ND_n)}{20\ (19\ \text{for last track})}$$

Note: If there is a remainder, add one track.

Code:

F = Number of phases to be fetched. If a phase is fetched more than once, it is counted each time.

L = Number of phases to be loaded or reloaded.

N = Number of occurrences expected for a panel (P) or a display (D). The value of N may be governed by the "ON" parameter.

P = A unique panel request. (P = 1 in the formula)

D = A unique display request.

$$D = \frac{\text{Display area end address} - \text{Display area start address}}{113}$$

To calculate the total number of tracks required for the DOS Autotest work area, the totals of the five components are used in the following formula:

$$T = 2S+CD+C+E+O+1$$

where:

T = total number of tracks

S = number of tracks required for symbols

CD = number of tracks required for control dictionary

C = number of tracks required for test-request control

E = number of tracks required for end-of-job dump

O = number of tracks required by the Autotest control program.

Assumptions: For all practical purposes, the preceding formula will determine the actual work area required for a job or will act as a guide for setting up a realistic maximum work area at a particular installation. The work area formula reflects the most representative total picture of the work area. At post-process time of Autotest, some of the described areas are overlaid by different information, but rarely will the work area size have to be increased by these post-process considerations.

Autotest Table

The Autotest table is built backwards starting at the end of the Autotest work area on SYSLNK by the control-card-analysis phase. If any Autotest service (test request or autopatch) requires an entry in the Autotest table, the table is built on SYSLNK and moved to the problem program area for use by the Autotest control program during execution of the user object program. After the user program is executed, the Autotest table area on SYSLNK is overlaid by the user EOJ dump information.

Figure 3 summarizes the length of the entry for each Autotest service that creates an entry in this table.

For all services that generate Autotest table entries (add, greater-than-original-length exchange, and test requests), the entry is:

1. Identification (SVC number).

2. Stored object program instruction(s).

3. A supervisor call instruction for linkage back to the user program.

A patch-point address (as used in add-patching or exchange with the new instruction greater in length than the old), and a test-point address (as used for test requests), is explained as follows:

> The test request or autopatch requiring a table entry involves a supervisor call which overlays the instruction at the test-point or patch-point address. The instruction that has been displaced is relocated to the Autotest table area. The test request or autopatch is executed before the instruction that was displaced at the test-point or patch-point address.

For the exchange (EXC) and replace constant (CON) functions, the patch-point address is the address of the leftmost byte (op code) of the old instruction, and the leftmost (high-order) byte of the old constant, respectively.

Autotest Service	Number of Bytes Required for Table Entry
TEST REQUEST:	
For each test point, four bytes are required, plus, for each:	
• Display (DSP)	8+D
• Panel (PNL)	2+D
• Display and Panel (DPL)	8+D
• Display with ON	14+D
• Panel with ON	8+D
• Display and Panel with ON	14+D
AUTOPATCH:	
For these two types of autopatch, four (4) bytes are required, plus, for each:	
• Add patch (ADD)	D+N+2
• Exchange patch (EXC) with more bytes in new instruction than old	2+N+(P-2)

Key: D = number of bytes in the displaced instruction
N = number of bytes in the new instruction(s)
P = number of bytes in the old instruction before exchange

Example: For a display together with a display and panel with ON at an address of a 4-byte instruction, the number of bytes required for the table entry is equal to 4 (for test point) + [8 (display) + 4 (displaced instruction)] + [14 (display and panel with ON) + 4 (displaced instruction)] = 34 bytes

Figure 3. Size of Autotest Table Entries

Label	Length in Bytes	Displacement	Function	Created by	Used by
PRBPGADD	4	0	Contains the following information: 1. Area available for Autotest table 2. Size of the Autotest table 3. Address plus one of last byte of main storage dumped on SYSLNK. This address is 16K if the machine size is 16K or 32K if the machine size is 32K or greater	1. Autotest linkage editor 2. Control card analysis 3. Dump user program (SSBATST3)	1. Control card analysis 2. Autotest control program initialization (ATLEFE1) 3. EOJ dump (ATLEFH3)
TBLPTR	4	4	Pointer to the length of the first Autotest table entry (CTLADD-4)	Autotest control program initialization (ATLEFE1)	Autotest control program (ATLEFE2)
PCHAREA	4	8	1. Pointer to beginning of Autotest table 2. Pointer to beginning address of logical transient area	1. Autotest control program initialization (ATLEFE1) 2. Dump user program (SSBATST3)	1. Autotest control program (ATLEFE2) 2. EOJ dump (ATLEFH3)
CTLADD	4	12	Pointer to beginning of Autotest control program	Autotest control program initialization (ATLEFE1)	Set up interface between Autotest control program and supervisor (SSBATST1), Autotest control program (ATLEFE2)
	3	16	Unassigned		
FLTSWT	1	19	1. X'08' - Floating-point hardware present 2. X'03' - No floating-point hardware	Autotest linkage editor	Control card analysis, Dump user program (SSBATST3)
ACTLIM	2	20	Limiting number of actions plus one	PCC statement processor (ATLEFC5)	Autotest control program (ATLEFE2)
DUMPSW	1	22	1. X'00' - No normal EOJ dump 2. X'FF' - Normal EOJ dump required	PCC statement processor (ATLEFC5)	Dump user program (SSBATST3)
TYPDMP	1	23	Reason user program canceled by Autotest 1. FA - Disk work area overflow 2. FC - Action count exceeded 3. FD - Table entry not found The cancel code (FA, FC, or FD) is passed to SSBATST3 which, in turn, passes the information to the EOJ dump phase (ATLEFH3) as the condition code.	Autotest control program (ATLEFE2)	Dump user program (SSBATST3)
OPTSWT	1	24	1. X'02' - SYM input present 2. X'04' - Disaster continue 3. X'08' - Dump supervisor on normal EOJ 4. X'20' - Autotest table present	1. Autotest linkage editor 2. Disaster continue (ATLECONT) 3. PCC statement processor (ATLEFC5) 4. Control card analysis	1. Control card analysis, PCC statement processor (ATLEFC5), Phase list and core map (ATLEFG1) 2. Phase list and core map (ATLEFG1), EOJ dump (ATLEFH3) 3. EOJ dump (ATLEFH3) 4. Autotest control program initialization (ATLEFE1)
SVCNO	1	25	1. Lowest SVC number used by Autotest 2. Number of sorted symbol strings	1. Control card analysis 2. Symbol sort (ATLEFD1)	1. Autotest control program (ATLEFE2) 2. Symbol merge (ATLEFD2)

Figure 4. Autotest Communication Area (Part 1 of 3)

Label	Length in Bytes	Displacement	Function	Created by	Used by
EOJ1P	4	26	Address of small 155-byte EOJ dump record on SYSLNK	Autotest linkage editor	Dump user program ($$BATST3), EOJ dump (ATLEFH3)
USERPH	8	30	Name of first user phase	Autotest linkage editor	Set up interface between Autotest control program and supervisor ($$BATST1)
T ROUTS	4	38	1. Beginning address of test-request output file on SYSLNK 2. Beginning address of core-map file on SYSLNK	Autotest linkage editor	1. Test-request output phase (ATLEFF1) 2. Symbol organization (ATLEFH2)
T ROUTE	4	42	1. End address of test-request output file on SYSLNK 2. End address of core-map file on SYSLNK	1. Autotest control program (ATLEFE2) 2. Phase list and core map (ATLEFG1)	1. Test-request output phase (ATLEFF1) 2. Symbol organization (ATLEFH2)
PATCHTS	4	46	Start address of Autotest table file on SYSLNK The Autotest table is built backwards on SYSLNK.	Control card analysis	Autotest control program initialization (ATLEFE1)
PATCHTE	4	50	End address of Autotest table file on SYSLNK.	Control card analysis	Autotest control program initialization (ATLEFE1)
EOJ2S	4	54	Start address of EOJ dump file on SYSLNK	Autotest linkage editor	Dump user program ($$BATST3), EOJ dump (ATLEFH3)
	7	58	Unassigned		
	1	65	Upper track limit for split cylinder	Autotest linkage editor	All Autotest phases
	1	66	Lower track limit for split cylinder	Autotest linkage editor	All Autotest phases
	3	67	Unassigned		
EXTRAD	1	70	1. X'00' - No test-request output 2. X'01' - Test-request output	Autotest control program (ATLEFE2)	Test-request output (ATLEFF1)
	6	71	Unassigned		
DMPFORMT	1	77	Format of EOJ dump A. Normal 1. X'01' - Hexadecimal 2. X'02' - Character (Alphameric) 3. X'04' - Mnemonic 4. X'08' - Symbolic B. Abnormal 1. X'10' - Hexadecimal 2. X'20' - Character (Alphameric) 3. X'40' - Mnemonic 4. X'80' - Symbolic	PCC statement processor (ATLEFC5)	EOJ dump (ATLEFH3)
	6	78	Unassigned		
SYMIS	4	84	1. Start address of SYM input on SYSLNK 2. Start address of merge input on SYSLNK	Autotest linkage editor	1. Symbol organization (ATLEFH2) 2. Symbol merge (ATLEFD2)

Figure 4. Autotest Communication Area (Part 2 of 3)

Label	Length in Bytes	Displacement	Function	Created by	Used by
SYM1E	4	88	1. End address of SYM input on SYSLNK 2. End address of merge input on SYSLNK	Autotest linkage editor	1. Symbol organization (ATLEFH2) 2. Symbol merge (ATLEFD2)
C/DS	4	92	Start address of control dictionary file on SYSLNK	Autotest linkage editor	Control card analysis, Phase list and core map (ATLEFG1)
C/DE	4	96	End address of control dictionary file on SYSLNK	Autotest linkage editor	Control card analysis, Phase list and core map (ATLEFG1)
T RCTLS	4	100	Start address of test-request control information on SYSLNK	Control card analysis	Test-request output (ATLEFF1)
T RCTLE	4	104	End address of test-request control information on SYSLNK	Control card analysis	Test-request output (ATLEFF1)
SYM2S	4	108	1. Start address of symbol input on SYSLNK 2. Start address of merge output on SYSLNK	Phase list and core map (ATLEFG1)	1. Symbol sort (ATLEFD1) 2. Symbol merge (ATLEFD2)
SYM2E	4	112	1. End address of symbol input on SYSLNK 2. End address of merge output on SYSLNK	Phase list and core map (ATLEFG1)	1. Symbol sort (ATLEFD1) 2. Symbol merge (ATLEFD2)

Total length of Autotest communication area = 116 bytes

Figure 4. Autotest Communication Area (Part 3 of 3)

Patches and test requests are accomplished as follows:

- Add Patch--The instruction beginning at the patch-point address is moved to the Autotest table and is called the displaced instruction. In the Autotest table, the new instruction(s) is placed immediately before the displaced instruction.

- Exchange Patch--For an exchange patch where the new instruction has more bytes than the old, the new instruction is moved to the Autotest table.

- Test Request--The instruction at the test-point address is moved to the Autotest table and is called the displaced instruction.

- Linkage--Two SVC's (supervisor call) instructions are then added to the program, one at the patch-point or test-point address, and the other after the displaced or new instruction(s) in the Autotest table. These SVC's serve as the linkage between Autotest and the user's program. Whenever an SVC occurs in the user program, control passes to Autotest. Autotest analyzes the SVC interruption code to determine its source. If it was an Autotest SVC (codes 55-254), the Autotest table is used to determine and to execute the service requested. Return to the user's program is caused by the second SVC (255).

Autotest Communication Area

Used by Autotest for communication between phases. It is initialized and written on the Autotest work area as the first record on track 1 by the Autotest linkage editor. The Autotest linkage editor then moves the 116-byte Autotest communication area to the beginning of the problem program area for use by the control-card-analysis phases. The PCC statement processor (ATLEFC5) writes the updated Autotest communication area on SYSLNK before fetching job control. ATLEFE1 moves the communication area to the end of the problem program area for use by the Autotest control program. $$BATST3 then moves it to the beginning of the problem program area for use by the post-processing phases. Figure 4 shows the format of the Autotest communication area.

AUTOTEST LINKAGE EDITING CHART 01

After job control reads the EXEC ATLEDT (execute Autotest linkage editor) control card to initiate the Autotest portion of the job, the normal linkage editor functions are performed with the following additions:

1. Saves the SYM cards (symbol table output from the assembler) and writes them on SYSLNK.
2. Writes the control dictionary containing only PH (phase), SD (section definition) and PC (private code) entries on SYSLNK.
3. Ensures that the first name in the system phase directory is ATLEFE1, the Autotest control program initialization phase.

The Autotest linkage editor prepares programs for execution on DOS, and accepts as input the relocatable object modules produced by the language translators. It processes these modules into program phases, which may be immediately executed via the core-image library. However, the Autotest linkage editor never performs a catalog function.

The linkage editor control cards direct the program to read input module(s) and to form phases from the control sections within the modules. Figure 5 shows how phases can be formed. The linkage editor relocates the origin of each control section in the phase, assigns each phase an area of main storage and a transfer address, and modifies the contents of the address constants in the phase.

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

The module TXT items are then built into phase blocks. The RLD items (address constants) are modified and inserted into the text. A transfer address is determined for each phase.

LANGUAGE TRANSLATOR MODULES

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

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

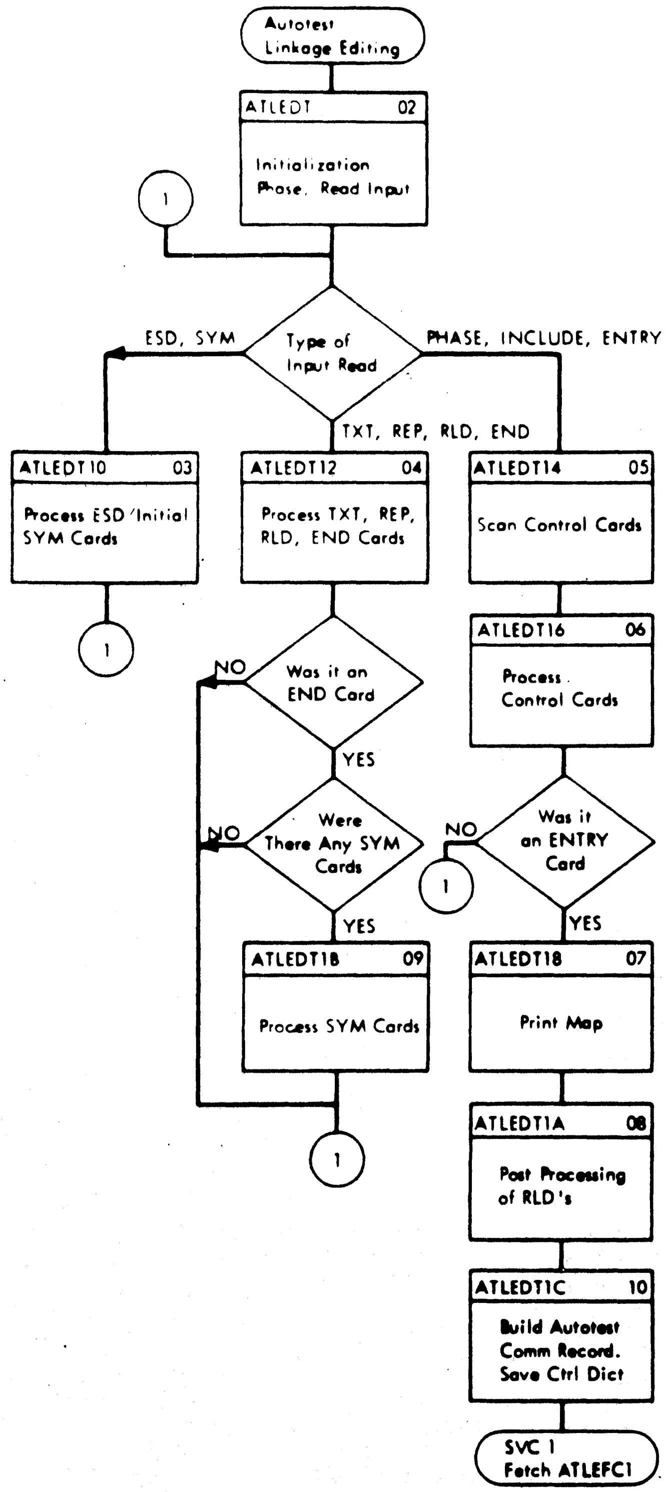
Six card types are produced by the language translators or the programmer to form a module. They appear in the following order:

Card Type	Definition
ESD	External symbol dictionary
SYM	Symbols (labels) referenced by program being linkage-edited
TXT	Text
RLD	Relocation list dictionary
REP	Replacement to text made by the programmer
END	End of module

Sample of a 2-module input resulting in a 3-phase output	
Language Translator Output	**Linkage Editor Output**
Module A ESDs TXT - CSECTA TXT - CSECTB TXT - CSECTC RLDs	Phase 1 CSECTA CSECTB
Module B ESDs TXT - CSECTD TXT - CSECTE TXT - CSECTF TXT - CSECTG RLDs	Phase 2 CSECTC CSECTD CSECTE
	Phase 3 CSECTG

Figure 5. Module Phase Relationship

Chart 01. Autotest Linkage Editing



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

ESD Type	Definition
SD	Section Definition: provides control section name, assembled origin and length.
PC	Private Code: provides assembled origin and length for an unnamed control section.
LD	Label Definition: specifies the assembled address and the associated SD of a label that may be referred to by another module.
	External Reference: specifies the location of a reference made to another module.
	Common: indicates the amount of main storage to be reserved for common use by different phases.

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

When the Autotest linkage editor reads a module, it stores ESD's in its control dictionary, writes TXT and REP items in core image blocks in the library, writes RLD items on an RLD file, and writes SYM items on a work file. Figure 6 shows the SYM record format. Each item, identified by the language translators with an ESID number, is identified by the linkage editor with a control dictionary number to avoid duplication of identification between modules.

AUTOTEST LINKAGE EDITOR PROGRAM FLOW

The Autotest linkage editor is physically divided into nine phases. The phase names assigned and functions are:

Phase Name	Function
Phase 1 ATLEDT	Initialize/overhead, Chart 02.
Phase 2 ATLEDT10	12-2-9 processor (ESD/initial SYM cards only), Chart 03.
Phase 3 ATLEDT12	12-2-9 processor (other than ESD/initial SYM cards), Chart 04.
Phase 4 ATLEDT14	Control card scanner, Chart 05.
Phase 5 ATLEDT16	Control card processor, Chart 06.
Phase 6 ATLEDT18	MAP processor, Chart 07.
Phase 7 ATLEDT1A	RLD post-processor, Chart 08.
Phase 8 ATLEDT1B	SYM card processor, Chart 09.
Phase 9 ATLEDT1C	Write control dictionary and Autotest communication area on SYSLNK, Chart 10.

The first phase, ATLEDT, is fetched by the job control program. This phase finds the existing machine configuration, checks I/O unit assignments, opens SYSLNK and SYS001, and saves the DTF table. ATLEDT sets up the control dictionary and linkage table, processes any action cards and loads main storage with the card processing phase required by the type of card image found in the input stream. The card processor loaded (12-2-9 or control card) overlays the initialization part of the ATLEDT phase, leaving the overhead portion for use by other Autotest linkage editor phases. When an ENTRY card is finished processing, indicating the end of a linkage editor run, the MAP, RLD post-processor and SYM card processors are fetched and executed followed by the phase that writes the control dictionary on disk (ATLEDT1C). Figure 7 shows an example of the MAP printout. When ATLEDT1C has completed its functions, it fetches the first Autotest phase required for execution of the linkage-edited problem program (ATLEFC1).

Figure 8 shows the I/O flow of the linkage editor program. Autotest linkage editor storage allocation is illustrated in Figure 9.

Each physical record is 196 bytes and has the following format:

Number of Records	From 1 to 13 Logical Records
A	B

A - 1 byte - Bits 0-6 - number of logical records in physical record
Bit 7 - last record indicator
0 - not last record
1 - last record

B - Variable - contains 15-byte logical records each having the following format:

Bytes	Contents
0-7	Symbol
8-9	Phase number
10-12	Symbol address (displacement of symbol within phase)
13-14	Control section (CSECT) number

Figure 6. SYM Record Format

PHASE	XFR-AD	LOCORE	HICORE	DSK-AD	ESD TYPE	LABEL	LOADED	REL-FR
ATLEDT	002348	001800	002BB7	35 7 1	CSECT	IJJCPD1N	001800	001800
					ENTRY	IJJCPD3	001800	
					CSECT	IJVDNK20	001988	000088
					* ENTRY	IJVDNLA	001988	
					* ENTRY	IJVDNK	001988	
					* ENTRY	IJVDOV	0020FC	
					CSECT	IJVDNL20	002348	000088
					* ENTRY	IJVDNL	002348	
ATLEDT10	002348	002348	002777	35 8 2	CSECT	IJVDSD20	002348	-0007E8
					* ENTRY	IJVDSD	002348	
					* ENTRY	IJVDSDND	002778	
ATLEDT12	002348	002348	002867	35 9 1	CSECT	IJVDTH20	002348	-000C18
					* ENTRY	IJVDTH	002348	
					* ENTRY	IJVDTHND	002868	
ATLEDT14	002348	002348	0027A3	35 9 2	CSECT	IJVDCN20	002348	-001138
					* ENTRY	IJVDCN	002348	
					* ENTRY	IJVDCNND	0027A4	
ATLEDT16	002348	002348	0027BF	36 0 1	CSECT	IJVDTL20	002348	-001598
					* ENTRY	IJVDTL	002348	
					* ENTRY	IJVDTLND	0027C0	
ATLEDT18	002100	002100	0026D7	36 0 2	CSECT	IJVDAP20	002100	-001C58
					* ENTRY	IJVDAP	002100	
					* ENTRY	IJVDAPND	0026D8	
ATLEDT1A	002100	002100	002893	36 1 1	CSECT	IJVDLD20	002100	-002230
					* ENTRY	IJVDLD	002100	
					* ENTRY	IJVDLDND	002894	
ATLEDT1B	002348	002348	002757	36 2 1	CSECT	IJVDXL20	002348	-002780
					* ENTRY	IJVDXL	002348	
					* ENTRY	IJVDXLND	002758	
ATLEDT1C	002348	002348	0027EF	36 2 2	CSECT	IJVDDO20	002348	-002B90
					* ENTRY	IJVDDO	002348	

Figure 7. Example of MAP Printout

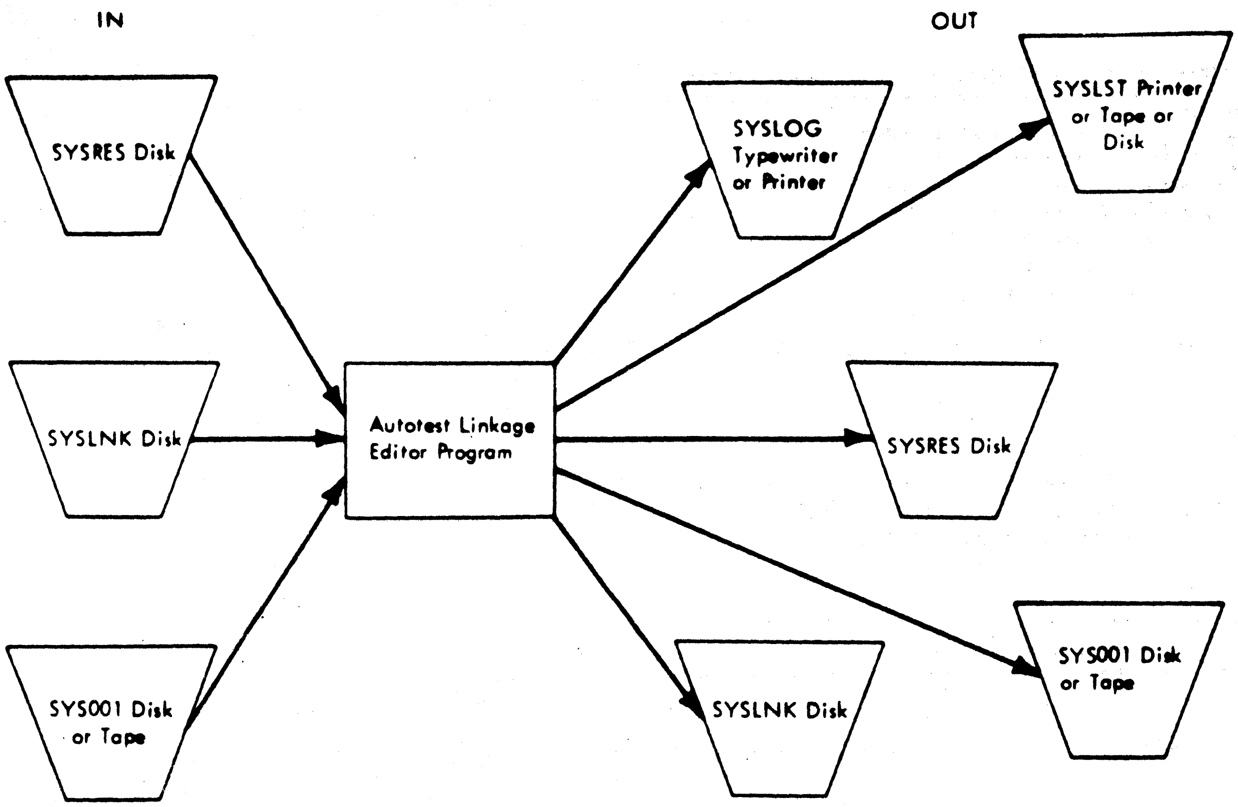


Figure 8. Autotest Linkage Editor I/O Flow

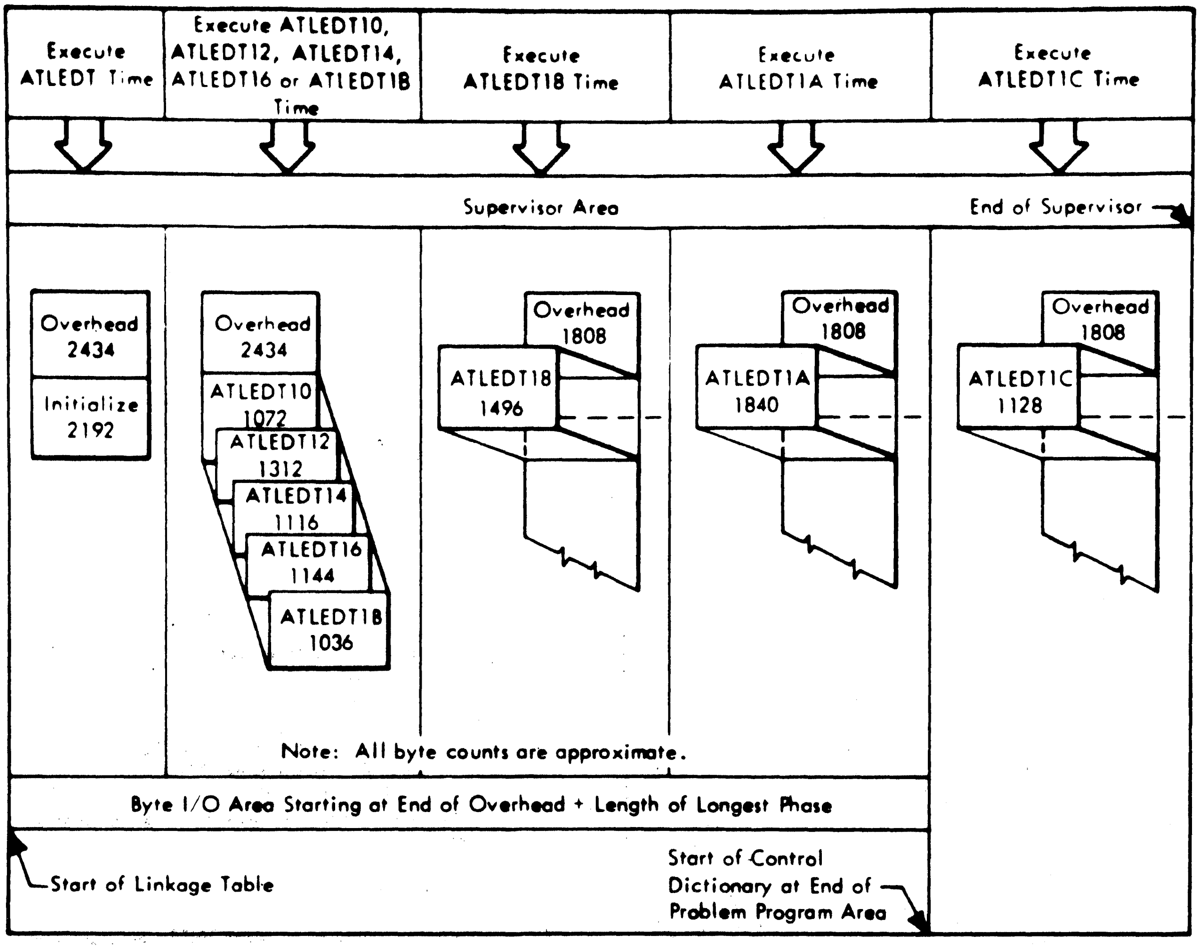


Figure 9. Autotest Linkage Editor Storage Allocation Map

KEY CONCEPTS

Overhead Processor

In addition to the initialization steps, the first phase of the linkage editor (ATLEDT) contains most of the subroutines used by the various other linkage editor phases. These are often called overhead and, therefore, this part of ATLEDT phase is often called the overhead processor. The subroutines in the first phase are labeled:

Subroutine	Use
RDS000	Reads blocked input.
LTESID	Finds control dictionary information and the relocation factor by using the linkage table.
SRCHCD	Searches the control dictionary for a matching label.
ALIGN	Aligns an address on a doubleword boundary.
CHVHEX	Converts hexadecimal input to binary output.
PRINT	Performs print and carriage control operations.
AD1DSK	Updates linkage addresses by using predetermined overflow factors to optimize the update operation.
XTPHNO	Extracts the phase number from a control dictionary entry.
READ/WRITE	Reads or writes core image library blocks.
ABTERR	Fetches the ATLEDT1A phase for abnormal termination (abort) error handling.

Subroutine

CDSIZE	Checks for control dictionary--linkage table overlap.
RDNEXT	Reads the input stream.
ALNKPR	Initializes for a scan of the relocatable directory, when the autolink feature is not suppressed. Puts all unresolved ER's, found in the control dictionary, into the correct collating sequence.
ERROR	Sets up to print non-abort error messages.
OVRLAY	Performs print and carriage control operations when a NOMAP option is found (overlays the first part of the print subroutine).

CONTROL DICTIONARY

The control dictionary is an internal linkage editor mechanism used to tabulate phase and external symbol dictionary information. It is composed of a variable number of fixed 16-byte entries. Each entry is numbered sequentially, and therefore, the physical structure of the control dictionary roughly outlines the structure of the program. Valid new entries (Phase or ESD) are posted when they are found by the ESD processing routines. Location CDENT1 contains the address of the first entry. Location CTLDAD contains the address of the last entry. The label CDENT1 is located in a high storage location relative to the label CTLDAD, because the control dictionary is built in reverse order. Figure 10 illustrates the control dictionary.

LINKAGE TABLE

The linkage table is an internal linkage editor mechanism used to link the ESID number supplied by the language translator output to the corresponding control dictionary number that belongs to a control dictionary entry. This table is composed of a variable number of fixed 3-byte entries. It is built separately for each object module. When an END card is processed, signaling the end of a module, the table is reset to zeros. (Location LTMIN3 contains the address of the first item in the linkage table minus 3 bytes. LNKTAD contains the address of the last item in the linkage table plus 3 bytes. The label LTMIN3 is located in a low storage location relative to the label LNKTAD. Figure 10 illustrates the linkage table.)

Control dictionary - Each physical record is 210 bytes and has the following format:

Code	Number of Records	From 1 to 13 Logical Records
A	B	C

A - 1 byte containing a C
B - 1 byte - Bits 0-6 - number of logical records in physical record
Bit 7 - last record indicator
0 - not last record
1 - last record
C - Variable - contains 16-byte logical records each having the following format:

Byte(s)	Contents
0-7	Name of phase or control section or blanks if private code (PC)
8	Bits 0-3: 4 high-order bits of phase number Bit 4: sign of relocation factor 0 = plus 1 = minus Bits 5-7: Type of entry 000 = Section definition (SD) 100 = Private code (PC) 111 = Phase (PH)
9-11	Load origin of phase or assembled origin of control section
12	Phase number
13-15	Highest load address in phase ENTRY or relocation factor for control section entry.

Linkage Table

Control Dictionary Number	ESD Type
2 Bytes	1 Byte

Figure 10. Control Dictionary/Linkage Table

Use of the Linkage Table and Control Dictionary

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

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

2. Go to the linkage table.

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

4. Go to the control dictionary entry.

5. Extract the relocation factor.

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

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

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

9. Get the proper core image block.

10. Put the text into the core image block.

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

Linkage Editor Fundamental Calculations

For the examples in this presentation:

- A/O = assembled origin
- R/F = relocation factor
- L/O = load origin
- P/O = phase origin

Example 1: The assembled origin of the CSECT being processed is subtracted from the address that is the next possible phase origin. This results in the relocation factor for that control section.

P/O - A/O = R/F

Example 2: The language translator provided A/O is added to an R/F that has been determined by the phase origin information. The result, the L/O, is the main-storage address that is the physical location of this text, RLD item, or control section.

A/O + R/F = L/O

Figure 11 shows the formulas given in examples 1 and 2. In this illustration when CSECT1 is being linkage-edited, the phase origin (P/O) has been established by the PHASE card at X'1800'. Suppose CSECT1 was assembled as a separate module with an assembled origin at X'000'. The relocation factor R/F is then:

P/O - A/O = R/F 1800 - 000 = 1800.

All assembled addresses in CSECT1 must be increased (relocated) by 1800 bytes. If CSECT2 had been assembled with CSECT1 as a single module, its R/F would also be 1800.

Consider the case where CSECT2 is assembled as a module by itself with an assembled origin of X'000'. The same formula for R/F still applies, but it must be noted that the address for the next possible phase origin has been updated by the length of CSECT1; that is, X'1800'+300=1B00. 1B00-000=1B00 is the relocation factor for all addresses within CSECT2.

Had CSECT2 been assembled starting at X'700', the R/F would be '1B00'-'700'= '1400'. An assembled origin greater than 1B00 would result in a minus R/F, moving CSECT2 downward in storage directly behind CSECT1. While CSECT2 is being processed, the next possible phase origin address has been updated by 400 bytes to X'1F00'.

The other formula, assembled origin (A/O) + relocation factor (R/F) = load origin (L/O) may now be tested. Within CSECT2 is LABELY, which is displaced 300 bytes from the beginning of CSECT2. If CSECT2 were assembled beginning at origin address X'700', LABELY's assembled address would be '700'+'300'='A00' and we have (A/O) 'A00' + (R/F) '1400' = L/O '1E00'. A glance at the storage map shows that 1E00

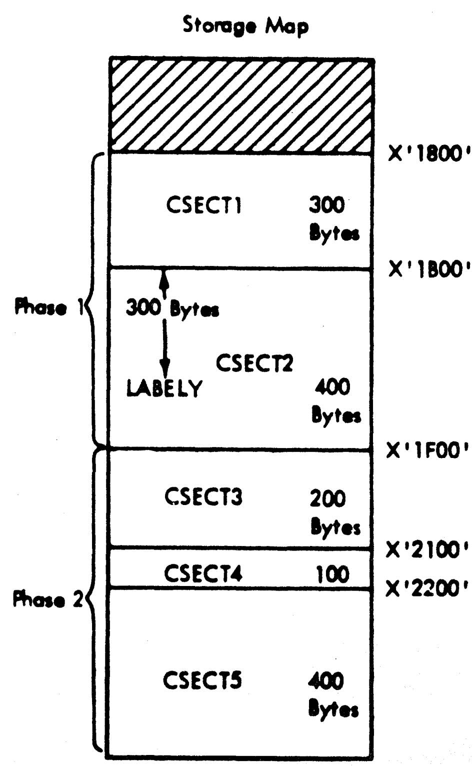


Figure 11. Illustration for Use in Computing Linkage Editor Formulas

is indeed the desired address for LABELY to be loaded at, once the program is linkage-edited and ready for execution.

Example 3: Current control dictionary entry - 16 = next control dictionary entry.

Example 4: Current linkage table entry + 3 = next linkage table entry.

Example 5: Disk address + overflow factors = updated disk address.

The overflow factors are a constant, established by the programmer, that simplifies the updating of disk addresses (CHHR). These factors, when added to the disk address, provide the correct cylinder and head after only one calculation.

Use of the Autolink Feature

This feature tries to locate a module in the relocatable library for any unresolved ER's found in the preceding phase. The signal indicating a phase has finished processing is either a new PHASE card or an ENTRY card. When the signal is detected, autolink is attempted unless the feature has been suppressed by a NOAUTO phase card or action card option.

Example of Autolink with LIOCS

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

Summary of Autotest Modifications to the Linkage Editor

Figure 12 summarizes the changes made by Autotest to the system linkage editor.

Label for Autotest Routine	Entry from Linkage Editor	Function
@ATINIT	In ATLEDT phase, from open routine for SYSLNK and SYS001.	The upper and lower extent limits of the work file area on SYSLNK are retrieved from the DTFPH for a 2311 output file named IJSYSAT (IJVWRK). IJSYSAT is checked to be certain this disk device is assigned the symbolic name SYSLNK. If not SYSLNK, a message '9900I Disk Work Area Invalid' is written on SYSLOG, and the job is canceled. The track limits for this work file are set according to the size of the machine (16K or 32K), and the EOJ dump limits are determined.
@OK1	In ATLEDT phase, from ATINIT routine.	Sets lower extent limits of this SYSLNK work file as the start address for writing the SYM card records on SYSLNK and saves this address at @STAR. A comparison to the EOJ address previously determined is made. If the EOJ address is exceeded at any time, the job is canceled and the message '9900I Disk Work Area Invalid' is written on SYSLOG.
@LAB1	In ATLEDT phase - entered as part of INTPT1 routine.	Job control switch 1 in the supervisor communications region is tested to see if the 'Catalog Linkage Editor Output' option bit is on. If on, it is turned off and a message is written on SYSLOG stating '9A02I OPTION CATAL IGNORED'. Then the program proceeds normally as if the option bit had been off when tested. This is done because the Autotest linkage editor program never catalogs its output. This routine also checks to determine if <u>all</u> of main storage is allocated to Autotest. If not, the job is canceled
@RDSW	In ATLEDT phase - used by ATLEDT1B phase	This is a branch instruction used as a switch by the ATLEDT1B phase to return to the desired point in the phase, after a card image has been read by the RDNEXT subroutine during SYM card processing.
@ESD	In ATLEDT10 phase - entered as part of ESDNXT routine.	ESDNXT checks input card images for presence of SYM cards. If a SYM card image is not present, a branch is taken to label @ESD for normal ESD card processing. When a SYM card is encountered, it is checked to see if it is the first SYM card encountered. If it is, the switch at label @SYMSW is turned on and the address of this first SYM card is saved for use by the SYM processing phase, ATLEDT1B. If this is not the first SYM card, the switch will have previously been set and the address saved; therefore, a branch is taken to read the next card image.
ENDPRC	In ATLEDT12 phase, entered from OTHTYP routine when a card image in the input stream is not identified as a TXT, RLD, or REP item.	The switch labeled @SYMSW is tested to see if any SYM cards were detected and saved for processing. If not, a branch to @NOSYM is taken for END card processing. If @SYMSW had been turned on at the time the first SYM card was encountered, it is now turned off and modified to indicate SYM items are present. The SYM processing phase (ATLEDT1B) is then fetched.
@RDDSK	In phase, ATLEDT1A, from BLKHDR routine.	A block of the phase directory, containing 18 entries, is read into main storage. If the desired phase is not found in the phase directory, the block is read from the core image directory. (The end of these directories is checked for by the @CHKEND portion of this routine.) The entries are searched for phases with names whose first 6 characters are ATLEFC or greater, because the required Autotest phases are ATLEFC1 and greater. If the phase name is less than ATLEFC, the @UPDAT routine is used to point to the next entry in the block. When all 18 entries of the current block are checked, the disk address is updated and the next block of the phase directory (or core image directory) is read. When a phase name equal to or greater than ATLEFC is found, the entire name is compared with the names of the actual phases being sought; the names of the actual phases are listed starting at the label @TABLE. On an equal compare, a branch to @FOUND is taken. The @FOUND routine moves the information from the directory entry into the appropriate location within the @TABLE area. This process continues until the end of the phase or core image directory is reached. By this time the @TABLE entries should be completely built. However, a check is made against the first phase listed in the table to be certain it has been found and the phase information has been placed there. If it is not found, the job is canceled. The routine at @SER4 is used to insert an asterisk in the last entry position of this block of phase headers just constructed. This block of Autotest phase headers is written on disk in the librarian work area of SYSRES. At user execution time, the job control program moves the block(s) to the phase directory. The result is improved Autotest performance due to faster phase retrieval.
IJVDXL	Entry to phase ATLEDT1B which is called by the ATLEDT phase for SYM card processing.	The entire ATLEDT1B phase processes SYM card images and constructs SYM records which are written on the Autotest work file area of SYSLNK. These blocked, 15-byte records contain the actual label (symbol), the phase number in which the symbol occurs, the symbol address (displacement of symbol within phase), and control section number in which the symbol occurs.

Figure 12. Summary of Autotest Modifications to the Disk Operating System Linkage Editor (Part 1 of 2)

Label for Autotest Routine	Entry from Linkage Editor	Function
IJVDDO	Entry to phase ATLEDT1C which is called by phase ATLEDT1A	This phase checks to see if any errors were detected during SYM card processing. If there were any errors, it writes a message on SYSLST. If SYM cards were present and processed without error, an indicator in the Autotest communications area is set to record that fact. The machine is checked for presence of the floating-point optional feature. If present, another Autotest communications area indicator is set. Control dictionary entries, built by a previous phase, are moved as 16-byte logical records to an output area where they are blocked. Full blocks are written on SYSLNK following the SYM records on the Autotest work file, but only after checking to see if there is space available on the disk file for them. Should there be no room, the message 'DISK WORK AREA TOO SMALL' is written on SYSLOG and SYSLST. The size of the control dictionary is calculated. This information and the disk address limits of the SYM and control dictionary records are stored in the Autotest communications area. Other disk addresses, storage addresses, and data that will be required by Autotest phases are saved in this communications area. Finally, this 116-byte Autotest communications area is moved to the end-of-supervisor address and a call is issued for phase ATLEFC1.

Figure 12. Summary of Autotest Modifications to the Disk Operating System Linkage Editor (Part 2 of 2)

Chart 02. Autotest Linkage Editor--Initialization Phase (ATLEDT)

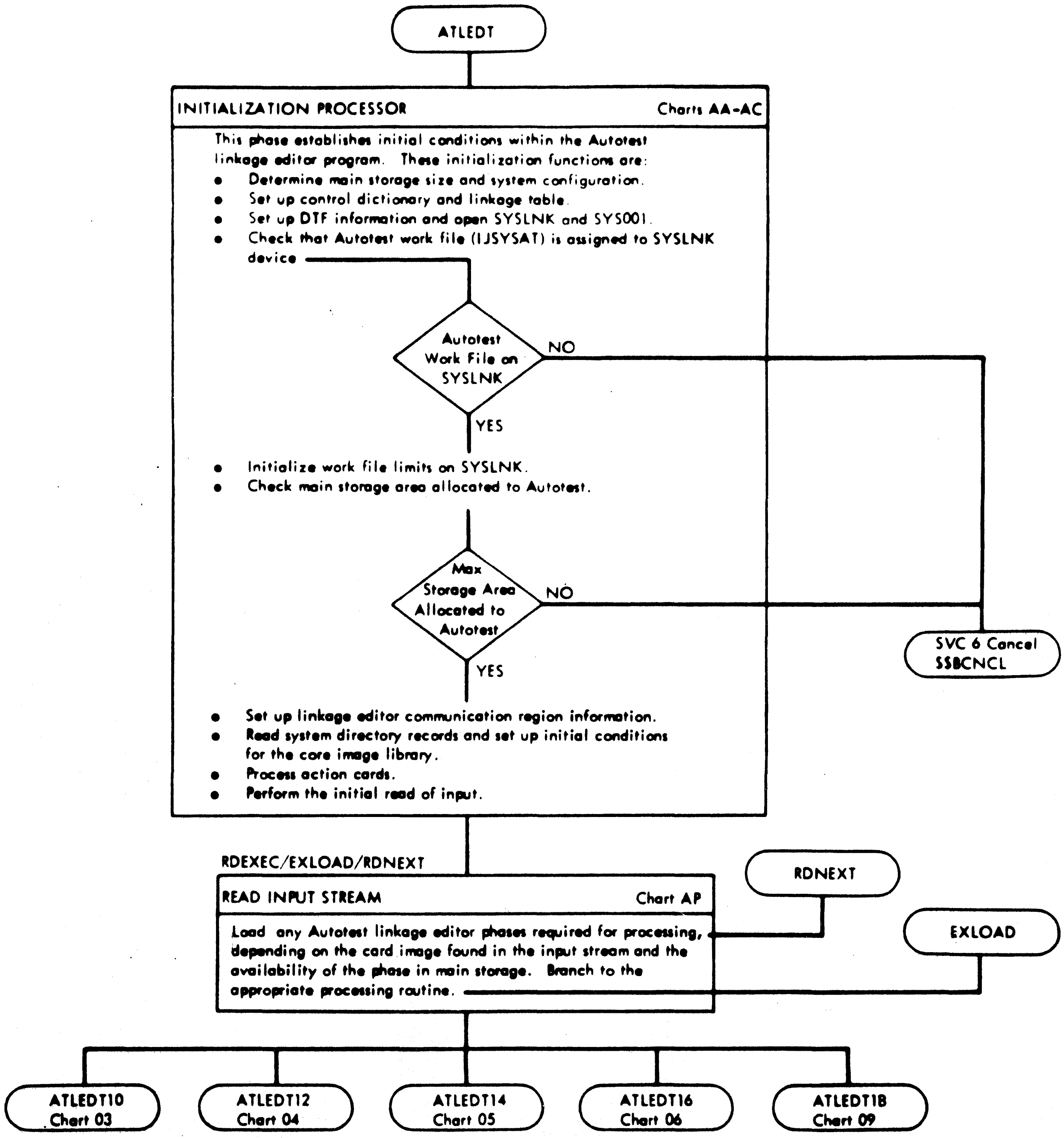


Chart 03. Autotest Linkage Editor--ESD/Initial SYM Card Processor (ATLEDT10)

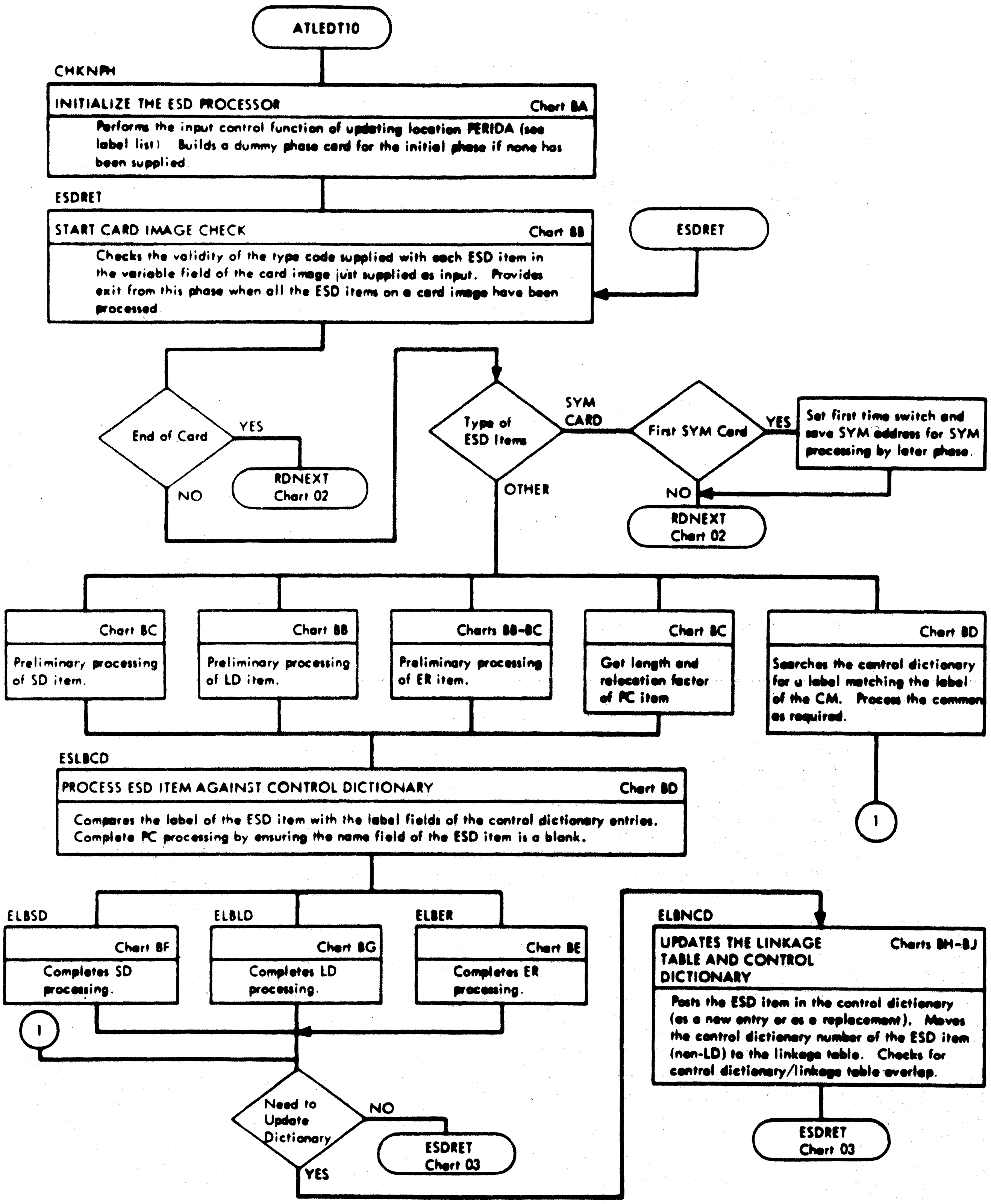


Chart 04. Autotest Linkage Editor--TXT/REP, RLD, and END Card Processor (ATLEDT12)

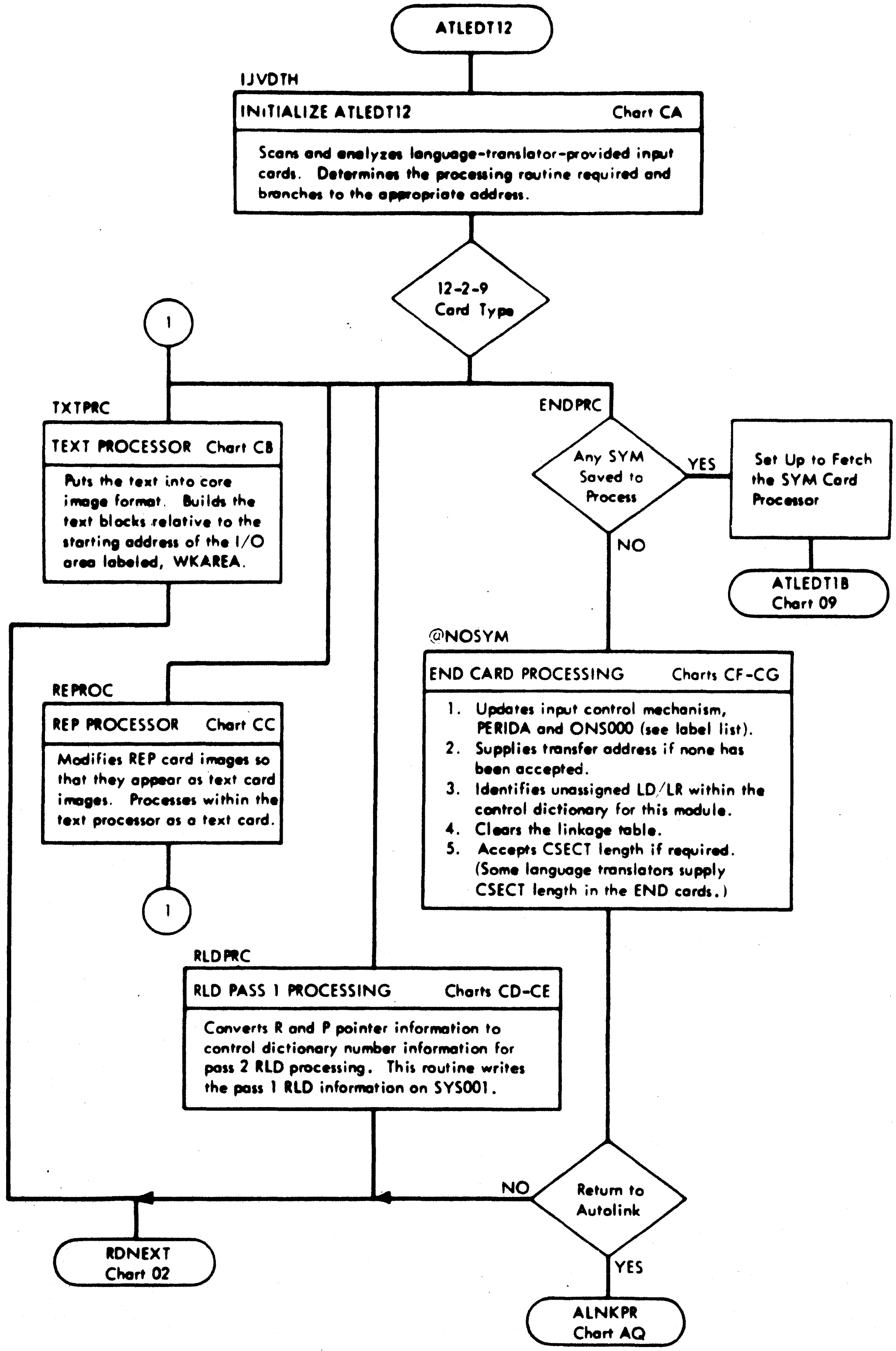


Chart 05. Autotest Linkage Editor--Control Card (INCLUDE, PHASE, and ENTRY) Scanner (ATLEDT14)

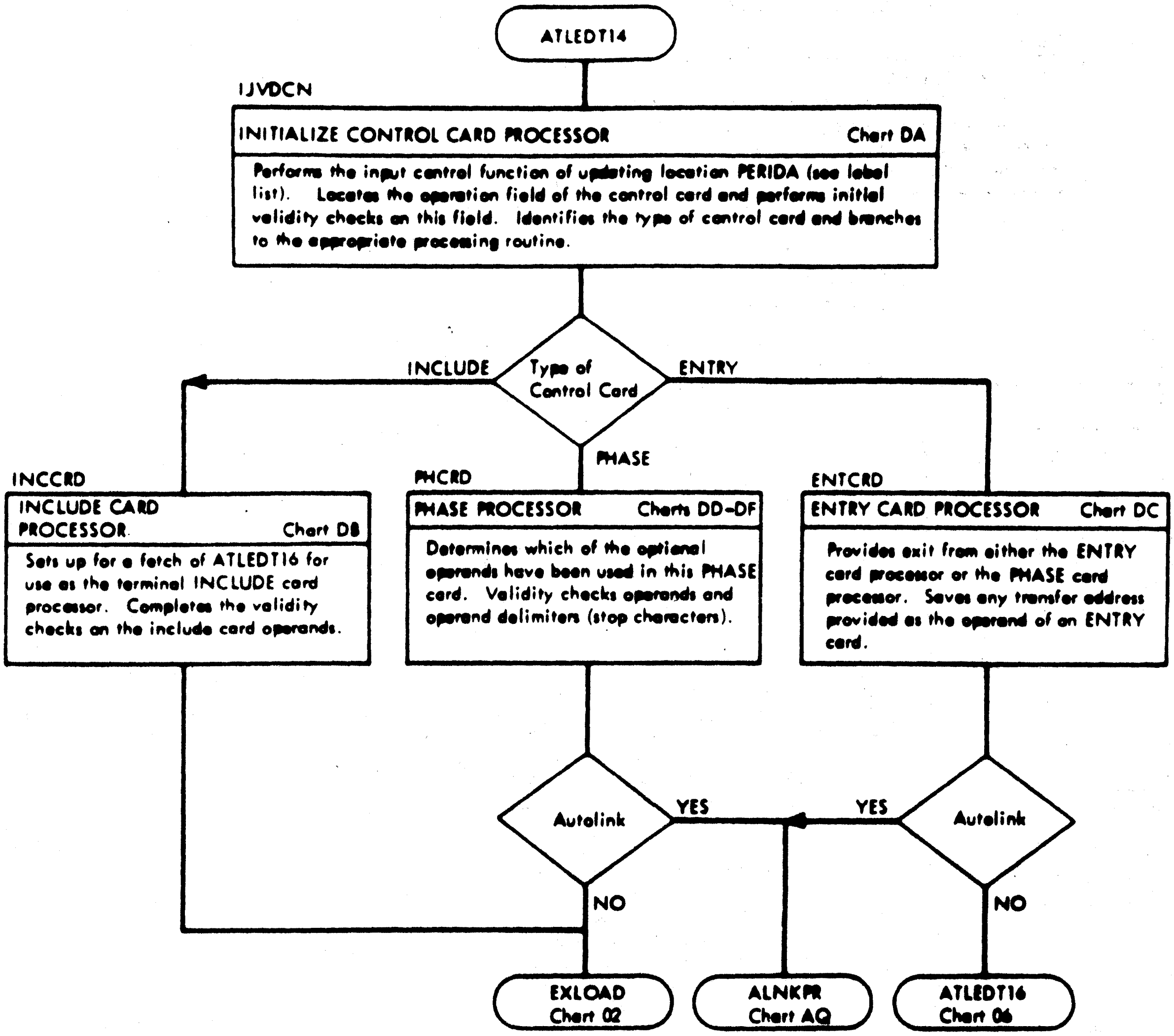


Chart 06. Autotest Linkage Editor--Control Card Processor (ATLEDT16)

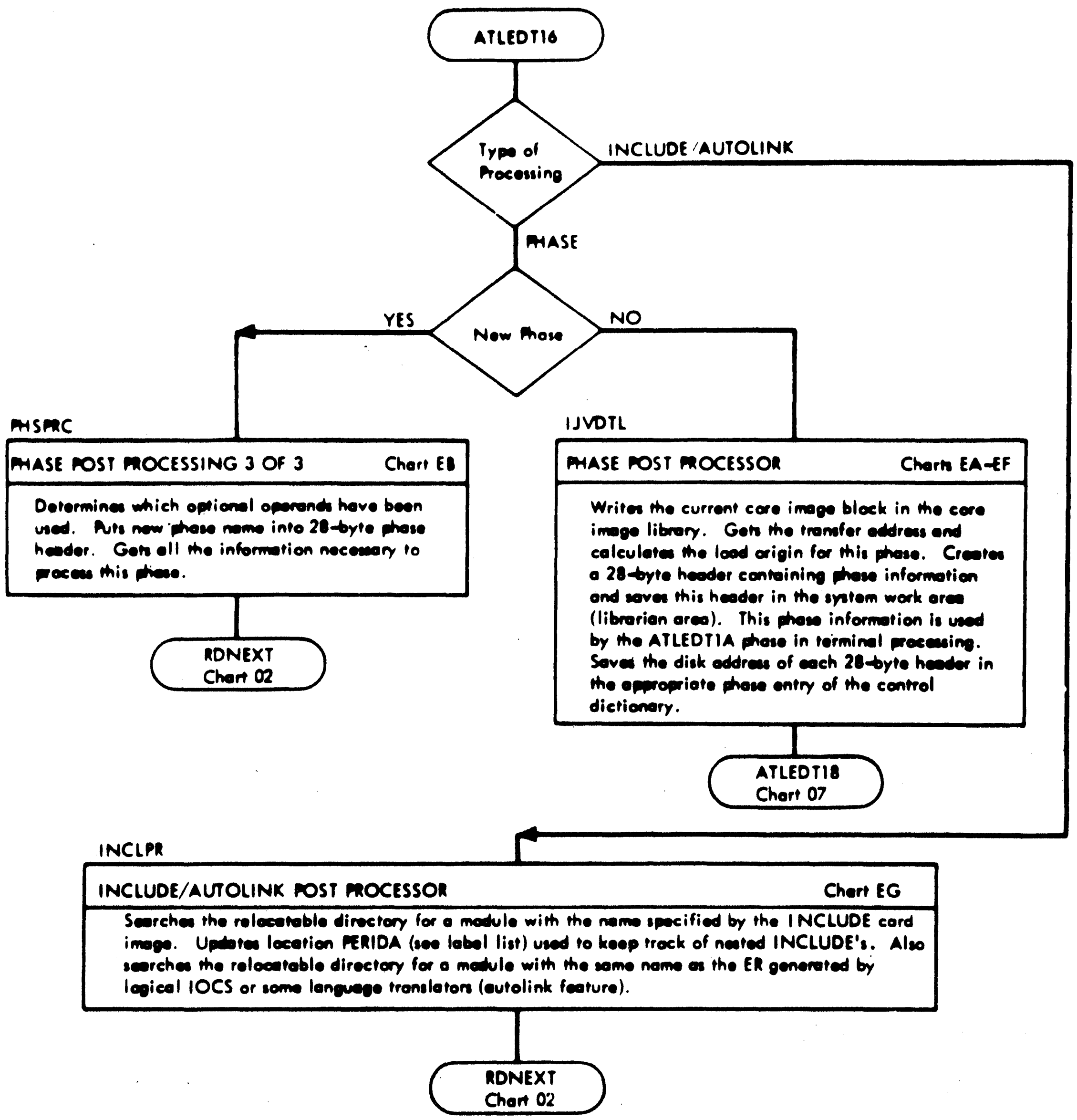


Chart 07. Autotest Linkage Editor--MAP Processor (ATLEDT18)

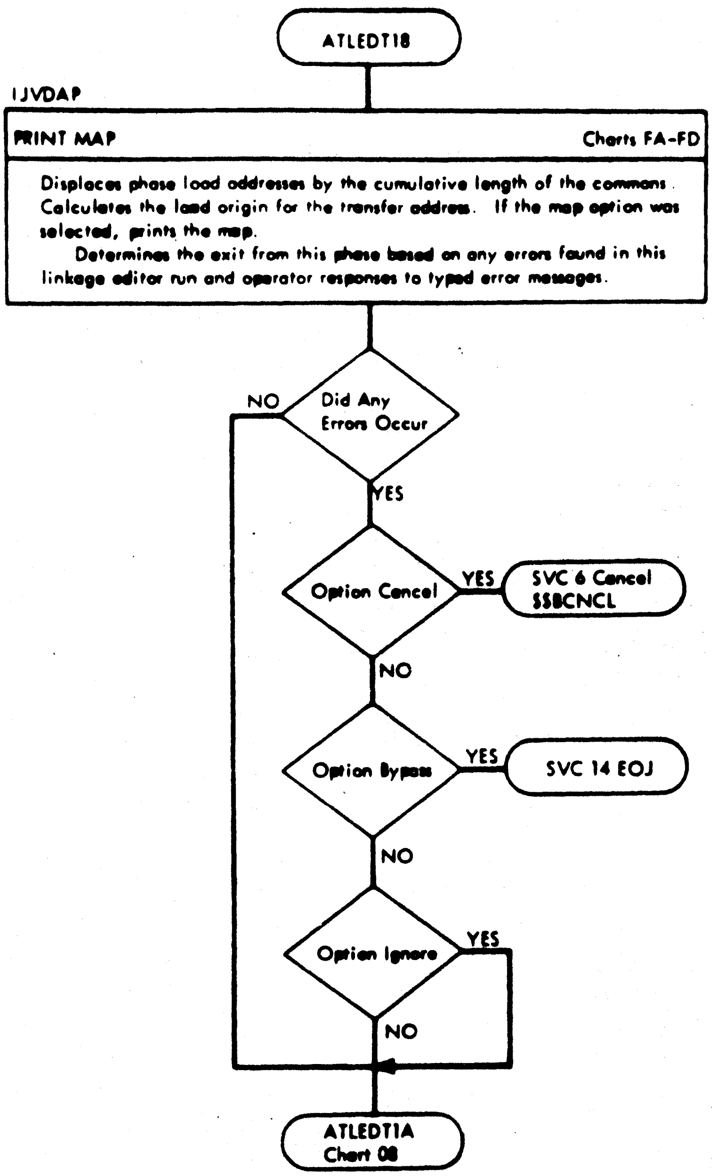


Chart 08. Autotest Linkage Editor--RLD Post-Processor (ATLEDT1A)

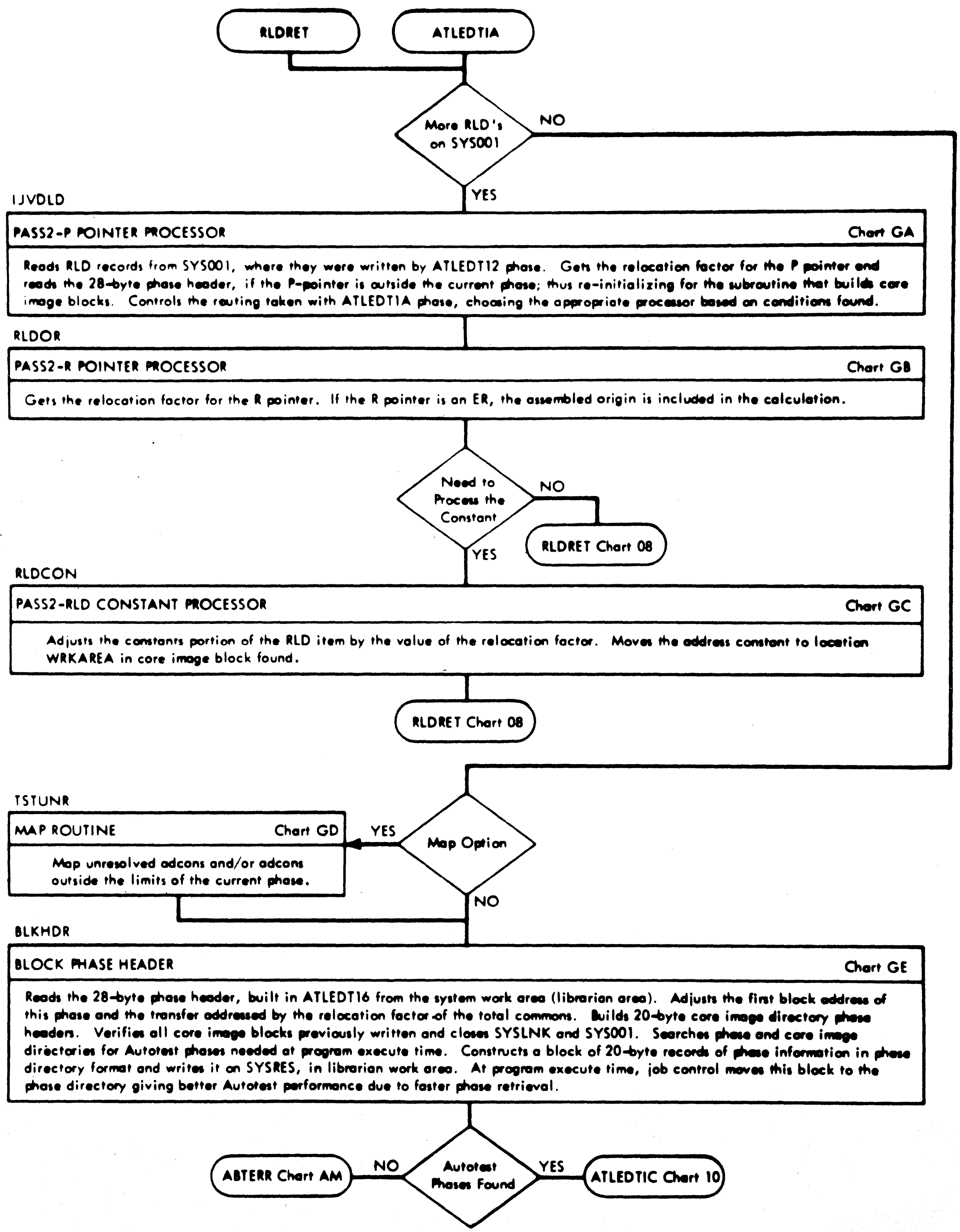


Chart 09. Autotest Linkage Editor--SYM Card Processor (ATLEDT1B)

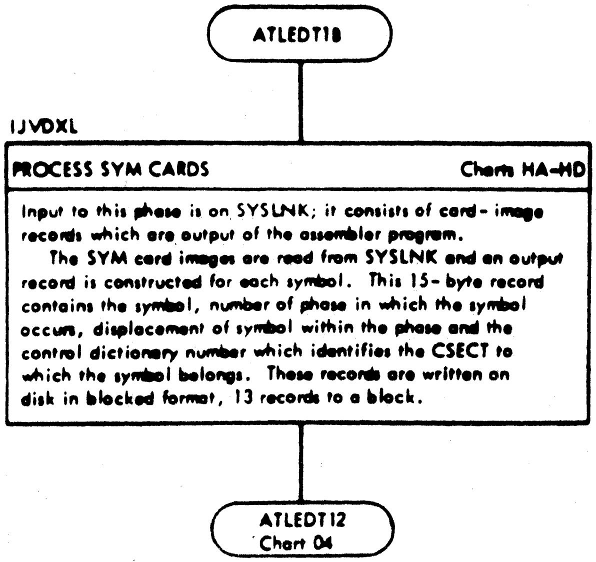
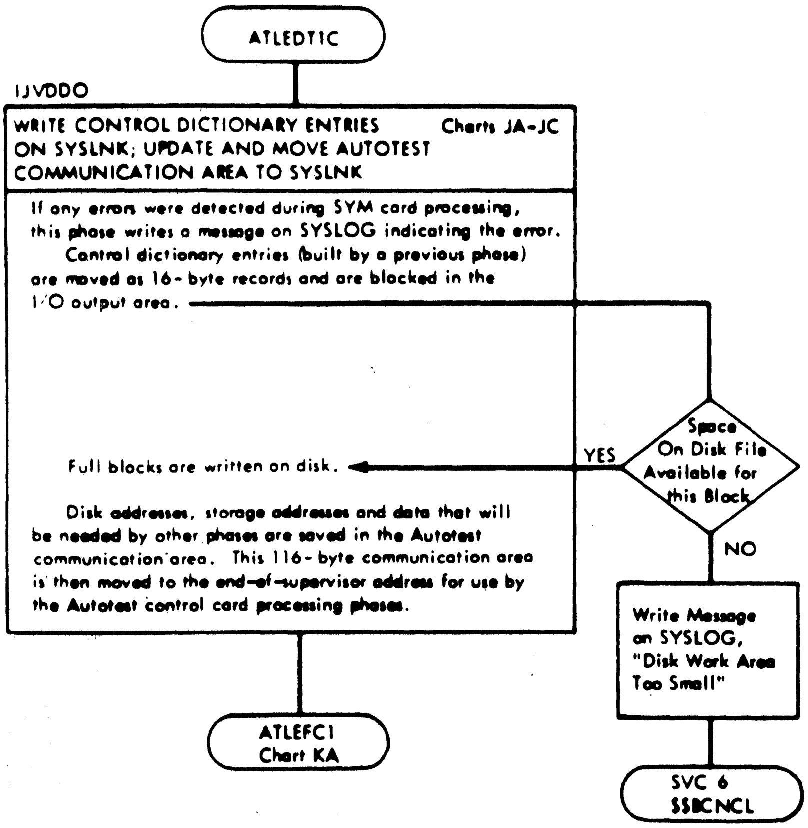


Chart 10. Autotest Linkage Editor--Write Control Dictionary and Autotest Communication Area on SYSLNK (ATLEDT1C)



The control-card-analysis phases read and analyze the following Autotest control cards:

1. PHQ (phase qualification)
2. CSQ (control section qualification)
3. Autopatch
   a. ADD (add)
   b. EXC (exchange)
   c. CON (replace constants)
4. TR (test request)
5. PCC (program control)

Figure 13 is an example of an Autotest control card diagnostic printout. Figure 14 shows the I/O flow for the control-card-analysis phases. Figure 15 gives the storage allocation for the control-card-analysis phases.

The Autotest linkage editor fetches the first control card analysis phase, ATLEFC1, into main storage for execution. ATLEFC1 performs the necessary initialization for reading, analyzing, and processing the Autotest control cards. It defines the work area, constants, and switches to be used by all the other control card analysis phases. Figure 16 shows the common switch settings for the control-card-analysis phases. These constants are stored at the beginning of the problem program area (following the Autotest communication area). The 1728-byte area containing the core image library records (CILAREA) follows the constant area in main storage. The logic for the other control card analysis phases follows the CILAREA in main storage.

ATLEFC1 fetches ATLEFC2 to read a card from SYSIPT. Depending on the type of card read, the following functions are performed:

Phase Qualification (PHQ)

When a PHQ card is read, the phase name is stored and all addresses on subsequent control cards are considered as belonging to the phase named on the PHQ card until another PHQ card is read. PHQ cards are optional and need be used only as required. If a module consists of only one phase, a PHQ card is not required. If there is no PHQ card and more than one phase is supplied, the first phase (assumed phase) processed by the linkage editor is used by Autotest to resolve addresses. The PHQ card must precede all autopatch and test request control cards for the phase involved. The last PHQ card read by Autotest governs all further address resolutions. ATLEFC2 and/or ATLEFC3 process PHQ cards.

```
SVC#  TR#                          CARD-IMAGE                                       PPA-TPA     ERROR/S
254   001  ./ TR CARD,,DPL,REG3,TOT3+3,,B,G                                         002A5A
           ./ TR HDRRAT+11,,DSP,ZERO,ZERO+3                                                     K
253   002  ./ TR HDRRAT,,DSP,ZERO,ZERO+3                                            00300E
252   003  ./ TR X'2F4',,PNL,,GSF                                                   0022F4
           ./ TR X'50',,DSP,X'50',X'88',(1,1,1),B,                       X          002050      *
                           PNL,(1,2,5),R,                                X                      R
                           PNL,(2,1,4),G,                                                       M
           ./ TR MXT3,,DSP,MXT3,PRTOUT,,B                                                       L K L O
           ./ CON X'165',45                                                         002165
                  CHAR D                       C44040                                           D
           ./ ADD FETC,2                                                            002088
            *  1844                                                                             E
            *1855
           ./ CON X'213F',1                                                         00413F
           *C24040
           ./ PCC MAS,MAS,2000
```

Figure 13. Example of Autotest Control Card Diagnostics

Chart 11. Control Card Analysis (Part 1 of 4)

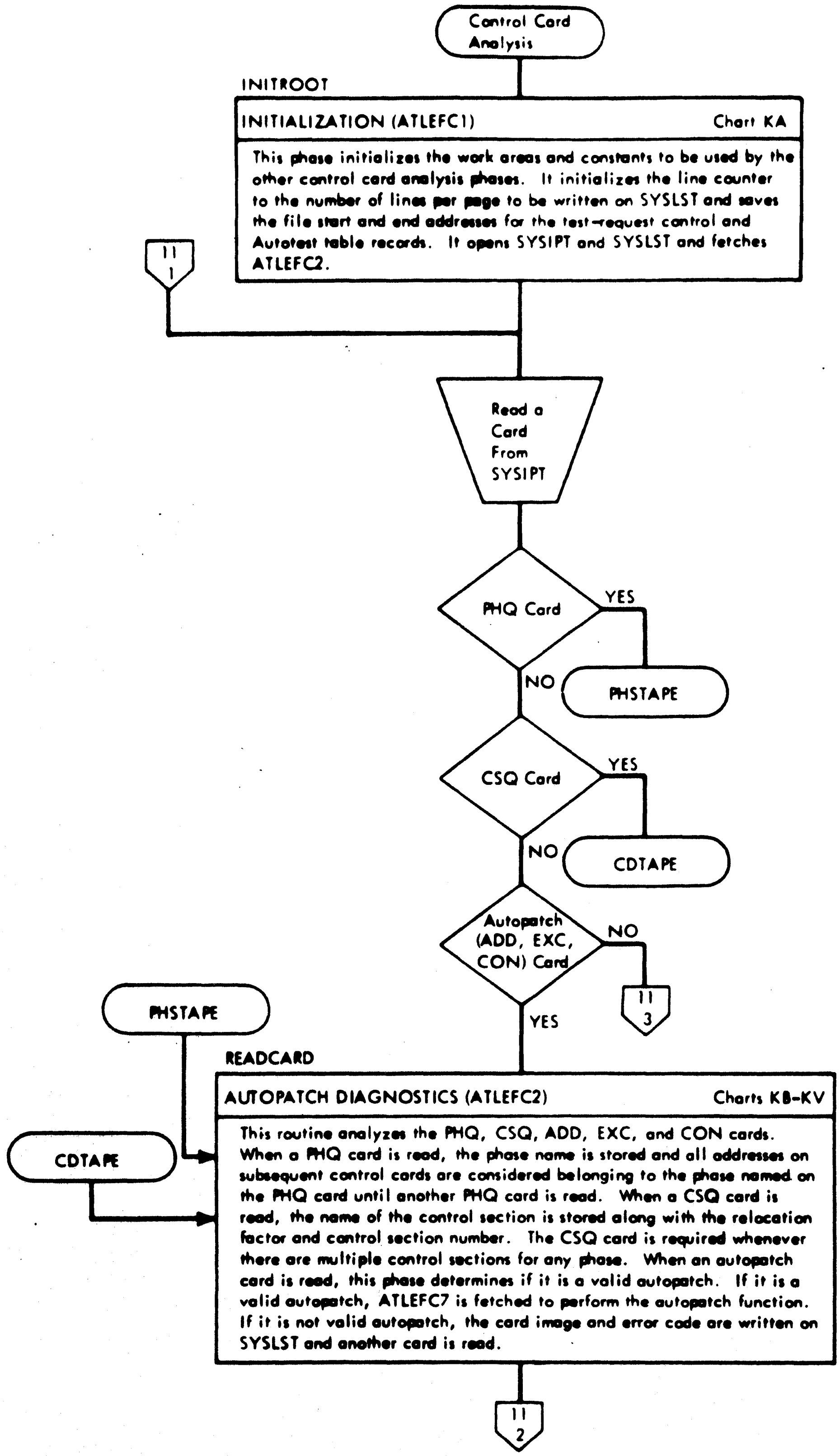
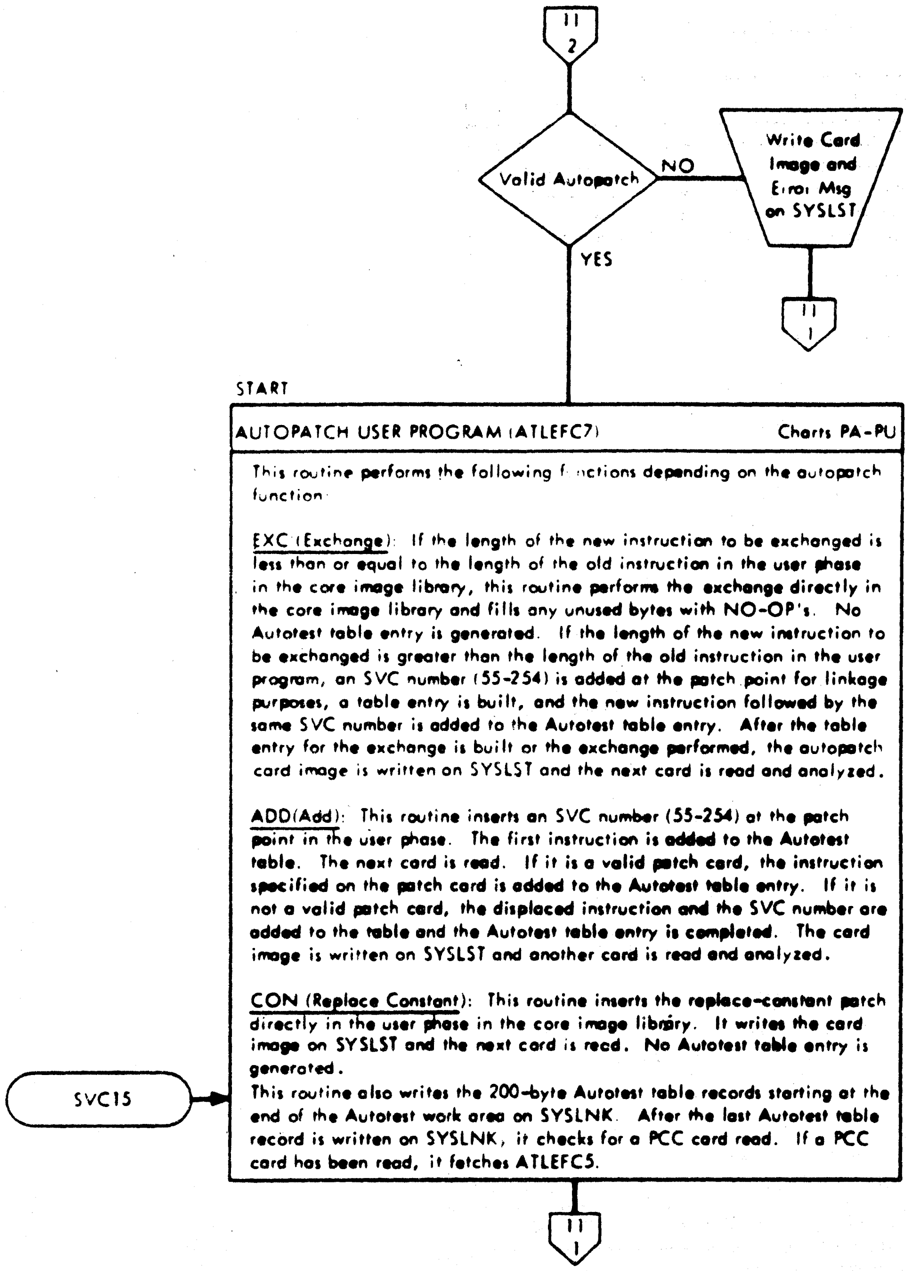
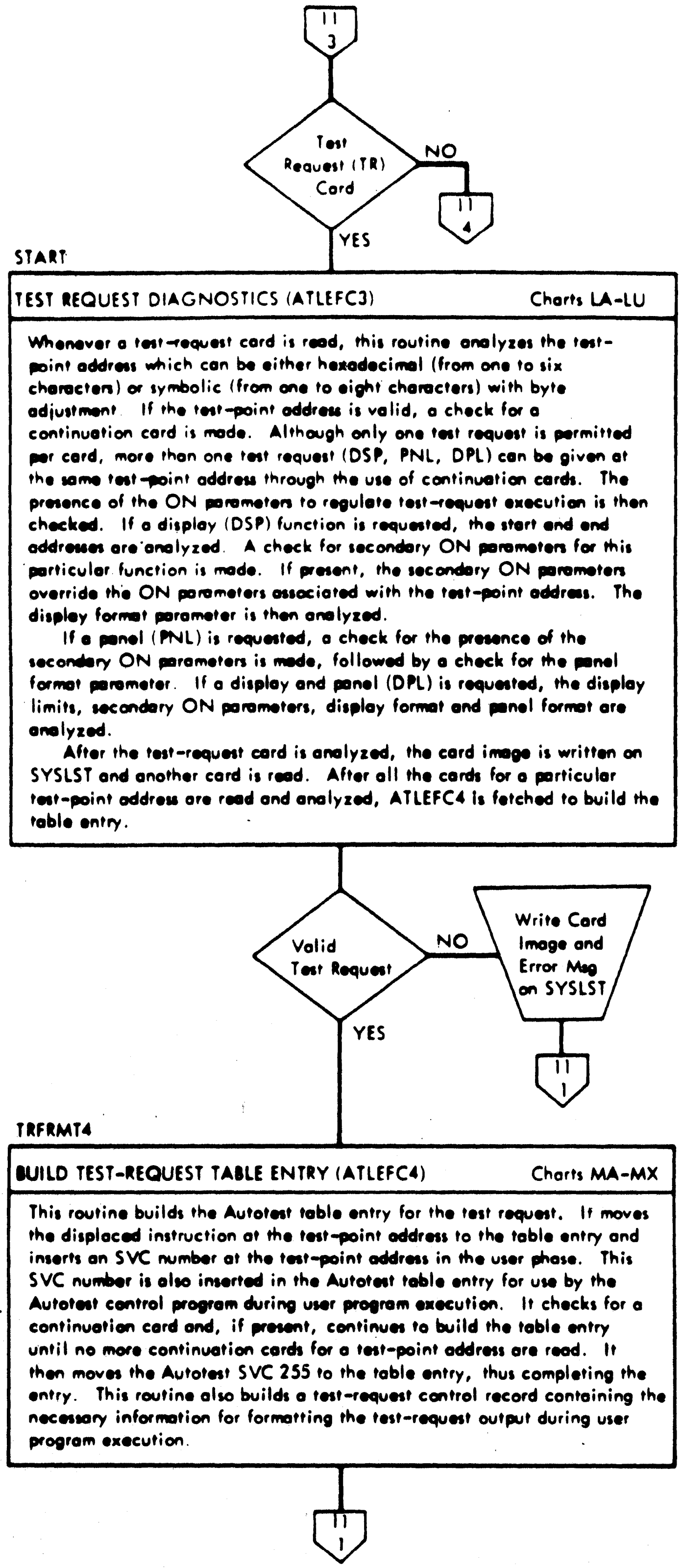
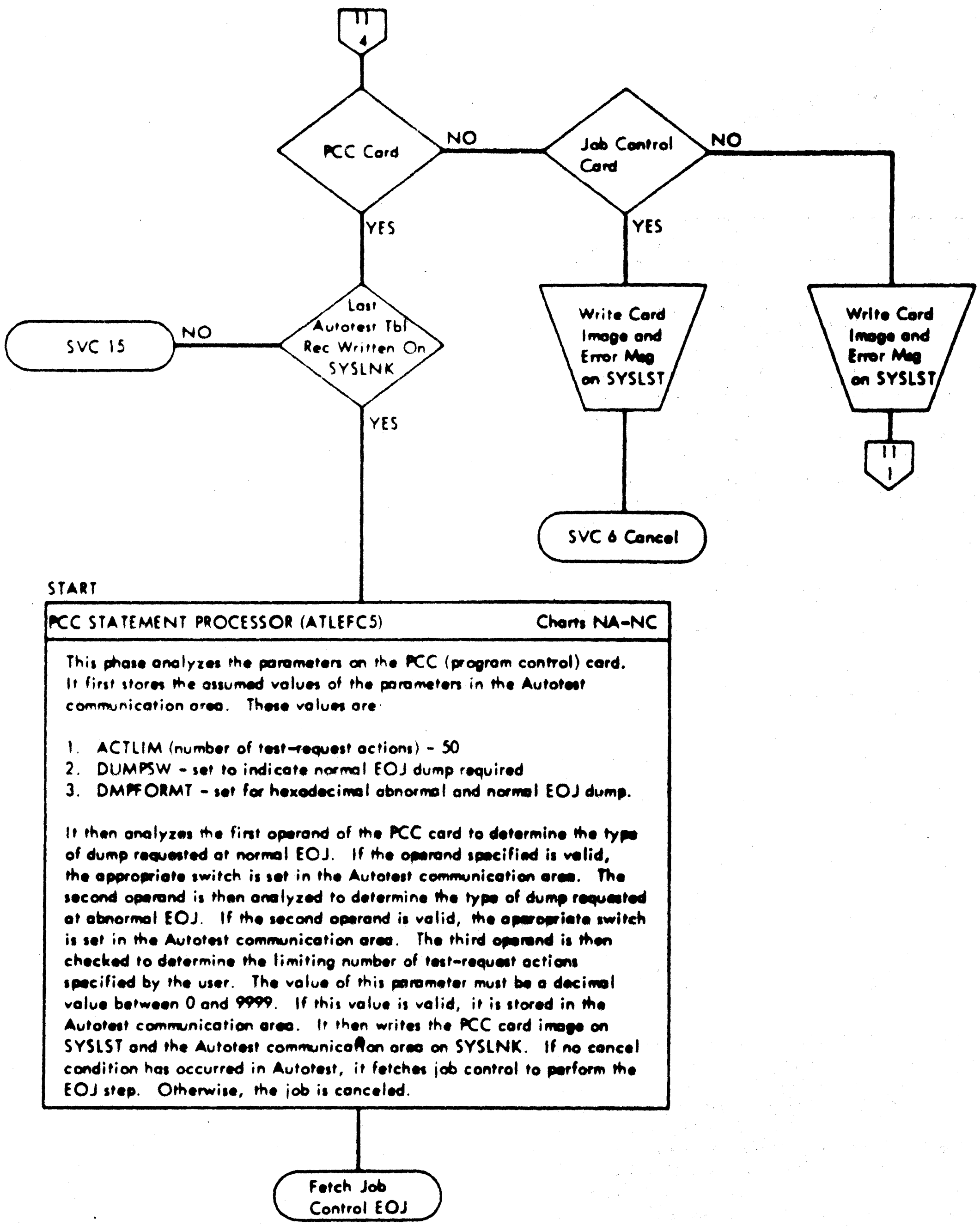


Chart 11. Control Card Analysis (Part 2 of 4)





11
3
Test Request (TR) Card
NO
11
4
YES
START
TEST REQUEST DIAGNOSTICS (ATLEFC3)
Charts LA-LU
Whenever a test-request card is read, this routine analyzes the test-point address which can be either hexadecimal (from one to six characters) or symbolic (from one to eight characters) with byte adjustment. If the test-point address is valid, a check for a continuation card is made. Although only one test request is permitted per card, more than one test request (DSP, PNL, DPL) can be given at the same test-point address through the use of continuation cards. The presence of the ON parameters to regulate test-request execution is then checked. If a display (DSP) function is requested, the start and end addresses are analyzed. A check for secondary ON parameters for this particular function is made. If present, the secondary ON parameters override the ON parameters associated with the test-point address. The display format parameter is then analyzed.
If a panel (PNL) is requested, a check for the presence of the secondary ON parameters is made, followed by a check for the panel format parameter. If a display and panel (DPL) is requested, the display limits, secondary ON parameters, display format and panel format are analyzed.
After the test-request card is analyzed, the card image is written on SYSLST and another card is read. After all the cards for a particular test-point address are read and analyzed, ATLEFC4 is fetched to build the table entry.
Valid Test Request
NO
Write Card Image and Error Msg on SYSLST
11
1
YES
TRFRMT4
BUILD TEST-REQUEST TABLE ENTRY (ATLEFC4)
Charts MA-MX
This routine builds the Autotest table entry for the test request. It moves the displaced instruction at the test-point address to the table entry and inserts an SVC number at the test-point address in the user phase. This SVC number is also inserted in the Autotest table entry for use by the Autotest control program during user program execution. It checks for a continuation card and, if present, continues to build the table entry until no more continuation cards for a test-point address are read. It then moves the Autotest SVC 255 to the table entry, thus completing the entry. This routine also builds a test-request control record containing the necessary information for formatting the test-request output during user program execution.
11
1


11
4
PCC Card
NO
Job Control Card
NO
YES
YES
Last Autotest Tbl Rec Written On SYSLNK
NO
SVC 15
YES
Write Card Image and Error Msg on SYSLST
Write Card Image and Error Msg on SYSLST
11
1
SVC 6 Cancel
START
PCC STATEMENT PROCESSOR (ATLEFC5)
Charts NA-NC
This phase analyzes the parameters on the PCC (program control) card. It first stores the assumed values of the parameters in the Autotest communication area. These values are:
1. ACTLIM (number of test-request actions) - 50
2. DUMPSW - set to indicate normal EOJ dump required
3. DMPFORMT - set for hexadecimal abnormal and normal EOJ dump.
It then analyzes the first operand of the PCC card to determine the type of dump requested at normal EOJ. If the operand specified is valid, the appropriate switch is set in the Autotest communication area. The second operand is then analyzed to determine the type of dump requested at abnormal EOJ. If the second operand is valid, the appropriate switch is set in the Autotest communication area. The third operand is then checked to determine the limiting number of test-request actions specified by the user. The value of this parameter must be a decimal value between 0 and 9999. If this value is valid, it is stored in the Autotest communication area. It then writes the PCC card image on SYSLST and the Autotest communication area on SYSLNK. If no cancel condition has occurred in Autotest, it fetches job control to perform the EOJ step. Otherwise, the job is canceled.
Fetch Job Control EOJ

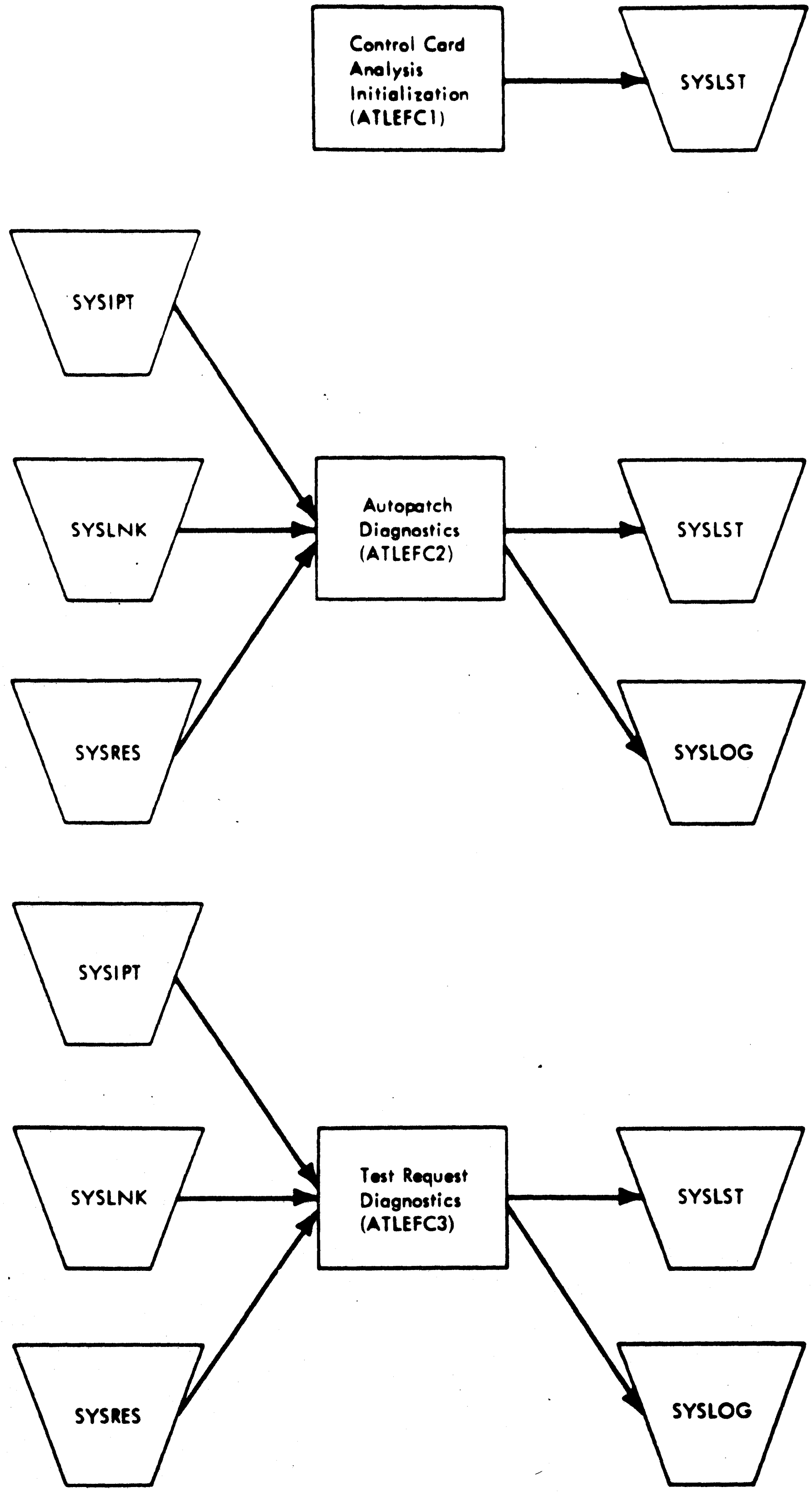


Figure 14. Control Card Analysis I/O Flow (Part 1 of 2)

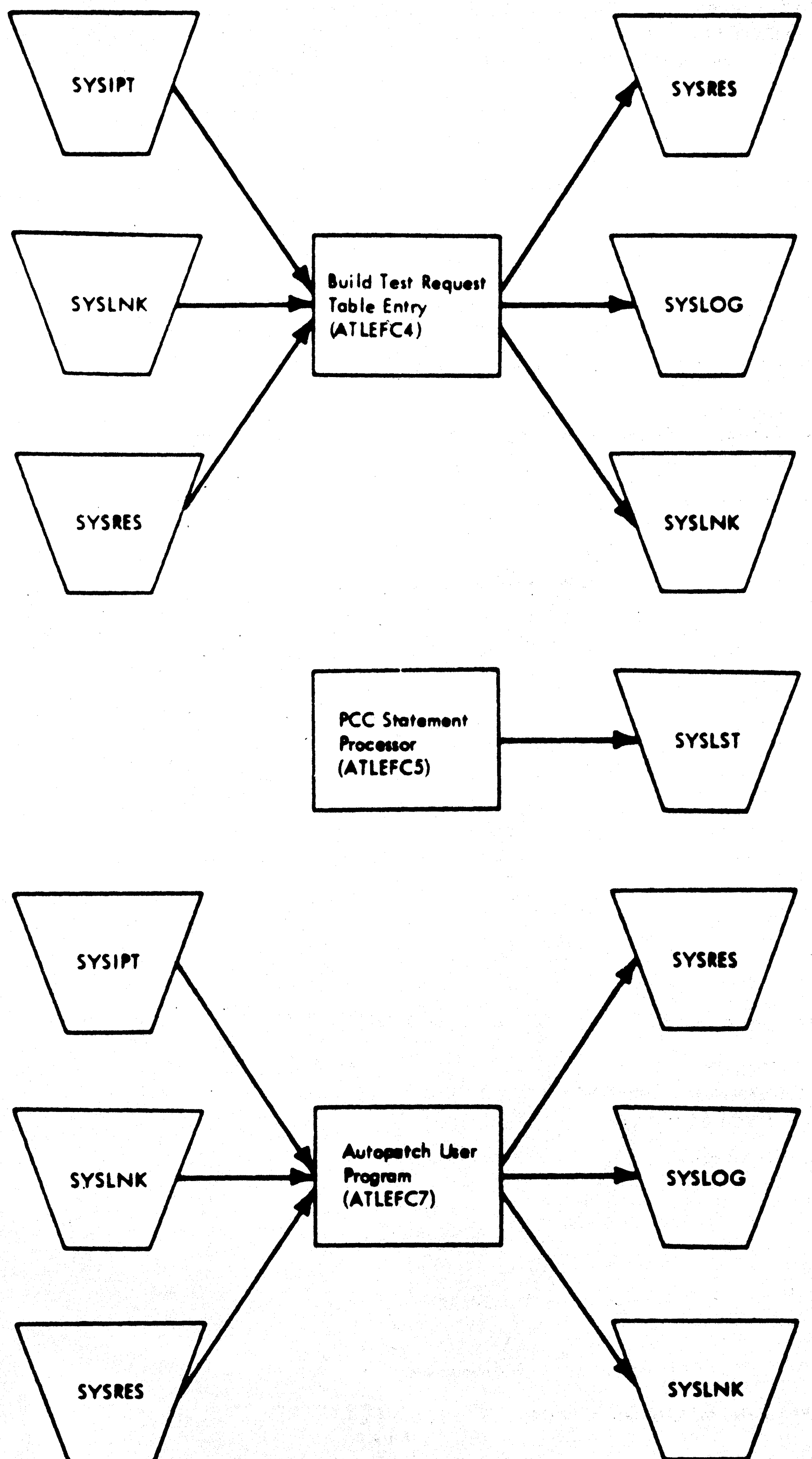


Figure 14. Control Card Analysis I/O Flow (Part 2 of 2)

	ATLEFC1	ATLEFC2	ATLEFC3	ATLEFC4	ATLEFC5	ATLEFC7
Supervisor						
Autotest Communication Area	116	116	116	116	116	116
Constants	2200	2200	2200	2200	2200	2200
Core Image Library Record Area	1728*	1728	1728	1728	1728	1728
Logic		3600	3800	3700	1400	2600

* Approximately 250 bytes of this area is used for opening files and initialization of constants and work areas to be used by the rest of the control card analysis phases. It is later overlaid when the first core image library (CIL) record is read into main storage from SYSRES. ATLEFC1 is resident during execution of the rest of the control card analysis phases.

Note: Byte counts are approximate.

Figure 15. Storage Allocation Map for Control Card Analysis

Control Section Qualification (CSQ)

If a phase consists of only one control section (assumed control section), no CSQ card is necessary. Whenever there are multiple control sections for any phase and a CSQ card is not present, the first control section (CSECT) of a phase is used by Autotest to resolve addresses. The CSQ card must precede all autopatch and test request cards for the control section involved. If used in conjunction with a PHQ card, it must not precede the PHQ card. When a CSQ card is read, the name of the control section is stored together with the relocation factor and control section number. ATLEFC2 and/or ATLEFC3 process CSQ cards.

Autopatch (ADD, EXC, CON)

ATLEFC2 processes all autopatch cards read from SYSIPT. The three functions of autopatch are: add (ADD), exchange (EXC), and replace constants (CON).

ADD: The add function allows the user to insert one or more instructions into his program by using an ADD card followed by one or more autopatch cards. ATLEFC2 analyzes the ADD card and, if the card is valid, it fetches ATLEFC7 to build the entry in the Autotest table for the instruction(s) to be added to the user program. ATLEFC7 inserts an SVC number (55-254) at the patch-point address specified on the ADD card and moves the instruction(s) to be added and the displaced instruction at the patch-point address to the Autotest table. The instructions to be added are executed before the instruction at the patch-point address. The displaced instruction in the Autotest table entry is followed by an SVC 255 indicating the end of the table entry. The table entry also contains the same SVC number that was inserted at the patch-point address in the user phase in the core image library. These SVC numbers serve as linkage between Autotest and the user program. Figure 17 shows the autopatch table entry record format. When the record output area contains 200 bytes of information, an Autotest table record is written on SYSLNK.

In addition to specifying the patch-point address, each ADD card specifies the card column of the patch indicator (*) in the actual patch card(s) that follow. The patch indicator in the patch card is followed immediately by the instruction to be added in hexadecimal format. Multiple patch cards are allowed for add and replace constant (CON) functions.

The patch-point address specified on an autopatch control card can be hexadecimal (from one to six characters) or symbolic with possible byte adjustment. If symbolic addressing is used, the symbol table output from the assembler must be supplied to Autotest. See Symbolic Capabilities for more information.

Switch Name	Hex Code	Condition	Function
CARDSWI	01	ON	ADD card being processed.
	02	ON	CON card being processed.
	04	ON	Test-request card being processed.
	80	ON	Test-request number being processed.
		OFF	SVC number being processed.
CSQSWI	40	ON	Processing phase qualification card (PHQ).
	80	ON	Processing control section qualification card (CSQ).
CSQSWI+1	01	ON	Exchange card being processed.
	02	ON	Error in multiple test-request card. First card valid.
	04	ON	Symbol error in the further qualified symbol.
	10	ON	Indicates program started to patch into Autotest table.
		OFF	Autotest table entry complete for this SVC number.
ENDSWI	01	ON	Complete instruction has not been put into core image library (end of record was found).
	02	ON	No autopatch table entry will be made. (Length of exchange instruction is equal to or less than the length of old instruction.)
ERRSWI	01	ON	Autopatch control card has been processed.
	02	ON	Indicates Autotest table is full, but more bytes to be written in Autotest table. (Record must be written.)
	80	ON	Error has occurred that will cause the job to be canceled when the PCC card is read.
E2SWI	80	ON	Write test-request control table on disk.
		OFF	Write Autotest table on disk.
ONSWITCH	01	ON	Processing the first parameter of the ON condition within the parentheses.
	02	ON	Processing the second parameter of the ON condition within the parentheses.
	04	ON	Processing the third parameter of the ON condition within the parentheses.
	80	ON	Third parameter of ON condition missing.
MTRSWI	01	ON	Multiple test-request card switch.
	40	ON	Entry in Autotest table is not complete.
PHASESWI	01	ON	Control section qualification (CSQ) was previously processed.
	20	ON	If multiple test-request ON parameters are missing, use the primary ON parameters.
	40	ON	Error in the test-request card.
	80	ON	Phase ATLEFC2 is in main storage.
PHSWI	01	ON	CSQ card switch.
	02	ON	Phase previously processed.
	04	ON	Control section qualification processed.
	08	ON	Assumed phase switch.
	20	ON	Assumed control section switch.
RLFSWI	20	ON	Minus relocation factor for further qualified CSECT.
	40	ON	Minus relocation factor for CSQ.
SWITCH	01	ON	Test-point address has been processed.
	02	ON	ON parameters being processed.
	04	ON	Start address of display parameter being processed.
	10	ON	G character of panel format processed.
	20	ON	S character of panel format processed.
	40	ON	If there is a blank column after the display end address or the secondary ON parameters (if present), a hexadecimal display and a general register panel is given.
	80	ON	Secondary ON parameters present.
SYMSWI	02	ON	Symbol adjustment is plus.
	04	ON	Symbol adjustment is minus.
	40	ON	Further qualified phase has been processed.
	80	ON	Further qualified control section assumed.
TABSWI	01	ON	Ready to create Autotest table entry.
TAPESWI	02	ON	Further qualified control section processed.
TPASWI	40	ON	Symbolic or decimal digit address being processed.
	80	ON	Test-point address has been processed correctly.

Figure 16. Common Switches used in Control Card Analysis

Code	Blank	Instructions	Displaced Instruction	AT SVC	Length	SVC No.	
A	B	C	D	E	F	G	H

A - 1 byte containing X'30'

B - 1 byte containing blanks

C - Variable length up to six bytes depending on
  1 ADD - number of bytes in instruction(s) to be added
  2 EXC (Exchange) - either four or six bytes containing the new instruction to be exchanged

D - Two, four, or six bytes containing the displaced instruction at the patch-point address. For an exchange, the displaced instruction is set to NO-OP's.

E - 2 bytes containing the special Autotest SVC of 255 indicating the end of the table entry.

F - 2 bytes containing the length of the table entry.

G - 1 byte containing the identifying SVC number of this entry.

H - 1 byte containing zeros.

Figure 17. Autopatch Table Entry Record Format

Example of ADD Patch: Assume the following instruction sequence in a program:

A	B	C	D	E	F	--

The programmer wants to add three instructions (L, M, and N) to his program after instruction D. Instruction E is displaced to allow the insertion of a supervisor call to provide linkage to the Autotest table. The new instructions and the displaced instruction are stored in the Autotest table and another supervisor call is added to provide return linkage to the program. The instructions in the program being tested are now executed in the following order:

A	B	C	D	L	M	N	E	F	--

ATLEFC7 writes the autopatch card image on SYSLST and continues to read and process patch cards until the ADD function is complete. After all patch cards for an ADD function have been processed and written on SYSLST, ATLEFC7 reads another card. If it is an Autotest control card, it fetches ATLEFC2 to analyze the card. Figure 18 shows an example of an ADD patch.

EXCHANGE: The exchange (EXC) function allows the user to replace an instruction with another instruction. If the length of the new instruction to be exchanged is equal to the length of the old instruction in the user phase in the core image library, ATLEFC7 performs the patch directly in the core image library.

If the length of the new instruction to be exchanged is less than the length of the old instruction, the new instruction is patched directly in the core image library and the unused bytes of the old instruction are set to NO-OP's. No Autotest table entry is generated in these two cases.

If the length of the new instruction is greater than the length of the old instruction, an SVC number (55-254) is inserted at the patch-point address for linkage purposes. In addition, an Autotest table entry is built containing the new instruction to be exchanged, the displaced instruction at the patch-point address, an SVC 255 (indicating the end of the table entry), the length of the table entry, and the same SVC number inserted at the patch-point address. See Figure 17 for the autopatch table entry record format. After the EXC table entry is built, ATLEFC7 writes the autopatch card image on SYSLST and fetches ATLEFC2 to read another card.

REPLACE CONSTANTS: This function allows the user to replace any number of contiguous bytes of main storage by the use of CON cards. This function can be utilized for changing constants or parts of constants. If the replace-constants function is used to replace instructions, Autotest executes the replace as if the instructions were constants.

This function does not require an entry in the Autotest table. ATLEFC7 inserts the constant patch (if valid) directly into the user phase in the core image library. It writes the autopatch card image for the replace-constant function on SYSLST and reads another card. If it is an Autotest control card, it fetches ATLEFC2 to analyze the card.

Test Request (TR)

Test request control cards allow the user to display selected portions of main storage, floating-point registers and general registers during execution of the user program. A test request always specifies an instruction address called the test-point address at which the action is to occur. The test-request action is taken immediately before the instruction is executed. Multiple test requests at the same test-point address can be given by means of continuation cards. Only TR (test-request) control cards can have continuation cards.

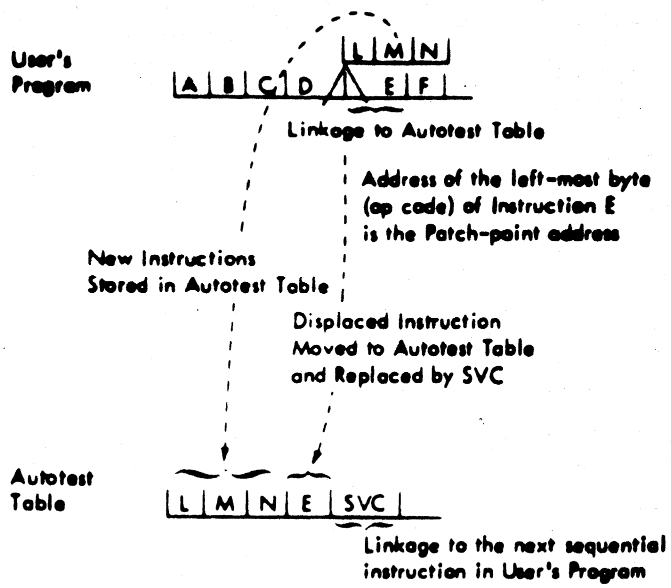


Figure 18. Example of ADD Patch

Continuation cards are recognized by a continuation character punch in column 72. This indicates the data for the test request is continued in column 16 of the following card. Card columns 1-15 of the continuation card must be blank. A parameter cannot be divided between two cards. If there is not sufficient card space to complete the parameter, a continuation card must be used. Autotest determines if the information punched in the test-request and continuation cards is valid. If the information is not valid, diagnostic messages are written on SYSLST.

The addressing possibilities for a test request are:

- Hexadecimal--Valid for test-point address and display limits. Can be one to six hexadecimal characters.

- Decimal--Valid for display limits; invalid for the test-point address. Can be one to six decimal digits.

- Symbolic--Valid for the test-point address and the display limits. Any symbol used must be present in the symbol table output from the assembler. Any symbolic address reference can be specified with byte displacement. This is done using a defined symbol followed by a signed decimal number of from one to four digits.

The three types of test requests available are as follows:

- Display
- Panel
- Display and Panel

Display (DSP)

The display test-request causes an area of main storage designated by the user to be temporarily written on the Autotest work area on SYSLNK during execution of the user program. This display data is later formatted and written on SYSLST by ATLEFF1, the test request output phase. Autotest provides the following variety of formats for the printout depending on the parameter specified:

Parameter	Format
Omitted	Hexadecimal representation
C	Character representation
B	Hexadecimal and character
E	Floating-point single precision
D	Floating-point double precision
F	Fixed-point fullword
H	Fixed-point halfword
M	Hexadecimal with mnemonics

When a character-only (C) display is given, data in the area to be displayed is converted to printable characters, if possible. If the conversion of the next line of data to be printed results in the same printable characters, in the same order and arrangement, the DITTO message is printed. This is done even though the actual bit configurations in the display area may not be the same before conversion. Multiple parameters cannot be specified for a display.

Panel (PNL)

If no parameters are specified, panel output consists of general registers (0-15). Other combinations of general registers, floating-point registers, and storage positions 24-127 can be specified. The parameters are:

Parameter	Meaning
G	General registers only
S	Storage positions 24-127
F	Floating-point registers

Multiple parameters for a panel can be specified.

Display and Panel (DPL)

The display and panel test requests can be combined if desired. The display parameter precedes the panel parameter(s) in the test-request control card.

ON Condition for Test Request

If the test-point address is reached more than once during execution of the program being tested (such as in a programmed loop), the user can regulate execution of a display and/or panel by specifying ON parameters.

Display and panel are independent functions and can be used separately, together, or with ON. The number of times they can be requested is limited by the number of branch-out linkages (maximum 200) to the Autotest table, the amount of main storage available for the Autotest table, and the space available on the Autotest work file on SYSLNK.

A request is executed when the test-point address is reached during user program execution (provided the ON conditions are satisfied). The ON condition is specified by three integers which serve as comparison values for a counter set up by Autotest. These integers can be respectively interpreted as BEGIN, END, and EVERY.

For example, the ON parameters 2, 8, 3 specified on a test-request control card are interpreted as follows:

When the test-point address has been reached twice, the test-request action is performed for the first time (BEGIN). From then on, the action is performed every third time (EVERY) until the eighth time (END), after which no more actions are taken.

The BEGIN and EVERY parameters are stored in the Autotest table entry for the test request along with the ON number which is calculated according to the following formula:

$$\text{ON number (number of actions)} = \frac{\text{END - BEGIN}}{\text{EVERY}} + 1$$

ON parameters can be specified for all test requests at a test-point address within the program and/or for a specific test-request function.

The secondary ON parameters for a specific test-request function override the primary ON parameters specified for the test-point address.

Format of Test Request (TR) Control Cards

The general format of a test-request control card requires a blank column between the identifier (./) and card type (TR), and another blank preceding the first parameter as follows:

./ TR tpa,(nl,n2,n3),fnc,Lower,Upper,(nl,n2,n3),d,p

Entry	Columns	Explanation
./	1-2	Identifies a control card.
TR	4-5	Identifies card type.
tpa	7-(as required by length of field)	Test-point address, can be in form X'x..x' (one to six hexadecimal characters) or can be a symbolic address (one to eight valid symbolic characters). Note: Address adjustment of labels is a feature of symbolic addressing. However, the adjustment factor must be a signed decimal number of from one to four digits.
(nl,n2,n3)	Immediately following test-point address field.	Test-request primary ON condition parameters. nl equals BEGIN, n2 equals END and n3 equals EVERY. Each n-number may be up to four decimal numbers in length. These test-request parameters apply to all actions taken at this test-point address.
fnc	Immediately following ON parameters field.	Test-request function as follows: DSP for display, PNL for panel, and DPL for display and panel.
Lower, Upper	Immediately following test-request function field.	If display, or display with panel, lower and upper limits must be shown. Values can be hexadecimal, decimal, or symbolic.
(nl,n2,n3)	Immediately following.	Secondary ON parameters specified for this test-point address apply only to this particular test-request function. The test-request primary ON condition associated with the test-point address is disregarded when this secondary ON condition is specified.

Entry	Columns	Explanation
d	Immediately following display limits or ON parameters if used.	Display parameter; can be C, B, E, D, F, H, or M, or can be omitted. See the Display (DSP) section.
p	Immediately following.	Panel parameter; can be G, S, or F. It can also be combinations of these, or can be omitted. See the Panel (PNL) section.

Multiple tests requests at a single test-point address are permitted. The ON condition parameters are inserted in the control card as follows:

- The test-request primary ON condition applies to all test requests at a certain address in the program.
- The secondary ON condition parameters must be inserted with the specific test-request function. If the test-request primary ON parameters are to be disregarded, the secondary ON parameters override the primary ON parameters specified for the test-point address.

When ATLEFC2 reads a TR (test-request) control card, it fetches ATLEFC3 to analyze and process the card. ATLEFC3 determines if the test-point address has a valid hexadecimal or symbolic (with or without byte adjustment) representation. If the test-point address is valid, a check for a continuation card is made. Although only one test request is permitted per card, more than one test request (DSP, DPL, PNL) can be given at the same test-point address through the use of continuation cards. The total number of test requests cannot exceed 200.

The presence of the ON parameters to regulate test-request execution is then checked. If present, the three ON parameters (BEGIN, END, and EVERY) are converted for execution. Both the BEGIN and EVERY parameters remain the same. A new parameter (ON number) to indicate the number of test-request actions to be taken is calculated according to the following formula:

$$\text{ON number (number of actions)} = \frac{\text{END} - \text{BEGIN}}{\text{EVERY}} + 1$$

If a display (DSP) function is requested, the start and end addresses are analyzed. These addresses can be either symbolic, hexadecimal, or decimal. A check for ON parameters for this particular test request is made. If present, the ON parameters for this particular request override the ON parameters associated with the test-point address.

After the ON parameters for this particular test request are analyzed, the display parameter is analyzed. Its format can be hexadecimal, character, hexadecimal and character, floating-point single precision, floating-point double precision, fixed-point fullword, fixed-point halfword, and hexadecimal with mnemonics.

If a panel (PNL) function is requested, a check for the ON parameters for this particular test request is made. If present, these ON parameters override the ON parameters associated with the test-point address. The ON parameters associated with the test-point address are ignored for this test request function only. The PNL parameter is then analyzed. Any combination of general registers, floating-point registers, and storage positions 24-127 can be specified.

If a display and panel (DPL) function is requested, the display limits are analyzed and the presence of the ON condition is checked. After checking for the ON condition, this routine analyzes the display parameter and then the panel parameter.

If any error occurs while analyzing the test-request card, the proper error code is moved to a table, the error is processed, and analyzing of the card continues.

After the test-request card is analyzed, the card image is written on SYSLST and another card is read. After all the cards for a particular test-point address are read and analyzed, ATLEFC4 is fetched to build the test-request table entry.

ATLEFC4 builds the Autotest table entry for a DSP (Display), PNL (Panel), or DPL (Display and Panel) request. It moves the instruction at the test-point address in the user phase to a save area and inserts an SVC identification number at the test-point address. This SVC identification number is also inserted in the Autotest table entry for use by the Autotest control program during user program execution. The routine then checks for a continuation card (punch in card column 72) and, if present, continues to build the table entry until no more continuation cards for a particular test-point address are read. It then completes the Autotest table entry by moving the displaced instruction in the user phase and the Autotest SVC 255 to the end of the table entry. Figure 19 shows the record format for the test-request table entry.

Code	T R Number	On	Display Limits		Displaced Instruction	A T SVC	Length	SVC No	
A	B	C	D	E	F	G	H	I	J

A - 1 byte containing the following information

Bits 0-2	Length of each test-request entry in half-words
Bit 3	Last test request for this number (0-NO, 1-YES)
Bit 4	Floating-point dump required (0-NO, 1-YES)
Bit 5	ON condition (0-NO, 1-YES)
Bit 6	Panel (0-NO, 1-YES)
Bit 7	Display (0-NO, 1-YES)

B - 1 byte containing the test-request number

C - 6 bytes containing the ON parameters as follows:

Bytes 0-1	BEGIN number - (first time the action is to occur)
Bytes 2-3	ON number - (number of times the action is to occur)
Bytes 4-5	EVERY number - (when the action is to occur)

D - 6 bytes containing display limits as follows:

Bytes 0-2	Start address
Bytes 3-5	End address

E - Additional number of bytes required for number of multiple test requests up to 255.

F - Two, four, or six bytes containing the displaced instruction at the test-point address.

G - 2 bytes containing the special Autotest SVC of 255 indicating the end of the table entry.

H - 2 bytes containing the length of the table entry for this particular SVC number.

I - 1 byte identifying the SVC number for this entry.

J - 1 byte containing zeros.

Figure 19. Test Request Table Entry Record Format

ATLEFC4 also builds a test-request control record containing the necessary information for a display, panel, or display and panel to be used by the test-request output phase (ATLEFF1) in formatting the test-request output generated during execution of the user program. Figure 20 shows the format of the test-request control record.

After the test-request table entry and the test-request control record for a particular test-request (TR) card are built, this routine fetches ATLEFC3 to read another Autotest control card.

Program Control (PCC) Card

This card follows the last test-request and/or autopatch card, and indicates that all autopatch and test-request cards for that test have been processed. This card is required for all test jobs. The card must be punched as follows:

./ PCC n,a,x

Entry	Columns	Explanation
./	1-2	Identifies control card.
PCC	4-6	Identifies card type.
n	8-(as required by length of field).	This code indicates the type of normal end-of-job dump desired. The supervisor is not included unless requested. Values of n are:

Core	Dump
(omitted)	Hexadecimal characters.
N	Suppress dump.
X	Includes storage printout of system supervisor with problem program.
A	Hexadecimal and alphameric characters. Each byte is printed in its EBCDIC (Extended Binary Coded Decimal) representation, if printable.
M	Hexadecimal characters and mnemonic representation. Every two bytes is considered an instruction, translated into a mnemonic operation code, and printed.
S	Hexadecimal characters and symbols. If the object program includes a symbol table, these symbols are related to the contents of main storage.

Each physical record is 200 bytes and has the following format:

Code	Number of Records	Variable Number of Logical Records
A	B	C

A. 1 byte containing a T

B. 1 byte - Bits 0-6 - contains number of logical records in physical record
Bit 7 - last record indicator 0 - not last record 1 = last record

C. Variable - contains logical records each 2-12 bytes in length. The format of the maximum length logical record is:

Byte(s)	Bit(s)	Contents
0	0	Display (0=NO, 1=YES)
	1	ON condition (0=NO, 1=YES)
	2	General register panel (0=NO, 1=YES)
	3	Storage positions (24-127) panel (0=NO, 1=YES)
	4	Floating-point register panel (0=NO, 1=YES)
	5-7	Display Parameters
		A. 000 - hexadecimal
		B. 001 - character
		C. 010 - hexadecimal and character
		D. 011 - floating-point single precision
		E. 100 - floating-point double precision
		F. 101 - fixed-point half word
		G. 110 - fixed-point full word
		H. 111 - hexadecimal and mnemonic
1	All	Test-request number
2-7	All	Display limits
8-11	All	ON condition (BEGIN and EVERY numbers)

The number of bytes in the logical record required for a test-request action are:

A. Display (DSP) - 8 bytes (0-7)
B. Panel (PNL) - 2 bytes (0-1)
C. Display and Panel (DPL) - 8 bytes (0-7)
D. Display with ON - 12 bytes (0-11)
E. Panel with ON - 6 bytes (0-1, 8-11)
F. Display and Panel with ON - 12 bytes (0-11)

Figure 20. Test-Request Control Record Format

Entry	Columns	Explanation
		Note: Any combination of X, A, M, or S can be used. No commas are permitted between codes. Example: MS specifies hexadecimal, mnemonic, and symbolic. The supervisor storage printout is always given in hexadecimal format, if requested. This occurs even though the problem program may have been requested in some other format.
a	Immediately following the comma.	This code specifies the type of abnormal end-of-job dump desired. The abnormal end-of-job dump prints all of main storage. The codes are the same as for the n-dump term, except that the N (suppress) and X (include supervisor) entries are invalid. Example: SA specifies hexadecimal, symbolic and alphameric. Note: A test-request action refers to every time the test-point address is encountered.
x	Immediately following the comma.	This operand specifies the loop protection option. A value equal to the number of test-request actions desired can be punched. This operand causes Autotest to terminate the program if the maximum number of test-request actions is exceeded. Any value up to 9999 decimal can be entered. If this value is omitted, a value of 50 is assumed.

A PCC card with no operand is valid and the following values are assumed:

for n, Hexadecimal dump
for a, Hexadecimal dump
for x, 50 maximum actions

If a comment is required in a PCC card that has no operand entries, the comment must be preceded by at least two blanks. If the PCC card has operand entries, at least one blank is required between the last operand entry and the comment.

ATLEFC5 analyzes the parameters on the PCC card and stores the option requested, if valid, in the appropriate field in the Autotest communication area. In the initialization routine, the assumed values for the parameters on the PCC card are put in the Autotest communication area. These assumed values are:

1. ACTLIM--set to 50
2. DUMPSW--set to indicate normal end-of-job (EOJ) dump required
3. DMPFORMT--set for hexadecimal normal and abnormal EOJ dump.

The first two parameters are then analyzed for types of dump requested at normal or abnormal EOJ, respectively. For a normal EOJ dump, the valid entries are:

1. N--Suppress dump on normal EOJ. Or, any combination of the following.
2. A--Alphameric or character dump
3. S--Symbolic dump
4. M--Mnemonic dump
5. X--Dump supervisor (in hexadecimal format).

For the second parameter, the valid entries for an abnormal EOJ dump are:

1. A--Alphameric or character dump
2. S--Symbolic dump
3. M--Mnemonic dump

When a valid entry is found in the first or second parameter, the appropriate switch is turned on in the Autotest communication area. When analyzing a particular parameter, a switch (NA, NS, NM, or NX) is turned on as a particular character (A, S, M, or X) is found. For example, if the parameter for normal EOJ dump is SSSS, scanning for that parameter stops after the first S is detected and DMPFORMT in the Autotest communication area is set for a symbolic dump on normal EOJ.

When an invalid entry is found in the first or second parameter, an error message is written on SYSLST and scanning of the card continues. When a comma cannot be found, an error message is written on SYSLST, scanning stops, and control is given to job control to perform the EOJ step.

The action count parameter is the third parameter analyzed. If it is numeric and not more than four decimal places in length, it is valid. If neither of these conditions is met, an error message is written on SYSLST, and the assumed value of 50 is stored in ACTLIM in the Autotest communication area.

After job control reads an EXEC card with a blank operand field, it moves the directory containing all the user and Autotest phases previously linkage-edited to the system phase directory. Job control then issues a fetch for the first phase name in the phase directory. The first phase to be fetched from the core image library by job control is ATLEFE1, the Autotest control program initialization phase.

ATLEFE1 first ensures that all of main storage has been allocated to Autotest. If the user attempts to run in foreground mode or background mode with any foreground job in operation, Autotest cancels the job. It performs the necessary initialization for user-program execution by loading the Autotest control program into main storage and reading the Autotest communication area record and the Autotest table record(s) from SYSLNK into main storage. ATLEFE1 turns on the switch in the system communication region to indicate Autotest is in process and fetches $$BATST1 to set up the interface between the Autotest control program and the supervisor.

$$BATST1 clears ATLEFE1 from main storage and gets the name of the first user phase to be loaded into main storage. The name of the first user phase, provided by the Autotest linkage editor, is found in the Autotest communication area. This phase is loaded into main storage for execution. The address of the Autotest control program in main storage is moved to the address portion of the SVC new PSW; thus ensuring that the Autotest control program will get control whenever an SVC interrupt occurs during execution of the user program. $$BATST1 also moves the address of the first user phase in the problem program area to the user PSW in the supervisor save area. This gives control to the user phase after execution of $$BATST1 is complete.

When an SVC interrupt occurs during execution of the user program, the Autotest control program gets control and determines the source of the interrupt. If the SVC interrupt comes from the supervisor or the Autotest control program, the SVC new PSW is moved into bytes 0-7 of main storage, the contents of register 15 (used by the supervisor as a base register in the SVC interrupt routine) is restored, and a LPSW is issued to give control to the supervisor to handle the interrupt.

If the SVC interrupt comes from the user, the contents of the general registers (0-15) and the SVC old PSW are saved to ensure that the Autotest control program will not destroy their contents before control is returned to the user program. A check is made to determine if the SVC interrupt from the user program is a valid Autotest SVC. If it is not, a test for an SVC fetch or load from the problem program area is made.

If the SVC was a fetch from the problem program area, a 117-byte phase-fetch record containing a code of F and the phase name (8 bytes) is written on the test-request output file on SYSLNK. If the SVC was a load from the problem program area, a 117-byte phase-load record containing a code of L, the phase name, and the load address is written on the test-request output file on SYSLNK. The phase-fetch or load record format is shown in Figure 21. If the SVC was not a fetch or load from the problem program area, it is invalid. The address of the invalid SVC is saved and control is returned to the supervisor.

If the SVC interrupt from the user program is a valid SVC generated by Autotest, the Autotest control program compares the SVC number to the corresponding entries in the Autotest table. Each time an Autotest-generated SVC is matched to its table entry, the action count limit specified by the user in the PCC card is decreased by one. If the action count limit is exceeded, an error code is moved to the Autotest communication area, general registers (0-15) are restored, and control is returned to the supervisor to terminate user execution. If the action count limit is not exceeded, the Autotest service(s) requested by the user at this point in his program (either a test request or autopatch) is performed and the next subentry in the matching table entry is analyzed. Each Autotest table entry is terminated by an SVC code of 255. When an SVC of 255 is encountered, the contents of the user registers are restored and control is returned to the user. The return address for the user program is contained in the SVC old PSW previously saved. This SVC old PSW is moved to bytes 0-7 of main storage. Thus, when the LPSW is issued, control is returned to the user program to continue execution.

If the table entry specifies that an autopatch service is to be performed by the Autotest control program, Autotest executes

Chart 12. User Program Execution

User Program Execution

START

AUTOTEST CONTROL PROGRAM INITIALIZATION (ATLEFE1) Charts QA-QB

After job control reads an EXEC card with a blank operand, this routine gets control and loads the Autotest control program into main storage. It modifies the address constants within the Autotest control program and reads the Autotest communication record from SYSLNK into high storage. It then reads the Autotest table record from SYSLNK into main storage immediately preceding the Autotest control program.

FIRST

SET UP INTERFACE BETWEEN AUTOTEST CONTROL PROGRAM AND SUPERVISOR (SSBATST1) Chart RA

This B-transient program clears ATLEFE1 from main storage and loads the first user phase into main storage. It builds a fetch record for the first user phase and writes the record on SYSLNK. It then modifies the SVC new PSW by storing the address of the Autotest control program in the address portion of the SVC new PSW.

SVCINTPT

AUTOTEST CONTROL PROGRAM EXECUTION (ATLEFE2) Charts SA-SF

The Autotest control program performs the following functions:

1. Monitors execution of the user program via SVC interrupts.
2. Writes the display and/or panel information for a test request on SYSLNK.
3. Executes ADD or EXC patch.
4. Writes phase fetch records containing the names of the user phases fetched into the problem program area on SYSLNK.
5. Writes phase load records containing the phase name and load address of all phases loaded into the problem program area on SYSLNK.

BATST320

DUMP USER PROGRAM (SSBATST3) Charts TA-TC

SSBATST3 is fetched into main storage by the B-transient phase, SSBEOJ, after the user program comes to either normal or abnormal EOJ. It dumps main storage up to 32K (depending on the size of the machine) except for the supervisor, on SYSLNK for processing by the EOJ dump phase. It also writes a small 155-byte EOJ dump record on SYSLNK. It returns control to the supervisor to handle all SVC interrupts and loads the test-request output phase, ATLEFF1, into main storage for execution.

Post-Processing Phases

either the EXC (Exchange) patch (where the length of the new instruction is greater than the length of the old instruction) or the ADD patch. See Figure 17 for the format of an autopatch table entry.

If the table entry specifies that a test-request service is to be performed, the area to be displayed as well as the test-request code and number is written on the test-request output file on SYSLNK. If a panel is requested, two records are written on SYSLNK. The first record contains bytes 0-112 of main storage and the second record contains bytes 113-127 of main storage, the contents of general registers (0-15), and the contents of the floating-point registers (if present). If a display is requested, records containing 113 bytes of display information are written on SYSLNK for the area specified by the display limits. See Figure 21 for the test-request output record format. If no test-request action is required at a particular time during execution of the user program as determined by the ON condition, the next subentry within the table entry is analyzed. It may be the displaced instruction from the user program or it may be another test-request action. If no more test-request actions are to be performed, the next subentry is assumed to be the instruction displaced from the user program by the Autotest-generated SVC followed by the Autotest SVC of 255 signifying the end of the Autotest table entry.

When the user program comes to either abnormal or normal end-of-job, the B-transient, $$BEOJ, as part of the EOJ step, fetches $$BATST3 into main storage to dump the user program onto SYSLNK and to save the cancel code giving the reason for user EOJ.

$$BATST3 dumps main storage from the beginning of the problem program label area up to end of main storage for a 16K machine or up to 32K for a machine size of 32K or greater. For a machine size greater than 32K, the EOJ dump phase, ATLEFH3, dumps the rest of main storage. $$BATST3 then writes a 155-byte record on SYSLNK containing the general registers, floating-point registers (if present), user PSW, cancel code, length of user label area, and 48-byte user communication region. See Figure 21 for the 155-byte EOJ dump record format. It resets the Autotest switch in the system communication region and loads the test-request output phase, ATLEFF1, into main storage. Finally, it moves the address of ATLEFF1 in the problem program area to the user PSW in the supervisor save area. This gives control to ATLEFF1 after execution of $$BATST3 is complete.

Figure 22 shows the I/O flow for user program execution. Figure 23 shows the storage allocation maps for user program execution.

1. Phase fetch record - Each physical record is 117 bytes having the following format:

Code		Phase Name	
A	B	C	D

A - 1 byte containing an F
B - 3 bytes unassigned
C - 8 bytes containing name of phase
D - 105 bytes unassigned

2. Phase load record - Each physical record is 117 bytes having the following format:

Code		Phase Name		Phase Address	
A	B	C	D	E	F

A - 1 byte containing an L
B - 3 bytes unassigned
C - 8 bytes containing name of phase
D - 1 byte unassigned
E - 3 bytes containing load address of phase
F - 101 bytes unassigned

3. Test-request output record for a panel - Each panel request generates two 117-byte physical records having the following format:

Record 1

Code	Test-Request Number		Data
A	B	C	D

A - 1 byte containing an R
B - 1 byte containing the test-request number for a particular test-request action
C - 2 bytes unassigned
D - 113 bytes containing panel data for bytes 0-112 of main storage

Figure 21. Record Formats for User Program Execution (Part 1 of 2)

Record 2

Code	Test-Request Number		Data		General Registers	Floating-Point Registers	
A	B	C	D	E	F	G	H

A - 1 byte containing an R
B - 1 byte containing the test-request number for a particular test-request action
C - 2 bytes unassigned
D - 15 bytes containing panel data for bytes 113-127 of main storage
E - 1 byte unassigned
F - 64 bytes containing contents of general registers 0-15
G - 32 bytes containing contents of floating-point registers (even-odd pair 0, 2, 4, 6)
H - 1 byte unassigned

4. Test-request output record for a display - Each display request generates as many 117-byte physical records as required by the display limits.

Code	Test-Request Number		Data
A	B	C	D

A - 1 byte containing an R
B - 1 byte containing the test-request number for a particular test-request action
C - 2 bytes unassigned
D - 113 bytes of display data

5. EOJ dump record - Each physical record contains 1728 bytes of dump information to be formatted by ATLEFH3, the EOJ dump phase.

6. 155-byte EOJ dump record - Special 155-byte record written on the Autotest work area on SYSLNK having the following format:

Bytes	Contents	Bytes	Contents
0-63	General registers (0-15)	104	Cancel code
64-95	Floating-point registers	105-106	Length of user label area
96-103	User PSW	107-154	48-byte user communication area

Figure 21. Record Formats for User Program Execution (Part 2 of 2)

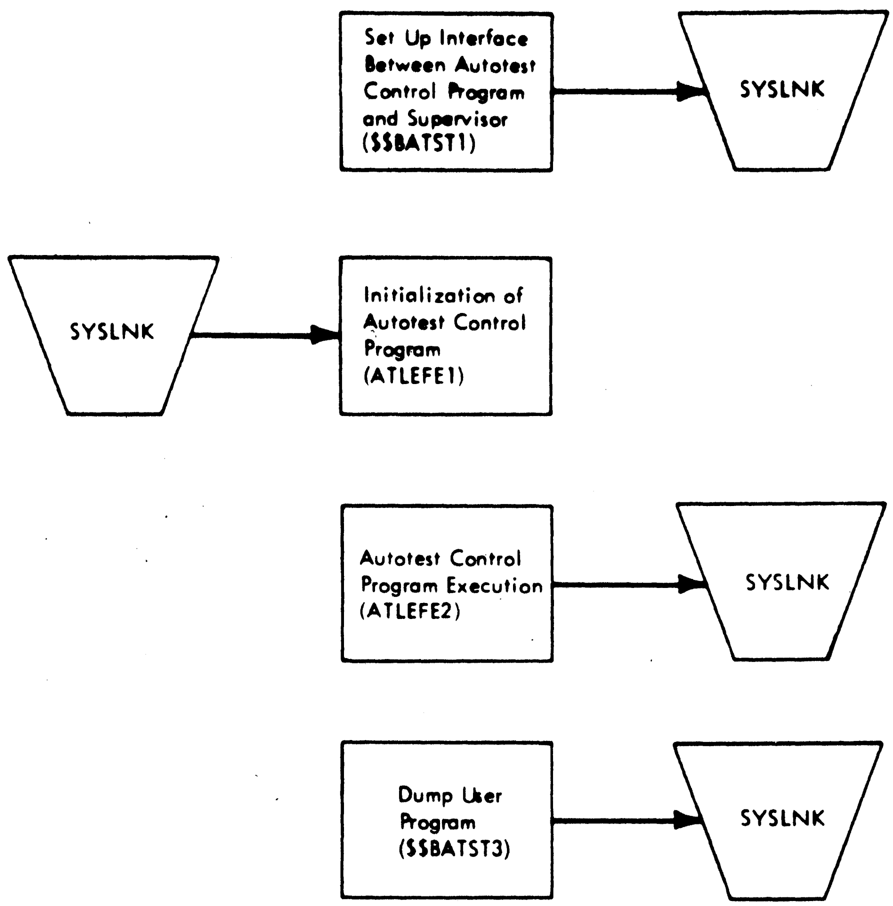


Figure 22. User Program Execution I/O Flow

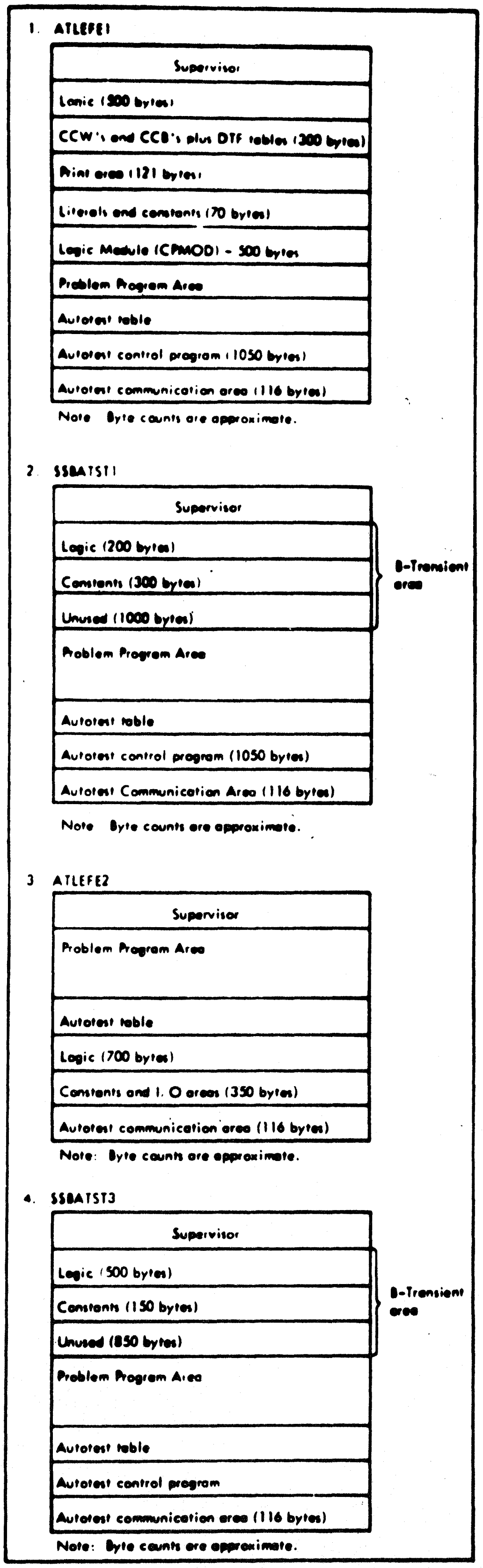


Figure 23. Storage Allocation Maps for User Program Execution

The first post-processing phase, ATLEFF1, is fetched into main storage by $SBATST3 to process any test-request output from user program execution. If no test-request output is present, the phase list and core map routine (ATLEFG1) is fetched. If test-request output is present, this phase reads the test-request control records written on the Autotest work file on SYSLNK by ATLEFC4 into main storage. Using the information in the test-request control records, it builds a test-request table in main storage containing the test-request number, type of display and/or panel requested, the display limits, and the BEGIN and EVERY ON parameters. For further information concerning the ON parameters, see Occurrence of Panel and Display.

After the test-request table is built, this phase reads the test-request output records on SYSLNK. It bypasses the phase fetch and load records which are interspersed with the test-request output records. When it finds a test-request output record, it converts the test-request identification number and the BEGIN number for the ON condition to decimal and writes this information along with the header line on SYSLST.

It then tests for a display only, panel only, or display and panel. If a panel is requested, the test-request output record containing the panel data for main storage positions 0 through 112 is stored and the second panel record is read. The second panel record contains panel data for main storage positions 113-127, contents of general registers and contents of the floating-point registers. Using the information in the test-request table, it writes the type of panel requested on SYSLST. Figure 24 shows an example of a test-request printout.

If the test request was for a display and panel, the third record containing 113 bytes of display information is read and processing of the display information in the format requested is begun. The test-request output records are then read until the end of the display, as determined by the display limits, is reached. If the test request was for a display only, the test-request record read would contain display data in place of panel data. When the EOF marker for the test-request output file is read, this phase fetches ATLEFG1 for execution.

ATLEFG1 first tests for a disaster-continue situation. If a disaster-continue situation has occurred, this routine writes a list of the user phases fetched or loaded into the problem program area on SYSLST and fetches ATLEFH3. If a disaster-continue situation has not occurred, this routine reads the control dictionary records on SYSLNK and uses the information in the records to build a phase list table containing 16-byte entries consisting of the phase name, phase number, length of phase, and start address of phase.

It then reads the test-request output records on SYSLNK and uses the information from the phase fetch or load records to build a phase fetch/load table. This table contains 8-byte entries each consisting of the phase number, phase origin, and the phase end address. Using these tables just built, it writes the phase list consisting of the record type (either fetch or load), phase name, and start and end addresses of phase on SYSLST.

After the phase list is written on SYSLST, the start and end address-sequence tables that overlay the phase list table are built using the phase fetch/load table. The start and end address-sequence tables contain the sorted start and end addresses of those phases not overlaid by a subsequent phase. ATLEFG1 uses these tables to build the 402-byte core map records to be written on SYSLNK. Each core map record contains up to 50 8-byte logical records consisting of the phase number, start or restart (if part of the phase is overlaid by another phase) address and phase displacement. These records are written on the Autotest work file on SYSLNK and are used by ATLEFH2 to organize the symbols in main storage at user EOJ. The core map record format is shown in Figure 25. If a symbolic dump is requested at normal or abnormal EOJ, this routine fetches ATLEFH2 for execution. If a symbolic dump has not been requested, this routine fetches the EOJ dump phase, ATLEFH3, for execution.

When a symbolic dump is requested, ATLEFH2 processes the SYM records written on the Autotest work file on SYSLNK by the Autotest linkage editor using the information in the core map records on SYSLNK to determine the valid symbols in main storage at the termination of the user program. Figure 26 is a schematic of main storage at the time of user EOJ. Only those symbols not

Chart 13. Post Processing (Part 1 of 3)

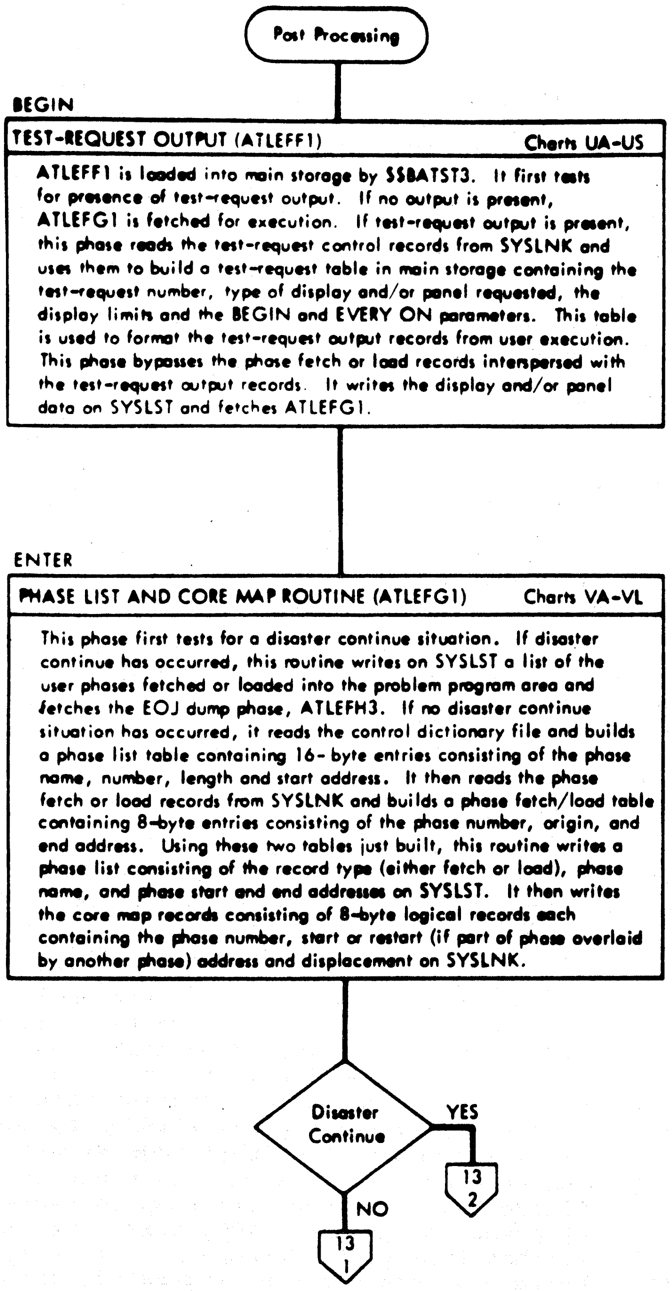


Chart 13. Post Processing (Part 2 of 3)

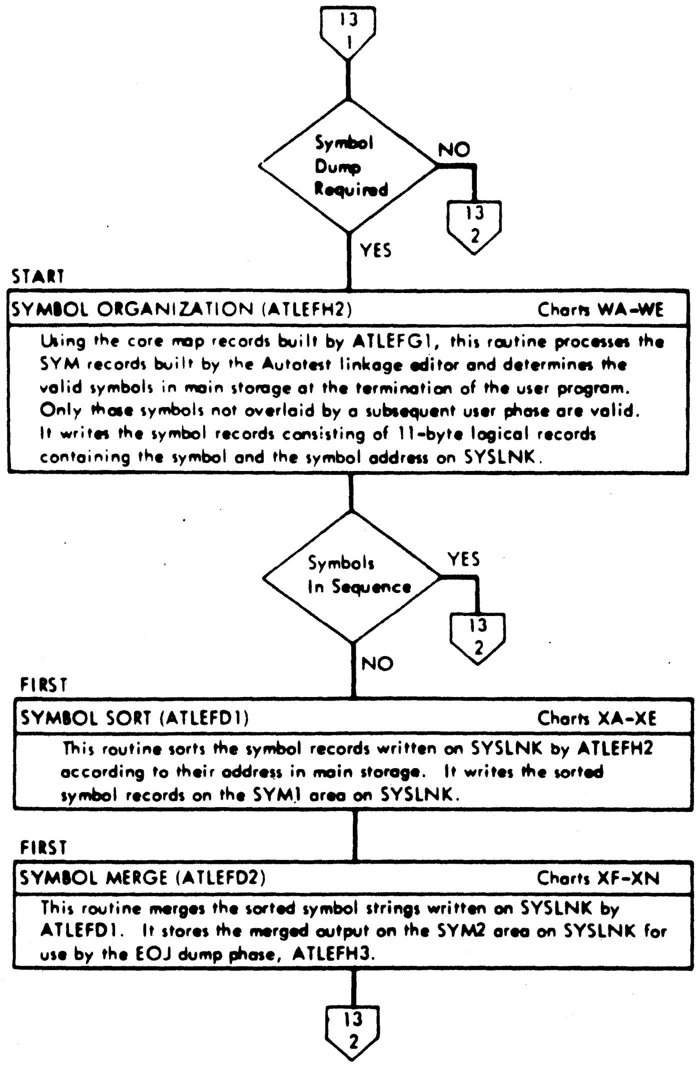


Chart 13. Post Processing (Part 3 of 3)

13
2

START

EOJ DUMP (ATLEFH3) — Charts YA-YN

This phase first tests for disaster continue. If a disaster continue situation has occurred, ATLEGO1 is fetched to terminate the job-step and initiate the next. If a disaster continue situation has not occurred, it tests for normal EOJ dump required. If a dump was not required at normal EOJ, it fetches ATLEGO1. Otherwise, it formats the EOJ dump records and the 155-byte EOJ record on SYSLNK and writes the following information on SYSLST

1. Reason for EOJ - either normal EOJ or as determined from the cancel code passed to ATLEFH3 by SSBATST3. See Autotest communication area for further information
2. Last user PSW - in hexadecimal format
3. General registers 0-15 - in hexadecimal format
4. Floating-point registers (even-odd pair 0, 2, 4, 6) - if present, in hexadecimal format
5. 48-byte user communication region - in hexadecimal format
6. Supervisor - if requested at normal EOJ, in hexadecimal format starting at storage location 320 and ending at the start of the logical transient area.
7. Length of label area - in hexadecimal format
8. User main storage - from the end of supervisor to the end of main storage in hexadecimal format only or in any combination of hexadecimal and character, mnemonic, and/or symbolic.

IJVTI120

AUTOTEST END ROUTINE (ATLEGO1) — Charts YP-YQ

This routine reads a card from SYSIPT and checks for a /&, job control (b) or ATEOF (Autotest end-of-file) card. If either the /& or ATEOF card is read, it fetches job control to perform the EOJ step. If a job control card was read and no ATEOF card was found, an error message indicating the ATEOF card was out of order is written on SYSLST and a check for an EXEC card is made. If the EXEC card is found and has a blank operand, an error message indicating more than one EXEC card is invalid in Autotest is written on SYSLST and the job is canceled. If the EXEC card does not have a blank operand, the job is canceled.

Fetch Job Control

```
TEST REQUEST ID  002  ON  00001

        003118    000000   0C

TEST REQUEST ID  001  ON  00001

REG0      00000013   00002808   5000299A   0000399A   00000008   0000FC31   A0002126   00000002
REG8      900029D2   00000000   60002A2A   00003C94   00002C18   00003C18   A0002A50   00003EF8

        003153    000000   0C000000   0C000000   0C000000      003162  0C
```

```
TEST REQUEST ID  002  ON  00004

        003118    000000   0C

TEST REQUEST ID  001  ON  00036

REG0      00000013   00002808   5000299A   0000399A   00000008   000033E1   00000015   00000032
REG8      A0002F4C   00000000   40002F70   00003C94   00002C18   00003C18   A0002A50   00003EF8

        003153    000600   0C000031   2C000006   5C000637      003162  7C
```

```
TEST REQUEST ID  003  ON  00001

FLOATING POINT REGISTERS    0                   2                   4                   6
                            0000000000000000    0000000000000000    0000000000000000    0000000000000000

              .00000000 E 00              .00000000 E 00              .00000000 E 00              .00000000 E 00
              .0000000000000000 E 00      .0000000000000000 E 00      .0000000000000000 E 00      .0000000000000000 E 00

REG0     00007B90   90002188   00007ACA   00002000   00007F8C   50002002   00003002   00004002
REG8     000000C5   00000000   40002002   00002000   00002C18   00003C18   00000188   00000088

40-CSW  KEY-10  ADDR-007F68  STATUS-0000110001000000  COUNT-0400      48-CAW  KEY-00  ADDR-000008
PSW CONTENTS          EXTERNAL INTERRUPT  SUPERVISOR CALL     PROGRAM CHECK       MACHINE CHECK       INPUT/OUTPUT
FIELD          FORMAT-OLD 18   -NEW 58   -OLD 20   -NEW 60   -OLD 28   -NEW 68   -OLD 30   -NEW 70   -OLD 38   -NEW 78
SYSTEM MASK       BIT 11111111  00000000  11111111  00000000  11111111  00000000  01011011  00000000  11111111  00000000
PROTECTION KEY    HEX 0         0         1         0         1         0         5         0         0         0
AMWP              BIT 0101      0100      0101      0100      0101      0100      1011      0000      0100      0100
INTERUPT CODE     HEX 0000      0000      00F9      0000      0006      0000      C2C5      0000      0192      0000
INSTR LENGTH      DEC 0         0         1         0         1         0         3         0         0         0
CONDITION CODE    DEC 0         0         1         0         0         0         1         0         0         0
PROGRAM MASK      BIT 0000      1111      0000      0000      0000      0000      0110      0000      0000      0000
INSTR ADDRESS     HEX 000000    000D28    0022F6    007B90    001303    000CFC    D1F340    000906    007DA2    000206

4C-UNUSED-00000000      50-TIMER-000B8800     54-T OF D- 00000BB8
```

Figure 24. Example of Test Request Printout

overlaid by a subsequent phase are valid. After all SYM records have been processed and written on the SYM2 area on SYSLNK, ATLEFH2 determines if the symbols are in sequence. If the symbols are not in sequence, the symbol sort phase, ATLEFD1, is fetched for execution. If the symbols are in sequence, the EOJ dump phase, ATLEFH3, is fetched for execution.

When a symbol sort is required, ATLEFD1 reads up to four symbol records from the SYM2 area on SYSLNK into main storage and sorts them according to their address in main storage. See Figure 25 for the sort input record format. It builds a sort-address table, SRTADD, containing the addresses of the sorted symbol records in the symbol input area, SRTIPT. Using the SRTADD table, it then writes the sorted symbol records on the SYM1 area on SYSLNK and fetches ATLEFD2 to merge the sorted symbol strings. ATLEFD2 writes the merged symbol records on the SYM2 area on SYSLNK and fetches ATLEFH3 to perform the EOJ dump printout. See Figure 25 for the sort and merge output record format.

1. Core map record - Each physical record is 402 bytes and has the following format:

Code	Number of Records	1 to 50 Logical Records
A	B	C

A 1 byte - containing an M

B 1 byte - Bits 0-6 - number of logical records in physical record
Bit 7 - last record indicator
0 - not last record
1 - last record

C Variable - contains 8-byte logical records having the following format:

Bytes	Contents
0-2	Start or restart address of phase
3-5	Phase displacement
6-7	Phase number

2. Sort input record - Each physical record is 640 bytes and has the following format:

Code	Number of Records	1 to 58 Logical Records
A	B	C

A 1 byte - containing an S

B 1 byte - Bits 0-6 - number of logical records in physical record
Bit 7 - last record indicator
0 - not last record
1 - last record

C Variable - contains 11-byte logical records having the following format:

Bytes	Contents
0-7	Symbol
8-10	Address of symbol (displacement of symbol within phase)

3. Sort and Merge output record - Each output record for the symbol sort and merge phases is 641 bytes and has the following format:

Code	Number of Records	1 to 58 Logical Records	
A	B	C	D

A 1 byte - unassigned

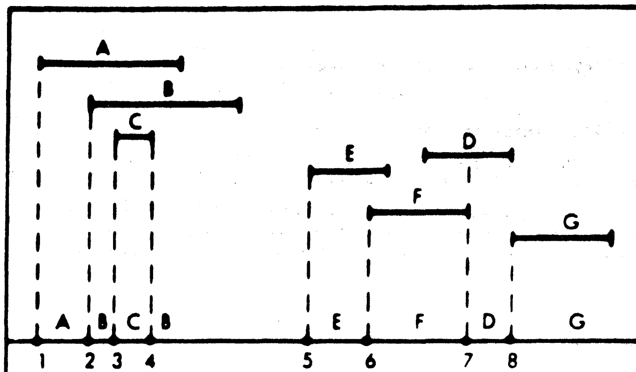
B. 1 byte - Bits 0-6 - number of logical records in physical record
Bit 7 - last record indicator
0 - not last record
1 - last record

C Variable - containing 11-byte logical records having the final format:

Bytes	Contents
0-7	Symbol
8-10	Address of symbol (displacement of symbol within phase)

D 1 byte - containing end-of-string indicator (? in EBCDIC)

Figure 25. Record Formats for Post-Processing



Where A, B, C, D, E, F, and G are phases in main storage at the time the EOJ dump was taken.

Numbers 1, 2, 3, 5, 6, and 8 are the start addresses of the phases or sections of phases in main storage at the time of the EOJ dump.

Number 4 is the restart address of phase B.

Number 7 is the restart address of phase D.

The addresses represented by numbers 1 - 8 are contained in the core map records written on SYSLNK for use in the EOJ dump phase, ATLEFH3.

Figure 26. Schematic of Main Storage at the Time of User EOJ

ATLEFH3 tests for a disaster continue situation. If a disaster continue situation has occurred, no EOJ dump is given and ATLEGO1 is fetched to terminate this job. ATLEGO1 is also fetched if the user requested no dump at normal EOJ.

If a dump is required, this routine formats the EOJ dump information written on the Autotest work file on SYSLNK according to the user specifications from the PCC (program control) card. It writes the following information on SYSLST:

1. The 155-byte EOJ dump record in the following order:

   a. Reason for EOJ--either normal EOJ or as determined from the cancel code

   b. Last user PSW--in hexadecimal

   c. General registers (0-15)--in hexadecimal

   d. Floating-point registers (even-odd pairs 0, 2, 4, 6)--in hexadecimal

   e. 48-byte user communication area--in hexadecimal

```
NORMAL END OF JOB
USER PSW - FF1500FE4000F872

GR 0-7  00003C94 00002808 5000299A 0000399A 0000001E 000033CC 00000015 00000032
GR 8-F  A0002F4C 00000000 40002F70 00003C94 00002C18 00003C18 80002874 00003EF8
FR 0-6  4118000000000000  4110000000000000  44AD71E000000000  4295400000000000

COMREG  F1F061F2 F761F6F6 20002000 00000000 00000000 00000000 C4C1C160 F0F14040 0000FFFF 0000FFFF 000045F8 0000

-SUPERVISOR-
000140  000742A0 010F41AA 84984400 A0045890 A0044220 A0009140 A0014710 0198D207 02209008 68009058 68209060 68409068
000170  68609070 98989010 82000220 4400A004 5890A004 98189030 98980220 82000038 928484F8 47F00188 9680A000 41100030
```

```
001880  91011005 4780F44E 9620101E 0A089108 10030784 9610101E 0A089608 101E0A08 D202C023 F49747F0 F0741923 07841823
001880  07F40000 000000F2 0CF268F2 D4F2D4F4 54F20CF4 54000005 C702C5E8 D2C5E840 C2D6C640 00C70013 09040000 00000000
0018E0  00000000 00000000
```

```
-LABEL LENGTH- 0000

        002000  00000000   00000000   00000000   00000000    002010    00000000   00000000   00000000   00000000
          CH
          MN
        0027A0  DITTO
```

```
   S4          WEST0259   WEST0263   WEST0267   WEST0271              WEST0275   WEST0279   WEST0283   WEST0287
   S3        WEST0258   WEST0262   WEST0266   WEST0270              WEST0274   WEST0278   WEST0282   WEST0286
   S2      WEST0257   WEST0261   WEST0265   WEST0269              WEST0273   WEST0277   WEST0281   WEST0285
   S1    WEST0256   WEST0260   WEST0264   WEST0268              WEST0272   WEST0276   WEST0280   WEST0284
0029E0   C1C1C1C1   C1C1C1C1   C1C1C1C1   C1C1C1C1    0029F0    C1C1C1C1   C1C1C1C1   C1C1C1C1   C1C1C1C1
   CH    A A A A    A A A A    A A A A    A A A A               A A A A    A A A A    A A A A    A A A A
   MN
```

```
   S1               IJFFDBL                                                IJFFEX1              IJFFHRU
004020   104C07FE   98B01044   4400102C   50B01044    004030    07F99180   10154780   F1E49120   10154710
   CH       ¤                              •                       9                   1 U
   MN    LPR BCR    LM  LPR    EX  LPR    ST  LPR               BCR TM     LPR BC     MVO TM     LPR BC

   S2               IJFFMAI
   S1        IJFFMU2           IJFFMU1                                                           IJFFRCT
004040   F0AC0A00   45A0F108   91011004   4710F16C    004050    58E01028   44001030   50E01028   48801000
   CH    0             1 Q                   1 %                                      •
   MN        SVC    BAL MVO    TM  LPR    BC  MVO               L   LPR    EX  LPR    ST  LPR    LH -LPR
```

```
007FE0   8F000100   C0000002   C0000100   C0000101    007FF0    C0000200   C0000200   00000000   00000000
   CH
   MN
   END
```

Figure 27. Example of EOJ Printout

2. Supervisor (if requested at normal EOJ)--in hexadecimal starting at location 320 and ending at the start of the logical transient area.

3. Length of user label area (from the 155-byte EOJ dump record)--in hexadecimal.

4. Main storage--from the end of supervisor to the end of main storage in one of the following formats. Hexadecimal only or any combination of hexadecimal and

   a. Character (alphameric)

   b. Mnemonic

   c. Symbolic

For a 16K machine, all of the user dump information from main storage is contained on SYSLNK. For a 32K or greater machine, up to 32K of the user dump information from main storage is contained on SYSLNK. The rest of the user information is dumped from main storage itself. Figure 27 shows an example of the EOJ dump printout.

After the EOJ dump is performed, ATLEGO1 is fetched to terminate the Autotest job. It reads a card from SYSIPT and checks for a /&, ATEOF (Autotest end-of-job) or job control (//b̸) card. If either the /& or ATEOF card is read, job control is fetched to perform the EOJ step. If a job control card was read and no ATEOF card was found, an error message indicating the ATEOF card was out of order is written on SYSLST and a check for an EXEC card is made.

If an EXEC card having a blank operand is read, an error message indicating more than one EXEC card is invalid in Autotest is written on SYSLST and the job is canceled. If the EXEC card does not have a blank operand, Autotest cancels the job because it has read a control card that should have been processed by job control.

Figure 28 shows the I/O flow for the post-processing phases. Figure 29 shows the storage allocation maps for the post-processing phases.

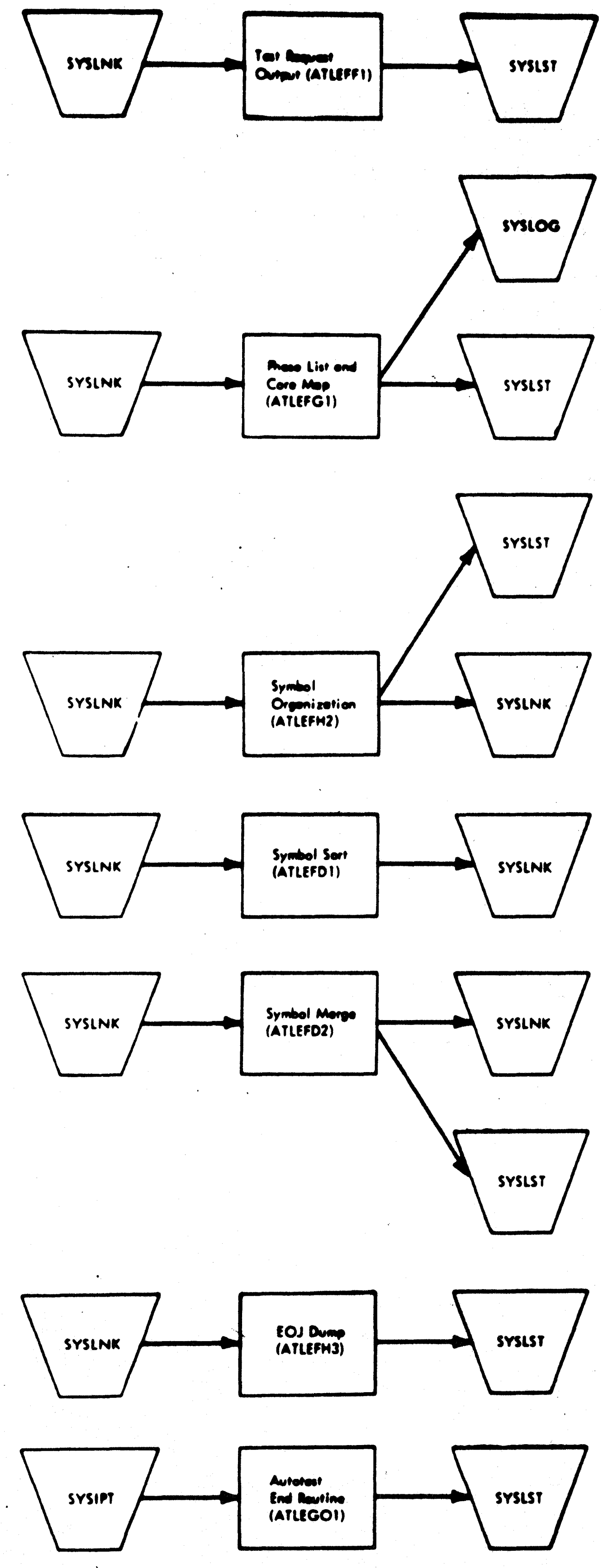


Figure 28. Post-Processing I/O Flow

1. ATLEFF1

Supervisor
Autotest Communication Area (116 bytes)
Logic Module (CPMOD) - 500 bytes
DTF Table for SYSLST (160 bytes)
Initialization - 14 bytes
I O Print Area - 121 bytes
I/O Area* (420 bytes)
Logic and Constants (5000 bytes)
Test-Request Control Table (2200 bytes)
Character and Mnemonic Conversion Table (IJVTAB) - 1280 bytes
Problem Program Area

* During initialization, 200 bytes of this area contain the routine to build the test-request control table.

Note: Byte counts are approximate.

2. ATLEFG1

Supervisor
Autotest Communication Area (116 bytes)
Logic Module (CPMOD) - 500 bytes
DTF Table for SYSLST (160 bytes)
Initialization - 14 bytes
I/O Print Area - 121 bytes
Logic (2100 bytes)
Constants and I/O Areas (1200 bytes)
USEREND: Start and End-Address-Sequence Tables
Phase List Table*
Problem Program Area
Phase Fetch/Load Table

COREEND**

* After the phase list is written on SYSLST, the phase list table is overlaid by the start and end-address-sequence tables.

** COREEND is 16K if the machine size is 16K or 32K if the machine size is 32K or greater.

Note: Byte counts are approximate.

3. ATLEFH2

Supervisor
Autotest Communication Area (116 bytes)
Logic Module (CPMOD) - 500 bytes
DTF Table for SYSLST (160 bytes)
Initialization (14 bytes)
I/O Print Area - 121 bytes
Logic and Constants - 1200 bytes
I/O Areas - 1500 bytes
Problem Program Area

Note: Byte counts are approximate.

4. ATLEFD1

Supervisor
Autotest Communication Area (116 bytes)
Logic Module (CPMOD) - 500 bytes
DTF Table for SYSLST (160 bytes)
Initialization (14 bytes)
I/O Print Area (121 bytes)
Logic (900 bytes)
Constants (500 bytes)
I/O Areas (4325 bytes)
Problem Program Area

Note: Byte counts are approximate.

Figure 29. Storage Allocation Maps for Post-Processing (Part 1 of 2)

5. ATLEFD2

Supervisor
Autotest Communication Area (116 bytes)
Logic Module (CPMOD) - 500 bytes
DTF Table for SYSLST (160 bytes)
Initialization (14 bytes)
I O Print Area - 121 bytes
Logic (1200 bytes)
Constants and I O Areas (3625)
Problem Program Area

Note: Byte counts are approximate.

6. ATLEFH3

Supervisor
Autotest Communication Area (116 bytes)
Logic Module (CPMOD) - 500 bytes
DTF Table for SYSLST (160 bytes)
Initialization (14 bytes)
I O Print Area (121 bytes)
Logic and Constants (3600 bytes)
I/O Areas (4000 bytes)
Character and Mnemonic Conversion Table (IJVTAB) (1280 bytes)
Problem Program Area

Note: Byte counts are approximate.

7. ATLEGO1

Supervisor
Initialization (14 bytes)
I/O Print Area (121 bytes)
DTF Table for SYSIPT (126 bytes)
Logic and Constants (300 bytes)
Card Input Area (80 bytes)
Logic Module (CPMOD) - 500 bytes
Problem Program Area

Note: Byte counts are approximate.

Figure 29. Storage Allocation Maps for Post-Processing (Part 2 of 2)

This routine is executed whenever the user's program goes into an unending loop, or destroys the supervisor or Autotest control program and the operator's usual job termination procedure is unsuccessful.

When a disaster-continue situation occurs, the operator performs the following:

1. Dumps main storage with a stand-alone utility program.

2. Performs the standard IPL procedure to restore the supervisor.

3. Ensures that all Autotest I/O logical unit assignments are the same as at the time of the disaster. This is done by inserting the ASSGN cards for the user program into the job stream. (See step 4.) (If the user program utilized the same set of logical unit assignments as the installation IPL set, this would not be necessary.)

4. Inserts the following cards into the input stream, followed by all cards that have not been read:

   a. A JOB control card for the program.

   b. ASSGN cards for the program, if needed.

   c. A disaster-continue control card. The format of this card is:

      // EXEC ATLECONT

5. Replaces the remainder of cards (from the point of intervention) in the input stream.

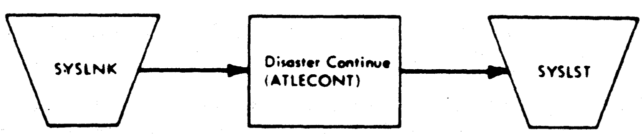


Figure 30. Disaster Continue I/O Flow

After job control reads the EXEC ATLECONT card, it fetches the disaster-continue phase, ATLECONT, from the core image library. ATLECONT turns on the disaster-continue switch in the Autotest communication area and reads the Autotest communication area record from SYSLNK into the beginning of the problem program area. It then fetches ATLEFF1 to process any test-request output data generated up to the point of disaster. After writing the phase list on SYSLST, ATLEGO1 is fetched to finish processing the terminated job and to give control to job control to initiate the next job-step.

Figure 30 shows the I/O flow for disaster continue. Figure 31 shows the storage allocation map for disaster continue.

Supervisor
Autotest Communication Area (116 bytes)
Logic (150 bytes)
CCB's and CCW's plus DTF tables (300 bytes)
Messages and I/O area (200 bytes)
Logic Module (CPMOD) - 500 bytes

Note: All byte counts are approximate.

Figure 31. Storage Allocation Map for Disaster Continue (ATLECONT)

The card to tape (variable) utility program allows the user to build tape records or files of any format or length depending on the amount of storage available for the output record. The maximum length output record that can be built is calculated according to the following formula:

> Maximum record length = Address of the uppermost byte of the problem program area as determined by the IPL program minus the end address of the card to tape (variable) utility program, ATLEJCTV.

This program is called from the core image library into main storage when job control reads an EXEC ATLEJCTV control statement.

Multiple files can be written on the output tape (SYS005). This tape can be rewound initially and/or after the last file, or rewinding can be suppressed. Necessary control information is supplied to the program by means of data cards read from SYSIPT. The cards contain control information in columns 1-10. Card columns 11-80 are reserved for the data. A special file delimiter card is required at the end of the data cards. Each tape to be written requires a separate set of data/control cards.

Data cards, which must be punched in Extended Binary Coded Decimal Interchange Code (EBCDIC), must be in the following format:

Card Columns	Punch
1-3	Not used.
4-6	Not checked.
7-8	TM, MR, blank or WT (see Note 1).
9	H if the input data is in hexadecimal format (see Note 2).
10	N--No rewind initially (first data card only). Non-N--Rewind initially (first data card only).
11-80	Data.

An asterisk punched in WT (write tape) or MR (move a record) cards has a special significance. The asterisk is punched in the column following the last data character in these cards and is used to end the data field.

Note 1:
TM--Write a tape mark. The contents of columns 11-80 are ignored.
MR--Move the contents of columns 11, through one column to the left of the rightmost asterisk, to the record output area, OUTAREA.
Blank--Store the contents of columns 11-80 in the output area. All asterisks are treated as data on this type of card only.
WT--Move the contents of columns 11 through one column to the left of the rightmost asterisk to the output area and write the contents of the output area on tape (SYS005).

In cases where the contents of columns 11-80 in successive cards are moved to the output area, the data is placed in contiguous bytes of the output area until a WT card is read.

Note 2: Hexadecimal data (H in column 9) requires 2 card columns for each character. Therefore, if a 35 hexadecimal character record is to be written, 70 card columns are required for the data, plus an asterisk to indicate end of record.

The following format is used for the file delimiter card.

Card Columns	Punch
1	/--Indicates a control card.
2	*--End of data file.
3	N--No rewind at End. Non-N--Rewind at End.
4-80	Not used.

Restrictions on input cards are as shown in the following:

1. No data cards should contain /* or /& in columns 1-2. This card indicates the end of a data file and must only be used following the last data card in the file.
2. No data cards should contain ./ƀ or //ƀ in columns 1-3. When Autotest reads such a card in the data-card file, it is assumed that the user forgot to insert the /* card. The test job is terminated.

When an error condition is detected by ATLEJCTV, a diagnostic message is written on SYSLST and the job is canceled. In the case of a physical tape record of less than 18 bytes, a warning message is written on SYSLST, the record is written on tape (SYS005), and processing continues. A card with /* in columns 1 and 2 is the normal means for terminating the utility. An SVC 14 is issued allowing job control to terminate this job step and initiate the next.

Figure 32 shows the I/O flow for the card-to-tape variable utility program. Figure 33 shows the storage allocation map for the card-to-tape variable utility program.

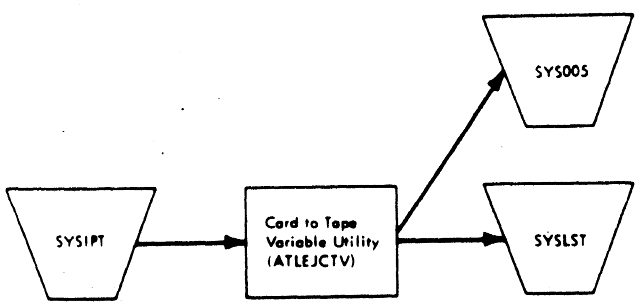


Figure 32. Card to Tape Variable I/O Flow

Supervisor
Logic (700 bytes)
DTF table for SYSIPT (125 bytes)
Logic (35 bytes)
DTF table for SYSLST (160 bytes)
Logic (350 bytes)
Constants, Messages, Literals, CCB's and CCW's (400 bytes)
Card area (80 bytes)
Print area (121 bytes)
Logic module (CPMOD) - 500 bytes
Output record area

Note: Byte counts are approximate.

Figure 33. Storage Allocation Map for Card to Tape Variable Utility

AUTOTEST LINKAGE EDITING

Initialization (ATLEDT) Charts AA-AU

@ATINIT - AB
Routine to determine file track limits for Autotest file during initialization.

@INVALD - AB
Preparation of invalid configuration message before cancellation.

@LAB1 - AB
Initialization of print line format.

ABTERR - AM
Loads ATLEDT1A so that abort error processing can continue. Entered when nonrecoverable errors are found.

ACTCLR - AU
Tests for CLEAR operand.

ACTLOP - AU
Zeros a core image record.

ACTNMP - AU
Tests for NOMAP operand.

ACTNPR - AU
Beginning of the action card processor routine.

ACTNTO - AU
Tests for NOAUTO operand.

ACTRET - AC
Entry point from the action processor (Chart AU).

ACTR16 - AC
Program switch. If the switch is NOP, there is no error and normal processing continues. If the switch has been modified to B by the action processor (Chart AU), an error handling subroutine is executed.

AD1DSK - AK
Starting label of the update disk address subroutine.

ALIGN - AG
Label at the beginning of a subroutine that places origins on a doubleword boundary.

ALNKCD - AQ
Tests for more ER's left to process.

ALNKGT - AQ
Entry point to the autolink routine when autolink is NOT to be done. Provides an exit to load ATLEDT16.

ALNKOF - AP
Entry point to the read input stream subroutine.

ALNKPR - AQ
Starting address of the autolink processor, which is entered whenever a phase has finished processing and autolink has been requested.

ALNKSC - AQ
Tests for end of control dictionary.

ALNKVL - AQ
Initially, loads register 6 with the first control dictionary address. Subsequently, ensures the lowest ER in the collating sequence is in register 6.

CDSIZE - AN
Starting label of a subroutine to check for control dictionary-linkage table overlap.

CHKSYM - AP

CLRLOP - AU
Loop established to clear the core image library.

CNVAHX - AH
Tests for more characters to convert.

CNVHEX - AH
Label at the beginning of a subroutine used to convert hexadecimal to binary notation.

CNVHSW - AH
Tests for a hexadecimal number in the range A-F.

CNVSHF - AH
Converts hexadecimal to binary notation.

CTLSKP - AP
Submodular structure causes skip of cards except an entry card.

DECONT - AJ
Instruction that can be modified within the print subroutine. Set to BCTR to decrease the lines-remaining-count by 1 extra line.

DERCAL - AT

DERDAD - AT
Beginning of subroutine to build core image blocks. Basically, this subroutine:

1. Ensures the text is within the limits of the phase.

2. Finds the core image block the text belongs in.

3. Reads the core image block required by the text into the I/O area.

DERDOK - AT

DERDSW - AT
Program switch used to force continued processing when a zero length control section is found by the ESD processor.

DERITE - AT

DERLOP - AT
Loop to find correct core image block for the text being processed.

DERSW1 - AT

DISKIO - AM
Entry point to the read/write subroutine when the operation code is already set to the desired operation, the disk search address is set up, and both the CCW and the CCB are already correctly set.

DMPHSW - AP
Program switch initialized as a MVC instruction. Modified to an effective NOP by the ESD processor when a dummy phase card is to be built. By the NOP modification, the disk address of the ESD card not yet processed is retained in location COMNRF for use after the phase processing is finished. (Dummy phase cards are treated as actual phase cards.)

DNTCPT - AB

ERRACT - AU
Exit from the action processor. Error messages are initialized in the action processor, but are actually issued during the execution of another initialization routine.

ERROR - AR
Beginning of the error handling subroutine.

ERR002 - AH

ERR035 - AU

ERR036 - AU

ERR044 - AN, AE

ERR044A - AE

ERR044C - AN

ERR050 - AT

ERR070 - AE

ERR093 - AT

ERR094 - AM

EXLOAD - AP

1. Entry point from the autolink routine (Chart AQ) when the control card processing phase is to be loaded.

2. Entry point from the ESD processor when a dummy phase card has been built and a control card processing phase is to be loaded.

3. Entry point when the ESD processing phase is to be loaded.

FNDENT - AP

FNDOP - AU
Loop control label.

FNDVRB - AC
Finds the operation field of the card image in the input stream.

HDGMAP - AC

HDGMSW - AC

HDGOVR - AC

HDNGSP - AJ
Tests to determine if a heading line was just printed. If a heading line was just printed, space one extra line.

INS000 - AD
Entry point when the read SYSLNK subroutine is used to test for the presence of a record. Also used internally by the RDS000 subroutine.

INTPT1 - AB
Begins part one of initialization. Executed after part two.

INTPT2 - AA
Begins part two of initialization. Executed before part one.

INTS01 - AA

LOADSW - AP
Program switch used to cancel individual phase loading. It is set to NOP by the initialization routine (Chart AB) when all phases are in main storage.

LOGPRT - AA

LSETB - AE

LTCDAD - AE
Entry point to the control dictionary search subroutine from the extract phase subroutine (Chart AL). Computes address of the control dictionary. This entry point is used when the calling routine already has a usable control dictionary number.

LTCDNO - AE
Entry point to the control dictionary search subroutine when a test for control dictionary number assignment is required by the calling routine. The control dictionary number was available but because its status was undetermined, entry is made to this subroutine.

LTCDRF - AE
Finds the relocation factor (sign controlled). Also used as an entry point from the phase processor.

LTESID - AE
Label at the beginning of the subroutine that finds control dictionary number, control dictionary address, or relocation factor using the language translator supplied ESID number and the linkage editor constructed linkage table. Entry point when the ESID number is supplied.

NDESLP - AR

NOLNPG - AB

NOTACT - AC
Entry point from the action processor when an error has occurred in the action card operand. Also the normal exit when no action card is found.

NOTCTL - AR

NTESLP - AR

OPEN - AA
Opens both SYSLNK and SYS001.

OVFHDG - AJ

OVFLOW - AJ
Resets lines per page count to 56. Also used as an entry point to the print subroutine. If used as entry point and the first part of the print/carriage control subroutine has been overlaid by the overlay subroutine, exit is provided by the first instruction of the overlay subroutine.

<u>Note</u>: Only the first part of the print/carriage control subroutine is overlaid.

OVRLAY - AS
Beginning of the subroutine that overlays the print/carriage control subroutine (Chart AJ), when a NOMAP option is found. OVRLAY types error messages on SYSLOG.

PRERR - AR

PRINT - AJ
Beginning of the print/carriage control subroutine. If the NOMAP option is selected on an action control card, the overlay subroutine (Chart AS) begins to overlay the print subroutine at this point.

PRNEOF - AJ
Clears the print line area.

PRTLOG - AJ
Entry point from the error processor.

PRTLST - AJ
Sets up the print area for map.

RDEXEC - AP
Entry point to the read input stream subroutine from the initialize routine.

RDNEXT - AP
Starting label of the read input stream subroutine. Normal entry point for this subroutine.

RDPHAS - AP
Sets up fetch of desired linkage editor processing phase.

RDS000 - AD
Beginning of read input subroutine.

READ - AM
Beginning of the read/write subroutine if a read operation is to be performed.

RECFOO - AD
Program switch set to branch and reset to NOP by either the ESD processor or the END card processor. The ESD processor ensures the correct disk address is in location PERIDA (see L/E Common Label section, PERIDA) by branching to the read SYSLNK subroutine at location INS000 after the CCW has been set to NOP. The disk address of the next card image is located by updating the disk until a record is found. Register 2 supplies the correct disk address for location PERIDA. The END card processor used the same technique to locate the disk address of the next card to be processed (put into location NDS000). RECFOO is set to branch to prevent updating beyond the proper disk address.

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

SPACE1 - AJ
Entry point to the print subroutine when an extra space is required.

SRCHCD - AF
Label at the beginning of a subroutine used to find the last duplicate label in the control dictionary.

SRLABL - AF
Tests for a duplicate label.

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

TAPE01 - AA
If SYS001 is a tape, the DTF table is saved by:

1. Writing the DTF table as the first record on SYS001.

2. Writing a tapemark after the DTF table on SYS001.

At retrieval time (closing of SYSLNK and SYS001), the DTF table is found by first backspacing the file and then backspacing the record.

TISESD - AR

TSTMAP - AJ
Tests for the MAP option by testing MAPSW.

TYPEVB - AQ

VALDVE - AM
Tests for a valid symbolic unit (SYSLNK-SYS001).

WRITE - AM
Beginning of the read/write subroutine if a write operation is to be performed.

XTPHGT - AL
Builds an 11-bit phase number in a register.

XTPHNO - AL
Starting label of a subroutine that extracts the phase number from the control dictionary.

ESD/Initial SYM Card Processor (ATLEDT10) Charts BA-BJ

@ESD - BA
After the input card image is checked to see if it is a SYM card, and it is found not to be one, this routine puts the starting storage address of the card image into register 6 and the ESID number of the item into register 5 as preparation for further processing.

CHKNPH - BA
Beginning of the ESD processor phase, ATLEDT10.

CNCALK - BC
The instructions starting at this label cancel autolink by moving a hexadecimal 'FF' into the variable field of the card image. This indicates that no autolink should be attempted on this ER either because autolink has already been tried or because autolink should never be performed.

EISDPC - BC

ELBCER - BD

ELBCM - BD
Beginning of CM (common) processing. Whenever both the ESD item in the card image and the control dictionary entry are commons with matching labels, puts the common with the longest length in the control dictionary. If commons with different labels are found, posts the new common in the control dictionary. The map processor of the linkage editor calculates a total length of commons. This total, or cumulative, length is used to adjust the phase origin.

ELBINT - BH
Posts the card image to the control dictionary.

ELBDSD - BF

ELBELR - BE

ELBER - BE
Beginning of terminal ER processing.

ELBGSD - BF

ELBINT - BH
Entry point to ELBNCD routine if control dictionary type is ER.

ELBLD - BG
Beginning of terminal LD/LR processing.

ELBLDR - BG

ELBNAS - BG

ELBNCD - BH

ELBNLR - BH

ELBPC - BD

ELBSD - BF
Beginning of terminal SD processing.

EPHLOP - BJ
Tests the control dictionary entry to determine if it is an unassigned LD/LR.

EPHSCD - BJ

EPHSCN - BJ
Loop control label.

ERR040 - BB, BD

ERR041 - BH

ERR042 - BB

ERR043 - BF, BG

ERR045 - BC

ERR046 - BD

ESCNCD - BJ
Sets up for control dictionary scan. Whenever a new ESD entry is posted, the ESD processor tries to assign any unassigned LD/LR's in the control dictionary. It sets up a loop to search for the unassigned LD/LR's. If the ESID has been processed, the ESD processor flags the LD/LR as assigned.

ESDIST - BA
Tests for the first ESD/SYM record.

ESDNXT - BA
A check is made to determine if the input card image was a SYM card. If not, a branch is taken to label @ESD. If it was, the @SYMSW byte is tested to see if this is the first SYM card encountered in the run. Should it prove to be the first, the disk address is saved so the SYM card processing phase can use it later. The next card image in the input stream is then read.

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

ESDSBM - BC
Loop control label for SD label/name list test.

ESLBCD - BD
Beginning at this label, the ESD processor searches the control dictionary for a label that corresponds to the label of the ESD item from the variable field of the card image.

EUPDCN - BH
Stores the control dictionary number in the linkage table, thereby completing the update procedure.

EUPDLT - BH
Linkage table update begins at this label. Entry point whenever the ESD processor requires the update only.

EUPDOK - BH
Moves the type field from the card image into the type field of the linkage table.

EUPDSW - BH
Tests for a nonprocess SD.

EUPDXT - BJ
Tests for control dictionary/linkage table overlap (see label ELBNCD).

EUPTRY - BH
Sets up for exit to a subroutine. Goes to LTESID to get the control dictionary number.

PRSDPC - BC
Beginning of SD/PC processing within the ESD processor.

TXT/REP, RLD, and END Card Processor (ATLEDT12) Charts CA-CJ

ACSLTH - CH
Some compilers supply control section length in the END card. This routine processes the control section length for this special case.

CLREXT - CJ

EISCSL - CH
At this location a length was found in the END card. The next possible phase origin can now be calculated.

EISXFR - CG

ENCRLT - CH
Beginning of a loop to clear the linkage table. The table is first built during initialization. Whenever a module is finished processing (signaled by an END card), the linkage table is cleared and the starting address becomes the next available table address.

ENDPRC - CF
Beginning of the END card processor.

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

ENDSCD - CH
Searches control dictionary for unassigned LD/LR's and ensures the control dictionary number for such items is negative.

ENDXFR - CG

ENOTOO - CF

ENOXFR - CG

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

ENUNAS - CH

ERR000 - CA

ERR013A - CC

ERR051 - CC

ERR055 - CD

ERR058 - CH

ERR091 - CJ

ERR091A - CJ

IJVDTH - CA
Beginning of ATLEDT12 phase. This phase processes TXT, REP, RLD, and END cards.

MOVPER - CF

OTHINC - CA

OTHTYP - CA
Loop control label.

REPROC - CC
Beginning of the REP card processor.

REPTXT - CC
Loop control label.

RLBYWR - CE

RLCONS - CD

RLDPRC - CD
Beginning of RLD pass-1 processing.

RLRET - CD

RLSTP - CD

RLSW1 - CD
Program switch set and reset within ATLEDT12 phase. Setting (NOP or branch) determines if the R and P pointers are to be processed. Several RLD items can have the same R and P pointers. Only the first set of identical R and P pointers are processed.

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

TSTESD - CF
Reference for execute instruction.

TXTALL - CB
Moves text to the work area and provides an exit for the text processor.

TXTGET - CB
Loop control label.

TXTPRC - CB
Beginning of the text processor.

WRS001 - CJ
Beginning of the write SYS001 subroutine.

WRST01 - CJ
Label of the instructions that perform I/O on SYS001 when SYS001 is a tape unit.

Control Card (INCLUDE, PHASE, and ENTRY) Scanner (ATLEDT14) Charts DA-DG

CHKRP - DB

CHKRPN - DF

CMDEL - DG
Beginning of a subroutine that checks for a comma.

CRDEND - DF

DECDSP - DE
Beginning of decimal displacement processing for the operand of a phase card.

DELEXT - DD

DSPRTN - DD
Beginning of displacement operand processing.

ENTALK - DF
Provides an exit from the ATLEDT14 phase of the linkage editor when a blank ENTRY card is found.

ENTCRD - DC
Beginning of the ENTRY card processor.

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

ENTPRT - DC

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

ERR010 - DA

ERR012 - DG

ERR013 - DG

ERR014 - DB

ERR015 - DF

ERR025 - DD

ERR033 - DB

EXTRCT - DG
Beginning of the extract subroutine. Used to put the operands into a work area.

FINDEL - DG
Beginning of a loop to find the delimiter that is adjacent to an operand.

FINDND - DB

FNDDEL - DF
Tests for a qualifier in the operand.

FONDEL - DG

GETVRB - DA

IJVDCN - DA
Beginning of the ATLEDT14 phase, the first control card processing phase.

INCCRD - DB
Beginning of the INCLUDE card processor.

LABST - DB
Builds the name list of control sections from the operand field of an INCLUDE card. These control sections are subsequently used to build a phase (see NMELST).

LKQUO - DD

NAUTO - DF
Tests for a NOAUTO option.

NTABS - DF

PHCRD - DD
Beginning of the phase processor in the ATLEDT14 phase. This part of the phase processor performs two basic functions:

1. Determines which optional operand has been used.

2. Validity checks the PHASE card image.

QUAPRO - DF
Beginning of qualifier processing for the operand of a PHASE card.

RESET - DG

RUNNING - DD

SAVCTL - DC
Provides exit to the autolink processor. Saves the disk address of the control card before exiting.

SEEBLK - DA

SETMDS - DB

SKIPB - DG
Beginning of the skip blanks subroutine.

STORR5 - DE

SUBMOD - DB

TSTLIM - DE

TSTNEG - DE

UPDATE - DG
Beginning of a subroutine to point to the next character in the input stream.

Control Card Processor (ATLEDT16) Charts EA-EG

ALKERR - EG

ALKFND - EG

CHEKQU - EC

CHQUAL - EC

CIMBLK - EE
Phase size check.

CINOBL - EE
Number of core image blocks required by this phase is equal to number of bytes loaded divided by block size. Add one block for any remainder.

ERR020 - EB

ERR021 - EB

ERR022 - EC

ERR023 - EA

ERR031 - EG

ERR081 - EA

ERR082 - EF

ERR092 - EE

IJVDTL - EA
Beginning of ATLEDT16 phase (control card terminal processing).

INCERR - EG

INCFND - EG

INCGET - EG
Entry point to the terminal INCLUDE card processor when an autolink is in progress.

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

INCLPR - EG
Beginning of INCLUDE card terminal processing.

INCRED - EG

ISDISP - ED
Processing when the phase card operand is a displacement.

ISROOT - ED
Processing when PHASE card operand is ROOT.

LABINV - EA

NEWPHS - EF
Beginning of processing for the first PHASE card.

NOT1ST - EB

NODUPL - EC

NTROOT - EC

PHSPRC - EB
Phase name processing.

PHXADD - EA

SCNSYM - EC

SCSYM1 - EC

SETPHS - EA

WRPHCD - EA

WRTHDR - EA

WRTRFR - EA
Saves the transfer address.

MAP Processor (ATLEDT18) Charts FA-FD

ADMDSW - FA
Program switch to indicate first time through the MAP processor.

CANCEL - FD
Exits to cancel.

CLRCMN - FA

CNVBIN - FD
Beginning of a subroutine to convert binary to hexadecimal.

CNVLOP - FD

COMCHK - FA
Beginning of common (CM) processing in the MAP processor. The cumulative length of discretely named commons is calculated at this point in the program.

CSCAN - FB
End of control dictionary test.

DUPLAB - FC
Tests for duplicate label.

ERRO85 - FC

EXSIXTA - FA

EXSIXTY - FA
Sets a switch when a transfer label is found.

EXECWR - FC
I/O for error.

EXTNLP - FC

EXTNSW - FC

EXTSCN - FC
End of control dictionary test.

FCHRLD - FC
Entry point when NOMAP option is found.

LDRGO - FB

LDRSCN - FB
End of control dictionary scan.

LDRTSD - FB

MAPCST - FB
Sets up for a scan of the control dictionary searching for control sections belonging to the phase previously identified in this routine.

MAPHAS - FB
Beginning of a control dictionary scan searching for phase entries.

MAPHNM - FB
Sets up MAP print area with the proper phase name.

MAPLDR - FB
Sets up for scan of control dictionary searching for LD/LR's.

MPLDSW - FB

NOTRFR - FA

OUT - FD
Exits to job control program.

OVRLSW - FC
Tests for overlay of root phase.

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

PHSCAN - FB
End of control dictionary test.

PHSTOR - FA
Reinitializes the phase information in location PHEADR (see common label section).

PREXTN - FC

PRTMAP - FA

SCNCMN - FA
End of control dictionary test.

SETJCS - FD
Sets job control flag to show good linkage editor output.

TERSXY - FC

TRYROT - FB
Tests for root phase before setting up root message in MAP print area.

UNRSPC - FC

RLD Post-Processor (ATLEDT1A) Charts GA-GH

@CHKEND - GE
Routine which checks for end of phase directory, or end of core image directory during search for desired Autotest phase. The first 6 characters of the name are checked until a match is found; if none is found, an exit is taken to abort the job.

@CMP - GE
During the search for a phase, when the first 6 characters of the phase name in the table match those of a phase name in the phase or core image directory, the entire name is compared to find the exact phase sought.

@EXIT - GE
If the Autotest phase desired is not found, an exit from the search loop is taken to abort the job; if found, the exit from the loop leads to a routine for writing phase entry information on disk in work area for use by the system at // EXEC statement time. Having the phases located and information recorded improves Autotest performance.

@FOUND - GE

@NOTFRST - GE

@RDDSK - GE
A block of the phase directory is read and scanned for phase name of the Autotest phase(s) desired. If the first 6 characters of the Autotest phase name do not agree with those of any entry in this phase directory block, the next block is read. When agreement is found, this block is checked for the entire Autotest phase name.

@SER4 - GE
Routine which constructs the phase headers in the work area from the phase directory (or core image directory) which will then be used at execute time to retrieve the desired Autotest phases more efficiently.

@UPDATE - GE
During scan of directory entries, address is updated to look at next directory entry of a directory block read into core; or if all entries of a block have been checked, the disk address is updated to read the next block of directory entries.

ABORT - GG
Routine for logging error message on SYSLOG and map on SYSLST (if not same device as SYSLOG) before canceling job which cannot be carried to its normal conclusion.

BLKHDR - GE
Routine to build 20-byte core image directory headers.

BLKLOP - GF
Beginning of a loop to read 28-byte phase headers from the system work area, and to build 20-byte core image directory headers.

CANCLE - GG

CLOSE - GH
Beginning of a subroutine to close SYSLNK. To perform a close operation, the DTF information is required. The DTF table was stored during initialization as the first record on SYS001. If SYS001 is a disk, the DTF information can be directly accessed. If SYS001 is a tape, the close subroutine must backspace to the first tapemark and then backspace to the beginning of the first record in order to get the DTF information. Backspace file-backspace record-read DTF-close SYSLNK is the I/O sequence for tape.

ERR100 - GE

EXCP01 - GH
Backspace either file or record subroutine.

EXIT - GF
Normal exit from this phase by fetching next phase ATLEDT1C.

ISCLOS - GH
Issues close to SYSLNK.

IJVDLD - GA
Entry point to this phase, ATLEDT1A.

RADD4 - GA

RDOK01 - GH

RDS001 - GH
Beginning of the read from SYS001 subroutine. Used primarily to get the pass-1 RLD information.

RDTP01 - GH

RECF01 - GH
Updates the disk address after the I/O operation if SYS001 is disk.

REP001 - GH
Beginning of a subroutine to reposition SYS001.

RESDCN - GC

RLADCN - GC

RLCTER - GB
Counts the number of unresolved ADCON's.

RLDCON - GC
Beginning of pass-2 RLD post-processor.

RLDOP - GA

RLDOR - GB

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

RLDRET - GA
Common entry point used during pass 2 whenever an RLD item has finished processing. A test is made for more RLD items on a card image followed by a test for more RLD's on SYS001.

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

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

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

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

RNXTRN - GB
Tests for unresolved ADCON.

ROTSID - GC
Counts ADCON's outside the phase limits.

TSTCNT - GD
Tests to determine if any error diagnostic information is available. If error diagnostic information is found, it is converted to unpacked decimal and printed on the MAP.

TSTSID - GD
Sets up MAP information (number of ADCON's outside the phase limits) in a test register. If the register is zeroed, there is no MAP information. If the content is nonzero, MAP information is printed.

TSTUNR - GD

VERLOP - GE
Beginning of a loop to read and verify all core image blocks written by linkage editor. All verification occurs at this point rather than after each individual write operation.

WRTLOP - GE
Beginning of a loop to write the 20-byte core image directory headers in the directory.

SYM Card Processor (ATLEDT1B) Charts HA-HD

@B - HA
Routine which calculates displacement of symbol within phase from assembled address, relocation factor, CSECT origin and phase origin. The SYM record is assembled and moved to the output area where the records are blocked.

@CTL1 - HA
Register 2 is zeroed. Register 2 is used to keep track of the total number of text characters to be moved, and the residual count of characters still to be moved, if any, after each return from the @GETCH subroutine.

@CTL2 - HA
Registers are prepared for entry to @GETCH subroutine which is then used to retrieve the organization byte from the SYM card field. The character length of the symbol (label) is then determined from bit structure of the organization byte.

@CTL3 - HA
Symbol address and symbol name are retrieved by use of @GETCH subroutine. It is determined if the symbol is associated with a DC statement; if so, @GETCH subroutine is again used to get the data-type byte. The form of the data determines how many characters must be bypassed to get to the next logical record. These characters are then moved by the @GETCH subroutine.

@GETCH - HB
Entry to a subroutine which moves SYM card information from I/O area to work area. If all characters have been moved, a branch occurs to read the next card image.

@GETER - HB
Part of @GETCH subroutine which is entered when this phase reads a card image not a SYM card. Register 4 is loaded with a hex 'F0' equivalent which becomes the modification register in an execute instruction. The execute instruction is then applied to a branch instruction and results in an unconditional branch. The branch instructions thus affected are those which immediately follow return from the @GETCH subroutine; thus this method affords a means of returning from the subroutine to various desired portions of the main program rather than simply continuing at the link register address.

@GET0 - HB

@GET1 - HB
One character at a time is moved from I/O area to work area by this loop of the @GETCH subroutine.

@LIOA - HC
If output block is not full of records, a return from subroutine @LIOSM is taken without an actual I/O operation.

@LIOB - HC
Record count per block is incremented each time a logical record is moved to the output blocking area. Registers 3, 4, and 5 are used as the starting address in the output blocking area, the indexing register and limiting register respectively to determine when the output area is full of records and must be written.

@LIOCA - HC
Linkage to the physical I/O routine causes actual write of output record when block is filled. "Records per block count" is reset and register 3 is again repointed at the starting address for the first record of a new block to be assembled.

@LIOCL - HC
Routine which forces a record to be written when the last SYM card has been processed, an END card has been read, and the block is only partially filled with records.

@LIOSM- HC
Subroutine which puts a logical record into the blocked-record output area, determines if the block is ready to be written, and links to the physical I/O routine if it is.

@PIO - HD
Subroutine which checks for space on disk each time an output record is to be written, updates the record number in the write CCW, links to the DISKIO routine for actual writing of the record, and updates the seek CCW address.

@CIOA - HD

@PIOAB - HD

@PIOB - HD
Seek address is updated. For discussion of this procedure see note on Chart HD.

@PIOC - HD

@PIOD - HD

@1 - HA
Test is made to detect when an END card has occurred in the input card-image stream. If no END card, next card will be read. If it is an END card, the last output block for the phase will be written, the disk address for the next SYM card, if any, is saved and return is to the END processor.

@2 - HA
If SYM card is detected with format error (such as blank fields) or another type of card not an END card is found, a switch is set to indicate an error and a return to @GETCH subroutine is taken to read the next card.

IJVDXL - HA
Entry point to phase ATLEDT1B.

Write Control Dictionary and Autotest Communication Area on SYSLNK (ATLEDT1C) Charts JA-JC

@CDMA - JA
The control dictionary entries are placed as records into the blocked output area. This loop is continued until all entries have been moved. The address of the last entry of the control dictionary is increased by 16 bytes to obtain the next entry. Since the control dictionary is constructed in reverse from high storage downward, it is accessed beginning at the address of the last entry. If the entry is a PHASE card entry, the length of the common area is added (bytes 8-10) to the phase origin and the updated phase origin is returned to the control dictionary.

@CDMB - JA
After all control dictionary entries have been moved as records to the output blocking area, a test is made to see if there is a partial block of unwritten records. If there is, the block count is set to simulate a full condition and the partial block of control dictionary entries is written on SYSLNK.

@CDMB1 - JA

@CDMC - JA

@CD1 - JA
Disk address to begin seek for recording control dictionary entries is chosen, depending upon presence or absence of symbol records. The address is updated for the actual write to the next record. The storage addresses of the beginning and end of the control dictionary are obtained.

@LCD - JB
Subroutine which does actual move of control dictionary entries to output area, determines when record block is full and branches to @PCD if it is.

@LCD1 - JB

@L1 - JA
Entry point of a loop which branch-and-links to an update subroutine. The disk address will be updated as often as necessary to ensure that the next output occurs on a new track because the next phase is about to be fetched.

@PCD - JB
Subroutine which determines if next output block will exceed the disk space available; should this be the case, a message is printed on SYSLOG and a B-transient phase is called to cancel the job. If there is disk space available, a branch-and-link is taken to perform the actual physical I/O operation. Upon return, this subroutine leads into the disk address updating routine.

@PCDA - JB

@PCDB - JB

@PCDC - JB

@PCDER - JB

@PCDUP - JB
Subroutine which updates the disk address between writing output record blocks. For discussion of this procedure see note on Chart JB.

@UPRLF - JA
When control dictionary entries are to be moved to the output area for writing on disk, they are first checked to see if they are PHASE card entries or not. If they were phase entries, they are treated as described under label @CDMA. If not phase entries, this routine (@UPRLF) is entered; the relocation factor for the item described in this entry is obtained and tested to see if positive or negative. In either event, the relocation factor is algebraically added to the length of the common area to obtain a new updated relocation factor which has been modified to take into account the existence of the common area. This updated relocation factor is put into the control dictionary entry before it is written on disk.

IJVDDO - JA
Entry point to start phase ATLEDT1C.

IXVLD - JB
Subroutine which writes a message on SYSLOG(if an error condition has occurred) to identify the nature of the error.

CONTROL CARD ANAYLSIS

Initialization (ATLEFC1) Charts KA

INITROOT - KA
Initializes track limits, carriage line count, test-request control file and Autotest table-file start and end addresses. It clears the save area and work areas, and opens SYSIPT and SYSLST. It then sets up the page header line, prints the header line, and fetches ATLEFC2.

NXTPHASE - KA
Fetches ATLEFC2.

Autopatch Diagnostics (ATLEFC2) Charts KB-KV

ABORT - KR
Moves an error code into the first byte of the error table in the print area.

ABORT1 - KR
Writes the autopatch card image and error code on SYSLST and branches to cancel job.

ADDM - KM
Subtracts the minus symbol adjustment value from the symbol address.

ADDP - KM
Adds the plus symbol adjustment value to the symbol address.

ADDSWI - KB
This routine is executed when an ADD card is read. It turns on the ADD card switch, resets the CON card switch, and resets the switch that indicates that the Autotest table entry is not closed.

ADD240 - KG
This routine is executed if the hexadecimal character being converted to a printable character is less than A, indicating that it is a hexadecimal character between 0 and 9. It adds F0 to the character to convert it to a printable character.

ADJRTN - KL
This routine is executed if there is an adjustment factor for the symbolic address present. It initializes the loop counter to the maximum number of characters in the adjustment factor plus the comma delimiter.

ADJSTP - KL
This routine is executed when a minus sign is found in a symbolic address (e.g., HERE-1234). It turns on a switch to indicate a minus adjustment factor and turns off the plus adjustment factor switch.

ASSUMCS - KR
This routine is executed when the qualified phase number is equal to the phase number of the control section record that was read. The label is a program switch normally set to an unconditional branch to MOVECS. It is set to NO-OP whenever a CSECT name has been supplied by the user.

ASSUMPH - KQ
This routine is executed if the assumed phase switch is on when processing the CSQ card. This means that the phase to be used is the first one in the control dictionary. It sets up the CCW and reads a physical record. It calculates the main storage address of the last logical record of the physical record and tests to ensure that the logical record is a phase record. It saves the phase name and turns off the assumed phase and the CSQ processed switches. It loads the return address of CHKTPA into register 5 and branches to PHQRTN.

ASTCK - KC
Sets the loop counter to 2, the number of decimal digits in the asterisk column indicator of the autopatch control card. It tests the asterisk column parameter, writes the autopatch control card image on SYSLST and reads another card from SYSIPT.

ASTER - KH
Tests to see if the card column of the patch indicator on the autopatch control card is 80 or less. If it is more than 80, it branches to an error routine. If it is 80 or less, the value is converted to binary and stored.

BACKDECR - KT

BALCHECK - KK
This routine tests to see if either the patch instruction or the displaced instruction from the core image library is a BAL, BALR, or SVC instruction. If either condition occurs, an error routine is entered because these instructions are invalid.

BALTEST - KC
Tests for an error condition and branches to an error routine, if error switch is on. If no error has occurred, a test is made for the type of autopatch card. ATLEFC7 is fetched if the card is a CON or EXC card.

BBACK - KP
This label is a program switch which is normally an unconditional branch. It is set to a NO-OP whenever a further qualified phase is processed.

BTRPH - KP
This label is a program switch which is normally set to a NO-OP and never changed in this phase.

CALADJ - KL
This routine is executed when a comma is found at the end of the symbolic address adjustment parameter. It packs and converts to binary the decimal adjustment value and stores it in ADJCOUNT. It then branches to process the symbol.

CARDAST - KC
This routine tests to see if the asterisk is in the column of the autopatch card specified by the autopatch control card.

CARDID - KB
Initializes the error table address, and register 5 to card input area, turns off ENDSWI (X'02') which indicates no Autotest table entry is necessary, and turns on PHASESWI (X'80') to indicate ATLEFC2 is in main storage.

CDSWI - KD
This routine is executed when a CSQ card is read. It turns on the CSQ card switch and the CSQ processed switch and branches to process the CSQ card.

CDTAPE - KD
Tests the assumed control section switch and stores the CSECT name if it is off.

CHKADJ - KL
Checks the decimal digit characters of the symbolic address adjustment factor (e.g., HERE +1234) for a comma which indicates the end of the parameter. It checks for valid decimal characters and stores the valid character in register 3.

CHKCARD - KB
Tests for the type of card read from SYSIPT. It then branches to the correct routine or fetches the correct phase to process the card.

CHKDCLH - KF
This routine is executed when the hexadecimal character being tested is found to be zero or more. It tests to see if the character is 9 or less. If it is, it indicates that this is a valid hexadecimal character and the next character is tested. If the character is greater than 9, a branch is taken to another routine which checks to see if it is a comma or a quote.

CHKDEC - KJ
This routine checks to see if a hexadecimal character being tested is between zero and nine.

CHKQUOT - KF
This routine is entered when the hexadecimal character is determined to be other than a character between 0 and F. It checks to see if this character is a quote, and branches to read the next card column if it is. If it is not a quote, it branches to test for a comma.

CHKSWI - KS
Restores registers 2 through 15 and tests to see if symbols are being processed and branches to either the symbol error routinue or the cancel routine.

CHKSYM - KL
This routine checks each character of the symbolic address for the following conditions: end-of-card, a period to indicate a further qualification, a plus or minus sign to indicate a positive or negative adjustment factor for the symbolic address, and a comma to indicate end of the parameter. If none of the above are found, it stores the alphabetic character and gets the next character to be checked.

CHKSZ - KG
This routine checks to see if the test-point address is within the limits of the user's phase. If it is not, it branches to an error routine.

CHKTPA - KC
This routine is executed after a card has been identified as an autopatch card. It checks to see if a PHQ card has been processed. If it has not, it turns on the assumed phase switch. If it has, it checks to see if a CSQ card has been processed and turns on either the assumed CSQ switch or the autopatch-control-card-processed switch.

CHRCNT - KJ
This routine is executed after the instruction parameter of the autopatch exchange or add card has been read and the character length of the instruction has been determined by the TRANS routine. It tests the first two bits of the instruction. These bits indicate the type of instruction (such as RR, RS, RX, or SS). If both bits are on, it indicates an SS instruction, which is 12 characters or 3 half words long. If either bit is on and the other off, it indicates an RS or RX instruction which is 8 characters or 2 half words long. If neither bit is on, it indicates an RR instruction which is 4 characters or 1 half word in length. The implied length is then compared against the length calculated by the TRANS routine. If they are equal, the instruction length is saved and a test is made to be sure the end of the instruction is within the phase specified. If the implied length and the calculated length do not compare or the end of the instruction is not within the phase specified, error routine ERINST is entered.

CILER - KK
This routine is executed whenever the patch-point instruction is a BAL, BALR, or SVC. It moves the error code, F, to the save area and processes the error. The above mentioned instructions cannot be patched.

CILREC - KH
This routine reads the record from the core image library containing the patch-point address indicated on the autopatch control card. It calculates this record number in the following manner:

It subtracts the start address of the user's phase from the patch-point address to get the displacement of the address within the user's phase. It divides the displacement within the phase by the record length of 1728. The quotient plus 1 is the record number, and the remainder is the displacement of the patch-point address within the record.

CLCRDSWI - KD

CLRSAVE - KD
This routine is executed when an ADD or EXEC card has been read. It converts the autopatch SVC number to decimal and moves it to the print area.

CNCLPCH - KN
This routine is executed when the error table in the print area is full. It initializes the error table pointer to the first position of the error table and branches to write the card image on SYSLST. It then reads another card.

COMPAR - KP
Checks to see if the phase directory record contains the same name as the phase specified by the PHQ card. If it is not, it updates the input area pointer to the next logical record and tests for end of file and end of physical record. It reads a new physical record if required, or gets another logical record. If the phase names are equal, a branch is taken to MDVEPH.

CONSWI - KB

CONVTPA - KG
Converts the patch-point address to printable characters.

CSQRTN - KQ
Entry point into a subroutine that processes the control section qualification (CSQ) card. It saves the phase and control section names, stores registers 2 through 15 and initializes the CCW to read a control dictionary record, starting at the highest disk address of the control dictionary. This is done because the control dictionary was built backwards in main storage with the first record in the highest position of main storage and succeeding records in lower positions of main storage. When the control dictionary was written on SYSLNK, the last control dictionary record was written starting at the lowest disk address. When retrieving these records, the last record written on SYSLNK will contain the first control dictionary record.

A test is then made to see if the assumed phase switch is on. If it is, a branch to ASSUMPH is made. If it is not, a branch to READCSQ is made.

CSTRRTN - KS

DECBLNK - KH
This routine tests for a blank column or a comma following the card column of the patch indicator (*) on the autopatch control card.

DECPAR - KH
This routine updates the card area pointer to the second operand of the autopatch control card. This operand contains a two digit decimal value indicating the card column of the patch indicator (*) on the autopatch card. The patch indicator (*) in the autopatch card is followed by the instruction to be added or exchanged, or the constants to be replaced. It tests the characters in the first and second columns of the operand for valid decimal characters and stores them in a temporary save area. If a character other than a decimal character 0 to 9 is found in the first column of the operand, it branches to an error routine. If the second character of the operand is other than a valid decimal character, the first valid decimal character is stored in the temporary save area and the second character is tested to see if it is a blank or a comma delimiter. If it is neither a blank nor a comma, an error routine is executed to indicate an invalid asterisk column was specified.

DISKOFL - KU
If the Autotest table containing the test request and autopatch entries exceeds the area allotted to it on SYSLNK, this routine is executed. It clears the print area, and writes error messages on SYSLOG and SYSLST stating '9903I DISK WORK AREA TOO SMALL'.

DSKDECR - KT
This routine decreases the disk address by one each time a physical record from the control dictionary is read. Each time the record number is decreased by one, a test is made to see if the low limit of the record number has been reached. If it has, the head number is decreased by one and the upper limit of the record number is moved into the disk address. The low limit of the head number is checked and, if it has been reached, the cylinder number is decreased by one and the upper limit of the head number is moved into the disk address. A test is then made for cylinder overflow, and the cylinder number is decreased by one, if required. If the minimum value is not reached for the record, head, or cylinder, a branch is taken to return via register 2.

DSKINCR - KT
This routine increases the disk address by one each time a physical record is read. Each time the record number is increased by one, a test is made to see if it is the last record of the track. If it is, the head number is increased by one and the low limit of the track is moved into the disk address. A test is then made to see if the maximum number of heads has been reached. If it has, the cylinder number is updated by one and the low limit head number is moved into the disk address. A test is then made for the cylinder overflow. If the maximum value for the record, head, or cylinder has not been reached, a branch is taken to return via register 2.

ENDSYM - KM
This routine is executed after the adjustment factor has been added to or subtracted from the symbolic address. It turns off the symbol- or decimal-digit-being-processed switch, clears the symbolic address adjustment area, turns off adjustment factor plus and minus switches, loads the return address of ASTCK into register 7 and branches to CHKSZ.

ERAST - KD
This error routine is entered when the autopatch card does not contain an asterisk in the column specified by the autopatch control card. It moves the error code, D, to the save area and processes the error. It then tests for an Autotest control card. If it is not an Autotest card, it writes the card image on SYSLST. If it is an Autotest card, a branch is taken to CARDID+8.

ERDCPNT - KF
This routine writes the card image on SYSLST, reads another card and tests to see if it is an Autotest control card or a job control card.

ERDCPNT1 - KF
Moves the error message 'Card Ignored' into the error table in the print area, turns off the exchange card switch and branches to ERDCPNT to write the card image on SYSLST.

ERDEC - KH
This routine is executed whenever there is an invalid asterisk column specified in the autopatch control card. It moves the error code, C, to the temporary save area, MVCODE, and processes the error. It then branches to ERDCPNT to write the card image on SYSLST.

ERDEC1 - KH
This routine is executed when the card column of the patch indicator (*) on the autopatch control card contains a character less than zero. It decreases the patch indicator (*) loop counter by one and tests to see if the first or second column is being tested. If the first column contains a character less than zero, this is an error condition because this column must be a decimal digit 0 through 9. If the second column contains a character less than zero, a branch is taken to MVASTER to save the first character before testing the second character to see if it is a blank or a comma.

ERINST - KJ
This routine is entered under the following conditions:

1. if a hexadecimal character other than zero through F is detected in hexadecimal parameter.
2. odd number of characters in the constant card.
3. the implied length of the instructions is not equal to the given length of the instructions.
4. the end of the instruction does not fall within this phase.
5. the hexadecimal instruction bit code representation does not equal 4, 8, or 12 characters.

It moves the error code, E, to a temporary save area and branches to an error routine to process the error.

ERRTN - KN
Loads the address of the last error table entry into register 1 and tests to see if 6 error table entries have been made. If so, a branch is taken to restore the error table pointer to the first position of the error table and to write the card image on SYSLST. If the table is not full, it moves the error code to the print area and updates the error table pointer.

ERTRSVC - KD
This routine is executed if the last valid SVC number has been used. It moves the error code, G, to the MVCODE save area and processes the error.

EXCSWI - KB

ERPNT - KD
Writes the card image and error codes on SYSLST, reads another card, and tests for an Autotest control card. If it is an Autotest control card, it branches to CARDID. If it is not, it tests to see if an exchange card was being processed and turns off the exchange card switch.

E1CMP - KS
It compares the qualified CSECT number to the CSECT number on the symbol record. If they are equal, it compares to see if the symbol in the record is the same as the symbol being analyzed. If it is, the symbol displacement is moved from the symbol record to a save area, E1DISP+1. If the CSECT numbers do not agree, a branch is taken to E1PT1.

E1PT1 - KS
Updates the input area pointer to the next logical record of the physical symbol record, tests for the end of the physical record, and branches to read a new physical record or logical record.

E1RD - KS
This routine moves the disk address from E1ADDR to a save area. This address will be used by the CCW to read a physical record. The disk address is then updated and the new address is moved into E1ADDR.

E1RD1 - KS
Initializes register 1 with the address of the CCB and reads a 196-byte symbol record. It tests for I/O complete and incorrect record length. It then calculates the end address of the physical record and pads 15 F's behind the last byte to indicate the end of the physical record.

E2OVFL - KU
Tests to see if the record just written on SYSLNK exceeds the disk area allotted to the Autotest table. If it does not, a return to linkage register address is taken. If the area was exceeded, the DISKOFL routine is entered.

E2PT1 - KU
Tests to determine if an Autotest table record is to be written on SYSLNK.

E2PT2 - KV
Upper disk limit for the Autotest table is decreased because the Autotest table is built backwards on SYSLNK. If the lower track limit is exceeded, a branch to E2OVFL is taken. If it is not exceeded, an Autotest table record is written on SYSLNK.

E2PT3 - KV

FCHPCC - KB
This routine is executed when a PCC card is detected, indicating the end of the test-request and autopatch cards. It fetches ATLEFC7 to write the last Autotest table record on disk. ATLEFC7 then fetches ATLEFC5 to process the PCC card.

FCHPTCH - KD
This routine fetches ATLEFC7 and branches to STRT1 in ATLEFC7.

FCHPTCH1 - KD
This routine saves the CON card image and then branches to fetch ATLEFC7.

FILEEND - KR
Tests for end-of-control dictionary file. If it is not the end-of-file, it branches to read another physical record. If it is end-of-file, this indicates the phase number has not been found and the error routine, GOABORT, is executed.

FINDCOMA - KF
Updates card pointer after a quote is found in a hexadecimal field.

FIRSTCS - KQ
Entered if there is an assumed CSECT, indicating that a CSECT name was not supplied by the user.

GETPCHDG - KC

GOABORT - KR
This routine is entered if the end of the control dictionary is reached before the correct phase number is found. It initializes the branch switches BTRPH, BBACK, TRSWITCH, and MOVECS before entering the CHKSWI routine.

GOBACK - KP
Restores registers 2 through 15 and resets the branch switches (BBACK and BTRPH) to their normal positions before returning to inline processing.

HEADLINE - KK
Moves the header line into the print area and sets up for a skip to beginning of page before the header line is printed.

HEXRTN - KF
Beginning of loop to test one character at a time to determine if it is a valid hexadecimal character between 0 and F. If it is, it stores that chatacter in register 3.

HEXRTN1 - KF
This routine is executed whenever it is determined that the test-point address is in hexadecimal characters. It clears work registers and turns off the symbols-being-processed switch (TPASWI, X'40').

HEXRTN2 - KF
Beginning of subroutine to determine if the parameters on the test-request card are in hexadecimal format. If they are in hexadecimal, it branches to HEXRTN1. If not, it branches to SYMRT to determine if it is in decimal or symbolic format.

ILPEXT - KT

INITINST - KC
This routine is entered when it has been determined that the asterisk is in the same card column of the autopatch card that was specified by the autopatch control card. It updates the card column pointer by one to the first character of the ADD or EXC instruction or the CON card constant.

INSTCHK - KC
Loads the address of the add or exchange instruction into register 2.

IO - KQ

This routine decreases the address of the record just read by one to get the address of the next physical record to be read. It then moves the second byte of the physical record into register 3. In this byte, bits 0 through 6 contain the number of logical records in the physical record and bit 7 is the last physical record indicator. Register 3 is shifted one bit to the right to remove the last record indicator. The number of logical records in the physical record is then multiplied by 16 to calculate the length of the physical record minus the two overhead bytes. The start address of the I/O area is added to the calculated length of the physical record minus the two overhead bytes. Fourteen is then subtracted from that result to get the starting address of the last logical record of the physical record.

IORTN - KS

Loads address of CCBIN into register 1, executes a channel program, and returns via register 2.

LOADSYM - KM

Loads the symbol address that was calculated in the symbol search routine into register 2 and tests to see if there is a plus or minus symbol adjustment for the symbol address.

LOOPTPA - KG

Clears register 2 so that the next character to be converted to a printable character can be inserted into register 2 for conversion.

MINUS1 - KR

Turns on RLFSWI (X'40') switch to indicate that the relocation factor for the control section is negative.

MINUS2 - KR

Turns on RLFSWI (X'20') switch to indicate that the relocation factor for the further qualified CSECT is negative.

MISPCC - KB

This routine is entered when a job control card is read before the PCC card. It moves the error message, 'PCC MISSING', to the print area and writes the card image on SYSLST.

MODIFCS - KR

Decreases register 4 (the input area pointer) by 16 to point to the next logical record. It updates the CSECT counter by one and tests for the end of the physical record to see if another record must be read.

MODIFPH - KQ

This routine decreases the disk input area pointer by 16 to point to the next logical record of the physical record. It tests to see if this is the last logical record of the physical record. If it is not the last record, it branches to TESTPH. If it is the last logical record of the physical record, it tests for end of the control dictionary, and either branches to read another record or branches to GOABORT.

MOVE - KF

MOVECS - KR

This routine is entered after the correct CSECT in the correct phase has been found in the control dictionary. It tests the further qualified CSECT switch, which is always set to NO-OP in this phase. It then retrieves the control section name, relocation factor and the sign of the relocation factor from the control dictionary record before returning via link register 10.

MOVEPH - KP

This routine is entered when the phase name in the phase directory record is the same as the phase name specified in the PHQ card or the test-request card. It stores the start address of the phase, the number of physical records in the phase, and the length of the last record. It then calculates the end address of the phase.

MVASTER - KH

This routine is entered when the second character of patch indicator (*) parameter on the autopatch control card contains a character less than zero. It saves the first character of the parameter and branches to test the second character for blank or comma.

MVCODE - KN

Moves error code from save area to error table. It updates the error table pointer by two and stores the new error table address.

NOCDTAPE - KC

This routine is executed if a PHQ card has been processed and there is no control section qualification (CSQ) card. It turns on the assumed CSECT switch and branches to CDTAPE.

ODDNR - KJ

Stores the length of the constants punched in the CON card, tests to see if there is an odd number of characters, and branches to an error routine if the character count is odd. The count should be even because the constants are punched in the card in hexadecimal format which requires 2 card columns per character.

PHQRTN - KP

This routine reads phase directory records from SYSRES until the phase record is found that contains the same name as the one specified in the phase qualification (PHQ) card. When the correct phase is found, it calculates the length of the phase and the phase start and end addresses.

PHSFOUND - KQ

This routine stores the phase number in the save area, PHSNUMB, and initializes the CSECT counter to one. Because of the limited length of the control dictionary record, the four high order bits of the phase number are stored in the four high order bits of byte 8 of the control dictionary record. The eight low order bits of the phase number are stored in byte 12 of the control dictionary record.

PHSTAPE - KE

This routine is entered when a PHQ card is read. It tests for a phase previously processed. If a phase has been processed, this routine stores the new phase name and turns off the CSQ-processed and phase-previously-processed switches. It branches to test again if the phase was previously processed. This time the switch is off. It is turned on, and a check for the assumed phase switch being on is made. If it is on, it processes the PHQ card and writes it on SYSLST. If the assumed phase switch is not on, a branch to CHKTPA is made.

PHSTAPE1 - KC

If a test-request card has been read and no PHQ card has been processed, it turns on the assumed phase switch.

PHTRRTN - KS

PNTID - KD

Subtracts one from the current SVC number to get the new SVC number to be used for the Autotest table entry.

PNTPAT - KD

Writes the SVC number and ADD card image on SYSLST.

POINTER - KC

Adds the core image library (CIL) input area address to the patch-point displacement within the CIL record. This points to the patch-point address in the core image library. This address is then tested for a BAL, BALR, or SVC instruction.

PQEXIT - KR

Restores registers 2 through 15 before exiting from the CSQ routine.

PRINTPAT - KK

This routine writes a line on SYSLST and updates the line counter. It then checks for a full page. If the page is full, it skips to the beginning of the next page and writes the header line on SYSLST.

RDSYM - KM

Searches the SYM records for the symbol address.

READCARD - KB

Reads a card from SYSIPT.

READCSQ - KQ

Clears register three (the record count register) which contains the number of logical records in the physical record to be processed. It then reads a physical record.

READPAT - KK

Reads a card from SYSIPT.

READPHQ - KP

Beginning of routine to read a phase directory record from SYSRES and to update the disk address for the next record.

READREC - KV

READRTN - KV

Subroutine which is used to read records from the core image library.

RTNE2 - KU
Subroutine to write Autotest table record on SYSLNK.

RTRNID - KD
Tests to see if SVC 53 has been reached. If it has, it branches to an error routine because SVC 54 is the lowest SVC number that can be used by Autotest.

SECNDPHS - KE
Stores the phase name and resets the CSQ-processed and the phase-previously-processed switches.

SKPCON - KC
The routine checks to see if the error switch is turned on and branches to an error routine if it is. If no errors have occurred, it branches to get the core image library record that contains the patch-point address. It then tests again for an error condition.

STDEC - KJ
This routine is entered after a valid hexadecimal character between 0 and F has been found in the parameter following the patch indicator (*) on the autopatch card. It tests the constant card switch to see if a CON card is being processed. If it is, it updates the card area pointer and character counter and branches to check another character. If the constant card switch is off, the valid hexadecimal character is stored in register 8, before updating the card pointer and character counter.

STORETPA - KG
Moves the patch-point address from register 2 into temporary storage and tests to see that the patch-point address is on a halfword boundary. If it is not on a halfword boundary, a branch is made to an error routine.

STPADR - KG
Stores a printable character from the patch-point address in the print area field, PPA.

SUBTRLF - KF

SYMCOM - KL
Beginning of an error routine that is entered if nine characters of a symbol have been checked and there was not a comma or a blank column found to indicate the end of the parameter.

SYMER - KL
Error routine for indicating an invalid symbolic address.

SYMERTN - KL
Loads register 7 with the return address of ENDSYM which is entered after writing on SYSLST the message to indicate an error in a symbolic address has been found.

SYMRTN - KL
This routine is entered when it is determined that the address being tested is not hexadecimal. This routine tests to see if the first character of the address is a decimal digit by "testing under mask" the first four bits of the byte for all ones. If it is a decimal digit, it branches to a decimal routine. Otherwise, it is assumed that it must be a symbolic address and it processes the address as a symbol.

SYMRTN1 - KL
Loads the address of the work area, SAVE3, into register 3 and sets register 4 to the maximum length of the symbolic address plus the comma delimiter.

SYMSRCH - KS
This routine is entered to search the SYM file records for the symbol address (displacement of symbol within the phase).

TESTCS - KR
This routine is entered after the correct phase name has been retrieved from the control dictionary. The control dictionary is re-read from the beginning. Each CSECT record is tested for the same phase number as the one in the phase record. When the correct phase number is found, a branch is taken to check for the correct CSECT within the phase.

TESTPH - KQ
Tests to see if the logical record is a phase record and if it is the same phase name as the phase which contains the CSECT being processed. If both conditions are present, a branch to PHSFOUND is made. If either condition is absent, a branch is taken to MODIFPH to check the next logical record.

TESTWI - KQ
This routine is executed if there has not been a further-qualified CSECT processed. It tests the assumed-control-section switch and branches to set the program switch, ASSUMCS, to either NO-OP or an unconditional branch depending on the setting of the assumed CSQ switch.

TPER - KL
Moves error code, A, into the MVCODE save area and processes the error, indicating an invalid patch-point address.

TRANS - KJ
This routine is executed when the end of the autopatch add or exchange instruction parameter or replace constant parameter has been found. It tests the CON card switch and, if the switch is on, it branches to ODDNR, otherwise it translates and packs the ADD or EXC instruction. It then moves a value of 12 into register 3 and compares this value to register 4 which contains the character length of the instruction. If they are not equal, register 3 is indexed to the value of 8 and it is compared to register 4. If not equal, register 3 is set to the value of 4 and it is again compared to register 4. If it is not equal, an error routine is entered. If any of the above comparisons result in an equal compare, a branch is taken to CHRCNT.

TRANSHEX - KF
This routine is executed after the patch-point address has been tested and found to be a valid hexadecimal address. It translates and packs the hexadecimal address and adds to or subtracts the relocation factor from the patch-point address and branches to see if it is within the user's phase.

TRMOVE - KR

TRMOVEPH - KP
Moves start and end addresses of the further qualified phase to the input save area.

TRSWI - KB
This routine is entered when a test-request card has been identified. It resets the diagnostic phase switch that indicates that ATLEFC2 is in main storage, and fetches ATLEFC3 to process the test-request card.

TRSWICH - KQ
This label is normally an unconditional branch to TESTWI and is not changed during this phase.

TSTCOM - KF
This routine is entered if, while testing a hexadecimal field, an invalid hexadecimal character is found and the character is not a quote. A test is made for a comma. If a comma is found, a branch is taken to the error routine because in the patch-point hexadecimal address field a quote must precede a comma. If a comma is not found, a test for a blank column is made.

TSTDLMT - KF
This routine is entered when the hexadecimal patch-point address contains a character other than a hexadecimal character from 0 to F and it is not a quote or a comma, or the character is a quote and the following column does not contain a comma. It updates the card area pointer and tests for the end of card or for a comma.

UPREG - KJ

VALIDHEX - KJ
The entry point into a routine that tests the validity of the hexadecimal character in the patch-point address parameter of the test-request control card.

WRITERTN - KV
Subroutine that writes core image library records on SYSRES.

Test Request Diagnostics (ATLEFC3) Charts LA-LU

ABRTR - LN
Tests for a multiple test-request card after an error was discovered in the test-request function parameter. If the card is not followed by a multiple test-request card, the error print routine is entered. If there is a multiple test-request card following the error card, the error card is written on SYSLST and another card read.

ADDCSQ - LG
Subroutine that adds to or subracts from the further-qualified control-section-adjustment factor the address of the symbol and tests to see if the address is within the further qualified control section.

ADDM - LM
Subtracts the minus symbol adjustment factor from the symbol address.

ADDP - LM
Adds the plus symbol adjustment factor to the symbol address.

ADDSWI - LA

ADD240 - LG
Adds F0 to the character to be converted to make a valid hexadecimal character of 0 through 9.

ADINPUT - LM
If a further qualified control section is being processed, this routine adds the start address of the further qualified user's phase to the symbolic address.

ADJRTN - LJ
This subroutine is entered whenever a plus or minus sign is found in a symbolic address. It indicates that the symbol has an adjustment factor. This adjustment factor can be up to 4 decimal characters long. This routine clears the work registers and work area and initializes the loop counter.

ADJSTM - LF
This routine is entered when a minus sign has been found in the symbolic address of a test-point address or start or end address parameter. It sets a switch to indicate that the adjustment factor of the symbolic address is minus.

ADJSTP - LF
This routine is entered when a plus sign has been found in the symbolic address of a test-point, start or end address parameter. It sets a switch to indicate that the adjustment factor of the symbolic address is plus.

ASMSYM - LF
This routine turns on the optional feature switch and loads register 10 with the return address of ASSUMED.

The optional feature switch is turned on when a blank column is found after the display end address or the secondary ON parameters of the test-request card. It indicates that the assumed format for the display and/or panel be used (hexadecimal display and general register panel).

ASSUMED - LH
This routine fetches ATLEFC4 and branches to DSFSMD in ATLEFC4 to test for display or panel present. It is entered, when a blank column is found after the end of a test point, start or end address parameter, and that parameter is processed.

ASSUMCS - LR
This label is a branch switch which is normally set to an unconditional branch to MOVECS. It is set to NO-OP whenever a CSECT name has been supplied by the user.

ASSUMPH - LQ
This routine is entered if the assumed phase switch is on when processing the CSQ card. It sets up the CCW and reads a physical record. It calculates the main storage address of the last logical record of the physical record, and tests to be sure that it is a phase record. It saves the phase name and turns off the assumed phase and CSECT processed switches, loads the return address of CHKPTA into register 5 and branches to PHQRTN.

BACKDEC - LU

BBACK - LT
This label is a branch switch which is normally set to an unconditional branch. It is changed to a NO-OP whenever a further qualified phase is processed.

BLER - LD
Loads the address of ABRTR into register 10 as a return point after it has been determined which parameter (either test-point, start or end address) caused this error routine to be executed.

BTRPH - LT
This label is a branch switch which is normally a NO-OP. It is set to an unconditional branch whenever a further qualified phase has been processed. If the switch is set to NO-OP, the end address of the phase, the phase name, the number of records in the phase, the length of the last record, and the disk address of the phase in the core image library are stored. If it is set to an unconditional branch, it branches to TRMOVEPH.

CALADJ - LJ
This routine is entered when a comma is found at the end of a symbolic address adjustment. It packs and converts to binary the decimal adjustment value and stores it in ADJCOUNT. It then branches to process the symbol.

CALADR - LM
Adds the start address of the user's phase or the start address of the user's further qualified phase (depending on whether a further qualified phase is being processed or not) to the address of the symbol.

CARDID1 - LA
Initializes the error table address in the print area to point to the first byte of the error table.

CARDID10 - LA
Turns off X'80' and X'40' bits in PHASESWI to indicate ATLEFC2 is no longer in main storage and that there is not an error in the test-request card. It also turns off SWITCH, the test-request-parameter switch, multiple-test-request-card switch, and initializes register 5 with the address of the card input area.

CARDID12 - LH
When Autotest fails to recognize a card as a valid test-request card, it writes an error message on SYSLST and reads another card. This routine is then executed. It tests to see if the multiple test-request switch is on. If it is on, a branch is taken to fetch ATLEFC4 and process the correct test-request card(s) for that test-request number. If it is off, it branches to initialize switches and identify the new card.

CARDID2 - LH
Tests to see if the test-point address (TPA) of the last test-request card was processed correctly. If it was, it tests to see if this is a multiple test-request card. If it is, it branches to fetch ATLEFC4 to process the correct TPA. If either the TPA was not processed correctly or the multiple test-request switch was on, a branch is taken to identify the card.

CDSQCARD - LM
This routine is entered if there is an assumed further phase qualification or after the assumed further phase qualification has been processed and there is an assumed CSECT. It loads the loop counter with the maximum length of the further qualified CSECT and checks for a period following a blank column which indicates a private code. It then tests to see if phase qualification has been processed. If it has, it branches to process further CSECT qualification. If not, it processes further phase qualification.

CDSWI - LE

CDTAPE - LE
This routine is entered when a CSQ card has been read. It processes the control section qualification card, writes it on SYSLST, and branches to read another card.

CDTAPE1 - LK
Tests to see if further phase qualification has been processed. If it has been processed, it stores the original phase relocation factor and the original CSECT number and processes further CSECT qualification. If further phase qualification has not been processed, it processes it.

CHKADJ - LJ
Checks the decimal digit characters of the symbol address adjustment factor for a comma, a blank, or a valid decimal character, and moves the valid character into register 3. If a comma is found, it indicates the end of the parameter and it branches to convert the adjustment factor to binary. If a blank column is detected, it also indicates the end of the parameter, but the optional feature switch is turned on to indicate that the assumed format is to be used for the display and panel. The adjustment factor is then converted to binary.

CHKCARD - LA
Tests for the type of card read from SYSIPT and either branches to the correct routine or fetches the correct phase to process the card.

CHKCST1 - LK
This routine is entered to process the further qualified control section parameters. It stores the original phase relocation factor and the original CSECT number before processing the further qualified CSECT.

CHKDCLH - LB
This routine is entered after it is determined that the hexadecimal character being tested is greater than zero. It tests to see if the hexadecimal character is less than 9. If it is, it is a valid hexadecimal character and another character is tested. If the character is greater than 9, the character is then tested to see if it is a quote.

CHKFNCT - LN
This routine is entered if there is a comma at the end of the test-request function parameter. It tests the card to see which type of test request is being requested, and it turns on indicators in the test-request control and Autotest table records. If there was not a valid test-request function, a branch is taken to FNCTCOMA which will enter into an error routine.

CHKHEXH - LB
This routine is executed after it has been determined that the hexadecimal character being checked is less than zero, but greater than A. It tests to see if the character is less than F, thus making it a valid hexadecimal character.

CHKLMT - LC
Tests for a comma after a quote is found, indicating the end of a hexadecimal field. If a comma is found, the hexadecimal address is translated and packed. If a comma is not found, a test is made for a blank column in the card.

CHKPARTR - LD
When this routine is entered, it has been determined that an error has occurred in either the start or end address of the display parameter. It tests the start address-being-processed switch. If it is on, the error routine to indicate an error in the start address is executed. If the switch is off, it assumes that the error must be in the display end address parameter.

CHKQUOT - LC
This routine is entered when the hexadecimal character is determined to be other than a character between 0 and F. It tests to see if the character is a quote. If it is not a quote, a branch is taken to check to see if it is a comma. If the character was a quote, a branch is taken to test the next card column.

CHKSWI - LR
Tests to see if symbols are being processed and branches to either the symbol routine or the no-phase-found-error routine.

CHKSWI2 - LS
Stores registers 2 through 15 and branches to an error routine if the bit of the CCB was set on after a record was read.

CHKSYM - LF
This routine checks each character of the symbol address for the following conditions: end of card, a period to indicate a further qualification, a plus sign to indicate a positive adjustment factor for the symbol address, a minus sign to indicate a minus adjustment factor for the symbol address, and a comma to indicate the end of a parameter. If none of these are found, it stores the alphabetic character and gets the next character to be checked.

CHKSZ - LC
This routine checks to ensure that the calculated test-point address, start or end address is within the limits of the current phase.

CHKSZCSQ - LG
Checks the symbol address to see if it is within the further qualified CSECT. If it is not, it branches to an error routine. If the address is within the further qualified CSECT, it branches to CLRTAPE.

CHKTP - LH
This routine is entered after a card is read and an error occurred in the previous card. The card read was not a test-request card indicating that it may be a multiple test-request card. This routine checks to see if the test-point address of the last card was processed correctly. If it was, it can be used with this multiple test-request card.

CHKTPA - LB
Checks the test-request card switch. If it is off, it turns on the switch to indicate ATLEFC2 is in main storage and fetches ATLEFC2 to test to see what type of card this is. If the switch is on, it tests to see if a phase card or CSQ card has been processed. If neither one has been processed, it turns on the assumed phase or the assumed CSQ card switch.

CHK711 - LL
This routine is entered if there is a blank column following the test-request function parameter of the test-request card, and there was an error in the test-request card. It tests to see if this is a multiple test-request card. If it is not, it is an error condition because the parameters are not complete. If it is a multiple test-request card, the card image is written on SYSLST and another card is read.

CLCRDSWI - LE
Turns off the CSQ assumed switch and the CSQ card switch.

CLRBLK - LF

CLRCSQ - LK
This routine is entered after the further qualified CSECT has been processed. It turns off the switches used while processing the further qualified phase and control section. It tests to see if the start address is being processed. If it is not, it branches to process the end address. If the start address is being processed, it branches to LOWLTS.

CLRTAPE - LG
This routine checks to see if there is an error in the further qualified control section symbol address. If there is, it branches to an error routine. If not, it branches to see which parameter is being processed and stores that parameter.

CLSFILE - LH
Turns off CSQ card-processed switch, further-qualified-phase-processed switch, and further-qualified-control-section-processed switch before entering the routine to identify the test-request card just read.

CLSRTRN - LL
Restores the error table address.

CNCLTR - LH
This routine fetches ATLEFC7 and branches to label ABORT7 which moves error code Z to print area, indicating an error in phase or CSECT qualification.

CNCLTR1 - LD
This routine turns off error-in-test-request-card switch, moves the error message "Card Ignored" to the print area, writes the card image on SYSLST, reads a card, and tests to see if it has ./ in the first two columns. If it has, it branches to test to see if it is a continuation card. If it does not have ./ in the first two columns, it tests to see if the test-point address of the previous card has been processed correctly so it can be used for this continuation card.

CNCLTR10 - LH
This is an error routine that is entered when a ./ is found in the first two columns of a card that was indicated to be a multiple test-request card by the previous card.

This routine checks to see if the Autotest table entry for this test request has been put into the Autotest table. If it has not, this card is checked to see which type of test-request card it is. If the Autotest table entry has been started, it fetches ATLEFC4 to process the correct test-request cards associated with this Autotest table entry.

COMPAR - LT
Tests to see if the phase directory record contains the same name as the phase qualified by the PHQ card or further qualified by the test-request card. If not, it updates the input area pointer to the next logical record, tests for end-of-file and end-of-physical record. It reads either a new physical record, if required, or a new logical record. If the phase names are equal, a branch is taken to MOVEPH.

CONSWI - LA

CONTCARD - LP
Subroutine that tests for a blank column in the card after the end of a parameter has been found. If a blank column is found, it branches to CONTRTN. If the column is not blank, it returns in-line to continue processing.

CONTRTN - LP
Tests for the multiple-test-request-card switch on. If on, it writes the card image on SYSLST, reads another card, tests to see if the new card is a multiple test-request card, and returns to processing. If it is not a multiple test-request card, it branches to an error routine because the last card indicated this new card would be a multiple test-request card.

CONVTPA - LG
Converts the test-point address to printable characters.

CSQRTN - LQ
Entry point into a subroutine that processes the CSQ card. Moves the phase and CSECT name to a save area. Stores registers 2 through 15 and initializes the CCW to read a control dictionary physical record, starting at the highest disk address in the control dictionary. This is done because the control dictionary is built backwards in main storage starting at the effective end-of-core address. When written on disk, the last control dictionary record was written first at the lowest disk address. When retrieving these records, the last record written on the disk will contain the first control dictionary record. A test is made to see if the assumed phase switch is on. If on, branch to ASSUMPH. If not, branch to READCSQ.

CSQ1 - LH

CSTRRTN - LS
This routine is entered whenever there is a further qualified CSECT in the test-request card. The phase name and the CSECT name are moved to a save area, the test-request switch and MOVECS branch switches initialized, and a branch made to the CSQRTN routine to find the correct CSECT from the control dictionary library.

DECPACK - LC
This routine is entered if the decimal parameter switch is on. The switch indicates that the parameter being processed is decimal. It packs the decimal parameter and converts it to binary.

DECRTN - LJ
This routine is entered when a decimal digit is found in the first column of a parameter. It moves up to six decimal digits into registers 2 and 3. When the end of the parameter is reached, it tests to see if the test-point address is being processed. If it is, this is an error, because the test-point address must always be hexadecimal or symbolic. If it is not the test-point address, the decimal address is processed.

DECRTN1 - LJ
This routine tests for end-of-card, comma, and a valid decimal digit. It moves the valid decimal digit into registers 2 and 3. It moves one character at a time until the end-of-card, a comma, or 6 valid characters have been moved.

DSKDECR - LU
This routine decreases the disk address by one each time a physical record from the control dictionary is read. The record number is decreased by one and a test is made to see if the lower limit of the record number has been reached. If so, the head number is decreased by one and the high limit of the record number is moved into the disk address. The low limit of the head number is checked and if it has been reached, the cylinder number is decreased by one and the high limit of the head number is moved into the disk address. A test is then made for cylinder overflow and the cylinder number is decreased, if required. If the minimum value is not reached for the record, head, or cylinder, a branch is taken to return via register 2.

DSKINCR - LU
This routine increases the disk address by one each time a physical record is read. The record number is increased by one and a test is made to see if it is the last record of the track. If it is, the head number is increased by one and the low limit of the track is moved into the disk address. A test is then made to see if the maximum number of heads has been reached. If it has, the cylinder number is updated by one and the low limit head number is moved into the disk address. A test is then made for the cylinder overflow. If the maximum value for the record, head, or cylinder has not been reached when it is tested, a branch is taken to return via register 2.

DSPMNT - LM
Tests to see if the symbol adjustment factor is plus or minus. The appropriate adjustment factor is added to or subtracted from the symbol address. This gives the absolute address of the symbol in main storage.

ENDSYM - LM
This routine is entered after the absolute address of the symbol is calculated. It clears work areas and resets symbol adjustment plus and minus switches and the optional feature switch. The routine next checks to see if further qualification of the CSECT is being processed. If it is, a branch is taken to see if the symbol is within the limits of the further qualified CSECT. If the CSECT is not being processed, a test is made for a symbol error and a branch is taken to either process the symbol error or test to see that the calculated symbol address is within the phase specified.

ENDTR - LL
This routine is entered when a comma is found as the first character of the operand following the display end address. It updates the card area pointer to the next column of the card.

ERFNCT - LN
Moves the error code N to the print area to indicate that the function parameter is other than a PNL, DSP, or DPL.

ERLOW - LD
Error routine that is entered if there is an error in the display start address on the test-request card.

ERPRINT - LL
Error routine that writes the card image on SYSLST and tests to see if the test-point address was processed correctly. If not, it branches to OFF. If the test-point address was processed correctly, the multiple test-request switch is tested. If the switch is on, a branch is taken to fetch ATLEFC4 to complete the Autotest table entry for this test-request.

ERRTN1 - LP
Loads the address of the last error code into register 2 and tests to see if six error table entries have been made. If so, it indicates the error table is full and a branch is taken to ABRTR to test for a multiple test-request card. If the error table is not full, the error-in-test-request-card switch is turned on and a branch is taken to MVCODE.

ERTPA - LD
This routine is entered when an error is discovered in the test-point address, display start address or end address parameter. This routine tests to see if the error occurred while the test-point address was being processed.

ERUP - LD
Error routine that is entered if there is an error in the display end address.

EXCSWI - LA

EXITOPTN - LD

E1CMP - LS
It compares the qualified CSECT number with the CSECT number on the symbol record. If they are equal, it compares to see if the symbol in the record is the same symbol being analyzed. If it is, the symbol displacement is moved from the symbol record to a save area, E1DISP+1. If the CSECT numbers do not agree, a branch is taken to E1PT1.

E1PT1 - LS
Updates the input area pointer to the next symbol logical record of the physical record, tests for the end of the physical record, and branches to read a new physical record or logical record.

E1RD - LS
This routine moves the disk address from E1ADDR to a hold area. This address will be used by the CCW to read a physical record. The disk address is then increased and the new address is moved into E1ADDR.

E1RD1 - LS
Initializes register 1 with the address of the CCB and reads a 196-byte symbol record. It tests for I/O complete and incorrect record length. It then calculates the end address of the physical record and pads 15 F's behind the last byte to indicate the end of the physical record.

FCHWRNG - LH
This routine is executed when an error in one of the multiple test-request cards is detected. It fetches phase ATLEFC4 to build the test-request table entry for this SVC.

FILEEND - LR
Tests for end of control dictionary file. If it is not the end-of-file, it branches to read another physical record. If end-of-file is found, it indicates the phase number has not been found and the error routine, GOABORT, is executed.

FINDCOMA - LC
Updates card pointer after a quote is found in a hexadecimal field.

FIRSTCS - LQ
Entered if there is an assumed CSECT, indicating that a CSECT name was not supplied by the user.

FNCTCOMA - LN
This routine is executed if there is no comma found at the end of the test-request function parameters and the test-request function is not a panel only. If the test-request function was invalid, it finds the end of the parameter and then enters an error routine to indicate an invalid test-request function parameter was specified.

FNCTL - LN
This routine checks the test-request function parameter, turns on the indicators in the test-request control and Autotest table records for the test-request function desired. The functions can only be DSP for display, PNL for panel or DPL for display and panel.

FTCHPCC - LA
Fetches ATLEFC7 to write the last Autotest table record on SYSLNK. ATLEFC7 then fetches ATLEFC5 to process the PCC card.

GOABORT - LR
This routine is entered if the end of the control dictionary is reached before the correct phase number is found. It initializes the branch switches BTRPH, BBACK, TRSWITCH and MOVECS before entering the CHKSWI routine.

GOBACK - LT
Restores registers 2 through 15 and resets the branch switches (BBACK and BTRPH) to their normal positions before returning to inline processing.

HEADLINE - LP
Moves the header line into the print area. Sets up for a skip to beginning of page before the header line is printed.

HEXRTN - LB
Beginning of loop to test one character at a time to determine if it is a valid hexadecimal character between 0 and F. If it is, it stores that character in register 3.

HEXRTN1 - LB
This routine is executed whenever it is determined that the test-point address is in hexadecimal characters. It clears work registers and turns off the symbols-being-processed switch (TPASWI, X'40').

HEXRTN2 - LB
Beginning of subroutine to determine if the parameters on the test-request card are in hexadecimal format. If they are in hexadecimal, it branches to HEXRTN1. If not, it branches to SYMRT to determine if it is in decimal or symbolic format.

ILPEXT - LU

INICS - LQ
This routine is entered after the correct phase has been found. It initializes the CCW to the starting address of the control dictionary and branches to read the first physical record again, looking for the correct control section.

INTFNCT - LH
This routine is entered if the card read is a multiple test-request card requesting a display and/or panel. It turns on a switch to indicate ON parameters are being processed, and checks to see if columns 1-15 are blank. If they are blank, it branches to TSTFNCT. If they are not blank, it branches to the error routine, CNCLTR10.

IO - LQ
This routine decreases the address of the record just read by one to get the address of the next physical record to be read. It then moves the second byte of the physical record into register 3. In this byte, bits 0 through 6 contain the number of logical records in the physical record and bit 7 is the last physical record indicator. Register 3 is shifted one bit to the right to remove the last record indicator. The number of logical records in the physical record is then multiplied by 16 to calculate the length of the physical record minus the two overhead bytes. The start address of the I/O area is added to the calculated length of the physical record minus the two overhead bytes. Fourteen is then subtracted from that result to get the starting address of the last logical record of the physical record.

IORTN - LT
Loads address of CCBIN into register 1, executes a channel program, and returns via register 2.

LOADNXT - LL
Tests to determine if further control section qualification has been processed. If it has been, it processes the display end address. If not, it updates the card area pointer to point beyond the second period in the test-request card and then processes the display end address.

LOOPTPA - LG
Clears register 2 so that the next character to be converted to a printable character can be inserted into register 2 for conversion.

LOWLTS - LK
This routine tests to see if a display is requested. If it is not, it branches to check the panel ON parameters. If a display is requested, it processes the start address of the display and then branches to process the display end address.

LSTWRNG - LH
Since the last card was an invalid multiple test-request card, an error-in-the-multiple-test-request-card switch is turned on. It then branches to FCHWRNG to fetch ATLEFC4 to process the correct test-request card for that test-request number.

MINUS1 - LR
Turns on RLFSWI (X'40') switch to indicate that the relocation factor for the control section is negative.

MINUS2 - LR
Turns on RLFSWI (X'20') switch to indicate that the relocation factor for the further qualified CSECT is negative.

MISON - LE
This routine is entered after a comma was found in the first position of the primary ON parameter field, indicating the ON parameters are missing.

MISPCC - LA
This routine is entered when a job control card is read before the PCC card. It moves the error code, U, to the print area and writes the card image on SYSLST.

MODIFCS - LR
Decreases register 4 (the input area pointer) by 16 to point to the next logical record. It updates the CSECT counter by one and tests for the end of the physical record to see if another record must be read.

MODIFPH - LQ
This routine decreases the disk input area pointer by 16 to point to the next logical record of the physical record. It tests to see if this is the last logical record of the physical record. If it is not the last record, it branches to TESTPH. If it is the last logical record of the physical record, it tests for end of the control dictionary, and either branches to read another record or branches to GOABORT.

MOVE - LB
This routine is entered when a valid hexadecimal character is found in a hexadecimal address. It shifts the bytes in register 2 and 3 one byte to the left and inserts the valid hexadecimal character into the units position of register 3. It then gets the next character to be checked from the card.

MOVECS - LR
This routine is entered after the correct CSECT in the correct phase has been found in the control dictionary. It tests to see if the further qualified CSECT switch is on, and branches to TRMOVE if it is. If the further qualified CSECT switch is off, it retrieves the control section name, relocation factor and sign of relocation factor, from the control dictionary record before returning via link register 10.

MOVEPH - LT
This routine is entered when the phase name in the phase directory record is the same as the phase name specified in the PHQ card or the test-request card. It stores the start address of the phase, the number of physical records in the phase, and the length of the last record. It then calculates the end address of the phase.

MTRFNCT - LH
This routine is entered when a card is read that is not a test-request card. It is assumed that it is a multiple test-request card. It tests card columns 16-18 to see if it is a display and/or panel. If it is not, a branch is taken to an error routine.

MTRCONT - LL

MVCODE - LP
Moves error code from save area to error table. It updates the error table pointer by two and stores the new error table address.

NOCDTAPE - LB
This routine is executed if a PHQ card has been processed and there is no CSQ card. It turns on the assumed-CSECT switch and branches to CDTAPE.

NOCSTQ - LK
This routine is entered, if at the end of the further-qualified-phase parameter, there are two periods in adjacent columns. This indicates that there is no further qualified CSECT. Therefore the CSECT number of the further qualified phase will be the same as that for the original phase qualification. It turns on the assumed-further-qualified-CSQ switch, and updates the card area pointer.

NOPHASE - LR
This routine is entered when the end of control dictionary is reached and the phase number has not been found. It turns on the error switch that will eventually cause the job to be canceled. It turns off the processing-phase-qualification switch and then branches to CNCLTR.

NOSYMBOL - LF

NOTPA - LM
This routine is entered if the symbolic address being processed is not the test-point address. It stores the relocatable symbolic address in FNCTAREA.

OFF - LL
Turns off the test-request error switch, clears the work area, restores the error table address, turns off the multiple test-request switch and branches to RDCARD.

OPTNADJ - LJ
This routine turns on the optional-feature switch, whenever a blank column is detected at the end of the symbolic address adjustment factor. This switch, when on, indicates that the assumed ON parameters will be used and that the assumed general register panel will be given, if a panel was requested.

OPTPNL1 - LL
This routine is executed if there is a blank column following a panel test-request function operand in the test-request card.

If there is not an error in the test-request card, it fetches ATLEFC4 to process the assumed panel parameter. If there is an error in the test-request card, it branches to CHK711.

PCHDGNTC - LB
Turns on PHASESWI to indicate that phase ATLEFC2 is in main storage. It fetches ATLEFC2 and branches to CARDID in ATLEFC2 to test to see what type of test-request card this is.

PHQRTN - LT
This routine reads phase directory records from SYSRES until the phase record is found that contains the same name as the one specified in the phase qualification card or the phase further qualified in the test-request card. When the correct phase is found, it calculates the length of the phase and the phase start and end addresses.

PHSFOUND - LQ
This routine stores the phase number in the save area, PHSNUMB, and initializes the CSECT counter to one.

Because of the limited length of the control dictionary record, the four high-order bits of the phase number are stored in the four high-order bits of byte 8 of the control dictionary record. The eight low order bits of the phase number are stored in byte 12 of the control dictionary record.

PHSTAPE - LE
This routine is entered when a PHQ card is read. It tests for a phase previously processed. If it has been, it stores the new phase name and turns off the CSQ-processed and phase-previously-processed switches. It branches to test again if the phase was previously processed. This time the switch is off. It is turned on, and a check for the assumed phase switch being on is made. If it is on, it processes the PHQ card and writes it on SYSLST. If the assumed phase switch is not on, a branch to CHKTPA is made.

PHSTAPE1 - LB
Turns on the assumed phase switch, if a test-request card has been read and no PHQ card has been processed.

PHTRRTN - LS
This routine is entered whenever there is a further qualified phase specified in the test-request card. It saves the phase name, initializes branch switches and branches to the PHQRTN routine to find the further qualified phase in the control dictionary library.

PNLON1 - LL
Tests for blank column following the test-request function (panel, display or panel and display) operand of the test-request card. If the column is not blank, it turns on the secondary ON parameter-present switch and fetches ATLEFC4 to process the ON parameters. If the column is blank, it branches to OPTPNL1.

PNLON2 - LL
Tests for a test-request error. If there is an error, it branches to CHK711. If there is no error, it fetches ATLEFC4 to process the assumed panel parameter.

PNTHD - LP
This routine is executed after the maximum number of lines for a page have been written on SYSLST. It restores the line counter to zero.

PQEXIT - LR

PRINTPAT - LP
This routine writes a line on SYSLST and updates the line counter. It then checks for a full page. If the page is full, it skips to the beginning of the next page and writes the header line on SYSLST.

PRVTCD - LM
This routine is entered when a period is found following a blank column, which, in turn, followed the period for the further qualified phase. This indicates that the CSECT does not have a number. The first CSECT in the phase will therefore be used. This routine updates the card area pointer by two to point it to the start address parameter.

QUALFD - LK
Beginning of subroutine entered when a period is found in a test-request card. The period indicates that there is further qualification to either the phase or the control section. This indicates that the display will be from a phase or CSECT different from that for the test-point address. This routine checks to see if the first period read is the first character of the field. If it is, it indicates that the phase is assumed to be the original phase that was qualified by the PHQ card and that only the CSECT will be further qualified. If the first period was not the first character of the field, a branch is taken to CDTAPE1. The first character of the field is determined by subtracting 9 (the maximum number of characters for the field plus the comma) from the loop counter which was initialized to 9. If the result is zero, it indicates that the first character of the field was the period.

RDCARD - LA
Reads a card from SYSIPT.

RDSYM - LF
This routine is entered after the symbolic address has been checked for length and validity. It checks to see if the symbols are present. If they are, the routine searches the SYM records for the symbol address. If the symbols are not present, an error routine is executed.

READCSQ - LQ
Clears register 3 (the record count register) which contains the number of logical records in the physical record to be processed. It then reads a physical record.

READPHQ - LT
Beginning of routine to read a phase directory record from SYSRES and to update the disk address for the next record.

SECNDPHS - LE
Stores the phase name and resets the CSQ-processed and the phase-previously-processed switches.

SETDPL - LN
Turns on both the display and panel function indicator in the test-request control table record and in the Autotest table record.

SETDSP - LN
Turns on both the display function indicator in the test-request control table record and the Autotest table record.

SETPNL - LN
Turns on the panel function indicator in the Autotest table record, and turns off the display indicator in the Autotest table record.

START - LA
Entry into ATLEFC4. Initializes base registers.

STORELOW - LC
Stores the start address of the display, updates the card pointer to the next parameter, and turns off the switch that indicates the start address is being processed.

STORETPA - LC
Moves the test-point address from register 2 into temporary storage and tests to see that the test-point address is on a halfword boundary. If it is not on a halfword boundary, a branch is made to an error routine.

STOREUP - LC
Stores end address of the display, updates card area pointer to next card column, and then tests to ensure that the end address of the display is not less than the start address of the display.

STPADR - LG
Stores a printable character for the test-request address in the print area.

STSYMADR - LM
Beginning of subroutine that calculates the absolute address of a symbol by adding the symbol displacement to the start address of the user's phase and then adding or subtracting the symbol adjustment factor from the result.

SUBTCSQ - LG

SUBTRLF - LC

SYMCOM - LF
Beginning of an error routine that is entered if nine characters of a symbol have been checked and there was not a comma or a blank column found to indicate the end of the parameter.

SYMER - LF
Error routine for indicating an invalid symbolic address.

SYMERTN - LF
Loads register 7 with the return address of ENDSYM, which is entered after printing the message to indicate an error in a symbolic address.

SYMHGH - LG
This routine is entered if the address called for does not lie within the further qualified CSECT.

SYMRTN - LF
This routine is entered when it is determined that the address being tested is not hexadecimal. This routine tests to see if the first character of the address is a decimal digit by testing under mask the first four bits of the byte for all ones. If it is a decimal digit, it branches to a decimal routine. Otherwise, it is assumed to be a symbolic address and is processed as a symbol.

SYMRTN1 - LF
Loads the address of the work area, SAVE1, into register 3 and sets register 4 to the value 9, the maximum length of the symbolic address plus the comma delimiter.

SYMSRCH - LS
This routine is entered to search the SYM file records for the symbol address (displacement of symbol within the phase).

TESTCS - LR
This routine is entered after the correct phase name has been retrieved from the control dictionary. The control dictionary is re-read starting from the beginning, and each CSECT record is tested for the same phase number as the one taken from the phase record. When the correct phase number is found, a branch is taken to check for the correct CSECT within the phase.

TESTPH - LQ
Tests to see whether the logical record is a phase record and whether it is the same phase name as the phase that contains the CSECT being processed. If both conditions are present, a branch to PHSFOUND is made. If either condition is absent, a branch is taken to MODIFPH to check the next logical record.

TESTWI - LQ
This routine is entered if there has not been a further qualified CSECT processed. It tests the assumed-control-section-qualification switch and branches to set the branch switch, ASSUMCS, to either NO-OP or unconditional branch depending on the setting of the assumed-CSQ switch.

TPER - LD
Error routine that is entered if a blank column was found while the test-point address parameter was being processed.

TRANSHEX - LC
Tests the further-control-section-qualification switch and the decimal-parameter switch. If the further-CSQ switch is on, or if the further-CSQ switch and the decimal-parameter switch are off, the hexadecimal address is translated and packed. If the further-CSQ switch is off and the decimal-parameter switch is on, the decimal address is packed and converted to binary.

TRFRMT - LL
Checks for a blank column in the card after it has been determined that there are no ON-condition parameters following the display end address of the test-request card. It then fetches ATLEFC4.

TRMAIN - LE
This routine is entered after the test-point address of the test-request card has been checked. It updates the card area pointer and tests that column to see if there is a comma, which indicates the ON condition is missing. If no comma is found, ATLEFC4 is fetched to process the primary ON-condition parameters.

TRMOVE - LR
This routine is entered if the correct phase number and CSECT has been found and it is a further-qualified CSECT. It sets the ASSUMCS and MOVECS branch switches to NO-OP and TRSWICH to an unconditional branch. It decreases the CSECT number in register 12 by one and stores the CSECT number and the relocation factor. It then turns either on or off the further-qualified-minus-relocation-factor switch, depending on the status of the sign bit of the control dictionary record.

TRMOVEPH - LT
Moves start and end address of the further qualified phase to the input save area.

TRNSHEX - LJ

TRSWI - LA

TRSWICH - LQ
This label is normally an unconditional branch to TESTWI. When a further-qualified control section is processed, the branch is changed to a NO-OP and the assumed-further-qualified-CSECT switch is tested. If the further-qualified-CSECT switch is off, it indicates that a CSECT name was supplied by the user.

TSTCOM - LD
This routine is entered if, while testing a hexadecimal field, an invalid hexadecimal character is found and the character is not a quote. A test is made for a comma. If a comma is found, a branch is taken to check if the test-point address field of the test-request card is being processed. If yes, it is an error condition because in the test-point address field a quote must precede a comma. If a comma is not found, a test for a blank column is made.

TSTCSQ - LL
This routine processes the display end address on the test-request card.

TSTDLMT - LD
This routine is entered when an error has been discovered in one of the parameters. It updates the card area pointer and tests for end-of-card. If it is not the end-of-card, the column is tested for a comma or a blank to determine the return point after the error message has been written on SYSLST. If it is the end-of-card, CNCLTR1 is entered.

TSTENDTR - LL
This routine is entered after the display end address of the test-request card has been processed. It turns off the switch that indicates a further-qualified control section is being processed. It then checks for a blank column following the comma for the display end address. If there is none, it tests for a comma. If no comma is found, it indicates ON parameters are present. ATLEFC4 is then fetched to process the ON parameters.

TSTFNCT - LH
Tests to see if the ON parameters are being processed. If yes, a branch is made to FNCTL to test for type of test request (panel, display, or display and panel).

TRTPA - LC
This routine translates and packs the hexadecimal address and tests the further-control-section-qualification switch. If the switch is on, it branches to see if the further-CSQ relocation factor should be added or subtracted from the further-control-section-qualification address.

TSTPAR - LD
This routine is entered if a quote is found (indicating the end of a hexadecimal parameter), and the next column of the card is blank. It tests to see which address is being processed; test-point address, display start or end address. The test-point address and the display start address cannot be followed by a blank column. If the display end address has a blank column following it, the assumed values for the secondary ON parameters and format of the display are used.

TSTPNLON - LD

TSTSWI - LC
A routine that tests to see which parameters of the test-request card are being processed; test-point address, display start or end address.

UPREG5 - LN
After the test-request function parameter has been successfully checked and the indicators for the type of function desired are set, this routine updates the card area pointer by 4 to point to the first character of the display start address.

<u>Build Test Request Table Entry (ATLEFC4)</u>
<u>Charts MA-MX</u>

ABRTR - MS
This routine is entered if the error table in the print area is full. It tests for a continuation (multiple test-request) card and branches to CHK71 if it is. If it is not a continuation card, it branches to MTRCONT1 to fetch ATLEFC3 to write the card image on SYSLST and to read another card.

ASSUMED1 - MB

BALCHECK - MM
Certain instructions (BAL, BALR, and SVC) cannot be patched. This routine checks for these instructions and branches to the CILER error routine if they are found at the test-point address in the core image library.

BRTN - ML
If the first character of the panel format parameter is a blank or a comma, it sets the function byte of the test-request control record to indicate a general register panel required.

CALON - MD
This routine tests to see if any of the ON-condition parameters are zero. If any ON-condition parameter is zero, it branches to ERCOND to process the error. If the BEGIN ON parameter is greater than the END ON parameter, a branch to ERCOND is made to process the error. The ON number is then calculated in the following manner:

$$\text{ON number} = \frac{\text{END-BEGIN}}{\text{EVERY}} + 1$$

CHÇKHIGH - MC
This routine is entered after an ON-condition decimal character is found to be zero or greater. It tests to see if the character is nine or less. If it is, it stores the decimal character. If the character is greater than nine, this is an error and the routine branches to test for a right parenthesis to delimit the ON condition.

CHKCONT - MH
This routine is entered to close the test-request table entry. It checks the multiple test-request card switch. If it is on, a branch is taken to write the card image on SYSLST and to read another card. If it is off, it branches to close the test-request table entry. It inserts the following information into the test-request table entry: The saved core image library instruction, the SVC of 255 to indicate the end of the test-request table entry, the length of the test-request table entry, the SVC number, and a zero following the SVC number. It then writes the core image library record back on SYSRES, clears the work area and turns off the switches that indicate ON parameters present and test-request table entry complete. Finally, it branches to FCHID1.

CHKCOUNT - MC
Tests for a character count greater than four for each ON-condition parameter.

CHKDSF - MF
Tests for a blank in the first column of the display format parameter on the test-request card. It branches to HEXFRMT to set the first byte of the test-request table entry to indicate a hexadecimal format for the display.

CHKER - ML

This routine is entered after the error routine, ERPNL. If a blank or comma is found in the panel format parameter, indicating the end of the parameter, this routine is executed. It tests for an error in the test-request card. If there is no error, it returns to processing via link register 10. If an error is found in the test-request card, a branch is taken to NXTREQ to fetch ATLEFC3. The entry point in ATLEFC3 is CHK711.

CHKFLPNT - MF

Tests to see if floating-point hardware is present in the system.

CHKG - ML

This routine tests for the panel format parameter of G for general registers, F for floating-point registers, and S for storage positions 24 to 128. It branches to the appropriate routine to set the first byte of the test-request control record to indicate the type of panel requested. It also tests for a blank column or a comma, indicating the end of the parameter.

CHKNOP - MG

This routine tests to see if the instruction saved from the core image library is greater than two bytes. If it is, a branch is taken to patch NO-OP's into the core image library following the 2-byte SVC instruction inserted by Autotest.

CHKRGHT - MM

This routine is entered if there is no comma following the third (EVERY) parameter of the ON condition. It updates the card column pointer and tests for a right parenthesis. If the right parenthesis is found, it branches to store the third parameter of the ON condition. If it is not found, a branch is taken to RIGHTEND to locate the right parenthesis.

CHKRITE - MD

This routine is entered after the character for the ON condition is determined to be other than a comma or a decimal digit between 0 and 9. It checks to see if the character is a right parenthesis.

CHKSUB - MN

When processing the ON-condition parameter, this routine checks for a comma sub-delimiter following a character count of four.

CHK71 - MA

This routine checks column 72 for a blank, which would indicate there is no multiple test-request card following this card.

CHRFMT - MF

CILER - MM

This routine is entered when an instruction that cannot be patched is detected.

CILPATCH - MS

This routine is entered to patch the SVC number and/or NO-OP's into the core image library. It tests for the end of the core image library record. If it is not found, it moves one character at a time into the core image library area until all characters have been moved. It then returns via register 7 to continue processing. If the end of the core image library is found, it branches to TAPEIOCL.

CILREC - MM

This routine reads from the core image library the record that contains the test-point address specified on the test-request card. It calculates this record number in the following manner: It subtracts the start address of the user's phase from the test-point address to get the displacement of the test-point address within the user's phase. It divides this displacement by the record length of 1728. The quotient plus 1 is the record number, and the remainder is the displacement of the test-point address within the record.

CLRHTAB - MR

Clears the test-request control record area after the record has been written on SYSLNK.

CLRSAVE - MU

Converts to decimal and unpacks the new SVC number or the new test-request number. It tests to see if either an SVC or test-request number is being processed. If a test-request number is being processed, a branch is taken to MVTRNR. If an SVC number is being processed, the new SVC number is moved to the print area and a branch to CONTNR is made.

CLRTAB - MR

After the completed Autotest table record has been written on SYSLNK, this routine clears the test-request table entry area.

CLTRAREA - MH
This routine is entered after the test-request table entry has been completed. It clears the work area, TPA, tests for an error in the multiple test-request card or the test-request card, and branches to fetch ATLEFC3. It enters ATLEFC3 at CARDID10 if there is an error. If there is no error, ATLEFC3 is fetched and entered at label RDCARD.

CNCLTR - MK
If, while processing the ON parameters, column 72 is reached before the end of the ON parameters is found, this routine is executed. It fetches ATLEFC3 to write the message 'CARD IGNORED' on SYSLST.

CNCLTR2 - MT
This routine is executed if one of the following conditions occur:

1. A blank column is found in the first character position of the test-request function parameter.

2. A blank column is found following the end of the test-request function parameter and a multiple test-request card is not being processed.

3. A multiple test-request card, as indicated by the previous card, is not present.

It fetches ATLEFC3 to write the error message, 'CARD IGNORED', on SYSLST.

COMADD - MN
If there are no ON-condition parameters present for this SVC number, but the ON-condition parameters are present in the multiple test-request card, this routine calculates the length of the test-request table entry for the multiple test-request card. It sets the code byte of the test-request table entry to indicate ON-condition parameters are present.

COMPLETE - MQ
This routine is executed if the end of the core image library record is reached before the complete instruction from the core image library record has been saved. After the SVC number has been patched into the current core image library record, the new record is read and the remaining portion of the instruction is saved. This routine patches NO-OP's into the core image library record that contains the remaining bytes of the saved instruction.

CONT - MG

CONTCARD - MT
This routine tests for a blank column in the card after the end of a parameter is found. If a blank column is found, a branch is taken to CONTRTN. If the column is not blank, it returns to continue processing.

CONTNR - MU
This routine tests the job cancel switch. If it is on, it returns to processing via register 7. If the switch is off, it turns on CARDSWI (X'80') to indicate a test-request number is being processed by the PNTID routine. It then moves the new SVC number from SVCNR to a temporary save area. It moves the current test-request number to the SVCNR area so that this area can be used by the PNTID routine to calculate the new test-request number. A branch is then taken to the PNTID routine to calculate the new test-request number.

CONTPRO - MK
Clears the ACODE work area and reads the next card, which should be a multiple test-request card. It tests to see if columns 1 through 15 are blank. If they are not, it indicates this card is not a multiple test-request card and an error routine is entered. If columns 1 through 15 are blank, it updates the card area pointer to column 16, stores the address of the error table, clears the switch labeled SWITCH and fetches ATLEFC3 to process the test-request function parameter on the multiple test-request card.

CONTR - MG
This routine is entered if the multiple test-request card switch is on. It calculates the new test-request number and moves it to the print area. It then checks to see if there are more multiple test-request cards to follow. If there are not, it writes the card image on SYSLST and turns off the multiple test-request card switch. It turns on the last test-request indicator in the first byte of the test-request table entry and then branches to INTLEN. If there are more multiple test-request cards to follow, a branch is taken to INTLEN.

CONTRTN - MT
This routine is entered to test for a blank column in either the first character position or at the end of the test-request panel parameter. It tests the multiple test-request card switch. If it is on, it writes the card image on SYSLST, reads another card, and tests the new card to be sure columns 1 through 15 are blank, indicating a multiple test-request card. If they are not blank, an error routine is entered because the last card processed indicated this card would be a multiple test-request card.

DELMT - ME
This routine is entered after the third (EVERY) parameter of the ON condition has been stored. It updates the card column pointer by one to point to the column following the right parenthesis. It checks to see if it contains a comma or is blank. If the column contains a comma, it branches to CALON. If it is not a comma, it tests for a blank column and branches to either FINDCOM or to OPTNPAR.

DISKOFL - MV
This routine is entered if a disk overflow condition occurred while writing on SYSLNK. It writes the error message 'DISK WORK AREA TOO SMALL' on SYSLOG and/or SYSLST.

DSFORMAT - MF
This routine is entered after the display end address of the test-request card is processed. It tests to see if a panel is also requested. If it is not, it stores the return address of TSTMTR in register 10 and then branches to test for the display format parameter.

DSFSMD - MB
This is an entry point into this phase from ATLEFC3. This routine checks the test-request table entry to see if a display or panel is present. It sets the panel or display indicator to indicate a hexadecimal display and/or general register panel.

DSKINCR - MT
This routine increases the disk address by one each time a physical record is read. The record number is increased by one and a test is made to see if it is the last record of the track. If it is, the head number is increased by one and the low limit of the track is moved into the disk address. A test is then made to see if the maximum number of heads has been reached. If it has, the cylinder number is updated by one and the low limit head number is moved into the disk address. A test is then made for the cylinder overflow. If the maximum value for the record, head or cylinder is not indicated when it is tested, a branch is taken to return via register 2.

DSPADD - MN

DSPASMD - MB
Sets the display indicator for the test-request control record to indicate a hexadecimal format.

DSPLTH - MJ
If a display is required, this routine is entered while building the test-request control table entry. It adds six, the length of the start and end addresses of the display, to the length counter. It moves the start and end addresses to the work area for building the test-request control record and branches to test if primary or secondary ON parameters are specified.

ERCOND - ME
Beginning of error routine that moves the error code of M to the print area, indicating an error in one or more parameters of the ON condition.

ERDSFMT - MK
Beginning of error routine that is entered if an invalid character is specified in the display format parameter of the test-request card, or if a floating-point display is requested and floating-point hardware is not present in the system.

ERFNCT - MX

ERPNL - ML
This error routine is entered if a character is used more than once in the panel format parameter, or if the character is not recognized as a valid panel format character.

ERPRINT1 - MA
Exit point to ERPRINT in ATLEFC3 if the test-request error switch is on after the format character has been analyzed.

ERRTN1 - MS
Loads the address of the print area error table into register 1 and tests to see if six error table entries have been made. If they have, it indicates the error table is full and a branch is taken to ABRTR to test for a multiple test-request card. If the error table is not full, the error-in-the-test-request card switch is turned on and a branch is taken to MVCODE.

ERSVC - MU
This routine is entered if all the SVC numbers (55-254) for Autotest have been used or the maximum number of test requests (200) has been reached. It fetches ATLEFC2 to process the error, indicating the maximum number of SVC numbers or test requests has been exceeded.

EVERY - MD
This routine is executed after it has been determined that the character being tested is a right parenthesis. It tests to see if the third parameter of the ON condition is being processed. If it is, a branch is taken to convert the third parameter (EVERY) to binary and to store it. If the third parameter is not being processed, it tests to see if the second parameter is being processed. It turns on the third parameter missing switch if the second parameter is being processed. If neither the second nor third ON parameter is being processed, a branch is taken to FINDCOM.

EVERYMIS - MD
This routine is executed if the right parenthesis is found after the second parameter of the ON condition has been processed. It sets a switch to indicate the third parameter is missing and branches to STCOUNT.

EXTPAR - MA
Error routine that is entered if a blank column is not found after the display format parameter has been processed. It checks to see if the card column after the panel format parameter is a comma. If it is, column 72 is checked for a blank column. If that is not blank, the column past the comma is tested for a blank column and if that is blank, TSTMTR1 routine is entered. If any of the preceding tested conditions do not exist, EXTPAR1 routine is entered to make an asterisk error table entry.

EXTPAR1 - MA

E2ADROK - MV
This routine is executed while writing the Autotest table record on SYSLNK. It sets byte 24 (OPTSWT) of the Autotest communication region to indicate the Autotest table is present. It gets the seek address and increases the record number by one. It then stores the new record number and tests to see if the maximum record number has been reached. If it has not, a branch is taken to E2OVFL1. If the maximum record number has been reached, it resets the record number to its low limit, and gets the head number. It tests for track overflow. If no track overflow condition exists, a branch is taken to E2OVFL1. If it does exist, the track number is reset to its high limit and the cylinder number is decreased by one and stored as the new cylinder number. A branch to E2OVFL1 is then made.

E2ADROKA - MW
This routine is executed to write the test-request control record on SYSLNK. It saves the record address and moves the maximum record number of 13 into the field labeled RECTBL, to be used in the disk increment routine. It then increases the disk address, gets the new address, stores it, and checks for a disk overflow. It branches to DISKOFL if overflow occurs. If there is no disk overflow, it writes the test-request control record on SYSLNK and moves the end address of the test-request control table to the Autotest communication region, byte 104.

E2OVFL - MV
This routine tests for disk overflow. If overflow has occurred, it branches to DISKOFL. If disk overflow has not occurred, the Autotest table record is written on SYSLNK and the Autotest table disk end address is updated in the Autotest communication region (byte 50).

E2OVFL1 - MV
This routine moves the new seek address to the count key field, E2ADDR.

E2PT2 - MW
This routine is executed to write the test-request control record on SYSLNK. It gets the record address and tests to see if it is zero. If it is zero, it branches to fetch the B-transient, $$BDUMP. If it is not zero, it branches to E2ADROKA.

FCHID1 - MH
This routine is executed after the test-request table entry has been completed. It turns off the error-in-the-test-request-card switch and fetches ATLEFC3. The entry point for ATLEFC3 is CARDID10.

FCHON - MB
This routine processes the primary ON parameters on the test-request card. This phase is fetched by ATLEFC3 after it has processed the test-point address and determined that the ON parameters are present. This label is the entry point into this phase.

FCHON1 - MB
After the end address of the test-request display has been processed and ON-condition parameters are found, this routine processes the secondary ON parameters. This phase is fetched by ATLEFC3 and this label is the entry point into this phase.

FCHTRTB - MA

FINDCOM - MD
If the first parameter (BEGIN) of the ON condition has been processed or the third parameter (EVERY) is found to be greater than 9999 and the right parenthesis has just been detected, this routine is executed. It checks each succeeding column of the card until a comma or end-of-card is found. If a comma is found, an error routine is executed to indicate invalid ON parameters. If end-of-card is found, a branch is taken to fetch ATLEFC3 to write the error message 'CARD IGNORED' on SYSLST.

FIXHTAB - MJ
Moves the contents of the test-request control table work area to the test-request control table and clears the work area before branching to HTABREC.

FLTPNTL - MF
Sets the first byte (function byte) of the test-request table entry to indicate a floating-point double precision display. It tests to see if the display start address is on a doubleword boundary. If the start address is not on a doubleword boundary, it is ignored and processing continues.

FLTPNTS - MF
Sets the first byte of the test-request control record to indicate a floating-point single precision display, and tests to see if the display start address is on a fullword boundary. If the start address is not on a fullword boundary, it branches to an error routine.

FRMTCOM - MF
Tests for a comma in the first column of the display format parameter on the test-request card. A comma indicates the format parameter is missing. Therefore, the first byte of the test-request table entry is set to indicate a hexadecimal display. If a comma is not found, the column is tested for a character C to indicate character format, a character B which indicates hexadecimal and character format, a character D which indicates floating-point long format, a character F which indicates fixed-point fullword format, a character H which indicates fixed-point halfword format, a character M which indicates hexadecimal and mnemonic format, and a character E which indicates floating-point short format for the display.

FXDFW - MF
Sets the first byte of the test-request table entry to indicate a fixed-point fullword display format. It then tests to see if the display start address is on a fullword boundary. If it is not on a fullword boundary, it branches to IGNORE. If it is on a fullword boundary, it returns via register 10.

FXDHW - MF
Sets the first byte of the test-request table entry to indicate a fixed-point halfword format and tests to see if the display start address is on a halfword boundary. If it is not, a branch is taken to IGNORE. If it is on a halfword boundary, it returns via link register 10.

GOWRITE - MG
Writes the core image library record back on SYSRES after the Autotest SVC number has been inserted at the test-point address in the core image library record.

HDSP - MJ
Tests to see if a display is requested. If it is, it branches to DSPLTH. If it is not requested, it branches to HON to test if the primary or secondary ON parameters are specified.

HEADLINE - MS
This routine moves the header line to the print area.

HEXCHR - MF

HEXFRMT - MF

HEXFRMT1 - MF

HEXNEM - MF

HON - MJ
Tests to see if primary or secondary ON parameters are to be specified in the test-request control table entry.

HTAB - MJ
This routine adds the current test-request control table address to the length of the test-request control table entry to calculate the new test-request control table address. It tests to see if this value is greater than the end address of the test-request control table record area. If it is, a branch is taken to write the test-request control record on SYSLNK. If it is not, the new test-request control table address is stored and register 4 is restored to the old table address before branching to FIXHTAB.

HTABREC - MK
This routine updates the record count by two for each test-request control table entry. It is updated by two for each entry so that the low-order bit of the entry counter can be saved as an indicator for the last physical record of the test-request control table.

IGNORE - MM
Error routine that is executed if there are too many parameters specified in the test-request card, or if an invalid fullword or doubleword boundary is specified for floating-point panel. It moves an asterisk (*) into the print area error table and ignores the error.

INITCIL - MN
This routine initializes the register before inserting the SVC number into the core image library. It loads the test-point address into register 2 and the address of the SVC to be inserted into the core image library into register 3. It then clears register 4 and inserts the SVC length of two into register 4.

INITHTAB - MJ
This routine moves the HCODE code byte and the test-request number to a work area for building the test-request control table entry. It updates the work area pointer and adds the length of the bytes moved to the work area to register 4.

INITPTAB - MP
This routine initializes the pointers to the ACODE work area and the test-request table. It also initializes the loop counter to one.

INSERT - MC
Increases the character counter by one and tests to see if the maximum number of characters in the parameter has been processed. If the maximum number has not been reached, another character is processed. If the maximum number has been reached, a branch is taken to test for a comma sub-delimiter.

INTLEN - MG
Clears register 5 and calculates the length of the test-request table entry for the multiple test-request card being processed.

INTPNL - MA
Updates the card area pointer to point to the format parameter, and tests for a blank column or a comma in the first column of the format parameter.

IORTN - MT
Beginning of subroutine to seek and read a record on SYSLNK.

IPEXT - MT

LEN1 - MN
This routine is entered when a multiple test-request card is read. It calculates the length in halfwords of the test-request table intry for this card. It adds the result to the current length of the test-request table and stores this new length in LENPAT.

The length is calculated by adding the length of the ON parameters (if present) to the length of the display parameters (if present). It then adds 2 for the code byte and the test-request identification number to the result.

LEN2 - MN

Tests to see if a display has been requested for the multiple test-request card being processed. If a display has been requested, it branches to DSPADD to add the length of the display parameters to the length of the ON parameters.

LEN3 - MN

Adds 2 to register 5 for the length of the code byte and the test-request identification number. It then shifts register 5 left four bits and inserts the length of this test-request entry into the first three bits of the first byte of the test-request table entry and also into a save area, LENCODE. It then restores register 5 by shifting right four bits, and adds the current length of the test-request table into register 5. It stores the new length of the test-request table in LENPAT.

LMTLNTH - MP

Initializes register 3 to point to the start address of the display and register 4 to three, the length of the start address. It then moves the start address into the Autotest table.

MDFYADR - MF

MISON2 - MB

This routine processes the secondary ON parameters (if present) for a panel and the panel format parameter. It is entered from ATLEFC3 after the test-request function parameter has been processed and found to be a request for a panel.

MLTON - MP

This routine is entered if secondary ON-condition parameters are present when building the test-request table entry. It loads the address of the secondary ON-condition BEGIN parameter into register 3 and branches to ONLNTH1.

MLTON1 - MJ

This routine is entered if the primary ON-condition parameters are to be specified in the test-request control table entry. It moves the primary BEGIN and EVERY ON parameters to the test-request control table work area. It then increases the work area pointer and turns off the primary ON-parameter switch.

MTRCONT1 - MA

Fetches ATLEFC3 and branches to MTRCONT in ATLEFC3.

MTREQ - MA

MTRMIS - MK

Error routine that is entered when a multiple test-request card is expected but another type of card is detected.

MTRTAB - MG

This routine initializes the pointers to the test-request table and the work areas. It moves the code byte, the test-request number, the ON-condition parameters (if present) and the display start and end addresses (if a display is requested) to the test-request record area. It then turns on a switch to indicate that it has started to enter the character into the test-request control table record and branches to INITHTAB.

MTR40 - MH

If the test-request table entry has been completed and there is not an error in the test-request or multiple test-request card, this routine fetches ATLEFC3 and branches to RDCARD.

MVCODE - MS

Moves the error code from a save area to the error table. It updates the error table pointer by two and stores the new error table pointer.

MVTRNR - MU

This routine is entered after the new test-request number has been calculated. It moves the new test-request number to the print area and turns off the test-request-number-being-processed switch. It then moves the new test-request number to the TRNR save area and the new SVC number to the SVCNR save area.

NONCOND - MC

Turns off the primary-ON-parameters-present switch, when a comma is found to be the first character of the ON parameter field.

NXTREQ - ML

Clears work area and switches and exits to CHK711 in ATLEFC3 to test for a multiple test-request card.

ONADD - MN

If the ON-condition parameters are present for this test-request table entry, it inserts the length of the ON-condition parameters into register 5.

ONCOND - MC
Tests for a comma in the first card column of the ON parameters. If a comma is found, it indicates the ON parameters are missing. The primary ON-parameter switch is turned off. If a comma is not found, a test is made for the left parenthesis. If a left parenthesis is found, the ON parameters are processed. If the ON parameters are not found, a branch is taken to RIGHTEND to look for a right parenthesis or end-of-card.

ONLNTH - MP
This routine is entered if the ON-condition parameters are present when the test-request table entry is being built. It tests to see if the secondary ON-condition parameters are present. If they are, it loads the address of the first parameter (BEGIN) of the secondary ON condition into register 3. If they are not present, it loads the address of the first parameter of the primary ON condition into register 3 and then branches to ONLNTH1.

ONLNTH1 - MP
Initializes the loop counter to six, the number of characters in the ON-condition parameters. It then branches to insert the ON-condition parameters into the test-request table entry work area.

ONLTH - MJ
This routine adds 4, the length of the BEGIN and EVERY ON parameters, to the length counter. It then tests to see if the primary or secondary ON condition is to be used and moves either the primary BEGIN and EVERY parameters or the secondary BEGIN and EVERY parameters to the test-request control table work area.

ONSTORE - MC
Stores the valid decimal character of the ON parameter in the work area, STRTTAB, and updates the work area pointer by one.

OPDSF - MA
Tests for a blank column following the comma in the first column of the display or panel format parameters.

OPTDSF - MA
Tests for a blank column. If the column is not blank, the format characters of the panel are analyzed. If it is a blank column, a branch is taken to OPTPNL.

OPTNL - MB

OPTNL1 - MX

OPTNPAR - ME
If a blank column has been found in the SUBPAR routine, this routine changes the exit from the CALON routine to return to DSFSMD.

OPTPNL - MB
This routine is entered when this phase is fetched by ATLEFC3 to process the assumed panel format.

PACKREG - MX

PATCH - MR
This routine tests to see if a complete Autotest table record (200 bytes) has been built. If it has, the record is written on SYSLNK. If the record is not complete, the number of bytes specified in the loop counter are moved into the Autotest table record area.

PATNOP - MH
In order to determine the number of bytes of NO-OP's to patch into the core image library record, this routine subtracts the length of the SVC instruction inserted at the test-point address in the core image library from the length of the saved instruction. It then inserts the number of required bytes of NO-OP's following the SVC instruction patched into the core image library record.

PNLASMD - MB
Sets the panel indicator for the test-request control record to indicate a general register panel is required.

PNLF - ML
This routine is entered if the character F is found in the panel format parameters on the test-request card. It tests to see if the F character has been previously processed and enters an error routine if it has. If it has not, it sets the code byte of the test-request table entry and the test-request control record to indicate a floating-point register panel. It then sets a switch to indicate the character F has been processed.

PNLFMT - MB
This routine is entered from ATLEFC3 to process the format parameter of the test-request card.

PNLG - ML
This routine is entered when the character G is found in the panel format parameter. It tests to see if the character G has been previously processed and branches to an error routine if it has. If it has not, it sets a switch to indicate the character G has been processed and sets the test-request control entry code byte to indicate a general register panel is requested.

PNLMTR - MB

PNLRTN - ML
This routine analyzes the characters in the panel format parameters on the test-request card and branches to the appropriate routine to set the first byte of the test-request control table entry to indicate the panel format requested.

PNLS - ML
This routine is entered if the S character is found in the panel format parameters. It tests to see if an S character has been previously processed. If it has, an error routine is entered. If it has not, it sets the code byte of the test-request control record to indicate a panel of storage positions 24 to 127, and sets a switch to indicate the S character has been processed.

PNTHD - MQ
This routine is entered when the SYSLST line counter reaches its maximum value, indicating the page is full. It resets the line counter, sets up the header line, skips to the next page, and prints the header line.

PNTID - MU
This routine is entered to calculate the new SVC number or the new test-request number. It tests to see if the job-cancel switch is on. If it is on, it bypasses the calculation of the new SVC or test-request number by returning via register 7. If the cancel switch is off, CARDSWI (X'80') is tested to see if the new SVC number or the new test-request number is being calculated. If the **switch is off, the current SVC number is decreased by one to get the new SVC number. If the switch is on, the test-request number is increased by one to get the new test-request number. A branch is taken to RTNID to store the new SVC or new test-request number.**

POINTER - MM
Locates the test-point address within the core image library record read from SYSRES.

PRINTPAT - MQ
This routine writes a line on SYSLST and updates the line counter. It then checks for a full page. If the page is full, it skips to the beginning of the next page and writes the header line on SYSLST.

PRINTPA1 - MX

PROCESS - MK
If the multiple test-request card switch is on, this routine is entered before closing the test-request table entry. It writes the card image on SYSLST, branches to CONTPRO to clear the work area, and reads the multiple test-request card.

PROCSTRT - MC
Initializes ONSWITCH to X'01' to indicate that the first parameter within the left and right parentheses (the BEGIN ON-condition parameter) is being processed.

PROCS1 - MC
Clears the character counter that is used to count the number of characters in a parameter, stores the work area address in register 4, and saves the address of the first character of the ON parameter in register 14.

PTABLNTH - MR
This routine calculates the total length of the Autotest table, and the length of the table entry for each SVC number.

PTABREC - MR
This routine is entered whenever the end of the Autotest table record is reached and there are more characters to be inserted in the table. It re-initializes the test-request table entry pointer to the first byte of the table. It turns off the E2SWI switch (X'80') to indicate the Autotest table record is ready to be written on SYSLNK. It writes the Autotest table record on SYSLNK and branches to PATCH to complete the test-request table entry.

READPAT - MP
Beginning of subroutine to read a card from SYSIPT.

READREC - MW

READRTN - MW
This routine reads a record from the core image library. It saves the instruction at the test-point address so that an SVC number and/or NO-OP can be inserted into the space vacated by the saved instruction.

RGHT - MM
After the third parameter of the ON condition has been processed, this routine tests for a right parenthesis.

RIGHTEND - ME
This routine is entered if an error is found in an ON-condition parameter. It tests for a right parenthesis and end-of-card. If the right parenthesis is found, a branch is taken to FINDCOM. If end-of-card is reached, ATLEFC3 is fetched to write the 'CARD IGNORED' message on SYSLST.

RTNE2 - MV
This routine is entered to write either an Autotest table record or a test-request control record on SYSLNK. It tests to see if the Autotest table record is ready to be written. If it is, it branches to E2PT2 to write the record. If a test-request control record is being written on SYSLNK, it tests to see if the disk address is zero. If it is, it branches to fetch the B-transient, $$BDUMP. If it is not, it branches to E2ADROK.

RTRNID - MU
Stores the new SVC or new test-request number in a save area, SVCNR, and tests to see if a test-request number is being processed. If it is, a branch is taken to CLRSAVE. If an SVC number is being processed, a test is made to see if all SVC numbers used by Autotest have been assigned. If not, it branches to CLRSAVE. If all SVC numbers have been assigned, an error routine is entered to fetch ATLEFC2 to process the error.

SETMTR - MX

STCOUNT - MC
This routine is entered after a sub-delimiter for the ON condition is found. It checks for a character count of zero and enters an error routine if the count is zero, indicating a parameter is missing. If the count is not zero, the parameter is packed, converted to binary, and stored in a save area. A test is then made to see which of the three parameters is being processed (BEGIN, END, or EVERY). A routine is then entered to store the parameter being processed.

STEND - MC
This routine is entered after the second parameter of the ON condition has been processed. It stores the END ON-condition parameter in ENDADR and tests to see if the value is less than 9999, which is the limiting number of an ON-condition parameter. It turns off the second-parameter-being-processed switch. It then tests to see if the third parameter is missing and branches to either check the third parameter or to store the assumed value of one for the third parameter.

STEVE - MD
Converts the third parameter of the ON condition to binary and tests to see if the value of the third parameter is valid.

STEVERY - MD
This routine is entered after the third parameter of the ON condition has been processed. It tests to see if the length of the parameter is zero. If the length is zero, it branches to store the assumed value of one for the third parameter (EVERY). If the length is not zero, it packs the third parameter before converting it to binary at label STEVE.

STLSND - MB

STONE - ME
This routine is entered if the third parameter of the ON condition is missing. It stores one as an assumed value for the third parameter (EVERY).

STOREEVE - ME

STRT - MC
This routine updates the card column pointer by one and tests to see if it contains a comma, indicating the end of a parameter within the left and right parenthesis of the ON condition. If a comma is found, a branch is taken to CHKCOUNT to test the length of the parameter. If the column does not contain a comma, a test is made for a valid decimal digit.

STSTART - MC

This routine is entered after the first parameter of the ON condition has been processed. It stores the BEGIN ON parameter in STRTADR, and tests to see if the value is less than maximum value (9999) allowed. It turns off the first-parameter-being-processed switch and turns on the second-parameter-being-processed switch. It then branches to process the second parameter of the ON condition.

SUBDLMT - ME

This routine is entered if the character count for the ON parameter reaches four and a sub-delimiter has not been found. It tests to see if the third parameter is being processed. If it is, it branches to STCOUNT. If it is not, it tests for a comma sub-delimiter in the next column and branches to an error routine if it is not found.

SUBPAR - ME

This routine is entered after the third parameter of the ON condition has been processed. It tests for a comma following the third parameter. If one is found, it branches to RIGHTEND. If a comma is not found, it tests the next column for a right parenthesis. If a right parenthesis is found, it branches to store the third parameter.

SVCIL - MQ

This routine is entered to get the saved instruction from the core image library record. It moves one character at a time from the input area to the instruction save area until either the end of the core image library record, or the end of the instruction is reached. When the end of the instruction is reached, a branch is taken to return via register 7 to insert the SVC number into the core image library record and to patch NO-OP's into the core image library record (if requested). If the end of the core image library record is found before the complete instruction is saved, a branch is taken to insert the SVC number and the NO-OP's (if required) into the current core image library record. This record is then written back into the core image library. The next record is read to get the rest of the saved instruction and to patch NO-OP's into the core image library record. A branch is then taken to build the test-request table entry.

SVINIT - MN

This routine loads the address of the saved core image library instruction (displacement of the instruction within the core image library record) into register 3. It moves the length of the saved instruction as calculated by subroutine TSTLNTH, into save area, LNTCNTR.

SYMER - MX

TABCRTN - MG

This routine locates and reads the record that contains the test-point address from the core image library. It locates the test-point address within the core image library record, tests for an invalid instruction at the test-point address (BAL, BALR, or SVC), and calculates the SVC and the test-request number to be used for this test-request entry. It then tests to see if this is a multiple test-request card. If it is not, it writes the card image on SYSLST and branches to TRTAB. If it is a multiple test-request card, it turns on the multiple test-request card switch before branching to TRTAB.

TABLNTH - MR

This routine calculates the length of the test-request table entry by adding the following values:

1. The length of the core image library saved instruction.

2. A fixed length of six (representing the length of the SVC numbers plus 2 bytes of zeros).

3. The length of the current test-request number.

It then stores this result in LENPAT as the length of the test-request table entry. Next, it adds the length of the test-request table entry to the total length of the Autotest table and stores this in LENTAB as the total length of the Autotest table. It then initializes register 4 to the length of this entry, and register 3 to the address of the length of this entry.

TABREC - MR
This routine is entered when 200 bytes have been moved into the Autotest table record. It stores registers 2 through 15 and tests to see if there are more characters to be inserted into the Autotest table. If there are, it branches to TRTAB1. If there are not, it branches to PTABREC.

TABREC1 - MR
Turns on ERRSWI (X'02') to indicate the Autotest table record is complete but there are more characters to be moved into the record.

TAPEIOCL - MS
This routine is entered if the end of the core image library is reached when inserting the SVC number into the core image library record. It writes the current record back into the core image library and reads the next record. It then branches to SVCIL.

TAPEIOSV - MQ
This routine is entered if the end of the core image library is found while saving the core image library instructions. It initializes registers, and sets a switch to indicate that the complete core image library instruction has not been saved. It then inserts the SVC number and NO-OP's (if required) into the core image library record. It then branches to CILPATCH to write that record back into the core image library and reads the next record.

TRFTMT4 - MA
An entry point into this phase from ATLEFC3. After the end address parameter on the test-request card has been processed and a comma is found, this routine branches to test for the display format, if a display is requested. If a display is not requested, it branches to INTPNL.

TRLAST - MP
This routine is entered if the multiple test-request switch is off when the initialization routine for building the test-request table entry is executed. It sets the first byte of the test-request table entry to indicate that this is the last entry for the current SVC number.

TRTAB - MG
This routine analyzes the length of the core image library instruction and stores the length of the instruction and the instruction itself. It initializes registers for inserting the SVC number in the core image library record and then moves the SVC number into the core image library record.

TRTAB1 - MR
This routine is entered to write the test-request control record on SYSLNK. It initializes the test-request control record pointer to the first byte of the record. It turns on a switch to indicate the test-request control record is being written and writes the record on SYSLNK.

TR2 - MN

TR4 - MN

TR6 - MN

TSTLMT - MP
This routine is entered when building the test-request table entry. It tests to see if a display is required. If it is, it inserts the start and end addresses into the test-request table entry work area. If a display is not required, it returns to build the test-request control table record.

TSTLNTH - MN
This routine determines the length of the saved core image library instruction by "testing under mask" with a X'C0' the first byte of the instruction. If the first two bits of the instruction are ones, it indicates the instruction length is 6 bytes long. If the first two bits are zeros, the instruction length is 2 bytes long. Any other combination indicates an instruction length of 4.

TSTMTR - MA
Tests for a blank column after the display or panel format. A blank column should be present because this is the last parameter on the test-request card. If the column is not blank, an error routine is entered.

TSTMTR1 - MA

TSTON - MP
This routine is entered when building the test-request table entry. It tests to see if the ON parameters have been processed. If they have not been processed and the primary ON parameters are not to be used, it returns to process the start and end addresses if a display is requested. If the secondary or primary ON parameters are to be used, it branches to ONLNTH to see which one is to be moved into the test-request table entry.

TSTPNL - MX

UPIT - MW

UPLNTH - MP
Initializes register 3 to point to the end address of the display. It initializes register 4 to three, the length of the end address, and moves the end address into the Autotest table record.

UPTRNR - MU
This routine is entered if the new test-request number is being processed by the PNTID routine. It increases the current test-request number by one to get the new test-request number.

WRITERTN - MW
This routine writes the core image library record back on SYSRES after it has been updated with the SVC number and/or NO-OP's.

WRNGMTR - MB
This routine is entered when an error is detected in the test-request card being analyzed by ATLEFC3. It turns off the multiple test-request error switch, turns on the test-request card error switch, resets the test-request code pointer, and turns on the indicator for the last entry for this SVC. It then branches to close the test-request table entry.

PCC Statement Processor (ATLEFC5) Charts NA-NC

BADSW
Listing only.
Program switch.

Turned on whenever an invalid character is found in the first or second operand of the PCC card.

BUMP - NB
Each time this routine is entered, it adds one to REG3 to point it to the next column of the PCC card. The column is then tested for a blank indicating the end of the card, or a comma indicating the end of the operand. If the column contains a value other than blank or a comma, a branch is taken to a routine that tests to determine if the character is valid.

CNCLNOP - NC
Program switch.

Checks to see if a cancel condition occurred in a previous control-card-analysis phase. If it has, the CANCEL B-transient ($$BCNCL) is fetched. If no cancel condition occurred, a test is made to see if there is room for the test-request output on disk. If not, error message 9902I is listed. If there is room, job control is fetched.

CVB - NC
Sets REG4 to point to the units position of LOOPCTR, and sets REG2 to control the right justification of the LOOPCTR.

CVBLOOP - NC
LOOPCTR is a 4-position counter, containing the number of test-request actions desired. This number can be any value from 0 to 9999, and it is moved from the PCC card into LOOPCTR, starting at the high-order position. If a number, three digits or less, is moved into LOOPCTR, it must be right justified before it can be analyzed. This right justification is done by CVBLOOP.

D - NA
Moves the error type character, "D" or "T", saved from the control-card-analysis phase, ATLEFC3, to the 108th position of the print area. It points REG6 at the 110th position of the print area for the next error-type code.

DRT - NA
Checks for a cancel condition from a previous control-card-analysis phase. It turns off the cancel switch, if there was not a cancel condition. It also tests the first operand of the PCC card to see if a normal EOJ dump has been requested.

END1 - NC
Tests BADSW at the end of each of the first two operands to determine if an invalid character has been found. Writes error message 1 or 3 on SYSLST if an invalid character was found.

EOJ - NC
Writes card image and error codes on SYSLST. Writes Autotest communication region on SYSLNK.

EOJ1 - NB
Tests BADSW. If BADSW is on, indicating a comma is missing between the first two operands of the PCC card, this routine writes error message 1 or 3 on SYSLST.

GO - NA
Stores assumed values for PCC card in Autotest communication area. These values may be changed, depending on what is punched in the PCC card. The assumed values are as follows:

1. Normal EOJ dump in hexadecimal characters.

2. Abnormal EOJ dump in hexadecimal characters.

3. Fifty test-request actions.

INVALD1 - NB
Turns on BADSW to indicate that an invalid character was punched in the first or second operand of the PCC card. Valid characters for the first operand are N, comma, S, A, M, or X. Valid characters for the second operand are comma, S, A, or M.

LOOK - NA
Tests first and second operand for a blank or comma. If it is blank, a branch is taken to EOJ. If a comma is detected, it indicates the end of the operand being tested. If it is neither a blank nor a comma, the character is tested to see if it is a valid character for the first or second operand.

LOOPICK - NC
Tests the number of test-request actions desired to ensure that it is a numeric value not more than four decimal places in length.

MSG1 - NB
If a comma is missing or an invalid character in the first or second operand is found, this routine writes the error message 1 or 3, indicating the condition, on SYSLST.

MSG2 - NB
Writes error message 2 or 4, indicating a comma missing after first or second operand of the PCC card, on SYSLST.

MSG3 - NC
Writes error message 1 or 3, indicating an invalid entry in the first or second operand of the PCC card, on SYSLST.

NA - NB
Program switch.

Turned on to indicate that an "A" is punched in the first or second operand of the PCC card. If the "A" is detected in the first operand, a hexadecimal and alphameric character dump is desired for normal EOJ. If it is detected in the second operand, a hexadecimal and alphameric character dump is requested for abnormal EOJ.

NM - NB
Program switch.

Turned on by an "M" in the first or second operand of the PCC card. If an "M" is in the first operand, a hexadecimal and a mnemonic normal EOJ dump is requested. If it is in the second operand, a hexadecimal and mnemonic abnormal EOJ dump is requested.

NONOREOJ - NA
Sets DUMPSW in Autotest communication region to indicate that an "N" was detected in the first operand of the PCC card, and that a core dump is not desired after normal EOJ.

NORALP - NB
Turns on switch NA to indicate a character, A, was found in the first or second operand of the PCC card.

NORMNE - NB
Turns on switch NM to indicate an "M" was detected in first or second operand of the PCC card.

NORSUP - NB
Turns on switch NX to indicate that supervisor is to be included in the normal EOJ dump.

NORSYM - NB
Turns on switch NS to indicate an "S" was detected in the first or second operand of the PCC card. It also checks to see if symbols are present. If they are not, error message 7 is written on SYSLST.

NOTNUM - NC
Writes error message 6, indicating that a value other than a numeric from 0 to 9 was used in the test-request action field of the PCC card.

NS - NB
A switch that is turned on by an "S" in the first or second operand of the PCC card. If the S is in the first operand, a hexadecimal character and symbolic dump is requested for normal EOJ dump. If it is in the second operand, a hexadecimal character and symbolic dump is requested for abnormal EOJ dump.

NX - NB
A switch that is set by an "X" in the first operand of the PCC card. It indicates that the supervisor is to be dumped with normal EOJ dump.

SECOND - NC
Program switch.

If on, indicates that the second operand of the PCC card has been tested. If off, sets DMPFORMT in Autotest communication region to indicate second parameter of PCC card. Sets error messages to indicate second parameter error.

SETALP - NB
Sets DMPFORMT flag in Autotest communication region to indicate the dump will be printed in its EBCDIC representation.

SETMNE - NB
Sets DMPFORMT flag in Autotest communication region to indicate the dump will be in hexadecimal characters and mnemonic.

SETSYM - NB
Sets DMPFORMT flag in Autotest communication region to indicate a hexadecimal character and symbolic dump, if symbol table is present.

START - NA
Opens SYSLST.

THIRD - NC
Checks to determine if there is a value punched in the third operand of the PCC card, indicating the maximum number of test-request actions to be performed.

Autopatch User Program (ATLEFC7) Charts PA-PU

ABORT1 - PK
ABORT1 is entered from ABORT7, MISPCC, or ERTRSVC. It uses the PRINTPAT subroutine to write on SYSLST either the 'Z' error code, 'PCC MISSING' message, or 'G' error code and its associated card image. It then executes an SVC 6 to cancel the job.

ABORT7 - PA
This routine is one of the three entry points into ATLEFC7. It inserts an error code 'Z' into the error table in the output area for SYSLST, and branches to ABORT1 to write the 'Z' on SYSLST and to cancel the job.

ADCOUNT - PF
This routine is entered from ADDCONT. It loads register 5 with the length of the current add patch entry in the Autotest table, and branches to PATCIL to insert this entry into the Autotest table.

ADDCONT - PF
ADDCONT is entered when PATINST has completed entering an autopatch table entry into the Autotest table from an ADD card. ADDCONT reads the next card from SYSIPT, and initializes the first instruction in the PATCIL routine to handle the insertion of a new instruction into the core image library. ADDCONT then checks to see if the new card is an Autotest control card. If it is, control branches to ADCOUNT; if not, it is checked to see whether it is a job control card. If it is, control branches to MISPCC to handle this error condition. If not, control branches to CARDAST to begin checking it for validity.

ADDEND - PE
This routine is executed after an ADD card has been processed. It clears CARDSWI and the work area where the length of add patch entries is calculated and then branches to FTCHID.

ADDRTN - PH

This routine finds the core image library instruction address at which the add patch is to be inserted. It determines the length of the instruction, saves this length and the instruction itself, and overlays the core image library address with a linking SVC instruction. Finally, it tests the length of the replaced user instruction to see whether the overlaying SVC was long enough to use up all the storage originally allocated to the instruction. If it was, ADDRTN branches to PTCIL24; if not, it branches to CILNOPP to fill in the unused bytes with NO-OP's.

ADD2 - PQ

This routine is entered from CILNTH when that routine has found the length of the core image library instruction to be two bytes. ADD2 loads this value, two, into register 4 and branches to ADDRTN via register 7.

ADD4 - PQ

This routine is entered from CILNTH when that routine has found the length of the core image library instruction to be four bytes. ADD4 loads this value, four, into register 4 and branches to ADDRTN via register 7.

ADD6 - PQ

This routine is entered from CILNTH when that routine has found the length of the core image library instruction to be six bytes. ADD6 loads this value, six, into register 4 and branches to ADDRTN via register 7.

BALCHECK - PQ

According to Autotest specifications, a patch card cannot specify a patch-point address in the core image library for a BAL, BALR, or SVC instruction, nor can the instruction to be inserted be one of these three. ATLEFC2 checks the former; BALCHECK in ATLEFC7 checks the latter. BALCHECK compares the op-code of the instruction addressed by register 2 to X'05', X'45', and X'0A', the op-code constants for a BAL, BALR, and SVC, respectively. If the instruction op-code is valid, control branches to the address contained in register 7. If the op-code is invalid, control branches to CILER.

CARDAST - PF

This routine is entered from either ADDCONT or ERPNT when a new card has been read and found to be a patch card. CARDAST performs five diagnostic functions on the patch card, using five subroutines. CHKAST ensures that the asterisk required on the card is present; VALIDHEX makes sure that the hexadecimal parameter on the card is correct; INSTCHK gets the address of the user patch; and BALCHECK checks for instructions that cannot be specified as patches. Finally, TESTER sees whether any other errors have been detected.

When CARDAST has determined that the card is valid, it uses the PRINTPAT subroutine to write the card image on SYSLST, initializes the necessary instruction in PATINST and branches to PATINST.

CARDCON - PJ

This routine is entered from CONCONT when it is determined that an Autotest control card has been read. CARDCON turns off CARDSWI, writes the core image library record on SYSRES, and fetches ATLEFC2, entry point CARDID, to analyze the card.

CHCKAST - PN

This routine checks for the presence of the required asterisk on an autopatch card. CHCKAST gets the addresses of the card input area and the error table in the output area for SYSLST, then checks to see whether the asterisk is on the card. If it is not, control branches to ERAST to insert an error code into the table and to write the card image on SYSLST. If the asterisk is there, CHCKAST branches to INITINST to see whether the patch instruction begins immediately after the asterisk.

CHKCNTRL - PA

This routine continues the function of SVC15. CHKCNTRL tests whether any test-requests have been used. If none have, it branches to SYSC5 to update the Autotest communication region. Otherwise, it loads into the output area of SYSLNK the last test-request control record, and uses RTNE2 to write the record on SYSLNK. The rest of the routine updates the disk address, checks for overflow, and branches to SYSC5 to update the Autotest communication region.

CHKDEC - PN

This routine tests the character for a valid hexadecimal representation. It branches to ERINST if it is not valid, and to STDEC if it is valid.

CHRCNT - PP

CHRCNT is entered from TRANS when that routine has determined that the hexadecimal constant has one of three allowable lengths: 12, 8, or 4 characters. CHRCNT checks the op-code of the constant to see whether the implied length contained in that code corresponds to the length determined in routine TRANS. If it does not correspond, control branches to ERINST. If it corresponds, a final test is made to see whether the end of instruction is within the phase. If it is not, a branch is taken to ERINST. If it is, control branches to the instruction in register 7.

CIL24 - PD

This routine initializes register 4 with the value 2 as the length of the core image library instruction, loads register 2 with the address of that instruction, and uses subroutine CILPATCH to insert the linking SVC instruction at the address in register 2. CIL24 then tests again to ensure that the core image instruction is two bytes long. If it is, control branches to PTCIL24; if not, it branches to CILNOPP to provide NO-OP fillers immediately following the SVC link instruction. (Because register 4 always has a value 2 before this final test, the result is always a branch to PTCIL24.)

CILADD - PQ

This routine is entered in order to get the address of the identifying SVC link to an Autotest table entry. CILADD loads register 3 with the address of this SVC, loads register 4 with the value 2 (an SVC length), and branches to the address in register 7.

CILCON - PR

This routine is used by CONRTN to pack the converted hexadecimal patch constant, and to move it to the core image library record. It first checks the core image library record to see if it has been filled. If it has, CILCON branches to CILWTREC to write the filled record on SYSRES and to get another. If the record has space yet to be used, CILCON packs the constant, moves it to the core image library record, and points to the next constant character to be moved. Each time, it checks to see whether all characters have been moved. If they have not all been moved, it branches to CILCON; but if the entire constant is now in the core image library record, it branches to the address in register 7.

CILER - PQ

This routine handles the error condition resulting from the user specifying a BAL, BALR, or SVC instruction in one of his autopatch cards. CILER initializes the beginning instruction of the MVCODE routine to move an error code 'F' into the error table in the output area for SYSLST. It then uses subroutines ERRTN and MVCODE to enter this code in the error table and branches to ERPNT to write the error code and the card image on SYSLST.

CILEXC - PB

This routine loads register 3 with the address of the patch instruction to be exchanged for one presently in the core image library. Register 4 is then cleared for use as the counter for the length of the core image library instruction. Finally, CILEXC tests the core image library instruction for length, branching to CILEXC2 if this length is two bytes; to CILEXC4 if the length is four bytes; or to CILEXC6 if the length is six bytes.

CILEXC2 - PB

This routine moves the value two into the core image library length counter and into register 4 for later use. It tests the op-code of the patch instruction to find its length, branching to CILEXC22 if this length is two bytes; to CILEXC26 if the length is six bytes; or to CILEXC24 if the length is four bytes.

CILEXC4 - PB

This routine moves the value four into the core image library length counter and into register 4 for later use. It tests the op-code of the user patch instruction to find its length, branching to CILEXC42 if this length is two bytes; to CILEXC46 if the length is six bytes; or to CILEXC44 if the length is four bytes.

CILEXC6 - PB

This routine moves the value six into the core image library length counter and into register 4 for later use. Then, because the length of the core image library instruction is great enough to contain any length exchange instruction, ENDSWI is turned on to indicate that no autopatch table entry is required for this exchange patch. CILEXC6 then tests the length of the user patch instruction, branching to CILEXC62 if this length is two bytes; to CILEXC66 if the length is six bytes; or to CILEXC64 if the length is four bytes.

CILEXC22 - PC
Because this routine handles the exchange of two instructions of equal length, ENDSWI is turned on to indicate that no autopatch table entry is required. The instructions are then exchanged in subroutine CILPATCH, and the exchange card image is written on SYSLST. CILEXC22 then branches to FTCH.

CILEXC24 - PD
This routine uses subroutine PNTID to calculate the number of the identifying SVC link to be assigned to the exchange patch. After converting it to a printable decimal format, it moves it to the output area for SYSLST, which already contains the card image of the exchange patch entry. This was done by ATLEFC2. CILEXC24 then uses the PRINTPAT subroutine to write the autopatch card image on SYSLST. Finally, the SVCIL subroutine is used to save the core image library instruction to be replaced by the user exchange patch. Control then branches to CIL24.

CILEXC26 - PF
This routine uses the subroutine PNTID to calculate the number of the identifying SVC link to be assigned to the exchange patch entry. After converting it to a printable decimal format, it moves it to the output area for SYSLST, which already contains the card image of the exchange patch entry. This was done by ATLEFC2. CILEXC26 then uses the PRINTPAT subroutine to write the autopatch card image on SYSLST. Finally, before branching to SVCIL, CILEXC26 sets register 7 to cause SVCIL to branch to CIL24 when finished.

CILEXC42 - PG
This routine handles an exchange patch where the user wishes to insert a new instruction of two bytes into a position containing an instruction four bytes long. Because the new instruction can fit easily into the space occupied by the one to be overlaid, ENDSWI is set to a value X'02' to indicate no autopatch table entry is required. The CILPATCH subroutine is then used to insert the new instruction at the patch-point address. CILNOP is used to set up the two bytes of NO-OP's to overlay the remainder of the old instruction, and CILPATCH is again used to insert these two bytes at the patch-point address plus 2. The exchange card image is then written on SYSLST, using subroutine PRINTPAT, and control branches to FTCH to write the record on SYSLNK and to fetch ATLEFC2.

CILEXC44 - PC
Because this routine handles the exchange of two instructions of equal length, ENDSWI is turned on to indicate that no autopatch table entry is required. The instructions are then exchanged in subroutine CILPATCH, and the exchange card image is written on SYSLST. CILEXC44 then branches to FTCH.

CILEXC46 - PH
This routine uses subroutine PNTID to calculate the number of the identifying SVC link to be assigned to the exchange patch entry. After converting it to printable decimal format, it moves it to the output area for SYSLST, which already contains the card image of the exchange patch entry. This has been done by ATLEFC2. The card image is then written on SYSLST, using subroutine PRINTPAT. CILEXC46 then saves the pointer to the core image library, and loads register 3 with the address of a work area for the displaced instruction. Finally, it loads register 7 to cause SVCIL to branch to CIL24, and branches to SVCIL to save the core image library instruction being exchanged.

CILEXC62 - PH
This routine is used to process an exchange patch when the user wishes to insert a new instruction of two bytes into a position containing an instruction six bytes long. Because the new instruction can fit easily into the space occupied by the one to be overlaid, no autopatch table entry is required. The CILPATCH subroutine is used to insert the new instruction at the patch-point address. Then CILEXC62 uses CILPATCH again to fill up the remaining four bytes of the old instruction with NO-OP's. The exchange card image is then written on SYSLST, using subroutine PRINTPAT, and control branches to FTCH to write the record on SYSLNK and to fetch ATLEFC2 to read another card.

CILEXC64 - PH
This routine is used to process an exchange patch when the user wishes to insert a new instruction of four bytes into a position containing an instruction six bytes long. Because the new instruction can fit easily into the space occupied by the one to be overlaid, no autopatch table entry is required. The CILPATCH subroutine is used to insert the new instruction at the patch-point address. Then CILEXC64 uses CILPATCH again to fill up the remaining two bytes of the old instruction with NO-OP's.

The exchange card image is then written on SYSLST, using subroutine PRINTPAT, and control branches to FTCH to write the record on SYSLNK and to fetch ATLEFC2 to read another card.

CILEXC66 - PC
This routine uses the subroutine CILPATCH to exchange the two instructions, writes the exchange card image on SYSLST, and branches to FTCH.

CILNOP - PQ
This routine is entered from CILEXC42 when a NO-OP of two bytes must be inserted into the core image library to overlay the remaining bytes of a displaced instruction. CILNOP loads register 4 with a length of 2, the length of the necessary NO-OP, and loads register 3 with the address of the NO-OP itself. A branch is then taken to the address in register 7.

CILNOPP - PD
This routine takes the length of the instruction overlaid by an SVC instruction for an add patch, and decreases it by two (the length of an SVC). It then inserts a series of NO-OP instructions of the same length into the core image library area immediately following the linking SVC instruction. CILNOPP then branches to PTCIL24.

CILNTH - PQ
This routine is entered from ADDRTN to determine the length of the core image library instruction at whose address an add patch entry must be made. CILNTH tests the core image library instruction length, branching to one of three routines, ADD4, ADD2, or ADD6, to load the appropriate length into register 4 and to branch back to ADDRTN.

CILPATCH - PG
This routine is used to insert new instructions and SVC links into core image library records that have been read from SYSRES. CILPATCH first checks to see whether the core image record presently in the work area has been used up in inserting the new instruction into this area. If so, CILPATCH branches to TAPEIOCL. Otherwise, CILPATCH begins moving the new instruction, character by character, into the above work area, increasing its pointer by one each time. When the entire new instruction has been moved, CILPATCH branches to the address contained in register 7.

CILWTREC - PR
This routine is executed when CILCON determines that an entire core image library record has been filled up with an autopatch entry. CILWTREC uses subroutine WRITERTN to write this record back on SYSRES, updates the counter of records read, and then has subroutine READRTN read the next core image library record. Finally, before branching back to CILCON, it loads register 2 with the address of the input area for SYSRES.

CLRPTAB - PL
This is the last step of TABREC before it branches to PATCH. In CLRPTAB the Autotest work area is cleared to zeros, and a branch is taken to PATCH.

CLRSAVE - PK
See PNTID (PK).

CNCLPCH - PJ
This routine is entered from ERRTN when it is determined that the error table in the output area for SYSLST is filled with error codes and unable to include the latest error. CNCLPCH uses PRINTPAT subroutine to write on SYSLST the card image and the error codes. It then branches to FTCH to restore the core image library instruction and to fetch ATLEFC2 to read another card.

COMPLETE - PF
This routine is entered from SVCIL when ENDSWI indicates that the instruction being saved started in the previous core image library record. COMPLETE loads register 2 with the address of the CILAREA, loads register 7 with the address to branch to when CILPATCH is completed, turns off ENDSWI, and branches to CILPATCH.

CONADR - PQ
This routine is used by CONRTN to get the address of the input card work area and to point to the card column following the asterisk on the patch card. It then returns to CONRTN by branching to the address in register 7.

CONAST - PJ
This routine is entered from CONCONT when a new card has been read by CONRTN and CONCONT establishes that it is neither an Autotest control card, nor a job control card. CONAST uses CHKAST subroutine to ensure that the card contains the required asterisk. It also

uses subroutine VALIDHEX to check the validity of the hexadecimal character on the card. It next checks the length of the parameter to ensure that there is an even number of hexadecimal characters in it, branching to ERINST if not. If all the tests show the card to be correct, CONAST then uses subroutine TESTER to check ERRSWI for a X'80' value, indicating an error found in some earlier test. If no error conditions have occurred, CONAST saves the card image in a work area and branches to CONRTN.

CONCONT - PJ
This routine tests the card read in routine CONRTN to see if it is an Autotest control card. If it is, control branches to CARDCON to fetch ATLEFC2. If not, CONCONT checks it to see if it is a job control card. If it is, control goes to MISPCC to write the error message on SYSLST and to cancel the job. If not, CONCONT branches to CONAST to analyze the card for errors.

CONRTN - PJ
This routine handles the insertion of replace constant patch entries into the core image library record. CONRTN uses subroutine CONADR to point beyond the asterisk on the patch card to the actual constant to be inserted. It then uses CONTRANS to translate the decimal characters on the card to hexadecimal format, and calculates the length of the converted (but as yet unpacked) constant. It divides this length by two to provide for its compression when packed, and uses subroutine CILCON to pack the constant and insert it into the core image library record. CONRTN then uses the PRINTPAT subroutine to write the card image on SYSLST, and finally uses READPAT to read the next card from SYSIPT. CONRTN then branches to CONCONT.

CONTRANS - PR
This routine is used by CONRTN to translate the decimal constant on a replace constant (CON) patch card to hexadecimal format. It loads register 8 with a count of the number of characters in the constant, gets its address, and branches to RDREG to translate it.

DSKINCR - PU
This routine increases the disk address by one each time a physical record is read. The record number is increased by one and a test is made to see if it is the last record of the track. If it is, the head number is increased by one and the track low limit is moved into the disk address. A test is then made to see if the maximum number of heads has been reached. If it has, the cylinder number is updated by one and the low limit head number is moved into the disk address. The maximum cylinder value is then tested. If the maximum value for the record, head, or cylinder has not been reached when tested, a branch is taken to the address in register 2.

DISKOFL - PS
This routine is executed when ATLEFC7 attempts to write on SYSLNK a record containing Autotest table entries under overflow conditions. DISKOFL writes on SYSLST and/or SYSLOG the message "DISK WORK AREA TOO SMALL" (number 9903I) and cancels the job with an SVC 6.

ERAST - PN
This routine is entered from CHCKAST when it is found that the asterisk required on autopatch cards is missing. ERAST inserts an error code 'D', signaling this condition, into the first instruction of MVCODE. It alters this instruction, causing it to place the code in the error table in the output area for SYSLST. ERAST then uses subroutines ERRTN to set up the output area for SYSLST and branches to ERPNT.

ERINST - PJ
This routine is used when, in CONAST, it is discovered that the hexadecimal parameter on a replace constant (CON) patch card is in error (for example, an odd number of characters). ERINST inserts an error code 'E' (signaling an incorrect parameter) into the first instruction of MVCODE. This causes MVCODE to place the 'E' in the error table of the output area for SYSLST. ERINST then uses subroutine ERRTN to set up the output area for SYSLST and branches to ERPNT.

ERPNT - PJ
This routine is entered whenever diagnostic procedures performed on an autopatch card have indicated a field in error. When ERPNT is executed, the error code has been entered into the output area for SYSLST, along with the card image. ERPNT uses subroutine PRINTPAT to write the card image and associated table of errors on SYSLST. READPAT is then executed to read the next card from SYSIPT. ERPNT finally tests the type of card read from SYSIPT, branching to FTCHID if it is an Autotest control card. If the last card read was a CON card, it branches to CONAST. Otherwise, it branches to CARDAST.

ERRTN - PJ
This routine sets up and writes on SYSLST a series of error codes associated with an autopatch card containing errors. ERRTN gets the address of this error table in the output area for SYSLST, and checks whether it has been filled by previous error codes entered for the card being checked for validity. If the field is full, control branches to CNCLPCH to write the card image and its error table on SYSLST. If the field still has positions available for error code entries, control branches to MVCODE to insert the present code into the table.

ERTRSVC - PK
See PNTID (PK).

E2ADROK - PS
This routine is executed when it is determined in RTNE2 that an Autotest table record is ready to be written on SYSLNK. It updates and validates the record, track, and cylinder numbers, then branches to E2OVFL1 to check for overflow.

E2OVFL - PS
This routine checks the disk address calculated in E2ADROK for overflow. If overflow exists, a branch is taken to DISKOFL. Otherwise, the record is written on SYSLNK and its address moved to the Autotest communication region. E2OVFL then branches to the address in register 3.

E2OVFL1 - PS
This routine, entered from E2ADROK, moves the address arrived at in the previous routine into the count key field to prepare it for testing against overflow. It then branches to E2OVFL to perform this test.

E2PT1 - PR
The test-request table area is checked for zeros. If it is not zeroed, control branches to E2ADROK.

FTCH - PC
FTCH uses the subroutine WRITERTN to write the core image library instruction on SYSRES, and then fetches phase ATLEFC2 at entry point READCARD.

FTCHID - PE
This routine fetches ATLEFC2 at entry point CARDID.

HEADLINE - PK
After a skip to a new page has been made, this routine sets up the output area for SYSLST to write a new header line. It moves into proper position a series of titles identifying the following fields: SVC for the identifying SVC link assigned to an Autotest table entry; TR for the number given a particular test request; "CARD-IMAGE" for the autopatch or test-request card processed; "PPA/TPA" for the patch-point or test-point address; and "ERROR/S" for a series of error codes pointing out omissions or errors in the pointing out omissions or errors in the card entries. HEADLINE, when it has finished formatting these titles, branches to PRINTPAT.

INITINST - PN
INITINST is entered from CHCKAST when that routine has determined that the required asterisk on a patch card is present. INITINST checks to see if the field following the asterisk is blank. If it is, a branch is taken to ERINST to handle this error. Otherwise, INITINST loads register 8 with the address of the input area, clears register 4, and exits to the address in register 7.

INPTAB - PD
This routine sets TABSWI to indicate that an autopatch table entry will be created. It gets the storage address for the next entry to be created, initializes code X'30' (a constant indicating that this table entry will be an exchange or add patch) and then uses the PATCH subroutine to insert this code into the autopatch table entry. INPTAB then branches to PATINST.

INSTCHK - PN
This routine is used by routine CARDAST to load register 2 with the address of the user's instruction to be checked for a valid patch instruction.

IORTN - PM
This subroutine reads core image library records from SYSRES, and when finished, branches to the address in register 7.

MISPCC - PF
This routine is entered from ADDCONT when the card just read is found to be neither an Autotest control card nor an autopatch card. MISPCC signals this error condition by inserting "PCC MISSING" in the error code field, and branches to ABORT1.

MVCODE - PJ

This routine is entered from ERRTN when it has been determined that the error table in the output area for SYSLST has available space for an error code associated with the current autopatch card. MVCODE moves this code into the first available position, updates the field address to the next position, sets a switch (ERRSWI) to value X'80' to indicate an error in the current card, and branches to the address contained in register 12.

NOPDISPL - PE

This routine loads register 3 with the address of NO-OP instructions that will be added to an autopatch table entry built for an exchange card. On an exchange, there is no need to save the core image library instruction overlaid by the linking SVC or the exchange patch itself. Accordingly, the area in the table entry reserved for saving this instruction is filled with the NO-OP's. NOPDISPL then branches to NOPEXC.

NOPEXC - PE

This routine uses the PATCH subroutine to insert into an autopatch table entry the instruction whose address has been previously loaded into register 3. If the autopatch card associated with this table entry is an add, register 3 points to the address of a core image library instruction displaced by a patch at that point. If the card requiring this table entry is an exchange, register 3 points to the address of the NO-OP instructions used to fill the table entry. NOPEXC then branches to SVCINIT.

OKTRACK - PA

This routine performs the final function of CHKCNTRL. When that routine has finished updating the disk address of the Autotest table, OKTRACK resets the record number to zero and loads the Autotest table address into the Autotest communication region. It then branches to SYSC5.

PATCH - PL

PATCH fills the work area in which autopatch table entries are accumulated until enough are present to be written on SYSLNK. It tests first to see whether the work area is filled. If it is, it branches to TABREC1 to write on SYSLNK the accumulated entries. If not, it gets the addresses of the entry character to be moved and of the position of the work area allocated to it. It loops through a process of moving characters and increasing addresses until the complete table entry is in the work area. It then stores the address of the next position available in the work area, turns ERRSWI off, and branches to the address contained in register 7.

PATCIL - PE

This routine continues building an autopatch table entry. It determines the length of the core image library instruction, gets the next available address for an entry in the Autotest table, and tests CARDSWI to see whether an add or an exchange card is being handled. If it is an exchange, control branches to NODISPL; if it is an add, the core image library instruction has to be inserted into the table entry. The address of this instruction is put into register 3, and control branches to NOPEXC.

PATINST - PD

This routine adds the length of the user patch instruction and the length of the present core image library instruction. This sum is part of the calculation of the number of bytes needed for this particular autopatch table entry. After initialization, it uses the PATCH subroutine to insert the exchange or add entry into the Autotest table. It then tests to determine whether it has been handling an add or an exchange, branching to ADDCONT if an add; branching to PATCIL if an exchange.

PNTHD - PK

This routine is entered when SYSLST has written a complete page of information and must skip to a new page. PNTHD zeroes out the line counter for SYSLST, and branches to HEADLINE to write a new set of header titles.

PNTID - PK

This routine calculates the number of the identifying SVC link to be assigned to an Autotest table entry. The number is originally set to its highest limit of 254, and as table entries are required, this original value is decreased by one until the value X'36' or decimal '54' is reached. This indicates that the Autotest table area is exhausted and no more entries are allowed.

PNTID gets the number of the last SVC assigned to an entry, decreases it by one, and passes control to RTRNID. It tests for a X'36' to see if all valid table entries have been used. If so, control passes to ERTRSVC; if not it branches to CLRSAVE.

ERTRSVC inserts an error code of 'G' into the output area for SYSLST to signal no more table entries, and exits to ABORT1.

CLRSAVE takes the valid identifying SVC link number, converts it to print format and moves it to the output area. CLRSAVE then branches to the address in register 7.

PRINTPAT - PK
This routine writes the contents of the output area on SYSLST. It does this by getting the address (held by the DTF) of the logic module. PRINTPAT branches and links to this module, treating it as a subroutine. It clears its output area, increases the line counter, which determines whether a full page has been written on SYSLST, and tests this counter to see whether a full-page condition has been reached with the writing of the latest line. If a full-page condition has occurred, PRINTPAT branches to PNTHD to write a header; if not, it sets the print code to cause SYSLST to space one line before writing another line of data, and then branches to the address contained in register 7.

PTCIL24 - PD
This routine initializes instructions in PATINST and in PATCIL to handle the case of an exchange patch involving a core image library instruction and a replace instruction of different lengths. It does this by altering the constants in the first instruction of both routines, then branches to INPTAB.

RDREG - PR
This routine takes the address calculated in CONTRANS, and loads it into register 14. Register 14 now points to the address of the constant to be converted to hexadecimal format, and register 8 has the number of characters in that constant. With these, RDREG executes the translate instruction labeled TRANSCON. TRANSCON translates the constant into hexadecimal format, and then RDREG branches to the address in register 7.

READPAT - PM
This subroutine is executed whenever a card is to be read from SYSIPT. READPAT calculates the address of the read-card logic module generated by the DTF for SYSIPT, branches and links to the module to read a card, and returns to the address stored in register 7.

READREC - PM
This routine reads a physical record. It allows entry back to the physical read without looping through the initialization of address and registers.

READRTN - PM
This routine reads in from SYSRES the next core image library record. It first updates all affected addresses, then uses IORTN to perform the actual read. It then checks to see whether the record just read finished up the number of records requested. If not, it branches to get another record. Otherwise, it restores needed registers to their values at the beginning of the routine, and branches to the address in register 10.

RTNCHECK - PA
See START1 (PA).

RTNE2 - PR
See E2PT1 (PR).

RTRNID - PK
See PNTID (PK).

START - PA
This is the initialization routine for ATLEFC7. It initializes the base registers, lists the entry points in the phase, and branches to STRT1.

STDEC - PN
This routine performs the last function of VALIDHEX. It tests to see if a CON card is being processed. If so, control goes to UPREG. Otherwise, the current character is stored. A check is then made to ensure that no more than 12 characters have been processed. If more than 12 characters have been processed, control goes to ERINST. Otherwise, the storage area address is increased and control goes to UPREG. At UPREG the address of the character to be checked and the length counter are increased and control goes to VALIDHEX to check the next character.

STRT1 - PA
This routine, one of three entry points into ATLEFC7, checks CARDSWI (set in phase ATLEFC2) to determine the card type being processed. If CARDSWI is set to value X'01', STRT1 branches to ADDRTN to process the card as an add patch to the user program. If it is set to the value X'02', it branches to CONRTN to process the card as a replace constant patch. Otherwise, it branches to CILEXC and begins processing the card as an exchange patch.

SVCIL - PF
This routine is entered when the core image library instruction involved in a patch entry is to be saved. SVCIL tests first to see whether the core image library I/O area has been filled. If it has, it branches to TAPEIOSV. Otherwise, it performs a process of moving instruction characters and updating the address of the character to be saved and the position of the work area to receive it. When the entire instruction has been moved, SVCIL checks ENDSWI for a value X'01'. If it has this value, the instruction has exceeded record boundaries. If not, SVCIL branches to the address in register 7. If the boundary has been exceeded, SVCIL branches to COMPLETE to finish inserting the instruction into the core image library record.

SVCINIT - PE
This routine continues building an autopatch table entry. It fills out the three remaining fields in the entry, and writes the completed Autotest table record on SYSLNK. The three fields are:

1. The special Autotest SVC of 255 to signal the end of the table entry;

2. The length of the table entry;

3. The SVC number identifying this particular table entry (this number has been calculated in PNTID).

SVCINIT uses subroutines TABLNTH to calculate the length of the table entry, subroutine PATCH to insert the field into the entry, and WRITERTN to write the completed Autotest table record on SYSLNK.

At the end, SVCINIT tests whether CARDSWI indicates that an add card is being processed. If it is, control branches to ADDEND. Otherwise, register 5 is cleared and control branches to FTCH.

SVC15 - PA
This routine is one of three entry points into ATLEFC7. It is used to write the last Autotest table record on SYSLNK. Bit 0 of E2SW1 is turned off to indicate that this record is an autopatch table entry. TABSWI is then interrogated to see whether any entries have been entered into the Autotest table. If none have been entered, SVC15 branches to SYSC5 to update the Autotest communication region. Otherwise, it uses RTNE2 to write the record on SYSLNK and branches to CHKCNTRL.

SYSC5 - PA
This routine enters into the Autotest communication region the number of the lowest linking SVC used and the length of the Autotest table.

It also loads register 8 with the address of the card input area and register 10 with the address of ERRSWI. It then fetches ATLEFC5.

TABLNTH - PL
This routine is entered and used by SVCINIT. Its function is to update the length of the current table entry and to update the current length of the entire Autotest table. The length of the current entry is stored in LENPAT, the current length of the entire table is stored in LENTAB and register 4 is set to value 2, the length of the SVC link to be inserted. Register 3 contains the address of the table entry length. TABLNTH then branches to the address in register 7.

TABREC - PL
This routine performs the initialization for writing the work area built in routine PATCH on SYSLNK. TABREC sets E2SW1 to indicate an autopatch table entry is present in the Autotest table, and moves the work area to the SYSLNK output area. RTNE2 is then used to write the record on SYSLNK. TABREC then restores the address of the work area to its starting point, clears the area to zeros, and branches to PATCH to see if any more characters in an entry are to be moved to the work area.

TABREC1 - PL
This routine is entered when the work area, provided for accumulating patch table entries before writing them on SYSLNK, is filled. When this occurs, TABREC1 sets ERRSWI to indicate that additional information remains to be inserted in the Autotest table. TABREC1 then branches to TABREC.

TAPEIOCL - PG
This routine is executed when it is found that the present core image library record read from SYSRES to contain new patches has been exhausted. TAPEIOCL uses WRITERTN to write this record back into the core image library, increases the record counter by one, and reads the next core image library record. Then, if ENDSWI indicates that no autopatch table entry is required, TAPIOCL branches

back to CILPATCH to continue moving the remainder of the new instruction into the latest core image library record. If ENDSWI indicates that an autopatch table entry is required for the new instruction, it branches to SVCIL to save the old contents of the core image record.

TAPEIOSV - PF
This routine is entered from SVCIL when it is determined that the end of a core image library record has been reached. SVCIL saves the displaced CIL instruction for an exchange patch. TAPEIOSV turns on ENDSWI to indicate that the instruction to be inserted has exceeded record boundaries, and saves registers 3 and 4. It then loads register 4 with the present CILADR and branches to CIL24 to patch as much as possible into this record.

TESTER - PQ
TESTER is used by both the CONAST and CARDAST routines to determine whether diagnostic functions previously performed have found any errors. TESTER tests ERRSWI for a X'80' value, signaling the preceding error condition. If no errors are indicated, TESTER branches to the address in register 7. Otherwise, it branches to ERPNT to write the card image and its error table on SYSLST.

TRANS - PP
TRANS is entered from VALIDHEX when a complete hexadecimal parameter has been checked for validity. It first checks to ensure that it is not dealing with a constant that must be entered untranslated. If it is such a constant, it branches immediately to the address in register 7. Otherwise, it translates, packs, and stores the constant in a save area. It then checks it for one of the three allowable lengths: 12, 8, or 4 characters. If the constant has a length other than one of these three, TRANS branches to ERINST; otherwise it branches to CHRCNT to check the operation code of the constant. The operation code has an implied length in it, and this must correspond with the length arrived at in TRANS.

TRANSCON - PR
See RDREG (PR).

UPREG - PN
UPREG is executed by STDEC just before branching to VALIDHEX. UPREG updates the address of the character to be checked, and increases by one the length counter of the hexadecimal parameter. It then branches to VALIDHEX.

VALIDHEX - PN
VALIDHEX is used whenever the hexadecimal parameter on an autopatch card must be checked. VALIDHEX first checks to see whether it has finished checking the whole parameter. It does this by testing for blanks and branching on an equal condition to TRANS. If there are still hexadecimal digits to be checked, it first tests for end-of-card, branching to ERINST to handle that error. If neither test caused a branch, VALIDHEX then ensures that the digit is a valid hexadecimal digit. On any error, a branch is taken to ERINST. If the character is valid, VALIDHEX branches to STDEC to initialize registers for the next branch to VALIDHEX.

WRITERTN - PM
This routine writes on SYSRES the altered core image library record just processed by ATLEFC7. It first updates all affected addresses, then uses an SVC 0 to perform the actual write. After it restores needed registers to their initial values, it branches to the address in register 10.

USER PROGRAM EXECUTION

Autotest Control Program Initialization (ATLEFE1) Charts QA-QB

ADD8 - QA
Relocates the address constants in the Autotest control program after it is loaded into storage. It modifies the address constants by adding the displacement factor of the address constant to the load address of the Autotest control program. It loops through this routine until all eight address constants are modified.

CORCHK - QA
Tests for all of main storage allocated to Autotest. In a multi-programming environment, the use of Autotest is restricted to background jobs only. If the user attempts to run in foreground mode or in background mode with any foreground job in operation, the Autotest job is canceled.

EQLSTLOG - QA
Cancels the job.

HANDREM - QB
Stores the length of the last Autotest table record (if the last record was not 200 bytes length) in the read CCW and sets the switch at PATCHRD to indicate all Autotest table records are processed.

NOTABLE - QB
Turns on the Autotest processing switch in the system communication region and fetches $$BATST1 to set up the interface between the Autotest control program and the supervisor.

PATCHRD - QB
Tests to ensure all the Autotest table records have been read from SYSLNK into main storage.

PATCHRDS - QB
Reads the Autotest table records into main storage. Each full Autotest table record is 200 bytes long.

RDDISK
Listing Only.

Entry point to the subroutine to read the Autotest communication record and the Autotest table records from SYSLNK.

RESETRCD - QB
Resets record count to one and checks the track count.

RESETTRK - QB
Resets track count to upper limit for split cylinder and subtracts one from cylinder count.

START - QA
Beginning of the Autotest control program initialization phase (ATLEFE1). Initializes the base register and opens SYSLST and SYSLNK.

Set Up Interface Between Autotest Control Program and Supervisor ($$BATST1) Chart RA

FIRST - RA
Beginning of routine to modify the ADCON's in $$BATST1.

LOWER - RA
Loads first user phase into the problem program area. It then builds a fetch record for the first phase of the user's program and writes it on the test-request output file on SYSLNK.

NXTADCON - RA
Updates the ADCON's in the disk CCB and the CCW's for $$BATST1.

TEST256 - RA
Clears ATLEFE1. Clearing is done in blocks of 256 bytes.

Autotest Control Program Execution (ATLEFE2) Charts SA-SF

ATSVC05 - SB
Indicates beginning of routine to search the Autotest table to find matching SVC number. When a match is found, the requested Autotest service (either a test request or autopatch) is performed.

ATSVCRTN - SB
Indicates the beginning of the routine to process a valid Autotest SVC.

DISKOP - SF
Indicates beginning of subroutine to write a record on SYSLNK.

DKINCR - SF
Indicates beginning of routine to update the disk address. If the maximum value for the field being updated has been exceeded, that field is reset to its low limit and the next field is updated. DKTABL is the field used for updating purposes. It contains the maximum values and the reset values of the field to be updated. See DKTABL for further information.

DKTABL
Listing only.

This is an 8-byte field containing maximum values and low limit values for disk address updating. Its format is:

```
CHHRCHHR   where:  C = Cylinder
                   H = Head
                   R = Record
```

```
The maximum
values are:  C = CA
             H = 9
             R = 20
```

DPLMOV - SE
Moves the display data to the output area, IOAREA.

DPLRTN - SE
Indicates beginning of routine to write display information on SYSLNK for later post-processing by ATLEFF1. All the display data between the display limits is written on SYSLNK in 117-byte records.

EXIT - SB

EXITNCR - SF

FCHSVC - SB
Moves phase record code of F and phase name to output area, IOAREA. Writes phase fetch or load record on SYSLNK.

LOADSVC - SA
Moves load address, phase name, and load record code of L to output area, IOAREA.

NOMOVE - SB

OVERFLOW - SF
Indicates disk work area overflow condition. It moves the error code of FA to TYPDMP in the Autotest communication area.

PCCTST - SC
Tests to see if the action count specified in the PCC (program control) card has been exceeded.

PCHRTN - SD
Tests for a multiple test-request service at a single test-point address. Multiple test requests can be given by means of continuation cards.

PNLRTN - SE
Indicates beginning of routine to write bytes 0-127 of main storage, the contents of the general registers, and of the floating-point registers (if requested) on SYSLNK. Each record is 117 bytes in length.

PNLWRT - SE

RESET1 - SF

RESTALIM - SC
Stores the updated action count in ACTLIM in the Autotest communication area.

RESTON - SD
Stores updated BEGIN number for a conditional regulation of test-request execution in the Autotest table entry. This conditional regulation of test-request action is called the ON condition. The ON condition is specified by three integers which serve as comparison values for a counter set up by Autotest. These three integers can be interpreted as BEGIN, END, and EVERY, respectively. For example, the ON parameters 2, 8, 3 specified on a test-request control card are interpreted as follows: When the test-point address has been reached twice, the test-request action is performed for the first time (BEGIN). From then on, the action is performed every third time (EVERY) until the eighth time (END), after which no more actions are taken.

The BEGIN and EVERY parameters are stored in the Autotest table entry along with the ON number, which is calculated according to the following formula:

$$\text{ON number (number of actions)} = \frac{\text{END-BEGIN}}{\text{EVERY}} + 1$$

ON parameters can be specified for all test requests at a certain address within the program and/or a specific test-request function.

The ON parameters for a specific test-request function override the ON parameters specified for the test-point address.

RETURN - SB

RTRNUSR - SA
Restores register 15 and exits to the address in the new SVC PSW in low storage.

STRT1 - SF
Points to the disk field to be updated.

SUPVEXIT - SA
Indicates the beginning of the routine to save the address of the invalid SVC in the Autotest communication area. Exit is made to the supervisor to handle the interrupt.

SVCINTPT - SA
Indicates the beginning of the Autotest control program (ATLEFE2). Stores the contents of general register 15 in byte 0 of main storage and then uses register 15 as its base register. It determines the origin of the SVC interrupt and processes accordingly.

TBLOOP01 - SC
Loop used to determine if the end of the Autotest table has been reached.

TBLOOP05 - SC
Loop used to find the next entry in the Autotest table.

TRLOOP - SC
Indicates beginning of loop used to point to next sub-entry within the Autotest table when there are multiple test requests at a test-point address.

TRQST - SD
Indicates beginning of routine to move the test-request record code and number to the output record. It turns on the EXTRAD (X'01') switch in the Autotest communication area to indicate that test-request output is present.

TRRTN - SE
Tests to determine which test-request service has been requested. It could be either panel, display, or display and panel.

TRRTN05 - SE

TRQST - SD
Indicates beginning of routine to build test-request output record for use by ATLEFF1.

WRTFLM - SF
Writes a file mark to indicate the end of the test-request file on SYSLNK. If an overflow condition occurs, this file mark replaces the last test-request record written on SYSLNK.

<u>Dump User Program ($$BATST3) Charts TA-TC</u>

ANOTHREC - TB
Keeps track of the main storage address of the next record to be written on SYSLNK.

BATST320 - TA
Indicates beginning of $$BATST3.

CRMOVE - TC
Moves Autotest communication region to beginning of problem program area. It stores the address of last byte written on SYSLNK in the Autotest communication region and develops a base address for the small 155-byte EOJ dump record. If there was an Autotest cancel, it changes the system-cancel code to reflect the reason for the Autotest cancel.

CRSIZCHK - TA
Determines whether the amount of main storage to be dumped is 32K or greater.

DISKWRT - TB
Writes core dump records on SYSLNK in record lengths of 1728 bytes.

LRGDUMP - TA
Calculates the number of records to be dumped, the length of the last record to be dumped, and the starting address of the dump. It stores the starting address in count field and seek field.

MVADDR - TB
Supplies address for the write CCW.

NOFLOAT - TC
Moves label length to output area and sets up the CCB and CCW's to write the 155-byte EOJ dump record on SYSLNK.

RECBLD - TC
Builds the 155-byte dump record in the output area. This small record is written on SYSLNK as the second record on track 1 of the Autotest work area.

REPTRKNO - TB
Stores new track number in seek field and count field after every other record is written on SYSLNK.

RESETTRK - TB
Resets track number to low limit and increases cylinder number by one, if there is a cylinder overflow.

SMALL - TA
If the amount of main storage to be dumped is less than 32K, the length field for the amount of storage to be dumped is set to 16K.

START - TA
Calculates address of Autotest communication region and stores beginning address of logical-transient area for supervisor dump. Saves cancel code for use by Autotest, then resets the cancel code and the Autotest switch in the system communication region. It checks for normal EOJ and determines if an EOJ dump is required.

SW1 - TB
Program Switch.

Turned on after last full-size record is written on SYSLNK and there is a partial record to be written. When the switch is on, it causes a branch to write the small 155-byte record on SYSLNK.

SW2 - TB
Program Switch.

During a core dump, two records are written per track. SW2 is tested after each record is written and it is reset each time it is tested.

If the switch is off, it signifies the first record on **the** track **was** just written and the track number of the CCW's do not need updating. If the switch is on, it signifies the second record of the track was just written and the track number must be updated by one, if a cylinder overflow has not occurred. (If a cylinder overflow occurs, see label RESETTRK.)

TRCOMPCK - TB
Tests to determine if the last record written was the last record of the cylinder. If it was not, it updates the track number in the CCW. If it was the last record of the cylinder, see Label RESETTRK.

UPIT - TA
Updates the address of the CCW's, the disk CCB and the data address of the CCW's.

POST PROCESSING

Test-Request Output (ATLEFF1)
Charts UA-US

BACKA - UA
Start of routine that reads the test-request control records into main storage. It clears the work table and determines if a display and/or panel is requested.

BEGIN - UA
Initializes register 9 as base register 1, register 11 as base register 2, register 14 as base register 3, and register 13 with the address of Autotest communication area.

CHAN1 - UP
Beginning of loop to convert one floating-point register at a time to floating-point long or floating-point short.

CHAN2 - UP

CHAN3 - UP
Sets up register 5 as a loop counter to count the number of registers to be converted to floating-point short or float-point long.

CHAN4 - UP

CHAN5 - UP

CHAN6 - UP
Entered when there are more floating-point registers to be converted to floating-point long or short. It updates the input area pointer and branches to convert another floating-point register.

CHAN7 - UP
Tests to see if all four registers have been converted to floating-point long or floating-point short. If yes, write a line on SYSLST.

CHAN8 - UP
Tests the floating-point-long switch. If the switch is off, it indicates that only a floating-point-short panel has been written. It turns on the floating-point-long switch and branches back to convert the floating-point registers to floating-point long. If the switch is on, it writes a line of floating-point-long information on SYSLST.

CHAR - UG
Unpacks and translates 32 bytes to printable characters.

CHSUB - UG
Beginning of routine to move 32 converted characters to the print area and to test to see whether the address of the display should be moved to the print area.

CNVT7 - UK

CONVT - UG
Tests for character-only display. If it is a character-only display, it converts 32 bytes to printable characters. If not, it initializes the loop counter and registers to convert 32 bytes to hexadecimal characters.

CVADR - UK
Beginning of subroutine to convert the display address for each half of the print line to hexadecimal.

DISPIN - UA
Moves the display limits from the input area to the work table.

DSPEND - UF

EDIT - UM
Tests for fixed-point halfword or fullword display and initializes loop counter to 4 for a halfword display and to 9 for a fullword display.

EDITA - UM
Stores the negative sign of the halfword converted to decimal. The negative sign is moved to the decimal area after the high-order zeros have been suppressed.

ENDDSP - UG
Writes a line on SYSLST and turns off the first line switch and the skip-right-side switch. It then determines if a character or mnemonic print line is required along with the hexadecimal line just written.

EQFUL - UF

FIN - UM
Moves the sign of the decimal field and the decimal field itself to the decimal save area, after the high-order zeros for the decimal field have been suppressed.

FIXPH - UD
Initializes the print counter to zero. This indicates the number of characters that have been moved into the print line. A test is then made to determine if the display will be fixed-point fullword or halfword.

FLTDL - UC
Sets up the link-register address and stores it in the save area.

FLTDS - UC
Sets up the link-register address and stores it in the save area.

FOUR - UK

FPOVR - UP
Tests to see if all four floating-point registers have been converted. If they have, a branch is made to write a line.

FPRTN - UP
Beginning of subroutine to dump floating-point registers.

GRRTN - UQ
Beginning of subroutine to write the contents of the general registers, when a general-register panel is requested.

HALF2 - UJ
Entry point for the MVCONV routine, which moves characters into the right half of the display-print line.

ISH2 - UB
Initializes CCW for reading test-request output records from SYSLNK.

LOOI - UQ
Initializes register 3 to four, the number of bytes in a register to be converted to hexadecimal.

LOOPD - UQ
Points register 6 to the disk input area, points register 2 to the print area, and initializes register 4 to eight (the number of registers to be printed per line).

LOOPER - UJ
Moves eight characters at a time into the print area.

LOOPF - UR
Converts the characters in the CSW (channel status word) byte count and the CAW (channel address word) key to hexadecimal characters, and moves this information to the print area.

LOOP1 - UP
Converts eight characters from the floating-point register to hexadecimal, one character at a time, and moves the characters to the print area.

LOOP10 - US
Converts the instruction length of the ten PSW's in storage positions 24-127 to decimal and writes the instruction-length line of the panel on SYSLST.

LOOP11 - US
Converts the condition code of the ten PSW's in storage positions 24-127 to decimal and writes the condition-code line of the panel on SYSLST.

LOOP12 - US
Converts program mask bits of all ten PSW's in storage positions 24-127 to printable characters and writes the program-mask line of the panel on SYSLST.

LOOP14 - US
Unpacks and translates to hexadecimal the instruction address of all ten PSW's in storage positions 24-127 and writes the instruction-address line of the panel on SYSLST.

LOOP15 - US
Unpacks and translates to hexadecimal the last line of the panel. This line consists of the system timer, system time-of-day data, plus four bytes of unused data.

LOOP2 - UQ
Converts one byte of a register to printable hexadecimal characters and moves the two characters to the print area. It then tests to see if all four bytes of the register are converted. If they are, and all eight registers of the print line have been converted, it writes a line on SYSLST, and branches to set up the second line of the general register panel.

LOOP3 - UR
Converts the three bytes of the CSW (channel status word) command address to hexadecimal and moves the converted characters to the print area.

LOOP3B - UR
Converts the command address of the channel address word to printable hexadecimal characters and moves these characters to the print area. The channel status word and channel address word line is then written on SYSLST (with a double space before and after this line). The two header lines for the PSW contents are then written.

LOOP4 - UR
Converts the system-mask bits of all ten PSW's in storage positions 24-127 to printable characters and writes the system-mask line of the panel on SYSLST.

LOOP6 - UR
Converts the protection key of the ten PSW's in storage positions 24-127 to a printable hexadecimal character. It then writes the protection-key line of the panel on SYSLST.

LOOP8 - US
Converts each bit of the AMWP field of the ten PSW's in storage positions 24-127 to a printable character and writes the AMWP line of the panel on SYSLST.

LOOP9 - US
Unpacks and translates to hexadecimal the interrupt code of the ten PSW's in storage positions 24-127 and writes the interrupt-code line of the panel on SYSLST.

MAYBE - UF
If it is determined that less than 32 characters remain to be processed, this routine determines whether another record should be read. If the number of characters remaining in the record is less than the number of characters remaining to be displayed, a new record must be read.

MINUS - UM
Stores the positive sign of the halfword converted to decimal. It moves the positive sign into the decimal area after the leading zeros are suppressed.

MNLINE - UH
Converts EBCDIC characters to mnemonic for a mnemonic display.

MODBH - UR
Converts status bytes of CSW to printable binary characters, zero or one.

The conversion is accomplished by shifting one bit from the status byte into the units position of register 7 and changing this bit to a hexadecimal character (zero or one), using the hexadecimal conversion table.

MODCNT - US
Tests to see that all three fields of the last panel line are converted to printable hexadecimal characters, and then writes the last line of the panel on SYSLST.

MORPAK - UG
Loop that converts 32 bytes to 64 printable hexadecimal characters.

MOVE - UM
Moves one character at a time, starting with the high-order character of the decimal field, to temporary storage and tests to see if the character is a zero. If it is a zero, the zero is cleared from the decimal field and the next character is tested. This loop continues until a significant character is found. The decimal field, with high-order zeros suppressed, is moved to the decimal save area.

MOVECH - UF
Moves 32 bytes from the input area to temporary storage and tests to see whether there are less than 32 bytes remaining in the record. The test is made by comparing the input-pointer address to the address of the input area plus 85. If the input pointer is pointing beyond the 85th byte, less than 32 characters remain in the record.

MVAD2 - UG
Moves the right-side address to the print line.

MVCONV - UJ
Beginning of subroutine to move the characters to the print area, one side at a time.

MVSAVE - UF
Moves 32 characters from temporary storage, GOT32, to print area.

NOFUL - UF

NXTMN - UH
Converts one byte to its mnemonic op code. The conversion is made by multiplying the byte by four and adding the address of the mnemonic table to find its location within the table. Only every other hexadecimal character of the print line can be converted to mnemonic because of the limited space in the print line.

OKAY - UF
Tests SW8 to see whether the starting address of the display is odd.

OTRTN - UB
Reads a test-request output record from SYSLNK and tests for end-of-file. If end-of-file is reached, ATLEFG1 is fetched. If not, processing continues.

OUT - UJ
Sets SW8 to X'01' to indicate the right-half of the print line of the display is not needed. This occurs at the end of the display when the last line is 32 characters or less.

PANELS - UR
When a panel of low core is requested, this subroutine converts the data from core positions 24 through 127 to printable characters, sets up the header lines for these characters, and writes the header lines and converted data on SYSLST.

PANEL5 - UR
Moves the constant 'SYSTEM MASK BIT' to the print area as the header for the system mask line of the panel.

PANEL6 - UR
Moves the constant 'PROTECTION KEY HEX' to the print area as the header for the protection-key line of the panel.

PA1 - UE
Tests to see if a floating-point register panel is requested. If it is, a branch to the routine to dump floating-point registers is made.

PA2 - UE
Tests for a general-register panel request. If it is requested, a branch to dump general registers is made.

PA3 - UE
Tests for a low-core panel request. If it is requested, a branch to dump low core (storage positions 24-127) is made.

PA4 - UE
Tests for a display request. If a display is requested, it reads a record from SYSLNK.

PCONVT - UF
Tests to see if the prior line was printed.

PLS2 - UA
Sets up the cylinder address for the CCW search field.

PRTFP - UP
Resets I/O area pointer and loop counters. It writes the contents of the floating-point registers in hexadecimal characters and sets up to double space before writing next line on SYSLST.

RDREC - UF
If there are less than 32 characters remaining in the display-record hold area, GOT32, and the display has 32 or more characters left to be written, this routine reads another record and moves the number of bytes required to complete the 32-byte GOT32 area to the print line.

RESET - UH
Tests for the last record of the track after a record has been read. If it is the last record, the record number is set to 1 and a test is made for end-of-cylinder. If it is end-of-cylinder, the head number is set to its lower limit, and cylinder number is updated by one. If it is not the last record or end-of-cylinder, the record number or the head number is increased by one.

RTNA - UK
Beginning of subroutine that unpacks and translates to hexadecimal the left-side and right-side addresses for a print line.

This routine is executed for the first line only of a hexadecimal, character, or mnemonic display. The addresses for all other lines of the previously mentioned displays and all lines of a fixed-point or floating-point display are converted by the CVADR subroutine.

RTNA1 - UK
This routine is executed when the address for the first line of the display is being converted to hexadecimal and the display format is other than hexadecimal, hexadecimal and character, character, or hexadecimal and mnemonic. It moves the left-side address of the first line to the print area and stores that address.

RTNA3 - UK
Moves the start address of the display to the print area, and tests for an odd-numbered starting address. If the start address is odd, it is decreased by one, so that the display will start on an even address.

RTNB - UQ
Beginning of subroutine that tests for the end of the test-request record. If it is the end of the record, a new record is read and the input pointer, register 6, is reset to zero to point to the first data byte of the new record.

RTNBE - UQ

RTND - UF
Beginning of subroutine that prints hexadecimal, character or mnemonic displays. The first part of the routine tests for an odd starting address. If the address is odd, a blank plus 31 data characters are written on the first half of the first line.

RTNF - UB
Clears the I/O buffer area, multiplies the test-request number by 11, and adds the test-request table address to it to find the table location which contains the information required to process the test-request output record.

RTNFD - UB
Gets the display parameters from the record, calculates the start and end addresses and the number of bytes in the display. It also tests for the type of format requested for the display printout.

RTNFL - UK
Subroutine that converts floating-point to decimal.

RTNG - UH
Subroutine that reads a record from the Autotest file, tests for end-of-file, and updates the pointer to the next record to be read.

RTNH - UB
Beginning of routine that converts the test-request number and the ON BEGIN number to decimal, and moves this information, along with the test-request header constants, to the print area. It writes on SYSLST the header line with a double space before and after the header line.

RTNHB - UJ
Subroutine to convert the binary value in register 4 to hexadecimal characters.

RTNHEX - UQ
Subroutine to convert one packed-decimal byte to two printable hexadecimal characters.

RTNI - UD
This routine is entered for a fixed-point halfword display. It stores one byte of data in the TESTH area for binary-to-decimal conversion.

RTNIE - UD

Turns on the end-of-display switch, writes a line on SYSLST and branches to read another test-request output record.

RTNIPT - UA

Reads an input record and determines if it is the end of the test-request control file. If not, it stores the number of logical records in the physical record in register 6 and builds the test-request table in main storage. If it is the last test-request control record, a branch is taken to process the output of test requests.

RTNI2 - UD

Tests for the end of the logical record. If it is not, it stores one byte in TESTH+4 for binary-to-decimal conversion, updates the pointer to point to the next input character, and tests again for end of logical record. It then converts two bytes from the TESTH field to decimal, moves the characters to the print area, and tests for the last character of display.

RTNI3 - UD

Writes a line only if end-of-display or end-of-print-line occurs. If neither condition occurs, a branch is taken to convert two more bytes to decimal.

RTNI5 - UD

Updates the input area pointer to point to the next character and tests for end-of-display.

RTNJ - UM

Converts binary to decimal and unpacks a halfword.

RTNJ1 - UM

Beginning of subroutine that suppresses all zeros to the left of the first significant digit in the fixed-point display.

RTNJ2 - UM

RTNJ3 - UM

RTNK - UL

Beginning of subroutine that sets up the print line for the fixed-point display and writes the fixed-point display on SYSLST.

RTNKH - UL

If a fixed-point halfword display is requested, this routine tests for end-of-print-line. If it is the end-of-print-line, it writes a line. If it is not, it calculates the print position of the next halfword of the display.

RTNK1 - UL

Writes a line of fixed-point display, and branches to convert the addresses for the next print line.

RTNK2 - UL

RTNL1 - UC

Converts four bytes to floating point and moves the bytes to the print area.

RTNL2 - UC

Initializes loop counter to 4, the number of bytes to be converted for short floating-point display, and places the address of the conversion field in register 4.

RTNL3 - UC

Writes a line on SYSLST when the print line is full, converts the address of the next print line to hexadecimal, and moves the new address to the print area. It also writes a line if it is the end of the display and the print line is not full. It then branches to read another test-request output record.

RTNL3P - UC

Sets up the print area pointer for the next floating-point word and tests for a full print line. If the print line is full, it branches to write a line. If the print line is not full, a test is made for the end of the display. If it is end of display, the end-of-display switch is turned on and a line is written. If it is not the end of the print line and not the end of the display, a branch is taken to convert another floating-point register.

RTNL4 - UC

Clears registers 5 and 10.

RTNL4F - UC

Sets format-floating-point switch to indicate floating-point-short display.

RTNL5 - UC
Clears registers 5 and 10.

RTNL5F - UC
Sets the format-floating-point switch to indicate floating-point-long display.

RTNL6 - UC
Initializes loop counter to 8, the number of bytes to be converted for floating-point-long display and places the address of the conversion field in register 4.

RTNL7 - UC
Converts eight bytes to floating point and moves the eight bytes to the print area.

RTNL8 - UC
Same as label RTNL3.

RTNL8P - UC
Same as label RTNL3P.

RTNQ - UD
The contents of the link register are stored and used to return to this address after testing to see if the print line is full or if it is the end of the display.

RTNQE - UD
Turns on SWEND to indicate the end of the display. End of display occurs when register 8, which contains the number of bytes left in the display, reaches zero.

RTNQ1 - UD
Stores one byte in the conversion area for conversion to floating-point-long and then tests for last character of the display.

RTNQ2 - UD
Reads another test-request output record, if end-of-record is reached. It then tests to see if a fullword has been moved to the conversion area.

RTNQ3 - UD
Register 3 points to the display byte and it is updated by 4, the number of bytes that have been moved to the conversion area. Register 2 is cleared and the contents of registers 3 and 6 are stored to clear the registers for use as work registers.

RTNQ5 - UD
Converts a fullword to binary, suppresses the leading zeros, and moves the information to the print area. If the end of the display is reached, a line is written and a branch is taken to read another test-request output record. If the end of print line is reached, a line is written and a branch is taken to convert another fullword. If neither end-of-display nor end-of-print-line is reached, a branch is taken to convert another fullword.

RTNS - UK
Calculates the number of bytes in the display by subtracting the start address from the end address.

RTPNL - UE
Reads a test-request output record from SYSLNK and determines the type of panel requested (floating-point, general registers or low-core). It then determines whether a display is requested. If it is, another record is read. If not, a branch is taken to START.

RTPRT - UL
Tests for end of the page and sets up a skip to channel 1 if it is the last line of the page. It writes a line, clears the print area and sets up to space for next print line.

RTPRTA - UN
Beginning of subroutine to write the address and the word 'DITTO' if the current print line does not compare equal to the last ditto line.

RTPRTB - UN

RTPRTB1 - UN
Saves the current print line so that it can be compared to the next print line and then branches to print the current line.

RTPRTC - UN
Compares the current print line with the previous print line. If they are equal, the print area is cleared and no writing occurs. If they are not equal, a test is made to see if the last line was a ditto line. If the last line was not a ditto line, the current line is saved for comparison to the next line and written on SYSLST. If the last line was a ditto line, the address of the last line and the word 'DITTO' are written.

RTPRTD - UN
Clears print area and initializes switches if the current print line is the same as the last print line.

RTPRTG - UN
Tests SWDT1 to see whether last print line was a ditto line.

RTPRT1 - UL
Saves the contents of registers 1 and 4 and link registers. Writes a line on SYSLST. Restores registers. Sets up to space one line and turns on duplicate-line switch.

RTX1 - UQ
A loop that converts one byte of packed decimal to two printable hexadecimal characters. The conversion is accomplished by inserting one character into register 4 and shifting right to move four bits into the low-order position of register 5. The address of the hexadecimal table plus the displacement of the character in register 5 points to the correct hexadecimal character stored in the temporary storage area.

START - UB
Initializes registers, switches, and I/O area before reading a test-request output record.

STOP - UB
This routine fetches phase ATLEFG1.

STRG - UP
Stores the contents of registers 2 through 8 and register 10 before entering the routine to convert floating-point registers. These registers are then used as work registers.

STRTRT - UA
Fetches a logical B-transient, $$BOPEN, to open SYSLST. Gets upper and lower track limits of work file from Autotest communication region, and initializes register 1 to point to system communication region.

SUBSET - UG
Converts the address of the left side of the display to hexadecimal characters and moves this address to the print area.

TCOUNT - UJ
Tests to see if all the characters of either half of the print line have been moved into the print area.

TESCUR - UF
Tests to see if the current line is less than 32 characters.

TH2 - UG
Tests to see if the right half of the display is required. If not, it branches to write the left half of the display. If the right half is required, it sets up the print area for the right side and tests to see if the address is required in the print line. If the address is required, it converts the address to hexadecimal, moves it to the print line and branches to write a line.

TMORE - UG
Turns on prior-line-printed switch and tests for end of the display. If it is end-of-display, a branch is taken to read another potential test-request output record. If it is not the end of the display, a branch is taken to set up another print line.

TRRT - UA
Calculates the storage address of the test request number. It moves the test-request output record to its proper location in main storage and tests for the end of a physical record. If it is the end, it branches to read another physical record. If it is not, it branches to process another logical record.

TRRT1 - UA
Sets the assumed values for ON condition BEGIN and EVERY number.

TSTFL - UK
Tests for floating-point short. If it is, it clears the fraction field.

TSTRTN - UB
Tests for the type of test request record read. The two types are:

1. Phase record

2. Test-request output record

UPADD - UK
Calculates the right-side address for the first line of the display, unpacks and translates this address to hexadecimal, and stores it in temporary storage. It then adds 16 to this right-side address to get the address of the left side of the second line of the display.

UPADDR - UH
Increases either the record address, the head number or the cylinder number by one, depending on whether the last record of the track or the last record of the cylinder was read.

Phase List and Core Map (ATLEFG1)
Charts VA-VL

BACK - VD
Program switch.

When this switch is on, indicating an invalid phase record has been found, the routine branches to read another phase record. When it is off (set to NO-OP), the routine exits to the return address in SBR3.

BEGIN - VA
Initializes line counter for SYSLST. Calculates start address of phase list table in main storage. This start address is stored in USEREND. Calculates address of phase fetch/load table in storage. This table is built backwards in storage starting at COREEND. COREEND is either the end of storage address if the machine size is 16K or 32,767 if the machine is 32K or greater.

Writes header for phase list on SYSLST. Tests for disaster continue and sets the program switch at PROCESS if a disaster-continue situation has occurred.

BELOW - VC
Moves the 'WITHIN SUPERVISOR' message to the print area and sets the switch at location BACK to indicate an invalid phase record has been found.

BINHI - VL
If the new address is greater than or equal to the address in the phase fetch/load table, this routine determines the mid-point of the remaining top half of the table.

BININL - VK
Beginning of subroutine to calculate size of address-sequence table and get the next entry point in table. It compares the address (input entry) to the table entry.

BINLO - VL
If the new address is less than the address in the phase fetch/load table, this routine determines the mid-point of the remaining bottom half of the table.

BINLOP - VL

BINOUT - VL
Calculates size of address-sequence table.

BINSTR - VK
Moves the element to address-sequence table when there is not a low compare between the address and the address in the table entry currently being analyzed.

BUILDOUT - VE
Checks for the end of the start-address-sequence table. If it is not the end of the table, the start address is compared to the end address in the end-address-sequence table. If the start address is less than or equal to the end address, the phase name, phase displacement equal to zero, and the start address of the phase are moved to the output area for the core map records. If the start address is greater than the end address, the routine branches to ENDRTN.

BUILDSEQ - VK
Beginning of subroutine to sort the start and end addresses of the phases in the phase fetch/load table.

CALBIN - VL
Initializes to do a binary search to determine the entry point for the address in the address-sequence table.

CALLOP - VL
Calculates mid-point of the table for binary search.

CHECKLIN - VD
Checks to determine if the maximum line count taken from the system communication region has been exceeded. If the maximum line count for SYSLST has been exceeded, the control character to skip to a new page is moved to the print area and a PUT to SYSLST is issued. The line count is reset to one. If the maximum line count has not been exceeded, it is increased by one and a PUT to SYSLST is issued to write the contents of the print area.

COMP1 - VK
Compares entry in phase fetch/load table to the input entry, which may be either a start or end address, depending on which address-sequence table is being built.

COMP2 - VK
When the input entry is less than the table entry, this routine checks the input entry against the table end address.

COMP256 - VL
Right-adjusts address-sequence table to make room for new address.

COMPARE - VH
Beginning of routine to determine if this phase has already been fetched or loaded at the same address. If it has, the phase fetch/load table is compressed so that it contains only the last phase found at that address.

CONSULT - VK
Sets pointer to end address of phase fetch/load table.

CTLADDR - VC
If the phase fetch or load address is zero, this routine moves the phase origin from the control dictionary record to the phase fetch/load table. It tests to determine if the fetch address lies within the supervisor. If it does, a message is moved to the print area, a switch is set to indicate an invalid phase record and processing continues.

DISASTER - VA
Loads return address of READPHS into SRB3 when there is a disaster continue situation. Initializes the disk address to read the control dictionary file on SYSLNK.

DSKEND - VF
If the core map file on SYSLNK overflows into the EOJ dump area (EOJ2) on SYSLNK, this routine writes message '9905I DISK WORK AREA TOO SMALL' on SYSLOG and turns off the SYM input indicator in OPTSWT in the Autotest communication region.

DSKINCR - VJ
Indicates beginning of routine to update disk address. If the maximum value of the field being updated has been exceeded, the field is reset to its low limit and the next field is updated. RECTBL is the field used for updating purposes. It contains the maximum and reset values of the field to be updated. See RECTBL for further information.

ENDADDR - VC
Calculates end address of phase and moves it to the phase fetch/load table.

ENDRTN - VG
Beginning of routine to find end address of phase.

ENTER - VA
Beginning of ATLEFG1 phase. Gets address of Autotest communication area.

FETCHNXT - VF
Tests for normal EOJ. If the user program reached normal EOJ and a symbolic dump was requested, this routine fetches ATLEFH2. If no symbolic dump was requested, ATLEFH3 is fetched to perform the EOJ dump. If abnormal EOJ, as in the case of disaster continue, was reached and, if a symbolic dump was requested, ATLEFH2 is fetched to perform symbol organization. If a symbolic dump was not requested at abnormal EOJ, ATLEFH3 is fetched.

FINDNAME - VC
Beginning of routine to find a phase name in the phase-list table to match the phase name in the control dictionary record. When a match is found, the phase number and phase origin are moved to phase fetch/load table. If the phase fetch or load address is zero, the phase origin is taken from the control dictionary record.

FINISH - VG
Writes the last core-map record on SYSLNK and moves the end address of the core map file to the Autotest communication region for use by ATLEFH2.

GO - VG
Determines to which phase the restart address belongs. After finding to which phase the address belongs, it moves the restart address, the phase number, and the phase displacement to the output area.

ILPEXT - VJ
Exit for the disk updating subroutine, DSKINCR. Exit is made to the address in SBR.

ININEXT - VC
Initializes pointer to phase-list table. Each entry in the phase-list table is 16 bytes.

LIST - VC
Moves phase name, start and end address of phase and record type (either fetch or load) to print area for phase list.

LOOK - VG
Points to next end-address-sequence table entry to determine to which phase the restart address belongs.

LOW256 - VJ
If there are less than 256 bytes to be moved, it moves the address-sequence table to its correct position in main storage.

MODIF1 - VL
Determines entry point for new address in address-sequence table.

MODIFIN - VB
Points to next logical record in control dictionary file and tests for end-of-file. It also subtracts one from the count of logical records. If end-of-file is reached, the end address of the phase-list table is stored in PHSTABND and the pointer to the start of the phase fetch/load table is initialized.

MODIFOUT - VE
Adds eight to the core-map output area pointer and adds one to the count of core-map logical records. If the output area is full, it writes the core-map record on SYSLNK. Each core-map record is 402 bytes and containes 50 8-byte logical records plus two bytes containing the record code, M, and the number of logical records.

MOVBYT - VL

MOVENEW - VL
Inserts new address in address-sequence table.

MOVESUBR - VJ
Beginning of subroutine to move the start and end address-sequence tables to their correct positions in main storage.

NEXT - VH
Updates pointer to phase fetch/load table to load next entry and checks for table overflow. If there is overflow, the routine moves 'CORE OVERFLOW' message to print area, resets the disaster continue switch at location PROCESS and turns off the symbolic dump switch in the Autotest communication area.

NMLEOJ - VF
Tests for symbolic dump at normal EOJ. If a symbolic dump was requested, the symbol organization phase (ATLEFH2) is fetched. If a symbolic dump was not requested, the EOJ dump phase (ATLEFH3) is fetched.

NODISAST - VH
Sets pointer to the end of the phase fetch/load table when there is no disaster continue situation.

NOTFOUND - VC
Moves 'PHASE NOT FOUND IN CONTROL DICTIONARY' error message to print area when a match between the phase name in the control dictionary record and the phase name in the phase-list table is not found. It also sets the program switch at location BACK to indicate an invalid phase record.

OVERLAID - VK
Points to next entry in the phase fetch/load table whenever an address is overlaid by another phase address.

PRESS - VH
Compresses the phase fetch/load table to contain only the last phase fetched or loaded at the same address.

PROCESS - VD
Beginning of routine to sort the valid start and end addresses of the phases in the phase fetch/load table. Only those addresses not overlaid by a subsequent phase are valid. The program switch at location PROCESS is set to branch if a disaster continue situation has occurred. It is set to NO-OP if there is no disaster continue situation.

PUT - VD
Adds one to the SYSLST line count and issues a PUT to write the contents of the print area on SYSLST.

READCTD - VB
Reads a control dictionary record from SYSLNK and moves the number of logical records in the control dictionary record to REGCNT.

READPHS - VC
Reads test-request output file on SYSLNK and tests for a phase fetch or load record. All test-request output records in the file are bypassed.

RECTBL
Listing only.

Eight-byte field containing maximum values and reset values for disk address updating. Its format is:

```
CHHRCHHR   where:  C = Cylinder
                   H = Head
                   R = Record
```

```
The maximum
values are:  C = CA
             H = 9
             R = 7
```

SEQ2 - VE
Beginning of routine to sort the end addresses of the phases in the phase fetch/load table. The sorted start and end addresses are moved to their respective positions in the start and end-address-sequence tables by performing a binary search on the elements already sorted. The start-address-sequence table overlays the phase-list table in main storage and is followed by the end-address-sequence table.

SORT - VK
Beginning of routine to sort the address and find its place in the address-sequence table.

SUP256 - VJ
Tests for number of bytes in start or end address-sequence-table to be moved. It moves 256 bytes at a time.

TESTNAME - VB
Tests for a phase record. Moves the phase number, phase name, length of phase, and start address of phase to phase-list table. The phase number is a 12-bit number found in the phase record in the control dictionary. The four high-order bits of the phase number are found in the first four bits of the byte following the phase name in the control dictionary record. The eight low-order bits of the phase number are found following the start address of the phase in the control dictionary record.

VALID - VK
Adds eight to the input pointer to test next address against the phase fetch/load table entry.

WRITE - VF
Writes core map records on SYSLNK.

Symbol Organization (ATLEFH2) Charts WA-WE

CALEND - WD
Tests to determine if all records have been processed and gets new phase start address if all records have not been processed.

CMP1 - WA
Determines whether the symbol is in a particular phase by comparing the displacement of the symbol with the displacement value in the core-map record. If the symbol displacement is greater than or equal to the displacement value in the core-map record, the actual phase start address is added to the symbol displacement to get the address of the symbol. If the symbol is in the phase, the symbol and its address are moved to the output area. If the symbol is not in the phase, the routine points to the next symbol to be compared.

DKINCR - WE
Indicates the beginning of the routine to update the disk address. If the maximum value for the field being updated has been exceeded, that field is set to its low limit and the next field is updated.

DKTBL
Listing only.

Eight-byte field containing maximum values for disk address updating. Its format is

CHHRCHHR <u>where</u>: C = Cylinder
H = Head
R = Record

The maximum values are:

1. For core-map file

   C = CA
   H = 9
   R = 7

2. For symbol input

   C = CA
   H = 9
   R = 13

3. For symbol output

   C = CA
   H = 9
   R = 5

EOJ - WD
Turns on program switches to indicate last record and all records processed.

EOJ4 - WC
Moves the end address of the symbol-output area (SYM2 area) to the Autotest communication area.

EOJ5 - WC
Fetches the symbol-sort routine (ATLEFD1).

MOVE2 - WB
Moves symbol and symbol address to output area.

NXTCM - WD
Points to next logical record in core-map record and tests for end-of-record.

NXTSRD - WD
Points to start of symbol records to begin search for phase containing symbol address. Each time the pointer to the core-map record is updated, the symbol file is searched from the beginning.

NXTSYM - WB
Points to next symbol in record. After all symbols in the file have been compared to the core-map record, the next core-map record is read. When all the core-map records have been read and processed, the next phase is fetched.

OVFLOW - WC
Sets switch to bypass symbols, since an overflow condition has occurred in the symbol-output area on SYSLNK.

PACT1
Listing only.

Save area for next phase-start address.

PACT2
Listing only.

Field containing phase-end address.

RD1 - WD
Beginning of subroutine to read a core-map record from SYSLNK into main storage.

RD2 - WD
Beginning of subroutine to read a symbol-input record from SYSLNK into main storage.

RDM1 - WD
Reads next core-map record into main storage.

RDS1 - WA
Indicates beginning of routine to read a symbol-input record from the SYM1 area on SYSLNK.

START - WA
Indicates beginning of ATLEFH2. Gets the start and end addresses of the symbol and core-map records on SYSLNK from the Autotest communication area. The symbol-record output is stored in the SYM2 area on SYSLNK. The start and end addresses of the SYM2 area can be found in the Autotest communication area. For a description of the Autotest work area on SYSLNK, see figures 1 and 2.

STRT1 - WE
Points to the disk field to be updated.

SWA - WC
Program switch to indicate if the symbols are in sequence. If it is set to NO-OP, the symbols are in sequence and the EOJ dump phase (ATLEFH3) is fetched. If it is set to branch, the symbols are not in sequence and the symbol-sort routine (ATLEFD1) is fetched.

SWEOJ - WC
Beginning of routine to determine if all records have been processed.

TBLCM
Listing only.

Eight-byte field containing the maximum and reset values for updating the core-map file.

The maximum values are: C = CA
H = 9
R = 7

TBLSI
Listing only.

Eight-byte field containing the maximum and reset values for updating the symbol input file.

The maximum values are: C = CA
H = 9
R = 13

TBLSO
Listing only.

Eight-byte field containing the maximum and reset values for updating the symbol output file.

The maximum values are: C = CA
H = 9
R = 5

WT1 - WB
Beginning of routine to write symbol-output records on SYM2 area on SYSLNK. Each physical record is 640 bytes and consists of 58 logical records, each 11 bytes in length (symbol plus symbol address).

Symbol Sort Phase (ATLEFD1) Charts XA-XE

BEGIN - XA
Gets disk address of the symbol-input area, SYM2, on SYSLNK from the Autotest communication area. Gets the disk address of the symbol-output area, SYM1, on SYSLNK from the Autotest communication area.

BININL - XC
Beginning of subroutine to compare the address of the symbol in the input area, SRTIPT, to the address of the last symbol sorted.

BINOUT - XD
Beginning of routine to relocate sort-address table, SRTADD, to make room for new entry.

BINSCH - XC
Beginning of routine to sort the symbol records by performing a binary search to determine where the address of the symbol record in the input area, SRTIPT, belongs in the SRTADD table. The SRTADD table contains the addresses of the sorted symbol records in SRTIPT. The SRTADD table is used to build the output records to be written on the SYM1 area on SYSLNK.

BINSTR - XC

BLOCK - XA

BUILD - XC
Beginning of routine to move the sorted-symbol records to the output area to be written on SYSLNK. After a string of symbol records has been moved to the output area, it is written on the SYM1 area on SYSLNK.

CALBIN - XC
Beginning of routine to perform a binary search to place the address of the symbol record in the sort-address table, SRTADD.

COMP256 - XD
Loop to relocate sort-address table, SRTADD, 256 bytes at a time.

DSKEOV - XD
Indicates symbol-output file overflows the EOJ dump (EOJ2) area on SYSLNK. '9906I DISK WORK AREA TOO SMALL' error message is written on SYSLST and SYSLOG.

DSKINCR - XE
Indicates beginning of routine to update the disk address. If the maximum value for the field being updated has been exceeded, that field is reset to its low limit and the next field is updated using RECTBL. RECTBL contains the maximum and reset values of the field to be updated. See RECTBL for further information.

FIRST - XA
Beginning of ATLEFD1.

ILPEXT - XE

INIREC
Listing only.

Two-byte field initialized to 4, indicating four physical records are to be read from SYSLNK into the symbol input area, SRTIPT, to be sorted.

LOOP - XB
Beginning of routine to sort the symbol records in SRTIPT according to the symbol address.

MOVBYT - XD

MOVREC - XC

MVLOOP - XC

NEWREC - XB
Loop to ensure that four physical records have been read into SRTIPT for sorting.

NXTPASS - XA
Beginning of routine to read four physical records from the SYM2 area on SYSLNK into the symbol-record input area, SRTIPT. The symbols are sorted according to their addresses and a SRTADD table is built containing the addresses of the sorted symbol records in the symbol input area, SRTIPT.

NXTPHASE - XB
Stores the number of sorted symbol strings written on the SYM1 area on SYSLNK in the Autotest communication region for use by the symbol merge (ATLEFD2) phase.

NXTREC - XC
Initializes pointer to beginning of output area.

OUTFUL - XC
Determines if the output area for the sorted symbol records is full.

READ - XA
Beginning of routine to read the symbol records written on SYSLNK by the symbol organization (ATLEFH2) phase into the sort input area, SRTIPT.

RECTBL
Listing only.

Eight-byte field containing maximum and reset values for disk-address updating. Its format is:

CHHRCHHR where: C = Cylinder
H = Head
R = Record

The maximum values are: C = CA
H = 9
R = 5

The reset values are: C = CB
H = 0
R = 1

SORT - XA

SRTSUB - XA

STRNUMB
Listing only.

Two-byte field contains the number of strings of symbol records to be merged by ATLEFD2.

WRITE - XD
Writes a 642-byte record on SYSLNK. Each record contains 58 11-byte sorted logical symbol records.

Symbol Merge Phase (ATLEFD2) Charts XF-XN

BEGIN - XF
Indicates beginning of routine to merge four strings of symbol records. As soon as a string is depleted, it is replaced by another string to be merged, and processing continues.

BEGPAS - XF

COMPBA - XK
Beginning of loop to compare an element in string B to an element in string A. If B is greater than A, then the element in string A is moved to the output area. If B is not greater than A, the element in string B is moved to the output area.

COMPCA - XJ
Beginning of loop to compare an element in string C to an element in string A. If C is greater than A, then the element in string A is moved to the output area. If C is not greater than A, then the element in string C is moved to the output area.

COMPCB - XJ
Beginning of loop to compare an element in string C to an element in string B. If C is greater than B, then B is compared to A. If C is not greater than B, then C is compared to A.

COMPDA - XJ
Beginning of loop to compare an element in string D to an element in string A. If D is greater than A, then the element in string A is moved to the output area. If D is not greater than A, then the element in string D is moved to the output area. Each element in the string is an 11-byte record containing the address of the symbol and the symbol itself.

COMPDB - XJ
Beginning of loop to compare an element in string D to an element in string B. If D is greater than B, then B is compared to A. If D is not greater than B, then D is compared to A.

COMPDC - XH
Beginning of loop to compare an element in string D to an element in string C. If D is greater than C, then C is compared to B. If D is not greater than C, then D is compared to B.

DSKEOV - XM
Indicates disk overflow condition for merge output file.

'9906I DISK WORK AREA TOO SMALL'

DSKINCR - XN
Indicates beginning of routine to update disk address. If the maximum value of the field being updated has been exceeded, the field is reset to its low limit and the next field is updated using RECTBL. RECTBL contains the maximum and reset values of the field to be updated. See RECTBL for further information.

DSKMOD - XM
Updates the disk address of the merge output file.

FILLA - XG
Determines whether all the elements in string A have been merged and whether there is another symbol record to be read to refill the depleted string A area. When end-of-file for symbol input is reached, the last string indicator is moved to the output area.

FILLB - XG
Determines whether all the elements in string B have been merged and whether there is another symbol record to refill the depleted string B area. When end-of-file is reached for symbol input, the last-string indicator is moved to the output area.

FILLC - XH
Determines if all the elements in string C have been merged and whether there is another symbol record to fill the depleted string C area. When end-of-file for symbol input is reached, the last string indicator is moved to the output area.

FILLD - XH
Determines whether all the elements in string D have been merged and whether there is another symbol record to fill the depleted string D area. When end-of-file is reached, the last string indicator is moved to the output area.

FIRST - XF
Indicates beginning of ATLEFD2.

GETA - XG
Reads another physical record from SYSLNK to refill the depleted area for string A.

GETB - XG
Reads another physical record from SYSLNK to refill the depleted area for string B.

GETC - XH
Reads another physical record from SYSLNK to refill the depleted area for string C.

GETD - XH
Reads another physical record from SYSLNK to refill the depleted area for string D.

GO - XF
Gets the disk address of the merge output area on SYSLNK and the number of symbol strings to be merged from the Autotest communication area.

ILPEXT - XN

MMPP - XL
Beginning of routine to write a symbol-merge output record on SYSLNK.

MODIF - XM

MODIFA - XG
Updates the disk address for the next string-A record.

MODIFB - XG
Updates the disk address for the next string-B record.

MODIFC - XH
Updates the disk address for the next string-C record.

MODIFD - XH
Updates the disk address for the next string-D record.

MOVE - XM
Moves a symbol record to the output area and increases the address of the output area by 11.

NXTAR - XK
Updates pointer to next symbol record (element) in string A.

NXTBR - XK
Updates pointer to next symbol record (element) in string B.

NXTCR - XJ
Updates pointer to next symbol record (element) in string C.

NXTDR - XJ
Updates pointer to next symbol record (element) in string D.

NXTPAS - XL
Decreases the number of passes required for the merge by one and switches the addresses of the input and output areas on SYSLNK. It then branches to BEGIN to read four more physical records into the areas for string A, string B, string C, and string D.

NXTPHASE - XL
Moves disk address of merge output area to Autotest communication area and fetches the EOJ dump phase, ATLEFH3.

OUTFUL - XM
Tests for end of output area.

PUTA - XK
Moves an element in string A to the output area.

PUTB - XK
Moves an element in string B to the output area.

PUTC - XJ
Moves an element in string C to the output area.

PUTD - XJ
Moves an element in string D to the output record.

RECTBL
Listing only.

Eight-byte field containing maximum and reset values for disk address updating. Its format is:

CHHRCHHR where: C = Cylinder
H = Head
R = Record

The maximum values are: C = CA
H = 9
R = 5

The reset values are: C = CB
H = 0
R = 1

USTOPA - XG
Indicates beginning of routine to open or close the loop to compare an element in string A to an element in string B, string C, or string D. When the smallest of the four elements is found, it is moved to the output area. When the output area is full, the output record is written on the SYM2 area on SYSLNK. Each output record is 642 bytes containing 58 11-byte logical symbol records. Each symbol record consists of the address of the symbol and the symbol itself.

USTOPB - XG
Indicates beginning of routine to open or close the loop to compare an element in string B to an element in string A, string C, or string D. After all the elements in string B are merged, another symbol record is read into the string-B area.

USTOPC - XH
Indicates beginning of routine to open or close the loop to compare an element in string C to an element in string A, string B, or string D. After all the elements in string C are merged, another symbol record is read into the string-C area.

USTOPD - XH
Indicates beginning of the routine to open or close the loop to compare an element in string D to an element in string A, string B, or string C. After all the elements in string D are merged, another record is read into the string-D area.

WRITE - XM

EOJ Dump (ATLEFH3) Charts YA-YN

ALLOUT - YG
At this point, after setting up a symbol on one of the four symbol print lines, the program updates the input pointer in preparation for the next symbol. The program next goes to EQAOUT to update the symbol table addresses to reference the next symbol and its address.

APT1 - YB
The starting address of supervisor dump is unpacked and translated. The input and print line pointers are increased and the line-length counter is set to 12.

APT2 - YB
The supervisor contents are unpacked and translated, four bytes at a time. The input and print line pointers are increased after handling each four bytes, and the counter is decreased by one. Until the counter is reduced to zero, the program loops to process four more bytes of supervisor information. Counter=0 indicates the end of a print line and the line is compared to previous line(s) to see if it is the same. Whenever the end of supervisor is detected, the line is written on SYSLST regardless of length, and the program proceeds to process user label area length.

CCRET - YA
The line heading and contents of the user PSW are set up and written on SYSLST. The line heading, GR 0-7, for the first of two groups of general registers is also set up.

CHDMP - YL
This is the beginning of the character-dump routine used to set up a line of main storage, in character format, for the EOJ dump. The routine retrieves the input start address, print line address, line identifier, and line displacement. It also initializes two counters (word and line) to 4, before going to LOOPC1 to process the first byte of the line.

CHEOC - YH
When a low-compare condition occurs at SYMPT1 in the SYMDMP routine and the end of a symbol record has not been reached, the routine tests for padding. If padding has occurred, the maximum storage address is set and the routine branches to SYMPT1 to force a high compare. When there is no padding, the routine goes to ERROR10 to set up and write the symbol error message.

CHEXIT - YL
At this point switch 1 is set off, following the setting up of the second half of a print line in character format. Next the print-line address is loaded, and the print routine (PRTRTN) called to write the line on SYSLST. After the line is written, the routine returns to TSTMNE to test for other requested formats for the main storage dump.

CHK1 - YF
At this point in the SYMDMP routine, a test is made to see if symbols have been set up on symbol line S1; if so, the print routine is called to write the line. If no symbols have been set up on S1, the routine goes to SDREST to store the symbol addresses.

CHK2 - YF
At this point in the SYMDMP routine, a test is made to see if symbols have been set up on symbol line S2; if so, the print routine is called to write the line. If there are no symbols on S2, the routine goes to CHK1 to test for symbols on symbol line S1.

CHK3 - YF
At this point in the SYMDMP routine, a test is made to see if symbols have been set up on symbol line S3; if so, the print routine (PRTRTN) is called to write the line. If no symbols have been set up on symbol line S3, the routine goes to CHK2 to test for symbols on symbol line S2.

CHSW1 - YL
At this point switch 1 is tested, following the setting up of half a print line in character format. When switch 1 is not on (first half line), switch 1 is set on, and the routine loops to LOOPC1 to convert the first byte for the second half line. When switch 1 is on, it is set off at CHEXIT before the complete line is written.

CKEQLN - YD
First, the next 32 bytes of main storage dump are moved to the print line. The program then tests to see if this line is equal in content to the previous line. If the lines are equal, the program loops back to TS2 to test for print formats. When the printout lines are not equal, the present line is held (SPVSW).

CPT1 - YB
Four bytes of communication region information are unpacked and translated into the print area. The line counter is decreased and tested; if it is not zero, the program loops to process the next four bytes of communication information. When the counter has been reduced from 12 to 0, the communication information is written on one line.

DITTO - YD
Switch SWA is set off when equal lines have been dropped on the main storage dump. The address of the print line is loaded, and the storage address for the line is obtained. The storage address is unpacked and translated to the print line. The DITTO message is moved to the print line and then written. The program then loops back to test for print formats requested for the main storage dump.

ENDING - YE
When the end of the dump is indicated by switch A being on, the END message is moved to the last print line.

ENDLNS - YF
When the end of a symbol line has been indicated during a main storage dump, the routine stores the symbol address. Next, it tests for and writes the symbol lines that contain symbols. First, a test of symbol line S4 is made. If symbols have been set up, the print routine (PRTRTN) is called to write the line on SYSLST. When no symbols are set up, the routine goes to CHK3 to test symbol line S3.

ENDPAS - YD
The one-time switch is tested, following an end-of-storage indication. If the switch is on, the program gets the end-of-storage address. If the switch is not on, the program tests switch A to determine if the end of the dump has been reached, or if the storage address for the current line is to be unpacked, translated, and written with a DITTO message.

EOP1 - YD
When the one-time switch at ENDPHS is on, the end address of main storage is stored and the SWEND switch is set to branch on not low. SW2 is set on, the one-time switch is set off, and a compare of end-of-record and end-of-logical-core is made. If end-of-record is greater than end-of-core, the program loops back to ENDPHS. If not greater, the program loops to CKEQLN.

EQAOUT - YG
At this point in the SYMDMP routine, the symbol table addresses are updated and the symbol address and symbol-record end address are retrieved. A test is made for end-of-symbol record. If end-of-symbol record is indicated, the program branches to SYMRD to read another symbol

record. If not end of symbol record, a test is made for the end of an input line. If not end of input line, the program branches to SYMPT1 to compare the current storage address to the symbol address. When the end of an input line is encountered, the program branches to ENDLNS to write the symbol lines that contain symbols.

ERROR10 - YH
If an error has been detected during the SYMDMP routine, the symbol error message is moved to the print line and the print routine (PRTRTN) is called to write the message. The print routine returns to SYMDMP, which cancels the EOJ dump.

FPT1 - YB
Eight bytes of floating-point register information are unpacked and translated. The register counter is decreased and tested. If the counter is not zero, the program loops to process the next eight bytes of floating-point information. Counter=0 causes the contents of the floating-point registers to be written and the formatting of the user communication region begun.

FPTO - YB
Sets up to write the contents of the even-odd pairs of floating-point registers. The line header, FR 0-6, is moved to the print line.

GET1 - YD
Retrieves the next logical record before setting up the new input-start address at GET2 of the main storage dump.

GET2 - YC, YD
The input starting address is retrieved at this point in preparation for MNLINE.

GPT2 - YA
The general register contents are unpacked and translated four bytes (1 register) at a time and moved to the print line. The input and print line pointers are increased, and the counter is decreased and tested after each four bytes. Until the counter reaches zero, the program loops to process the next four bytes. Counter=0 results in the writing of a line, giving the contents of GR 0-7. The routine repeats for GR 8-15.

GPTO - YA
The print line pointer for the general register printout is increased. The general register group counter is also set up to control the printout.

HEXDMP - YK
This is the beginning of the hexadecimal dump routine used for main storage dump. The routine first retrieves the print line and input start addresses, and adds the displacement to the print line. A test is next made to see if a storage dump is to be performed. If so, the routine branches to HX1 to set up the first line. If a storage dump is not indicated, the routine tests to see if the end of an input disk record has been reached. If not, the routine branches to HX1. If the end of an input record is indicated, a new record is read, and the routine proceeds to HX1 to set up the print line.

HEXITA - YK
After 2 complete print lines are set up in hexadecimal format, the storage address for the first (LH) half of the line is unpacked, translated, and moved to the print line. Next, the input pointer is increased to the next storage address and the address is stored. The print line pointer is now increased to the second (RH) half-line storage address position. The RH storage address is unpacked, translated, and moved to the print line. The print routine then writes the complete line. After the line is written, switch 1 is set off and the routine branches to TSTCHR to determine the other dump formats requested for the main storage dump.

HX1 - YK
After setting up for a new line in hexadecimal format, the line counter is set to four, and the current storage address is retrieved. The HEXDMP routine now proceeds to LOOPH1, to convert four bytes into hexadecimal format.

HXSW1 - YK
The HEXDMP routine now tests switch 1 for the end of a print line (second or RH half-line set up). If the switch is not on, it is set on (first or LH half line), the print-line pointer increased, and the line counter reset to 4. The routine then loops to LOOPH1 to begin processing the first four bytes of the second half-line. If switch 1 is on, the routine goes to HEXITA to set up the storage address for the left-hand (LH) half line.

INGET - YA

The address of the Autotest communication region is obtained from low core along with the disk addresses of the core-map file, symbol file, and the 155-byte EOJ dump record stored on SYSLNK. If disaster-continue has occurred, the EOJ dump is bypassed and phase ATLEG01 fetched.

LOOPC1 - YL

A byte of main storage is converted to character format and moved to the print line. The input and print line pointers are then updated and the word counter decreased and tested. If the counter is not zero, the routine loops to convert the next byte. When the word counter is zero, the end of a word (four bytes) is indicated, the print-line pointer is increased by 3 (space between words), and the word counter reset to 4. The line counter is now decreased by 1 and tested.

If the line count is not zero, the routine loops to convert the first byte in the next word to character format. If the line counter is zero, the end of a half-line is indicated, the print line pointer is updated, the line counter reset to 4, and the routine goes to CHSW1 to see if a complete print line has been set up.

LOOPH1 - YK

At this point in the main storage dump, four bytes of input are unpacked and translated to hexadecimal and the input and print line pointers updated. Next, the storage address is updated, and the line counter decreased by 1 and tested. If the counter is not zero, the routine loops to process the next four bytes of input. When the counter goes to zero, the end of a half-print line is indicated and the routine goes to HXSW1 to test for end-of-print line.

LOOPM1 - YM

The first byte of a line of main storage is converted to mnemonic and moved to the print line. The input and print line pointers are updated and the byte counter is decreased by 1 and tested. If the byte counter is not zero, the routine loops to LOOPM1 again to convert the next input byte. When the byte counter is zero, the print-line pointer is increased by 3 (space between groups), the byte counter reset to 2, and the group counter decreased by 1 and tested. If the counter is zero, a half line is indicated, the print-line counter is updated, the group counter reset to 4 and the mnemonic switch tested (MNSW). If the group is not zero, the program loops to LOOPM1 to convert the next byte of main storage to mnemonic.

LOOPP1 - YG

At this point the SYMDMP routine tests for a blank in the symbol set up on symbol line S1 of the main storage dump. When a blank is found, the routine branches to UPREG6 to increase the print line pointer by 1. If a blank is not found, a test is made to see if an eight character symbol is present. If the symbol has eight characters, the routine goes to UPREG6. If not eight characters, the print line pointer and blank counter are increased by 1 and the routine loops to test again for a blank.

LOOPP2 - YG

At this point the SYMDMP routine tests for a blank in the symbol set up on symbol line S2 of the main storage dump. When a blank is found, the routine branches to UPREG7 to increase the print line pointer by 1. If a blank is not found, a test is made to see if an 8 character symbol is present. If the symbol has eight characters, the routine goes to UPREG7. If not eight characters, the print line pointer and blank counter are increased by 1 and the routine loops to test aqain for a blank.

LOOPP3 - YH

At this point, the SYMDMP routine tests for a blank in the symbol set up on symbol line S3 of the main storage dump. When a blank is found, the routine branches to UPREG8 to increase the print-line pointer by 1. If a blank is not found, a test is made to see if an eight-character symbol is present. If the symbol has eight characters, the routine goes to UPREG8. If not eight characters, the print-line pointer and block counter are increased by 1 and the routine loops to test again for a blank.

LOOPP4 - YJ

At this point the SYMDMP routine tests for a blank in the symbol set up on symbol line S4 of the main storage dump. When a blank is found, the routine branches to UPREGA to increase the print line pointer by 1. If a blank is not found, a test is made to see if an eight-character symbol is present. If the symbol has eight characters, the routine goes to UPREGA. If not eight characters, the print-line pointer and blank counter are increased by 1 and the routine loops to test again for a blank.

LPTO - YC

The line header for the length of the user label area is moved to the print line. The label length is unpacked, translated, and moved into the print area and then written. The end address of the last supervisor print line is set to the next even address in preparation for the main storage dump.

MNDMP - YM

This is the beginning of the mnemonic dump routine for the main storage dump. The input start address, print-line identifier, print-line start address, and print-line displacement are retrieved. Two counters are set to 2 and 4 to check number of bytes in a group and number of groups in a print line, respectively. The routine then goes to LOOPM1 to process the first input byte.

MNEXIT - YM

At this point, the mnemonic switch is set to NO-OP, after setting up a complete print line of the main storage dump in mnemonic format. The print line address is loaded and the print routine (PRTRTN) called to write the line. When the line has been written on SYSLST, the routine returns to UPPTR1 to set up for the next line of the main storage dump.

MNLINE - YC

The Autotest communication region address is retrieved along with the dump format and disk-record end address. The format switches are set for both normal and abnormal dumps. The program next tests for a requested character or mnemonic dump of main storage. When requested, the line count is updated, and the line-skip switch is set before going to TS2. When a character or mnemonic dump is not requested, the program goes directly to TS1.

MNSW - YM

After half a print line is set up in mnemonic for the main storage dump, the mnemonic switch is tested. If the switch is not on (first half line), it is set on and the program branches to LOOPM1 to begin converting the next half-line. If the mnemonic switch is on (second half-line), it is set off, the print line address loaded, and the print routine (PRTRTN) called to write the complete line on SYSLST.

NMLDMP - YA

If the user program reached normal EOJ, a test is made to see if a dump has been requested. If not, the EOJ dump is bypassed and phase ATLEG01 fetched. When a dump is requested at normal EOJ, a test is made to see if the supervisor is to be included. If not, the bypass switch is set. A 'NORMAL EOJ' message is set up, and the message is written.

OUTOSYM - YH

When it is determined that the end of the symbol list has been reached during SYMDMP, the symbol-dump switch is set off. The routine then branches to ENDLNS to store the last symbol address and to write the symbol lines that have symbols set up on them.

PRERROR - YE

At this point, the 'END' message for the printout is written, a page is skipped, the address of the Autotest communication region is loaded, and phase ATLEG01 is fetched.

PRTN1 - YN

At this point in the print routine, PRTRTN, it has been determined that the end of a page has not been reached and a line can be written. The routine returns to the main program after writing a line.

PRTRTN - YN

This is the print routine used to set up for and write a line of the EOJ dump. The routine stores registers, increases the line count and tests for full page. When a full page is completed, the line count is reset, a skip to a new page performed, and the line written. When full page is not indicated, the routine writes a line. After the line is written on SYSLST, the routine returns to the main program.

PRTTYP - YA

The printer carriage skips to 1 and the abnormal end-of-job reason message is written.

PUTLN1 - YG

After determining that the symbol currently processed must go on the S1 symbol line of the main storage dump, the routine proceeds to move the symbol to the print line. The print-line pointer is retrieved and the blank counter set to 1. The symbol is then moved to the print line. The routine next goes to LOOPP1 to test for a blank in the symbol.

PUTLN2 - YG
When it is determined that the symbol currently processed must go on the S2 symbol line of the main storage dump, the routine proceeds to move the symbol to the print line. The print-line pointer is retrieved and the blank counter set to 1. The symbol is then moved to the print line. The routine next goes to LOOPP2 to test for a blank in the symbol.

PUTLN3 - YH
After determining that the symbol currently processed must go on the S3 symbol line of the main storage dump, the routine proceeds to move the symbol to the print line. The print-line pointer is retrieved and the blank counter set to 1. The symbol is then moved to the print line. The routine next goes to LOOPP3 to test for a blank in the symbol.

PUTLN4 - YJ
After determining that the symbol currently processed must go on the S4 symbol line of the main storage dump, the routine proceeds to move the symbol to the print line. The print-line pointer is retrieved and the blank counter set to 1. The symbol is then moved to the print line. The routine next goes to LOOPP4 to test for a blank in the symbol.

RDPANL - YA
The 155-byte EOJ dump record is fetched from SYSLNK. The line count is retrieved from the supervisor communication region along with the end address for logical core, beginning address of the B-transient area, and the end address of the supervisor. When an abnormal end-of-job is indicated, the cancel code is analyzed and the appropriate EOJ reason message set up.

SDREST - YF
After any symbols on the symbol lines have been written, the current symbol addresses are stored and the SYMDMP routine returns to TSTHEX to test for any other dump formats requested for the main storage dump.

SPVSW - YD
At this point, the present print line of main storage dump is stored, following the test for equal lines at CKEQLN, when unequal lines are indicated.

SSW0 - YB
The supervisor-bypass switch is tested. If on, a supervisor dump is not requested and the program branches to write the user label area length. When a supervisor dump is requested, the header is moved to the print area and written. The input address is set to location 320 and the address of the print line, the start address of the B-transient area, and the end-of-supervisor address are retrieved.

START - YA
The base register is initialized in preparation for the dump.

STRSMB - YF
When it is determined that the symbol address is not the end of storage or there is not a blank symbol, the SYMDMP routine gets the symbol location from the symbol table and updates the print-line pointer. Next, the SYMDMP routine tests to determine which of the four symbol lines should receive the current symbol. The routine branches to the appropriate routine PUTLN1, PUTLN2, PUTLN3, or PUTLN4, where the symbol is set up on the print line in the next available position. When the symbol can not go on any of the four symbol lines, the routine branches to ERROR10 to set up and write the symbol error message.

SU0 - YC
The even adjusted supervisor-end-address is increased by 32 and compared with the actual supervisor end address. When the compare is low, the program loops, increases the address by 32, and compares again until an equal occurs. An equal causes 32 to be subtracted from the adjusted address before aligning to a doubleword boundary as performed at SU1.

SU1 - YC
The end-of-supervisor address is aligned to an even doubleword boundary and compared to the supervisor end address. If not equal, the address is aligned upward to the next doubleword and compared again. This process continues until an equal compare occurs. The displacement is then added to the print-line address, which is stored at SU2.

SU2 - YC
The print-line start address is stored at this point.

SWA - YD
A test is made to see if any equal lines of the main storage dump were dropped. If so, switch A is set off at DITTO. If not, the program loops back to retest for the requested print formats for the main storage dump.

SWEND - YD
The program tests for the end of main storage following execution of the assumed and/or specified dump formats for main storage. When end-of-storage is not indicated, the input pointer is updated by 32 and a test for end-of-input record is made at SWZ. End of main storage results in a test of the one-time switch at ENDPHS.

SWH1 - YD
At the end of a line of supervisor information the program tests for equal lines. If not equal, a line is written at SWH1A. When equal lines are indicated, the switch, SWH2, is set to a NO-OP, the input pointer is updated by 48 and another test for equal lines is performed.

SWH1A - YC
When successive lines of the supervisor dump are not equal, a line is written, and the print-line address updated. The program then tests for a NO-OP on equal lines at SWH2.

SWH2 - YF
Following the writing of a supervisor line, a test for equal lines is made (SWH2 off). If not equal lines, the program returns via SWH3 to APT1 to process the storage address of the next supervisor line. With equal lines, the switch, SWH2, is set on and the storage address of the last equal line is unpacked, translated, and moved into the print area. The DITTO message is moved to the print line and the line is written. A return to APT1 via SWH3 is made to process the next supervisor line.

SWH3 - YC
After writing a line of the supervisor when equal lines are not present, the program returns to APT1, to unpack and translate the storage address for the next supervisor line.

SWZ - YD
A test is made for end of input record, when the end of main storage is not indicated. If end of input record is not indicated, the program branches to CKEQLN to test for equal lines. When the end of an input record is indicated, the next logical record is read.

SYMDMP - YF
This is the beginning of the symbol dump routine for the main storage dump. The routine retrieves the current storage address, the end-of-input line address, the print-line pointers and symbol addresses. The symbol print-line areas are cleared and the identifiers moved to the symbol print lines.

SYMERR - YH
When a low compare of the symbol address and the current storage address are determined at SYMPT1, the SYMDMP routine tests for equal symbols. When equal symbols are present, the routine branches to EQAOUT to update the symbol addresses. When equal symbols are not recognized, the Autotest communication region address is retrieved, the symbol address is updated, and the routine tests to see if the end of the symbol list has been reached. If the end of the symbol list has been reached, the routine goes to OUTOSYM to set the symbol dump switch off before branching to ENDLNS. When the end of the symbol list has not been reached, the routine tests for the end of a symbol record. At the end of a symbol record, the routine goes to SYMRD to read another symbol record. When not the end of a symbol record, the routine goes to CHEOC to test for blank padding.

SYMHI - YF
When the symbol address compares high to the current storage address, the SYMDMP routine updates the storage address pointer and compares the storage address to the end-of-line address. If the compare is low, the routine returns to SYMPT1 to compare the storage address to the current symbol address. A high compare causes a branch to ENDLNS, where the symbol address is stored and the symbol lines with symbols set up are written. When an equal compare is encountered, the routine tests for a symbol at this last storage address. If there is a symbol at the last storage address, the routine returns to SYMPT1 to compare the symbol address to the current storage address. When there is no symbol at the last storage address, the routine goes to ENDLNS, the last symbol address is stored, and the symbol lines with symbols set up are written on SYSLST.

SYMPT1 - YF
The symbol address is compared to the current storage address during the SYMDMP routine. If the compare is low, the routine branches to SYMERR and tests for equal symbols. If the compare is high, the routine branches to SYMHI and the storage address pointer is updated. An equal compare results in a test for end-of-storage. If not end of storage, the routine goes to STRSMB and retrieves the symbol address, the symbol location and updates the print-line pointer.

SYMRD - YF
At this point in the SYMDMP routine, the registers are stored and a symbol record is read. The registers are restored and the symbol and symbol-address pointer and linkage registers are reset. The routine then goes to SYMPT1 to compare the current storage address with the symbol address.

TOMOVER - YA

TS1 - YD
Depending on the type of EOJ dump requested, the line counter is increased by the following number:

Hexadecimal dump only - 2

Hexadecimal and character dump - 3

Hexadecimal and mnemonic dump - 3

Hexadecimal and symbolic dump - 6

Hexadecimal, character, and mnemonic dump - 4

Hexadecimal, character, and symbolic dump - 7

Hexadecimal, mnemonic, and symbolic dump - 7

Hexadecimal, character, mnemonic and symbolic dump - 8

A test for end-of-page is then made. If it is the end-of-page, the line count is zeroed and a switch is set to skip to a new page. The routine tests for a symbolic dump requested. If end-of-page has not been reached, the routine branches to the symbolic dump test at TS2.

TS2 - YD
A test for symbolic format for the main storage dump is made. If requested, the program goes to the symbol dump routine (SYMDMP - Chart XF). When symbolic format is not requested, the program tests for hexadecimal format at TSTHEX.

TSTCHR - YD
A test is made to determine if character format is requested for the main storage dump. If so, the program goes to the character-dump routine (CHDMP - Chart XL). When character format is not specified, the program tests for mnemonic format at TSTMNE.

TSTHEX - YD
A test is made for hexadecimal format for the main storage dump. If hexadecimal is requested, the program goes to the hexadecimal dump routine (HXDMP - Chart XK). When hexadecimal format is not indicated, the program tests for character format (TSTCHR).

TSTMNE - YD
A test is made to see if mnemonic format is specified for the main storage dump. If so, the program goes to the mnemonic dump routine (MNDMP - Chart XM). When mnemonic format is not specified, the program retrieves the input pointer, and updates the storage address pointer (this occurs at UPPRT1).

UPREG6 - YG
If a blank position or an eight-character symbol is present on an S1 line of the symbolic dump, the print-line pointer is increased by 1 and the contents of the blank counter are added to the print-line pointer to get the next print position address on the S1 symbol line. The routine next goes to ALLOUT to update the input pointer in preparation for the next symbol.

UPREG7 - YG
When a blank position or an eight-character symbol is present on an S2 line of the symbol dump, the print-line pointer is increased by 1 and the contents of the blank counter added to the print-line pointer to get the next print position address on the S2 symbol line. The routine next goes to ALLOUT to update the input pointer in preparation for the next symbol.

UPREG8 - YH
When a blank position or an eight-character symbol is present on an S3 line of the symbol dump, the print-line pointer is increased by 1 and the contents of the blank counter added to the print-line pointer to get the next print-position address on the S3 symbol line. The routine next goes to ALLOUT to update the input pointer in preparation for the next symbol.

UPREGA - YJ
After a blank position or an eight-character symbol is detected in an S4 line of the symbol dump, the print-line pointer is increased by 1 and the contents of the blank counter are added to the print-line pointer to get the next print-position address on the S4 symbol line. The routine next goes to ALLOUT to update the input pointer in preparation for the next symbol.

UPPRT1 - YD
At this point, following the test and execution of the various dump formats, the input pointer is retrieved and the storage-address pointer updated. The program next tests for end-of-main storage at SWEND.

Autotest End Routine (ATLEG01) Charts YP-YQ

AONE
Listing only.

Label for error message '1 ATEOF CARD ORDER ERROR'.

CHKEXEC - YP
Beginning of routine to write the ATEOF card order error message on SYSLST and to set up a scan of the job control card. The scan starts with card column 4 and ends with card column 68.

EXECLOOP - YP
Checks for EXEC฿ in the operation field of the job control card read. Because job control cards have free format, all the card columns through column 68 are checked for a blank or EXEC฿.

EXECYES - YQ
Scans the operand field following //฿EXEC฿ on the job control card. If the operand field is not blank, the job is canceled. If the operand field is blank, the error message 'MORE THAN ONE EXEC CARD IS INVALID IN AUTOTEST' is written on SYSLST and the job is canceled.

FJC - YP
Whenever a /& or ATEOF card is read, this routine issues an SVC 14 to cancel the job and fetch job control for the EOJ step.

IJVTI120 - YP
Initializes base register.

NOXALOWD - YQ
If no operand field is found following EXEC฿, this routine writes the error message, ATEOF CARD ORDER ERROR, on SYSLST and cancels the job.

OPEN - YP
Beginning of routine to reset the second job control byte in the system communication region to ignore the next EXEC card read. This situation will occur if an EXEC card is read before the ATEOF or /& card. This routine also opens SYSIPT.

RDCD - YP
Beginning of routine to read a card from SYSIPT and to determine if a /&, ATEOF, or job control (//฿) card was read. If none of these were read, another card is read from SYSIPT. If a /& or ATEOF card was read, the job is canceled and job control is fetched to execute the EOJ step. An error message is written on SYSIPT if the ATEOF card is out of order or if a //฿EXEC฿ card is read.

T
Listing only.

Label for error message 'MORE THAN ONE EXEC CARD IS INVALID IN AUTOTEST'.

Disaster-Continue Routine (ATLECONT) Chart ZA

ERROR1 - ZA
Opens SYSLST and writes the error message indicating the Autotest communication record is not on SYSLNK. The job is canceled.

HERE - ZA
Indicates beginning of ATLECONT. Initializes the base register.

MSG2
Listing only.

Indicates error message '9F02I AUTOTEST COMMUNICATION RECORD NOT ON SYSLNK'.

THERE - ZA
Opens SYSLNK and reads the Autotest communication record into the first byte of the problem program area.

Card-To-Tape Variable Utility (ATLEJCTV) Charts ZB-ZE

ATRTN - ZB
This routine is executed when an Autotest control card is read from SYSIPT. It moves the message, AUTOTEST CONTROL CARD ASSUMED, to the print area.

BLNKRTN - ZC
This routine is executed when a card with blanks in columns 7 and 8 is read from SYSIPT. It initializes the card character count to 70, the number of card columns containing data information. It then tests for hexadecimal input (H in column 9).

CCANAL - ZC
Reads a card from SYSIPT and checks type of card read.

CHKTAP - ZB
Calculates the maximum length output record allowed by subtracting the end address of ATLEJCTV from the address of the uppermost byte of the problem program area as determined by the IPL program. It also checks the assignment of SYS005.

CONHEX - ZC
This routine is executed when hexadecimal input is specified on the data card. Because hexadecimal information requires two card columns for each character, this routine divides the card character count by two and stores the result as the actual number of characters to be moved to the output record area. It then multiplies the actual character count by two to determine if an even number of hexadecimal characters was punched in the data card.

CREATEN - ZE
Creates the record number for the short record warning message to be written on SYSLST. It converts the record count number to decimal and suppresses any leading zeros.

ENDRTN - ZE
This routine is executed when a /* card is read from SYSIPT. It tests to determine if the record output area is empty and if a rewind of SYS005 is requested. If the output area is not empty, an error message is written on SYSLST, indicating a WT card to write the contents of the output area on tape was not read before the /* card.

EOFRTN - ZB
This routine is executed when a /& card is read from SYSIPT. This routine moves the error message, NO /* AT /&, indicating a /* card should have preceded the /& card in the job stream, to the print area.

EOJRTN - ZE
Writes the message, EOJ CARD TO TAPE VARIABLE, on SYSLST and fetches job control to perform the EOJ step.

EXECTP - ZE
Executes the appropriate tape operation and checks for end-of-volume on SYS005.

GOTIT - ZD
This routine is executed after the rightmost asterisk to delimit the data field in the MR or WT card is found. It subtracts one from the card character count to determine the number of data characters in the card. It then tests to determine if hexadecimal input has been specified (H in column 9).

INTRTN - ZD
Clears work registers and initializes pointer to beginning of output area. It then branches to CCANAL.

ISIT - ZD
Indicates beginning of routine to determine if the rightmost asterisk has been found in the MR or WT card. If the rightmost asterisk to delimit the data field has not been found, it subtracts one from the card character count and points to the next card column. This routine then checks to see if all the card columns (11-80) in the data field have been analyzed. If 70 decrements have not been made, a check for a rightmost asterisk is performed again. If the rightmost asterisk is found, the routine branches to GOTIT.

LUNASGTP - ZB
Moves the error message, SYS005 IMPROPERLY ASSIGNED -- LOG UNIT, indicating SYS005 is not assigned to a logical unit, to the print area.

MRRTN - ZD
This routine is executed when an MR card is read from SYSIPT. It initializes the card character count to 70, the maximum number of card columns containing data information, and points to the last column of the data card to begin the scan for the delimiting asterisk.

MVHEXCHR - ZC
Converts two hexadecimal characters at a time to binary and moves the converted characters to the print area for temporary storage.

MVPRT - ZC
Moves the converted hexadecimal characters from temporary storage in the print area to the card area.

NOTTAP - ZB
Moves the error message, SYS005 IMPROPERLY ASSIGNED -- NOT A TAPE, indicating SYS005 is not assigned to a tape, to the print area.

PRTANDCN - ZD
Writes the appropriate error message and its associated card image on SYSLST and cancels the job.

PRTMSG1 - ZD
Moves the error message, CONTROL INFO INVALID, indicating an invalid data card was read, into the print area.

PRTMSG2 - ZE
Moves the error message, NO WT AT /*, indicating a WT card to write the contents of the output record area on tape was not read before the /* card, to the print area.

PRTMSG3 - ZD
Moves the error message, NO WT AT TM, indicating a TM card was read before the WT card, to the print area.

PRTMSG4 - ZB
Moves the error message, JOB CONTROL CARD ASSUMED, indicating a job control card was read from SYSIPT, to the print area.

PRTMSG6 - ZD
Moves the error message, NO * FOUND, indicating the asterisk on the WT or MR card was not found, to the print area.

PRTMSG8 - ZC
Moves the error message, UNEVEN NO. OF HEX. CHARACTERS, indicating an uneven number of hexadecimal characters was punched in the data card, to the print area.

PRTMSG9 - ZC
Clears the print area and moves the error message, INVALID HEX. CHAR., indicating an error occurred while converting the hexadecimal character to binary, to the print area.

PRTNAME - ZB
Writes the message, BEGIN CARD TO TAPE VARIABLE, on SYSLST.

PRTSRM - ZE
Writes the short record message, RECORD NO. XXXXX IS SHORT, on SYSLST and changes the return address in LINKREG to INTRTN.

PUNASGTP - ZB
Moves the error message, SYS005 IMPROPERLY ASSIGNED -- PHY UNIT, indicating SYS005 has not been assigned to a physical unit, to the print area.

RDFRSTCD - ZB
Reads the first card from SYSIPT, determines if SYS005 is to be rewound initially, and checks the type of card read.

REWIND - ZB, ZE
Rewinds SYS005.

SHTRECRT - ZE
Checks for a record length of zero when a record is less than 18 bytes.

START - ZB
Beginning of ATLEJCTV. It initializes the base register and opens SYSLST and SYSIPT.

TMRTN - ZD
This routine is executed when a TM card is read from SYSIPT. It checks to determine if the output record area is empty. If it is empty, it writes a tapemark on SYS005 and branches to CCANAL to read another card. If the output area is not empty, it branches to PRTMSG3 to process the error.

TOOMUCH - ZC
Moves the error message, DATA EXCEEDED, indicating the maximum output record length has been exceeded, to the print area.

TPEOF - ZE
Writes the error message, 9J01I EOV ON SYS005, indicating end-of-volume has occurred on SYS005, on SYSLST and SYSLOG. The job is canceled.

UPCNT - ZC
Updates the output area character count by the actual number of characters in the data field. It checks to determine if the output area character count exceeds the maximum length output record. If it does, this routine branches to TOOMUCH to process the error. If it does not exceed the maximum output record length, it moves the characters from the card area to the output record area and updates the output area pointer by the number of characters in the card area. It tests the write-tape switch. If it is on, it branches to WRTTPRTN to write an output record on SYS005. If it is off, it branches to CCANAL to read another card from SYSIPT.

WRITETP - ZE
Updates the record count by one and tests for a short record (less than 18 bytes). If it is not a short record, the routine branches to EXECTP to perform the required tape operation. If it is a short record, the SHTRECRT routine is executed to check for a record length of zero.

WRTTPRTN - ZD
Stores the output record character count in the CCW, turns off the write tape switch, writes the output area record on SYS005, and branches to INTRTN to initialize the pointer to the beginning of the output area.

WTRTN - ZB
This routine is executed when a WT card is read from SYSIPT. It turns on the write-tape switch and branches to MRRTN.

ZERREC - ZE
Moves the error message, ZERO LENGTH RECORD, to the print area and branches to PRTANDCN to write the message and card image on SYSLST and to cancel the job.

Chart AA. Initialization (ATLEDT) (Refer to Chart 02)

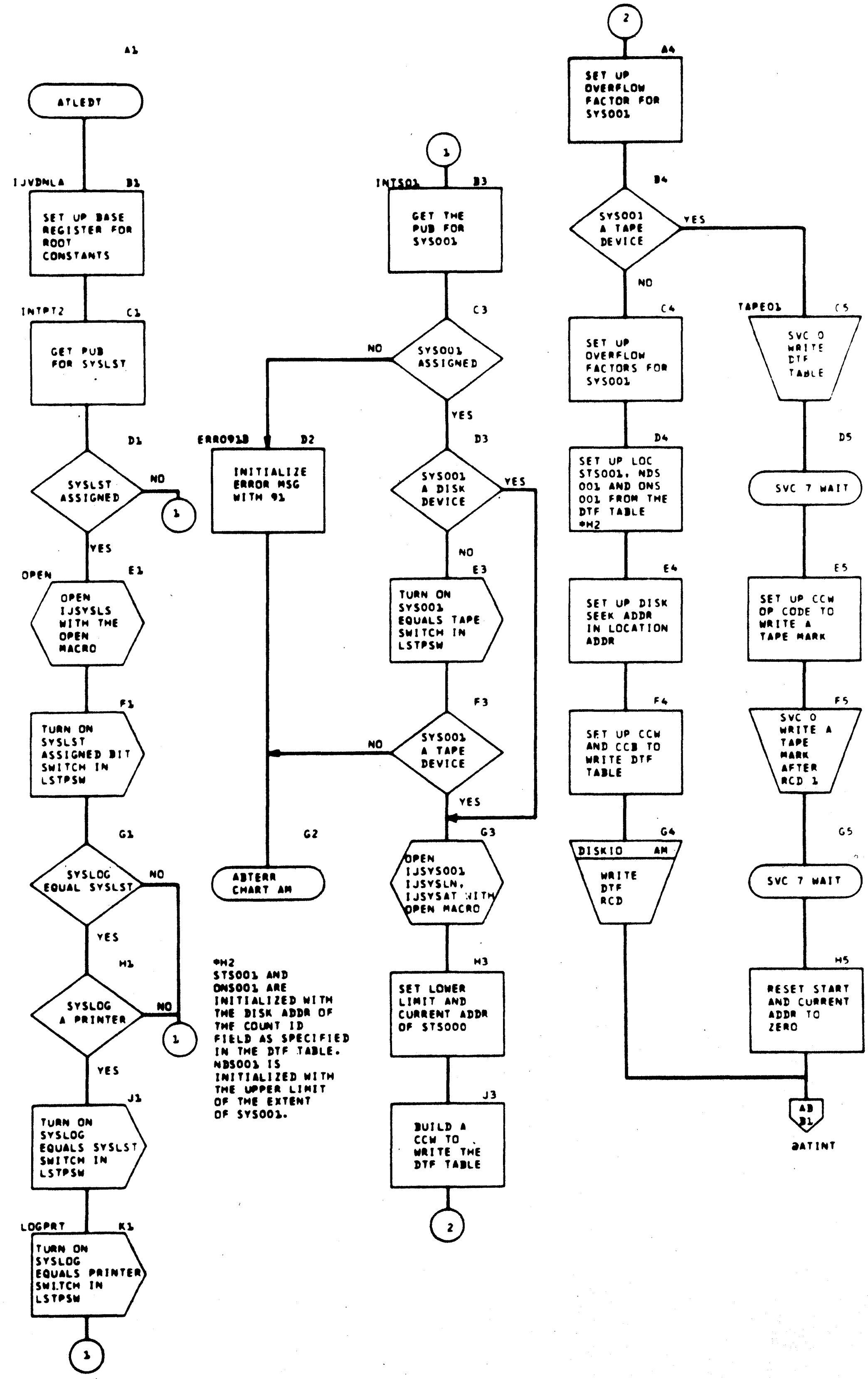
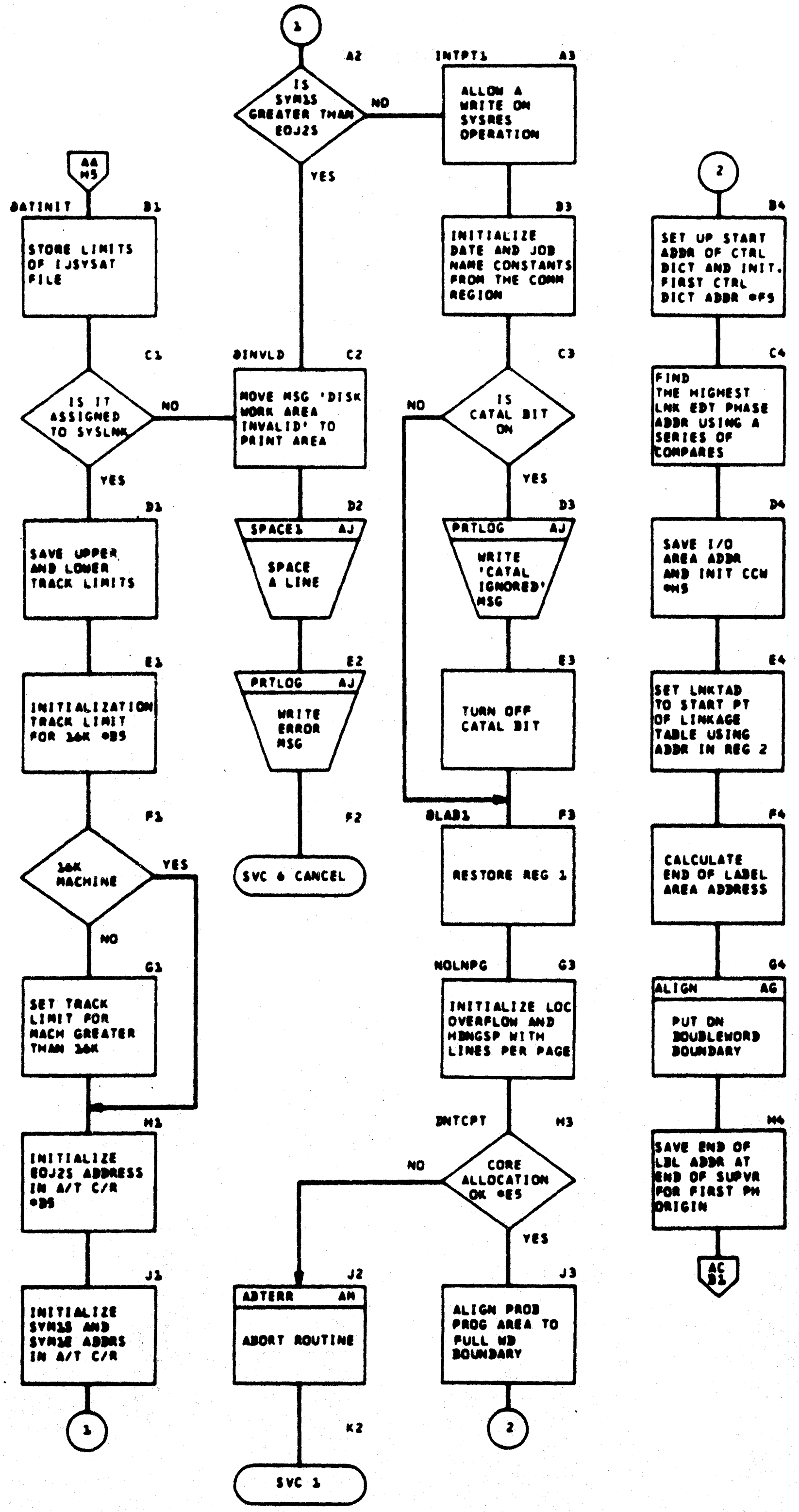


Chart AB. Initialization (ATLEDT) (Refer to Chart 02)



*B5
THERE IS ALWAYS ENOUGH ROOM ON THE AUTOTEST FILE FOR END OF JOB DUMP. 3 TRACKS ON 16K MACHINES, 8 TRACKS ON LARGER MACHINES

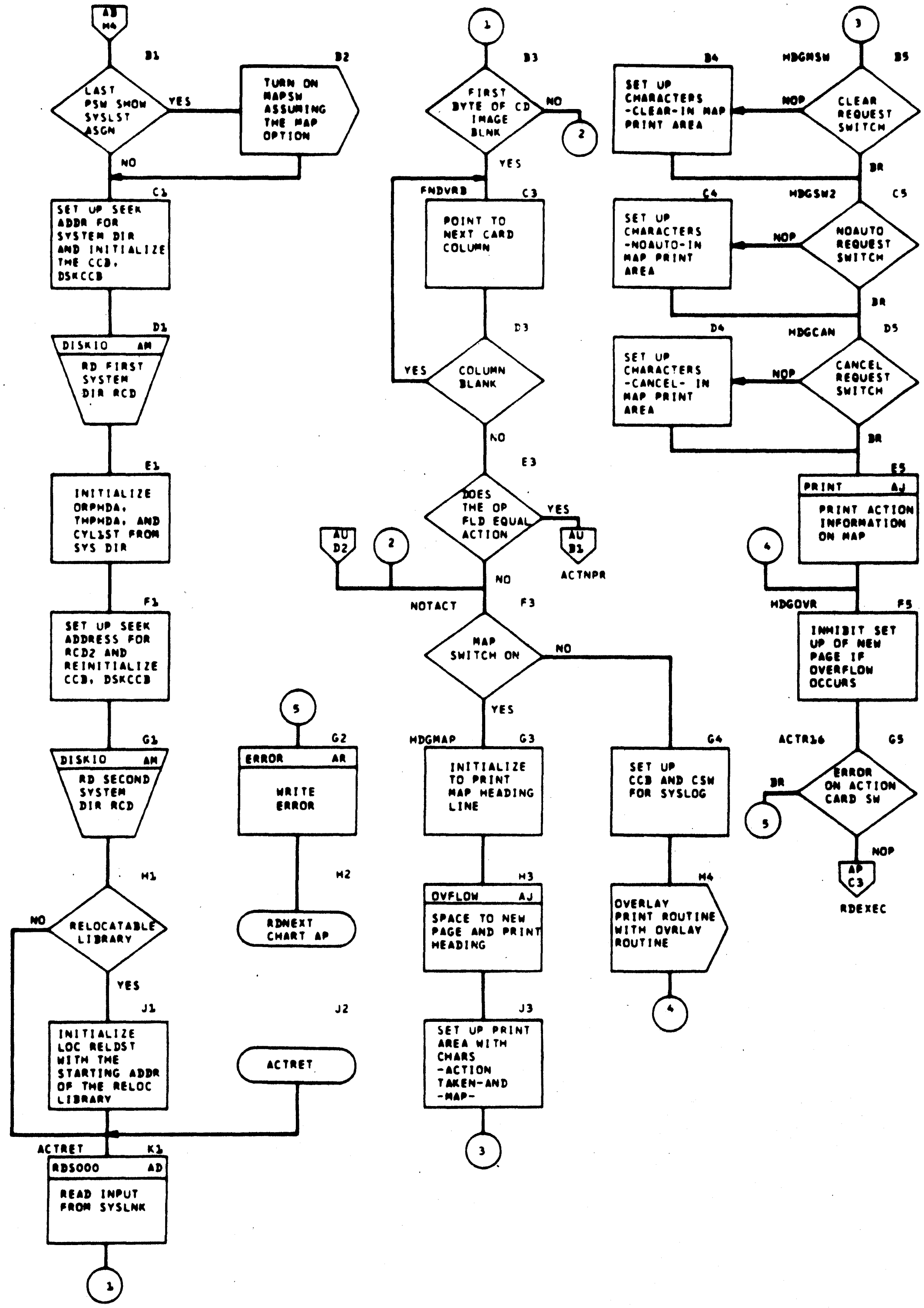
*D5
THE EOJ2S ADDR IS START ADDR OF EOJ DUMP. IT IS EQUAL TO EOF ADDR FOR IJSYSAT LESS TRACK LIMIT.

*E5
ADDR OF UPPERMOST BYTE OF PROBLEM PROGRAM AREA IS NOT LESS THAN ADDR OF LOGICAL END OF MAIN STORAGE.

*F5
1. CTLDAD IS INITIALIZED WITH THE HIGHEST PROBLEM PROGRAM ADDRESS
2. CDENT1 IS INITIALIZED TO THE FIRST ADDR AVAILABLE FOR A CONTROL DICTIONARY ENTRY.

*H5
WHEN LOADING LINKAGE EDITOR PHASES INDIVIDUALLY, THE I/O AREA IS ORIGINED AT THE END OF THE LONGEST LINKAGE EDITOR PHASE.

Chart AC. Initialization (ATLEDT) (Refer to Chart 02)



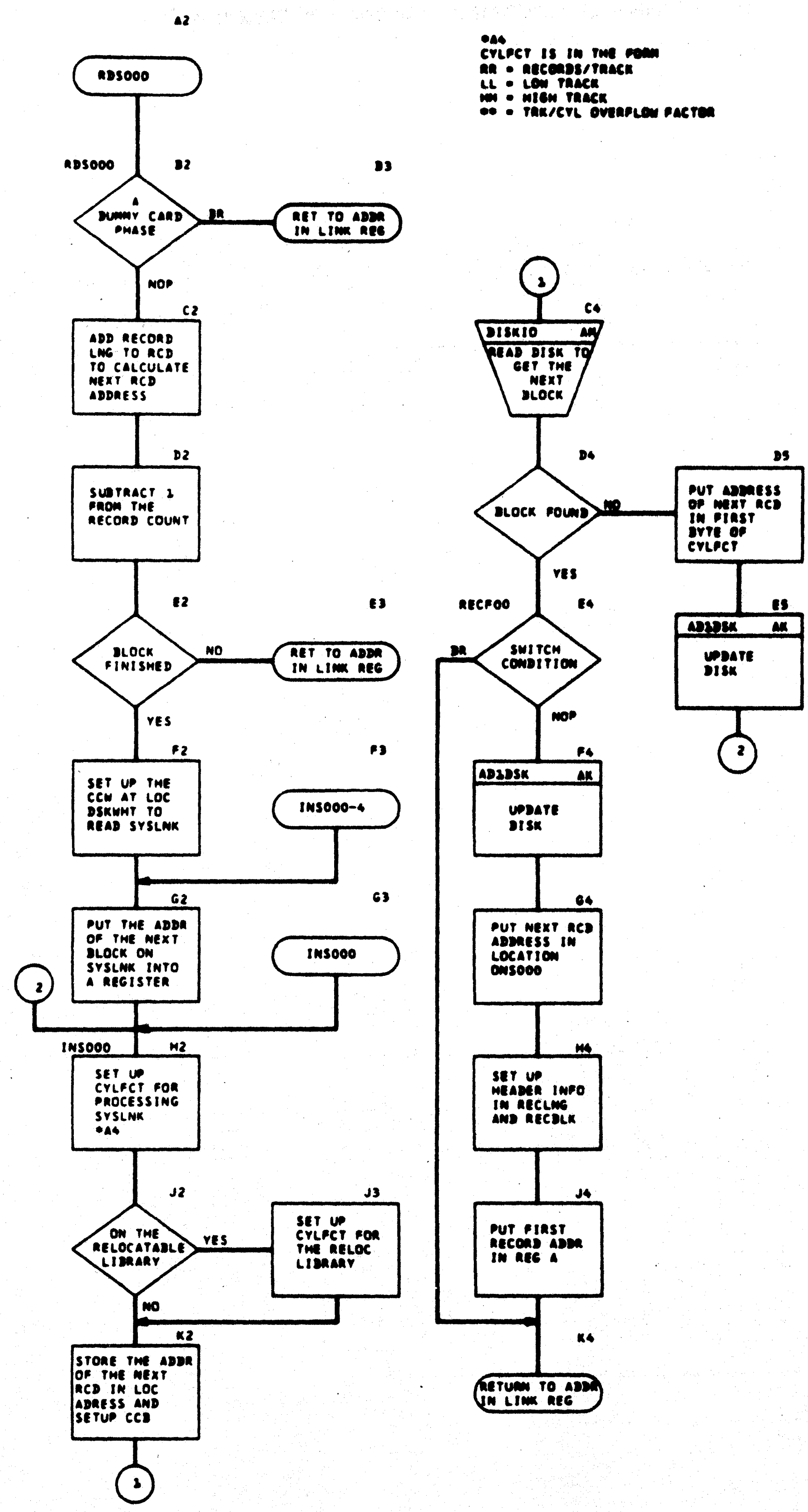

*A4
CYLFCT IS IN THE FORM
RR = RECORDS/TRACK
LL = LOW TRACK
HH = HIGH TRACK
** = TRK/CYL OVERFLOW FACTOR
RDS000
A DUMMY CARD PHASE
BR
NOP
RET TO ADDR IN LINK REG
ADD RECORD LNG TO RCD TO CALCULATE NEXT RCD ADDRESS
SUBTRACT 1 FROM THE RECORD COUNT
BLOCK FINISHED
NO
YES
RET TO ADDR IN LINK REG
SET UP THE CCW AT LOC DSKWHT TO READ SYSLNK
INS000-4
PUT THE ADDR OF THE NEXT BLOCK ON SYSLNK INTO A REGISTER
INS000
INS000
SET UP CYLFCT FOR PROCESSING SYSLNK *A4
ON THE RELOCATABLE LIBRARY
YES
NO
SET UP CYLFCT FOR THE RELOC LIBRARY
STORE THE ADDR OF THE NEXT RCD IN LOC ADRESS AND SETUP CCB
DISKIO AM
READ DISK TO GET THE NEXT BLOCK
BLOCK FOUND
NO
YES
PUT ADDRESS OF NEXT RCD IN FIRST BYTE OF CYLFCT
AD1DSK AK
UPDATE DISK
RECF00
SWITCH CONDITION
BR
NOP
AD1DSK AK
UPDATE DISK
PUT NEXT RCD ADDRESS IN LOCATION ONS000
SET UP HEADER INFO IN RECLNG AND RECBLK
PUT FIRST RECORD ADDR IN REG A
RETURN TO ADDR IN LINK REG

Chart AE. LTCDAD, LTESID, and LTCDNO Subroutines (ATLEDT) (Refer to Chart 02)

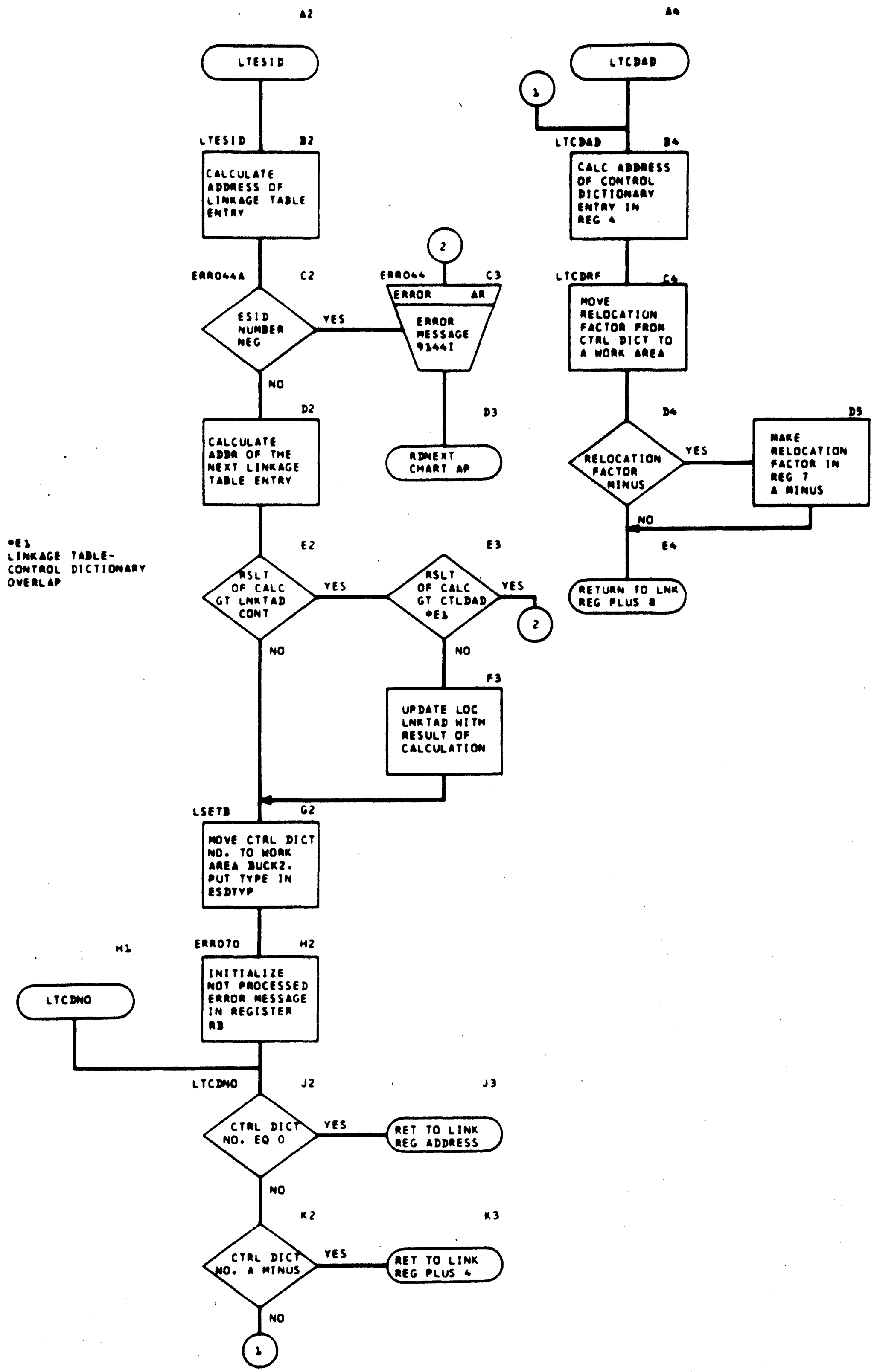


Chart AF. SRCHCD Subroutine (ATLEDT) (Refer to Chart 02)

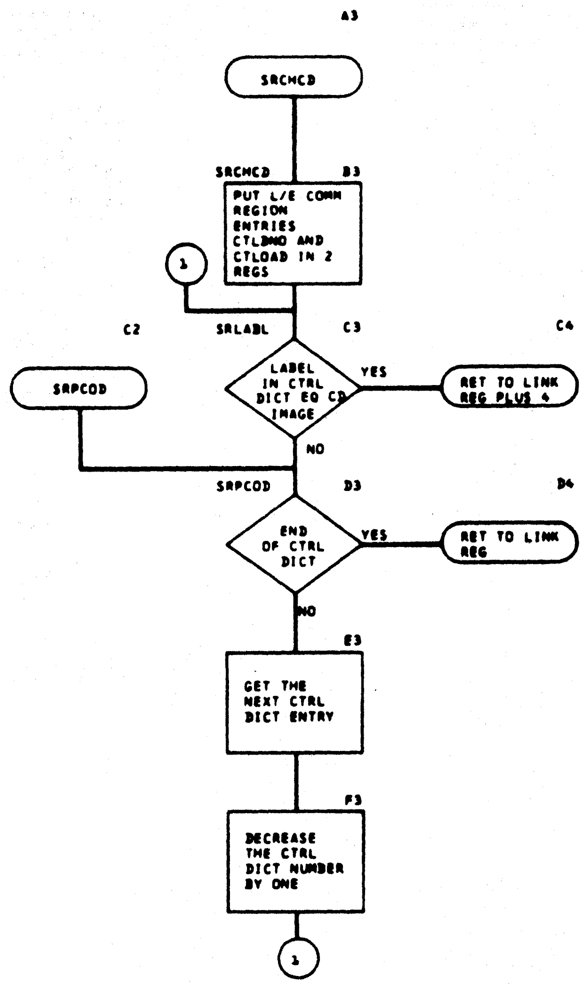
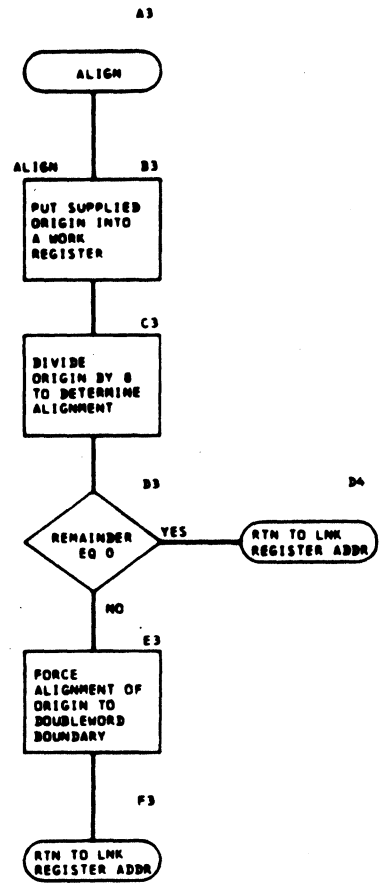


Chart AG. ALIGN Subroutine (ATLEDT) (Refer to Chart 02)



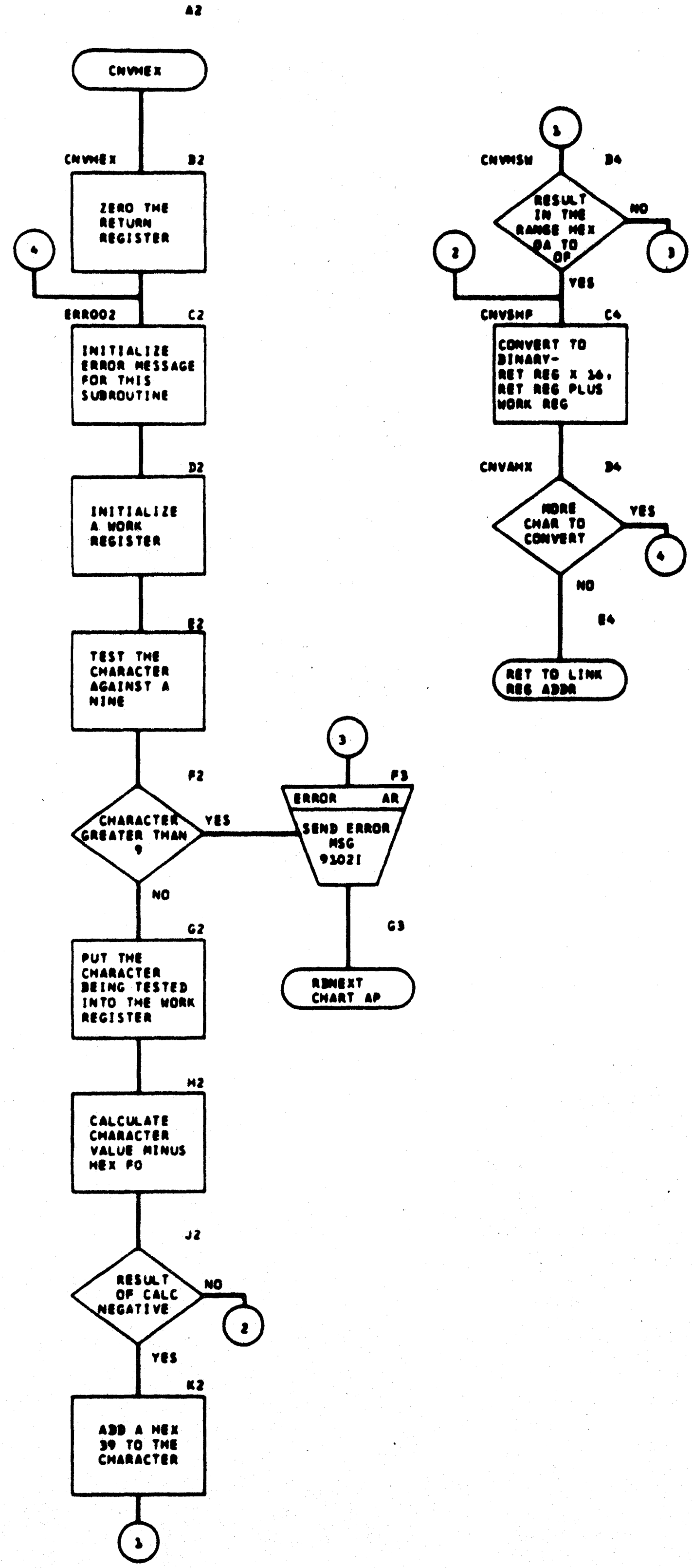

A2
CNVHEX
CNVHEX B2
ZERO THE RETURN REGISTER
4
ERR002 C2
INITIALIZE ERROR MESSAGE FOR THIS SUBROUTINE
D2
INITIALIZE A WORK REGISTER
E2
TEST THE CHARACTER AGAINST A NINE
F2
CHARACTER GREATER THAN 9
YES
NO
3
F3
ERROR AR
SEND ERROR MSG 9102I
G3
RBNEXT CHART AP
G2
PUT THE CHARACTER BEING TESTED INTO THE WORK REGISTER
H2
CALCULATE CHARACTER VALUE MINUS HEX F0
J2
RESULT OF CALC NEGATIVE
NO
2
YES
K2
ADD A HEX 39 TO THE CHARACTER
1
1
CNVHSW B4
RESULT IN THE RANGE HEX 0A TO 0F
NO
3
2
YES
CNVSHF C4
CONVERT TO BINARY- RET REG X 16, RET REG PLUS WORK REG
CNVAHX D4
MORE CHAR TO CONVERT
YES
4
NO
E4
RET TO LINK REG ADDR

Chart AJ. PRTLOG, PRINT, OVFLOW, PRTLST, and SPACE1 Subroutines (ATLEDT) (Refer to Chart 02)

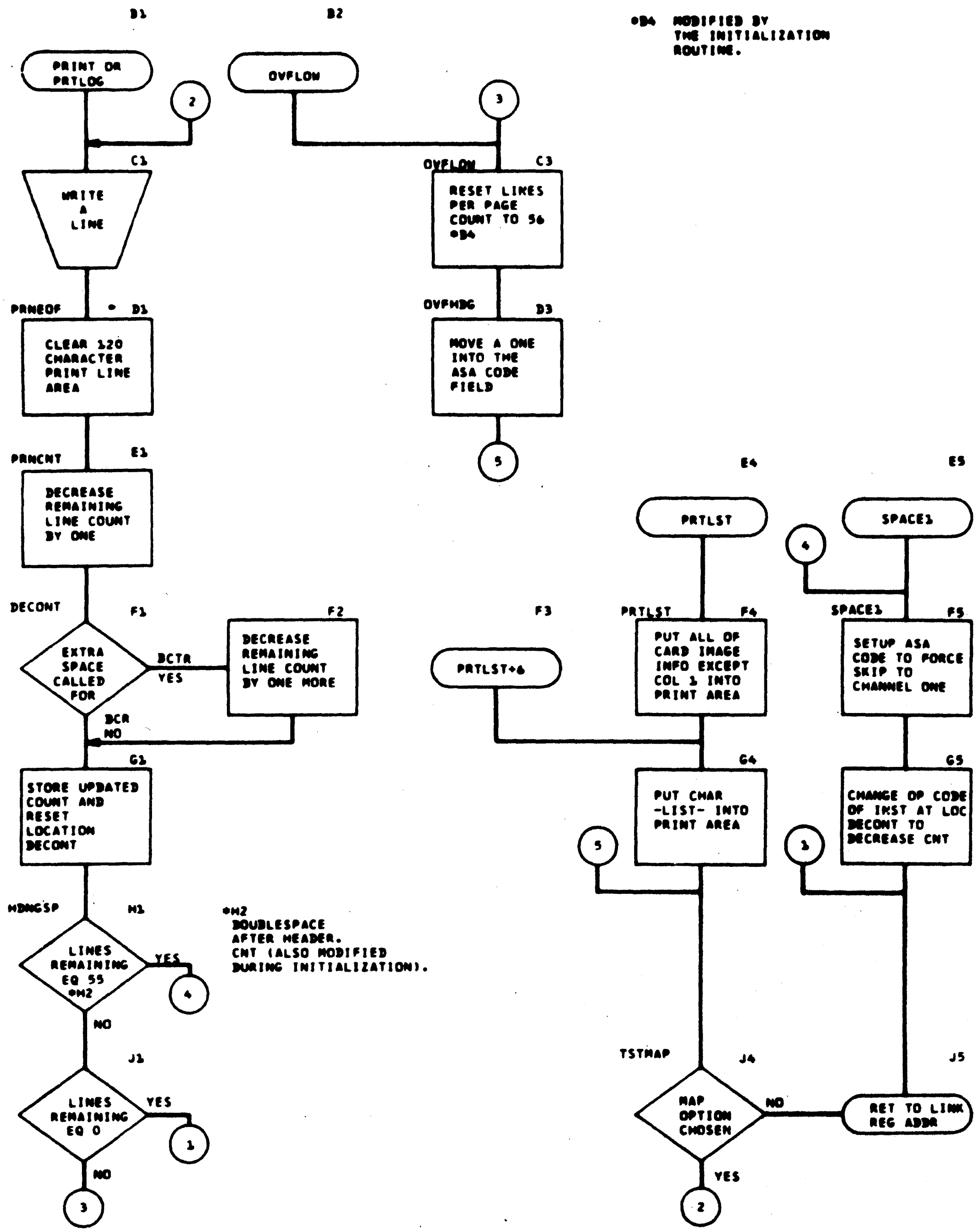


Chart AK. AD1DSK Subroutine (ATLEDT) (Refer to Chart 02)

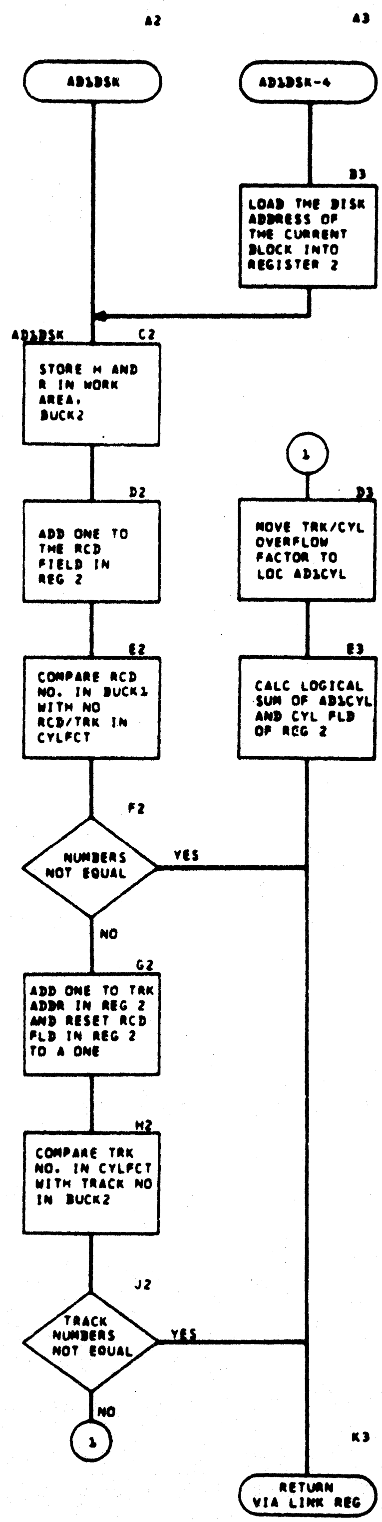


Chart AL. XTPHNO Subroutine (ATLEDT) (Refer to Chart 02)

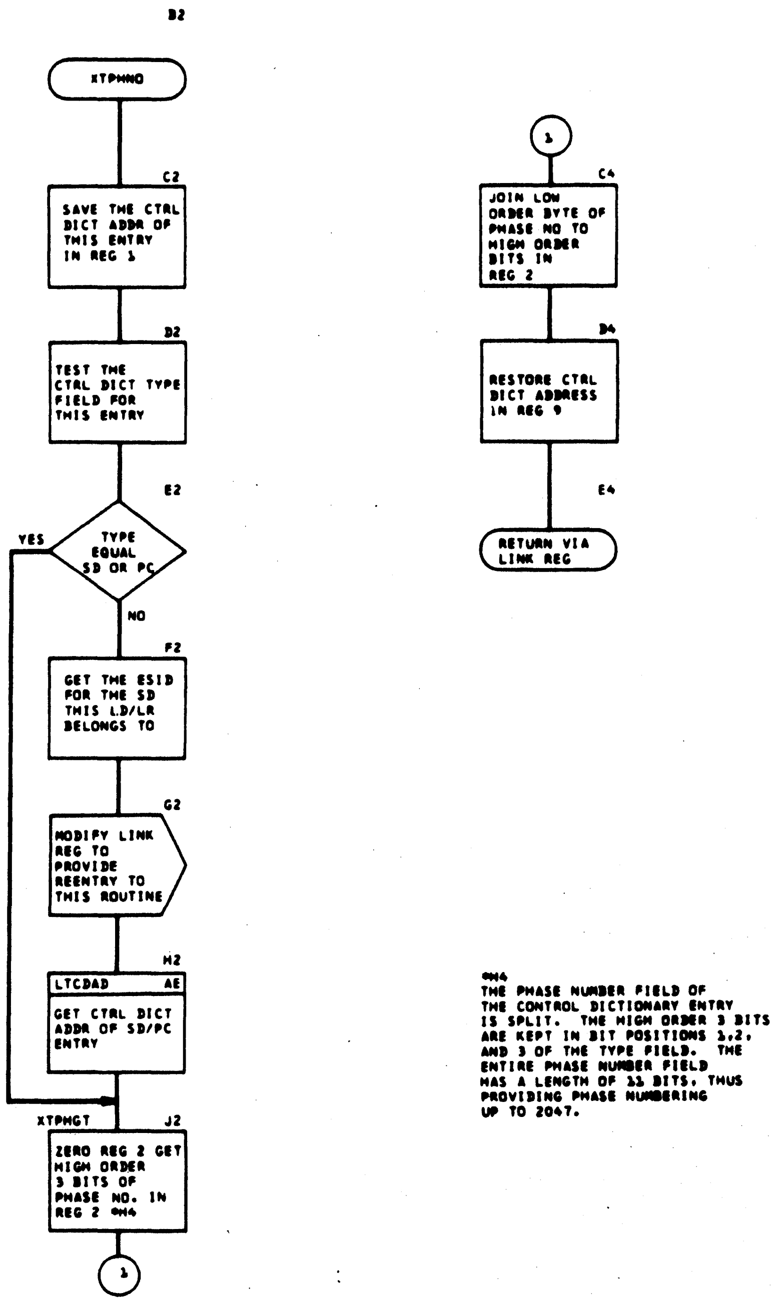


Chart AM. READ, WRITE, DISKIO, and ABTERR Subroutines (ATLEDT) (Refer to Chart 02)

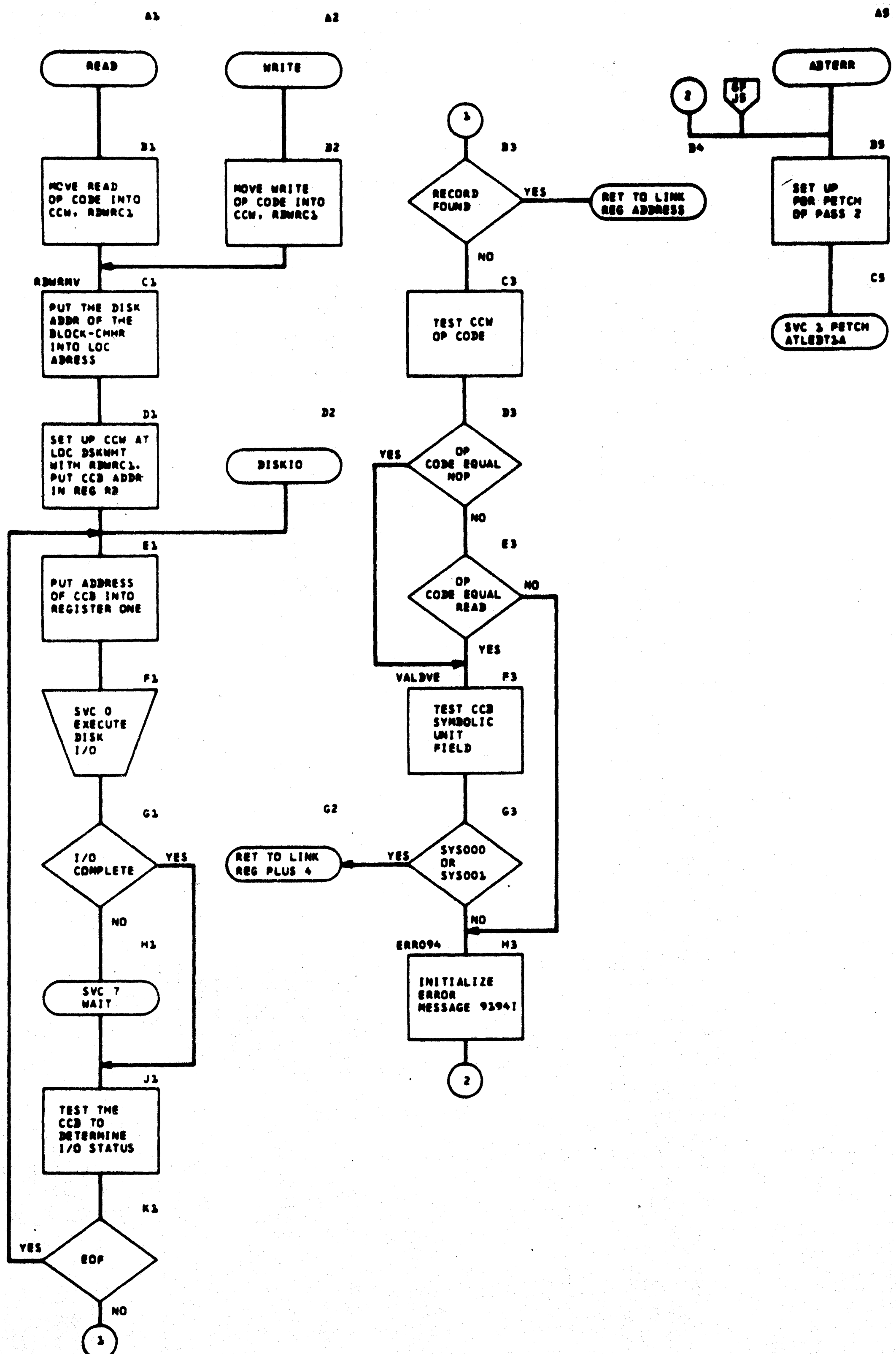


Chart AN. CDSIZE Subroutine (ATLEDT) (Refer to Chart 02)

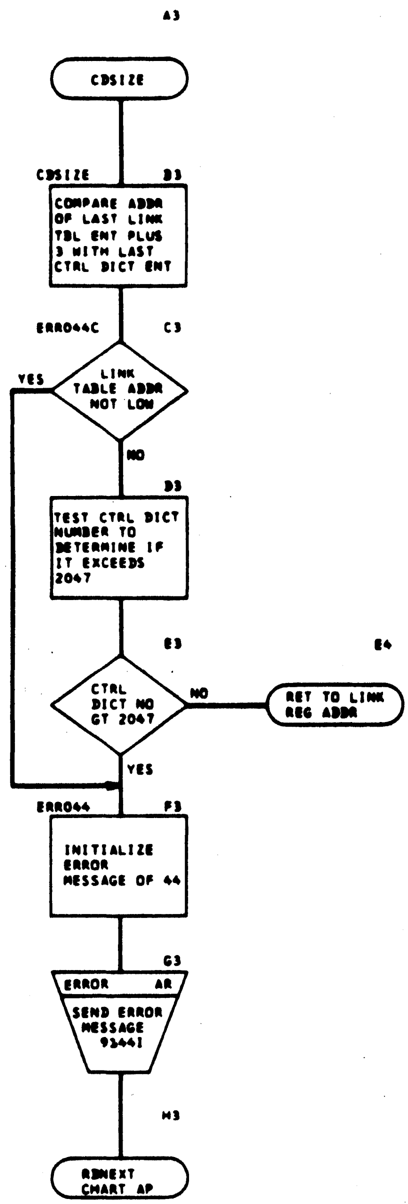
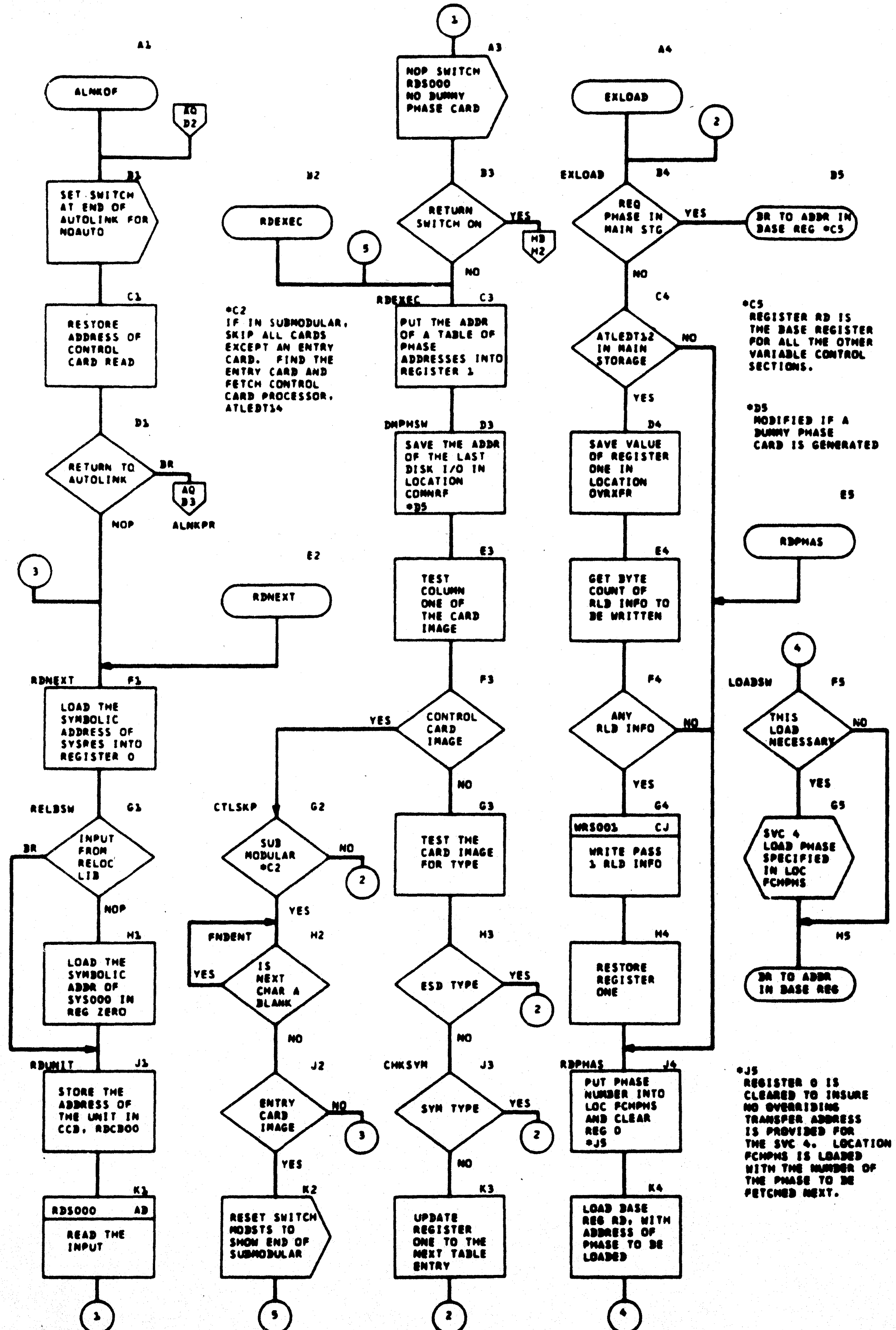


Chart AP. ALNKOF, RDEXEC, RDNEXT, and EXLOAD Subroutines (ATLEDT) (Refer to Chart 02)



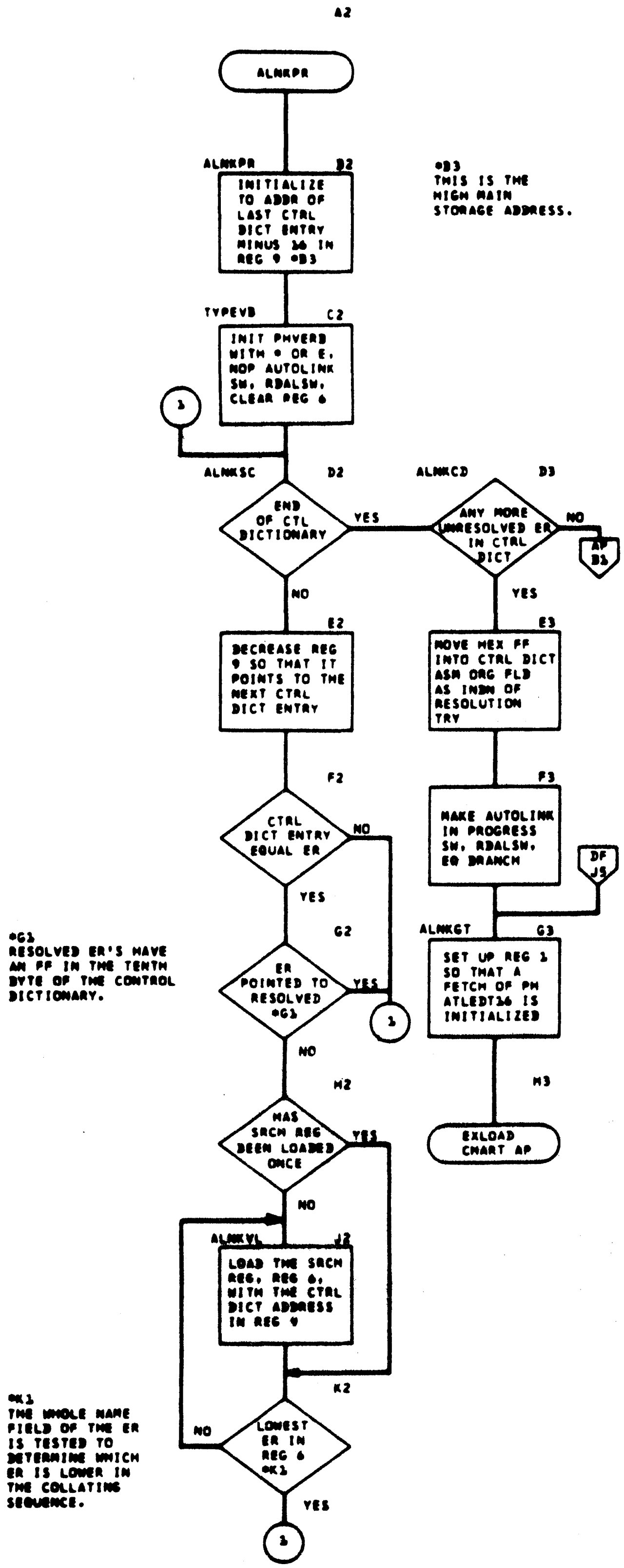

A2
ALNKPR
ALNKPR B2
INITIALIZE TO ADDR OF LAST CTRL DICT ENTRY MINUS 16 IN REG 9 *B3
*B3 THIS IS THE HIGH MAIN STORAGE ADDRESS.
TYPEVB C2
INIT PHVERB WITH * OR E. NOP AUTOLINK SW, RDALSW, CLEAR REG 6
1
ALNKSC D2
END OF CTL DICTIONARY
YES
ALNKCD D3
ANY MORE UNRESOLVED ER IN CTRL DICT
NO
AP B1
NO
E2
DECREASE REG 9 SO THAT IT POINTS TO THE NEXT CTRL DICT ENTRY
YES
E3
MOVE HEX FF INTO CTRL DICT ASM ORG FLD AS INDN OF RESOLUTION TRY
F2
CTRL DICT ENTRY EQUAL ER
NO
F3
MAKE AUTOLINK IN PROGRESS SW, RDALSW, EQ BRANCH
DF J5
YES
ALNKGT G3
*G1 RESOLVED ER'S HAVE AN FF IN THE TENTH BYTE OF THE CONTROL DICTIONARY.
G2
ER POINTED TO RESOLVED *G1
YES
1
SET UP REG 1 SO THAT A FETCH OF PH ATLEDT16 IS INITIALIZED
NO
H2
H3
HAS SRCH REG BEEN LOADED ONCE
YES
EXLOAD CHART AP
NO
ALNKVL J2
LOAD THE SRCH REG, REG 6, WITH THE CTRL DICT ADDRESS IN REG 9
K2
*K1 THE WHOLE NAME FIELD OF THE ER IS TESTED TO DETERMINE WHICH ER IS LOWER IN THE COLLATING SEQUENCE.
NO
LOWEST ER IN REG 6 *K1
YES
1

Chart AR. ERROR Subroutine (ATLEDT) (Refer to Chart 02)

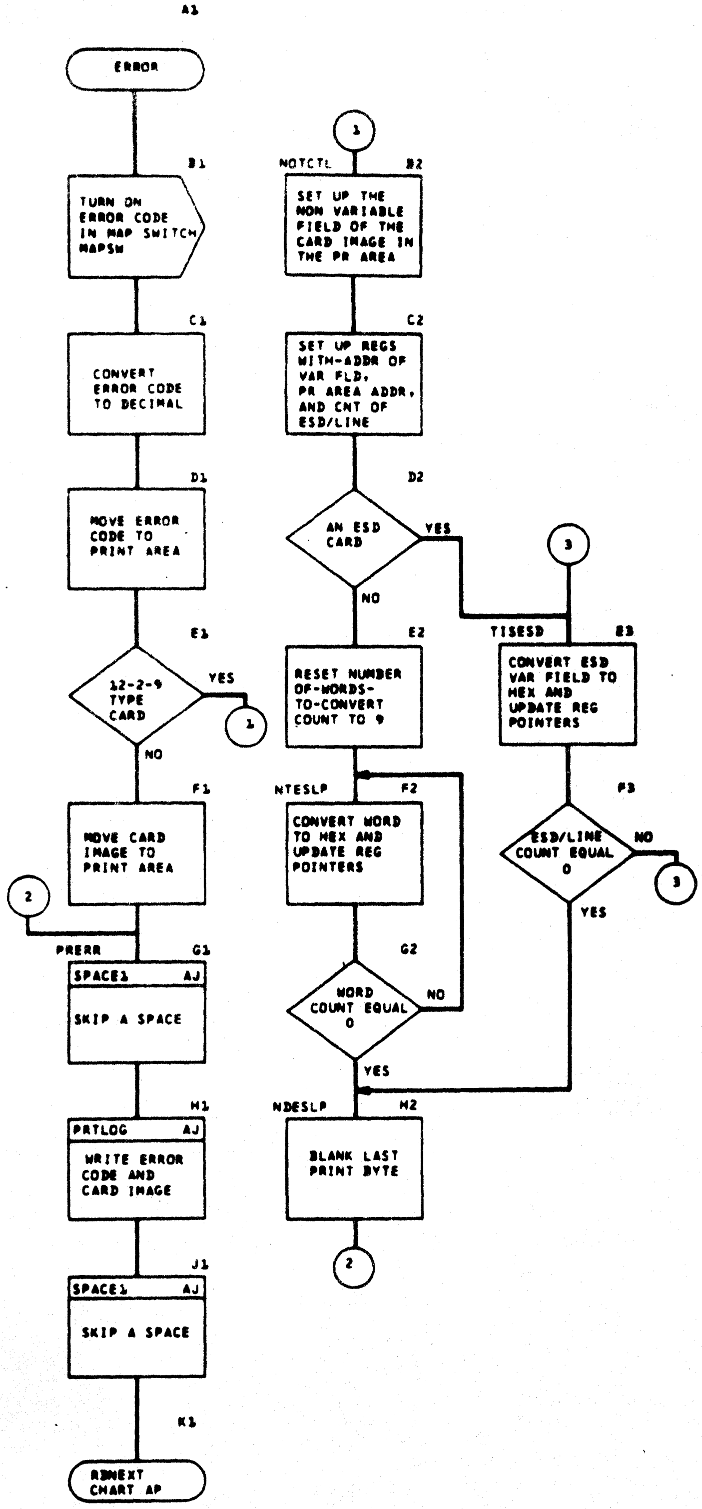
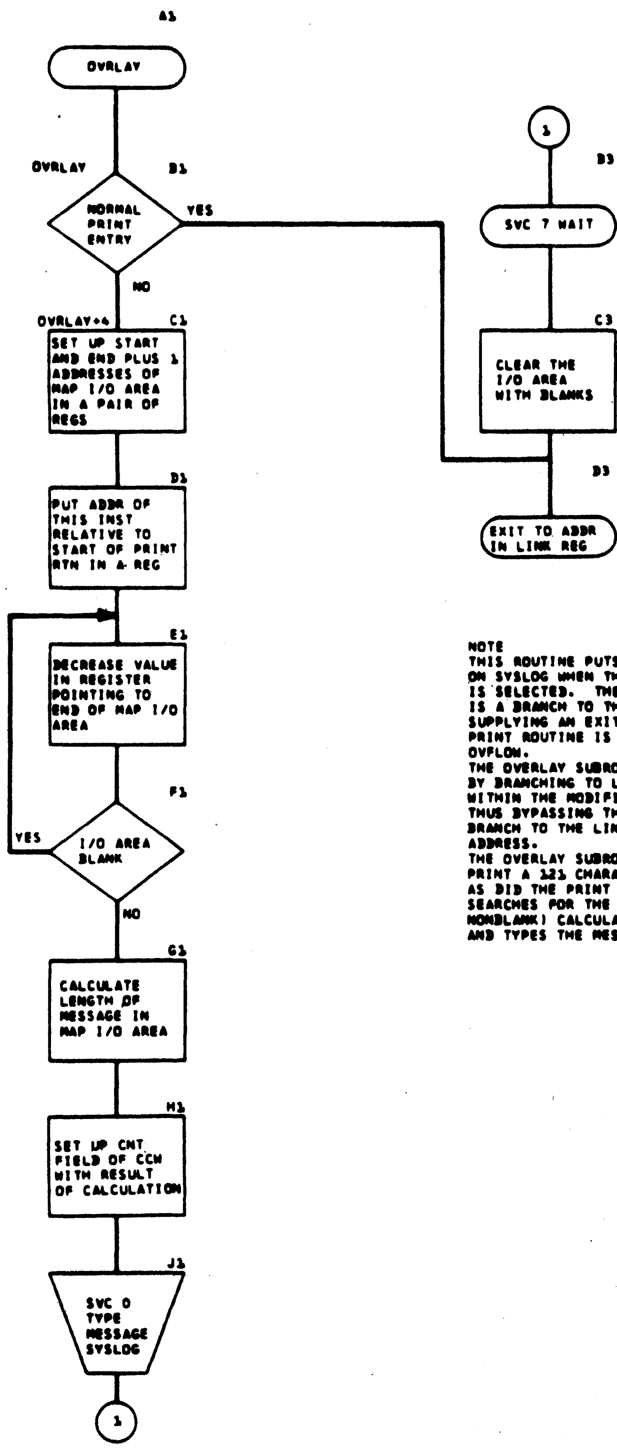


Chart A8. OVRLAY Subroutine (ATLEDT) (Refer to Chart 02)



NOTE
THIS ROUTINE PUTS OUT ERROR MESSAGES ON SYSLOG WHEN THE NOMAP OPTION IS SELECTED. THE FIRST INSTRUCTION IS A BRANCH TO THE LINK REGISTER ADDRESS, SUPPLYING AN EXIT WHEN THE MODIFIED PRINT ROUTINE IS ENTERED AT LOCATION OVFLOW.
THE OVERLAY SUBROUTINE IS ENTERED BY BRANCHING TO LOCATION PRTLOG WITHIN THE MODIFIED PRINT ROUTINE, THUS BYPASSING THE UNCONDITIONAL BRANCH TO THE LINKAGE REGISTER ADDRESS.
THE OVERLAY SUBROUTINE DOES NOT PRINT A 121 CHARACTER LONG MESSAGE AS DID THE PRINT ROUTINE. IT SEARCHES FOR THE MESSAGE (A NONBLANK) CALCULATES ITS LENGTH, AND TYPES THE MESSAGE ON SYSLOG.

Chart AT. DERDAD Subroutine (ATLEDT) (Refer to Chart 02)

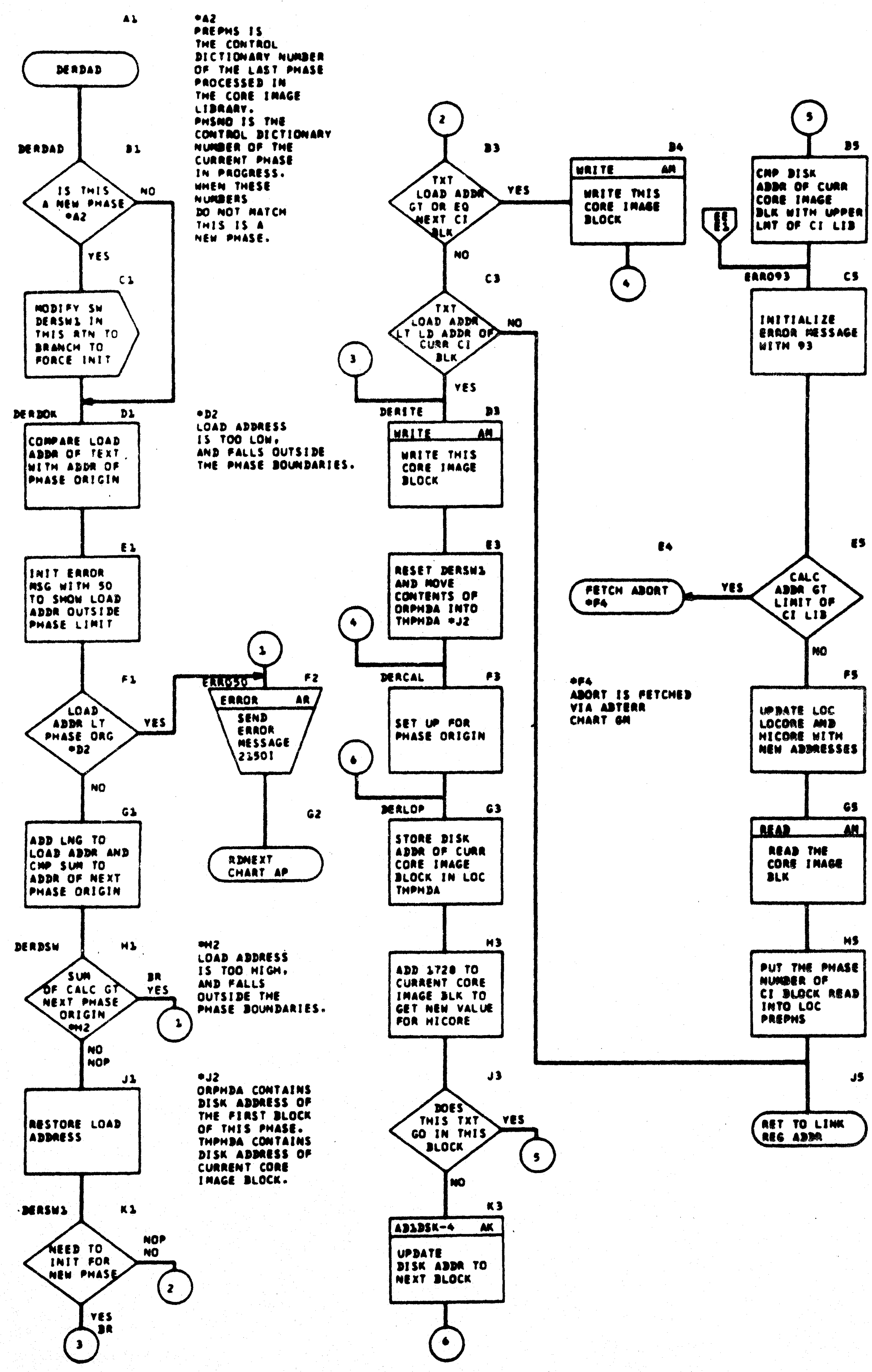


Chart AU. ACTMPR Subroutine (ATLEDT) (Refer to Chart 02)

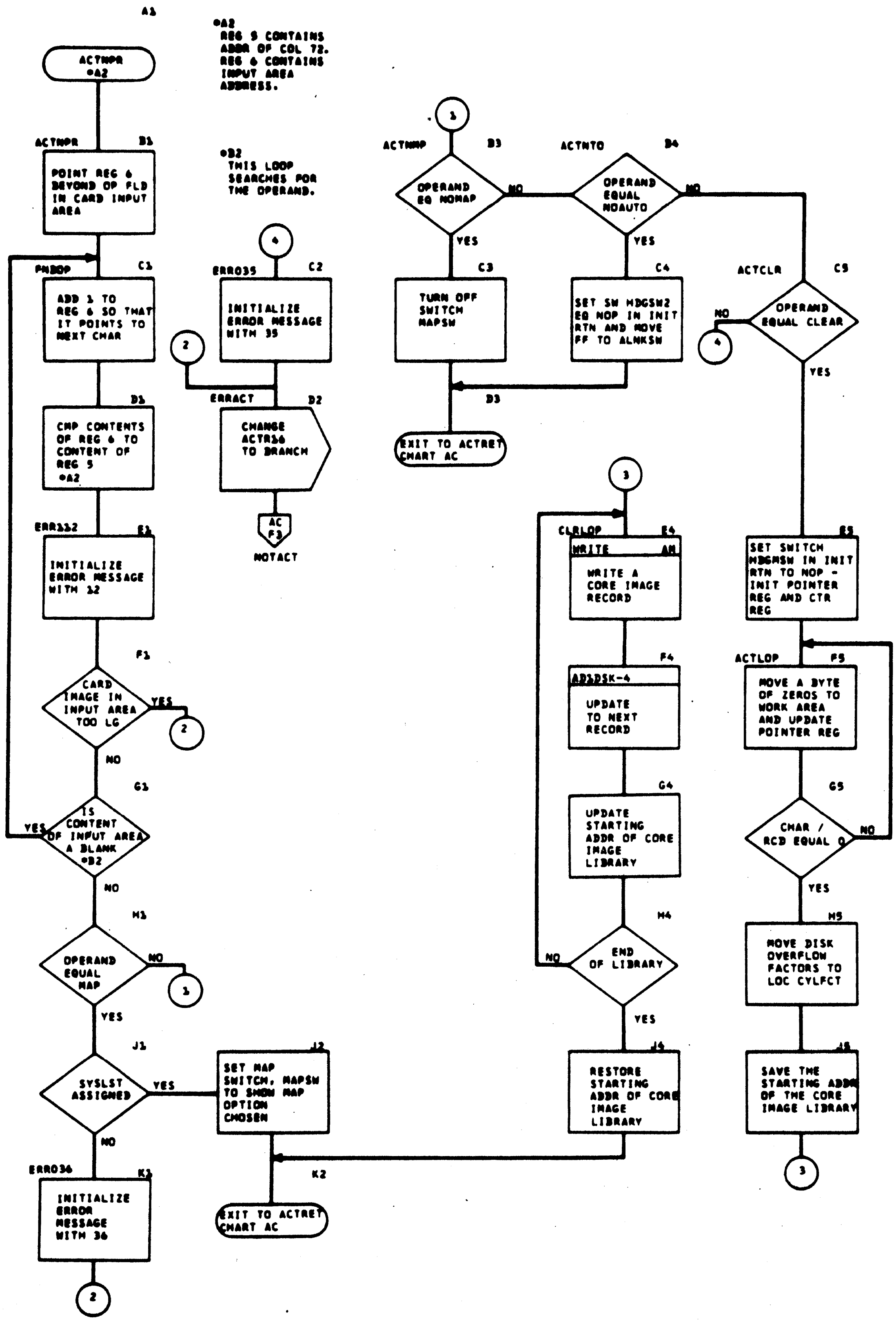


Chart BA. ESD Processor (ATLEDT10) (Refer to Chart 03)

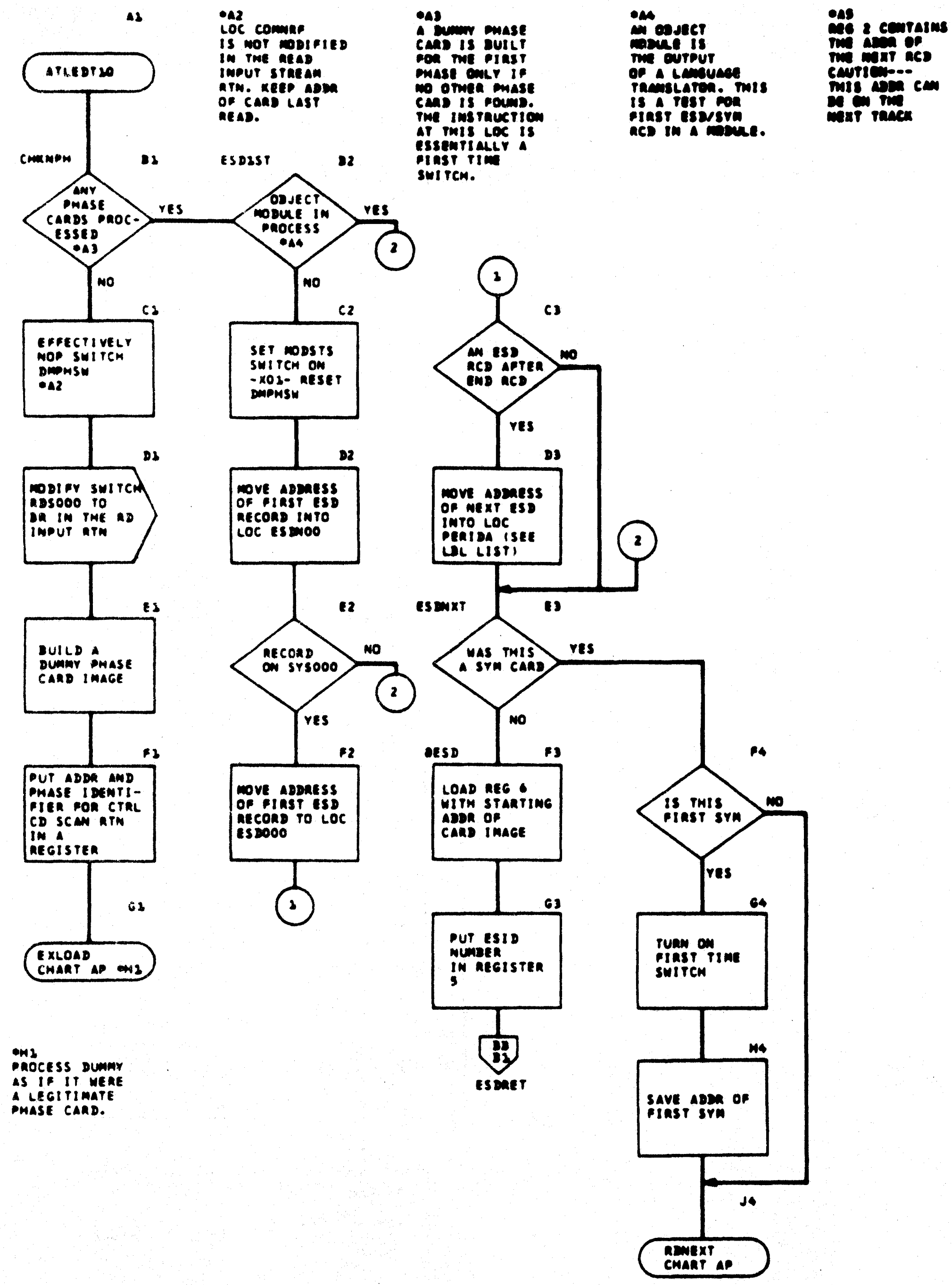


Chart BB. ESD Processor (ATLEDT10) (Refer to Chart 03)

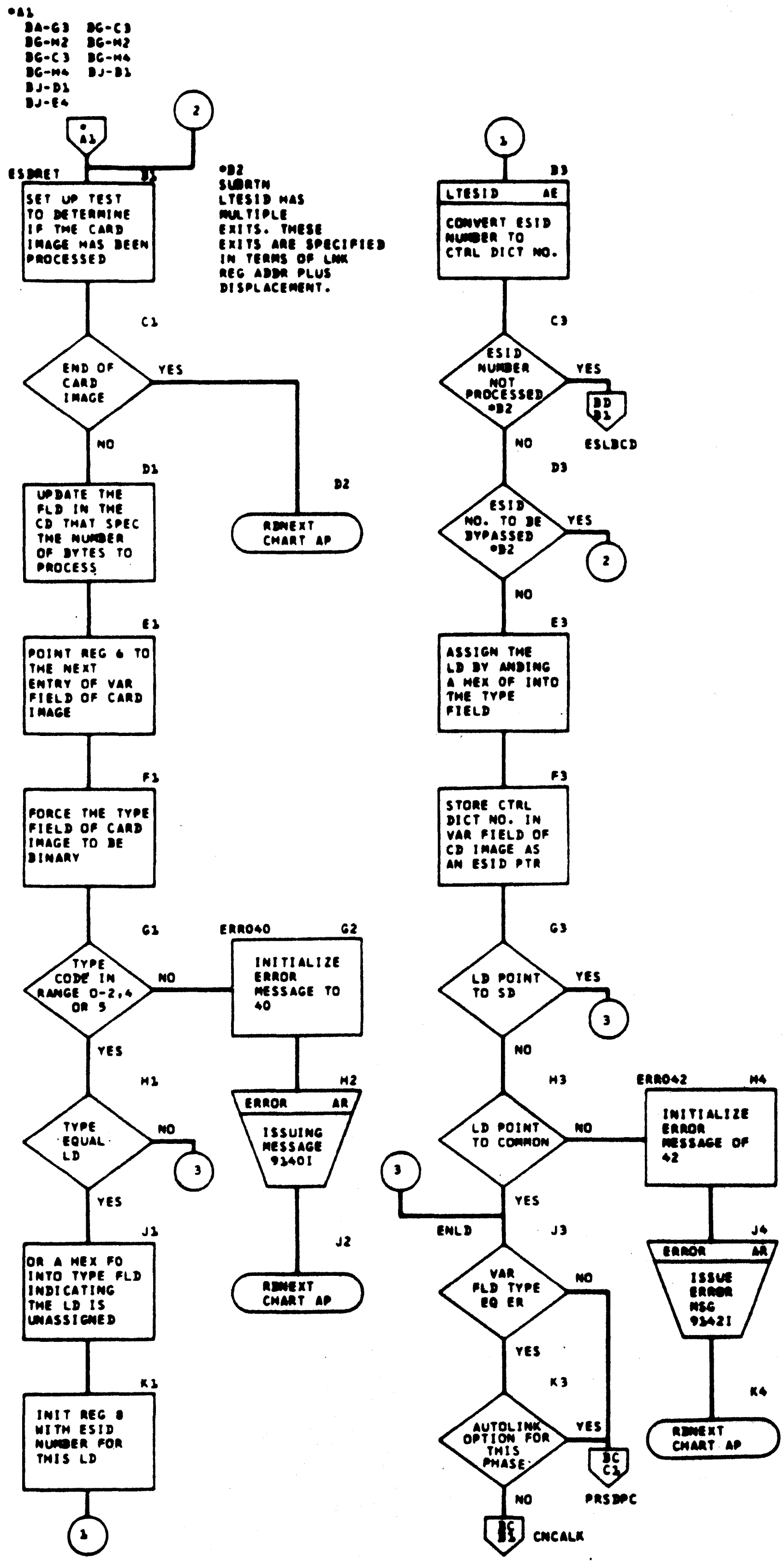


Chart BC. ESD Processor (ATLEDT10) (Refer to Chart 03)

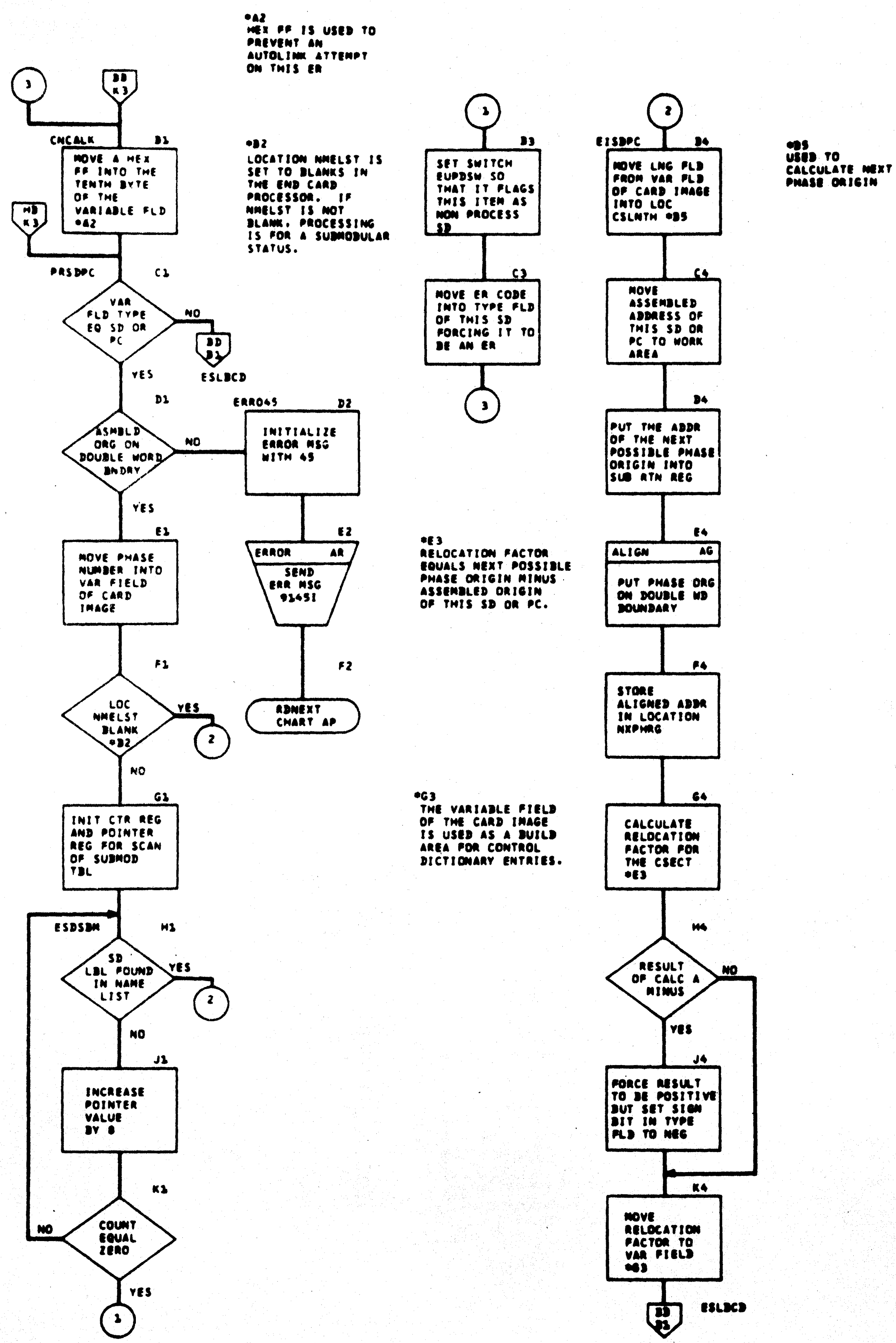


Chart BD. ESD Processor, Process ESD Items Against Control Dictionary (ATLEDT10) (Refer to Chart 03)

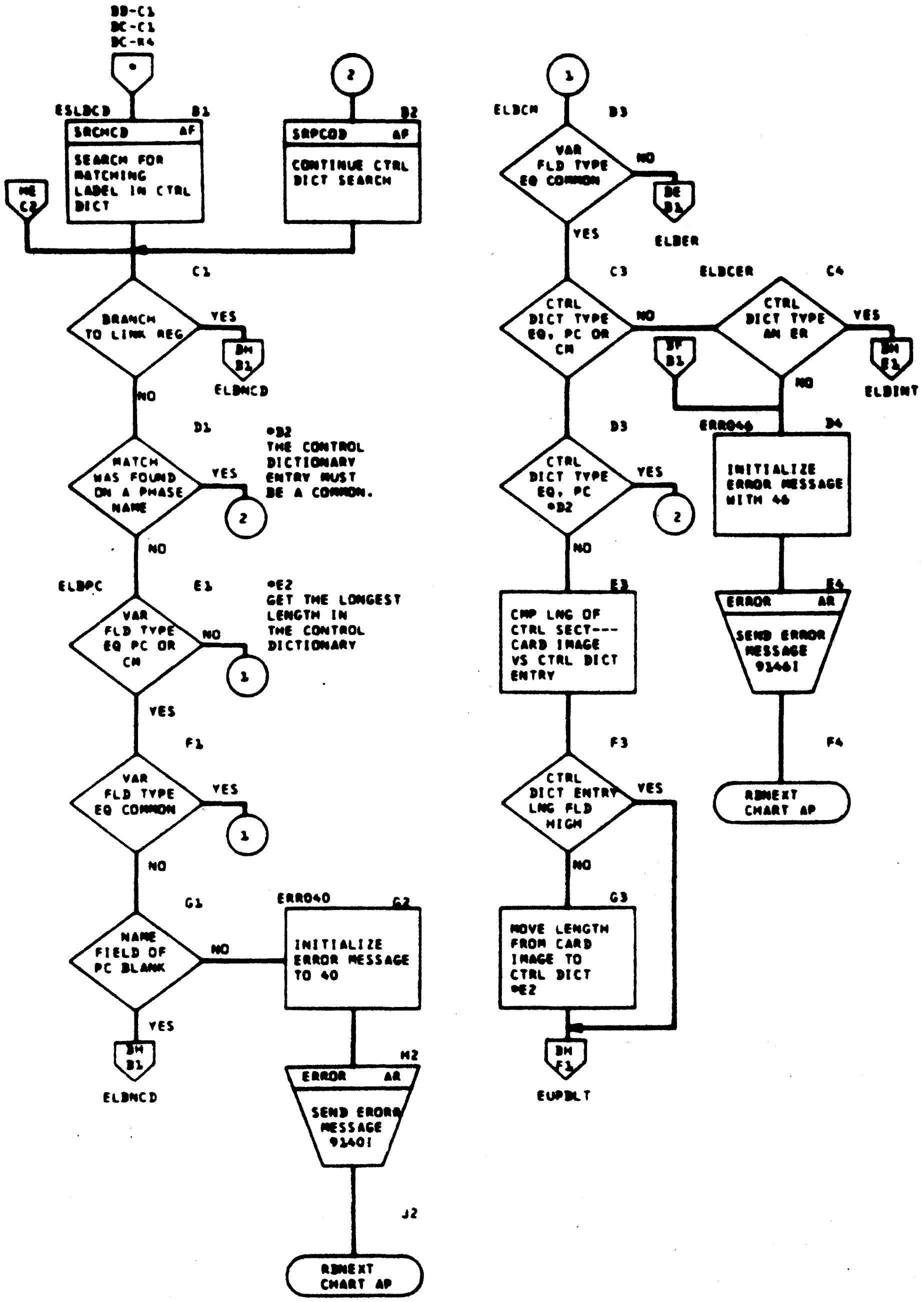


Chart BE. ESD Processor, Process ER (ATLEDT10) (Refer to Chart 03)

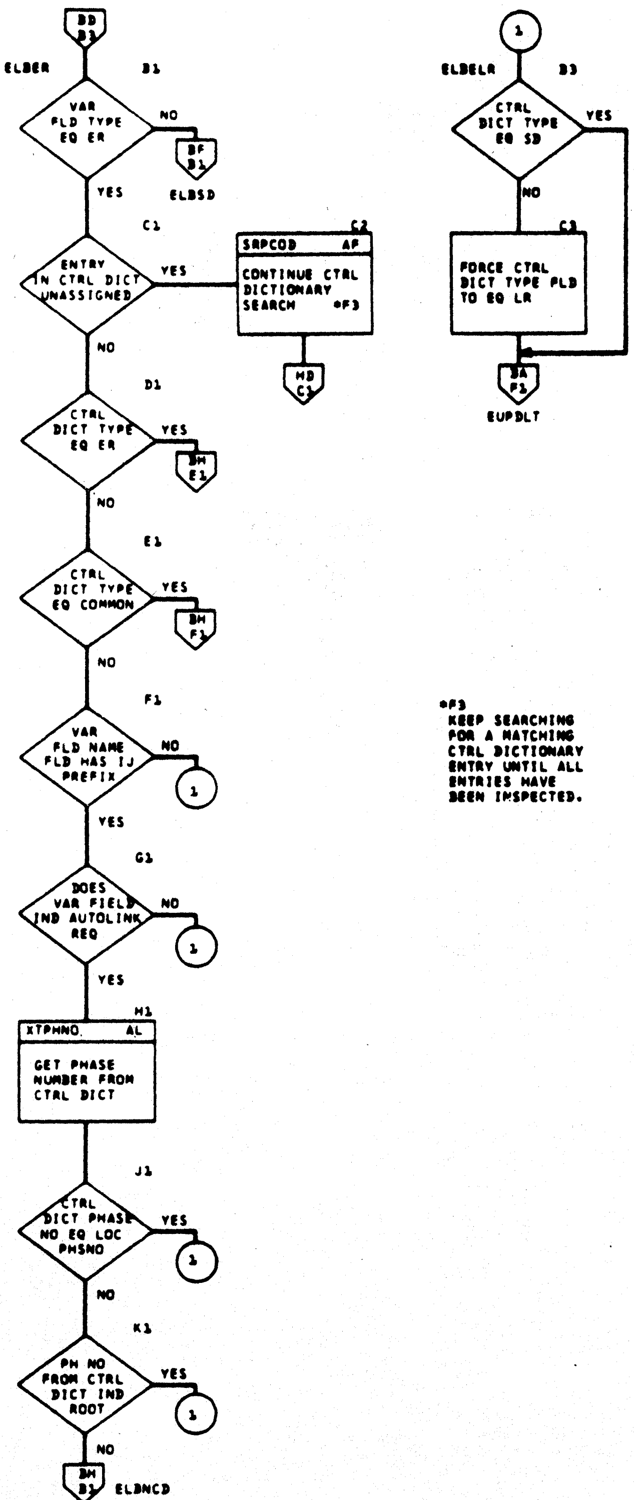


Chart BF. ESD Processor, Process SD (ATLEDT10) (Refer to Chart 03)

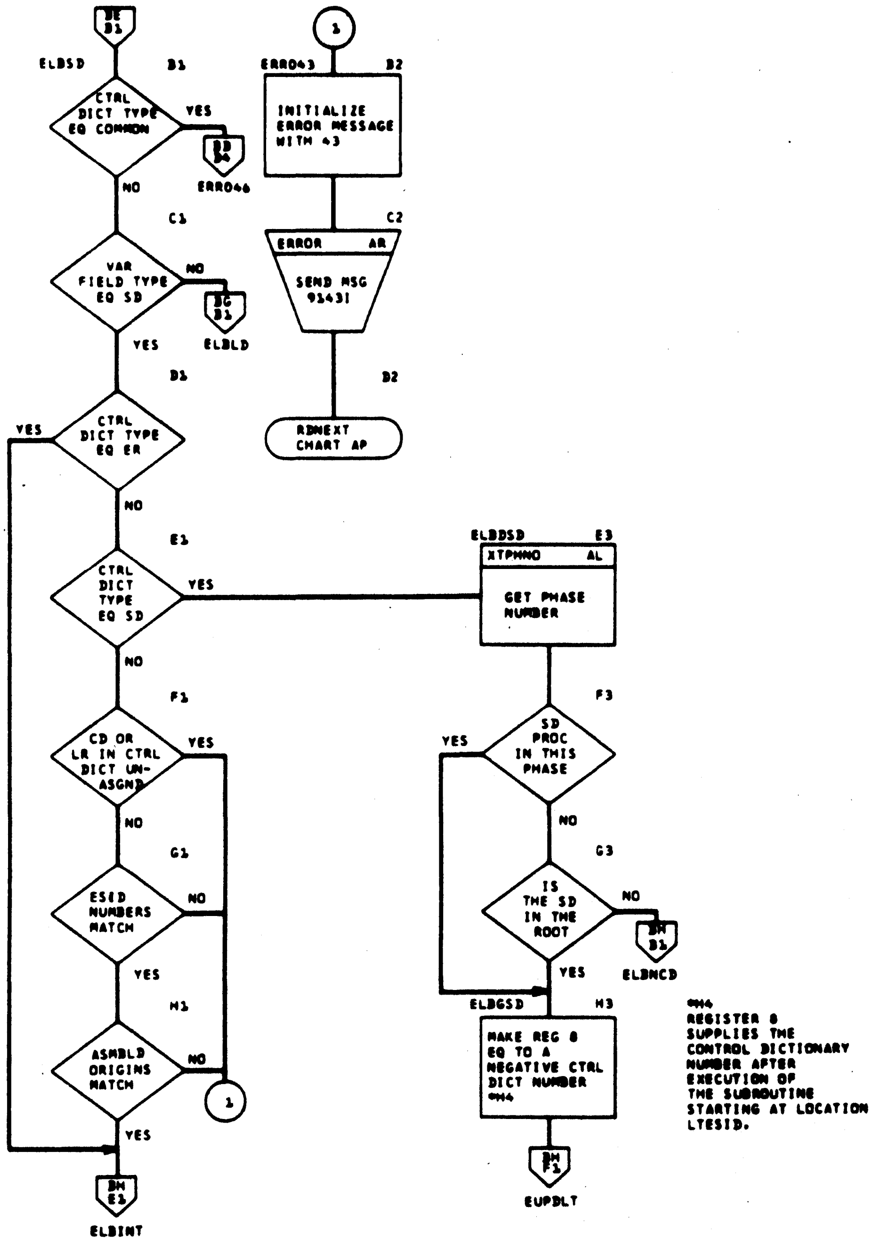


Chart BG. ESD Processor, Process LD/LR (ATLEDT10) (Refer to Chart 03)

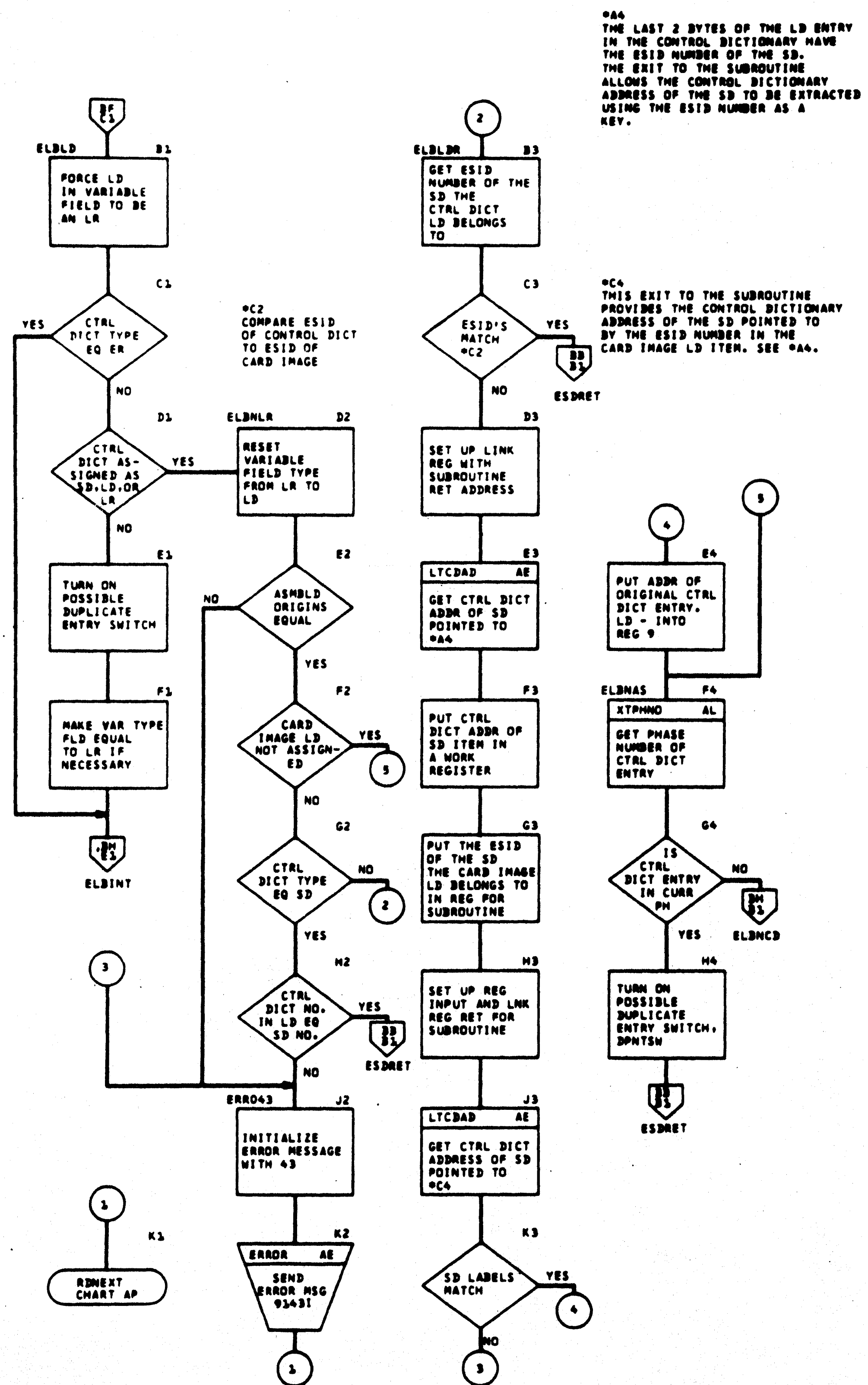


Chart BH. ESD Processor, Update Linkage Table and Control Dictionary (ATLEDT10) (Refer to Chart 03) (Part 1 of 2)

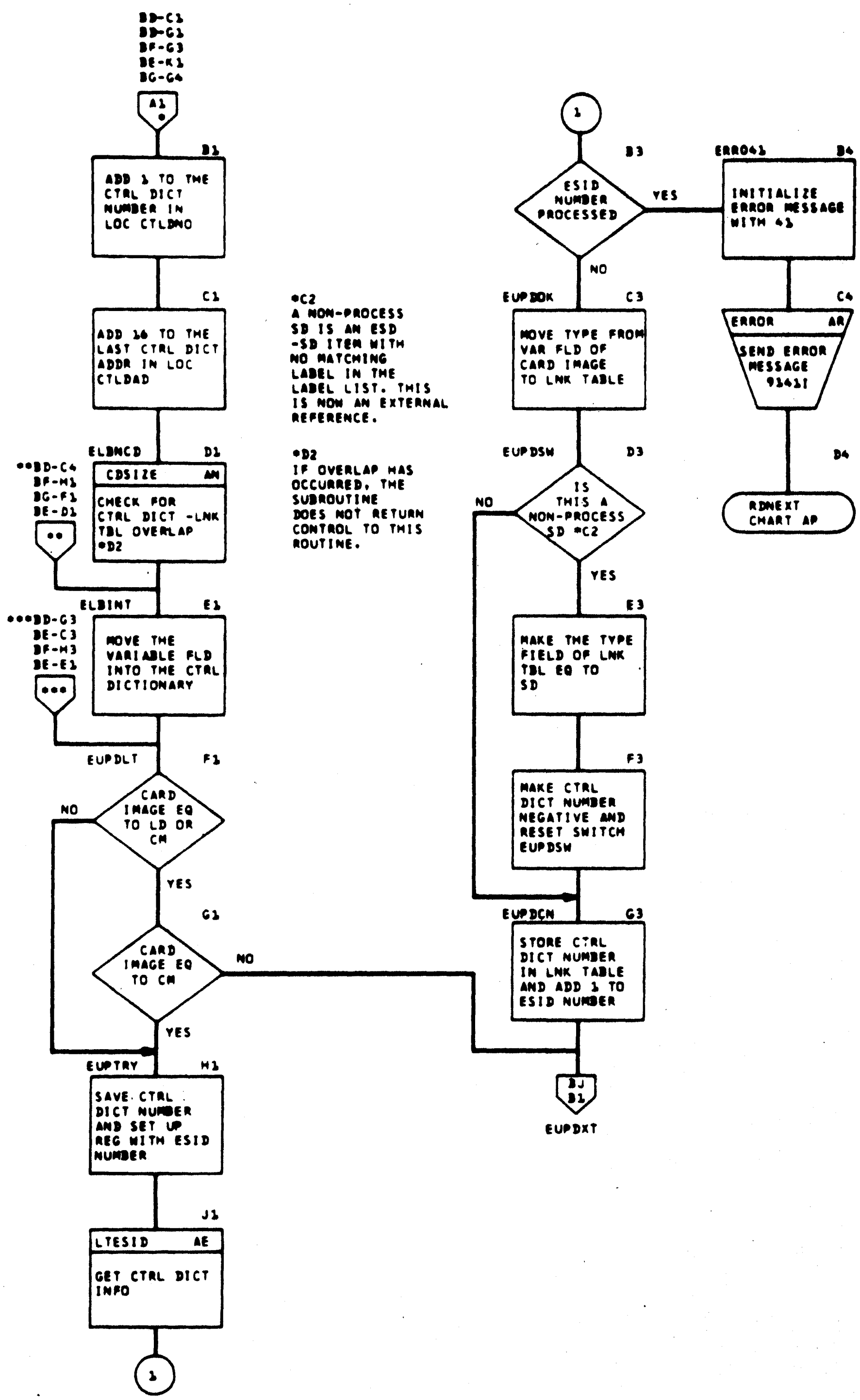


Chart BJ. ESD Processor, Update Linkage Table and Control Dictionary (ATLEDT10) (Refer to Chart 03) (Part 2 of 2)

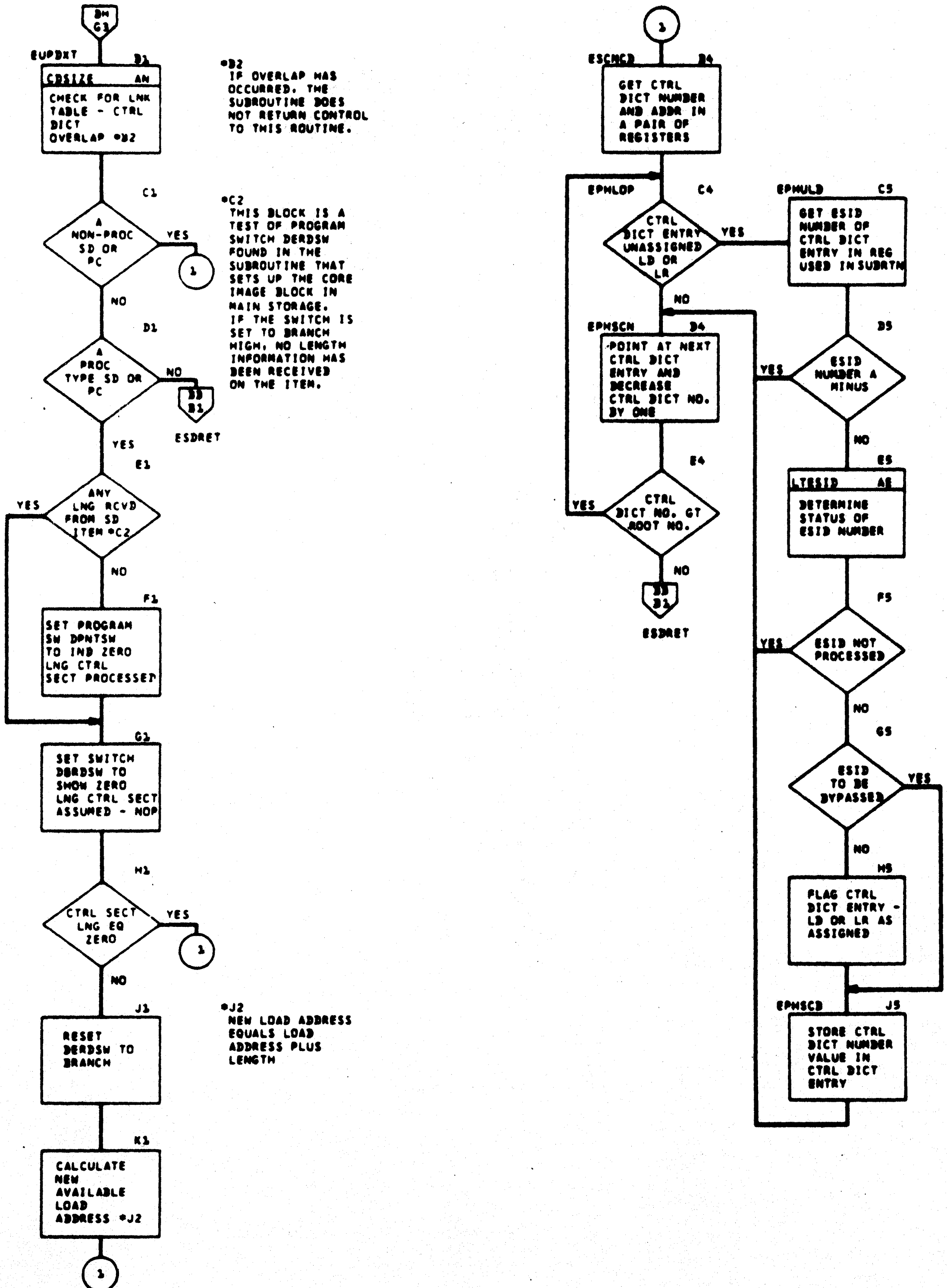


Chart CA. Initialization (ATLEDT12) (Refer to Chart 04)

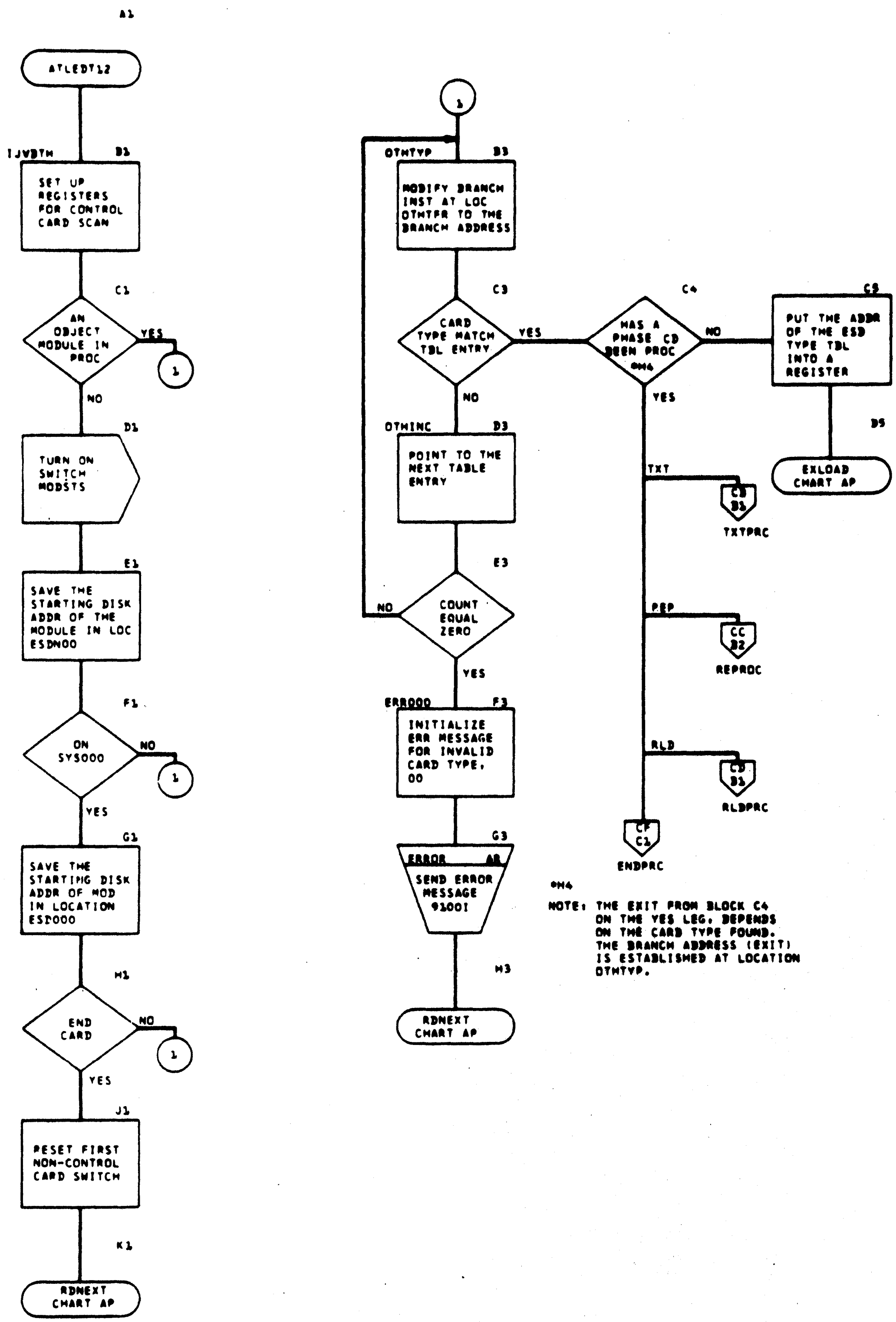


Chart CB. Text Processor (ATLEDT12) (Refer to Chart 04)

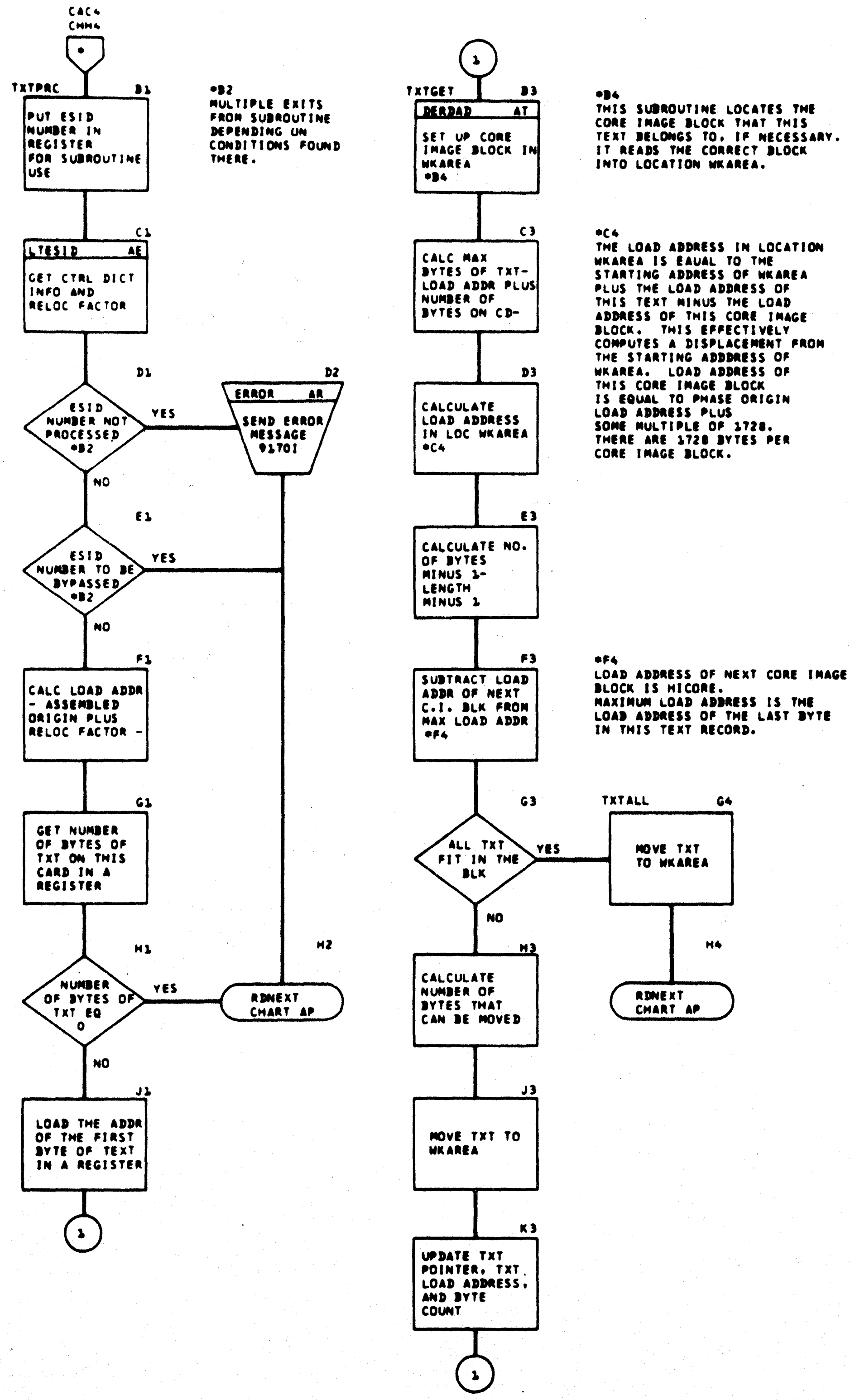


Chart CC. REP Processor (ATLEDT12) (Refer to Chart 04)

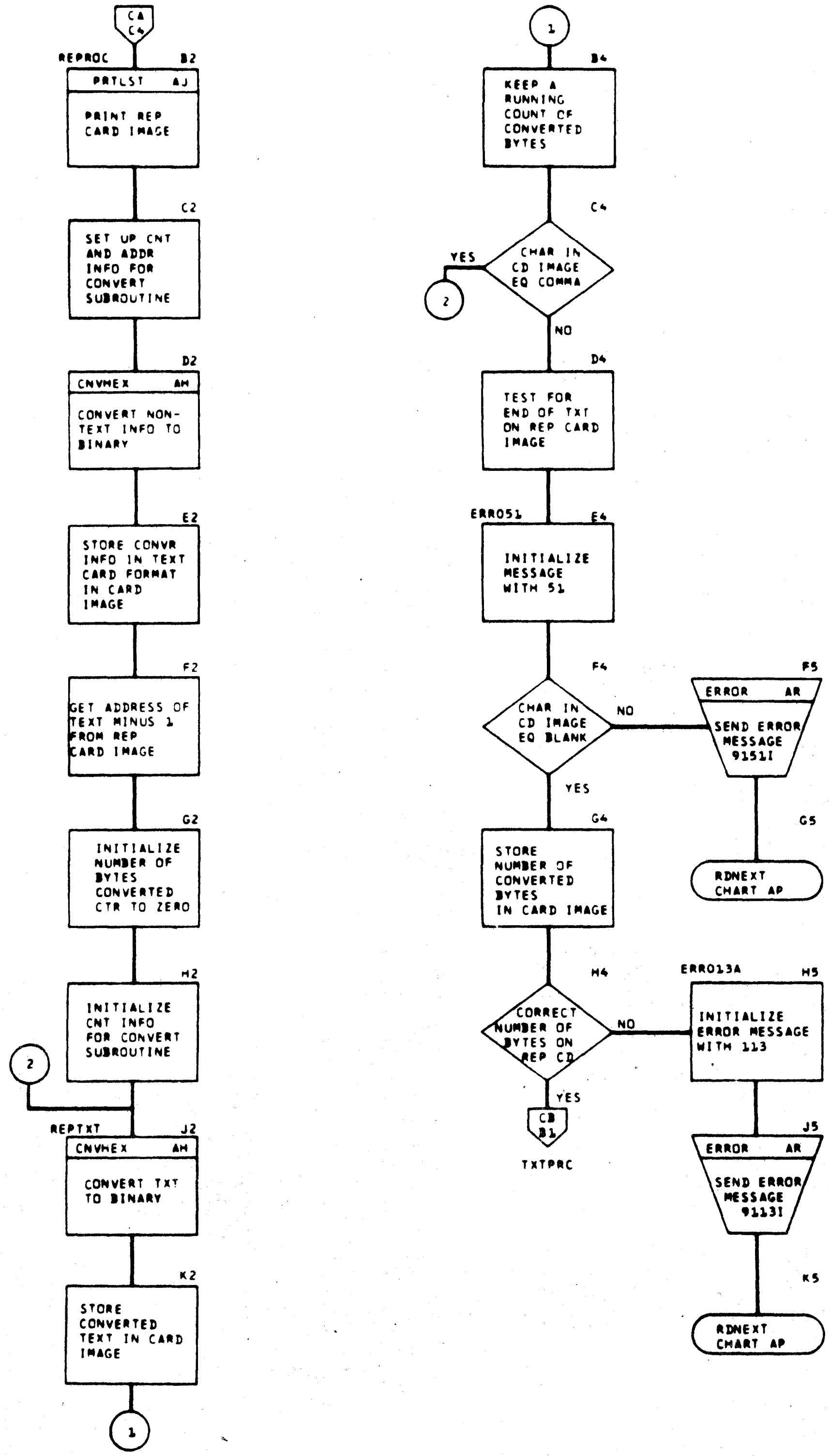


Chart CD. RLD Processor (ATLEDT12) (Refer to Chart 04) (Part 1 of 2)

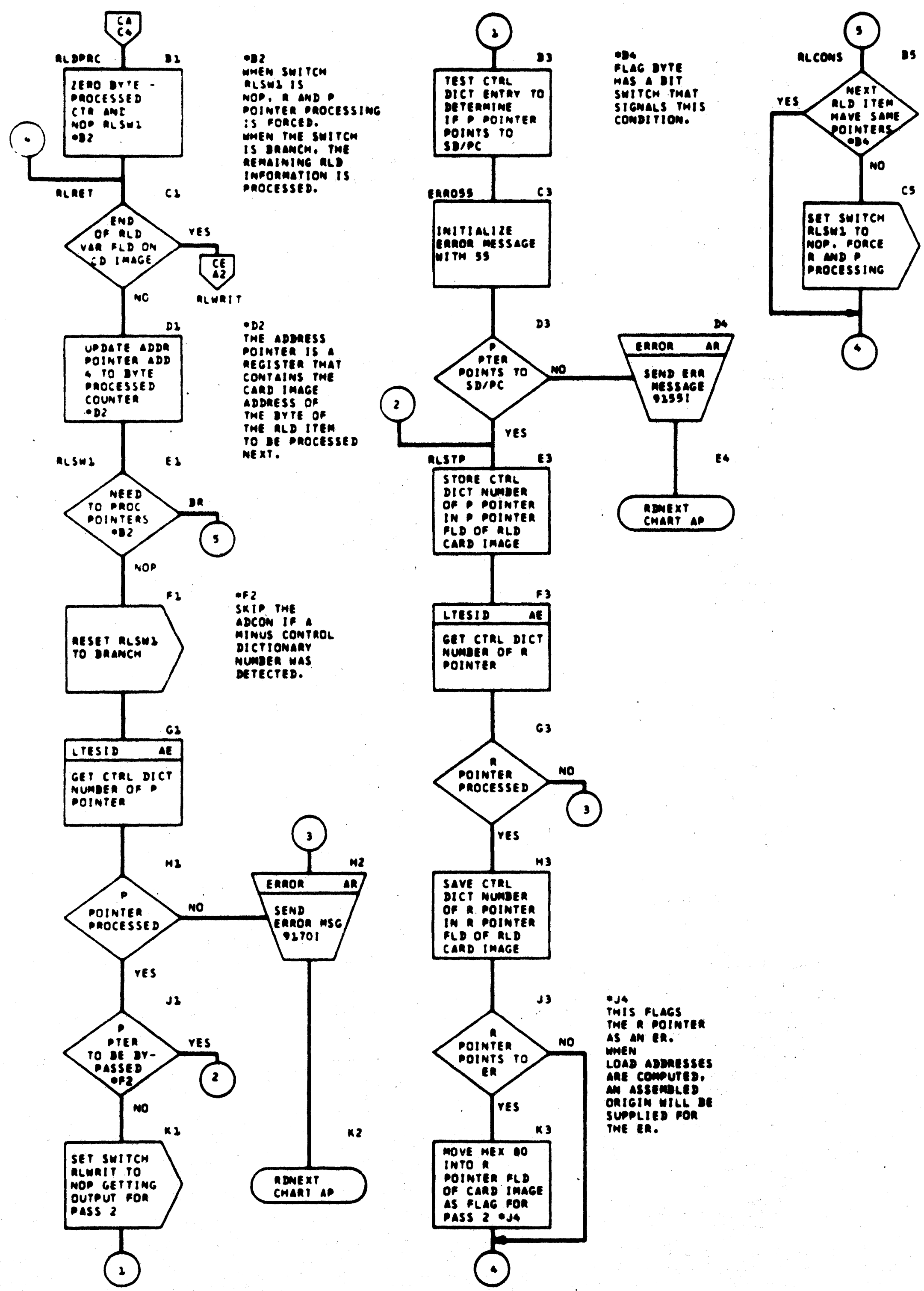
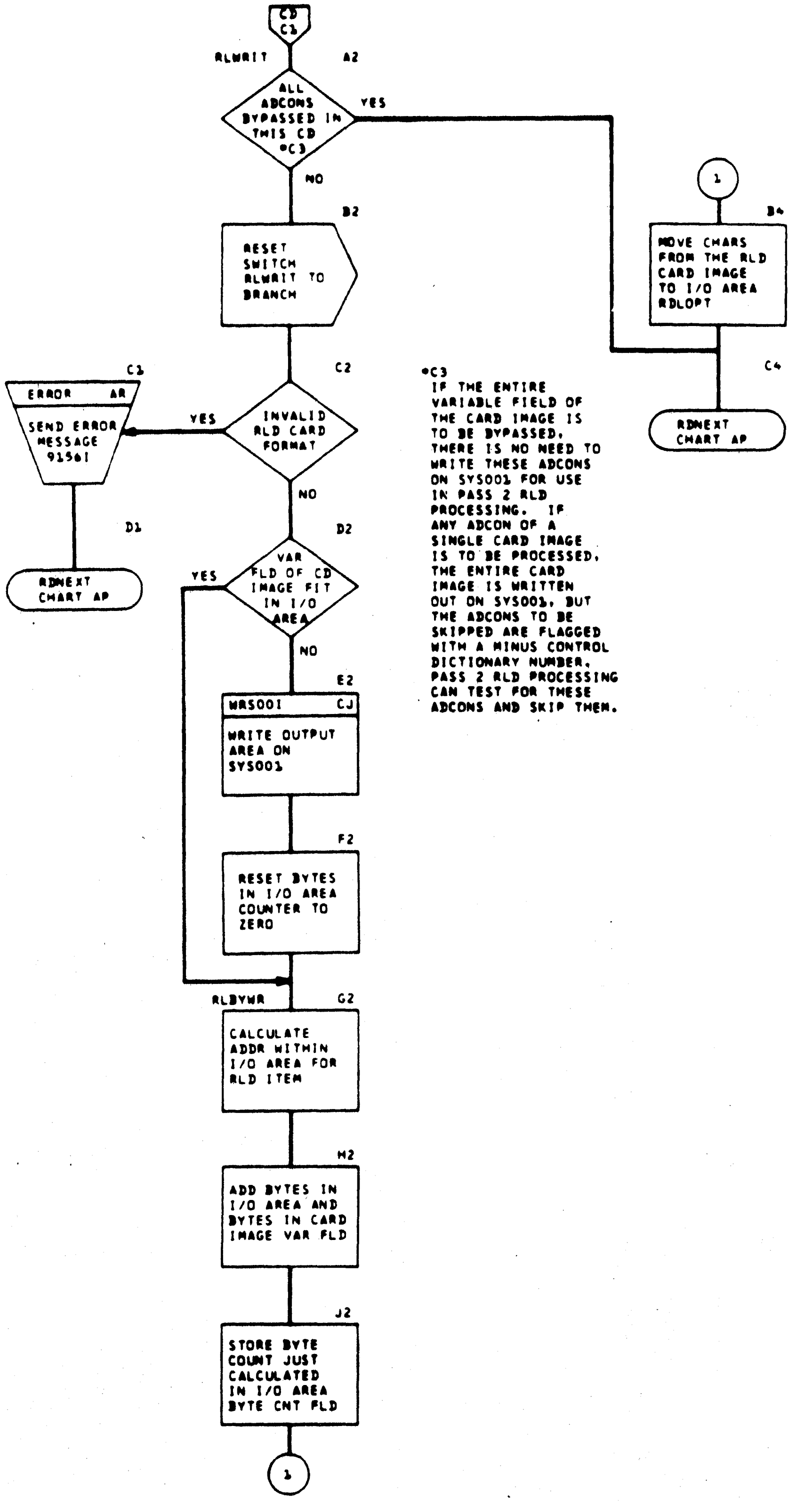
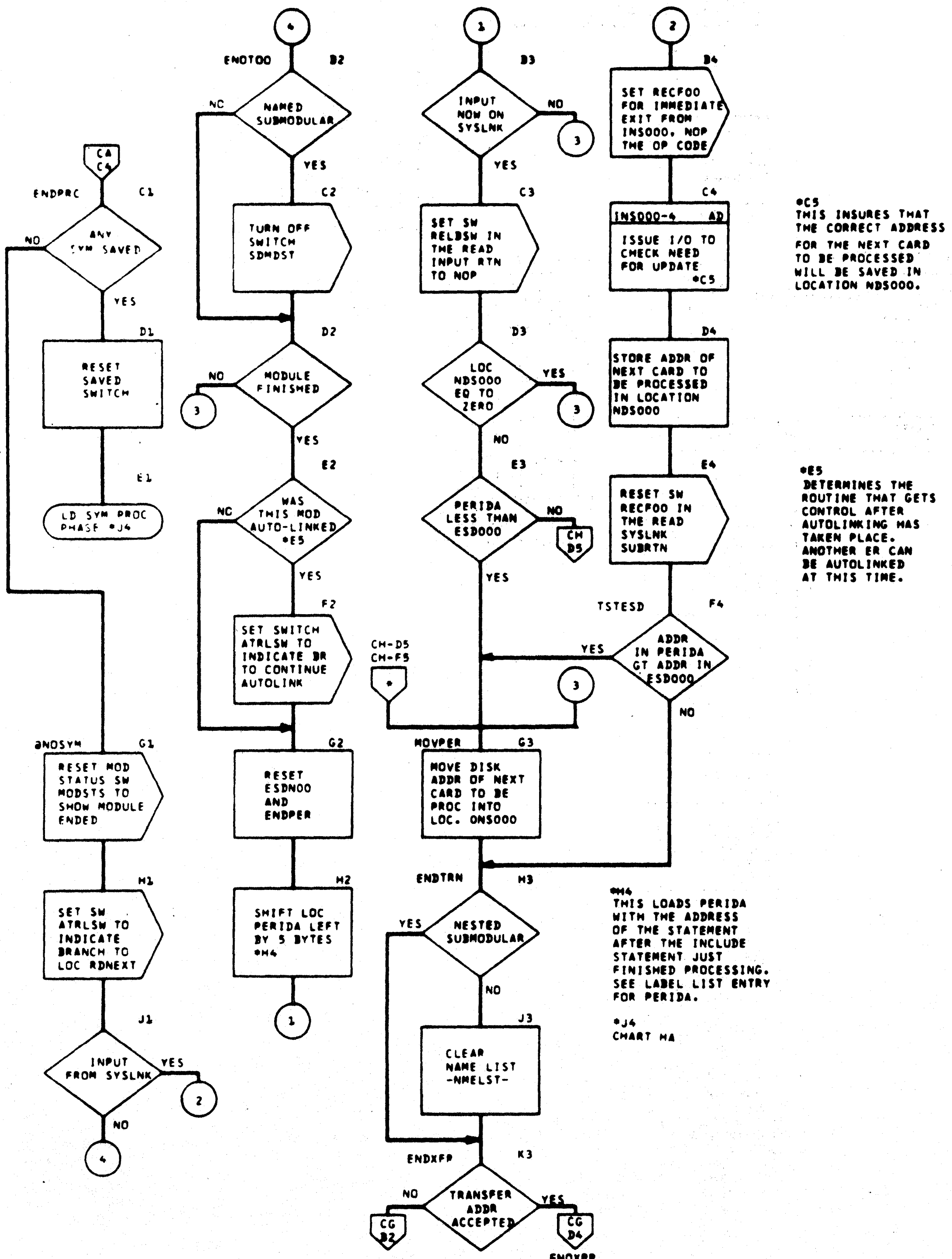
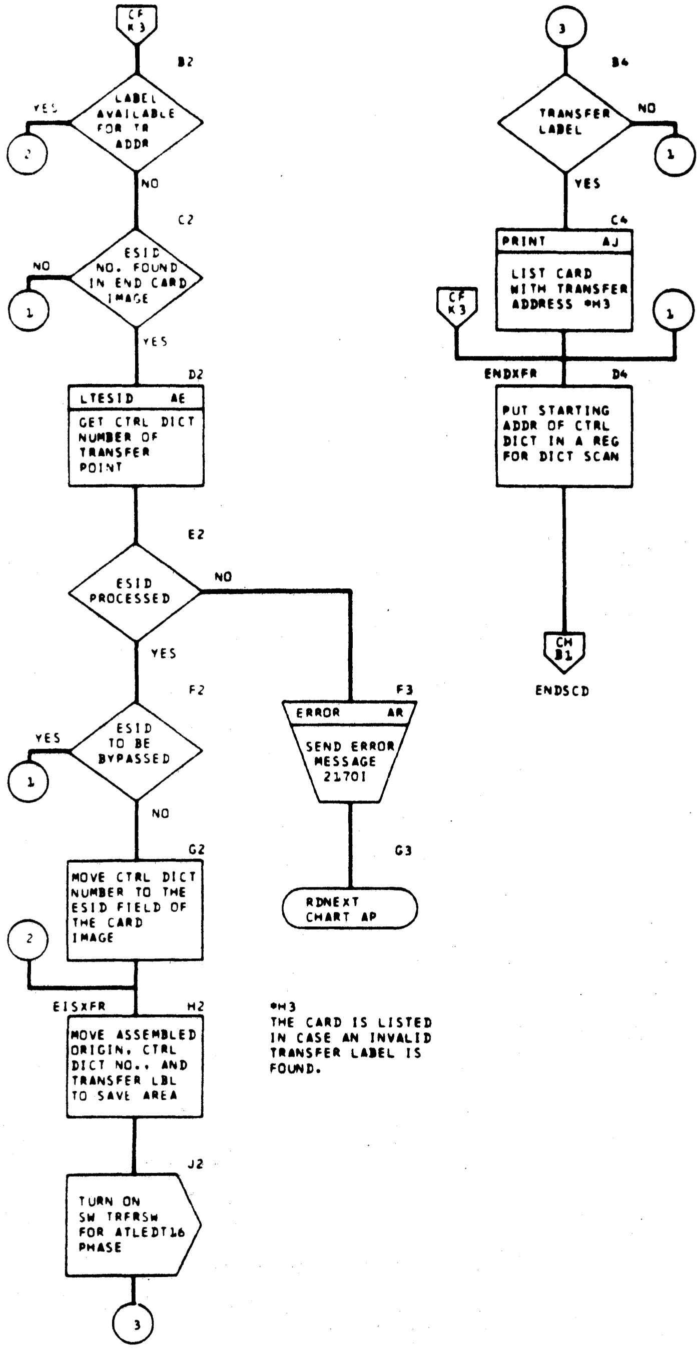


Chart CE. RLD Processor (ATLEDT12) (Refer to Chart 04) (Part 2 of 2)




ENDPRC C1
ANY SYM SAVED
RESET SAVED SWITCH
LD SYM PROC PHASE *J4
@NOSYM G1
RESET MOD STATUS SW MODSTS TO SHOW MODULE ENDED
SET SW ATRLSW TO INDICATE BRANCH TO LOC RDNEXT
INPUT FROM SYSLNK
ENDT00 B2
NAMED SUBMODULAR
TURN OFF SWITCH SDMDST
MODULE FINISHED
WAS THIS MOD AUTO-LINKED *E5
SET SWITCH ATRLSW TO INDICATE BR TO CONTINUE AUTOLINK
RESET ESDN00 AND ENDPER
SHIFT LOC PERIDA LEFT BY 5 BYTES *H4
INPUT NOW ON SYSLNK
SET SW RELBSW IN THE READ INPUT RTN TO NOP
LOC NDS000 EQ TO ZERO
PERIDA LESS THAN ESD000
CH-D5
CH-F5
MOVPER G3
MOVE DISK ADDR OF NEXT CARD TO BE PROC INTO LOC. ONS000
ENDTRN H3
NESTED SUBMODULAR
CLEAR NAME LIST -NMELST-
ENDXFR K3
TRANSFER ADDR ACCEPTED
ENDXPR
SET RECF00 FOR IMMEDIATE EXIT FROM INS000. NOP THE OP CODE
INS000-4 AD
ISSUE I/O TO CHECK NEED FOR UPDATE *C5
STORE ADDR OF NEXT CARD TO BE PROCESSED IN LOCATION NDS000
RESET SW RECF00 IN THE READ SYSLNK SUBRTN
TSTESD F4
ADDR IN PERIDA GT ADDR IN ESD000
*C5
THIS INSURES THAT THE CORRECT ADDRESS FOR THE NEXT CARD TO BE PROCESSED WILL BE SAVED IN LOCATION NDS000.
*E5
DETERMINES THE ROUTINE THAT GETS CONTROL AFTER AUTOLINKING HAS TAKEN PLACE. ANOTHER ER CAN BE AUTOLINKED AT THIS TIME.
*H4
THIS LOADS PERIDA WITH THE ADDRESS OF THE STATEMENT AFTER THE INCLUDE STATEMENT JUST FINISHED PROCESSING. SEE LABEL LIST ENTRY FOR PERIDA.
*J4
CHART HA


CF K3
B2
LABEL AVAILABLE FOR TR ADDR
YES
2
NO
C2
ESID NO. FOUND IN END CARD IMAGE
NO
1
YES
D2
LTESID AE
GET CTRL DICT NUMBER OF TRANSFER POINT
E2
ESID PROCESSED
NO
YES
F2
ESID TO BE BYPASSED
YES
1
NO
G2
MOVE CTRL DICT NUMBER TO THE ESID FIELD OF THE CARD IMAGE
2
EISXFR
H2
MOVE ASSEMBLED ORIGIN, CTRL DICT NO., AND TRANSFER LBL TO SAVE AREA
J2
TURN ON SW TRFRSW FOR ATLEDT16 PHASE
3
F3
ERROR AR
SEND ERROR MESSAGE 21701
G3
RDNEXT CHART AP
*H3
THE CARD IS LISTED IN CASE AN INVALID TRANSFER LABEL IS FOUND.
3
B4
TRANSFER LABEL
NO
1
YES
C4
PRINT AJ
LIST CARD WITH TRANSFER ADDRESS *H3
CF K3
1
ENDXFR
D4
PUT STARTING ADDR OF CTRL DICT IN A REG FOR DICT SCAN
CH B1
ENDSCD

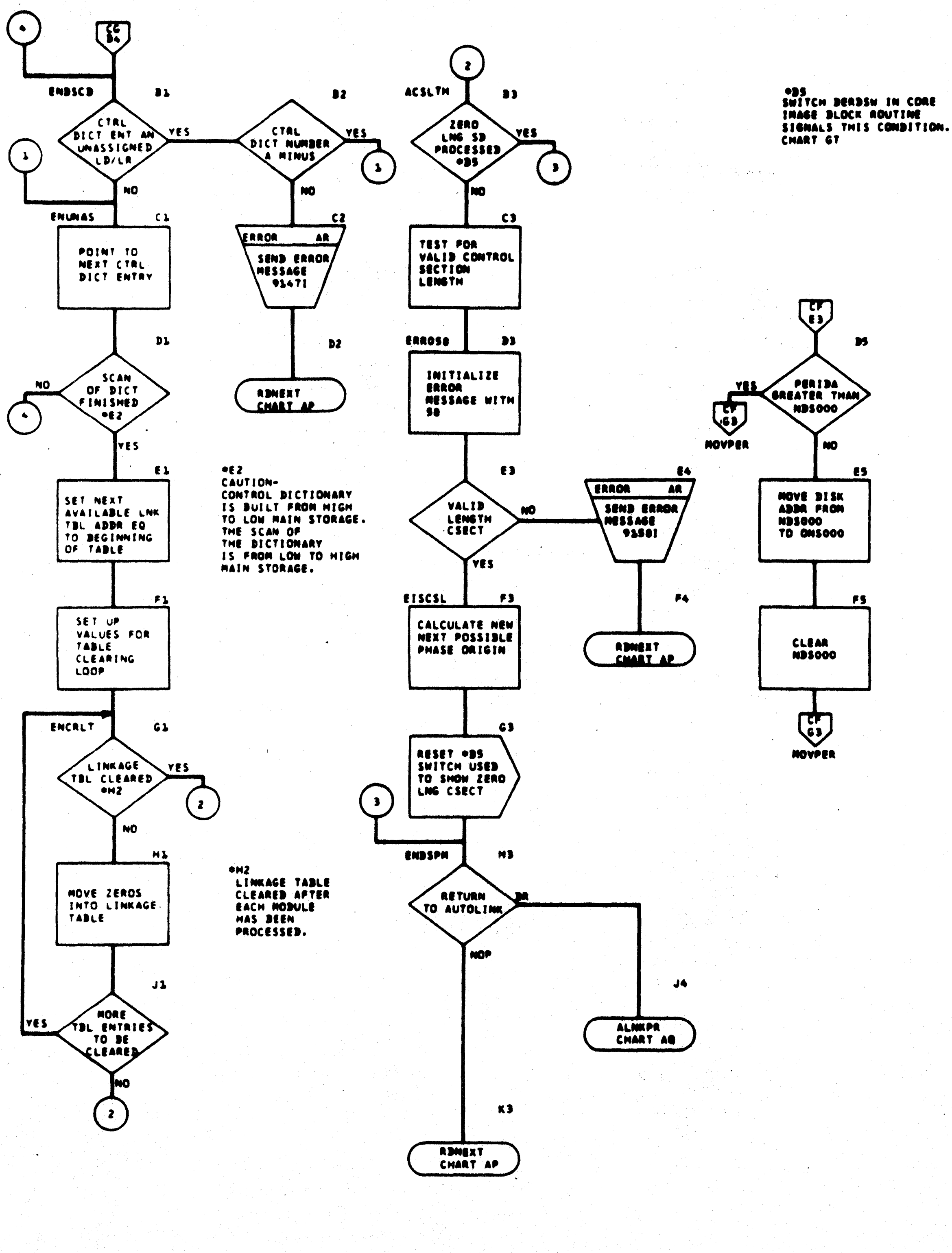

CG B4
ENDSCD B1
CTRL DICT ENT AN UNASSIGNED LD/LR
YES
NO
B2
CTRL DICT NUMBER A MINUS
YES
NO
ACSLTH B3
ZERO LNG SD PROCESSED *D5
YES
NO
*D5 SWITCH DERDSW IN CORE IMAGE BLOCK ROUTINE SIGNALS THIS CONDITION. CHART G7
ENUNAS C1
POINT TO NEXT CTRL DICT ENTRY
C2
ERROR AR
SEND ERROR MESSAGE 9147I
C3
TEST FOR VALID CONTROL SECTION LENGTH
D1
SCAN OF DICT FINISHED *E2
NO
YES
D2
RDNEXT CHART AP
ERROS8 D3
INITIALIZE ERROR MESSAGE WITH 58
CF E3
D5
PERIDA GREATER THAN NDS000
YES
CF G3
MOVPER
NO
E1
SET NEXT AVAILABLE LNK TBL ADDR EQ TO BEGINNING OF TABLE
*E2 CAUTION- CONTROL DICTIONARY IS BUILT FROM HIGH TO LOW MAIN STORAGE. THE SCAN OF THE DICTIONARY IS FROM LOW TO HIGH MAIN STORAGE.
E3
VALID LENGTH CSECT
NO
YES
E4
ERROR AR
SEND ERROR MESSAGE 9158I
E5
MOVE DISK ADDR FROM NDS000 TO ONS000
F1
SET UP VALUES FOR TABLE CLEARING LOOP
EISCSL F3
CALCULATE NEW NEXT POSSIBLE PHASE ORIGIN
F4
RDNEXT CHART AP
F5
CLEAR NDS000
CF G3
MOVPER
ENCRLT G1
LINKAGE TBL CLEARED *H2
YES
NO
G3
RESET *D5 SWITCH USED TO SHOW ZERO LNG CSECT
H1
MOVE ZEROS INTO LINKAGE TABLE
*H2 LINKAGE TABLE CLEARED AFTER EACH MODULE HAS BEEN PROCESSED.
ENDSPH H3
RETURN TO AUTOLINK
BR
NOP
J1
MORE TBL ENTRIES TO BE CLEARED
YES
NO
J4
ALNKPR CHART AQ
K3
RDNEXT CHART AP

Chart CJ. Write SYS001 Subroutine (ATLEDT12) (Refer to Chart 04)

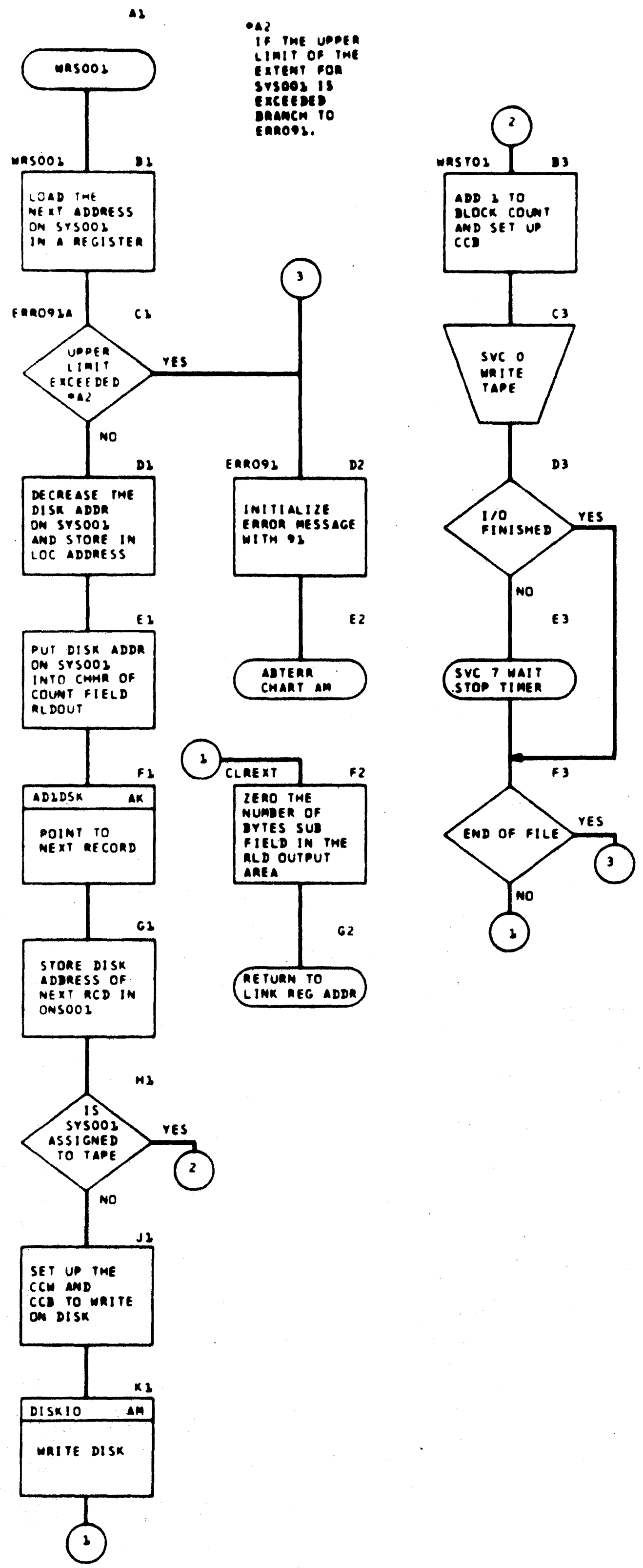
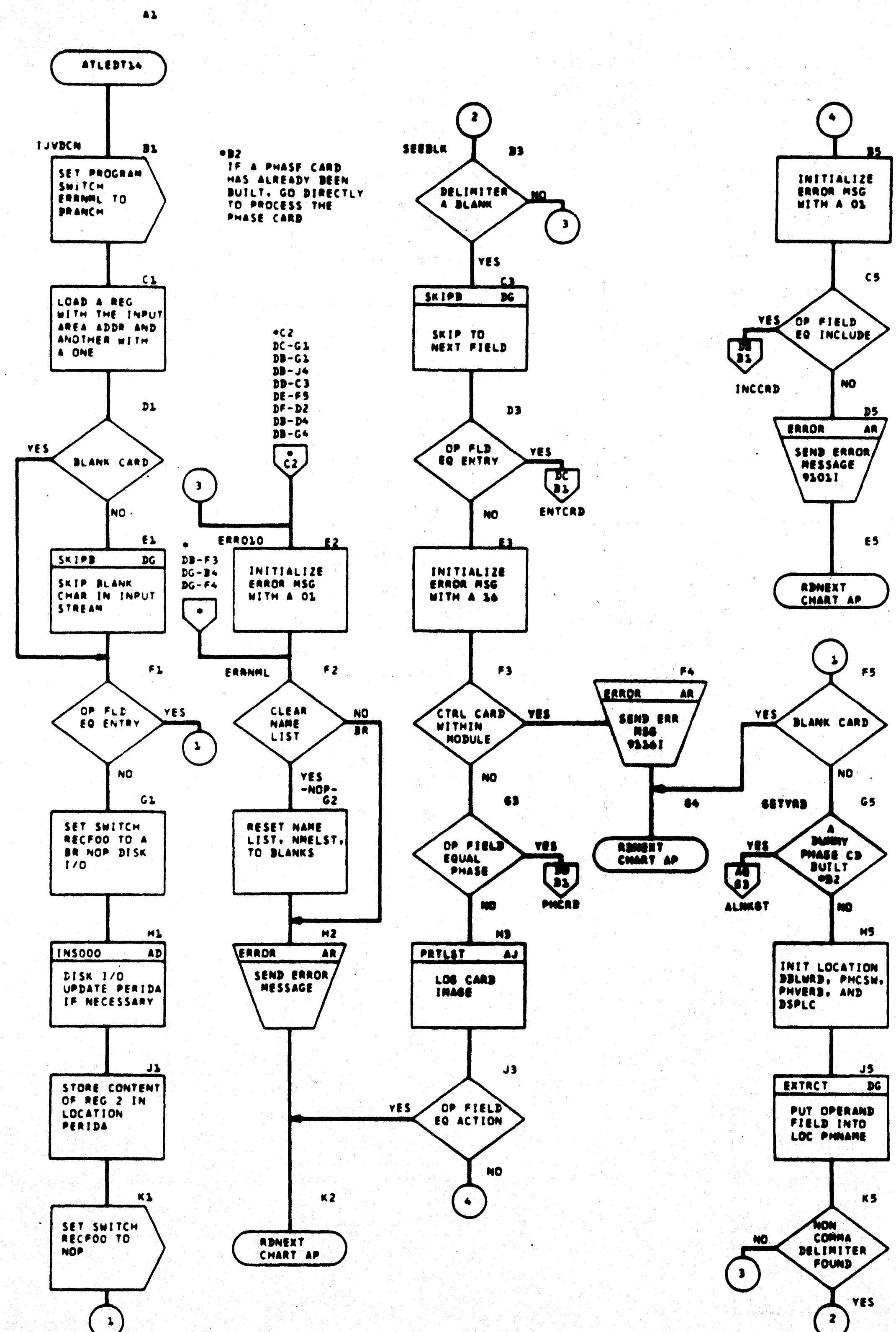
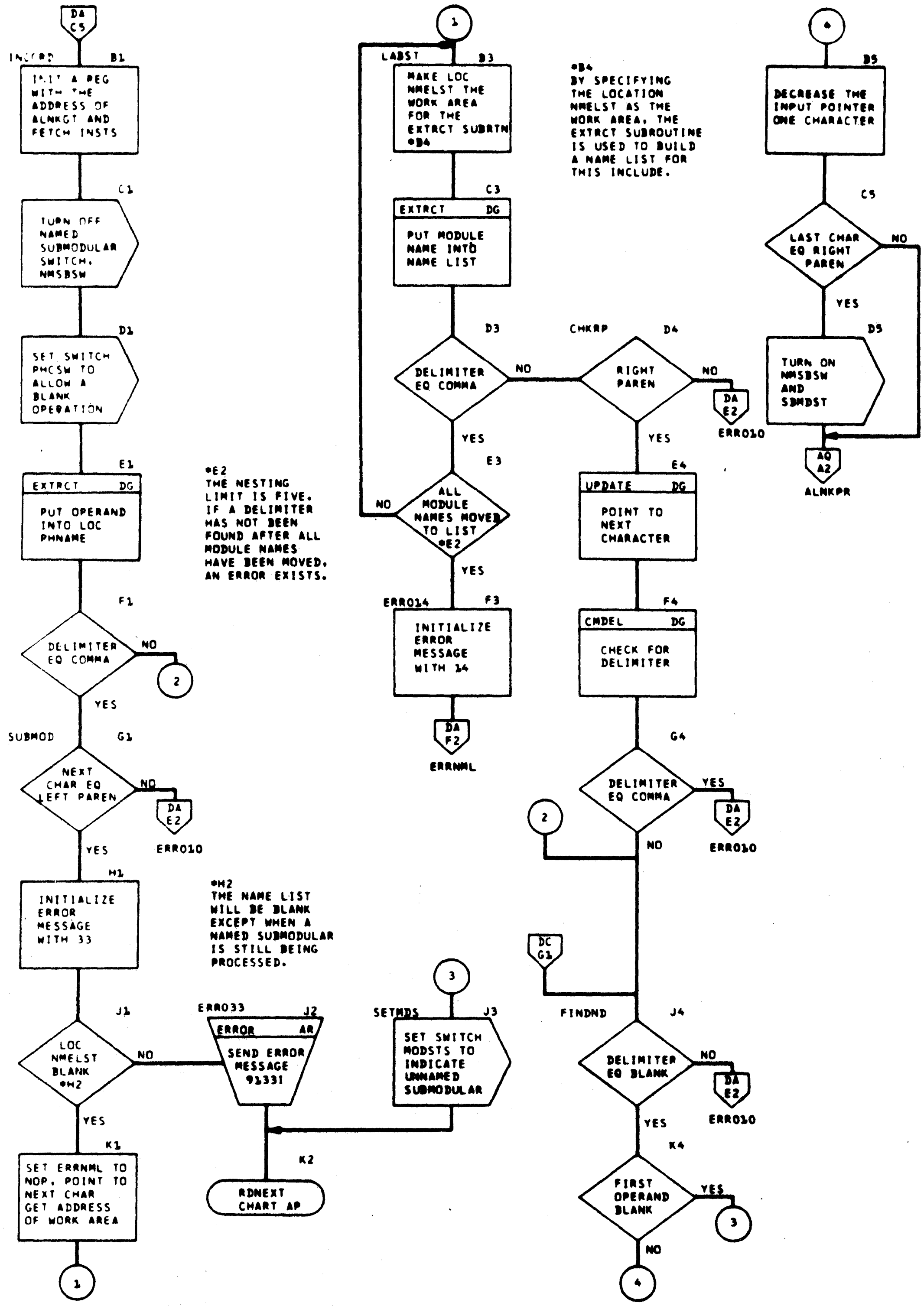
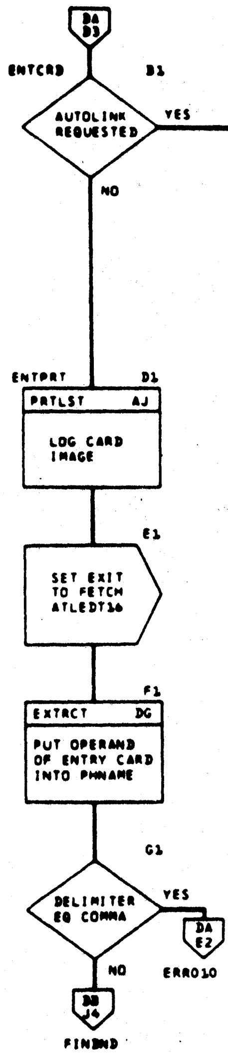
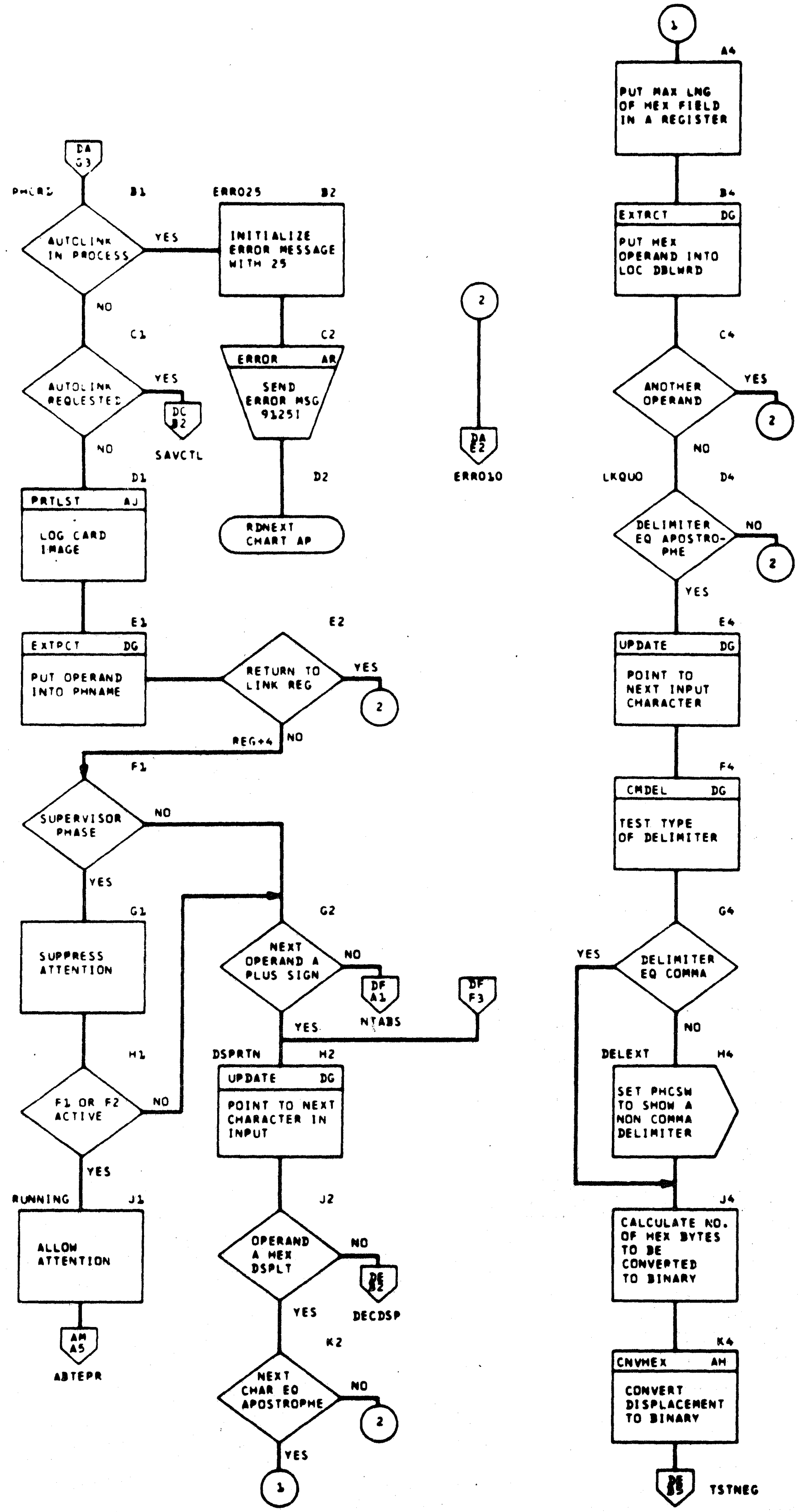


Chart DA. Initialize Control Card Processor (ATLEDT14) (Refer to Chart 05)




DA C5
INCCRD B1
INIT A REG WITH THE ADDRESS OF ALNKGT AND FETCH INSTS
C1
TURN OFF NAMED SUBMODULAR SWITCH, NMSBSW
D1
SET SWITCH PHCSW TO ALLOW A BLANK OPERATION
E1
EXTRCT DG
PUT OPERAND INTO LOC PHNAME
F1
DELIMITER EQ COMMA
NO
YES
SUBMOD G1
NEXT CHAR EQ LEFT PAREN
NO
DA E2
ERR010
YES
H1
INITIALIZE ERROR MESSAGE WITH 33
J1
LOC NMELST BLANK *H2
NO
YES
K1
SET ERRNML TO NOP, POINT TO NEXT CHAR GET ADDRESS OF WORK AREA
*E2 THE NESTING LIMIT IS FIVE. IF A DELIMITER HAS NOT BEEN FOUND AFTER ALL MODULE NAMES HAVE BEEN MOVED, AN ERROR EXISTS.
*H2 THE NAME LIST WILL BE BLANK EXCEPT WHEN A NAMED SUBMODULAR IS STILL BEING PROCESSED.
ERR033 J2
ERROR AR
SEND ERROR MESSAGE 91331
K2
RDNEXT CHART AP
LABST B3
MAKE LOC NMELST THE WORK AREA FOR THE EXTRCT SUBRTN *B4
C3
EXTRCT DG
PUT MODULE NAME INTO NAME LIST
D3
DELIMITER EQ COMMA
NO
YES
E3
ALL MODULE NAMES MOVED TO LIST *E2
NO
YES
ERR014 F3
INITIALIZE ERROR MESSAGE WITH 14
DA F2
ERRNML
SETMDS J3
SET SWITCH MODSTS TO INDICATE UNNAMED SUBMODULAR
*B4 BY SPECIFYING THE LOCATION NMELST AS THE WORK AREA, THE EXTRCT SUBROUTINE IS USED TO BUILD A NAME LIST FOR THIS INCLUDE.
CHKRP D4
RIGHT PAREN
NO
DA E2
ERR010
YES
E4
UPDATE DG
POINT TO NEXT CHARACTER
F4
CMDEL DG
CHECK FOR DELIMITER
G4
DELIMITER EQ COMMA
YES
DA E2
ERR010
NO
DC G1
FINDND J4
DELIMITER EQ BLANK
NO
DA E2
ERR010
YES
K4
FIRST OPERAND BLANK
YES
NO
B5
DECREASE THE INPUT POINTER ONE CHARACTER
C5
LAST CHAR EQ RIGHT PAREN
NO
YES
D5
TURN ON NMSBSW AND SBMDST
AQ A2
ALNKPR


DA
B3
ENTCRD
B1
AUTOLINK
REQUESTED
YES
NO
DB
C1
SAVCTL
B2
SAVE DISK
ADDRESS OF
THIS ENTRY
CARD
C2
ALNKPR
CHART AQ
ENTPRT
D1
PRTLST AJ
LOG CARD
IMAGE
E1
SET EXIT
TO FETCH
ATLEDT16
F1
EXTRCT DG
PUT OPERAND
OF ENTRY CARD
INTO PHNAME
G1
DELIMITER
EQ COMMA
YES
DA
E2
ERR010
NO
DB
J4
FINBND



PHCRD
AUTOLINK IN PROCESS
ERR025
INITIALIZE ERROR MESSAGE WITH 25
ERROR AR
SEND ERROR MSG 9125I
RDNEXT CHART AP
AUTOLINK REQUESTED
SAVCTL
PRTLST AJ
LOG CARD IMAGE
EXTPCT DG
PUT OPERAND INTO PHNAME
RETURN TO LINK REG
REG+4
SUPERVISOR PHASE
SUPPRESS ATTENTION
F1 OR F2 ACTIVE
RUNNING
ALLOW ATTENTION
ABTEPR
NEXT OPERAND A PLUS SIGN
NTABS
DSPRTN
UPDATE DG
POINT TO NEXT CHARACTER IN INPUT
OPERAND A HEX DSPLT
DECDSP
NEXT CHAR EQ APOSTROPHE
ERR010
PUT MAX LNG OF HEX FIELD IN A REGISTER
EXTRCT DG
PUT HEX OPERAND INTO LOC DBLWRD
ANOTHER OPERAND
LKQUO
DELIMITER EQ APOSTRO-PHE
UPDATE DG
POINT TO NEXT INPUT CHARACTER
CMDEL DG
TEST TYPE OF DELIMITER
DELIMITER EQ COMMA
DELEXT
SET PHCSW TO SHOW A NON COMMA DELIMITER
CALCULATE NO. OF HEX BYTES TO BE CONVERTED TO BINARY
CNVHEX AH
CONVERT DISPLACEMENT TO BINARY
TSTNEG

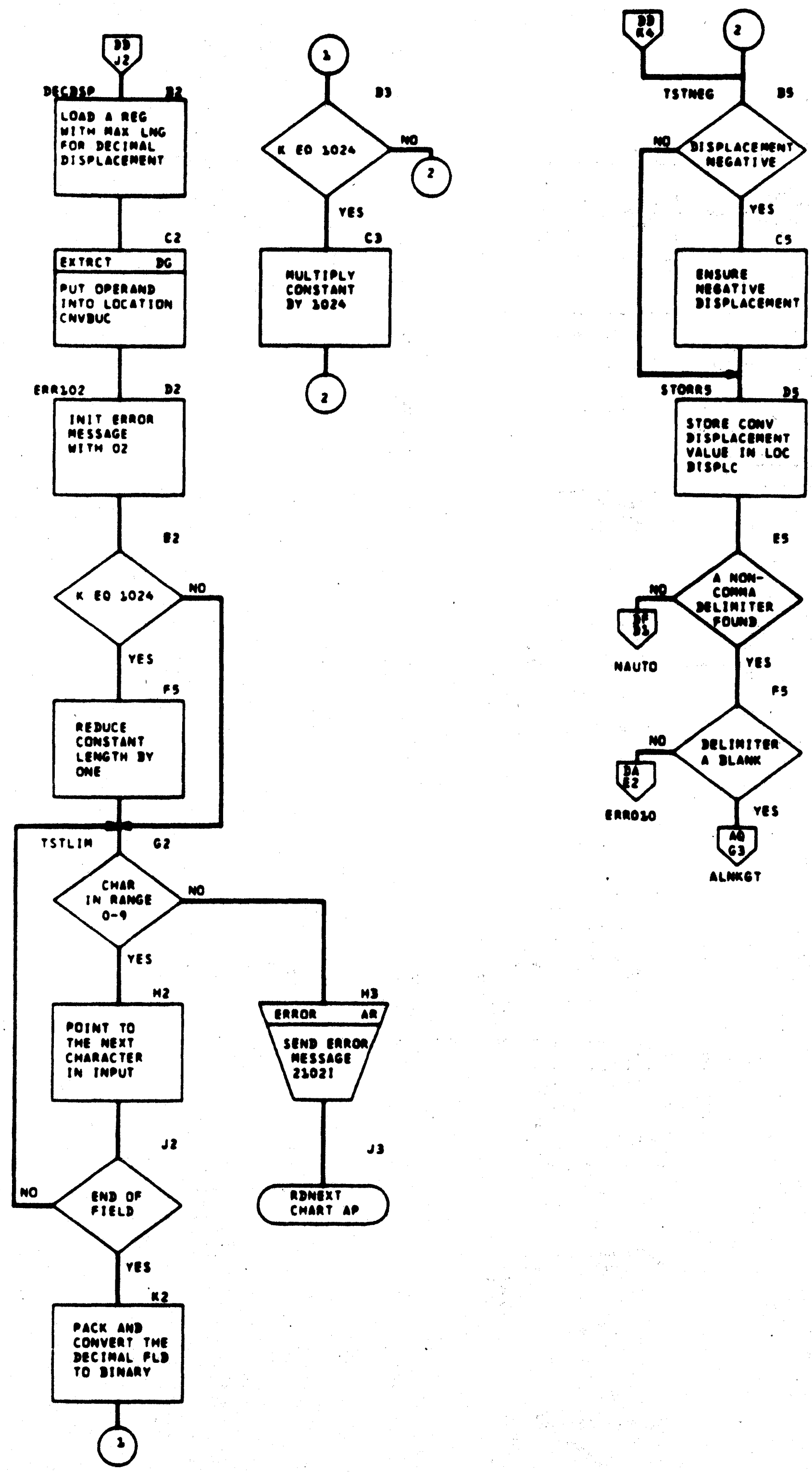
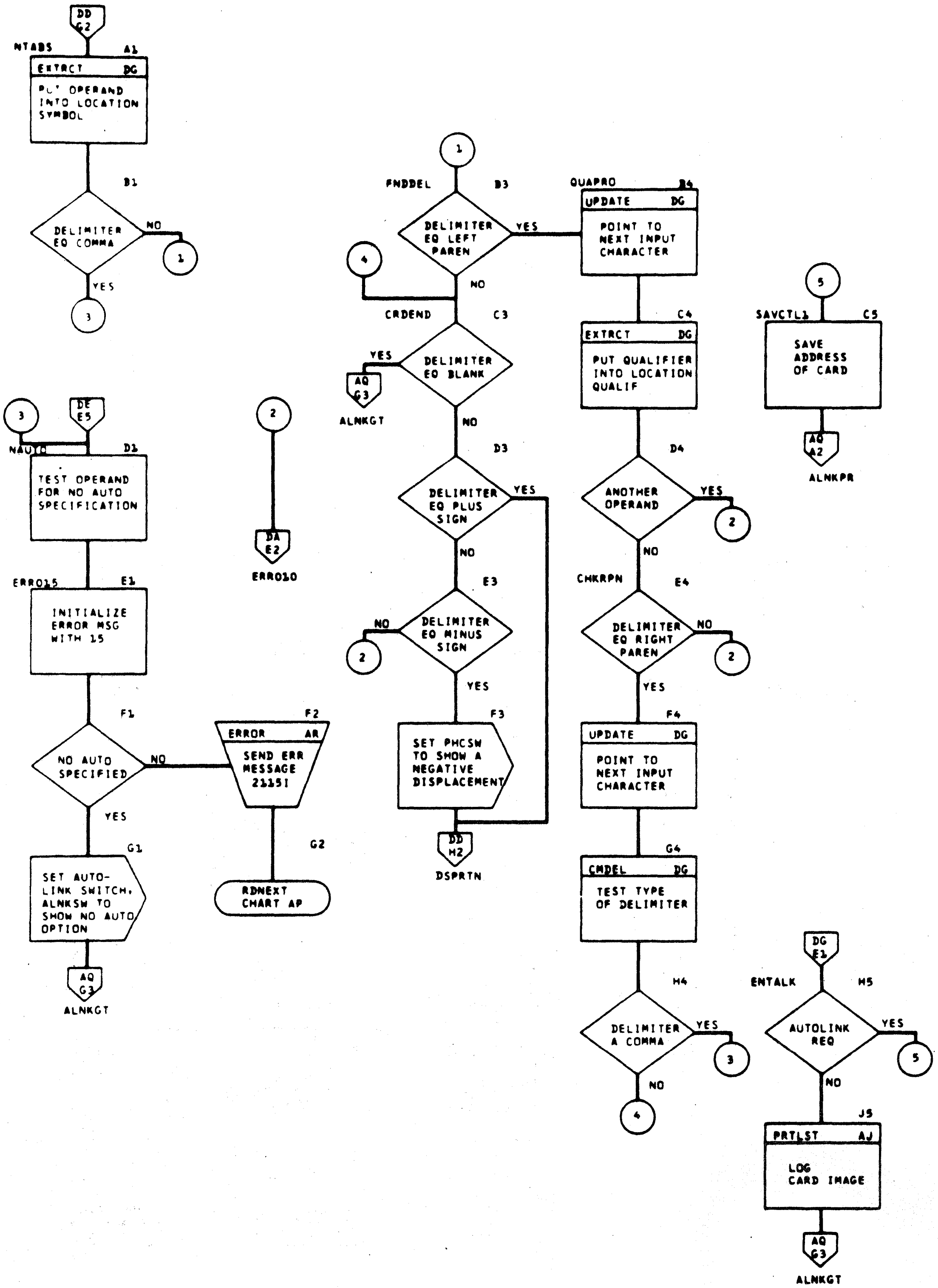
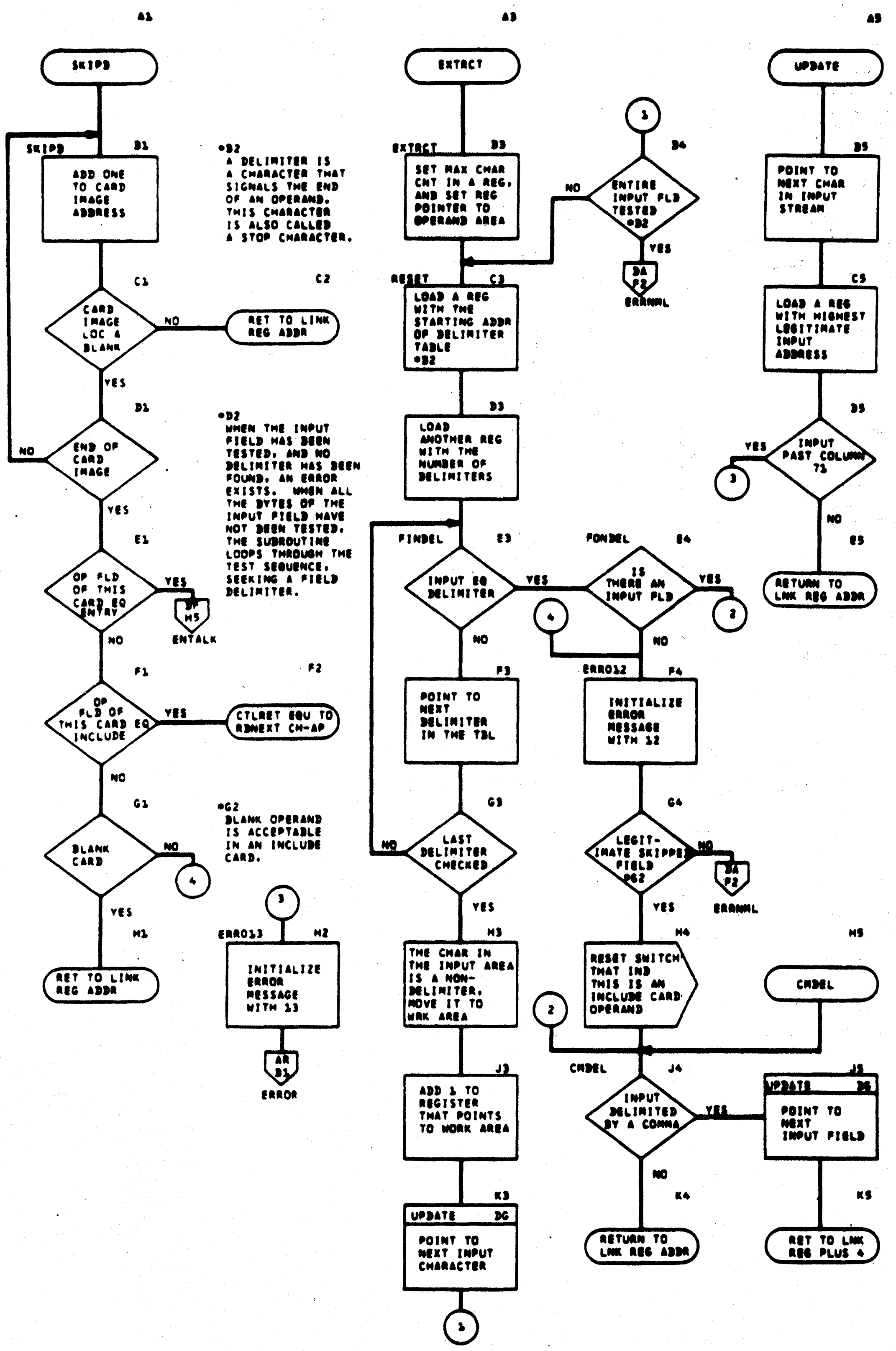
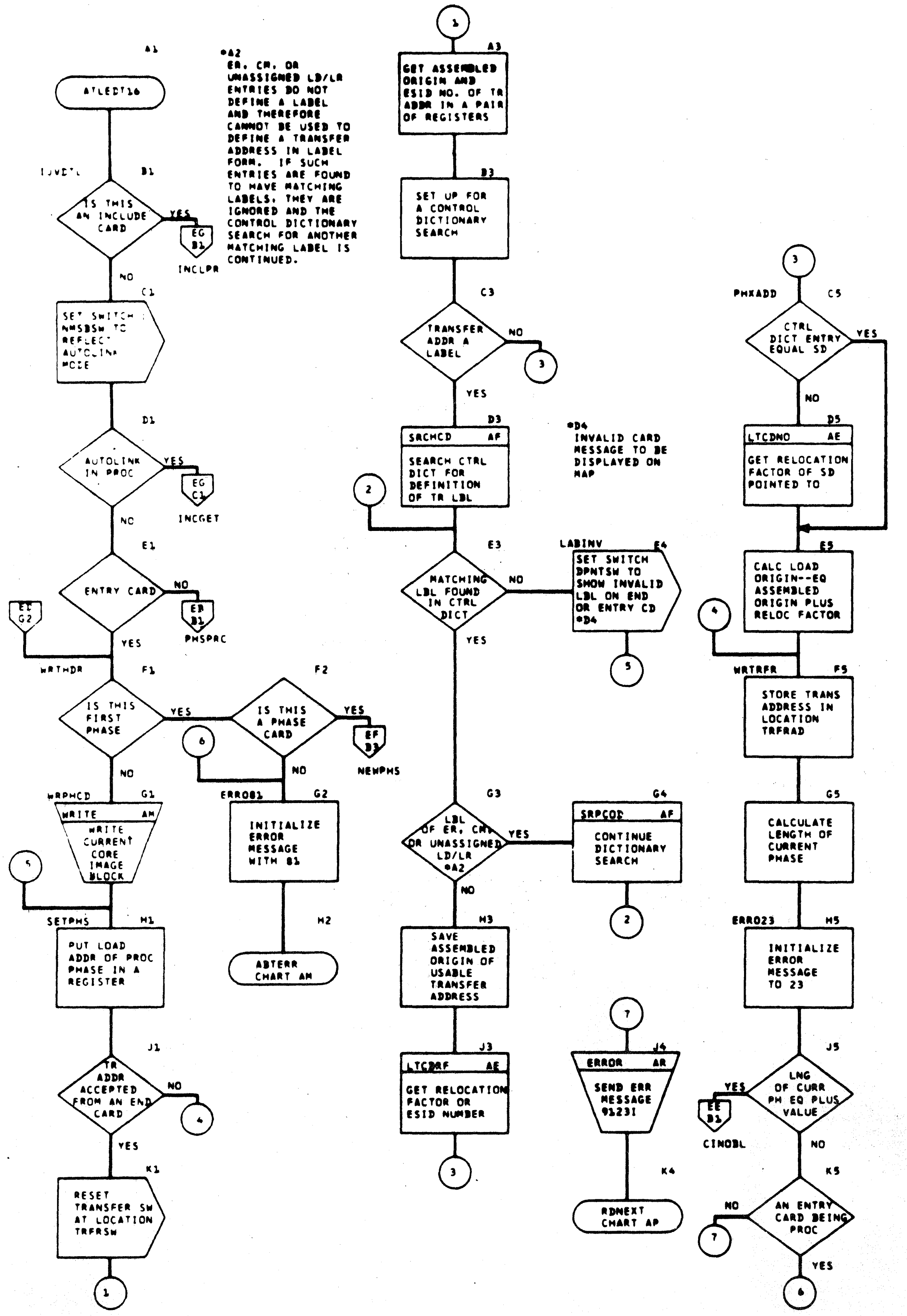
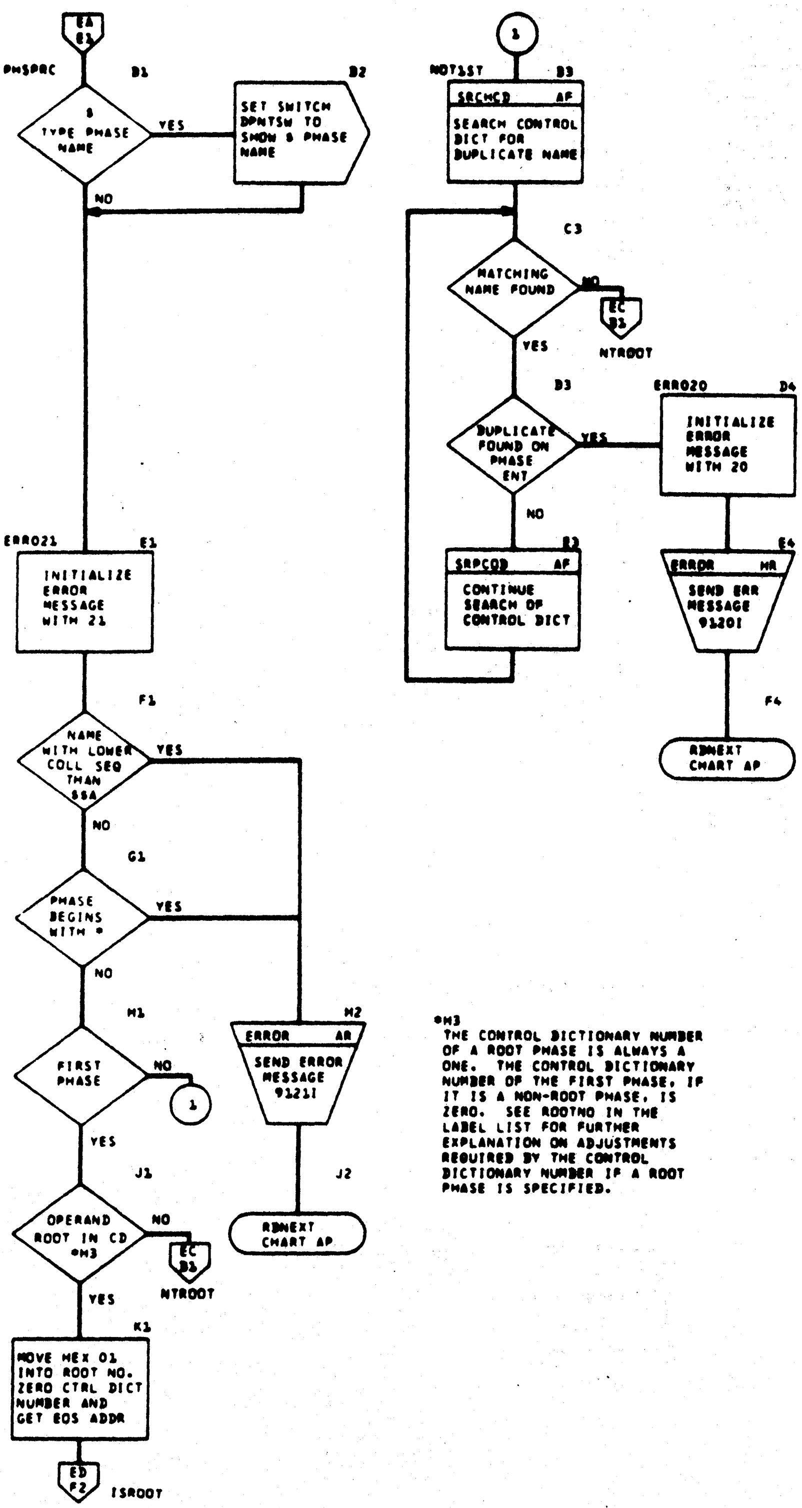

DD J2
DECDSP B2
LOAD A REG WITH MAX LNG FOR DECIMAL DISPLACEMENT
C2
EXTRCT DG
PUT OPERAND INTO LOCATION CNVBUC
ERR102 D2
INIT ERROR MESSAGE WITH 02
E2
K EQ 1024
NO
YES
F5
REDUCE CONSTANT LENGTH BY ONE
TSTLIM G2
CHAR IN RANGE 0-9
NO
YES
H2
POINT TO THE NEXT CHARACTER IN INPUT
J2
END OF FIELD
NO
YES
K2
PACK AND CONVERT THE DECIMAL FLD TO BINARY
1
1
B3
K EQ 1024
NO
2
YES
C3
MULTIPLY CONSTANT BY 1024
2
H3
ERROR AR
SEND ERROR MESSAGE 2102I
J3
RDNEXT CHART AP
DD K4
2
TSTNEG B5
DISPLACEMENT NEGATIVE
NO
YES
C5
ENSURE NEGATIVE DISPLACEMENT
STORR5 D5
STORE CONV DISPLACEMENT VALUE IN LOC DISPLC
E5
A NON-COMMA DELIMITER FOUND
NO
DE B5
NAUTO
YES
F5
DELIMITER A BLANK
NO
DA E2
ERR010
YES
AQ G3
ALNKGT

Chart DF. PHASE Card Processor (ATLEDT14) (Refer to Chart 05) (Part 3 of 3)

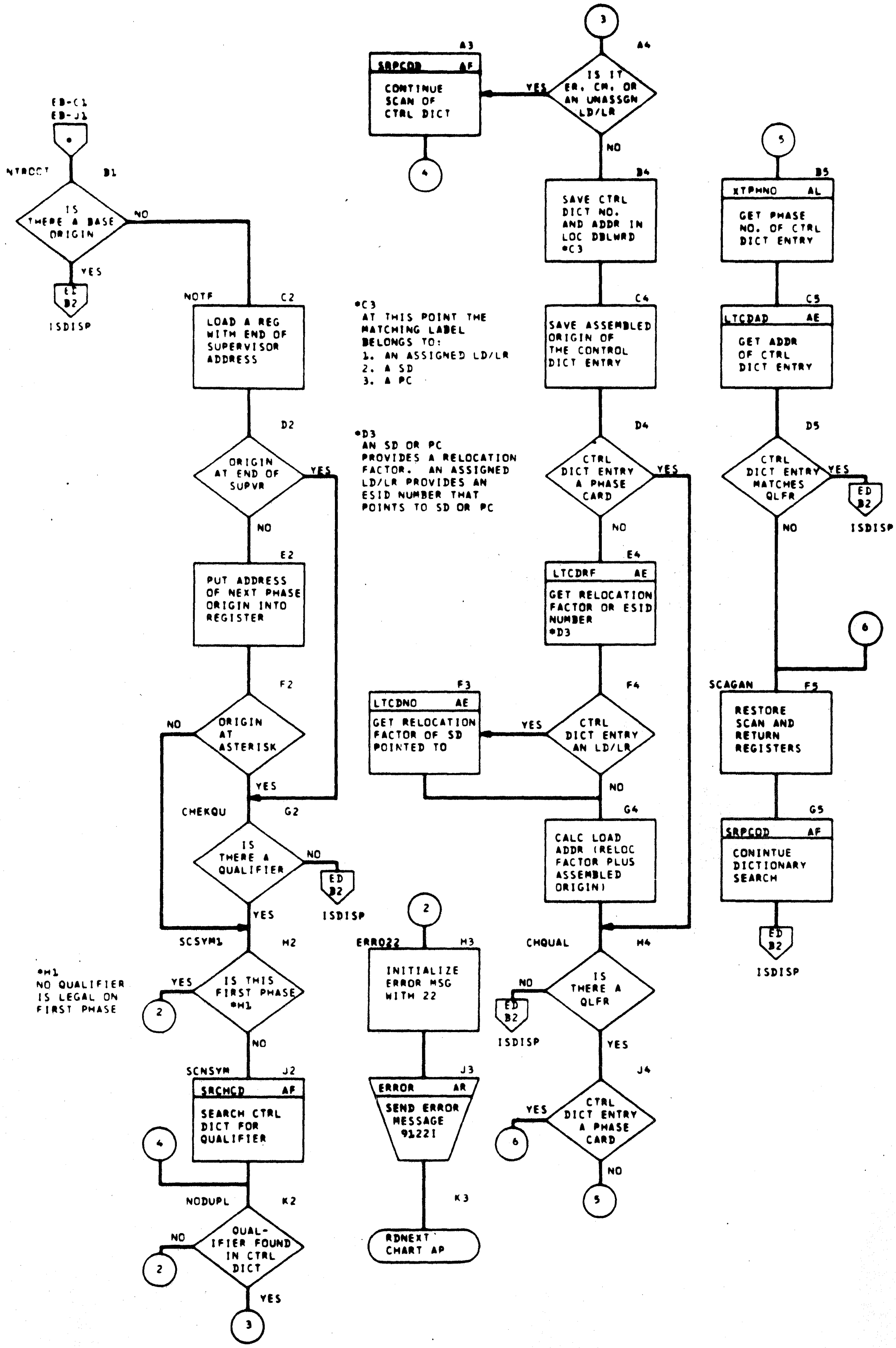



SKIPB
ADD ONE TO CARD IMAGE ADDRESS
*B2 A DELIMITER IS A CHARACTER THAT SIGNALS THE END OF AN OPERAND. THIS CHARACTER IS ALSO CALLED A STOP CHARACTER.
CARD IMAGE LOC A BLANK
RET TO LINK REG ADDR
END OF CARD IMAGE
*D2 WHEN THE INPUT FIELD HAS BEEN TESTED, AND NO DELIMITER HAS BEEN FOUND, AN ERROR EXISTS. WHEN ALL THE BYTES OF THE INPUT FIELD HAVE NOT BEEN TESTED, THE SUBROUTINE LOOPS THROUGH THE TEST SEQUENCE, SEEKING A FIELD DELIMITER.
OP FLD OF THIS CARD EQ ENTRY
ENTALK
OP FLD OF THIS CARD EQ INCLUDE
CTLRET EQU TO RDNEXT CM-AP
BLANK CARD
*G2 BLANK OPERAND IS ACCEPTABLE IN AN INCLUDE CARD.
ERR013
INITIALIZE ERROR MESSAGE WITH 13
ERROR
EXTRCT
SET MAX CHAR CNT IN A REG, AND SET REG POINTER TO OPERAND AREA
ENTIRE INPUT FLD TESTED *D2
ERRMML
RESET
LOAD A REG WITH THE STARTING ADDR OF DELIMITER TABLE *B2
LOAD ANOTHER REG WITH THE NUMBER OF DELIMITERS
FINDEL
INPUT EQ DELIMITER
FONDEL
IS THERE AN INPUT FLD
POINT TO NEXT DELIMITER IN THE TBL
ERR012
INITIALIZE ERROR MESSAGE WITH 12
LAST DELIMITER CHECKED
LEGIT-IMATE SKIPPED FIELD *G2
THE CHAR IN THE INPUT AREA IS A NON-DELIMITER, MOVE IT TO WRK AREA
RESET SWITCH THAT IND THIS IS AN INCLUDE CARD OPERAND
ADD 1 TO REGISTER THAT POINTS TO WORK AREA
CMDEL
INPUT DELIMITED BY A COMMA
UPDATE
POINT TO NEXT INPUT FIELD
POINT TO NEXT INPUT CHARACTER
RETURN TO LNK REG ADDR
RET TO LNK REG PLUS 4
POINT TO NEXT CHAR IN INPUT STREAM
LOAD A REG WITH HIGHEST LEGITIMATE INPUT ADDRESS
INPUT PAST COLUMN 71
YES
NO

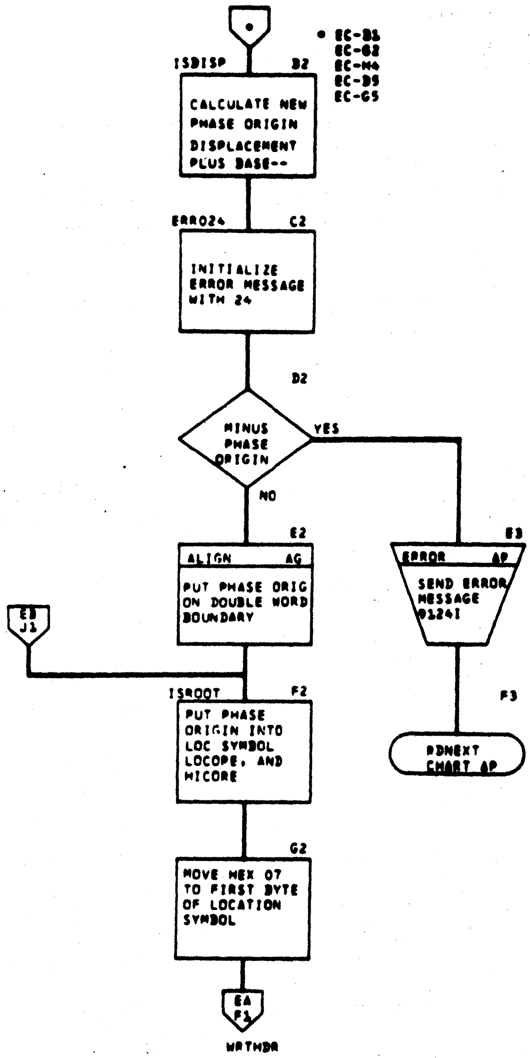

ATLEDT16
IS THIS AN INCLUDE CARD
YES EG B1 INCLPR
NO
SET SWITCH NMSDSW TO REFLECT AUTOLINK MODE
AUTOLINK IN PROC
YES EG C1 INCGET
ENTRY CARD
NO EB B1 PHSPRC
EC G2
WRTHDR
IS THIS FIRST PHASE
IS THIS A PHASE CARD
YES EF B3 NEWPHS
WRPHCD
WRITE AM
WRITE CURRENT CORE IMAGE BLOCK
ERR081
INITIALIZE ERROR MESSAGE WITH 81
ABTERR CHART AM
SETPHS
PUT LOAD ADDR OF PROC PHASE IN A REGISTER
TR ADDR ACCEPTED FROM AN END CARD
RESET TRANSFER SW AT LOCATION TRFRSW
*A2
ER, CM, OR UNASSIGNED LD/LR ENTRIES DO NOT DEFINE A LABEL AND THEREFORE CANNOT BE USED TO DEFINE A TRANSFER ADDRESS IN LABEL FORM. IF SUCH ENTRIES ARE FOUND TO HAVE MATCHING LABELS, THEY ARE IGNORED AND THE CONTROL DICTIONARY SEARCH FOR ANOTHER MATCHING LABEL IS CONTINUED.
GET ASSEMBLED ORIGIN AND ESID NO. OF TR ADDR IN A PAIR OF REGISTERS
SET UP FOR A CONTROL DICTIONARY SEARCH
TRANSFER ADDR A LABEL
SRCHCD AF
SEARCH CTRL DICT FOR DEFINITION OF TR LBL
*D4
INVALID CARD MESSAGE TO BE DISPLAYED ON MAP
MATCHING LBL FOUND IN CTRL DICT
LABINV
SET SWITCH DPNTSW TO SHOW INVALID LBL ON END OR ENTRY CD *D4
LBL OF ER, CM, OR UNASSIGNED LD/LR *A2
SRPCOD AF
CONTINUE DICTIONARY SEARCH
SAVE ASSEMBLED ORIGIN OF USABLE TRANSFER ADDRESS
LTCDRF AE
GET RELOCATION FACTOR OR ESID NUMBER
ERROR AR
SEND ERR MESSAGE 91231
RDNEXT CHART AP
PHXADD
CTRL DICT ENTRY EQUAL SD
LTCDNO AE
GET RELOCATION FACTOR OF SD POINTED TO
CALC LOAD ORIGIN--EQ ASSEMBLED ORIGIN PLUS RELOC FACTOR
WRTRFR
STORE TRANS ADDRESS IN LOCATION TRFRAD
CALCULATE LENGTH OF CURRENT PHASE
ERR023
INITIALIZE ERROR MESSAGE TO 23
LNG OF CURR PH EQ PLUS VALUE
EE B1 CINOBL
AN ENTRY CARD BEING PROC



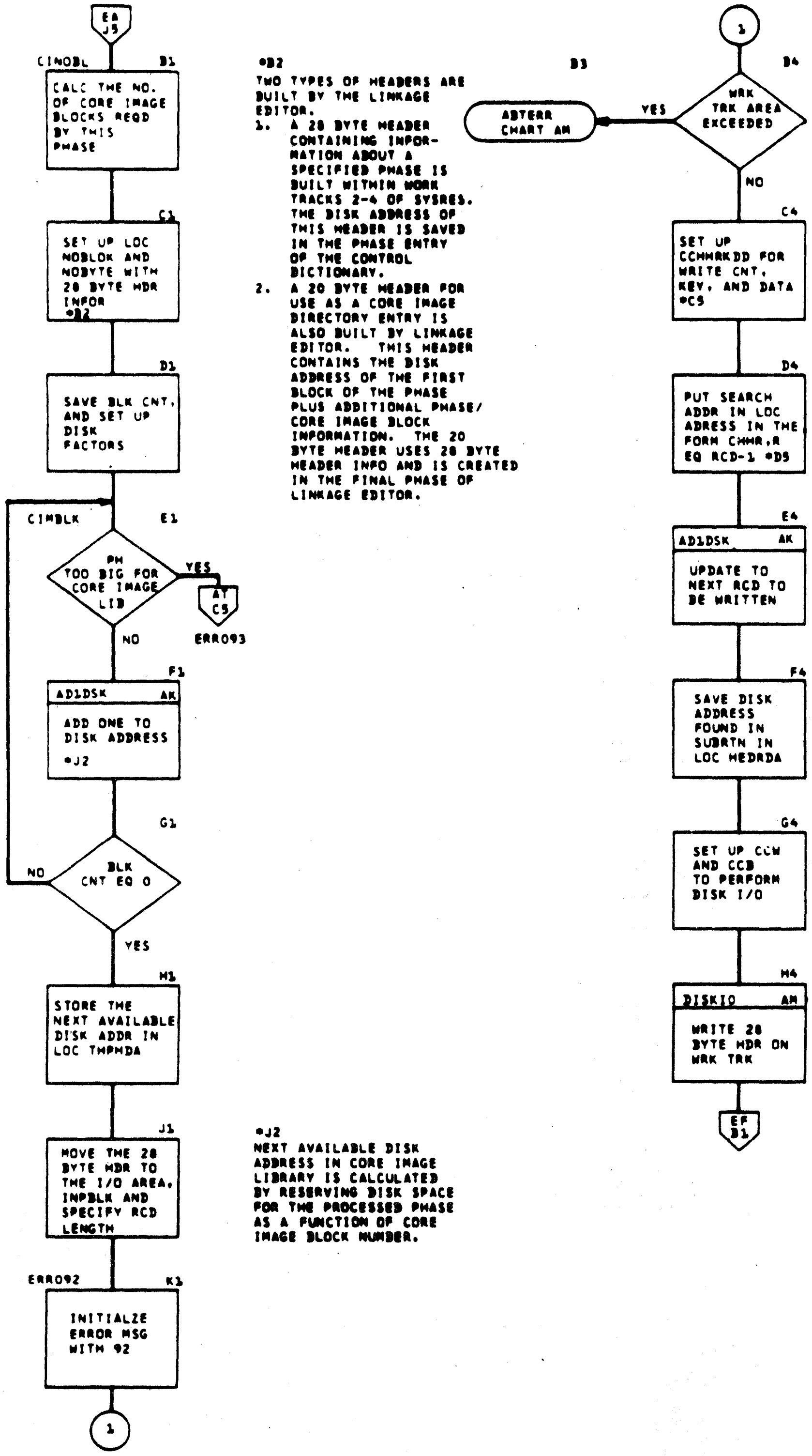
*H3
THE CONTROL DICTIONARY NUMBER OF A ROOT PHASE IS ALWAYS A ONE. THE CONTROL DICTIONARY NUMBER OF THE FIRST PHASE, IF IT IS A NON-ROOT PHASE, IS ZERO. SEE ROOTNO IN THE LABEL LIST FOR FURTHER EXPLANATION ON ADJUSTMENTS REQUIRED BY THE CONTROL DICTIONARY NUMBER IF A ROOT PHASE IS SPECIFIED.



EB-C1
EB-J1
NTRCCT
B1
IS THERE A BASE ORIGIN
NO
YES
EC B2
ISDISP
NOTF C2
LOAD A REG WITH END OF SUPERVISOR ADDRESS
D2
ORIGIN AT END OF SUPVR
YES
NO
E2
PUT ADDRESS OF NEXT PHASE ORIGIN INTO REGISTER
F2
ORIGIN AT ASTERISK
NO
YES
CHEKQU G2
IS THERE A QUALIFIER
NO
ED B2
ISDISP
YES
SCSYM1 H2
*H1
NO QUALIFIER IS LEGAL ON FIRST PHASE
IS THIS FIRST PHASE *H1
YES
NO
SCNSYM J2
SRCHCD AF
SEARCH CTRL DICT FOR QUALIFIER
NODUPL K2
QUAL-IFIER FOUND IN CTRL DICT
NO
YES
A3
SRPCOD AF
CONTINUE SCAN OF CTRL DICT
A4
IS IT ER, CM, OR AN UNASSGN LD/LR
YES
NO
B4
SAVE CTRL DICT NO. AND ADDR IN LOC DBLWRD *C3
*C3
AT THIS POINT THE MATCHING LABEL BELONGS TO:
1. AN ASSIGNED LD/LR
2. A SD
3. A PC
C4
SAVE ASSEMBLED ORIGIN OF THE CONTROL DICT ENTRY
*D3
AN SD OR PC PROVIDES A RELOCATION FACTOR. AN ASSIGNED LD/LR PROVIDES AN ESID NUMBER THAT POINTS TO SD OR PC
D4
CTRL DICT ENTRY A PHASE CARD
YES
NO
E4
LTCDRF AE
GET RELOCATION FACTOR OR ESID NUMBER *D3
F3
LTCDNO AE
GET RELOCATION FACTOR OF SD POINTED TO
F4
CTRL DICT ENTRY AN LD/LR
YES
NO
G4
CALC LOAD ADDR (RELOC FACTOR PLUS ASSEMBLED ORIGIN)
ERR022 H3
INITIALIZE ERROR MSG WITH 22
J3
ERROR AR
SEND ERROR MESSAGE 91221
K3
RDNEXT CHART AP
CHQUAL H4
IS THERE A QLFR
NO
ED B2
ISDISP
YES
J4
CTRL DICT ENTRY A PHASE CARD
YES
NO
B5
XTPHNO AL
GET PHASE NO. OF CTRL DICT ENTRY
C5
LTCDAD AE
GET ADDR OF CTRL DICT ENTRY
D5
CTRL DICT ENTRY MATCHES QLFR
YES
ED B2
ISDISP
NO
SCAGAN F5
RESTORE SCAN AND RETURN REGISTERS
G5
SRPCOD AF
CONINTUE DICTIONARY SEARCH
ED B2
ISDISP



• EC-B1
EC-G2
EC-H4
EC-D5
EC-G5
ISDISP B2
CALCULATE NEW PHASE ORIGIN DISPLACEMENT PLUS BASE--
ERR024 C2
INITIALIZE ERROR MESSAGE WITH 24
D2
MINUS PHASE ORIGIN
YES
NO
E2
ALIGN AG
PUT PHASE ORIG ON DOUBLE WORD BOUNDARY
E3
ERROR AP
SEND ERROR MESSAGE 0124I
EB J1
ISROOT F2
PUT PHASE ORIGIN INTO LOC SYMBOL LOCORE, AND HICORE
F3
RDNEXT CHART AP
G2
MOVE HEX 07 TO FIRST BYTE OF LOCATION SYMBOL
EA F1
WRTHDR

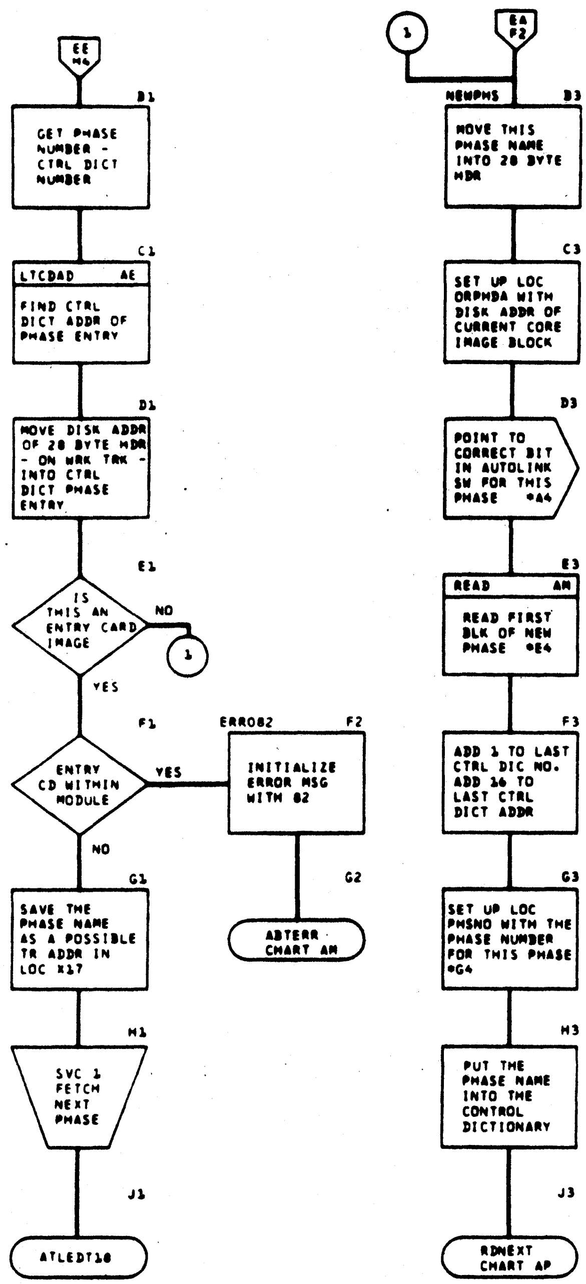


*B2
TWO TYPES OF HEADERS ARE BUILT BY THE LINKAGE EDITOR.
1. A 28 BYTE HEADER CONTAINING INFORMATION ABOUT A SPECIFIED PHASE IS BUILT WITHIN WORK TRACKS 2-4 OF SYSRES. THE DISK ADDRESS OF THIS HEADER IS SAVED IN THE PHASE ENTRY OF THE CONTROL DICTIONARY.
2. A 20 BYTE HEADER FOR USE AS A CORE IMAGE DIRECTORY ENTRY IS ALSO BUILT BY LINKAGE EDITOR. THIS HEADER CONTAINS THE DISK ADDRESS OF THE FIRST BLOCK OF THE PHASE PLUS ADDITIONAL PHASE/ CORE IMAGE BLOCK INFORMATION. THE 20 BYTE HEADER USES 28 BYTE HEADER INFO AND IS CREATED IN THE FINAL PHASE OF LINKAGE EDITOR.

*C5
THIS INITIALIZES THE COUNT FIELD OF COUNT, KEY, AND DATA.

*D5
THE COMPLETE SEARCH ADDRESS IS BBCCHHR. BBC ARE SET TO ZERO FOR THESE DISK APPLICATIONS.

*J2
NEXT AVAILABLE DISK ADDRESS IN CORE IMAGE LIBRARY IS CALCULATED BY RESERVING DISK SPACE FOR THE PROCESSED PHASE AS A FUNCTION OF CORE IMAGE BLOCK NUMBER.



*A4
THE AUTOLINK SWITCH, ALNKSW, IS INITIALIZED WITH ZEROS IN ALL BIT POSITIONS SIGNIFYING AN AUTOMATIC AUTOLINK OPTION FOR ALL PHASES. IF AN ACTION CARD WITH A NOAUTO OPERAND IS FOUND, THE BIT POSITIONS OF ALNKSW ARE SET TO ONES. THIS INDICATES THE AUTO-LINK FEATURE IS SUPPRESSED FOR ALL PHASES. IF AUTOLINK IS SUPPRESSED BY PHASE, ONLY THE BIT POSITION OF ALNKSW CORRESPONDING TO THAT PHASE IS SET TO ONE. THEREFORE, THE PROPER BIT POSITION FOR THIS PHASE IS SHIFTED INTO THE LOW ORDER BIT POSITION OF ALNKSW FOR TESTING BY OTHER PARTS OF THE LINKAGE EDITOR PROGRAM.

*E4
THIS READ INITIALIZES FOR THE DERDAD SUBROUTINE USED DURING TEXT CARD PROCESSING.

*G4
THE PHASE NUMBER IS THE SAME VALUE AS THE CONTROL DICTIONARY NUMBER.

Chart EG. INCLUDE Post-Processing (ATLEDT16) (Refer to Chart 06)

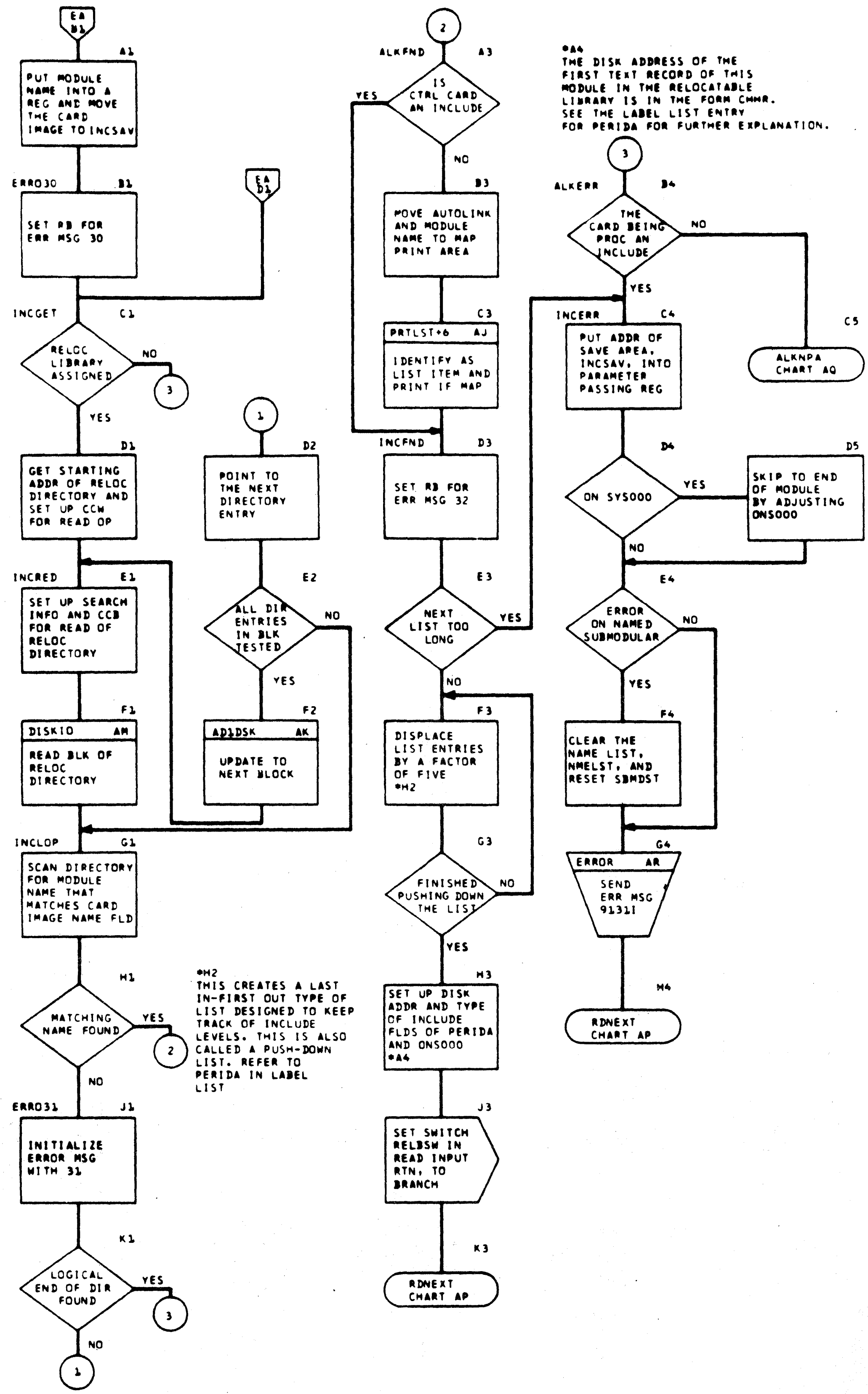
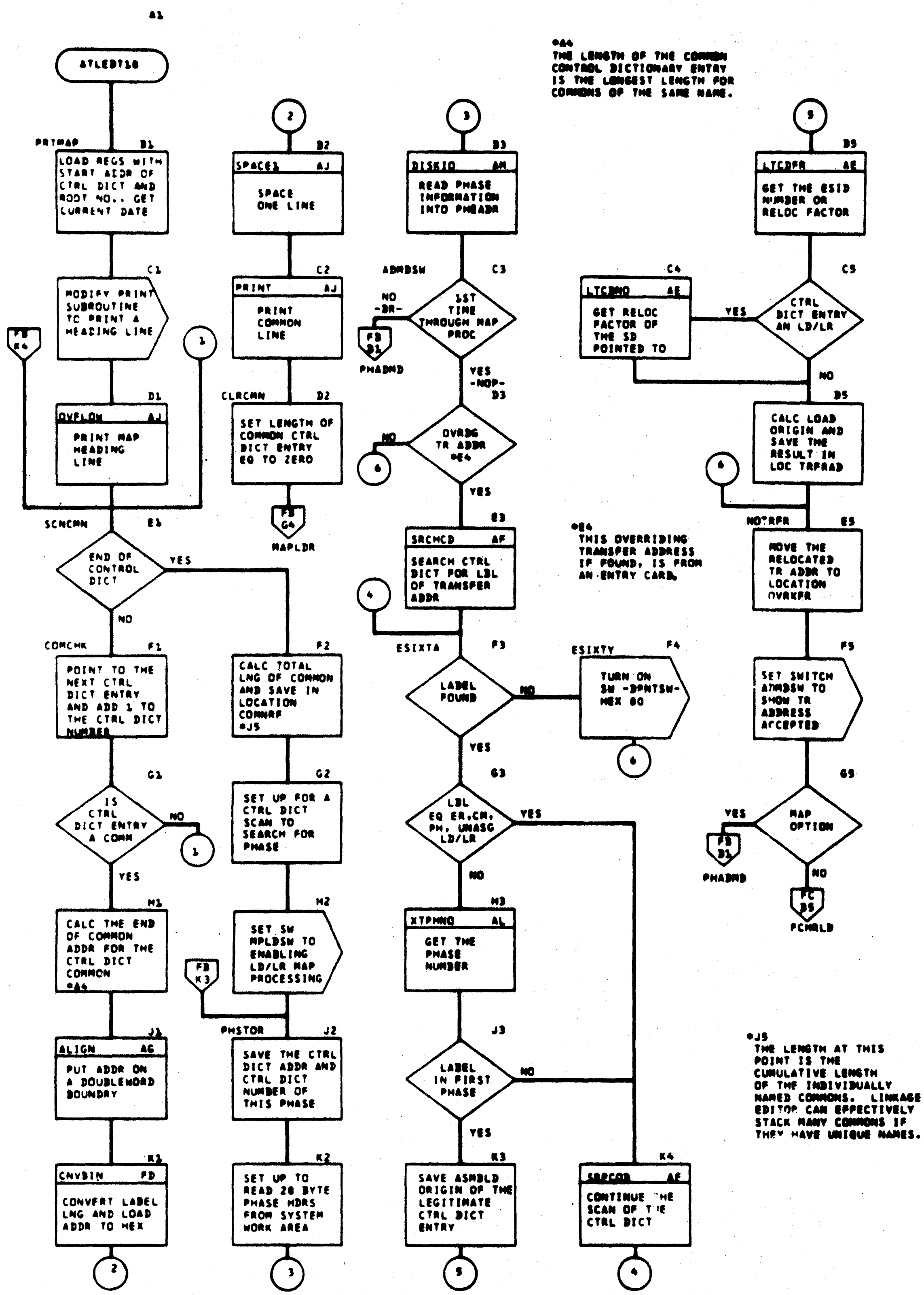
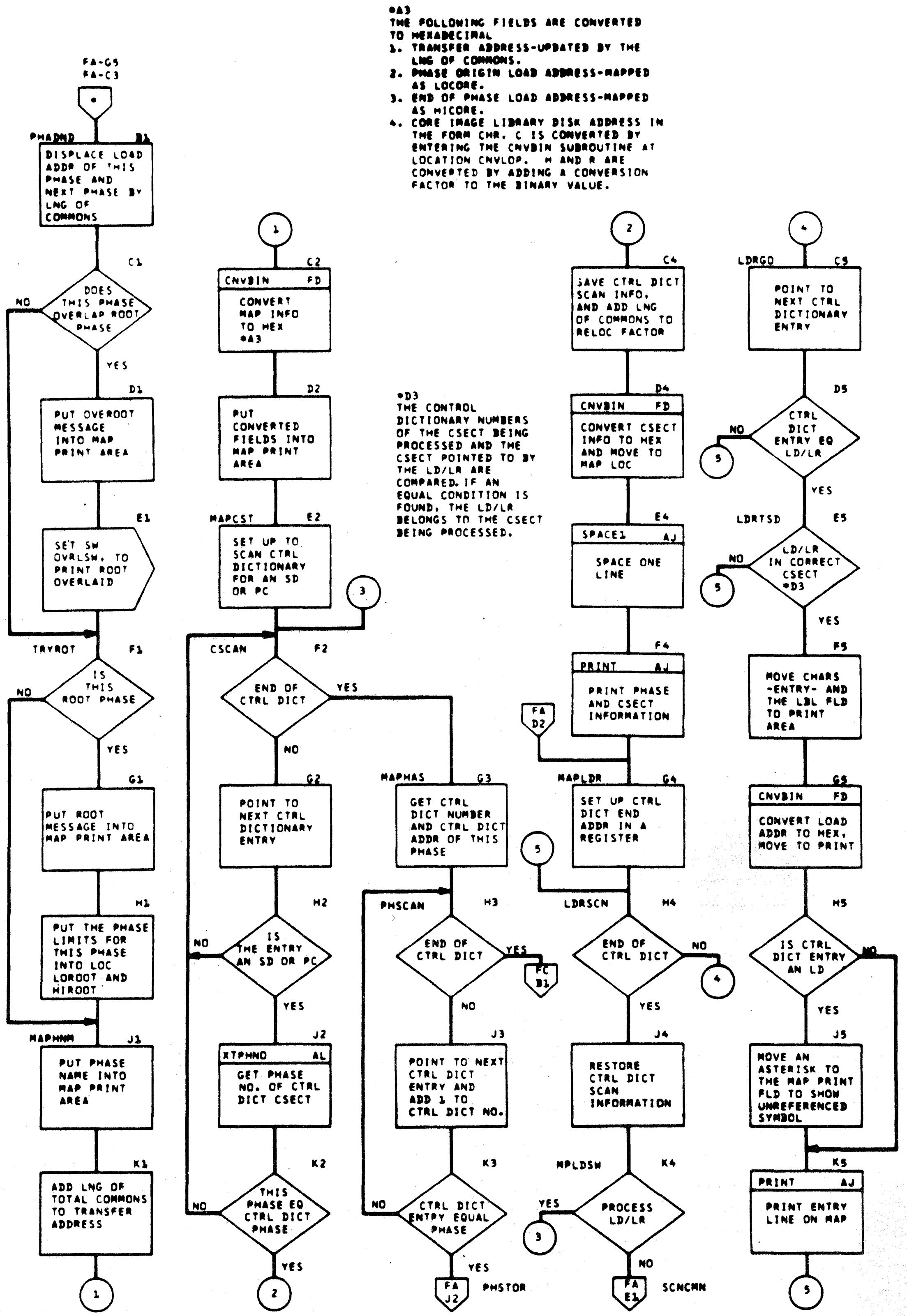
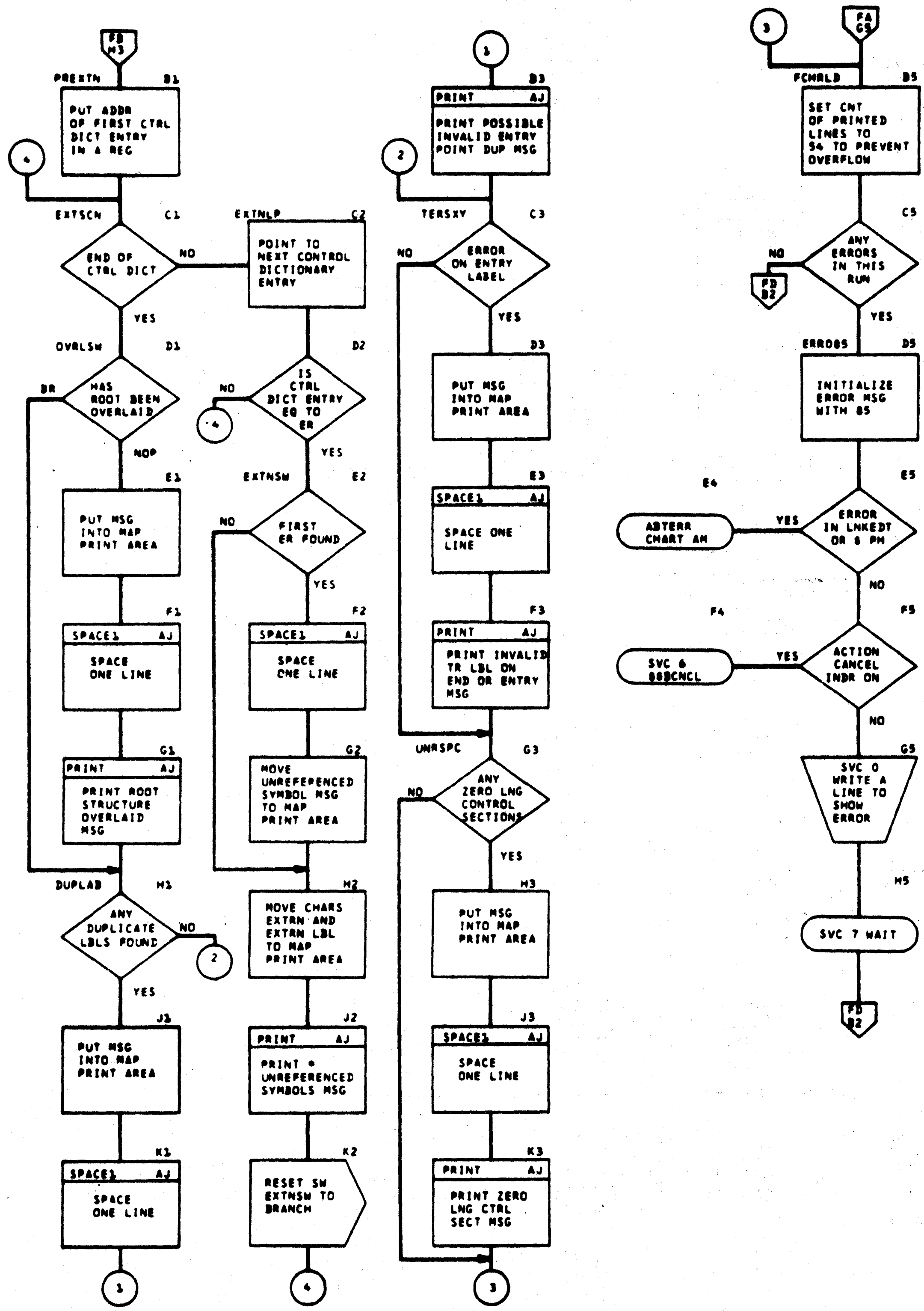


Chart FA. Print MAP (ATLEDT18) (Refer to Chart 07) (Part 1 of 4)




*A3
THE FOLLOWING FIELDS ARE CONVERTED TO HEXADECIMAL
1. TRANSFER ADDRESS-UPDATED BY THE LNG OF COMMONS.
2. PHASE ORIGIN LOAD ADDRESS-MAPPED AS LOCORE.
3. END OF PHASE LOAD ADDRESS-MAPPED AS HICORE.
4. CORE IMAGE LIBRARY DISK ADDRESS IN THE FORM CHR. C IS CONVERTED BY ENTERING THE CNVBIN SUBROUTINE AT LOCATION CNVLOP. H AND R ARE CONVERTED BY ADDING A CONVERSION FACTOR TO THE BINARY VALUE.
*D3
THE CONTROL DICTIONARY NUMBERS OF THE CSECT BEING PROCESSED AND THE CSECT POINTED TO BY THE LD/LR ARE COMPARED. IF AN EQUAL CONDITION IS FOUND, THE LD/LR BELONGS TO THE CSECT BEING PROCESSED.
FA-G5
FA-C3
PHADND B1
DISPLACE LOAD ADDR OF THIS PHASE AND NEXT PHASE BY LNG OF COMMONS
C1
DOES THIS PHASE OVERLAP ROOT PHASE
NO
YES
D1
PUT OVEROOT MESSAGE INTO MAP PRINT AREA
E1
SET SW OVRLSW, TO PRINT ROOT OVERLAID
TRYROT F1
IS THIS ROOT PHASE
G1
PUT ROOT MESSAGE INTO MAP PRINT AREA
H1
PUT THE PHASE LIMITS FOR THIS PHASE INTO LOC LOROOT AND HIROOT
MAPHNM J1
PUT PHASE NAME INTO MAP PRINT AREA
K1
ADD LNG OF TOTAL COMMONS TO TRANSFER ADDRESS
1
C2
CNVBIN FD
CONVERT MAP INFO TO HEX *A3
D2
PUT CONVERTED FIELDS INTO MAP PRINT AREA
MAPCST E2
SET UP TO SCAN CTRL DICTIONARY FOR AN SD OR PC
3
CSCAN F2
END OF CTRL DICT
G2
POINT TO NEXT CTRL DICTIONARY ENTRY
H2
IS THE ENTRY AN SD OR PC
XTPHNO AL J2
GET PHASE NO. OF CTRL DICT CSECT
K2
THIS PHASE EQ CTRL DICT PHASE
2
MAPHAS G3
GET CTRL DICT NUMBER AND CTRL DICT ADDR OF THIS PHASE
PHSCAN H3
END OF CTRL DICT
FC B1
J3
POINT TO NEXT CTRL DICT ENTRY AND ADD 1 TO CTRL DICT NO.
K3
CTRL DICT ENTPY EQUAL PHASE
FA J2
PHSTOR
C4
SAVE CTRL DICT SCAN INFO, AND ADD LNG OF COMMONS TO RELOC FACTOR
D4
CNVBIN FD
CONVERT CSECT INFO TO HEX AND MOVE TO MAP LOC
E4
SPACE1 AJ
SPACE ONE LINE
F4
PRINT AJ
PRINT PHASE AND CSECT INFORMATION
FA D2
MAPLDR G4
SET UP CTRL DICT END ADDR IN A REGISTER
5
LDRSCN H4
END OF CTRL DICT
4
J4
RESTORE CTRL DICT SCAN INFORMATION
MPLDSW K4
PROCESS LD/LR
FA E1
SCNCMN
LDRGO C5
POINT TO NEXT CTRL DICTIONARY ENTRY
D5
CTRL DICT ENTRY EQ LD/LR
LDRTSD E5
LD/LR IN CORRECT CSECT *D3
F5
MOVE CHARS -ENTRY- AND THE LBL FLD TO PRINT AREA
G5
CNVBIN FD
CONVERT LOAD ADDR TO HEX, MOVE TO PRINT
H5
IS CTRL DICT ENTRY AN LD
J5
MOVE AN ASTERISK TO THE MAP PRINT FLD TO SHOW UNREFERENCED SYMBOL
K5
PRINT AJ
PRINT ENTRY LINE ON MAP


FB M3
PREXTN B1
PUT ADDR OF FIRST CTRL DICT ENTRY IN A REG
EXTSCN C1
END OF CTRL DICT
NO
YES
EXTNLP C2
POINT TO NEXT CONTROL DICTIONARY ENTRY
OVRLSW D1
HAS ROOT BEEN OVERLAID
BR
NOP
D2
IS CTRL DICT ENTRY EQ TO ER
NO
YES
E1
PUT MSG INTO MAP PRINT AREA
EXTNSW E2
FIRST ER FOUND
NO
YES
F1
SPACE1 AJ
SPACE ONE LINE
F2
SPACE1 AJ
SPACE ONE LINE
G1
PRINT AJ
PRINT ROOT STRUCTURE OVERLAID MSG
G2
MOVE UNREFERENCED SYMBOL MSG TO MAP PRINT AREA
DUPLAB H1
ANY DUPLICATE LBLS FOUND
NO
YES
H2
MOVE CHARS EXTRN AND EXTRN LBL TO MAP PRINT AREA
J1
PUT MSG INTO MAP PRINT AREA
J2
PRINT AJ
PRINT * UNREFERENCED SYMBOLS MSG
K1
SPACE1 AJ
SPACE ONE LINE
K2
RESET SW EXTNSW TO BRANCH
B3
PRINT AJ
PRINT POSSIBLE INVALID ENTRY POINT DUP MSG
TERSXV C3
ERROR ON ENTRY LABEL
NO
YES
D3
PUT MSG INTO MAP PRINT AREA
E3
SPACE1 AJ
SPACE ONE LINE
F3
PRINT AJ
PRINT INVALID TR LBL ON END OR ENTRY MSG
UNRSPC G3
ANY ZERO LNG CONTROL SECTIONS
NO
YES
H3
PUT MSG INTO MAP PRINT AREA
J3
SPACE1 AJ
SPACE ONE LINE
K3
PRINT AJ
PRINT ZERO LNG CTRL SECT MSG
FA G5
FCHRLD B5
SET CNT OF PRINTED LINES TO 54 TO PREVENT OVERFLOW
C5
ANY ERRORS IN THIS RUN
NO
FD B2
YES
ERROR5 D5
INITIALIZE ERROR MSG WITH 85
E4
E5
ABTERR CHART AM
YES
ERROR IN LNKEDT OR 8 PH
NO
F4
F5
SVC 6 $$BCNCL
YES
ACTION CANCEL INDR ON
NO
G5
SVC 0 WRITE A LINE TO SHOW ERROR
H5
SVC 7 WAIT
FD B2

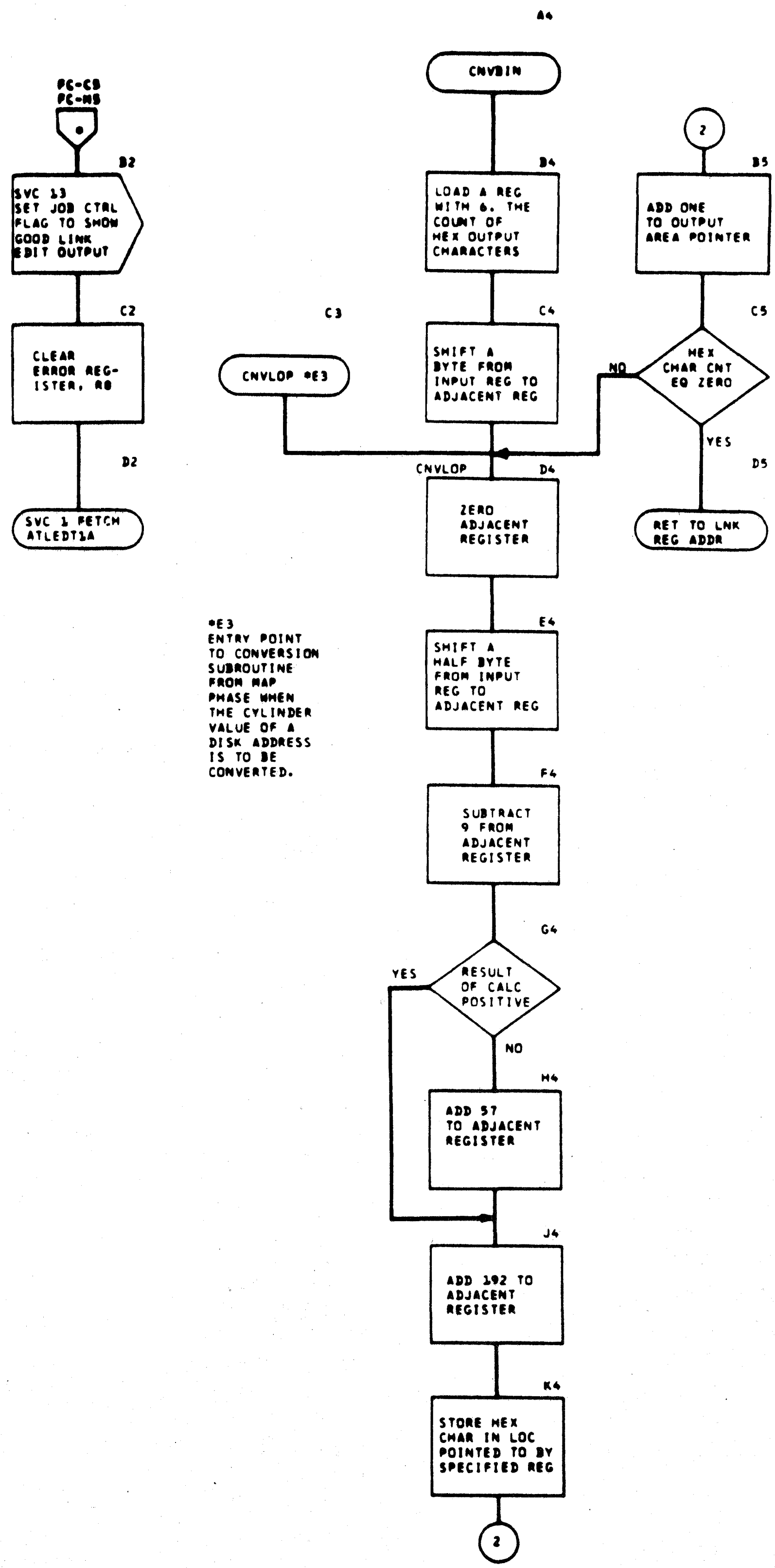

A4
CNVBIN
FC-C8
FC-H8
B2
SVC 13 SET JOB CTRL FLAG TO SHOW GOOD LINK EDIT OUTPUT
B4
LOAD A REG WITH 6, THE COUNT OF HEX OUTPUT CHARACTERS
2
B5
ADD ONE TO OUTPUT AREA POINTER
C2
CLEAR ERROR REG-ISTER, R8
C3
CNVLOP *E3
C4
SHIFT A BYTE FROM INPUT REG TO ADJACENT REG
C5
HEX CHAR CNT EQ ZERO
NO
YES
D2
SVC 1 FETCH ATLEDT1A
CNVLOP
D4
ZERO ADJACENT REGISTER
D5
RET TO LNK REG ADDR
*E3 ENTRY POINT TO CONVERSION SUBROUTINE FROM MAP PHASE WHEN THE CYLINDER VALUE OF A DISK ADDRESS IS TO BE CONVERTED.
E4
SHIFT A HALF BYTE FROM INPUT REG TO ADJACENT REG
F4
SUBTRACT 9 FROM ADJACENT REGISTER
G4
RESULT OF CALC POSITIVE
YES
NO
H4
ADD 57 TO ADJACENT REGISTER
J4
ADD 192 TO ADJACENT REGISTER
K4
STORE HEX CHAR IN LOC POINTED TO BY SPECIFIED REG
2

Chart GA. Pass 2 P-Pointer Processor (ATLEDT1A) (Refer to Chart 08)

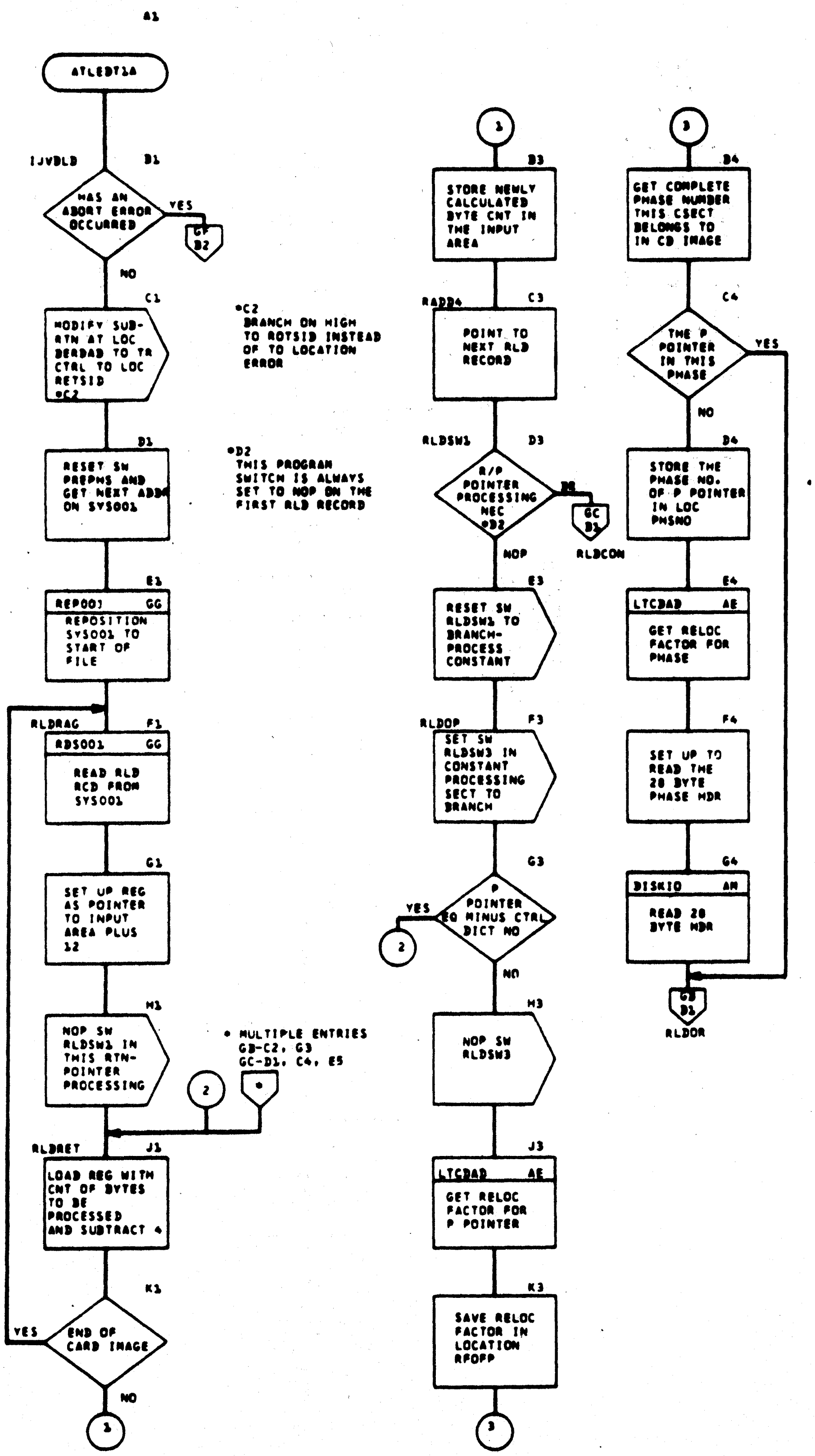
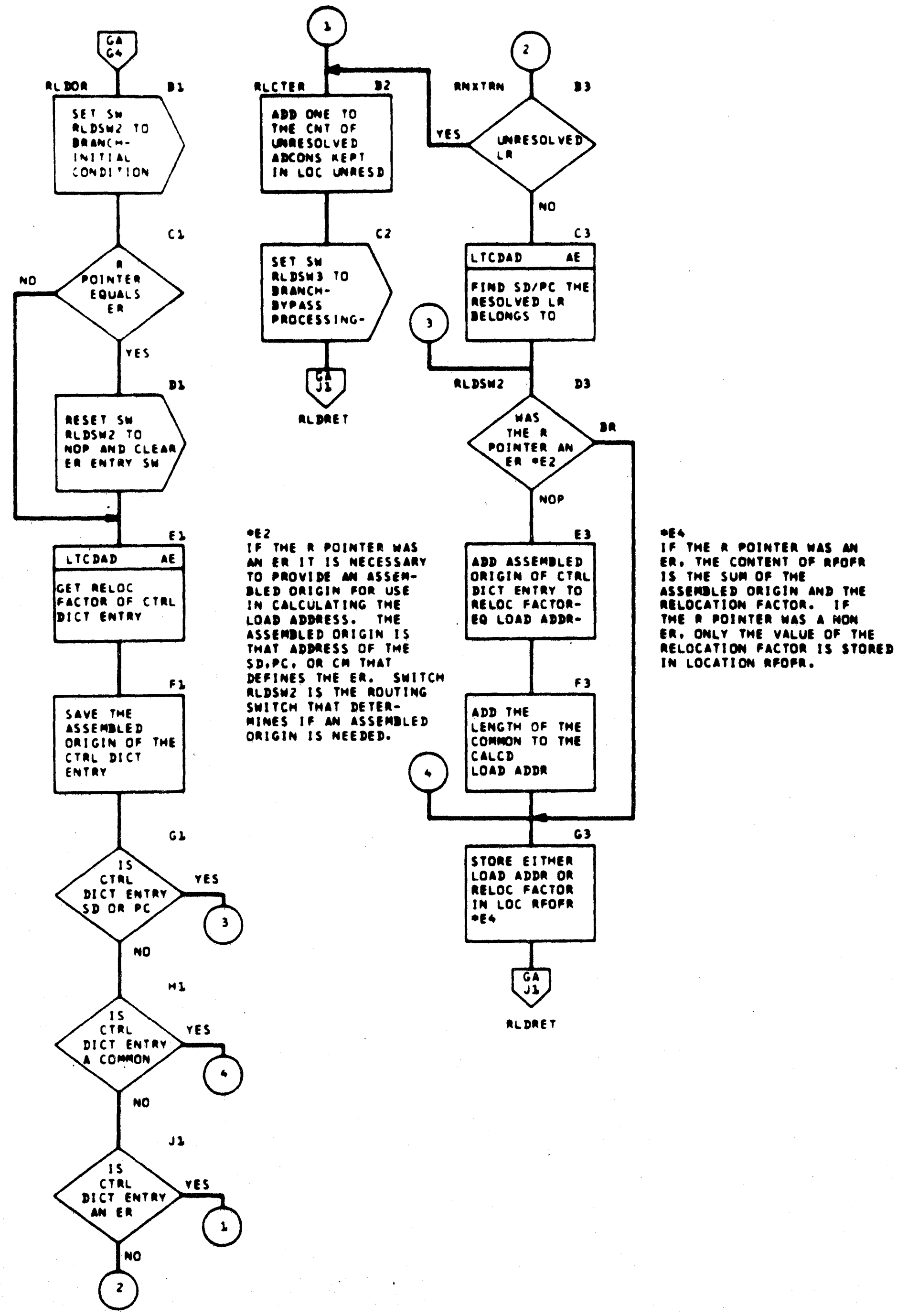
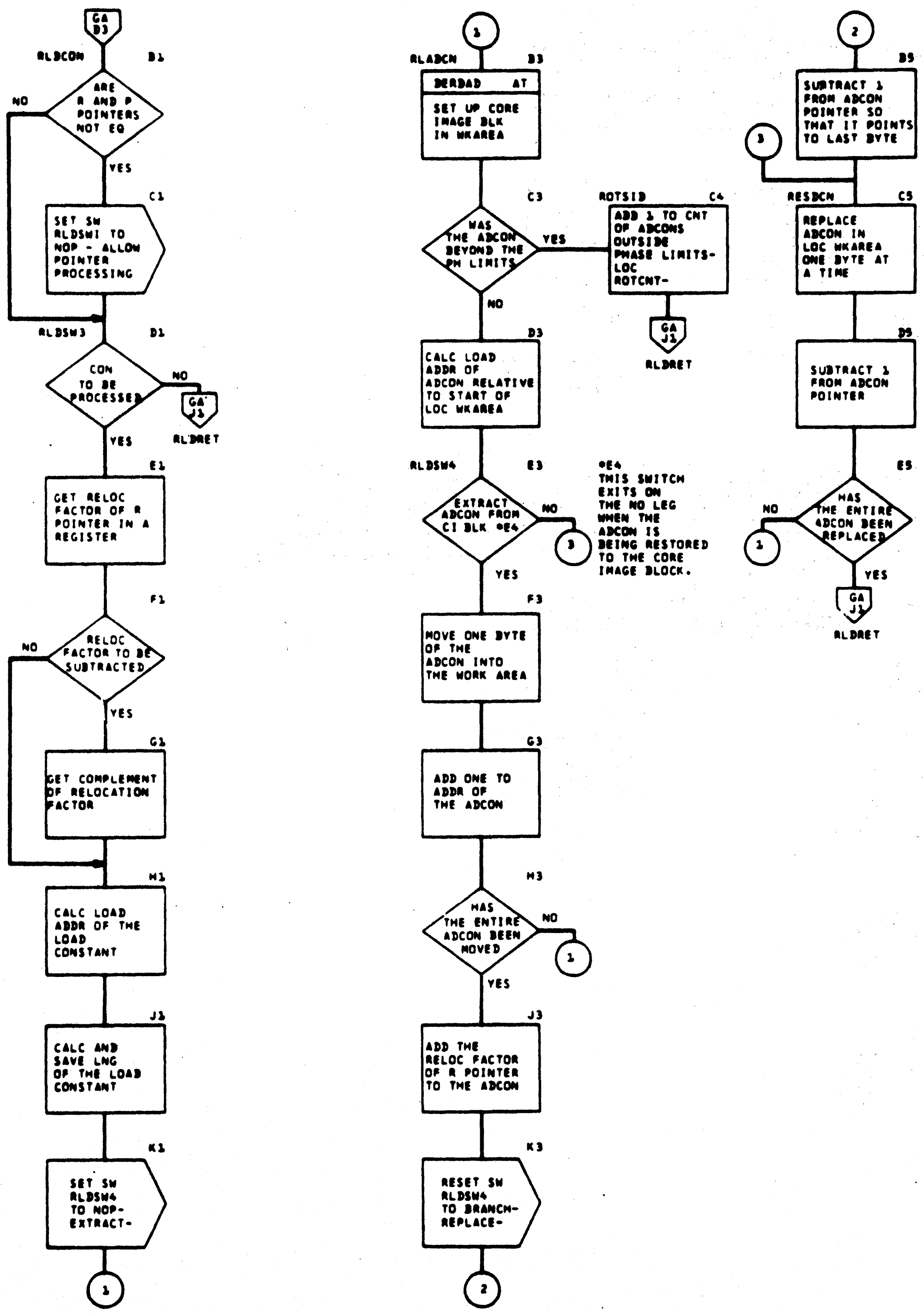
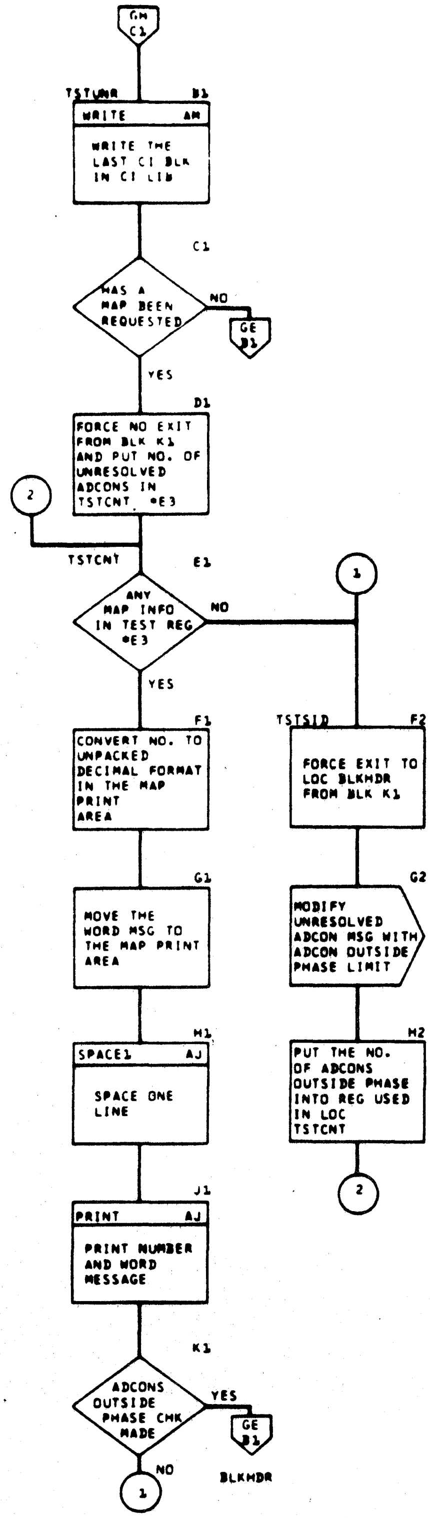


Chart GB. Pass 2 R-Pointer Processor (ATLEDT1A) (Refer to Chart 08)




GA B3
RLDCON
B1
ARE R AND P POINTERS NOT EQ
NO
YES
C1
SET SW RLDSW1 TO NOP - ALLOW POINTER PROCESSING
RLDSW3
D1
CON TO BE PROCESSED
NO
GA J1
RLDRET
YES
E1
GET RELOC FACTOR OF R POINTER IN A REGISTER
F1
RELOC FACTOR TO BE SUBTRACTED
NO
YES
G1
GET COMPLEMENT OF RELOCATION FACTOR
H1
CALC LOAD ADDR OF THE LOAD CONSTANT
J1
CALC AND SAVE LNG OF THE LOAD CONSTANT
K1
SET SW RLDSW4 TO NOP-EXTRACT-
1
1
RLADCN
B3
DERDAD AT
SET UP CORE IMAGE BLK IN WKAREA
C3
HAS THE ADCON BEYOND THE PH LIMITS
YES
NO
ROTSID
C4
ADD 1 TO CNT OF ADCONS OUTSIDE PHASE LIMITS-LOC ROTCNT-
GA J1
RLDRET
D3
CALC LOAD ADDR OF ADCON RELATIVE TO START OF LOC WKAREA
RLDSW4
E3
EXTRACT ADCON FROM CI BLK *E4
NO
3
YES
*E4
THIS SWITCH EXITS ON THE NO LEG WHEN THE ADCON IS BEING RESTORED TO THE CORE IMAGE BLOCK.
F3
MOVE ONE BYTE OF THE ADCON INTO THE WORK AREA
G3
ADD ONE TO ADDR OF THE ADCON
H3
HAS THE ENTIRE ADCON BEEN MOVED
NO
1
YES
J3
ADD THE RELOC FACTOR OF R POINTER TO THE ADCON
K3
RESET SW RLDSW4 TO BRANCH-REPLACE-
2
2
B5
SUBTRACT 1 FROM ADCON POINTER SO THAT IT POINTS TO LAST BYTE
3
RESDCN
C5
REPLACE ADCON IN LOC WKAREA ONE BYTE AT A TIME
D5
SUBTRACT 1 FROM ADCON POINTER
E5
HAS THE ENTIRE ADCON BEEN REPLACED
NO
1
YES
GA J1
RLDRET

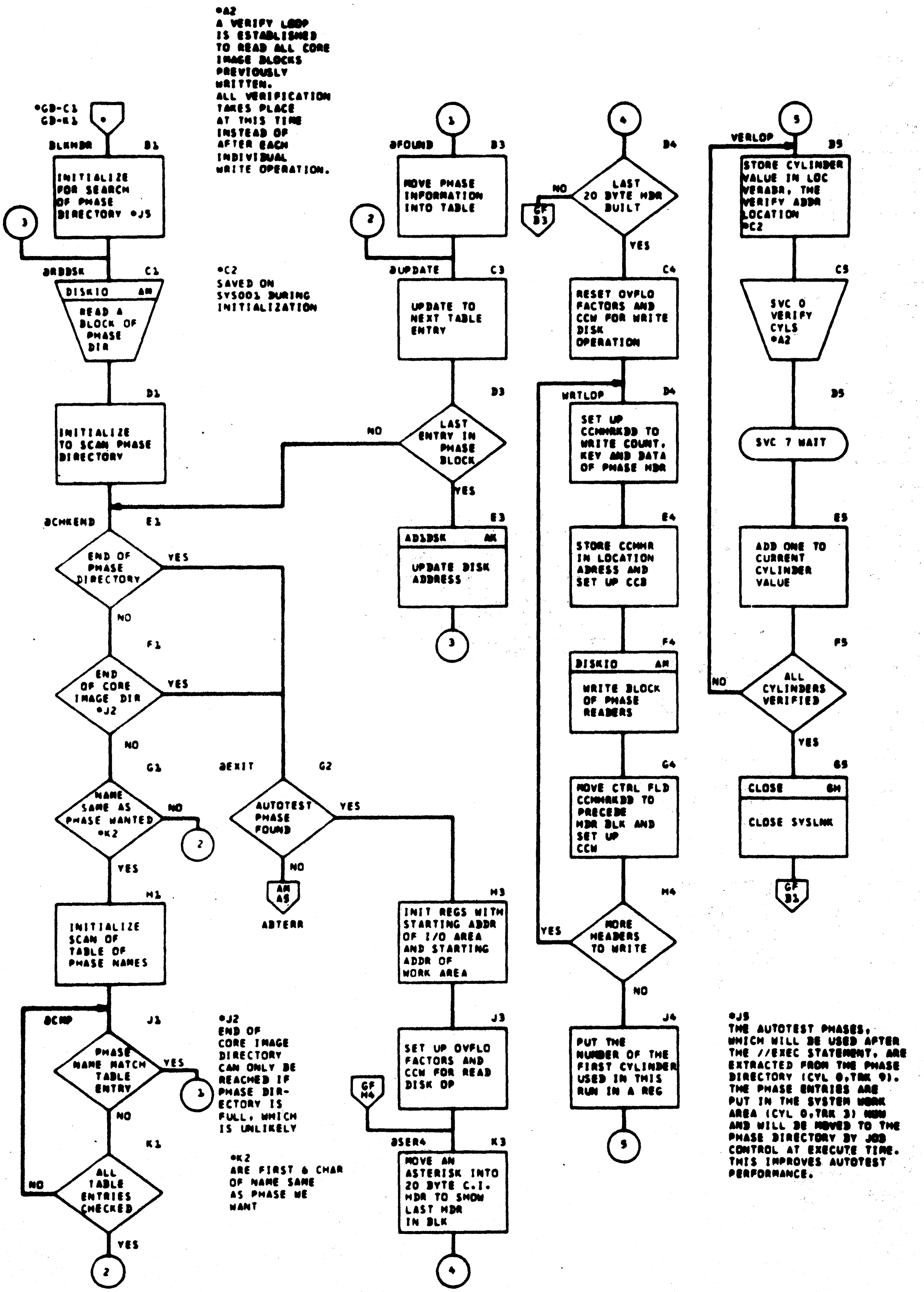


*E3
STARTING AT LOCATION TSTCNT, THE ROUTINE IS A GENERALIZATION TO PRINT---

1. THE NUMBER OF UNRESOLVED ADCONS, IF ANY, ADD THE WORD MESSAGE -UNRESOLVED ADDRESS CONSTANTS-
2. THE NUMBER OF ADCONS OUTSIDE THE PHASE LIMITS, IF ANY, AND THE WORD MESSAGE- - ADDRESS CONSTANTS OUTSIDE THE LIMITS OF THE PHASE-.

THE PRINTING IS DONE IN THE ORDER SPECIFIED ABOVE.

Chart GE. Pass 2 Block Phase Header (ATLEDT1A) (Refer to Chart 08) (Part 1 of 2)



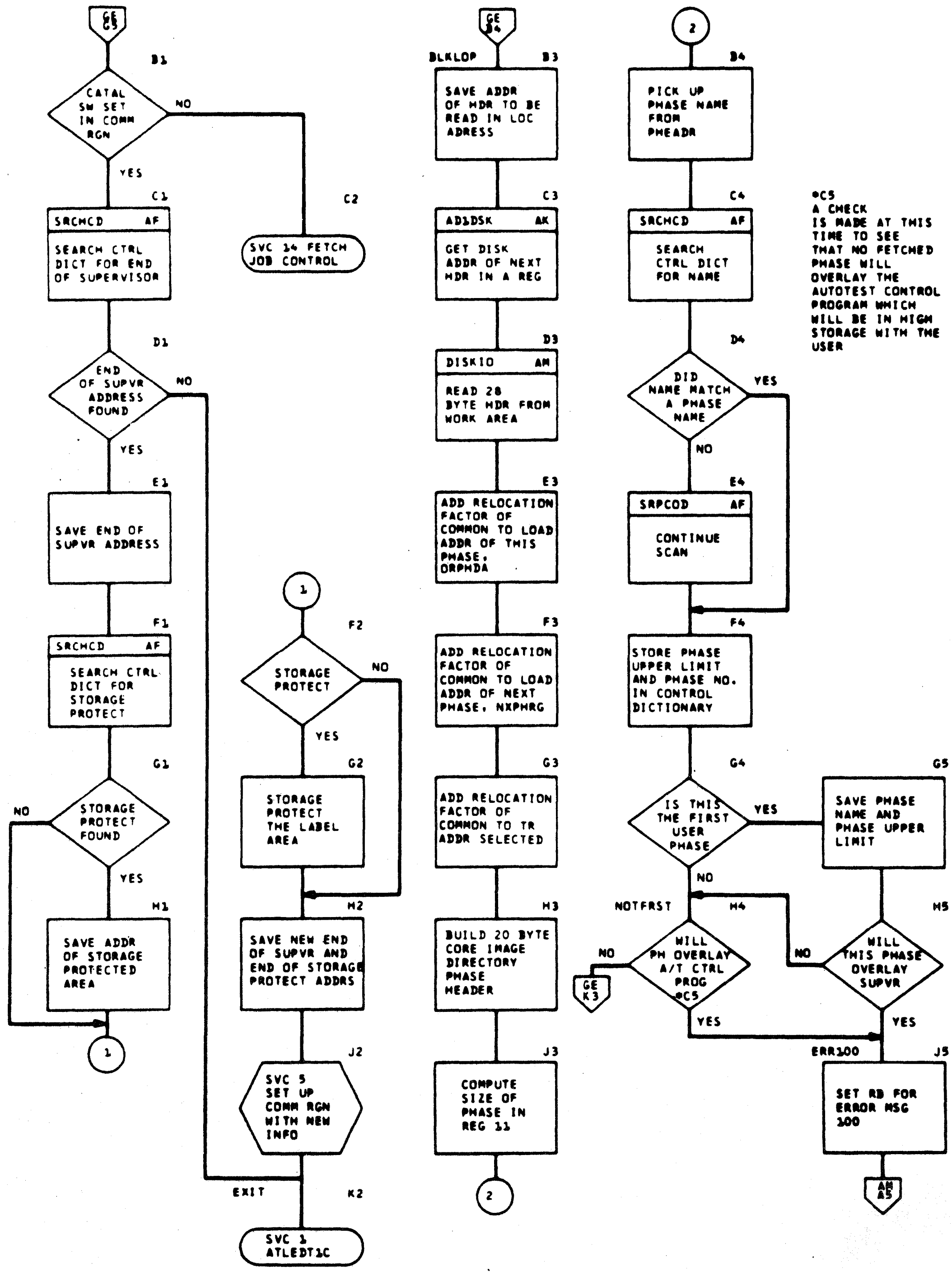

GE C5
B1
CATAL SW SET IN COMM RGN
NO
YES
C1
SRCHCD AF
SEARCH CTRL DICT FOR END OF SUPERVISOR
C2
SVC 14 FETCH JOB CONTROL
D1
END OF SUPVR ADDRESS FOUND
E1
SAVE END OF SUPVR ADDRESS
F1
SRCHCD AF
SEARCH CTRL DICT FOR STORAGE PROTECT
G1
STORAGE PROTECT FOUND
H1
SAVE ADDR OF STORAGE PROTECTED AREA
1
F2
STORAGE PROTECT
G2
STORAGE PROTECT THE LABEL AREA
H2
SAVE NEW END OF SUPVR AND END OF STORAGE PROTECT ADDRS
J2
SVC 5 SET UP COMM RGN WITH NEW INFO
EXIT
K2
SVC 1 ATLEDT1C
GE B4
BLKLOP
B3
SAVE ADDR OF HDR TO BE READ IN LOC ADRESS
C3
AD1DSK AK
GET DISK ADDR OF NEXT HDR IN A REG
D3
DISKIO AM
READ 28 BYTE HDR FROM WORK AREA
E3
ADD RELOCATION FACTOR OF COMMON TO LOAD ADDR OF THIS PHASE, ORPHDA
F3
ADD RELOCATION FACTOR OF COMMON TO LOAD ADDR OF NEXT PHASE, NXPHRG
G3
ADD RELOCATION FACTOR OF COMMON TO TR ADDR SELECTED
H3
BUILD 20 BYTE CORE IMAGE DIRECTORY PHASE HEADER
J3
COMPUTE SIZE OF PHASE IN REG 11
2
B4
PICK UP PHASE NAME FROM PHEADR
C4
SRCHCD AF
SEARCH CTRL DICT FOR NAME
*C5
A CHECK IS MADE AT THIS TIME TO SEE THAT NO FETCHED PHASE WILL OVERLAY THE AUTOTEST CONTROL PROGRAM WHICH WILL BE IN HIGH STORAGE WITH THE USER
D4
DID NAME MATCH A PHASE NAME
E4
SRPCOD AF
CONTINUE SCAN
F4
STORE PHASE UPPER LIMIT AND PHASE NO. IN CONTROL DICTIONARY
G4
IS THIS THE FIRST USER PHASE
G5
SAVE PHASE NAME AND PHASE UPPER LIMIT
NOTFRST
H4
WILL PH OVERLAY A/T CTRL PROG *C5
GE K3
H5
WILL THIS PHASE OVERLAY SUPVR
ERR100
J5
SET RB FOR ERROR MSG 100
AM A5

Chart GG. Pass 2 Abort Error Processor (ATLEDT1A) (Refer to Chart 08)

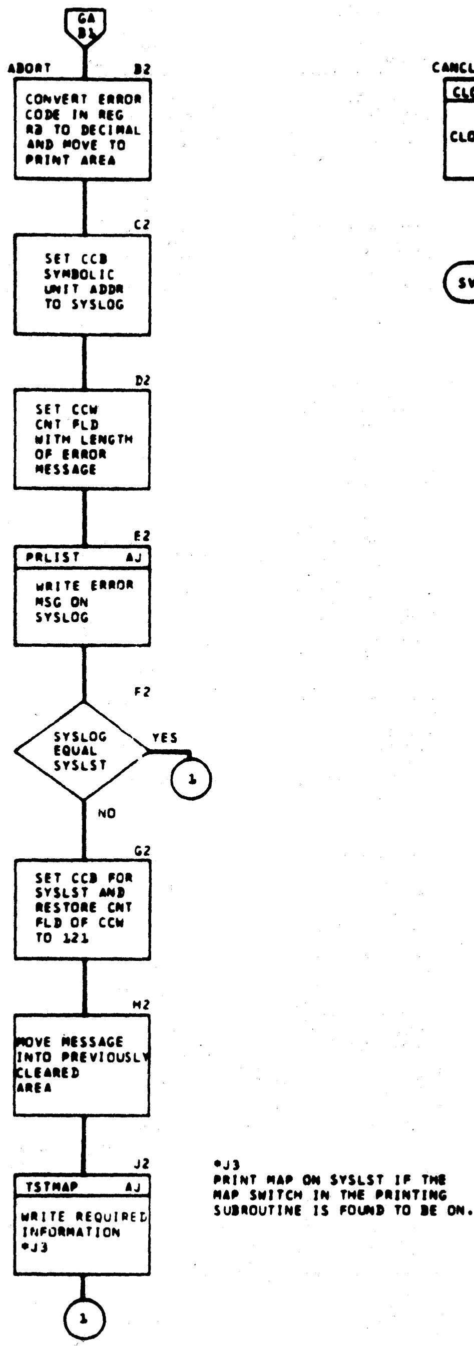


Chart GH. Subroutines (ATLEDT1A) (Refer to Chart 08)

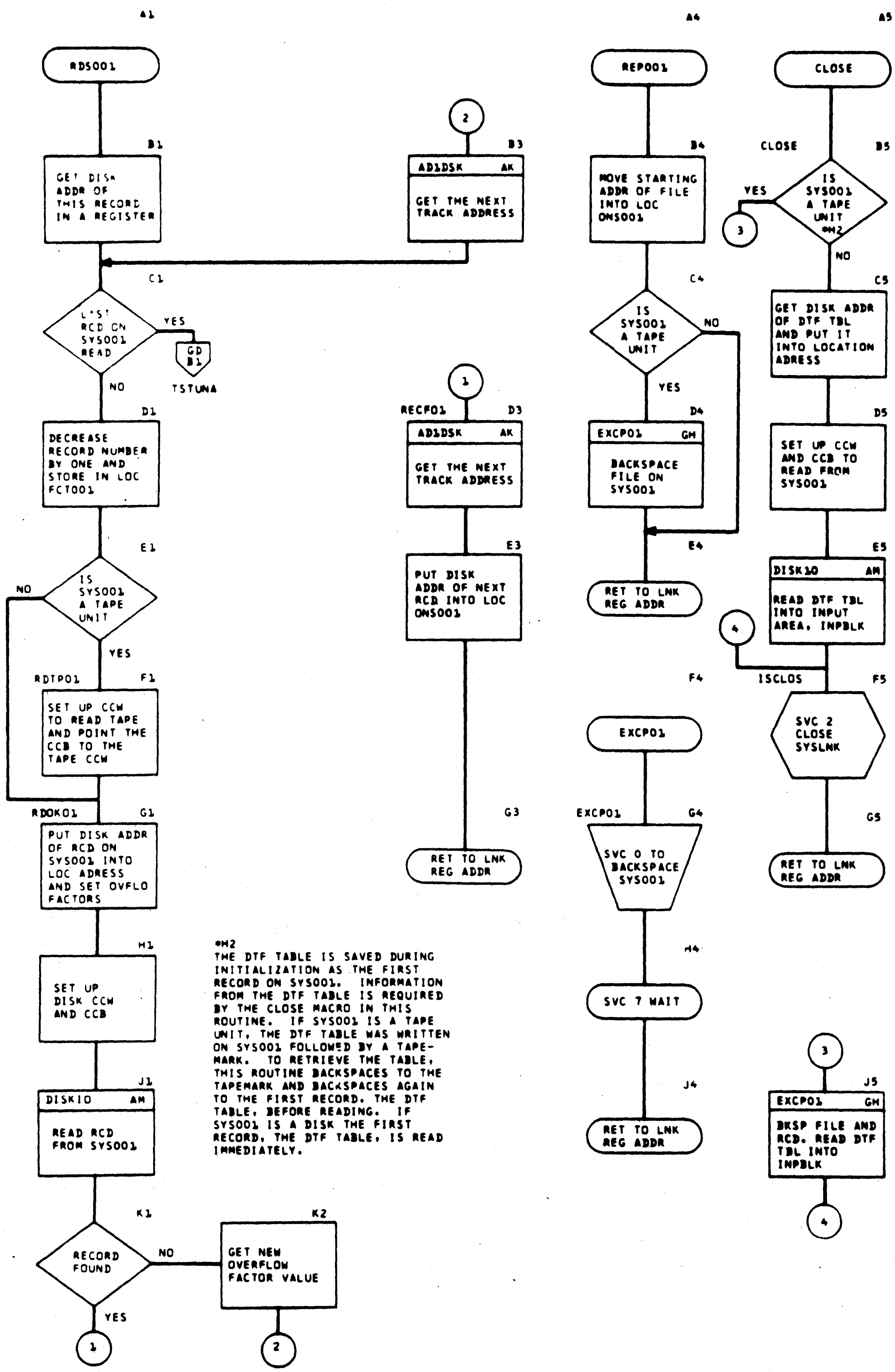


Chart HA. SYM Card Processor (ATLEDT1B) (Refer to Chart 09)

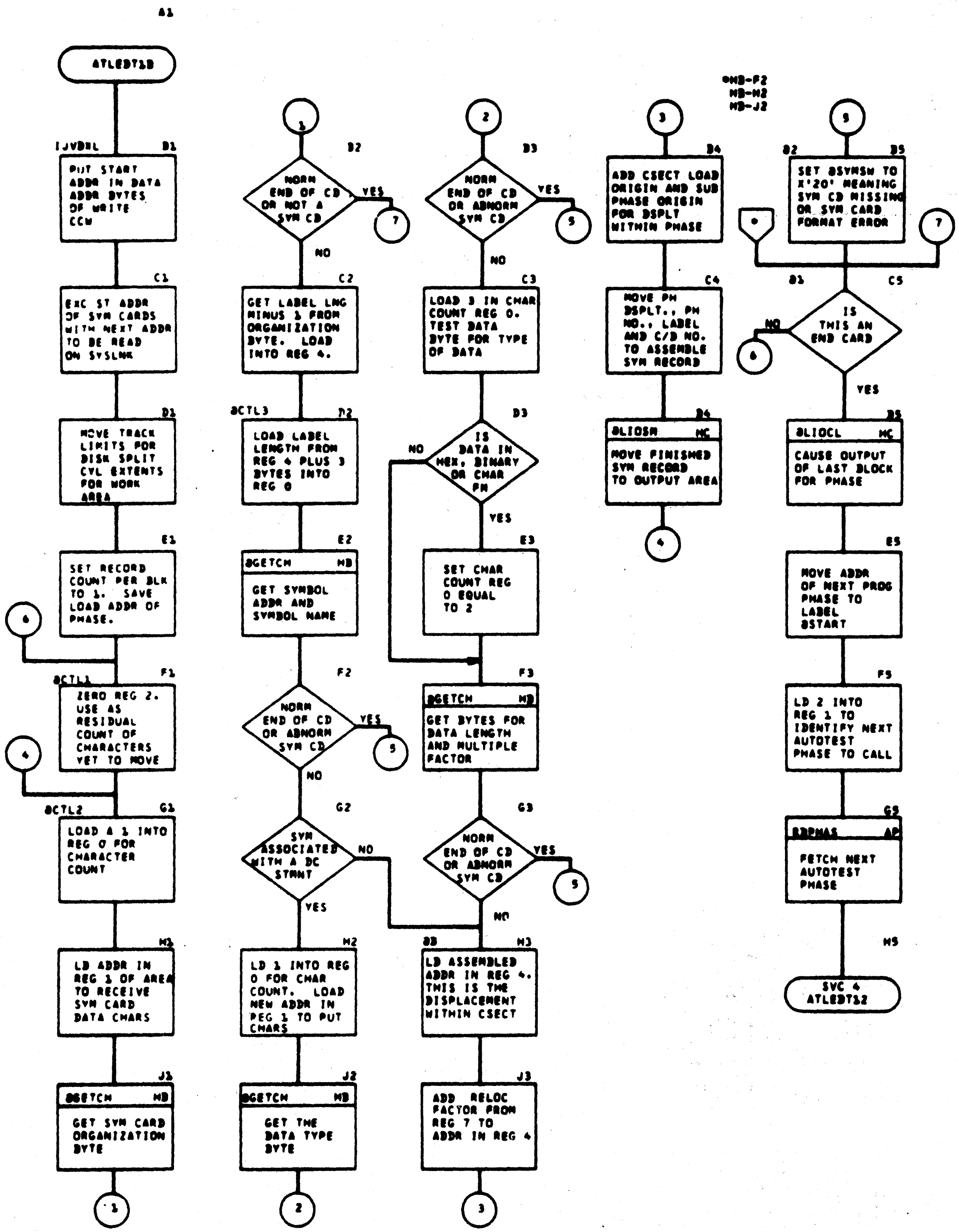
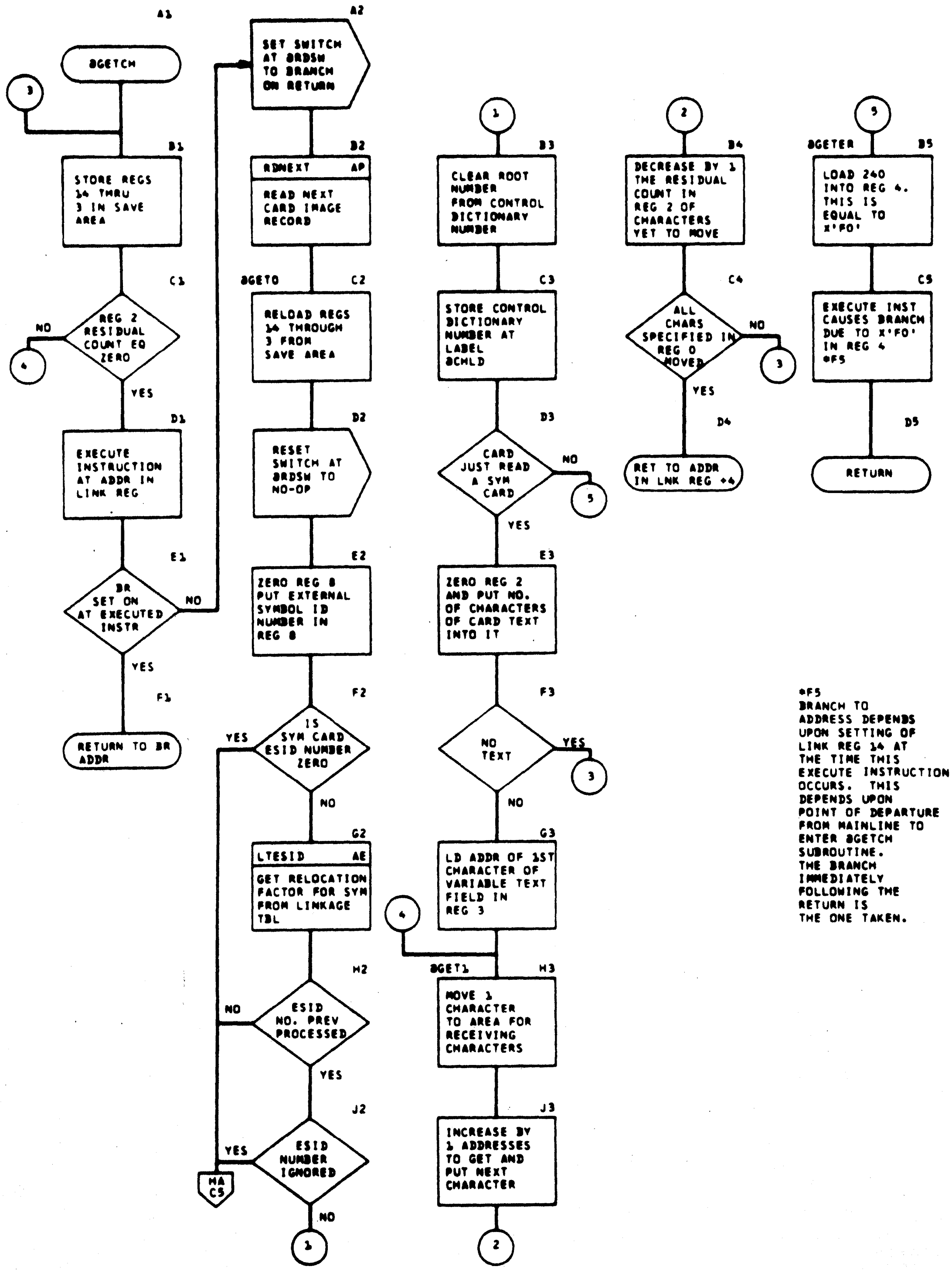


Chart HB. @GETCH Subroutine (ATLEDT1B) (Refer to Chart 09)



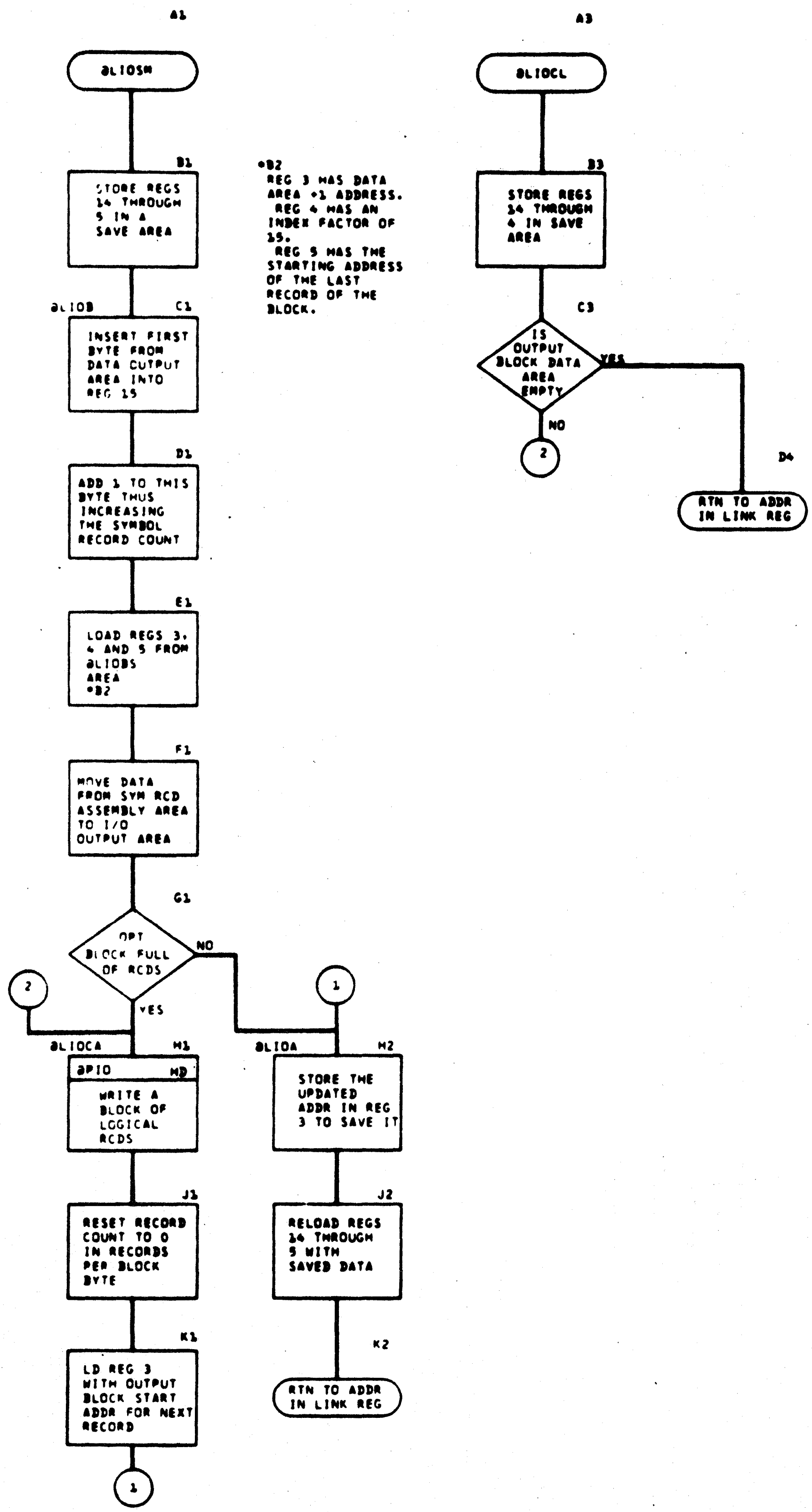
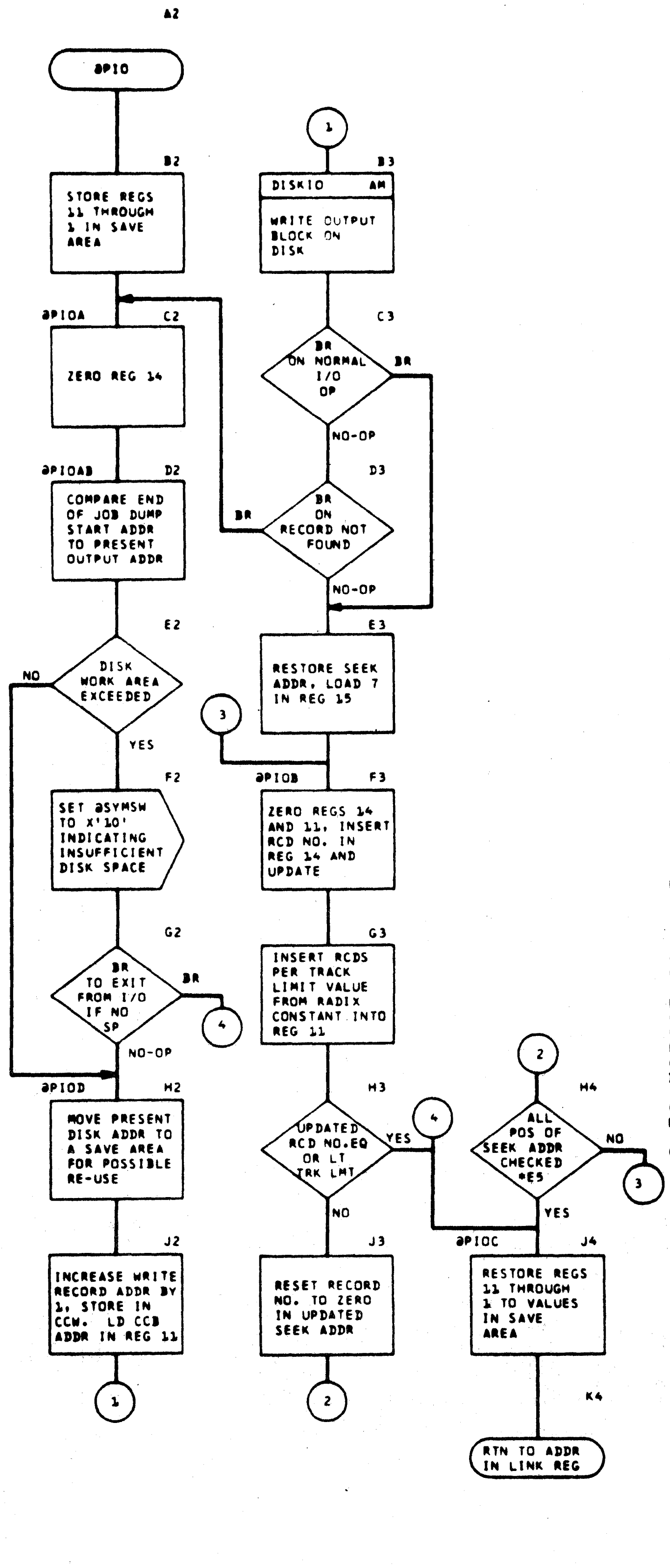

A1
@LIOSM
B1
STORE REGS 14 THROUGH 5 IN A SAVE AREA
*B2
REG 3 HAS DATA AREA +1 ADDRESS. REG 4 HAS AN INDEX FACTOR OF 15. REG 5 HAS THE STARTING ADDRESS OF THE LAST RECORD OF THE BLOCK.
@LIOB
C1
INSERT FIRST BYTE FROM DATA OUTPUT AREA INTO REG 15
D1
ADD 1 TO THIS BYTE THUS INCREASING THE SYMBOL RECORD COUNT
E1
LOAD REGS 3, 4 AND 5 FROM @LIOBS AREA *B2
F1
MOVE DATA FROM SYM RCD ASSEMBLY AREA TO I/O OUTPUT AREA
G1
OPT BLOCK FULL OF RCDS
NO
YES
2
1
@LIOCA
H1
@PIO
HD
WRITE A BLOCK OF LOGICAL RCDS
@LIOA
H2
STORE THE UPDATED ADDR IN REG 3 TO SAVE IT
J1
RESET RECORD COUNT TO 0 IN RECORDS PER BLOCK BYTE
J2
RELOAD REGS 14 THROUGH 5 WITH SAVED DATA
K1
LD REG 3 WITH OUTPUT BLOCK START ADDR FOR NEXT RECORD
K2
RTN TO ADDR IN LINK REG
1
A3
@LIOCL
B3
STORE REGS 14 THROUGH 4 IN SAVE AREA
C3
IS OUTPUT BLOCK DATA AREA EMPTY
YES
NO
2
D4
RTN TO ADDR IN LINK REG

Chart HD. @PIO Subroutine (ATLEDT1B) (Refer to Chart 09)



*E5
THE FIELD CALLED RADIX CONSTANT CONTAINS LIMITING VALUES FOR RECORDS PER TRACK, HEAD VALUE, AND CYLINDER VALUE. IT IS USED IN THIS UPDATE LOOP PROCEDURE. REGISTER 15 IS DECREASED BY 1 EACH TIME THROUGH LOOP, THEREBY POINTING AT EACH BYTE OF THE SEEK ADDRESS TO SEE IF UPDATING REQUIRES RESET OF RECORD OR HEAD VALUE. EXIT FROM LOOP OCCURS WHEN RECORD OR HEAD OR CYLINDER DOES NOT EXCEED LIMITS IN RADIX CONSTANT. THEREFORE, ONLY RECORD IS UPDATED UNTIL LIMIT EXCEEDED WHICH CAUSES RESET OF RECORD NUMBER AND UPDATE OF HEAD NUMBER. EXCEEDING HEAD LIMIT CAUSES ITS VALUE TO BE RESET AND CYLINDER TO BE UPDATED. CYLINDER LIMIT SHOULD NEVER BE EXCEEDED DUE TO OTHER PREVIOUS CHECKS.

Chart JA. Write Control Dictionary and Autotest Communication Area on SYSLNK (ATLEDT1C) (Refer to Chart 10) (Part 1 of 3)

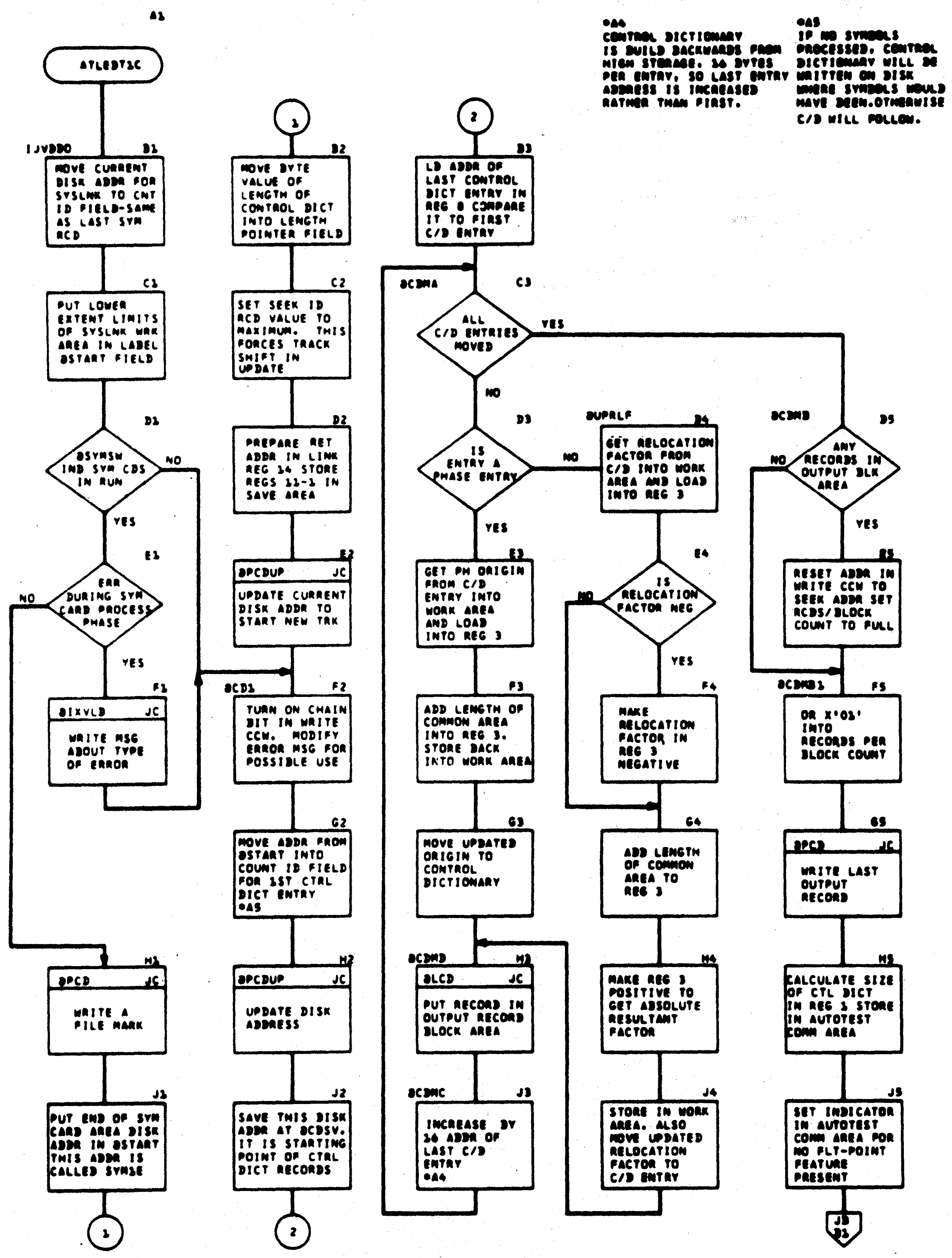


Chart JB. Write Control Dictionary and Autotest Communication Area on SYSLNK (ATLEDT1C) (Refer to Chart 10) (Part 2 of 3)

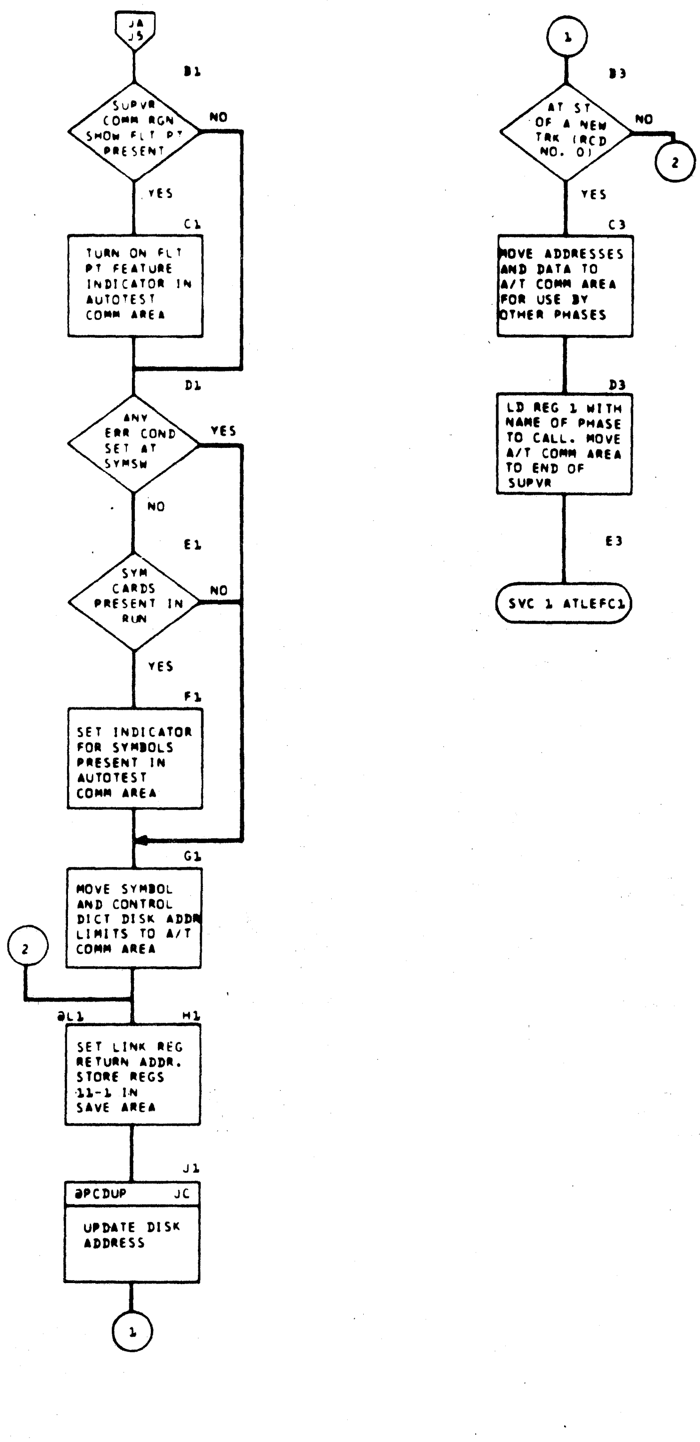


Chart JC. Write Control Dictionary and Autotest Communication Area on SYSLNK (ATLEDT1C) (Refer to Chart 10) (Part 3 of 3)

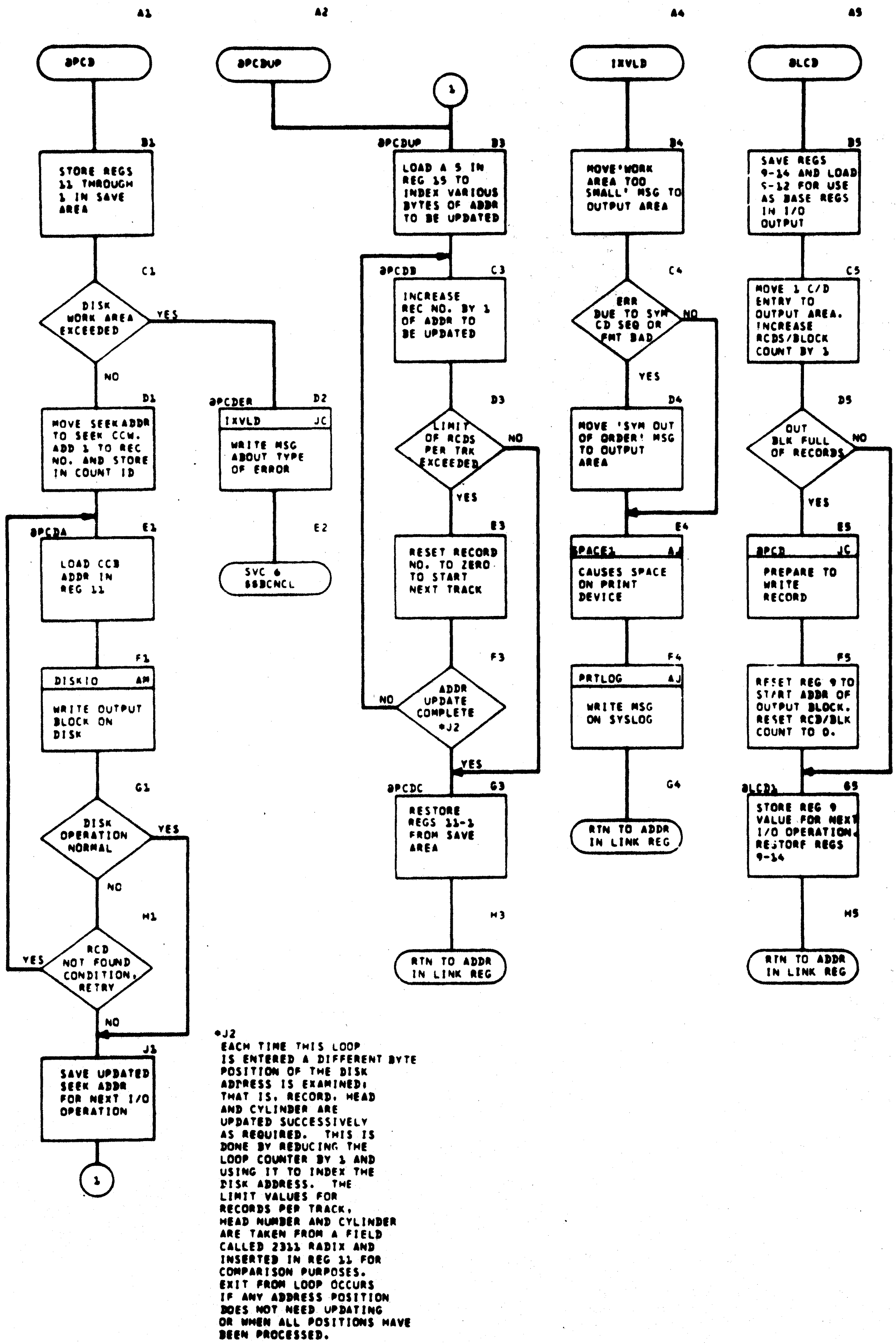
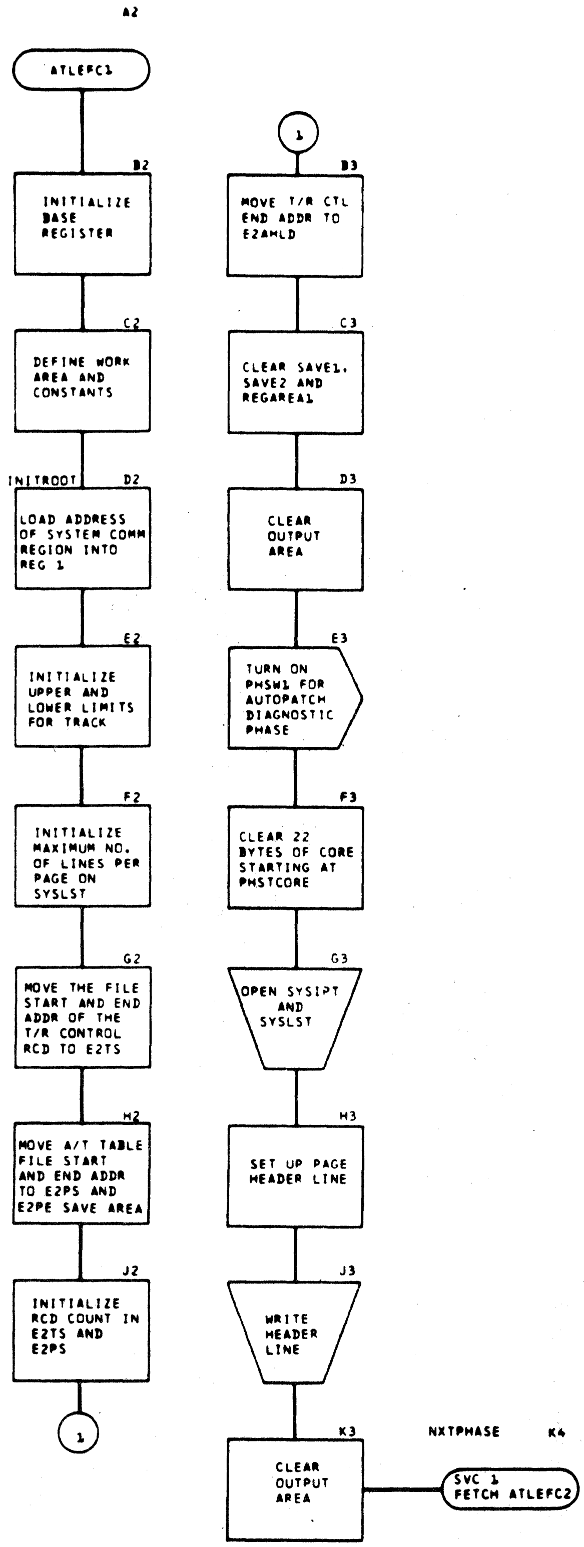
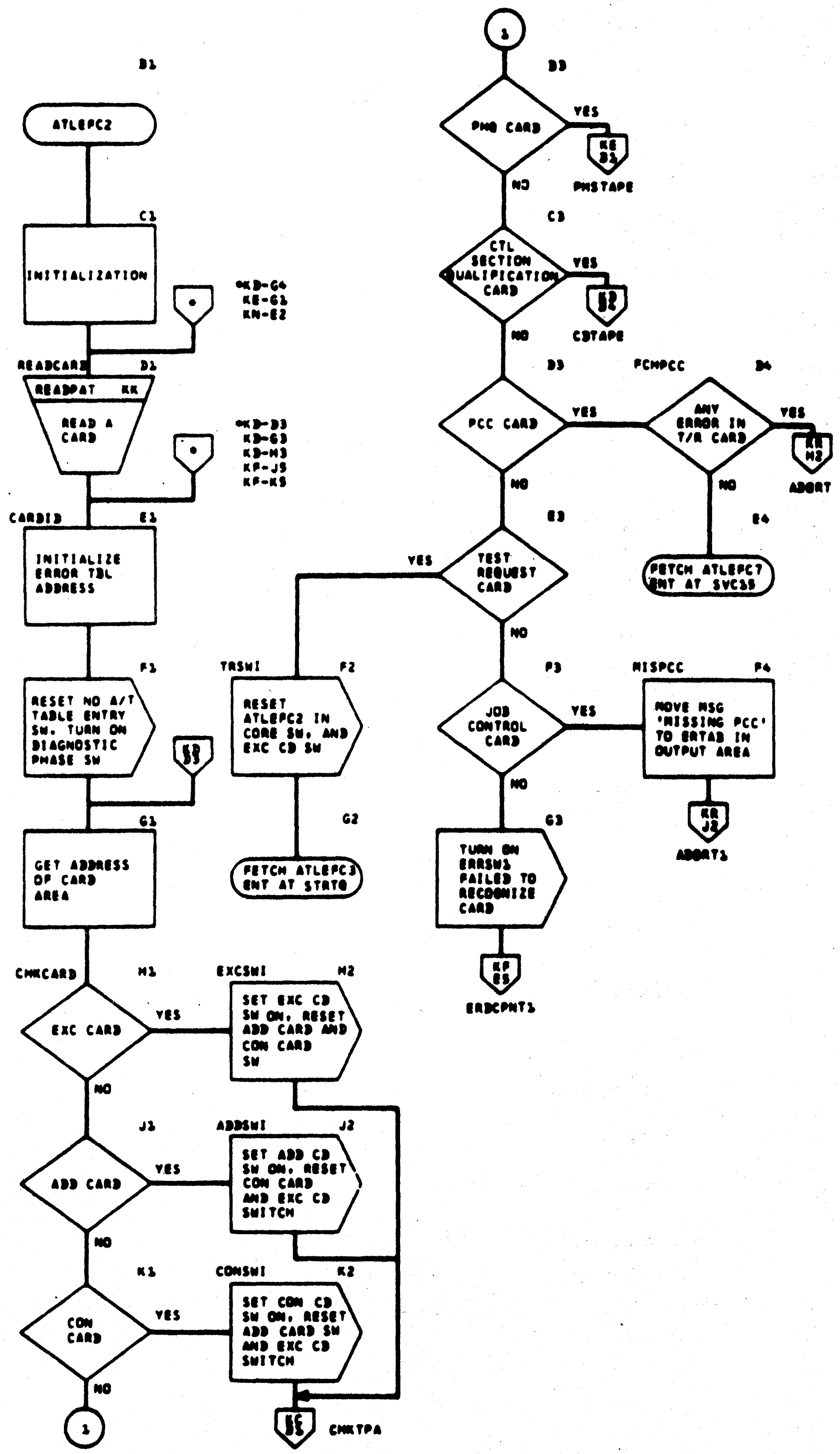
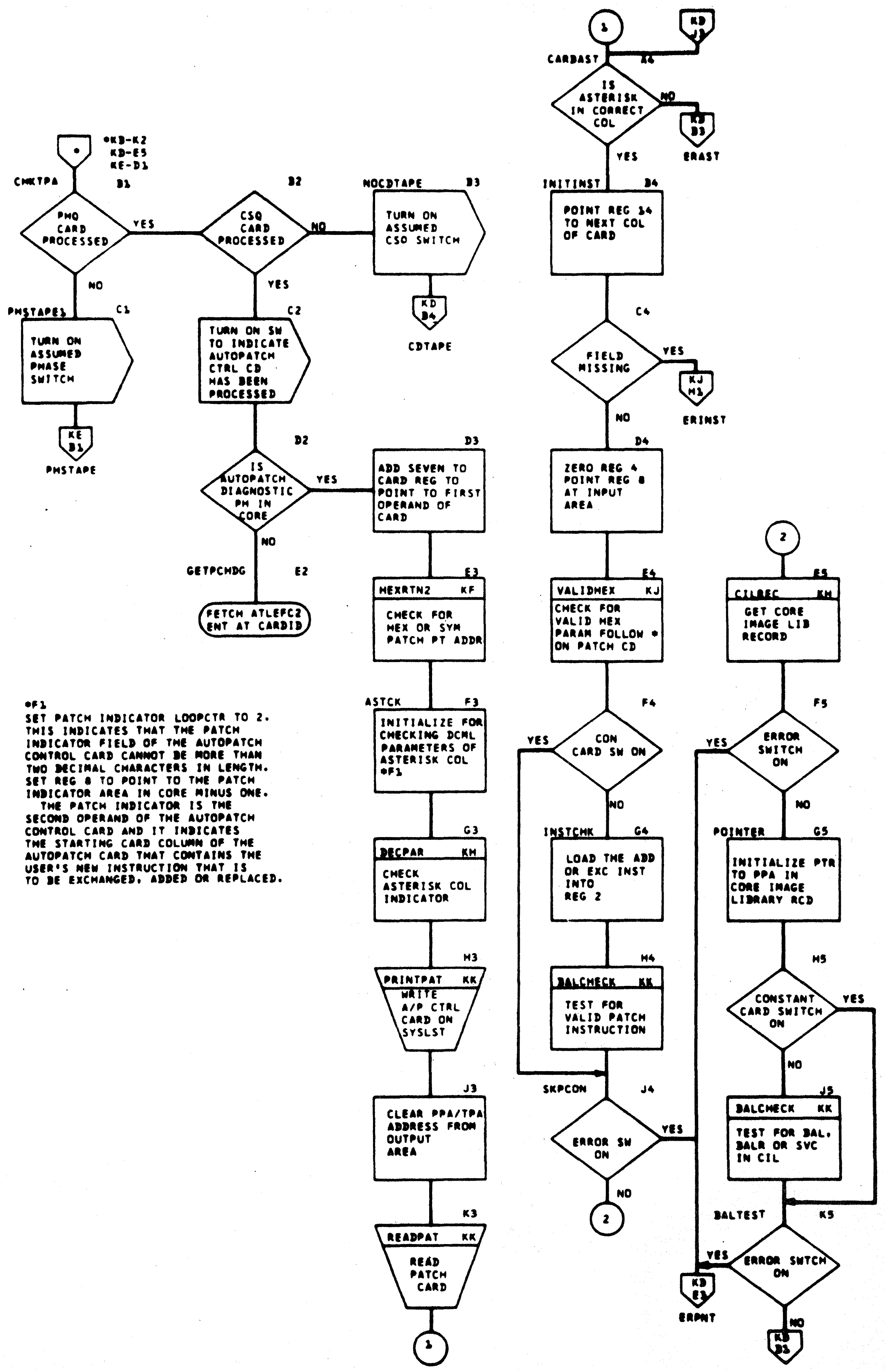
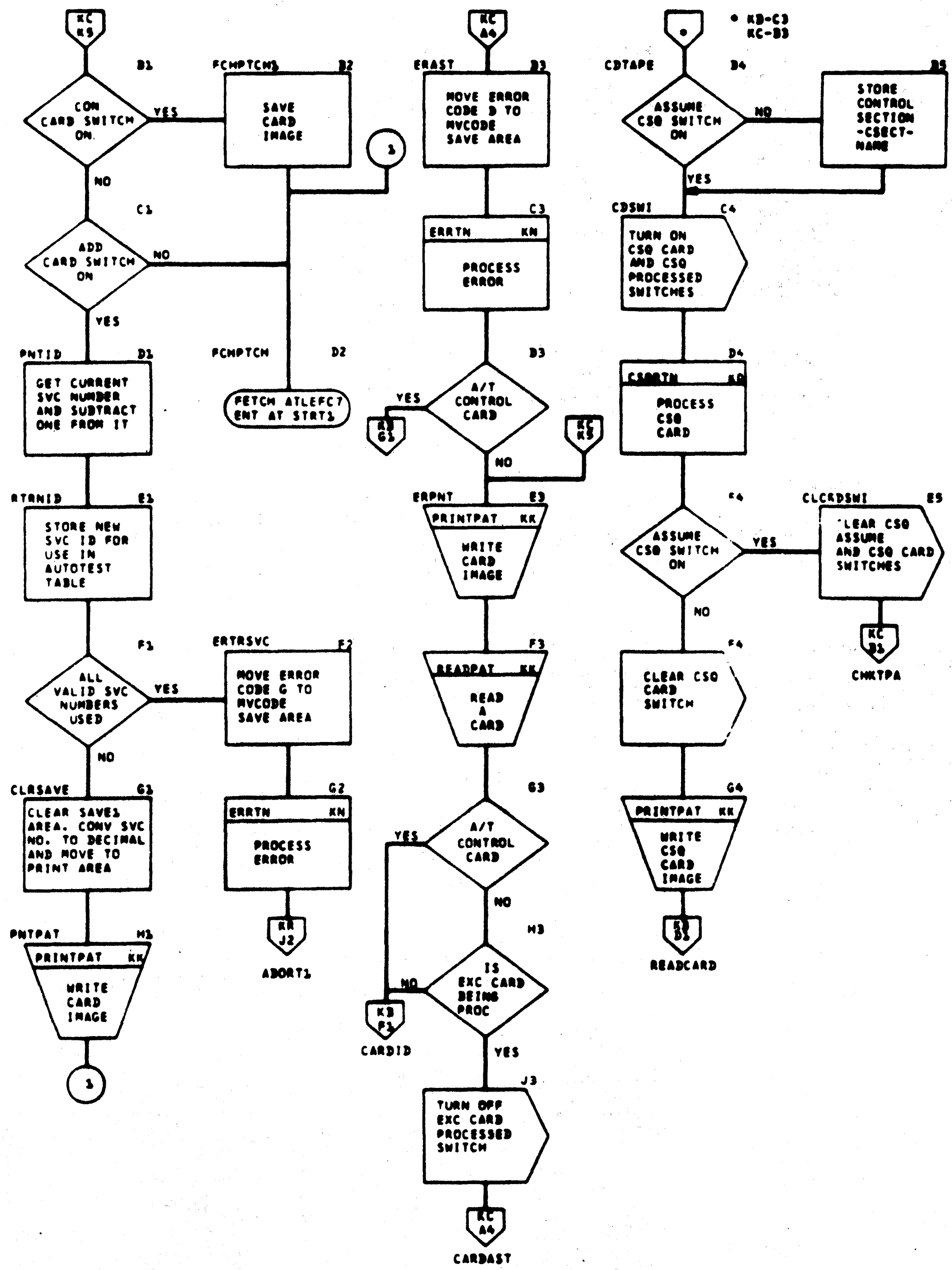


Chart KA. Initialization (ATLEFC1) (Refer to Chart 11)



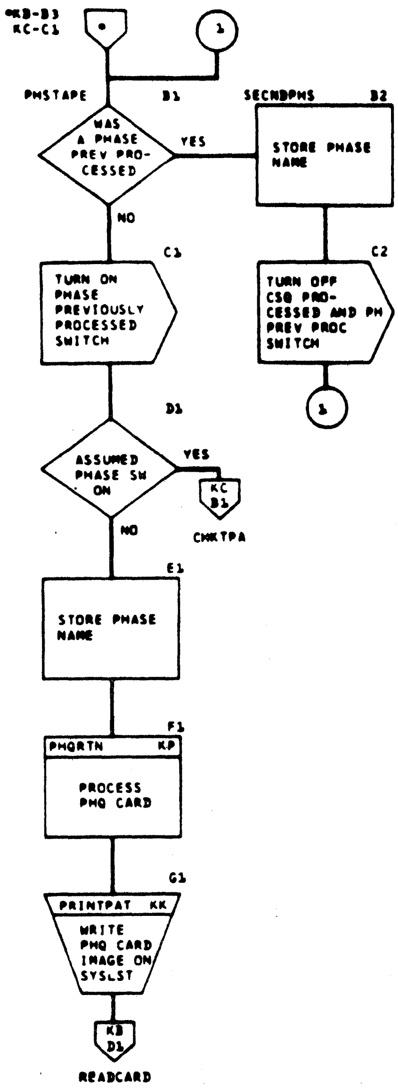

ATLEFC2
INITIALIZATION
READCARD
READPAT KK
READ A CARD
CARDID
INITIALIZE ERROR TBL ADDRESS
RESET NO A/T TABLE ENTRY SW. TURN ON DIAGNOSTIC PHASE SW
GET ADDRESS OF CARD AREA
CHKCARD
EXC CARD
ADD CARD
CON CARD
EXCSW1
SET EXC CD SW ON, RESET ADD CARD AND CON CARD SW
ADDSW1
SET ADD CD SW ON, RESET CON CARD AND EXC CD SWITCH
CONSW1
SET CON CD SW ON, RESET ADD CARD SW AND EXC CD SWITCH
CHKTPA
TRSW1
RESET ATLEFC2 IN CORE SW, AND EXC CD SW
FETCH ATLEFC3 ENT AT STRT0
PHS CARD
PHSTAPE
CTL SECTION QUALIFICATION CARD
CDTAPE
PCC CARD
FCHPCC
ANY ERROR IN T/R CARD
ABORT
FETCH ATLEFC7 ENT AT SVC55
TEST REQUEST CARD
JOB CONTROL CARD
MISPCC
MOVE MSG 'MISSING PCC' TO ERTAB IN OUTPUT AREA
ABORT1
TURN ON ERRSW1 FAILED TO RECOGNIZE CARD
ERDCPNT1

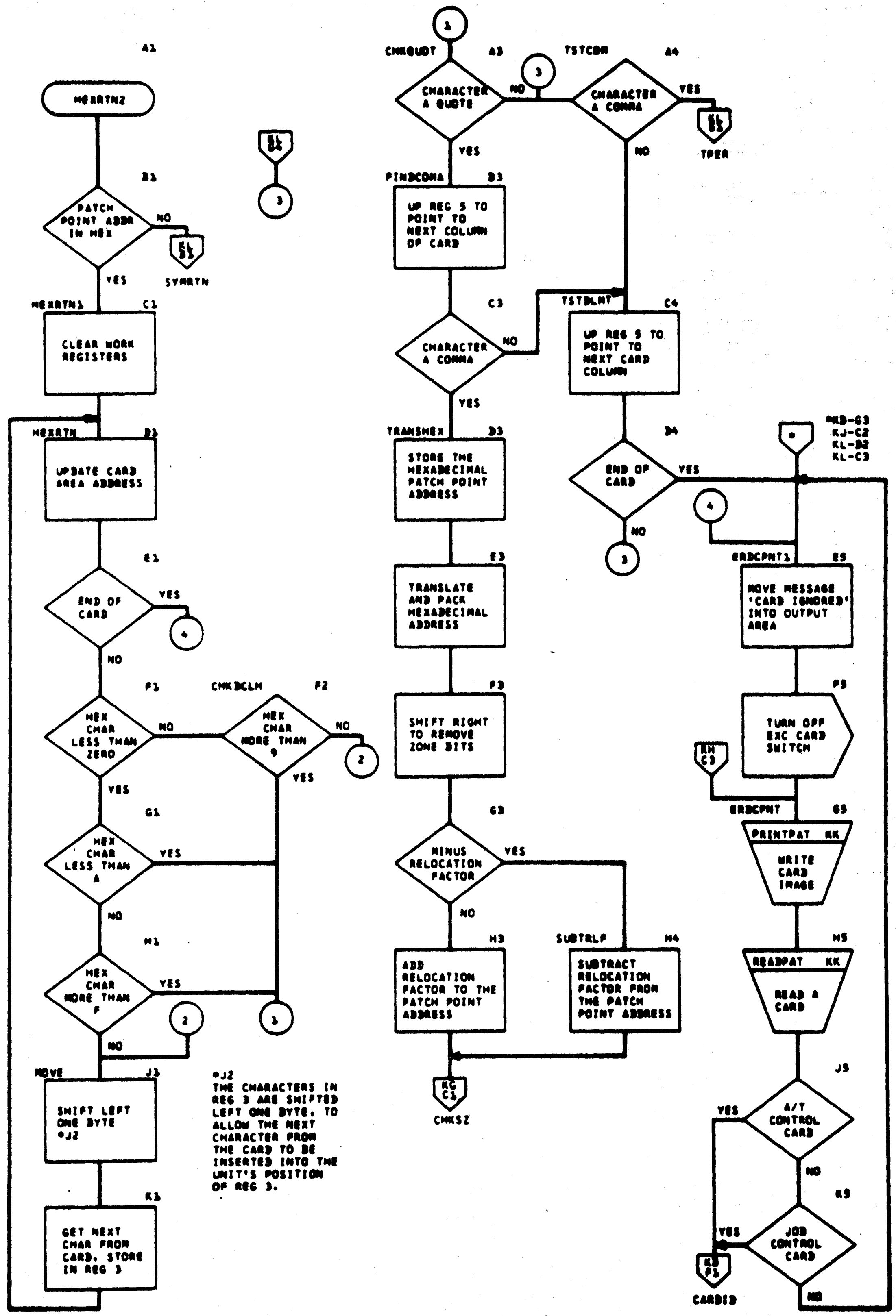

*KB-K2
KB-E5
KE-D1
CHKTPA B1
PHO CARD PROCESSED
YES
NO
PHSTAPE1 C1
TURN ON ASSUMED PHASE SWITCH
KE B1
PHSTAPE
B2
CSQ CARD PROCESSED
NO
YES
C2
TURN ON SW TO INDICATE AUTOPATCH CTRL CD HAS BEEN PROCESSED
D2
IS AUTOPATCH DIAGNOSTIC PH IN CORE
YES
NO
GETPCHDG E2
FETCH ATLEFC2 ENT AT CARDID
NOCDTAPE B3
TURN ON ASSUMED CSO SWITCH
KD B4
CDTAPE
D3
ADD SEVEN TO CARD REG TO POINT TO FIRST OPERAND OF CARD
E3
HEXRTN2 KF
CHECK FOR HEX OR SYM PATCH PT ADDR
ASTCK F3
INITIALIZE FOR CHECKING DCML PARAMETERS OF ASTERISK COL *F1
G3
DECPAR KH
CHECK ASTERISK COL INDICATOR
H3
PRINTPAT KK
WRITE A/P CTRL CARD ON SYSLST
J3
CLEAR PPA/TPA ADDRESS FROM OUTPUT AREA
K3
READPAT KK
READ PATCH CARD
1
*F1
SET PATCH INDICATOR LOOPCTR TO 2. THIS INDICATES THAT THE PATCH INDICATOR FIELD OF THE AUTOPATCH CONTROL CARD CANNOT BE MORE THAN TWO DECIMAL CHARACTERS IN LENGTH. SET REG 8 TO POINT TO THE PATCH INDICATOR AREA IN CORE MINUS ONE.
THE PATCH INDICATOR IS THE SECOND OPERAND OF THE AUTOPATCH CONTROL CARD AND IT INDICATES THE STARTING CARD COLUMN OF THE AUTOPATCH CARD THAT CONTAINS THE USER'S NEW INSTRUCTION THAT IS TO BE EXCHANGED, ADDED OR REPLACED.
1
KD J3
CARDAST A4
IS ASTERISK IN CORRECT COL
NO
KD B3
ERAST
YES
INITINST B4
POINT REG 14 TO NEXT COL OF CARD
C4
FIELD MISSING
YES
KJ H1
ERINST
NO
D4
ZERO REG 4 POINT REG 8 AT INPUT AREA
E4
VALIDHEX KJ
CHECK FOR VALID HEX PARAM FOLLOW * ON PATCH CD
F4
CON CARD SW ON
YES
NO
INSTCHK G4
LOAD THE ADD OR EXC INST INTO REG 2
H4
BALCHECK KK
TEST FOR VALID PATCH INSTRUCTION
SKPCON J4
ERROR SW ON
YES
NO
2
2
E5
CILREC KH
GET CORE IMAGE LIB RECORD
F5
ERROR SWITCH ON
YES
NO
POINTER G5
INITIALIZE PTR TO PPA IN CORE IMAGE LIBRARY RCD
H5
CONSTANT CARD SWITCH ON
YES
NO
J5
BALCHECK KK
TEST FOR BAL, BALR OR SVC IN CIL
BALTEST K5
ERROR SWTCH ON
YES
KD E1
ERPNT
NO
KD B1

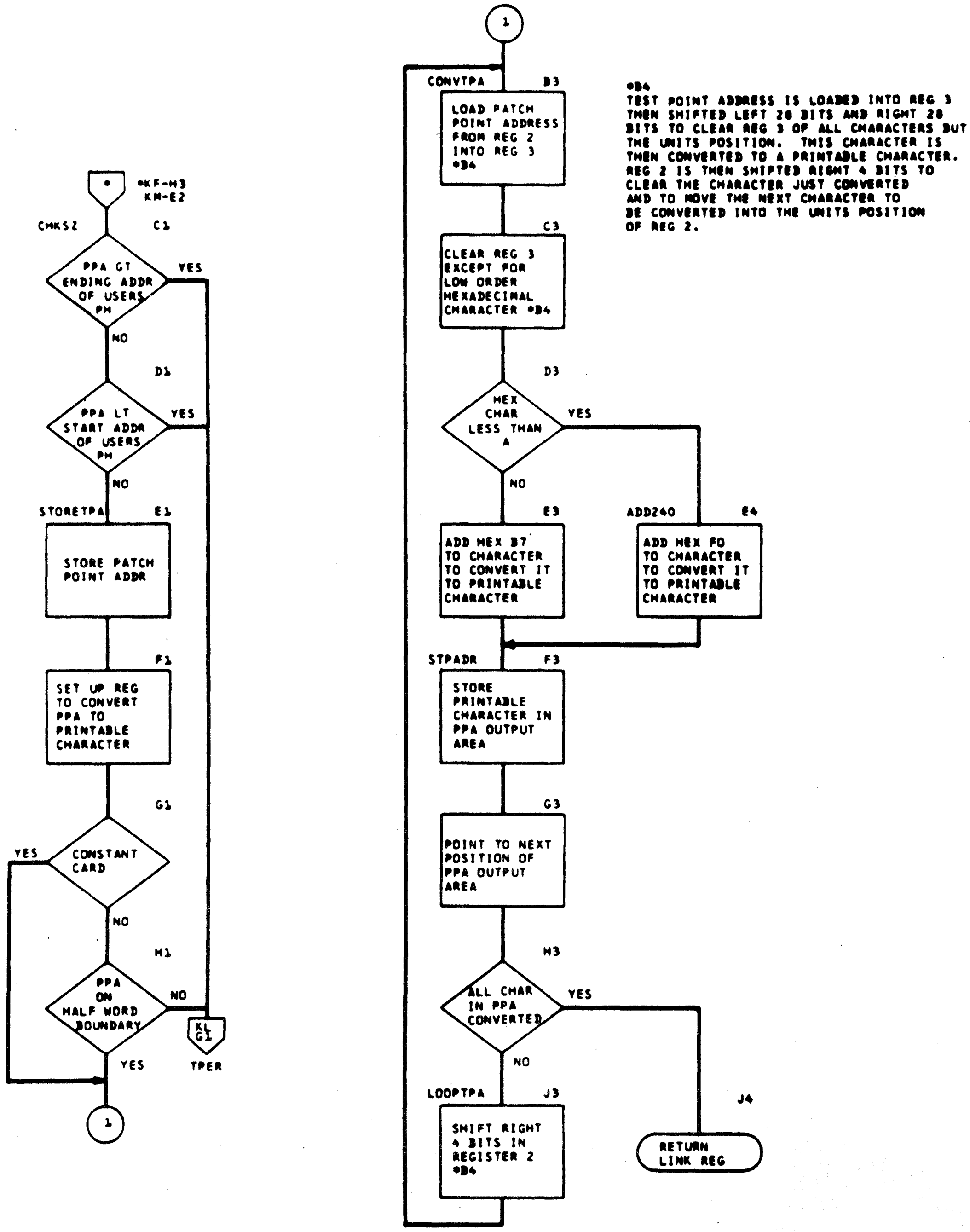


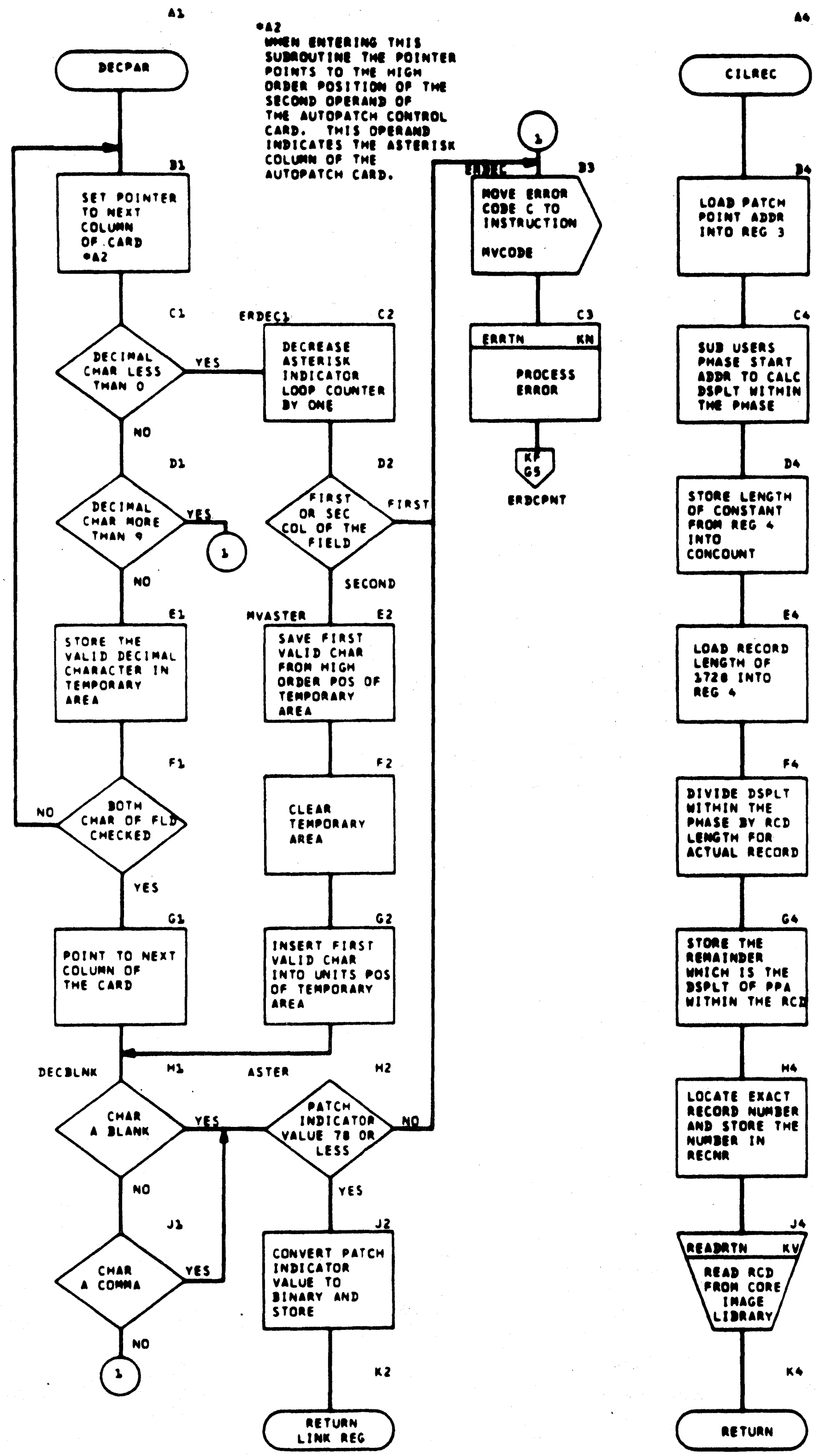
KC K5
B1
CON CARD SWITCH ON
YES
FCHPTCH1
B2
SAVE CARD IMAGE
1
NO
C1
ADD CARD SWITCH ON
NO
YES
PNTID
D1
GET CURRENT SVC NUMBER AND SUBTRACT ONE FROM IT
FCHPTCH
D2
FETCH ATLEFC7 ENT AT STRT1
RTRNID
E1
STORE NEW SVC ID FOR USE IN AUTOTEST TABLE
F1
ALL VALID SVC NUMBERS USED
YES
ERTRSVC
F2
MOVE ERROR CODE G TO MVCODE SAVE AREA
NO
CLRSAVE
G1
CLEAR SAVE1 AREA. CONV SVC NO. TO DECIMAL AND MOVE TO PRINT AREA
ERRTN KN
G2
PROCESS ERROR
KR J2
ABORT1
PNTPAT
H1
PRINTPAT KK
WRITE CARD IMAGE
1
KC A4
ERAST
B3
MOVE ERROR CODE D TO MVCODE SAVE AREA
C3
ERRTN KN
PROCESS ERROR
D3
A/T CONTROL CARD
YES
KB G1
KC K5
NO
ERPNT
E3
PRINTPAT KK
WRITE CARD IMAGE
F3
READPAT KK
READ A CARD
G3
A/T CONTROL CARD
YES
NO
H3
IS EXC CARD BEING PROC
NO
KB F1
CARDID
YES
J3
TURN OFF EXC CARD PROCESSED SWITCH
KC A4
CARDAST
* KB-C3 KC-B3
CDTAPE
B4
ASSUME CSQ SWITCH ON
NO
B5
STORE CONTROL SECTION -CSECT-NAME
YES
CDSWI
C4
TURN ON CSQ CARD AND CSQ PROCESSED SWITCHES
D4
CSQRTN KP
PROCESS CSQ CARD
E4
ASSUME CSQ SWITCH ON
YES
CLCRDSWI
E5
CLEAR CSQ ASSUME AND CSQ CARD SWITCHES
KC B1
CHKTPA
NO
F4
CLEAR CSQ CARD SWITCH
G4
PRINTPAT KK
WRITE CSQ CARD IMAGE
KB B1
READCARD

Chart KE. Autopatch Diagnostics (ATLEFC2) (Refer to Chart 11) (Part 4 of 19)




HEXRTN2
PATCH POINT ADDR IN HEX
SYMRTN
HEXRTN1
CLEAR WORK REGISTERS
HEXRTN
UPDATE CARD AREA ADDRESS
END OF CARD
HEX CHAR LESS THAN ZERO
CHKDCLM
HEX CHAR MORE THAN 9
HEX CHAR LESS THAN A
HEX CHAR MORE THAN F
MOVE
SHIFT LEFT ONE BYTE *J2
GET NEXT CHAR FROM CARD. STORE IN REG 3
*J2 THE CHARACTERS IN REG 3 ARE SHIFTED LEFT ONE BYTE, TO ALLOW THE NEXT CHARACTER FROM THE CARD TO BE INSERTED INTO THE UNIT'S POSITION OF REG 3.
CHKQUOT
CHARACTER A QUOTE
FINDCOMA
UP REG 5 TO POINT TO NEXT COLUMN OF CARD
CHARACTER A COMMA
TRANSHEX
STORE THE HEXADECIMAL PATCH POINT ADDRESS
TRANSLATE AND PACK HEXADECIMAL ADDRESS
SHIFT RIGHT TO REMOVE ZONE BITS
MINUS RELOCATION FACTOR
ADD RELOCATION FACTOR TO THE PATCH POINT ADDRESS
SUBTRLF
SUBTRACT RELOCATION FACTOR FROM THE PATCH POINT ADDRESS
CHKSZ
TSTCOM
CHARACTER A COMMA
TPER
TSTBLNT
UP REG 5 TO POINT TO NEXT CARD COLUMN
END OF CARD
*KB-G3 KJ-C2 KL-B2 KL-C3
ERDCPNT1
MOVE MESSAGE 'CARD IGNORED' INTO OUTPUT AREA
TURN OFF EXC CARD SWITCH
ERDCPNT
PRINTPAT KK
WRITE CARD IMAGE
READPAT KK
READ A CARD
A/T CONTROL CARD
JOB CONTROL CARD
CARDID


*KF-H3
KM-E2
CHKSZ C1
PPA GT ENDING ADDR OF USERS PH
YES
NO
D1
PPA LT START ADDR OF USERS PH
YES
NO
STORETPA E1
STORE PATCH POINT ADDR
F1
SET UP REG TO CONVERT PPA TO PRINTABLE CHARACTER
G1
CONSTANT CARD
YES
NO
H1
PPA ON HALF WORD BOUNDARY
NO
YES
KL G1
TPER
1
CONVTPA B3
LOAD PATCH POINT ADDRESS FROM REG 2 INTO REG 3 *B4
C3
CLEAR REG 3 EXCEPT FOR LOW ORDER HEXADECIMAL CHARACTER *B4
D3
HEX CHAR LESS THAN A
YES
NO
E3
ADD HEX B7 TO CHARACTER TO CONVERT IT TO PRINTABLE CHARACTER
ADD240 E4
ADD HEX F0 TO CHARACTER TO CONVERT IT TO PRINTABLE CHARACTER
STPADR F3
STORE PRINTABLE CHARACTER IN PPA OUTPUT AREA
G3
POINT TO NEXT POSITION OF PPA OUTPUT AREA
H3
ALL CHAR IN PPA CONVERTED
YES
NO
LOOPTPA J3
SHIFT RIGHT 4 BITS IN REGISTER 2 *B4
J4
RETURN LINK REG
*B4
TEST POINT ADDRESS IS LOADED INTO REG 3 THEN SHIFTED LEFT 28 BITS AND RIGHT 28 BITS TO CLEAR REG 3 OF ALL CHARACTERS BUT THE UNITS POSITION. THIS CHARACTER IS THEN CONVERTED TO A PRINTABLE CHARACTER. REG 2 IS THEN SHIFTED RIGHT 4 BITS TO CLEAR THE CHARACTER JUST CONVERTED AND TO MOVE THE NEXT CHARACTER TO BE CONVERTED INTO THE UNITS POSITION OF REG 2.



A1
DECPAR
*A2
WHEN ENTERING THIS SUBROUTINE THE POINTER POINTS TO THE HIGH ORDER POSITION OF THE SECOND OPERAND OF THE AUTOPATCH CONTROL CARD. THIS OPERAND INDICATES THE ASTERISK COLUMN OF THE AUTOPATCH CARD.
B1
SET POINTER TO NEXT COLUMN OF CARD *A2
C1
DECIMAL CHAR LESS THAN 0
YES
NO
ERDEC1
C2
DECREASE ASTERISK INDICATOR LOOP COUNTER BY ONE
D1
DECIMAL CHAR MORE THAN 9
YES
NO
1
D2
FIRST OR SEC COL OF THE FIELD
FIRST
SECOND
E1
STORE THE VALID DECIMAL CHARACTER IN TEMPORARY AREA
MVASTER
E2
SAVE FIRST VALID CHAR FROM HIGH ORDER POS OF TEMPORARY AREA
F1
BOTH CHAR OF FLD CHECKED
NO
YES
F2
CLEAR TEMPORARY AREA
G1
POINT TO NEXT COLUMN OF THE CARD
G2
INSERT FIRST VALID CHAR INTO UNITS POS OF TEMPORARY AREA
DECBLNK
H1
CHAR A BLANK
YES
NO
ASTER
H2
PATCH INDICATOR VALUE 78 OR LESS
NO
YES
J1
CHAR A COMMA
YES
NO
1
J2
CONVERT PATCH INDICATOR VALUE TO BINARY AND STORE
K2
RETURN LINK REG
1
ERDEC
B3
MOVE ERROR CODE C TO INSTRUCTION
MVCODE
C3
ERRTN
KN
PROCESS ERROR
KF
G5
ERBCPNT
A4
CILREC
B4
LOAD PATCH POINT ADDR INTO REG 3
C4
SUB USERS PHASE START ADDR TO CALC DSPLT WITHIN THE PHASE
D4
STORE LENGTH OF CONSTANT FROM REG 4 INTO CONCOUNT
E4
LOAD RECORD LENGTH OF 1728 INTO REG 4
F4
DIVIDE DSPLT WITHIN THE PHASE BY RCD LENGTH FOR ACTUAL RECORD
G4
STORE THE REMAINDER WHICH IS THE DSPLT OF PPA WITHIN THE RCD
H4
LOCATE EXACT RECORD NUMBER AND STORE THE NUMBER IN RECNR
J4
READRTN
KV
READ RCD FROM CORE IMAGE LIBRARY
K4
RETURN

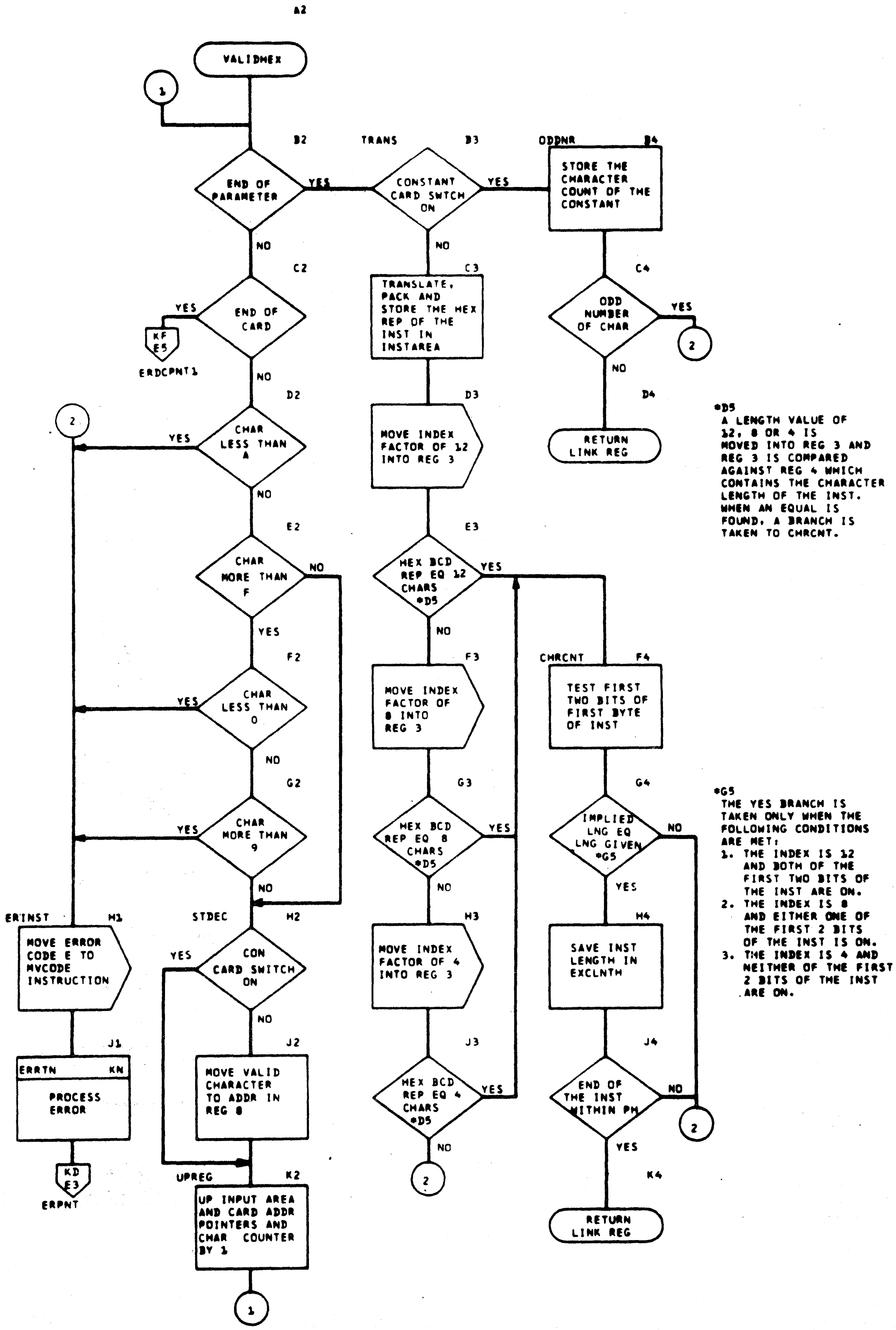
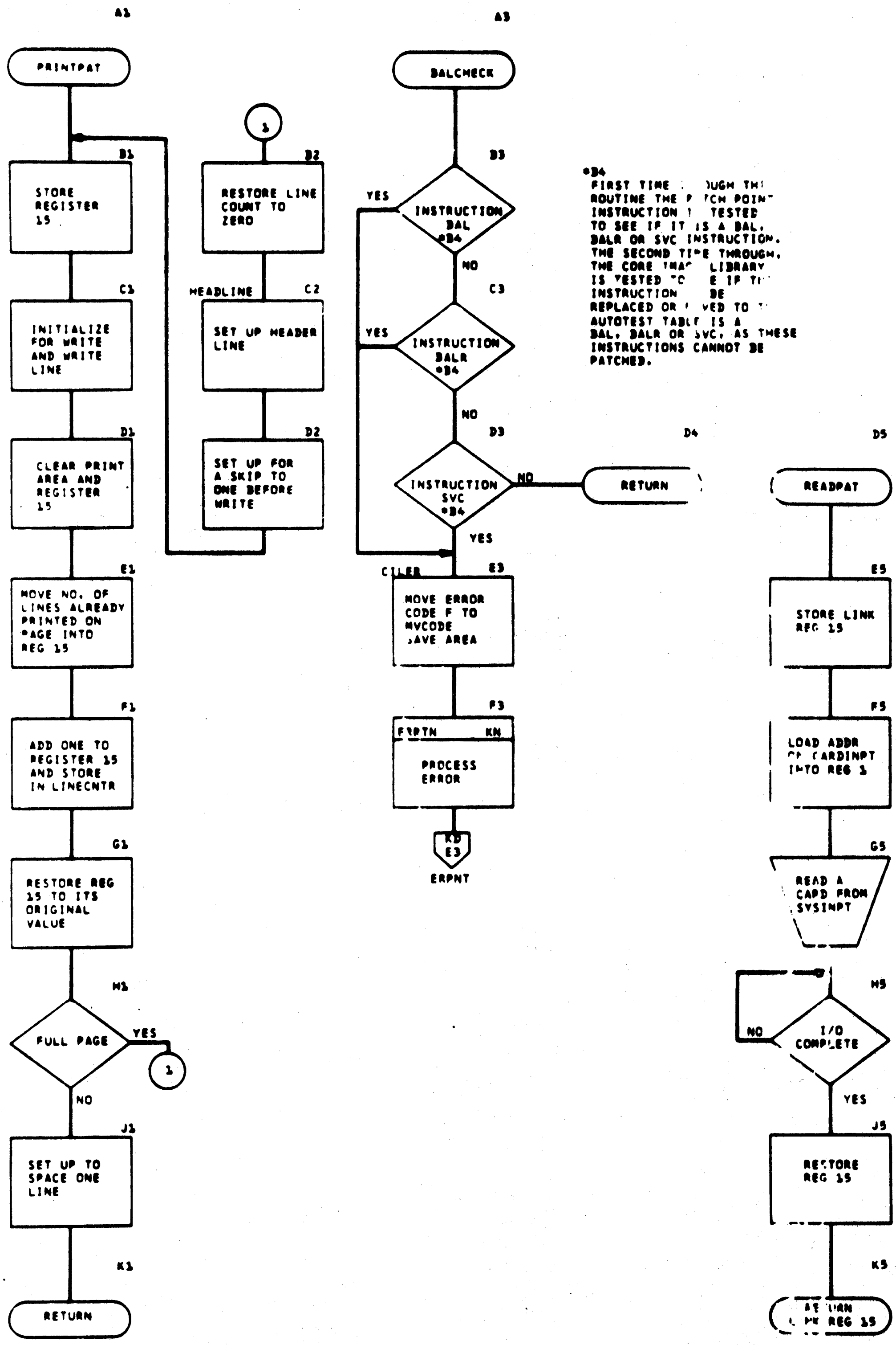
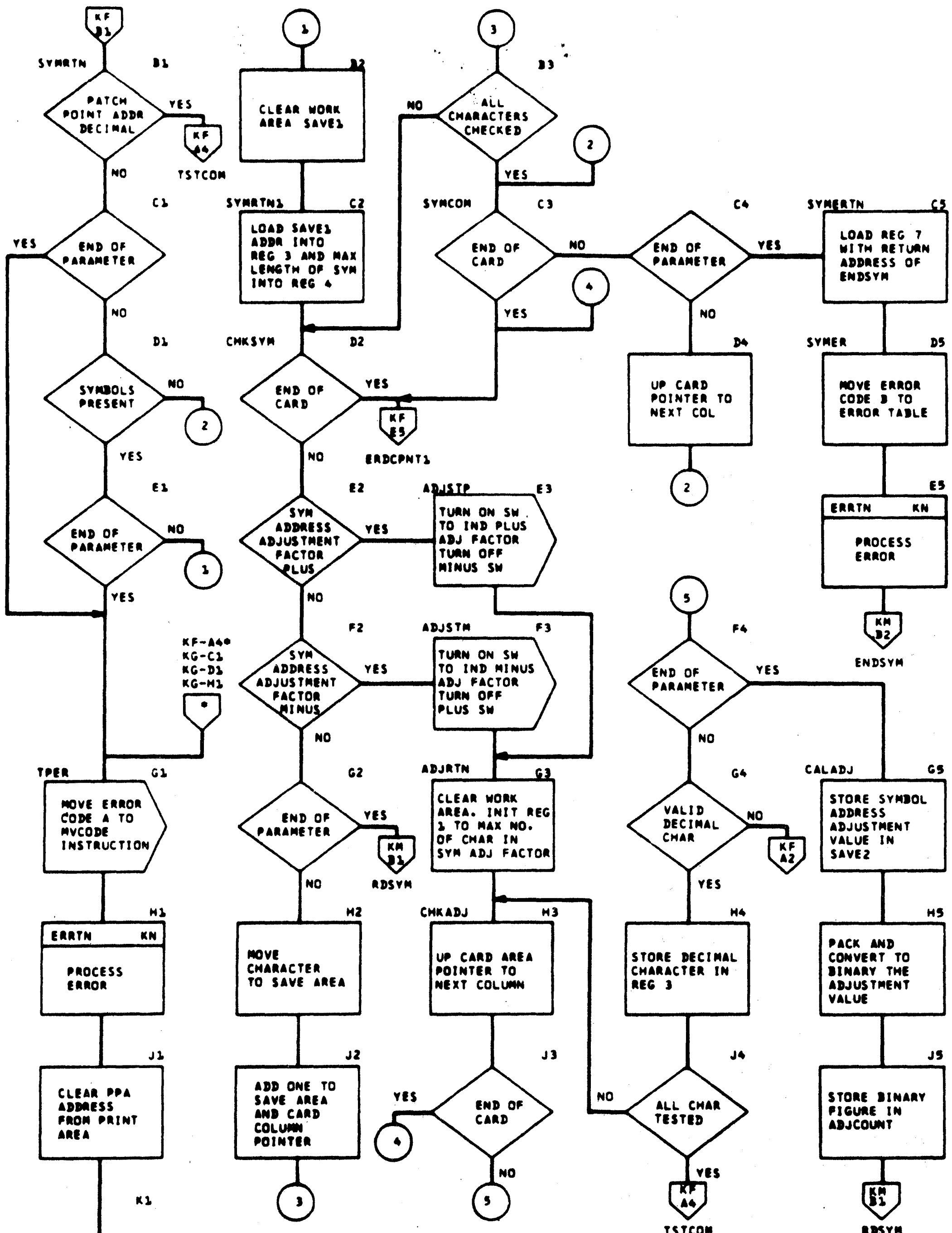

A2
VALIDHEX
1
B2
END OF PARAMETER
YES
NO
TRANS
B3
CONSTANT CARD SWTCH ON
YES
NO
ODDNR
B4
STORE THE CHARACTER COUNT OF THE CONSTANT
C2
END OF CARD
YES
KF E5
ERDCPNT1
C3
TRANSLATE, PACK AND STORE THE HEX REP OF THE INST IN INSTAREA
C4
ODD NUMBER OF CHAR
YES
2
NO
D4
RETURN LINK REG
D2
CHAR LESS THAN A
D3
MOVE INDEX FACTOR OF 12 INTO REG 3
*D5
A LENGTH VALUE OF 12, 8 OR 4 IS MOVED INTO REG 3 AND REG 3 IS COMPARED AGAINST REG 4 WHICH CONTAINS THE CHARACTER LENGTH OF THE INST. WHEN AN EQUAL IS FOUND, A BRANCH IS TAKEN TO CHRCNT.
E2
CHAR MORE THAN F
E3
HEX BCD REP EQ 12 CHARS *D5
F2
CHAR LESS THAN 0
F3
MOVE INDEX FACTOR OF 8 INTO REG 3
CHRCNT
F4
TEST FIRST TWO BITS OF FIRST BYTE OF INST
G2
CHAR MORE THAN 9
G3
HEX BCD REP EQ 8 CHARS *D5
G4
IMPLIED LNG EQ LNG GIVEN *G5
*G5
THE YES BRANCH IS TAKEN ONLY WHEN THE FOLLOWING CONDITIONS ARE MET:
1. THE INDEX IS 12 AND BOTH OF THE FIRST TWO BITS OF THE INST ARE ON.
2. THE INDEX IS 8 AND EITHER ONE OF THE FIRST 2 BITS OF THE INST IS ON.
3. THE INDEX IS 4 AND NEITHER OF THE FIRST 2 BITS OF THE INST ARE ON.
ERINST
H1
MOVE ERROR CODE E TO MVCODE INSTRUCTION
STDEC
H2
CON CARD SWITCH ON
H3
MOVE INDEX FACTOR OF 4 INTO REG 3
H4
SAVE INST LENGTH IN EXCLNTH
J1
ERRTN
KN
PROCESS ERROR
KD E3
ERPNT
J2
MOVE VALID CHARACTER TO ADDR IN REG 8
J3
HEX BCD REP EQ 4 CHARS *D5
J4
END OF THE INST WITHIN PM
UPREG
K2
UP INPUT AREA AND CARD ADDR POINTERS AND CHAR COUNTER BY 1
K4
RETURN LINK REG

Chart KK. Autopatch Diagnostics (ATLEFC2) (Refer to Chart 11) (Part 9 of 19)





SYMRTN
PATCH POINT ADDR DECIMAL
TSTCOM
END OF PARAMETER
SYMBOLS PRESENT
END OF PARAMETER
KF-A4*
KG-C1
KG-D1
KG-H1
TPER
MOVE ERROR CODE A TO MVCODE INSTRUCTION
ERRTN KN
PROCESS ERROR
CLEAR PPA ADDRESS FROM PRINT AREA
RETURN TO LINK REG ADDR
CLEAR WORK AREA SAVE1
SYMRTN1
LOAD SAVE1 ADDR INTO REG 3 AND MAX LENGTH OF SYM INTO REG 4
CHKSYM
END OF CARD
ERDCPNT1
SYM ADDRESS ADJUSTMENT FACTOR PLUS
SYM ADDRESS ADJUSTMENT FACTOR MINUS
END OF PARAMETER
RDSYM
MOVE CHARACTER TO SAVE AREA
ADD ONE TO SAVE AREA AND CARD COLUMN POINTER
ALL CHARACTERS CHECKED
SYMCOM
END OF CARD
ADJSTP
TURN ON SW TO IND PLUS ADJ FACTOR TURN OFF MINUS SW
ADJSTM
TURN ON SW TO IND MINUS ADJ FACTOR TURN OFF PLUS SW
ADJRTN
CLEAR WORK AREA. INIT REG 1 TO MAX NO. OF CHAR IN SYM ADJ FACTOR
CHKADJ
UP CARD AREA POINTER TO NEXT COLUMN
END OF CARD
END OF PARAMETER
UP CARD POINTER TO NEXT COL
END OF PARAMETER
VALID DECIMAL CHAR
STORE DECIMAL CHARACTER IN REG 3
ALL CHAR TESTED
TSTCOM
SYMERTN
LOAD REG 7 WITH RETURN ADDRESS OF ENDSYM
SYMER
MOVE ERROR CODE B TO ERROR TABLE
ERRTN KN
PROCESS ERROR
ENDSYM
CALADJ
STORE SYMBOL ADDRESS ADJUSTMENT VALUE IN SAVE2
PACK AND CONVERT TO BINARY THE ADJUSTMENT VALUE
STORE BINARY FIGURE IN ADJCOUNT
RDSYM

Chart KM. Autopatch Diagnostics (ATLEFC2) (Refer to Chart 11) (Part 11 of 19)

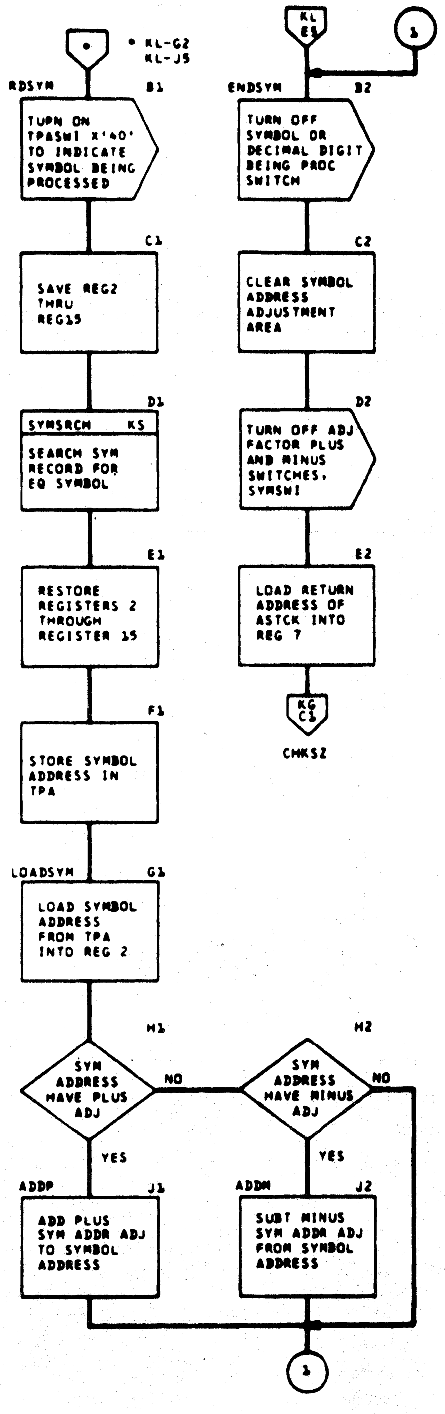


Chart KN. Autopatch Diagnostics (ATLEFC2) (Refer to Chart 11) (Part 12 of 19)

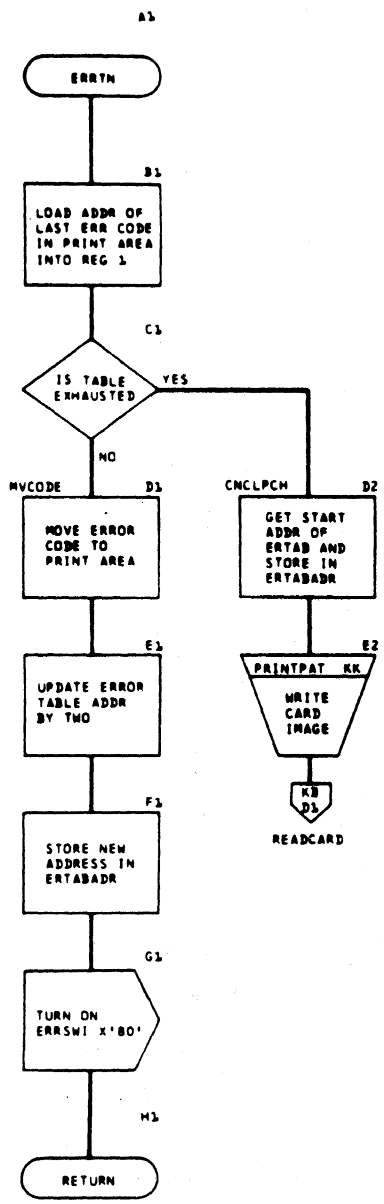
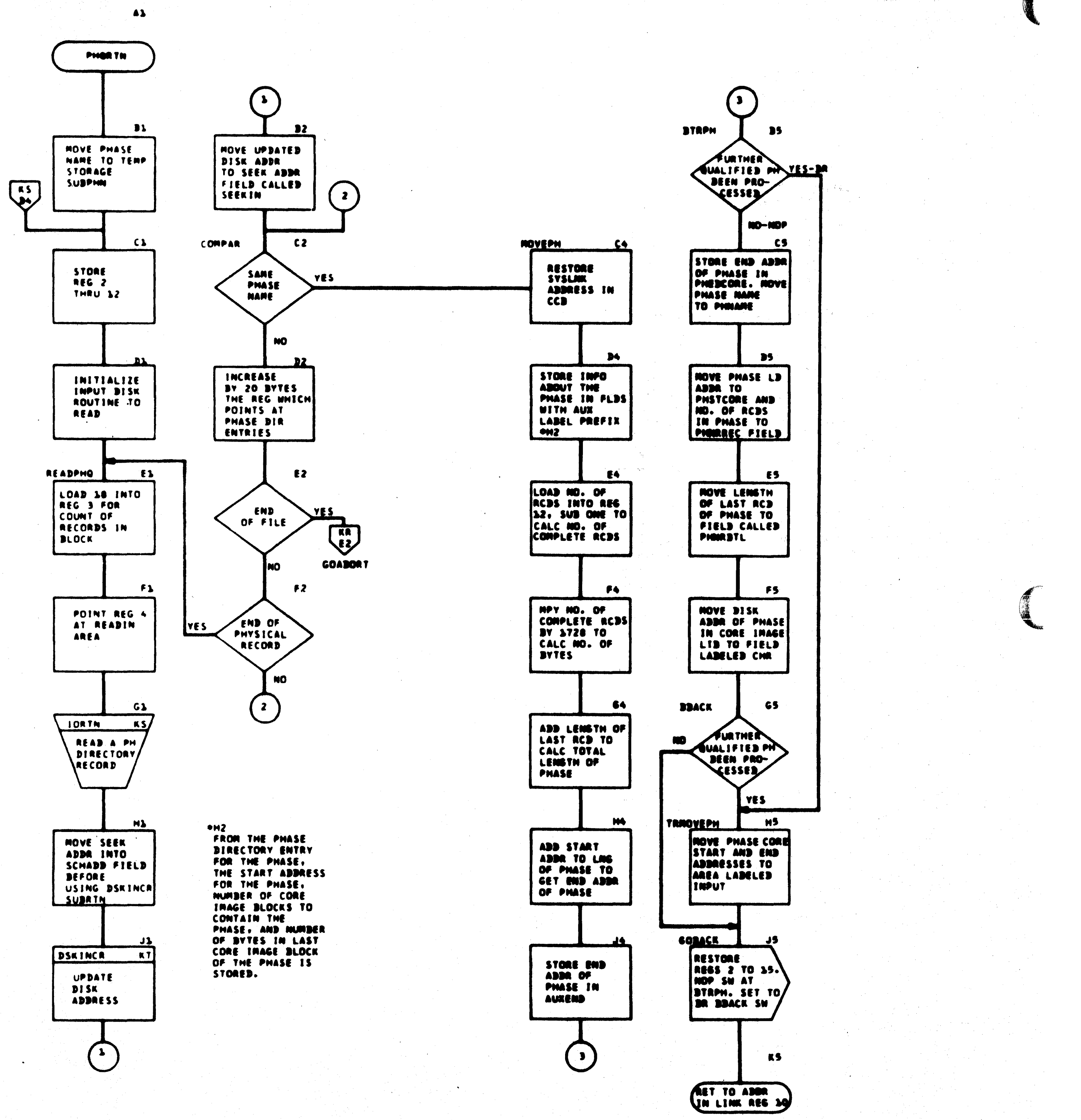
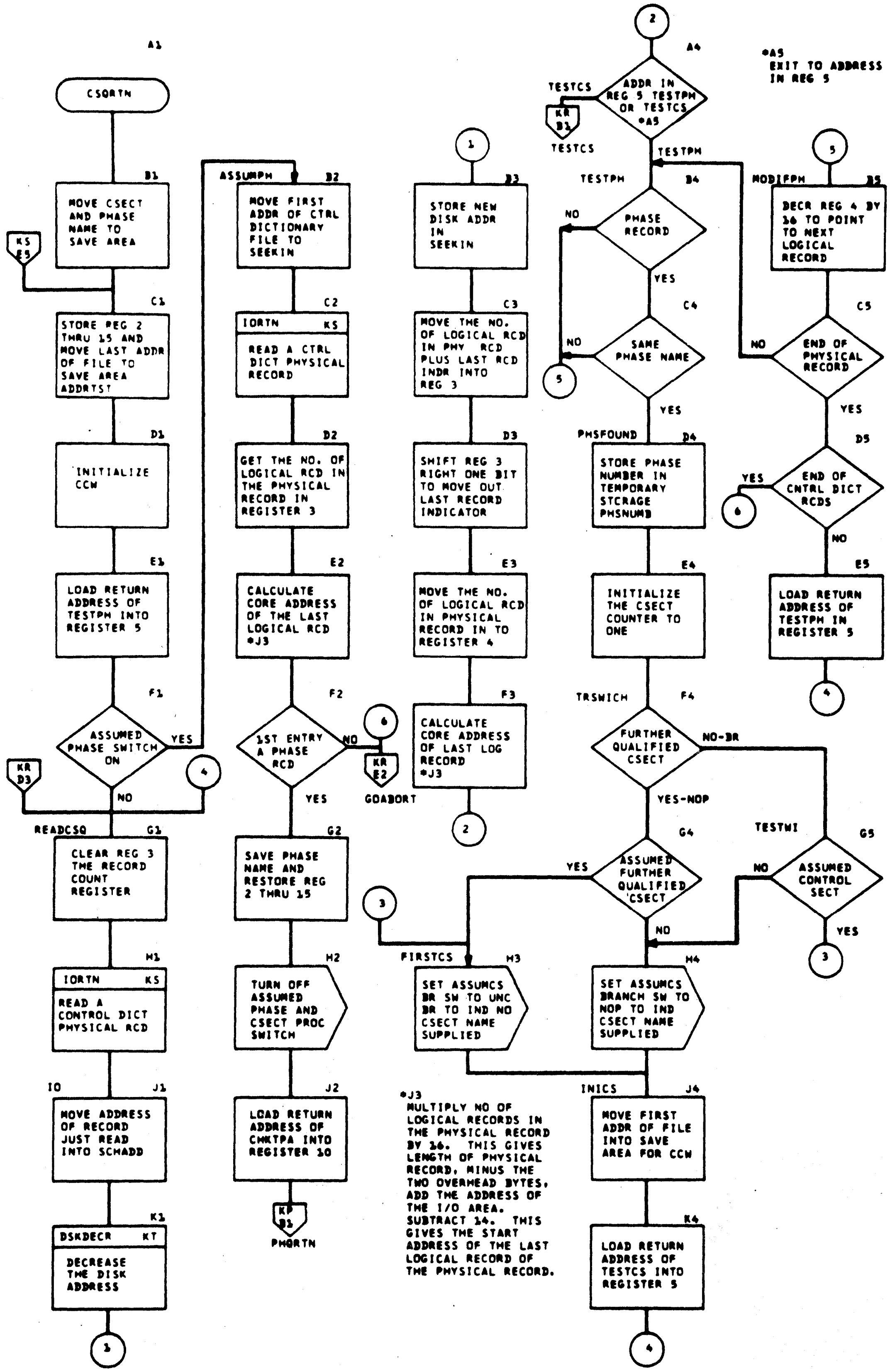
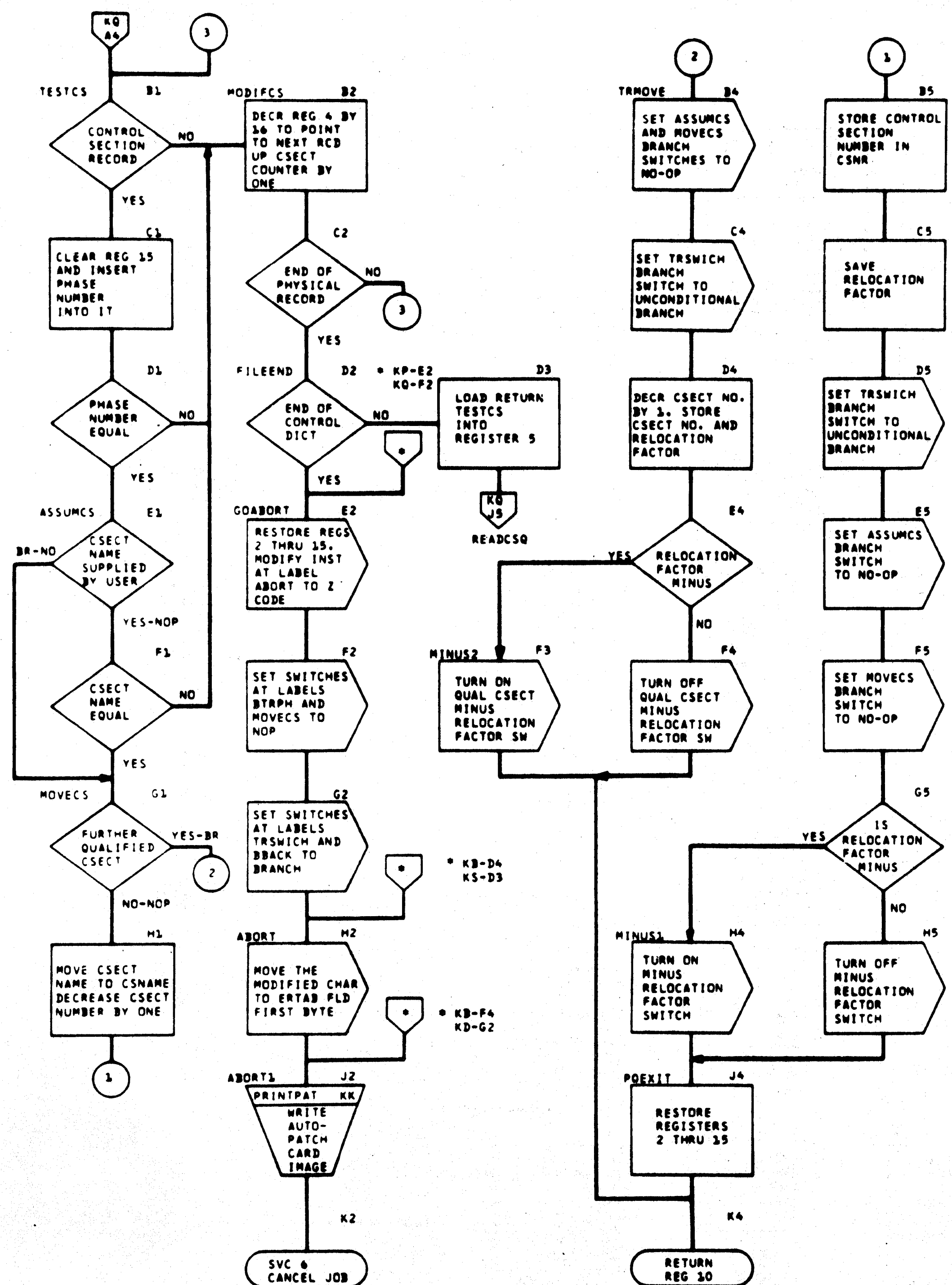
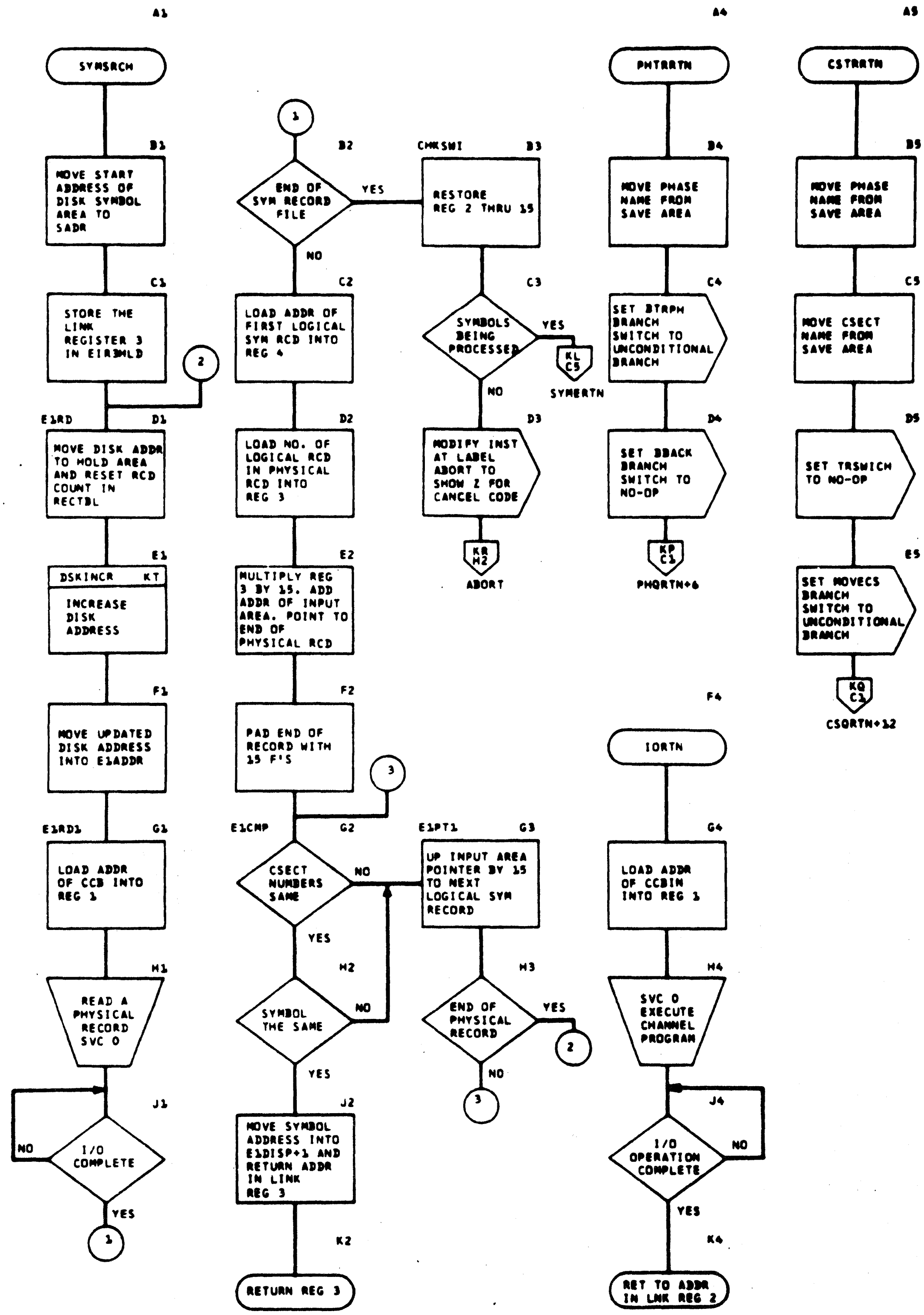


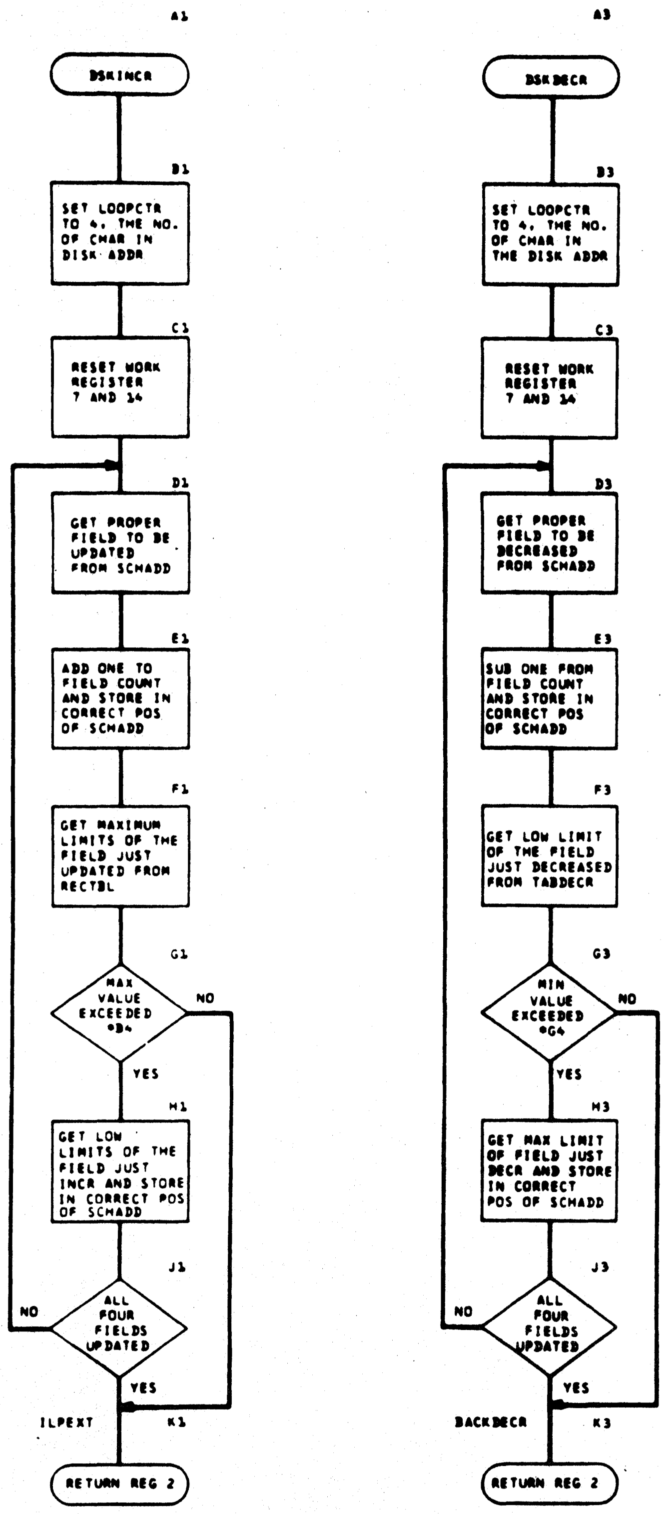
Chart KP. Autopatch Diagnostics (ATLEPC2) (Refer to Chart 11) (Part 13 of 19)




CSORTN
MOVE CSECT AND PHASE NAME TO SAVE AREA
STORE REG 2 THRU 15 AND MOVE LAST ADDR OF FILE TO SAVE AREA ADDRTST
INITIALIZE CCW
LOAD RETURN ADDRESS OF TESTPH INTO REGISTER 5
ASSUMED PHASE SWITCH ON
READCSQ
CLEAR REG 3 THE RECORD COUNT REGISTER
IORTN KS
READ A CONTROL DICT PHYSICAL RCD
MOVE ADDRESS OF RECORD JUST READ INTO SCHADD
DSKDECR KT
DECREASE THE DISK ADDRESS
ASSUMPH
MOVE FIRST ADDR OF CTRL DICTIONARY FILE TO SEEKIN
READ A CTRL DICT PHYSICAL RECORD
GET THE NO. OF LOGICAL RCD IN THE PHYSICAL RECORD IN REGISTER 3
CALCULATE CORE ADDRESS OF THE LAST LOGICAL RCD *J3
1ST ENTRY A PHASE RCD
GOABORT
SAVE PHASE NAME AND RESTORE REG 2 THRU 15
TURN OFF ASSUMED PHASE AND CSECT PROC SWITCH
LOAD RETURN ADDRESS OF CHKTPA INTO REGISTER 10
PHORTN
STORE NEW DISK ADDR IN SEEKIN
MOVE THE NO. OF LOGICAL RCD IN PHY RCD PLUS LAST RCD INDR INTO REG 3
SHIFT REG 3 RIGHT ONE BIT TO MOVE OUT LAST RECORD INDICATOR
MOVE THE NO. OF LOGICAL RCD IN PHYSICAL RECORD IN TO REGISTER 4
CALCULATE CORE ADDRESS OF LAST LOG RECORD *J3
FIRSTCS
SET ASSUMCS BR SW TO UNC BR TO IND NO CSECT NAME SUPPLIED
*J3
MULTIPLY NO OF LOGICAL RECORDS IN THE PHYSICAL RECORD BY 16. THIS GIVES LENGTH OF PHYSICAL RECORD, MINUS THE TWO OVERHEAD BYTES, ADD THE ADDRESS OF THE I/O AREA. SUBTRACT 16. THIS GIVES THE START ADDRESS OF THE LAST LOGICAL RECORD OF THE PHYSICAL RECORD.
ADDR IN REG 5 TESTPH OR TESTCS *A5
TESTCS
TESTPH
PHASE RECORD
SAME PHASE NAME
PHSFOUND
STORE PHASE NUMBER IN TEMPORARY STORAGE PHSNUMB
INITIALIZE THE CSECT COUNTER TO ONE
TRSWICH
FURTHER QUALIFIED CSECT
NO-BR
YES-NOP
ASSUMED FURTHER QUALIFIED CSECT
SET ASSUMCS BRANCH SW TO NOP TO IND CSECT NAME SUPPLIED
INICS
MOVE FIRST ADDR OF FILE INTO SAVE AREA FOR CCW
LOAD RETURN ADDRESS OF TESTCS INTO REGISTER 5
*A5
EXIT TO ADDRESS IN REG 5
MODIFPH
DECR REG 4 BY 16 TO POINT TO NEXT LOGICAL RECORD
END OF PHYSICAL RECORD
END OF CNTRL DICT RCDS
LOAD RETURN ADDRESS OF TESTPH IN REGISTER 5
TESTMI
ASSUMED CONTROL SECT


KQ A4
TESTCS B1
CONTROL SECTION RECORD
MODIFCS B2
DECR REG 4 BY 16 TO POINT TO NEXT RCD UP CSECT COUNTER BY ONE
C1
CLEAR REG 15 AND INSERT PHASE NUMBER INTO IT
C2
END OF PHYSICAL RECORD
D1
PHASE NUMBER EQUAL
FILEEND D2
END OF CONTROL DICT
* KP-E2 KQ-F2
D3
LOAD RETURN TESTCS INTO REGISTER 5
KQ J5
READCSQ
ASSUMCS E1
CSECT NAME SUPPLIED BY USER
BR-NO
YES-NOP
GOABORT E2
RESTORE REGS 2 THRU 15. MODIFY INST AT LABEL ABORT TO Z CODE
F1
CSECT NAME EQUAL
F2
SET SWITCHES AT LABELS BTRPH AND MOVECS TO NOP
MOVECS G1
FURTHER QUALIFIED CSECT
YES-BR
NO-NOP
G2
SET SWITCHES AT LABELS TRSWICH AND BBACK TO BRANCH
* KB-D4 KS-D3
H1
MOVE CSECT NAME TO CSNAME DECREASE CSECT NUMBER BY ONE
ABORT H2
MOVE THE MODIFIED CHAR TO ERTAB FLD FIRST BYTE
* KB-F4 KD-G2
ABORT1 J2
PRINTPAT KK
WRITE AUTO-PATCH CARD IMAGE
K2
SVC 6 CANCEL JOB
TRMOVE B4
SET ASSUMCS AND MOVECS BRANCH SWITCHES TO NO-OP
C4
SET TRSWICH BRANCH SWITCH TO UNCONDITIONAL BRANCH
D4
DECR CSECT NO. BY 1. STORE CSECT NO. AND RELOCATION FACTOR
E4
RELOCATION FACTOR MINUS
MINUS2 F3
TURN ON QUAL CSECT MINUS RELOCATION FACTOR SW
F4
TURN OFF QUAL CSECT MINUS RELOCATION FACTOR SW
MINUS1 H4
TURN ON MINUS RELOCATION FACTOR SWITCH
POEXIT J4
RESTORE REGISTERS 2 THRU 15
K4
RETURN REG 10
B5
STORE CONTROL SECTION NUMBER IN CSNR
C5
SAVE RELOCATION FACTOR
D5
SET TRSWICH BRANCH SWITCH TO UNCONDITIONAL BRANCH
E5
SET ASSUMCS BRANCH SWITCH TO NO-OP
F5
SET MOVECS BRANCH SWITCH TO NO-OP
G5
IS RELOCATION FACTOR MINUS
H5
TURN OFF MINUS RELOCATION FACTOR SWITCH


SYMSRCH
MOVE START ADDRESS OF DISK SYMBOL AREA TO SADR
STORE THE LINK REGISTER 3 IN EIR3HLD
E1RD
MOVE DISK ADDR TO HOLD AREA AND RESET RCD COUNT IN RECTBL
DSKINCR KT
INCREASE DISK ADDRESS
MOVE UPDATED DISK ADDRESS INTO E1ADDR
E1RD1
LOAD ADDR OF CCB INTO REG 1
READ A PHYSICAL RECORD SVC 0
I/O COMPLETE
NO
YES
END OF SYM RECORD FILE
YES
NO
LOAD ADDR OF FIRST LOGICAL SYM RCD INTO REG 4
LOAD NO. OF LOGICAL RCD IN PHYSICAL RCD INTO REG 3
MULTIPLY REG 3 BY 15. ADD ADDR OF INPUT AREA. POINT TO END OF PHYSICAL RCD
PAD END OF RECORD WITH 15 F'S
E1CMP
CSECT NUMBERS SAME
SYMBOL THE SAME
MOVE SYMBOL ADDRESS INTO E1DISP+1 AND RETURN ADDR IN LINK REG 3
RETURN REG 3
CHKSW1
RESTORE REG 2 THRU 15
SYMBOLS BEING PROCESSED
SYMERTN
MODIFY INST AT LABEL ABORT TO SHOW Z FOR CANCEL CODE
ABORT
E1PT1
UP INPUT AREA POINTER BY 15 TO NEXT LOGICAL SYM RECORD
END OF PHYSICAL RECORD
PHTRRTN
MOVE PHASE NAME FROM SAVE AREA
SET BTRPH BRANCH SWITCH TO UNCONDITIONAL BRANCH
SET BBACK BRANCH SWITCH TO NO-OP
PHQRTN+6
IORTN
LOAD ADDR OF CCBIN INTO REG 1
SVC 0 EXECUTE CHANNEL PROGRAM
I/O OPERATION COMPLETE
RET TO ADDR IN LNK REG 2
CSTRRTN
MOVE PHASE NAME FROM SAVE AREA
MOVE CSECT NAME FROM SAVE AREA
SET TRSWICH TO NO-OP
SET MOVECS BRANCH SWITCH TO UNCONDITIONAL BRANCH
CSQRTN+12



*B4
RECTBL IS AN 8-BYTE FIELD IN THE FOLLOWING FORMAT: CCHRCCHR WHERE C IS CYLINDER NUMBER, H IS HEAD NUMBER, AND R IS RECORD NUMBER. THE FIRST 4 BYTES OF RECTBL FIELD, CCHR, CONTAIN THE MAX VALUES OF THE CYL, HEAD AND RECORD. THE NEXT 4 BYTES, CCHR, CONTAIN THE LOW LIMIT VALUE THAT THE CYL, HEAD AND RECORD ARE RESET TO WHEN THE MAXIMUM VALUE HAS BEEN REACHED FOR THEIR RESPECTIVE FIELD.

*G4
TABDECR IS AN 8-BYTE FIELD IN THE FOLLOWING FORMAT: CCHRCCHR WHERE C IS CYLINDER NO. H IS HEAD NUMBER, AND R IS RECORD NUMBER. THE FIRST 4 BYTES OF TABDECR FIELD, CCHR CONTAIN THE MINIMUM VALUE OF THE CYL, HEAD AND RECORD. THE NEXT 4 BYTES, CCHR, CONTAIN THE HIGH LIMIT VALUE THAT THE CYL, HEAD AND RECORD ARE RESET TO WHEN THE MINIMUM VALUE HAS BEEN REACHED FOR THEIR RESPECTIVE FIELDS.

Chart KU. Autopatch Diagnostics (ATLEFC2) (Refer to Chart 11) (Part 18 of 19)

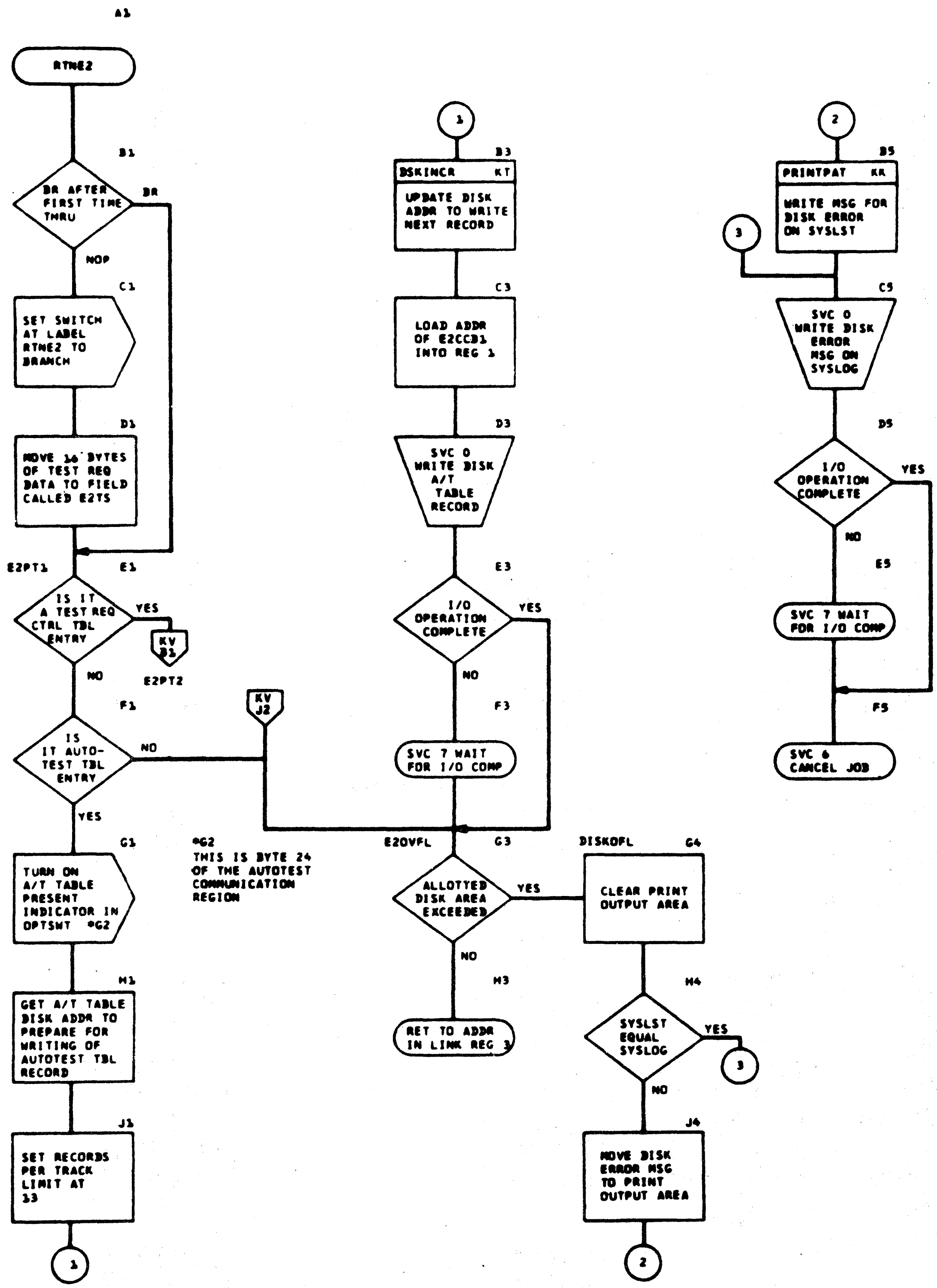
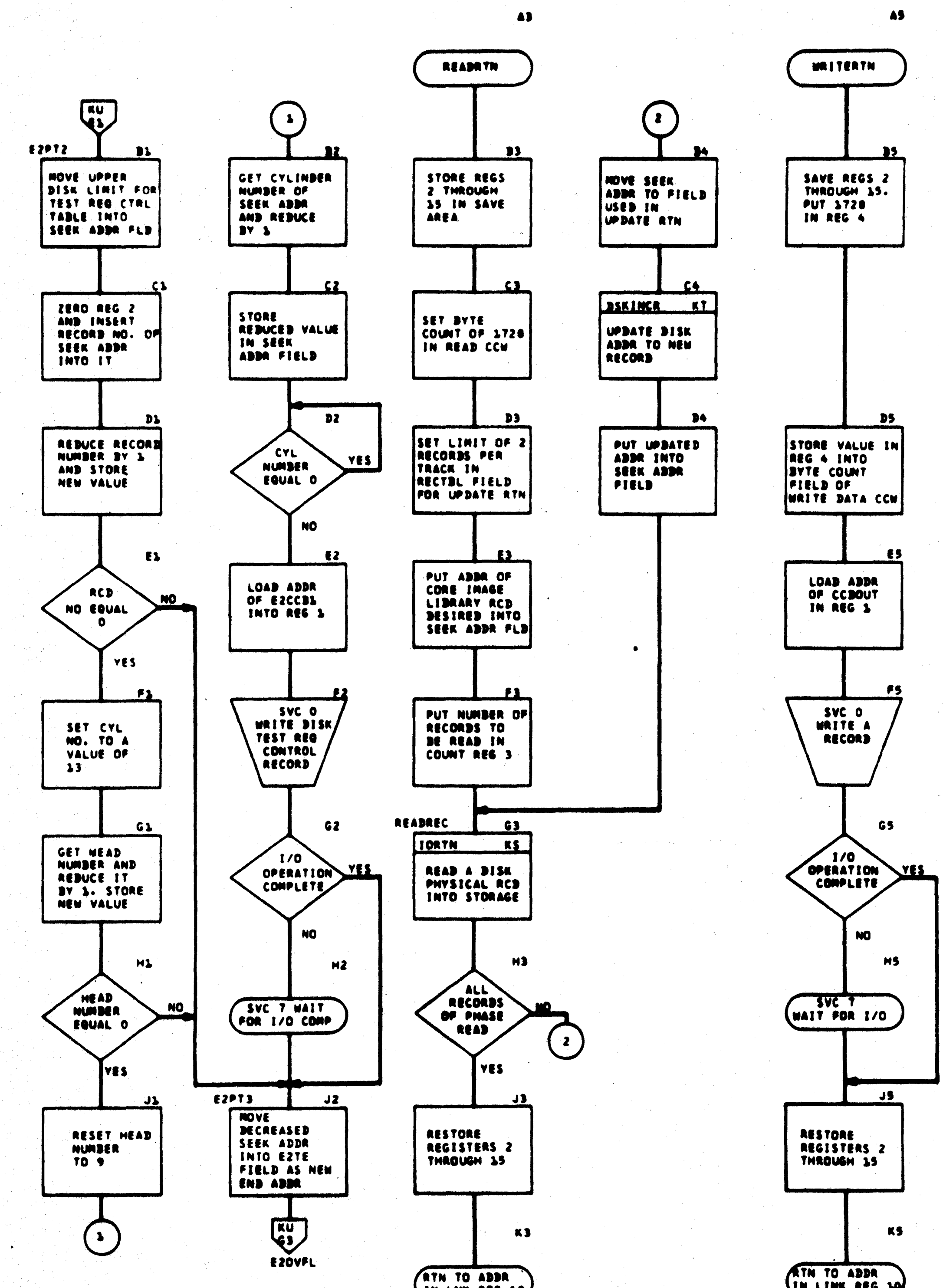
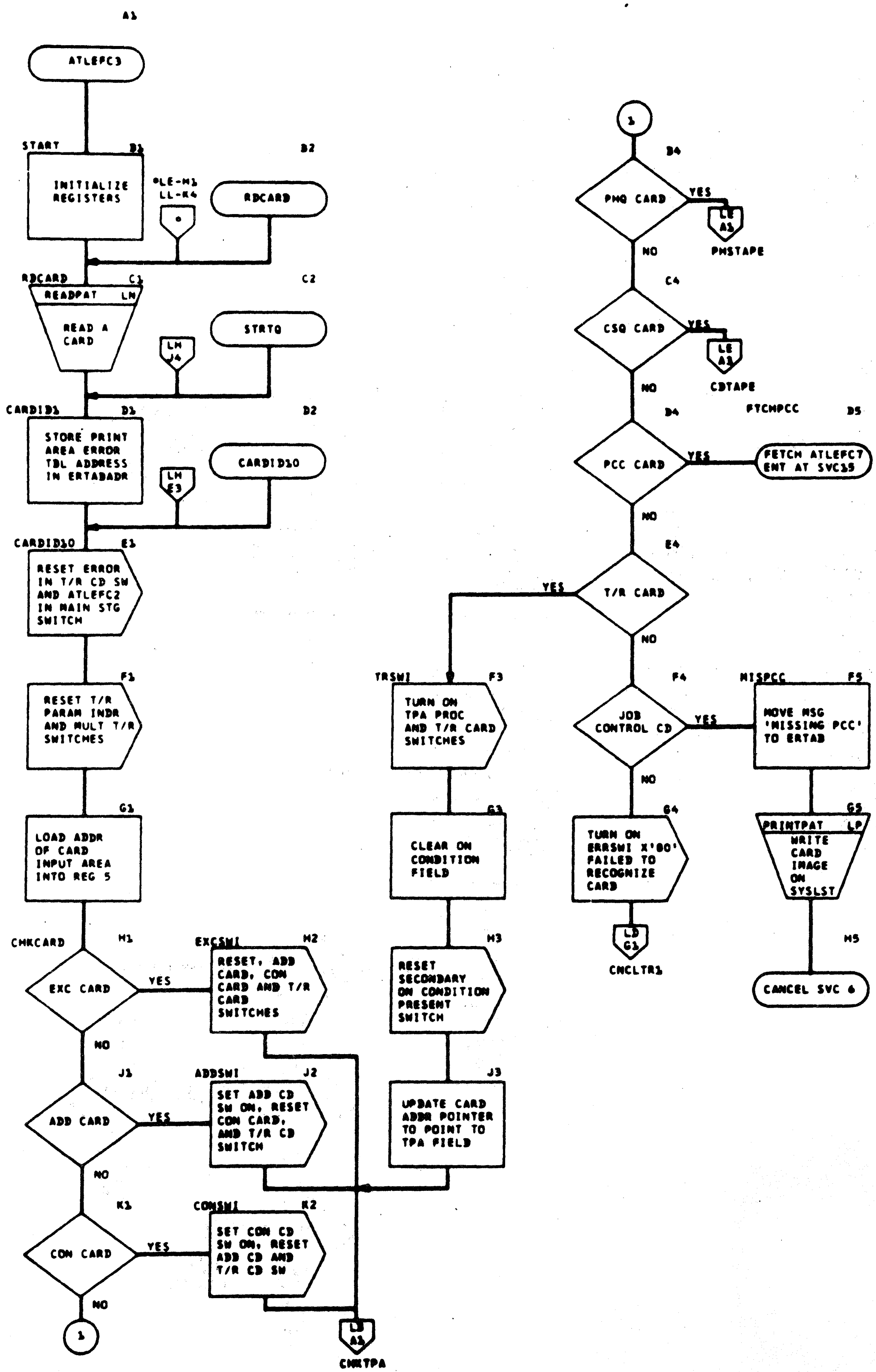


Chart KV. Autopatch Diagnostics (ATLEFC2) (Refer to Chart 11) (Part 19 of 19)




ATLEFC3
START
INITIALIZE REGISTERS
*LE-H1 LL-K4
RDCARD
READPAT LN
READ A CARD
STRTO
CARDID1
STORE PRINT AREA ERROR TBL ADDRESS IN ERTABADR
CARDID10
RESET ERROR IN T/R CD SW AND ATLEFC2 IN MAIN STG SWITCH
RESET T/R PARAM INDR AND MULT T/R SWITCHES
LOAD ADDR OF CARD INPUT AREA INTO REG 5
CHKCARD
EXC CARD
EXCSWI
RESET, ADD CARD, CON CARD AND T/R CARD SWITCHES
ADD CARD
ADDSWI
SET ADD CD SW ON, RESET CON CARD, AND T/R CD SWITCH
CON CARD
CONSWI
SET CON CD SW ON, RESET ADD CD AND T/R CD SW
CHKTPA
PHQ CARD
PHSTAPE
CSQ CARD
CDTAPE
PCC CARD
FTCHPCC
FETCH ATLEFCT ENT AT SVC15
T/R CARD
TRSWI
TURN ON TPA PROC AND T/R CARD SWITCHES
CLEAR ON CONDITION FIELD
RESET SECONDARY ON CONDITION PRESENT SWITCH
UPDATE CARD ADDR POINTER TO POINT TO TPA FIELD
JOB CONTROL CD
TURN ON ERRSWI X'80' FAILED TO RECOGNIZE CARD
CNCLTR1
MISPCC
MOVE MSG 'MISSING PCC' TO ERTAB
PRINTPAT LP
WRITE CARD IMAGE ON SYSLST
CANCEL SVC 6
YES
NO

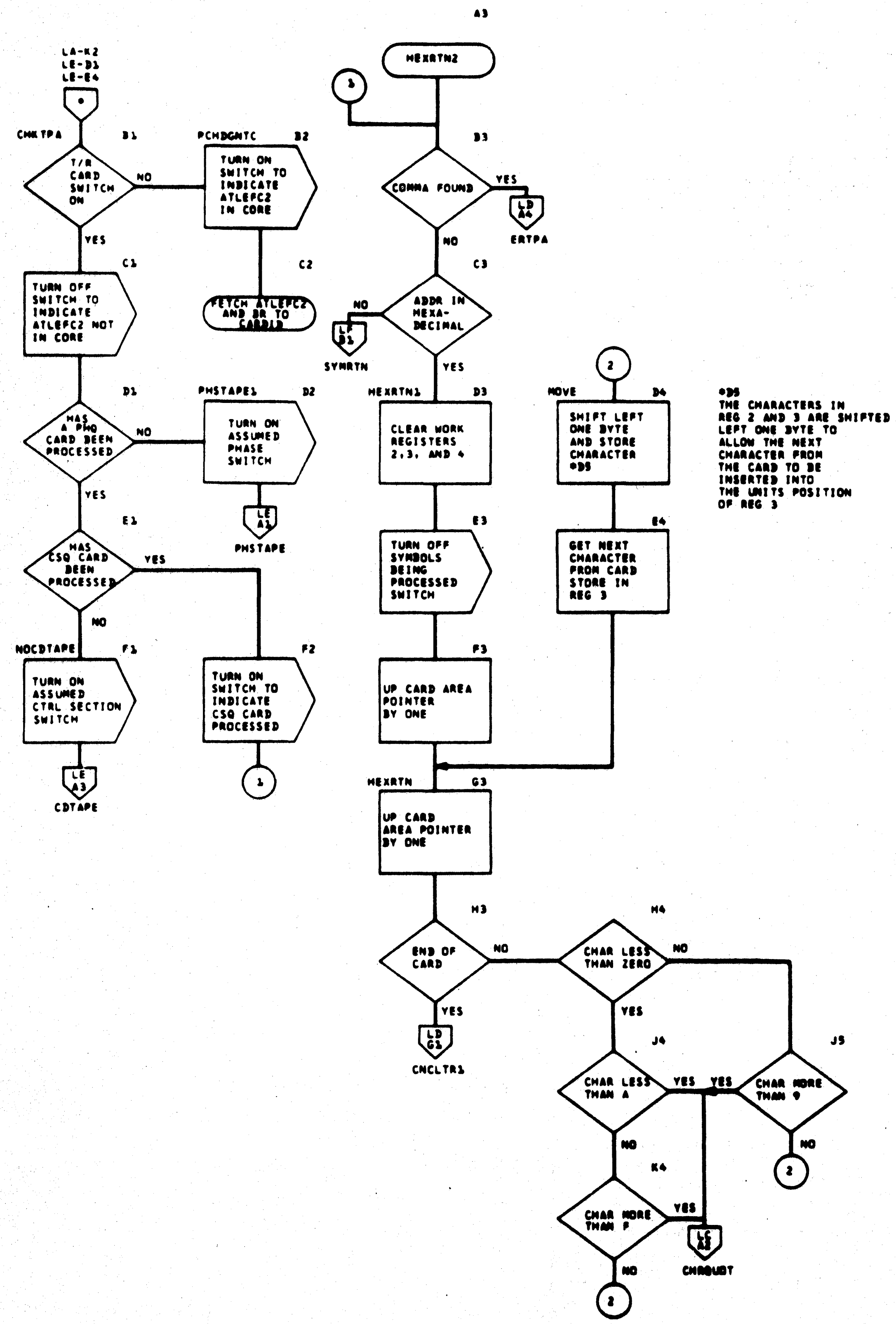
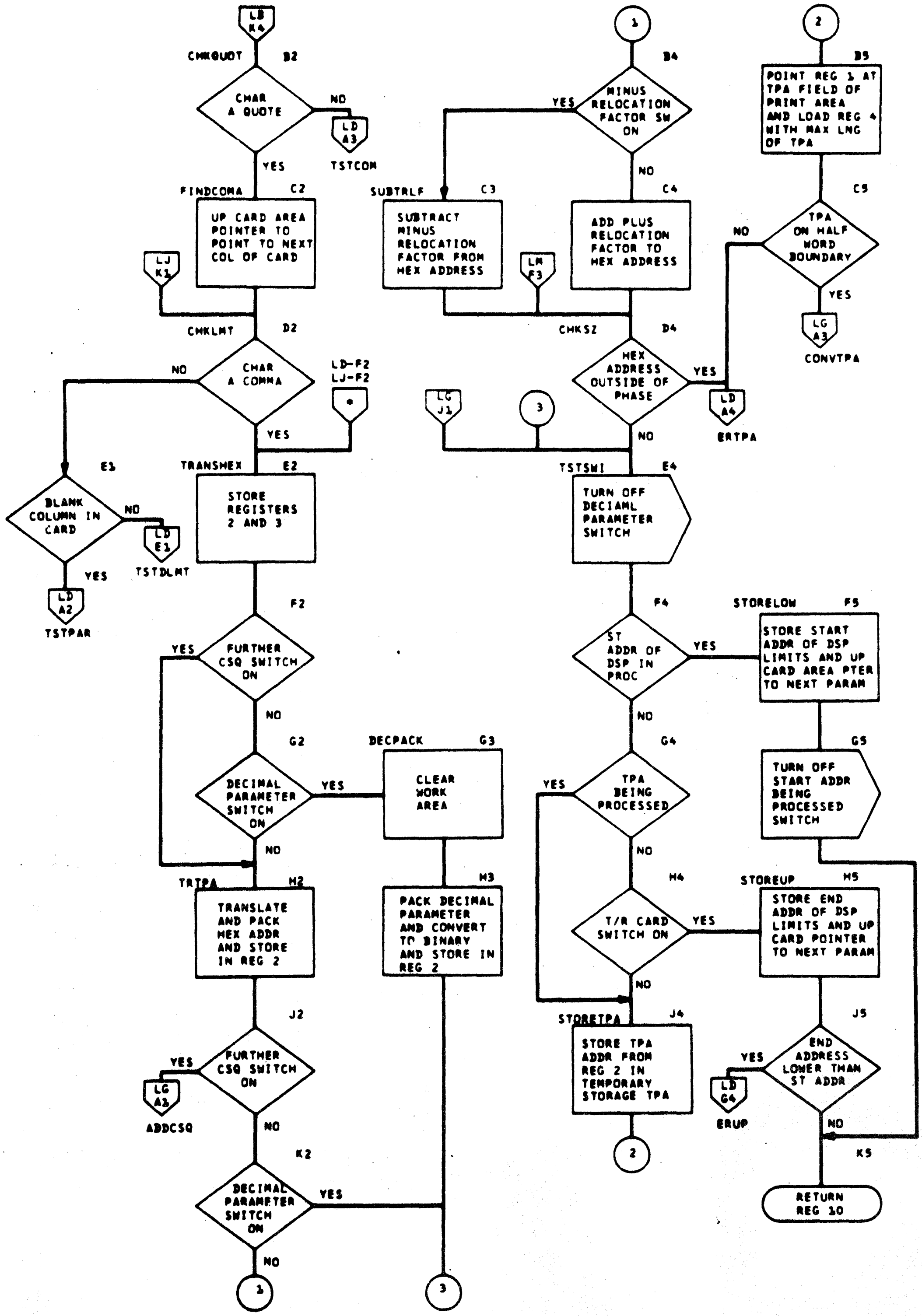

LA-K2
LE-B1
LE-E4
CHKTPA
B1
T/R CARD SWITCH ON
NO
PCHDGNTC
B2
TURN ON SWITCH TO INDICATE ATLEFC2 IN CORE
C2
FETCH ATLEFC2 AND BR TO CARDID
YES
C1
TURN OFF SWITCH TO INDICATE ATLEFC2 NOT IN CORE
D1
HAS A PHQ CARD BEEN PROCESSED
NO
PHSTAPE1
D2
TURN ON ASSUMED PHASE SWITCH
LE A1
PHSTAPE
YES
E1
HAS CSQ CARD BEEN PROCESSED
YES
NO
NOCDTAPE
F1
TURN ON ASSUMED CTRL SECTION SWITCH
LE A3
CDTAPE
F2
TURN ON SWITCH TO INDICATE CSQ CARD PROCESSED
1
A3
HEXRTN2
1
B3
COMMA FOUND
YES
LD A4
ERTPA
NO
C3
ADDR IN HEXA-DECIMAL
NO
LF B1
SYMRTN
YES
HEXRTN1
D3
CLEAR WORK REGISTERS 2,3, AND 4
E3
TURN OFF SYMBOLS BEING PROCESSED SWITCH
F3
UP CARD AREA POINTER BY ONE
HEXRTN
G3
UP CARD AREA POINTER BY ONE
H3
END OF CARD
YES
LD G1
CNCLTR1
NO
2
MOVE
D4
SHIFT LEFT ONE BYTE AND STORE CHARACTER *D5
E4
GET NEXT CHARACTER FROM CARD STORE IN REG 3
*D5
THE CHARACTERS IN REG 2 AND 3 ARE SHIFTED LEFT ONE BYTE TO ALLOW THE NEXT CHARACTER FROM THE CARD TO BE INSERTED INTO THE UNITS POSITION OF REG 3
H4
CHAR LESS THAN ZERO
NO
YES
J4
CHAR LESS THAN A
YES
YES
J5
CHAR MORE THAN 9
NO
2
NO
K4
CHAR MORE THAN F
YES
LC A2
CHRQUOT
NO
2

Chart LC. Test Request Diagnostics (ATLEFC3) (Refer to Chart 11) (Part 3 of 19)



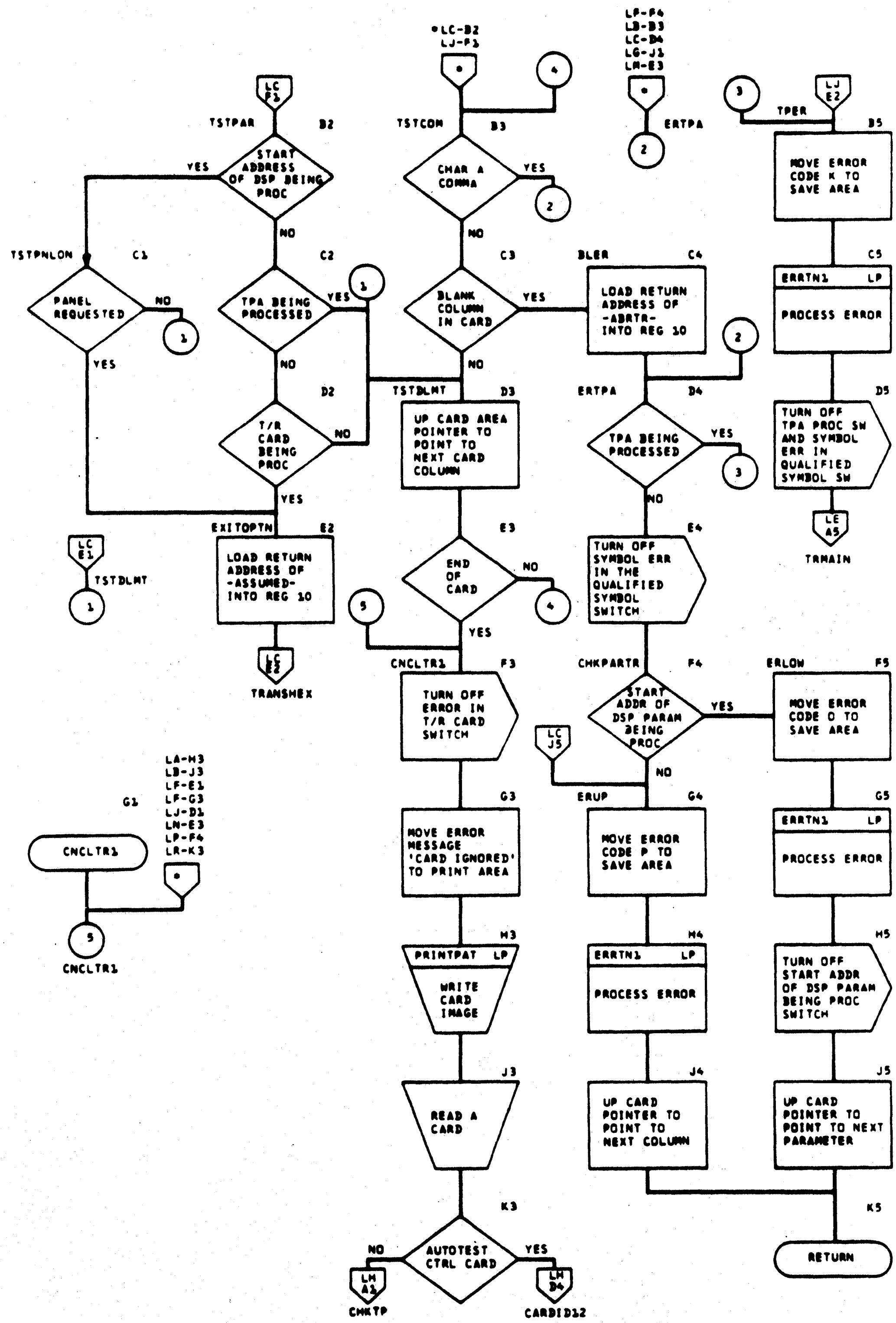

TSTPAR
B2
START ADDRESS OF DSP BEING PROC
YES
NO
TSTCOM
B3
CHAR A COMMA
ERTPA
TPER
B5
MOVE ERROR CODE K TO SAVE AREA
TSTPNLON
C1
PANEL REQUESTED
C2
TPA BEING PROCESSED
C3
BLANK COLUMN IN CARD
BLER
C4
LOAD RETURN ADDRESS OF -ABRTR- INTO REG 10
C5
ERRTN1 LP
PROCESS ERROR
D2
T/R CARD BEING PROC
TSTBLMT
D3
UP CARD AREA POINTER TO POINT TO NEXT CARD COLUMN
D4
TPA BEING PROCESSED
D5
TURN OFF TPA PROC SW AND SYMBOL ERR IN QUALIFIED SYMBOL SW
TRMAIN
EXITOPTN
E2
LOAD RETURN ADDRESS OF -ASSUMED- INTO REG 10
TSTBLMT
E3
END OF CARD
E4
TURN OFF SYMBOL ERR IN THE QUALIFIED SYMBOL SWITCH
TRANSHEX
CNCLTR1
F3
TURN OFF ERROR IN T/R CARD SWITCH
CHKPARTR
F4
START ADDR OF DSP PARAM BEING PROC
ERLOW
F5
MOVE ERROR CODE O TO SAVE AREA
LA-H3
LB-J3
LF-E1
LF-G3
LJ-D1
LN-E3
LP-F4
LR-K3
G1
CNCLTR1
G3
MOVE ERROR MESSAGE 'CARD IGNORED' TO PRINT AREA
ERUP
G4
MOVE ERROR CODE P TO SAVE AREA
G5
ERRTN1 LP
PROCESS ERROR
CNCLTR1
H3
PRINTPAT LP
WRITE CARD IMAGE
H4
ERRTN1 LP
PROCESS ERROR
H5
TURN OFF START ADDR OF DSP PARAM BEING PROC SWITCH
J3
READ A CARD
J4
UP CARD POINTER TO POINT TO NEXT COLUMN
J5
UP CARD POINTER TO POINT TO NEXT PARAMETER
K3
AUTOTEST CTRL CARD
CHKTP
CARDID12
K5
RETURN

Chart LE. Test Request Diagnostics (ATLEFC3) (Refer to Chart 11) (Part 5 of 19)

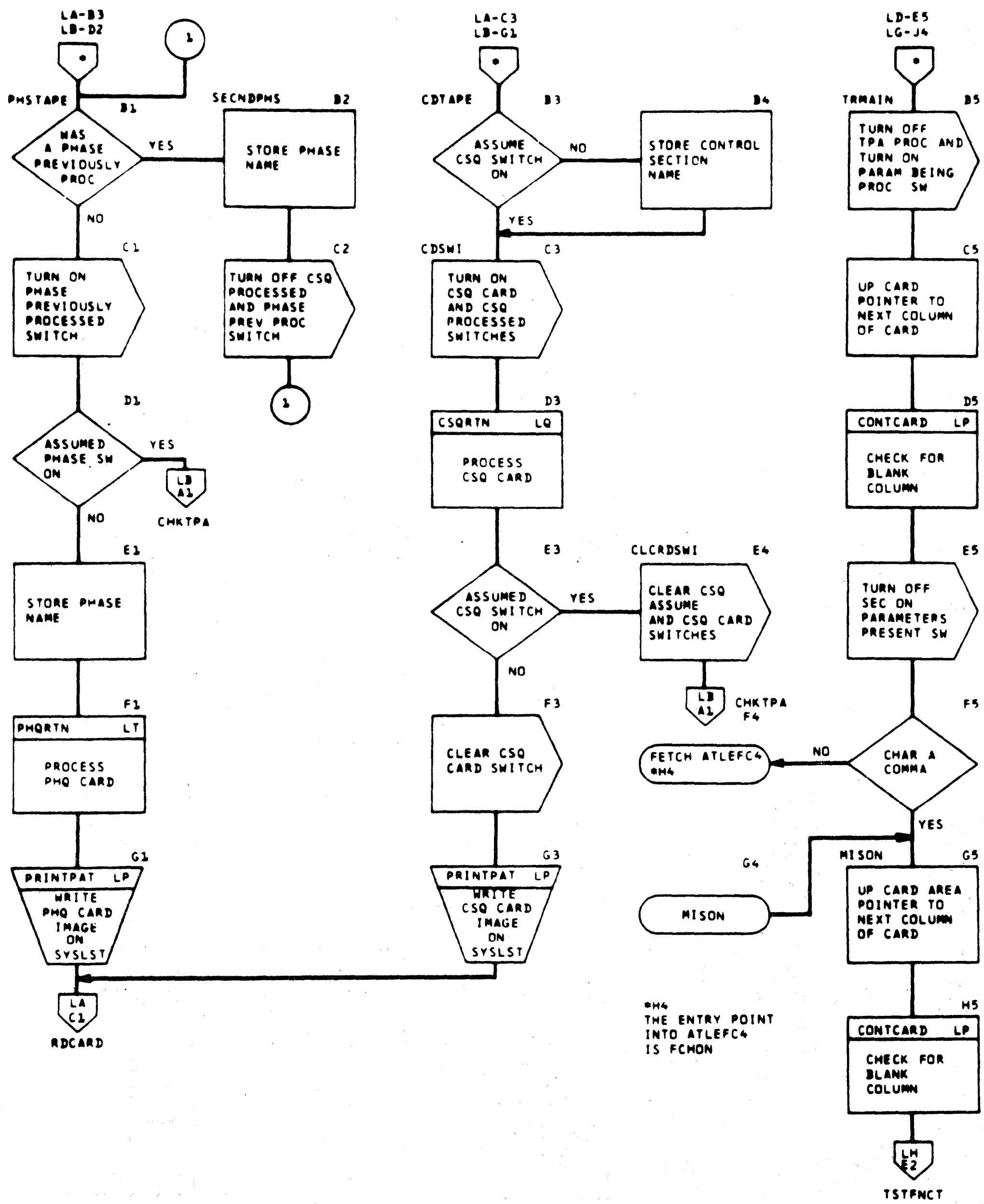
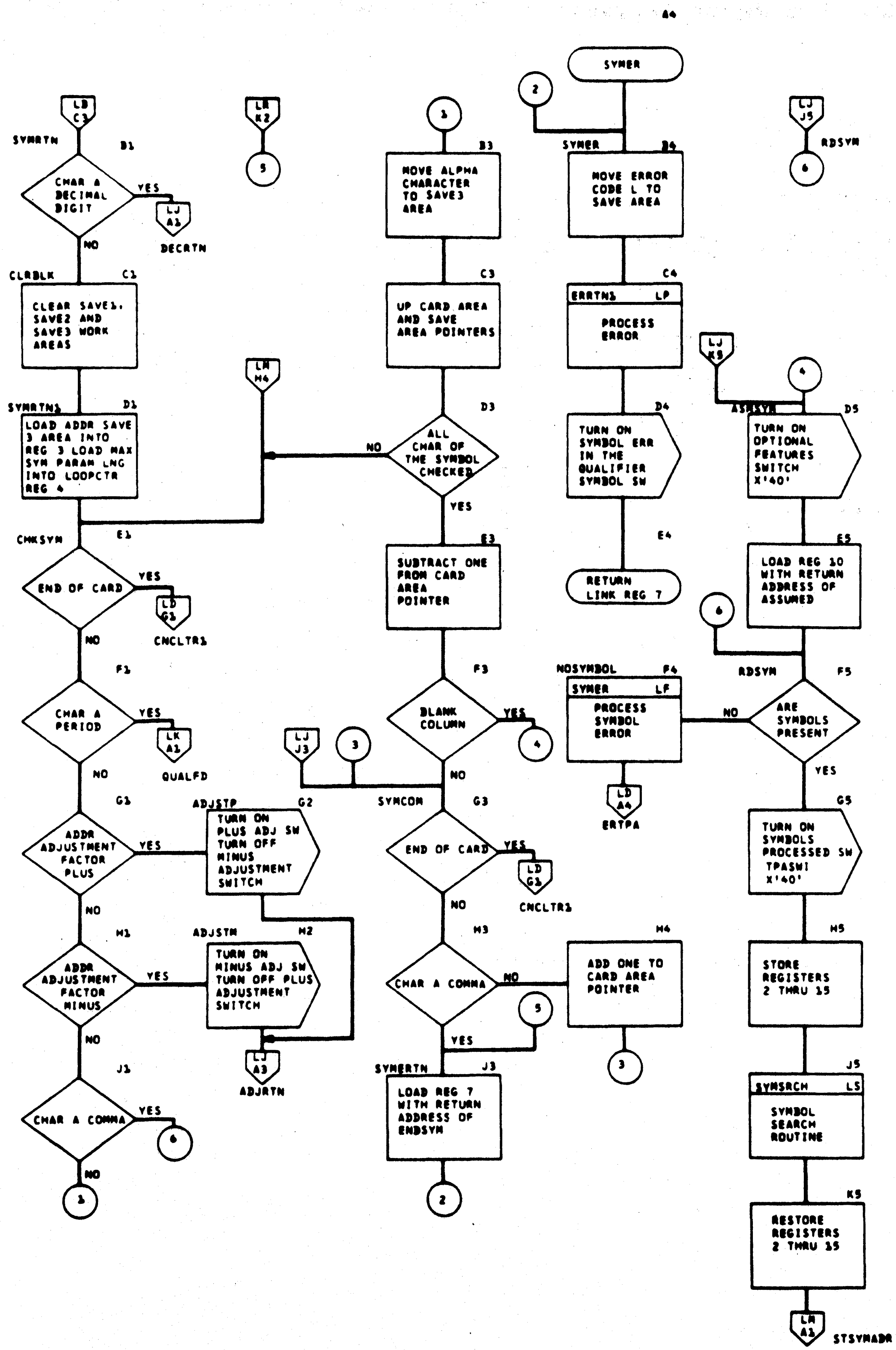
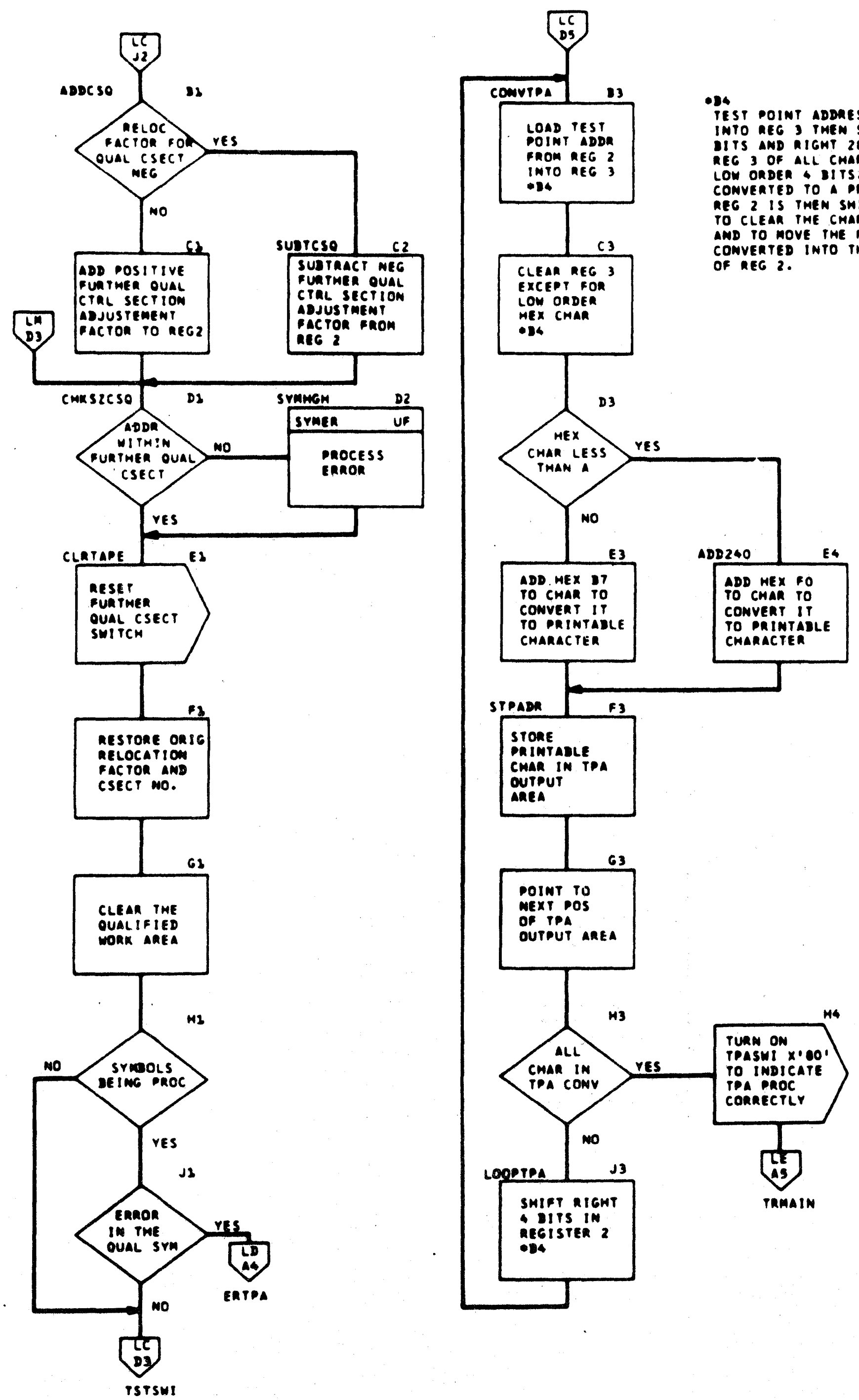
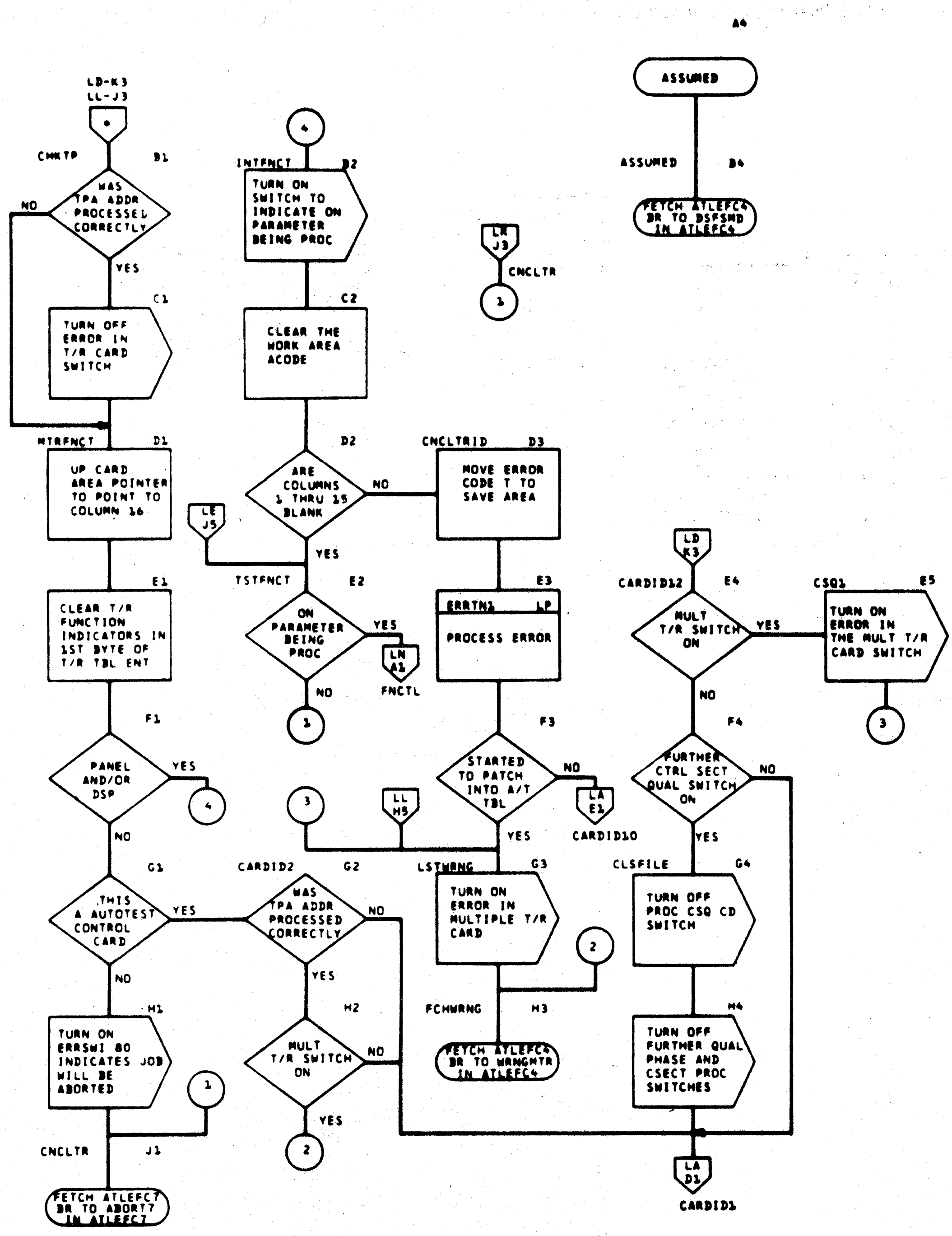


Chart LF. Test Request Diagnostics (ATLEFC3) (Refer to Chart 11) (Part 6 of 19)

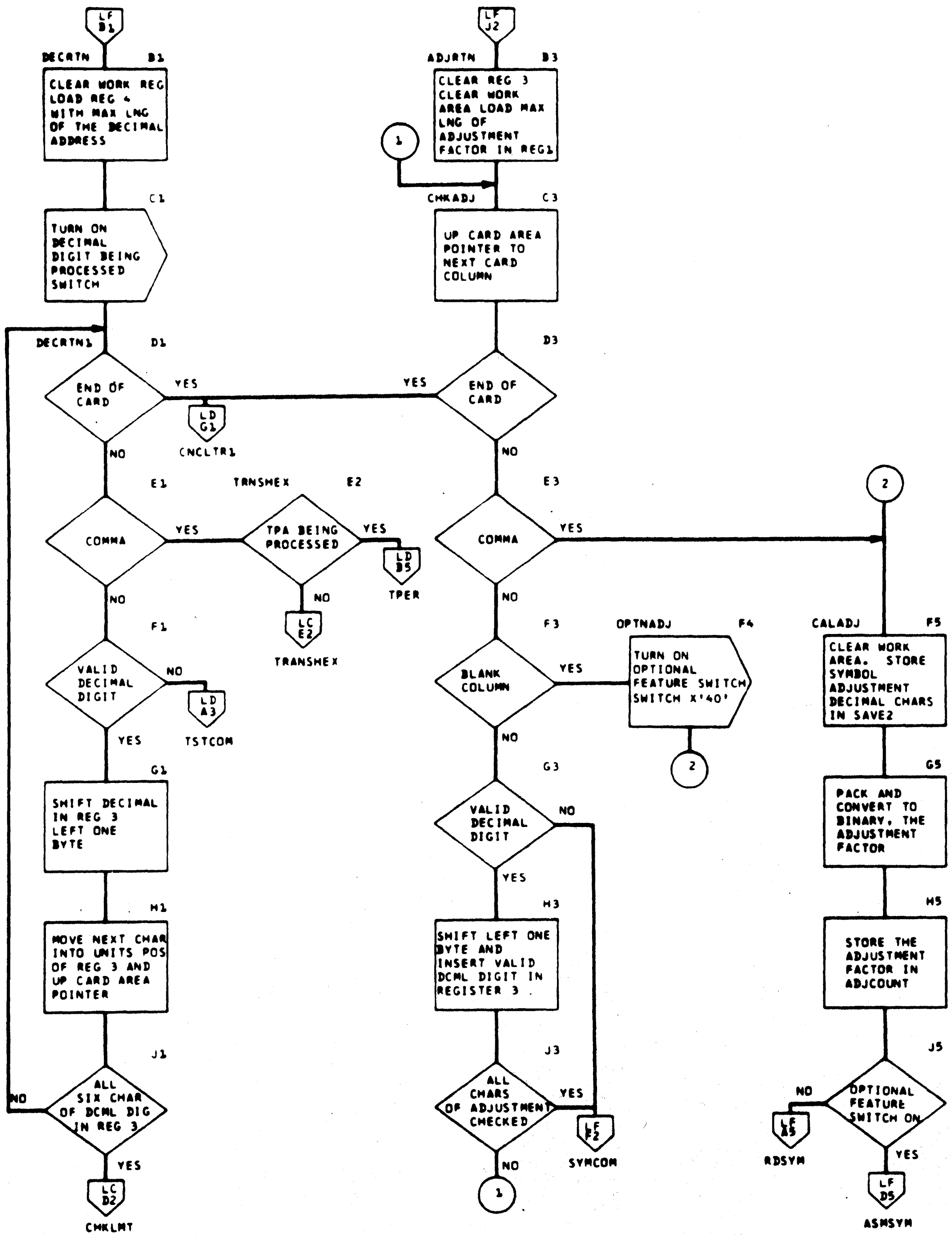




*B4
TEST POINT ADDRESS IS LOADED INTO REG 3 THEN SHIFTED LEFT 28 BITS AND RIGHT 28 BITS TO CLEAR REG 3 OF ALL CHARACTERS BUT THE LOW ORDER 4 BITS. THIS IS THEN CONVERTED TO A PRINTABLE CHARACTER. REG 2 IS THEN SHIFTED RIGHT 4 BITS TO CLEAR THE CHARACTER JUST CONVERTED AND TO MOVE THE NEXT CHARACTER TO BE CONVERTED INTO THE UNITS POSITION OF REG 2.



A4
ASSUMED
ASSUMED B4
FETCH ATLEFC4 BR TO DSFSMD IN ATLEFC4
LD-K3
LL-J3
CHKTP B1
WAS TPA ADDR PROCESSED CORRECTLY
NO
YES
C1
TURN OFF ERROR IN T/R CARD SWITCH
NTRFNCT D1
UP CARD AREA POINTER TO POINT TO COLUMN 16
E1
CLEAR T/R FUNCTION INDICATORS IN 1ST BYTE OF T/R TBL ENT
F1
PANEL AND/OR DSP
G1
THIS A AUTOTEST CONTROL CARD
H1
TURN ON ERRSWI 80 INDICATES JOB WILL BE ABORTED
CNCLTR J1
FETCH ATLEFC7 BR TO ABORT7 IN ATLEFC7
INTFNCT B2
TURN ON SWITCH TO INDICATE ON PARAMETER BEING PROC
C2
CLEAR THE WORK AREA ACODE
D2
ARE COLUMNS 1 THRU 15 BLANK
LE J5
TSTFNCT E2
ON PARAMETER BEING PROC
LN A1
FNCTL
CARDID2 G2
WAS TPA ADDR PROCESSED CORRECTLY
H2
MULT T/R SWITCH ON
LK J3
CNCLTR
CNCLTRID D3
MOVE ERROR CODE T TO SAVE AREA
E3
ERRTN1 LP
PROCESS ERROR
F3
STARTED TO PATCH INTO A/T TBL
LA E1
CARDID10
LL H5
LSTWRNG G3
TURN ON ERROR IN MULTIPLE T/R CARD
FCHWRNG H3
FETCH ATLEFC4 BR TO WRNGMTR IN ATLEFC4
LD K3
CARDID12 E4
MULT T/R SWITCH ON
CSQ1 E5
TURN ON ERROR IN THE MULT T/R CARD SWITCH
F4
FURTHER CTRL SECT QUAL SWITCH ON
CLSFILE G4
TURN OFF PROC CSQ CD SWITCH
H4
TURN OFF FURTHER QUAL PHASE AND CSECT PROC SWITCHES
LA D1
CARDID1



LF B1
DECRTN B1
CLEAR WORK REG LOAD REG 4 WITH MAX LNG OF THE DECIMAL ADDRESS
C1
TURN ON DECIMAL DIGIT BEING PROCESSED SWITCH
DECRTN1 D1
END OF CARD
YES
LD G1
CNCLTR1
NO
E1
COMMA
YES
TRNSHEX E2
TPA BEING PROCESSED
YES
LD B5
TPER
NO
LC E2
TRANSHEX
NO
F1
VALID DECIMAL DIGIT
NO
LD A3
TSTCOM
YES
G1
SHIFT DECIMAL IN REG 3 LEFT ONE BYTE
H1
MOVE NEXT CHAR INTO UNITS POS OF REG 3 AND UP CARD AREA POINTER
J1
ALL SIX CHAR OF DCML DIG IN REG 3
NO
YES
LC D2
CHKLMT
LF J2
ADJRTN B3
CLEAR REG 3 CLEAR WORK AREA LOAD MAX LNG OF ADJUSTMENT FACTOR IN REG1
1
CHKADJ C3
UP CARD AREA POINTER TO NEXT CARD COLUMN
D3
END OF CARD
YES
NO
E3
COMMA
YES
2
NO
F3
BLANK COLUMN
YES
OPTNADJ F4
TURN ON OPTIONAL FEATURE SWITCH SWITCH X'40'
2
NO
G3
VALID DECIMAL DIGIT
NO
YES
H3
SHIFT LEFT ONE BYTE AND INSERT VALID DCML DIGIT IN REGISTER 3 .
J3
ALL CHARS OF ADJUSTMENT CHECKED
YES
LF F2
SYMCOM
NO
1
CALADJ F5
CLEAR WORK AREA. STORE SYMBOL ADJUSTMENT DECIMAL CHARS IN SAVE2
G5
PACK AND CONVERT TO BINARY, THE ADJUSTMENT FACTOR
H5
STORE THE ADJUSTMENT FACTOR IN ADJCOUNT
J5
OPTIONAL FEATURE SWITCH ON
NO
LF A5
RDSYM
YES
LF D5
ASMSYM

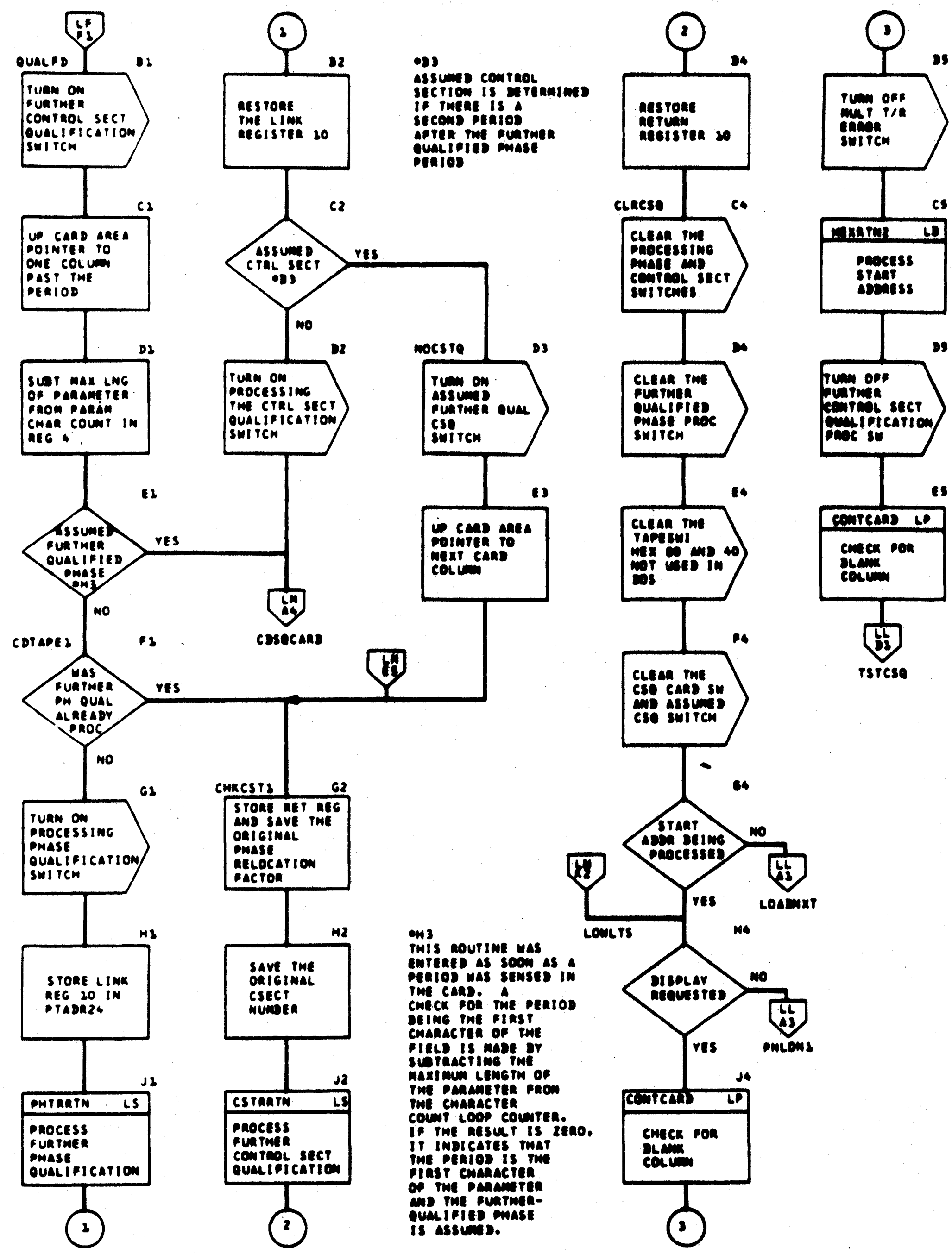
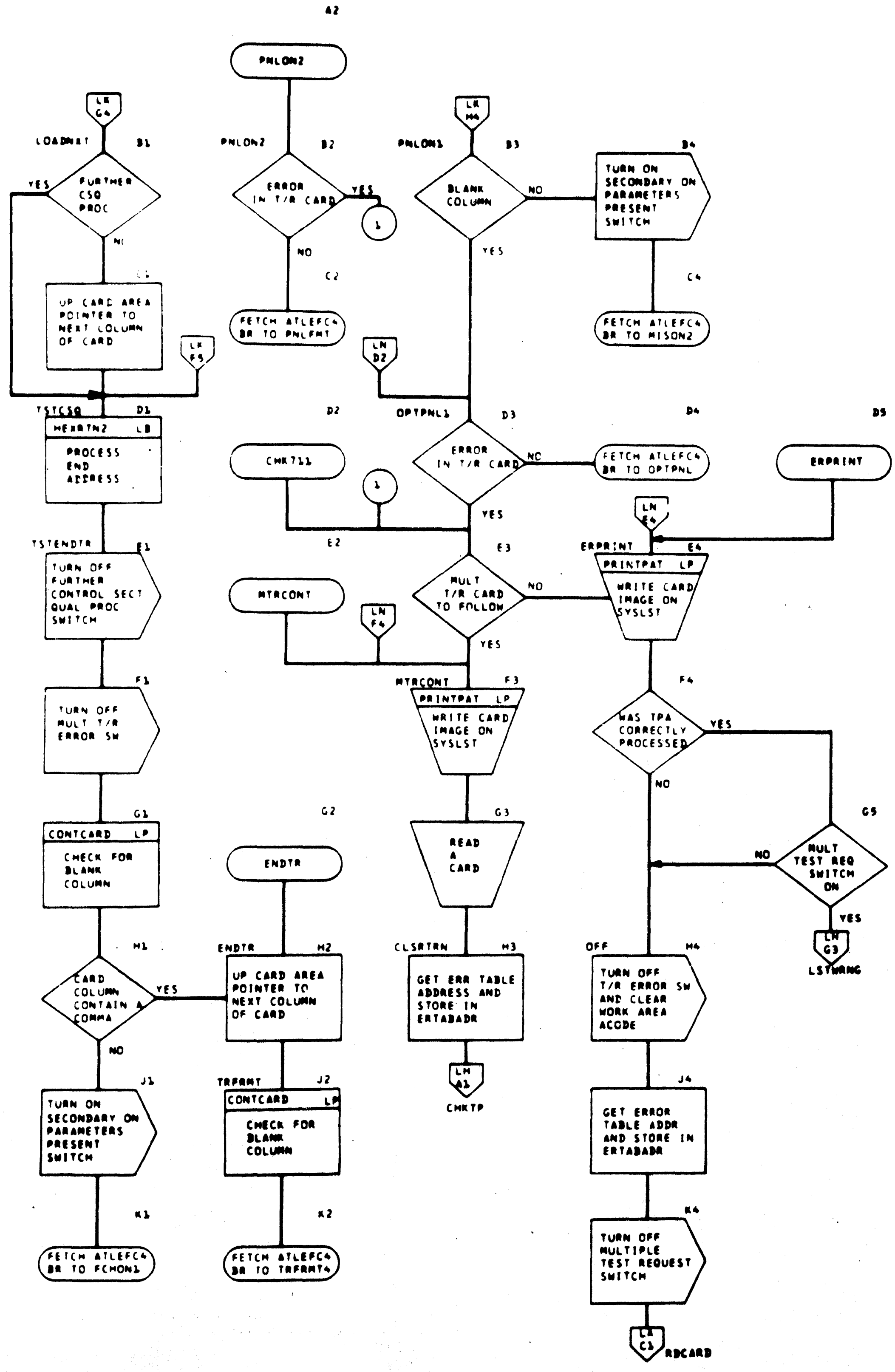
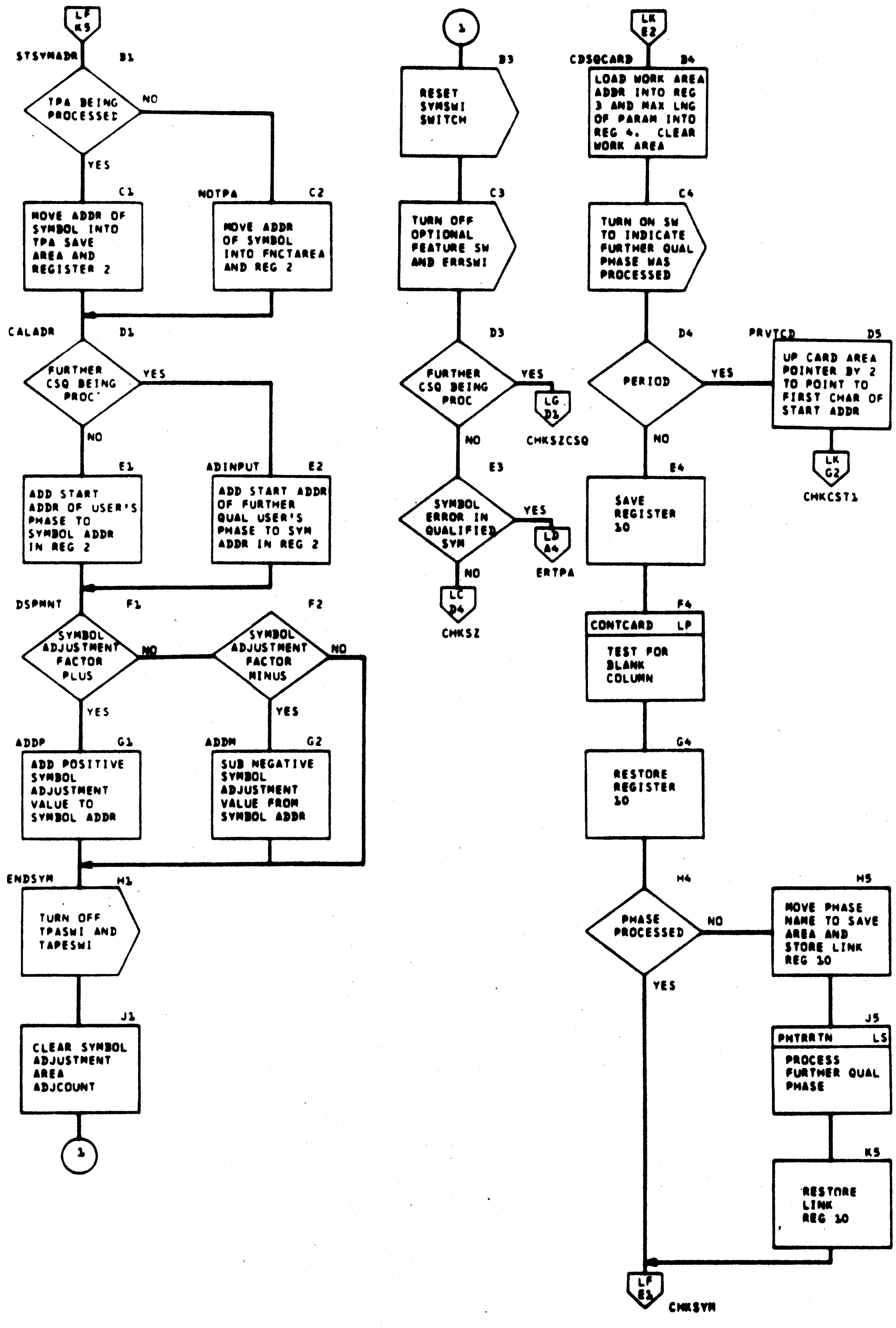

LF F1
QUALFD
B1
TURN ON FURTHER CONTROL SECT QUALIFICATION SWITCH
C1
UP CARD AREA POINTER TO ONE COLUMN PAST THE PERIOD
D1
SUBT MAX LNG OF PARAMETER FROM PARAM CHAR COUNT IN REG 4
E1
ASSUMED FURTHER QUALIFIED PHASE *H3
YES
NO
CDTAPE1
F1
WAS FURTHER PH QUAL ALREADY PROC
YES
NO
G1
TURN ON PROCESSING PHASE QUALIFICATION SWITCH
H1
STORE LINK REG 10 IN PTADR24
J1
PHTRRTN LS
PROCESS FURTHER PHASE QUALIFICATION
1
B2
RESTORE THE LINK REGISTER 10
C2
ASSUMED CTRL SECT *B3
YES
NO
D2
TURN ON PROCESSING THE CTRL SECT QUALIFICATION SWITCH
LM A4
CDSQCARD
CHKCST1
G2
STORE RET REG AND SAVE THE ORIGINAL PHASE RELOCATION FACTOR
H2
SAVE THE ORIGINAL CSECT NUMBER
J2
CSTRRTN LS
PROCESS FURTHER CONTROL SECT QUALIFICATION
2
*B3
ASSUMED CONTROL SECTION IS DETERMINED IF THERE IS A SECOND PERIOD AFTER THE FURTHER QUALIFIED PHASE PERIOD
NOCSTQ
D3
TURN ON ASSUMED FURTHER QUAL CSQ SWITCH
E3
UP CARD AREA POINTER TO NEXT CARD COLUMN
LM E5
*H3
THIS ROUTINE WAS ENTERED AS SOON AS A PERIOD WAS SENSED IN THE CARD. A CHECK FOR THE PERIOD BEING THE FIRST CHARACTER OF THE FIELD IS MADE BY SUBTRACTING THE MAXIMUM LENGTH OF THE PARAMETER FROM THE CHARACTER COUNT LOOP COUNTER. IF THE RESULT IS ZERO, IT INDICATES THAT THE PERIOD IS THE FIRST CHARACTER OF THE PARAMETER AND THE FURTHER-QUALIFIED PHASE IS ASSUMED.
2
B4
RESTORE RETURN REGISTER 10
CLRCSQ
C4
CLEAR THE PROCESSING PHASE AND CONTROL SECT SWITCHES
D4
CLEAR THE FURTHER QUALIFIED PHASE PROC SWITCH
E4
CLEAR THE TAPESW1 HEX 80 AND 40 NOT USED IN DOS
F4
CLEAR THE CSQ CARD SW AND ASSUMED CSQ SWITCH
G4
START ADDR BEING PROCESSED
NO
LL A1
LOADNXT
YES
LM K2
LOWLTS
H4
DISPLAY REQUESTED
NO
LL A3
PNLON1
YES
J4
CONTCARD LP
CHECK FOR BLANK COLUMN
3
3
B5
TURN OFF MULT T/R ERROR SWITCH
C5
HEXRTN2 LD
PROCESS START ADDRESS
D5
TURN OFF FURTHER CONTROL SECT QUALIFICATION PROC SW
E5
CONTCARD LP
CHECK FOR BLANK COLUMN
LL D1
TSTCSQ

Chart LL. Test Request Diagnostics (ATLEFC3) (Refer to Chart 11) (Part 11 of 19)





LF K5
STSYMADR B1
TPA BEING PROCESSED
NO
YES
C1
MOVE ADDR OF SYMBOL INTO TPA SAVE AREA AND REGISTER 2
NOTPA C2
MOVE ADDR OF SYMBOL INTO FNCTAREA AND REG 2
CALADR D1
FURTHER CSQ BEING PROC
YES
NO
E1
ADD START ADDR OF USER'S PHASE TO SYMBOL ADDR IN REG 2
ADINPUT E2
ADD START ADDR OF FURTHER QUAL USER'S PHASE TO SYM ADDR IN REG 2
DSPMNT F1
SYMBOL ADJUSTMENT FACTOR PLUS
NO
F2
SYMBOL ADJUSTMENT FACTOR MINUS
NO
YES
YES
ADDP G1
ADD POSITIVE SYMBOL ADJUSTMENT VALUE TO SYMBOL ADDR
ADDM G2
SUB NEGATIVE SYMBOL ADJUSTMENT VALUE FROM SYMBOL ADDR
ENDSYM H1
TURN OFF TPASWI AND TAPESWI
J1
CLEAR SYMBOL ADJUSTMENT AREA ADJCOUNT
1
1
B3
RESET SYMSWI SWITCH
C3
TURN OFF OPTIONAL FEATURE SW AND ERRSWI
D3
FURTHER CSQ BEING PROC
YES
LG D1
CHKSZCSQ
NO
E3
SYMBOL ERROR IN QUALIFIED SYM
YES
LD A4
ERTPA
NO
LC D4
CHKSZ
LK E2
CDSQCARD B4
LOAD WORK AREA ADDR INTO REG 3 AND MAX LNG OF PARAM INTO REG 4. CLEAR WORK AREA
C4
TURN ON SW TO INDICATE FURTHER QUAL PHASE WAS PROCESSED
D4
PERIOD
YES
PRVTCD D5
UP CARD AREA POINTER BY 2 TO POINT TO FIRST CHAR OF START ADDR
LK G2
CHKCST1
NO
E4
SAVE REGISTER 10
F4
CONTCARD LP
TEST FOR BLANK COLUMN
G4
RESTORE REGISTER 10
H4
PHASE PROCESSED
NO
H5
MOVE PHASE NAME TO SAVE AREA AND STORE LINK REG 10
YES
J5
PHTRRTN LS
PROCESS FURTHER QUAL PHASE
K5
RESTORE LINK REG 10
LF E1
CHKSYM

Chart LN. Test Request Diagnostics (ATLEFC3) (Refer to Chart 11) (Part 13 of 19)

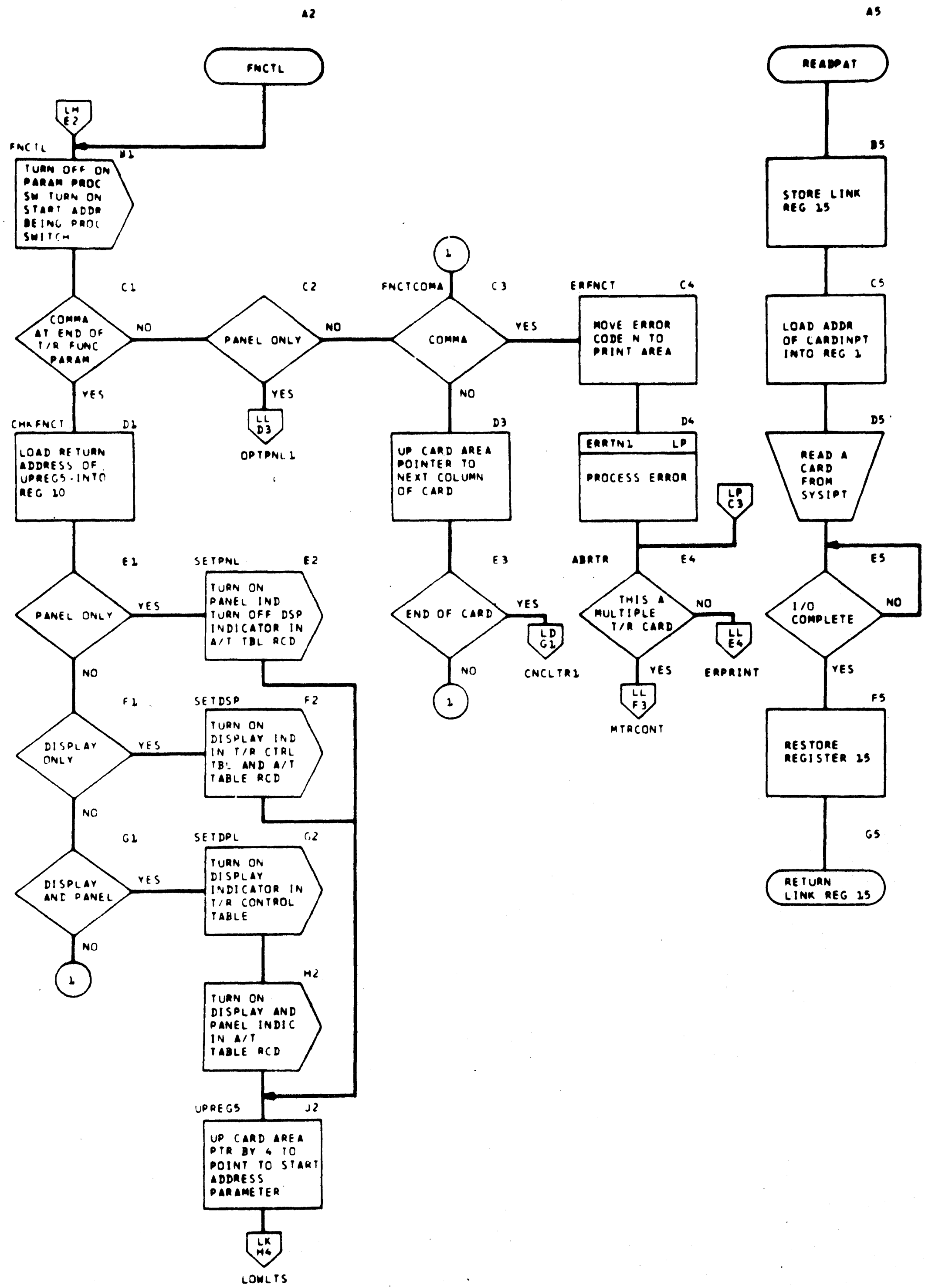
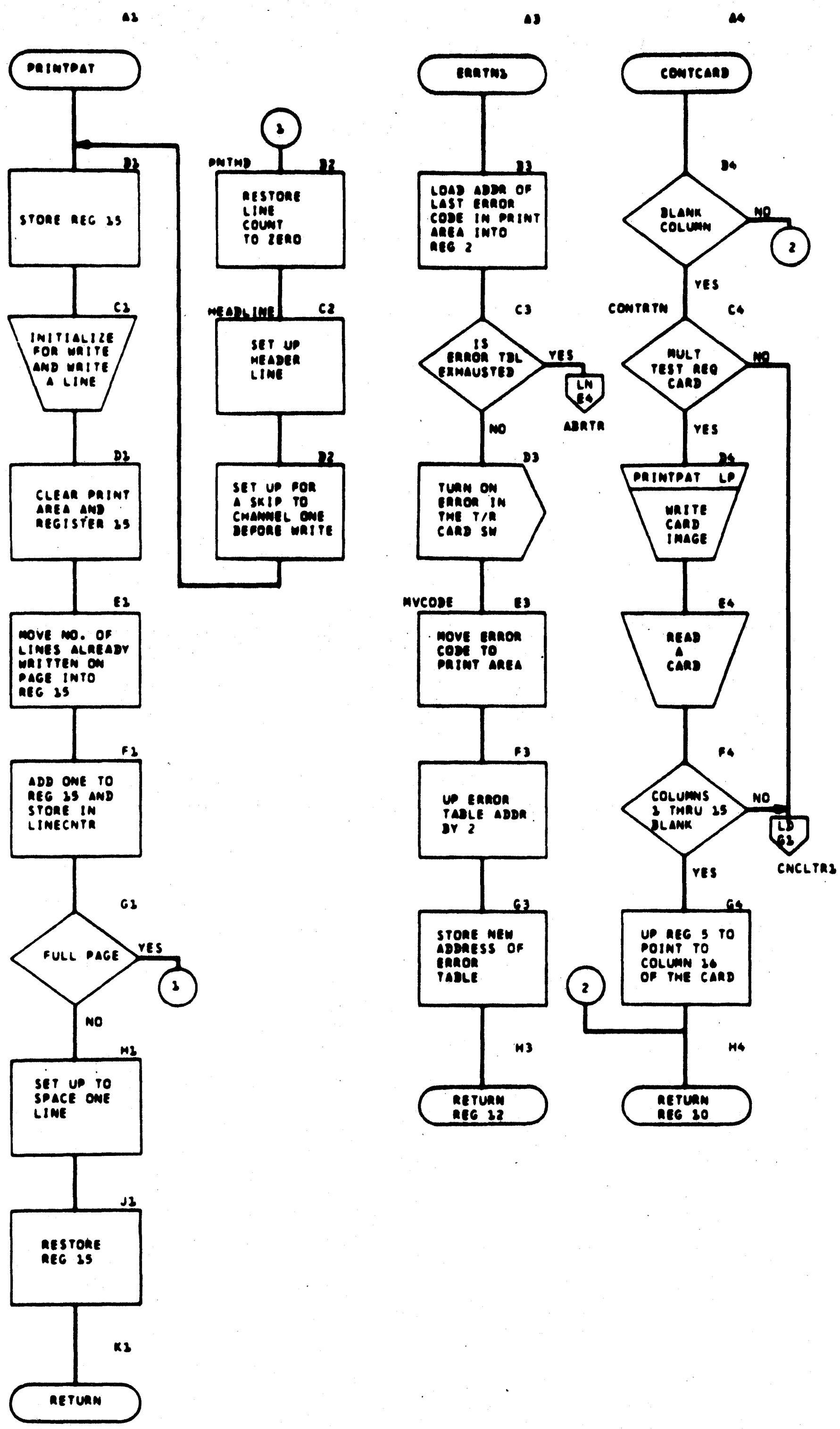


Chart LP. Test Request Diagnostics (ATLEFC3) (Refer to Chart 11) (Part 14 of 19)



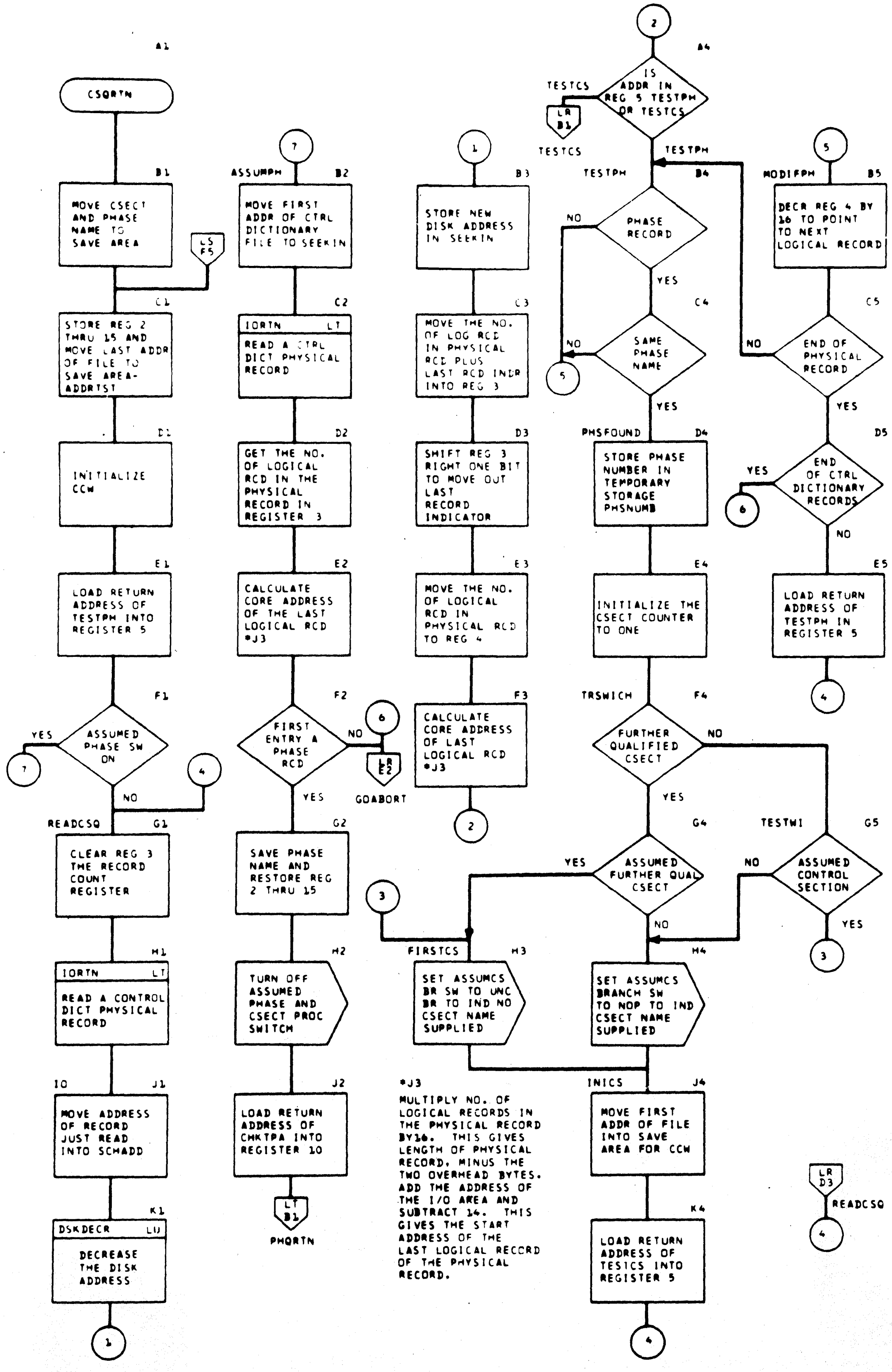
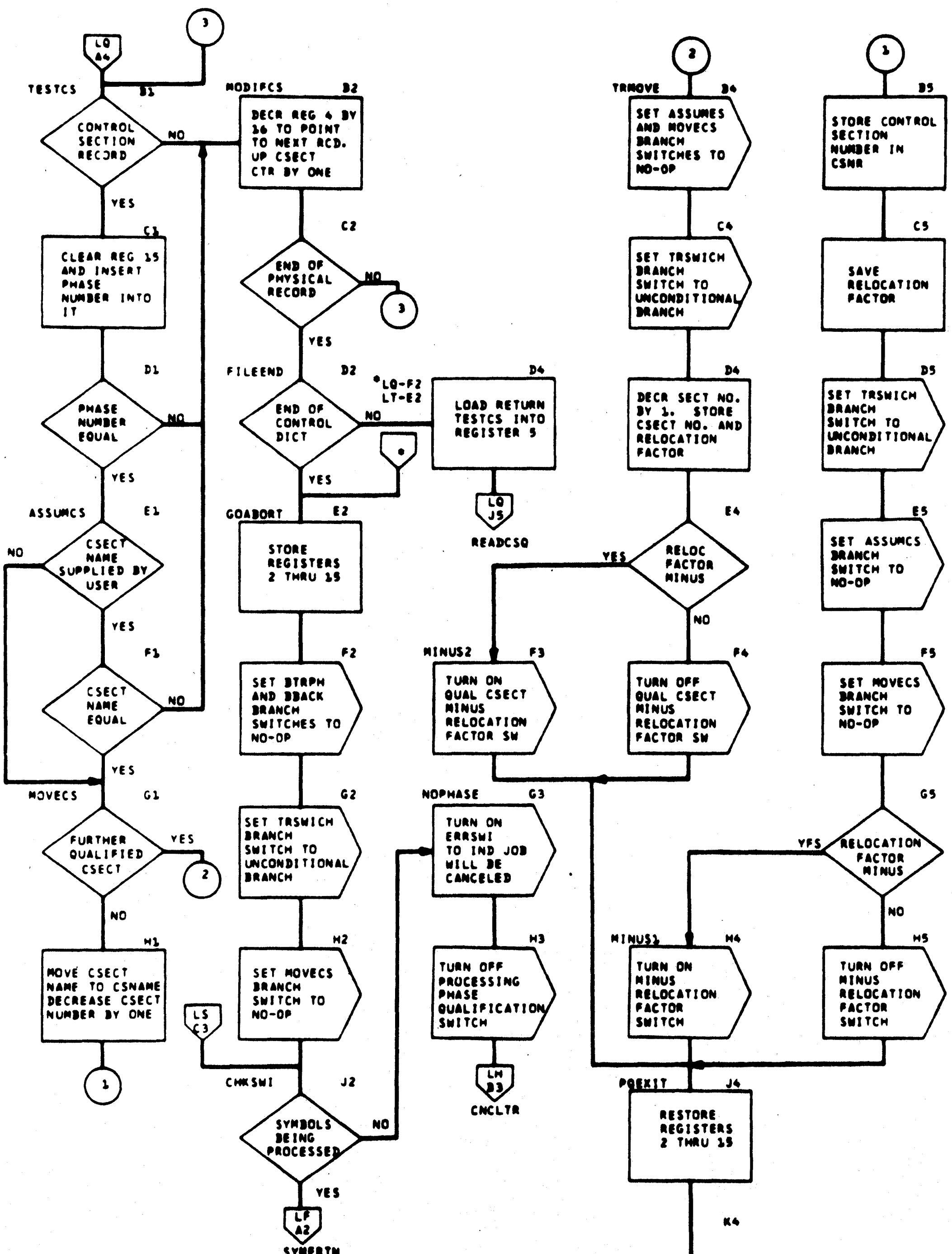

CSQRTN
MOVE CSECT AND PHASE NAME TO SAVE AREA
LS F5
STORE REG 2 THRU 15 AND MOVE LAST ADDR OF FILE TO SAVE AREA-ADDRTST
INITIALIZE CCW
LOAD RETURN ADDRESS OF TESTPH INTO REGISTER 5
ASSUMED PHASE SW ON
YES
NO
READCSQ
CLEAR REG 3 THE RECORD COUNT REGISTER
IORTN LT
READ A CONTROL DICT PHYSICAL RECORD
IO
MOVE ADDRESS OF RECORD JUST READ INTO SCHADD
DSKDECR LU
DECREASE THE DISK ADDRESS
ASSUMPH
MOVE FIRST ADDR OF CTRL DICTIONARY FILE TO SEEKIN
IORTN LT
READ A CTRL DICT PHYSICAL RECORD
GET THE NO. OF LOGICAL RCD IN THE PHYSICAL RECORD IN REGISTER 3
CALCULATE CORE ADDRESS OF THE LAST LOGICAL RCD *J3
FIRST ENTRY A PHASE RCD
LR E2
GOABORT
SAVE PHASE NAME AND RESTORE REG 2 THRU 15
TURN OFF ASSUMED PHASE AND CSECT PROC SWITCH
LOAD RETURN ADDRESS OF CHKTPA INTO REGISTER 10
LT B1
PHQRTN
STORE NEW DISK ADDRESS IN SEEKIN
MOVE THE NO. OF LOG RCD IN PHYSICAL RCD PLUS LAST RCD INDR INTO REG 3
SHIFT REG 3 RIGHT ONE BIT TO MOVE OUT LAST RECORD INDICATOR
MOVE THE NO. OF LOGICAL RCD IN PHYSICAL RCD TO REG 4
CALCULATE CORE ADDRESS OF LAST LOGICAL RCD *J3
FIRSTCS
SET ASSUMCS BR SW TO UNC BR TO IND NO CSECT NAME SUPPLIED
*J3
MULTIPLY NO. OF LOGICAL RECORDS IN THE PHYSICAL RECORD BY16. THIS GIVES LENGTH OF PHYSICAL RECORD, MINUS THE TWO OVERHEAD BYTES. ADD THE ADDRESS OF THE I/O AREA AND SUBTRACT 14. THIS GIVES THE START ADDRESS OF THE LAST LOGICAL RECORD OF THE PHYSICAL RECORD.
TESTCS
IS ADDR IN REG 5 TESTPH OR TESTCS
LR B1
TESTPH
PHASE RECORD
SAME PHASE NAME
PHSFOUND
STORE PHASE NUMBER IN TEMPORARY STORAGE PHSNUMB
INITIALIZE THE CSECT COUNTER TO ONE
TRSWICH
FURTHER QUALIFIED CSECT
ASSUMED FURTHER QUAL CSECT
SET ASSUMCS BRANCH SW TO NOP TO IND CSECT NAME SUPPLIED
INICS
MOVE FIRST ADDR OF FILE INTO SAVE AREA FOR CCW
LOAD RETURN ADDRESS OF TESICS INTO REGISTER 5
MODIFPH
DECR REG 4 BY 16 TO POINT TO NEXT LOGICAL RECORD
END OF PHYSICAL RECORD
END OF CTRL DICTIONARY RECORDS
LOAD RETURN ADDRESS OF TESTPH IN REGISTER 5
TESTWI
ASSUMED CONTROL SECTION
LR D3
READCSQ

Chart LR. Test Request Diagnostics (ATLEFC3) (Refer to Chart 11) (Part 16 of 19)



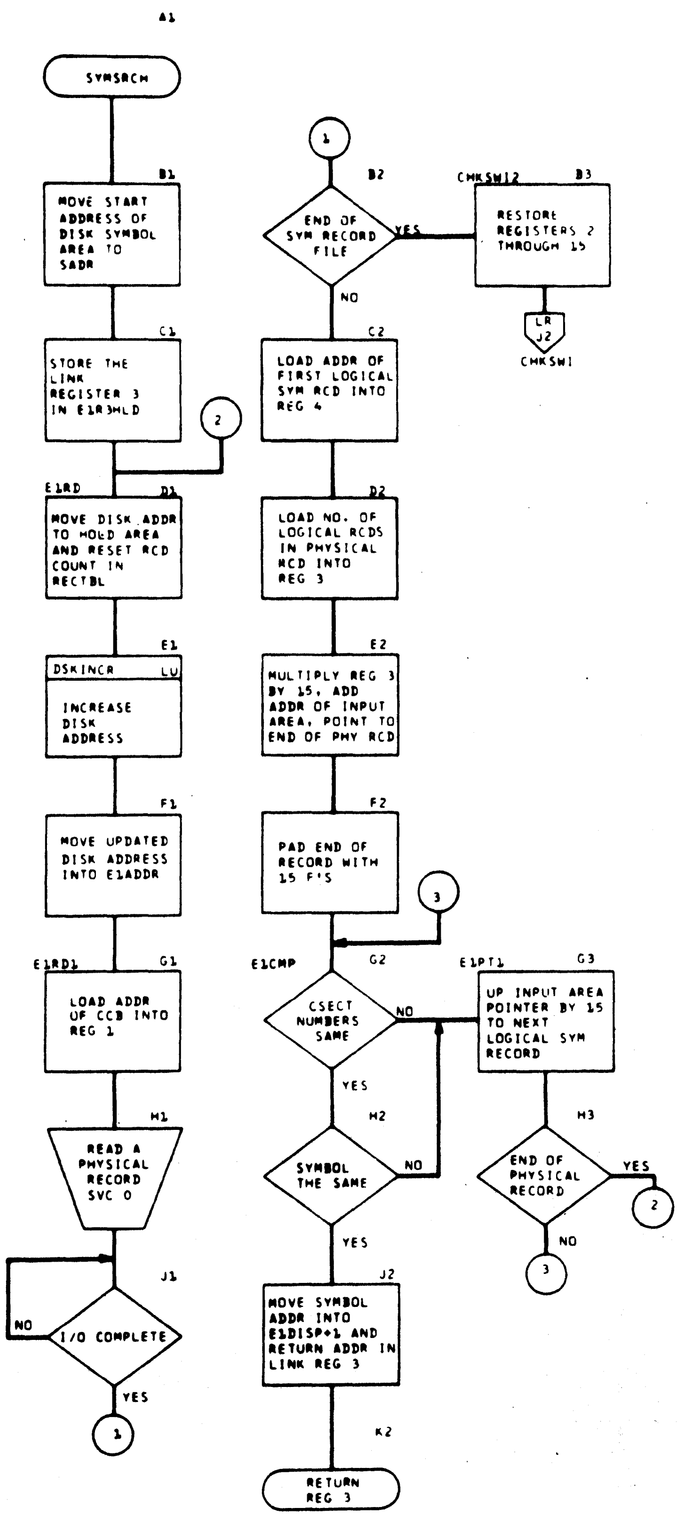
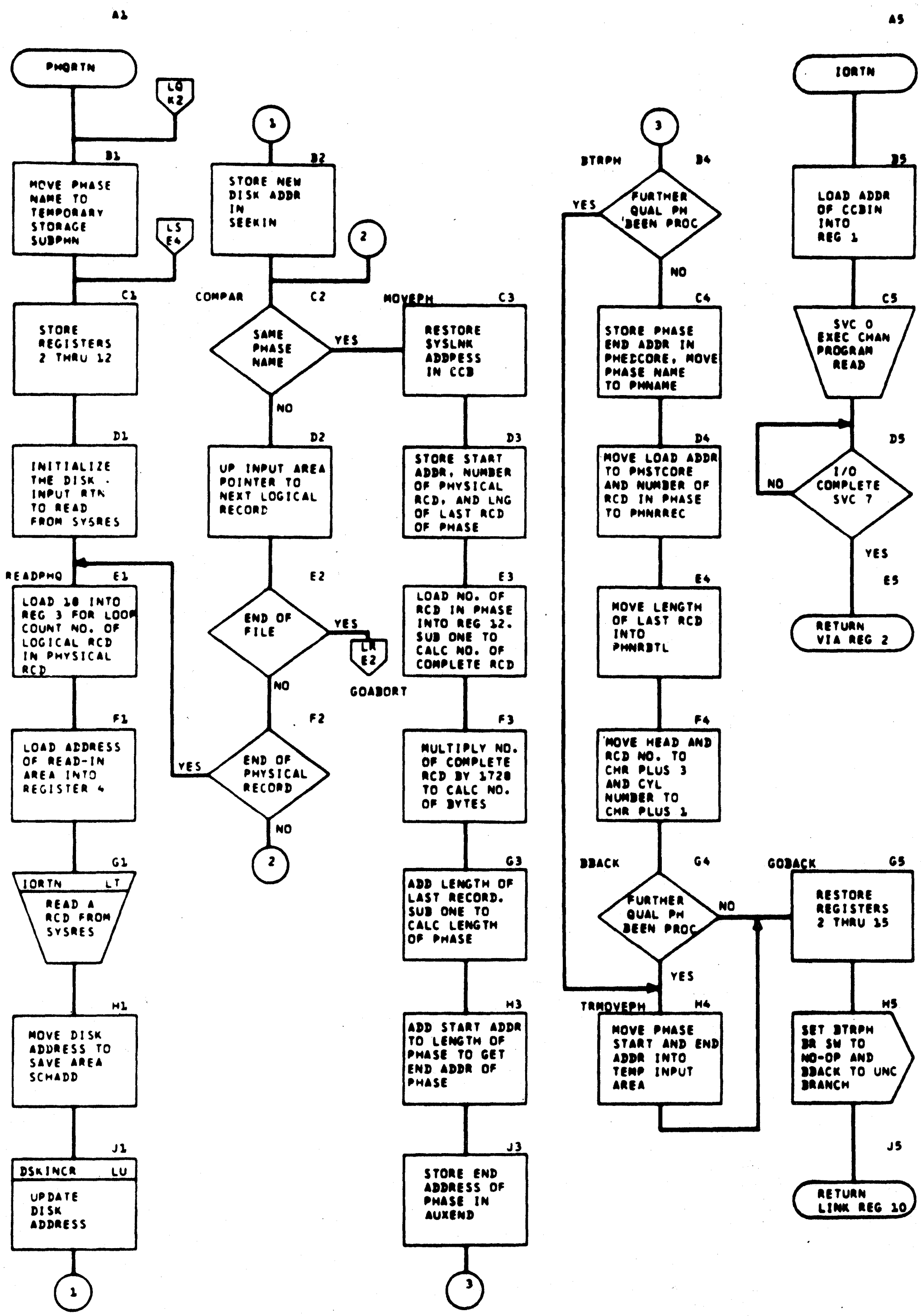
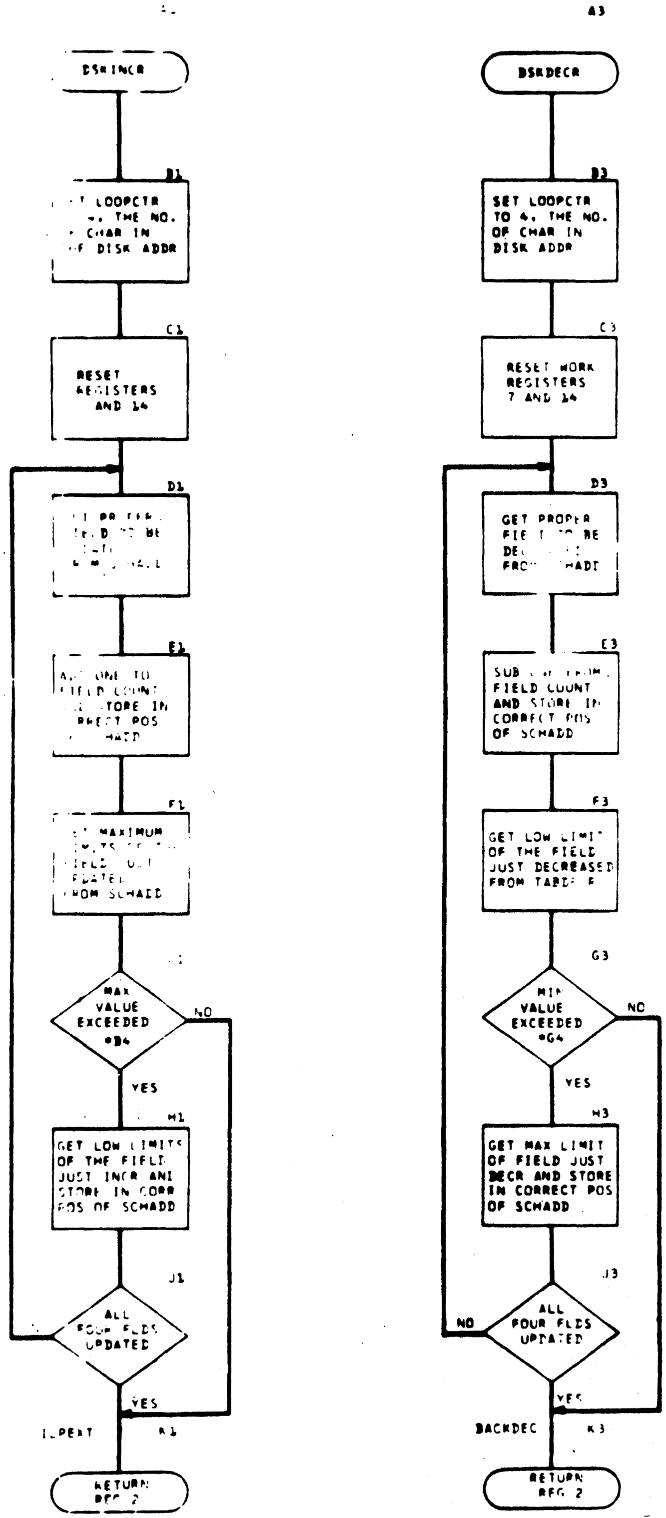

SYMSRCH
MOVE START ADDRESS OF DISK SYMBOL AREA TO SADR
STORE THE LINK REGISTER 3 IN E1R3HLD
E1RD
MOVE DISK ADDR TO HOLD AREA AND RESET RCD COUNT IN RECTBL
DSKINCR
INCREASE DISK ADDRESS
MOVE UPDATED DISK ADDRESS INTO E1ADDR
E1RD1
LOAD ADDR OF CCB INTO REG 1
READ A PHYSICAL RECORD SVC 0
I/O COMPLETE
END OF SYM RECORD FILE
CHKSW12
RESTORE REGISTERS 2 THROUGH 15
CHKSW1
LOAD ADDR OF FIRST LOGICAL SYM RCD INTO REG 4
LOAD NO. OF LOGICAL RCDS IN PHYSICAL RCD INTO REG 3
MULTIPLY REG 3 BY 15, ADD ADDR OF INPUT AREA, POINT TO END OF PHY RCD
PAD END OF RECORD WITH 15 F'S
E1CMP
CSECT NUMBERS SAME
E1PT1
UP INPUT AREA POINTER BY 15 TO NEXT LOGICAL SYM RECORD
SYMBOL THE SAME
END OF PHYSICAL RECORD
MOVE SYMBOL ADDR INTO E1DISP+1 AND RETURN ADDR IN LINK REG 3
RETURN REG 3
PHTRRTN
MOVE PHASE NAME TO SAVE AREA
SET BTRPH BRANCH SWITCH TO UNCOND BRANCH
SET BBACK BRANCH SWITCH TO NO-OP
PHQRTN+6
CSTRRTN
MOVE PHASE NAME TO SAVE AREA
MOVE CSECT NAME TO SAVE AREA
SET TRSWICH BRANCH SWITCH TO NO-OP
SET MOVECS BRANCH SWITCH TO UNCOND BRANCH
CSQRTN+12

Chart LT. Test Request Diagnostics (ATLEFC3) (Refer to Chart 11) (Part 18 of 19)





*B4
RECTBL IS AN 8-BYTE FIELD IN THE FOLLOWING FORMAT: CCHRCCHR WHERE C IS CYLINDER NUMBER, H IS HEAD NUMBER, AND R IS RECORD NUMBER. THE FIRST 4 BYTES OF RECTBL FIELD, CCHR, CONTAIN THE MAX VALUES OF THE CYL, HEAD AND RECORD. THE NEXT 4 BYTES, CCHR, CONTAIN THE LOW LIMIT VALUE THAT THE CYL, HEAD AND RECORD ARE RESET TO WHEN THE MAXIMUM VALUE HAS BEEN REACHED FOR THEIR RESPECTIVE FIELDS.

*G4
TABDECR IS AN 8-BYTE FIELD IN THE FOLLOWING FORMAT: CCHRCCHR WHERE C IS CYLINDER NO., H IS HEAD NUMBER, AND R IS RECORD NUMBER. THE FIRST 4 BYTES OF TABDECR FIELD, CCHR, CONTAIN THE MINIMUM VALUE OF THE CYL, HEAD AND RECORD. THE NEXT 4 BYTES, CCHR, CONTAINS THE HIGH LIMIT VALUE THAT THE CYL, HEAD AND RECORD ARE RESET TO WHEN THE MINIMUM VALUE HAS BEEN REACHED FOR THEIR RESPECTIVE FIELDS.

Chart MA. Build Test Request Table Entry (ATLEFC4) (Refer to Chart 11) (Part 1 of 22)

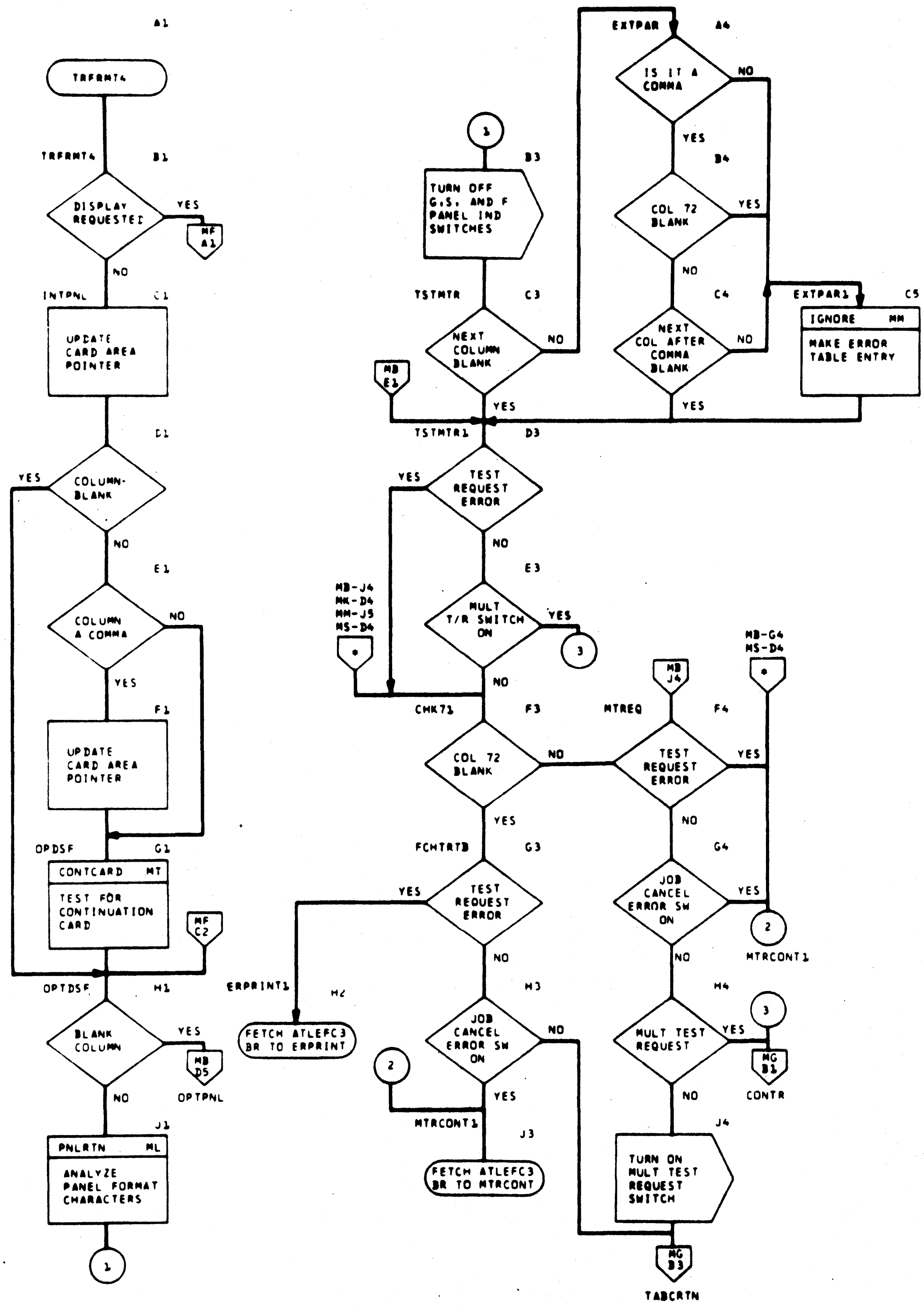
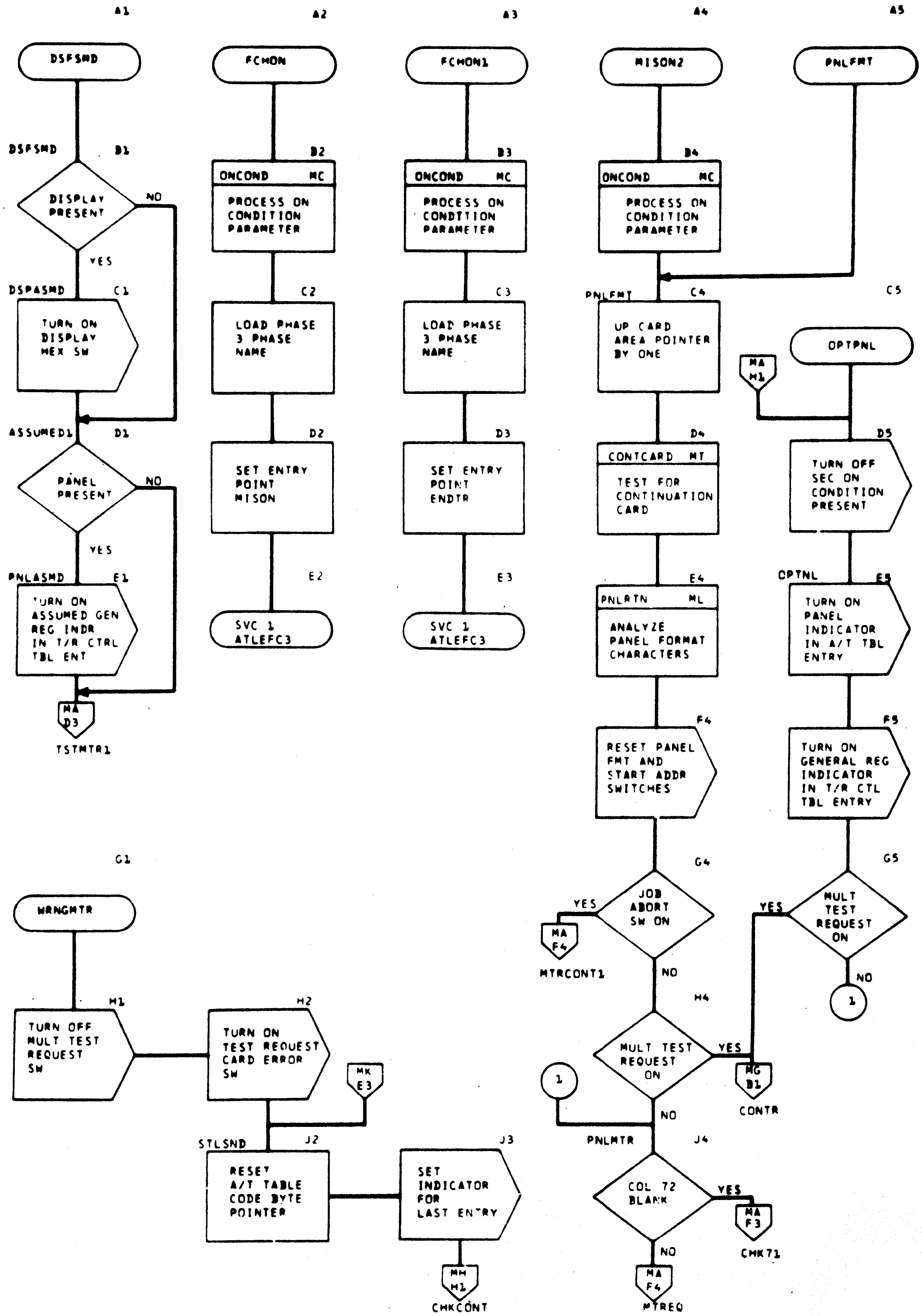
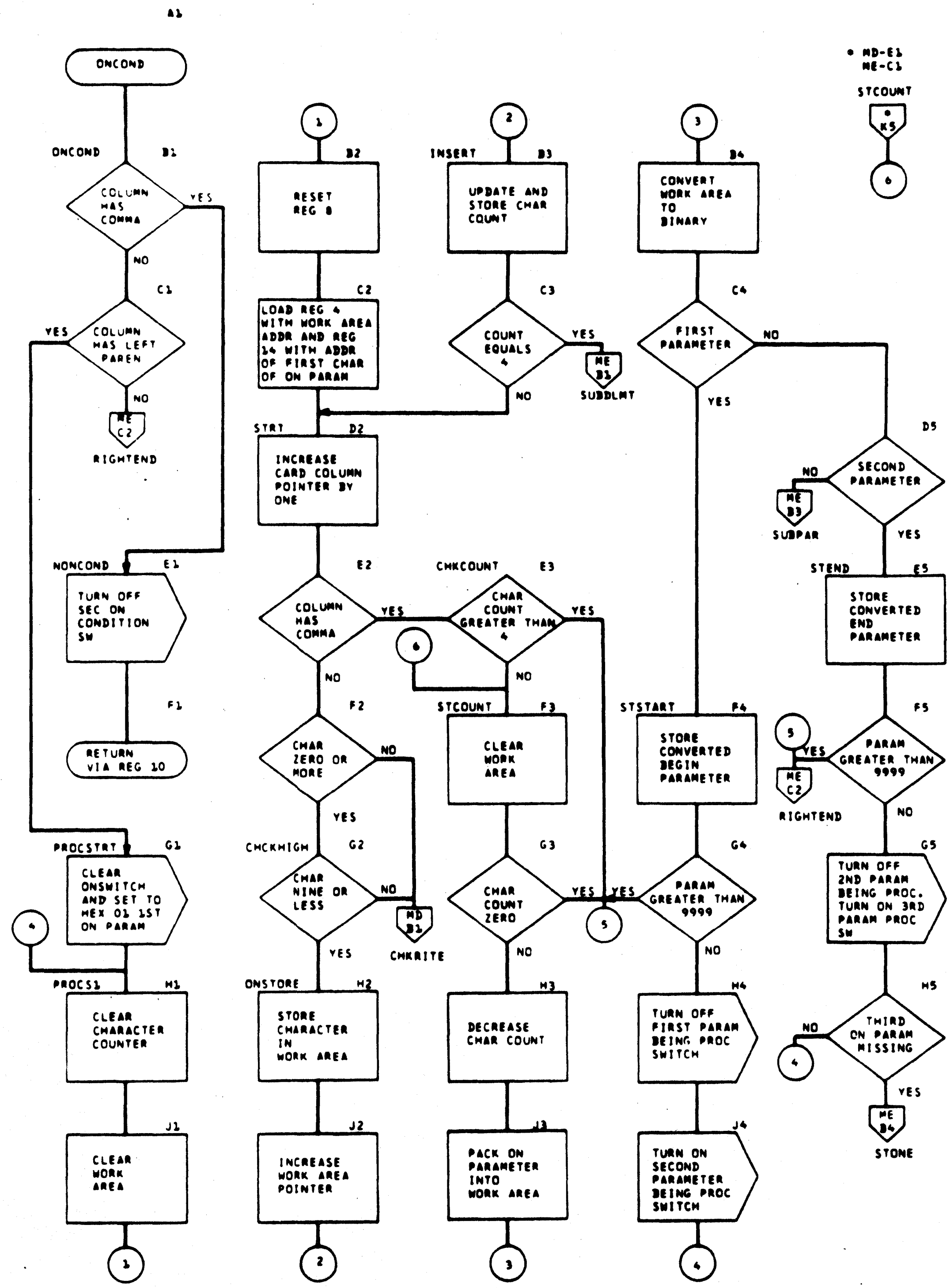
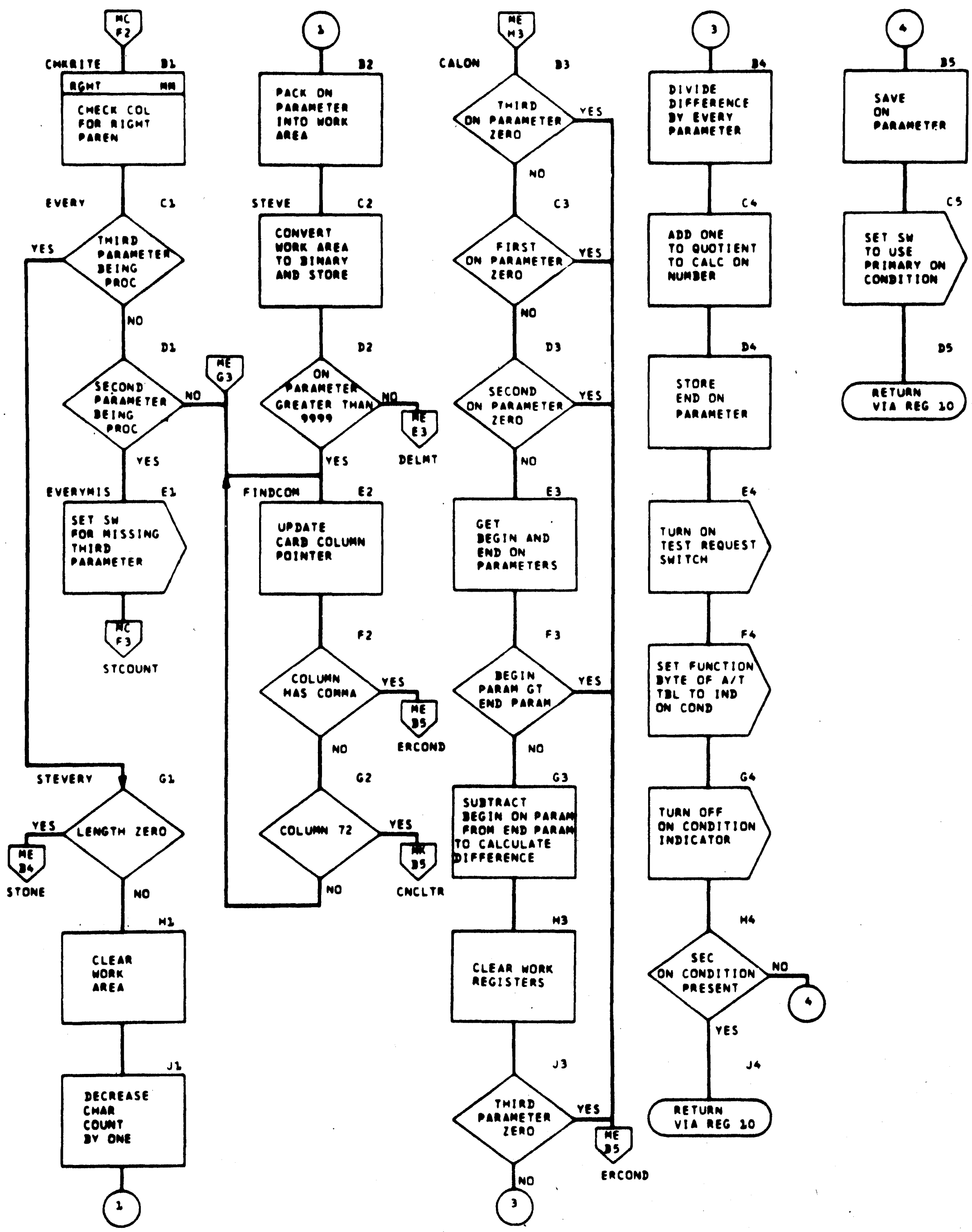


Chart MB. Build Test Request Table Entry (ATLEFC4) (Refer to Chart 11) (Part 2 of 22)




ONCOND
COLUMN HAS COMMA
COLUMN HAS LEFT PAREN
ME C2 RIGHTEND
TURN OFF SEC ON CONDITION SW
RETURN VIA REG 10
CLEAR ONSWITCH AND SET TO HEX 01 1ST ON PARAM
CLEAR CHARACTER COUNTER
CLEAR WORK AREA
RESET REG 8
LOAD REG 4 WITH WORK AREA ADDR AND REG 14 WITH ADDR OF FIRST CHAR OF ON PARAM
INCREASE CARD COLUMN POINTER BY ONE
CHAR ZERO OR MORE
CHAR NINE OR LESS
MD B1 CHKRITE
STORE CHARACTER IN WORK AREA
INCREASE WORK AREA POINTER
UPDATE AND STORE CHAR COUNT
COUNT EQUALS 4
ME B1 SUBDLMT
CHAR COUNT GREATER THAN 4
CLEAR WORK AREA
CHAR COUNT ZERO
DECREASE CHAR COUNT
PACK ON PARAMETER INTO WORK AREA
CONVERT WORK AREA TO BINARY
FIRST PARAMETER
STORE CONVERTED BEGIN PARAMETER
PARAM GREATER THAN 9999
TURN OFF FIRST PARAM BEING PROC SWITCH
TURN ON SECOND PARAMETER BEING PROC SWITCH
MD-E1 ME-C1
STCOUNT
SECOND PARAMETER
ME B3 SUBPAR
STORE CONVERTED END PARAMETER
PARAM GREATER THAN 9999
ME C2 RIGHTEND
TURN OFF 2ND PARAM BEING PROC. TURN ON 3RD PARAM PROC SW
THIRD ON PARAM MISSING
ME B4 STONE



CHKRITE
B1
RGHT
MR
CHECK COL FOR RIGHT PAREN
EVERY
C1
THIRD PARAMETER BEING PROC
D1
SECOND PARAMETER BEING PROC
EVERYMIS
E1
SET SW FOR MISSING THIRD PARAMETER
STCOUNT
STEVERY
G1
LENGTH ZERO
STONE
H1
CLEAR WORK AREA
J1
DECREASE CHAR COUNT BY ONE
B2
PACK ON PARAMETER INTO WORK AREA
STEVE
C2
CONVERT WORK AREA TO BINARY AND STORE
D2
ON PARAMETER GREATER THAN 9999
DELMT
FINDCOM
E2
UPDATE CARD COLUMN POINTER
F2
COLUMN HAS COMMA
ERCOND
G2
COLUMN 72
CNCLTR
CALON
B3
THIRD ON PARAMETER ZERO
C3
FIRST ON PARAMETER ZERO
D3
SECOND ON PARAMETER ZERO
E3
GET BEGIN AND END ON PARAMETERS
F3
BEGIN PARAM GT END PARAM
G3
SUBTRACT BEGIN ON PARAM FROM END PARAM TO CALCULATE DIFFERENCE
H3
CLEAR WORK REGISTERS
J3
THIRD PARAMETER ZERO
ERCOND
B4
DIVIDE DIFFERENCE BY EVERY PARAMETER
C4
ADD ONE TO QUOTIENT TO CALC ON NUMBER
D4
STORE END ON PARAMETER
E4
TURN ON TEST REQUEST SWITCH
F4
SET FUNCTION BYTE OF A/T TBL TO IND ON COND
G4
TURN OFF ON CONDITION INDICATOR
H4
SEC ON CONDITION PRESENT
J4
RETURN VIA REG 10
B5
SAVE ON PARAMETER
C5
SET SW TO USE PRIMARY ON CONDITION
D5
RETURN VIA REG 10
YES
NO

Chart ME. Build Test Request Table Entry (ATLEFC4) (Refer to Chart 11) (Part 5 of 22)

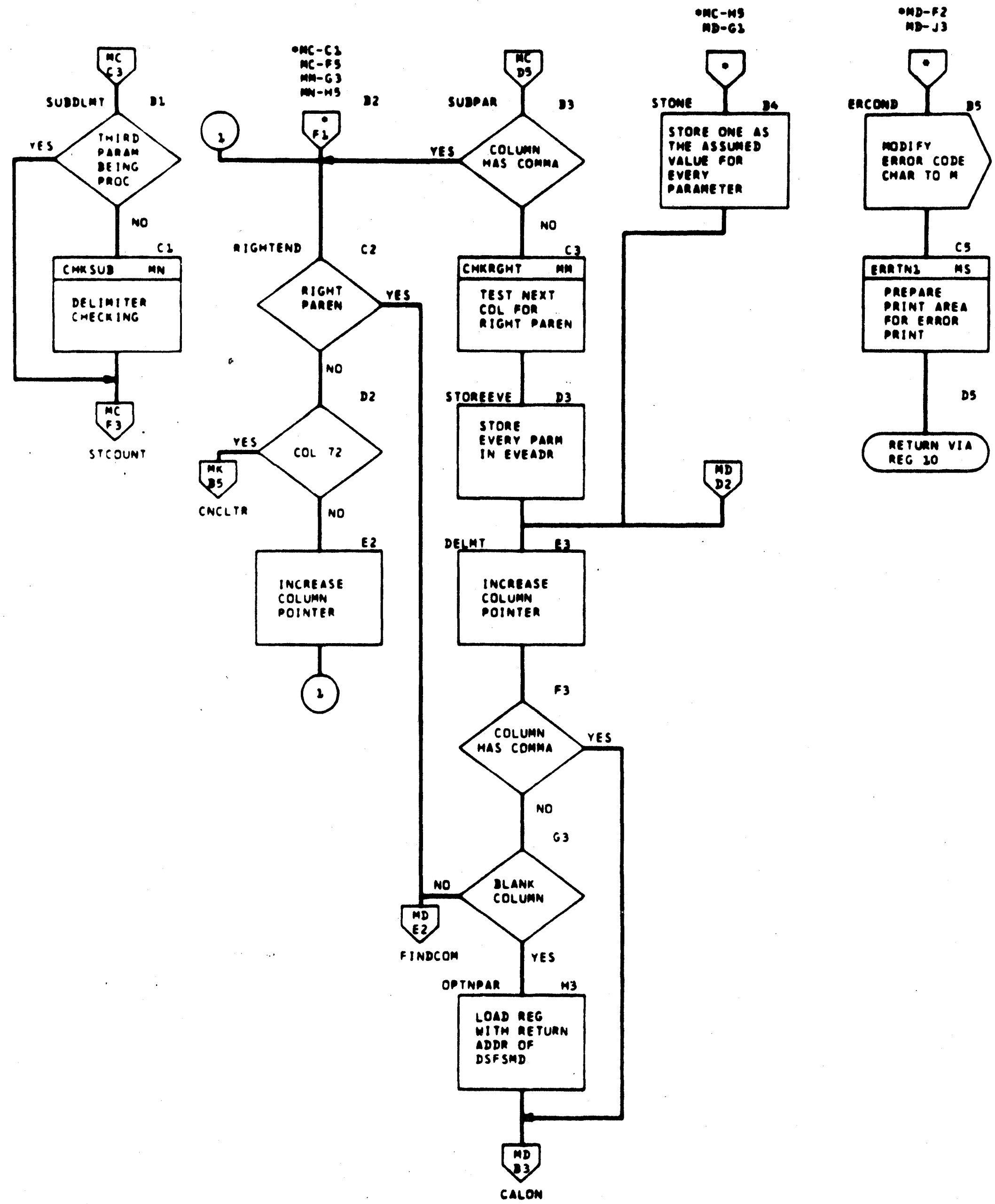


Chart MF. Build Test Request Table Entry (ATLEFC4) (Refer to Chart 11) (Part 6 of 22)

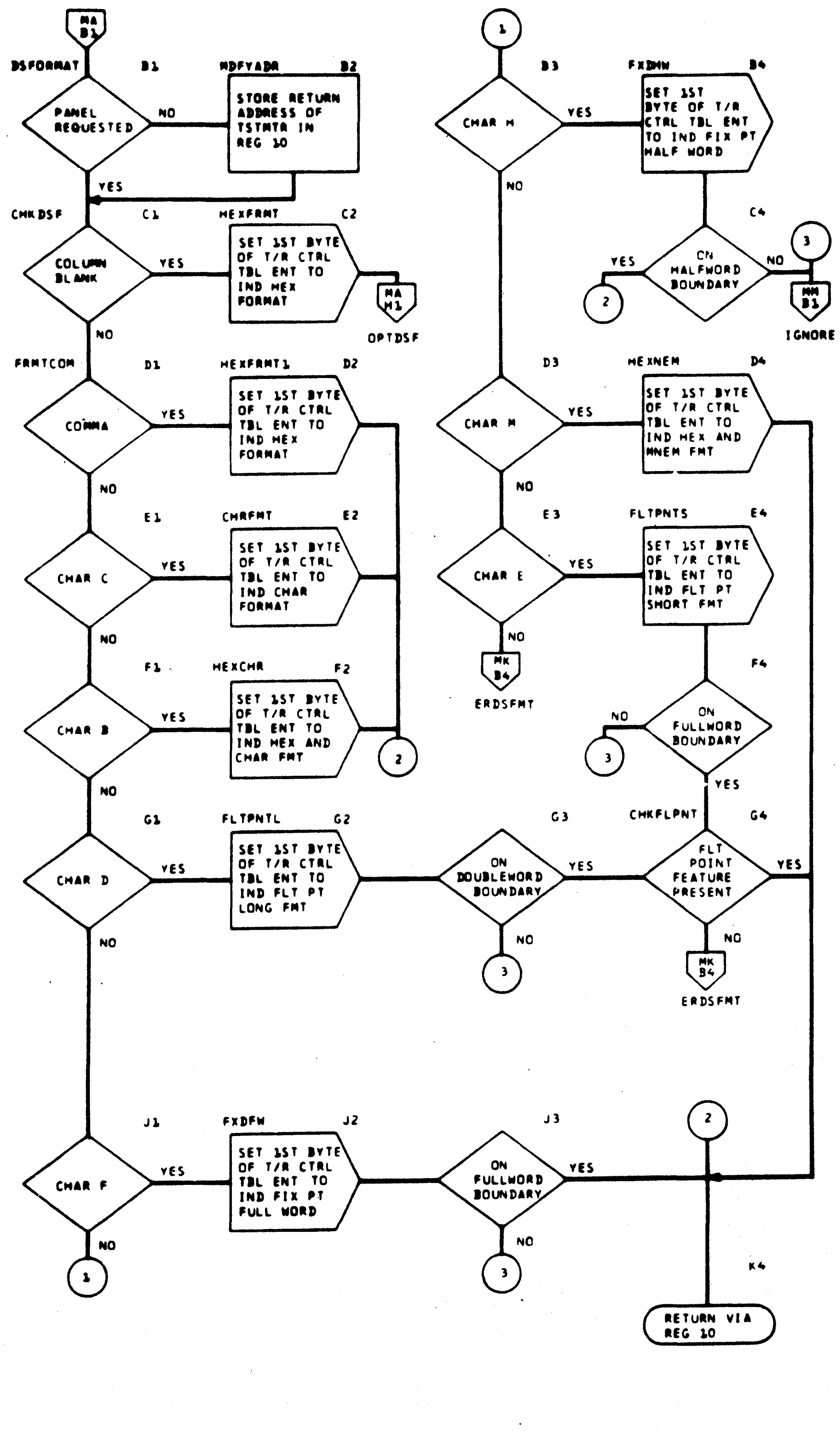
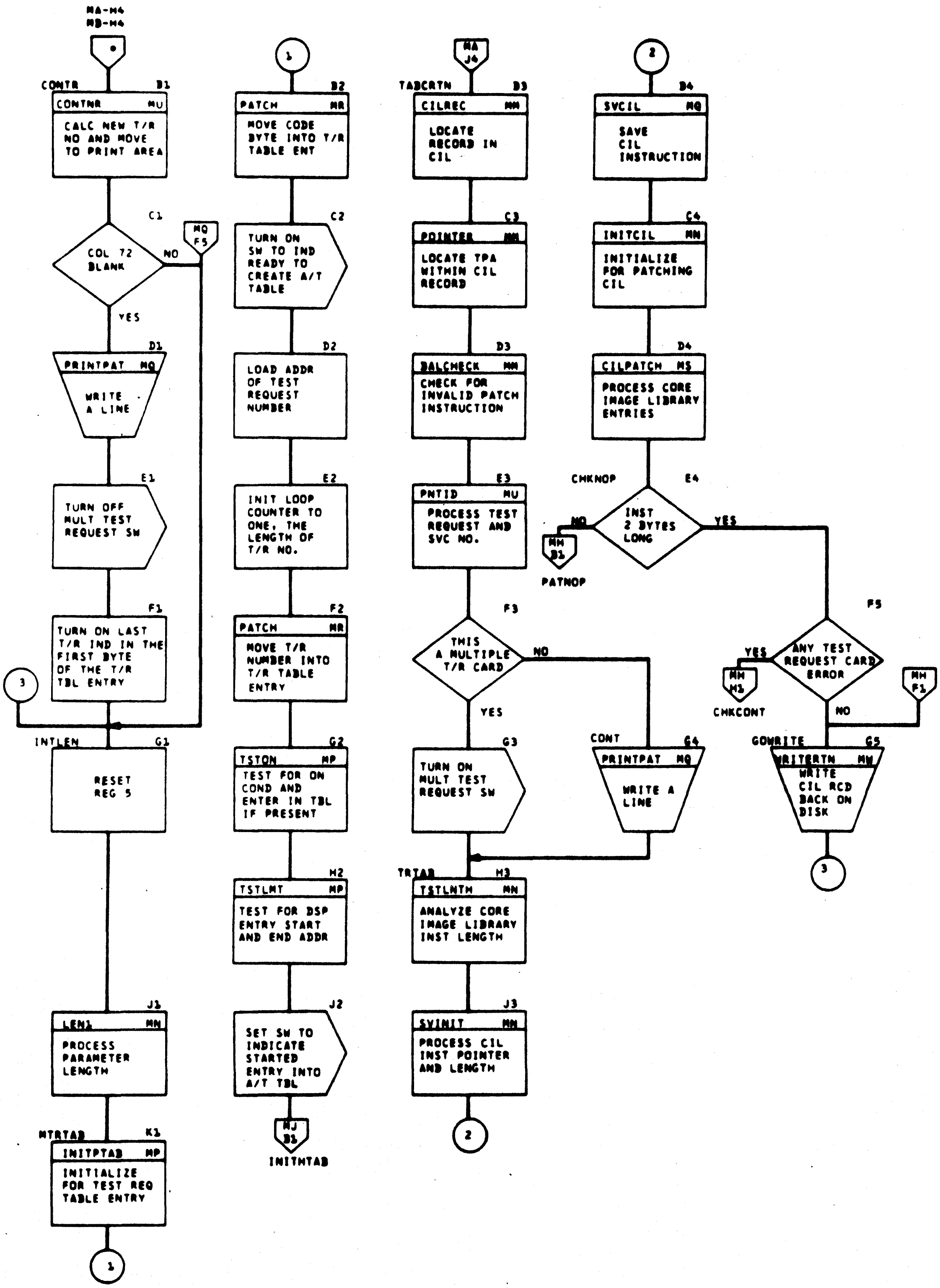


Chart MG. Build Test Request Table Entry (ATLEFC4) (Refer to Chart 11) (Part 7 of 22)



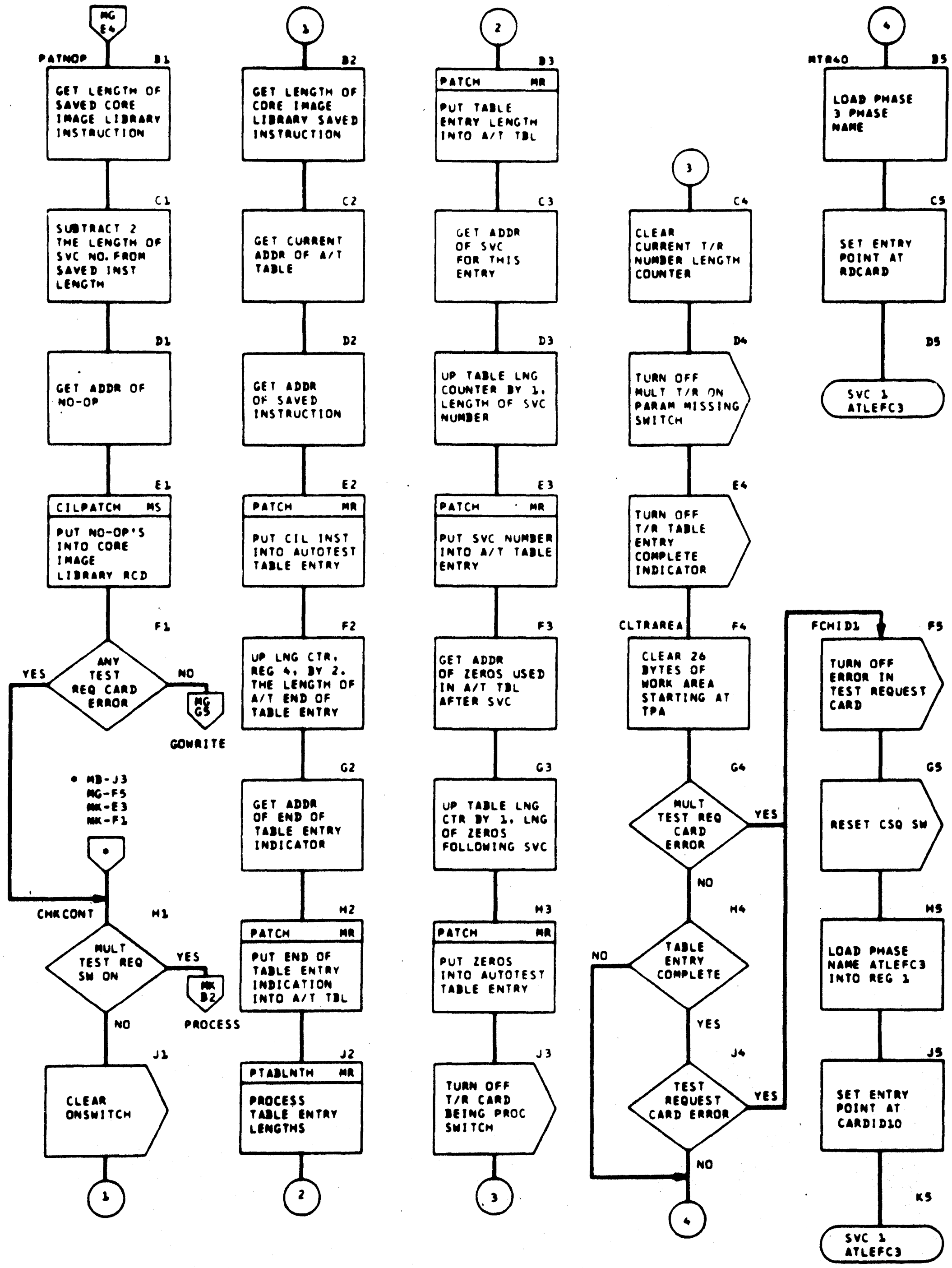

MG E4
PATNOP B1
GET LENGTH OF SAVED CORE IMAGE LIBRARY INSTRUCTION
C1
SUBTRACT 2 THE LENGTH OF SVC NO. FROM SAVED INST LENGTH
D1
GET ADDR OF NO-OP
E1
CILPATCH MS
PUT NO-OP'S INTO CORE IMAGE LIBRARY RCD
F1
ANY TEST REQ CARD ERROR
YES
NO
MG G5
GOWRITE
MB-J3
MG-F5
MK-E3
MK-F1
CHKCONT H1
MULT TEST REQ SW ON
YES
MK B2
PROCESS
NO
J1
CLEAR ONSWITCH
1
1
B2
GET LENGTH OF CORE IMAGE LIBRARY SAVED INSTRUCTION
C2
GET CURRENT ADDR OF A/T TABLE
D2
GET ADDR OF SAVED INSTRUCTION
E2
PATCH MR
PUT CIL INST INTO AUTOTEST TABLE ENTRY
F2
UP LNG CTR, REG 4, BY 2. THE LENGTH OF A/T END OF TABLE ENTRY
G2
GET ADDR OF END OF TABLE ENTRY INDICATOR
H2
PATCH MR
PUT END OF TABLE ENTRY INDICATION INTO A/T TBL
J2
PTABLNTH MR
PROCESS TABLE ENTRY LENGTHS
2
2
B3
PATCH MR
PUT TABLE ENTRY LENGTH INTO A/T TBL
C3
GET ADDR OF SVC FOR THIS ENTRY
D3
UP TABLE LNG COUNTER BY 1, LENGTH OF SVC NUMBER
E3
PATCH MR
PUT SVC NUMBER INTO A/T TABLE ENTRY
F3
GET ADDR OF ZEROS USED IN A/T TBL AFTER SVC
G3
UP TABLE LNG CTR BY 1, LNG OF ZEROS FOLLOWING SVC
H3
PATCH MR
PUT ZEROS INTO AUTOTEST TABLE ENTRY
J3
TURN OFF T/R CARD BEING PROC SWITCH
3
3
C4
CLEAR CURRENT T/R NUMBER LENGTH COUNTER
D4
TURN OFF MULT T/R ON PARAM MISSING SWITCH
E4
TURN OFF T/R TABLE ENTRY COMPLETE INDICATOR
CLTRAREA F4
CLEAR 26 BYTES OF WORK AREA STARTING AT TPA
G4
MULT TEST REQ CARD ERROR
YES
NO
H4
TABLE ENTRY COMPLETE
NO
YES
J4
TEST REQUEST CARD ERROR
YES
NO
4
4
MTR40 B5
LOAD PHASE 3 PHASE NAME
C5
SET ENTRY POINT AT RDCARD
D5
SVC 1 ATLEFC3
FCHID1 F5
TURN OFF ERROR IN TEST REQUEST CARD
G5
RESET CSQ SW
H5
LOAD PHASE NAME ATLEFC3 INTO REG 1
J5
SET ENTRY POINT AT CARDID10
K5
SVC 1 ATLEFC3

Chart MJ. Build Test Request Table Entry (ATLEFC4) (Refer to Chart 11) (Part 9 of 22)

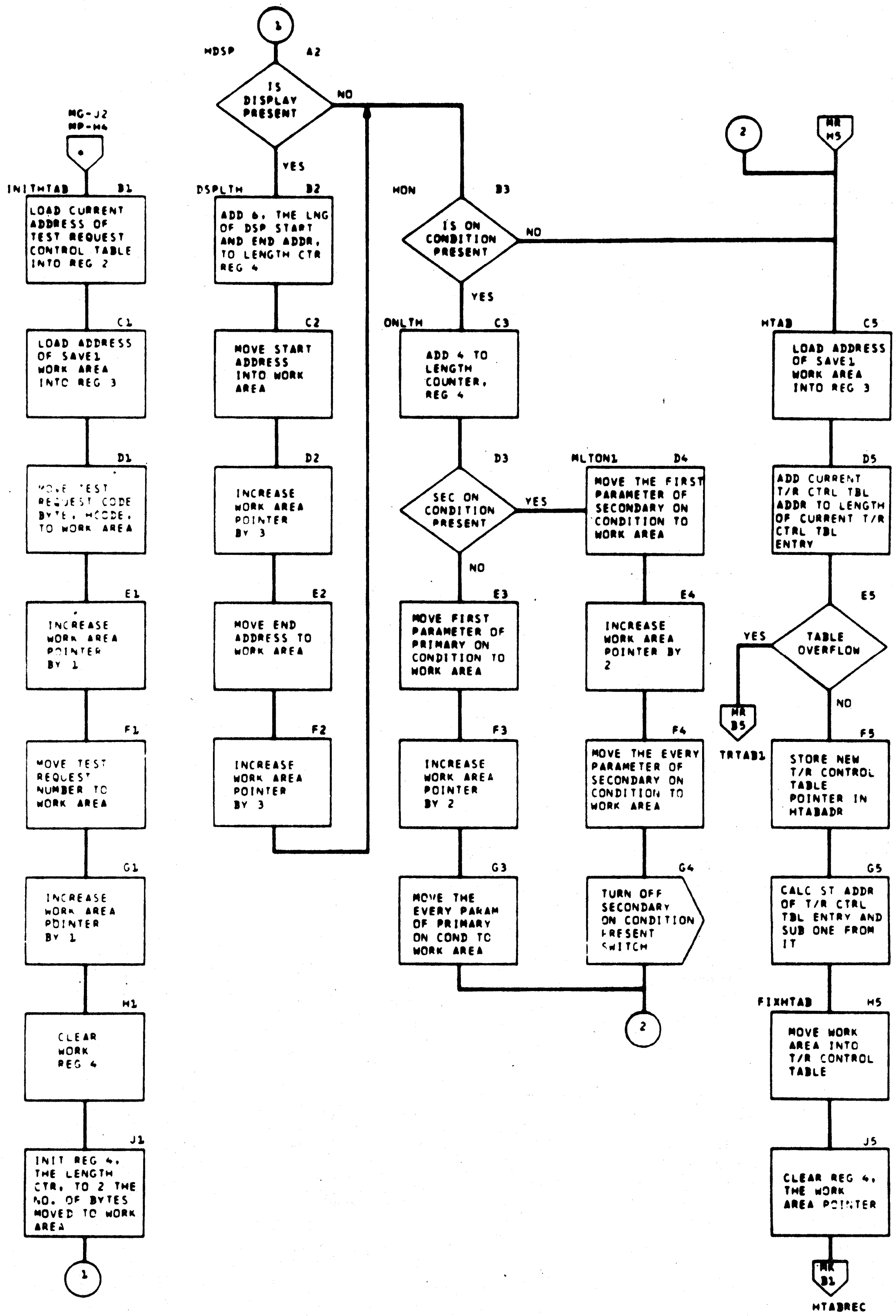


Chart MK. Build Test Request Table Entry (ATLEFC4) (Refer to Chart 11) (Part 10 of 22)

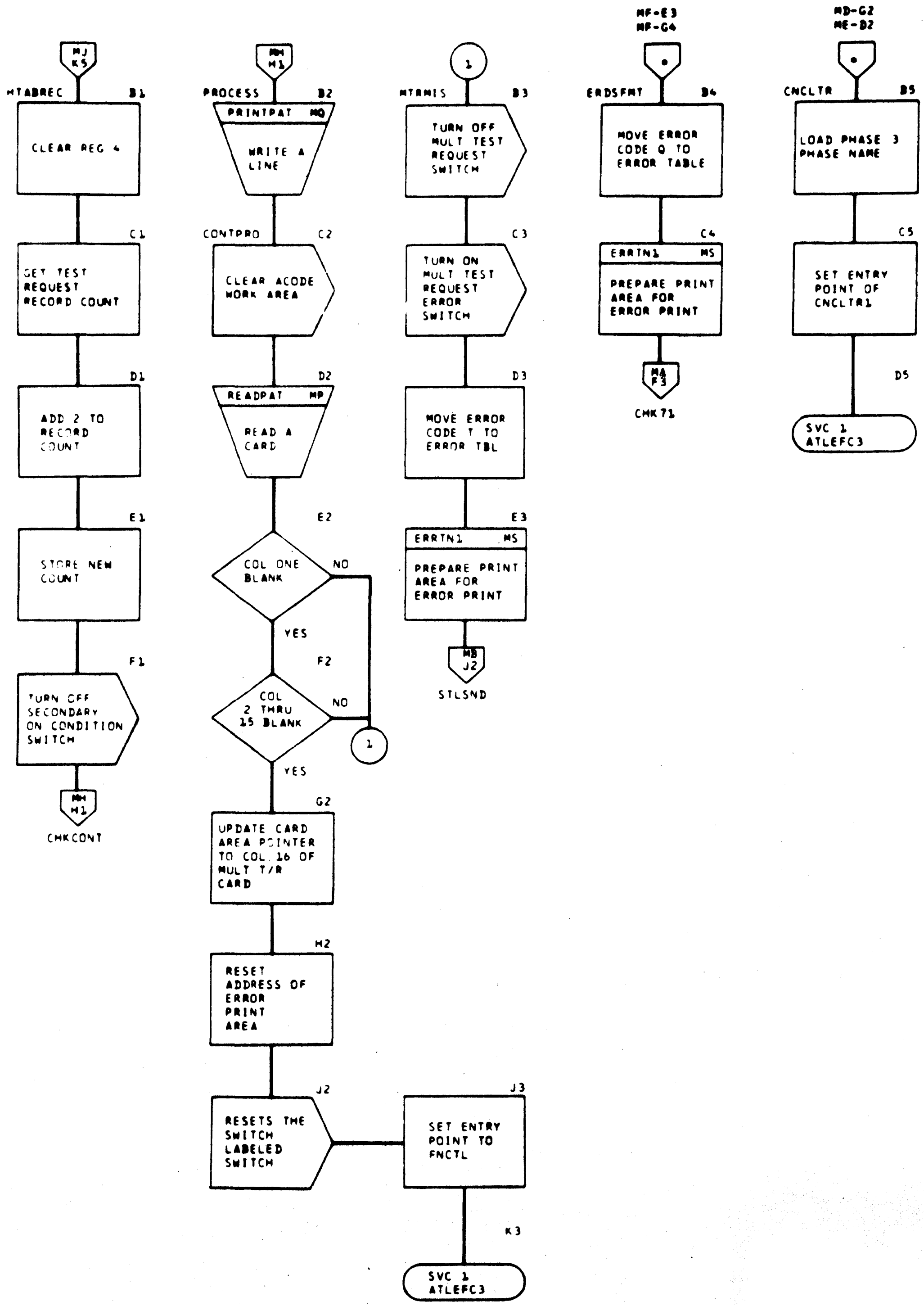


Chart ML. Build Test Request Table Entry (ATLEFC4) (Refer to Chart 11) (Part 11 of 22)

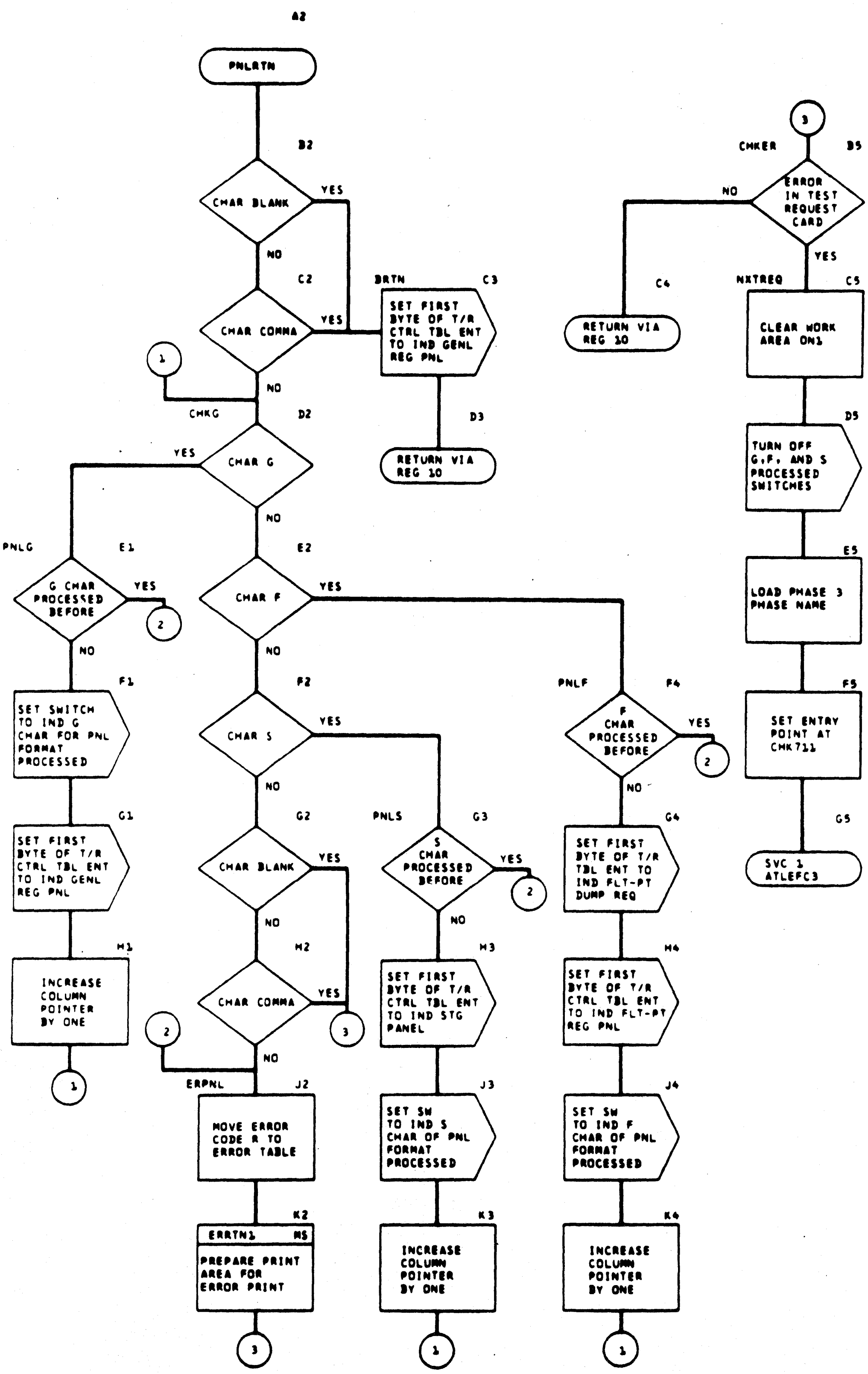


Chart MM. Build Test Request Table Entry (ATLEFC4) (Refer to Chart 11) (Part 12 of 22)

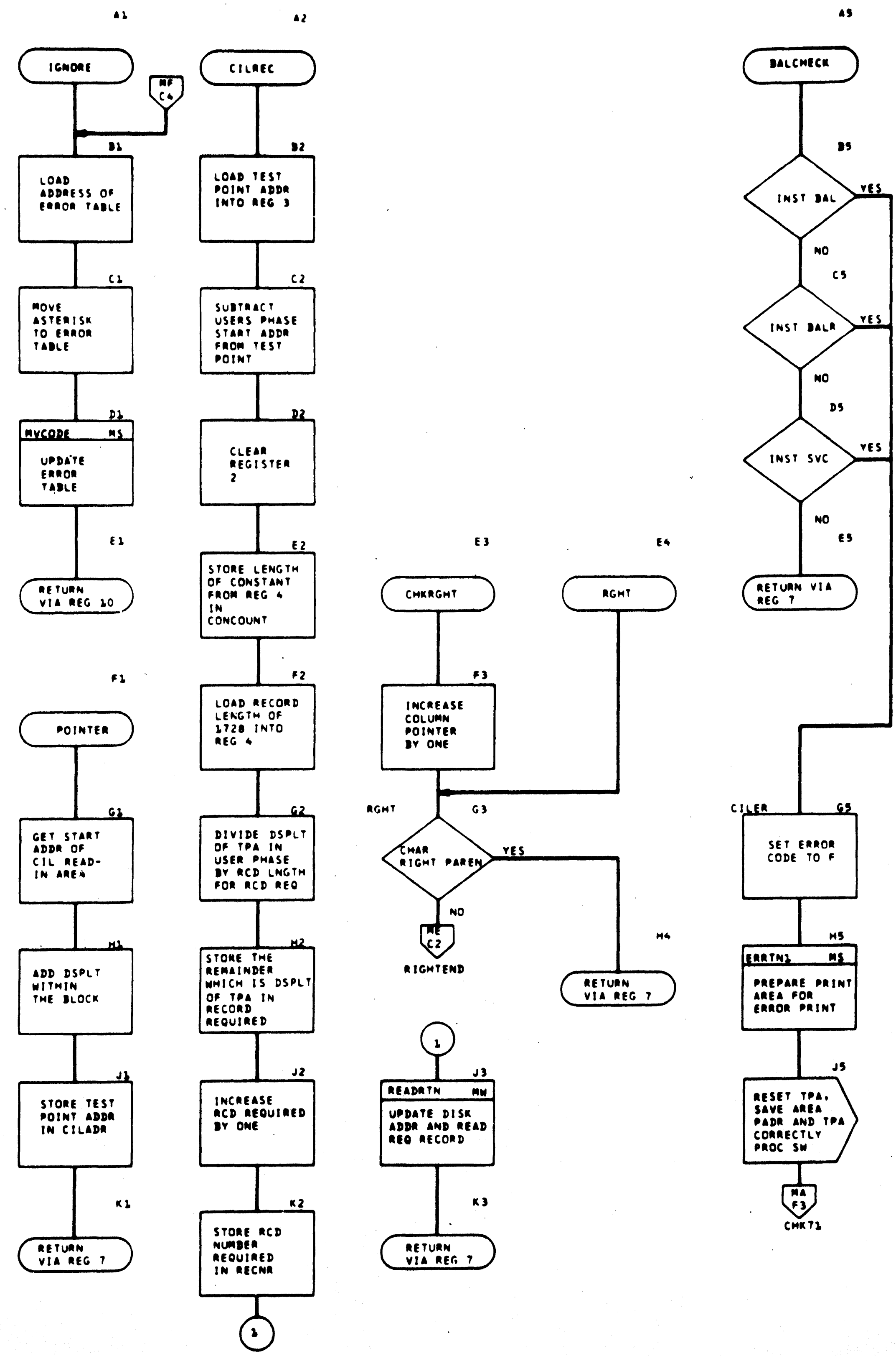


Chart MN. Build Test Request Table Entry (ATLEFC4) (Refer to Chart 11) (Part 13 of 22)

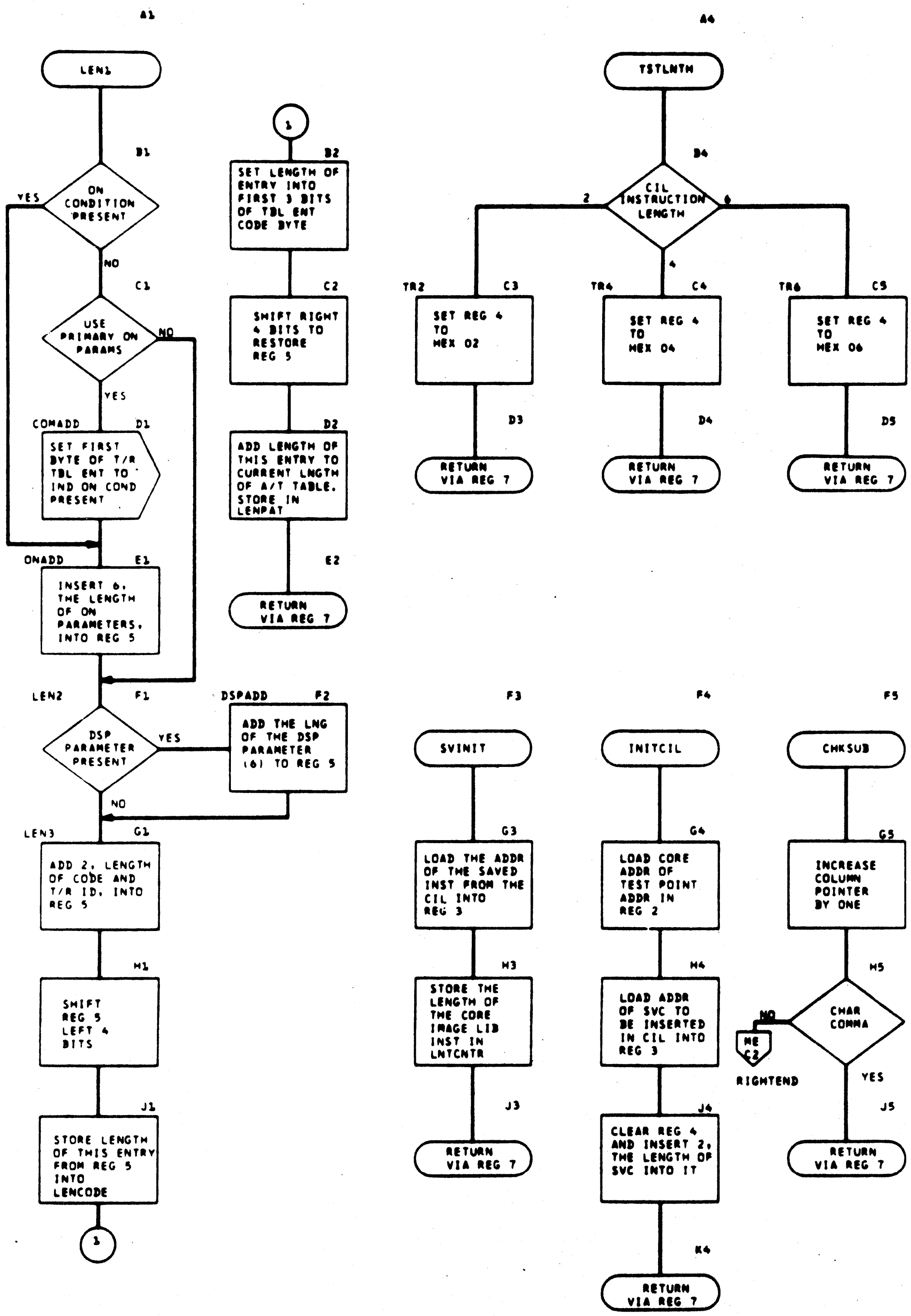
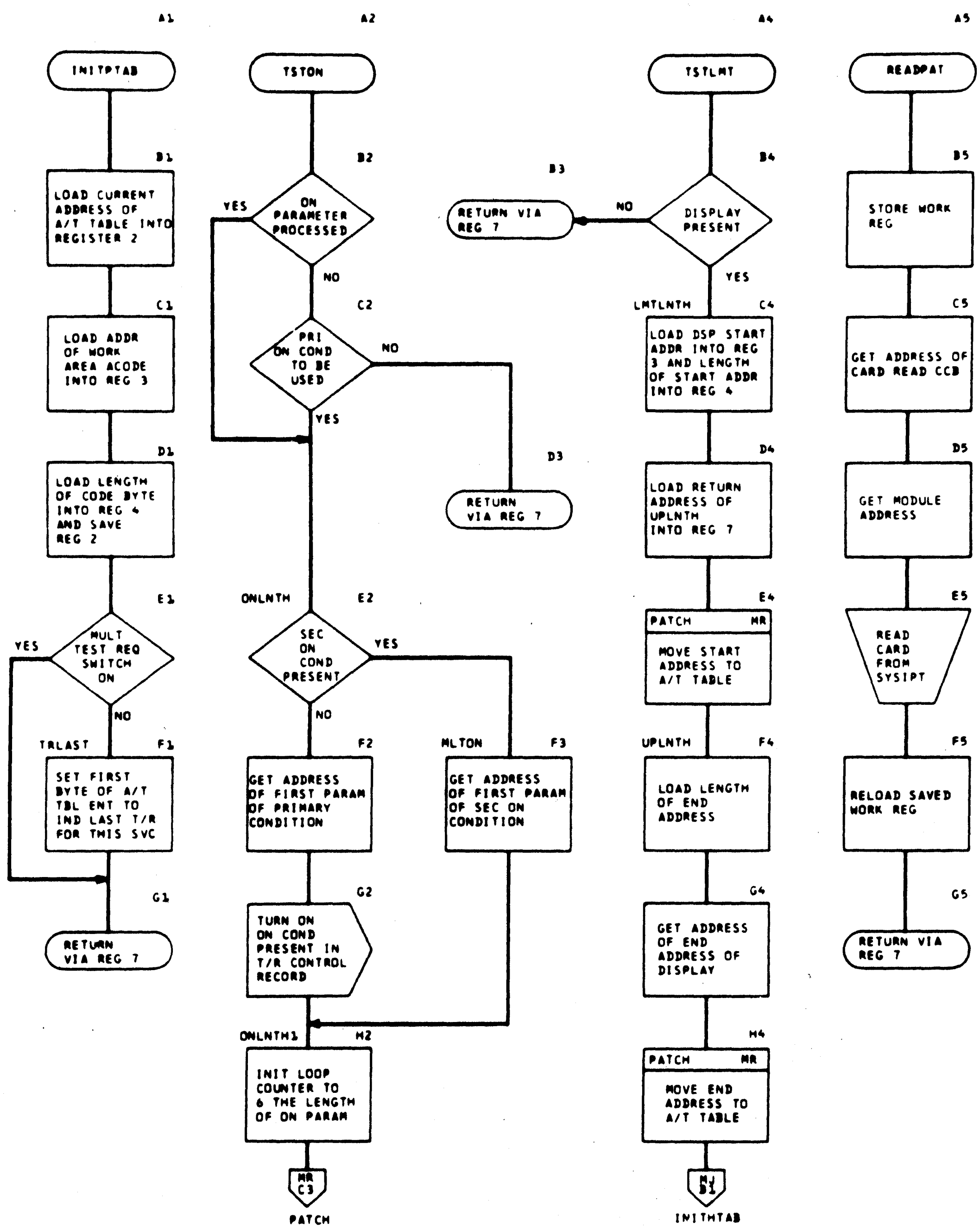
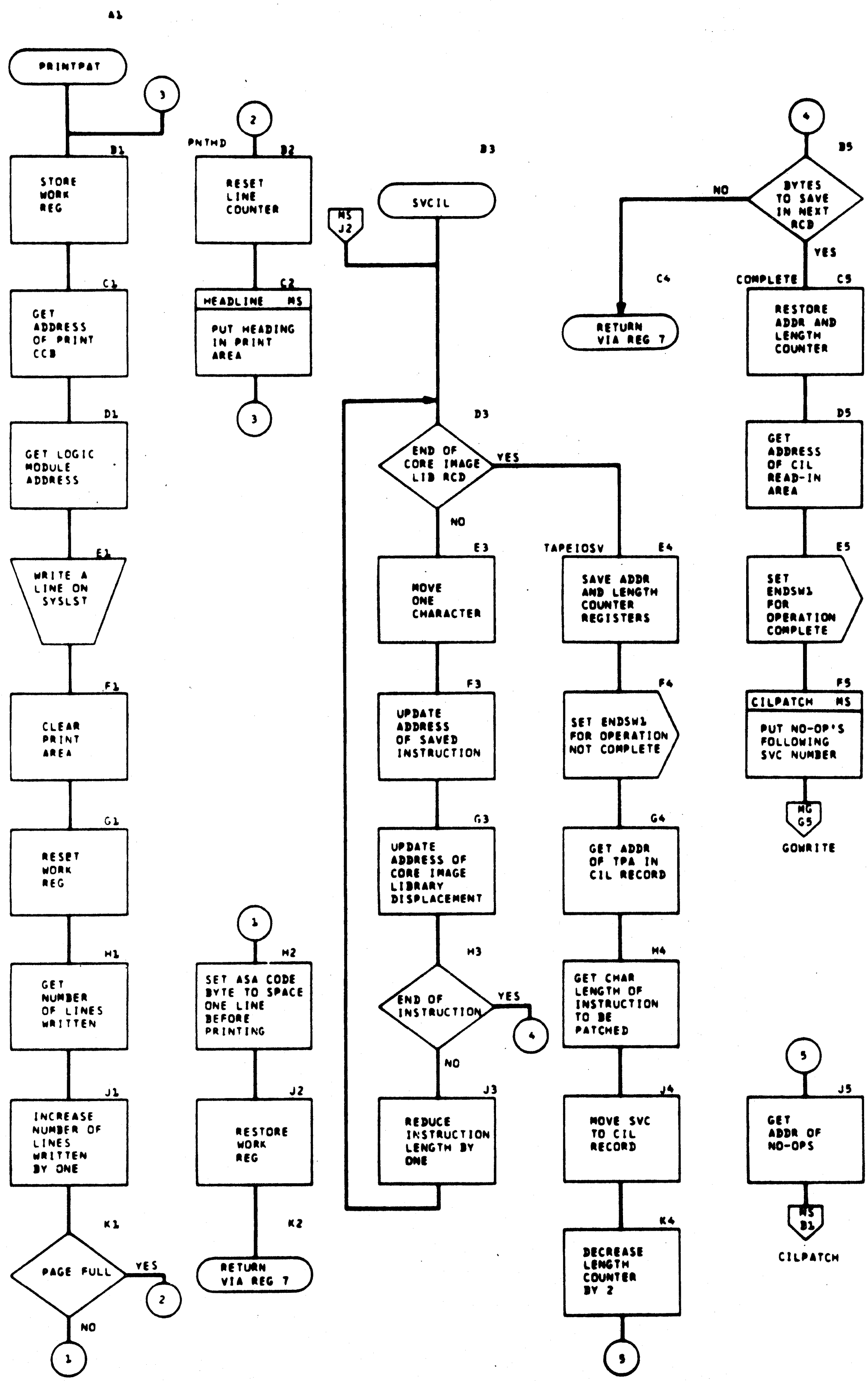


Chart MP. Build Test Request Table Entry (ATLEFC4) (Refer to Chart 11) (Part 14 of 22)




PRINTPAT
STORE WORK REG
GET ADDRESS OF PRINT CCB
GET LOGIC MODULE ADDRESS
WRITE A LINE ON SYSLST
CLEAR PRINT AREA
RESET WORK REG
GET NUMBER OF LINES WRITTEN
INCREASE NUMBER OF LINES WRITTEN BY ONE
PAGE FULL
YES
NO
PNTHD
RESET LINE COUNTER
HEADLINE MS
PUT HEADING IN PRINT AREA
SET ASA CODE BYTE TO SPACE ONE LINE BEFORE PRINTING
RESTORE WORK REG
RETURN VIA REG 7
SVCIL
MS J2
END OF CORE IMAGE LIB RCD
MOVE ONE CHARACTER
UPDATE ADDRESS OF SAVED INSTRUCTION
UPDATE ADDRESS OF CORE IMAGE LIBRARY DISPLACEMENT
END OF INSTRUCTION
REDUCE INSTRUCTION LENGTH BY ONE
TAPEIOSV
SAVE ADDR AND LENGTH COUNTER REGISTERS
SET ENDSW1 FOR OPERATION NOT COMPLETE
GET ADDR OF TPA IN CIL RECORD
GET CHAR LENGTH OF INSTRUCTION TO BE PATCHED
MOVE SVC TO CIL RECORD
DECREASE LENGTH COUNTER BY 2
RETURN VIA REG 7
BYTES TO SAVE IN NEXT RCD
COMPLETE
RESTORE ADDR AND LENGTH COUNTER
GET ADDRESS OF CIL READ-IN AREA
SET ENDSW1 FOR OPERATION COMPLETE
CILPATCH MS
PUT NO-OP'S FOLLOWING SVC NUMBER
MG G5
GOWRITE
GET ADDR OF NO-OPS
MS B1
CILPATCH

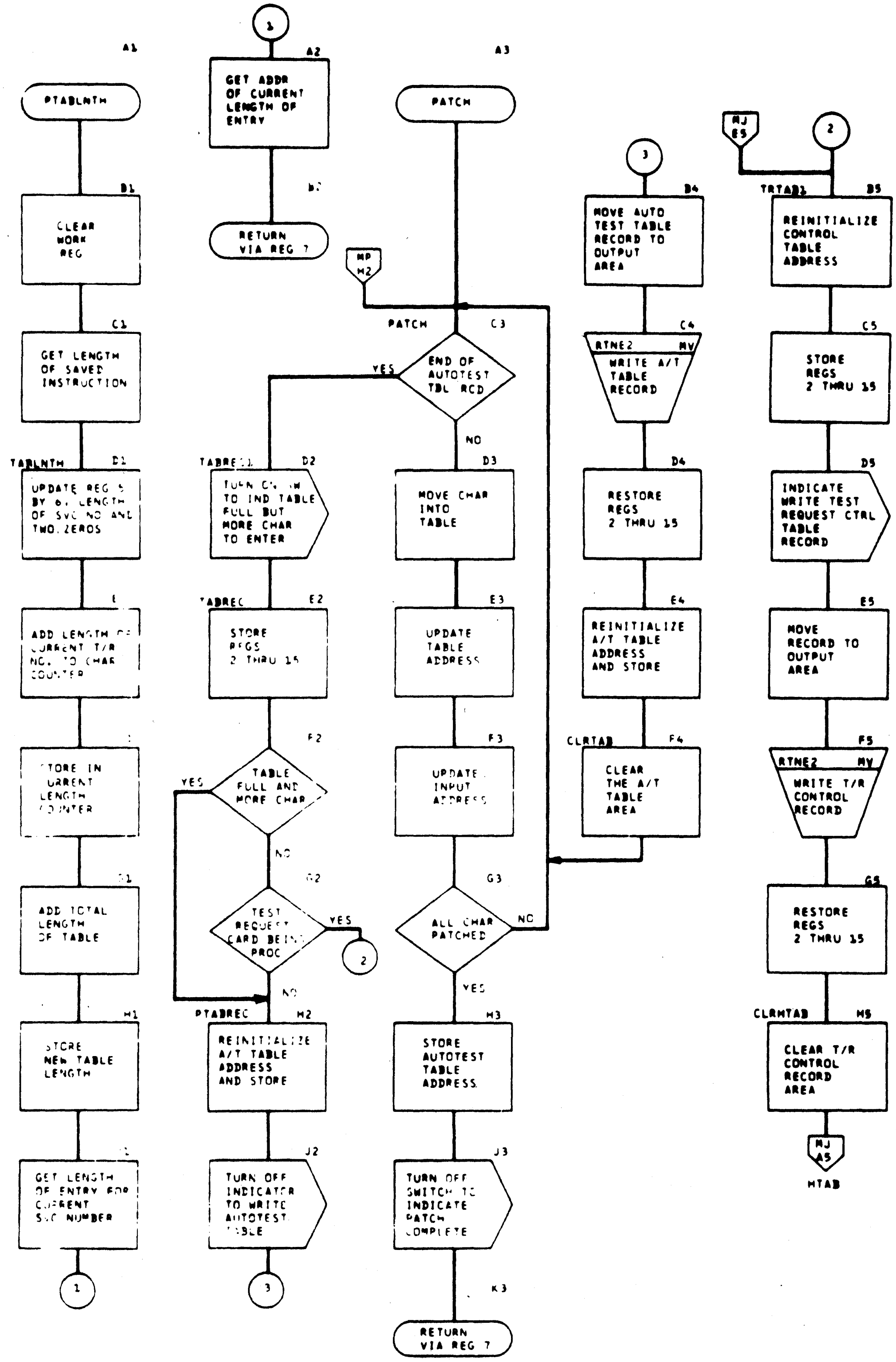

PTABLNTH
CLEAR WORK REG
GET LENGTH OF SAVED INSTRUCTION
TABLNTH
UPDATE REG 5 BY 6, LENGTH OF SVC NO AND TWO ZEROS
ADD LENGTH OF CURRENT T/R NO. TO CHAR COUNTER
STORE IN CURRENT LENGTH COUNTER
ADD TOTAL LENGTH OF TABLE
STORE NEW TABLE LENGTH
GET LENGTH OF ENTRY FOR CURRENT SVC NUMBER
GET ADDR OF CURRENT LENGTH OF ENTRY
RETURN VIA REG 7
TABREC1
TURN ON SW TO IND TABLE FULL BUT MORE CHAR TO ENTER
TABREC
STORE REGS 2 THRU 15
TABLE FULL AND MORE CHAR
YES
NO
TEST REQUEST CARD BEING PROC
YES
NO
PTABREC
REINITIALIZE A/T TABLE ADDRESS AND STORE
TURN OFF INDICATOR TO WRITE AUTOTEST TABLE
PATCH
MP H2
END OF AUTOTEST TBL RCD
YES
NO
MOVE CHAR INTO TABLE
UPDATE TABLE ADDRESS
UPDATE INPUT ADDRESS
ALL CHAR PATCHED
NO
YES
STORE AUTOTEST TABLE ADDRESS
TURN OFF SWITCH TO INDICATE PATCH COMPLETE
RETURN VIA REG 7
MOVE AUTO TEST TABLE RECORD TO OUTPUT AREA
RTNE2 MV
WRITE A/T TABLE RECORD
RESTORE REGS 2 THRU 15
REINITIALIZE A/T TABLE ADDRESS AND STORE
CLRTAB
CLEAR THE A/T TABLE AREA
MJ E5
TRTAB1
REINITIALIZE CONTROL TABLE ADDRESS
STORE REGS 2 THRU 15
INDICATE WRITE TEST REQUEST CTRL TABLE RECORD
MOVE RECORD TO OUTPUT AREA
RTNE2 MV
WRITE T/R CONTROL RECORD
RESTORE REGS 2 THRU 15
CLRHTAB
CLEAR T/R CONTROL RECORD AREA
MJ A5
HTAB

Chart MB. Build Test Request Table Entry (ATLEPC4) (Refer to Chart 11) (Part 17 of 22)

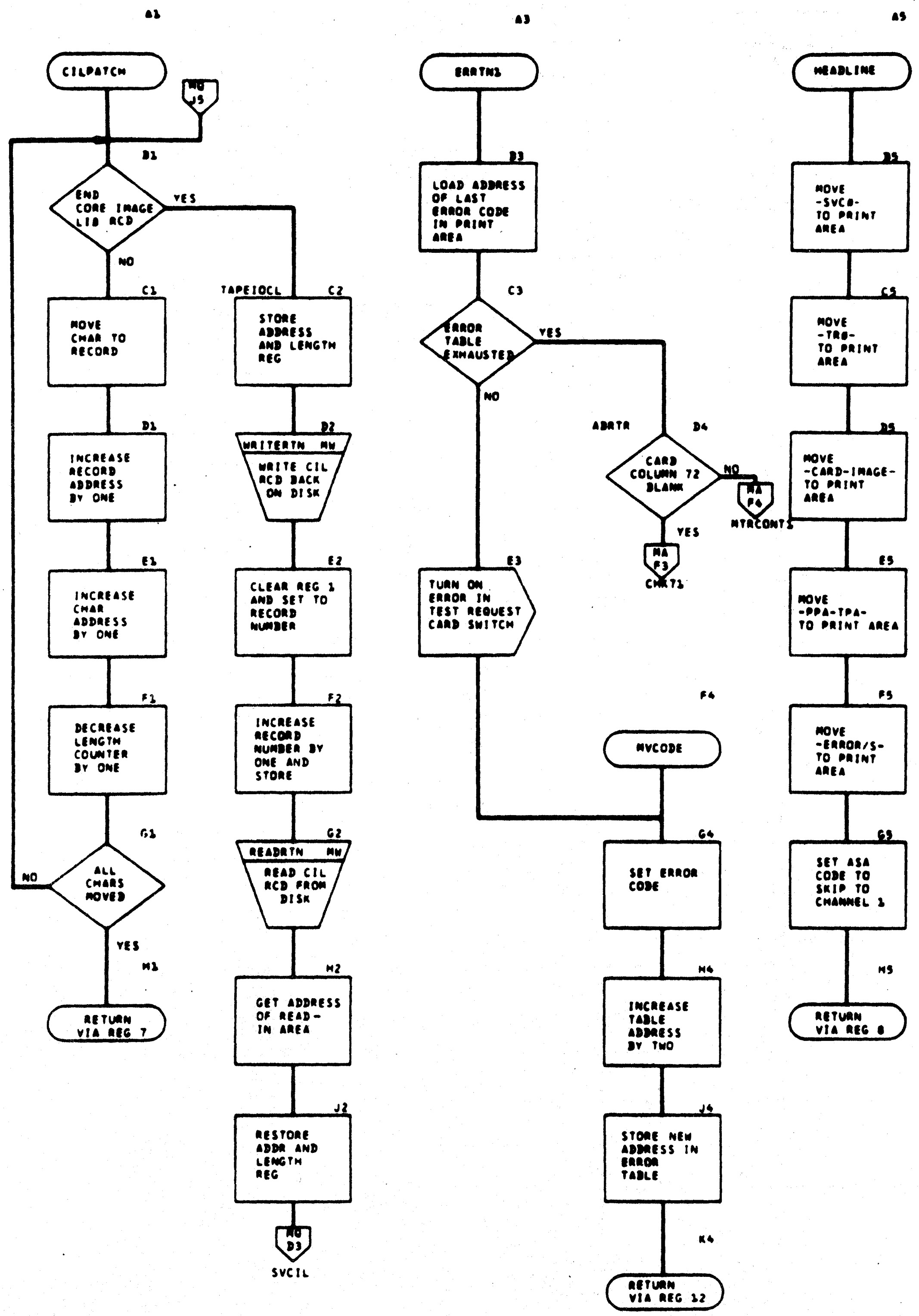


Chart MT. Build Test Request Table Entry (ATLEFC4) (Refer to Chart 11) (Part 18 of 22)

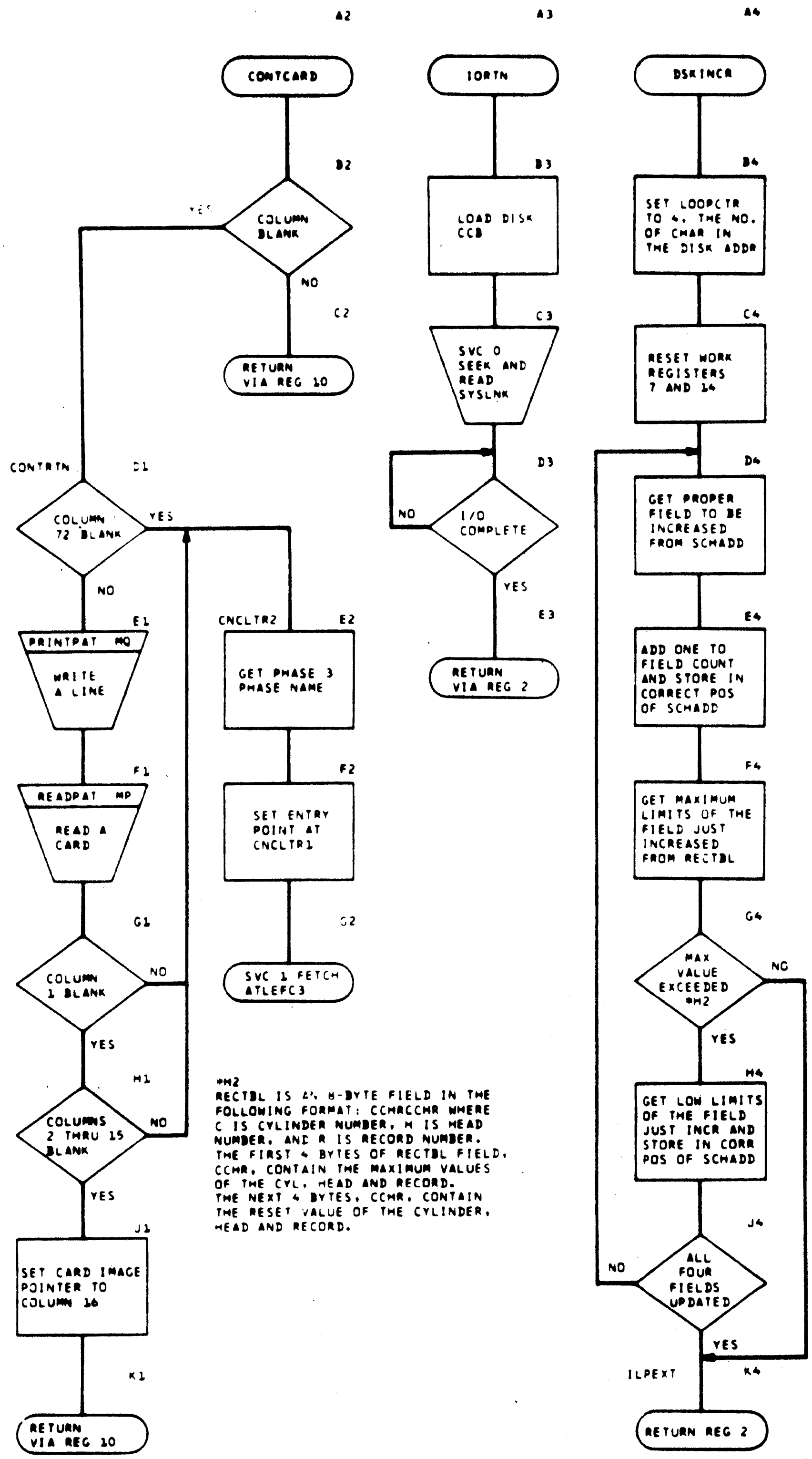


Chart MU. Build Test Request Table Entry (ATLEFC4) (Refer to Chart 11) (Part 19 of 22)

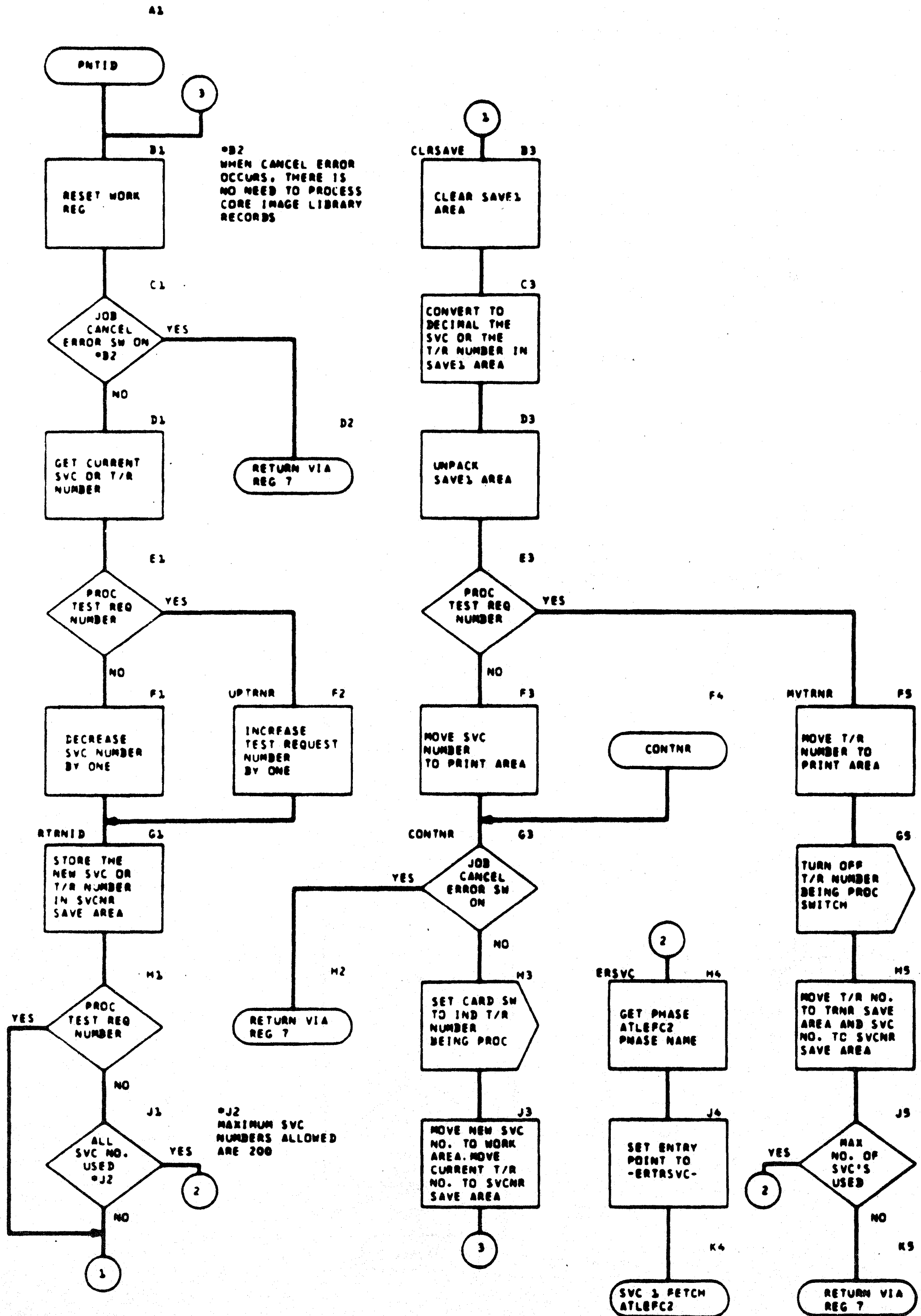


Chart MV. Build Test Request Table Entry (ATLEFC4) (Refer to Chart 11) (Part 20 of 22)

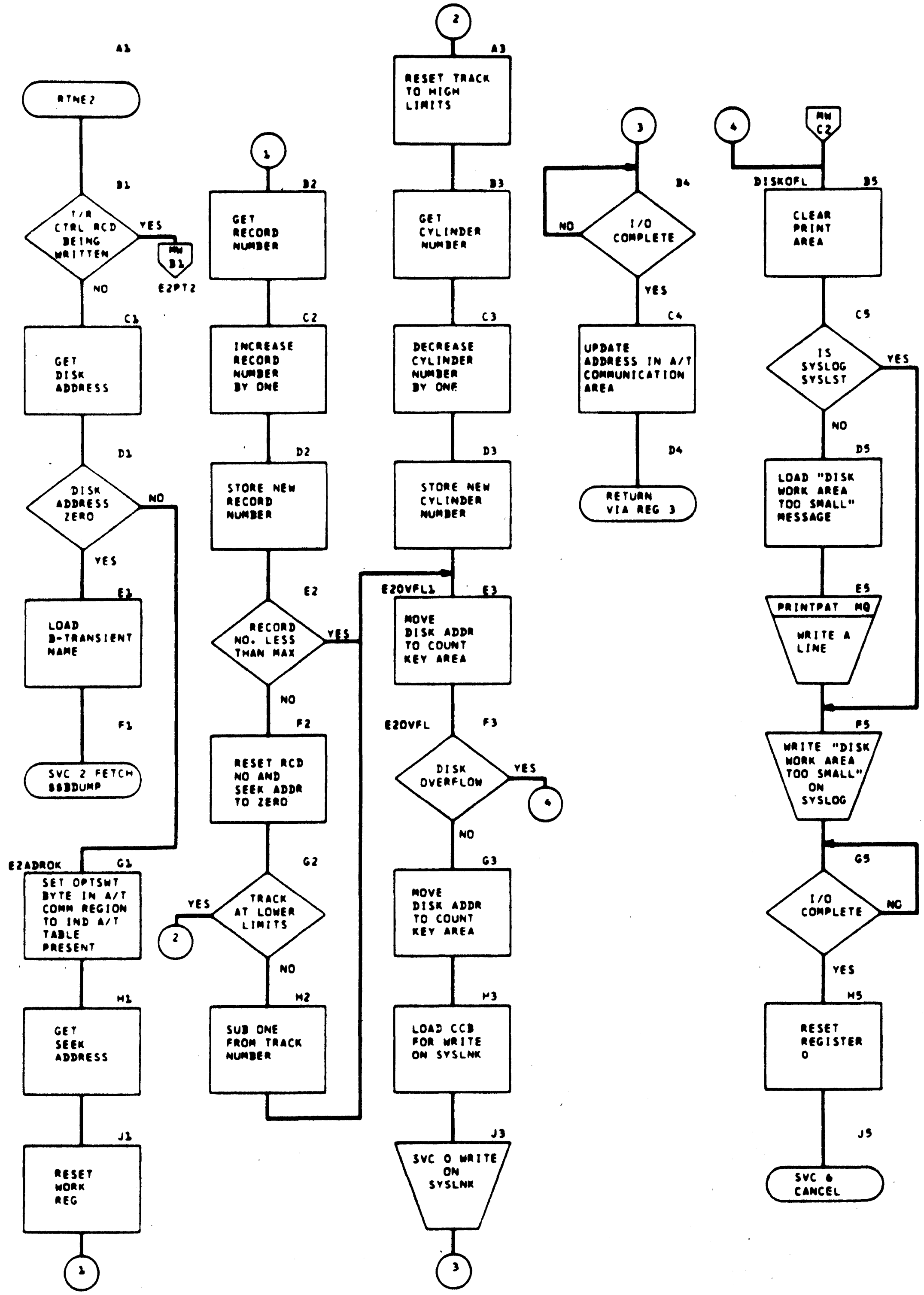


Chart MW. Build Test Request Table Entry (ATLEFC4) (Refer to Chart 11) (Part 21 of 22)

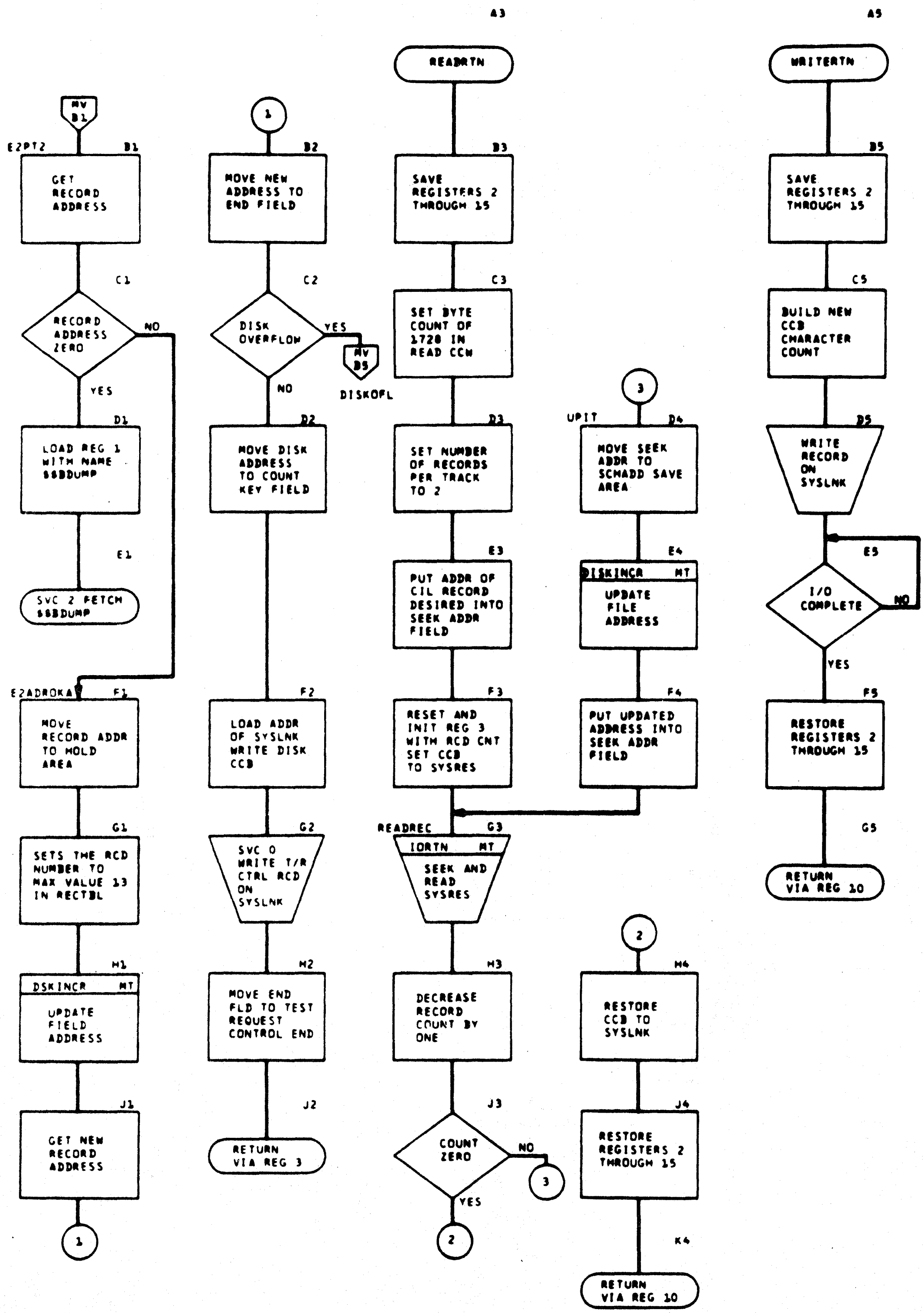


Chart MX. Build Test Request Table Entry (ATLEFC4) (Refer to Chart 11) (Part 22 of 22)

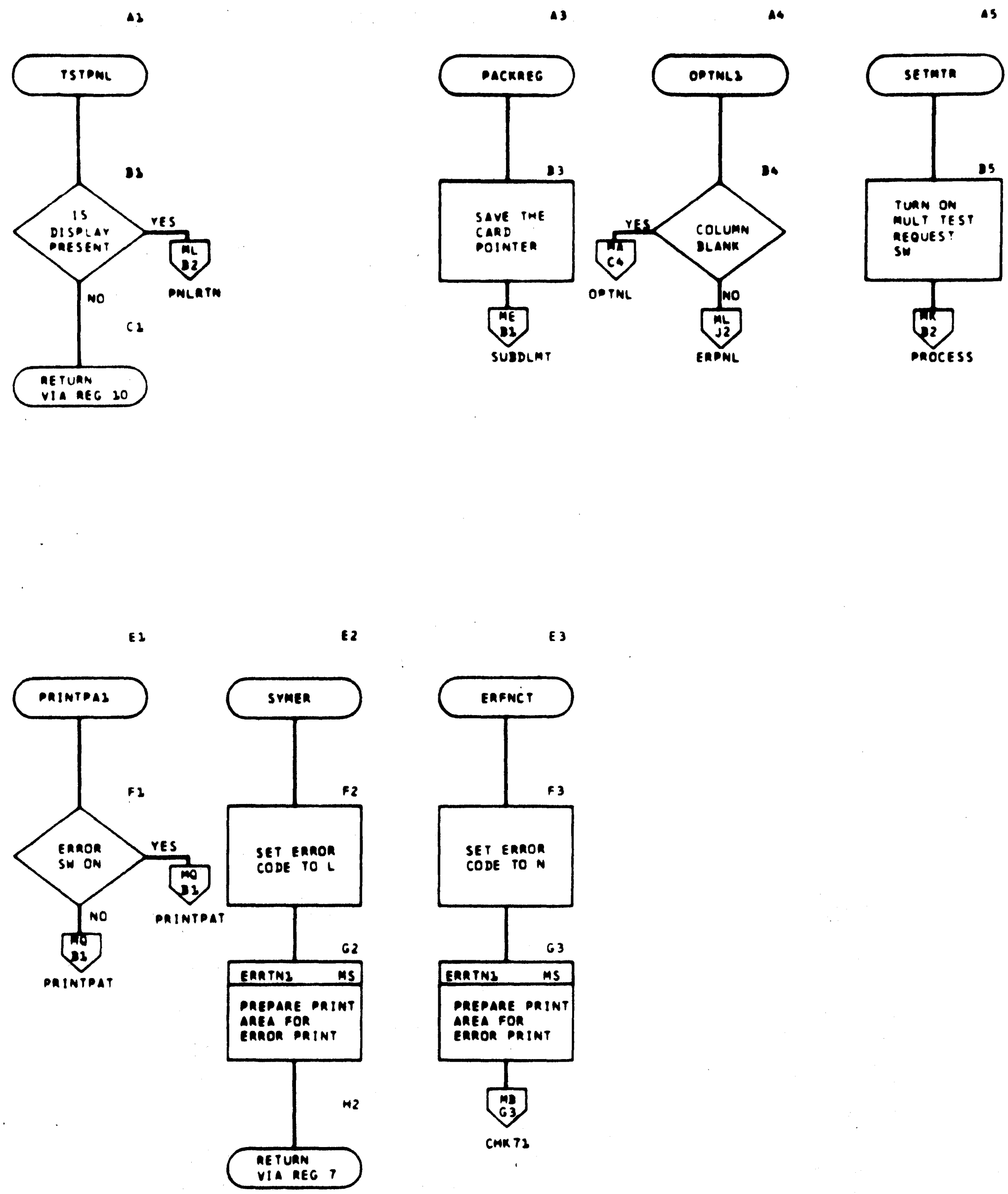


Chart NA. PCC Statement Processor (ATLEFCS) (Refer to Chart 11) (Part 1 of 3)

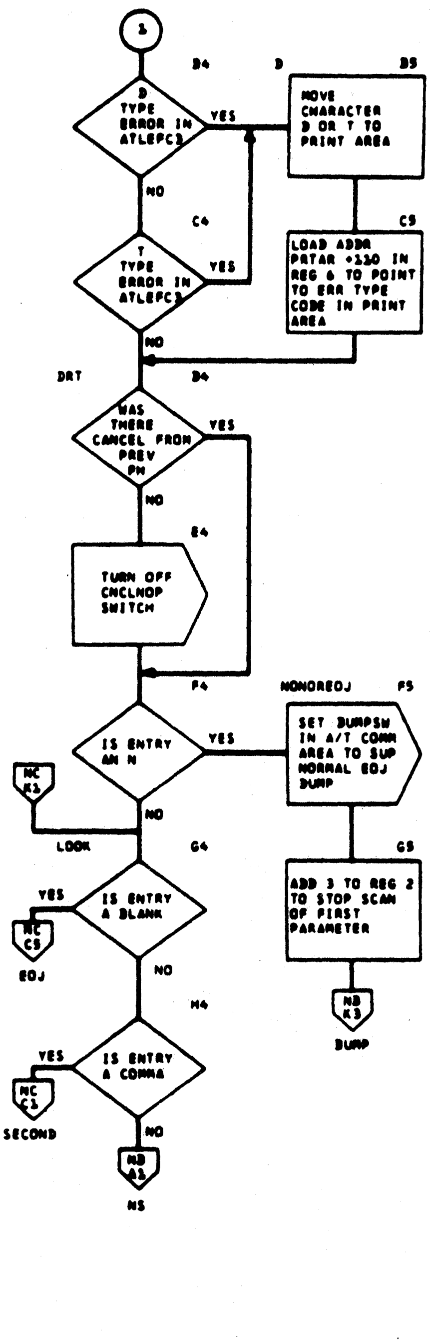


Chart NB. PCC Statement Processor (ATLEPC5) (Refer to Chart 11) (Part 2 of 3)

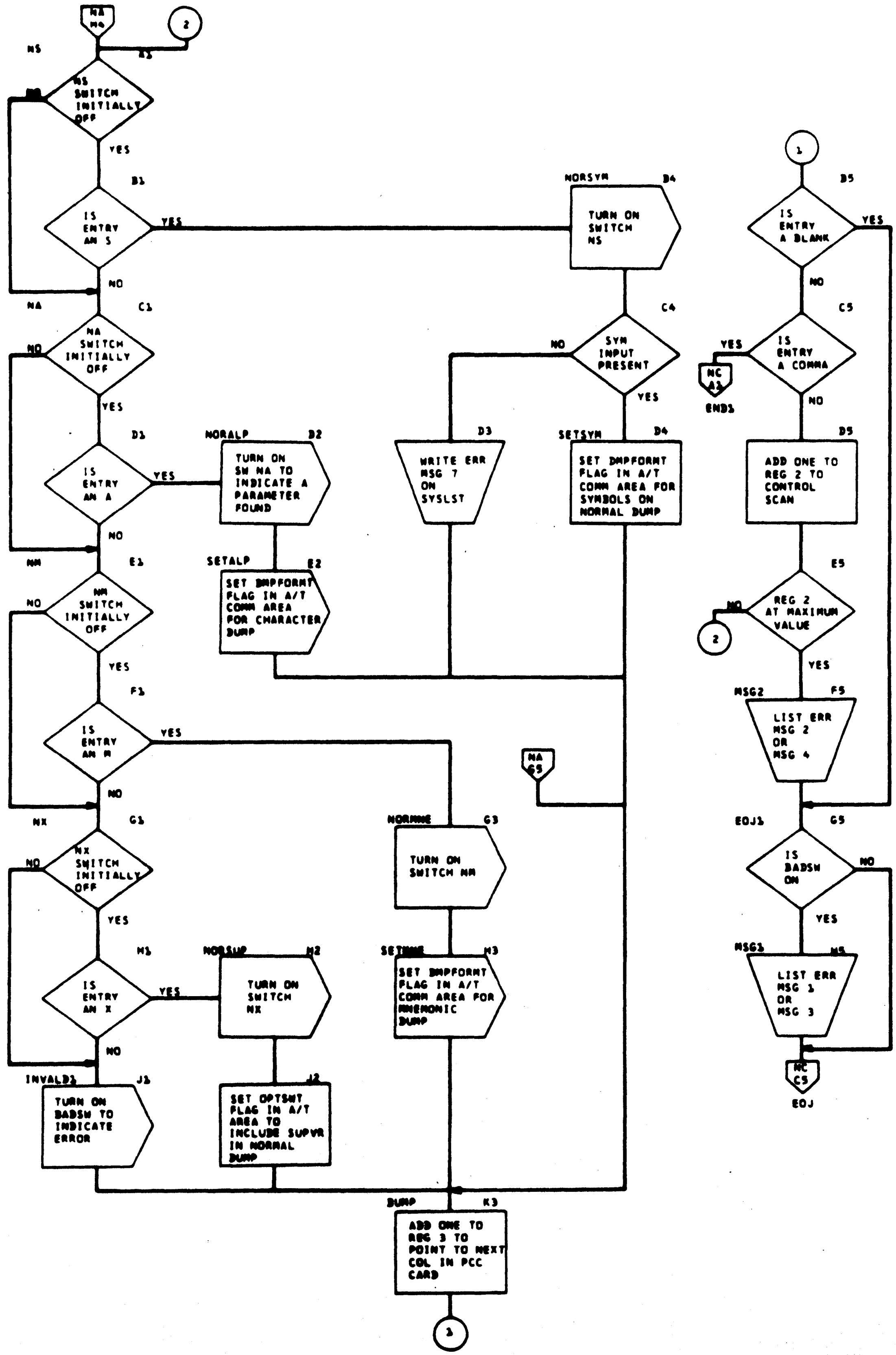


Chart NC. PCC Statement Processor (ATLEPC5) (Refer to Chart 11) (Part 3 of 3)

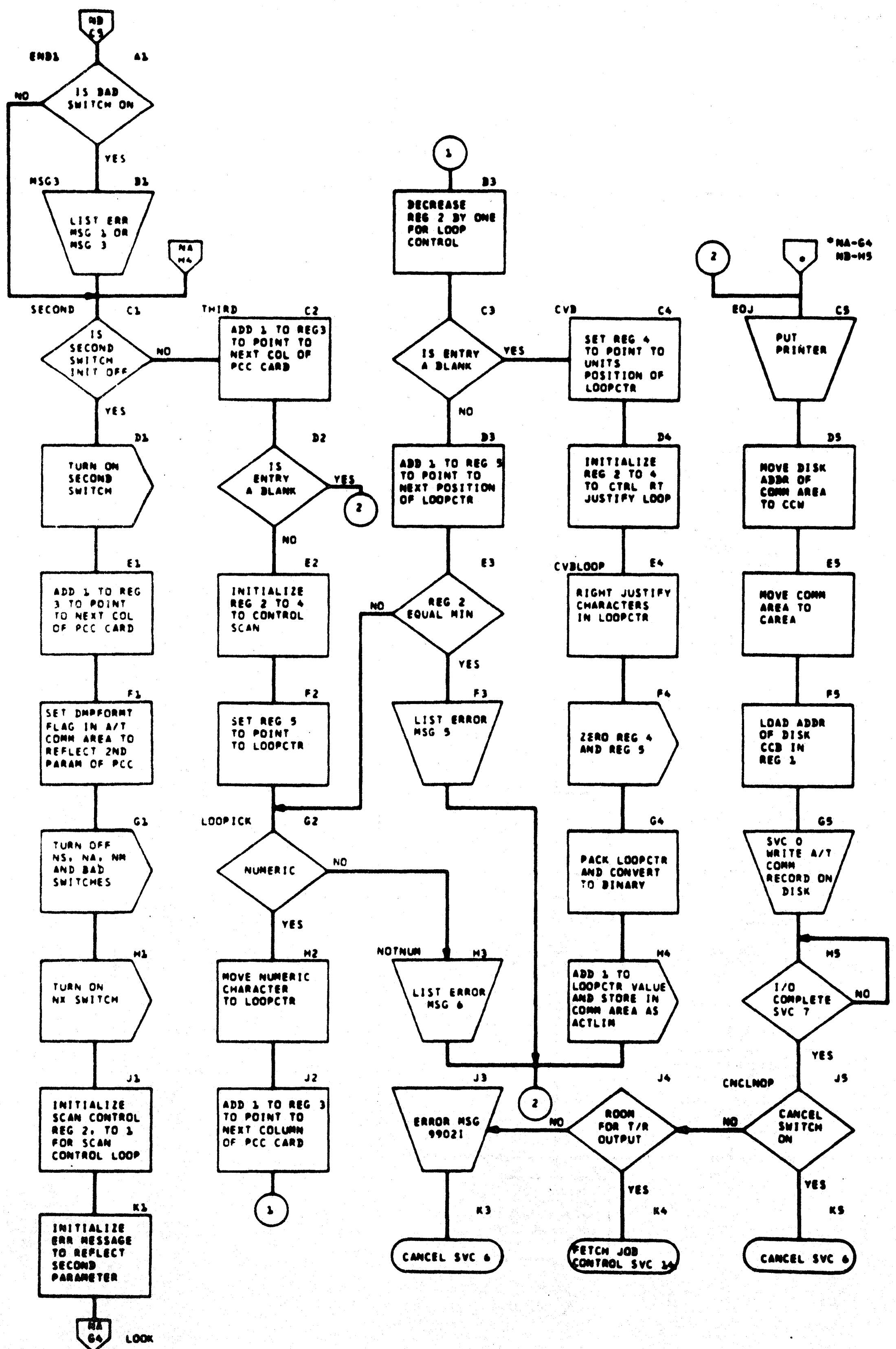
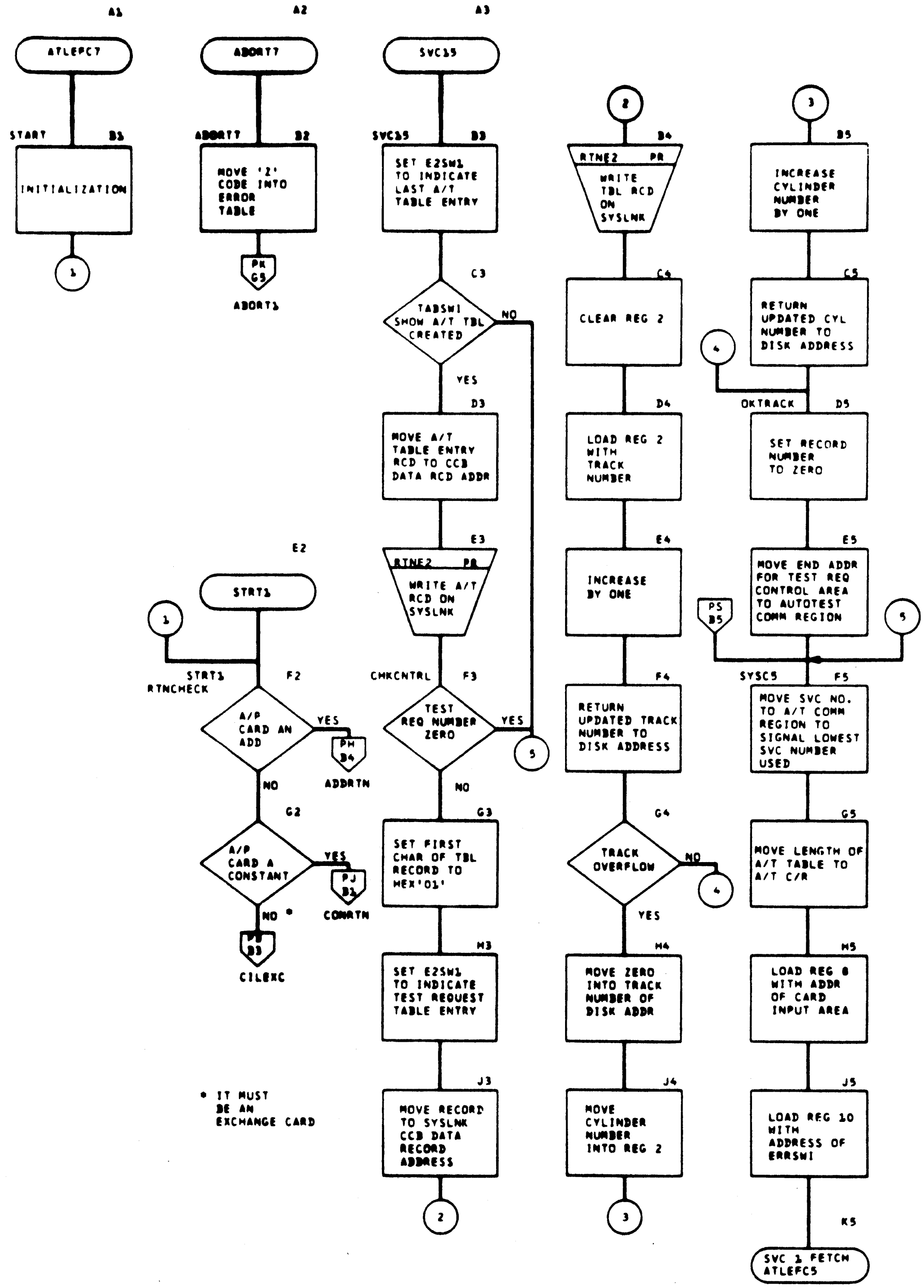
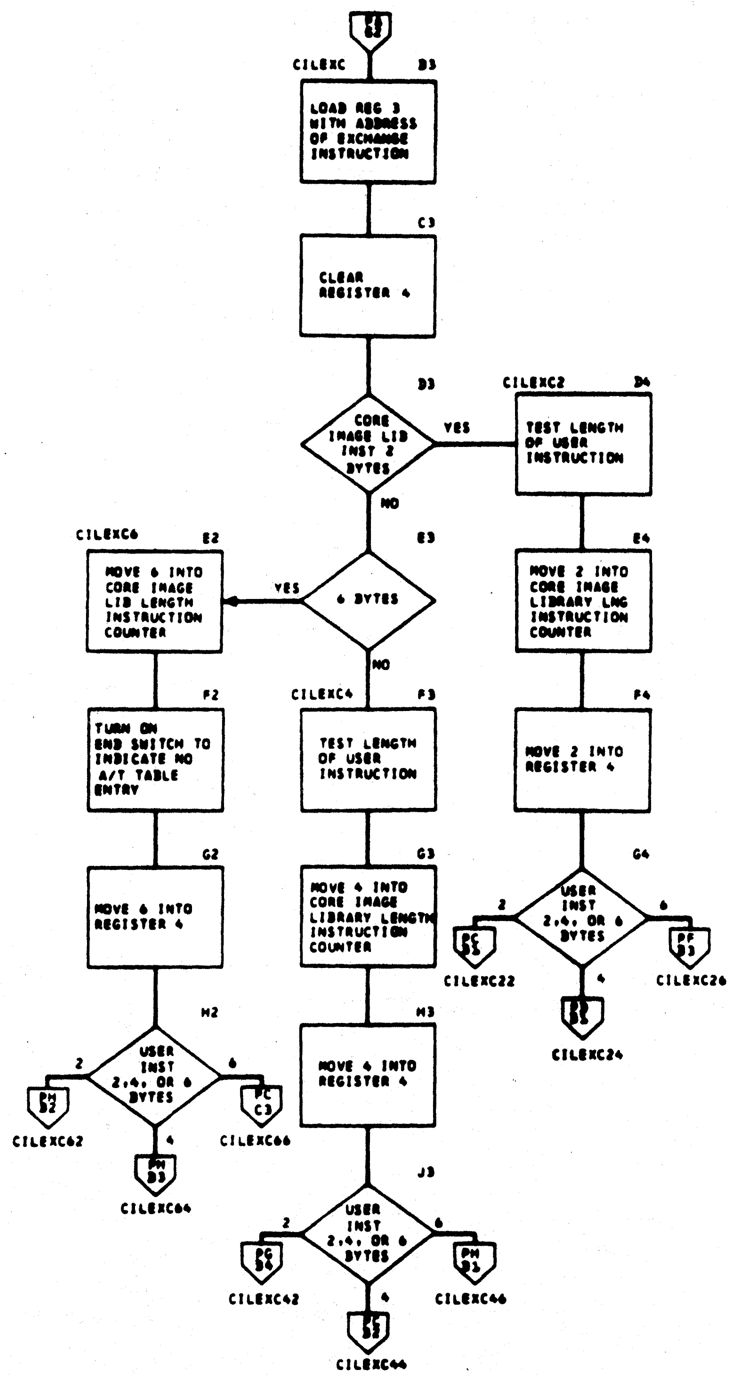
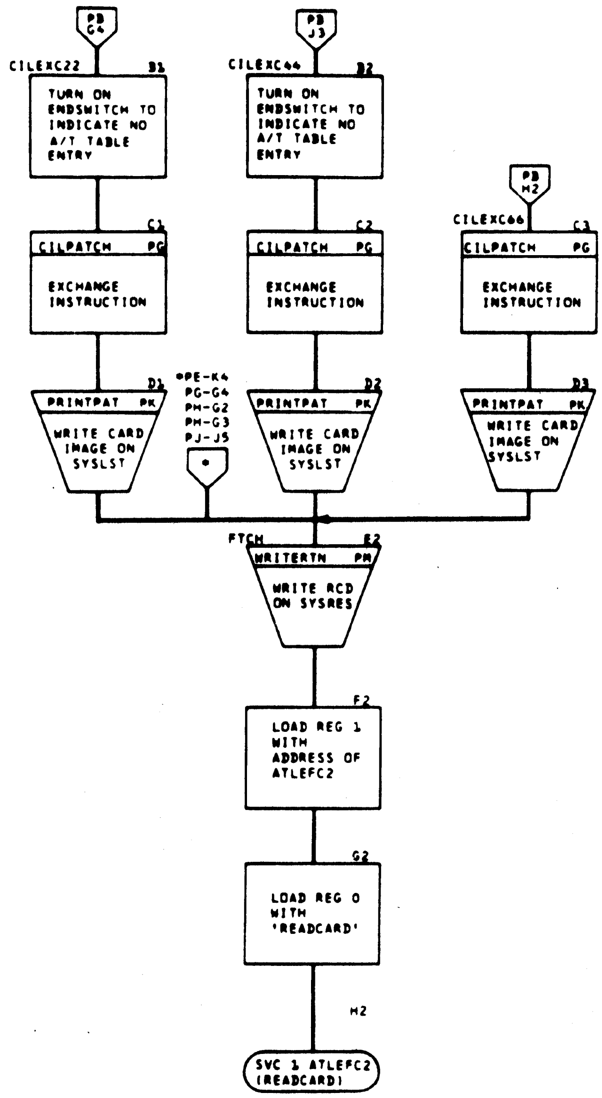


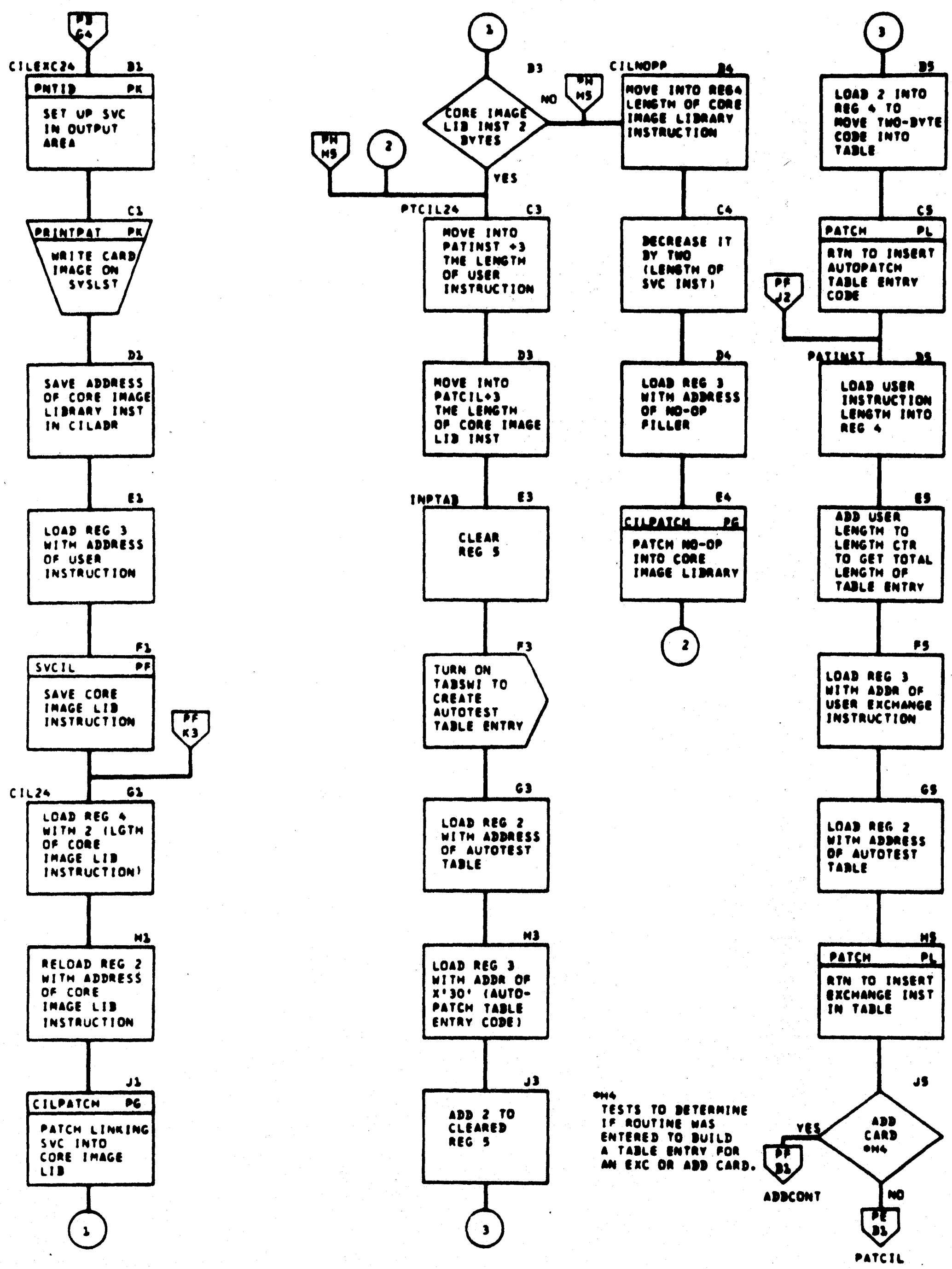
Chart PA. Autopatch User Program (ATLEFC7) (Refer to Chart 11) (Part 1 of 19)



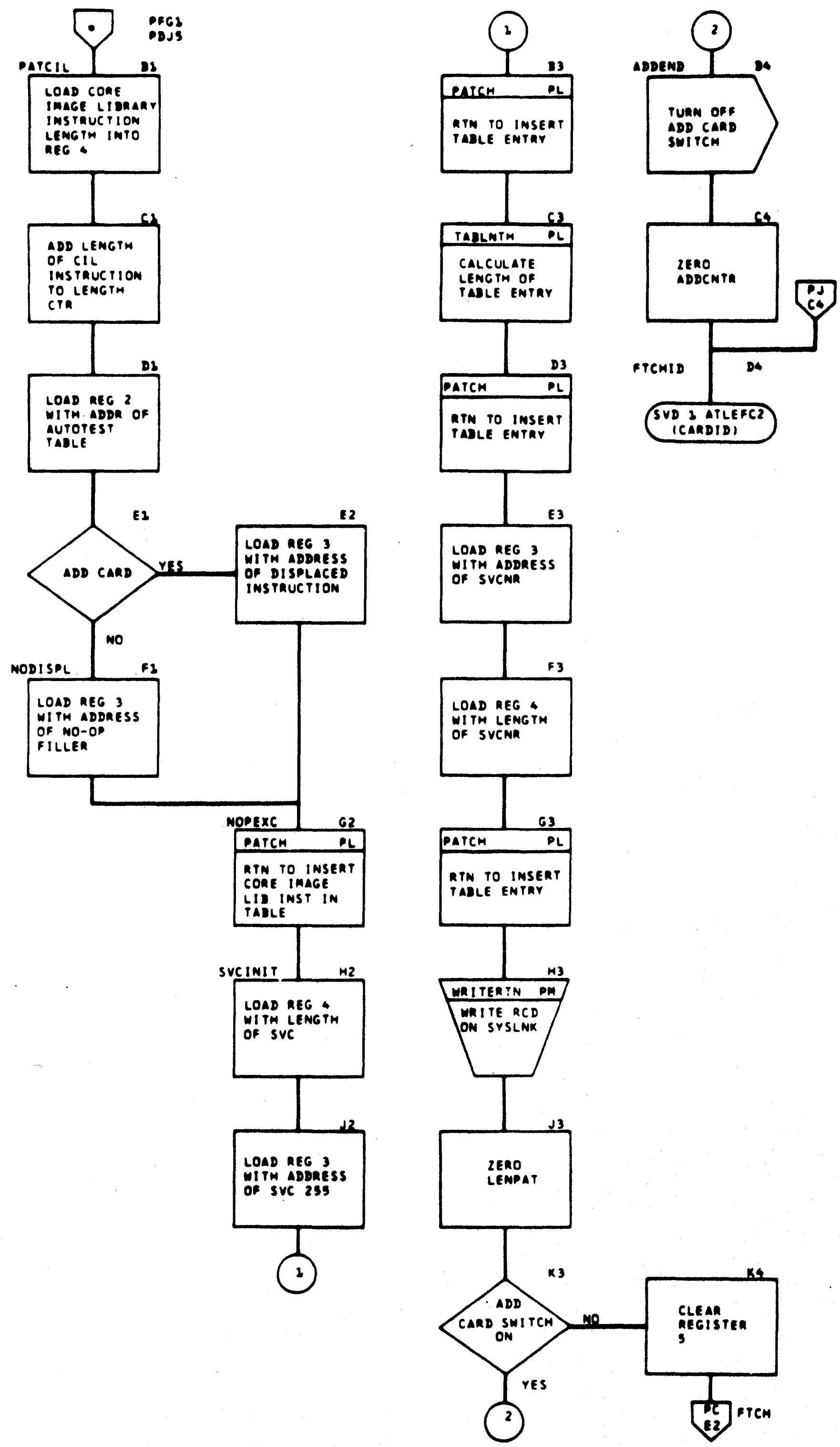

CILEXC
B3
LOAD REG 3 WITH ADDRESS OF EXCHANGE INSTRUCTION
C3
CLEAR REGISTER 4
D3
CORE IMAGE LIB INST 2 BYTES
YES
NO
CILEXC2
D4
TEST LENGTH OF USER INSTRUCTION
E4
MOVE 2 INTO CORE IMAGE LIBRARY LNG INSTRUCTION COUNTER
F4
MOVE 2 INTO REGISTER 4
G4
USER INST 2,4, OR 6 BYTES
2
4
6
CILEXC22
CILEXC24
CILEXC26
PF
B3
E3
6 BYTES
YES
NO
CILEXC6
E2
MOVE 6 INTO CORE IMAGE LIB LENGTH INSTRUCTION COUNTER
F2
TURN ON END SWITCH TO INDICATE NO A/T TABLE ENTRY
G2
MOVE 6 INTO REGISTER 4
H2
USER INST 2,4, OR 6 BYTES
2
4
6
PH
B2
PH
B3
PC
C3
CILEXC62
CILEXC64
CILEXC66
CILEXC4
F3
TEST LENGTH OF USER INSTRUCTION
G3
MOVE 4 INTO CORE IMAGE LIBRARY LENGTH INSTRUCTION COUNTER
H3
MOVE 4 INTO REGISTER 4
J3
USER INST 2,4, OR 6 BYTES
2
4
6
PG
B4
PH
B1
CILEXC42
CILEXC44
CILEXC46

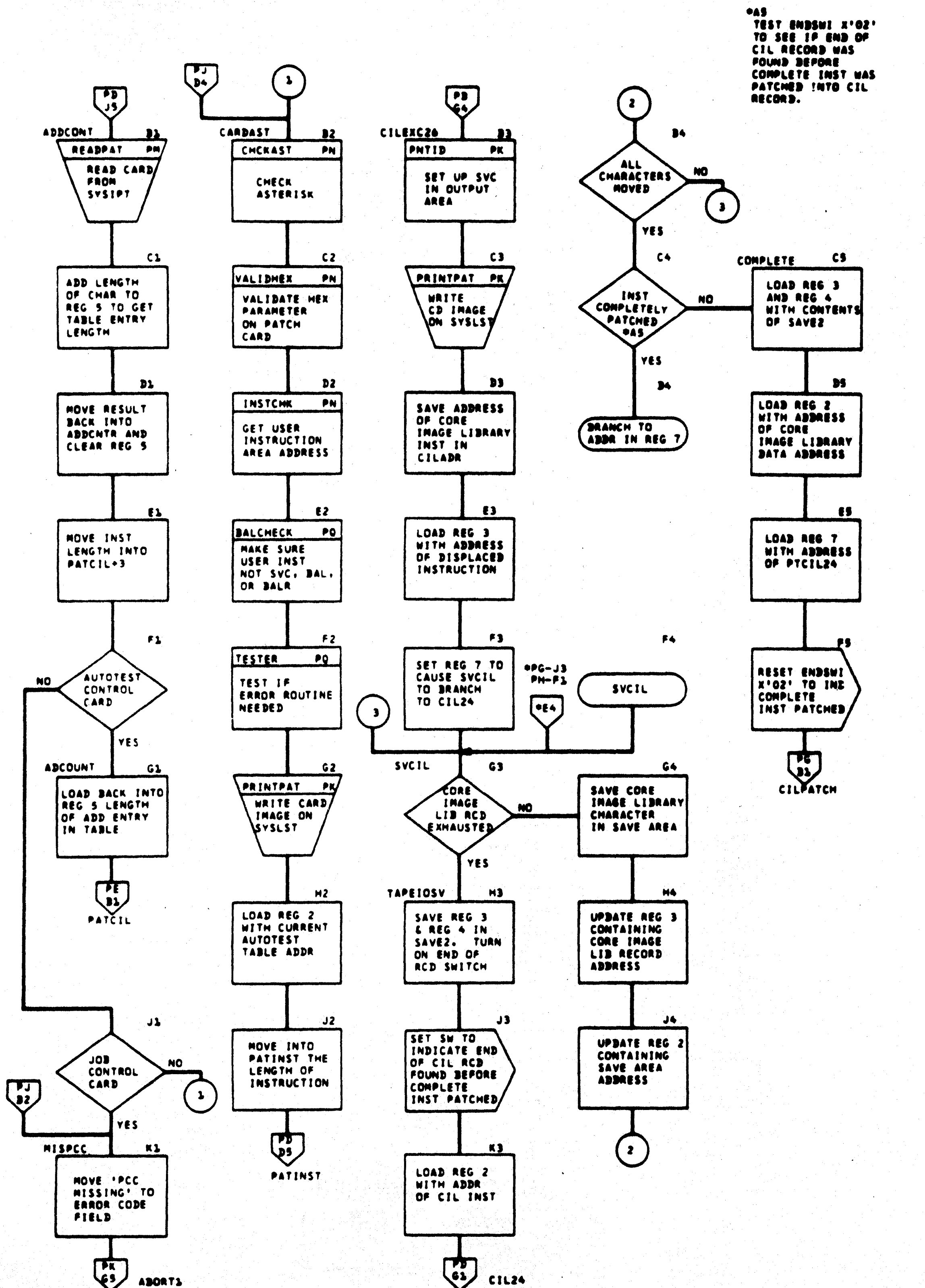


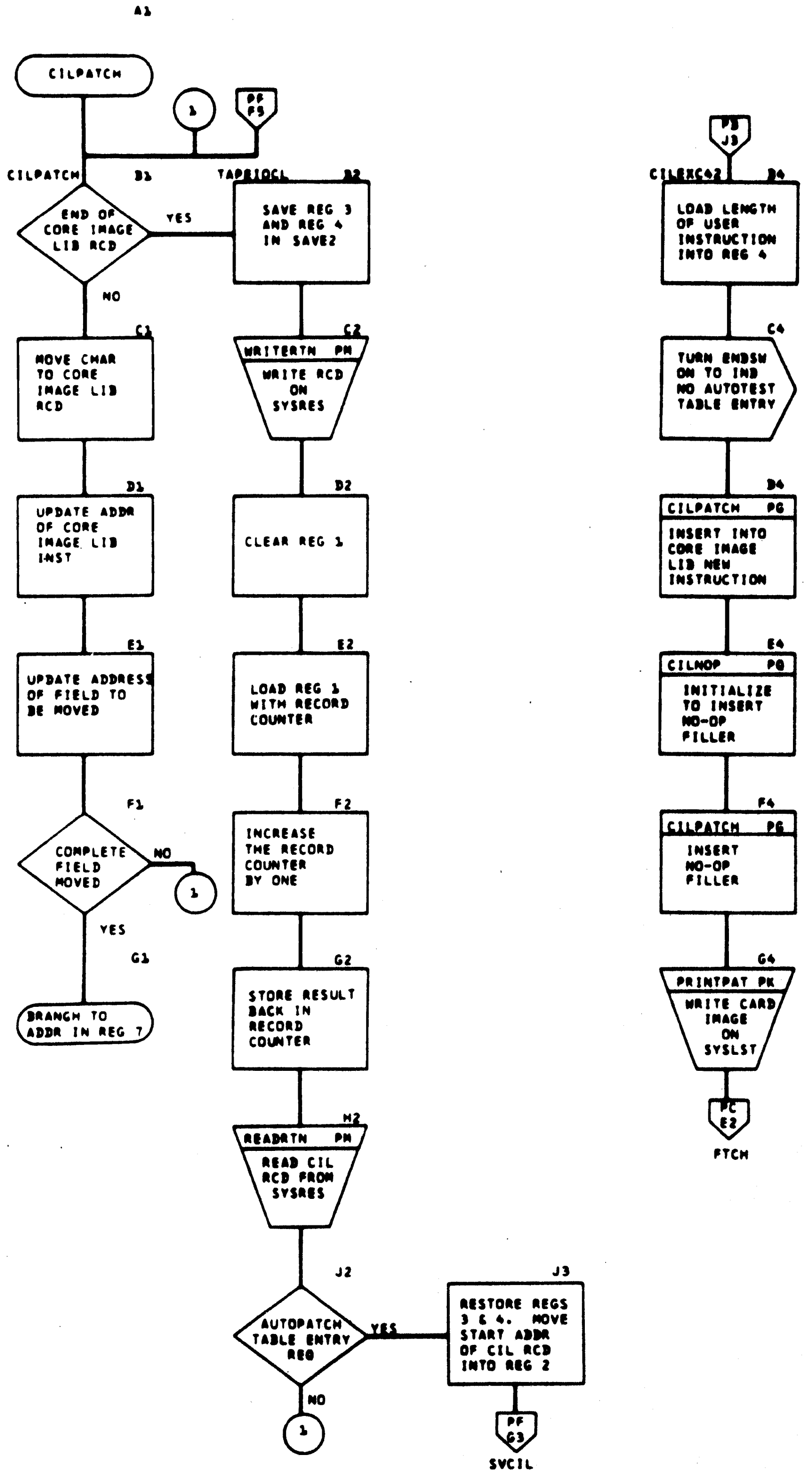
PB G4
CILEXC22 B1
TURN ON ENDSWITCH TO INDICATE NO A/T TABLE ENTRY
PB J3
CILEXC44 B2
TURN ON ENDSWITCH TO INDICATE NO A/T TABLE ENTRY
PB H2
CILEXC66 C3
C1 CILPATCH PG EXCHANGE INSTRUCTION
C2 CILPATCH PG EXCHANGE INSTRUCTION
CILPATCH PG EXCHANGE INSTRUCTION
D1 PRINTPAT PK WRITE CARD IMAGE ON SYSLST
*PE-K4 PG-G4 PH-G2 PH-G3 PJ-J5
D2 PRINTPAT PK WRITE CARD IMAGE ON SYSLST
D3 PRINTPAT PK WRITE CARD IMAGE ON SYSLST
FTCH E2
WRITERTN PH WRITE RCD ON SYSRES
F2 LOAD REG 1 WITH ADDRESS OF ATLEFC2
G2 LOAD REG 0 WITH 'READCARD'
H2
SVC 1 ATLEFC2 (READCARD)

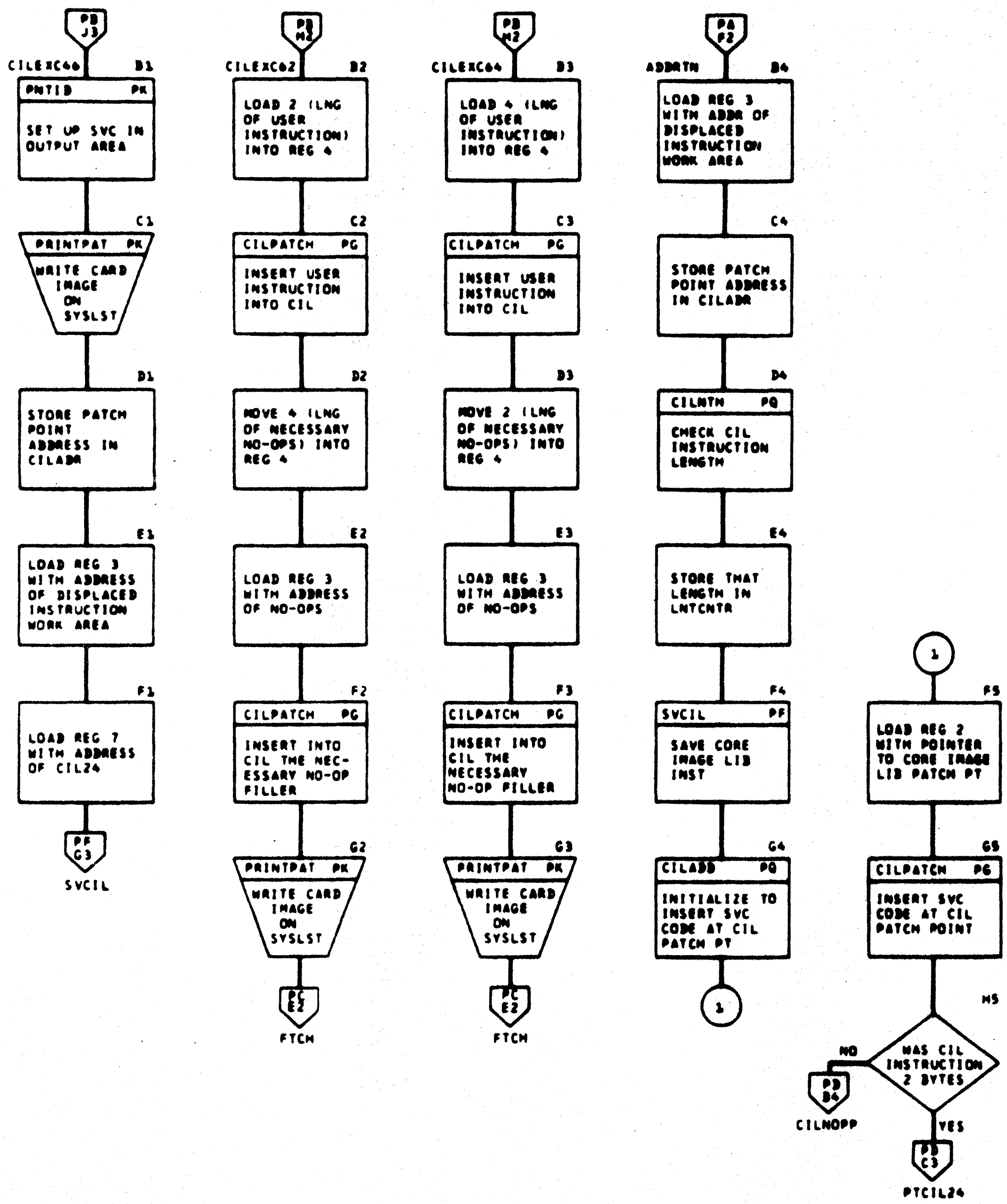


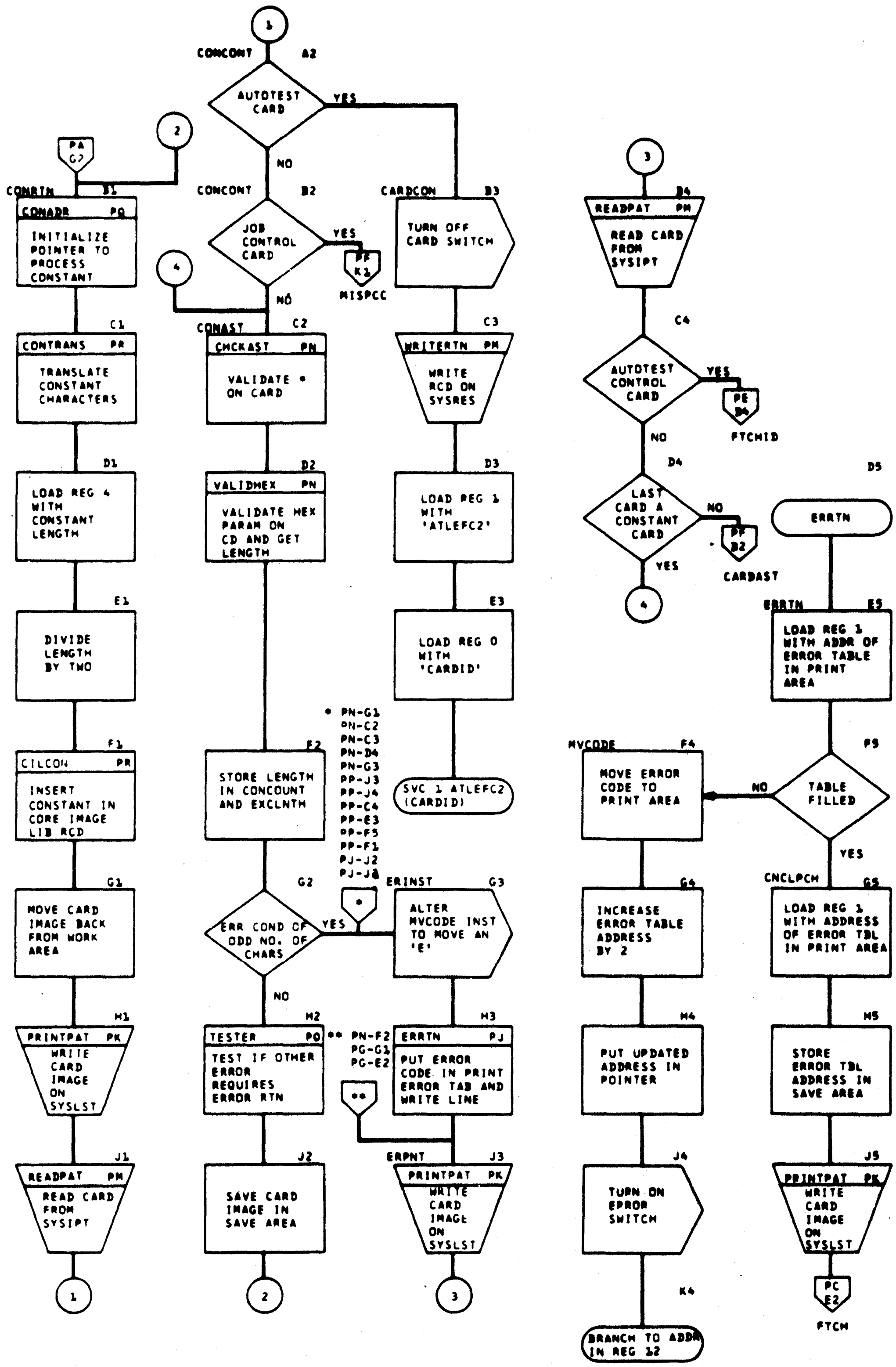
CILEXC24
PNTID PK
SET UP SVC IN OUTPUT AREA
PRINTPAT PK
WRITE CARD IMAGE ON SYSLST
SAVE ADDRESS OF CORE IMAGE LIBRARY INST IN CILADR
LOAD REG 3 WITH ADDRESS OF USER INSTRUCTION
SVCIL PF
SAVE CORE IMAGE LIB INSTRUCTION
CIL24
LOAD REG 4 WITH 2 (LGTH OF CORE IMAGE LIB INSTRUCTION)
RELOAD REG 2 WITH ADDRESS OF CORE IMAGE LIB INSTRUCTION
CILPATCH PG
PATCH LINKING SVC INTO CORE IMAGE LIB
CORE IMAGE LIB INST 2 BYTES
NO
YES
PTCIL24
MOVE INTO PATINST +3 THE LENGTH OF USER INSTRUCTION
MOVE INTO PATCIL+3 THE LENGTH OF CORE IMAGE LIB INST
INPTAB
CLEAR REG 5
TURN ON TABSWI TO CREATE AUTOTEST TABLE ENTRY
LOAD REG 2 WITH ADDRESS OF AUTOTEST TABLE
LOAD REG 3 WITH ADDR OF X'30' (AUTO-PATCH TABLE ENTRY CODE)
ADD 2 TO CLEARED REG 5
CILNOPP
MOVE INTO REG4 LENGTH OF CORE IMAGE LIBRARY INSTRUCTION
DECREASE IT BY TWO (LENGTH OF SVC INST)
LOAD REG 3 WITH ADDRESS OF NO-OP FILLER
CILPATCH PG
PATCH NO-OP INTO CORE IMAGE LIBRARY
LOAD 2 INTO REG 4 TO MOVE TWO-BYTE CODE INTO TABLE
PATCH PL
RTN TO INSERT AUTOPATCH TABLE ENTRY CODE
PATINST
LOAD USER INSTRUCTION LENGTH INTO REG 4
ADD USER LENGTH TO LENGTH CTR TO GET TOTAL LENGTH OF TABLE ENTRY
LOAD REG 3 WITH ADDR OF USER EXCHANGE INSTRUCTION
LOAD REG 2 WITH ADDRESS OF AUTOTEST TABLE
PATCH PL
RTN TO INSERT EXCHANGE INST IN TABLE
ADD CARD *H4
*H4 TESTS TO DETERMINE IF ROUTINE WAS ENTERED TO BUILD A TABLE ENTRY FOR AN EXC OR ADD CARD.
ADDCONT
PATCIL


PFG1
PDJ5
PATCIL B1
LOAD CORE IMAGE LIBRARY INSTRUCTION LENGTH INTO REG 4
C1
ADD LENGTH OF CIL INSTRUCTION TO LENGTH CTR
D1
LOAD REG 2 WITH ADDR OF AUTOTEST TABLE
E1
ADD CARD
YES
NO
E2
LOAD REG 3 WITH ADDRESS OF DISPLACED INSTRUCTION
NODISPL F1
LOAD REG 3 WITH ADDRESS OF NO-OP FILLER
NOPEXC G2
PATCH PL
RTN TO INSERT CORE IMAGE LIB INST IN TABLE
SVCINIT H2
LOAD REG 4 WITH LENGTH OF SVC
J2
LOAD REG 3 WITH ADDRESS OF SVC 255
1
B3
PATCH PL
RTN TO INSERT TABLE ENTRY
C3
TABLNTH PL
CALCULATE LENGTH OF TABLE ENTRY
D3
PATCH PL
RTN TO INSERT TABLE ENTRY
E3
LOAD REG 3 WITH ADDRESS OF SVCNR
F3
LOAD REG 4 WITH LENGTH OF SVCNR
G3
PATCH PL
RTN TO INSERT TABLE ENTRY
H3
WRITERTN PM
WRITE RCD ON SYSLNK
J3
ZERO LENPAT
K3
ADD CARD SWITCH ON
NO
YES
2
2
ADDEND B4
TURN OFF ADD CARD SWITCH
C4
ZERO ADDCNTR
PJ C4
FTCHID D4
SVD 1 ATLEFC2 (CARDID)
K4
CLEAR REGISTER 5
PC E2
FTCH


*A5
TEST ENDSWI X'02' TO SEE IF END OF CIL RECORD WAS FOUND BEFORE COMPLETE INST WAS PATCHED INTO CIL RECORD.
ADDCONT
READPAT PM
READ CARD FROM SYSIPT
ADD LENGTH OF CHAR TO REG 5 TO GET TABLE ENTRY LENGTH
MOVE RESULT BACK INTO ADDCNTR AND CLEAR REG 5
MOVE INST LENGTH INTO PATCIL+3
AUTOTEST CONTROL CARD
ADCOUNT
LOAD BACK INTO REG 5 LENGTH OF ADD ENTRY IN TABLE
PATCIL
JOB CONTROL CARD
MISPCC
MOVE 'PCC MISSING' TO ERROR CODE FIELD
ABORT1
CARDAST
CHCKAST PN
CHECK ASTERISK
VALIDHEX PN
VALIDATE HEX PARAMETER ON PATCH CARD
INSTCHK PN
GET USER INSTRUCTION AREA ADDRESS
BALCHECK PO
MAKE SURE USER INST NOT SVC, BAL, OR BALR
TESTER PQ
TEST IF ERROR ROUTINE NEEDED
PRINTPAT PK
WRITE CARD IMAGE ON SYSLST
LOAD REG 2 WITH CURRENT AUTOTEST TABLE ADDR
MOVE INTO PATINST THE LENGTH OF INSTRUCTION
PATINST
CILEXC26
PNTID PK
SET UP SVC IN OUTPUT AREA
PRINTPAT PK
WRITE CD IMAGE ON SYSLST
SAVE ADDRESS OF CORE IMAGE LIBRARY INST IN CILADR
LOAD REG 3 WITH ADDRESS OF DISPLACED INSTRUCTION
SET REG 7 TO CAUSE SVCIL TO BRANCH TO CIL24
*PG-J3 PH-F1
SVCIL
CORE IMAGE LIB RCD EXHAUSTED
TAPEIOSV
SAVE REG 3 & REG 4 IN SAVE2. TURN ON END OF RCD SWITCH
SET SW TO INDICATE END OF CIL RCD FOUND BEFORE COMPLETE INST PATCHED
LOAD REG 2 WITH ADDR OF CIL INST
CIL24
ALL CHARACTERS MOVED
INST COMPLETELY PATCHED *A5
BRANCH TO ADDR IN REG 7
SAVE CORE IMAGE LIBRARY CHARACTER IN SAVE AREA
UPDATE REG 3 CONTAINING CORE IMAGE LIB RECORD ADDRESS
UPDATE REG 2 CONTAINING SAVE AREA ADDRESS
COMPLETE
LOAD REG 3 AND REG 4 WITH CONTENTS OF SAVE2
LOAD REG 2 WITH ADDRESS OF CORE IMAGE LIBRARY DATA ADDRESS
LOAD REG 7 WITH ADDRESS OF PTCIL24
RESET ENDSWI X'02' TO IND COMPLETE INST PATCHED
CILPATCH


CILPATCH
END OF CORE IMAGE LIB RCD
YES
NO
TAPEIOCL
SAVE REG 3 AND REG 4 IN SAVE2
MOVE CHAR TO CORE IMAGE LIB RCD
WRITERTN PH
WRITE RCD ON SYSRES
UPDATE ADDR OF CORE IMAGE LIB INST
CLEAR REG 1
UPDATE ADDRESS OF FIELD TO BE MOVED
LOAD REG 1 WITH RECORD COUNTER
COMPLETE FIELD MOVED
INCREASE THE RECORD COUNTER BY ONE
BRANCH TO ADDR IN REG 7
STORE RESULT BACK IN RECORD COUNTER
READRTN PH
READ CIL RCD FROM SYSRES
AUTOPATCH TABLE ENTRY REQ
RESTORE REGS 3 & 4. MOVE START ADDR OF CIL RCD INTO REG 2
SVCIL
CILEXC42
LOAD LENGTH OF USER INSTRUCTION INTO REG 4
TURN ENDSW ON TO IND NO AUTOTEST TABLE ENTRY
CILPATCH PG
INSERT INTO CORE IMAGE LIB NEW INSTRUCTION
CILNOP PG
INITIALIZE TO INSERT NO-OP FILLER
INSERT NO-OP FILLER
PRINTPAT PK
WRITE CARD IMAGE ON SYSLST
FTCH


PB J3
CILEXC46 B1
PNTID PK
SET UP SVC IN OUTPUT AREA
C1
PRINTPAT PK
WRITE CARD IMAGE ON SYSLST
D1
STORE PATCH POINT ADDRESS IN CILADR
E1
LOAD REG 3 WITH ADDRESS OF DISPLACED INSTRUCTION WORK AREA
F1
LOAD REG 7 WITH ADDRESS OF CIL24
PF G3
SVCIL
PB M2
CILEXC62 B2
LOAD 2 (LNG OF USER INSTRUCTION) INTO REG 4
C2
CILPATCH PG
INSERT USER INSTRUCTION INTO CIL
D2
MOVE 4 (LNG OF NECESSARY NO-OPS) INTO REG 4
E2
LOAD REG 3 WITH ADDRESS OF NO-OPS
F2
CILPATCH PG
INSERT INTO CIL THE NEC-ESSARY NO-OP FILLER
G2
PRINTPAT PK
WRITE CARD IMAGE ON SYSLST
PC E2
FTCH
PB M2
CILEXC64 B3
LOAD 4 (LNG OF USER INSTRUCTION) INTO REG 4
C3
CILPATCH PG
INSERT USER INSTRUCTION INTO CIL
D3
MOVE 2 (LNG OF NECESSARY NO-OPS) INTO REG 4
E3
LOAD REG 3 WITH ADDRESS OF NO-OPS
F3
CILPATCH PG
INSERT INTO CIL THE NECESSARY NO-OP FILLER
G3
PRINTPAT PK
WRITE CARD IMAGE ON SYSLST
PC E2
FTCH
PA F2
ADDRTN B4
LOAD REG 3 WITH ADDR OF DISPLACED INSTRUCTION WORK AREA
C4
STORE PATCH POINT ADDRESS IN CILADR
D4
CILNTH PQ
CHECK CIL INSTRUCTION LENGTH
E4
STORE THAT LENGTH IN LNTCNTR
F4
SVCIL PF
SAVE CORE IMAGE LIB INST
G4
CILADD PQ
INITIALIZE TO INSERT SVC CODE AT CIL PATCH PT
1
1
F5
LOAD REG 2 WITH POINTER TO CORE IMAGE LIB PATCH PT
G5
CILPATCH PG
INSERT SVC CODE AT CIL PATCH POINT
H5
WAS CIL INSTRUCTION 2 BYTES
NO
PD B4
CILNOPP
YES
PD C3
PTCIL24


CONCONT A2
AUTOTEST CARD
YES
NO
CONCONT B2
JOB CONTROL CARD
YES
PF K1
MISPCC
NO
PA G7
CONRTN B1
CONADR PQ
INITIALIZE POINTER TO PROCESS CONSTANT
C1
CONTRANS PR
TRANSLATE CONSTANT CHARACTERS
D1
LOAD REG 4 WITH CONSTANT LENGTH
E1
DIVIDE LENGTH BY TWO
F1
CILCON PR
INSERT CONSTANT IN CORE IMAGE LIB RCD
G1
MOVE CARD IMAGE BACK FROM WORK AREA
H1
PRINTPAT PK
WRITE CARD IMAGE ON SYSLST
J1
READPAT PM
READ CARD FROM SYSIPT
CONAST C2
CHCKAST PN
VALIDATE * ON CARD
D2
VALIDHEX PN
VALIDATE HEX PARAM ON CD AND GET LENGTH
F2
STORE LENGTH IN CONCOUNT AND EXCLNTH
G2
ERR COND OF ODD NO. OF CHARS
YES
NO
H2
TESTER PO
TEST IF OTHER ERROR REQUIRES ERROR RTN
J2
SAVE CARD IMAGE IN SAVE AREA
CARDCON B3
TURN OFF CARD SWITCH
C3
WRITERTN PM
WRITE RCD ON SYSRES
D3
LOAD REG 1 WITH 'ATLEFC2'
E3
LOAD REG 0 WITH 'CARDID'
SVC 1 ATLEFC2 (CARDID)
* PN-G1 PN-C2 PN-C3 PN-D4 PN-G3 PP-J3 PP-J4 PP-C4 PP-E3 PP-F5 PP-F1 PJ-J2 PJ-J8
ERINST G3
ALTER MVCODE INST TO MOVE AN 'E'
** PN-F2 PG-G1 PG-E2
H3
ERRTN PJ
PUT ERROR CODE IN PRINT ERROR TAB AND WRITE LINE
ERPNT J3
PRINTPAT PK
WRITE CARD IMAGE ON SYSLST
B4
READPAT PM
READ CARD FROM SYSIPT
C4
AUTOTEST CONTROL CARD
YES
PE D4
FTCHID
NO
D4
LAST CARD A CONSTANT CARD
NO
PF B2
CARDAST
YES
MVCODE F4
MOVE ERROR CODE TO PRINT AREA
G4
INCREASE ERROR TABLE ADDRESS BY 2
H4
PUT UPDATED ADDRESS IN POINTER
J4
TURN ON ERROR SWITCH
K4
BRANCH TO ADDR IN REG 12
D5
ERRTN
ERRTN E5
LOAD REG 1 WITH ADDR OF ERROR TABLE IN PRINT AREA
F5
TABLE FILLED
NO
YES
CNCLPCH G5
LOAD REG 1 WITH ADDRESS OF ERROR TBL IN PRINT AREA
H5
STORE ERROR TBL ADDRESS IN SAVE AREA
J5
PRINTPAT PK
WRITE CARD IMAGE ON SYSLST
PC E2
FTCH

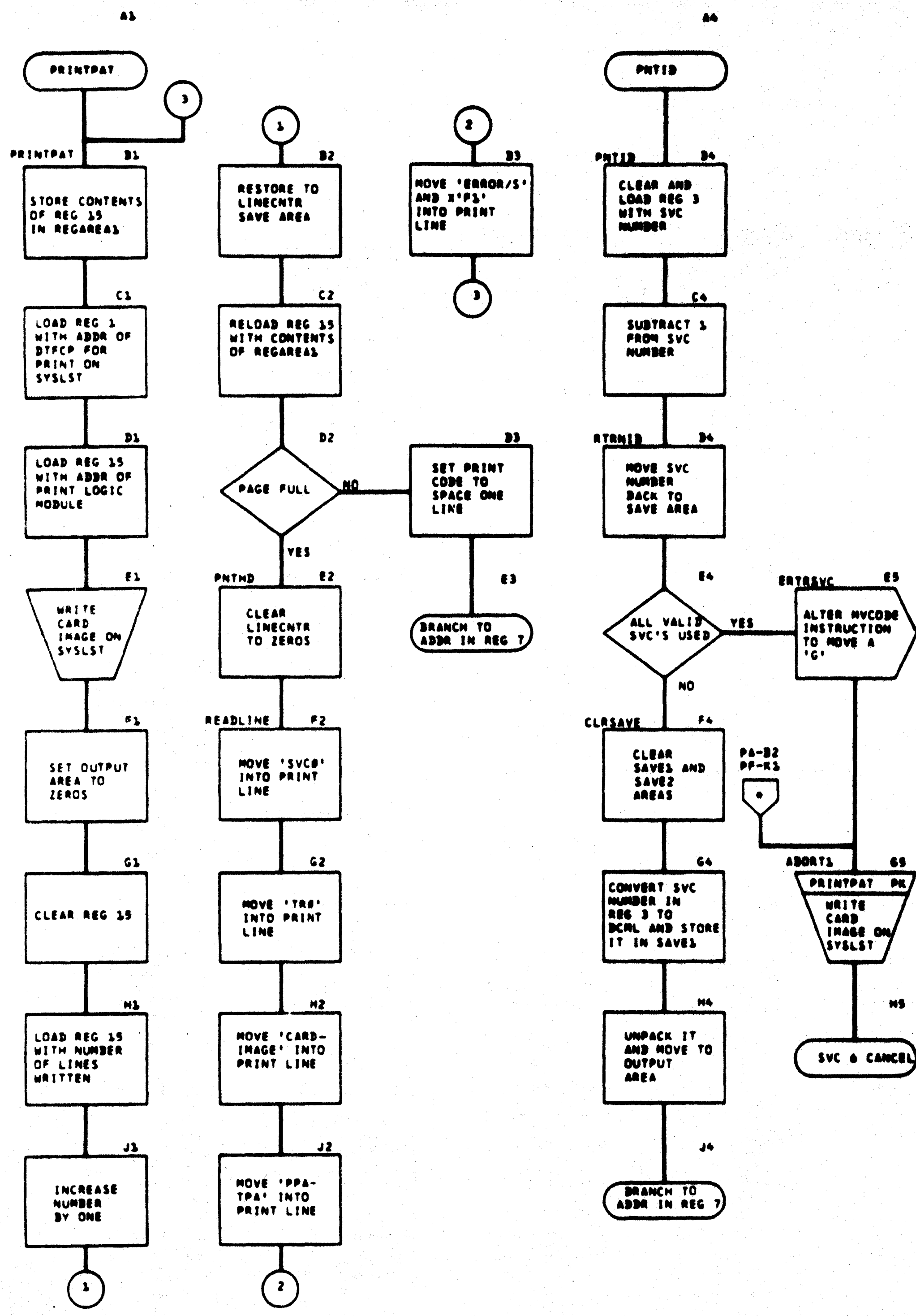
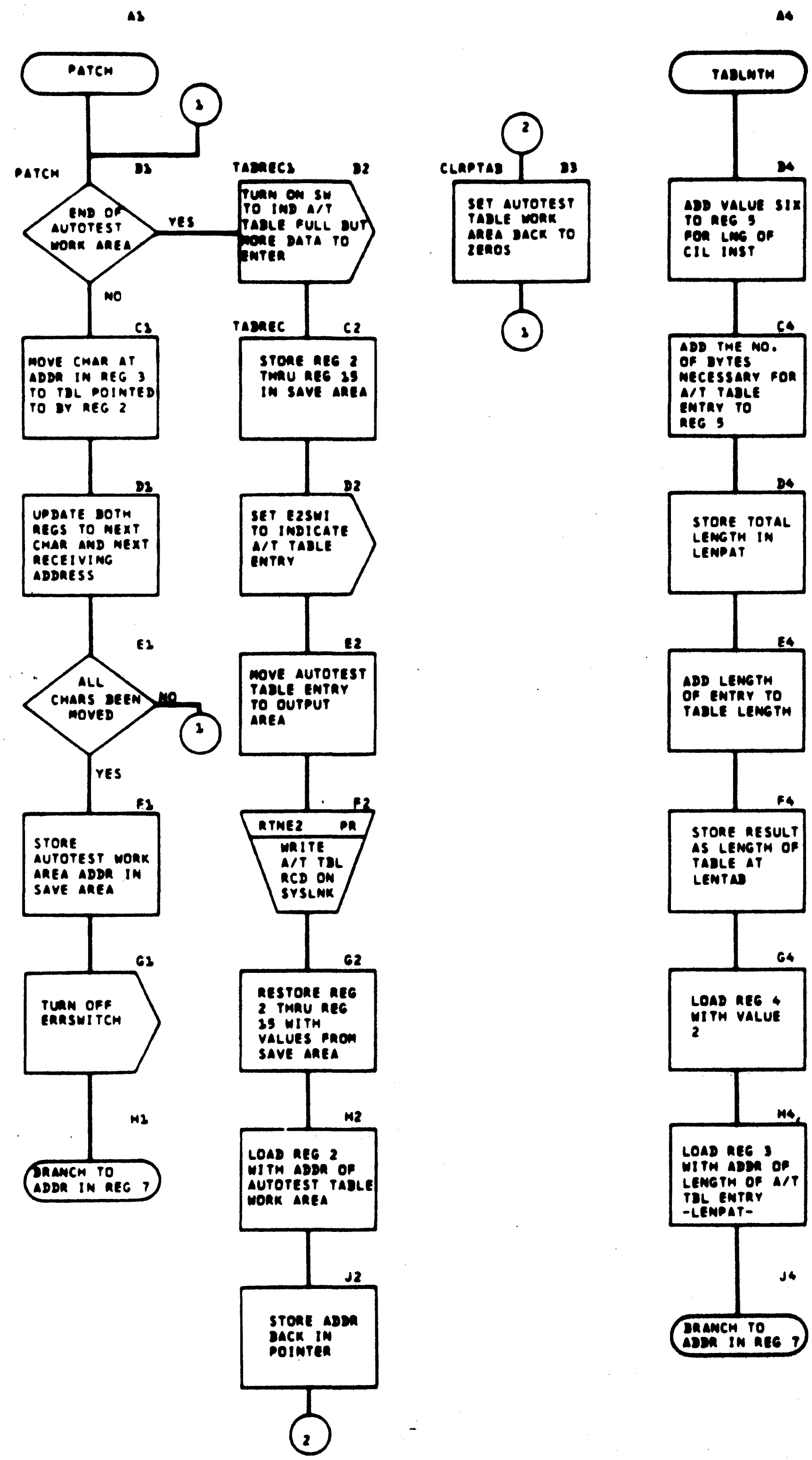
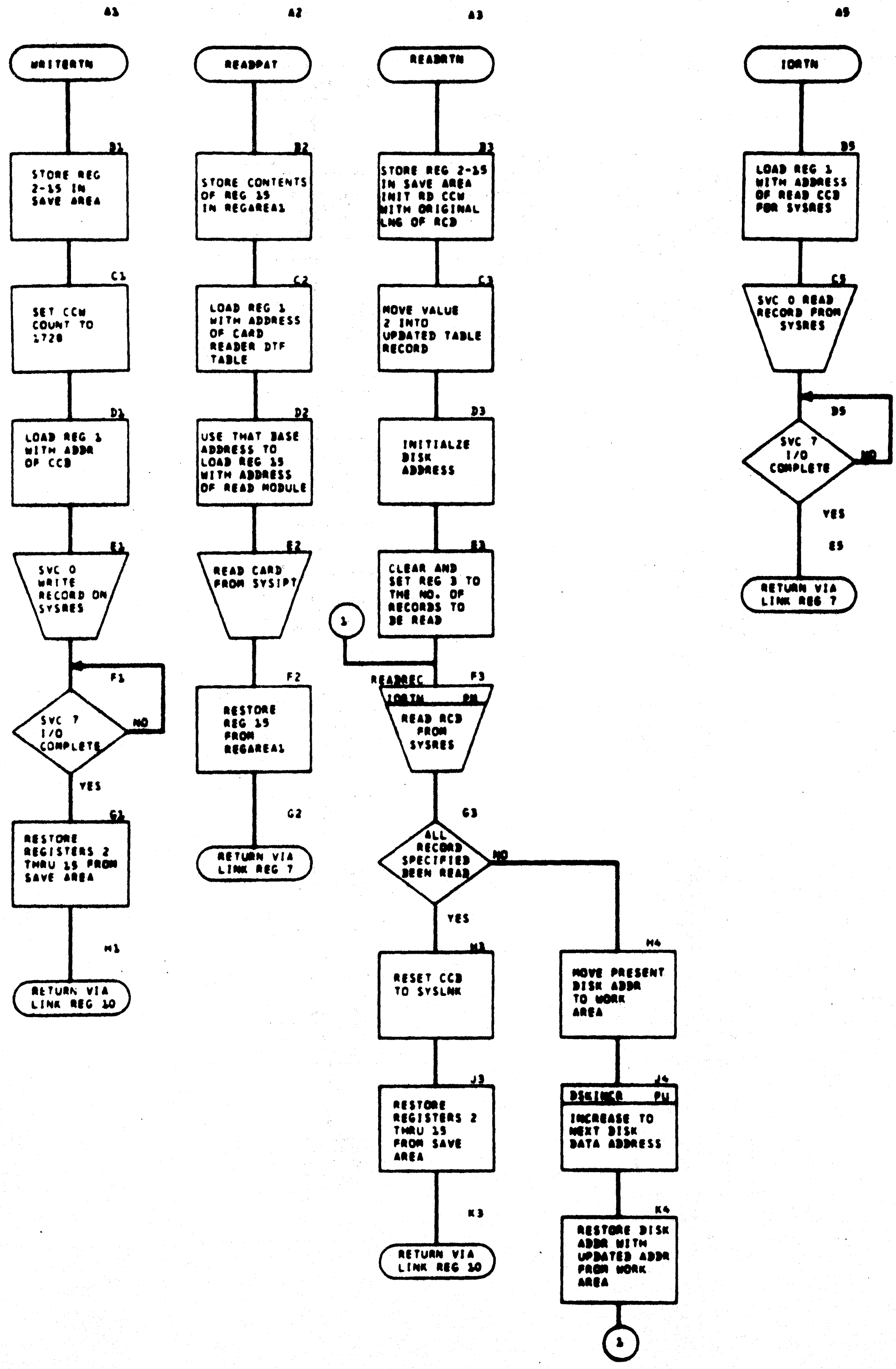
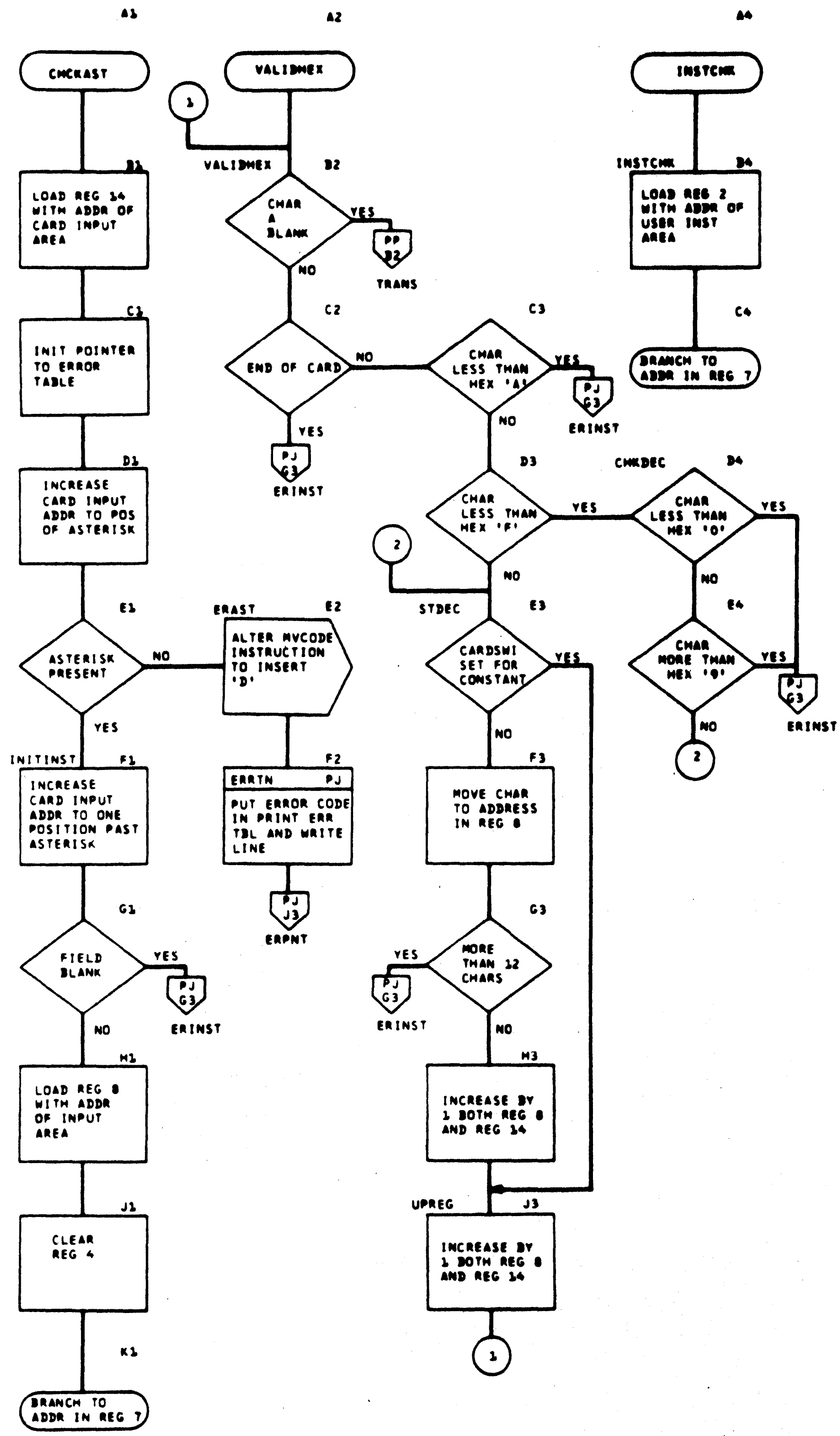
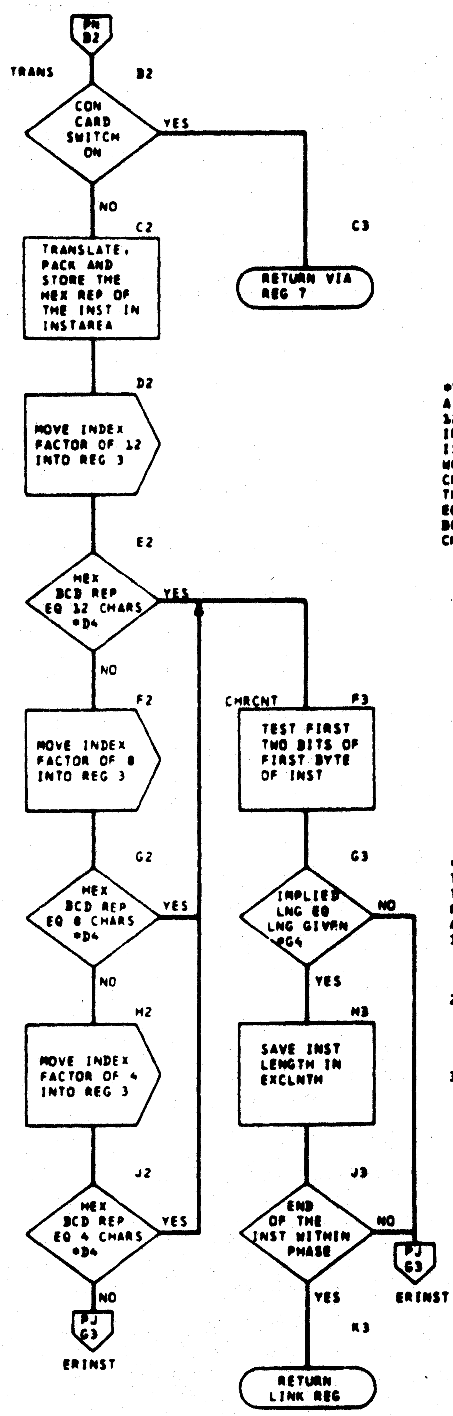

PRINTPAT
STORE CONTENTS OF REG 15 IN REGAREA1
LOAD REG 1 WITH ADDR OF DTFCP FOR PRINT ON SYSLST
LOAD REG 15 WITH ADDR OF PRINT LOGIC MODULE
WRITE CARD IMAGE ON SYSLST
SET OUTPUT AREA TO ZEROS
CLEAR REG 15
LOAD REG 15 WITH NUMBER OF LINES WRITTEN
INCREASE NUMBER BY ONE
RESTORE TO LINECNTR SAVE AREA
RELOAD REG 15 WITH CONTENTS OF REGAREA1
PAGE FULL
NO
YES
PNTHD
CLEAR LINECNTR TO ZEROS
READLINE
MOVE 'SVC#' INTO PRINT LINE
MOVE 'TR#' INTO PRINT LINE
MOVE 'CARD-IMAGE' INTO PRINT LINE
MOVE 'PPA-TPA' INTO PRINT LINE
MOVE 'ERROR/S' AND X'F1' INTO PRINT LINE
SET PRINT CODE TO SPACE ONE LINE
BRANCH TO ADDR IN REG 7
PNTID
CLEAR AND LOAD REG 3 WITH SVC NUMBER
SUBTRACT 1 FROM SVC NUMBER
RTRNID
MOVE SVC NUMBER BACK TO SAVE AREA
ALL VALID SVC'S USED
YES
NO
ERTRSVC
ALTER MVCODE INSTRUCTION TO MOVE A 'G'
CLRSAVE
CLEAR SAVE1 AND SAVE2 AREAS
PA-B2
PF-K1
CONVERT SVC NUMBER IN REG 3 TO DCML AND STORE IT IN SAVE1
UNPACK IT AND MOVE TO OUTPUT AREA
BRANCH TO ADDR IN REG 7
ABORT1
PRINTPAT PK
WRITE CARD IMAGE ON SYSLST
SVC 6 CANCEL

Chart FL. Autopatch User Program (ATLEPC7) (Refer to Chart 11) (Part 11 of 19)




WRITERTN
STORE REG 2-15 IN SAVE AREA
SET CCW COUNT TO 1728
LOAD REG 1 WITH ADDR OF CCB
SVC 0 WRITE RECORD ON SYSRES
SVC 7 I/O COMPLETE
NO
YES
RESTORE REGISTERS 2 THRU 15 FROM SAVE AREA
RETURN VIA LINK REG 10
READPAT
STORE CONTENTS OF REG 15 IN REGAREA1
LOAD REG 1 WITH ADDRESS OF CARD READER DTF TABLE
USE THAT BASE ADDRESS TO LOAD REG 15 WITH ADDRESS OF READ MODULE
READ CARD FROM SYSIPT
RESTORE REG 15 FROM REGAREA1
RETURN VIA LINK REG 7
READRTN
STORE REG 2-15 IN SAVE AREA INIT RD CCW WITH ORIGINAL LNG OF RCD
MOVE VALUE 2 INTO UPDATED TABLE RECORD
INITIALZE DISK ADDRESS
CLEAR AND SET REG 3 TO THE NO. OF RECORDS TO BE READ
READREC
IORTN PM
READ RCD FROM SYSRES
ALL RECORD SPECIFIED BEEN READ
NO
YES
RESET CCB TO SYSLNK
RESTORE REGISTERS 2 THRU 15 FROM SAVE AREA
RETURN VIA LINK REG 10
MOVE PRESENT DISK ADDR TO WORK AREA
DSKINCR PU
INCREASE TO NEXT DISK DATA ADDRESS
RESTORE DISK ADDR WITH UPDATED ADDR FROM WORK AREA
IORTN
LOAD REG 1 WITH ADDRESS OF READ CCB FOR SYSRES
SVC 0 READ RECORD FROM SYSRES
SVC 7 I/O COMPLETE
NO
YES
RETURN VIA LINK REG 7


CHCKAST
LOAD REG 14 WITH ADDR OF CARD INPUT AREA
INIT POINTER TO ERROR TABLE
INCREASE CARD INPUT ADDR TO POS OF ASTERISK
ASTERISK PRESENT
YES
NO
ERAST
ALTER MVCODE INSTRUCTION TO INSERT 'D'
ERRTN PJ
PUT ERROR CODE IN PRINT ERR TBL AND WRITE LINE
PJ J3
ERPNT
INITINST
INCREASE CARD INPUT ADDR TO ONE POSITION PAST ASTERISK
FIELD BLANK
YES
PJ G3
ERINST
NO
LOAD REG 8 WITH ADDR OF INPUT AREA
CLEAR REG 4
BRANCH TO ADDR IN REG 7
VALIDHEX
VALIDHEX
CHAR A BLANK
YES
PP B2
TRANS
NO
END OF CARD
YES
PJ G3
ERINST
NO
CHAR LESS THAN HEX 'A'
YES
PJ G3
ERINST
NO
CHAR LESS THAN HEX 'F'
YES
NO
STDEC
CARDSWI SET FOR CONSTANT
YES
NO
MOVE CHAR TO ADDRESS IN REG 8
MORE THAN 12 CHARS
YES
PJ G3
ERINST
NO
INCREASE BY 1 BOTH REG 8 AND REG 14
UPREG
INCREASE BY 1 BOTH REG 8 AND REG 14
CHKDEC
CHAR LESS THAN HEX '0'
YES
NO
CHAR MORE THAN HEX '9'
YES
PJ G3
ERINST
NO
INSTCHK
INSTCHK
LOAD REG 2 WITH ADDR OF USER INST AREA
BRANCH TO ADDR IN REG 7

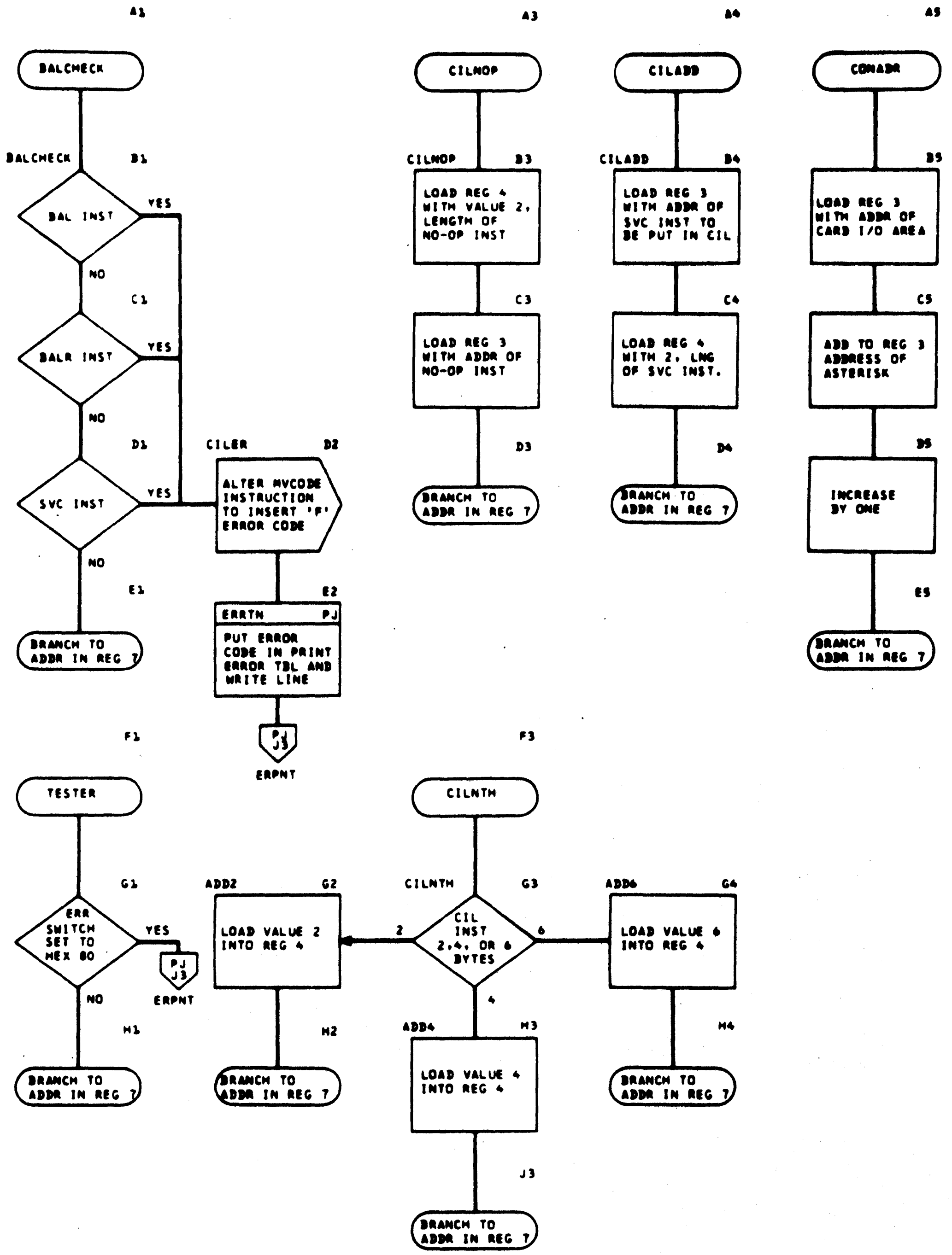


*D4
A LENGTH VALUE OF 12, 8 OR 4 IS MOVED INTO REG 3 AND REG 3 IS COMPARED TO REG 4 WHICH CONTAINS THE CHARACTER LENGTH OF THE INST. WHEN AN EQUAL IS FOUND, A BRANCH IS TAKEN TO CHRCNT.

*G4
THE YES BRANCH IS TAKEN ONLY WHEN THE FOLLOWING CONDITIONS ARE MET:

1. THE INDEX IS 12 AND BOTH OF THE FIRST TWO BITS OF THE INST ARE ON.
2. THE INDEX IS 8 AND EITHER ONE OF THE FIRST 2 BITS OF THE INST ARE ON.
3. THE INDEX IS 4 AND NEITHER OF THE FIRST 2 BITS OF THE IN'T ARE ON.

Chart PQ. Autopatch User Program (ATLEFC7) (Refer to Chart 11) (Part 15 of 19)



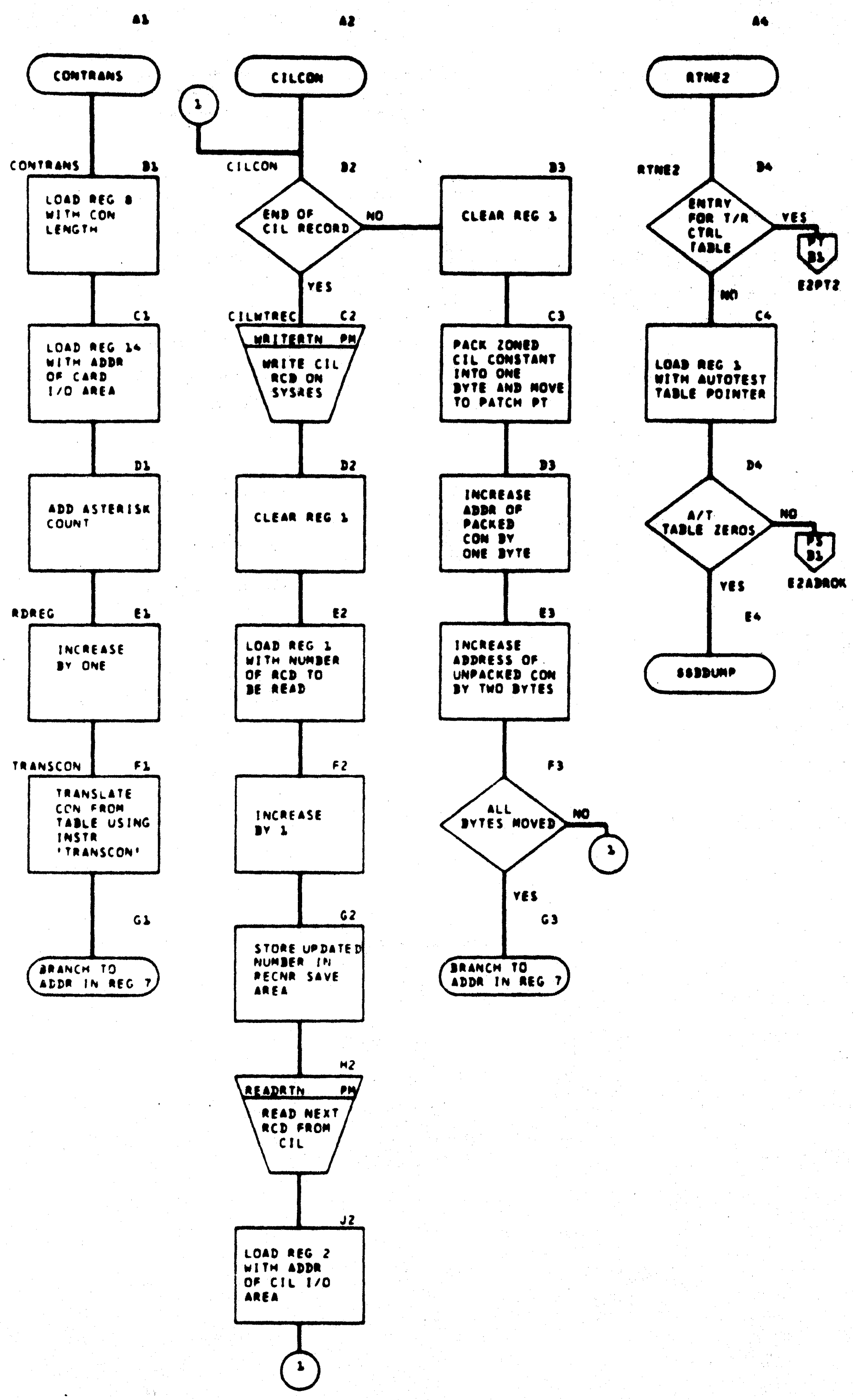
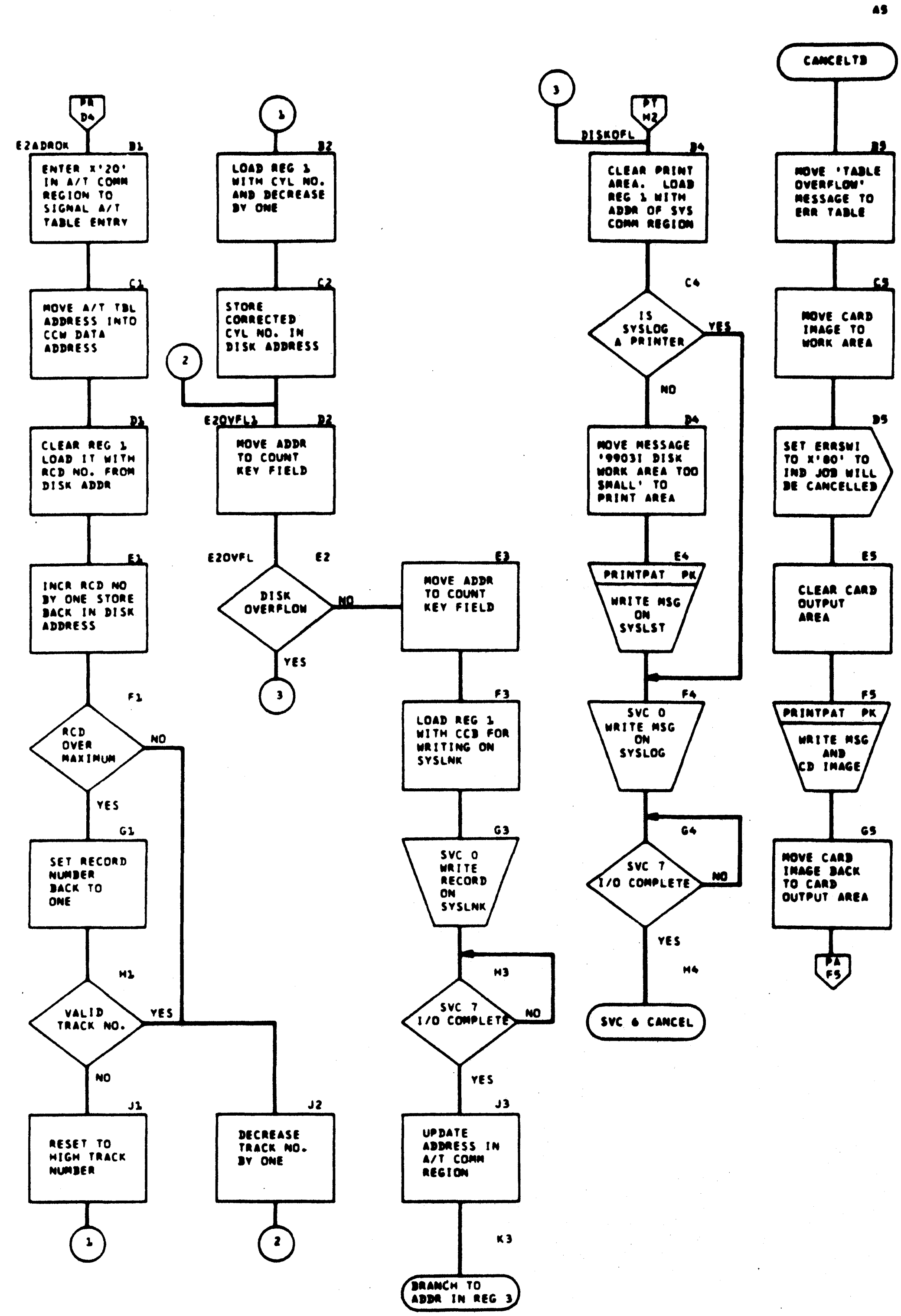
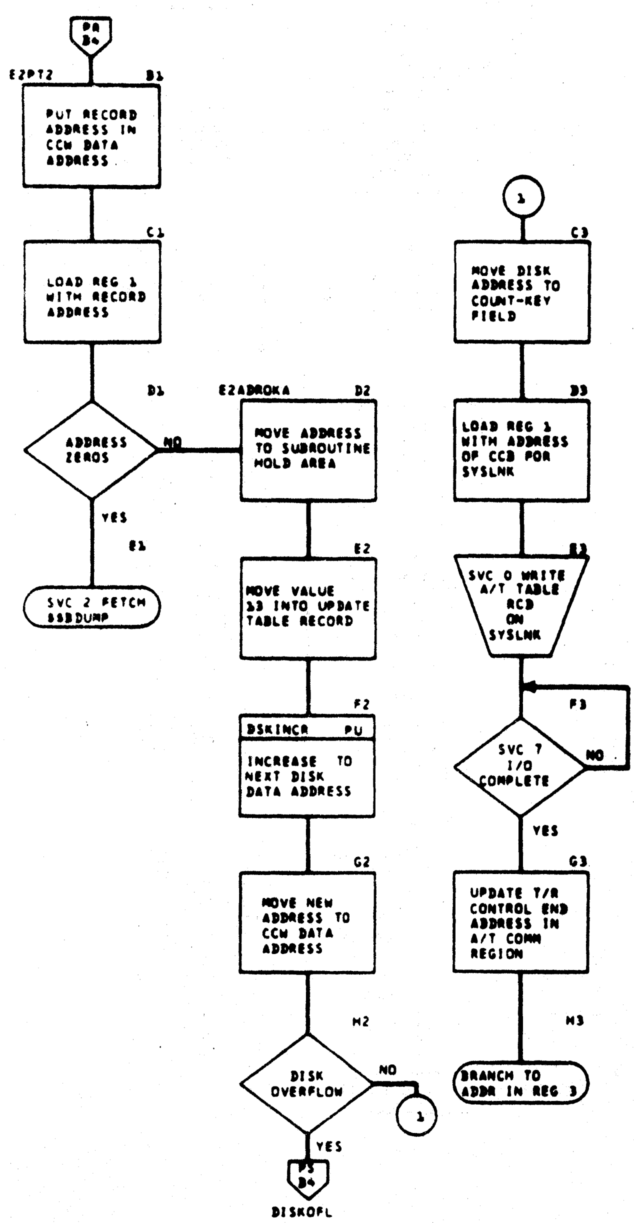
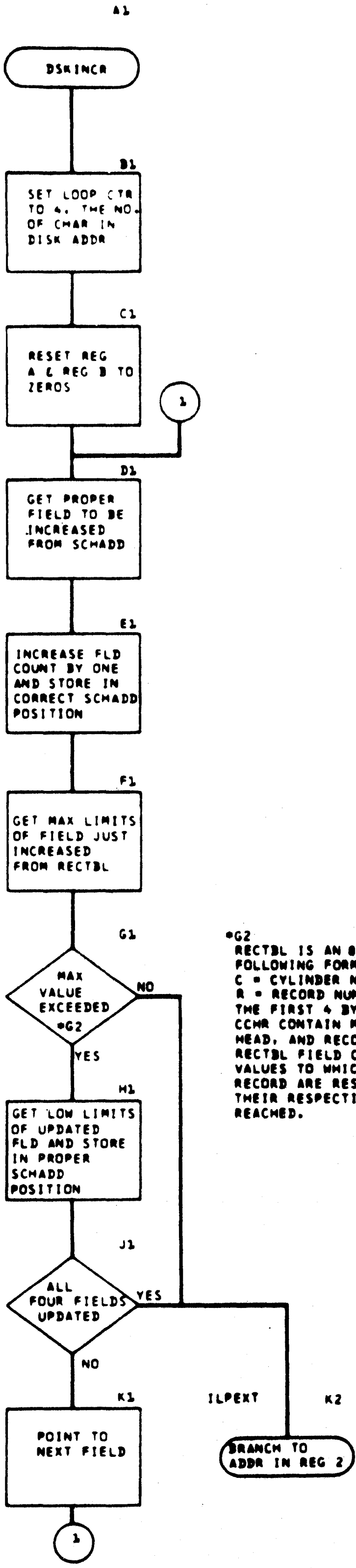

CONTRANS
LOAD REG 8 WITH CON LENGTH
LOAD REG 14 WITH ADDR OF CARD I/O AREA
ADD ASTERISK COUNT
RDREG
INCREASE BY ONE
TRANSCON
TRANSLATE CON FROM TABLE USING INSTR 'TRANSCON'
BRANCH TO ADDR IN REG 7
CILCON
END OF CIL RECORD
NO
YES
CILWTREC
WRITERTN PM
WRITE CIL RCD ON SYSRES
CLEAR REG 1
LOAD REG 1 WITH NUMBER OF RCD TO BE READ
INCREASE BY 1
STORE UPDATED NUMBER IN RECNR SAVE AREA
READRTN PM
READ NEXT RCD FROM CIL
LOAD REG 2 WITH ADDR OF CIL I/O AREA
CLEAR REG 1
PACK ZONED CIL CONSTANT INTO ONE BYTE AND MOVE TO PATCH PT
INCREASE ADDR OF PACKED CON BY ONE BYTE
INCREASE ADDRESS OF UNPACKED CON BY TWO BYTES
ALL BYTES MOVED
BRANCH TO ADDR IN REG 7
RTNE2
ENTRY FOR T/R CTRL TABLE
E2PT2
LOAD REG 1 WITH AUTOTEST TABLE POINTER
A/T TABLE ZEROS
E2ADROK
$$BDUMP

Chart PS. Autopatch User Program (ATLEFC7) (Refer to Chart 11) (Part 17 of 19)





PR B4
E2PT2 B1
PUT RECORD ADDRESS IN CCW DATA ADDRESS
C1
LOAD REG 1 WITH RECORD ADDRESS
D1
ADDRESS ZEROS
NO
YES
E1
SVC 2 FETCH $$BDUMP
E2ADROKA D2
MOVE ADDRESS TO SUBROUTINE HOLD AREA
E2
MOVE VALUE 13 INTO UPDATE TABLE RECORD
F2
DSKINCR PU
INCREASE TO NEXT DISK DATA ADDRESS
G2
MOVE NEW ADDRESS TO CCW DATA ADDRESS
H2
DISK OVERFLOW
NO
1
YES
PS B4
DISKOFL
1
C3
MOVE DISK ADDRESS TO COUNT-KEY FIELD
D3
LOAD REG 1 WITH ADDRESS OF CCB FOR SYSLNK
E3
SVC 0 WRITE A/T TABLE RCD ON SYSLNK
F3
SVC 7 I/O COMPLETE
NO
YES
G3
UPDATE T/R CONTROL END ADDRESS IN A/T COMM REGION
H3
BRANCH TO ADDR IN REG 3



*G2
RECTBL IS AN 8-BYTE FIELD IN THE FOLLOWING FORMAT: CCHRCCHR, WHERE C = CYLINDER NUMBER, H = HEAD NUMBER, R = RECORD NUMBER. .
THE FIRST 4 BYTES OF RECTBL FIELD CCHR CONTAIN MAX VALUES OF CYLINDER, HEAD, AND RECORD. THE NEXT 4 BYTES OF RECTBL FIELD CCHR CONTAIN THE MINIMUM VALUES TO WHICH CYLINDER, HEAD, AND RECORD ARE RESET WHEN MAX VALUE FOR THEIR RESPECTIVE FIELD HAS BEEN REACHED.

Chart QA. Autotest Control Program Initialization (ATLEFE1) (Refer to Chart 12) (Part 1 of 2)

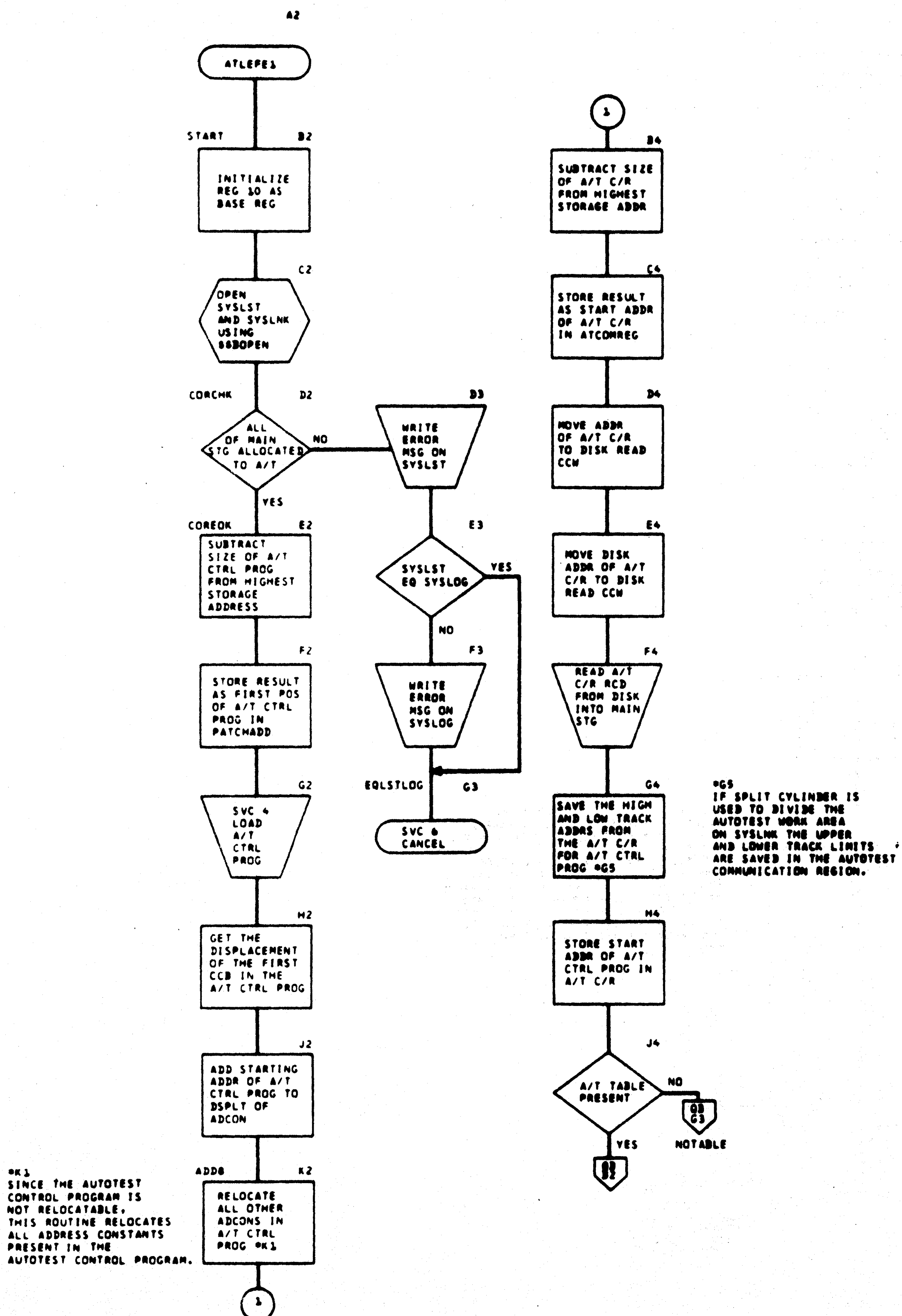


Chart QB. Autotest Control Program Initialization (ATLEFE1) (Refer to Chart 12) (Part 2 of 2)

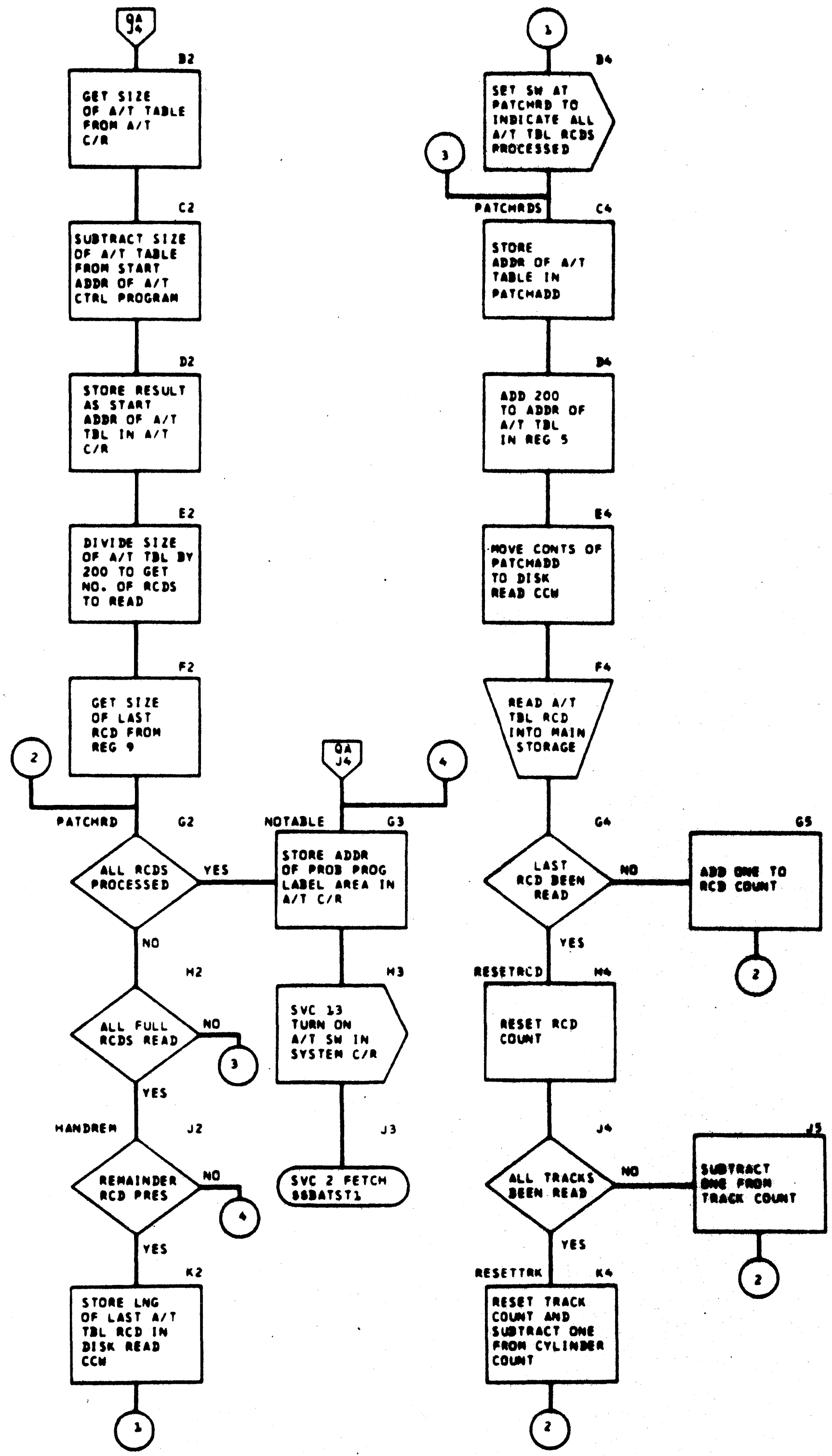


Chart RA. Set up Interface Between Autotest Control Program and Supervisor ($$BATST1) (Refer to Chart 12)

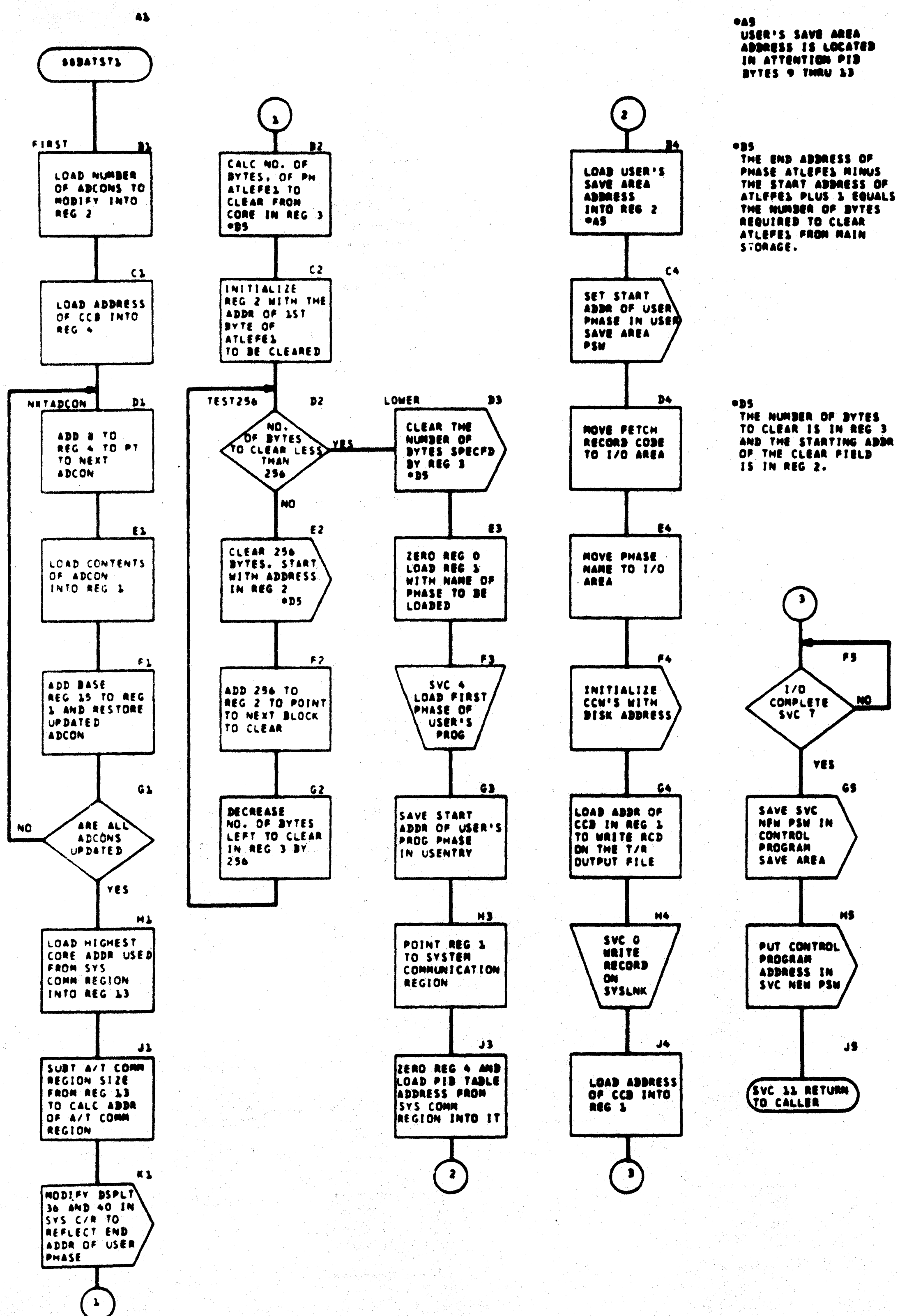


Chart 8A. Autotest Control Program Execution (ATLEFE2) (Refer to Chart 12) (Part 1 of 4)

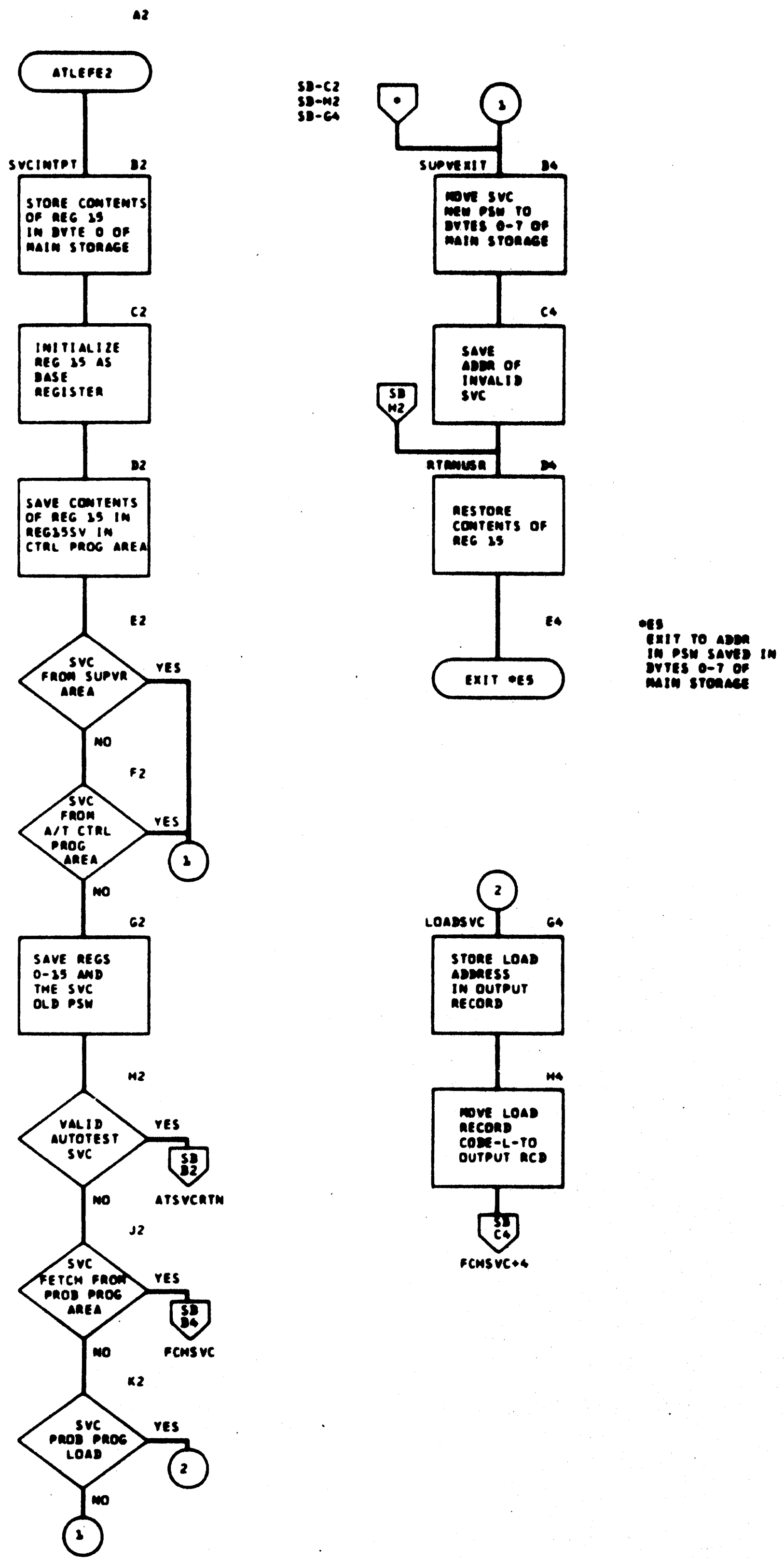


Chart 5B. Autotest Control Program Execution (ATLRPE2) (Refer to Chart 12) (Part 2 of 4)

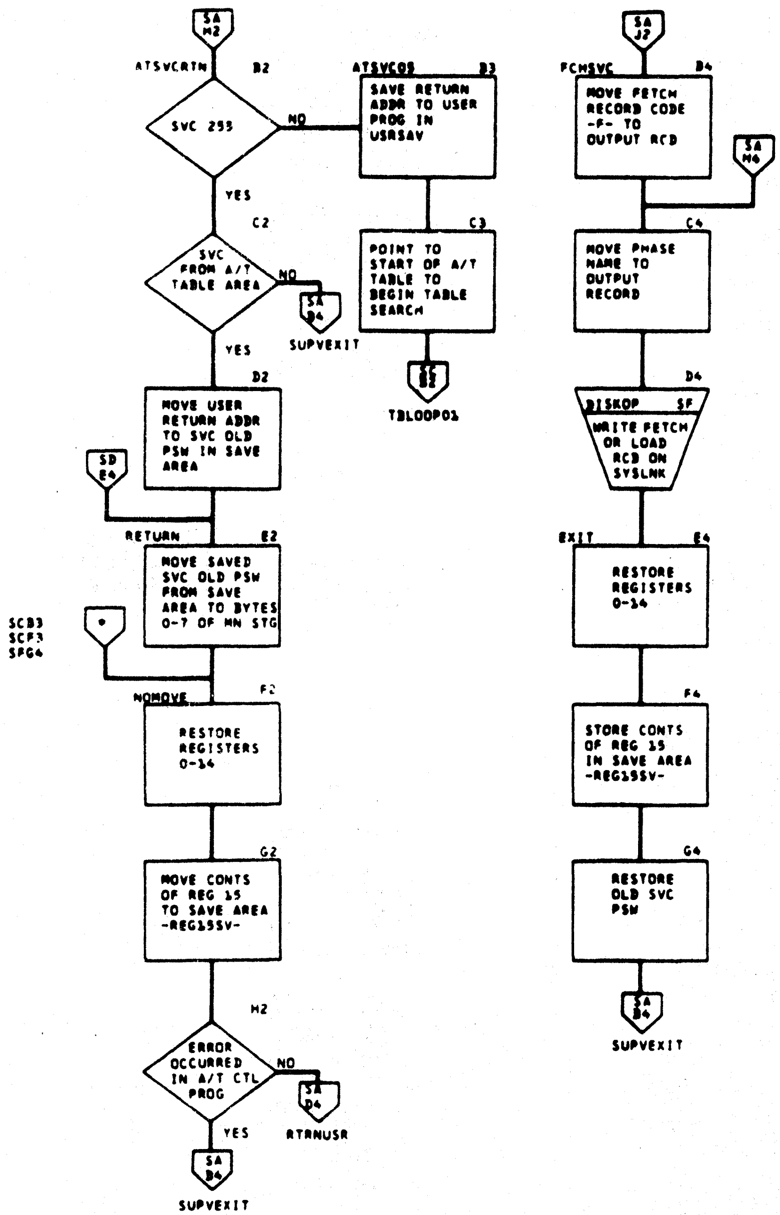


Chart 8C. Autotest Control Program Execution (ATLEFE2) (Refer to Chart 12) (Part 3 of 4)

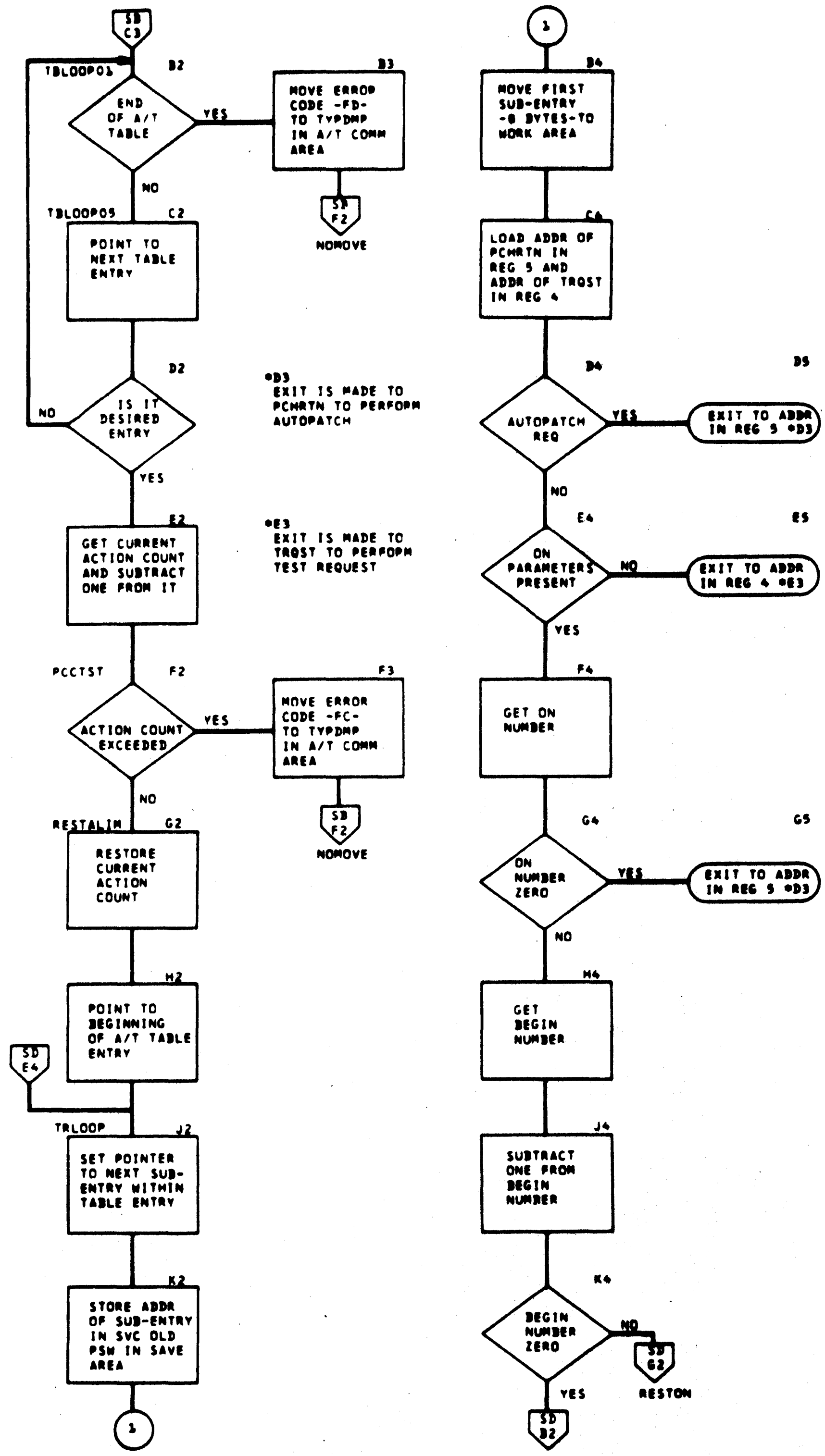


Chart SD. Autotest Control Program Execution (ATLEFE2) (Refer to Chart 12) (Part 4 of 4)

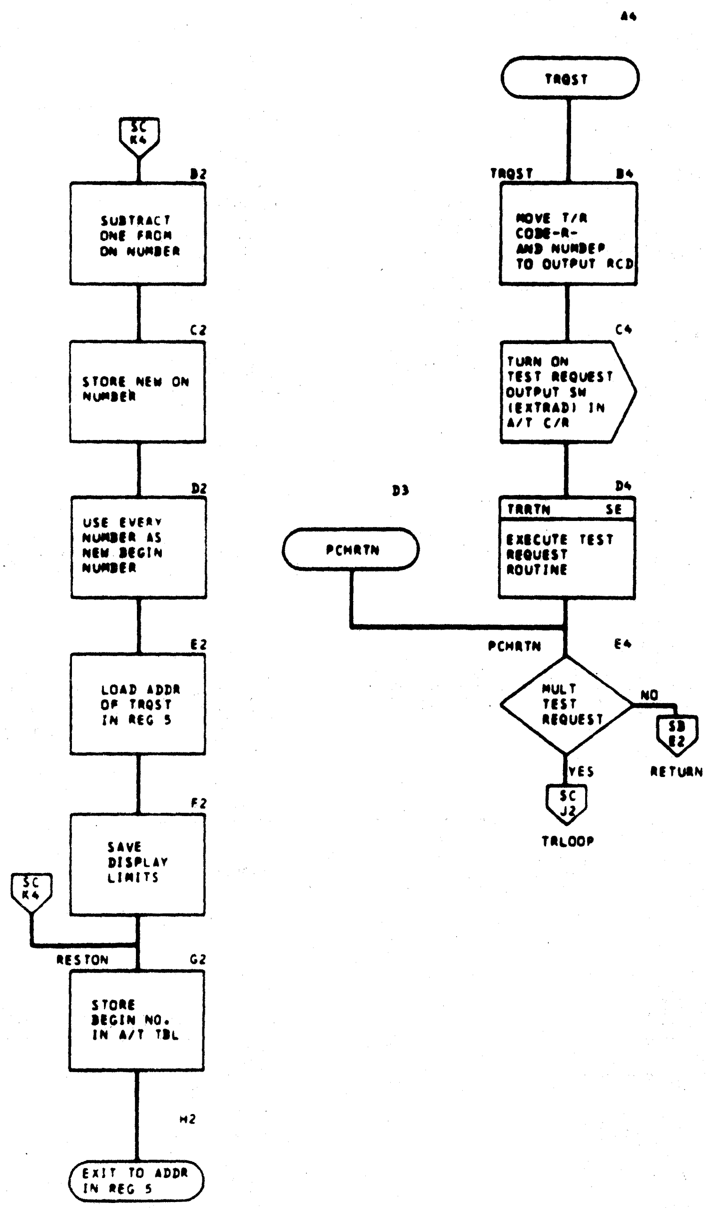


Chart 8E. Autotest Control Program Execution Subroutines (ATLEFE2) (Refer to Chart 12) (Part 1 of 2)

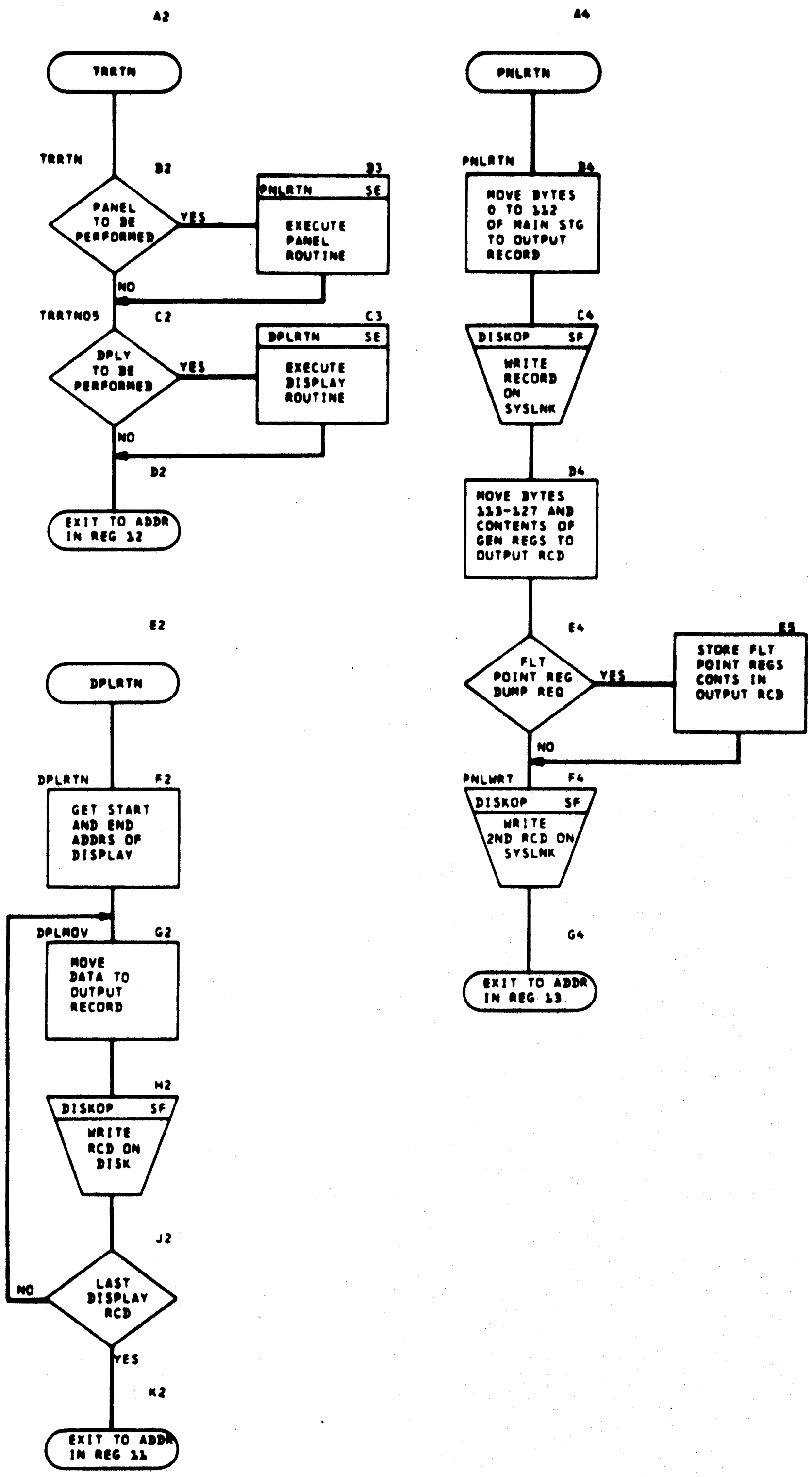
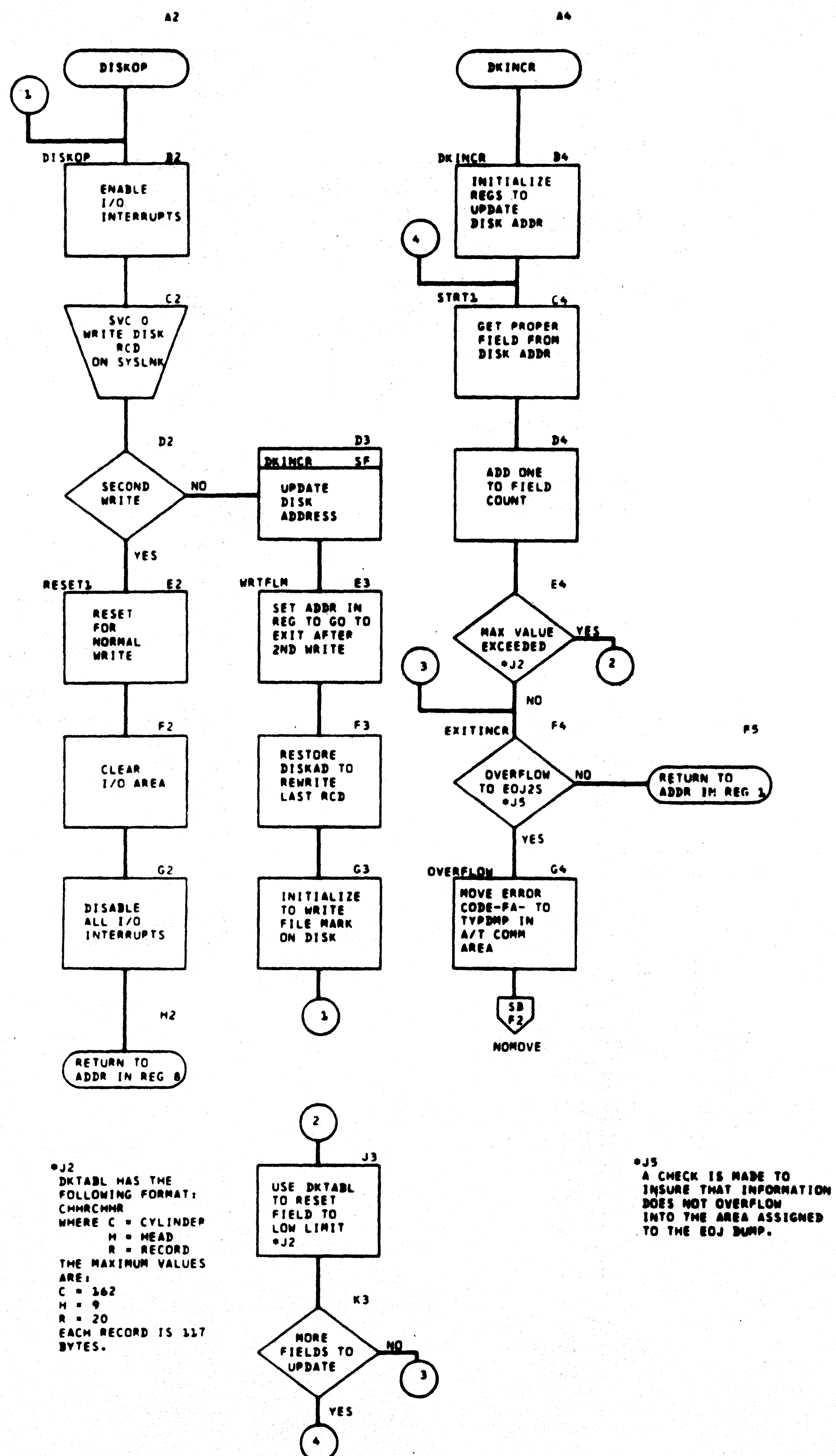
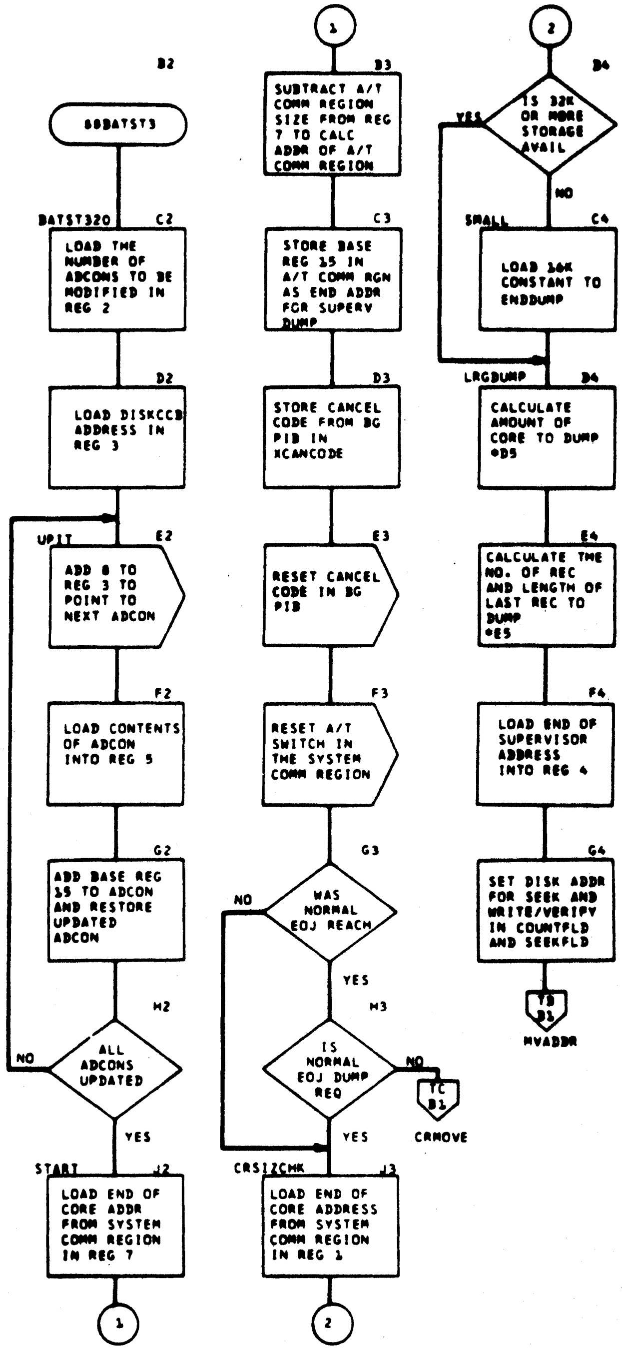


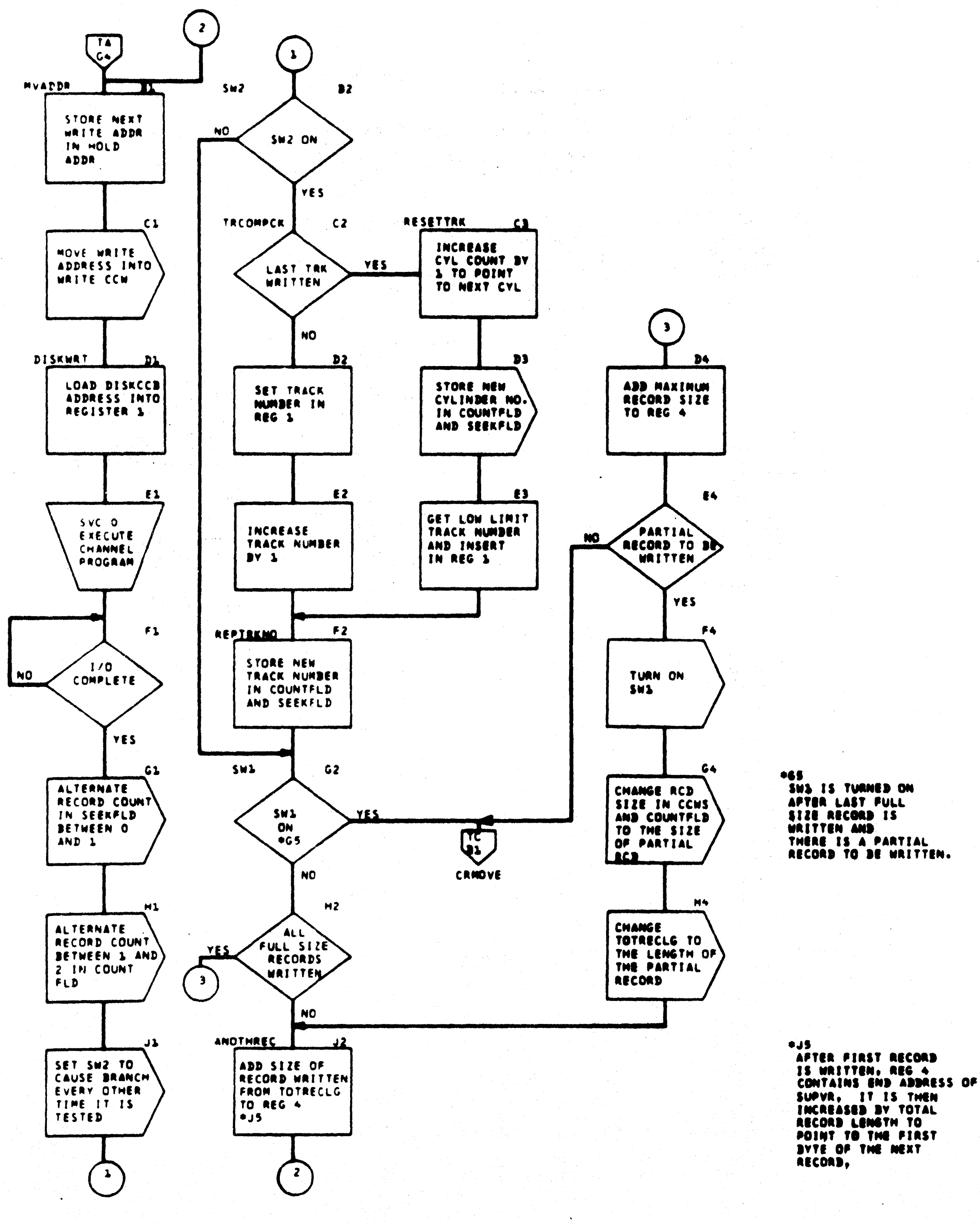
Chart SF. Autotest Control Program Execution Subroutines (ATLEFE2) (Refer to Chart 12) (Part 2 of 2)



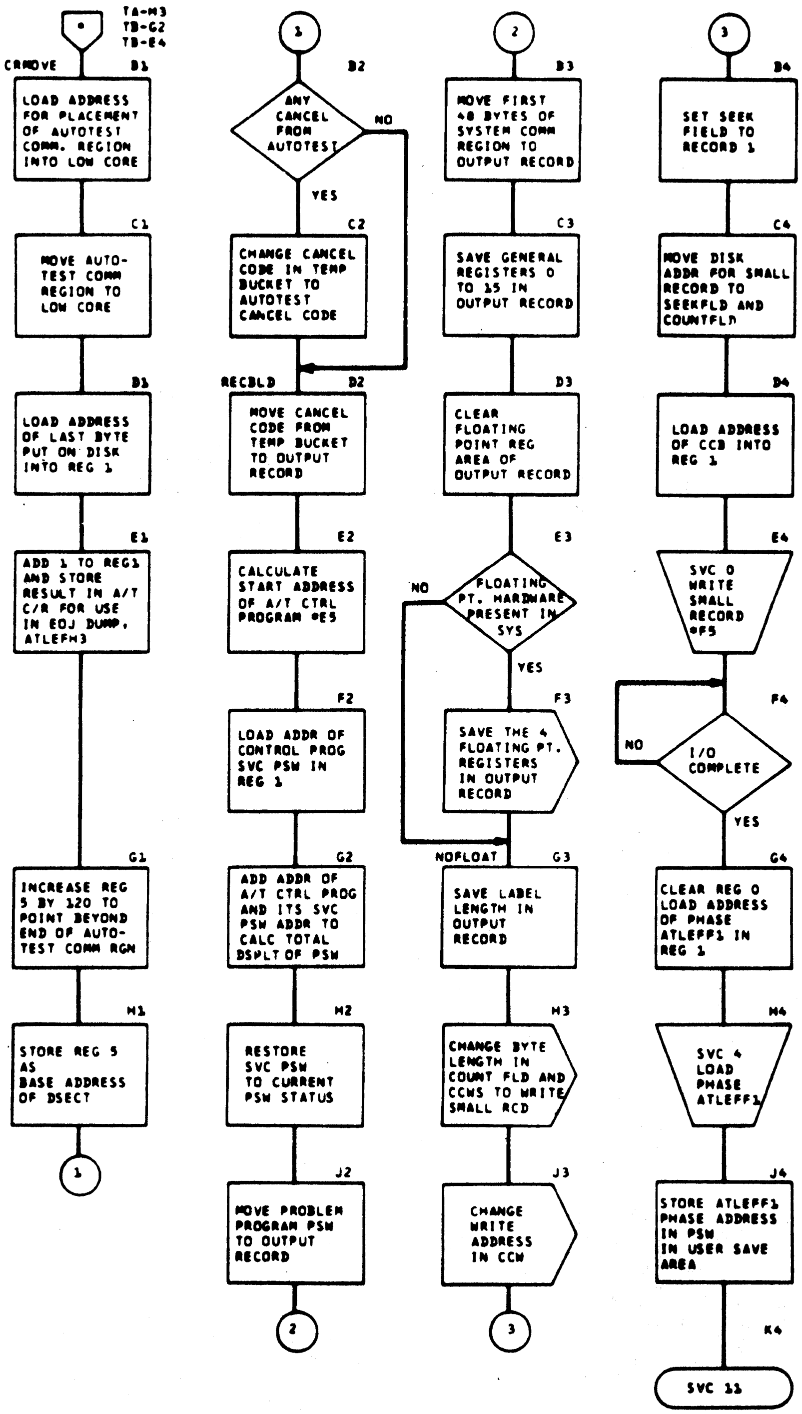


*D5
LOAD CORE SIZE FROM ENDDUMP IN REG 9 AND SUBTRACT THE LENGTH OF SUPERVISOR. ENDDUMP IS 16K FOR 16K MACHINE OR 32K FOR MACHINE 32K OR GREATER.

*E5
DIVIDE THE AMOUNT OF CORE TO DUMP BY THE RECORD LENGTH. REG 9 WILL CONTAIN THE NUMBER OF FULL LENGTH RECORDS AND REG 8 WILL CONTAIN THE REMAINDER (LENGTH OF LAST RECORD).

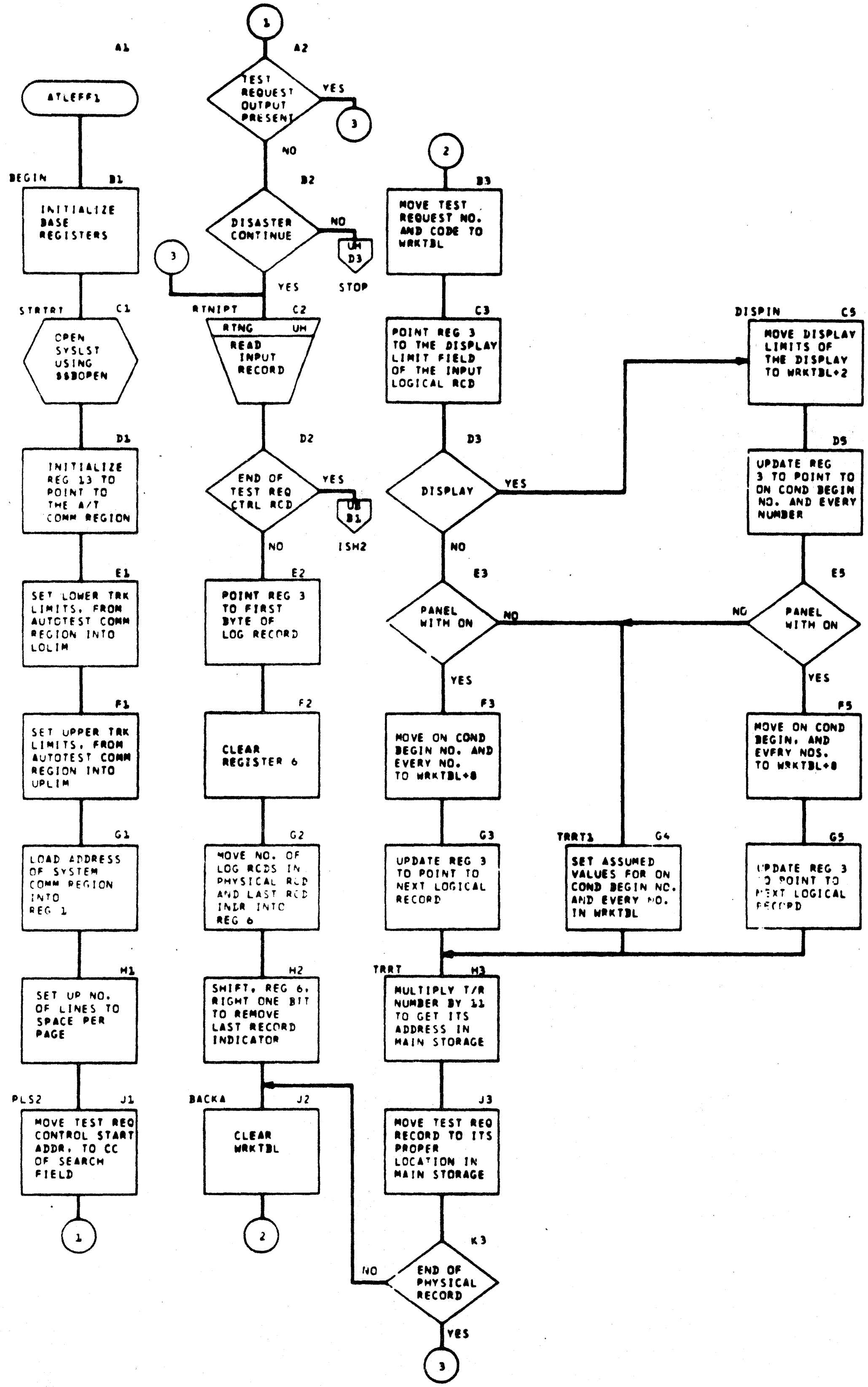


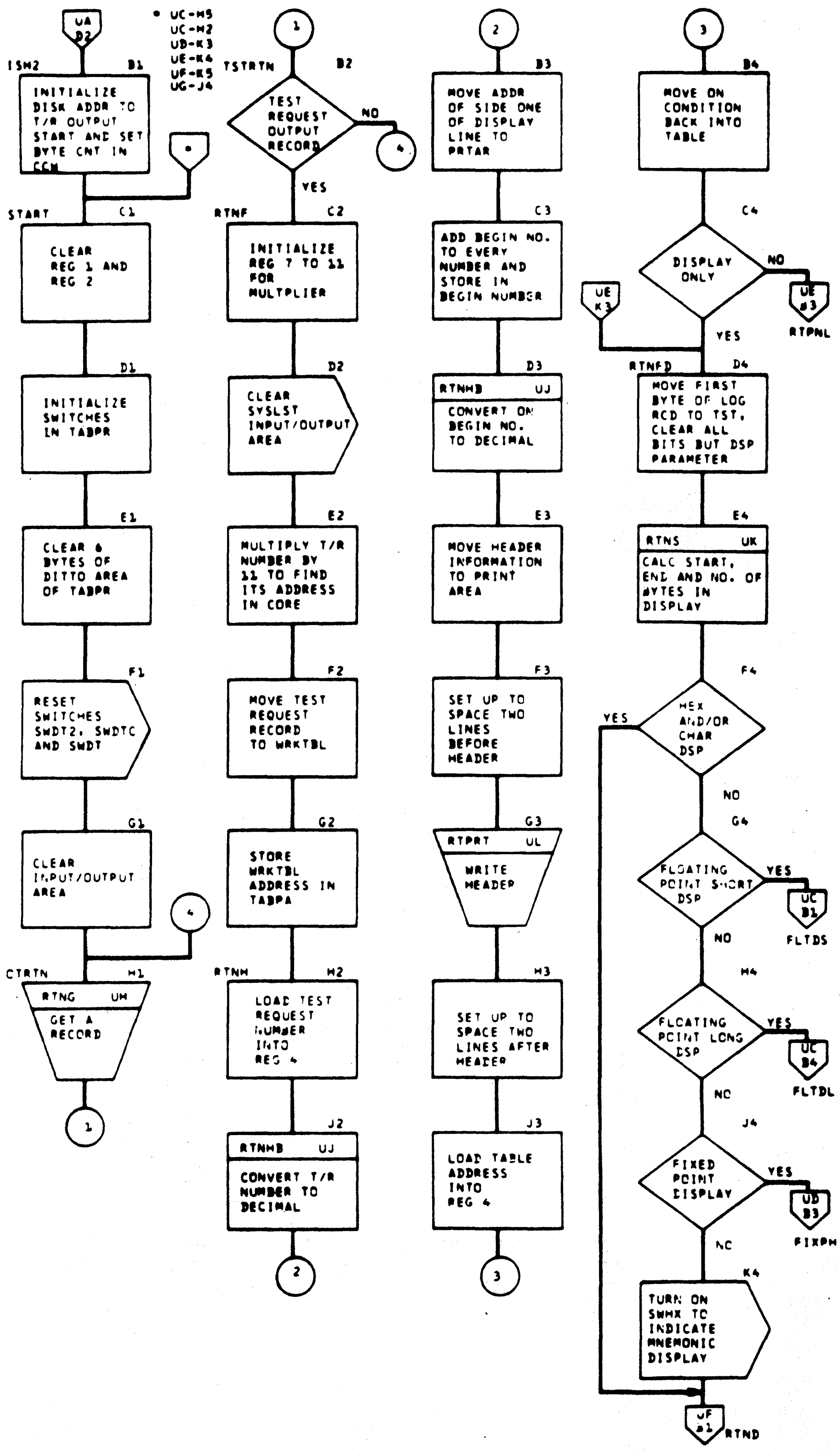
TA G4
MVADDR B1
STORE NEXT WRITE ADDR IN HOLD ADDR
C1
MOVE WRITE ADDRESS INTO WRITE CCW
DISKWRT D1
LOAD DISKCCB ADDRESS INTO REGISTER 1
E1
SVC 0 EXECUTE CHANNEL PROGRAM
F1
I/O COMPLETE
NO
YES
G1
ALTERNATE RECORD COUNT IN SEEKFLD BETWEEN 0 AND 1
H1
ALTERNATE RECORD COUNT BETWEEN 1 AND 2 IN COUNT FLD
J1
SET SW2 TO CAUSE BRANCH EVERY OTHER TIME IT IS TESTED
SW2 B2
SW2 ON
NO
YES
TRCOMPCK C2
LAST TRK WRITTEN
RESETTRK C3
INCREASE CYL COUNT BY 1 TO POINT TO NEXT CYL
D2
SET TRACK NUMBER IN REG 1
D3
STORE NEW CYLINDER NO. IN COUNTFLD AND SEEKFLD
E2
INCREASE TRACK NUMBER BY 1
E3
GET LOW LIMIT TRACK NUMBER AND INSERT IN REG 1
REPTRKNO F2
STORE NEW TRACK NUMBER IN COUNTFLD AND SEEKFLD
SW1 G2
SW1 ON *G5
TC B1
CRMOVE
H2
ALL FULL SIZE RECORDS WRITTEN
ANOTHREC J2
ADD SIZE OF RECORD WRITTEN FROM TOTRECLG TO REG 4 *J5
D4
ADD MAXIMUM RECORD SIZE TO REG 4
E4
PARTIAL RECORD TO BE WRITTEN
F4
TURN ON SW1
G4
CHANGE RCD SIZE IN CCWS AND COUNTFLD TO THE SIZE OF PARTIAL RCD
H4
CHANGE TOTRECLG TO THE LENGTH OF THE PARTIAL RECORD
*G5
SW1 IS TURNED ON AFTER LAST FULL SIZE RECORD IS WRITTEN AND THERE IS A PARTIAL RECORD TO BE WRITTEN.
*J5
AFTER FIRST RECORD IS WRITTEN, REG 4 CONTAINS END ADDRESS OF SUPVR. IT IS THEN INCREASED BY TOTAL RECORD LENGTH TO POINT TO THE FIRST BYTE OF THE NEXT RECORD.



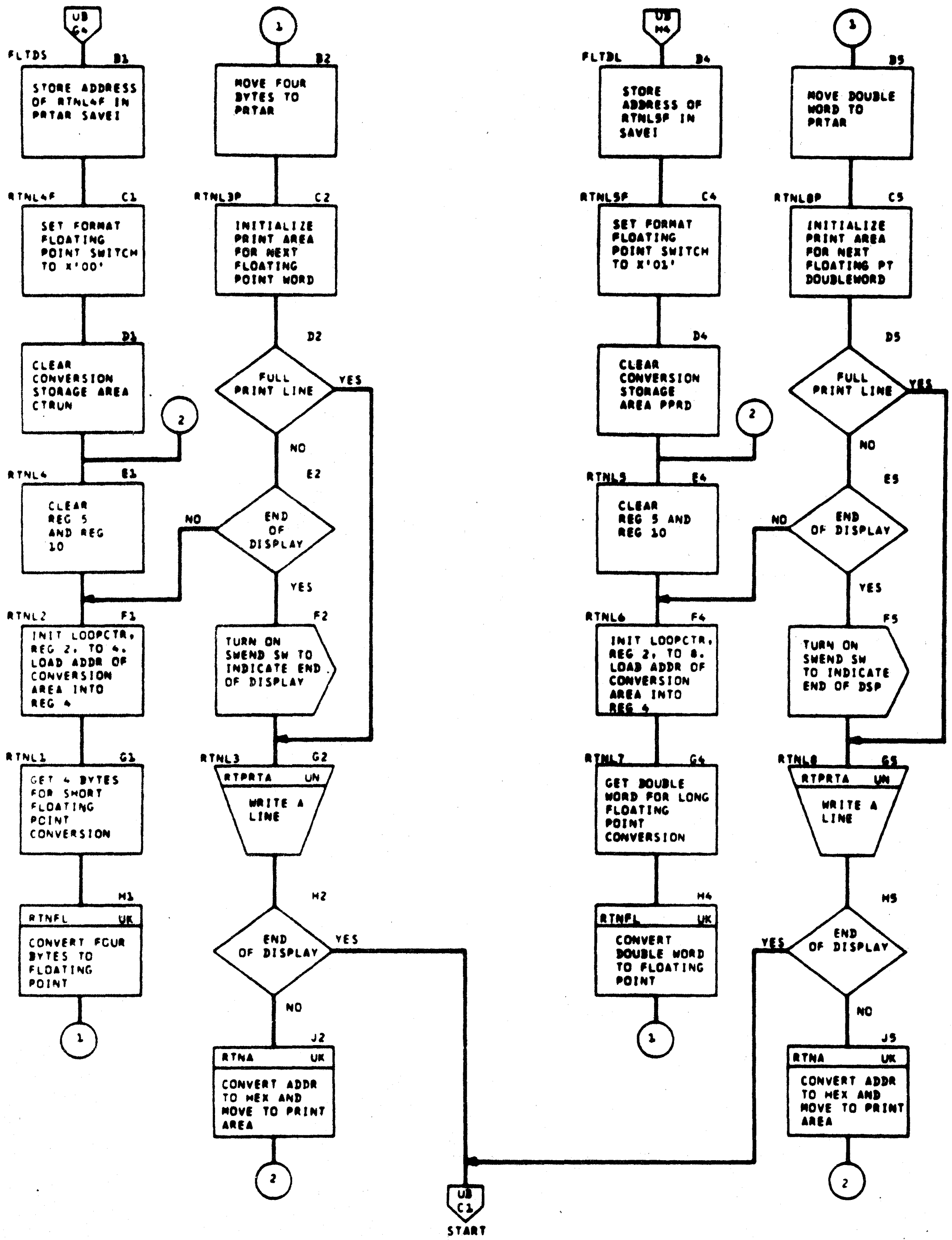
*E5
GET HIGHEST STORAGE ADDRESS USED FROM SYSTEM COMM REGION AND SUBTRACT LENGTH OF AUTOTEST CONTROL PROGRAM.

*F5
THIS SMALL EOJ DUMP RECORD IS 155 BYTES CONTAINING THE GENERAL REGISTERS, FLOATING POINT REGISTERS, USER PSW, CANCEL CODE, LENGTH OF LABEL AREA, AND USER COMMUNICATION AREA. IT IS WRITTEN AS THE 2ND RECORD ON TRACK 1 OF THE AUTOTEST WORK AREA ON SYSLNK


ATLEFF1
BEGIN B1
INITIALIZE BASE REGISTERS
STRTRT C1
OPEN SYSLST USING $$BOPEN
D1
INITIALIZE REG 13 TO POINT TO THE A/T COMM REGION
E1
SET LOWER TRK LIMITS, FROM AUTOTEST COMM REGION INTO LOLIM
F1
SET UPPER TRK LIMITS, FROM AUTOTEST COMM REGION INTO UPLIM
G1
LOAD ADDRESS OF SYSTEM COMM REGION INTO REG 1
H1
SET UP NO. OF LINES TO SPACE PER PAGE
PLS2 J1
MOVE TEST REQ CONTROL START ADDR. TO CC OF SEARCH FIELD
A2
TEST REQUEST OUTPUT PRESENT
YES
NO
B2
DISASTER CONTINUE
NO
UH D3
STOP
YES
RTNIPT C2
RTNG UH
READ INPUT RECORD
D2
END OF TEST REQ CTRL RCD
YES
UB B1
ISH2
NO
E2
POINT REG 3 TO FIRST BYTE OF LOG RECORD
F2
CLEAR REGISTER 6
G2
MOVE NO. OF LOG RCDS IN PHYSICAL RCD AND LAST RCD INDR INTO REG 6
H2
SHIFT, REG 6, RIGHT ONE BIT TO REMOVE LAST RECORD INDICATOR
BACKA J2
CLEAR WRKTBL
B3
MOVE TEST REQUEST NO. AND CODE TO WRKTBL
C3
POINT REG 3 TO THE DISPLAY LIMIT FIELD OF THE INPUT LOGICAL RCD
D3
DISPLAY
YES
NO
E3
PANEL WITH ON
NO
YES
F3
MOVE ON COND BEGIN NO. AND EVERY NO. TO WRKTBL+8
G3
UPDATE REG 3 TO POINT TO NEXT LOGICAL RECORD
TRRT H3
MULTIPLY T/R NUMBER BY 11 TO GET ITS ADDRESS IN MAIN STORAGE
J3
MOVE TEST REQ RECORD TO ITS PROPER LOCATION IN MAIN STORAGE
K3
END OF PHYSICAL RECORD
NO
YES
TRRT1 G4
SET ASSUMED VALUES FOR ON COND BEGIN NO. AND EVERY NO. IN WRKTBL
DISPIN C5
MOVE DISPLAY LIMITS OF THE DISPLAY TO WRKTBL+2
D5
UPDATE REG 3 TO POINT TO ON COND BEGIN NO. AND EVERY NUMBER
E5
PANEL WITH ON
NO
YES
F5
MOVE ON COND BEGIN, AND EVERY NOS. TO WRKTBL+8
G5
UPDATE REG 3 TO POINT TO NEXT LOGICAL RECORD



UA B2
ISM2 B1
INITIALIZE DISK ADDR TO T/R OUTPUT START AND SET BYTE CNT IN CCW
* UC-H5 UC-H2 UD-K3 UE-K4 UF-K5 UG-J4
START C1
CLEAR REG 1 AND REG 2
D1
INITIALIZE SWITCHES IN TABPR
E1
CLEAR 6 BYTES OF DITTO AREA OF TABPR
F1
RESET SWITCHES SWDT2, SWDTC AND SWDT
G1
CLEAR INPUT/OUTPUT AREA
CTRTN H1
RTNG UH
GET A RECORD
TSTRTN B2
TEST REQUEST OUTPUT RECORD
NO
YES
RTNF C2
INITIALIZE REG 7 TO 11 FOR MULTPLIER
D2
CLEAR SYSLST INPUT/OUTPUT AREA
E2
MULTIPLY T/R NUMBER BY 11 TO FIND ITS ADDRESS IN CORE
F2
MOVE TEST REQUEST RECORD TO WRKTBL
G2
STORE WRKTBL ADDRESS IN TABPA
RTNH H2
LOAD TEST REQUEST NUMBER INTO REG 4
J2
RTNHB UJ
CONVERT T/R NUMBER TO DECIMAL
B3
MOVE ADDR OF SIDE ONE OF DISPLAY LINE TO PRTAR
C3
ADD BEGIN NO. TO EVERY NUMBER AND STORE IN BEGIN NUMBER
D3
RTNHB UJ
CONVERT ON BEGIN NO. TO DECIMAL
E3
MOVE HEADER INFORMATION TO PRINT AREA
F3
SET UP TO SPACE TWO LINES BEFORE HEADER
G3
RTPRT UL
WRITE HEADER
H3
SET UP TO SPACE TWO LINES AFTER HEADER
J3
LOAD TABLE ADDRESS INTO REG 4
B4
MOVE ON CONDITION BACK INTO TABLE
C4
DISPLAY ONLY
NO
UE B3
RTPNL
UE K3
YES
RTNFD D4
MOVE FIRST BYTE OF LOG RCD TO TST, CLEAR ALL BITS BUT DSP PARAMETER
E4
RTNS UK
CALC START, END AND NO. OF BYTES IN DISPLAY
F4
HEX AND/OR CHAR DSP
YES
NO
G4
FLOATING POINT SHORT DSP
YES
UC B1
FLTDS
NO
H4
FLOATING POINT LONG DSP
YES
UC B4
FLTDL
NO
J4
FIXED POINT DISPLAY
YES
UD B3
FIXPH
NO
K4
TURN ON SWHX TO INDICATE MNEMONIC DISPLAY
UF A1
RTND


UB C4
FLTDS B1
STORE ADDRESS OF RTNL4F IN PRTAR SAVE1
RTNL4F C1
SET FORMAT FLOATING POINT SWITCH TO X'00'
D1
CLEAR CONVERSION STORAGE AREA CTRUN
2
RTNL4 E1
CLEAR REG 5 AND REG 10
RTNL2 F1
INIT LOOPCTR, REG 2, TO 4, LOAD ADDR OF CONVERSION AREA INTO REG 4
RTNL1 G1
GET 4 BYTES FOR SHORT FLOATING POINT CONVERSION
H1
RTNFL UK
CONVERT FOUR BYTES TO FLOATING POINT
1
1
B2
MOVE FOUR BYTES TO PRTAR
RTNL3P C2
INITIALIZE PRINT AREA FOR NEXT FLOATING POINT WORD
D2
FULL PRINT LINE
YES
NO
E2
END OF DISPLAY
NO
YES
F2
TURN ON SWEND SW TO INDICATE END OF DISPLAY
RTNL3 G2
RTPRTA UN
WRITE A LINE
H2
END OF DISPLAY
YES
NO
J2
RTNA UK
CONVERT ADDR TO HEX AND MOVE TO PRINT AREA
2
UB C1
START
UB H4
FLTDL B4
STORE ADDRESS OF RTNL5F IN SAVE1
RTNL5F C4
SET FORMAT FLOATING POINT SWITCH TO X'01'
D4
CLEAR CONVERSION STORAGE AREA PPRD
2
RTNL5 E4
CLEAR REG 5 AND REG 10
RTNL6 F4
INIT LOOPCTR, REG 2, TO 8, LOAD ADDR OF CONVERSION AREA INTO REG 4
RTNL7 G4
GET DOUBLE WORD FOR LONG FLOATING POINT CONVERSION
H4
RTNFL UK
CONVERT DOUBLE WORD TO FLOATING POINT
1
1
B5
MOVE DOUBLE WORD TO PRTAR
RTNL8P C5
INITIALIZE PRINT AREA FOR NEXT FLOATING PT DOUBLEWORD
D5
FULL PRINT LINE
YES
NO
E5
END OF DISPLAY
NO
YES
F5
TURN ON SWEND SW TO INDICATE END OF DSP
RTNL8 G5
RTPRTA UN
WRITE A LINE
H5
END OF DISPLAY
YES
NO
J5
RTNA UK
CONVERT ADDR TO HEX AND MOVE TO PRINT AREA
2

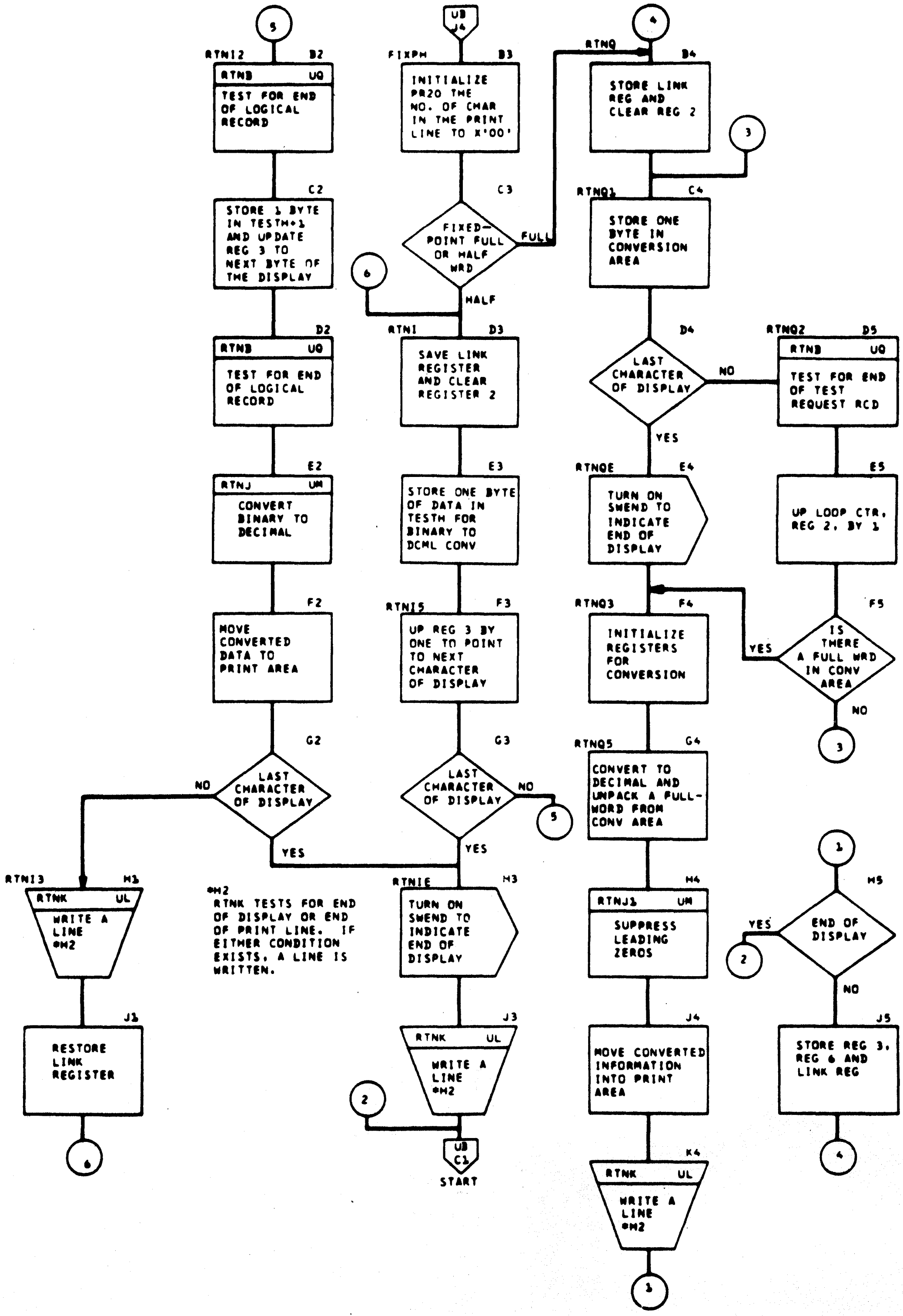
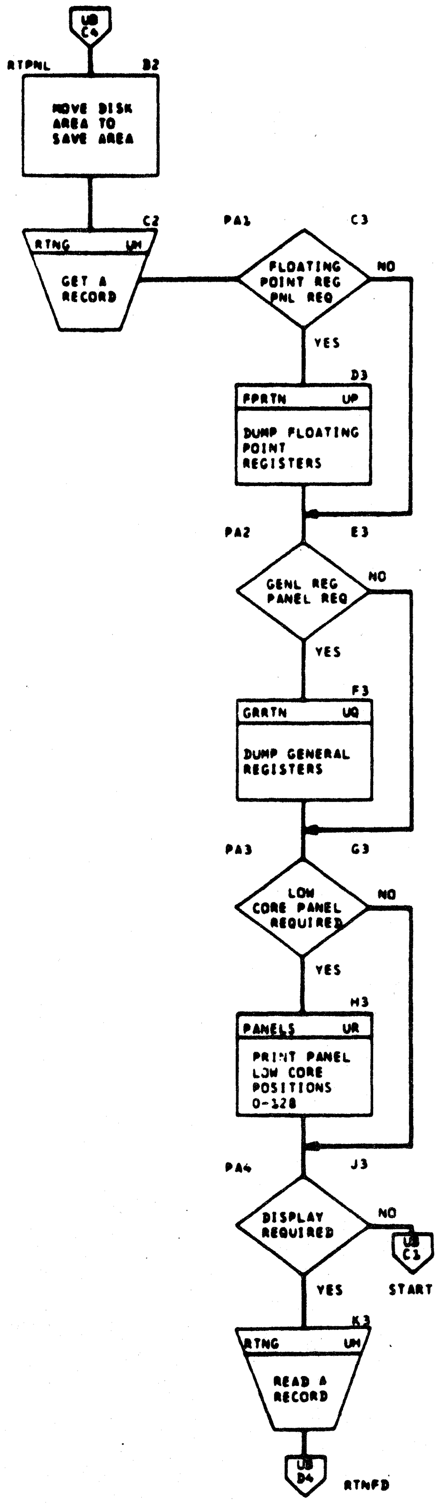
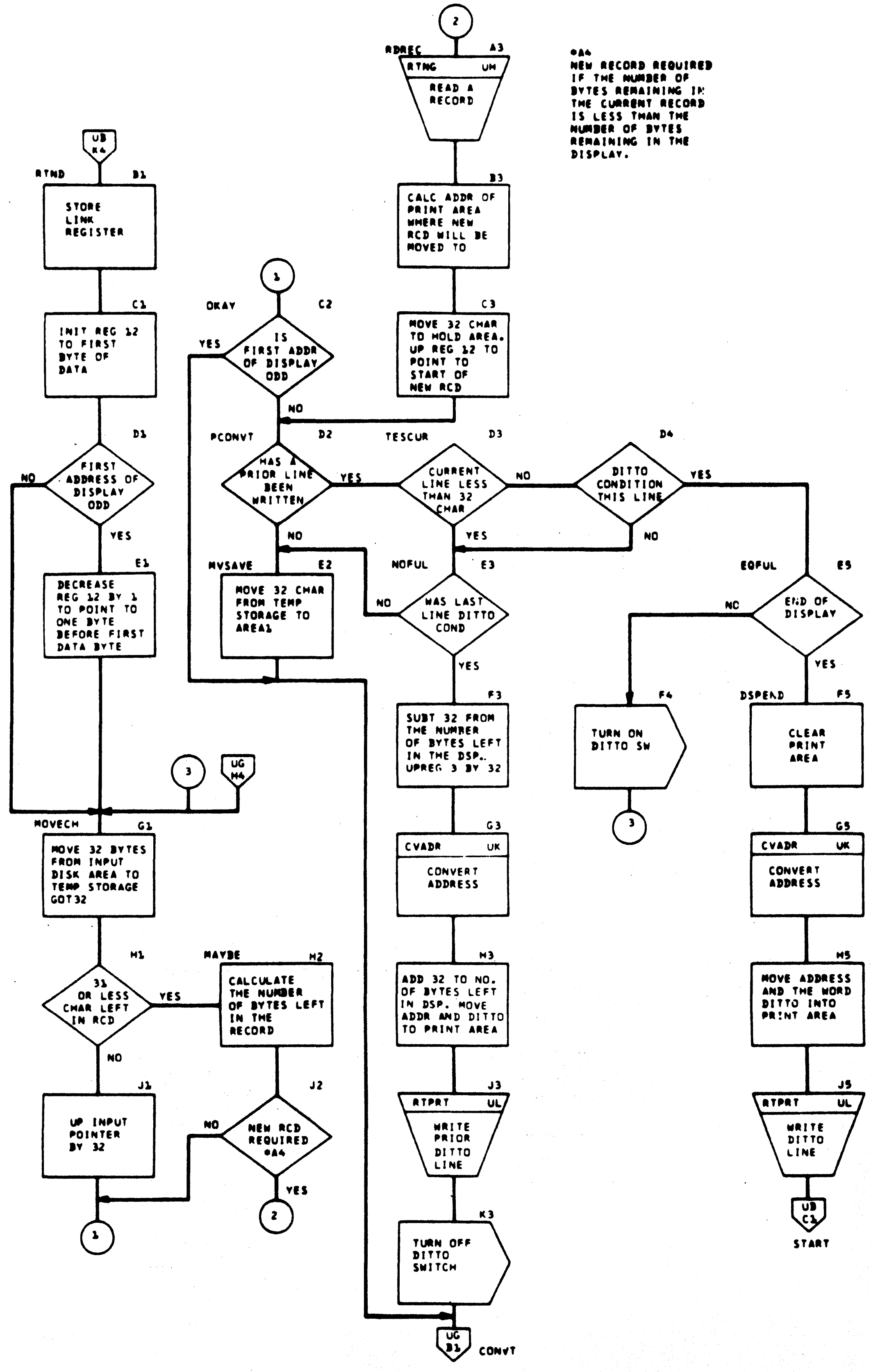

RTN12 B2
RTNB UQ
TEST FOR END OF LOGICAL RECORD
C2
STORE 1 BYTE IN TESTH+1 AND UPDATE REG 3 TO NEXT BYTE OF THE DISPLAY
D2
RTNB UQ
TEST FOR END OF LOGICAL RECORD
E2
RTNJ UM
CONVERT BINARY TO DECIMAL
F2
MOVE CONVERTED DATA TO PRINT AREA
G2
LAST CHARACTER OF DISPLAY
NO
YES
RTN13 H1
RTNK UL
WRITE A LINE *H2
J1
RESTORE LINK REGISTER
*H2
RTNK TESTS FOR END OF DISPLAY OR END OF PRINT LINE. IF EITHER CONDITION EXISTS, A LINE IS WRITTEN.
UB J4
FIXPH B3
INITIALIZE PR20 THE NO. OF CHAR IN THE PRINT LINE TO X'00'
C3
FIXED-POINT FULL OR HALF WRD
FULL
HALF
RTN1 D3
SAVE LINK REGISTER AND CLEAR REGISTER 2
E3
STORE ONE BYTE OF DATA IN TESTH FOR BINARY TO DCML CONV
RTN15 F3
UP REG 3 BY ONE TO POINT TO NEXT CHARACTER OF DISPLAY
G3
LAST CHARACTER OF DISPLAY
NO
YES
RTNIE H3
TURN ON SWEND TO INDICATE END OF DISPLAY
J3
RTNK UL
WRITE A LINE *H2
UB C1
START
RTNQ B4
STORE LINK REG AND CLEAR REG 2
RTNQ1 C4
STORE ONE BYTE IN CONVERSION AREA
D4
LAST CHARACTER OF DISPLAY
NO
YES
RTNQE E4
TURN ON SWEND TO INDICATE END OF DISPLAY
RTNQ3 F4
INITIALIZE REGISTERS FOR CONVERSION
RTNQ5 G4
CONVERT TO DECIMAL AND UNPACK A FULL-WORD FROM CONV AREA
RTNJ1 UM H4
SUPPRESS LEADING ZEROS
J4
MOVE CONVERTED INFORMATION INTO PRINT AREA
K4
RTNK UL
WRITE A LINE *H2
RTNQ2 D5
RTNB UQ
TEST FOR END OF TEST REQUEST RCD
E5
UP LOOP CTR, REG 2, BY 1
F5
IS THERE A FULL WRD IN CONV AREA
YES
NO
H5
END OF DISPLAY
YES
NO
J5
STORE REG 3, REG 6 AND LINK REG

Chart UE. Test Request Output (ATLEFF1) (Refer to Chart 13) (Part 5 of 17)





RDREC A3
RTNG UM
READ A RECORD
*A4
NEW RECORD REQUIRED IF THE NUMBER OF BYTES REMAINING IN THE CURRENT RECORD IS LESS THAN THE NUMBER OF BYTES REMAINING IN THE DISPLAY.
UB K4
RTND B1
STORE LINK REGISTER
B3
CALC ADDR OF PRINT AREA WHERE NEW RCD WILL BE MOVED TO
C1
INIT REG 12 TO FIRST BYTE OF DATA
OKAY C2
IS FIRST ADDR OF DISPLAY ODD
C3
MOVE 32 CHAR TO HOLD AREA. UP REG 12 TO POINT TO START OF NEW RCD
D1
FIRST ADDRESS OF DISPLAY ODD
PCONVT D2
HAS A PRIOR LINE BEEN WRITTEN
TESCUR D3
CURRENT LINE LESS THAN 32 CHAR
D4
DITTO CONDITION THIS LINE
E1
DECREASE REG 12 BY 1 TO POINT TO ONE BYTE BEFORE FIRST DATA BYTE
MVSAVE E2
MOVE 32 CHAR FROM TEMP STORAGE TO AREA1
NOFUL E3
WAS LAST LINE DITTO COND
EOFUL E5
END OF DISPLAY
F3
SUBT 32 FROM THE NUMBER OF BYTES LEFT IN THE DSP. UPREG 3 BY 32
F4
TURN ON DITTO SW
DSPEND F5
CLEAR PRINT AREA
UG H4
MOVECH G1
MOVE 32 BYTES FROM INPUT DISK AREA TO TEMP STORAGE GOT32
G3
CVADR UK
CONVERT ADDRESS
G5
CVADR UK
CONVERT ADDRESS
H1
31 OR LESS CHAR LEFT IN RCD
MAYBE H2
CALCULATE THE NUMBER OF BYTES LEFT IN THE RECORD
H3
ADD 32 TO NO. OF BYTES LEFT IN DSP. MOVE ADDR AND DITTO TO PRINT AREA
H5
MOVE ADDRESS AND THE WORD DITTO INTO PRINT AREA
J1
UP INPUT POINTER BY 32
J2
NEW RCD REQUIRED *A4
J3
RTPRT UL
WRITE PRIOR DITTO LINE
J5
RTPRT UL
WRITE DITTO LINE
UB C1
START
K3
TURN OFF DITTO SWITCH
UG B1
CONVT
YES
NO

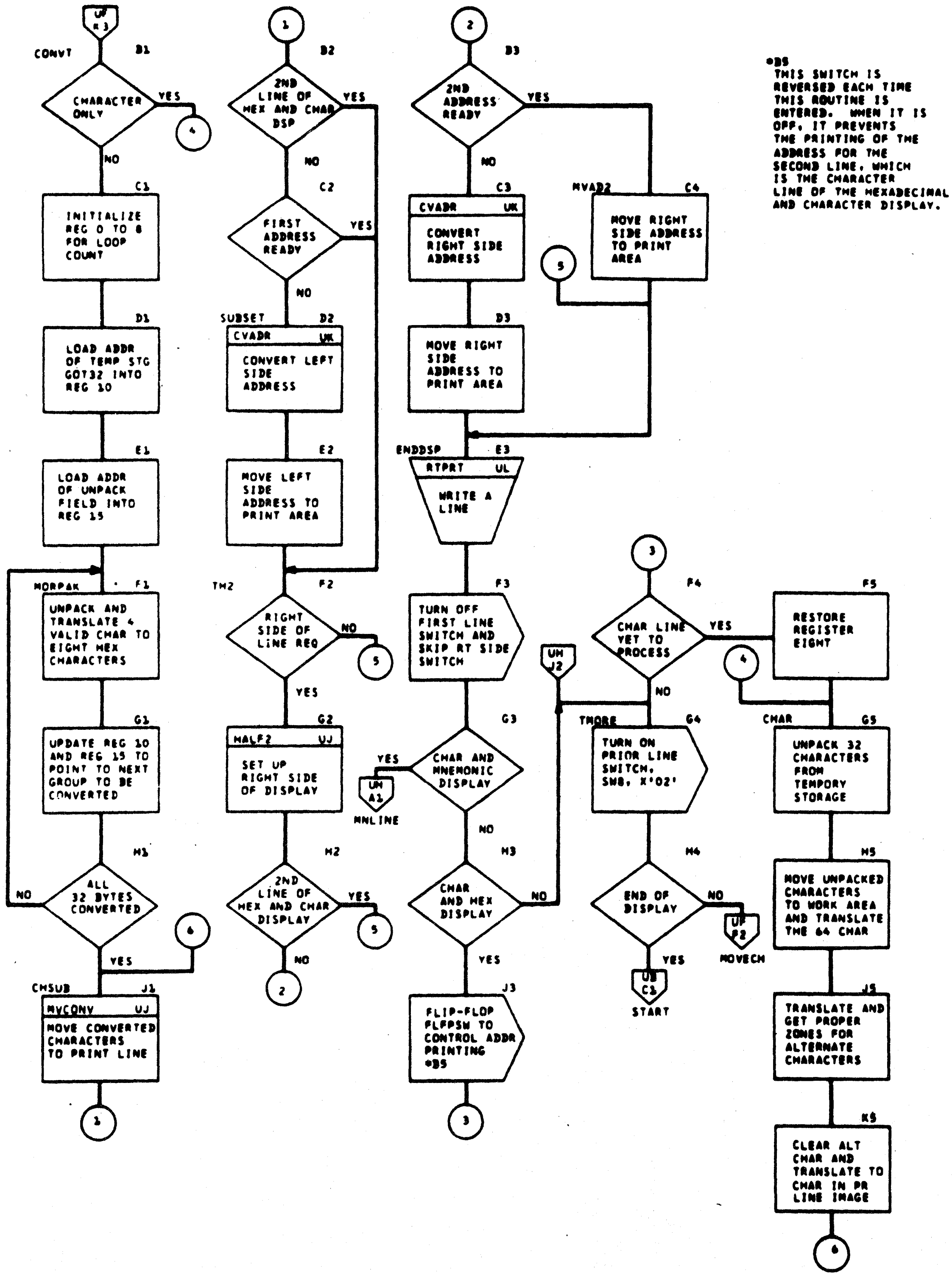

CONVT
CHARACTER ONLY
INITIALIZE REG 0 TO 8 FOR LOOP COUNT
LOAD ADDR OF TEMP STG GOT32 INTO REG 10
LOAD ADDR OF UNPACK FIELD INTO REG 15
MORPAK
UNPACK AND TRANSLATE 4 VALID CHAR TO EIGHT HEX CHARACTERS
UPDATE REG 10 AND REG 15 TO POINT TO NEXT GROUP TO BE CONVERTED
ALL 32 BYTES CONVERTED
CHSUB
MVCONV UJ
MOVE CONVERTED CHARACTERS TO PRINT LINE
2ND LINE OF HEX AND CHAR DSP
FIRST ADDRESS READY
SUBSET
CVADR UK
CONVERT LEFT SIDE ADDRESS
MOVE LEFT SIDE ADDRESS TO PRINT AREA
TH2
RIGHT SIDE OF LINE REQ
HALF2 UJ
SET UP RIGHT SIDE OF DISPLAY
2ND LINE OF HEX AND CHAR DISPLAY
2ND ADDRESS READY
CVADR UK
CONVERT RIGHT SIDE ADDRESS
MVAD2
MOVE RIGHT SIDE ADDRESS TO PRINT AREA
MOVE RIGHT SIDE ADDRESS TO PRINT AREA
ENDDSP
RTPRT UL
WRITE A LINE
TURN OFF FIRST LINE SWITCH AND SKIP RT SIDE SWITCH
CHAR AND MNEMONIC DISPLAY
MNLINE
CHAR AND HEX DISPLAY
FLIP-FLOP FLFPSW TO CONTROL ADDR PRINTING *B5
CHAR LINE YET TO PROCESS
THORE
TURN ON PRIOR LINE SWITCH, SW8, X'02'
END OF DISPLAY
START
MOVECH
RESTORE REGISTER EIGHT
CHAR
UNPACK 32 CHARACTERS FROM TEMPORY STORAGE
MOVE UNPACKED CHARACTERS TO WORK AREA AND TRANSLATE THE 64 CHAR
TRANSLATE AND GET PROPER ZONES FOR ALTERNATE CHARACTERS
CLEAR ALT CHAR AND TRANSLATE TO CHAR IN PR LINE IMAGE
*B5
THIS SWITCH IS REVERSED EACH TIME THIS ROUTINE IS ENTERED. WHEN IT IS OFF, IT PREVENTS THE PRINTING OF THE ADDRESS FOR THE SECOND LINE, WHICH IS THE CHARACTER LINE OF THE HEXADECIMAL AND CHARACTER DISPLAY.

Chart UH. Test Request Output (ATLEFF1) (Refer to Chart 13) (Part 8 of 17)

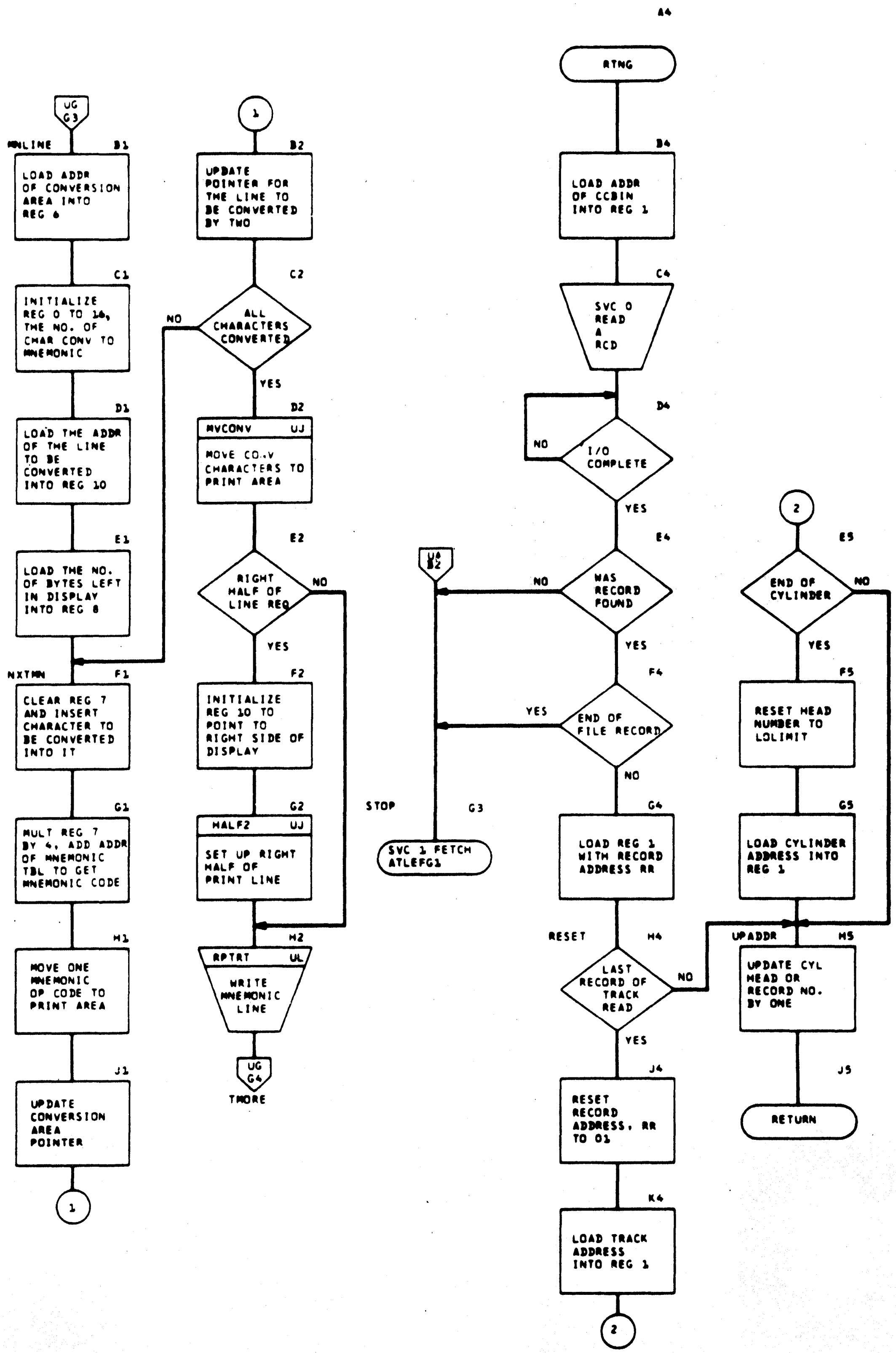


Chart UJ. Test Request Output (ATLEFF1) (Refer to Chart 13) (Part 9 of 17)

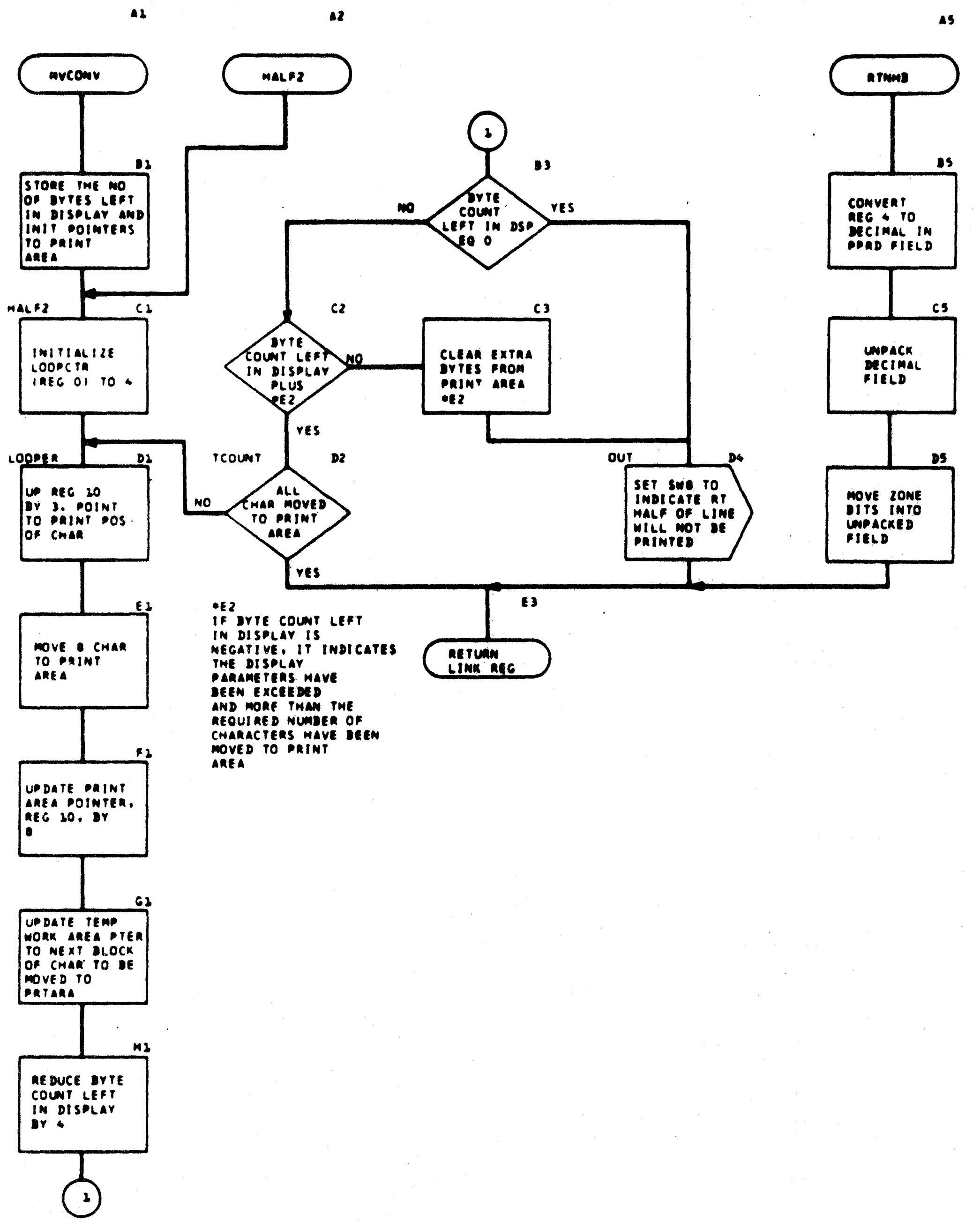


Chart UK. Test Request Output (ATLEFF1) (Refer to Chart 13) (Part 10 of 17)

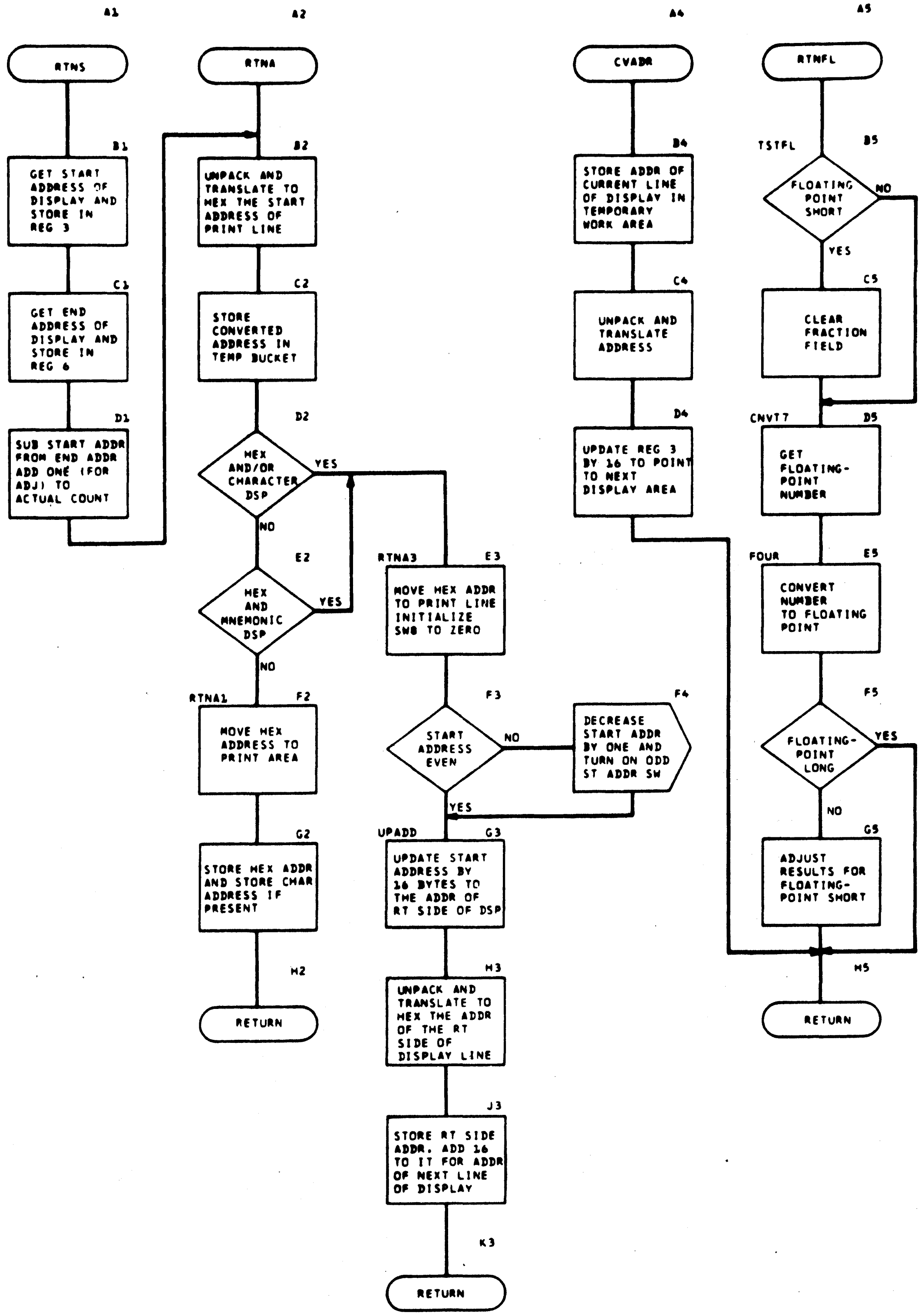


Chart UL. Test Request Output (ATLEFF1) (Refer to Chart 13) (Part 11 of 17)

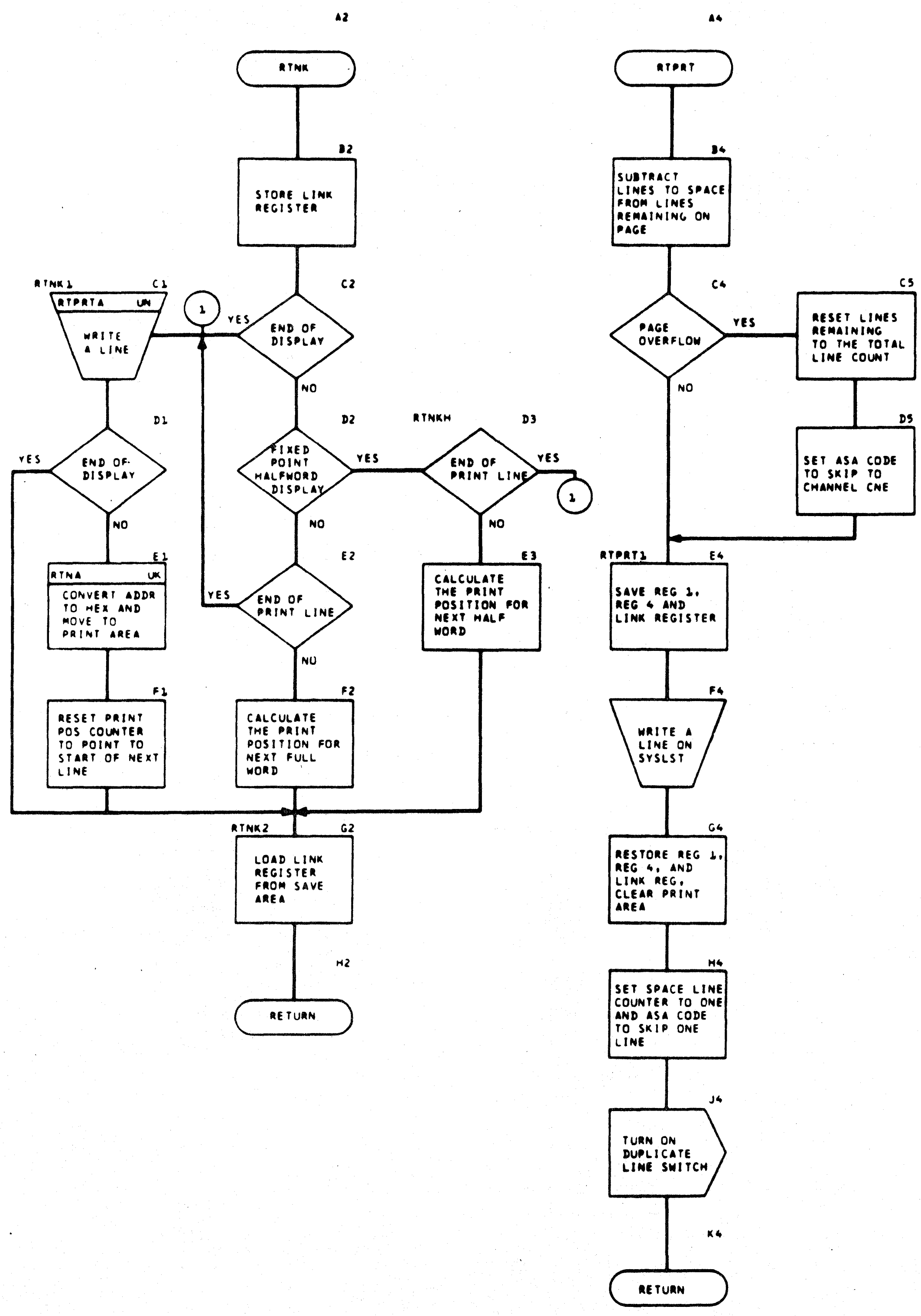
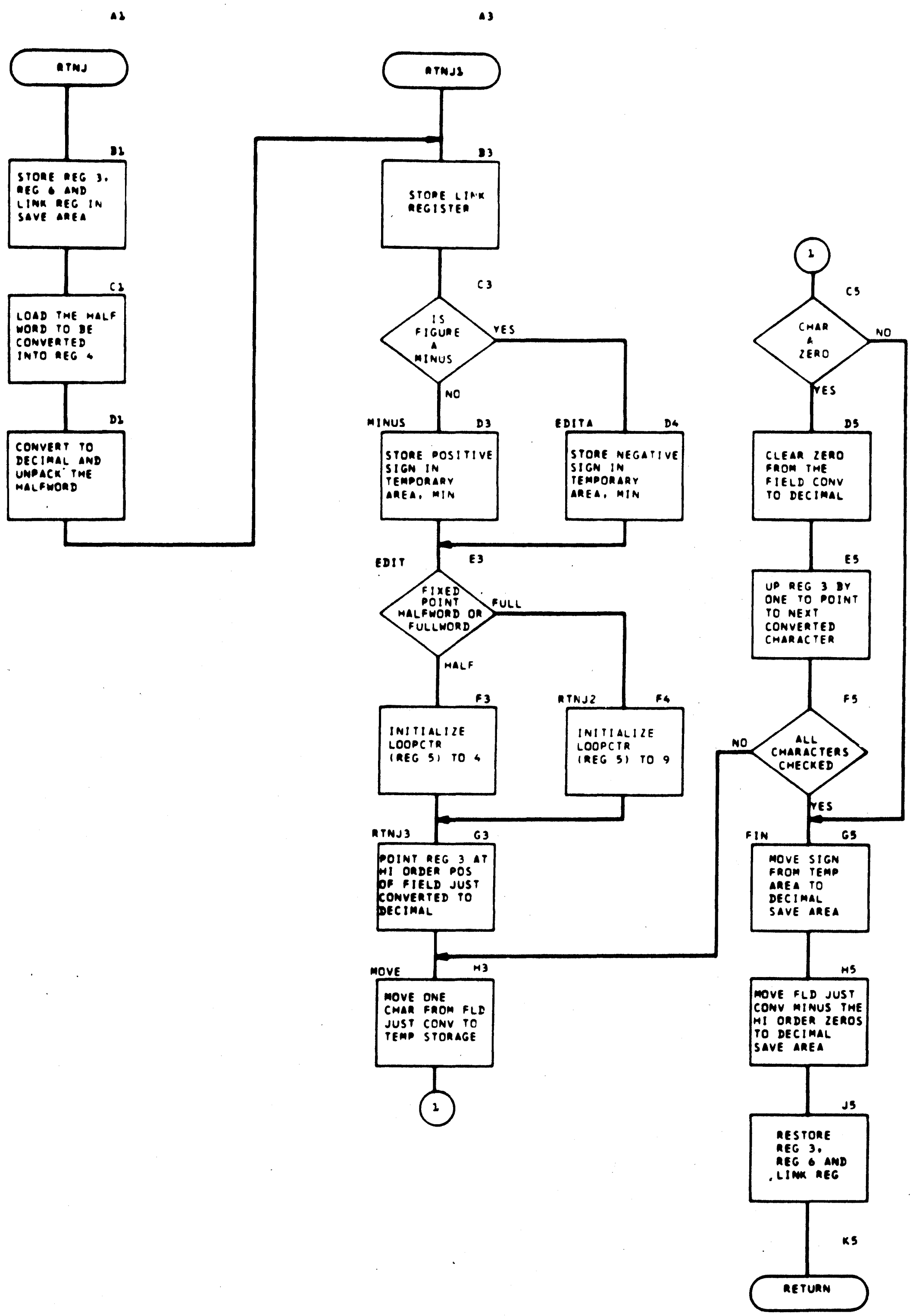


Chart UM. Test Request Output (ATLEFF1) (Refer to Chart 13) (Part 12 of 17)



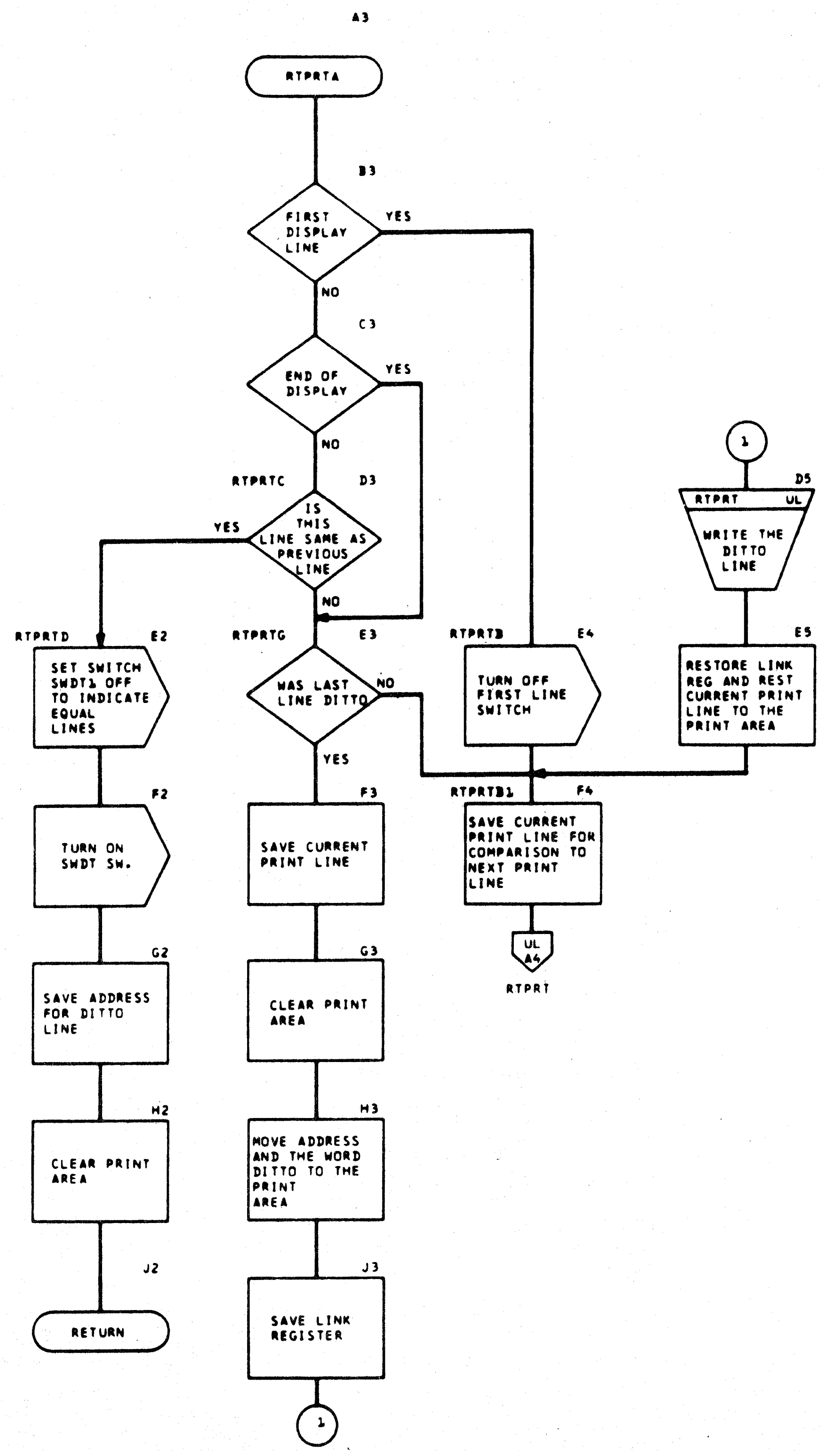
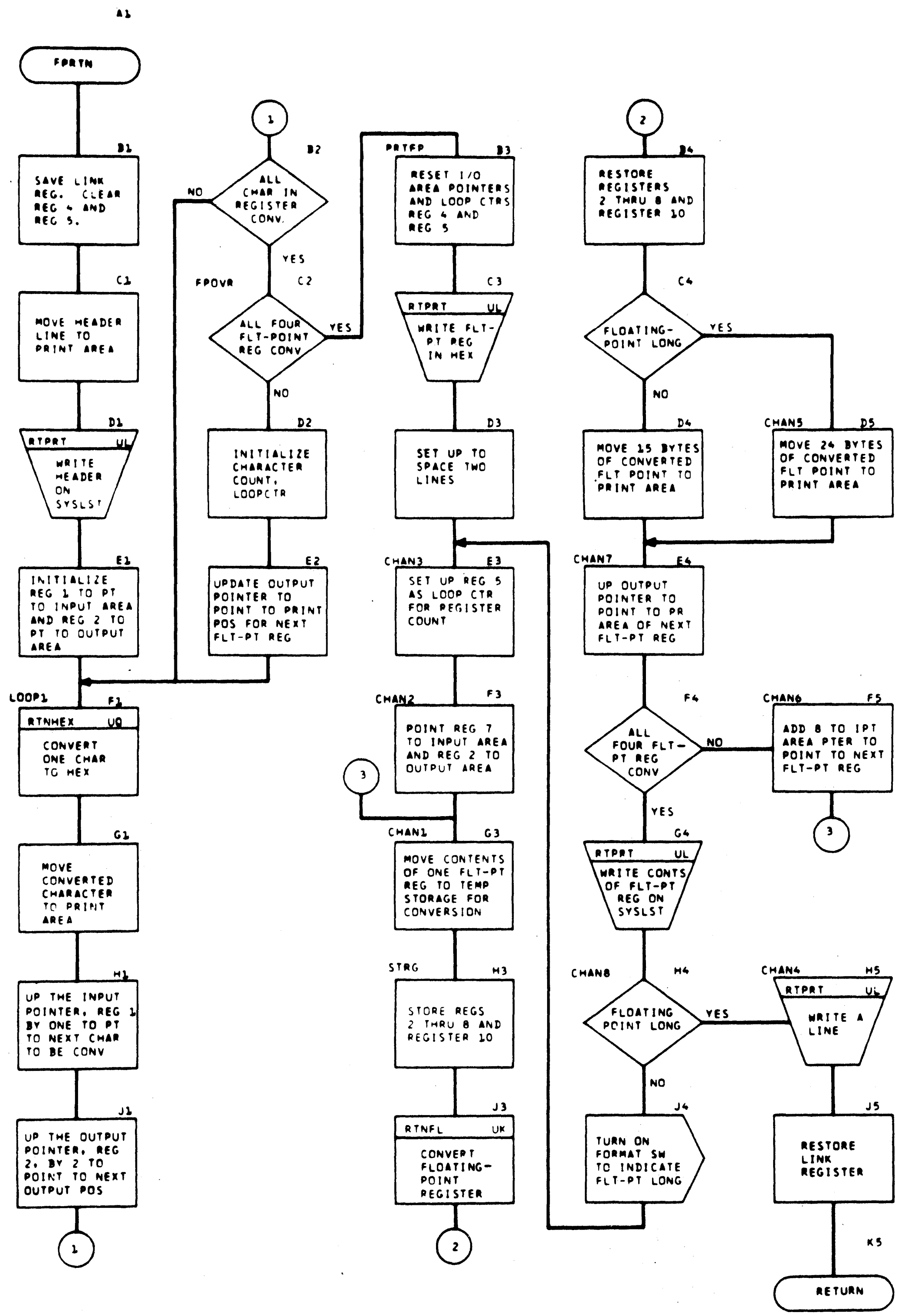

A3
RTPRTA
B3
FIRST DISPLAY LINE
YES
NO
C3
END OF DISPLAY
YES
NO
RTPRTC
D3
IS THIS LINE SAME AS PREVIOUS LINE
YES
NO
1
D5
RTPRT
UL
WRITE THE DITTO LINE
RTPRTD
E2
SET SWITCH SWDT1 OFF TO INDICATE EQUAL LINES
RTPRTG
E3
WAS LAST LINE DITTO
NO
YES
RTPRTB
E4
TURN OFF FIRST LINE SWITCH
E5
RESTORE LINK REG AND REST CURRENT PRINT LINE TO THE PRINT AREA
F2
TURN ON SWDT SW.
F3
SAVE CURRENT PRINT LINE
RTPRTB1
F4
SAVE CURRENT PRINT LINE FOR COMPARISON TO NEXT PRINT LINE
UL
A4
RTPRT
G2
SAVE ADDRESS FOR DITTO LINE
G3
CLEAR PRINT AREA
H2
CLEAR PRINT AREA
H3
MOVE ADDRESS AND THE WORD DITTO TO THE PRINT AREA
J2
RETURN
J3
SAVE LINK REGISTER
1

Chart UP. Test Request Output (ATLEFF1) (Refer to Chart 13) (Part 14 of 17)



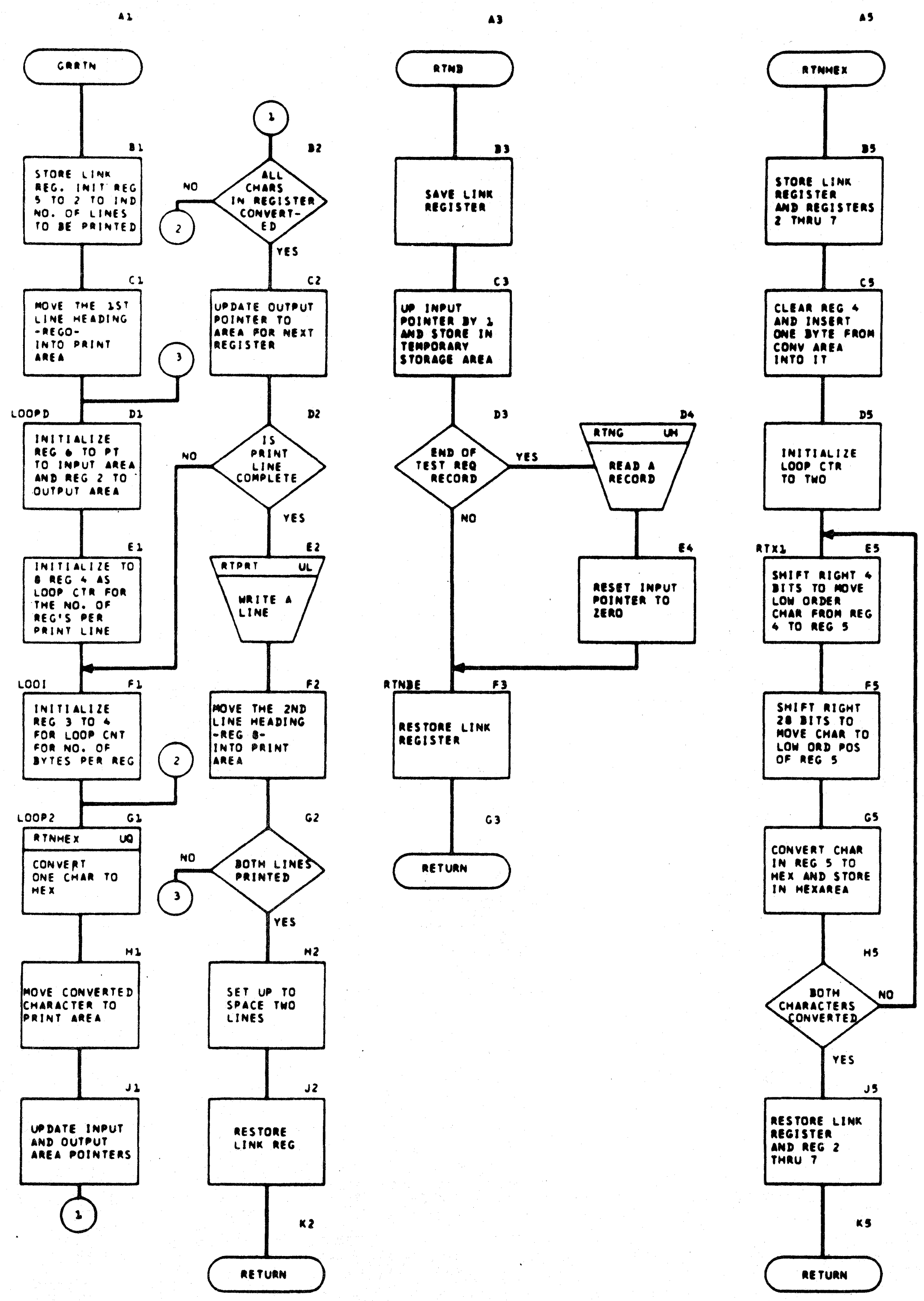
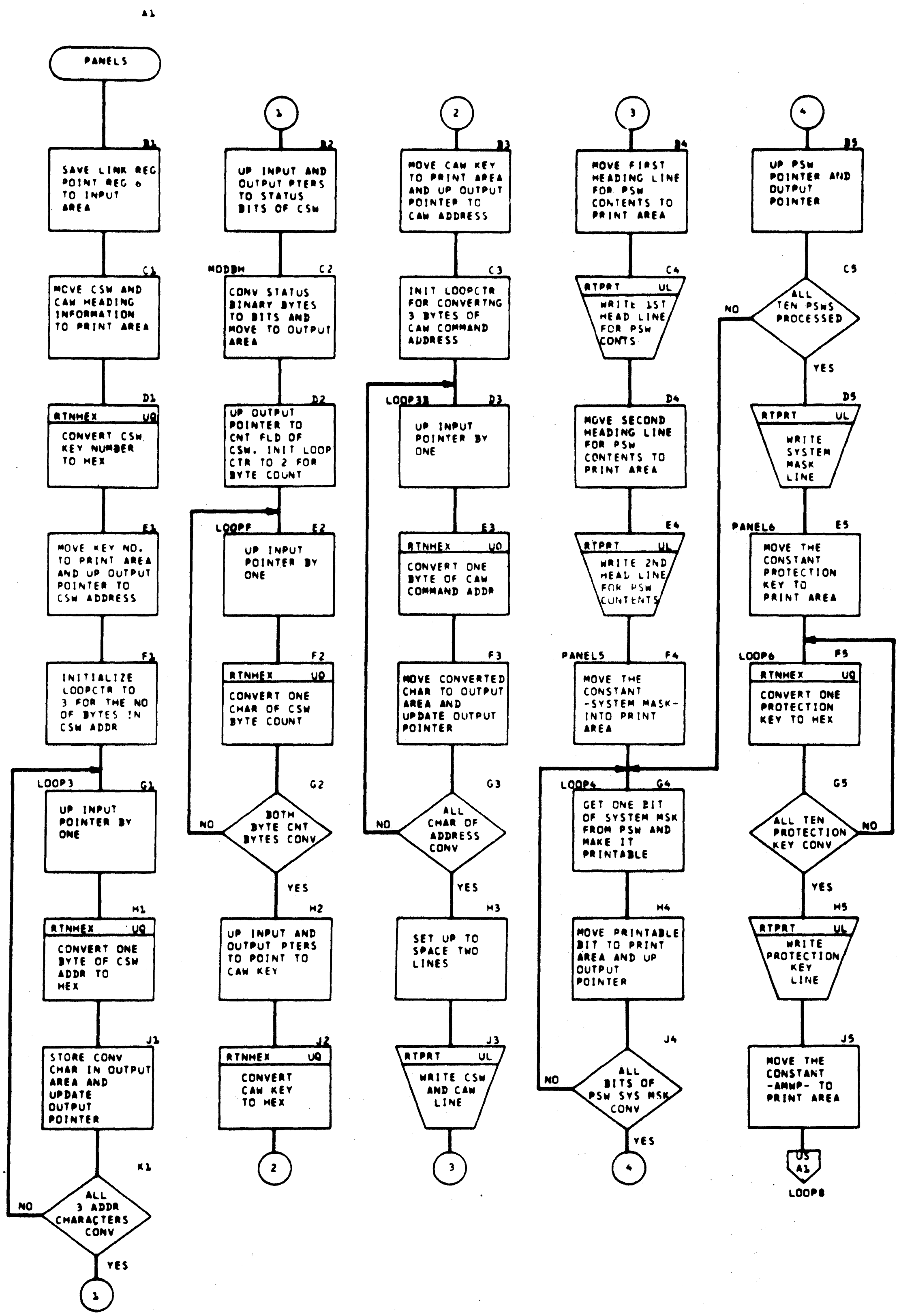

GRRTN
STORE LINK REG. INIT REG 5 TO 2 TO IND NO. OF LINES TO BE PRINTED
MOVE THE 1ST LINE HEADING -REGO- INTO PRINT AREA
LOOPD
INITIALIZE REG 6 TO PT TO INPUT AREA AND REG 2 TO OUTPUT AREA
INITIALIZE TO 8 REG 4 AS LOOP CTR FOR THE NO. OF REG'S PER PRINT LINE
LOOI
INITIALIZE REG 3 TO 4 FOR LOOP CNT FOR NO. OF BYTES PER REG
LOOP2
RTNHEX UQ
CONVERT ONE CHAR TO HEX
MOVE CONVERTED CHARACTER TO PRINT AREA
UPDATE INPUT AND OUTPUT AREA POINTERS
ALL CHARS IN REGISTER CONVERTED
NO
YES
UPDATE OUTPUT POINTER TO AREA FOR NEXT REGISTER
IS PRINT LINE COMPLETE
RTPRT UL
WRITE A LINE
MOVE THE 2ND LINE HEADING -REG 8- INTO PRINT AREA
BOTH LINES PRINTED
SET UP TO SPACE TWO LINES
RESTORE LINK REG
RETURN
RTNB
SAVE LINK REGISTER
UP INPUT POINTER BY 1 AND STORE IN TEMPORARY STORAGE AREA
END OF TEST REQ RECORD
RTNG UH
READ A RECORD
RESET INPUT POINTER TO ZERO
RTNBE
RESTORE LINK REGISTER
RETURN
RTNHEX
STORE LINK REGISTER AND REGISTERS 2 THRU 7
CLEAR REG 4 AND INSERT ONE BYTE FROM CONV AREA INTO IT
INITIALIZE LOOP CTR TO TWO
RTX1
SHIFT RIGHT 4 BITS TO MOVE LOW ORDER CHAR FROM REG 4 TO REG 5
SHIFT RIGHT 28 BITS TO MOVE CHAR TO LOW ORD POS OF REG 5
CONVERT CHAR IN REG 5 TO HEX AND STORE IN HEXAREA
BOTH CHARACTERS CONVERTED
RESTORE LINK REGISTER AND REG 2 THRU 7
RETURN

Chart UR. Test Request Output (ATLEFF1) (Refer to Chart 13) (Part 16 of 17)



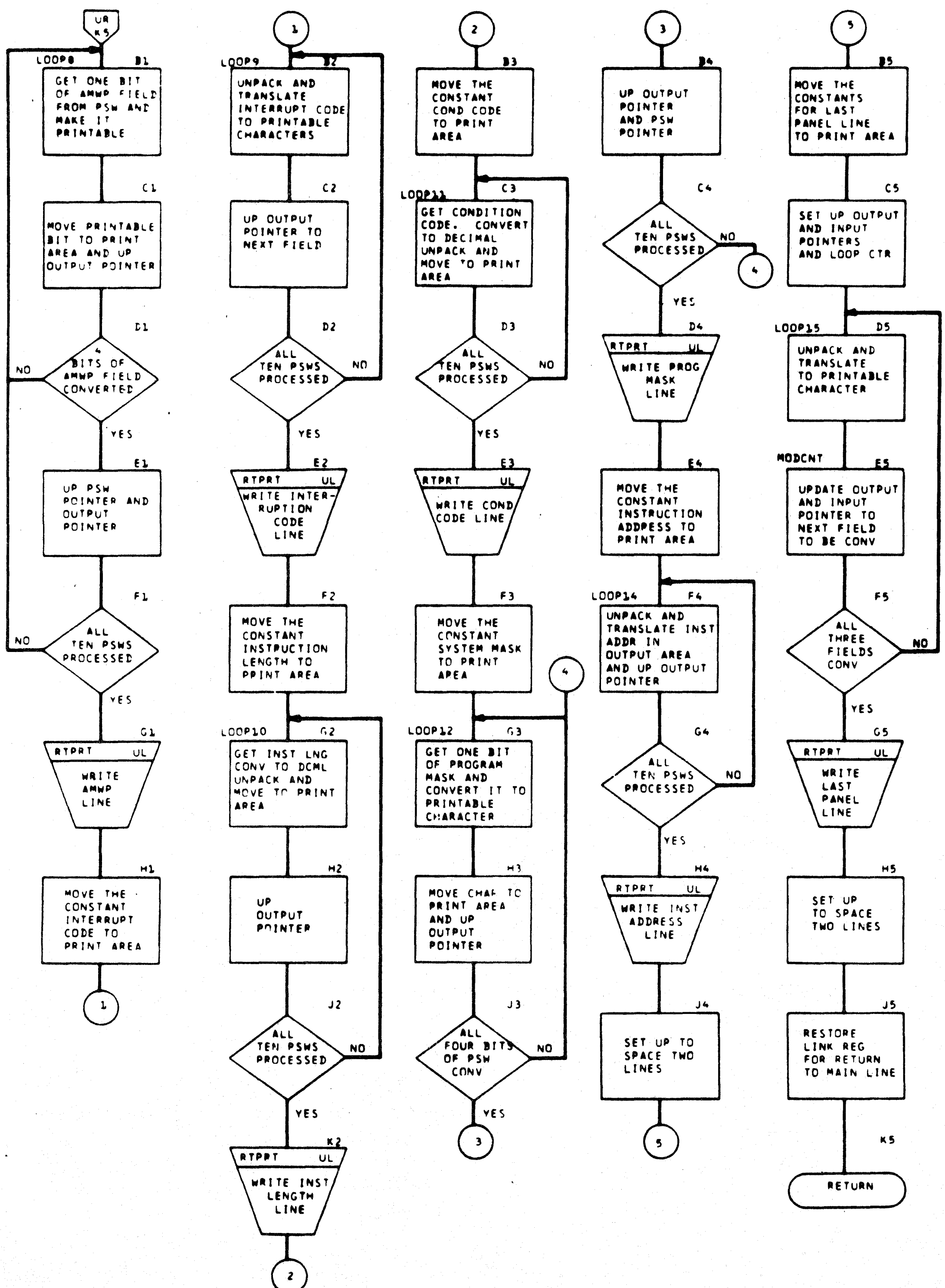
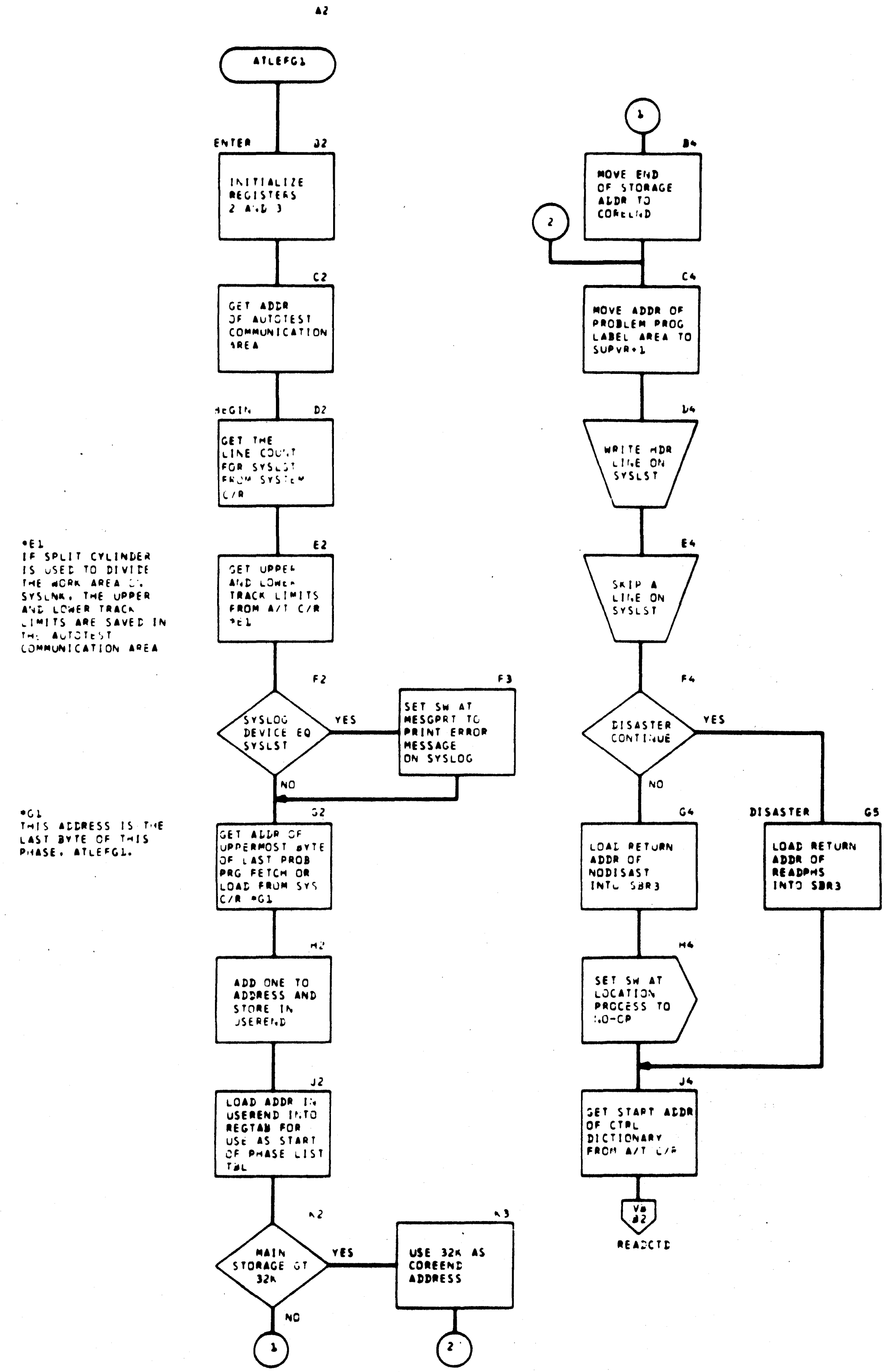
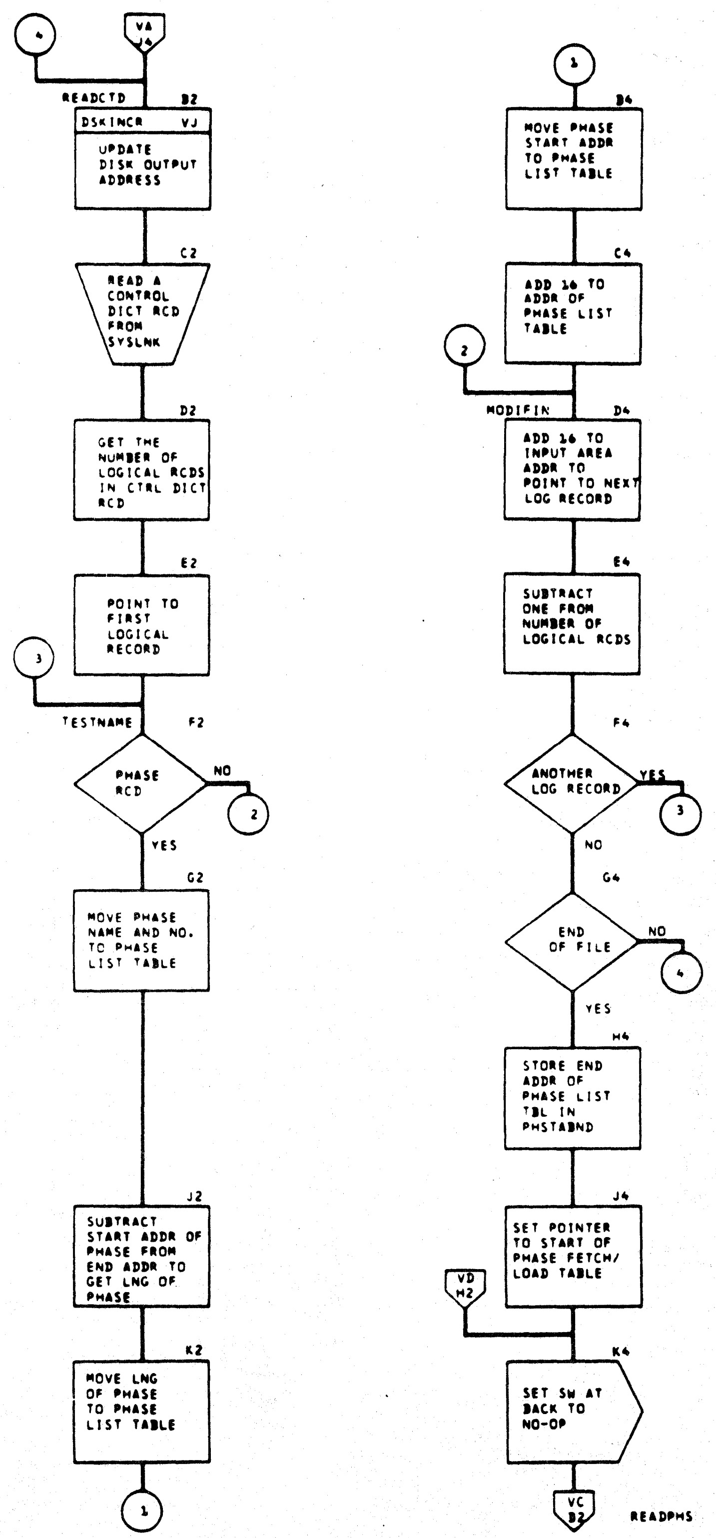
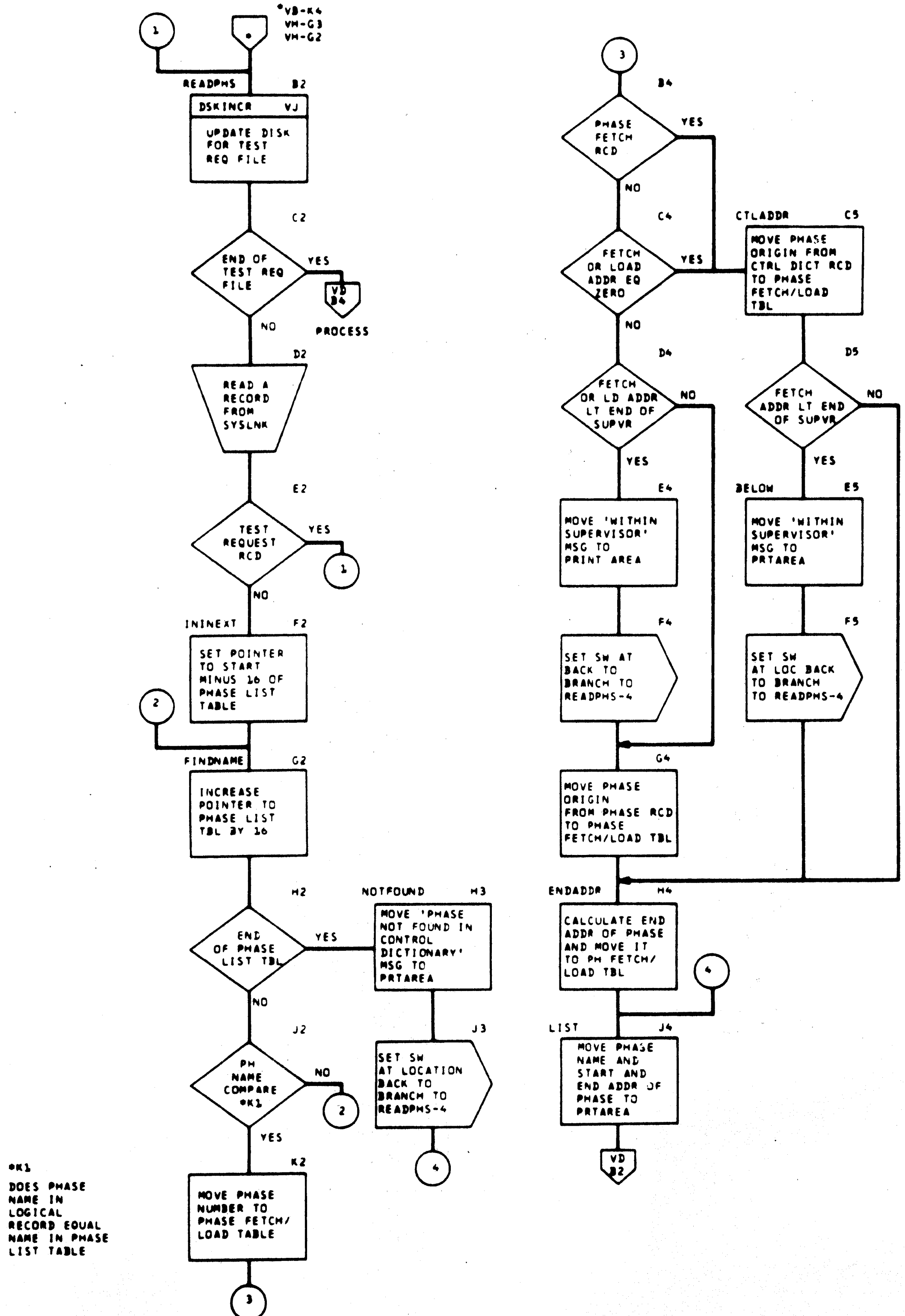

UR K5
LOOP8 B1 GET ONE BIT OF AMWP FIELD FROM PSW AND MAKE IT PRINTABLE
C1 MOVE PRINTABLE BIT TO PRINT AREA AND UP OUTPUT POINTER
D1 4 BITS OF AMWP FIELD CONVERTED
NO
YES
E1 UP PSW POINTER AND OUTPUT POINTER
F1 ALL TEN PSWS PROCESSED
G1 RTPRT UL WRITE AMWP LINE
H1 MOVE THE CONSTANT INTERRUPT CODE TO PRINT AREA
LOOP9 B2 UNPACK AND TRANSLATE INTERRUPT CODE TO PRINTABLE CHARACTERS
C2 UP OUTPUT POINTER TO NEXT FIELD
D2 ALL TEN PSWS PROCESSED
E2 RTPRT UL WRITE INTER-RUPTION CODE LINE
F2 MOVE THE CONSTANT INSTRUCTION LENGTH TO PRINT AREA
LOOP10 G2 GET INST LNG CONV TO DCML UNPACK AND MOVE TO PRINT AREA
H2 UP OUTPUT POINTER
J2 ALL TEN PSWS PROCESSED
K2 RTPRT UL WRITE INST LENGTH LINE
B3 MOVE THE CONSTANT COND CODE TO PRINT AREA
LOOP11 C3 GET CONDITION CODE. CONVERT TO DECIMAL UNPACK AND MOVE TO PRINT AREA
D3 ALL TEN PSWS PROCESSED
E3 RTPRT UL WRITE COND CODE LINE
F3 MOVE THE CONSTANT SYSTEM MASK TO PRINT AREA
LOOP12 G3 GET ONE BIT OF PROGRAM MASK AND CONVERT IT TO PRINTABLE CHARACTER
H3 MOVE CHAR TO PRINT AREA AND UP OUTPUT POINTER
J3 ALL FOUR BITS OF PSW CONV
B4 UP OUTPUT POINTER AND PSW POINTER
C4 ALL TEN PSWS PROCESSED
D4 RTPRT UL WRITE PROG MASK LINE
E4 MOVE THE CONSTANT INSTRUCTION ADDRESS TO PRINT AREA
LOOP14 F4 UNPACK AND TRANSLATE INST ADDR IN OUTPUT AREA AND UP OUTPUT POINTER
G4 ALL TEN PSWS PROCESSED
H4 RTPRT UL WRITE INST ADDRESS LINE
J4 SET UP TO SPACE TWO LINES
B5 MOVE THE CONSTANTS FOR LAST PANEL LINE TO PRINT AREA
C5 SET UP OUTPUT AND INPUT POINTERS AND LOOP CTR
LOOP15 D5 UNPACK AND TRANSLATE TO PRINTABLE CHARACTER
MODCNT E5 UPDATE OUTPUT AND INPUT POINTER TO NEXT FIELD TO BE CONV
F5 ALL THREE FIELDS CONV
G5 RTPRT UL WRITE LAST PANEL LINE
H5 SET UP TO SPACE TWO LINES
J5 RESTORE LINK REG FOR RETURN TO MAIN LINE
K5 RETURN

Chart VA. Phase List and Core Map (ATLEFG1) (Refer to Chart 13) (Part 1 of 8)




4
VA J4
READCTD
B2
DSKINCR VJ
UPDATE DISK OUTPUT ADDRESS
C2
READ A CONTROL DICT RCD FROM SYSLNK
D2
GET THE NUMBER OF LOGICAL RCDS IN CTRL DICT RCD
E2
POINT TO FIRST LOGICAL RECORD
3
TESTNAME
F2
PHASE RCD
NO
2
YES
G2
MOVE PHASE NAME AND NO. TO PHASE LIST TABLE
J2
SUBTRACT START ADDR OF PHASE FROM END ADDR TO GET LNG OF PHASE
K2
MOVE LNG OF PHASE TO PHASE LIST TABLE
1
1
B4
MOVE PHASE START ADDR TO PHASE LIST TABLE
C4
ADD 16 TO ADDR OF PHASE LIST TABLE
2
MODIFIN
D4
ADD 16 TO INPUT AREA ADDR TO POINT TO NEXT LOG RECORD
E4
SUBTRACT ONE FROM NUMBER OF LOGICAL RCDS
F4
ANOTHER LOG RECORD
YES
3
NO
G4
END OF FILE
NO
4
YES
H4
STORE END ADDR OF PHASE LIST TBL IN PHSTABND
J4
SET POINTER TO START OF PHASE FETCH/ LOAD TABLE
VD H2
K4
SET SW AT BACK TO NO-OP
VC B2
READPHS



*VB-K4
VH-G3
VH-G2
READPHS B2
DSKINCR VJ
UPDATE DISK FOR TEST REQ FILE
C2
END OF TEST REQ FILE
YES
VD B4
PROCESS
NO
D2
READ A RECORD FROM SYSLNK
E2
TEST REQUEST RCD
YES
NO
ININEXT F2
SET POINTER TO START MINUS 16 OF PHASE LIST TABLE
FINDNAME G2
INCREASE POINTER TO PHASE LIST TBL BY 16
H2
END OF PHASE LIST TBL
YES
NO
NOTFOUND H3
MOVE 'PHASE NOT FOUND IN CONTROL DICTIONARY' MSG TO PRTAREA
J2
PH NAME COMPARE *K1
NO
YES
J3
SET SW AT LOCATION BACK TO BRANCH TO READPHS-4
K2
MOVE PHASE NUMBER TO PHASE FETCH/ LOAD TABLE
*K1
DOES PHASE NAME IN LOGICAL RECORD EQUAL NAME IN PHASE LIST TABLE
B4
PHASE FETCH RCD
YES
NO
C4
FETCH OR LOAD ADDR EQ ZERO
YES
NO
CTLADDR C5
MOVE PHASE ORIGIN FROM CTRL DICT RCD TO PHASE FETCH/LOAD TBL
D4
FETCH OR LD ADDR LT END OF SUPVR
NO
YES
D5
FETCH ADDR LT END OF SUPVR
NO
YES
E4
MOVE 'WITHIN SUPERVISOR' MSG TO PRINT AREA
BELOW E5
MOVE 'WITHIN SUPERVISOR' MSG TO PRTAREA
F4
SET SW AT BACK TO BRANCH TO READPHS-4
F5
SET SW AT LOC BACK TO BRANCH TO READPHS-4
G4
MOVE PHASE ORIGIN FROM PHASE RCD TO PHASE FETCH/LOAD TBL
ENDADDR H4
CALCULATE END ADDR OF PHASE AND MOVE IT TO PH FETCH/ LOAD TBL
LIST J4
MOVE PHASE NAME AND START AND END ADDR OF PHASE TO PRTAREA
VD B2

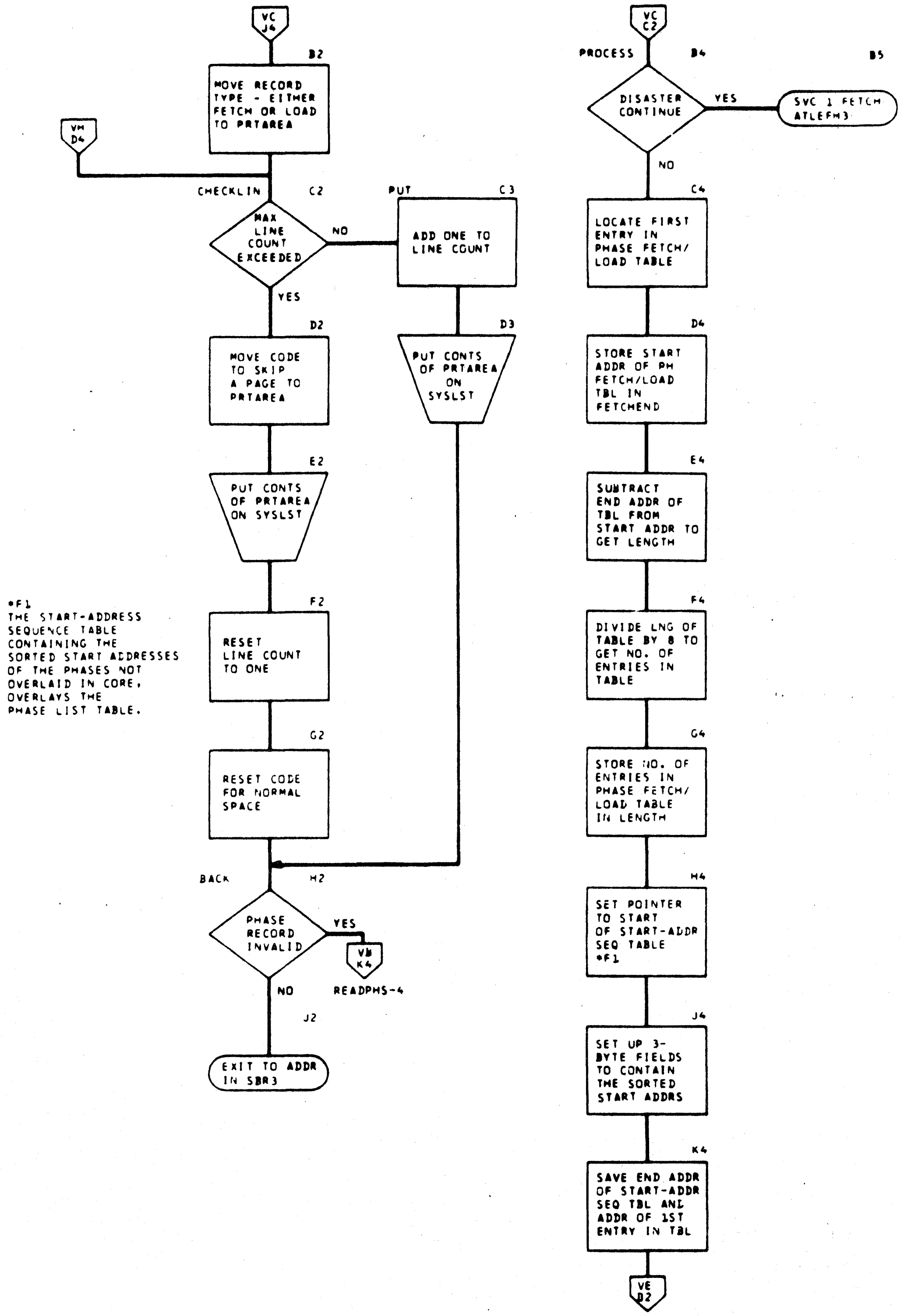
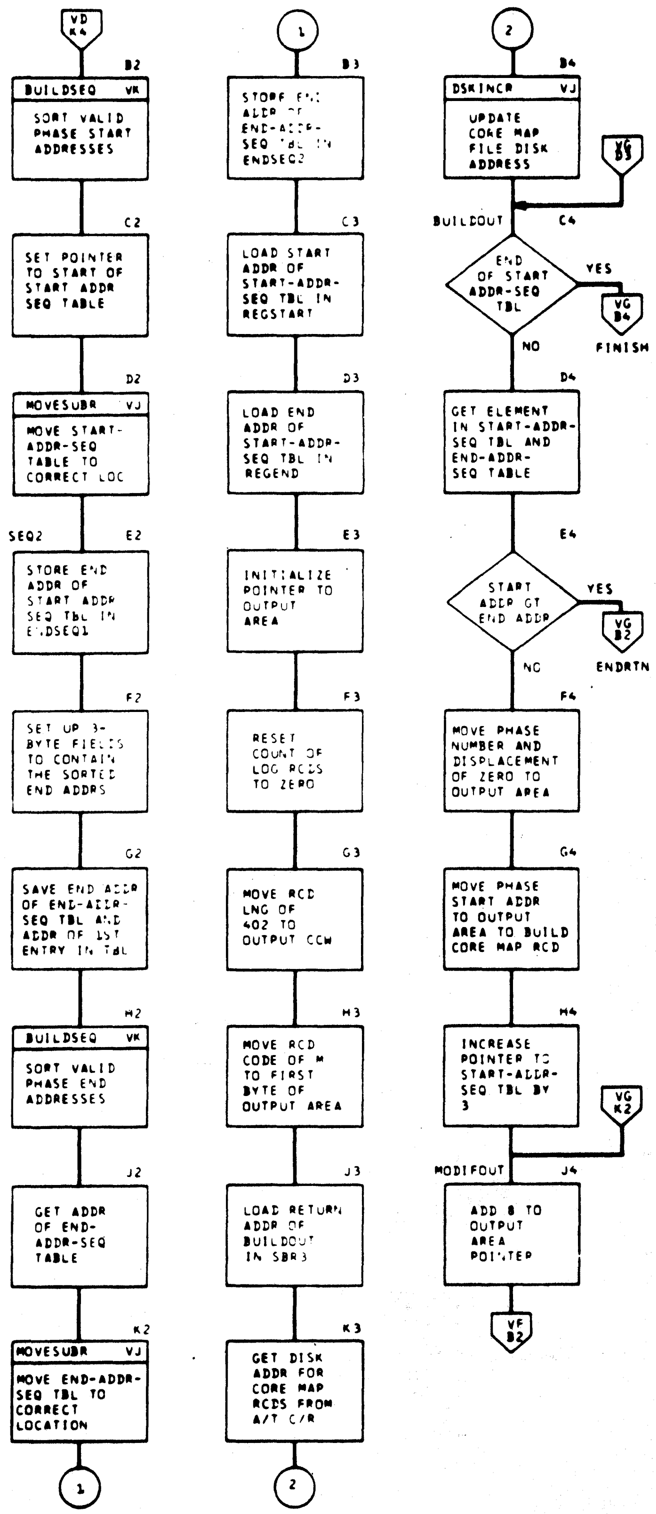
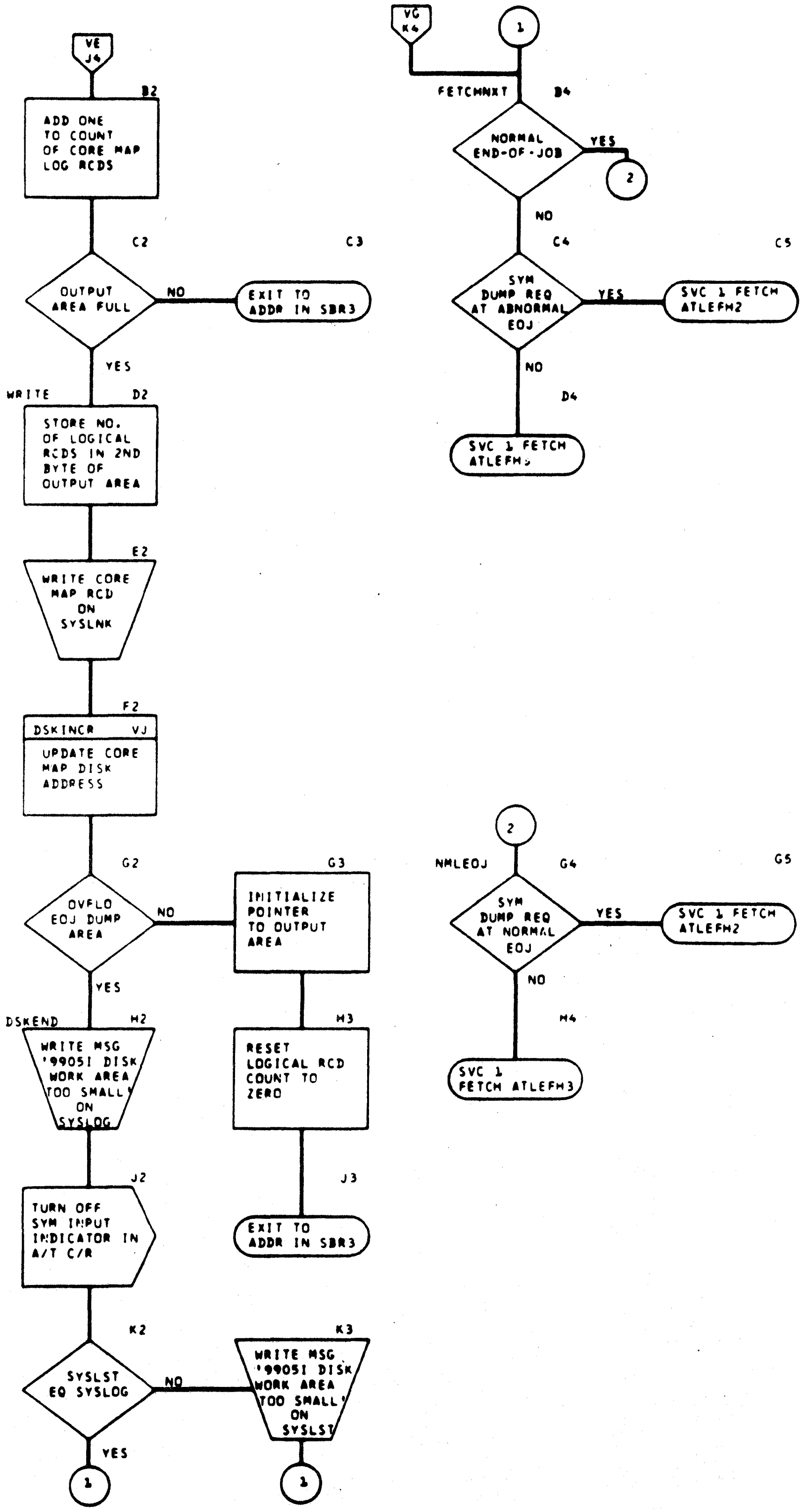
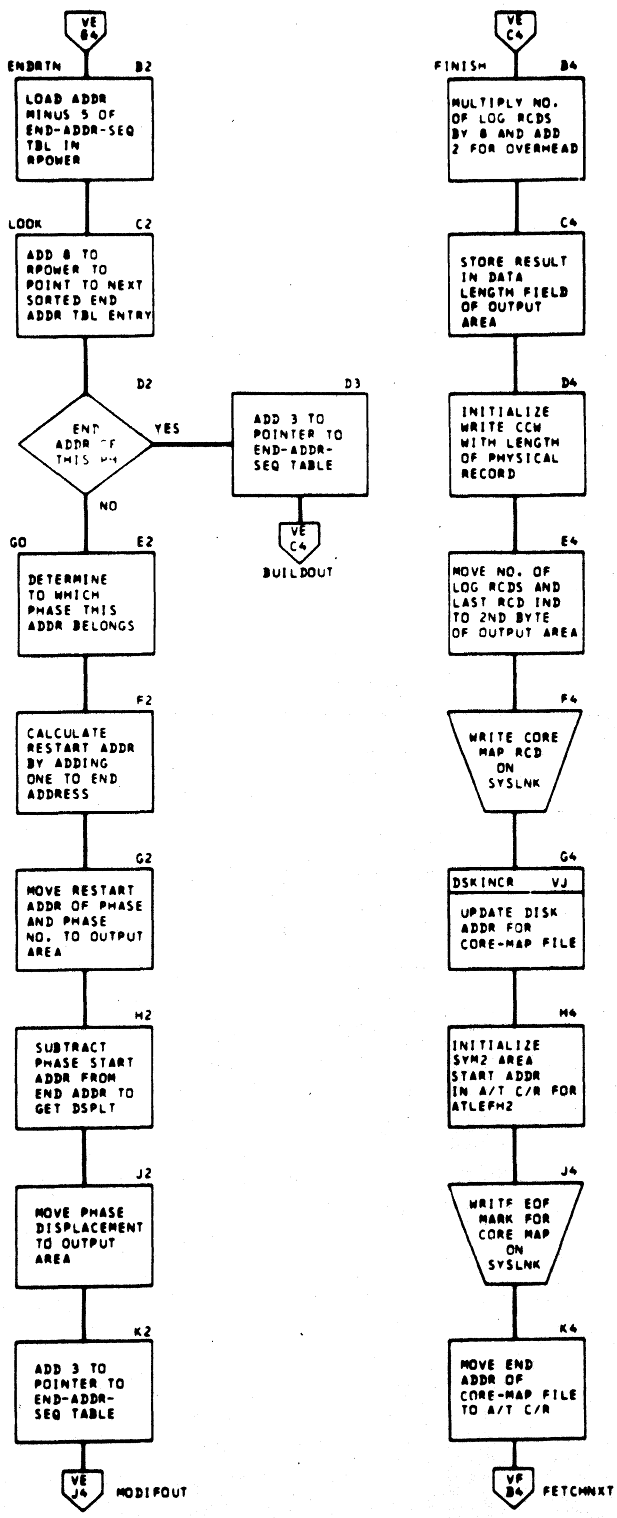

VC J4
B2
MOVE RECORD TYPE - EITHER FETCH OR LOAD TO PRTAREA
VH D4
CHECKLIN
C2
MAX LINE COUNT EXCEEDED
NO
YES
PUT
C3
ADD ONE TO LINE COUNT
D2
MOVE CODE TO SKIP A PAGE TO PRTAREA
D3
PUT CONTS OF PRTAREA ON SYSLST
E2
PUT CONTS OF PRTAREA ON SYSLST
*F1
THE START-ADDRESS SEQUENCE TABLE CONTAINING THE SORTED START ADDRESSES OF THE PHASES NOT OVERLAID IN CORE, OVERLAYS THE PHASE LIST TABLE.
F2
RESET LINE COUNT TO ONE
G2
RESET CODE FOR NORMAL SPACE
BACK
H2
PHASE RECORD INVALID
YES
VB K4
READPHS-4
NO
J2
EXIT TO ADDR IN SBR3
VC C2
PROCESS
B4
B5
DISASTER CONTINUE
YES
SVC 1 FETCH ATLEFH3
NO
C4
LOCATE FIRST ENTRY IN PHASE FETCH/ LOAD TABLE
D4
STORE START ADDR OF PH FETCH/LOAD TBL IN FETCHEND
E4
SUBTRACT END ADDR OF TBL FROM START ADDR TO GET LENGTH
F4
DIVIDE LNG OF TABLE BY 8 TO GET NO. OF ENTRIES IN TABLE
G4
STORE NO. OF ENTRIES IN PHASE FETCH/ LOAD TABLE IN LENGTH
H4
SET POINTER TO START OF START-ADDR SEQ TABLE *F1
J4
SET UP 3- BYTE FIELDS TO CONTAIN THE SORTED START ADDRS
K4
SAVE END ADDR OF START-ADDR SEQ TBL AND ADDR OF 1ST ENTRY IN TBL
VE B2

Chart VE. Phase List and Core Map (ATLEFG1) (Refer to Chart 13) (Part 5 of 8)




VE J4
B2
ADD ONE TO COUNT OF CORE MAP LOG RCDS
C2
OUTPUT AREA FULL
NO
C3
EXIT TO ADDR IN SBR3
YES
WRITE
D2
STORE NO. OF LOGICAL RCDS IN 2ND BYTE OF OUTPUT AREA
E2
WRITE CORE MAP RCD ON SYSLNK
F2
DSKINCR VJ
UPDATE CORE MAP DISK ADDRESS
G2
OVFLO EOJ DUMP AREA
NO
G3
INITIALIZE POINTER TO OUTPUT AREA
YES
DSKEND
H2
WRITE MSG '9905I DISK WORK AREA TOO SMALL' ON SYSLOG
H3
RESET LOGICAL RCD COUNT TO ZERO
J2
TURN OFF SYM INPUT INDICATOR IN A/T C/R
J3
EXIT TO ADDR IN SBR3
K2
SYSLST EQ SYSLOG
NO
K3
WRITE MSG '9905I DISK WORK AREA TOO SMALL' ON SYSLST
YES
1
1
VG K4
1
FETCHNXT
B4
NORMAL END-OF-JOB
YES
2
NO
C4
SYM DUMP REQ AT ABNORMAL EOJ
YES
C5
SVC 1 FETCH ATLEFH2
NO
D4
SVC 1 FETCH ATLEFH3
2
NMLEOJ
G4
SYM DUMP REQ AT NORMAL EOJ
YES
G5
SVC 1 FETCH ATLEFH2
NO
H4
SVC 1 FETCH ATLEFH3


VE B4
ENDRTN B2
LOAD ADDR MINUS 5 OF END-ADDR-SEQ TBL IN RPOWER
LOOK C2
ADD 8 TO RPOWER TO POINT TO NEXT SORTED END ADDR TBL ENTRY
D2
END ADDR OF THIS PH
YES
D3
ADD 3 TO POINTER TO END-ADDR-SEQ TABLE
VE C4
BUILDOUT
NO
GO E2
DETERMINE TO WHICH PHASE THIS ADDR BELONGS
F2
CALCULATE RESTART ADDR BY ADDING ONE TO END ADDRESS
G2
MOVE RESTART ADDR OF PHASE AND PHASE NO. TO OUTPUT AREA
H2
SUBTRACT PHASE START ADDR FROM END ADDR TO GET DSPLT
J2
MOVE PHASE DISPLACEMENT TO OUTPUT AREA
K2
ADD 3 TO POINTER TO END-ADDR-SEQ TABLE
VE J4
MODIFOUT
VE C4
FINISH B4
MULTIPLY NO. OF LOG RCDS BY 8 AND ADD 2 FOR OVERHEAD
C4
STORE RESULT IN DATA LENGTH FIELD OF OUTPUT AREA
D4
INITIALIZE WRITE CCW WITH LENGTH OF PHYSICAL RECORD
E4
MOVE NO. OF LOG RCDS AND LAST RCD IND TO 2ND BYTE OF OUTPUT AREA
F4
WRITE CORE MAP RCD ON SYSLNK
G4
DSKINCR VJ
UPDATE DISK ADDR FOR CORE-MAP FILE
H4
INITIALIZE SYM2 AREA START ADDR IN A/T C/R FOR ATLEFH2
J4
WRITE EOF MARK FOR CORE MAP ON SYSLNK
K4
MOVE END ADDR OF CORE-MAP FILE TO A/T C/R
VF B4
FETCHNXT

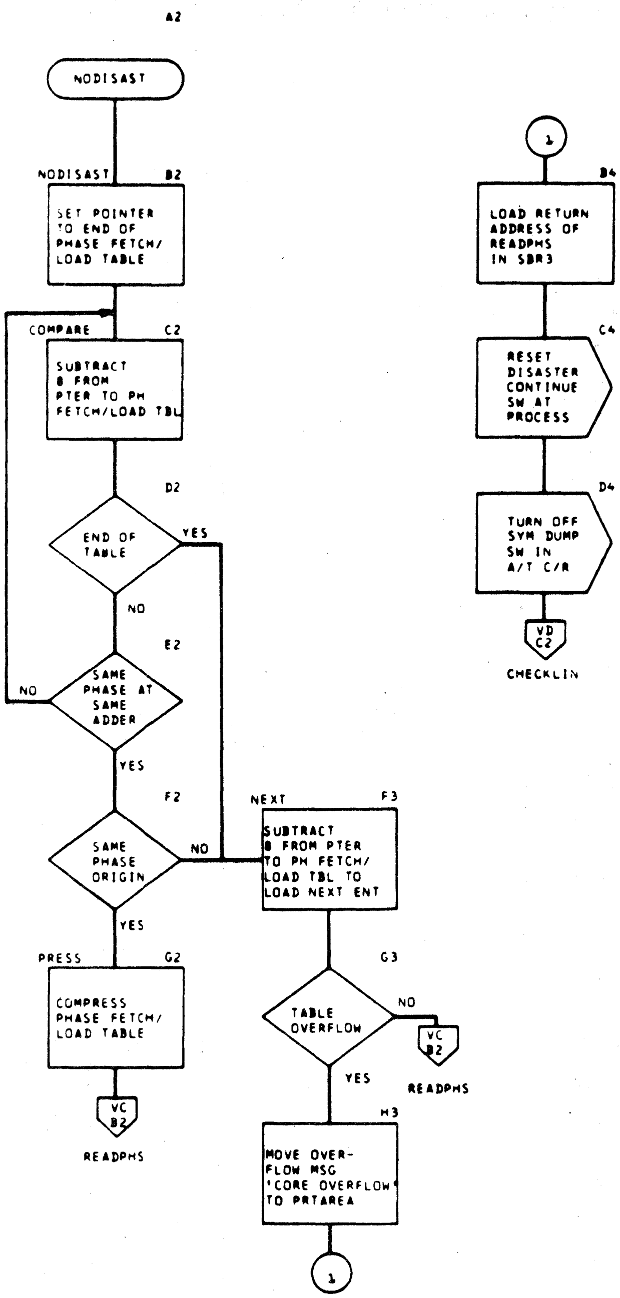

A2
NODISAST
NODISAST B2
SET POINTER TO END OF PHASE FETCH/ LOAD TABLE
COMPARE C2
SUBTRACT 8 FROM PTER TO PH FETCH/LOAD TBL
D2
END OF TABLE
YES
NO
E2
SAME PHASE AT SAME ADDER
NO
YES
F2
SAME PHASE ORIGIN
NO
YES
PRESS G2
COMPRESS PHASE FETCH/ LOAD TABLE
VC B2
READPHS
NEXT F3
SUBTRACT 8 FROM PTER TO PH FETCH/ LOAD TBL TO LOAD NEXT ENT
G3
TABLE OVERFLOW
NO
VC B2
READPHS
YES
H3
MOVE OVER-FLOW MSG 'CORE OVERFLOW' TO PRTAREA
1
1
B4
LOAD RETURN ADDRESS OF READPHS IN SBR3
C4
RESET DISASTER CONTINUE SW AT PROCESS
D4
TURN OFF SYM DUMP SW IN A/T C/R
VD C2
CHECKLIN

Chart VJ. Phase List and Core Map Subroutines (ATLEFG1) (Refer to Chart 13) (Part 1 of 3)

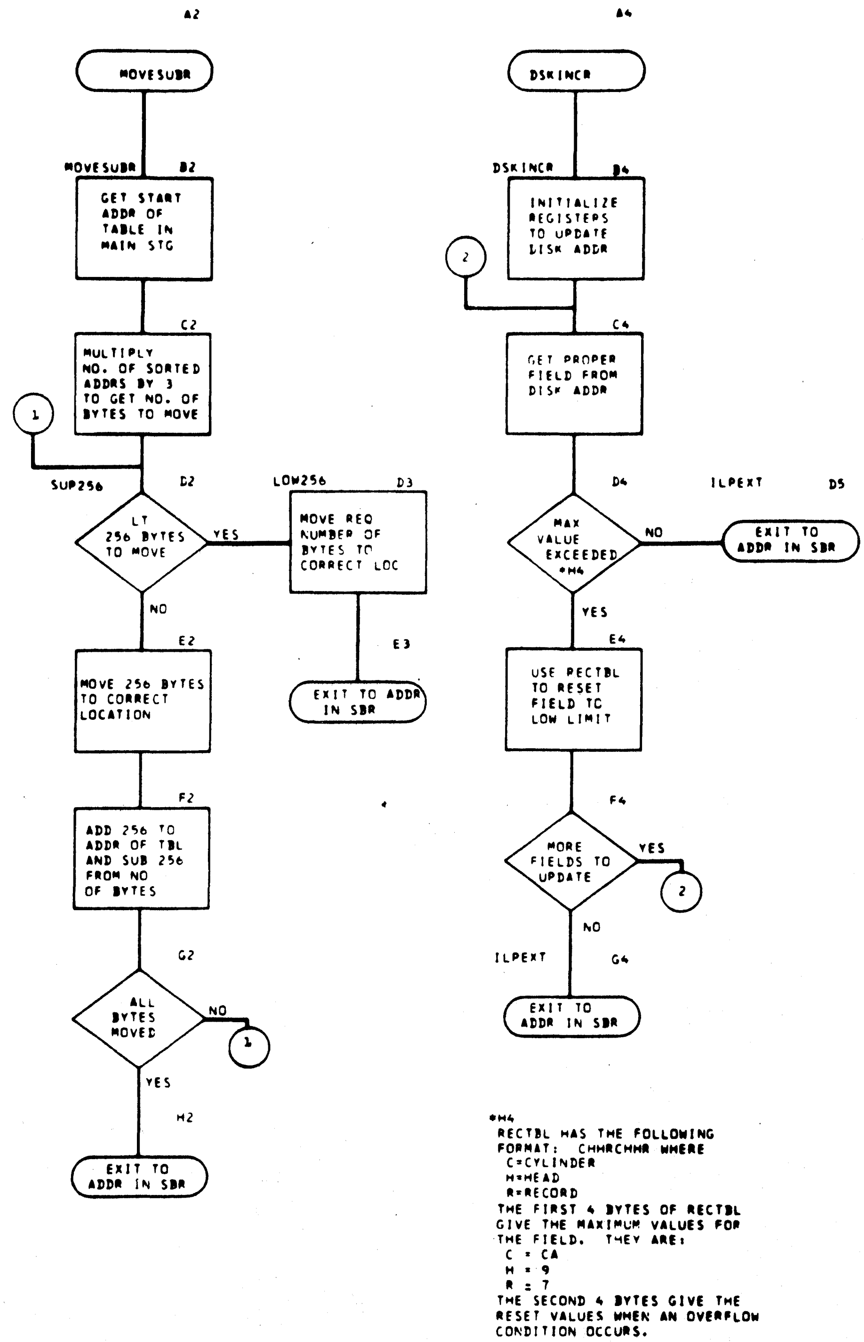


Chart VK. Phase List and Core Map Subroutines (ATLEFG1) (Refer to Chart 13) (Part 2 of 3)

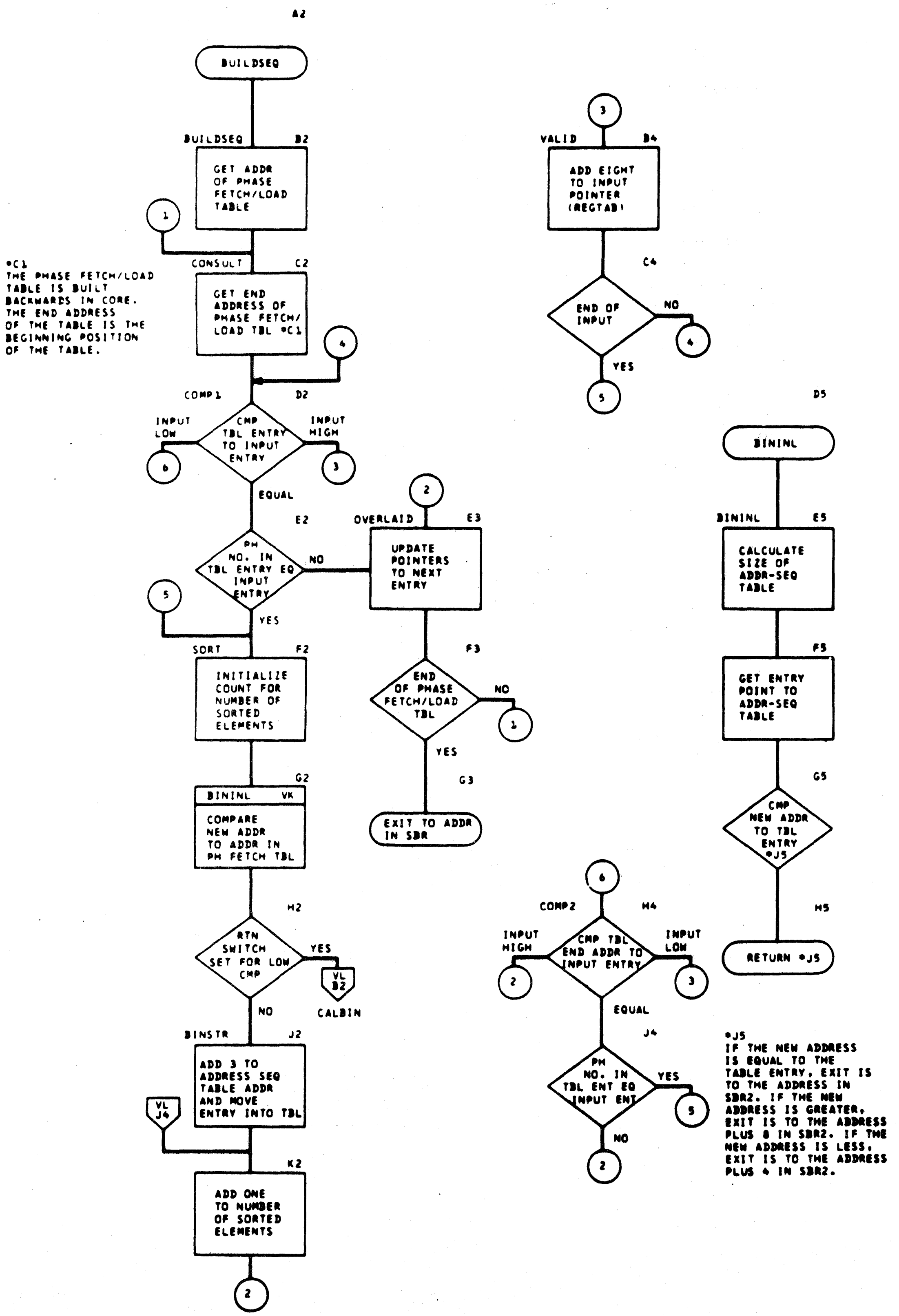
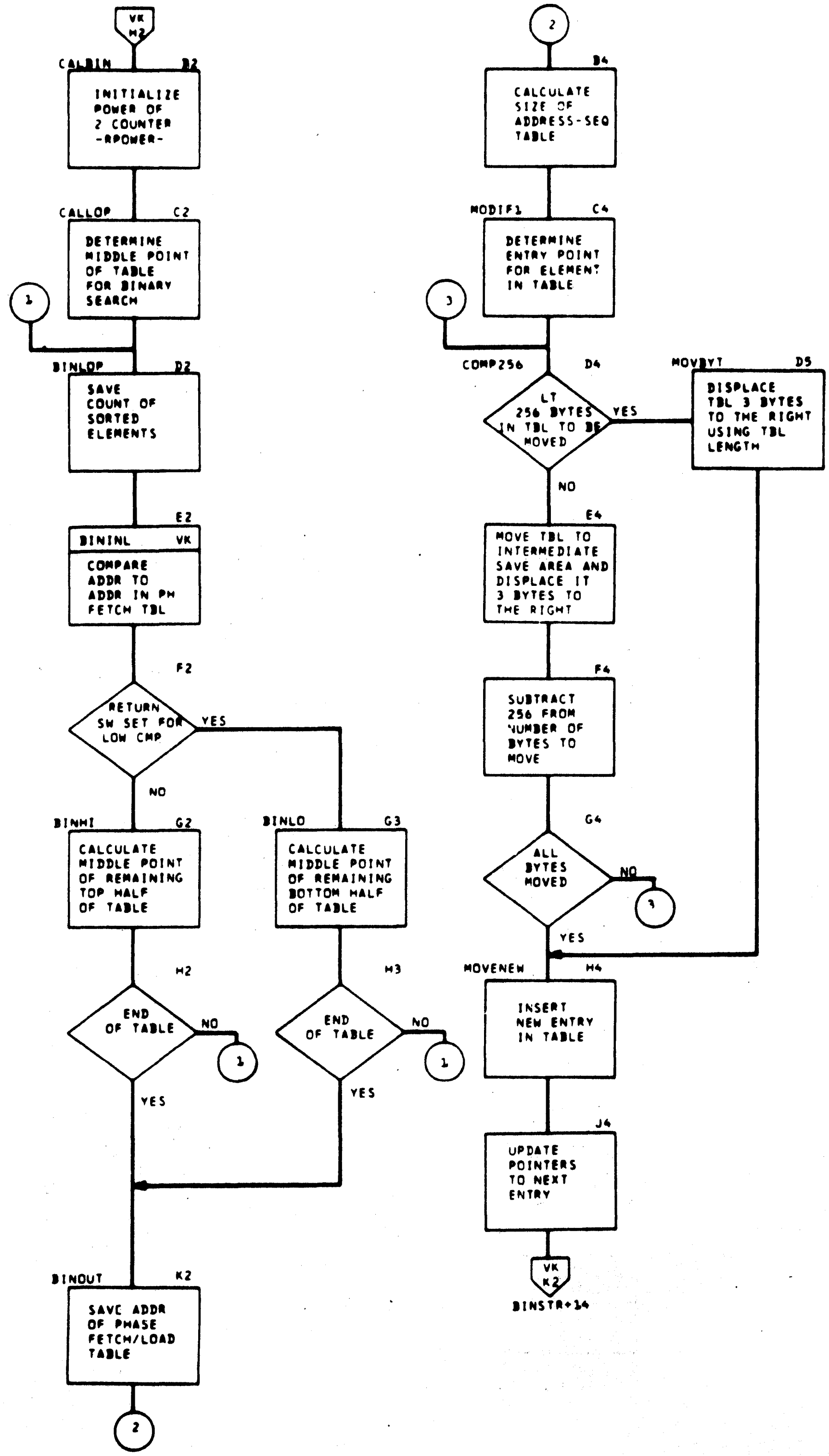
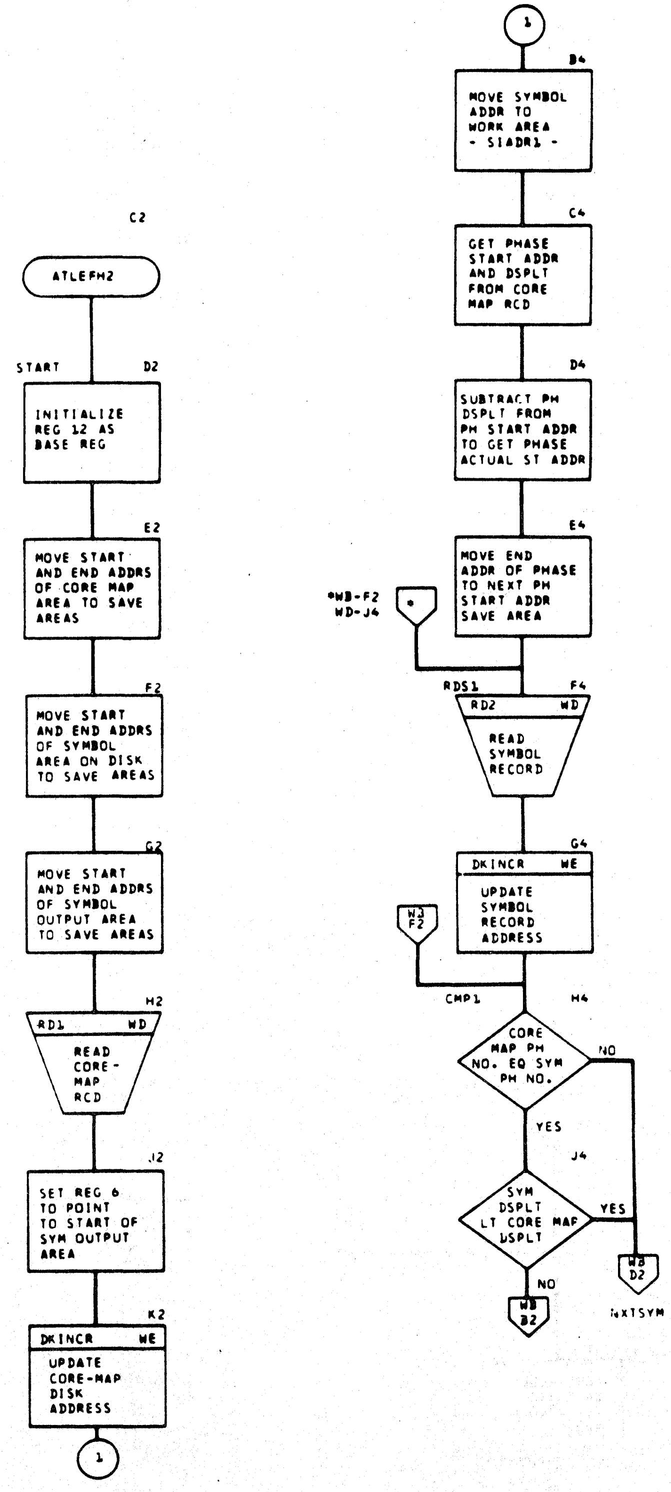


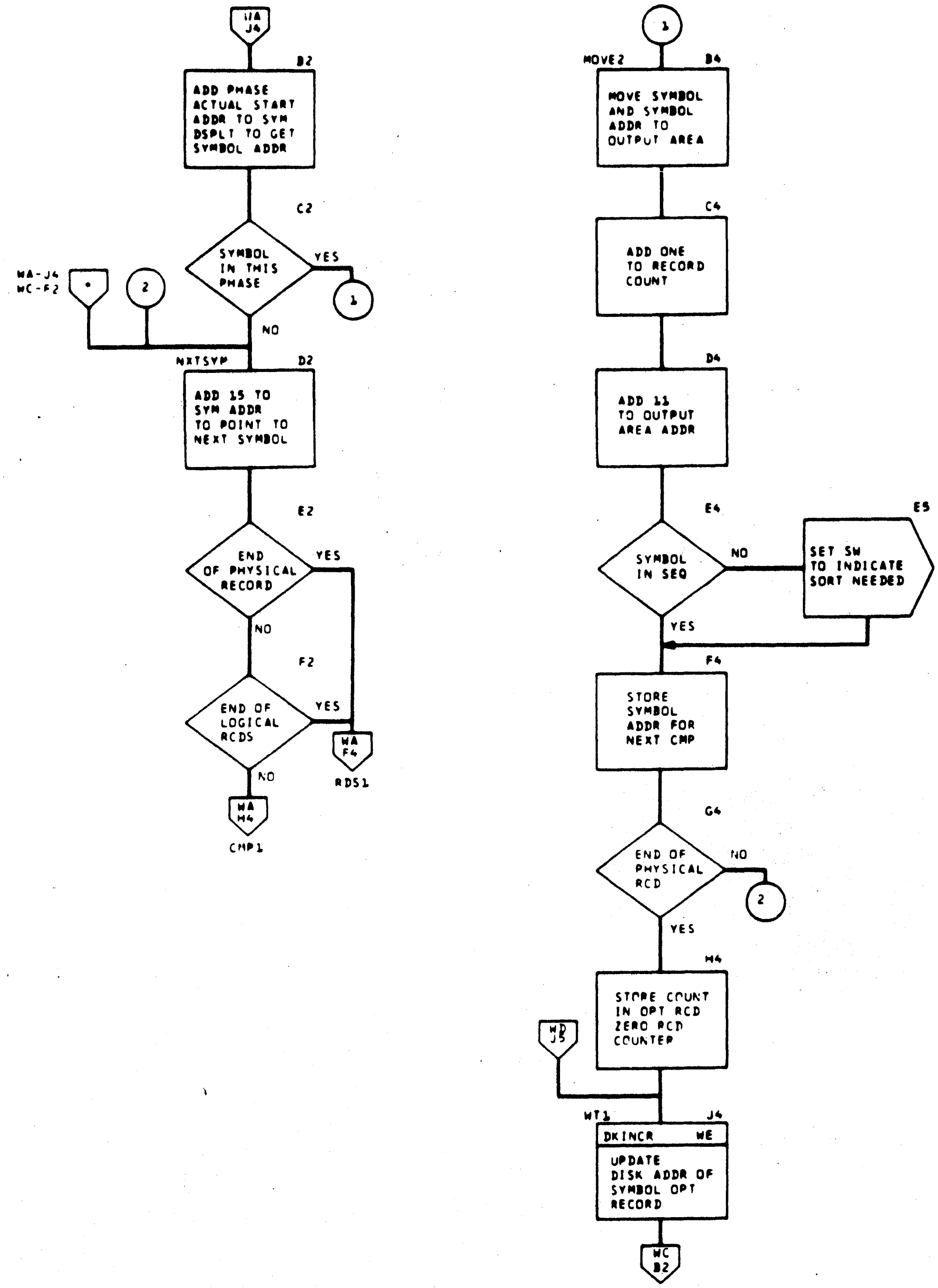
Chart VL. Phase List and Core Map Subroutines (ATLEFG1) (Refer to Chart 13) (Part 3 of 3)

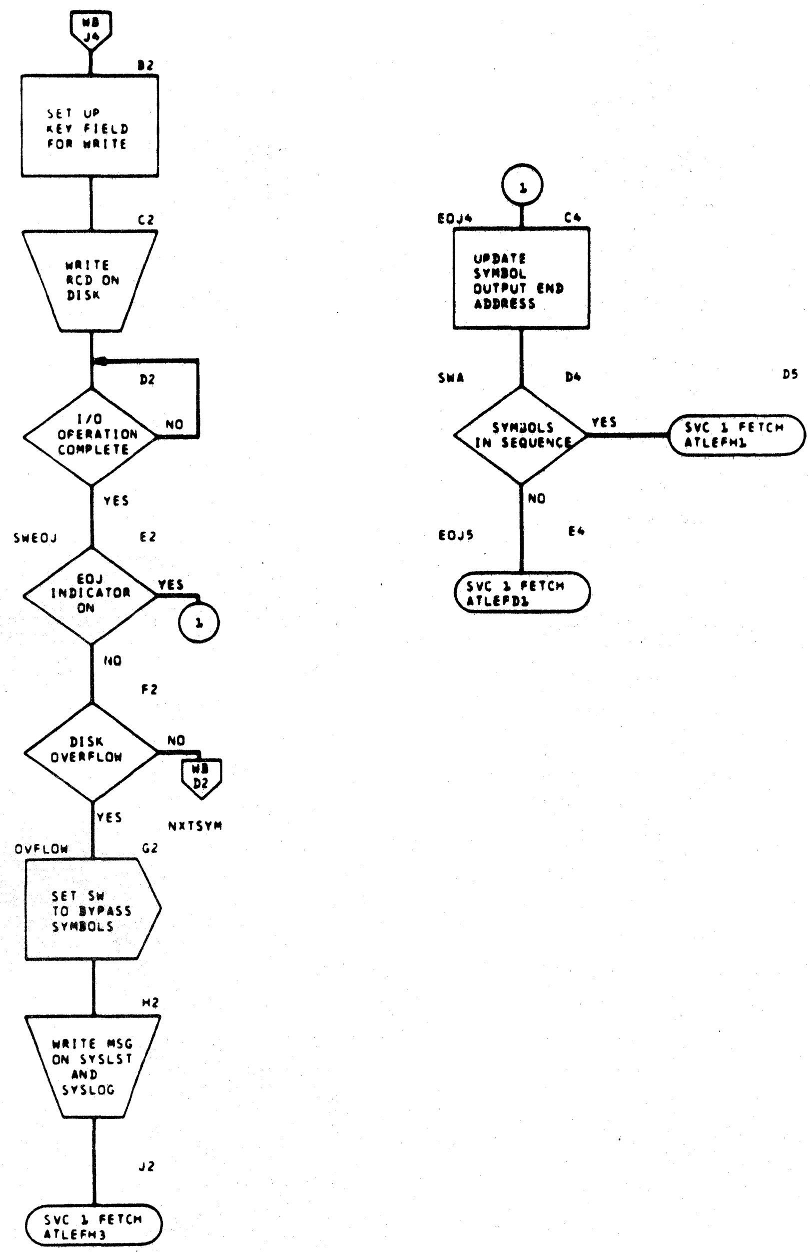




C2
ATLEFH2
START D2
INITIALIZE REG 12 AS BASE REG
E2
MOVE START AND END ADDRS OF CORE MAP AREA TO SAVE AREAS
F2
MOVE START AND END ADDRS OF SYMBOL AREA ON DISK TO SAVE AREAS
G2
MOVE START AND END ADDRS OF SYMBOL OUTPUT AREA TO SAVE AREAS
H2
RD1 WD
READ CORE-MAP RCD
J2
SET REG 6 TO POINT TO START OF SYM OUTPUT AREA
K2
DKINCR WE
UPDATE CORE-MAP DISK ADDRESS
1
1
B4
MOVE SYMBOL ADDR TO WORK AREA - S1ADR1 -
C4
GET PHASE START ADDR AND DSPLT FROM CORE MAP RCD
D4
SUBTRACT PH DSPLT FROM PH START ADDR TO GET PHASE ACTUAL ST ADDR
E4
MOVE END ADDR OF PHASE TO NEXT PH START ADDR SAVE AREA
*WB-F2
WD-J4
RDS1 F4
RD2 WD
READ SYMBOL RECORD
G4
DKINCR WE
UPDATE SYMBOL RECORD ADDRESS
WB F2
CMP1 H4
CORE MAP PH NO. EQ SYM PH NO.
NO
YES
J4
SYM DSPLT LT CORE MAP DSPLT
YES
NO
WB B2
WB D2
NXTSYM

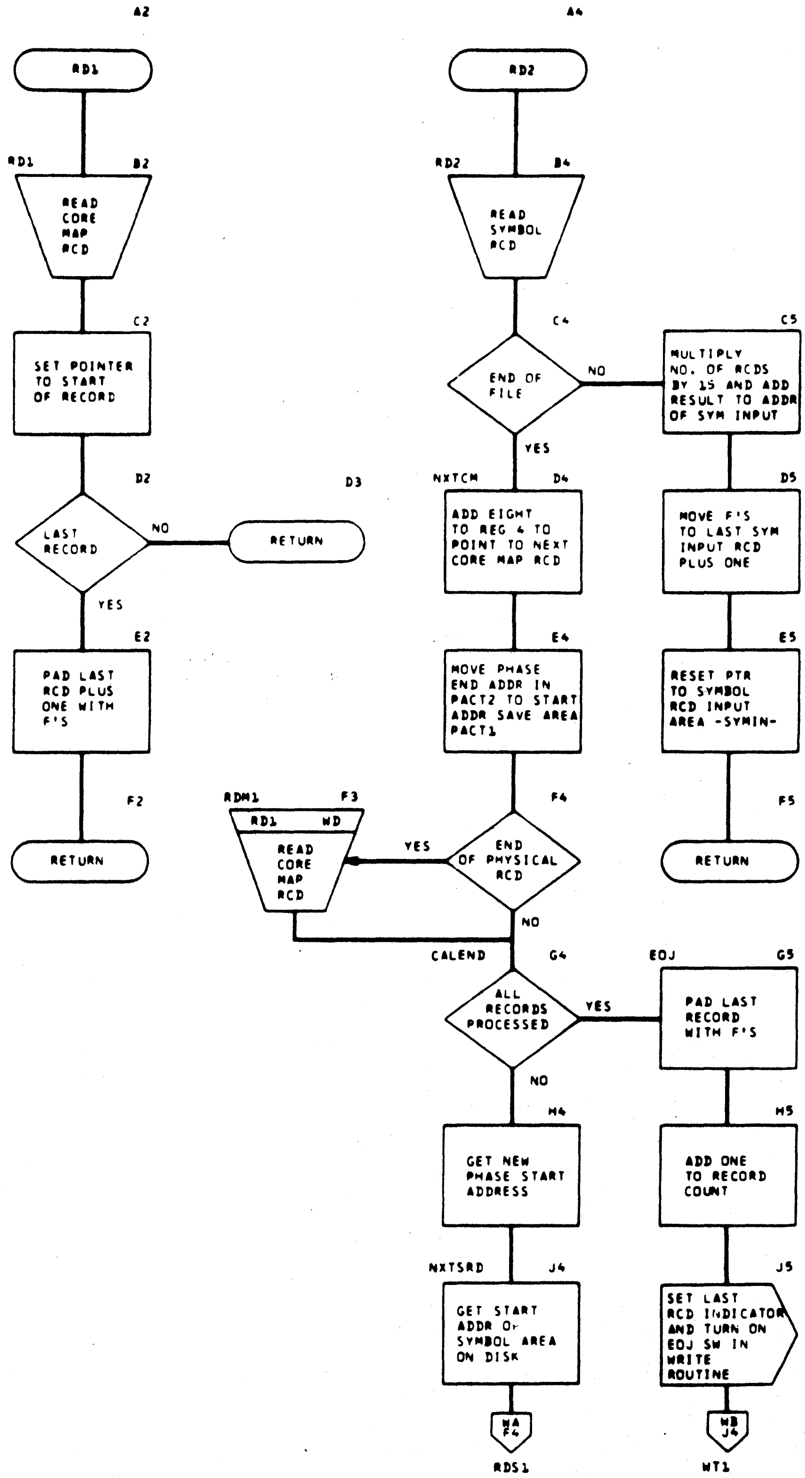
Chart WB. Symbol Organization (ATLEFH2) (Refer to Chart 13) (Part 2 of 3)





WB J4
B2
SET UP KEY FIELD FOR WRITE
C2
WRITE RCD ON DISK
D2
I/O OPERATION COMPLETE
NO
YES
SWEOJ
E2
EOJ INDICATOR ON
YES
1
NO
F2
DISK OVERFLOW
NO
WB D2
NXTSYM
YES
OVFLOW
G2
SET SW TO BYPASS SYMBOLS
H2
WRITE MSG ON SYSLST AND SYSLOG
J2
SVC 1 FETCH ATLEFH3
1
EOJ4
C4
UPDATE SYMBOL OUTPUT END ADDRESS
SWA
D4
D5
SYMBOLS IN SEQUENCE
YES
SVC 1 FETCH ATLEFH1
NO
EOJ5
E4
SVC 1 FETCH ATLEFD1

Chart WD. Symbol Organization Subroutines (ATLEFH2) (Refer to Chart 13) (Part 1 of 2)



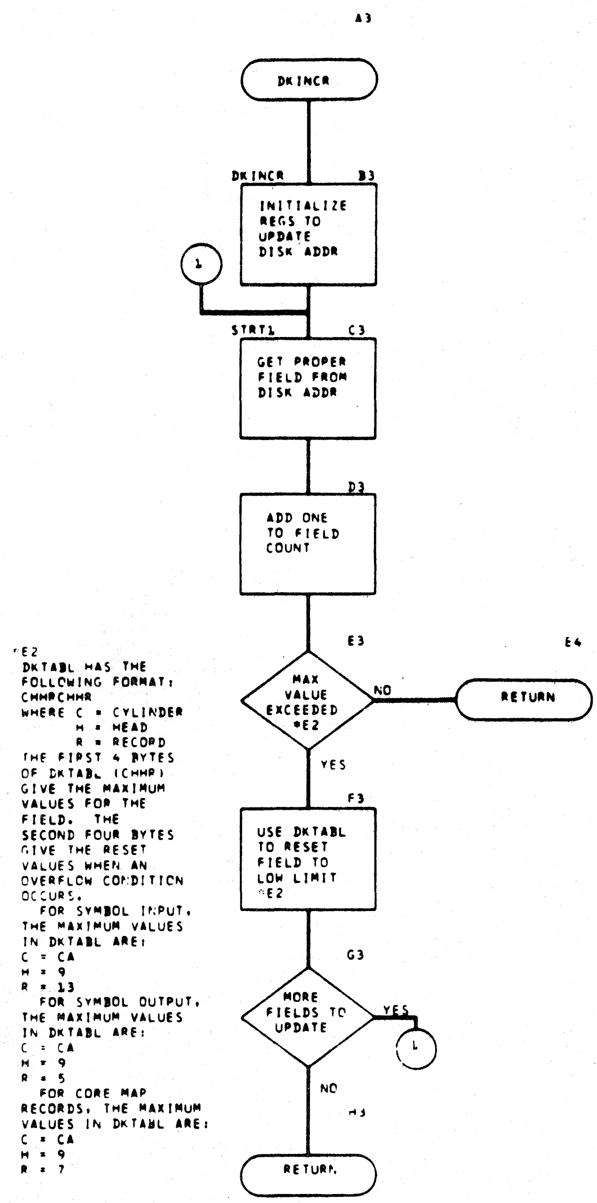

A3
DKINCR
DKINCR B3
INITIALIZE REGS TO UPDATE DISK ADDR
1
STRT1 C3
GET PROPER FIELD FROM DISK ADDR
D3
ADD ONE TO FIELD COUNT
E3
E4
*E2
DKTABL HAS THE FOLLOWING FORMAT:
CHHRCHHR
WHERE C = CYLINDER
H = HEAD
R = RECORD
THE FIRST 4 BYTES OF DKTABL (CHHR) GIVE THE MAXIMUM VALUES FOR THE FIELD. THE SECOND FOUR BYTES GIVE THE RESET VALUES WHEN AN OVERFLOW CONDITION OCCURS.
FOR SYMBOL INPUT, THE MAXIMUM VALUES IN DKTABL ARE:
C = CA
H = 9
R = 13
FOR SYMBOL OUTPUT, THE MAXIMUM VALUES IN DKTABL ARE:
C = CA
H = 9
R = 5
FOR CORE MAP RECORDS, THE MAXIMUM VALUES IN DKTABL ARE:
C = CA
H = 9
R = 7
MAX VALUE EXCEEDED *E2
NO
RETURN
YES
F3
USE DKTABL TO RESET FIELD TO LOW LIMIT *E2
G3
MORE FIELDS TO UPDATE
YES
1
NO
H3
RETURN

Chart XA. Symbol Sort (ATLEFD1) (Refer to Chart 13) (Part 1 of 2)

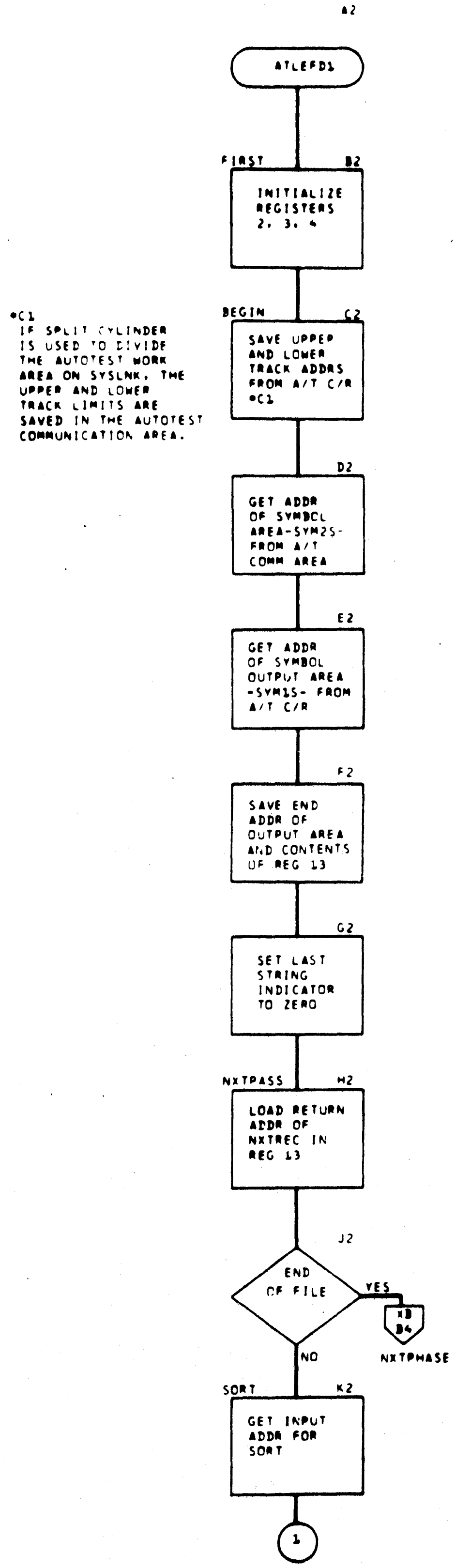


Chart XB. Symbol Sort (ATLEFD1) (Refer to Chart 13) (Part 2 of 2)

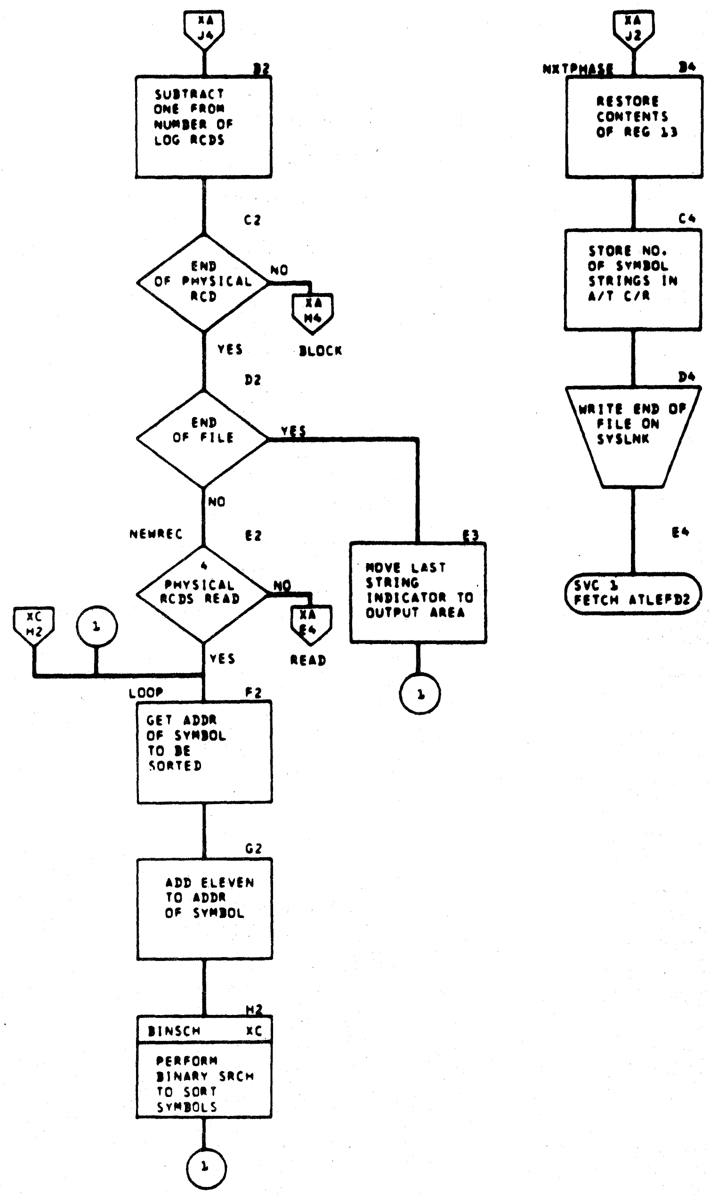
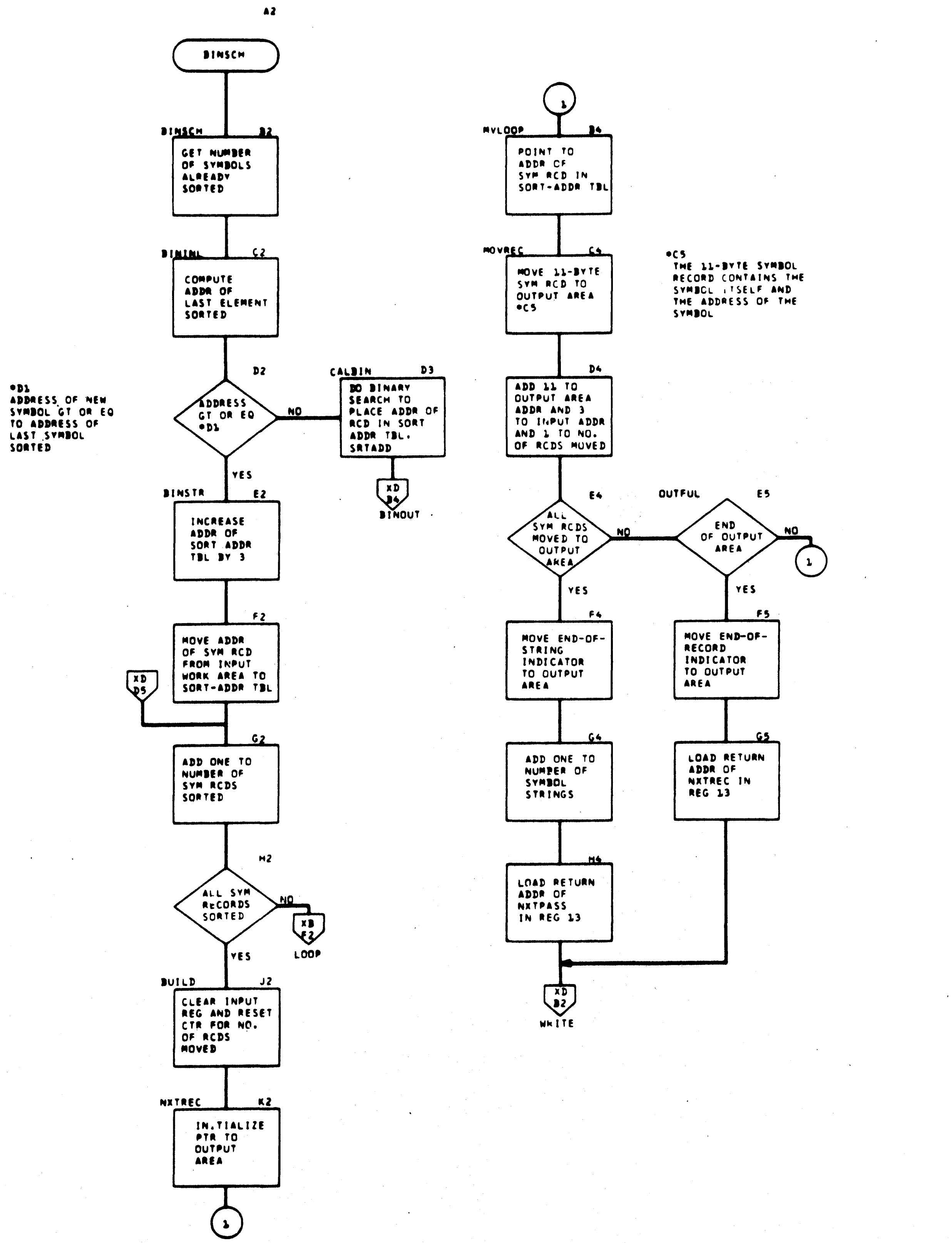


Chart XC. Symbol Sort Subroutines (ATLEFD1) (Refer to Chart 13) (Part 1 of 3)



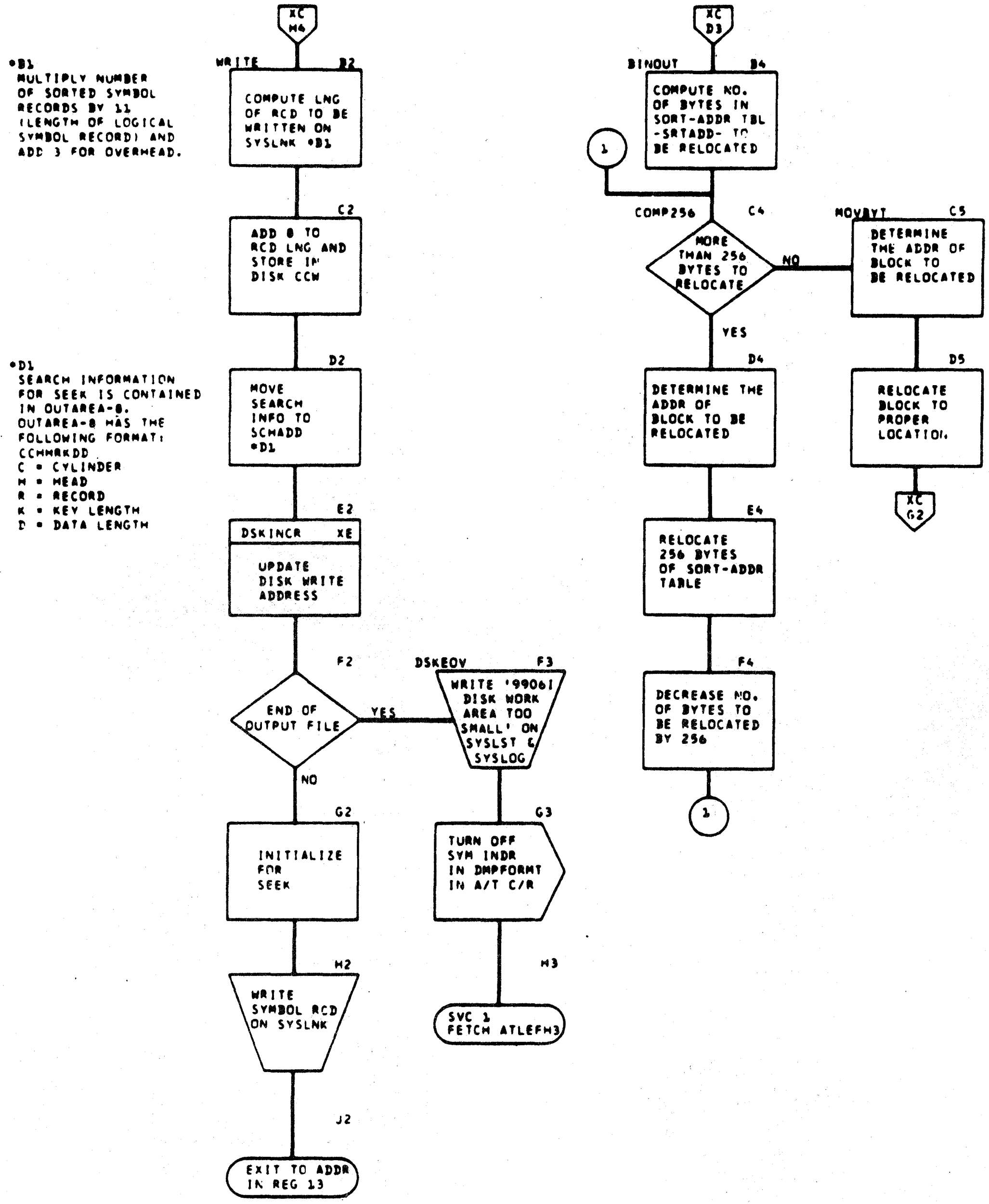

*B1
MULTIPLY NUMBER OF SORTED SYMBOL RECORDS BY 11 (LENGTH OF LOGICAL SYMBOL RECORD) AND ADD 3 FOR OVERHEAD.
*D1
SEARCH INFORMATION FOR SEEK IS CONTAINED IN OUTAREA-8. OUTAREA-8 HAS THE FOLLOWING FORMAT: CCHHRKDD
C = CYLINDER
H = HEAD
R = RECORD
K = KEY LENGTH
D = DATA LENGTH
XC M4
WRITE B2
COMPUTE LNG OF RCD TO BE WRITTEN ON SYSLNK *B1
C2
ADD 8 TO RCD LNG AND STORE IN DISK CCW
D2
MOVE SEARCH INFO TO SCHADD *D1
E2
DSKINCR XE
UPDATE DISK WRITE ADDRESS
F2
END OF OUTPUT FILE
YES
NO
DSKEOV F3
WRITE '9906I DISK WORK AREA TOO SMALL' ON SYSLST & SYSLOG
G2
INITIALIZE FOR SEEK
G3
TURN OFF SYM INDR IN DMPFORMT IN A/T C/R
H2
WRITE SYMBOL RCD ON SYSLNK
H3
SVC 1 FETCH ATLEFH3
J2
EXIT TO ADDR IN REG 13
XC D3
BINOUT B4
COMPUTE NO. OF BYTES IN SORT-ADDR TBL -SRTADD- TO BE RELOCATED
1
COMP256 C4
MORE THAN 256 BYTES TO RELOCATE
NO
YES
MOVBYT C5
DETERMINE THE ADDR OF BLOCK TO BE RELOCATED
D4
DETERMINE THE ADDR OF BLOCK TO BE RELOCATED
D5
RELOCATE BLOCK TO PROPER LOCATION
XC G2
E4
RELOCATE 256 BYTES OF SORT-ADDR TABLE
F4
DECREASE NO. OF BYTES TO BE RELOCATED BY 256
1

Chart XE. Symbol Sort Subroutines (ATLEFD1) (Refer to Chart 13) (Part 3 of 3)

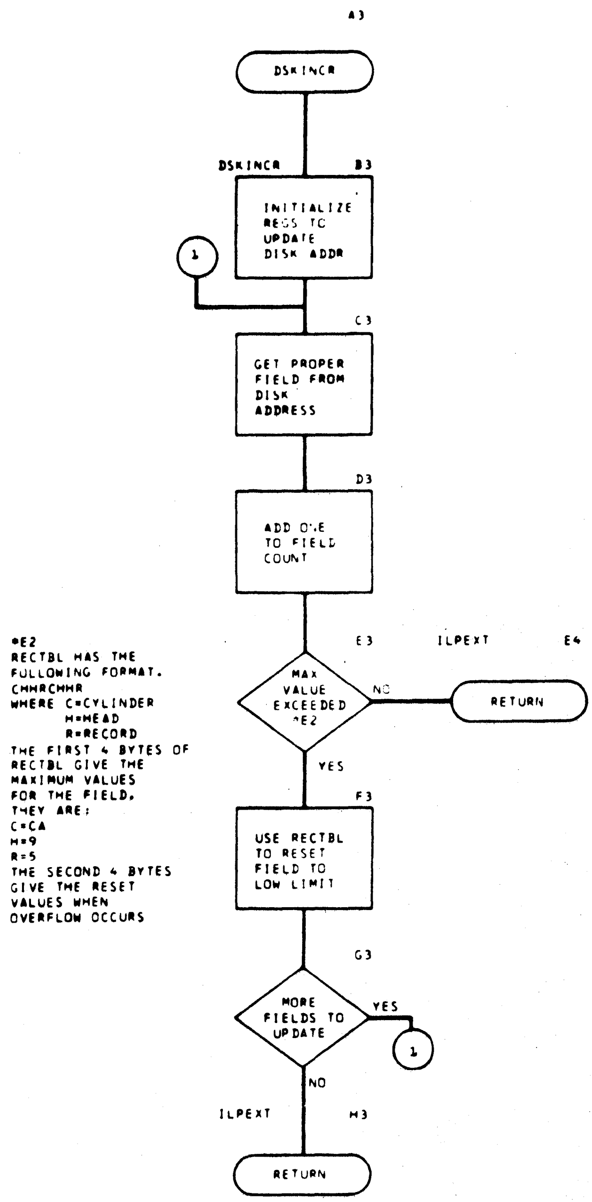


Chart XF. Symbol Merge (ATLEFD2) (Refer to Chart 13) (Part 1 of 6)

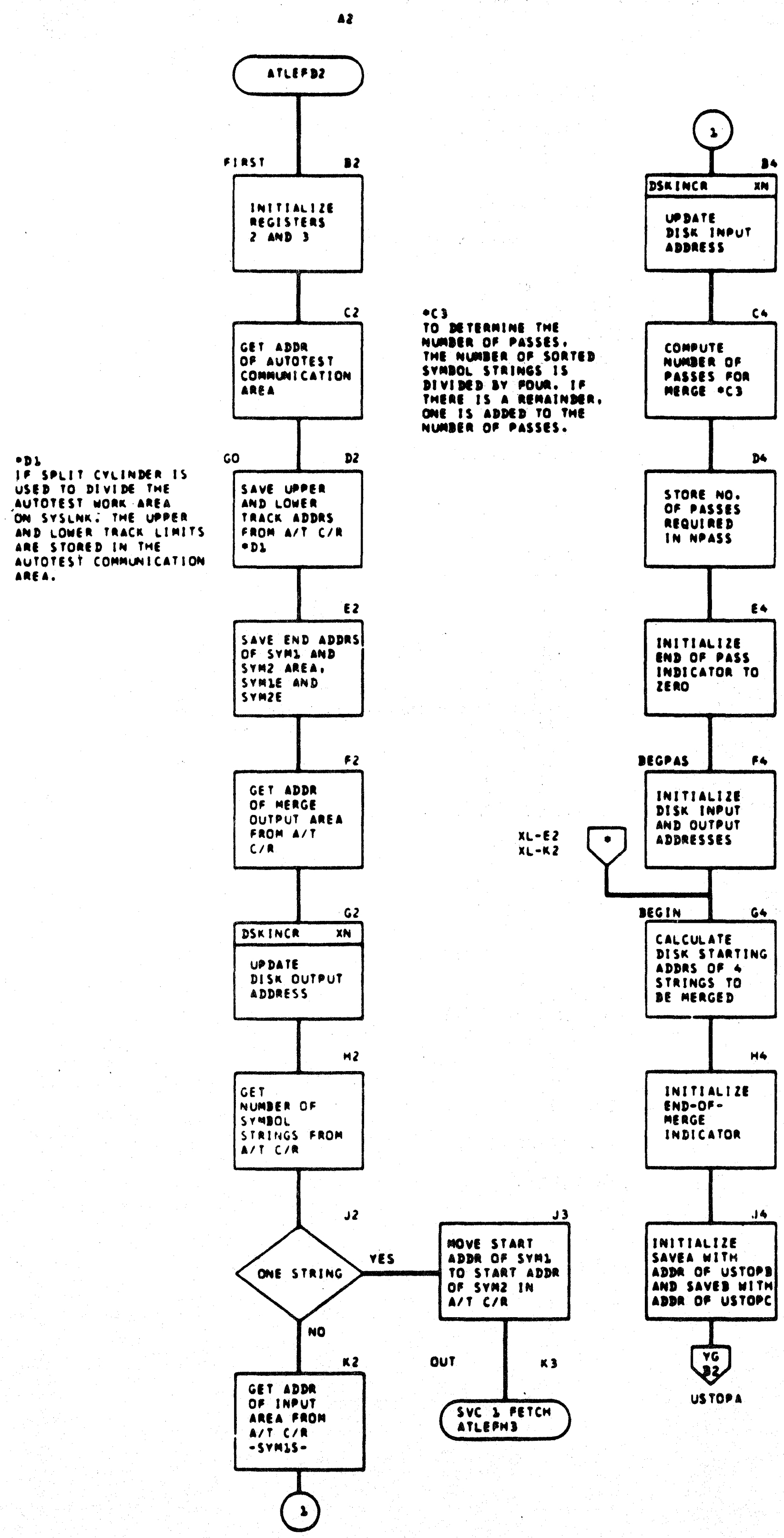
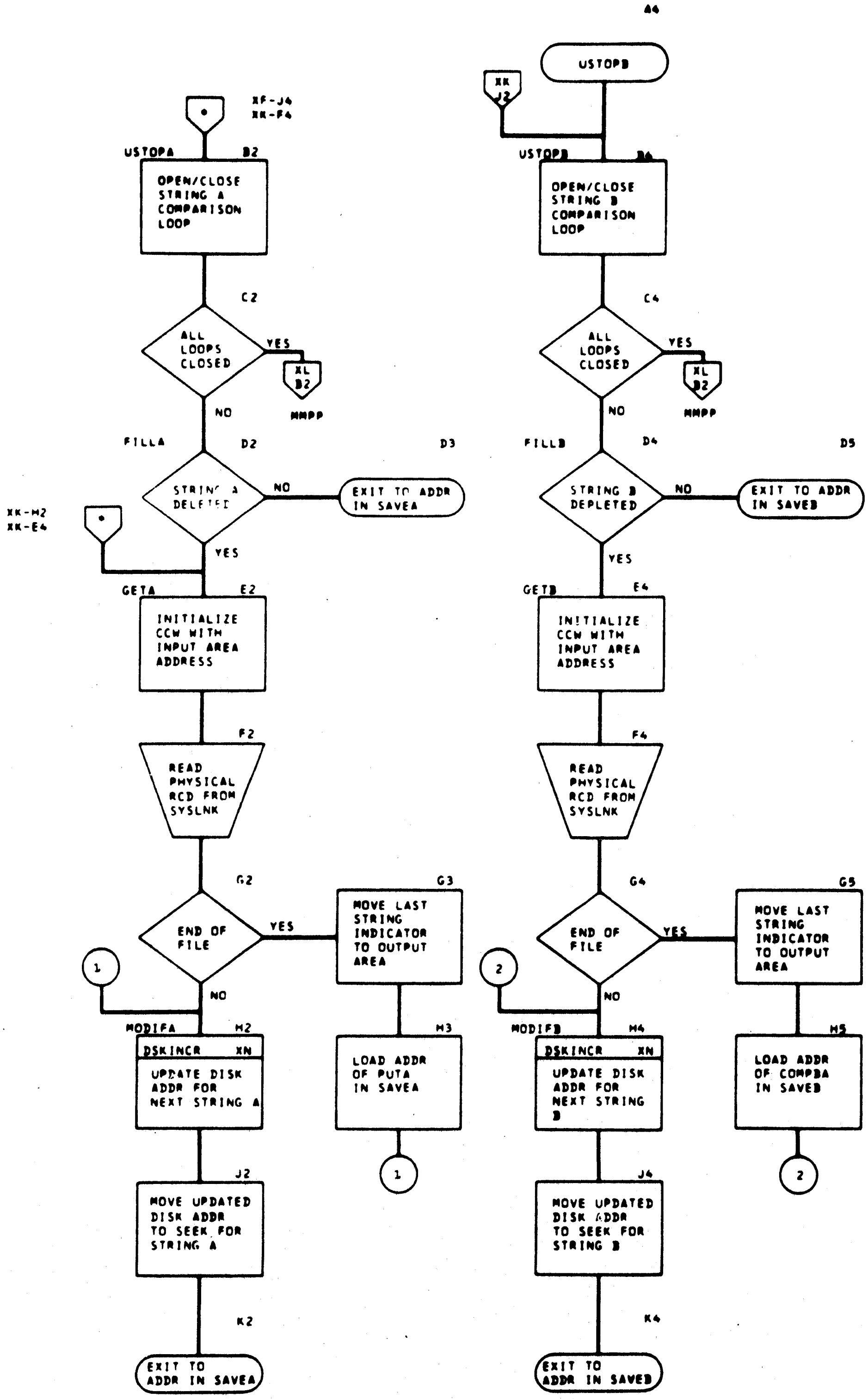
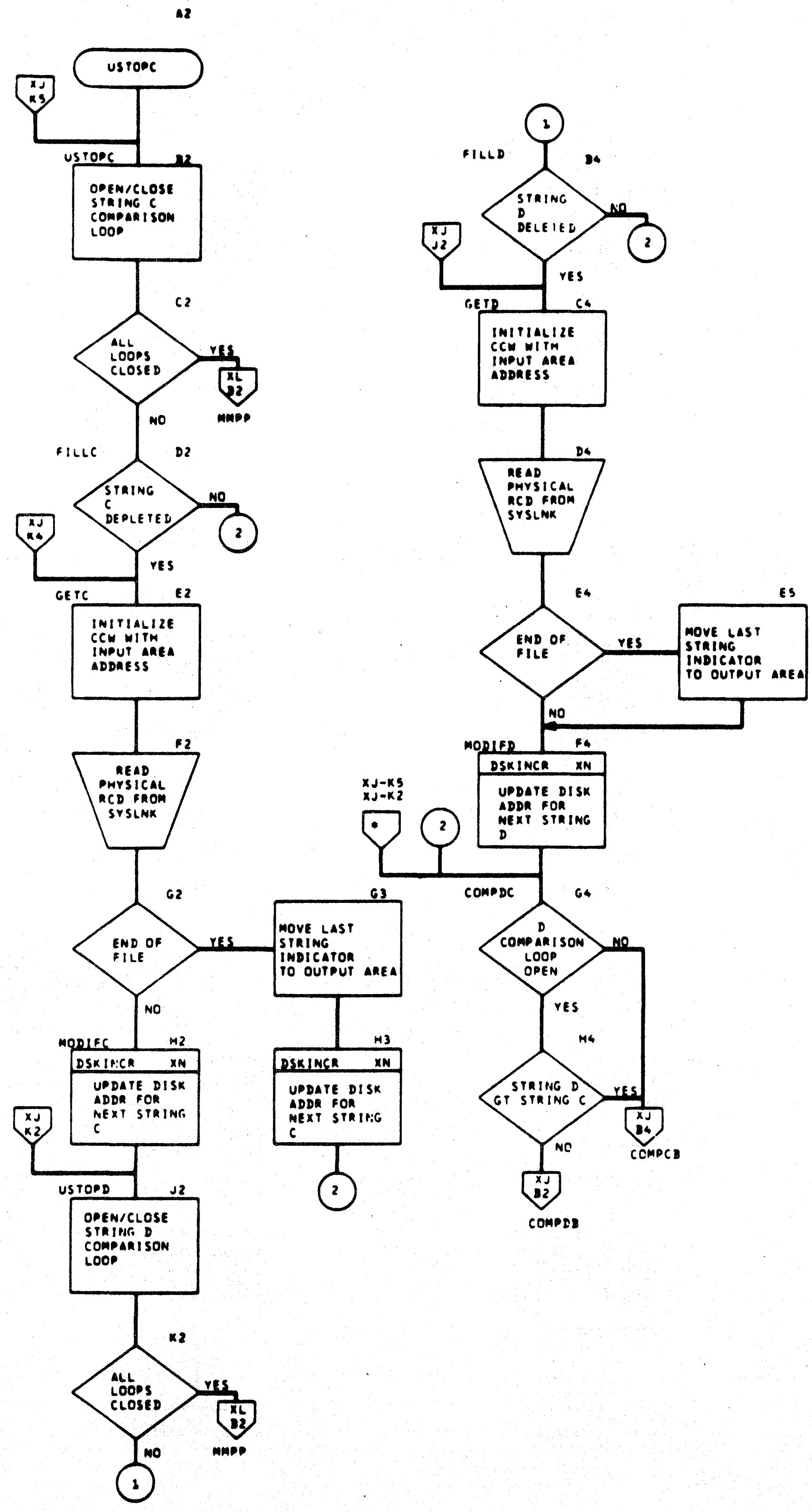


Chart XG. Symbol Merge (ATLEFD2) (Refer to Chart 13) (Part 2 of 6)




A2
USTOPC
XJ K5
USTOPC B2
OPEN/CLOSE STRING C COMPARISON LOOP
C2
ALL LOOPS CLOSED
YES
XL B2
MMPP
NO
FILLC D2
STRING C DEPLETED
NO
2
XJ K4
YES
GETC E2
INITIALIZE CCW WITH INPUT AREA ADDRESS
F2
READ PHYSICAL RCD FROM SYSLNK
G2
END OF FILE
YES
G3
MOVE LAST STRING INDICATOR TO OUTPUT AREA
NO
MODIFC H2
DSKINCR XN
UPDATE DISK ADDR FOR NEXT STRING C
H3
DSKINCR XN
UPDATE DISK ADDR FOR NEXT STRING C
2
XJ K2
USTOPD J2
OPEN/CLOSE STRING D COMPARISON LOOP
K2
ALL LOOPS CLOSED
YES
XL B2
MMPP
NO
1
1
FILLD B4
STRING D DELETED
NO
2
XJ J2
YES
GETD C4
INITIALIZE CCW WITH INPUT AREA ADDRESS
D4
READ PHYSICAL RCD FROM SYSLNK
E4
END OF FILE
YES
E5
MOVE LAST STRING INDICATOR TO OUTPUT AREA
NO
MODIFD F4
DSKINCR XN
UPDATE DISK ADDR FOR NEXT STRING D
XJ-K5
XJ-K2
*
2
COMPDC G4
D COMPARISON LOOP OPEN
NO
YES
H4
STRING D GT STRING C
YES
XJ B4
COMPCB
NO
XJ B2
COMPDB

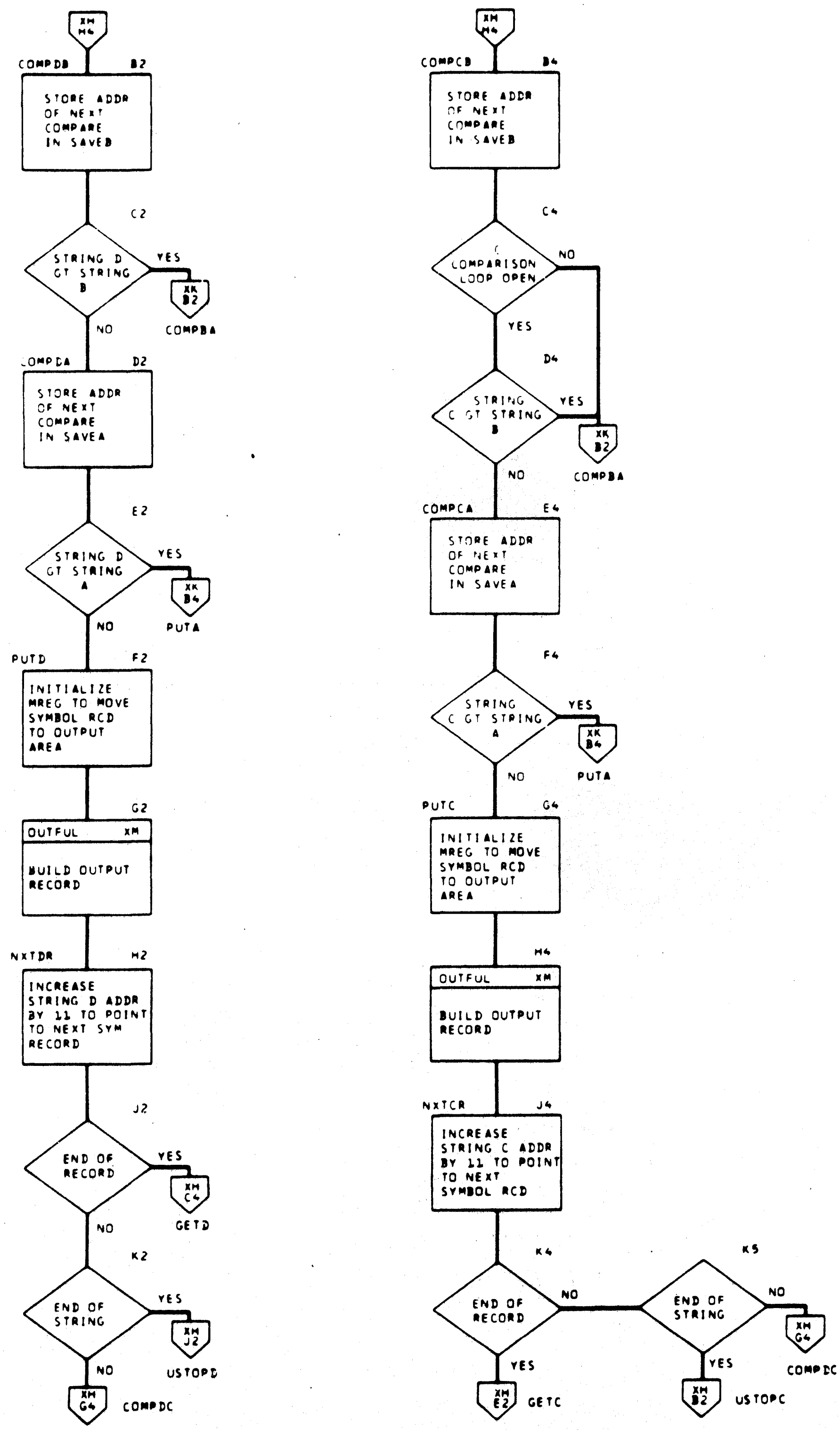
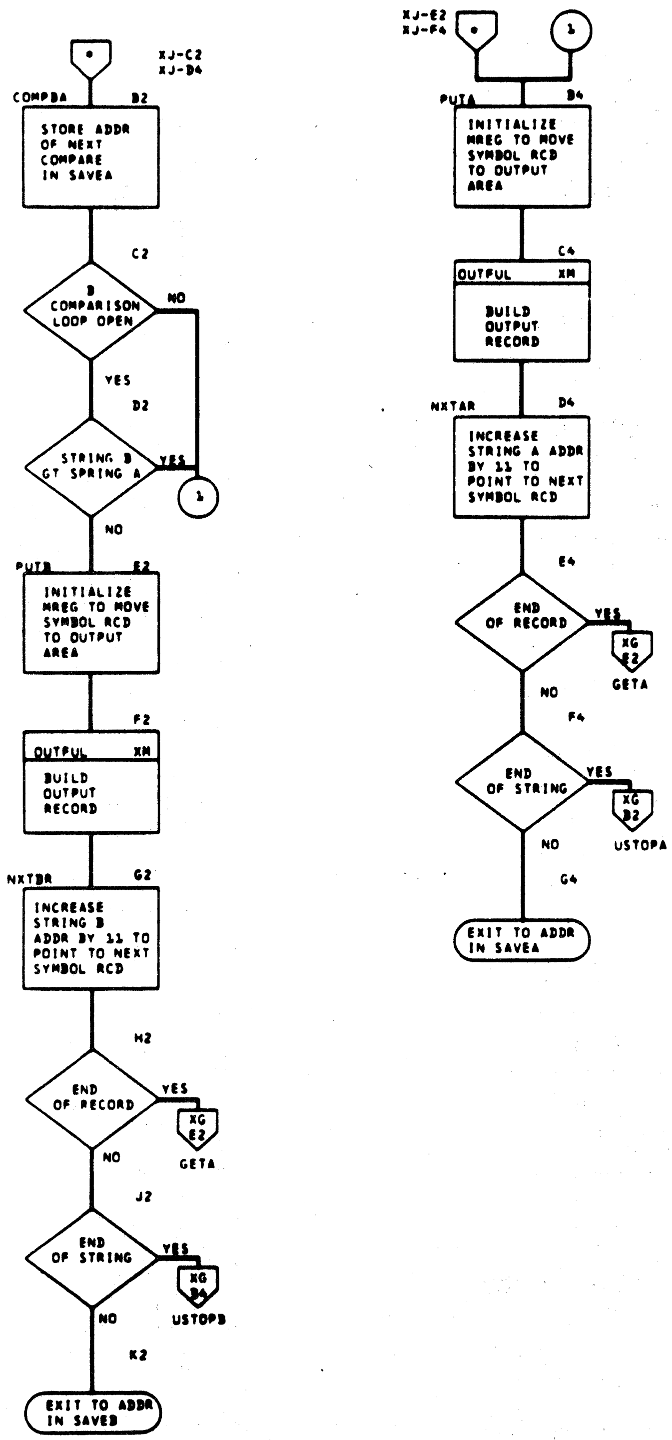

XM H4
COMPDB B2
STORE ADDR OF NEXT COMPARE IN SAVEB
C2
STRING D GT STRING B
YES
XK B2
COMPBA
NO
COMPDA D2
STORE ADDR OF NEXT COMPARE IN SAVEA
E2
STRING D GT STRING A
YES
XK B4
PUTA
NO
PUTD F2
INITIALIZE MREG TO MOVE SYMBOL RCD TO OUTPUT AREA
G2
OUTFUL XM
BUILD OUTPUT RECORD
NXTDR H2
INCREASE STRING D ADDR BY 11 TO POINT TO NEXT SYM RECORD
J2
END OF RECORD
YES
XM C4
GETD
NO
K2
END OF STRING
YES
XM J2
USTOPD
NO
XM G4
COMPDC
XM H4
COMPCB B4
STORE ADDR OF NEXT COMPARE IN SAVEB
C4
C COMPARISON LOOP OPEN
NO
YES
D4
STRING C GT STRING B
YES
XK B2
COMPBA
NO
COMPCA E4
STORE ADDR OF NEXT COMPARE IN SAVEA
F4
STRING C GT STRING A
YES
XK B4
PUTA
NO
PUTC G4
INITIALIZE MREG TO MOVE SYMBOL RCD TO OUTPUT AREA
H4
OUTFUL XM
BUILD OUTPUT RECORD
NXTCR J4
INCREASE STRING C ADDR BY 11 TO POINT TO NEXT SYMBOL RCD
K4
END OF RECORD
NO
YES
XM E2
GETC
K5
END OF STRING
NO
XM G4
COMPDC
YES
XM B2
USTOPC

Chart XK. Symbol Merge (ATLEFD2) (Refer to Chart 13) (Part 5 of 6)



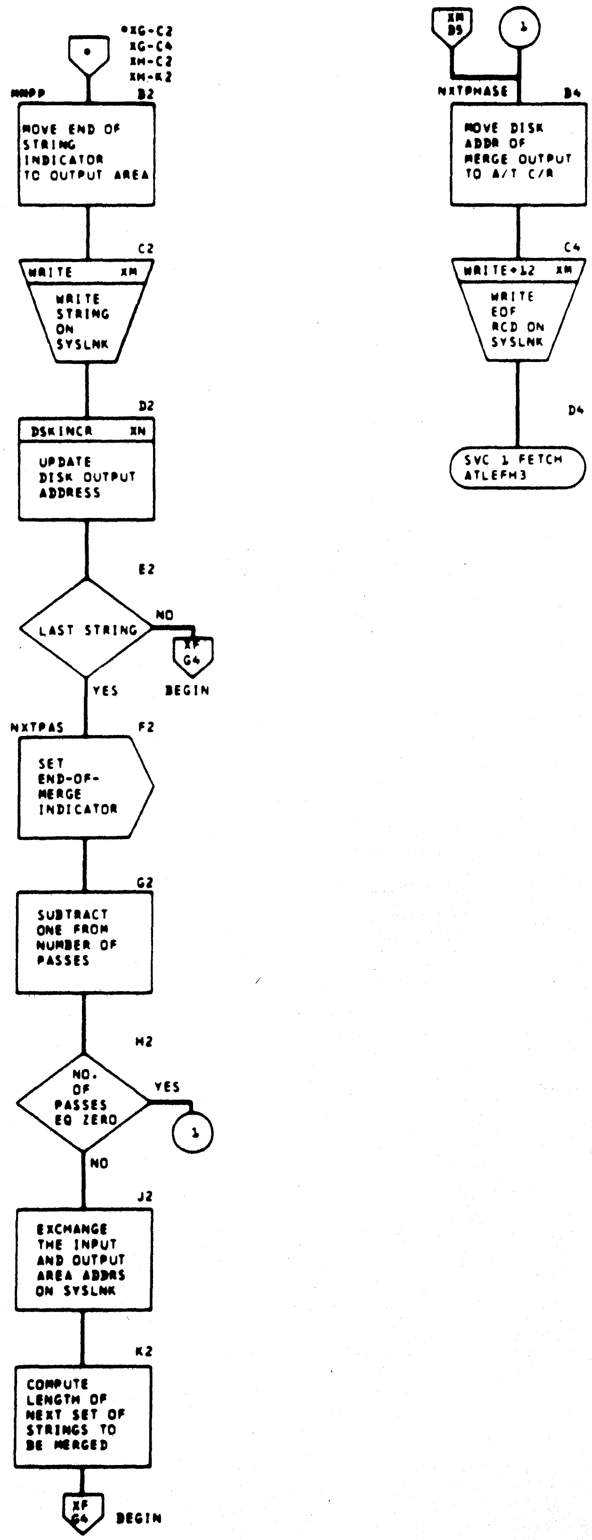
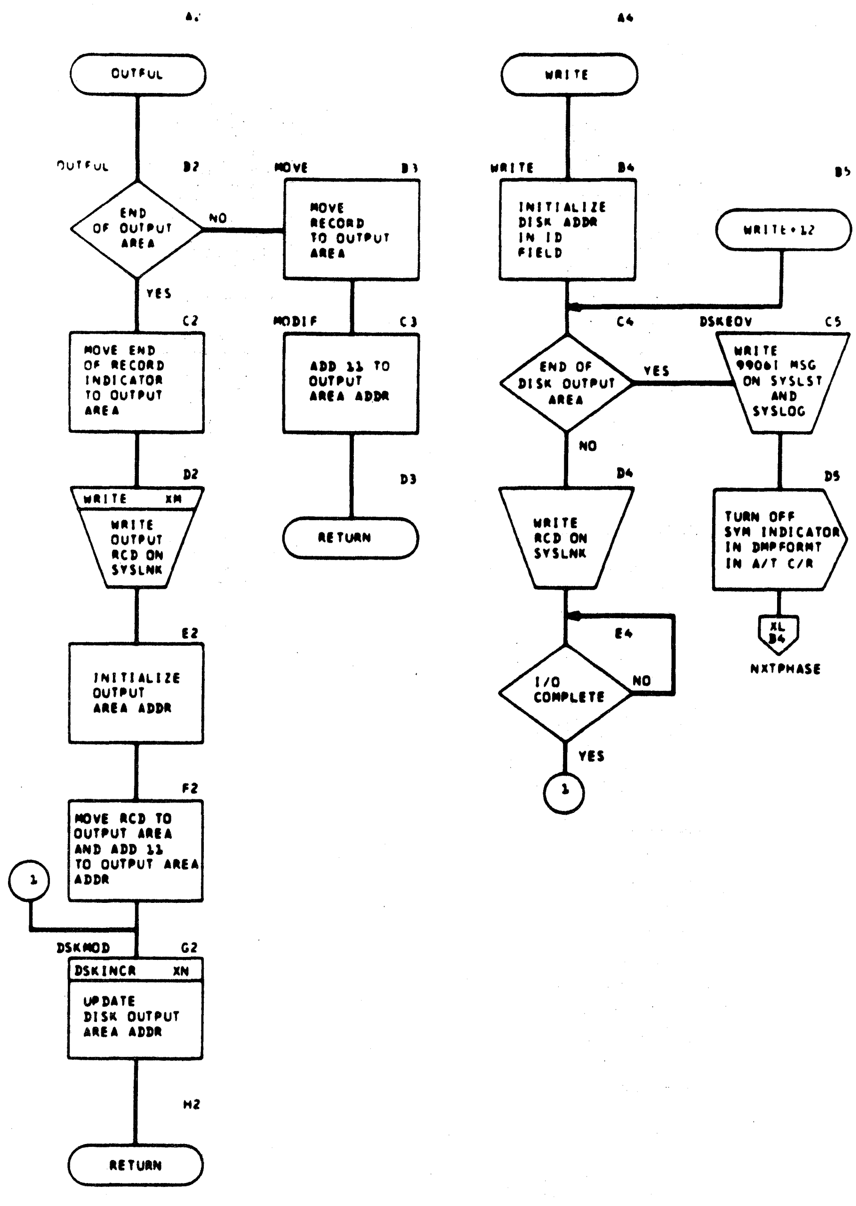

*XG-C2
XG-C4
XH-C2
XH-K2
MMPP B2
MOVE END OF STRING INDICATOR TO OUTPUT AREA
C2
WRITE XM
WRITE STRING ON SYSLNK
D2
DSKINCR XN
UPDATE DISK OUTPUT ADDRESS
E2
LAST STRING
NO
XF G4
BEGIN
YES
NXTPAS F2
SET END-OF-MERGE INDICATOR
G2
SUBTRACT ONE FROM NUMBER OF PASSES
H2
NO. OF PASSES EQ ZERO
YES
1
NO
J2
EXCHANGE THE INPUT AND OUTPUT AREA ADDRS ON SYSLNK
K2
COMPUTE LENGTH OF NEXT SET OF STRINGS TO BE MERGED
XF G4
BEGIN
XM B5
1
NXTPHASE B4
MOVE DISK ADDR OF MERGE OUTPUT TO A/T C/R
C4
WRITE+12 XM
WRITE EOF RCD ON SYSLNK
D4
SVC 1 FETCH ATLEFH3

Chart XM. Symbol Merge Subroutines (ATLEFD2) (Refer to Chart 13) (Part 1 of 2)



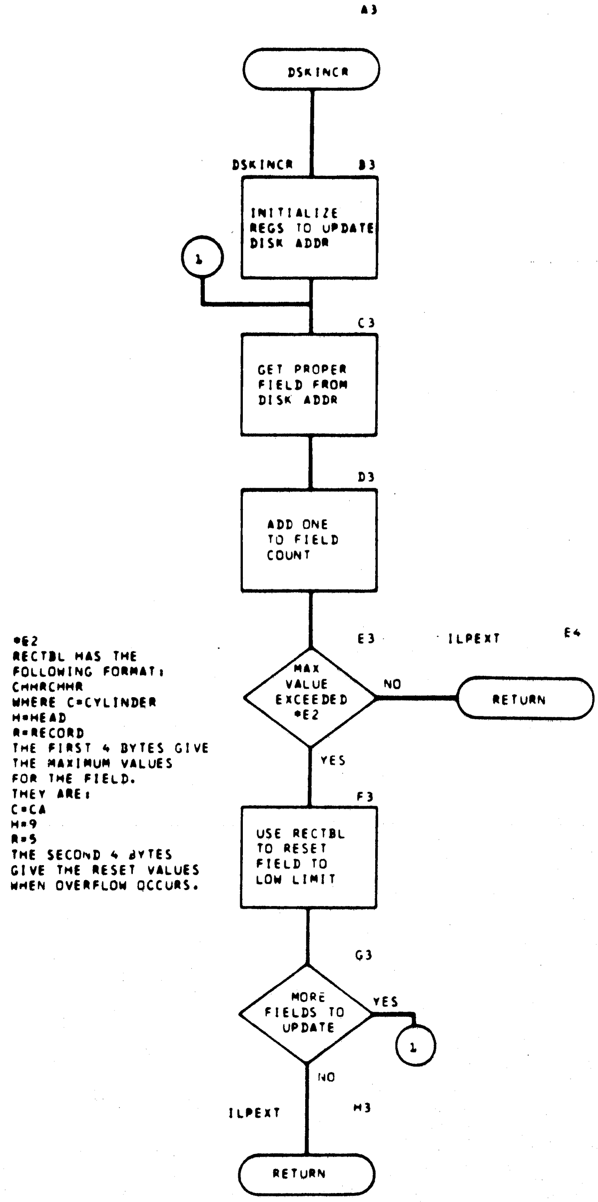

A3
DSKINCR
DSKINCR
B3
INITIALIZE REGS TO UPDATE DISK ADDR
1
C3
GET PROPER FIELD FROM DISK ADDR
D3
ADD ONE TO FIELD COUNT
E3
ILPEXT
E4
MAX VALUE EXCEEDED *E2
NO
RETURN
YES
*E2
RECTBL HAS THE FOLLOWING FORMAT:
CHHRCHHR
WHERE C=CYLINDER
H=HEAD
R=RECORD
THE FIRST 4 BYTES GIVE THE MAXIMUM VALUES FOR THE FIELD.
THEY ARE:
C=CA
H=9
R=5
THE SECOND 4 BYTES GIVE THE RESET VALUES WHEN OVERFLOW OCCURS.
F3
USE RECTBL TO RESET FIELD TO LOW LIMIT
G3
MORE FIELDS TO UPDATE
YES
1
NO
ILPEXT
H3
RETURN

Chart YA. EOJ Dump (ATLEFH3) (Refer to Chart 13) (Part 1 of 13)

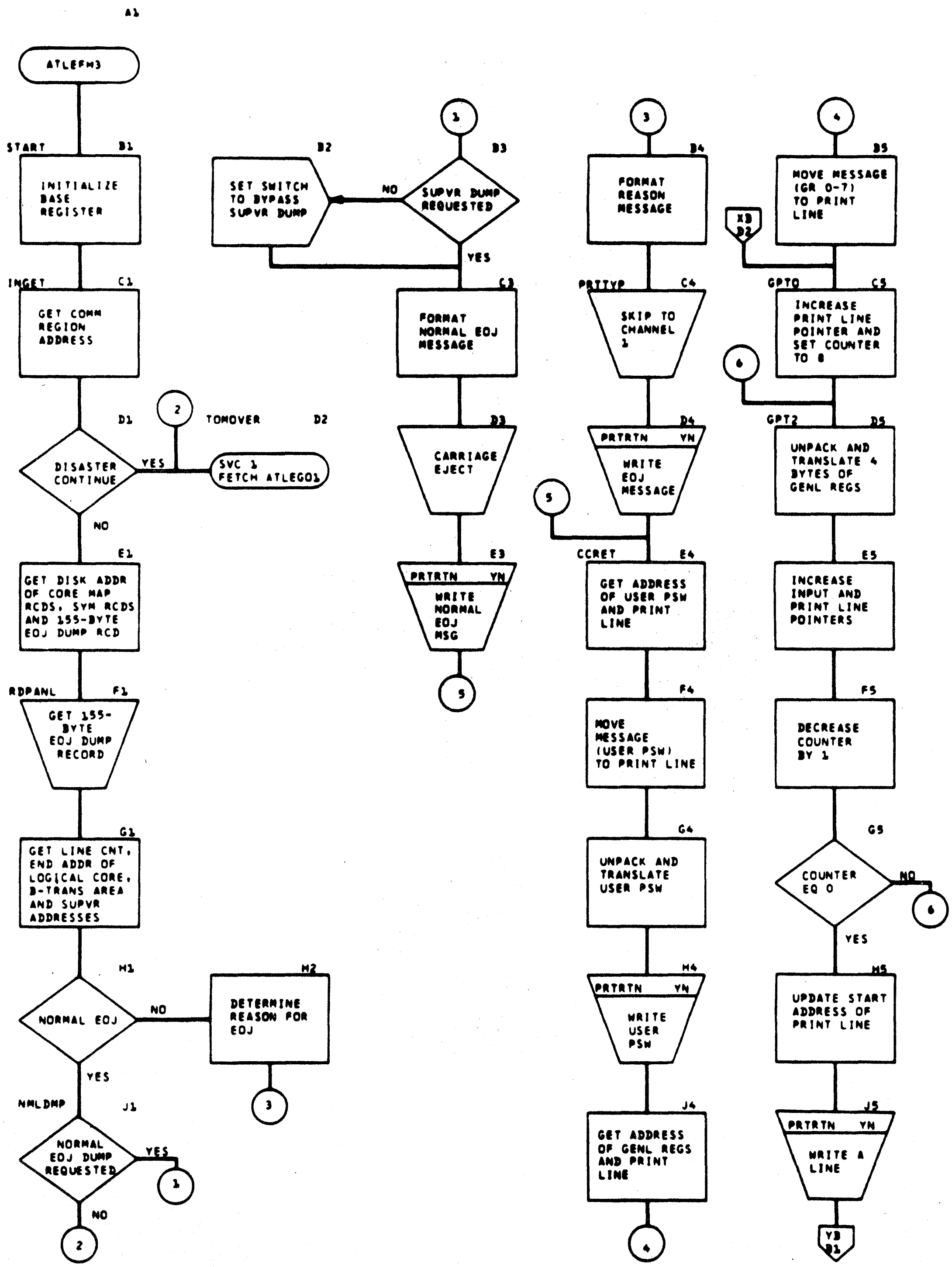


Chart YB. EOJ Dump (ATLEFH3) (Refer to Chart 13) (Part 2 of 13)

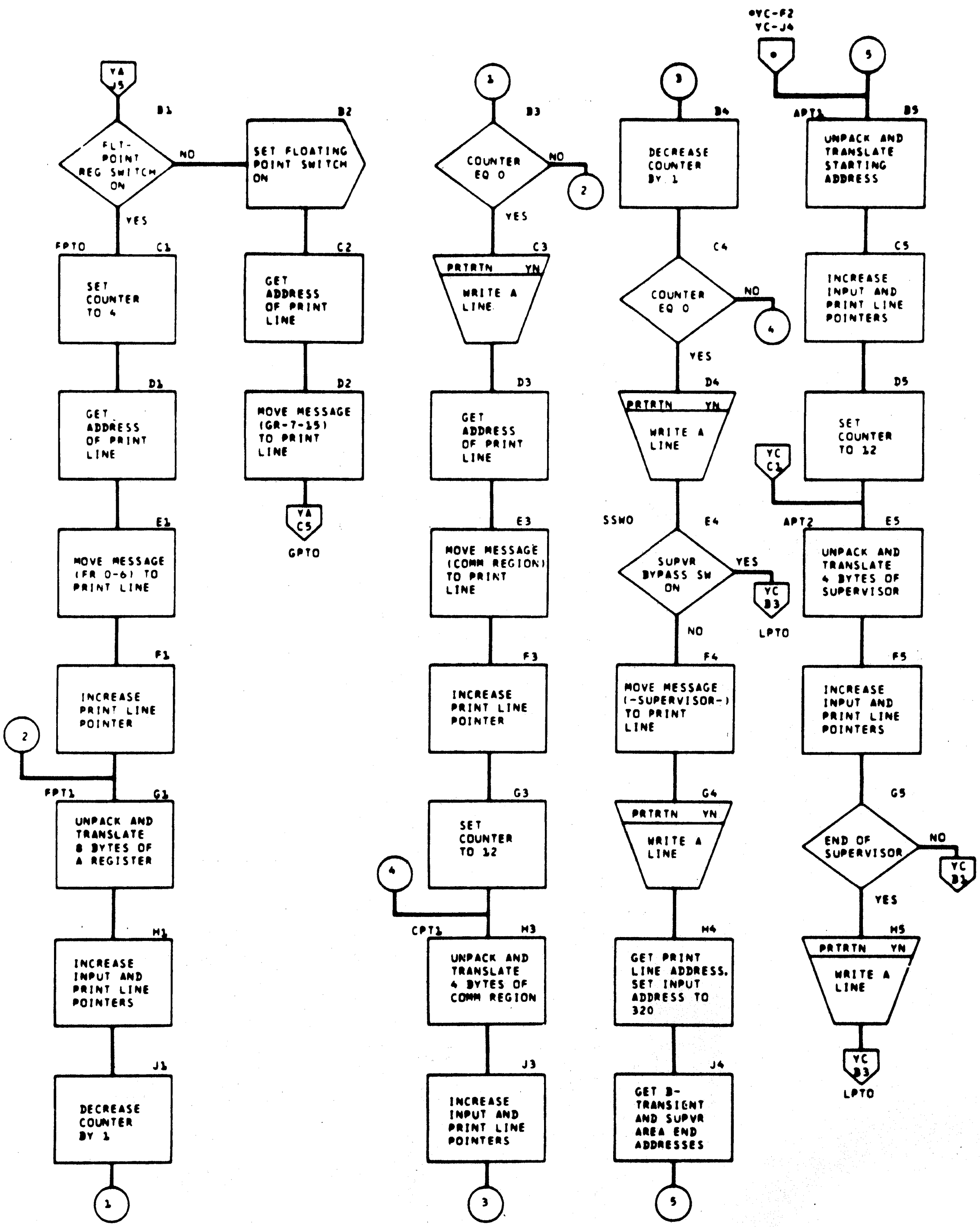
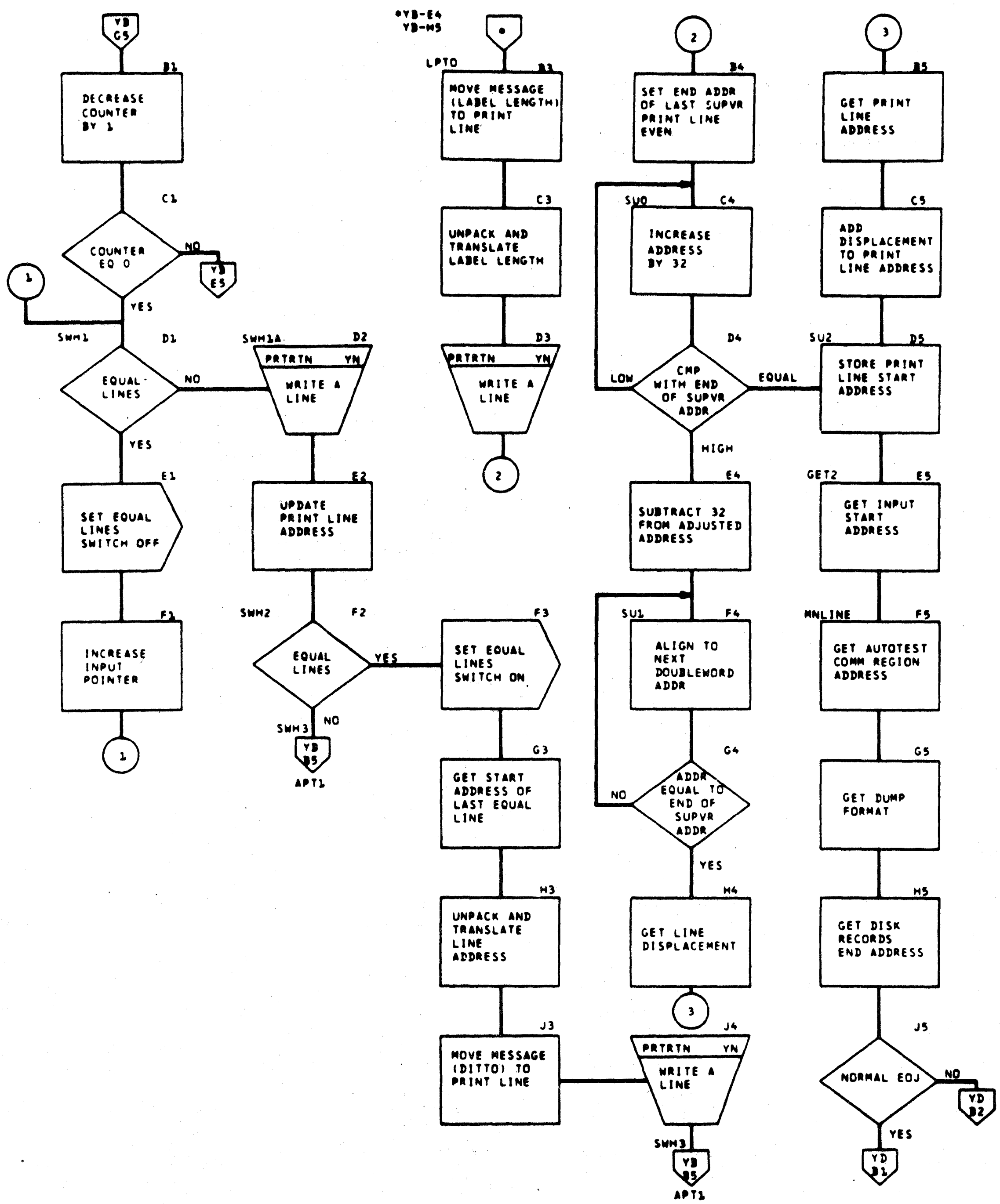


Chart YC. EOJ Dump (ATLEFH3) (Refer to Chart 13) (Part 3 of 13)



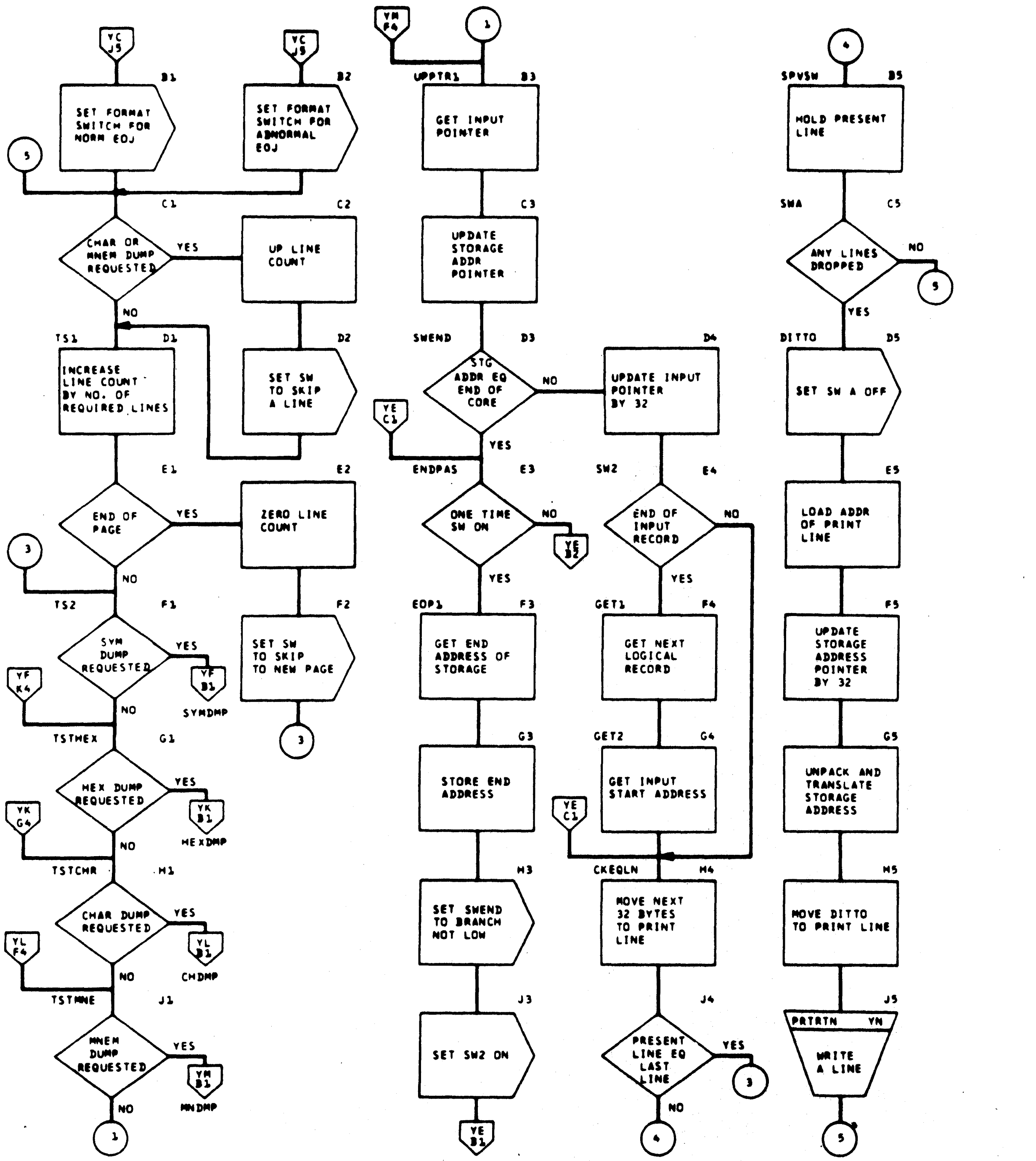

SET FORMAT SWITCH FOR NORM EOJ
SET FORMAT SWITCH FOR ABNORMAL EOJ
CHAR OR MNEM DUMP REQUESTED
UP LINE COUNT
INCREASE LINE COUNT BY NO. OF REQUIRED LINES
SET SW TO SKIP A LINE
END OF PAGE
ZERO LINE COUNT
SYM DUMP REQUESTED
SET SW TO SKIP TO NEW PAGE
SYMDMP
HEX DUMP REQUESTED
HEXDMP
CHAR DUMP REQUESTED
CHDMP
MNEM DUMP REQUESTED
MNDMP
UPPTR1
GET INPUT POINTER
UPDATE STORAGE ADDR POINTER
SWEND
STG ADDR EQ END OF CORE
UPDATE INPUT POINTER BY 32
ENDPAS
ONE TIME SW ON
SW2
END OF INPUT RECORD
EOP1
GET END ADDRESS OF STORAGE
GET1
GET NEXT LOGICAL RECORD
STORE END ADDRESS
GET2
GET INPUT START ADDRESS
SET SWEND TO BRANCH NOT LOW
CKEQLN
MOVE NEXT 32 BYTES TO PRINT LINE
SET SW2 ON
PRESENT LINE EQ LAST LINE
SPVSW
HOLD PRESENT LINE
SWA
ANY LINES DROPPED
DITTO
SET SW A OFF
LOAD ADDR OF PRINT LINE
UPDATE STORAGE ADDRESS POINTER BY 32
UNPACK AND TRANSLATE STORAGE ADDRESS
MOVE DITTO TO PRINT LINE
PRTRTN
WRITE A LINE
TS1
TS2
TSTHEX
TSTCHR
TSTMNE
YES
NO

Chart YE. EOJ Dump (ATLEFH3) (Refer to Chart 13) (Part 5 of 13)

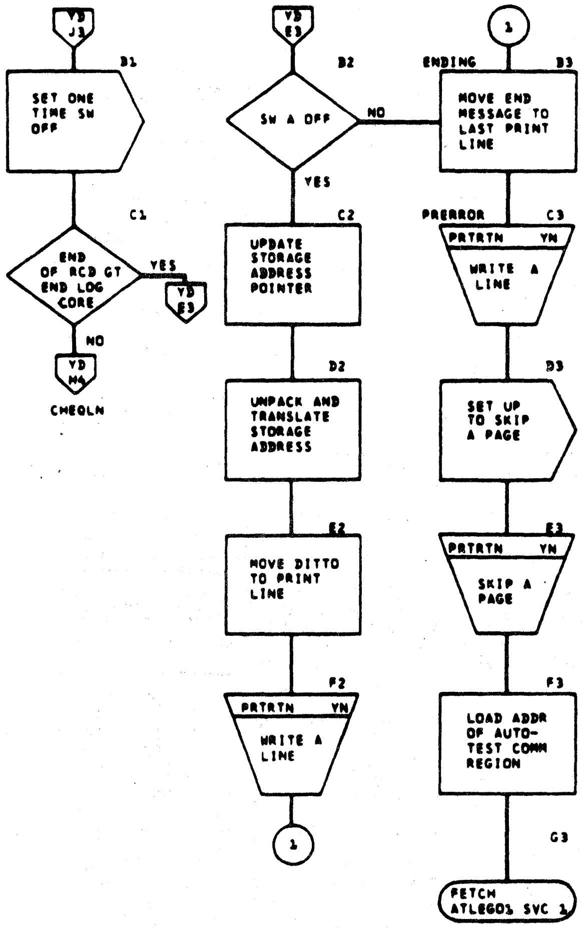
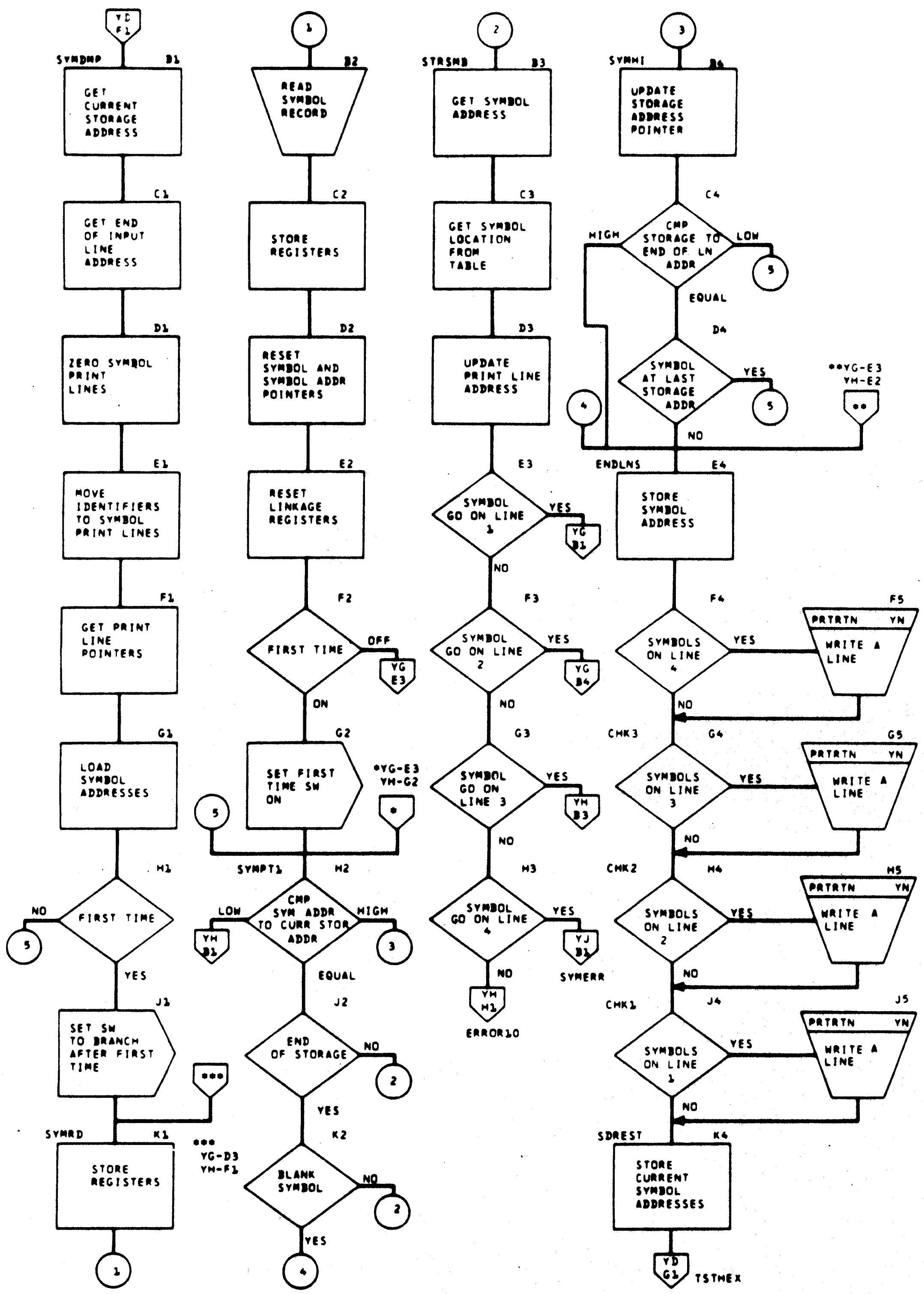
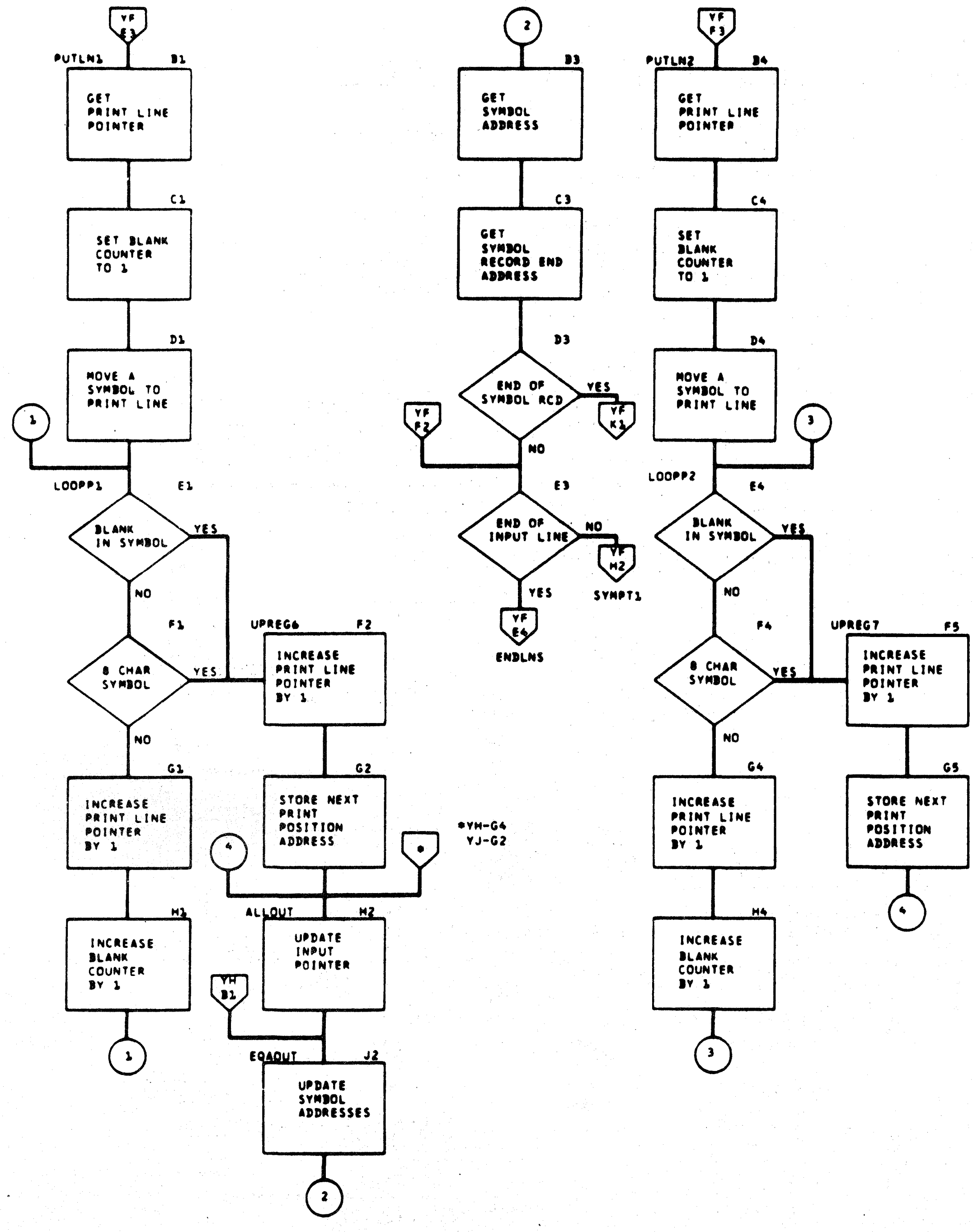
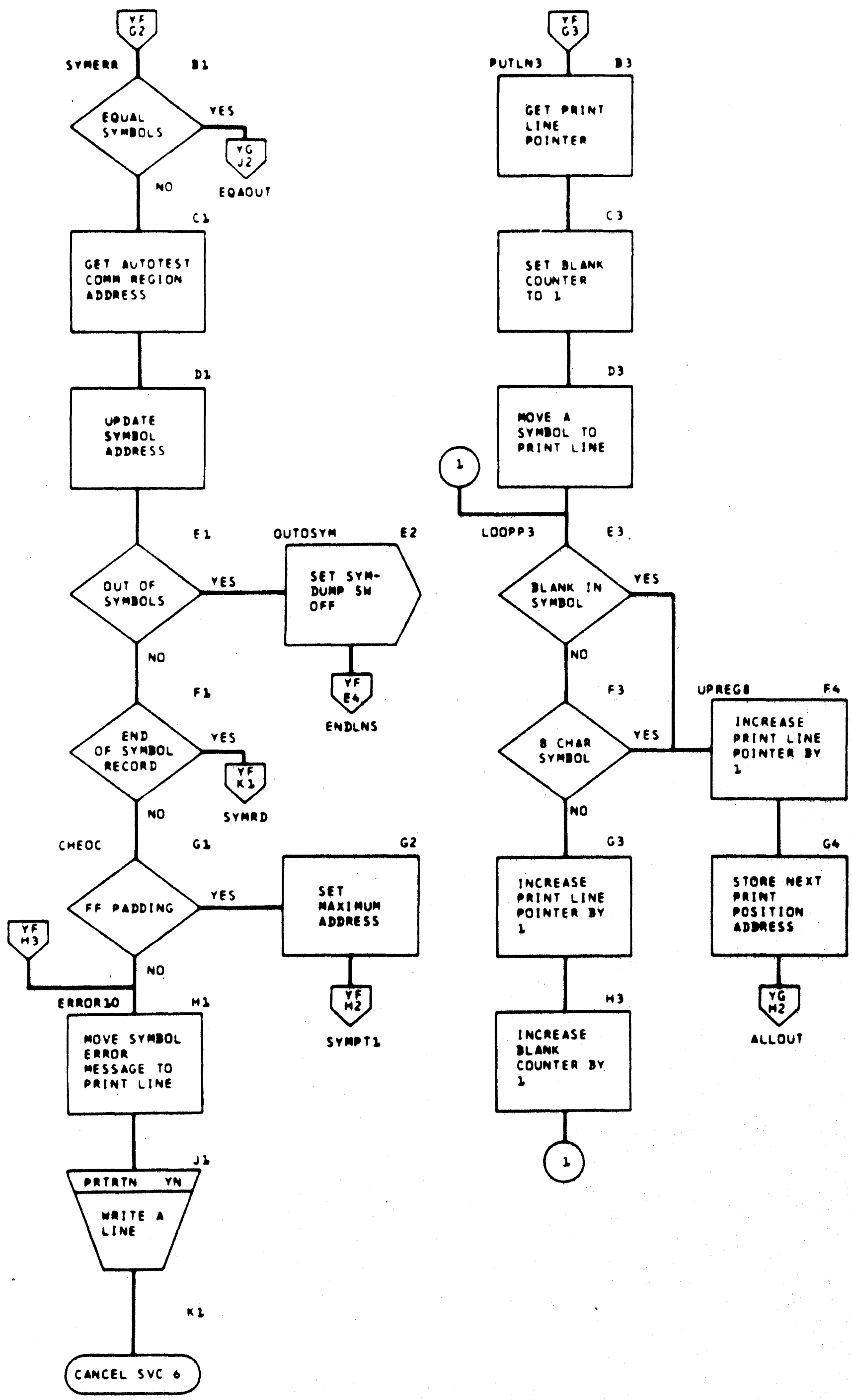


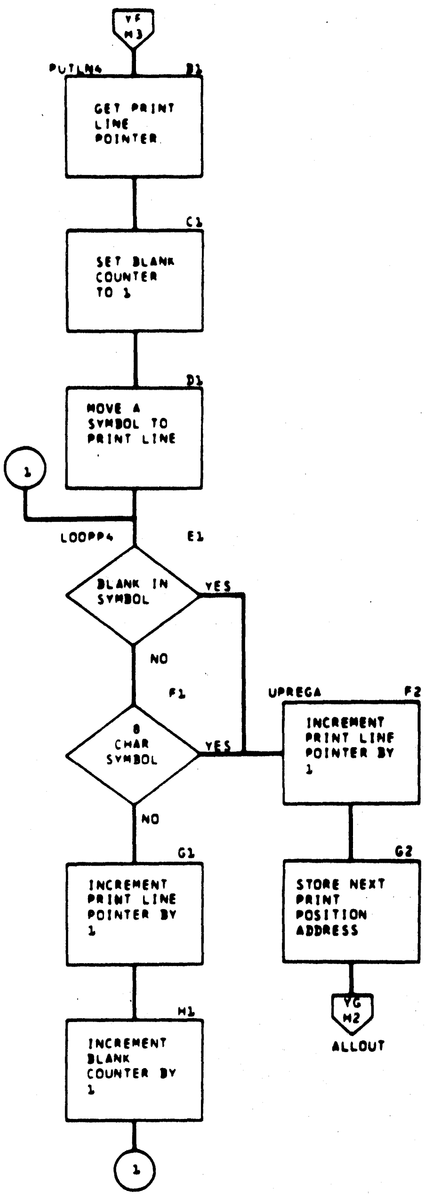
Chart YF. EOJ Dump (ATLEFH3) (Refer to Chart 13) (Part 6 of 13)

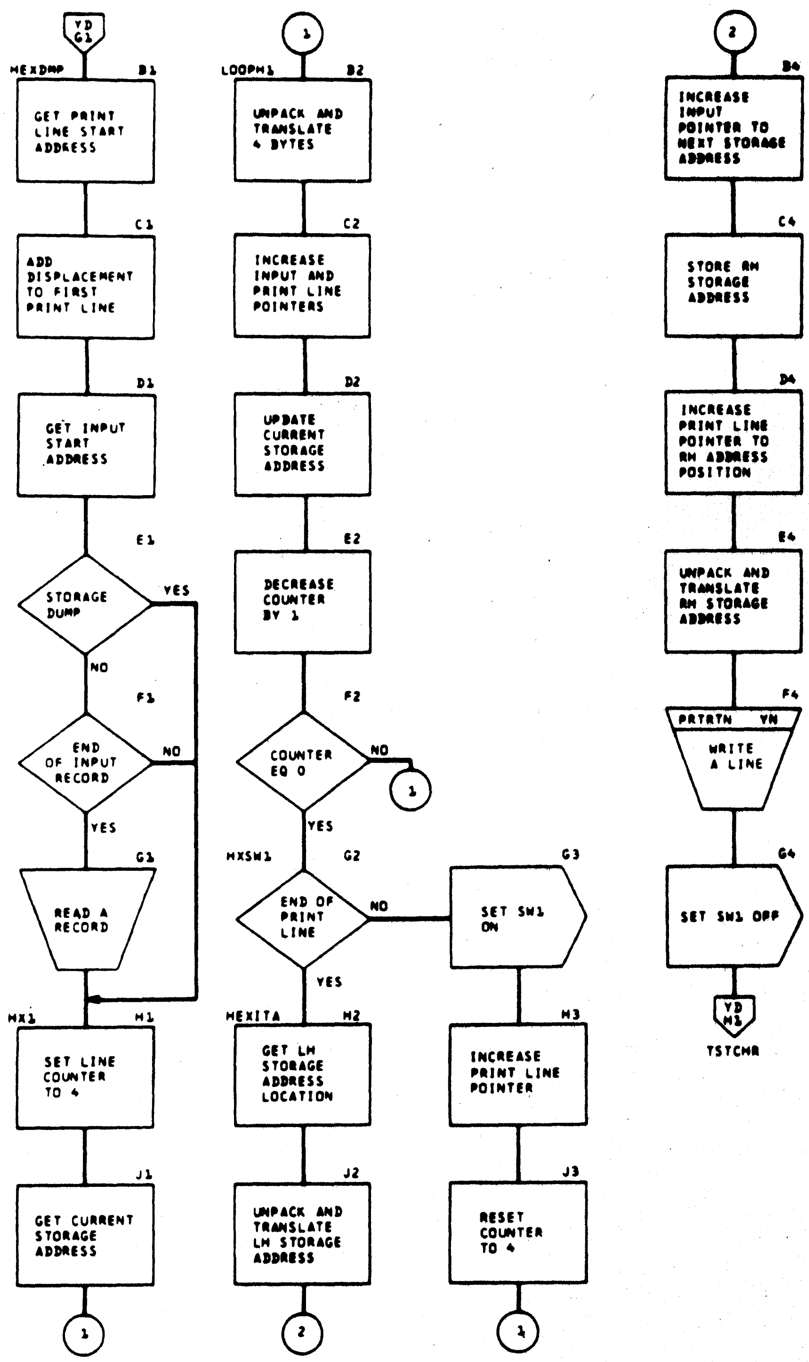




YF E3
PUTLN1 B1
GET PRINT LINE POINTER
C1
SET BLANK COUNTER TO 1
D1
MOVE A SYMBOL TO PRINT LINE
LOOPP1 E1
BLANK IN SYMBOL
YES
NO
F1
8 CHAR SYMBOL
UPREG6 F2
INCREASE PRINT LINE POINTER BY 1
G1
INCREASE PRINT LINE POINTER BY 1
G2
STORE NEXT PRINT POSITION ADDRESS
*YH-G4 YJ-G2
H1
INCREASE BLANK COUNTER BY 1
ALLOUT H2
UPDATE INPUT POINTER
YH B1
EOAOUT J2
UPDATE SYMBOL ADDRESSES
B3
GET SYMBOL ADDRESS
C3
GET SYMBOL RECORD END ADDRESS
D3
END OF SYMBOL RCD
YF K1
YF F2
E3
END OF INPUT LINE
YF H2
SYMPT1
YF E4
ENDLNS
YF F3
PUTLN2 B4
GET PRINT LINE POINTER
C4
SET BLANK COUNTER TO 1
D4
MOVE A SYMBOL TO PRINT LINE
LOOPP2 E4
BLANK IN SYMBOL
F4
8 CHAR SYMBOL
UPREG7 F5
INCREASE PRINT LINE POINTER BY 1
G4
INCREASE PRINT LINE POINTER BY 1
G5
STORE NEXT PRINT POSITION ADDRESS
H4
INCREASE BLANK COUNTER BY 1


YF G2
SYMERR B1
EQUAL SYMBOLS
YES
NO
YG J2
EQAOUT
C1
GET AUTOTEST COMM REGION ADDRESS
D1
UPDATE SYMBOL ADDRESS
E1
OUT OF SYMBOLS
OUTOSYM E2
SET SYM-DUMP SW OFF
YF E4
ENDLNS
F1
END OF SYMBOL RECORD
YF K1
SYMRD
CHEOC G1
FF PADDING
G2
SET MAXIMUM ADDRESS
YF H2
SYMPT1
YF H3
ERROR10 H1
MOVE SYMBOL ERROR MESSAGE TO PRINT LINE
J1
PRTRTN YN
WRITE A LINE
K1
CANCEL SVC 6
YF G3
PUTLN3 B3
GET PRINT LINE POINTER
C3
SET BLANK COUNTER TO 1
D3
MOVE A SYMBOL TO PRINT LINE
1
LOOPP3 E3
BLANK IN SYMBOL
F3
8 CHAR SYMBOL
UPREG8 F4
INCREASE PRINT LINE POINTER BY 1
G3
INCREASE PRINT LINE POINTER BY 1
G4
STORE NEXT PRINT POSITION ADDRESS
YG H2
ALLOUT
H3
INCREASE BLANK COUNTER BY 1
1


YF H3
PUTLN4 B1
GET PRINT LINE POINTER
C1
SET BLANK COUNTER TO 1
D1
MOVE A SYMBOL TO PRINT LINE
1
LOOPP4 E1
BLANK IN SYMBOL
YES
NO
F1
8 CHAR SYMBOL
YES
NO
UPREGA F2
INCREMENT PRINT LINE POINTER BY 1
G2
STORE NEXT PRINT POSITION ADDRESS
YG H2
ALLOUT
G1
INCREMENT PRINT LINE POINTER BY 1
H1
INCREMENT BLANK COUNTER BY 1
1


YD G1
HEXDMP B1
GET PRINT LINE START ADDRESS
C1
ADD DISPLACEMENT TO FIRST PRINT LINE
D1
GET INPUT START ADDRESS
E1
STORAGE DUMP
YES
NO
F1
END OF INPUT RECORD
NO
YES
G1
READ A RECORD
HX1 H1
SET LINE COUNTER TO 4
J1
GET CURRENT STORAGE ADDRESS
1
LOOPH1 B2
UNPACK AND TRANSLATE 4 BYTES
C2
INCREASE INPUT AND PRINT LINE POINTERS
D2
UPDATE CURRENT STORAGE ADDRESS
E2
DECREASE COUNTER BY 1
F2
COUNTER EQ 0
NO
YES
HXSW1 G2
END OF PRINT LINE
NO
YES
HEXITA H2
GET LH STORAGE ADDRESS LOCATION
J2
UNPACK AND TRANSLATE LH STORAGE ADDRESS
2
G3
SET SW1 ON
H3
INCREASE PRINT LINE POINTER
J3
RESET COUNTER TO 4
B4
INCREASE INPUT POINTER TO NEXT STORAGE ADDRESS
C4
STORE RH STORAGE ADDRESS
D4
INCREASE PRINT LINE POINTER TO RH ADDRESS POSITION
E4
UNPACK AND TRANSLATE RH STORAGE ADDRESS
F4
PRTRTN YN
WRITE A LINE
G4
SET SW1 OFF
YD H1
TSTCHR

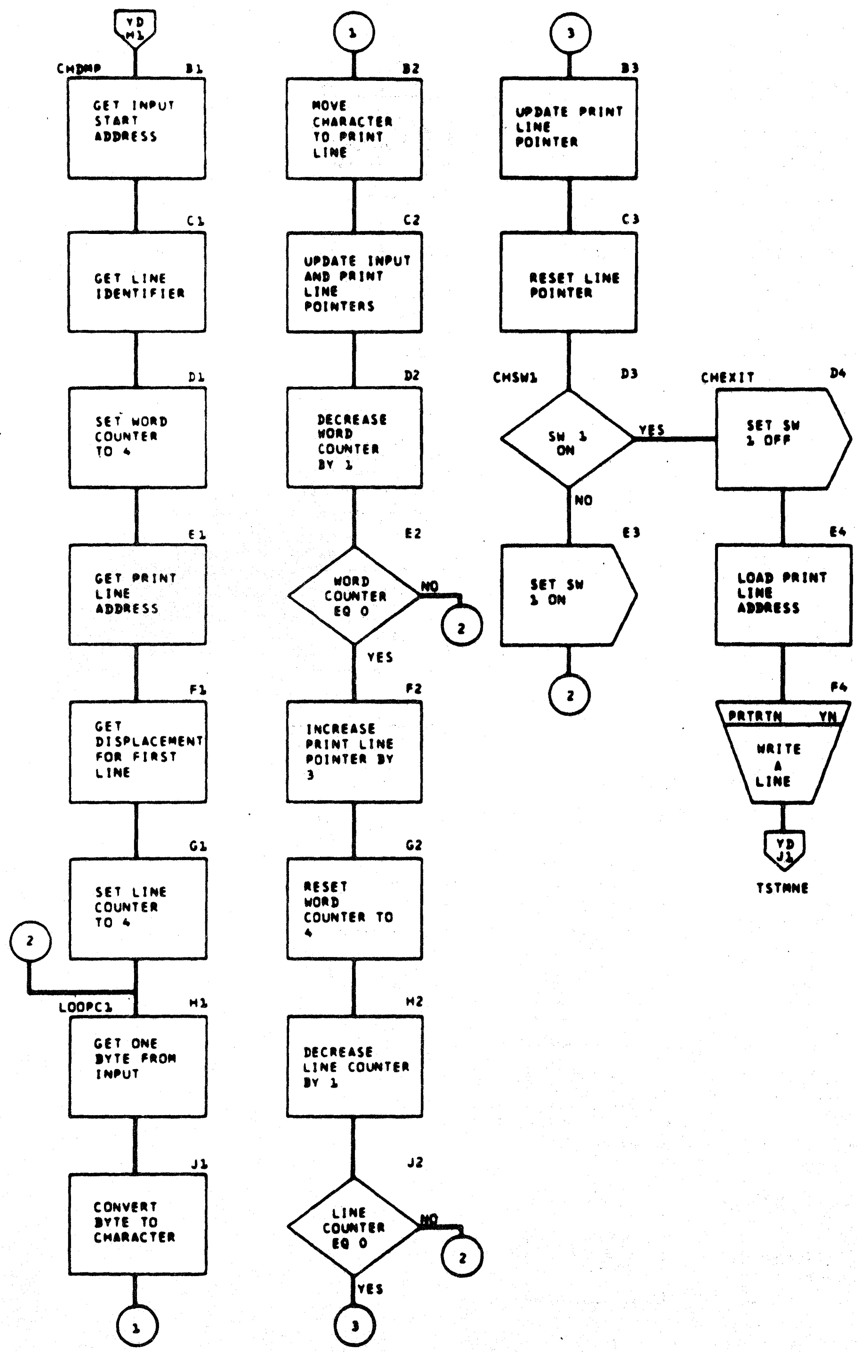

YD H1
CHDMP
B1
GET INPUT START ADDRESS
C1
GET LINE IDENTIFIER
D1
SET WORD COUNTER TO 4
E1
GET PRINT LINE ADDRESS
F1
GET DISPLACEMENT FOR FIRST LINE
G1
SET LINE COUNTER TO 4
2
LOOPC1
H1
GET ONE BYTE FROM INPUT
J1
CONVERT BYTE TO CHARACTER
1
1
B2
MOVE CHARACTER TO PRINT LINE
C2
UPDATE INPUT AND PRINT LINE POINTERS
D2
DECREASE WORD COUNTER BY 1
E2
WORD COUNTER EQ 0
NO
2
YES
F2
INCREASE PRINT LINE POINTER BY 3
G2
RESET WORD COUNTER TO 4
H2
DECREASE LINE COUNTER BY 1
J2
LINE COUNTER EQ 0
NO
2
YES
3
3
B3
UPDATE PRINT LINE POINTER
C3
RESET LINE POINTER
CHSW1
D3
SW 1 ON
YES
NO
E3
SET SW 1 ON
2
CHEXIT
D4
SET SW 1 OFF
E4
LOAD PRINT LINE ADDRESS
F4
PRTRTN
YN
WRITE A LINE
YD J1
TSTMNE

Chart YM. EOJ Dump (ATLEFH3) (Refer to Chart 13) (Part 12 of 13)

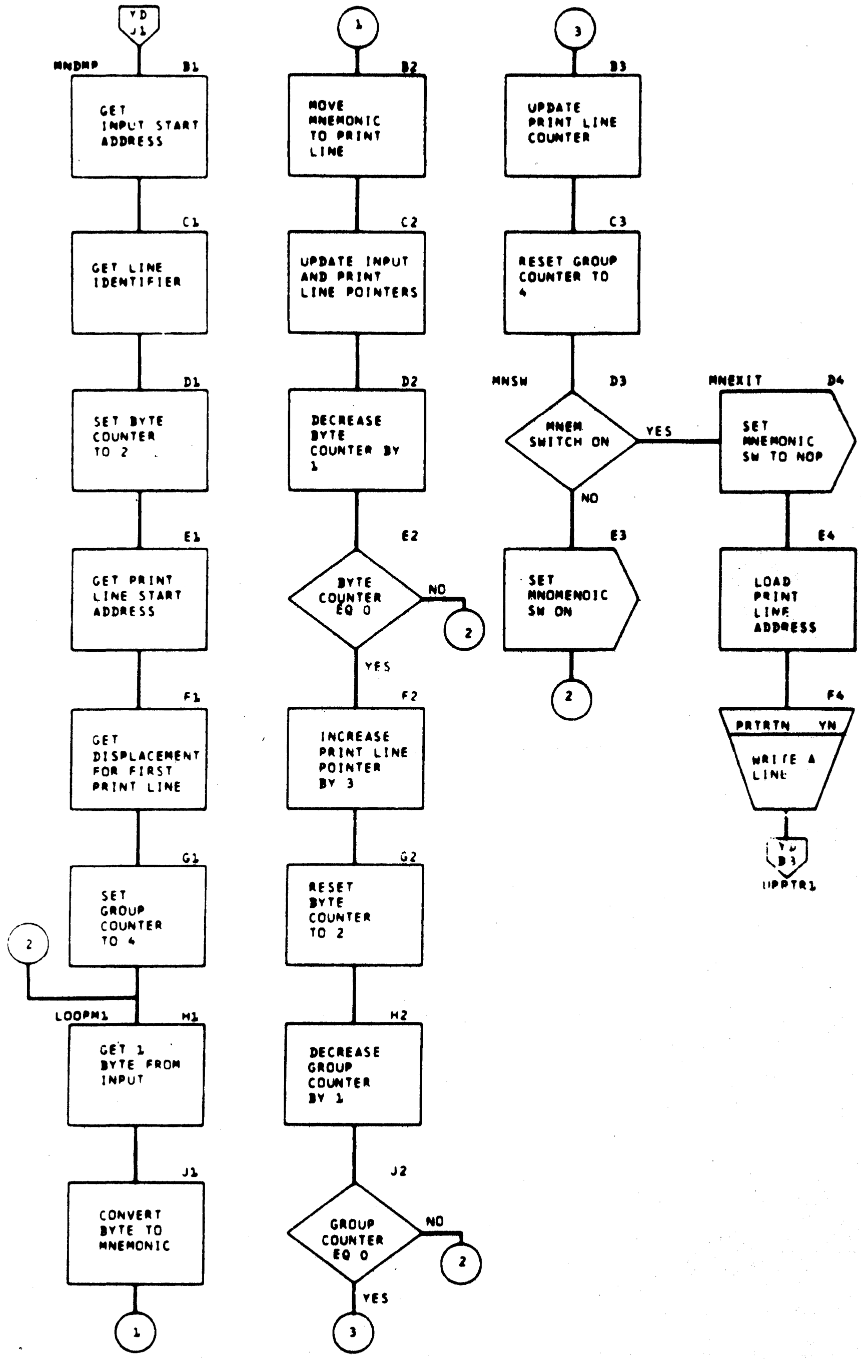


Chart YN. EOJ Dump (ATLEFH3) (Refer to Chart 13) (Part 13 of 13)

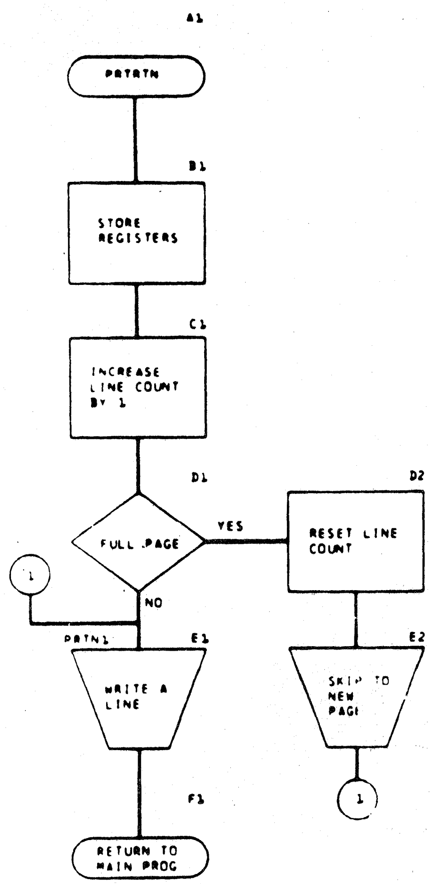


Chart YP. Autotest End Routine (ATLEG01) (Refer to Chart 13) (Part 1 of 2)

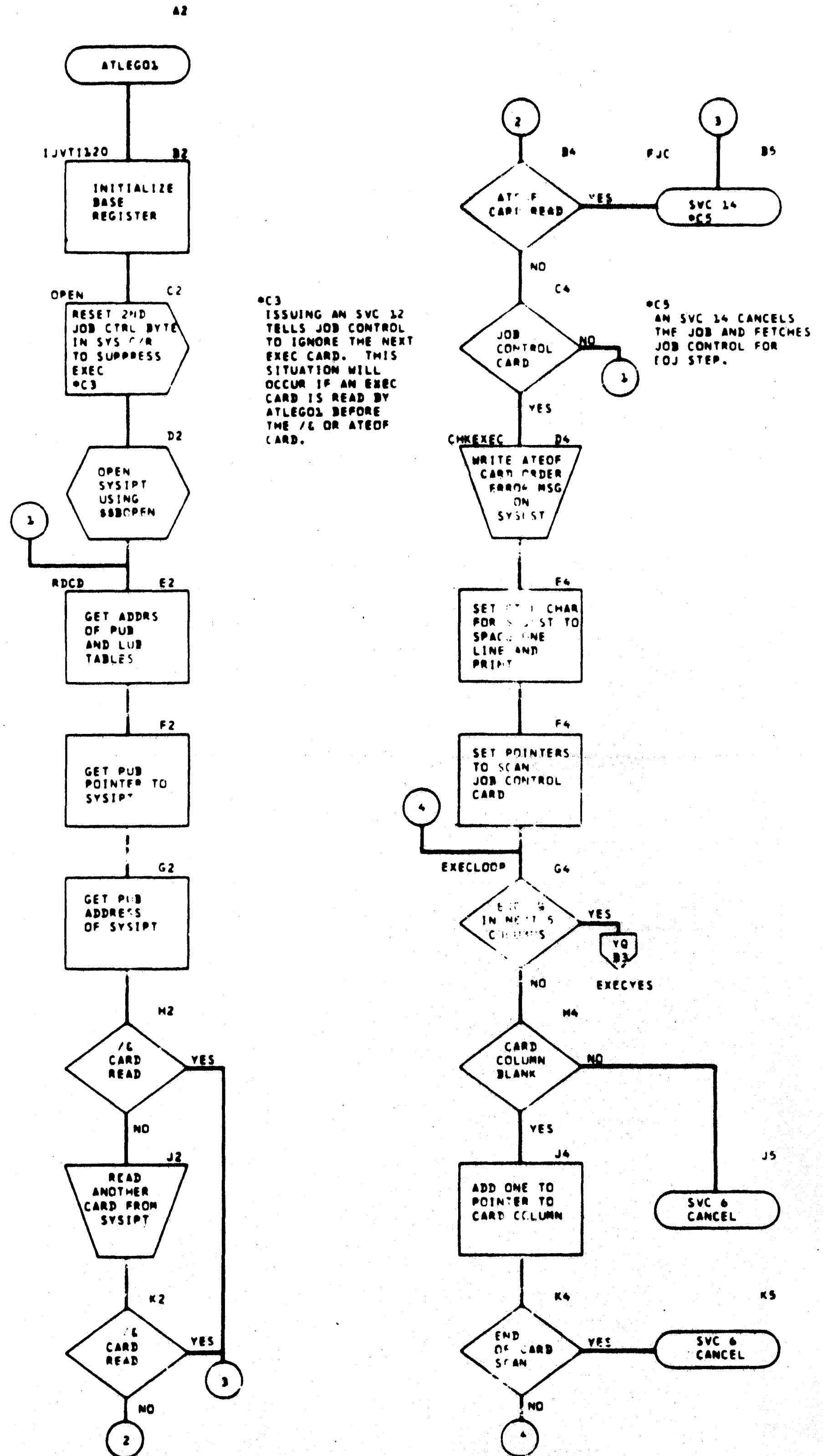


Chart YQ. Autotest End Routine (ATLEG01) (Refer to Chart 13) (Part 2 of 2)

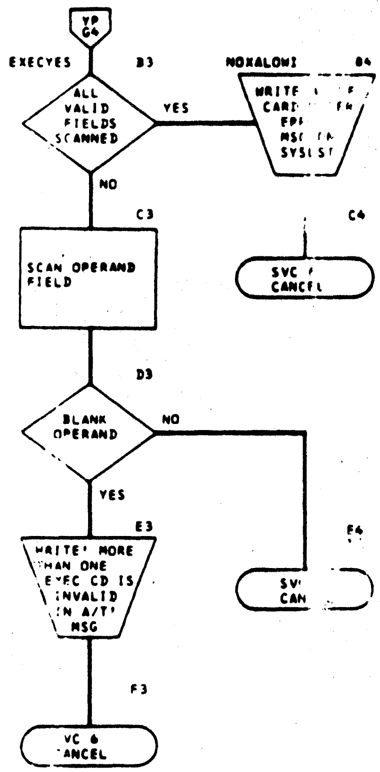


Chart ZA. Disaster Continue (ATLECONT) (Refer to Chart 00)

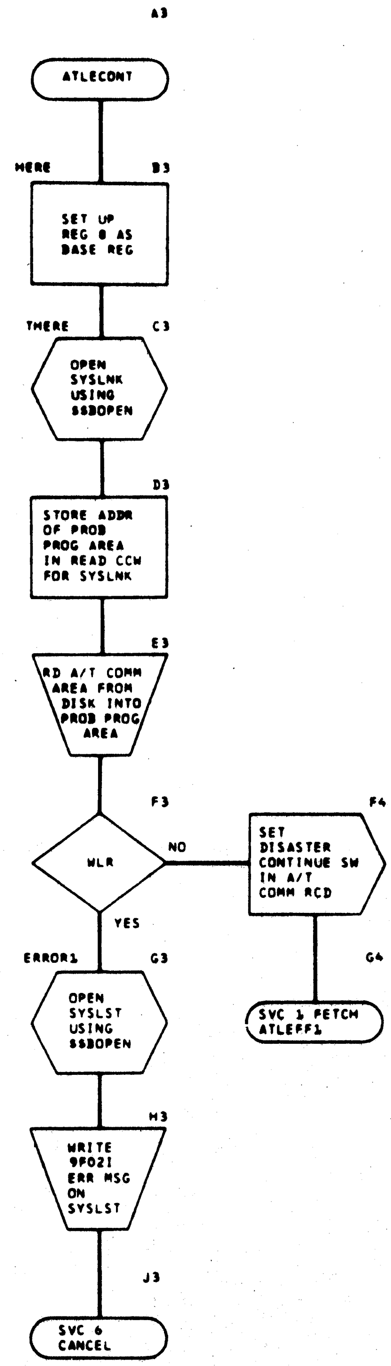
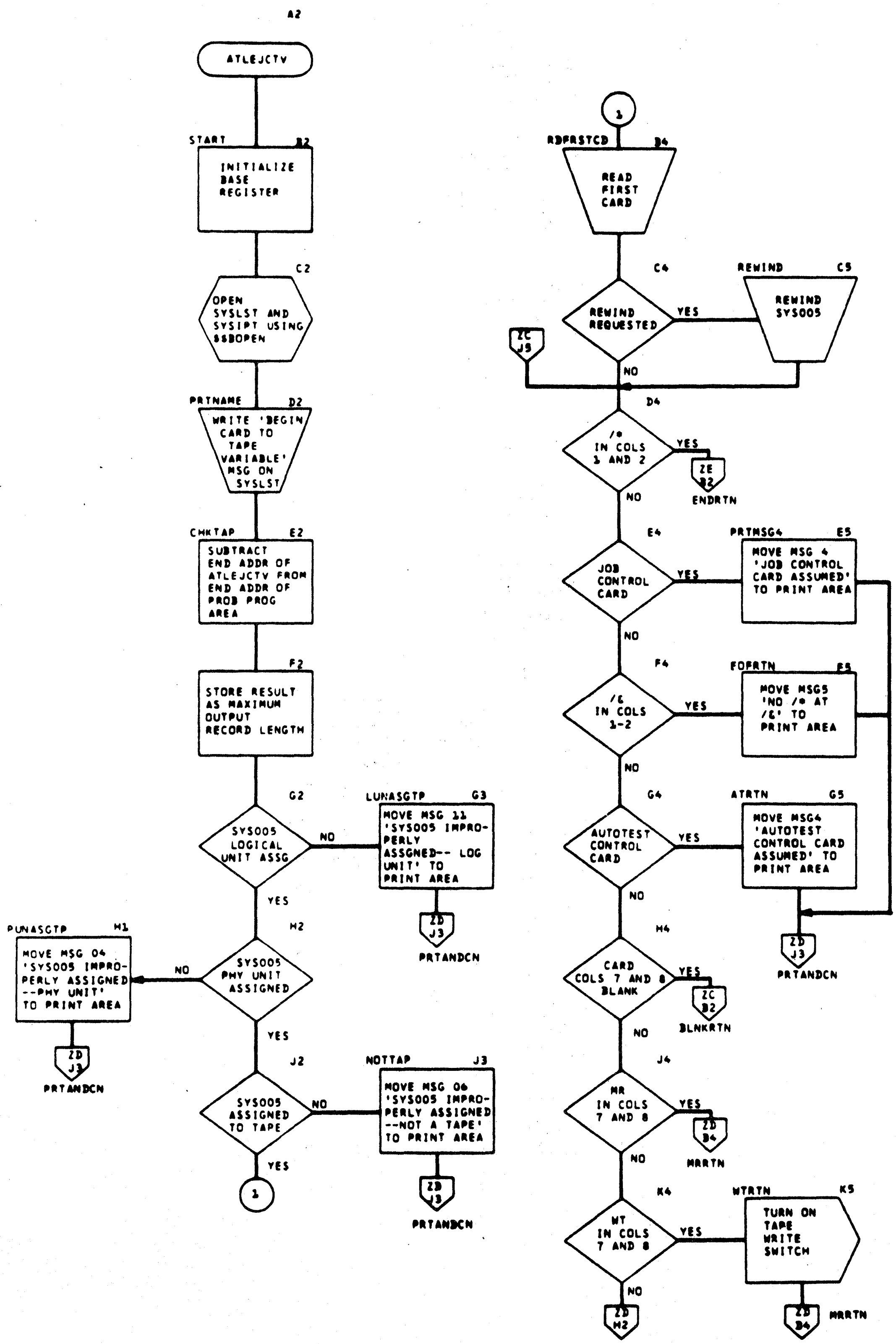
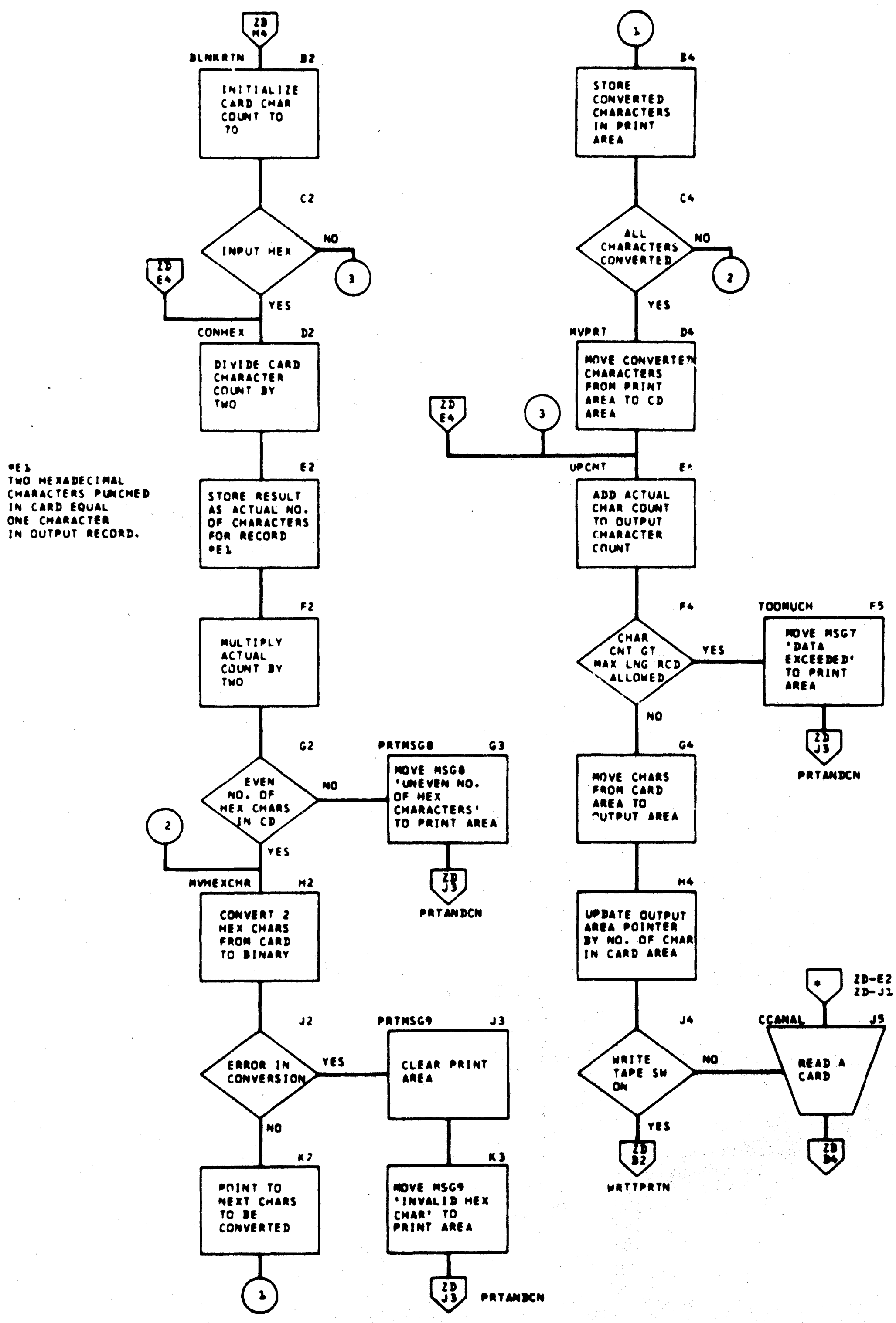
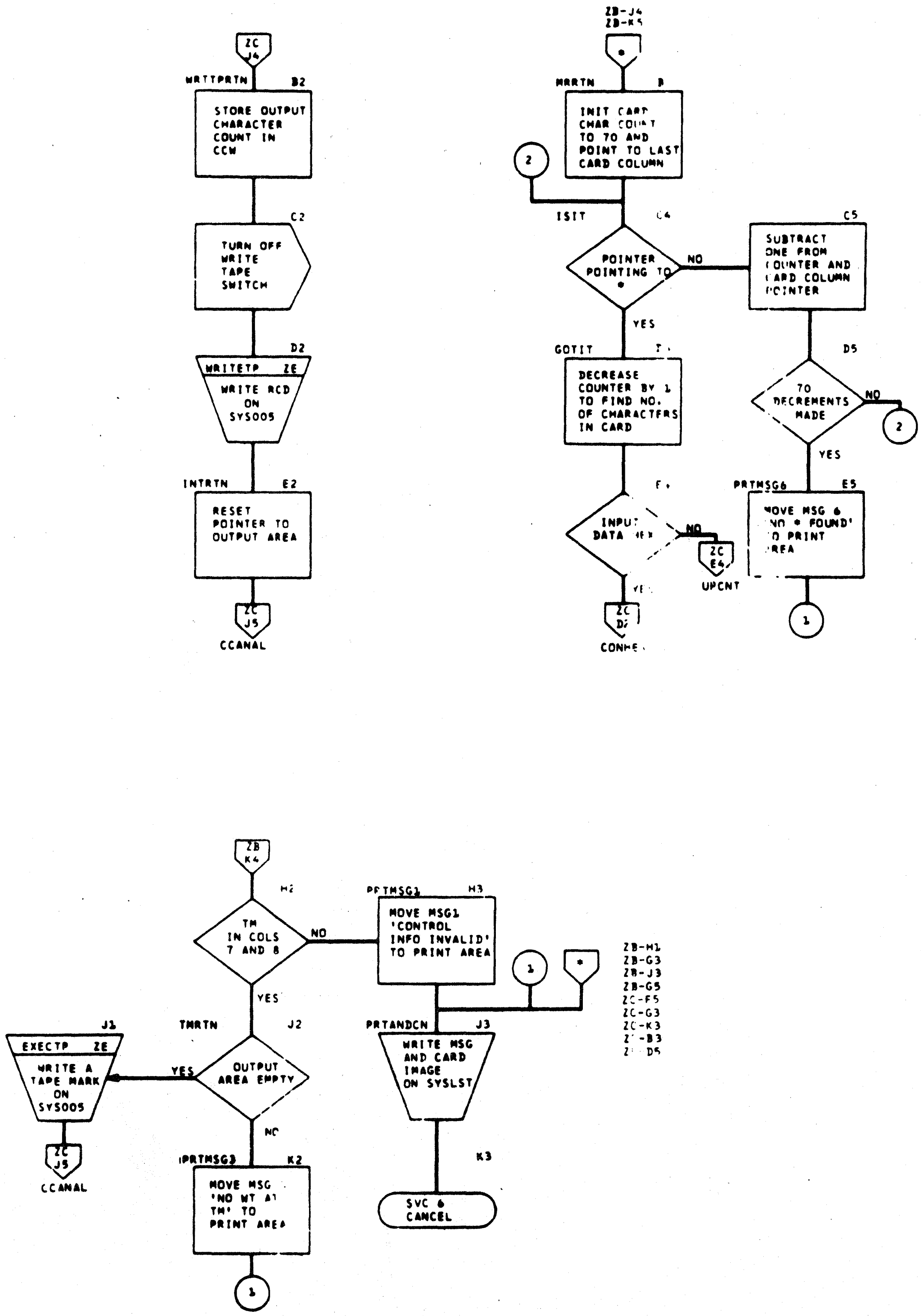
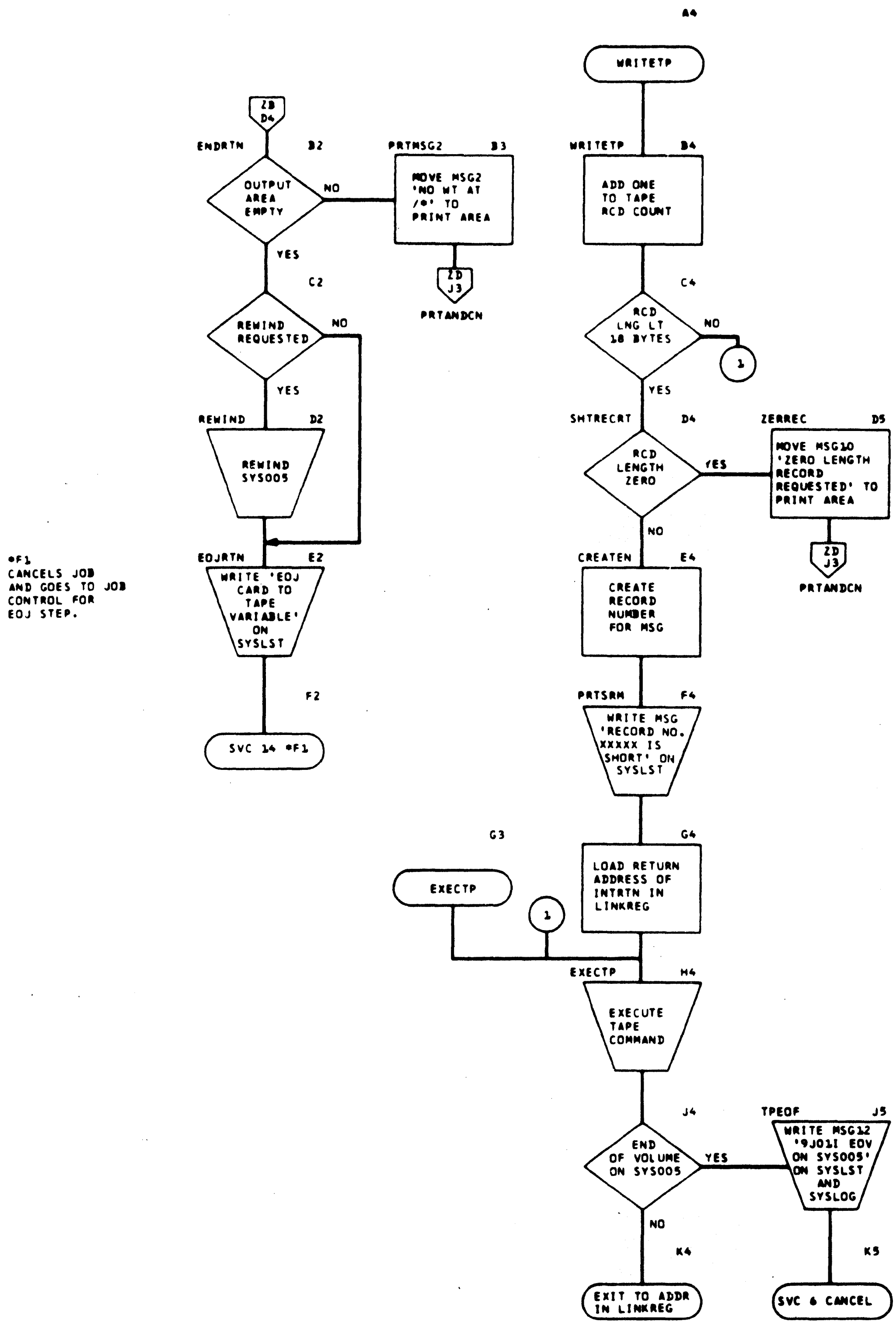


Chart ZB. Card to Tape Variable Utility (ATLEJCTV) (Refer to Chart 00) (Part 1 of 4)




ZD M4
BLNKRTN B2
INITIALIZE CARD CHAR COUNT TO 70
C2
INPUT HEX
NO
YES
ZD E4
CONHEX D2
DIVIDE CARD CHARACTER COUNT BY TWO
*E1
TWO HEXADECIMAL CHARACTERS PUNCHED IN CARD EQUAL ONE CHARACTER IN OUTPUT RECORD.
E2
STORE RESULT AS ACTUAL NO. OF CHARACTERS FOR RECORD *E1
F2
MULTIPLY ACTUAL COUNT BY TWO
G2
EVEN NO. OF HEX CHARS IN CD
NO
YES
PRTMSG8 G3
MOVE MSG8 'UNEVEN NO. OF HEX CHARACTERS' TO PRINT AREA
ZD J3
PRTANDCN
MVHEXCHR H2
CONVERT 2 HEX CHARS FROM CARD TO BINARY
J2
ERROR IN CONVERSION
YES
NO
PRTMSG9 J3
CLEAR PRINT AREA
K2
POINT TO NEXT CHARS TO BE CONVERTED
K3
MOVE MSG9 'INVALID HEX CHAR' TO PRINT AREA
ZD J3
PRTANDCN
B4
STORE CONVERTED CHARACTERS IN PRINT AREA
C4
ALL CHARACTERS CONVERTED
NO
YES
MVPRT D4
MOVE CONVERTED CHARACTERS FROM PRINT AREA TO CD AREA
ZD E4
UPCNT E4
ADD ACTUAL CHAR COUNT TO OUTPUT CHARACTER COUNT
F4
CHAR CNT GT MAX LNG RCD ALLOWED
YES
NO
TOOMUCH F5
MOVE MSG7 'DATA EXCEEDED' TO PRINT AREA
ZD J3
PRTANDCN
G4
MOVE CHARS FROM CARD AREA TO OUTPUT AREA
H4
UPDATE OUTPUT AREA POINTER BY NO. OF CHAR IN CARD AREA
J4
WRITE TAPE SW ON
NO
YES
ZD B2
WRTTPRTN
ZD-E2
ZD-J1
CCANAL J5
READ A CARD
ZD B4


WRTTPRTN
STORE OUTPUT CHARACTER COUNT IN CCW
TURN OFF WRITE TAPE SWITCH
WRITETP ZE
WRITE RCD ON SYS005
INTRTN
RESET POINTER TO OUTPUT AREA
CCANAL
ZB-J4
ZB-K5
WRRTN
INIT CARD CHAR COUNT TO 70 AND POINT TO LAST CARD COLUMN
ISIT
POINTER POINTING TO *
NO
YES
SUBTRACT ONE FROM COUNTER AND CARD COLUMN POINTER
GOTIT
DECREASE COUNTER BY 1 TO FIND NO. OF CHARACTERS IN CARD
70 DECREMENTS MADE
UPCNT
PRTMSG6
MOVE MSG 6 'NO * FOUND' TO PRINT AREA
TM IN COLS 7 AND 8
PRTMSG1
MOVE MSG1 'CONTROL INFO INVALID' TO PRINT AREA
ZB-H1
ZB-G3
ZB-J3
ZB-G5
ZC-F5
ZC-G3
ZC-K3
EXECTP ZE
WRITE A TAPE MARK ON SYS005
TMRTN
OUTPUT AREA EMPTY
PRTANDCN
WRITE MSG AND CARD IMAGE ON SYSLST
PRTMSG3
SVC 6 CANCEL


A4
WRITETP
2B D4
ENDRTN B2
OUTPUT AREA EMPTY
NO
PRTMSG2 B3
MOVE MSG2 'NO WT AT /*' TO PRINT AREA
2D J3
PRTANDCN
YES
C2
REWIND REQUESTED
NO
YES
REWIND D2
REWIND SYS005
*F1
CANCELS JOB AND GOES TO JOB CONTROL FOR EOJ STEP.
EOJRTN E2
WRITE 'EOJ CARD TO TAPE VARIABLE' ON SYSLST
F2
SVC 14 *F1
WRITETP B4
ADD ONE TO TAPE RCD COUNT
C4
RCD LNG LT 18 BYTES
NO
1
YES
SHTRECRT D4
RCD LENGTH ZERO
YES
ZERREC D5
MOVE MSG10 'ZERO LENGTH RECORD REQUESTED' TO PRINT AREA
2D J3
PRTANDCN
NO
CREATEN E4
CREATE RECORD NUMBER FOR MSG
PRTSRM F4
WRITE MSG 'RECORD NO. XXXXX IS SHORT' ON SYSLST
G3
EXECTP
1
G4
LOAD RETURN ADDRESS OF INTRTN IN LINKREG
EXECTP H4
EXECUTE TAPE COMMAND
J4
END OF VOLUME ON SYS005
YES
TPEOF J5
WRITE MSG12 '9J01I EOV ON SYS005' ON SYSLST AND SYSLOG
NO
K4
EXIT TO ADDR IN LINKREG
K5
SVC 6 CANCEL

DOS AUTOTEST LINKAGE EDITOR DIAGNOSTICS

All I-type diagnostic messages are written on SYSLST if the MAP option is assumed or specified. If NOMAP is operational, all I-type messages are written on SYSLOG. Also, diagnostic messages 9200I to 9299I are written on SYSLOG if MAP is operational.

The message 9299I is written on SYSLOG to inform the operator that errors occurred in the linkage editor input and were written on SYSLST (if MAP is operational). The job continues, if the ACTION CANCEL option is not used.

The diagnostic listing on SYSLST is intended to provide the programmer with a complete summary of user-prepared input, the names of the modules autolinked from the relocatable library, and all error messages.

Number	Message	Chart	Cause	Action
9100I	Content of statement in error. See the note following 9170I.	CA	Invalid input card type.	Correct the statement in error and resubmit the job.
9101I	Content of statement in error. See the note following 9170I.	DA	Invalid operation in control statement.	Correct the statement in error and resubmit the job.
9102I	Content of statement in error. See the note following 9170I.	AH DE	Non-decimal or non-hexadecimal character in decimal or hexadecimal field.	Correct the statement in error and resubmit the job.
9110I	Content of statement in error. See the note following 9170I.	DA DG	Invalid or missing field delimiter on control statement.	Correct the statement in error and resubmit the job.
9111I	Content of statement in error. See the note following 9170I.	DA	An operand field is greater than the maximum length on a user-prepared control statement or REP card.	Correct the statement in error and resubmit the job.
9112I	Content of statement in error. See the note following 9170I.	AU DG	An operand field is missing.	Correct the statement in error and resubmit the job.

Number	Message	Chart	Cause	Action
9113I	Content of statement in error. See the note following 9170I.	CC DG	Control statement extends beyond column 71.	Correct the statement in error and resubmit the job.
9114I	Content of statement in error. See the note following 9170I.	DB	Sub-modular namelist is too long.	Correct the statement in error and resubmit the job.
9115I	Content of statement in error. See the note following 9170I.	DF	NOAUTO expected, but not found.	Correct the statement in error and resubmit the job.
9116I	Content of statement in error. See the note following 9170I.	DA	Control statement present between first ESD and END statements of a module.	Correct the statement in error and resubmit the job.
9120I	Content of statement in error. See the note following 9170I.	EB	Phase name duplicated.	Correct the statement in error and resubmit the job.
9121I	Content of statement in error. See the note following 9170I.	EB	Phase name lower in sequence than $$A or phase name begins with *.	Correct the statement in error and resubmit the job.
9122I	Content of statement in error. See the note following 9170I.	EC	1. Symbol or phase name, designated in origin, not previously defined. 2. An F parameter has been detected in a PHASE card. Autotest does not permit linkage editing of foreground phases.	Correct the statement in error and resubmit the job.
9123I	Content of statement in error. See the note following 9170I.	EA	Previous phases processed contained no valid storage assignment.	Correct the statement in error and resubmit the job.

Number	Message	Chart	Cause	Action
9124I	Content of statement in error. See the note following 9170I.	ED	Phase origin is negative.	Correct the statement in error and resubmit the job.
9125I	Content of statement in error. See the note following 9170I.	DD	PHASE statement encountered during AUTOLINK.	Correct the statement in error and resubmit the job.
9130I	Content of statement in error. See the note following 9170I.	EG	Relocatable library not present.	Specify that the correct system pack be used when the job is resubmitted.
9131I	Content of statement in error. See the note following 9170I.	EG	Module requested by INCLUDE statement not present in the relocatable library.	Catalog the requested module in the relocatable library and resubmit the job.
9132I	Content of statement in error. See the note following 9170I.	EG	Too many nesting levels of INCLUDE attempted.	Correct the statement in error and resubmit the job.
9133I	Content of statement in error. See the note following 9170I.	DB	Nested sub-modular INCLUDE.	Correct the statement in error and resubmit the job.
9135I	Content of statement in error. See the note following 9170I.	AU	ACTION statement has invalid operand.	Correct the statement in error and resubmit the job.
9136I	Content of statement in error. See the note following 9170I.	AU	ACTION MAP specified, but SYSLST was not assigned.	Assign SYSLST when the job is resubmitted.
9140I	Content of statement in error. See the note following 9170I.	BB BD	ESD items of invalid type.	Correct the statement in error and resubmit the job.

Number	Message	Chart	Cause	Action
9141I	Content of statement in error. See the note following 9170I.	BH	Duplicated ESID number: 1. No END statement in last module. 2. Duplicate ESD card. 3. Extraneous ESD card.	Correct the statement in error and resubmit the job.
9142I	Content of statement in error. See the note following 9170I.	BB	ESD entry point label does not point to ESD named control section or COMMON.	Correct the statement in error and resubmit the job.
9143I	Content of statement in error. See the note following 9170I.	BF BG	Invalid duplication of entry point label.	Correct the statement in error and resubmit the job.
9144I	Content of statement in error. See the note following 9170I.	AE AN	Invalid ESID number or control dictionary and linkage table overlap.	Correct the statement in error and resubmit the job.
9145I	Content of statement in error. See the note following 9170I.	BC	Origin of control section not on a double-word boundary.	Correct the statement in error and resubmit the job.
9146I	Content of statement in error. See the note following 9170I.	BD	COMMON has the same label as a named control section or an entry point label.	Correct the statement in error and resubmit the job.
9147I	Content of statement in error. See the note following 9170I.	CH	ESD entry point label does not belong to a defined control section.	Correct the statement in error and resubmit the job.
9150I	Content of statement in error. See the note following 9170I.	AT	Load address encountered outside phase.	Correct the statement in error and resubmit the job.
9151I	Content of statement in error. See the note following 9170I.	CC	Invalid delimiter on REP card.	Correct the statement in error and resubmit the job.

Number	Message	Chart	Cause	Action
9155I	Content of statement in error. See the note following 9170I.	CD	The TXT or REP record, or address constant in an RLD record, does not have an ESID pointer to a defined control section.	Correct the statement in error and resubmit the job.
9156I	Content of statement in error. See the note following 9170I.	CE	Invalid format of RLD card.	Correct the statement in error and resubmit the job.
9158I	Content of statement in error. See the note following 9170I.	CH	END statement should contain the length of the control section, but does not.	Correct the statement in error and resubmit the job.
9170I	Content of statement in error. See the following note.	AE CB CD CG	ESID number not previously processed.	Correct the statement in error and resubmit the job.

Note: Statements in error (Error Numbers 9100I to 9170I) are written in the following formats:

1. If no 12-2-9 code in column 1 of card image, columns 1-80 of card image written in EBCDIC.

2. If 12-2-9 code in column 1 of card image:

Print Positions	Contains Card Image Columns
8-15	73-80 (identification) in EBCDIC
17-19	2-4 (card type) in EBCDIC
21-26	6-8 (assembled origin) in hexadecimal
28-31	11-12 (number of bytes in card image) in hexadecimal
33-36	15-16 (ESID number) in hexadecimal

Remainder of line depends on type of card image (ESD or non-ESD).

If non-ESD type card image, columns 17-52 are written in print positions 38-128. They are written in hexadecimal in blocks of 9 words (36 bytes), separated by one blank.

If ESD type card image, print positions 38-128 contain 3 fields of ESD information. Each field is 16 columns, broken down as follows:

Columns	Contains
17-24	ESD item name in EBCDIC
25	ESD type in EBCDIC
26-28	Assembled origin in hexadecimal
30-32	Length/ESID number in hexadecimal

Number	Message	Chart	Cause	Action
9200I	Linkage editor cannot continue.	GF	1. Highest byte of user program would overlay the area reserved for the Autotest control program at user program execution time. 2. The initial user's phase to be fetched would lie partially or wholly within the supervisor area.	Job is canceled.
9201I	Linkage editor cannot continue.	GG	Required Autotest phase not found in core image library.	Job is canceled.
9202I	Linkage editor cannot continue.	AB	All of user's core is not allocated to Autotest.	Job is canceled.
9203I	SYM out of order.	JB	Error in symbol processing. (SYM cards out of sequence.)	Processing continues. All symbols ignored.
9281I	Linkage editor cannot continue.	EA	No valid storage assignment in final phase.	Correct previous errors and resubmit the job.
9282I	Linkage editor cannot continue.	EF	No END record encountered before ENTRY statement.	Supply END record and resubmit the job.
9285I	Linkage editor cannot continue.	FC	An error occurred during the linkage editing of a $ phase.	Correct the errors and resubmit the job.
9291I	Linkage editor cannot continue.	AA CJ	1. End of file or extents exceeded on SYS001. 2. SYS001 not assigned to disk or tape.	Correct the SYS001 usage and resubmit the job.
9292I	Linkage editor cannot continue.	EE	End of librarian work area. Too many phases to process.	Break program down into less than 120 phases and resubmit the job.

Number	Message	Chart	Cause	Action
9293I	Linkage editor cannot continue.	AT	Core image library exceeded.	Condense the core image library or reallocate limits to create enough room, and resubmit the job.
9294I	Linkage editor cannot continue.	AM	Disk error. Invalid no-record-found condition.	Resubmit the job.
9299I	Error has occurred during linkage editing.	FC	Printed on SYSLOG if any of the errors 9200I through 9270I have occurred.	Job continues if the ACTION CANCEL option is not used; otherwise job is canceled.
9900I	Disk work area invalid.	AB	1. Minimum work area size requirement not met. (6 tracks for 16K main storage, 11 tracks for 32K or higher.) 2. Work area (IJYSAT) not assigned to SYSLNK.	Job is canceled, rerun with appropriate correction(s).
9901I	Disk work area too small.	JB	Insufficient work area for SYM card input.	Processing continues without symbolic capability.
9902I	Disk work area too small.	JB	1. Insufficient work area detected while writing linkage editor control dictionary on disk. 2. Insufficient work area detected while writing test request output record on disk.	Job is canceled. Rerun with larger work area.

Number	Message	Chart	Cause	Action
9A01I	Autotest cannot continue.	GG	All of user's main storage not allocated to Autotest. A change in core allocation has taken place by means of job control before execution of a // EXEC card.	Job is canceled. Rerun with corrected input deck.
9A02I	Option CATAL ignored.	AB	User supplied "OPTION CATAL".	Option ignored by Autotest linkage editor. Processing continues.

AUTOPATCH DIAGNOSTICS

The autopatch card image is always written on SYSLST. If any errors are found during the analysis of this card, one or more codes appear following the card image, as in the following example:

Card Image	Error
./ EXO X'1040',1A	C

Each error code can have more than one meaning as shown in the following chart.

Message	Code	Chart	Meaning	Action
Card Image	A	KF KL	Invalid patch-point address. 1. Address missing. 2. Decimal address used. 3. Address contains invalid character. 4. Address not within user's program. 5. Address not on halfword boundary (add and exchange constant only). 6. Missing quote sign.	Diagnostic processing continues. Job will be canceled.

Message	Code	Chart	Meaning	Action
Card Image	B	KL	Invalid symbolic patch-point address. 1. Symbol missing in the assumed or specified control section. 2. Symbols not present in user's program. 3. Address not within user's program. 4. Address not on halfwork boundary.	Diagnostic processing continues. Job will be canceled.
Card Image	C	KH	Invalid asterisk column specified. 1. Column number specified in control card not decimal. 2. Column number outside card limits. Patch data must not exceed column 71. 3. Column number missing.	Diagnostic processing continues. Job will be canceled.
Card Image	D	PN	Invalid patch card. 1. Asterisk missing or wrong column punched in patch card. 2. Patch card missing.	Diagnostic processing continues. Job will be canceled.
Card Image	E	KJ PJ	Invalid patch card information. 1. Hexadecimal representation of an instruction or constant incorrect. 2. Implied length of instruction incorrect (add or exchange constant only). 3. Blank column in hexadecimal field. 4. Odd number of hexadecimal characters for a constant replace.	Diagnostic processing continues. Job will be canceled.

Message	Code	Chart	Meaning	Action
Card Image	F	PQ KK	Invalid usage of SVC, BAL, or BALR. 1. New instruction to be placed in Autotest table is a SVC, BAL, or BALR. 2. Patch-point address is either a SVC, BAL, or BALR.	Diagnostic processing continues. Job will be canceled.
Card Image	G	PK KD	Number of generated SVC's (branch-out linkages) exceeds 200.	Job canceled.
Card Image	CARD IGNORED	KF	1. One parameter divided between two cards. 2. Incomplete parameter (missing comma). 3. Not a recognizable Autotest control card.	Diagnostic processing continues. Job will be canceled.
Card Image	PCC MISSING	KA	1. PCC card missing. 2. A job control card encountered before PCC card. 3. EOF occurred on SYSIPT.	Job will be canceled.
Card Image	TABLE OVERFLOW	PS	Autotest table would overlay highest user phase.	Diagnostic processing continues for PCC card only. Job will be canceled.

Number	Message	Chart	Cause	Action
9903I	DISK WORK AREA TOO SMALL.	KU KS	Test-request control table record or Autotest table record exceeds capacity of disk work area.	Job is canceled. Rerun with larger work area.

TEST-REQUEST DIAGNOSTICS

The test-request card-image is always written on SYSLST. If any errors are detected during the test-request diagnostic phase, one or more codes appear following the card image, as in the following example:

Card Image	Error
./ TR START,,DSP,X'08',ANY,,Z	Q

An SVC, BAL, or BALR instruction cannot be used at a test-point address. However, this condition may not be diagnosed if any other error occurs on a test request. An exception to this may occur if an error is diagnosed for the * code. The error codes can have more than one meaning, as shown in the following chart.

Message	Code	Chart	Meaning	Action
Card Image	F	MM	Invalid use of SVC, BAL, or BALR at test-point address.	Processing continues.
Card Image	K	LD	Invalid test-point address. 1. Test-point address specified in an invalid format. (Address must be specified either as a symbol or as hexadecimal characters.) 2. Relocated address not in phase used for relocation. 3. Test-point address plus implied length is invalid. 4. Test-point address not on a halfword boundary.	Processing continues.
Card Image	L	LF	Invalid symbolic address. 1. Symbol used is not contained in the assumed control section of phase used for relocation. 2. Symbol not contained in the control section specified, or there are no symbols present.	Processing continues.

Message	Code	Chart	Meaning	Action
Card Image	M	MX	One or more of the ON terms n1, n2, or n3 is invalid. 1. The term n1, n2, or n3 is zero. 2. One of the terms contains a character other than 0-9. 3. One of the terms exceeds 9999. 4. One term contains more than four decimal characters. 5. Term n1 greater than term n2. 6. Left parenthesis missing.	Processing continues.
Card Image	N	LN ME	Function names for DSP, PNL, or DPL incorrect. The incorrect parameter is flagged and no more diagnostics are performed on that card.	Processing continues with next card.
Card Image	O	LD MX	Invalid start address 1. The start address is missing. 2. The start address is invalid. 3. The decimal value used in the start address is in characters other than 0-9. 4. Relocated start address not in phase. 5. Address not within main storage limits.	Processing continues.

Message	Code	Chart	Meaning	Action
Card Image	P	LD	Invalid end address. 1. The end address is missing. 2. The end address is invalid. 3. The end address is less than the start address. 4. Relocated end address not in phase. 5. The decimal value used in the start address is in characters other than 0-9. 6. Address not within main storage limits.	Processing continues.
Card Image	Q	MK	1. Invalid character used to indicate type of display. 2. Floating-point dump requested when feature not present.	Processing continues.
Card Image	R	ML	Invalid output paramater. 1. Panel output term invalid.	Processing continues.
Card Image	T	LA MK	A continuation card was expected, based on the previous card. However, the current card contains ./ blank in columns 1-3 (not a valid continuation card).	Processing continues.
Card Image	PCC MISSING	KA	1. PCC card missing. 2. Job control card encountered before PCC card. 3. EOF occurred on SYSIPT.	Job will be canceled.
Card Image	*	MM	1. Too many parameters given on display. 2. Invalid halfword boundary specified. 3. Floating-point fullword or doubleword boundary incorrect.	Processing continues, test request is considered valid.

Message	Code	Chart	Meaning	Action
Card Image	CARD IGNORED	LD	1. Incomplete parameter or missing right parenthesis in the case of ON condition. 2. a. Blank column following a comma, when no continuation character is punched in column 72. b. One blank column following phase qualification is a valid control section; more than one blank is invalid.	Processing continues.
Card Image	CARD IGNORED	LD	Not a recognizable Autotest control card.	Job canceled.
Card Image	CARD IGNORED	LD	One parameter divided between two cards.	Job will be canceled, or, based on next card, processing continues if, 1. The current card contains a valid test-point address, blanks in columns 1-15, and DSP, PNL, or DPL in columns 16-18, or 2. Columns 1-3 contain ./ blank.

Number	Message	Chart	Cause	Action
9902I	DISK WORK AREA TOO SMALL.	NC	1. Insufficient work area detected while writing linkage editor control dictionary on disk. 2. Insufficient work area detected while writing test-request output record on disk.	Job canceled. Rerun with large work.
9903I	DISK WORK AREA TOO SMALL.	MV	Test-request control table records or Autotest table records exceed capacity of disk work area.	Job is canceled. Rerun with larger work area.

PHQ AND CSQ DIAGNOSTICS

Message	Code	Chart	Meaning	Action
Card Image	Z	KP KR KS PA	1. Control section specified in CSQ card not found. 2. Phase specified in PHQ card not found. 3. Control section not in specified phase. 4. Control section not in assumed phase.	Job Canceled.

PCC DIAGNOSTICS (ATLEFC5)

The PCC card image is always written on SYSLST. If any errors are detected during the PCC analysis phase, one or more numeric codes appear under the heading "Errors", as in the following example:

Card Image	Errors
./ PCC N,SAX,45	3

Message	Code	Meaning	Action
Card Image	1	Invalid entry in parameter 1.	Invalid entry ignored.
Card Image	2	Cannot find comma after first parameter.	Remainder of card ignored.
Card Image	3	Invalid entry in parameter 2.	Invalid entry ignored.
Card Image	4	Cannot find comma after second parameter.	Remainder of card ignored.
Card Image	5	Loop protection option exceeds four places.	A value of 50 is assumed.
Card Image	6	Loop protection option contains non-decimal entry.	A value of 50 is assumed.
Card Image	7	Symbolic dump requested, but symbols not present.	Invalid entry ignored.

PHASE LIST AND CORE MAP (ATLEFG1)

Number	Message	Chart	Meaning	Action
	PHASE NOT FOUND IN CONTROL DICTIONARY	VC	The phase fetched or loaded was not Autotest linkage edited in this job, but may be in the core image library.	Processing continues.
	CORE OVERFLOW	VH	Two internal tables have overlapped.	Processing continues, but no symbolic dump can be given.
	WITHIN SUPERVISOR	VC	User attempted to fetch or load a phase into an address within the supervisor area.	Fetch or load is not permitted, processing continues.
9905I	DISK WORK AREA TOO SMALL.	VF	Internal core map required for symbolic EOJ dump exceeds capacity of work area.	Job processing continues without symbolic EOJ dump. Rerun with larger work area.

SYMBOL SORT ROUTINE DIAGNOSTICS

Number	Message	Chart	Cause	Action
9906I	DISK WORK AREA TOO SMALL.	YD YM	Insufficient work area for symbol table post-processing.	Job processing continues without symbolic EOJ dump. Rerun with larger work area.

EOJ DUMP (ATLEFH3)

Certain diagnostic messages are written on SYSLST during the end-of-job dump phase of Autotest. Some of these messages have more than one meaning, as shown in the following chart.

Message	Chart	Meaning	Abnormal EOJ
PROGRAM CHECK INTERRUPTION--HEX LOCATION xxxxxx--CONDITION CODE n--(up to 32-byte description)	XA	A program check occurred in the user program.	Yes
ILLEGAL SVC AT X'nnnnnn' (nnnnnn is address of SVC)	XA	Invalid SVC detected by the supervisor at address nnnnnn.	Yes
PHASE NOT FOUND	XA	A user phase to be fetched or loaded was not found in the core image library.	Yes
OPERATOR REQUEST	XA	The operator has canceled the user program.	Yes
PROGRAM REQUEST	XA	The user program requested the cancel.	Yes
PHYSICAL I/O ERROR-n (n = 1 through 8 as shown under meaning)	XA	1. Invalid disk address. 2. Undefined logical unit. 3. Device not assigned. 4. Reading past /& card. 5. I/O error. 6. I/O operator option (operator canceled). 7. I/O error queue overflow. 8. User tried to seek outside of limits specified.	Yes
NORMAL END OF JOB	XA	Self explanatory.	No
ACTION COUNT OVERFLOW	XA	PCC action count exceeded.	Yes
SYMBOL ROUTINE IN ERROR	XH	Error detected in Autotest during end-of-job dump.	Yes
INVALID ADDRESS	XA	Invalid address furnished to supervisor by problem program.	Yes
NO SVC FOUND IN PATCH TABLE	XA	SVC number not in Autotest table. (Autotest table may have been altered by user.)	Yes
DISK WORK AREA OVERFLOW	XA	Overflow has occurred at object time execution. Either the work area was insufficient to record test-request output, and/or phase fetches and loads or a loop has occurred in the user's program.	Yes

AUTOTEST END ROUTINE (ATLEG01)

Message	Chart	Meaning	Action
ATEOF CARD ORDER ERROR	YP YQ	1. Job control card (// blank in cols. 1-3) has been encountered before the expected ATEOF card.	Job canceled.
MORE THAN ONE EXEC CARD IS INVALID IN AUTOTEST	YQ	Attempt made to re-execute job.	Job canceled.

DISASTER-CONTINUE DIAGNOSTICS (ATLECONT)

Number	Message	Chart	Cause	Action
9F02I	AUTOTEST COMMUNI-CATION RECORD NOT ON SYSLNK.	ZA	The diaster-continue routine has detected a wrong-length record in the first Autotest record of IJSYSAT. (The user program has written over Autotest information.)	Job canceled.

CARD-TO-TAPE VARIABLE DIAGNOSTICS (ATLEJCTV)

The Disk Operating System Autotest card-to-tape variable program utilizes two types of diagnostic messages. One type of diagnostic refers to the card-to-tape variable program itself while the other type refers to control card errors. In the case of a control card error, the control card image is also written on the next line following the message. These messages are written on SYSLST in the format shown in the following charts.

Message	Chart	Cause	Action
BEGIN CARD TO TAPE VARIABLE	ZB	This message is written at start of execution.	Processing continues.
RECORD NO. nnnnnn IS SHORT	ZE	A record of less than 18 characters is to be written. nnnnnn is the record number.	The short record is written and processing continues.
SYS005 IMPROPERLY ASSIGNED--LOG. UNIT	ZB	Improper logical unit assignment for SYS005.	Job canceled.
SYS005 IMPROPERLY ASSIGNED--PHY. UNIT	ZB	Improper physical unit assignment for SYS005.	Job canceled.

Message	Chart	Cause	Action
SYS005 IMPROPERLY ASSIGNED--NOT A TAPE	ZB	SYS005 not assigned to a tape unit.	Job canceled.
EOJ CARD TO TAPE VARIABLE	ZB	This message is written at normal termination.	Processing continues.
AUTOTEST CONTROL CARD ASSUMED (Card Image)	ZB	Last card read contains ./ columns 1 and 2.	Job canceled.
CONTROL INFO INVALID (Card Image)	ZD	Card not recognized as a control card.	Job canceled.
NO WT AT /* (Card Image)	ZE	/* card read while output area still contained unwritten data.	Job canceled.
NO WT AT TM (Card Image)	ZD	TM card read before output area was written.	Job canceled.
JOB CONTROL CARD ASSUMED (Card Image)	ZB	Card contains // blank in columns 1, 2, and 3.	Job canceled.
NO /* AT /& (Card Image)	ZB	Card containing /& in columns 1 and 2 was read before /* card.	Job canceled.
NO * FOUND (Card Image)	ZD	No terminating * on last WT or MR card read.	Job canceled.
DATA EXCEEDED (Card Image)	ZC	Output record exceeds available main storage size.	Job canceled.
UNEVEN NO. OF HEX. CHARACTERS (Card Image)	ZC	Card specified as hexadecimal information contains uneven number of characters in data.	Job canceled.
INVALID HEX. CHAR. (Card Image)	ZC	A pair of characters in a hexadecimal card could not be decoded.	Job canceled.
ZERO LENGTH RECORD REQUESTED (Card Image)	ZE	WT card was read when no data present in output area.	Job canceled.

Number	Message	Chart	Cause	Action
9J01I	EOV ON SYS005	ZE	End-of-volume on SYS005 (output tape) during card-to-tape variable program.	Job Canceled

APPENDIX B. PHASE NAME IDENTIFICATION

Phase Name	Phase Description	Control Section Name*	Chart(s)
ATLEDT	Autotest Linkage Editor - Initialization	1. IJVDNK20 2. IJVDNL20	1. AA-AR 2. AS-AU
ATLEDT10	Autotest Linkage Editor - ESD/Initial SYM Card Processor	IJVDSD20	BA-BJ
ATLEDT12	Autotest Linkage Editor - TXT/REP, RLD and END Card Processor	IJVDTH20	CA-CJ
ATLEDT14	Autotest Linkage Editor - Control Card Scanner	IJVDCN20	DA-DG
ATLEDT16	Autotest Linkage Editor - Control Card Processor	IJVDTL20	EA-EG
ATLEDT18	Autotest Linkage Editor - MAP Processor	IJVDAP20	FA-FD
ATLEDT1A	Autotest Linkage Editor - RLD Post-Processor	IJVDLD20	GA-GH
ATLEDT1B	Autotest Linkage Editor - SYM Card Processor	IJVDXL20	HA-HD
ATLEDT1C	Autotest Linkage Editor - Write Control Dictionary and Autotest Communication Area on SYSLNK	IJVDBO20	JA-JB
ATLEFC1	Control Card Analysis - Initialization	IJVTC120	KA
ATLEFC2	Control Card Analysis - Autopatch Diagnostics	IJVTC220	KB-KV
ATLEFC3	Control Card Analysis - Test-Request Diagnostics	IJVTC320	LA-LU
ATLEFC4	Control Card Analysis - Build Test-Request Table Entry	IJVTC420	MA-MX
ATLEFC5	Control Card Analysis - PCC Statement Processor	IJVTC520	NA-NC
ATLEFC7	Control Card Analysis - Autopatch User Program	IJVTC720	PA-PU
ATLEFE1	User Program Execution - Autotest Control Program Initialization	IJVFE120	QA-QB
$$BATST1	User Program Execution - Set Up Interface Between Autotest Control Program and Supervisor	IJVSS120	RA
ATLEFE2	User Program Execution - Autotest Control Program Execution	IJVTE220	SA-SF
$$BATST3	User Program Execution - Dump User Program	IJVST320	TA-TC
ATLEFF1	Post-Processing-Test-Request Output	IJVTF120	UA-US
ATLEFG1	Post-Processing-Phase List and Core Map	IJVTG120	VA-VL
ATLEFH2	Post-Processing - Symbol Organization	IJVTH220	WA-WE
ATLEFD1	Post-Processing - Symbol Sort	IJVTD120	XA-XE
ATLEFD2	Post-Processing - Symbol Merge	IJVTD220	XF-XN
ATLEFH3	Post-Processing - EOJ Dump	IJVTH320	YA-YN
ATLEGO1	Post-Processing - Autotest End Routine	IJVTI120	YP-YQ
ATLECONT	Disaster Continue	IJVTA020	ZA
ATLEJCTV	Card to Tape Variable Utility	IJVTJ120	ZB-ZE

*Note: The two terminating digits of the control section name are the version and modification level, respectively.

Figure 34. Phase Name Identification

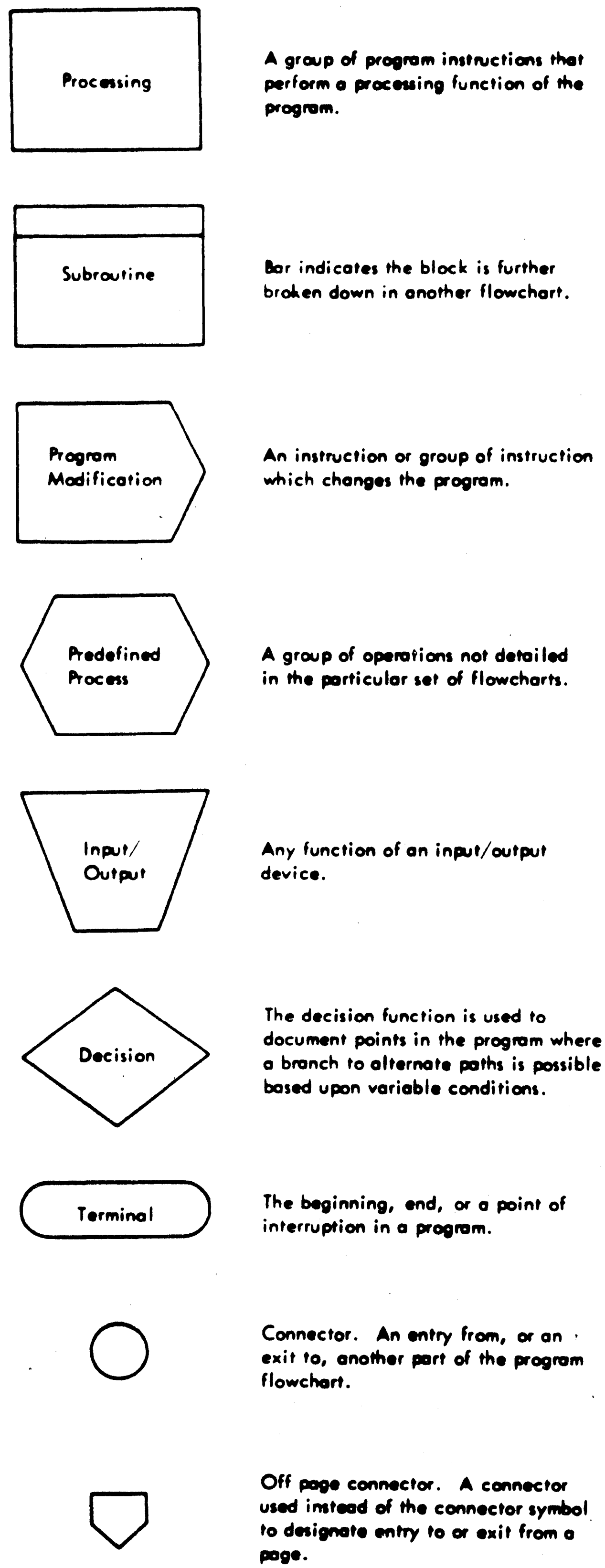


Figure 35. Flowchart Symbols

A/P	Autopatch
A/T	Autotest
A/T C/R	Autotest Communication Region
ABNORM	Abnormal
ADCON	Address Constant
ADDR	Address
ADJ	Adjust
ALPHA	Alphabetic
ASA	Carriage Control Character
ASMBLD	Assembled
ATEOF	Autotest End-Of-File
AUTOPAT	Autopatch
BG	Background
BKSP	Backspace
BLK	Block
BR	Branch
C/D	Control Dictionary
C/R	Communication Region
CALC	Calculate
CD	Card
CH	Chart
CHAR	Character
CI	Core Image
CI LIB	Core Image Library
CIL	Core Image Library
CMP	Compare
CNT	Count
COL	Column
COLL	Collating
COMM	Communication
COMP	Complete
CON	Constant
COND	Condition
CONTS	Contents
CONV	Convert
CSECT	Control Section
CSQ	Control Section Qualification
CTR	Counter
CTRL	Control
CTRL DICT	Control Dictionary
CURR	Current
CYL	Cylinder
DCML	Decimal
DECR	Decrease
DICT	Dictionary
DIG	Digit
DIR	Directory
DPL	Display and Panel
DSP	Display
DSPLT	Displacement
DTF	Define-The-File
DUP	Duplicate
EDT	Edited
ELEM	Element
ENT	Entry
ENTR	Enter
EOF	End-Of-File
EOJ	End-Of-Job
EQ	Equal
ERR	Error
ESD	External Symbol Dictionary
ESID	External Symbol Identification
EXC	Exchange
FLD	Field
FLT	Floating
FM	Form
FMT	Format
FUNC	Function
GENL	General
GT	Greater Than
HDR	Header
HEX	Hexadecimal
HI	High
I/O	Input/Output
ID	Identification
INCR	Increase
IND	Indicate
INDN	Indication
INDR	Indicator
INFO	Information
INIT	Initialize
INST	Instruction
L/E	Linkage Editor
LBL	Label
LD	Load
LG	Long
LH	Left Hand
LIB	Library
LMT	Limit
LNG	Length
LNK	Link
LNK EDT	Linkage Edited
LOC	Location
LOG	Logical
LST	Last
LT	Less Than
LUB	Logical Unit Block
MACH	Machine
MAX	Maximum
MIN	Minimum
MN	Main
MNEM	Mnemonic
MPL	Multiply
MSG	Message
MSK	Mask
MULT	Multiple
NEG	Negative
NO-OP	No Operation
NO.	Number
NOP	No Operation
NORM	Normal
OP	Operation
OPT	Output

ORD	Order
ORG	Origin
OVFLO	Overflow
OVRDG	Overriding
OVRLAY	Overlay
P PROG	Problem Program
PARAM	Parameter
PAREN	Parenthesis
PCC	Program Control Card
PH	Phase
PHQ	Phase Qualification
PHY	Physical
PIB	Program Information Block
PNL	Panel
POS	Position
PPA	Patch Point Address
PR	Print
PRES	Present
PREV	Previous
PRI	Primary
PROB	Problem
PROC	Process
PROG	Program
PT	Point
PTER	Pointer
PUB	Physical Unit Block
QLFR	Qualifier
QUAL	Qualified
RCD	Record
RCVD	Received
RD	Read
REG	Register
RELOC	Relocatable
REP	Representation
REQ	Request, Require
RET	Return
RGN	Region
RH	Right Hand
RLD	Relocatable List Dictionary
RSLT	Result
RT	Right
RTN	Routine
SEC	Secondary
SECT	Section
SEQ	Sequence
SP	Space
SPECFD	Specified
SRCH	Search
ST	Start
STG	Storage
STMNT	Statement
STRG	Starting
SUB	Subtract
SUP	Suppress
SUPVR	Supervisor
SYM	Symbol, Symbolic
SYS	System
SW	Switch
T/R	Test Request
TBL	Table
TEMP	Temporary
THRU	Through
TPA	Test Point Address
TR	Transfer
TRANS	Transient
TRK	Track
TXT	Text
UNC	Unconditional
VAR	Variable
VS	Versus
WD	Word
WLR	Wrong Length Record
WRK	Work

GY24-5027-0

IBM S/360 Printed in U.S.A. GY24-5027-0

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

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

READER'S COMMENT FORM

IBM System/360
Disk Operating System
Autotest

GY24-5027-0

- Your comments, accompanied by answers to the following questions, help us produce better publications for your use. If your answer to a question is "No" or requires qualification, please explain in the space provided below. All comments will be handled on a non-confidential basis. Copies of this and other IBM publications can be obtained through IBM Branch Offices.

	Yes	No
● Does this publication meet your needs?	▭	▭
● Did you find the material:		
Easy to read and understand?	▭	▭
Organized for convenient use?	▭	▭
Complete?	▭	▭
Well illustrated?	▭	▭
Written for your technical level?	▭	▭

- What is your occupation? ____________________
- How do you use this publication?

As an introduction to the subject?	▭	As an instructor in a class?	▭
For advanced knowledge of the subject?	▭	As a student in a class?	▭
For information about operating procedures?	▭	As a reference manual?	▭

Other ____________________

- Please give specific page and line references with your comments when appropriate.

COMMENTS:

GY24-5027-0

Fold

Fold

FIRST CLASS
PERMIT NO. 170
ENDICOTT, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

IBM Corporation
P. O. Box 6
Endicott, N. Y. 13760

Attention: Programming Publications, Dept. 157

Fold

Fold

Cut Along Line

IBM S/360 Printed in U.S.A. GY24-5027-0

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
[U.S.A. only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

Additional Comments: