

Program Logic

IBM System/360 Disk Operating System Logical IOCS

Volume 2: Unit Record, Magnetic Tape and Device Independent Files

Program Numbers

CDMOD	360N-IO-453
CPMOD	360N-CL-453
DIMOD	360N-CL-457
DTFCN	360N-CL-453
MRMOD	360N-IO-477
MTMOD	360N-IO-456
ORMOD	360N-IO-478
PRMOD	360N-CL-453
PTMOD	360N-IO-458

This reference publication is one of four Program Logic Manuals that describe the internal logic of the Logical IOCS (Input/Output Control System) programs for the IBM System/360 Disk Operating System. The four related Program Logic Manuals are:

- Volume 1: Introduction, Form Y24-5020.
- Volume 2: Unit Record, Magnetic Tape, and Device Independent Files, Form Y24-5087.
- Volume 3: Sequential and Direct Access DASD Files, Form Y24-5088.
- Volume 4: Indexed Sequential File Management System, Form Y24-5089.

This manual is intended for persons involved in program maintenance and by system programmers who are altering the program design. Program logic information is not necessary for the operation of the programs described. Therefore, distribution is limited to those with maintenance and alteration responsibilities.

Effective use of this publication requires an understanding of IBM System/360 operation and the Disk Operating System Assembler language and its associated macro definition language. Reference publications for this information are listed in the Preface.

For the titles and abstracts of other related publications, refer to the IBM System/360 Bibliography, Form A22-6822.

Restricted Distribution

RESTRICTED DISTRIBUTION: This publication is intended primarily for use by IBM personnel involved in program design and maintenance. It may not be made available to others without approval of local IBM management.

Second Edition (October 1968)

This edition applies to Release 19 of IBM System/360 Disk Operating System and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the specifications herein; before using this publication in connection with the operation of IBM systems, consult the latest System/360 SRL Newsletter, Form N20-0360, for the editions that are applicable and current.

This edition, Y24-5087-1, is a major revision of, and obsoletes Y24-5087-0 and Technical Newsletters Y24-5098 and Y24-5104.

Summary of Amendments

In addition to consolidating information in the obsoleted publications, this manual incorporates documentation describing changes made to Version 3 Disk Operating System Logical IOCS programs by Release 19 (Asynchronous Processing).

Changes are indicated by a vertical line to the left of affected text and to the left of affected parts of figures. A dot (•) next to a figure title or a page number indicates that the entire figure or page should be reviewed.

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

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Programming Publications, Endicott, New York 13760.

This Program Logic Manual (PLM) is the second in a series of four volumes providing a detailed guide to the IBM System/360 Disk Operating System Logical IOCS programs. It consists of four parts:

1. Logical IOCS support for Unit Record Files.
2. Logical IOCS support for Magnetic Tape Files.
3. Logical IOCS support for Device Independent Files.
4. Flowcharts.

Parts 1, 2, and 3 supply descriptions of the declarative and imperative macros, DTF tables, and initialization and termination procedures for each of the file-types described. Part 4 supplies the detailed flowcharts associated with the descriptions in the first three parts.

The appendixes in the back of this manual supply information intended to provide service aids to maintenance personnel. Included in this part are error message, cross-reference, and flowchart label lists.

PREREQUISITE PUBLICATIONS

- IBM System/360 Principles of Operation, Form A22-6821
- IBM System/360 Disk Operating System, Data Management Concepts, Form C24-3427
- IBM System/360 Disk Operating System, Supervisor and Input/Output Macros, Form C24-5037
- IBM System/360 Disk Operating System, System Control and System Service Programs, Form C24-5036
- IBM System/360 Disk and Tape Operating Systems, Assembler Specifications, Form C24-3414
- IBM System/360 Disk Operating System, Introduction to System Control

Programs, Program Logic Manual, Form Y24-5017

- IBM System/360 Disk Operating System, IPL and Job Control Programs, Program Logic Manual, Form Y24-5086
- IBM System/360 Disk Operating System, Supervisor and Physical and Logical Transients, Program Logic Manual, Form Y24-5084
- IBM System/360 Disk Operating System, Librarian Maintenance and Service Programs, Program Logic Manual, Form Y24-5079
- IBM System/360 Disk Operating System, Linkage Editor, Program Logic Manual, Form Y24-5080
- IBM System/360 Disk Operating System, Logical IOCS Program Logic Manual, Volume 1: Introduction, Form Y24-5020

RELATED PUBLICATIONS

- IBM System/360 Disk Operating System, Logical IOCS Program Logic Manual, Volume 3: Sequential and Direct Access DASD Files, Form Y24-5088
- IBM System/360 Disk Operating System, Logical IOCS Program Logic Manual, Volume 4: Indexed Sequential File Management System, Form Y24-5089
- IBM System/360 Disk Operating System, Basic Telecommunications Access Method Program Logic Manual, Form Y30-5001
- IBM System/360 Disk Operating System, Queued Telecommunications Access Method Program Logic Manual, Form Y30-5002
- IBM System/360 Disk Operating System, Operating Guide, Form C24-5022
- IBM System/360 Basic Programming Support, Basic Operating System, Tape Operating System, Disk Operating System, Tape Labels, Form C24-5070.

CONTENTS

INTRODUCTION	11	Characteristics of DTFCP files	76
UNIT RECORD FILES	12	Device Independent System Files	86
Initialization and Termination	12	Initialization and Termination	91
Card Device Files	13	CHARTS	99
Console Files	17	APPENDIX A: LABEL CROSS-REFERENCE LIST .277	
Magnetic Ink Character Recognition Files	19	APPENDIX B: MESSAGE CROSS-REFERENCE LIST286
Optical Reader Files	27	APPENDIX C: SUPERVISOR CALLS (SVC's) .288	
Printer Files	31	APPENDIX D: MICROFICHE CROSS-REFERENCE LIST FOR VOLUME 2 . . .290	
Paper-Tape Files	34	APPENDIX E: COMMAND CONTROL BLOCK (CCB)291
MAGNETIC TAPE FILES	42	APPENDIX F: CONTROL CODES294
DTFMT Macro	42	INDEX296
Data Files	42		
Workfiles	42		
DTFPH Macro (Magnetic Tape)	42		
MTMOD Macro	53		
Data Files	53		
Workfiles	53		
Initialization and Termination	58		
DEVICE INDEPENDENT FILES	75		
Compiler Files	75		

FIGURES

Figure 1. DTFCD: Input (Reader) . . .	14	Figure 15. DTFMT. Data Files (Part 1	of 8)	43
Figure 2. DTFCD: Output (Punch) . . .	15	Figure 16. DTFMT. Workfiles	51	
Figure 3. DTFCD: Combined		Figure 17. DTFPH. Magnetic Tape (Part	1 of 2)	52
Reader/Punch File	16	Figure 18. Tape File with Standard	Labels	58
Figure 4. DTFCN: Console	18	Figure 19. Tape File with Standard	and User Labels	58
Figure 5. DTFMR (Part 1 of 4)	20	Figure 20. Standard Tape Volume Label		59
Figure 6. MICR Buffer Format		Figure 21. Standard Tape File Label		60
(1412/1419)	25	Figure 22. Format of SYSRES Tape		
Figure 7. 1412/1419 Pocket Light		Label Information		61
Indicator	25	Figure 23. DTFCP: DISK=YES (Part 1 of	3)	77
Figure 8. DTFOR (Part 1 of 2)	28	Figure 24. DTFCP: DISK=NO (Part 1 of	2)	80
Figure 9. DTFPR	32	Figure 25. DTFCP: DISK= Parameter	Omitted (Part 1 of 2)	82
Figure 10. STL Control Field	33	Figure 26. DTFDI (Part 1 of 3)		87
Figure 11. DTFPT. No Translation, No				
Shifts or Deletes	35			
Figure 12. DTFPT. Translation, No				
Shifts or Deletes	36			
Figure 13. DTFPT. Translation, Shifts				
and Deletes, With Fixed Unblocked				
Records (Part 1 of 2)	37			
Figure 14. DTFPT. Translation, Shifts,				
and Deletes, with Undefined Records				
(Part 1 of 2).	39			

CHARTS

Chart 01. Magnetic Tape Open Routines	99	Chart EN. ORMOD: GET Macro, Blocked	
Chart 02. Magnetic Tape Close and		Records (Section 5 of 5)	134
EOF/EOV Routine	100	Chart EP. ORMOD: RDLNE Macro,	
Chart AA. \$\$BOUR01: Open Unit Record		Unblocked Records	135
(Section 1 of 2)	101	Chart EQ. ORMOD: RDLNE Macro,	
Chart AB. \$\$BOUR01: Open Unit Record		Blocked Records	136
(Section 2 of 2)	102	Chart ER. ORMOD: READ Macro	137
Chart BA. CDMOD: CNTRL Macro	103	Chart ES. ORMOD: RESCN Macro	
Chart BB. CDMOD: GET Macro (Section		(Section 1 of 2)	138
1 of 3)	104	Chart ET. ORMOD: RESCN Macro	
Chart BC. CDMOD: GET Macro (Section		(Section 2 of 2)	139
2 of 3)	105	Chart EU. ORMOD: WAITF Macro	
Chart BD. CDMOD: GET Macro (Section 3		(Section 1 of 4)	140
of 3)	106	Chart EV. ORMOD: WAITF Macro	
Chart BE. CDMOD: PUT Macro (Section 1		(Section 2 of 4)	141
of 4)	107	Chart EW. ORMOD: WAITF Macro	
Chart BF. CDMOD: PUT Macro (Section 2		(Section 3 of 4)	142
of 4)	108	Chart EX. ORMOD: WAITF Macro	
Chart BG. CDMOD: PUT Macro (Section 3		(Section 4 of 4)	143
of 4)	109	Chart EY. \$\$BOOR01: Open Optical	
Chart BH. CDMOD: PUT Macro (Section 4		Reader	144
of 4)	110	Chart FA. PRMOD: CNTRL Macro	145
Chart CA. DTFCN: GET Macro	111	Chart FB. PRMOD: PRTOV Macro	146
Chart CB. DTFCN: PUT Macro	112	Chart FC. PRMOD: PUT Macro (Section	
Chart DA. MRMOD: GET, READ, and		1 of 3)	147
CHECK Macros (Section 1 of 3)	113	Chart FD. PRMOD: PUT Macro (Section	
Chart DB. MRMOD: GET, READ, and		2 of 3)	148
CHECK Macros (Section 2 of 3)	114	Chart FE. PRMOD: PUT Macro (Section	
Chart DC. MRMOD: GET, READ, and		3 of 3)	149
CHECK Macros (Section 3 of 3)	115	Chart GA. PTMOD: GET Macro, No	
Chart DD. MRMOD: DISEN Macro		Translation	150
(Disengage)	116	Chart GB. PTMOD: GET Macro,	
Chart DE. MRMOD: LITE Macro	117	Translation of Shifted Code	151
Chart DF. MRMOD: WAITF Macro	118	Chart GC. PTMOD: GET Macro;	
Chart DG. \$\$BOMR01: Open MICR	119	Translation of Shifted Code, Fixed	
Chart DH. \$\$BCMR01: Close MICR	120	Unblocked Records (Section 1 of 2)	152
Chart DJ. \$\$BMMR20: MICR Message		Chart GD. PTMOD: GET Macro;	
Writer	121	Translation of Shifted Code, Fixed	
Chart EA. ORMOD: CNTRL Macro		Unblocked Records (Section 2 of 2)	153
(Section 1 of 3)	122	Chart GE. PTMOD: GET Macro;	
Chart EB. ORMOD: CNTRL Macro		Translation of Shifted Code, Undefined	
(Section 2 of 3)	123	Record (Section 1 of 2)	154
Chart EC. ORMOD: CNTRL Macro		Chart GF. PTMOD: GET Macro;	
(Section 3 of 3)	124	Translation of Shifted Code, Undefined	
Chart ED. ORMOD: DSPLY	125	Record (Section 2 of 2)	155
Chart EE. ORMOD: GET Macro,		Chart HA. MTMOD: CHECK Macro,	
Unblocked Records (Section 1 of 4)	126	Workfile	156
Chart EF. ORMOD: GET Macro,		Chart HB. MTMOD: CNTRL Macro, Data	
Unblocked Records (Section 2 of 4)	127	Files	157
Chart EG. ORMOD: GET Macro,		Chart HC. MTMOD: CNTRL Macro,	
Unblocked Records (Section 3 of 4)	128	Workfile	158
Chart EH. ORMOD: GET Macro,		Chart HD. MTMOD: FEOV Macro	159
Unblocked Records (Section 4 of 4)	129	Chart HE. MTMOD: GET Macro (Section	
Chart EJ. ORMOD: GET Macro, Blocked		1 of 3)	160
Records (Section 1 of 5)	130	Chart HF. MTMOD: GET Macro (Section	
Chart EK. ORMOD: GET Macro, Blocked		2 of 3)	161
Records (Section 2 of 5)	131	Chart HG. MTMOD: GET Macro (Section	
Chart EL. ORMOD: GET Macro, Blocked		3 of 3)	162
Records (Section 3 of 5)	132	Chart HH. MTMOD: NOTE Macro, Workfile	163
Chart EM. ORMOD: GET Macro, Blocked		Chart HJ. MTMOD: POINTR Macro,	
Records (Section 4 of 5)	133	Workfile	164

Chart HK. MTMOD: POINTS Macro, Workfile	165	Chart KH. \$\$BOMT05: Open Input or Output, Nonstandard or Unlabeled (Section 1 of 2)	194
Chart HL. MTMOD: POINTW Macro, Workfile	166	Chart KJ. \$\$BOMT05: Open Input or Output, Nonstandard or Unlabeled (Section 2 of 2)	195
Chart HM. MTMOD: PUT Macro (Section 1 of 2)	167	Chart KK. \$\$BOMT06: Open Workfiles, Magnetic Tape (Part 1 of 2)	196
Chart HN. MTMOD: PUT Macro (Section 2 of 2)	168	Chart KKA. \$\$BOMT06: Open Workfiles, Magnetic Tape (Part 2 of 2)	197
Chart HP. MTMOD: READ Macro, Workfile	169	Chart KL. \$\$BJCOPT: Job Control Tape Open Routine (Section 1 of 3)	198
Chart HQ. MTMOD: RELSE Macro	170	Chart KM. \$\$BJCOPT: Job Control Tape Open Routine (Section 2 of 3)	199
Chart HR. MTMOD: TRUNC Macro	171	Chart KN. \$\$BJCOPT: Job Control Tape Open Routine (Section 3 of 3)	200
Chart HS. MTMOD: WRITE Macro, Workfile	172	Chart KP. \$\$BCEOV1: EOF/EOV Monitor (Section 1 of 2)	201
Chart HT. MTMOD: Blocking/Deblocking Subroutines	173	Chart KQ. \$\$BCEOV1: EOF/EOV Monitor (Section 2 of 2)	202
Chart HU. MTMOD: Work Area Subroutines, Fixed and Variable Records	174	Chart KR. \$\$BCMT01: Tape Close, EOF/EOV Input Forward, Phase 1 (Section 1 of 2)	203
Chart HV. MTMOD: Work Area Subroutine, Undefined Records	175	Chart KS. \$\$BCMT01: Tape Close, EOF/EOV Input Forward, Phase 1 (Section 2 of 2)	204
Chart HW. MTMOD: Read/Write Subroutine, Fixed Records (Section 1 of 3)	176	Chart KT. \$\$BCMT02: Tape Close, Alternate Switching for EOVS (Section 1 of 2)	205
Chart HX. MTMOD: Read/Write Subroutine, Fixed Records (Section 2 of 3)	177	Chart KU. \$\$BCMT02: Tape Close, Alternate Switching for EOVS (Section 2 of 2)	206
Chart HY. MTMOD: Read/Write Subroutine, Fixed Records (Section 3 of 3)	178	Chart KV. \$\$BCMT03: Tape Close, EOF Input Backward	207
Chart HZ. MTMOD: Read/Write Subroutine, Variable Records (Section 1 of 3)	179	Chart KW. \$\$BCMT04: Tape Close, EOF Output Forward (Section 1 of 2)	208
Chart JA. MTMOD: Read/Write Subroutine, Variable Records (Section 2 of 3)	180	Chart KX. \$\$BCMT04: Tape Close, EOF Output Forward (Section 2 of 2)	209
Chart JB. MTMOD: Read/Write Subroutine, Variable Records (Section 3 of 3)	181	Chart KY. \$\$BCMT05: Close Standard, Nonstandard, and Unlabeled Files, All Types Except Workfiles (Section 1 of 2)	210
Chart JC. MTMOD: Read/Write Subroutine, Undefined Records (Section 1 of 3)	182	Chart KZ. \$\$BCMT05: Close Standard, Nonstandard, and Unlabeled Files, All Types Except Workfiles (Section 2 of 2)	211
Chart JD. MTMOD: Read/Write Subroutine, Undefined Records (Section 2 of 3)	183	Chart LA. \$\$BCMT06: Close Tape Workfiles	212
Chart JE. MTMOD: Read/Write Subroutine, Undefined Records (Section 3 of 3)	184	Chart LB. \$\$BCMT07: Alternate Switching for System Units (Section 1 of 2)	213
Chart JF. MTMOD: Bypass Checkpoint Records routine	185	Chart LC. \$\$BCMT07: Alternate Switching for System Units (Section 2 of 2)	214
Chart JG. MTMOD: Workfile Subroutines	186	Chart LD. Tape Open/Close Subroutines	215
Chart KA. \$\$BOMT01: Open Input Standard Labels, Forward (Section 1 of 3)	187	Chart LE. \$\$BOMT0M and \$\$BOMTOW: Tape Open Message Writers	216
Chart KB. \$\$BOMT01: Open Input Standard Labels, Forward (Section 2 of 3)	188	Chart LF. \$\$BMSGWR: Tape Open/Close Message Writer	217
Chart KC. \$\$BOMT01: Open Input Standard Labels, Forward (Section 3 of 3)	189	Chart MA. CPMOD: GET Macro, 2 I/O Areas	218
Chart KD. \$\$BOMT02: Open Input Standard Labels, Backward	190	Chart MB. CPMOD: GET Macro, 1 I/O Area	219
Chart KE. \$\$BOMT03: Open Output Standard Labels, Forward, Phase 1 (Section 1 of 2)	191	Chart MC. CPMOD: GET Macro, IOPTR=YES	220
Chart KF. \$\$BOMT03: Open Output Standard Labels, Forward, Phase 1 (Section 2 of 2)	192	Chart MD. CPMOD: PUT Macro, 2 I/O Areas (Section 1 of 3)	221
Chart KG. \$\$BOMT04: Open Output Standard Labels, Forward, Phase 2	193	Chart ME. CPMOD: PUT Macro, 2 I/O Areas (Section 2 of 3)	222
		Chart MF. CPMOD: PUT Macro, 2 I/O Areas (Section 3 of 3)	223

Chart MG. CPMOD: PUT Macro, 1 I/O Area (Section 1 of 2)224	Chart QM. \$\$BOCPT1: Open DTFCP and DTFDI Input Tape (Section 1 of 5)250
Chart MH. CPMOD: PUT Macro, 1 I/O Area (Section 2 of 2)225	Chart QN. \$\$BOCPT1: Open DTFCP and DTFDI Input Tape (Section 2 of 5)251
Chart MJ. CPMOD: PUT Macro, IOPTR=YES (Section 1 of 2)226	Chart QP. \$\$BOCPT1: Open DTFCP and DTFDI Input Tape (Section 3 of 5)252
Chart MK. CPMOD: PUT Macro, IOPTR=YES (Section 2 of 2)227	Chart QR. \$\$BOCPT1: Open DTFCP and DTFDI Input Tape (Section 4 of 5)253
Chart NA. DIMOD: GET Macro, 1 I/O Area (Section 1 of 2)228	Chart QS. \$\$BOCPT1: Open DTFCP and DTFDI Input Tape (Section 5 of 5)254
Chart NB. DIMOD: GET Macro, 1 I/O Area (Section 2 of 2)229	Chart QT. \$\$BOCPT2: Open DTFCP and DTFDI Output Tape (Section 1 of 6)255
Chart NC. DIMOD: GET Macro, 2 I/O Areas (Section 1 of 2)230	Chart QU. \$\$BOCPT2: Open DTFCP and DTFDI Output Tape (Section 2 of 6)256
Chart ND. DIMOD: GET Macro, 2 I/O Areas (Section 2 of 2)231	Chart QV. \$\$BOCPT2: Open DTFCP and DTFDI Output Tape (Section 3 of 6)257
Chart NE. DIMOD: PUT Macro, 1 I/O Area (Section 1 of 3)232	Chart QW. \$\$BOCPT2: Open DTFCP and DTFDI Output Tape (Section 4 of 6)258
Chart NF. DIMOD: PUT Macro, 1 I/O Area (Section 2 of 3)233	Chart QX. \$\$BOCPT2: Open DTFCP and DTFDI Output Tape (Section 5 of 6)259
Chart NG. DIMOD: PUT Macro, 1 I/O Area (Section 3 of 3)234	Chart QY. \$\$BOCPT2: Open DTFCP and DTFDI Output Tape (Section 6 of 6)260
Chart NH. DIMOD: PUT Macro, 2 I/O Areas (Section 1 of 3)235	Chart QZ. \$\$BCCPT1: Close DTFCP and DTFDI Tape Files (Section 1 of 3)261
Chart NJ. DIMOD: PUT Macro, 2 I/O Areas (Section 2 of 3)236	Chart RA. \$\$BCCPT1: Close DTFCP and DTFDI Tape Files (Section 2 of 3)262
Chart NK. DIMOD: PUT Macro, 2 I/O Areas (Section 3 of 3)237	Chart RB. \$\$BCCPT1: Close DTFCP and DTFDI Tape Files (Section 3 of 3)263
Chart PA. \$\$BERRTN: Punch Error Recovery Routine238	Chart RC. DTFCP and DTFDI: I/O Subroutines264
Chart QA. \$\$BOCP01: Open Device Independent Files, Phase 1 (Section 1 of 3)239	Chart RD. \$\$BCLOSP: Punch File Close (Section 1 of 4)265
Chart QB. \$\$BOCP01: Open Device Independent Files, Phase 1 (Section 2 of 3)240	Chart RE. \$\$BCLOSP: Punch File Close (Section 2 of 4)266
Chart QC. \$\$BOCP01: Open Device Independent Files, Phase 1 (Section 3 of 3)241	Chart RF. \$\$BCLOSP: Punch File Close (Section 3 of 4)267
Chart QD. \$\$BOCP02: Open Device Independent Files, Phase 2 (Section 1 of 3)242	Chart RG. \$\$BCLOSP: Punch File Close (Section 4 of 4)268
Chart QE. \$\$BOCP02: Open Device Independent Files, Phase 2 (Section 2 of 3)243	Chart RH. \$\$BOCPM1 and \$\$BOCPM2: DTFCP/DTFDI Message Writers (Section 1 of 3)269
Chart QF. \$\$BOCP02: Open Device Independent Files, Phase 2 (Section 3 of 3)244	Chart RJ. \$\$BOCPM1 and \$\$BOCPM2: DTFCP/DTFDI Message Writers (Section 2 of 3)270
Chart QG. \$\$BOCP11: Open DTFCP (Version 1 Only), Phase 1 (Section 1 of 3)245	Chart RK. \$\$BOCPM1 and \$\$BOCPM2: DTFCP/DTFDI Message Writers (Section 3 of 3)271
Chart QH. \$\$BOCP11: Open DTFCP (Version 1 Only), Phase 1 (Section 2 of 3)246	Chart RL. \$\$BOMSG1: Disk Open Error Message Writer, Phase 1 (Section 1 of 2)272
Chart QJ. \$\$BOCP11: Open DTFCP (Version 1 Only), Phase 1 (Section 3 of 3)247	Chart RM. \$\$BOMSG1: Disk Open Error Message Writer, Phase 1 (Section 2 of 2)273
Chart QK. \$\$BOCP12: Open DTFCP (Version 1 Only), Phase 2 (Section 1 of 2)248	Chart RN. \$\$BOMSG2: Disk Open Error Message Writer, Phase 2 (Section 1 of 3)274
Chart QL. \$\$BOCP12: Open DTFCP (Version 1 Only), Phase 2 (Section 2 of 2)249	Chart RP. \$\$BOMSG2: Disk Open Error Message Writer, Phase 2 (Section 2 of 3)275
		Chart RQ. \$\$BOMSG2: Disk Open Error Message Writer, Phase 2 (Section 3 of 3)276

This volume of the Disk Operating System Logical IOCS Program Logic Manual contains detailed information on the logical IOCS support of unit record, magnetic tape, and device independent files. It is intended for trained maintenance personnel experienced in the concept of using the Logical Input/Output Control System (LIOCS) for file processing.

Beyond brief introductory descriptions of each of the files covered, this volume does not contain information of a general nature. If the reader requires basic knowledge or a review of the general concept and function of logical IOCS, he should refer to Volume 1 of Disk Operating System Logical IOCS listed on the front cover of this volume.

This volume contains information on all the logical IOCS items (modules, DTF tables, imperative macros, open and close routines, etc.) required for the particular files discussed. The only exceptions are certain common and special purpose routines which cannot be related to any specific file. These routines; namely, the Open and Close Monitors, the Open routines for self-relocating programs, and the Checkpoint/Restart routines are described in detail in Volume 1.

The files discussed in this volume are divided into three parts:

- Unit Record Files
- Magnetic Tape Files
- Device Independent Files

Files within a given group are presented in alphabetic sequence according to the last two letters of the DTFxx macro that defines the file (i.e., DTFCD, DTFCN...DTFPT). Access to information on a particular file type can be made through either the Table

of Contents or, more specifically, the Index. The information relating to a file type includes, in the order presented:

- The file definition (DTFxx) macro.
- The module generation (xxMOD) macro.
- The imperative LIOCS macros (GET, READ, etc) used with the file.
- The special Open and Close routines, if applicable. (Open phase \$\$BOUR01 for unit record files is presented first, because it applies to most of the unit record files supported by DOS Logical IOCS.)
- The special purpose routines, such as message writers, if applicable.

Part 4 contains the generalized and detailed flowcharts of the imperative logical IOCS macros supported by each of the data handling logic modules and the logical transient routines required for Open, Close, and other special functions.

The logic supporting each of the imperative macros has been flowcharted from macro language (source statement) listings. In some instances these flowcharts contain decision blocks to illustrate the logic (i.e., coding) included in the modules for certain xxMOD macro parameter options. The reader should realize that these decisions do not appear in an assembly listing, but rather that a particular assembly listing is the result of these decisions being made at the time the logic module is generated.

If an assembly listing is not available for a specific logic module, a listing of the source statements that generate the module can be obtained from microfiche cards referenced in Appendix D of this manual.

UNIT RECORD FILES

The IBM System/360 Disk Operating System provides logical IOCS support for files on the following devices generally categorized as unit-record equipment:

- 1050 Console Keyboard Printer
- 1259 Magnetic Character Reader
- 1285/1287 Optical Readers
- 1403 Printer
- 1404 Printer (continuous forms only)
- 1412/1419 Magnetic Ink Character Readers
- 1442N1 Card Reader-Punch
- 1442N2 Card Punch
- 1443/1445 Printers
- 2501 Card Reader
- 2520B1 Card Reader-Punch
- 2520B2/B3 Card Punch
- 2540R Card Reader
- 2540P Card Punch
- 2671 Paper Tape Reader

The files used with these devices are defined by a DTFxx declarative macro and the data handling LIOCS module is generated, except for Console Files, by an associated xxMOD macro. (The DTFCN declarative macro not only defines the file but also provides the data handling logic module for console files.)

The files described in this part include:

- Card - card readers and punches
- Console
- Optical Reader
- Magnetic Ink Character Recognition
- Printer
- Paper Tape

INITIALIZATION AND TERMINATION

Processing of a file by logical IOCS requires that the file be initialized, or opened, prior to the transfer of any data by the problem program. Likewise, when the transfer of all data is complete, the file is closed.

With the exception of magnetic ink and optical reader files, which are handled separately, all unit-record files are opened by the unit record open logical transient phase, \$\$BOUR01, fetched by the Open Monitor (refer to Volume 1). On the other hand, unit-record files (except magnetic ink character reader files) require no special termination procedures and are closed by the Close Monitor that simply resets the open indicator in the DTF table for the file.

\$\$BOUR01: Open Unit Record Charts AA-AB

Objective: To open a unit-record file.

Entry: From the Open Monitor.

Exits: To \$\$BOPEN, if the next file to be opened is not a unit-record file.

To the problem program, if no more files are to be opened.

To \$\$BOMSG1, to print out the error message:

4883I INVALID LOGICAL UNIT

Method: Entry to \$\$BOUR01 is from the OPEN Monitor, \$\$BOPEN, after the monitor has built the Open table in the top of the transient area and validated the limits of the DTF. This is the proper phase for opening DTFs types 02 and 04 through 08 as indicated in byte 20 of the DTF.

The SYSIR macro finds the referenced LUB, and message 4883I is issued if it is unassigned. If the LUB specifies that the logical unit is to be ignored, a bit is set in the DTF but no message is given. If the LUB is assigned, this routine opens unit record files for the following DTF types by performing the functions indicated:

1. Reader. The open switch is turned on.

2. Printer. The open switch is turned on, and an I/O register is loaded if there are two I/O areas. If the printer has the universal character set feature, a set mode instruction is issued to allow or suppress data checks in accordance with user's choice.
3. Punch. The open switch is turned on, and an I/O register is loaded if there are two I/O areas.

After the DTF has been initialized by turning on the open indicator, this phase determines if there is another file to be opened. If not, control is returned to the problem program, via an SVC 11. If the next file is not unit record, the Open Monitor, \$\$BOPEN, is called.

CARD DEVICE FILES

Card Device (CD) files are those files associated with a card reader and/or a card punch. The devices supported include:

- 1442N1 Reader-Punch
- 1442N2 Punch
- 2501 Reader
- 2520B1 Reader-Punch
- 2520B2 Punch

- 2520B3 Punch
- 2540R Reader
- 2540P Punch

The files associated with these devices are defined by the DTFCD macro.

DTFCD Macro

Three types of DTF tables can be generated by the DTFCD macro. The DTF table type generated for a particular file depends on the TYPEFLE= parameter specified by the user in the DTFCD macro. The three table types are:

- DTFCD: Input (Reader) if TYPEFLE=INPUT (Figure 1).
- DTFCD: Output (Punch) if TYPEFLE=OUTPUT (Figure 2).
- DTFCD: Combined if TYPEFLE=CMBND (Figure 3). This parameter can be specified for 1442N1 or 2520B1 reader-punch, or a 2540 punch with the punch-feed-read (PFR) feature.

The generated DTFCD table contains information describing the file and serves as a linkage to the CD logic module that is generated by a corresponding CDMOD macro.

Bytes	Bits	Contents	Function
0-15			CCB.
16	0-1		Not used.
	2		COBOL open; ignore option.
	3		Not used.
	4		DTF table address constants relocated by OPENR.
	5-7		Not used.
17-19			Address of logic module.
20		X'02'	DTF type.
21	0		1 = open; 0 = closed.
	1		First time switch.
	2		1 = 1442; 0 = other.
	3		Reserved.
	4		Reserved.
	5		1 = 2 I/O areas; 0 = 1 I/O area.
	6		1 = 2520; 0 = other.
	7		1 = 2540; 0 = other.
22			Normal command code.
23			Control command code.
24-27			Address of IOAREA2.
28-31			Address of EOF routine.
32-39			CCW.
40-43		LA &IOREG,0(14)	
44-49		MVC 0(&BLKSIZE,13),0(14)	

Figure 1. DTFCD: Input (Reader)

Bytes	Bits	Contents	Function
0-15			CCB.
16	0-1		Not used.
	2		COBOL open; ignore option.
	3		Not used.
	4		DTF table address constants relocated by OPENR.
	5-7		Not used.
17-19			Address of logic module.
20		X'04'	DTF type.
21	0		1 = open; 0 = closed.
	1		First time switch.
	2		1 = CTLCHR.
	3		1 = fixed unblocked.
	4		1 = variable unblocked.
	5		1 = 2 I/O areas.
	6		1 = workarea.
	7		1 = 2 CCW's in table.
	7		0 = 1 CCW in table.
22			Normal command code.
23			Control command code.
24-27			Address of IOAREA1.
28-31			Bucket. ¹
32-33		LR 12, (RECSIZE)	
34-37		LA &IOREG, 4(14)	
38	0		Not used.
	1		Not used.
	2		Not used.
	3		Not used.
	4		Not used.
	5		Not used.
	6		1 = 1442.
	7		1 = 2520B1.
39			Blank.
40-47			CCW.
48-55			CCW2 for ejecting last program card if device is a 2520B1.
			Used for repunching error card on punch file if device is a 2540.
56-135			80-byte area for saving card image for 2540 punch error retry.

¹The bucket bytes handle undefined records.

Figure 2. DTFCD: Output (Punch)

Bytes	Bits	Contents	Function
0-15			CCB.
16	0-1		Not used.
	2		COBOL open; ignore option.
	3		Not used.
	4		DTF table address constants relocated by OPENR.
	5-7		Not used.
17-19			Address of logical module.
20		X'00'	DTF type.
21			Command code (X'02' for 1442, X'C2' for 2525,2540)
22			Command code (X'01' for 1442, X'09' for 2520,2540)
23			Command code (X'01' for 1442, X'09' for 2520,2540)
24-31			CCW.
32-35			Input area address.
36-39			Output area address.
40-41			Input blocksize.
42-43			Output blocksize.
44-49		MVC 0(&BLKS,13),0(14)	
50-55		MVC 0(&OUBL,14),0(13)	
56-59			End of file address.
60-67			Savearea.
68-73		MVC 1(&OUBL-1,13),0(13)	
74-77		MVI 0(13),X'40'	
78-79			Constant (blanks).
80-83			Constant address (bytes 78-79)

Figure 3. DTFC: Combined Reader/Punch File

CDMOD Macro

The CDMOD macro provides the model statements needed to generate logic modules that contain the GET, PUT, and CNTRL logic for a card reader or punch. It includes the capacity to handle a combined file, which uses the Punch Feed Read (PFR) feature on the IBM 2540. The combined file can also be handled on the IBM 1442-N1.

Over 200 different logical modules can be generated from the CDMOD macro. Because it would be impractical to flowchart and describe every possible variation, the CDMOD macro internals are flowcharted to indicate the many parameter variations and combinations that can be handled.

The CDMOD macro can generate logic modules needed to handle all the normal record forms applicable to a card reader/

punch. A particular CD logic module is designed to handle only those items that are indicated by the CDMOD macro parameters. For instance, if TYPEFLE=INPUT and RECFORM=FIXUNB are specified, the generated module does not handle output files, nor variable or undefined records. The logic module can include the use of one or two I/O areas, the use of a work area to process the records, and the logic to handle the CNTRL macro if stacker selection is desired.

A separate logic module is needed for every reader/punch record format variation.

CDMOD: CNTRL Macro Chart BA

Objective: To stacker select a card into a specified stacker pocket.

Entry: From a CNTRL macro expansion.

Exit: To the problem program.

Method: The control code specified by the user passes to the CNTRL routine in general register 0. The CNTRL routine then modifies the control code as required for the file type and device specified for the file. The modified control code is stored in the command code byte in the DTF table.

If the file is an input file defined for a device other than a 1442, the control operation is performed immediately. In all other cases, the control operation is delayed until the next GET or PUT macro is issued.

CDMOD: GET Macro Charts BB-BD

Objective: To read a card (normal or combined file).

Entry: From a GET macro expansion.

Exit: To the problem program or to the user's EOF routine.

Method: For regular input files when two I/O areas are used, I/O and processing overlap is possible. On the first GET operation, one I/O area is filled and another SVC 0 is issued before any processing is allowed. After the second SVC 0 is issued control returns to the problem program so that processing of data in the first I/O area can begin. If two I/O areas are specified, the first GET macro instruction sets a switch, which changes the logic flow of all the following

GET instructions used (in the problem program) with this logic module.

For combined files, IOAREA2 serves as the output area if it is specified; and IOAREA1 is the input area. If IOAREA2 is not specified, IOAREA1 serves for both input and output.

CDMOD: PUT Macro Charts BE-BH

Objective: To punch a card (normal or combined files).

Entry: From a PUT macro expansion.

Exit: To the problem program.

Method: For regular output card files, the PUT routine punches the next sequential record in the file into a card. If WORKA=YES is specified in the DTFCD and CDMOD macros, the record is built in the workarea and then moved to the appropriate output area before punching is performed. If IOAREA2=YES is specified in the DTFCD and CDMOD macros, overlap of I/O and processing is possible by alternately switching output areas.

For combined files (1442 or 2540 with PFR), IOAREA2 if it is specified, is used as the output area. If IOAREA2 is not specified, IOAREA1 serves as both the input and the output area.

CONSOLE FILES

Console files (CN) are files associated with the system 1050 console. These files are defined, by a DTFCN macro, as input or output files containing either fixed unblocked or undefined records.

Console files are neither opened nor closed by logical IOCS.

DTFCN Macro

Logical IOCS support of console files differs from other files. One macro, DTFCN, generates both the DTF table and the logic module (Figure 4). The Version 3 DTFCN macro generates both table and module from statements contained in the source statement library. (Previous versions required linkage editing between the DTFCN table and a section definition in the relocatable library.) If TYPEFLE=INPUT is

specified in the DTFCN macro, the module generated supplies the logic for both the GET and PUT functions. If TYPEFLE=OUTPUT is specified, the generated module supplies the logic for only the PUT function.

Bytes	Contents	Function
0-15		CCB.
16	X'20' X'08'	COBOL open; ignore option. DTF table address constants relocated by OPENR.
17-19		Address of logic module: GET and PUT logic if TYPEFLE=INPUT; PUT logic if TYPEFLE=OUTPUT.
20	X'03'	DTF type.
21-23		Not used.
24-31	X'09', IOAREA1, X'00', BLKSIZE	CCW.
End of table if RECFORM=FIXUNB and WORKA not specified.		
The following bytes are added to the table if WORKA is specified.		
32-35	DC A(IOAREA1)	Address of I/O area.
36-39	DC F'0'	Register savearea.
40-43	DC F'0'	Register savearea.
End of table if RECFORM=FIXUNB.		
The following bytes are added to the table if RECFORM=UNDEF.		
	DC F'0'	Register savearea.
	DC F'0'	Register savearea.
	DC H'BLKSIZE'	I/O area size.
	DC AL2(BLKSIZE-1)	For input files only.

Figure 4. DTFCN: Console

DTFCN: GET Macro Chart CA

Objective: To read from the system console, that is, to allow a record to be typed in from the console keyboard.

Entry: From a GET macro expansion to the label IJ2xxxxx.

Exit: To the problem program.

Method: Upon entry to the GET routine, the CCW command code is set to hex '0A' for a read operation. If UNDEF is specified in the RECFORM= parameter in the DTFCN macro, the user specified BLKSIZE is moved into the byte count area of the CCW. (If FIXUNB is specified in the RECFORM= parameter in the DTFCN macro, the byte count area of the generated CCW automatically contains a count of 80.) An SVC 0 is then issued to read the record into IOAREA1. If a workarea was not specified in the DTFCN macro, control returns to the problem

program immediately after completion of the I/O operation.

If WORKA=YES is specified in the DTFCN macro, the contents of IOAREA1 are moved to the workarea before control returns to the problem program.

DTFCN: PUT Macro Chart CB

Objective: To write a record on the system console printer.

Entry: From a PUT macro expansion to the label IJ2MXXXX.

Exit: To the problem program.

Method: Upon entry to the PUT routine, the CCW command code is set to hex '09' for a write operation. If a workarea is specified in the DTFCN macro, a test is made, before the routine proceeds, to determine if a previous I/O operation is complete and if not an SVC 7 is issued. If the file definition specifies undefined records (RECFORM=UNDEF parameter in the DTFCN macro) a check is made to determine if the BLKSIZE is greater than the record length, RECSIZE. (The byte count in the generated CCW is automatically equal to the BLKSIZE specified by the user.) If BLKSIZE is greater, the CCW byte count is modified to the value of RECSIZE.

Note: If RECSIZE is greater than BLKSIZE, the output record will be truncated.

In all cases where a workarea is specified, the record is moved from the workarea to IOAREA1 before an SVC 0 is issued to write the record on the console printer. After the SVC 0 is issued, control returns directly to the problem program.

If no workarea is specified, the record to be written is available in IOAREA1, and an SVC 0 is issued directly. The routine then waits for completion of the I/O operation before control is returned to the problem program.

MAGNETIC INK CHARACTER RECOGNITION FILES

Magnetic Ink Character Recognition (MICR) files are input files processed by 1259 or 1412/1419 magnetic ink characters readers. MICR files are defined by a DTFMR macro and the processing logic module is generated by an associated MRMOD macro.

DTFMR Macro

The DTFMR Macro (Figure 5) can generate two types of DTF tables. The type generated for a particular file depends on the ADDRESS= parameter specified by the user in the DTFMR macro. Although the two table types have been combined in this publication, the variation in the entries for ADDRESS=DUAL are noted in the table.

Bytes	Bits	Function
0-5		CCB indicators.
6-7		Logical class and unit numbers. (Primary if DUAL addressing.)
8		Zero.
9-11		CCW address.
12-15		Zeros.
16	0-1	Not used.
	2	COBOL open; ignore option.
	3	Not used.
	4	DTF table address constants relocated by OPENR.
	5-7	Not used.
17-19		Address of logical module.
20		DTF type = X'0B'
21		Logical module option switches.
	0	User disengage 0 = Off; 1 = On
	1	Program sort mode 0 = No; 1 = Yes
	2	First time switch (after engage) 0 = No; 1 = Yes
	3	Addressing = DUAL 0 = No; 1 = Yes
	4	Waiting 0 = No; 1 = Yes
	5	Read logic indicator 0 = No; 1 = Yes
	6	Not used.
	7	Supervisor initial read (after open) 0 = No; 1 = Yes
22-29		Symbolic filename.
30		Open/Close option switch.
	0	Open indicator 0 = Closed; 1 = Open
31-33		Open/Close option switches.
34-35		Logic module option switches.
36-39		Error information status.

Figure 5. DTFMR (Part 1 of 4)

Bytes	Bits	Function
40-41		Length of DTF table.
42-43		Device type indicator.
44-45		Record type.
46-49		Reserved for future use.
50-51		I/O register.
52-55		End of file address.
56-59		IOAREA2/1 address.
60-63		Document buffer size.
64-65		Blocking factor/Number of buffers.
66-67		I/O area size.
68-71		Record length.
72-76		Sense information.
77		Supervisor switch.
78-79		Logical Class and Unit numbers (Secondary--for DUAL addressing only).
80-81		Register alignment bytes.
82-83		Logical Class and Unit numbers (Primary--for DUAL addressing).
84-87		Document buffer size.
88		Command code (4C).
89-91		Address of last byte of first document buffer.
92		Command code (4C).
93-95		Address of last byte of last document buffer.
96-99		Stacker-select routine address.
100-103		Address of stacker select CCW chain.
104-107		Current buffer address pointer (Supervisor).
108-111		Supervisor count.
112-113		Number of buffers (5 for SINGLE, 7 for DUAL).

Figure 5. DTFMR (Part 2 of 4)

Bytes	Bits	Function
114-115		Message indicator.
116-119		ERROPT routine address.
120-121		Logical Class and Unit numbers (Secondary--for DUAL addressing only).
122-123		Reserved for future use.
124-127		Address of last buffer given to user.
128-131		Address of first byte of last buffer.
132-139		Channel status word (CSW).
140-143		Address of active GET record.
144-147		GET counter.
148-159		Reserved for future use.
For SINGLE Addressing		
160-167		CCW - Engage.
168-175		CCW - Read.
176-183		CCW - Sense.
184-191		CCW - NOP.
192-199		CCW - Stacker Select.
200-207		CCW - TIC.
208-215		CCW - Control.
216-223		CCW - BN.
224-231		CCW - Read.
232-239		CCW - Sense.
240-247		CCW - Disengage.
For DUAL Address Adapter		
160-167		CCW - Engage.
168-175		CCW - Read Buffer 1.

Figure 5. DTFMR (Part 3 of 4)

Bytes	Bits	Function
176-183		CCW - Sense.
184-191		CCW - NOP.
192-199		CCW - Read Buffer 2.
200-207		CCW - MOD Sense.
208-215		CCW - Read Buffer 1.
216-223		CCW - MOD Sense.
224-231		CCW - TIC TO NOP.
232-239		CCW - NOP.
240-247		CCW - MOD CTL.
248-255		CCW - Stacker Select.
256-263		CCW - MOD Sense.

Figure 5. DTFMR (Part 4 of 4)

MRMOD Macro

The MRMOD macro provides the necessary logic to process Magnetic Ink Character Recognition (MICR) files. The logic module can handle MICR files with either dual or single addressing.

The functions of the MR logical IOCS module are:

1. To locate the record (document buffer) to be processed (GET/READ).
2. To test for a "document ready for processing" condition (CHECK).
3. To start and stop the feeding of documents through the device (DISEN; disengage---engage is an automatic function of GET or READ).
4. To turn on the pocket lights (LITE).
5. To exit from the partition in a MPS environment, or to enter the wait state in a Batched Job System (BJS) environment, when no useful processing can be performed (WAITF).

The READ, CHECK, WAITF macro combination is provided for users operating more than one 1412/1419 in a given partition of the system. The READ macro performs all the functions of the GET macro except the

inherent wait that occurs when no document is ready for processing.

The READ macro posts an indicator in the current document buffer signifying that the buffer is not ready for processing. The user examines that status of this indicator by issuing the CHECK macro.

The WAITF macro determines if processing can continue. If processing cannot continue, the system enters the wait state if it is operating in a BJS environment, or the system is allowed to continue processing in another partition if it is operating in a MPS environment.

MRMOD: CHECK Macro Charts DA-DC

Objectives: To determine the status of the current document buffer:

1. Buffer ready for processing.
2. Buffer waiting for data.

Entry: From the CHECK macro expansion to the label IJUCHEK.

Exit: To the problem program at one of three locations:

1. The next sequential instruction if the document buffer contains data and is ready for processing.

2. The address specified in the label operand of the macro if the document buffer is waiting for data.
3. The ERROPT address specified in the DTF if the document buffer contains posted error-condition indicators.

Method: The CHECK macro first tests the condition of the wait bit (bit 5 of the first byte) within the current document buffer. If this bit is on, indicating that the device is waiting for a document to be read, a branch is made to the address specified in the second (label) operand of the CHECK macro. This bypasses normal record processing and allows continuation of other processing in the partition. If the second (label) operand is not specified in the CHECK macro, control is passed to the ERROPT address. If neither of these addresses is specified, a branch is made to the next sequential instruction following the CHECK macro expansion.

If the wait bit is off, a second test is made to determine if the Buffer Status Indicator (BSI) bit (bit 0 of the first byte within the current document buffer) is on. If the bit is on, indicating that the data is valid and ready for processing, a branch is made to the next sequential instruction following the CHECK macro expansion where normal processing resumes.

If both bits are off (bits 0 and 5), indicating either that the device is not operational or that the document buffer data is in error, a branch is made to the ERROPT location specified in the DTF for further analysis of the error conditions by the user. If the ERROPT location is not specified, a branch is made to the next sequential instruction following the CHECK macro expansion.

MRMOD: DISEN Macro Chart DD

Objective: To disengage the 1412/1419 MICR device and stop the feeding of documents whenever necessary.

Entry: From a DISEN macro expansion to the label IJUDISEN.

Exit: To the problem program.

Method: The DISEN macro turns on the user disengage indicator (bit 0 of byte 21) in the DTF and checks for the type of address adapter used with the device. If dual

addressing is specified, the NOP instruction in the CCW chain is changed to a disengage instruction, X'DF'. If single addressing is specified, the chaining bit is set on in the CCW chain.

Note: A DISEN macro must be issued before a LITE macro can be issued.

MRMOD: GET Macro Charts DA-DC

Objectives:

1. To provide a pointer, in the user register IOREG, to the next document buffer to be processed in the document buffer area (Figure 6).
2. To issue engage commands to the MICR device when necessary.

Entry: From a GET macro expansion to the label IJUGETCK.

Exit: To the problem program after a document buffer is filled with a stacker-selected document, or after error conditions are posted in the buffer.

Method: When more than three records remain in the document buffer area, the user's I/O register (IOREG) is loaded with the address of the next document buffer to be processed and the previous document buffer is reset to binary zeros. Control then returns to the user.

When three, or fewer than three, records remain in the document buffer area, a test is made to determine if three or more document buffers are empty and it is necessary to engage the MICR device to feed more documents. If it is necessary to issue engage commands, a branch is made to IJUENGCK (Chart DB) where the necessary modifications are made to the CCW chain and the I/O operation executed.

If an engage is not required or after the engage commands are issued, the status of the document buffer is checked. If the buffer is ready for processing, the procedure followed is the same as for more than three records remaining in the I/O area. If the document buffer is not ready for processing, checks are made for errors and flags are set in the buffer as indicators of the error conditions found. This information is then passed to the user for further analysis.

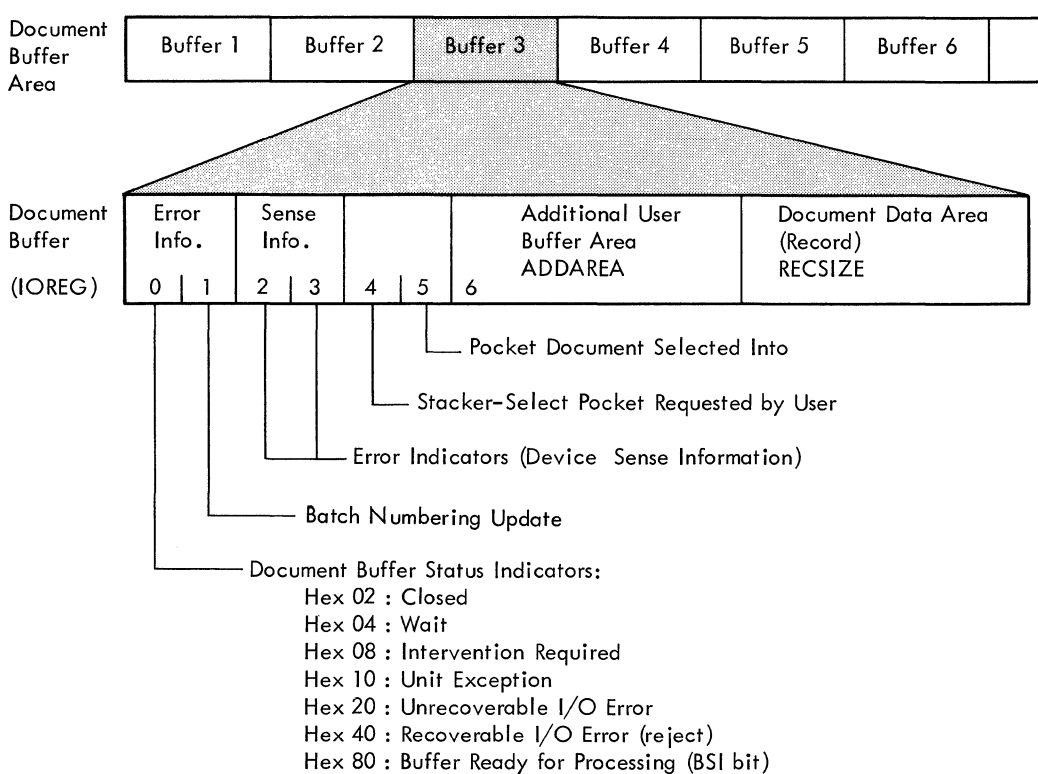


Figure 6. MICR Buffer Format (1412/1419)

MRMOD: LITE Macro Chart DE

Objective: To turn on the pocket lights specified in the pocket light indicator by the user.

Entry: From a LITE macro expansion to the label IJULITE.

Exit: To the problem program.

Prerequisites:

1. A DISEN macro must be issued before the LITE macro is issued.
2. The unit exception bit must be turned on in the CCB in the DTF table before the LITE macro is issued.

Method: The LITE macro checks the pocket light indicator (Figure 7) bit by bit, to determine and turn on the lights specified by the user. After each specified pocket light is turned on, a check is made for the occurrence of an unrecoverable I/O error during the turning on of the light. If an error occurred, bit 15 of the pocket light indicator is set.

Label	Byte 0								Byte 1							
Bit	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Pocket Light	A	B	0	1	2	3	4	5	6	7	8	9	Zeros	Binary	Err	Ind

Figure 7. 1412/1419 Pocket Light Indicator

The pocket light indicators are tested in the sequence: B, A, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0. To turn on a particular pocket light, a pocket code is combined with a stacker select command code.

Example: Assume that the light in pocket B is to be turned on, that is, bit 1 of the pocket indicator is on.

Upon entry to the pocket light routine of the MRMOD, a count of 12 is initially loaded into a register. When the actual light program loop is entered (at location IJUNXTLT) and it is determined that the pocket B light is to be turned on, the count is reduced by one to 11. The 11 is then multiplied by 16. The result is 176, or a hexadecimal B0.

For the single address adapter, the hex B0 is ORed with a hex 0F and results in a command code of hex BF. For the dual address adapter, the hex B0 is ANDeD with a hex F7 and results in a command code of hex B7. After the pocket light is successfully turned on, the initial count (12) is reduced by one and the routine returns to location IJUNXTLT to test for the next pocket light.

MRMOD: READ Macro Charts DA-DC

Objectives:

1. To provide a pointer, in the user register IOREG, to the next document buffer to be processed in the document buffer area (Figure 6).
2. To issue engage commands to the MICR device when necessary.

Entry: From a READ macro expansion to the label IJUREAD.

Exit: To the problem program after a document buffer is filled with a stacker-selected document, or after error conditions are posted in the buffer.

Method: Same as the GET macro except for the inherent wait that occurs when the document buffer is not ready for processing. Instead of issuing a wait SVC (SVC 7), the READ macro posts an indicator in the document buffer (bit 5 of first buffer byte) to signal that the buffer is not ready for processing. The user, after examining the indicator by issuing a CHECK macro, can then transfer processing to another location within the partition until the buffer is ready for processing.

MRMOD: WAITF Macro Chart DF

Objective: To test whether any 1412/1419 MICR devices specified in the macro operand (wait list) are operative and processing should continue, or if all the devices specified are inoperative and IOCS should enter the wait state.

Entry: From a WAITF macro expansion to the label IJUWAITF.

Exit: To the problem program.

Method: The WAITF macro loads a work register with the address of the DTF table for the file specified by the first operand. Tests are then made to determine

if the file is in an operative condition. If the file is operative and ready, processing is resumed at the next sequential instruction following the macro expansion.

Note: When more than one file is specified in the operand of the WAITF macro (wait list), normal processing is resumed after the first operational file is detected, thereby omitting checking of any remaining files in the wait list. To accomplish this, the pointer in the wait list is repeatedly incremented by 4 (the length of each operand) until the end of the wait list is reached (the first nonzero byte). When the end-of-the-wait list has been detected, the pointer (register 14) then contains the address of the next sequential instruction following the macro expansion.

If the file currently being tested is not operational, the address of the DTF table for the next file specified in the macro operand is obtained and the tests are repeated for the next file.

If none of the files specified in the operand are operational, an SVC 29 is issued and the wait state is entered. If the system is operating in an MPS environment, processing is allowed to continue in another partition. When one of the files becomes operational, processing can resume in the partition in which the device is operating.

Initialization and Termination of MICR Files

Magnetic Ink Character Recognition files are one of the very few types of unit record files that are opened and closed by logical transients included in the system to handle a specific file type. These logical transients, \$\$BOMR01 and \$\$BCMR01, are fetched by the Open and Close Monitors respectively (refer to Volume 1).

\$\$BOMR01: Open MICR (Magnetic Ink Character Recognition) Chart DG

Objective: To open 1412/1419 MICR files and to initialize the document buffer area.

Entry: From the Open Monitor, \$\$BOPEN, to the label FRSTINST.

Exit: To the Open Monitor, \$\$BOPEN.

Method: The \$\$BOMR01 routine clears the entire document buffer area and checks for device assignment. It then calculates the Physical Unit Block (PUB) entry address for the device, and determines from the PUB information which entry in the Supervisor table of DTF addresses (PDTABB) is to be used. The address of the DTF table is then inserted into the proper entry in the PDTABB table. Refer to the Program Logic Manual (PLM), IBM System/360 Disk Operating System, Supervisor and Physical and Logical Transients, listed in the Preface of this manual for the format and use of the PDTABB table.

The unit exception bit in the CCB is turned on and the remainder of the DTF is initialized. The open indicator (bit 0 of byte 30) is set on in the DTF to signal that the file is open and the Open Monitor, \$\$BOPEN, is fetched to determine if more files are to be opened.

\$\$BCMR01: Close MICR (Magnetic Ink Character Recognition) Chart DH

Objective: To close 1412/1419 MICR files and to reinitialize the DTFMR table.

Entry: From the Close Monitor, \$\$BCLOSE, to the label FRSTINST.

Exit: To the Close Monitor, \$\$BCLOSE.

Method: The \$\$BCMR01 routine resets the open indicator in the DTF and the traffic (wait) bit in the CCB, and turns off the external line status indicator for the file.

MICR Error Messages

Error conditions occurring on 1412/1419 MICR devices are reported to the operator via a special MICR error message writer logical transient, \$\$BMMR20.

\$\$BMMR20: MICR Message Writer Chart DJ

Objective: To print one of two error messages for MICR devices:

1. 4MR1I - EXTERNAL INTERRUPT I/O ERROR
2. 4MR2I - SCU NOT OPERATIONAL

Entry: From the MR logical IOCS module when an error occurs during a GET or READ.

Exit: To the logical IOCS module after the proper message is printed.

Method: The \$\$BMMR20 message writer routine examines the message indicator in the DTF table (bytes 114-115) to determine the appropriate message. It then extracts information from the DTF and CSW, converts the information to printable form, and inserts it into the message.

The proper message is then printed on SYSLOG and control is returned to the logical IOCS module.

OPTICAL READER FILES

Optical Reader (OR) files are input files associated with 1285/1287 optical readers and defined by a DTFOR macro. An ORMOD macro provides the data handling logic module for this type of file.

DTFOR Macro

The DTF table generated by the DTFOR macro services both the 1285 and the 1287 optical readers. If the device is a 1287, either journal tapes or documents can be processed (Figure 8).

Bytes	Bits	Function
0-15		Dummy CCB.
16	0-1	Not used.
	2	COBOL open; ignore option.
	3	Not used.
	4	DTF table address constants relocated by OPENR
	5-7	Not used.
17-19		Address of logic module.
20		DTF type=X'09'. DTF type=X'0A' (HEADER=YES).
21	0	PIOCS switches. 1 = Open; 0 = Closed.
	1	1 = Input.
	2	1 = Control.
	3	1 = Device is 1285 or 1287.
	4	1 = Header.
	5	Reserved for future use.
	6	1 = RDLNE.
	7	Not used.
22-23		Not used.
24-39		CCB.
40-47		Sense CCW.
48-51		Lost lines (equipment check).
52-55		After 10 retries.
56-59		Wrong length records.
60-63		After 10 retries.
64-67		Keyboard corrections.
68-71		Data checks.
72-75		Lines marked.
76-79		Total lines read.
80		Error indicators.
81	0	LIOCS switches. 1 = First time.
	1	1 = 2 I/O areas.
	2	1 = WORKA=YES
	3	1 = RECFORM=FIXUNB
	4	1 = RECFORM=UNDEF
	5-7	Not used.

Figure 8. DTFOR (Part 1 of 2)

Bytes	Bits	Function
82		Normal command code.
83		Control command code.
84-87		IOAREA2 address.
88-95		Read CCW.
96-103		Go to next line CCW.
104-111		Control CCW.
112-115		EOF address.
116-119		Correction exit address.
120-123		IOAREA1 address.
124-127		DC A(&BLKS-1)
128-129		SR 13,&RECS
130-131		LR &RECS,13
132-133		LR &IOR,13
134-135		Sense.

Figure 8. DTFOR (Part 2 of 2)

ORMOD Macro

The module generated by the ORMOD macro provides the logic to perform the GET, CNTRL, and RDLNE functions for the 1285 and 1287 (tape mode) Optical Reader, and the READ, WAITF, CNTRL, RESCN, and DSPLY functions for the 1287 (Document Mode) Optical Reader.

For 1287 operating in document mode, the logic module handles only unblocked records, and supports fixed and undefined record formats. For both 1285 and 1287, blocked records are also handled if the device is operating in tape mode.

The ORMOD is capable of generating many logic modules, each tailored to specific parameters. The number of different modules that can be generated is so great that it would be impractical to flowchart and describe every possible variation individually. To stay within practical limits, the internals of the OR module are flowcharted and described to indicate all variations.

ORMOD: CNTRL Macro Charts EA-EC

Objective: To execute a control operation for 1285 and 1287 optical reader.

- For 1285 or 1287 operating in tape mode, the control operation is either mark a line or read a complete line from the keyboard.
- For 1287 operating in document mode, the control operation is either eject, eject and stacker-select, stacker-select, or increment the document.

Entry: From a CNTRL macro expansion.

Exits:

- Normal exit from the CNTRL routine is to the problem program.
- To the user's end-of-file routine when an EOF condition is reached.
- To the user's correction routine when a recoverable error occurs.

Method: The CNTRL routine ensures that any previous I/O operation is complete. It then sets the appropriate control command code (supplied by the user in general register 0) in the CCW and causes the control function to be performed.

If the control function is to read a line from the keyboard, the routine checks for a unit exception after the read operation is complete. If the control function is other than read a line from the keyboard, and a unit check occurs, the routine checks for command reject, late stacker-select, or recoverable errors.

If a unit exception or a read keyboard operation occurs, or if a recoverable type error occurs, the CNTRL routine branches to the address supplied in the DTF for the user's correction routine to attempt a recovery from the error before returning control to the problem program.

ORMOD: DSPLY Macro Chart ED

Objective: To cause a complete field of data from a document to be displayed on the display scope of the 1287.

Entry: From the DSPLY macro expansion.

Exit: To the problem program.

Method: The routine modifies the read backward CCW by changing the command code, input address, and chaining flags. It then blanks out the portion of I/O area into which it is reading. An EXCP is then issued for this CCB. When the I/O operation is complete, the routine checks for a unit exception indicating an error was made in keying in corrections. The command is re-executed if necessary. If a unit exception is not present, an exit is made to the problem program.

ORMOD: GET Macro Charts EE-EH and EJ-EN

Objective: To access a record from the 1285 and 1287 (tape mode) optical character readers.

Entry: From the GET macro expansion.

Exit: To the problem program.

Method: The GET routine reads a record from the 1285/1287 optical reader and places it in main storage starting at the rightmost position of the I/O area and continuing to the left until the last character has been read.

If two I/O areas or one I/O area and a workarea are used, I/O and processing overlap is possible. On the first GET operation, one area is filled and another EXCP is issued before any processing is allowed. After the second EXCP is issued, control is returned to the problem program so that processing can begin on the data in the first I/O area. If two I/O areas are specified, the first GET macro instruction sets a switch that changes the logic flow of all the following GET instructions used in the problem program with this logic module. If a workarea is specified, the contents of the I/O area are moved to the workarea. The next EXCP is issued, and an exit to the problem program is made. If RECFORM=UNDEF, the contents of the I/O area are left-justified in the workarea.

ORMOD: RDLNE Macro Chart EP-EQ

Objective: To cause a line of data from a journal tape to be read in the on-line mode while processing is in the off-line correction mode.

Entry: From the RDLNE macro expansion.

Exit: To the problem program.

Method: The routine indicates a keyboard correction, modifies the CCW with a read backward on-line command, and issues an EXCP. When the I/O operation is complete, the routine checks for equipment, wrong-length record, and data checks. If present, these conditions are posted to the DTF, and an exit is made to the problem program.

ORMOD: READ Macro Chart ER

Objective: To access a record or records from the 1287 optical character reader when processing documents.

Entry: From the READ macro expansion.

Exit: To the problem program.

Method: The READ routine reads a record or records from a document by executing the users CCW's. It is the user's responsibility to properly set up the CCW's before the READ macro is issued.

ORMOD: RESCN Macro Charts ES-ET

Objectives: To allow up to nine retries of an unreadable line of data on a document. Also to allow a read in the on-line mode while processing in the off-line correction mode.

Entry: From the RESCN macro expansion.

Exit: To the problem program.

Method: The routine modifies the CCW to allow data checks, and an EXCP is issued. After the I/O operation is complete, the routine checks for and posts equipment checks, data checks, and wrong-length records. The line of data is retried the number of times specified.

When the retries are completed, control returns to the problem program unless forced on-line correction was specified. In the case of forced on-line correction, the CCW is modified to force on-line correction and another EXCP is issued.

ORMOD: WAITF Macro Charts EU-EX

Objective: To provide storage resident error recovery in addition to its normal function of waiting for I/O completion.

Entry: From the WAITF macro expansion.

Exit: To the user specified COREXIT routine if a read error occurred or to the next sequential instruction following the WAITF macro expansion in the problem program if read was error-free.

Method: After checking for completion of the previous I/O, the routine tests for unit check. If unit check occurred, a check is made to determine if the cause of the interrupt is due to hardware or an unreadable character, lost line, etc. If a hardware error has occurred, a transient error routine is called, and a message is printed to the operator. If the error is due to an unreadable character or line, it is retried up to nine times (depending on error), posted to the user and then a branch is taken to the user's COREXIT routine. On a return from COREXIT, the operation is restarted from where the CCW chain was broken. Control is then returned to the problem program.

Initialization and Termination of Optical Reader Files

Optical reader files are opened by the logical transient \$\$BOOR01 that is fetched

by the Open Monitor (refer to Volume 1). These files are closed by the Close Monitor, \$\$BCLOSE, which simply resets the open indicator in byte 21 of the DTF table.

\$\$BOOR01: Open Optical Reader Chart EY

Objective: To open an optical reader file.

Entry: From the Open Monitor (\$\$BOPEN).

Exit: To \$\$BOPEN or cancel.

Method: If the optical reader file contains a header, this phase reads it into IOAREA1. If it does not contain a header, an I/O NOP is performed.

If a valid DTF type was found (indicating the presence or absence of the header), the routine then returns to the Open Monitor (\$\$BOPEN) to determine if any more files need to be opened. The routine aborts the job if an invalid DTF type is present.

PRINTER FILES

Printer (PR) files are defined by a DTFPR macro for continuous forms output files on 1403, 1404, 1443, and 1445 printers. In addition, the Selective Tape List (STL) feature for printing journal tapes on a 1403 is also supported.

Data handling logic for printer files is supplied by the associated PRMOD macro.

Printer files are opened and closed by the Open and Close Monitors, respectively, (refer to Volume 1).

DTFPR Macro

The DTFPR macro (Figure 9) generates a DTF table containing all the necessary information to describe an output printer file on any of the following devices:

- IBM 1403 Printer (with or without the SLI feature)
- IBM 1404 Printer (continuous forms only)
- IBM 1443 Printer
- IBM 1445 Printer

Bytes	Bits	Contents	Function
0-15			CCB.
16	0-1		Not used.
	2		COBOL open; ignore option.
	3		Not used.
	4		DTF table address constants relocated by OPENR.
	5-7		Not used.
17-19			Address of logic module.
20		X'08'	DTF type.
21	0		1 = Open; 0 = Closed.
	1		First time switch.
	2		1 = Control character.
	3		1 = Fixed unblocked records.
	4		1 = Variable unblocked records.
	5		1 = 2 I/O areas.
	6		1 = Workarea.
	7		1 = Print overflow channel 9.
22		X'09'	Normal command code.
23		X'09'	Control command code.
24-27			Address of IOAREA1.
28-31			Bucket. ¹
32-33		LR 12,(RECSIZE) NOPR 0	For undefined records only.
34-37		LA &IOREG,4(14) NOP 0	Only if IOREG=(r).
38-39			Bucket 2. ²
40-47		11,*,X'60',1	CCW--Set up Selective Tape List Control.
48-55			CCW--Print

¹The bucket bytes handle undefined records. Bit 0 of byte 28 at open time determines the mode set of a printer with UCS. If bit 0 is 1, the mode is set so that data checks occur if an invalid character is printed. Otherwise, mode is set to suppress data checks. The use of the UCS parameter determines the setting of this bit.

²The 2 byte bucket saves print overflow conditions if CTLCHR=ASA.

Figure 9. DTFPR

PRMOD Macro

The module generated by the PRMOD macro provides the logic to perform the CNTRL, PRTOV, and PUT functions for a printer file. The logic module handles only

unblocked records, and supports fixed, undefined, and variable record formats.

The PRMOD macro can generate many logic modules, each designed to handle the conditions specified by the macro parameters. Because it is not possible to describe all the variations, the PRMOD is

flowcharted to show the internal decisions made for the specified parameters.

PRMOD: CNTRL Macro Chart FA

Objective: To control the carriage space and skip operations.

Entry: From a CNTRL macro expansion.

Exit: To the next sequential instruction in the problem program following the CNTRL macro expansion.

Method: This routine provides completely independent control of the printer carriage. It is used when the PRMOD macro parameter CTLCHR is not specified.

The PRMOD CNTRL routine waits for a previous I/O operation to finish and then inserts the control character into the CCW command code. The required carriage operation is started and control returns to the problem program without waiting for completion of the carriage operation.

PRMOD: PRTOV Macro Chart FB

Objective: To cause and control an overflow skip.

Entry: From a PRTOV macro expansion.

Exit: To the user's carriage overflow routine if the address of the routine is supplied, or to the problem program at the next sequential instruction following the PRTOV macro expansion.

Method: The PRTOV (printer overflow) macro instruction is used with a logical printer file to specify the operation to be performed on a carriage overflow condition.

PRTOV requires two parameters. The first parameter must be the name of the logical file specified in the DTF header entry. The second parameter must specify the number of the carriage tape channel (9 or 12) used to indicate the overflow condition. When an overflow condition occurs, IOCS skips the printer carriage to channel 1.

An optional third parameter causes a branch to a user routine instead of a skip to channel 1 on an overflow condition. This parameter specifies the symbolic name representing the address of the user's routine. In the user's routine, any desired function can be performed except another PRTOV.

PRMOD: PUT Macro Charts FC-FE

Objective: To print a line and space, or to print a line and skip the appropriate tape, if the Selective Tape Lister feature is available.

Entry: From a PUT macro expansion.

Exits: To the problem program or cancel.

Method: This routine causes a record to be printed on the output device. The logic determines if two I/O areas are used, if a workarea is specified, and if CTLCHR controls the carriage. A test is made to determine if CNTRL is specified. If so, the CTLCHR cannot be used, CTLCHR and CNTRL are mutually exclusive (if one is used, the other cannot be specified).

If the American Standard Association (ASA) CTLCHR is used, this routine translates it to System/360 language.

If the Selective Tape Lister feature is used, selected tapes are controlled through the use of a one-byte control field. This field is accessed by the optional operand, either STLSP=label, or STL SK=label, of the PUT macro. Figure 10 shows the format of the field specified by label.

Bit	0	1	2	3	4	5	6	7
Tape Positions	8	7	6	5	4	3	2	1
<u>Note:</u> Tape 1 is the leftmost tape on the Selective Tape Lister.								

Figure 10. STL Control Field

The control field is used as the data address in the control CCW in the DTF table. Double-width tapes are controlled by using the two bits corresponding to the tape positions occupied.

Note: The label CONCMD used in this logic is a one byte save area in the DTF table used to store the control character.

PAPER-TAPE FILES

Paper-Tape (PT) files are input files associated with 2671 paper-tape reader and defined by a DTFPT macro. A corresponding PTMOD macro supplies the data handling logic module which supports only fixed unblocked and undefined records.

Paper-tape files are opened and closed by the Open and Close Monitors, respectively (refer to Volume 1).

DTFPT Macro

The DTFPT macro is capable of generating several different DTF tables depending on

the parameters specified in the macro operand. The four basic table types are:

- DTFPT: No Translation, No Shifts, or Deletes (Figure 11)
- DTFPT: Translation, No Shifts or Deletes (Figure 12)
- DTFPT: Translation, Shifts, and Deletes with Fixed Unblocked Records (Figure 13)
- DTFPT: Translation, Shifts, and Deletes with Undefined Records (Figure 14)

Translation refers to code translation of paper tapes punched in other than EBCDIC code. Shifts refers to letter or figure shift required for 5-channel paper tape. Deletes refers to acceptance of delete characters punched in the tape.

Bytes	Contents	Function
0-15		CCB (X'08' in byte 2).
16	X'20' X'08'	COBOL open; ignore option. DTF table address constants relocated by OPENR.
17-19		Address of logical module.
20	X'01'	DTF type.
21		Bit 0 Not used. 1 Two I/O areas. 2-5 Not used. 6 Equipment check. 7 Read error.
22-25	MVI 26(1),X'07' NOP 0(0)	NOP first SVC 0 if 2 areas. If 1 area.
26-27	SVC 0	Read a record.
28-29	SVC 0 NOPR 0	Read another record if 2 areas. If 1 area.
30-31	LR &RECSIZE,14 NOPR 0	Put record length in user's register. No RECSIZE entry.
32-35	A(&IOAREA2) A(&IOAREA1)	If 2 areas. If 1 area.
36-39	A(&EOFADDR)	End-of-file address.
40-43	A(&ERROPT) SR 0, 0 SVC 6 B 12(15) B 138(15)	Address of user's error routine if ERROPT=name. ERROPT omitted. ERROPT=SKIP. ERROPT=IGNORE.
44-47	A(&WLRERR) B 12(15) B 152(15) B 152(15)	Address of user's WLR routine if WLRERR=name. WLRERR omitted and ERROPT=SKIP. WLRERR and ERROPT both omitted, or WLRERR omitted and ERROPT=IGNORE. RECFORM=FIXUNB or omitted.
48-55	X'02',&IOAREA1,X'00',&BLKSIZE X'06',&IOAREA1,X'00',&BLKSIZE	CCW: if RECFORM=FIXUNB or omitted. CCW: if RECFORM=UNDEF.
56-63		Duplicate CCW.
64-67	F'0'	Savearea for register 14.
68-71	L &IOREG,48(1) NOP 0(0)	Put input area address into user's register. No IOREG entry.

Figure 11. DTFPT. No Translation, No Shifts or Deletes

Bytes	Contents	Function
0-15		CCB (X'08' in byte 2).
16	X'20' X'08'	COBOL open; ignore option. DTF table address constants relocated by OPENR.
17-19		Address of logical module.
20	X'01'	DTF type.
21		Bit 0 Not used. 1 Two I/O areas. 2-5 Not used. 6 Equipment check. 7 Read error.
22-25	MVI 26(1),X'07' NOP 0(0)	NOP first SVC 0 if 2 areas. If 1 area.
26-27	SVC 0	Read a record.
28-29	SVC 0 NOPR 0	Read another record if 2 areas. If 1 area.
30-31	LR &RECSIZE,14 NOPR 0	Put record length in user's register. No RECSIZE entry.
32-35	A(&IOAREA2) A(&IOAREA1)	If 2 areas. If 1 area.
36-39	A(&EOFADDR)	End-of-file address.
40-43	A(&ERROPT) SR 0, 0 SVC 6 B 12(15) B 138(15)	Address of user's error routine if ERROPT=name. ERROPT omitted. ERROPT=SKIP. ERROPT=IGNORE.
44-47	A(&WLRERR) B 12(15) B 182(15)	Address of user's WLR routine if WLRERR=name. WLRERR omitted and ERROPT=SKIP, WLRERR and ERROPT both omitted, or WLRERR omitted and ERROPT=IGNORE. RECFORM=FIXUNB or omitted.
48-55	X'02',&IOAREA1,X'00',&BLKSIZE X'06',&IOAREA1,X'00',&BLKSIZE	CCW: if RECFORM=FIXUNB or omitted. If RECFORM=UNDEF.
56-63		Duplicate CCW.
64-67	A(&TRANS)	Address of user's translate table.
68-71	L &IOREG,32(1) NOP 0(0)	Put input area address into user's register. No RECSIZE entry.
72-75	LH &RECSIZE,30(1) NOP 0(0)	Put record length into user's register. No IOREG entry.

Figure 12. DTFPT. Translation, No Shifts or Deletes

Bytes	Contents	Function
0-15		CCB (X'00' in byte 2).
16	X'20' X'08'	COBOL open; ignore option. DTF table address constants relocated by OPENR.
17-19		Address of logical module.
20	X'01'	DTF type.
21		Bit 0 Not used. 1 Two I/O areas. 2-6 Not used. 7 EOF indicator.
22-25	MVI 26(1),X'07' NOP 0(0)	NOP first SVC 0 if 2 areas. If 1 area.
26-27	SVC 0	Read a record.
28-29	SVC 0 NOPR 0	Read another record if 2 areas. If 1 area.
30-31	H'0'	Record length field.
32	C'02'	
33-35	AL3(&IOAREA2) AL3(&IOAREA1)	If 2 areas. If 1 area.
36-39	A(&EOFADDR)	End-of-file routine address.
40-43	A(0)	
44-47	NOP 0	
48-55	X'02',&IOAREA1,X'00',&OVBLKSZ X'02',&IOAREA1,X'00',&BLKSIZE	CCW: if OVBLKSZ specified. If OVBLKSZ not specified.
56-59	L &IOREG,96(1) NOP 0(0)	Put input area address into user's register. No IOREG entry.
60-63	LH &RECSIZE 30(1) NOP 0(0)	Put record length into user's register. No RECSIZE entry.
64	X'02'	
65-67	AL3(&IOAREA1)	Address of current input area.
68-71	A(0)	Address of remainder in input area.
72-75	F'0'	Length of remainder.
76-79	A(&FTRANS)	Address of figure shift table.

Figure 13. DTFTP. Translation, Shifts and Deletes, With Fixed Unblocked Records (Part 1 of 2)

Bytes	Contents	Function
80-83	A(<RANS)	Address of letter shift table.
84-87	A(&TRANS) A(&FTRANS) A(0)	If LTRANS & FTRANS not specified. Address of current translate table. If no translation is specified.
88-91	A(&SCAN)	Address of scan table.
92-95	F'&BLKSIZE'	Required record length.
96-103		Duplicate CCW.
104-109	TR 0(0,4),0(5) TM 0,0 NOPR 0	If TRANS, LTRANS or FTRANS specified. If no translation specified.

Figure 13. DTFPT. Translation, Shifts, and Deletes, with Fixed Unblocked Records (Part 2 of 2)

Bytes	Contents	Function
0-15		CCB (X'08' in byte 2).
16	X'20' X'08'	COBOL open; ignore option. DTF table address constants relocated by OPENR.
17-19		Address of logical module.
20	X'01'	DTF type.
21		Bit 0 Not used. 1 Two I/O areas. 2-5 Not used. 6 Equipment check. 7 Read error.
22-25	MVI 26(1),X'07' NOP 0(0)	NOP first SVC 0 if 2 areas. If 1 area.
26-27	SVC 0	Read a record.
28-29	SVC 0 NOPR 0	Read another record if 2 areas. If 1 area.
30-31	H'0'	Record length field.
32-35	A(&IOAREA2) A(&IOAREA1)	If 2 areas. If 1 area.
36-39	A(&EOFADDR)	End of file routine address.
40-43	A(&ERROPT) SR 0,0 SVC 6 B 16(15) B 246(15)	Address of user's routine if ERROPT=name. ERROPT omitted. ERROPT=SKIP. ERROPT=IGNORE.
44-47	A(&WLRERR) B 16(15) B 260(15)	Address of user's WLR routine if WLRERR=name. WLRERR omitted and ERROPT=SKIP. WLRERR omitted and ERROPT=IGNORE omitted.
48-51	L &IOREG,32(1) NOP 0(0)	Put input area address into user's register. No IOREG entry.
52-55	LH &RECSIZE,30(1) NOP 0(0)	Put record length into user's register. No IOREG entry.
56-59	A(&FTRANS)	Address of figure shift table.
60-63	A(LTRANS)	Address of letter shift table.
64-67	A(&FTRANS) A(&TRANS) A(0)	Address of current translate table. If LTRANS and FTRANS not specified. If no translation specified.

Figure 14. DTFPT Translation, Shifts, and Deletes, with Undefined Records
(Part 1 of 2)

Bytes	Contents	Function
68-71	A(&SCAN)	Address of scan table.
72-79		Duplicate CCW.
80-87	X'06',&IOAREA1,X'00'&BLKSIZE	CCW.
88-93	TR 0(0,4),0(5) TM 0,0 NOPR 0	If TRANS, LTRANS or FTRANS specified. If no translation specified.

Figure 14. Translation, Shift, and Deletes, with Undefined Records
(Part 2 of 2)

PTMOD Macro

The PTMOD macro can generate four logic modules providing the GET function for paper-tape files. The particular module generated depends on TRANS= and SCAN= macro parameters. The TRANS= parameter specifies whether or not the generated module contains the logic needed to translate nonshifted punched paper-tape characters into EBCDIC code. The SCAN= parameter specifies whether or not the generated module contains the logic to handle records containing 5-channel shift characters or delete characters.

All modules support the use of either 1 or 2 I/O areas. Decision blocks are included in the flowcharts to indicate the procedure followed for both 1 and 2 I/O areas.

PTMOD: GET Macro, No Translation Chart GA

Objective: To read a record from paper tape punched in EBCDIC code.

Entry: From a GET macro expansion.

Exit: To the problem program, or to the user's EOF routine.

Method: To support the use of either 1 or 2 I/O areas, the GET routine makes use of three key instructions contained in the DTF table. The three instructions, repeated here for convenience, are:

	1 Input Area	2 Input Areas
IJE1MOVE	NOP 0	MVI IJE1SVC1,X'07' (NOP first SVC 0 instruction)
IJE1SVC1	SVC 0	SVC 0 (first)
IJE1SVC2	NOPR 0	SVC 0 (second)

If only one I/O area is specified for the file, the GET routine uses the 'first' SVC 0 instruction to read a record from the paper-tape device. After the complete record has been read into the I/O area, the I/O area address in the CCW and the I/O area address in the DTF table are exchanged. (In this case, the exchange is meaningless because both addresses are the same when only one I/O area is used.) The record is then checked for errors and control is returned to the problem program.

If two I/O areas are specified for the file, the GET routine uses the 'first' SVC 0 instruction to read the first record only. After the first record is read, the first SVC 0 instruction is made a NOP by the execution of the instruction contained in the DTF table at the label IJE1MOVE. The first SVC 0 instruction is only restored if it is needed to handle error conditions. After checking for I/O complete and reading a complete record into the I/O area, the addresses of the two I/O areas are exchanged, and the record is checked for errors. The GET routine then NOPS the first SVC 0 instruction and uses the 'second' SVC 0 instruction to read the next record. From this point on, only the second SVC 0 instruction is used exclusively, except for handling error condition. The GET routine returns control to the problem program without waiting for

completion of the I/O operation started by the second SVC 0 instruction.

PTMOD: GET Macro, Translation of Shifted Code Chart GB

This GET routine functions the same as the GET routine for no translation. It differs only in the logic included to accomplish the required translation once a complete record has been read into the I/O area.

PTMOD: GET Macro, Translation, Shifted Code, Fixed Unblocked Records Charts GC-GD

The basic module is the same as the GET with Translation of Shifted Code except for the method of obtaining the correct number of bytes to complete the fixed-length record. Because the number of control characters scattered throughout the tape record is unpredictable, a routine is included to ensure that the correct number of valid characters is included. The user specifies the number of characters to be read in (OVBLKSZ) to produce a specified number of bytes (BLKSIZE) after the control characters have been deleted.

For example: a group of 70 punched-tape characters (OVBLKSZ) are to be read to

produce a fixed-length record of 50 characters (BLKSIZE). However, because excessive delete characters are encountered, the record contains only 45 characters. To reduce the number of physical I/O operations, a second group of 25 punched tape characters is read to ensure that the 5 characters needed to fill the original record are obtained. On the next GET, the remaining characters (if any) from the second group of punched-tape characters are moved into the leftmost positions of the input area. The byte count and the data address in the CCW are modified accordingly before the next SVC 0 is issued.

PTMOD: GET Macro, Translation and Shifted Code for Undefined Records Charts GE-GF

This GET routine for undefined records functions in much the same manner as the GET routine for fixed-length records. The routine differs in the use of two translate tables; one table is required for letter shift, and the other for figure shift.

Shift characters and delete characters contained in the punched paper tape are removed from the record as translation takes place. The record in main storage is thereby compressed and left-justified.

MAGNETIC TAPE FILES

Magnetic Tape (MT) files are associated with IBM 2400 series magnetic tape drives and are defined for logical IOCS by a DTFMT macro. Files so defined can be either input or output data files, or work files. The function of a particular file is determined, by the user, in the TYPEFLE= parameter in the operand of the DTFMT macro.

Magnetic tape files can also be defined for physical IOCS if the user intends to use physical IOCS macros such as EXCP, WAIT, etc. These files are defined by a DTFPH macro.

In addition, magnetic tape system files can be defined by the device independent macros, DTFDI and DTFCP. These files are described under Device Independent Files later in this manual.

The data handling logic modules for files defined for logical IOCS by the DTFMT macro are provided by the associated MTMOD macro.

Magnetic tape files are opened and closed by logical transient routines fetched by the Open and Close monitors respectively (refer to Volume 1). The open routines provide procedures for checking each file before any records are processed. The close routines provide procedures to terminate the use of each file after all records are processed.

DTFMT MACRO

DATA FILES

The DTFMT macro used to define magnetic tape data files can generate three basic DTF tables, one for each of the following record types:

- Fixed length
- Variable length
- Undefined

The three basic DTF tables have been combined into a single DTFMT table, Figure 15, in this publication. The values included in the table for a specified record type are noted.

WORKFILES

Figure 16 illustrates the DTF table that is generated when the TYPEFLE=WORK parameter is specified by the user in the operand of a DTFMT macro.

DTFPH MACRO (MAGNETIC TAPE)

Figure 17 is generated by the DTFPH macro when the parameter DEVICE=TAPE is specified in the macro operand. The table contains the information necessary to define a magnetic tape file for physical IOCS.

Bytes	Bits	Contents ¹	Function	Record ² Length
0-15			CCB.	
16	0-1		Not used.	
	2		COBOL open; ignore option.	
	3		Not used.	
	4		DTF table address constants relocated by OPENR.	
	5-7		Not used.	
17-19			Address of logic module.	
20		X'11'	Nonstandard or unlabeled.	
		X'12'	Standard labeled, output.	
		X'13'	Standard labeled, input, backwards.	
		X'14'	Standard labeled, input, forwards.	
21	0		First time switch.	
	1		1 = blocked. 0 = unblocked.	
	2		1 = 2 I/O areas. 0 = 1 I/O area.	
	3		1 = workarea, 0 = no workarea.	
	4		1 = input, 0 = output	
	5		1 = backwards. 0 = forwards.	
	6		1 = checkpoint. 0 = no checkpoint.	
	7		1 = TRUNC required during Close.	
22-29			Symbolic filename.	
30			Same as OP code in CCW. (X'01', X'02', or X'0c').	

Figure 15. DTFMT: Data Files (Part 1 of 8)

Bytes	Bit	Contents ¹	Function	Record ² Length
31	0-4		(Not used in Versions 1 and 2)	
			Bits 0-4 are used as displacements by OPEN to determine the location of variable fields of the DTF.	
		B'01110'	Input.	F
		B'01100'	Output.	F
		B'10001'	Input.	V
		B'01111'	Output.	V
		B'01101'	Input.	U
		B'01011'	Output.	U
	5		1 = Tape label information included in DTF (see bytes 88-95). 0 = Tape label information not included in DTF.	
	6		Used by COBOL.	
	7		1 = Header label and EOVS information wanted. 0 = No header label and EOVS information wanted.	
32	0		Standard labels: 1 = yes, 0 = no.	
	1		Labels: 1 = nonstandard, 0 = unlabeled.	
	2		Rewind unload: 1 = yes, 0 = no.	
	3		Rewind option: 1 = no rewind, 0 = rewind.	
	4		Drive direction: 1 = backwards, 0 = forward.	
	5		User label address: 1 = yes, 0 = no.	
	6		Tapemark option: 1 = no, 0 = yes.	
	7		EOF-EOVS switch (used by IBM SORT): 1 = yes, 0 = no.	

Figure 15. DTFMT: Data Files (Part 2 of 8)

Bytes	Bits	Contents ¹	Function	Record ² Length
33-35			User label routine address.	
36	0		DTFPH: 1 = yes, 0 = no.	
	1		COBOL indicator: 1 = yes, 0 = no.	
	2		File type: 1 = input, 0 = output.	
	3		FEOV switch: 1 = yes, 0 = no.	
	4		EOF-EOV switch (output): 1 = EOF, 0 = EOVS.	
	5		Open indicator: 1 = open, 0 = closed.	
	6		1 = variable records.	
	7		1 = undefined records.	
37-39			EOF address.	
40-43			Block count.	
44-47 ¹		BXH 11,12,24(15)	Forward.	F
		BXLE 11,12,24(14)	Backward.	F
		L &VARBLD,DEBLOCKER		V
		DC F'0'		U
48-51		LA 14,1(14)		F,V
		BCTR 14,0+NOPRO	Backward.	F,V
		L &RECSIZE,DEBLOCKER1	If RECSIZE given.	U
		NOP 0(0)	For input if not NOP.	U
52-55		L &IOREG,DEBLOCKER1	If IOREG specified.	F,V
		L &IOREG,DEBLOCKER2	If IOREG specified.	U
		NOP 0(0)	If no IOREG.	
56-63			CCW.	

Figure 15: DTFMT: Data Files (Part 3 of 8)

Bytes	Bits	Contents ¹	Function	Record ² Length
64-67		DC A(IOAREA1)	One I/O area.	
		DC A(IOAREA1+BLKSIZE-1)	One I/O area, read backward	
		DC A(IOAREA2)	Two I/O areas.	
		DC A(IOAREA2+BLKSIZE-1)	Two I/O areas read backward	
68-71		DC F'0'	Input.	F
		DC A(IOAREA1+BLOCKSIZE -RECSIZE)	Input backward: DEBLOCKER1.	F
		DC A(IOAREA1)	1 I/O area, output: DEBLOCKER1.	F
		DC A(IOAREA2)	2 I/O areas, output: DEBLOCKER1.	F
		DC F'BLKSIZE'	DEBLOCKER2.	V
		DC A(IOAREA1)	1 I/O area: DEBLOCKER2.	U
		DC A(IOAREA2)	2 I/O areas: DEBLOCKER2.	U
72-75		DC F'RECSIZE'	Forward: DEBLOCKER2.	F
		DC F'-RECSIZE'	Backward: DEBLOCKER2.	F
		DC A(IOAREA1)	1 I/O area: DEBLOCKER3.	V
		DC A(IOAREA2)	2 I/O areas: DEBLOCKER3.	V
		LA 14,1(14)	Forward.	U
		BCTR 14,0+NOPRO	Backward.	U
76-79		DC F'0'	Input.	F
		DC A(IOAREA1+BLKSIZE -RECSIZE)	Input backwards: DEBLOCKER3.	F
		DC A(IOAREA1+BLKSIZE-1)	Output, 1 I/O area: DEBLOCKER3.	F
		DC A(IOAREA2+BLKSIZE-1)	Output, 2 I/O areas: DEBLOCKER3.	F
		DC F'0'	DEBLOCKER4.	V
		DC Y(BLKSIZE)+Y(BLKSIZE-1)	(Versions 1 and 2 only)	U
		DC Y(BLKSIZE)	(Bytes 76-77 only)	U
	DC Y(BLKSIZE-1)	(Bytes 78-79 only)	U	

Figure 15. DTFMT: Data Files (Part 4 of 8)

Bytes	Bits	Contents ¹	Function	Record ² Length
80-83		DC Y(BLKSIZE)+Y(BLKSIZE-1)	Forward.	F
		DC &(BLKSIZE)+Y(BLKSIZE+1)	Backward.	F
		DC F'0'	DEBLOCKER5.	V
		LR 12,RECSIZE	(Bytes 80-81 only.)	U
		DC H'0'	(Bytes 82-83 if Input- Versions 1 and 2 only.)	V
		DC H'0'	(Bytes 82-83.)	U
84-87		DC Y(RECSIZE-1)	(Bytes 84-85.)	F
		DC H'0'	(Bytes 86-87 if Input- Versions 1 and 2 only.)	F
		DC H'0'	(Bytes 86-87) Output, standard labels.	F
		DC A(IOAREA1+4)	1 I/O area: DEBLOCKER1.	V
		DC A(IOAREA2+4)	2 I/O areas: DEBLOCKER1.	V
		DC 2X'00'	(Bytes 84-85 output only) Reserved for OPEN in Versions 2.1 and 3 only.	U
		B 24(15)	Input only, ERROPT omitted.	U
		B 24(15)	Input only, ERROPT=SKIP.	U
		B 28(15)	Input only, ERROPT=IGNORE.	U
		DC A(ERROPT)	(Versions 1 and 2 Input only if ERROPT=ADDRESS.)	U
88-91		DC A(WLRERR)	Input only WLRERR=ADDRESS.	For Fixed Length Records Only
		B 24(15)	Input only, WLRERR omitted and ERROPT=SKIP.	
		B 28(15)	Input only, WLRERR omitted and ERROPT=IGNORE or omitted.	
		DC 2X'00'	Versions 2.1 and 3 Output only, standard labels (bytes 88-89), reserved for OPEN.	
90-95		File Serial Number	Standard labels, output only (Versions 2.1 and 3).	

Figure 15. DTFMT: Data Files (Part 5 of 8)

Bytes	Bits	Contents ¹	Function	Record ² Length
92-95		DC A(ERROPT) B 24(15) B 24(15) B 28(15)	Input only, ERROPT=ADDRESS. Input only, ERROPT omitted. Input only, ERROPT=SKIP. Input only, ERROPT=IGNORE.	For Fixed Length Records Only
End of Version 1 and Version 2 DTFMT table for fixed length records.				
96-103		Volume Sequence Number DC 2X'00' File Serial Number File Sequence Number	Standard labels, output only (bytes 96-99). Standard labels, input only (bytes 96-97), reserved for OPEN. Standard labels, input only, (bytes 98-103). Standard labels, output only, (bytes 100-103).	
104-107		Volume Sequence Number	Standard labels, input only.	
108-111		File Sequence Number	Standard label, input only.	
88-91		DC F'0'	DEBLOCKER6.	For Vari- able Length Only
92-95		DC Y (BLKSIZE)+ Y (BLKSIZE-1) DC Y(BLKSIZE-4)+ YY(BLKSIZE-1)	Input, Versions 1 and 2. Output, Versions 1 and 2.	
92-93		DC Y(BLKSIZE) DC Y(BLKSIZE-4)	Input only. Output only.	
94-95		DC Y(BLKSIZE-1)		
96-97		DC Y(RECSIZE-1)		
98-99		DC H'0'		
100-103		DC A(WLRERR) B 24(15) B 32(15) DC 2X'00'	Input only, WLRERR=ADDRESS. Input only, WLRERR omitted and ERROPT=SKIP. Input only, WLRERR omitted and ERROPT=IGNORE or omitted. Versions 2.1 and 3 Output only, standard labels, (bytes 100-101) reserved for OPEN.	

Figure 15. DTFMT: Data Files (Part 6 of 8)

Bytes	Bits	Contents ¹	Function	Record ² Length	
102-107		File Serial Number	Standard labels, output only (Versions 2.1 and 3).	For Vari- able Length Only	
104-108		DC A(ERROPT) B 24(15) B 24(15) B 28(15)	Input only, ERROPT=ADDRESS. Input only, ERROPT omitted. Input only, ERROPT=SKIP. Input only, ERROPT=IGNORE.		
End of Version 1 and Version 2 DTFMT table for variable length records.					
108-111		Volume Sequence Number	Standard labels, output only.		
112-115		File Sequence Number	Standard labels, output only.		
112-113		DC 2X'00'	Standard labels, input only, reserved for OPEN.		
114-119		File Serial Number	Standard labels, input only.		
120-123		Volume Sequence Number	Standard labels, input only.		
124-127		File Sequence Number	Standard labels, input only.		
Version 1 and Version 2 DTFMT tables for undefined records end at byte 87.					
86-91		File Serial Number	Standard labels, output only.	For Un- defined Records Only	
88-91		DC A(WLRERR) B 24(15) B 28(15)	Input only, WLRERR=ADDRESS Input only, WLRERR omitted and ERROPT=SKIP. Input only, WLRERR omitted and ERROPT=IGNORE or omitted.		

Figure 15. DTFMT: Data Files (Part 7 of 8)

Bytes	Bits	Contents ¹	Function	Record ² Length
92-95		Volume Sequence Number	Standard labels, output only.	For Un- defined Records Only
		DC 2X'00'	Standard labels, input only (bytes 92-93), reserved for OPEN.	
94-99		File Serial Number	Standard labels, input only.	
		File Sequence Number	Standard labels, output only (Bytes 96-99).	
104-107		File Sequence Number	Standard label, input only.	

¹The format of the tape data file DTF table is different starting at byte 44. The location indicated by the numbers in the left hand column can contain only one of the factors listed under Contents. The factor used for any given DTF table is determined by whether the file record format is fixed, variable, or undefined, and by other DTF parameters as indicated. A blank in the record column indicates that the contents apply to all record types.

²Record Length Explanation

- F = Fixed Record
- V = Variable
- U = Undefined

The deblockers are scratch areas used by the modules to save data from one GET/PUT macro instruction to another. In the text and listings, they are referred to by names DEBLOCKER1 to 6. These are not labels; they are comments used to make it easier to follow in the listings.

Figure 15. DTFMT: Data Files (Part 8 of 8)

Bytes	Bits	Function
0-15		CCB.
16	0-1	Not used.
	2	COBOL open; ignore option.
	3	Not used.
	4	DTF table address constants relocated by OPENR.
	5-7	Not used.
17-19		Address of logic module.
20		DTF type = X'10'.
21	0	1 = no rewind.
	1	1 = rewind unload.
	2	1 = workfile.
	3	1 = read backward.
	4	1 = write.
	5	1 = POINTW.
	6	1 = force checking of read or write.
	7	1 = forward-space file before next operation.
22-23		Not used.
24-25		Record length.
26-27		Maximum BLKSIZE.
28		Read Op code (X'02' for read forward, X'0C' for read backward).
29-31		EOF address.
32-39		CCW.
40-43		Block count, initialized 00000000 for read forward, 00400000 for read backward.
44	0	1 = error routine.
	1	1 = ignore.
	2	1 = read next record switch.
	3	1 = record fixed unblocked.
	4-7	Not used.
45-47		DC A(ERROPT) Address of error routine.

Figure 16. DTFMT: Workfiles

Bytes	Bits	Contents	Function
0-15			CCB.
16	0-1		Not used.
	2		COBOL open; ignore option.
	3		Not used.
	4		DTF table address constants relocated by OPENR.
	5-7		Not used.
17-19		3X'00'	
20		X'12'	Standard labeled, output
		X'14'	Standard labeled, input, forward.
21	0-3		Not used.
	4		1 = input, 0 = output
	5-7		Not used.
22-29			Symbolic filename.
30			Not used.
31	0-4	B'01100'	Used as displacement by OPEN.
	5		1 = Not Version 1 or 2 type table.
			0 = Version 1 or 2 type table.
	6-7		Reserved.
32	0		1 = standard labels.
	1-2		Not used.
	3		1 = no rewind.
	4		Not used.
	5		User label address:
	6-7		1 = yes, 0 = no
33-35			User label routine address.
36	0		1 = DTFPH table.
	1		Not used.
	2		File switch: 1 = input, 0 = output.
	3		Not used.
	4		1 = EOF switch.
	5-7		Not used.

Figure 17. DTFPH: Magnetic Tape (Part 1 of 2)

Bytes	Bits	Contents	Function
37-39			User label exit (not used in Versions 1 and 2)
40-43		DC F'0'	Reserved for OPEN (not used in Versions 1 and 2).
End of table if Version 1 or 2.			
44-87			EOV Routine
88-89		DC 2X'00'	Reserved for OPEN
90-95		File Serial number	
96-99		Volume Sequence number	
100-103		File Sequence number	

Figure 17. DTFPH: Magnetic Tape (Part 2 of 2)

MTMOD MACRO

The MTTMOD macro provides the logic modules required to process both magnetic tape data files and workfiles. In all, four distinct modules can be generated; three for data files with various record formats, and one for workfiles. These modules supply the logic necessary to support all the logical IOCS imperative macros used with magnetic tape files.

Because some of the LIOCS imperative macros are used with both data files and workfiles and some are used only with one file type, the magnetic tape modules have been combined and flowcharted to illustrate the functions of the individual imperative macros.

DATA FILES

A separate logic module is generated by the MTTMOD macro for data files containing records in each of the following formats:

1. Fixed, blocked and unblocked.
2. Variable, blocked and unblocked.
3. Undefined.

These modules provide the logic to perform the functions of the LIOCS imperative macros CNTRL, GET, PUT, RELSE, and TRUNC.

Note: The last two imperative macros are not used with undefined records.

WORKFILES

The workfile module is generated by the MTTMOD macro, when TYPEFLE=WORK is specified, to allow a tape drive to be used for both input and output in one program. Although usually considered as an output file first, the function of a workfile can be switched at any time, thereby allowing records to be written on or read from the same file.

The logic of the workfile module uses the imperative macros READ and WRITE, instead of the GET and PUT used by data files, to transfer records between the device and main storage. The workfile module also differs from the data file modules in that neither blocked records nor switching of I/O areas is supported.

In addition to the READ and WRITE imperative macros, the workfile module supports the NOTE, POINTS, POINTR, POINTW, and CHECK imperative macros. CHECK must always be issued after a READ or WRITE if the user wishes to ensure completion of the operation before issuing another instruction. NOTE is used with the POINTR and POINTW macros to position the tape to a predetermined record. The POINTS macro instruction repositions the tape at the beginning of a file and sets an indicator which causes the tape to be spaced over any tapemark or header labels when the next READ or WRITE macro is issued.

MTMOD: CHECK Macro, Workfile Chart HA

Objective: To ensure that a previously issued READ or WRITE macro has been satisfactorily completed.

Entry: From a CHECK macro expansion to the label IJFWCHEK.

Exits: To the problem program, or to the user's EOF or ERROPT routine.

Method: This macro instruction waits for the completion of the input/output operation started by a READ or WRITE macro instruction. If the I/O operation is completed without detecting an error or an exceptional condition, CHECK returns control to the next sequential instruction following the CHECK macro expansion in the problem program. If the operation results in a read error, CHECK processes the user's option specified in the ERROPT entry of the DFT. If CHECK finds an end-of-file condition, control passes to the user's end-of-file routines.

MTMOD: CNTRL Macro Chart HB

Objective: To execute a nondata physical operation on the file.

Entry: From a CNTRL macro expansion to the label:

IJFFCNT for files containing fixed-length records.

IJFUCNT for files containing undefined records.

IJFVCNT for files containing variable-length records.

Exit: To the problem program.

Method: The CNTRL macro is available to the user for nondata operations on the file. For a magnetic tape file, these operations are: backspace, write a tapemark, forward space file, etc.

The control routine first checks to ensure that any previous I/O operation has been completed. Then the routine sets the control operation code in the CCW, and causes the control function to be performed. The original operation code is restored in the CCW, and, when I/O complete is indicated, the routine returns to the problem program.

MTMOD: CNTRL Workfile Chart HC

Objective: To perform a nondata physical operation.

Entry: From a CNTRL macro expansion to the label IJFWCTL.

Exit: To the problem program.

Method: This routine is for nondata operations on the file. For magnetic tape these operations are: backspace, forward space, rewind, write a tapemark, etc.

The routine puts the control operation code in the CCW, causes the control function to be performed, waits for the I/O operation to finish, and return control to the problem program.

MTMOD: FEOV Macro Chart HD

Objective: To cause an EOVS to occur before a true EOVS condition is reached.

Entry: From an FEOV macro expansion to the label:

IJFFFEV for files containing fixed length records.

IJFUFEV for files containing undefined records.

IJFVFEV for files containing variable length record.

Exit: To the problem program.

Method: The FEOV macro causes an end of volume condition to occur before the physical end of volume is reached. On an input file, the trailer labels are not checked and the user specified rewind option is executed. For an output file, if standard labels are specified, a trailer label is created, and the user rewind option is executed.

This routine sets the FEOV switch on in the DTF table and calls \$\$BCEOV1 to close the file and execute the user specified rewind option. If no rewind option is specified, a message is issued indicating the tape should be rewound and unloaded.

MTMOD: GET Macro Charts HE-HG

Objective: To access a logical record from a magnetic tape data file.

Entry: From a GET macro expansion to the label:

IJFFGET for files containing fixed-length records.

IJFUGET for files containing undefined records.

IJFVGET for files containing variable-length records.

Exit: To the problem program or to the EOF address.

Method: The GET routine accesses the next sequential logical record in a magnetic tape data file. The record is moved from the I/O area to the workarea if WORKA=YES is specified in the DTF. Otherwise, the record is available in the I/O area. Overlap occurs if two I/O areas are specified.

The routine performs deblocking, if required (for variable-length records, the user specified blocksize is obtained first), and then goes to a read/write subroutine. The read/write subroutine checks for first time, two I/O areas, checkpoint records, etc, and performs the required I/O operation. The GET routine next branches to a workarea subroutine. The workarea subroutine checks for a workarea. If a workarea has been specified, the contents of the I/O area are moved into the workarea. If a workarea has not been specified, the address of the current I/O area is loaded into the user's IOREG. For variable-length and undefined records, the record size is calculated and made available to the user. The GET routine then returns control to the problem program.

MTMOD: NOTE Macro, Workfile Chart HH

Objective: To put the tape block count in register 1.

Entry: From NOTE macro expansion to the label IJFWNOTE.

Exit: To the problem program.

Method: The NOTE macro instruction obtains the identification of the last physical record that was read or written in the specified file. The user must insure that the last operation was completed satisfactorily by issuing a CHECK macro before issuing a NOTE. The record identification is put in register 1. This identification is a three-byte block count. The user is responsible for saving the

block count to be used with a POINT macro at some later time.

MTMOD: POINTR Macro, Workfile Chart HJ

Objective: To position the tape to read a record previously identified by a NOTE macro.

Entry: From a POINTR macro expansion to the label IJFWPNTR.

Exit: To the problem program.

Method: This macro instruction repositions the file to read a record previously identified by a NOTE macro. The block count previously supplied by the NOTE macro is used by the POINTR macro to position the tape to the specified record on the file.

The block count is updated in the DTF and compared with a count that was saved by the user from a previous NOTE macro. Then, the tape is spaced backward or forward, and the block count in the DTF is correspondingly decreased or increased until it is equal to the NOTE block count.

When the equal condition is sensed, and if read backward is not specified in the DTF, the tape is backspaced one block. The tape is now positioned to read the desired record.

MTMOD: POINTS Macro, Workfile Chart HK

Objective: To position the tape at the beginning of the file for the next READ or WRITE macro.

Entry: From a POINTS macro expansion to the label IJFWSPNT.

Exit: To the problem program.

Method: The POINTS macro instruction repositions a tape file to the beginning of the file. When a POINTS is issued to the tape file, the forward space file switch, bit 7 of byte 21 in the DTF, is set on and the tape is rewound. If header labels are present, they are bypassed when the next READ, WRITE, POINTR, or POINTW macro is issued.

MTMOD: POINTW Macro, Workfile Chart HL

Objective: To position the tape to write a record following a record identified by a previous NOTE macro.

Entry: From a POINTW macro expansion to the label IJFWPNTW.

Exit: To the problem program.

Method: The POINTW macro repositions the file to write a record after the record identified by a NOTE macro. The block count supplied by a previous NOTE macro is used by the POINTW to position the tape to write a record immediately after the one identified by the NOTE macro.

The block count is updated in the DTF and compared with a count that was saved by the user from a previous NOTE macro. Then the tape is spaced backward or forward, and the block count in the DTF is correspondingly decreased or increased until it is equal to the NOTE block count. The tape is then in position to WRITE the desired record.

MTMOD: PUT Macro Charts HM-HN

Objective: To write a logical record on a magnetic tape file.

Entry: For a PUT macro expansion to the label:

IJFFPUT for files containing fixed-length records.

IJFUPUT for records containing undefined records.

IJFVPUT for files containing variable-length records.

Exit: To the problem program

Method: The PUT routine writes a record in the next sequential location on the file. If a workarea is specified, the PUT routine moves the contents of the workarea to the I/O area. For blocked (fixed and variable length) records, if the record fills the I/O area, the PUT routine writes the contents of the I/O area on the file. If unblocked or undefined records are specified, each PUT causes a record to be written on the file. If no workarea is specified, the user must build the physical record in the I/O area.

The PUT routine goes first to a workarea subroutine to determine if a workarea has

been specified (for undefined records, the record size is obtained prior to this action). If a workarea is specified, the record is moved from the workarea to the I/O area and any required blocking is performed. The PUT routine next goes to a read/write subroutine which checks for first time, two I/O areas, checkpoint records, etc, and performs the required I/O operation. The address of the next available I/O area is then loaded into the user's IOREG and the PUT routine returns control to the problem program.

MTMOD: READ Macro, Workfiles Chart HP

Objective: To read a physical record from tape.

Entry: From a READ macro expansion IJFWREAD.

Exit: To either the problem program or to the CHECK macro.

Method: The READ macro allows the user to access data on an input work file. This macro requires the user to specify the area the record is read into, and also allows the user to read only a portion of the record if he desires. The READ macro allows the user to read his records forward or backward. However, only one direction of reading is allowed for a particular file. Deblocking of blocked records is not performed by the READ macro.

The READ macro routine branches to the READ/WRITE Common Subroutine, Chart JG, to test for record format and length and to set up the proper controls. The decision block that contains S SPECIFIED IN PARAMETER LIST determines if the entire physical record is to be read for undefined records. If S is not specified, only a portion of the record is read. WLR checking is not performed.

Before returning to the READ macro routine, a check of bit 7 of byte 21 in the DTF is made to determine if a POINTS macro was issued before the READ. If the bit is on, indicating the POINTS was issued, the tape is at loadpoint. It is, therefore, necessary to execute a forward-space file operation to bypass any labels and position the tape to the first data record. When this operation is complete, control returns to the READ macro routine.

The READ macro routine issues an EXCP to start a read operation and returns to the problem program.

MTMOD: RELSE Macro Chart HQ

Objective: To skip the remaining records in a block.

Entry: From a RELSE macro expansion to the label:

IJFFREL for files containing fixed-length records.

IJFVREL for files containing variable-length records.

Exit: To the problem program.

Method: This routine causes the remaining records in the input block to be bypassed. Conditions are set up that cause the next GET instruction for this file to read in a new physical record from the file. The GET instruction makes the first logical record of the new block available to the user.

The RELSE routine sets the end-of-block pointer to 0, and returns control to the problem program.

The RELSE macro can only be issued to an input file.

MTMOD: TRUNC Macro Chart HR

Objective: To write the current block even if it is not full.

Entry: From a TRUNC macro expansion to the label:

IJFFTRU for files containing fixed-length records.

IJFVTRU for files containing variable-length records.

Exit: To the problem program.

Method: This routine causes the current block to be considered full, and then writes it on the file. It sets up conditions that cause the next PUT macro instruction to begin placing records at the beginning of a new block.

The TRUNC macro can only be issued to a blocked output file.

The routine gets the current blocksize, and checks to determine if it is 0. If yes, control returns to the problem program. If no, a new blocksize is set in the CCW, and control branches to the read/write subroutine to write the truncated record. Then the CCW is

restored, and control returns to the problem program.

MTMOD: WRITE Macro Workfile Chart HS

Objective: To write a physical record on tape.

Entry: From a WRITE macro expansion to IJFWRITE.

Exit: To the problem program.

Method: The WRITE macro instruction causes a record to be written from an indicated area to the output file. This routine requires the user to specify the area from which the record is written, and allows the user to specify the actual number of bytes to be written. If the length is not specified, the user specifies RECFORM=UNDEF in the DTF. Blocking of records is not provided for in the WRITE macro instruction. If blocking is desired, it is the user's responsibility to block the records before issuing a WRITE.

The WRITE macro routine branches to the READ/WRITE Common Subroutine, Chart JG, to determine the record length and set up the proper controls. Before returning control, the subroutine checks bit 7 of byte 21 in the DTF to determine if a POINTS macro was issued before the WRITE. If the bit is on, indicating that POINTS was issued, the tape is at loadpoint. It is, therefore, necessary to execute a forward-space file operation to bypass any labels and position the tape to write the first data record. When this operation is complete, control is returned to the WRITE macro routine.

The WRITE macro routine issues an EXCP to start a write operation and returns to the problem program.

MTMOD: Bypass Checkpoint Records Routine Chart JF

Objective: To bypass checkpoint records within an input tape data file.

Entry: From the Read/Write subroutine when CHKPTREC=YES is specified in the DTFMT entry and there are checkpoint records in the file.

Exit: To the Read/Write subroutine.

Method: For any mode input tape, this routine checks the first 12 bytes of each record in the input area for /// CHKPT ///

indicating a checkpoint header record has been read. For read backwards, the checkpoint header record is located in the 20 lower-order bytes of the input area.

Two methods for bypassing checkpoint records are:

1. For read forward, continue to read and bypass records until the checkpoint trailer label is encountered.

For read backwards, continue to read and bypass records until the checkpoint header label is encountered. The checkpoint header and trailer labels are identical.

2. Extract bytes 14-15 of the checkpoint header label (or trailer label if reading backwards), pack the field (convert to binary) and forward space (or backspace) that number of records.

Method 1 bypasses a Version 1 type checkpoint. Method 2 bypasses checkpoint types other than Version 1.

INITIALIZATION AND TERMINATION

Magnetic tape files processed by logical IOCS must conform to certain standards. These standards concern labels, placement of tapemarks, and the grouping (blocking) of tape records.

Tape files, with or without labels, can be processed by logical IOCS. It is the function of the open and close logical transients to process (check or create) the labels as required. Figures 18 and 19 illustrate the organization of tape file labels.

When standard labels are specified for a tape file, logical IOCS processes only the standard volume label, VOL1, and the standard file labels, HDR1, EOF1, and EOV1. (Refer to Figures 20 and 21 for the format of these labels.) User labels and nonstandard labels must be processed by the user.

Volume 1 of this set of four PLMs contains a detailed discussion on the use of tape labels.

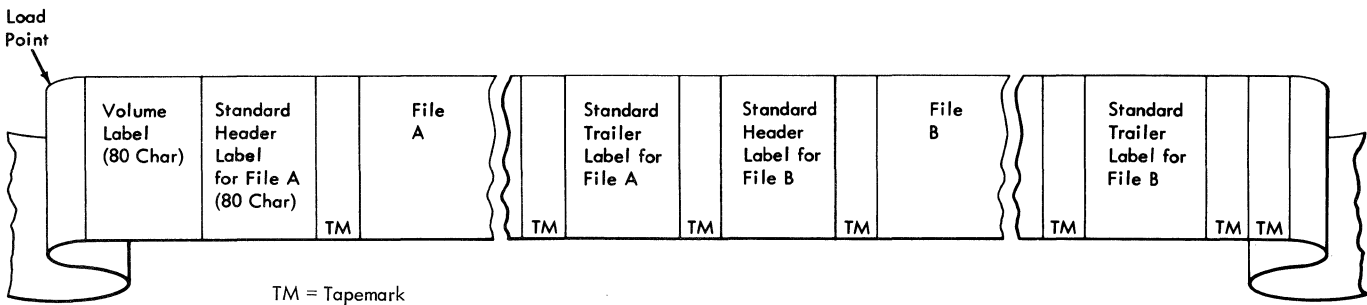


Figure 18. Tape File with Standard Labels

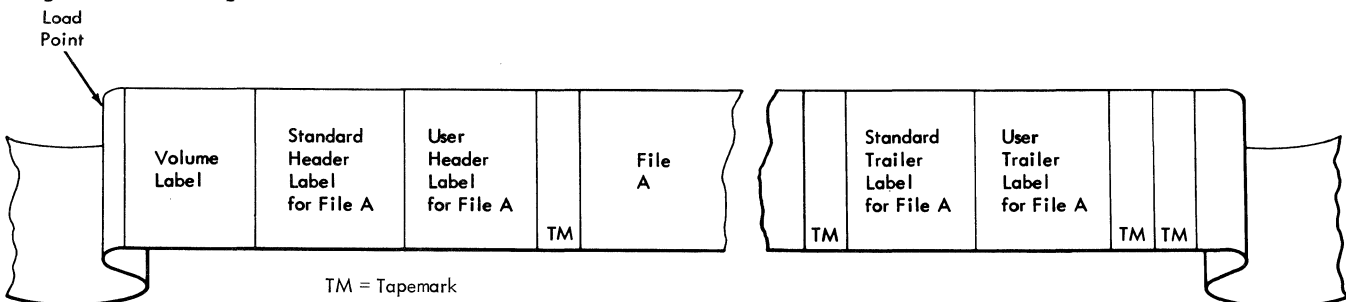
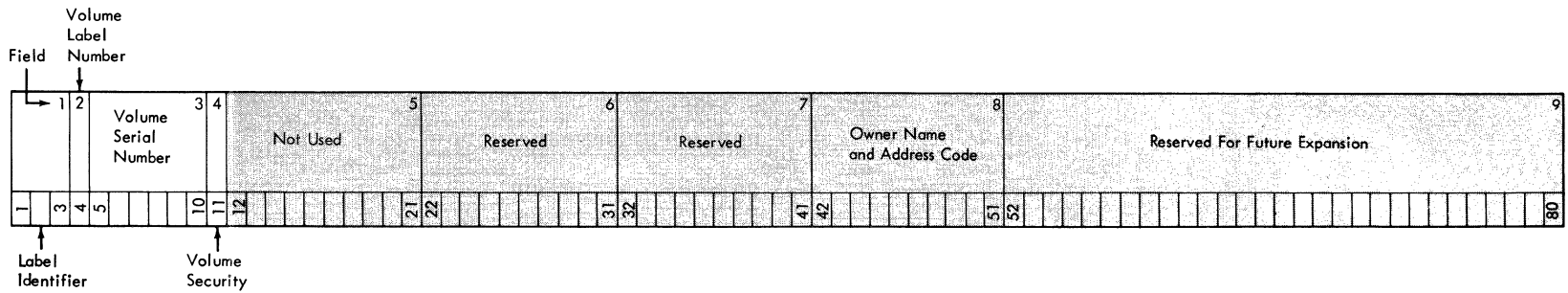


Figure 19. Tape File with Standard and User Labels

Figure 20. Standard Tape Volume Label

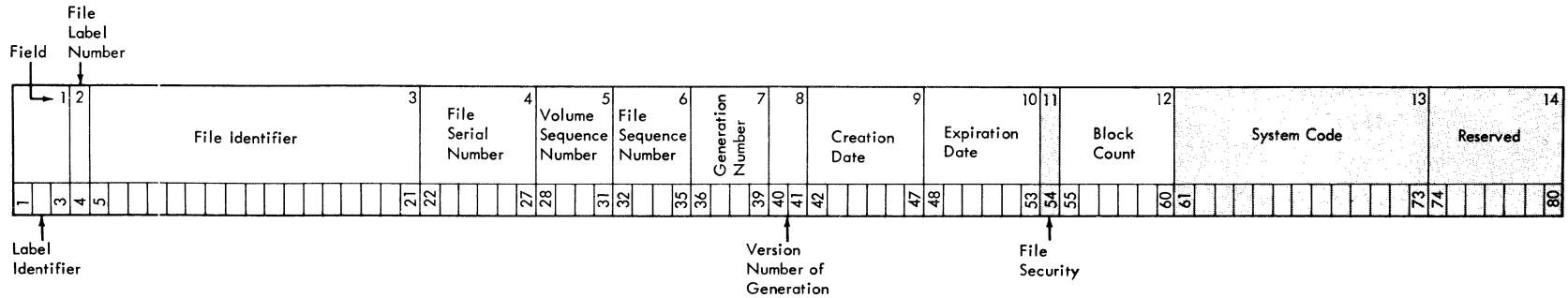


Volume Label Format (80 bytes) for Tape. (Shaded areas are not processed by DOS/360).

<u>FIELD</u>	<u>NAME AND LENGTH</u>	<u>DESCRIPTION</u>	<u>FIELD</u>	<u>NAME AND LENGTH</u>	<u>DESCRIPTION</u>
1.	<u>LABEL IDENTIFIER</u> 3 bytes	Must contain VOL to indicate that this is a Volume Label.	5.	<u>DATA FILE DIRECTORY</u> 10 bytes	Not used.
2.	<u>VOLUME LABEL NUMBER</u> 1 byte	Indicates the relative position (1-8) of a volume label within a group of volume labels. 1 is the standard volume label.	6.	<u>RESERVED</u> 10 bytes	Reserved.
3.	<u>VOLUME SERIAL NUMBER</u> 6 bytes	A unique identification code which is assigned to a volume when it enters an installation. This code may also appear on the external surface of the volume for visual identification. It is normally a numeric field 000001 to 999999, however any or all of the 6 bytes may be alphameric.	7.	<u>RESERVED</u> 10 bytes	Reserved.
4.	<u>VOLUME SECURITY</u> 1 byte (OS/360 only)	Indicates security status of the volume: 0 = no further identification for each file of the volume is required. 1 = further identification for each file of the volume is required before processing.	8.	<u>OWNER NAME AND ADDRESS CODE</u> 10 bytes	Indicates a specific customer, installation and/or system to which the volume belongs. This field may be a standardized code, name, address, etc.
			9.	<u>RESERVED</u> 29 bytes	Reserved.

Note: All reserved fields should contain blanks to facilitate their use in the future. Any information appearing in these fields at the present time will be ignored by the Disk Operating System programs as well as the Operating System/360 programs.

Figure 21. Standard Tape File Label



The standard tape file label format and contents are as follows: (Shaded areas are not processed by DOS/360)

FIELD	NAME AND LENGTH	DESCRIPTION	FIELD	NAME AND LENGTH	DESCRIPTION												
1.	<u>LABEL IDENTIFIER</u> 3 bytes, EBCDIC	identifies the type of label HDR = Header -- beginning of a data file EOF = End of File -- end of a set of data EOV = End of Volume -- end of the physical reel	9.	<u>CREATION DATE</u> 6 bytes	indicates the year and the day of the year that the file was created: <table border="1"> <thead> <tr> <th>Position</th> <th>Code</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>blank</td> <td>none</td> </tr> <tr> <td>2-3</td> <td>00-99</td> <td>Year</td> </tr> <tr> <td>4-6</td> <td>001-366</td> <td>Day of Year</td> </tr> </tbody> </table> (e.g., January 31, 1965, would be entered as 65031).	Position	Code	Meaning	1	blank	none	2-3	00-99	Year	4-6	001-366	Day of Year
Position	Code	Meaning															
1	blank	none															
2-3	00-99	Year															
4-6	001-366	Day of Year															
2.	<u>FILE LABEL NUMBER</u> 1 byte, EBCDIC	always a 1	10.	<u>EXPIRATION DATE</u> 6 bytes	indicates the year and the day of the year when the file may become a scratch tape. The format of this field is identical to Field 9. On a multi-file reel, processed sequentially, all files are considered to expire on the same day.												
3.	<u>FILE IDENTIFIER</u> 17 bytes, EBCDIC	uniquely identifies the entire file, may contain only printable characters.	11.	<u>FILE SECURITY</u> 1 byte	indicates security status of the file. 0 = no security protection 1 = security protection. Additional identification of the file is required before it can be processed.												
4.	<u>FILE SERIAL NUMBER</u> 6 bytes, EBCDIC	uniquely identifies a file/volume relationship. This field is identical to the Volume Serial Number in the volume label of the first or only volume of a multi-volume file or a multi-file set. This field will normally be numeric (000001 to 999999) but may contain any six alphanumeric characters.	12.	<u>BLOCK COUNT</u> 6 bytes	indicates the number of data blocks written on the file from the last header label to the first trailer label, exclusive of tape marks. Count does not include checkpoint records. This field is used in trailer labels.												
5.	<u>VOLUME SEQUENCE NUMBER</u> 4 bytes	indicates the order of a volume in a given file or multi-file set. The first volume can be any number. Subsequent numbers must be in proper numeric sequence.	13.	<u>SYSTEM CODE</u> 13 bytes	uniquely identifies the programming system.												
6.	<u>FILE SEQUENCE NUMBER</u> 4 bytes	assigns numeric sequence to a file within a multi-file set. The first file can be any number.	14.	<u>RESERVED</u> 7 bytes	Reserved. Should be recorded as blanks.												
7.	<u>GENERATION NUMBER</u> 4 bytes	uniquely identifies the various editions of the file. May be from 0001 to 9999 in proper numeric sequence.															
8.	<u>VERSION NUMBER OF GENERATION</u> 2 bytes	indicates the version of a generation of a file.															

Tape Open, Close and EOF/EOV Routines
Charts 01 and 02

Note: To simplify standard tape label processing and creation of tape files, Versions 2.1 and 3 provide a means of identifying tape files by a single job control statement, // TLBL. The // TLBL card, entered into the system with other job control statements, replaces both the // VOL and // TPLAB statements used with Versions 1 and 2. Note that Job Control accepts either type statement, and that a mandatory change of job control statements by the user is not required.

Figure 22 illustrates the format of the label information supplied by the job control statement(s) as it appears on SYSRES and in main storage. The tape open and close logical transients use this information to check the actual labels on the tape being processed.

In addition, Versions 2.1 and 3 incorporate certain changes in the DTFMT file definition declarative macro which, when used, includes some label information in the DTF table (refer to Figure 15). Once again, it is not mandatory for the user to redefine tape files defined by either Version 1 or Version 2 DTFMT macros.

The reader is cautioned, where reference is made to the // TLBL job control

statement in the text and flowcharts in this manual, that the reference applies to the collective // VOL and // TPLAB statements as well. Also, the decision block (that appears in some of the flowcharts) that contains the question

TLBL INFORMATION IN DTF

determines if the DTF table contains the tape label information included by the Version 2.1 and 3 DTFMT macros.

\$\$BOMT01: Open Input Standard Labels, Forward Charts KA-KC

Objective: To open, in a forward direction, input tape files with standard labels.

Entry: From the Open Monitor, \$\$BOPEN, or from a message writer phase to the label RELOCATE.

Input: From the label information stored on the SYSRES label information cylinder by Job Control (// TLBL card image) and from the tape currently being processed.

Exits:

1. To the logical IOCS module if a non-DTFPH multivolume file is being processed, or

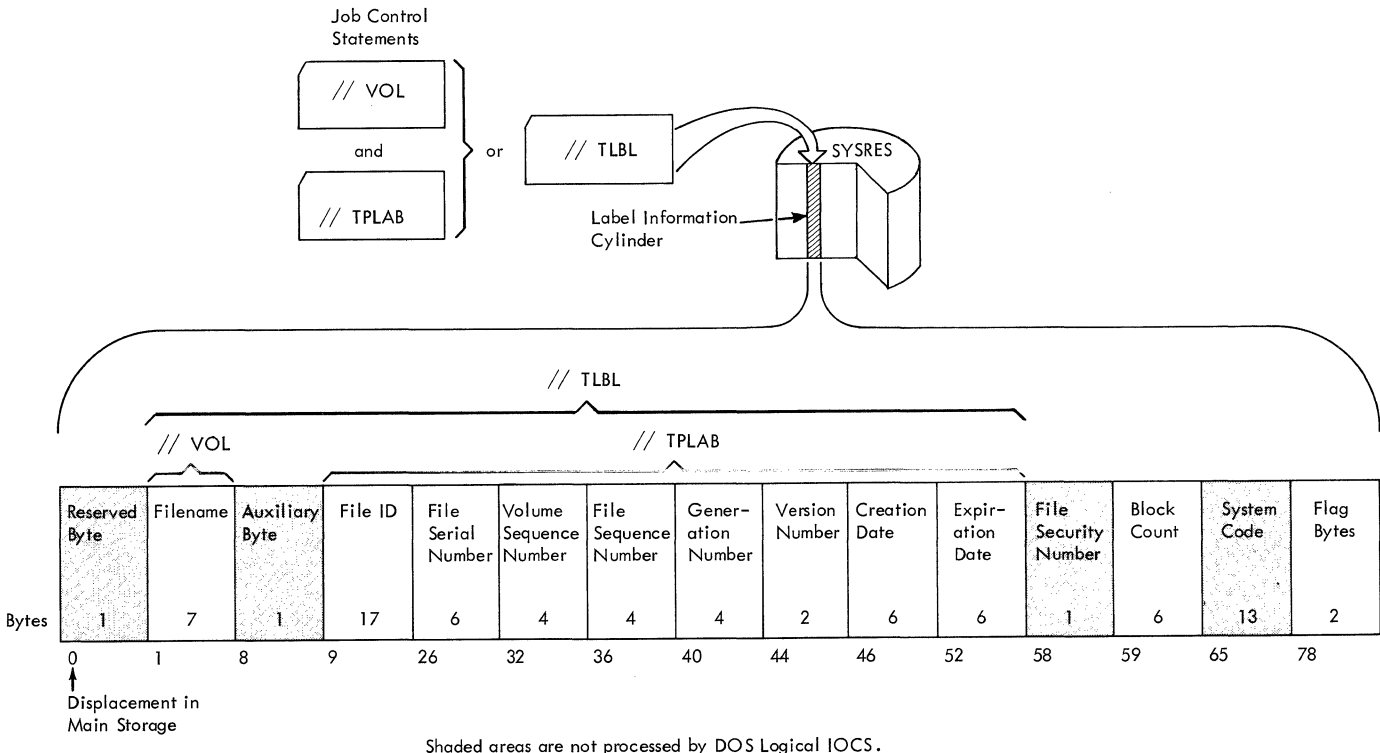


Figure 22. Format of SYSRES Tape Label Information

2. To the Open Monitor, \$\$BOPEN, if an OPEN macro was issued and/or a DTFPH file is specified.

Method: This routine opens an input tape file in a forward direction. It reads the standard volume label (VOL1) and compares the volume serial number contained in the label to the volume serial number obtained from SYSRES. The routine then reads the standard tape file header label (HDR1) and compares it to corresponding entries specified in the SYSRES label information (// TLBL card).

All differences are noted by messages to the operator. If user labels are detected and the address of the user's label routine (LABADDR) is specified in the DTF table for the file, an exit is made to that routine to read and process the user's labels. After the tapemark signifying the end of the label area is read, the file is posted open. As a result of a successful open, the file is posted open in the DTF table, and the tape is positioned at the IRG preceding the first data record.

Upon entry into the \$\$BOMT01 open phase, a test is made to determine if the phase was entered as a return from a message writer phase. If this is the case, a branch is made to the address specified by the particular response received from the operator for the message printed. This address is represented in the flowcharts by the exit(s) from the decision block following the message block.

If the DTF table was generated by Version 2.1 or Version 3 of DOS, it contains tape label information. If entry to phase \$\$BOMT01 was not by return from a message writer phase, a test is made for the presence of this label information. If the DTF contains such information, it must then be used to update the SYSRES label information.

Next, the \$\$BOMT01 open phase reads the tape for a standard volume label, VOL1 (refer to Figure 20). If this label is not found, error message 4111A is printed. If the VOL1 label is found, the DTF table is checked to determine if the file is already open. If the open indicator is on, byte 36 of the DTF table is examined to determine if the file is defined for processing by physical IOCS (DTFPH). If the file is so defined, the volume sequence number entry in the // TLBL card image is incremented by 1 unless the entry is blank.

If the file is not already open, the volume serial number in the tape VOL1 label is compared to the file serial number in the // TLBL card image. If the numbers are not the same, error message 4112A is

printed. If the response to this message is IGNORE, the // TLBL file serial number is made equal to the volume serial number. If the response is NEWTAP, indicating that the operator has mounted a new tape, checking of the standard volume label is repeated. If the response is BYPASS, or if the serial numbers are equal, the next record on the tape is read.

Any volume labels after the VOL1 label are bypassed. When a standard tape file label, HDR1 (see Figure 21), is read, the following label entries are checked against the corresponding entries in the // TLBL card image as specified: file serial number, volume sequence number, file sequence number, generation number, version number, creation date, and file identifier (FILEID). Any discrepancies are noted by error messages.

After the standard file label (HDR1) is read and checking is complete, all additional file labels are bypassed and the phase proceeds to check for user labels. If the address of the user's label routine is specified in the DTF table (LABADDR), any user labels on the tape are checked. If the user's label routine address is not specified, checking of user labels is bypassed.

The reading of a tapemark indicates the end of the label area on the tape. If the DTF table contains TLBL information, the table is updated from the new // TLBL card image. If the DTF table does not contain TLBL information, the new // TLBL card image is stored back in the SYSRES label information cylinder. The file is posted open in the DTF table and the \$\$BOMT01 phase is terminated.

\$\$BOMT02: Open Input Standard Labels, Backward Chart KD

Objective: To open, in a backward direction, input tape files with standard labels.

Entry: From the Open Monitor, \$\$BOPEN, or from a message writer phase to the label RELOCATE.

Input: Input is from the SYSRES label information cylinder (// TLBL card image) and the tape currently being processed.

Exits:

1. To Open Monitor, \$\$BOPEN, if an open statement was issued.

2. To CANCEL.

Method: This routine opens input tape files in a backward direction by reading the trailer labels instead of header labels.

Upon entry into the \$\$BOMT02 open phase, a test is made to determine if the phase was entered on a return from a message writer phase. If this is the case, a branch is made to the address determined by the response received from the operator for the message printed. This address is represented in the flowchart by the exit(s) from the decision block following the message block.

The \$\$BOMT02 routine then checks for a tapemark to determine that the tape is positioned correctly. If a tapemark is not read, error message 4117D is printed. After reading the tapemark, the routine reads the trailer labels.

If user trailer labels are present and the address of the user's label routine, LABADDR, is specified in the DTF table, the routine exits to the user's label routine to process the user trailer labels. If LABADDR is not specified, the user trailer labels are bypassed.

After processing the user labels, if required, the \$\$BOMT02 routine reads the file trailer labels until either a standard end-of-file or end-of-volume, EOF1 or EOV1, label is encountered (refer to Figure 21 for the label format). All intervening file trailer labels are ignored. When EOF1 or EOV1 label is read, only the file-identifier (FILEID) portion of the label is checked against its corresponding entry in the // TLBL card image in main storage.

Note: If the FILEID in the // TLBL card image is blank, no checking of the standard file trailer label is performed.

If the FILEID numbers are not equal, error message 4118D is printed.

After the file trailer label is processed, the routine reads the tapemark that precedes the trailer label area, posts the file open in the DTF table, and returns to the Open Monitor, \$\$BOPEN.

A successful open positions the tape at the IRG that follows the last data record.

\$\$BOMT03: Open Output Standard Labels, Forward (Phase 1) Charts KE-KF

Objective: To open, in a forward direction, output tape files with standard labels.

Entry: From the Open Monitor, \$\$BOPEN, or from a message writer phase.

Input: From the SYSRES label information cylinder (// TLBL card image) and from the tape currently being processed.

Exit: To the second phase of Open Output Standard Labels, Forward, \$\$BOMT04.

Method: The \$\$BOMT03 routine checks and builds labels for output tape files. The routine:

1. Checks the information in the // TLBL card image and inserts the proper data when blank entries are encountered.
2. Reads the standard volume label from the tape and compares it to the information in the // TLBL card image.
3. Determines the availability of the tape from the standard file label.
4. Writes new standard file and user labels as required.

When the required labels have been written on the tape, a tapemark is written and the file is posted open in the DTF table. At this point, the tape is positioned at the beginning of the data area.

Upon entry into the \$\$BOMT03 phase, a check is made to determine if the phase was entered on a return from a message writer phase. If this is the case, a branch is made to the address determined by the response received from the operator for the message printed. This address is represented in the flowcharts by the exit(s) from the decision block following a message block.

If a \$\$BOMT03 phase was not entered from a message writer phase, a test is made to determine if tape label information is contained in the DTF table. If so, (table generated by a Version 2.1 or Version 3 DTFMT macro), it is necessary to update the // TLBL card image in main storage with information contained in the DTF table.

This routine then checks the following fields of the // TLBL card image and inserts default values, as necessary, when blank entries are encountered:

- Volume sequence number (default-0001).
- File sequence number (default-0001).
- Generation number (default-0001).
- Version number (default-01).
- Expiration date (default-retention period = 0).
- Creation date (today's date).

The block count in the DTF table is set to zero and a check is made for specification of no-rewind (NORWD). If NORWD is not specified, the tape is rewound to loadpoint. If NORWD is specified and the tape is not at loadpoint, a branch is made to location BACKSPC2, Chart KF. If the tape is at loadpoint, a check is made on the density of the tape mounted and assigned. After the density check, the tape is read in search of a standard volume label, VOL1 (refer to Figure 20 for the format of a standard volume label).

If a VOL1 label is not found, error message 4110A is printed. If the operator response is a volume serial number, a new VOL1 label, followed by a dummy record to prevent possible detection of a noise record, is written. A response of NEWTAP indicates that the operator has mounted a new tape and the check of the VOL1 label is repeated.

If a standard volume label is found and the file is not already posted open in the DTF table, the file serial number in the // TLBL card image is checked. If this field is blank, the volume serial number from the VOL1 label is inserted into the card image. The volume serial number and the // TLBL card file serial number are compared. Any discrepancy is noted by error message 4112A. If the serial numbers are equal, the next record is read. If any additional volume labels are read, they are bypassed.

When the standard file label, HDR1 (refer to Figure 21 for the format of a standard file label), is read, the expiration date is checked to determine the availability of the tape. If the tape is available (that is, if the retention period has expired or the expiration date has been reached) or if no HDR1 label was found, the tape is backspaced and a new standard file label is built and written on the tape.

To build the new standard file label, information is obtained from the // TLBL card image. If the file identifier (FILEID) field of the // TLBL card is blank or zero, the filename is obtained from the DTF table and inserted into the card image.

The information from the // TLBL card image used to build the label includes:

- File identifier.
- File serial number.
- Volume sequence number.
- File sequence number.
- Generation number.
- Version number.
- Creation date.
- Expiration date.
- Block count.

If the tape is not at loadpoint, the routine at BACKSPC2 is entered to backspace two records and read one record in a forward direction. This sequence of operations is executed in an attempt to locate and retrieve information from the EOF1 label of the preceding file. The volume sequence, the file serial numbers, and the file sequence number +1 from the EOF1 label are inserted into the // TLBL card image. A new header label is built and written for the file being opened as previously described. If no preceding file labels are found, the tape is rewound and unloaded and error message 4120I is printed.

The normal exit from this phase is made to fetch \$\$BOMT04.

\$\$BOMT04: Open Output Standard Labels, Forward (Phase 2) Chart KG

Objective: To write user labels and open output tape data files with standard labels.

Entry: From the first phase of the Open Output Standard Labels, \$\$BOMT03.

Input: From the DTF table and the user's label routine.

Exits:

1. To the Open Monitor, \$\$BOPEN, if the file is not already posted open.
2. To the logical IOCS module if the file is already posted open.

Method: If phase \$\$BOMT04 is re-entered from a message writer phase, the return bit in byte 89 of the communications region is

reset. Control is then passed to the address in register 14.

When phase \$\$BOMT04 is entered normally (or re-entered from a message writer with no return address specified), a test is made to determine the version level of the DTF table. If the table was generated by a Version 2.1 or Version 3 DTFMT macro, the file serial, volume sequence, and file sequence numbers from the // TLBL card image (bytes 26 through 39) are inserted into the corresponding entries of the DTF table (bytes 42 through 55). Otherwise, the new // TLBL card image is written back onto the label information cylinder of SYSRES.

The \$\$BOMT04 phase then exits to the user's label routine if specified, to process and write up to eight user labels. When the user's label routine is complete, a tapemark is written to indicate the end of the tape label area, and the open indicator in the DTF table is checked. If the open indicator is already on, control returns to the logical IOCS module. If the open indicator is not on, it is turned on and the open monitor, \$\$BOPEN, is fetched.

A successful open posts the file open in the DTF table, and positions the tape at the IRG preceding the area of the first data record.

\$\$BOMT05: Open Input or Output,
Nonstandard or Unlabeled Charts KH-KJ

Objective: To open nonstandard and unlabeled files, both input and output, in either a forward or a backward direction.

Entry: From the Open Monitor, \$\$BOPEN.

Input: From the tape being processed.

Exits:

1. To the Open Monitor, \$\$BOPEN, after the file is posted open in the DTF table.
2. To the logical IOCS module if the file is already posted open in the DTF table (multireel file).

Method: This routine checks for 7-track or 9-track tape and sets the correct mode. If nonstandard labels are specified, an exit is made to the user label address. After the return from the user, a tapemark may or may not be written for an output file, according to the option specified in the DTF. For input files, a check is made to determine if the tapemark was read by the user.

For unlabeled files, tests for options such as tapemark, rewind, etc., are made to determine the various factors stated in the DTF parameters and appropriate action is taken. If the file has been previously opened (multireel file), control returns to logical IOCS. If the file has not been opened, the routine opens the file and calls the Open Monitor.

Upon entry to the \$\$BOMT05 phase, a test is made to determine if the phase was entered on a return from a message writer phase. If this is the case, a branch is made to the address (in register 14) specified by the response received from the operator for the message printed. This address is represented in the flowcharts by the exit(s) from the decision block following the message block.

If the phase was not entered on a return from a message writer phase, the DTF table is examined to determine the direction of the file to be opened and the appropriate action is taken.

1. For nonstandard labeled input files opened in a forward direction, the labels are processed by the user if the address of the user's label routine is specified in the DTF table. Detection of a tapemark indicates the end of the label area, and the file is posted open.
2. For unlabeled input files opened in a forward direction, the tape is read in search of a tapemark. If a tapemark is not read, the tape is backspaced one record to return it to its original position. In either case, the file is posted open in the DTF table.
3. For nonstandard labeled output files opened in a forward direction, the labels are processed by the user if the address of the user's label routine is specified in the DTF table. If the tapemark option is specified in the DTF table, a tapemark is written after the last label. The file is then posted open.
4. For unlabeled output files opened in a forward direction, the tape is read to determine if the tape contains any labels. If a label is detected, error message 4125D is printed and the operator can elect either to ignore the label or to mount a new tape. If the operator chooses to ignore the label and the tape is a 9-track tape, it is backspaced, written, and again backspaced. This sets the mode and erases the label. If the tapemark option is specified in the DTF table, a

tapemark is written. The file is then posted open in the DTF table.

5. For nonstandard labeled input files opened in a backward direction, the labels are processed by the user if the user's label routine address is specified in the DTF table and the file is posted open in the DTF table.
6. For unlabeled input files opened in a backward direction, the only check is for the existence of a tapemark. This test positions the tape properly. The file is then posted open in the DTF table.
7. For nonstandard labeled output files opened in a backward direction, the same action as in item 5 occurs.
8. For unlabeled output files opened in a backward direction, the same action as in item 6 occurs.

If a standard volume label is read, additional read commands are executed to bypass any additional volume labels. If a standard file label (HDR1) is read that is neither blank nor all zeros, the expiration date is checked to determine the availability of the tape. If the tape is available or if no HDR1 label is read, the tape is backspaced, a dummy file label of zeros is built and written, and a tapemark is written to indicate the end of the label area. The file is then considered open and control returns to the Open Monitor, \$\$BOPEN.

If the standard file label (HDR1) read is blank or zero, another read command is issued. If a tapemark is not read, the tape is backspaced and a tapemark is written. The file is then considered open.

\$\$BJCOPT: Job Control Tape Open Charts
KL-KN

\$\$BOMT06: Open Workfiles (Magnetic Tape) Chart KK-KKA

Objective: To open a tape workfile.

Entry: From the Open Monitor, \$\$BOPEN, to the label TRANSENT.

Input: From the tape being processed.

Exit: To the Open Monitor, \$\$BOPEN.

Method: This routine opens standard labeled or unlabeled, input or output, tape workfiles.

If this phase is reentered on a return from a message writer phase, a branch is made to the address specified by the response received from the operator for the message printed. This address is represented in the flowchart by the exit(s) from the decision block that follows a message block.

If this phase was entered from the Open Monitor the tape is rewound to loadpoint, unless the NORWD option is specified in the DTF table, and a read command issued.

If a tapemark is read, the file is immediately considered open and control is returned to the Open Monitor, \$\$BOPEN.

If a standard volume label (VOL1) is not read, the tape is backspaced and a tapemark is written. The file is then considered Open and control is returned to the Open Monitor, \$\$BOPEN.

Objective: To open SYSRDR, SYSIPT, SYSPCH, and SYSLST when these logical units are assigned to tape units.

Entry:

- From Job Control to open SYSRDR, SYSIPT, SYSPCH, and SYSLST when these logical units are assigned to tape units.
- From the Linkage Editor to open SYS001 if SYS001 is a tape.
- From Phase \$\$BCMT07 to open an alternate assignment.

Exit: To the problem program via SVC 11.

Method: This routine opens SYSRDR, SYSIPT, SYSPCH, and SYSLST when these logical units are assigned to tape units. Three attempts, using the following assumptions, are made to read the first record:

1. A label in standard mode and read first record; no retries.
2. Unlabeled file and read first record in user mode; no retries.
3. Read first record in standard mode; allow retries.

This is a table of actions taken by the routine depending upon the result of reading the first record:

Result	Input File	Output File
Unreadable	Rewind	Error
Unlabeled	Rewind	Rewind
Standard Label Set	Skip Labels, and tapemark.	Skip volume labels, write HDR1 and tapemark.

Note: If SYSLST, SYSPCH, or SYSOUT is assigned to a tape file, Job Control Open preserves the volume label; and if the header label is expired, Job Control writes a new header label with the format HDR1 followed by 76 binary zeros.

\$\$BCEOV1: EOF/EOV Monitor Charts KP-KQ

Objective: To initialize for End of File-End of Volume (EOF-EOV) routines, and to fetch the proper EOF-EOV routine. This phase is used for tape files only.

Entry: From the problem program when an end-of-file or end-of-volume condition is sensed (SEOV). Forced EOF (FEOV macro) also calls this routine.

Exit: To the proper EOF-EOV routine.

Method: This phase determines the type and format of the file from the DTF table, and selects the name of the required tape EOF or EO V phase. It then reads SYSRES label information for the file into the open table at the end of the logical transient area. This information is passed to the required tape EOF/EOV phase fetched by the EOF/EOV Monitor.

\$\$BCMT01: Tape Close, EOF/EOV Input Forward Charts KR-KS

Objective: To close an input tape file reading in a forward direction.

Entry: From the EOF/EOV Monitor, \$\$BCEOV1, to the label TRANSENT.

Input: From the SYSRES label information cylinder (// TLBL card image) and from the tape currently being processed.

Exit:

1. To the user's EOF routine if specified in the DTF table.

2. To phase \$\$BCMT02 for an EOF condition to switch to an alternate tape.
3. To CANCEL if an error condition occurs.

Method: This routine is called by the EOF/EOV Monitor. The routine reads and processes the standard trailer label. An exit and return are provided for processing user labels if LABADDR is specified. It checks to determine if an EOF or EO V condition is present. If an EOF condition exists, \$\$BCMT02 (alternate switching routine) is called. If an EOF condition exists, control passes to the user's EOF routine by branching to the user's EOFADDR.

Upon entry into the \$\$BCMT01 phase, a test determines if the phase was entered on a return from a message writer phase. If this is the case, a branch is made to the address specified (in register 14) by the response received from the operator for the message printed. This address is represented, in the flowcharts, by the exit(s) from the decision block following a message block.

If the phase was not entered on a return from a message writer phase, a check of the DTF table (byte 36) then determines if the file is being closed as a result of an FEOV (Force-End-of-Volume) macro being issued. If bit 3 of byte 36 is on, an FEOV macro was issued. The tape is rewound, unless the NORWD option is specified in the DTF. No label checking is performed. Phase \$\$BCMT02 is then fetched to determine if an alternate tape is available.

If an FEOV macro was not issued, the \$\$BCMT01 phase reads the tape in search of standard trailer labels if such are specified in the DTF table. Refer to Figures 18 and 19 for the organization of tape files and to Figure 21 for the format of standard tape file trailer labels, EO V1 and EOF1. Only the block count contained in these labels is checked against the block count contained in the DTF table. Any discrepancy is noted by error message 4131D. User labels are then checked if specified in the DTF table; otherwise, the user's trailer labels are bypassed. The tape is rewound, unless the NORWD option is specified, and a check determines the type of standard trailer label read (that is, EO V1 or EOF1). An EOF1 label sets the EOF-EOV switch, bit 4 or byte 36 in the DTF table. If the switch is turned on, the file is posted closed in the DTF table and an exit is made to user's end-of-file address, EOFADDR. If the switch is off, indicating that an end-of-volume condition exists, a test determines if the file is a COBOL file that must be rewound and unloaded. If so, the rewind-unload operation is executed. Phase \$\$BCMT02 is

fetched to determine if an alternate tape is available.

If nonstandard labels are specified, exit is made to the user's label routine (LABADDR) to process these labels. If no labels are specified or if the user specifies the end-of-file in the user label routine, exit is made to the user's end-of-file routine, EOFADDR. If the user specifies end-of-volume, phase \$\$BCMT02 is fetched to determine the availability of an alternate tape.

If the user specifies neither end of volume nor end of file, message 4130A is printed and the operator can specify either EOF or EOY. If the operator's reply is EOF, an exit is made to the user's end-of-file routine, EOFADDR. If the operator's reply is EOY, a test determines if the file is a COBOL file that must be rewound and unloaded. If so, the rewind-unload operation is executed. Phase \$\$BCMT01 then fetches phase \$\$BCMT02 to determine the availability of an alternate tape.

\$\$BCMT02: Tape Close, Alternate Switching for EOY Charts KT-KU

Objective: To switch tape drives when EOY is sensed.

Entry: From \$\$BCMT01 or \$\$BCMT04.

Input: From the trailer label of the active file and from the SYSRES label information (// TLBL card image) in main storage.

Exits: To \$\$BOMT01, to \$\$BOMT05, to \$\$BOPEN, or to CANCEL.

Method: This routine switches from one drive to another, if an alternate drive is specified and if EOY is sensed in the trailer label of the tape file previously processed. It switches either input or output tape files and calls in the appropriate open phase to open the alternate file. If EOY is not sensed and if no alternate drive is specified, a message is issued. The user can cancel the job or mount a new tape on the same drive and continue.

If the \$\$BCMT02 phase is reentered on a return from a message writer phase, a branch is made to the address (in register 14) specified by the response received from the operator for the message printed. This address is represented in the flowcharts by the exit(s) from the decision block following a message block.

If the \$\$BCMT02 phase was not entered from a message writer phase, a test determines if tape label information is contained in the DTF table. If so, (indicating that the table was generated by a Version 2.1 or Version 3 DTFMT macro), it is necessary to update the // TLBL card image in main storage with information contained in the DTF table.

An SVC 22 is issued to seize the system; that is, to disable the multiprogramming operation. The address of the logical unit block (LUB) entry for the tape being closed is obtained. From this entry, the addresses of the Physical Unit Block (PUB) entry and the first Job Information Block (JIB) entry, if any, are calculated. Any alternate assignment for the LUB is stored in the JIB table. After the alternate PUB is obtained and checked for availability, an SVC 22 is again issued to release the system, that is, to return to multiprogramming operation.

Note: Further description of the SVC 22 and of the functions of the LUB, PUB, and JIB tables is found in the IBM System/360 Disk Operating System, Supervisor and Physical and Logical Transients, Program Logic Manual listed in the Preface to this publication.

If the file being closed (the file for which an alternate assignment has been found) is an output file, the address of the DTF table is saved and the Open Monitor, \$\$BOPEN, is fetched. If the file is an input file with nonstandard labels, phase \$\$BOMT05 is fetched to open the alternate assignment.

If the file being closed is a standard labeled input file, the volume sequence number in the // TLBL card image in main storage is updated by 1. If the DTF table contains tape label information, it is also necessary to update the volume sequence number in the DTF table. If the DTF table does not contain tape label information, the updated // TLBL card image is written back onto the label information cylinder on SYSRES. In either case, phase \$\$BOMT01 is fetched to open the alternate assignment.

If no alternate assignment is available for the file being closed, message 4140A is printed. The operator can mount an alternate tape and give a NEWTAP response to the message. The phase then proceeds to reopen the file as if an alternate had been assigned.

\$\$BCMT03: Tape Close, EOF Backward Chart
KV

Objective: To close an input tape file reading in a backward direction.

Entry: From the EOF/EOV Monitor, \$\$BCEOV1, to the label TRANSENT.

Input: From the SYSRES label information cylinder (// TLBL card image) and from the tape being processed.

Exits:

1. To the user's end-of-file routine, EOFADDR.
2. To cancel, if an error condition is encountered.

Method: The \$\$BCMT03 phase reads the file header labels and processes the labels as trailer labels. The block count is checked and the file is then ready to be closed by the user's end-of-file routine.

Upon entry into the \$\$BCMT03 phase, a test determines if the phase was entered on a return from a message writer phase. If this is the case, a branch is made to the address (in register 14) specified by the response received from the operator for the message printed. This address is represented in the flowchart by the exit(s) from the decision block that follows a message block.

If the phase was not reentered from a message writer phase, a check of the DTF table determines the type of labels specified. If the file is unlabeled, that is, if neither standard nor nonstandard labels are specified in the DTF table, the end-of-file indicator (bit 4 of byte 35) is turned on and the first time switch (bit 0 of byte 21) is turned off in the DTF table. An exit is made to the user's end-of-file routine, EOFADDR.

If nonstandard labels are specified in the DTF table, an exit is made to the user's label routine (LABADDR) to process the labels. After the user's label routine is complete, or if the address of the user's label routine is not specified, the \$\$BCMT03 phase proceeds as described for unlabeled files.

If standard labels are specified in the DTF table, the tape is read in search of the standard file header label, HDR1. Refer to Figures 18 and 19 for tape file organization and to Figure 21 for the format of a HDR1 label.

All user header labels are processed by the user's label routine (LABADDR), or they are bypassed if the address of the user's label routine is not specified. All file header labels preceding (in a backward direction) the standard HDR1 label are bypassed. When the HDR1 label is read, only the block count contained in the label is checked against the block count contained in the DTF table. Any discrepancy is noted by error message 4131D. After checking the block count, the \$\$BCMT03 phase proceeds as described for unlabeled files.

\$\$BCMT04: Tape Close, EOY Output
Forward Charts KW-KX

Objective: To close an output tape file reading in a forward direction.

Entry: From the EOF/EOV monitor, \$\$BCEOV1, to the label TRANSENT.

Input: From the SYSRES label information cylinder (// TLBL card image) and from the tape being processed.

Exit: To phase \$\$BCMT02 to determine the availability of an alternate assignment.

Method: The \$\$BCMT04 phase closes the current active file by writing trailer labels and a tapemark, and fetches the alternate switching phase, \$\$BCMT02, to open the next tape reel.

Upon entry to the \$\$BCMT04 phase, a check determines the type of labels specified in the DTF table. If standard labels are specified, the version level of the DTF table is verified by examining bit 5 of byte 31 in the table. If the bit is on, the table was generated by a Version 2.1 or Version 3 DTFMT macro. It is then necessary to modify the corresponding entries in the // TLBL card image in main storage with the file serial, volume sequence, and file sequence numbers from the DTF table. After the // TLBL card image is modified with the DTF table information, the following card entries are checked and the proper default value inserted if any are blank:

- File identification (FILEID) number (default = filename from the DTF table).
- Generation number (default = 0001).
- Version number (default = 01).
- Expiration date (default = retention period set to 0 days).

Note: If expiration date is not absolute, the expiration date is calculated by adding the retention period to today's date.

After updating the // TLBL card image or if bit 5 of byte 31 in the DTF table is not on (indicating that the DTF table does not contain tape label information and updating is bypassed), the volume sequence number in the // TLBL card image is incremented by 1 and the entire updated card image is used to build a standard EOF1 file label. Refer to Figure 21 for the format of a standard file label.

The updated // TLBL card image is then written back onto the label information cylinder of SYSRES. If the DTF table does contain tape label information, it is also updated with the information from the // TLBL card image. The EOF1 label is then written on the tape.

An exit is made to the user's label routine (LABADDR) to process and write user labels as specified. After the user's label routine is complete, a tapemark is written to indicate the end of the label area. The tape is rewound, rewound and unloaded, or left positioned, as specified by the user in the DTF table. The block count in the DTF table is reset to zero, and phase \$\$BCMT02 is fetched to determine the availability of an alternate assignment.

If nonstandard labels are specified in the DTF table, an exit is made to the user's label routine to process these labels and the \$\$BCMT04 phase is concluded in the manner just described.

\$\$BCMT05: Close Standard, Non-standard, and Unlabeled Files, All Types Except Workfiles Charts KY-KZ

Objective: To close a tape input or output file, reading in the forward or backward direction.

Entry: From the Close Monitor, \$\$BCLOSE, to the label TRANSENT.

Input: From the SYSRES label information cylinder (// TLBL card image) and from the tape being processed.

Exit: To the Close Monitor, \$\$BCLOSE.

Method: The \$\$BCMT05 phase closes standard, nonstandard, and unlabeled input or output files in either direction, forward or backward. The phase writes a standard EOF1 trailer label on output files when standard labels are specified and/or

exits to the user's label routine to process user labels or nonstandard labels. No label checking is performed on input files.

Upon entry into the \$\$BCMT05 phase, a check is made of the open indicator (bit 5 of byte 36) in the DTF table. If the bit is not on (file not open), the block count in the DTF table is set to zero, the first time switch (bit 0 of byte 21) in the DTF table is turned off, and the Close Monitor, \$\$BCLOSE, is fetched. If the open indicator is on, a test determines if the file is input or output.

If the file is an input file, the tape is rewound, rewound and unloaded, or left positioned as specified in the DTF table by the user. The file is then posted closed in the DTF table. The \$\$BCMT05 phase is concluded by resetting the DTF block count to zero and turning off the DTF first time switch. The Close Monitor, \$\$BCLOSE, is then fetched.

If the file is an output file, a tapemark is written to indicate the end of the data area. For nonstandard labeled output files, an exit is made to the user's label routine (LABADDR), if specified. When the user's label routine is complete, or if the address of the user's label routine is not specified for nonstandard labels, or if the file is unlabeled, a tapemark is written and the DTF table is checked for the NORWD option. If the NORWD option is not specified, the tape is either rewound, or, rewound and unloaded, as the user requires. If NORWD is specified, the tape is backspaced over the last tapemark written. The \$\$BCMT05 phase is then concluded by posting the file closed, resetting the DTF block count to zero, turning off the first time switch in the DTF, and fetching the Close Monitor, \$\$BCLOSE.

For standard labeled output files, the version level of the DTF table must be determined by examination of bit 5 of byte 31 in the DTF table. If this bit is on, the table was generated by a Version 2.1 or Version 3 DTFMT macro and contains information needed to update the // TLBL card image in main storage.

The first step in the updating procedure inserts the file serial, volume sequence, and file sequence numbers from the DTF table into the corresponding entries of the // TLBL card image. The second step checks the following entries of the // TLBL card image and inserts the proper default value where blank entries are encountered:

- File identification (FILEID) number (default = the DTF filename).

- Generation number (default = 0001).
- Version number (default = 01).
- Expiration date (default = set retention period to 0 days).

Note: If expiration date is not absolute, the date is calculated by adding the retention period to today's date.

If the DTF table was not generated by a Version 2.1 or Version 3 DTFMT macro, no modification of the // TLBL card image is necessary and the update procedure is bypassed.

A standard EOF1 label (refer to Figure 21 for the format of a standard file label) is then built from the // TLBL card image information and written on the tape. If user labels are specified in the DTF table, an exit is made to the user's label routine (LABADDR) to process and write the user's labels.

When all labels have been written, two tapemarks are written to indicate the end of the label area and the last file on the reel. If the user has specified the NORWD option, the tape is backspaced over the last tapemark written. On the other hand, if the NORWD option is not specified, the tape is either rewound, or rewound and unloaded, as the user requires. The file is then posted closed, the block count in the DTF table reset to zero, the first time switch in the DTF table turned off, and the Close Monitor, \$\$BCLOSE, fetched.

\$\$BCMT06: Close Workfiles Chart LA

Objective: To close a tape workfile.

Entry: From the Close Monitor, \$\$BCLOSE, to the label WORKFILE.

Input: From the DTF table.

Exit: To the Close Monitor, \$\$BCLOSE.

Method: This routine closes any magnetic tape workfile. It checks to determine if the last tape operation was a read or a write. If write, a tapemark is written, and the rewind option is tested. If read, the rewind option is tested. In either case, if NORWD was specified, the rewind function is bypassed. If NORWD was not specified, the tape is either rewound to the loadpoint, or, rewound and unloaded, as the user requires. The block count in the DTF table is reset to zero, and the Close Monitor is fetched.

\$\$BCMT07: Alternate Switching for System Units Charts LB-LC

Objective: To terminate and switch to an alternate drive when an EOF condition is encountered on an output tape file assigned to either SYSLST or SYSPCH.

Entry: From the EOF/EOV Monitor, \$\$BCEOV1, or from an output CP module.

Input: From the Supervisor I/O tables. Refer to the IBM System/360 Disk Operating System, Supervisor and Physical and Logical Transients Program Logic Manual, listed in the Preface to this publication.

Exits:

1. To the message writer phase, \$\$BMSGWR, if an alternate drive is not available.
2. To the Job Control open phase, \$\$BJCOPT, to open the alternate assignment.
3. To the logical IOCS module if a unit exception occurs while writing a tapemark.

Method: This phase closes the active tape output file by writing a tapemark and by rewinding and unloading the tape. The alternate device is determined, if specified, or an exit is made to the message writer phase, \$\$BMSGWR. If an alternate is not specified, or not available, the operator can mount a new tape on the same drive and continue processing.

Upon entry into the \$\$BCMT07, a test determines if the no rewind option is specified in job control switch byte JBCSW2 (byte 58 in the communications region). If the option is not specified, a tapemark is written and a test determines if the unit exception bit is on in the CCB (bit 7 of byte 4). If the bit is not on, alternate switching is not required. In this case, the tape is backspaced over the last tapemark written and control is returned to the logical IOCS module via an SVC 11.

If the unit exception bit in the CCB is on, alternate switching is required. The tape is rewound and unloaded and an SVC 22 is issued to seize the system (that is, to disable multiprogramming operation).

Note: Additional information concerning SVC 22 and the functions of the LUB, PUB, and JIB tables is found in the IBM System/360 Disk Operating System, Supervisor and Physical and Logical Transients Program Logic Manual, listed in the Preface to this publication.

The address of the Logical Unit Block (LUB) entry is calculated and the pointer to the Physical Unit Block (PUB) entry is obtained from the first byte of the LUB entry. The address of the PUB assigned to the LUB is calculated, and the PUB mode byte (byte 5) is saved for later use.

The second byte of the LUB entry contains a pointer to an entry in the Job Information Block (JIB) table if any JIB's are specified (chained) for the LUB. If there is an alternate PUB assigned, it is stored in a JIB. The first JIB in the chain is examined for a stored alternate PUB. If the JIB does not contain an alternate PUB, the next JIB in the chain is checked. When a stored alternate assignment is found, the pointer to that PUB is obtained from the JIB and inserted into the first byte of the LUB. Any remaining JIB's are scanned and updated. The standard PUB assignment is stored in the last JIB in the chain.

After the alternate assignment is checked for availability, the mode byte saved from the standard PUB is inserted into the alternate PUB. The \$\$BCMT07 phase concludes by issuing an SVC 22 to release the system (restore multiprogramming operation), and phase \$\$BJCOPT is fetched to open the alternate assignment.

If an alternate assignment is not found, or not available, initialization to print message 4121A takes place.

\$\$BOMTOM and \$\$BOMTOW: Tape Open Message Writers Chart LE

Note: The tape open message writer phases \$\$BOMTOM and \$\$BOMTOW perform the same function. The two phases differ only in the messages printed and the calling phases.

Objective: To print messages to the operator and analyze the operator's response to a particular message.

Entry:

Phase \$\$BOMTOM--from tape open phase
\$\$BOMT01

Phase \$\$BOMTOW--from tape open phases
\$\$BOMT02, \$\$BOMT03, \$\$BOMT04, \$\$BOMT05,
\$\$BOMT06.

Input: Pointer, received from an open routine, to indicate the desired message.

Exits:

1. To the calling open routine or to the Open Monitor (\$\$BOPEN), if a legal response to the message is received, or
2. To CANCEL if no 1052 is available, or if the response is EOB (end-of-communication) or illegal.

\$\$BOMTOM Messages:

4110A NO VOL1 LBL FOUND
4112A VOL SERIAL NO. ERROR
4117D NO TM FOUND ON READBK
4118D FILE ID ERROR, READBK
4119A FILE UNEXPIRED
4120I TAPE POSITIONED WRONG
4123D WRONG POSITN, READBK
4124I TOO MANY UHL'S
4144A 1600 BPI TAPE MOUNTED
4151I HDR1 LBL INFORMATION

\$\$BOMTOW Messages:

4111A NO VOL1 LBL FOUND
4112A VOL SERIAL NO. ERROR
4113D NO HDR1 LBL FOUND
4114A FILE SEQ NO. ERROR
4115A SERIAL NO. ERROR
4116A VOLUME SEQ NO. ERROR
4132D ERROR IN FILE ID
4133D ERROR IN HDR LBL
4151I HDR1 LBL INFORMATION
4883I INVALID LOGICAL UNIT

Note: Message 4883I is issued by any tape open/close phase in which this error is encountered.

Method: The message writer phases \$\$BOMTOM and \$\$BOMTOW print message and interrogate responses.

The calling phase passes the following information to the message writers:

- Register 0. The number of the desired message.

- Register 12. The BYPASS reply return address.
- Register 13. The NEWTAP reply return address.
- Register 14. The IGNORE, RETRY, VOLUME SERIAL NO. reply, or the information type message return address.
- The last two bytes in the name of the phase returned to.

Each message contained in the message writer phases consists of four sections:

1. Option byte. Indicates the action to be taken after printing the message or the valid responses.

Bit 0 = Automatic terminate (cancel).

Bit 1 = IGNORE.

Bit 2 = NEWTAP.

Bit 3 = RETRY.

Bit 4 = BYPASS.

Bit 5 = Information message (no response required).

Bit 6 = Volume serial number.

Bit 7 = Print TLBL = XXXXXX.

2. Displacement byte. Contains displacement, in bytes, from the start of the label to the field(s) to be printed.
3. Length byte. Contains the length, in bytes, of the label information to be printed.
4. Message.

A message writer phase assembles and writes the required message, and receives and analyzes the response received from the operator. The phase then passes control to the proper open phase or to the cancel routine. If control is returned to an open phase, bit 4 of byte 89 in the communications region is set to indicate to the open phase that entry is made on a return from a message writer phase.

\$\$BMSGWR: Tape Open/Close Message Writer Chart LF

Objective: To print messages to the operator and analyze the operator's response to a particular message.

Entry: From tape Open/Close phases:

\$\$BOMT05

\$\$BCMT01

\$\$BCMT02

\$\$BCMT03

\$\$BCMT07

Input: Pointer, received from the calling phase, to indicate the desired message.

Exits:

1. To the calling phase if a legal response to the message is received, or
2. To CANCEL if no 1052 is available, or if the response is EOB (end-of-communication), or is illegal.
3. To \$\$BJCOPT if entered from tape close phase \$\$BCMT07.

Messages:

4117D NO TM FOUND ON READBK

4119A FILE UNEXPIRED

4121A NO ALTERN DRIVE ASSGN

4122I EOF ENCOUNTERED

4125D VOL1 LBL FOUND

4126I EOF ENCOUNTERED

4130A EOF OR EOF INQUIRY

4131D BLOCK COUNT ERROR

4140A NO ALTERN DRIVE ASSGN

Note: Messages not requiring printing of label information may also be passed to the \$\$BMSGWR phase.

Method: The \$\$BMSGWR routine is used for message writing and response interrogation.

The calling routine passes the following information to the \$\$BMSGWR routine:

- Register 0. The number of the desired message.
- Register 12. The RETRY reply return address.
- Register 13. The NEWTAP or EOF reply return address.

- Register 14. The IGNORE or EOF reply, or the information type message return address.

- The last two bytes in the name of the phase returned to.

Each message contained in this phase consists of four sections:

1. Option byte. Indicates action to be taken after printing the message or valid response(s).

Bit 0 = EOF or EOF

Bit 1 = IGNORE

Bit 2 = NEWTAP

Bit 3 = Double message

Bit 4 = RETRY

Bit 5 = Information message (no response required)

Bit 6 = System unit possible and CANCEL not legal response

Bit 7 = Leave DTF blank.

2. Displacement byte. Contains displacement, in bytes, from the start of the label to the field(s) to be printed.
3. Length byte. Contains the length, in bytes, of the label information to be printed.
4. Message.

The \$\$BMSGWR message writer phase assembles and writes the required message and receives and analyzes the response received from the operator. The phase then passes control to the proper open phase or to the cancel routine. If control is returned to an open phase, bit 4 of byte 89 in the communications region is set to indicate to the open phase that entry is made on a return from a message writer phase.

Device independent files are those files defined by either a DTFCP macro or DTFDI macro.

The DTFCP macro defines files used by IBM compilers: COBOL, FORTRAN and PL/I.

The DTFDI macro defines files assigned to the device independent system units SYSRDR, SYSIPT, SYSPCH, and SYSLST. The DTFDI macro and its associated DIMOD macro, therefore, provide DOS Assembler users with the same capabilities extended by DTFCP.

COMPILER FILES

Compiler (CP) files are files provided specifically for IBM internal programs such as COBOL, FORTRAN, and PL/I. These files, defined by a DTFCP macro, provide limited device independence. Because this file definition does not conform to standards established for other logical IOCS component and is tested for use only by IBM internal programs, it is not documented in any Systems Reference Library publications.

Some of the differences between CP components (DTFCP and CPMOD) and other LIOCS components are:

1. No provision for separate assembly.
2. Error recovery not the same.
3. The DTFCP table is not self-initializing; that is, the user must initialize the table if the file is reopened.

DTFCP Macro

A DTFCP macro instruction can be used for each file that has fixed, unblocked records, and limited device independence. When the file is opened, a channel program for reading/writing on a particular device is built in the DTFCP table area (Figures 23, 24, and 25). Only the following devices can be accessed by DTFCP: IBM 1442, 1443, 1403, 2540, 2400 series, 2501, 2520, 1404 (continuous forms only), 1445, 2311, and 2314.

The DTF table generated at assembly time is initialized by the DTFCP open phase

according to the device type. The device type is found by the open phase by checking the device type set in the PUB table entry. After it is found, the proper indicators are set in the DTF table, and the work areas and CCW's are modified. Standard labels are not required on tape files.

The DTFCP header card is followed by a series of parameter cards describing the file, and specifying symbolic addresses of routines and areas used for processing the file. Because keyword parameters are specified, the parameter cards may appear in any order. This group of cards generates the necessary logical IOCS DTFCP tables during assembly.

The parameter cards following the DTFCP header card have keyword entries in the operand field. All cards used in the DTFCP macro instruction, except the last, have continuation punches in column 72.

Filename: This is the first entry in a DTFCP macro instruction. It assigns a symbolic name to the file, which appears in all I/O macro statements referencing this file. The symbolic name of the file is in the name field, and DTFCP is in the operation field.

DEVADDR=SYSxxx: This parameter specifies the symbolic unit to be associated with the file. If DEVADDR=SYSPCH, the DTF is generated to include error recovery procedures for the IBM 2540 and 2520 punch units.

IOAREAL=name: The I/O area to be used by the file is defined by an address expression.

IOAREA2=name: If two I/O areas are needed for overlapped GET/PUT processing, this parameter is required.

IOREG=(n): For input files, this parameter specifies the general purpose register (n) into which IOCS inserts the address of the next logical record available for processing. For output files, IOCS inserts the address of the area where the user can build the next logical record. Any register 2-12 may be specified. The same register may be used for different files.

This parameter must be specified whenever two I/O areas are used.

RECSIZE=n: For fixed-length records, this parameter specifies the number of characters in the record. I/O routines use this factor for checking record length. If this entry is not specified, the maximum record length of 81 bytes is assumed.

TYPEFLE=INPUT or OUTPUT: This parameter defines whether the file is an input or output file. From this specification, IOCS creates the appropriate channel program for this file. If this entry is not specified, an input file type is assumed.

EOFADDR=name: This parameter defines a problem program address to which IOCS branches on an end-of-file condition. This entry must be supplied if TYPEFLE=INPUT. The user must perform any necessary end-of-file processing; for example, issue a CLOSE macro.

DISK=YES, NO or omitted: If the DISK= parameter is omitted, a tape system is assumed, and the same DTF table (Figure 25) is generated as in the tape system. If the DISK= parameter is present, the disk resident system is assumed. If DISK=YES, a DTF table (Figure 23) is generated that supports any device including the IBM 2311 and 2314. If DISK=NO, a DTF table (Figure 24) is generated for a disk resident system that supports a device other than DASD.

IOPTR=YES or omitted: If YES is specified, the I/O area address must be put into register 0 before a GET or a PUT is issued. The module inserts the contents of the register into the data address portion of the CCW and proceeds as in the case of one

I/O area. The parameter must be included if a variable I/O area is required.

RONLY=YES or omitted: If YES or blank, the file uses a reentrant CP logic module. Unless RONLY is omitted, the problem program must provide the logic module with the address of the unique 72-byte, doubleword aligned, savearea for the file. This address is supplied in general register 13 before a logical IOCS imperative macro is issued to the file.

CHARACTERISTICS OF DTFCP FILES

1. Only fixed unblocked records formats are supported.
2. Forward read only is supported. No read backwards is permitted.
3. Employs standard physical IOCS error recovery procedures.
4. Control character (ASA) is always required for printer and punch files.
5. GET and PUT are supported.
6. Multivolume output of SYSLST and SYSPCH on tape is supported.
7. Variable I/O area is supported if IOPTR is specified. For example, any area pointed to by register 0 when a GET or PUT is issued is considered the I/O area.

Bytes	Bits	Function
0-15		CCB.
16	0-1	Not used.
	2	COBOL open; ignore option.
	3	Not used.
	4	DTF table address constants relocated by OPENR.
	5-7	Not used.
17-19		Logic module address.
20		DTF type X'32' except in the case of disk assigned to units SYS000 to SYSnnn. In this case, a DTFCP open phase changes it to X'20'.
21		Open indicators: X'02' input, X'00' output, except for tapes assigned to SYS000 to SYSnnn when X'00' = input and X'08' = output.
		X'08' DISK=YES indicator for other than Version 1.
	0	1 = No rewind, 0 = Rewind for Versions 2.1 and 3.
22-28		Filename (see byte 29).
29		Last byte of filename except in Version 3. Version 3 device type code: X'00' = 2311 X'01' = 2314
30-35		File address for disk.
36-41		Volume serial number if labeled output tape (present only after open).
36-37		Volume sequence number (DASD), Versions 1 and 2.
38		Open switch.
39		Sequence number of current extent.
40		Sequence number of last extent, or X'80' if 1442 punch.
41		X'80' indicates request for standard label tape OPEN.
42		X'20' device is a DASD. X'10' device is a tape. X'08' device is a printer. X'04' device is a punch. X'02' device is a reader.

Figure 23. DTFCP: DISK=YES (Part 1 of 3)

Bytes	Bits	Function
43		X'F3' device is a DASD. X'F1' device is a reader. X'F0' device is other type.
44	0	1 = input, 0 = output.
	1	1 = eject needed for a reader-punch. 0 = no eject.
	2	0 = first pass, 1 = not first pass.
	3	1 = 2 I/O areas, 0 = 1 I/O area.
	4	1 = 2540 punch.
	5	1 = SYSLST or SYSPCH.
	6	1 = SYSLST or SYSPCH on output tape.
	7	1 = TLBL is present and tape is labeled.
45-47		IOAREA2 address.
48-51		X'80000000'.
52-53		Lower head limit.
54-57		Extent upper limit.
58-64		BECCHHR seek address.
65-67		EOF address.
68-71		Control bucket CCHH.
72		Number of records per track for output, number of records per track + 1 for input.
73		X'00' for output, X'01' for input.
74-75		X'20' for output, X'18' for input for DASD. X'0000' for nondisk device.
76-80		CCHHR for count field.
81		Key length.
82-83		Data length.

Figure 23. DTFCP: DISK=YES (Part 2 of 3)

Bytes	Bits	Function
84-87		Instruction to load user I/O area address to I/O register.
88-111		Seek, search, TIC CCW's.
112-119		CCW for DASD input and first CCW for DASD output. This CCW can be used for other devices if unit is not a DASD.
End-of-table if DTF is defined for an input file.		
120-127		Second CCW for output.
128-151		Verify CCW's for output.
End-of-table if DTF is defined for output file and DEVADDR does not equal SYSPCH.		
152-159		2540 punch error recovery CCW 1.
160-167		2540 punch error recovery CCW 2.
168-231		Reserved.
When the CP open initializes the table and determines that the device is a 2540 punch, the following bytes in the table are changed:		
30		X'FF' indicator to DTFCP open phases and logic module.
32-35		Instruction to load user I/O area to I/O register.
48-55		CCW.
56-63		2540 punch error recovery CCW 1.
64-71		2540 punch error recovery CCW 2.
72-151		80-byte card image, savearea 1.
152-231		80-byte card image, savearea 2.

Figure 23. DTFCP: DISK=YES (Part 3 of 3)

Bytes	Bits	Function
0-15		CCB.
16	0-1	Not used.
	2	COBOL open; ignore option.
	3	Not used.
	4	DTF table address constants relocated by OPENR.
	5-7	Not used.
17-19		Logic module address.
20		DTF type X'32' except in the case of tape assigned to units SYS000 to SYSnnn. In this case, a DTFCP open phase changes it to X'10'. ¹
21		Open indicators X'02' input, X'00' output (except for tapes assigned to SYS000 to SYSnnn when it is X'00' input, X'08' output).
22-28		Filename (see byte 29).
29		Reserved for future use in Versions 2.1 and 3. Last byte of filename in Versions 1 and 2.
30		X'FF' indicator to DTFCP open, and logic.
31		Reserved for future use.
32-35		Instruction to load user's I/O area address into I/O register.
36-39		Reserved for future use.
40		X'80' if 1442 punch.
41		X'20'
42		X'10' device is a tape.
		X'08' device is a printer.
		X'04' device is a punch.
		X'02' device is a reader.
43		X'F1' device is a reader or tape.
		X'F0' device is other type.

Figure 24. DTFCP: DISK=NO (Part 1 of 2)

Bytes	Bits	Function
44	0	1 = input, 0 = output.
	1	1 = eject needed for a reader-punch. 0 = no eject.
	2	1 = not first pass, 0 = first pass.
	3	1 = 2 I/O areas, 0 = 1 I/O area.
	4	1 = 2540 punch.
	5	1 = SYSLST or SYSPCH.
	6	1 = SYSLST or SYSPCH on output tape.
	7	Reserved for future use.
45-47		IOAREA2 address.
48-55		CCW.
End-of-table is DTF is defined as output file and DEVADDR is not equal to SYSPCH.		
56-63		2540 punch error recovery CCW 1.
64-71		2540 punch error recovery CCW 2.
65-67		EOF address, input only.
End-of-table if DTF is defined as input file.		
72-151		80-byte card image, savearea 1.
152-231		80-byte card image, savearea 2.

¹DTF type X'30' found in DOS LIOCS Version 1 only.

Figure 24. DTFCP: DISK=NO (Part 2 of 2)

Input or output table generated when the parameter DISK=parameter is omitted from the DTFCP macro (Tape Resident System Type).

Bytes	Bits	Function
0-15		CCB.
16	0-1	Not used
	2	COBOL open; ignore option.
	3	Not used.
	4	DTF table address constants relocated by OPENR.
	5-7	Not used.
17-19		Logic module address.
20		DTF type X'31' except in the case of tape assigned to units SYS000 to SYSnnn. In this case DTFCP open phase changes it to X'10'.
	0	1= no rewind 0= rewind
21		Open indicators X'02' input, X'00' output (except for tapes assigned to SYS000 to SYSnnn when it is X'00' input, X'08' output).
22-28		Filename.
29		Reserved for future use.
30		X'FF' indicator to DTFCP open, and logic.
31		Reserved for future use.
32-35		Instruction to load user's I/O area address into I/O register.
36-41		Volume serial number if labeled output tape.
36-39		Reserved for future use.
40		X'80' if 1442 punch.
41		X'20'
42		X'10' device is a tape. X'08' device is a printer. X'04' device is a punch. X'02' device is a reader.

Figure 25. DTFCP: DISK= parameter omitted (Part 1 of 2)

Bytes	Bits	Function
43		X'F1' device is a reader or tape. X'F0' device is other type.
44	0	1 = input, 0 = output.
	1	1 = eject needed for a reader-punch. 0 = no eject.
	2	1 = not first pass, 0 = first pass.
	3	1 = 2 I/O areas, 0 = 1 I/O area.
	4	1 = 2540 punch.
	5	1 = SYSLST or SYSPCH
	6	1 = SYSLST or SYSPCH on output tape.
	7	1 = TLBL specified and tape is labeled
45-47		IOAREA2 address.
48-55		CCW.
End-of-table if DTF is defined as output file and DEVADDR is not equal to SYSPCH.		
56-63		2540 punch error recovery CCW 1.
64-71		2540 punch error recovery CCW 2.
65-67		EOF address, input only.
End-of-table if DTF is defined as input file.		
72-151		80-byte card image, savearea 1.
152-231		80-byte card image, savearea 2.

Figure 25. DTFCP: DISK= parameter omitted (Part 2 of 2)

CPMOD Macro

The CPMOD macro is used with IBM internal programs only. Therefore, the CPMOD macro is not documented in any Systems Reference Library publications--see Compiler Files.

The CPMOD macro generates nine different logic modules. Each CPMOD logic module is flowcharted and described in detail. Output modules where REPLY=NO is specified are not flowcharted because they are the same as the other output modules except for the 2520 and 2540 punch error recovery procedures.

The GET-PUT logic modules for two I/O areas, with IOPTR and REPLY omitted handles all other cases except those modules where IOPTR=YES is specified.

The parameters of the CPMOD macro are keyword parameters. Because keyword parameters are specified, the parameter cards after the header card may appear in any order. This group of cards generates

the necessary logical IOCS CPMOD during assembly.

The parameter cards following the CPMOD header card have key entries in the operand field. All cards used in the CPMOD macro instruction, except the last, have continuation punches in column 72.

NAME: If NAME is specified, it is generated as the CSECT name of the module. No other entry points are generated.

TYPEFLE=INPUT or OUTPUT: If INPUT is specified, IOCS generates a module to handle input only. If OUTPUT is specified, IOCS generates a module that handles both input and output. If the TYPEFLE parameter is omitted, OUTPUT is assumed.

IOAREA2=YES or Omitted: If YES is specified, the generated logic module provides automatic switching between two I/O areas. If this parameter is omitted, one I/O area is assumed.

IOPTR=YES or Omitted: If YES is specified, the I/O area address is assumed to be in register 0. This parameter allows the user to change the I/O area address used with 1 I/O area. If IOPTR is specified in CPMOD, it must also be specified in the DTFCP macro.

RETRY=NO or Omitted: If NO is specified, the IOCS error recovery procedures for the IBM 2520 and the IBM 2540 punch devices are eliminated from the generated module. If the parameter is omitted, the 2520 and the 2540 punch error recovery procedures are included on the generated module. When the file is defined for input, this parameter is ignored.

DISK=YES or Omitted: If YES is specified, a module is generated that supports the 2311 and 2314 devices. If omitted, the module generated is the same as in the tape resident system.

RDONLY=YES or Omitted: If omitted, the module generated by the CPMOD macro is not reentrant. To be reentrant, general register 13 is used to point to a 72-byte, doubleword aligned, savearea unique to the particular task using the module. In this way several tasks can asynchronously use the same module if each task provides its own savearea.

CPMOD: GET Macro, Two I/O Areas Chart MA

Objective: To cause two I/O areas to be loaded from an input device and to allow overlap.

Entry: From a GET macro expansion issued by the problem program.

Exit: To the problem program, or to user's EOF routine.

Method: This routine is similar to the PUT two I/O areas routine, except it does not contain the logic to support a printer or punch. It contains only the logic necessary to retrieve records from an input file.

During the first pass, both I/O areas are read into. Overlap is made possible by allowing the second I/O area to be filled while the first one is being processed. A wait for I/O complete is issued to allow the previous read to be completed before processing the data of that area.

When an end-of-file is sensed, the user's registers are restored and a branch to the user's EOF routine is made.

All user's registers are saved upon entry into CPMOD and are restored when control is returned to the problem program.

CPMOD: GET Macro, One I/O Area Chart MB

Objective: To read a logical record.

Entry: From a GET macro issued by the problem program.

Exit: To the problem program or to the user's EOF routine.

Method: This routine makes a logical record available to the user in the I/O area. When an end-of-file is sensed, a branch to the user's EOF routine is made. The routine is similar to the GET two I/O areas routine except the logic to handle I/O area switching is not included.

CPMOD: GET Macro, IOPTR=YES Chart MC

Objective: To read a record into the I/O area pointed to by register 0 (IOPTR).

Entry: From a GET macro expansion.

Exit: To the problem program or EOF address.

Method: When the parameter IOPTR=YES is included, the CP module generated allows the user to use any area in main storage (other than a storage protected area) as an I/O area. The I/O area is pointed to by the address loaded into register 0. With this exception, this routine is the same as the GET with one I/O area.

CPMOD: PUT Macro, Two I/O Areas Charts MD-MF

Objective: To write a logical record with overlap.

Entry: From a PUT macro expansion, or from an SD workfile close.

Exits:

1. To the problem program.
2. To the user's EOF address, if EOF occurs on an output device other than a printer.

3. To \$\$BCMT07 if EOF occurs on an output tape assigned as SYSLST or SYSPCH.
4. To \$\$BERRTN error recovery routine, if an error occurs on a 2540 or 2520 punch.
5. To \$\$BERRTN if the upper extent is exceeded on a DASD file.

Method: If entered for the first time, the PUT routine performs an I/O operation immediately. (After the initial entry, this I/O operation is bypassed.) When the I/O operation is complete, the routine checks for an EOF condition. If an end-of-file is detected on a unit record device (other than a printer), control is passed to the user's end-of-file routine. If an end-of-file is detected on a magnetic tape device assigned to SYSPCH or SYSLST, the PUT routine fetches phase \$\$BCMT07 to determine if an alternate device is available. If it is not necessary to handle an EOF condition, the addresses of the two I/O areas are exchanged, another I/O operation is performed, and control is returned to the problem program.

The I/O subroutine first tests to determine the device type, and appropriate action is taken. If the device is a DASD, a routine is initialized to determine if various specified limits have been exceeded, and to update the seek address and count. The input/output operation is performed, and control returns to the problem program.

If an error occurs and the device is an IBM 2540 or a 2520 punch, the error recovery transient, \$\$BERRTN, is called. When an error recovery is complete, control returns to the module.

Note: If RETRY=NO is specified as a parameter in the CPMOD macro, the error recovery facility is not present in the module. \$\$BERRTN is also called to cancel the job if the upper extent is exceeded on a DASD file.

CPMOD: PUT Macro, 1 I/O Area Charts MG-MH

Objective: To write a logical record.

Entry: From PUT macro expansion, or from an SD workfile close.

Exits:

1. To the problem program.
2. To the user's EOF address if EOF occurs on an input device.

3. To \$\$BCMT07 if EOF occurs on an output tape assigned as SYSLST or SYSPCH.
4. To \$\$BERRTN error recovery routine if an error occurs on a 2540 or 2520 punch.
5. To \$\$BERRTN if the upper extent is exceeded on a DASD file.

Method: The PUT routine saves the user's registers and then checks for various devices. If the device is a tape unit, the I/O operation is performed immediately. If the device is a DASD, the specified limits are checked and the seek address and count are updated.

If the device is a printer or punch, the ASA control character is picked up and converted to System/360 coding. (See Appendix F for ASA codes.)

Next, the specified control operation is performed. When the control operation is complete, the I/O operation indicated by the PUT macro instruction is performed. The user's registers are restored and the routine returns control to the problem program.

Error and EOF conditions are handled in the same manner as described in the PUT routine for two I/O areas.

CPMOD: PUT Macro, IOPTR=YES Charts MJ-MK

Objective: To write a logical record from the I/O area pointed to by register 0 (IOPTR).

Entry: From a PUT macro expansion or from \$\$BOSDC1 - Close SD Workfile.

Exits:

1. To the problem program.
2. To the user's EOF routine if EOF occurs on punch or DASD device.
3. To \$\$BCMT07 if EOF occurs on an output tape assigned as SYSLST or SYSPCH.
4. To \$\$BERRTN error recovery routine if an error occurs on a 2540 or 2520 punch.
5. To \$\$BERRTN if the upper extent is exceeded on a DASD file.

Method: When the parameter IOPTR=YES is included, the CP module generated allows the user to use any area in main storage (other than a storage protected area) as an

I/O area. The I/O area is pointed to by the address loaded into register 0. With this exception, this module is the same as the PUT with one I/O area.

DEVICE INDEPENDENT SYSTEM FILES

Device Independent (DI) system files are files assigned to the system logical units SYSRDR, SYSIPT, SYSPCH, and SYSLST. The files are defined by a DTFDI macro for fixed unblocked records on the following devices:

- 1443 Printer
- 1404 Printer
- 1442 Reader-punch
- 1443 Printer

- 2311 Disk drive
- 2314 Multiple disk drive
- 2400 Series magnetic tape drive (7 or 9 track) - forward read only.
- 2501 Reader
- 2520 Reader punch
- 2540 Reader punch

DTFDI Macro (Figure 26)

The DTFDI macro defines the file for device independent system units and generates a table containing the information necessary to describe the file for processing by the DIMOD logic module.

Bytes	Bits	Function
0-15		CCB.
16	0-1	Not used.
	2	COBOL open; ignore option.
	3	Not used.
	4	DTF table address constants relocated by OPENR.
	5-7	Not used.
17-19		Address of logic module.
20		DTF Type = X'33'
21		Open/Close indicators - X'02' = input, X'00' = output.
22-28		Symbolic filename.
29		DASD device indicators X'01' = 2314 X'00' = 2311
30-35		DASD address of Format 1 label.
36-37		DASD volume sequence number.
38		Open communications switch - X'08'
39		Sequence number of current extent.
40		Sequence number of last extent, or X'80' for 1442 reader-punch.
41		Open indicator = X'20'.
42		Device type indicators
	0-1	Unused
	2	1 = DASD
	3	1 = tape
	4	1 = printer
	5	1 = punch
	6	1 = reader
	7	Unused.
43		Logic module device indicators X'F3' = DASD device X'F1' = reader or tape device X'F0' = other type device

Figure 26. DTFDI (Part 1 of 3)

Bytes	Bits	Function
44		Logic module option switches
	0	1 = input; 0 = output.
	1	1 = eject for RDR-PCH; 0 = no eject.
	2	1 = not first pass; 0 = first pass.
	3	1 = two I/O areas; 0 = one I/O area.
	6	1 = SYSLST/SYSPCH assigned to output tape.
	5	1 = SYSLST/SYSPCH.
	6	1 = Tape SYSLST/SYSPCH.
	7	Reserved for future use.
45-47		Alternate I/O area address.
48-51		Reserved for future use.
52-53		Extent lower head limit.
54-57		Extent upper head limit.
58-64		DASD seek address.
65-67		Users EOF address.
68-72		Control bucket CCHHR.
73		Logic module switches X'01' = input X'00' = output
74-75		Logic module constants X'0020' DASD output X'0018' DASD input X'0000' Non-DASD devices
76-80		Count field CCHHR.
81		Key length.
82-83		Data length.
84-87		Instruction to load IOREG with correct I/O area address.
88-111		Seek, Search, TIC CCWs.

Figure 26. DTFDI (Part 2 of 3)

Bytes	Bits	Function
112-119		Input/output CCW.
120-127		Second output CCW.
128-151		Verify CCWs for output.
152-159		Error CCW1.
160-167		Error CCW2.
168-231		Savearea (64 bytes).
232-235		DC A(WLRERR) if WLRERR=Address. B 28(15) if ERROPT= omitted. B 25(15) if ERROPT=SKIP. B 28(15) if ERROPT=IGNORE.
236-239		DC A(ERROPT) if ERROPT=Address. B 0(15) if ERROPT= omitted. B 24(15) if ERROPT=SKIP. B 28(15) if ERROPT=IGNORE.

Figure 26. DTFDI (Part 3 of 3)

DIMOD Macro

Device independent files provide users of DOS Assembler with the same capabilities already extended users of COBOL, FORTRAN, and PL/I in files defined by the DTFCP macro. Like the CP logic module, the device independent module generated by a DIMOD macro provides the logic to accomplish the logical IOCS functions of GET and PUT.

DIMOD: GET Macro, 1 I/O Area Charts NA-NB

Objective: To read the next sequential logical record into an input area specified by the user.

Entry: From a GET macro expansion.

Exit:

- To the next sequential instruction following the GET macro expansion in the problem program after a record is read into the input area.

- To the user's EOF routine if an end-of-file condition occurs.
- To cancel via an SVC 50 if an error occurs and the ERROPT parameter is not specified in the DTFDI macro.

Method: The GET routine first determines the device type specified in the DTFDI table. If the device is DASD, the routine validates the record number and updates the CCHHR seek address if necessary. The routine then issues an SVC 0 and waits for the I/O operation to finish. If the device is DASD, the record number is increased by one at this time in preparation for the next GET.

A check is made for error and end-of-file conditions. Errors are handled in the manner specified by the user in the DTFDI macro parameters. An end-of-file condition causes the GET routine to exit to the address of user's EOF routine instead of to the address of the next sequential instruction in the problem program.

DIMOD: GET Macro, 2 I/O Areas Charts NC-ND

Objectives: To read the next sequential logical record into an input area specified by the user, and to provide processing overlap by alternately reading records into one of the specified input areas and then into the other.

Entry: From a GET macro expansion.

Exit:

- To the next sequential instruction in the problem program following the GET macro expansion.
- To the user's EOF routine if an end-of-file condition occurs.
- To cancel via an SVC 50 if an error occurs and the ERROPT parameter is not specified in the DTFDI table.

Method: The GET routine, when two I/O areas are employed, functions in the same manner as a GET with one I/O area. The difference between the two routines is the utilization of two I/O areas by alternately exchanging the addresses of the two areas each time a GET macro is issued. Therefore, to the user it appears that the record is available in the same input area for every GET.

DIMOD: PUT Macro, 1 I/O Area Charts NE-NG

Objective: To write the next sequential logical record to the file from a user specified output area.

Entry: From a PUT macro expansion in the problem program.

Exit:

- To the next sequential instruction in the problem program following the PUT macro expansion after the record is written to the file.
- To phase \$\$BCMT07 if an EOF condition is reached on an output tape assigned to either SYSLST or SYSPCH.
- To phase \$\$BERRTN if a punch error occurs.
- To the user's EOF routine if an EOF condition is reached on devices other than SYSLST or SYSPCH assigned to an output tape.

- To the user's WLR or ERROPT routine, if specified in the DTFDI macro parameters, to handle other error conditions.

Method: The PUT routine determines the device type and selects the proper I/O operation to write the record to the file:

- If the device is a reader or a magnetic tape, an SVC 0 is issued directly.
- If the device is a DASD, the number of the record to be written is checked and the CCHHR seek address is updated, if necessary, before the SVC 0 is issued.
- If the device is a printer or punch, the control code is determined, converted to System/360 code if necessary (refer to Appendix F), and the control function is performed. When the control function is complete, an SVC 0 is issued to write the record.

After the record is written, tests determine if a unit exception or error condition occurred. If a unit exception occurred, the following action is taken:

- For printers, the unit exception is ignored.
- For output tapes assigned to either SYSLST or SYSPCH, phase \$\$BCMT07 is fetched to determine the availability of an alternate device.
- For all other devices, the address of the user's EOF routine is obtained from the DTF table and used as the return address from the PUT routine.

Other error conditions are handled as specified by the user in the DTFDI macro parameters.

DIMOD: PUT Macro, 2 I/O Areas Charts NH-NK

Objectives: To write the next sequential logical record to the file from a user specified output area and to provide overlap through the use of two I/O areas.

Entry: From a PUT macro expansion in the problem program.

Exit:

- To the next sequential instruction in the problem program following the PUT macro expansion after the record is written to the file.

- To phase \$\$BCMT07 if an EOF condition is reached on an output tape assigned to either SYSLST or SYSPCH.
- To the user's EOF routine if an end-of-file condition is reached on devices other than SYSLST or SYSPCH assigned to an output tape.
- To phase \$\$BERRTN if a punch error occurs.
- To the user's WLR or ERROPT routine, if specified in the DTFDI macro parameters, to handle other error conditions.

Method: The PUT routine, when two I/O areas are employed, functions in the same manner as a PUT with one I/O area. The difference between the two routines is in alternately exchanging the addresses of the two I/O areas each time a PUT macro is issued. Even though two areas are used, the exchange of addresses makes it possible for the user to insert the record to be written at the same address for each PUT macro issued.

\$\$BERRTN: Punch Error Recovery Chart PA

Objectives: To attempt a recovery from a punch error condition on either a 2520 or 2540 punch, or to cancel the job when an extent is exceeded while writing on a DASD.

Entry: From an output DIMOD, or an output CPMOD if RETRY=NO is not specified.

Exit: To the calling routine after the error is corrected or to the message writer phase \$\$BOMSG1 if an extent is exceeded.

Method: Phase \$\$BERRTN first determines the type of device on which the error occurred.

If a punch error occurred on a 2520, the message 4000I is printed to indicate that a retry is being made and the card in error is repunched.

If a punch error occurred on a 2540, the card in error is repunched, the retry message 4000I is printed, and the card following the error card is repunched. (It is necessary to repunch the card following the error card because it is in the process of being punched when the punch error is detected and both cards are stacker selected.)

Control returns to the calling routine via an SVC 11 when the punch error is corrected.

If an error occurred on a 2311 or 2314 DASD device, preparation is made to print message 4887I and the message writer phase \$\$BOMSG1 is fetched.

INITIALIZATION AND TERMINATION

Device independent files are opened by the same logical transients used to open compiler files. The close routine for a DTFDI file provides for re-initialization of the DTF table. This procedure removes from the DTFDI files the restriction on reopening imposed on DTFCP files.

\$\$BOCP01: Open Device Independent Files - Phase 1 Charts QA-QC

Objective: To open DOS device independent files defined by either DTFCP or DTFDI (Version 1 DTFCP excluded).

Entry: From the Open Monitor.

Exits:

1. To \$\$BOCP02 if the DTF is a tape resident system type DTFCP. (This DTF table type is generated for DOS if the DISK= parameter is omitted from the DTFCP macro.)
2. To \$\$BOCPT1 if the device type is tape and the file type is input.
3. To \$\$BOCPT2 if the device type is tape and the file type is output.
4. To \$\$BOPEN if the device type is DASD or additional files remain to be opened.
5. To the problem program if no more files remain to be opened.
6. To \$\$BOMSG1 to issue the error message, 4883I INVALID LOGICAL UNIT.

Method: This routine is called by the Open Monitor when the file is found to be a DTFCP or DTFDI type (Version 1 DTFCP excluded). Because the logic modules for both file types provide for device independence, this phase is needed to determine the device type associated with the logical unit. The PUB is located for the logical unit and the DTF is initialized according to the device type (refer to Figures 23, 24, and 25).

The routine at the label UNTRCP initializes an I/O area. This routine

adjusts the addresses of the I/O area in the CCW and the alternate I/O area to bypass the control character when it is determined that the device is a printer or punch. This routine also reinitializes the DTF table if the device is a reader-punch.

The LDIOREG routine initializes the user's I/O register. If the device is tape, a check determines if Job Control has already opened the device. If the file is open and if there are more files to open, the Open Monitor, \$\$BOPEN, is fetched. If no more files remain to be opened, control returns to the problem program. If the file has not been opened, \$\$BOCP1 is called to open the input files, and \$\$BOCP2 is called to open the output files.

If the device is a 1403 printer with the Universal Character Set (UCS) a set mode command is given to suppress data checks.

If device type cannot be determined, an error message is issued and the job is canceled. If the device is a tape drive and found to be file-protected, an error message is issued and the operator may either insert a ring in the tape reel or cancel.

\$\$BOCP02: Open Device Independent Files - Phase 2 Charts QD-QF

Objective: To open the DTFCP tables of the tape resident system type.

Note: A tape resident type DTFCP table is generated for DOS if the DISK= parameter is omitted from the DTFCP macro.

Entry: From \$\$BOCP01, or from \$\$BOMSG1.

Exits:

1. To \$\$BOPEN, if additional files remain to be opened.
2. To the problem program, if no more files remain to be opened.
3. To \$\$BOMT06, if the device type is tape and the CP type is input.
4. To \$\$BOMSG1, to print the following messages:

4883I INVALID LOGICAL UNIT

4884D FILE PROTECT RING NEEDED

Method: This phase functions in the same manner as \$\$BOCP01, except when it is entered from a message routine. If this is

the case, a sense command is issued and a test is made for file protect. If the file is not file-protected, \$\$BOMT06 is fetched. If it is, a message is printed, via \$\$BOMSG1.

\$\$BOCP11: Open DTFCP (Version 1 Only) Phase 1 Charts QG-QJ

Objective: To open Version 1 DTFCP tables for disk resident systems.

Entry: From the Open Monitor, or from \$\$BOMSG1.

Exits:

1. To \$\$BOCP12, if DTFCP is Version 1 tape resident system type (this DTFCP table type is generated for DOS if the DISK= parameter is omitted from the DTFCP macro).
2. To \$\$BOPEN, if the device type is DASD or additional files remain to be opened.
3. To the problem program, if no additional files remain to be opened.
4. To \$\$BOMT06, if the device type is tape.
5. To \$\$BOMSG1, to put out the following messages:

4883I INVALID LOGIC UNIT

4884D FILE PROTECT RING NEEDED

Method: This phase is called by the Open Monitor when the file is found to be a Version 1 DTFCP table type. The phase performs the same functions as \$\$BOCP01, except that it applies only to Version 1.

\$\$BOCP12: Open DTFCP (Version 1 Only) Phase 1 Charts QK-QL

Objective: To open Version 1 DTFCP tables of the tape resident system type.

Note: A tape resident type DTFCP table is generated for DOS if the DISK= parameter is omitted from the DTFCP macro.

Entry: From \$\$BOCP11, or \$\$BOMSG1.

Exits:

1. To \$\$BOMT06, if device type is tape.

2. To \$\$BOPEN, if additional files remain to be opened.
3. To the problem program, if no more additional files remain to be opened.
4. To \$\$BOMSG1, to put out the following messages:

4883I INVALID LOGIC UNIT

4884D FILE PROTECT RING NEEDED

Method: This phase is called by \$\$BOCP11 when the file is found to be a TOS Version 1 DTFCP. The phase performs the same functions as \$\$BOCP02, with the exception that it applies only to Version 1.

\$\$BOCP11: Open DTFCP and DTFDI Input Tape Charts QM-QS

Objective: To open input tape files.

Entry: From either \$\$BOCP01 (if the file is defined by a DOS DTFCP or DTFDI and the device is a tape) or \$\$BOCPM1 (after a message has been issued and a non-cancel reply has been received).

Exits:

1. To the Open Monitor, \$\$BOPEN, if more files remain to be opened.
2. To the problem program, if no more files remain to be opened.
3. To the message writer, \$\$BOCPM1, when an error condition is encountered.

Method: This routine performs various open procedures after determining:

1. If the user has provided a TLBL (or TPLAB) card.
2. The position of the tape with respect to its loadpoint.
3. Whether the tape contains standard labels or a tapemark.

If an unlabeled input tape file is to be opened, this routine skips the tapemark, if present, and positions the tape at the first data record.

If a standard labeled input tape file is to be opened and a TLBL (or TPLAB) card has been provided, this routine uses the label information supplied to check the HDR1 label. It skips the additional header, volume, and user labels and positions the tape at the first data record. Only single

volume files or multifile volumes can be opened by this routine.

\$\$BOCP2: Open DTFCP and DTFDI Output Tape Charts QT-QY

Objective: To open output tape files.

Entry: From either \$\$BOCP01 (if the file is defined by a DOS DTFCP or DTFDI for output and the device is a tape) or \$\$BOCPM2 (after a message has been issued and a non-cancel reply has been received).

Exits:

1. To the Open Monitor, \$\$BOPEN, if more files remain to be opened.
2. To the problem program, if no more files remain to be opened.
3. To the message writer, \$\$BOCPM2, when an error condition occurs.

Method: This routine performs various open procedures after determining:

1. If the user has provided a TLBL (or TPLAB) card.
2. The position of the tape with respect to its loadpoint.
3. Whether the tape contains standard labels or a tapemark.

If an unlabeled output tape file is to be opened, this routine determines if the tape has a tapemark. If the tape has a tapemark, it is retained. If the tape lacks a tapemark, no tapemark is written.

If a standard labeled output tape file is to be opened, this routine first checks the expiration date. If label information from a TLBL (or TPLAB) card has been provided, a new standard HDR1 label is written on tape. If label information has not been provided, a dummy HDR1 label is written on tape. No additional standard header or user header labels are written.

\$\$BOCP3: Close DTFCP or DTFDI Tape Files Charts QZ-RB

Objective: To close DTFCP and DTFDI input and output tape files.

Entry: From the Close Monitor after it is determined that the file is either a DTFCP DOS or TOS type or DTFDI type, and the device is a tape.

Exits: To the Close Monitor if additional files remain to be closed, to the problem program if no additional files remain to be closed, or to the message writer, \$\$BOMSG1, when an incorrect TLBL (or TPLAB) card image record is found.

Method: This routine performs various close procedures after determining whether the rewind option has been specified, whether a TLBL (or TPLAB) card has been provided, whether the tape file is input or output, and whether the tape has an EOF1 trailer label.

If the file is an input tape, this routine checks to determine if the rewind option has been specified. If the rewind option has been specified, the tape is rewound. If it has not been specified and the tape has standard trailer labels, all standard labels and user trailer labels are bypassed and the tape is positioned to the first record after the tapemark which follows the set of trailer labels.

If the file is an unlabeled output tape, this routine writes a tapemark and rewinds the tape if the rewind option has been specified. If the file is a standard labeled output tape, this routine writes a tapemark. An EOF1 trailer label is built and written on tape followed by two tapemarks. If the rewind option has been specified, the tape is rewound. If it has not been specified, the tape is backspaced one record.

\$\$BCLOSP: Punch File Close Charts RD-RG

Objective: To close DTFCP, DTFDI, and DTFCD punch files and recover possible errors occurring when the last card in the file is punched.

Entry: From the Close Monitor, \$\$BCLOSE, to the entry point BEGINRTN.

Exits: To the Close Monitor \$\$BCLOSE if there are additional files to be closed. If no additional files remain to be closed, return is to the problem program.

Method: This routine first determines the device type. If the device is a 1442 punch or a 1442 reader-punch, the routine exits to the Close Monitor or to the problem program if no additional files remain to be closed. Depending upon the file type and whether there are one or two I/O areas, the following actions are taken:

1. For a DTFCD file with two I/O areas where the device is a 2540 punch, an error, if it has occurred, is corrected

on the card preceding the last card. Then, any error detected on the last card is corrected.

2. For a DTFCD file with one I/O area where the device is a 2540 punch, any error detected on the last card is corrected.
3. For a DTFCD file with two I/O areas where the device is a 2520 punch, any error detected on the last card is corrected.
4. For a DTFCD file with one I/O area where the device is a 2520 punch, no error recovery is needed and the routine takes the proper exit.
5. For a DTFCP and DTFDI files (one or two I/O areas) where the device is a 2540 punch, error recovery is performed first on the card preceding the last card punched. Then, any error detected on the last card is corrected.
6. For a DTFCP and DTFDI files (one or two I/O areas) where the device is a 2520 punch, any error on the last card is corrected.

In all cases, whenever an error card is repunched correctly, the message 4000I RETRY is printed on SYSLOG to inform the operator that the error was retried.

\$\$BOCPM1 and \$\$BOCPM2: DTFCP/DTFDI Message Writers Charts RH-RK

Objective: To write messages on SYSLOG and to process operator responses to the messages issued.

Entry: From the DTFCP and DTFDI tape open routines, \$\$BOCPT1 and \$\$BOCPT2.

Exits:

1. For certain messages, automatic job termination.
2. For CANCEL response or no 1052, job canceled.
3. For IGNORE, NEWTAP, RETRY responses or information-type messages, the next phase to be executed is fetched.

\$\$BOCPM1 Messages:

- 4112A - VOL SERIAL NO. ERROR
- 4113D - NO HDR1 LBL FOUND

4114A - FILE SEQ NO. ERROR
 4115A - SERIAL NO. ERROR
 4116A - VOLUME SEQ. ERROR
 4132D - ERROR IN FILE ID
 4133D - ERROR IN HDR LABEL

\$\$BOCPM2 Messages:

4110A - NO VOL1 LBL FOUND
 4112A - VOL SERIAL NO. ERROR
 4119A - FILE EXPIRED
 4144A - 1600 BPI TAPE MOUNTED - \$\$BOCPM2
 4184D - NEED FILE PROTECT RNG

Method: Both \$\$BOCPM1 and \$\$BOCPM2 message writer phases perform the same function and differ only in the messages written. Each phase writes, on SYSLOG, the appropriate message for the information it receives from the calling phase. The calling phase supplies as input a message number, a return address for each legal response, and the name of the next phase to be fetched. After the correct message and label information is written on SYSLOG, the message writers analyze and process the operator response and fetch the next appropriate phase.

\$\$BOMSG1: Disk Open Error Message Writer, Phase 1 Charts RL-RM

Objectives: To initialize the message output area, SYSLOG CCB and CCW's and to fetch the second phase of the message writer, \$\$BOMSG2.

Entry:

- From the DASD VTOC display phase, \$\$BODSPW.
- From a DASD open or close phase.
- From the DTFCP open phases \$\$BOCP01, \$\$BOCP02, \$\$BOCP11, or \$\$BOCP12.

Exit: To phase 2 of the Disk Open Error Message Writer, \$\$BOMSG2.

Method: The calling phase supplies the following information to the message writer:

- Register 0 contains the last four characters of the phase name requesting the message. On cancel messages,

register 0 does not have to be initialized. \$\$BO is assumed for the first four characters of the phase name.

- Register 2 contains the address of the DTF table for the current file.
- Register 3 contains the message code (in binary) for the message to be printed. This code is converted to the last two digits of the message number (XX in the example 4nXXI).
- Transient region +1185 contains the numeric decimal value assigned to the various open/close phases for message numbering. (n in the example 4nXXI.)
- Transient region +1000 contains the start of the CCB.

The message writer overlays the first 888 bytes of the transient region. Therefore, any information that the calling phase needs to save is located beyond that point.

This phase first saves the last four characters of the phase name requesting the message. It initializes the SYSLOG message output area with the organization type numeric code, DTF filename, symbolic unit, and constant. It builds the SYSLOG CCWs for writing the message and reading the response and determines if the required message is in this phase of the message writer. If it is not in this phase, the routine determines in which overlay phase the message is located (either \$\$BOMSG3, \$\$BOMSG4, \$\$BOMSG5 or \$\$BOMSG6) and fetches \$\$BOMSG2 to load the required overlay phase.

\$\$BOMSG2: Disk Open Error Message Writer, Phase 2 Charts RN-RQ

Objective: To issue an error message to the operator, read the operator's reply (if a 1052 is assigned to SYSLOG) or exit to the phase that requested the message (after insuring the validity of the operator's response). Also, to cancel the job either by operator request or if the message type indicates termination of the job.

Entry: From phase 1 of the Disk Open Error Message Writer, \$\$BOMSG1. Entered at MSGROUT2.

Exit: To the VTOC dump phase, \$\$BOVDMP, to phase 1 of the VTOC display routine, \$\$BODSPV, or to the DASD Open/Close phase initiating the message (if a cancel was not encountered).

Method: \$\$BOMSG1 supplied the following information to this phase:

- Register 1 contains the name (last four characters) of the message overlay phase to fetch if the required message appears in some phase other than \$\$BOMSG1.
- Register 3 contains the address of the message to be written on SYSLOG.

This phase determines the message type. It can be either a file overlap pack, wrong pack, or other.

For a file overlap type, the last character of the message is initialized to a 'D', the CCW is initialized to write the 44-byte file key, and the file overlap switch is set to NO-OP. The file overlap switch set to NO-OP allows a test for deleting an unexpired file later in the routine.

For wrong pack type, the message is initialized with the pack number and the wrong pack switch is turned on. This switch is interrogated later in the routine to test if the operator has mounted the next pack.

Next, the routine determines if the message to be written on SYSLOG is in main storage. If the message is not in main storage, the message overlay phase containing the required messages is loaded into main storage. The message overlay phases consist of \$\$BOMSG3, \$\$BOMSG4, \$\$BOMSG5 and \$\$BOMSG6. These phases contain messages only. The message is then moved to the SYSLOG output area and an SVC 0 is issued to type the message and read the reply.

If the message indicates the job is not to be canceled, the routine determines if the user wants a VTOC display. If he does

want a VTOC display, the routine issues an SVC 2 to fetch \$\$BODSPV, the DASD VTOC Display phase. If the user does not want a VTOC display, the routine tests for a D-type message.

If yes, the message return indicator is set, the address of the next phase name is retrieved, and a SVC 2 is issued to fetch the return phase. If the message is not a D-type, the routine tests the file overlap switch and the wrong pack switch as previously mentioned. During this portion of the routine, a B for bypass, or a D for delete is stored in the transient region +1186 for use by the calling phase.

The message writer issues an illegal response message for the following conditions:

1. Operator reply of IGNORE for a D-type message.
2. Message for index sequential file.
3. Equal file ID message.
4. No extent to be bypassed.
5. Next pack not mounted.

If the job is to be canceled, a test determines if the job control open switch (in communications region) is on. If so, a SVC 11 is issued to return to Job Control. If the switch is not on, the routine checks to determine if a request has been made for a VTOC dump. If yes, a SVC 2 is issued to call the VTOC Dump transient, \$\$BOVDMP. If a VTOC dump has not been requested, a SVC 6 is issued and the job is canceled.

The following message lists show the code (passed via register 3) together with the last two digits and action indicator of the associated message number. For reference purposes, the text of the message is also included.

Message Code	Message Number	Message
0	44A	OVERLAP ON UNEXPRD FILE
1	55A	WRONG PACK, MOUNT nnnnnn
2	40A	EXTENT OVERLAP ON ANOTHER
3	41A	EXTENT OVERLAPS ON VTOC
4	42A	NO MATCHING EXTENT
5	33A	EQUAL FILE ID IN VTOC
6	66A	1 TRACK USER LBL EXTENT
7	59A	INVALID EXTENT
8	47A	EXTENTS NOT ON SAME UNIT
9	A	Reserved for A type msg.
15	84D	NEED FILE PROTECT RING
17	38D	USER HDR LBL IS NOT STD
18	39D	USER TRL LBL IS NOT STD
19	08D	NO UTLO FILE MARK FOUND
20	D	Reserved for D type msg.
21	D	Reserved for D type msg.
22	00I	NO RECORD FOUND
23	01I	NO RECORD FOUND
24	02I	NO RECORD FOUND
25	03I	NO RECORD FOUND
26	04I	NO RECORD FOUND
27	05I	NO RECORD FOUND
28	06I	NO RECORD FOUND
29	07I	NO RECORD FOUND
30	08D	NO RECORD FOUND
31	09I	NO RECORD FOUND
32	01I	NO LABEL SPACE IN VTOC
33	01I	NO FORMAT 1 LABEL FOUND
34	02I	NO FORMAT 2 LABEL FOUND
35	03I	NO FORMAT 3 LABEL FOUND
36	04I	NO FORMAT 4 LBL IN VTOC

Message Code	Message Number	Message
37	06I	NO STANDARD VOL1 LABEL
38	41I	EXTENT OVERLAPS ON VTOC
39	46I	DISCONT INDEX EXTENTS
40	51I	SYSUNITS NOT IN SEQUENCE
41	52I	DISCONT TYPE 1 EXTENTS
42	54I	DSKXTN ENTRY TABLE FULL
43	62I	NO PRIME DATA EXTENT
44	45I	TOO MANY EXTENTS
45	49I	DATA TRACK LIMIT INVALID
46	59I	INVALID EXTENT
47	60I	NO EXTENTS, ALL BYPASSED
48	61I	INVALID DLBL FUNCTION
49	63I	LOAD FILE NOT CLOSED
50	80I	INVALID FILE TYPE
51	81I	NO LABEL INFORMATION
52	83I	INVALID LOGICAL UNIT
53	90I	NO AVAILABLE JIB
54	87I	SYS FILE EXTENT EXCEEDED
55	35I	DELETED WORKFILE LABEL
56	34I	CURRENT FILE LBL DELETED
57	40I	EXTENT OVERLAP ON ANOTHER
58	36I	NO MORE AVAIL/MATCH EXTENT
59	48I	SYSIN/SYSOUT UNSUPPORTED
60	70I	LST EXTENT CD NOT INDX VOL
61	71I	EXTENT INFO NEEDED
62	72I	MOD AND DTF INCOMPATIBLE
63	58I	NO EXTENT FOR OUTPUT FILE

Chart 01. Magnetic Tape Open Routines

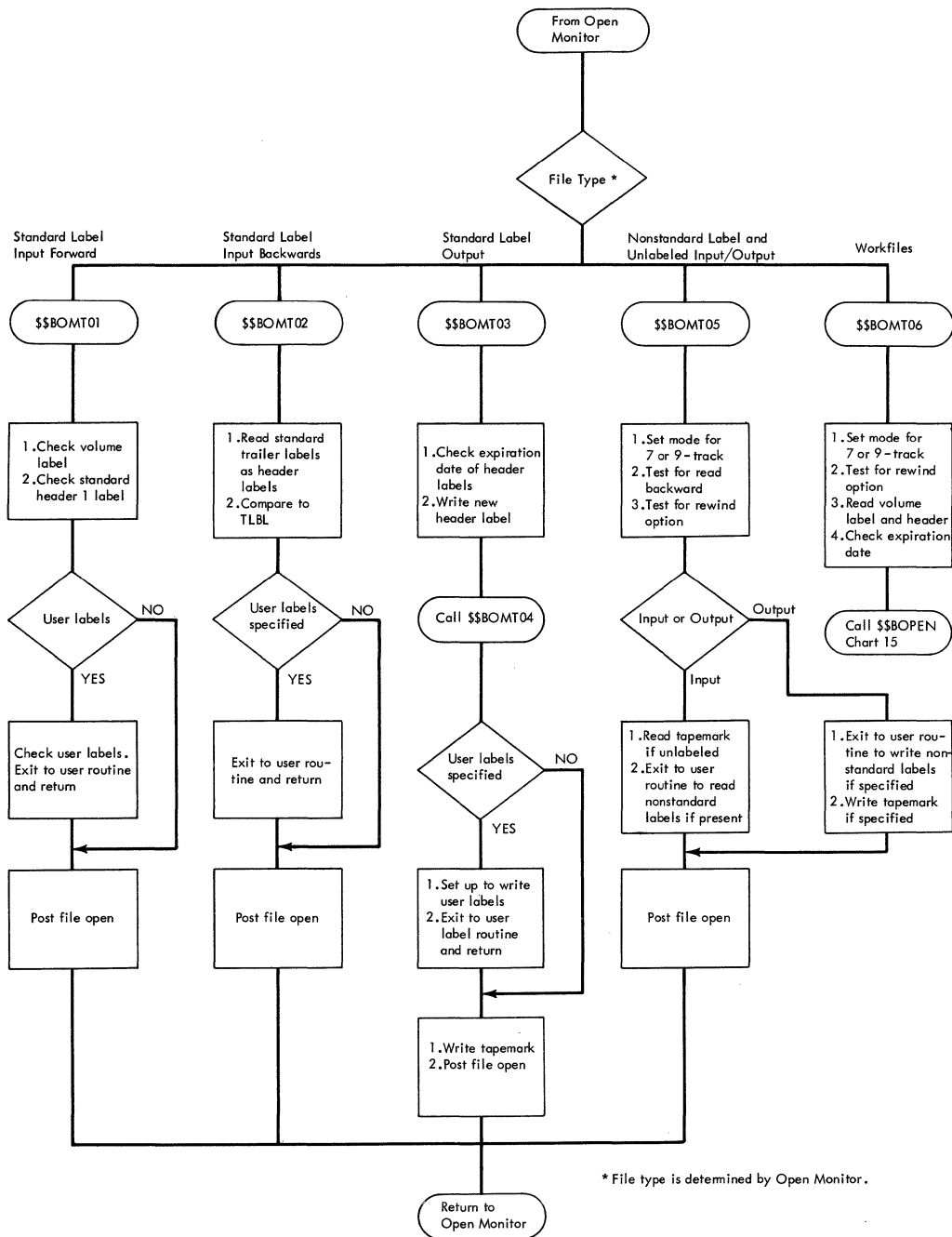
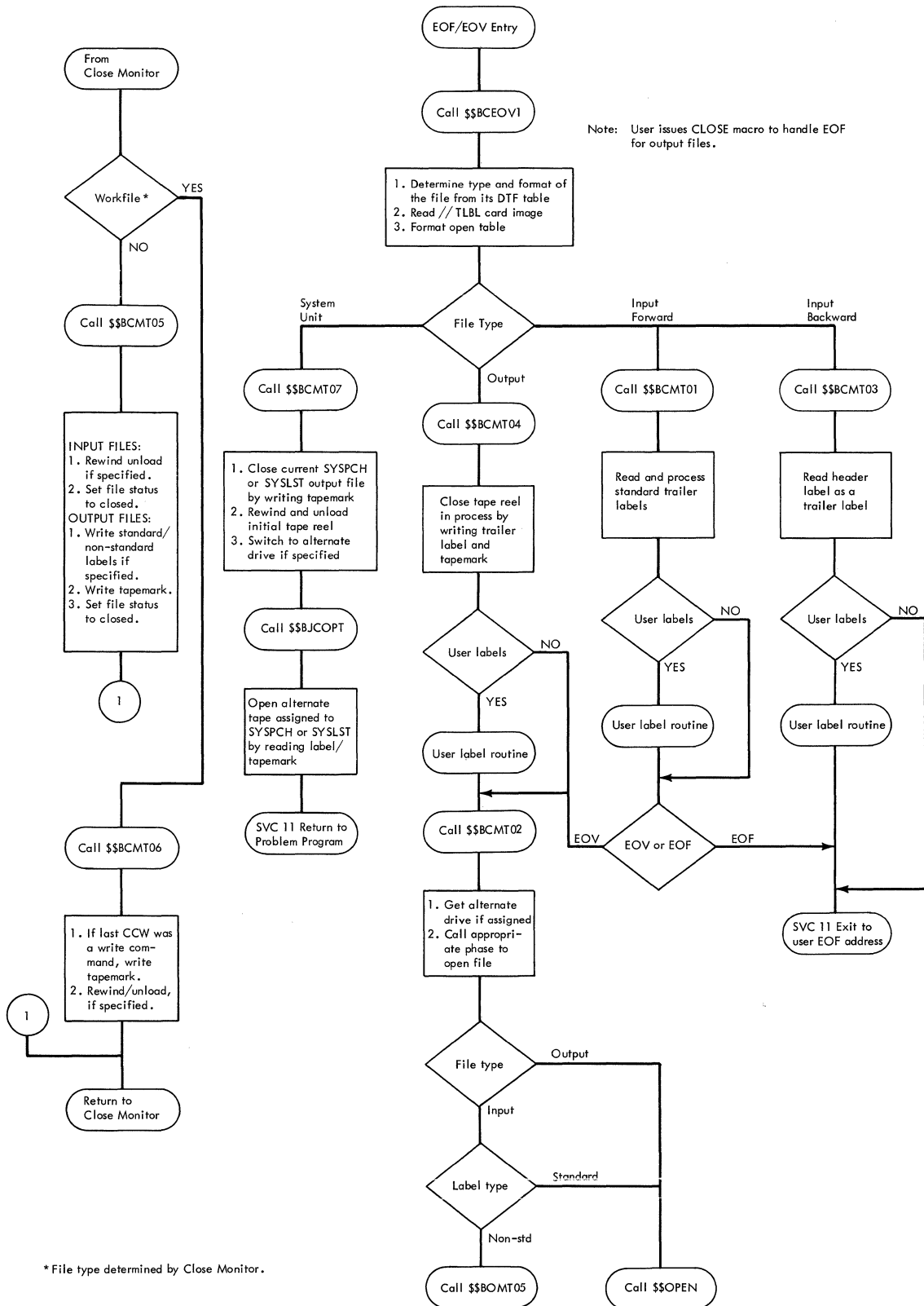
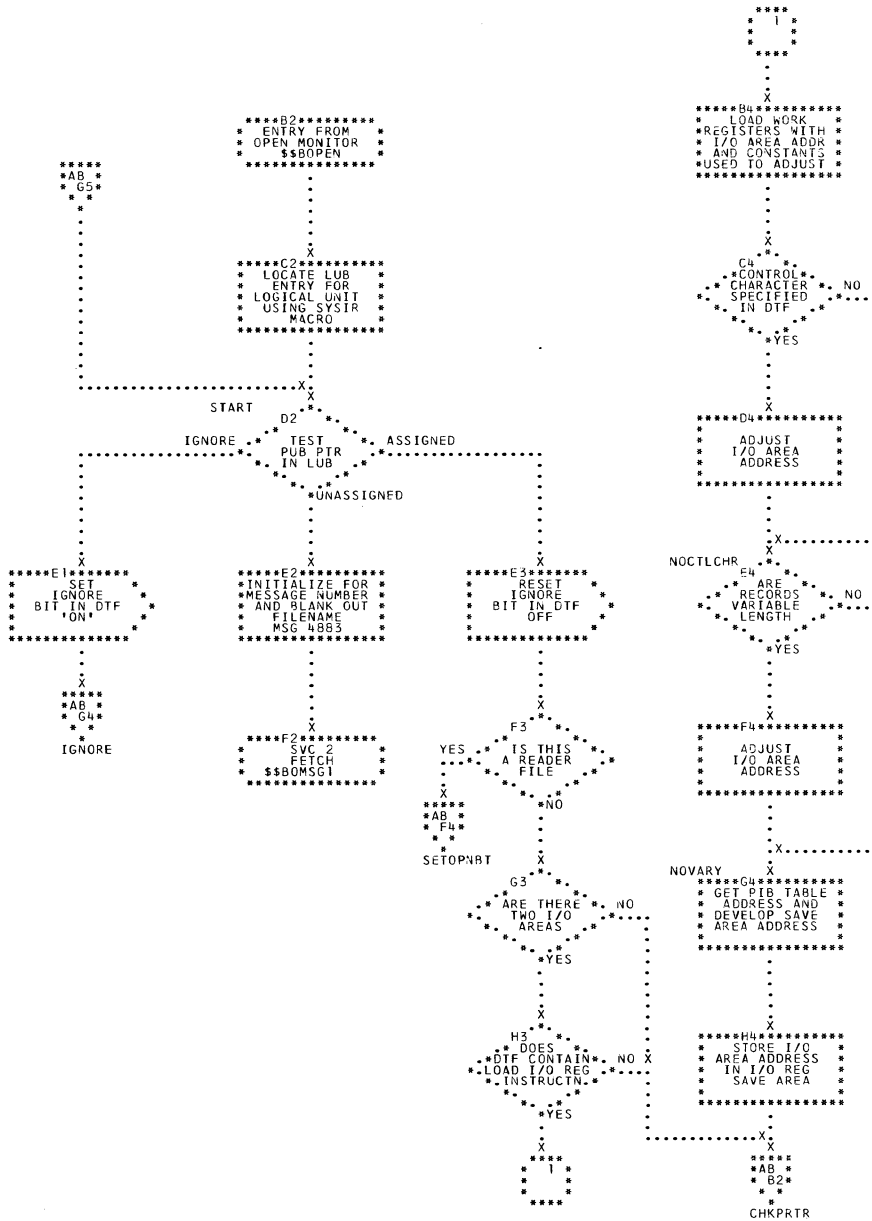


Chart 02. Magnetic Tape Close and EOF/EOV Routine

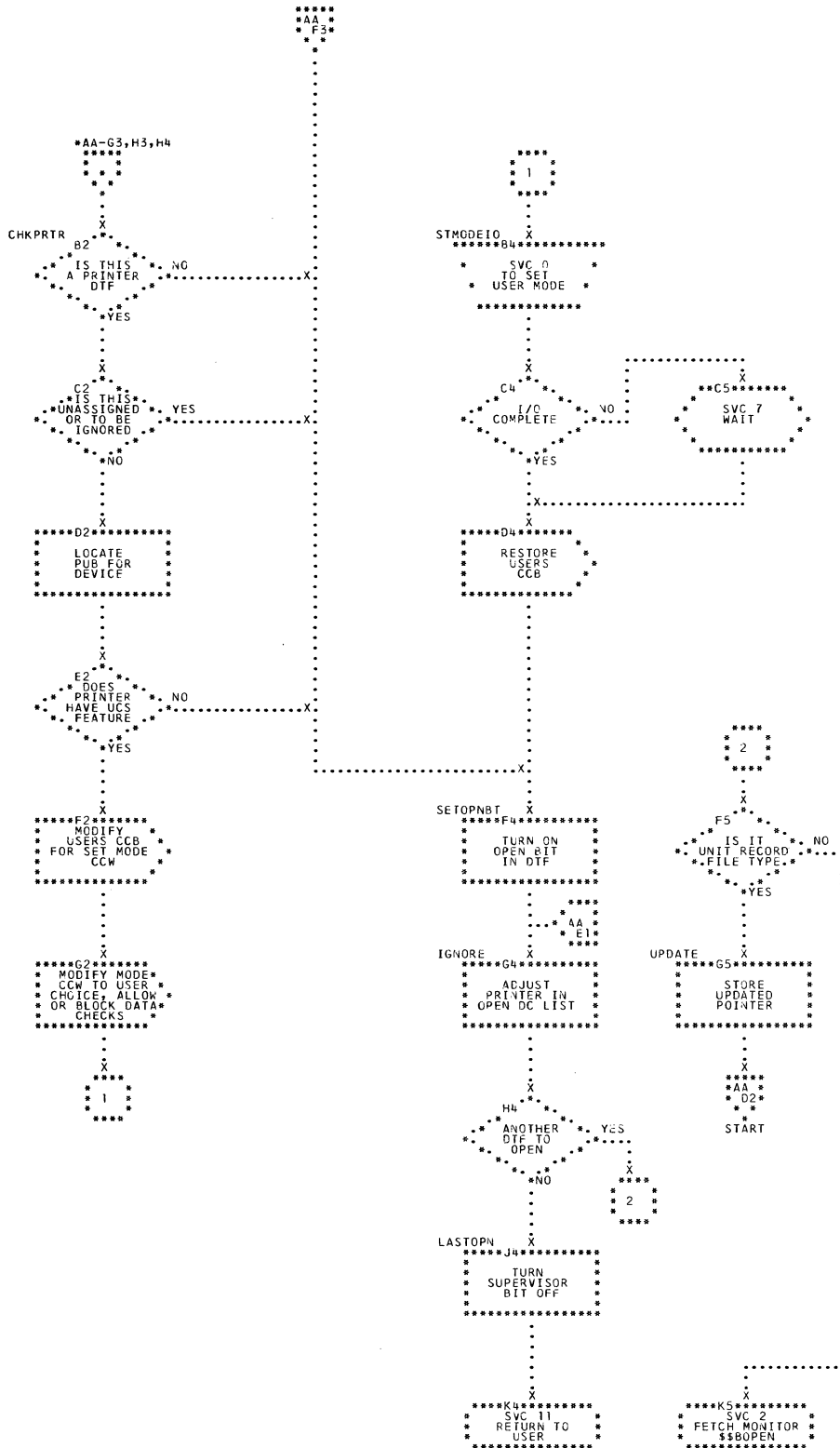


* File type determined by Close Monitor.

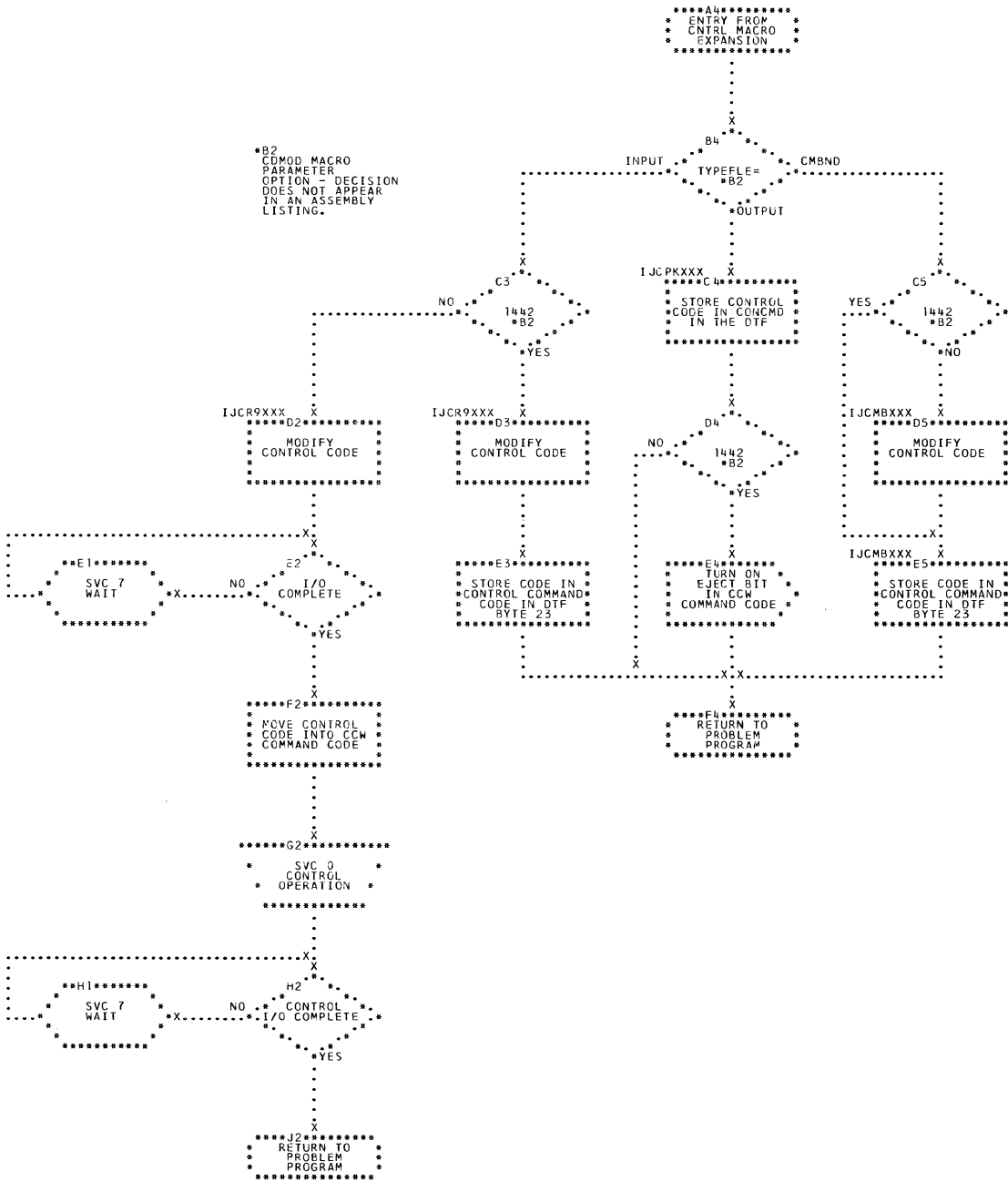
● Chart AA. \$\$BOUR01: Open Unit Record (Section 1 of 2)



● Chart AB. \$\$BOUR01: Open Unit Record (Section 2 of 2)

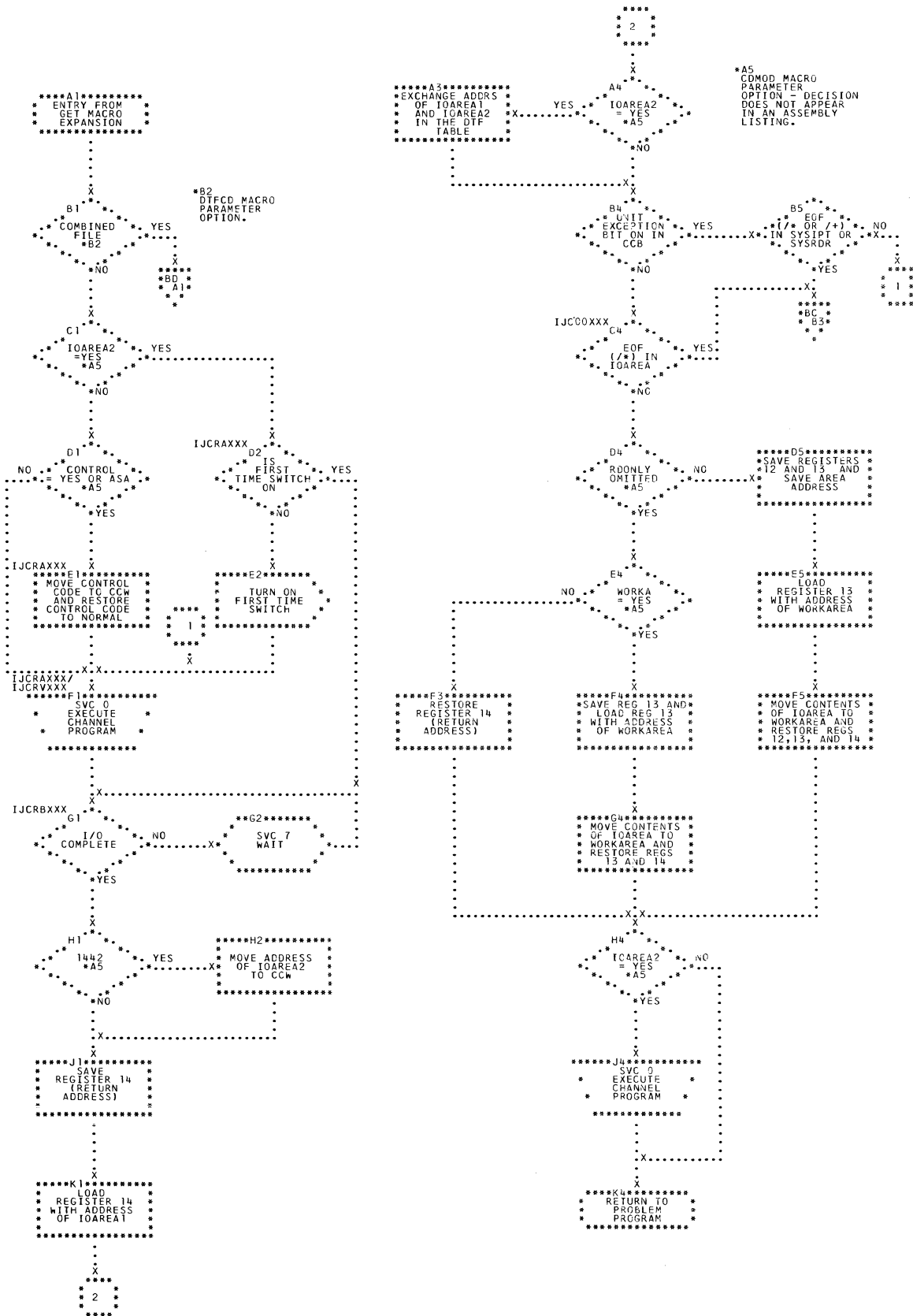


● Chart BÀ. CDMOD: CNTRL Macro

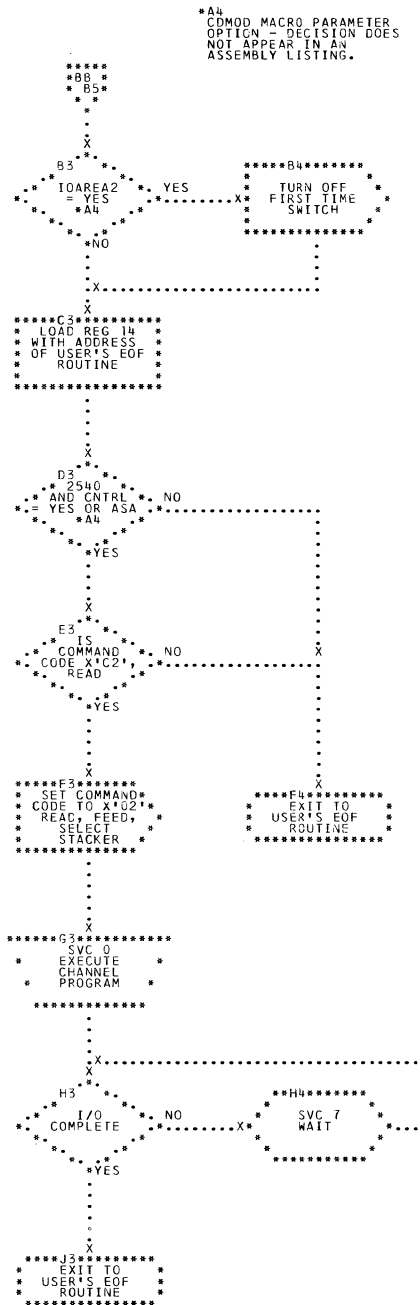


*B2
CDMOD MACRO
PARAMETER
OPTION - DECISION
DOES NOT APPEAR
IN AN ASSEMBLY
LISTING.

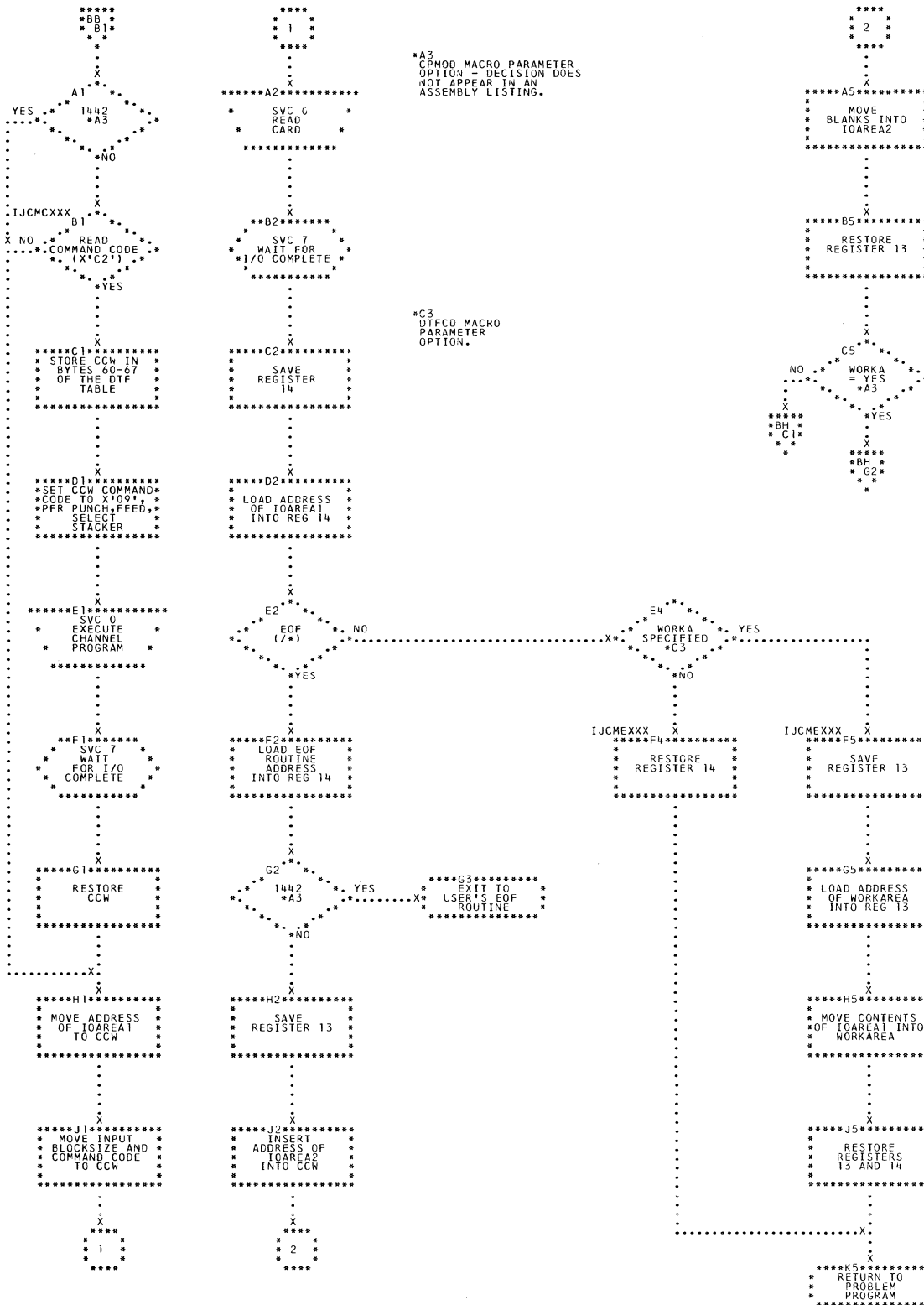
●Chart BB. CDMOD: GET Macro (Section 1 of 3)



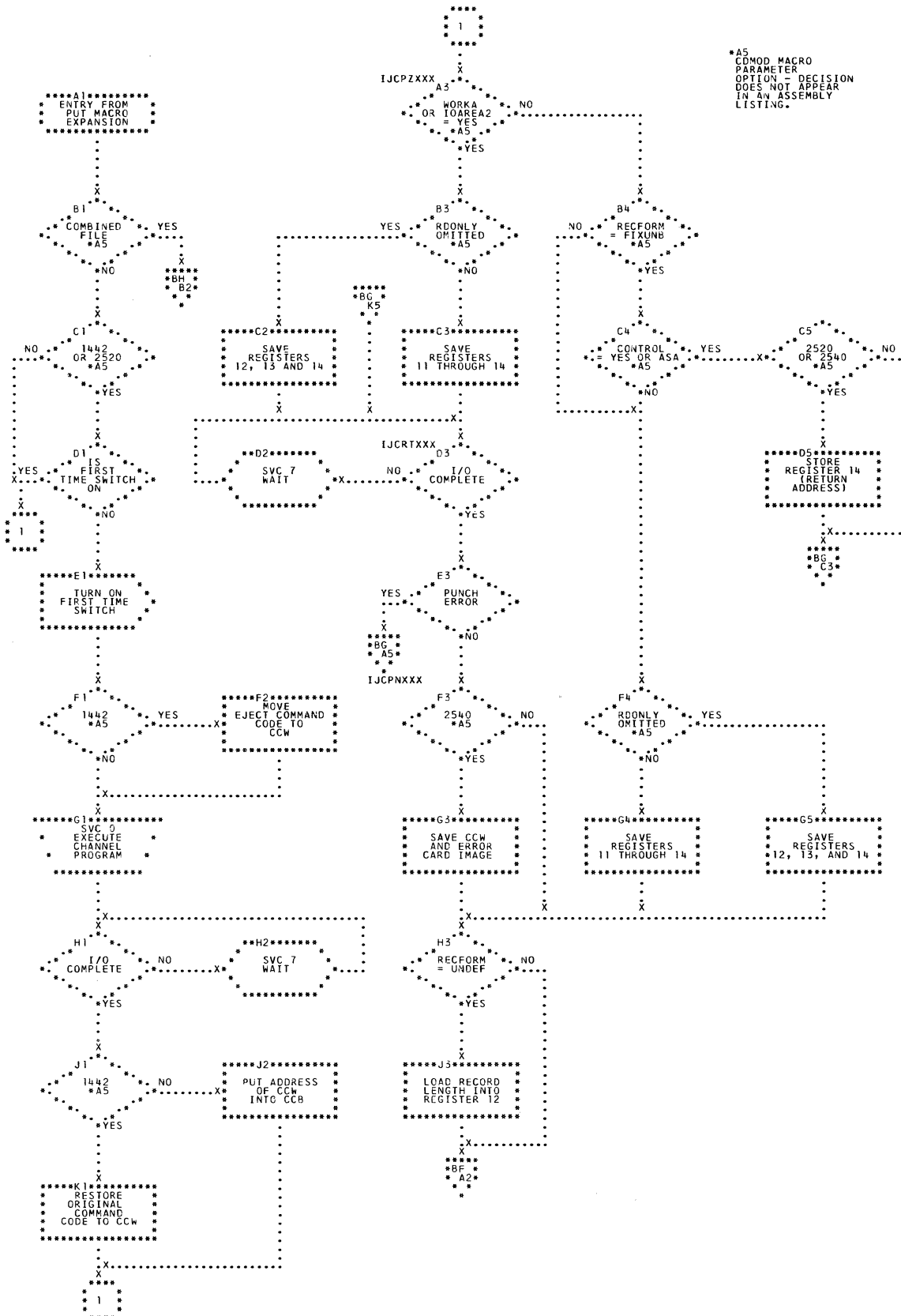
● Chart BC. CDMOD: GET Macro (Section 2 of 3)



●Chart BD. CDMOD: GET Macro (Section 3 of 3)

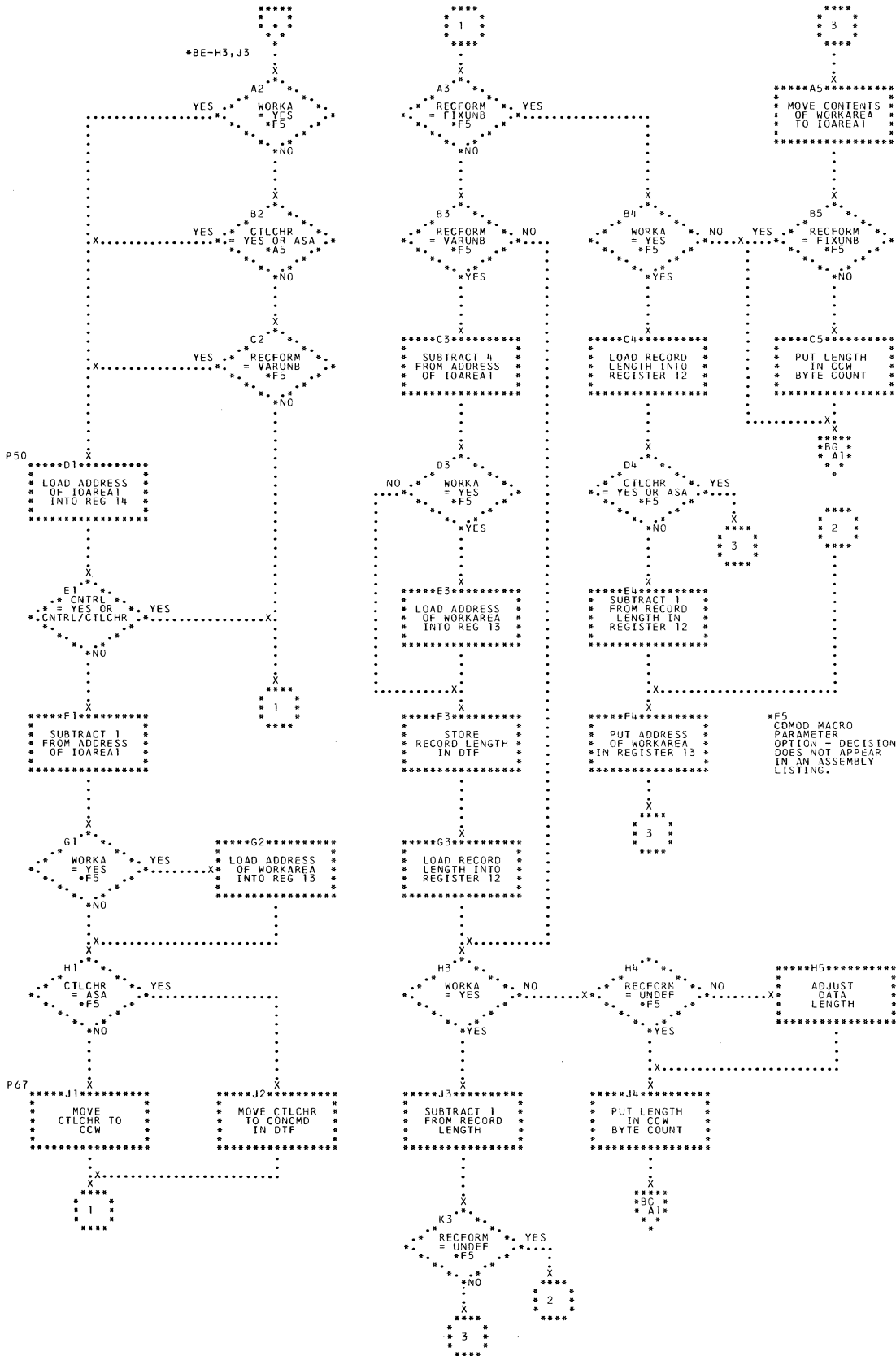


●Chart BE. CDMOD: PUT Macro (Section 1 of 4)

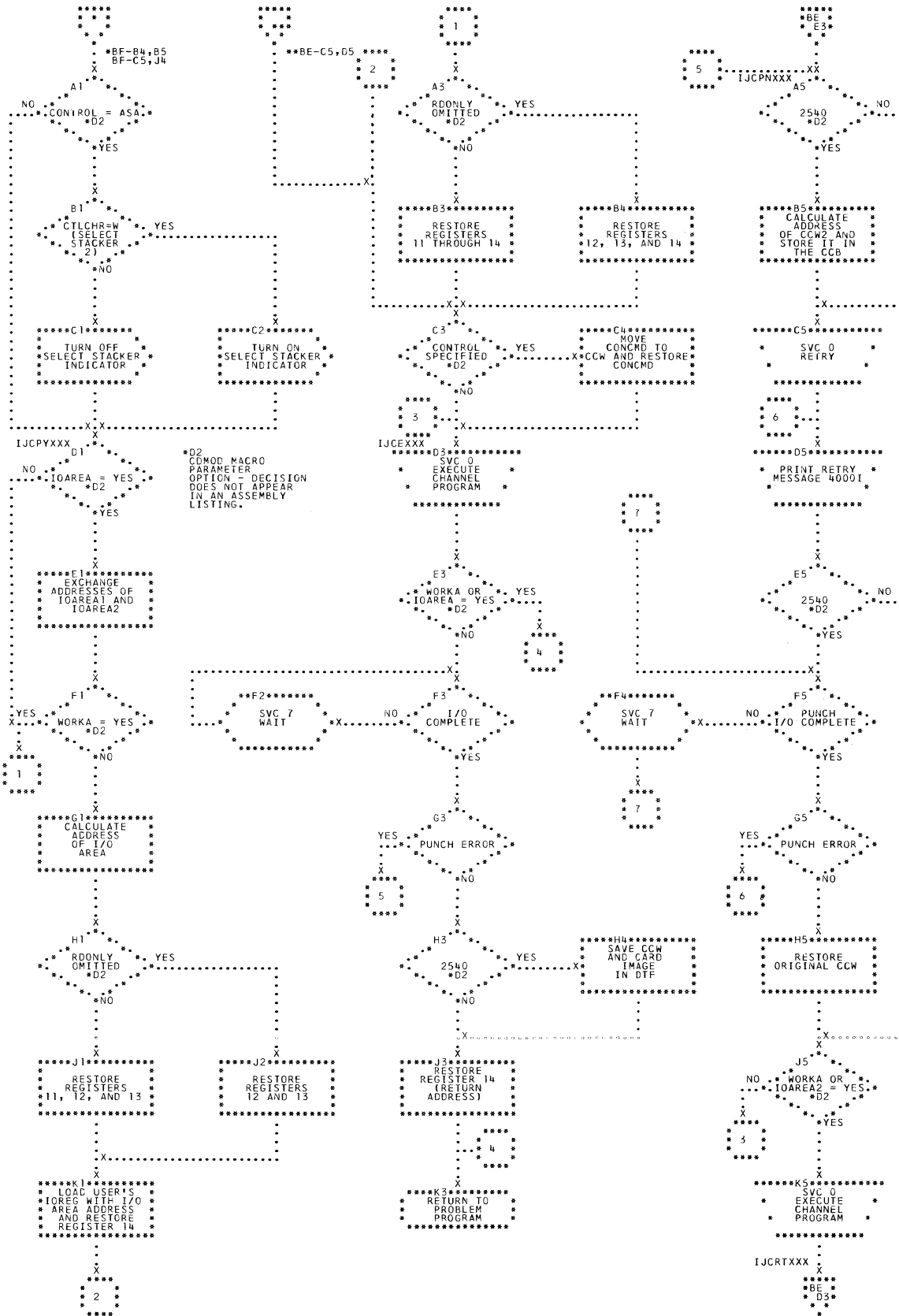


*A5 CDMOD MACRO PARAMETER OPTION - DECISION DOES NOT APPEAR IN AN ASSEMBLY LISTING.

●Chart BF. CDMOD: PUT Macro (Section 2 of 4)



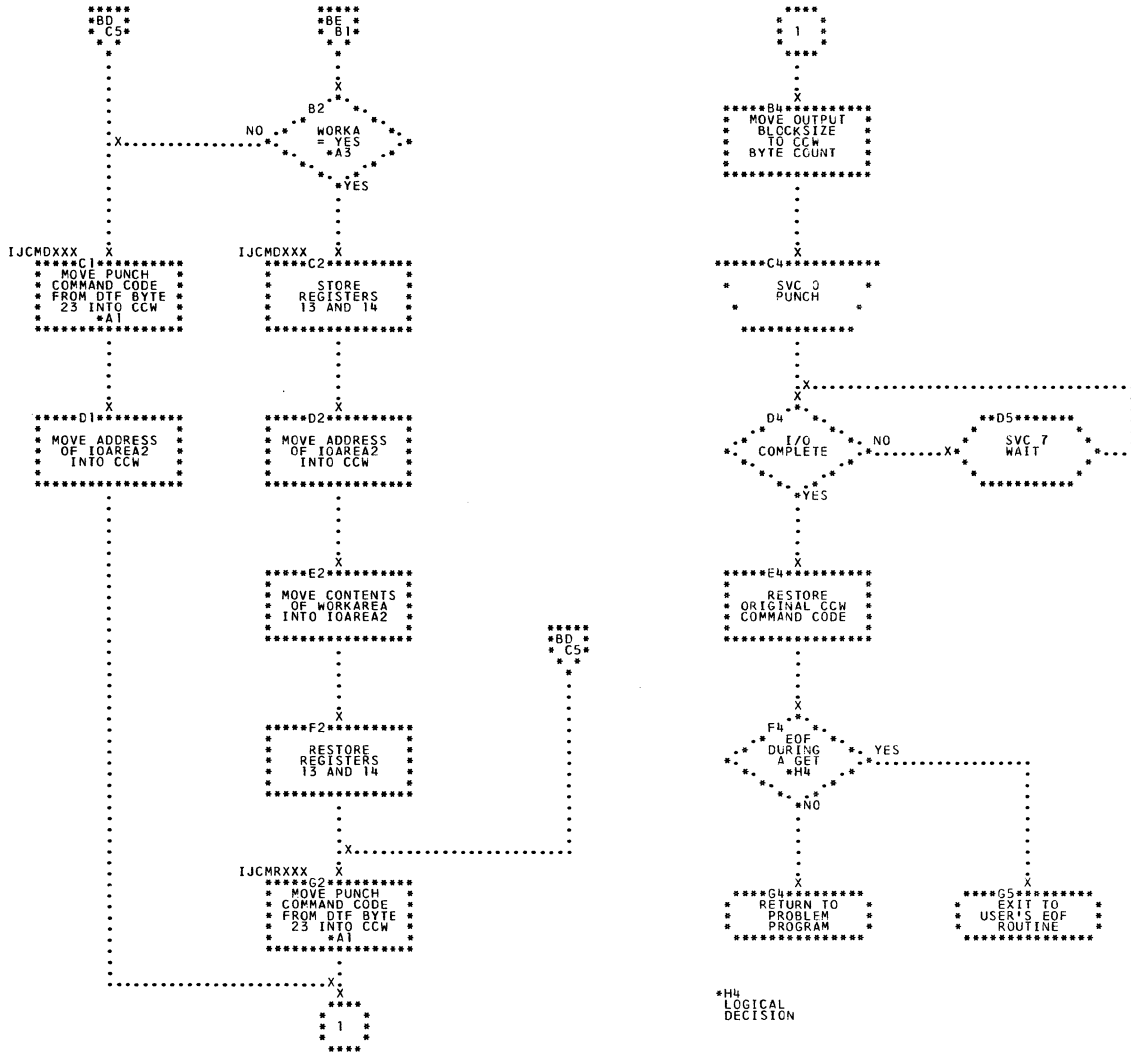
●Chart BG. CDMOD: PUT Macro (Section 3 of 4)



● Chart BH. CDMOD: PUT Macro (Section 4 of 4)

*A1
 X*01 FOR 1442
 X*09 FOR 2520,2540

*A3
 CDMOD MACRO
 PARAMETER
 OPTION - DECISION
 DOES NOT APPEAR
 IN AN ASSEMBLY
 LISTING.



*H4
 LOGICAL
 DECISION

Chart DA. MRMOD: GET, READ, and CHECK Macros (Section 1 of 3)

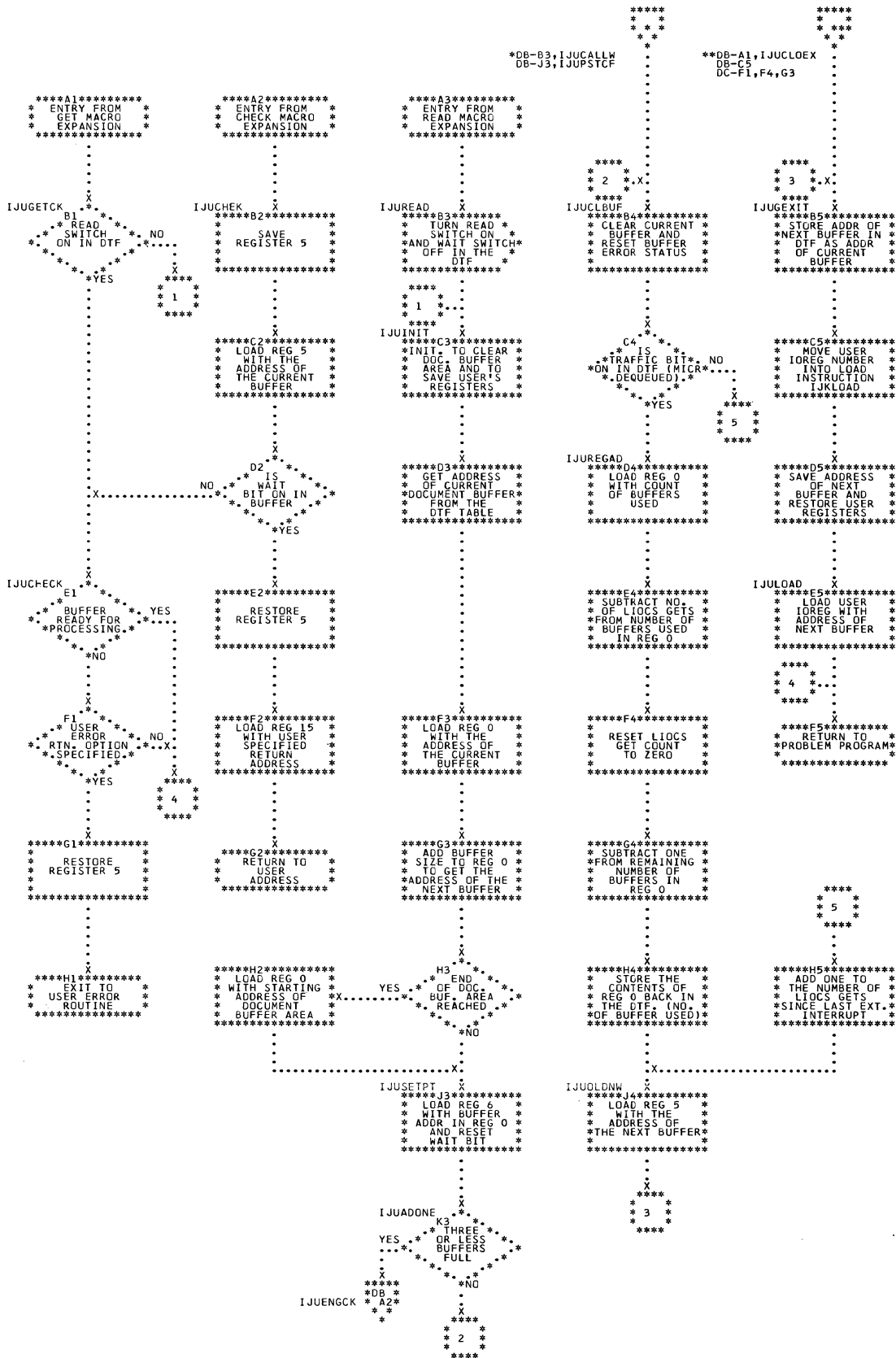


Chart DE. MRMOD: LITE Macro

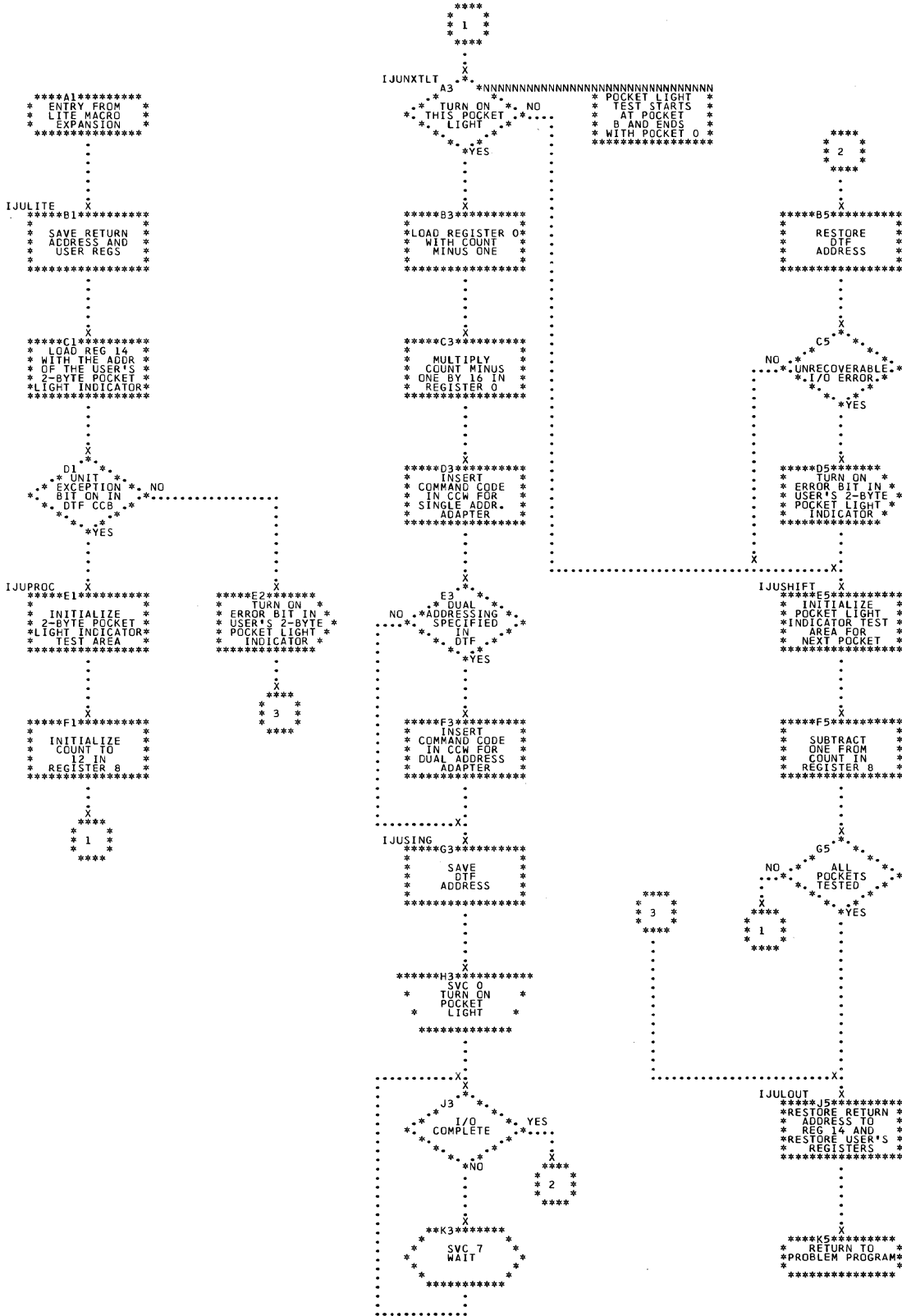


Chart DG. \$\$BOMR01: Open MICR

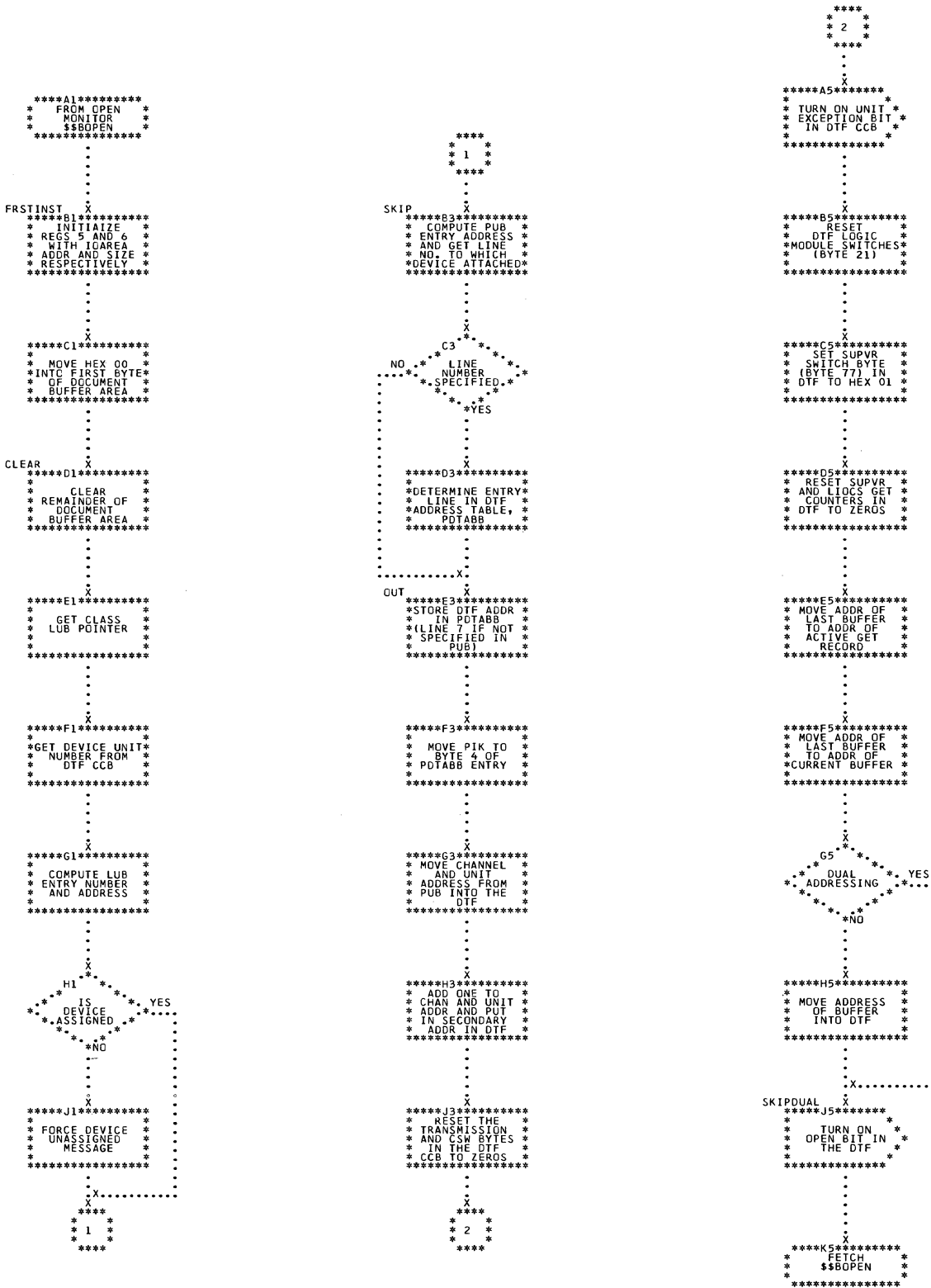


Chart DH. \$\$BCMR01: Close MICR

```

*****A2*****
* ENTRY FROM *
* CLOSE MONITOR *
* $BCLD2 *
*****
.
.
.
FRSTINST X
*****B2*****
* *
* *SET OPEN/CLOSE*
* *IND. IN THE DTF*
* *TO CLOSE (0) *
* *
*****
.
.
.
*****C2*****
* *
* *RESET *
* *TRAFFIC BIT IN *
* *THE CCB *
* *
*****
.
.
.
X-----
CHECK
.
.
.
D2
* *IS FILE*
NO * *OPERATING*
* *IN THIS*
* *PARTITION*
* *
* *YES
.
.
.
X
*****E2*****
* *TURN OFF *
* *EXTERNAL LINE *
* *STATUS IND *
* *FOR THIS ENTRY *
* *IN PDTABB *
* *
*****
.
.
.
X-----
SKIP
X
*****F2*****
* *UPDATE ADDRESS *
* *FOR NEXT *
* *ENTRY IN *
* *THE PDTABB *
* *TABLE *
* *
*****
.
.
.
X
G2
* *LAST PDTABB* NO
* *TABLE ENTRY *
* *CHECKED *
* *
* *YES
.
.
.
X
*****H2*****
* *FETCH *
* * $BCLD2 *
* *
*****

```


Chart EA. ORMOD: CNTRL Macro (Section 1 of 3)

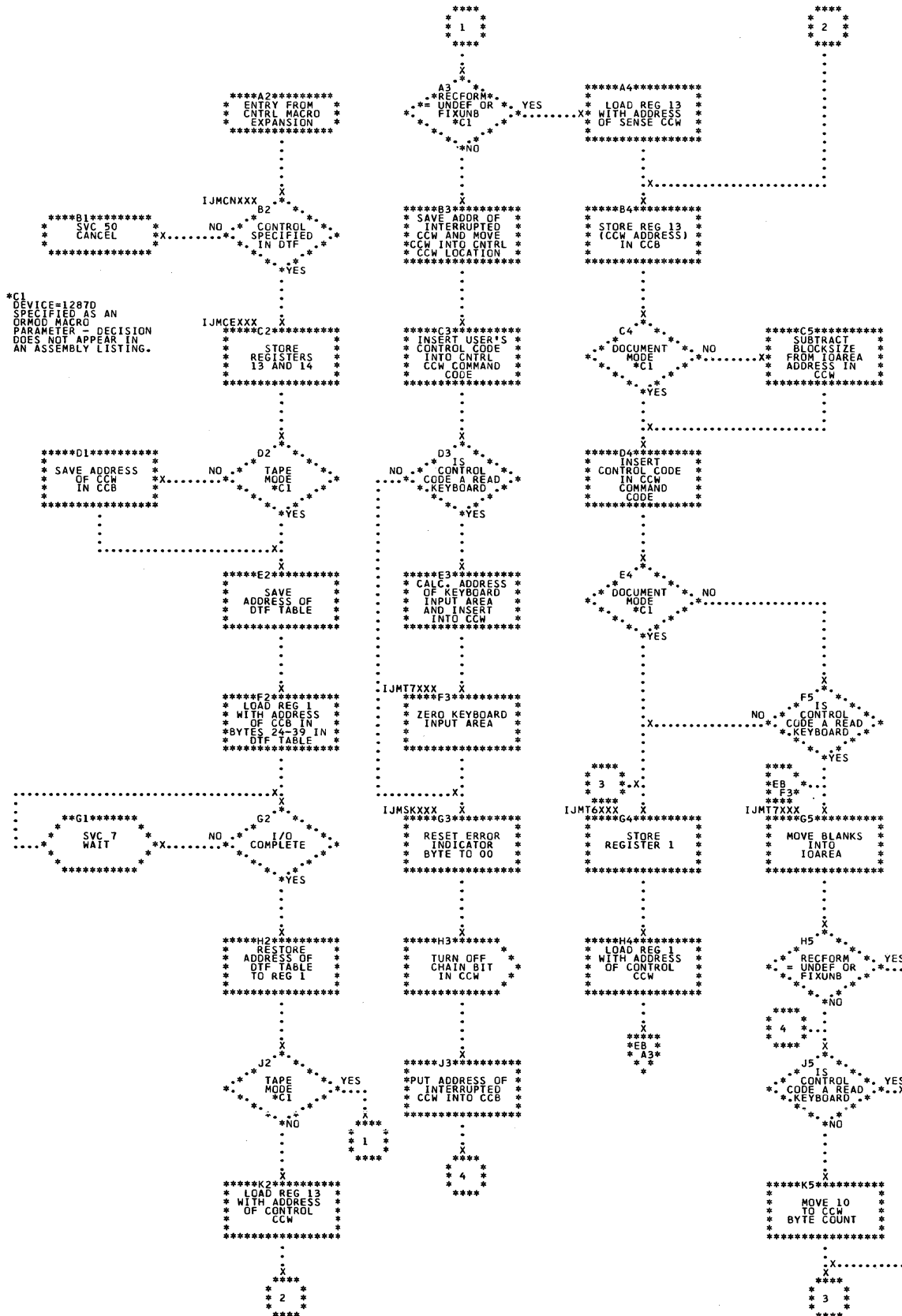


Chart EB. ORMOD: CNTRL Macro (Section 2 of 3)

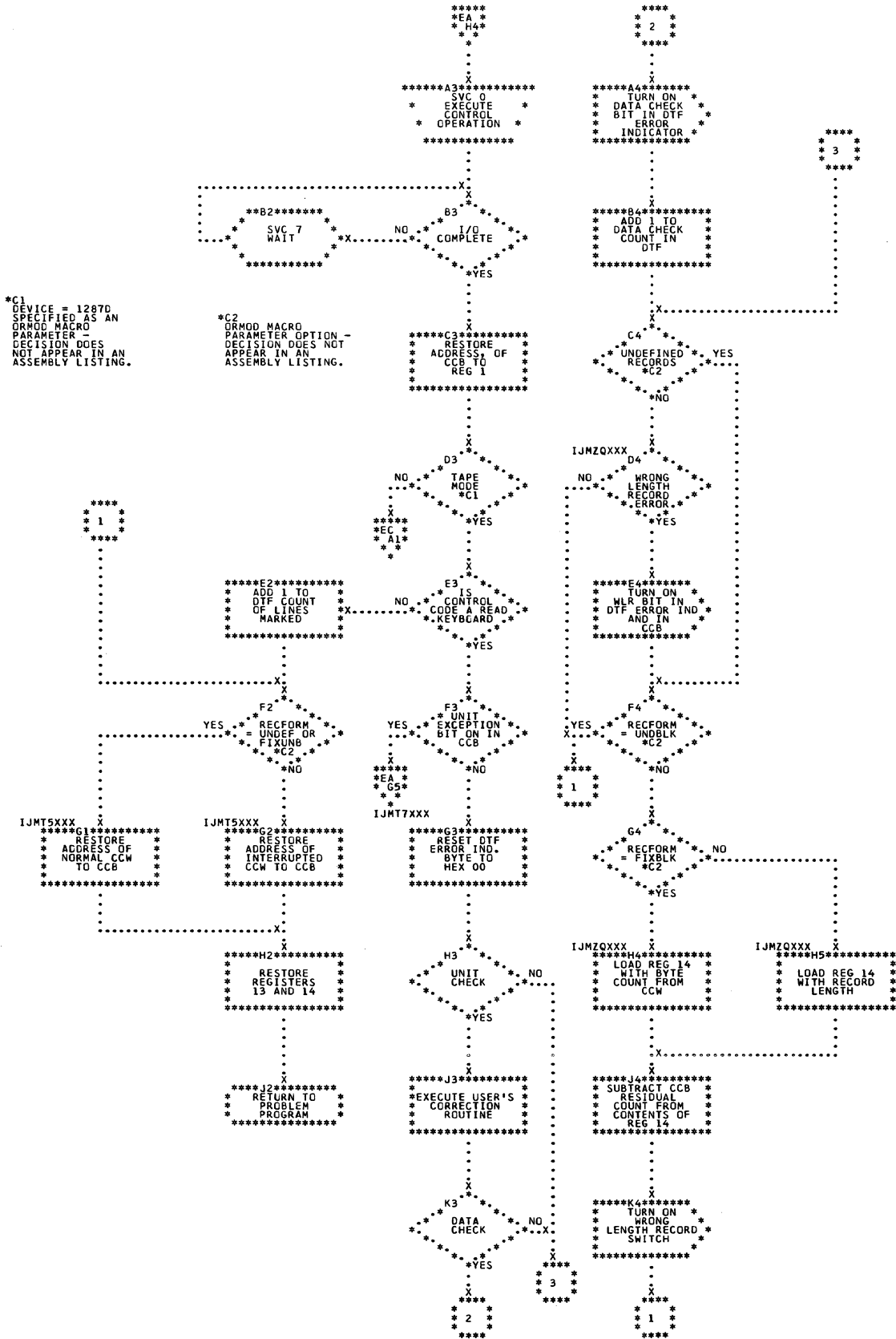


Chart EC. ORMOD: CNTRL Macro (Section 3 of 3)

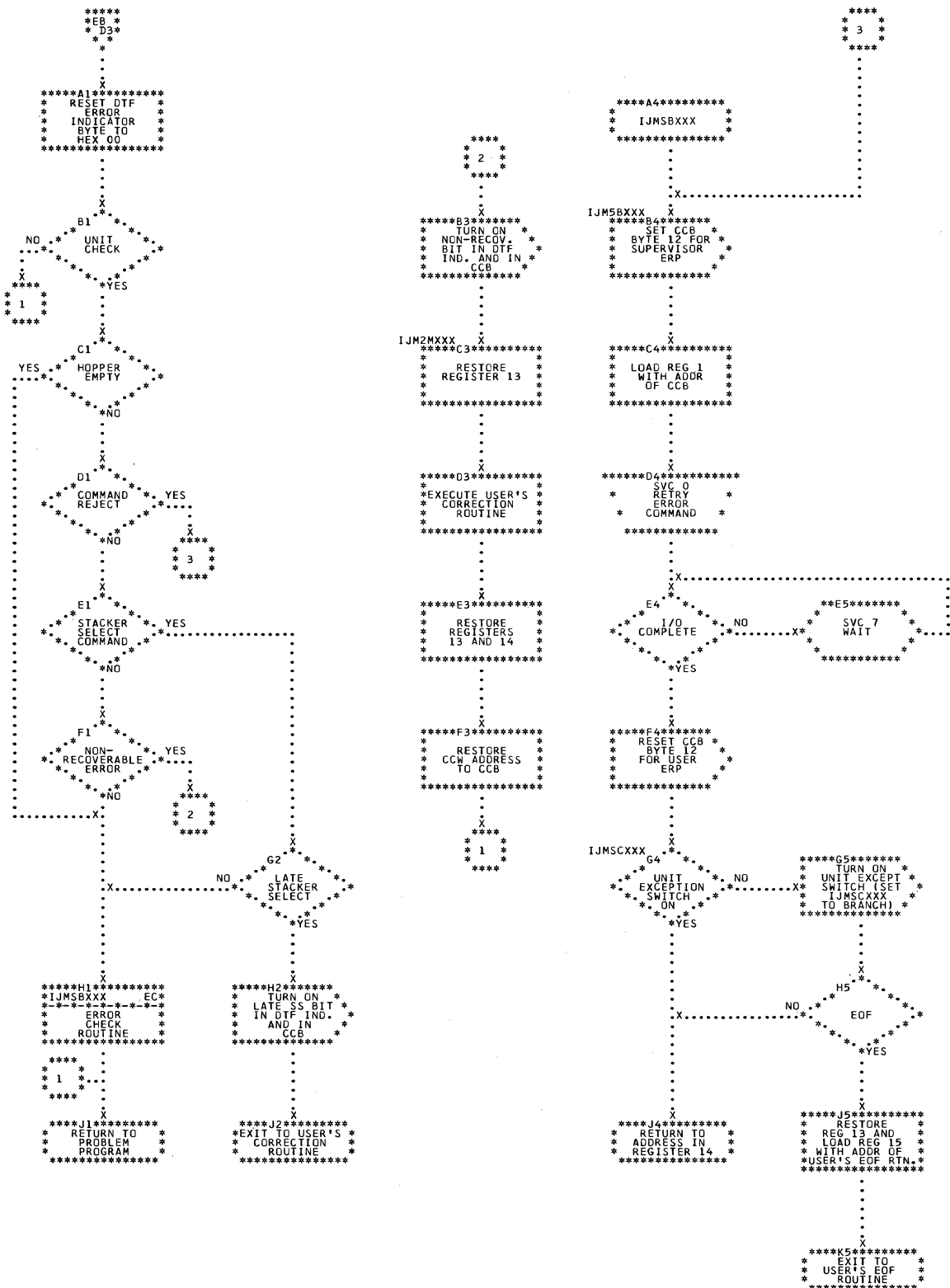


Chart ED. ORMOD: DSPLY

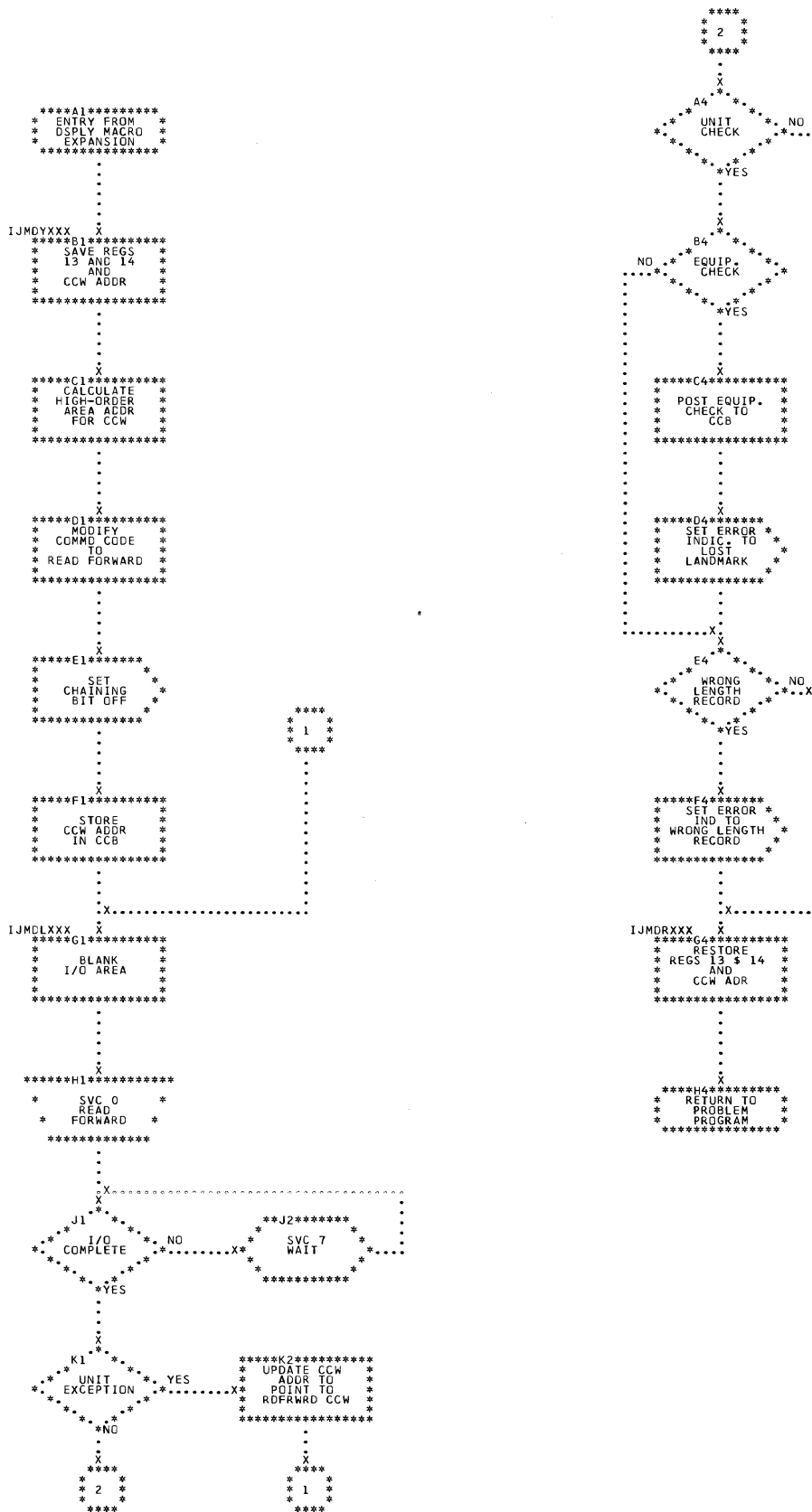


Chart EE. ORMOD: GET Macro, Unblocked Records (Section 1 of 4)

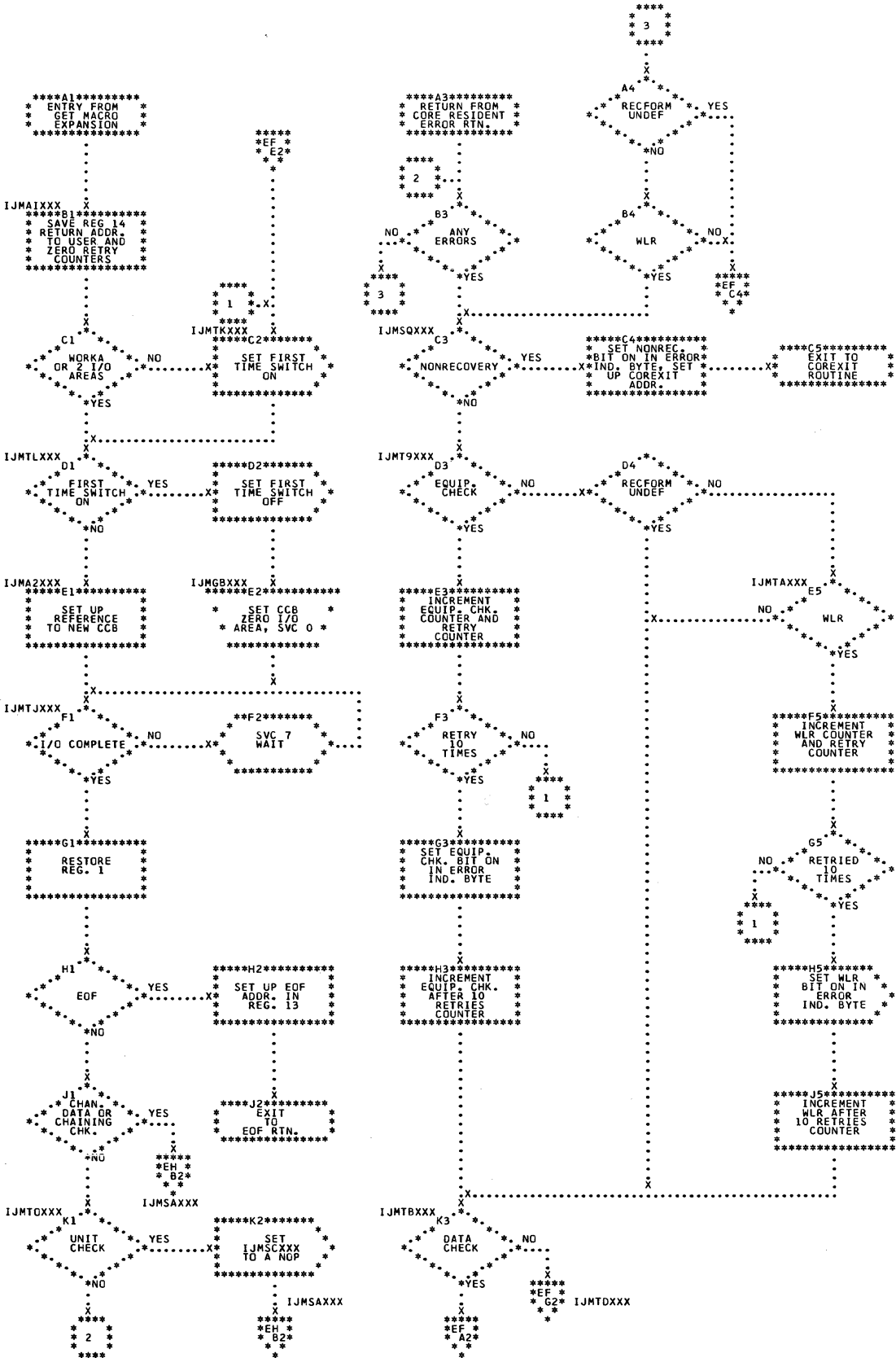


Chart EF. ORMOD: GET Macro, Unblocked Records (Section 2 of 4)

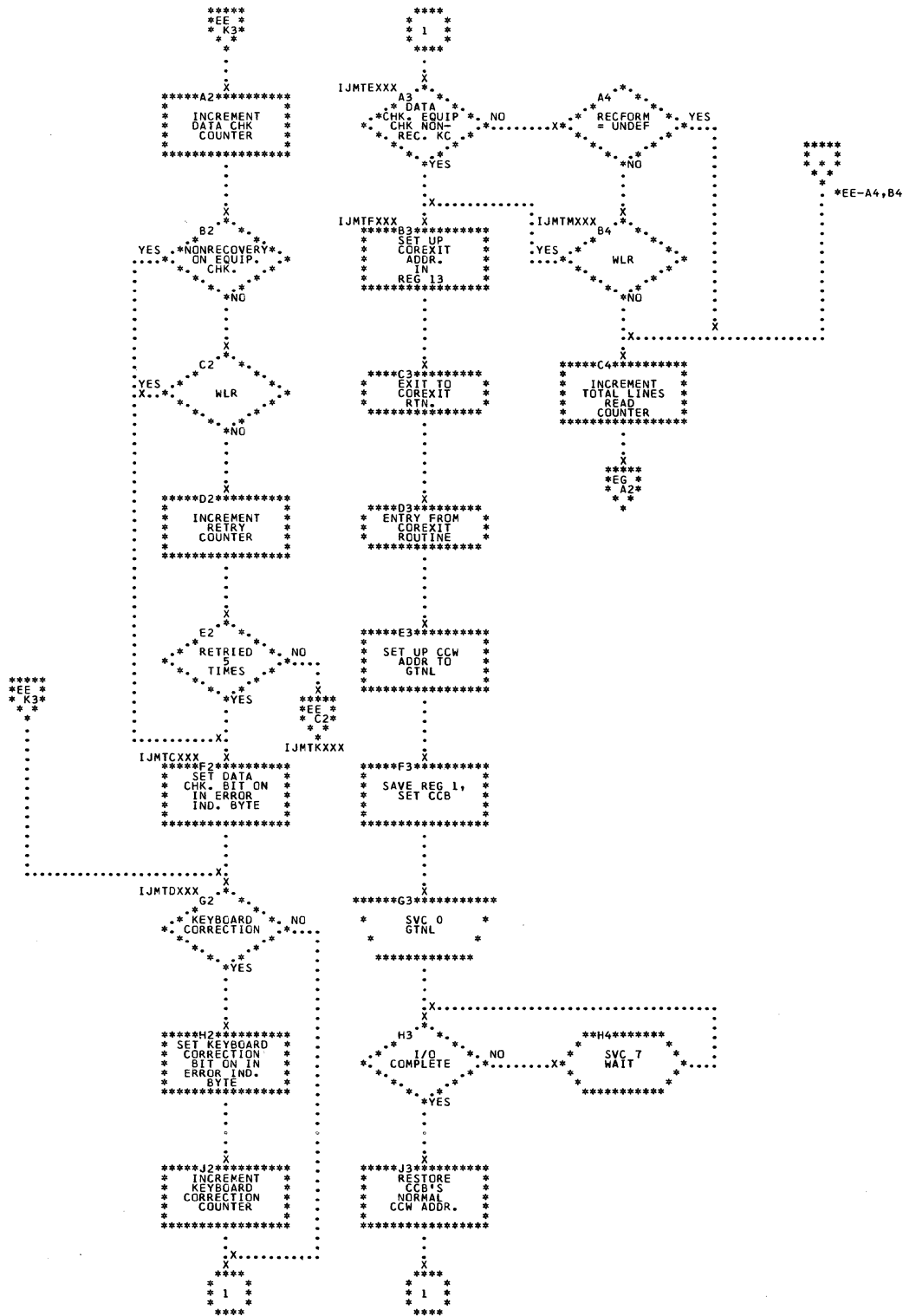


Chart EG. ORMOD: GET Macro, Unblocked Records (Section 3 of 4)

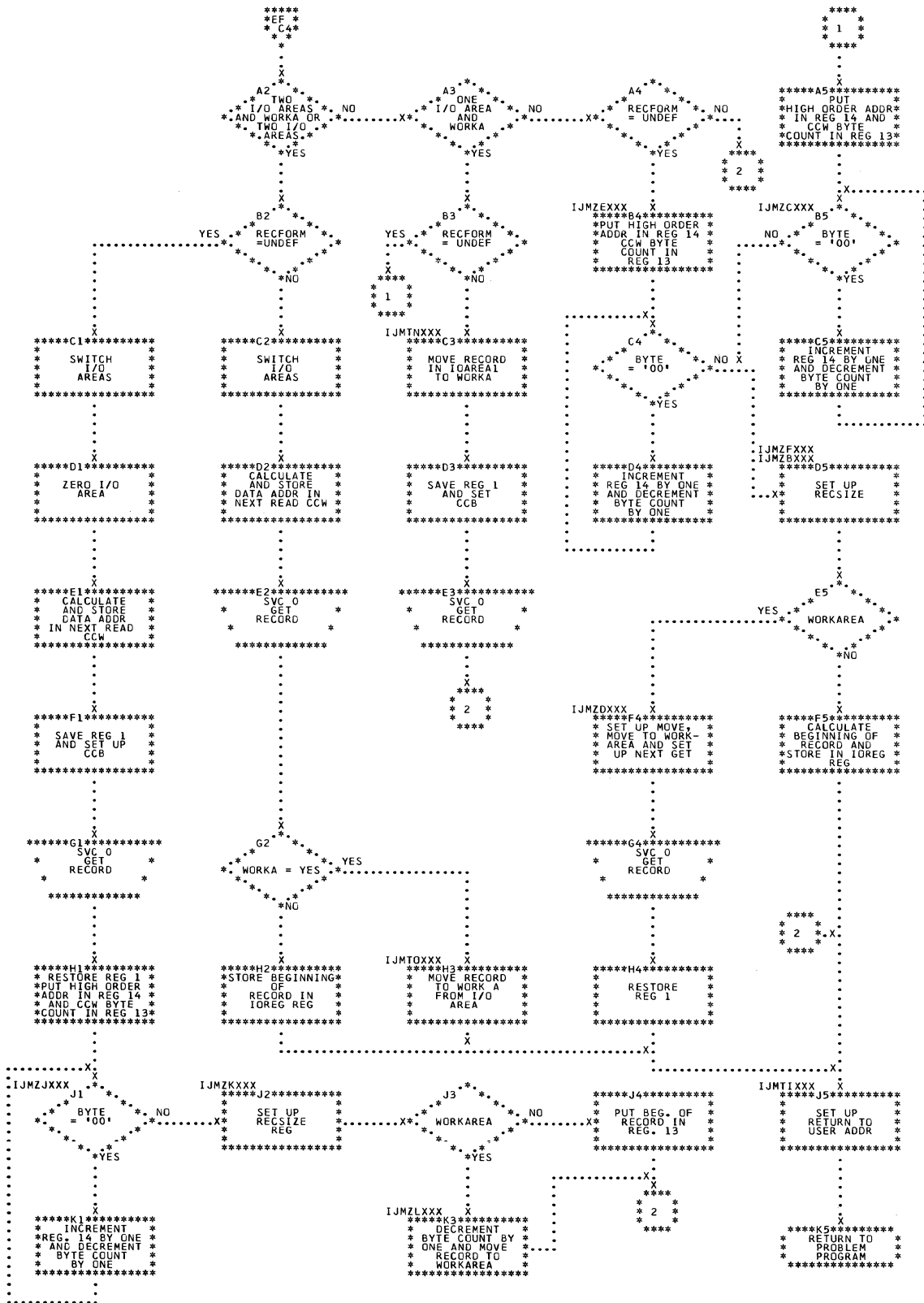


Chart EH. ORMOD: GET Macro, Unblocked Records (Section 4 of 4)

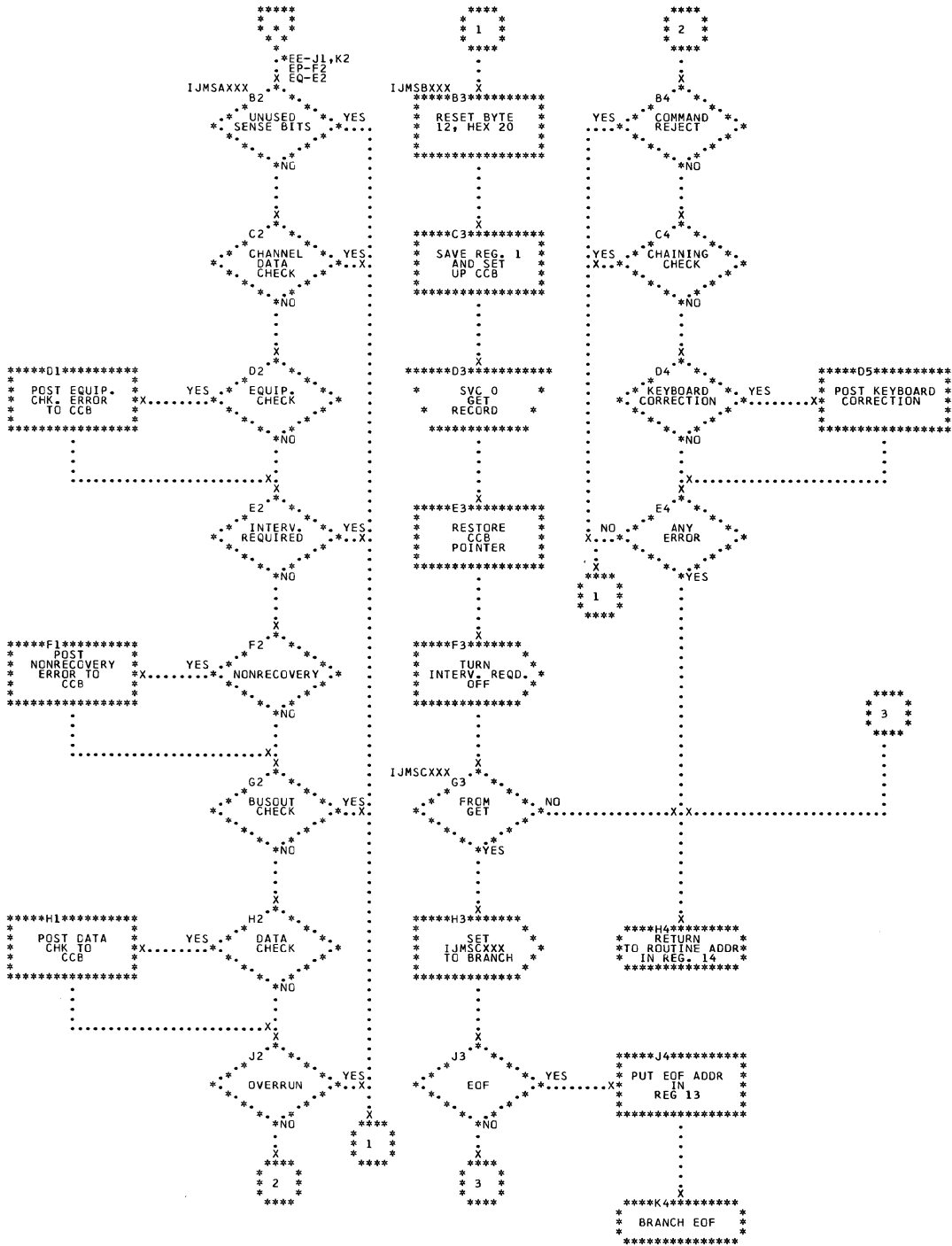


Chart EJ. ORMOD: GET Macro, Blocked Records (Section 1 of 5)

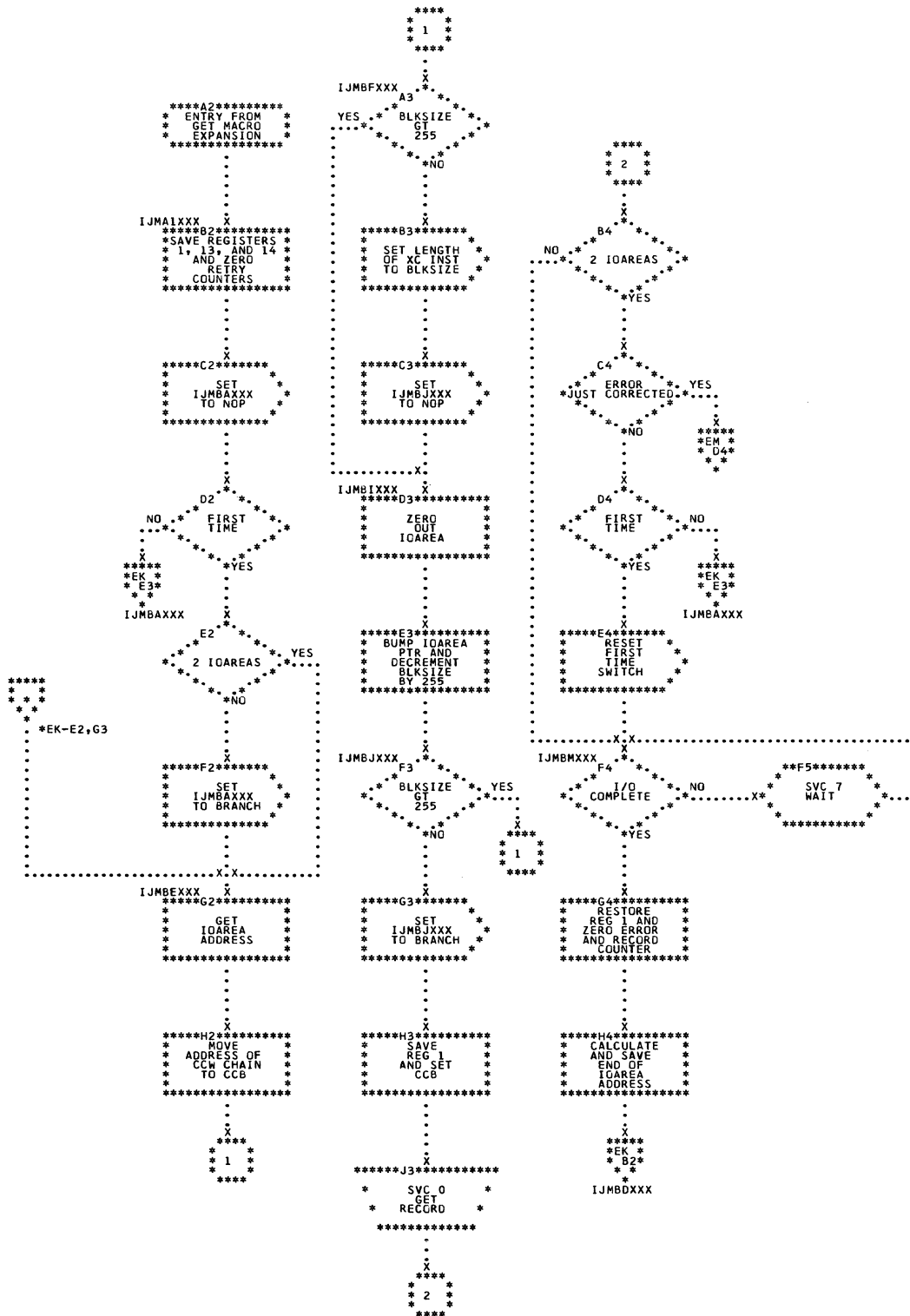


Chart EK. ORMOD: GET Macro, Blocked Records (Section 2 of 5)

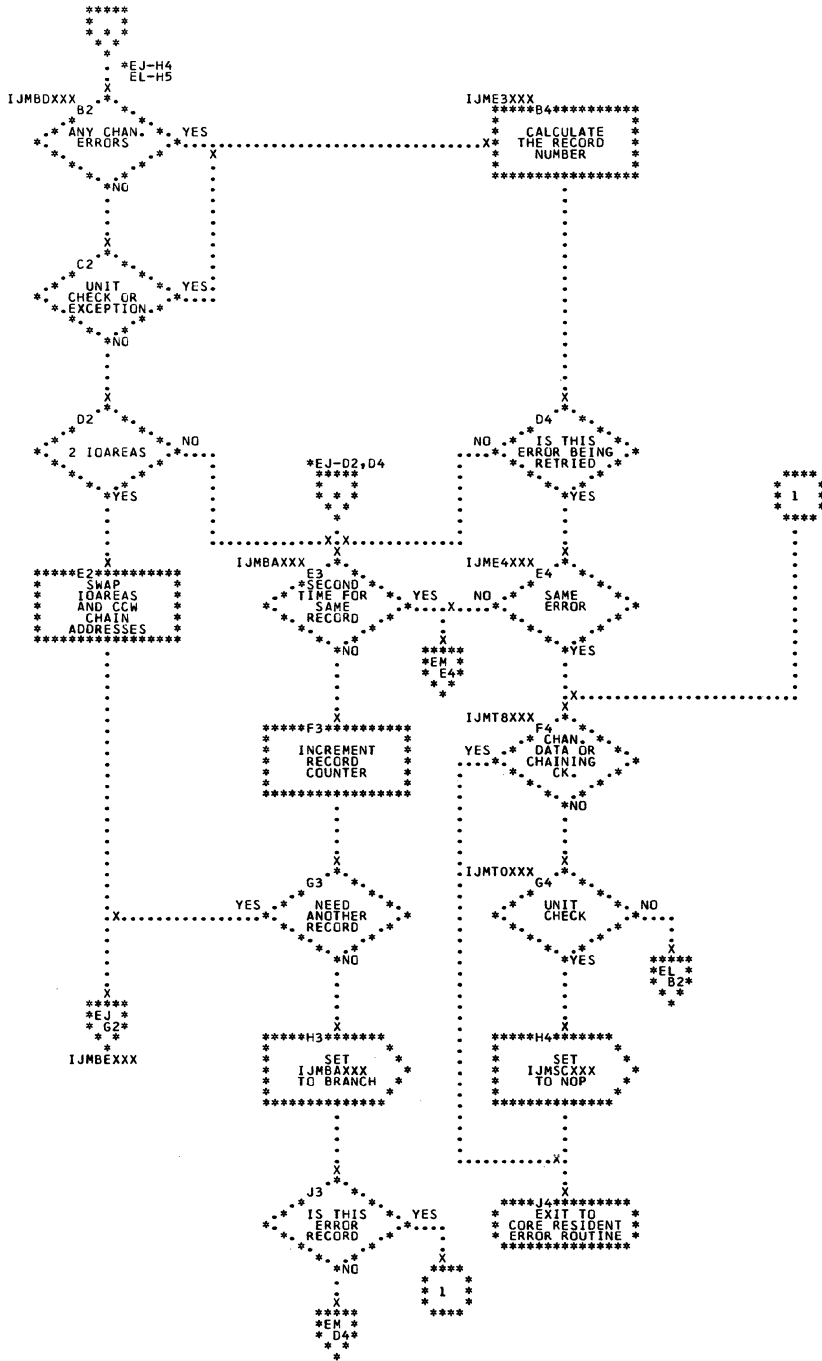


Chart EM. ORMOD: GET Macro, Blocked Records (Section 4 of 5)

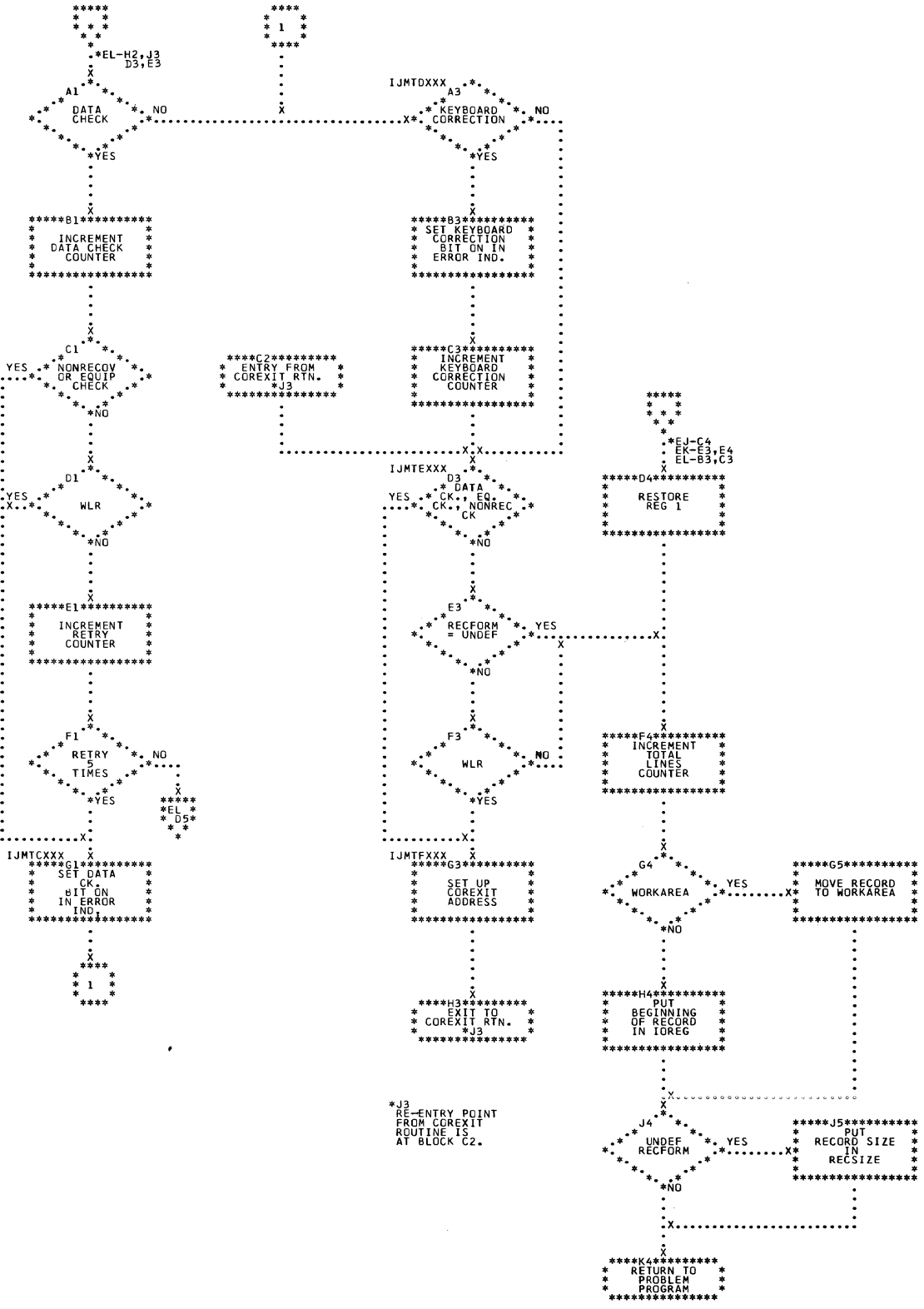


Chart EN. ORMOD: GET Macro, Blocked Records (Section 5 of 5)

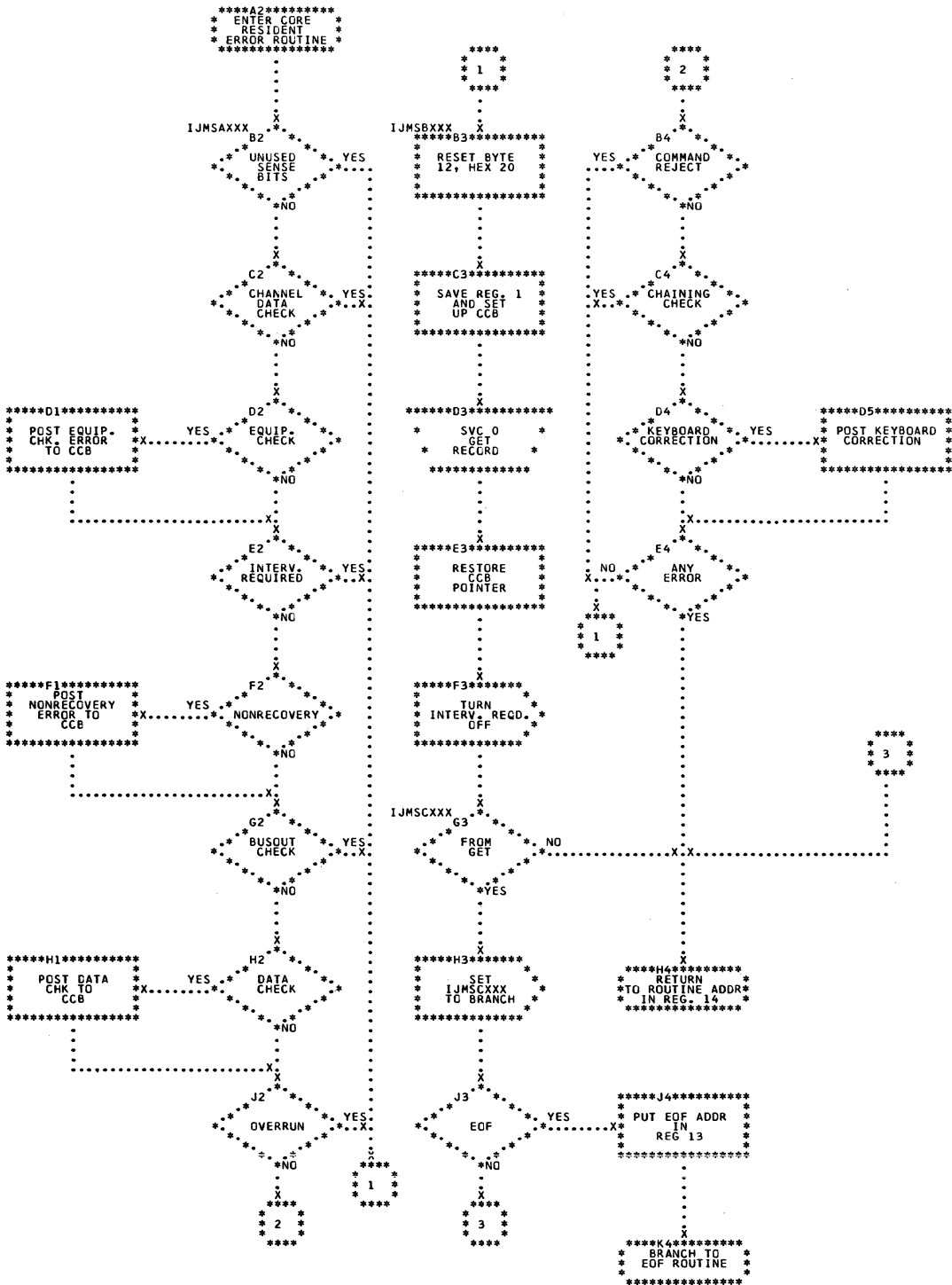


Chart EP. ORMOD: RDLNE Macro, Unblocked Records

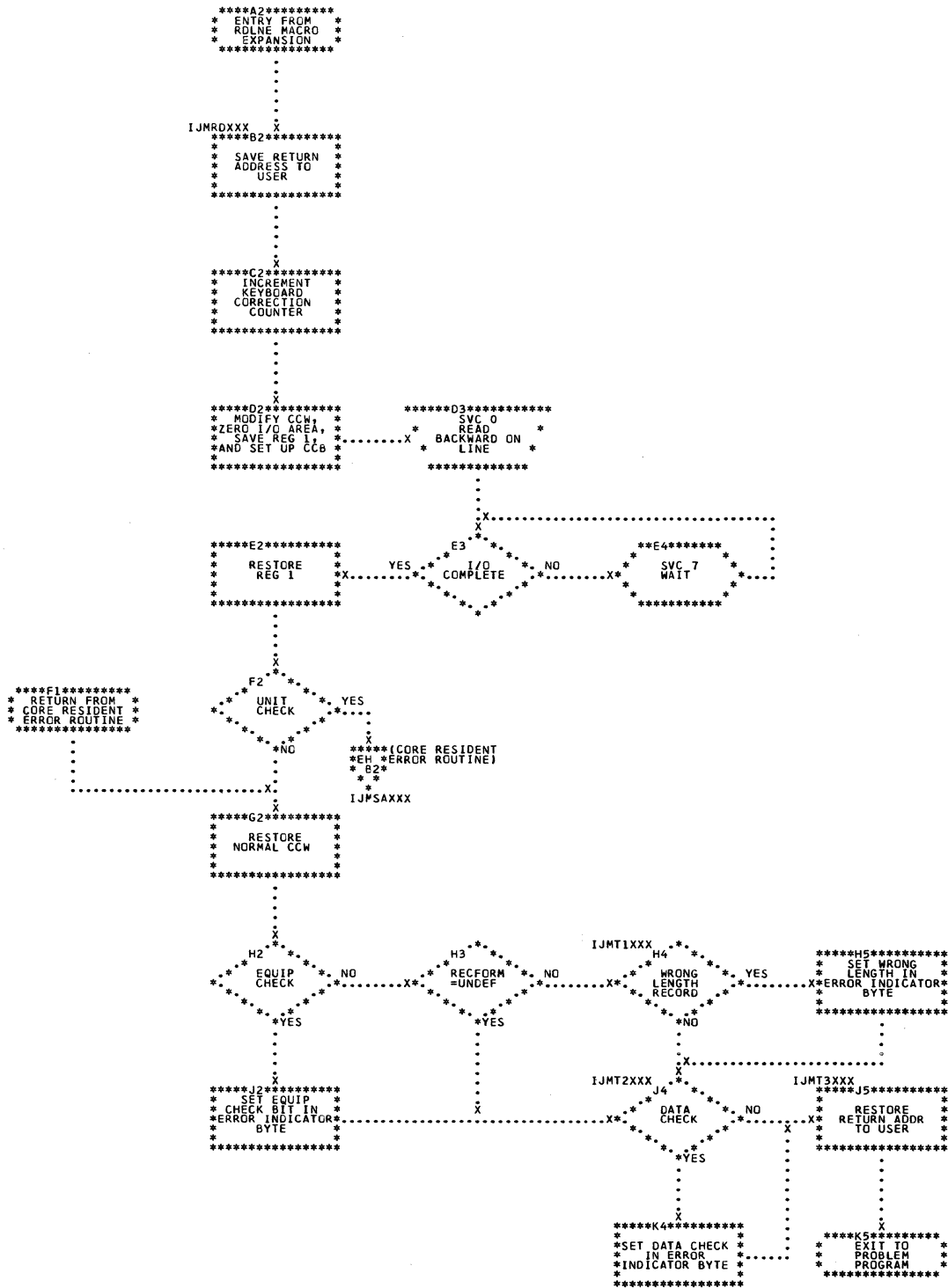


Chart EQ. ORMOD: RDLNE Macro, Blocked Records

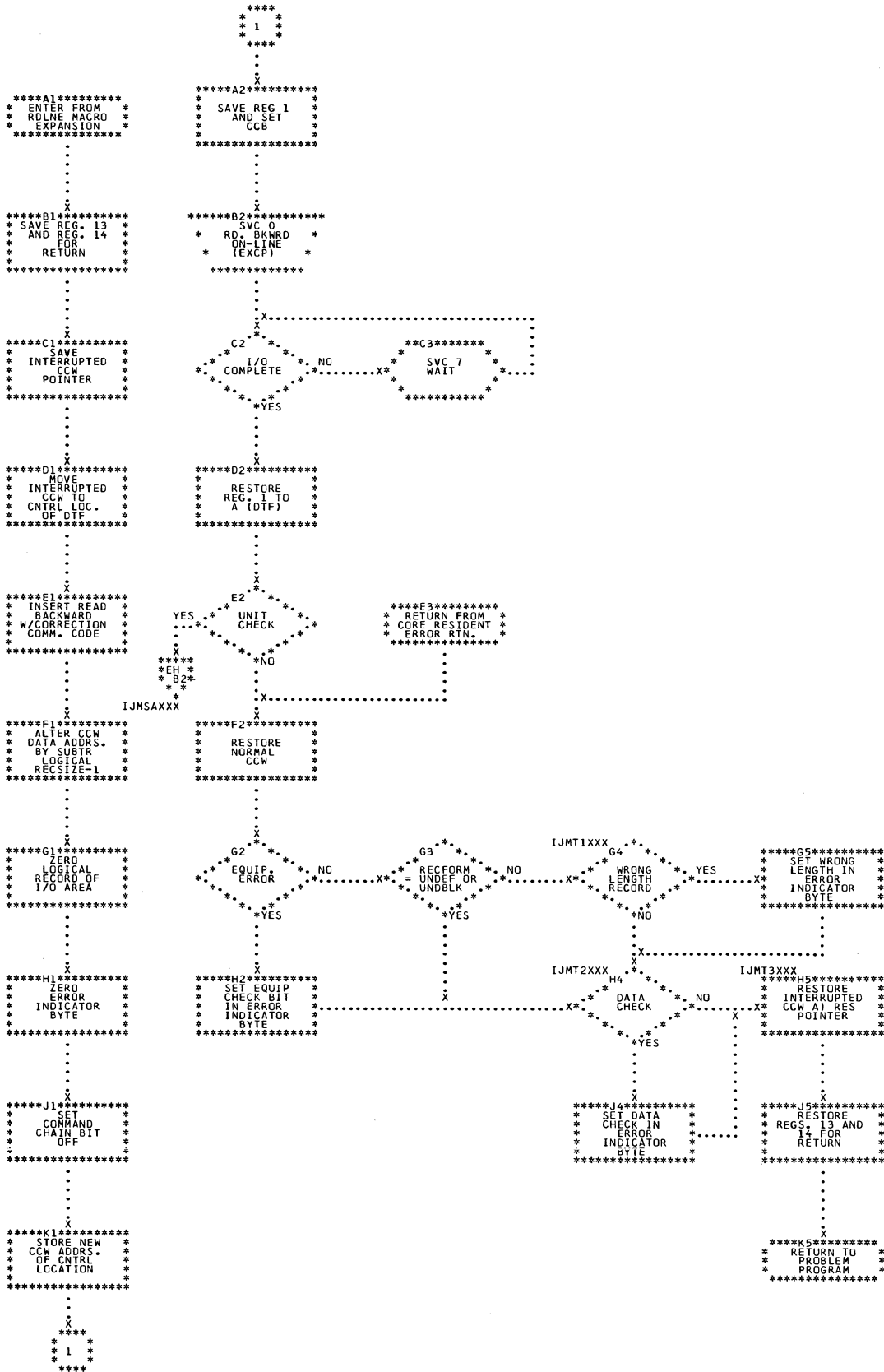


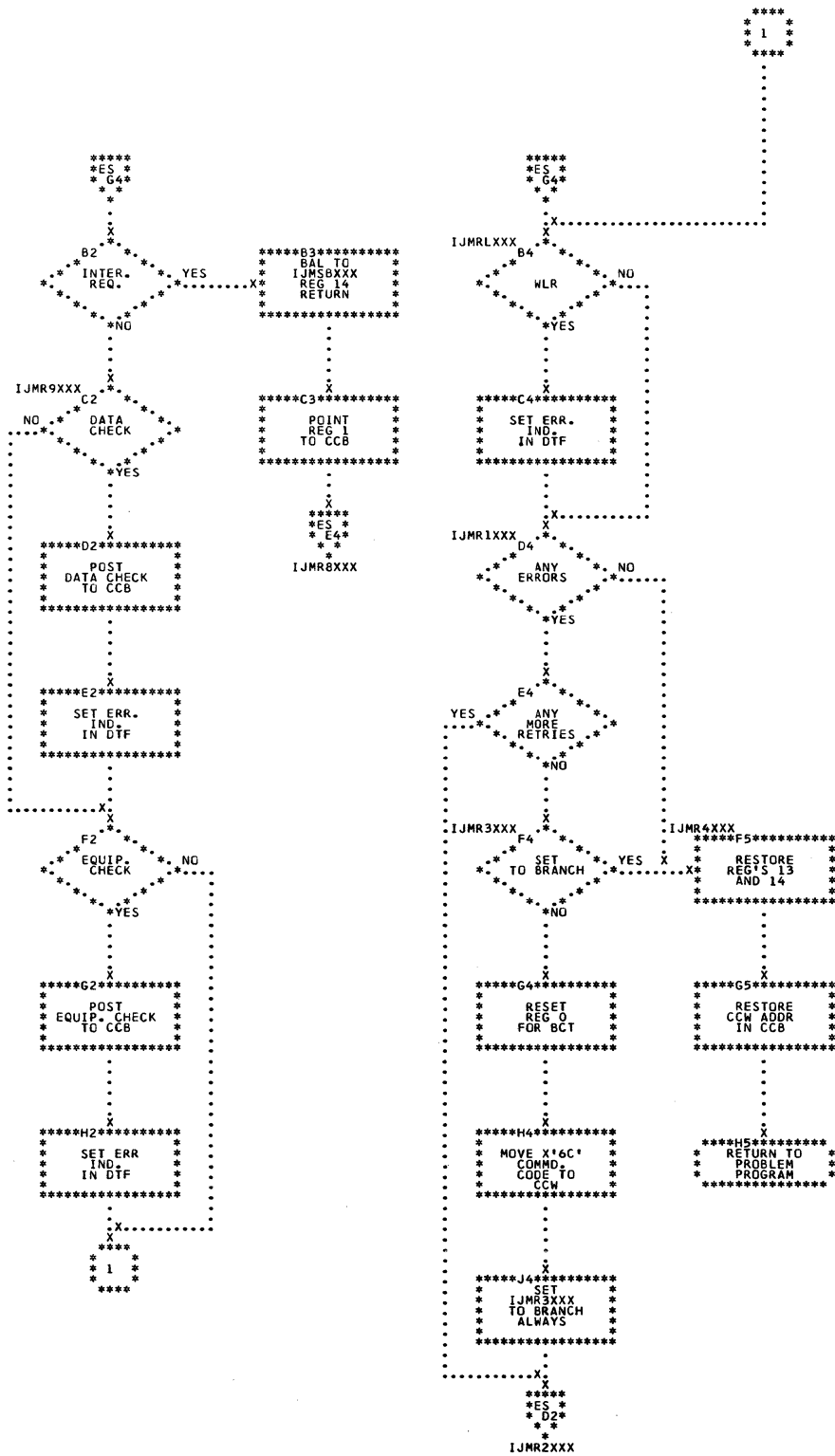
Chart ER. ORMOD: READ Macro

```

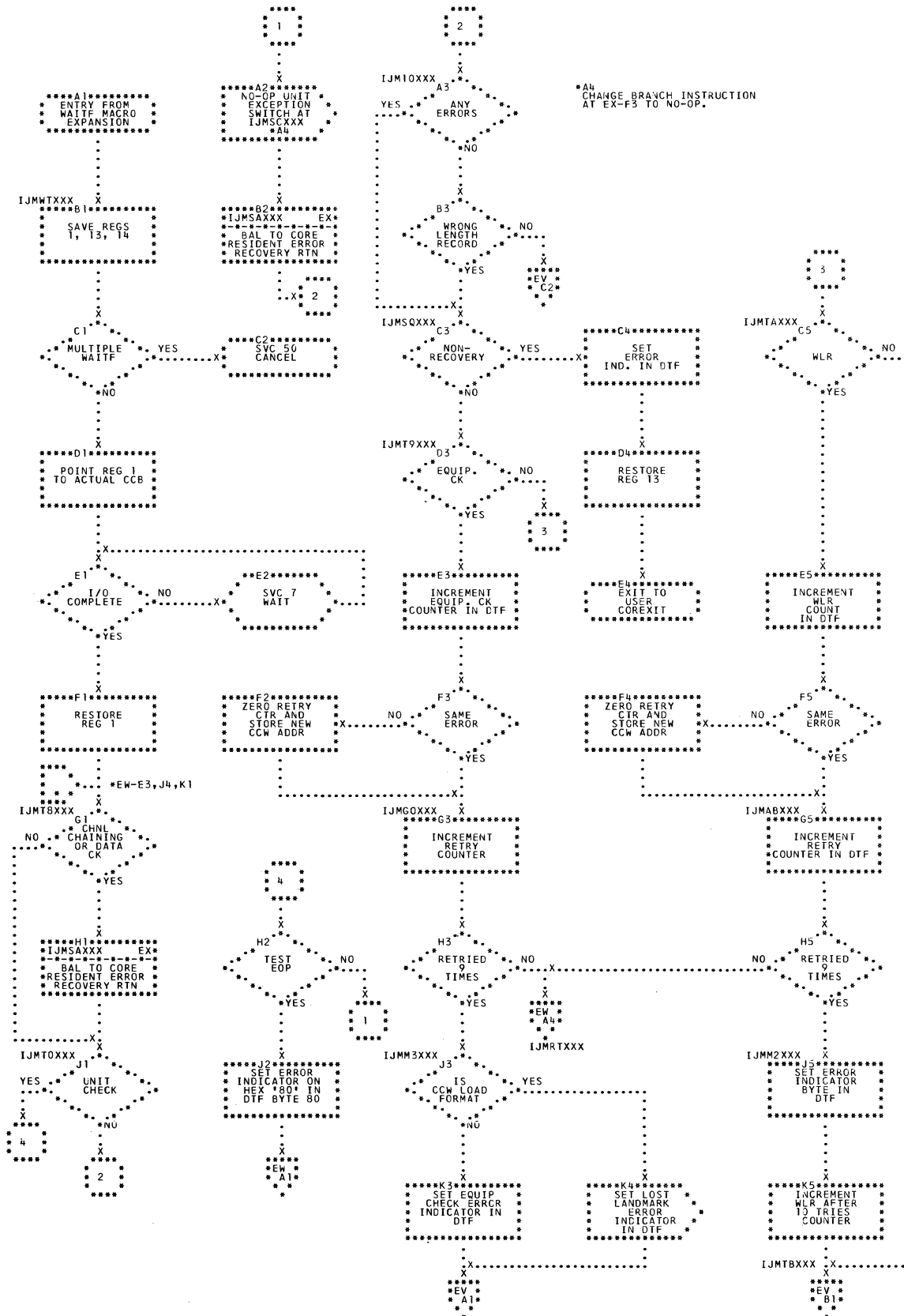
*****B3*****
* ENTRY *
* FROM READ *
* EXPANSION *
*****
.
.
.
.
.
IJM1XXX X
*****C3*****
* SAVE REGS *
* 1,13 AND *
* 14 *
*****
.
.
.
.
.
*****D3*****
* ZERO ERROR *
* INDICATORS *
* AND RETRY *
* COUNTERS *
*****
.
.
.
.
.
*****E3*****
* PUT CCB *
* ADDR IN *
* CCB IN DTF *
* TABLE *
*****
.
.
.
.
.
*****F3*****
* SET UP *
* REFERENCE *
* TO NEW *
* CCB *
*****
.
.
.
.
.
*****G3*****
* SVC 0 *
* EXCP *
*****
.
.
.
.
.
*****H3*****
* RESTORE *
* REGS 1, *
* 13 AND 14 *
*****
.
.
.
.
.
*****J3*****
* RETURN TO *
* PROBLEM *
* PROGRAM *
*****

```

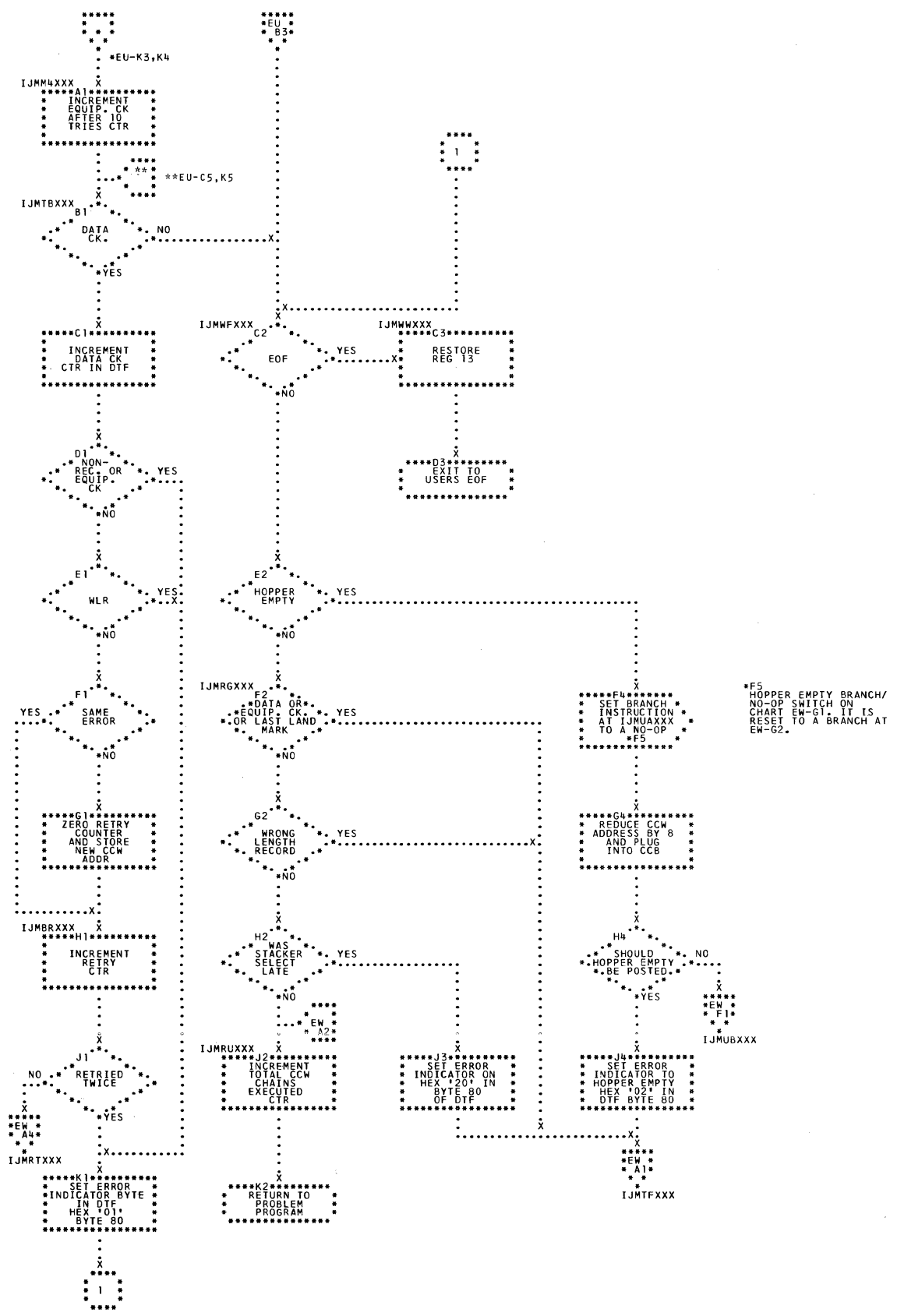

Chart ET. ORMOD: RESCN Macro (Section 2 of 2)



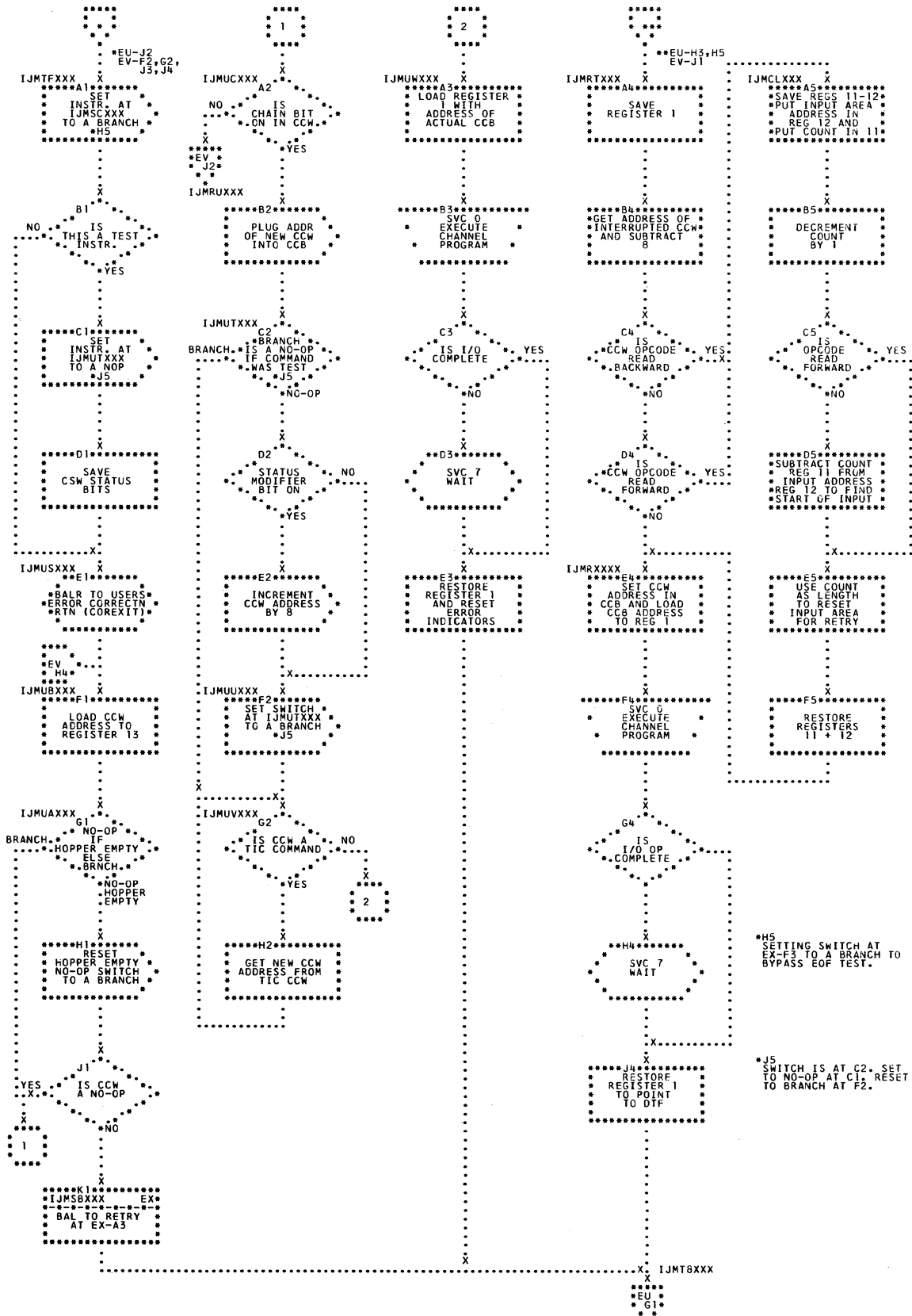
● Chart EU. ORMOD: WAITF Macro (Section 1 of 4)



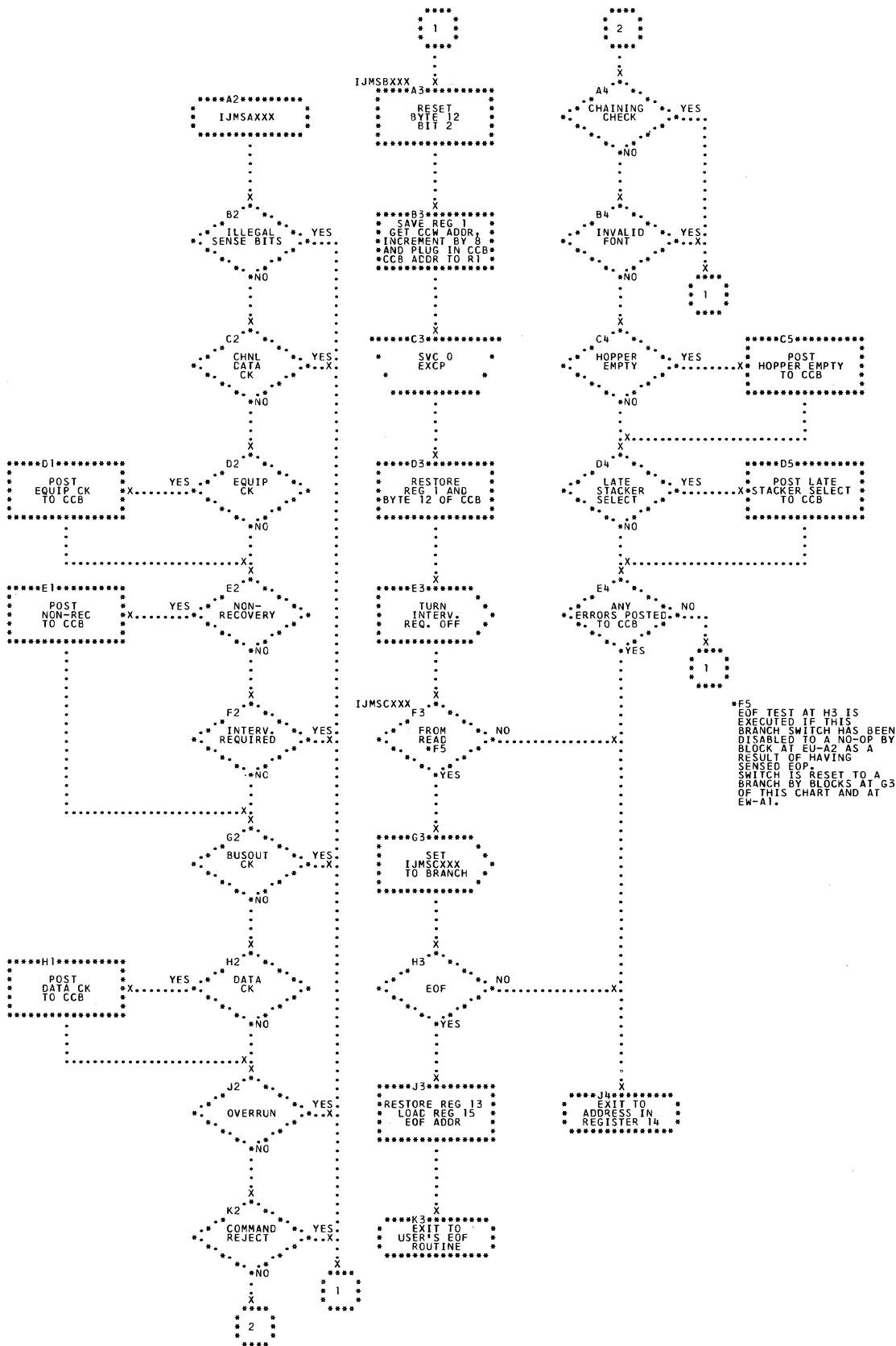
● Chart EV. ORMOD: WAITF Macro (Section 2 of 4)



● Chart EW. ORMOD: WAITF Macro (Section 3 of 4)



● Chart EX. ORMOD: WAITF Macro (Section 4 of 4)



● Chart EY. \$\$\$BOOR01: Open Optical Reader

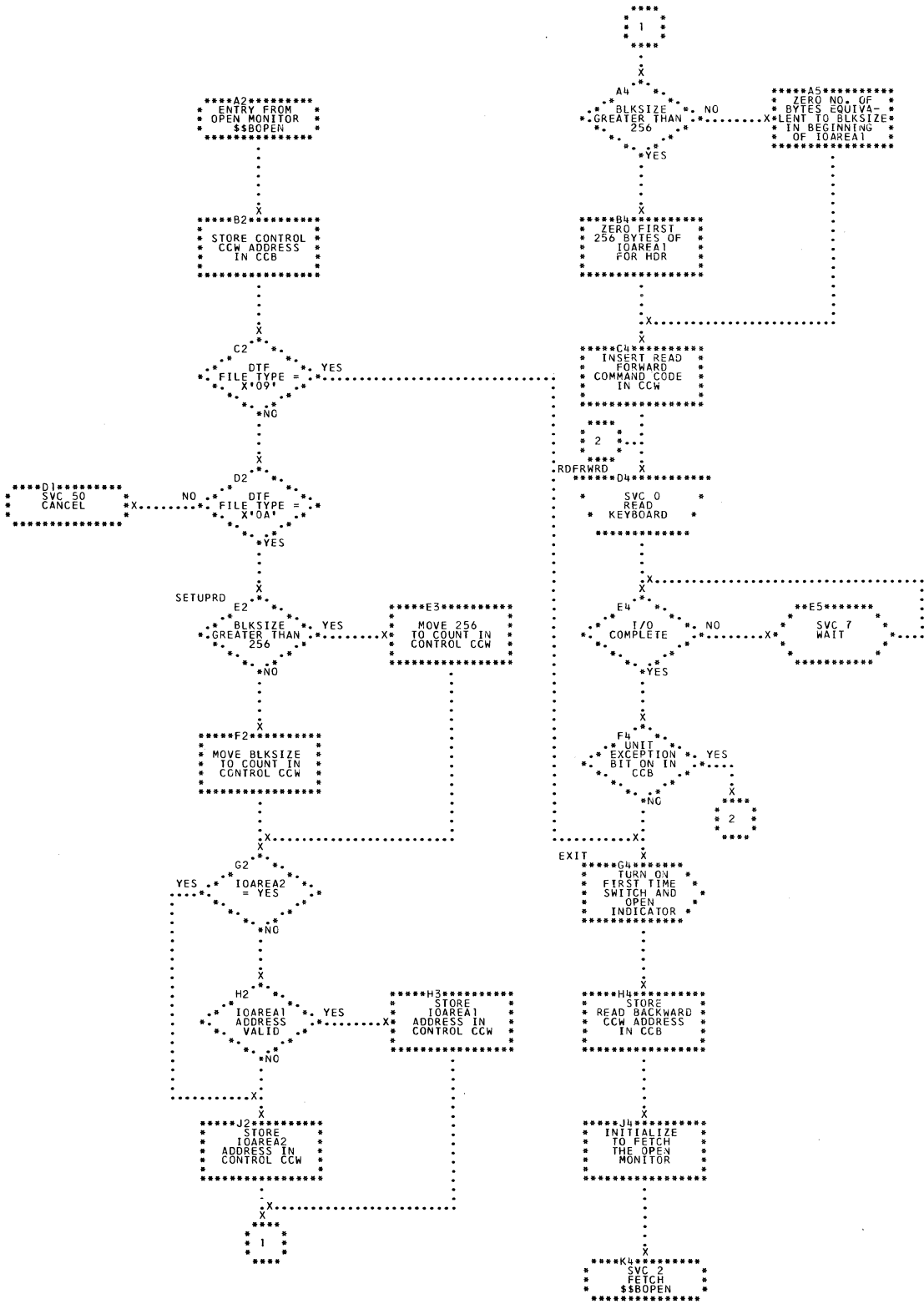
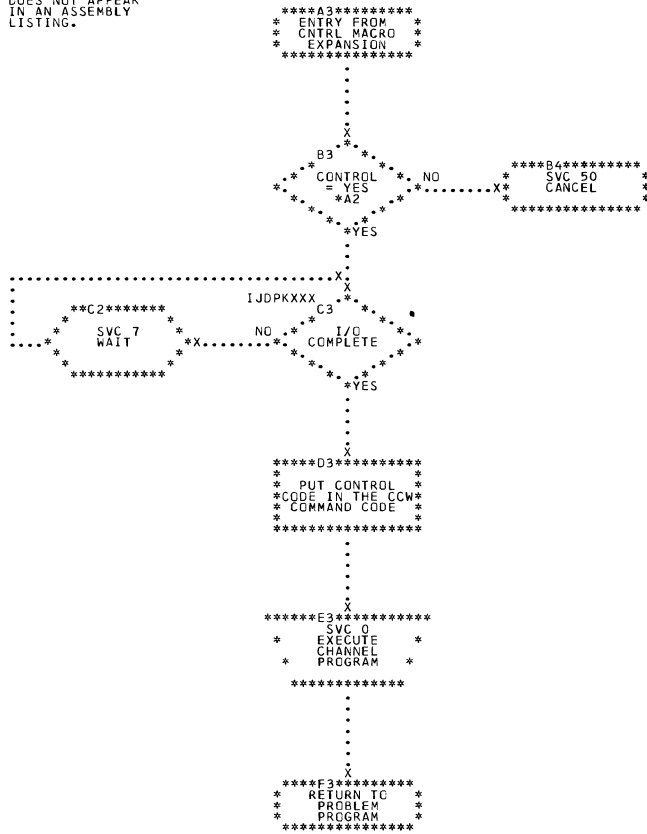
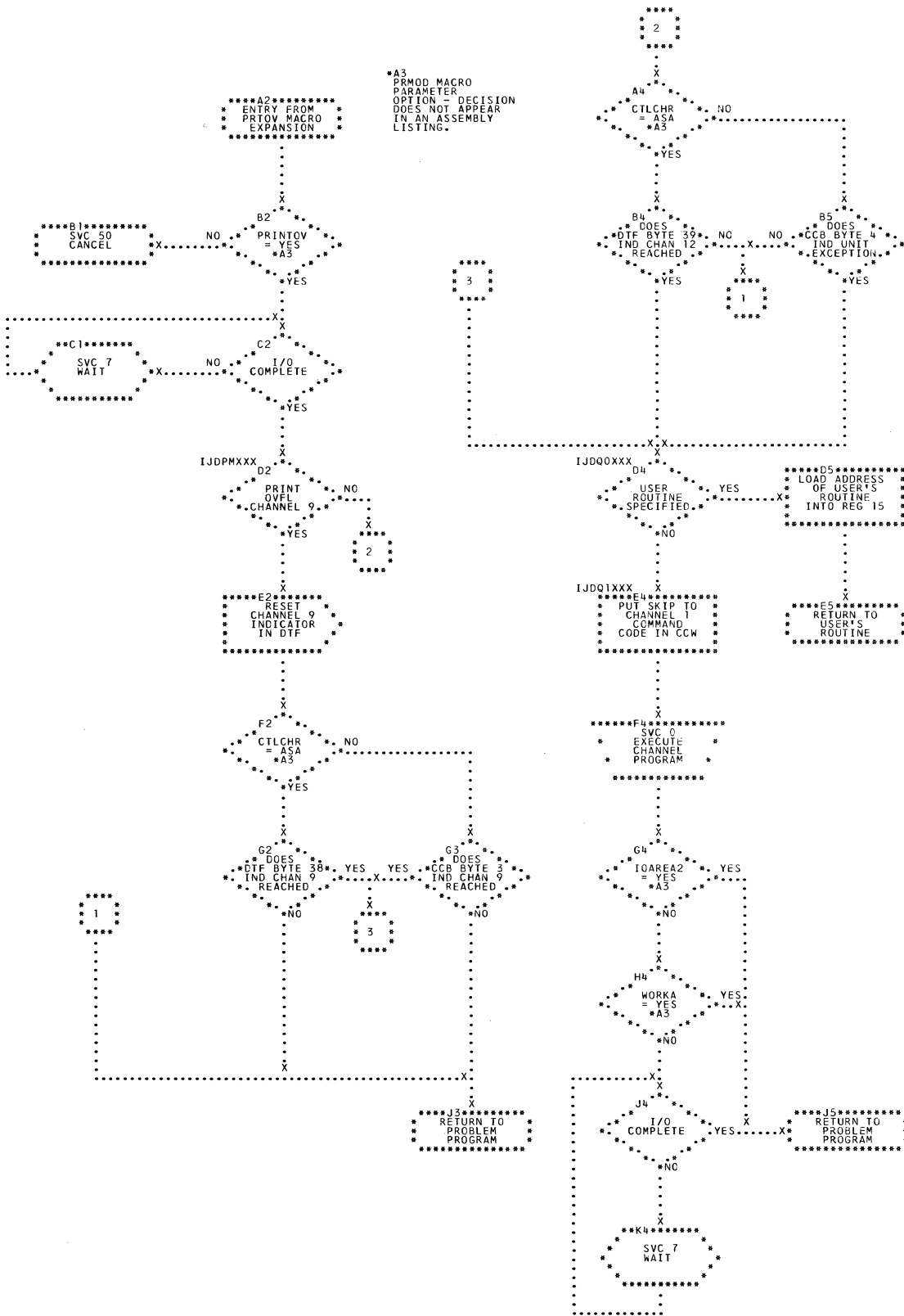


Chart FA. PRMOD: CNTRL Macro

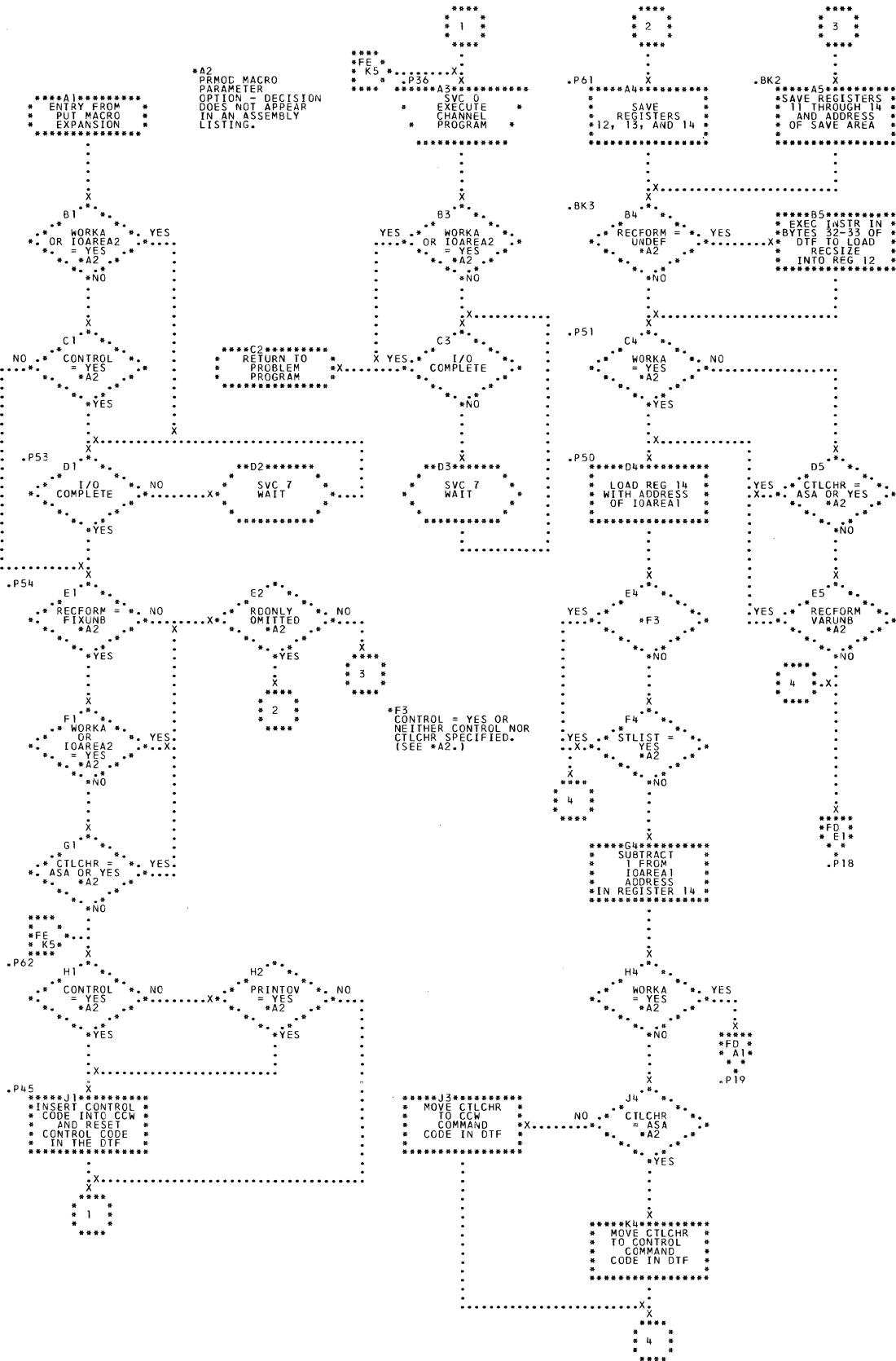
*A2
 PRMOD MACRO
 PARAMETER
 OPTION - DECISION
 DOES NOT APPEAR
 IN AN ASSEMBLY
 LISTING.



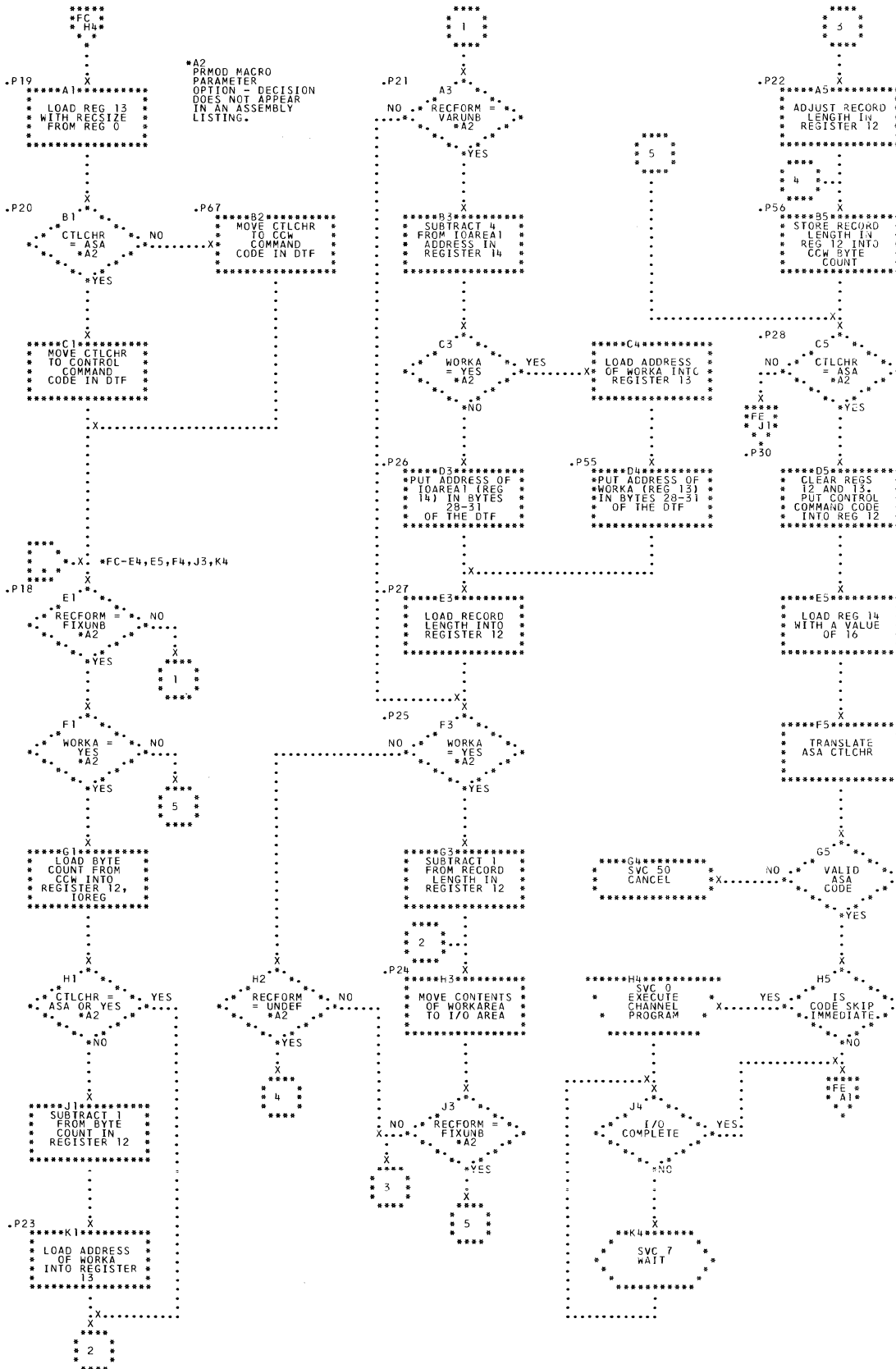
● Chart FB. PRMOD: PRTOV Macro



● Chart FC. PRMOD: PUT Macro (Section 1 of 3)



● Chart FD. PRMOD: PUT Macro (Section 2 of 3)



● Chart FE. PRMOD: PUT Macro (Section 3 of 3)

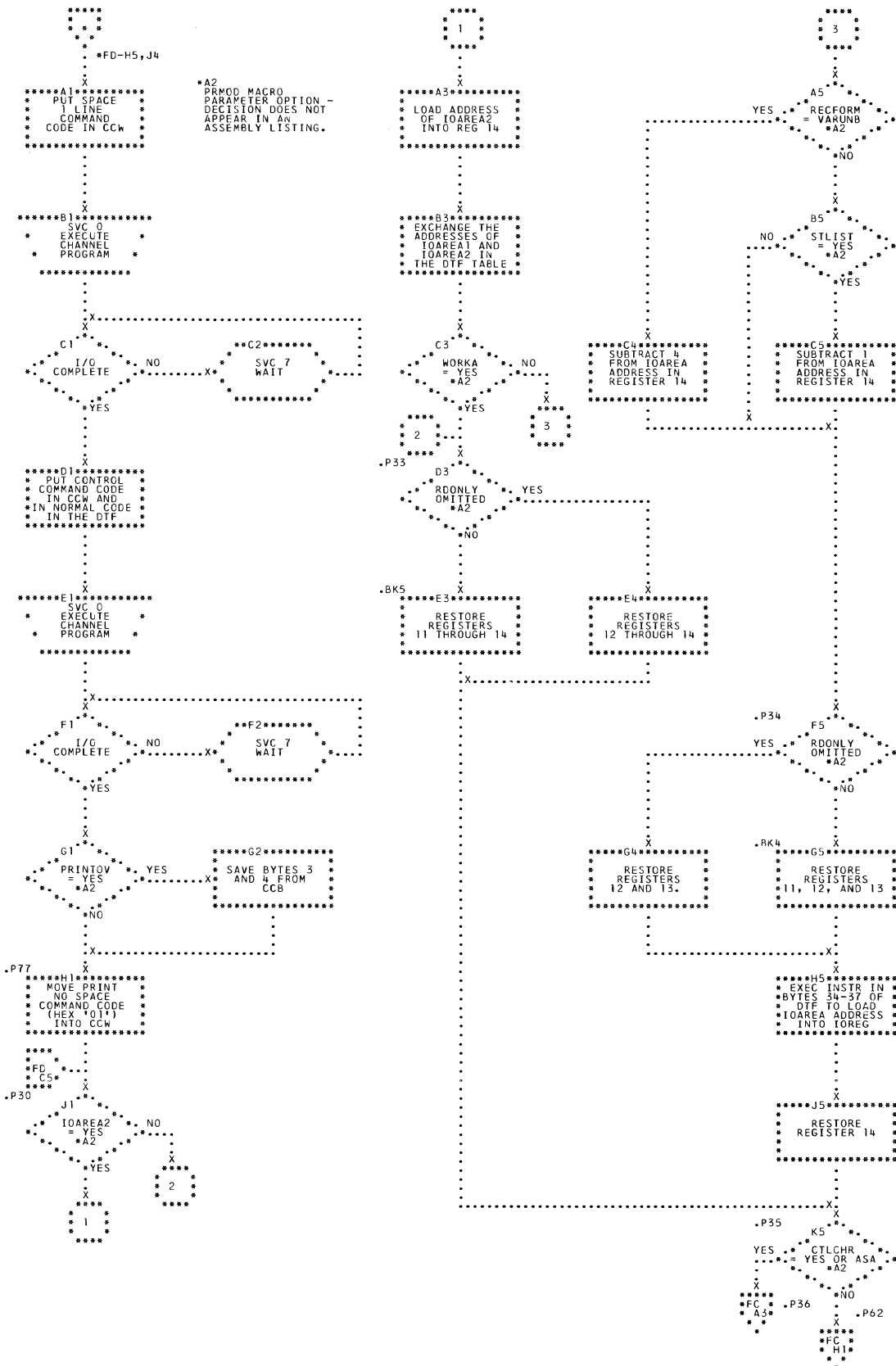


Chart GA. PTMOD: GET Macro, No Translation

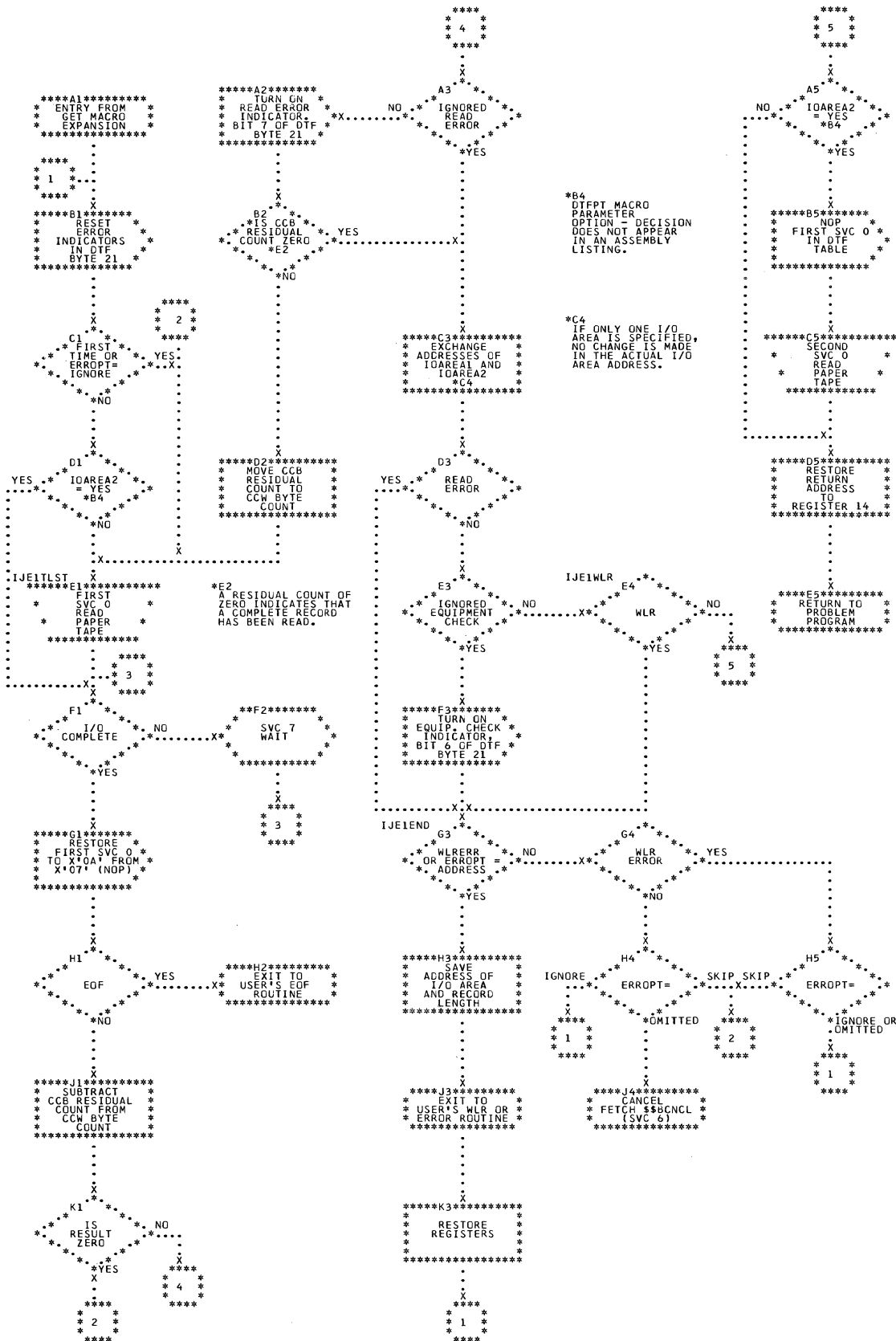


Chart GB. PTMOD: GET Macro, Translation of Shifted Code

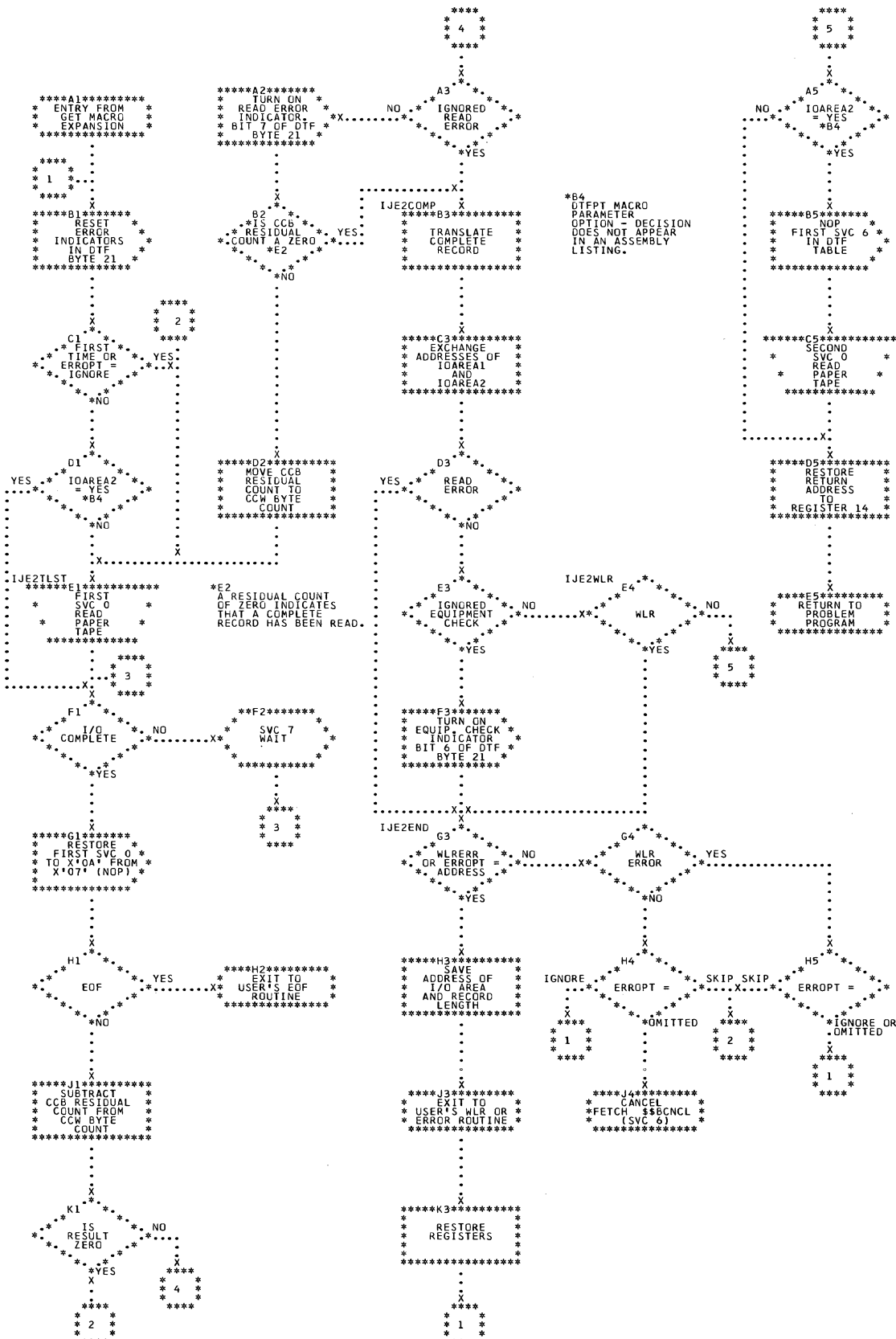


Chart GC. PTMOD: GET Macro; Translation of Shifted Code, Fixed Unblocked Records (Section 1 of 2)

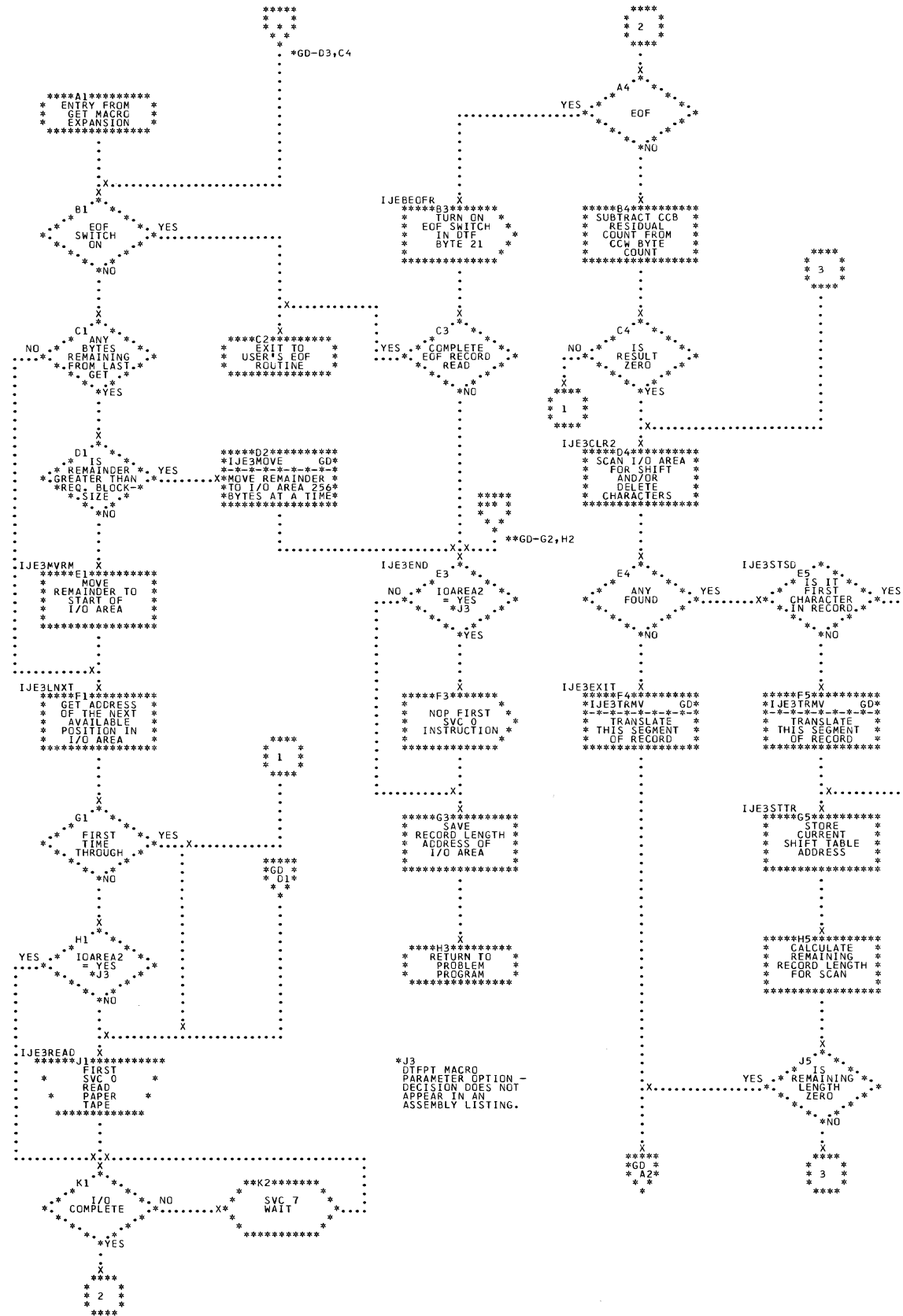


Chart GD. PTMOD: GET Macro; Translation of Shifted Code, Fixed Unblocked Records (Section 2 of 2)

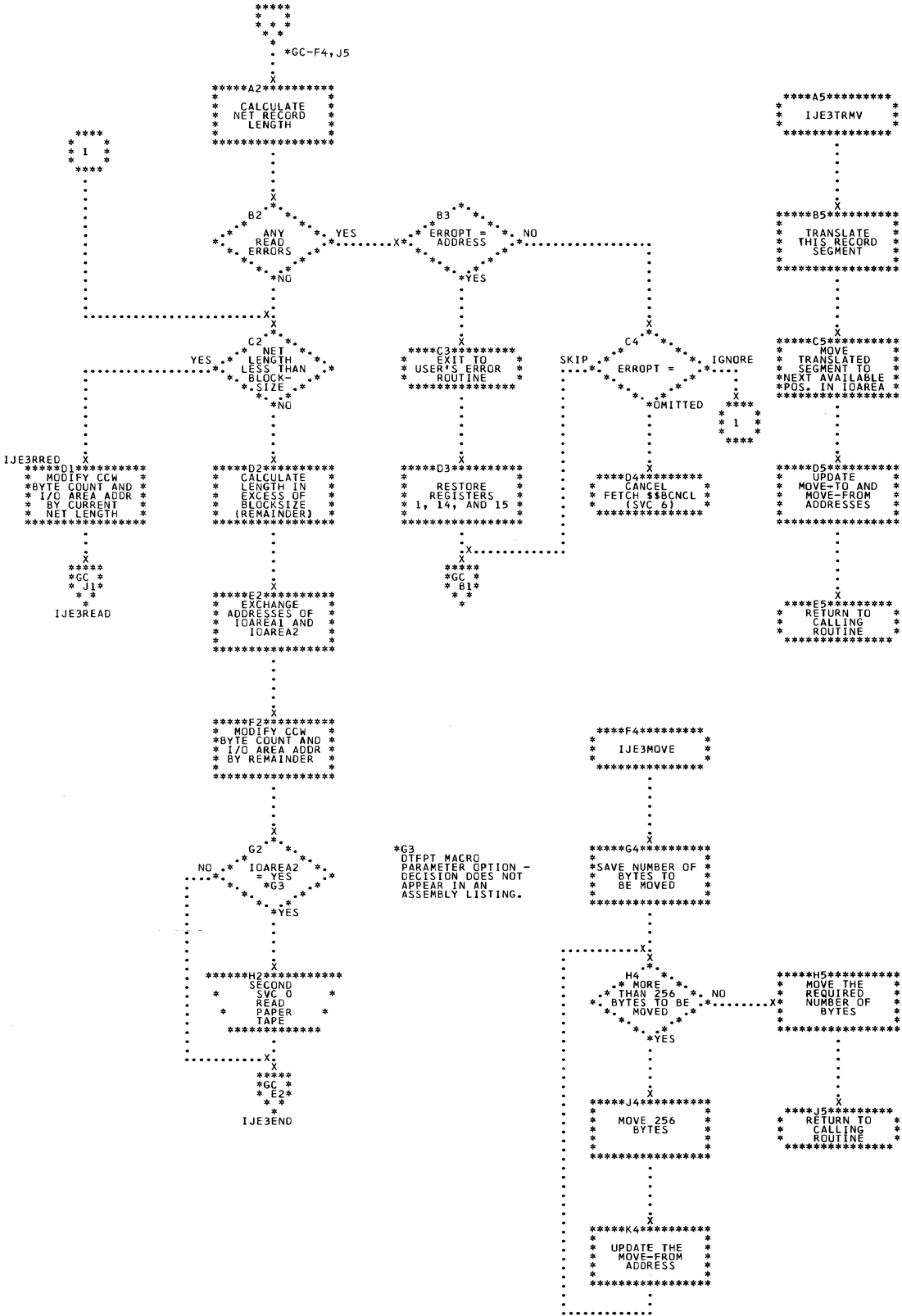


Chart GE. PTMOD: GET Macro; Translation of Shifted Code, Undefined Record
 (Section 1 of 2)

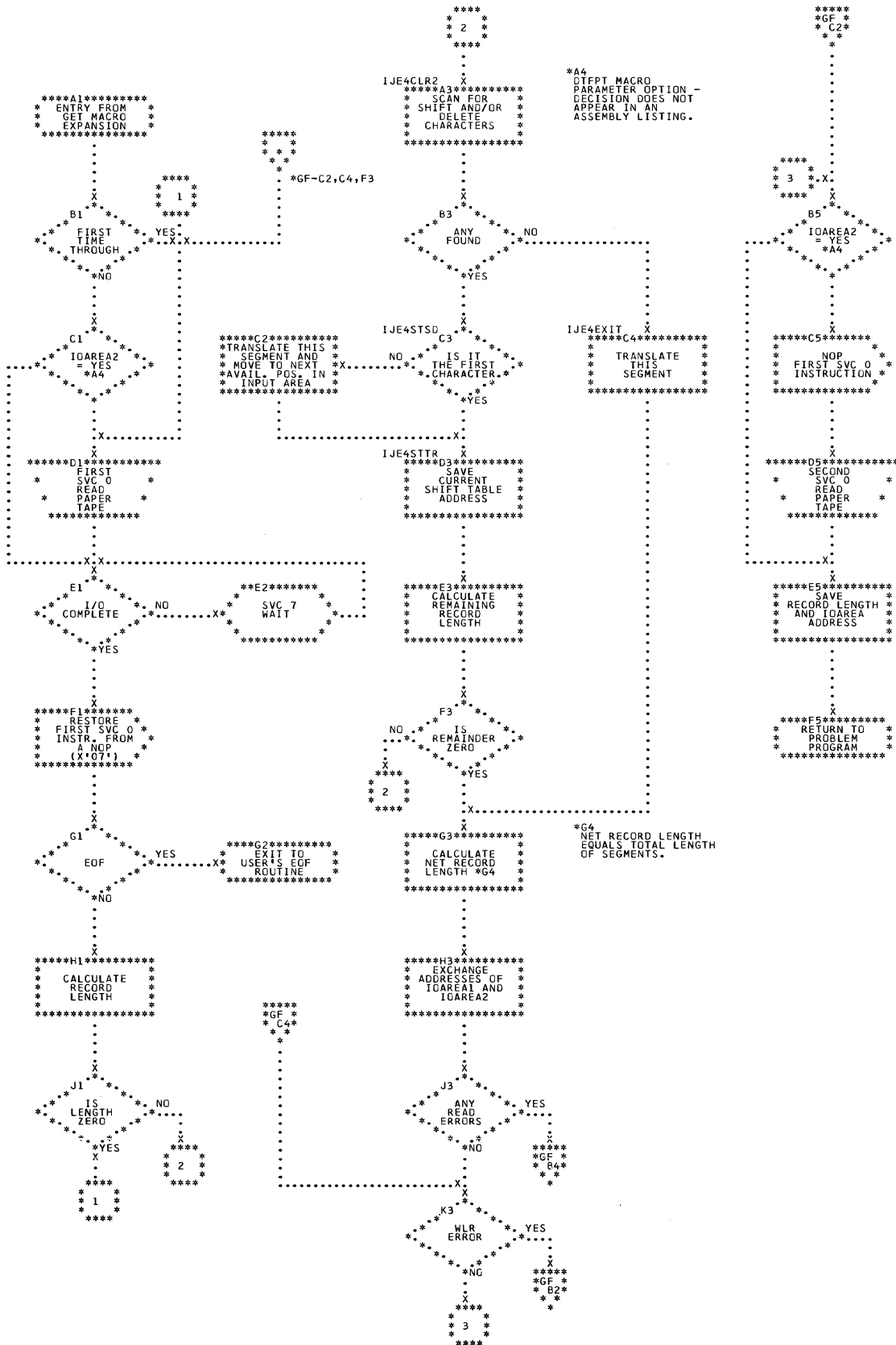
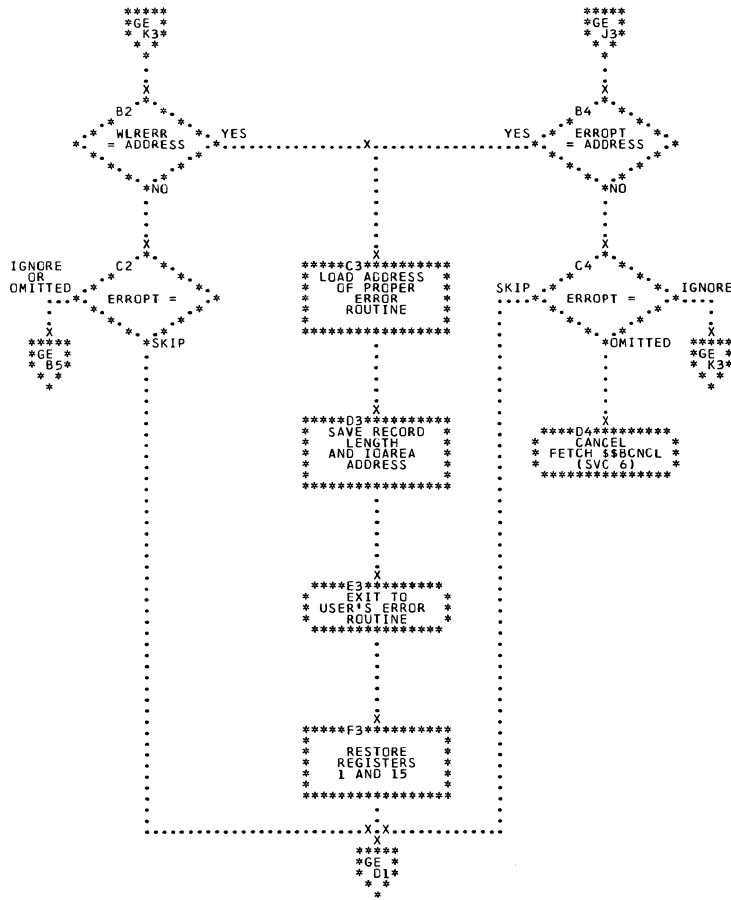
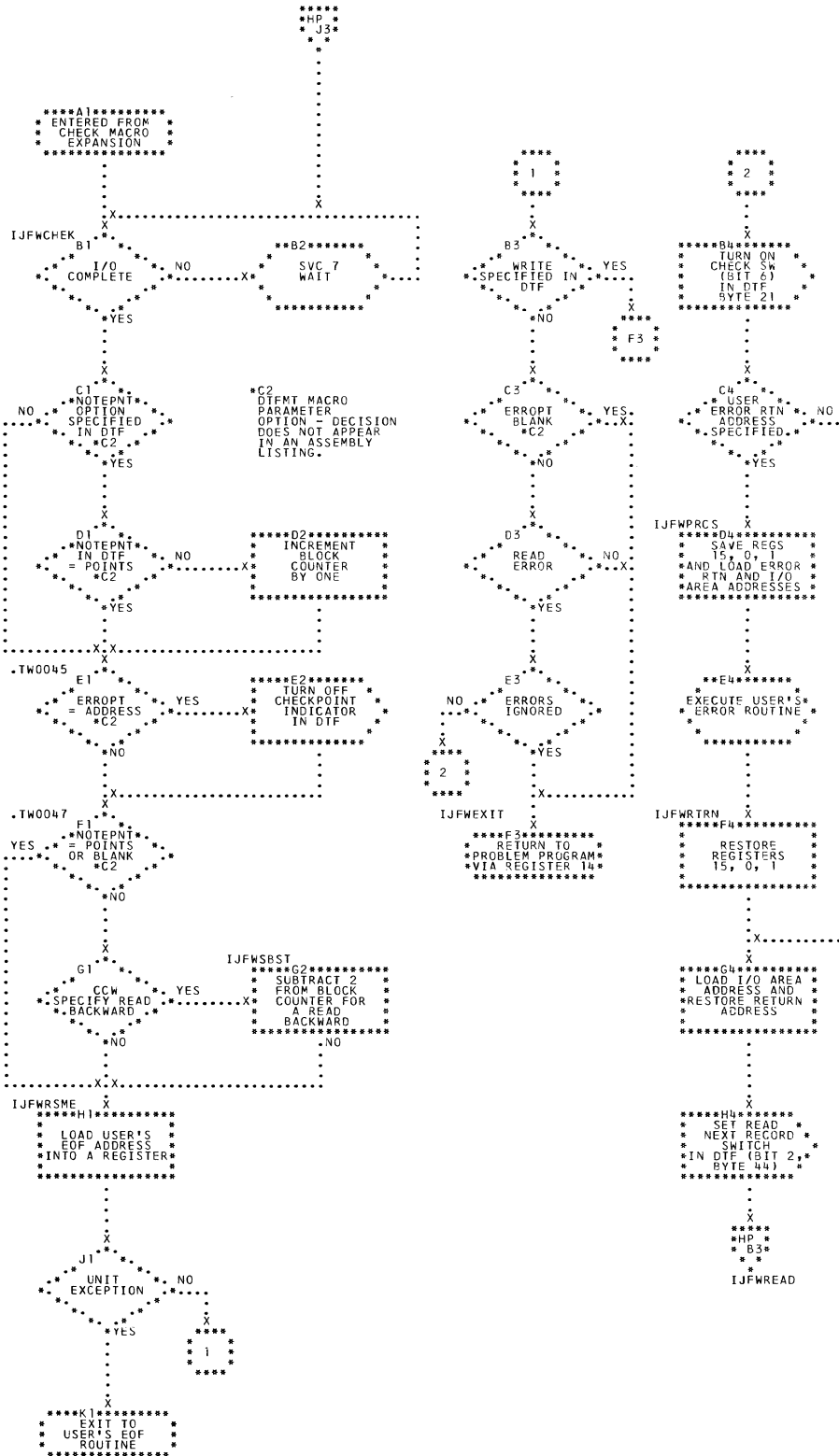


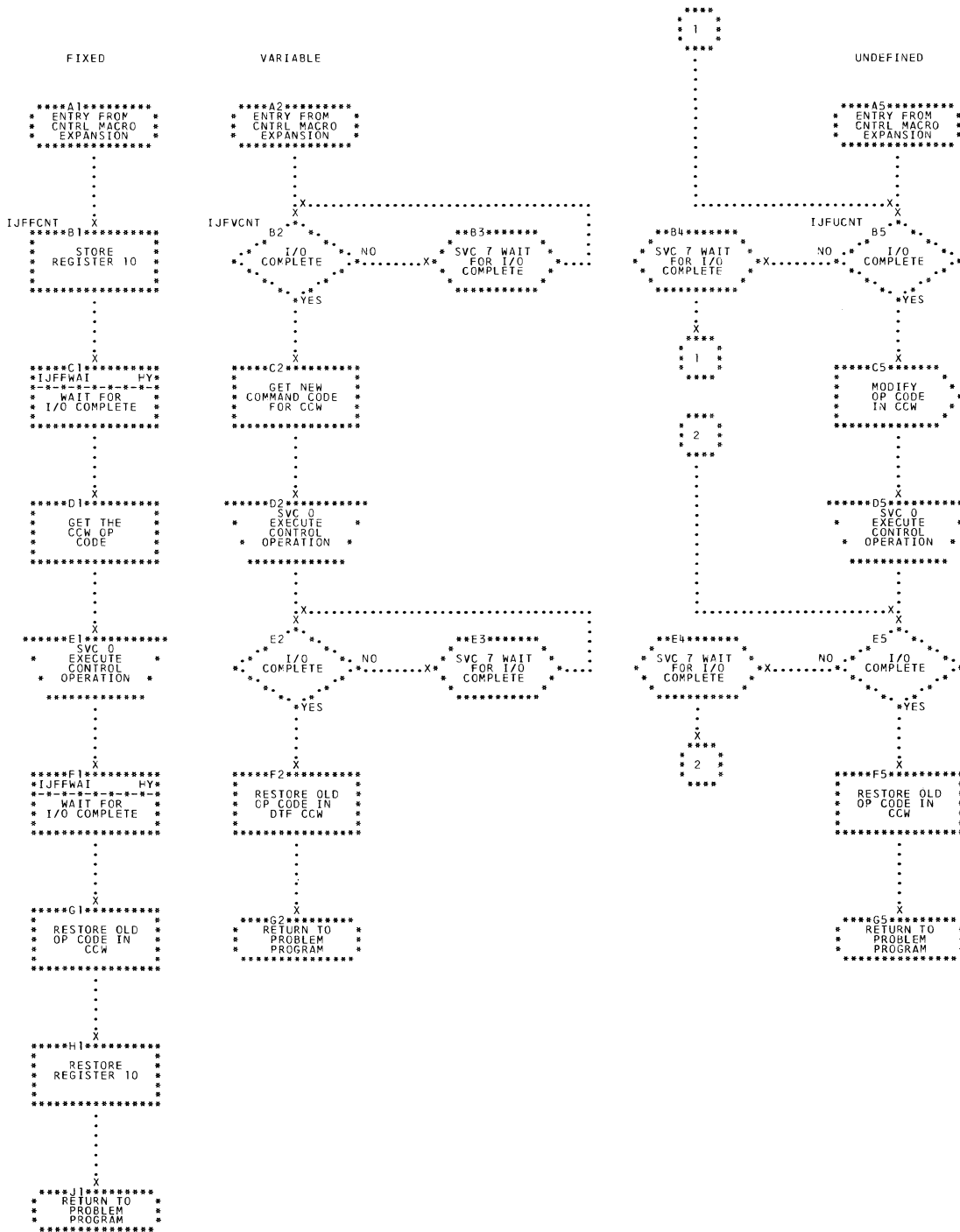
Chart GF. PTMOD: GET Macro; Translation of Shifted Code, Undefined Record
(Section 2 of 2)



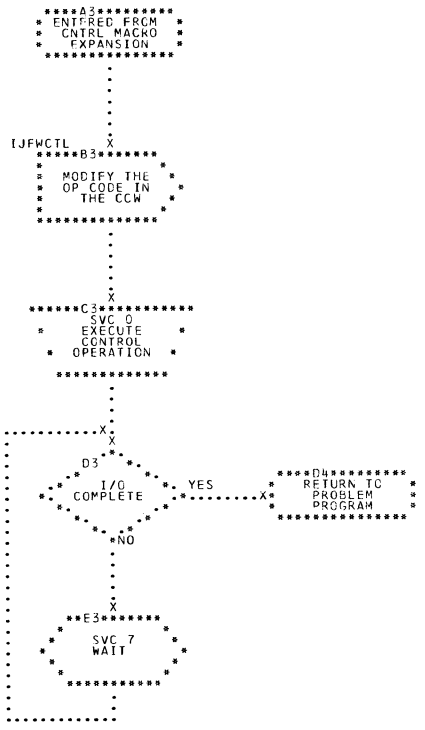
● Chart HA. MTMOD: CHECK Macro, Workfile



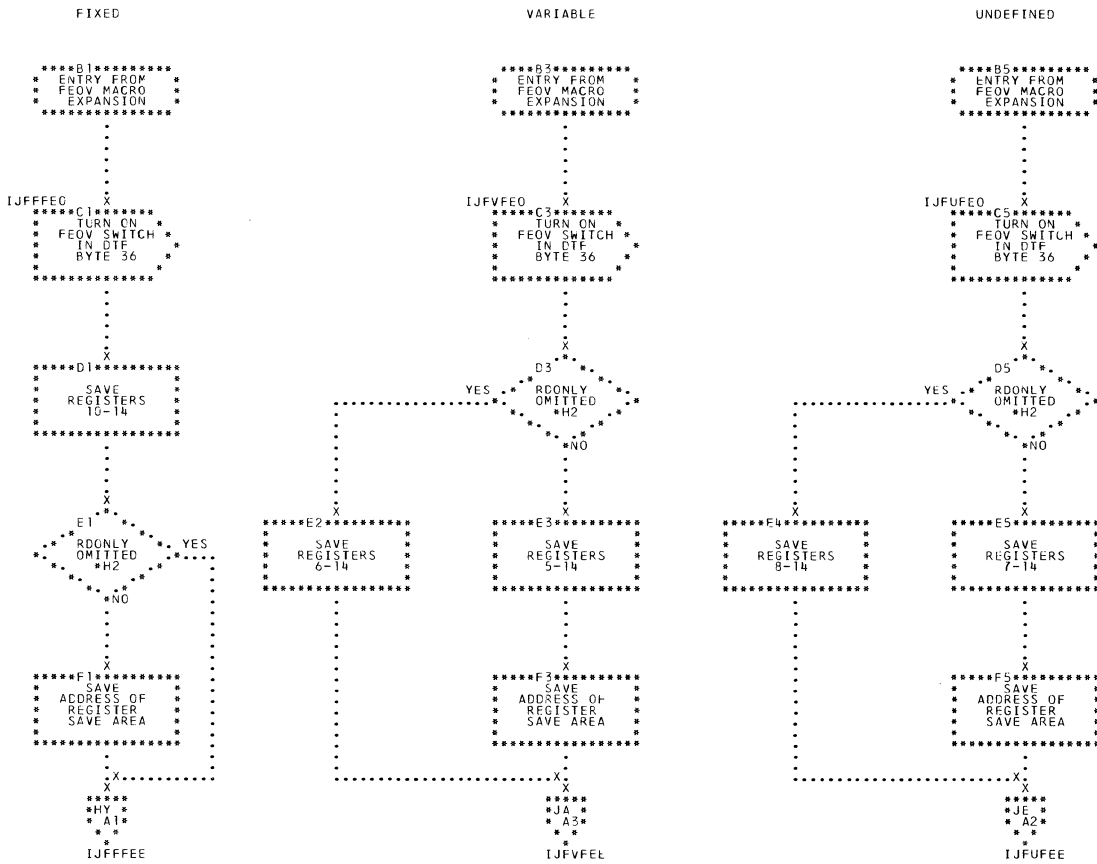
● Chart HB. MTMOD: CNTRL Macro, Data Files



● Chart HC. MTMOD: CNTRL Macro, Workfile

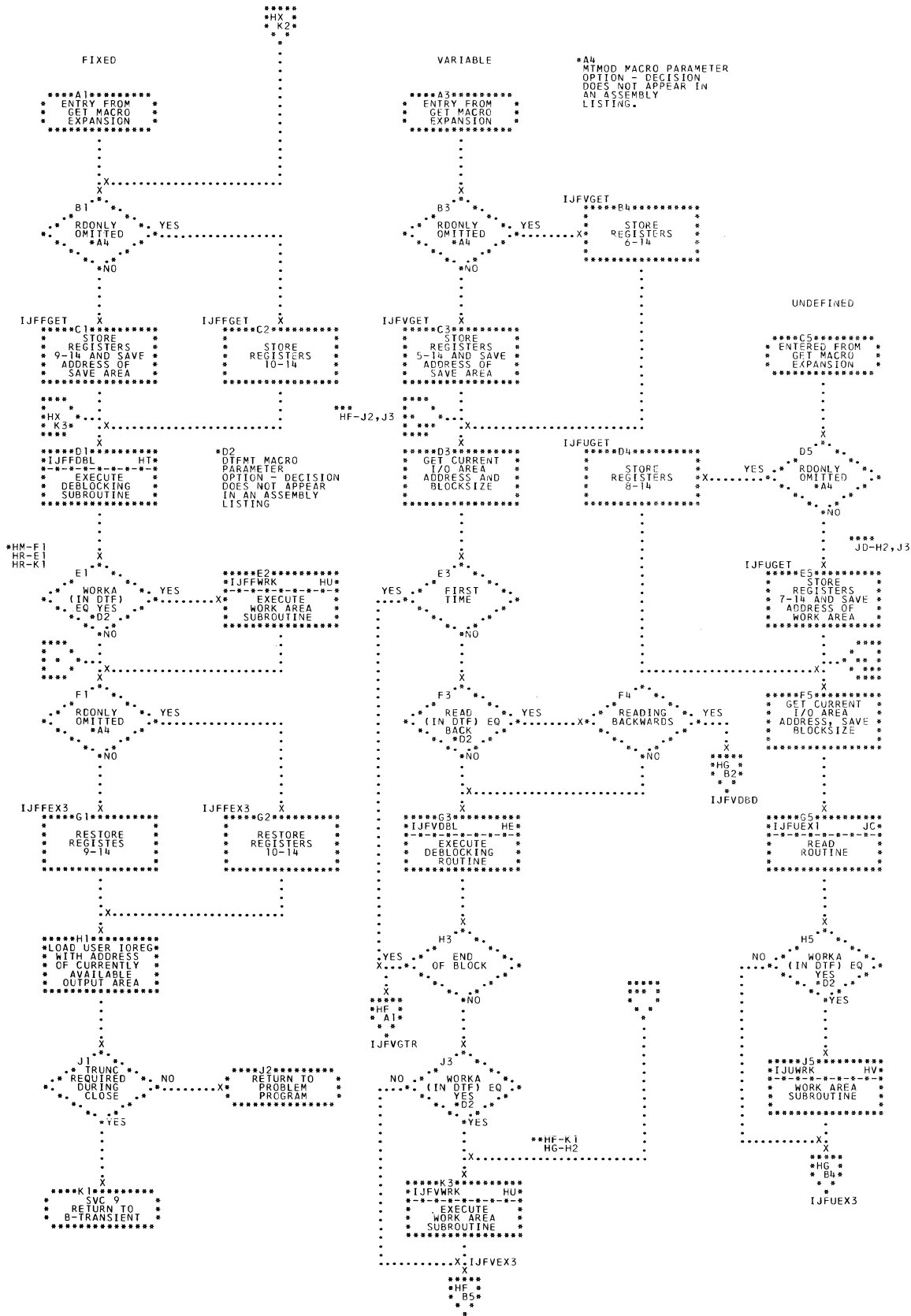


● Chart HD. MTMOD: FEOV Macro

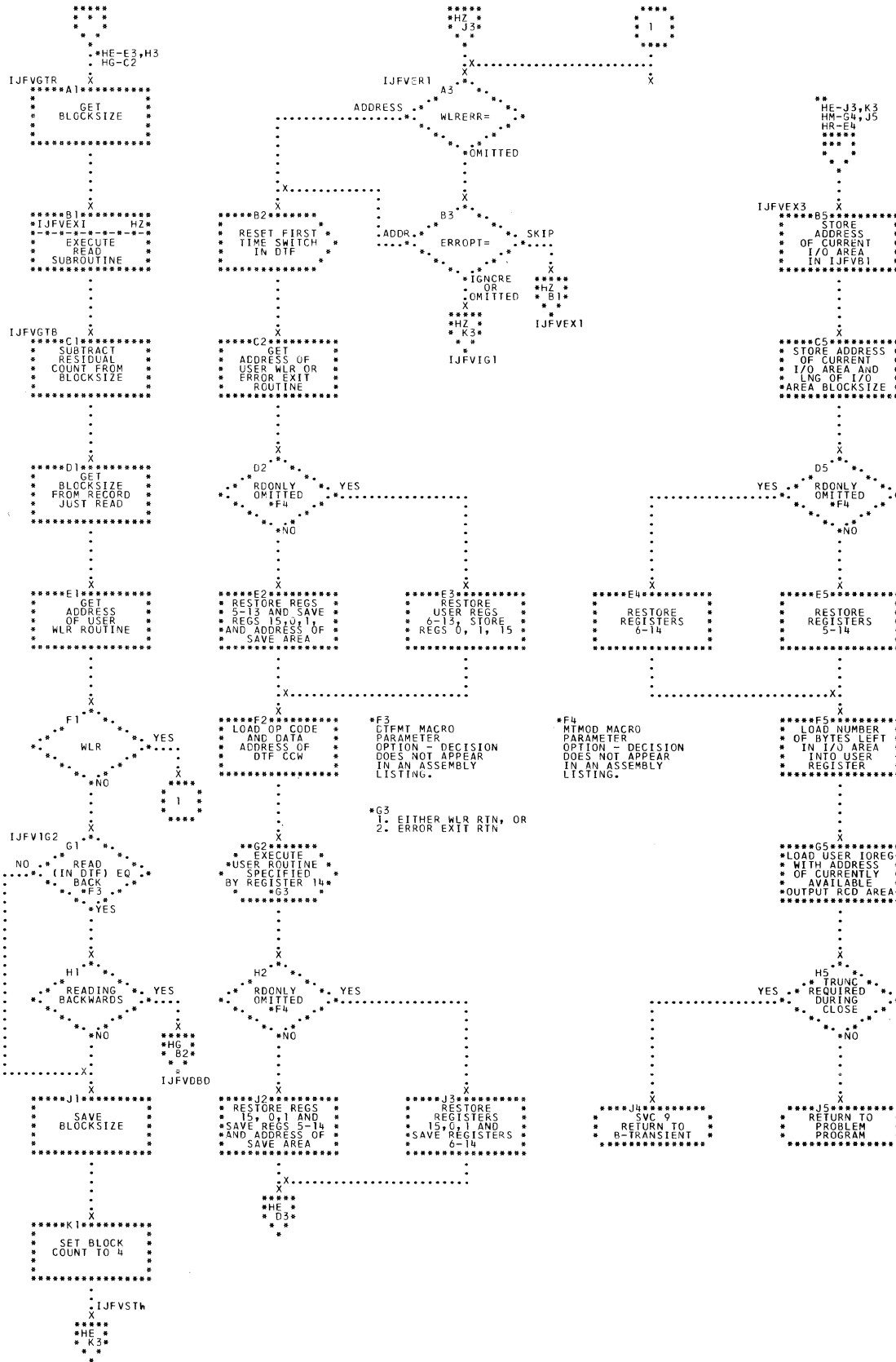


*H2
 MTMOD MACRO PARAMETER
 OPTION - DECISION
 DOES NOT APPEAR IN AN
 ASSEMBLY LISTING.

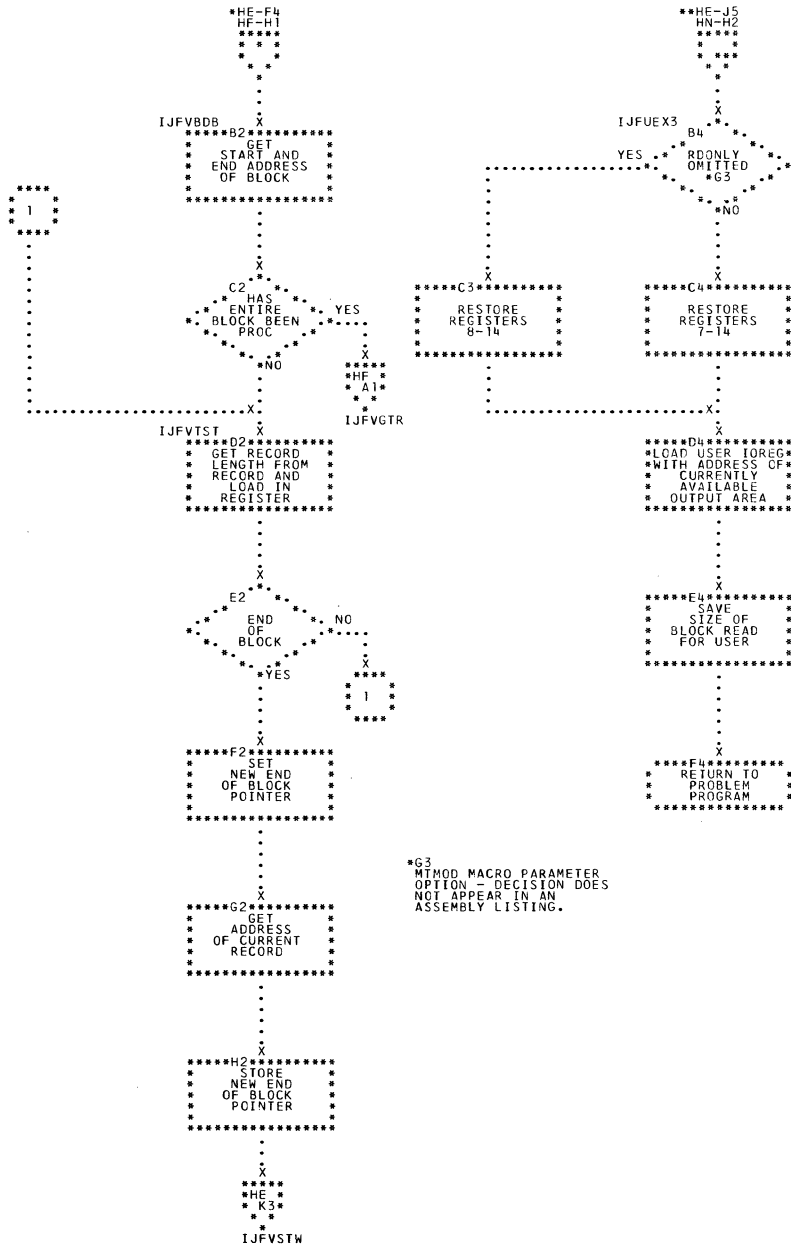
● Chart HE. MTMOD: GET Macro (Section 1 of 3)



● Chart HF. MTMOD: GET Macro (Section 2 of 3)



● Chart HG. MTMOD: GET Macro (Section 3 of 3)



```
*****B3*****  
* ENTERED FROM *  
* NOTE MACRO *  
* EXPANSION *  
*****  
.  
.  
.  
.  
.  
X  
IJFWNOTE  
*****C3*****  
* LOAD REG 1 *  
* WITH BLOCK *  
* COUNT FROM *  
* BYTES 40-43 *  
* OF DTF TABLE *  
*****  
.  
.  
.  
.  
.  
X  
*****D3*****  
* RETURN TO *  
* PROGRAM *  
*****
```

Chart HJ. MTMOD: POINTR Macro, Workfile

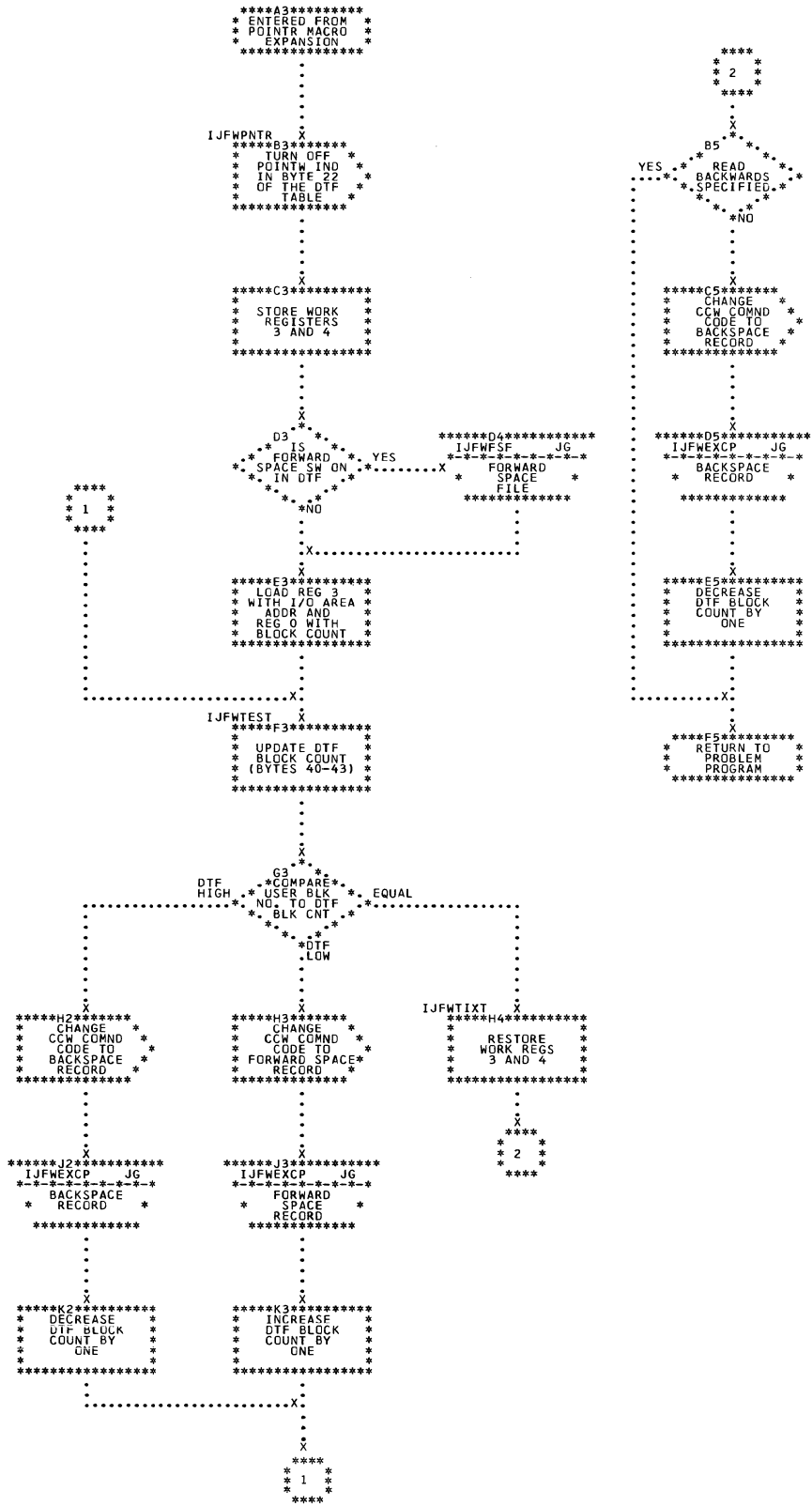
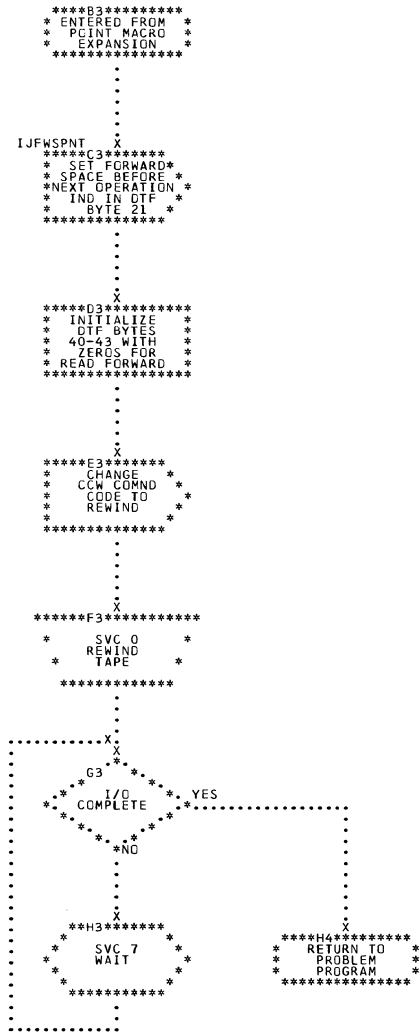
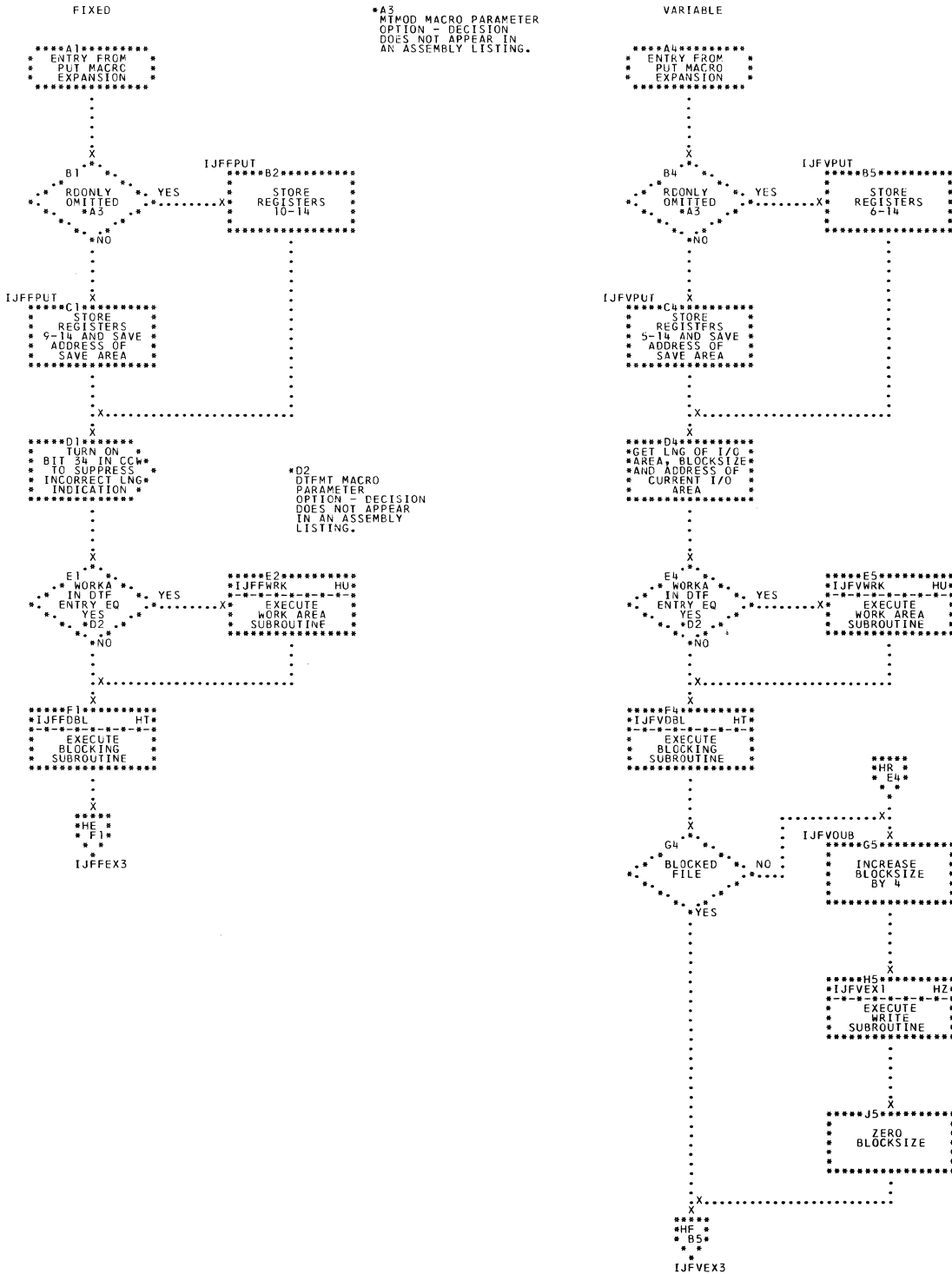


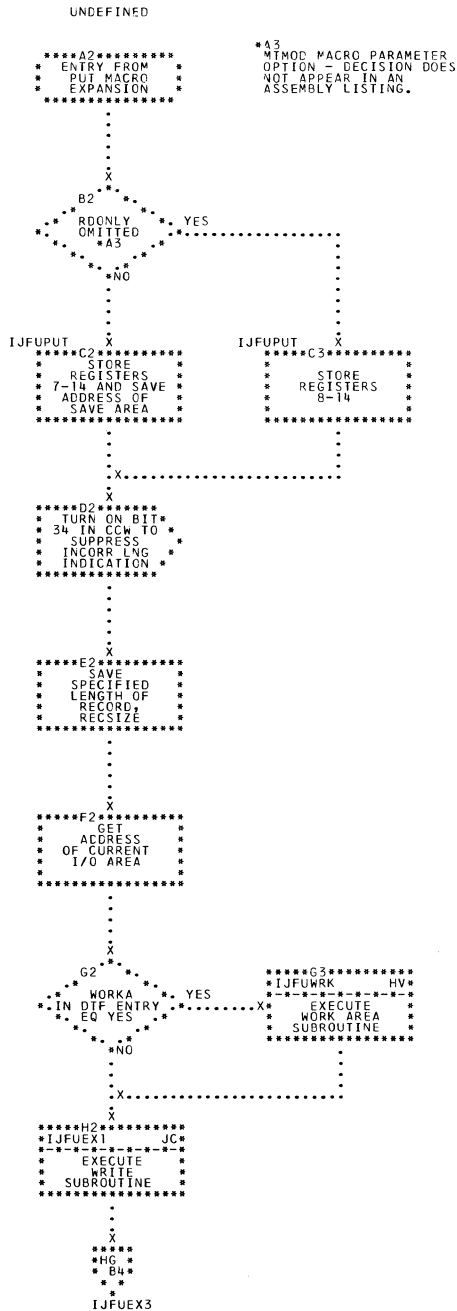
Chart HK. MTMOD: POINTS Macro, Workfile



● Chart HM. MTMOD: PUT Macro (Section 1 of 2)



● Chart HN. MTMOD: PUT Macro (Section 2 of 2)



● Chart HP. MTMOD: READ Macro, Workfile

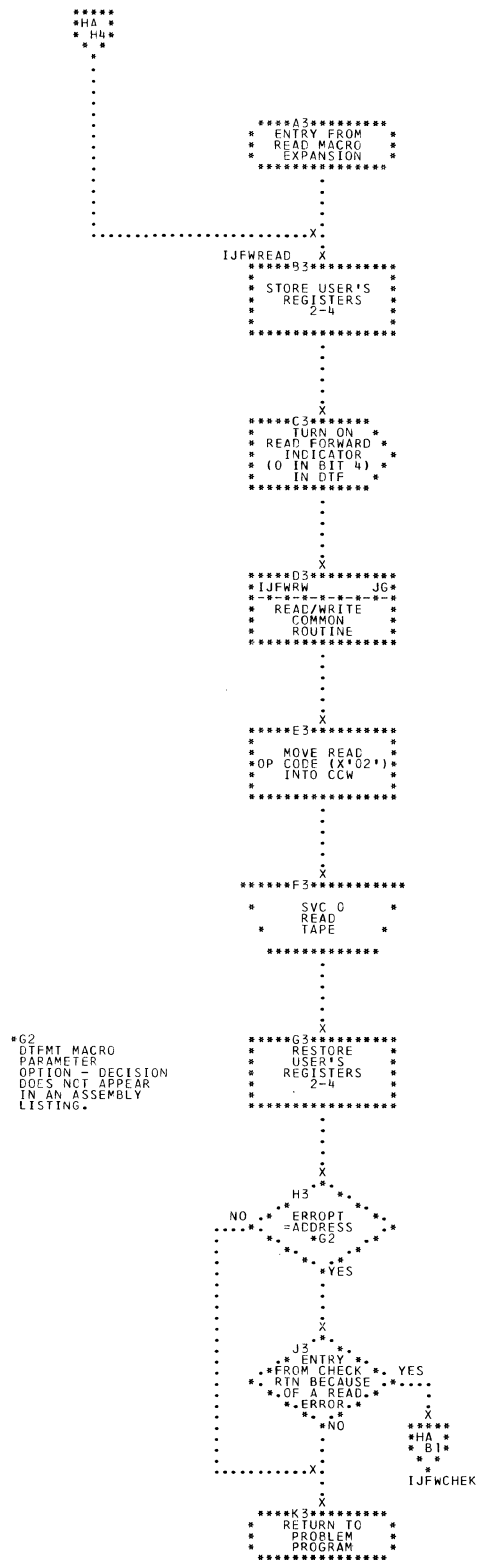
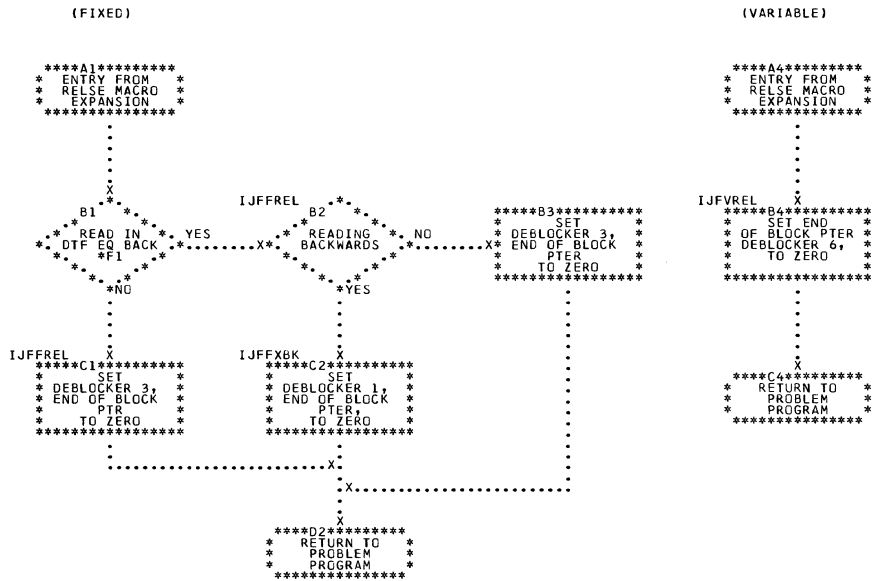
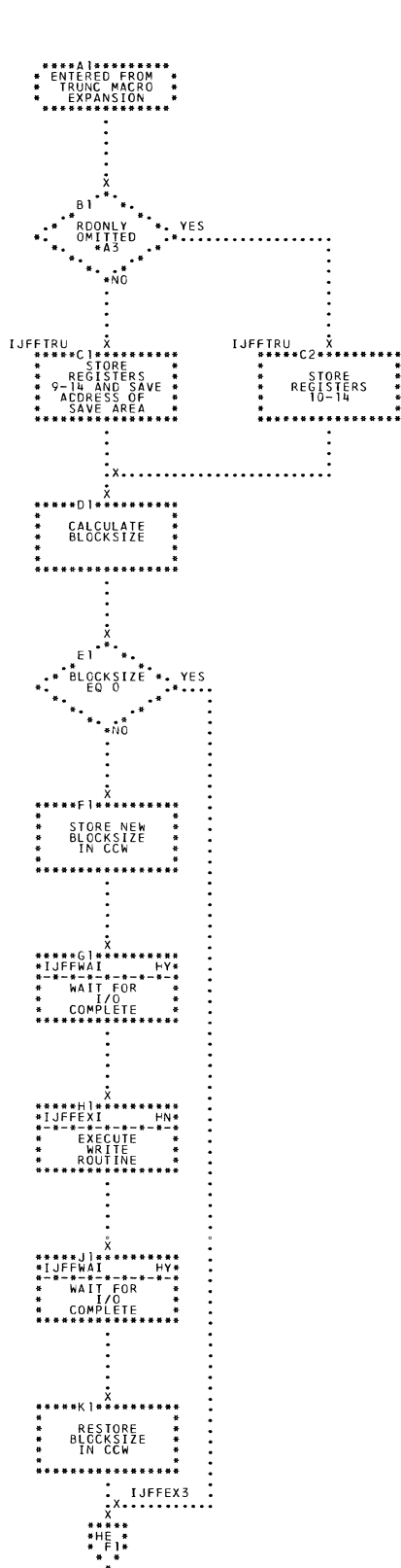


Chart HQ. MTMOD: RELSE Macro



*F1
DTFMT MACRO
PARAMETER OPTION -
DECISION DOES NOT
APPEAR IN AN
ASSEMBLY LISTING.

● Chart HR. MTMOD: TRUNC Macro



*A3
METHOD MACRO PARAMETER
OPTION - DECISION DOES
NOT APPEAR IN AN
ASSEMBLY LISTING.

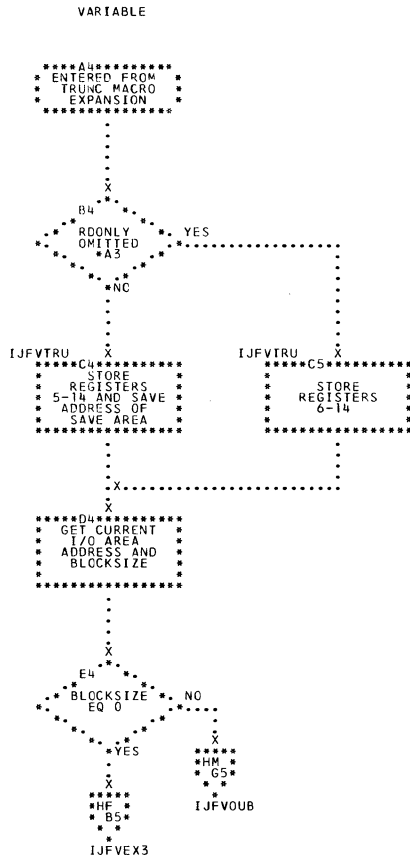
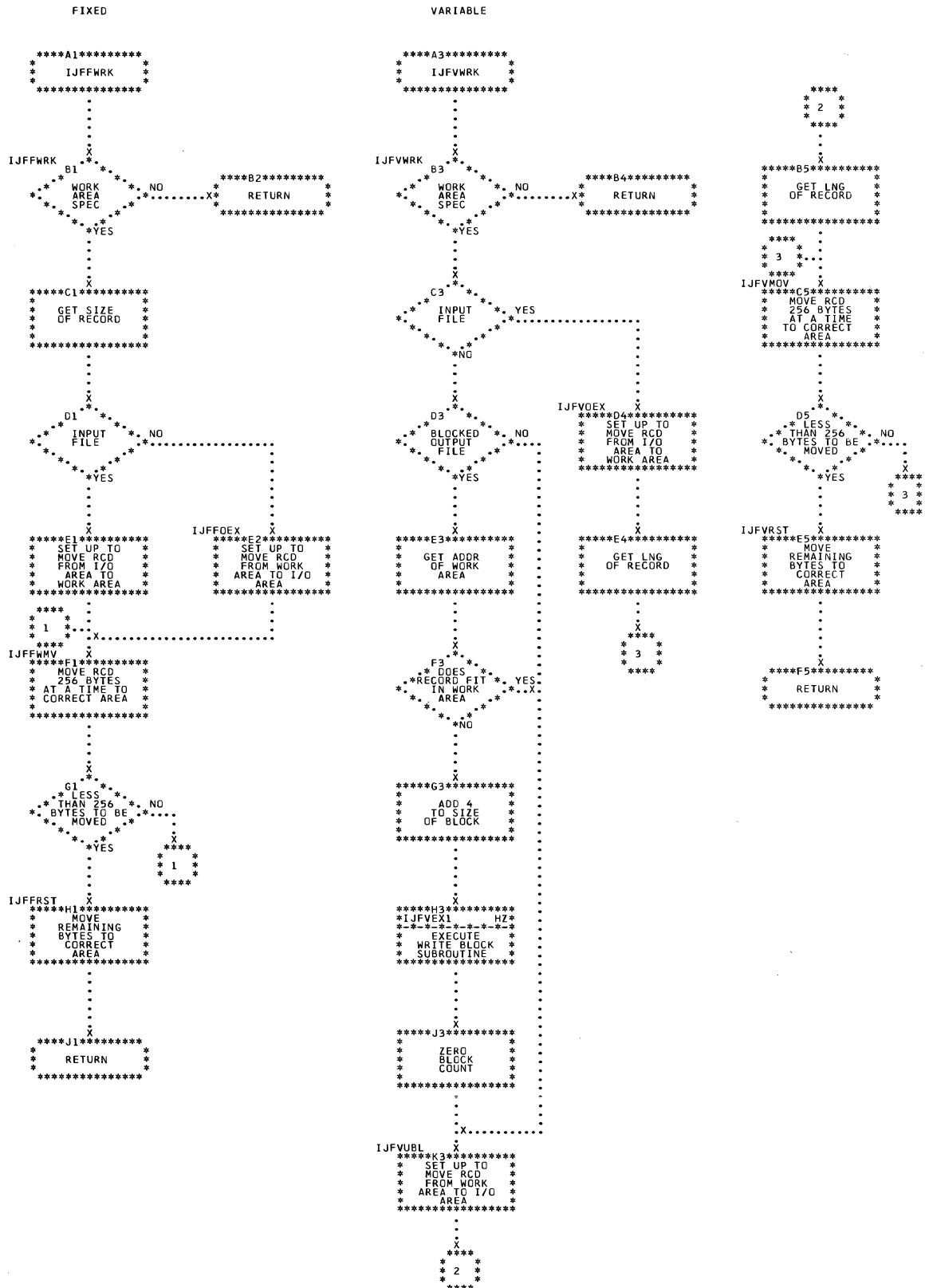
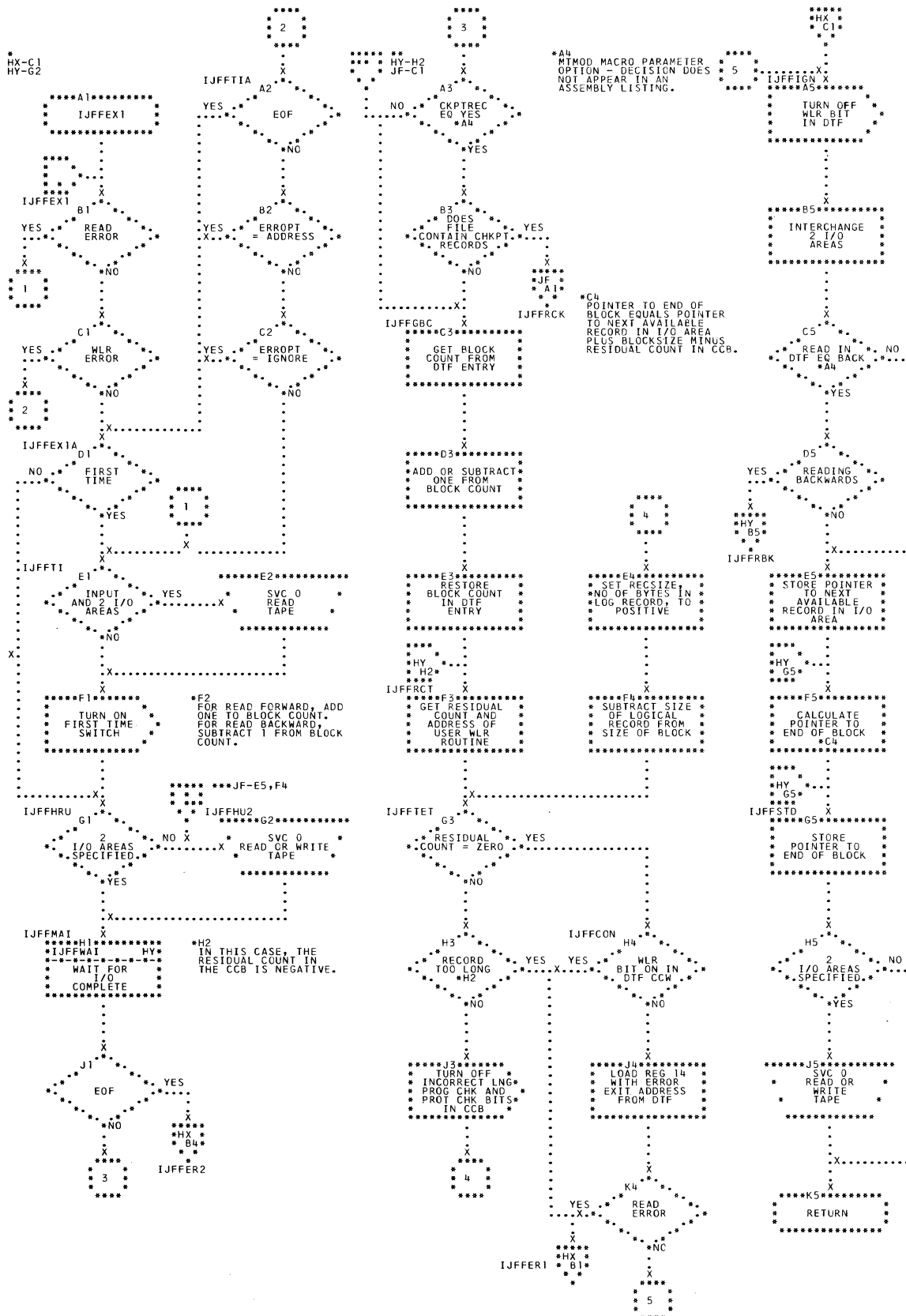


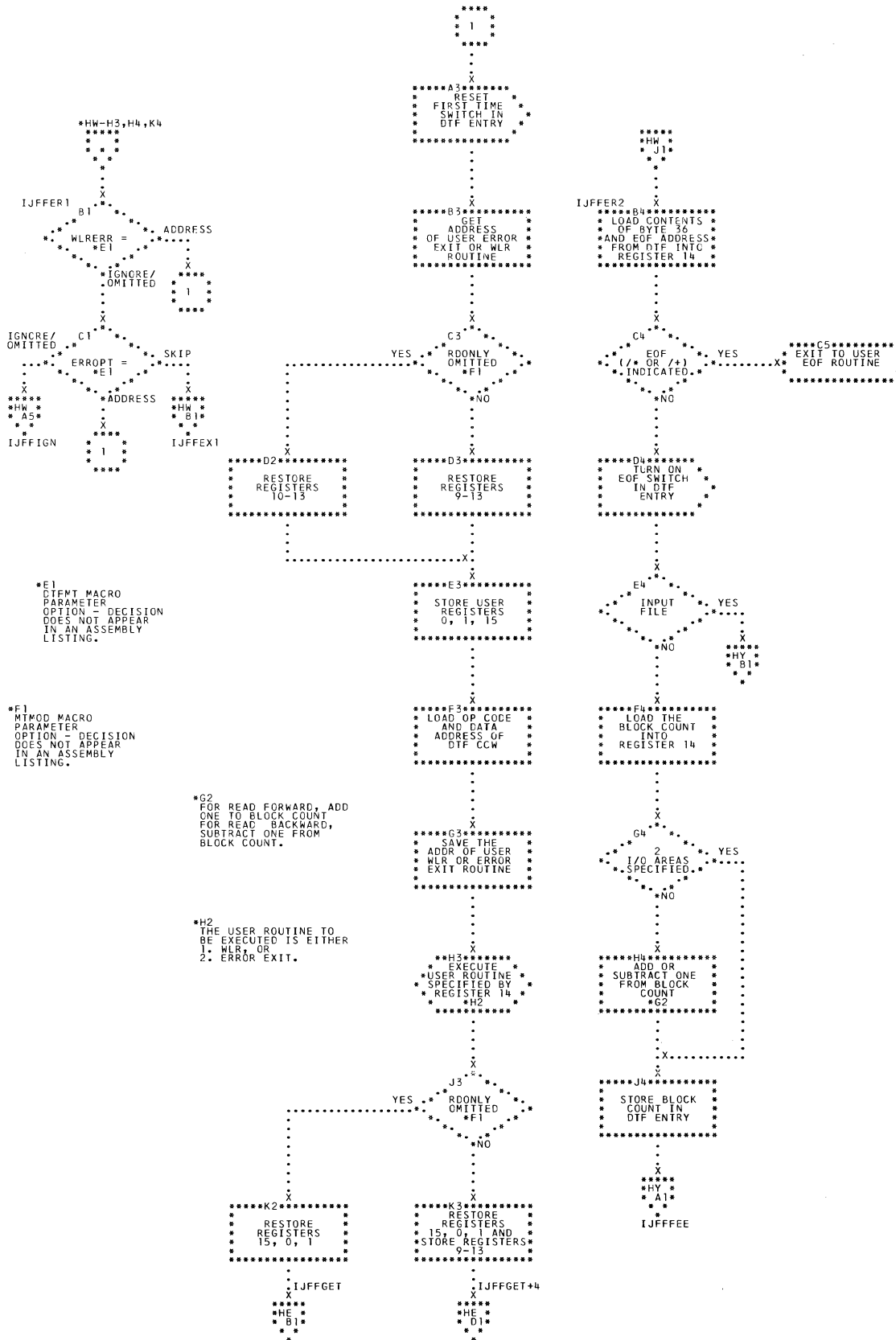
Chart HU. MTMOD: Work Area Subroutines, Fixed and Variable Records



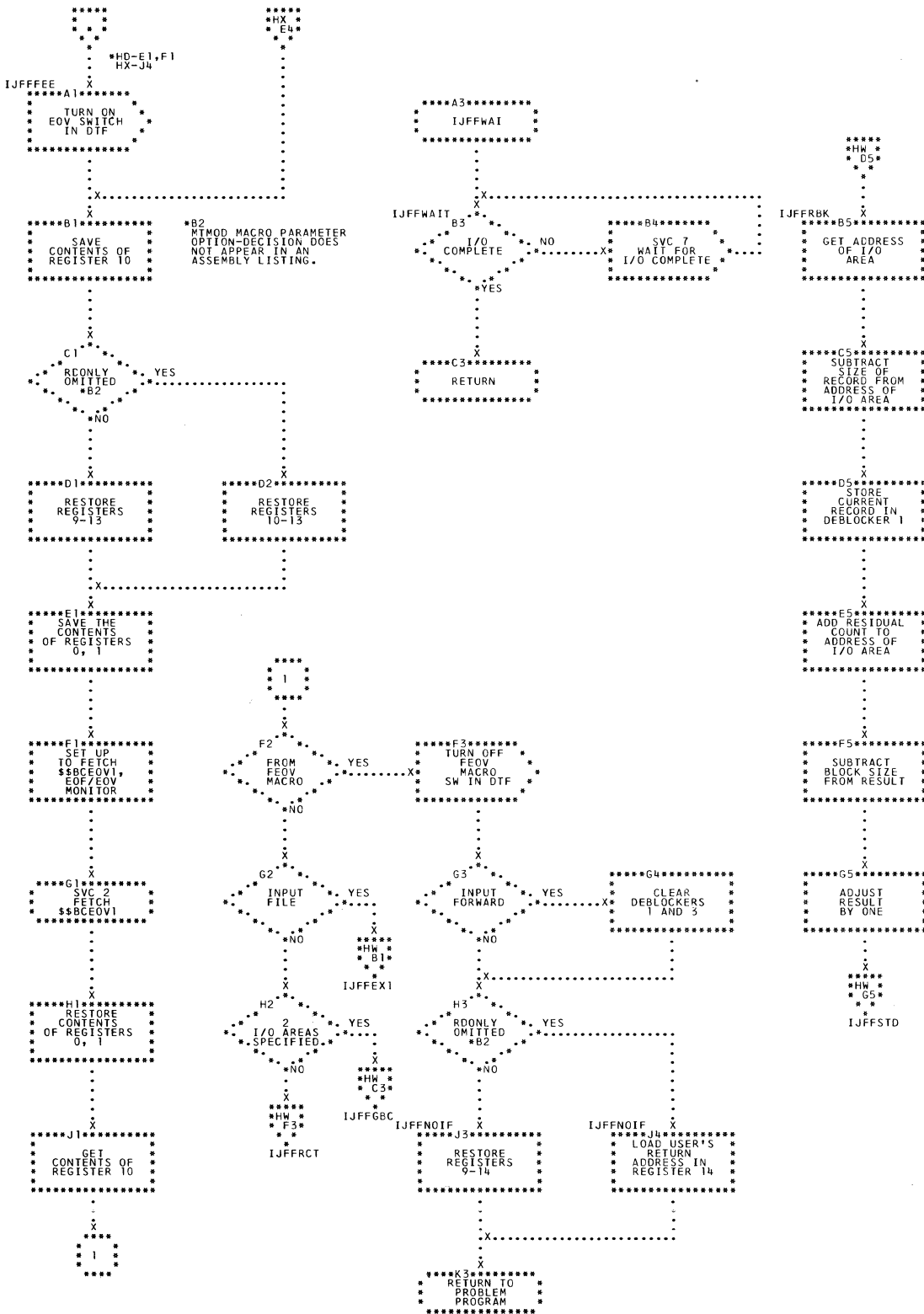
● Chart HW. MTMOD: Read/Write Subroutine, Fixed Records (Section 1 of 3)



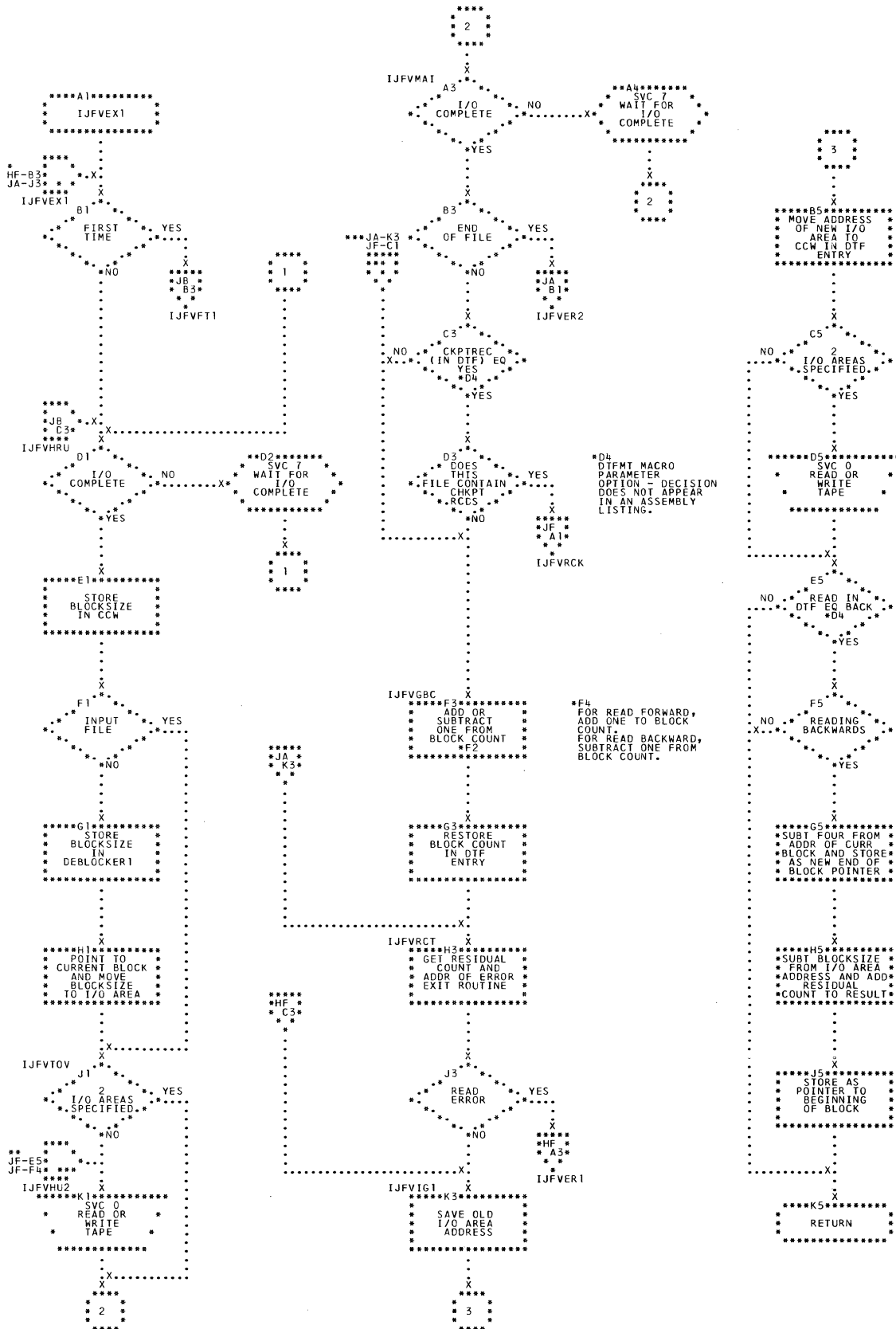
● Chart HX. MTMOD: Read/Write Subroutine, Fixed Records (Section 2 of 3)



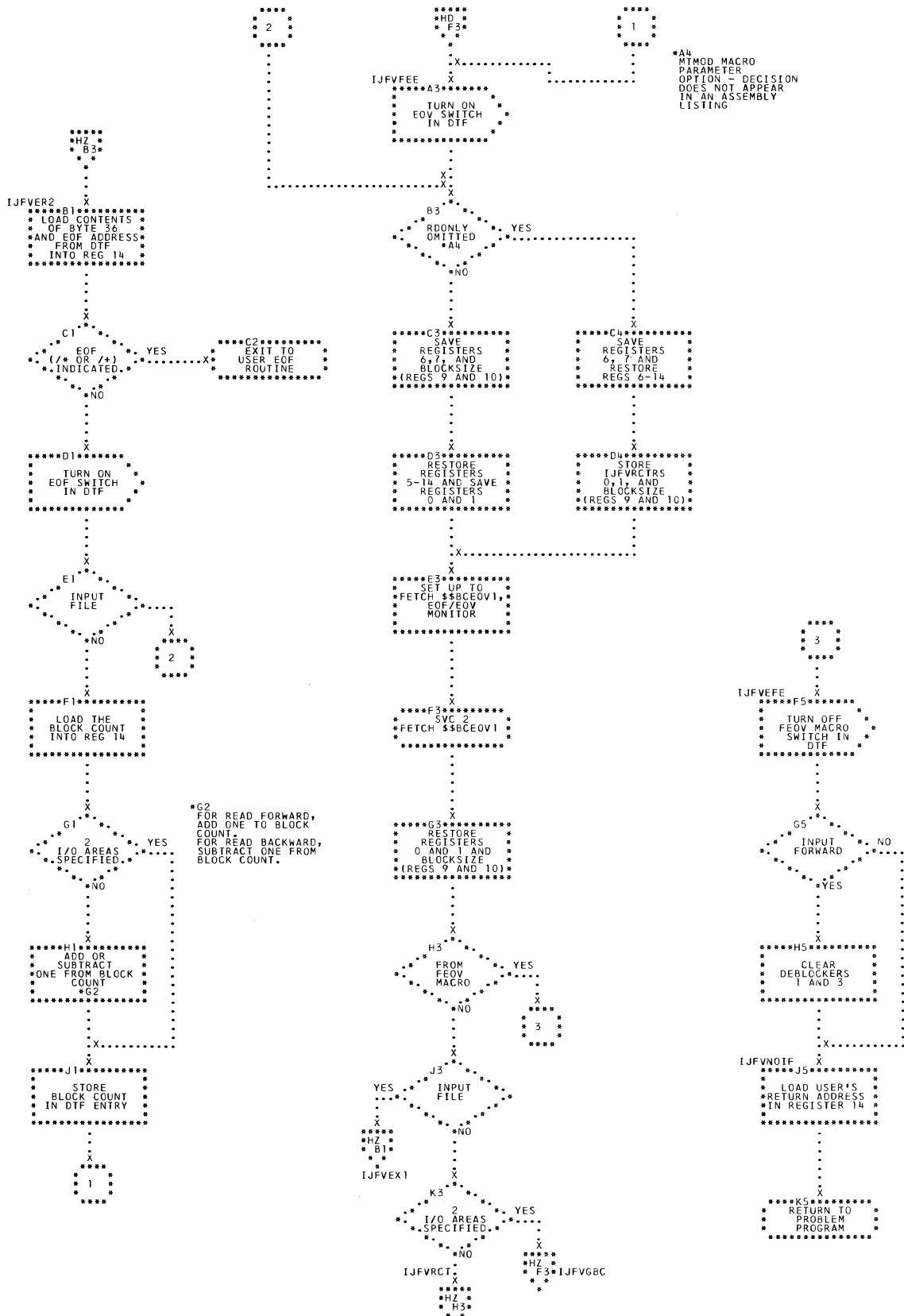
● Chart HY. MTMOD: Read/Write Subroutine, Fixed Records (Section 3 of 3)



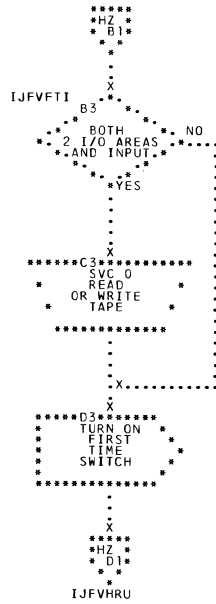
● Chart HZ. MTMOD: Read/Write Subroutine, Variable Records (Section 1 of 3)



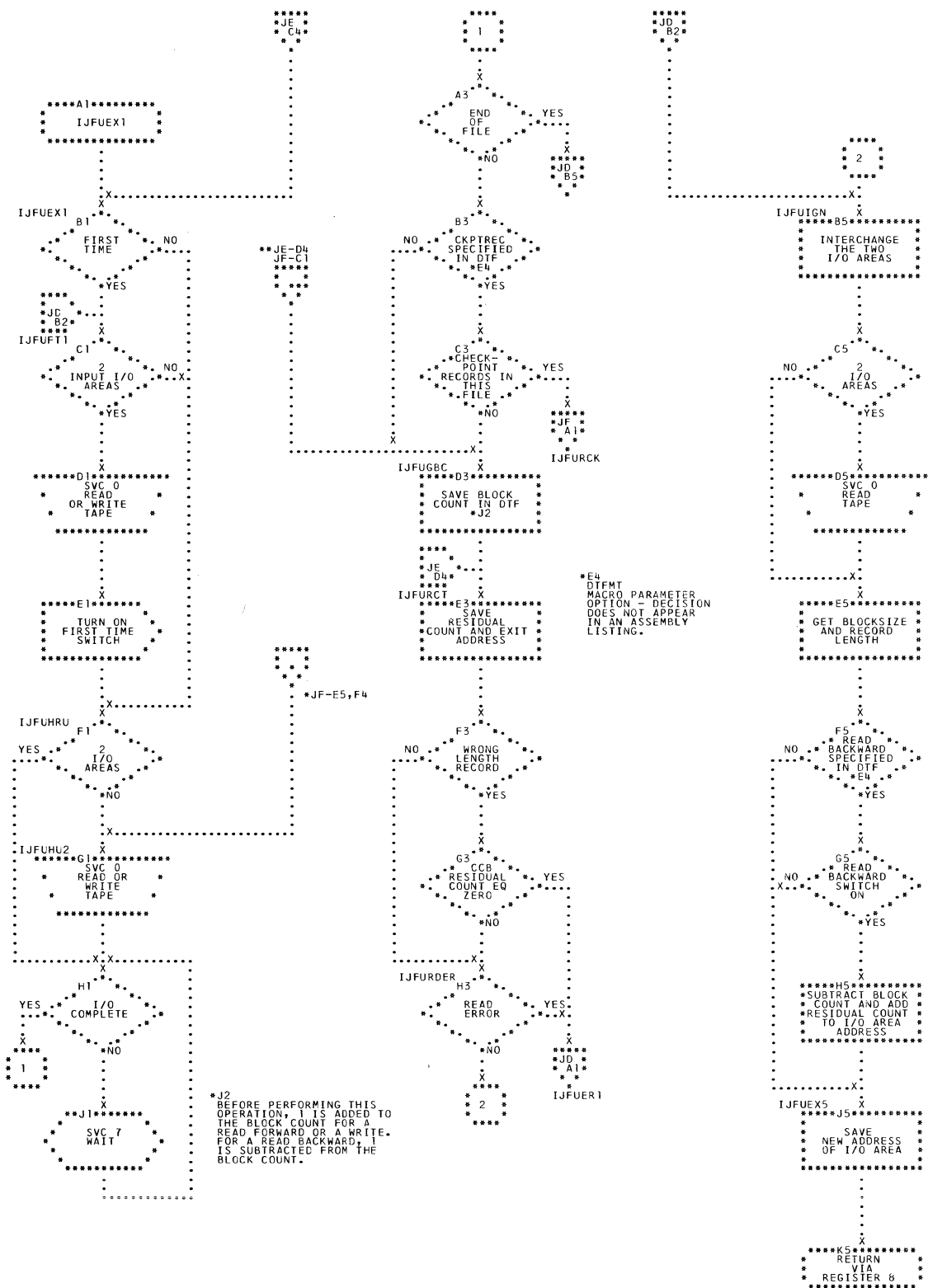
● Chart JA. MTMOD: Read/Write Subroutine, Variable Records (Section 2 of 3)



● Chart JB. MTMOD: Read/Write Subroutine, Variable Records (Section 3 of 3)



● Chart JC. MTMOD: Read/Write Subroutine, Undefined Records (Section 1 of 3)



● Chart JD. MDMOD: Read/Write Subroutine, Undefined Records (Section 2 of 3)

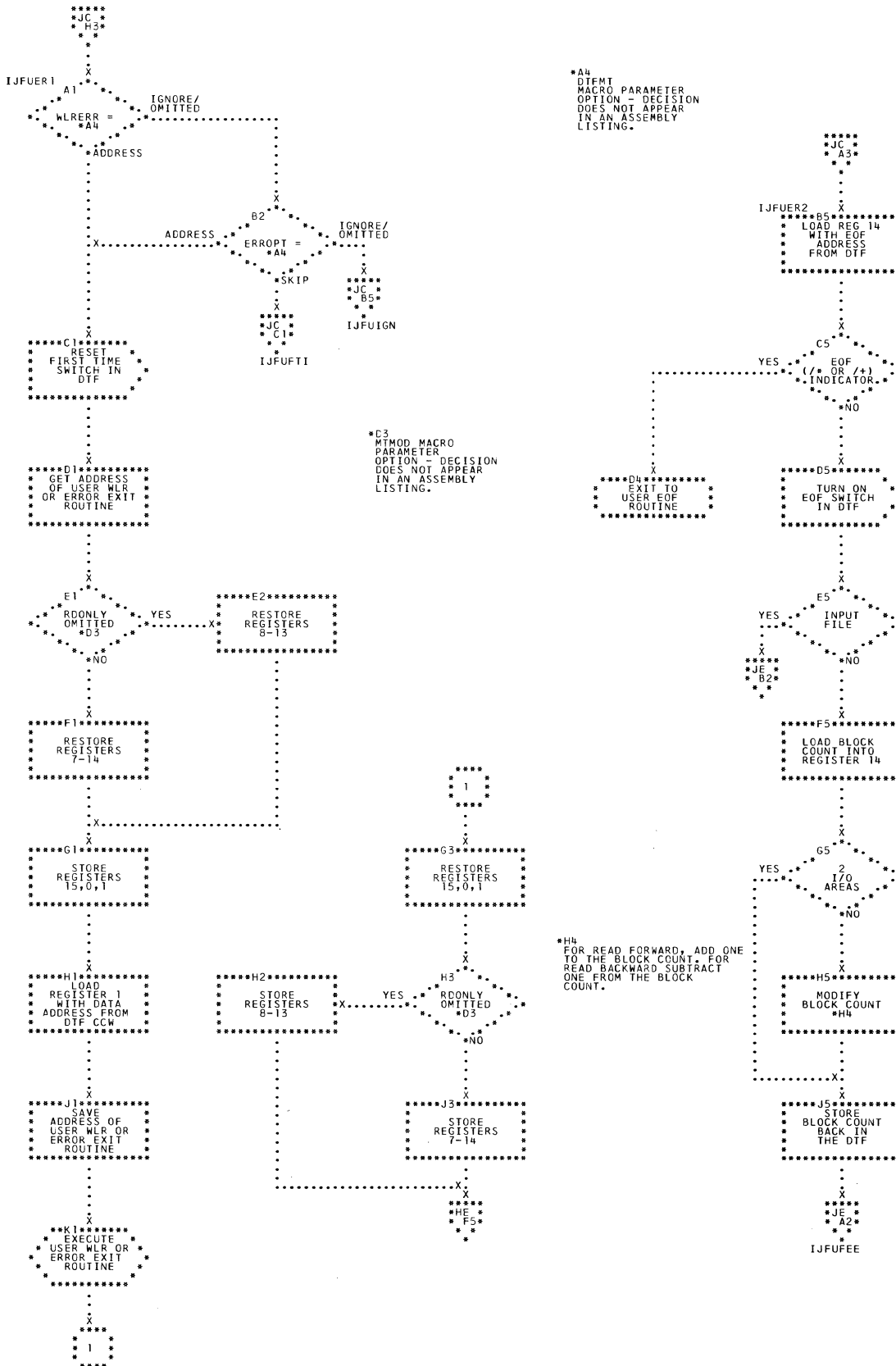
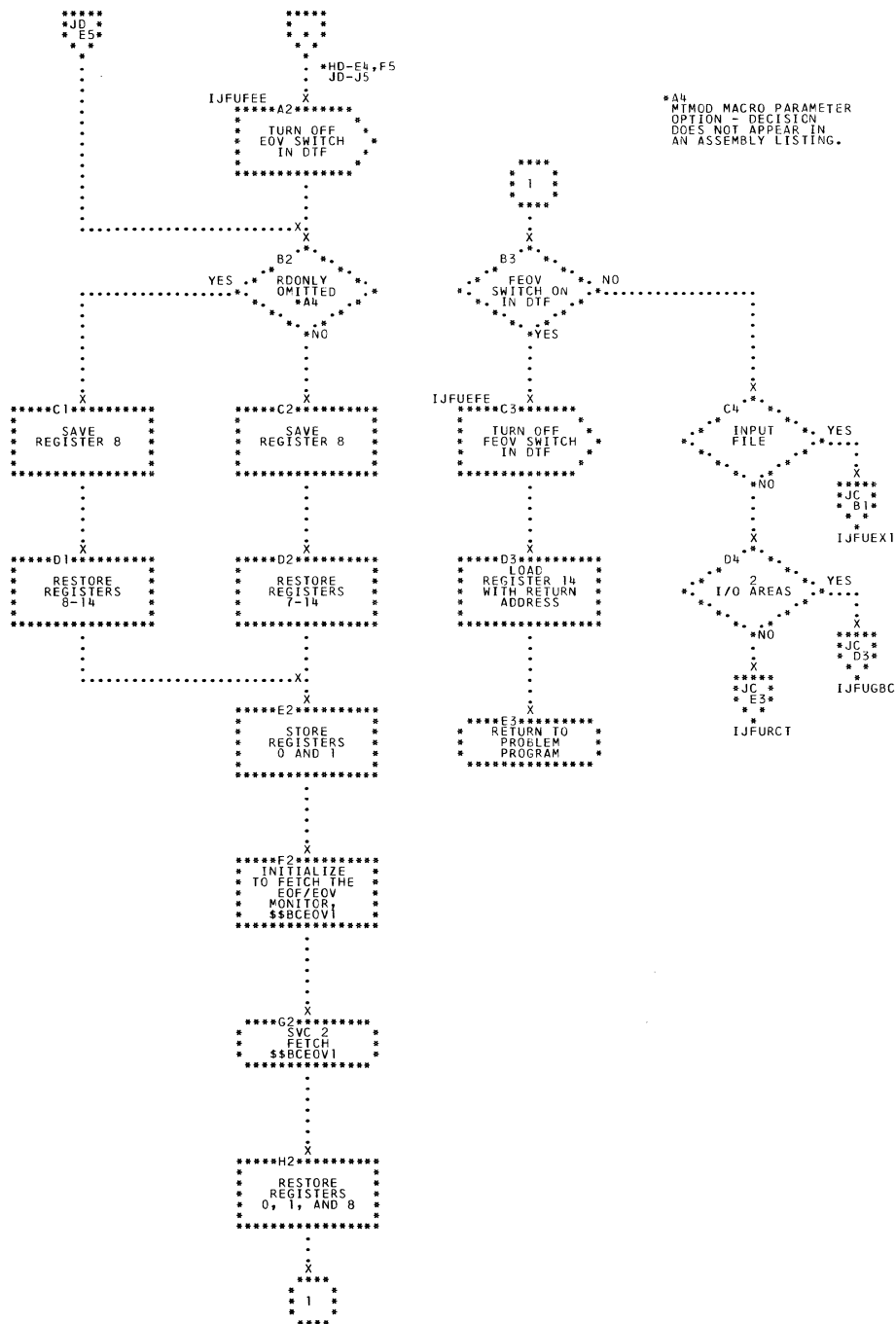
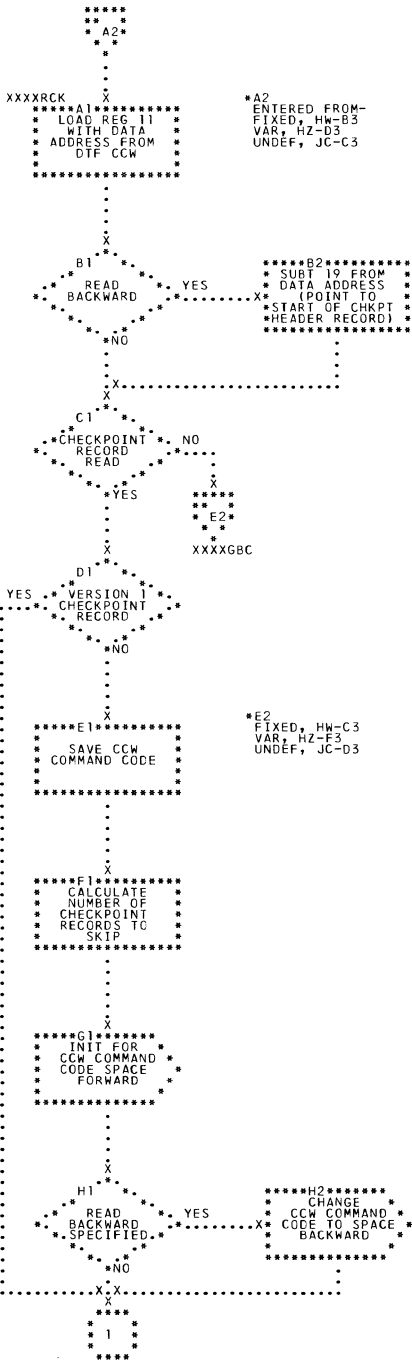


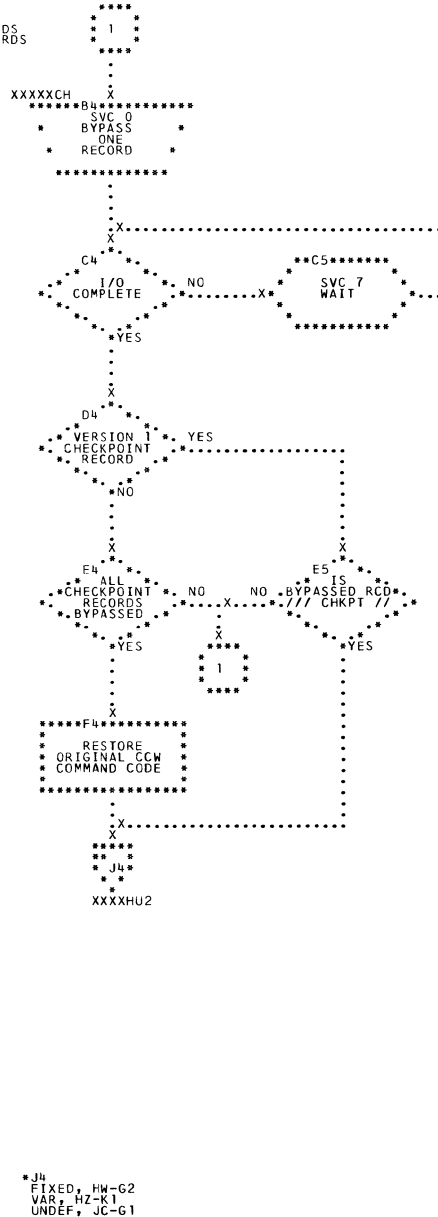
Chart JE. MTMOD: Read/Write Subroutine, Undefined Records (Section 3 of 3)



● Chart JF. MTMOD: Bypass Checkpoint Records Routine



NOTE--
 THIS ROUTINE IS USED
 FOR FIXED, VARIABLE, AND
 UNDEFINED RECORDS. THE
 ONLY DIFFERENCE IN THE
 ROUTINE IS THE LABEL
 PREFIX USED (XXXX)--
 IJFF, FOR FIXED RECORDS
 IJFV, FOR VARIABLE RECORDS
 IJFU, FOR UNDEFINED RECORDS



● Chart JG. MTMOD: Workfile Subroutines

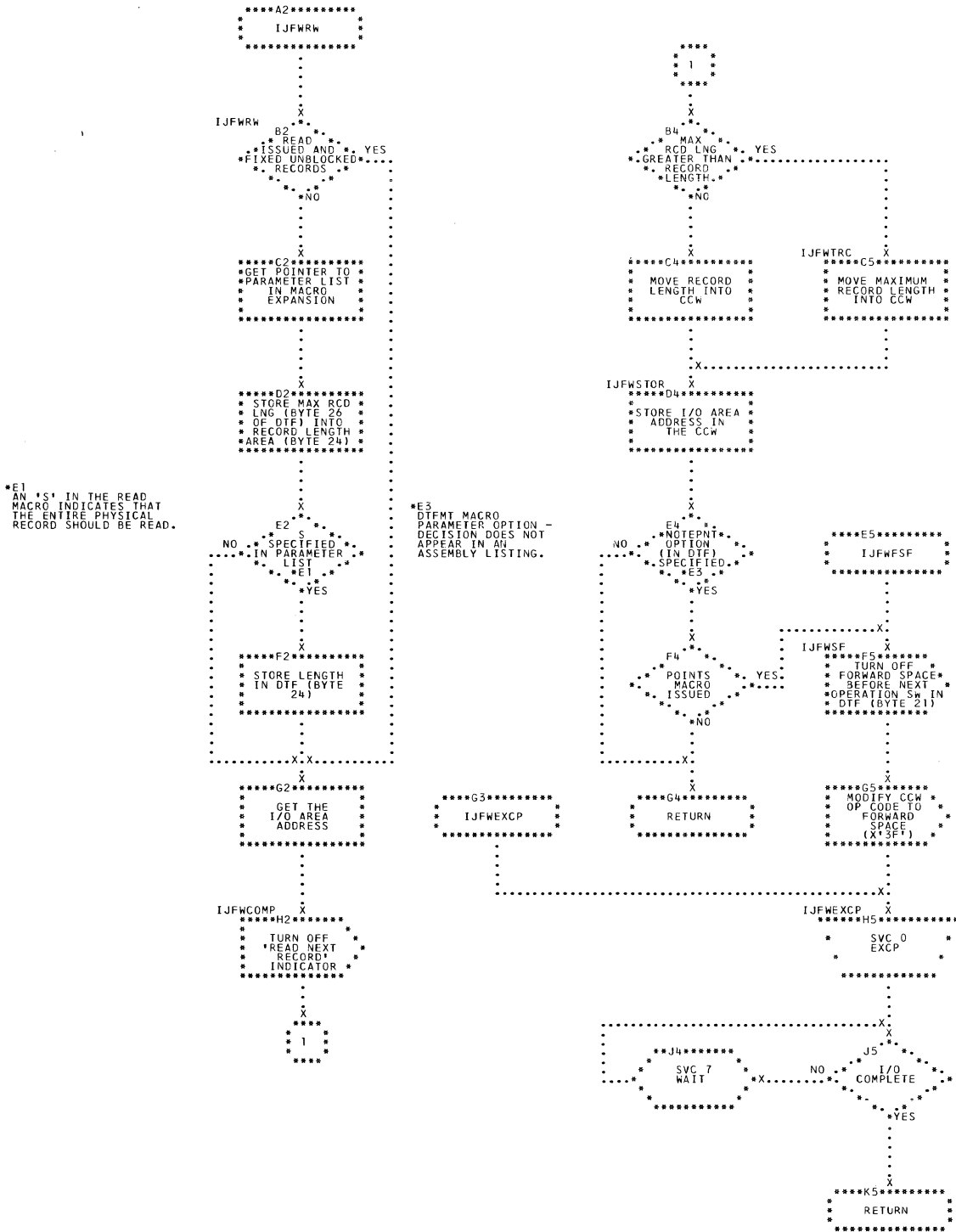


Chart KA. \$\$\$BOMT01: Open Input Standard Labels, Forward (Section 1 of 3)

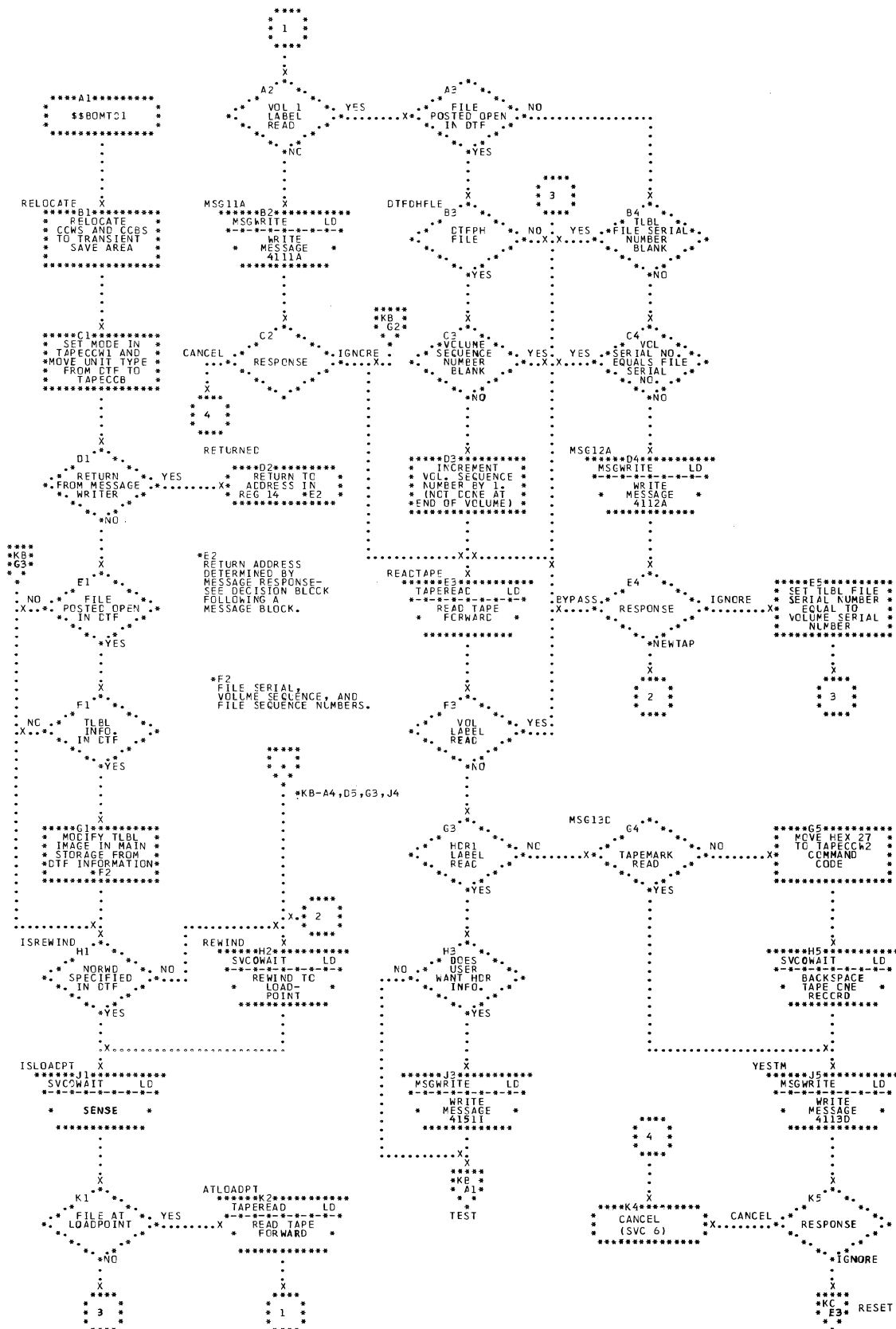


Chart KC. \$\$BOMT01: Open Input Standard Labels, Forward (Section 3 of 3)

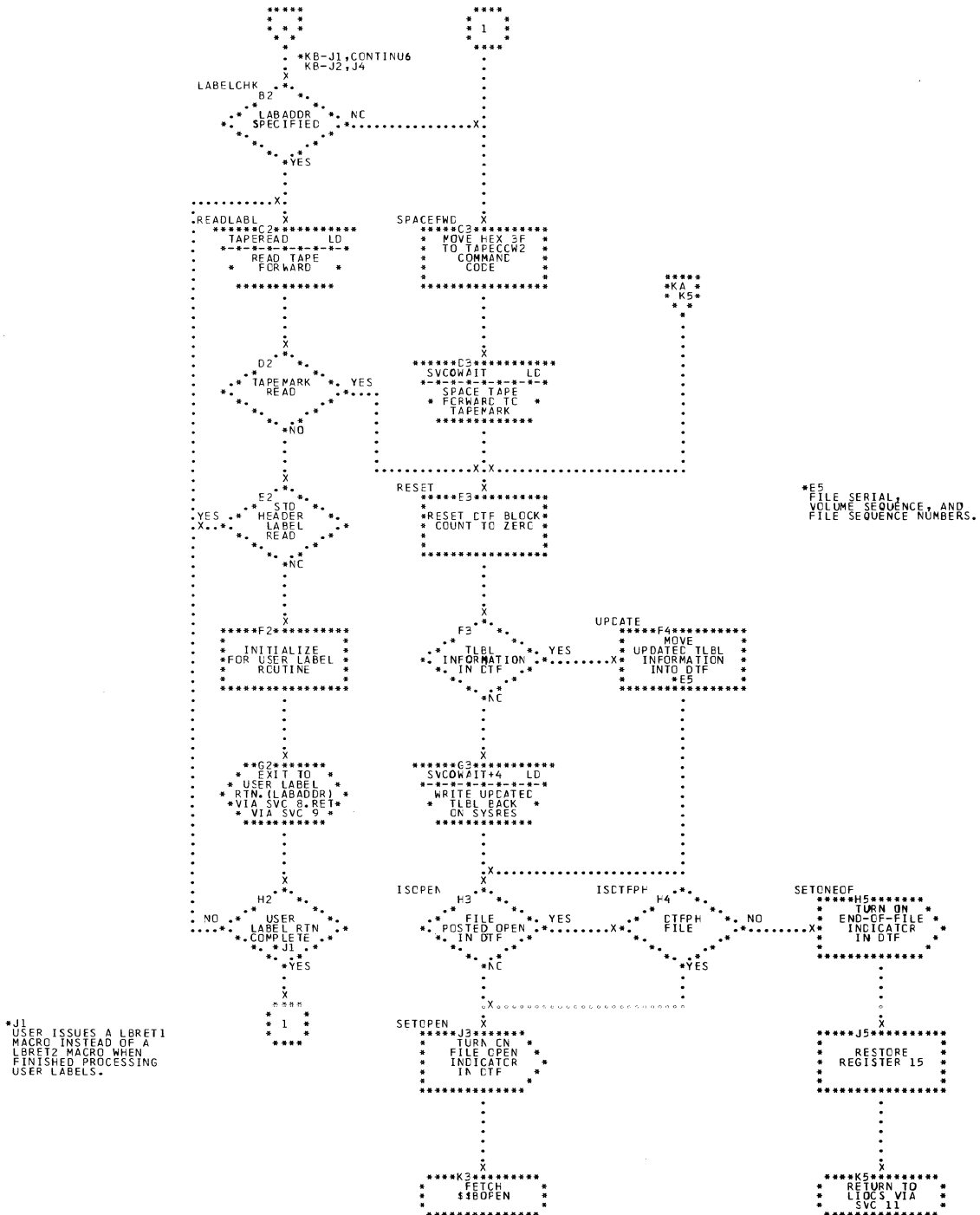


Chart KD. \$\$BOMT02: Open Input Standard Labels, Backward

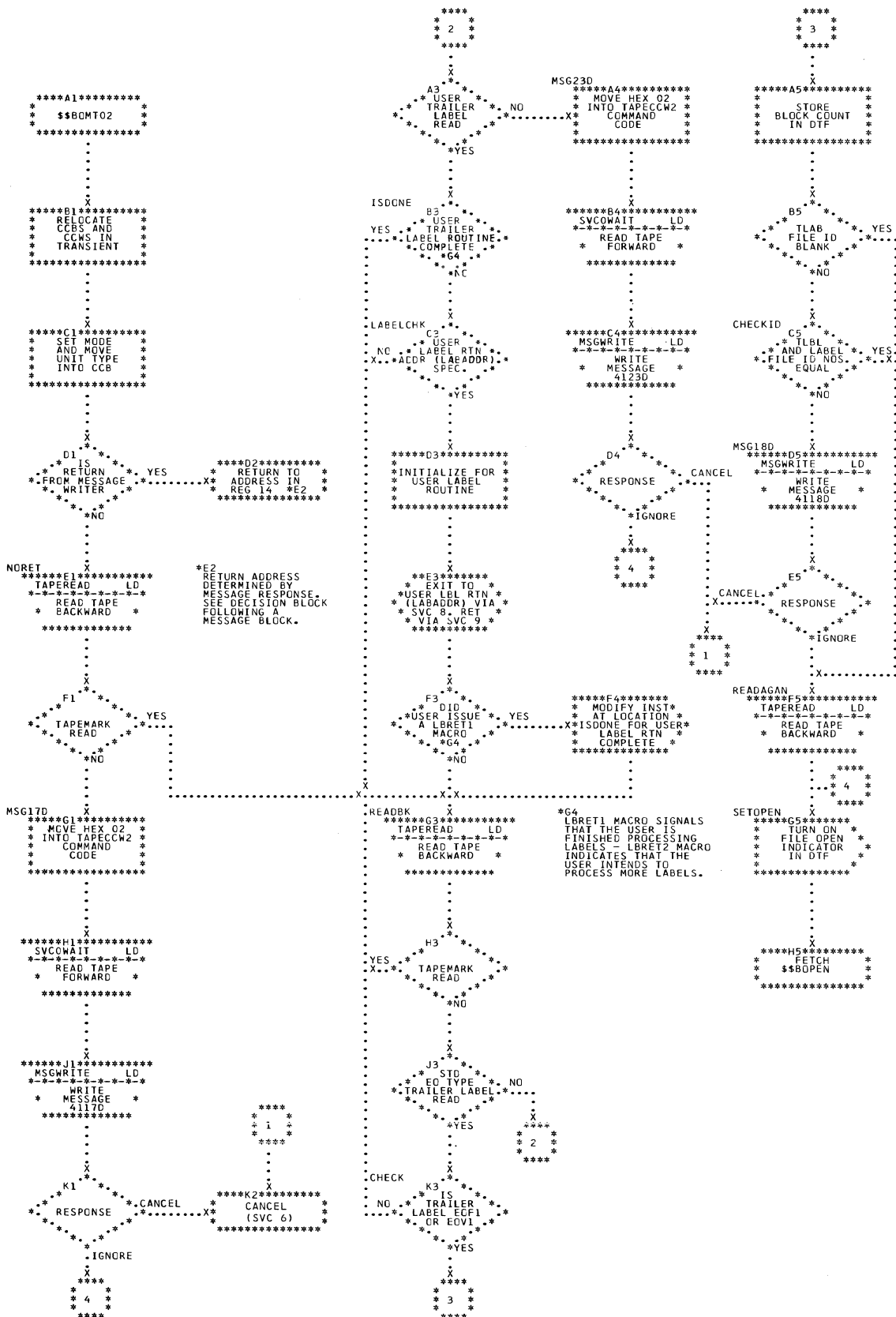


Chart KF. \$\$\$BOMT03: Open Output Standard Labels, Forward, Phase 1 (Section 2 of 2)

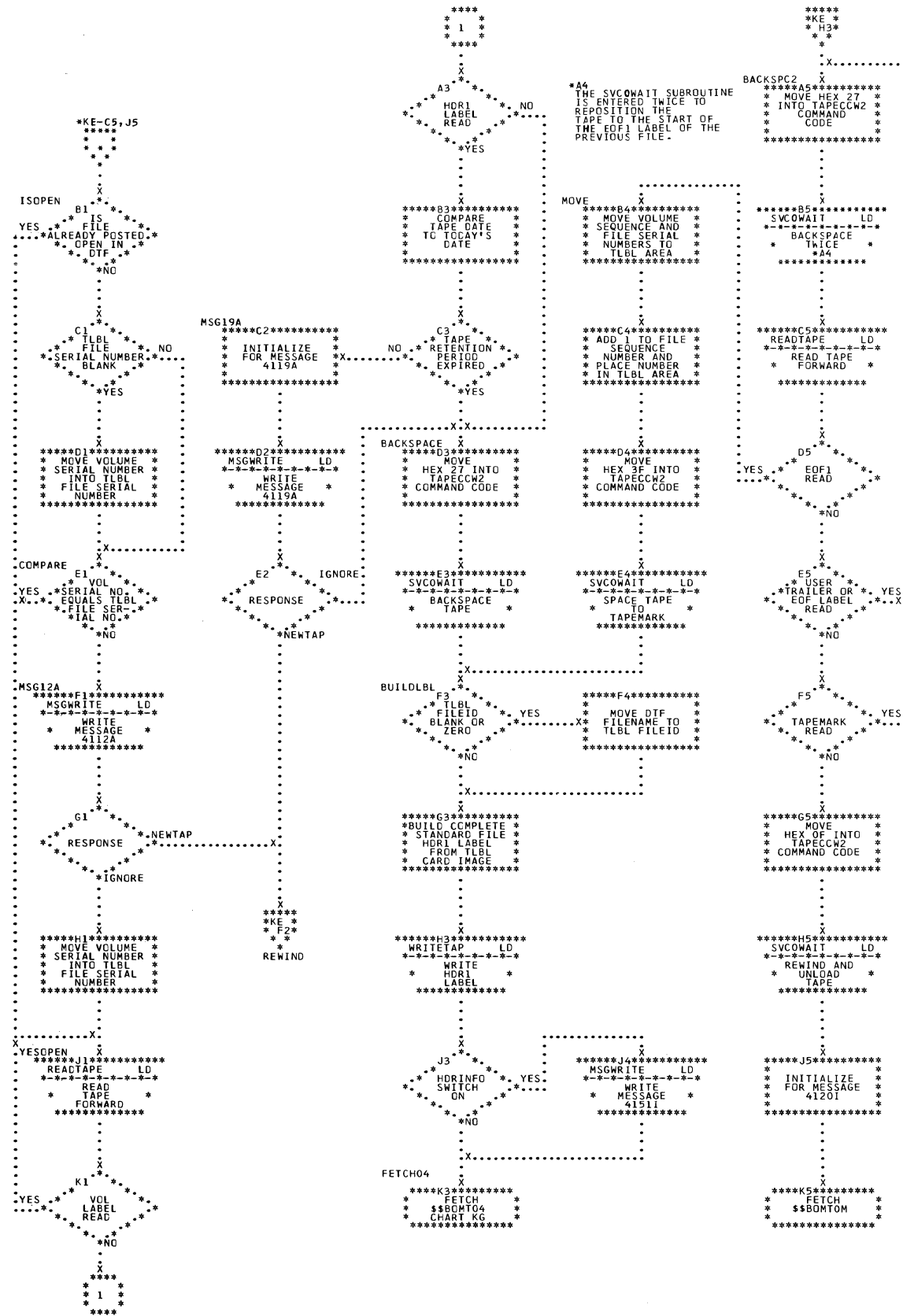


Chart KG. \$\$BOMT04: Open Output Standard Labels, Forward, Phase 2

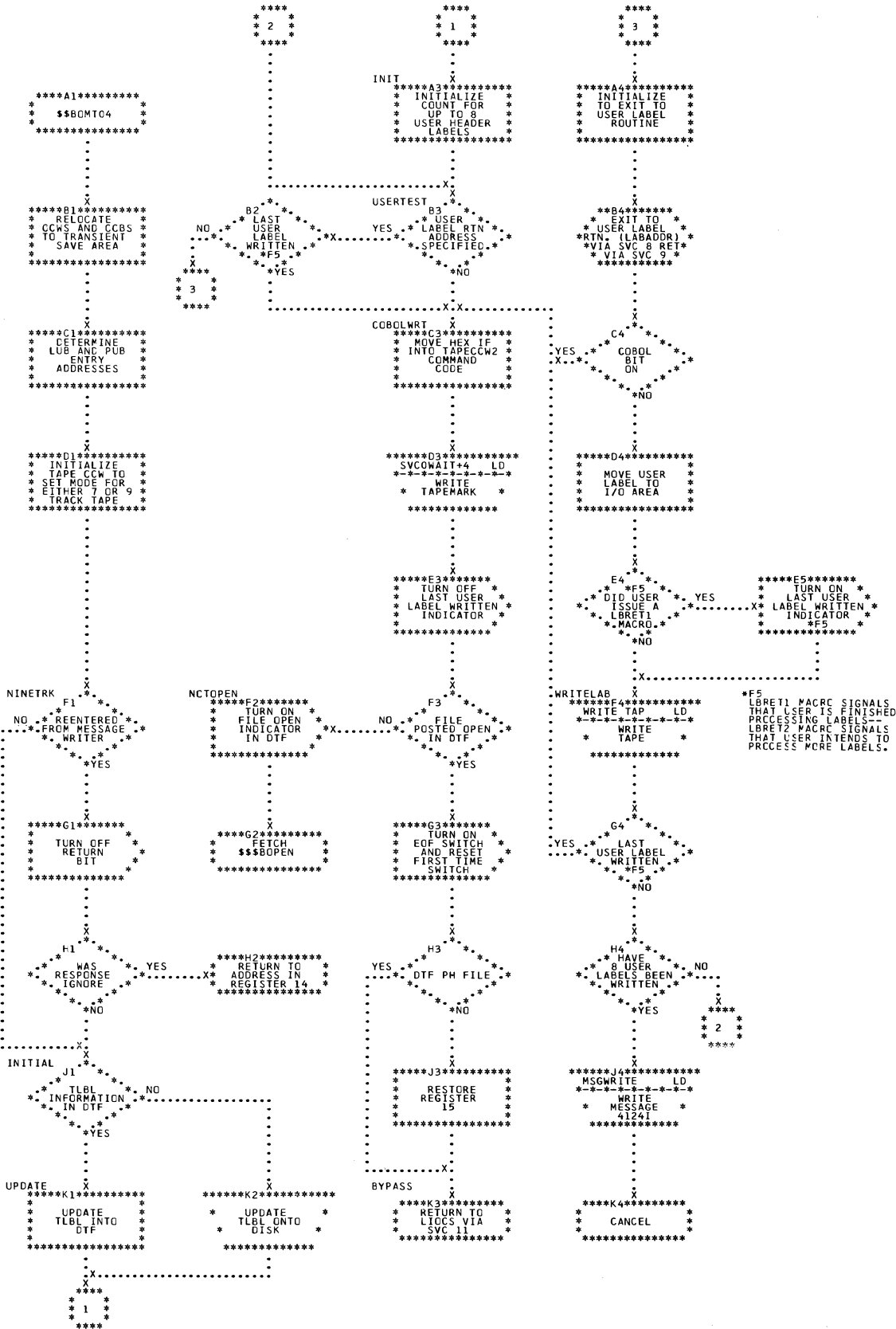


Chart KH. \$\$BOMT05: Open Input or Output, Nonstandard or Unlabeled (Section 1 of 2)

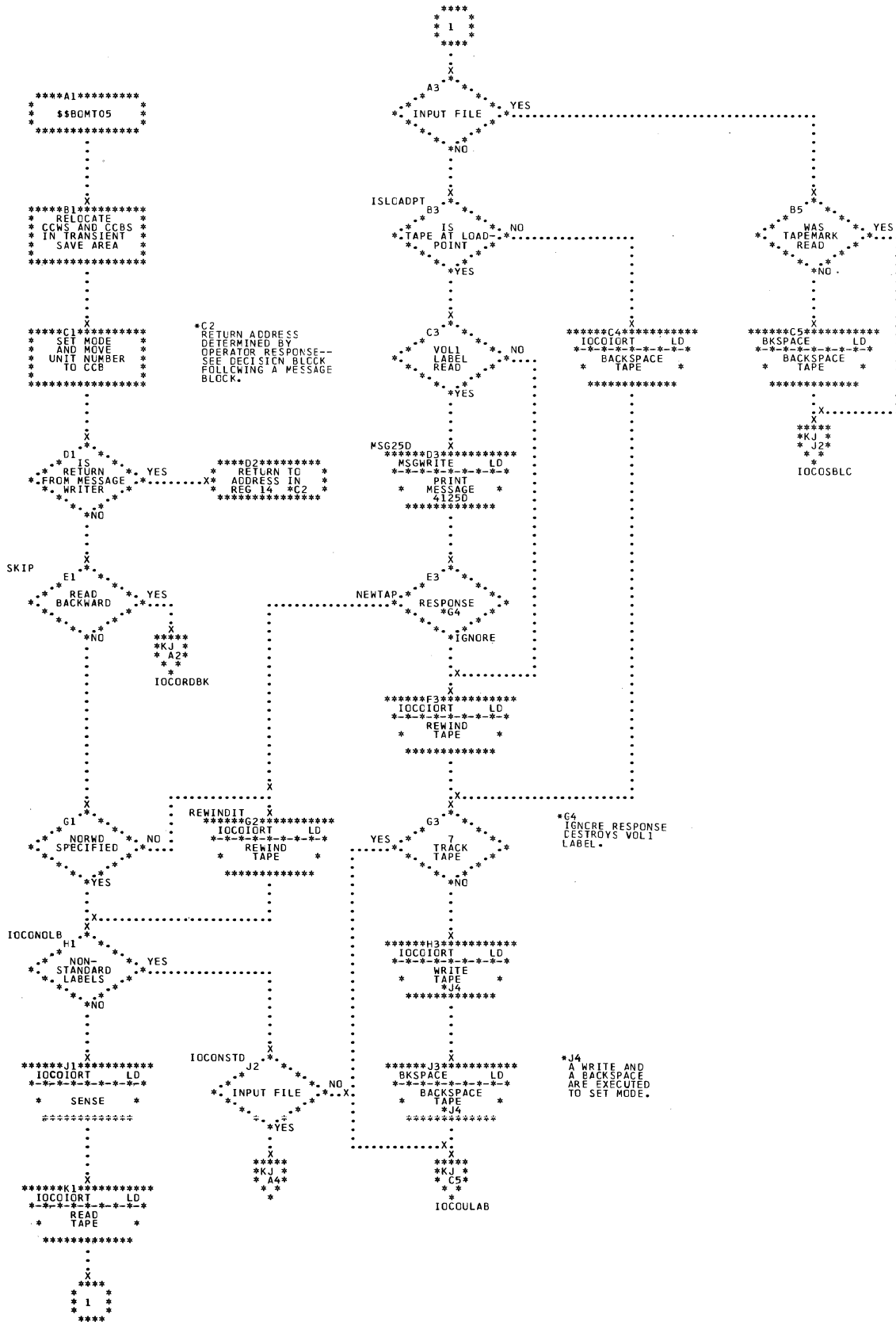


Chart KJ. \$\$BOMT05: Open Input or Output, Nonstandard or Unlabeled (Section 2 of 2)

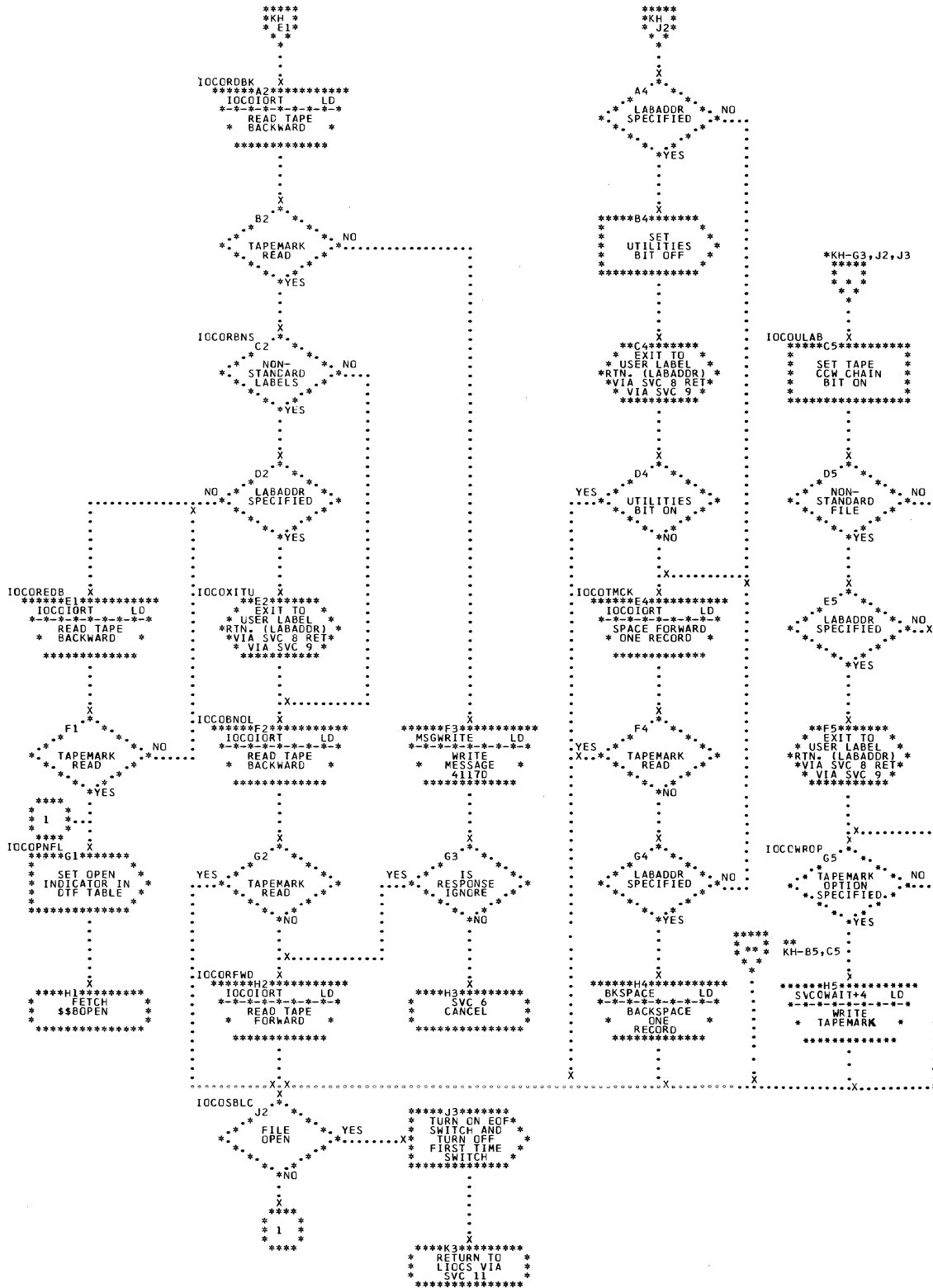


Chart KK. \$\$BOMT06: Open Workfiles, Magnetic Tape (Part 1 of 2)

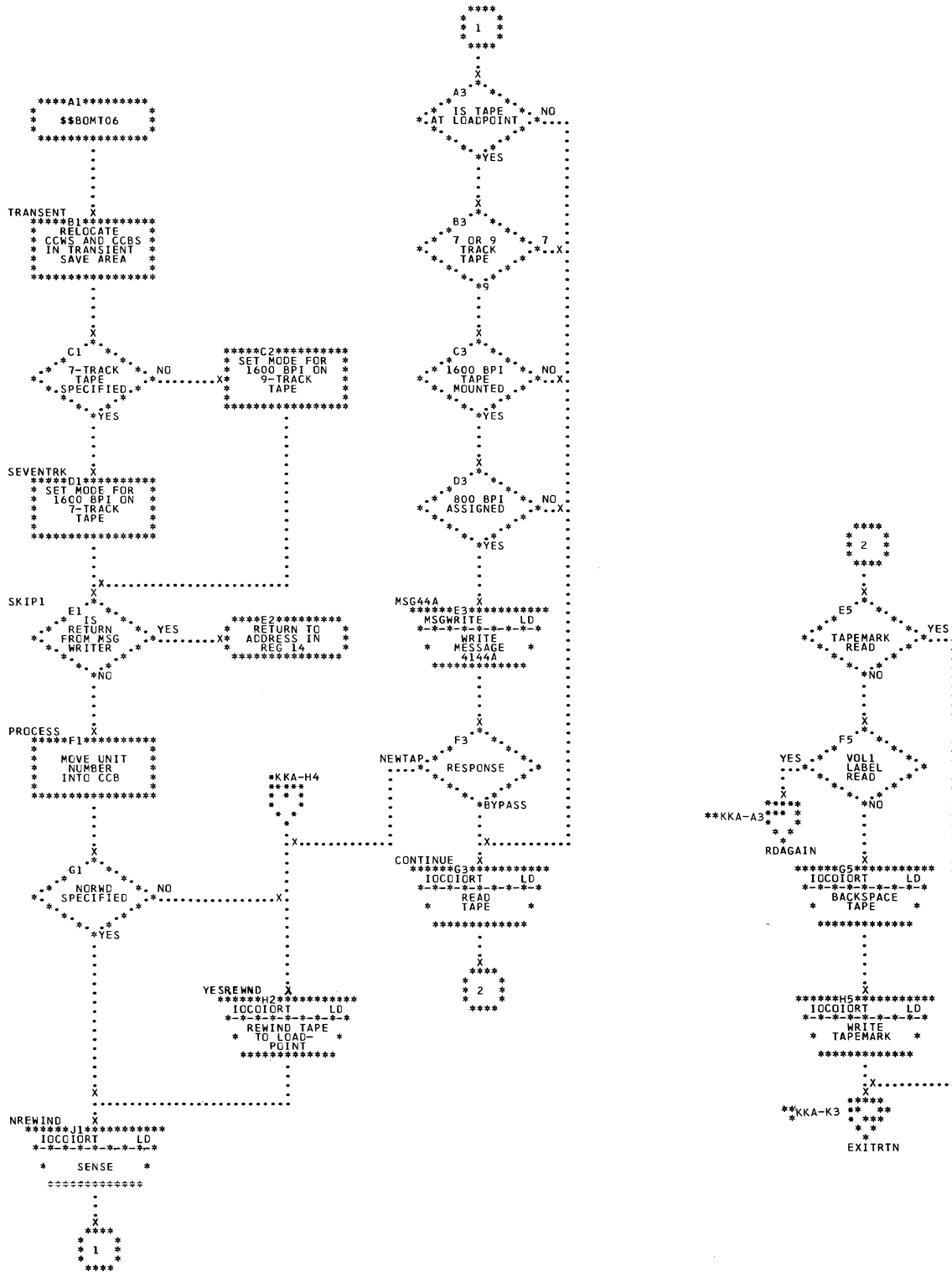


Chart KKA. \$\$BOMT06: Open Workfiles, Magnetic Tape (Part 2 of 2)

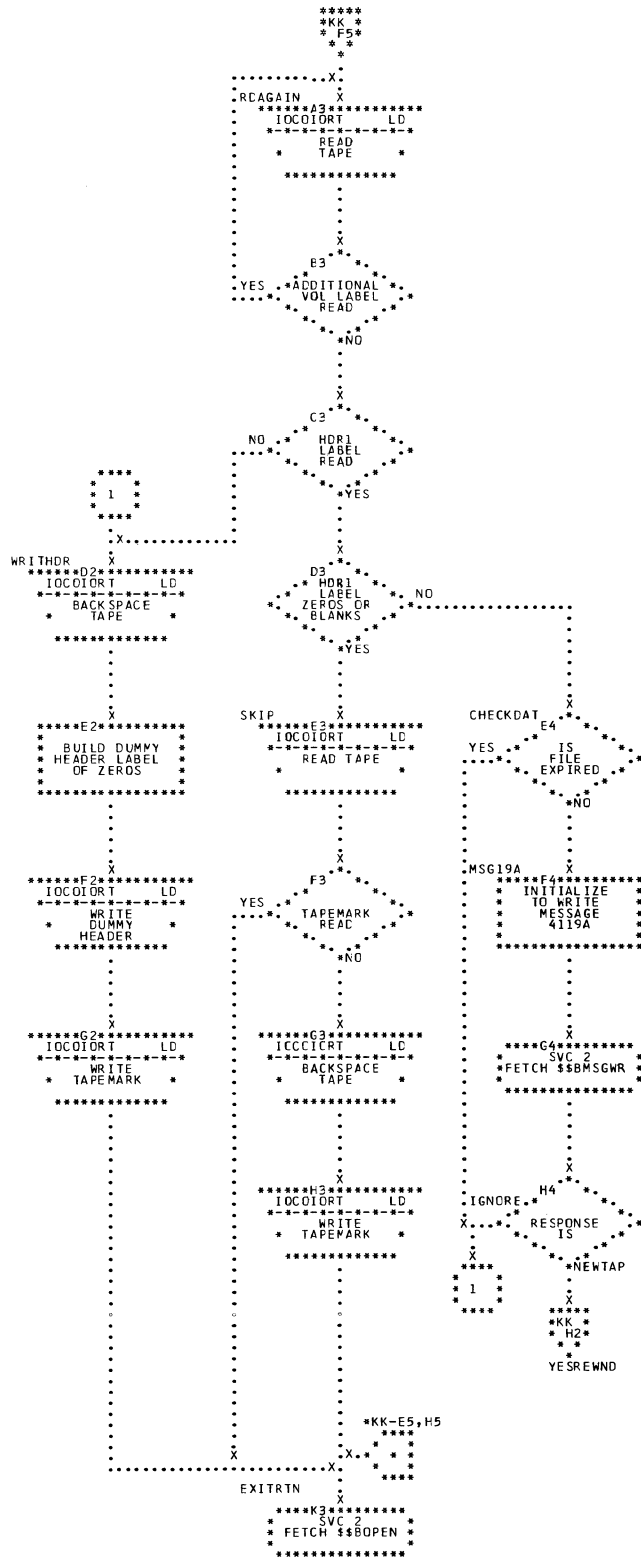


Chart KL. \$\$BJCOPT: Job Control Tape Open Routine (Section 1 of 3)

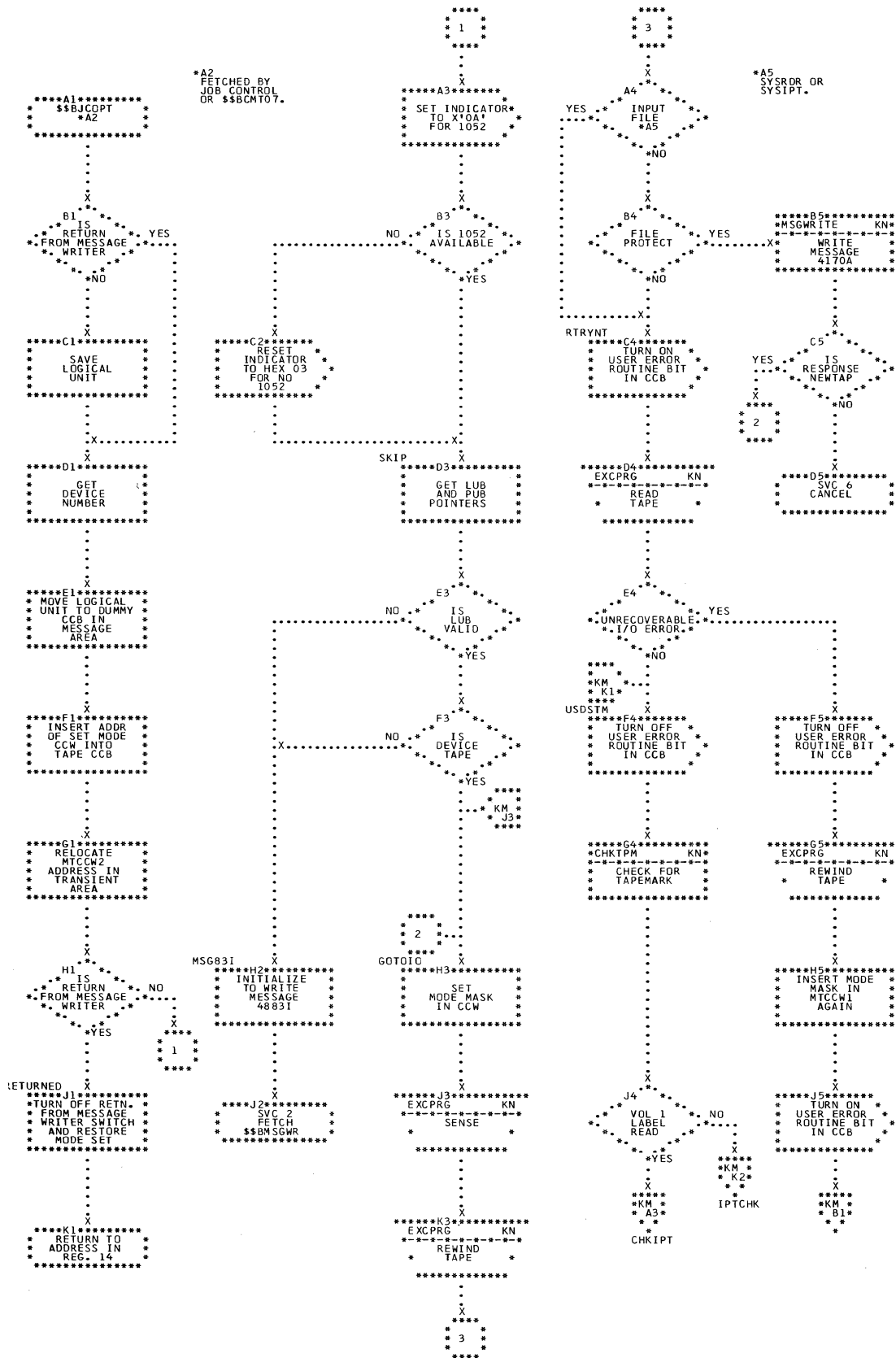


Chart KM. \$\$\$BJCOPT: Job Control Tape Open Routine (Section 2 of 3)

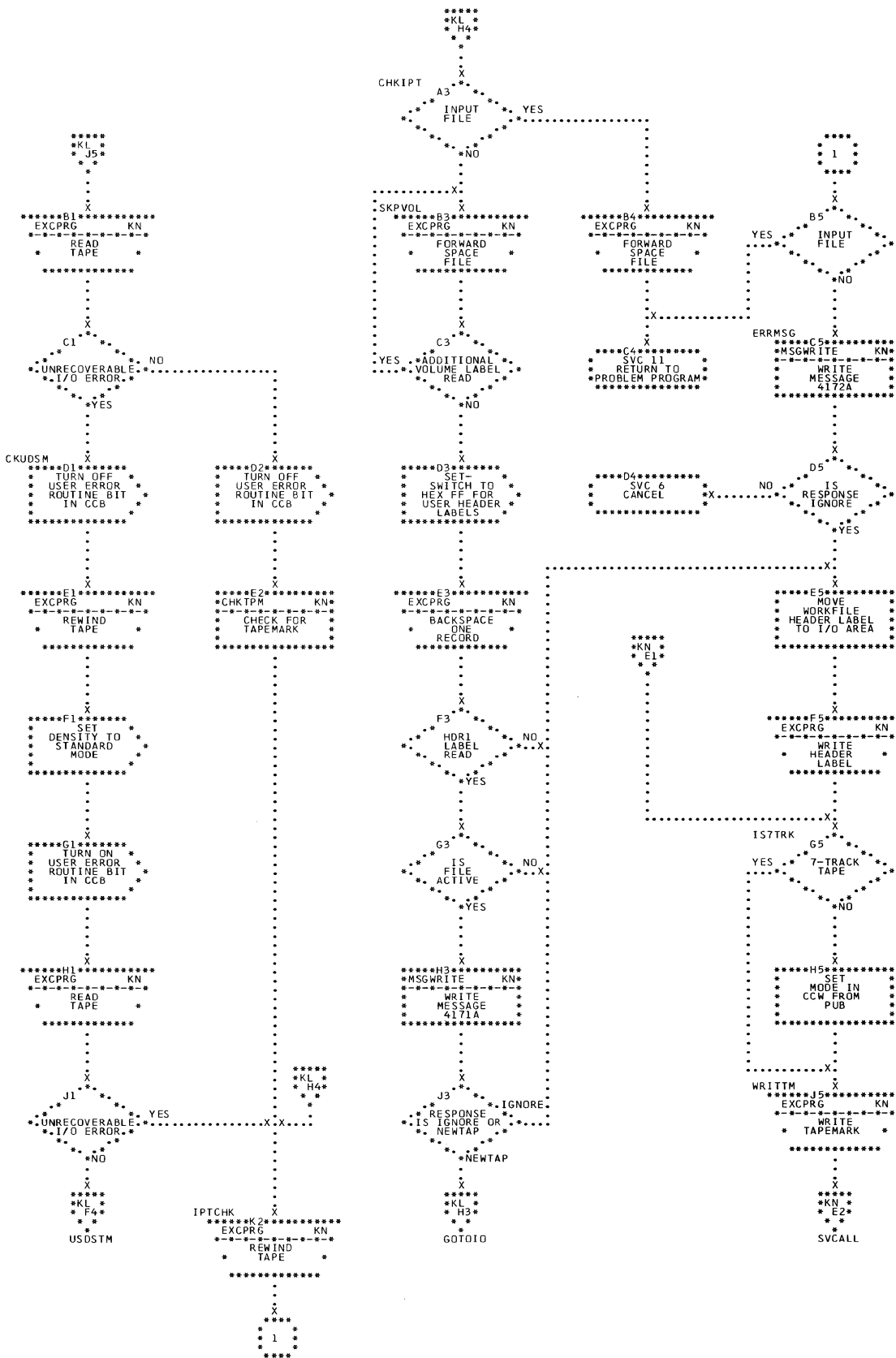


Chart KN. §§BJCOPT: Job Control Tape Open Routine (Section 3 of 3)

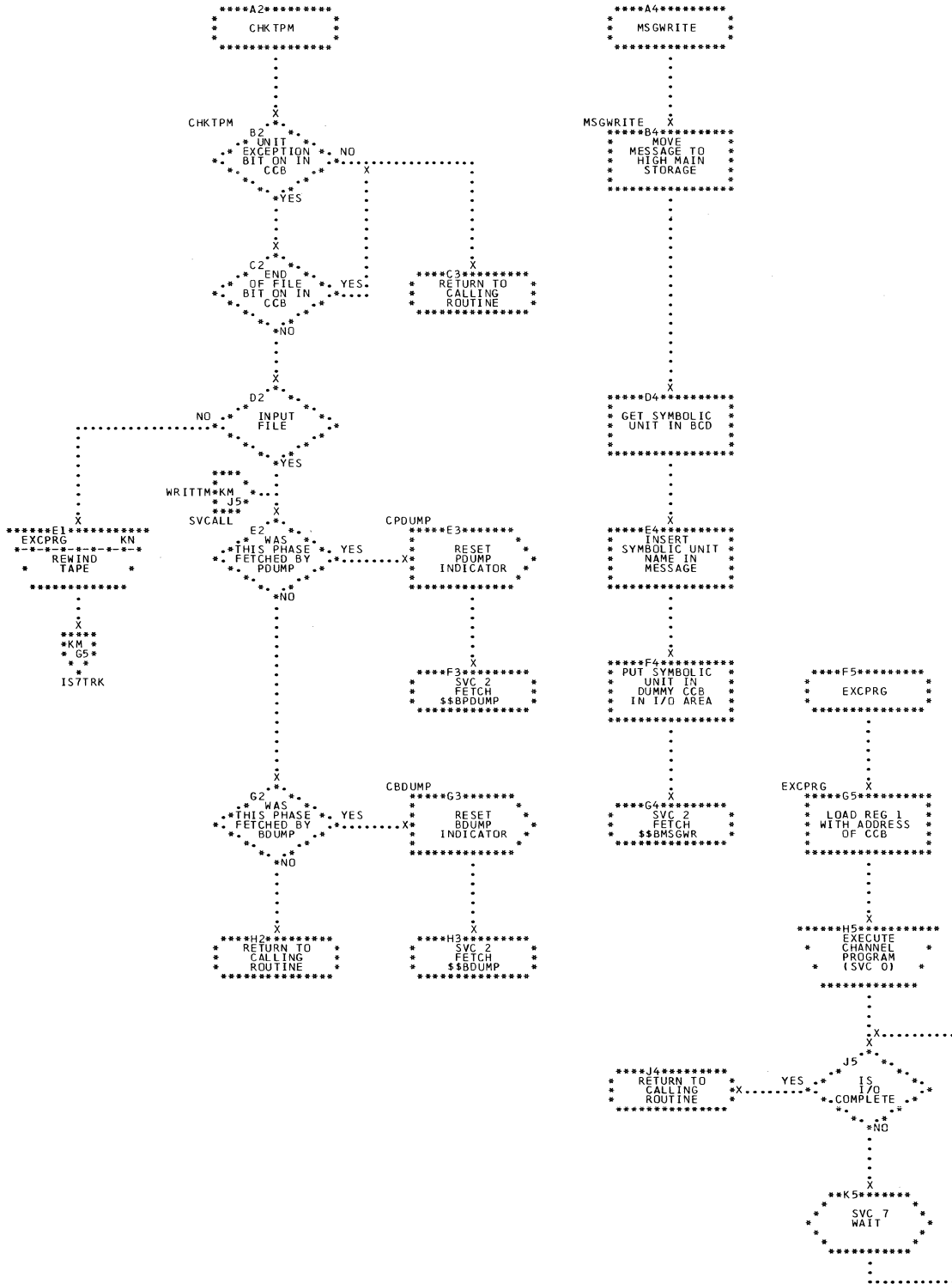


Chart KP. \$\$BCEOV1: EOF/EOV Monitor (Section 1 of 2)

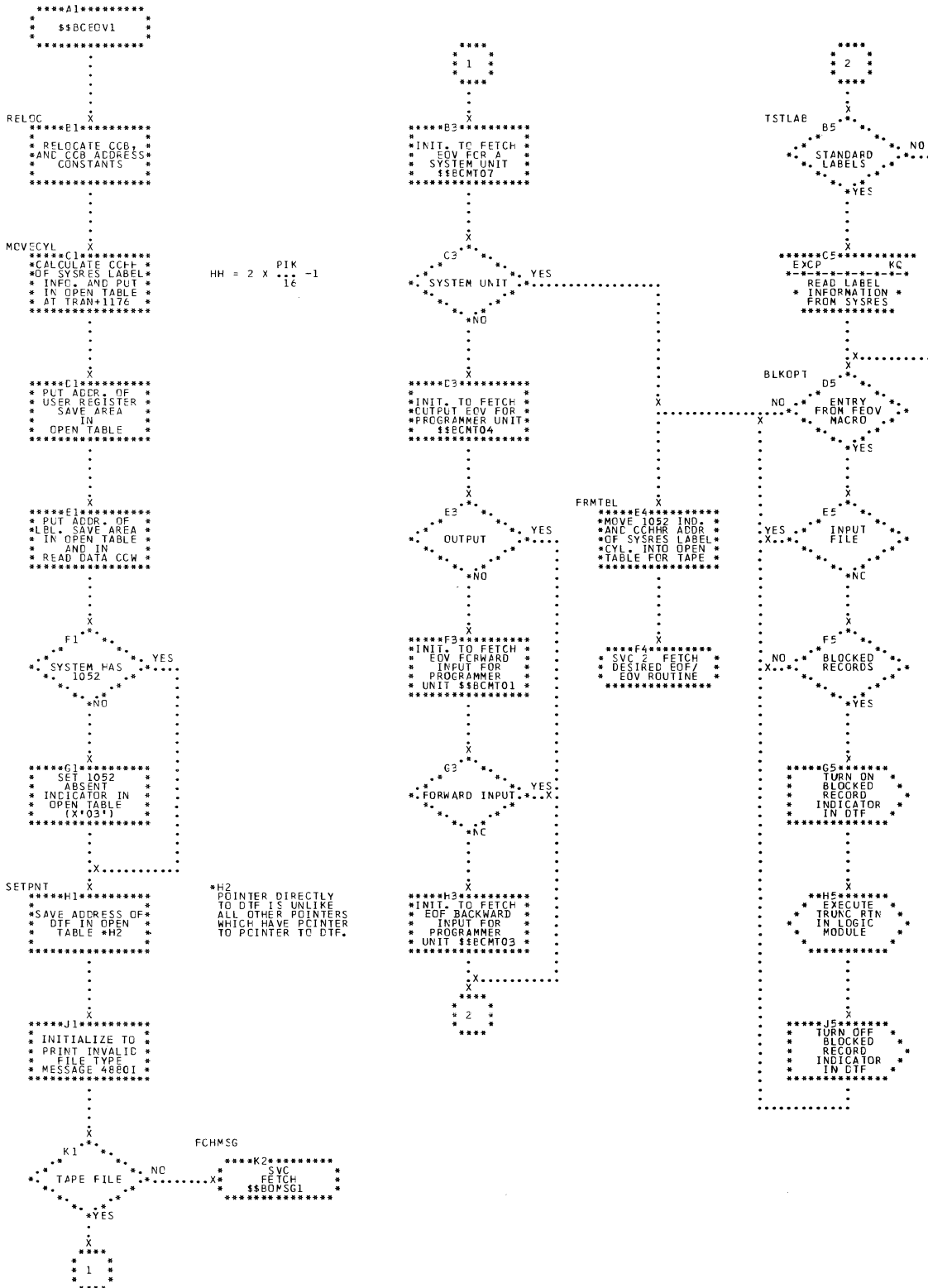


Chart KQ. \$\$BCEOV1: EOF/EOV Monitor (Section 2 of 2)

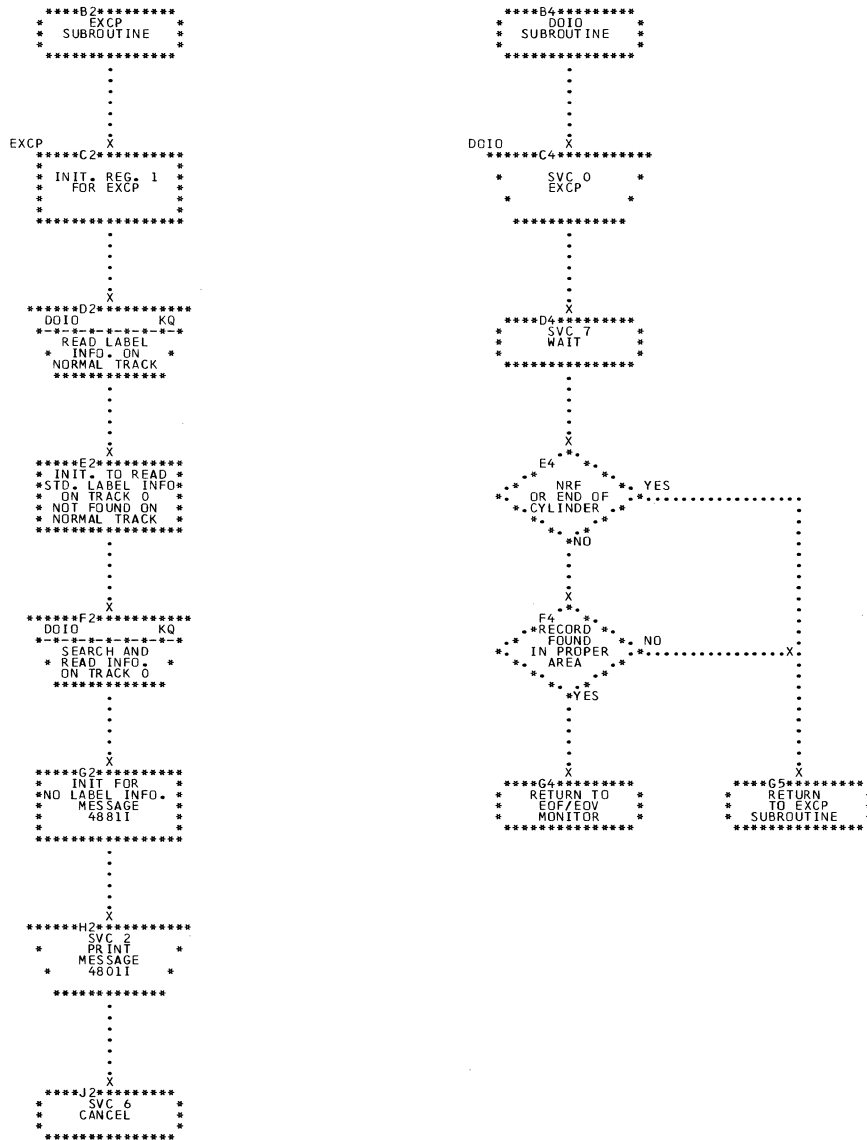
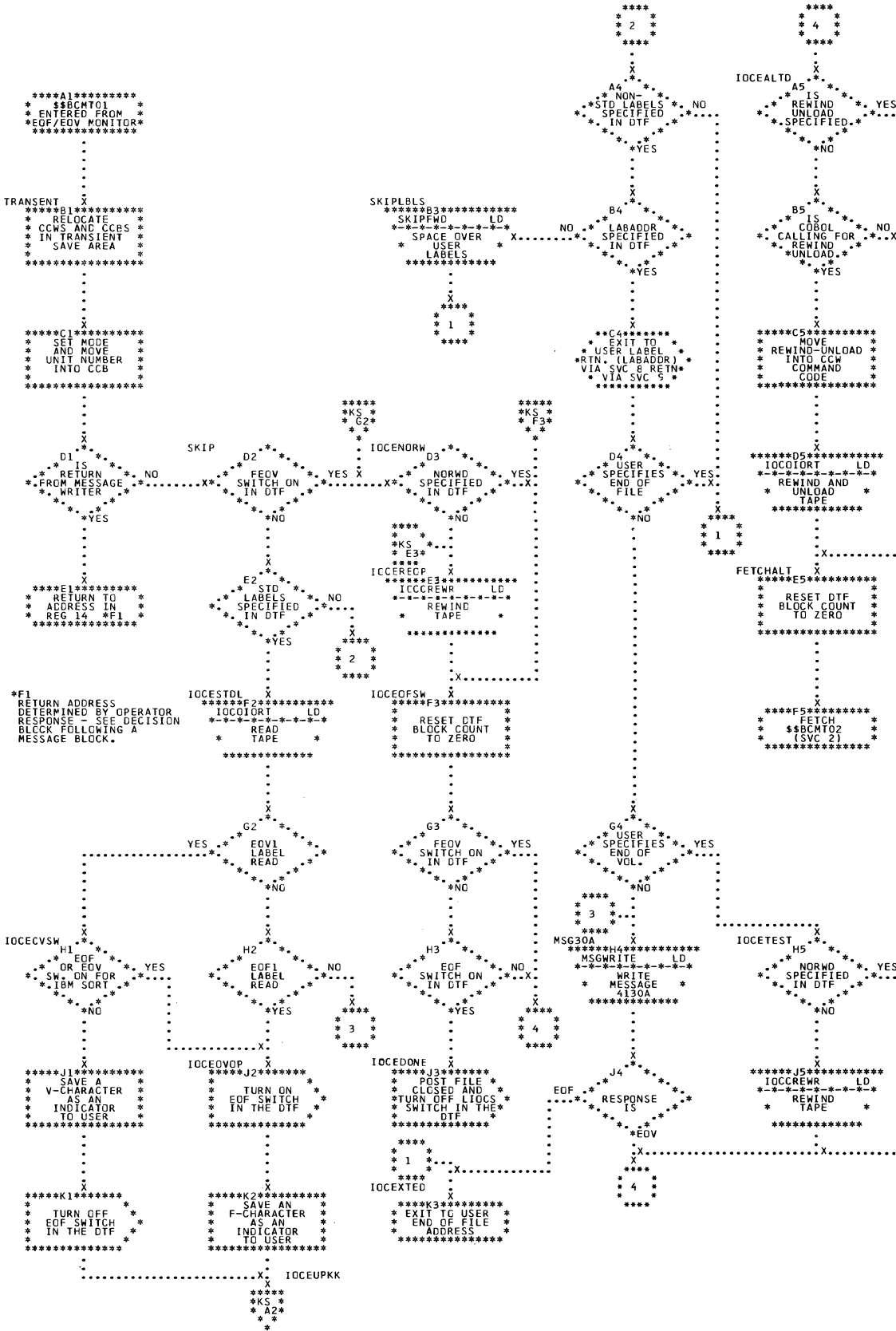


Chart KR. \$\$BCMT01: Tape Close, EOF/EOV Input Forward, Phase 1 (Section 1 of 2)



● Chart KT. \$\$BCMT02: Tape Close, Alternate Switching for EOJ (Section 1 of 2)

*A1
ENTERED FROM
\$\$BCMT01 OR
\$\$BCMT04.

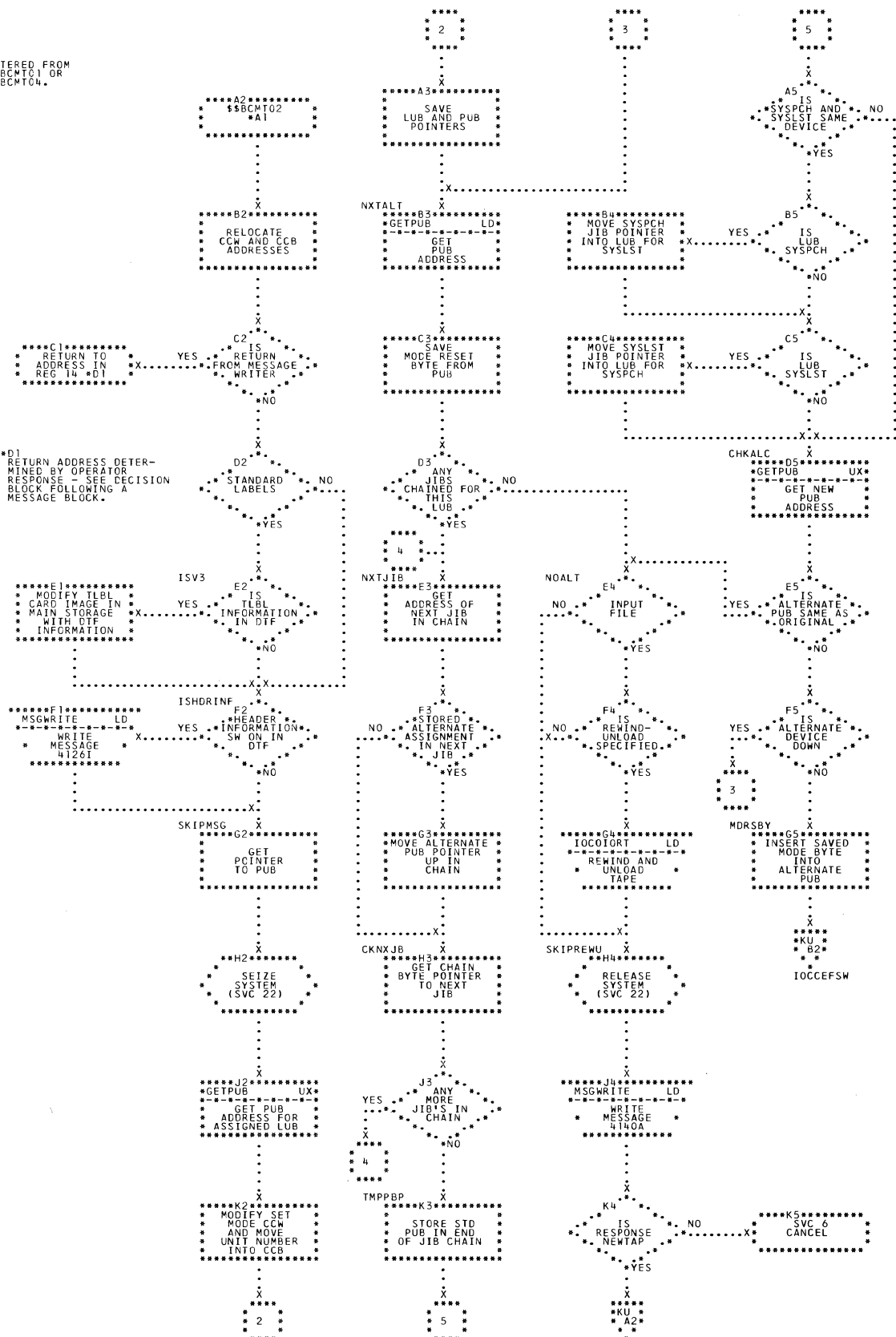
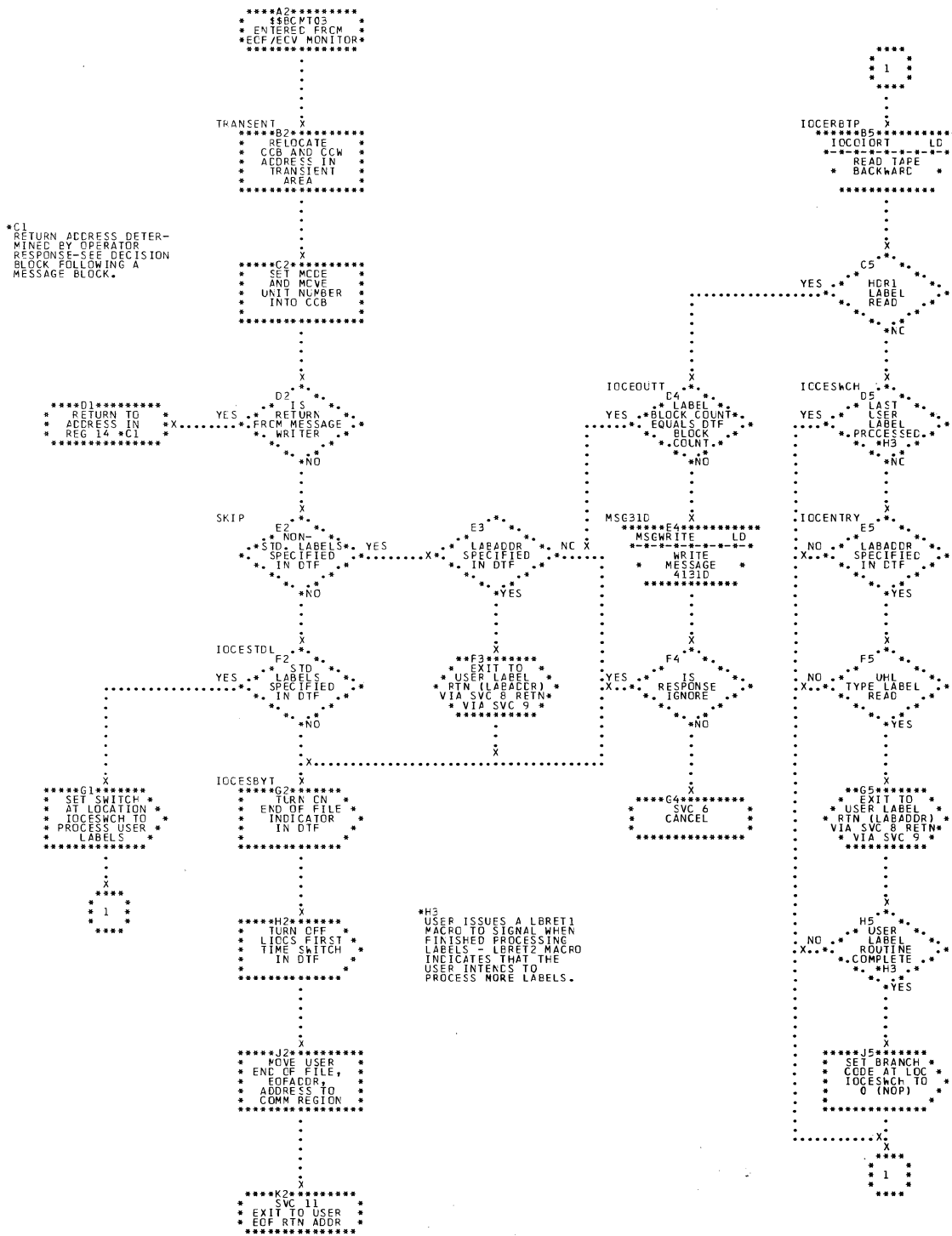
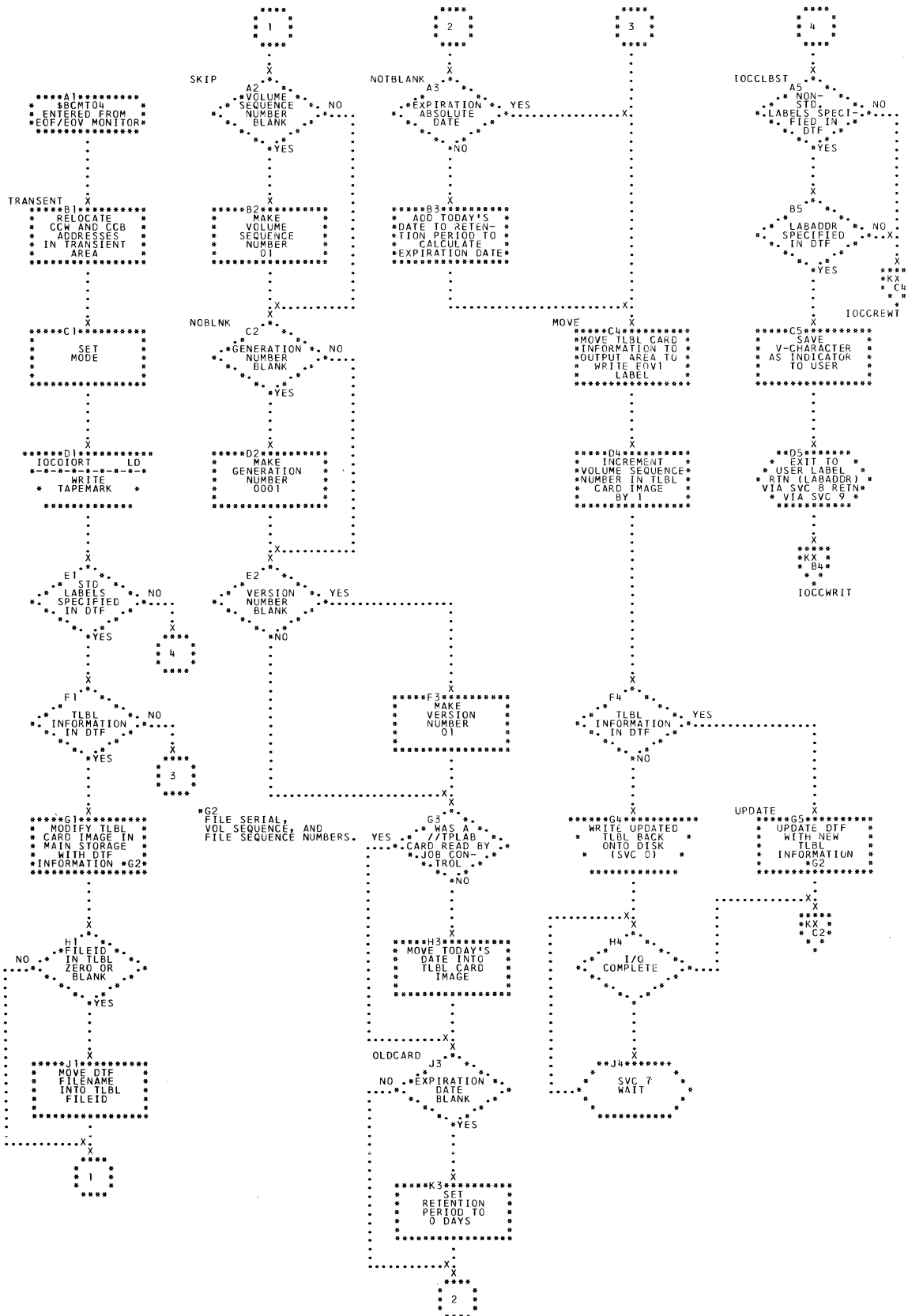


Chart KV. \$\$BCMT03: Tape Close, EOF Input Backward



● Chart KW. \$\$\$BCMT04: Tape Close, EOVS Output Forward (Section 1 of 2)



● Chart KX. \$\$BCMT04: Tape Close, EOY Output Forward (Section 2 of 2)

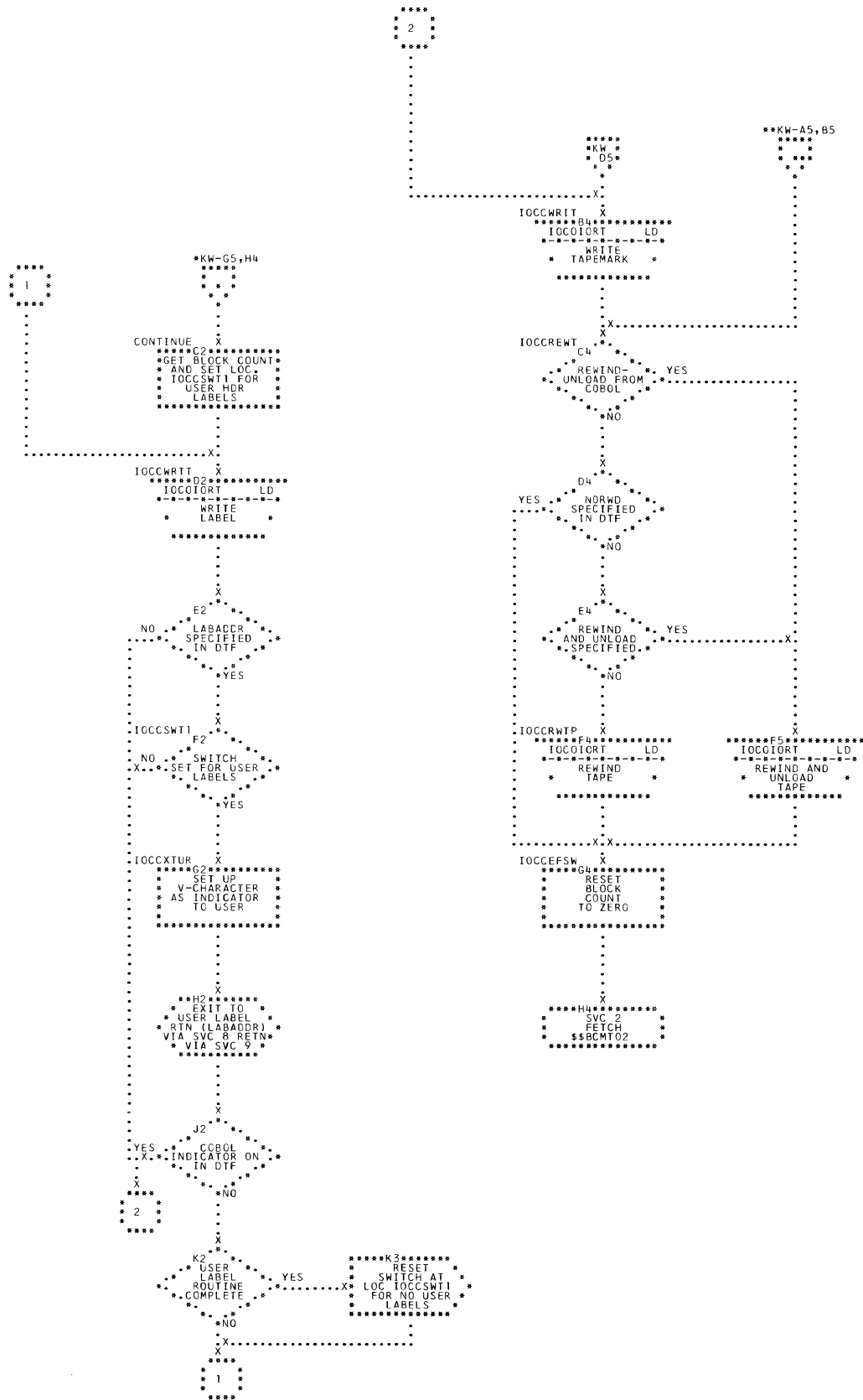


Chart KY. \$\$BCMT05: Close Standard, Nonstandard, and Unlabeled Files, All Types Except Workfiles (Section 1 of 2)

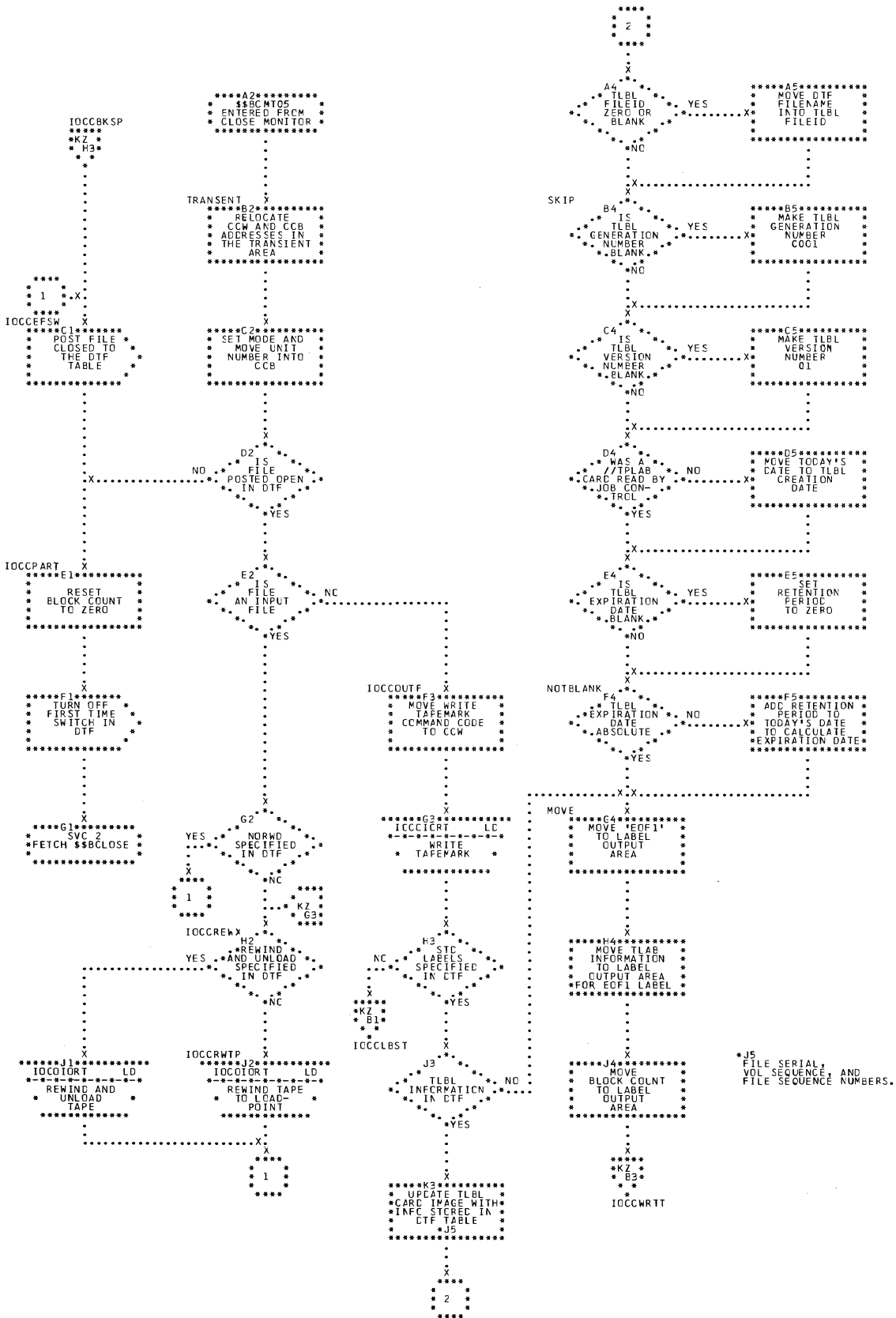


Chart KZ. \$\$BCMT05: Close Standard, Nonstandard, and Unlabeled Files, All Types Except Workfiles (Section 2 of 2)

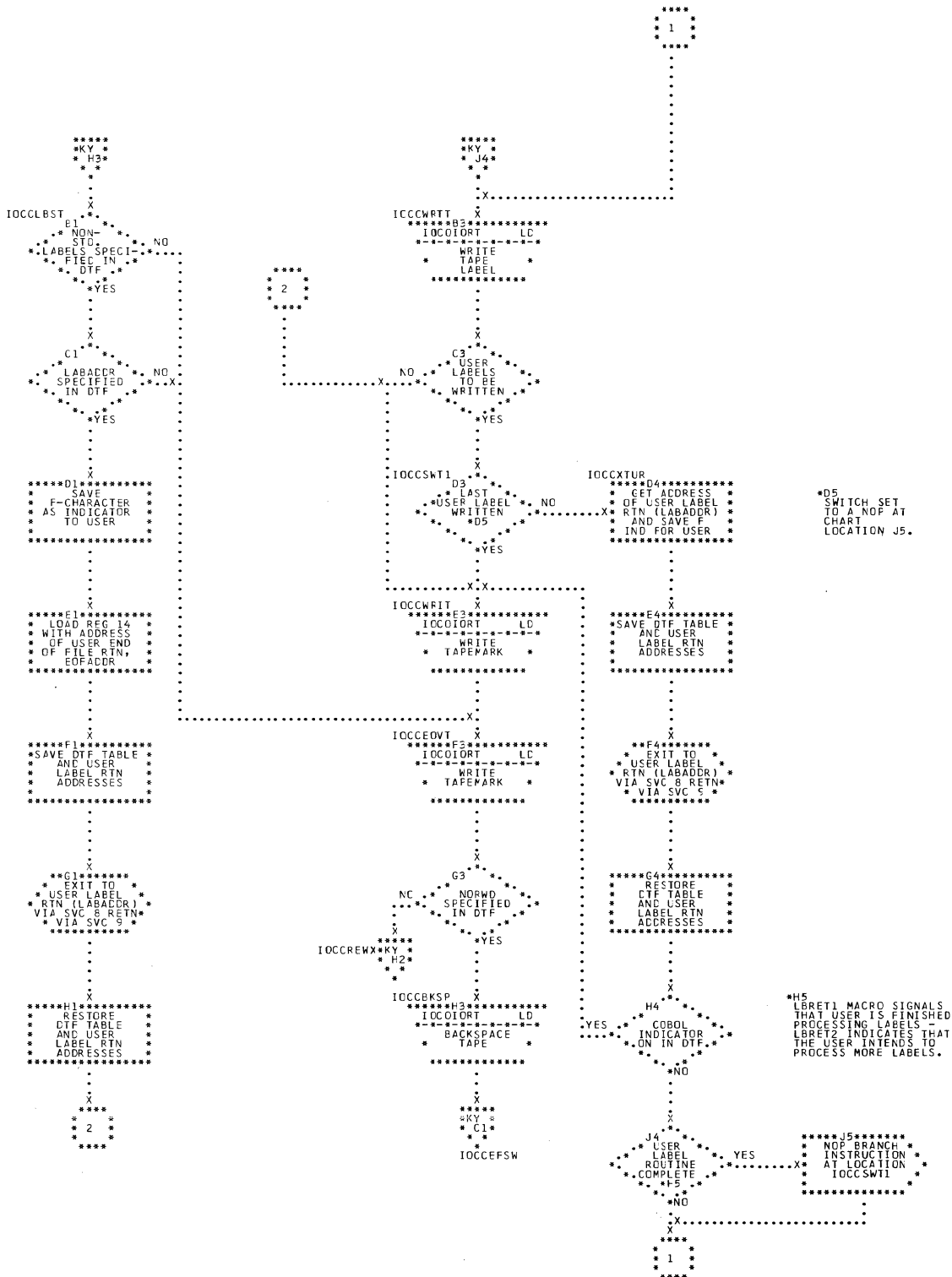


Chart LA. \$\$BCMT06: Close Tape Workfiles

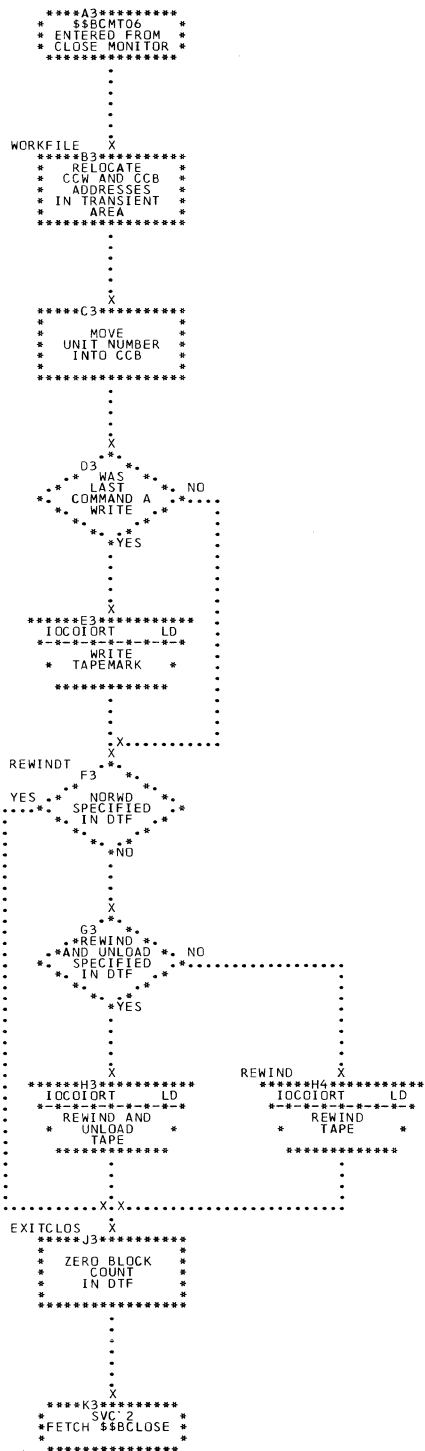


Chart LD. Tape Open/Close Subroutines

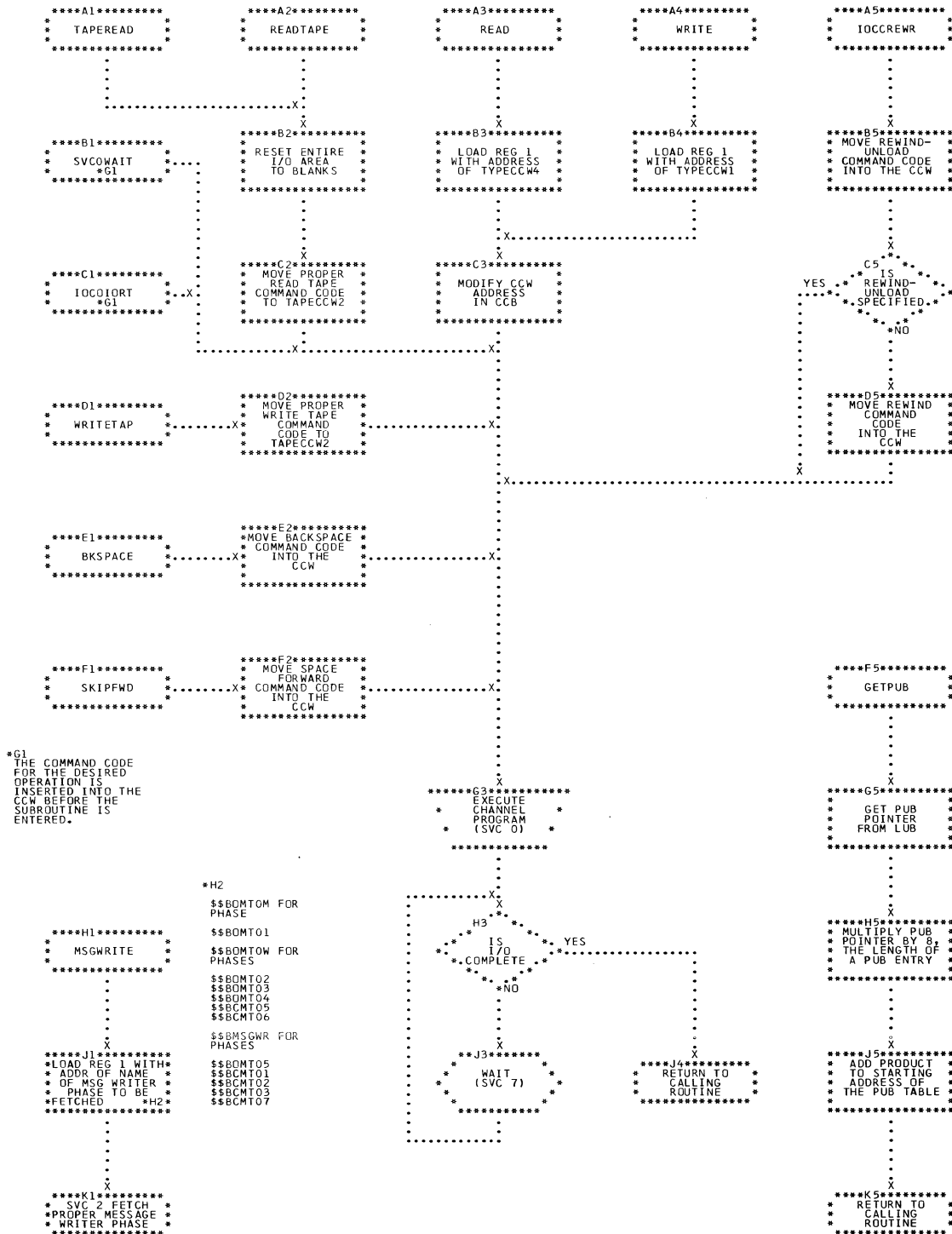


Chart LE. \$\$BOMTOM and \$\$BOMTOW: Tape Open Message Writers

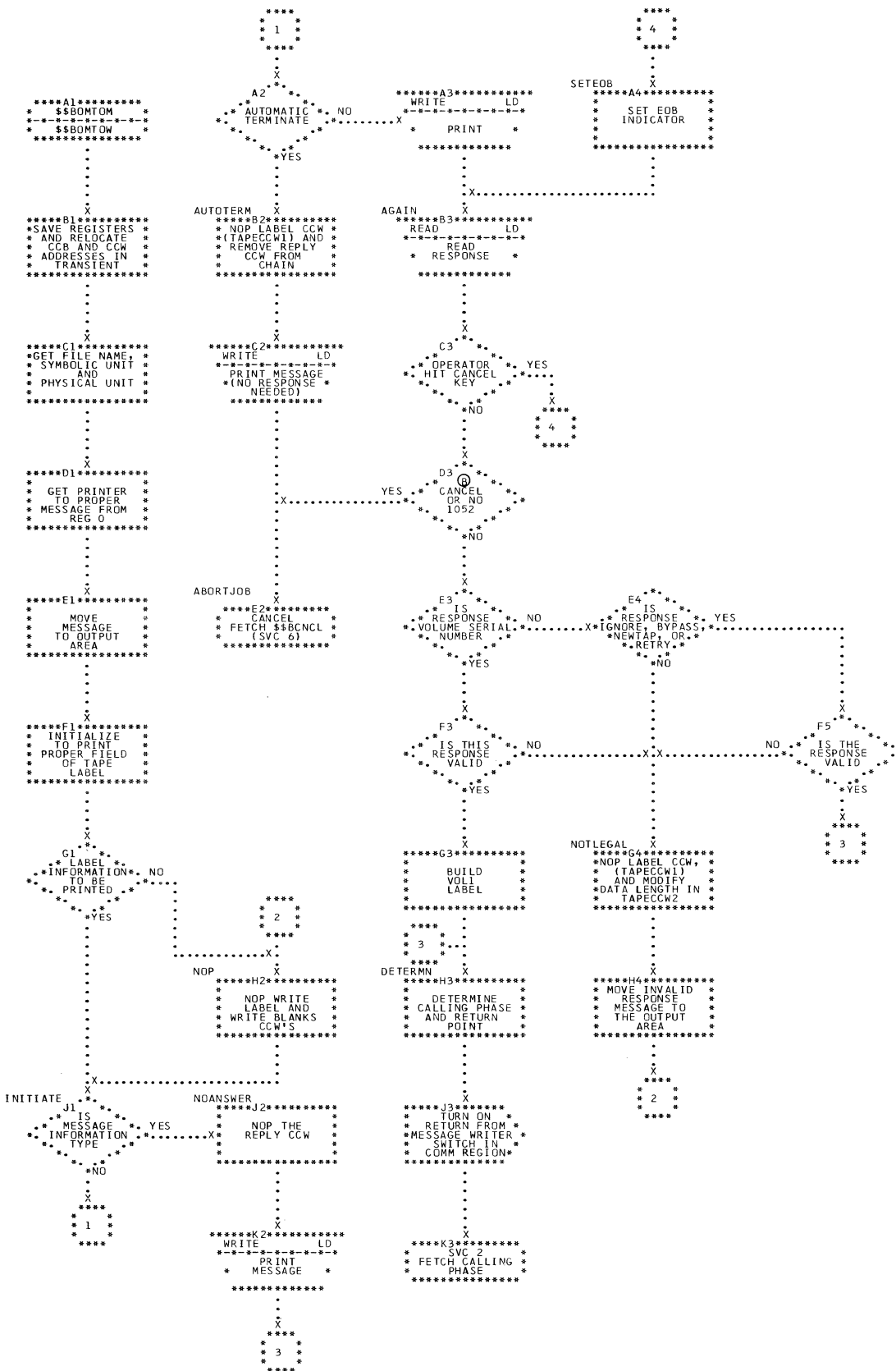
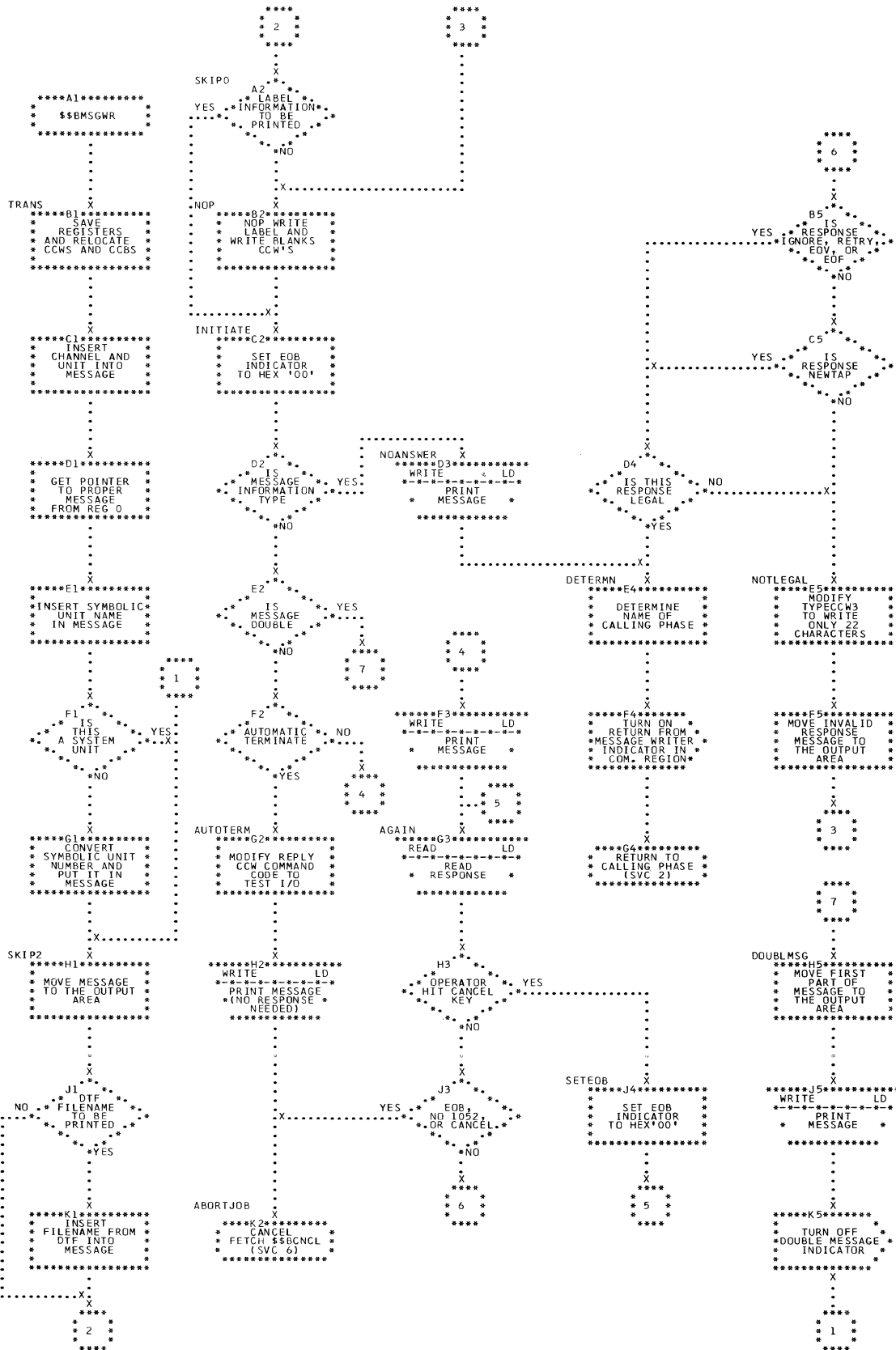
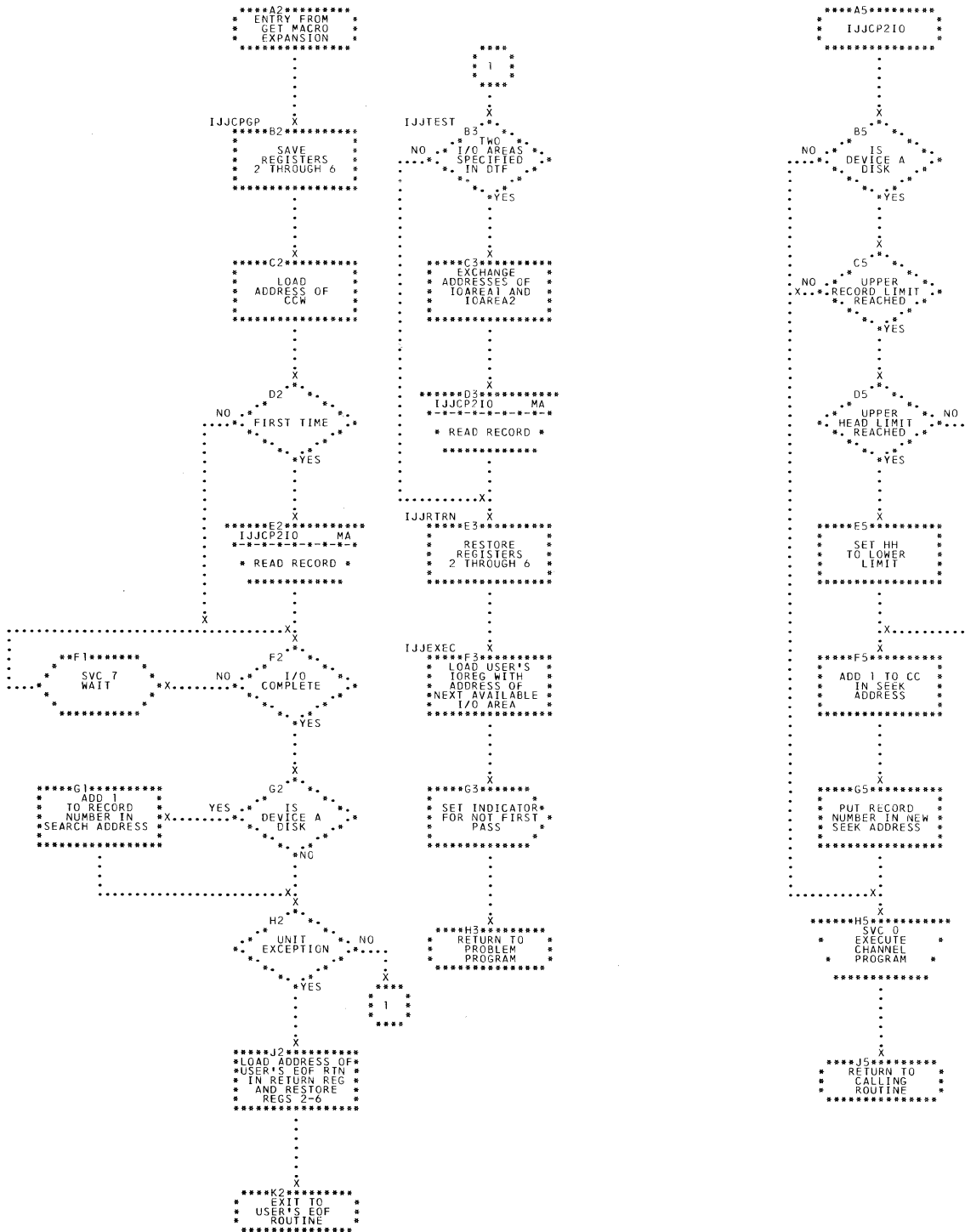


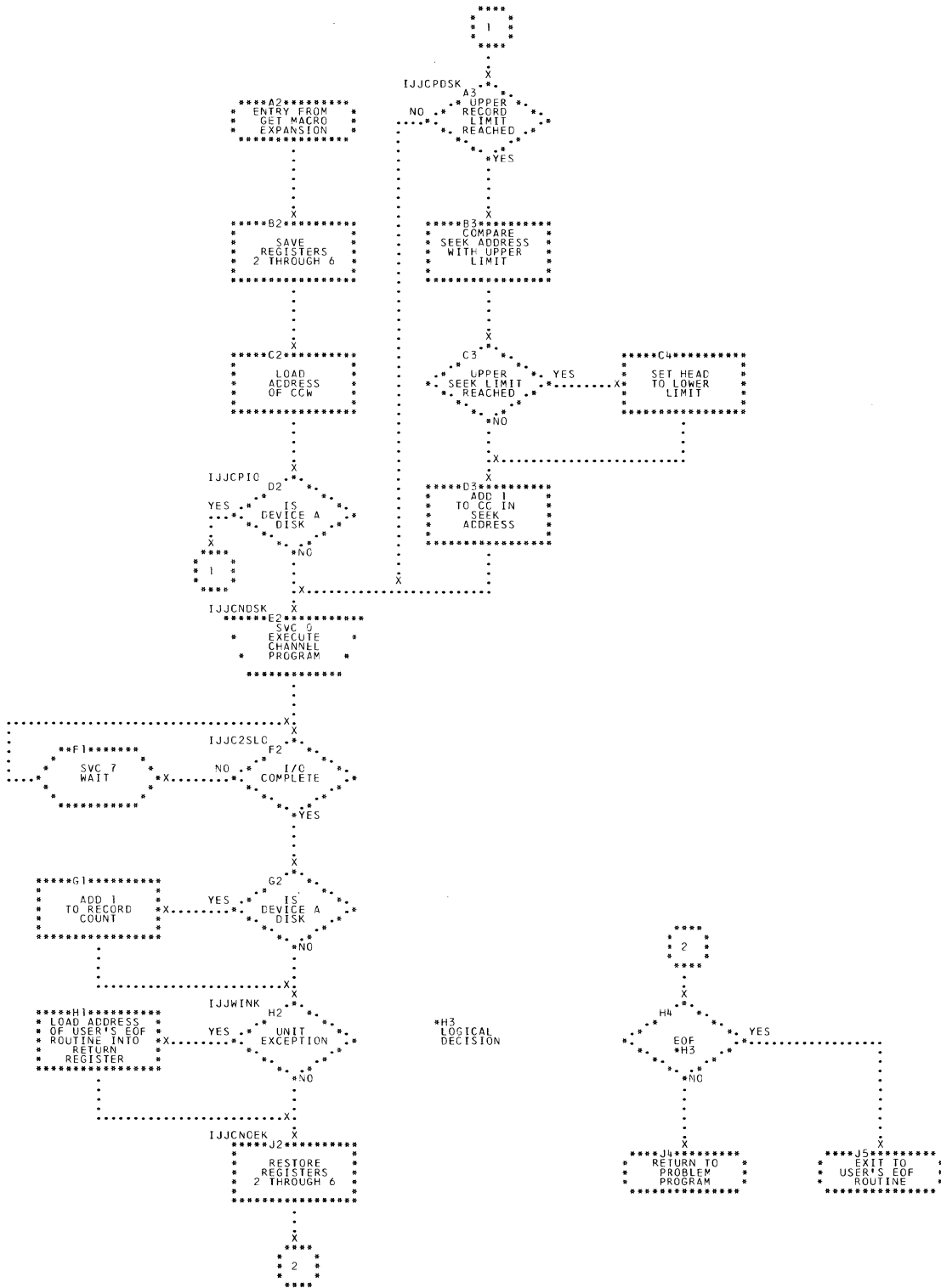
Chart LF. \$\$BMSGWR: Tape Open/Close Message Writer



● Chart MA. CPMOD: GET Macro, 2 I/O Areas



● Chart MB. CPMOD: GET Macro, 1 I/O Area



● Chart MC. CPMOD: GET Macro, IOPTR=YES

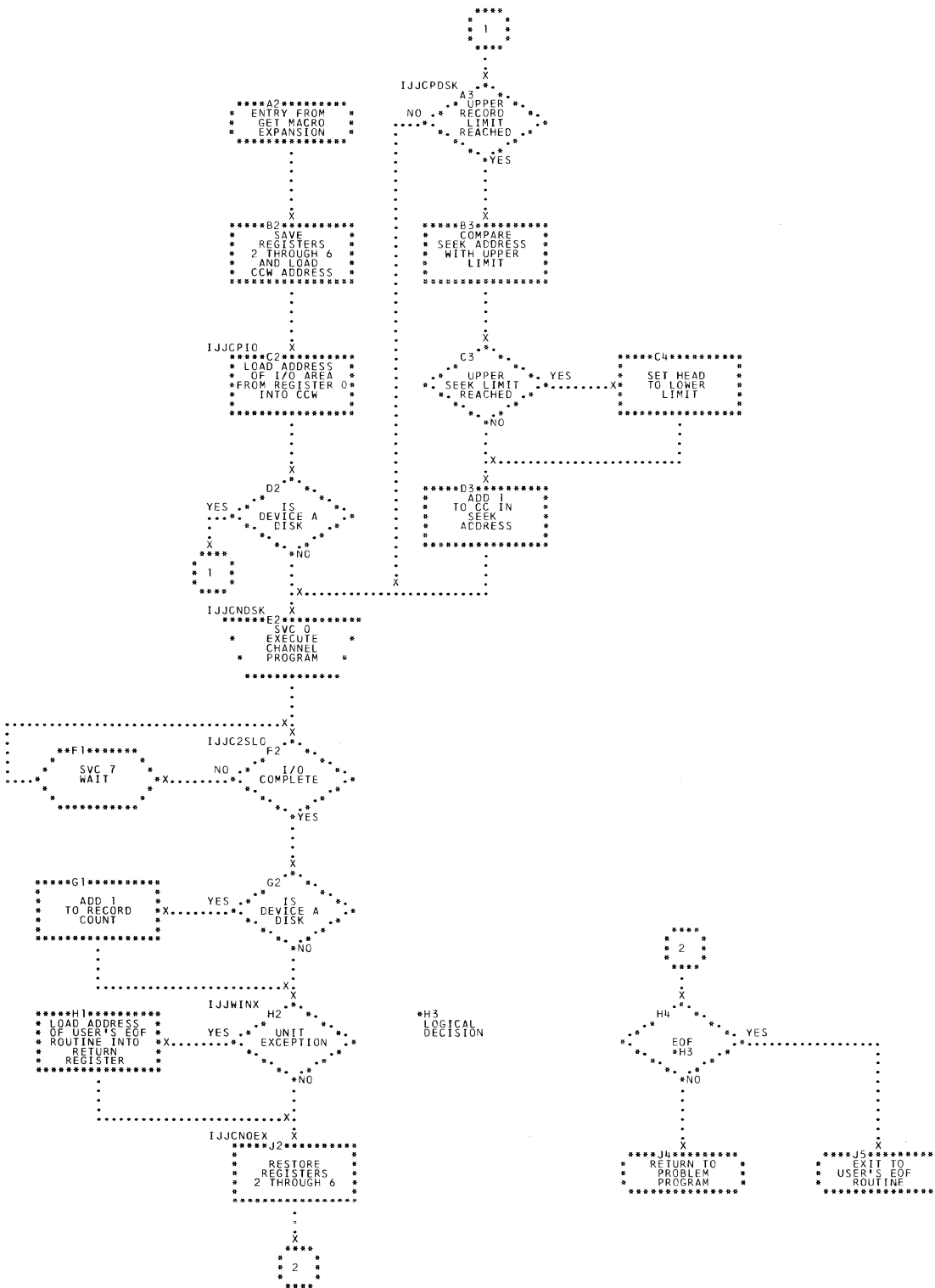


Chart MD. CPMOD: PUT Macro, 2 I/O Areas (Section 1 of 3)

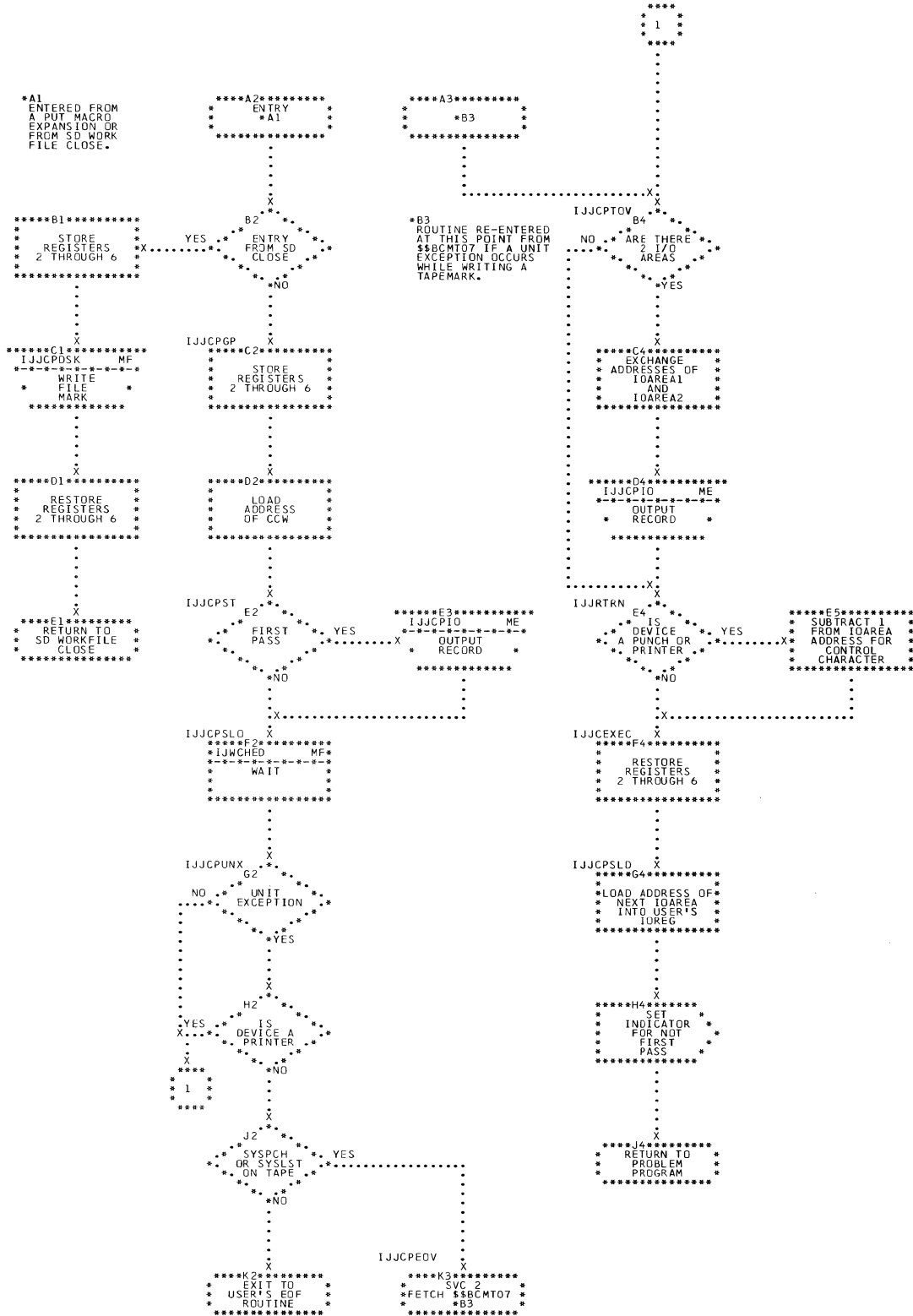


Chart ME. CPMOD: PUT Macro, 2 I/O Areas (Section 2 of 3)

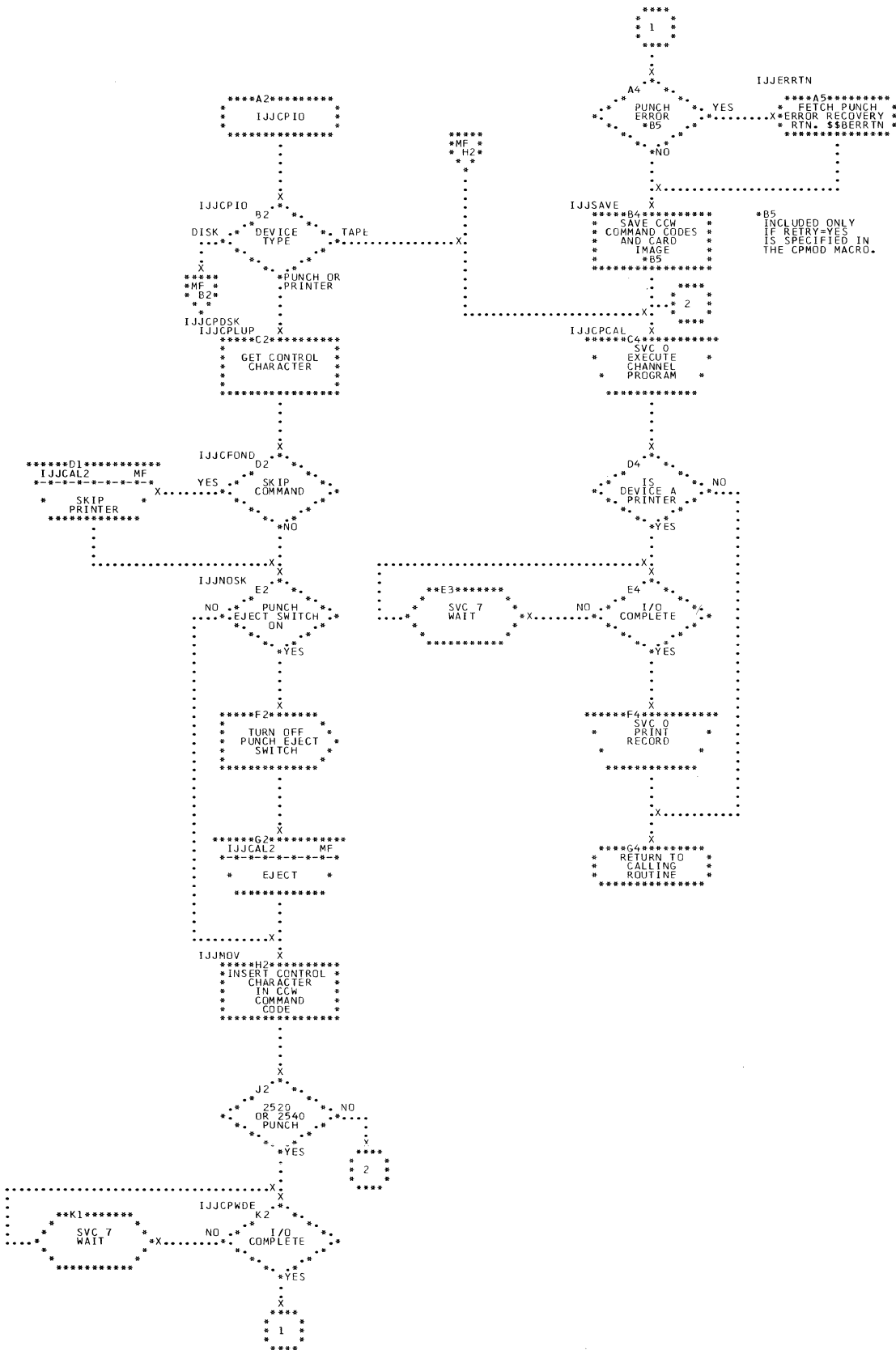
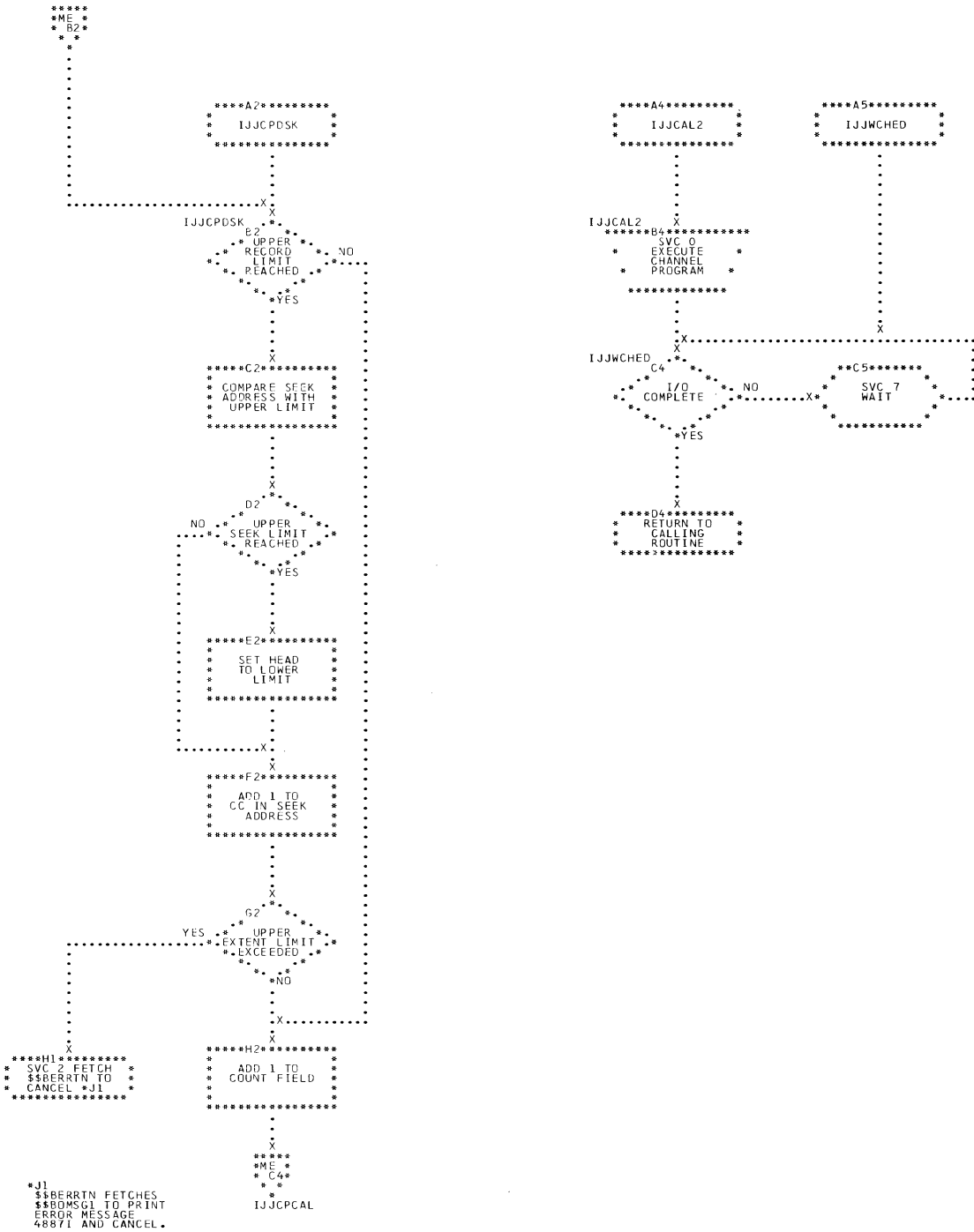


Chart MF. CPMOD: PUT Macro, 2 I/O Areas (Section 3 of 3)



● Chart MG. CPMOD: PUT Macro, 1 I/O Area (Section 1 of 2)

*A1
ENTERED FROM A PUT
MACRO EXPANSION OR
FROM AN SD WORKFILE
CLOSE.

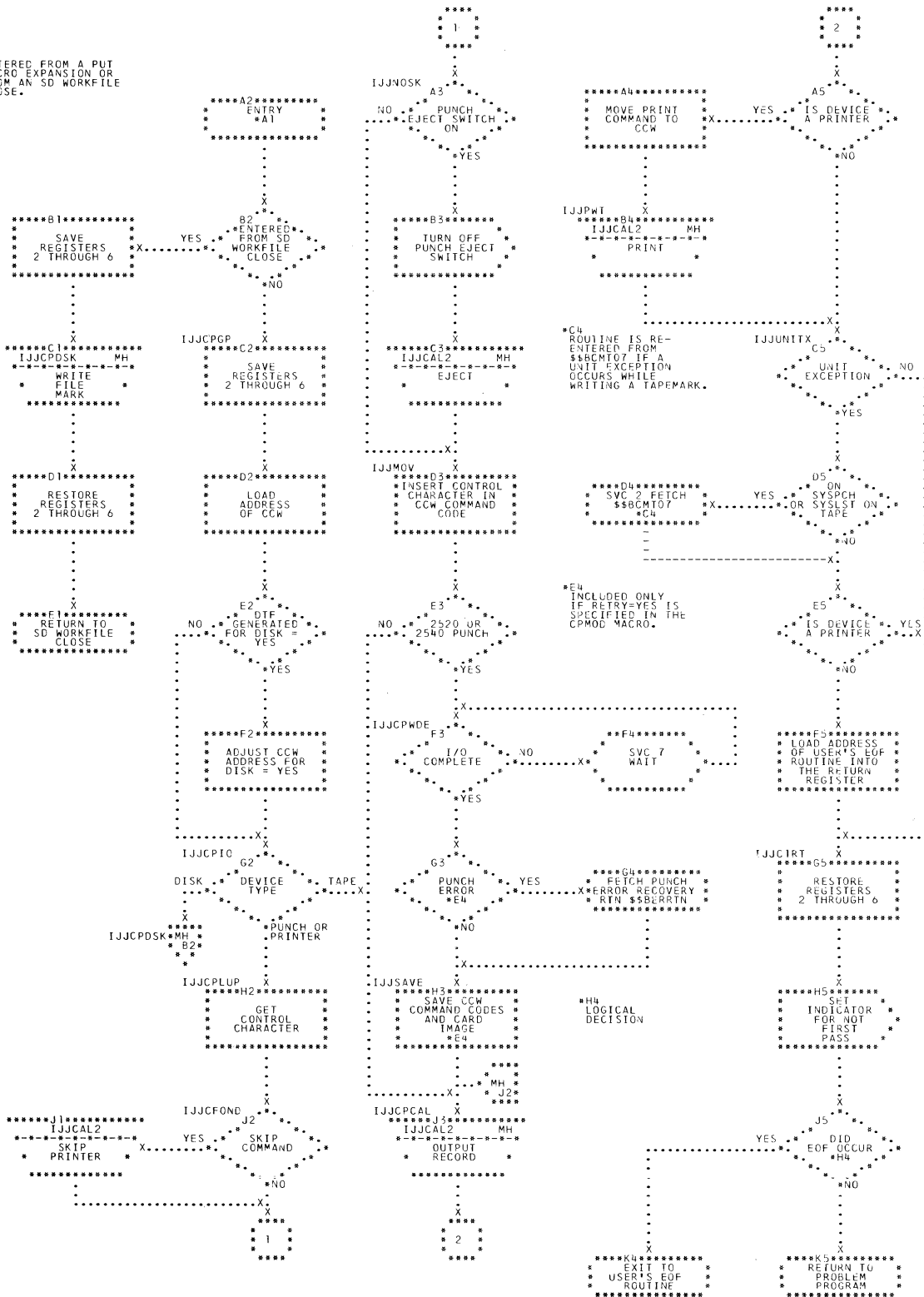
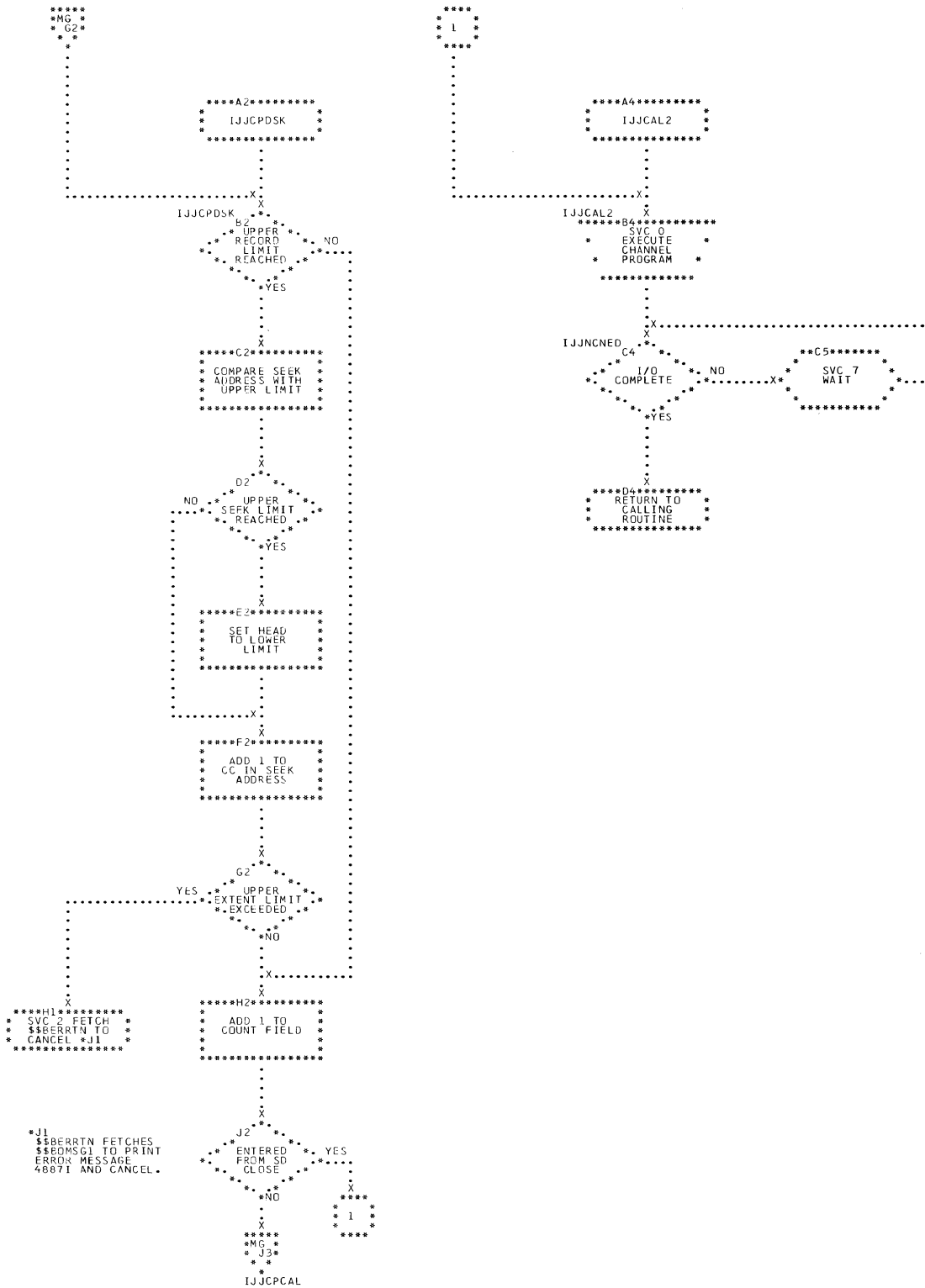


Chart MH. CPMOD: PUT Macro, 1 I/O Area (Section 2 of 2)



● Chart MJ. CPMOD: PUT Macro, IOPTR=YES (Section 1 of 2)

*A1
ENTERED FROM A PUT
MACRO EXPANSION OR
FROM AN SD WORKFILE
CLOSE.

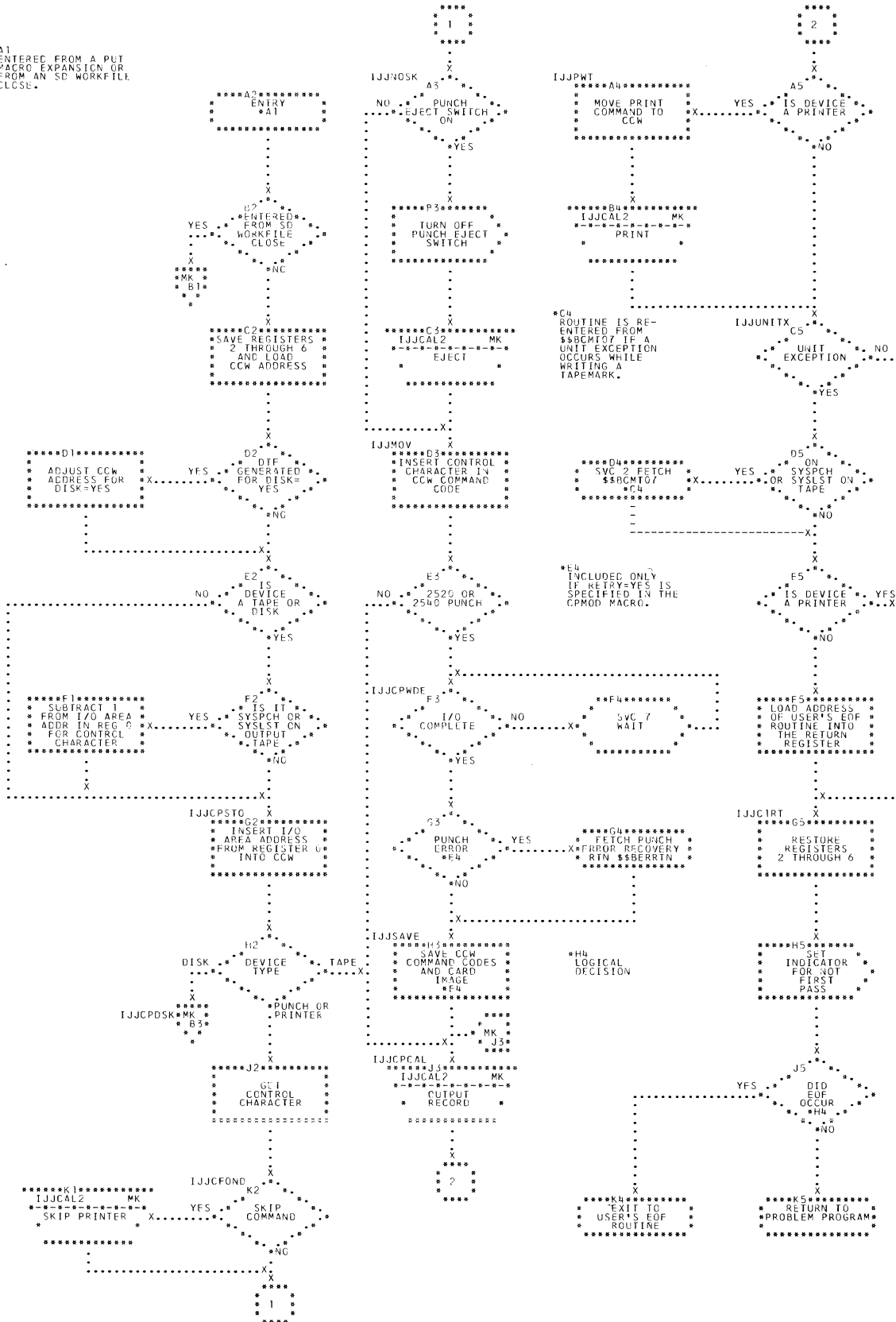
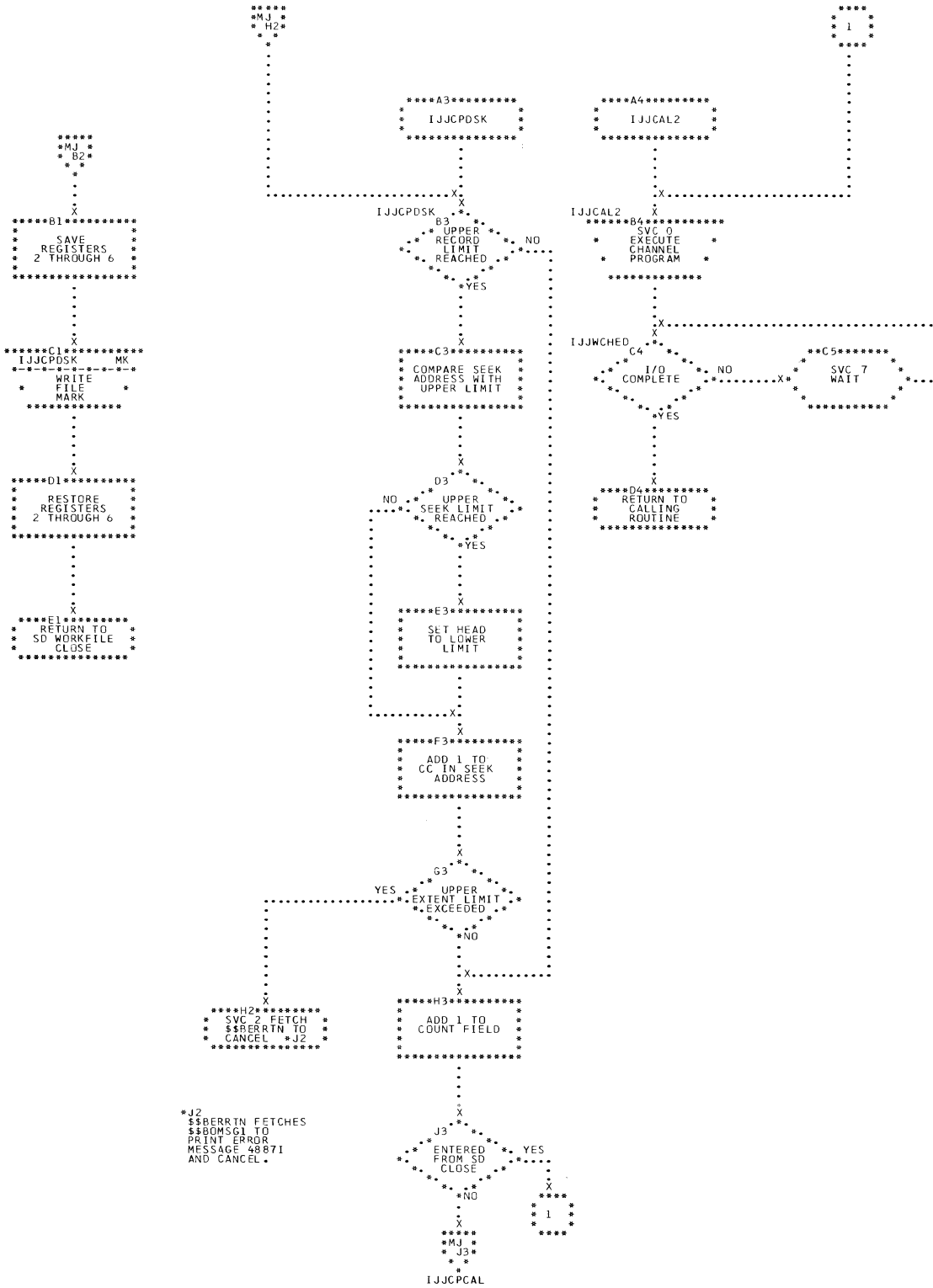
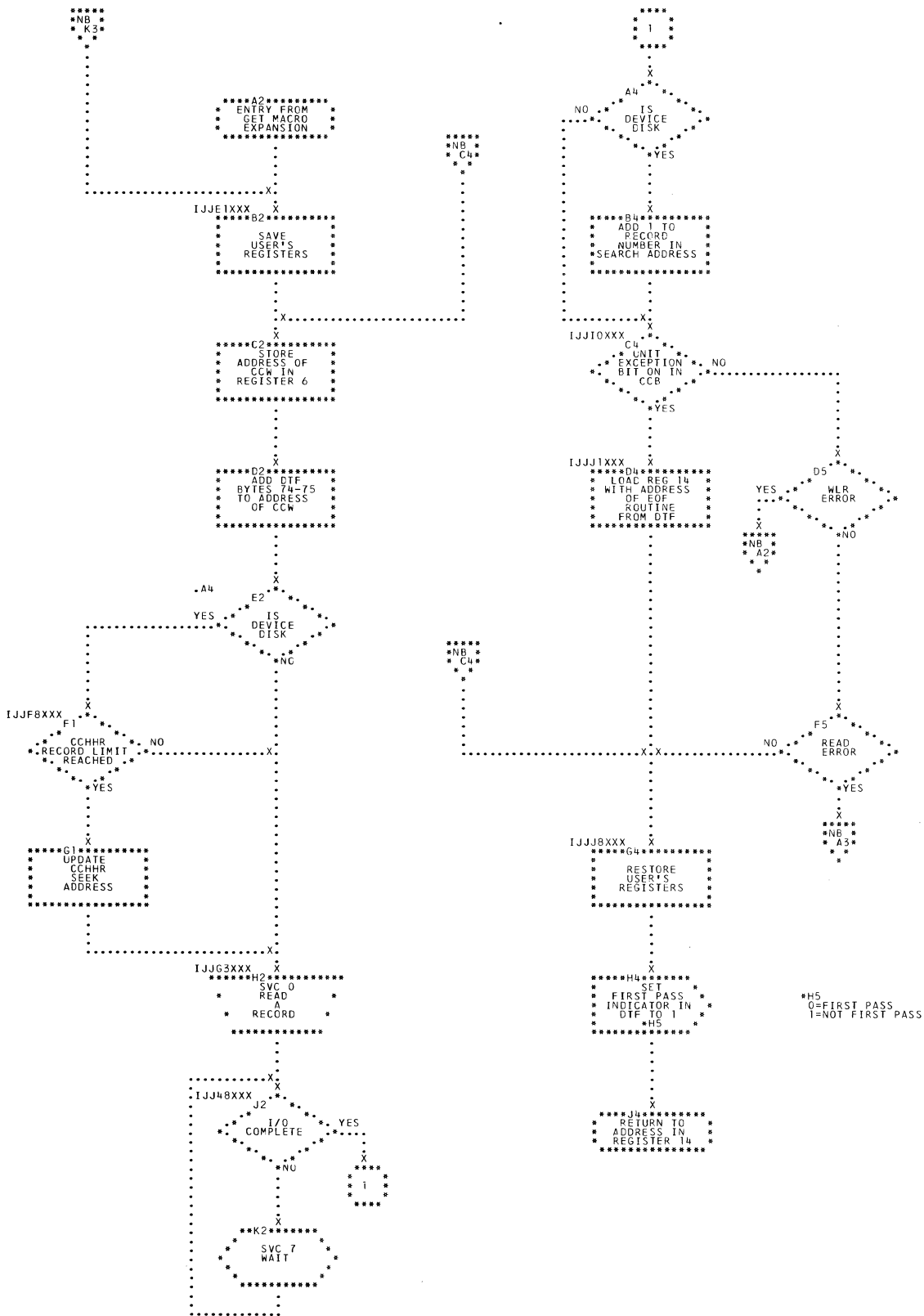


Chart MK. CPMOD: PUT Macro, IOPTR=YES (Section 2 of 2)



*J2
 \$\$\$BERRTN FETCHES
 \$\$\$BMSG1 TO
 PRINT ERROR
 MESSAGE 48871
 AND CANCEL.

● Chart NA. DIMOD: GET Macro, 1 I/O Area (Section 1 of 2)



● Chart NC. DIMOD: GET Macro, 2 I/O Areas (Section 1 of 2)

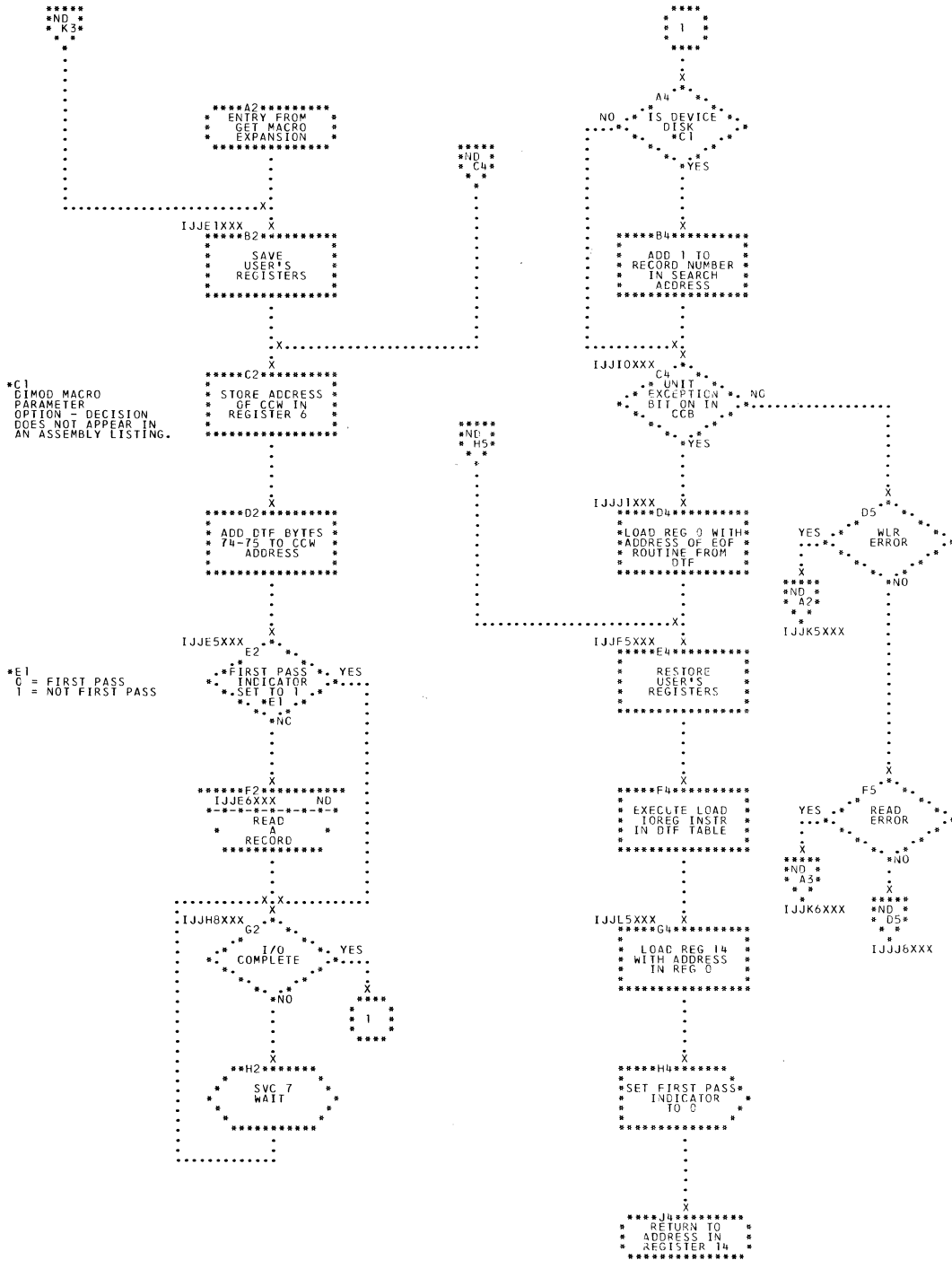


Chart ND. DIMOD: GET Macro, 2 I/O Areas (Section 2 of 2)

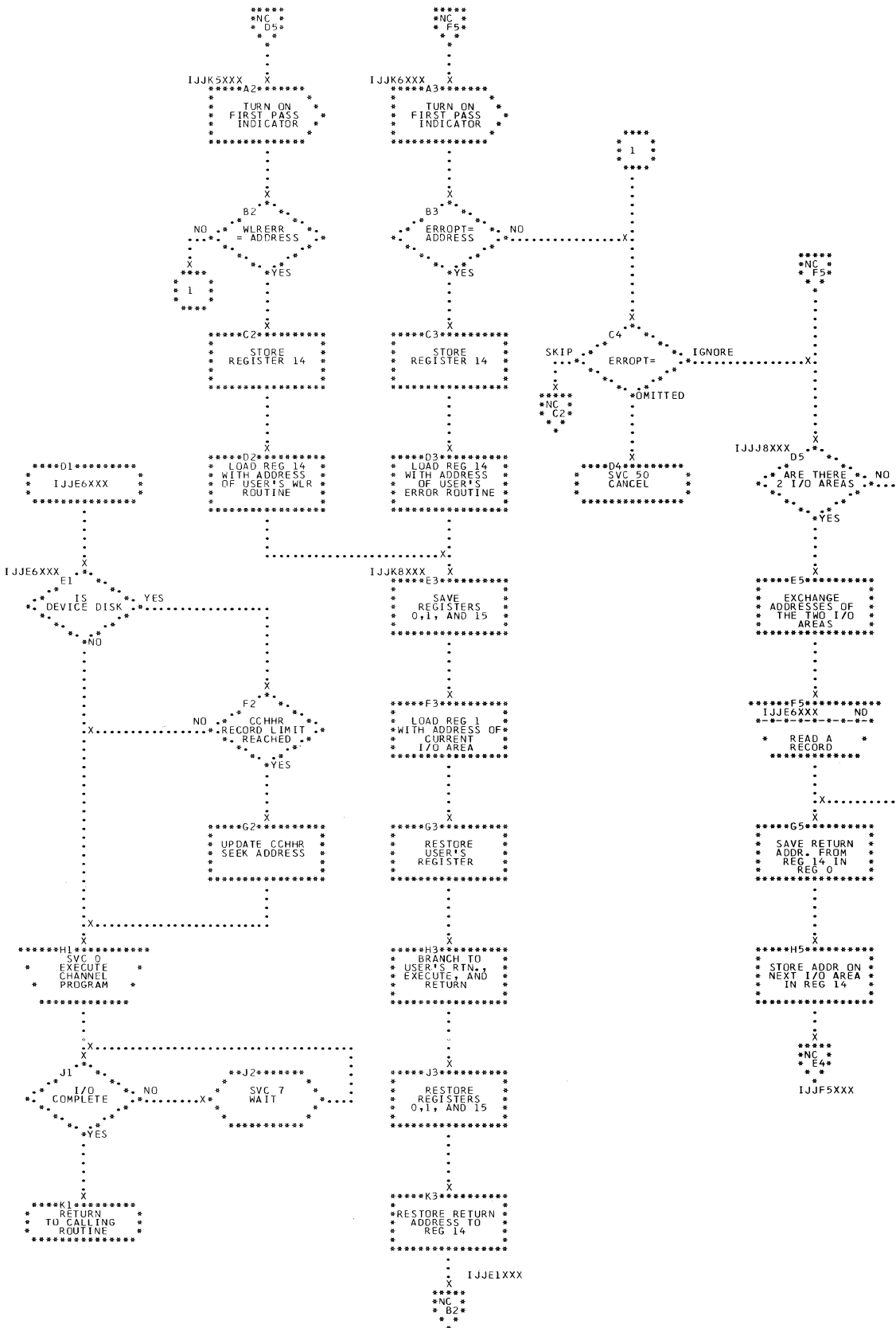
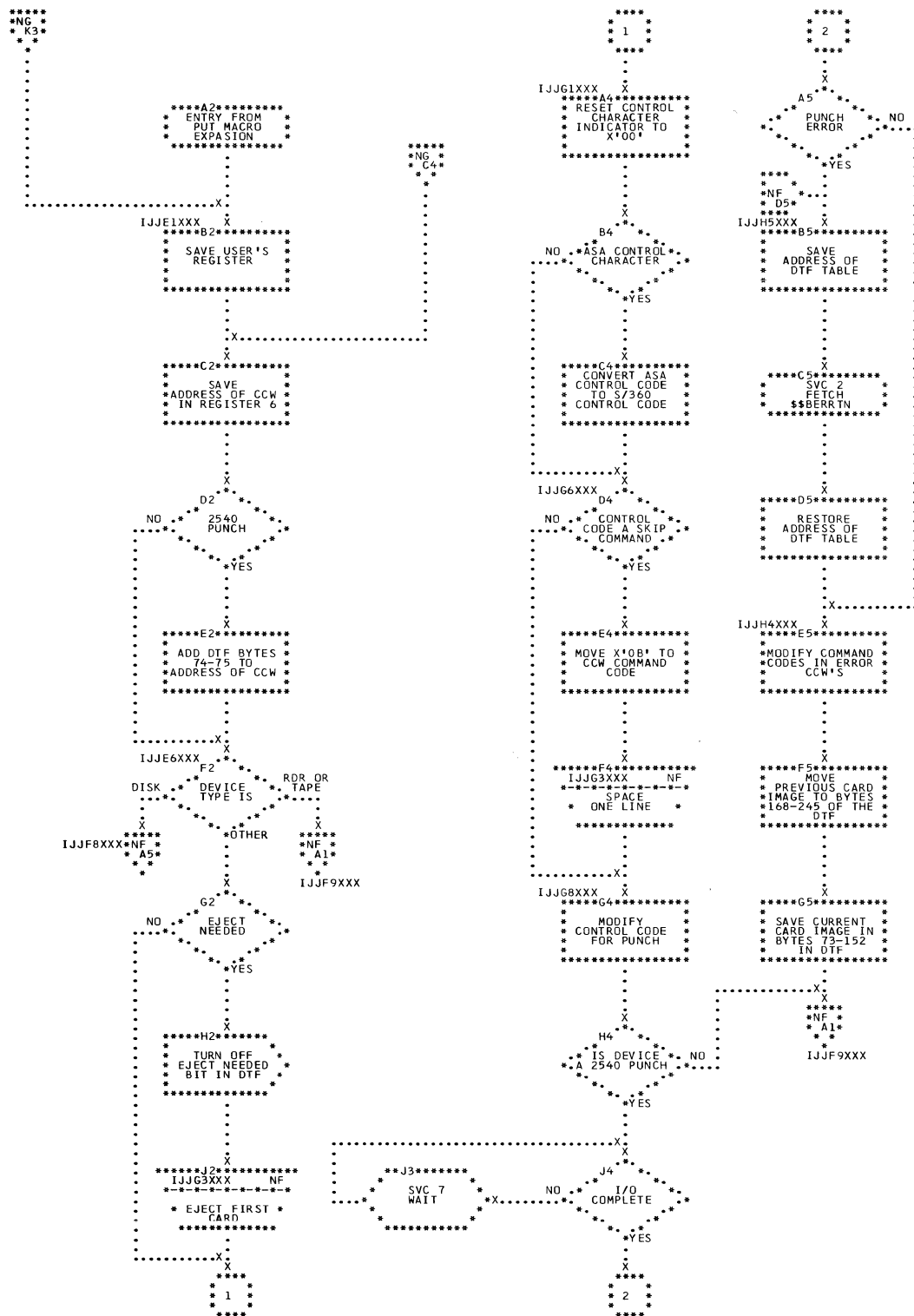
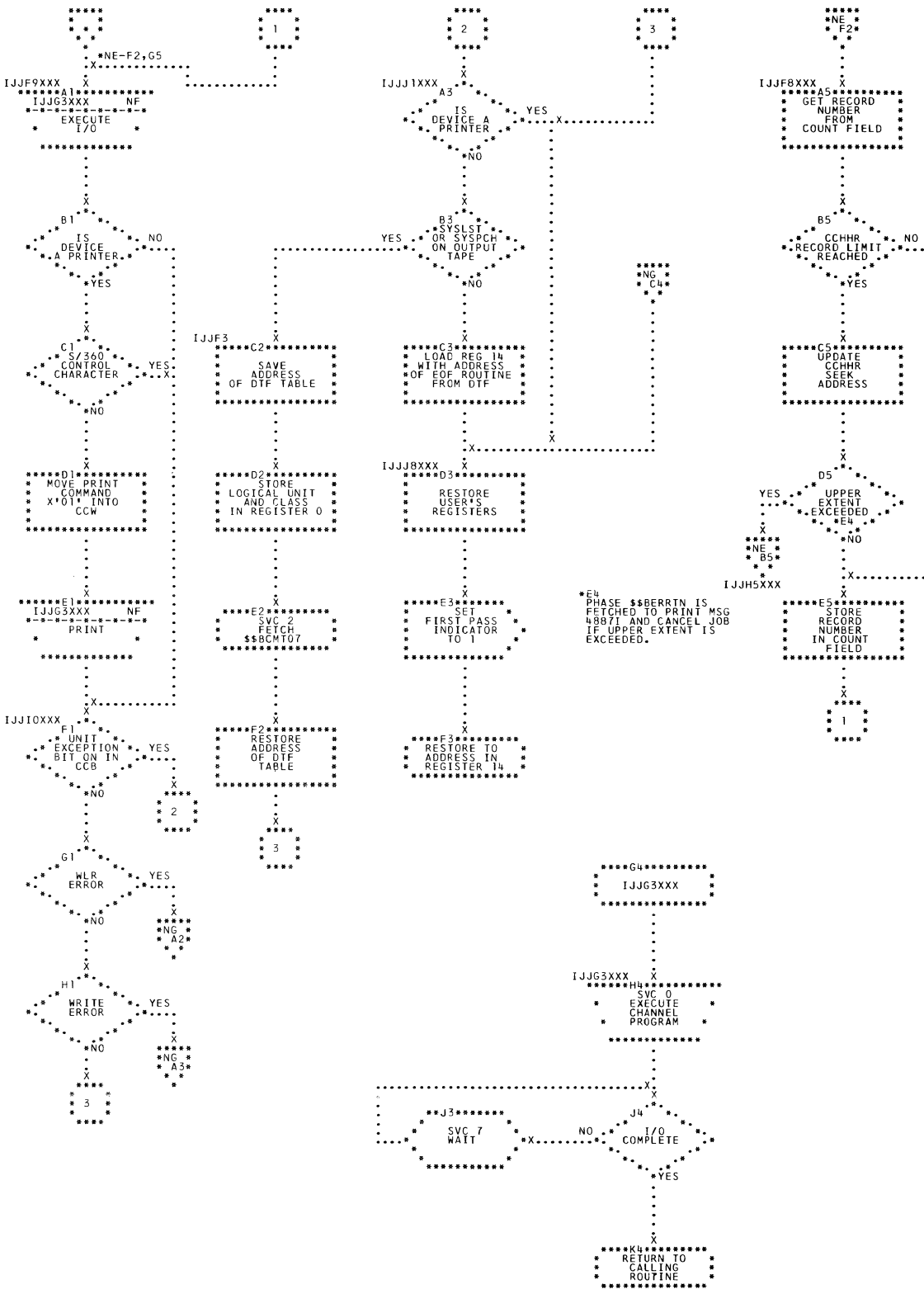


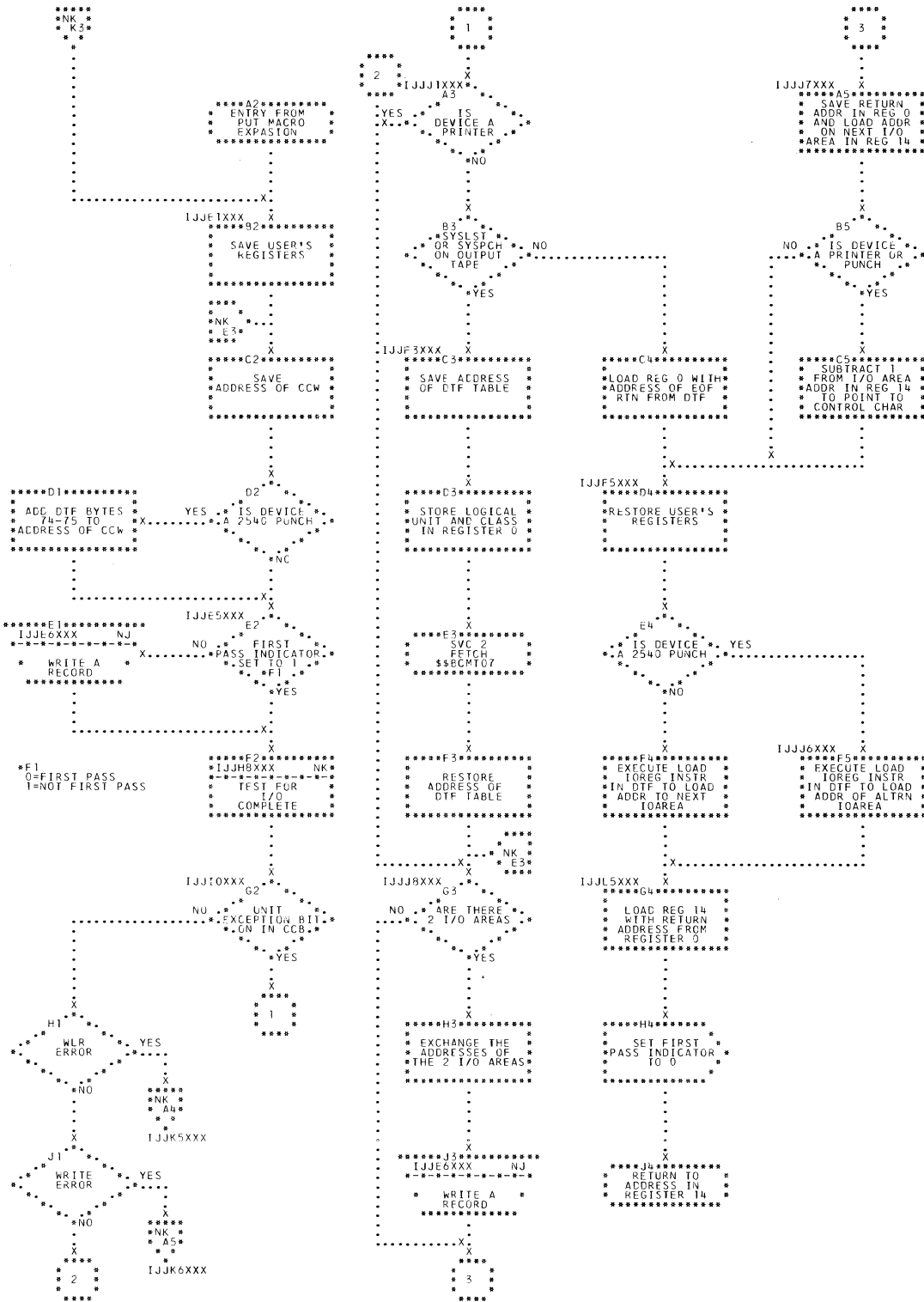
Chart NE. DIMOD: PUT Macro, 1 I/O Area (Section 1 of 3)



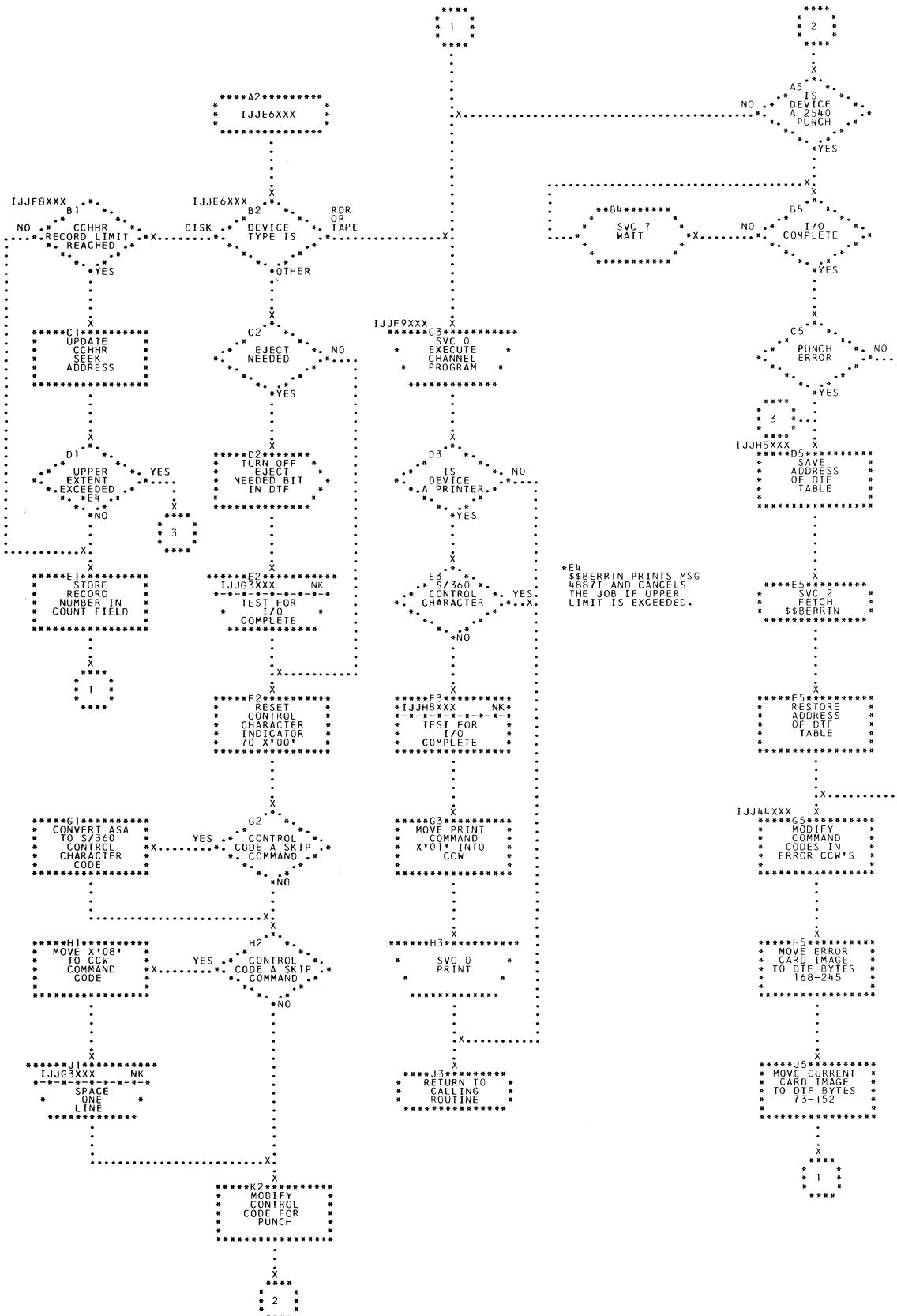
● Chart NF. DIMOD: PUT Macro, 1 I/O Area (Section 2 of 3)



● Chart NH. DIMOD: PUT Macro, 2 I/O Areas (Section 1 of 3)



● Chart NJ. DIMOD: PUT Macro, 2 I/O Areas (Section 2 of 3)



● Chart NK. DIMOD: PUT Macro, 2 I/O Areas (Section 3 of 3)

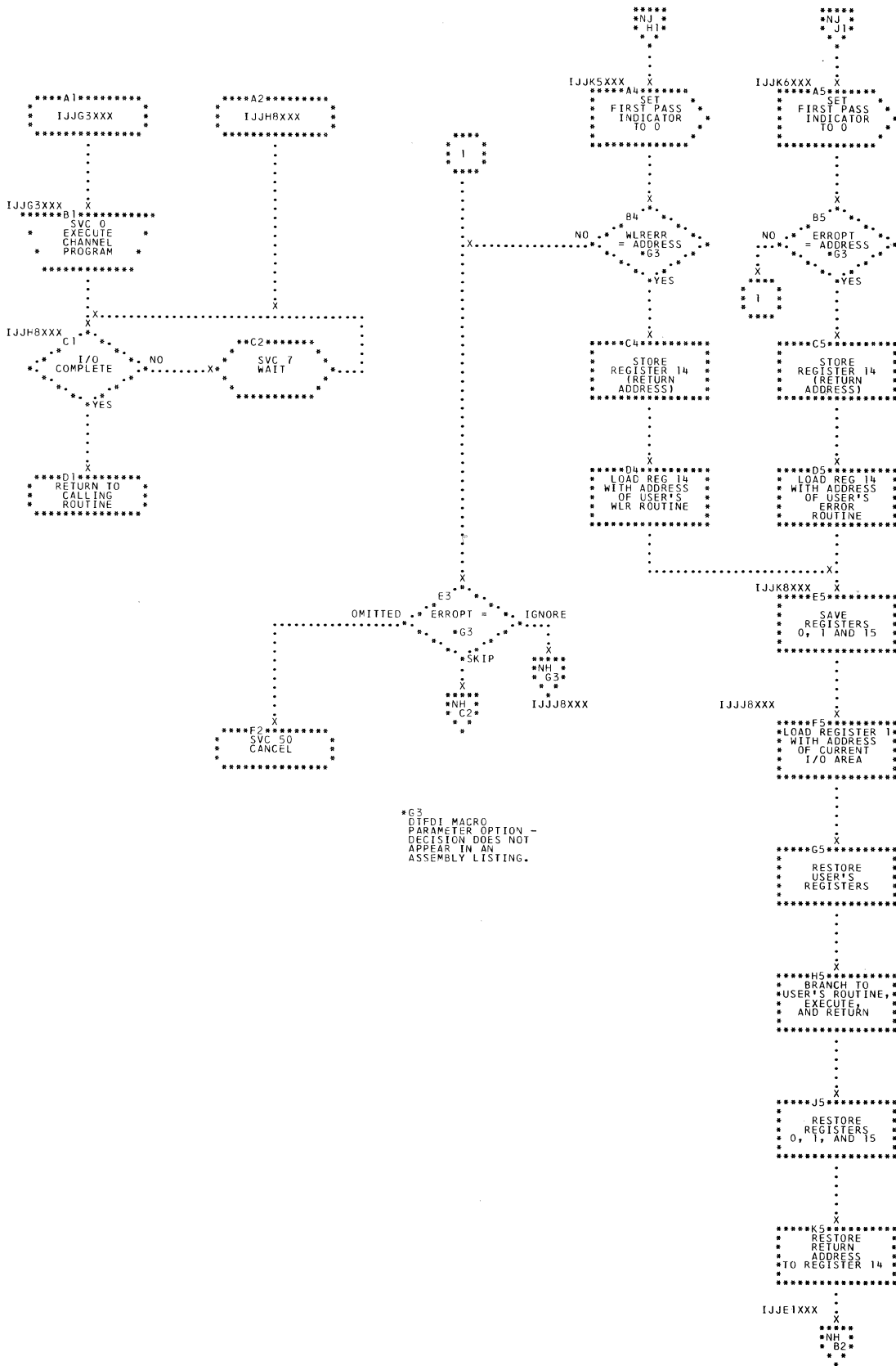


Chart QB. \$\$\$BOCP01: Open Device Independent Files, Phase 1 (Section 2 of 3)

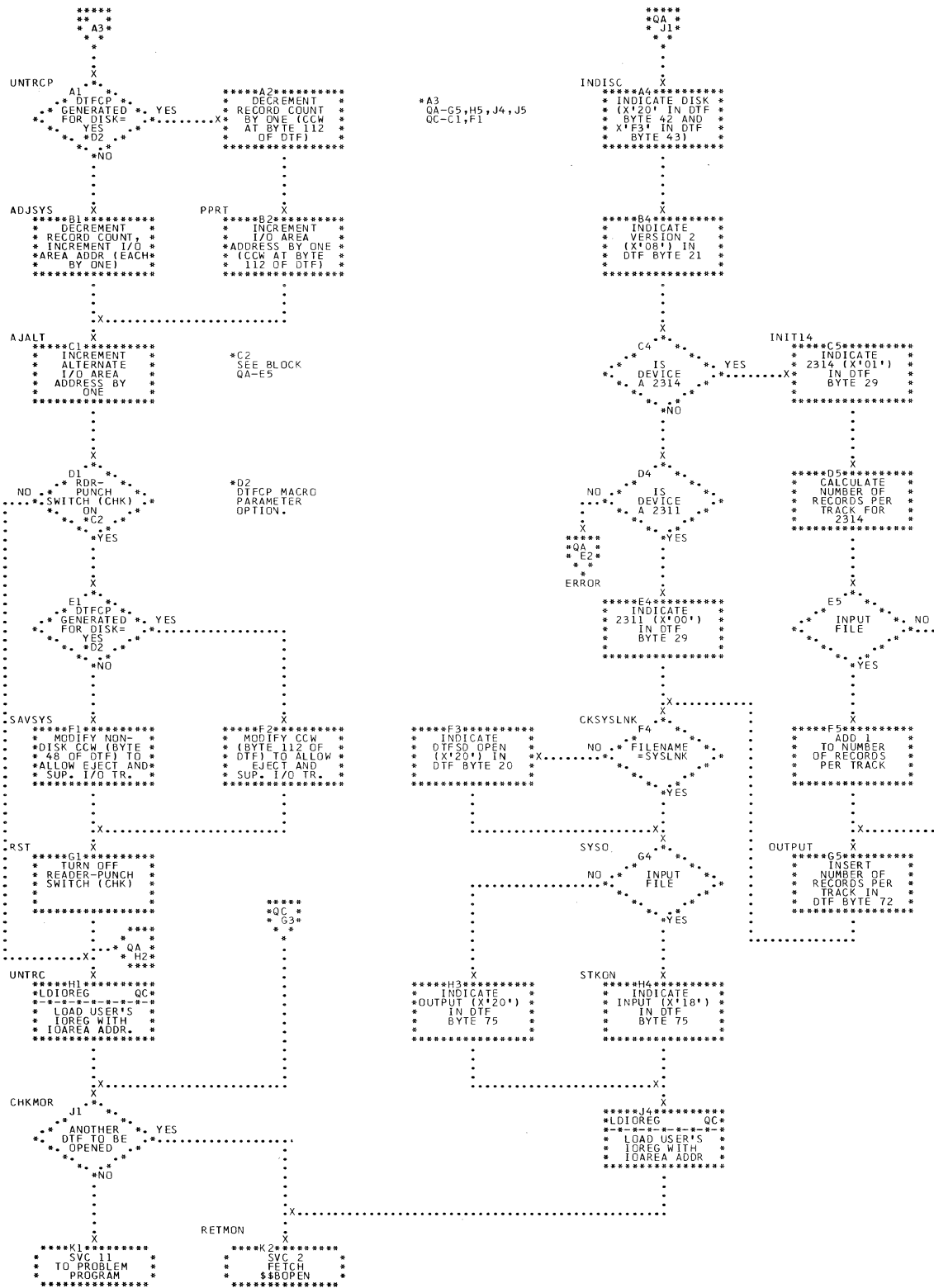


Chart QC. \$\$\$BOCP01: Open Device Independent Files, Phase 1 (Section 3 of 3)

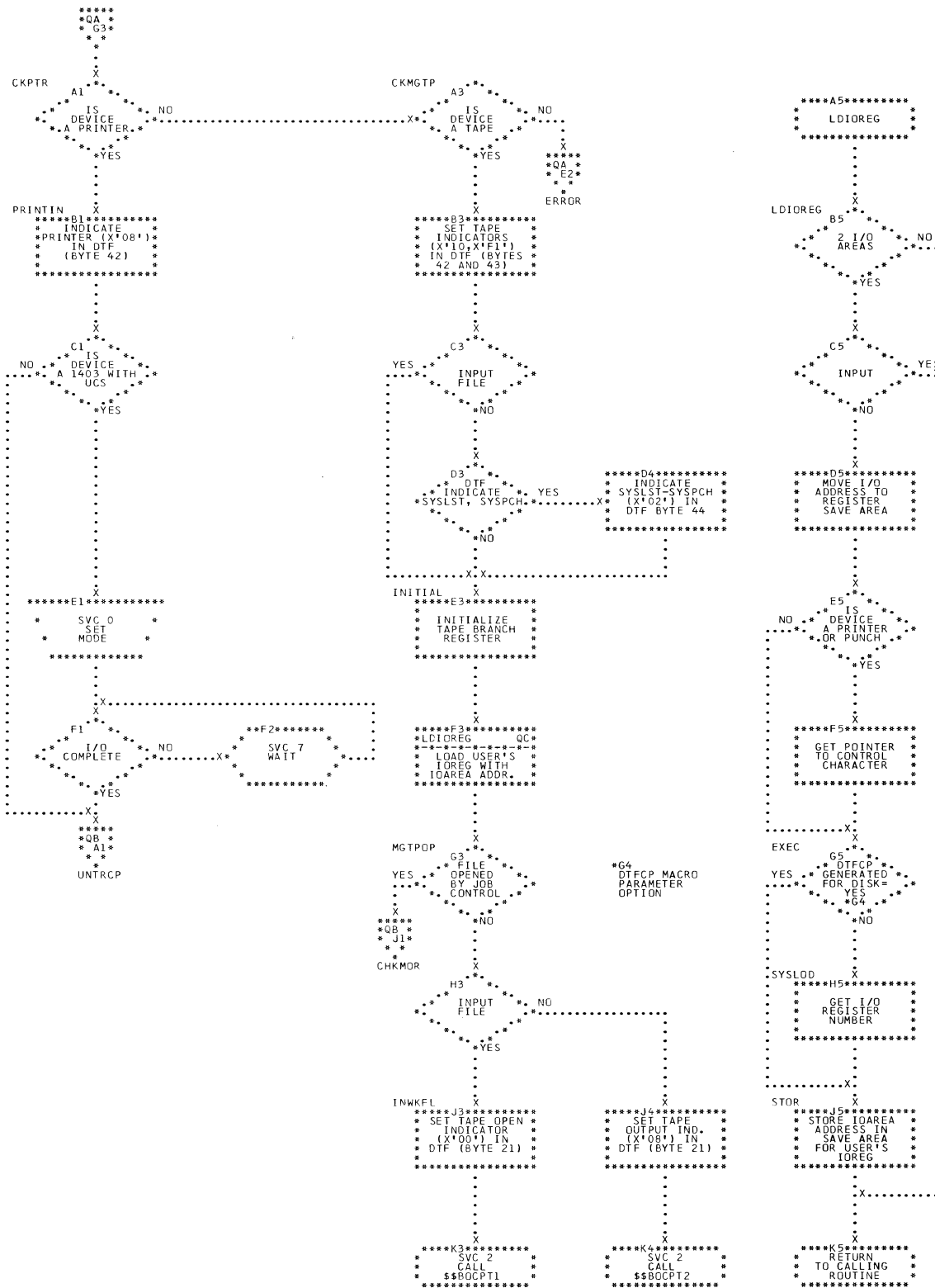


Chart QD. \$\$\$BOCP02: Open Device Independent Files, Phase 2 (Section 1 of 3)

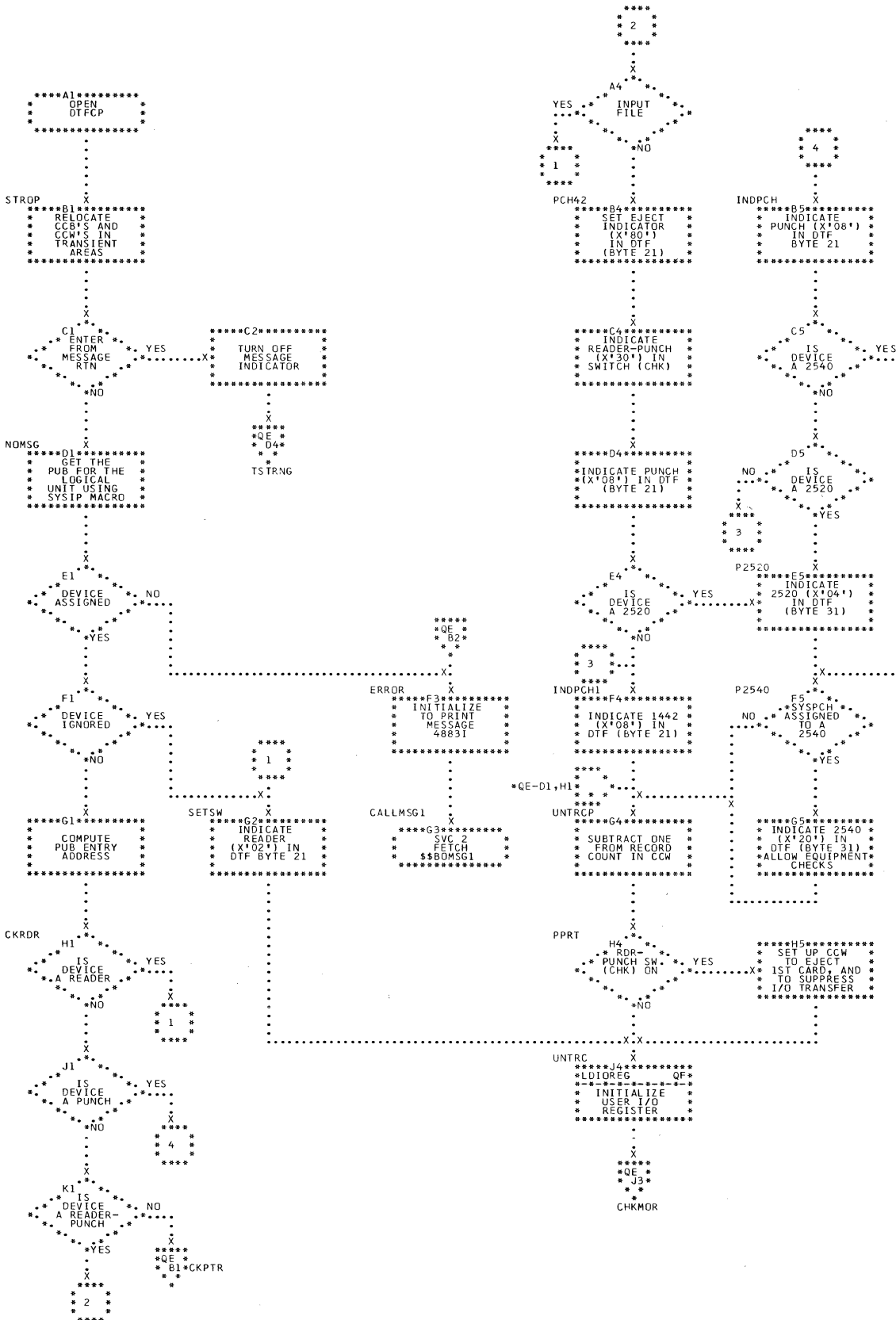


Chart QE. \$\$BOCP02: Open Device Independent Files, Phase 2 (Section 2 of 3)

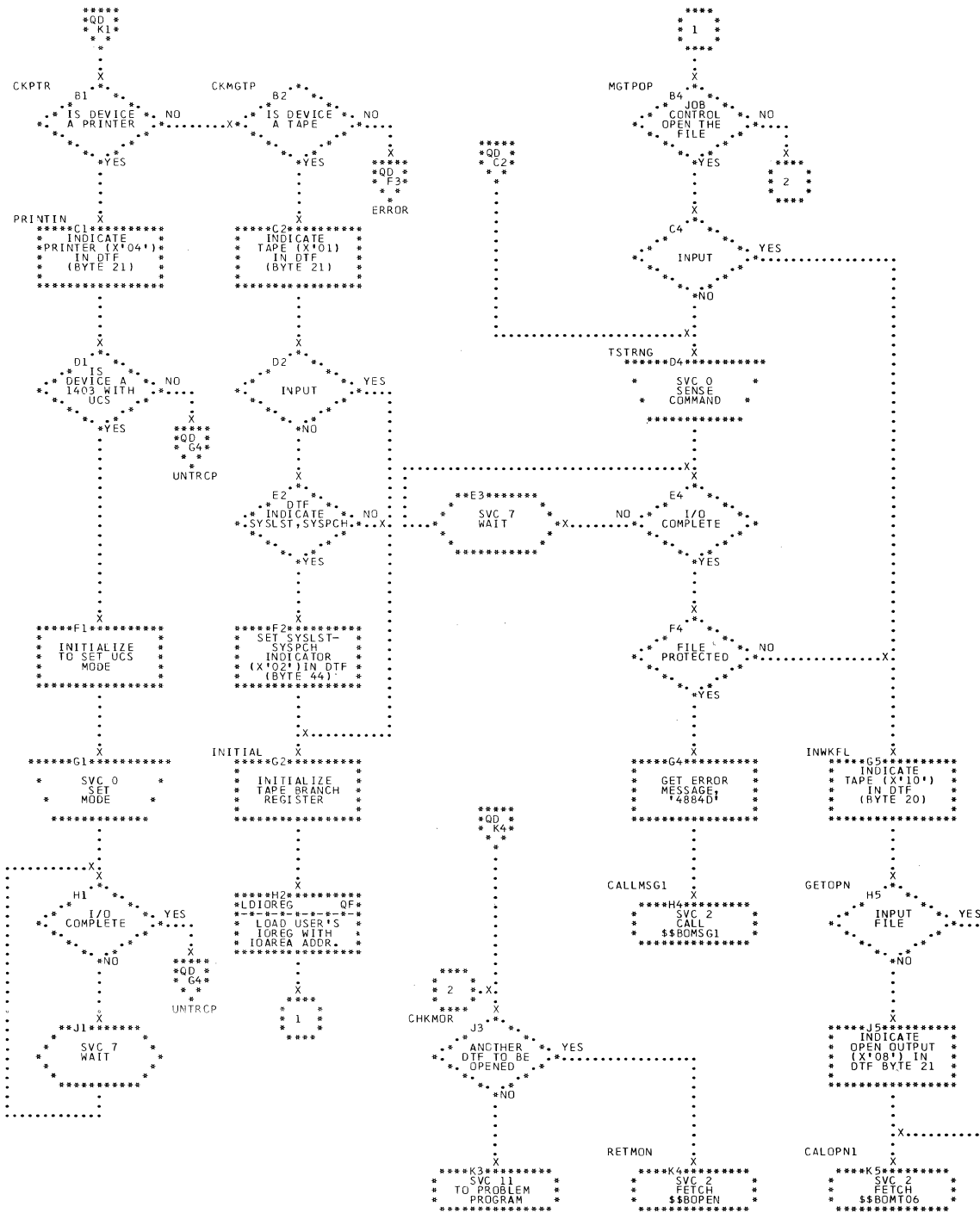


Chart QF. \$\$BOCP02: Open Device Independent Files, Phase 2 (Section 3 of 3)

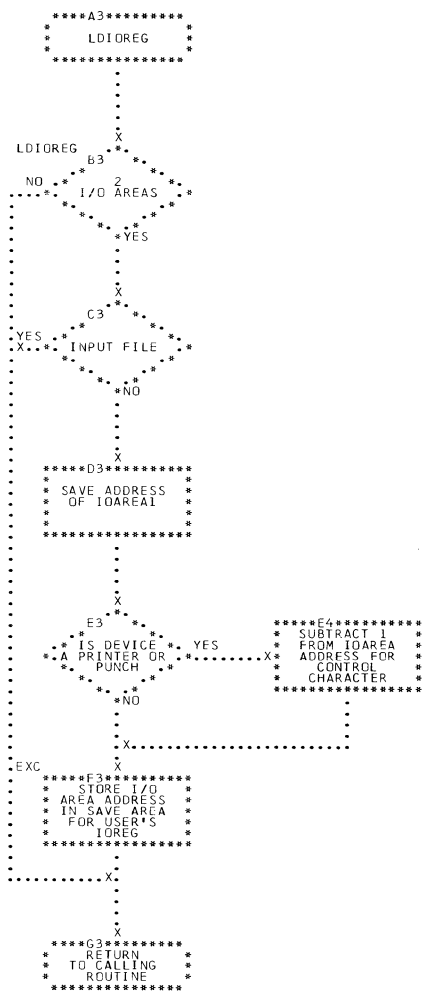


Chart QG. \$\$\$BOCP11: Open DTFCP (Version 1 Only), Phase 1 (Section 1 of 3)

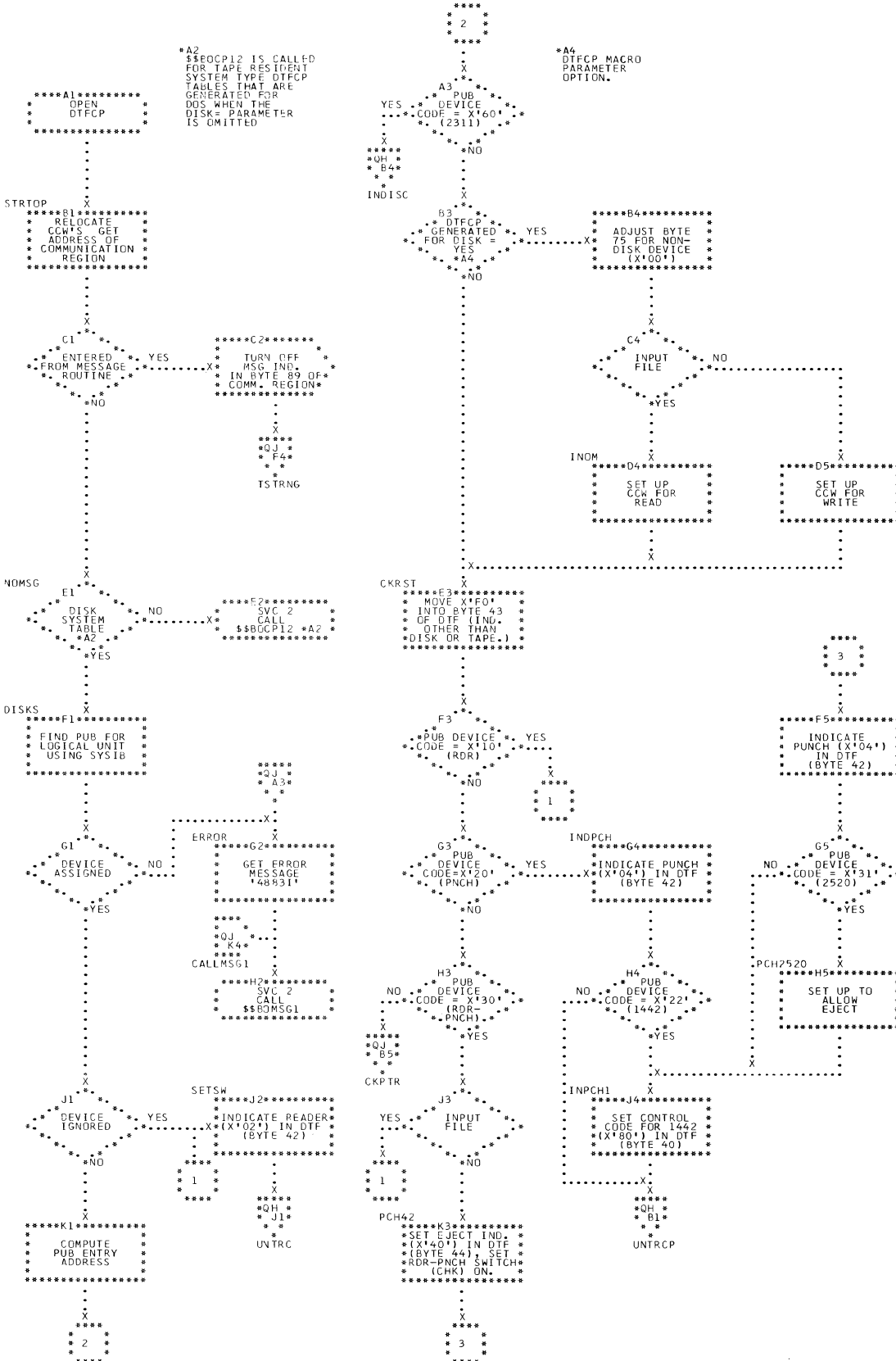


Chart QH. \$\$\$BOCP11: Open DTFCP (Version 1 Only), Phase 1 (Section 2 of 3)

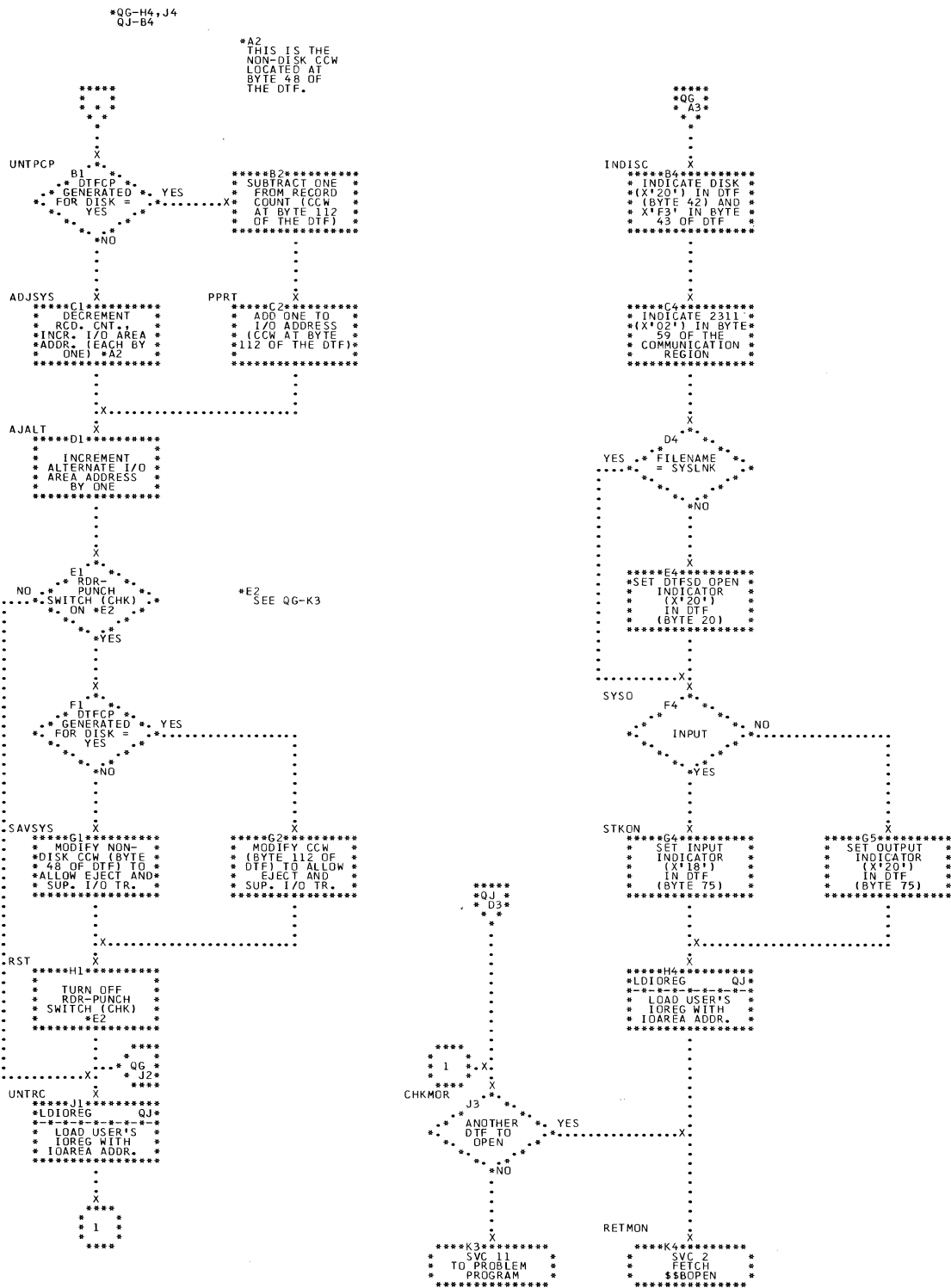


Chart QJ. \$\$\$BOCP11: Open DTFCP (Version 1 Only), Phase 1 (Section 3 of 3)

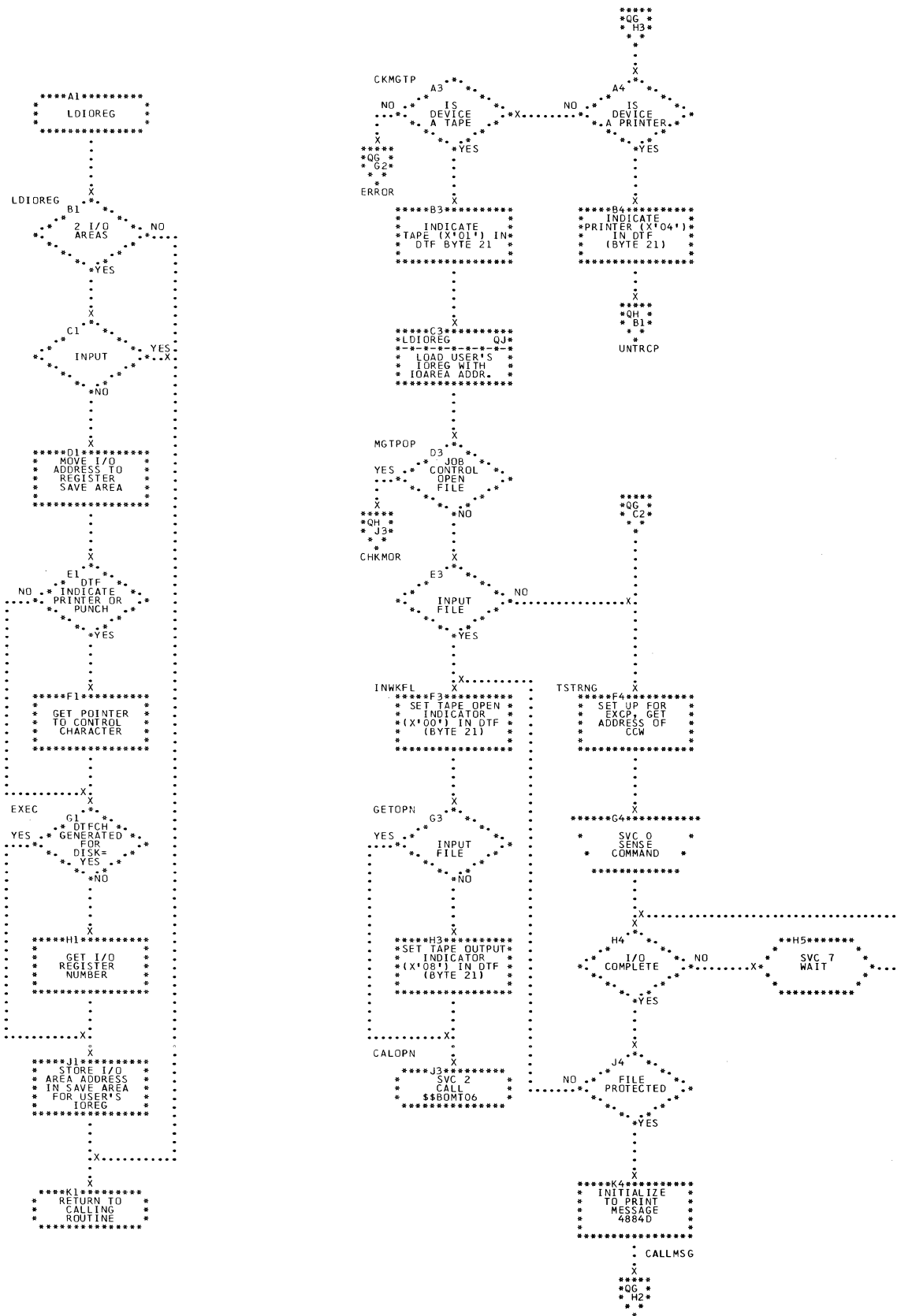


Chart QK. \$\$\$BOCP12: Open DTFCP (Version 1 Only), Phase 2 (Section 1 of 2)

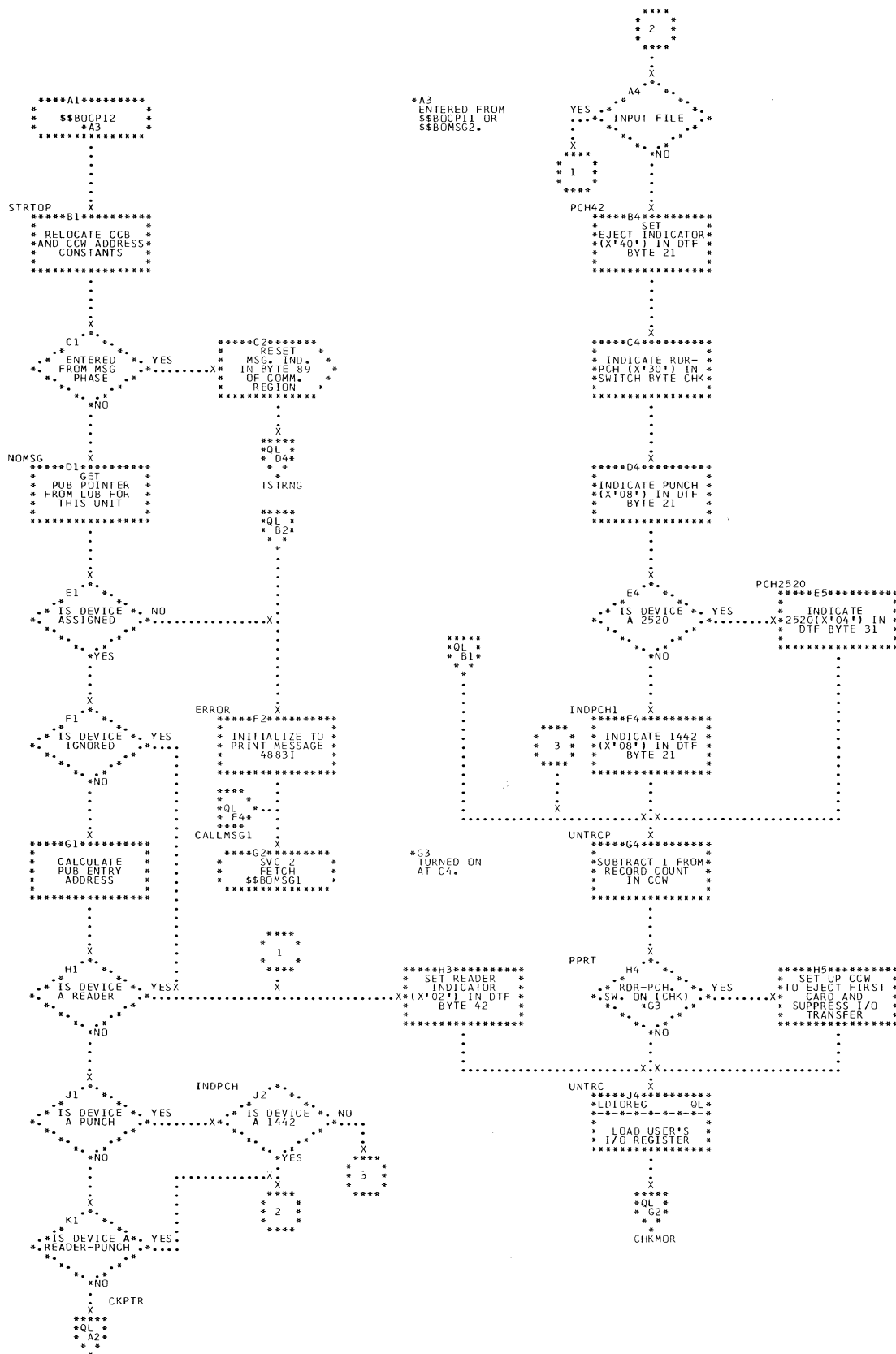


Chart QL. \$\$\$BOCP12: Open DTFCP (Version 1 Only), Phase 2 (Section 2 of 2)

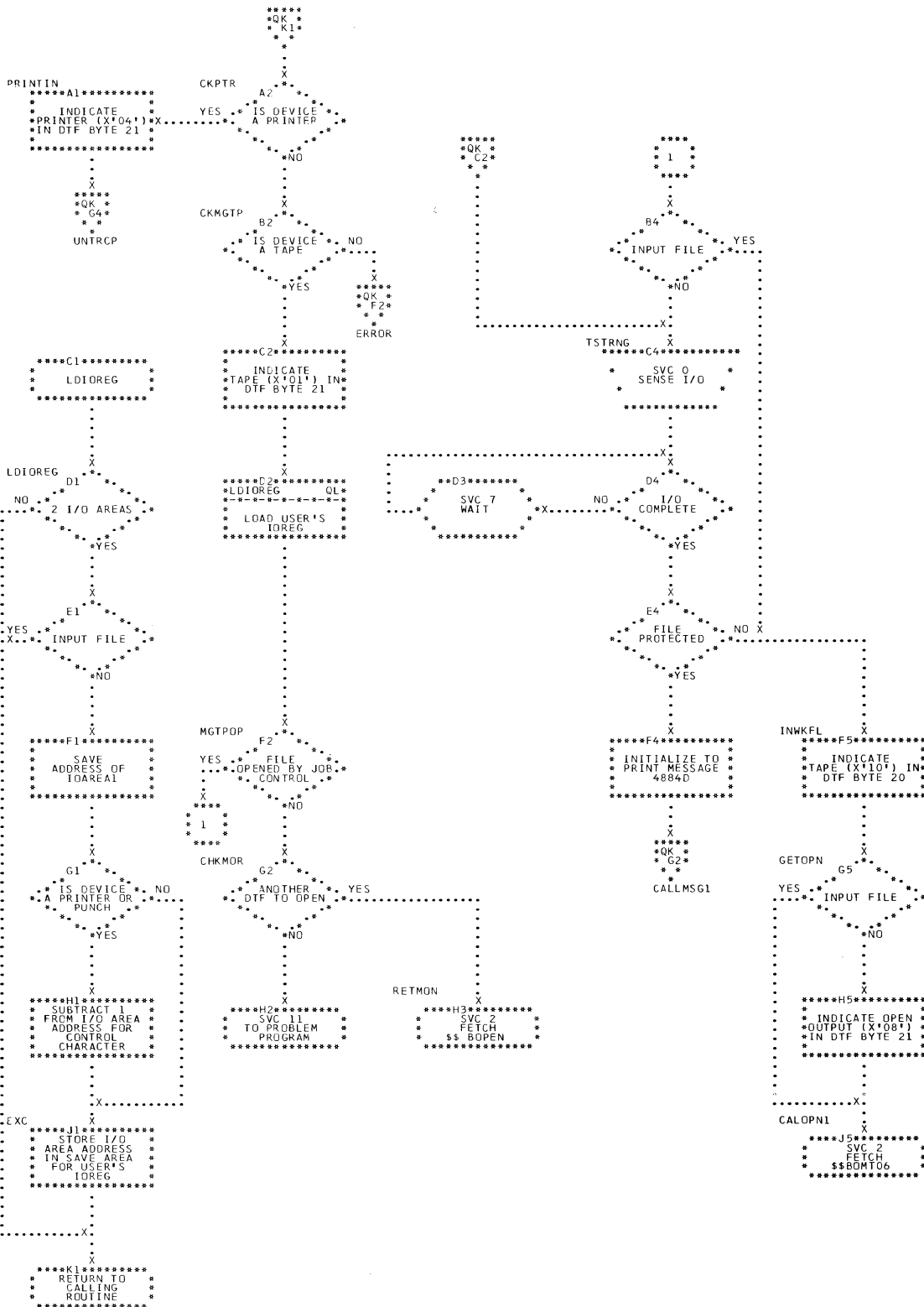


Chart QM. \$\$BOCP1: Open DTFCP and DTFDI Input Tape (Section 1 of 5)

*A1
 ENTERED FROM--
 \$\$BOCP01 - IF
 THE FILE IS
 DEFINED BY A
 DISK SYSTEM
 INPUT DTFCP
 AND THE DEVICE
 IS TAPE.
 2. \$\$BOCPM1 - AFTER
 A MESSAGE HAS
 BEEN ISSUED AND
 A NON-CANCEL
 REPLY HAS BEEN
 RECEIVED.

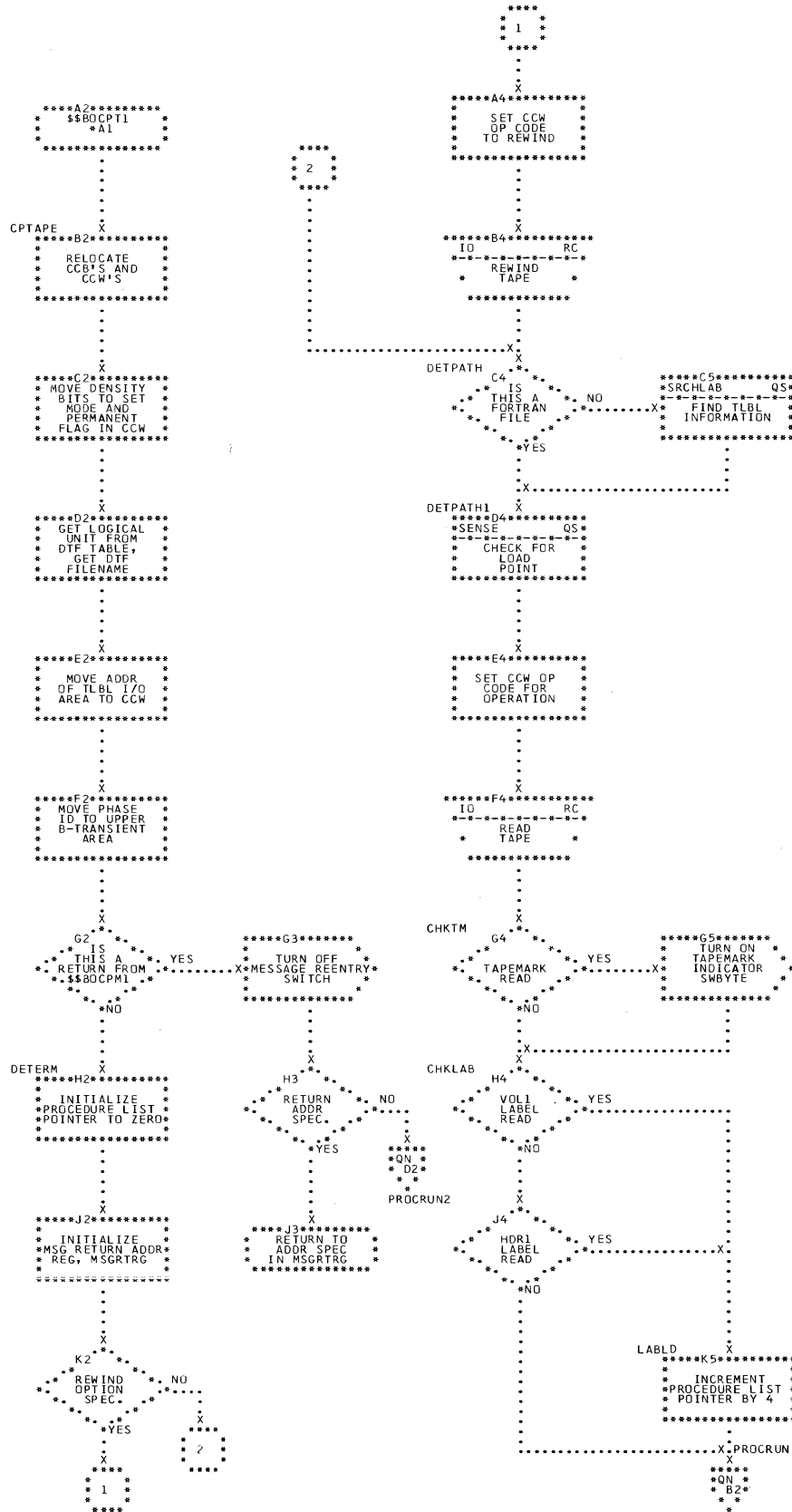


Chart QP. \$\$\$BOCPT1: Open DTFCP and DTFDI Input Tape (Section 3 of 5)

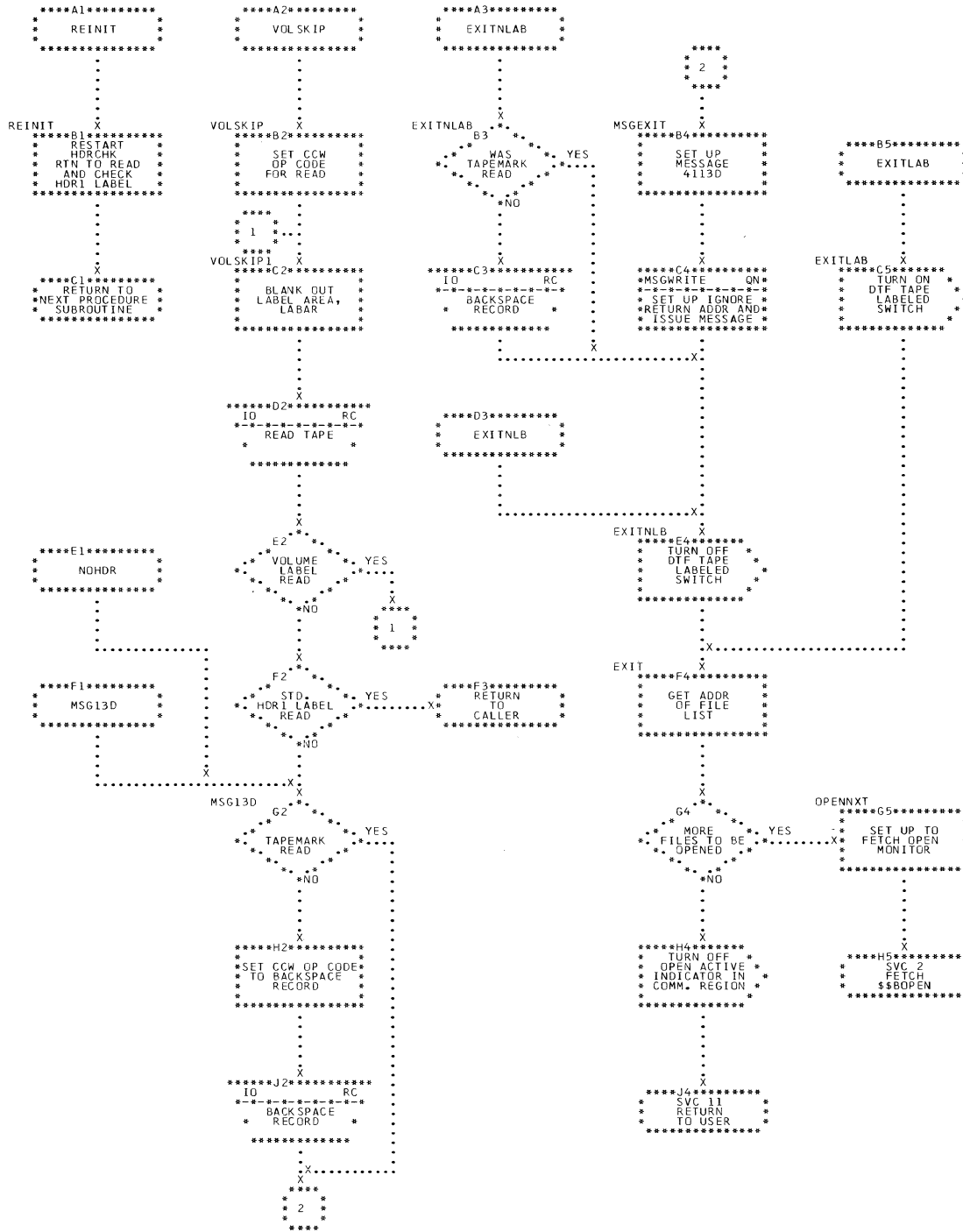


Chart QR. \$\$BOCP1: Open DTFCP and DTFDI Input Tape (Section 4 of 5)

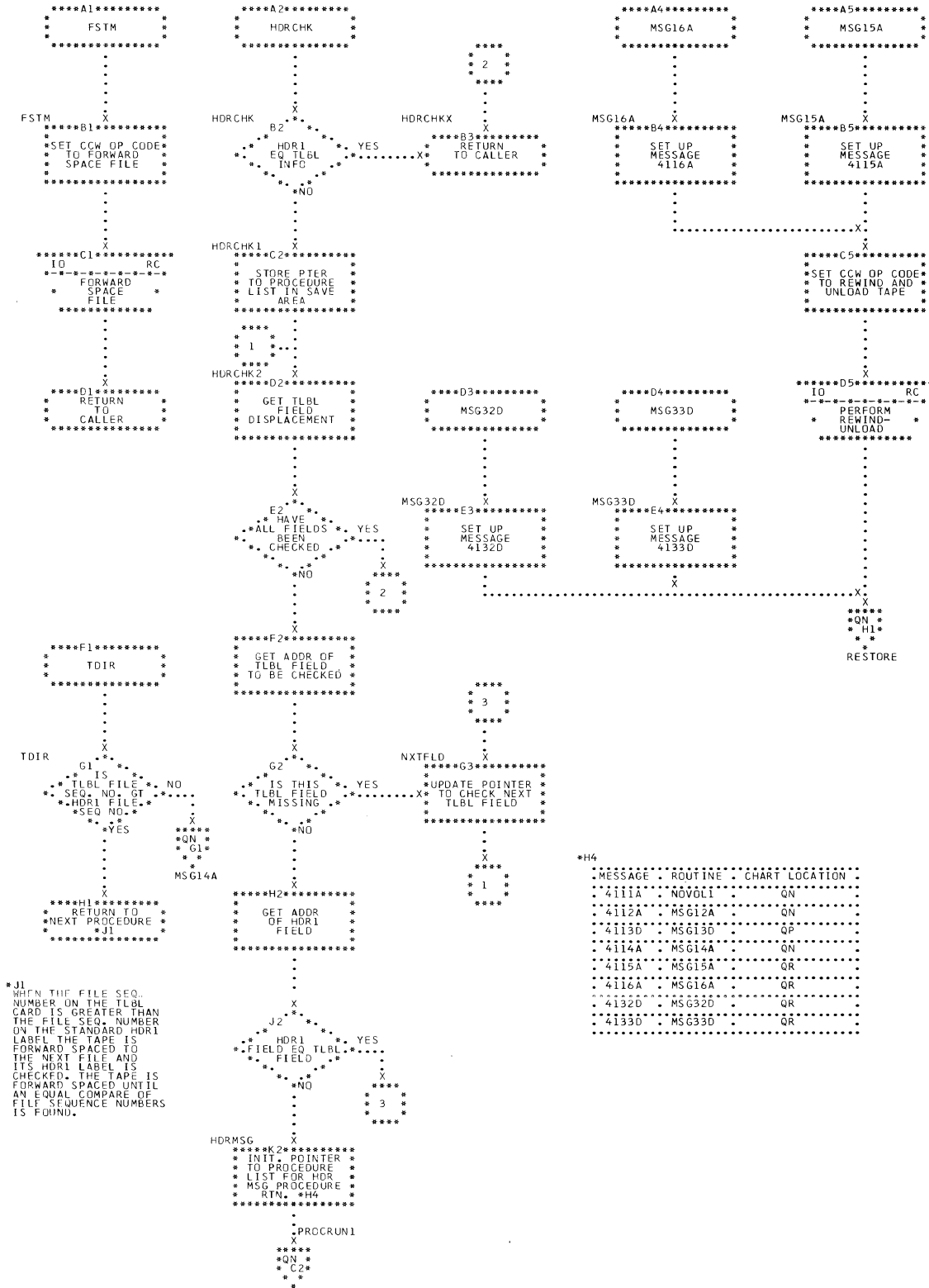


Chart QS. \$\$\$BOCPT1: Open DTFCP and DTFDI Input Tape (Section 5 of 5)

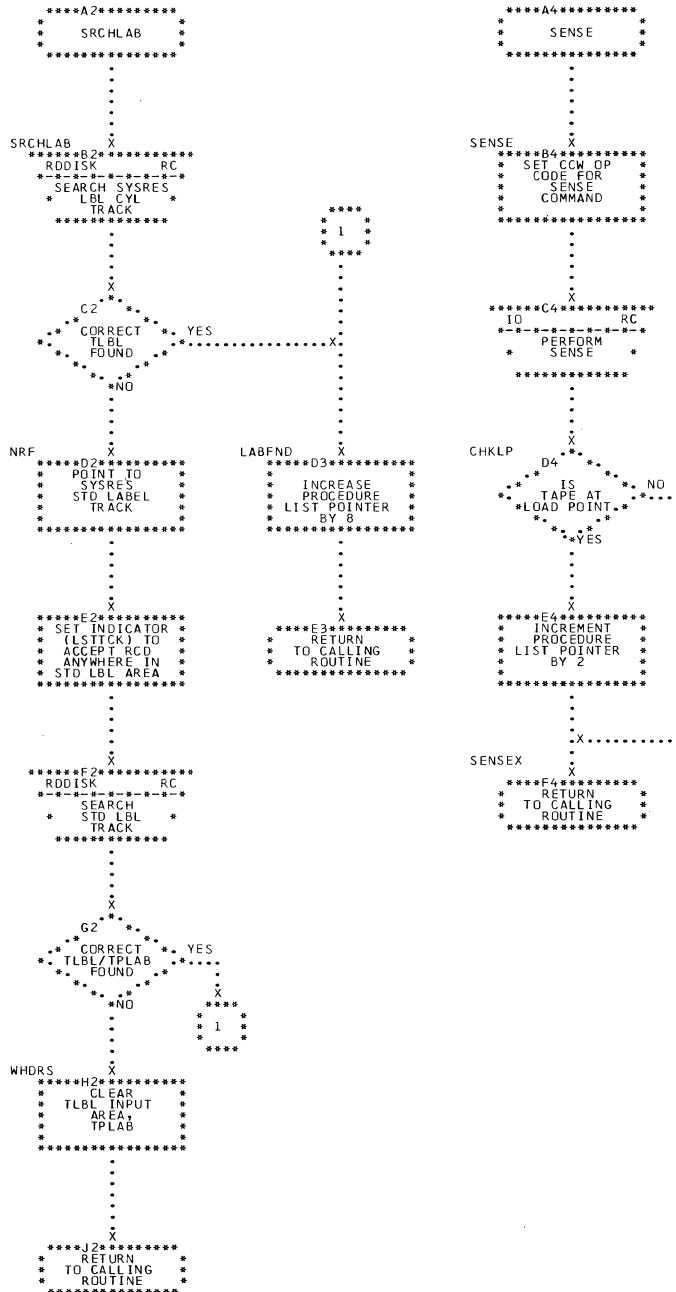


Chart QT. \$\$BOCP2: Open DTFCP and DTFDI Output Tape (Section 1 of 6)

*A1
 ENTERED FROM--
 1. \$\$BOCP1-IF THE
 FILE IS DEFINED
 BY A DISK SYSTEM
 OUTPUT DTFCP AND
 THE DEVICE IS TAPE.
 2. \$\$BOCP2-AFTER
 THE MESSAGE HAS
 BEEN ISSUED AND A
 NON-CANCEL REPLY
 HAS BEEN RECEIVED.

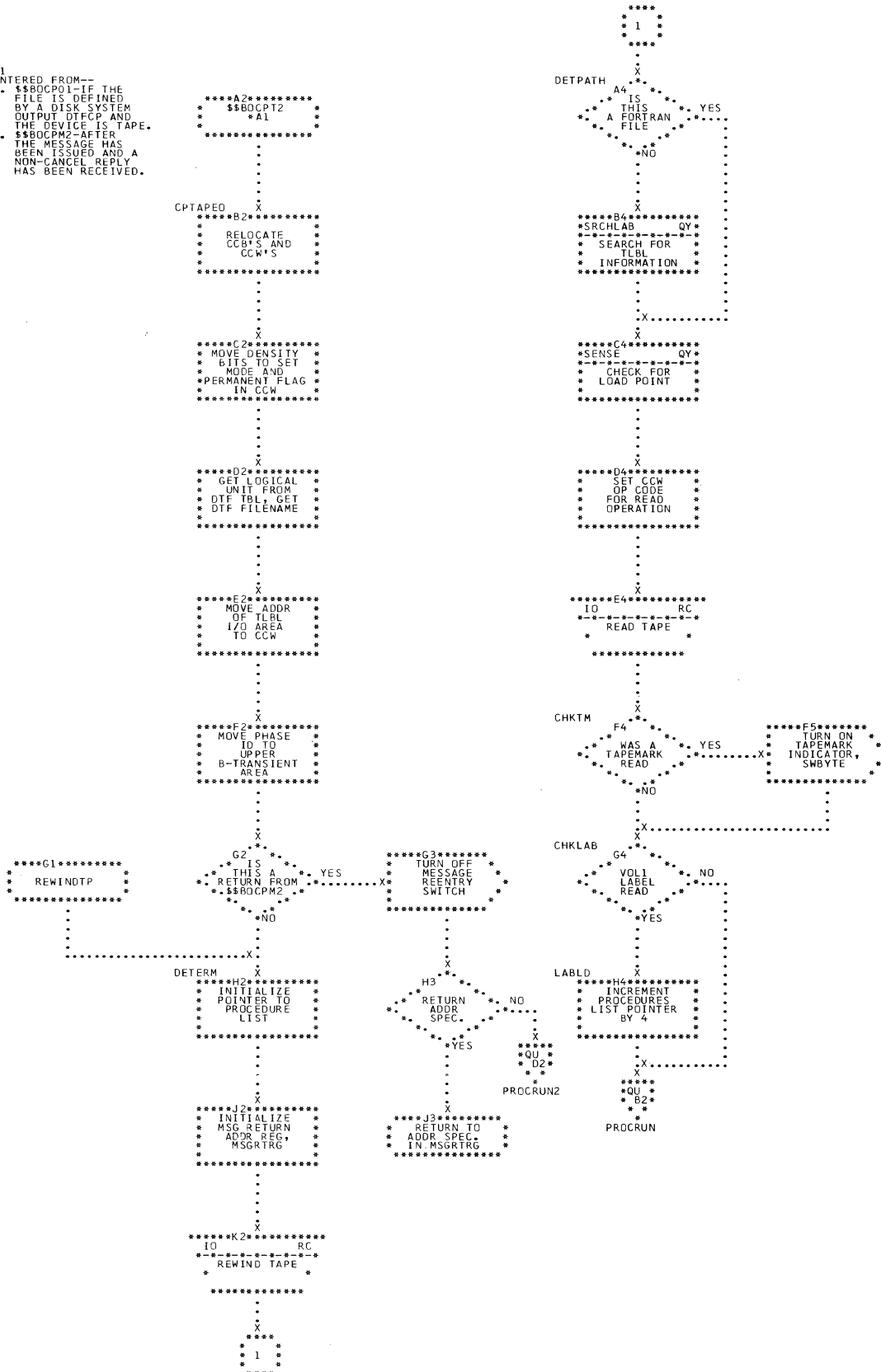
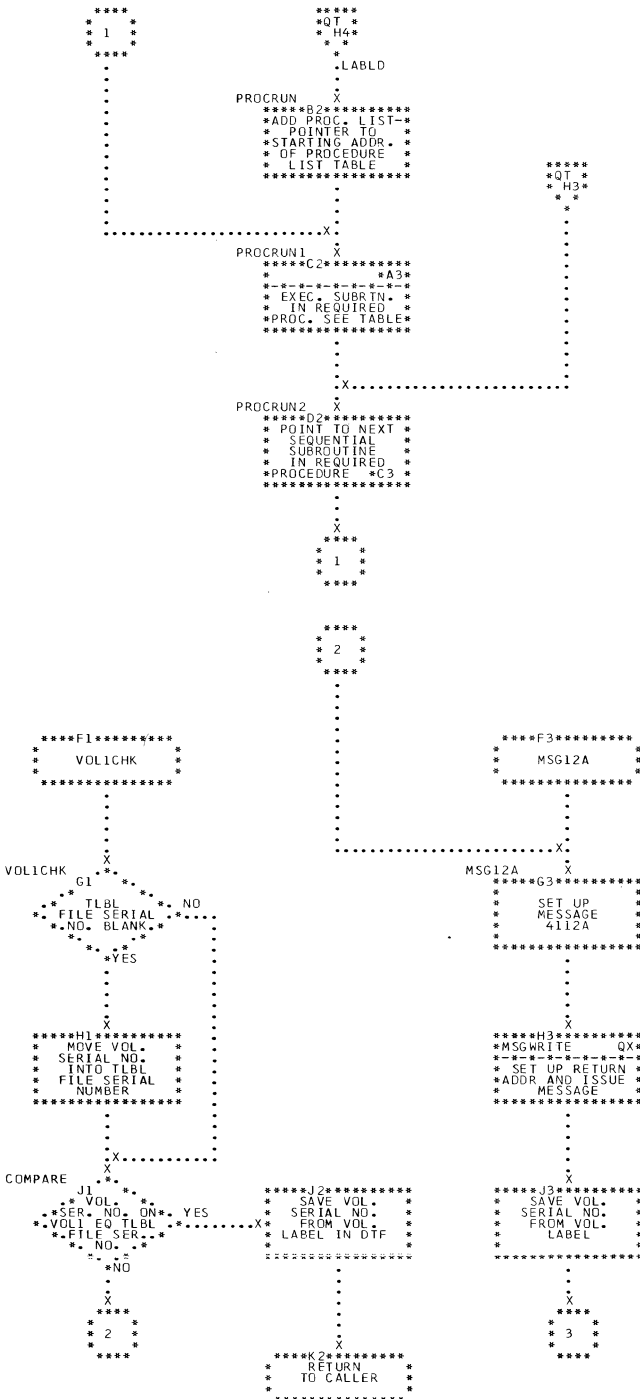


Chart QU. \$\$\$BOCPT2: Open DTFCP and DTFDI Output Tape (Section 2 of 6)



*A3

CONDITION TABLE

Label Information Supplied							
YES				NO			
Tape at Loadpoint				Tape at Loadpoint			
YES		NO		YES		NO	
VOL1 READ	VOLI READ	VOLT READ	VOLI READ	VOLT READ	VOLI READ	VOLT READ	VOLI READ
YES	NO	YES	NO	YES	NO	YES	NO
1	3	2	2	4	5	2	2

Numerics in bottom row reference Procedure List in Table at *C3.

*C3

PROC. LIST	1	2	3	4	5	CHART
NOVOLO		X				QW
VOLICHK	X	X				QU
VOLSKIP	X		X			QV
XPIRCHK	X		X			QV
BKSP	X		X			QV
HDRBLD	X	X				QW
DUMHDRI			X			QW
TMWRITE	X	X	X			QV
EXITLAB	X	X				QX
EXITNLB				X		QX
EXITNLB			X			QX
REWINDTP	X					QT

Chart QV. \$\$BOCPT2: Open DTFCP and DTFDI Output Tape (Section 3 of 6)

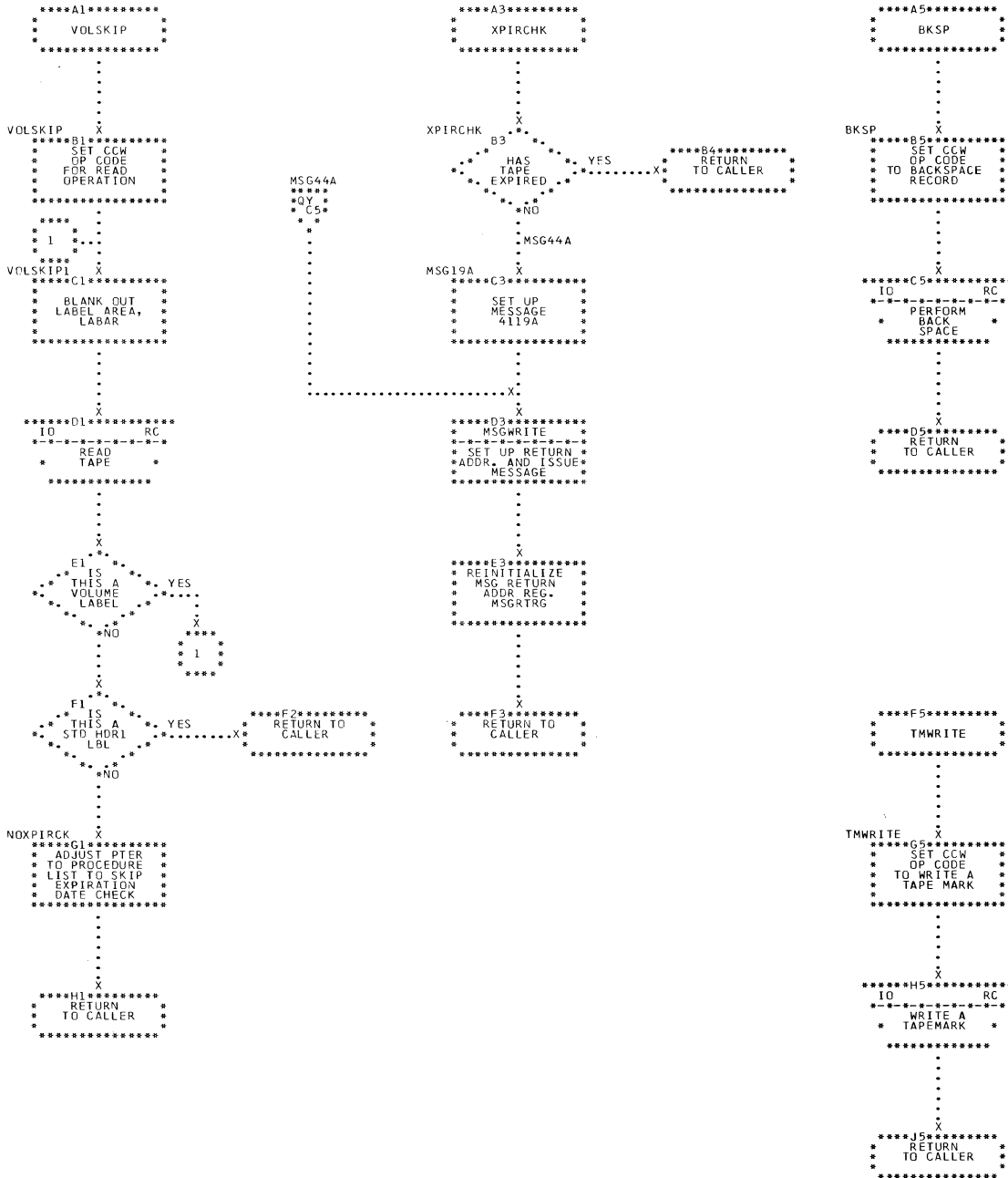


Chart QW. \$\$\$BOCPT2: Open DTFCP and DTFDI Output Tape (Section 4 of 6)

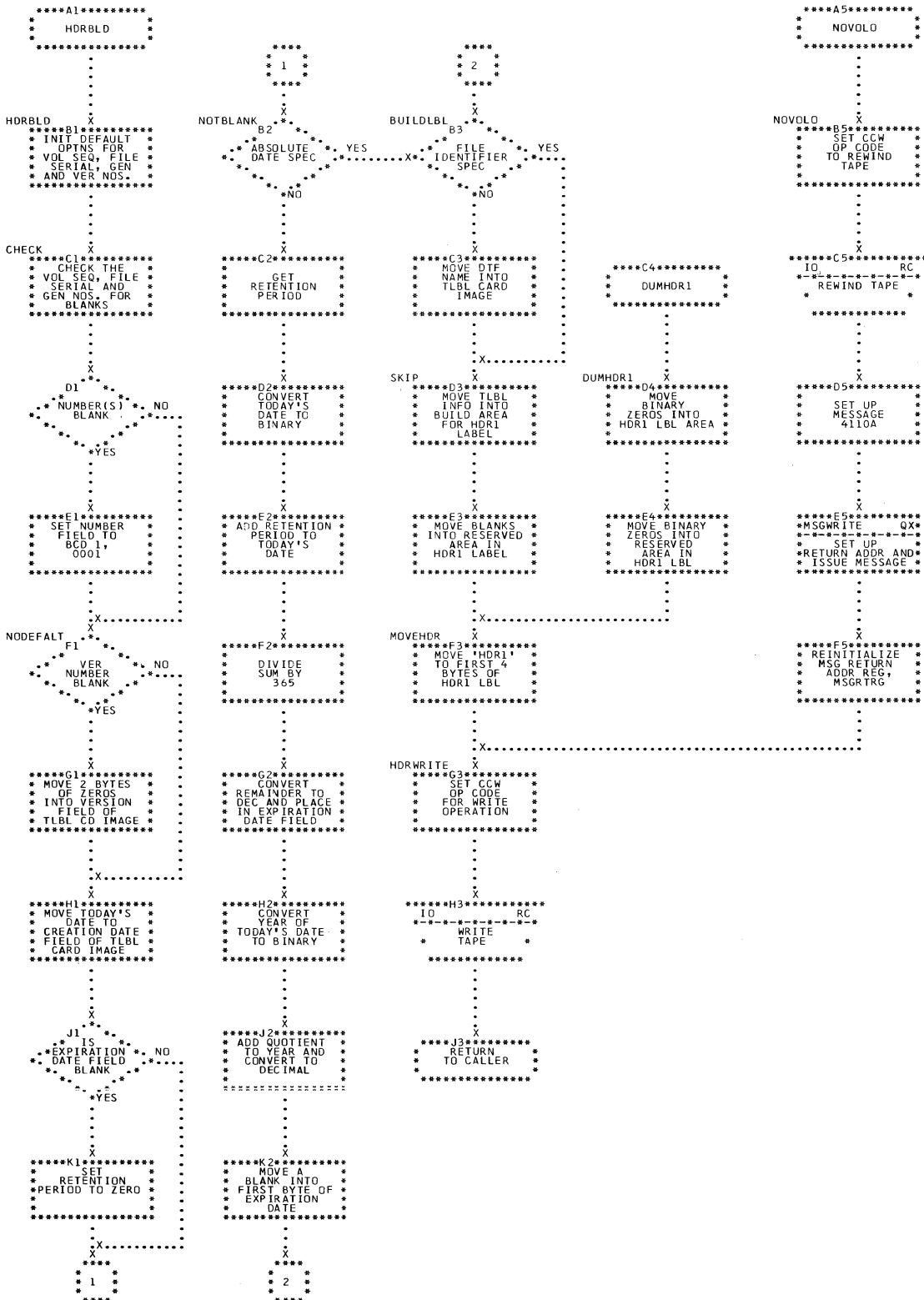
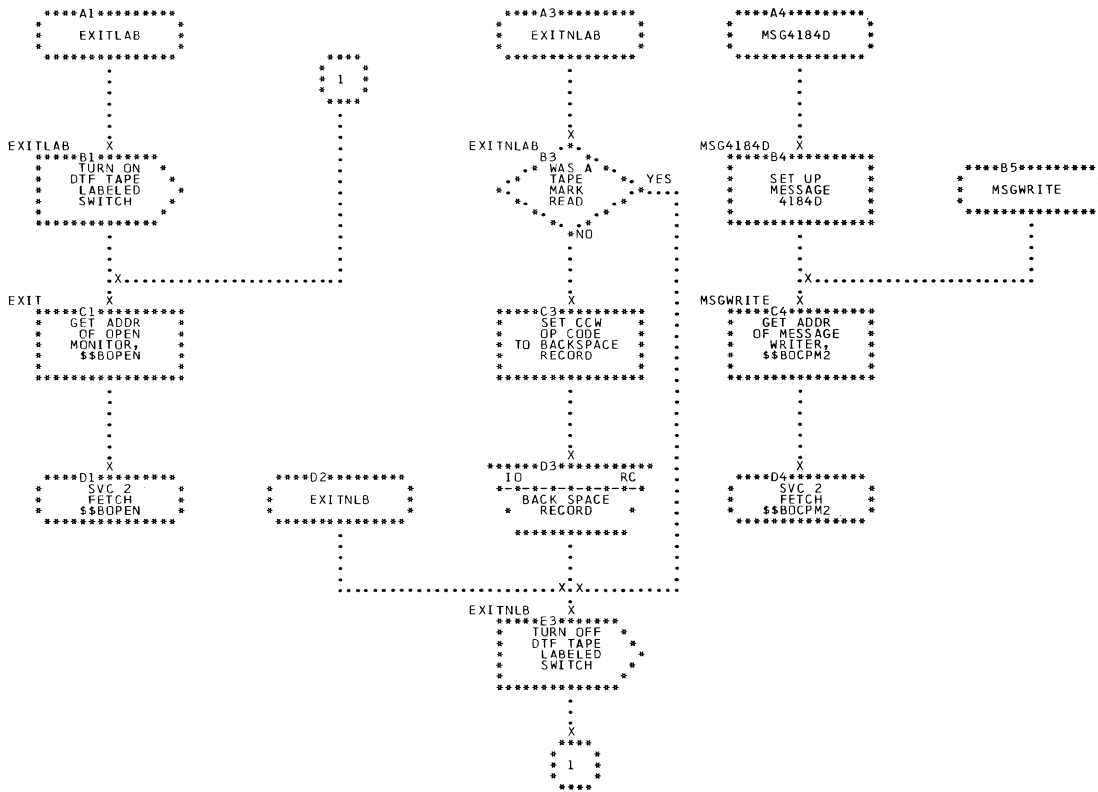


Chart QX. \$\$BOCPT2: Open DTFCP and DTFDI Output Tape (Section 5 of 6)



● Chart QY. \$\$BOCPT2: Open DTFCP and DTFDI Output Tape (Section 6 of 6)

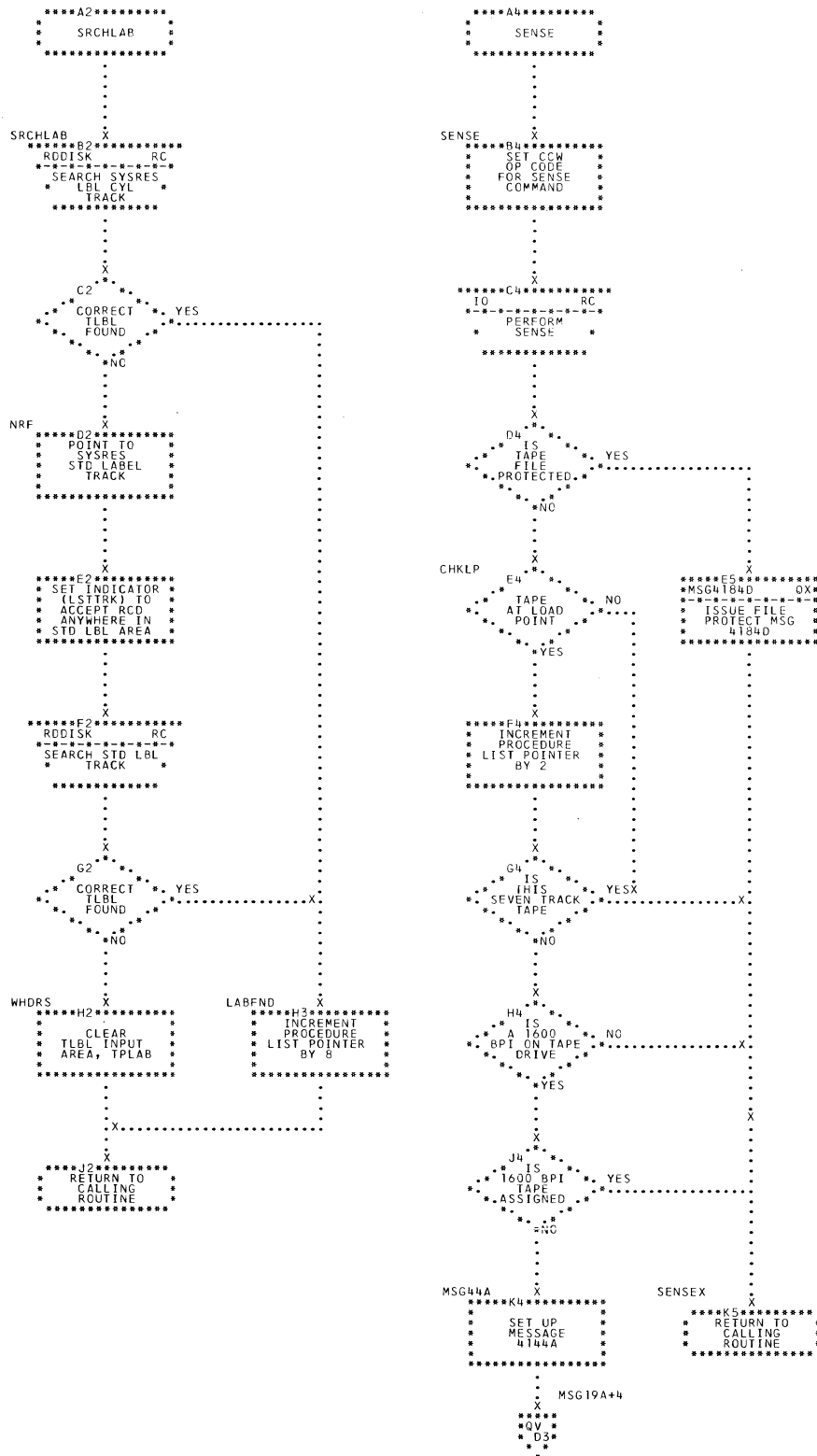


Chart RA. \$\$\$BCCPT1: Close DTFCP and DTFDI Tape Files (Section 2 of 3)

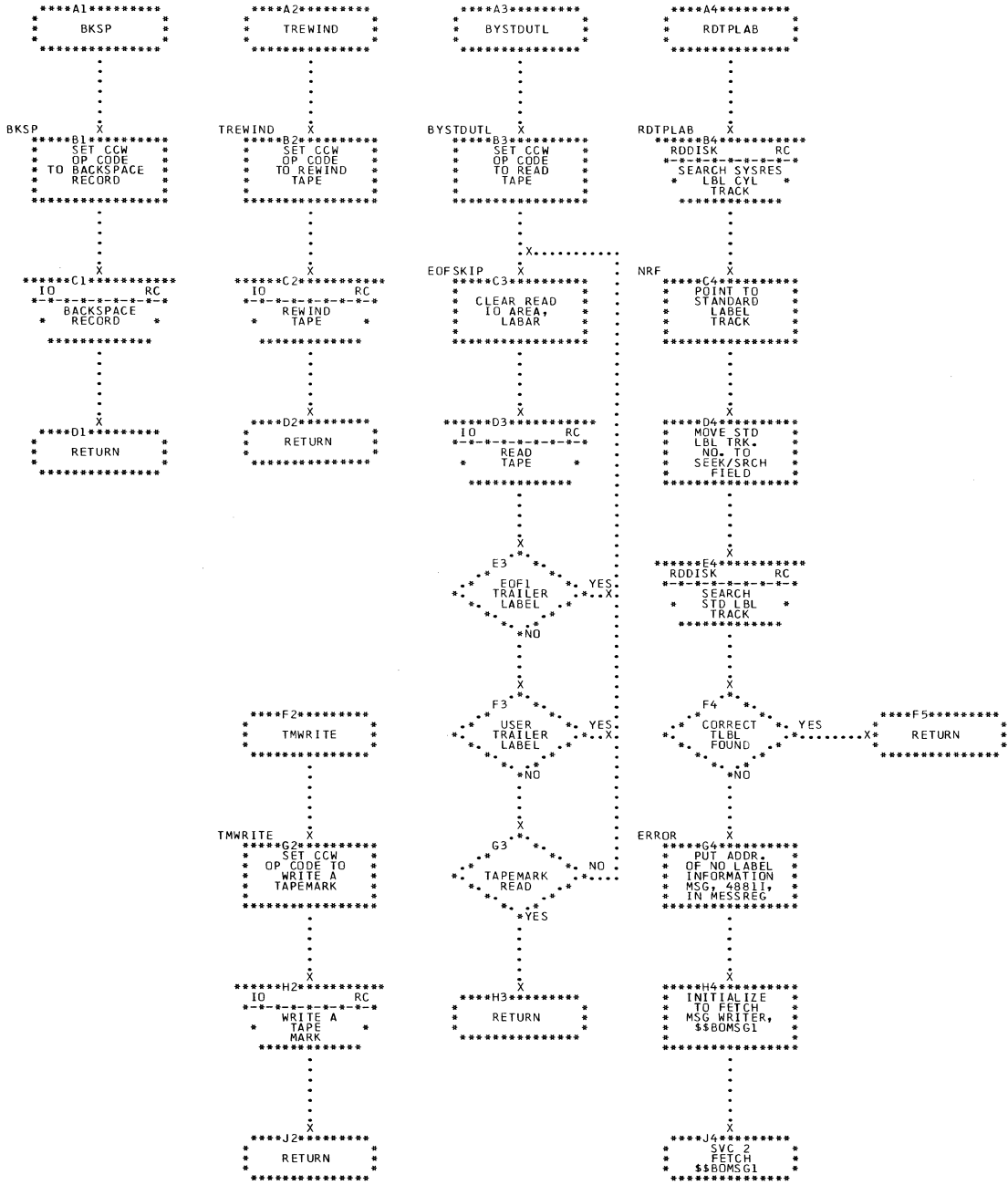


Chart RB. \$\$\$BCCPT1: Close DTFCP and DTFDI Tape Files (Section 3 of 3)

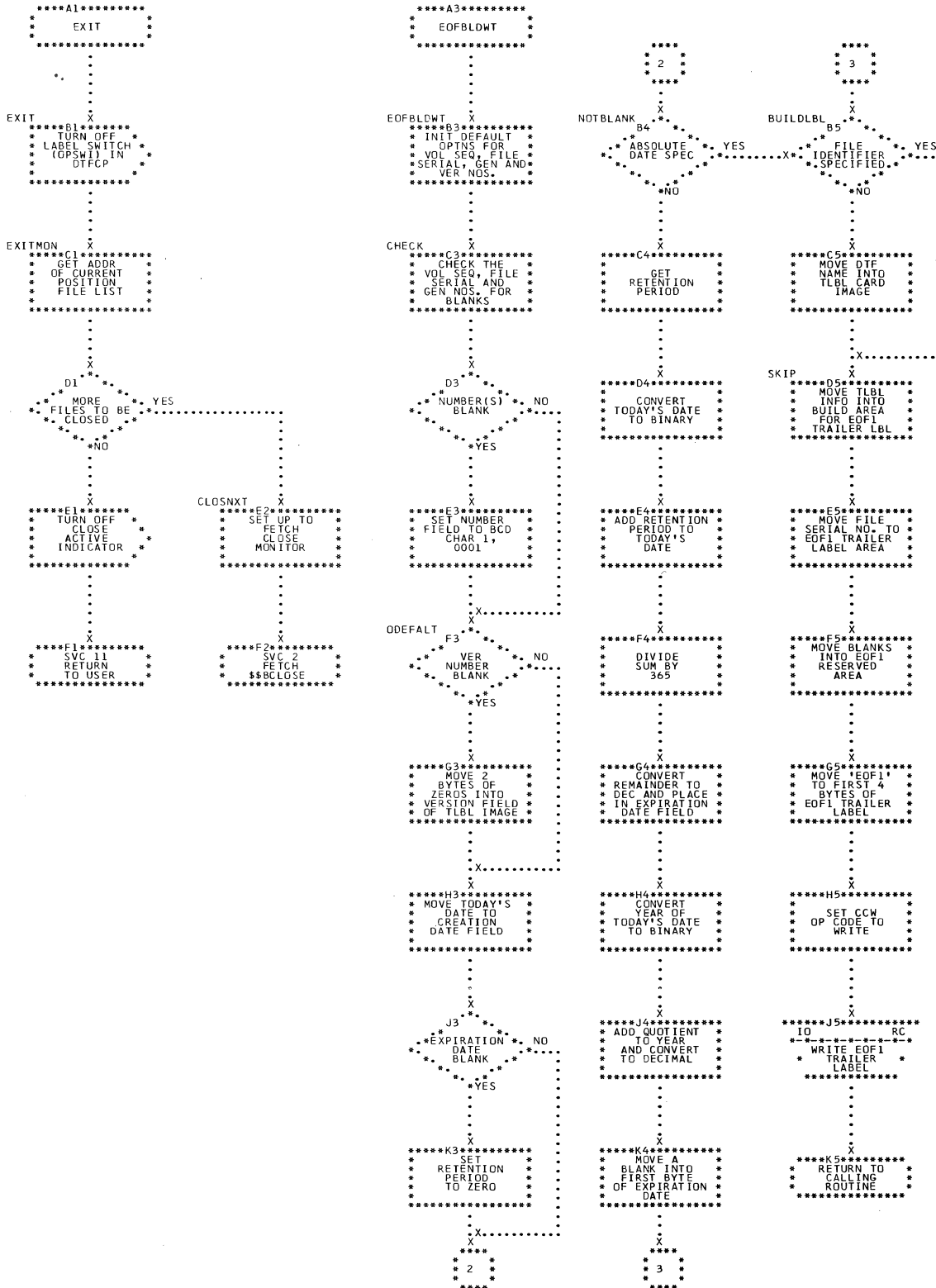


Chart RC. DTFCP and DTFDI: I/O Subroutines

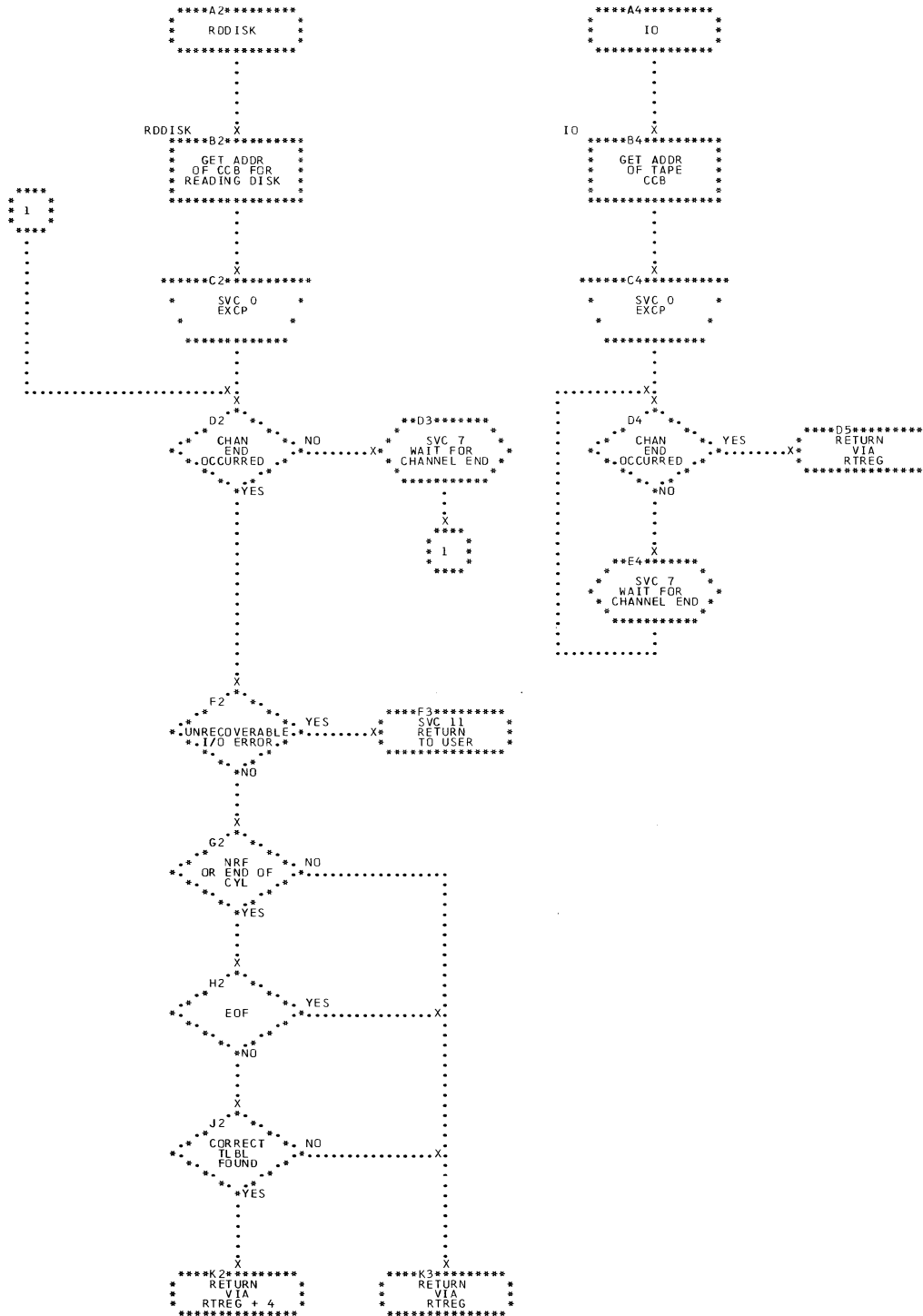


Chart RD. \$\$\$BCLOSP: Punch File Close (Section 1 of 4)

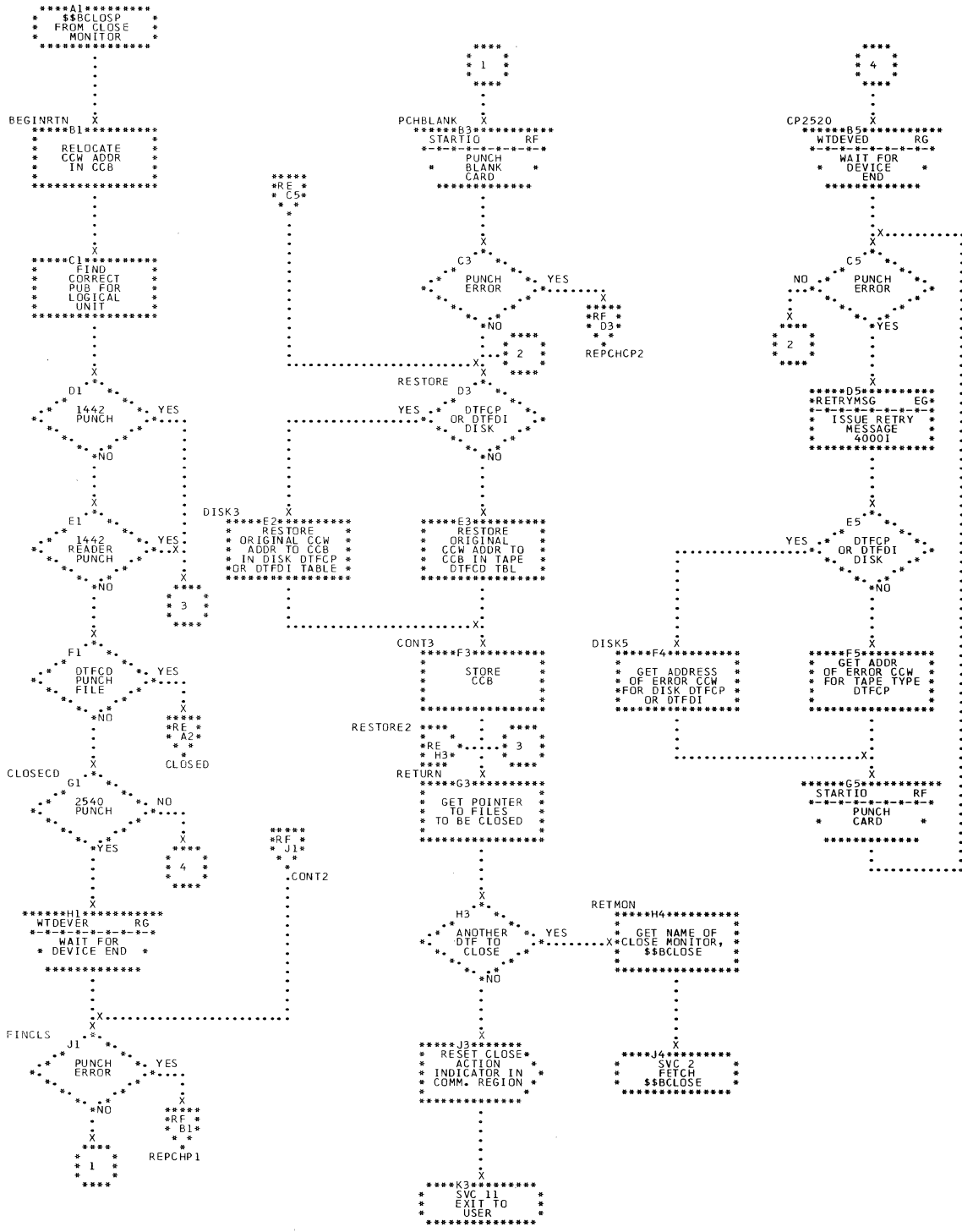


Chart RE. \$\$\$BCLOSP: Punch File Close (Section 2 of 4)

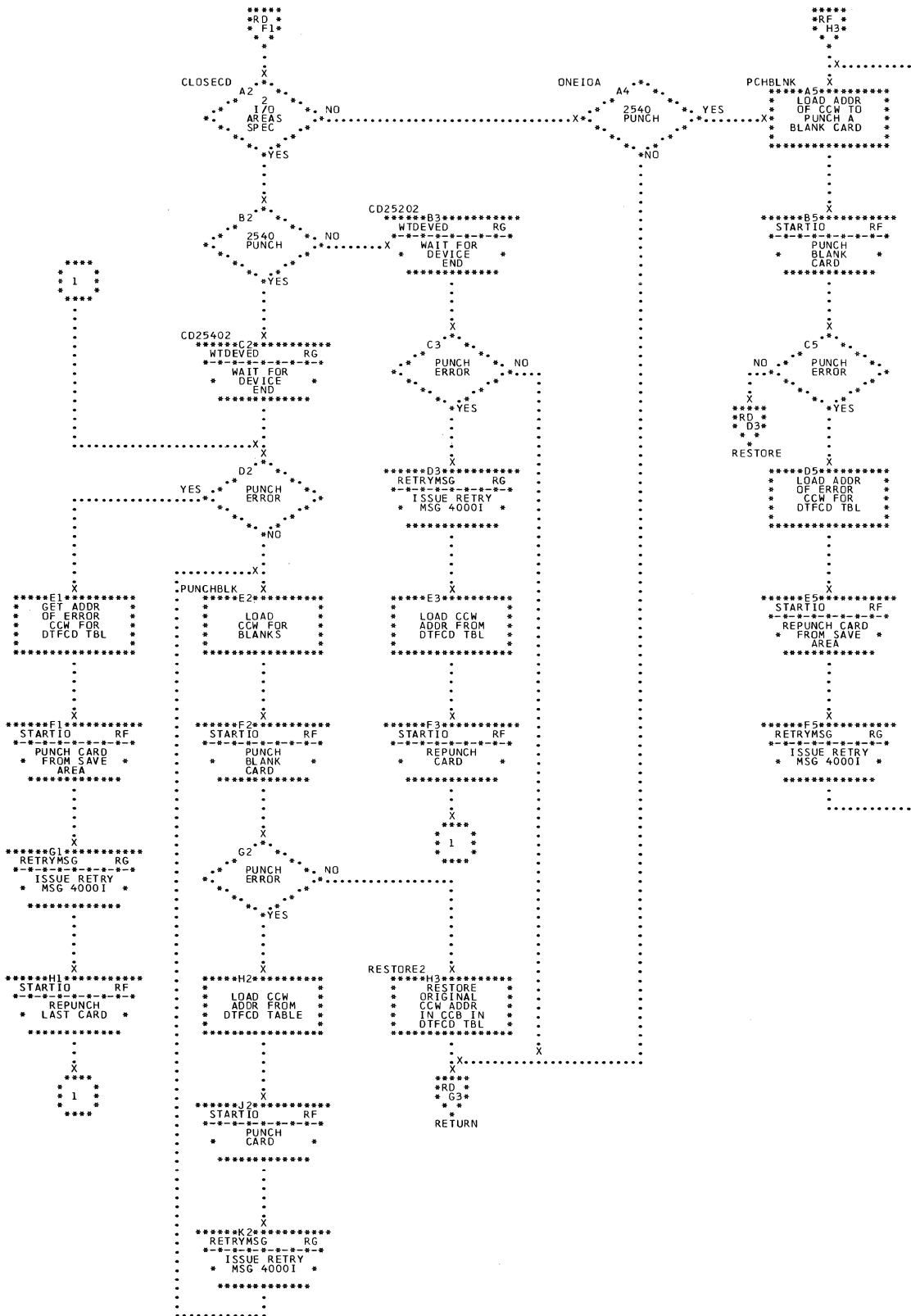


Chart RF. \$\$\$BCLOSP: Punch File Close (Section 3 of 4)

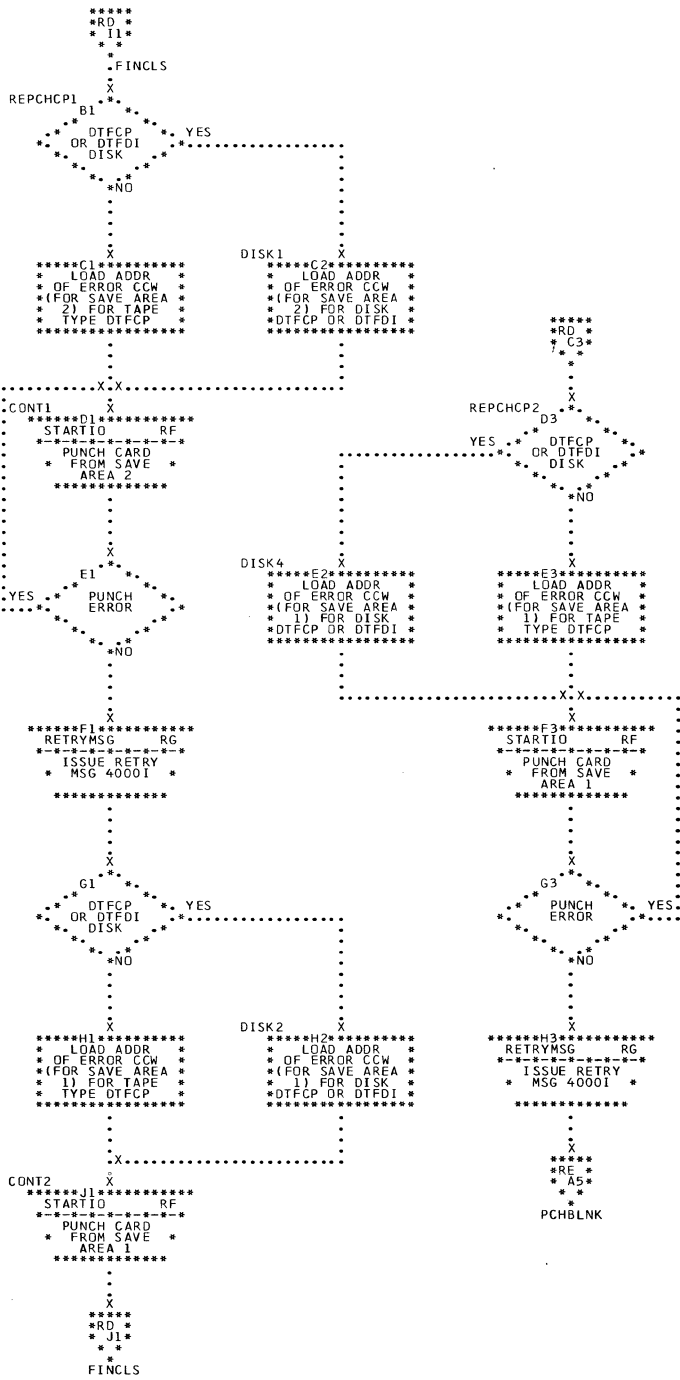


Chart RG. \$\$\$BCLOSP: Punch File Close (Section 4 of 4)

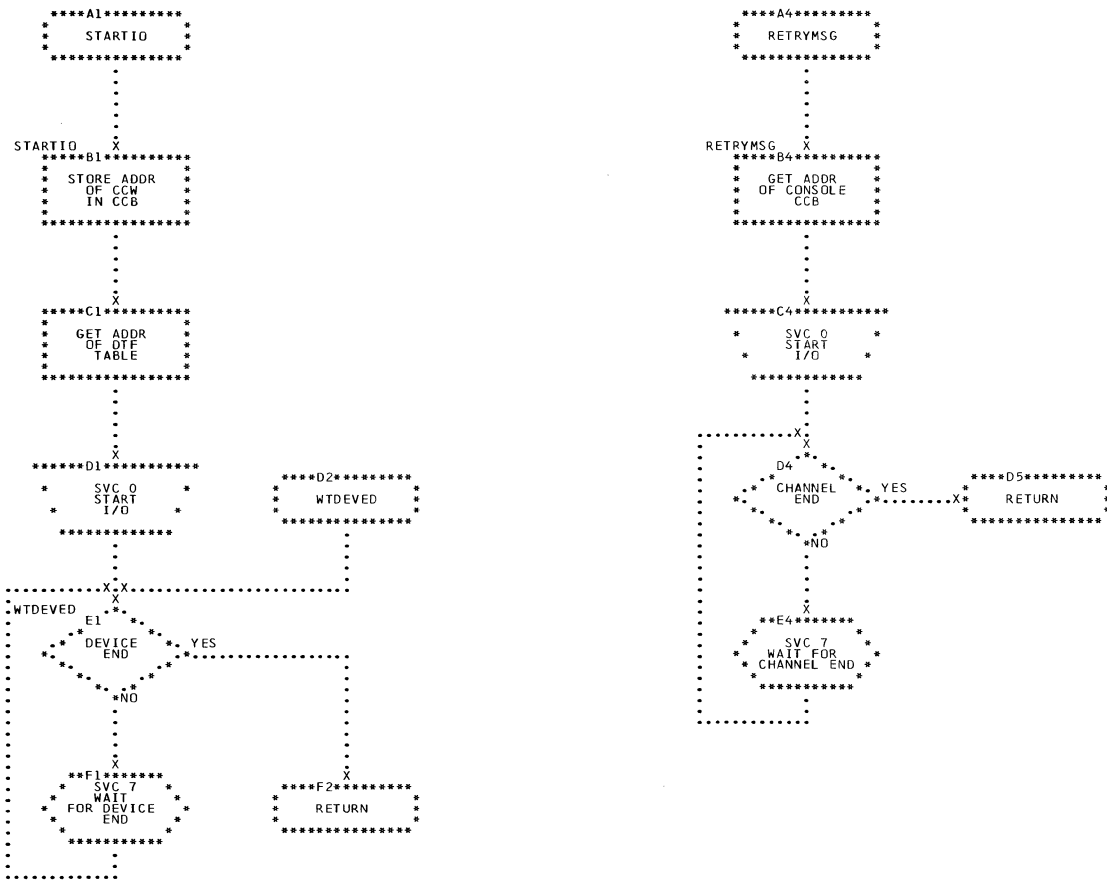


Chart RH. \$\$BOCPM1 and \$\$BOCPM2: DTFCP/DTFDI Message Writers (Section 1 of 3)

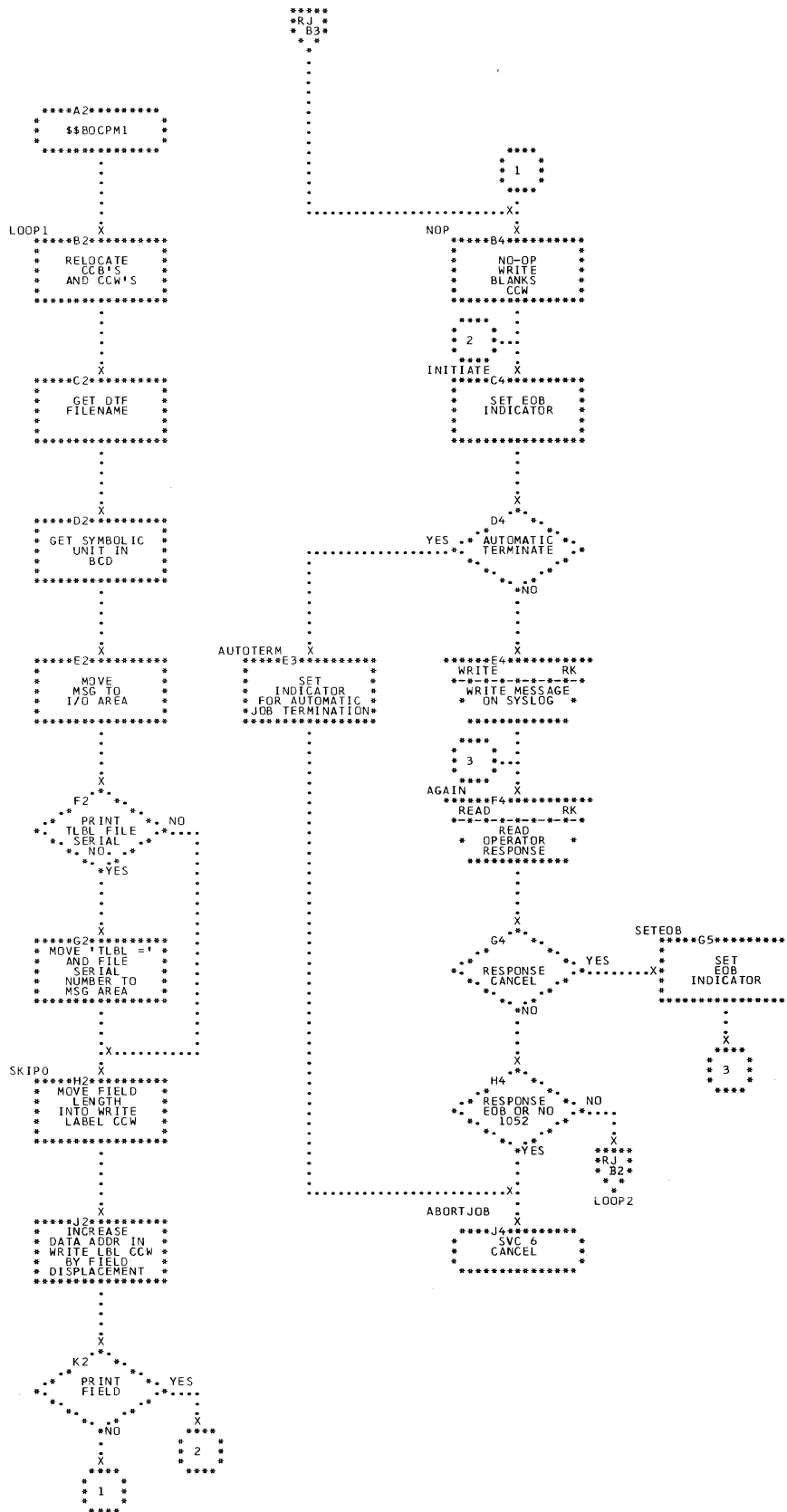
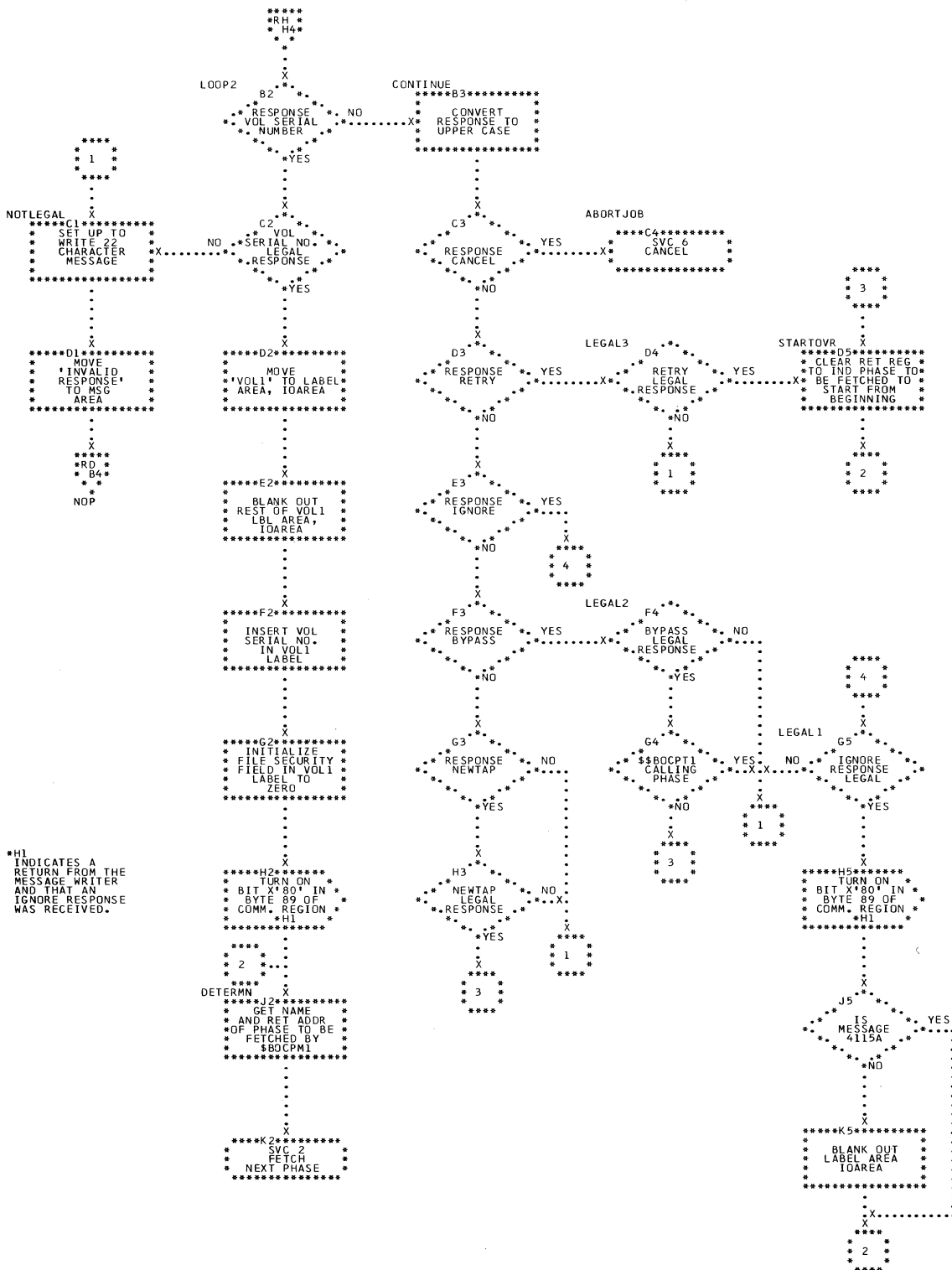


Chart RJ. \$\$BOCPM1 and \$\$BOCPM2: DTFCP/DTFDI Message Writers (Section 2 of 3)



*H1 INDICATES A RETURN FROM THE MESSAGE WRITER AND THAT AN IGNORE RESPONSE WAS RECEIVED.

Chart RK. \$\$BOCPM1 and \$\$BOCPM2: DTFCP/DTFDI Message Writers (Section 3 of 3)

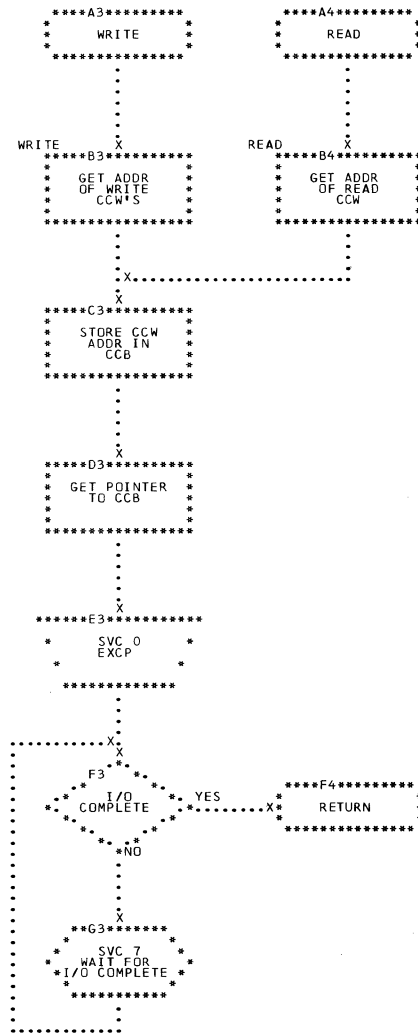


Chart RL. \$\$BOMSG1: Disk Open Error Message Writer, Phase 1 (Section 1 of 2)

*A2
1. \$\$BERRTN
2. \$\$BCCPT1

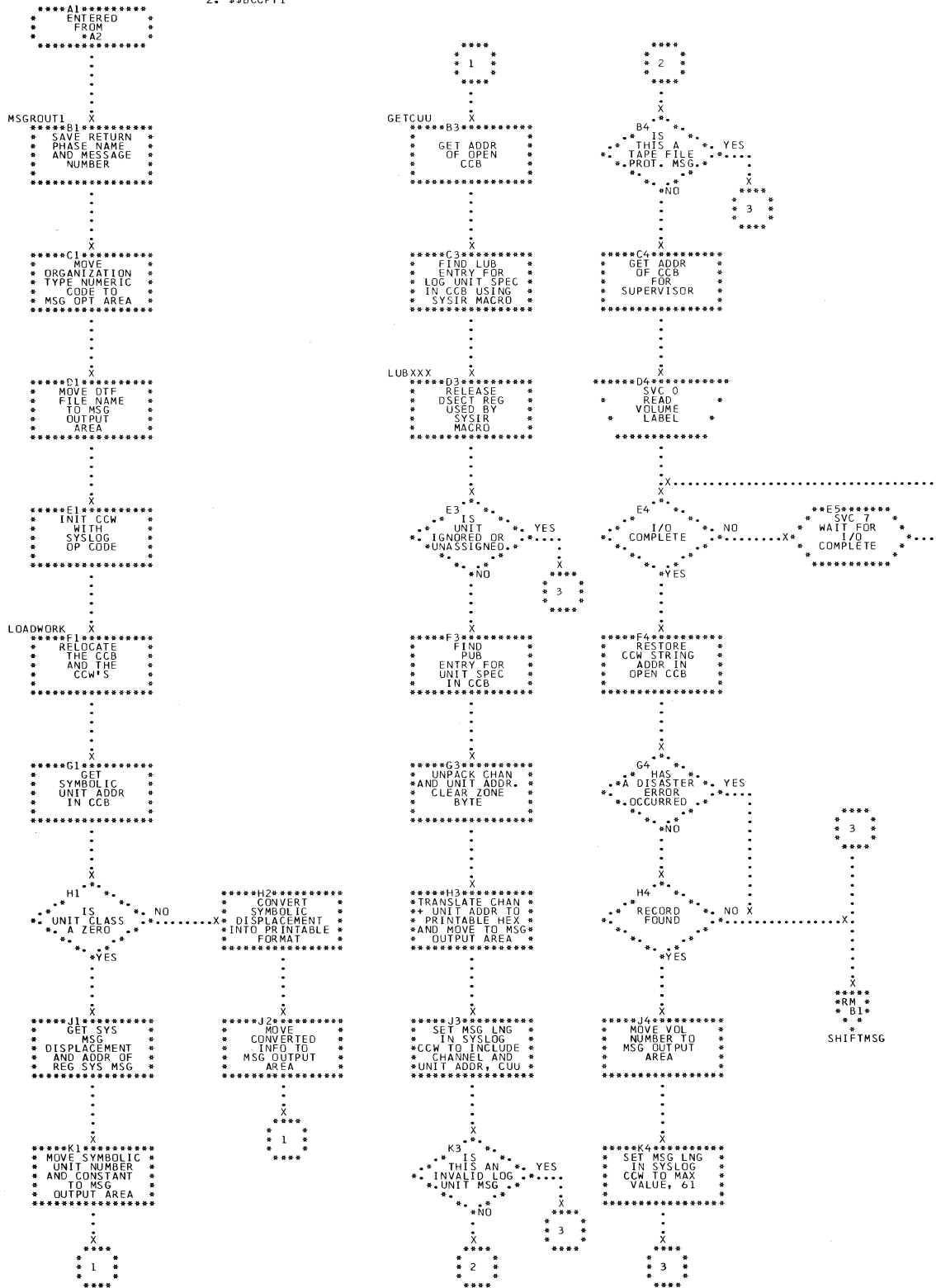
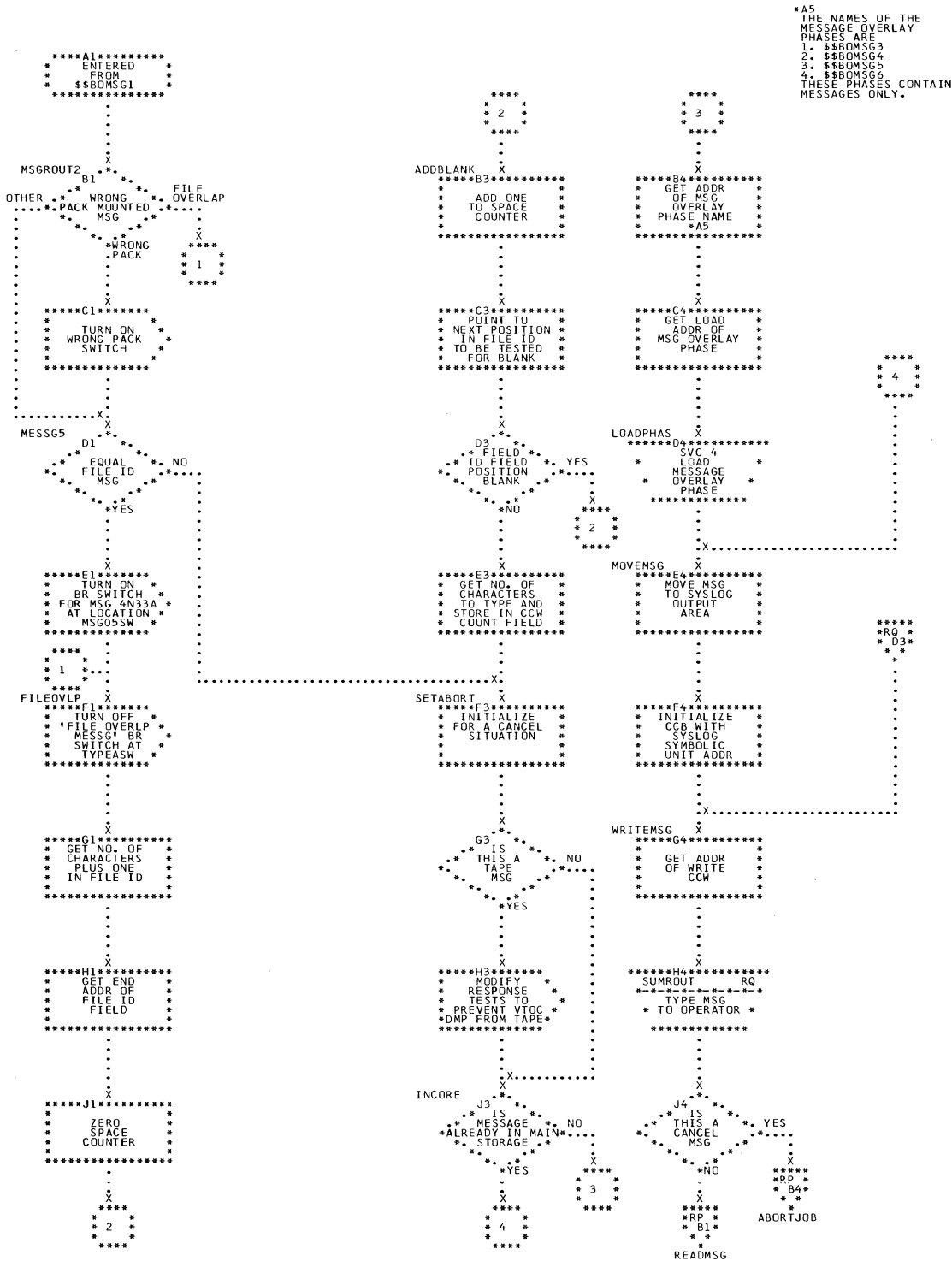


Chart RN. \$\$BOMSG2: Disk Open Error Message Writer, Phase 2 (Section 1 of 3)



APPENDIX A: LABEL CROSS-REFERENCE LIST

This appendix provides additional information for servicing DOS logical IOCS programs. It is primarily intended for the reader with maintenance responsibilities.

The following group of flowchart labels are macro language listing labels and do not appear on assembly listings.

<u>Label</u>	<u>Module or Phase</u>	<u>Chart</u>			
.A4	DIMOD	NA	ADJSYS	\$\$BOCP01	QB
.BK2	PRMOD	FC	ADJSYS	\$\$BOCP11	QH
.BK3	PRMOD	FC	AGAIN	\$\$BMSGWR	LF
.BK4	PRMOD	FE	AGAIN	\$\$BOCPM1	RH
.BK5	PRMOD	FE	AGAIN	\$\$BOCPM2	RH
.P18	PRMOD	FD	AGAIN	\$\$BOMT0M	LE
.P19	PRMOD	FD	AGAIN	\$\$BOMT0W	LE
.P20	PRMOD	FD	AJALT	\$\$BOCP01	QB
.P21	PRMOD	FD	AJALT	\$\$BOCP11	QH
.P22	PRMOD	FD	ATLOADPT	\$\$BOMT01	KA
.P23	PRMOD	FD	AUTOTERM	\$\$BMSGWR	LF
.P24	PRMOD	FD	AUTOTERM	\$\$BOMT0M	LE
.P25	PRMOD	FD	AUTOTERM	\$\$BOMT0W	LE
.P26	PRMOD	FD	BACKSPCE	\$\$BOMT03	KF
.P27	PRMOD	FD	BACKSPC2	\$\$BOMT03	KF
.P28	PRMOD	FD	BEGINRTN	\$\$BCLOSP	RD
.P29	PRMOD	FE	BEGINRTN	\$\$BERRTN	PA
.P30	PRMOD	FE	BKSP	\$\$BCCPT1	RA
.P31	PRMOD	FE	BKSP	\$\$BOCPT2	QV
.P32	PRMOD	FE	BUILDLBL	\$\$BCCPT1	RB
.P33	PRMOD	FE	BUILDLBL	\$\$BOCPT2	QW
.P34	PRMOD	FE	BUILDLBL	\$\$BOMT03	KF
.P35	PRMOD	FE	BYSTDUTL	\$\$BCCPT1	RA
.P36	PRMOD	FC	CALLMON	\$\$BOUR01	AB
.P45	PRMOD	FC	CALLMSG	\$\$BOCP01	QA
.P50	PRMOD	FC	CALLMSG1	\$\$BOCP02	QD
.P51	PRMOD	FC	CALLMSG1	\$\$BOCP02	QE
.P53	PRMOD	FC	CALLMSG1	\$\$BOCP11	QG
.P54	PRMOD	FC	CALLMSG1	\$\$BOCP12	QK
.P55	PRMOD	FD	CALOPN	\$\$BOCP11	QJ
.P56	PRMOD	FD	CALOPN1	\$\$BOCP02	QE
.P61	PRMOD	FC	CALOPN1	\$\$BOCP12	QL
.P62	PRMOD	FC	CBDUMP	\$\$BJCOPT	KN
.P67	PRMOD	FD	CCPTAPE	\$\$BCCPT1	QZ
.P77	PRMOD	FE	CD25202	\$\$BCLOSP	RE
.TW0045	MTMOD	HA	CD25402	\$\$BCLOSP	RE
.TW0047	MTMOD	HA	CHECK	\$\$BCCPT1	RB
.TW045	MTMOD	JG	CHECK	\$\$BCMR01	DH
			CHECK	\$\$BOCPT2	QW
			CHECK	\$\$BOMT02	KD
			CHECKDAT	\$\$BOMT06	KK
			CHECKID	\$\$BOMT01	KB
			CHECKID	\$\$BOMT02	KD
			CHKALC	\$\$BCMT02	KT
			CHKALC	\$\$BCMT07	LC
			CHKIPT	\$\$BJCOPT	KM
			CHKLAB	\$\$BOCPT1	QM
			CHKLAB	\$\$BOCPT2	QT
			CHKLP	\$\$BOCPT1	QS
			CHKLP	\$\$BOCPT2	QY
			CHKMOR	\$\$BOCP01	QB
			CHKMOR	\$\$BOCP02	QE
			CHKMOR	\$\$BOCP11	QH
			CHKMOR	\$\$BOCP12	QL
			CHKPRTR	\$\$BOUR01	AB
			CHKTM	\$\$BOCPT1	QM
			CHKTM	\$\$BOCPT2	QT
			CHKTPM	\$\$BJCOPT	KN

The following flowchart labels are assembly listing labels.

ABORTJOB	\$\$BMSGWR	LF
ABORTJOB	\$\$BOCPM1	RH
ABORTJOB	\$\$BOCPM1	RJ
ABORTJOB	\$\$BOCPM2	RH
ABORTJOB	\$\$BOCPM2	RJ
ABORTJOB	\$\$BOMSG1	RP
ABORTJOB	\$\$BOMT0M	LE
ABORTJOB	\$\$BOMT0W	LE
ADDBLANK	\$\$BOMSG1	RN
ADJINPUT	\$\$BCCPT1	QZ
ADJNOEOF	\$\$BCCPT1	QZ

CKDINPT	\$\$BCCPT1	QZ	DOIO	\$\$BCEOV1	KQ
CKMGTP	\$\$BOCP01	QC	DONTBUMP	\$\$BCMT02	KU
CKMGTP	\$\$BOCP02	QE	DOUBLMSG	\$\$BMSGWR	LF
CKMGTP	\$\$BOCP11	QJ	DTFPHFLE	\$\$BOMT01	KA
CKMGTP	\$\$BOCP12	QL	DUMHDR1	\$\$BOCPT2	QW
CKNXJB	\$\$BCMT02	KT			
CKNXJB	\$\$BCMT07	LC	EOFBLDWT	\$\$BCCPT1	RB
CKPTR	\$\$BOCP01	QC	EOFSKIP	\$\$BCCPT1	RA
CKPTR	\$\$BOCP02	QE	ERRMSG	\$\$BJCOPT	KM
CKPTR	\$\$BOCP12	QL	ERROR	\$\$BCCPT1	RA
			ERROR	\$\$BOCP01	QA
			ERROR	\$\$BOCP02	QD
CKRDR	\$\$BOCP01	QA	ERROR	\$\$BOCP11	QG
CKRDR	\$\$BOCP02	QD	ERROR	\$\$BOCP12	QK
CKRST	\$\$BOCP11	QG	EXC	\$\$BOCP02	QF
CKSYLNK	\$\$BOCP01	QB	EXC	\$\$BOCP12	QL
CKUDSM	\$\$BJCOPT	KM	EXCP	\$\$BCEOV1	KQ
CLEAR	\$\$BOMR01	DG	EXCPRG	\$\$BJCOPT	KN
CLEAROUT	\$\$BOMSG2	RQ	EXEC	\$\$BOCP01	QC
CLOSECD	\$\$BCLOSP	RE			
CLOSECP	\$\$BCLOSP	RD			
CLOS NXT	\$\$BCCPT1	RB	EXEC	\$\$BOCP11	QJ
COBOLWRT	\$\$BOMT04	KG	EXIT	\$\$BCCPT1	RB
COMPARE	\$\$BOCPT2	QU	EXIT	\$\$BOCPT1	QP
COMPARE	\$\$BOMT03	KF	EXIT	\$\$BOCPT2	QX
COMPARE3	\$\$BOMSG1	RM	EXIT	\$\$BOOR01	EY
CONTINUE	\$\$BCMT04	KX	EXITCLOS	\$\$BCMT06	LA
			EXITLAB	\$\$BOCPT1	QP
CONTINUE	\$\$BOCPM1	RJ	EXITLAB	\$\$BOCPT2	QX
CONTINUE	\$\$BOCPM2	RJ	EXITMON	\$\$BCCPT1	RB
CONTINU0	\$\$BOMT01	KB	EXITNLAB	\$\$BOCPT1	QP
CONTINU1	\$\$BOMT01	KB	EXITNLAB	\$\$BOCPT2	QX
CONTINU2	\$\$BOMT01	KB	EXITNLB	\$\$BOCPT1	QP
CONTINU3	\$\$BOMT01	KB	EXITNLB	\$\$BOCPT2	QX
CONTINU4	\$\$BOMT01	KB	EXITRTN	\$\$BOMT06	KK
CONTINU5	\$\$BOMT01	KB			
CONTOPN	\$\$BOUR01	AA	FCHMSG	\$\$BCEOV1	KP
CONT1	\$\$BCLOSP	RF	FETCH	\$\$BOMSG1	RP
CONT1	\$\$BERRTN	PA	FETCH04	\$\$BOMT03	KF
CONT2	\$\$BCLOSP	RF	FILEOVLP	\$\$BOMSG1	RM
CONT2	\$\$BERRTN	PA	FILEOVLP	\$\$BOMSG1	RN
CONT3	\$\$BCLOSP	RD	FINCLS	\$\$BCLOSP	RD
CONT3	\$\$BERRTN	PA	FRMTBL	\$\$BCEOV1	KP
CPDUMP	\$\$BJCOPT	KN	FRSTINST	\$\$BCMR01	DH
CPTAPE	\$\$BOCPT1	QM	FRSTINST	\$\$BMMR20	DJ
CPTAPEO	\$\$BOCPT2	QT	FRSTINST	\$\$BOMR01	DG
CP2520	\$\$BCLOSP	RD	FSTM	\$\$BOCPT1	QR
CUNAD	\$\$BMMR20	DJ			
			GETCUU	\$\$BOMSG1	RL
DEPATH1	\$\$BOCPT1	QM	GETDATE	\$\$BOMT03	KE
DETERM	\$\$BCCPT1	QZ	GETDEFLT	\$\$BOMT03	KE
DETERM	\$\$BOCPT1	QM	GETDSPLY	\$\$BOMSG1	RP
DETERM	\$\$BOCPT2	QT	GETOPN	\$\$BOCP02	QE
DETERMN	\$\$BMSGWR	LF	GETOPN	\$\$BOCP11	QJ
DETERMN	\$\$BOCPM1	RJ	GETOPN	\$\$BOCP12	QL
DETERMN	\$\$BOCPM2	RJ	GOTOIO	\$\$BJCOPT	KK
DETERMN	\$\$BOMT0M	LE			
DETERMN	\$\$BOMT0W	LE	HDRBLD	\$\$BOCPT2	QW
DETPATH	\$\$BCCPT1	QZ	HDRCHK	\$\$BOCPT1	QR
DETPATH	\$\$BOCPT1	QM	HDRCHKX	\$\$BOCPT1	QR
DETPATH	\$\$BOCPT2	QT	HDRCHK1	\$\$BOCPT1	QR
			HDRCHK2	\$\$BOCPT1	QR
			HDRMSG	\$\$BOCPT1	QR
			HDRWRITE	\$\$BOCPT2	QW
DISKERR	\$\$BERRTN	PA			
DISKS	\$\$BOCP01	QA	IGNORE	\$\$BOUR01	AB
DISKS	\$\$BOCP11	QG	IJCEXXXX	CDMOD	BF
DISK1	\$\$BCLOSP	RF	IJCEXXXX	CDMOD	BG
DISK2	\$\$BCLOSP	RF	IJCMBXXX	CDMOD	BA
DISK4	\$\$BCLOSP	RF			
DISK5	\$\$BCLOSP	RD			

IJCMCXXX	CDMOD	BC	IJFFRCK	MTMOD	JF
IJCMCXXX	CDMOD	BD	IJFFRCT	MTMOD	HW
IJCMDXXX	CDMOD	BH	IJFFREL	MTMOD	HQ
IJCMEXXX	CDMOD	BC	IJFFRST	MTMOD	HU
IJCMEXXX	CDMOD	BD	IJFFSTD	MTMOD	HW
IJCMRXXX	CDMOD	BH	IJFFTET	MTMOD	HW
IJCPKXXX	CDMOD	BA	IJFFTI	MTMOD	HW
IJCPNXXX	CDMOD	BF			
IJCPNXXX	CDMOD	BG			
IJCPYXXX	CDMOD	BF	IJFFTIA	MTMOD	HW
IJCPYXXX	CDMOD	BG	IJFFTRU	MTMOD	HR
			IJFFWAI	MTMOD	HY
IJCPZXXX	CDMOD	BE	IJFFWMV	MTMOD	HU
IJCRAXXX	CDMOD	BB	IJFFWRK	MTMOD	HU
IJCRBXXX	CDMOD	BB	IJFFWRK	MTMOD	HU
IJCRTXXX	CDMOD	BE	IJFFXCH	MTMOD	JF
IJCRVXXX	CDMOD	BB	IJFUCNT	MTMOD	HB
IJCR9XXX	CDMOD	BA	IJFUFEF	MTMOD	JE
IJCZ4XXX	CDMOD	BC	IJFUER1	MTMOD	JD
IJCZ4XXX	CDMOD	BD	IJFUER2	MTMOD	JD
IJC00XXX	CDMOD	BB	IJFUEX1	MTMOD	JC
IJDPKXXX	PRMOD	FA	IJFUEX3	MTMOD	HG
IJDPMXXX	PRMOD	FB	IJFUEX5	MTMOD	JC
IJDQ0XXX	PRMOD	FB	IJFUFEE	MTMOD	JE
IJDQ1XXX	PRMOD	FB			
IJE1END	PTMOD	GA	IJFUFE0	MTMOD	HD
IJE1TLST	PTMOD	GA	IJFUFTI	MTMOD	JC
			IJFUGBC	MTMOD	JC
IJE1WLR	PTMOD	GA	IJFUGET	MTMOD	HE
IJE2COMP	PTMOD	GB	IJFUHRU	MTMOD	JC
IJE2END	PTMOD	GB	IJFUHU2	MTMOD	JC
IJE2TLST	PTMOD	GB	IJFIGN	MTMOD	JC
IJE3CLR2	PTMOD	GC	IJFUPUT	MTMOD	HN
IJE3END	PTMOD	GC	IJFURCK	MTMOD	JF
IJE3EOFR	PTMOD	GC	IJFURCT	MTMOD	JC
IJE3EXIT	PTMOD	GC	IJFURDER	MTMOD	JC
IJE3LNXT	PTMOD	GC	IJFURST	MTMOD	HV
IJE3MURM	PTMOD	GC	IJFUWIN	MTMOD	HV
IJE3READ	PTMOD	GC	IJFUWMV	MTMOD	HV
IJE3RRED	PTMOD	GD	IJFUWRK	MTMOD	HV
IJE3STSD	PTMOD	GC			
IJE3STR	PTMOD	GC	IJFUXCH	MTMOD	JF
IJE4CLR2	PTMOD	GE	IJFVBDB	MTMOD	HG
			IJFVCNT	MTMOD	HB
IJE4EXIT	PTMOD	GE	IJFVDBL	MTMOD	HT
IJE4STSD	PTMOD	GE	IJFVEFE	MTMOD	JA
IJE4STR	PTMOD	GE	IJFVER1	MTMOD	HF
IJFFCNT	MTMOD	HB	IJFVER2	MTMOD	JA
IJFFCON	MTMOD	HW	IJFVEX1	MTMOD	HZ
IJFFDBL	MTMOD	HT	IJFVEX3	MTMOD	HF
IJFFEFE	MTMOD	HY	IJFVFEE	MTMOD	JA
IJFFER1	MTMOD	HX	IJFVFEO	MTMOD	HD
IJFFER2	MTMOD	HX	IJFVFTI	MTMOD	JB
IJFFEX1	MTMOD	HW	IJFVGBC	MTMOD	HZ
IJFFEX1A	MTMOD	HW	IJFVGET	MTMOD	HE
IJFFEX3	MTMOD	HE	IJFVGTB	MTMOD	HF
IJFFEE	MTMOD	HY			
IJFFEO	MTMOD	HD	IJFVGTR	MTMOD	HF
IJFFGBC	MTMOD	HW	IJFVHRU	MTMOD	HZ
			IJFVHU2	MTMOD	HZ
IJFFGET	MTMOD	HE	IJFVIG1	MTMOD	HZ
IJFFHRU	MTMOD	HW	IJFVIG2	MTMOD	HF
IJFFHU2	MTMOD	HW	IJFVMAI	MTMOD	HZ
IJFFIGN	MTMOD	HW	IJFVMOV	MTMOD	HU
IJFFMAI	MTMOD	HW	IJFVNOIF	MTMOD	JA
IJFFNOIF	MTMOD	HY	IJFVOEX	MTMOD	HU
IJFFPUT	MTMOD	HM	IJFVOUB	MTMOD	HM
IJFFRBK	MTMOD	HY	IJFVPUT	MTMOD	HM

IJFVRCK	MTMOD	JF	IJJCPST	CPMOD	MA
IJFVRCT	MTMOD	HZ	IJJCPST	CPMOD	MD
IJFVREL	MTMOD	HQ	IJJCPSTO	CPMOD	MJ
IJFVRST	MTMOD	HU	IJJCPSTOV	CPMOD	MD
			IJJCPUNX	CPMOD	MD
IJFVSTW	MTMOD	HE	IJJCPWDE	CPMOD	ME
IJFVTOV	MTMOD	HZ	IJJCPWDE	CPMOD	MG
IJFVTRU	MTMOD	HR	IJJCPWDE	CPMOD	MJ
IJFVTST	MTMOD	HG	IJJC1RT	CPMOD	MG
IJFVUBL	MTMOD	HU	IJJC1RT	CPMOD	MJ
IJFVWRK	MTMOD	HU	IJJC2SLO	CPMOD	MB
IJFVXCH	MTMOD	JF	IJJERRTN	CPMOD	ME
IJFWCHEK	MTMOD	HA	IJJEXEC	CPMOD	MA
IJFWCOMP	MTMOD	JG	IJJE1XXX	DIMOD	NA
IJFWCTL	MTMOD	HC	IJJE1XXX	DIMOD	NE
IJFWEXCP	MTMOD	JG			
IJFWEXIT	MTMOD	HA			
IJFWFSF	MTMOD	JG	IJJE1XXX	DIMOD	NH
IJFWPNTR	MTMOD	HJ	IJJE1XXX	DIMOD	NJ
IJFWPNTW	MTMOD	HL	IJJE5XXX	DIMOD	NC
			IJJE5XXX	DIMOD	NH
IJFWPRCS	MTMOD	HA	IJJE5XXX	DIMOD	NJ
IJFWREAD	MTMOD	HP	IJJE6XXX	DIMOD	NE
IJFWRSME	MTMOD	HA	IJJF3XXX	DIMOD	NF
IJFWRTRN	MTMOD	HA	IJJF3XXX	DIMOD	NJ
IJFWRW	MTMOD	JG	IJJF5XXX	DIMOD	NC
IJFWSBST	MTMOD	HA	IJJF5XXX	DIMOD	NH
IJFWSPNT	MTMOD	HK	IJJF5XXX	DIMOD	NJ
IJFWSTOR	MTMOD	JG	IJJF8XXX	DIMOD	NA
IJFWTEST	MTMOD	HJ	IJJF8XXX	DIMOD	NF
IJFWTEST	MTMOD	HL	IJJF9XXX	DIMOD	NF
IJFWTIXT	MTMOD	HJ	IJJG1XXX	DIMOD	NE
IJFWTIXT	MTMOD	HL			
IJFWTRC	MTMOD	JG	IJJG3XXX	DIMOD	NA
IJFWWRIT	MTMOD	HS	IJJG3XXX	DIMOD	NK
IJFXBK	MTMOD	HQ	IJJG6XXX	DIMOD	NE
			IJJG8XXX	DIMOD	NE
IJJCAL2	CPMOD	MF	IJJH4XXX	DIMOD	NE
IJJCAL2	CPMOD	MH	IJJH5XXX	DIMOD	NE
IJJCAL2	CPMOD	MK	IJJH8XXX	DIMOD	NC
IJJCEXEC	CPMOD	MD	IJJH8XXX	DIMOD	NK
IJJCFOND	CPMOD	ME	IJJI0XXX	DIMOD	NA
IJJCFOND	CPMOD	MG	IJJI0XXX	DIMOD	NC
IJJCFOND	CPMOD	MJ	IJJI0XXX	DIMOD	NF
IJJCNDSK	CPMOD	MB	IJJI0XXX	DIMOD	NJ
IJJCNDSK	CPMOD	MC	IJJJ1XXX	DIMOD	NA
IJJCNOEX	CPMOD	MB	IJJJ1XXX	DIMOD	NC
IJJCNOEX	CPMOD	MC	IJJJ1XXX	DIMOD	NF
IJJCPCAL	CPMOD	ME			
IJJCPCAL	CPMOD	MG	IJJJ1XXX	DIMOD	NJ
IJJCPCAL	CPMOD	MJ	IJJJ6XXX	DIMOD	NH
IJJCPDSK	CPMOD	MB	IJJJ7XXX	DIMOD	NH
			IJJJ7XXX	DIMOD	NJ
IJJCPDSK	CPMOD	MC	IJJJ8XXX	DIMOD	NA
IJJCPDSK	CPMOD	MF	IJJJ8XXX	DIMOD	ND
IJJCPDSK	CPMOD	MH	IJJJ8XXX	DIMOD	NF
IJJCPDSK	CPMOD	MK	IJJJ8XXX	DIMOD	NH
IJJCPGP	CPMOD	MA	IJJJ8XXX	DIMOD	NJ
IJJCPGP	CPMOD	MD	IJJK5XXX	DIMOD	NB
IJJCPGP	CPMOD	MG	IJJK5XXX	DIMOD	ND
IJJCPIO	CPMOD	MB	IJJK5XXX	DIMOD	NG
IJJCPIO	CPMOD	MC	IJJK5XXX	DIMOD	NK
IJJCPIO	CPMOD	ME	IJJK6XXX	DIMOD	NB
IJJCPIO	CPMOD	MG	IJJK6XXX	DIMOD	ND
IJJCPLUP	CPMOD	ME			
IJJCPLUP	CPMOD	MG	IJJK6XXX	DIMOD	NK
IJJCPSLD	CPMOD	MD	IJJK8XXX	DIMOD	NB
IJJCPSLD	CPMOD	MD	IJJK8XXX	DIMOD	ND

IJKK8XXX	DIMOD	NG	IJMR5XXX	ORMOD	ES
IJKK8XXX	DIMOD	NK	IJMR8XXX	ORMOD	ES
IJL5XXX	DIMOD	NC	IJMR9XXX	ORMOD	ET
IJL5XXX	DIMOD	NH	IJMSAXXX	ORMOD	EH
IJL5XXX	DIMOD	NJ	IJMSAXXX	ORMOD	EN
IJJMOV	CPMOD	ME	IJMSAXXX	ORMOD	EX
IJJMOV	CPMOD	MG	IJMSBXXX	ORMOD	EC
IJJMOV	CPMOD	MJ	IJMSBXXX	ORMOD	EH
IJJNOSK	CPMOD	ME	IJMSBXXX	ORMOD	EN
IJJNOSK	CPMOD	MG	IJMSBXXX	ORMOD	EX
IJJNOSK	CPMOD	MJ	IJMSBXXX	ORMOD	EC
IJJPWT	CPMOD	MG			
			IJMSCXXX	ORMOD	EH
IJJPWT	CPMOD	MJ	IJMSCXXX	ORMOD	EN
IJJRTRN	CPMOD	MA	IJMSCXXX	ORMOD	EX
IJJRTRN	CPMOD	MD	IJMSKXXX	ORMOD	EA
IJJSAVE	CPMOD	ME	IJMSQXXX	ORMOD	EE
IJJSAVE	CPMOD	MG	IJMSQXXX	ORMOD	EL
IJJSAVE	CPMOD	MJ	IJMSQXXX	ORMOD	EU
IJJTEST	CPMOD	MA	IJM T A X X X	ORMOD	EE
IJJUNITX	CPMOD	MG	IJM T A X X X	ORMOD	EU
IJJUNITX	CPMOD	MJ	IJM T A X X X	ORMOD	EL
IJJWCHED	CPMOD	MF	IJM T B X X X	ORMOD	EE
IJJWCHED	CPMOD	MH	IJM T B X X X	ORMOD	EV
IJJWCHED	CPMOD	MK	IJM T C X X X	ORMOD	EF
IJJWINX	CPMOD	MB	IJM T C X X X	ORMOD	EM
IJJWINX	CPMOD	MC	IJM T D X X X	ORMOD	EF
IJJ2SLO	CPMOD	MC	IJM T D X X X	ORMOD	EM
IJJ48XXX	DIMOD	NA	IJM T E X X X	ORMOD	EF
IJMA B X X X	ORMOD	EU	IJM T E X X X	ORMOD	EM
IJMA1XXX	ORMOD	EJ	IJM T F X X X	ORMOD	EF
IJMA1XXX	ORMOD	ER	IJM T F X X X	ORMOD	EM
IJMA2XXX	ORMOD	EE	IJM T F X X X	ORMOD	EW
IJMA2XXX	ORMOD	EE	IJM T I X X X	ORMOD	EG
IJMA B X X X	ORMOD	EK	IJM T J X X X	ORMOD	EE
IJMA D X X X	ORMOD	EK	IJM T K X X X	ORMOD	EE
IJMA E X X X	ORMOD	EJ	IJM T L X X X	ORMOD	EE
IJMA F X X X	ORMOD	EJ	IJM T N X X X	ORMOD	EG
IJMA I X X X	ORMOD	EJ	IJM T O X X X	ORMOD	EG
IJMA J X X X	ORMOD	EJ	IJM T 0 X X X	ORMOD	EE
IJMA M X X X	ORMOD	EJ	IJM T 0 X X X	ORMOD	EK
IJMA R X X X	ORMOD	EV	IJM T 0 X X X	ORMOD	EU
IJMCEXXX	ORMOD	EA	IJM T 1 X X X	ORMOD	EP
IJMCLXXX	ORMOD	EW			
			IJM T 1 X X X	ORMOD	EQ
IJMDLXXX	ORMOD	ED	IJM T 2 X X X	ORMOD	EP
IJMDRXXX	ORMOD	ED	IJM T 2 X X X	ORMOD	EQ
IJMDYXXX	ORMOD	ED	IJM T 3 X X X	ORMOD	EP
IJME3XXX	ORMOD	EK	IJM T 3 X X X	ORMOD	EQ
IJME4XXX	ORMOD	EK	IJM T 5 X X X	ORMOD	EB
IJMGBXXX	ORMOD	EE	IJM T 6 X X X	ORMOD	EA
IJMGOXXX	ORMOD	EU	IJM T 7 X X X	ORMOD	EA
IJMM2XXX	ORMOD	EU	IJM T 8 X X X	ORMOD	EK
IJMM3XXX	ORMOD	EU	IJM T 8 X X X	ORMOD	EU
IJMM4XXX	ORMOD	EV	IJM T 9 X X X	ORMOD	EE
IJMRDXXX	ORMOD	EP	IJM T 9 X X X	ORMOD	EL
IJMRFXXX	ORMOD	EV	IJM T 9 X X X	ORMOD	EU
IJMRGXXX	ORMOD	EV	IJM U A X X X	ORMOD	EW
IJMRLXXX	ORMOD	ET	IJM U B X X X	ORMOD	EW
IJMRTXXX	ORMOD	EW	IJM U C X X X	ORMOD	EW
IJMRUXXX	ORMOD	EV	IJM U T X X X	ORMOD	EW
IJMRXXX	ORMOD	EW	IJM U S X X X	ORMOD	EW
			IJM U T X X X	ORMOD	EW
IJMR1XXX	ORMOD	ET			
IJMR2XXX	ORMOD	ES	IJM U O X X X	ORMOD	EW
IJMR3XXX	ORMOD	ET	IJM U V X X X	ORMOD	EW
IJMR4XXX	ORMOD	ET	IJM U W X X X	ORMOD	EW

IJMWXXXX	ORMOD	EV	INDISC	\$\$BOCP01	QB
IJMWXXXX	ORMOD	EU	INDISC	\$\$BOCP11	QH
IJMWXXXX	ORMOD	EV	INDPCH	\$\$BOCP01	QA
IJMZXBX	ORMOD	EG	INDPCH	\$\$BOCP02	QD
IJMZXCX	ORMOD	EG	INDPCH	\$\$BOCP11	QG
IJMZXDX	ORMOD	EG	INDPCH	\$\$BOCP12	QK
IJMZXEX	ORMOD	EG	INDPCH1	\$\$BOCP01	QA
IJMZXFX	ORMOD	EG	INDPCH1	\$\$BOCP02	QD
IJMZXJX	ORMOD	EG	INDPCH1	\$\$BOCP11	QG
IJMZXKX	ORMOD	EG			
IJMZXLX	ORMOD	EG	INDPCH1	\$\$BOCP12	QK
IJMZXQX	ORMOD	EB	INIT	\$\$BOMT04	KG
IJM10XXX	ORMOD	EU	INITDUMP	\$\$BOMSG1	RP
			INITIAL	\$\$BOCP01	QC
IJM2MXXX	ORMOD	EC	INITIAL	\$\$BOCP02	QE
IJUADONE	MRMOD	DA	INITIATE	\$\$BMSGWR	LF
IJUAGAIN	MRMOD	DF	INITIATE	\$\$BOCPM1	RH
IJUCALLW	MRMOD	DB	INITIATE	\$\$BOCPM2	RH
IJUCHECK	MRMOD	DA	INITIATE	\$\$BOMT0M	LE
IJUCHEK	MRMOD	DA	INITIATE	\$\$BOMT0W	LE
IJUCKMES	MRMOD	DC	INITREG	\$\$BOMSG1	RM
IJUCLBUF	MRMOD	DA	INIT14	\$\$BOCP01	QB
IJUCLOEX	MRMOD	DB	INOM	\$\$BOCP01	QA
IJUDISDL	MRMOD	DD	INOM	\$\$BOCP11	QG
IJUDISEN	MRMOD	DD	INPUTCK	\$\$BCMT02	KU
IJUENDTS	MRMOD	DC			
IJUENGCK	MRMOD	DB	INWKFL	\$\$BOCP01	QC
IJUENLST	MRMOD	DF	INWKFL	\$\$BOCP02	QE
IJUEXIT	MRMOD	DF	INWKFL	\$\$BOCP11	QJ
			INWKFL	\$\$BOCP12	QL
IJUFEED	MRMOD	DB	IO		RC
IJUFETCH	MRMOD	DC	IOCCBKSP	\$\$BCMT05	KZ
IJUGETCK	MRMOD	DA	IOCCFSW	\$\$BCMT02	KT
IJUGEXIT	MRMOD	DA	IOCCFSW	\$\$BCMT04	KW
IJUINCR	MRMOD	DB	IOCCFSW	\$\$BCMT05	KY
IJUINIT	MRMOD	DA	IOCCFOVT	\$\$BCMT05	KZ
IJULITE	MRMOD	DE	IOCCLBST	\$\$BCMT04	KW
IJULOAD	MRMOD	DA	IOCCLBST	\$\$BCMT05	KZ
IJULOUT	MRMOD	DE	IOCCOUTF	\$\$BCMT05	KY
IJUNXTBF	MRMOD	DB	IOCCPART	\$\$BCMT05	KY
IJUNXTLT	MRMOD	DE	IOCCREW	\$\$BCMT04	KW
IJUOLDNW	MRMOD	DA			
IJUPROC	MRMOD	DE	IOCCREW	\$\$BCMT05	KY
IJUPSTCF	MRMOD	DB	IOCCRWTP	\$\$BCMT04	KW
IJUPSTOV	MRMOD	DB	IOCCRWTP	\$\$BCMT05	KY
			IOCCSWT1	\$\$BCMT04	KX
IJUREAD	MRMOD	DA	IOCCSWT1	\$\$BCMT05	KZ
IJUREGAD	MRMOD	DA	IOCCWRIT	\$\$BCMT04	KW
IJURETRY	MRMOD	DF	IOCCWRIT	\$\$BCMT05	KZ
IJUSETPT	MRMOD	DA	IOCCWRTT	\$\$BCMT04	KX
IJUSHIFT	MRMOD	DE	IOCCWRTT	\$\$BCMT05	KZ
IJUSING	MRMOD	DE	IOCCXTUR	\$\$BCMT04	KX
IJUTDSN	MRMOD	DC	IOCCXTUR	\$\$BCMT05	KZ
IJUTSTAS	MRMOD	DB	IOCEA1TD	\$\$BCMT01	KR
IJUTSTFC	MRMOD	DF	IOCEDONE	\$\$BCMT01	KR
IJUTSTRF	MRMOD	DB	IOCENORW	\$\$BCMT01	KR
IJUTTRFC	MRMOD	DF	IOCENTRY	\$\$BCMT03	KV
IJUTUE	MRMOD	DC			
IJUWAITF	MRMOD	DF	IOCEOFSW	\$\$BCMT01	KR
IJUXCPEG	MRMOD	DB	IOCEOUTT	\$\$BCMT03	KV
IJZXXXXX	DTFCN	CA	IOCEOVOP	\$\$BCMT01	KR
			IOCEOVSW	\$\$BCMT01	KR
IJ2MXXXX	DTFCN	CB	IOCEBTP	\$\$BCMT03	KV
IJ2OXXXX	DTFCN	CB	IOCEREOP	\$\$BCMT01	KR
IJ2PXXXX	DTFCN	CA	IOCEREWT	\$\$BCMT01	KS
IJ2RXXXX	DTFCN	CA	IOCERTAP	\$\$BCMT01	KS
ILLEGAL	\$\$BOMSG2	RQ	IOCESBYT	\$\$BCMT03	KV
INCORE	\$\$BOMSG1	RN	IOCESETR	\$\$BCMT01	KS

IOCESKP4	\$\$BCMT01	KS	MGTPOP	\$\$BOCP02	QE
IOCESTD	\$\$BCMT01	KR	MGTPOP	\$\$BOCP11	QJ
IOCESTD	\$\$BCMT03	KV	MGTPOP	\$\$BOCP12	QL
IOCESWCH	\$\$BCMT03	KV	MOVE	\$\$BCMT04	KW
IOCETEST	\$\$BCMT01	KR	MOVE	\$\$BCMT05	KY
			MOVE	\$\$BOMT03	KF
IOCEUPKK	\$\$BCMT01	KS	MOVECYL	\$\$BCEOV1	KP
IOCEXTED	\$\$BCMT01	KR	MOVEHDR	\$\$BOCPT2	QW
IOCOBNOL	\$\$BOMT05	KJ	MOVEMSG	\$\$BOMSG1	RN
IOCONOLB	\$\$BOMT05	KH	MSG	\$\$BCMT07	LC
IOCONSTD	\$\$BOMT05	KH	MSGEXIT	\$\$BOCPT1	QP
IOCOPNFL	\$\$BOMT05	KH			
IOCORBNS	\$\$BOMT05	KJ			
IOCORDBK	\$\$BOMT05	KJ	MSGROUT1	\$\$BOMSG1	RL
IOCOREAD	\$\$BOMT05	KH	MSGROUT2	\$\$BOMSG1	RN
IOCOREDB	\$\$BOMT05	KJ	MSGWRITE	\$\$BCMT07	LC
IOCORFWD	\$\$BOMT05	KJ	MSGWRITE	\$\$BJCOPT	KN
IOCOSBLC	\$\$BOMT05	KH	MSGWRITE	\$\$BOCPT1	QN
IOCOTMCK	\$\$BOMT05	KJ	MSGWRITE	\$\$BOCPT2	QX
IOCOULAB	\$\$BOMT05	KH	MSG05SW	\$\$BOMSG2	RQ
IOCOWROP	\$\$BOMT05	KH	MSG10A	\$\$BOMT03	KE
			MSG11A	\$\$BOMT01	KA
IPTCHK	\$\$BJCOPT	KM	MSG12A	\$\$BOCPT1	QN
ISADRINF	\$\$BCMT02	KT	MSG12A	\$\$BOCPT2	QU
ISDONE	\$\$BOMT02	KD	MSG12A	\$\$BOMT01	KA
ISDONE	\$\$BOMT04	KG	MSG12A	\$\$BOMT03	KF
ISDTFPH	\$\$BOMT01	KC	MSG13D	\$\$BOCPT1	QP
ISLOADPT	\$\$BOMT01	KA	MSG13D	\$\$BOMT01	KA
ISLOADPT	\$\$BOMT03	KE	MSG14A	\$\$BOCPT1	QN
ISOPEN	\$\$BOMT01	KC	MSG14A	\$\$BOMT01	KB
ISOPEN	\$\$BOMT03	KF	MSG15A	\$\$BOCPT1	QR
ISREWIND	\$\$BOMT01	KA			
ISV3	\$\$BCMT02	KT			
IS7TRK	\$\$BJCOPT	KM	MSG15A	\$\$BOMT01	KB
			MSG16A	\$\$BOCPT1	QR
JCTEXIT	\$\$BOMSG1	RP	MSG16A	\$\$BOMT01	KB
			MSG17D	\$\$BOMT02	KD
LABELCHK	\$\$BOMT01	KC	MSG18D	\$\$BOMT02	KD
LABELCHK	\$\$BOMT02	KD	MSG19A	\$\$BOCPT2	QV
LABFND	\$\$BOCPT1	QS	MSG19A	\$\$BOMT03	KF
LABFND	\$\$BOCPT2	QY	MSG19A	\$\$BOMT06	KK
LABLD	\$\$BOCPT1	QM	MSG23D	\$\$BOMT02	KD
LABLD	\$\$BOCPT2	QT	MSG3LD	\$\$BCMT01	KS
LASTOPN	\$\$BOUR01	AB	MSG30A	\$\$BCMT01	KR
LDIOREG	\$\$BOCP01	QC	MSG31D	\$\$BCMT03	KV
LDIOREG	\$\$BOCP02	QF	MSG32D	\$\$BOCPT1	QR
LDIOREG	\$\$BOCP11	QJ	MSG32D	\$\$BOMT01	KB
LDIOREG	\$\$BOCP12	QL	MSG33D	\$\$BOCPT1	QR
LEGAL1	\$\$BOCPM1	RJ	MSG33D	\$\$BOMT01	KB
LEGAL1	\$\$BOCPM2	RJ	MSG4184D	\$\$BOCPT2	QX
			MSG44A	\$\$BOCPT2	QY
LEGAL2	\$\$BOCPM1	RJ	MSG83I	\$\$BCMT07	LB
LEGAL2	\$\$BOCPM2	RJ	MSG83I	\$\$BJCOPT	KK
LEGAL3	\$\$BOCPM1	RJ			
LEGAL3	\$\$BOCPM2	RJ	NEXTTEST	\$\$BOMSG2	RQ
LOADPHAS	\$\$BOMSG1	RN	NOALT	\$\$BCMT02	KT
LOADREG	\$\$BOMSG1	RM	NOALT	\$\$BCMT07	LC
LOADWORK	\$\$BOMSG1	RL	NOANSWER	\$\$BMSGWR	LF
LOOP1	\$\$BOCPM1	RH	NOANSWER	\$\$BOMTOM	LE
LOOP1	\$\$BOCPM2	RH	NOANSWER	\$\$BOMTOW	LE
LOOP2	\$\$BOCPM1	RJ	NOCTLCHR	\$\$BOUR01	AA
LOOP2	\$\$BOCPM2	RJ	NODEFALT	\$\$BCCPT1	RB
LUBXXX	\$\$BOMSG1	RL	NODEFALT	\$\$BOCPT2	QW
			NOHDR	\$\$BOOR01	EY
MDRSBY	\$\$BCMT02	KT	NOMSG	\$\$BOCP01	QA
MDRSBY	\$\$BCMT07	LC	NOMSG	\$\$BOCP02	QD
MESSG5	\$\$BOMSG1	RN	NOMSG	\$\$BOCP11	QG
MGTPOP	\$\$BOCP01	QC	NOMSG	\$\$BOCP12	QK

NOP	\$\$BMSGWR	LF	P2540	\$\$BOCP02	QD
NOP	\$\$BOCPM1	RH	RDAGAIN	\$\$BOMT06	KK
NOP	\$\$BOCPM2	RH			
NOP	\$\$BOMT0M	LE	RDDISK		RC
NOP	\$\$BOMT0W	LE	RDFRWRD	\$\$BOOR01	EY
NORETURN	\$\$BCMT07	LB	RDTPLAB	\$\$BCCPT1	RA
NOREWUN	\$\$BCMT07	LB	READ	\$\$BOCPM1	RK
NOTBLANK	\$\$BCCPT1	RB	READ	\$\$BOCPM2	RK
NOTBLANK	\$\$BCMT04	KW	READAGAN	\$\$BOMT02	KD
NOTBLANK	\$\$BCMT05	KY	READBK	\$\$BOMT02	KD
NOTBLANK	\$\$BOCPT2	QW	READLABL	\$\$BOMT01	KC
NOTLEGAL	\$\$BMSGWR	LF	READMSG	\$\$BOMSG1	RP
NOTLEGAL	\$\$BOCPM1	RJ	READTAPE	\$\$BOMT01	KA
NOTLEGAL	\$\$BOCPM2	RJ	READ3TM	\$\$BOMT01	KB
			REINIT	\$\$BOCPT1	QP
NOTLEGAL	\$\$BOMT0M	LE	RELOC	\$\$BCEOV1	KP
NOTLEGAL	\$\$BOMT0W	LE	RELOCATE	\$\$BOMT01	KA
NOTOPEN	\$\$BOMT04	KG	REPCHCP1	\$\$BCLOSP	RF
NOVARY	\$\$BOUR01	AA			
NOVOLO	\$\$BOCPT2	QW	REPCHCP2	\$\$BCLOSP	RF
NOVOL1	\$\$BOCPT1	QN	REPUNCH	\$\$BERRTN	PA
NOXPIRCK	\$\$BOCPT2	QV	RESET	\$\$BOMT01	KC
NREWIND	\$\$BOMT06	KK	RESETCCB	\$\$BOMT05	KH
NRF	\$\$BCCPT1	RA	RESTORE	\$\$BCLOSP	RD
NRF	\$\$BOCPT1	QS	RESTORE	\$\$BOCPT1	QN
NRF	\$\$BOCPT2	QY	RESTORE2	\$\$BCLOSP	RE
NXTALT	\$\$BCMT02	KT	RETMON	\$\$BCLOSP	RD
NXTALT	\$\$BCMT07	LB	RETMON	\$\$BOCP02	QE
NXTFLD	\$\$BOCPT1	QR	RETMON	\$\$BOCP11	QH
NXTJIB	\$\$BCMT02	KT	RETRYMSG	\$\$BCLOSP	RG
NXTJIB	\$\$BCMT07	LC	RETRYMSG	\$\$BERRTN	PA
			RETURN	\$\$BCLOSP	RD
OLDCARD	\$\$BCMT04	KW			
OLDCARD	\$\$BOMT03	KE	RETURN	\$\$BCMT02	KU
OPENINIT	\$\$BCCPT1	QZ	RETURNED	\$\$BCMT07	LB
OPENNXT	\$\$BOCPT1	QP	RETURNED	\$\$BJCOPT	KK
OUT	\$\$BOMR01	DG	REWIND	\$\$BCMT06	LA
OUTFILE	\$\$BERRTN	PA	REWIND	\$\$BOMT01	KA
OUTPUT	\$\$BOCP01	QB	REWIND	\$\$BOMT03	KE
			REWINDIT	\$\$BOMT05	KH
PCHBLANK	\$\$BCLOSP	RD	REWINDT	\$\$BCMT06	LA
PCHBLNK	\$\$BCLOSP	RE	RST	\$\$BOCP01	QB
PCH2520	\$\$BOCP11	QG	RST	\$\$BOCP11	QH
PCH2520	\$\$BOCP12	QK	RTRYNT	\$\$BJCOPT	KK
PCH42	\$\$BOCP01	QA	RTVOL1	\$\$BOCPT1	QN
PCH42	\$\$BOCP02	QD			
PCH42	\$\$BOCP11	QG	SAVSYS	\$\$BOCP01	QB
PCH42	\$\$BOCP12	QK	SAVSYS	\$\$BOCP11	QH
PPRT	\$\$BOCP02	QD	SENSE	\$\$BOCPT1	QS
PPRT	\$\$BOCP11	QH	SENSE	\$\$BOCPT2	QY
PPRT	\$\$BOCP12	QK	SENSEX	\$\$BOCPT1	QS
PRINTIN	\$\$BOCP01	QC	SETABORT	\$\$BOMSG1	RN
PRINTIN	\$\$BOCP02	QE	SETEOB	\$\$BOCPM1	RH
PRINTIN	\$\$BOCP12	QL	SETEOB	\$\$BOCPM2	RH
			SETEOB	\$\$BOMT0M	LE
PROCESS	\$\$BOMT06	KK	SETEOB	\$\$BOMT0W	LE
PROCRUN	\$\$BCCPT1	QZ	SETINTSW	\$\$BOMSG1	RP
PROCRUN	\$\$BOCPT1	QN	SETMODE	\$\$BOMT05	KH
PROCRUN	\$\$BOCPT2	QU	SETONEOF	\$\$BOMT01	KC
PROCRUN1	\$\$BCCPT1	QZ	SETOPEN	\$\$BOMT01	KC
PROCRUN1	\$\$BOCPT1	QN	SETOPEN	\$\$BOMT02	KD
PROCRUN1	\$\$BOCPT2	QU			
PROCRUN2	\$\$BCCPT1	QZ	SETOPNBT	\$\$BOUR01	AB
PROCRUN2	\$\$BOCPT1	QN	SETPNT	\$\$BCEOV1	KP
PROCRUN2	\$\$BOCPT2	QU	SETSW	\$\$BOCP01	QA
PUNCHBLK	\$\$BCLOSP	RE	SETSW	\$\$BOCP02	QD
P2520	\$\$BOCP02	QD	SETSW	\$\$BOCP11	QG
P2540	\$\$BOCP01	QA	SETUPRD	\$\$BOOR01	EY

SEVENTRK	\$\$BOMT06	KK	TRANSENT	\$\$BCMT03	KV
SHIFTMSG	\$\$BOMSG1	RM	TRANSENT	\$\$BCMT04	KW
SKIP	\$\$BCCPT1	RB	TRANSENT	\$\$BCMT05	KY
SKIP	\$\$BCMR01	DH			
SKIP	\$\$BCMT01	KR			
SKIP	\$\$BCMT03	KV	TRANSENT	\$\$BOMT06	KK
SKIP	\$\$BCMT04	KW	TREWIND	\$\$BCCPT1	RA
SKIP	\$\$BCMT05	KY	TSTIGN	\$\$BOUR01	AA
SKIP	\$\$BCMT07	LB	TSTLAB	\$\$BCEOV1	KP
			TSTPCH	\$\$BERTN	PA
SKIP	\$\$BJCOPT	KK	TSTRNG	\$\$BOCP02	QE
SKIP	\$\$BOCPT2	QW	TSTRNG	\$\$BOCP11	QJ
SKIP	\$\$BOMR01	DG	TSTRNG	\$\$BOCP12	QL
SKIP	\$\$BOMT05	KH	TYPE	\$\$BMMR20	DJ
SKIP	\$\$BOMT06	KK	TYPEASW	\$\$BOMSG2	RQ
SKIPLBLS	\$\$BCMT01	KR	TYPEEOB	\$\$BOMSG1	RP
SKIPMSG	\$\$BCMT02	KT			
SKIPREWU	\$\$BCMT02	KT	UNTRC	\$\$BOCP01	QB
SKIPO	\$\$BMSGWR	LF	UNTRC	\$\$BOCP02	QD
SKIPO	\$\$BOCPM1	RH	UNTRC	\$\$BOCP11	QH
SKIPO	\$\$BOCPM2	RH	UNTRC	\$\$BOCP12	QK
SKIP1	\$\$BOMT06	KK	UNTRCP	\$\$BOCP01	QB
SKIP2	\$\$BMSGWR	LF	UNTRCP	\$\$BOCP02	QD
SKPVOL	\$\$BJCOPT	KM	UNTRCP	\$\$BOCP11	QH
SPACEFWD	\$\$BOMT01	KC	UNTRCP	\$\$BOCP12	QK
			UPDATE	\$\$BCMT02	KU
SRCHLAB	\$\$BOCPT1	QS	UPDATE	\$\$BCMT04	KW
SRCHLAB	\$\$BOCPT2	QY	UPDATE	\$\$BOMT01	KC
START	\$\$BOUR01	AA	UPDATE	\$\$BOMT04	KG
STARTIO	\$\$BCLOSP	RG	UPDATE	\$\$BOUR01	AB
STARTOVR	\$\$BOCPM1	RJ	USDSTM	\$\$BJCOPT	KK
STARTOVR	\$\$BOCPM2	RJ	USERTEST	\$\$BOMT04	KG
STKON	\$\$BOCP01	QB			
STKON	\$\$BOCP11	QH	VOLSKIP	\$\$BOCPT1	QP
STMODEIO	\$\$BOUR01	AB	VOLSKIP	\$\$BOCPT2	QV
STOR	\$\$BOCP01	QC	VOLSKIP1	\$\$BOCPT1	QP
STOR	\$\$BOCP11	QJ	VOLSKIP1	\$\$BOCPT2	QV
			VOL1CHK	\$\$BOCPT1	QN
STRTOP	\$\$BOCP01	QA	VOL1CHK	\$\$BOCPT2	QU
STRTOP	\$\$BOCP02	QD	VOL1CHK	\$\$BOMT05	KH
STRTOP	\$\$BOCP11	QG			
STRTOP	\$\$BOCP12	QK	WHDRS	\$\$BOCPT1	QS
SUMROUT	\$\$BOMSG2	RQ	WHDRS	\$\$BOCPT2	QY
SVCALL	\$\$BJCOPT	KN	WORKFILE	\$\$BCMT06	LA
SYSLOD	\$\$BOCP01	QC	WRITE	\$\$BOCPM1	RK
SYSLOD	\$\$BOCP11	QJ	WRITE	\$\$BOCPM2	RK
SYSNO	\$\$BCMT07	LC	WRITELAB	\$\$BOMT04	KG
YSO	\$\$BOCP01	QB	WRITEMSG	\$\$BOMSG1	RN
YSO	\$\$BOCP11	QH	WRITHDR	\$\$BOMT06	KK
			WRITTM	\$\$BJCOPT	KM
TDIR	\$\$BOCPT1	QR	WRNGPKSW	\$\$BOMSG2	RQ
TEST	\$\$BOMT01	KB	WTDEVED	\$\$BCLOSP	RG
TMPPBP	\$\$BCMT02	KT			
TMPPBP	\$\$BCMT07	LC	XPIRCHK	\$\$BOCPT2	QV
TMWRITE	\$\$BCCPT1	RA			
TMWRITE	\$\$BOCPT2	QV	YESLDPT	\$\$BOMT03	KE
TRANS	\$\$BMSGWR	LF	YESOPEN	\$\$BOMT03	KF
TRANSENT	\$\$BCMT01	KR	YESTM	\$\$BOMT01	KA

APPENDIX B: MESSAGE CROSS-REFERENCE LIST

Message Number	Message	Issuing Routine	Chart ID
4000I	RETRY	\$\$BCLOSP \$\$BERRTN	RD RE PA
4110A	NO VOL1 FOUND TLBL=xxxxxx filename SYSxxx=cuu	\$\$BOCPT2 \$\$BOMT03	QW KE
4111A	NO VOL1 FOUND filename SYSxxx=cuu	\$\$BOCPT1 \$\$BOMT01	QN KA
4112A	VOL. SERIAL NO. ERROR TLBL=xxxxxx filename SYSxxx=cuu	\$\$BOCPT1 \$\$BOCPT2 \$\$BOMT01 \$\$BOMT03	QN QU KA KF
4113D	NO HDR1 LBL FOUND filename SYSxxx=cuu	\$\$BOCPT1 \$\$BOMT01	QP KA
4114A	FILE SEQ NO. ERROR filename SYSxxx=cuu	\$\$BOCPT1 \$\$BOMT01	QN KB
4115A	FILE SERIAL NO. ERROR TLBL=xxxxxx filename SYSxxx=cuu	\$\$BOCPT1 \$\$BOMT01	QR KB
4116A	VOLUME SEQ NO. ERROR filename SYSxxx=cuu	\$\$BOCPT1 \$\$BOMT01	QR KB
4117D	NO TM FOUND ON READBK filename SYSxxx=cuu	\$\$BOMT02 \$\$BOMT05	KD KJ
4118D	FILE ID ERROR READBK filename SYSxxx=cuu	\$\$BOMT02	KD
4119A	FILE UNEXPIRED filename SYSxxx=cuu	\$\$BOCPT2 \$\$BOMT03 \$\$BOMT06	QV KF KK
4120I	TAPE POSITIONED WRONG filename SYSxxx=cuu	\$\$BOMT03	KF
4121A	NO ALTERN DRIVE ASSGN SYSxxx=cuu	\$\$BCMT07	LC
4122I	EOV ENCOUNTERED SYSxxx=cuu	\$\$BCMT07	LB
4123D	WRONG POSITION, READBK filename SYSxxx=cuu	\$\$BOMT02	KD
4124I	TOO MANY UHL's filename SYSxxx=cuu	\$\$BOMT04	KG
4125D	VOL1 LBL FOUND filename SYSxxx=cuu	\$\$BOMT05	KH

Message Number	Message	Issuing Routine	Chart ID
4126I	EOV ENCOUNTERED filename SYSxxx=cuu	\$\$BCMT02	KT
4130A	EOF OR EOV INQUIRY filename SYSxxx=cuu	\$\$BCMT01	KR
4131D	DTF=xxxxxxx LBL=xxxxxx BLOCK COUNT ERROR filename SYSxxx=cuu	\$\$BCMT01 \$\$BCMT03	KS KV
4132D	ERROR IN FILE ID filename SYSxxx=cuu	\$\$BOCPT1 \$\$BOMT01	QR KB
4133D	ERROR IN HDR LBL filename SYSxxx=cuu	\$\$BOCPT1 \$\$BOMT01	QR KB
4140A	NO ALTERN DRIVE ASSIGN filename SYSxxx=cuu	\$\$BCMT02	KT
4144A	1600 BPI TAPE MOUNTED filename SYSxxx=cuu	\$\$BOCPT2 \$\$BOMT03	QY KE
4151I	HDR1 LBL INFORMATION filename SYSxxx=cuu	\$\$BOMT01 \$\$BOMT03	KA KF
4170A	FILE PROTECTED TAPE SYSxxx=cuu	\$\$BJCOPT	KL
4171A	UNEXPIRED FILE SYSxxx=cuu	\$\$BJCOPT	KM
4172A	INVALID LABEL SET SYSxxx=cuu	\$\$BJCOPT	KM
4880	INVALID FILE TYPE	\$\$BCEOV1	KP
4881I	NO LABEL INFORMATION	\$\$BCCPT1 \$\$BCEOV1	RA KQ
4883I	INVALID LOGICAL UNIT filename SYSxxx=cuu	Any Open or Close Phase	
4884D	FILE PROTECT RING NEEDED filename SYSxxx=cuu	\$\$BOCP02 \$\$BOCP11 \$\$BOCP12 \$\$BOCPT2	QE QJ QL QX
4887I	SYS FILE EXTNT EXCEEDED filename SYSxxx=cuu	\$\$BERRTN	PA
4888I	EOF ON SYSTEM INPUT FILE filename SYSxxx=cuu	\$\$BERRTN	PA
4MR1I	EXTERNAL INTERRUPT I/O ERROR	MRMOD	DC
4MR2I	SCU NOT OPERATIONAL	MRMOD	DC

APPENDIX C: SUPERVISOR CALLS (SVC'S)

Macro Supported	SVC	Function
EXCP	0	Execute channel programs.
FETCH	1	Fetch any phase.
	2	Fetch a logical transient (B-transient).
	3	Fetch or return from a physical transient (A-transient).
LOAD	4	Load any phase.
MVCOM	5	Modify supervisor communications region.
CANCEL	6	Cancel a problem program.
WAIT	7	Wait on a CCB or TECB.
	8	Transfer control to the problem program from a logical transient (B-transient).
LBRET	9	Return to a logical transient (B-transient) from the problem program after a SVC 8.
SETIME	*10	Set timer interval.
	11	Return from a logical transient (B-transient).
	12	Logical AND (reset) to second Job Control byte (displacement 57 in communications region).
	13	Logical OR (set) to second Job Control byte (displacement 57 in communications region).
EOJ	14	Cancel job and go to Job Control for end of job step.
	15	Same as SVC 0 except ignored if CHANQ table is full.

Macro Supported	SVC	Function
STXIT (PC)	*16	Provides supervisor with linkage to user's PC routine for program check interrupts.
EXIT (PC)	*17	Return from user's PC routine.
STCIT (IT)	*18	Provides Supervisor with linkage to user's IT routine for interval timer interrupts.
EXIT (IT)	*19	Return from user's IT routine.
STXIT (OC)	*20	Provides Supervisor with linkage to user's OC routine for external or attention interrupts (operator communications).
EXIT (OC)	*21	Return from user's OC routine.
	*22	The first SVC 22 seizes the system for the issuing program by disabling multiprogram operation. The second SVC 22 releases the system (enables multiprogram operation).
	*23	Load phase header. Phase load address is stored at user's address.
	*24	Provide Supervisor with linkage to user's TECB and set timer interval.
	*25	Issues Halt I/O (HIO) on a Teleprocessing device
	*26	Validate address limits.
	*27	Halt I/O (HIO) on a Teleprocessing device.
EXIT (MR)	*28	Return from user's stacker select routine.
	*29	Multiple wait.
	*30	Reserved for QTAM.
	*31	Reserved for QTAM.
	*32	Reserved for QTAM.
COMRG	33	Internal.
GETIME	*34	Internal.
	*35	Track Hold Function.
FREE	*36	Free held track.
STXIT(AB)	*37	Abnormal termination exit.
ATTACH	*38	Attach subtask.
DETACH	*39	Detach subtask.
POST	*40	Post subtask termination.
DEQ	*41	Dequeue resource.
ENQ	*42	Enqueue resource.
	50	Special error code.

* = Optional

APPENDIX D: MICROFICHE CROSS-REFERENCE LIST FOR VOLUME 2

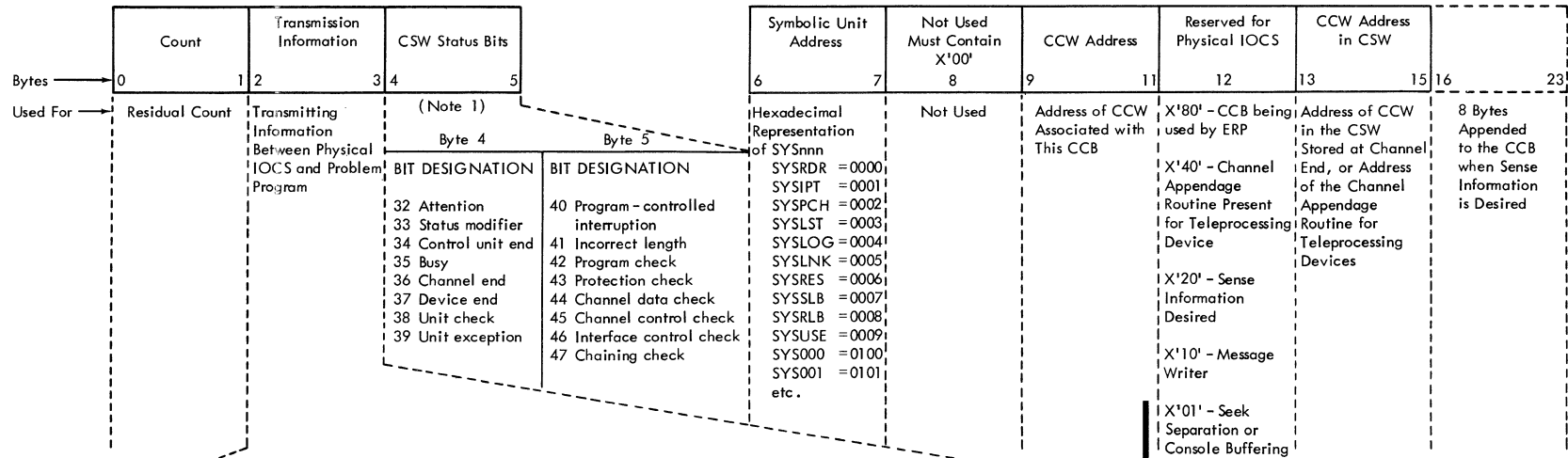
Logical IOCS Macros

		SEOV	453.012
		TRUNC	453.016
<u>Macro</u>	<u>Microfiche</u>	WAITF	453.016
	<u>Card Number</u>	WRITE	453.016

CDMOD	453.002
CHECK	453.002
CHNG	453.002
CLOSE(R)	453.003
CNTRL	453.003
DIMOD	453.004

Logical Transient Phases

		<u>Phase</u>	<u>Microfiche</u>
			<u>Card Number</u>
DISEN	477.001		
DSPLY	453.016		
DTFCD	453.004	\$\$BCCPT1	CTL.033.00
DTFCN	453.004	\$\$BCEOV1	CTL.034.00
DTFDI	453.005	\$\$BCMR01	IOM.003.00
DTFEN	453.005	\$\$BCMT01	CTL.044.00
DTFMR	477.001	\$\$BCMT02	CTL.045.00
DTFMT	456.001	\$\$BCMT03	CTL.046.00
DTFOR	478.001	\$\$BCMT04	CTL.047.00
DTFPH	453.005	\$\$BCMT05	CTL.048.00
DTFPR	453.006	\$\$BCMT06	CTL.049.00
DTFPT	458.001	\$\$BCMT07	CTL.050.00
		\$\$BERRTN	CTL.059.00
FEOV	456.001	\$\$BJCOPT	CTL.061.00
FREE	453.006.50	\$\$BMMR20	IOM.004.00
		\$\$BMSGWR	CTL.063.05
GET	456.007		
		\$\$BOCPM1	CTL.064.10
LBRET	453.008	\$\$BOCPM2	CTL.065.20
LITE	477.001	\$\$BOCPT1	CTL.066.30
		\$\$BOCPT2	CTL.067.40
MRMOD	477.001	\$\$BOCP01	CTL.068.00
MTMOD	456.002	\$\$BOCP02	CTL.069.00
		\$\$BOCP11	CTL.070.00
NOTE	453.009	\$\$BOCP12	CTL.071.00
		\$\$BOMR01	IOM.005.00
OPEN(R)	453.009	\$\$BOMSG1	CTL.089.00
ORMOD	478.001	\$\$BOMSG2	CTL.090.00
		\$\$BOMSG3	CTL.091.00
POINTR	453.010	\$\$BOMSG4	CTL.092.00
POINTS	453.010	\$\$BOMSG5	CTL.093.00
POINTW	453.010	\$\$BOMSG6	CTL.094.00
PRMOD	453.010	\$\$BOMT0M	CTL.095.00
PRTOV	453.011	\$\$BOMT0W	CTL.096.00
PTMOD	458.001	\$\$BOMT01	CTL.097.00
PUT	453.011	\$\$BOMT02	CTL.098.00
		\$\$BOMT03	CTL.099.00
		\$\$BOMT04	CTL.100.00
RDLNE	478.001	\$\$BOMT05	CTL.101.00
READ	453.011	\$\$BOMT06	CTL.102.00
RELSE	453.011	\$\$BOOR01	IOR.002.00
RESCN	478.001	\$\$BOUR01	CTL.127.00



Byte 2

Traffic Bit (Wait) (Note 5)	End-of-File (/ * or / &) (Note 2)	Unrecoverable I/O Error	Acceptable Unrecoverable I/O Error	Return DASD Data Checks or 2671 errors to the User	Post at Device End (Note 5)	Return Tape Read Data Check, 2540 or 2520 Equipment Check, or DASD Data Checks on Read or Verify Command (Note 3)	User Error Routine
0	1	2	3	4	5	6	7
Set On By →	PIOCS	PIOCS	PIOCS	Pr. Pr.	Pr. Pr.	Pr. Pr.	Pr. Pr.

Byte 3

DASD - Data Check in Count Area MICR - SCU Not Operational 1285/1287 - Data Check in Journal Tape Mode	DASD - Track Overrun MICR - Intervention Required 1285/1287 - Keyboard Correction in Journal Tape Mode	DASD - End of Cylinder MICR - (Note 4) 1285/1287 - Hopper Empty in Document Mode	2540, 2520 - Equipment Check Tape - Read Data Check DASD - Any Data Check 1285/1287 - Equipment Check	Non-Recovery Questionable Condition: Card - Unusual Command Sequence DASD - No Record Found 1285/1287 - Document Jam or Torn Tape	No - Record - Found Condition (Retry on 2311 or 2314)	Carriage Channel 9 Overflow or Verify Error for DASD 1287 Document Mode - Late Stacker Select	Command Chaining Retry from the next CCW to be executed
0	1	2	3	4	5	6	7
Set On By →	PIOCS	PIOCS	PIOCS	PIOCS	Pr. Pr.	PIOCS	Pr. Pr.

PIOCS = Physical IOCS
Pr. Pr. = Problem Program

Note 1. Bytes 4 and 5 contain the status bytes of the Channel Status Word (Bits 32 - 47). If byte 2, bit 5 is on and device end results as a separate interrupt, device end will be ORed into CCB byte 4.

Note 2. Indicates / * or / & statement encountered on SYSRDR or SYSIPT. Byte 4, bit 7 (unit exception) is also on.

Note 3. DASD data checks on count not returned.

Note 4. For 1412/1419, Disengage. For 1419D, I/O Error in external interrupt routine (channel data check or busout check).

Note 5. The traffic bit (Byte 2, bit 0) is normally set on at channel end to signify that the I/O was completed. If byte 2, bit 5 has been set on, the traffic bit and bits 2 and 6 in byte 3 will be set on at device end. Also see Note 1.

Byte	Bit	Condition Indicated		Mask for Test Under Mask Instruction	
		1 (ON)	0 (OFF)		
2	0	Traffic Bit (WAIT)	I/O Completed. Normally set at Channel End. Set at Device End if Bit 5 is ON.	I/O Requested and not completed.	X'80'
	1	End of File on System Input	/* or /& on SYSRDR or SYSIPT. Byte 4, Unit Exception Bit is also ON.		X'40'
	2	Unrecoverable I/O Error	I/O error passed back due to program option or operator option	No program or operator option error was passed back	X'20'
	3*	Accept Unrecoverable I/O Error (Bit 2 is ON)	Return to User after Physical IOCS Attempts to correct I/O Error. +	Operator Option: Dependent on the Error	X'10'
	4*	2671 Data Check Return any DASD Data Checks	Operator Options: Ignore, Retry, or Cancel Return to User	Operator Option: Retry or Cancel	X'08'
	5*	Post at Device End	Device End Condition will be posted i.e., byte 2, bit 0 and byte 3, bits 3 and 6 set at Device End. Also byte 4, bit 5 is set (see Figure 37).	Device End Conditions will not be posted. Traffic Bit is set at Channel End.	X'04'
	6*	Return: Uncorrectable tape read data check; 2540 or 2520 punch equipment check; or DASD read or verify data check. (Data checks on count not returned.)	Return to user after physical IOCS attempts to correct	Operator Option: Ignore or Cancel for Tapes and punches. Retry or cancel for DASD.	X'02'
	7*	User Error Routine	User will handle error recovery (Test Bit 2). II	A Physical IOCS Error routine will be used.	X'01'
3	0	Data check in DASD count field. MICR - SCU not operational. Data Check - 1285 or 1287 in Journal Tape Mode	Yes - Byte 3, bit 3 is OFF; Byte 2, bit 2 is ON. Yes Yes	No No No	X'80'
	1	DASD Track overrun. MICR Intervention required. Keyboard correction 1285 or 1287 in Journal Tape Mode	Yes Yes Yes	No No No	X'40'
	2	End of DASD Cylinder MICR - 1412/1419 disengage - 1419D, I/O error in external interrupt routine Hopper Empty 1287 Document Mode	Yes Document feeding Stopped. Channel data check or Busout check Yes	No No No	X'20'
	3	Tape read data check; 2540 or 2520 punch equipment check; or any DASD data check 1285 or 1287 equipment check	Operation was unsuccessful. Byte 2, Bit 2 is also ON. Byte 3, Bit 0 is OFF. Yes	No No No	X'10'

Command Control Block Storage Layout (Part 2 of 3)

Byte	Bit	Condition Indicated		Mask for Test Under Mask Instruction	
		1 (ON)	0 (OFF)		
3	4	Questionable Condition Non-recovery	Card: Unusual Command sequence (2540). DASD: No record found. 1285/1287: Document Jam or Torn Tape		X'08'
	5	No record found condition	Retry command if no record found condition occurs (2311 or 2314)	Set ON questionable condition bit and return to user.	X'04'
	6	Verify Error for DASD or Carriage Channel 9 Overflow 1287 Document Mode - Late Stacker Select	Yes. (Set ON when Channel 9 is reached only if Byte 2, Bit 5 is ON.)	No	X'02'
	7*	Command Chain Retry	Retry begins at last CCW executed.	Retry begins at first CCW of channel program	X'01'

- * User Option Bits. Set in CCB macro. Physical IOCS sets the other bits OFF at EXCP time and ON when the condition specified above occurs.
- + I/O program check, command reject, or tape equipment check always terminates the program.
- ⌘ User must handle all error or exceptional conditions except Channel Control Check, Interface Control Check, I/O Program Check, and I/O Protection Check.

Command Control Block Storage Layout (Part 3 of 3)

APPENDIX F: CONTROL CODES

CTLCHR=ASA

A control character must appear in each logical record if the ASA option is chosen. If the control character for the printer is not valid, a message is given and the job is canceled. If the control character for the card punch is not V or W, the card is selected into pocket 1. The codes are as follows:

<u>Code</u>	<u>Interpretation</u>
(blank)	Space one line before printing
0	Space two lines before printing
-	Space three lines before printing
+	Suppress space before printing
1	Skip to channel 1 before printing
2	Skip to channel 2 before printing
3	Skip to channel 3 before printing
4	Skip to channel 4 before printing
5	Skip to channel 5 before printing
6	Skip to channel 6 before printing
7	Skip to channel 7 before printing
8	Skip to channel 8 before printing
9	Skip to channel 9 before printing
A	Skip to channel 10 before printing
B	Skip to channel 11 before printing
C	Skip to channel 12 before printing
V	Select stacker 1
W	Select stacker 2

CTLCHR=YES

The control character is the command-code portion of the System/360 Channel Command Word used in printing a line or spacing the forms. If the character is not one of the following characters, unpredictable events will occur.

<u>8-Bit Code</u>	<u>Punch Combination</u>	<u>Function</u>
-------------------	--------------------------	-----------------

Stacker Selection on 1442

10000001	12,0,1	Select into stacker 1
11000001	12,1	Select into stacker 2

Pocket Selection on 2540

00000001	12,9,1	Select into pocket 1
01000001	12,0,9,1	Select into pocket 2
10000001	12,0,1	Select into pocket 3

Stacker Selection on 2520

00000001	12,9,1	Select into stacker 1
01000001	12,0,9,1	Select into stacker 2

Printer Control

00000001	12,9,1	Write (no automatic space)
00001001	12,9,8,1	Write and space 1 line after printing
00010001	11,9,1	Write and space 2 lines after printing
00011001	11,9,8,1	Write and space 3 lines after printing
10001001	12,0,9	Write and skip to channel 1 after printing
10010001	12,11,1	Write and skip to channel 2 after printing
10011001	12,11,9	Write and skip to channel 3 after printing
10100001	11,0,1	Write and skip to channel 4 after printing
10101001	11,0,9	Write and skip to channel 5 after printing
10110001	12,11,0,1	Write and skip to channel 6 after printing
10111001	12,11,0,9	Write and skip to channel 7 after printing

<u>8-Bit Code</u>	<u>Punch Combination</u>	<u>Function</u>	<u>8-Bit Code</u>	<u>Punch Combination</u>	<u>Function</u>
11000001	12,1	Write and skip to channel 8 after printing	10010011	12,11,3	Skip to channel 2 immediately
11001001	12,9	Write and skip to channel 9 after printing	10011011	12,11,8,3	Skip to channel 3 immediately
11010001	11,1	Write and skip to channel 10 after printing	10100011	11,0,3	Skip to channel 4 immediately
11011001	11,9	Write and skip to channel 11 after printing	10101011	11,0,8,3	Skip to channel 5 immediately
11100001	11,0,9,1	Write and skip to channel 12 after printing	10110011	12,11,0,3	Skip to channel 6 immediately
00001011	12,9,8,3	Space 1 line immediately	10111011	12,11,0,8,3	Skip to channel 7 immediately
00010011	11,9,3	Space 2 lines immediately	11000011	12,3	Skip to channel 8 immediately
00011011	11,9,8,3	Space 3 lines immediately	11001011	12,0,9,8,3	Skip to channel 9 immediately
10001011	12,0,8,3	Skip to channel 1 immediately	11010011	11,3	Skip to channel 10 immediately
			11011011	12,11,9,8,3	Skip to channel 11 immediately
			11100011	0,3	Skip to channel 12 immediately
			00000011	12,9,3	No op

INDEX

- Alternate Switching for System Units, Tape (\$\$BCMT07) 71
- Alternate Switching for EOVS, Tape (\$\$BCMT02) 68
- ASA Control Codes 294

- BSI Buffer Status Indicator (MICR) 24
- Buffer (MICR) 25
- Buffer Status Indicator, BSI (MICR) 24

- Card Device Files (CD) 13
- Card Device Files (CD) Logic Module, CDMOD 16
- CCB, Command Control Block 291
- CDMOD Macro 16
- CDMOD Module
 - CNTRL Macro 17
 - CNTRL Macro Chart BA 103
 - GET Macro 17
 - GET Macro Charts BB-BD 104-106
 - GET Macro, Combined Files 17
 - PUT Macro 17
 - PUT Macro Charts BD-BH 106-110
 - PUT Macro, Combined Files 17
- CHECK Macro
 - MRMOD 24
 - MRMOD Charts DA-DC 113
 - MTMOD Workfiles Chart HA 54, 156
- Checkpoint Records, MTMOD Bypass Routine 57
- Checkpoint/Restart - Refer to Volume 1
- Close
 - Alternate Switching for EOVS (\$\$BCMT02) 68
 - Alternate Switching System Units (\$\$BCMT07) 71
 - DTFCP/DTFDI Tape Files (\$\$BCCPT1) 93
 - EOF Backward (\$\$BCMT03) 68
 - EOF/EOVS Input Forward (\$\$BCMT01) 67
 - EOVS Output Forward (\$\$BCMT04) 69
 - Magnetic Ink Character Recognition 27
 - Magnetic Tape except Workfiles (\$\$BCMT05) 70
 - Optical Reader Files 31
 - Paper Tape Files 34
 - Punch Files (\$\$BCLOSP) 94
 - Unit Record Files 12
 - Workfiles (\$\$BCMT06) 71
- CNTRL Macro
 - CDMOD 17
 - MTMOD 54
 - MTMOD Workfile 54
 - ORMOD 29
 - PRMOD 33
- Combined File (DTFCD) 13
- Command Control Block, CCB 291
- Compiler Files (CP) 75
 - Characteristics 76
 - CPMOD Macro 83
 - DTFCP Macro 75
 - Initialization and Termination 91
 - Logic Module (CPMOD) 83
- Console Files (CN) 17
 - Close 17
 - DTFCN Macro 17
 - GET Macro 18
 - GET Macro Chart CA 111
 - PUT Macro 19
 - PUT Macro Chart CB 112
 - Logic Module 17
 - Open 17
- CPMOD Macro 83
 - Parameters 83-84
 - GET, IOPTR=YES 84
 - GET, IOPTR=YES Chart MC 200
 - GET, One I/O Area 84
 - GET, One I/O Area Chart MB 219
 - GET, Two I/O Areas 84
 - GET, Two I/O Areas Chart MA 218
 - PUT, IOPTR=YES 85
 - PUT, IOPTR=YES Charts MJ-MK 226-227
 - PUT, One I/O Area 85
 - PUT, One I/O Area Charts MG-MH 224-225
 - PUT, Two I/O Areas 84
 - PUT, Two I/O Areas Charts MD-MF 221-223
- DEVADDR, DTFCP Macro Parameter 75
- Device Independent Files 75
 - Initialization and Termination 91
 - Open (\$\$BOCP01) 91
 - Open (\$\$BOCP02) 92
- Device Independent System Files (DI) 85
- DIMOD Macros
 - GET, one I/O Area 89
 - GET, One I/O Area Charts NA-NB 228-229
 - GET, Two I/O Areas 90
 - GET, Two I/O Areas Charts NC-ND 230-231
 - PUT, One I/O Area 90
 - PUT, One I/O Area Charts NE-NG 232-234
 - PUT, Two I/O Areas 90
 - PUT, Two I/O Areas Charts NH-NK 235-237
- DIMOD Macro 89
- DIMOD Module 89
- DISEN Macro, MRMOD 24
- DISEN Macro, MRMOD Chart DD 116
- Disk Open Message Writer
 - Phase 1 (\$\$BOMSG1) 95
 - Phase 2 (\$\$BOMSG2) 95
- DISK=, CPMOD Macro Parameter 84
- DSPLY Macro, ORMOD 30
- DTF Tables
 - DTFCD Combined 16
 - DTFCD Input 14
 - DTFCD Output 15
 - DTFCP DISK=NO 80
 - DTFCP DISK=OMITTED 82
 - DTFCP DISK=YES 77
 - DTFDI 87
 - DTFMR 20
 - DTFMT Data Files 43
 - DTFMT Workfiles 51
 - DTFOR 28
 - DTFPH Magnetic Tape 52

DTFPR	32	DTFPR (Printer) Files	31
DTFPT No Translation, Shifts, Deletes	35	Macro	31
DTFPT Translation, No Shift or Deletes	36	Table	32
DTFPT Translation, Shifts, Deletes, Fixed Unblocked Records	37	DTFPT	
DTFPT Translation, Shifts, Deletes, Undefined Records	39	Paper Tape	34
DTFCD Macro	13	Macro	34
DTFCD Tables		No Translation, Shifts, or Deletes	35
Combined	16	Translation, Shifts, Deletes with Fixed Unblocked Records	37
Input, Reader	14	Translation, Shifts, Deletes with Undefined Records	39
Output, Punch	5	Translation, No Shifts or Deletes	36
DTFCN Macro	17	DTFXX Macros	
DTFCN Table	18	DTFCD	13
DTFCP Close Tape Files (\$\$BCCPT1)	93	DTFCN	17
DTFCP I/O Subroutines	264	DTFCP	75
DTFCP Macro	75	DTFDI	86
Parameters	75	DTFMR	19
DTFCP Message Writers (\$\$BOCPM1 and \$\$BOCPM2)	94, 95	DTFMT	42
DTFCP Open		DTFOR	27
Input Tape (\$\$BOCPT1)	93	DTFPH (Magnetic Tape)	42
Output Tape (\$\$BOCPT2)	93	DTFPR	31
Version 1 Only, Phase 1 (\$\$BOCP11)	92	DTFPT	34
Version 1 Only, Phase 2 (\$\$BOCP12)	92	EOF/EOV Monitor (\$\$BCEOV1)	67
DTFCP Tables		EOF/EOV Monitor, Magnetic Tape (\$\$BCEOV1)	67
DISK=NO	80	EOFADDR=, DTFCP Macro Parameter	76
DISK=OMITTED	82	Error Messages	286
DISK=YES	77	Error Recovery, Punch (\$\$BERRTN)	91
DTFDI (Device Independent System) Files	86	FEOV Macro	54
DTFDI Close Tape Files (\$\$BCCPT1)	93	FEOV Macro, MTMOD Chart HD	159
DTFDI I/O Subroutines Chart RC	264	File Labels	
DTFDI Macro	89	General Information - See Volume 1	
DTFDI Message Writers (\$\$BOCPM1 and \$\$BOCPM2)	94	Standard Magnetic Tape	60
DTFDI Open		Flowchart Symbols	98
Input Tape (\$\$BOCPT1)	93	GET Macro	
Output Tape (\$\$BOCPT2)	93	CDMOD	17
DTFDI Table	87-89	CDMOD Charts BB-BD	104-106
DTFMR		CPMOD, IOPTR=YES	84
Magnetic Ink Character Recognition	19	CPMOD, IOPTR=YES Chart MC	220
Close (\$\$BCMR01)	27	CPMOD, One I/O Area	84
Macro	19	CPMOD, One I/O Area Chart MB	219
Message Writer (\$\$BMMR20)	27	CPMOD, Two I/O Areas	84
Open (\$\$BOMR01)	26	CPMOD, Two I/O Areas Chart MA	218
Table	20	DIMOD, One I/O Area	89
DTFMT Macro		DIMOD, One I/O Area Charts NA-NB	228
Data Files	42	DIMOD, Two I/O Areas	90
Workfiles	42	DIMOD, Two I/O Areas Charts NC-ND	230
DTFMT		DTFCN	18
Magnetic Tape Close - See Close, Magnetic Tape		DTFCN Chart CA	111
Macro, Data Files	42	MRRMOD	24
Macro, Workfiles	42	MRRMOD Charts DA-DC	112-115
Open - See Open, Magnetic Tape		MTMOD, Fixed Records	54
Tables Data Files	43	MTMOD, Fixed Records Charts HE-HG	160
Tables Workfiles	51	MTMOD, Undefined Records	54
DTFOR		MTMOD, Undefined Records Charts HE-HG	160
Optical Reader	27	MTMOD, Variable Records	54
Close	31	MTMOD, Variable Records Charts HE-HG	160
Macro	27	ORMOD, Blocked Records	30
Open (\$\$BOOR01)	31	ORMOD, Blocked Records Charts EJ-EN	130
Table	28	ORMOD, Unblocked Records	30
DTFPH Macro	42	ORMOD, Unblocked Records Charts EE-EH	126
DTFPH Macro (Magnetic Tape)	42		
DTFPH Table, Magnetic Tape	52		

Magnetic Tape Close			
All Files but Work (\$\$BCMT05)	70		
Alternate Switching for EOF (\$\$BCMT02)	68		
Alternate Switching for System Units (\$\$BCMT07)	71		
EOF/EOV, Input for (\$\$BCMT01)	67		
EOV, Backward (\$\$BCMT03)	69		
EOV, Output for (\$\$BCMT04)	69		
Workfiles (\$\$BCMT06)	71		
Magnetic Ink Character Recognition (MR) Files	19		
Buffer	25		
Close (\$\$BCMR01)	27		
DTFMR Macro	19		
Error Messages	27		
Initialization and Termination	26		
Logic Module (MRMOD)	23		
Message Writer (\$\$BMMR20)	27		
MRMOD Macro	23		
Open (\$\$BOMR01)	26		
Pocket Light Indicator	25		
Pocket Lights, Example	25		
Magnetic Tape CLOSE and EOF/EOV Routines	100		
Magnetic Tape OPEN Routines	99		
Magnetic Tape	42		
Alternate Switching for EOF (\$\$BCMT02)	68		
Alternate Switching for System Units (\$\$BCMT07)	71		
Checkpoint/Restart - See Volume 1			
Close All Files except Work (\$\$BCMT05)	70		
EOF/EOV Input Forward (\$\$BCMT01)	67		
EOV Backward (\$\$BCMT03)	68		
EOV Backward (\$\$BCMT03) Chart KV	207		
EOV Output Forward (\$\$BCMT04)	69		
Workfiles (\$\$BCMT06)	71		
Message Cross-Reference List	286		
Message Writer Open			
I/O Nonstandard/Unlabeled (\$\$BOMT05)	65		
Input Standard Labels, Backward (\$\$BOMT02)	62		
Input Standard Labels, Forward (\$\$BOMT01)	61		
Job Control (\$\$BJCOPT)	66		
Output Standard Labels (\$\$BOMT03)	63		
Output Standard Labels (\$\$BOMT04)	64		
SYSRES Label Information	61		
Tapemarks, Placement of	58		
Workfiles (\$\$BOMT06)	66		
Message Writer			
Disk Open (\$\$BOMSG1, \$\$BOMSG2)	95		
DTFCP/DTFDI (\$\$BOCPM1, 2)	94		
Magnetic Ink Character Recognition (\$\$BMMR20)	27		
Magnetic Tape (\$\$BOMTOM, \$\$BOMTOW, \$\$BMSGWR)	72		
MICR - See Magnetic Ink Character Recognition			
Microfiche Reference List	290		
Modules - See Logic Modules			
MRMOD (Magnetic Ink Character Recognition)	23		
MRMOD Macro	23		
MRMOD Module			
CHECK Macro	24		
CHECK Macro Charts DA-DC	113		
DISEN Macro	24		
DISEN Macro Chart DD	116		
GET Macro	24		
GET Macro Charts DA-DC	113		
LITE Macro	25		
LITE Macro Chart DE	117		
READ Macro	26		
READ Macro Charts DA-DC	113		
WAITF Macro	26		
WAITF Macro Chart DF	118		
MTMOD (Magnetic Tape)	53		
MTMOD Macro	53		
Block/Deblock Subroutine	173		
Work Subroutines, Fixed and Variable Records	174		
Work Subroutines, Undefined Records	175		
CHECK	54		
CHECK Chart HA	156		
CNTRL, Data Files	54		
CNTRL, Data Files Chart HB	157		
CNTRL, Workfiles	54		
CNTRL, Workfiles Chart HC	158		
FEOV	54		
FEOV Chart HD	159		
GET, Fixed Records	54		
GET, Fixed Records Charts HE-HG	160		
GET, Undefined Records	54		
GET, Undefined Records Charts HE-HG	160		
GET, Variable Records	54		
GET, Variable Records Charts HE-HG	160		
NOTE, Workfiles	55		
NOTE, Workfiles Chart HH	163		
POINTR, Workfiles	55		
POINTR, Workfiles Chart HJ	164		
POINTS, Workfiles	55		
POINTS, Workfiles Chart HK	165		
POINTW, Workfiles	56		
POINTW, Workfiles Chart HL	166		
PUT, Fixed Records	56		
PUT, Fixed Records Charts HM-HN	167		
PUT, Undefined Records	56		
PUT, Undefined Records Charts HM-HN	167		
PUT, Variable Records	56		
PUT, Variable Records Charts HM-HN	167		
READ, Workfiles	56		
READ, Workfiles Chart HP	169		
RELSE	57		
RELSE Chart HQ	170		
TRUNC	57		
TRUNC Chart HR	171		
WRITE, Workfiles	57		
WRITE, Workfiles Chart HS	172		
MTMOD Modules	53		
Bypass Checkpoint Records Routine	57		
Nonstandard Label Magnetic Tape Open (\$\$BOMT05)	65		
NOTE Macro, MTMOD Workfiles	55		
Open Magnetic Tape			
I/O Nonstandard/Unblocked (\$\$BOMT05)	65		
Input Standard Label (\$\$BOMT01)	61		

Input Standard Label (\$\$BOMT02)	62	POINTS Macro, MTMOD Workfiles	55
Optical Reader Files (\$\$BOOR01)	31	POINTW Macro, MTMOD Workfiles	56
Output Standard Label (\$\$BOMT03, \$\$BOMT04)	63	Printer (PR) Files	31
Printer Files	12	DTFPR Macro	31
Punch Files	12	DTFPR Table	32
Reader Files	12	Logic Module	32
Unit Record Files (\$\$BOUR01)	12	Open	12
Workfiles (\$\$BOMT06)	66	PRMOD Macro	32
Open		STL (Selective Tape Lister)	33
Console (CN) Files	17	PRMOD Macro	32
Device Independent DTFCP/DTFDI Files (\$\$BOCP01)	91	PRMOD Module	
Device Independent DTFCP/DTFDI Files (\$\$BOCP02)	92	CNTRL Macro	33
DTFCP (Version 1 Only) Phase 1 (\$\$BOCP11)	92	CNTRL Macro Chart FA	145
DTFCP (Version 1 Only) Phase 2 (\$\$BOCP12)	92	PRTOV Macro	33
DTFCP/DTFDI Input Tape (\$\$BOCPT1)	93	PRTOV Macro Chart FB	146
DTFCP/DTFDI Output Tape (\$\$BOCPT2)	93	PUT Macro	33
Job Control, Magnetic Tape (\$\$BJCOPT)	66	PUT Macro Charts FC-FE	147
Magnetic Ink (\$\$BOMR01)	26	PRTOV Macro, PRMOD Chart FB	33, 146
Optical Reader (OR) Files	27	PTMOD Macro	40
Close	31	PTMOD Module	
DTFOR Macro	27	GET Macro, No Translation	40
Initialization/Termination	31	GET Macro, No Translation Chart GA	150
Logic Module (ORMOD)	29	GET Macro, Translation, Shift, Fixed Unblocked Records	41
Open (\$\$BOOR01)	31	GET Macro, Translation, Shift, Fixed Unblocked Records Charts GC-GD	152
ORMOD Macro	29	GET Macro, Translation, Shift, Undefined Records	41
ORMOD Macros	29	GET Macro, Translation, Shift, Undefined Records Charts GE-GF	154
ORMOD Module		GET Macro, Translation of Shifted Code	41
CNTRL Macro	29	GET Macro, Translation of Shifted Code Chart GB	151
CNTRL Macro Charts EA-EC	122	Punch Error Recovery (\$\$BERRTN)	91
DSPLY Macro	30	Punch File Close (\$\$BCLOSP)	94
DSPLY Macro Chart ED	125	Punch File Open	12
READ Macro	30	Punch-Feed-Read (PFR), Card Device Files	16
READ Macro Chart ER	137	PUT Macro	
RESCN Macro	31	CPMOD	17
RESCN Macro Charts ES-ET	138	CPMOD Charts BD-BH	107
WAITF Macro	31	CPMOD, Combined Files	17
WAITF Macro Charts EU-EX	140	CPMOD, IOPTR=YES	85
GET Macro, Blocked Records	30	CPMOD, Charts MJ-MK	226
GET Macro, Blocked Records Charts EJ-EN	130	CPMOD, One I/O Area	85
GET Macro, Unblocked Records	30	CPMOD, One I/O Area Charts MG-MH	224
GET Macro, Unblocked Records Charts EE-EH	126	CPMOD, Two I/O Areas	84
Paper Tape	34	CPMOD, Two I/O Areas Charts MD-MF	221
DTFPT Macro	34	DIMOD, One I/O Area	90
DTFPT Tables No Translation, Shifts, Delete	35	DIMOD, One I/O Area Charts NE-NG	232
DTFPT Tables Translation, Shifts, Deletes Unblocked Records	37, 38	DIMOD, Two I/O Areas	90
DTFPT Tables Translation, Shifts, Deletes Undefined Records	39, 40	DIMOD, Two I/O Areas Charts NH-NK	235
DTFPT Tables Translation, No Shifts, Deletes	36	DTFCN, Console Files	19
Logic Module (PTMOD)	40	DTFCN, Console Files Chart CB	112
PDTABB, Magnetic Ink Character Recognition	27	MTMOD, Fixed Records	56
PFR (Punch Feed Read), Card Device Files	16	MTMOD, Fixed Records Charts HM-HN	167
Physical IOCS, Magnetic Tape (DTFPH)	42	MTMOD, Undefined Records	56
POINTR Macro, MTMOD Workfiles	55	MTMOD, Undefined Records Charts HM-HN	167
		MTMOD, Variable Records	56
		MTMOD, Variable Records Charts HM-HN	167
		PRMOD	33
		PRMOD Charts FC-FE	147
		PRMOD with STL	33
		PRMOD with STL Charts FC-FE	147

RDLIN Macro, ORMOD 30
 RDLIN Macro ORMOD Charts EP-EQ 135
 RDONLY=, CPMOD Macro Parameter 84
 RDONLY=, DTFCP Macro Parameter 76
 READ Macro
 MRMOD 26
 MRMOD Charts DA-DC 113
 MTMOD, Workfile 56
 MTMOD, Workfile Chart HP 169
 ORMOD 30
 ORMOD Chart ER 137
 Read/WRITE Subroutines Fixed Records
 Charts HW-HY 176
 Read/WRITE Subroutines Undefined Records
 Charts JC-JE 182
 Read/WRITE Subroutines Variable Records
 Charts HZ-JB 179
 Reader File Open 12
 RECSIZE=, CPMOD Macro Parameter 76
 RELSE Macro, MTMOD 57
 RELSE Macro, MTMOD Chart HQ 170
 RESCN Macro, ORMOD 31
 RESCN Macro, ORMOD Charts ES-ET 138
 Restart - See Volume 1
 RETRY=, CPMOD Macro Parameter 84

 Selective Tape Lister Feature (STL) 33
 Standard Magnetic Tape Labels - See
 Tape Labels
 STL Control Field 33
 Supervisor Calls, SVCs 288
 SYSIR Macro 12
 SYSRES Magnetic Tape Label Information 61

 Tables
 DTF - See DTF Tables and Individual
 DTFXX Macro
 PDTABB, Magnetic Ink Character
 Recognition

 Tape - Also See Magnetic Tape
 Tape Labels
 Processing, General - See Volume 1
 Standard File 60
 Standard Volume (VOL1) 59
 Tape Open Close Subroutines Chart LD 215
 Tapemarks, Placement of 58
 Transients, Logical (B) - See
 Microfiche Reference 290
 Translation, Paper Tape Files 34
 TRUNC Macro, MTMOD 57
 TRUNC Macro, MTMOD Chart HR 171
 TYPEFLE=, DTFCP Macro Parameter 76
 TYPEFLE=, CPMOD Macro Parameter 83

 Unit Record Files 12
 Close 12
 Initialization and Termination 12
 Open (\$\$BOUR01) 12
 Unlabeled Magnetic Tape File Open
 (\$\$BOMT05) 65

 VOL1, Standard Volume Label,
 Magnetic Tape 59

 WAITF Macro
 MRMOD 26
 MRMOD Chart DF 118
 ORMOD 31
 ORMOD Charts EU-EX 140
 Work Area Subroutines Fixed and Variable
 Records Chart HU 174
 Work Area Subroutines Undefined Records
 Chart HV 175
 Work File Subroutines Chart JG 186
 WRITE Macro, MTMOD Workfiles 57
 WRITE Macro, MTMOD Workfiles Chart HS 172



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

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

Reader's Comments Form

IBM System/360
Disk Operating System Logical IOCS
Volume 2: Unit Record, Magnetic Tape,
and Device Independent Files

Y24-5087-1

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. Please give specific page and line references with your comments when appropriate. All comments will be handled on a non-confidential basis. Copies of this and other IBM publications can be obtained through IBM branch offices.

- | | <i>Yes</i> | <i>No</i> |
|--|--------------------------|---|
| ● Does the publication meet your needs? | <input type="checkbox"/> | <input type="checkbox"/> |
| ● Did you find the material: | | |
| Easy to read and understand? | <input type="checkbox"/> | <input type="checkbox"/> |
| Organized for convenient use? | <input type="checkbox"/> | <input type="checkbox"/> |
| Complete? | <input type="checkbox"/> | <input type="checkbox"/> |
| Well illustrated? | <input type="checkbox"/> | <input type="checkbox"/> |
| Written for your technical level? | <input type="checkbox"/> | <input type="checkbox"/> |
| ● What is your occupation? _____ | | |
| ● How do you use this publication: | | |
| As an introduction to the subject? <input type="checkbox"/> | | As an instructor in a class? <input type="checkbox"/> |
| For advanced knowledge of the subject? <input type="checkbox"/> | | As a student in a class? <input type="checkbox"/> |
| For information about operating procedures? <input type="checkbox"/> | | As a reference manual? <input type="checkbox"/> |

Your comments:

Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

Your comments, please . . .

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

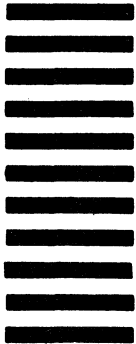
Please note: Requests for copies of publications and for assistance in using your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

Fold

Fold

FIRST CLASS
PERMIT NO. 170
ENDICOTT, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP 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



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

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

Additional Comments: