SY33-8551-1
File No. S370-36

Systems

DOS/VS Supervisor Logic

Program Numbers 5745-SC-SUP
5745-SC-AIT
5745-SC-DKE

Release 30 (POWER/VS Version)

IBM

Second Edition (November 1973)

This edition, together with Technical Newsletters SN33-8769, SN33-8770, and SN33-8790, applies to Version 5, Release 30 (POWER/VS Version), of the Disk Operating System/Virtual Storage, DOS/VS, and to all subsequent versions and releases until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest Virtual Storage Supplement (to IBM System/360 and System/370 Bibliography), GC20-0001, for the editions that are applicable and current.

Note: For the availability dates of the Virtual Telecommunications in this manual, please contact your IBM representative or the IBM branch office serving your locality.

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

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Laboratory, Publications Department, P.O. Box 24, Uithoorn, The Netherlands. Comments become the property of IBM.

PREFACE

This logic manual is one of three detailed guides to the logic of the IBM Disk Operating System Virtual Storage (DOS/VS) Supervisor. This manual supplements the program listings by providing text and charts describing:

1. The organization of the supervisor area in real storage.
2. The internal logic of the resident nucleus routines.
3. The physical input/output control system.
4. The internal logic of the physical attention transient routines ($$ABERRZ and $$ABERZ1).

The internal logic of other physical transients (error recovery and recording) and the logical transients is described in the companion manuals:

- DOS/VS Error Recovery and Recording Transients, SY33-8552
- DOS/VS Logical Transients, SY33-8553.

For overall system logic, these three manuals are to be used with:

- DOS/VS IPL and Job Control, SY33-8555
- DOS/VS Linkage Editor, SY33-8556
- DOS/VS Librarian, SY33-8557.

Prerequisite knowledge for the effective use of these PLMs is contained in the following reference manuals and guides:

- IBM System/370 Principles of Operation, GA22-7000
- IBM System/360 Principles of Operation, GA22-6821
- Introduction to DOS/VS, GC33-5370
- DOS/VS System Management Guide, GC33-5371
- OS/VS and DOS/VS Assembler Language, GC33-4010.

For details of the logic of the POWER, POWER/RJE programs, refer to the following manuals:

- DOS/VS POWER, SY33-8565
- DOS/VS POWER/RJE, SY33-8566

For details of the Virtual Telecommunications Access Method (VTAM) refer to:

- VTAM Function Summary, SY27-7256 (for high-level information)
- DOS/VS VTAM Logic, SY33-7262 (for low-level information)

Publications related in subject matter to the system logic manuals are:

- DOS/VS System Generation, GC33-5377
- DOS/VS Operating Procedures, GC33-5378
- DOS/VS Messages, GC33-5379
- DOS/VS Supervisor and I/O Macros, GC33-5373
- DOS/VS Data Management Guide, GC33-5372
- DOS/VS System Control Statements, GC33-5376
- DOS/VS Serviceability Aids and Debugging Procedures, GC33-5380.

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

CONTENTS

INTRODUCTION 15

SUPERVISOR GENERATION AND ORGANIZATION . 19
Supervisor Generation 19
Supervisor Organization 19
Low Real Storage 19

TASK SELECTION 77
System Task Selection 78
Dispatching a System Task 78
Interruption of a System Task 78
User Task Selection 78
Example of User Task Selection 79
DAT Dispatcher 81
Asynchronous Processing (AP) 83

INTERRUPT PROCESSORS 85
I/O Interrupt 85
Program Check Interrupt 85
Page Fault Handler 86
External Interrupt 88
Machine Check Interrupt 88
Supervisor Call Interrupt (SVC) 88

PAGE MANAGEMENT 113
Tables 113
Page Frame Table 113
Segment and Page Tables 114
Selection Pool 115
Page Frame Selection 116
Page Handling Routines 117
Page Fault 118
Page Fault Handling Overlap 118
GETREAL Request 118
TFIX Request 119
PFIX Request 120
PAGEIN Request 120
FREEREAL, FREE, and PFREE Requests . .121
RELPAG and FCEPGOUT Requests 121
Partition Deactivation and Reactivation
(Load Leveler) 121
Partition Deactivation 122
Partition Reactivation 125

PHYSICAL INPUT/OUTPUT CONTROL SYSTEM
(PIOCS) 129
Channel Scheduler 129
Selector and Block Multiplexer
Channel Switching 138
Channel Queue 139
Channel Buckets 144
Command Control Block 145

I/O Interrupt Handler 150
Burst-Mode and High-speed Byte-Mode
Devices on Byte Multiplexer Channel .152
Channel Program Translation 153
Translation Control and Copy Blocks .153
Copying and Translating Channel
Programs 160
Processing After Completion of an I/O
Request 174
BTAM Considerations 174
Track-Hold Facility 174
Seek Separation 179
Console Buffering 180
Error Queueing and Dequeueing 181
Loading ERP Physical Transients . . .183
Disk Error Recovery 183
2311 and 2314 DASD Error Recovery . .183
3330 Disk Storage Error Recovery . . .185
3340 DASD Error Recovery 186

PROGRAM RETRIEVAL 189
Shared Virtual Area (SVA) 189
Directory List Support 189
Search for Directory Entry 190
Relocating Load 193

PHYSICAL ATTENTION 195

MACHINE CHECK AND CHANNEL CHECK
HANDLING 197
Machine Check Analysis and Recording . .197
Channel Check Handler 198
Recovery Transients and RAS Monitor . .200

CHARTS 205

APPENDIX A: LABEL LIST 441

APPENDIX B: TASK AND PARTITION KEY
DEFINITIONS 447

APPENDIX C: DEVICE TYPE CODES 449

APPENDIX D: TAPE DENSITY DATA 451

APPENDIX E: CANCEL CODES AND MESSAGES .453

APPENDIX F: MESSAGE REFERENCE LIST . .455

APPENDIX G: TOTAL MESSAGE CROSS
REFERENCE 457

INDEX 465

FIGURES

Figure 1.1. Multiprogramming Storage Organization 16
Figure 1.2. Supervisor Storage Allocation 17
Figure 2.1. Error Bytes in Low Real Storage (Part 2 of 2) 21
Figure 2.2. Supervisor Macros (Part 1 of 11) 23
Figure 2.3. Macro Function (Part 1 of 3) 34
Figure 2.4. Global Settings (Part 1 continued) 36
Figure 2.5. System Communication Region (SYSCOM) (Part 1 of 2). 42
Figure 2.6. Partition Communications Region (Part 1 of 6) 44
Figure 2.7. Program Information Block (PIB) Table (Part 1 of 2) 50
Figure 2.8. Indications of Logical Transient Area Occupancy and Activity . 52
Figure 2.9. Program Information Block Table Extension (PIB2TAB) 53
Figure 2.10. LTA and Problem Program Save Area 53
Figure 2.11. Boundary Box (BBOX) . . . 54
Figure 2.12. Tables for MICR DTF Addresses and Pointers (Part 1 of 2) . . 54
Figure 2.13. Job Accounting Interface Common Table (ACCTCOMN) (Part 1 of 2) . 55
Figure 2.14. Job Accounting Interface Partition Table (ACCTxx) (Part 1 of 2) . 56
Figure 2.15. Disk Information Block (DIB) Table 58
Figure 2.16. Deleted 59
Figure 2.17. Line Mode Table (LMT) . . 60
Figure 2.18. Recorder File Table (Part 1 of 2) 61
Figure 2.19. Accessing the PUB2 Table (PUB2AREA) 63
Figure 2.20. PUB2 Table Entry Format for Unit Record and Unsupported - Devices 64
Figure 2.21. PUB2 Table Entry Format for 3886 Optical Character Reader. . . . 64
Figure 2.22. PUB2 Table Entry Format for DASD 65
Figure 2.23. PUB2 Table Entry Format for 3540 Diskette. 66
Figure 2.24. PUB2 Table Entry Format for Tapes (Part 1 of 2) 67
Figure 2.25. Accessing PDAID Routines 69
Figure 2.26. Page Data Set Table (DPDTAB) 70
Figure 2.27. CRT Constant Table (CRTTAB) 70
Figure 2.28. System Task Save Areas and CRT Constants 71
Figure 2.29. ASCII Translation Tables (Part 1 of 4) 72
Figure 3.1. TRTFLD, TRFLD, TIDFLD, and MVCFLD, assuming that NPARTS=3 and AP=NO 77
Figure 3.2. Resource Table (TRTMASK) . 77
Figure 3.3. System Task Identification - Field (STID) 78
Figure 3.4. System Task Blocks 79
Figure 3.5. Example of User Task Selection (Part 1 of 2). 80
Figure 3.6. Examples of Dispatching by DAT Dispatcher (Part 1 of 3) 82
Figure 3.7. TIDTRT Field 83
Figure 4.1. Program Check Option Table 86
Figure 4.2. Routine Identifiers (RID) as Used by the Page Fault Handler (PFFLIH) in Handling Page Requests . . 87
Figure 4.3. Supervisor Calls (Part 1 of 4) 89
Figure 4.4. Headqueue PUB Table 95
Figure 4.5. Operator Communication Option Table 95
Figure 4.6. Abnormal Termination Option Table 96
Figure 4.7. Interval Timer Option Table 96
Figure 4.8. Interval Timer Request Table 97
Figure 4.9. Format of the User's Save Area for AB, PC, OC, and IT Routines . . 97
Figure 4.10. Event Control Block (ECB) 100
Figure 4.11. Resource Control Block (RCB) 101
Figure 4.12. Emulator ECB Table (EUECBTAB) 101
Figure 4.13. Resource Usage Record Table (RURTAB) 106
Figure 4.14. Anchor Table (ANCHTAB) . .107
Figure 4.15. FIXWTAB 108
Figure 4.16. Page Fault Handling Overlap Option Table (PFATAB) 109
Figure 4.17. Restart-PFIX Parameter List Entry 109
Figure 4.18. PAGEIN Table (PAGETAB) . .112
Figure 5.1. Page Frame Table 114
Figure 5.2. Segment and Page Tables . .114
Figure 5.3. Selection Pool Queue - Identifiers and Queue Headers 116
Figure 5.4. Selection Pool Queue Headers and Queue Elements 116
Figure 5.5. Selection Pool Queue Switching 117
Figure 5.6. Page Queue (PGQU) 117
Figure 5.7. Actions after TFREE . . .121
Figure 5.8. Criteria for Partition Deactivation 122
Figure 5.9. Example of Usage of Workareas DEACTFLD and WAITFLD when Selecting a Partition for Deactivation .124
Figure 5.10. Example of Usage of Workareas DEACTFLD and WAITFLD when Selecting a Partition for Reactivation .127

Figure 6.1. I/O Table Interrelationships (Part 1 of 3) 130
Figure 6.2. Channel Queue Table (CHANQ) and Entry 133
Figure 6.3. Physical Unit Block (PUB) - Table 133
Figure 6.4. Physical Unit Block Ownership (PUBOWNER) Table 135
Figure 6.5. Job Information Block (JIB) Table 136
Figure 6.6. Tape Error Block Table (TEBTAB) 137
Figure 6.7. Channel Control Table (CHNTAB) 137
Figure 6.8. Logical Unit Block (LUB) Table 138
Figure 6.9. Example of Pointers to PUBs for Switchable Devices 139
Figure 6.10. Initial Contents of Significant Bytes used in Queueing I/O Requests 140
Figure 6.11. Queueing the First I/O Request 140
Figure 6.12. Queueing the First I/O Request for Second Device 141
Figure 6.13. Queueing an I/O Request for a Device with a Request Already Queued 141
Figure 6.14. Example of Channel Queue with Several Requests Queued for Each of Two Devices 142
Figure 6.15. Dequeueing First Request on Second Device, Leaving Outstanding Requests Chained 142
Figure 6.16. Adding a New Request to Chain for First Device 143
Figure 6.17. Dequeueing Last Request in Chain for First Device 143
Figure 6.18. Queueing New Request for First Device 144
Figure 6.19. Channel Buckets 145
Figure 6.20. Command Control Block (CCB) (Part 1 of 3) 147
Figure 6.21. Channel Command Word (CCW) 150
Figure 6.22. Program Status Word (PSW) 150
Figure 6.23. Channel Address Word (CAW) 150
Figure 6.24. Channel Status Word (CSW) 150
Figure 6.25. CSW Testing in I/O Interrupt Handler 152
Figure 6.26. Translation Control Block (CCWTCB) 155
Figure 6.27. Chain of Used Translation Control Blocks 156
Figure 6.28. CCB Copy Block 156
Figure 6.29. Locating CCB Copy Blocks .157
Figure 6.30. CCW Copy Block 157
Figure 6.31. Locating CCW Copy Blocks .158
Figure 6.32. Relation of IDAL Blocks to Other Blocks 159
Figure 6.33. Fix Information Bit String and Block 160
Figure 6.34. Initializing Special Command List Pointers in TCB 161
Figure 6.35. Schematic Representation of Channel Program Translation 162
Figure 6.36. CCW Translation for Channel Programs Without TIC or Status Modifier Commands 162
Figure 6.37. Locating a copy location for a CCW pointed to by a TIC when the location is in an already used copy block 163
Figure 6.38. Enqueueing a new copy block to the correct location in the CCW copy block chain to handle a CCW pointed to by a TIC. 164
Figure 6.39. CCW Copy block queueing calling for the creation of a "short" block to maintain alignment 165
Figure 6.40. Enqueueing new copy block which cannot be aligned to existing block because CCW too far removed from VBA of any existing block 167
Figure 6.41. A channel program containing status modifier commands after the first pass has been made at copying it 168
Figure 6.42. Channel program containing status modifier commands after the first pass has been made . . .169
Figure 6.43. The channel program containing status modifier commands after the second pass has been made. . .170
Figure 6.44. The channel program containing status modifier commands after translation has been completed . .171
Figure 6.45. Copied CCW requiring an IDAL to be built (normal READ or WRITE command) 172
Figure 6.46. Copied CCW requiring an IDAL to be built (READ Backward command) 173
Figure 6.47. Track-Hold Table: 175
Figure 6.48. Track-Hold Table (THTAB) .175
Figure 6.49. Track-Hold Table: 176
Figure 6.50. Example of Tracks Held and Freed by Three Tasks 177
Figure 6.51. Track-Hold Table: 178
Figure 6.52. Track-Hold Table: Situation after total release 178
Figure 6.53. Seek Address Block Table - (SAB) 179
Figure 6.54. Seek Address Block (SAB) Table Entries 179
Figure 6.55. Console Buffering Table (CBTAB) 181
Figure 6.56. Operation of Error Queue with Three Entries 182
Figure 6.57. Error Recovery Block (ERBLOC) and Error Queue Entry 182
Figure 7.1. Layout of Shared Virtual Area 189
Figure 7.2. Layout of Virtual Library .189
Figure 7.3. Fetch Parameter Save Area (CALLSEQ) 190
Figure 7.4. Fetch Table (FTTAB) 191
Figure 7.5. Relationship Between Directory and its Second Level Directory 191
Figure 7.6. Layout of SYSRES 192
Figure 7.7. Layout of Directory Entry .192
Figure 7.8. Core Image Library Block for Relocatable Phase 194

Figure 8.1. Physical Attention Messages ($$ABERRZ) 196
Figure 9.1. Channel Check Severity Detect Routine 199
Figure 9.2. Error Recovery Procedure Information Block (ERPIB) 200
Figure 9.3. RAS Monitor Table (RASTAB) (Part 1 of 2) 201
Figure 9.4. RAS Linkage Area (RASLINK) 203
Figure 10.1. Device Type Codes (Part 1 of 2) 449
Figure 10.2. Tape Density Data 451
Figure 10.3. Cancel Codes and Messages (Part 1 of 2) 453

CHARTS

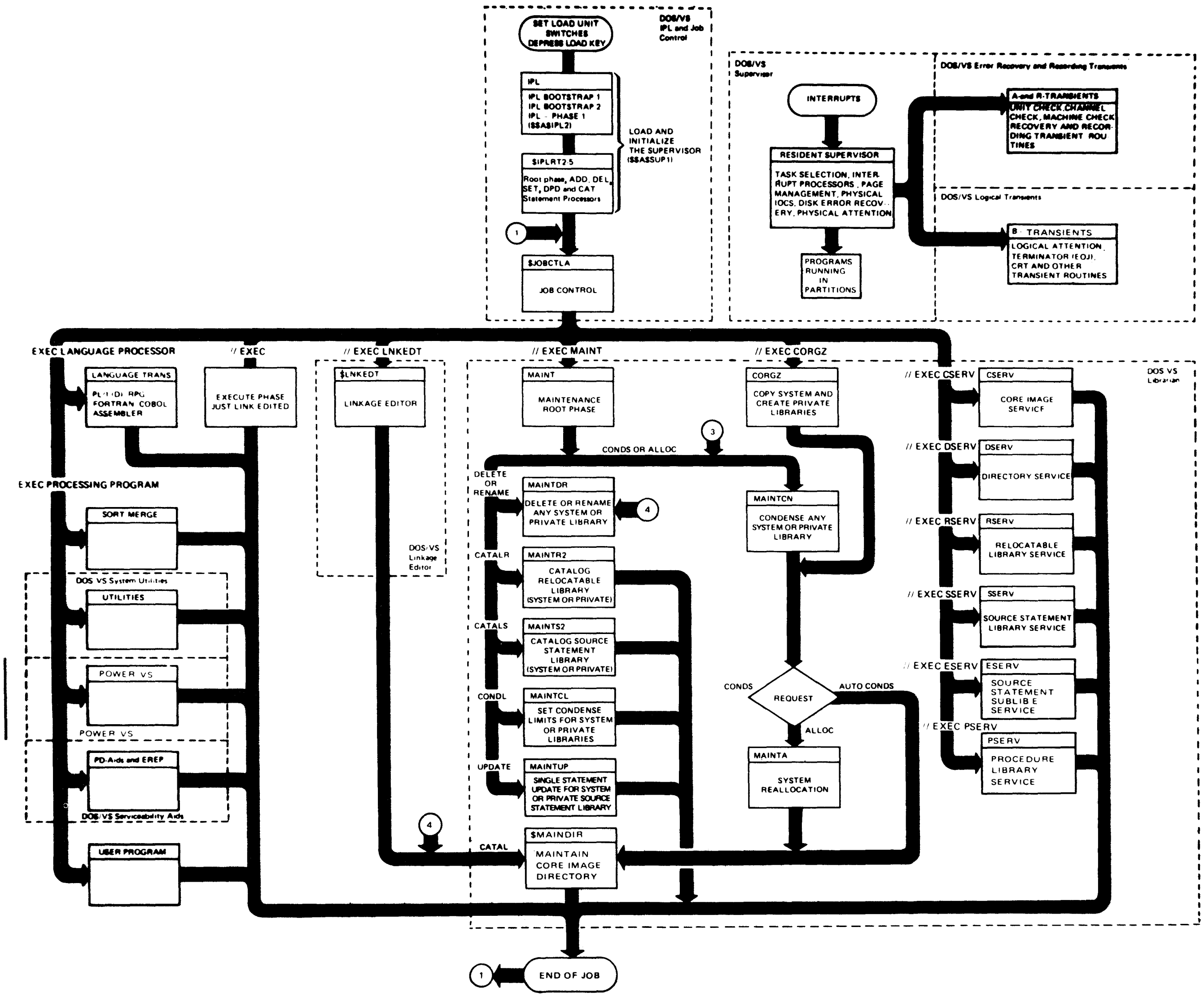
Chart 00. Disk Operating System Program Flow 14
Chart 01.1. Supervisor General Entry .205
Chart 02.1. Supervisor General Exit, Select Task 206
Chart 02.2. Supervisor General Exit, Cancel Exit 207
Chart 02.3. Supervisor General Exit, DAT Dispatcher 208
Chart 03.1. Program Check Handler, and Cancel Routine 209
Chart 03.2. Program Check Handler, Page Fault Handler 210
Chart 04.1. Physical Attention Routines 211
Chart 05.1. Supervisor Call Routines .212
Chart 05.2. Supervisor Call Routines .213
Chart 06.1. I/O Routines, I/O Interrupt Handler 214
Chart 06.2. I/O Routines, I/O Interrupt Handler 215
Chart 06.3. I/O Routines, Channel Scheduler 216
Chart 06.4. I/O Routines, Channel Scheduler 217
Chart 07.1. External Interrupt Routines 218
Chart 08.1. Fetch Routine 219
Chart 08.2. Fetch Routine 220
Chart 09.1. Unit Check Routine 221
Chart 10.1. Resident Disk Error Recovery Routine 222
Chart 11.1. Physical Transient Interface, Exit Routine 223
Chart 11.2. Physical Transient Interface, Return Routine 224
Chart 12.1. MCH/CCH, Machine Check Handler and Emergency Exit Routine . . .225
Chart 12.2. MCH/CCH, Channel Check Handler 226
Chart 12.3. MCH/CCH, RAS Monitor . . .227
Chart 13.1. Page Management, Enqueue Request to Page Queue 228
Chart 13.2. Page Management, Handle Page Queue Entry 229
Chart 13.3. Page Management, Dequeue Request from Page Queue 230
Chart 13.4. Page Management, Handle Selection Pool Empty Condition 231
Chart 13.5. Page Management, Subroutines, Reset Fetch 232
Chart 13.6. Page Management, Deactivation of Partition 233
Chart 13.7. Page Management, Reactivation of Partition 234
Chart 13.8. Page Management, Reactivation of Partition 235
Chart 13.9. Page Management, TFIX Routines 236
Chart 13.10. Page Management, TFREE Routines 237
Chart 13.11. Page Management, PFIX Routines 238
Chart 13.12. Page Management, PFIX, Find a Page Frame 239
Chart 13.13. Page Management, Exchange Pages, PFIX-Restart 240
Chart 13.14. Page Management, PFREE Routines 241
Chart 13.15. Page Management, PAGEIN Routine 242
Chart 13.16. Page Management, RELPAG and FCEPGOUT Routines 243
Chart 13.17. Page Management, GETREAL Routines 244
Chart 13.18. Page Management, Initialize Partition, Initialize Page Table Entries 245
Chart 13.19. Page Management, FREEREAL Routines 246
Chart 14.1. Channel Program Translation, General Routine 247
Chart 14.2. Channel Program Translation, General Routine 248
Chart 14.3. Channel Program Translation, Subroutines 249
Chart 14.4. Channel Program Translation, Build IDAL, TFIX Data Area 250
Chart 14.5. Channel Program Translation, LOCATE Routine 251
Chart 14.6. Channel Program Translation, Status Modifier Command . .252
Chart 14.7. Channel Program Translation, Handle TIC and Block End Chains, and IDALs 253
Chart 14.8. Channel Program Translation, Subroutines 254
Chart 14.9. Channel Program Retranslation 255
Chart 14.10. Channel Program Translation, Reset or Cancel 256
Chart 14.11. Channel Program Translation, SVC 72, SVC 77 257
Chart A00. Entry Index 258
Chart A01. $$A$SUP1 - IOINTER Macro, I/O Interrupt Handler, Dequeue Channel Queue Entry 259
Chart A02. $$A$SUP1 - PIOCS Macro, Start I/O 260
Chart A03. $$A$SUP1 - FOPT Macro, Task Selection, Initial Selection of Attention 261
Chart A04. $$A$SUP1 - FOPT Macro, Task Selection, Initial Selection of Attention; 262
Chart A05. $$A$SUP1 - FOPT Macro, Task Selection, System Task Dispatcher .263
Chart A06. $$A$SUP1 - FOPT Macro, Task Selection, System Task Dispatcher .264
Chart A07. $$A$SUP1 - SMICR Macro, External Interrupt Handler 265
Chart A08. $$A$SUP1 - PIOCS Macro, Check Seek Address 266

Chart A09. $$A$SUP1 - PIOCS Macro, Cancel Exit 267
Chart A10. $$A$SUP1 - SGTHAP Macro, AP Cancel Routine 268
Chart A11. $$A$SUP1 - SGTHAP Macro, AP Cancel Routine, Store Cancel Codes .269
Chart A12. $$A$SUP1 - SGTHAP Macro, Detach Subtask (SVC 39) 270
Chart A13. $$A$SUP1 - SGTHAP Macro, Detach Subtask (SVC 39) 271
Chart A14. $$A$SUP1 - IOINTER Macro, Release a Resource (SVC 64) 272
Chart A15. $$A$SUP1 - IOINTER Macro, Dequeue Channel Queue Entry 273
Chart A16. $$A$SUP1 - IOINTER Macro, Reschedule Channel 274
Chart A17. $$A$SUP1 - IOINTER Macro, Initialize Headqueuing 275
Chart A18. $$A$SUP1 - IOINTER Macro, I/O Interrupt Handler, PUBSCAN Routine .276
Chart A19. $$A$SUP1 - IOINTER Macro, I/O Interrupt Handler, Channel Appendage Interface 277
Chart A20. $$A$SUP1 - IOINTER Macro, I/O Interrupt Handler, Unit Check Routine 278
Chart A21. $$A$SUP1 - IOINTER Macro, I/O Interrupt Handler, Active ERP . . .279
Chart A22. $$A$SUP1 - IOINTER Macro, I/O Interrupt Handler, Activate Attention 280
Chart A23. $$A$SUP1 - IOINTER Macro, I/O Interrupt Handler 281
Chart A24. $$A$SUP1 - IOINTER Macro, I/O Interrupt Handler 282
Chart A25. $$A$SUP1 - IOINTER Macro, I/O Interrupt Handler, Unit Check Routine 283
Chart A26. $$A$SUP1 - IOINTER Macro, I/O Interrupt Handler, Unit Check Routine 284
Chart A27. $$A$SUP1 - SGDSK Macro, Resident Disk ERP 285
Chart A28. $$A$SUP1 - SGDSK Macro, Resident Disk ERP 286
Chart A29. $$A$SUP1 - SGDSK Macro, Resident Disk ERP 287
Chart A30. $$A$SUP1 - SGDSK Macro, Resident Disk ERP 288
Chart A31. $$A$SUP1 - SGDSK Macro, Resident Disk ERP 289
Chart A32. $$A$SUP1 - SGDSK Macro, Resident Disk ERP 290
Chart A33. $$A$SUP1 - FOPT Macro, Interval Timer 291
Chart A34. $$A$SUP1 - FOPT Macro, Interval Timer 292
Chart A35. $$A$SUP1 - FOPT Macro, Fetch Exit, Load Phase (SVC 4) 293
Chart A36. $$A$SUP1 - FOPT Macro, General Entry Routine 294
Chart A37. $$A$SUP1 - FOPT Macro, General Entry Routine 295
Chart A38. $$A$SUP1 - SMICR Macro, MICR Interrupt Handler 296
Chart A39. $$A$SUP1 - SMICR Macro, MICR Interrupt Handler 297
Chart A40. $$A$SUP1 - SMICR Macro, MICR Interrupt Handler 298
Chart A41. $$A$SUP1 - SMICR Macro, MICR Interrupt Handler 299
Chart A42. $$A$SUP1 - SMICR Macro, MICR Interrupt Handler 300
Chart A43. $$A$SUP1 - SMICR Macro, MICR Interrupt Handler 301
Chart A44. $$A$SUP1 - SMICR Macro, MICR Error Recovery for SIO 302
Chart A45. $$A$SUP1 - SMICR Macro, MICR Interrupt Handler 303
Chart A46. $$A$SUP1 - FOPT Macro, Interrupt Entry Routines, Job Accounting Routine 304
Chart A47. $$A$SUP1 - PIOCS Macro, Calculate PUB Address 305
Chart A48. $$A$SUP1 - PIOCS Macro, EXCP (SVC 0) 306
Chart A49. $$A$SUP1 - PIOCS Macro, EXCP (SVC 0) 307
Chart A50. $$A$SUP1 - PIOCS Macro, Subroutines 308
Chart A51. $$A$SUP1 - PIOCS Macro, RMSR Recording EXCP (SVC 76), Halt I/O (SVC 25) 309
Chart A52. $$A$SUP1 - PIOCS Macro, Halt I/O (SVC 25) 310
Chart A53. $$A$SUP1 - PIOCS Macro, SVC 3, SVC 5, Return to Supervisor from PTA 311
Chart A54. $$A$SUP1 - PIOCS Macro, Return to Supervisor from PTA 312
Chart A55. $$A$SUP1 - PIOCS Macro, Reset SVC 44 Request 313
Chart A56. $$A$SUP1 - PIOCS Macro, Check Whether Deactivated Task has SVC 44 Request with TFIX 314
Chart A57. $$A$SUP1 - SGSVC Macro, Entry for SVCs 7, 8, 9, and 10 315
Chart A58. $$A$SUP1 - SGSVC Macro, Set Timer Interval, Provide Linkage (SVC 24) 316
Chart A59. $$A$SUP1 - SGSVC Macro, Entry for SVCs 12, 13, and 16 317
Chart A60. $$A$SUP1 - SGSVC Macro, Entry for SVCs 17-21, 37 318
Chart A61. $$A$SUP1 - SGSVC Macro, Entry for SVCs 22, 23, and 24 319
Chart A62. $$A$SUP1 - SGSVC Macro, Entry for SVCs 26 and 52 320
Chart A63. $$A$SUP1 - SGSVC Macro, Entry for SVCs 29, 33, and 34 321
Chart A64. $$A$SUP1 - SGSVC Macro, Entry for SVC 44 322
Chart A65. $$A$SUP1 - SGSVC Macro, Entry for SVCs 45, 46, 48, 51, 54, and 56 323
Chart A66. $$A$SUP1 - SGSVC Macro, Entry for SVCs 57-59, 61-62, 65-69 . . .324
Chart A67. $$A$SUP1 - SGSVC Macro, Entry for SVCs 70-72 325
Chart A68. $$A$SUP1 - SGSVC Macro, Entry for SVCs 73, 74, and 77 326
Chart A69. $$A$SUP1 - SGSVC Macro, Address Validation 327

Chart A70. $$A$SUP1 - SGSVC Macro, Provide Virtual Address of Location in I/O Area (SVC 60) 328
Chart A71. $$A$SUP1 - SGTHAP Macro, Hold Track (SVC 35), Free Track (SVC 36) 329
Chart A72. $$A$SUP1 - SGTHAP Macro, Free Track (SVC 36) 330
Chart A73. $$A$SUP1 - SGTHAP Macro, Hold Track (SVC 35), Free Track (SVC 36) 331
Chart A74. $$A$SUP1 - SGTHAP Macro, Cancel Program or Task (SVC 6) 332
Chart A75. $$A$SUP1 - SGTHAP Macro, Attach Subtask (SVC 38) 333
Chart A76. $$A$SUP1 - SGTHAP Macro, Post ECB (SVC 40) 334
Chart A77. $$A$SUP1 - SGTHAP Macro, Investigate Waiting Tasks 335
Chart A78. $$A$SUP1 - SGTHAP Macro, Investigate Waiting Tasks 336
Chart A79. $$A$SUP1 - SGTHAP Macro, Enqueue (SVC 42) and Dequeue (SVC 41) Task from a Resource 337
Chart A80. $$A$SUP1 - SGTHAP Macro, Dequeue (SVC 41) Task from a Resource .338
Chart A81. $$A$SUP1 - IOINTER Macro, USE (SVC 63)/RELEASE (SVC 64) Common Subroutines 339
Chart A82. $$A$SUP1 - IOINTER Macro, USE (SVC 63)/RELEASE (SVC 64) Common Subroutines 340
Chart A83. $$A$SUP1 - SMICR Macro, Return from User Stacker Select Routine (SVC 28) 341
Chart A84. $$A$SUP1 - SMICR Macro, Return from User Stacker Select Routine (SVC 28) 342
Chart A85. $$A$SUP1 - IOINTER Macro, CSW Stored at SIO Time 343
Chart A86. $$A$SUP1 - IOINTER Macro . .344
Chart A87. $$A$SUP1 - IOINTER Macro .344.1
Chart B01. $$A$SUP1 - SGEND Macro, Activate CRT Hooks, LTA Routine Entered from BONCE 344.2
Chart B02. $$A$SUP1 - SGEND Macro, Activate Dynamic Address Translation, Initialize Page Table 345
Chart B03. $$A$SUP1 - SGDFCH Macro, Fetch Routine 346
Chart B04. $$A$SUP1 - SGDFCH Macro, Fetch Routine 347
Chart B05. $$A$SUP1 - SGDFCH Macro, Fetch Routine, Phase Not Found 348
Chart B06. $$A$SUP1 - SGDFCH Macro, Channel Program Translation for Fetch Program 349
Chart B07. $$A$SUP1 - SGDFCH Macro, Calculate Load Point and Entry Point of Phase 350
Chart B08. $$A$SUP1 - SGDFCH Macro, Load the Phase 351
Chart B09. $$A$SUP1 - SGDFCH Macro, Channel Program Retranslation 352
Chart B10. $$A$SUP1 - SGDFCH Macro, Relocate User Adcons 353
Chart B11. $$A$SUP1 - SGDFCH Macro, Scan Directory List for Equal Phase Name 354
Chart B12. $$A$SUP1 - SGDFCH Macro, Scan Link Directory 355
Chart B13. $$A$SUP1 - SGDFCH Macro, Check for More Buffer Space 356
Chart B14. $$A$SUP1 - SGDFCH Macro, Restart Fetch Routine 357
SGCCWT Macro, Channel Program Translation 357
Chart B15. $$A$SUP1 - SGCCWT Macro, Channel Program Translation 358
Chart B16. $$A$SUP1 - SGCCWT Macro, Channel Program Translation 359
Chart B17. $$A$SUP1 - SGCCWT Macro, Channel Program Translation 360
Chart B18. $$A$SUP1 - SGCCWT Macro, Channel Program Translation 361
Chart B19. $$A$SUP1 - SGCCWT Macro, Copy CCW 362
Chart B20. $$A$SUP1 - SGCCWT Macro, Set TCB Flags 363
Chart B21. $$A$SUP1 - SGCCWT Macro, Build Indirect Address List 364
Chart B22. $$A$SUP1 - SGCCWT Macro, Find Address of Copy Location 365
Chart B23. $$A$SUP1 - SGCCWT Macro, TFIX Data Area 366
Chart B24. $$A$SUP1 - SGCCWT Macro, TFREE Data Area 367
Chart B25. $$A$SUP1 - SGCCWT Macro, Get/Release TCB 368
Chart B26. $$A$SUP1 - SGCCWT Macro, Get/Release Copy Block 369
Chart B27. $$A$SUP1 - SGCCWT Macro, Release Blocks, Get Virtual Address of CCW 370
Chart B28. $$A$SUP1 - SGCCWT Macro, Channel Program Retranslation 371
Chart B29. $$A$SUP1 - SGCCWT Macro, Reset Channel Program Translation . . .372
Chart B30. $$A$SUP1 - SGCCWT Macro, Move Copied CCB to Virtual CCB 373
Chart B31. $$A$SUP1 - SGCCWT Macro, Reset Translation, GETCBUF, FREECBUF . .374
Chart B32. $$A$SUP1 - SGPMGR Macro, Entry Point for Program Check 375
Chart B33. $$A$SUP1 - SGPMGR Macro, Program Check Entry Routine 376
Chart B34. $$A$SUP1 - SGPMGR Macro, Enqueue Request to Page Queue 377
Chart B35. $$A$SUP1 - SGPMGR Macro, Page Fault Handling Overlap Interface .378
Chart B36. $$A$SUP1 - SGPMGR Macro, Page Manager System Task 379
Chart B37. $$A$SUP1 - SGPMGR Macro, Dequeue Page Queue Entry 380
Chart B38. $$A$SUP1 - SGPMGR Macro, Handle No Page Frames Available Condition 381
Chart B39. $$A$SUP1 - SGPMGR Macro, Return to Supervisor Routine 382
Chart B40. $$A$SUP1 - SGPMGR Macro, GETREAL (SVC 55) Routine 383
Chart B41. $$A$SUP1 - SGPMGR Macro, GETREAL (SVC 55) Routine 384

Chart B42. $$A$SUP1 - SGPMGR Macro, GETREAL Routine 385
Chart B43. $$A$SUP1 - SGPMGR Macro, FREEREAL (SVC 54) Routine 386
Chart B44. $$A$SUP1 - SGPMGR Macro, TFIX and TFREE Routines 387
Chart B45. $$A$SUP1 - SGPMGR Macro, Post Requesting Task Ready to Run . . .388
Chart B46. $$A$SUP1 - SGPMGR Macro, Dequeue Page Queue Entry 389
Chart B47. $$A$SUP1 - SGPMGR Macro, Reset Fetch 390
Chart B48. $$A$SUP1 - SGPMGR Macro, Initialize Partition (SVC 58) 391
Chart B49. $$A$SUP1 - SGPMGR Macro, PFIX Restart 392
Chart B50. $$A$SUP1 - SGPMGR Macro, PFIX Routine 393
Chart B51. $$A$SUP1 - SGPMGR Macro, PFIX, Find Page Frame 394
Chart B52. $$A$SUP1 - SGPMGR Macro, PFREE Routine (SVC 68) 395
Chart B53. $$A$SUP1 - SGPMGR Macro, PFREE Routine (SVC 68) 396
Chart B54. $$A$SUP1 - SGPMGR Macro, PFIX Routine (SVC 67) 397
Chart B55. $$A$SUP1 - SGPMGR Macro, PFIX Routine (SVC 67) 398
Chart B56. $$A$SUP1 - SGPMGR Macro, PFIX Routine (SVC 67) 399
Chart B57. $$A$SUP1 - SGPMGR Macro, DTFIX Subroutine 400
Chart B58. $$A$SUP1 - SGPMGR Macro, DPFIX Subroutine, Exchange Pages 401
Chart B59. $$A$SUP1 - SGPMGR Macro, Exchange Pages 402
Chart B60. $$A$SUP1 - SGPMGR Macro, Scan Queues for Page Frame 403
Chart B61. $$A$SUP1 - SGPMGR Macro, Perform Page I/O 404
Chart B62. $$A$SUP1 - SGPMGR Macro, DTFREE and DPFREE Subroutines 405
Chart B63. $$A$SUP1 - SGPMGR Macro, Subroutines 406
Chart B64. $$A$SUP1 - SGPMGR Macro, Update Page-In Counters 407
Chart B65. $$A$SUP1 - SGPMGR Macro, Deactivate a Partition 408
Chart B66. $$A$SUP1 - SGPMGR Macro, Deactivate a Partition 409
Chart B67. $$A$SUP1 - SGPMGR Macro, Deactivate a Partition 410
Chart B68. $$A$SUP1 - SGPMGR Macro, Reactivate a Deactivated Partition . . .411
Chart B69. $$A$SUP1 - SGAM Macro, CDLOAD Routine (SVC 65) 412
Chart B70. $$A$SUP1 - SGAM Macro, CDLOAD Routine (SVC 65), GETVIS Routine 413
Chart B71. $$A$SUP1 - SGAM Macro, GETVIS Routine 414
Chart B72. $$A$SUP1 - SGAM Macro, FREEVIS, GETVISB and GTVSINIT Routines .415
Chart B73. $$A$SUP1 - SGPOPT Macro, PAGEOUT (SVC 85/86) Routine 416
Chart B74. $$A$SUP1 - SGPOPT Macro, PAGEOUT (SVC 85/86) Routine 417
Chart B75. $$A$SUP1 - SGPOPT Macro, PAGEIN (SVC 87) Routine 418
Chart B76. $$A$SUP1 - SGPOPT Macro, PAGEIN (SVC 87) Routine 419
Chart B77. $$A$SUP1 - SGPOPT Macro, Validate Parameter List 420
Chart B78. $$A$SUP1 - SGPOPT Macro, Delete PAGETAB Entries if Task Terminates 421
Chart B79. $$A$SUP1 - MCRAS Macro, Machine Check Handler 422
Chart B80. $$A$SUP1 - MCRAS Macro, Determine Machine Check Severity 423
Chart B81. $$A$SUP1 - MCRAS Macro, Emergency Situation Handling 424
Chart B82. $$A$SUP1 - MCRAS Macro, Channel Check Handler 425
Chart B83. $$A$SUP1 - MCRAS Macro, Channel Check Handler 426
Chart B84. $$A$SUP1 - MCRAS Macro, Channel Check Handler 427
Chart B85. $$A$SUP1 - MCRAS Macro, Channel Check Handler 428
Chart B86. $$A$SUP1 - MCRAS Macro, DASD Channel Check Handler 429
Chart B87. $$A$SUP1 - MCRAS Macro, Dequeue RAS I/O Request, ERPIB Queue Scan 430
Chart B88. $$A$SUP1 - MCRAS Macro, ERPIB Queue Scan 431
Chart B89. $$A$SUP1 - MCRAS Macro, RTA Function Scan 432
Chart B90. $$A$SUP1 - MCRAS Macro, Perform Functions for RTA 433
Chart B91. $$A$SUP1 - MCRAS Macro, Perform Function for RTA 434
SGDFCH Macro, Scan System Directory List (SDL) 434
Chart B92. $$A$SUP1 - SGDFCH Macro, Scan System Directory List (SDL) 435
Chart B93. $$A$SUP1 - SGDFCH Macro, Read Directory Blocks 436
Chart B94. $$A$SUP1 - SGDFCH Macro, Sector Convert Routine (Part 1 of 2) . .436
Chart B95. $$A$SUP1 - SGDFCH Macro, Sector Convert Routine (Part 2 of 2) . .436
Chart C01. $$ABERRZ - Physical Attention 437
Chart C02. $$ABERRZ - Physical Attention 438
Chart C03. $$ABERRZ - Physical Attention 439
Chart D01. $$ABERZ1 - Quiesce LTA I/O .440

Chart 00. Disk Operating System Program Flow



The resident version of the IBM Disk Operating System (DOS), System Control, requires 30K for a Model 115 or 125, 32K for a Model 135 or 145, and 34K for a Model 155-II or 158.

The system is disk resident, using an IBM disk storage device for online storage of all programs. Depending on the requirements of the particular application, the system can be expanded to include all processing programs used to perform the various jobs of a particular installation, or it can be tailored to a minimum system to control a single program.

The operating system includes the following components: CPU, input/output channels, input/output control units, input/output devices, microprogramming, system control programs, support programs, user programs, user data files, teleprocessing capability, and multiple programming capability.

The supervisor and physical IOCS are specifically designed for a particular configuration by means of a one-shot assembly (generation time). They require reassembly if the configuration changes.

The supervisor operates with problem programs when job processing (problem program execution) occurs. The supervisor is divided into two parts:

1. The resident part, called the <u>supervisor nucleus</u>;

2. The nonresident part, called <u>supervisor transients</u>.

The nucleus is loaded into real storage at IPL time and remains there throughout job processing. Transients are loaded, as needed, from the core image library. When one transient has finished performing its service, it can be overlaid by another required transient. This technique makes maximum use of real storage allotted to the supervisor.

Real storage locations are reserved in the supervisor area for:

Logical ($$B) transients - LTA (1200 bytes).

Physical ($$A) transients - PTA (1024 bytes).

Recovery ($$R) transients - RTA (1000 bytes).

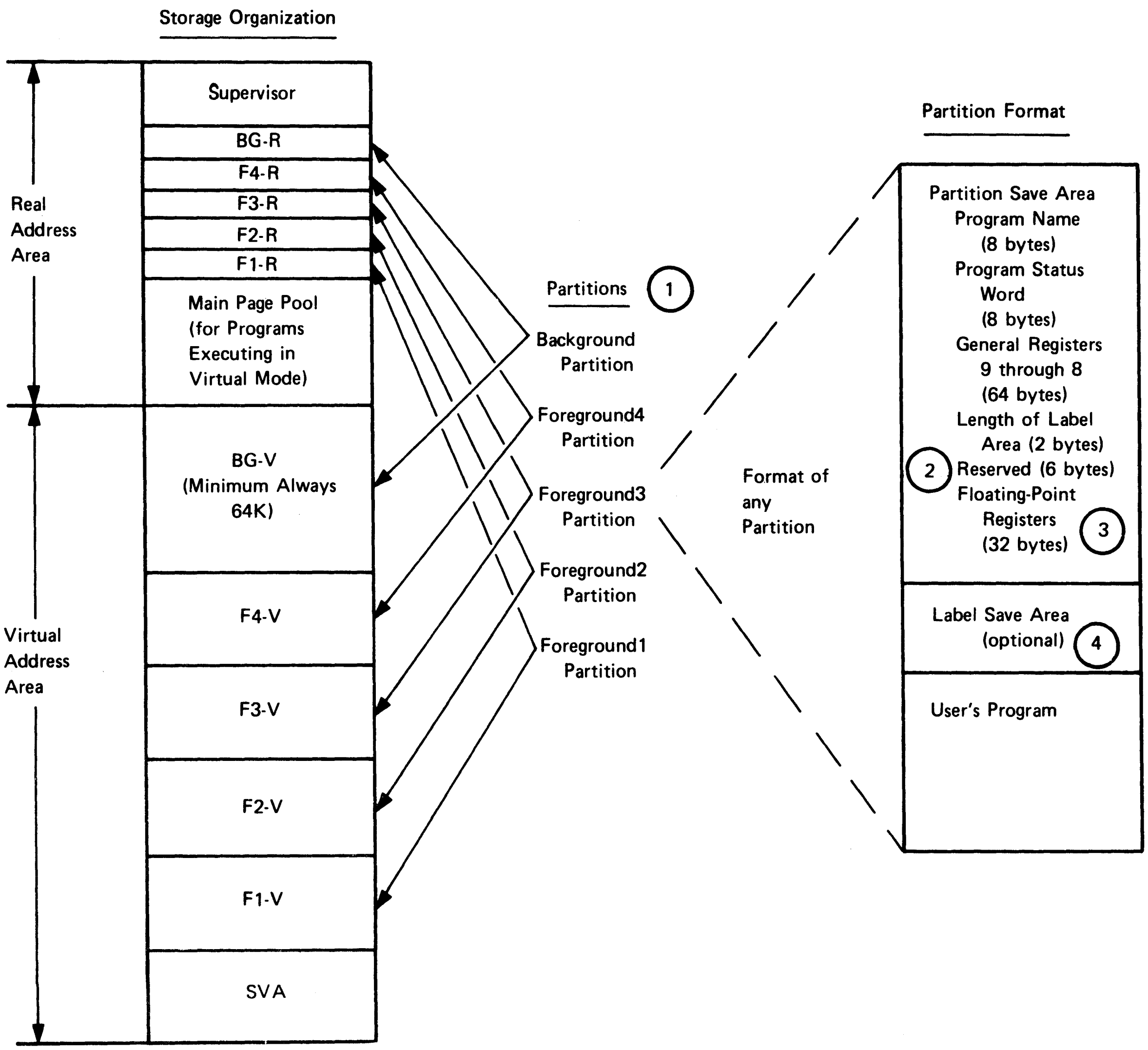
CRT ($$BOCRT) transients - CRTTRNS (952 bytes).

The CRT Transient area is only generated in a supervisor with Display Operator Console (DOC) support.

The basic functions performed by the supervisor are:

- Storage protection.
- Interrupt handling.
- Channel scheduling.
- Device error recovery and recording.
- Machine Check and Channel Check recovery and recording.
- Operator communications.
- Program retrieval (fetch or load).
- End-of-Job processing.
- Timer service.
- Page Management.
- Channel Program Translation.

Each installation must generate its own custom-made supervisor by means of a one-shot assembly. Supervisor generation macros control the generation of the supervisor control program. The user must reassemble the supervisor if its functions are to be modified (for example, when an installation configuration changes).



(1) Up to five partitions may be specified: one background and four foreground. Each partition consists of the pair 'real partition - virtual partition'.

An active virtual partition comprises at least 64K bytes in the virtual address area; the number of bytes is always a multiple of 2K.

An active real partition comprises any multiple of 2K bytes in the real address area; whenever a real partition is active, the corresponding virtual partition is also active.

An inactive partition, whether real or virtual, may comprise any multiple of 2K bytes or may be reduced to zero.

The virtual background partition (BG-V) is always active and therefore always comprises at least 64K bytes.

(2) Job start time, for time stamp, is stored in last 4 bytes of this area (bytes 84-87) when specified.

(3) Floating point register save area is required only when floating point feature is specified for supervisor generation.

(4) Standard tape labels = 80 bytes; sequential DASD and DTFPH (MOUNTED = SINGLE) = 0 bytes; and DTFIS, DTFDA, and DTFPH (MOUNTED = ALL) = 84 bytes + 20 bytes per extent statement.

Figure 1.1. Multiprogramming Storage Organization

Storage area	Base Registers
HARDWARE/SOFTWARE INTERFACE (PSWs, Logout Areas, etc.)	R0, R11, R12, R13
TABLES AND CONSTANTS, that <u>must</u> be below 4K. Contained in FOPT macro (System Communications Region, Background Communication Region, etc.)	R0, R11, R12, R13
CANCEL, GENERAL ENTRY, TASK SELECTION; these routines <u>must</u> be below 4K. Contained in FOPT macro. CHANNEL SCHEDULER, SVC ROUTINES AND AP ROUTINES; need <u>not</u> be below a certain boundary.	R0, R11, R12, R13
Some CONSTANTS that <u>must</u> be below 16K.	R0, R11, R12, R13
SAVE AREAS, OPTION TABLES, PIB TABLES, I/O TABLES, SYSTEM TASK BLOCKS, FOREGROUND COMMUNICATION REGIONS, etc., having Y-type address pointers in low storage. <u>Must</u> be below 32K.	
I/O INTERRUPT HANDLER, DISK ERP	R9,R10
EXTERNAL INTERRUPT HANDLER	R14
TABLES, SAVE AREAS, TRANSIENT AREAS, having A-type address pointers in low storage (System Task Save Area, Boundary Box, PTA, LTA, etc.)	* Up to this point addressability should be possible with Y-type address constants
FETCH ROUTINE	R6
CCW TRANSLATION ROUTINES	R9
PAGE MANAGEMENT ROUTINES	R14, R15, R13
VSAM ROUTINES	R14
MACHINE CHECK HANDLER, CHANNEL CHECK HANDLER, RAS MONITOR, RTA	R15
PAGE MANAGEMENT TABLES, CCW TRANSLATION COPY BLOCKS	

Figure 1.2. Supervisor Storage Allocation

SUPERVISOR GENERATION

The supervisor is assembled with a series of macros that describe the installation's functional requirements and its configuration. At system generation time, a source deck containing the supervisor generation macros is assembled into an object deck.

The list of supervisor macros and optional operands in Figure 2.2 gives the reader:

- Supervisor generation macro names.
- Required macro sequence (as listed in Figure 2.2).
- Macro parameters. (Where there is an assumed value, that value is underlined.)
- A brief description of what the generated macro does.
- A brief description of what the individual parameter options do.

The code generated by the assembler for any selected supervisor generation is a function of the generation macros described in Figure 2.2 and of a group of inner macros called by the generation macros. The specific instructions assembled depend on the global settings.

The most important global values used in supervisor generation are the B-globals. Therefore, this subsection emphasizes the generation macros that establish B-global values. However, some A-globals that are tested in the same manner as B-globals are also described in this subsection. C-globals are not described. Two figures in this subsection show macro relationships. Figure 2.3 shows the code generated, if any, and the globals set, if any. Figure 2.4 indicates the on/off conditions of the globals.

SUPERVISOR ORGANIZATION

The physical organization of the supervisor depends on the sequence of the supervisor generation macros. The sequence is predetermined and cannot be changed by the user. The logical organization depends on the parameter options selected at generation time. Figure 1.2 is a map of the assembled supervisor in real storage, and illustrates the supervisor physical organization.

The logical organization is not described in this manual because of the variety of options available. You must determine the logical organization for individual supervisor generations. By using the program level flowcharts to point to the detailed flowcharts, you may select the correct group of flowcharts for the desired generation.

LOW REAL STORAGE

The allocation of the first 512 bytes of real storage is standard for any IBM System/370 CPU:

Byte (Hex)	Function
0-13	(Reset to zeros after IPL)
0-3:	Hard Wait Message codes (MCH, CCH, IPL). See Figure 2.1.
0-4:	Device error message codes in case of I/O error, and SYSLOG disabled. See Figure 2.1.
10-13	In a system with VTAM, the address of the VTAM ACTVT.
14-17	Address of Communications Region for Active Partition
18-1F	External Old PSW
20-27	SVC Old PSW
28-2F	Program Check Old PSW
30-37	Machine Check Old PSW
38-3F	I/O Old PSW
40-47	CSW
48-4B	CAW

Byte Hex	Function
4C-4F	Interval Timer Savearea
50-53	Hardware Timer
54-57	Save area for Interval Timer
58-5F	External New PSW
60-67	SVC New PSW
68-6F	Program Check New PSW
70-77	Machine Check New PSW
78-7F	I/O New PSW
80-83	Address of SCP Communications Region
84-85	External Interrupt Information
86-87	External Interrupt Code
88-89	SVC Interrupt Information
	Byte 89, bits 5 and 6: Instruction Length Code
8A-8B	SVC Interrupt Code
8C-8D	Program Check Interrupt Information
8E-8F	Program Check Interrupt Code
90-93	Address that Caused Page Fault
94-B7	Reserved
B8-B9	Interrupt Information
BA-BB	I/O Address on Interrupt
BC-E7	Reserved
E8-EF	Machine Check Interrupt Code (MCIC)

Model-Independent Log-Out Area:

F0-F7	Reserved
F8-FB	Failing-Storage Address
FC-FF	Region Code
100-15F	Fixed Log-Out Area
160-17F	Floating-Point Register Save Area
180-1BF	General Register Save Area
1C0-1FF	Control Register Save Area

Beyond byte 511, storage allocation depends on the CPU model.

Model-Dependent Log-Out Area:

200-5DF for System/370 Models 155-II/158

200-49F for System/370 Model 145

200-317 for System/370 Model 135

The remainder of this chapter contains only figures. Figures 2.2 - 2.4 provide information about supervisor generation macros, parameters and globals. Figure 2.5 and following figures concern communications regions, information tables, option tables, etc. It should be noted that figures for tables etc., that are closely related to routines described in the following chapters, are to be found in those chapters.

Byte 0	Byte 1	Byte 2	Byte 3	Explanation
MCH/CCH/IPL Hard Wait Codes 3				
X'C1'	X'E2'	A,I,S	Not used	Irrecoverable machine check.
X'C2'	X'E2'	Not used	Not used	Irrecoverable channel failure during RMS fetch.
X'C3'	X'E2'	A,I,S	Not used	Channel failure on SYSLOG when RMS message scheduled.
X'C4'	X'E2'	A,I,S	Not used	No ECSW stored.
X'C5'	X'E2'	A,I,S	Not used	Channel failure; ERPIBs exhausted.
X'C6'	X'E2'	A,I,S	Not used	Channel failure; two channels damaged or a damaged channel situation occurred while RMS was executing an I/O operation.
X'C7'	X'E2'	A,I,S	Not used	Channel failure; system reset was presented by a channel.
X'C8'	X'E2'	A,I,S	Not used	Channel failure; system codes in ECSW are invalid.
X'C9'	X'E2'	A,I,S	Not used	Channel failure; channel address invalid.
X'D1'	X'E2'	A,I,S	Not used	Irrecoverable channel failure on SYSVIS.
X'07'	X'E6'	Channel	Unit or X'00'	IPL I/O error or Equipment Malfunction, condition code 2, during STIDC. Channel and unit indicate whether SYSRES or communication device. When byte 3 = X'00', byte 2 indicates the channel for which STIDC was issued. Re-IPL system.
Other Hard Wait Codes (also contained in Register 11)				
X'00000F'			X'FF'	Program Check in Supervisor or SDAID 3 .
X'00000F'			X'FE'	I/O error during fetch from System CIL.
X'00000F'			X'FD'	Channel check (only Model 115/125, with MCH=NO and RMS=NO).
X'00000F'			X'FC'	Machine Check (only Model 115/125, with MCH=NO and RMS=NO).
X'00000F'			X'FB'	Page Fault in Supervisor routine with identifier RID=X'00' (see Figure 4.02).
X'00000F'			X'FA'	Translation Specification Exception.
X'00000F'			X'F9'	Error on Paging I/O.
X'00000F'			X'F8'	CRT phase not found.
X'00000F'			X'F7'	No copy blocks available for BTAM appendage I/O request.
X'00000F'			X'F6'	$MAINDIR canceled during system CIL update. If this occurs, the system CIL is only partially updated and must be corrected before use. This hard wait condition can also occur if the FETCH QUEUE BIT (FCHQ) is set in the linkage control byte in the partition communication region owned by the terminating partition.

Figure 2.1. Error Bytes in Low Real Storage (Part 2 of 2)

Byte 0	Byte 1	Byte 2	Byte 3	Explanation
Device Error Recovery Wait Codes				
X'08' to X'60'	X'C1' or X'C4'	Channel	Unit	Error recovery messages. Refer to OP messages in DOS/VS Messages.
SDAID Hard Wait Code				
X'61'	X'E6'	Channel	Unit	Another device is running burst mode on same channel as SDAID output device. Re-IPL system.
SDAID Soft Wait Code				
X'62'	X'C5'	Not used	Not used	SDAID output device became unready. Make printer ready and press the EXTERNAL INTERRUPT Key.
X'00'	X'00'	X'00'	X'00'	SDAID stop on event. To continue, press the EXTERNAL INTERRUPT Key.

Notes:

1 A (X'C1') = SYSREC recording unsuccessful.
  I (X'C9') = SYSREC recording incomplete.
  S (X'E2') = SYSREC recording successful.

2 S (X'E2') = Run SEREP.

3 SDAID waits are identified by X'EEEE' in the address part of the wait PSW.

Figure 2.1. Error Bytes in Low Real Storage (Part 2 of 2)

Macro Name and Purpose	Parameter=Option	Option Description
SUPVR		System residence (SYSRES) must be on a disk device.
Describes System Environment		
	NPARTS= {1 / 3 / n}	Specifies the number of partitions to be supported. The maximum value for n is 5. The default value is 1 if the AP parameter is omitted, or if AP=NO. The default value is 3 if AP=YES. If POWER=YES is specified, NPARTS=2, 3, 4, or 5 must be specified unless AP=YES is specified. A communications region is generated for each partition supported.
	TP= {NO / VTAM / BTAM[,VTAM] / QTAM[,VTAM] / QTAMn[,VTAM]}	Specifies the Teleprocessing Access Method (Basic or Queued). QTAM includes BTAM support. If AP=YES, n specifies the maximum number of active QTAM message processing programs in the system. From 2 to 12 may be specified. The default value for n is 2. If AP=NO, n is always 2. If QTAM is specified, then NPARTS=2, 3, 4, or 5 must be specified unless AP=YES is specified. If BTAM runs virtual, PFIX=YES is required.
	ERRLOG= {YES / RDE}	The Recovery Management Support Recorder (RMSR) to record device errors on the System Recorder file (IJSYSRC) is always supported. If RDE is specified, the Reliability Data Extractor is also supported. RDE gathers hardware reliability data to be used by IBM personnel in evaluating machine performance. For Model 115 or 125, specify only if RMS=YES and/or CHAN=YES.
	MICR= {NO / 1419 / 1419D}	Indicates support for magnetic ink or optical character reader/sorters. The specification 1419 indicates support for 1419s with Single Address Adapter, 1255s, 1259s, or 1270s. 1419D specification gives support for 1419s with Dual Address Adapter, or 1275s. If support for burst-mode devices on multiplexer channels is generated (FIOCS macro parameter BMPX=YES), MICR devices must not be attached to the same channel as burst-mode devices.
	AP= {NO / YES}	Indicates Asynchronous Processing support for dependent programs to run concurrently within a partition (multitasking). If YES is specified, multiprogramming and WAITM macro support are generated whether specified or not. An invalid specification of AP causes AP=YES to be assumed. If AP=YES is specified and NPARTS is omitted or specifies 1, NPARTS=3 is forced. AP=YES must be specified if VTAM is specified in the TP parameter.
	EU= {NO / YES / RELOC}	EU=YES or RELOC generates support for mixed-parity tape processing in all partitions for the emulators, the tape preprocessor and tape postprocessor programs.
	POWER/VS= {NO / YES}	Specifies support for POWER/VS. If YES is specified: • NPARTS=3 is forced if NPARTS is omitted or specifies1,

Figure 2.2. Supervisor Macros (Part 1 of 11)

Macro Name and Purpose	Parameter=Option	Option Description
		• ECPREAL=YES is forced, • PHO=YES and consequently PFIX=YES are forced.
SUPVR (Continued)	ASCII={NO, YES}	Indicates whether the supervisor is to support the American National Standard Code for Information Interchange (ASCII). YES adds 512 bytes to the supervisor size.
	PHO={NO, YES}	Specifies if the supervisor is to support page fault handling overlap. POWER=YES forces PHO=YES provides support for the valid use of the SETPFA macro by programs that execute in virtual mode and perform their own multitasking functions. PHO=YES forces PFIX=YES.
	PAGEIN={NO, n}	Specifies that paging activity is to be controlled by means of the macros PAGEIN, RELPAG, and FCEPGOUT. (For more details about these macros see DOS/VS Supervisor and I/O Macros). The value n indicates the maximum number of page-in requests that can be queued for execution. Each PAGEIN macro issued in a program represents a page-in request. To obtain the support for the three macros, the value n must be 1 or higher.
	The following parameters are for Model 115 or 125 only; the support they generate is always provided for other models by the CONFG macro specification.	
	CHAN={NO, YES}	Generates RMSR support for all channel-attached devices. Required if tapes, TP devices, or channel-attached devices are to be used. If such devices are present in the system, IPL will succeed only if the RMSR support is generated.
	MCH={YES, NO}	Specifies whether the MCAR and CCH functions are to be supported.
	RMS={NO, YES}	Specifies whether the entire RMS package (MCAR, CCH, and RMSR) is to be generated. RMS=YES overrides CHAN=NO and MCH=NO.
CONFG Describes Hardware Features	MODEL={115, 125, 135, 145, 155-II, 158}	Defines the System/370 model number. Full Recovery Management Support (RMS) is generated except for Model 115 or 125. For these models, the generation of RMS is controlled by the CHAN and RMS parameters of the SUPVR macro. Omission of this parameter, or an invalid specification, causes generation to be terminated.
	FP={NO, YES}	Specifies whether the floating-point feature is to be supported.
STDJC Sets standard values for job control variables	DECK={YES, NO}	Output of object modules of language translators on SYSPCH.
	LIST={YES, NO}	Source modules listings from language translators on SYSLST.
	XREF={YES, NO}	Language translators output symbolic cross-reference lists on SYSLST.

Figure 2.2. Supervisor Macros (Part 2 of 11)

Macro Name and Purpose	Parameter=Option	Option Description
STDJC (Continued)	ERRS={YES / NO}	Compilers summarize all errors in source programs on SYSLST.
	LOG={YES / NO}	Listing of all control statements on SYSLST.
	DUMP={YES / NO}	Dump of registers and partition storage on SYSLST.
	LINES={56 / nn}	Number of lines per printed page on SYSLST (nn must be between 30 and 99).
	DATE={MDY / DMY}	Format of date.
	CHARSET={48C / 60C}	Specifies the 48- or 60-character set for PL/I translator input on SYSIPT.
	LISTX={NO / YES}	Hexadecimal object module listings from compilers on SYSLST.
	SYM={NO / YES}	To be specified if the PL/I compiler is to produce a symbol and offset table listing or if ANS COBOL is to produce a data division glossary.
	SPARM={NO / YES}	Support of assembler variable symbol &SYSPARM.
	EDECK={NO / YES}	Output of edited macros on SYSPCH.
	ALIGN={YES / NO}	Support of assembler data-alignment function.
	ACANCEL={NO / YES}	Specifies standard ACANCEL/NOACANCEL option.
FOPT Describes Functional Supervisory Options	OC={NO / YES}	Operator-initiated communications to problem programs. If OC=YES, the facility is available for all main tasks in multiple partitions. Must be YES if emulator is used. OLTEP=YES and RETAIN=YES force OC=YES.
	PC={NO / YES}	Problem program routine for program check. If PC=YES, the facility is available for all tasks in multiple partitions.
	IT={NO / YES}	Problem program ability to set timer intervals and to specify a timer interrupt routine. IT=YES generates timer support for all tasks in multiple partitions if EG20 is on (see Figure 2.4).
	AB={NO / YES}	Provides support for user abnormal termination exit routine. If AB=YES, the facility is available for all tasks in multiple partitions. VSAM=YES forces AB=YES.
	TEB={NO / n}	Error statistics on tape cartridge readers are to be accumulated and logged. n is the number of tape cartridge readers (maximum 255) attached to the system.

Figure 2.2. Supervisor Macros (Part 3 of 11)

Macro Name and Purpose	Parameter=Option	Option Description
FOPT (Continued)	TEBV={IR, CR}	Specifies whether individual (IR) or combined (CR) recording of error statistics is required for unlabeled or nonstandard 2400 or 3400 series tapes.
	EVA={NO, (r,w)}	Specifies support for error by volume analysis for 2400 and 3400 series tapes. r is the read error threshold. w is the write error threshold.
	SKSEP={NO, YES, n}	Specifies if SEEKs are to be separated from the remainder of channel programs. SKSEP=YES generates a Seek Address Block (SAB) table entry for each DASD specified by the DVCGEN macro. SKSEP=n specifies the number of entries to be generated; n must not be less than the number of DASD specified by the DVCGEN macro and not more than 254.
	RPS={NO, YES}	Provides support for the Rotational Position Sensing (RPS) capabilities of DASD devices supporting the feature. Since RPS provides channel overlap during a Seek operation as well as RPS operations, SKSEP need not be specified for devices with the RPS feature. If direct access storage devices without the RPS feature are configured in the system, SKSEP can be specified. RPS=YES forces GETVIS=YES and RELLDR=YES in the FOPT macro and BLKMPX=YES in the PIOCS macro.
	PD={NO, YES, n}	Specifies the number of bytes to be allocated to PDAID routines. PD=YES reserves 1400 bytes which is also the minimum value for n.
	CBF={NO, n}	Specifies whether output to a console printer assigned as SYSLOG is to be buffered. n specifies the number of buffers to be generated. Accepted values are from 1 to 50. An invalid specification causes 1 buffer to be generated. CBF=NO is forced if MODEL=115 or 125.
	IDRA={NO, YES}	Specifies whether the Independent Directory Read-in Area is to be generated in a supervisor that supports multiprogramming. IDRA reduces contention for the PTA in fetch operations when the ERP system task is active.
	TRKHLD={NO, n}	Specifies whether the Track-Hold feature is to be supported for DASD in a supervisor that supports multiprogramming. n indicates the maximum number of tracks to be held at any one time. Accepted values are from 1 to 225. A Track-Hold Table (THTAB) with n entries is generated in the supervisor. An invalid specification results in 10 entries being generated.
	RETAIN={NO, YES}	Specifies support for the data link to the Remote Analysis Center.

Figure 2.2. Supervisor Macros (Part 4 of 11)

Macro Name and Purpose	Parameter=Option	Option Description
	DASDFP={(n,n[,{NO | 2311 | 2314 | 3330 | 3340 | 2321}])}	Specifies support for protection of DASD files. n,n indicates the range of channels to which the devices may be attached. Specifying 2311, 2314, 3330, or 3340 generates support for 2311s, 2314s, 2319s, 3330s, and 3340s. Specifying 2321 generates support for all DASD devices.
	WAITM={NO | YES}	Specifies support for multiple waits (WAITM macro). If AP=YES is specified in the SUPVR macro, WAITM=YES is forced.

Figure 2.2. Supervisor Macros (Part 4 continued)

Macro Name and Purpose	Parameter=Option	Option Description
FOPT (Continued)	ERRQ={3 | 5 | n}	Specifies the number of entries to be generated for the error queue. Without multiprogramming, n may be from 3 to 25. With multiprogramming support, n may be from 5 to 25. The lower value is the default value in each case.
	VSAM={NO | YES}	Specifies whether the Virtual Storage Access Method is to be supported. VSAM=YES forces AB=YES, GETVIS=YES, and TOD=YES. If not specified, VSAM=NO is assumed.
	GETVIS={NO | YES}	Provides for the valid use of GETVIS and FREEVIS macros which reserve or free parts of a virtual partition during program execution. This facility is required when VSAM=YES. GETVIS=YES forces RELLDR=YES.
	SYSFIL={NO | YES | (YES,n1,n2)}	Specify if system input and system output files (SYSRDR, SYSIPT, SYSLST, SYSPCH) in any partition may be assigned to a disk device or diskette, or if extended support for the procedure library is desired. Specification of YES gives support for all disk devices including diskette. n1 indicates the residual capacity (in records) for beginning of operator notification when SYSLST is assigned to a disk device. n2 indicates the same for SYSPCH. Acceptable values for n1 and n2 are from 100 to 65535; if either is omitted, 1000 is assumed. For diskette this notification is not supported.
	JA={NO | YES | (n1,n2,n3,n4,n5)}	Generates Job Accounting Interface support. YES specifies basic support (SIOs not counted). (n1,n2,n3,n4,n5) specifies the number of devices per partition for which SIOs are to be counted. The maximum value of n for any partition is 255; the default value is 0. n1 always specifies the number for the background partition. The partitions to which n1 through n5 refer depend on the number of partitions in the system, as follows: n2 n3 n4 n5 2 partitions F1 3 partitions F2 F1 4 partitions F3 F2 F1 5 partitions F4 F3 F2 F1
	JALIOCS={NO | n1,n2}	Generates a user save area and an alternative label area for job accounting. n1 specifies the number of bytes for the save area for user-written job-accounting routines. The maximum acceptable value is 1024 and the default value is 16. n2 specifies the number of bytes for the alternative label area. The maximum is 244 and the default value is 0.

Figure 2.2. Supervisor Macros (Part 5 of 11)

Macro Name and Purpose	Parameter=Option	Option Description
FOPT (Continued)	ZONE={NO, {EAST, WEST},hh,mm}	Indicates the difference between Greenwich Mean Time and local time to obtain the local time of day. NO is assumed if the parameter is omitted or is invalid, or if TOD=NO is specified.
	TOD={NO, YES}	Generates support for the system Time-of-Day clock. VSAM=YES forces TOD=YES.
	PCIL={NO, YES}	Generates support for the private core image library.
	RELLDR={NO, YES}	Generates support for relocating relocatable phases. YES is forced by VSAM=YES, CLTEP=YES, and GETVIS=YES.
	PRTY=(p1[,p2[,p3[,p4[,p5]]]])	Specifies the dispatching priorities of partitions in ascending order. The number of operands must be equal to the number of partitions in the system. The default sequence is BG,F4,F3,F2,F1; that is, BG has the lowest priority and F1 has the highest priority.
	OLTEP={YES, NO}	Generates support for the online testing function. The default value is YES. OLTEP=YES forces RELLDR=YES and OC=YES. RETAIN=YES forces CLTEP=YES.
	PFIX={NO, YES}	Generates support for the valid use of the PFIX macro to fix pages of virtual mode programs in real storage frames and subsequently to free them by PFREE macros. PHO=YES forces PFIX=YES.
	DOC={NO, 125D}	Specifies whether support is required for Display Operator Control (DOC). The default is NO if MODEL=135, 145, 155-II or 158. For MODEL=115 or 125, DOC=125D is forced.
	SLD={5, n}	Specifies the number of entries in the Second Level Directory. The minimum value for n is 5. This value is assumed in case of an incorrect specification. A performance decrease will result if the number specified is less than the number of actually used directory tracks of the System Core Image Library.
	PSLD={NO, n}	Specifies the number of entries in the Private Second Level Directories. A PSLD is created for each partition. The minimum value that may be specified for n is 5. NO is assumed in case of an incorrect specification. A performance decrease will result if the number specified is less than the number of actually used directory tracks of a Private Core Image Library. PCIL=NO forces PSLD=NO.
	ECPREAL={NO, YES}	Specifies support for REALAD and VIRTAD. POWER=YES forces ECPREAL=YES.

Figure 2.2. Supervisor Macros (Part 6 of 11)

Macro Name and Purpose	Parameter=Option	Option Description
PIOCS Describes the System I/O Configuration	BMPX={NO, YES}	Specifies whether burst-mode devices are to be supported on a byte multiplexer channel.
	BLKMPX={NO, YES}	Specifies whether support is to be generated for block multiplexer channels to run in block multiplex mode. Must be NO for disk emulation. RPS=YES forces BLKMPX=YES.
	CHANSW={NO, TSWTCH, RWTAU}	Specifies whether channel switching is to be supported for devices on selector or block multiplexer channels. RWTAU=2404 or 2804; TSWTCH=2816 (with a 2403 or 2803) or 3803.
	TAPE={7, 9, NO}	Specifies whether PIOCS is to support tape. Support is generated for 7-track and 9-track tape unless NO is specified.
	MRSLCH={NO, YES}	Specifies whether a MICR-type device (magnetic ink or optical reader/sorter) is to be assigned to a selector channel.
	DISK={2311, 2314, 3330, 3340, (3330,3340)}	Specifies the type of disk devices to be supported. If this parameter is omitted or 2311 or 2314 is specified, support is generated for 2311, 2314, and 2319 devices. Specifying 3330 or 3340 also generates support for 2311 and 2314. For Model 125, 3330 is forced, if neither 3330 nor 3340 is specified. For Model 115, 3340 is forced.
VSTAB Describes Real and Virtual Storage and the Channel Program Translation Copy Blocks	RSIZE=nK	Specifies the maximum size of real storage. n may be up to 8192 and must be a multiple of 2. If not specified, or specified incorrectly, 96 is assumed.
	VSIZE=nK	Specifies the size of virtual storage including SVA. n must be a multiple of 2, larger than 128 and not larger than 16384 minus the value specified for n in RSIZE. If not specified, or incorrectly specified, the value assumed is 64 x NPARTS plus size of SVA.
	BUFSIZE=n	Specifies the number of copy blocks to be generated for the Channel Program Translation routine. The minimum value for n is 10, and the default value is 10 if NPARTS=1 and 30 if NPARTS=2, 3, 4, or 5.
	SVA={(64K,0K), (nK,mK)}	nK specifies the size of the SVA, mK specifies the size of the system GETVIS area in the SVA. n must be a multiple of 2 and at least 64. m must be a multiple of 2, less than n. In case of incorrect specification, the default values are taken. This parameter is only valid if the NPARTS parameter of the SUPVR macro specifies more than one partition (NPARTS=>1).

Figure 2.2. Supervisor Macros (Part 7 of 11)

Macro Name and Purpose	Parameter=Option	Option Description
ALLOC	F1=nK[,F2=nK[,F3=nK[,F4=nK]]]	
Partitions Virtual Storage for Multiprogramming		Specifies virtual storage partitioning where n is at least 64 and is a multiple of 2, or may be 0. The total storage allocated to virtual foreground partitions plus 64K must not exceed the size of the virtual storage specified by the VSIZE parameter of the VSTAB macro minus the size of the SVA. For any foreground partition the default value is 0. If any value from 2 to 62 is specified, the default value is taken.
ALLOCR	BGR=nK[,F1R=nK[,F2R=nK[,F3R=nK[,F4R=nK]]]]	
Partitions Real Storage for Multiprogramming		Specifies real storage partitioning where n is a multiple of 2 or may be zero. The total storage allocated to real partitions must not exceed the size of real storage minus the size of the supervisor and the main page pool.
IOTAB Describes installation requirements for I/O tables	IODEV={10, n}	Specifies the number of I/O devices attached to the system. The maximum allowable specification is 255, and the minimum is 5.
	BGPGR={10, n*}	Specifies the number of programmer logical units (SYSnnn) for the BG partition. The minimum value for n is 10, while the maximum depends on the number of partitions (see note). A partition LUB table is generated with a two-byte entry for each system logical unit and additional entries for the number of units specified by n.
	F1PGR={5, n*}	Specifies the number of programmer logical units for the F1 partition. The minimum value for n is 5, while the maximum depends on the number of partitions (see note). A partition LUB table is generated with entries for system logical units and for the number of programmer logical units specified by n.
	F2PGR={5, n*}	Specifies the number of programmer logical units for the F2 partition.
	F3PGR={5, n*}	Specifies the number of programmer logical units for the F3 partition.
	F4PGR={5, n*}	Specifies the number of programmer logical units for the F4 partition.

*Note: The maximum values for the sum of all are as follows:

	Number of Partitions				
	1	2	3	4	5
maximum value for sum of n's	241	227	213	199	185

For an unsupported partition n=0 is assumed.

Figure 2.2. Supervisor Macros (Part 8 of 11)

Macro Name and Purpose	Parameter=Option	Option Description
IOTAB (Continued)	CHANQ={6,n}	Specifies the number of 8-byte entries to be generated for the channel queue. At least six entries are always generated, but if more than three partitions are specified by the NPARTS parameter of the SUPVR macro, then this minimum is increased by two for each additional partition and is further increased by the number of console buffers specified by the CBF parameter in the FOPT macro.
	D2311={0,n} D2314={0,2,n} D2321={0,n} D2400={0,4,n} D3330={0,2,n} D3340={0,2,n} D3410={0,2,n} D3420={0,n} D3540={0,n} D3886={0,n}	Specifies the number of each type of special device attached to the system. The number specified for each device may be more than the actual number attached, but if the total number of special devices specified exceeds the number of devices specified in the IODEV parameter generation is terminated. Dxxxx can have two functions: 1. For each special device specified, a PUB table extension (PUB2) entry is generated of appropriate length for that device. 2. For devices on a block multiplexer channel running in block multiplex mode additional CCWs are generated. Default Options:

Models / Dxxxx	M115	M125	Other Models
2314	0	0	2
3330	0	2	0
3340	2	0	0
2400	0	0	4
3410	2	2	0

Macro Name and Purpose	Parameter=Option	Option Description
	JIB={5,n}	Specifies the number of Job Information Blocks (JIBs) for the system. Requirements are: 1. One JIB for each temporary logical unit assignment. 2. One JIB for each alternate logical unit assignment. 3. One JIB for each open 2311 or 2314 extent with the DASD file protect feature. 4. Two JIBs for each open 2321, 3330, and 3340 extent with the DASD file protect feature.

Figure 2.2. Supervisor Macros (Part 9 of 11)

Macro Name and Purpose	Parameter=Option	Option Description
IOTAB (Continued)	SSLNS={4 | n} (default 4)	Specifies the maximum number of Start/Stop lines for the Model 115 or 125 with the Integrated Communications Adapter (ICA). The minimum specification is 0 and the maximum is 16. The specified value is entered in the first byte of the Line Mode Table (LMT) and the table is generated with this number of entries for Start/Stop line mode settings. If the parameter is omitted when TP=BTAM or QTAM and MODEL=115 or 125, then n defaults to 4.
	BSCLNS={1 | n} (default 1)	Specifies the maximum number of BSC lines for Model 115 or 125 with Integrated Communications Adapter (ICA). The minimum specification is 0 and the maximum is 6. The specified value is entered in the second byte of the Line Mode Table (LMT) and the table is generated with this number of entries for BSC line mode settings. If the parameter is omitted, n defaults to 1.
DVCGEN Specifies I/O devices. Each device type requires a separate DVCGEN macro.	CHUN=X'cuu'	Specifies the hexadecimal number for the channel and unit for the device. When channel switching is supported for the device c should indicate the lowest channel of the two.
	DVCTYP=xxxxxx	Specifies the device type (refer to "Appendix C").
	CHANSW={NO | YES} (default NO)	Specifies whether the device is attached to two selector or block multiplexer channels.
	MODE={X'ss' | X'ssss' | X'ssssss'}	1. For tapes, MODE specifies the tape mode as listed in Figure 10.2. For 9-track units the default value is X'C0'. For 7-track units the default value is X'90'. 2. For teleprocessing lines: When DVCTYP=2702, MODE designates the SADxxx command as follows: X'00' = SAD0 (default value) X'01' = SAD1 X'02' = SAD2 X'03' = SAD3 When DVCTYP=2703, MODE specifies the line mode setting for a start/stop line or a BSC line. The setting is three bytes long. 3. When DVCTYP=2260(local) or 3270(local) and CHUN=X'cuu' refers to a 1053, 3284, or 3286 printer attached to a 2848, or either a 3284 or 3286 attached to a 3272, MODE=X'01' specifies a 1053. 4. When DVCTYP=3705, MODE specifies the channel adapter as follows: X'01' = type 1 channel adapter X'02' = type 2 channel adapter No default is assumed.

Figure 2.2. Supervisor Macros (Part 10 of 11)

Macro Name and Purpose	Parameter=Option	Option Description
DVGEN Continued)		5. When DVCTYP=1419, 1419p, or 1419S, MODE designates the external interrupt bit associated with magnetic ink or optical character readers. The modes X'01' through X'20' correspond to the external interrupt code in low real storage byte 87, bits 7 through 2 respectively. The corresponding external lines to which the control units are attached are as follows: X'01' = Line 7 X'08' = Line 4 X'02' = Line 6 X'10' = Line 3 X'04' = Line 5 X'20' = Line 2
		6. When DVCTYP=1018, MODE specifies whether the error correction feature is present: X'01' = feature present X'00' = feature not present
ASSGN Sets standard assignments. A separate macro is required for each standard assignment. (Optional macro)	SYSxxx,X'cuu' [,BG / F1 / F2 / F3 / F4] [,H1 / H2] (BG and H1 underlined as defaults)	SYSxxx is a system or programmer logical unit. X'cuu' is the hexadecimal number of the channel and unit to which the logical unit is assigned. When channel switching is supported for the device, c should be the lower channel of the two. BG, F1, F2, F3, or F4 indicates the partition to which the assignment is being made. H1 or H2 indicates the desired hopper of a 2560 or 5425 for device-independent files associated with SYSIPT, SYSRDR, SYSIN, or SYSPCH, and is otherwise ignored. If both hoppers are used, they must be separately assigned to the same partition.
DPD Defines the residence of the page data set (SYSVIS) required for virtual storage support (Optional macro)	UNIT=X'cuu'	Specifies the channel and unit number of the device that is to contain the page data set.
	CYL=nnn	Specifies the number of the cylinder where the page data set is to begin.
	VOLID=xxxxxx	Specifies the volume serial number of the disk pack for label checking. If this parameter is not specified in the DPD macro or in the DPD command, the volume serial number is not checked.
SEND Indicates end of supervisor generation	[n]	Specifies the beginning real address of the problem program area. An area should be reserved for supervisor expansion and maintenance; otherwise, no area is reserved beyond the assembled last address of the supervisor.

Figure 2.2. Supervisor Macros (Part 11 of 11)

Macro	Type	Code Generated	Critical Globals Set		
SUPVR	generation	Defines low real storage and provides the entry points to the supervisor on interrupts.	BG13 BG20 BG21 BF24 BG27 BG35 BG36 BG39	BG50 BG51 BG92 BG93 BG94 BG100 BG1QT	BGCEAPP BGDAT0 BGPOW AG11 AG1QT MAX NPART
CONFG	generation	None.	BG23 BG39	AG11 AG12	AG56
STDJC	generation	None.	BG34 BG41 BG42 BG43 BG44	BG45 BG46 BG47 BG48 BG49	BG81 BG86 BG87 AG41
FOPT	generation	General entry and exit routines. All first level interrupt routines. Constants and tables.	BG4 BG6 BG7 BG8 BG12 BG14 BG15 BG16 BG26 BG32 BG33 BG38 BG40 BG60 BG62 BG71 BG72	BG73 BGCEAPP BGDAT1 BGDAT2 BGETVIS BGMTI BGRELDR BGTOD BGVSAM FP2321 AG2 AG7 AG10 AG14 AG16 AG17 AG18	AG19 AG27 AG28 AG2A AG31 AG32 AG56 AGJA(1) AGJA(2) AGJA(3) AGJA(4) AGJA(5) AGJAL(1) AGJAL(2) AGZN ITKEY NSLD(1) NSLD(2)
PIOCS	generation	Address validation routine. Some supervisor call rcutines. Attention routine.	BGBLKMP BG3 BG9	BG11 BG12 BG1ST	MG01 MG02 AG54
SGSVC	inner	Various superisor calls.	None.		
IJLQTSVC	inner	Generates SVC 31 for QTAM.	None.		
SGTHAP	inner	Track hold, track free, set abnormal exit routine address, and asynchronous processing rcutines.	None.		
VSTAB	generation	Defines size of real and virtual address area.	BG99 BGDAT4 NSVAPGS	AGDAT1 AGDAT2 AGDAT3	AGDAT5(1) AGDAT6(1)
ALLOC	generation	None.	BG99 BGDAT3	AGDAT6(1) AGDAT6(2) AGDAT6(3)	AGDAT6(4) AGDAT6(5)
ALLOCR	generation	None.	BGDAT4 AGDAT4(1)	AGDAT4(2) AGDAT4(3)	AGDAT4(4) AGDAT4(5)

Figure 2.3. Macro Function (Part 1 of 3)

Macro	Type	Code Generated	Critical Globals Set
IOTAB	generation	Supervisor table expansions, channel buckets, PIBs, PUB, SVC table, channel queue table, JIB, TEB, PUBOWNER, POWER, LUB, NICL, FICL, ASCII, IT, OC, AB, and other tables.	BG99 AG1 BGDAT4 AG9 AG11 AG13 AG53 AG54 AG55 AGDAT5(1) AGDAT5(2) AGDAT5(3) AGDAT5(4) AGDAT5(5) AGDAT6(1) AGDAT6(3) AGDAT6(4) AGPGR(1) AGDAT6(5) AGPGR(2) AGPGR(3) AGPGR(4) AGPGR(5)
CRTGEN	inner	Generates CRT constants.	None.
IOINTER	inner	I/O interrupt routines. Error recovery exits.	BG1ST
SGDSK	inner	Disk error recovery routine.	None.
SMICR	inner	External interrupts for MICR-type devices. Timer Interrupt Handler. Program checks in stacker select routine. Error recovery for test I/O and start I/O.	None.
CRTSAV	inner	Generates CRT save areas.	None.
DVCGEN	generation	Overlays for PUB table entries.	BG28 BG56 BG126 AG1 AG2 AG3 AG8
ASSGN	generation	Overlays for LUB table entries.	BG28 BG57 BG58 BG59
DPD	generation	Overlays for DPD table entries.	None.
SEND	generation	None.	None.
IHBRDWRA	inner	Used for VTAM	None.
SGEND	inner	Generates patch area, user save area, label area, segment table, page table, page frame table, page frame table extension, copy buffer area, and all equates. Defines end of supervisor nucleus, beginning of IDRA, PD area, transient areas, and start of problem program area.	AG7
SGDFCH	inner	Fetch subroutine.	None.
SGCCWT	inner	CCW Translation routines.	None.
SGPMGR	inner	Page Management routines.	None.
SGAM	inner	Generates VSAM/GETVIS support.	None.
SGPOPT	inner	Generates PAGEIN support.	None.
MCRAS	inner	Generates RAS monitor and RTA. Resident Machine Check Handler, Resident Channel Check Handler, and DASD Channel Check Handler.	None.

Figure 2.3. Macro Function (Part 2 of 3)

Macro	Type	Code Generated	Critical Globals Set
SGSCVRT	inner	Sector convert routine for RPS	None.
MAPPUB2	inner	Maps layout of PUB2 entry.	None.
IJLQDSCT	inner	Generates DSECT for QTAM.	None.
BDYBOX	inner	Generates DSECT of boundary box.	None.
ISTAVT	inner	Generates VTAM Adress Vector Table.	None.

Figure 2.3. Macro Functions (Part 3 of 3)

Global	On Setting	Purpose
BG1QT	TP=QTAM	Determines QTAM support.
BG1ST	TP=VTAM	Determines VTAM support.
BG3	CHANSW=TSWTCH or RWTAU	Determines whether channel switching is supported (2816).
BG4	TEB=n	Determines if 2495 Tape Cartridge Reader error statistics are to be accumulated and logged.
BG6	OC=YES OLTEP=YES	Determines if the asynchronous user interrupt key routine is supported.
BG7	IT=YES	Determines whether the interval timer option is supported.
BG8	PC=YES	Determines if the user program check routine is supported.
BG9	CHANSW=RWTAU	Determines whether channel switching is supported (2404, 2804).
BG11	BMPX=YES	Determines whether burst mode devices will be supported on the multiplexer channel.
BG12		Determines tape support required.
BG13	AP=YES	Determines if asynchronous processing is supported. If YES, force on BG14, BG20.
BG14	WAITM=YES AP=YES	Determines if the wait multiple function is supported. Force on if AP is specified.
BG15	AB=YES	Determines if user abnormal termination routine is supported.
BG16	TRKHLD=n	Determines if the track hold function is supported. BG20 must be on (if it is not, TRKHLD defaults to NO).
BG18	CBF=n	Determines if console buffering option is supported.
BG20	MPS=YES or BJF AP=YES NPARTS=2,3,4, or 5	Determines whether multiprogramming support is required.

Figure 2.4. Global Settings (Part 1 of 6)

Global	On Setting	Purpose
BG21	TP=BTAM, VTAM, or QTAM	Determines whether teleprocessing support is required.
BG23	FP=YES	Determines if the floating point feature is used.
BG24	MPS=BJF or YES NPARTS=2,3,4, or 5 AP=YES	Determines if batched jobs will be run in foreground partitions.
BG26	PCIL=YES	Determines if PCIL is supported.
BG27	ERRLOG=RDE	Determines if RDE Support is required.
BG28		Sequence check for DVCGEN and ASSGN macros.
BG31	RPS=YES	Determines if Rotational Position Sensing is supported.
BG32	DASDFP=n,n	Determines whether the DASD file protect feature is supported.
BG33	SYSFIL=YES	Determines if system logical I/O units are supported for a disk device or diskette.
BG34	DATE=MDY	Determines the type of date configuration to be supported.

Figure 2.4. Global Settings (Part 1 continued)

Global	On Setting	Purpose
BG35	MICR=1419, 1419D	Determines if any MICR type device is supported.
BG36	MICR=1419D	Determines if a magnetic ink or optical character reader with dual address adapter is supported.
BG38	OLTEP=YES	Indicates whether OLTEP is supported; forces BG6 and BGRELDR.
BG39	MODEL=135, 145, 155-II, 158 <u>or</u> MODEL=115, 125 <u>and</u> MCH=YES or RMS=YES	Indicates support for MCAR and CCH functions of RMS.
BG40	RETAIN=YES	Indicates RETAIN is supported. Forces on BG38.
BG41	DECK=YES	Job Control options.
BG42	LIST=YES	Job Control options.
BG43	LISTX=YES	Job Control options.
BG44	SYM=YES	Job Control options.
BG45	XREF=YES	Job Control options.
BG46	ERRS=YES	Job Control options.
BG47	CHARSET=48C	Job Control options.
BG48	DUMP=YES	Job Control options.
BG49	LOG=YES	Job Control options.
BG50	EU=YES	Determines if emulator interface is generated.
BG51	EU=RELOC	This global is set by the EU=RELOC parameter, which is still accepted, but ignored. Forces on BG50.
BG52	IDRA=YES	Indicates whether IDRA is supported.
BG56	DVCTYP=2560 or 5425	Indicates that a MFCM or MFCU is attached to the system.
BG57	DVCTYP=2560 or 5425 <u>and</u> SYSnnn=SYSRDR,H2	SYSRDR is assigned to hopper 2 of a MFCM or MFCU.
BG58	DVCTYP=2560 or 5425 <u>and</u> SYSnnn=SYSIPT,H2	SYSIPT is assigned to hopper 2 of a MFCM or MFCU.
BG59	DVCTYP=2560 or 5425 <u>and</u> SYSnnn=SYSPCH,H2	SYSPCH is assigned to hopper 2 of a MFCM or MFCU.
BG60	TEBV=IR	Determines if individual recording for the Tape Error Statistics by Volume is supported. If BG60 is not on, the combined recording (CR) is in effect.
BG62	EVA=(rth,wth)	Determines if Error Volume Analysis is supported.

Figure 2.4. Global Settings (Part 2 of 6)

Global	On Setting	Purpose
BG71	JA=YES or JA=(n^1,n^2,n^3,n^4,n^5)	Indicates whether JAI is supported.
BG72	JA=(n_1,n_2,n_3,n^4,n5)	Indicates whether count of SIOs is made.
BG73	JALIOCS=(n_1,n_2)	Indicates whether JAI label processing is supported.
BG81	SPARM=YES	Determines if SYSPARM is supported.
BG86	ALIGN=YES	Job Control option.
BG87	EDECK=YES	Job Control option.
BG92	CHAN=YES	Indicates RMSR support for tapes, TP devices, and channel-attached devices for Model 115 or 125.
BG93	RMS=YES	Indicates full RMS support (MCAR/CCH and RMSR) for Model 115 or 125.
BG94	MCH=YES	Indicates MCAR/CCH support for Model 115 or 125.
BG99	ALLOC F1=nK, F2=nK, F3=nK, F4=nK	Special sequence check for ALLOC macro within supervisor.
BG100	ASCII=YES	Determines if ASCII translation tables are generated.
BG126	CHANSW=YES	Channel switching. Device can be attached to more than one selector channel.
BG127	---	Sequence check for DVCGEN and ASSGN macros.
BGACNCL	ACANCEL=YES	Job Control option.
BGBLKMP	BLKMPX=YES RPS=YES	Support for block multiplexer channels to run in block multiplex mode.
BGCEAPP	POWER=YES VSAM=YES	Indicates that the channel end appendage routine is to be included.
BGDAT0	PHO=YES	Page-fault appendage support.
BGDAT1	PFIX=YES or VSAM=YES	PFIX/PFREE support.
BGDAT2	ECPREAL=YES	Support for REALAD/VIRTAD.
BGDAT3	Total ALLOC specification plus 64K exceeds VSIZE specification	The sum of the virtual storage allocated to foreground partitions plus 64K (background partition) exceeds the virtual storage specified by VSIZE. ALLOC specification is invalid.
BGDAT4	ALLOCR	Sequence check for ALLOCR.
BGETVIS	GETVIS=YES or VSAM=YES	Determines if the GETVIS/FREEVIS functions are to be supported.
BGMTI	IT=YES and AP=YES, or NPARTS=2, 3, 4, or 5	Indicates timer support for all tasks in multiple partitions.

Figure 2.4. Global Settings (Part 3 of 6)

Global	On Setting	Purpose
BGPOW	POWER=YES	Generates the POWER/VS hooks in the supervisor.
BGRELDR	RELLDR=YES or VSAM=YES or OLTEP=YES or GETVIS=YES	Determines if Relocating Loader Support is required.
BGTOD	TOD=YES or VSAM=YES	Indicates that Time-of-Day is supported.
BGVSAM	VSAM=YES	Determines if VSAM support is required. Forces on BGETVIS, BGCEAPP, and BGRELDR.
FP2321	DASDFP=2321	Indicates whether the DASD file protect feature is supported for 2321.
MG01	DISK=3330	Generates code to support 3330 disk files.
MG02	DISK=3340	Generates code to support 3340 disk files.
AG1QT	TP=QTAM$_n$	Number of problem programs in QTAM system at one time.
AG1	IODEV=n	Number of entries for PUB Table.
AG2	Can be set by TEB, TEBV, EVA	Number of volume error blocks.
AG2A	TEB=n	Indicates number of Tape Error Blocks generated in FOPT macro.
AG3	CHUN=X'cuu'	Specifies number of channel (for example, X'cuu')
AG7	SKSEP=YES or n	Determines that the seek separation option is desired.
AG8	SKSEP=YES or n	Sets count of direct-access storage devices generated in DVCGEN.
AG9	CHANQ=n	Length of channel queue.
AG10	PD=YES or n	Indicates that an area is reserved for PDAID routines.
AG11	MODEL=135, 145, 155-II, 158 <u>or</u> MODEL=115 or 125 <u>and</u> RMS=YES or CHAN=YES	Indicates RMSR support.
AG12	MODEL=nnn	Indicates model of CPU.
AG13	PCIL=YES	Number of system class LUBs.
AG14	ERRQ=n	Indicates the number of entries in the error queue table to be generated by the ICTAB macro.
AG15	---	Checks macro sequence of supervisor macros.
AG16	DASDFP= n,n, {2311, 2314, 3330, 3340, 2321}	First channel with DASD devices for file protect.

Figure 2.4. Global Settings (Part 4 of 6)

Global	On Setting	Purpose
AG17	DASDFP= n,n, {2311, 2314, 3330, 3340, 2321}	Last channel with DASD devices for file protect.
AG18	SYSFIL= YES , n_1,n_2	n_1 Parameter of SYSFIL.
AG19	SYSFIL= YES , n_1,n_2	n_2 Parameter of SYSFIL.
AG27	TRKHLD=n	Indicates number of tracks that can be held.
AG28	CBF=n	Indicates number of console buffers generated in LUBGEN.
AG31	EVA=(rth,wth)	Indicates the Read Error Threshold specified in EVA.
AG32	EVA=(rth,wth)	Indicates the Write Error Threshold specified in EVA.
AG41	LINES=n	SYSLST line count.
AG53	---	Number of LUBs.
AG54	MRSLCH=YES	Indicates MICR type device is on a selector channel.
AG55		Set total number of devices for RMS.
AG56	MODEL=115 or 125 and DOC=125D	Indicates that CRT is supported.
AGDAT1	RSIZE=nK	Defines end of real address area.
AGDAT2	VSIZE=nK	Defines end of virtual address area.
AGDAT3	BUFSIZE=n	Number of copy buffers for channel program translation.
AGDAT4(1) AGDAT4(2) AGDAT4(3) AGDAT4(4) AGDAT4(5)	BGR=nK F1R=nK,F2R=nK, F3R=nK,F4R=nK	AGDAT4(1) specifies the number of blocks allocated to the real background partition. AGDAT4(2) through AGDAT4(5) specify the number of blocks allocated to the real foreground partitions. The partitions to which these globals refer depend on the number of partitions supported as follows: Global — Number of Partitions: 2, 3, 4, 5 AGDAT4(2): F1, F2, F3, F4 AGDAT4(3): —, F1, F2, F3 AGDAT4(4): —, —, F1, F2 AGDAT4(5): —, —, —, F1
AGDAT5(1) AGDAT5(2) AGDAT5(3) AGDAT5(4) AGDAT5(5)	 F1=nK,F2=nK, F3=nK,F4=nK	Defines the origin of the virtual background partition. Defines the origin of virtual foreground partitions. The partitions to which these globals refer depend on the number of partitions in the same way as for AGDAT4(2) through AGDAT4(5) above.

Figure 2.4. Global Settings (Part 5 of 6)

Global	On Setting	Purpose
AGDAT6(1)		Defines the end address of the virtual background partition.
AGDAT6(2)	F1=nK,F2=nK,	Defines the end address of the virtual foreground partitions.
AGDAT6(3) AGDAT6(4) AGDAT6(5)	F3=nK,F4=nK	The partitions to which these globals refer depend on the number of partitions in the same way as for AGDAT4(2) through AGDAT4(5) above.
AGDAT7	PAGEIN=n	Specifies the maximum number of requests that can be queued.
AGJA(1) AGJA(2) AGJA(3) AGJA(4) AGJA(5)	JA=(n_1,n_2,n_3,n4,n5)	AGJA(1) indicates the number of devices for the background partition for which SIO counts are to be made. AGJA(2) through AGJA(5) indicate the same for the foreground partitions. The associated partitions depend on the number of partitions in the same way as for AGDAT4(2) through AGDAT4(5) above.
AGJAL(1) AGJAL(2)	JALIOCS(n_1,n_2)	Indicate whether user save area and label processing are supported.
AGPGR(1)	BGPGR=n	Number of BG programmer LUBs.
AGPGR(2)	F1PGR=n	Number of F1 programmer LUBs.
AGPGR(3)	F2PGR=n	Number of F2 programmer LUBs.
AGPGR(4)	F3PGR=n	Number of F3 programmer LUBs.
AGPGR(5)	F4PGR=n	Number of F4 programmer LUBs.
AGZN	ZONE= {EAST / WEST},hh,mm	Difference in hours and minutes between GMT and local time.
IJBSGEN		Set by SUPVR macro to indicate for SEND macro that a supervisor generation is in process.
ITKEY	IT=BG,F1,F2,F3, F4, or YES	Sets key of timer-owning partition.
MAX		Indicates the maximum number of partitions that can be supported by the DOS/VS supervisor, and is always set to 5.
NPART	NPARTS=n AP=YES	Number of partitions supported by this DOS/VS supervisor. If none of these parameters is included, the global is set to 1.
NSLD(1)	SLD=n	Number of entries in the Second Level Directory.
NSLD(2)	PSLD=n	Number of entries in the Private Second Level Directory.
NSVAPGS	SVA= nK,mK	Number of pages in the SVA.

Figure 2.4. Global Settings (Part 6 of 6)

SYSCOM

Displacement (Hex)	Displacement (Dec)	Contents	Length
0	0	Address of Error Block	xxxx
4	4	Address of Attention Exit	xxxx
8	8	Address of Operator Option Cancel Exit	xx
0A	10	Address of Operator Request Cancel Exit	xx
0C	12	Address of SYSRES PUB	xxxx
10	16	Address of Fetch Routine	xxxx
14	20	Address of I/O Interr. Routine	xxxx
18	24	Address of Ext. Interr. Routine	xxxx
1C	28	Address of Logical Transient Area	xxxx
20	32	Address of 1st byte of Problem Program Area	xxxx
24	36	Free List Pointer	x
25	37	Address of Channel Queue	xxx
28	40	Number of Channel Queue Entries	xx
2A	42	Length of One Error Queue Entry	xx
2C	44	Number of Partitions	xx
2E	46	Not used	xx
30	48	Address of Channel Buckets	xxxx
34	52	Address of CRT Table	xxxx
38	56	Address of SAB Table	xxxx
3C	60	Address of Channel Control Table	xxxx
40	64	Flags and Switches (See expansion) *	xxxx
44	68	System Task Selection Control Field *	xx
46	70	Address of Task Selection	xx
48	72	Address of PD Area	xxxx
4C	76	TH Free List Pointer	x
4D	77	Address of TH Table	xxx
50	80	Address of Timer Request Table	xxxx
54	84	Address of AB Table	xxxx
58	88	Key of Task owning LTA (LIK)	xx
5A	90	Key of Task running (TIK)	xx
5C	92	Address of POWER/VS Partition	xxxx
60	96	Address of VTAM Address Vector Table	xxxx
64	100	Address of RF Table	xxxx
68	104	Address of EU ECB Table	xxxx
6C	108	Address of OLTEP bucket	xxxx
70	112	Address of RAS Linkage Area	xxxx
74	116	Address of ASCII Translate Table	xxxx
78	120	Address of PUB Ownership Table	xxxx
7C	124	Address of Job Accounting Common Table	xxxx
80	128	Base Address of Page Management Routine	xxxx
84	132	Base Address of Channel Program Translation Routine	xxxx
88	136	Address of SDAID Comm. Area	xxxx
8C	140	Address of Line Mode Table	xxxx
90	144	Address of VSAM Communication Area	xxxx
94	148	Address of PTA	xxxx
98	152	Address of first System Task Block	xxxx
9C	156	Address of Task Block of Active System Task	xxxx
A0	160	1 byte for Alignment	x
A1	161	Pointer to RAS Task Block	x
A2	162	Pointer to PMGR Task Block	x
A3	163	Pointer to SUPVR Task Block	x
A4	164	Pointer to CRT Task Block	x
A5	165	Pointer to ERP Task Block	x
A6	166	Pointer to PAGEIN Task Block	x
A7	167	Reserved (9 X'00')	xxxxxxxxx
B0	176	Not used	xxxx
B4	180	Address of MVCFLD	xxxx
B8	184	TRTMSK Pointer	xxxx
BC	188	Not used	xx
BE	190	Not used	xx
C0	192	Repositioning Information for MFCM ERP	xxxxxxxxxxx
CB	203	Number of Error Queue Entries	x
CC	204	Length of PUB Table in bytes	xx
CE	206	Number of Active Partitions	xx
D0	208	Address of Segment Table	xxxx
D4	212	Address of Page Frame Table	xxxx
D8	216	Address of Page Frame Table Extension	xxxx
DC	220	Address of Boundary Box	xxxx
E0	224	Address of DPD Table	xxxx
E4	228	Reserved	xxxx
E8	232	Address of VIRTAD Routine	xxxx
EC	236	Address of End of Real Storage	xxxx
F0	240	Address of Fetch Table	xxxx
F4	244	SVA Flag (see expansion)	x
F5	245	Address of SVA	xxx
F8	248	Address of System GETVIS AREA	xxxx
FC	252	Address of RPS Local Directory List	xxxx
100	256	Address of RPS Sector Calculation Routine	xxxx

*See next page for further explanation

Figure 2.5. System Communication Region (SYSCOM) (Part 1 of 2)

Expansion of SYSCOM Flag Bytes

Byte Dec	Byte Hex	Description
64	40	Reserved for RMS in Model 115 and 125
		X'80' RMSR for channel attached devices, tapes and TP devices
		X'40' Full RMS support (MCAR/CCH and RMSR)
		X'20' MCAR/CCH support
65	41	X'80' Initial selection of ERP
		X'40' Reserved
		X'20' Timer interrupt pending
		X'10' MICR Stacker-select active
		X'08' Invalid address during fetch
		X'04' SIO routine entered after interrupt
		X'02' Reserved
		X'01' IPL in progress
66	42	X'80' Initial RAS request
		X'40' RAS Wait request outstanding
		X'20' RAS IPL in progress
		X'10' Reserved
		X'08' POWER/VS supported
		X'04' POWER/VS initialized
		X'02' GETREAL for SDAID or PDAID in progress
		X'01' Fetch for system task in progress (used by PDAID)
67	43	Reserved
244	F4	SVA Flag
		X'80' Do not test for warm start copy of SVA
		X'40' SDL active
		X'20' No 'Set SVA' or 'Set SDL' allowed
		X'10' Build of SDL in progress
		X'08' SDL overflow
		X'04' Reserved
		X'02' Reserved
		X'01' Reserved
252	FC	X'00000000' RPS not initialized
		X'00xxxxxx' Pointer to RPS LDL in SVA
256	100	X'00000000' No RPS support
		X'00xxxxxx' Pointer to Sector Calculation Routine

Layout of System Task Selection Control Field

Byte Dec	Byte Hex	Description
68	44	Always zero
69	45	SELECT byte:
		X'00' No system task active
		X'01' RAS active
		X'02' PMGR active
		X'03' SUPVR active
		X'04' CRT active
		X'05' ERP active
		X'06' PAGEIN active

Note: The address of SYSCOM can be found at fixed location X'80' - X'83'.

Figure 2.5. System Communications Region (SYSCOM) (Part 2 of 2)

nnCOMREG

Displacement (Hex / Dec)

Hex	0	8	0A	0C	17	18	20	24	28	2C
Dec	0	8	10	12	23	24	32	36	40	44
Field	Date	Address of PPBEG	Address of EOSSP	Problem Program Use	UPSI Byte	Job Name	Highest Storage Address of the Partition	End Address of Last Phase Fetched or loaded	Address of uppermost Byte of Phase with highest Ending Address	Label Area Length
Length	xxxxxxxx	xx	xx	xxxxxxxxxxx	x	xxxxxxxx	xxxx	xxxx	xxxx	xx

Hex	2E	30	34	35	36	37	38	39	3A	3B	3C	3E
Dec	46	48	52	53	54	55	56	57	58	59	60	62
Field	PIK	End of Virtual Storage Address	Machine Configur. Byte	System Configur. Byte	Standard Language Translator I/O Options	Dump, Log, RELLDR and ASCII Options	Job Control Byte	Linkage Control Byte	Language Translator Control Byte	Job Duration Indicator Byte	Disk Address of Label Cylinder	Address of FOCL
Length	xx	xxxx	x	x	x	x	x	x	x	x	xx	xx

(Offsets 36 through 3B: Job Control Switches)

Hex	40	42	44	46	48	4A	4C	4E	4F	58	5A	5C
Dec	64	66	68	70	72	74	76	78	79	88	90	92
Field	Address of PUBTAB	Address of FAVP	Address of JIBTAB	Address of TEBTAB	Address of FICL	Address of NICL	Address of LUBTAB	Line Count for SYSLST	System Date	LIOCS Comm. Bytes	Address of PIB Table	ID Number of last Checkpoint or DASDFP Indicator
Length	xx	xx	xx	xx	xx	xx	xx	x	xxxxxxxxx	xx	xx	xx

Hex	5E	60	62	64	66	68	6A	6C	6E
Dec	94	96	98	100	102	104	106	108	110
Field	Job Zone in Minutes	Address of Disk Information Block (DIB)	Reserved	Address of PC Option Table less 8 bytes	Address of IT Option Table	Address of OC Option Table less 8 bytes	Key of Program with Timer Support	Reserved	Logical Transient Key (LTK)
Length	xx	xx	xx	xx	xx	xx	xx	xx	xx

Hex	70	74	78	7C	7E	80	84	86	87
Dec	112	116	120	124	126	128	132	134	135
Field	Address of SYSPARM	Address of J.A. Partition Table	Address of TOD clock Common Area	Address of PIB Table Extension	Address of MICR DTF Table (PDTABB)	Address of QTAM Vector Table	Address of BG Comm. Region	Option Indicator	System Configuration Byte 2 and RMSR Open Flag Byte
Length	xxxx	xxxx	xxxx	xx	xx	xxxx	xx	x	x

Hex	88	8C	8D	8E	8F	97	98	9F
Dec	136	140	141	142	143	151	152	159
Field	Pointer to Option Table in SYSCOM Reserved for compatibility reasons	Standard Job Control Options	Temporary Job Control Options	Disk Configuration	Catalog Procedure Name	Switch for Catalog Procedure	JCL Statement Name	81 bytes SYSIN Indicator
Length	xxxx	x	x	x	xxxxxxxx	x	xxxxxxx	x

Hex	A0	A4	A5
Dec	160	164	165
Field	Address of POWER/VS Partition Control Block	POWER/VS Flag Byte 1	POWER/VS Flag Byte 2
Length	xxxx	x	x

Note: A communication region exists from each partition supported by the system. The address of the communication region of the active partition is in fixed location X'14' - X'17'.

Figure 2.6. Partition Communications Region (Part 1 of 6)

Key to Communication Region Displacement

Key	Description of Use
0	MM/DD/YY or DD/MM/YY either set permanently by the job control date statement, or updated every time a GETIME macro is issued when Time-of-Day support is provided. Format controlled by BGCOMREG+53. (System Configuration Byte, date convention bit 0.)
8	Address of the problem program area (Note: Maintained for compatibility. Correct value is found in SYSCOM table displacement X'20'.).
10	Address of the beginning of the problem program area (see above). Y(EOSSP) equals Y(PPBEG)
12	User area. If seek separation option is specified, bytes 12 and 13 are used at IPL time for the address of the seek address block.
23	User program switch indicator.
24	Job name set by the job control program from information found in the job statement
32	Address of the uppermost byte available to the problem program, that is, either the address of the uppermost byte of the partition as determined during processing of the ALLOC or ALLOCR macro or statement, or the end address of the area specified by the SIZE parameter in the EXEC statement.
36	Address of the uppermost byte of the last phase of the problem program fetched or loaded. Not filled in when phase is in SVA.
40	Highest ending main-storage address of all the phases having the same first four characters as operand on the EXEC statement. For the phase $LNKEDT, this field is not filled in. The address value may be incorrect if the program loads any of this phase above or below its link-edited origin address. If the EXEC statement has no operand, job control places in this location the highest ending address of all programs just link-edited.
44	Length of the problem program label area.
46	Partition Identification Key (PIK). The low-order byte identifies the active partition. Only significant for BG communication region.
48	End address of virtual storage.
52	Machine Configuration Byte (values set at supervisor generation time). Bit 0: Always set to indicate standard storage protect 1: 1 = Decimal feature (always set) 2: 1 = Floating-point feature 0 = No floating-point feature 3: 1 = Physical transient overlap option 0 = No physical transient overlap option 4: Always set to indicate standard timer feature 5: 1 = Channel switching device 0 = No channel switching device 6: 1 = Burst mode on multiplex channel support 0 = No burst mode on multiplex channel support 7: Indicates MCH/CCH in system
53	System Configuration byte. Bit 0: 1 = DDMMYY (Date convention bit set at generation time by STDJC) 0 = MMDDYY (Date convention bit set at generation time by STDJC) 1: 1 = Two or more partitions 0 = One partition only

Figure 2.6. Partition Communications Region (Part 2 of 6)

Key	Description of Use
	53 System Configuration byte (Cont'd.) 2: 1 = DASD file-protect supported 0 = No file-protect support for DASD 3: 1 = DASD SYSIN - SYSOUT 0 = No DASD SYSIN - SYSOUT 4: 1 = Teleprocessing 0 = No teleprocessing 5: 1 = Two or more partitions 0 = One partition only 6: 1 = Asynchronous processing 0 = No asynchronous processing 7: 1 = Track Hold 0 = No Track Hold
54	This byte contains the standard language translator I/O options (set by STDJC macro). Bit 0: DECK option 1 = yes, output object modules on SYSPCH 1: LIST option 1 = yes, output source module listings and diagnostics on SYSLST 2: LISTX option 1 = yes, output hexadecimal object module listings on SYSLST (compilers only) 3: SYM option 1 = yes, output symbol tables on SYSLST/SYSPCH 4: XREF option 1 = yes, output symbolic cross-reference list on SYSLST 5: ERRS option 1 = yes, output diagnostics on SYSLST (compilers only) 6: CHARSET option 1 = 48, input on SYSIPT is 48 or 60 character set 7: Reserved
55	This byte contains the standard supervisor options for abnormal EOJ, Relocating Loader and Control statement display and the indicator for the presence of the ASCII-EBCDIC and EBCDIC-ASCII translation tables. Bit 0: Always on 1: DUMP option 1 = yes, dump registers and storage on SYSLST 2: 1 = partition in wait state, because a volume is to be mounted 3: LOG option 1 = yes, list all control statements on SYSLST 4: 1 = dummy device search in progress; do not enter ERP 5: Not used 6: Relocating Load option 1 = yes, Relocating Loader supported 7: ASCII option 1 = yes, ASCII supported
56	Job Control byte. Bit 0: 1 = Job Accounting Interface (JA) not supported 0 = Job Accounting Interface (JA) is supported 1: 1 = Return to caller on LIOCS disk open failure 0 = Do not return to caller on LIOCS disk open failure 2: 1 = Job Control input from SYSRDR 0 = Job Control input from SYSLOG 3: 1 = Job Control output on SYSLOG 0 = Job Control output not on SYSLOG 4: 1 = Cancel job 0 = Do not cancel job 5: 1 = Pause at End-of-Job step 0 = No pause at End-of-Job step 6: 1 = SYSLOG is not a console printer-keyboard or DOC 0 = SYSLOG is a console printer-keyboard or DOC 7: 1 = SYSLOG is assigned to the same device as SYSLST 0 = SYSLOG is not assigned to the same device as SYSLST

Figure 2.6. Partition Communications Region (Part 3 of 6)

Key	Description of Use
57	Linkage control byte. Bit 0: 1 = SYSLNK open for output 0 = SYSLNK not open for output 1: 1 = Update of Second Level Directory and RAS Load list in progress (interface between $MAINDIR and supervisor) 2: 1 = Allow EXEC 0 = Suppress EXEC 3: 1 = Catalog Linkage Editor output 0 = Do not catalog Linkage Editor output 4: 1 = Supervisor has been updated 0 = Supervisor has not been updated 5: Reserved 6: 1 = Update of System Core Image Library in progress (interface between $MAINDIR and Supervisor) 7: 1 = Check automatic condense limits at End-of-Job (interface between Librarian and Job Control)
58	Language processor control byte. This is a set of switches used to specify nonstandard language translator options. The switches within the byte are controlled by job control OPTION statements and when set to 1, override standard options. The format of this byte is identical to the standard option byte (displacement 54) with one exception: Bit 7 in this byte is used to indicate to LIOCS that the rewind and unload option has been specified.

59	Job duration indicator byte.
	
	Bit 0: 1 = Job in progress
	0 = No job in progress
	1: 1 = Dump on an abnormal End-of-Job condition
	0 = No dump on abnormal EOJ
	2: 1 = Pause at EOJ step (Set by attention routine for Job Control)
	0 = No pause at EOJ (Set by attention routine for Job Control)
	3: 1 = Job control output on SYSLST
	0 = Output not on SYSLST
	4: 1 = Job is being run out of sequence with a temporary assignment for SYSRDR
	0 = Conditions for 1-setting not met
	5: 1 = PCIL is being condensed
	0 = PCIL is not being condensed
	6: 1 = //DATE statement processed for current job
	0 = No //DATE statement processed for current job
	7: 1 = Batch command just issued
	0 = Condition for 1-setting did not occur

60	Binary disk address of the volume label area (label cylinder).
62	As illustrated (for detailed figures see Index).
76	As illustrated (for detailed figures see Index).
78	Set to the value nn specified in the LINES=nn parameter of the STDJC macro.
79	The format of the system date contained within this field is determined by the IPL program from information supplied in the date convention bit (displacement 53). Bytes 85-87 contain the day count.
88	Bytes reserved for use by LIOCS. Transient dump programs insert a key to indicate to the LIOCS End-of-Volume routine, $$BCMT07, that it was called by a B-transient.
90	Address of the first part of the Program Information Block (PIB) table.

Figure 2.6. Partition Communications Region (Part 4 of 6)

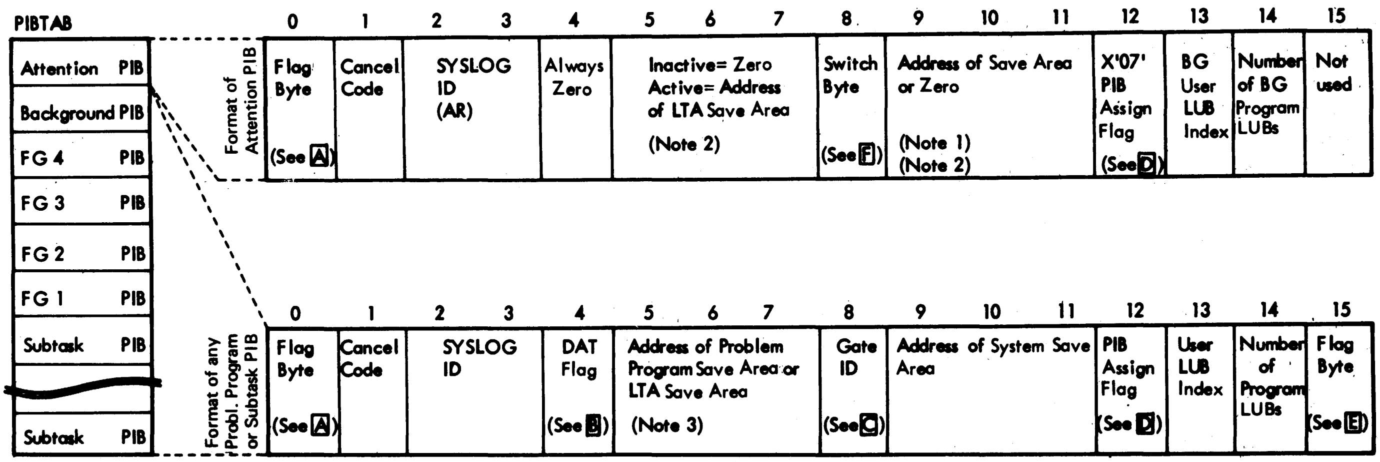
Key	Description of Use
92	ID number of the last checkpoint. Byte 92 is also the temporary indicator of file protected DASD. Bits 0-6 correspond to channels 0-6. A bit ON means DASDFP for that channel. Bit 7 indicates 2321 DASDFP support. Byte 93 is used at IPL time by PIOCS.
93	Bit 0: 1 = 3330 file protection 1: 1 = 3340 file protection
94	Job zone for Time-of-Day. If ZONE=EAST, value is positive; if ZONE=WEST, value is negative.
96	Address of disk I/O position data. This is the starting address of the Disk Information Block (DIB) table for the partition.
98	Reserved.
100	PC option table (zero if not specified).
102	IT option table (zero if not specified).
104	OC option table (zero if not specified).
106	X'0010' if interval timer support. X'0000' if no interval timer support.
108	Reserved.
110	Logical Transient Key (LTK) contains the same value as the PIK (PID) (Displacement 46) when the logical transient is requested. When the transient area is not in use, LTK is equal to zero. The SVC 2 routine sets the LTK. The SVC 11 routine resets the LTK. (Only significant in BG communication regions.)
112	Address of SYSPARM field.
116	Address of Job Accounting partition table.
120	Address of the Time-of-Day Clock common area.
124	Address of second part of Program Information Block (PIB) table.
126	Address of PDTABB, table of DTF addresses for MICR support.
128	Address of QTAM vector table (IJLQTTAD).
132	Address of background communications region.
134	Option Indicator byte. Bit 0: Reserved 1: 1 = EU interface active 0 = EU interface inactive 2: 1 = Teleprocessing request 0 = No teleprocessing request 3: 1 = Supervisor support for tape 0 = Supervisor does not support tape 4: Reserved 5: 1 = RETAIN support generated 0 = RETAIN support not generated 6: 1 = Linkage to Channel End Appendage Routine allowed 0 = Linkage to Channel End Appendage Routine not allowed 7: 1 = GETVIS function has been initiated 0 = GETVIS function has not been initiated

Figure 2.6. Partition Communications Region (Part 5 of 6)

Key	Description of Use
135	System Configuration byte 2 and RMSR Open Flag byte. Bit 0: 1 = PCIL supported 1: 1 = TOD-clock supported 2: 1 = PFIX macro supported 3: 1 = Fetch $$BOPEN by $JOBCTLJ 4: 1 = Fetch $$BOPEN by $JOBCTLD 5: 1 = Fetch $$BOPEN by $JOBCTLJ for WTM 6: 1 = QTAM supported 7: 1 = RPS supported
136	Pointer to Option table in SYSCOM Reserved for compatibility reasons.
140	Standard Job Control Option byte. Bit 0: 1 = EDECK Standard Option 1: 1 = ALIGN Standard Option 2-5: Not used 6: SUBLIB=DF Temporary Option 7: 1 = ACANCEL Standard
141	Temporary Job Control Option byte. Bit 0: 1 = EDECK Temporary Option 1: 1 = ALIGN Temporary Option 2-5: Not used 6: SUBLIB=DF Temporary Option 7: 1 = ACANCEL Temporary
142	Disk Configuration byte. Bit 0-4: Not used 5: 1 = 3340 supported 6: 1 = 3330 supported 7: Always 1; indicates 2311 and 2314/2319 supported
143	Cataloged Procedure Name.
151	Interface byte for Cataloged Procedures. Bit 0: 1 = Procedure being executed 1: 1 = Overwrite processing 2: 1 = Procedure with data 3: 1 = Overwrite request for Job Control 4: 1 = Insert request for Job Control 5: 1 = Procedure end 6: 1 = SYSLOG procedure 7: 1 = Overwrite request for Supervisor
152	JCL statement name for Cataloged Procedure.
159	SYSIN 81 bytes indicator. Bit 0: 1 = Permanent 81 bytes on SYSRDR 1: 1 = Permanent 81 bytes on SYSIPT 2: 1 = Temporary 81 bytes on SYSRDR 3: 1 = Temporary 81 bytes on SYSIPT 4-6: Not used 7: 1 = Allow /& for MAINT CATALS

Figure 2.6. Partition Communications Region (Part 6 of 6)

Key	Description of Use
160	Pointer to POWER/VS Partition Control Block.
164	POWER/VS Flag Byte 1. Bit 0: 1 = POWER/VS Accounting supported 1: 1 = Partition under control of POWER/VS 2: 1 = POWER/VS Partition 3-7: Reserved
165	POWER/VS Flag Byte 2. Bit 0-7: Reserved



Notes:

1. a. When LTA is inactive = LTA save area address.

   b. When LTA is active for Problem Programs, this address is exchanged with that in the Problem Program PIB.

2. When LTA is active for Logical Attention, bytes 9-11 are zero and bytes 5-7 contain the LTA save area address.

3. When the Logical Transient Area is active the save area address in the Problem Program PIB is exchanged with that in the Attention PIB.

The number of Problem Program PIBs generated depends on the number of partitions specified during system generation. Subtask PIBs are generated only if AP=YES has been specified during system generation. The number of subtask PIBs generated depends on the number of partitions, that is:

Number of Partitions	Number of Subtasks
2	13
3	12
4	11
5	10

Bytes 90-91 (X'5A' - X'5B') of the partition communication region(s) contain the address of the PIB table. Label PIBTAB identifies the first byte of the table.

[A] Flag Byte (First byte in PIB)

The following flags are always used:

X'71' = Program is waiting for SVC 58
X'73' = Program is waiting because system is seized
X'75' = Program is waiting for copy block
X'77' = Program is waiting for TFREE
X'79' = Program is waiting for channel queue entry
X'7B' = Program is waiting for CCW translation
X'7D' = Program is waiting for free console buffer table entry (used only when CBF=n)
X'80' = Program is not active
X'81' = Program is SVC 2-bound (waiting for the LTA to be released)
X'82' = Program is SVC 7-bound (waiting for an I/O interruption)
X'83' = Program is ready to run
X'85' = Program is SVC 5-bound (waiting for the PTA to be released)
X'86' = Initial selection of RAS (used only for RAS PIB Flag)
X'87' = Program is set to common bound condition

Figure 2.7. Program Information Block (PIB) Table (Part 1 of 2)

The following flags are used only if NPARTS>1. X'61' through X'69' are used by the load leveller to deactivate a partition. The partition to which a flag refers depends on NPARTS as follows:

		NPARTS= 2	3	4	5
X'61'	refers to	BG	BG	BG	BG
X'63'	refers to	F1	F2	F3	F4
X'65'	refers to	-	F1	F2	F3
X'67'	refers to	-	-	F1	F2
X'69'	refers to	-	-	-	F1

X'6A' = Program is SVC 35-bound
X'6B' = Program is SVC 35-bound
X'6D' = Program is waiting for next freed page frame
X'6F' = Program is IDRA-bound

The following flags are only used if AP=YES:

X'51' = Program is SVC 38-bound
X'53' = Program is SVC 41/42-bound

The following codes are only used if AP=YES and PFIX=YES. The codes are used by the PFIX routines to set a partition PFIX-bound. The partition to which a flag refers depends on NPARTS as follows:

		NPARTS= 2	3	4	5
X'47'	refers to	BG	BG	BG	BG
X'49'	refers to	F1	F2	F3	F4
X'4B'	refers to	-	F1	F2	F3
X'4D'	refers to	-	-	F1	F2
X'4F'	refers to	-	-	-	F1

The following codes are used only if AP=YES and VSAM=YES. The codes are used by the VSAM routines to set a partition PFIX-bound. The partition to which a flag refers depends on NPARTS as follows:

		NPARTS= 2	3	4	5
X'3D'	refers to	BG	BG	BG	BG
X'3F'	refers to	F1	F2	F3	F4
X'41'	refers to	-	F1	F2	F3
X'43'	refers to	-	-	F1	F2
X'45'	refers to	-	-	-	F1

The following flag is only used when CBF=n: X'7D' = Program is waiting for free console buffer table entry.

The following flag is only used when TP=QTAM: X'8B' = Task in QTAM wait.

B PIB DAT Flag

X'01' = Return to reentrant supervisor routine
X'02' = Return to gated supervisor routine
X'04' = Move CCB at dispatching time
X'08' = Service delayed external interrupt
X'10' = Deactivation of this task is being delayed
X'20' = Reserved
X'40' = Task has seized the system
X'80' = Program is running in virtual mode

C Gate Identifier

X'71' = Gating of SVC 58 required
X'53' = Gating of SVC 41/42 required

The flags are only used if the PIB DAT Flag is X'03', that is, the first two flags are on. (See B.)

D PIB Assign Flag

X'80' = SYSRES DASD file protect inhibited (allow write operation on SYSRES)
X'40' = Channel appendage exit allowed (BTAM)
X'20' = Cancel in progress (used in terminator function)
X'10' = Cancel control (set on a foreground cancel)
X'08' = Hold foreground assignments
X'07' = Attention PIB

E Program Program PIB Flag (Last byte in PIB)

Bit 0: 1 = Batched job in foreground (always on when tested)
1: 1 = Cancel in LTA and device not assigned
2: 1 = /& on SYSIN if DASD
3: 1 = Partition in stopped state
4: 1 = Fetch EOJ monitor
5: 1 = Task is canceled
6: 1 = Subtask(s) attached
7: 1 = in AB routine

F Attention PIB Switch Byte

Bit 0: Reserved
1: 1 = Fetch Physical Attention Transient $$ABERRZ
2: 1 = Delay cancellation (fetch $$ABERZ1)
3: 1 = Emergency cancel request
4: Reserved
5: 1 = Command available (DOC)
6: 1 = Fetch Logical Attention Routine ($$BATTNA)
7: 1 = External Interrupt request

Figure 2.7. Program Information Block (PIB) Table (Part 2 of 2)

Indications of Logical Transient Area Occupancy and Activity					
Status	BGCOMREG	Attention PIB	Problem PIB	Condition of LTA	Notes
SVCs issued	Contents of LTK + 1 (1 Byte)	Address in ARFLG + 1 (3 Bytes)	Address in PIBSAVE + 1 (3 Bytes)		
(1)	zero	Logical Transient Save Area LTASAVE)	Problem Program Save area	Free	Initial condition before issuing SVC 2
(2) SVC 2	Owner's Partition Identification Key	Problem Program Save Area	Logical Transient Save Area (LTASAVE)	Active	
(3) SVC 2 SVC 11	zero	Logical Transient Save Area (LTASAVE)	Problem Program Save Area	Free	Restored to (1)
(4) SVC 2 SVC 8	Owner's Partition Identification Key	Logical Transient Save Area (LTASAVE)	Problem Program Save Area	Occupied but Inactive	SVC 8 may be issued only from LTA. General register 14 contains address of entry point to the user routine
(5) SVC 2 SVC 8 SVC 9	Owner's Partition Identification Key	Problem Program Save Area	Logical Transient Save Area (LTASAVE)	Active	Restored to (2). SVC 9 may be issued only from Problem Program.
(6) SVC 2 SVC 8 SVC 9 SVC 11	zero	Logical Transient Save Area (LTASAVE)	Problem Program Save Area	Free	Restored to (1).

Figure 2.8. Indications of Logical Transient Area Occupancy and Activity

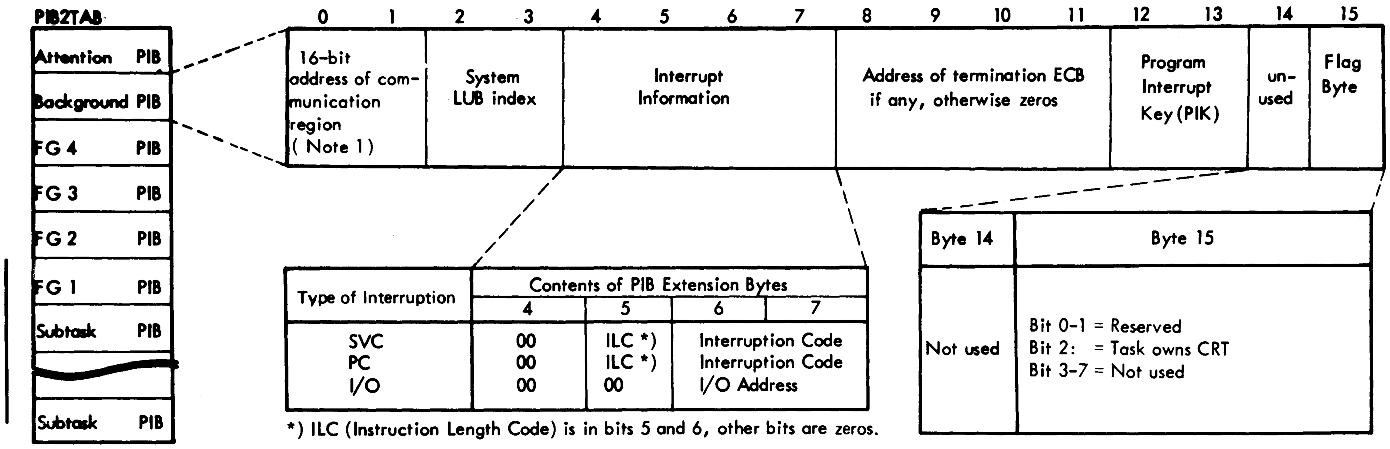


Figure 2.9. Program Information Block Table Extension (PIB2TAB)

Note: Always BG communication region in Attention- and Background PIB extension. Appropriate communication region in other PIB extensions when a multiprogram system has been generated. To place this address in a register the instruction ICM should be used.

For each PIB Table entry an entry exists in the PIB Table Extension.

Bytes 124-125 (X'7C' - X'7D') of the partition communication region(s) contain the address of the PIB Table extension. Label PIB2TAB identifies the first byte of the PIB Table extension.

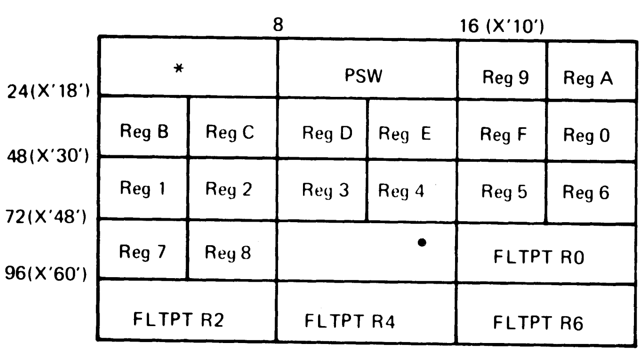


Figure 2.10. LTA and Problem Program Save Area

*

Problem Program Save Area:

Program Name

LTA Save Area: Not used

•

Problem Program Save Area:

First half-word: Label area length
Second half-word: Reserved
Third and fourth half-word: Job start time

LTA Save Area: Reserved

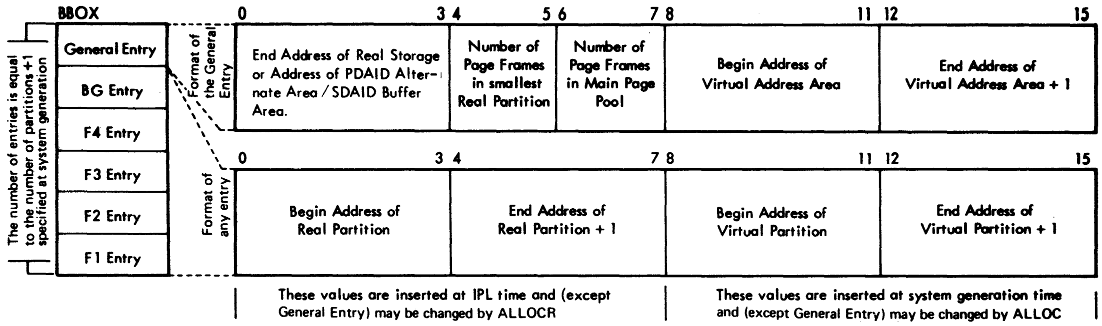


Figure 2.11. Boundary Box (BBOX)

Note: If a partition is not allocated, its begin and end address entries contain the begin address of the following partition.

Bytes 220-223 (X'DC' - X'DF') of the System Communication Region (SYSCOM) contain the address of the Boundary Box. Label BBOX identifies the first byte of the table.

The table of DTF addresses (PDTABB) contains six 8-byte entries; one for each line of the direct control feature on the system.

PDTABB

Byte	0	1	2	3	4	5	6	7
	AND Instruction				Ownership Flags	DTF Address for MICR		
0	NI	PDSTAT+1		X'FE'		Device on line 7		
8	NI	PDSTAT+1		X'FD'		Device on line 6		
16	NI	PDSTAT+1		X'FB'		Device on line 5		
24	NI	PDSTAT+1		X'F7'		Device on line 4		
32	NI	PDSTAT+1		X'EF'		Device on line 3		
40	NI	PDSTAT+1		X'DF'		Device on line 2		

Figure 2.12. Tables for MICR DTF Addresses and Pointers (Part 1 of 2)

Bytes:

0-3 Contain an 'AND' instruction that is executed in main line coding to turn off the external line status after its detection.

PDSTAT+1 will contain one or more of the following interrupt codes:

External Interrupt Code Bit	Interrupt Code (byte X'87')	External Interrupt Cause
15	nnnnnnn1	External signal 7
14	nnnnnn1n	External signal 6
13	nnnnn1nn	External signal 5
12	nnnn1nnn	External signal 4
11	nnn1nnnn	External signal 3
10	nn1nnnnn	External signal 2

n = other external interrupt conditions

4 Contains the flag of the partition containing the DTF
Background = 10
Foreground = 20-50, depending on the number of partitions

5-7 Contain the address of the DTF table

Table of pointers (PDTABA) to DTF addresses with the external interrupt line. The table is set up to handle the status in descending order from bit 15 to bit 10 of the external interrupt code.

PDTABA

Byte	0	1	2	3	4	5	6	7
0	00	08	00	10	00	08	00	18
8	00	08	00	10	00	08	00	20
16	00	08	00	10	00	08	00	18
24	00	08	00	10	00	08	00	28
32	00	08	00	10	00	08	00	18
40	00	08	00	10	00	08	00	20
48	00	08	00	10	00	08	00	18
56	00	08	00	10	00	08	00	

Figure 2.12. Tables for MICR DTF Addresses (Part 2 of 2)

Bytes 126 and 127 (X'7E' - X'7F') of the partitions communication region contain the address of these tables. Label PDTABB identifies the first byte of the first table. The tables are used for handling external interrupts on magnetic ink or optical character recognition devices.

Displacement	Label	Description
	(ACCTCOMN)	
0-15	ACCTSVRG	Temporary register save area
16-17	ACCTSVRX	Save area for remainder of overhead counter times distributed by partition on exit
18-19	ACCTSVRE	Save area for remainder of all-bound counter times distributed by partitions on entry
20-23	ACCTPCNT	Count of partitions using the Job Accounting interface
24	ACCTSAID	Owner of physical transient area *)
25	ACCTFAID	Interrupted program *)
26	ACCTRAID	ACTIVE PROGRAM *)
27	ACCTSWCH	Accounting switches: if bit=1, true; if bit=0, not true bit 0: cancel accounting bit 4: IPL indicator bit 1: no active partitions bit 5: not used bit 2: catalog in process bit 6: not used bit 3: alternate label area bit 7: not used
28-31	ACCTIME	Start time of current accounting interval, in complement format
32-33	ACCTRESC	Reserved
34-35	ACCTUSEP	Address of user save area (ACCTUSER)
36-37	ACCTUSEL	Length of user save area (set with 1st operand of FOPT macro parameter JALIOCS)
38-39	ACCT$JOB	Job accounting partition indication
40-43	ACCTBLES	Address of BG Job Accounting Table

If multiprogramming is supported, this table is to be extended with one of the following fields (depending on the number of supported partitions), otherwise the table ends here.

NPARTS=2

44-47		Address of F1 Job Accounting Table
48-51	ACCTSEAS	Control Field: prevents the accounting routine being loaded twice

NPARTS=3

44-47		Address of F2 Job Accounting Table
48-51		Address of F1 Job Accounting Table
52-57	ACCTSEAS	Control Field: prevents the accounting routine being loaded twice

*) These values are the same as the PIK values for the relevant tasks.

Figure 2.13. Job Accounting Interface Common Table (ACCTCOMN) (Part 1 of 2)

NPARTS=4

Displacement	Label	Description
44-47		Address of F3 Job Accounting Table
48-51		Address of F2 Job Accounting Table
52-55		Address of F1 Job Accounting Table
56-63	ACCTSEAS	Control Field: prevents the accounting routine being loaded twice

NPARTS=5

44-47		Address of F4 Job Accounting Table
48-51		Address of F3 Job Accounting Table
52-55		Address of F2 Job Accounting Table
56-59		Address of F1 Job Accounting Table
60-69	ACCTSEAS	Control Field: prevents the accounting routine being loaded twice

Bytes 124-127 (X'7C' - X'7F') of the System Communication Region (SYSCOM) contain the address of the Job Accounting Interface Common Table. Label ACCTCOMN identifies the first byte of the table.

Figure 2.13. Job Accounting Interface Common Table (ACCTCOMN) (Part 2 of 2)

Displacement	Label	Description
	(ACCTABLE)	
0-3	ACCTWK1	Work area used in SIO update
4-7	ACCTWK2	Work area used with ACCTWK1 in start/stop time routine
8-11	ACCTSVPT	Job card pointer; address of job card field following jobname
12-13	ACCTPART	ID of partition in charge (partition switch name)
14-15	ACCTLEN	Length of SIO area=6n+1, where n=number of devices for this partition in SYSGEN option JA=(n1,n2,n3,n4,n5)
16-21	ACCTLOAD	Label area instruction; moves JAI label area address to OPEN/CLOSE transients
22-23	ACCTRES3	Reserved
24-27	ACCTLADD	Address of alternate label area
28-31	ACCTCPUT	Counter for CPU time elapsed in a jobstep, counted in 300th of a second
32-35	ACCTOVHT	Counter for overhead time; time not charged to any partition
36-39	ACCTBNDT	Counter for all-bound time; system wait state time divided between running partitions

Figure 2.14. Job Accounting Interface Partition Table (ACCTxx) (Part 1 of 2)

Displacement	Label	Description
40-47	ACCTSVJN	Save area for job name during simulated EOJ
48-55	ACCTJBNM	Job name; taken from job card
56-71	ACCTUSRS	User information; 16 bytes from job card
72-73	ACCTPTID	Partition ID: 'BG', 'F4', 'F3', 'F2', or 'F1' in EBCDIC format
74	ACCTCNCL	Cancel code; see Cancel Codes and Messages
75	ACCTYPER	Type of record: 'S' = job step, 'L' = last step of job
76-83	ACCTDATE	Date in format specified at SYSGEN (MM/DD/YY or DD/MM/YY)
84-87	ACCTSTRT	Start time of a job step, in packed decimal (DHHMMSSF; F = sign)
88-91	ACCTSTOP	Stop time of a job step in the same format as ACCTSTRT. This value is used as the start time for the next step.
92-95	ACCTRES	Reserved
96-103	ACCTEXEC	Phase name; taken from execute card
104-107	ACCTHICR	<u>Real mode</u>: the problem program end address reflecting ALLOCR or the SIZE= parameter on the EXEC statement. <u>Virtual mode</u>: the virtual start address of the partition incremented by the amount of virtual storage referenced during this step.
108-111	ACCTIMES	CPU time elapsed in a job step; counted in 300th of a second
112-115	--------	Overhead time; elapsed time not charged to any partition, in 300th of a second
116-119	--------	All-bound time; system wait state time divided between running partitions, in 300th of a second
120	ACCTSIOS	SIO tables: 6 bytes for each device specified by SYSGEN options, as follows: 2 bytes for device address (0cuu), 4 bytes for count of SIOs in current jobstep
------	--------	Overflow byte: normally X'20', but is X'30' if more devices are used within a partition than specified by SYSGEN options

<u>Notes</u>:

1. DSECT ACCTABLE symbolically addresses the JAI Partition Tables with labels as shown. Each partition in which JAI is supported has its own JAI Partition Table, labeled ACCTBG, ACCTF4, ACCTF3, ACCTF2, ACCTF1, for active partitions BG, F4, F3, F2, and F1, respectively.

2. The address of this table is in the partition Communication Region at displacement 116 (X'74').

Figure 2.14. Job Accounting Interface Partition Table (ACCTxx) (Part 2 of 2)

DISK INFORMATION BLOCK (DIB) TABLE

DSKPOSBG	BG DIB Table
DSKPOSF4	F4 DIB Table
DSKPOSF3	F3 DIB Table
DSKPOSF2	F2 DIB Table
DSKPOSF1	F1 DIB Table

The number of DIB tables depends on the number of partitions specified at supervisor generation.

Format of any DIB table if SYSFIL= YES

	0						6	7		9	10						16	17	18	19	20	21	22	23
	Current Address							K	D	D	End Address							UL	LL	M	RC			
SYSLNK	C	C	H	H	00	00	00	P	P	00■	00	00	00	00	00	00	00	00	00	00	00	00	00	00
SYSIN	B	B	C	C	H	H	R	00	00	50	B	B	C	C	H	H	R	H	H	XX	XX	XX	00	00
SYSPCH	B	B	C	C	H	H	R	00	00	51	B	B	C	C	H	H	R	H	H	XX	XX	XX	00	00
SYSLST	B	B	C	C	H	H	R	00	00	78	B	B	C	C	H	H	R	H	H	XX	XX	XX	00	00
PRCDIB	B	B	C	C	H	H	R	00	00	50	B	B	C	C	H	H	R	H	H	XX	XX	XX	00	00

Format of any DIB table if SYSFIL= NO

	0						6	7		9	10						16	17	18	19	20	21	22	23
	Current Address							K	D	D	End Address							UL	LL	M	RC			
SYSLNK	C	C	H	H	00	00	00	P	P	00■														
PRCDIB	B	B	C	C	H	H	R	00	00	50	B	B	C	C	H	H	R	H	H	XX	XX	XX	00	00

■) BG SYSLNK DIB contains the PUB pointer for CLB.
For FG SYSLNK DIBs this byte is unused.

Format of 3540 Diskette DIB table

	0						6	7		9	10						16	17	18	19	20	21	22	23
	Current Address							K	D	D	End Address									M	RC			
SYSIN	00	00	00	00	C	H	R	00	00	00	FLG	EXT #	HDR #	00	C	H	R	00	00	XX	00	00	00	00
SYSPCH	00	00	00	00	C	H	R	00	00	00	FLG	EXT #	HDR #	00	C	H	R	00	00	XX	00	00	00	00
SYSLST	00	00	00	00	C	H	R	00	00	00	FLG	EXT #	HDR #	00	C	H	R	00	00	XX	00	00	00	00

Figure 2.15. Disk Information Block (DIB) Table

Bytes

0-6: Current Address

0-5: Current address of key: the next address to be used (both for input and output).
6: Record number of current address.

7-9: KDD

Key and data length of the symbolic device. PP: starting cylinder of Private Core Image Library if PCIL is assigned; otherwise zero.

10-16: End Address

10-15: End address of key: the last address within the limits of the extent.
16: Record number of end address.

17: UL Upper head limit.

18: LL Lower head limit.

19: M Maximum number of records per track.

20-21: RC Record count: Residual capacity for beginning of operator notification. Set at system generation time with SYSFIL parameter, or after IPL with SET statement (RCLST and/or PCPCH operands). A warning message is issued by job control after End-of-Job step when the minimum number of remaining records has been reached or exceeded during the previous job. Not supported for 3540.

22: X'40' Indicates RPS support.

23: Reserved.

Note: The block is initialized by job control with extent information and updated by physical IOCS. When the PCIL option is used the DIB is updated each time the PCIL is assigned.

There is one DIB table for each partition. Label DSKPCSBG identifies the first byte of the BG DIB Table (DSKPCSF1, DSKPCSF2, DSKPCSF3, DSKPCSF4, for the other partitions). The addresses of the DIB tables are contained in bytes 96 and 97 (X'60' and X'61') of the appropriate partition communication region.

Figure 2.16. Deleted

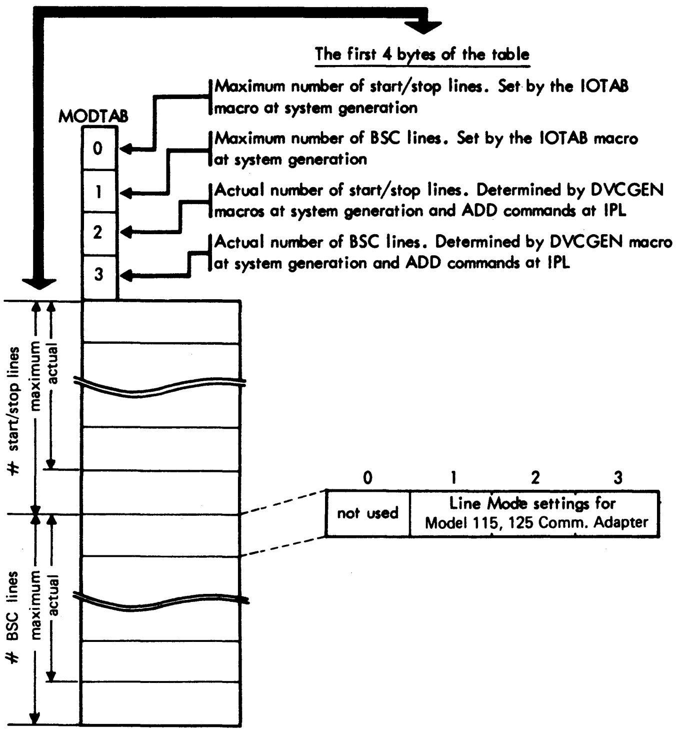


Figure 2.17. Line Mode Table (LMT)

Bytes 140-143 (X'8C' - X'8F') of the System Communication Region (SYSCOM) contain the address of the table. Label MODTAB identifies the first byte of the table.

Displacement		Label	Byte Length	Description
Dec	Hex			
		RFTABLE		Label of Starting Address
0	0	RFFLAGS1	1	Bit 0: 1 = File full 1: 1 = RDE option included 2: 1 = Initial IPL 3: Reserved 4: 1 = File is to be created 5: 1 = File has been created 6: Reserved 7: 1 = File open and ready
1	1	RFFLAGS2	1	Bit 0: 1 = File full message request 1: 1 = Last track message request 2: 1 = I/O error message request 3: 1 = Data lost message request 4: 1 = EVA message request 5: 1 = File owned by RTA recorder 6: 1 = File owned by PTA recorder 7: 1 = File being accessed by EREP
2	2	RFFLAGS3	1	Bit 0: 1 = Last track message issued once 1: 1 = Error is to be recorded 2: 1 = Short form record request 3: 1 = Individual records for unlabeled tapes 4: Reserved 5: Reserved 6: 1 = Exit to $$BOMT05 indicator for $$BOPEN 7: 1 = Exit to $$BOMT01 indicator for $$BOPEN
3	3	RFFLAGS4	1	Work area switches for various transients including BTAM Bit 0: 1 = multiple records required (3211 recording) 1: 1 = PIB record required (3211 recording) 2: 1 = FCB record required (3211 recording) 3: 1 = UCB record required (3211 recording) 4: 1 = Ignore exit requested (3211 recording) 5: Not used 6: Not used 7: 1 = record not written
4	4	RFFLAGS5	1	Reserved
5	5	RFNOFN	1	N of N records (low order 4 bits contain the number of records to be processed and high order 4 bits contain the number of the record being processed)
6	6	RFRECTYP	1	Record type code

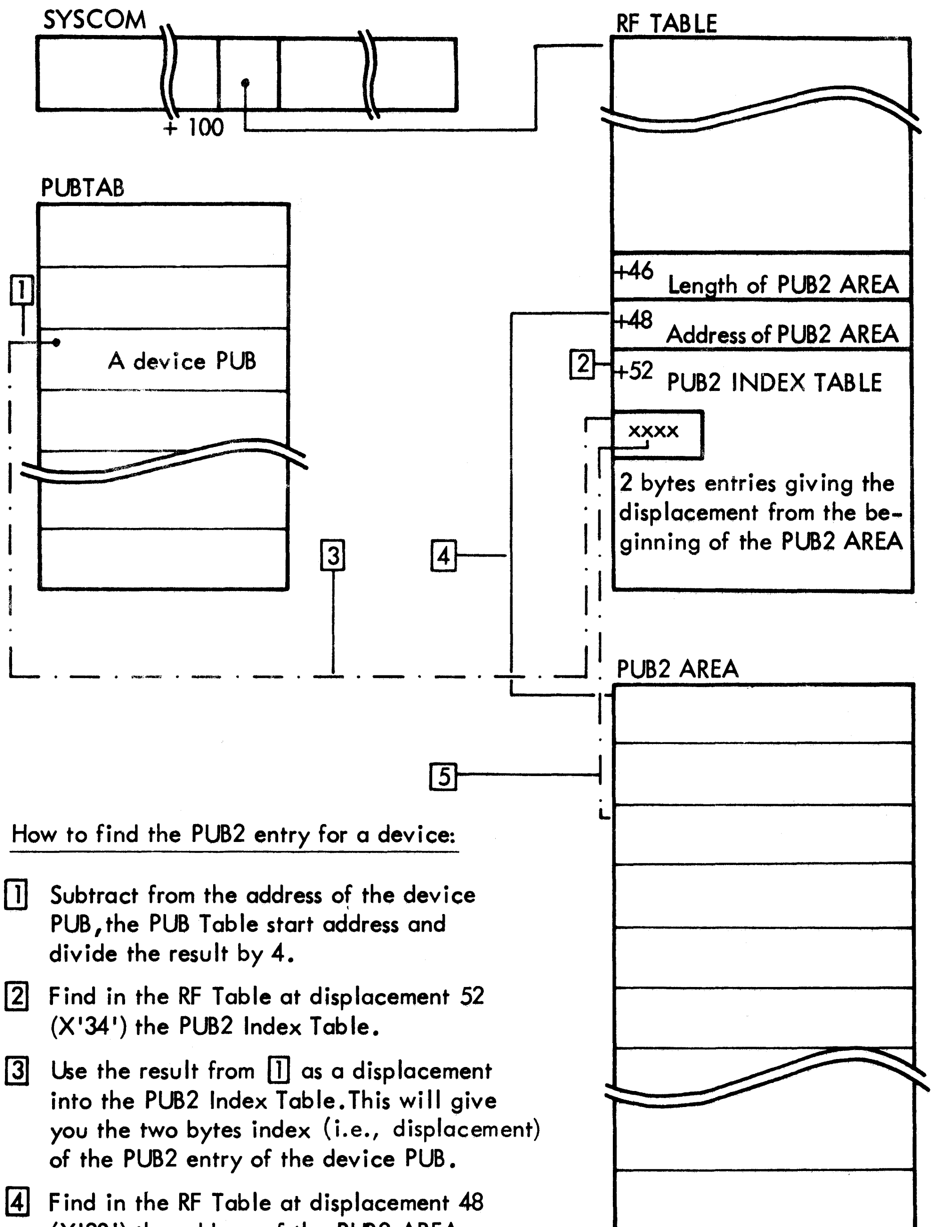
Figure 2.18. Recorder File Table (Part 1 of 2)

Displacement Dec	Displacement Hex	Label	Byte Length	Description
7	7	RFREL	1	DOS ID/Release level code
8	8	RFRDSW1	1	Record dependent switch 1
9	9	RFRDSW2	1	Record dependent switch 2
10	A	RFEXIT	2	Exit phase name or exit address
12	C	RFMCONST	2	Multiplier for track balance
14	E	RFDCONST	2	Divisor for track balance
16	10	RFOCONST	2	Overhead for track balance
18	12	RFRECLEN	2	Length of record
20	14	RFTIMEA	4	Address of RMSR time entry
24	18	RFRECADR	4	Address of record
28	1C	RFSEEK	7	Work area for seek address (BBCCHHR)
35	23	RFEREPK	1	Key of EREP partition
36	24	RFHDRCH	4	SYSREC cylinder/head
40	28	RFCHMAP	1	Map of supported channels
41	29	RFCHIDC	3	Channel ID codes
The following entries are not generated for Model 115 and 125, without RMSR support.				
44	2C	RFEVARTH	1	EVA read threshold
45	2D	RFEVAWTH	1	EVA write threshold
46	2E	RFP2ENTL	2	Length of PUB2 Entry Area
48	30	RFP2ENT	4	Address of PUB2 Entry Area
52	34	RFP2ITAB	*	PUB2 Index Table

*Two bytes are generated for each PUB2 entry in the system. See Figure 2.19 for using the PUB2 Index Table to access the PUB2 entries.

Bytes 100-103 (X'64'-'67') of the system communication region (SYSCOM) contain the address of the Recorder File Table. Label RFTABLE identifies the first byte of the table.

Figure 2.18. Recorder File Table (Part 2 of 2)



How to find the PUB2 entry for a device:

1. Subtract from the address of the device PUB, the PUB Table start address and divide the result by 4.
2. Find in the RF Table at displacement 52 (X'34') the PUB2 Index Table.
3. Use the result from 1 as a displacement into the PUB2 Index Table. This will give you the two bytes index (i.e., displacement) of the PUB2 entry of the device PUB.
4. Find in the RF Table at displacement 48 (X'30') the address of the PUB2 AREA.
5. Use the value from 3 as a displacement in the PUB2 AREA.

Note: PUB2 Area is not generated for Model 115 and 125, without RMSR suppcrt.

Figure 2.19. Accessing the PUB2 Table (PUB2AREA)

Decimal Displacement	Byte Length	Description
0	3	Usage Count (number of non-ERP SIOs)
3	1	Flag Byte Bit 0: 1 = Device in intensive mode 1: 1 = Device in diagnostic mode 2: 1 = No recording mode 3: 1 = Call statistics transient 2 4: 1 = Use PUB2 name completion field 5: 1 = Volume opened on this device 6,7: Reserved
4	1	CE mode limit byte
5	1	CE mode byte/bit mask
6	6	Statistical data counters
Total length 12		

Note: PUB2 Table is not generated for Model 115 and 125, without RMSR support.

Figure 2.20. PUB2 Table Entry Format for Unit Record and Unsupported Devices

Decimal Displacement	Byte Length	Description
0	3	Usage Count (number of non-ERP SIOs)
3	1	Flag Byte Bit 0: 1 = Device in intensive mode 1: 1 = Device in diagnostic mode 2: 1 = No recording mode 3: 1 = Call statistics transient 2 4: 1 = Use PUB2 name completion field 5: 1 = Volume opened on this device 6,7: Reserved
4	1	CE mode limit byte
5	1	CE mode byte/bit mask
6	20	Statistical data counters
Total length 26		

Note: PUB2 Table is not generated for Model 115 and 125, without RMSR support.

Figure 2.21. PUB2 Table Entry Format for 3886 Optical Character Reader

Decimal Displacement	Byte Length	Description
0	3	Usage Count (number of non-ERP SIOs)
3	1	Flag Byte Bit 0: 1 = Device in intensive mode 1: 1 = Device in diagnostic mode 2: 1 = No recording mode 3: 1 = Call statistics transient 2 4: 1 = Use PUB2 name completion field 5: 1 = Volume opened on this device 6,7: Reserved
4	1	CE mode limit byte
5	1	CE mode byte/bit mask
6	1	Flag byte Bit 0: 1 = Soft DASD error is queued 1: 1 = ERP requests logging of error 2-7: Reserved
7	2	Reserved
9	1	Physical module identifier
10	6	Volume serial number
End 3340 Total Length 16		
16	8	Statistical data counters
End DASD except 3330 Total length 24		
24	8	Additional statistical data counters (3330 only)
End 3330 Total length 32		

<u>Note</u>: PUB2 Table is not generated for Model 115 and 125, without RMSR support.

Figure 2.22. PUB2 Table Entry Format for DASD

Decimal Displacement	Byte Length	Description
0	3	Usage Count (number of non-ERP SIOs)
3	1	Flag Byte Bit 0: 1 = Device in intensive mode 1: 1 = Device in diagnostic mode 2: 1 = No recording mode 3: 1 = Call statistics transient 2 4: 1 = Use PUB2 name completion field 5: 1 = Volume opened on this device 6,7: Reserved
4	1	CE mode limit byte
5	1	CE mode byte/bit mask
6	8	Statistical data counters
14	1	Flag byte Bit 0-5: Reserved 6: System File opened by Job Control 7: System File opened by Problem Program
15	1	Reserved for future use
Total length 16		

Figure 2.23. PUB2 Table Entry Format for 3540 Diskette

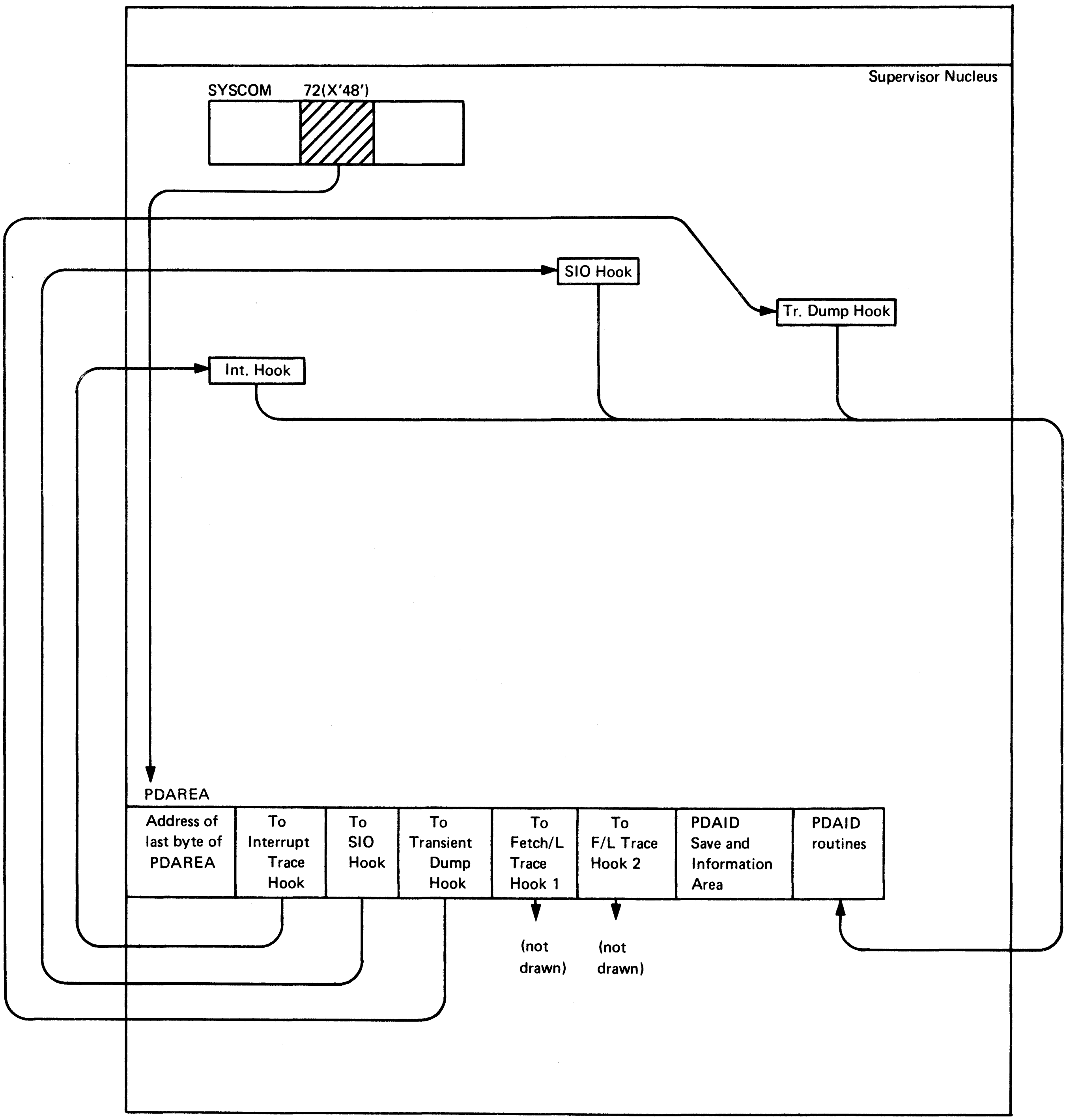
Decimal Displacement	Byte Length	Description
0	3	Usage Count (number of non-ERP SIOs)
3	1	Flag Byte Bit 0: 1 = Device in intensive mode 1: 1 = Device in diagnostic mode 2: 1 = No recording mode 3: 1 = Call statistics transient 2 4: 1 = Use PUB2 name completion field 5: 1 = Volume opened on this device 6,7: Reserved
4	1	CE mode limit byte
5	1	CE mode byte/bit mask
6	2	Name of ERP that wants control
8	1	Flag byte 1 Bit 0: Reserved 1: 1 = Unsolicited interrupt 2: 1 = ERP is in control 3: 1 = ERP requests repositioning 4: 1 = Use original TIE byte 0 = Use opposite TIE byte 5: 1 = Intercept next SIO request 6: 1 = ERP read opposite request 7: 1 = Restart user's CCW chain
9	1	Flag byte 2 Bit 0: 1 = Last ERP operation was ERG 1: 1 = Last ERP operation was reposition 2: 1 = Cleaner action in progress 3: 1 = Read Opposite Recovery in progress 4: 1 = Message stored in P2ORGTIE 5: 1 = Error on attempt to recover by repositioning 6: 1 = Data check after ERP in control 7: Reserved
10	1	Flag byte 3 Bit 0: 1 = Failing CCW is Write or Control command 1: 1 = User reading backwards 2: 1 = Read Opposite Recovery (ROR) 3: 1 = Maximum ROR retries 4: 1 = Command chaining ROR 5: 1 = ROR suppressed incorrect length 6,7: Reserved

Figure 2.24. PUB2 Table Entry Format for Tapes (Part 1 of 2)

Decimal Displacement	Byte Length	Description
11	1	Temporary read count
12	1	Temporary write count
13	1	Noise record count
14	2	Erase gap count
16	2	Cleaner action count
18	1	Permanent read errors count
19	1	Permanent write errors count
20	1	TIE original direction
21	1	TIE opposite direction
22	1	ERP counter 0
23	1	ERP counter 1
24	8	ERP work area
32	6	Tape serial number
38	2	Block length
40	4	User ROR command address from CSW
44	2	User ROR residual count from CSW
46	2	Reserved
48	10	2400-series statistical data counter area
58	2	Reserved
End 2400-series: Total length 60		
48	20	3410/3420 statistical data counter area
End 3410/3420: Total length 68		

Note: PUB2 Table is not generated for Model 115 and 125, without RMSR support.

Figure 2.24. PUB2 Table Entry Format for Tapes (Part 2 of 2)



Note: The PDAREA is present when PD= YES or PD= n was specified on the FOPT macro. The minimum size is 1400 bytes.

Figure 2.25. Accessing PDAID Rcutines

Extent Information	Number of Pages supported	Volume ID for Page Data Set	Address of Constant Table for Load Leveller
0 … 11	12 … 13	14 … 19	20 … 23

Figure 2.26. Page Data Set Table (DPDTAB)

Extent Information (Bytes 0-11):

Bytes:

0- 1: Channel and Unit number of SYSVIS Device
2- 3: Internal device code of SYSVIS Device (offset in Page Manager Device Constants List)

0 = 2314
4 = 3330
8 = 3340

4- 7: Relative track address of Lower Limit
8-11: Relative track address of Upper Limit

Bytes 224-227 (X'E0' - X'E3') of the System Communication Region (SYSCOM) contain the address of the DPDTAB. Label DPDTAB identifies the first byte of the table.

CRTTAB

Name of CRT Routine	Dummy Sense Byte	Address of CRT Save Area	Flag Byte 1	Address of CRT Trans.Area	Flag Byte 2	Address of System Task Save Area	CRT error Information *)
0 … 7	8	9 … 11	12	13 … 15	16	17 … 19	20 … 27

*) limited channel logout is saved here

Figure 2.27. CRT Constant Table (CRTTAB)

Byte 8 - Sense Byte:

Bit 0: 1 = Command reject
1: 1 = Intervention required (only if Console Printer is attached)
4: 1 = Equipment check (only if Console Printer is attached)
7: 1 = Operation check

or: this byte is used for saving Name Indicator

Byte 12 - Flag Byte 1:

X'80' = ERP message
X'40' = Unit check for CRT
X'20' = CRT Fetch bound
X'10' = Device End simulated
X'08' = Validation error
X'04' = Redisplay mode
X'02' = CRT error

Byte 16 - Flag Byte 2:

X'80' = CRT busy
X'40' = Sense byte prepared
X'20' = End of CRT routine
X'10' = Data already read
X'08' = Attention pending
X'04' = Request pending
X'02' = Attention request
X'01' = EOJ on CRT

Bytes 52-55 (X'34' - X'37') of the System Communication Region (SYSCOM) contain the address of the CRT CONSTANT Table. Label CRTTAB identifies the first byte of the table.

System Task	Label(s)	Contents	Length in Bytes
RAS	RASSAVAR	Interrupt Information and General Registers	72
PMGR	PMRSAVAR	Interrupt Information and General Registers	72
SUPVR+ ERP	SUPSAVAR ERPSAVAR	Interrupt Information and General Registers	72
PAGEIN	PGNSAVAR	Interrupt Information and General Registers	72
CRT	CRTSAV	CRT Save Area - Set on Doubleword Boundary	
	ATTBUF	Attention Buffer	80
	SYSTEMSV(+8)	Save Area for System Tasks	72
	CRTSAVAR(+8)	Interrupt Information and General Registers	72
	CSAVEAR	Channel Scheduler Save Area	28
	SCRIMG	Buffer for Screen Image	634
	CRTCCW	Constants Needed for CCW Processing	47
	CRTFLGHC	Constants Needed for Hard Copy Processing	38
	PARTRED	Constants Used by CRT Redisplay Feature	40
	SEGVAL0	Constants Needed for Screen Management	37
	SCRNCTL	Screen Control Table	33

Figure 2.28. System Task Save Areas and CRT Constants

ASCII to EBCDIC Correspondence (0/0 to 3/15)

ASCII					EBCDIC				
Character	Col	Row	Bit Pattern		Col (in Hex)	Row (in Hex)	Bit Pattern		Comments
NUL	0	0	0000	0000	0	0	0000	0000	
SOH	0	1	0000	0001	0	1	0000	0001	
STX	0	2	0000	0010	0	2	0000	0010	
ETX	0	3	0000	0011	0	3	0000	0011	
EOT	0	4	0000	0100	3	7	0011	0111	
ENQ	0	5	0000	0101	2	D	0010	1101	
ACK	0	6	0000	0110	2	E	0010	1110	
BEL	0	7	0000	0111	2	F	0010	1111	
BS	0	8	0000	1000	1	6	0001	0110	
HT	0	9	0000	1001	0	5	0000	0101	
LF	0	10	0000	1010	2	5	0010	0101	
VT	0	11	0000	1011	0	B	0000	1011	
FF	0	12	0000	1100	0	C	0000	1100	
CR	0	13	0000	1101	0	D	0000	1101	
SO	0	14	0000	1110	0	E	0000	1110	
SI	0	15	0000	1111	0	F	0000	1111	
DLE	1	0	0001	0000	1	0	0001	0000	
DC1	1	1	0001	0001	1	1	0001	0001	
DC2	1	2	0001	0010	1	2	0001	0010	
DC3	1	3	0001	0011	1	3	0001	0011	
DC4	1	4	0001	0100	3	C	0011	1100	
NAK	1	5	0001	0101	3	D	0011	1101	
SYN	1	6	0001	0110	3	2	0011	0010	
ETB	1	7	0001	0111	2	6	0010	0110	
CAN	1	8	0001	1000	1	8	0001	1000	
EM	1	9	0001	1001	1	9	0001	1001	
SUB	1	10	0001	1010	3	F	0011	1111	
ESC	1	11	0001	1011	2	7	0010	0111	
FS	1	12	0001	1100	1	C	0001	1100	
GS	1	13	0001	1101	1	D	0001	1101	
RS	1	14	0001	1110	1	E	0001	1110	
US	1	15	0001	1111	1	F	0001	1111	
SP	2	0	0010	0000	4	0	0100	0000	
|	2	1	0010	0001	4	F	0100	1111	Logical OR
"	2	2	0010	0010	7	F	0111	1111	
#	2	3	0010	0011	7	B	0111	1011	
$	2	4	0010	0100	5	B	0101	1011	
%	2	5	0010	0101	6	C	0110	1100	
&	2	6	0010	0110	5	0	0101	0000	
'	2	7	0010	0111	7	D	0111	1101	
(	2	8	0010	1000	4	D	0100	1101	
)	2	9	0010	1001	5	D	0101	1101	
*	2	10	0010	1010	5	C	0101	1100	
+	2	11	0010	1011	4	E	0100	1110	
,	2	12	0010	1100	6	B	0110	1011	
-	2	13	0010	1101	6	0	0110	0000	Hyphen, Minus
.	2	14	0010	1110	4	B	0100	1011	
/	2	15	0010	1111	6	1	0110	0001	
0	3	0	0011	0000	F	0	1111	0000	
1	3	1	0011	0001	F	1	1111	0001	
2	3	2	0011	0010	F	2	1111	0010	
3	3	3	0011	0011	F	3	1111	0011	
4	3	4	0011	0100	F	4	1111	0100	
5	3	5	0011	0101	F	5	1111	0101	
6	3	6	0011	0110	F	6	1111	0110	
7	3	7	0011	0111	F	7	1111	0111	
8	3	8	0011	1000	F	8	1111	1000	
9	3	9	0011	1001	F	9	1111	1001	
:	3	10	0011	1010	7	A	0111	1010	
;	3	11	0011	1011	5	E	0101	1110	
<	3	12	0011	1100	4	C	0100	1100	
=	3	13	0011	1101	7	E	0111	1110	
>	3	14	0011	1110	6	E	0110	1110	
?	3	15	0011	1111	6	F	0110	1111	

Figure 2.29. ASCII Translation Tables (Part 1 of 4)

ASCII to EBCDIC Correspondence (4/0 to 7/15)

ASCII					EBCDIC				
Character	Col	Row	Bit Pattern		Col (in Hex)	Row	Bit Pattern		Comments
@	4	0	0100	0000	7	C	0111	1100	
A	4	1	0100	0001	C	1	1100	0001	
B	4	2	0100	0010	C	2	1100	0010	
C	4	3	0100	0011	C	3	1100	0011	
D	4	4	0100	0100	C	4	1100	0100	
E	4	5	0100	0101	C	5	1100	0101	
F	4	6	0100	0110	C	6	1100	0110	
G	4	7	0100	0111	C	7	1100	0111	
H	4	8	0100	1000	C	8	1100	1000	
I	4	9	0100	1001	C	9	1100	1001	
J	4	10	0100	1010	D	1	1101	0001	
K	4	11	0100	1011	D	2	1101	0010	
L	4	12	0100	1100	D	3	1101	0011	
M	4	13	0100	1101	D	4	1101	0100	
N	4	14	0100	1110	D	5	1101	0101	
O	4	15	0100	1111	D	6	1101	0110	
P	5	0	0101	0000	D	7	1101	0111	
Q	5	1	0101	0001	D	8	1101	1000	
R	5	2	0101	0010	D	9	1101	1001	
S	5	3	0101	0011	E	2	1110	0010	
T	5	4	0101	0100	E	3	1110	0011	
U	5	5	0101	0101	E	4	1110	0100	
V	5	6	0101	0110	E	5	1110	0101	
W	5	7	0101	0111	E	6	1110	0110	
X	5	8	0101	1000	E	7	1110	0111	
Y	5	9	0101	1001	E	8	1110	1000	
Z	5	10	0101	1010	E	9	1110	1001	
[	5	11	0101	1011	4	A	0100	1010	
\	5	12	0101	1100	E	0	1110	0000	Reverse Slant
]	5	13	0101	1101	5	A	0101	1010	
¬	5	14	0101	1110	5	F	0101	1111	Logical NOT
_	5	15	0101	1111	6	D	0110	1101	Underscore
`	6	0	0110	0000	7	9	0111	1001	Grave Accent
a	6	1	0110	0001	8	1	1000	0001	
b	6	2	0110	0010	8	2	1000	0010	
c	6	3	0110	0011	8	3	1000	0011	
d	6	4	0110	0100	8	4	1000	0100	
e	6	5	0110	0101	8	5	1000	0101	
f	6	6	0110	0110	8	6	1000	0110	
g	6	7	0110	0111	8	7	1000	0111	
h	6	8	0110	1000	8	8	1000	1000	
i	6	9	0110	1001	8	9	1000	1001	
j	6	10	0110	1010	9	1	1001	0001	
k	6	11	0110	1011	9	2	1001	0010	
l	6	12	0110	1100	9	3	1001	0011	
m	6	13	0110	1101	9	4	1001	0100	
n	6	14	0110	1110	9	5	1001	0101	
o	6	15	0110	1111	9	6	1001	0110	
p	7	0	0111	0000	9	7	1001	0111	
q	7	1	0111	0001	9	8	1001	1000	
r	7	2	0111	0010	9	9	1001	1001	
s	7	3	0111	0011	A	2	1010	0010	
t	7	4	0111	0100	A	3	1010	0011	
u	7	5	0111	0101	A	4	1010	0100	
v	7	6	0111	0110	A	5	1010	0101	
w	7	7	0111	0111	A	6	1010	0110	
x	7	8	0111	1000	A	7	1010	0111	
y	7	9	0111	1001	A	8	1010	1000	
z	7	10	0111	1010	A	9	1010	1001	
{	7	11	0111	1011	C	0	1100	0000	
|	7	12	0111	1100	6	A	0110	1010	Vertical Line
}	7	13	0111	1101	D	0	1101	0000	
~	7	14	0111	1110	A	1	1010	0001	Tilde
DEL	7	15	0111	1111	0	7	0000	0111	

Figure 2.29. ASCII Translation Tables (Part 2 of 4)

EBCDIC to ASCII Correspondence (X'00' to X'82')

EBCDIC					ASCII				
Character	Col (in Hex)	Row (in Hex)	Bit Pattern		Col	Row	Bit Pattern		Comments
NUL	0	0	0000	0000	0	0	0000	0000	
SOH	0	1	0000	0001	0	1	0000	0001	
STX	0	2	0000	0010	0	2	0000	0010	
ETX	0	3	0000	0011	0	3	0000	0011	
HT	0	5	0000	0101	0	9	0000	1001	
DEL	0	7	0000	0111	7	15	0111	1111	
VT	0	B	0000	1011	0	11	0000	1011	
FF	0	C	0000	1100	0	12	0000	1100	
CR	0	D	0000	1101	0	13	0000	1101	
SO	0	E	0000	1110	0	14	0000	1110	
SI	0	F	0000	1111	0	15	0000	1111	
DLE	1	0	0001	0000	1	0	0001	0000	
DC1	1	1	0001	0001	1	1	0001	0001	
DC2	1	2	0001	0010	1	2	0001	0010	
DC3	1	3	0001	0011	1	3	0001	0011	
BS	1	6	0001	0110	0	8	0000	1000	
CAN	1	8	0001	1000	1	8	0001	1000	
EM	1	9	0001	1001	1	9	0001	1001	
FS	1	C	0001	1100	1	12	0001	1100	
GS	1	D	0001	1101	1	13	0001	1101	
RS	1	E	0001	1110	1	14	0001	1110	
US	1	F	0001	1111	1	15	0001	1111	
LF	2	5	0010	0101	0	10	0000	1010	
ETB	2	6	0010	0110	1	7	0001	0111	
ESC	2	7	0010	0111	1	11	0001	1011	
ENQ	2	D	0010	1101	0	5	0000	0101	
ACK	2	E	0010	1110	0	6	0000	0110	
BEL	2	F	0010	1111	0	7	0000	0111	
SYN	3	2	0011	0010	1	6	0001	0110	
EOT	3	7	0011	0111	0	4	0000	0100	
DC4	3	C	0011	1100	1	4	0001	0100	
NAK	3	D	0011	1101	1	5	0001	0101	
SUB	3	F	0011	1111	1	10	0001	1010	
SP	4	0	0100	0000	2	0	0010	0000	
[	4	A	0100	1010	5	11	0101	1011	
	4	B	0100	1011	2	14	0010	1110	
<	4	C	0100	1100	3	12	0011	1100	
(	4	D	0100	1101	2	8	0010	1000	
+	4	E	0100	1110	2	11	0010	1011	
|	4	F	0100	1111	2	1	0010	0001	Logical OR
&	5	0	0101	0000	2	6	0010	0110	
]	5	A	0101	1010	5	13	0101	1101	
$	5	B	0101	1011	2	4	0010	0100	
*	5	C	0101	1100	2	10	0010	1010	
)	5	D	0101	1101	2	9	0010	1001	
;	5	E	0101	1110	3	11	0011	1011	
¬	5	F	0101	1111	5	14	0101	1110	Logical NOT
-	6	0	0110	0000	2	13	0010	1101	Hyphen, Minus
/	6	1	0110	0001	2	15	0010	1111	
¦	6	A	0110	1010	7	12	0111	1100	Vertical Line
,	6	B	0110	1011	2	12	0010	1100	
%	6	C	0110	1100	2	5	0010	0101	
_	6	D	0110	1101	5	15	0101	1111	Underscore
>	6	E	0110	1110	3	14	0011	1110	
?	6	F	0110	1111	3	15	0011	1111	
`	7	9	0111	1001	6	0	0110	0000	Grave Accent
:	7	A	0111	1010	3	10	0011	1010	
#	7	B	0111	1011	2	3	0010	0011	
@	7	C	0111	1100	4	0	0100	0000	
'	7	D	0111	1101	2	7	0010	0111	
=	7	E	0111	1110	3	13	0011	1101	
"	7	F	0111	1111	2	2	0010	0010	
a	8	1	1000	0001	6	1	0110	0001	
b	8	2	1000	0010	6	2	0110	0010	

Figure 2.29. ASCII Translation Tables (Part 3 of 4)

EBCDIC to ASCII Correspondence (X'83' to X'F9')

EBCDIC					ASCII				
Character	Col (in Hex)	Row (in Hex)	Bit Pattern		Col	Row	Bit Pattern		Comments
c	8	3	1000	0011	6	3	0110	0011	
d	8	4	1000	0100	6	4	0110	0100	
e	8	5	1000	0101	6	5	0110	0101	
f	8	6	1000	0110	6	6	0110	0110	
g	8	7	1000	0111	6	7	0110	0111	
h	8	8	1000	1000	6	8	0110	1000	
i	8	9	1000	1001	6	9	0110	1001	
j	9	1	1001	0001	6	10	0110	1010	
k	9	2	1001	0010	6	11	0110	1011	
l	9	3	1001	0011	6	12	0110	1100	
m	9	4	1001	0100	6	13	0110	1101	
n	9	5	1001	0101	6	14	0110	1110	
o	9	6	1001	0110	6	15	0110	1111	
p	9	7	1001	0111	7	0	0111	0000	
q	9	8	1001	1000	7	1	0111	0001	
r	9	9	1001	1001	7	2	0111	0010	
~	A	1	1010	0001	7	14	0111	1110	Tilde
s	A	2	1010	0010	7	3	0111	0011	
t	A	3	1010	0011	7	4	0111	0100	
u	A	4	1010	0100	7	5	0111	0101	
v	A	5	1010	0101	7	6	0111	0110	
w	A	6	1010	0110	7	7	0111	0111	
x	A	7	1010	0111	7	8	0111	1000	
y	A	8	1010	1000	7	9	0111	1001	
z	A	9	1010	1001	7	10	0111	1010	
{	C	0	1100	0000	7	11	0111	1011	
A	C	1	1100	0001	4	1	0100	0001	
B	C	2	1100	0010	4	2	0100	0010	
C	C	3	1100	0011	4	3	0100	0011	
D	C	4	1100	0100	4	4	0100	0100	
E	C	5	1100	0101	4	5	0100	0101	
F	C	6	1100	0110	4	6	0100	0110	
G	C	7	1100	0111	4	7	0100	0111	
H	C	8	1100	1000	4	8	0100	1000	
I	C	9	1100	1001	4	9	0100	1001	
}	D	0	1101	0000	7	13	0111	1101	
J	D	1	1101	0001	4	10	0100	1010	
K	D	2	1101	0010	4	11	0100	1011	
L	D	3	1101	0011	4	12	0100	1100	
M	D	4	1101	0100	4	13	0100	1101	
N	D	5	1101	0101	4	14	0100	1110	
O	D	6	1101	0110	4	15	0100	1111	
P	D	7	1101	0111	5	0	0101	0000	
Q	D	8	1101	1000	5	1	0101	0001	
R	D	9	1101	1001	5	2	0101	0010	
\	E	0	1110	0000	5	12	0101	1100	Reverse Slant
S	E	2	1110	0010	5	3	0101	0011	
T	E	3	1110	0011	5	4	0101	0100	
U	E	4	1110	0100	5	5	0101	0101	
V	E	5	1110	0101	5	6	0101	0110	
W	E	6	1110	0110	5	7	0101	0111	
X	E	7	1110	0111	5	8	0101	1000	
Y	E	8	1110	1000	5	9	0101	1001	
Z	E	9	1110	1001	5	10	0101	1010	
0	F	0	1111	0000	3	0	0011	0000	
1	F	1	1111	0001	3	1	0011	0001	
2	F	2	1111	0010	3	2	0011	0010	
3	F	3	1111	0011	3	3	0011	0011	
4	F	4	1111	0100	3	4	0011	0100	
5	F	5	1111	0101	3	5	0011	0101	
6	F	6	1111	0110	3	6	0011	0110	
7	F	7	1111	0111	3	7	0011	0111	
8	F	8	1111	1000	3	8	0011	1000	
9	F	9	1111	1001	3	9	0011	1001	

Figure 2.29. ASCII Translation Tables (Part 4 of 4)

The dispatching of system and user tasks is controlled by several fields in the supervisor. The SELECT byte, which is the second byte in the System Task Identification Field (STID), and the first part of the fields TRTFLD and TIDFLD are used to select system tasks. The second part of TRTFLD (labeled TRFLD) and the second part of TIDFLD (labeled MVCFLD) are required for user task selection.

The field TRTFLD contains the PIB flags of the tasks, in descending order of task priority, while the field TIDFLD is used to store the ID or PIK of a task. Refer to Figure 3.1 for the layout of the contiguous fields TRTFLD and TIDFLD.

To actually select a task, an additional table, TRTMASK, is required. This table gates resources and contains the various

TRTFLD: System Task PIB Flags						TRFLD: User PIB Flags (after TR instruction)				TIDFLD: System Task ID						MVCFLD: User Task ID			
87	87	87	87	87	87	xx	xx	xx	xx	01	02	06	03	04	05	00	30	20	10
RAS	PMGR	PGN	SUPVR	CRT	ERP					RAS	PMGR	PGN	SUPVR	CRT	ERP				

Figure 3.1. TRTFLD, TRFLD, TIDFLD, and MVCFLD, assuming that NPARTS=3 and AP=NO

Label	Generated if	Related Flags in PIB*	Function	Length in Bytes
TRTAM	AP=YES and VSAM=YES	X'3D' through X'45'	Used by VSAM to set partition PFIX-bound	10
TRTFX	AP=YES and PFIX=YES	X'47' through X'4F'	Used by PFIX routines to set partition PFIX-bound	10
TRTEXT	AP=YES	X'51' through X'53'	Used to gate SVCs 38, 41, and 42	16
TRTLL	NPARTS>1	X'61' through X'6F'	Used by load leveler to deactivate partition; gate SVC 35; used for TFIX and IDRA	16
TRTSYS	Always	X'71' through X'7D'	Used to gate SVC 58; to seize system; indicates 'no buffer available', 'all pages fixed', 'channel queue full', 'console buffer table full' (only if CBF=n)	16
TRTMSK	Always	X'80' through X'87'	Mask for task selection. Indicates LTA and PTA in use and common wait condition	8

*See also Figure 2.7

Figure 3.2. Resource Table (TRTMASK)

selection masks used to scan the PIB flags. The structure of TRTMASK is shown in Figure 3.2.

SYSTEM TASK SELECTION

Whenever exit is taken through the General Exit Routine, the SELECT byte (see Figure 3.3) is tested. If it is not zero, that is, a system task is active, this system task is dispatched immediately. If the SELECT byte is zero, TRTFLD is scanned for a dispatchable system task. If a task is found, it is dispatched by the System Task Dispatcher. Otherwise TRFLD is scanned for a dispatchable user task. The system enters a wait state if neither a system nor a user task is ready to run.

```
┌─────────┬─────────┐
│         │  Byte   │
└─────────┴─────────┘
     0         1
```

Figure 3.3. System Task Identification Field (STID)

Byte 0: Always zero

Byte 1 - Select Byte:

X'00' = No System Task active
X'01' = RAS active
X'02' = PMGR active
X'03' = SUPVR active
X'04' = CRT active
X'05' = ERP active
X'06' = PAGEIN active

The STID field is located in SYSCOM.

DISPATCHING A SYSTEM TASK

To dispatch a system task, the supervisor sets up an address in the System Communication Region (SYSCOM) which points to the system task block (see Figure 3.4) for the selected system task. The Partition Identification Key (PIK) of the serviced user task is then moved from the system task block (from field SYSPIK) to the Background Communications Region (BGCOMREG). If AP=YES is specified, the TIK is retrieved from SYSPIK. The TIK is then used to obtain the PIK of the serviced user task from the user's PIB extension. The PSW and the general registers are restored from the System Task Save Area.

If job accounting is supported, the job accounting ID is retrieved from the system task block and moved to the Job Accounting Interface Table (ACCTCOMN).

INTERRUPTION OF A SYSTEM TASK

When a system task is interrupted, the Partition Identification Key (PIK) of the serviced user task is stored in the System Task Block, and the Old PSW and the general registers are saved in the System Task Save Area.

If Job Accounting is supported, the Job Accounting ID is also saved in the STB.

If the interruption is processed without initiating another system task, return is made to the system task selection in the General Exit Routine which, on finding that the selected task was interrupted, restores the entry registers and returns control immediately to the interrupted task.

In processing the interruption, another system task having a higher priority may be activated. When the higher priority task terminates, control is returned to the interrupted system task at the point where it was interrupted.

USER TASK SELECTION

If no system task is ready to run, the General Exit Routine tries to dispatch a user task. Dispatching of user tasks is similar to that of system tasks.

Two fields, TRFLD and MVCFLD (see Figure 3.1), are used to select a dispatchable user task. The TRFLD contains initially all zeros. It is used later on to store the translated PIB flags.

The first byte of the MVCFLD contains the key of the attention task (X'00'), and the remaining bytes contain the keys of the partitions in descending order of priority.

The lengths of both fields are equal to NPARTS plus 1 if AP=NO. If AP=YES, the lengths of MVCFLD and TRFLD are 17 bytes, the last of which is always zero. Initially, the high-order MVCFLD bytes contain the same values as for the supervisor without AP, and the remaining bytes are zeros. The bytes that are not used for the maintask partition keys are available for subtask keys, each of which is inserted immediately before the key of the maintask to which it is being attached. The first subtask attached therefore has priority for selection within its partition until it is detached, whereupon its key in the MVCFLD field is overlaid by moving the contents of all the lower-order bytes one byte higher.

In the task selection routine, the values in the MVCFLD bytes are moved to the corresponding TRFLD bytes and are then used to reference the PIB flag bytes which are translated sequentially into the TRFLD field. The flag byte of each PIB indicates the status of the program to which that PIB refers (refer to the section "PIB Flag Expansions"). The flag bytes in the TRFLD are then used to scan the TRTMSK field (see Figure 3.2). When any of the bytes in TRTMSK field is not zero, the program or task to which that PIB refers is selected to run.

EXAMPLE OF USER TASK SELECTION

Figure 3.5 illustrates how a user task is selected. It is assumed that no system task was ready to run and that the supervisor was generated with NPARTS=3 and AP=NO.

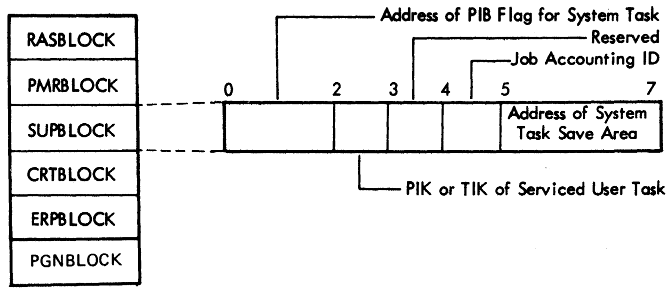
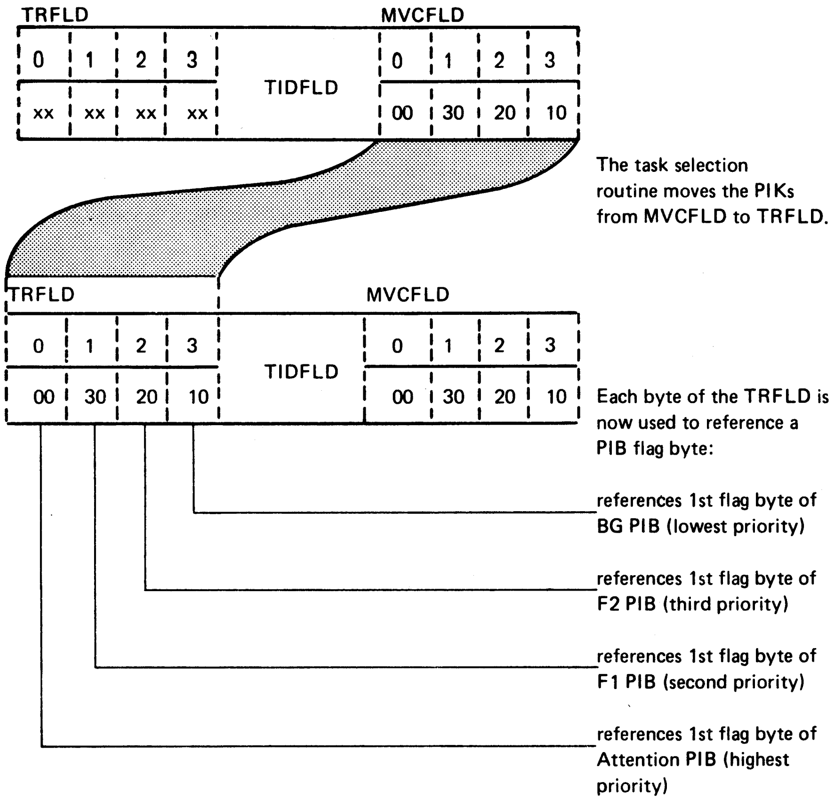


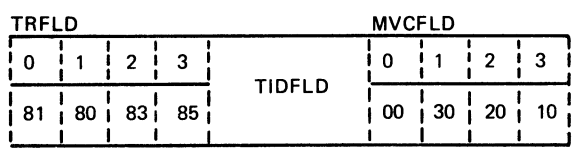
Figure 3.4. System Task Blocks

Notes: Bytes 152-155 (X'98' - X'9B') of the System Communication Region (SYSCOM) contain the address of the first System Task Block.

*These labels identify the first byte of the appropriate corresponding Text Block.



The referenced PIB flag byte is translated into TRFLD.
It is assumed that the following values are translated into TRFLD:



The bytes of the TRFLD field reference the bytes of the TRTMSK which is part of the table TRTMASK (see Figure 3.2). TRTMSK has the following format:

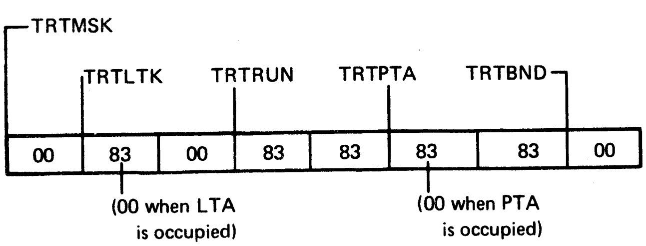
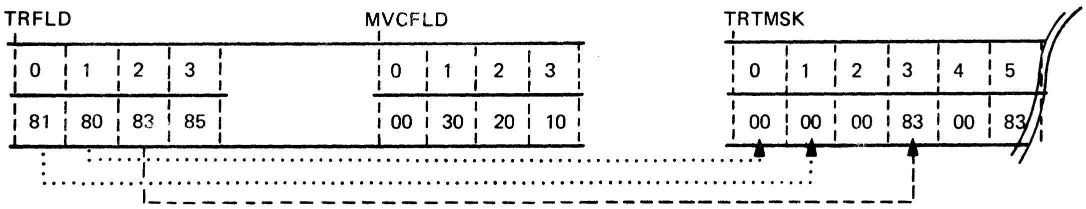


Figure 3.5. Example of User Task Selection (Part 1 of 2)

When the first nonzero TRTMSK function byte is referenced the address of the TRFLD argument byte is inserted in general register 1 which is then incremented by the length of the TRFLD field so that register 1 points to the corresponding constant in the MVCFLD field. In the example, it is assumed that TRTMSK has the following values:



The sequence of events is:

1. TRFLD byte 0 references TRTMSK byte 1. Since the latter is zero, there is no selection (Attention routine LTA–bound and LTA is occupied).

2. TRFLD byte 1 references TRTMSK byte 0. Since the latter is zero, there is no selection (no program or task in F1 partition).

3. TRFLD byte 2 references TRTMSK byte 3. Since the latter is nonzero (X'83') selection takes place:

   a. General register 1 points to TRFLD byte 2.

   b. The length of TRTFLD is added to register 1 which then points to MVCFLD byte 2.

   c. The partition key for the selected program (F2 partition) is moved from MVCFLD byte 2 to the low–order byte of the partition identification key (PIK) in the BG communication region (BGCOMREG).

Figure 3.5. Example of User Task Selection (Part 2 of 2)

DAT DISPATCHER

Before a user task is dispatched, the task selection routine tests whether control has to be transferred to the DAT Dispatcher (bits 1-7 of PIB DAT Flag are not zero). The DAT Dispatcher selects the routine to be called using the PIB DAT Flag byte. Execution of one of the following routines may be necessary:

- SVRETURN (PIB DAT Flag = X'01') - Return to reenterable supervisor routine.

- INITSVC (PIB DAT Flag = X'03') - Open SVC routine previously gated and return to interrupted SVC.

- MOVECCB (PIB DAT Flag = X'04') - Move CCB which could not be copied back after completion of channel program translation because the page containing the virtual CCB was not in real storage.

- EXTRETRN (PIB DAT Flag = X'08') - Service page fault interrupt caused by external interrupt handler for selected task.

- DETASK (PIB DAT Flag = X'10') - The selected task belongs to a deactivated partition. Deactivation of this task was delayed. The DETASK routine deactivates this task.

After execution, the PIB DAT Flag for the corresponding routine is reset to zero and return is made either

- directly to the interrupted SVC routine (SVRETURN and INITSVC), or

- to task selection (MOVECCB, EXTRETRN, and DETASK).

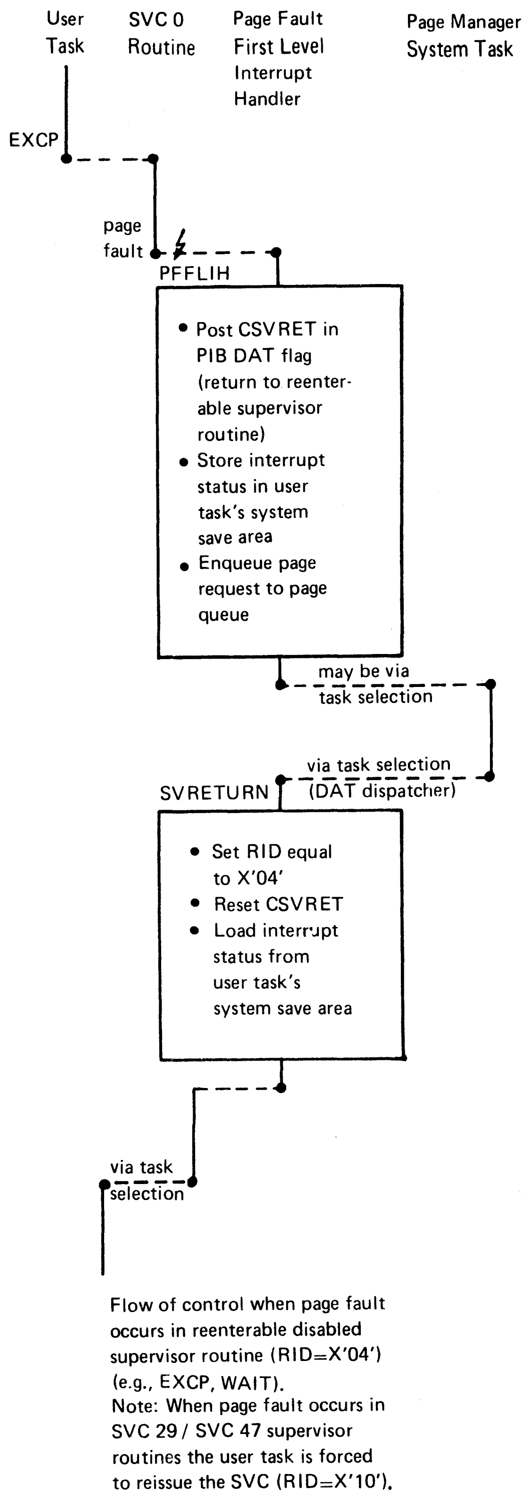


Figure 3.6. Examples of Dispatching by DAT Dispatcher (Part 1 of 3)

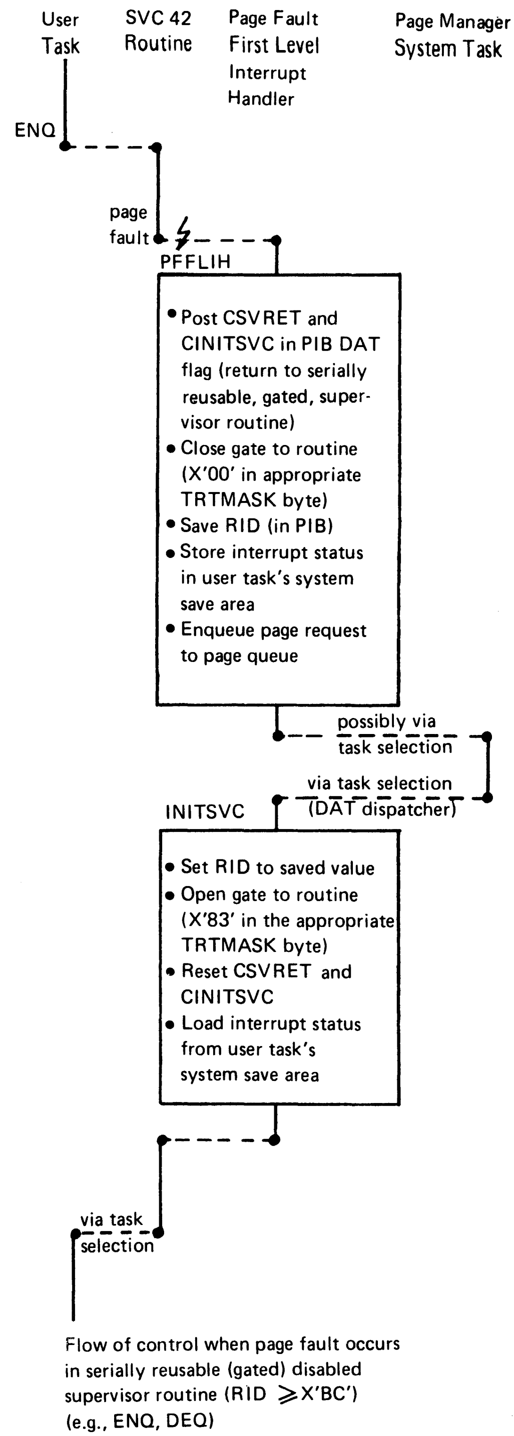


Figure 3.6. Examples of Dispatching by DAT Dispatcher (Part 2 of 3)

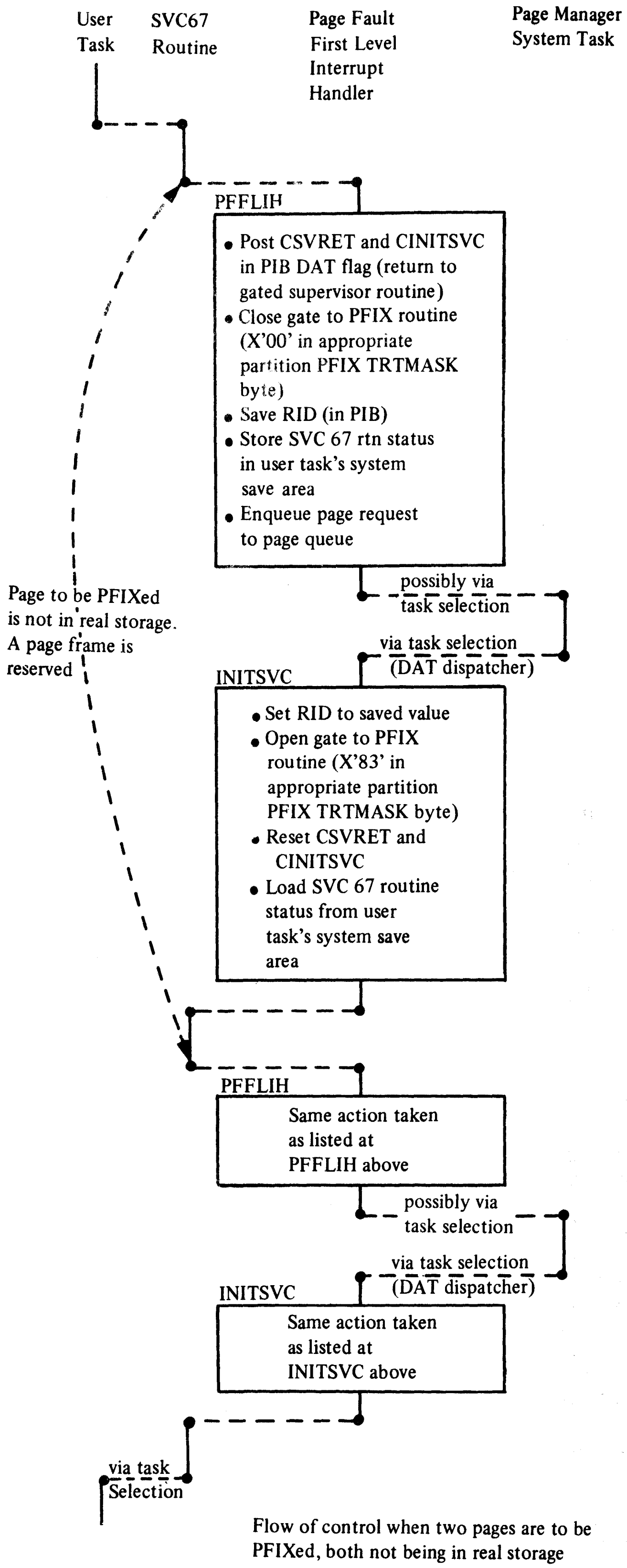


Figure 3.6. Examples of Despatching by DAT Dispatcher (Part 3 of 3)

ASYNCHRONOUS PROCESSING (AP)

If AP=YES was specified at supervisor generation time, the ATTACH macro is supported. When a maintask issues the ATTACH macro, provided a program information block (PIB) is available and the subtask routine and save area are in the same partition as the maintask, the supervisor builds a subtask PIB and PIB extension. This is achieved by copying the maintask PIB and PIB extension and then moving the subtask save area address specified by the ATTACH macro into the subtask PIB. If the ATTACH macro also specified a termination event control block (ECB) address in the same partition, this address is moved to the subtask PIB extension.

The registers in the maintask save area are copied into the subtask save area specified by the ATTACH macro, and the TIK of the subtask is moved to the subtask save area.

In the maintask save area, register 0 is pointed to the first byte beyond the subtask save area, and register 1 is made positive.

In the subtask save area, register 1 is pointed to the maintask save area.

The subtask entry point address is moved to the PSW in the subtask save area and the subtask key (TIK) is inserted in the MVCFLD (See Figure 3.1) immediately before the maintask partition key.

TIDTRT (16 bytes)

00	00	00	00	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F

Initially 12 Program Information Blocks Available for Subtasks in a 3-partition system.

Figure 3.7. TIDTRT Field

If 4 partitions, byte 4 is zero and there are initially 11 subtask PIBs available; if 5 partitions, byte 5 is also zero and there are 10 PIBs; if only 2 partitions, byte 3 is X'03' and there are 13 subtask PIBs initially available.

The maintask PIB is flagged for subtask attached.

To indicate whether a PIB is available for a subtask that is to be attached, a 16-byte field called TIDTRT (see Figure 3.7) is used. Each byte of this field is related to a PIB. A nonzero value in a byte indicates that the corresponding PIB is available. The bytes of TIDTRT corresponding to partition PIBs are always zero. These PIBs are obviously not available for subtasks. When a PIB is wanted for a subtask, the four low-order bits of the first available (nonzero) byte in the table are used (multiplied by 16) to create the TIK for the subtask. The TIK is equal to the displacement of the PIB from the start of the PIB Table (PIBTAB). That byte in the table is then made zero to indicate that the PIB is in use.

If the subtask cannot be attached because no PIB is available, the supervisor sets the event bit off in a supervisor event control block (SPVECB), puts the address of this ECB in register 1 in the maintask save area, makes this register negative, and returns to task selection.

A maintask that tests register 1 and finds it negative can wait on this register until a PIB becomes available through a subtask being detached in another partition.

The DOS/VS Supervisor is designed to operate in an IBM System/370 CPU in the Extended Control (EC) mode, which is determined by the Program Status Word bit 12 being set ON. The format of the PSW (see Figure 6.22) differs from that used in IBM System/360.

When processing is interrupted as a result of one of five possible conditions:

End of an I/O operation,
Program Check,
Machine Check,
Supervisor Call, or
External Interrupt,

in addition to the old Program Status Word, information is stored in an area of low real storage at IOINF, PGMINF, MCIC, SVCINF, or EXTINF, respectively.

The applicable new PSW is loaded and the supervisor routines are entered at ENTIO, ENTPCK, MACHEK, ENTSVC, or ENTEXT, respectively.

In most cases, the General Entry Routine is called. This routine saves the interrupt information from low real storage in the PIB extension (PIB2) of the interrupted task, if the interrupted task is a user task. The registers of the interrupted task and the old PSW are saved in the appropriate system task save area (if system task interrupted) or in the appropriate partition or subtask save area (user main- or subtask interrupted).

The normal exit from the supervisor is through the General Exit Routine which is entered at EXIT. System tasks have priority and any system task that is ready to run will be dispatched before any user task can be selected. If the task that was interrupted was a system task, and no higher priority task is dispatchable, the entry registers are restored and the old PSW is reloaded to continue processing without checking for any other conditions.

If no system task is selectable, then the highest priority user task is dispatched. In a multiprogramming environment foreground programs have priority over background programs unless the order of priority has been changed by the PRTY command or the PRTY parameter of the FOPT macro. If the supervisor supports asynchronous processing, the first subtask attached has priority over all other tasks in the same partition until it is detached. A main task has a lower priority than any of its subtasks.

I/O INTERRUPT

Microprogramming detects an I/O interrupt and loads the I/O new PSW. Refer to the section "I/O Interrupt Handler" in "Physical Input/Output System" and Charts 06.1 and 06.2.

PROGRAM CHECK INTERRUPT

When a program check is encountered, microprogramming loads the program check new PSW (which has the DAT bit turned off, that is, DAT is deactivated).

The interrupt code is first checked to see if the program check is a page translation exception (page fault). If the program check is a page fault, control is passed immediately to the page fault handler (PFFLIH).

If the program check is not a page fault, it is next checked to see if it is a translation specification exception, that is, if DAT is defective. If this is the case, the system is immediately put into a hard wait state.

If neither of these are the case, DAT is reactivated and normal program check handling continues with segment translation exceptions being changed to addressing exceptions.

If the program check occurs in the supervisor state, two tests are made:

1. A list of addresses in the supervisor is scanned to see if the program check occurred at any of these addresses. If so, the user program is canceled.
2. A check is made to see if VTAM was active and executing a VTAM SVC (49 or 53) or a VTAM appendage. If so, the VTAM partition is canceled.

Otherwise, the system goes into a hard wait (Register 11 contains X'FFFF').

If the supervisor was generated with the FOPT macro parameter PC=YES, the routine entered at PCROUT saves the interrupt status information and general registers in the save area specified by the user's STXIT (PC) macro for the following purposes:

1. to restore for continuation.

2. to enable the user's PC routine to analyze the status.

3. to facilitate analysis of a dump (the dump then contains all interruption information).

To enter the user's PC routine, the PSW is set up in the current save area (SVEPSW) and the user's routine address in the PC table is made negative to indicate that the routine is in use; if another program check condition is subsequently detected, the job is then canceled.

The interrupt status information and general registers are restored to the program save area, and the user's PC routine address is made positive by SVC 17.

For the format of the PC option table, refer to Figure 4.1, and for the format of the user's save area to Figure 4.9.

PAGE FAULT HANDLER

Page faults are a special type of program check and are handled by an extension to the program check handler called the page fault first level interrupt handler (PFFLIH). A page fault is first checked to see if it was caused during the processing of a MICR stacker select routine. If this is the case, the task is canceled.

If the page fault is caused by an enabled user task or by the fetch task, the PSW and registers are saved in the appropriate save areas.

If the page fault is caused by other than one of these two types of tasks, the RID (Routine IDentifier) is tested to see which type processing is to be carried out before the request is queued. Figure 4.2 shows the various routine identifiers along with the action taken when one of these routines causes a page fault. (The RID may also indicate that a partition issued a PFIX request for a page which is not in real storage, or that the SVC 58 routine issued a GETREAL request.)

The page fault handler also sets the PIB DAT Flag. This flag tells the dispatcher how to dispatch the task after the page fault has been handled. The PIB DAT Flag indicates that control is to be passed to INITSVC if the page fault is handled for a gated SVC or to SVRETURN if a reenterable SVC is to be reactivated.

If no page fault appendage is provided for the interrupted task, the page fault request is queued for handling by the page management routines and the interrupted task is made not dispatchable.

If an appendage is present for the task, control is passed to the appendage, and the task causing the page fault remains dispatchable unless the page fault occurred in a supervisor service of the task. (Refer to the DOS/VS Supervisor and I/O Macros manual for a more detailed explanation of Page Fault appendages.)

Whether a request is queued or not depends upon whether the task owning the appendage has a page fault request pending. If a page fault is already in the queue for the task, the new page fault is not queued. If, however, there is no request pending for the task, the new page fault request is queued.

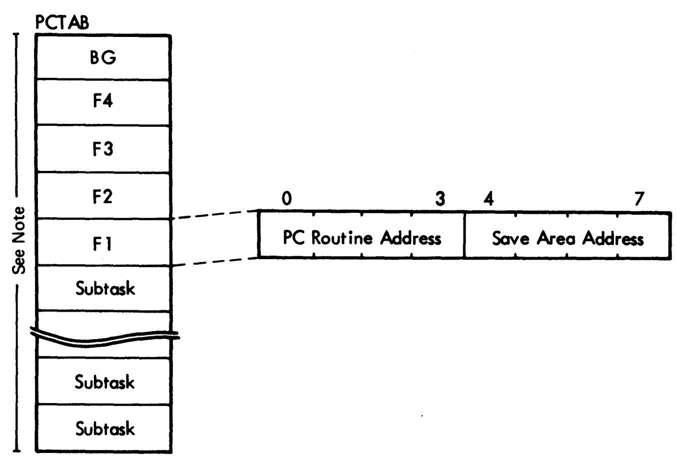


Figure 4.1. Program Check Option Table

Bytes:

0-3: No STXIT issued: zero

STXIT issued: Address of the user Program Check Routine

STXIT issued and the user routine is already in use: Complement of user Program Check Routine address

4-7: No STXIT issued: zero

STXIT issued: Address of the user Save Area

Note: In a supervisor without multiprogramming support, there is only one entry (BG) in each generated table. With multiprogramming support, there is one entry for each partition supported.

With asynchronous processing support, each generated table always comprises 15 entries; the subtask entries occupy the higher address locations in the table.

Bytes 100-101 (X'64' - X'65') of the partition communication region contain the address of the PC Option Table. Label PCTAB identifies the first byte of the table.

NAME	ID	MEANING	ACTION
SYSTEMID	00	System error condition, for example, page fault in I/O interrupt handler.	Hard Wait.
REENTRID	04	Page fault or GETREAL request in a reenterable routine.	Save PSW and REGs to user task's system save area, set PIB DAT Flag to call SVRETURN, and ENQU request.
USERTID	08	Page fault from a disabled user task or disabled B-transient.	Cancel user task error code X'15'.
APPENDID	0C	Page fault in I/O appendage routine.	Cancel user task error code X'36'.
RESVCID	10	Page fault in SVC 29 or 47.	Set saved PSW to reissue SVC when task is dispatched, ENQU page fault request.
DISPID	14	Page fault in a routine which requires no information to be saved, for example, a page fault in the dispatcher.	ENQU page fault request.
PFARID	16	Page fault in a page fault appendage routine.	Cancel user task error code X'15'.
	47 through 4F	Request for a page which is to be PFIXed. RID identifies the partition which issued the PFIX request. Used only if AP=YES.	Save PSW and registers to user task's system save area, set PIB DAT Flag to call INITSVC when task is dispatched next.
G41BND	53	Page fault in the reusable SVC 41 or 42.	Close gate to routine (routine cannot be used until gate is opened). ENQU page request. (Any task trying to use a gated resource is placed in a wait state and marked resource bound. It is released from the wait state when the resource is ungated after the page request has been handled.)
G58BND	71	GETREAL request from the reusable SVC 58.	

Figure 4.2. Routine Identifiers (RID) as Used by the Page Fault Handler (PFFLIH) in Handling Page Requests

EXTERNAL INTERRUPT

Microprogramming detects an external interrupt and loads the external new PSW. External interrupts can be caused by:

- Timer
- External interrupt key
- Signal

Refer to External Interrupt Processor on Chart 07.1.

MACHINE CHECK INTERRUPT

Microprogramming detects a machine check interrupt and loads the machine check new PSW. The resident Machine Check Handler analyzes the Machine Check Interruption Code (MCIC) at location X'E8' and tests the Problem State bit (Old PSW bit 15). The action taken depends on the conditions detected, as follows:

System Termination Condition (hard wait): Control is passed to the R-Transient $$RAST00 to attempt to record the error on the Recorder File (IJSYSRC) and to log an appropriate message before terminating the system.

Task Cancelation Condition: The affected task is canceled and the RAS System Task is activated before branching to Task Selection.

Recoverable Error Condition (soft machine check): The RAS System Task is activated. If the CPU was in the Problem State or if a System Task was active, a branch is taken to Task Selection. If a Supervisor Function was interrupted, the Machine Check Old PSW is reloaded to continue processing from the point of interruption.

SUPERVISOR CALL INTERRUPT (SVC)

SVC is detected by microprogramming, which loads the SVC new PSW. The SVC interruption processor (See Chart 05) analyzes the SVC code placed in low main storage by the microprogramming. Control is transferred to the appropriate processing routine. Some SVCs are optional and cause a cancel if the supervisor was generated without the option.

Return from SVC routines that free a resource (for which another task may be waiting) or that put the issuing task in a wait state, must be through task selection. For all other SVC routines return may be to the same task or through task selection (which will result in selection of the same task as it has the highest priority).

SVC 0: Execute the channel program (EXCP). The address of the user's Command Control Block (CCB) must be supplied in general register 1 before this SVC is issued. If POWER/VS is supported and the I/O operation for the device need to be emulated, control is given to the SVC 0 appendage in the POWER/VS nucleus code.

SVC 1: Fetches a phase. A fetch loads a phase from the System Core Image Library or a Private Core Image Library (PCIL) and branches to the entry address in that phase. If the phase is found to be in the SVA, it is not loaded. The directory entry may be found in storage, in the PCIL directory (if a PCIL is assigned), or in the SCIL directory. For a more detailed description, refer to section "Fetch Routine".

The load and entry addresses are obtained from the directory entry for the phase being fetched. These addresses are relocated if the phase is relocatable. The storage address of the phase name or the address of the parameter list (see "Fetch Routine") must be supplied in general register 1 before this SVC is issued. The entry and load addresses are relocated for a relocatable phase. The entry address can be overridden by a user-supplied entry address in general register 0.

SVC 2: Fetches a Logical Transient. Loads a Logical Transient program name (phase name prefix is $$B) from the system core image library or a Private Core Image Library (PCIL) into the Logical Transient Area (LTA) and enters the logical transient at its load address plus 8 bytes. The directory entry for the phase may be found in storage, in the SCIL directory, or in the PCIL directory (if a PCIL is assigned). For a more detailed description, refer to the section "Fetch Routine".

The storage address of the logical transient phase name or the address of a parameter list (refer to the section "Fetch Routine") must be supplied in general register 1 before this SVC is issued.

The logical transient is loaded at the origin of the Logical Transient Area (LTA) and this address is put into general register 15 which may then be used by the transient as a base register. The supervisor ignores the contents of general register 0 which may be used to pass information to the logical transient.

SVC Dec	SVC Hex	Macro Supported	Function
*Optional			
0	0	EXCP	Execute Channel Program
1	1	FETCH	Fetch any phase
2	2		Fetch a logical transient (B-transient)
3	3		Force dequeue
4	4	LOAD	Load any phase
5	5	MVCOM	Modify supervisor communication region (if issued by MVCOM macro) Fetch another physical transient (if issued by a physical transient)
6	6	CANCEL	Cancel a problem program or task
7	7	WAIT	Wait for a CCB or TECB
8	8		Transfer control to the problem program from a logical transient (B-transient)
9	9	LBRET	Return to a logical transient (B-transient) from the problem program after an SVC 8
10*	A	SETIME	Set timer interval
11	B		Return from a logical transient (B-transient)
12	C		Reset switches in partition communications region
13	D		Set switches in partition communications region
14	E	EOJ	Terminate job and go to job control for end of job step
15	F	SYSIO	Headqueue and execute channel program
16*	10	STXIT(PC)	Provide supervisor with linkage to user's PC routine for program check interrupts
17*	11	EXIT(PC)	Return from user's PC routine
18*	12	STXIT(IT)	Provide supervisor with linkage to user's IT routine for interval timer interrupt
19*	13	EXIT(IT)	Return from user's IT routine
20*	14	STXIT(OC)	Provide supervisor with linkage to user's OC routine for external or attention interrupts (operator command)
21*	15	EXIT(OC)	Return from user's OC routine
22	16	SEIZE	Seize/release system; enable/disable for external and I/O interrupts; set key in user's PSW
23*	17		Load phase header. Phase load address is stored at user's address
24*	18	SETIME	Set timer interval and provide supervisor with linkage to user's TECB, if any

Figure 4.3. Supervisor Calls (Part 1 of 4)

SVC Dec	SVC Hex	Macro Supported	Function
*Optional			
25*	19		Issue HALT I/O on a teleprocessing device, or HALT I/O ON ANY DEVICE IF ISSUED BY OLTEP. With multiprogramming, dequeue an unstarted OLTEP I/O request to a shared device
26*	1A		Validate address limits
27*	1B		Special HIO on teleprocessing devices
28*	1C	EXIT(MR)	Return from user's stacker select routine (MICR type devices only)
29*	1D	WAITM	Provide support for multiple wait macro WAITM
30*	1E	QWAIT	Wait for a QTAM element
31*	1F	QPOST	Post a QTAM element
32	20		Reserved
33	21		Reserved for internal macro COMRG
34	22	GETIME	Provides Time-of-Day and updates the DATE field
35*	23	HOLD	Hold a track for use by the requesting task only
36*	24	FREE	Free a track held by the task issuing the FREE
37*	25	STXIT(AB)	Provide supervisor with linkage to user's AB routine for abnormal termination of a task
38*	26	ATTACH	Initialize a subtask and establish its priority
39*	27	DETACH	Perform normal termination of a subtask. It includes calling the FREE routine to free any tracks held by the subtask
40*	28	POST	Inform the system of the termination of an event and ready any waiting tasks
41*	29	DEQ	Inform the system that a previously enqueued resource is now available
42*	2A	ENQ	Prevent tasks from simultaneous manipulation of a shared data area (resource)
43	2B		Reserved
44*	2C		Provide supervisor support for external creation of unit check records by specific request
45*	2D		Provide emulator interface
46*	2E		Provide OLTEP with the facility to operate in supervisory state
47*	2F	WAITF	Provide support for multiple wait macro WAITF for MICR type devices
48*	30		Fetch a CRT transient
49	31		Used by VTAM

Figure 4.3. Supervisor Calls (Part 2 of 4)

SVC Dec	SVC Hex	Macro Supported	Function
*Optional			
50	32		Reserved for LIOCS error recovery
51	33		Return phase header
52*	34	TTIMER	Return the remaining time interval, or cancel a time interval
53	35		Used by VTAM
54	36	FREEREAL	Release page frames to selection pool
55	37	GETREAL	Provide interface between SDAID and PDAID initialization routine and page management routine, to create the PDAID alternate area or the SDAID buffer area
56	38		Reserved.
57	39		Reserved.
58	3A		Provide interface between job control and the supervisor. Get real storage for real jobs
59	3B		Provide interface between EOJ and the supervisor. Initialize specified page table entries
60	3C	GETADR	Provide virtual address of location within I/O areas for ERP and CRT routines
61*	3D	GETVIS	Get storage in virtual partition
62*	3E	FREEVIS	Free storage in virtual partition
63	3F	USE	Use a resource
64	40	RELEASE	Release a resource
65*	41	CDLOAD	Load VSAM or CI phase
66	42	RUNMODE	Return mode in which program is running
67*	43	PFIX	Fix page(s) in real storage
68*	44	PFREE	Free page(s) in real storage
69*	45	REALAD	Return real address corresponding to a given virtual address
70*	46	VIRTAD	Return virtual address corresponding to a given real address
71*	47	SETPFA	Establish or terminate the linkage between the supervisor and a user page-fault appendage routine
72*	48	GETCBUF FREECBUF	Get or free copy buffer for IDAL or tape ERP
73*	49	SETAPP	Allow linkage to channel and appendage routines
74*	4A		Fix page(s) in real storage for restart

Figure 4.3. Supervisor Calls (Part 3 of 4)

SVC Dec	SVC Hex	Macro Supported	Function
*Optional			
75*	4B	SECTVAL	Calculate a sector value for a disk device with the RPS feature
76	4C		Initiate recording of an RMSR I/O error
77	4D	TRANSCSW	Returns the virtual address of a copied CCW
78-84			Reserved
85*	55	RELPAG	Release contents of one or more pages
86*	56	FCEPGOUT	Force a page-out for one or more pages
87*	57	PAGEIN	Pagein one or more pages
90*	5A	PUTACCT	Provide interface with POWER/VS for additional account information (by user).
91*	5B		Provide interface with POWER/VS for standard account information (DOS/VS).

Figure 4.3. Supervisor Calls (Part 4 of 4)

Only one program can use the logical transient area at a time. If the area is already occupied when it is wanted by another program, then this latter program becomes SVC 2-bound (first PIB flag byte = X'81') until the LTA is released by the occupying program issuing a SVC 11. (See also Figure 2.8.)

SVC 3: Provides an interface between the supervisor and $$BEOJ5. SVC 3 allows forced dequeueing of channel queue entries that belong to devices assigned to the partition or task that is being canceled or has reached end-of-job. The PUB whose address is in register 1 is dequeued.

SVC 4: Loads a phase from the system core image library or a private core image library (PCIL) and returns to task selection. The directory entry for the phase may be found in storage, in the SCIL directory, or in the PCIL directory (if a PCIL is assigned). For a more detailed description, refer to the section "Fetch Routine".

The storage address of the phase name or the address of a parameter list (refer to the section "Fetch Routine") must be supplied in general register 1 before this SVC is issued. The user may override the link-edited load address by supplying a load address in general register 0. Upon return to the user, general register 1 contains the phase entry address adjusted for any changes in the phase's load address, and general register 0 points to the active In-core Directory Entry if one was supplied, or found by a local or system directory list search.

SVC 5: When issued by a user through the MVCOM macro, modifies the supervisor communication region. It supplies the supervisor support for the MVCOM macro. The sequence of events is:

1. MVCOM macro issues a SVC 5.

2. The resident routine alters the supervisor communication region as specified by the parameters of the MVCOM macro.

When a physical transient issues a SVC 5, another physical transient program (phase name prefix $$A) is loaded from the System Core Image Library into the Physical Transient Area (PTA), and is entered at its load address plus 10 bytes.

The calling transient sets up the physical transient name in the error block. The storage address of the physical transient phase name is loaded in general register 1 before the fetch is made.

The physical transient is loaded at the origin of the Physical Transient Area (PTA) and this address is put into general register 11 which may then be used by the transient as a base register.

SVC 6: Cancels a program (task) or partition. This is usually achieved by the requesting program, task, or subtask issuing a CANCEL or CANCEL ALL macro.

If a subtask issues CANCEL, only that subtask is terminated. If a maintask issues CANCEL, or a subtask issues CANCEL ALL, then the entire partition is canceled, the maintask being always the last to be terminated.

CANCEL macro issued by maintask without subtasks: the issuing task is terminated normally.

- Cancel code X'23' is posted to the issuer's PIB.
- Message '(issuer*) CANCELED DUE TO PROGRAM REQUEST'.

CANCEL macro issued by subtask: the issuing subtask is terminated normally.

- Cancel code X'23' is posted to the issuer's PIB.
- Message '(issuer*) CANCELED DUE TO PROGRAM REQUEST'.

CANCEL macro issued by maintask with subtasks attached: the maintask is terminated normally; attached subtasks are terminated abnormally.

- Cancel code X'1D' is posted to each subtask PIB.
- Cancel code X'17' or X'18' is posted to the maintask PIB.
- Message '(subtask*) CANCELED DUE TO MAINTASK TERMINATION'.
- Message '(issuer*) CANCELED DUE TO PROGRAM REQUEST'.
- A dump is generated at termination of the maintask if the cancel code in the maintask PIB is X'18'.

CANCEL ALL macro issued by a subtask: the issuing subtask is terminated normally; other subtasks and the maintask are terminated abnormally.

- Cancel code X'23' is posted to the issuing subtask PIB.

----

*First 8 bytes of task or subtask save area.

- Cancel code X'1C' is posted to each of the other subtasks PIBs and to the maintask PIB.
- Message '(issuer*) CANCELED DUE TO PROGRAM REQUEST'.
- Message '(main or subtask*) CANCELED DUE TO CANCEL ALL MACRO'.
- A dump is generated at termination of the maintask.

If the FOPT macro parameter AB=YES was included when the supervisor was generated and linkages to the user's AB routines have been established through the STXIT (AB) macro, these routines are entered for all tasks that are terminated abnormally by the CANCEL or CANCEL ALL macro, so the task that issues a SVC 6 never enters its AB routine.

An AB routine normally terminates through a DETACH, EOJ, or CANCEL macro, but an abnormal condition encountered in an AB routine also terminates that AB routine.

SVC 7: Waits for the completion of an I/O operation or for a timer interruption to occur. It supplies the supervisory support for the WAIT macro.

If the traffic bit (CCB) or event bit (TECB) has been posted, SVC 7 branches directly to task selection.

If the traffic bit or event bit has not been posted the following action is taken:

- If a system task was interrupted, the system task is deactivated and its PIB flag is set to I/O-bound.
- If a user task was interrupted, the PIB of the interrupted program is set to SVC 7-bound (not ready to run).

If a page fault occurs during execution of a SVC 7, the SVC 7 is reissued after the page fault has been handled.

SVC 8: Supplies the supervisory support to temporarily return from a logical transient to the problem program. This SVC may be issued only from the logical transient area (LTA) and does not free this area. The entry address to the problem program must be specified in general register 14. The task selection exit loads the problem program registers. General registers 0 and 1 are passed to the problem program.

To return to the logical transient, the problem program issues a SVC 9.

SVC 9: Supplies the supervisory support to return to the logical transient after a SVC 8 has been issued. A SVC 9 may be issued only by the problem program. The task selection exit loads the logical transient registers. General registers 0 and 1 are passed to the logical transient program.

SVC 10: Sets a timer interval. This SVC is optional, and the issuing program is canceled if the supervisor was generated without the IT option for the partition or task. Only the timer supported program can issue an SVC 10. Others are canceled.

SVC 10 uses the same routines as SVC 24 to set the timer interval (see SVC 24) but does not store the address of the user's TECB.

If the supervisor provides interval timer support for multiple partitions, the code for SVC 10 is not generated. The address of the SVC 24A routine is then placed in the SVC table (SVCTAB) instead of the address of the SVC 10 routine.

SVC 11: Returns from a B-transient releasing the B-transient area. SVC 11 is invalid if issued by other than a B-transient. The logical transient area is released for use by other programs or tasks. Return is to the highest priority program ready to run.

SVC 12: The function of this SVC depends on the contents of general register 1.

If this register contains a zero value, bit 5 (PCIL being condensed) in byte 59 of the partition communications region is set to zero.

If general register 1 contains a nonzero value, the function depends on bit 8 of this register.

If this bit is zero, this SVC supplies the supervisory support to reset flags in the linkage control byte (displacement 57 in the partition communications region). The user has provided the address of a mask (1 byte) in general register 1. This mask is ANDed with the linkage control byte. If bit 6 of the linkage control byte is turned off by this SVC, the Fetch routine is dequeued (PTA or IDRA posted available). If bit 1 of the linkage control byte is turned off by this SVC, the key in the user's PSW is reset (it was set to zero when this bit was turned on, see SVC 13).

If bit 8 of general register 1 is one, this SVC supplies the supervisory support to reset flags in a specified byte of the partition communications region. The user has provided a displacement in byte 2 and a mask in byte 3 of general register 1. The mask is ANDed with the byte at the specified displacement in the partition communications region.

SVC 13: The function of this SVC depends on the contents of general register 1.

If this register contains a zero value, bit 5 (PCIL being condensed) in byte 59 of the partition communications region is set to one.

If general register 1 contains a nonzero value, the function depends on bit 8 of this register.

If this bit is zero, this SVC supplies the supervisory support to set flags in the linkage control byte (displacement 57 in the partition communications region). The user has provided the address of a mask (1 byte) in general register 1. This mask is ORed with the linkage control byte. If bit 1 of the linkage control byte is posted the key in the user's PSW is set to zero.

If bit 8 of general register 1 is one, this SVC supplies the supervisory support to set flags in a specified byte of the partition communications region. The user has provided a displacement in byte 2 and a mask in byte 3 of general register 1. The mask is ORed with the byte at the specified displacement in the partition communications region.

SVC 14: This is the normal end of job (EOJ). Cancel code X'10' is posted to the PIB for the program issuing the SVC 14. Refer to Figure 2.7 for the format of the PIB tables and to Chart 03 for the General Cancel Routine. The next time the canceled program is selected on general exit, a branch is made to the SVC 2 routine to fetch the cancel B-transient program $$BEOJ3 if teleprocessing is supported, or $$BEOJ4 if teleprocessing is not supported. Job control is loaded by $$BEOJ to perform the End-of-Job step.

SVC 15: Execute the channel program before all other I/O requests queued for the same device (headqueueing). The address of the user's Command Control Block (CCB) must be supplied in general register 1 before this SVC is issued. The high order bit (bit 0) of this register is used to indicate whether the request is conditional (off) or unconditional (on). A conditional request is delayed in certain cases (see below). This SVC can be used by system tasks only. A user task issuing this SVC is canceled.

The PUB is copied to the Headqueue PUB table (HQPUBTAB) Entry (see Figure 4.4) for the requesting system task. This copied PUB is initialized to allow the I/O request to be queued to it immediately by the Channel Scheduler. Headqueue in process is posted in the original PUB, and register 3 is pointed to the copied PUB instead of the original. A branch is then taken to the Channel Scheduler to schedule the I/O.

If on entry of the SVC 15 routine nc channel queue entry is available, the task is posted channel queue entry bound (X'79'), deactivated, and set up to reissue the SVC when selected next time.

If headqueue in progress was already posted in the PUB, the action depends on the type of the new headqueue request. If the new request is conditional, headqueue wanted is posted in the HQPUBTAB entry, to wait for the pending request to finish. If the new request is unconditional, it is not delayed and is serviced immediately. If the pending headqueue request has a higher priority, its HQPUBTAB entry is posted for headqueue in progress for later reactivation. The priority is not the same as the dispatching priority of system tasks and is determined by the order of the entries in the Headqueue PUB Table.

A conditional headqueue request is also delayed when the device is busy, with a nonerror request queued to its PUB. The headqueue wanted bit is posted then in both the copied and the original PUB.

It should be noted that when in a PUB or copied PUB both of the headqueue switches are on, no I/O is started for the PUB. However, in this situation there is one associated copied PUB with these switches off and I/O is started on this copied PUB.

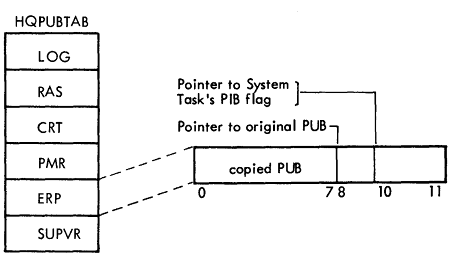


Figure 4.4. Headqueue PUB Table

The Headqueue PUB Table has a maximum of six entries (for RAS, CRT, PMR, ERP, and SUPVR system tasks). The entries for the RAS and CRT tasks are not generated when these functions are not supported. The LOG headqueue entry is only generated for a model 115/125.

The order of the entries determines the headqueueing priority of the system tasks. The RAS system task has the highest priority.

Bytes 20-23 (X'14' - X'17') of the RASLINK area contain the address of the Headqueue PUB Table. Label HQPUBTAB identifies the first byte of the table.

SVC 16, 18, and 20: These supervisor calls provide supervisory support for the STXIT macro. They are optional, and the issuing program is canceled if the supervisor was generated without the applicable option.

For each of these STXIT SVCs, the address of the user routine is specified in general register 0 and the address of the user save area is specified in general register 1. An invalid save area address causes the problem program to be canceled (ERR25).

SVC 16 stores the address of the user's program check (PC) routine and save area address in the PC option table.

SVC 18 stores the address of the user's interval timer (IT) routine and save area address in the IT option table and sets a pointer to the correct entry in the table. This SVC may be issued only by the timer-supported program. If multitasking, only the maintask can process the interruption unless the supervisor was generated with full timer support (IT=YES).

SVC 20 stores the address of the user's operator-communication (OC) routine and save area address in the OC option table. If multitasking, only the maintask can process the interruption.

For the format of the option tables, refer to Figures 4.1 and 4.5 through 4.7, and for the format of the save area to Figure 4.9.

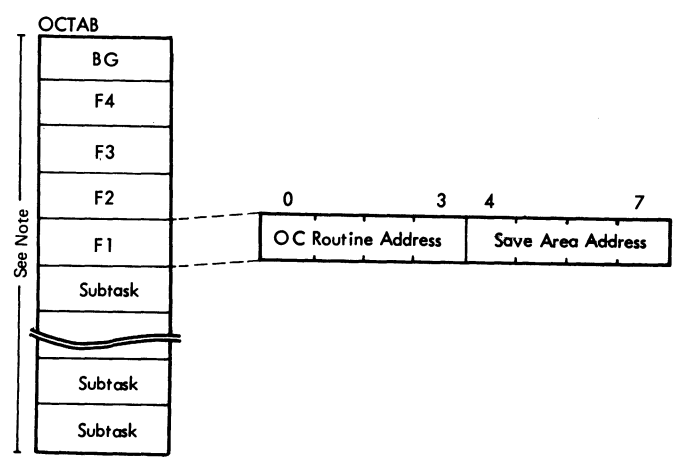


Figure 4.5. Operator Communication Option Table

Bytes:

0-3: No STXIT issued: zero

STXIT issued: Address of the user Operator Communication Routine
STXIT issued and the user routine is already in use: Complement of the user Operator Communication Routine address

4-7: No STXIT issued: zero

STXIT issued: Address of the user Save Area

Note: In a supervisor without multiprogramming support, there is only one entry (BG) in each generated table.

With multiprogramming support, there is one entry for each partition supported. With asynchronous processing support, each generated table always comprises 15 entries; the subtask entries occupy the higher address locations in the table.

Bytes 104-105 (X'68' - X'69') of the partition communication region contain the address of the OC Option Table. Label OCTAB identifies the first byte of the table.

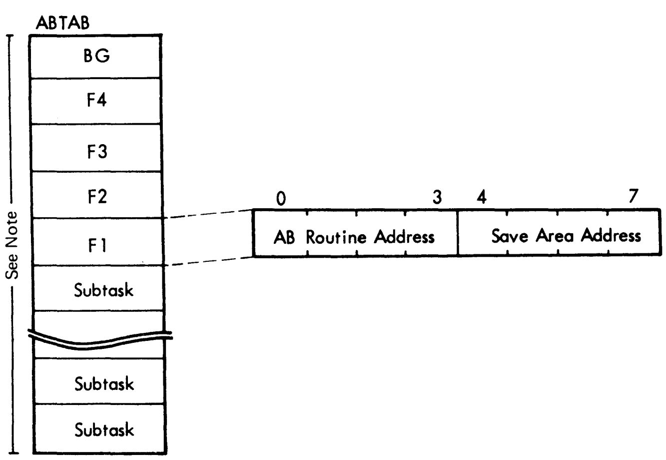


Figure 4.6. Abnormal Termination Option Table

Bytes:

0-3: No STXIT issued: zero

STXIT issued and RTNADDR parameter passed: Address of entry point of user's abnormal termination routine. If AP (Asynchronous Processing) is supported, the maintask and subtasks may have the same or different AB routines. When a subtask is ATTACHed after a STXIT AB macro has been issued by the maintask, the subtask will receive the AB routine address specified by the maintask only if the ATTACH macro for that subtask has the ABSAVE parameter specified. The subtask can override this by issuing its own STXIT AB macro.

4-7: No STXIT issued or no save area parameter passed: zero

STXIT issued and save area parameter passed: Address of a 72-byte save area used by the supervisor to store the interrupt status information and the contents of the general registers.

Note: One table entry is generated for each partition supported. With asynchronous processing support, the table always comprises 15 entries; the subtask entries occupy the higher address locations in the table.

Bytes 84-87 (X'54' - X'57') of the System Communication Region (SYSCOM) contain the address of the AB Option Table. Label ABTAB identifies the first byte of the table.

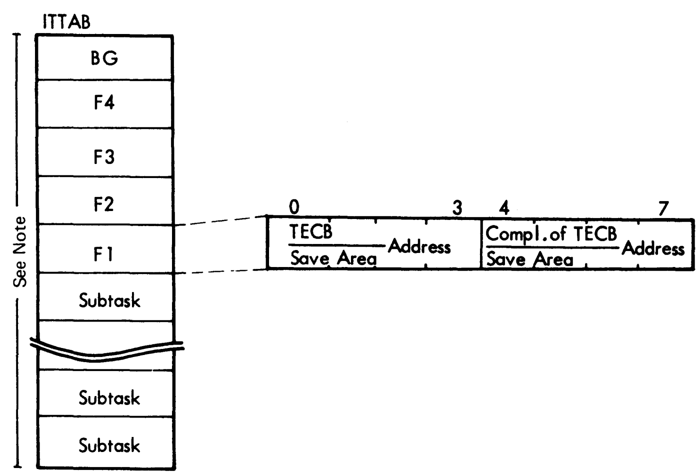


Figure 4.7. Interval Timer Option Table

Bytes:

0-3: No TECB or STXIT issued: zero

TECB issued: Address of the timer event control block
STXIT issued: Address of user interval timer routine
STXIT issued and user routine is already in use: Complement of the user interval timer routine

4-7: No TECB or STXIT issued: zero

TECB issued: Complement of the TECB address
STXIT issued: Address of the user save area

Note: One table entry is built for each partition (multiple timer support) and an IT Request table is also built.

With multiple timer and asynchronous processing supported, the table always comprises 15 entries; the subtask entries occupy the higher address locations in the table.

Bytes 102-103 (X'66' - X'67') of the partition communication region contain the address of the IT Option Table. Label ITTAB identifies the first byte of the table.

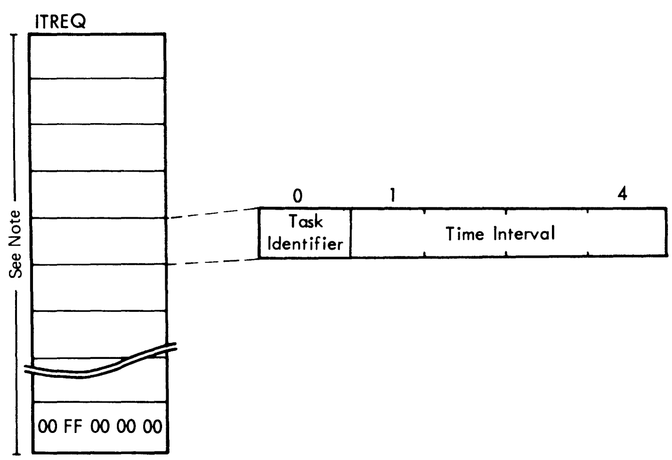


Figure 4.8. Interval Timer Request Table

Bytes:

0: No SETIME issued: X'00', the last entry is always X'00'

SETIME issued: PIK or TIK (X'10' - X'F0') of the program or task that issued the SETIME macro

1-4: No SETIME issued: Highest time interval possible (X'FF000000'). The last entry is always X'FF000000'

SETIME issued: Time interval that will elapse between the occurrence of the currently pending interrupt in SYSTIMER and the occurrence of the interrupt itself. The entries are in ascending order of magnitude, that is the smallest interval appears first.

The interval corresponding to the currently pending interrupt is in SYSTIMER (address X'50') and the key of the respective program or task is in TWTIMS

Note: This table is generated only for multiple timer support (IT=YES). The number of entries is one more than the number of partitions supported, but with multiple timer and asynchronous processing supported, the table always comprises 16 entries.

Bytes 80-83 (X'50' - X'53') of the System Communication Region (SYSCOM) contain the address of the IT Request Table. Label ITREQ identifies the first byte of the table.

Interrupt Status Information					
0	1	2	4	5	8 ... 71
Unused	Protection Key and Master from PSW Byte 1	Interruption Code	*	Instruction Address	General Registers 0-15

* Byte 4 bits 0-1 Instruction Length Code
2-3 Condition Code
4-7 Program Mask

The address of the save area specified by the user in the STXIT macro parameter is stored in the option table.

Figure 4.9. Format of the User's Save Area for AB, PC, OC, and IT Routines

SVC 17, 19, and 21: These supervisor calls provide supervisory support for the EXIT macro. They are optional, and the issuing program is canceled if the supervisor was generated without the applicable option.

SVC 17 provides a return from the user's PC routine to the next sequential instruction in the program that was interrupted due to a program check.

The user-supplied save area is restored to the problem program save area. Refer also to the section also "Program Check Interrupt".

SVC 19 provides a return from the user's IT routine to the timer-supported program. This SVC may be issued only by the timer-supported program.

SVC 21 provides a return from the user's OC routine to the program that was interrupted by the external interrupt key (background) or by the attention routine MSG command (foreground).

For the format of the option tables, refer to Figures 4.1 and 4.5 through 4.7, and for the format of the save area to Figure 4.9.

SVC 22: Seizes the system and provides a release from such a seizure in a multiprogramming system. This SVC may be issued only by job control and logical transient programs. The PSW protection field must be zero, otherwise the issuing program is canceled.

The first SVC 22 issued seizes the system. Until the next SVC 22 is issued, the task selection mechanism is disabled so that the issuing task is the only one that can be selected. The next SVC reenables the task selection mechanism.

If the low-order byte of general register 0 is zero, the system mask is set to disable I/O and external interrupts; if nonzero, the system mask is set to enable I/O and external interrupts.

If general register 0 is negative, the user protection key is set in the user's PSW.

If the supervisor was generated with NPARTS=1, SVC 22 provides only for enabling/disabling and for setting the key in user's PSW.

SVC 23: Retrieves the load address for a specified phase from the directory entry for the phase. The program issuing an SVC 23 is canceled if supervisor was generated with NPARTS=1 or if the PSW protection key does not equal 0. (Only job control and B-transient programs can issue an SVC 23.)

The user must specify the storage address of the phase name in general register 1 and the address where the load address is to be stored in general register 0. The main fetch subroutine scans the System Directory List, the System Core Image Library and the Private Core Image Library (if supported and assigned) for a directory entry for the phase. Refer to the section "Fetch Routine Logic" for a more detailed description of the scan. The load address (3 bytes) is retrieved and stored at the address specified by general register 0. If the phase is relocatable and the supervisor supports relocating load, the load address returned is the relocated load address.

SVC 24: Stores the address of the user's timer event control block (TECB) and sets a timer interval. This SVC is optional, and the issuing program is canceled if the supervisor was generated without IT option for the partition or task. ~~Only the timer supported program can issue an SVC 24. Others are canceled.~~

The address of the user's TECB is specified in general register 0, and the time interval is specified in general register 1.

The event bit is reset in the TECB, and the TECB address is stored in the IT option table (ITTAB). See Figure 4.7 for the format of the IT option table.

Note: The event bit is set when a timer interrupt occurs.

The interval, specified by the user in general register 1, is either set in SYSTIMER (X'50') or an adjusted value is entered in the Interval Timer Request (ITREQ) table (see Figure 4.8), as follows:

If the supervisor supports the multiple timer (ITREQ table generated by the FOPT macro parameter IT=YES specified with global BG20 on):

a. if no timer interrupt is pending for another task (SYSTIMER = X'FF000000'), or if the current pending timer interrupt is for this same task (PIK or TIK = TWTIMS), the interval value in general register 1 is set in SYSTIMER. If job accounting was specified at system generation time (JA=YES), SYSTOD (X'54') is updated to the time of day at which the interrupt is to occur.

b. if this interrupt is to occur earlier than the current pending interrupt for another task, the difference between the old and the new value of SYSTIMER is added to each interval in ITREQ and the intervals are pushed down one entry. The interval (difference between old and new value of SYSTIMER) of the previously pending interrupt and the PIK or TIK of the task owning the interrupt are inserted in the first entry in ITREQ. In addition, the PIK or TIK of the task owning the new interrupt is inserted in TWTIMS.

c. if this interrupt is to occur later than the current pending interrupt for another task, the value of the interval specified in general register 1 is decremented by the value in SYSTIMER and the result is entered in the ITREQ table (any entries with higher values for other tasks are moved down, but any entry for this same task is deleted).

   In this case, SYSTCD is not updated.

The user causes the program to wait for the timer interrupt to occur by issuing an SVC 7. (See SVC 7 in this list.)

SVC 25: Halts I/O on a teleprocessing device, or halts I/O on any device if issued by OLTEP. If the supervisor is generated without teleprocessing and without OLTEP, any issuing program is canceled.

The address of any command control block (CCB) containing the symbolic address of the device to be halted must be supplied in register 1 before issuing the SVC 25. If the SVC 25 is used by a program other then OLTEP, an HDV instruction is issued to the device if:

1. It is a teleprocessing device, and

2. An I/O interrupt is pending for the device.

If OLTEP is the issuing program, an HDV instruction is issued to the device if there is I/O pending for the device. If MPS=YES was specified during system generation, OLTEP uses SVC 25 to dequeue an unstarted I/O operation to a shared device. The channel queue entry is removed from the channel queue, and in this case an HDV instruction is not executed.

SVC 26: Validate address limits. The program issuing an SVC 26 is canceled if the PSW protection key does not equal 0. (Only job control and B-transient programs can issue an SVC 26.)

The upper address must be specified in general register 2, and the lower address must be specified in general register 1.

When a CRT routine issues a SVC 26, control is always returned to the CRT routine, even in case of an error. For any other routine, is either address is outside the requestor's partition then the task is canceled (ERR25).

SVC 27: Same as SVC 25, except that SVC 27 cannot be used by OLTEP. It can be used only to halt I/O on a teleprocessing device. The CCB is not dequeued if the CSW has been stored after an HIO command.

SVC 28: Provides return from user's stacker select routine to the MICR external interrupt routine in the supervisor. This SVC is optional and causes a cancel if issued at any point other than in a stacker select routine with MICR devices.

SVC 29: Provides supervisory support for the WAITM macro, if WAITM=YES was specified in the supervisor generation FOPT macro. On entry, general register 1 contains the address of an ECB list.

For MICR type devices, WAITF must be used to identify the multiple wait on the MICR ECB. The WAITF macro is supported by SVC 47.

The ECBs are all checked for the traffic bit. When an ECB is found with the traffic bit posted, the SVC 29 routine returns with the address of the posted ECB in register 1. If all ECBs are checked and no traffic bits are posted, the task is made I/O bound (bit 7 in PIBFLG set to zero) and its PSW is modified, so that the SVC will be reissued when the task is selected again.

SVCs 30, 31, and 32: Reserved for QTAM. Refer to the QTAM Logic Manual.

SVC 33: When issued by a user task, immediate exit is taken to task selection.

When issued by a system task, it results in deactivation of this system task (the select byte in the system communications region is set to zero). The deactivated system task is posted ready to run (X'83'). This procedure allows task selection to give control to any higher priority system task that is ready to run.

SVC 34: Reserved for the GETIME macro. SVC 34 updates the date field in the communications region of the active partition. Upon return general register 1 contains the time of day in timer units (1/300 sec.)

SVC 35: Protects a track from use by more than one task at a time. An X'FF' is moved to byte 2 of the SVC old PSW to indicate track hold. A requesting task not owning a held track is made inactive and must wait until the track is free. If more than sixteen holds on a track are attempted, the requesting task is canceled.

Exits are to execute the I/O, or to RESVC if the track is already held. At RESVC, the problem program old PSW is set to execute the SVC 35 again, and a branch is taken for task selection. See Figure 6.34 for the Track-Hold Table.

SVC 36: Frees a track that is held by the task issuing the FREE. An attempt to free a track not owned by the requestor results in cancelation of that task.

Exits on a successful FREE are to task selection, or to the DETACH routine if the FREE was issued by that routine.

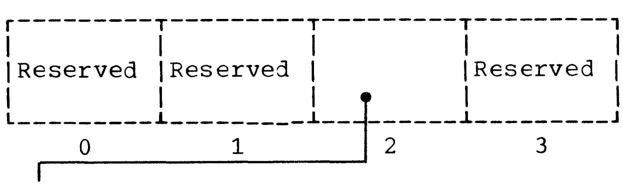
SVC 37: Establishes linkage from the supervisor to a problem program abnormal termination (AB) routine. It stores the routine's entry point and save area address in the AB option table. (The save area is a 72-byte area in which the interrupt status information and general registers 0-15 are stored.) Return is to the program that issued the STXIT macro. See Figure 4.6 for AB option table and Figure 4.9 for the save area.

SVC 38: Initializes a subtask. The main task's PIB and save area are copied to the subtask's PIB and save area. The subtask's PIB flag is set to X'83' to indicate 'ready-to-run'. Bit 0 of the maintask's R1 is set to 0 to indicate a successful attach, and absolute priority is established for the subtask. A subtask attempting to issue an ATTACH is canceled.

SVC 39: Performs normal termination of a subtask. DETACH may be issued by either the subtask being terminated or by the main task.

The subtask's PIB is set inactive (byte 0 = X'80'), and its ECB (see Figure 4.10) is posted for termination. This routine calls the free routine to free any tracks held by this subtask, and a waiting task is removed from the wait state.

If the task to be detached has a timer interrupt pending, the corresponding entry is deleted from the IT option table (ITTAB). Also, if the multiple timer is supported, the IT request table (ITREQ) is updated either by deleting the entry (if in ITREQ) or, if the task owns the current pending timer interrupt (interval value in SYSTIMER), by setting TWTIMS to zero. When the interrupt occurs, it is ignored.



X'80': Normal termination of subtask
X'C0': Abnormal termination of subtask

Figure 4.10. Event Control Block (ECB)

SVC 40: Used for intertask communication. POST may be issued by either a maintask or a subtask. It is issued so that a task is aware of the termination of an event. Normal completion of the specified event is posted in the ECB (byte 2, bit 0 = 1). If the SAVE= parameter is present, only the task waiting for this ECB is taken out of the wait state; otherwise, all waiting tasks are removed from the wait state.

SVC 41: Informs the system that a resource (shared data area) is now available for use by another task. A task may issue the DEQ macro only to a resource that it currently owns. If it attempts to issue the DEQ macro to some other resource, the task is canceled.

If any other tasks are waiting for the resource, the highest priority task ready to run is removed from the wait state and gains control. If no other task is waiting for the resource, control returns to the task that issued the DEQ.

If a task terminates without DEQuing all of its ENQued resources, either in its normal coding or in its abnormal termination exit routine, any task subsequently attempting to ENQ the resource is canceled. See Figure 4.11 for the Resource Control Block (RCB).

SVC 42: ENQ prevents tasks from simultaneous manipulation of a shared data area (resource). This is accomplished, using the TS instruction, by setting to ones all bits of byte 0 of the specified Resource Control Block (RCB). Then the Event Control Block (ECB) address is placed in bytes 4-7 of the RCB.

A task attempting to ENQ a resource that is already engueued by another task is placed in a queue and put in a waiting condition. The old PSW is set to reexecute the SVC 42 and task selection is performed.

A task is canceled if it attempts to nest ENQ(s) of a resource or if it attempts to ENQ a resource that is still owned by a terminated task.

When a task is finished with a resource, it should inform the system by issuing the DEQ macro. If it does not, tasks subsequently requesting that resource are canceled. See Figure 4.16 for the Resource Control Block.

Resource Control Block (RCB)

0	1	2	3	4	5	6	7
	Unused					ECB Address	

Figure 4.11. Resource Control Block (RCB)

Bytes

0: X'FF' if resource is in use, X'00' if resource is not in use

1-3: Unused

4: Bit 0 = 1 Another task waiting for the resource
= 0 No other task waiting for the resource

5-7: ECB address of current resource owner

SVC 44: Supplies support for specific requests to write records on the recorder file. The SVC44 routine checks for an available error queue entry. If none is available, the SVC is reissued until an entry becomes available. The error queue name is then set to call $$ABERA3, and the task select exit is taken.

$$ABERA3 writes the record and takes the supervisor ignore exit. A second $$A transient is called to write messages, if necessary, before returning to the supervisor ignore exit.

SVC 45: Provides emulator interface. It posts the partition communications region byte 134, bit 1, indicating that the emulator is active.

Additional action depends on the contents of register 1. If this register contains a zero value, no additional action is taken.

If register 1 is negative, it contains an ECB address (with the high order bit on), which is made positive and stored in the entry for the issuing partition in the emulator ECB table (see Figure 4.12). If register 1 is positive, it contains a PUB address. The mode setting in this PUB (byte 5) is changed.

SVC 46: Provides OLTEP with the facility to operate in the supervisory state. In the initial issuing of the SVC, register 1 contains an entry point in OLTEP. The next time the SVC is issued, register 1 is zeroed out, forcing task selection.

SVC 47: Provides identification to the supervisor for MICR type device multiple waits (WAITF). The same routine is entered as for SVC 29. However, when no ECB is posted, the task's PSW is not modified to reissue the SVC, as is done when SVC 29 is issued.

SVC 48: Fetches a CRT-transient phase and makes the CRT task dispatchable. The program issuing an SVC 48 is canceled if the PSW protection key does not equal zero.

The first SVC 48 is issued at IPL time to activate the CRT hooks in the supervisor and to load the CRT root phase $$BOCRTA into the CRT transient area (CRTTRNS). Before control is given to task selection, SVC 48 is modified, so that the next time it is called, it will not only load the specified phase but also make the CRT task dispatchable.

SVC 49: Used by VTAM.

SVC 50: Reserved for LIOCS error recovery.

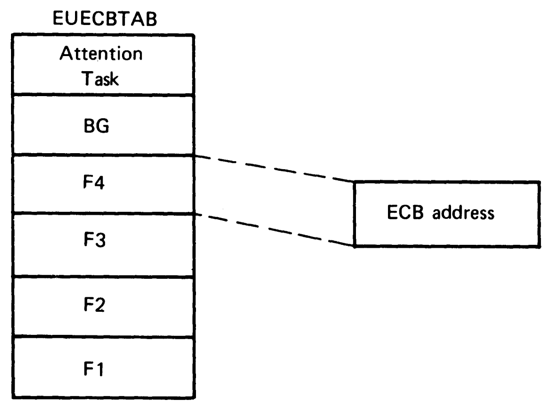


Figure 4.12. Emulator ECB Table (EUECBTAB)

Note: The entries for F1 - F4 are only generated in a supervisor which supports the corresponding partitions.

Bytes 104-107 (X'68' - X'6B') of the System Communications Region (SYSCOM) contain the address of the Emulator ECB Table. Label EUECBTAB identifies the first byte of the table.

SVC 51: Provides the ability to determine the length of a phase without loading it. On entry to this routine register 1 must point to the storage address of the phase name. This 8 byte area must be followed by an area in which as many halfwords of the directory entry of the phase are returned to the user as he requested. The number of halfwords to be returned must be specified in byte 3 of this area. The number of halfwords returned to the user is equal to the requested number plus 2. (See Figure

7.6: Layout of Directory Entry.) The area is not altered if no directory entry for the phase is found in the System Directory List, the System Core Image Library, or a Private Core Image Library.

SVC 52: Returns the remaining time interval if IT=YES is specified at system generation. It provides support for the TTIMER macro.

The remaining interval (in hundredths of seconds) to elapse before the timer interrupt occurs is returned as an unsigned 32-bit binary number in general register 0.

All zeros are returned if no timer interval was set by the program or task issuing the TTIMER macro.

If the task issuing the TTIMER macro owns the current pending timer interrupt, the value in SYSTIMER is returned.

If the task issuing the TTIMER macro owns an entry in the IT Request (ITREQ) table, the value returned is the sum of the values in SYSTIMER and the ITREQ table entry.

If TTIMER CANCEL is specified, in addition to returning the remaining interval, SVC 52 deletes the relevant entry in the IT Table (ITTAB) and, if the multiple timer is not supported, initializes SYSTIMER. If the multiple timer is supported and the task owns an entry in the ITREQ table, then that entry is deleted.

If the currently pending interval is to be canceled, TWTIMS, which contains the PIK or TIK of the task, is set to zero. This indicates that the interrupt, when it occurs, can be ignored because it is no longer required by a user task.

SVC 53: Used by VTAM.

SVC 54: This supervisor call provides supervisory support for the FREEREAL macro to release page frames to the main page pool. These page frames may be released from a real partition, the PDAID alternate area, or the SDAID buffer area.

When the request is issued by the terminator (the page frames belonging to a real partition are freed), the lower and upper addresses (on page boundary) of the real partition are passed to the SVC 54 routine in the registers 2 and 3.

A zero value in register 2 indicates that the request is issued by SDAID or PDAID. In this case the lower and upper limit of the area to be released are obtained from the boundary box (see Figure 2.11). Control is passed immediately to task selection if no SDAID buffer area or PDAID alternate area exists.

The page frames are freed, one after the other, by updating the corresponding page frame table entries (see Figure 5.1) as follows:

- Bits 14 and 15 are reset (page frame returned to the selection pool; temporarily fixing in it allowed).
- Bytes 4 and 5 are set to X'FFFF' (page frame not in use).

The address bits of the corresponding page table entries are cleared, the page frame's storage key is set in bits 8-11, and bit 0 is set to 1.

The released page frames are added to Q00 (refer to the section "Selection Pool Queues" in "Page Management").

If SVC 54 is called by the terminator, the number of active virtual partitions (entry in SYSCOM) is increased by one.

If SVC 54 is called by SDAID or PDAID, the boundary box is updated as follows:

- The end of real storage address is inserted in the entry for the SDAID/PDAID area address.
- The entry for the number of page frames in the main page pool is incremented by the number of freed page frames.

The SVC 54 routine posts the Page Manager System Task ready to run (it may have been fix-bound). It also posts any tasks that are waiting for a page frame ready to run.

SVC 55: This supervisor call provides supervisory support for the GETREAL macro to request page frames from the main page pool for the SDAID buffer area or the PDAID alternate area (further on called SDAID/PDAID area). Control is passed immediately to task selection, if such a request is already in progress or if the SDAID/PDAID area already exists.

The number of requested page frames is passed in register 0. This value is replaced by the number of page frames that are available for GETREAL, if this number is less than the requested number. After handling the request, the number of page frames taken from the main page pool and the address of the SDAID/PDAID area are passed to the user in the registers 0 and 1. Register 0 contains zero and register 1 contains the end address of real storage, in case no page frames are available.

The SVC 55 routine passes the begin and end addresses of the requested SDAID/PDAID area as parameters to the GETREAL routine. Refer to the section "Handling of a GETREAL Request" in "Page Management".

After the SDAID/PDAID area has been assigned, the SVC 55 routine updates the boundary box by inserting:

- The address of the SDAID/PDAID area,
- The updated number of page frames in the main page pool if NPARTS>1, or the end address of the real background partition if NPARTS=1.

SVC 56: Reserved.

SVC 57: Reserved.

SVC 58: Issued by job control to initialize a partition. The task issuing this SVC is canceled if it does not run under a protection key of 0.

The following parameters are passed to this routine:

R2: begin address of partition (if execution in real mode to be initiated).

R3,
R4: addresses of first and last page table entries for partition.

R5: The two low order bytes are used to initialize page table entries. If bit 0 is on: virtual mode.

R7: address of uppermost byte available for problem program.

If the next program in the partition is to run in virtual mode, the following action is taken:

The address of the uppermost byte available to the problem program (either the uppermost byte of the partition, or a lower byte determined by job control from the SIZE parameter of the EXEC statement) is moved to the partition communications region. The instruction address in the PSW in the save area is changed to point to a routine, also located in the save area, that will fetch the program to be executed.

The page table entries, except the first, belonging to the virtual partition are initialized (invalidated).

For any pages that are in real storage, the storage key of the associated page frames is set to zero and the page frame table entries are initialized. They are set to unused and are queued to Q00 of the selection pool (refer to the section "Selection Pool" in "Page Management"). For this the same routines are used as described for SVC 59 (see SVC 59, below). The Translation Look Aside Buffer in the CPU (if there is one), is made invalid. If a page is found to be TFIXed, the system enters a hard wait state.

If the partition is to run in real mode, in addition to the action described above, the following action is taken:

- The partition save area is moved to a work area in the supervisor, updated there, and moved to the real partition.
- All page table entries belonging to the virtual partition are initialized.
- The fifth byte in the PIB is set to indicate real mode and the entry in SYSCOM indicating the number of active virtual partitions is decreased by one. The appropriate entry in the PIB is updated to point to the real save area instead of the virtual.

To reserve the page frames that are required for the partition to run in real mode, the GETREAL routines are used (refer to the section "Handling of a GETREAL Request" in "Page Management"). The number of the page frames to be reserved depends on the SIZE parameter of the EXEC statement (analyzed by job control), or on the size of the real partition.

If, during the execution of the GETREAL a page frame in the real partition is found to be failing, the job that is to be initialized is canceled (cancel code X'2D'). If it is not the first page frame that is failing, the save area set up in the supervisor is moved to the real partition and the entry in the partition communications region for the uppermost byte available to the problem program is updated.

If the first page frame is failing, the first page of the virtual partition is initialized, the save area pointer in the PIB is changed to point to the virtual save area, virtual mode is posted in the PIB, the save area set up in the supervisor is moved to the virtual partition and the entry in the partition communications region for the uppermost byte available to the problem program is updated.

SVC 59: Initializes the page table and page frame table entries belonging to specific pages. The task issuing an SVC 59 is canceled if the protection key in the PSW is not 0.

Upon entry register 3 contains the address of page table entry of the lowest page and register four the address of the page table entry of the highest page. Register 5 contains the information which must go into the page table entries in its lower two bytes. Each page table entry between the two limits is initialized by storing the lower two bytes (with storage key and bit to set the page invalid, see Figure 5.2) of register 5 into the entry. If the page referred to by an entry is in real storage, the page frame table entry of the corresponding page frame is initialized as follows:

- The frame is marked as unused (bytes 4 and 5 are set to X'FFFF') and given to the selection pool (byte 1 bit 6 is set to 0).
- The frame is removed from its queue (if it was queued) and added to Q00. (Refer to the section "Page Management" for an explanation of selection pool and Q00.)

If the corresponding entry in the page frame table extension (PFIX counter) is not zero, it is set to 0. The key of the storage block in this page frame is set to zero. The process is repeated until the entries for all requested pages have been initialized.

SVC 60: Calculates from the real address, the virtual address of a location within the data area of an I/O request.

Before issuing this SVC, general register 8 must contain the address of the CCW, and general register 0 must contain the displacement of the desired address from the start of the I/O area. Using the data address or the address of the Indirect Addressing list (IDAL), specified in the CCW, the supervisor calculates the virtual address and returns it in general register 15.

If the real address is invalid (in an unused page frame or beyond the end of real storage), all zeros are returned.

SVC 61: Provides the supervisor support for the imperative GETVIS macro. It reserves part of the GETVIS area which may either be part of a virtual partition or part of the Shared Virtual Area (SVA).

On successful completion of the operation, X'00' is returned in general register 15; the start address of the reserved area returned in general register 1; and the length of the area, which must be specified by the user, is contained in general register 0.

If the GETVIS area is part of a partition, the length of the specified area must be a multiple of 128 bytes; if it is part of the SVA, it must be a multiple of 512 bytes. If the specified length is not a multiple of 128 or 512, it is rounded to the next higher multiple of 128 or 512, respectively.

Reservation of the required area is registered in the Virtual Storage Table (VISTAB) of either the SVA GETVIS area or the partition GETVIS area.

The VISTAB is a bitstring that starts at byte 1024 of the GETVIS area. In a partition GETVIS area, each bit in the VISTAB refers to 128 bytes; the first VISTAB bit refers to the first 128 bytes which start at byte 2048 of the GETVIS area. In the SVA GETVIS area, each bit in the VISTAB refers to 512 bytes; the first VISTAB bit corresponds to the first 512 bytes which start at byte 4096 of the GETVIS area. If the VISTAB bit is 1, the associated 128 or 512 bytes are already reserved; if it is 0, they are free.

Each bit in VISTAB is checked until a string of zeros representing the required length is found. The area associated with this bit string is then reserved by setting each bit to 1. The start address of the reserved area is placed in general register 1, and the successful completion code is returned in general register 15.

If the operand PAGE is specified, general register 15, bit 31, is set to 1 before searching for the requested area starts. GETVIS then looks for free blocks that start on a page boundary. Free blocks that do not start on a page boundary are ignored.

If the operand POOL is specified, general register 15, bit 30, is set to 1 before searching for the requested area begins. Searching then starts at the address specified in general register 1.

If the operand SVA is specified, general register 15, bit 29, is set to 1 and the SVA GETVIS area is searched for the requested storage block. Otherwise, the partition GETVIS area is searched.

Return Codes in Register 15: The following codes are returned in register 15 after a GETVIS has been issued:

X'00' GETVIS completed successfully.

X'04' The GETVIS area is (a) 0K if it is part of a partition, or (b) 0K or 2K if it is part of the SVA.

X'08' The GETVIS macro was issued by a program running in real mode.

X'0C' No more virtual storage is available in the GETVIS area, or the length specified is smaller than 0K.

SVC 62: Provides the supervisor support for the imperative macro FREEVIS. It releases a block of virtual storage. The start address of the area to be released is contained in register 1; the length of the area to be released is in register 0.

If the operand SVA is specified, general register 15, bit 29, is set to 1 and the SVA GETVIS area is searched for the storage block to be released. Otherwise, the partition GETVIS area is searched.

The area is released by setting the associated bits in the VISTAB to zero. (The VISTAB is explained in SVC 61.)

Return Codes in Register 15: The following codes are returned in register 15 after a FREEVIS has been issued:

X'00' FREEVIS completed successfully.

X'04' The GETVIS area is (a) 0K if it is a part of a partition, or (b) 0K if it is in the SVA.

X'08' The FREEVIS macro was issued by a program running in real mode.

X'0C' The specified address is not within the GETVIS area or the address is not (a) a multiple of 128 bytes if the GETVIS area is part of a partition, or (b) a multiple of 512 bytes if the GETVIS area is in the SVA.

X'10' The specified storage block (ADDRESS+LENGTH) to be released exceeds the GETVIS area, or the length specified is smaller than zero.

If the return code is not zero, no action is taken by FREEVIS.

SVC 63: Allows supervisor controlled access to system resources as requested by the internal macro USE. The use of one or more of the following system resources may be requested:

Type I System Resources

- procedure library.

Type I system resources are those which are frequently used by a partition for the duration of one or more jobs. Type I resources are mainly used by job control.

Type II System Resources

- VSAM master catalog open routine
- label cylinder on SYSRES
- SYSRES file
- VSAM master catalog
- all disk packs represented by their VTOCs.

Type II system resources are those which are normally used for a short period of time, that is, never longer than for one job step. Type II resources are mainly used by OPEN, CLOSE, and EOV routines.

When SVC 63 is called, register 1 must point to the list of resources that are requested. If a resource is available, it is assigned to the task by means of a Resource Usage Record (RUR) that identifies the owner(s) of the resource and the type of usage. (shared or exclusive). See Figure 4.13, which shows the format of a RUR, the layout of the RUR table, and the names that must be in the USE macro for each RUR.

If a resource is not available and FAIL=RETURN is specified in the USE macro, control is returned to the interrupted task and the return code 4 (= resource unavailable) is passed in register 0. Register 1 points to the resource that is unavailable. The requests for resources following the unavailable resource are ignored.

If a resource is not available and the operand WAIT is specified in the USE macro, the task is put in the wait state and the resource it is waiting for is indicated in the wait list. The wait list is located immediately behind the RUR table. It contains a 2-byte entry for each task in ascending order of PIKs, that is, the first entry refers to the attention task (PIK=X'00'), the second to the BG task (PIK=X'10'), etc. The wait list has 16 entries if AP=YES, otherwise the number of entries is equal to NPARTS plus 1. An entry in the wait list contains either the code of the resource for which the task is waiting or X'FFFF' (= task is not waiting for a resource). The various resource codes are:

Resource Name in USE Macro	Associated Resource Code in Wait List
SYSPL	X'8000'
SYSMCO	X'C000'
SYSLBL	x'C001'
SYSRES	X'C002'
SYSCTLG	X'C003'
SYSVTOC	X'C004'

The resource is assigned to the task after it has been released. Any following requests are then processed.

A task issuing the USE macro with the operand WAIT is canceled if it requests use of a resource it already owns exclusively or if it requests exclusive use of a resource for which it has already shared use.

The task is also canceled if the parameter list in the USE macro is invalid (for instance, if resource names are not in ascending order).

<u>SVC 64</u>: Releases all the resources specified in the internal macro RELEASE. When SVC 64 is called, register 1 must point to the list of resources that are to be released. The resources are released, one after the other, by setting the ownership bit in the corresponding RURs to 0. (Refer to <u>SVC 63</u> for a description of the RURs). A release request is ignored if it is given for a resource the task does not own.

An SVC 64 is normally issued by the terminator ($$BEOJ) and $$BEOJS1) with a zero value in register 1. If SVC 64 is called after the end of a job step, it releases all resources owned by the task except for which the Job Control owner bit in byte 1 of the RUR is on (see Figure 4.13).

If SVC 64 is entered and register 1 is negative, all resources owned by the task are released, including those that are marked in byte 1 of the RUR.

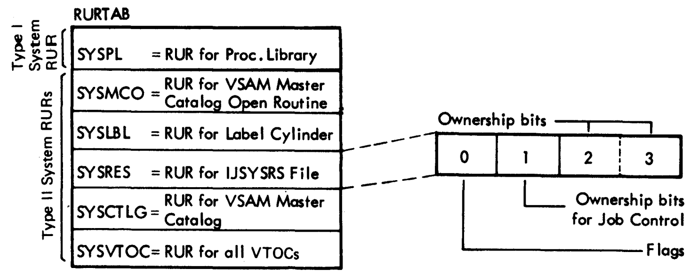


Figure 4.13. Resource Usage Record Table (RURTAB)

<u>Byte 0</u>:

Bit 3: 1 = Another task waiting for this resource
5: 1 = Resource shared among owners indicated
6: 1 = Resource used exclusively by indicated task
7: 1 = Shared or exclusive use of resource

<u>Byte 1</u>: Used by Job Control to indicate that the resource is to be held for the duration of a job.

Bit 2: 1 = TIK or PIK is X'50'
3: 1 = TIK or PIK is X'40'
4: 1 = TIK or PIK is X'30'
5: 1 = TIK or PIK is X'20'
6: 1 = TIK or PIK is X'10'

<u>Byte 2</u>:

Bit 0: 1 = TIK is X'F0'
1: 1 = TIK is X'E0'
2: 1 = TIK is X'D0'
3: 1 = TIK is X'C0'
4: 1 = TIK is X'B0'
5: 1 = TIK is X'A0'
6: 1 = TIK is X'90'
7: 1 = TIK is X'80'

<u>Byte 3</u>:

Bit 0: 1 = TIK is X'70'
1: 1 = TIK is X'60'
2: 1 = TIK or PIK is X'50'
3: 1 = TIK or PIK is X'40'
4: 1 = TIK or PIK is X'30'
5: 1 = TIK or PIK is X'20'
6: 1 = TIK or PIK is X'10' (BG)
7: 1 = TIK or PIK is X'C0' (Attention)

<u>Note</u>: For explanation of TIK and PIK see Task Interrupt Key and Partition Identification Key. Label RURTAB identifies the first byte of the table.

SVC 65: Loads a phase dynamically when called by the internal macro CDLOAD. The name of the phase to be loaded, specified as first operand in the CDLOAD macro, must be pointed to by register 1 before SVC 65 is entered.

SVC 65 first issues a LOAD with the parameters DE=YES and TXT=NO, that is, the FETCH routine will move only the directory entry of the requested phase into the area specified by SVC 65 (a 34-byte area starting at CALLNAME in the system save area of the corresponding task).

SVC 65 then checks the directory entry. If the phase is not found, control is passed to ERR22. If the phase is loaded in the SVA, the required parameters are retrieved from the directory entry, the address of the load point is moved to register 0, the address of the entry point is placed in register 1, and the length of the phase is stored in register 14. In addition, return code X'00' (successful completion) is passed in register 15.

If the phase is not loaded in the SVA, SVC 65 checks the anchor table of the partition (see Figure 4.14 for the layout of ANCHTAB) to determine whether the phase has already been loaded in the partition. If so, the parameters are retrieved from the ANCHTAB entry and stored as described above.

If the phase name is not in the anchor table, the phase name is inserted in the first free place in the table and the status switch for the phase is set to X'00'. SVC 65 then obtains the length of the phase to be loaded from the directory entry and passes this information to the GETVIS routine (see SVC 61).

The GETVIS routine reserves the required storage (see SVC 61) and returns the load address of the phase to SVC 65. SVC 65 then loads the phase by issuing a LOAD with the parameters TXT=YES and DE=YES. After completion of the load operation, the load point, the entry point, and the length of the phase are stored in the anchor table and in registers 0, 1, and 14, respectively. Successful completion is indicated by passing the return code (X'00') in register 15.

When multitasking is supported (AP=YES), it is possible that a phase requested by a task is in the process of being loaded by another task. This is indicated in the anchor table entry for the phase (status switch =X'FF'). The task requesting such a phase is made resource-bound.

If the anchor table is full and the phase cannot be stored in the table, return code X'16' is passed in register 15. A task is canceled if the requested phase is not contained in the core image library.

SVC 66: Indicates mode in which the program is running. The routine tests the PIB DAT flag (fifth byte in PIB) and returns 0 in register 1 if the program is running in virtual mode (PIB DAT Flag = X'80'). If the program is running in real mode, 4 is returned in register 1.

SVC 67: Fixes one or more pages as requested by the PFIX macro. When SVC 67 is entered, register 1 must point to the list of pages that are to be fixed. Each entry in the list consists of 8 bytes. The first four bytes contain the beginning address of the area that is to be fixed and the last four bytes contain the length of that area.

The PFIX requests are gated, that is, a task is set to PFIX-bound if it issues a PFIX request for a partition for which another PFIX request is still being processed.

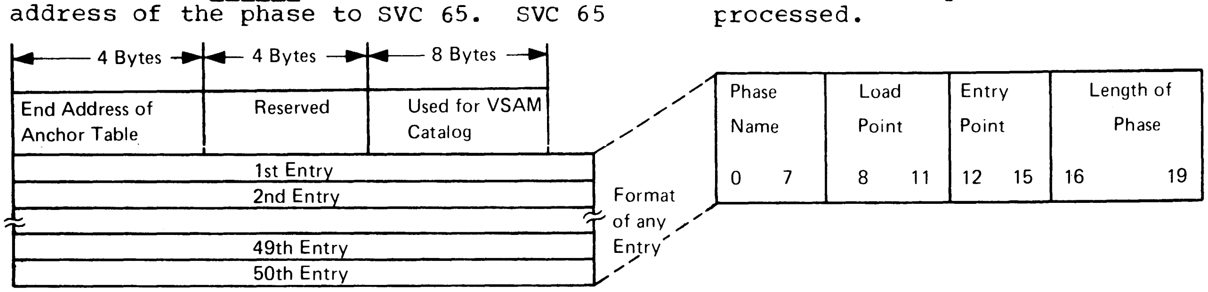


Figure 4.14. Anchor Table (ANCHTAB)

The anchor table is located at the beginning of the GETVIS area in the virtual partition.

Byte 16 - Status Switch:

X'00' = Phase must be loaded.
X'7F' = Phase is already in storage.
X'FF' = Requested phase is just loaded by another task (only if AP=YES).

Before a page can be fixed, a page frame must be selected for the page. The address of the reserved page frame is stored in the table FIXWTAB either by the PFIX routine itself or by the PFREE (see SVC 68) if a page has first to be freed before the page frame can be reserved. The format of FIXWTAB is shown in Figure 4.15.

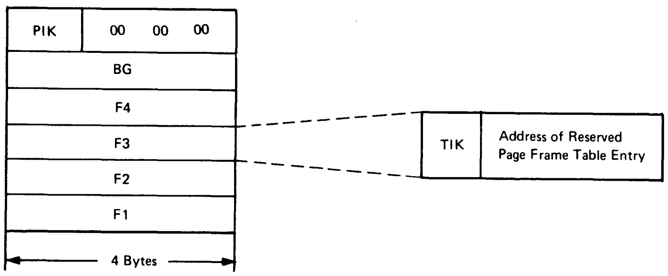


Figure 4.15. FIXWTAB

First entry:

Byte 0: PIK of issuer of GETREAL request (SVC 55). Reset to zero when request has been handled.

Following entries (one for each partition):

Byte 0: TIK of task issuing the PFIX request. Inserted when the PFIX request is started. Reset to 0 after the request has been handled.

Note: The number of entries generated is equal to NPARTS+1, if PFIX=YES. Only the first byte of the first entry is generated of PFIX=NO.

The addresses in FIXWTAB are used to fix the page. The PFIX routines fix the pages, one after the other, until the list of requests is exhausted. How a page is actually fixed is described in the section "Page Management" under "Handling of a PFIX Request".

If a page to be fixed has an invalid address, all pages that have already been fixed for the request are freed, return code 12 is placed in register 15, and control is returned to task selection. Control is also returned to task selection and the already fixed pages are freed if not enough page frames are available to fix the requested number of pages. Return code 4 will be passed in register 15 if the real partition is too small to ever satisfy the request. Return code 8 is passed, if the partition is large enough to satisfy the request but not enough page frames are available to satisfy the request at this time.

If no page frame is available but some of them are only temporarily fixed, all page frames in the partition are set to "not temporarily fixable" and the task is put into the wait state. Processing of the request continues after a page has been freed by a TFREE request.

The PFIX request is ignored if it was issued by a program running in real mode, and return code 0 is passed in register 15.

SVC 68: Frees one or more pages as requested by the PFREE macro. A PFREE request may come from a user task or from the RSTRT command processor. When SVC 68 is entered, register 1 points to the list of pages to be freed. Each entry in the list is 8 bytes long. The first 4 bytes contain the address of the area to be freed and the last 4 bytes contain the length of that area. The pages are freed one after the other until the list of requests is exhausted. If a page is not in real storage, the PFREE request for this page is ignored.

When a page is freed, the PFIX counter (PFTX) is decreased by 1. If the counter is not zero after 1 has been subtracted or if the page frame is temporarily fixed, the next page to be freed is determined. Otherwise the page frame is released and added to the hold queue of the selection pool. (Refer to the section "Page Management" for a description of the selection pool and the PFIX counter.) Any task waiting for a freed page is then posted and the next page to be freed is selected if a page may be temporarily fixed in the released page frame (bit 15 in the page frame table entry is 0). If a page may not be fixed in the released page frame, the address of the page frame is inserted into FIXWTAB (see SVC 67) and thus reserved for the next PFIX request. All page frames in the partition, except the reserved one, are set to temporarily fixable before the next request is processed.

SVC 69: Returns the real address of the virtual address specified in the REALAD macro. On entry to SVC 69, the virtual address must be contained in register 1. The real address is returned in register 0.

No address is returned if

- the page containing the address is not in real storage;
- the real address is invalid;
- the address is within a page that is not fixed.

SVC 70: Returns the virtual address of the real address specified in the VIRTAD macro. On entry to the routine, the real address must be contained in register 1, and register 0 must contain zero. The virtual address is returned in register 0.

No address is returned (register 0 contains zero) if

- the address is contained in a page that is not used;
- the real address is invalid;
- the address is within a page that is not fixed.

SVC 71: Provides support for the SETPFA macro. SVC 71 establishes linkage between the supervisor and the user-written page fault appendage routine by storing the address of the routine in the page fault appendage table PFATAB. (See Figure 4.16 for the format of PFATAB.) The address of the appendage routine must be contained in register 0 when the routine is entered.

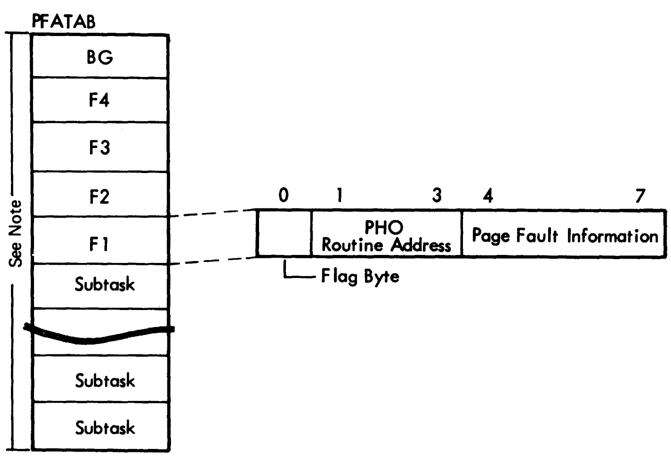


Figure 4.16. Page Fault Handling Overlap Option Table (PFATAB)

SETPFA Issued

Bytes:

0: Bit 0: 1 = A page fault from the task is in the Page Queue
   1: 1 = A page fault occurring in a supervisor service of the task is queued in the Page Queue Table
   2-7: Unused

1-3: Address of the user's page fault appendage routine

4-7: Page fault information (for detailed layout see Page Queue Table)

This information is stored here only when a page fault occurs in a supervisor service working for the task, while another page fault is queued for the task in the Page Queue, otherwise zero.

NO SETPFA Issued

Bytes 0-7: zero

Note: One table entry is generated for each partition supported. With asynchronous processing support, the table always comprises 15 entries; the subtask entries occupy the higher address locations in the table. Label PFATAB identifies the first byte of the table.

PFATAB is only built if PHO=YES was specified in the SUPVR macro at supervisor generation.

SVC 72: Gets or releases a copy block used for channel program translation. The program issuing an SVC 72 is canceled if the PSW protection key does not equal 0.

If a request for copy blocks is made, the chain of free copy blocks is searched. The requesting task is set into the wait state if the chain is empty. If the chain is not empty, a copy block is dequeued from the chain.

A copy block is released by enqueueing it to the chain of free blocks. Any tasks waiting for copy blocks are then posted.

SVC 73: Authorizes linkage to a channel end appendage routine by setting byte 134 bit 6 in the partition communications region to 1.

SVC 74: Fixes a page in case of restart. When a restart is made, a page has to be fixed in the same page frame in which it was fixed at checkpoint time. On entry to SVC 74, register 1 points to a list of pages that have to be fixed. Each entry in the list consists of 8 bytes and has the following format:

Address of Page to be Fixed 0 3	PFIX Counter 4 *	Address of Page Frame 5 7

*Indicates the number of times the page has to be fixed.

Figure 4.17. Restart-PFIX Parameter List Entry

SVC 74 inserts the address of the reserved page frame into the FIXWTAB entry (see Figure 4.17) and transfers control to the

PFIX routines to fix the page (see SVC 67). If more pages have to be fixed (indicated by bit 31 in the FIXWTAB entry) control is returned to SVC 74 to select the next page to be fixed. The process is repeated until all pages in the list have been fixed.

SVC 75: Calculates the sector value for a position on a track of a DASD device supporting Rotational Position Sensing (RPS). As requested by the SECTVAL macro instruction, the routine calculates the position for either fixed or variable length records. The sector value is used as the argument for the RPS channel command Set Sector.

On entry to SVC 75 for fixed length records, Register 0 contains the data length, key length, and record number. On entry to SVC 75 for variable length records, Register 0 contains the number of bytes used on the track up to the current record, the record number, and an indication to whether the records are keyed. In either case, Register 1 contains a device type code (3330=X'04', 3340=X'08'). Register 0 contains the calculated sector value on return to the caller.

If the supervisor was generated without RPS specified in the FOPT macro instruction, the issuing program is canceled.

SVC 76: Initiates the recording of a RMSR record on the Recorder File (SYSREC). If the system runs under VM/370, not all information in the record may be valid. VM/370 gains control to perform the recording function. When not running under VM/370, the effect of this SVC is the same as for SVC 15 (SYSIO).

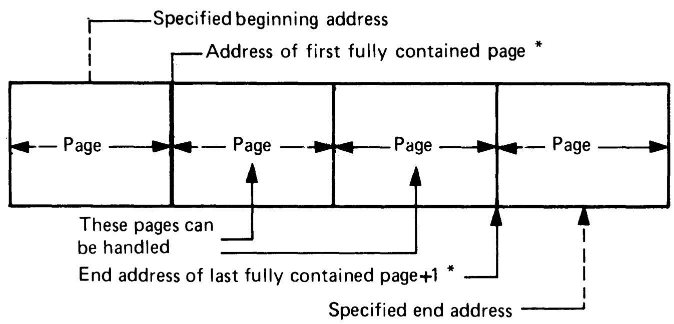
The address of the user's CCB must be supplied in general register 1 before this SVC is issued. The data address must be supplied in general register 0. Register 1 must have the high-order bit on to indicate that VM/370 must intercept the interruption. After having intercepted, VM/370 zeros out this register so that on return, the issuing program can check whether VM/370 handled the I/O.

SVC 77: Used by routines which print the CCW address of a failing I/O operation, such as the ERP message writer. The virtual address of a copied CCW is calculated. On entry to the SVC 77 routine, register 0 contains the address of the copied CCW, and register 1 the address of the copied CCB.

The retranslated CCW address is returned in register 0, if the address passed in this register by the user is found to be actually pointing to a copied CCW related to the I/O operation being handled. If the user passed an invalid address, register 0 will contain the value zero on return. The contents of register 1 are not changed.

SVC 85: Provides the support for the RELPAG macro. Part of the code is common for both this SVC and SVC 86 (FCEPGOUT macro). If the code is entered as a result of an SVC 85, byte SVCID is set to C'R' to indicate that a RELPAG macro is to be processed.

When the SVC routine receives control, register 1 points to a list of 8-byte area specifications that are to be processed. The specified beginning and end addresses hardly ever coincide with the beginning and end addresses, respectively, of a page. Therefore, these addresses have to be changed as shown below before an area specification can be processed.



*The specified addresses are changed to the corresponding page boundaries.

Page handling for an area specification is done in a loop at NEWPAGE. This loop is executed once for each page that is fully contained within a specified area. The loop checks each page for the conditions listed below, sets the appropriate bit in the return code if the condition is found to exist, and discontinues the processing for the page. The conditions are:

- The page is outside the address range of the requesting program's partition.
- The page is currently fixed.
- The page has an entry in the page queue PGQU.

For a page that can be handled, the loop computes the location of the associated entries in the page table and the page frame table. Further processing for the page depends on whether a RELPAG or a FCEPGOUT request is being processed.

Processing a RELPAG request (SVC 85): If bit 13 of the page-table entry (see Figure 5.02 for its format) is 1, that is, the page is marked as not in real storage, the loop is reentered to handle the next (fully contained) page. If the bit is 0, the loop sets it to 1 and inserts the storage key of the requesting program's partition into the page's page-table entry (bits 8-11). In addition, the loop removes the associated page-frame table entry from its current queue location and enqueues it at the beginning of page-frame selection queue Q00. This makes the page frame available for reuse, and a later reference to the page does not result in a page-in operation.

Processing a FCEPGOUT request (SVC 86): If bit 13 of the page-table entry (see Figure 5.02 for its format) is 1, that is, the page is marked as not in real storage, the loop is reentered to handle the next (fully contained) page. If the bit is 0, the loop sets the page frame's reference bit to 0; in addition, the loop removes the associated page-frame table entry from its current queue location and enqueues it at the beginning of

queue Q00 if the change bit is 0,
queue Q01 if the change bit is 1.

Looping is controlled by incrementing the value in BEGAD by 2K each time the loop is executed and by comparing this value with the value in ENDADR. BEGAD is initially set to the beginning address of the first fully contained page within a specified area, ENDADR to the end address of the last fully contained page plus 1. When the values in BEGAD and ENDADR are equal, register 1 is incremented by 8 to point to the next area specification and control is passed to label CHKEND.

When the first byte of the entry pointed to by register 1 contains a nonzero value, the current page-handling request (a RELPAG or a FCEPGOUT macro) has been serviced and control is returned to the program that issued the request.

A return code is provided in register 15 as follows:

X'00' = Successful completion of the page-handling request.

X'02' = One or more negative area lengths were specified.

X'04' = One or more of the pages to be handled were outside the requesting program's partition -- for these pages, the request was ignored.

X'08' = One or more of the pages to be handled were fixed -- for these pages, the request was ignored; or one or more of the pages to be handled had an entry in the page queue PGQU -- for these pages, the request was ignored.

Note: Several or all of conditions 2, 4, and 8 may be combined.

SVC 86: When a program issues a FCEPGOUT macro, this SVC provides the required linkage to the common support for that macro and the RELPAG macro. This SVC routine sets byte SVCID to C'O' to indicate that a FCEPGOUT macro is to be processed and then transfers control to the SVC 85 routine at label BEGIN. For further information, see SVC 85 above.

SVC 87: Provides support for the PAGEIN macro by initiating the PAGEIN task.

The SVC routine ignores a user PAGEIN request if one of the following conditions exists:

- The PAGEIN macro was issued by a program running in real mode.
- The list of areas that are to be paged in is not completely contained in the requesting program's partition.
- The table PAGETAB (further described below) is full.
- The ECB address, if specified, is outside the requesting program's partition.

When the SVC routine receives control, register 1 points to a list of area specifications. Each entry in that list is eight bytes long and contains the following:

Bytes 0-3: the address of the area to be paged in.
Bytes 4-7: the length of that area minus 1.

Register 0 points to an ECB if an ECB address was given in the PAGEIN macro, or it contains zero.

For each PAGEIN request, the SVC routine builds an 8-byte entry in a table called PAGETAB (see Figure 4.18).

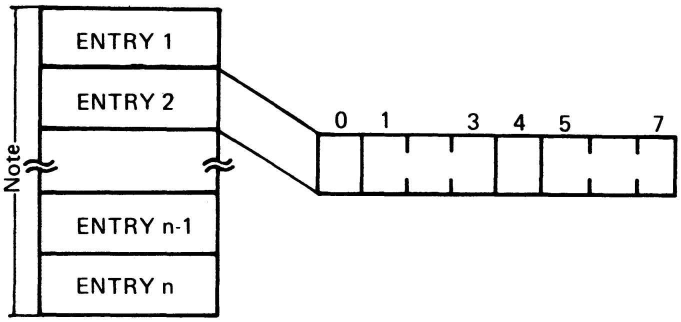


Figure 4.18. PAGEIN Table (PAGETAB)

where TIK = identifier of task that issued the PAGEIN macro
IND = indicators as follows:

Bits	Meaning if 1
0	PAGEIN request has been completed.
1	Reserved.
2	One or more of the requested pages are outside the requesting program's partition.
3	One or more negative length specifications were detected.
6-7	Reserved.

A1 = pointer to be paged in.
A2 = pointer to ECB (if used) or zero.

The entries of the table are stacked (and processed by the PAGEIN task) FIFO. The maximum number of table entries is specified in the PAGEIN parameter of the SUPVR system generation macro.

For a valid PAGEIN request, the SVC routine passes control to the PAGEIN task either directly or via task selection.

If the address of an ECB was specified in the PAGEIN macro, information is returned in byte 2 of that ECB as shown below.

Bit	Meaning if 1	Set by SVC Routine	Set by PAGEIN Task
0	PAGEIN request completed (see Note below).	X	X
1	PAGEIN is full.	X	
2	One or more of the requested pages are outside the address range of the requesting program's partition.		X
3	At least one negative length has been detected in the processed area specifications.		X
4	List of areas that are to be paged-in is not completely contained in the requesting program's partition.	X	
5-7	Reserved.		

Note: Bit 0 is set by the PAGEIN task if that task receives control to process the pertinent PAGEIN request, otherwise the bit is set by the SVC routine.

SVC 90: Provides support for the PUTACCT macro. If POWER/VS provides account support, the POWER/VS account (SVC90/91) appendage is entered. Otherwise the user ECB is posted (byte 2, bit 0).

SVC 91: Provides interface between Job Control and POWER/VS. On entry the POWER/VS account (SVC90/91) appendage is given control.

For a page (2K bytes) of virtual storage one of the following conditions may be valid:

- It may be assigned to a page frame in real storage, in which case its contents may be processed by the CPU.
- If it is not in real storage but still a usable page, the correct contents may be on the page data set.
- It may be usable but not yet contain any information, that is, logically cleared to zero. In this case it is not necessary to have a valid copy anywhere until it is used.
- Its contents may be non-addressable, that is, it doesn't exist at the moment.

Which of the above states a page has is indicated in its page table entry. The page table entries are initialized both during SYSGEN and again at IPL. During operation, however, the state of an individual page may be changed by one of the following conditions:

- page translation exception (page fault),
- TFIX/TFREE (temporary fix and corresponding free),
- PFIX/PFREE (user fix and corresponding free), or
- GETREAL/FREEREAL request.
- PAGEIN, RELPAG and FCEPGOUT requests.

The above conditions do not only affect the status of the pages but also the status of the page frames they occupy. The status of the individual page frames is indicated in the page frame table entry for each page frame (PFTE). The page management routines of the supervisor use both the page tables and page frame tables in their operation. A third table, the segment table, is used only by DAT to reference page tables and is not modified during system operation.

TABLES

Pages in virtual storage are managed using three tables:

1. Page frame table, which contains one entry for each page frame in real storage.
2. Segment table, which contains one entry for each page table in virtual storage.
3. Page tables, each of which contains entries for 32 pages of virtual storage. Together the page tables contain entries for every page in virtual storage.

The last two tables are used by the DAT feature to translate virtual addresses into real addresses.

PAGE FRAME TABLE

The page frame table is built at supervisor generation time and contains one 8-byte entry for each 2K block of real storage (page frame) as specified in the RSIZE parameter of the VSTAB macro.

Initialization of the Page Frame Table

At IPL all page frame table entries which have corresponding page frames in real storage are initialized as follows:

Bytes 0 and 1:

Bits 0-13 = 0

Bit 14 = 0 if the page frame is not in the supervisor area;
= 1 if the page frame is in the supervisor area.

Bit 15 = 0

Bytes 4 and 5:

X'FFFF' if the page frame is not in the supervisor area;

a/2048 (the page number) or a÷2048 if the page frame is in the supervisor area, whereby a is the address of the corresponding page in virtual storage.

All page frames in the selection pool (i.e., those not in the supervisor) are initially queued to Q00.

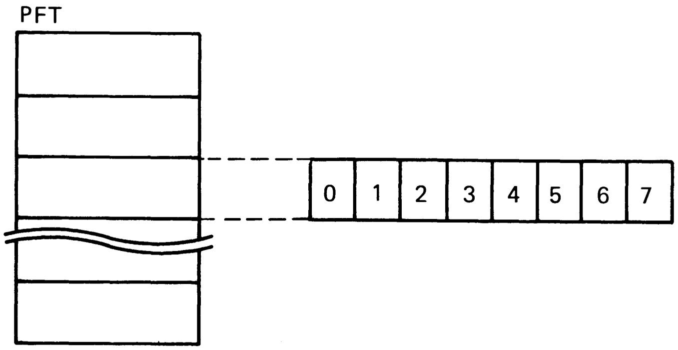


Figure 5.1. Page Frame Table

Bytes 0-1:

Bits 0-10: Counter for temporary fixes (TFIX counter)
11: Reserved for later use
12: 1 = PFIX request (into another page frame) pending for the page (NFF bit)
13: If on, page frame is unusable (hardware failure, DRAP bit)
14: If 0, page frame belongs to selection pool (SP bit)
15: 1 = PFIX (different page) or GETREAL request pending for the page frame (NF bit)

Note: If bit 12 or 15 is on, TFIX requests for the page occupying the page frame are only honored when the page is already TFIXed (TFIX counter not equal to zero).

Bytes 2-3: Pointer to next page frame in queue or to queue header if last element in the queue (all page frames in selection pool are queued)

Bytes 4-5: If the frame is occupied: the page number (the virtual address of the page divided by 2048); if unused X'FFFF'

Bytes 6-7: Pointer to the previous frame in the queue or to queue header if first element in the queue (all page frames in the selection pool are queued)

Note: Bytes 212-215 (X'D4' - X'D7') of the System Communication Region (SYSCOM) contain the address of the Page Frame Table. Label PFT identifies the first byte of the table.

Page Frame Table Extension (PFTX)

For each page frame table entry in the system there is a one-byte entry in the page frame table extension. Each entry is a counter for the number of times a page has been permanently fixed in the page frame (PFIX counter).

Bytes 216-219 (X'D8' - X'DB') of the System Communication Region (SYSCOM) contain the address of the Page Frame Table Extension. Label PFTX identifies the first byte of the table.

SEGMENT AND PAGE TABLES

When the supervisor is generated, one segment table is generated for the total of virtual storage. This table contains one entry for each 64K-byte segment of virtual storage (total of RSIZE and VSIZE) or part thereof. For each segment a page table is built containing 32 entries (one entry for each page in the segment). The page tables are used to manage the pages in virtual storage and to provide the correct page frame number for the DAT feature. The segment table facilitates address translation by DAT. It is not used by the control program for page management activities. A schematic representation of the segment table and the page tables is shown in Figure 5.2.

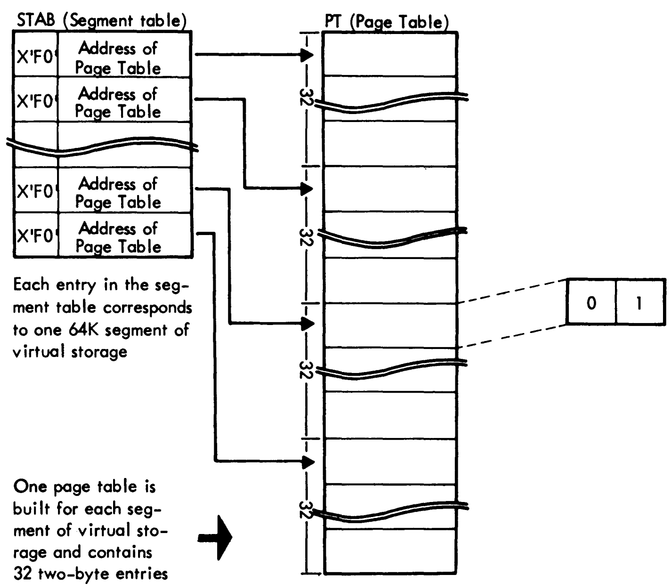


Figure 5.2. Segment and Page Tables

Bytes 0-1:

Bit 0: 1 = Address of any byte in the corresponding page is invalid
0-12: Leftmost 13 bits of address of page in real storage if bit 13=0
8-11: Storage key of corresponding partition if bit 13=1
13: 0 = Page is in real storage
1 = Page is not in real storage

14: Always zero
15: Used bit:

0 = Page must be read from page data set (a valid copy of the page is on the page data set)

1 = Page need not be read from page data set (no valid copy of the page on the page data set)

Note: Bits 4-7 may contain value of PIK divided by 16 of partition which requires PFIXing of the page currently located in corresponding page frame. The page must be PFIXed in a reserved page frame in the correct real partition.

Bytes 208-211 (X'D0' - X'D3') of the System Communication Region (SYSCOM) contain the address of the Segment Table. Label STAB identifies the first byte of the table. Label PT identifies the first byte of the Page Table.

Page Table Initialization

After IPL page table entries are initialized as follows:

- all page table entries belonging to the supervisor area (nucleus and transient areas):

  Bit 13 = 0
  Bit 15 = 1
  Bits 0 - 12 the leftmost 13 bytes of the address of the corresponding page frame.

- all page table entries for allocated real partitions:

  Bit 13 = 0
  Bit 15 = 1
  Bit 0 = 1
  Bits 8 - 11 = storage key of the partition

- page table entries belonging to virtual partitions:

  Bit 13 = 1
  Bit 15 = 1
  Bit 0 = 0
  Bits 8 - 11 = storage key of corresponding partition.

- all remaining page table entries:

  Bit 13 = 0
  Bit 15 = 1
  Bit 0 = 1
  Bits 1 - 12 = 0

SELECTION POOL

The selection pool consists of all page frames which can be selected by the page management routines for paging. The page frames that can be selected are those which do not belong to the supervisor, to active real partitions, or to the PDAID alternate or SDAID buffer area, and which are not fixed in some way (either by TFIX or PFIX). The page frame table entries of the page frames in the selection pool are queued in one of five queues (Q00, Q01, Q10, Q11, or HQ), which are used for page frame selection. Each queue has a queue header, which is 8 bytes long (see Figure 5.3). Bytes 2 and 3 point to the first queue entry and bytes 6 and 7 to the last queue entry. How the selection pool page frame entries are queued is explained in the following section and in the section on the page frame selection.

Selection Pool Queues

The Page Frame Table (PFT) entries of the page frames in the selection pool are organized in five queues. Each PFT entry of the selection pool belongs to one queue. One of the queues is the Hold Queue (HQ). When the page management routines have selected a page frame for a paging operation, its PFT entry is queued to the HQ. The selection mechanism (refer to the section "Page Frame Selection") is such that because the PFT entry is in the HQ, the probability that this page frame is selected again for another paging operation before it had a chance to be used, is relatively small.

The other four queues are Q00, Q01, Q10, and Q11. To which of these queues a PFT entry is queued (if it does not belong to the HQ) depends on the setting of the reference and change bits in the storage key of the corresponding page frame. The PFT entries are queued as indicated in the following table.

Queue	Reference bit, Change bit combination
Q00	(0,0)
Q01	(0,1)
Q10	(1,0)
Q11	(1,1)

The Page Manager uses the Reset Reference Bit (RRB) instruction to determine the setting of the reference and change bits of a page frame. The setting of these bits before the instruction is executed is returned in the condition code and determines in which queue the PFT entry is to be placed. Meanwhile the reference bit is reset. Therefore PFT entries may be in Q10 or Q11 while the corresponding reference bit is zero. Because of this procedure the reference bit, change bit combination (0,1) can occur when the reference bit is reset by the RRB instruction.

The five queues are identified by five fullwords Q00, Q10, HQ, Q01, and Q11, each containing the address of a queue header. The queue headers are 8 bytes long and situated immediately after the PFT. They contain the displacements (relative to the begin of the PFT) of the first and the last element of the queue. See Figures 5.3 and 5.4.

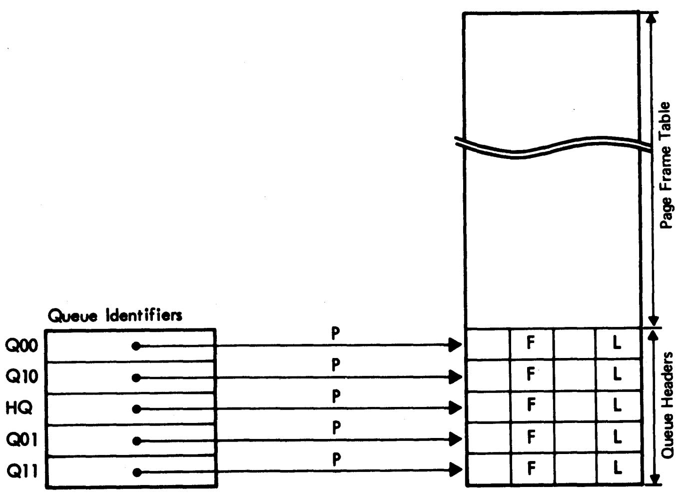


Figure 5.3. Selection Pool Queue Identifiers and Queue Headers

F: Pointer to first PFT entry in queue
L: Pointer to last PFT entry in queue
P: Initially these pointers are as drawn. They may be changed by queue switching.

Note: The Queue Headers are located directly behind the Page Frame Table. Each Queue Header is 8 bytes long. Bytes 2 and 3 contain a pointer to the first queue entry and bytes 6 and 7 contain a pointer to the last queue entry.

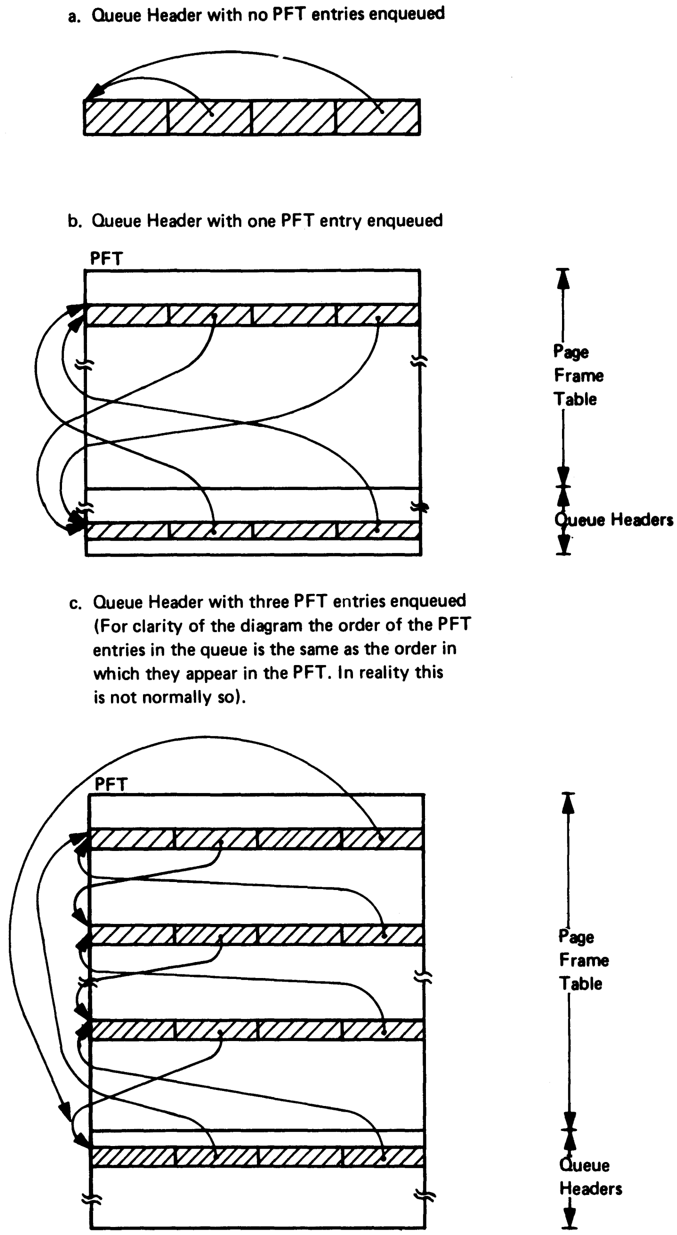


Figure 5.4. Selection Pool Queue Headers and Queue Elements

PAGE FRAME SELECTION

When a page frame is required for a paging operation, the Page Manager uses the following selection procedure.

1. The protection key of the first page frame in Q00 is examined, using the Reset Reference Bit (RRB) instruction. The reference bit is reset and the condition code indicates the status of the reference and change bits before the intruction was executed. If the combination (0,0), unreferenced and unchanged, is found, this page frame is selected. If the combination was not (0,0), the PFT entry is removed from Q00 and appended to Q10 or Q11, depending on the value of the combination. The combination (0,1) cannot occur in this situation. The process described above is repeated, starting with examining the protection key of the page frame which is now first in Q00. The procedure continues until a selectable page frame is found or Q00 is empty.

2. If Q00 is empty, or is emptied by the search, Q01 is processed similarly. If the first page frame in this queue has a combination of (0,1) before the reference bit is reset by RRB, it is selected. The combinations (0,0) and (1,0) cannot occur in this situation. If the value (1,1) is found, the PFT entry is removed from Q01 and appended to Q11. The now first page frame in Q01 is examined, etc.

3. If Q01 is empty, or becomes empty by the search, a test is made whether the selection pool is empty (remaining three queues, Q10, Q11, and HQ empty). If it is, exit is taken to attempt to free page frames (DEQUX routine). If the selection pool is not empty, the queues are switched in the following way:

   Q10 becomes Q00,
   HQ becomes Q10,
   Q01 (empty) becomes HQ,
   Q11 becomes Q01,
   Q00 (empty) becomes Q11.

This switch is achieved by changing the pointers in the queue identifiers. See Figure 5.5. After the switch the search is continued by starting with step 1 again.

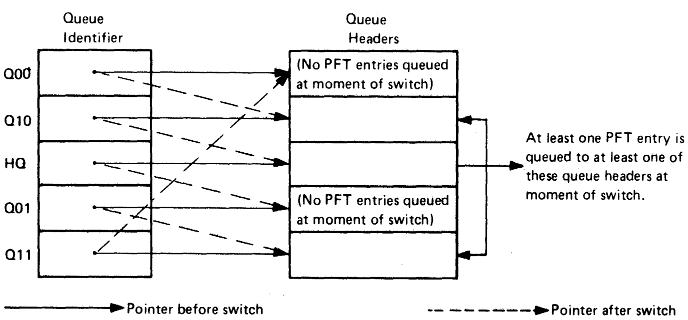


Figure 5.5. Selection Pool Queue Switching

PAGE HANDLING ROUTINES

The following conditions result in some form of page movement or reassignment of page frames and may require activity by the page manager system task:

- Page Fault
- GETREAL request
- TFIX request
- PFIX request
- PAGEIN request

The requests that require activity of the page manager system task (always the case for page fault and GETREAL) are queued in the page queue (PGQU) and handled on a FIFO (First-In/First-Out) basis by this task. The page queue (see Figure 5.6) has 16 four-byte entries.

However, the page manager system task is not activated for the following requests:

- FREEREAL request
- TFREE request
- PFREE request.
- RELPAG/FCEPGOUT request

In the following sections the handling of those requests which may require activation of the page manager system task, is described first.

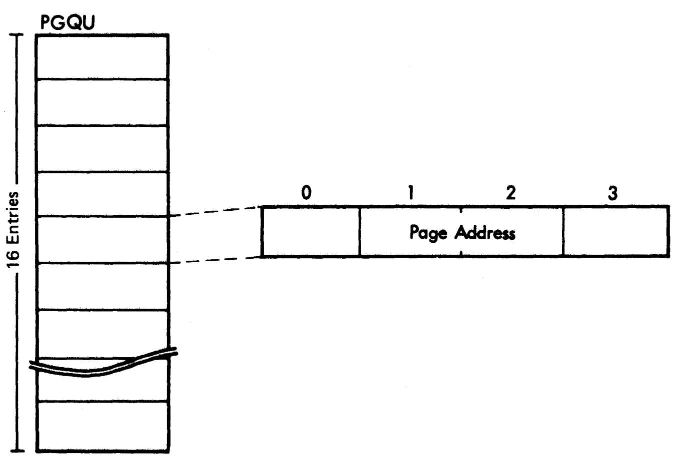


Figure 5.6. Page Queue (PGQU)

Bytes:

0:	Bits 0-3:	Bits 0-3 of PIK or TIK (if AP supported) of user task	
	4-7:	B'0000'	Page Translation Exception
		B'1000'	Dummy Request
		B'0100'	PFIX Request
		B'0010'	TFIX Request
		B'0001'	GETREAL Request

1-2: Leftmost 16 bits of the address of the page (which is also the address of the page frame, if GETREAL) to be handled, the remaining 8 bits of the address are assumed to be zero

3: Task identifier (PIK or TIK) of user task or, task identifier (select byte in STID field) of system task

Note: Label PGQU identifies the first byte of the table.

Bytes 24-27 (X'18' - X'1B') of the RASLINK area contain the address of the Page Queue Table.

PAGE FAULT

A page fault is always enqueued to the page queue and the Page Manager System task must do the following when handling it:

- Select a page frame for the requested page (see "Page Frame Selection").
- If the selected page frame is in use, invalidate the page table entry of the page that occupies the page frame (set bit 13 to 1) and insert its storage key into bits 8-11 of the entry.
- If the selected page frame is in use, save the contents to the page data set. This is only done when the contents of the page frame differ from the copy on the page data set (change bit in storage key is 1).
- Read the requested page from the page data set. This is only done if bit 15 (used bit) in the page table entry of the requested page is 0. Otherwise the page frame is cleared to zero.
- Set the storage key of the new page into the page frame, with the reference and change bits equal to zero.
- Validate the page table entry of the new page (set bit 13 to 0) and insert the leftmost 13 bits of the page frame address into bits 0-12.
- Insert the page number of the new page (its virtual address divided by 2048) into bytes 4 and 5 of the page frame table entry of the selected page frame.
- Remove the selected page frame table entry from its queue and append it to the hold queue (HQ).

PAGE FAULT HANDLING OVERLAP

Programs that execute in virtual mode and perform their own multitasking, can make use of the page fault handling overlap facility. For this PHO=YES must be specified on the SUPVR macro at system generation. This option gives the user the opportunity to control the PGQU entry for the page fault caused by its own task. This is done by a user written page fault appendage routine.

Whenever a page fault occurs in a system with this option page management first checks if a page fault appendage has been initiated for the task.

If the task has an appendage, control is first passed to that appendage, unless the task is using a supervisor service or the LTA. The appendage routine either returns a suitable PGQU entry which is then enqueud in the PGQU, or it gives an indication that a page fault is already pending for that task. The task causing the page fault is not deactivated.

If the page fault was caused by a supervisor service or logical transient, no overlap is performed. If another page fault is pending for the task, the page fault information is saved in PFATAB (see Figure 4.17) and the task is deactivated. Exit is taken to task selection.

If no other page fault is pending, the page fault is handled like any normal page fault condition and the task is deactivated.

When a page fault has been handled for a task having a page fault appendage, the page fault request is not immediately dequeued, but the appendage is entered again to see if there are any more page faults to be processed. If so, the page fault information returned from the appendage is put in the page queue entry of the page fault which has just been handled (first entry in PGQU). This page fault will be the next one handled by the page manager.

GETREAL REQUEST

The GETREAL routine is called by the SVC 55 (request for SDAID buffer or PDAID alternate area) and the SVC 58 (if initialization of a real partition is requested) routines, to reserve an area of real storage. See also the descriptions of these SVCs in "Supervisor Call Interrupt" in "Interrupt Processors".

On entry, register 2 and 3 contain the begin and end addresses of the requested area. All PFT entries of the page frames in this area are posted unfixable (bit 15 on) and the TFIX-counter in each PFT entry (bits 0-10) is checked for a zero value (page is not temporarily fixed). If any page frame is found to be TFIXed, the requesting task is set to common bound (X'87' in PIB). The handling of the GETREAL request is continued when none of the requested page frames is TFIXed.

For each page frame in turn a page queue entry is built and enqueued. Byte 0, bit 7 is posted in the page queue entry to indicate to the Page Manager system task that it is for a GETREAL request. When the Page Manager system task processes this entry, it must take the following action:

- If the page frame is in use, invalidate the page table entry of the page that occupies the page frame (set bit 13 to 1) and insert its storage key into bits 8-11 of the entry.
- If the page frame is in use, save the contents to the page data set. This is only done when the contents of the page frame differ from the copy on the page data set (change bit in storage key is 1).
- Remove the selected page frame table entry from its selection pool queue and post it as not belonging to the selection pool (bit 14 in page table entry on).
- Clear the page frame to zeros.
- Set the storage key of the new page into the page frame, with the reference and change bits equal to zero.
- Validate the page table entry of the new page (set bit 13 to 0) and insert the leftmost 13 bits of the page frame address into bits 0-12.
- Insert the page number of the new page (its virtual address divided by 2048) into bytes 4 and 5 of the page frame table entry of the selected page frame.

When a page frame is found to be unusable because of a hardware error (bit 13 in its PFT entry on), the following action is taken. No area is allocated when this condition is found in the first page frame. If the page frame in error is not the first, the allocated area ends at the start address of the failing page frame. The NF bit (bit 15) in the PFT entries of the remaining page frames is reset.

TFIX REQUEST

The TFIX routine is called by routines that want a page to be fixed temporarily (TFIXed) in real storage. It is called, for instance, by CCW translation routines, the Fetch routine (virtual load), and the SVC 44 routine. Temporarily fixing of a page means fixing it in a page frame for the duration of an I/O operation. A virtual address is passed to the TFIX routine in register 2. The page containing this address is to be TFIXed.

If the page is already in real storage, the request is handled by the DTFIX routine without enqueueing it to the page queue when one of the following conditions is valid:

- the page is already TFIXed or PFIXed in the frame in which it is located. This is so when bit 14 of the Page Frame Table (PFT) entry is on (page frame does not belong to the selection pool).
- the page is not yet TFIXed or PFIXed, and bits 12 and 15 of the PFT entry are zero. This indicates that no PFIX request is pending for the page (into a page frame of its associated real partition), and that no PFIX or GETREAL request is pending for the page frame. When this situation exists, the page is TFIXed in the page frame in which it is located.
- the page is not TFIXed or PFIXed. Bit 12 of the PFT entry is zero. However, bit 15 is on. In this situation it is not possible to TFIX the page in the page frame in which it happens to be located. If another page frame can be found, in which the page may be TFIXed, the contents of the page frames are swapped, and the page is TFIXed in the newly found page frame. To find such a page frame, two scan mechanisms are employed:

    a. Q00 is scanned for an unused page frame.
    b. If no unused page frame is found in Q00, the whole PFT is scanned for an entry which has bits 8-15 equal to zero.

When a page is TFIXed, the TFIX counter in the PFT entry is incremented by 1, and, in the second and third cases above, the page frame is removed from the selection pool.

If the page to be TFIXed is not in real storage, or if it is, but none of the above mentioned conditions is met, the request must be queued to the page queue. The request is then handled by the Page Manager system task.

A page queue entry is built and enqueued. Byte 0, bit 6 of the page queue entry is posted to indicate that this concerns a TFIX request.

When the Page Manager system task processes this entry it takes the following action:

1. If the page is in real storage another attempt is made to TFIX it by calling the routine DTFIX (see above). When it is not possible to honour the request now, exit is taken to the DEQUX routine to attempt to free page frames.

2. If the page is not in real storage, the request is first handled as a page fault (refer to the section "Handling of a Page Fault"). When the page is paged in, the action described in item 1 is taken.

PFIX REQUEST

A PFIX request may come from the following sources:

- user task, or
- the RSTRT statement processor.

A PFIX request may be issued for a number of pages but the PFIX routines handle one page at the time. The page manager system task is <u>not</u> activated if the desired page is in real storage and the following conditions are met:

- The page frame is in the correct real partition. In that case, it is only necessary to increase the PFIX counter by one and remove the page frame from the selection pool if it has not already been removed.

- The page frame is not in the correct real partition but a page frame in the real partition is available for PFIXing. The two pages are then exchanged.

If neither of the above conditions is true it is necessary to enqueue the request to PGQU for handling by the page manager system task. Before the request is enqueued, an attempt is made to find a page frame in the selection pool for the PFIX request. If there is a page frame in the real partition, which is in the selection pool, the NF bit (bit 15) in the PFT entry is posted (PFIX request pending for page frame). The address of this PFT entry is saved in the partition's entry in the FIXWTAB (see Figure 4.13) and the request is enqueued to the page queue (see also Figure 3.6, part 2). When the page manager system task handles the PFIX request, the requested page is brought into real storage using the normal selection procedure and the page is exchanged with the contents of the page frame whose PFT entry address was saved in FIXWTAB.

If there are no page frames available because they are all PFIXed, all of the pages which have been PFIXed for this request are freed and a return code is passed to the requesting task indicating that it is impossible to carry out the request at this time. If, however, there are not enough page frames because some of them are TFIXed, the request can still be carried out. All of the page frames in the real partition are marked not temporarily fixable (bit 15 of the PFT entry is set to 1) and the requesting task is put into the wait state until a page frame is freed by TFREE (see handling TFREE requests).

Handling PFIX requests for the RSTRTR processor presents special problems because each PFIXed page must be returned to the same page frame where it was when the program was checkpointed. When a page is PFIXed by the RSTRT processor, not only the page address is passed but also the page frame address and the value of the PFIX counter (see Figure 4.14). If the page is not in real storage, the address of the reserved page frame is placed in the FIXWTAB entry for that task and the PFIX counter for that page frame is set to one less than its value at the checkpoint and the request is queued for handling by the page manager.

PAGEIN REQUEST

A valid PAGEIN request is handled by the PAGEIN task, which is activated when the SVC 87 routine has received such a request. The task's dispatching priority is higher than that of the FETCH task, but lower than that of the page manager system task. The PAGEIN task runs asynchronously with the requesting user task.

For a page in real storage, the task determines (by looking up the corresponding PFT entry) whether this page is fixed.

- If the page is fixed, the request for that page is ignored.

- If the page is not fixed, its reference bit is set to 0 and the associated page frame is added to the end of frame selection queue HQ.

For a page not in real storage, the PAGEIN task uses the ENQU routine (see Chart 13.1) to have this page enqueued on queue PGQU. The request is then handled like a page-in request that resulted from a normal page fault; however, no exit is taken to a private routine that may be specified in a SETPFA macro in the program which issued the PAGEIN request.

The PAGEIN task detects the following error conditions and takes the actions indicated:

- If a page is outside the partition in which the requesting program executes, the request for that page is ignored.
- If an area specification contains a negative length, the request for that area is ignored.

The task posts an ECB (if one is specified) as shown for SVC 87 in "Supervisor Call Interrupt" under "Interrupt Processors". The ECBs address is obtained from the currently processed PAGETAB entry.

Whenever a task is terminated, the scan routine SCANPGT scans table PAGETAB and deletes all entries that carry the task's TIK. If the PAGEIN task is processing a PAGEIN request of a task which is being terminated, the PAGEIN task discontinues the processing of that PAGEIN request.

FREEREAL, FREE, AND PFREE REQUESTS

For handling of FREEREAL and PFREE requests the page manager system task is not activated. See SVC 54 (FREEREAL) and SVC 68 (PFREE) in "Supervisor Call Interrupt" in "Interrupt Processors" for descriptions of the actions taken.

The TFREE routine is called by routines such as Channel Program Translation, SVC 44, and the Fetch Routine (if virtual load) to release TFIXed pages. The page manager system task is not activated. Upon entry to TFREE, register 3 contains the number of the page frame to be TFREEed. TFREE reduces the TFIX counter by one.

If the page frame is not PFIXed and if the TFIX counter has been reduced to zero, the page frame is placed in the selection pool. If the TFREE requestor was the SUPVR system task (fetch), the PFT entry is queued to the top of Q00 or Q01, depending on the change bit. If it is not fetch the PFT entry is queued to the bottom of the HQ. The reference bit is reset. Any tasks waiting for freed page frames are allowed to be selected (X'83' in TRTFIX and TRTPFG). Depending on the bits 12 and 15 in the PFT entry additional action may be taken, as indicated in Figure 5.7.

RELPAG AND FCEPGOUT REQUESTS

The page manager system task is not activated when a user task issues a RELPAG or a FCEPGOUT request. For a description of the actions taken, see SVC 85 and SVC 86, respectively in "Supervisor Call Interrupt" under "Interrupt Processors".

<u>PARTITION DEACTIVATION AND REACTIVATION (LOAD LEVELER)</u>

Processing in one or more partitions may have to be temporarily stopped if paging activity rises to the point where little effective work is being accomplished (this condition is commonly called thrashing). Once paging activity drops to an acceptable level, the deactivated partition(s) can be reactivated.

Bit Combination			Action
SP bit 14	NFF bit 12	NF bit 15	
1	0 or 1	0 or 1	Return to caller.
0	0	0	Return to caller.
0	1	0	The page is to be PFIXed in another page frame. The contents of the two page frames are exchanged and the page is PFIXed in the other page frame. The task waiting for this page to be PFIXed is posted ready to run.
0	0	1	Post the task waiting for this page frame (needed for GETREAL or PFIX) ready to run. If page frame is needed for a PFIX, reset bit 15 in the other PFT entries associated with the real partition.
0	1	1	Perform processing as for both the above cases.

Figure 5.7. Actions after TFREE

PARTITION DEACTIVATION

To count the number of page-in requests, two counters are maintained. One of these counters, PIDCTR, is used by the routines which determine whether a partition must be deactivated. The other, PIRCTR, is used by the routines which determine whether a deactivated partition may be reactivated. The counters are increased by one, every time the page manager determines that a page is to be read from the page data set (page-in).

The routine which updates the page-in counters, CPGIN, checks whether PIDCTR has reached a prespecified value, which is contained in the location NPI. As long as the prespecified value is not reached, no further action is taken and control is returned to the PMR routine to continue handling the currently pending paging requests.

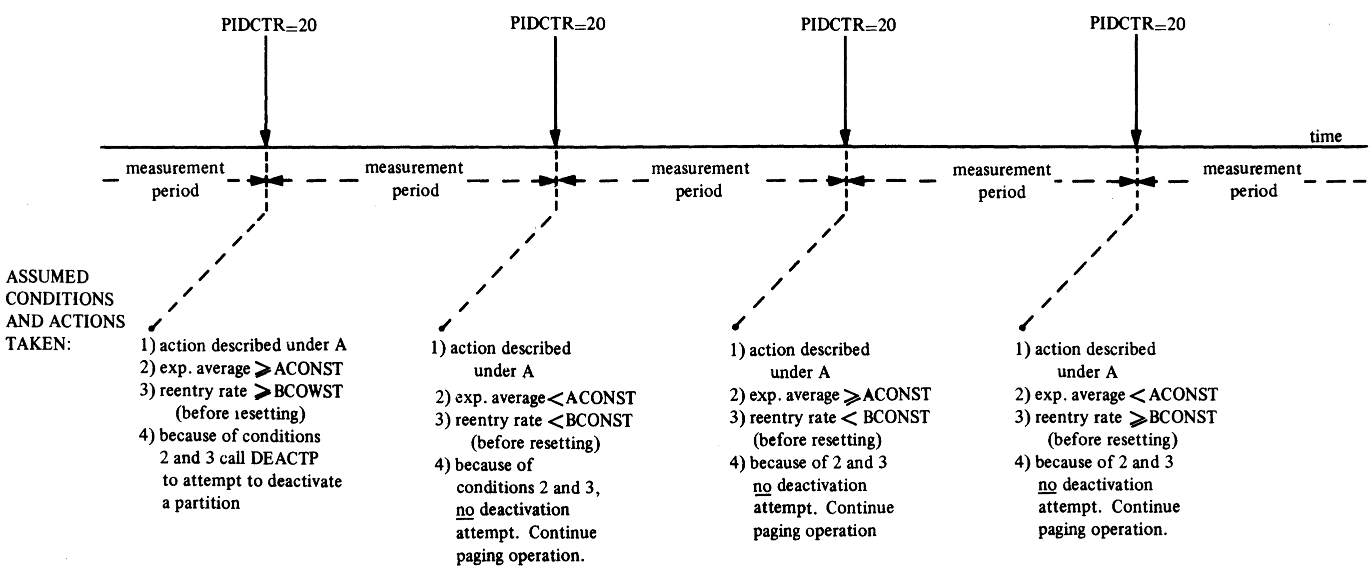
If the prespecified value, NPI, is reached, PIDCTR is reset to zero and the routine DEACT is called to measure the paging activity. The paging activity is therefore measured at specific points in time (PIDCTR=NPI). The time interval between two consecutive measurement moments is called a measurement period.

Measuring paging activity means establishing and checking two values:

- the exponential average of page-ins per second, which gives an indication of the page-in activity since IPL. This quantity is calculated giving more weight to recent page-ins and less to page-ins that occurred in the past.
- the reentry rate, which is equal to the number of page-ins of pages paged out earlier in the same measurement period.

The routine DEACT calculates the exponential average (see below) and checks the exponential average and the reentry rate against two prespecified values, located in ACONST and BCONST. If the exponential average and the reentry rate are both greater or equal these prespecified values, DEACTP is called to attempt to deactivate a partition. See Figure 5.8.

Now follows a description of the exponential average of page-ins per second and the reentry rate, followed by a description of the actions taken by DEACTP if both have reached their prespecified maximum values.



A: reset PIDCTR to zero, calculate exponential average, reset reentry rate counter to zero, and clear reentry rate table.

Note: It is assumed that NPI=20.

Figure 5.8. Criteria for Partition Deactivation

Exponential Average of Page-Ins per Second (for Deactivation)

The exponential average is a value which is calculated periodically (every time NPI page-ins have occurred). To calculate the new exponential average, the old exponential average is used:

new exp. av. = 1/2 (old exp. av. + NPI/measurement period)

(see note at the end of this chapter).

The measurement period is the time elapsed between the moment of the calculation of the old exponential average and the moment of the calculation of the new exponential average. In other words, it is the time interval between the time that PIDCTR reached the value NPI (and was reset to zero) and the moment that it reaches this value again.

When NPI page-ins have occurred for the first time after IPL, the old exponential average does not exist. It is therefore set equal to NPI-measurement period, and then the above formula is applied.

It should be noted that there also exists an exponential average of page-ins per second for reactivation, which is explained in the section "Partition Reactivation".

Reentry Rate

The reentry rate is equal to the number of page-ins of pages that were paged out earlier in the same measurement period. To establish this value, a reentry rate counter RRCTR is maintained. This counter is set to zero at the start of every measurement period. If the page manager determines that a page is to be paged in, while it was paged out earlier in the same measurement period, it increases the reentry rate counter by one. This procedure makes use of the reentry rate table (RTAB), which is a bit string containing a bit for every page in the virtual storage. At the beginning of a measurement period all bits of RTAB are set to zero. When the page manager selects a page frame for a page-in operation, while this page frame is already occupied by another page, the bit in RTAB for the latter page is posted.

When the page manager determines that a page is to be read in from the page data set, the bit in RTAB corresponding to the page is tested. If this bit is on, reentry is detected and the reentry rate counter is increased by one.

Selection of a Partition for Deactivation

The DEACTP routine attempts to deactivate a virtual partition, if more than one virtual partition is active. If any task seized the system, the deactivation request is ignored.

To determine which partition is to be deactivated, the load leveler TRTMASK bytes (see Figure 5.9) are copied to a workarea (WAITFLD). Each partition has its own TRTMASK byte. If it is zero, the partition is already deactivated. If a partition uses the LTA, DEACTP sets its copied TRTMASK byte to zero, to avoid deactivation of this partition. The partition in which POWER/VS resides is never deactivated.

Figure 5.9 shows an example of the selection procedure for a partition to be deactivated. When a partition is selected, its PIB is checked to determine whether it is active (and not stopped), whether it runs in virtual mode, and whether it is <u>not</u> in the process of being canceled. If these conditions are not simultaneously met, the search is continued to find the next higher priority partition that can be deactivated. The search is also continued when the selected partition has a task which uses PHO and has a page queue entry enqueued.

The TRTMASK byte for the selected partition is set to zero. The entry in the system communication region indicating the number of active virtual partitions is decreased by one. The page table entries of the partition are scanned for any pages that are in real storage. Selection pool page frames containing such pages are queued to Q00 or Q01, depending on the setting of the change bit. Any fixed pages of the partition are left as they are. The reactivation count for the partition is initialized. This contains the number of not fixed pages that the partition has in real storage at the moment of deactivation. It is used when the partition is about to be reactivated.

The tasks of the partition are handled in turn. They are, depending on their status, deactivated (PIB posted load leveler bound) or deactivation is delayed (posted in PIB DAT Flag). Any page queue entries for tasks that are being deactivated are deleted.

If all tasks have been handled, return is to the RMR routine to continue the paging operation, when no page queue entry was deleted. If one or more page queue entries were deleted, the pending paging operation is not continued, but exit is taken to the routine DEQUXE. This routine

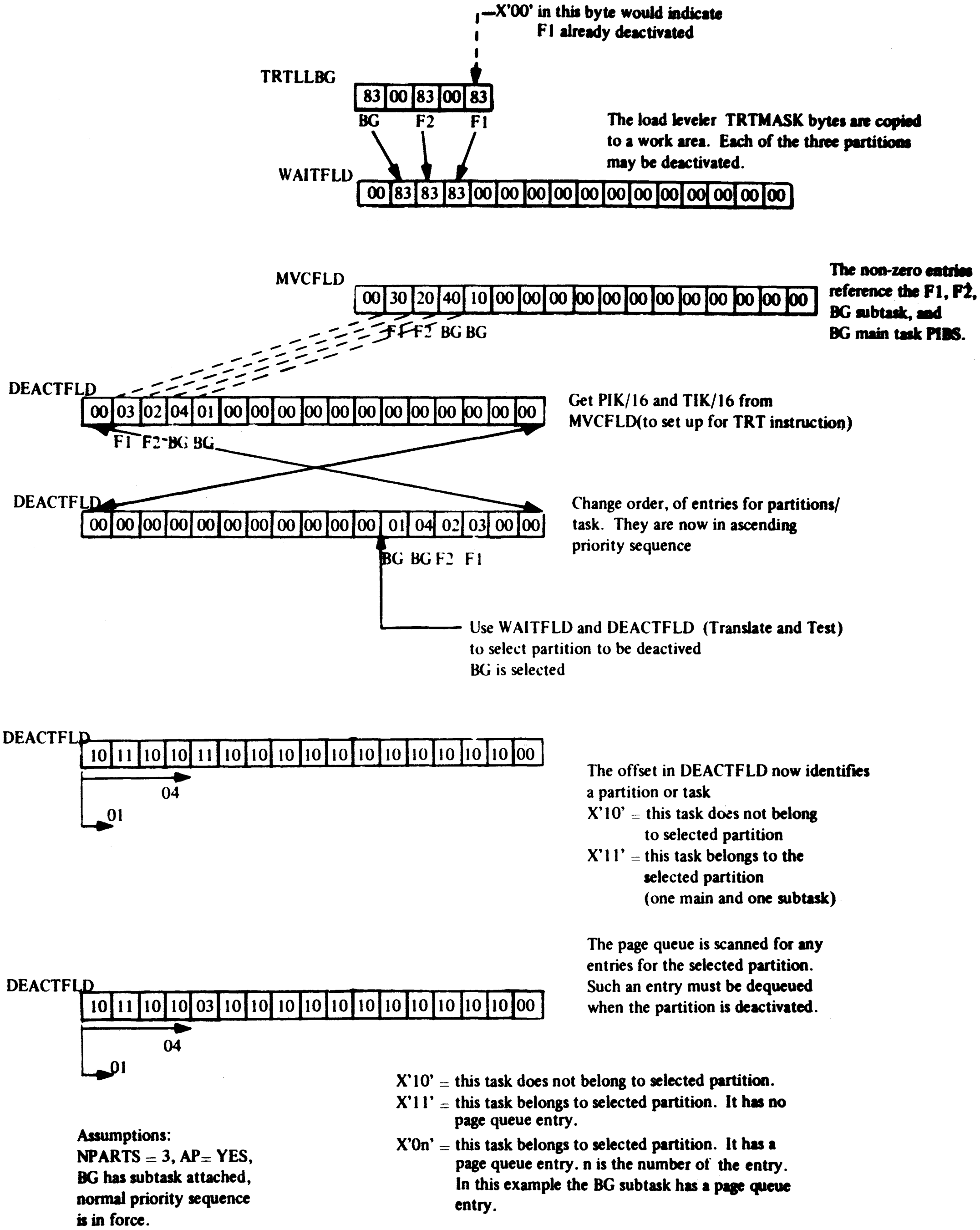


Figure 5.9. Example of Usage of Workareas DEACTFLD and WAITFLD when Selecting a Partition for Deactivation

exits to task selection if the page queue is empty, or passes control to the PMR routine to handle the first page queue entry.

Handling of a task by the deactivation routines depends on the status of the task, as the following list shows.

- If the task has no page queue entry, is not using a supervisor service, and is ready to run (X'83' in PIB), then the task is deactivated.

- If the above conditions apply, but the task is not ready to run, then deactivation is delayed. Delayed deactivation is posted in the PIB. This will result in deactivation of the task when it is selected by task selection a next time.

- Deactivation is also delayed when the task is using a supervisor service in a gated routine.

- If the task has a page queue entry associated with a fetch request (SUPVR system task), the fetch routine is reset. The task's system save area is initialized to allow restart of the fetch procedure. All page frames that were TFIXed for this fetch request are freed. The task is then deactivated.

If the task is using a reenterable supervisor service routine, the following situations may exist.

- CCW translation is in progress for the task. The task's system save area is initialized to allow restart of the CCW translation procedure. The associated translation control block is reset and the copy and IDAL blocks are freed. The task is then deactivated.

- SVC 44 is in progress. The task's system save area is modified to allow restart of this routine. Any pages TFIXed for this request are freed. The task is then deactivated.

- A pending GETREAL request (from PDAID or SDAID) for the task is stopped if the task is posted common bound (X'87' in PIB), or if the task has a page queue entry. The task is then deactivated.

- A pending PFIX request is reset if the task is posted common bound, or if the task has a page queue entry. Otherwise, deactivation is delayed. The PFIX request is reset by modifying the task's system save area to allow entry to a routine that will PFREE already PFIXed pages and set up the task to reissue the SVC. Deactivation is delayed.

PARTITION REACTIVATION

When task selection finds that no system or user task is ready to run, the system enters an enabled wait state. Just before the wait PSW is loaded, the reactivation routines are called to reactivate a deactivated partition (there may not be a deactivated partition).

Unconditional reactivation (see below) is performed when there are one or more deactivated partitions and one of the following conditions is fulfilled.

- There is no active virtual partition.

- No I/O requests are queued to any PUB, other than requests for console or TP devices, or requests with an error condition.

When the reactivation routines are entered (at REACTC) for <u>conditional</u> reactivation, a number of conditions must be satisfied before an actual attempt is made to reactivate the deactivated partition with the highest dispatching priority. If these conditions are not simultaneously fulfilled, control is returned to caller and the wait state is entered. When these routines are entered (at REACTUC) for <u>un</u>conditional reactivation, reactivation is attempted under all circumstances.

The conditions for conditional reactivation are:

- The time elapsed since previous entrance to these routines must not be less than a prespecified value, contained in location MINTIME.

- The page manager system task must not be active.

- The exponential average of page-ins per second (for reactivation) must not be greater than a prespecified value, contained in CCONST (paging rate must be relatively low).

- The number of page frames in the selection pool must not be less than the reactivation count of the partition to be reactivated. The reactivation count is set up when the partition is deactivated and is equal to the number of unfixed pages of the partition in real storage at the moment of deactivation.

Exponential Average of Page-Ins per Second (for Reactivation)

The exponential average of page-ins per second for reactivation is calculated for both conditional and unconditional requests. The calculation is similar to the calculation of the exponential average for deactivation:

new exp. av. = 1/2 (old exp. av. +PIRCTR/time interval)

(see note at the end of this chapter).

Note that two other quantities are used. PIRCTR is the page-in counter for reactivation. It is reset to zero after the calculation of the new exponential average, and is increased by one every time a page-in occurs. Time interval is the elapsed time between the previous call of the reactivation routines and this call.

Figure 5.10 shows an example of the procedure followed to find the deactivated partition with the highest dispatching priority. When the partition is found, its reactivation count is checked, if the reactivation request is conditional. If it is too high, exit is taken to task selection without reactivating the partition. If the reactivation count is not too high, or if the request is unconditional, the partition is reactivated. The TRTMASK byte for the partition is posted X'83' (was X'00'), and the entry in the system communications region indicating the number of active virtual partitions is increased by one. Exit is taken to task selection.

Parameters

The following is a summary of the labels of the deactivation/reactivation parameters.

- ACONST is the upper limit for the exponential average of page-ins per second for deactivation. As long as the exponential average is less than this value, no deactivation is attempted.
- BCONST is the upper limit for the reentry rate for any one measurement period. When the reentry rate is below this value, no deactivation is attempted.

An attempt is made to deactivate one partition when ACONST and BCONST are simultaneously equalled or exceeded.

- CCONST is the upper limit for the exponential average of page-ins per second for reactivation. If the exponential average is greater than this value, no conditional reactivation attempt is to be honoured.
- MINTIME is the minimum amount of time which must have passed since the last previous reactivation request to allow conditional reactivation.
- NPI is the number of page-ins which must have occurred before the deactivation routines are entered. The counter is then set to zero.

Note: The calculation of the exponential averages is based on the general formula for calculation of the exponential average of the values x1, x2,.......xk:

Ak=CXk + (1-C)Ak-1,0<C<1,A0=X0

The weighting factor, C, has been set to 1/2.

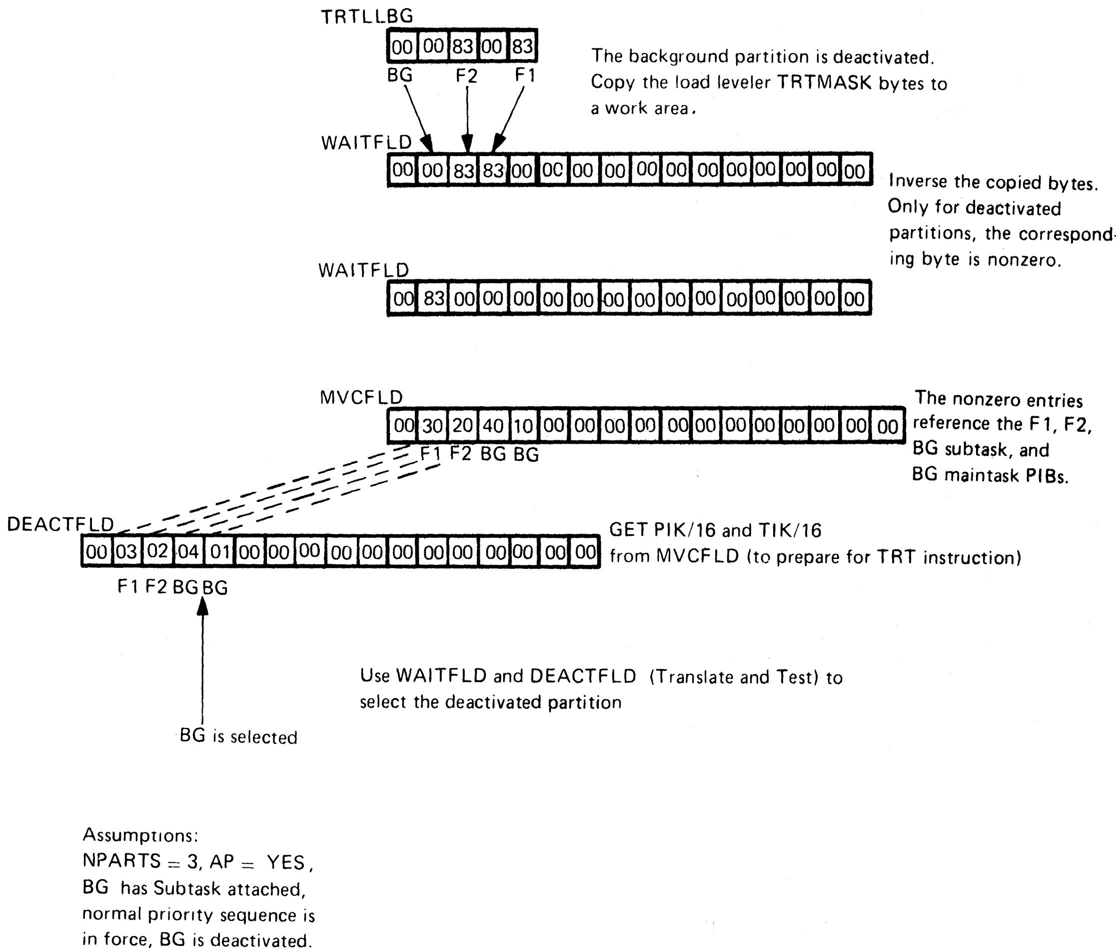


Figure 5.10. Example of Usage of Workareas DEACTFLD and WAITFLD when Selecting a Partition for Reactivation

Physical IOCS is that portion of the resident supervisor that:

- Builds a schedule of I/O operations for all devices on the system (CHANQ Table). Refer to Channel Scheduler on Chart 06.3.
- Starts the actual I/O operations on a device (SIO). Refer to Chart 06.4.
- Schedules the starting of all I/O operations and monitors all events associated with I/O. Refer to I/O Interrupt Handler on Chart 06.1.
- Performs error recovery procedures (ERP). Refer to Unit Check, ERP Exits, and Resident Disk Error Recovery on Charts 09.1 - 11.2.
- Calculates the sector value for DASD devices with the RPS feature (3330 and 3340).

Figures 6.20 through 6.24 illustrate: Command Control Block (CCB), Channel Command Word (CCW), Program Status Word (PSW), Channel Address Word (CAW), and Channel Status Word (CSW). See Figure 6.25 for CSW testing.

CHANNEL SCHEDULER

When a channel program is to be executed for a user, the channel scheduler routine first checks to see if a channel queue entry is available.

If the channel queue is full, an interruptable wait loop is forced (re-SVC) until a channel queue entry is freed by an I/O interruption.

Note: The occurrence of this interruptable wait loop is an indication that the channel queue is too short; low performance is the result. The situation could have been prevented, when the supervisor was generated, by specifying a higher value in the CHANQ parameter of the IOTAB macro.

If an entry is available in the channel queue, the I/O request is queued at the end of the chain for the specified device.

If this is the only request for the device, and the channel and device are both available, I/O is started for this request.

If the I/O request is from a virtual-mode program, the channel program must be copied into blocks in the supervisor and all virtual addresses must be translated into the correct real addresses before the I/O operation can be carried out. In addition, all pages containing I/O areas must be fixed in real storage to prevent them from being paged out during the I/O operation.

The channel scheduler does any further processing for the I/O request using the copied and translated program and not the channel program as it stands in the user program.

If any condition prevents the start of the operation, the request is left in the channel queue and return is made to the user. Examples of some conditions that prevent the start of I/O are:

- The request is not the first in the queue for the device.
- The device is busy, or in error, or has an operator intervention pending.
- The channel is busy for another device on the same selector channel and the requested device is not switchable.
- The device is switchable on two selector channels but both channels are busy.

As soon as the condition is cleared by an I/O interruption, this new request is started.

If the I/O request was made for a CRT device, control is given to the CRT routines to translate the channel programs to CRT channel programs. Control is then returned to the channel scheduler to handle the request.

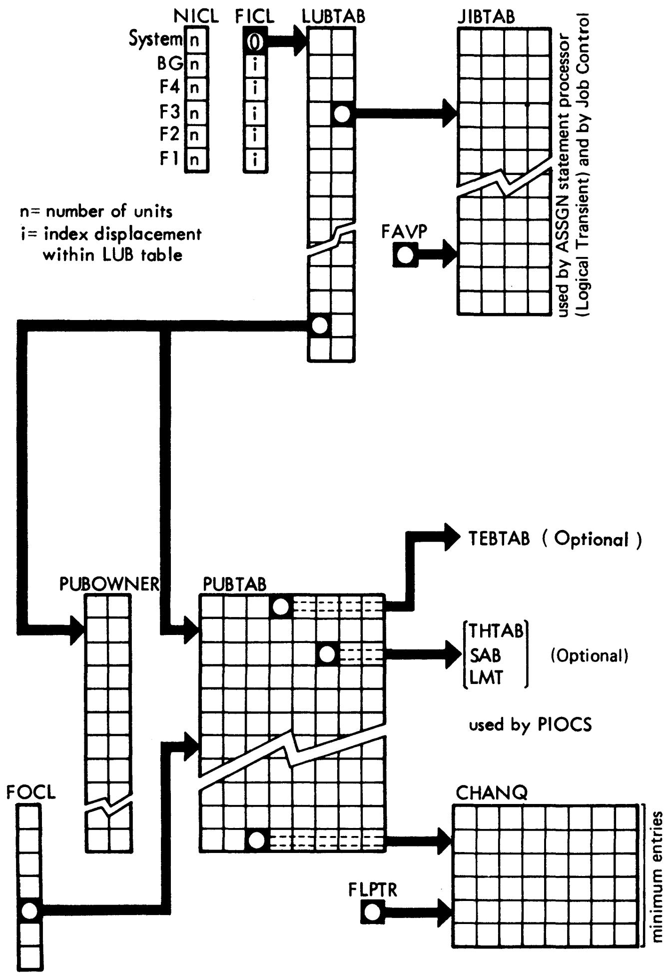


Figure 6.1. I/O Table Interrelaticnships (Part 1 of 3)

Key	Explanation
NICL (Number in Class)	Byte 0 contains the number of system class LUBs. The remaining bytes contain the number of programmer class LUBs for each partition. The total number of bytes is one more than the number of partitions supported.
FICL (First in Class)	Byte 0 points to the first system class LUB in the LUB table (LUBTAB). This is always the first entry in the LUB table. The remaining bytes point to the first programmer class LUBs in the LUB table partition areas. The total number of bytes is one more than the number of partitions supported.
LUBTAB (Logical Unit Block Table) (Figure 6.8)	Byte 0 of each entry is an index pointer to an entry in the PUB Table (PUBTAB) and to an entry in the PUB Ownership Table (PUBOWNER), or contains X'FF' if no logical unit is assigned. Byte 1 points to an entry in the JIB Table (JIBTAB) or contains X'FF'.
PUBTAB (Physical Unit Block Table) (Figure 6.3)	Bytes 0 and 1 of each entry contain the channel and unit address of the physical device. Byte 2 points to the entry in the channel queue (CHANQ) table or contains X'FF'. Byte 3 is a retry counter or, if the unit is a tape cartridge reader and the TEB=n parameter was included in the FOPT macro, it contains a pointer to the Tape Error Block table (TEBTAB) entry for the device. Byte 4 contains the device type code. Byte 5 is an index pointer to the entry in the Track-Hold Table, the seek-address block table, or the mode table (MODEL 125 only) when one of these options is active for the device. Otherwise this byte contains X'FF'. Byte 6 contains the channel scheduler flags and byte 7 the job control flags.
FOCL (First on Channel List)	Byte 0 points to the first PUBTAB entry for a device on channel 0. Byte 1 points to the first PUBTAB entry for a device on channel 1, and so on. X'FF' indicates that the associated channel is not supported.
PUBOWNER (Figure 6.4)	Byte 0 of each entry is reserved. Byte 1 identifies the partition that owns the corresponding PUB.
FAVP (First Available Pointer)	This one-byte pointer to the next available entry in the JIB Table (JIBTAB) is used by the ASSGN statement processor and by job control.
JIBTAB (Job Information Block Table (Figure 6.5)	Bytes 0 and 1 of each entry contain extent or LUB information used by job control and the ASSGN statement processor. Byte 2 is a flag byte. Byte 3 is a chain byte.

Figure 6.1. I/O Table Interrelationships (Part 2 of 3)

Key	Explanation
CHANQ (Channel Queue Table) (Figure 6.2)	Byte 0 in each entry points to the next entry in the queue for the same device (or the next free entry if in the free list), or it contains X'FF' if the entry is the last in a chain. Bytes 1, 2, and 3 contain the CCB address. Byte 4, if I/O was requested by a user program, contains the Partition Identification Key (PIK). If a system task requested I/O, then the zone field is all zeros and the numeric field contains the ID of the specific system task. Byte 5 contains a displacement index pointing to the LUBTAB entry related to the I/O requesr (absolute LUBTAB index). Byte 6 contains the relative LUBTAB index for system LUBs, or X'FF' for programmer LUBs. Byte 7 contains the displacement index of the PIBTAB entry for the task requesting I/O (TIK), or X'FF' if the channel queue entry is free.
FLPTR (Free List Pointer)	This one-byte pointer contains the displacement index of the next free entry in the channel queue table (CHANQ).
TEBTAB (Tape Error Block Table) (Figure 6.6)	One entry is built for each tape cartridge reader at supervisor generation time if the FOPT macro parameter TEB=n is included.
THTAB (Track-Hold Table) (Figure 6.34)	This table is built at supervisor generation time if the TRKHLD=n parameter is included in the FOPT macro. Byte 0 in each entry points to the next entry in the chain of requests for a track to be held on a specific DASD (or the next free entry if in the free list), or it contains X'FF' if the entry is the last in a chain. Bytes 1, 2, and 3 contain the CCB address. Bytes 4 through 9 contain the disk address (BBCCHH) of the held track. Byte 10 contains the key of the owning task, or all zeros when the entry is free. Byte 11 is a flag and counter byte: bit 0 is turned on when a task requests a track already held by another task, and the value in the low-order half-byte is incremented by one each time a task requests a hold on a track that it already holds itself. Note: when multiple holds by one task are effective, the value in the lower-order half-byte is one less than the actual number of holds.
SAB (Seek Address Block) (Figure 6.39)	This table is built at supervisor generation time if the SKSEP=n or YES parameter is included in the FOPT macro. Bytes 0 through 3 of each entry contain the current disk address (BCCH) for the device. Byte 4 contains X'FF' or points to the THTAB entry (refer to the section "Seek Separation").
LMT (Line Mode Table) (Figure 2.17)	This table is built at supervisor generation time when the TP=BTAM, or QTAM parameter is included in the SUPVR macro, and MODEL=115 or 125. An entry is built for each device for which the DVGEN macro includes the MODE=X'ssss' or X'ssssss' parameter. Each entry contains the actual mode setting for the device.

Figure 6.1. I/O Table Interrelationships (Part 3 of 3)

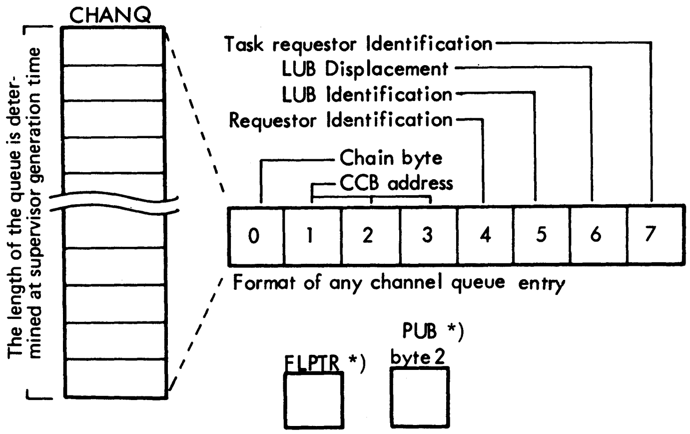


Figure 6.2. Channel Queue Table (CHANQ) and Entry

Bytes:

0: Contains the displacement within the channel queue of the next entry in the free list or in the list for a specific device, or X'FF' when it is the last entry in the free list or a device list.

1-3: Contains the CCB address for the specified device.

4: Contains a code identifying the task making the I/O request. This one byte entry indicates to which task the CCB belongs, and is in the form X'nk' where:

n = user storage protection key. (Attention or system task = 0, BG task = 1, FG tasks = 2-5 depending on the number of partitions).
k = 0 for Attention and all user tasks.

1 for RAS
2 for PMGR
3 for SUPVR nk=FF for any unused channel queue entries.
4 for CRT
5 for ERP
6 for PAGEIN

If the I/O request is from a subtask, this byte does not identify this subtask, but the partition.

5: Contains pointer (displacement index) to the entire LUB table identifying the logical unit making the I/O request. This is doubled to get the actual displacement into the full LUB table.

6: Contains X'FF' if the LUB is a programmer class, or the displacement within the partition LUB if it is a system class. (Not used by a non MPS Supervisor.)

7: Contains the displacement within the PIB table of the PIB of the task requesting I/O, or X'FF' when the channel queue entry is in the free list. (Not used by a non MPS Supervisor.)

*) Notes: FLPTR: The free list pointer contains the displacement within the channel queue of the first entry in the free list or X'FF' when the channel queue is full.

Byte 36 (X'24') of the System Communication Region (SYSCOM) contains the address of the Free List Pointer. Label FLPTR identifies the location of the pointer (1 byte).

PUB byte 2: The PUB channel queue pointer contains the displacement within the channel queue of the first entry for a specific device.

Bytes 37-39 (X'25' - X'27') of the System Communication Region (SYSCOM) contain the address of the Channel Queue table entry. Label CHANQ identifies the first byte of the table.

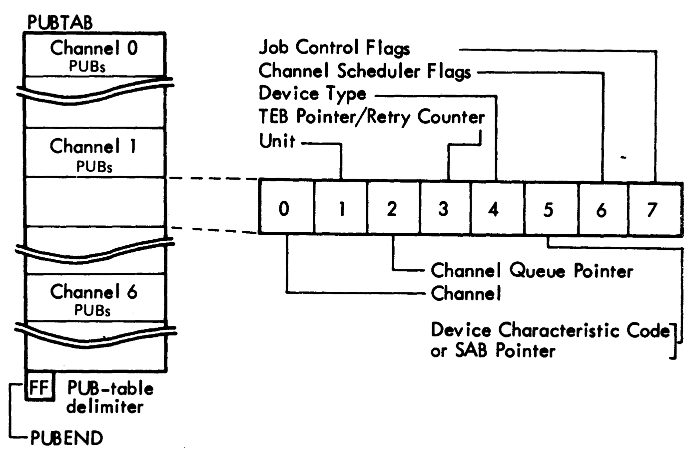


Figure 6.3. Physical Unit Block (PUB) Table

Bytes:

0: Channel number. (Hex 0-6, FF=NULL)

1: I/O device unit number

2: Hex 0, 1, 2, ... points to the first channel queue entry for this device

3: If device is a 2495 Tape Cartridge Reader and TEBs are specified, this byte is a TEB pointer. (Hex 1, 2, 3, ...). Otherwise, this byte is an ERP retry counter.

4: Device type code

5: SS of the MODE=parameter in the DVCGEN macro for tape unit. (See Section 2.)

For ICA line (Model 115, or 125), this byte contains the displacement index of the entry in the Line Mode Table (LMT). The address of the LMT is in SYSCOM.

For DASD with seek separation, this byte is used as the SAB Pointer. With Track Hold but not seek separation supported, this byte contains a pointer to the Track-Hold Table entry or X'FF' (with both SKSEP and TRKHLD specified, the Track-Hold pointer is found in the SAB entry).

For MICR type devices, this byte indicates which external interrupt line is in use.

For a 3704/3705 Communications Controller, this byte contains the type number of the Channel Adapter.

For MFCM or MFCU:

Bit 0: 1 = Repositioning required
1: 0 = SYSPCH temporarily assigned to hopper 1
1 = SYSPCH temporarily assigned to hopper 2
2: 0 = SYSIPT temporarily assigned to hopper 1
1 = SYSIPT temporarily assigned to hopper 2
3: 0 = SYSRDR temporarily assigned to hopper 1
1 = SYSRDR temporarily assigned to hopper 2
5: 0 = SYSPCH permanently assigned to hopper 1
1 = SYSPCH permanently assigned to hopper 2
6: 0 = SYSIPT permanently assigned to hopper 1
1 = SYSIPT permanently assigned to hopper 2
7: 0 = SYSRDR permanently assigned to hopper 1
1 = SYSRDR permanently assigned to hopper 2

6: Channel Scheduler Flags

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

7: Job Control Flags

Bit 0-4: Standard MODE assignment for 7-track tape (all ones if not tape, all zeros if device is down)

5: 1 = Device supports RPS

6-7: B'11' (both on) = Headqueue in progress*

B'01' = Headqueue requested*

*No I/O is started on a PUB or copied PUB that has both these switches on. If only bit 7 is on I/O can be started after seek separation.

Notes: A null entry is generated at supervisor generation time for each device to be supported by the supervisor. Then standard physical unit assignments are made to the PUB table. Physical unit assignments can also be made during IPL. PUBs are ordered by channel and priority within a channel.

An entry in the PUB Ownership Table is associated with each entry in the PUB Table, if the supervisor has been generated to support multiprogramming.

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

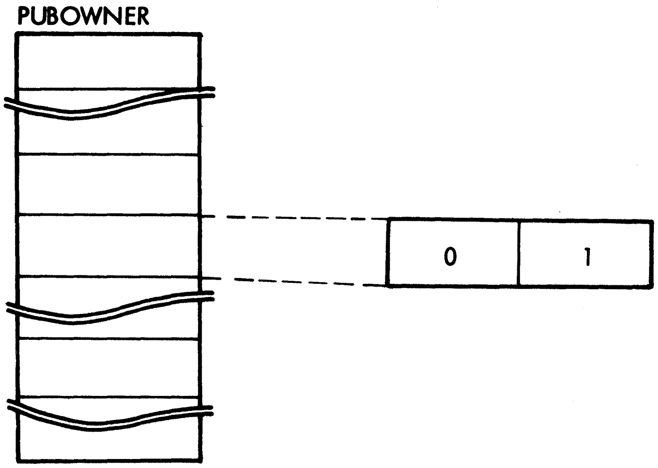


Figure 6.4. Physical Unit Block Ownership (PUBOWNER) Table

Bytes:

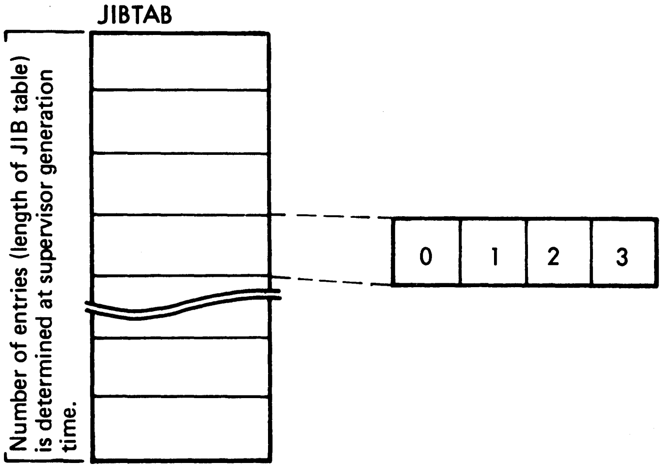
0: Bit 0 Reserved
   1 = 1 waiting for volume to be mounted
   2-7 Reserved

1: Identifies the partition that owns the PUB according to the following table:

Flag	Partition owning PUB if number of partitions is: 2	3	4	5
X'01'	BG	BG	BG	BG
X'02'	F1	F2	F3	F4
X'04'		F1	F2	F3
X'08'			F1	F2
X'10'				F1

Note: The number of entries in the PUB Ownership Table is equal to the number of entries in the PUB Table. Associated with each PUB entry is an entry in the PUB Ownership Table.

Bytes 120-123 (X'78' - X'7B') of the System Communication Region (SYSCOM) contain the address of the PUB Ownership Table. Label PUBOWNER identifies the first byte of the table.



Byte(s)	Description	
0-1 (Contents depends on the bit setting in byte 2)	Bit Setting Byte 2	Contents
	Bit 0=1 Stored standard assignment	LUB entry of stored standard assignment. (PUB and JIB pointer)
	Bit 1=1 Alternate assignment	Byte 0: PUB pointer Byte 1: X'00'
	Bit 2=1 2311/2314/2319 Extent	Byte 0: Cylinder lower limit Byte 1: Cylinder upper limit (Note 1)
	Bit 3=1 2321/3330/3340 Extent	For 2321: Lower limit (Cell or combined subcell and strip), or Upper limit (Cell or combined subcell and strip) For 3330 or 3340: Cylinder lower limit, or Cylinder upper limit (One cylinder number uses two bytes) (Note 2) (Note 3)
2	Meaning if bit = 1: Bit 0: Stored standard assignment 1: Alternate assignment 2: 2311/2314/2319 extent 3: 2321/3330/3340 extent 4: The alternate assignment indicated in bit 1 is permanent. This bit is also on when one of the extent indicators (bit 2 and bit 3) is on. 5: Catalogued procedures processing 6/7: Reserved	
3	Chain byte Contains the displacement index of the next JIB. X'FF' defines the end of the chain.	

Figure 6.5. Job Information Block (JIB) Table

Note 1: Only when file-protect on DASD.

Note 2: Two JIBs are required for a 2321/3330/3340 extent; one for lower limit and one for upper limit.

The lower limit defining JIB must be chained to the upper limit defining JIB.

For 2321, byte 1 of this type JIB contains the subcell number times 10 plus the strip number in binary.

Note 3: Extent information is supplied by the program initiator and logical IOCS open transient routines. The supervisor can then perform the file protect function for the specified file limits. File protection does not include supervisor and transient originated I/O.

Bytes 68-69 (X'44' - X'45') of the partition communication region contain the address of the JIB table entry. Label JIBTAB identifies the first byte of the table.

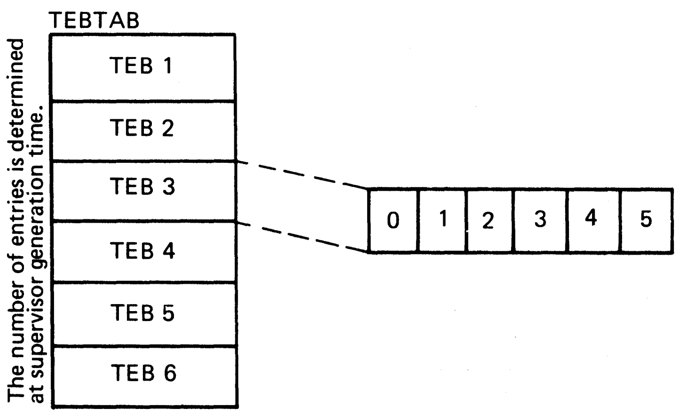


Figure 6.6. Tape Error Block Table (TEBTAB)

Bytes:

0: Error recovery retry count.

1: Permanent read data check error count.

2: Number of times the read data check error routine is entered.

3: Number of times the write data check error routine is entered.

4: Write skip (erase gap) count.

5: Noise record count.

One TEB is generated for each 2495 Tape Cartridge Reader unit if the FOPT macro contains the TEB=n parameter. Job Control resets each TEB at normal or abnormal End-of-Job. An unused TEB contains HEX'FF0000000000'. A TEB is referenced from byte 3 of a Tape Cartridge Reader unit PUB.

Bytes 70-71 (X'46' - X'47') of the partition communications region(s) contain the address of the TEB Table entry. Label TEBTAB identifies the first byte of the table.

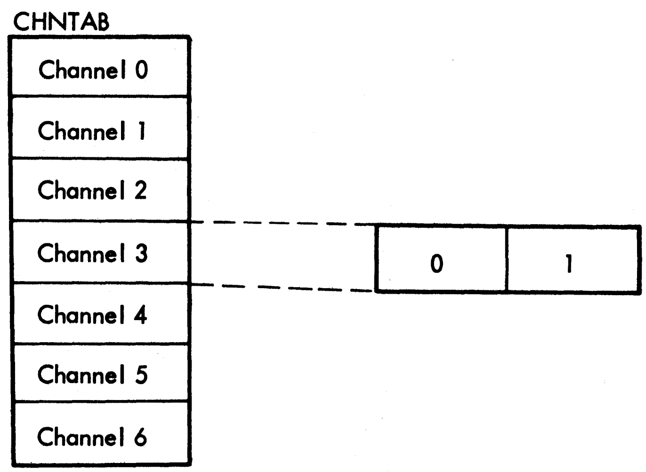


Figure 6.7. Channel Control Table (CHNTAB)

Bytes:

0: X'10' = Byte Multiplexer Channel
X'11' = Byte Multiplexer Channel with burst mode support
X'13' = Byte Multiplexer Channel running in burst mode
X'20' = Block Multiplexer Channel
X'C0' = Selector Channel
X'80' = Channel not operational or not present in the system

1: Always zero

Note: Bytes 60-63 (X'3C' - X'3F') of the System Communication Region (SYSCOM) contain the address of the Channel Control Table. Label CHNTAB identifies the first byte of this table.

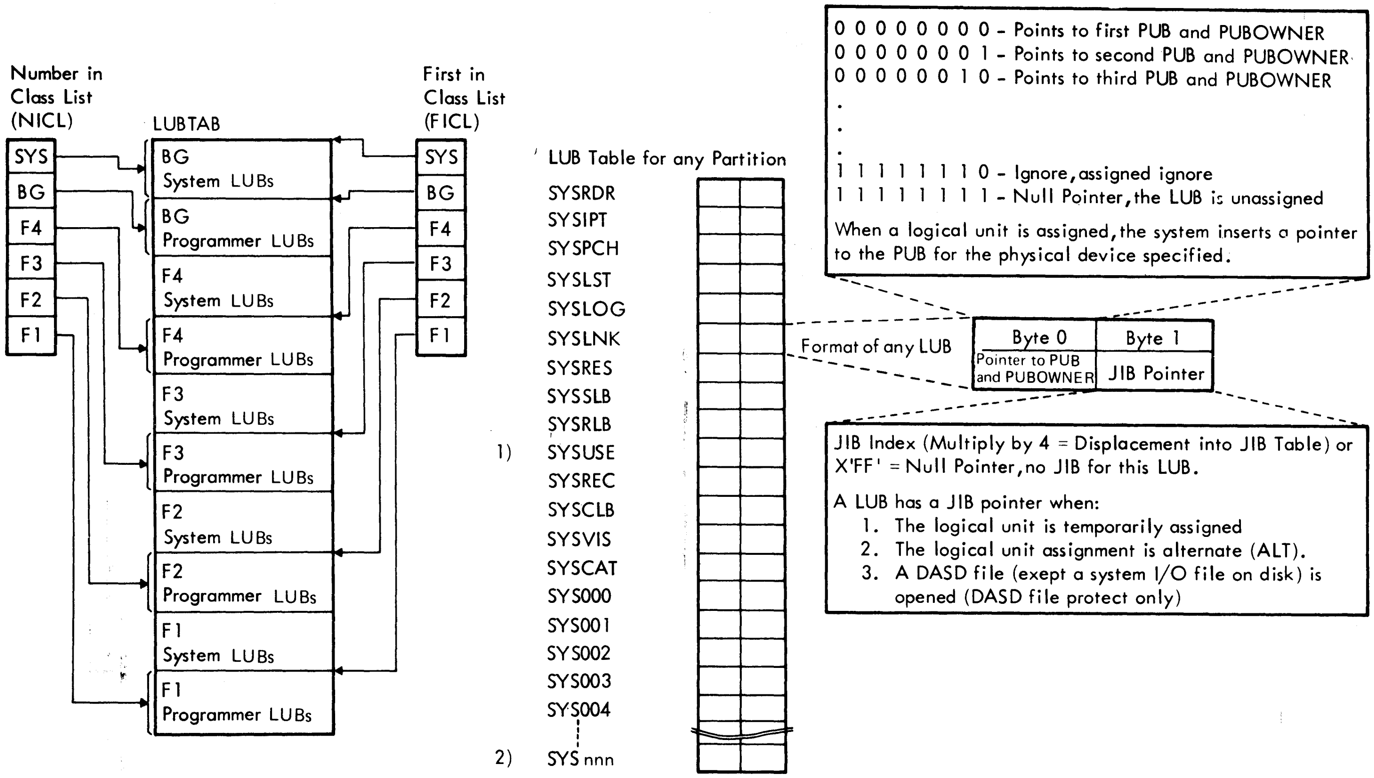


Figure 6.8. Logical Unit Block (LUB) Table

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

Notes:

1. SYSUSE may be called SYSCTL in error recovery messages.

2. Not more than 255 LUBs can be generated into the system. As there are 14 systems LUBs for each partition the maximum number of programmer LUBs in the system is 241 if NPARTS=1, 227 if NPARTS=2, 213 if NPARTS=3, 199 if NPARTS=4, or 185 if NPARTS=5.

3. Null entries are generated at supervisor generation time. Standard assignments can then be made to PUBs at supervisor generation time or during IPL.

SELECTOR AND BLOCK MULTIPLEXER CHANNEL SWITCHING

Some devices may be attached to two consecutive selector or block multiplexer channels. If the CHANSW parameter was specified in the PIOCS macro, the supervisor supports switching I/O requests from the (busy) primary channel to the (available) secondary channel.

If, when an I/O request is queued for a device attached to two selector or block multiplexer channels, the result of the TEST CHANNEL instruction is CHANNEL NOT AVAILABLE, one is added to the channel number and the instruction is reexecuted. If the result is then CHANNEL AVAILABLE, I/O is started on the secondary channel. If neither channel is available, return is made to the General Exit Routine and the request is left in the channel queue. Then, when the previous operation on one of the channels terminates, I/O for this new request is started on whichever channel the interruption occurred. Possible at SIO time the channel is busy, while it was available at TEST CHANNEL time. This can occur in a system that supports block multiplex mode, when the switchable devices are on a block multiplexer channel together with devices that have the disconnect command chaining feature. Therefore, the channel switching routine is also entered when the channel is found busy at SIO time.

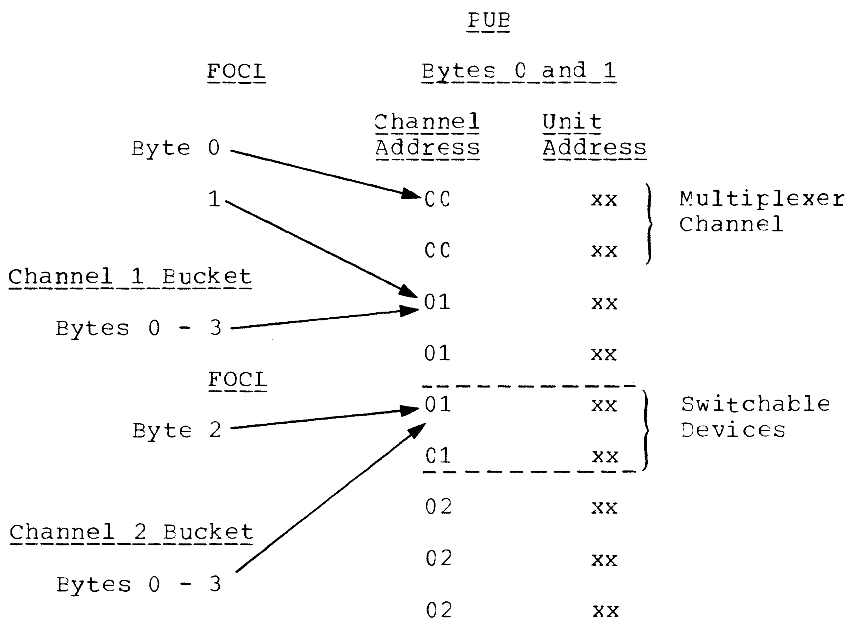


Figure 6.9. Example of Pointers to PUBs for Switchable Devices

In a supervisor that supports channel switching, the FOCL byte for the first PUB on the secondary channel is adjusted to point to the PUB of the first of the switchable devices, which must have been added as the last devices on the primary (lower) of the two consecutive channels (see Figure 6.9).

Since, in setting up bytes 0 - 3 of the channel bucket, the FOCL was used by IPL to calculate the address (refer to the section "Channel Buckets"), these bytes of the channel bucket contain the actual address of the PUB for the first switchable device. Therefore, when an I/O interruption occurs for this device on the secondary channel while the primary channel is still busy, the new request is started on the secondary channel.

CHANNEL QUEUE

Initially, the Free List Pointer (FLPTR) points to the first entry in the Channel Queue Table. The Chain Byte in this entry points to the second entry, whose chain byte points to the third entry, and so on. The Chain Byte in the last entry is initially a X'FF' delimiter (see Figure 6.10).

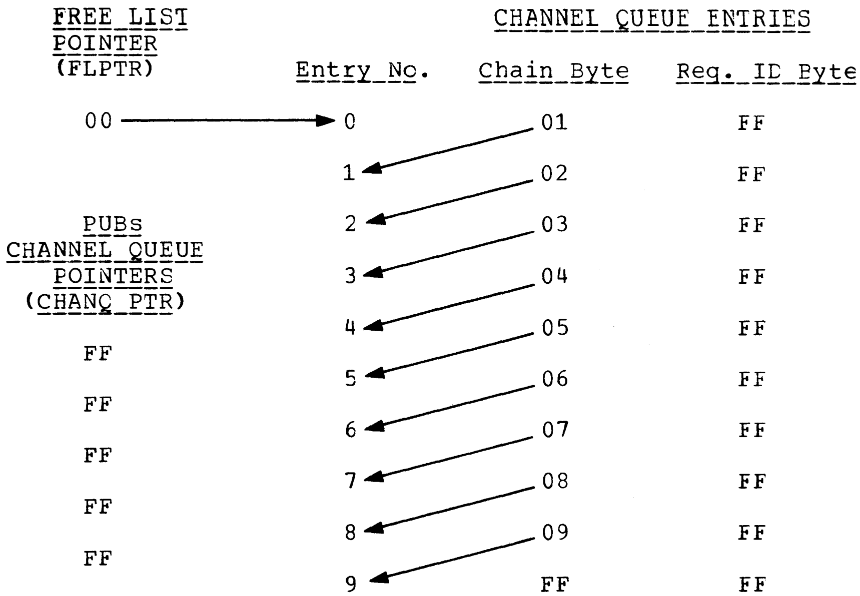


Figure 6.10. Initial Contents of Significant Bytes used in Queueing I/O Requests

Queueing I/O requests

At the first request for I/O, the contents of FLPTR is moved to CHANQ PTR in the PUB for the specified device, the old CHANQ PTR is moved to the Chain Byte of the first Channel Queue entry, and the old Chain Byte is moved to FLPTR (see Figure 6.11). The PIK in the Requestor's Identifier Byte (Req. ID) indicates that this entry is now occupied.

If, subsequently, I/O is requested on another device, the same sequence of events occurs: FLPTR moved to CHANQ PTR, CHANQ PTR moved to Chain Byte, and Chain Byte moved to FLPTR (Figure 6.12).

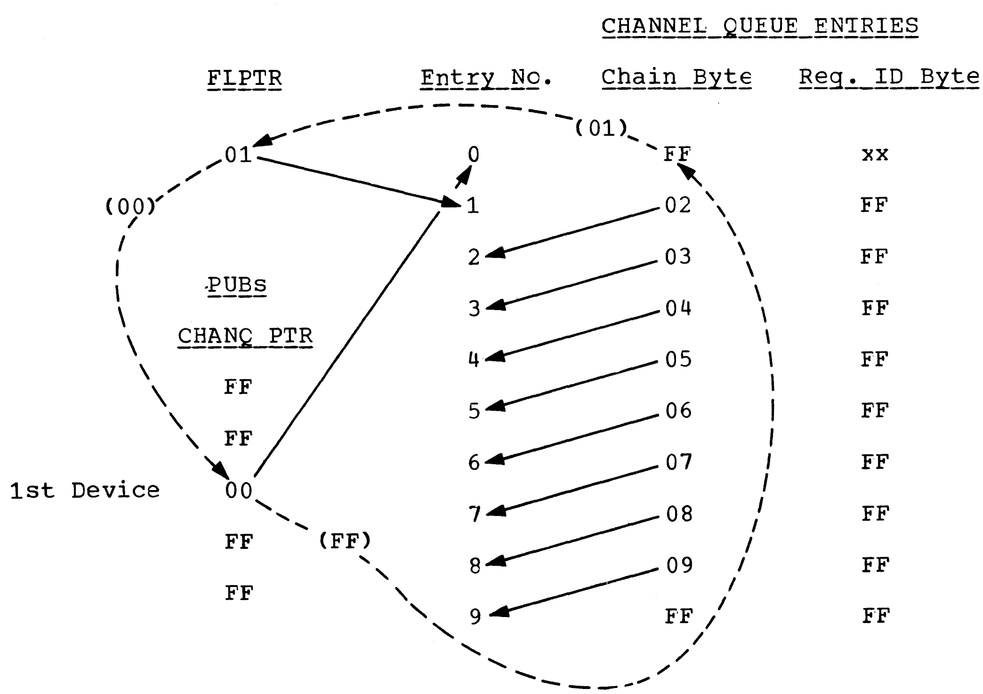


Figure 6.11. Queueing the First I/O Request

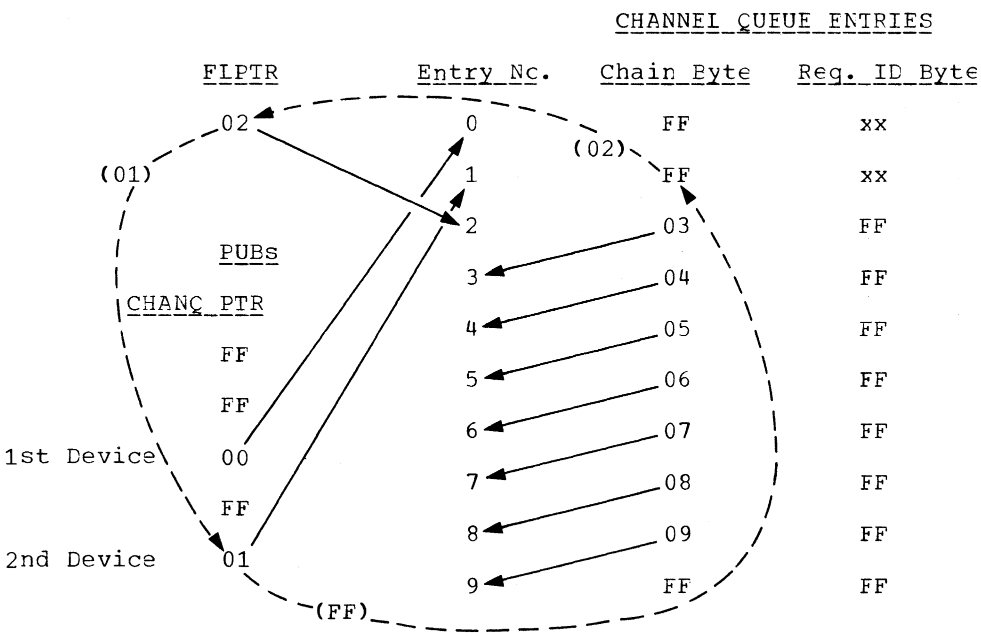


Figure 6.12. Queueing the First I/O Request for Second Device

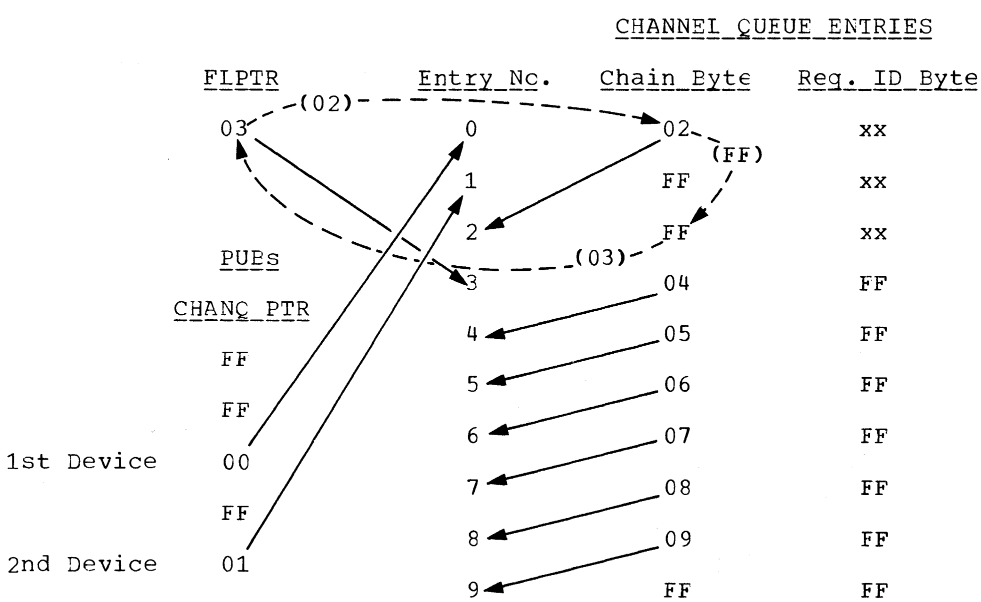


Figure 6.13. Queueing an I/O Request for a Device with a Request Already Queued

If now I/O is requested on a device on which a request is already queued (PUB CHANQ PTR not X'FF'), the Chain Byte of the First Free Entry is moved to FLPTR, the old FLPTR is moved to the Chain Byte of the previous Channel Queue Entry for this device, and the old Chain Byte is moved to the new entry (see Figure 6.13).

Filling the Channel Queue

The queueing process may continue until several requests are queued for each device (see Figure 6.14). If a program or routine causes I/O requests in rapid succession, the channel queue table may become completely filled with active entries; the FLPTR then contains X'FF' (the Chain Byte from the last free entry) and no more requests can then be queued until an I/O interruption occurs.

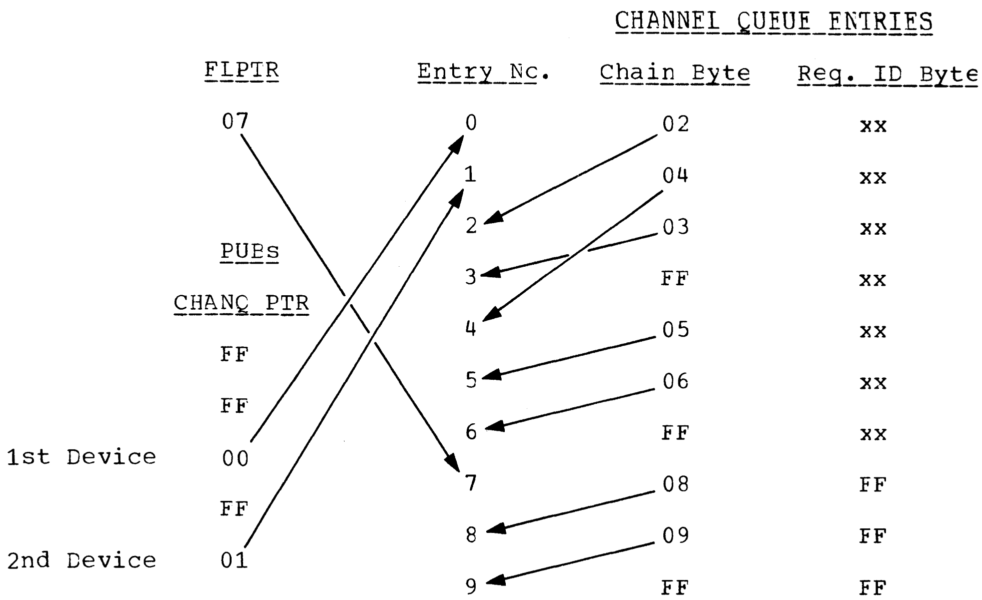


Figure 6.14. Example of Channel Queue with Several Requests Queued for Each of Two Devices

Dequeueing

On completion of an I/O operation, the channel signals an I/O interruption. The Chain Byte of the first channel queue entry for the device is moved to the PUB CHANQ PTR, the old CHANQ PTR is moved to the FLPTR, and the old FLPTR is moved to the Chain Byte as shown in Figures 6.15 and 6.16. The last released entry thus becomes the first in the chain of free entries.

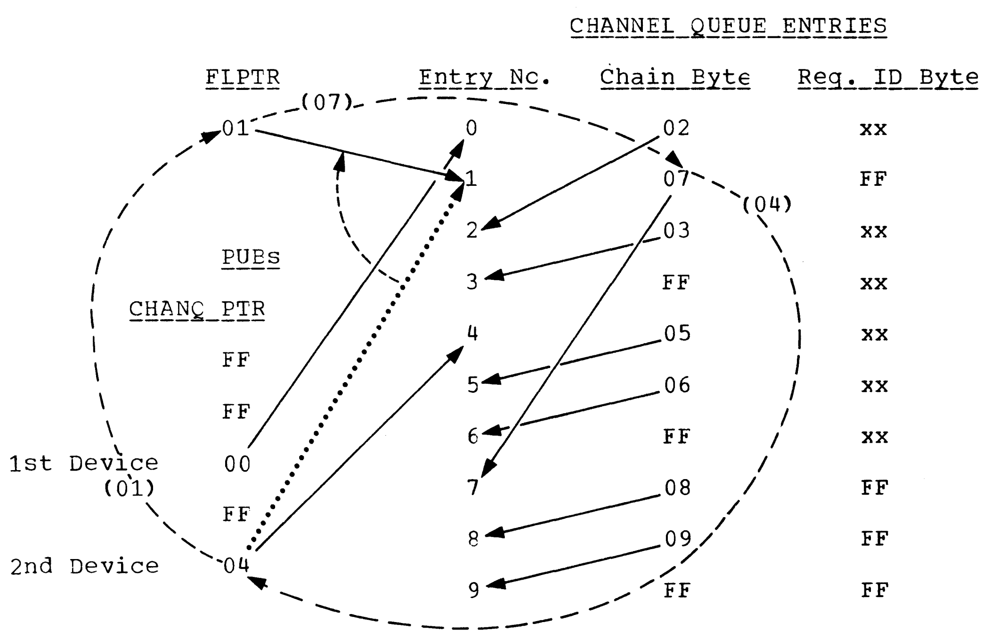


Figure 6.15. Dequeueing First Request on Second Device, Leaving Outstanding Requests Chained

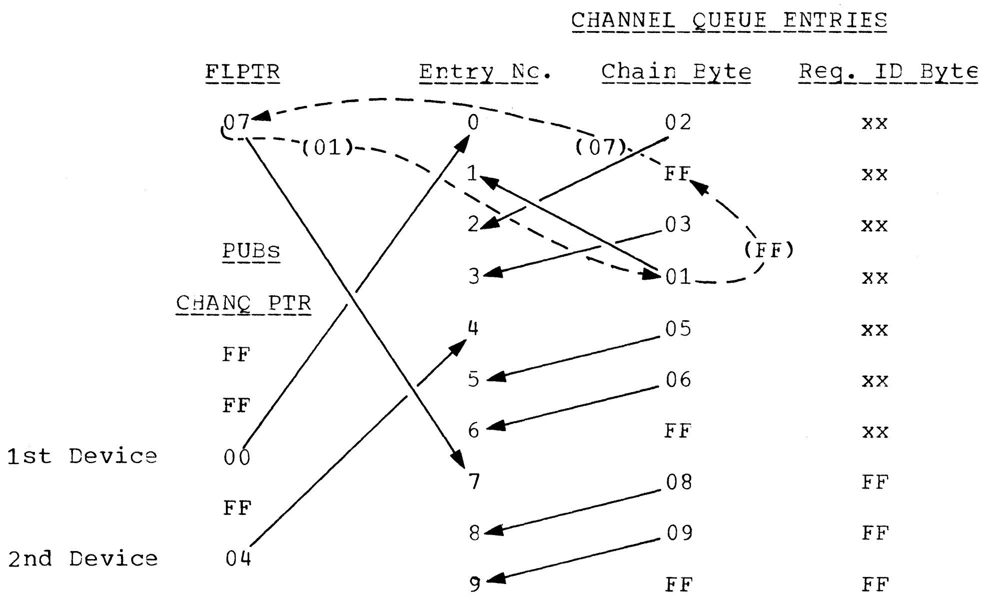


Figure 6.16. Adding a New Request to Chain for First Device

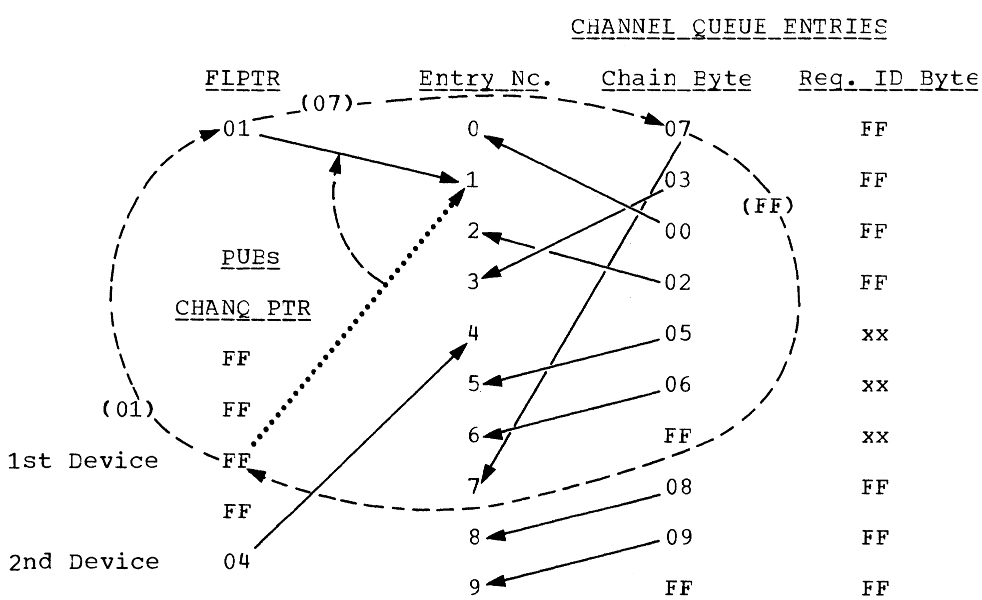


Figure 6.17. Dequeueing Last Request in Chain for First Device

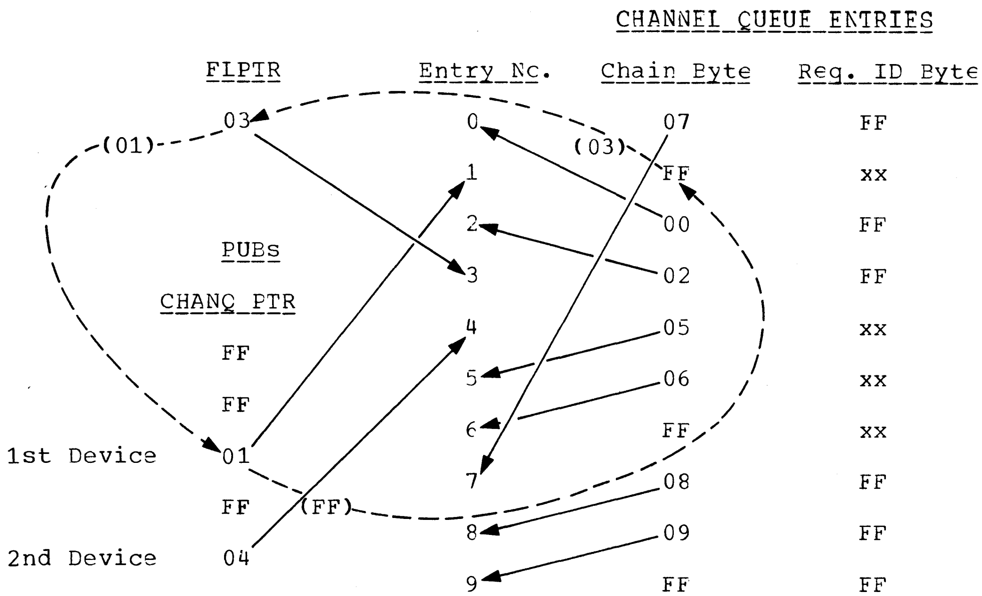


Figure 6.18. Queueing New Request for First Device

Queueing a New Request

When a new request is queued, the way the pointers are moved depends on whether a request is already queued for the same device.

If a request is already queued (PUB CHANQ PTR not X'FF'):

1. First Free Entry Chain Byte moves to FLPTR.

2. Old FLPTR moves to Chain Byte of Previous Entry.

3. Old Chain Byte moves to Chain Byte of New Entry (See Figures 6.13 and 6.16).

If this is the First Request for the device (PUB CHANQ PTR = X'FF'):

1. First Free Entry Chain byte moves to FLPTR.

2. Old FLPTR moves to CHANQ PTR in the device PUB.

3. Old CHANQ PTR moves to Chain Byte of New Entry (See Figures 6.11, 6.12, and 6.18).

CHANNEL BUCKETS

A channel bucket is an area for saving the information relating to the last I/O request that was started on a channel.

Each I/O channel in the system has a 24-byte bucket in which the I/O registers and the PIB pointer are saved. Also in the bucket, the actual address of the first PUB on the channel is permanently stored.

The address of the channel bucket table is in bytes 48-51 (X'30' - X'33') of SYSCOM. The first byte is identified by the label REGSAV.

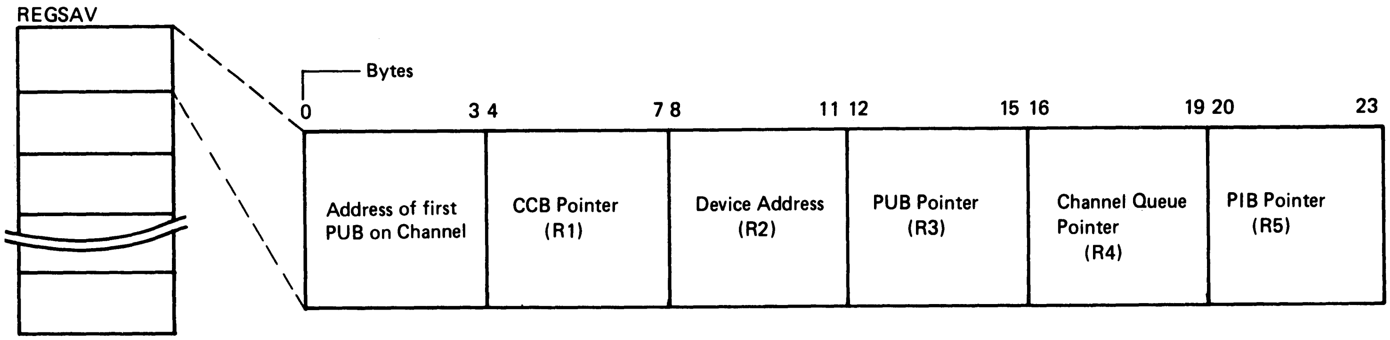


Figure 6.19. Channel Buckets

A channel bucket contains information related to the last I/O started on the channel.

The number of channel buckets in a system equals the number of I/O channels in the system.

Bytes 48-51 (X'30' - X'33') of the System Communication Region (SYSCOM) contain the address of the Channel Buckets. Label REGSAV identifies the first byte of the table.

Bytes 0-3 are set up by IPL which uses the FOCL to calculate the actual address of the first PUB on the channel. These bytes are never modified.

The remaining bytes (4-23) are set up by START I/O. They contain all pertinent I/O information relating to the last request started on the channel.

Whenever I/O is initiated on a channel, the I/O registers are saved in bytes 4-19 of the channel bucket and the PIB pointer is saved in bytes 20-23.

When an I/O interruption occurs, the I/O Interrupt Handler first checks the device address in low real storage to see if it is the same as the device address in the channel bucket. If the addresses are the same, then the interruption is from the last device started on this channel. The I/O information is then retrieved from the channel bucket and loaded into the registers for processing the interruption. Since an interruption from the last device started is most common, by avoiding a scan of the PUB table the performance is improved.

If the I/O interruption is from a device other than the last one started on the channel, the PUB table is scanned for a device address equal to that in low real storage. When the correct PUB is found, the I/O registers and PIB pointer are set up for processing the interruption.

COMMAND CONTROL BLOCK

The CCB establishes communication between the problem program and physical IOCS. The CCB is two double words in length with eight major fields and an optional field, as shown in Figure 6.20. All data in the CCB is in the hexadecimal format. The nine fields of the CCB are listed and described as follows:

Bytes 0 and 1

Count Field: Contains the residual count, which is stored in these two bytes by PIOCS when the CCB is removed from the queue.

Bytes 2 and 3

Transmission Information: Used for communication between PIOCS and the problem program.

Note: Bytes 2 through 5 are ANDed off, by PIOCS, when the CCB is placed in the queue. However, the communication bits that were set on by the problem program are left on.

Bytes 4 and 5

CSW Status Bits: Contains the CSW status information, which is stored in these two bytes by PIOCS before control is returned to the problem program.

Note: The particular bits that are turned on in bytes 2 through 5 indicate the conditions that were detected by the problem program and PIOCS.

Bytes 6 and 7

CCB Type, EOC Switch, and Symbolic Unit Class (byte 6): The first half-byte contains the type of CCB and the EOC switch. The second half-byte indicates the class of the symbolic unit whose hexadecimal representation is contained in byte 7.

Symbolic Unit Address (byte 7): Contains the hexadecimal representation of SYSnnn. This value, in conjunction with the second half of byte 6, represents the location of the logical unit in the LUB table (see Figure 6.8) and is placed in the CCB by the problem program.

Byte 8

Buffer Offset: Is not used in EBCDIC files, and must then contain hexadecimal 0.

Contains the buffer offset (block prefix length) in ASCII files

Input (F, V, U)	X'00' - X'63'
Output (F, U)	X'00'
Output (V)	X'00' or X'04'

Bytes 9 to 11

CCW Address: Contains the address of the CCW that is associated with this CCB. This address is placed in the CCB by the problem program.

Byte 12

PIOCS Byte:

X'80' - CCB being used by ERP.
X'40' - channel appendage routine for a teleprocessing device, VSAM or POWER/VS.
X'20' - sense information desired.
X'10' - message writer use.
X'08' - EU tape error indicator.
X'01' - seek separation active.
X'04' - OLTEP appendage available.

Bytes 13 to 15

CCW Address in CSW: Contains the CCW address from the CSW. If the I/O is for a virtual-mode request, this address is first translated into the virtual address of the CCW. This address is stored by PIOCS before control is returned to the problem program. A CCB that has been queued, by PIOCS, to service a problem program I/O request cannot be used for a second problem program I/O request until the first request has been completed.

Note: Bytes 13-15 contain the address of the channel appendage routine when bit X'40' is set in byte 12.

Bytes 16 to 23

Optional-Sense CCW: Bytes 16-23 are appended to the CCB by the CCB macro expansion when the user desires sense information to be returned on unrecoverable I/O errors. The macro expansion also turns on bit 2 (X'20') in byte 12 of the CCB. User handles all error or exceptional conditions except program check, protection check, channel control check, and interface control check.

Count	Transmission Information	CSW Status Bits	Type Code and Logical Unit	Reserved for Logical IOCS	CCW Address	Reserved for Physical IOCS	CCW Address in CSW	Optional Sense CCW
0 1	2 3	4 5	6 7	8	9 11	12	13 15	16 23

Byte(s)	Description	Set on by
0-1 Used for residual count.		
2-3 Transmitting information between Physical IOCS and Problem Program	Byte 2	Set on by
	Bit 0: Traffic Bit (Wait) (Note 5).	PIOCS*
	Bit 1: End-of-File (/* or /&) 3211-UCSB Parity Check (Line Complete) (Note 2).	PIOCS
	Bit 2: Irrecoverable I/O error.	PIOCS
	Bit 3: Accept Irrecoverable I/O error.	Pr.Pr.**
	Bit 4: Return DASD Data Checks, 3540 Diskette Data Checks, 2671 errors, or 1017/1018 errors to the user; indicate action-type messages for DOC; return 5425 not ready.	Pr.Pr.
	Bit 5: Post at Device End (Note 5).	Pr.Pr.
	Bit 6: Return Tape Read Data Check; 1018 or 2560 Data Check; 2520, 2540, 2560, 3881, or 5425 Equipment Check; Accept 3504, 3505, or 3525 Permanent Error; DASD Data Checks on Read or Verify Command on 3203, 3211, or 5203 Passback Requested. (Notes 3, 6, and 8)	Pr.Pr.
	Bit 7: User Error Routine (Note 10).	Pr.Pr.
	Byte 3	Set on by
	Bit 0: DASD Data Check in Count Area; Permanent Error for 3330, or 3340; MICR-SCU Not Operational; 1287/1288 Data Check; 3203, 3211, or 5205 Print Check/Equipment Check; 3540 Special Record Transferred.	PIOCS
	Bit 1: DASD Track Overrun; MICR Intervention required; 1287-Keyboard Correction in Journal Tape Mode; 1017-Broken Tape; 3211-Print Quality/Equipment Check.	PIOCS
	Bit 2: DASD End-of-Cylinder; MICR-(Note 4) 1287/1288-Hopper Empty in Document Mode; 3211/2245 Line Position Error (Note 7).	
	Bit 3: 2520, 2540, 3881-Equipment Check; 2560, 3203, 5203, 5425 Data Check/Equipment Check; Tape-Read Data Check; DASD-Any Data Check; 1287-Equipment Check; 1017/1018-Data Check; 3211-Print Check/Data Check; 3504, 3505, 3525 Permanent Error (Note 8); 3540 Diskette Data Check.	PIOCS
	Bit 4: Non-Recovery Questionable Condition: Card-Unusual Command Sequence; DASD-No Record Found; 1287/1288-Document Jam or Torn Tape; 3211-UCSB Parity Check (Command retry); 5425 not ready.	PIOCS

Figure 6.20. Command Control Block (CCB) (Part 1 of 3)

Count 0 1	Trans-mission Informa-tion 2 3	CCW Status Bits 4 5	Type Code and Logical Unit 6 7	Reserved for Logical IOCS 8	CCW Address 9 11	Reserved for Physical IOCS 12	CCW Address in CSW 13 15	Optional Sense CCW 16 23

Byte(s)	Description	Set on by
	Bit 5: No Record Found Condition (Retry on 2311, 2314, 2319, 3330, or 3340).	Pr.Pr.
	Bit 6: Carriage Channel 9 Overflow or Verify Error for DASD; 1287-Document Mode-Late, Stacker Select; 1288-End of Page.	PIOCS
	Bit 7: Command Chaining, Retry from the next CCW to be executed.	Pr.Pr.

Byte(s)	Description	
4-5 CSW Status Bits	Byte 4 (Note 1)	Byte 5
	Bits:	Bits:
	0(32): Attention	0(40): Program Controlled Interruption
	1(33): Status Modifier	1(41): Incorrect Length
	2(34): Control Unit End	2(42): Program Check
	3(35): Busy	3(43): Protection Check
	4(36): Channel End	4(44): Channel Data Check
	5(37): Device End	5(45): Channel Control Check
	6(38): Unit Check	6(46): Interface Control Check
	7(39): Unit Exception	7(47): Chaining Check
6-7 Type Code and Logical Unit	Byte 6	
	X'0u' Original CCB (Bytes 9-11 and 13-15 contain virtual addresses) X'2u' Translated CCB (Bytes 9-11 contain real address, bytes 13-15 virtual address) X'4u' BTAM request original CCB (Bytes 9-11 and 13-15 contain virtual address) X'6u' BTAM request translated CCB (Bytes 9-11 contain real address, bytes 13-15 virtual address) X'8u' User-translated CCB in virtual partition (Bytes 9-11 and 13-15 contain real addresses) Note: Any one of the above incremented by X'10' (bit 3 on) indicates automatic switching to the beginning of the next cylinder at End-of-Cylinder condition. u: 0 = The address in byte 7 refers to a System Logical Unit. 1 = The address in byte 7 refers to a Programmer Logical Unit.	
	Byte 7	
	Hexadecimal representation of SYSnnn: SYSRDR = 00 SYSSLB = 07 SYS000 = 00 SYSIPT = 01 SYSRLB = 08 SYS001 = 01 SYSPCH = 02 SYSUSE = 09 SYS002 = 02 SYSLST = 03 SYSREC = 0A . SYSLOG = 04 SYSCLB = 0B . SYSLNK = 05 SYSVIS = 0C SYSnnn SYSRES = 06 SYSCAT = 0D (Note 9)	
8 Reserved for Logical IOCS	Buffer Offset: ASCII Input Tapes X'00' - X'63' ASCII Output Tapes Fixed X'00' Variable X'00' or X'04' Undefined X'00'	

Figure 6.20. Command Control Block (CCB) (Part 2 of 3)

Count	Trans-mission Informa-tion	CCW Status Bits	Type Code and Logical Unit	Reserved for Logical IOCS	CCW Address	Reserved for Physical IOCS	CCW Address in CSW	Optional Sense CCW
0 1	2 3	4 5	6 7	8	9 11	12	13 15	16 23

Byte(s)	Description
9-11 CCW Address	Virtual or real address of CCW associated with this CCB depending on byte 6: Real address if byte 6=X'2u', X'6u', or X'8u'; Virtual address if byte 6=X'0u', or X'4u'.
12 Reserved for Physical IOCS	X'80' CCB being used by ERP. X'40' Channel Appendage Routine present for TP device, VSAM or POWER/VS. X'20' Sense Information desired (Note 10). X'10' Message writer. X'08' EU Tape Error. X'04' OLTEP Appendage available. X'02' Tape ERP Read Opposite Recovery. X'01' Seek Separation.
13-15 CCW Address in CSW	Virtual Address of CCW pointed to by CSW at Channel End (if byte 6 = X'8u', it is the real address) or address of the Channel End Appendage Routine for TP devices, VSAM or POWER.
16-23 Optional Sense CCW	8 bytes appended to the CCB when Sense Information is desired.

Figure 6.20. Command Control Block (CCB) (Part 3 of 3)

<u>Notes</u>:

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.

2. Indicates /* or /& statement on SYSRDR or SYSIPT. Byte 4, bit 7 (unit exception) is also on.

3. DASD data checks on count not returned.

4. For 1255/1259/1270/1275/1419, disengage. For 1275/1419D, I/O Error is external interrupt routine (Channel data check or bus-out check).

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.

6. 1018 ERP does not support the Error Correction Function.

7. This error occurs as an equipment check, data check or FCB parity check. For 2245, this error occurs as a data check or FCB parity check.

8. For 3504, 3505, 3525 input or output files using ERROPT, byte 3, bit 3 is set on if a permanent error occurs. Byte 2, bit 6 is set on to allow you to accept permanent errors.

9. SYSnnn=255 - (Number of partitions times 14).

10. If User Error Routine is specified and the user needs the sense information to further process the error, byte 12, bit 2 must also be set. Otherwise, the supervisor error routine will mask off the status on return and the sense information is not available.

Command Code	Data Address
0 – 7	8 – 15, 16 – 23, 24 – 31

Flags	0 0	TP Op. Code	Byte Count
32 – 37	38	40 – 47	48 – 55, 56 – 63

Figure 6.21. Channel Command Word (CCW)

Flags

Bit	Description	
32	CD-bit (80):	causes use of address portion of next CCW
33	CC-bit (40):	causes use of command code and data address of next CCW
34	SLI-bit (20):	causes suppression of possible incorrect length indication
35	Skip bit (10):	suppresses transfer of information to main storage
36	PCI-bit (08):	causes a channel Program Controlled Interruption (ignored by DOC routines)
37	IDA-bit (04):	specifies indirect data addressing

PROGRAM STATUS WORD

0	R	0	0	0	T	I/O	E	KEY	1	M	W	P	00	CC	PROG MASK	0 0 0 0 0 0 0 0
0								8					16			24 – 31

0 0 0 0 0 0 0 0	INSTRUCTION ADDRESS
32	40 – 63

Figure 6.22. Program Status Word (PSW)

Bit	Description
0	Always zero
1	Program Event Recording (PER) mask
2-4	Always zero
5	Translate mode (DAT)
6	I/O interrupt mask
7	External interrupt mask
8-11	CPU protection key
12	Always one (EC mode) (I)
13	Machine Check mask (M)
14	Wait state (W)
15	Problem state (P)
16-17	Always zero
18-19	Condition code
20	Fixed-point overflow mask
21	Decimal overflow mask
22	Exponent overflow mask
23	Significance mask
24-31	Always zero
32-39	Always zero
40-63	Instruction address

Key	0000	Command Address
0 – 3	4 – 7	8 – 15, 16 – 23, 24 – 31

Figure 6.23. Channel Address Word (CAW)

Key	0000	Command Address
0 – 3	4 – 7	8 – 15, 16 – 23, 24 – 31

Status	Byte Count
32 – 39, 40 – 47	48 – 55, 56 – 63

Figure 6.24. Channel Status Word (CSW)

Command Address: eight higher than the address of the last CCW used (not higher on a command reject).

Status

Bit	Description
32	(8000) Attention
33	(4000) Status Modifier
34	(2000) Control unit end
35	(1000) Busy
36	(0800) Channel end
37	(0400) Device end
38	(0200) Unit check
39	(0100) Unit exception
40	(0080) Program controlled interruption
41	(0040) Incorrect length
42	(0020) Program check
43	(0010) Protection check
44	(0008) Channel data check
45	(0004) Channel control check
46	(0002) Interface control check
47	(0001) Chaining check

Byte count: Bits 48-63 form the residual count for the last CCW used.

I/O INTERRUPT HANDLER

The address of the I/O interrupt handler is in the SCP Communication Region (SYSCOM). Before branching to any entry point in this routine, general register 9 is set up as a base register.

An I/O interruption occurs when an I/O operation terminates or the operator intervenes on the device. It also occurs (as a Channel Available Interrupt) on a block multiplexer channel after a TEST CHANNEL instruction or after a SIO, as soon as the channel becomes available. Before processing the interruption, the device address in low real storage is checked to see if it is the same as the device address in the channel bucket for the channel on which the interrupt occurred.

If so, the interruption was from the last device started on the channel, and the I/O registers and the PIB pointer are loaded from the channel bucket (refer to the section "Channel Buckets").

If it was not the last device started, the PUBs are scanned sequentially from the first PUB on the channel list. If the channel list contains no PUB for the interrupting device, the device is not supported by the system so the interruption is ignored. In this case an immediate attempt is made to reschedule the channel because I/O might be pending for other devices. This action is also taken for a channel available interrupt.

When the correct PUB is found, the routine sets up pointers to the PUB, the Channel Queue Entry, the PIB, and the CCB.

If the device is owned by the On-Line Testing function, exit is taken to the OLTEP appendage routine.

Exit is taken to the Channel Appendage routine if all the following conditions exist:

1. The supervisor supports teleprocessing (TP=BTAM, QTAM, or VTAM).
2. The user specified an appendage routine address in the CCB.
3. The PIB Assign flag indicates that a channel appendage exit is allowed.

Before control is transferred to the appendage routine the CCW address in the CSW must be retranslated if the appendage belongs to a BTAM program running in virtual mode.

If on return the appendage routine for BTAM running in virtual mode, requests I/O, the new virtual channel program is translated. At the same time the copy buffers are freed that were used for I/O requests by the BTAM program.

If none of the above conditions exists, the CSW is evaluated and action is taken according to the table in Figure 6.25.

If the device status indicates that the channel program has been successfully completed, the traffic bit is posted in the CCB. If device-end posting is not required, the channel queue entry is now released.

If the requesting program is running in virtual mode and the user CCB is posted, control is given to the routine CSWTRANS (refer to the section "Channel Program Translation") to release the copy blocks and fixed pages and update the user CCB.

Rescheduling the Selector or Block Multiplexer Channel

The PUBs in the channel list are scanned, beginning with the PUB following the last one started, and the first one found with I/O pending is then started. If the end of the channel list is reached, the scan continues from the first PUB in the channel list. If there is no I/O pending in the channel, a branch is taken to the General Exit routine. This rotating PUB scan ensures that the channel is shared by all programs.

Rescheduling the Byte Multiplexer Channel

If the device is on a byte multiplexer channel with more I/O pending on the same device, exit is taken to the channel scheduler routine to start the next request immediately. If no I/O pending on that device, exit is taken to the General Exit routine.

Command Chaining for Retry on I/O Operations

When an error occurs in an I/O operation and the Command Chaining Retry bit in the CCB is on, the last CCW is reexecuted instead of the first CCW in the channel program as in a normal retry.

CSW Status Bit On	Status Condition	Action
45 46	Channel Control Check Interface Control Check	Branch to the Channel Check Handler to interrogate the bits for attempting recovery.
38 42 43 44 47	Unit Check Program Check Protection Check Channel Data Check Channel Chaining Check	If user routine available, post error to CCB and set up for exit to user, otherwise exit to Unit Check routine for error recovery.
32	Attention	For attention from the console printer keyboard or display operator console (DOC), include attention task in task selection and branch to the General Exit routine. Attention interruptions are ignored if: 1. System reallocation or condense is in operation. 2. Attention is not from the console printer keyboard or DOC.
35	Device Busy	Skip channel-end test.
36	Channel End	If a selector channel, or if interruption was from a burst-mode device on a multiplexer channel, attempt to reschedule the channel.
37 34	Device End Control Unit End	If a selector channel, attempt to reschedule the channel. If the interruption was from a burst-mode device on a multiplexer channel, attempt to reschedule channel <u>and</u> device. If a Byte-mode device attempt to reschedule the device only.
33 and 35	Control Unit Busy	Reset device to available. The status is not tested if channel-end, device-end, or control-unit-end has occurred.

Figure 6.25. CSW Testing in I/O Interrupt Handler

BURST-MODE AND HIGH-SPEED BYTE-MODE DEVICES ON BYTE MULTIPLEXER CHANNEL

When an interruption occurs on a byte multiplexer channel and burst-mode devices are supported (PIOCS macro parameter BMPX=YES):

If the interruption was from a burst-mode device, and not the last device started, the PUBs are scanned from the PUB following the last one started.

If the interruption was from a byte-mode device that was not the last one started, and there are more requests in the same channel queue, the next request is started immediately.

On a byte multiplexer channel operating in <u>multiplexer mode</u>, the channel is considered to be available. When operating in <u>burst mode</u>, however, and a burst mode or overrunnable device is to be started on the channel it is considered to be <u>not</u> available.

Suppose that START I/O on a overrunnable high-speed byte-mode device is followed immediately by a request for I/O on a burst-mode device. Without any software precautions, because the channel is available (operating in multiplex mode), I/O will start immediately on the burst-mode device. The channel, now operating in burst mode, will be monopolized by this device. Any interruption from the overrunnable byte-mode device may be lost, and the device may run over.

To prevent this overrun, overrunnable high-speed byte-mode devices are added as burst-mode devices, and whenever I/O is started on one of these devices, a switch is set in the Channel Control Table by the Start I/O routine. This switch is tested at every attempt to start I/O on a burst-mode device and, if found on, the request is left in the channel queue and exit is taken to the General Exit routine. Whenever a byte multiplexer channel is operating in burst mode, therefore, a second burst-mode (or high-speed byte-mode) device can never be started.

CHANNEL PROGRAM TRANSLATION

The supervisor must do the following before initiating an I/O operation for a virtual-mode program:

- Copy the CCB and the entire channel program into copy blocks in the supervisor.
- Translate the addresses used by the CCB and the channel program into real storage addresses and place these addresses in the copied CCB and channel program.
- Build IDALs (Indirect Data Address Lists) in the supervisor for all user requests which have IDALs and for all I/O areas which cross one or more page boundaries.
- Fix all pages containing I/O areas in real storage for the duration of the I/O operation.

These functions are performed by the routine CCWTRANS. CCWTRANS is called by the channel scheduler every time a virtual-mode I/O request is made. For I/O requests from BTAM channel appendages this routine is entered at its entry point CCWTRBT2 (for further information see the section "BTAM Considerations").

At the completion of an I/O operation, the routine CSWTRANS is called by the I/O interrupt handler. It must do the following:

- Retranslate the address of the last CCW pointed to by the CSW at channel end to its correct virtual address. This address is placed in the copied CCB.
- Free the data areas.
- Release the copy blocks used for the translation except the CCB copy block.
- If possible, transfer the CCB information which has changed to the original CCB. If this is not possible (because the original CCB is not in real storage) indicate to the dispatcher that this must be done before the user task is to be given control again. In that case, the dispatcher calls a special routine (MOVECCB) to transfer the end of channel information from the copied CCB to the CCB in the user program.

TRANSLATION CONTROL AND COPY BLOCKS

The following control and copy blocks are used to copy and translate a CCB and channel program for a virtual mode I/O request:

- A translation control block (CCWTCB or TCB abbreviated). This block is used as a work and save area during translation.
- A CCB copy block. The user CCB and sense CCW (if any) are copied into this block. The CCB copy block also contains information about the copied and translated channel program.
- CCW copy blocks. Each block contains copy locations for up to 7 contiguous CCWs and queueing information.
- IDAL blocks. Blocks used for building Indirect Data Address Lists for CCWs having IDALs or for data areas which cross page boundaries.
- Fix information blocks. Each block contains a bit string for fix information for a block of 1088K of real storage. One or more fix information blocks are generated if a page is fixed at a location greater than 384K (information for page frames up to that address are kept in the CCB copy block).

The TCB for a request is located behind the system save area for the requesting task. The other blocks are 72-byte blocks located at the end of the supervisor. They are dequeued from the free copy block queue (pointed to by AFCB) as needed, and enqueued again when they are no longer needed by the requesting task.

If the queue of free copy blocks is empty when a request for a copy block is made one of the following actions will be taken:

- If the request is from a BTAM appendage routine, the system will enter a hard wait (refer to the section "BTAM Considerations").
- If the requesting task is the only one using the CCW translation routines, it will be canceled (not enough copy blocks available to ever satisfy the request).

- If the request is for a CCB copy block or if at least one request has been handled successfully the requesting task is set copy block bound (set PIBFLG=X'75' and TRTBUF=X'00').

- If no other task is complete and if the request is not for a CCB copy block, the used copy blocks are freed and the fixed pages are freed and the task is set translation bound (PIBFLG=X'7B' and TRTCCW=X'00'). When a translation has been successfully completed, the request will be started again from the beginning.

The Translation Control Block (CCWTCB)

Because a translation request may be interrupted (by a page fault, wait, etc.), it is necessary that the translation routine be partially reenterable so that several requests may be handled simultaneously. To accomplish this, a TCB is kept for each request and used as a dynamic save and work area.

The TCB for a task requesting translation is enqueued to the used TCB queue (pointed to by AUTCB) when a translation is begun and released again when the translation is complete. This is not true however, for a translation for a BTAM appendage. It is translated without the TCB being enqueued (for further details refer to the section "BTAM Considerations").

Figure 6.26 shows the fields of the TCB and their explanations. The fields LINEPTR and BENDPTR are explained in more detail in the section "Copying Status Modifier Commands".

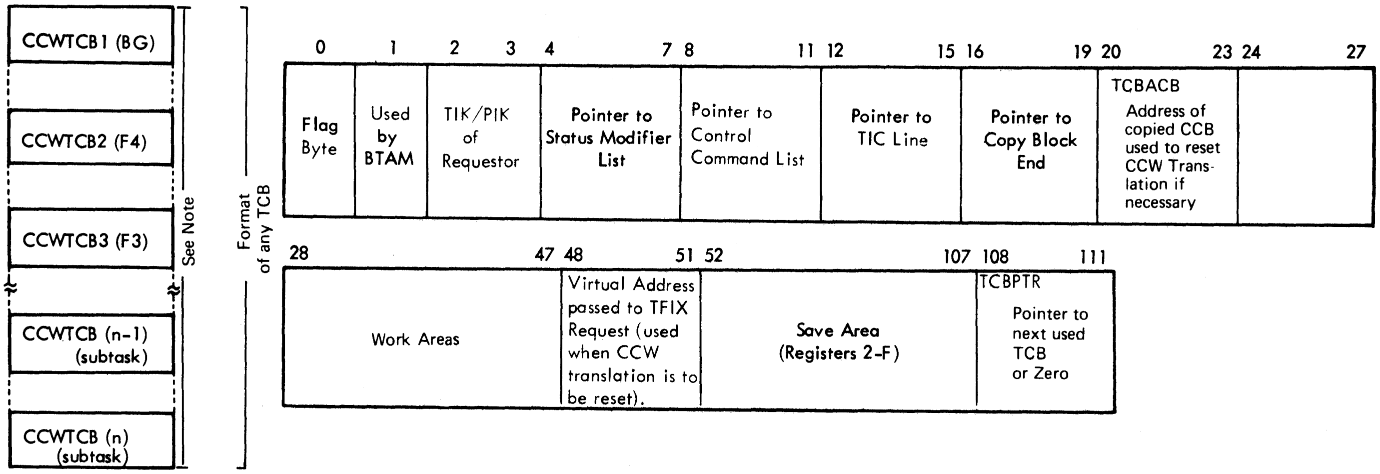


Figure 6.26. Translation Control Block (CCWTCB)

Byte 0 (TCBFLAG):

```
bit 0 = 1:  data chaining specified
    1 = 1:  Read/Sense command specified
    2 = 1:  Read backward command specified
    3 = 1:  Status modifier command with data chaining
    4 = 1:  Status modifier command with command chaining
        5:  Reserved
        6:  Reserved
        7:  Reserved
```

Byte 1 (ADBTAMCB):
Number of copy blocks needed in addition to those required for current CCW-translation request (refer to the section "BTAM Considerations").

Byte 4 (DEVSTPTR):
Pointer to status modifier list belonging to handled device. Zero if device does not support status modifier commands.

Byte 8 (DEVCDPTR):
Pointer to control command list which belongs to handled device. Zero if device does not support control commands with data area.

Byte 12 (LINEPTR):
Chain of knots of tree structure caused by TIC following status modifier command. (Refer to the section "Translating Status Modifier Commands".) Zero if no knots exist in CCW chain.

Byte 16 (BENDPTR):
Chain of knots built because status modifier command is last one fitting in CCW copy block. Zero if no status modifier commands at end of CCW copy blocks.

Note: One TCB is generated for each partition supported. With asynchronous processing support 15 TCBs are generated.

To locate the TCB (associated with the partition/task), add X'50' to the address of the System Save Area (displacement X'C9' of the appropriate PIB). Labels CCWTCB1 - CCWTCBn identify the first byte of the appropriate TCB.

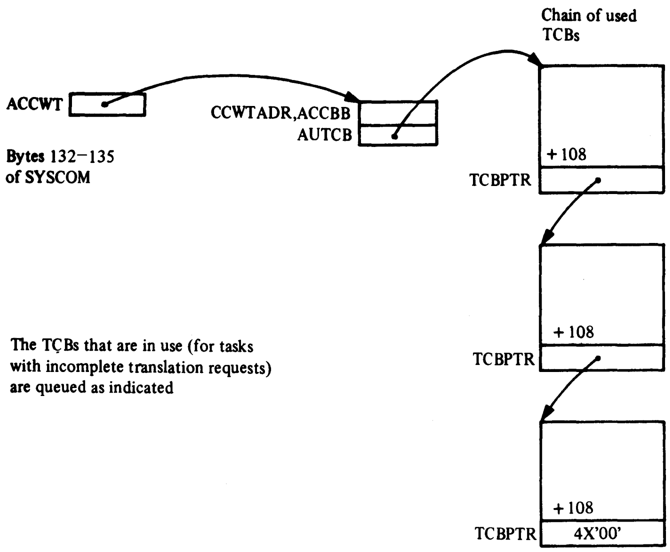


Figure 6.27. Chain of Used Translation Control Blocks

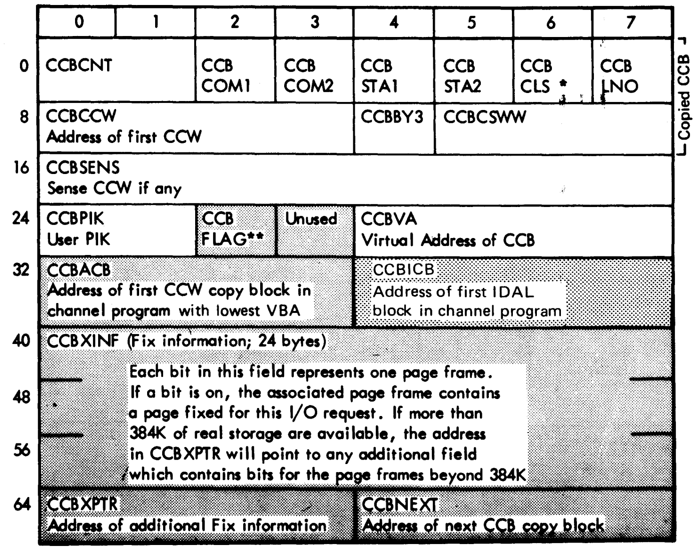


Figure 6.28. CCB Copy Block

*Bit 2 is set (X'20') to indicate copied CCB.

**Legend CCBFLAG:

Bits:

0: indicates that CCW-translation of this request is complete; indicator is set before I/O-request is enqueued in channel queue
1: indicates that at least one time during CCW-translation control has been transferred to TFIX-routine; if 0 scan through CCBXINF for freeing pages is skipped; indicator is set immediately before control is transferred to TFIX- routine
2: unused
3: indicates that the next CCW-translation request from BTAM is from BTAM channel appendage. This indicator is set immediately after 1st time request from BTAM has been completed.
4-7: unused

CCB Copy Blocks

For each virtual-mode I/O request one copy block is used to contain the copied CCB and its sense CCW, if any. The rest of the block contains control information about the translated program. Figure 6.28 shows the layout of the CCB copy block. The not shaded areas are initialized when the block is first retrieved and the CCB is copied. The shaded areas are initially zero and are filled in during the translation process.

All the CCB copy blocks in use are queued in the queue pointed to by ACCBB. Each CCB copy block is also individually pointed to by a field in the TCB for that request. After a translation is completed and enqueued in the channel queue, the CCB is also pointed to by its channel queue entry. Figure 6.29 shows the relationship of the CCB copy blocks to one another and to the other blocks.

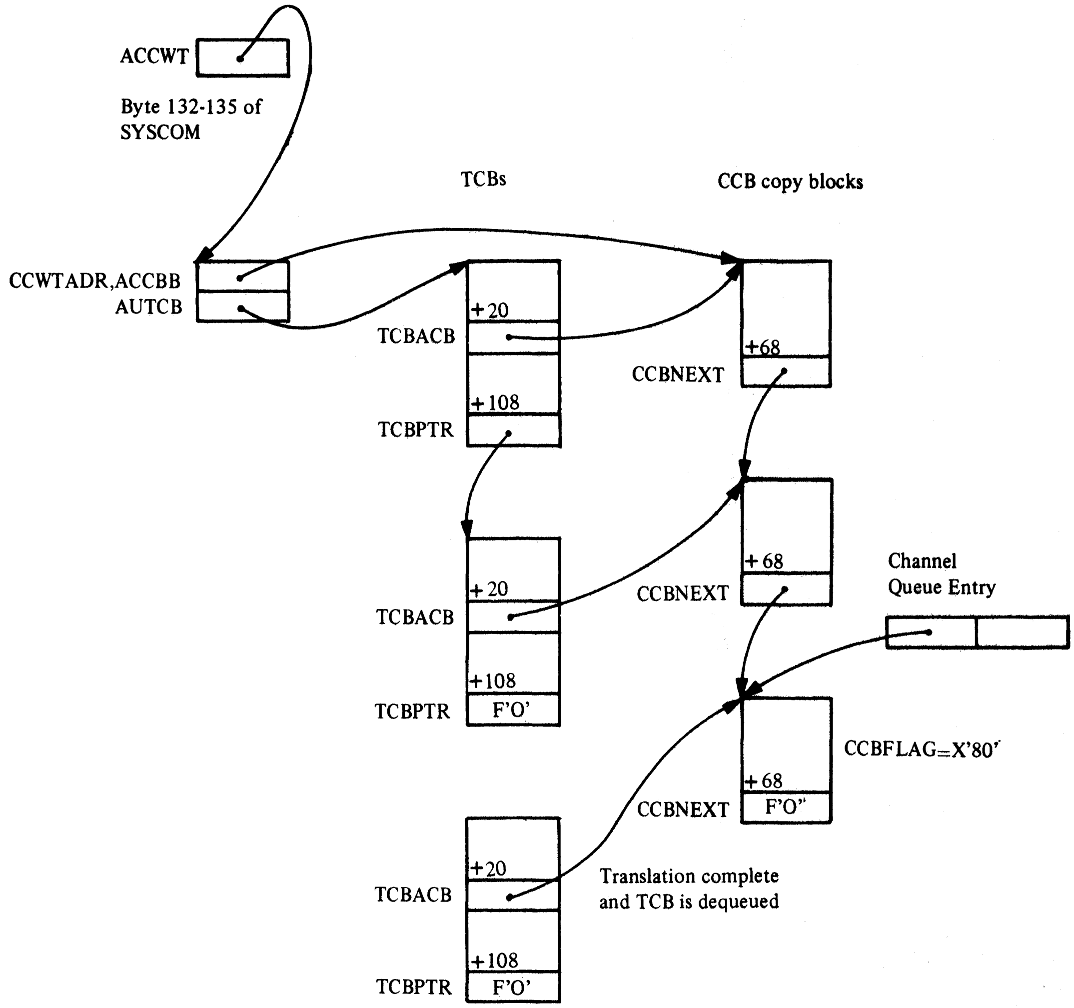


Figure 6.29. Locating CCB Ccpy Blccks

CCW Copy Blocks

Each CCW copy block consists of 7 copy locations and 16 bytes for pointers and inserted TIC commands. The layout of a CCW copy block is shown in Figure 6.30. The non-shaded areas are initialized when a copy block is dequeued from the free block queue. The shaded areas are originally zero and the relevant information is entered when the CCWs are copied.

	0	1	2	3	4	5	6	7
0	1st Copy location for CCW							
8	2nd Copy location for CCW							
16	3rd Copy location for CCW							
24	4th Copy location for CCW							
32	5th Copy location for CCW							
40	6th Copy location for CCW							
48	7th Copy location for CCW							
56	X'80'*	X'000000'			Virtual address of first CCW in the Copy block (VBA)			
64	X'88'**	X'000000'			Address of next CCW Copy block in the chain (ANB)			

Figure 6.30. CCW Copy Block

*X'80' indicates the end of the CCW copy locations in the block. It is replaced by a TIC (Transfer in Channel command) if the 7th copy location contains a copied CCW with data- or command chaining. Bytes 57-59 will then point to the copy location of the CCW following the CCW in the 7th copy location. Bytes 56-59 will not be changed if the CCW in the 7th copy location is a TIC.

**X'88' indicates the last 8-byte entry in the block. It is replaced by a TIC if the CCW in the 7th copy location is a status modifier CCW. Bytes 65-67 will then point to the copy location of the second CCW following the status modifier CCW.

The CCW copy blocks for a translation are queued in order of increasing VBAs (see Figure 6.30) with the lowest one being pointed to by the field CCBACB in the CCB copy block. Figure 6.31 shows the relation of CCW copy blocks to one another.

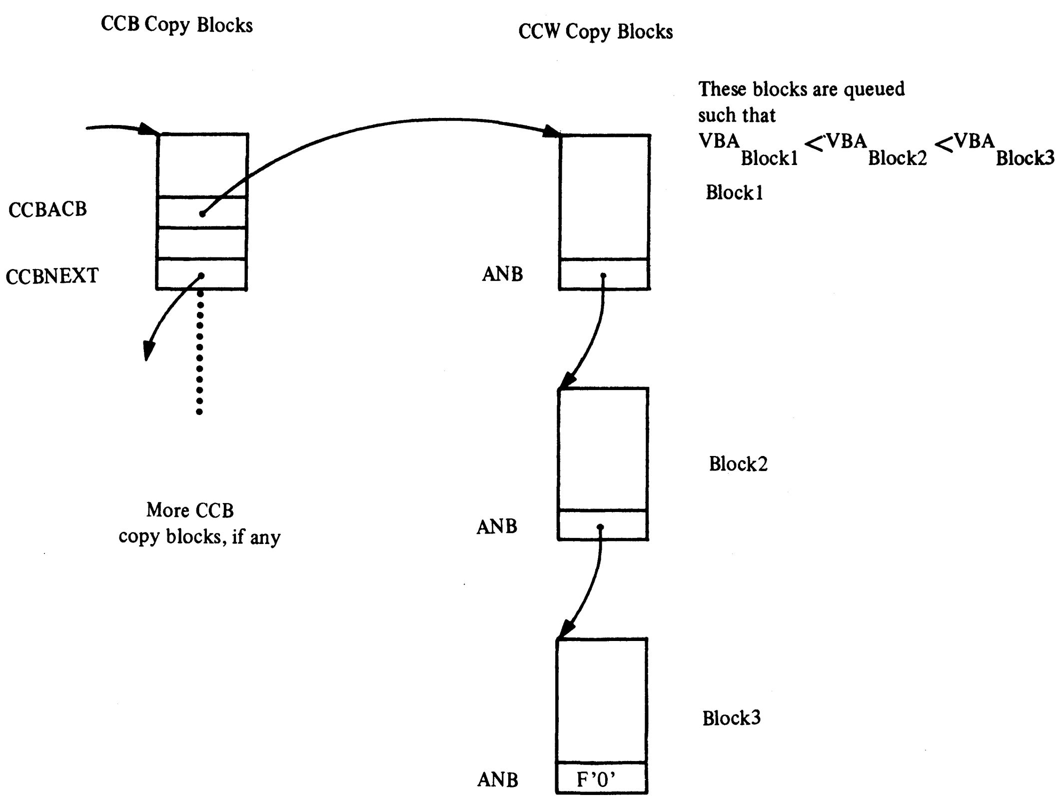


Figure 6.31. Locating CCW Copy Blocks

IDAL Blocks

User CCWs which have IDALs (indirect data address lists) or whose data areas cross page boundaries must have an IDAL in the copied channel program.

If a data area crosses a page boundary, the CCW is changed to show that an IDAL is used (bit 37 of the copied CCW is set) and the address of the IDAL is placed in the data address of the CCW. The IDAL pointed to contains one entry for the beginning of the data area and one entry for each page boundary crossed.

An IDAL must be located in consecutive copy block locations, so that if an IDAL cannot fit into the last block in the queue (the count in IDALCNT is less than the number required) a new block must be enqueued. Each IDAL block has 17 locations for Indirect Data Address Words (IDAWs).

An I/O area in a program running in virtual mode must be less than 32K bytes long (which can be covered by 16 IDAWs).

After an I/O area has been TFIXed in real storage, the addresses in the IDAL are translated to point to the correct real storage locations (the address of the beginning of the I/O area and the beginning of the page frames for the rest (the address of the last bytes in the page frames for a read backward command).

Each IDAL is pointed to by the CCW which references it. In addition, the IDAL blocks are queued with the first one being pointed to by the field CCBICB in the CCB copy block. Figure 6.32 shows the relation between the IDAL blocks and the other blocks.

Fix Information Blocks

In order to keep track of which page frames have been TFIXed for a request, a bit string is used which has a bit for at least every page frame up to the highest one which is TFIXed. If no page is TFIXed in an address higher than 384K, then the bit string in CCBXINF is sufficient (192 bits = 384K). Whenever a page is TFIXed, the bit corresponding to its page frame is set to one. If a page is used more than once by a request, it is only TFIXed once.

If a page is TFIXed at a location beyond 384K, one or more additional bit strings must be added. This is done by enqueueing a copy block. Each copy block thus enqueued provides fix information for an additional 1088K of real storage. The additional blocks are queued with the first one being pointed to by the field CCBXPTR in the CCB copy block. Figure 6.33 shows how fix information is kept.

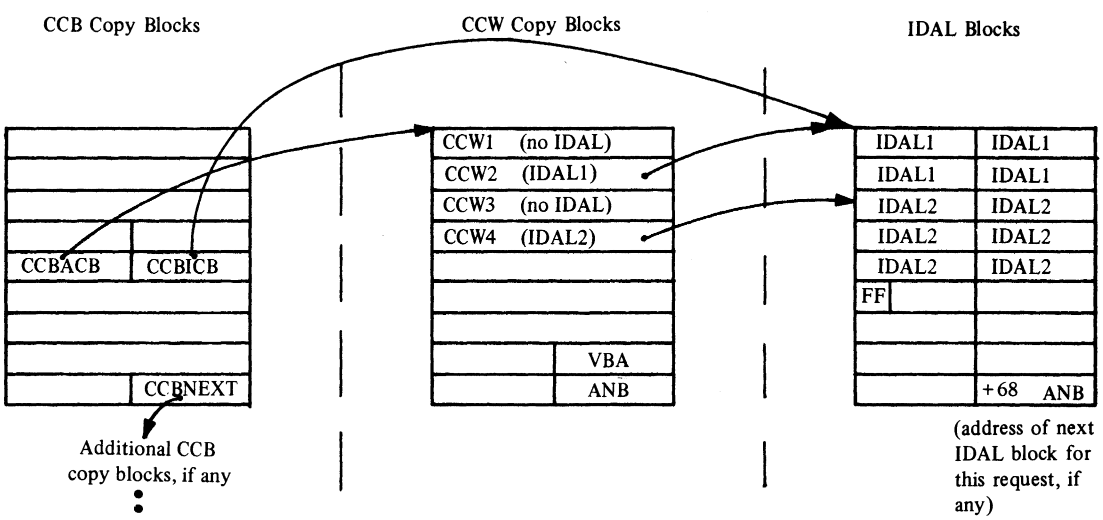


Figure 6.32. Relation of IDAL Blocks to Other Blocks

Notes: The X'FF' in the first byte of the 11th IDAW indicates the end of the IDAWs for the block. In this case, the IDALCNT field in the TCB would show seven free copy locations.

The data area of CCW2 crosses three page boundaries (may be up to 8K) and the data area of CCW4 crosses five page boundaries (may be up to 12K).

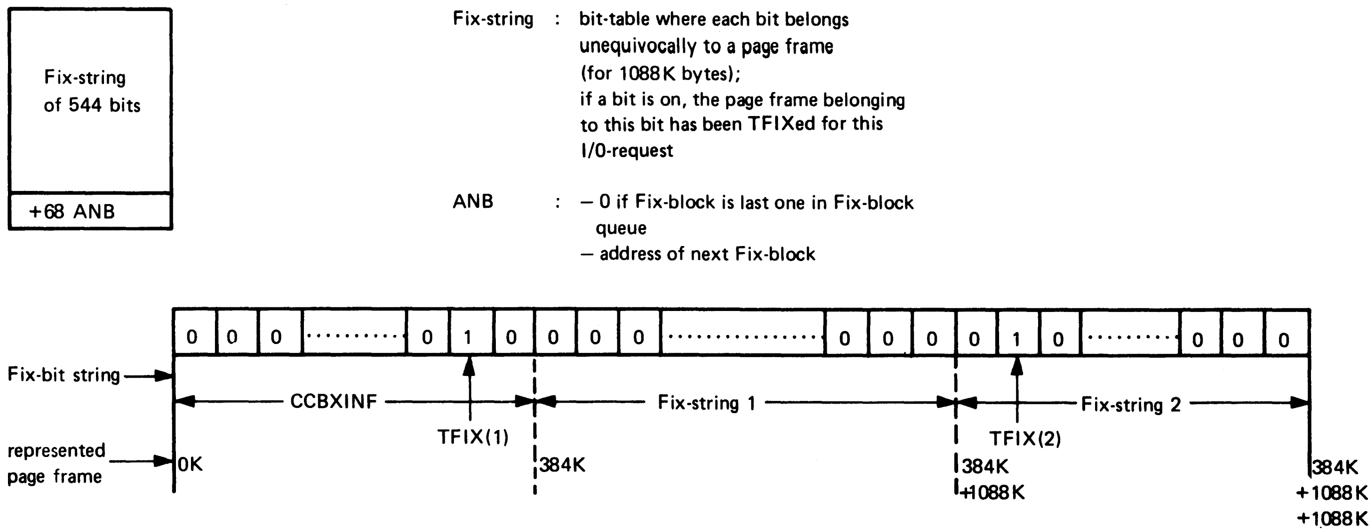


Figure 6.33. Fix Information Bit String and Block

- if for a specific page frame the Fix-bit is already on, no TFIX-request is transferred to the page manager.
- the TFIX-blocks are freed after I/O-request has been posted complete.

COPYING AND TRANSLATING CHANNEL PROGRAMS

User channel programs are copied into the copy blocks described in the previous section by either the routine CCWTRANS (entered at CCWTRBT2 for BTAM channel appendage I/O request).

By way of initialization, the following is done before the actual copying and translation is begun:

- The TCB for the requesting task is initialized and enqueued to the used TCB queue. As part of the initialization procedure, the TCB pointers to the two special command lists for the device are filled in (see Figure 6.34).
- Two copy blocks are dequeued from the free copy block queue for the CCB copy block and the first CCW copy block.
- The CCB is copied and initialized so that the CCW address points to the first location in the first CCW copy block. The VBA in the first CCW copy block is set to the virtual address of the CCW the virtual CCB is pointing to (which is the virtual address of the first CCW to be executed).
- If a sense CCW was present, it is also copied into the CCB copy block and its data area is TFIXed in real storage (unless it crosses a page boundary, in which case an IDAL is built), and the address is translated.

The channel program is then copied and any necessary IDALs are built. The channel programs translated can be divided into three classes according to the types of commands they contain. They are described in the following order:

1. Channel Programs without TIC or Status Modifier Commands.
2. Channel Programs with TIC Commands.
3. Channel Programs with Status Modifier Commands.

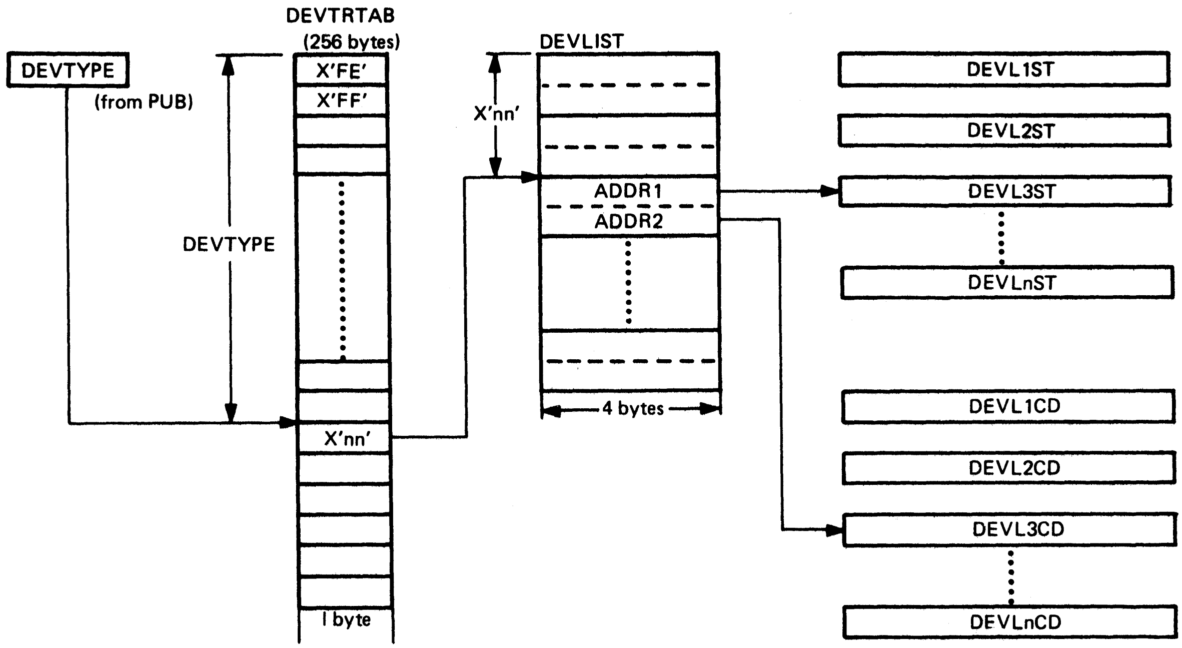


Figure 6.34. Initializing Special Command List Pointers in TCB

DEVTYPE: device type code from PUB

Entries in DEVTRTAB:

X'FF' = unsupported device.

X'FE' = device does not support status modifier commands or control commands with data area.

X'nn' = displacement to entry in DEVLIST if device supports status modifier commands and/or control commands with data area.

DEVLIST: list of pointers to the special command lists. The two entries (if any) for the device on which the I/O is requested are moved to the TCB when this is initialized.

DEVLnST: status modifier command list for device type n.

DEVLnCD: control command with data area list for device type n.

DEVLnST and DEVLnCD are bit strings. When a CCW is copied, the command code is used to refer to a bit in these strings. By testing this referred bit it is determined whether a CCW is a status modifier command or a control command with data area, or does not belong to these categories.

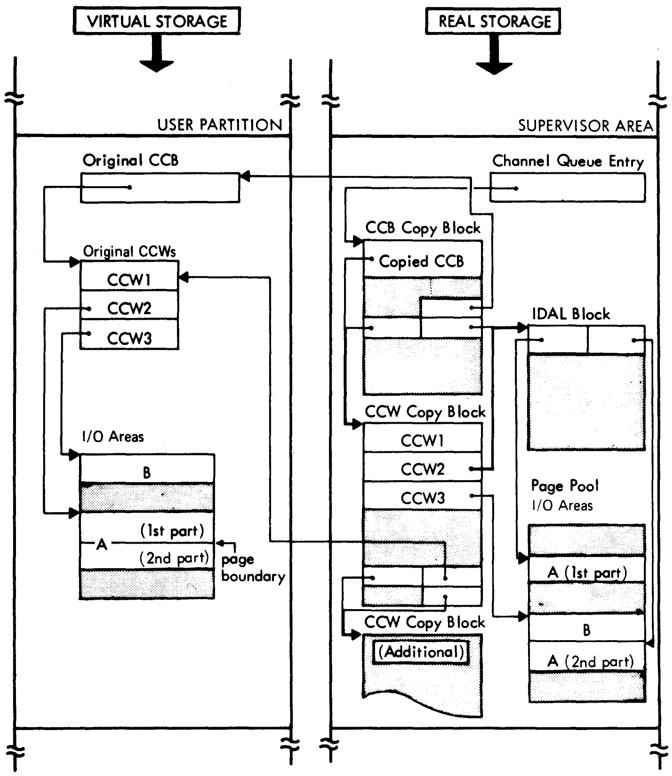


Figure 6.35. Schematic Representation of Channel Program Translation

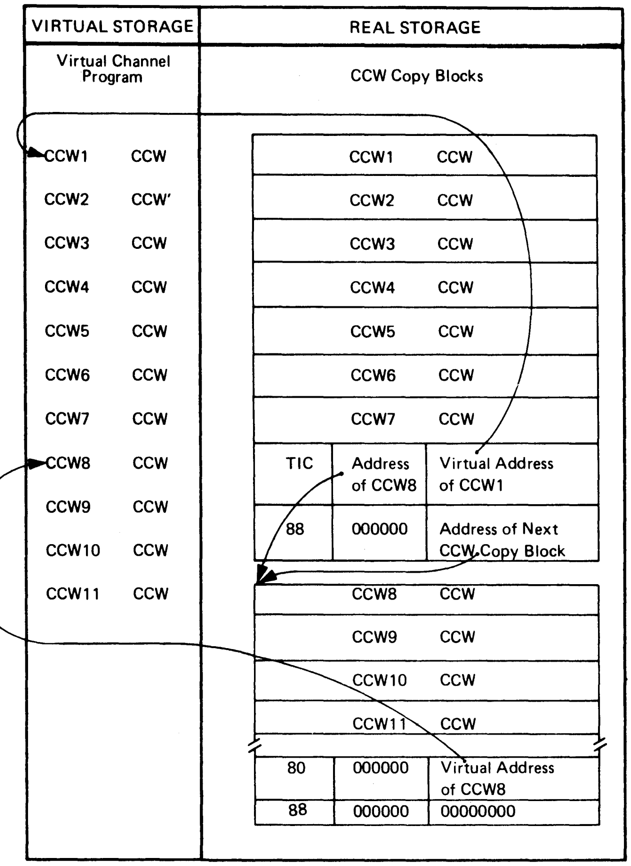


Figure 6.36. CCW Translation for Channel Programs Without TIC or Status Modifier Commands

Copying Channel Programs without TIC or Status Modifier Commands

The first CCW in a channel program is always copied into the first copy location pointed to by the copied CCB. If command chaining or data chaining is specified in the CCW the following chained CCWs are copied into successive copy locations.

If a program of chained CCWs should contain 8 or more commands, a new CCW copy block must be used. The eighth copy location of the first copy block is then converted into a TIC command pointing to the first location of the next copy block. The VBA of the next copy block is set to the virtual address of the eighth chained CCW.

Figure 6.35 is an example of a copied channel program containing 11 chained CCWs.

Copying Programs containing TIC Commands but no Status Modifier Commands

A TIC command (Transfer in Channel) command is, when encountered, copied into the next copy location just as any other chained command is. Although a TIC is 8 bytes long, only the first 4 bytes have any meaning (the command code and transfer address). The second four bytes of the copied TIC are set to zero. These bytes are used as a chain pointer for TICs which follow status modifier commands (refer to the section "Copying Status Modifier Commands"). The command code of a copied TIC is set to X'08' (standard user TIC).

The virtual storage location pointed to by the TIC command must be mapped into a location in the copied channel program. This mapped location is then placed in the copied TIC (unless the copied TIC is the first location of a copy block, in which case the address is placed in the end-of-block TIC (eighth copy location) of the previous copy block) and used as the copy location for the CCW pointed to by the TIC. The mapped location is determined in the following way:

- If the CCW pointed to by the TIC command has a copy location in an existing copy block (i.e., there is a block such that the virtual CCW address lies between the block's VBA and the block's VBA+56), place the location thus found in the TIC and copy the CCW in the location if it is free. If the location is not free, go to the translation termination routines. Figure 6.37 is an example of a TIC which points to an already existing copy location.

- If there is no existing copy location a new CCW copy block must be enqueued. The new block is enqueued at either end of the existing queue or between two existing blocks depending upon where the virtual address in the TIC is in relation to the VBAs of the existing blocks. Figure 6.38 shows how a new CCW copy block is queued to provide a copy location for a CCW pointed to by a TIC. Once enqueued, the VBA of the new copy block must be determined. If at all possible, the new block will be aligned to the one either above or below it (the VBA is 56 greater than the VBA of the lower block or 56 less than the VBA of the upper block). This is only possible if the address pointed to by the TIC lies within one of the ranges (i.e., is less than 56 below the VBA of the above block or less than 112 above the VBA of the block chained below). If possible to align to both blocks the alignment is made to the lower block. Considering the example in Figure 6.38 again it is seen to be possible to align the new block to the lower copy block and place the CCW to be copied in the 4th copy location.

- If it is possible to align the new block to both the upper and lower blocks but not to both at the same time (the difference between the VBAs of the two blocks is less than 112), a short block must be created by moving the end-of-block indicators to the copy location following the last logical copy locations. Figure 6.39 shows how a short block is enqueued.

- If no alignment of the new block with either of its neighbors is possible, the VBA of the new block is made equal to the virtual address pointed to by the TIC and the first copy location in the block is used. Figure 6.40 shows such a copy block being enqueued.

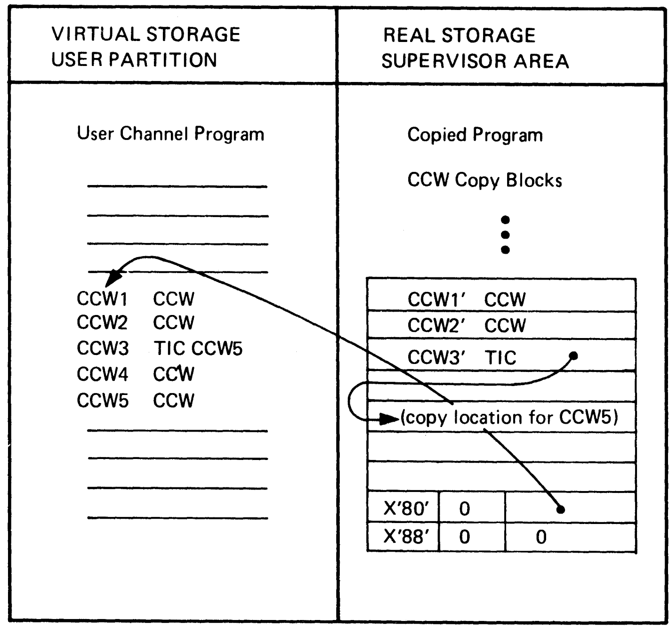
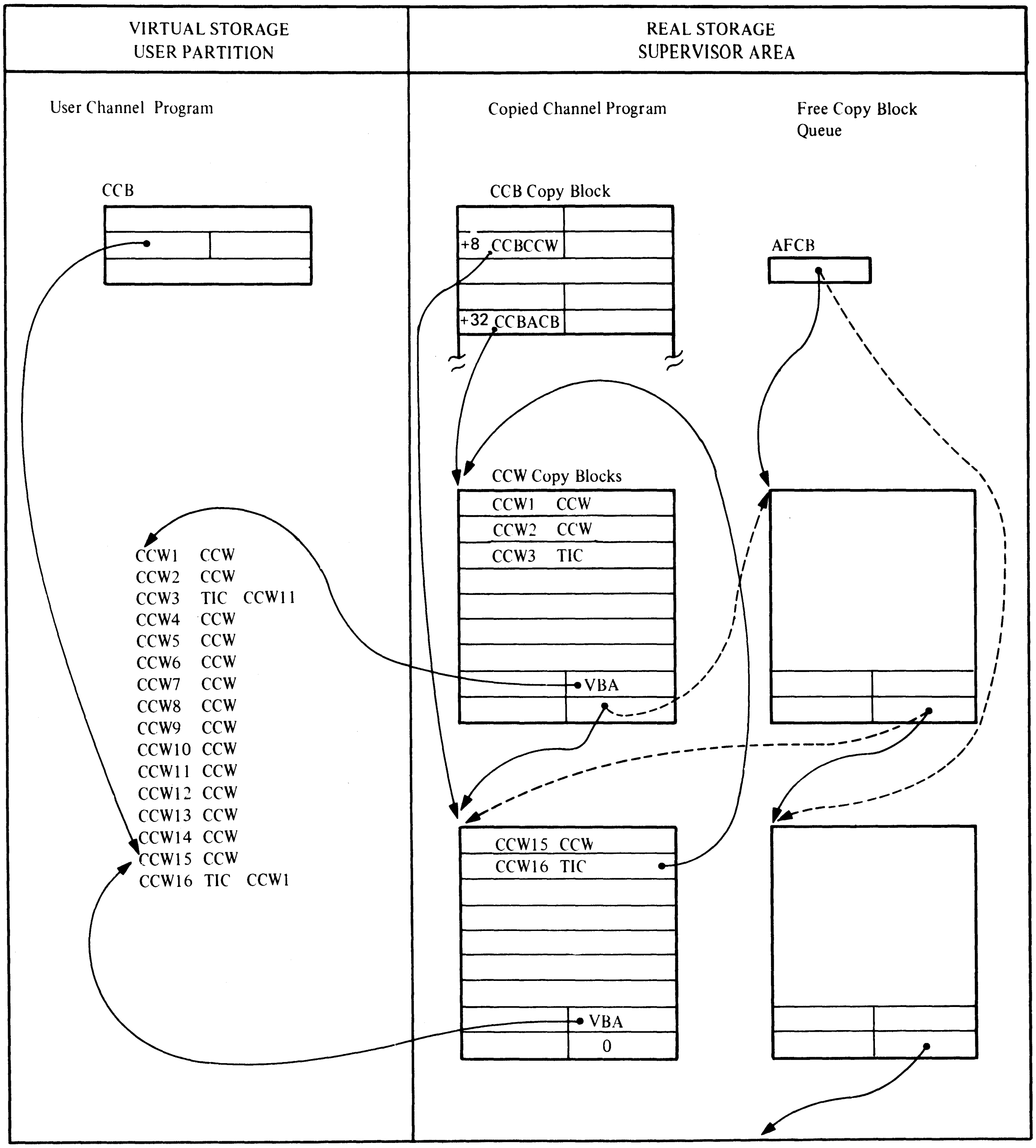


Figure 6.37. Locating a copy location for a CCW pointed to by a TIC when the location is in an already used copy block

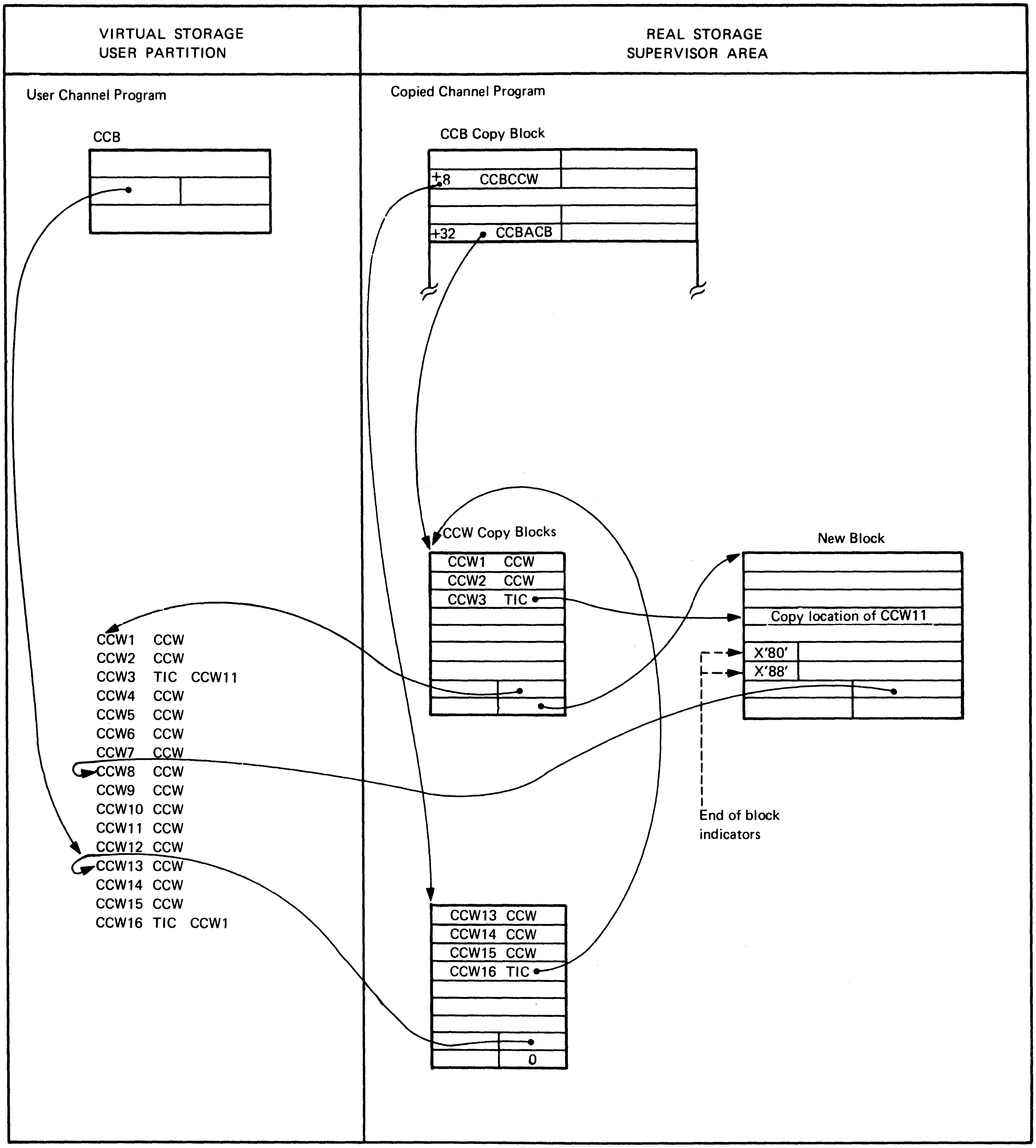


CCW3 has just been copied. The problem is to find the copy location for CCW11.

Free copy block is queued between A and B because the address used by the TIC at CCW3 lies between the VBA for A and the VBA for B. The solid line shows the conditions before the new block is enqueued and the dotted lines the conditions afterward.

Once enqueued the VBA in the newly enqueued block will point to CCW8 (the block is aligned to the next lower block) and the TIC in CCW3 will point to the fourth copy location in the new block. Copying will then continue with CCW11 being copied into that location.

Figure 6.38. Enqueueing a new copy block to the correct location in the CCW copy block chain to handle a CCW pointed to by a TIC



CCW3 has just been copied and the copy block for CCW11 has been enqueued. The problem is to align the block.

Solution: Make the new block a 'short' block in that the end of block indicators are moved to the copy position following that for CCW12.

Figure 6.39. CCW Copy block queueing calling for the creation of a "short" block to maintain alignment

Copying Status Modifier Commands

Status modifier commands add another level of complexity to the copying process, because they may transfer control to either of the next two following CCWs depending upon the result of the status modifier's operation. If, for example, a SEARCH command is unsuccessful, control is transferred to following CCW. If it is successful, on the other hand, the following CCW is skipped and control is passed to the second following command.

This characteristic makes it possible to create tree structures. Consider the following chain of commands:

```
      READ
      READ
      SEEK
      SEARCH
      TIC    A
      READ
      READ
A     WRITE
      WRITE
      SEARCH
      TIC   B
      READ
      READ
B     READ
      READ
```

If the first SEARCH in this program is successful, no branch is taken as the TIC command is skipped. If the SEARCH is not successful the chained commands beginning at A are executed. The same is true when the second SEARCH is encountered. This can be done any number of times in a program. Since a program is copied as it is executed, the presence of status modifier commands makes it necessary to take several passes through a program in order to cover all the possible branches.

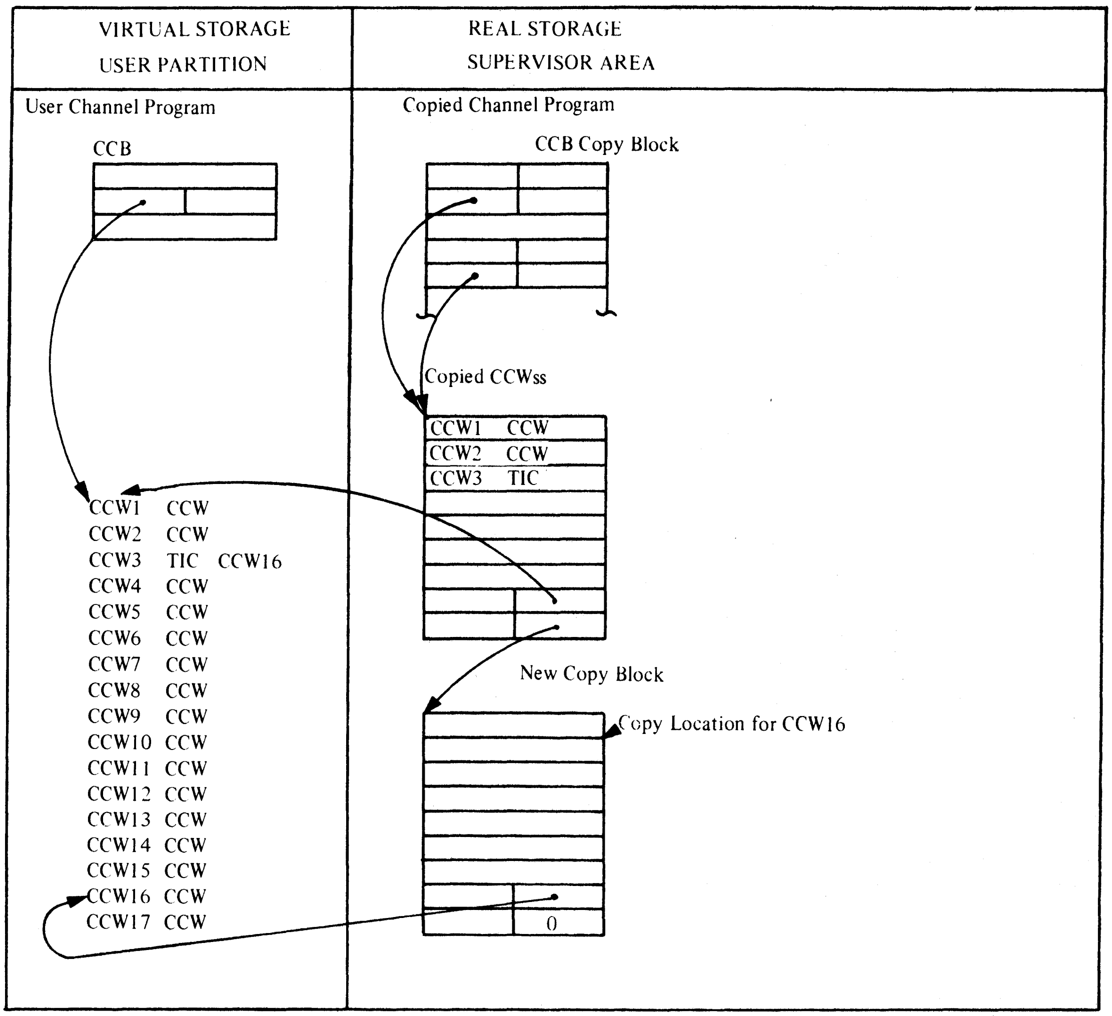
In the first pass through a program, a TIC following a status modifier command is copied but otherwise ignored (unless the status modifier is copied into the last copy location of a copy block). The TICs thus encountered are queued in a line pointed to by LINEPTR in the TCB (the queueing addresses are in the second 4 bytes of the copied TICs). Figure 6.41 shows a program with status modifier commands after the first pass has been made at copying it.

If a status modifier command happens to be copied into the last copy location of a block, an entry in a different queue is made. This contains as entries the last locations of blocks where a status modifier command is copied into the last copy location. The first entry in the queue is pointed to by BENDPTR in the TCB. The queuing addresses are in bytes 1-3 of the queue elements (last location of the CCW copy blocks concerned). Copying continues with the first CCW following the status modifier command being copied into the first location of the next queued copy block, and, if chained, copying continues with the following command. If, as is usually the case, the first command after the status modifier command is a TIC, the branch taken by the TIC command is copied. Figure 6.42 shows how a program with a status modifier command in the last copy position.

As soon as an end is reached in copying a program (a command without data or command chaining is copied or a copy location for a command is already filled) the program checks to see if there are any members in the queue pointed to by LINEPTR or BENDPTR. The members of these queues are handled one at a time. See Figures 6.42 - 6.44. Note that LINEPTR and BENDPTR entries can be created while others are being handled. Translation is complete when both LINEPTR and BENDPTR are zero (i.e., no more entries in either queue).

Translating Data Addresses and Page Fixing

Parallel to the copying of a channel program, the pages containing the data areas for the various CCWs are TFIXed in real storage and the virtual addresses of the data areas are translated into real addresses. If the user program has an IDAL (indirect data address list) or if an I/O area crosses one or more page boundaries and IDAL must be built for the copied channel program. IDALs are first built using the virtual addresses of the beginning of the data area and the page boundaries. When the individual pages are TFIXed in real storage these addresses are replaced with the correct real addresses. Figure 6.45 shows an IDAL built for a data area both before and after the pages have been TFIXed. Figure 6.46 shows how the IDAL looks if the command is a read backward command.



CCW3 has just been copied and it is necessary to find a copy location for CCW16, the next CCW copied.

Solution: Enqueue a new copy block behind the first one and use the first copy location for CCW16 because it is impossible to align the new block to an existing block.

Figure 6.40. Enqueueing new copy block which cannot be aligned to existing block because CCW too far removed from VBA of any existing block

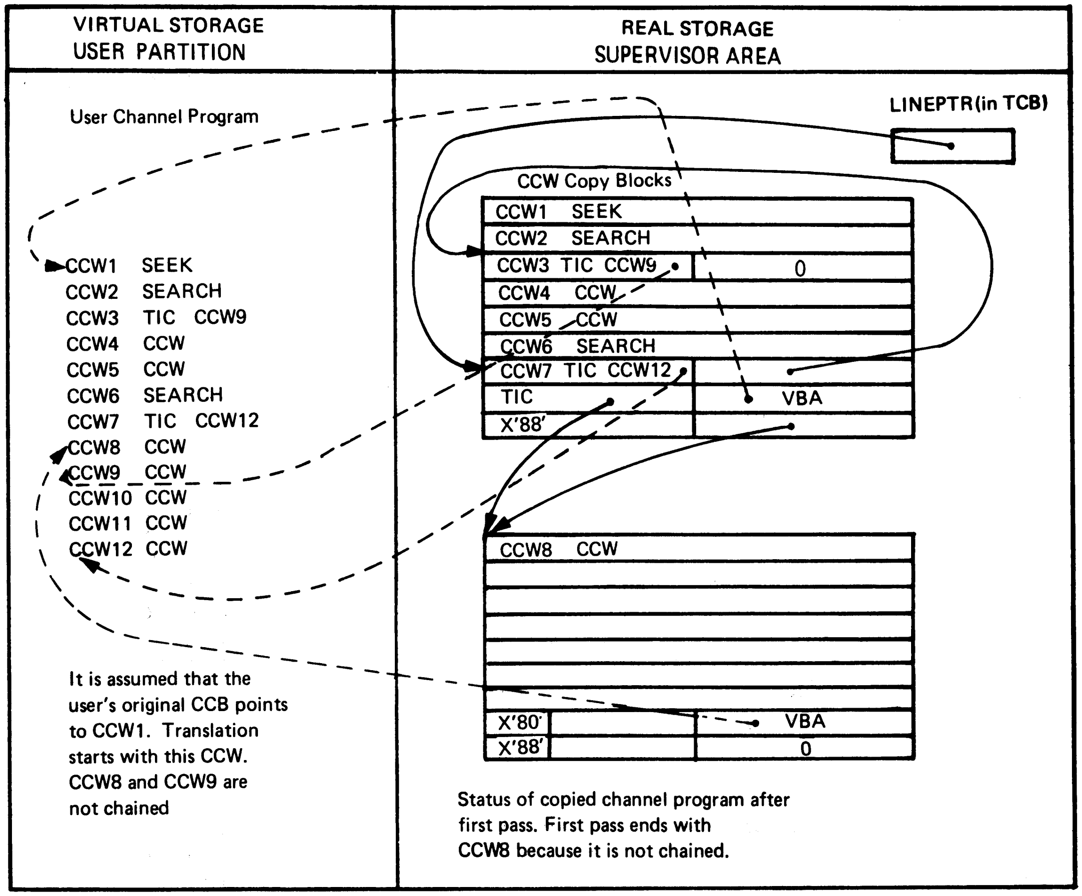


Figure 6.41. A channel prcgram containing status modifier commands after the first pass has been made at copying it

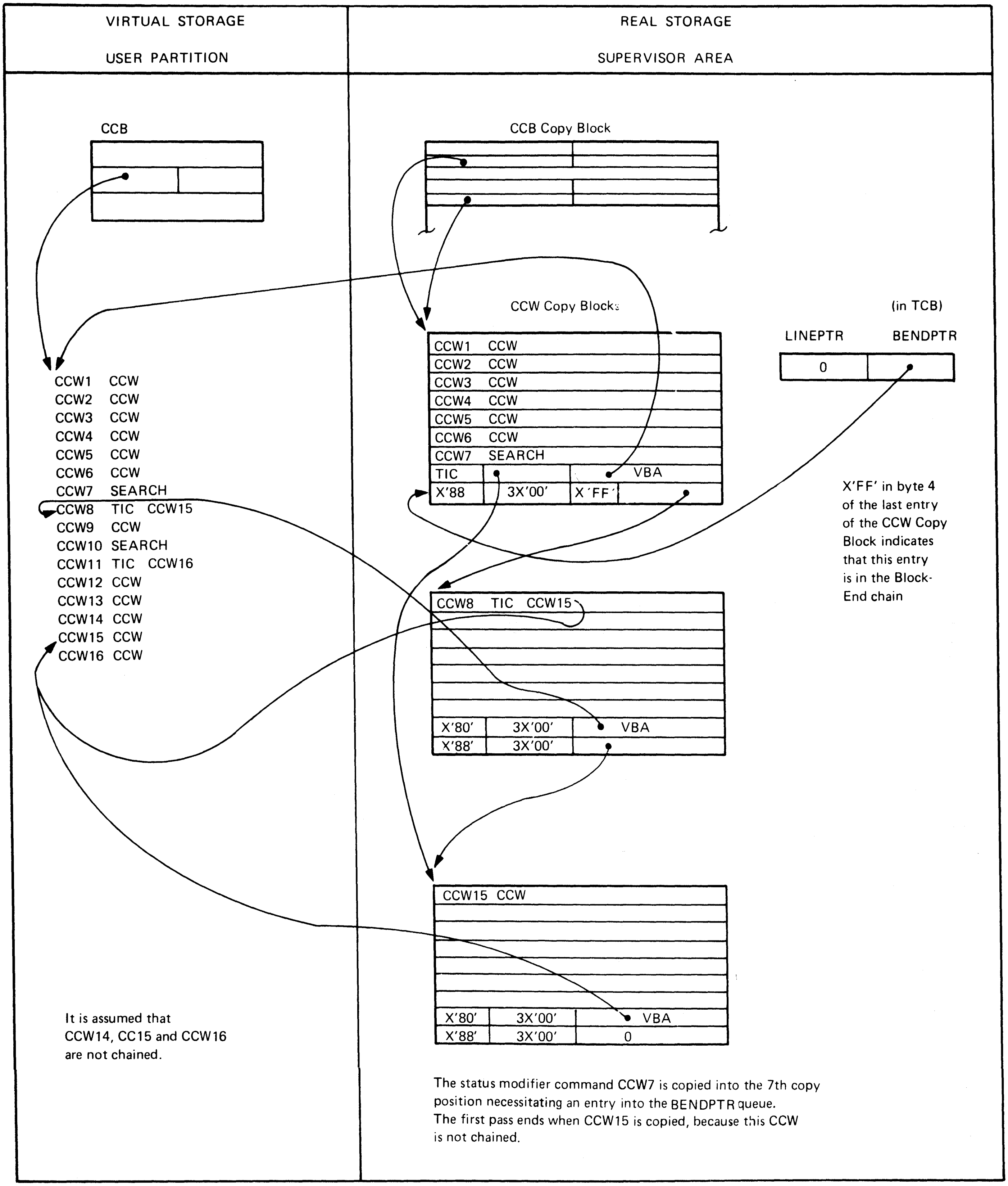


Figure 6.42. Channel program containing status modifier commands after the first pass has been made

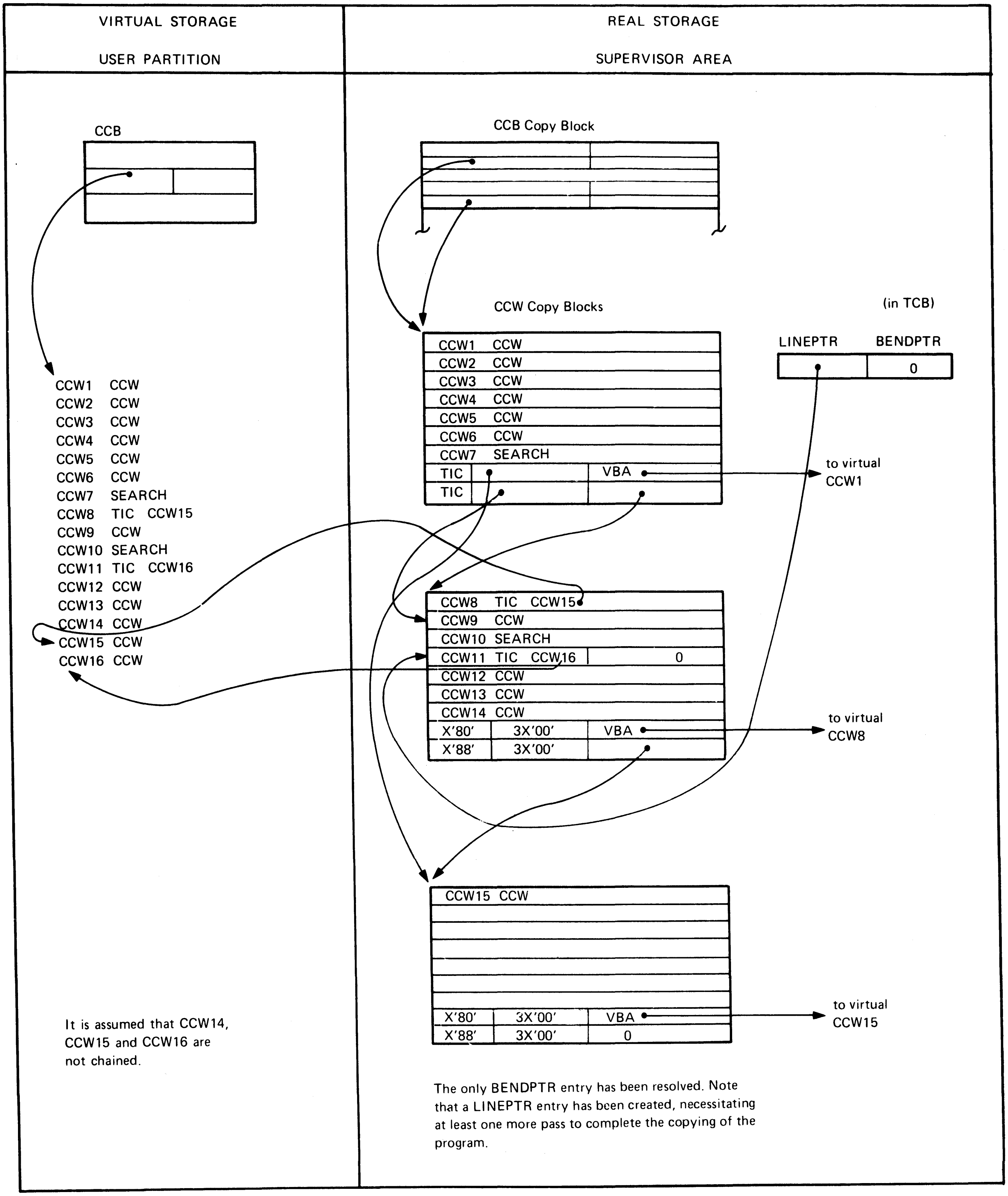


Figure 6.43. The channel program containing status modifier commands after the second pass has been made

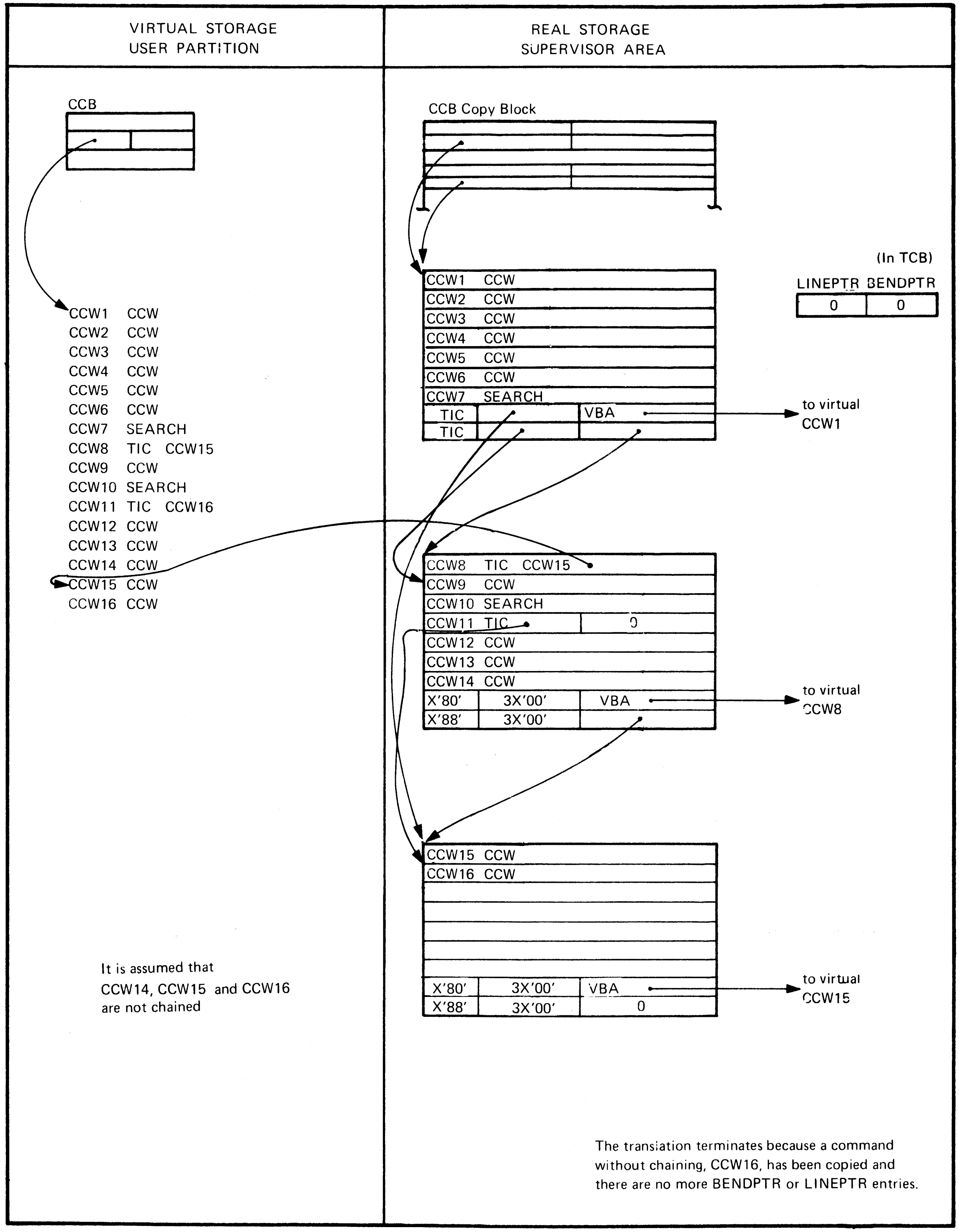


Figure 6.44. The channel program containing status modifier commands after translation has been completed

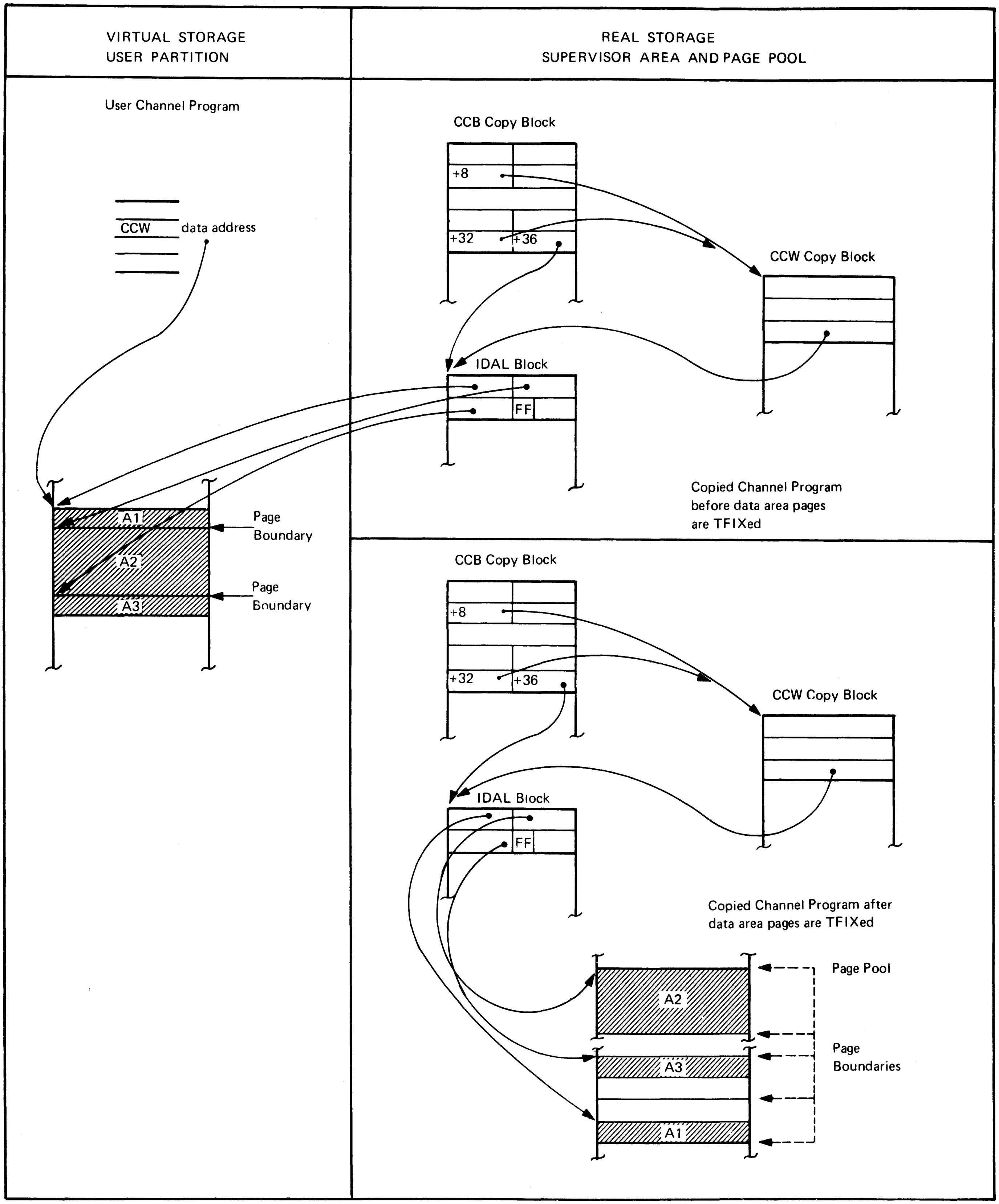


Figure 6.45. Copied CCW requiring an IDAL to be built (normal READ or WRITE command)

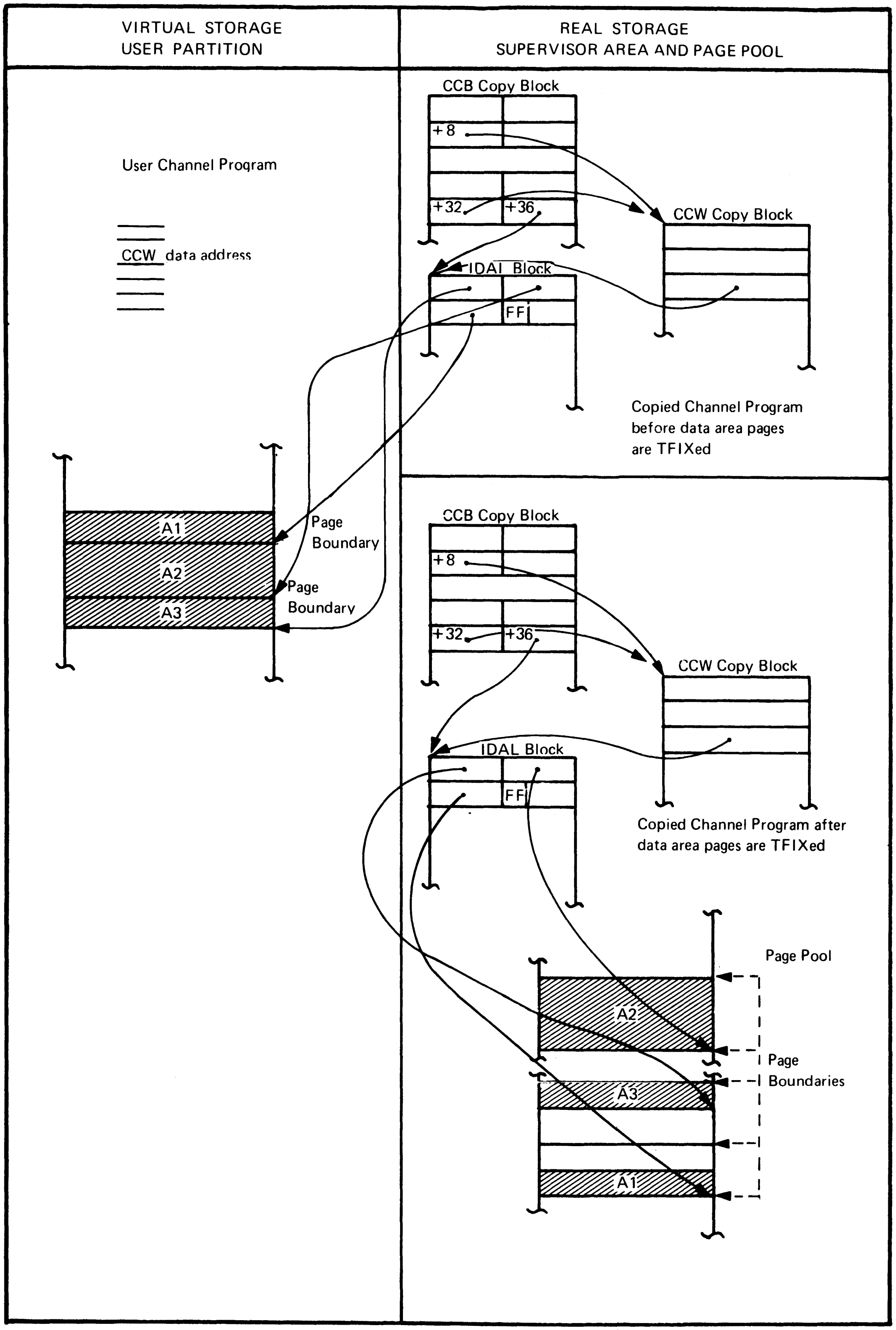


Figure 6.46. Copied CCW requiring an IDAL to be built (READ Backward command)

Queueing an I/O Request

After the channel program has been translated, the request is queued in the channel queue and control is given back to the channel scheduler. If there is no free channel queue entry for the request, the task is put into the wait state with the indication that it is channel-queue-bound (first byte in PIB is set to X'79').

PROCESSING AFTER COMPLETION OF AN I/O REQUEST

At a normal end of channel condition for a translated channel program, the I/O interrupt handler transfers control to the routine CSWTRANS. This routine must then:

- Free all pages fixed for I/O areas.
- Retranslate the CCW address placed in the CCB at channel end to the correct virtual address.
- Release the CCW copy blocks, IDAL blocks.
- Move changed parts of the CCB to the virtual-mode program and release the CCB copy block. If the virtual CCB is not in real storage, the end of channel information is not copied to the virtual CCB by the routine CSWTRANS. Instead, CSWTRANS posts a bit in the tasks PIB (PIB DAT Flag = X'04') indicating to the dispatcher that the CCB should be moved before the task is dispatched. This is necessary because CSWTRANS may not cause a page fault and, therefore, cannot request that the virtual CCB be brought into real storage.
- Activate tasks waiting for copy blocks (set TRTBUF to X'83').

CSWTRANS returns control to the interrupt handler when it has finished processing.

BTAM CONSIDERATIONS

If a BTAM channel appendage is to be called after a BTAM request, the CCW address left in the CSW at channel end is retranslated by CSWTRBTM and returned to the I/O interrupt handler before the appendage is given control.

A BTAM I/O request is translated and copied in the same manner as normal I/O requests unless it comes from a BTAM channel appendage.

An I/O request coming from a BTAM channel appendage via the I/O Interrupt Handler must be translated without incurring any interruption (wait, page fault, etc.). In order to do this BTAM specifies the maximum number of copy blocks that will be needed by an appendage in addition to the number used by the original request. This number is contained in the residual count field of the CCB when an I/O request is made. After a BTAM request has been translated and before it is put in the channel queue, the additional blocks specified are taken from the free queue and saved so that they will be available when the request for the appendage is made.

All of the copy blocks used by BTAM (except the CCB copy block) are first freed when a translation request for the BTAM channel appendage comes. They cannot be lost however, because no other task can gain control of the CCW translation routines when an appendage routine is being processed. The special BTAM TCB is used for a BTAM channel appendage I/O request.

Once the channel program for an appendage has been translated, control is returned to the I/O interrupt handler. There is no need to enqueue the request in the channel queue since the original request has not been dequeued.

TRACK-HOLD FACILITY

The Track-Hold Table (THTAB) is similar to the Channel Queue Table: when a task requests a track to be protected, an entry in the Track-Hold Table is filled in the same way as a channel queue entry for an I/O request. Requests for more tracks to be held on a specific device are chained in the same manner as a device chain in the channel queue. For the format of the Track-Hold Table, see Figure 6.48.

Track-Hold Without Seek Separation

(FOPT macro TRKHLD=n, SKSEP=NO)

Figure 6.47 shows the initialized significant bytes of the Track-Hold mechanism.

Figure 6.49 illustrates the pointers and table entries after several Track-Hold requests have been issued. Referring to this figure, if task aa releases track 1A (entry No. 0) or track 2A (entry No. 3), the way the pointers are then exchanged may be compared with dequeueing an entry in the channel queue.

FREE LIST POINTER (THFLPTR): 00 → Entry No. 0

PUBs TRACK HOLD POINTERS (PUBOPTN): FF, FF, FF, FF, FF

TRACK-HOLD TABLE ENTRIES

Entry No.	Chain Byte	CCB Addr.	Track Addr. (BBCCHH)	Req. ID	Flag and Counter
0	01	zeros	zeros	zeros	zeros
1	02	zeros	zeros	zeros	zeros
2	03	zeros	zeros	zeros	zeros
3	04	zeros	zeros	zeros	zeros
4	05	zeros	zeros	zeros	zeros
5	06	zeros	zeros	zeros	zeros
6	07	zeros	zeros	zeros	zeros
7	08	zeros	zeros	zeros	zeros
8	09	zeros	zeros	zeros	zeros
9	FF	zeros	zeros	zeros	zeros

Figure 6.47. Track-Hold Table:
Initial contents of significant bytes used in responding to track-hold requests

When a task requests a hold on a track that is already held by another task, the high-order bit of the flag-and-counter byte is turned on (for example, entry No. 1 in Figure 6.49). When a task requests a hold on a track that it already holds itself, the flag-and-counter byte is incremented by one (for example, entry No. 4 in Figure 6.49).

On release of a track by the holding task, and provided the counter is zero before the release, any task or tasks that are waiting for that track are brought out of the wait state. The supervisor then returns to task selection and, if the next selected task was waiting for this track, its hold request is honoured. Any other task or tasks that were waiting for the track now remain ready-to-run, but if such a task gains control before the track has again been released, that task returns to the wait state.

If the counter is more then zero before the release, then only the counter is decremented by one so that the track remains held by the same task. For illustrations of these operations, compare Figures 6.49 and 6.51, entry Nos. 1 and 4.

Figure 6.50 summarizes the sequence of events leading to the situations shown in Figures 6.49, 6.51, and 6.52.

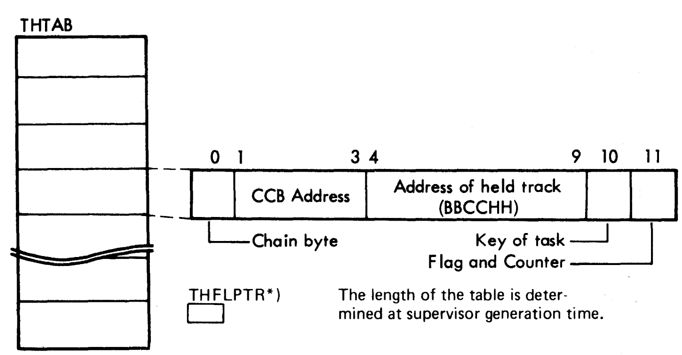


Figure 6.48. Track-Hold Table (THTAB)

Bytes:

0: Initially, pointer to next sequential entry or X'FF' (table delimiter) in last entry. After requests have been issued, this byte points to the entry for the next request for a track on the same device, or contains X'FF' (in entry for the last request), or it is a pointer in the free list chain.

1-3: Address of CCB associated with the task requesting the hold.

4-9: Disk address of the track being held (in the form BBCCHH).

10: Key of the task owning the track.

11: Bit 0=1: Indicatess a task is waiting for this track

1-3: Unused

4-7: Counter of number of holds on the track. (The number of holds is one more than this value.)

*)Note:

THFLPTR: The Track-Hold Free List Pointer (1 byte) contains the displacement within the Track-Hold Table of the first entry in the free list or X'FF' when the Track-Hold Table is full.

Bytes 77-79 (X'4D' - X'4F') of the System Communication Region (SYSCOM) contain the address of the Track-Hold Table. Label THTAB identifies the first byte of the table.

Byte 76 (X'4C') of the System Communication Region (SYSCOM) contains the address of the Track-Hold Free List Pointer. Label THFLPTR identifies the location of the pointer.

The half-word at THTAB-2 contains the total number of 12 byte entries comprising the Track-Hold Table.

Track-Hold with Seek Separation

(FOPT macro TRKHLD=n, SKSEP=n, or YES)

When the supervisor supports Seek Separation, the PUB points to the Seek Address Block (SAB) which contains a pointer to the Track-Hold Table (THTAB), but the operation of the Track Hold facility remains the same as without seek separation.

TRACK-HOLD TABLE ENTRIES

	THFLPTR	Entry No.	Chain Byte	CCB Addr.	Track Addr. (BBCCHH)	Req. ID	Flag and Counter
	05	0	01	xxx	Track 1A	aa	00
		1	02	xxx	Track 1B	aa	80
		2	C4	xxx	Track 1C	bb	C0
		3	FF	xxx	Track 2A	aa	00
	PUBs PUBOPTN	4	FF	xxx	Track 1D	aa	01
		5	C6	zeros	zeros	zeros	zeros
2nd Device	03	6	C7	zeros	zeros	zeros	zeros
	FF	7	C8	zeros	zeros	zeros	zeros
	FF	8	C9	zeros	zeros	zeros	zeros
1st Device	00	9	FF	zeros	zeros	zeros	zeros
	FF						

Figure 6.49. Track-Hold Table:
Task aa holding tracks 1A, 1B, and 1D (2 holds) on 1st device, and track 2A on 2nd device; Task bb holding track 1C on 1st device; A task is waiting to hold track 1B on 1st device.

SEQUENCE of REQUESTS \ TASKS:---	aa	bb	cc	Remarks
	Hold 1A			Entry queued.
	Hold 1B			Entry queued.
		Hold 1C		Entry queued.
	Hold 2A			Entry queued.
	Hold 1D			Entry queued.
	Hold 1D			Counter incremented.
			Hold 1B	Entry flagged and requestor put into wait state.

---The table entries and pointers at this stage are illustrated by Figure 6.49.---

SEQUENCE of REQUESTS \ TASKS:---	aa	bb	cc	Remarks
	Free 1B			Flag turned off, waiting task (cc) made ready-to-run, and task selection entered; if task cc is selected, its request for track 1B is honoured.
	Free 1A			Entry dequeued.
	Free 1D			Counter decremented.

---The table entries and pointers at this stage are illustrated by Figure 6.51.---

SEQUENCE of REQUESTS \ TASKS:---	aa	bb	cc	Remarks
	Free 2A			Entry dequeued.
	Free 1D			Entry dequeued.
		Free 1C		Entry dequeued.
			Free 1B	Entry dequeued.

---All tracks have now been freed as shown in Figure 6.52.---

Figure 6.50. Example of Tracks Held and Freed by Three Tasks

TRACK-HOLD TABLE ENTRIES

THFLPTR: 00 → Entry 0

Entry No.	Chain Byte	CCB Addr.	Track Addr. (BBCCHH)	Req. ID	Flag and Counter
0	05	xxx	Track 1A	00	00
1	FF	xxx	Track 1B	cc	C0
2	04	xxx	Track 1C	bb	C0
3	FF	xxx	Track 2A	aa	00
4	01	xxx	Track 1D	aa	C0
5	06	zeros	zeros	00	C0
6	07	zeros	zeros	00	C0
7	08	zeros	zeros	00	C0
8	09	zeros	zeros	00	C0
9	FF	zeros	zeros	00	C0

PUBs PUBOPTN

2nd Device	03
	FF
	FF
1st Device	02
	FF

Figure 6.51. Track-Hold Table:
Task aa has released holds on tracks 1B and 1A, and one of the holds on track 1D; Task cc has been brought out of the wait state and has been selected to run, so it now holds track 1B.

TRACK-HOLD TABLE ENTRIES

THFLPTR: 01 → Entry 1

Entry No.	Chain Byte	CCB Addr.	Track Addr. (BBCCHH)	Req. ID	Flag and Counter
0	05	xxx	Track 1A	00	00
1	02	xxx	Track 1B	00	C0
2	04	xxx	Track 1C	00	C0
3	00	xxx	Track 2C	00	00
4	03	xxx	Track 1D	00	C0
5	06	zeros	zeros	00	C0
6	07	zeros	zeros	00	C0
7	08	zeros	zeros	00	C0
8	09	zeros	zeros	00	C0
9	FF	zeros	zeros	00	C0

PUBs PUBOPTN

2nd Device	FF
	FF
	FF
1st Device	FF
	FF

Figure 6.52. Track-Hold Table: Situation after total release

SEEK SEPARATION

(FOPT macro SKSEP=n, or YES)

A supervisor that supports Seek Separation on DASD includes a Seek Address Block (SAB) table with a 5-byte entry for each DASD in the system. The SAB entry for a device is pointed to by the PUB and, if the Track-Hold facility is also supported (TRKHLD=n), byte 4 of the SAB points to the Track-Hold Table (THTAB) entry when a task requests a track to be protected.

The first four bytes of the SAB table entry contain the current disk address (BCCH) for the device. These bytes are set up at initiation of the first long seek (command code X'07') in the user's channel program.

The Seek Separation routine is bypassed for diskette and devices with the disconnect command chaining feature on block multiplexer channels, running in block multiplex mode.

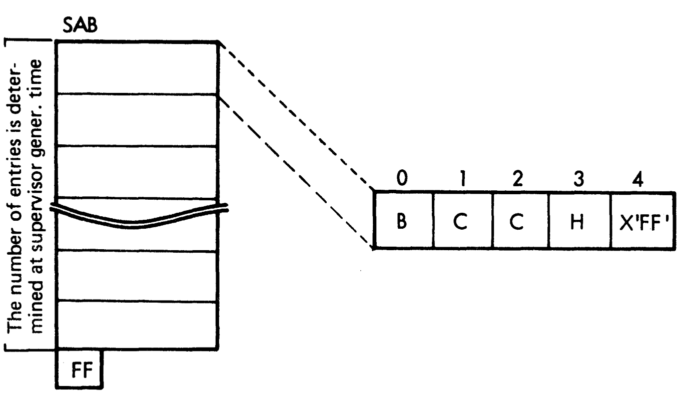


Figure 6.53. Seek Address Block Table (SAB)

Bytes:

0-3: Current Disk Address (BCCH)

4: X'FF' or: Pointer to the Track-Hold Table entry.

Note: A SAB table is only generated if Seek Separation has been specified at supervisor generation time.

Bytes 56-59 (X'38' - X'3B') of the System Communication Region (SYSCOM) contain the address of the SAB Table. Label SAB identifies the first byte of the table.

Executing the Channel Program

Before executing the channel program, the user's seek address is compared with the address in the SAB table entry. If they are the same, I/O is started immediately. Note that this Start I/O could be for a chain of commands or for an isolated user's seek.

If the addresses are unequal but the CCW is not chained (isolated user's seek), the 4-byte seek address is stored in the SAB table entry before starting I/O on the channel.

The following is the sequence of events when the seek address is different from the address in the SAB table entry and the Seek CCW is the first in a chain:

1. The user's Seek CCW Chain Bit is turned off.
2. The CCB 1st Communication Byte is saved.
3. The CCB 1st Communication Byte is made X'04' (Device-End Posting Required).
4. The CCB 3rd Communication Byte bit 7 is turned on (Seek Separation Active).
5. The 4-byte Seek Address is stored in the SAB table entry.
6. I/O is started on the channel.

Initialized Entry

0	1	2	3	4
00	00	00	00	FF

Entry with Seek Address Stored

0	1	2	3	4
B	C	C	H	FF

Entry with Seek Address Stored and with Pointer to Track-Hold Table Entry

0	1	2	3	4
B	C	C	H	xx

Figure 6.54. Seek Address Block (SAB) Table Entries

Figure 6.54 shows the format and contents of the entries.

At Channel End:

If I/O is pending for other devices, the channel is rescheduled. Otherwise, return directly to task selection.

Note: If unit check occurs when a separated seek is being executed, the user's CCW chain is restored and I/O is restarted. If a second unit check occurs, the original user's chain will then be available for ERP or dump analysis.

When Device End occurs:

1. The user's Seek CCW Chain Bit is restored.
2. The CCB 1st Communication Byte is restored.
3. The CCB 3rd Communication Byte is reset.
4. I/O is restarted on the channel, using the same long seek command.

This time the entire user's channel program is executed; the arm is already at the user's seek address and the CCW is chained to the next command.

CONSOLE BUFFERING

(FOPT macro CBF=n)

If the FOPT macro parameter CBF=n was included, the supervisor supports buffering of output to the console printer (unless MODEL=125). This enables the supervisor to return immediately to the caller when a write I/O command is issued.

The console buffering routine is bypassed, however, if one of the following conditions exists:

1. The I/O request is not a write.
2. Device-end posting is requested by the user.
3. Unrecoverable I/O errors can be accepted by the user.
4. The user has his own I/O routine.
5. The user wants sense information returned.
6. The CCW or data are not in the user's partition.
7. The data length is more than 80 bytes.
8. The user's CCW is chained.
9. The user's CCW PCI bit is on.

The Console Buffer Table (see Figure 6.55) is a 104 multiplied-by-n bytes area of main storage, where n is 1 - 50. Each buffer entry in the table is 104 bytes. CBNEXT is a fullword constant that points to the next buffer entry. CBNEXT is initialized with the address of CBTAB and is incremented by 104 every time a buffer is used, so that it points to the next entry. When its value becomes greater than CBEND, CBNEXT is reinitialized with the value of CBTAB.

Each entry in the console buffer comprises three areas:

1. The Console Buffer CCW (CBCCW): an area in CCW-format whose data address field always points to the Console Buffer Data (CBDATA) area.
2. The Console Buffer CCB (CBCCB): an area in CCB-format whose CCW address field (CCB bytes 9-11) always points to the Console Buffer CCW (CBCCW).
3. The Console Buffer Data (CBDATA) area: an output area to which the user's output data is moved.

When the console buffering routine is entered from the channel scheduler routine, provided the next buffer entry (pointed to by CBNEXT) is free, the command code, flag byte, and byte count in the user's CCW are moved to the CBCCW, the user's output data is moved to the CBDATA area, and the user's symbolic unit address to the CBCCB area in the buffer entry.

If the next buffer entry is not free (CBCCB traffic bit not yet posted) return is made to the user to reissue the SVC. The user is posted console buffer table entry bound (X'7D').

The console buffering routine posts the traffic bit in the user's CCB and moves the CBCCB address to general register 1 before returning to the channel scheduler routine.

At channel end, the I/O interrupt handler sets the traffic bit in the CBCCB.

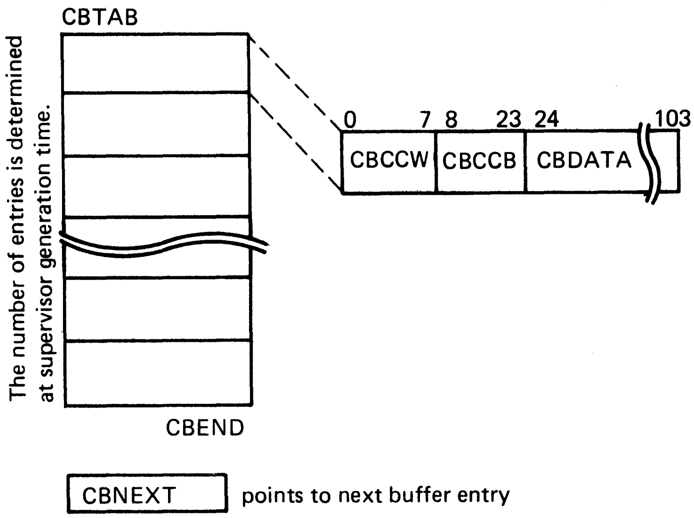


Figure 6.55. Console Buffering Table (CBTAB)

Bytes:

0-7 CCW: Command code, chain byte, and count field copied from the user's CCW. The data address is always the address of the data portion of the buffer entry.

8-23 CCB: The CCB used by the console buffering routine; the CCW address is always the address of the CCW in the buffer entry.

24-103 DATA: Contains the data moved from the requestor's output area.

Label CBTAB identifies the first byte of the Console Buffer Table.

ERROR QUEUEING AND DEQUEUEING

The error queue is a series of entries containing information required by supervisor resident or transient routines for analysis in attempting recovery. The last entry in the queue is used only by the resident routines in the supervisor. Whenever exit has to be taken to a transient routine, this entry is moved up to the first free entry and is used by the transient when it becomes the first in the queue.

In a supervisor without MPS, there are at least three error queue entries. With MPS, the minimum number of entries is five, but any number from 5 to 25 may be specified by the ERRQ parameter in the FOPT macro at supervisor generation time.

When an error occurs, the supervisor checks the availability of the last error queue entry. If available, the last entry is filled with the current error information but it is not flagged active. If recovery is accomplished entirely by resident supervisor routines, this entry is immediately available in case another error occurs. If the last entry is not available, the task is canceled due to error queue overflow.

When exit is to be taken to a physical transient routine, the contents of the last error queue entry are moved to the first free entry which is then flagged active. The last entry therefore remains inactive unless it is the only free entry at the time this decision is taken.

The physical transients serve each successive first entry in the queue. When error recovery is completed, control is returned to a resident exit routine which dequeues the first entry by moving all other entries upwards one position. The last entry is made inactive in case the queue was full or the bit was turned on by overlay.

See Figures 6.56 and 6.57 for operation of error queue and format of the entries.

			1ST ENTRY	2ND ENTRY	LAST ENTRY
A.	1st Error Information entered into Last Error Queue Entry.	Status	Inactive	Inactive	Inactive
		Contents	Free	Free	1st Error Information
B.	1st Error Information moved to First Free Entry for use by Transient		Active	Inactive	Inactive
			1st Error Information	Free	Free
C.	1st Error Information being used by Transient; 2nd Error Information entered into Last Error Queue Entry.		Active	Inactive	Inactive
			1st Error Information	Free	2nd Error Information
D.	1st Error Information being used by Transient: 2nd Error Information moved to First Free Entry for use by Transient.		Active	Active	Inactive
			1st Error Information	2nd Error Information	Free
E.	1st Error Recovered by Transient, 1st Entry Dequeued: 3rd Error Information entered into Last Error Queue Entry.		Active	Inactive	Inactive
			2nd Error Information	Free	3rd Error Information
F.	2nd Error Recovered by Transient, 1st Entry Dequeued; 3rd Error Recovered by Resident Routine.		Inactive	Inactive	Inactive
			Free	Free	Free

Figure 6.56. Operation of Error Queue with Three Entries

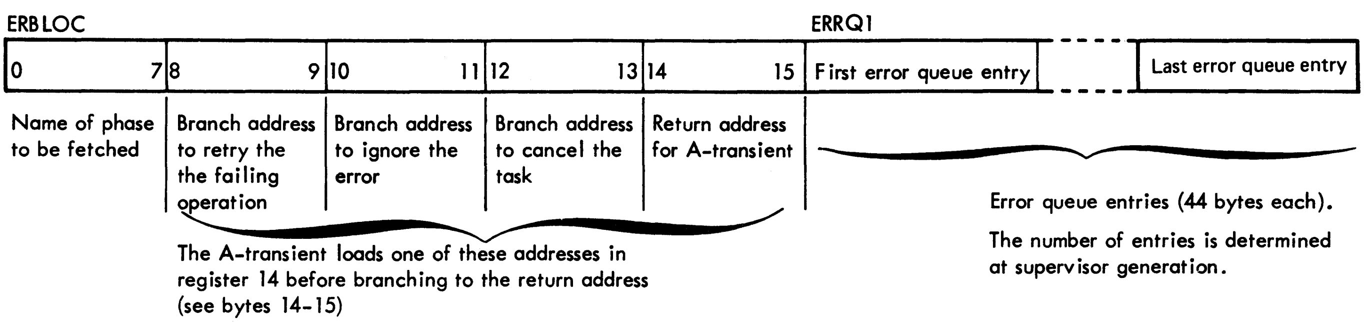


Figure 6.57. Error Recovery Block (ERBLOC) and Error Queue Entry

Layout of any Error Queue Entry

Bytes:

0-7 CSW

8-9 Address of PUB for device in error

10: Flag byte:

Bit 0: 1 = No record found on DASD
1: 1 = Intervention required
2: 1 = Passback (Set by device ERP)
3: 1 = Allow ignore
4: Not used
5: 1 = Allow retry
6: Not used
7: 1 = Active entry

11: Message code: may refer to a device error recovery message generated by physical IOCS or:

This location may contain one of the following:

X'E2' = The error is recoverable

X'AE' = A record is to be recorded on the system recorder file for SVC 44 or a BTAM appendage routine, and a physical transient is to be fetched (last two characters of phase name are in bytes 20-21).

12-15: Disk seek address

16-19: Address of CCB

20-43: Sense data: The number of sense bytes generated depends on the options specified; the minimum is 24 bytes, or:

Alternate entry name: If byte 11 contains X'AE', bytes 20-21 contain the last two characters of the phase name of the physical transient to be fetched for SVC 44 (A3) or BTAM (A5). X'AF' in byte 22 indicates that the I/O area associated with an alternate entry has been fixed temporarily.

Note: The address of the Error Block can be found in SYSCOM at displacement 0 (X'00').

LOADING ERP PHYSICAL TRANSIENTS

ERP is a System Task, and when exit is to be taken to a physical transient a switch is set in the SCP Communication Region (SYSCOM) to indicate the initial selection of the ERP task. The initial entry in the error queue (last entry) is moved to the first free entry, which is then made active, and return is made to task selection in the General Exit Routine.

When the entry has become the first in the error queue and the ERP system task is selected, the initial selection switch in SYSCOM is turned off and pointers to the current system task and to the error PUB are set up.

The Fetch Routine is then entered and the ERP task is made to wait until the required transient has been loaded. The Fetch Routine makes the Supervisor System Task Dispatchable.

On completion of the Fetch, the SUPVR bit is turned off and the ERP system task is reactivated. The physical transient phase that was read into the physical transient area is then entered.

To fetch another physical transient, an ERP transient issues a SVC 5. The Fetch Routine is then entered to deactivate the ERP task while the Fetch is in progress. On completion of the Fetch, the ERP task is reactivated and the fetched phase is entered.

DISK ERROR RECOVERY

Disk error recovery routines are the only resident device error recovery routines. They are described below. A-transients are only fetched when the error is to be recorded, or when an operator message is required.

For all other devices error recovery and recording is performed by A-transients. These transients are fully described in the DOS/VS Error Recovery and Recording Transients Logic manual.

2311 AND 2314 DASD ERROR RECOVERY

This routine is entered at label DISKERR. The following list gives the conditions tested for. Refer also to DOS/VS Error Recovery and Recording Transients Logic.

- CSW Bit 44 - Channel Data Check.

  Action: One retry. If error persists, exits to disk ERP transient and ERP message writer with the allow retry switch on.

  Message: 0P28 CHAN DTCHK.

- Sense Byte 0, Bit 3 - Equipment Check.

  Action: Exits to disk ERP transient and ERP message writer with the allow retry switch on.

  Message: 0P10 EQUIP CHK.

- Sense Byte 1, Bit 4 - No Record Found.

  Action: Tests for byte 1, bit 6 (Missing Address Marker). If present, executes restore command and exits to retry operation. If the error still persists after 10 retries, exits to disk ERP transient and ERP message writer with the allow retry switch on. If not present, reads Home Address and compares it to user's Seek Address. If equal, posts No Record Found to the CCB and exits to continue processing. If not equal, the condition is treated as a Seek Check.

Message: 0P21 NRF - MADDMK (No Record Found/Missing Address Marker), or 0P15 SEEK CHECK (Home Address unequal to Seek Address).

Note: Home Address is read, and the track address is provided for the error message. For other errors, the track address is obtained from the user seek address if error occurs during channel program execution.

- Sense Byte 0, Bit 7 - Seek Check.

  Action: If byte 0, bit 0 (command reject) is on, exits to disk ERP transient and ERP message writer to issue message after which the task is canceled. Otherwise, executes restore command and exits to retry the operation. If the error still persists after ten retries, exits to disk ERP transient and ERP message writer with the allow retry switch on.

  Message: 0P26 INVAL SEEK (Seek Check/ Command Reject), or 0P15 SEEK CHECK.

- Sense Byte 0, Bit 1 - Intervention Required.

  Action: Exits to disk ERP transient and ERP message writer for operator intervention.

  Message: 0P08 INTERV REQ.

- Sense Byte 0, Bit 2 - Bus-Out Check.

  Action: Exits to retry the operation. If the error still persists after ten retries, exits to disk ERP transient and ERP message writer with the allow retry switch on.

  Message: 0P09 BUSOUT CHK.

- Sense Byte 0, Bit 4 - Data Check.

  Action: CCB options (all data checks, data check on read or verify). Exits to retry the operation. If the error still persists after 256 retries, exits to disk ERP transient and ERP message writer with the allow retry switch on. After each 16 retries, a recalibrate is performed until the maximum of 256 retries is reached. If error persists after 256 retries, posts data check on count to CCB, if present; otherwise, posts data check. If command code is verify (implied), posts verify error in CCB.

  The verify command is implied under the following conditions:

  The CCW has a command code X'C1' (Write - Special Count, Key and Data), and the skip and SLI flags are set on.

  The CCW has a command code X'1E' (Read - Count, Key and Data), the skip and SLI flags are set on, and the CCW follows a CCW with a write command code X'1D'.

  Message: 0P12 VERIFY CHK (Data Check on Verify command), or 0P11 DATA CHECK (Data Check/not Data Check on Count or Verify), or 0P16 DTACHK CT (Data Check on Count).

  Note: Home Address is read, and the track address is provided for the error message. For other errors, the track address is obtained from the user seek address if error occurs during channel program execution.

- Sense Byte 0, Bit 5 - Overrun.

  Action: Exits to retry the operation. If the error still persists after ten retries, exits to disk ERP transient and ERP message writer with the allow retry switch on.

  Message: 0P14 OVERRUN.

- Sense Byte 1, Bit 6 - Missing Address Markers.

  Action: Exits to retry operation. If the error still persists after ten retries, exits to disk ERP transient and ERP message writer with the allow retry switch on.

  Message: 0P13 ADDR MRKER.

  Note: Home Address is read, and the track address is provided for the error message. For other errors, the track address is obtained from the user seek address if error occurs during channel program execution.

- Sense Byte 0, Bit 0 - Command Reject.

  Action: Check for Byte 1, Bit 5 (File Protect); exits in either case to disk ERP transient and ERP message writer to issue message after which the task is canceled.

  Message: 0P18 COMM REJCT, or 0P17 FILE PROT.

- Sense Byte 0, Bit 6 - Track Condition Check.

  Action: Reads Home Address and R0 in the error recovery routine and moves CCHH from R0 to Seek command executed below.

  If alternate track: updates seek address to the next track address. If the track address equals 10, treats it as End-of-Cylinder; otherwise proceeds to next step.

  Sets up the channel program: Seek, Read Home Address (with skip bit on), TIC to CSW address minus eight. Executes this channel program in error recovery. At channel end, exits to channel scheduler CSW processing routine. If DASD file protection is present, sets the appropriate file mask following Seek.

- Sense Byte 1, Bit 1 - Track Overrun.

  Action: Posts track overrun to the CCB and exits to continue processing.

- Sense Byte 1, Bit 2 - End of Cylinder.

  Action: Posts End of Cylinder to the CCB and exits to continue processing.

- Sense Byte 1, Bit 5 - File Protect.

  Action: Exits to disk ERP transient and ERP message writer to issue message after which the task is canceled.

  Message: 0P17 FILE PROT.

- CSW Bit 47 - Chaining Check.

  Action: Ten retries. If error persists, exits to disk ERP transient and ERP message writer with the allow retry switch on.

  Message: 0P14 OVERRUN.

  Note: If the error routine gets an error while trying to execute a Restore command or Read Home Address or R0, exit is taken to the ERP message writer with the allow retry switch on. Message: 0P20 ERR ON REC (error during recovery).

3330 DISK STORAGE ERROR RECOVERY

This routine is entered at label DISKERR. The following list gives the conditions tested for.

- CSW Bit 44 - Channel Data Check.

  Action: One retry. If error persists, exits to disk ERP transient and ERP message writer with the allow retry switch on.

  Messages: 0P28 CHAN DTCHK.

- Sense Byte 1, Bit 0 - Permanent Error.

  Action: Exits to retry the operation.

- Sense Byte 0, Bit 3 - Equipment Check.

  Action: Ten retries. If error persists, exits to disk ERP transient and ERP message writer with the allow retry switch on.

  Message: 0P10 EQUIP CHK.

- Sense Byte 0, Bit 2 - Bus-Out Check.

  Action: One retry. If error persists, exits to disk ERP transient and ERP message writer with the allow retry switch on.

  Message: 0P09 BUSOUT CHK.

- Sense Byte 0, Bit 1 - Intervention Required.

  Action: One retry. If error persists, exits to disk ERP transient and ERP message writer with the allow retry switch on.

  Message: 0P08 INTERV REQ.

- Sense Byte 0, Bit 0 - Command Reject.

  Action: Exits to disk ERP transient and ERP message writer to issue message after which the task is canceled.

  Message: 0P18 COMM REJCT.

- Sense Byte 1, Bit 4 - No Record Found.

  Action: Retries nine times. If condition persists, posts no-record-found to CCB and exits to continue processing.

  Message: 0P21 NRF-MADDMK.

- Sense Byte 0, Bit 5 - Overrun.

  Action: Nine retries. If error persists, exits to disk ERP transient and ERP message writer with the allow retry switch on.

  Message: 0P14 OVERRUN.

- Sense Byte 1, Bit 1 - Invalid Track Format.

  Action: Posts track overflow to CCB and exits to continue processing.

- Sense Byte 0, Bit 4 - Data Check.

  Action: If error is not correctable (Byte 2, Bit 1 = 0), then retries nine times and if error persists, exits to disk ERP transient and ERP message writer with the allow retry switch on.

  If error is correctable and track overflow occurred (Byte 1, Bit 7), then exit is taken to retry the operation.

  If error is correctable with no track overflow, then the error correction bytes are ORed to error bytes in the data area, channel end and device end are posted in the CCB, and exit is taken to continue processing.

  If the error correction function detects incorrect length and the SLI flag bit is off, then incorrect length is also posted to the CCB before exit is taken.

  If chaining and end-of-record has been detected by error correction before the end of the user's channel program, then a Read Count CCW and TIC to user's next CCW is built, and exit is taken to restart the operation.

  If channel truncation exists, the equipment check bit in sense byte 0 is turned on and the exit is taken to retry the operation.

  Message: 0P11 DATA CHECK.

- Sense Byte 1, Bit 3 - End of Cylinder.

  Action: Posts End of Cylinder to the CCB and exits to continue processing.

- Sense Byte 1, Bit 5 - File Protect.

  Action: Exits to disk ERP transient and ERP message writer to issue message after which the task is canceled.

  Message: 0P17 FILE PROT.

- Sense Byte 1, Bit 7 - Track Overflow.

  Action: Exits to retry the operation.

- Sense Byte 2, Bit 3 - Environmental Data Present.

  Action: Decrements error count and exits to retry the operation.

- CSW Bit 47 - Chaining Check.

  Action: Ten retries. If error persists, exits to disk ERP transient and ERP message writer with the allow retry switch on.

  Message: 0P14 OVERRUN.

3340 DASD ERROR RECOVERY

- Sense Byte 0 Bit 0 - Command reject.

  Action: Permanent error - the existing supervisor permanent error routine will be used.

  Message: 0P18 COMM REJCT.

- Sense Byte 1 Bit 1 - Invalid track format.

  Action: Post the track overflow bit in the CCB and continue.

- Sense Byte 1 Bit 4 - No record found.

  Action: Post the no record found bit in the CCB and take the continue exit. If user wants to retry, retry 10 times, then take a permanent error exit.

  Message: 0P21 NRF - MADDMK (If retried).

- Sense Byte 0 Bit 1 - Intervention required.

  Action: The 2314 error routine will be used. The logging will be done if a nonzero condition exists in byte 10, bits 1-3 of the error queue entry. Return is to the supervisor which calls the message routine.

  Message: 0P08 INTERV REQ.

- Sense Byte 0 Bit 2 - Bus out parity check.

  Action: Retry one time, then consider error permanent. The 2314 error routine will be used. The error logging bit will be turned on.

  Message: 0P09 BUSOUT CHK.

- Sense Byte 2 Bit 3 - Environmental data present.

  Action: Decrement the error count by one and retry. The 3330 error routine will be used. The logging bit will be turned on.

- Sense Byte 0 Bit 3 - Equipment check.

  Action: Retry 10 times, then consider error permanent. The 2314 error routine will be used. The logging bit will be turned on.

  Message: 0P10 EQUIP CHK.

- Sense Byte 0 Bit 4 - Data check noncorrectable (byte 2 bit 1 is zero).

  Action: Retry 10 times, then consider error permanent. The 2314 error routine will be used. The logging bit will be turned on.

  Message: 0P11 DATA CHECK.

- Sense Byte 0 Bit 5 - Overrun.

  Action: Retry 10 times, then consider error permanent. The 2314 error routine will be used. The logging bit will be turned on.

  Message: 0P14 OVERRUN.

- Sense Byte 0 Bit 6 - Track condition check.

  Action: Read home address and record 0. If defective track, get the address of the alternate track from record 0, seek to the alternate track, then search and resume the operation. If alternate track, get the address of the defective track from record 0, seek to the defective track plus 1, then search and resume the operation. The 2314 error routine will be used with modifications to allow for 3340 cylinder size when checking for End-of-Cylinder.

- Sense Byte 0 Bit 7 - Seek check.

  Action: Execute a recalibrate CCW and issue a seek to the originaladdress. Retry 10 times, then consider the error permanent. The 2314 error routine will be used. The logging bit will be turned on.

  Message: 0P15 SEEK CHECK.

- Sense Byte 0 Bit 4, Byte 2 Bit 1 - Data check and correctable.

  Action: The 3330 error correction routine will be used with modifications to allow for a 2-byte error correction code for 3340 and a 3-byte error correction code for 3330. The logging bit will be turned on. The following action is taken depending on the type of error detected:

  a. If the error is correctable, OR the error correction bytes to error bytes in the data area, post channel end and device end bits in the CCB and take the continue exit.

  b. If the error correction routine detects the incorrect length and the SLI flag bits are off, then post incorrect length bit in the CCB before taking the continue exit.

  c. If the chaining and End-of-Record conditions have been detected by the error correction routine before the end of the user's channel program, then build a read count CCW, TIC to the user's next CCW, and take the restart exit.

  d. If the channel truncation condition exists, turn on the equipment check bit in the sense byte and take the retry exit.

- Sense Byte 1 Bit 2 - End-of-Cylinder.

  Action: Post the end-of-cylinder bit in the CCB and continue. The 3330 error routine will be used.

- Sense Byte 1 Bit 5 - File protected.

  Action: The permanent error routine in the supervisor will be used.

  Message: 0P17 FILE PROT.

SHARED VIRTUAL AREA (SVA)

The Shared Virtual Area (see Figure 7.1) is located in the high end of virtual storage and has a storage protect key equal to zero. It is built by job control after IPL. The SVA contains:

- a System Directory List (SDL) of phases in the SVA and other highly used phases, resident in the System Core Image Library
- highly used programs that are resident and can be shared between partitions (virtual library). These programs run with the PSW of the partition which invokes their service. Only phases that are relocatable and reenterable can be put in the SVA
- a GETVIS area for the system (which can be used as a VSAM work area).

A program is only loaded into the requesting partition if it is not in the virtual library (see Figure 7.2). However, the presence of the program in the virtual library is ignored if the main page pool contains less than eight page frames.

In general, a phase is loaded into the SVA at the next available location if its length is not greater than the still available length of the page. If it is greater, it would cross a page boundary and in this case it is, for performance reasons, loaded at the next page boundary.

Length	Contents
8	date plus time SVA has been updated (Note 1)
4	Start address of the System Directory List
4	start address of the virtual library (Note 2)
4	address of next available location in SVA
4	address of end of the SVA
10	communications area for Job Control and $MAINDIR
up to 32K	System Directory List (Note 3)
<8	alignment bytes for doubleword boundary
any	virtual library (see Figure 7.2) containing reenterable and relocatable phases
any	GETVIS area for the system (starts on page boundary)

Figure 7.1. Layout of Shared Virtual Area

Notes:

1. As one of the tests to determine whether a warm start copy of the SVA is available, Job Control compares this entry after IPL against the date and time in the SCIL descriptor entry.

2. Address of first doubleword aligned byte after SDL.

3. The layout of this area is compatible with a directory block in the Core Image Library. The SDL has fixed-length entries of 34 bytes. The last entry contains 8X'FF' as phasename.

```
        |page boundaries  |
r-------+-----------------+----------
|   A   |                 |
+-------+-----------------+----------
|doubleword aligned
```

Figure 7.2. Layout of Virtual Library

A: occupied by first phase in virtual library if the phase fits into this area. If the phase is longer than this area, it is loaded on page boundary.

DIRECTORY LIST SUPPORT

Directory List Support allows the user to create in-storage directories of highly used phases. LOADs and FETCHs of such selected phases are made without searching the Core Image Library directories on disk. A System Directory List, available to all partitions, is provided in the SVA for phases resident in the SVA, and other high usage phases. This is created by job control after IPL. Local Directory Lists may be created at any time, and are local to the partition which creates them. A local directory list exists for the duration of the jobstep, in which it is created.

Note: The local directory list used by RPS is located in the SVA.

It should be noted that an in-storage directory entry in the user's partition does not contain any valid information, except for the phasename, until the first FETCH or LOAD request for the phase specifying this entry has been honoured. The first FETCH or LOAD request for the phase fills in the entry and subsequent requests can use this entry.

Parameter List

On entry to the FETCH routine, register 1 points to a parameter list or to a phase name. If register 1 points to a parameter list, the parameters are saved in the save area CALLSEQ (see Figure 7.3). This area is also used when no parameter list is passed. In this case bytes 4-7 contain zeros.

```
  0          3  4  5       7
┌────────────┬───┬─────────┐
│  CALLPNAM  │   │CALLLNAM │
└────────────┴───┴─────────┘
              LISTFLAG
```

Dis-place-ment	Length	Contents
0	4	Phasename pointer (first byte must be zero)
4	1	Option Flags (X'00' when no parameter list is provided) X'80' : not used X'40' : not used X'20' : not used X'10' : SVA load request. (See note 1) X'08' : inactive entry X'04' : SCIL to be scanned before PCIL X'02' : directory entry provided (DE=YES) X'01' : bypass load (phase in SVA or TXT=NO)
5	3	Pointer to local directory list, or zero if no local directory list is provided

Note 1: set and used by $MAINDIR for loading phases into the SVA

Figure 7.3. Fetch Parameter Save Area (CALLSEQ)

SEARCH FOR DIRECTORY ENTRY

The user may have provided in his own partition a directory entry for the requested phase. He may have done this in two ways:

1. He may have provided a directory element to which the phasename parameter (see Figure 7.3) is pointing (DE=YES specified on the FETCH or LOAD macro).

2. A local directory list is provided, which may contain a directory entry for the requested phase (LIST parameter specified on FETCH or LOAD macro).

If the user has provided such a directory entry, it can only be used when it has been filled in (activated) as result of an earlier request for the phase specifying the same directory element or list. If the directory entry is found to be active, the search is ended and exit is taken to the routines that use the contents of the entry, usually to load the phase.

If the entry is not yet active (not filled in), its presence is noted and the search continues. When an active entry is found later (or is read from disk), an earlier found inactive entry is activated.

When no active entry is found during the search described above, the System Core Image Library (SCIL) and Private Core Image Library (PCIL, if supported and assigned) directories are searched. The System Directory List (SDL) in the SVA is scanned immediately before the main SCIL. The search takes place in one of two possible sequences:

- If a $-phase is requested, or if SYS=YES on FETCH or LOAD macro:

  1. SCIL main directory (preceded by scan of SDL)

  2. PCIL main directory, if PCIL is assigned

  3. PCIL link directory, if PCIL assigned and link directory present

  4. SCIL link directory, if it is present and if PCIL not assigned

- If not a $-phase requested and SYS=NO on FETCH or LOAD macro:

  1. PCIL link directory, if PCIL assigned and link directory present

  2. SCIL link directory, if it is present and if PCIL not assigned

  3. PCIL main directory, if PCIL is assigned

  4. SCIL main directory (preceded by scan of SDL).

An active entry with a condense counter value which does not match the corresponding value in the Fetch Table is treated as if it were not active. SDL entries and disk directory entries are exempted from this check.

The Fetch Table

The Fetch Table (see Figure 7.4) contains an entry for the System Core Image Library and an additional entry for each partition if the supervisor supports Private Core Image Libraries. Each entry contains the address of the corresponding Second Level Directory (see below), a condense counter, the disk addresses of the directory and the link area of the library, and the number of tracks per cylinder and the number of library blocks per track of the library. If PCIL is supported and a PCIL has been assigned for the partition issuing the Fetch request, the Fetch Routine sets up a pointer to the appropriate Fetch Table entry. This pointer is zero, when no PCIL is assigned.

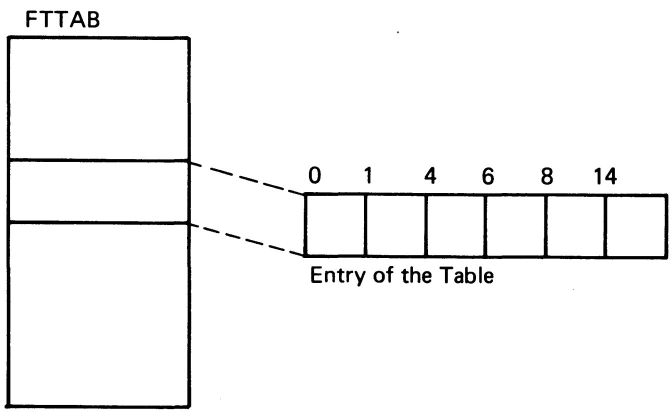


Figure 7.4. Fetch Table (FTTAB)

Dis-place-ment	Length	Contents
0	1	condense counter (incremented for each condense of the library. Reset to zero when a new library is assigned - PCIL only)
1	3	address of the corresponding Second Level Directory (zero if no SLD present)
4	2	number of tracks per cylinder
6	2	number of library blocks per track (device indicator if RPS device)
8	6	address of directory of cataloged phases in the form BBCCHH
14	6	address of linkarea in the form BBCCHH (zero is no linkarea present)

Notes: The address of the Fetch Table is in SYSCOM. The Fetch Table is used as an interface between Job Control, Linkage Editor, Librarian, and Fetch.

The entries for the partitions will not exist if the support for Private Core Image Libraries is not included in the Supervisor.

The Second Level Directory

The address of the Second Level Directory is found in the appropriate entry in the FETCH table. It is a table of the highest phase names found on each track of the corresponding directory. A Private Second Level Directory is optional for PCILs.

The Fetch routine first scans the Second Level Directory (if present) of the appropriate directory and determines on which track the directory search should start. Then the directory itself will be searched by means of a search on key for the directory block containing the required directory entry.

The last active entry in any SLD is always 8X'FF'. If the SLD has as many or more entries than there are active tracks in the corresponding directory, the 8X'FF' entry corresponds with the directory track. Remaining SLD entries are not used. If the SLD has less entries than there are active tracks, the last SLD entry is X'FF' and the remaining directory tracks are not represented. The multitrack channel search for directory blocks on unrepresented tracks begins on the last represented track, corresponding to the 8X'FF' SLD entry.

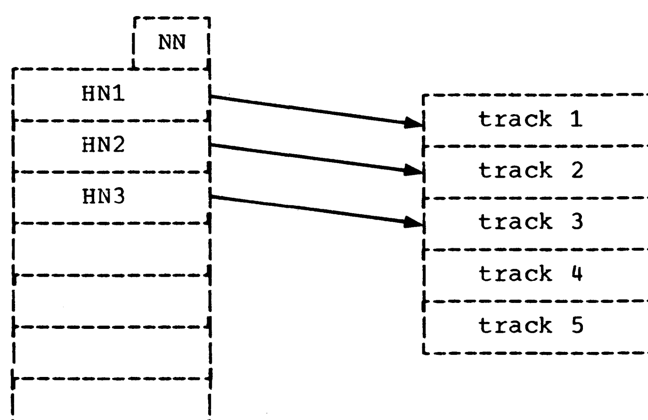


Figure 7.5. Relationship Between Directory and its Second Level Directory

NO	Component		Address TT	Address R	NO TR	R = Required O = Optional
1	IPL Bootstrap Record 1 ($$A$IPL1)		00	1	1	R
	IPL Bootstrap Record 2 ($$A$IPL2)		00	2		R
	Volume Label		00	3		R
	User Volume Label		00	4		O
2	System Directory	Record 1	01	1	1	R
		Record 2	01	2		R
		Record 3	01	3		R
		Record 4	01	4		R
	IPL Retrieval Program ($$A$IPL2)		01	5		R
3	Core Image Library Directory	Directory of Cataloged Phases	02	1	X	R
		Directory of Linked Phases				
4	Core Image Library		YY	1	X	R
5	Relocatable Directory		YY	1	X	O
6	Relocatable Library		YY	1	X	O
7	Source Statement Library Directory		YY	1	X	O
8	Source Statement Library		YY	1	X	O
9	Procedure Library Directory		YY	1	X	O
10	Procedure Library		YY	1	X	O
11	Label Cylinder		YY	1	X	R

X: depends on user allocation Y: depends on user allocation

Figure 7.6. Layout of SYSRES

0	8	11	12	14	16	17	18
name	TTR	N	TT	LL	C	T	PPP

21	24	26	27	30	31
EEE	RR	R	AAA	K	VEE

Figure 7.7. Layout of Directory Entry

Field	Length	Contents
name	8	phasename
TTR	3	relative track address and record number of phase
N	1	number of half-words containing user data
TT	2	number of text blocks
LL	2	number of text bytes in last text block
C	1	switch indicating type of phase: X'80' = selfrelocating phase X'40' = relocatable phase X'20' = SVA eligible X'10' = phase is in the SVA (Note 1) X'08' = phase found in a PCIL (Note 1) X'04' = phase not found (Note 1) X'02' = active entry (filled in) X'01' = not used

T	1	always X'00' (used as type byte for stow table)
PPP	3	load point at LNKEDT time (Note 3)
EEE	3	entry point at LNKEDT time (Note 3)
RR	2	number of RLD items (Note 4)
R	1	number of additional RLD blocks (Note 4)
AAA	3	partition start address at LNKEDT time (Note 4)
K	1	condense counter at the time when entry was activated (Note 7)
VEE	3	entry point of phase in SVA (Notes 1 and 5)

Notes:

1. Only used for directory entries that are in storage.

2. The TTR is relative to the beginning of the directory.

3. PPP and EEE are not present if both are zero and the phase is not relocatable.

4. RR, R, and AAA are only present if the phase is relocatable.

5. VEE is only present if the phase is SVA eligible.

6. The last entry in the directory is a 12-byte entry with a dummy phasename (contains 8X'FF'), a dummy TTR (contains XL3'00') and a dummy N (contains X'00'). Directory entries in storage always have the standard length of 34 bytes (including the last entry).

7. Condense counter is used for in-core directory entries (not SDL).

RELOCATING LOAD

If the FOPT macro included the parameter RELLDR=YES, the generated supervisor updates the entry-point address and the address constants in a relocatable phase that is fetched or loaded, and updates the load address of a relocatable phase which is returned as a result of SVC 23. It also updates the load address of a relocatable phase in the header which is returned as a result of SVC 51.

Load Address Specified in the LOAD Macro

When the second parameter of the LOAD macro is an explicit address, SVC 4 branches to the FETCH routine to read in the phase at the specified address if the phase is not in the SVA. If the phase is relocatable, the entry-point address and the address constants in the phase are updated with the relocation factor, which is the specified load address decremented by the value of the load address stored in the phase entry of the Core Image Directory. The entry-point address, or updated entry-point address, is returned to the user in general register 1. If the phase is in the SVA, it is not loaded at the specified address. The entry-point address in the SVA is returned in general register 1.

Load Address not specified

When the second parameter of a LOAD macro is an implicit address, or a FETCH macro was issued, SVC 4 or SVC 1 branches to the FETCH routine to read the phase into main storage.

If the phase is non-relocatable, it is read into storage at the address stored in the phase entry of the core image directory. The entry-point address stored in the phase entry is returned in general register 1.

If the phase is relocatable, it is read into storage in the caller's partition at an address equal to the address of the first byte beyond the register save area plus the value of the load address in the core image directory (see Figure 7.1) decremented by the value of the partition address stored in the control block of the Core Image Library Block (see Figure 7.2). The entry-point address and the address constants are updated with the relocation factor, which is the difference between the actual address at which the phase is loaded and the load address stored in the phase entry of the Core Image Directory. The updated entry-point address is returned in general register 1. If the phase is in the SVA, it is not loaded in the caller's partition. The entry-point address in the SVA is returned in general register 1.

Request for Phase Load Address by PSW Protection Key Zero Programs

When a SVC 23 is issued, general register 1 contains the address of the phase name.

If the phase is non-relocatable, the load address in the directory entry for the phase is returned to the user at the address specified by general register 0.

If the phase is relocatable, the load address returned at the address specified by register 0 is equal to the address of the first byte beyond the register save area plus the value of the load address in the phase entry of the core image directory decremented by the value of the partition address in the control block of the Core Image Library Block.

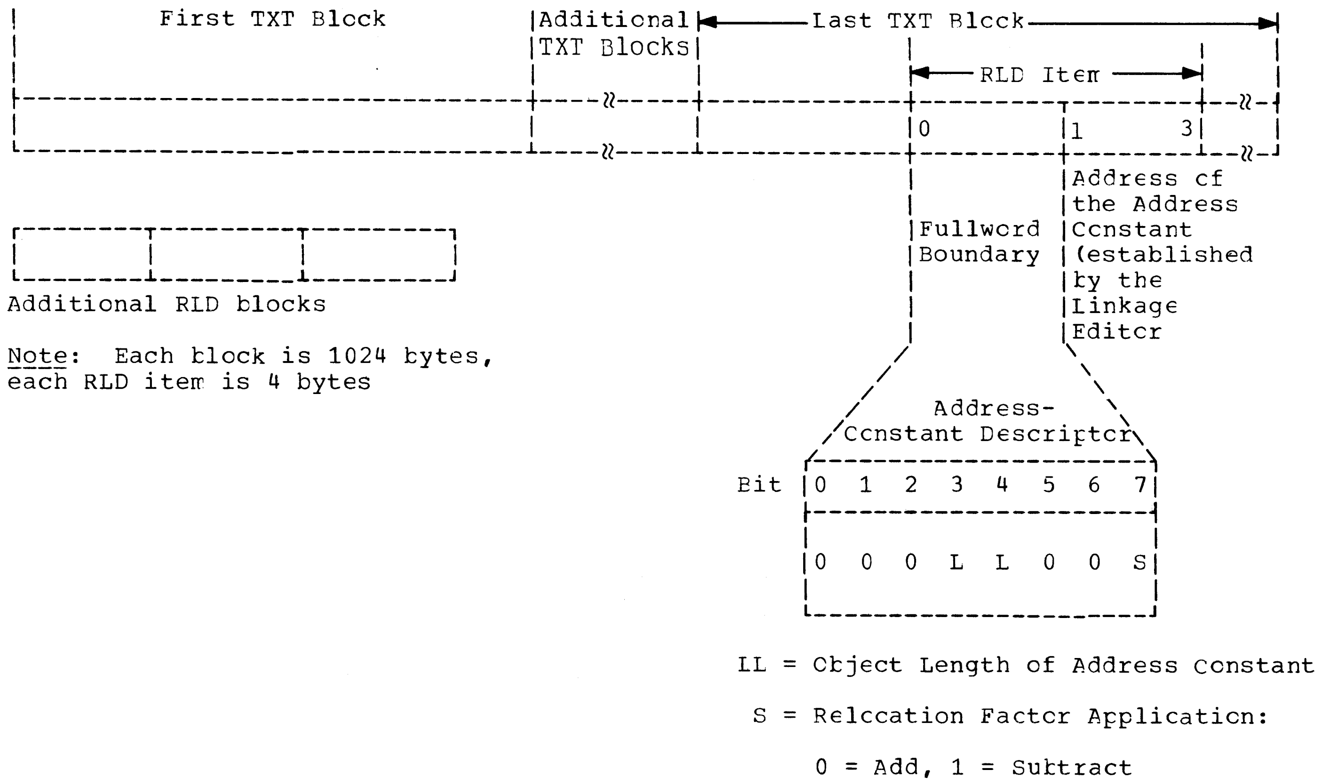


Figure 7.8. Core Image Library Block for Relocatable Phase

The program issuing an SVC 23 is canceled if the supervisor was generated with NPARTS=1, or if the PSW protection key is not equal to zero (only job control and B-transient programs can issue an SVC 23).

Request for Phase Header

When a SVC 51 is issued, general register 1 contains the address of the phase name. This phase name is followed by a save area to which a number of bytes of the phase header are moved.

If the last TXT block is not completely filled with TXT, the remaining bytes contain RLD information starting at the next fullword boundary (intermediate bytes are padded).

For information about the core image library, refer to the IBM publication DOS/VS Librarian listed in the Preface to this manual.

IDRA (Independent Directory Read-in Area)

IDRA is a supervisor option that generates a supplemental area to reduce contention for the physical transient area. Core Image Library directories are read into the IDRA for all fetch and load operations. Thus, the PTA is left free to perform ERP. The IDRA is large enough to contain at least two text blocks.

The physical attention routine consists of two transients, $$ABERRZ and $$ABERZ1. The following four conditions result in activation of the attention task:

- The operator presses the console request key. This initiates an attention I/O interrupt from the console. When this interrupt comes in, the attention task is activated and the attention PIB switch byte (see Figure 2.7, item B) is posted for fetch of $$ABERRZ (REQPTA) and consecutive fetch of the logical attention root phase $$BATTNA (LTAFCH). When the operator wishes to cancel a partition quickly, he presses the request key twice in quick succession. The second attention interrupt is detected when the attention task is already active as a result of the first interrupt. Emergency cancel (REQCAN) is posted in the attention PIB switch byte.

- The operator presses the external interrupt key for communication with the background partition. This causes an external interrupt. The attention task is activated when the interrupt comes in. The attention task PIB switch byte is posted to indicate external interrupt request (EXTMSK). This causes $$BATTNA to be fetched when the attention task is selected.

- A task is canceled in the logical transient area, in which it may still have I/O pending. This I/O is to be quiesced before the task is finally canceled. The transient $$ABERZ1 performs this function. The attention task is activated and the attention task PIB switch byte is posted for fetch of $$ABERZ1 (DELCAN). This transient searches the channel queue for I/O requests from the LTA and any request found is to be quiesced.

- The operator presses the ENTER key on the Display Operator Console (DOC) after entering a command or data. The CRT system task is activated. If a screen management command was entered, the appropriate CRT transient is called to execute the command. If an attention command was entered, the CRT task makes the attention task selectable. The attention task PIB switch byte is posted to indicate command available (REQLTA). This causes $$BATTNA to be fetched when the attention task is selected.

When the attention task is selected by task selection, the attention PIB switch byte is interrogated.

1. When the fetch $$ABERRZ indicator (REQPTA) is on, $$ABERRZ is fetched. $$ABERRZ resets this indicator and determines whether a emergency cancel condition (REQCAN) exists or a normal request for logical attention is being serviced.

   a. <u>Request for logical attention</u>: $$ABERRZ starts scanning the PUBs to see whether intervention is required. If intervention is anywhere required, a message is printed for the operator to reply with cancel or ignore. If the operator replies with cancel, the attention task is deactivated (unless DELCAN, fetch $$ABERZ1, is posted in the attention PIB switch byte) and a branch through AREXIT to the supervisor leads to the resident cancel routine, to cancel the task that is attempting to use the device with the intervention required condition. However, if the failing I/O operation is already dequeued by the time the operator replied with cancel, or if POWER/VS is running in the partition to be canceled, a cancel ignored message is issued, and the scan of the PUBs continues. If no intervention is required, or the operator replied with ignore, or the cancel reply is ignored, control is returned to the supervisor (the attention task is not deactivated) to fetch the logical attention root phase $$BATTNA.

   b. <u>Emergency cancel condition</u>: The message 'request cancel' is issued. According to the operator's reply, cancelation of the appropriate partition is initiated. The attention task is deactivated (unless DELCAN, fetch $$ABERZ1, is posted in the attention PIB switch byte) and a branch through AREXIT to the supervisor leads to the resident cancel routine. The emergency cancel request is ignored, however, if the operator replies with END/ENTER, or if POWER/VS is running in the partition. The attention task is deactivated, unless DELCAN (fetch $$ABERZ1), EXTMSK (external interrupt), or REQLTA (DOC command available) are posted in the attention PIB switch byte.

2. When the fetch $$ABERZ1 indicator (DELCAN) is on, while the fetch $$ABERRZ indicator (REQPTA) is not on, $$ABERZ1 is fetched. A task was canceled in the LTA and its I/O is to be quiesced. $$ABERZ1 scans all PUBs for a PUB which has an I/O request queued to it from the LTA. If such an I/O request is found, SVC 3 is issued to attempt to dequeue the request. Control is then returned to the supervisor. As DELCAN is not reset, a fetch loop will result. When this phase is fetched again, the PUB scan is continued with the PUB on which the request from the LTA was found. If no request from the LTA is found (on first or subsequent testing by $$ABERZ1), the canceling task is posted ready to run and LTA I/O complete is indicated in bit 0 of the cancel code in its PIB (the cancel procedure can continue). DELCAN is reset as subsequent fetches of this phase are no longer required.

3. If neither $$ABERRZ, nor $$ABERZ1 are to be fetched, the logical attention root phase $$BATTNA is fetched. This is the case when the attention task was activated because the operator pressed the external interrupt key, or he entered an attention command on DOC and pressed the ENTER key, or he pressed the request key once and $$ABERRZ released control (having reset the fetch $$ABERRZ switch, REQPTA, in the attention PIB switch byte). $$BATTNA determines which of these conditions exists (see DOS/VS Logical Transients Logic).

Message Code	Message Text
0P60D	INTERV REQ prog.id. cuu
0P89I	I/O CANCEL IGNORED
1I32K	AREA NOT ACTIVE
1I40D	REQUEST CANCEL
1S01D	INVALID STATEMENT
1S02D	INVALID STATEMENT
1Q62I	CANCEL FOR POWER/VS IGNORED

Figure 8.1. Physical Attention Messages ($$ABERRZ)

MACHINE CHECK ANALYSIS AND RECORDING

MCAR responds to MCIs, attempts recovery, and provides operator messages on SYSLOG. Machine check records are recorded on the recorder file IJSYSRC by the RMSR transients $$RAST01 and $$RAST08.

When a machine check occurs, hardware first logs the error in the Machine Check Logout Area in low real storage and then retries the failure by CPU retry and ECC (Error Checking and Correction). If the retry is successful, a soft machine check occurs on the Models 135, 145, 155-II and 158 (if enabled). The Model 115 or 125 does not generate an interrupt for a machine check from which it could recover. For soft MCIs the recording is controlled through the Error Frequency Limit (EFL) feature. If a specified error count is reached, the recording mode is changed from recording to quiet. The MODE command gives the operator control of soft MCIs. It permits the operator three options:

- Determine whether the system is in quiet or recording mode.
- Alter mode of operation.
- Change error threshold values.

If hardware retry is not successful, MCAR determines machine check severity. A hard MCI occurs when:

- CPU retry is not successful.
- Interrupted instruction cannot be retried.
- Storage failure is permanent.

In the event of a hard MCI, the affected task is canceled. MCAR assesses the damage and continues system operation when possible. The system enters the hard wait state when a hard MCI:

- Interrupts supervisor coding.
- Occurs while assessing critical information or phases from SYSRES.
- Damages privileged coding through a permanent storage error.

MCAR attempts to notify the operator about:

- Machine check type.
- Wait state, re-IPL.
- Problem program termination.
- Mode operation change.
- Buffer deletion.

The Resident Machine Check Handler analyzes the Machine Check Interruption Code (MCIC) and the Problem State bit (bit 13) of the Machine Check Old PSW. It categorizes errors into three classes:

a. System operation termination condition. The MCIC indicates:

   - System damage.
   - CPU-mask (1MWP) in old PSW is invalid.
   - Instruction processing damage (while the CPU is in the supervisor state).
   - One or more old PSW bits, other than in the CPU-mask, are invalid (while the CPU is in the supervisor state).
   - Storage or Protection error while the failing storage address is invalid.

   Action: Post C'A' in location 0 (system termination code), post C'S' in location 1 (for SEREP), and post Emergency Exit (EMEXIT,X'08') in ID00SLCT (indicator to $$RAST00 to terminate the system).

b. Hard Machine Checks. The system can continue but the damaged task is to be canceled. While the CPU is in the problem state, the MCIC indicates:

   - Instruction processing damage.
   - One or more old PSW bits, other than in the CPU-mask, are invalid.

   Action: Activate RAS System task and branch to the cancel routine to cancel the task. If storage error on GETREAL request, exit to task selection (do not cancel task).

c. Soft Machine Check (if none of the above conditions is present). Only recording is required for errors from which hardware recovered successfully.

   Action: If interruption occurred while in problem state, activate RAS system task and exit to Task Selection.

   If supervisor function is being performed on system task active, activate RAS system task (if not already active) and return to the interrupted code by loading the machine check old PSW.

Nonresident machine check handling is described in the DOS/VS Error Recovery and Recording Transients Logic manual.

CHANNEL CHECK HANDLER

The channel check handler responds to channel error conditions. It first records the error. Then, it evaluates the damage and attempts to reduce the impact of the error on the system as a whole.

CCH assesses channel control checks and interface control checks to determine if the system can continue. Figure 9.1 shows the results of the Channel Check Severity Routine. Resident CCH performs severity analysis and builds an ERPIB entry. See Figure 9.2.

Resident CCH coding performs no recovery or recording action, but fetches the proper R-transient to attempt restoration of system operation and to record the error on the recorder file. Transients needed by CCH operate under control of the RAS Monitor.

System termination results when:

- Hard channel error occurs during accessing of critical information or phases from SYSRES.
- Hard channel error occurs on the paging channel.
- System reset has occurred.
- Channel address or reset codes are invalid.

A channel error is detected when a CSW is stored with either the interface control check or channel control check bit on. Thus, when an I/O interrupt with channel check occurs or when channel check occurs on an I/O instruction, the logging is done under control of CR14 bit 2.

This routine gains control from either the SIO or I/O interrupt routines when a channel check occurs. The ECSW is inspected to determine if enough information is valid to isolate the damage to either a channel or a device. (See Figure 9.1.)

When a system termination condition exists, the EMEXIT bit is posted in location ID00SLOT to indicate to $$RAST00 that the system is to be terminated. The applicable termination code is posted at location 0. The following list gives the termination codes for the various types of disastrous channel errors:

B Irrecoverable channel check on fetch.

C Channel check on log with RASMSG.

D ECSW not stored.

E ERPIB queue has been exhausted.

F Two channels damaged on RTA I/O active.

G System reset code in ECSWW.

H Retry reset codes invalid.

I Channel address invalid.

J Irrecoverable channel check on paging channel.

If the damage cannot be isolated to a device, the entire channel is considered to be damaged. An ERPIB is created with the PUB field containing the address of the damaged channel. The CSW and ECSW are saved in the ERPIB for the nonresident channel check handler. If the damage can be isolated to a device, the entire ERPIB is filled for the nonresident channel check handler.

Nonresident channel check handling is described in the DOS/VS Error Recovery and Recording Transients Logic manual.

Channel Address Valid	Reset Codes Valid	System Reset Code On	Start I/O Time	Unit Address Valid	RTA I/O Active	SYSRES Channel	Action Taken
No							1, 2
Yes	No						1, 2
Yes	Yes	Yes					1, 3
Yes	Yes	No	Yes				5
Yes	Yes	No	No	No	No	No	1, 3
Yes	Yes	No	No	No	Yes		1, 2
Yes	Yes	No	No	No	No	Yes	1, 3, 4, 5
Yes	Yes	No	No	Yes			5

Action Codes

1. Schedule recording.
2. Schedule system termination with proper message.
3. Set the damaged channel byte for nonresident channel check handler.
4. Assume the error is on SYSRES device.
5. Error can be isolated to a device, use Part 2.

Note: Every Channel Check that occurs in the system is passed through the Channel Check Severity Detect Routine. You can determine the disposition of the check by using Part 1. When a Channel Check has been isolated to a device, use Part 2 to determine the action taken.

Channel Check Device Isolation Results (Part 2):

Active ERPIB Exist	Channel Check Entry	DASD	Channel Retry Request	Action Taken
No	No			6
No	Yes			7
Yes	No	No	No	8
Yes	No	No	Yes	6, 9, 10
Yes	No	Yes		6
Yes	Yes	No	No	8
Yes	Yes	No	Yes	8, 9, 10
Yes	Yes	Yes		11

Action Codes

6. Exit to supervisor I/O routine after other functions performed.
7. Find a free ERPIB in queue and fill with information for ERPs.
8. Exit to task selection after other functions performed.
9. Post ERPIB complete.
10. Dequeue the RAS CCB and requeue the user's CCB.
11. Exit to DASD channel check handler.

Figure 9.1. Channel Check Severity Detect Routine

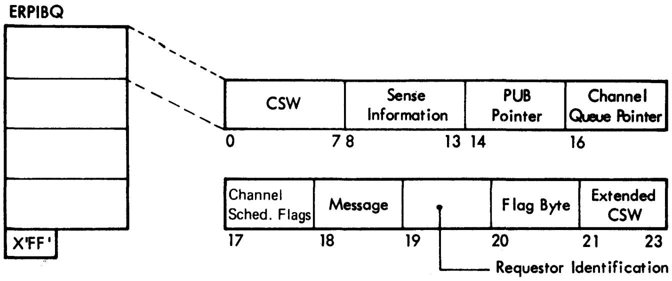


Figure 9.2. Error Recovery Procedure Information Block (ERPIB)

Note: A free entry is indicated by X'FF' in byte zero. Label ERPIBQ identifies the first byte in the Queue.

Byte 20: Flag Byte:

Bit 0: 1 = Channel Check on SIO
1-4: Not used
5: 1 = Sense data stored
6: 1 = DASD ERPIB active
7: Not used

RECOVERY TRANSIENTS AND RAS MONITOR

The Recovery Transients (R-transients) perform machine check and channel check recovery and recording. When required, they are called by the resident Channel Check Handler via the RAS Monitor.

The RAS Monitor is a resident control program which controls the RTA. At system generation the RAS Monitor forms a portion of the resident supervisor. The RAS Monitor and MCAR/CCH coding are always generated for the Models 135, 145, 155-II and 158. For Model 115 or 125, the RAS Monitor and MCAR/CCH coding are generated only if RMS=YES or MCH=YES was specified at system generation time. The RAS Monitor:

- Fetches R-transients into the RTA.
- Schedules I/O requests from the RTA.
- Accepts RTA I/O request complete postings from CCH.
- Provides an exit interface from R-transients.

DISPLACEMENT
DEC REC

<table>
<tr><td>0(0)
LD00SLOT
($$RAST00)
(1) (2)</td><td>4(4)
LD01SLOT
($$RAST01)
(3)</td><td>8(8)
LD02SLOT
($$RAST02)
(5)</td><td>12(C)
LD03SLOT
($$RAST03)
(5)</td><td>16(10)
LD04SLOT
($$RAST04)
(5)</td><td>20(14)
LD05SLOT
($$RAST05)
(5)</td><td>24(18)
LD06SLOT
($$RAST06)
(5)</td><td>28(1C)
LD07SLOT
($$RAST07)
(5)</td><td>32(20)
LD08SLOT
($$RAST08)
(5)</td></tr>
<tr><td>36(24)
LD09SLOT
($$RAST09)
(5)</td><td>40(28)
LD10SLOT
($$RAST10)
(4)</td><td>44(2C)
LD11SLOT
($$RAST11)
(5)</td><td>48(30)
LD12SLOT
($$RAST12)
(5)</td><td>52(34)
LD13SLOT
($$RAST13)
(5)</td><td>56(38)
LD14SLOT
($$RAST14)
(5)</td><td>60(3C)
LD15SLOT
($$RAST15)
(5)</td><td>64(40)
RASCCB
Residual Count</td><td>66(42)
RASTIB
Transmission information</td></tr>
<tr><td>68(44)
CCW Status bytes</td><td>70(46)
SYSRES LUB</td><td>72(48)
RASCCBF
RAS CCB indicator</td><td>73(49)
RAS Fetch CCWs address</td><td>76(4C)
CCW stored address</td><td>80(50)
RASCCWS
RAS seek CCW</td><td>88(58)
RASRCG
RAS search CCW</td><td>96(60)
RASTIC
RAS TIC CCW</td><td>104(68)
RASREAD
CCW to read module into RTA</td></tr>
<tr><td>112(70)
RASEEK
Seek Address</td><td>119(77)
RTAOWN
Index into load list for RTA owner</td><td>120(78)
MCPIK
PIK of task interrupted by machine check</td><td>122(7A)
MCTIK
TIK of task interrupted by machine check</td><td>124(7C)
ERPIBA
ERPIB queue address</td><td>128(80)
CCENTADR
Address of Channel Check address</td><td>132(84)
RTAID
RTA I/O requestor ID Routine</td><td>133(85)
ERPID
WTOR request return load index</td><td>134(86)
RASRES
SYSRES I/O address</td></tr>
<tr><td rowspan="2">136(88)
RASREC
SYSREC I/O address</td><td rowspan="2">138(8A)
RASLOG
SYSLOG device address</td><td colspan="7">140(8C) TRANSAV - RTA Register Save Area</td></tr>
<tr><td>140(8C)
TRANSAV0
Register 0</td><td>144(90)
TRANSAV1
Register 1</td><td colspan="4">148(94)
TRANSAV2 - TRANSAVE
← — — — — — →
Registers 2 - 14</td><td>200(C8)
TRANSAVF
Register 15</td></tr>
<tr><td colspan="7">204(CC) SYSREGS - System Register Save Area</td><td rowspan="2">268(10C)
LINKFLAG
(8)</td><td rowspan="2">269(10D)
SUPLINK
Address of RAS system function scan routine</td></tr>
<tr><td>204(CC)
SYSREG0
Register 0</td><td>208(D0)
SYSREG1
Register 1</td><td colspan="4">212(D4)
SYSREG2 - SYSREGE
← — — — — — →
Registers 2 - 14</td><td>264(108)
SYSREGF
Register 15</td></tr>
<tr><td colspan="4">182(B6) (HIR - Hardware Instruction Retry accumulators)</td><td colspan="4">284(11C) (ECCMAIN-Main storage error accumulators)</td><td rowspan="2">296(128)
RESTARTA
Disk restart address</td></tr>
<tr><td>272(110)
HIRACNT
HIR accumulated count</td><td>274(112)
HIRLCNT
Count threshold value</td><td>276(114)
HIR1TIME
Time of day for first error of group</td><td>280(118)
HIRLTIME
Time threshold in clock units</td><td>284(11C)
ECMACNT
Accumulated ECC count for main storage</td><td>286(11E)
ECMLCNT
Count threshold value</td><td>288(120)
ECM1TME
Time of day for first error of count</td><td>292(124)
ECMLTME
Time threshold in clock units</td></tr>
<tr><td>300(12C)
RESTARTP
PUB address of unit to be restarted</td><td>302(12E)
MCMODE
Mode status for machine checks (9)</td><td>303(12F)
BUFDEL
Count of buffers deleted</td><td>304(130)
RASMSG1
Message byte 1
(6)</td><td>305(131)
RASMSG2
Message byte 2
(7)</td><td></td><td></td><td></td><td></td></tr>
</table>

Figure 9.3. RAS Monitor Table (RASTAB) (Part 1 of 2)

Notes:

1 Areas labeled LDxxSLOT (bytes 0-63) are called the Load List and each of the 16 entries are formatted as follows:

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

2 LD00SLOT flag byte:

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

3 LD01SLOT flag byte:

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

4 LD10SLOT flag byte:

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

5 LDxxSLOT flag byte:

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

6 RASMSG1:

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

7 RASMSG2:

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

8 LINKFLAG: Indicates which function is requested by R-transient. This byte is tested by the RAS system function scan routine.

Bit	Flag	Description
0	X'80'	Normal I/O request.
1	X'40'	Emergency I/O request.
0,1	X'C0'	Restart I/O request.
2	X'20'	Fetch request.
3	X'10'	Request for exit.
0,3	X'90'	Wait request.
4	X'08'	Request to initialize registers for I/O.
5	X'04'	Dequeue CCB request.
6	X'02'	Request to cancel task.
7	X'01'	Getime request.

9 MCMODE flag byte:

Bit	Flag	Description
0	X'80'	HIR recording mode.
1	X'40'	HIR quiet mode.
2	X'20'	ECC recording mode in processor storage.
3	X'10'	ECC quiet mode in processor storage.
4	X'08'	ECC recording mode in control storage.
5	X'04'	ECC quiet mode in control storage.
6	X'02'	ECC threshold mode in control storage.
7	X'01'	Reserved.

Bytes 12-13 (X'0C' - X'0D') of the RAS Linkage Area (RASLINK) contain the address of the RAS Monitor Table. Label RASTAB identifies the first byte of the area.

Figure 9.3. RAS Monitor Table (RASTAB) (Part 2 of 2)

RASLINK

Hex / Dec displacement	0 / 0	8 / 8	9 / 9	0A / 10	0B / 11	0C / 12	10 / 16	14 / 20	18 / 24	1C / 28
Field	CPU ID Field	Damaged Channel Byte	RAS Flag Byte	Machine Check Flags	Largest CPU Model	RAS Table (RASTAB) Address	Base Address for RAS Monitor	Address of Headqueue PUB Table	Address of CCW Savearea	Address of Page Queue
	xxxxxxxx	x	x	x	x	xxxx	xxxx	xxxx		xxxx

Key to RAS Linkage Area displacement

0 CPU ID Field

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

9 RAS Flag Byte

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

10 Machine Check Flags

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

11 Largest CPU Model

12 Address of RAS Monitor Table (RASTAB)

16 Address used for base register in RAS Monitor Program

20 Address of Headqueue PUB Table

24 Address of CCW savearea

28 Address of Page Queue Table

Bytes 112-115 (X'70'-X'73') of the System Communication Region (SYSCOM) contain the address of the area.
Label RASLINK identifies the first byte of the area.

Figure 9.4. RAS Linkage Area (RASLINK)

Chart 01.1. Supervisor General Entry

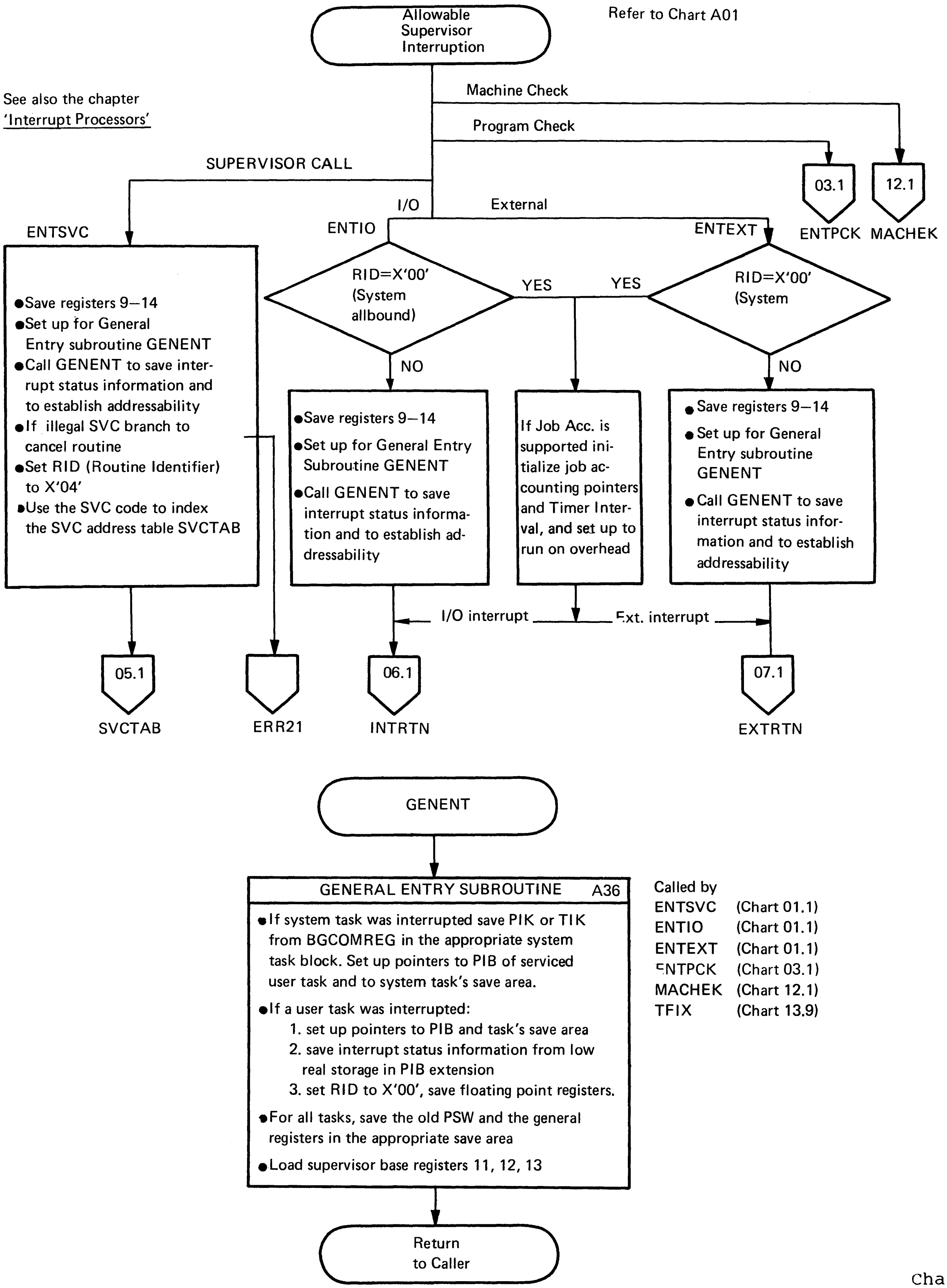
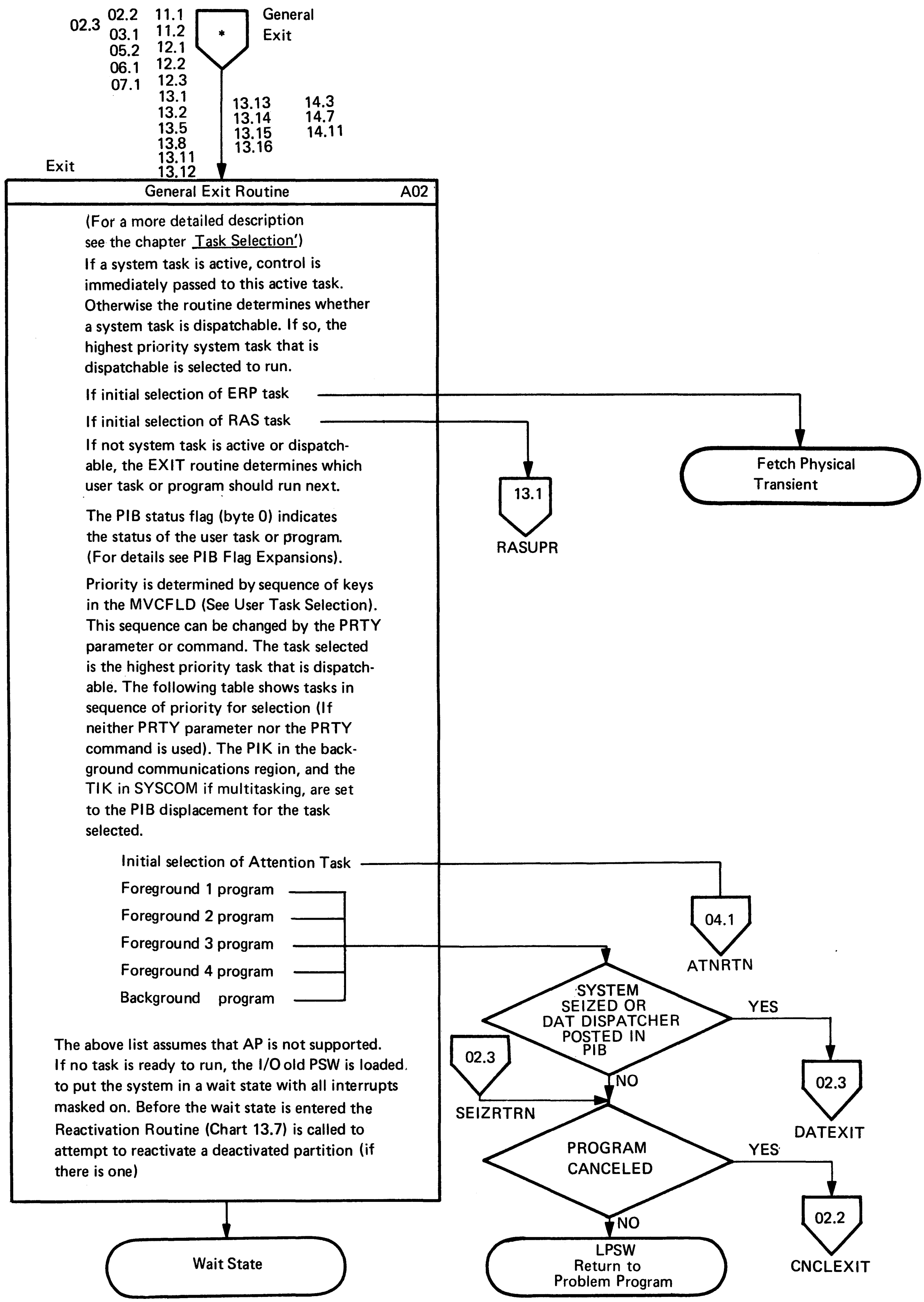


Chart 02.1. Supervisor General Exit, Select Task



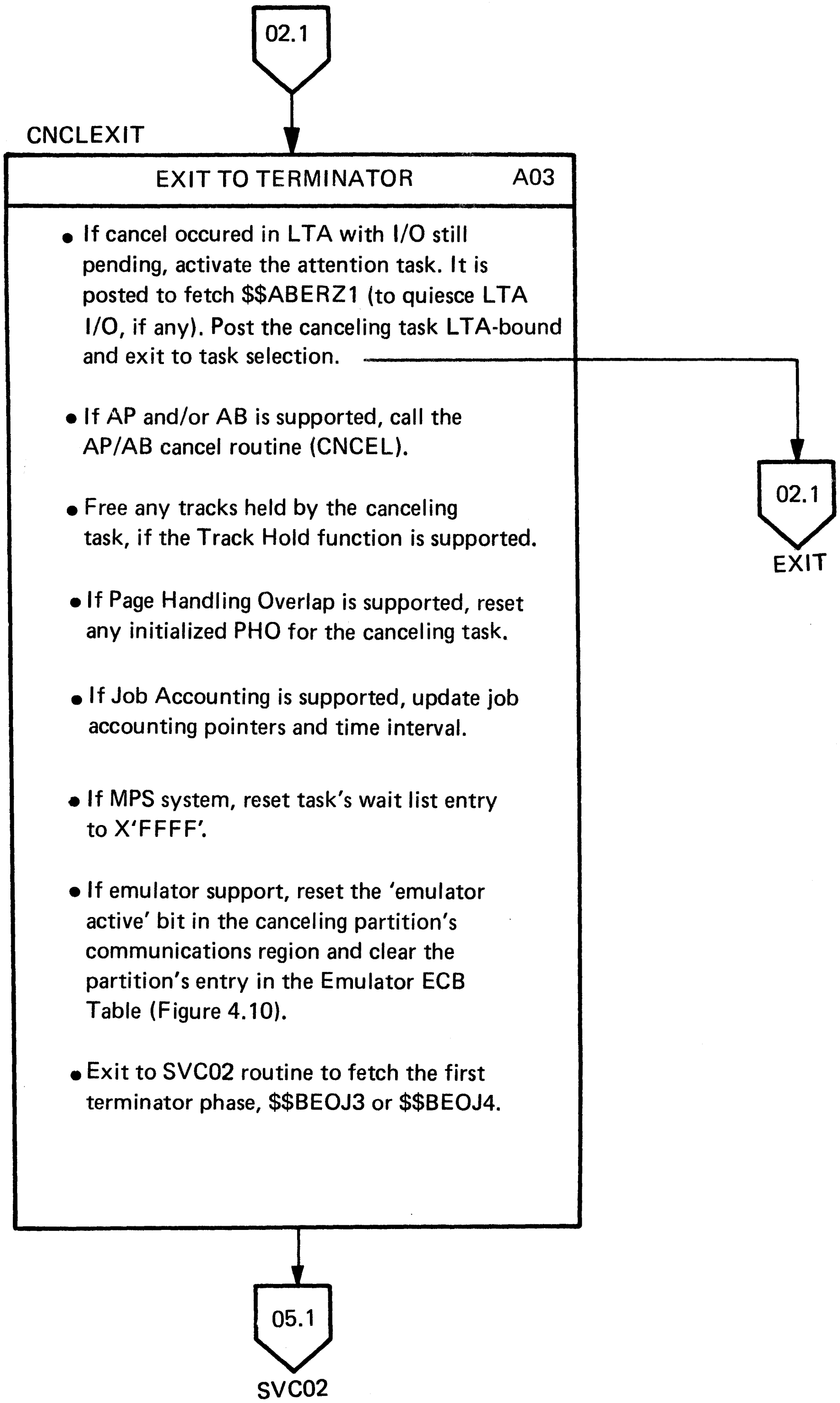

02.1
CNCLEXIT
EXIT TO TERMINATOR A03
• If cancel occured in LTA with I/O still pending, activate the attention task. It is posted to fetch $$ABERZ1 (to quiesce LTA I/O, if any). Post the canceling task LTA-bound and exit to task selection.
• If AP and/or AB is supported, call the AP/AB cancel routine (CNCEL).
• Free any tracks held by the canceling task, if the Track Hold function is supported.
• If Page Handling Overlap is supported, reset any initialized PHO for the canceling task.
• If Job Accounting is supported, update job accounting pointers and time interval.
• If MPS system, reset task's wait list entry to X'FFFF'.
• If emulator support, reset the 'emulator active' bit in the canceling partition's communications region and clear the partition's entry in the Emulator ECB Table (Figure 4.10).
• Exit to SVC02 routine to fetch the first terminator phase, $$BEOJ3 or $$BEOJ4.
02.1
EXIT
05.1
SVC02

Chart 02.3. Supervisor General Exit, DAT Dispatcher

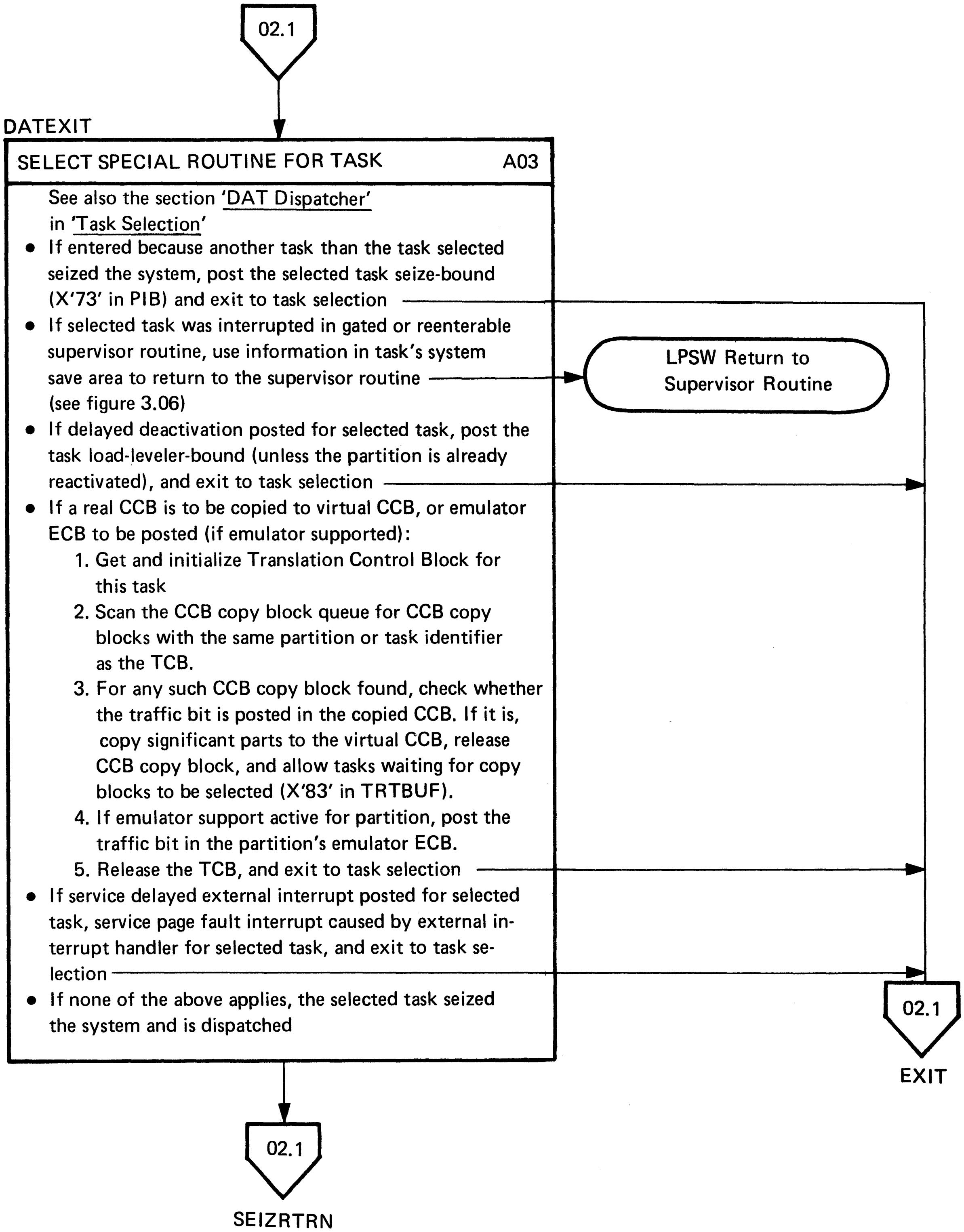
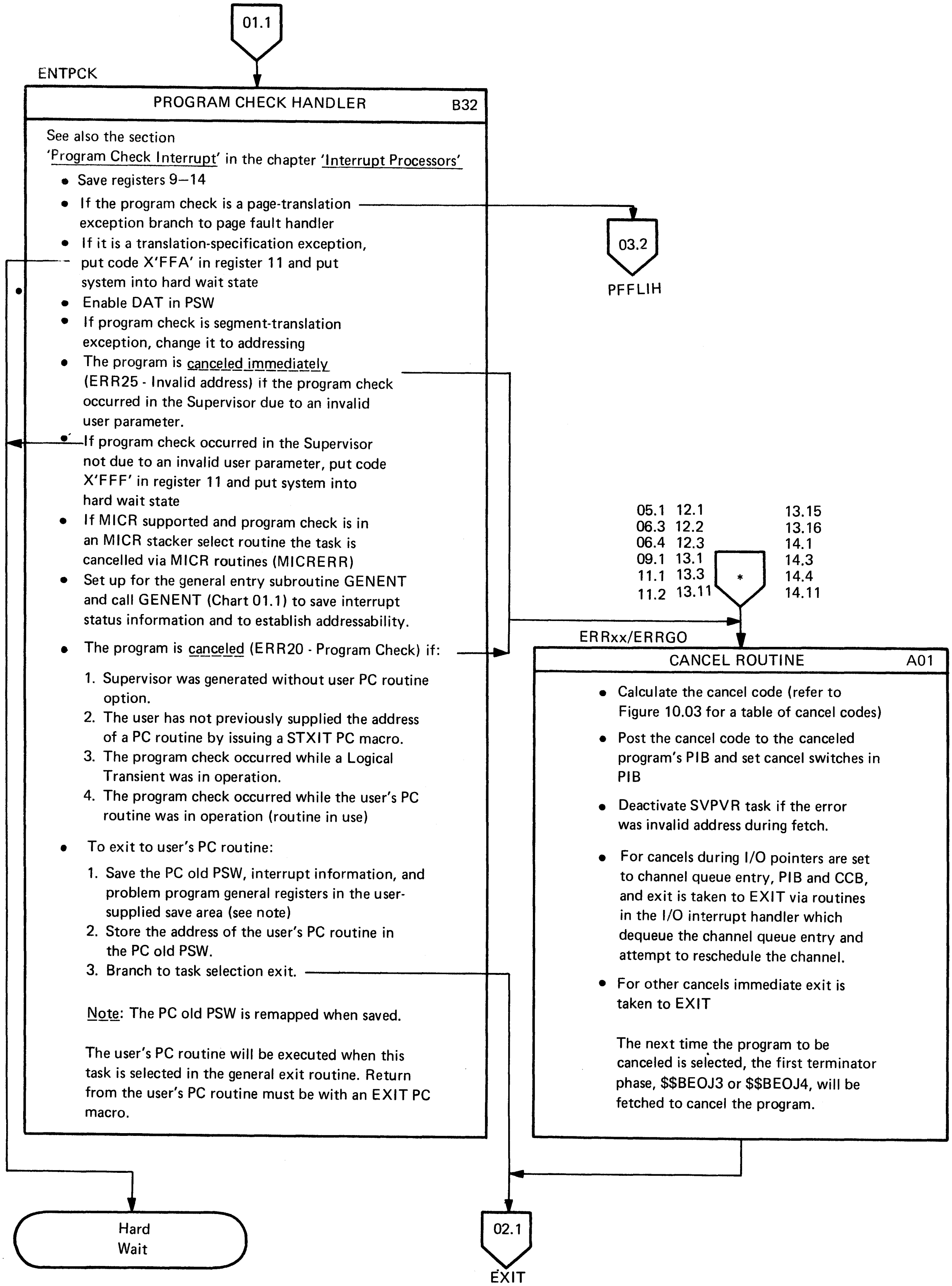
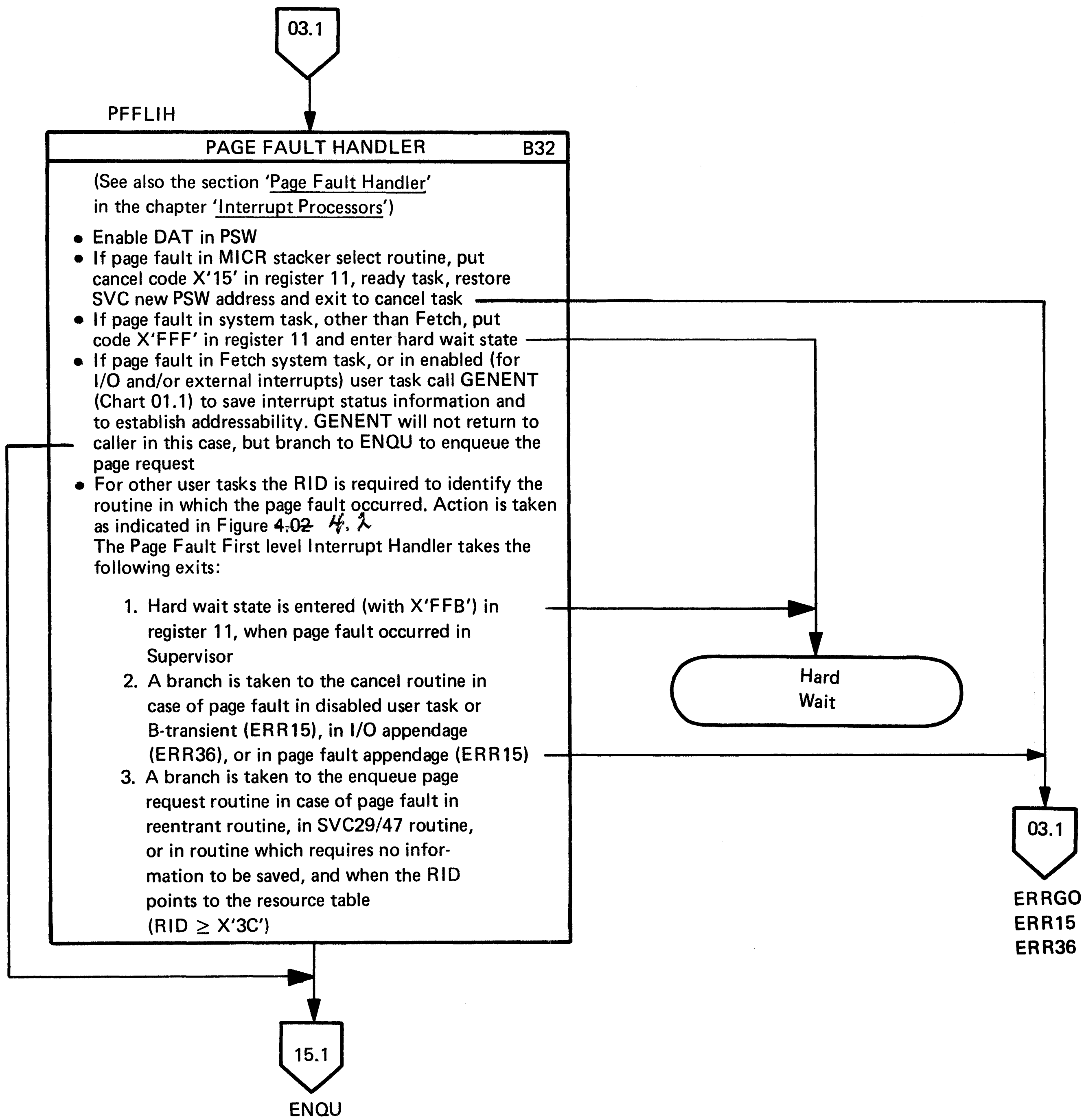


Chart 03.1. Program Check Handler, and Cancel Routine




03.1
PFFLIH
PAGE FAULT HANDLER
B32
(See also the section 'Page Fault Handler' in the chapter 'Interrupt Processors')
• Enable DAT in PSW
• If page fault in MICR stacker select routine, put cancel code X'15' in register 11, ready task, restore SVC new PSW address and exit to cancel task
• If page fault in system task, other than Fetch, put code X'FFF' in register 11 and enter hard wait state
• If page fault in Fetch system task, or in enabled (for I/O and/or external interrupts) user task call GENENT (Chart 01.1) to save interrupt status information and to establish addressability. GENENT will not return to caller in this case, but branch to ENQU to enqueue the page request
• For other user tasks the RID is required to identify the routine in which the page fault occurred. Action is taken as indicated in Figure ~~4.02~~ 4.2
The Page Fault First level Interrupt Handler takes the following exits:
1. Hard wait state is entered (with X'FFB') in register 11, when page fault occurred in Supervisor
2. A branch is taken to the cancel routine in case of page fault in disabled user task or B-transient (ERR15), in I/O appendage (ERR36), or in page fault appendage (ERR15)
3. A branch is taken to the enqueue page request routine in case of page fault in reentrant routine, in SVC29/47 routine, or in routine which requires no information to be saved, and when the RID points to the resource table (RID ≥ X'3C')
Hard Wait
03.1
ERRGO
ERR15
ERR36
15.1
ENQU

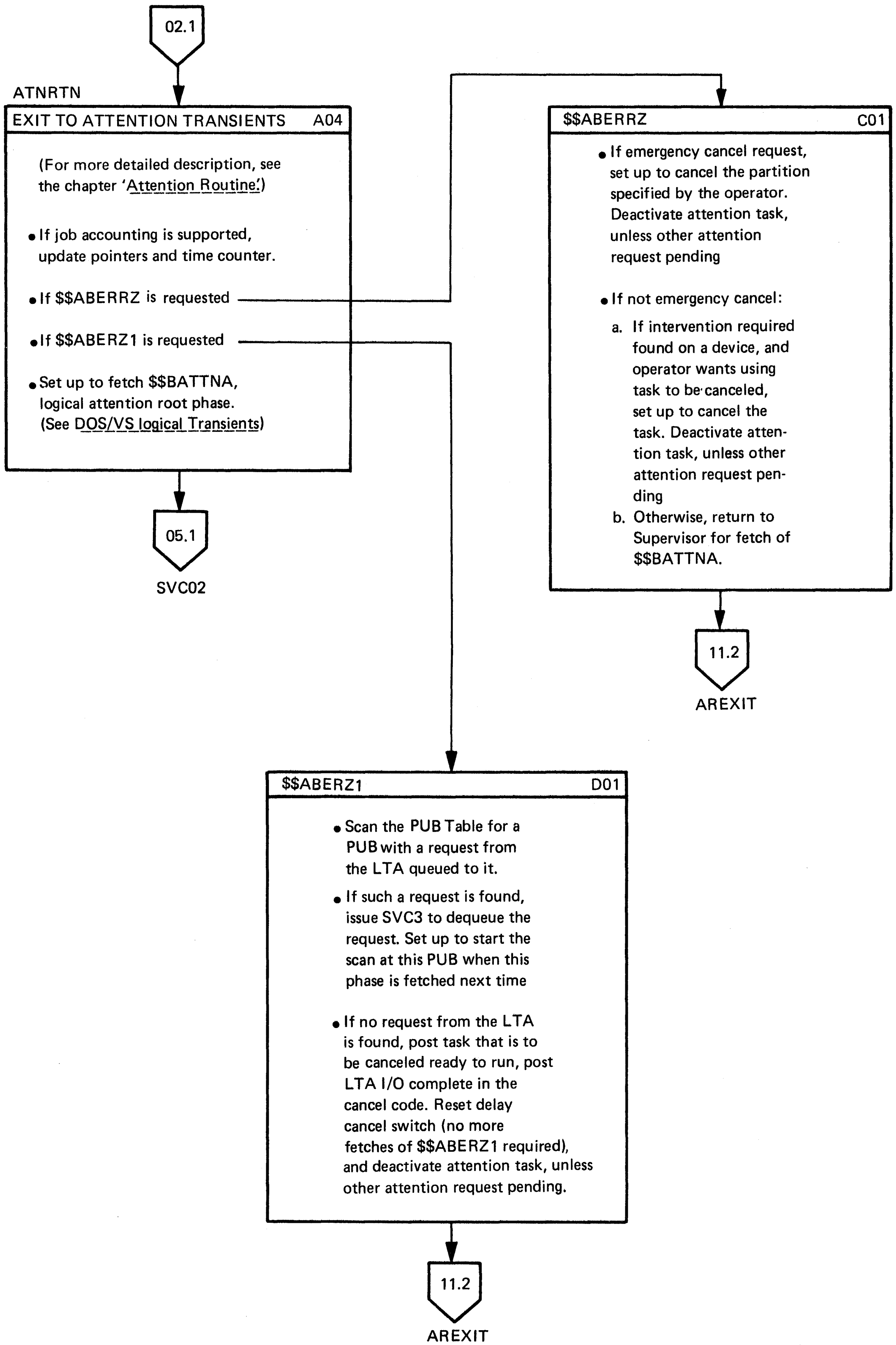

02.1
ATNRTN
EXIT TO ATTENTION TRANSIENTS A04
(For more detailed description, see the chapter 'Attention Routine'.)
• If job accounting is supported, update pointers and time counter.
• If $$ABERRZ is requested
• If $$ABERZ1 is requested
• Set up to fetch $$BATTNA, logical attention root phase. (See DOS/VS logical Transients)
05.1
SVC02
$$ABERRZ C01
• If emergency cancel request, set up to cancel the partition specified by the operator. Deactivate attention task, unless other attention request pending
• If not emergency cancel:
a. If intervention required found on a device, and operator wants using task to be canceled, set up to cancel the task. Deactivate attention task, unless other attention request pending
b. Otherwise, return to Supervisor for fetch of $$BATTNA.
11.2
AREXIT
$$ABERZ1 D01
• Scan the PUB Table for a PUB with a request from the LTA queued to it.
• If such a request is found, issue SVC3 to dequeue the request. Set up to start the scan at this PUB when this phase is fetched next time
• If no request from the LTA is found, post task that is to be canceled ready to run, post LTA I/O complete in the cancel code. Reset delay cancel switch (no more fetches of $$ABERZ1 required), and deactivate attention task, unless other attention request pending.
11.2
AREXIT

Chart 05.1. Supervisor Call Routines

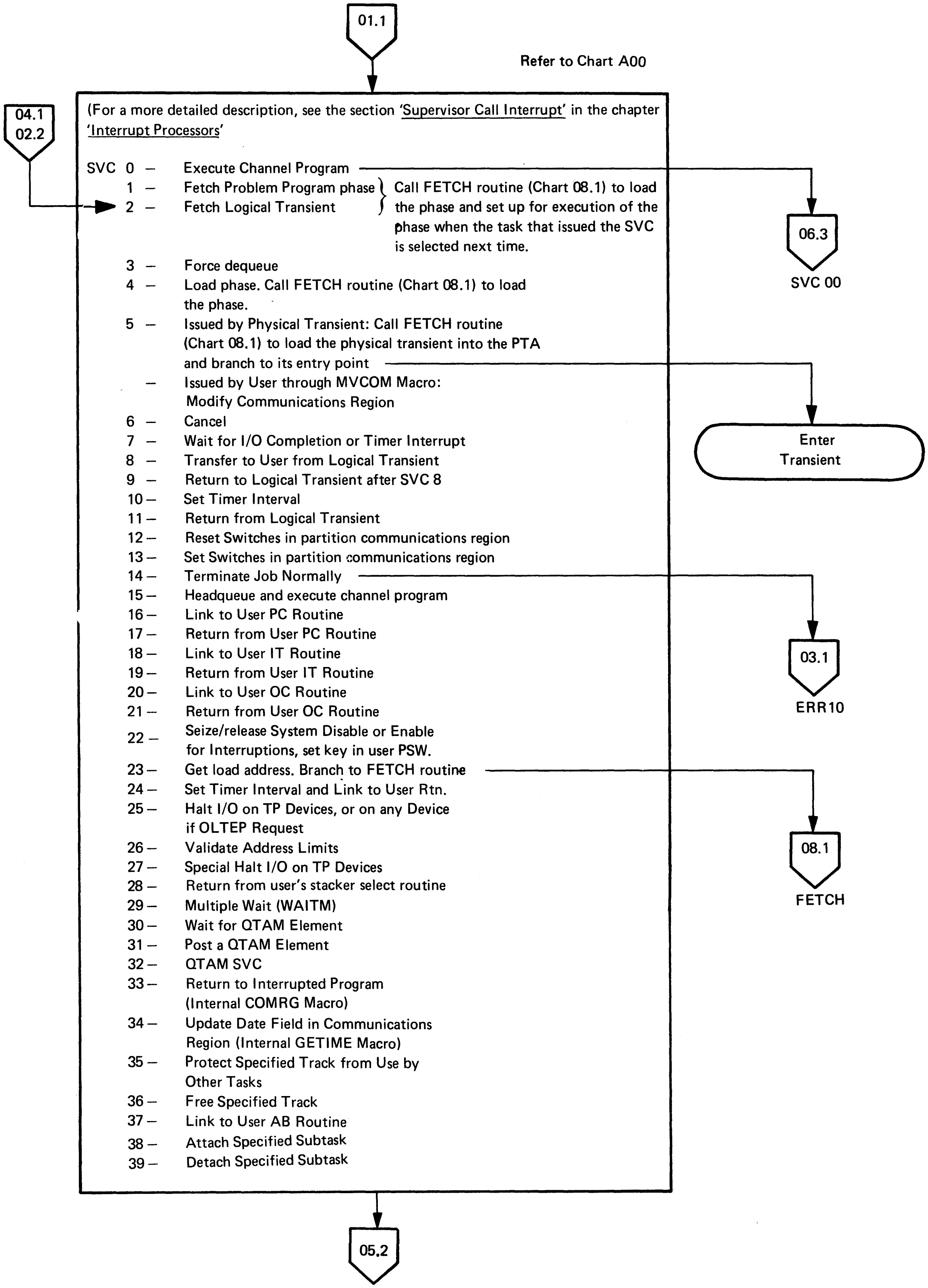


Chart 05.2. Supervisor Call Routines

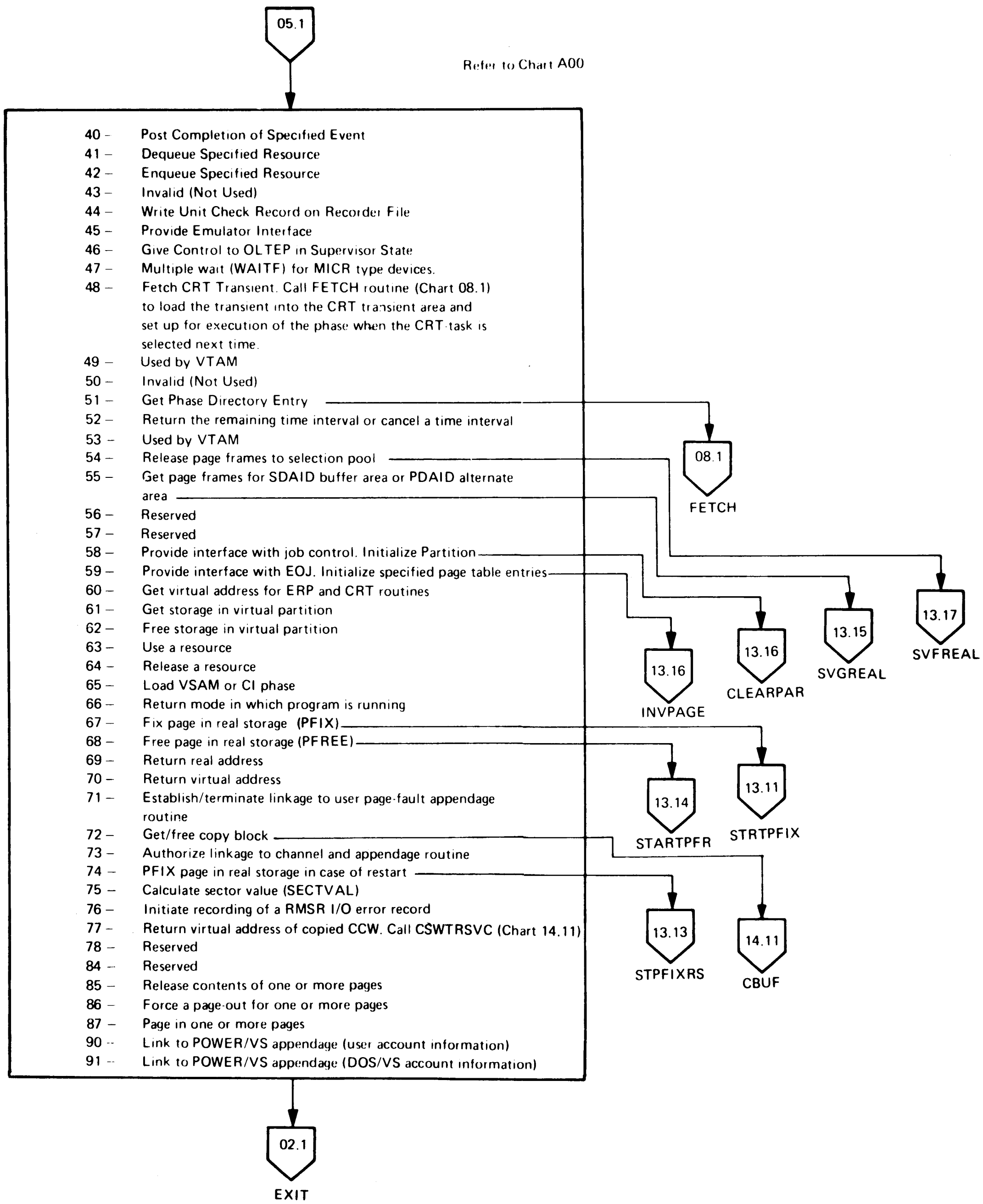


Chart 06.1. I/O Routines, I/O Interrupt Handler

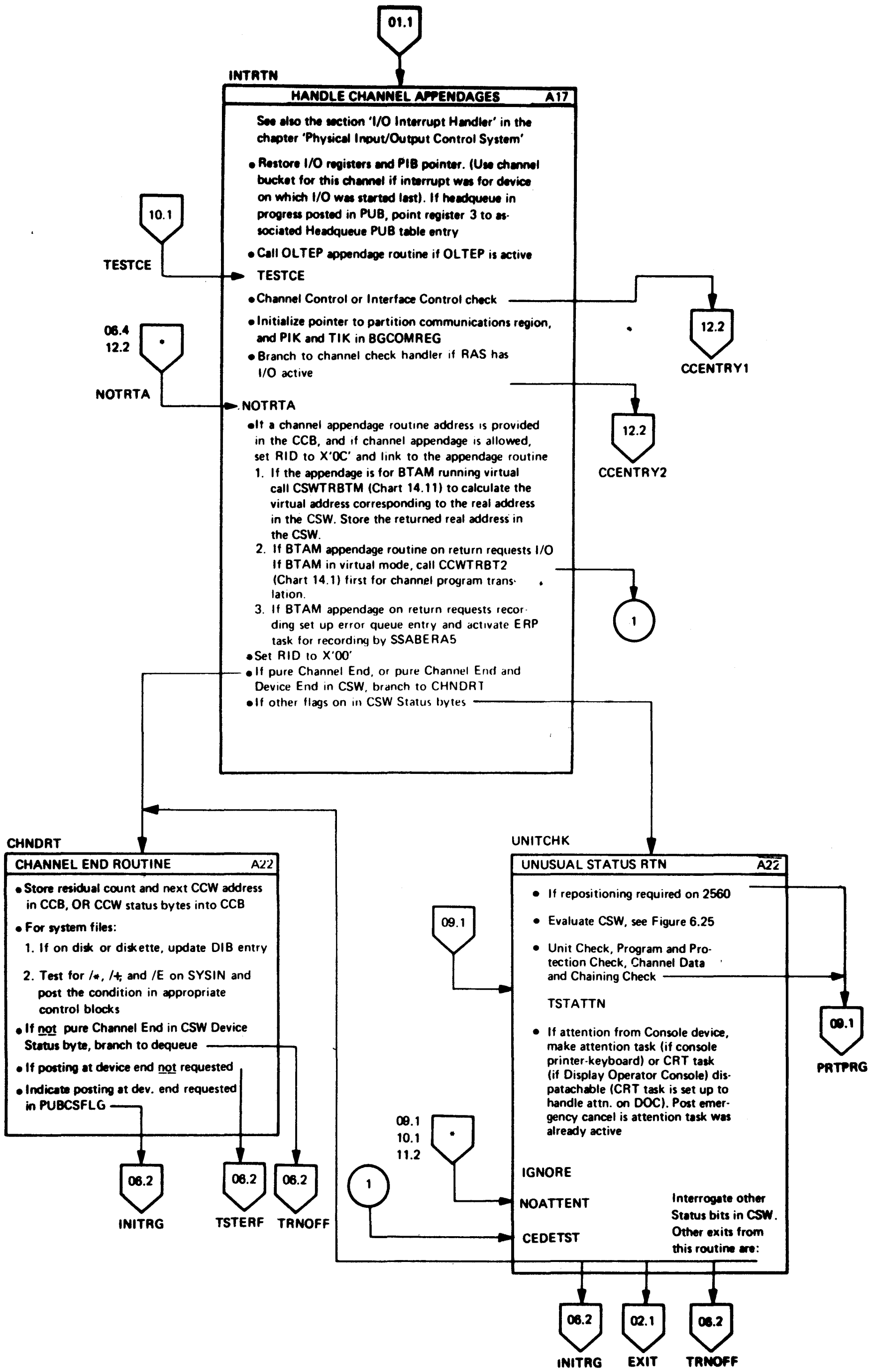


Chart 06.2. I/O Routines, I/O Interrupt Handler

See also the section 'I/O Interrupt Handler' in the chapter 'Physical Input/Output Control System'

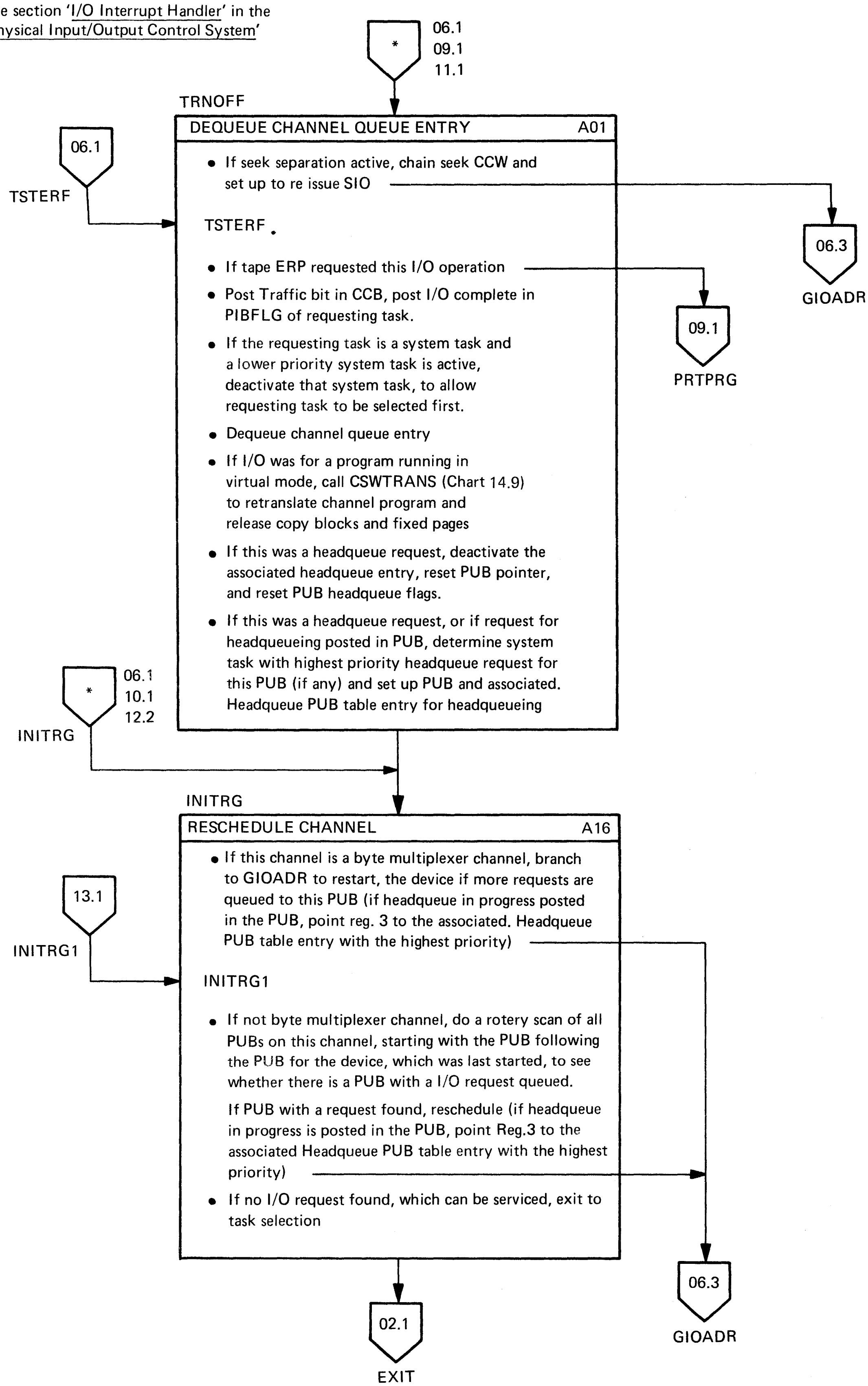


Chart 06.3. I/O Routines, Channel Scheduler

See also the section 'Channel Scheduler' in the chapter 'Physical Input/Output Control System'

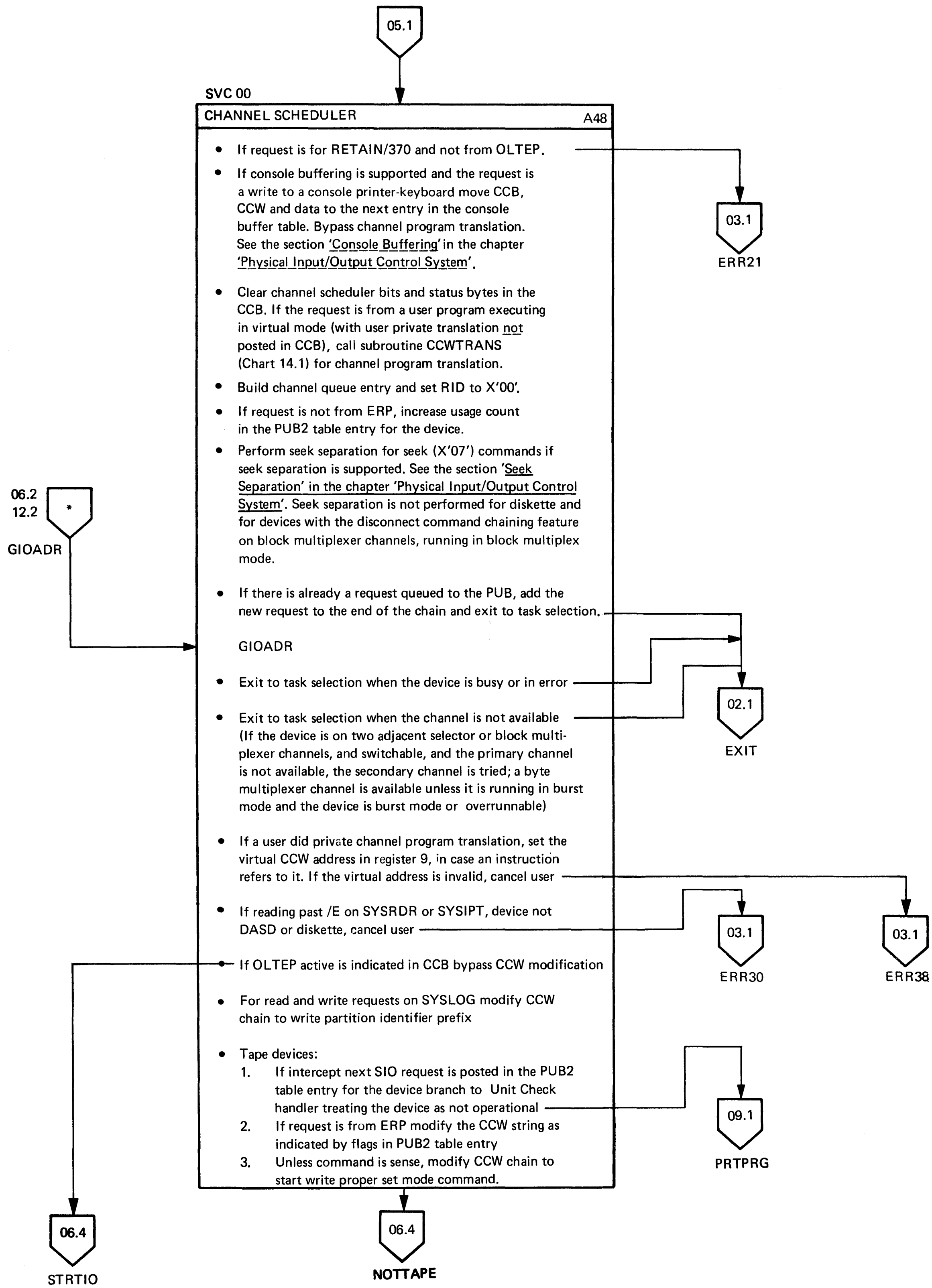


Chart 06.4. I/O Routines, Channel Scheduler

See also the section 'Channel Scheduler' in the chapter 'Physical Input/Output Control System'

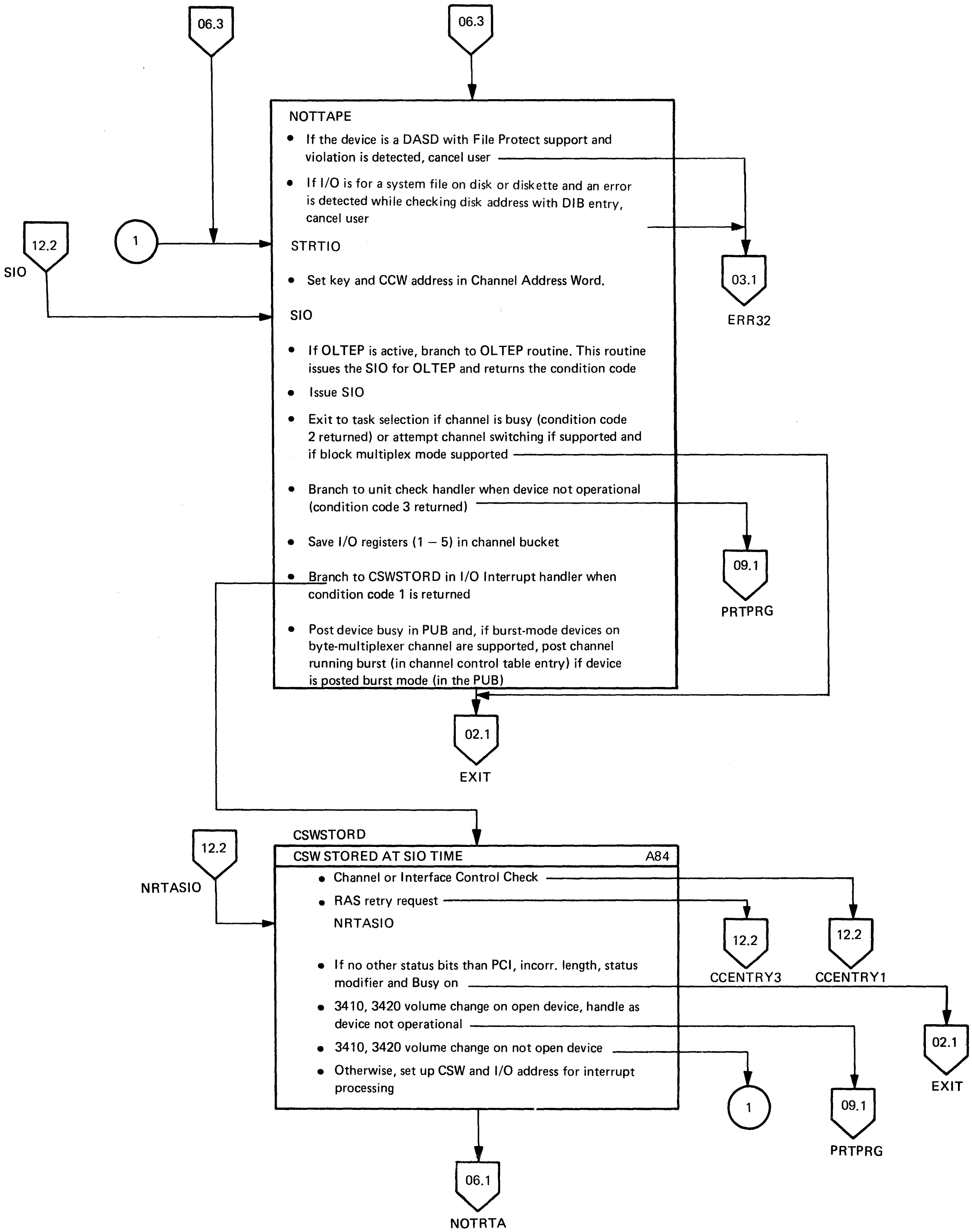


Chart 07.1. External Interrupt Routines

01.1

EXTRTN

TYPE

MICR SIGNAL / TIMER / KEY

MICR INTERRUPT — A41

This interrupt is ignored if MICR devices are not supported

1. If OLTEP active link to OLTEP appendage routine.
2. Locate the DTF table for the device causing the interrupt. See Figure 2.12 for tables PDTABA (DTF pointers) and PDTABB (DTF addresses). From the DTF table, get the channel and unit number of the MICR type device and data concerning the record in the buffer and the stacker select routine.
3. Issue the TIO to clear the pending interrupt. If any error conditions are detected, branch to the TIO error recovery routine for 1419/1270/1275 or for 1419D. If an off-line sort is requested with 1419D, the subsequent SIO will be bypassed.
4. After a successful TIO has been completed, update record and buffer data, prepare CCW, and exit to the user stacker select routine to get the pocket selection for the last document read.
5. The Supervisor is reentered via an EXIT (MR) macro. Prepare CCWs and buffer for batch numbering and/of auto select, if required.
6. Issue an SIO to stack the document. If any error conditions are detected, branch to the SIO error recovery routine for 1419/1270/1275 or for 1419D. If more than one MICR type device interrupted simultaneously, process these interrupts beginning at step 2 above.
7. Check for IT or OC interrupts that may have occurred with the MICR interrupt:

   Timer Interrupt - Update the job accounting timer to a minimum value so that it will interrupt again without any other status. It will then be processed as described in the Timer Interrupt block.
   Key Interrupt - Process immediately as described in the Key Interrupt block.

EXTERNAL INTERRUPT, TIMER — A38

If no further interval is pending, the timer (SYSTIMER) is reset to the maximum value X'FF000000'. If job accounting is supported (JA=YES), SYSTOD (X'54') is set to the current time plus 15.5 hours.
If there are pending intervals, the following actions are taken:

- SYSTIMER is set to the time interval of the next interrupt (first entry in ITREQ is moved to SYSTIMER).
- SYSTOD is set to the time of day at which the interrupt is to occur (first entry in ITREQ is added to SYSTOD).
- ITREQ is updated (first entry in ITREQ is subtracted from the other entries in ITREQ, if any).
- The first entry in ITREQ is deleted and the other intervals, if any, are moved up one entry.

With User TECB Address: The event bit is posted in the user's TECB (timer event control block) and the timer supported program is included in task selection. The TECB address is cleared in the IT option table (ITTAB).

With User IT Routine: To exit to the user's IT routine:

1. Save the interrupt status information and timer-supported-program registers in the user-supplied save area (72 bytes).
2. Store the address of the user IT routine in this save area.
3. Branch to general exit. The user IT routine will be executed when this task is selected in the general exit routine. Return from the user's IT routine must be with an EXIT IT macro.
   See Figures 4.07 and 4.08 for the timer tables (ITTAB and ITREQ). Refer to Figure 6.20 for the format of the CCB (TECB has the same format as the CCB. Only the event bit X'80' in byte 2 is used).

Note: If a B-transient is operating for the timer supported program when the timer interrupt occurs for a user IT routine, the exit is deferred. When on return from LTA the SVC11 routine finds that a timer interrupt is pending, a branch is taken to the timer interrupt handler to resume processing of the interrupt.

EXTERNAL INTERRUPT, KEY — A38

The attention routine PIB is posted to include the attention routine in task selection. The nonresident attention routine ($$BATTNA) processes the request when the attention task is selected in general exit.

Note: If Supervisor was generated without user OC (operator communications) routine option, the interrupt is ignored.
If timer interrupt occurred at the same time, handle it. Otherwise, exit to task selection.

02.1

EXIT

Chart 08.1. Fetch Routine

(For information about relocating load, see the section 'Relocating load' in the chapter 'Program Retrieval'.)

05

FETCH ROUTINE

FETCH
FETCHERP
CRTFCH2

FETCH A PHASE B03, B10

For a more detailed description, see the section 'Fetch Routine' in this logic manual.

Objectives:

1. Load a problem program phase or transient program from a private core image library (PCIL) or from a system core image library, unless it is in the Shared Virtual Area (SVA) or TXT=NO was specified.
2. Supply the load address for a phase (SVC 23), or
3. Supply the directory entry for a phase (SVC 51).

- Search for a directory entry for the requested phase. An active (filled in) directory entry may be found in storage (for example in Local or System Directory List). If no active directory entry is found in storage, the PCIL (if supported and assigned) and SCIL directories are to be searched. The order of the search depends on the type of phase ($ phase or not), and on the option specified by the user.
- The directory block containing the directory entry (if the phase is in the library being searched) is read into the PTA, if IDRA=NO was specified at sysgen. It is read into the Independent Directory Read-In Area (IDRA) if IDRA= YES was specified. When searching for a CRT transient ($$BOCRTx), the directory block is read into the CRT transient area (CRTTRNS).

  If the fetch routine was entered by the SVC 23 routine, move the load address of the phase (relocated load address for relocatable phase) to the user area and exit to task selection → 02 EXIT

  If the fetch routine was entered by the SVC 51 routine, move the requested number of halfwords of the directory entry of the phase to the user area and exit to task selection → 02 EXIT

- Relocate the entry address if an overriding load address has been specified, and pass this address to the user in general register 1.
- Load the phase, unless it is in the SVA or TXT=NO was specified. (Phase in SVA is only used if there are 8 or more page frames in the main page pool.)
- Store the end address of the phase in the partition's communications region, unless the phase is in the SVA or TXT=NO was specified.

RETURN TO CALLER

FETCH A PHASE, phase not found
If the phase to be fetched is not found. 1. For SVC 23 and 51 the task is not canceled; return to task selection. 2. If an in-storage directory is found and TXT NO was specified: post directory entry active and phase not found. The task is not canceled; return to task selection. 3. In the other cases: X'22' cancel code is set. The task is canceled; return to task selection.

Chart 09.1. Unit Check Routine

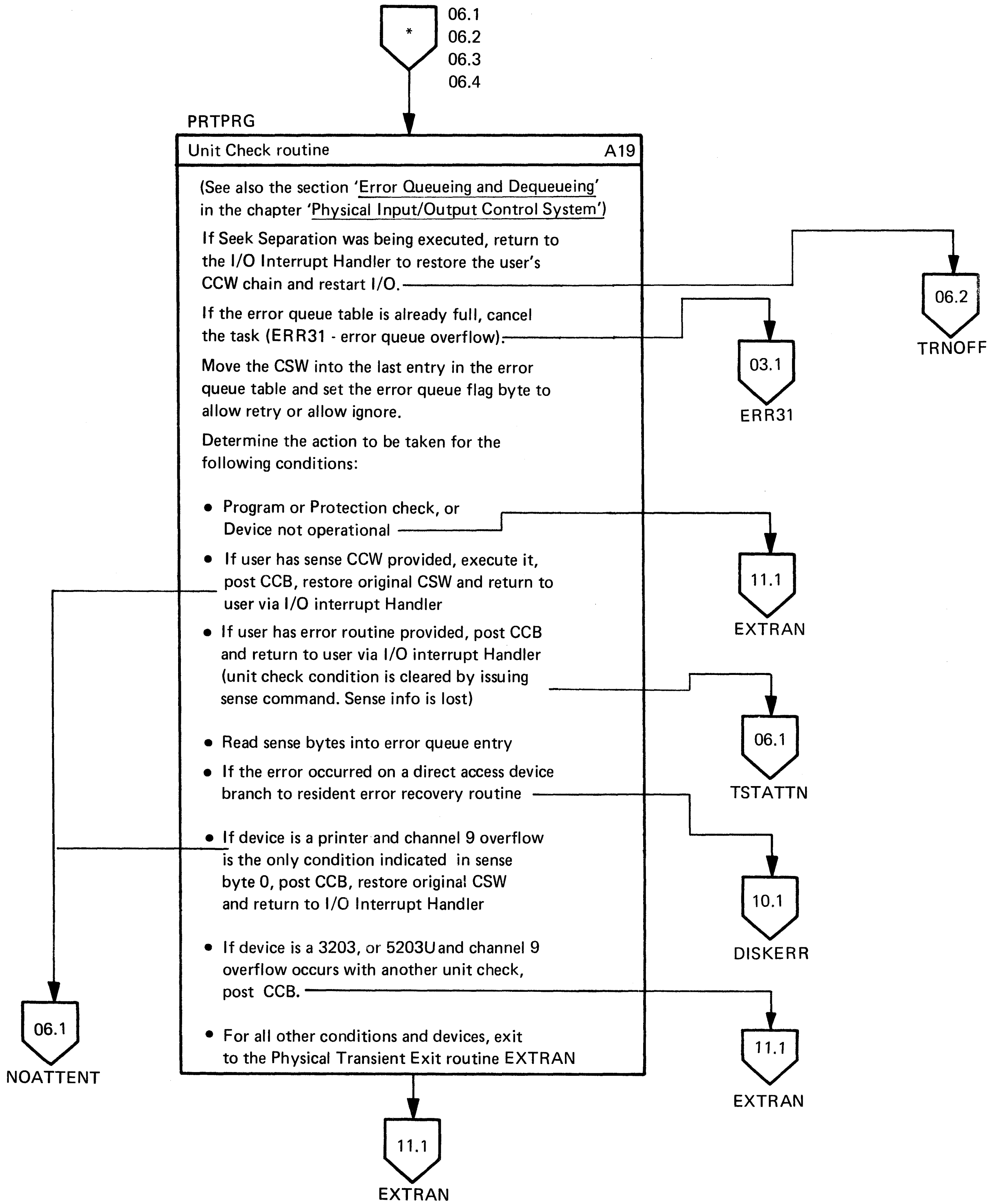
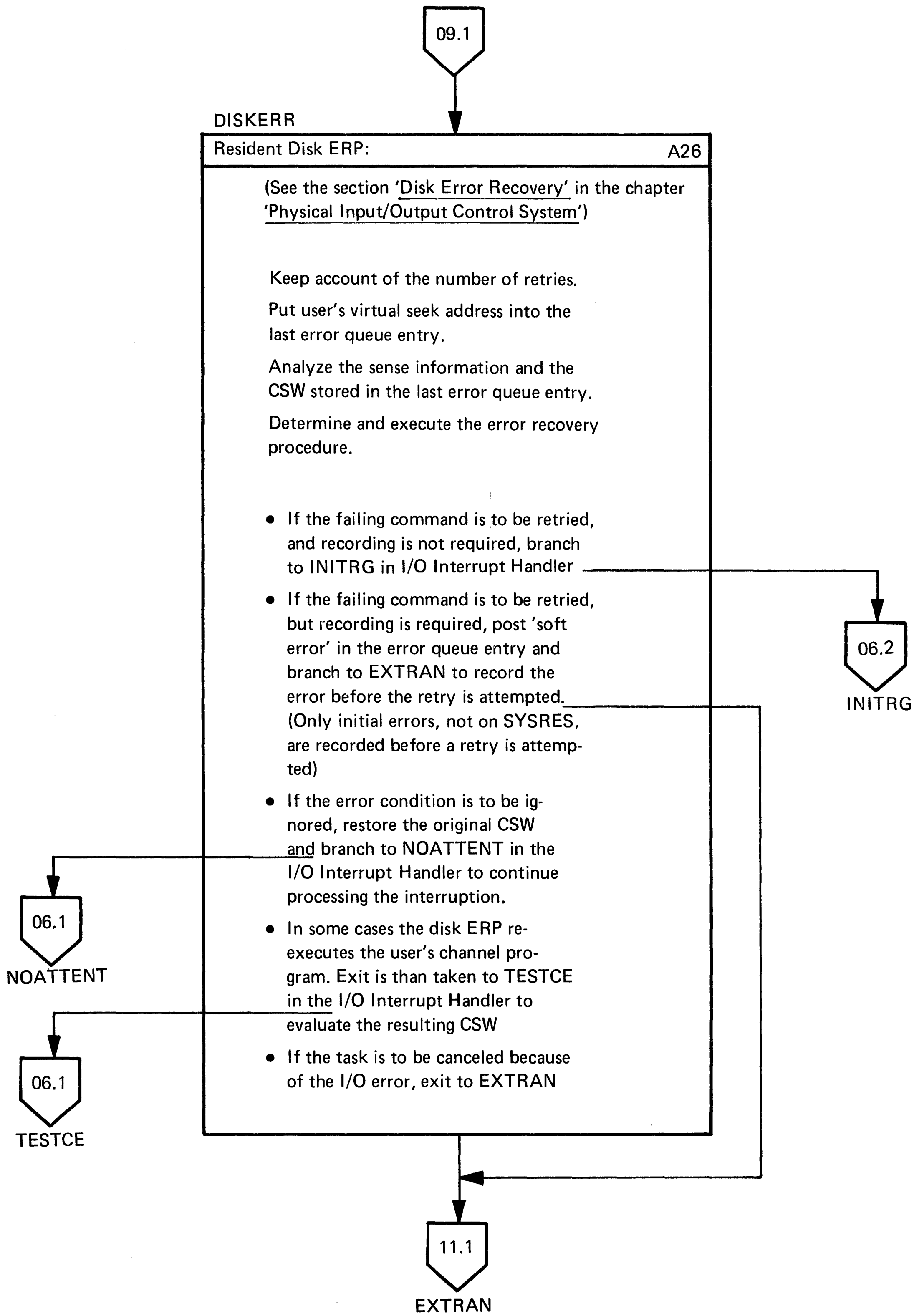
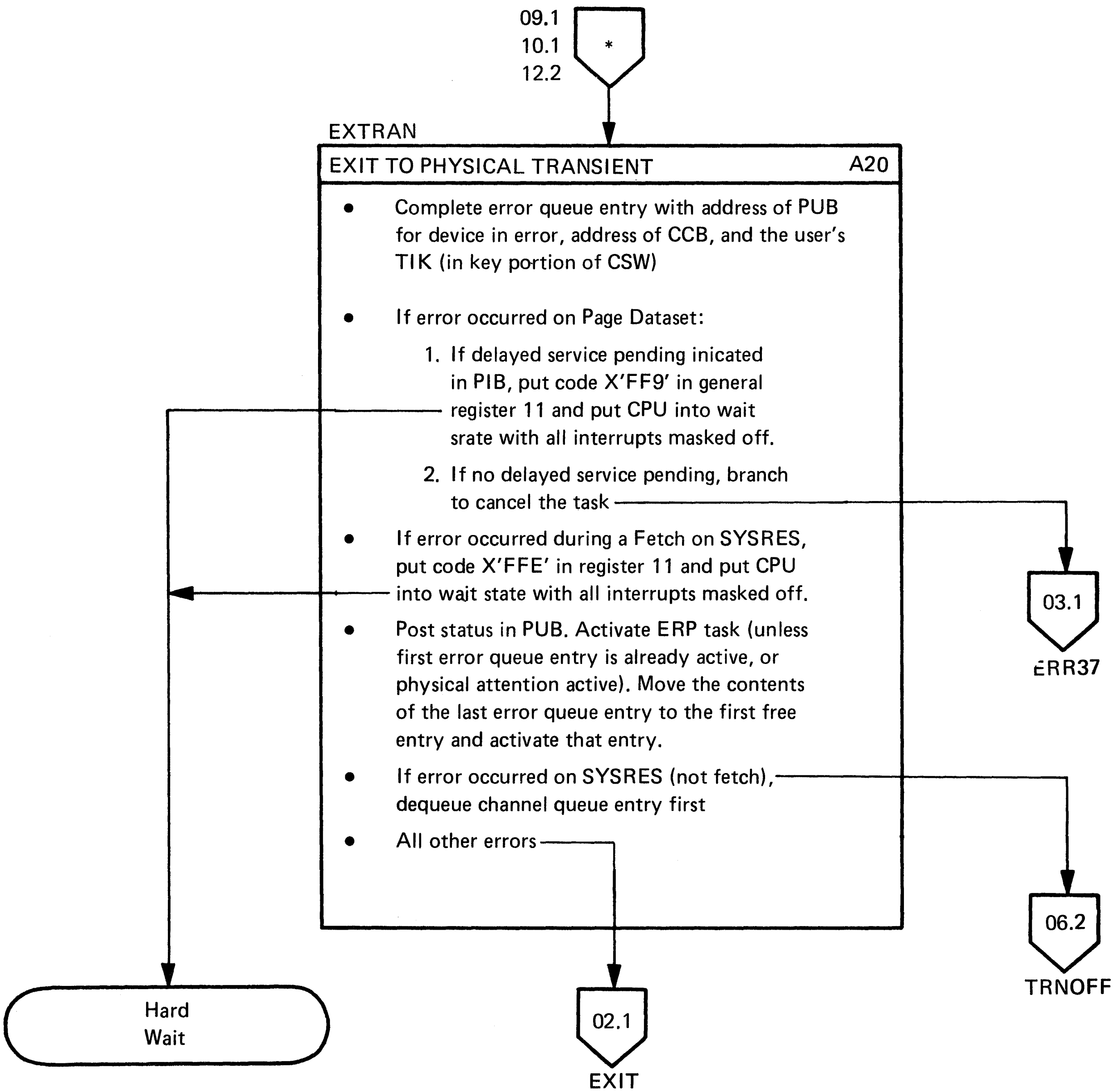
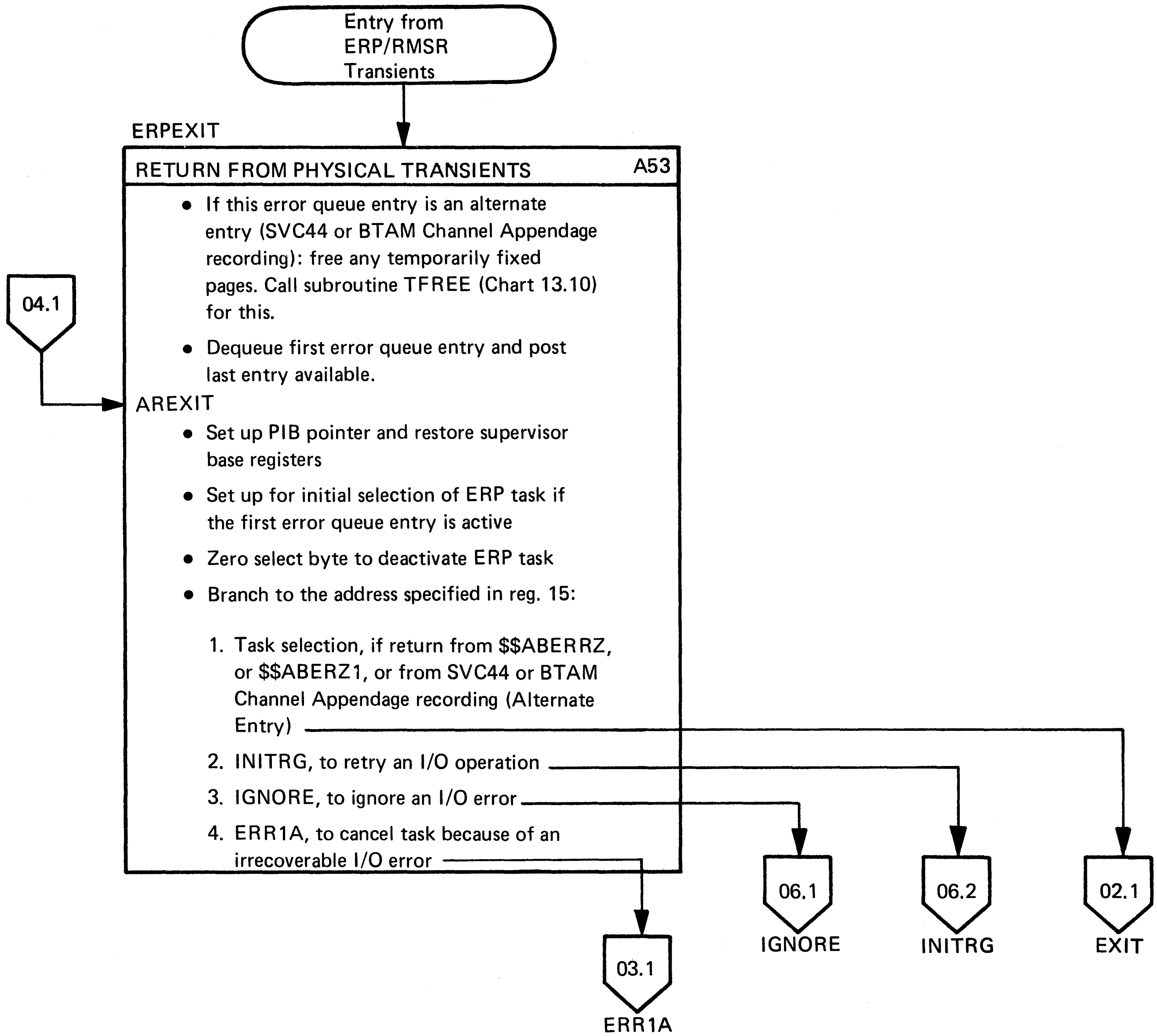


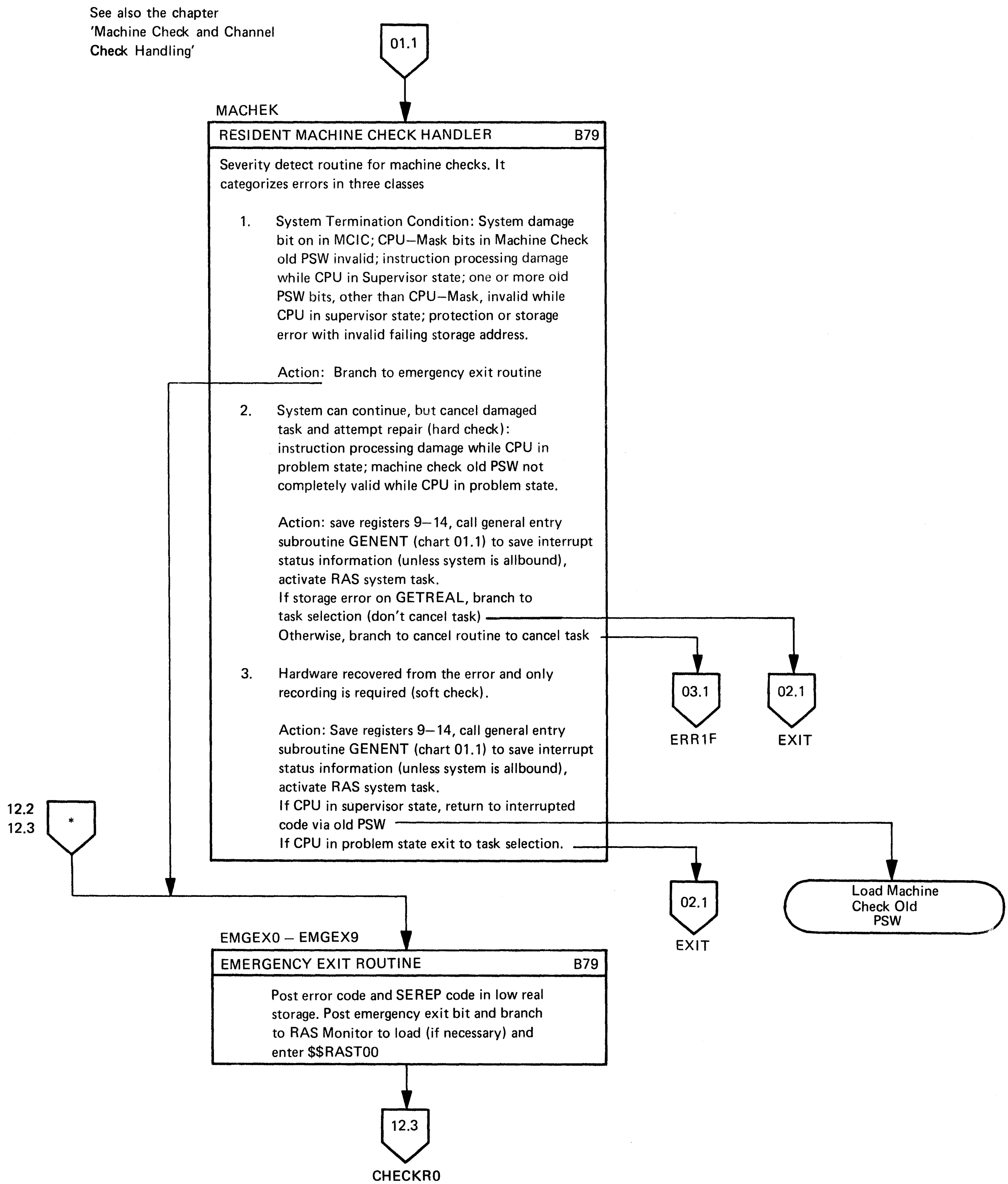
Chart 10.1. Resident Disk Error Recovery Routine




09.1
10.1
12.2
*
EXTRAN
EXIT TO PHYSICAL TRANSIENT
A20
• Complete error queue entry with address of PUB for device in error, address of CCB, and the user's TIK (in key portion of CSW)
• If error occurred on Page Dataset:
1. If delayed service pending inicated in PIB, put code X'FF9' in general register 11 and put CPU into wait srate with all interrupts masked off.
2. If no delayed service pending, branch to cancel the task
• If error occurred during a Fetch on SYSRES, put code X'FFE' in register 11 and put CPU into wait state with all interrupts masked off.
• Post status in PUB. Activate ERP task (unless first error queue entry is already active, or physical attention active). Move the contents of the last error queue entry to the first free entry and activate that entry.
• If error occurred on SYSRES (not fetch), dequeue channel queue entry first
• All other errors
03.1
ERR37
06.2
TRNOFF
Hard
Wait
02.1
EXIT



Entry from ERP/RMSR Transients
ERPEXIT
RETURN FROM PHYSICAL TRANSIENTS
A53
• If this error queue entry is an alternate entry (SVC44 or BTAM Channel Appendage recording): free any temporarily fixed pages. Call subroutine TFREE (Chart 13.10) for this.
• Dequeue first error queue entry and post last entry available.
04.1
AREXIT
• Set up PIB pointer and restore supervisor base registers
• Set up for initial selection of ERP task if the first error queue entry is active
• Zero select byte to deactivate ERP task
• Branch to the address specified in reg. 15:
1. Task selection, if return from $$ABERRZ, or $$ABERZ1, or from SVC44 or BTAM Channel Appendage recording (Alternate Entry)
2. INITRG, to retry an I/O operation
3. IGNORE, to ignore an I/O error
4. ERR1A, to cancel task because of an irrecoverable I/O error
03.1
ERR1A
06.1
IGNORE
06.2
INITRG
02.1
EXIT


See also the chapter 'Machine Check and Channel Check Handling'
01.1
MACHEK
RESIDENT MACHINE CHECK HANDLER B79
Severity detect routine for machine checks. It categorizes errors in three classes
1. System Termination Condition: System damage bit on in MCIC; CPU–Mask bits in Machine Check old PSW invalid; instruction processing damage while CPU in Supervisor state; one or more old PSW bits, other than CPU–Mask, invalid while CPU in supervisor state; protection or storage error with invalid failing storage address.
Action: Branch to emergency exit routine
2. System can continue, but cancel damaged task and attempt repair (hard check): instruction processing damage while CPU in problem state; machine check old PSW not completely valid while CPU in problem state.
Action: save registers 9–14, call general entry subroutine GENENT (chart 01.1) to save interrupt status information (unless system is allbound), activate RAS system task.
If storage error on GETREAL, branch to task selection (don't cancel task)
Otherwise, branch to cancel routine to cancel task
3. Hardware recovered from the error and only recording is required (soft check).
Action: Save registers 9–14, call general entry subroutine GENENT (chart 01.1) to save interrupt status information (unless system is allbound), activate RAS system task.
If CPU in supervisor state, return to interrupted code via old PSW
If CPU in problem state exit to task selection.
03.1
ERR1F
02.1
EXIT
12.2
12.3
*
02.1
EXIT
Load Machine Check Old PSW
EMGEX0 – EMGEX9
EMERGENCY EXIT ROUTINE B79
Post error code and SEREP code in low real storage. Post emergency exit bit and branch to RAS Monitor to load (if necessary) and enter $$RAST00
12.3
CHECKR0

See also the chapter 'Machine Check and CHannel Check Handling'

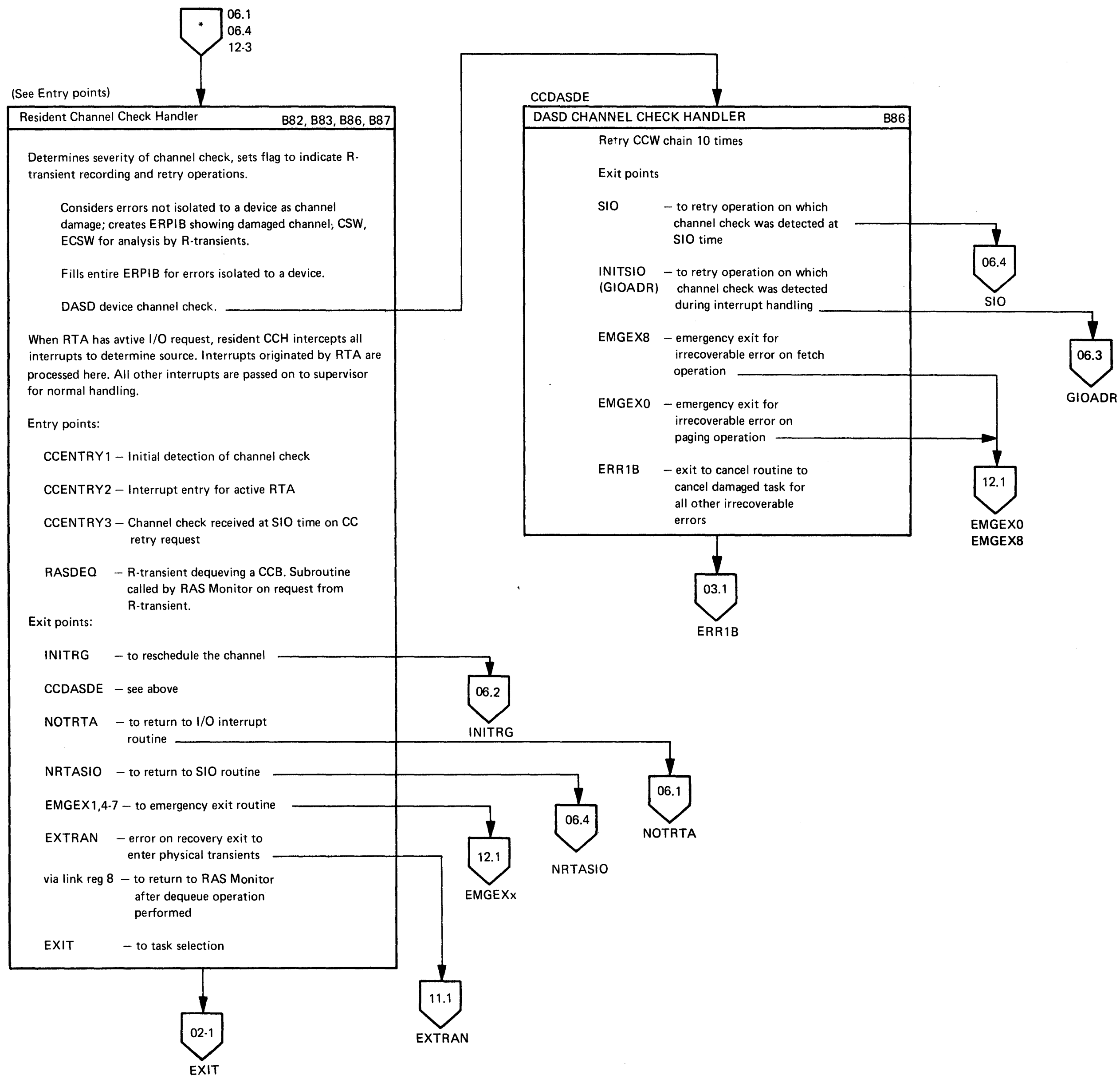


Chart 12.3. MCH/CCH, RAS Monitor

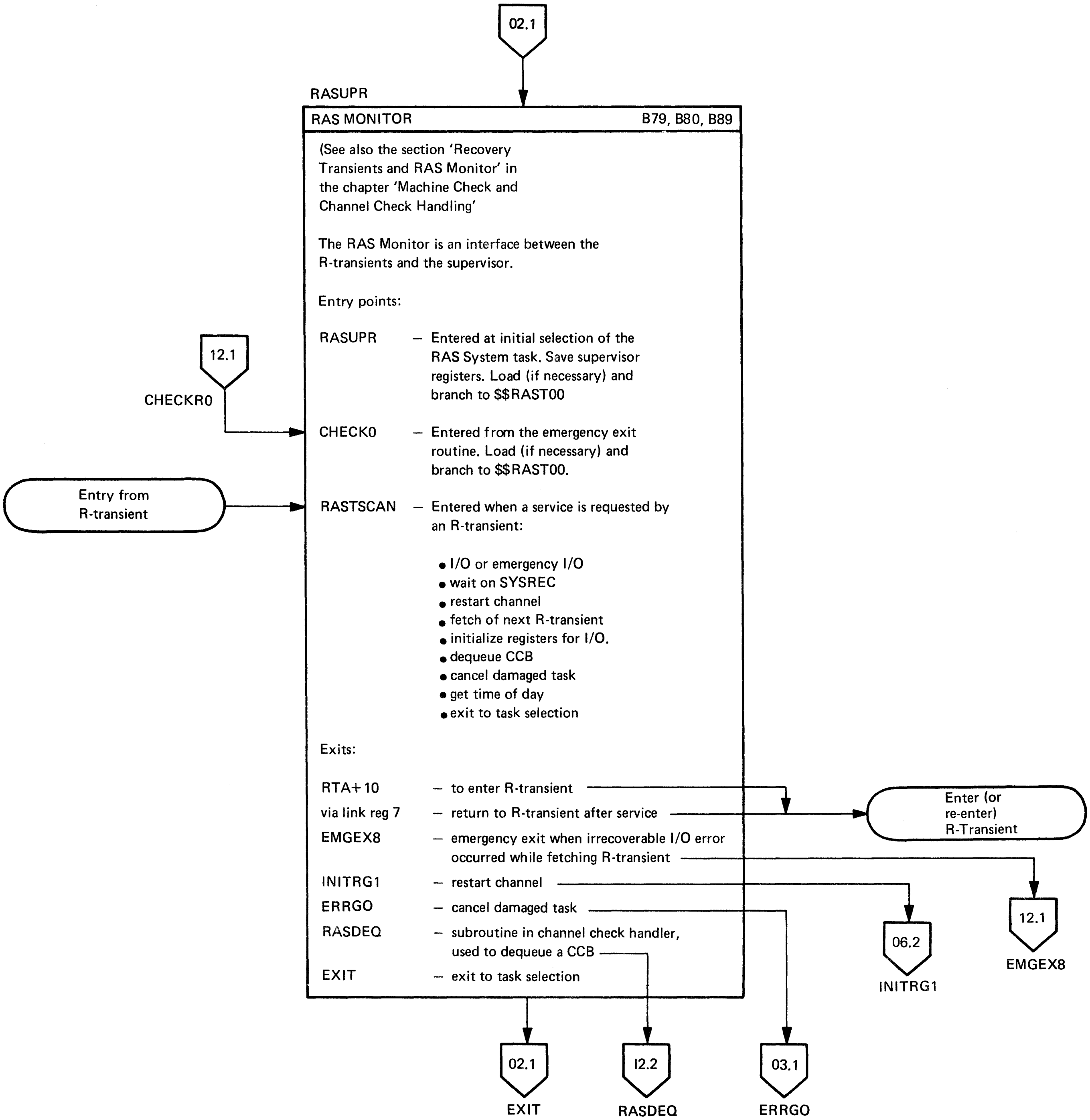


Chart 13.1. Page Management, Enqueue Request tc Page Queue

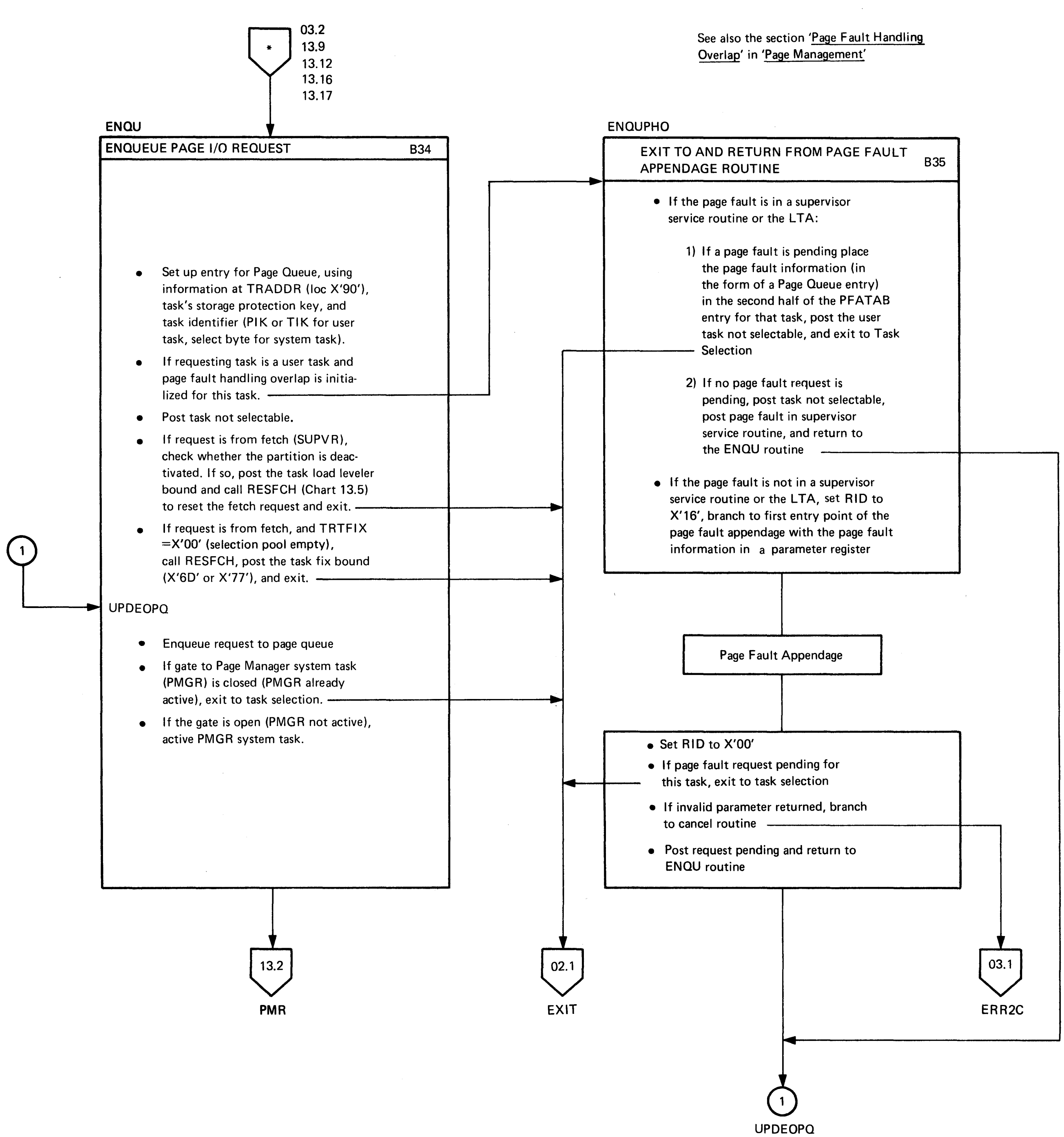


Chart 13.2. Page Management, Handle Page Queue Entry

* 13.1 13.3 13.4

PMR

HANDLING OF PAGE QUEUE ENTRY B36

This routine handles the first entry in the page queue.

- If the request is not from GETREAL and the requested page is not in real storage:
  1) Update reentry rate counter in case of page reentry (see section 'Partition Deactivation').
  2) Call BQCL (Chart 13.5) to select a page frame from the selection pool.
  3) If the requested page must be read from the page data set (which contains a valid copy), call CPGIN (Chart 13.6) to update page-in counters.

PMR1

- If the selected page frame (see above), or the requested page frame (GETREAL) is being used:
  1) Post reentry rate table (see section 'Partition Deactivation')
  2) If contents of the page frame were changed, indicate the necessity of page-out.
  3) Insert page frame's storage key into page table entry of page that is occupying the page frame.

PMR2

- The following only applies when the requested page is not yet in real storage.
  1) Remove page frame from selection pool.
  2) Call PAGEIO (Chart 13.5) if page-in and/or page-out is to be performed.
  3) Set storage key in page frame, using the key from page table entry of requested page, and insert page frame number into this page table entry.
  4) If GETREAL request, post invalid address space in corresponding page table entry (high-order bit on) and exit to dequeue the page queue entry → 13.3 DEQU
  5) Add page frame to the selection pool again, and append it to the end of the Hold Queue.
  6) If, however, the request is not GETREAL and the requested page is found to be non-addressable (high-order bit in page table entry on), instead of action 3 and 5, the storage key is set to zero and the page frame is added to the top of Q00 and posted unused.
  7) If delayed exchanging is posted, reset NFF (bit 12) in 'old' page frame table entry and call EXCH (Chart 13.13) and DPFIX (Chart 13.11) to exchange and PFIX

PMRA

- If PFIX request, call PFIXPGE (Chart 13.12) to PFIX the page If PFIX is not successful exit to dequeue the page queue entry without posting the requesting task → 13.3 DELETE
- If TFIX request, call DTFIX (Chart 13.9) to TFIX the page. If TFIX not successful, exit to DEQUX → 13.4 DEQUX

→ 13.3 DEQU

Chart 13.3. Page Management, Dequeue Request from Page Queue

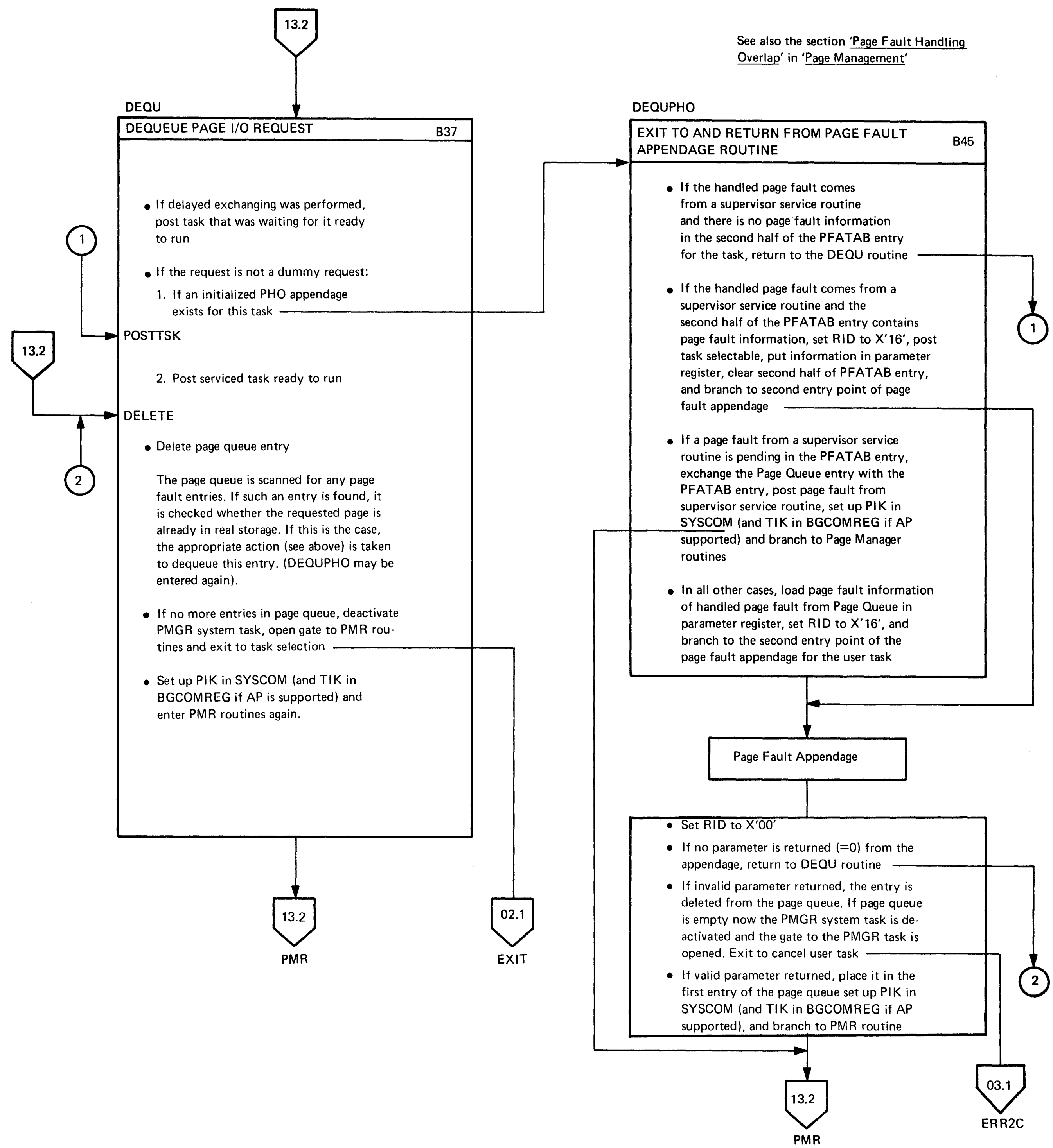
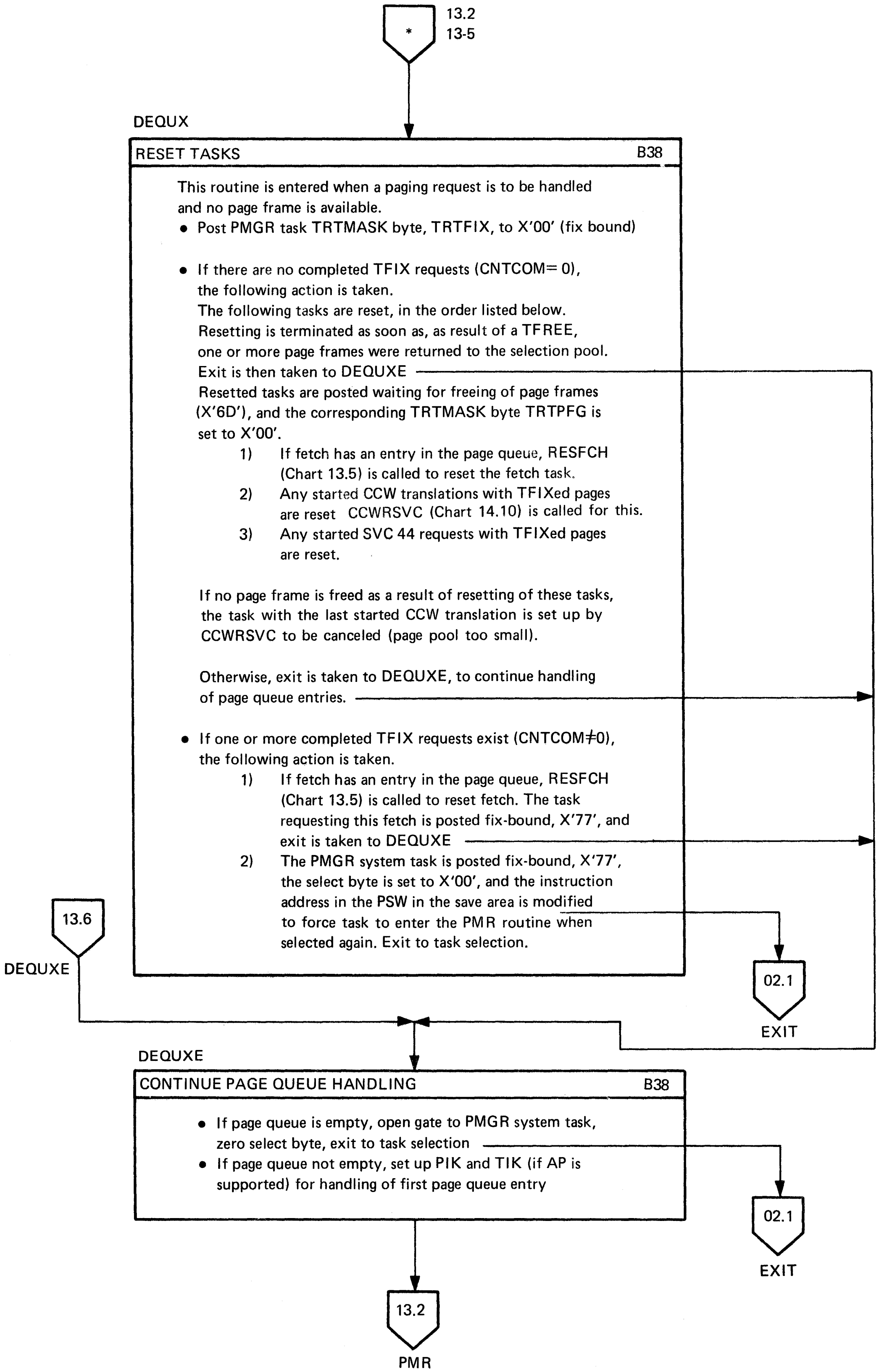


Chart 13.4. Page Management, Handle Selection Pool Empty Condition



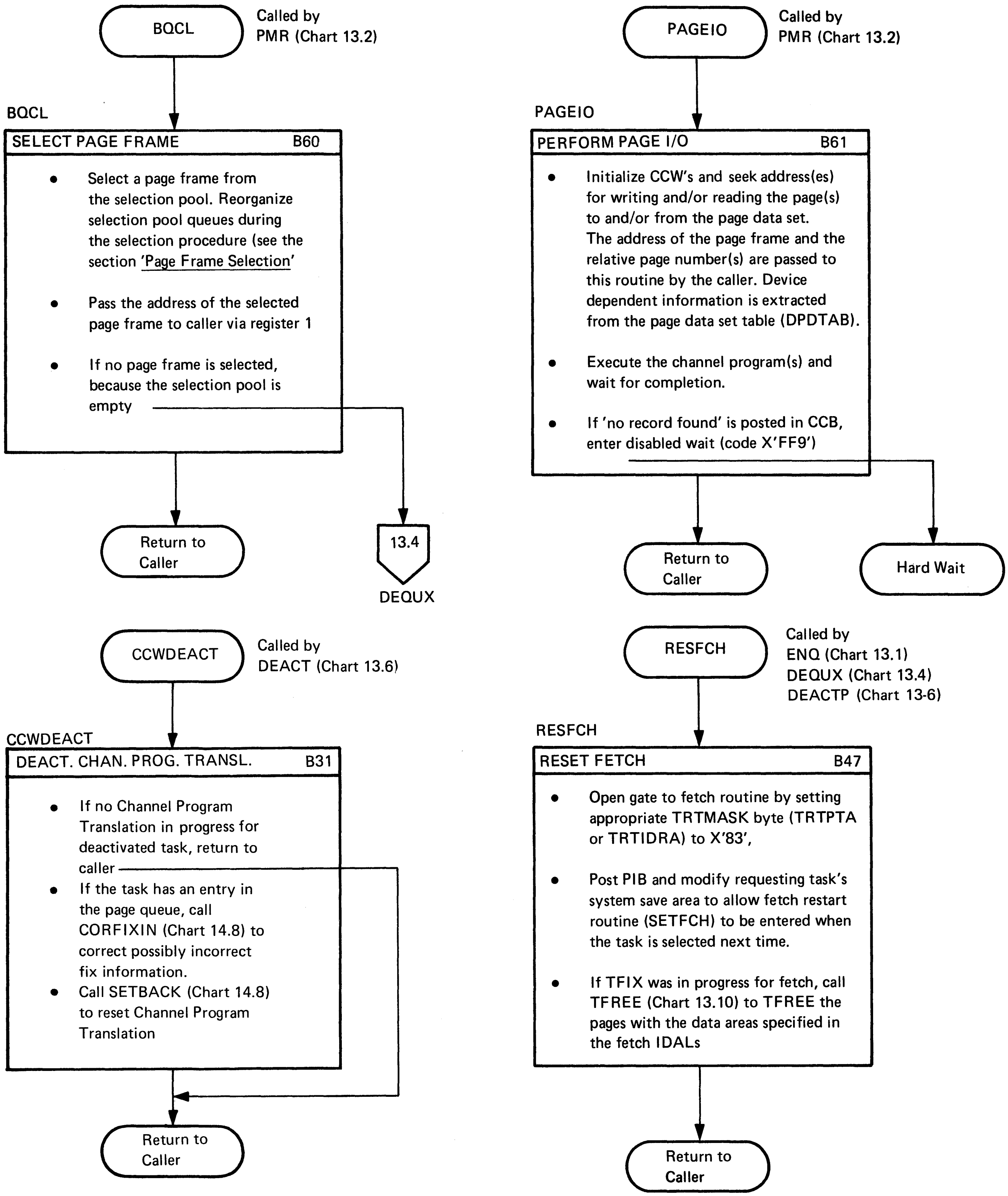

BQCL
Called by
PMR (Chart 13.2)
BQCL
SELECT PAGE FRAME B60
• Select a page frame from the selection pool. Reorganize selection pool queues during the selection procedure (see the section 'Page Frame Selection'
• Pass the address of the selected page frame to caller via register 1
• If no page frame is selected, because the selection pool is empty
Return to Caller
13.4
DEQUX
PAGEIO
Called by
PMR (Chart 13.2)
PAGEIO
PERFORM PAGE I/O B61
• Initialize CCW's and seek address(es) for writing and/or reading the page(s) to and/or from the page data set. The address of the page frame and the relative page number(s) are passed to this routine by the caller. Device dependent information is extracted from the page data set table (DPDTAB).
• Execute the channel program(s) and wait for completion.
• If 'no record found' is posted in CCB, enter disabled wait (code X'FF9')
Return to Caller
Hard Wait
CCWDEACT
Called by
DEACT (Chart 13.6)
CCWDEACT
DEACT. CHAN. PROG. TRANSL. B31
• If no Channel Program Translation in progress for deactivated task, return to caller
• If the task has an entry in the page queue, call CORFIXIN (Chart 14.8) to correct possibly incorrect fix information.
• Call SETBACK (Chart 14.8) to reset Channel Program Translation
Return to Caller
RESFCH
Called by
ENQ (Chart 13.1)
DEQUX (Chart 13.4)
DEACTP (Chart 13-6)
RESFCH
RESET FETCH B47
• Open gate to fetch routine by setting appropriate TRTMASK byte (TRTPTA or TRTIDRA) to X'83',
• Post PIB and modify requesting task's system save area to allow fetch restart routine (SETFCH) to be entered when the task is selected next time.
• If TFIX was in progress for fetch, call TFREE (Chart 13.10) to TFREE the pages with the data areas specified in the fetch IDALs
Return to Caller

Chart 13.6. **Page Management,** Deactivation of Partition

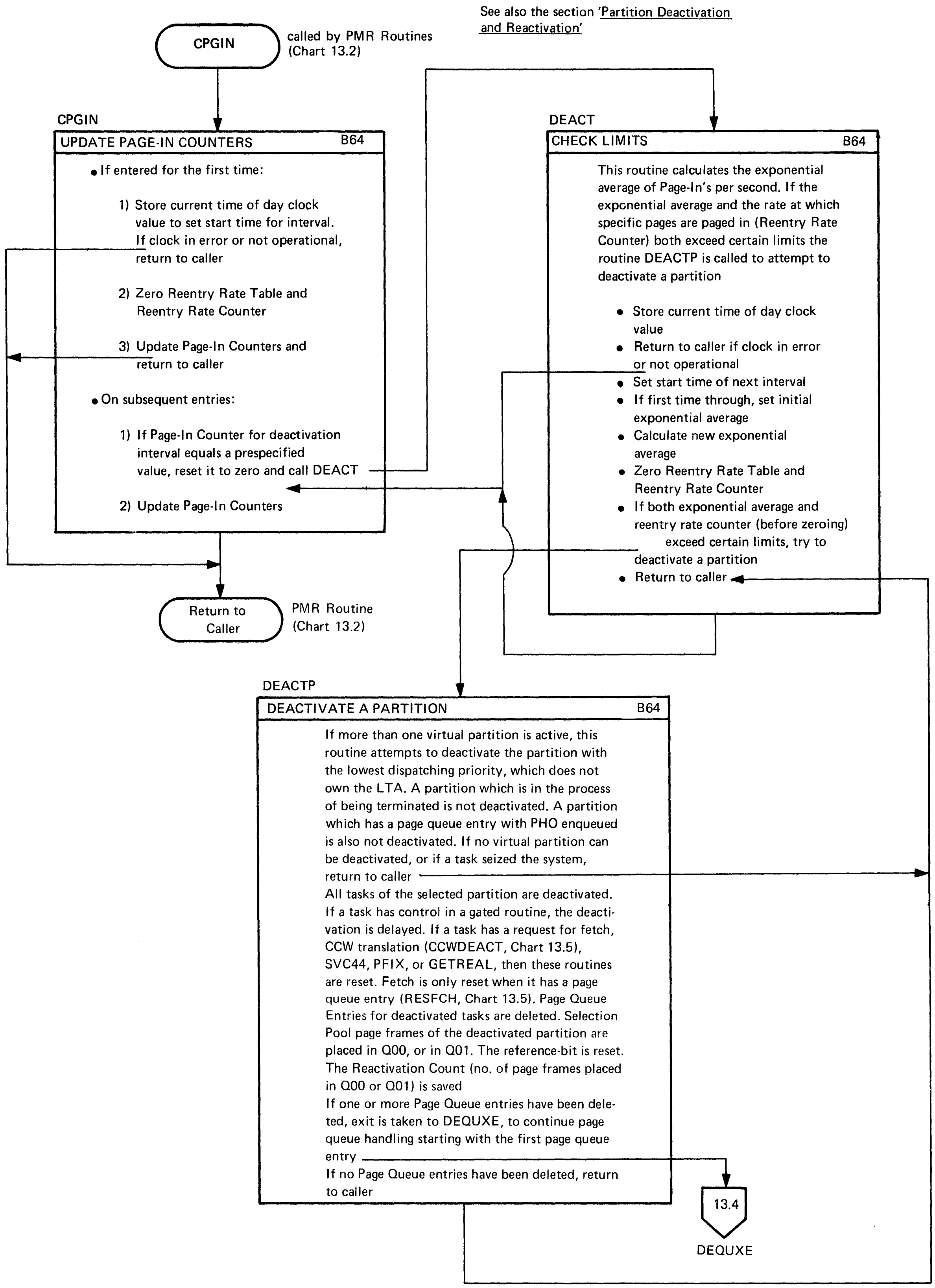
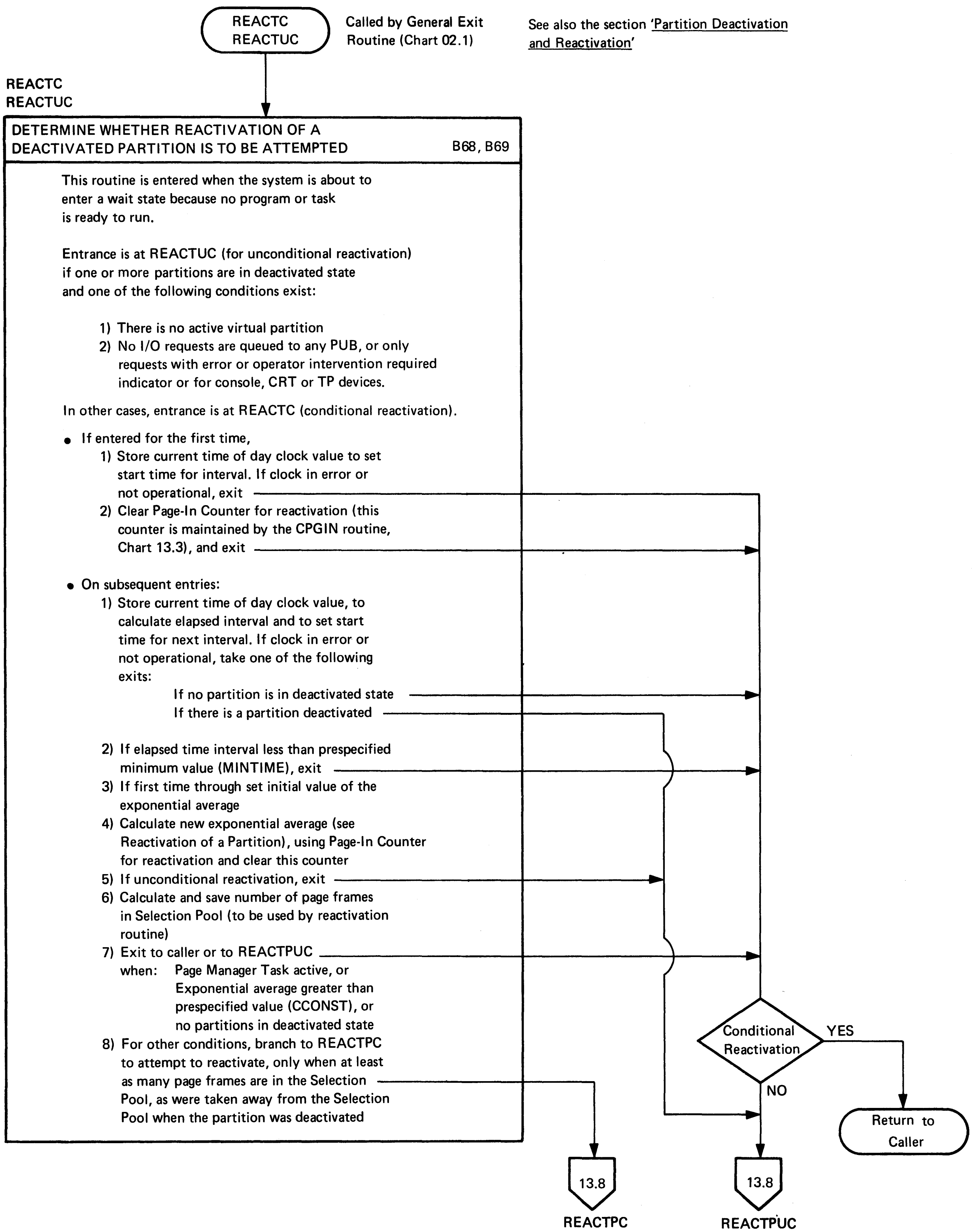
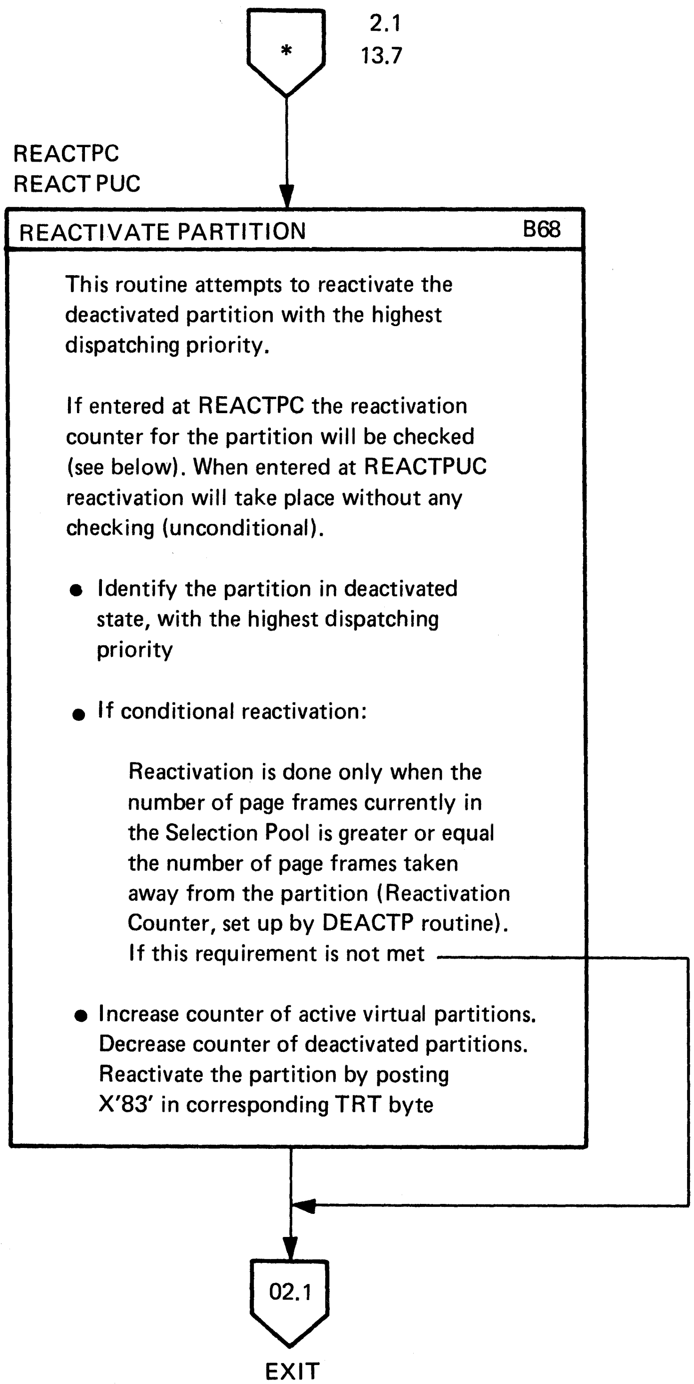


Chart 13.7. Page Management, Reactivation of Partition




*
2.1
13.7
REACTPC
REACTPUC
REACTIVATE PARTITION
B68
This routine attempts to reactivate the deactivated partition with the highest dispatching priority.
If entered at REACTPC the reactivation counter for the partition will be checked (see below). When entered at REACTPUC reactivation will take place without any checking (unconditional).
● Identify the partition in deactivated state, with the highest dispatching priority
● If conditional reactivation:
Reactivation is done only when the number of page frames currently in the Selection Pool is greater or equal the number of page frames taken away from the partition (Reactivation Counter, set up by DEACTP routine). If this requirement is not met
● Increase counter of active virtual partitions. Decrease counter of deactivated partitions. Reactivate the partition by posting X'83' in corresponding TRT byte
02.1
EXIT

See also the section 'TFIX Request'

TFIX

Called by
FIXAREA (Chart 14.4)
SVC44
Fetch (Chart 08.1)
SDAID

TFIX

TFIX A PAGE — B44

In register 2 a virtual address is passed. The page which contains this address is to be temporarity fixed.

- If the page is in real storage call DTFIX (Chart 13.9) to TFIX the page.
  1) If on return, the page is fixed, return to caller
  2) If the page could not be TFIXed by DTFIX, take the action described below.
- If the page is not in real storage, or DTFIX could not TFIX the page, set up address of area to be TFIXed and TFIX identifier in TRADDR (location X'90'). This information is used by the ENQU routine to build a page queue entry.
- If the requestor is a system task, set up 'old PSW' and call the general entry routine GENENT (Chart 01.1) to save task's status information. When the task is selected again, processing will continue with the next sequential instruction after the TFIX request.
- For other requestors, set up 'PSW', save 'PSW' and registers in the task's system save area and indicate in task's PIB that this procedure was followed (PIB tested at task selection) When the task is selected again, processing will continue with the next sequential instruction after the TFIX request

→ 13.1 ENQU

Return to Caller

DTFIX

Called by
PMR (Chart 13.2)
TFIX (Chart 13.9)

DTFIX

TFIX A PAGE WHICH IS IN REAL STORAGE — B57

DTFIX is called when the page to be TFIXed is in real storage.

- If the page is already fixed:
  1) If the TFIX counter has readed the maximum value (255), enter a one-instruction loop.
  2) If not, add one to the TFIX counter and return to caller → Return to Caller
- If the page is not yet fixed, the following two conditions make it impossible to TFIX the page in the page frame which currently is containing it:
  1) PFIX request pending for the page (NFF, bit 12 in PFT entry on)
  2) PFIX or GETREAL request pending for the page frame (NF bit, bit 15 in PFT entry on)
- If these conditions are not present, the TFIX counter is set to one, the page frame is deleted from the selection pool, and return is made to caller → Return to Caller
- If the first condition exists (NFF bit on), the TFIX attempt is indicated as not successful by returning to the address in the return register plus four.
- If the second condition holds (NF bit on), the following action is taken:
  1) A search is made for a page frame in which the page could be TFIXed, and with which the contents could be exchanged. First Q00 is searched for a PFT entry with the NF bit off. If such an entry could not be found in Q00, or if one is found but the corresponding page frame is in use, the entire PFT is searched for an entry with bits 12-15 zero (in selection pool, not failing, and above two conditions not posted)
  2) If such a page frame is found, call EXCH (Chart 13.13) to perform the exchange. The newly found page frame is removed from the selection pool and its TFIX counter is set to one. Return to caller → Return to Caller
  3) If such a page frame could not be found indicate unsuccessful TFIX by returning to address in return register plus four.

Return to Caller

TFREE

Called by
CCW Translation (Chart 14.2)
CCW Retranslation (Chart 14.3)
SVC44
Fetch (Chart 08.1)
SDAID
RESFCH (Chart 13.5)
CORFIXIN (Chart 14.8)

TFREE

TFREE A PAGE — B44

On entry to this routine, register 3 contains the number of the page frame containing the page to be TFREEd. To decrement the TFIX counter, call DTFREE (Chart 13.10).

- If on return from DTFREE the page is still fixed (TFIX counter ≠ 0 and/or PFIX counter ≠ 0), return to caller.
- If both the TFIX and PFIX counters contain zero, additional action is taken to service any tasks waiting for freed page frames, for this page frame, or for this page:
  1) Allow tasks waiting for freed page frames to be selected by moving X'83' into the TRTFIX and TRTPFG bytes of the resource table TRTMASK.
  2) If a task is waiting for this page frame (NF, bit 15 in PFT entry on), make the task selectable. If this waiting task is in virtual mode (therefore this concerns a PFIX request), reset the NF bit in all other PFT entries for the corresponding real partition.
  3) If a task (PFIX request) is waiting for this page (NFF, bit 12 in PFT entry is on), get address of PFT entry of reserved page frame from FIXWTAB. Call EXCHANGE (Chart13.13) to exchange contents of page frames and PFIX the page in the reserved page frame. Post waiting task ready to run, unless exchange and PFIX have been delayed by the EXCHANGE routine.

Return to Caller

DTFREE

Called by
TFREE (Chart 13.10

DTFREE

DECREASE TFIX COUNTER — B62

- Reduce TFIX counter by one
- If page is still TFIXed or PFIXed, return to caller
- If page is no longer fixed, return the page frame to the selection pool
  1) If the request is from fetch, enqueue the PFT entry to the top of Q00 or Q01, depending on the status of the change bit. Reset the reference bit.
  2) If the request is not from fetch append PFT entry to the end of the Hold Queue
- Indicate page no longer fixed by returning to address in return register plus four.

Return to Caller

See also the section '<u>FREEREAL TFREE and PFREE Requests</u>'

Chart 13.11. Page Management, PFIX Routines

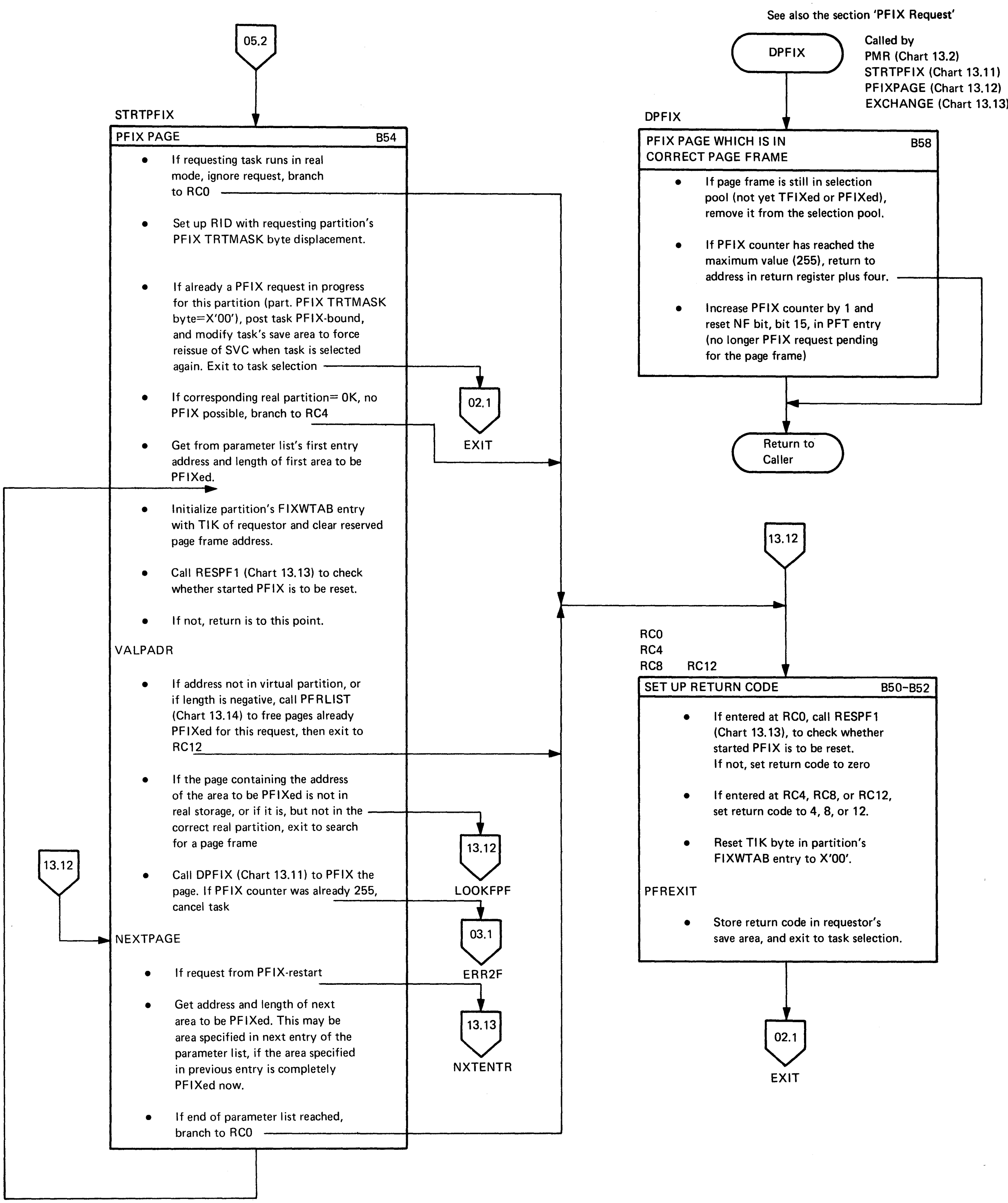


Chart 13.12. Page Management, PFIX, Find a Page Frame

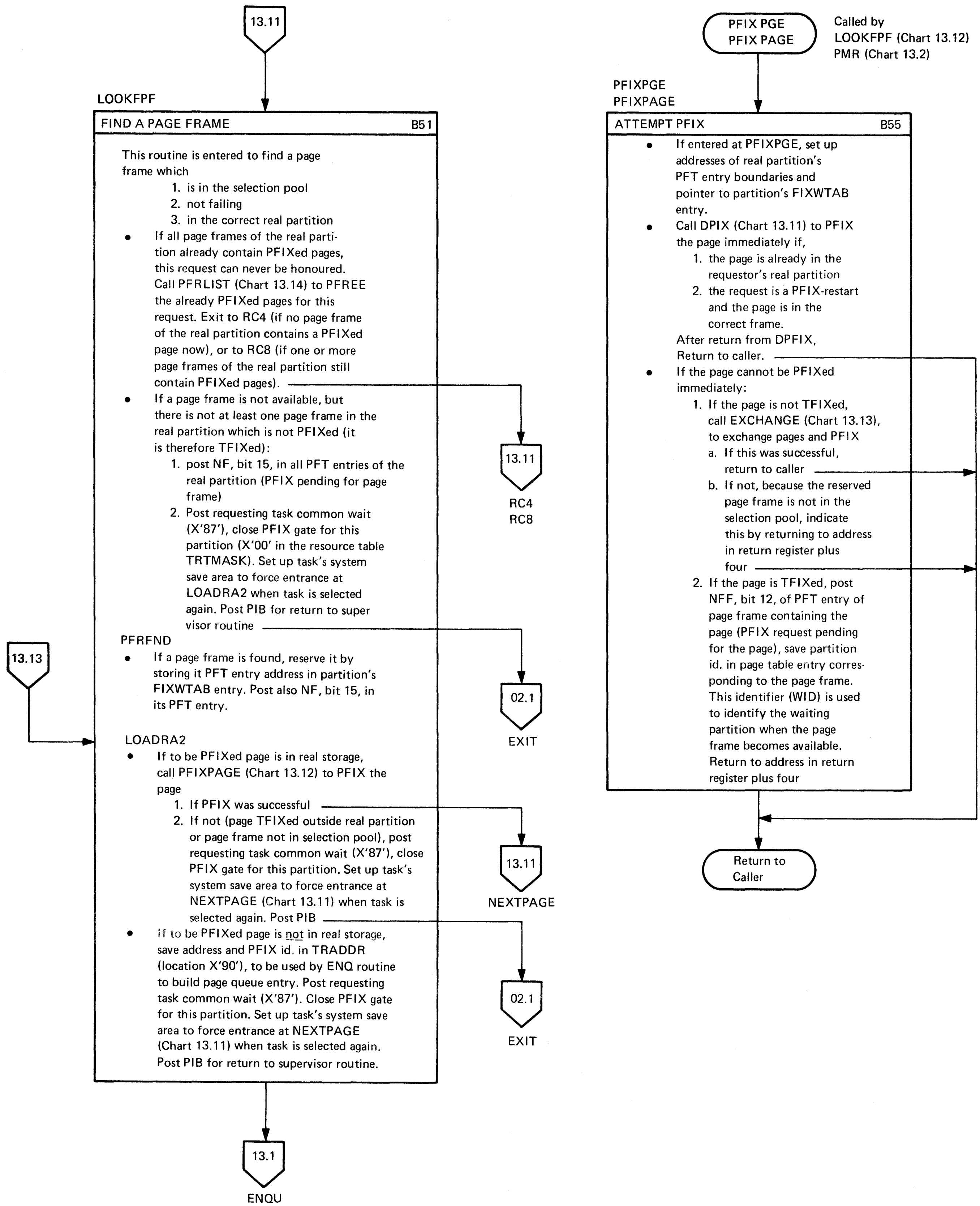
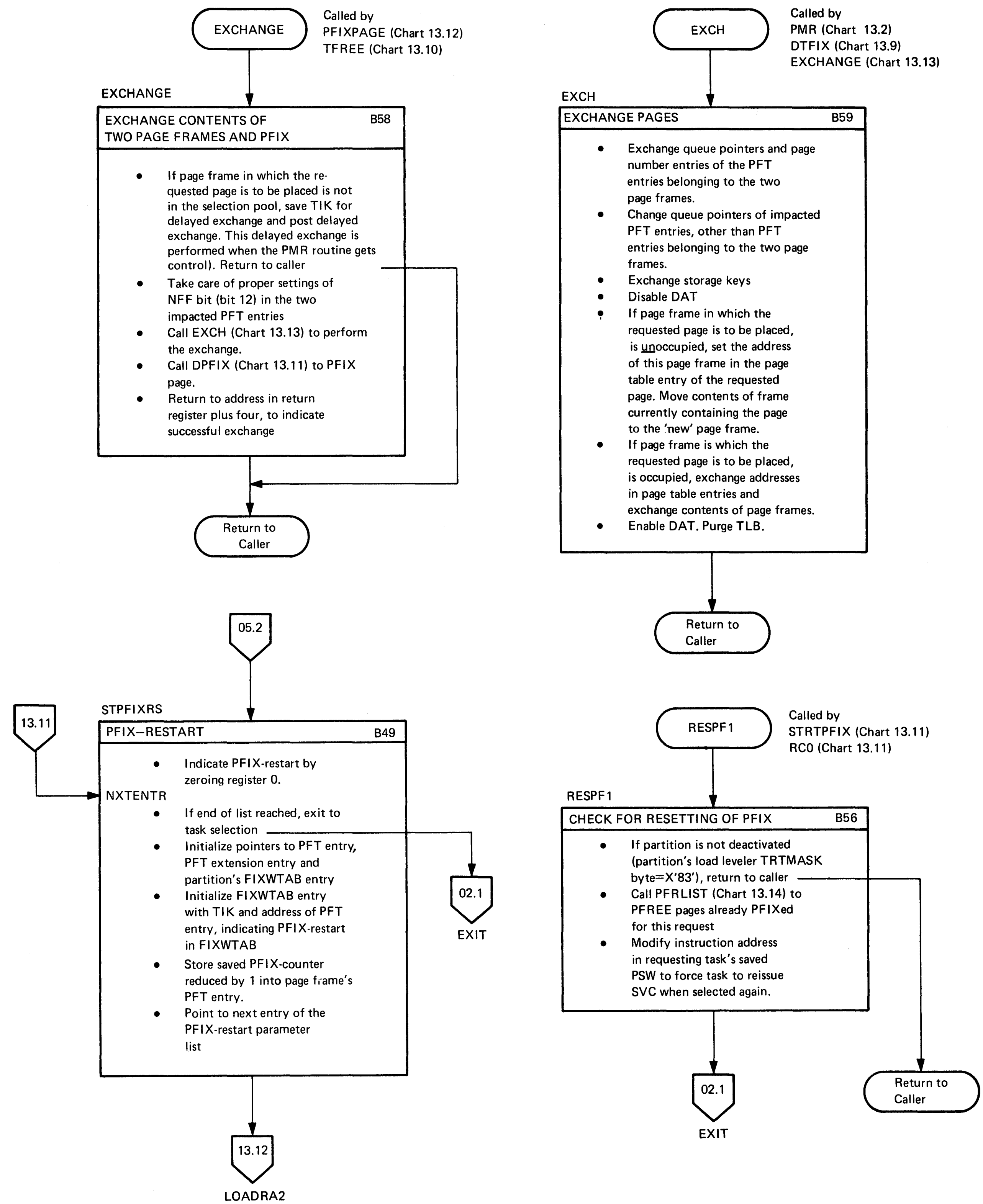
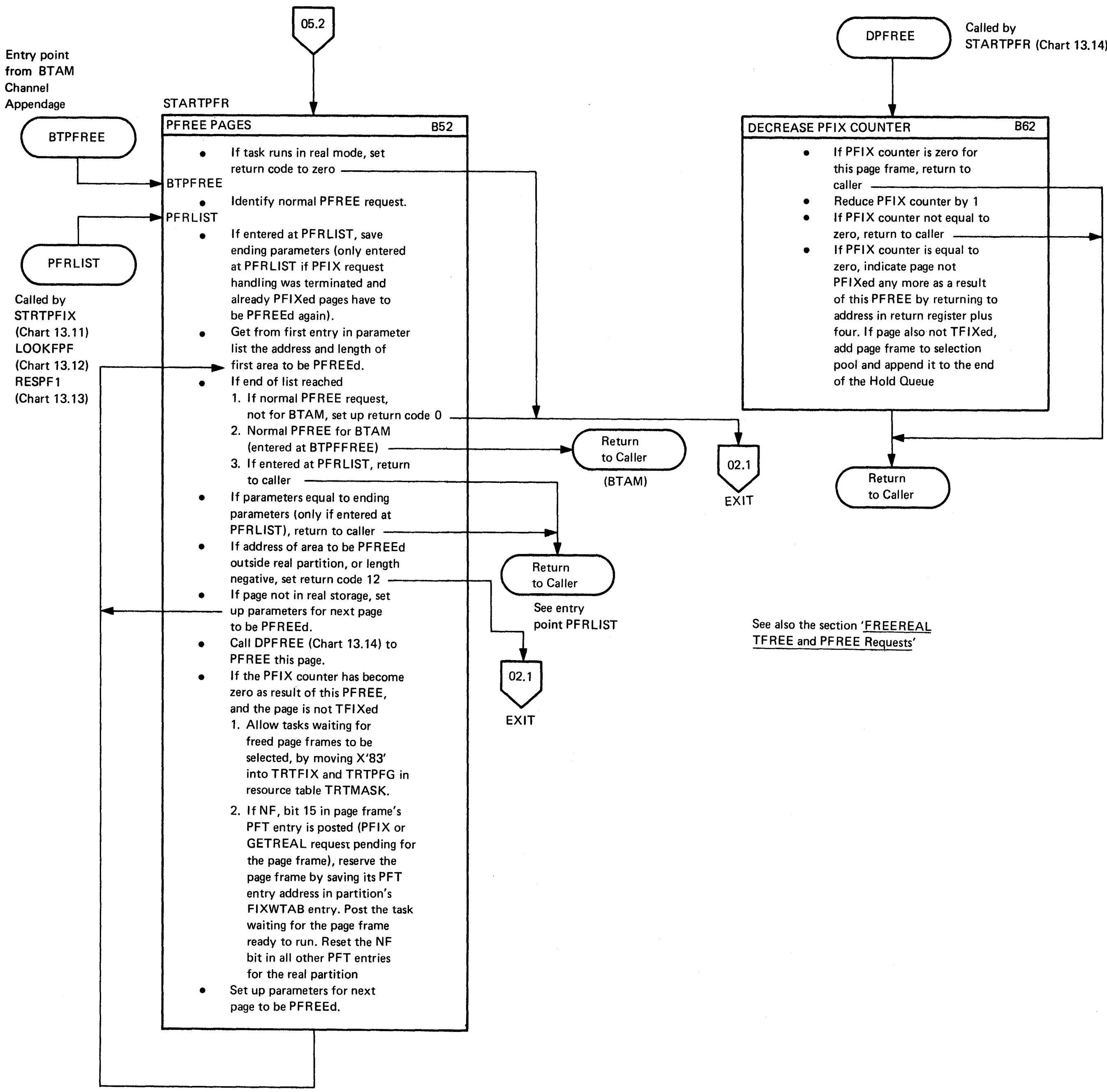


Chart 13.13. Page Management, Exchange Pages, PFIX-Restart





05.2
Entry point from BTAM Channel Appendage
BTPFREE
STARTPFR
PFREE PAGES B52
• If task runs in real mode, set return code to zero
BTPFREE
• Identify normal PFREE request.
PFRLIST
• If entered at PFRLIST, save ending parameters (only entered at PFRLIST if PFIX request handling was terminated and already PFIXed pages have to be PFREEd again).
• Get from first entry in parameter list the address and length of first area to be PFREEd.
• If end of list reached
1. If normal PFREE request, not for BTAM, set up return code 0
2. Normal PFREE for BTAM (entered at BTPFFREE)
3. If entered at PFRLIST, return to caller
• If parameters equal to ending parameters (only if entered at PFRLIST), return to caller
• If address of area to be PFREEd outside real partition, or length negative, set return code 12
• If page not in real storage, set up parameters for next page to be PFREEd.
• Call DPFREE (Chart 13.14) to PFREE this page.
• If the PFIX counter has become zero as result of this PFREE, and the page is not TFIXed
1. Allow tasks waiting for freed page frames to be selected, by moving X'83' into TRTFIX and TRTPFG in resource table TRTMASK.
2. If NF, bit 15 in page frame's PFT entry is posted (PFIX or GETREAL request pending for the page frame), reserve the page frame by saving its PFT entry address in partition's FIXWTAB entry. Post the task waiting for the page frame ready to run. Reset the NF bit in all other PFT entries for the real partition
• Set up parameters for next page to be PFREEd.
PFRLIST
Called by STRTPFIX (Chart 13.11) LOOKFPF (Chart 13.12) RESPF1 (Chart 13.13)
Return to Caller
(BTAM)
Return to Caller
See entry point PFRLIST
02.1
EXIT
02.1
EXIT
DPFREE
Called by STARTPFR (Chart 13.14)
DECREASE PFIX COUNTER B62
• If PFIX counter is zero for this page frame, return to caller
• Reduce PFIX counter by 1
• If PFIX counter not equal to zero, return to caller
• If PFIX counter is equal to zero, indicate page not PFIXed any more as a result of this PFREE by returning to address in return register plus four. If page also not TFIXed, add page frame to selection pool and append it to the end of the Hold Queue
Return to Caller
See also the section 'FREEREAL TFREE and PFREE Requests'

Chart 13.15. Page Management, PAGEIN Routine

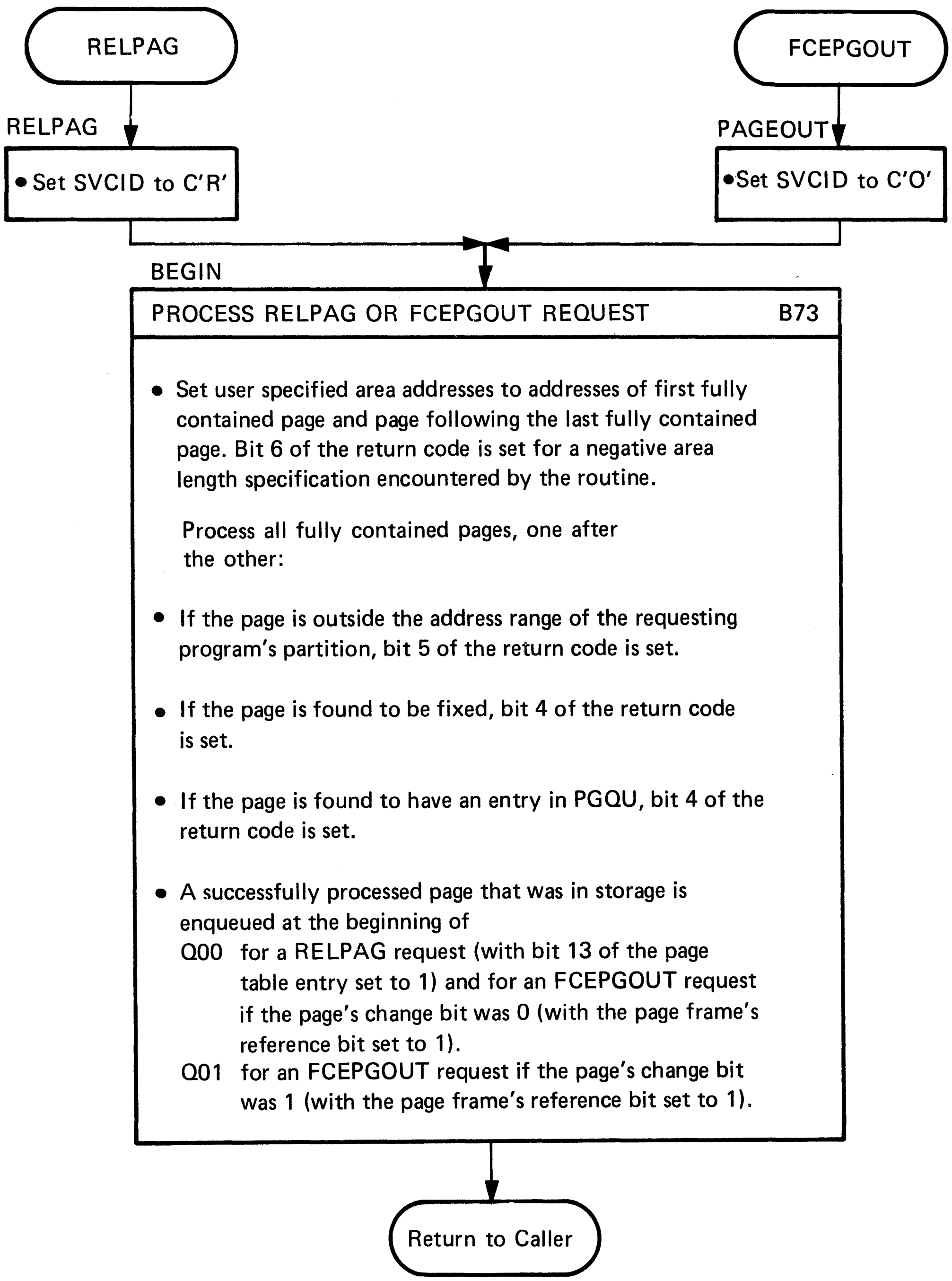


Chart 13.16. Page Management, RELPAG and FCEPGOUT Routines

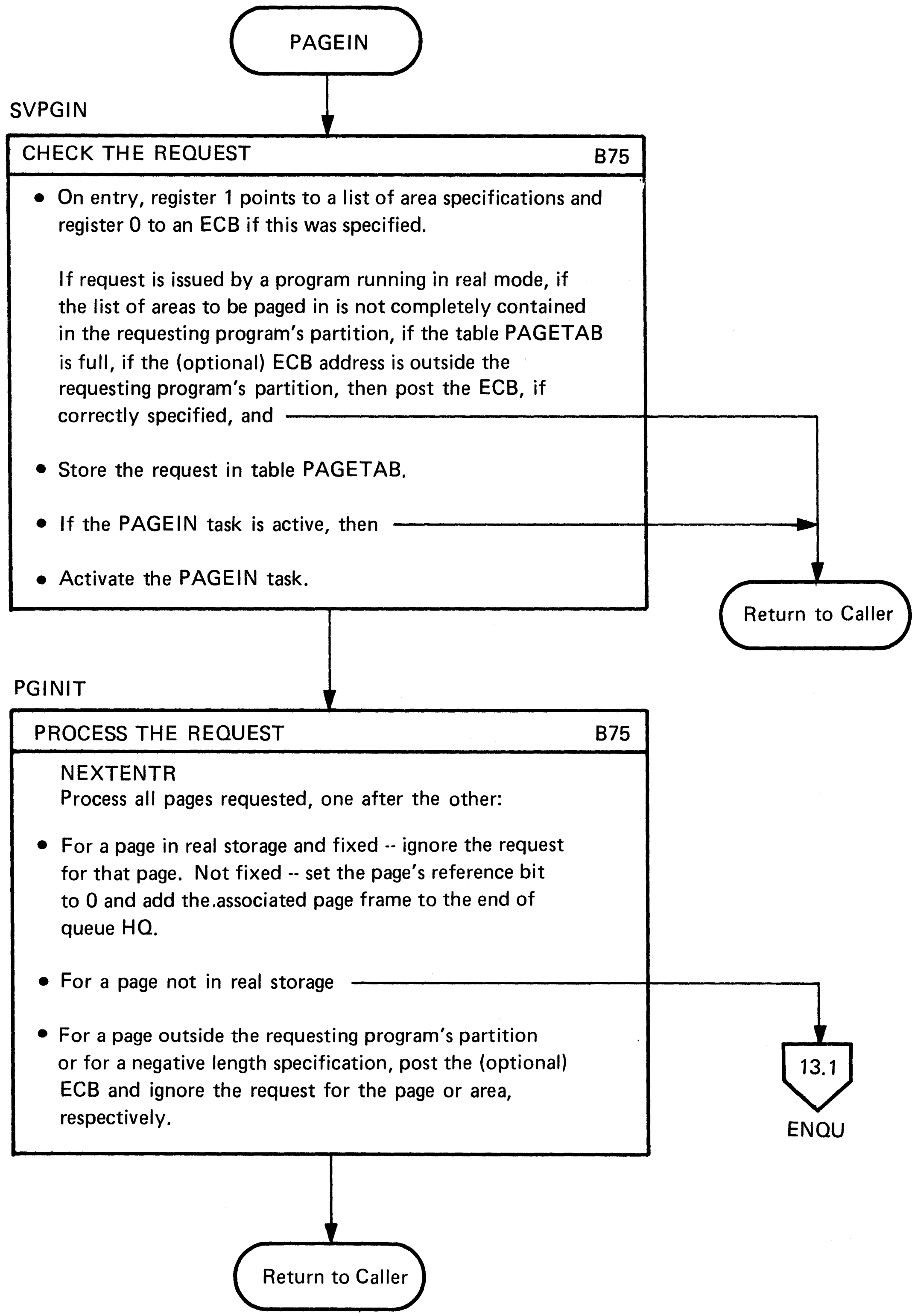
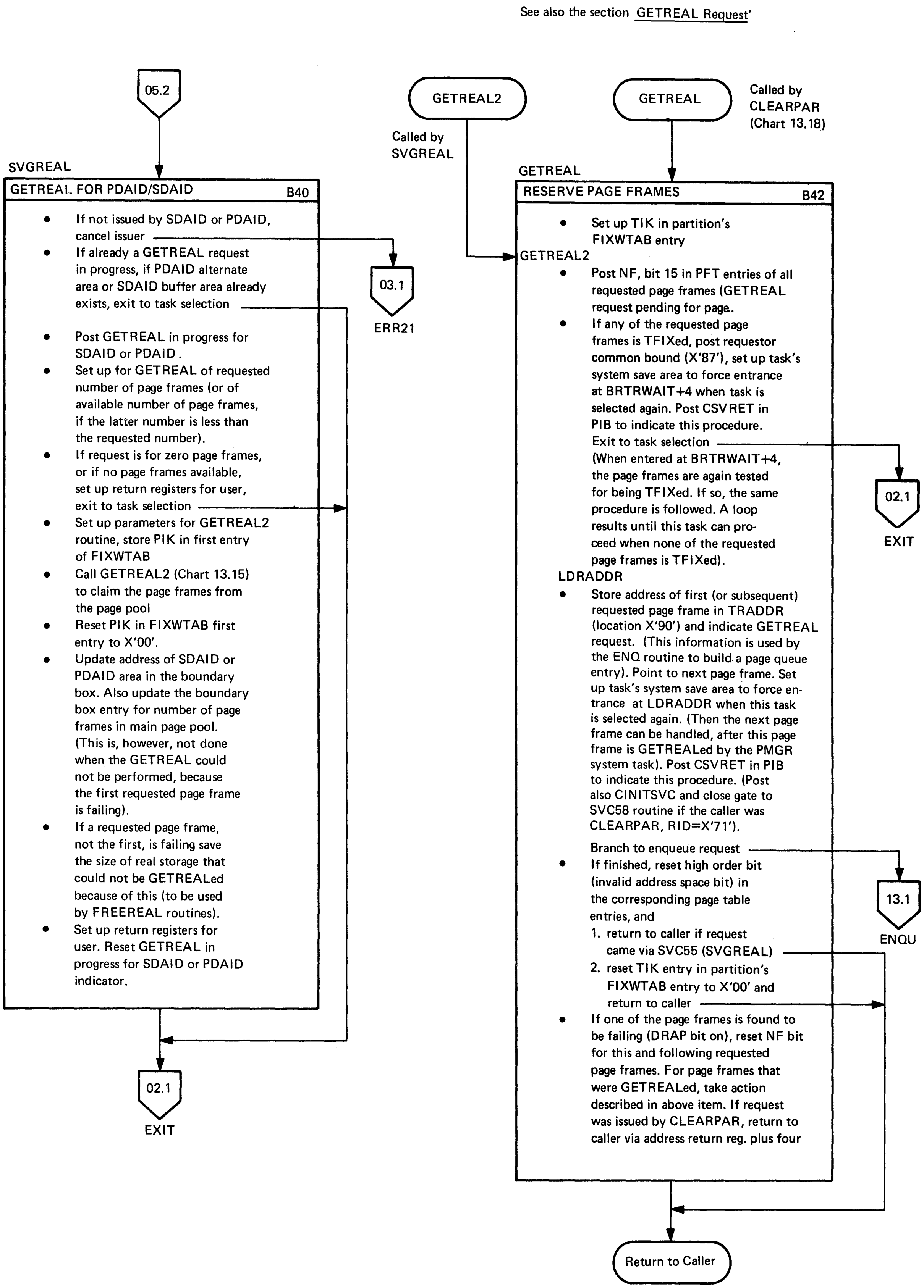


Chart 13.17. Page Management, GETREAL Routines



05.2

CLEARPAR

INITIALIZE PARTITION B48

- If key of issuing task is not zero, cancel task. → 03.1 ERR21
- If next job is to execute in virtual mode:
  1. Fill in entry in partition communications region for uppermost byte available to problem program.
  2. Set up PSW instruction address in save area.
  3. Call INVPAGE1 (Chart 13.16) to initialize page table and page frame table entries except partition's first page table entry.
  4. Clear first page of partition after save area.
  5. Exit to task selection → 02.1 EXIT
- If next job is to execute in real mode:
  1. Post RID=X'71'
  2. If gate to SVC58 routine closed (SVC58 TRTMASK byte ≠ X'83'), post task SVC58 resource bound (X'71' in PIB). Modify PSW address in save area to force task to reissue SVC when selected next time → 02.1 EXIT
  3. Set address of real save area in PIB.
  4. More virtual save area to work-area RESA2 and perform actions described in 1., 2. and 3. for virtual mode. (The copy of the save area is updated and all page table entries for the partition are initialized).
  5. Post real mode in PIB. Reduce entry in SYSCOM for number of active virtual partitions by one.
  6. Call GETREAL (Chart 13.15) to claim required page frames.
  7. If GETREAL successful, move copied save area to patition and exit to task selection → 02.1 EXIT
  8. If a page frame is found to be failing (DRAP bit on), the job is cancelled after the following action is taken → 03.1 ERR2D
     a. If not first page frame failing, move save area to real partition change uppermost byte available entry in SYSCOM
     b. If first page frame failing, call INVPAGE1 (Chart 13.16) to initialize first page of partition. Post virtual mode in PIB. Move copied save area to virtual patition. Update uppermost byte available entry in SYSCOM

INVPAGE1 — Called by CLEARPAR (Chart 13.18)

05.2

INVPAGE

INITIALIZE PAGE TABLE ENTRIES B49

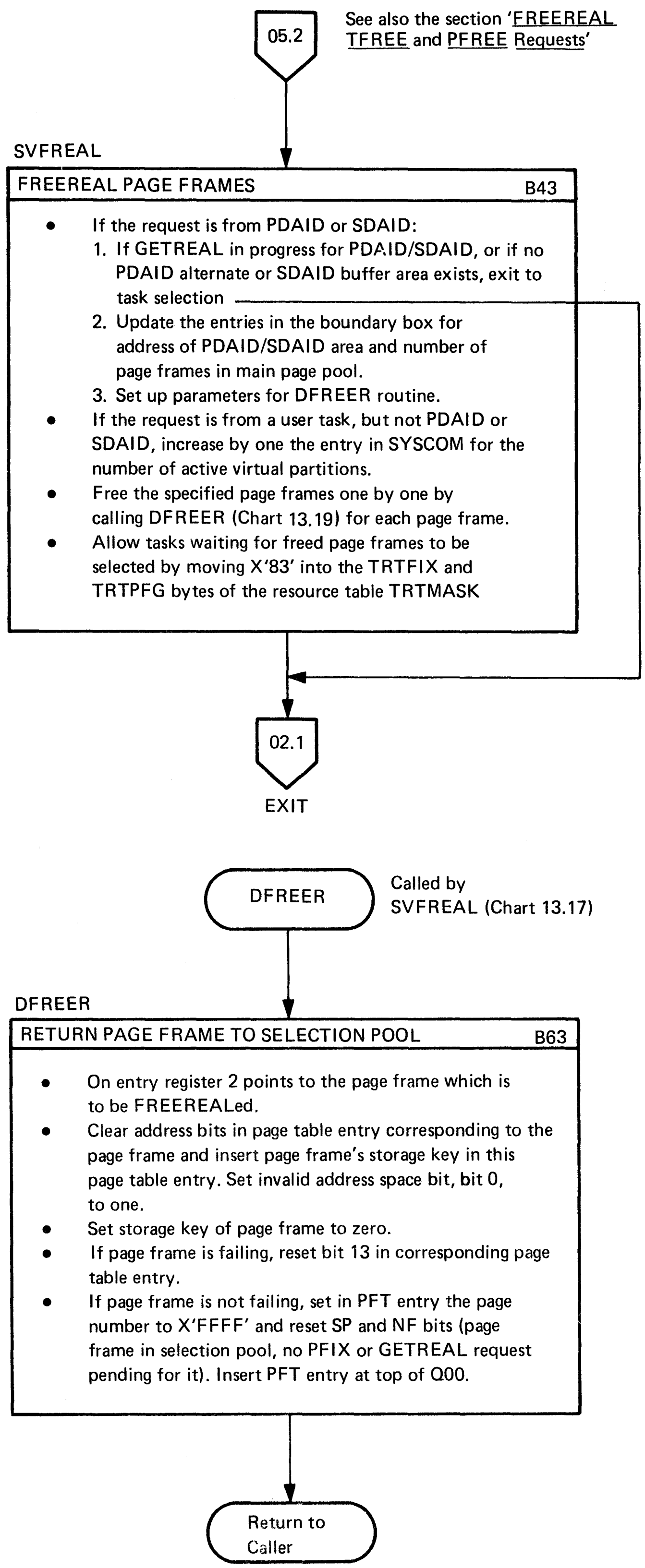
- If key of issuing task is no zero, cancel task → 03.1 ERR21
- Load parameters from user save area to registers 3-5.

INVPAGE1

R3 points to the first and R4 to the last page table entry to be initialized R5 contains in its two low order bytes the contents of an initialized page table entry

- Each page table entry is initialized with the contents if the two low order bytes of R5. Purge TLB
- If, before initializing a page table entry, it is found that the page is in real storage (while the invalid address space bit, bit 0, is not on):
  1. set storage key of page frame containing the page to X'00'.
  2. Set PFT entry of page frame to unused and add it to the selection pool.
  3. Remove PFT entry from its queue (if it is on a queue) and add it to the top of Q00. Zero PFIX counter
- If entered from CLEARPAR (SVC58), return to caller → Return to Caller
- If entered from from SVC59, exit to task selection. → 02.1 EXIT

Chart 13.19. Page Management, FREEREAL Routines



See also the section *'Channel Program Translation'*

CCWTRANS
CCWTRBT2

Called by
Channel Scheduler (Chart 06.3)
I/O Interrupt Handler (Chart 06.1)

CCWTRANS
CCWTRBT2

CHANNEL PROGRAM TRANSLATION B14, B19

Build a real channel program for an I/O request from a program running in virtual mode. The real CCB and channel program is built in copy blocks located in the supervisor. The real channel program preserves the structure of the virtual mode channel program, with the exception that TIC commands may be inserted and that IDALS (Indirect Data Address lists) are build for I/O areas which cross page boundaries. All virtual addresses in the original program are translated to the correct real addresses in the copied program, and any pages containing data areas are TFIXed in real storage.

- If entered at CCWTRANS the Translation Control Block (CCWTCB) of the requesting task is initialized and enqueued to the queue of used Translation Control Blocks. The PIK (or TIK, if AP) and registers 2-15 of requestor are saved in the TCB.
- If entered CCWTRBT2 (for BTAM channel appendage I/O request), initialize the BTAM TCB. Requestor's PIK (or TIK, if AP) and registers 2-15 are saved in the TCB.

CCWTRB1
- Initialize in TCB the pointers to the 'status modifier command' and 'control command with data' lists for the device for which the I/O is requested (only if the device supports such commands).
- If the device is not supported, dequeue the TCB from the queue of used TCBs. Allow tasks waiting for channel program translation to be selected by moving X'83' into TRTCCW TRTMASK byte (only if no channel program translation is in progress). Exit to cancel task. → 03.1 ERR11

CCWTR00
- If BTAM channel appendage I/O request, release all associated CCW copy blocks and IDAL blocks by enqueueing than to the free copy block queue. Allow tasks waiting for copy blocks to be selected by moving X'83' into TRTBUF resource table byte. Set up number of required additional copy blocks in TCB, and pointer to first virtual CCW in copied CCB.
- If request comes from BTAM (not appendage) set up number of required additional copy blocks in TCB.

CCWREST
- If not BTAM Channel appendage I/O request, call GETBLOCK (Chart 14.3) to get a copy block for the CCB. Enqueue the block to the CCB copy block queue. Initialize the CCB copy block and copy the virtual CCB into it.

CCWTR02
- If user provided a sense CCW, copy it to the CCB copy block and call TESTIDAL (Chart 14.4) to translate data address and TFIX data area or build Indirect Data Address list (IDAL).

14.2
CCWTR1

Chart 14.2. Channel Program Translation, General Routine

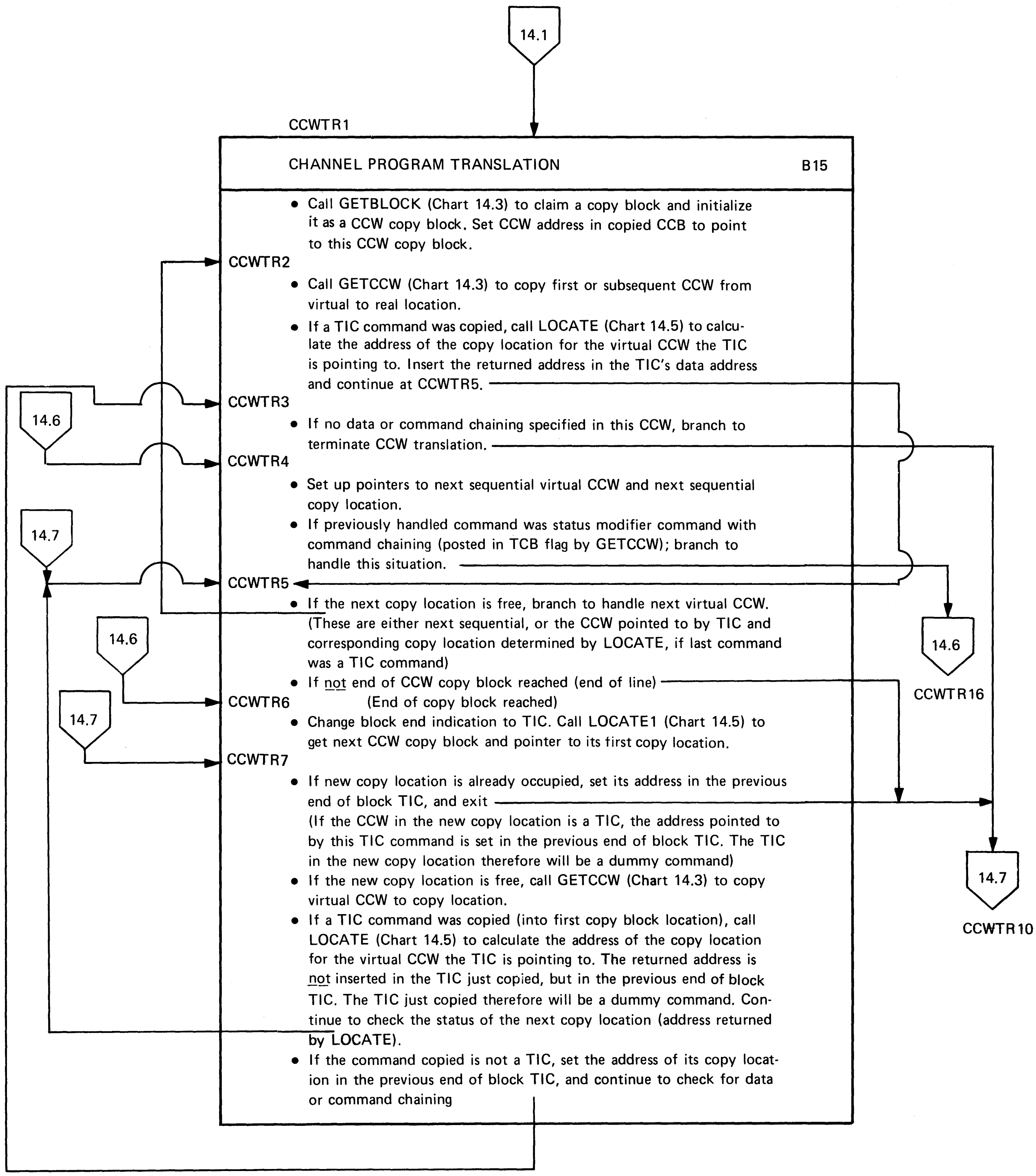


Chart 14.3. Channel Program Translation, Subroutines

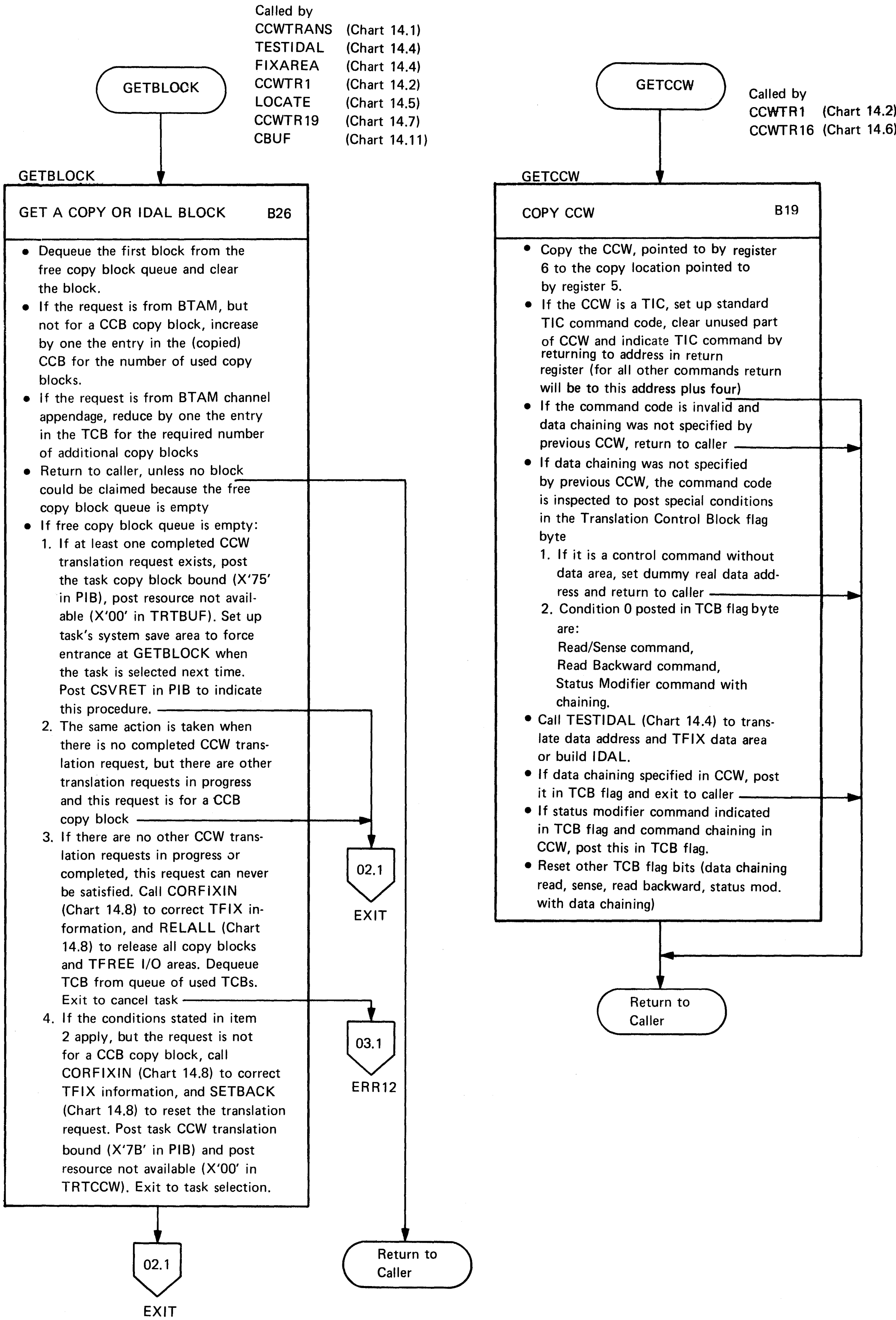


Chart 14.4. Channel Program Translation, Build IDAL, TFIX Data Area

TESTIDAL

Called by
CCWTRANS (Chart 14.1)
GETCCW (Chart 14.3)

TESTIDAL

BUILD IDAL IF REQUIRED B21

- If the CCW is a READ or SENSE with the skip bit on, reset IDAL bit in CCW and store a dummy real data address in CCW. Return to caller
- Return to caller if count in CCW is zero
- If count in CCW is not less than 32K, call RELALL (Chart 14.8) to release all copy blocks and TFREE I/O areas. Dequeue TCB from queue of used TCBs. Exit to cancel task
- If data area does not cross page boundary and user has not provided an IDAL, branch to FIXAREA to translate the data address and TFIX the data area.
- If IDAL is to be created or copied:
  1. If no IDAL block is available, or if there are not enough Indirect Data Address Word locations available in the current IDAL block, call GETBLOCK (Chart 14.3) to claim a new block and enqueue the block to the IDAL block queue.
  2. If the IDAL is supplied by the user, copy it to the appropriate IDAW locations in the current IDAL block. Return to caller
  3. If IDAL not supplied by user, build one. The virtual addresses of the begin of the data area and the page boundary crossings are placed in the appropriate IDAW locations in the current IDAL block. (Page boundary crossings minus one in case of read backward command). The address of the IDAL is set in the CCW and the IDA bit in the CCW is posted. Return to caller

03.1
ERR13

FIXAREA

Called by
CCWTR19
(Chart 14.7)

FIXAREA

TFIX DATA AREA B23

On entry register 5 points to a CCW or Indirect Data Address Word containing the (virtual) address of the area to be TFIXed. Before returning to caller this address is replaced by the corresponding real address.

- If the page containing the passed address is in real storage:
  1. If the real address is equal to the virtual address, or if the request is from BTAM, or if this page is already TFIXed for this I/O request, return to caller
  2. Otherwise, call TFIX (chart 13.9) to TFIX the page and indicate that this page is TFIXed for this I/O request.

  The same action as described in the above item 2 is taken when the page is not in real storage.

Note:

- For a number of error conditions, such as real address beyond real storage or not in correct partition, a dummy real address is used and TFIX is not called.
- A bit string in the CCB copy block (CCBXINF) is used to indicate whether pages are TFIXed for current I/O request or not. (One to one correspondence between bits and page frames). It may be necessary to call GETBLOCK (Chart 14.3) one or more times to obtain additional Fix blocks if page frames beyond 384K are involved.

Return to Caller

Chart 14.5. Channel Program Translation, LOCATE Routine

LOCATE
LOCATE1

Called by
CCWTR1 (Chart 14.2)
CCWTR10 (Chart 14.7)

LOCATE
LOCATE1

FIND ADDRESS OF COPY LOCATION B22

The virtual address of a CCW is passed to this routine in register 6. This routine determines the correct copy location for this CCW, respective to the locations of already copied CCWs. (CCWs are copied in the order of increasing virtual addresses). If necessary, GETBLOCK (Chart 14.3) is called to claim a block to be used as a new CCW copy block. The address of the copy location is returned in register 5. See the section *'Channel Programs Containing TIC commands'* in *'Copying and Translating Channel Programs'.*
This routine is entered at LOCATE1 when it is known that the next copy location will be the first copy location of the next (to be created) CCW copy block in the queue.

- The queue of CCW blocks for this I/O request is searched for a block for which the following applies. The virtual storage location of the first copied CCW in the block (VBA) is below the virtual storage location passed to this routine (VA). Also the offset of the passed address (VA) from VBA is less than 7 CCWs (VA–VBA <56). If such a CCW copy block is found, the copy location in the block can be specified. The offset of the copy location from the begin of the copy block will be the same as the offset of the passed virtual address from VBA (copy location - start CCW copy block =VA – VBA). Return to caller

- If such a CCW Copy block cannot be found, a new block is to be claimed and enqueued. It is enqueued to the begin of the queue, or the end, or somewhere in the middle, depending on the passed virtual address. (The order of virtual addresses of CCWs is to be preserved). The copy location within the new CCW copy block is determined as follows:
  1. If the passed virtual address is less than 14 CCW's <u>above</u> the virtual CCW corresponding to the first copy location of the next <u>lower</u> block (if any), $VA < VBA_l + 112$, this offset minus 56 determines the copy location. In this case the new CCW Copy block may become a <u>short</u> block
  2. If the passed virtual address is less than 8 CCWs <u>below</u> the virtual CCW corresponding to the first copy location of the next <u>higher</u> block (if any), $VA \geq VBA_h - 56$, 56 minus this offset gives the appropriate copy location
  3. If the above is not possible, the first copy location in the new CCW copy block is chosen.

Return to
Caller

Chart 14.6. Channel Program Translation, Status Modifier Command

14.2

CCWTR16

HANDLE STATUS MODIFIER COMMAND B18

This routine is entered when the command previously copied is a status modifier command. If the next CCW copied is a TIC, the copy location of this TIC is enqueued to the chain of TICs following status modifier commands, which will be handled later.

If however the status modifier command occupies the seventh copy location in the copy block, the ninth location is enqueued to the end of block chain, which will be handled later. The next CCW is not copied by this routine.

- If the next copy location (after the copied status modifier command) is already occupied, return to continue with the second virtual CCW and second copy location after the status modifier CCW and its copy location
- If the status modifier command was copied to the last (seventh) copy location of a copy block, enqueue the ninth location to the end of block chain. Return to continue with handling the end of block condition
- If the next copy location after the copied status modifier command is free, call GETCCW (Chart 14.3) to copy next virtual CCW to next copy location.
  1. If the copied CCW is not a TIC, return to continue with the following virtual CCW and the following copy location
  2. If the copied CCW is a TIC, the copy location of this TIC is enqueued to the chain of TICs after status modifier commands. Then exit is taken as in item 1.

14.2

CCWTR6

14.2

CCWTR4

Chart 14.7. Channel Program Translation, Handle TIC and Block End Chains, and IDALs

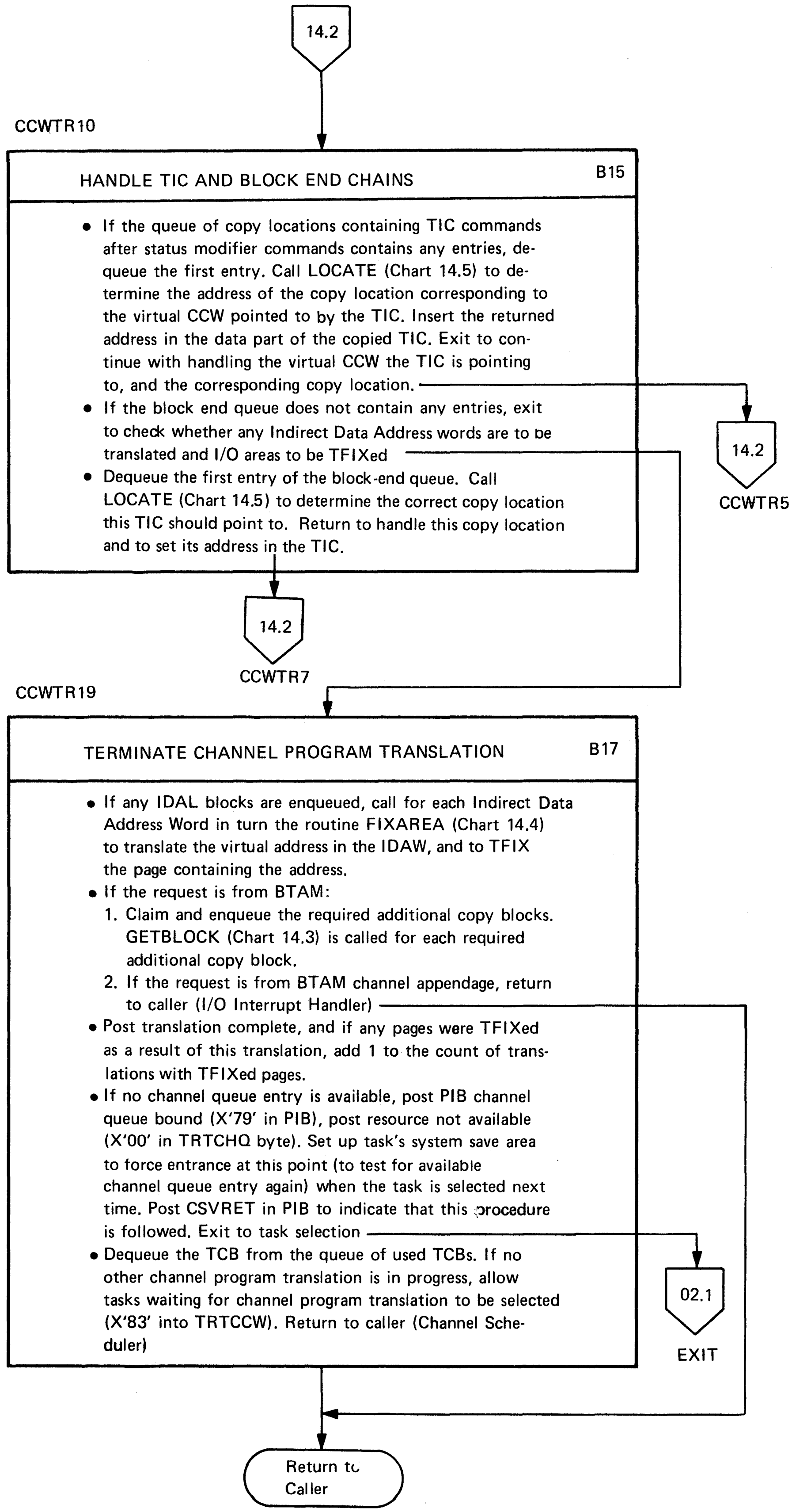


Chart 14.8. Channel Program Translation, Subroutines

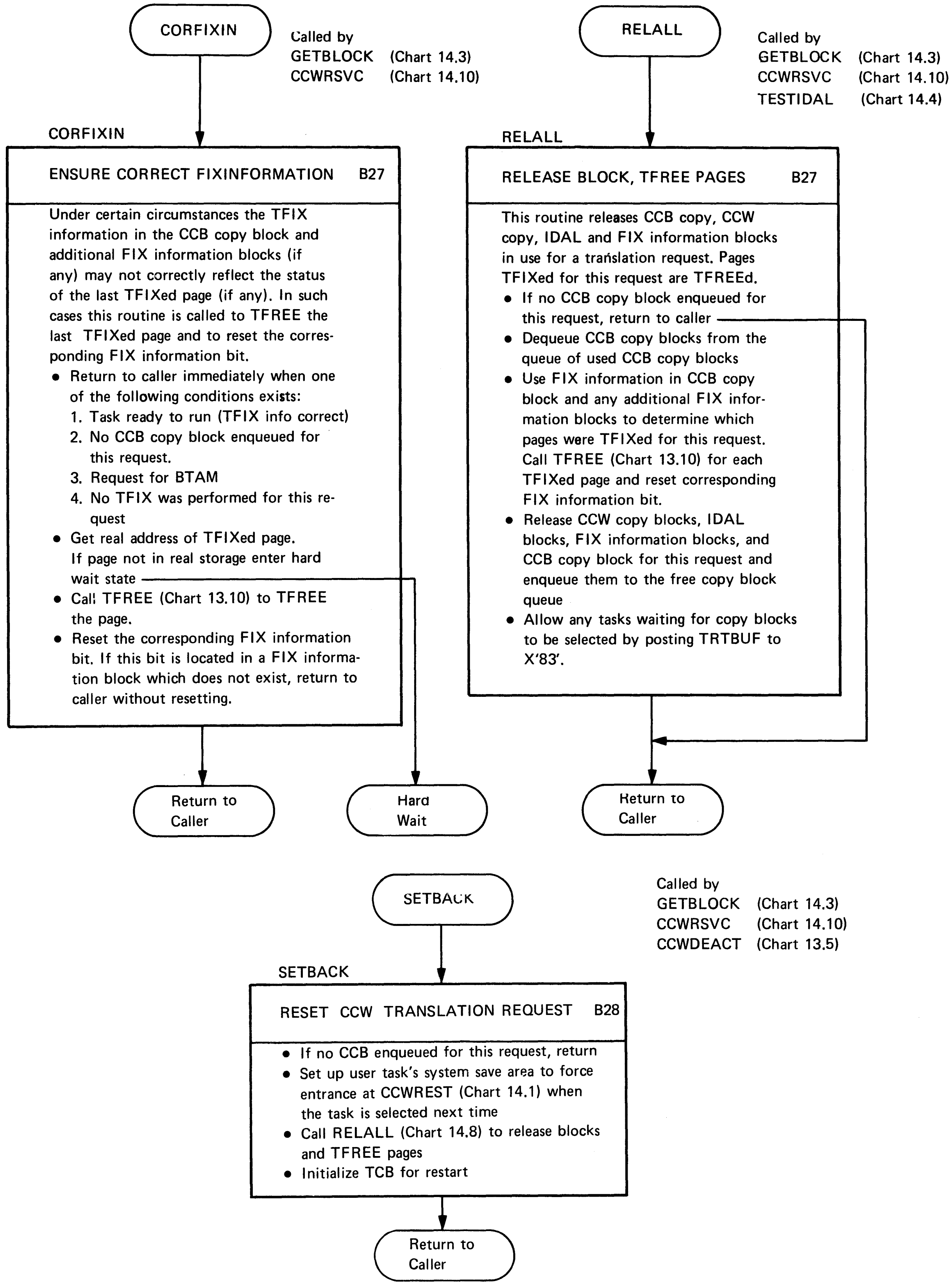


Chart 14.9. Channel Program Retranslation

See also the section *'Processing after completion of an I/O Request'*

CSWTRANS

Called by
I/O Interrupt Handler (Chart 06.2)

CSWTRANS

RETRANSLATE, COPY CCB, RELEASE BLOCKS B28

- If the copied CCB contains a (real) address in its location 13 (CCW address in CSW), and no channel appendage routine is present, calculate the virtual address corresponding to the real address and replace the real address by the virtual address.
- If any pages were TFIXed for this request, decrease the count of completed CCW translations with TFIXed pages by 1. Use FIX information in CCB copy block and any additional FIX information blocks to determine which pages were TFIXed for this request. Call TFREE (Chart 13.10) for each TFIXed page and reset corresponding FIX information bit.
- Release CCW copy blocks, IDAL blocks and FIX information blocks for this request and enqueue them to the free copy block queue.
- Allow any tasks waiting for copy blocks to be selected by posting X'83' in TRTBUF.
- If virtual CCB is in real storage, and emulator ECB is not to be posted, move significant parts of the copied CCB to the virtual CCB, release the CCB copy block and enqueue it to the free copy block queue.
- If virtual CCB is not in real storage, post PIB to indicate that copying of CCB and release of CCB copy block is to be done at a later stage (when the task is selected next time, as is also the case when the emulator ECB is to be posted).
- Allow tasks waiting for channel program translation to be selected by moving X'83' into TRTCCW.
- If no page frames have become available for TFIX as result of TFREEing by this routine, and if the count of completed translation with TFIXed pages is zero, while tasks are waiting for page frames, an attempt is made to make page frames available by resetting a task. The routine CCWRSVC (Chart 14.10) is called for this function
- Return to caller

Return to Caller

CCWRSVC

Called by
DEQUX (Chart 13.4)
CSWTRANS (Chart 14.9)

CCWRSVC

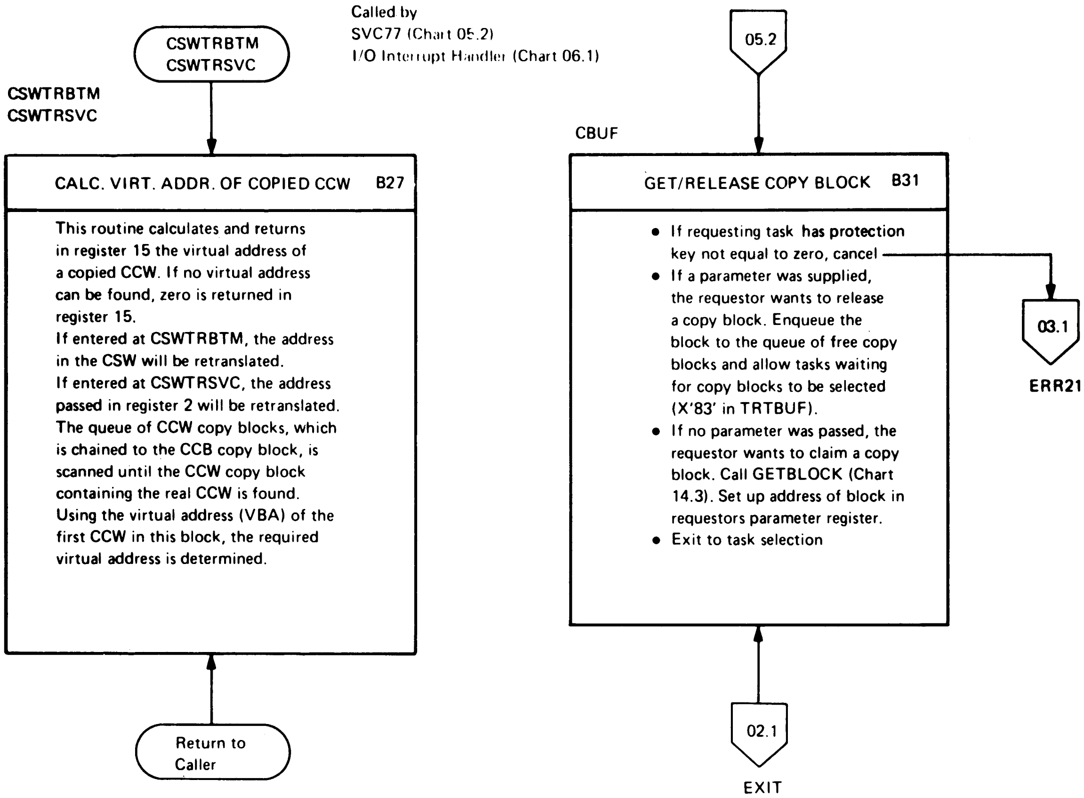
RESET OR CANCEL CCW TRANSLATION B29

This routine is called to make page frames available by resetting (or canceling) a pending CCW translation which has pages TFIXed. It is called when TRTFIX = X'00' and there are no completed CCW translations with TFIXed pages.

- Scan the queue of used TCBs for a translation in progress with TFIXed pages
- If none found, return to caller
- If the CCW translation found is the only pending CCW translation, and no pages are TFIXed by SVC44 or SDAID:
  1. delete task's entry in page queue (if any)
  2. call CORFIXIN (Chart 14.8) to correct the FIX information of the translation request.
  3. call RELALL (Chart 14.8) to release all CCB copy, CCW copy, IDAL and FIX information blocks of the request and to TFREE any pages TFIXed for the request.
  4. Modify task's system save area to force task to be canceled (page pool too small) when selected next time. Post task ready to run.
  5. Return to caller
- If the CCW translation found is not the only pending CCW translation, or if it is, but there are also pages TFIXed by SVC44 or SDAID, the task is not canceled:
  1. delete task's entry in page queue (if any)
  2. call CORFIXIN (Chart 14.8) to correct the FIX information of the translation request.
  3. call SETBACK (Chart 14.8) to reset the CCW translation
  4. post task waiting for freeing of page frames (X'6D' in PIB), and post resource not available (X'00' in TRTPFG)
  5. if page frames have become available, as a result of resetting this CCW translation request (TRTFIX ≠ X'00'), return to caller
  6. if not, continue the scan of the queue of used TCBs

Return to Caller

Chart 14.11. Channel Program Translation, SVC 72, SVC 77



The following flowcharts are organized in Sections A, B, C, and D. Offpage connectors always refer to another chart within the same Section, unless another Section is specifically mentioned.

Interrupt	Chart
external	A46
input/output	A36
machine check	B79
program check	B32
supervisor call	A46
attention	
CRT	A65
MICR	A38
console	A46

Supervisor Call	Chart
00	A48
01	A18
02	A09
03	A53
04	A35
05	A53
06	A74
07	A57
08	A57
09	A57
10	A57
11	A58
12	A59
13	A59
14	A01
15	A51
16	A59
17	A60
18	A60
19	A60
20	A60
21	A60
22	A61
23	A61
24	A61
25	A51
26	A62
27	A51
28*	A83
29	A63
30, 31, 32	QTAM
33	A63
34	A63
35	A71
36	A71

* Not entered via branch table.

Supervisor Call	Chart
37	A60
38	A75
39	A12
40	A76
41	A79
42	A79
43	Reserved
44	A64
45	A65
46	A65
47	A63
48	A65
49, 50	Reserved
51	A65
52	A62
53	Reserved
54	A65
55	A65
56, 57	Reserved
58	A66
59	A66
60	A70
61	A66
62	A66
63	A81
64	A13
65	A66
66	A66
67	A66
68	A66
69	A66
70	A67
71	A67
72	A67
73	A68
74	A68
75	B94
76	A51
77	A68
78, 79, 80, 81, 82, 83, 84	Reserved
85	A73
86	A73
87	A75
90	A67
91	A67

Chart A00. Entry Index

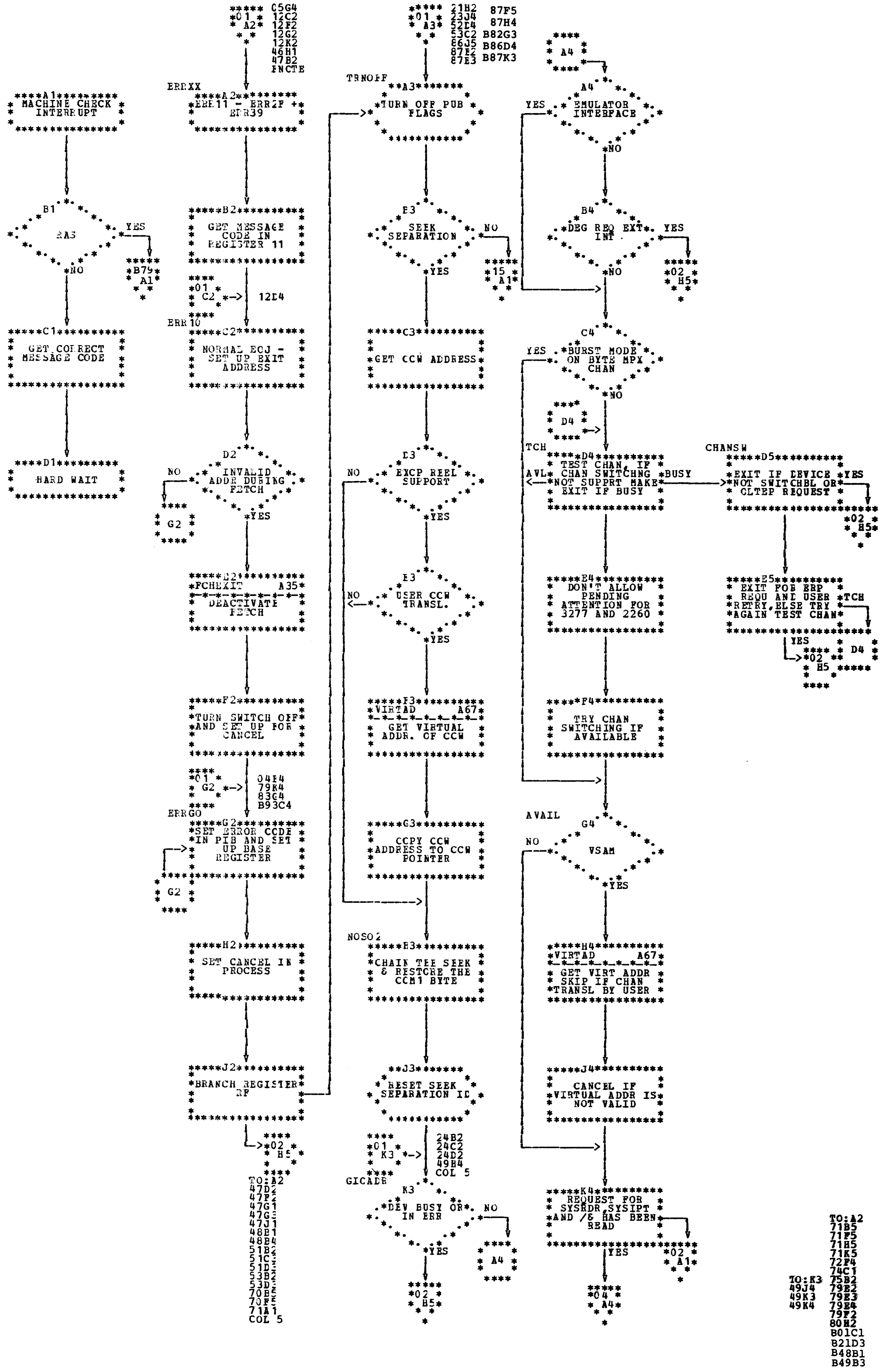


Chart A01. $$A$SUP1 - IOINTER Macro, I/O Interrupt Handler, Dequeue Channel Queue Entry
Refer to Chart 6.2.

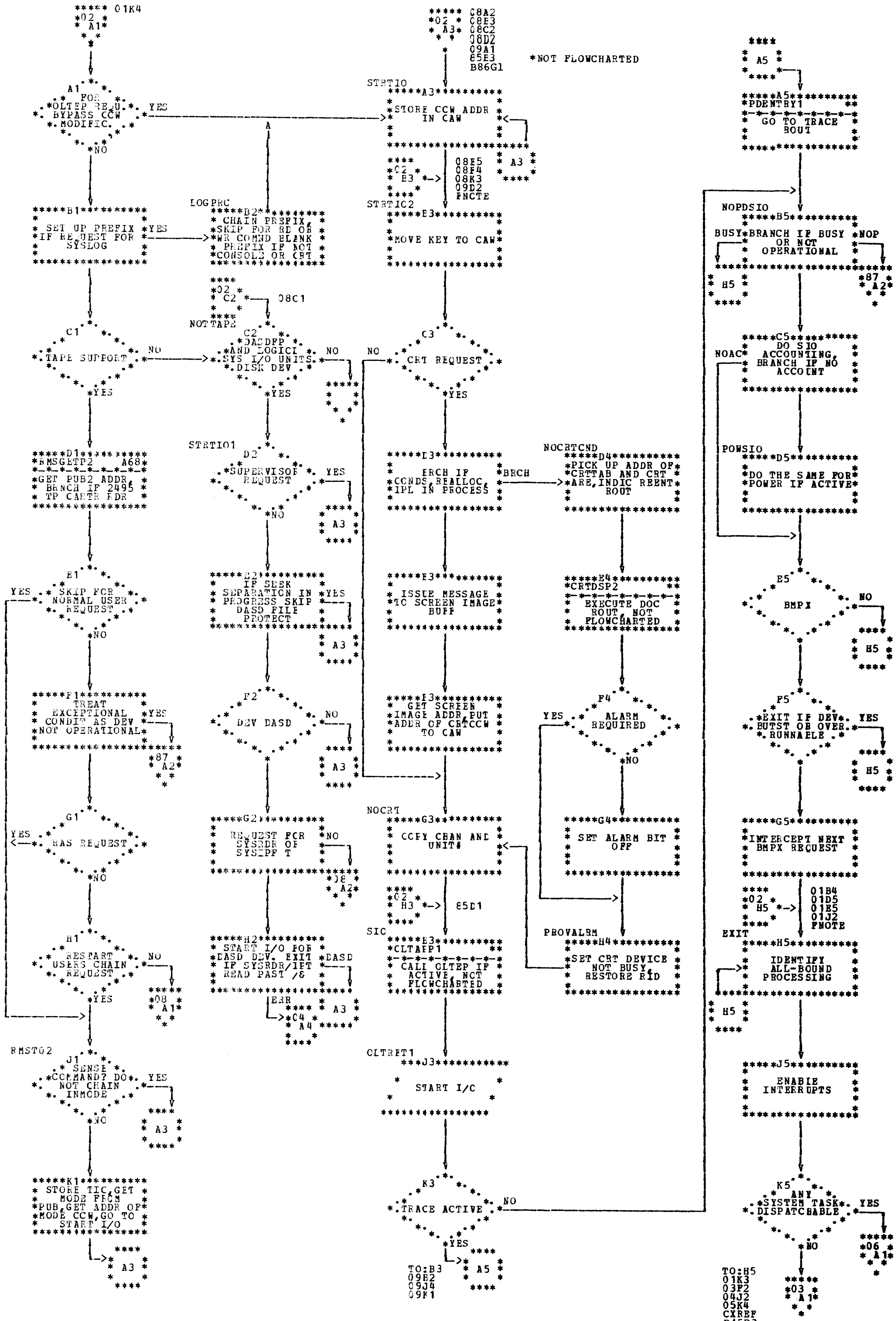


Chart A02. $$A$SUP1 - PIOCS Macro, Start I/O
Refer to Charts 2.1 and 6.4.

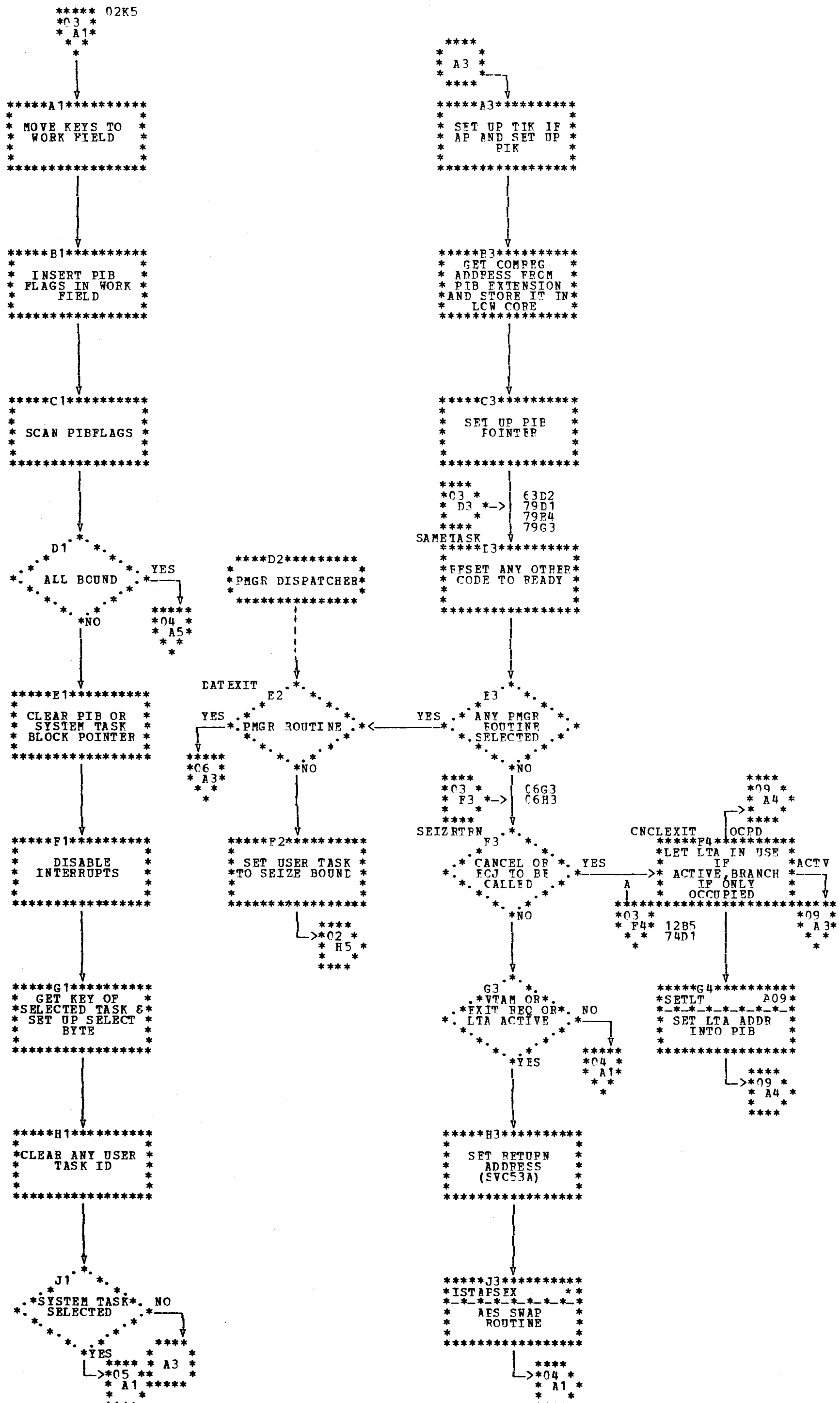


Chart A03. $$A$SUP1 - FOPT Macro, Task Selection, Initial Selection of Attention
Refer to Charts 2.1 and 4.1.

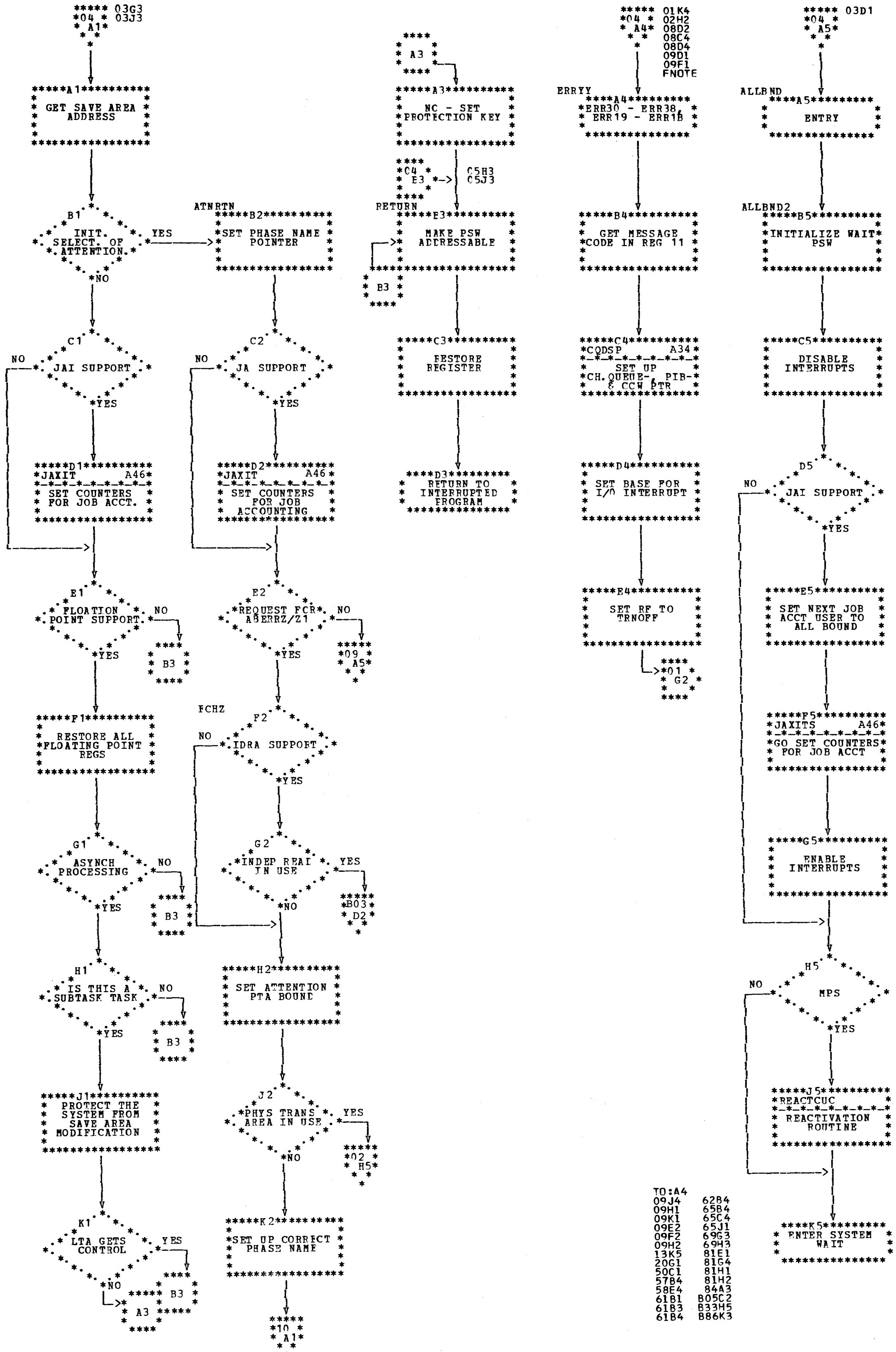


Chart A04. $$A$SUP1 - FOPT Macro, Task Selection, Initial Selection of Attention; System Allbound
Refer to Charts 2.1 and 4.1.

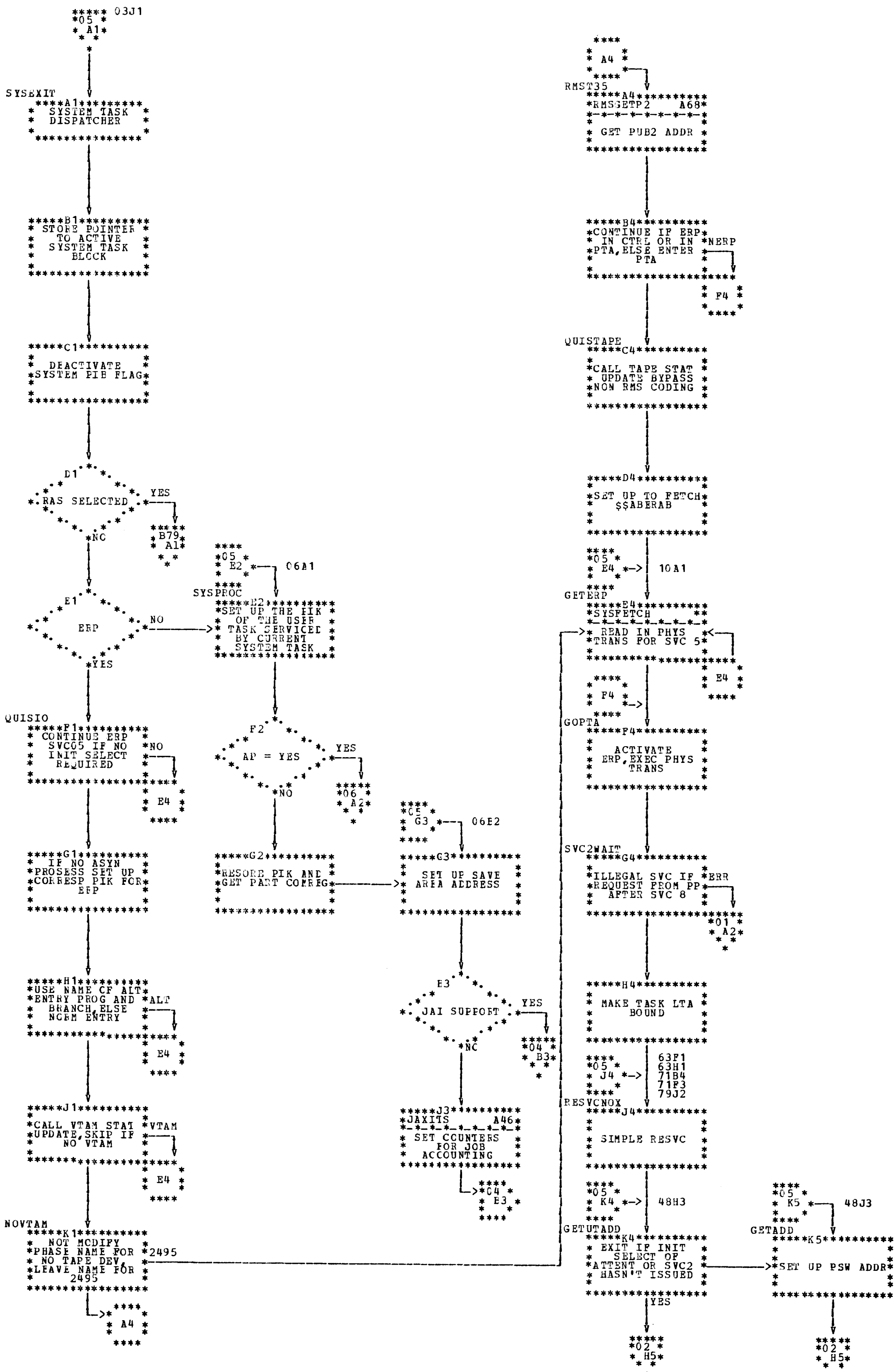


Chart A05. $$A$SUP1 - FOPT Macro, Task Selection, System Task Dispatcher
Refer to Chart 2.1.
PIOCS Macro, Initial Selection of ERP
Refer to Charts 2.1 and 2.3.

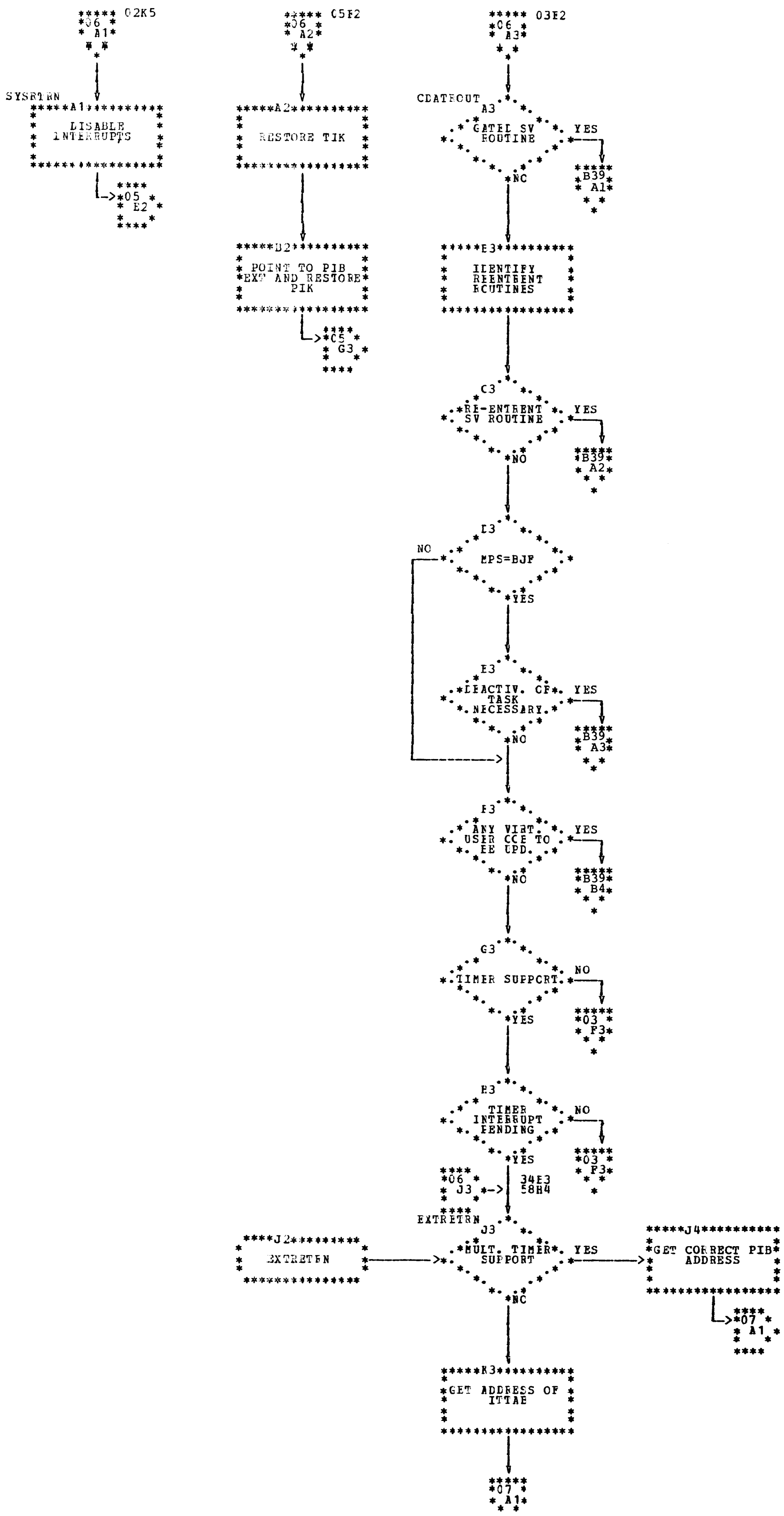


Chart A06. $$A$SUP1 - FOPT Macro, Task Selection, System Task Dispatcher
Refer to Chart 2.1.

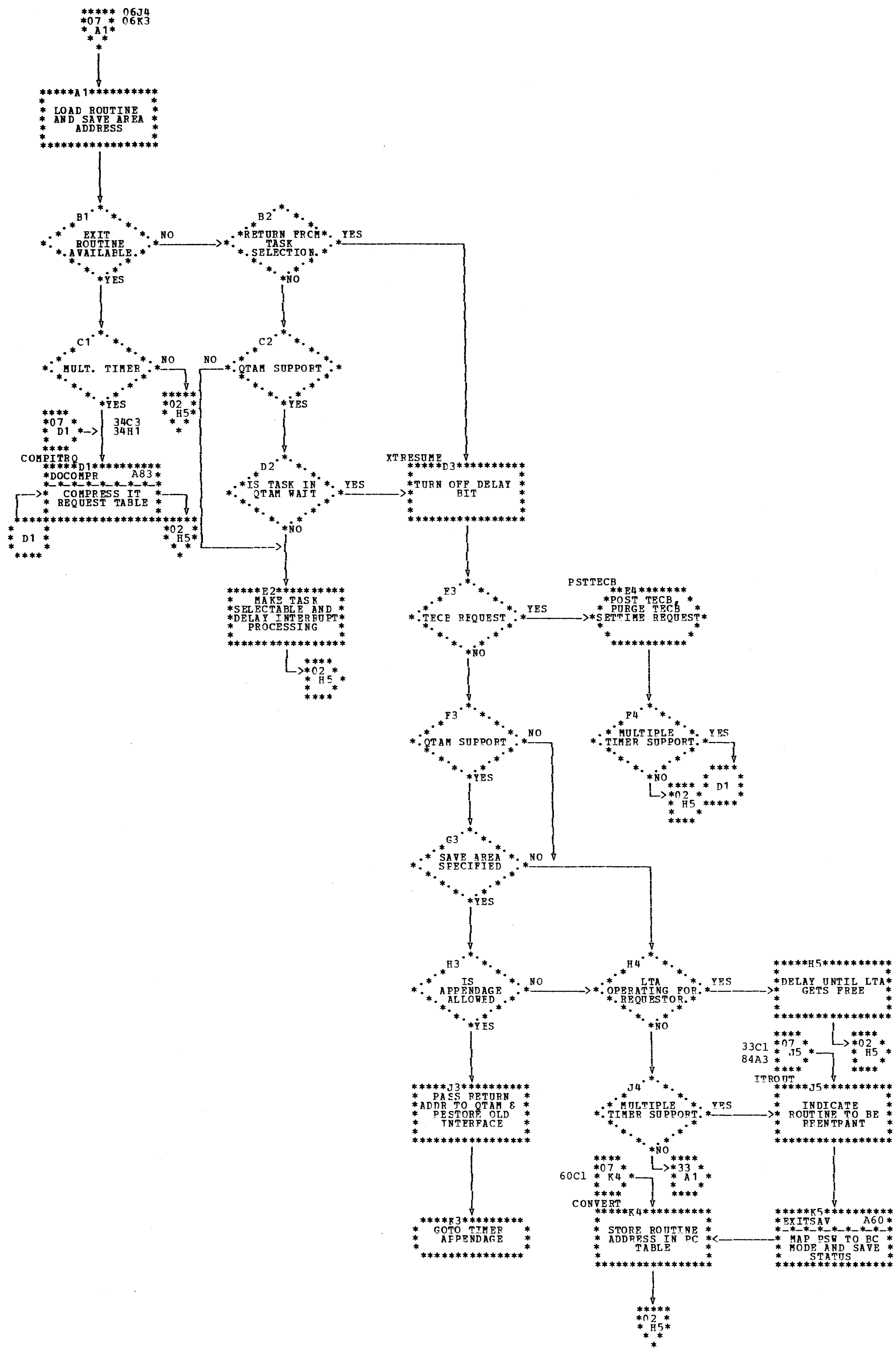


Chart A07. $$A$SUP1 - SMICR Macro, External Interrupt Handler
Refer to Chart 07.1.

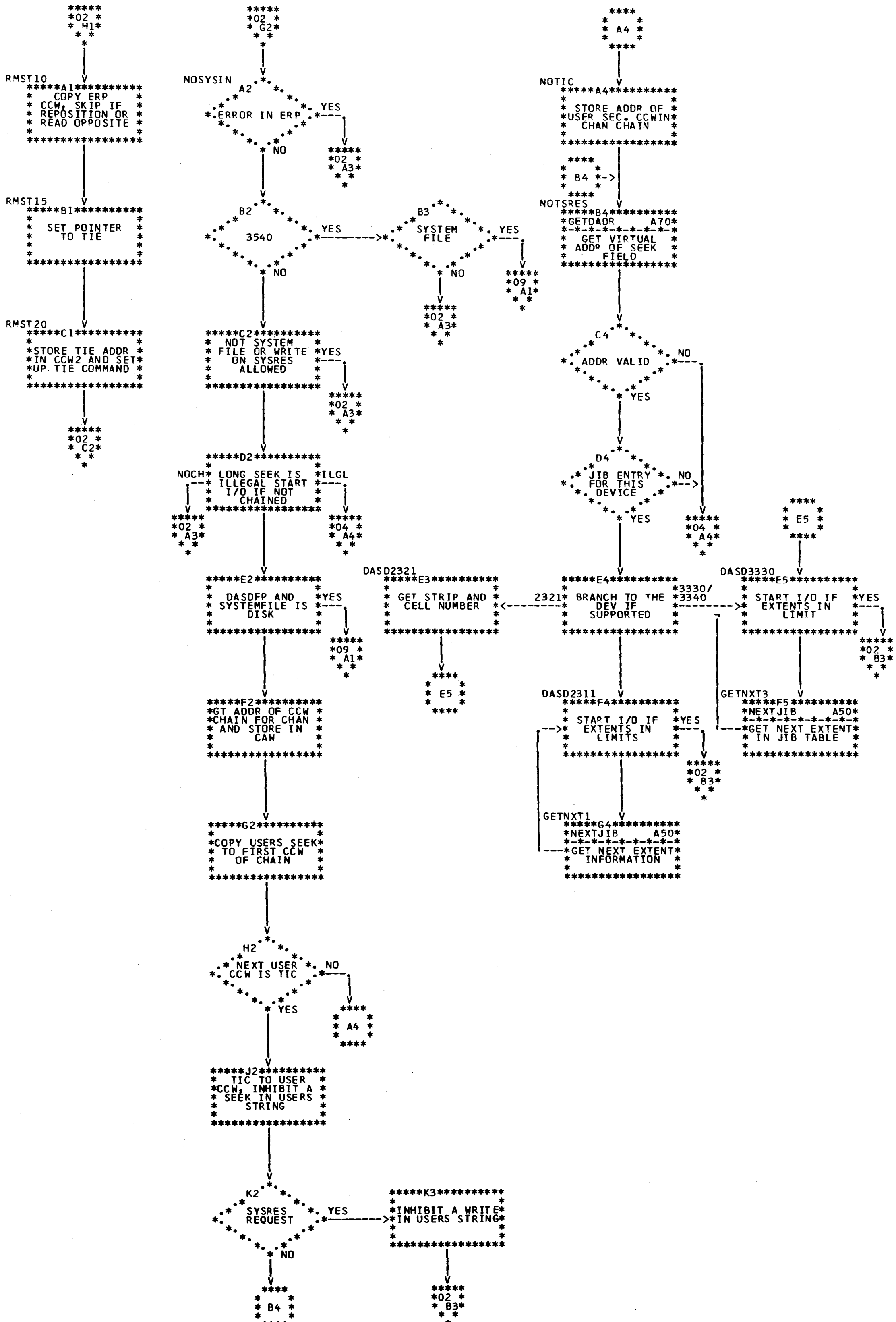


Chart A08. $$A$SUP1 - PIOCS Macro, Check Seek Address
Refer to Chart C6.4

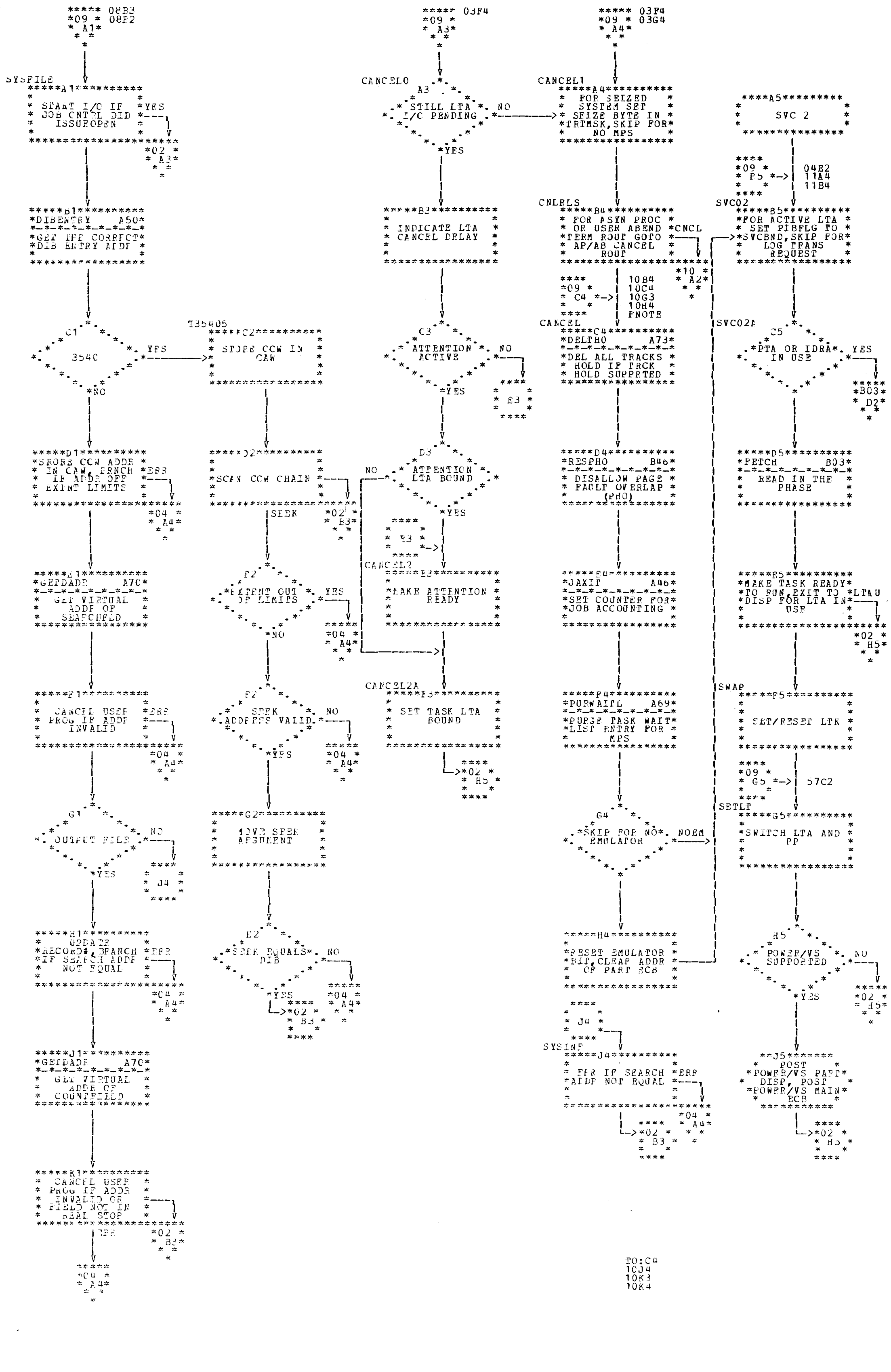


Chart A09. $$A$SUP1 - PIOCS Macro, Cancel Exit
Refer to Chart 02.2.

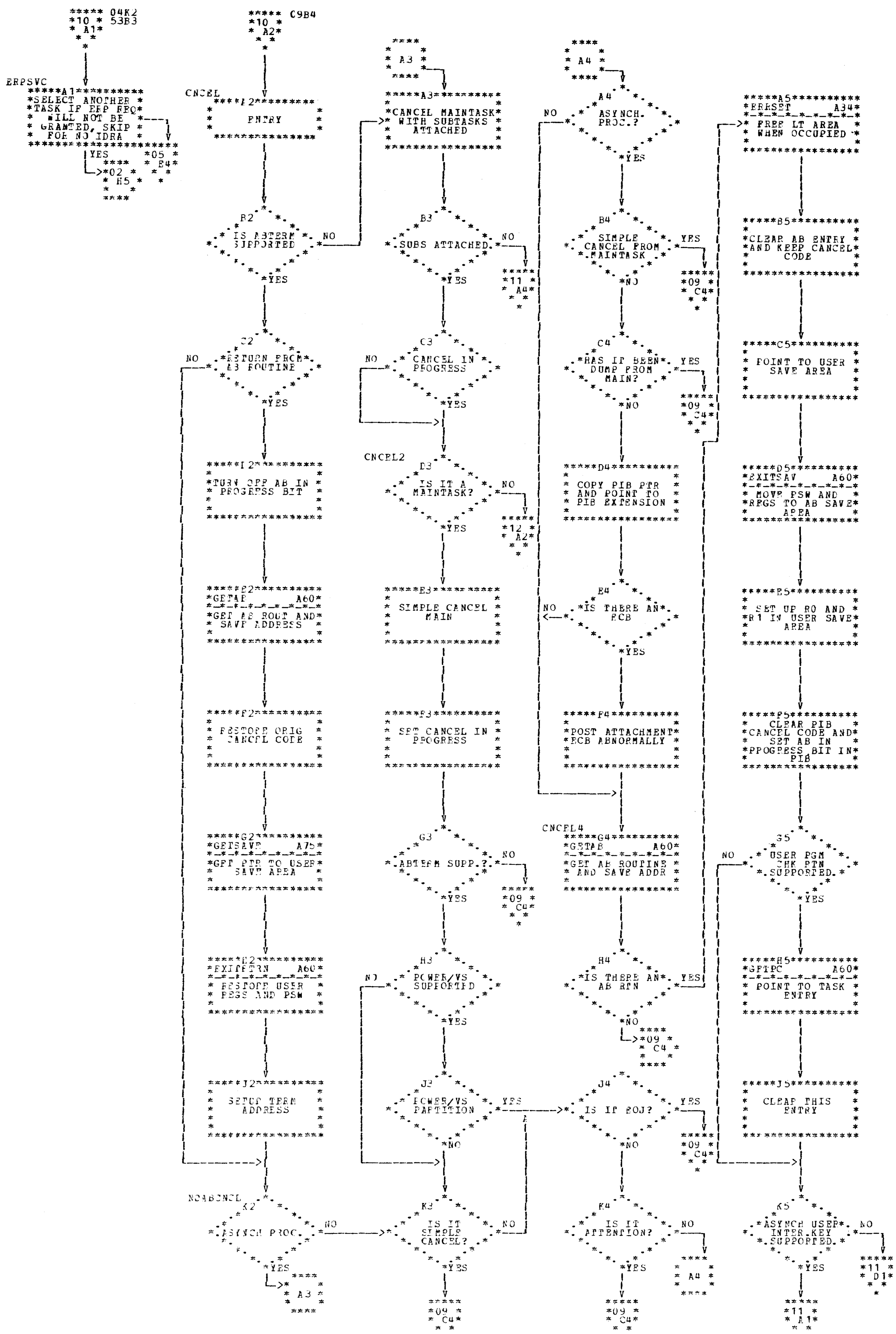


Chart A10. $$A$SUP1 - SGTHAP Macrc, AP Cancel Routine

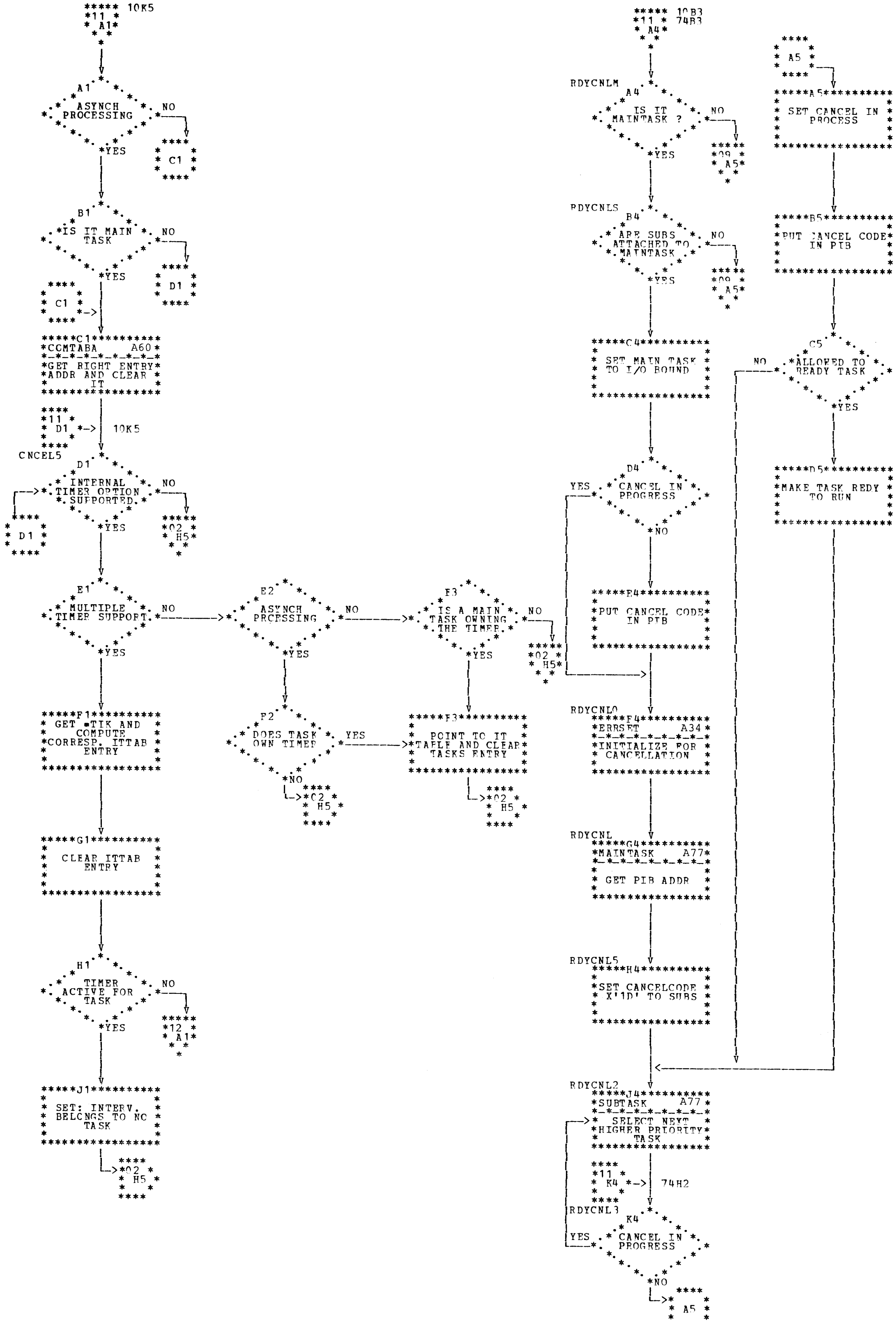


Chart A11. $$A$SUP1 - SGTHAP Macrc, AP Cancel Rcutine, Store Cancel Ccdes

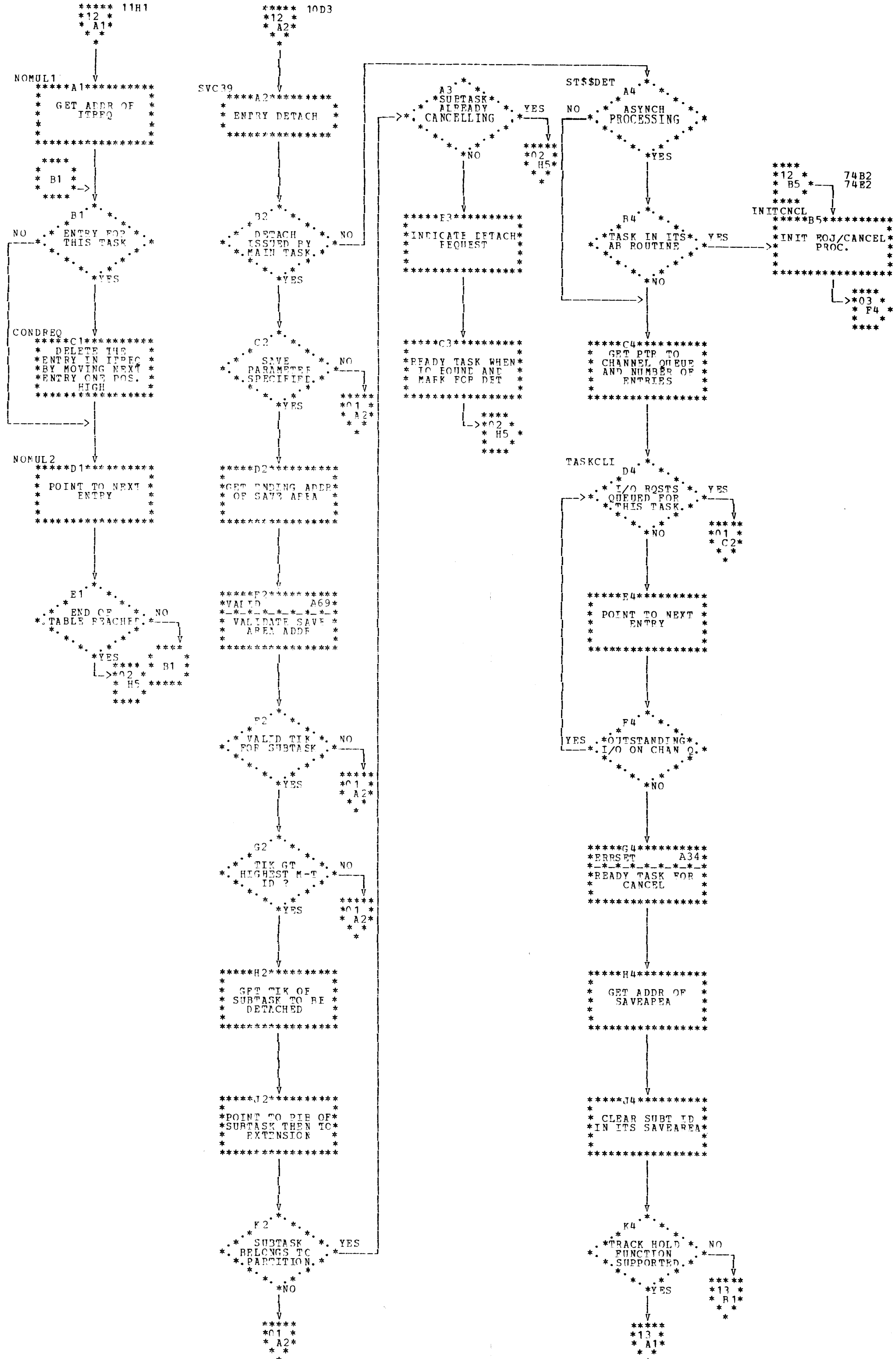


Chart A12. $$A$SUP1 - SGTHAP Macro, Detach Subtask (SVC 39)
Refer to Chart 05.1.

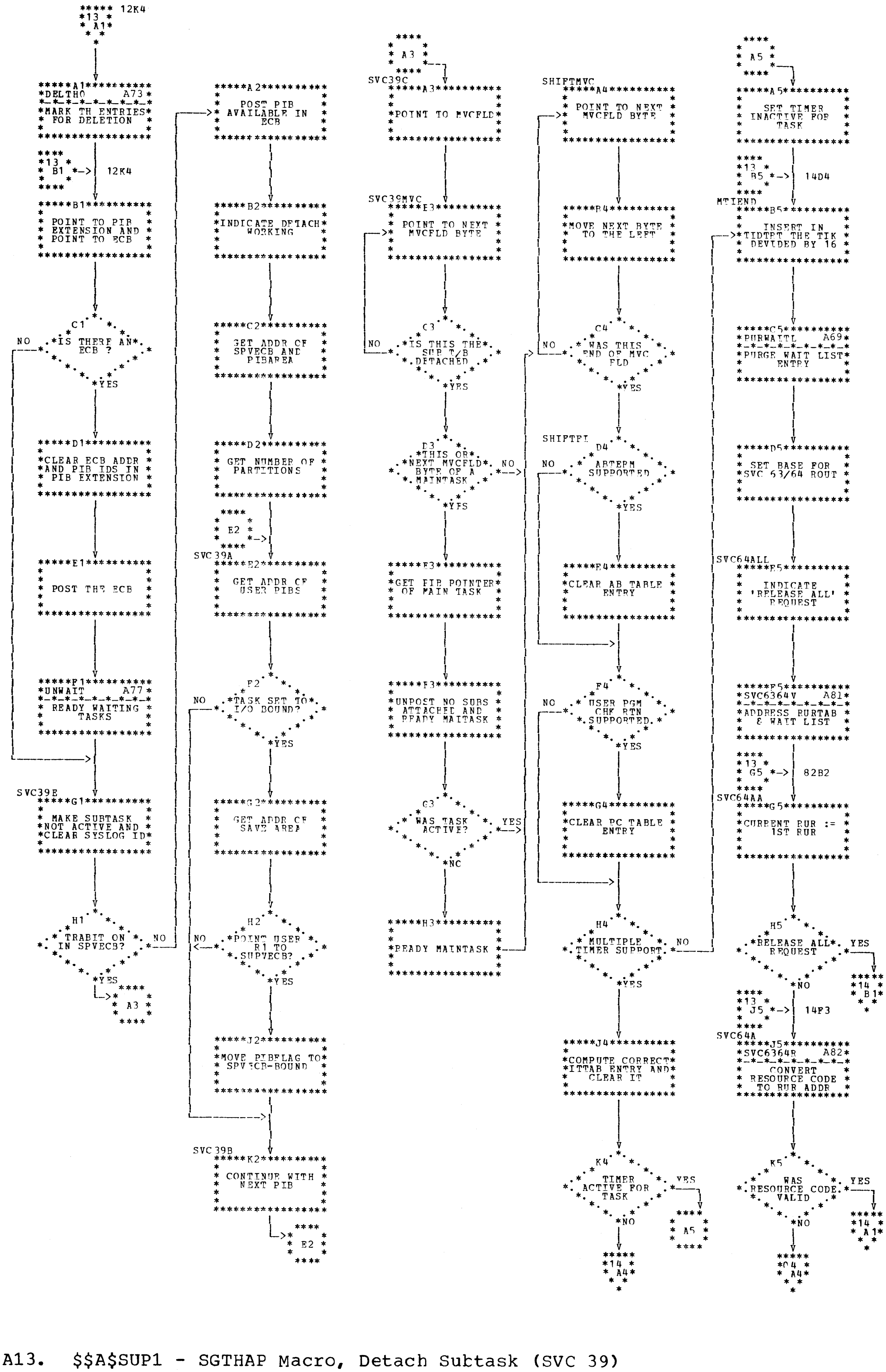


Chart A13. $$A$SUP1 - SGTHAP Macro, Detach Subtask (SVC 39)
Refer to Chart 05.1.

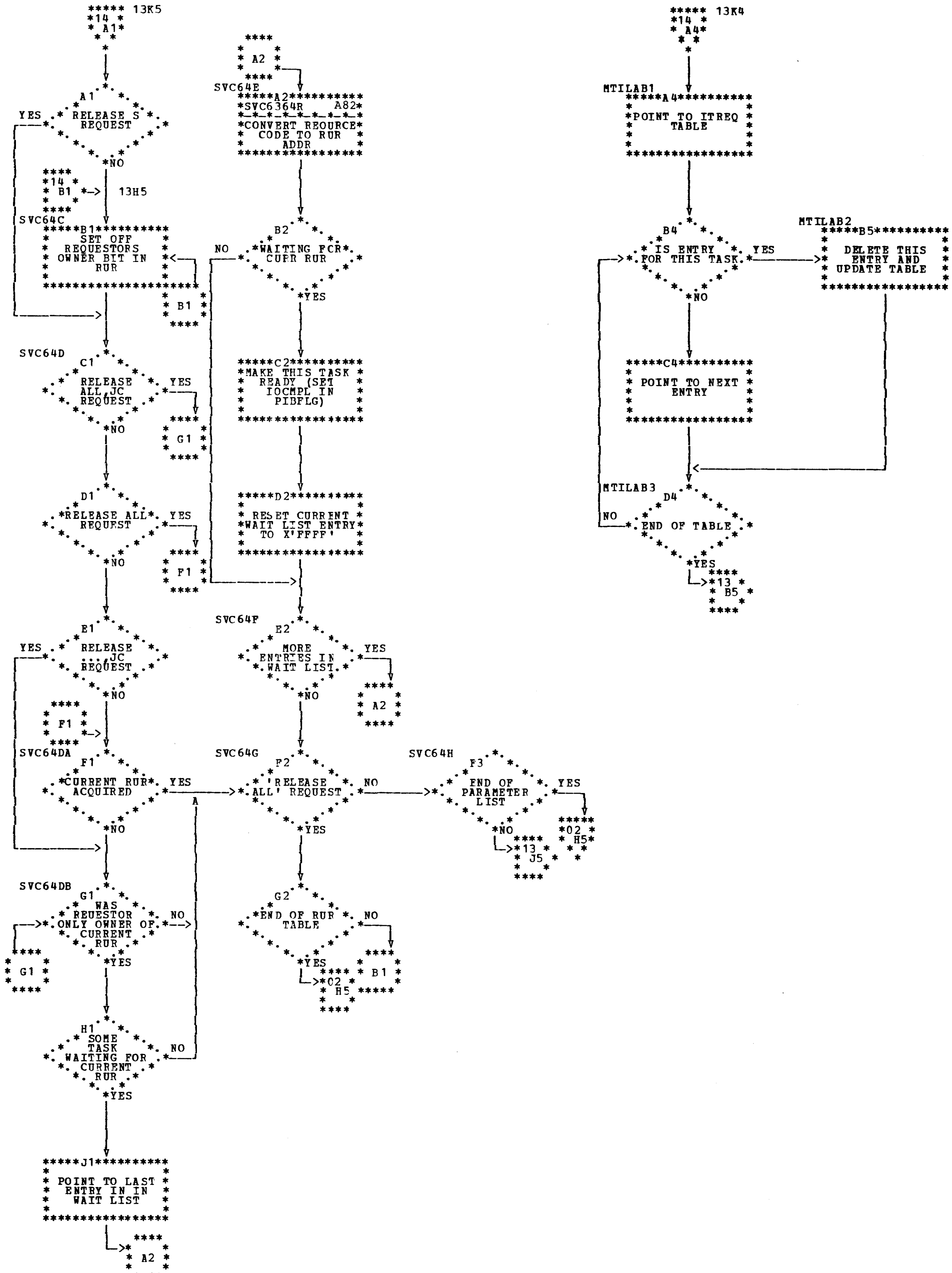


Chart A14. $$A$SUP1 - IOINTER Macro, Release a Resource (SVC 64)
Refer to Chart 05.2.

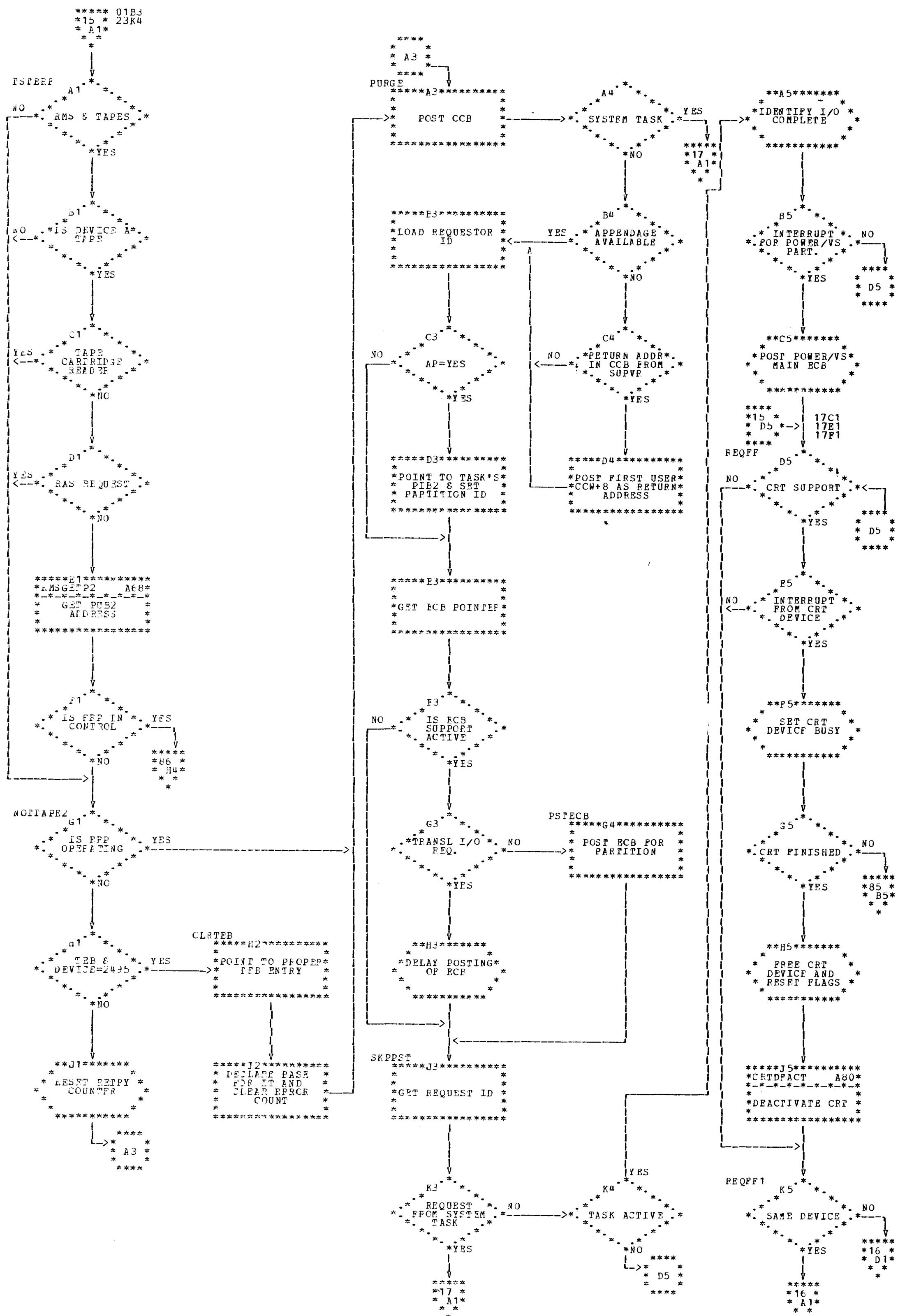


Chart A15. $$A$SUP1 - IOINTER Macro, Dequeue Channel Queue Entry
Refer to Chart 06.2.

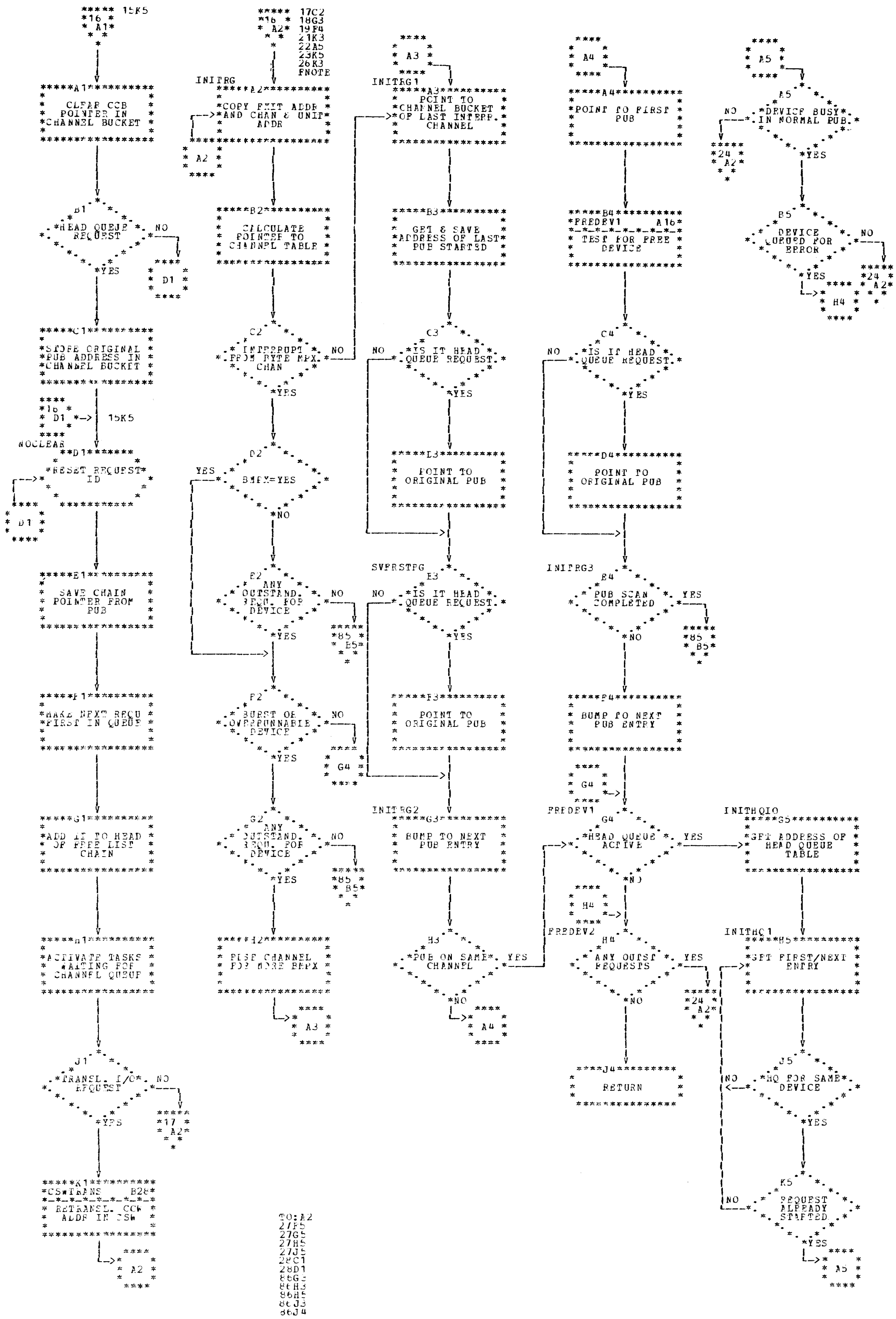


Chart A16. $$A$SUP1 - IOINTER Macro, Reschedule Channel
Refer to Chart 06.2.

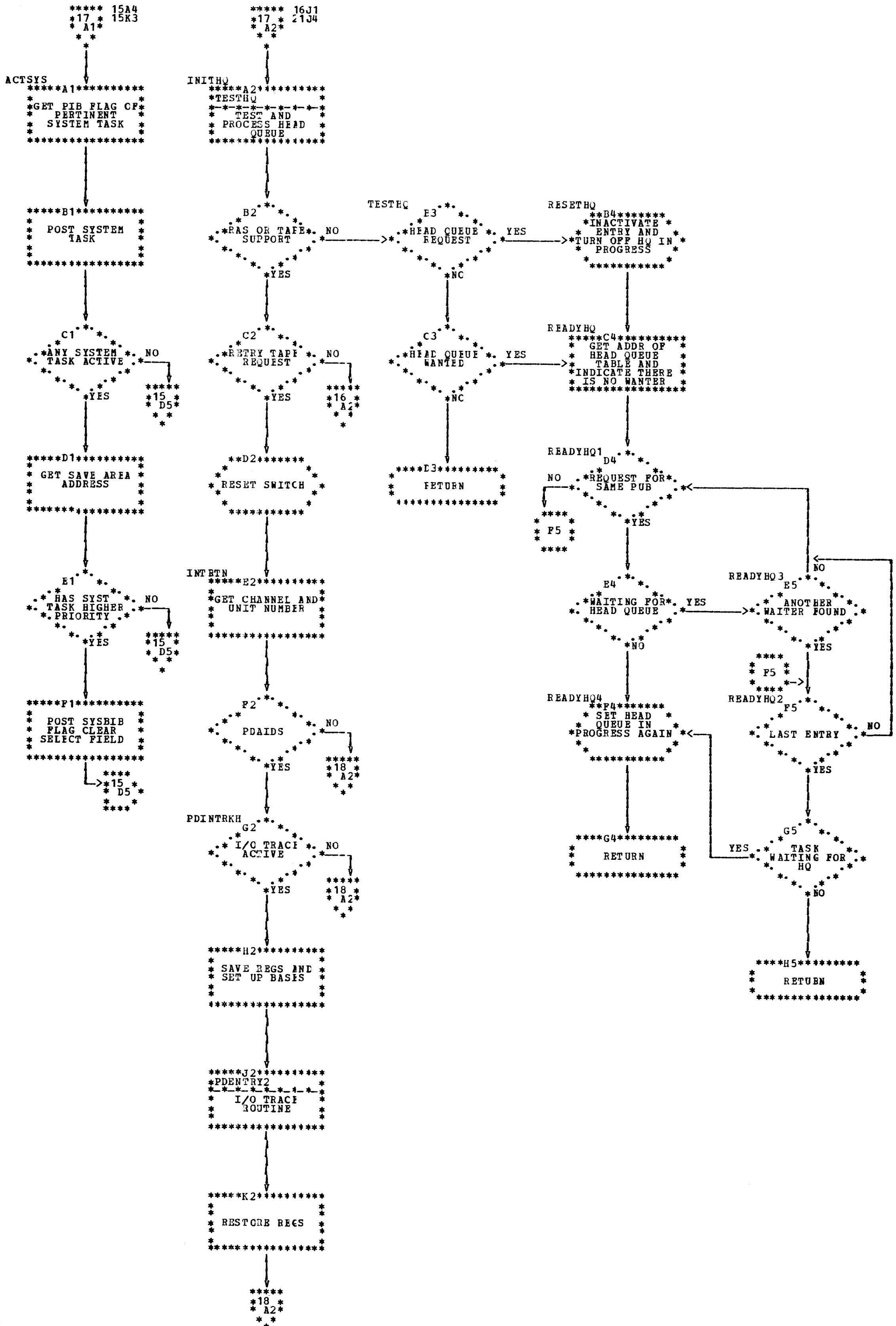


Chart A17. $$A$SUP1 - IOINTER Macro, Initialize Headqueuing
Refer to Chart 06.2.

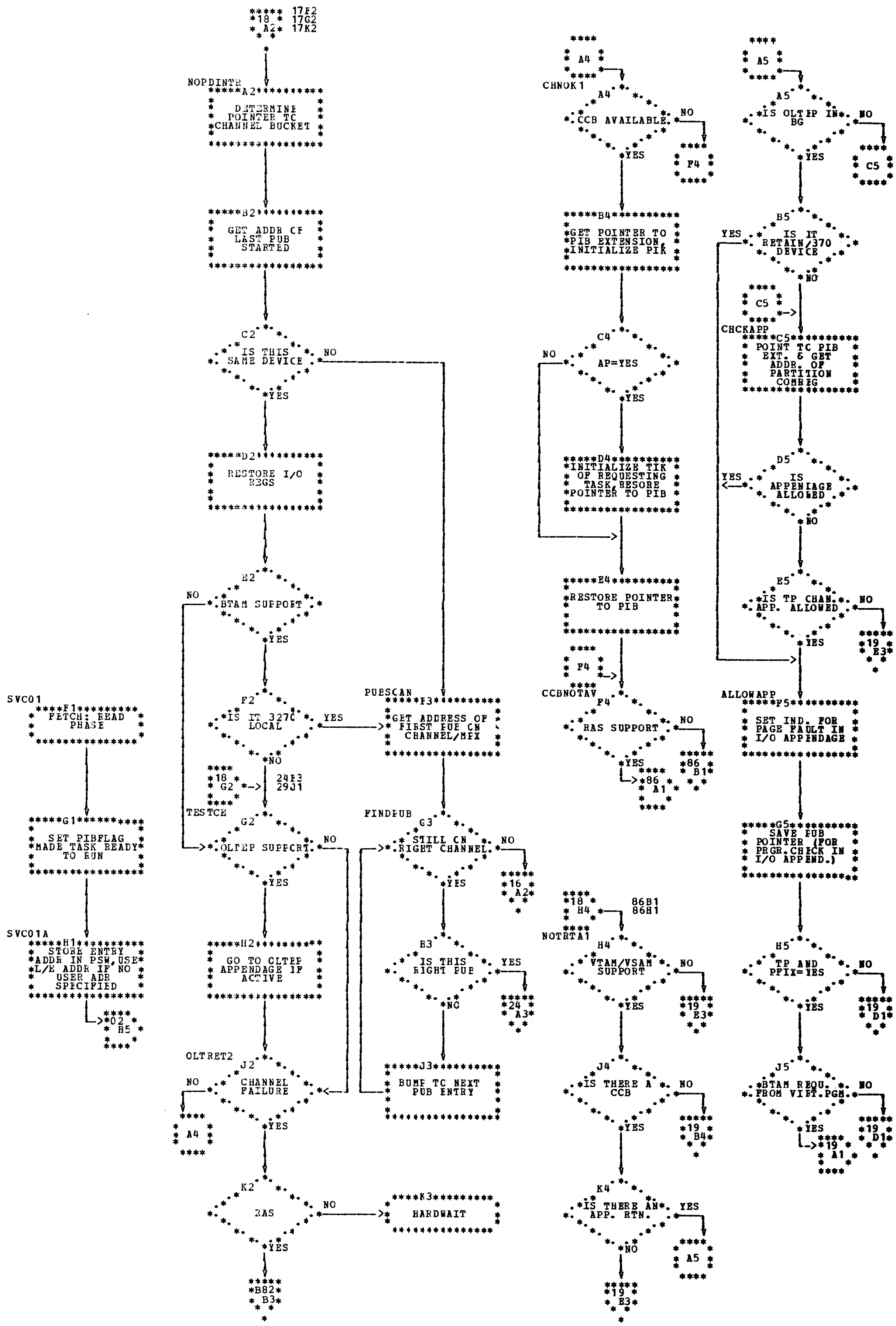


Chart A18. $$A$SUP1 - IOINTER Macro, I/O Interrupt Handler, PUBSCAN Routine
Refer to Chart 06.1.

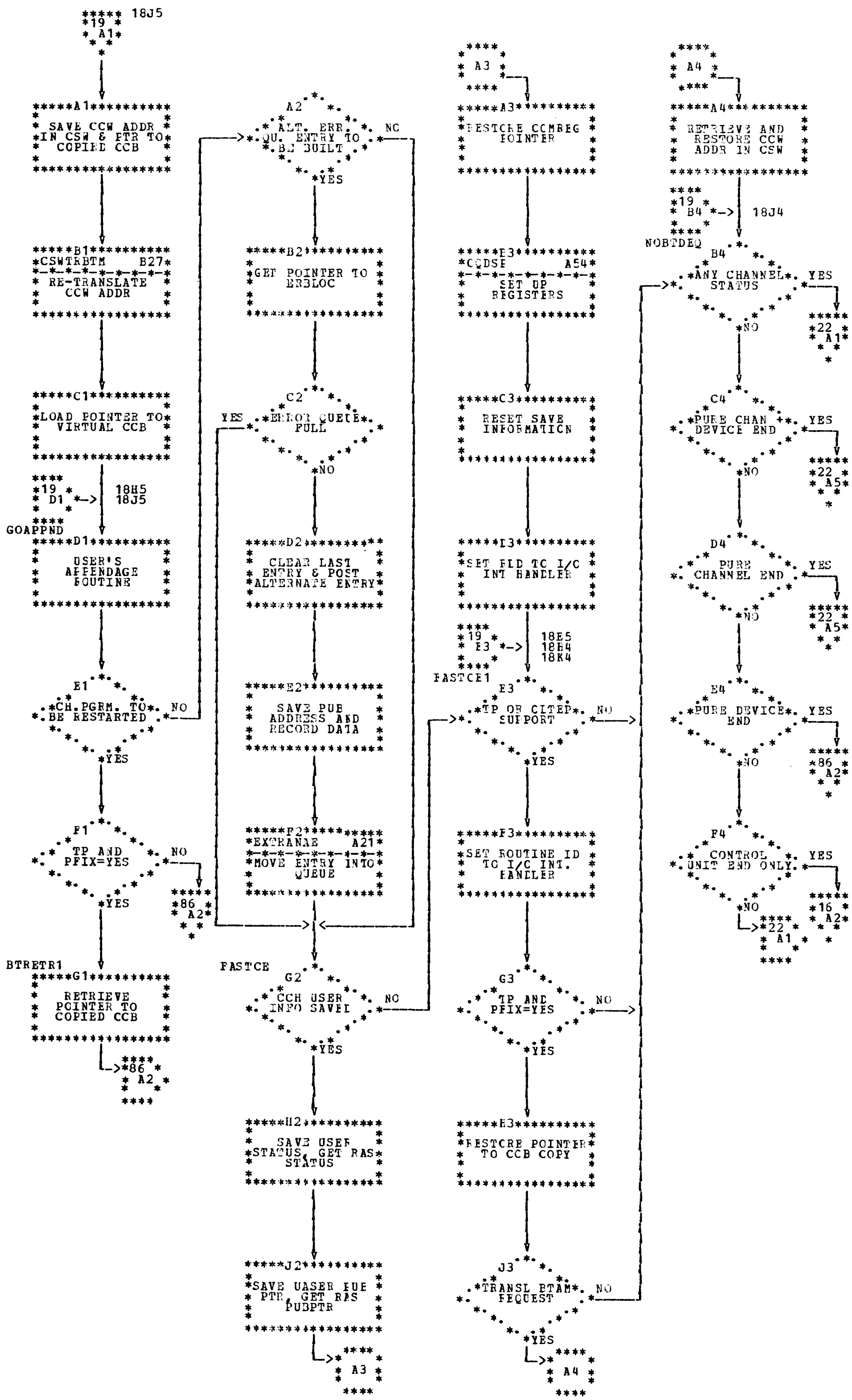


Chart A19. $$A$SUP1 - IOINTER Macro, I/O Interrupt Handler, Channel Appendage Interface
Refer to Chart 06.1.

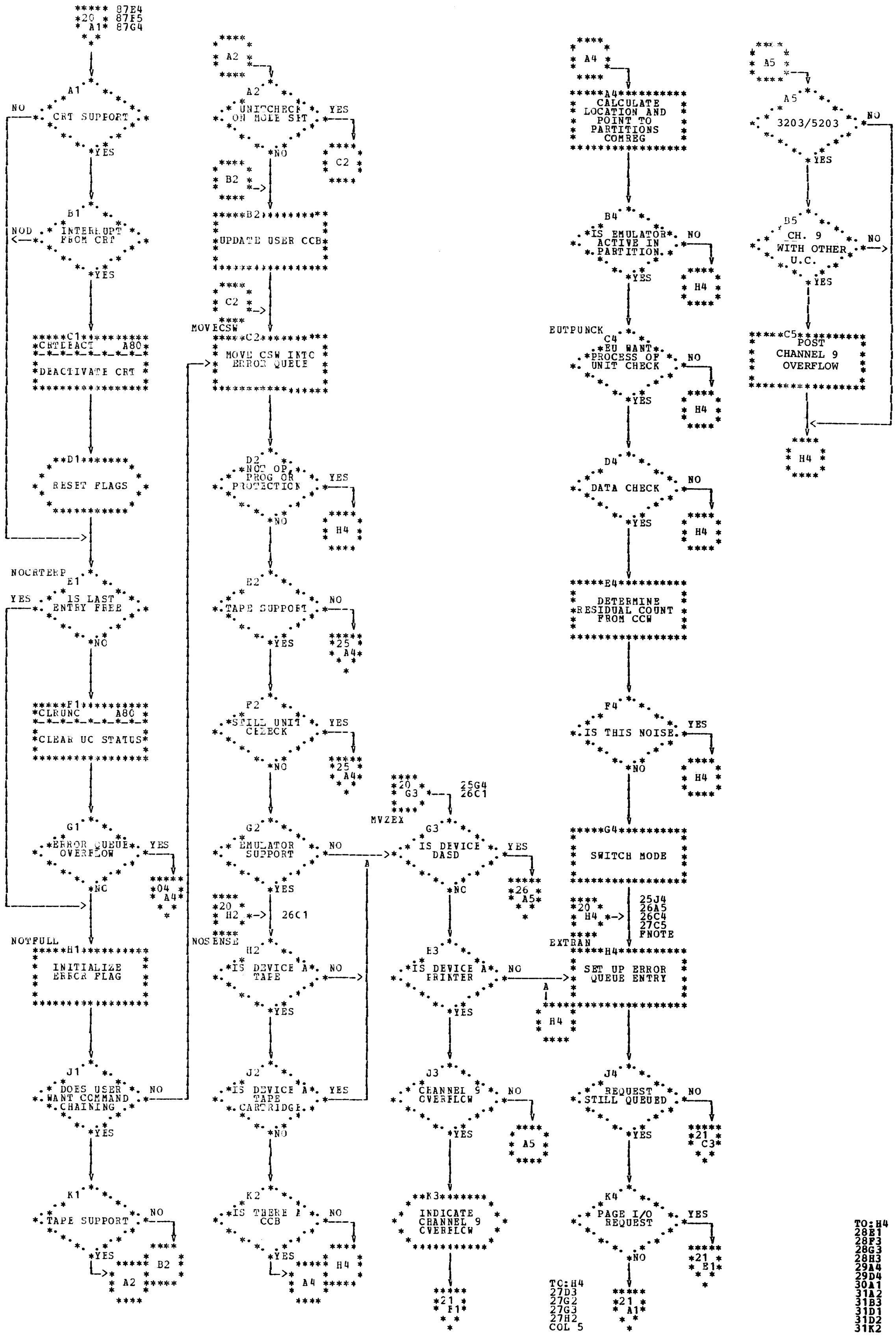


Chart A20. $$A$SUP1 - IOINTER Macro, I/O Interrupt Handler, Unit Check Routine
Refer to Charts 09.1 and 11.1.

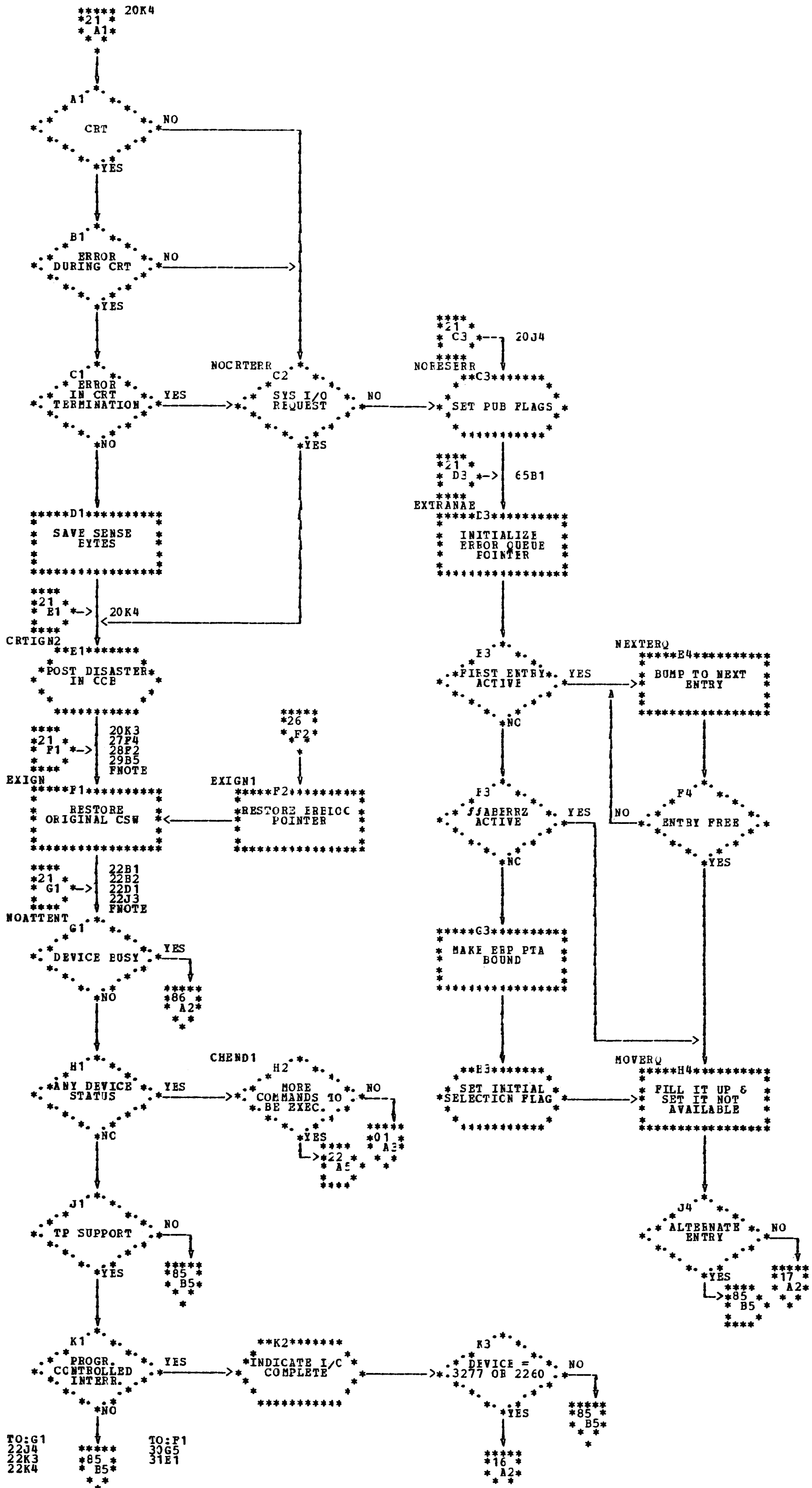


Chart A21. $$A$SUP1 - IOINTER Macro, I/O Interrupt Handler, Active ERP
Refer to Charts 06.1 and 09.1.

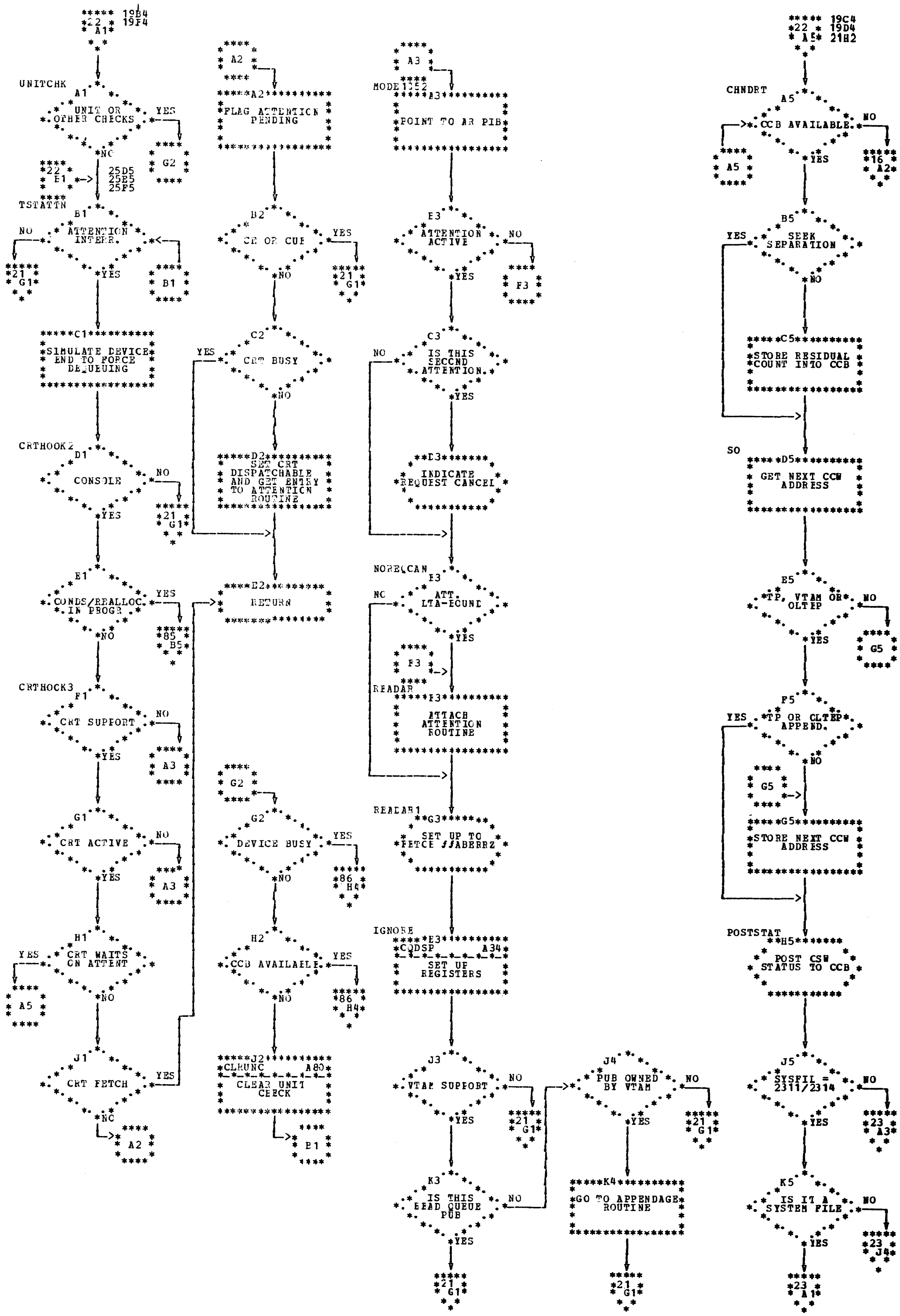


Chart A22. $$A$SUP1 - IOINTER Macro, I/O Interrupt Handler, Activate Attention
Refer to Chart 06.1.

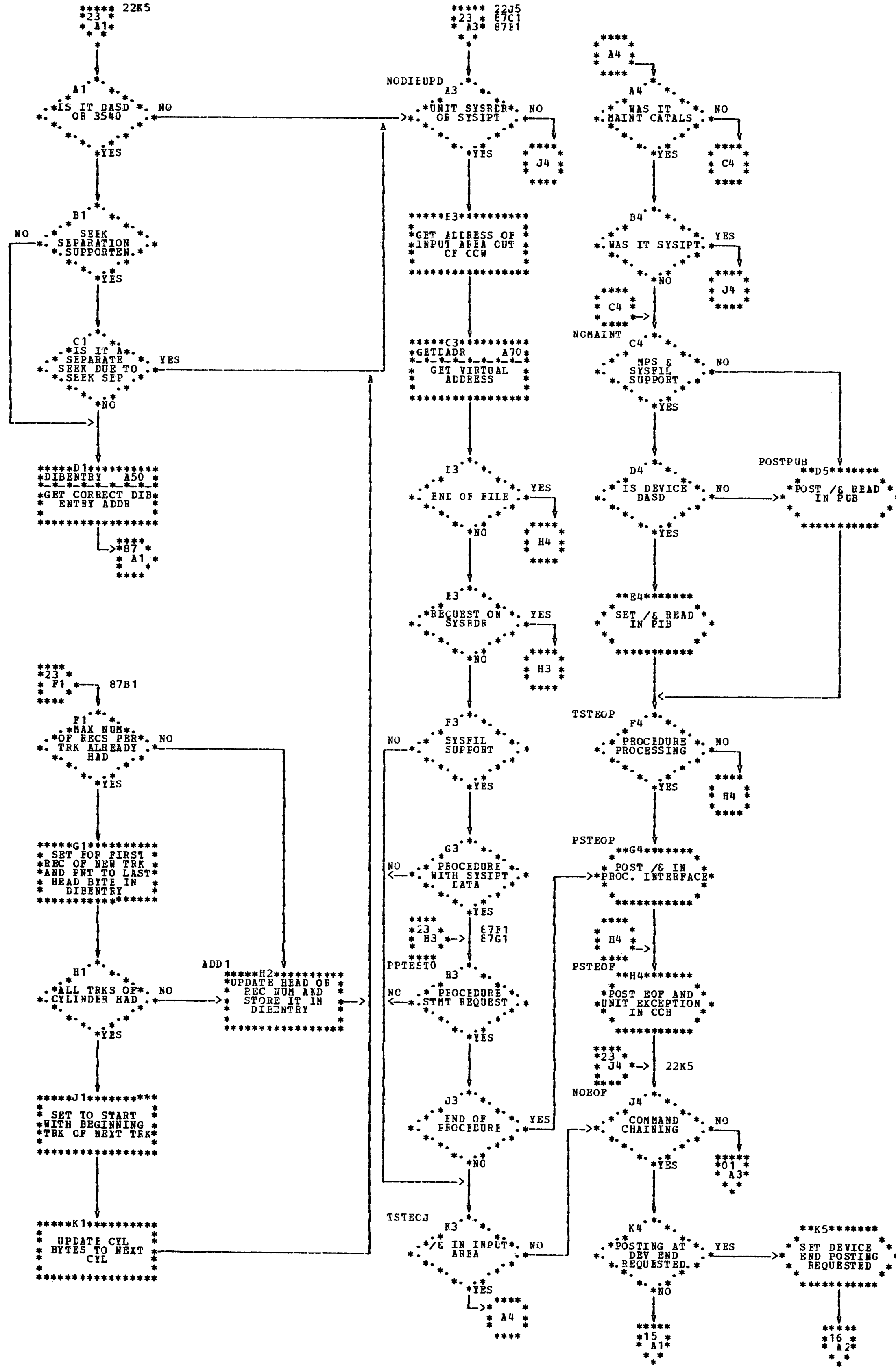


Chart A23. $$A$SUP1 - IOINTER Macro, I/O Interrupt Handler
Refer to Chart 06.1.

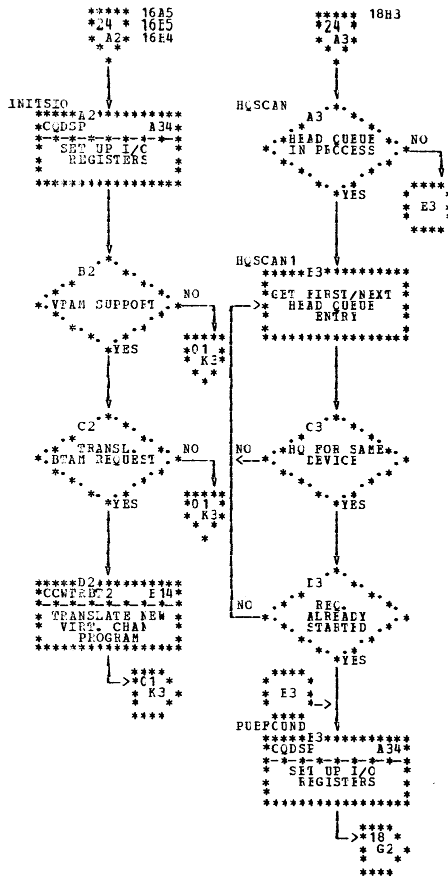


Chart A24. $$A$SUP1 - IOINTER Macro, I/O Interrupt Handler
Refer to Chart 06.2.

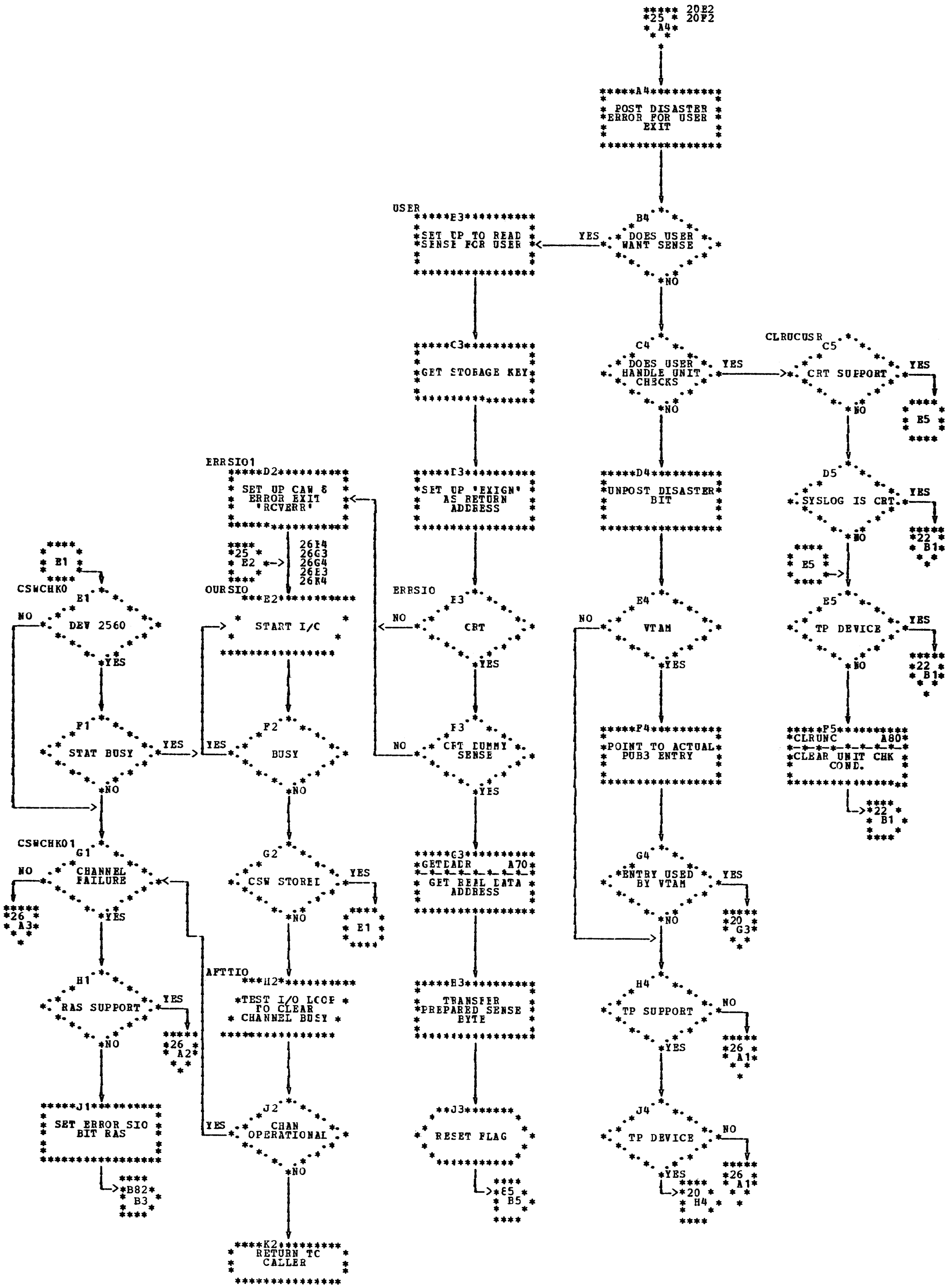


Chart A25. $$A$SUP1 - IOINTER Macro, I/O Interrupt Handler, Unit Check Routine
Refer to Charts 06.1 and 09.1.

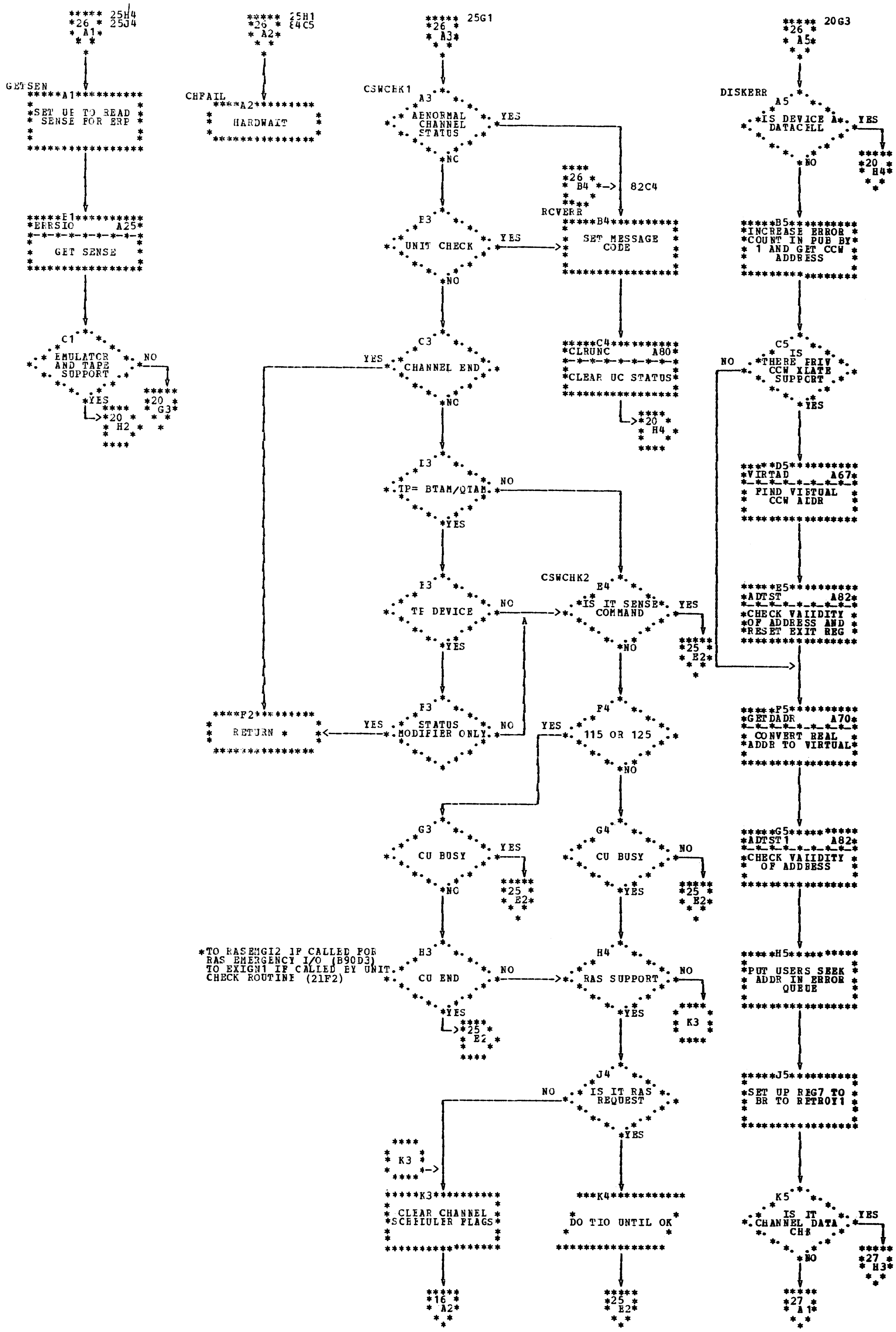


Chart A26. $$A$SUP1 - IOINTER Macro, I/O Interrupt Handler, Unit Check Routine
Refer to Charts 09.1 and 10.1.

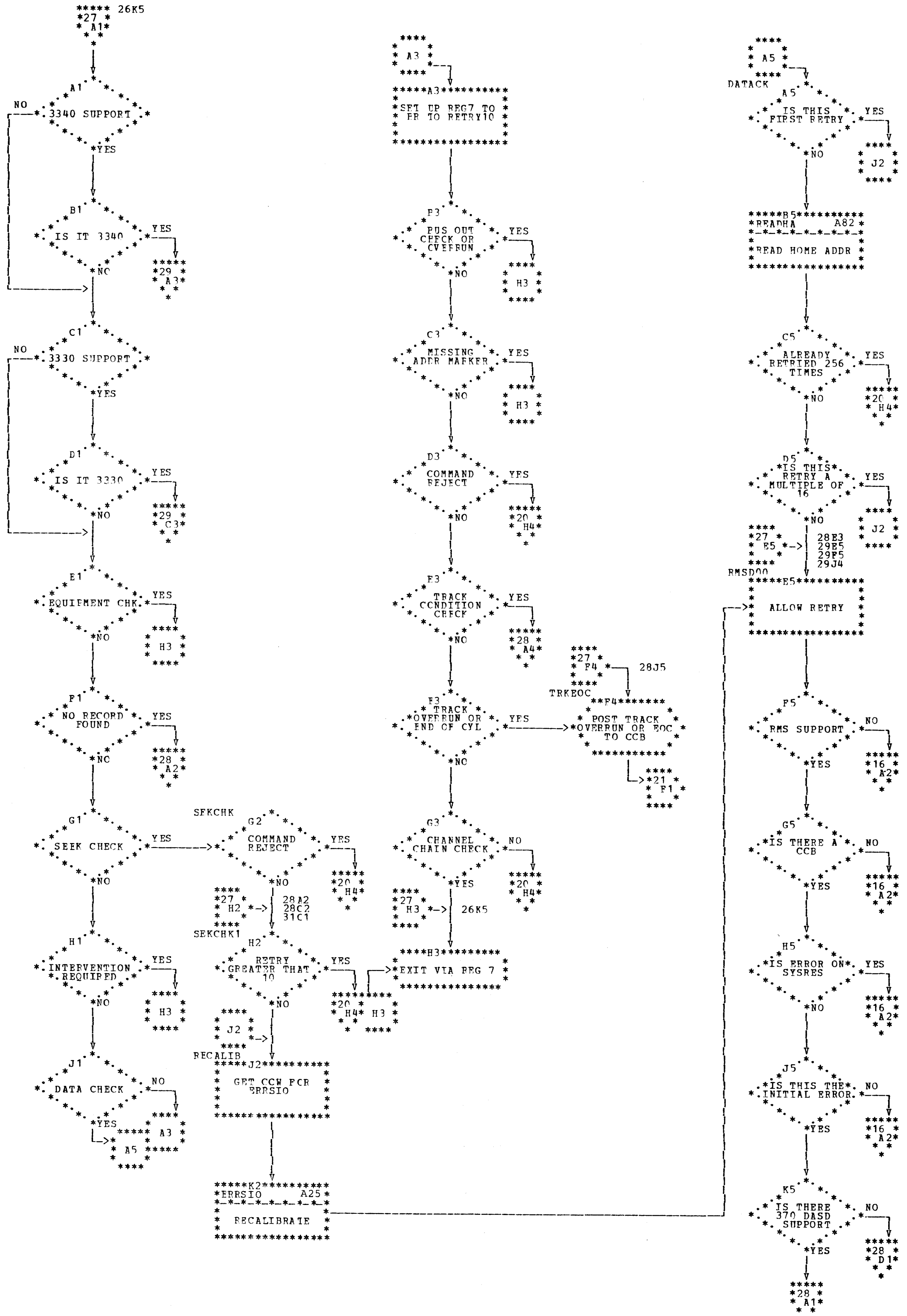


Chart A27. $$A$SUP1 - SGDSK Macro, Resident Disk ERP
Refer to Chart 10.1.

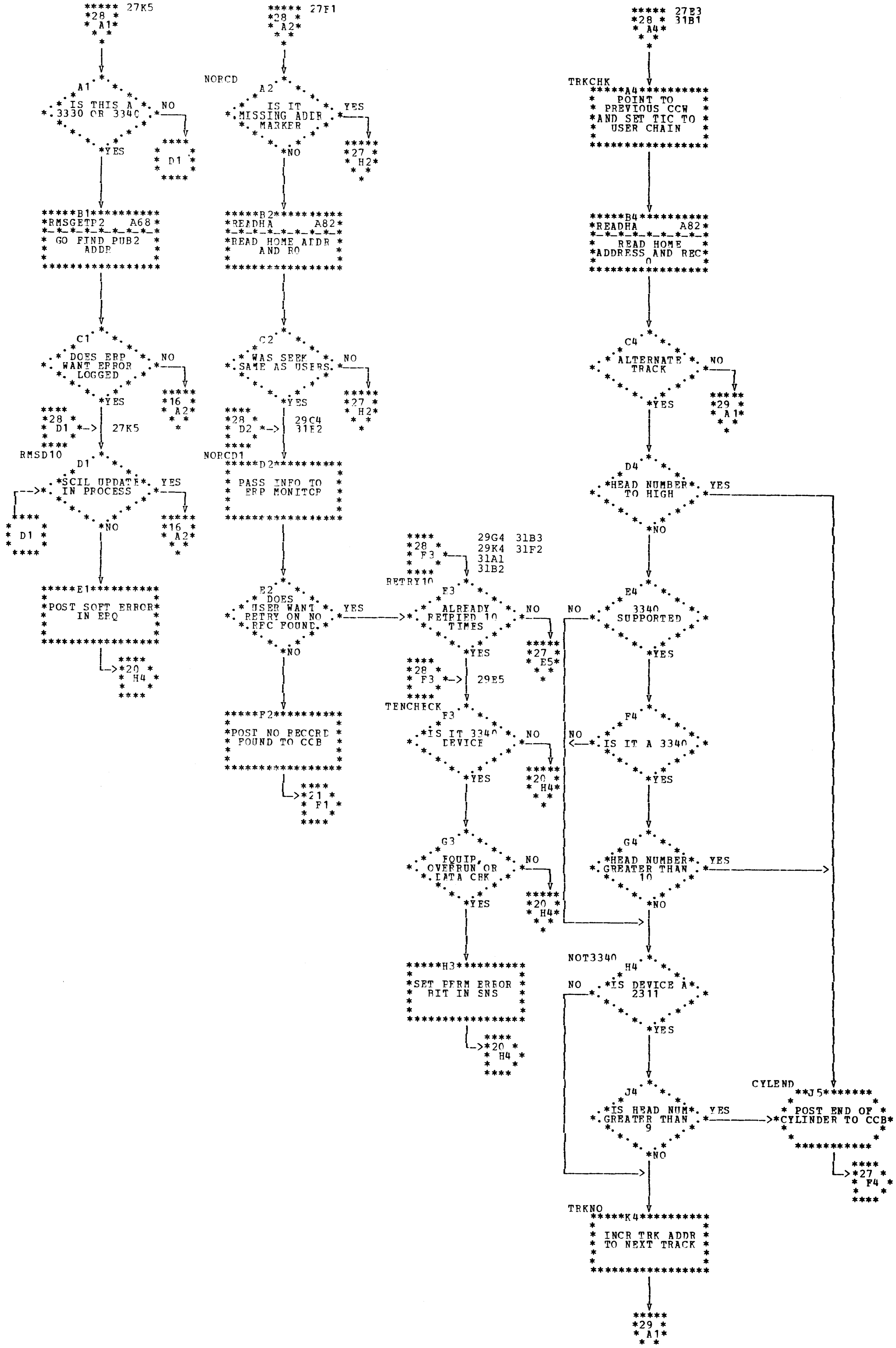


Chart A28. $$A$SUP1 - SGDSK Macro, Resident Disk ERP
Refer to Chart 10.1.

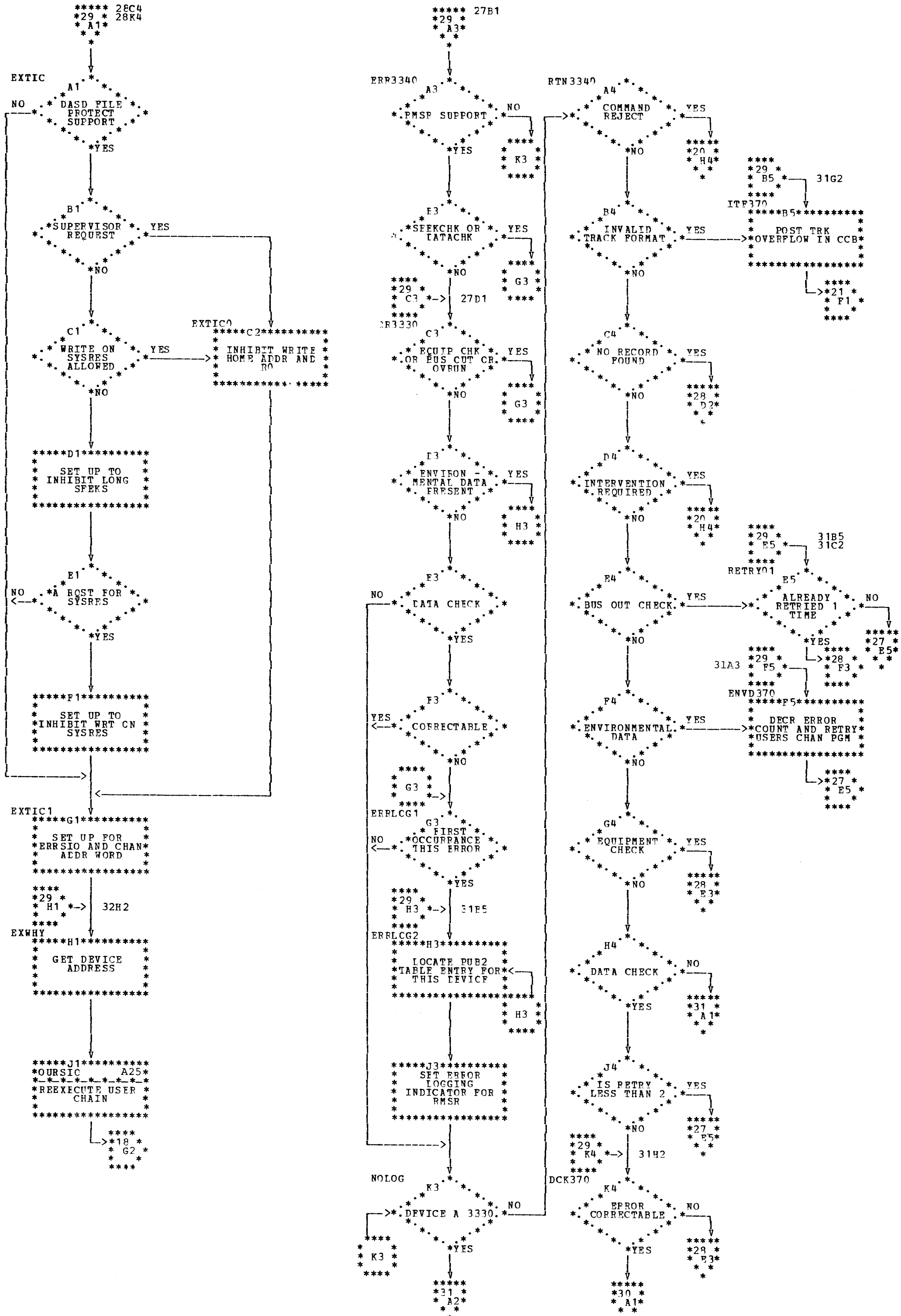


Chart A29. $$A$SUP1 - SGDSK Macro, Resident Disk ERP
Refer to Chart 10.1.

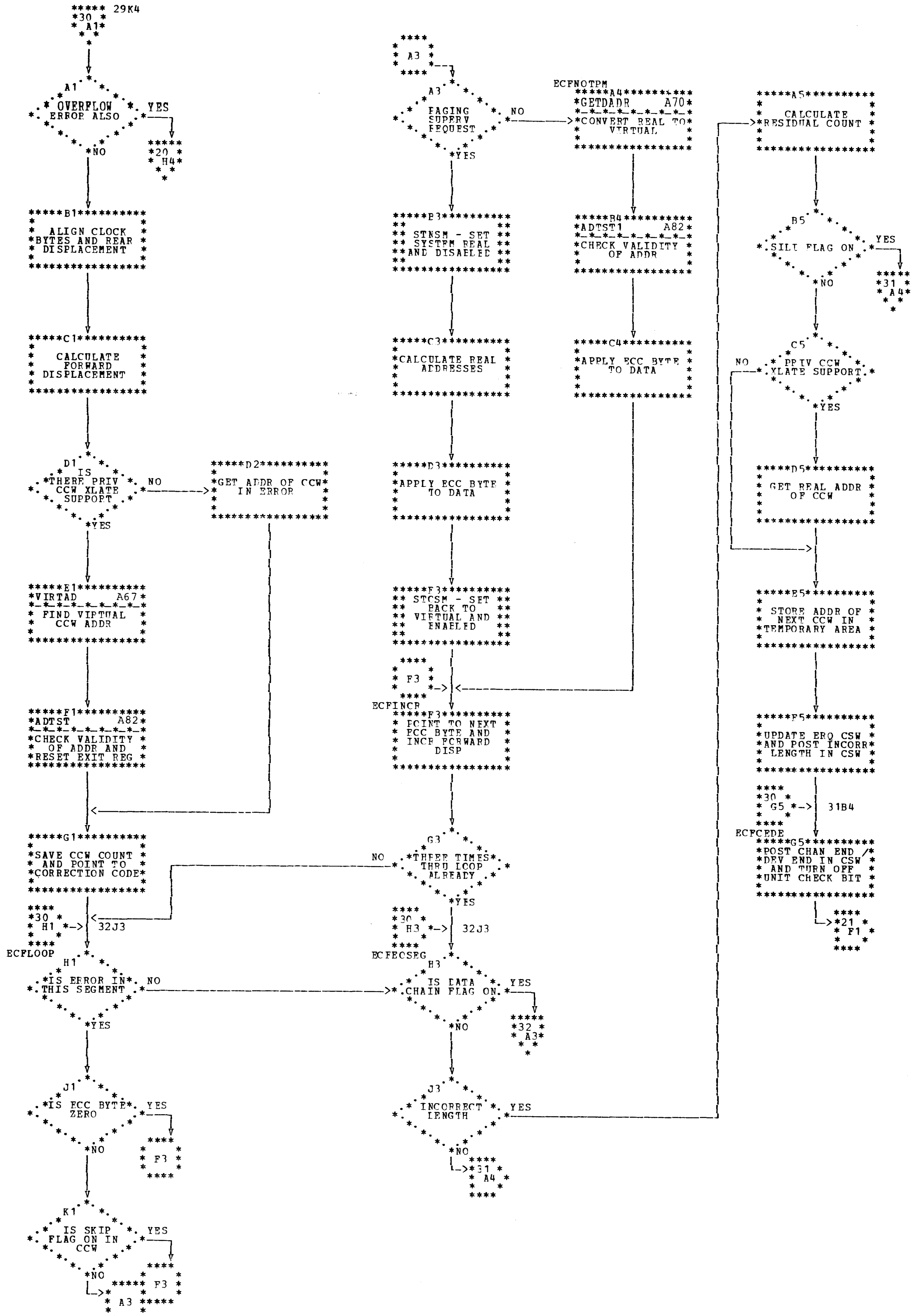


Chart A30. $$A$SUP1 - SGDSK Macro, Resident Disk ERP
Refer to Chart 10.1.

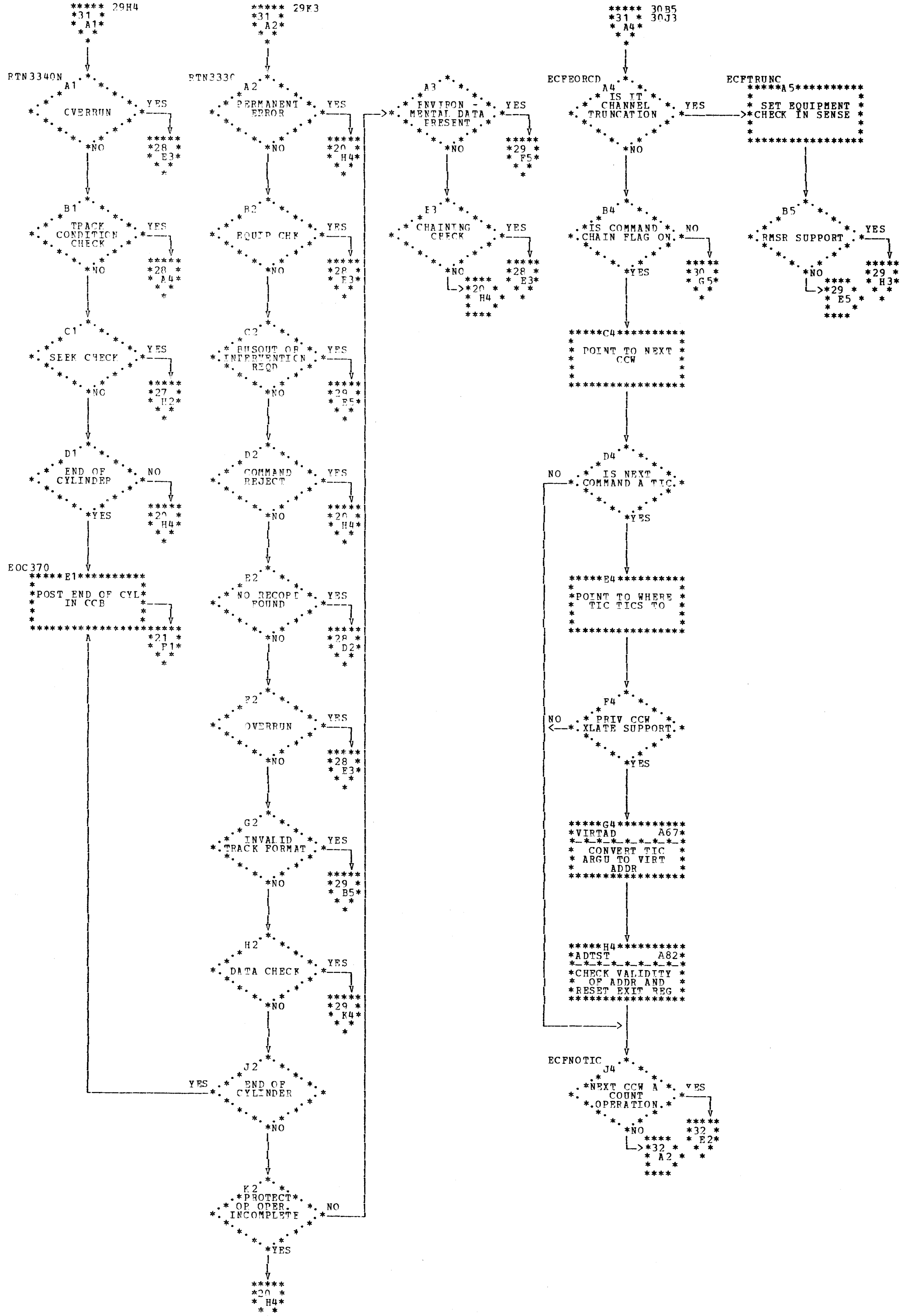


Chart A31. $$A$SUP1 - SGDSK Macro, Resident Disk ERP
Refer to Chart 10.1.

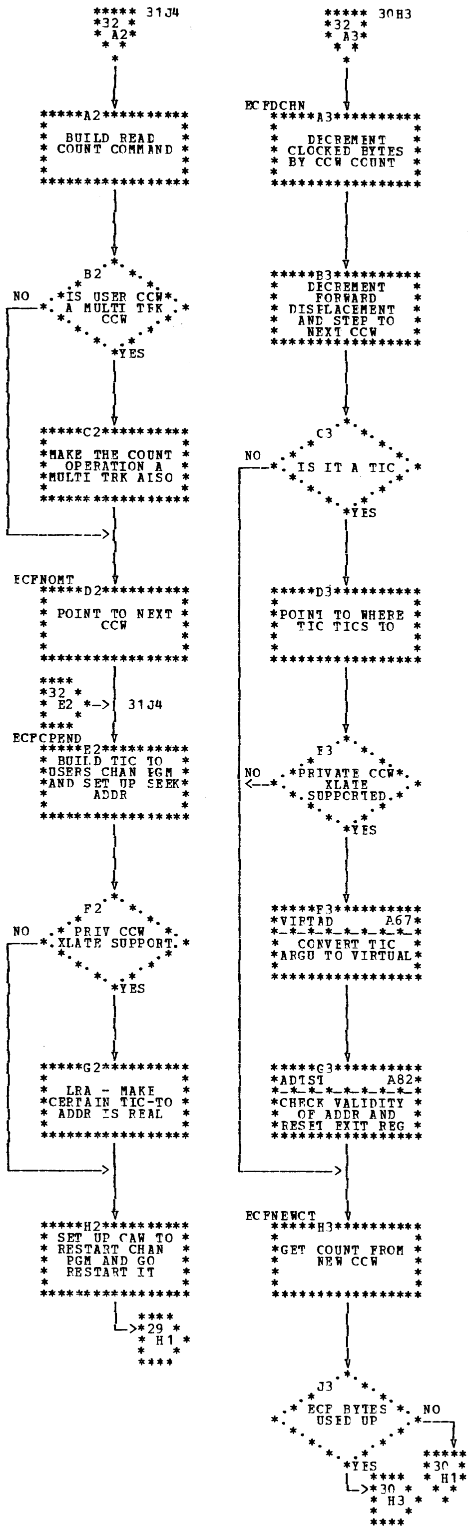


Chart A32. $$A$SUP1 - SGDSK Macrc, Resident Disk ERP
Refer to Chart 10.1.

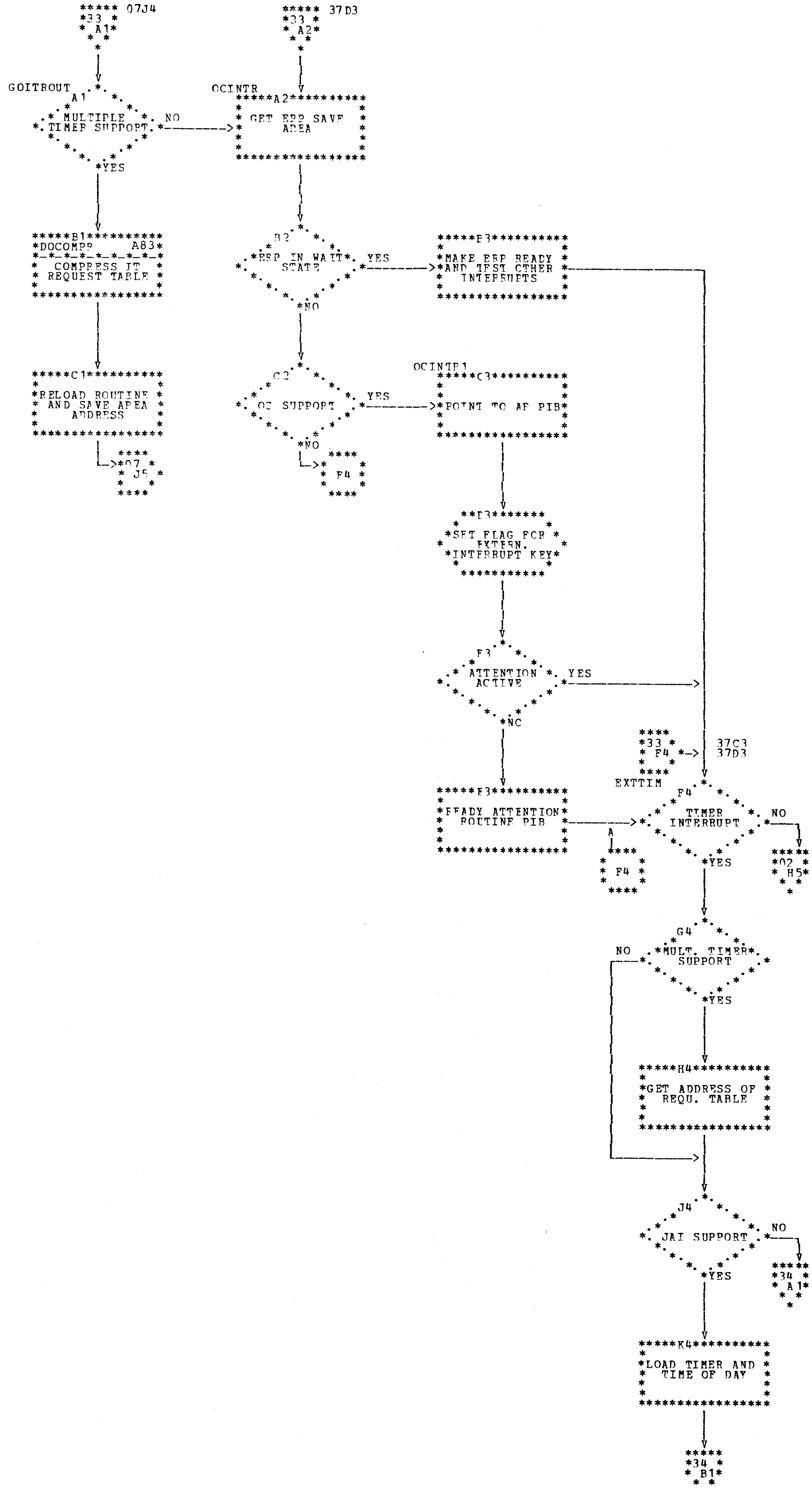


Chart A33. $$A$SUP1 - FOPT Macro, Interval Timer

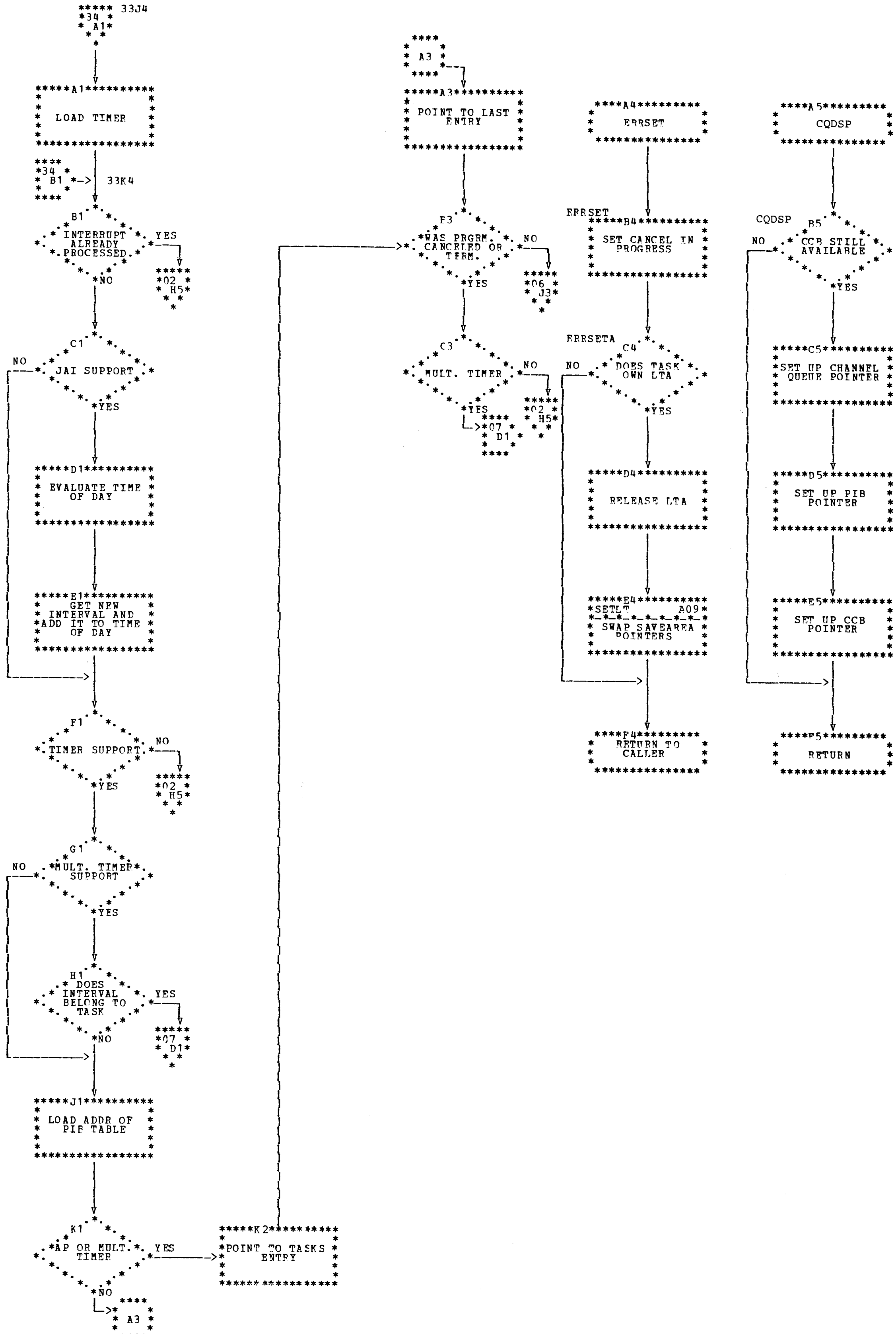


Chart A34. $$A$SUP1 - FOPT Macro, Interval Timer

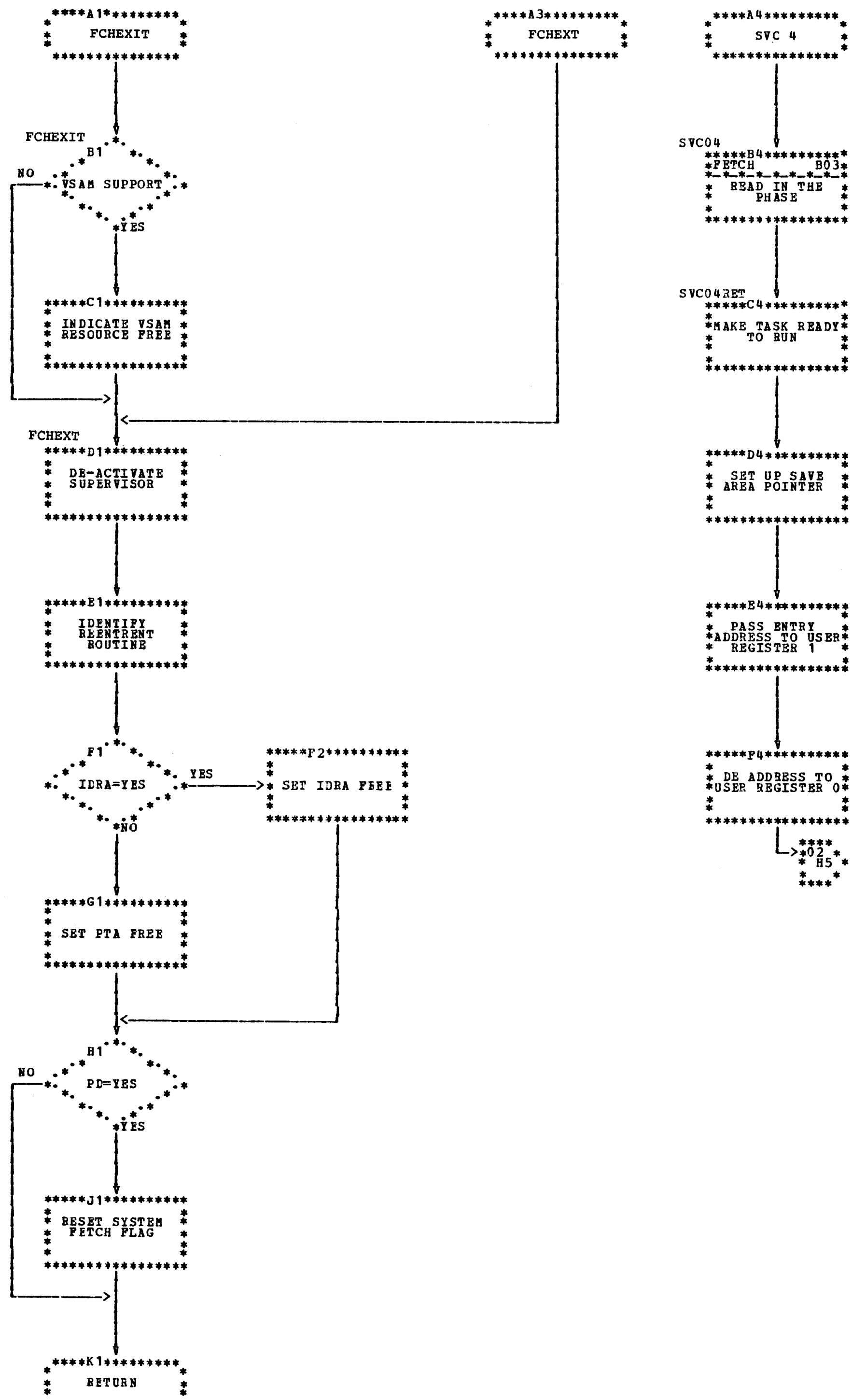


Chart A35. $$A$SUP1 - FOPT Macro, Fetch Exit, Load Phase (SVC 4)

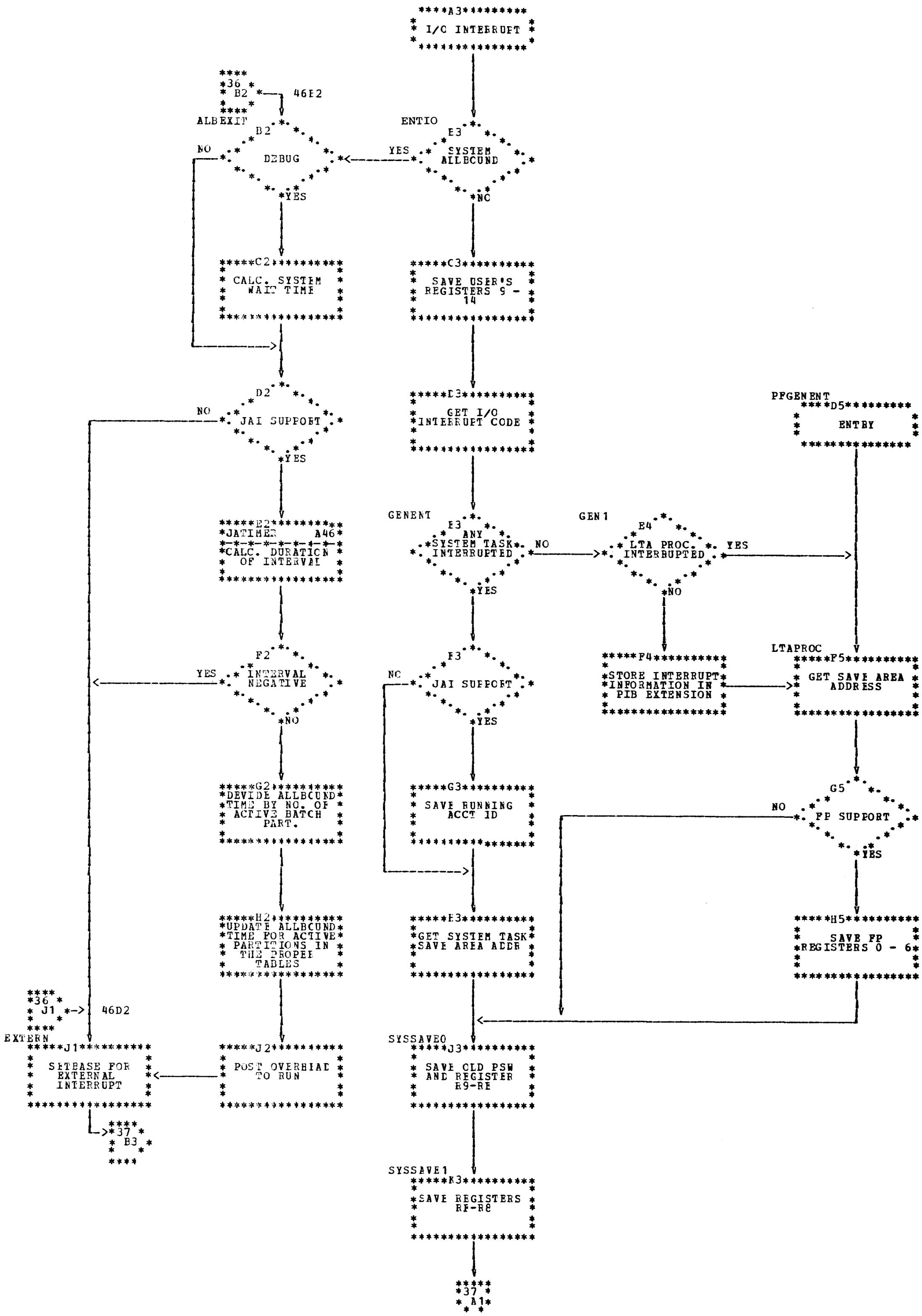


Chart A36. $$A$SUP1 - FOPT Macro, General Entry Routine

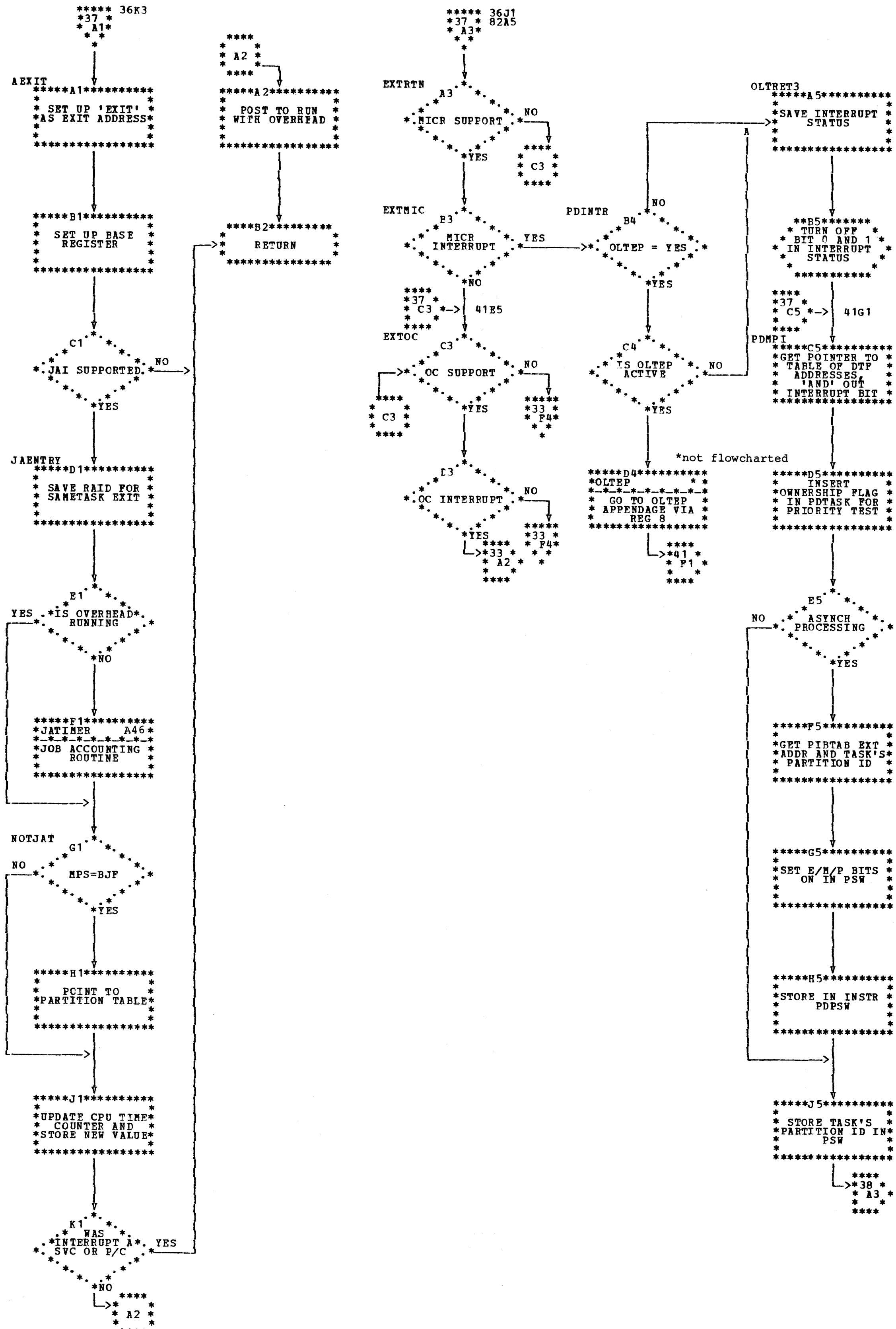


Chart A37. $$A$SUP1 - FOPT Macro, General Entry Routine
SMICR Macro, MICR Interrupt Handler

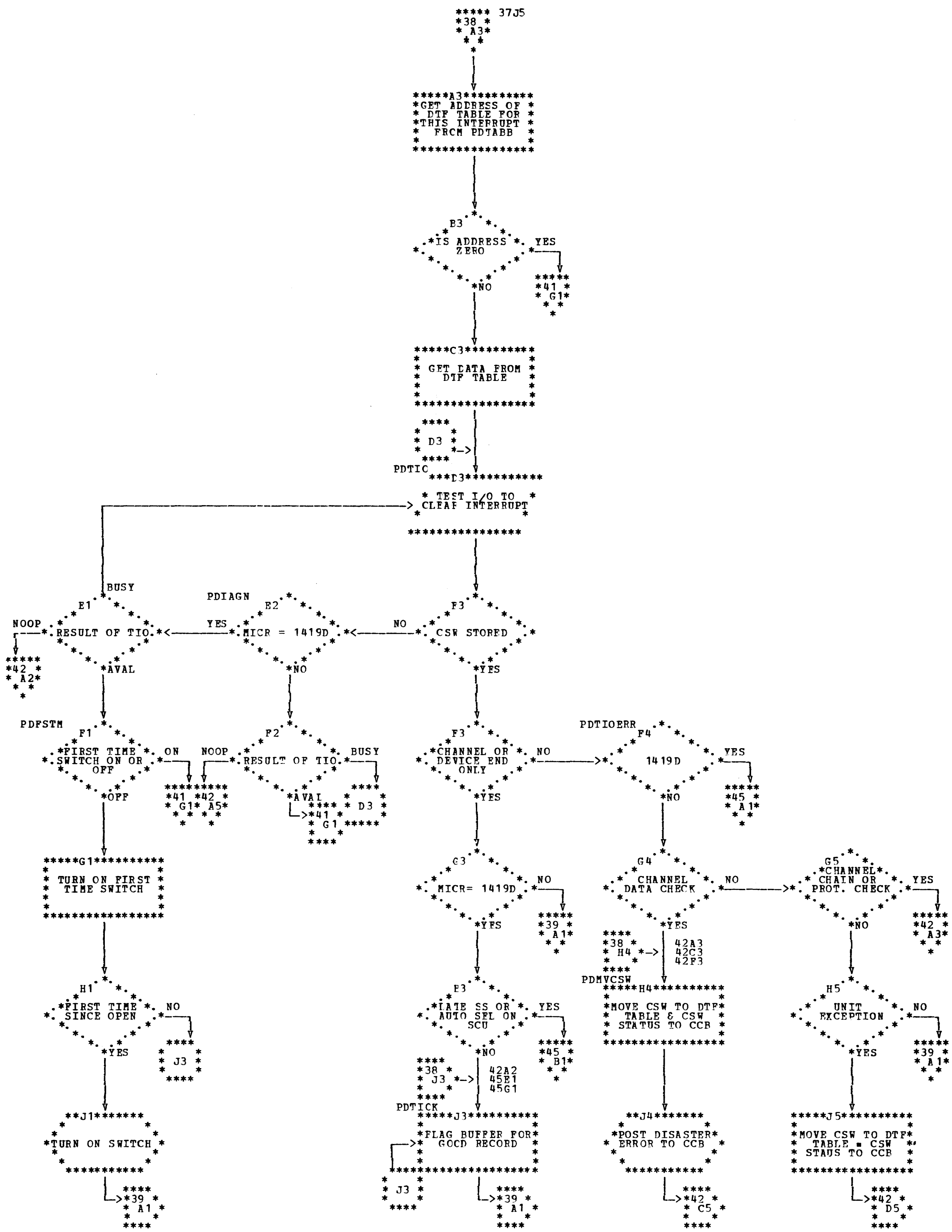


Chart A38. $$A$SUP1 - SMICR Macrc, MICR Interrupt Handler
Refer to Chart 07.1.

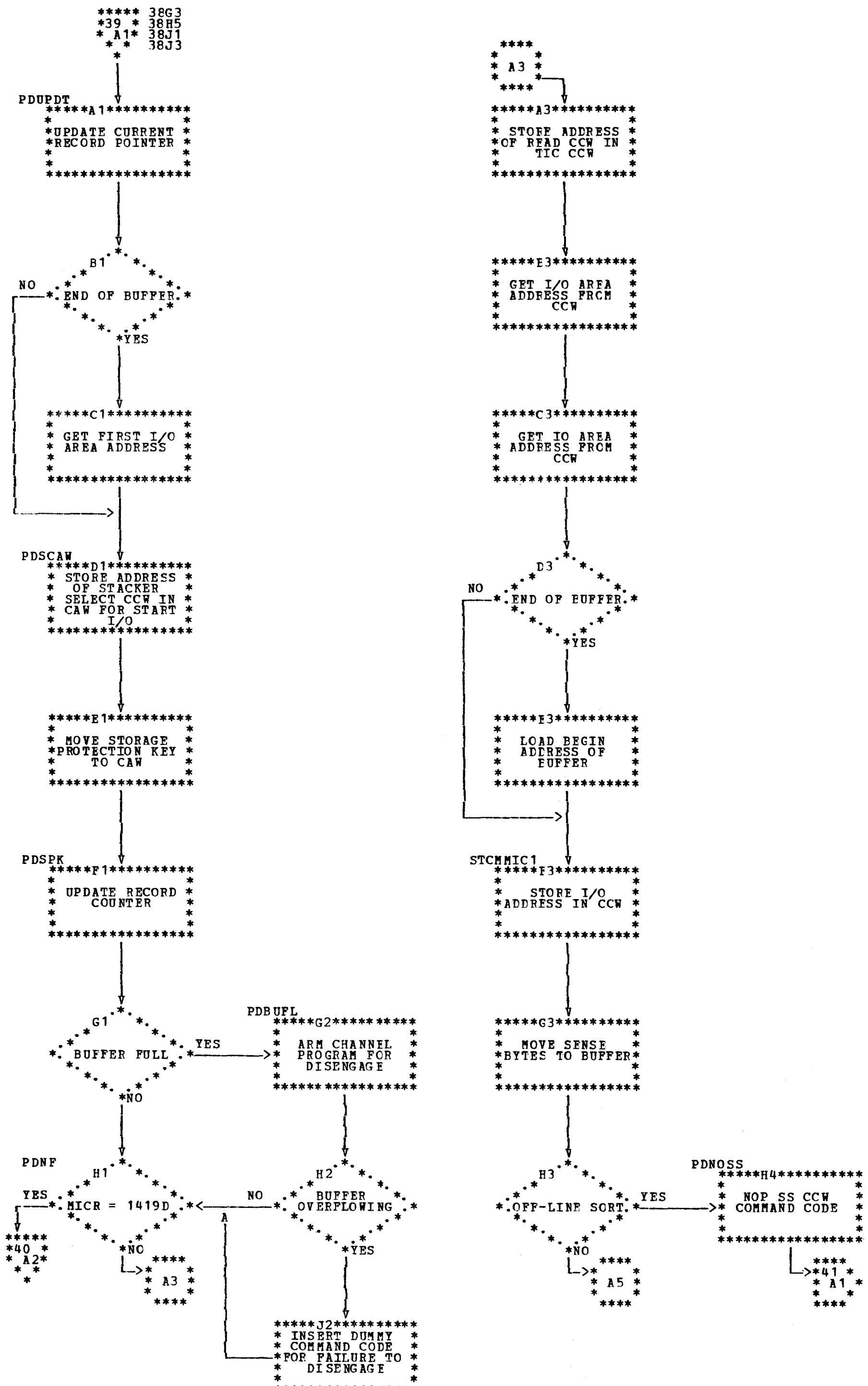


Chart A39. $$A$SUP1 - SMICR Macro, MICR Interrupt Handler
Refer to Chart 07.1.

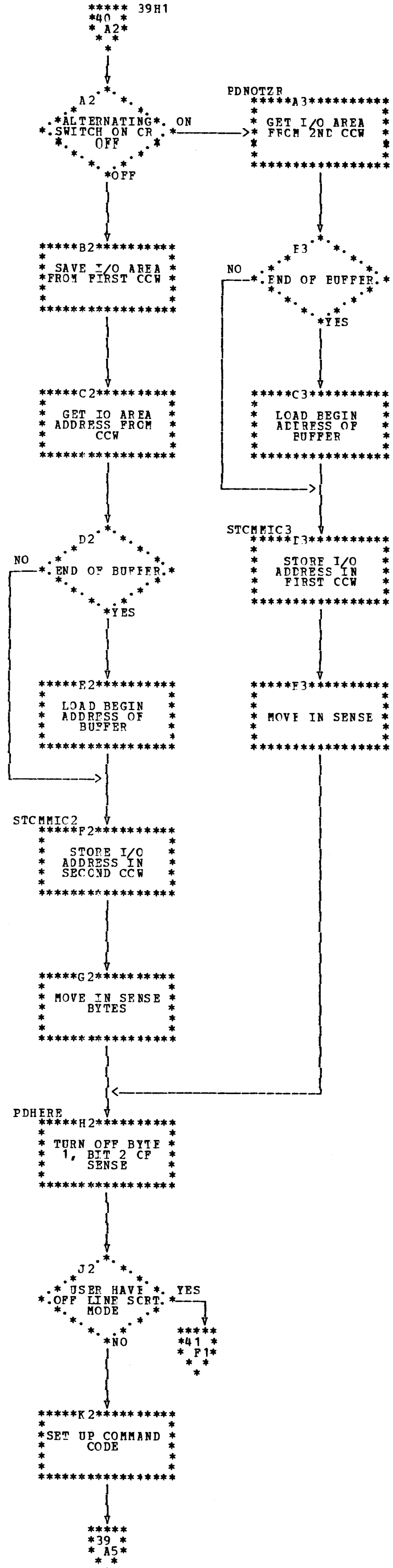


Chart A40. $$A$SUP1 - SMICR Macro, MICR Interrupt Handler
Refer to Chart 07.1.

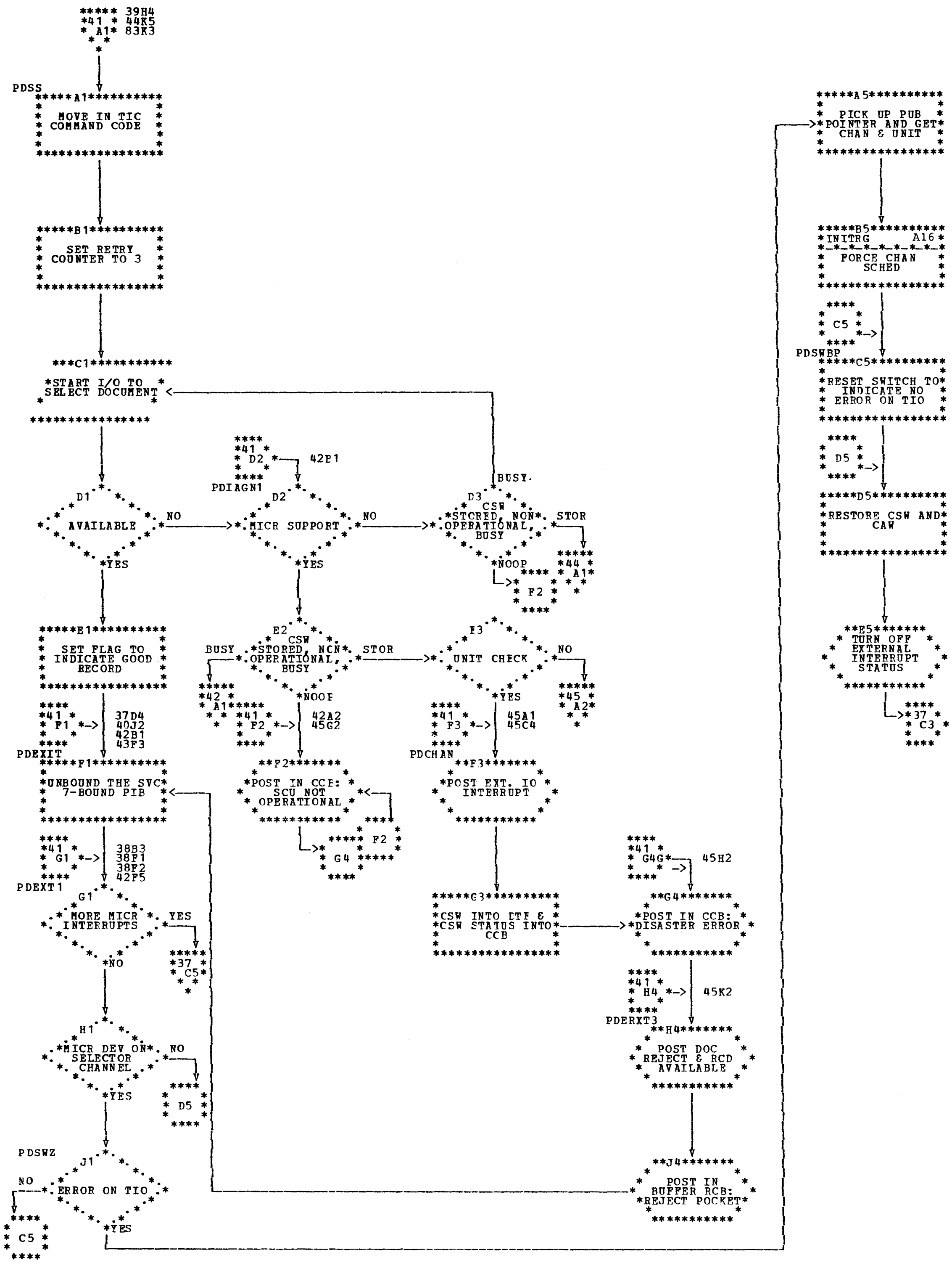


Chart A41. $$A$SUP1 - SMICR Macro, MICR Interrupt Handler
Refer to Chart 07.1.

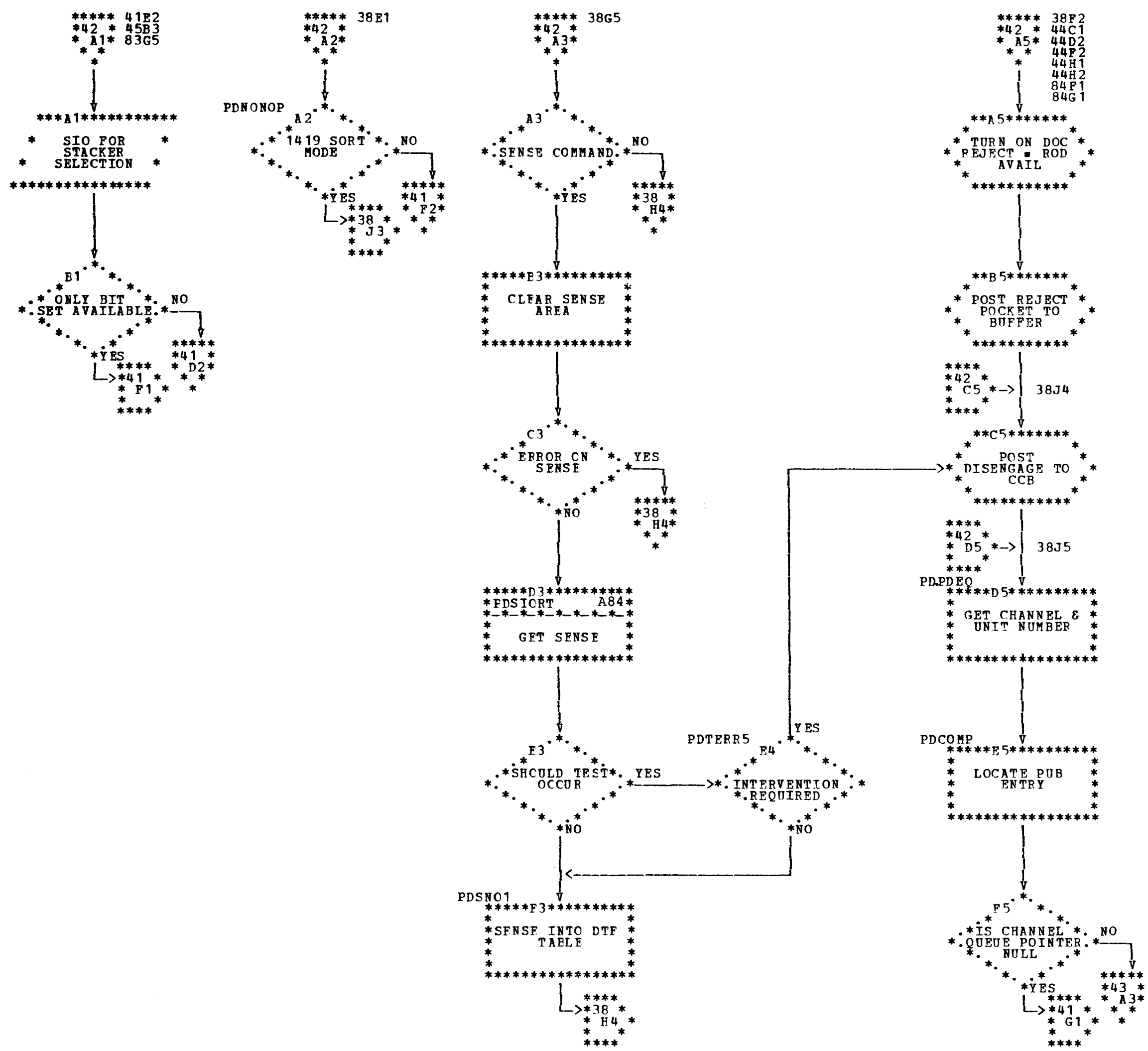


Chart A42. $$A$SUP1 - SMICR Macro, MICR Interrupt Handler
Refer to Chart 07.1.

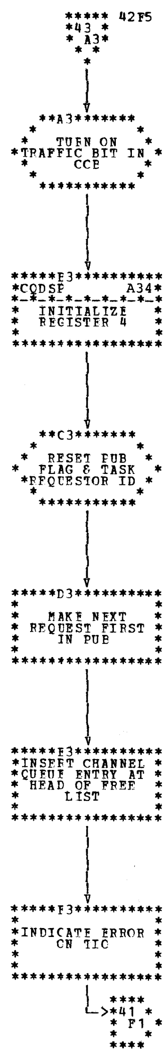


Chart A43. $$A$SUP1 - SMICR Macro, MICR Interrupt Handler
Refer to Chart 07.1.

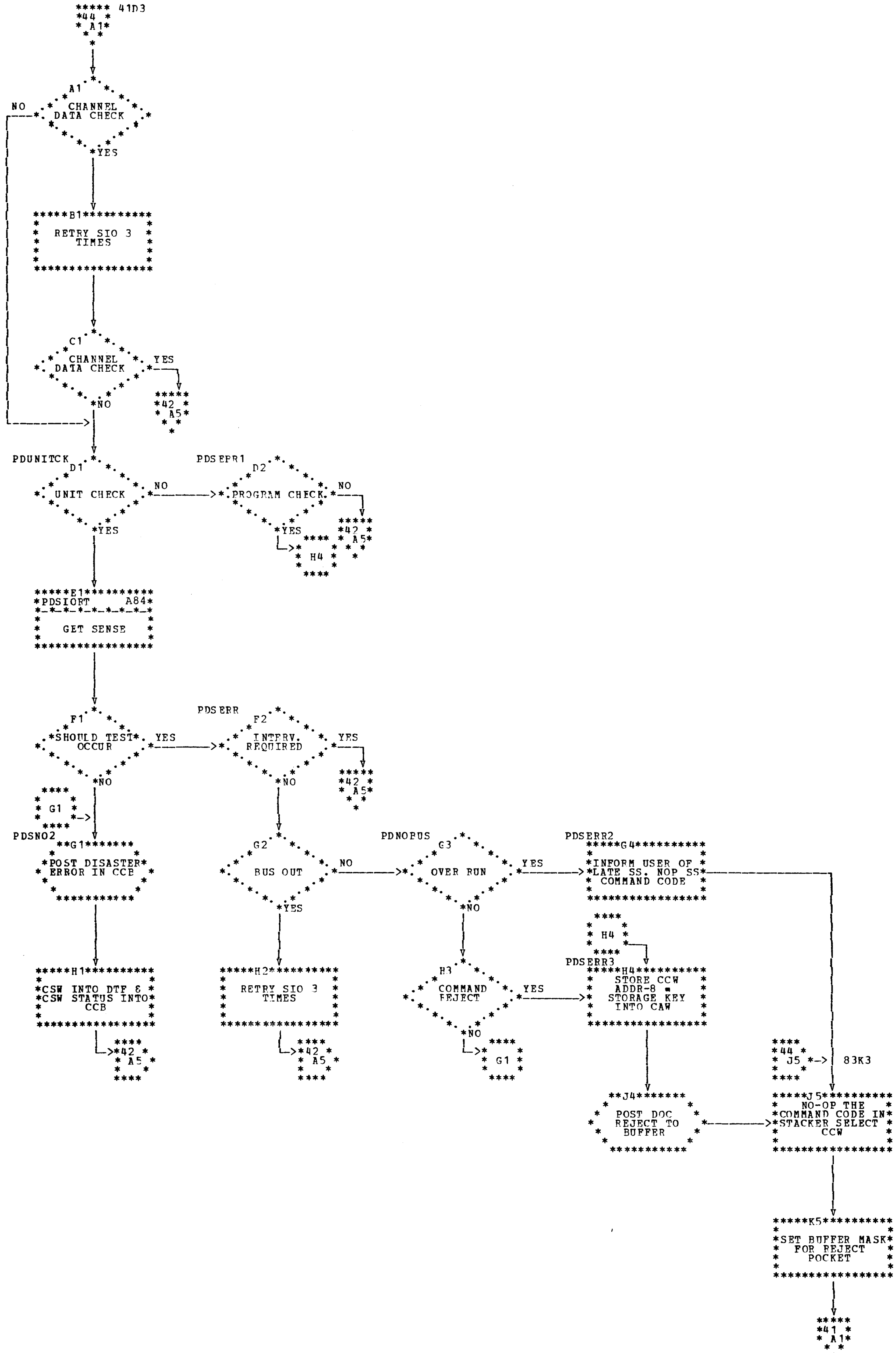


Chart A44. $$A$SUP1 - SMICR Macro, MICR Error Recovery for SIO
Refer to Chart 07.1.

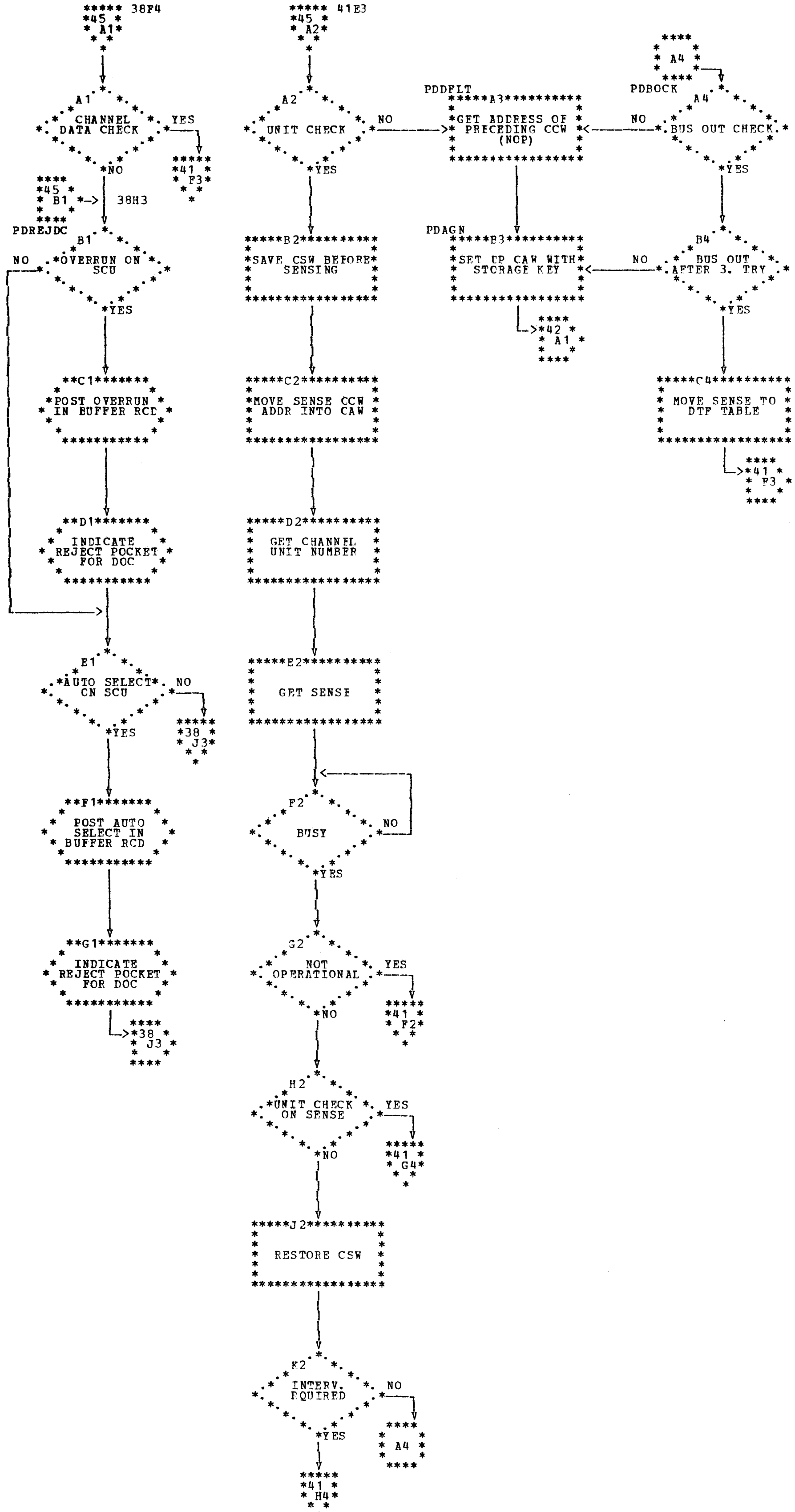


Chart A45. $$A$SUP1 - SMICR Macro, MICR Interrupt Handler
Refer to Chart 07.1.

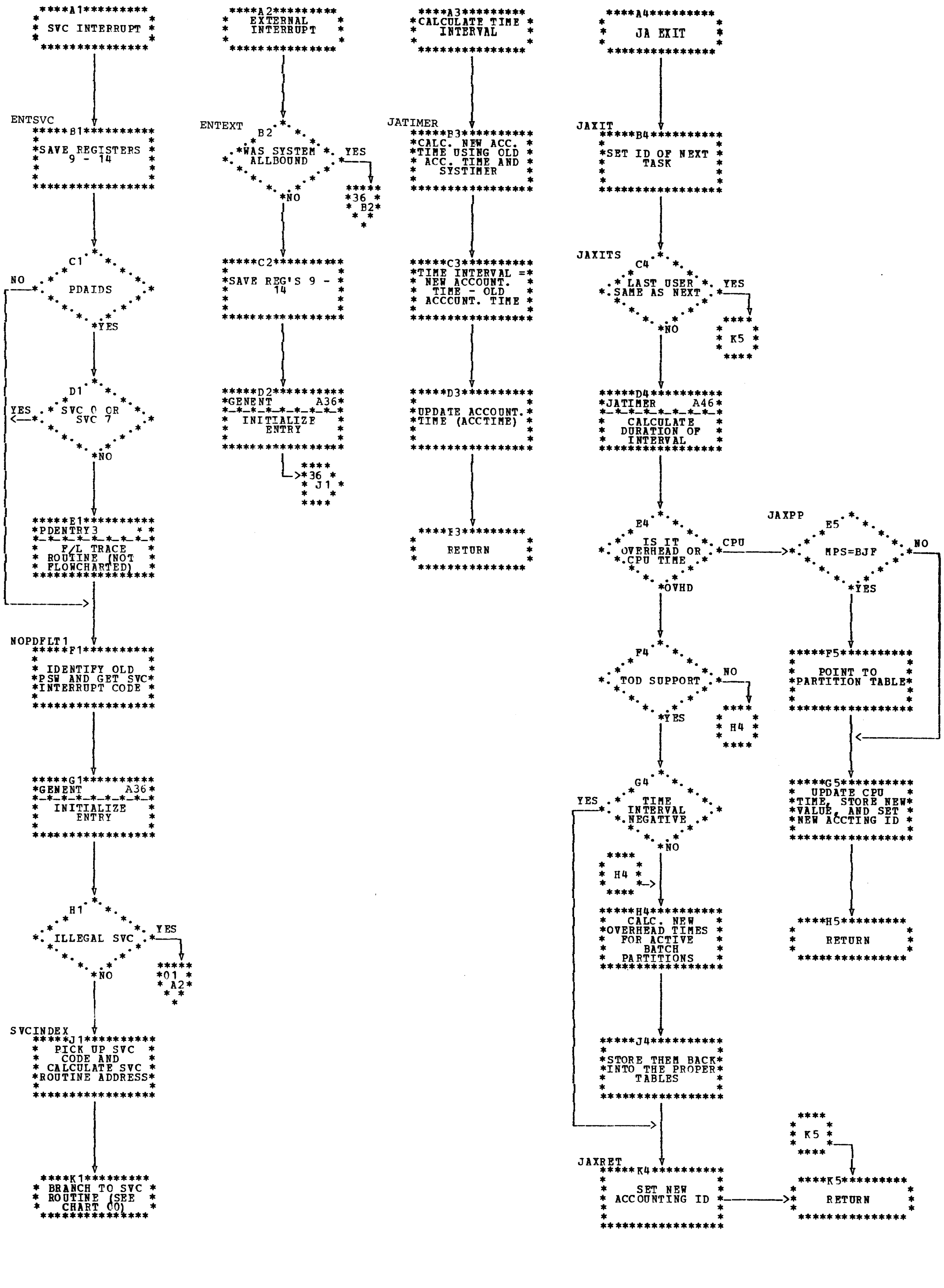


Chart A46. $$A$SUP1 - FOPT Macro, Interrupt Entry Routines, Job Acccunting Routine
Refer to Chart 01.1.

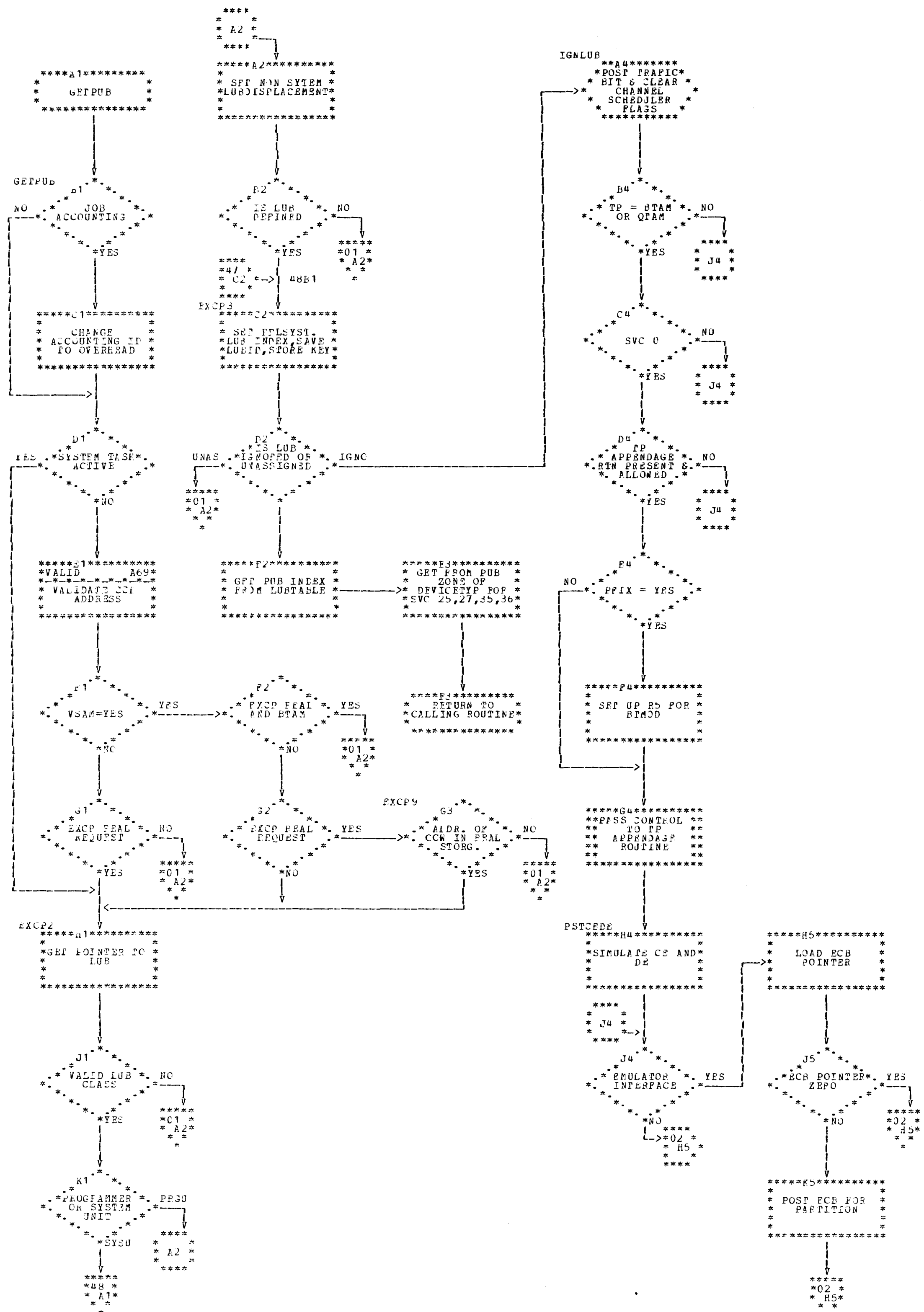


Chart A47. $$A$SUP1 - PIOCS Macro, Calculate PUB Address
Refer to Chart 06.1.

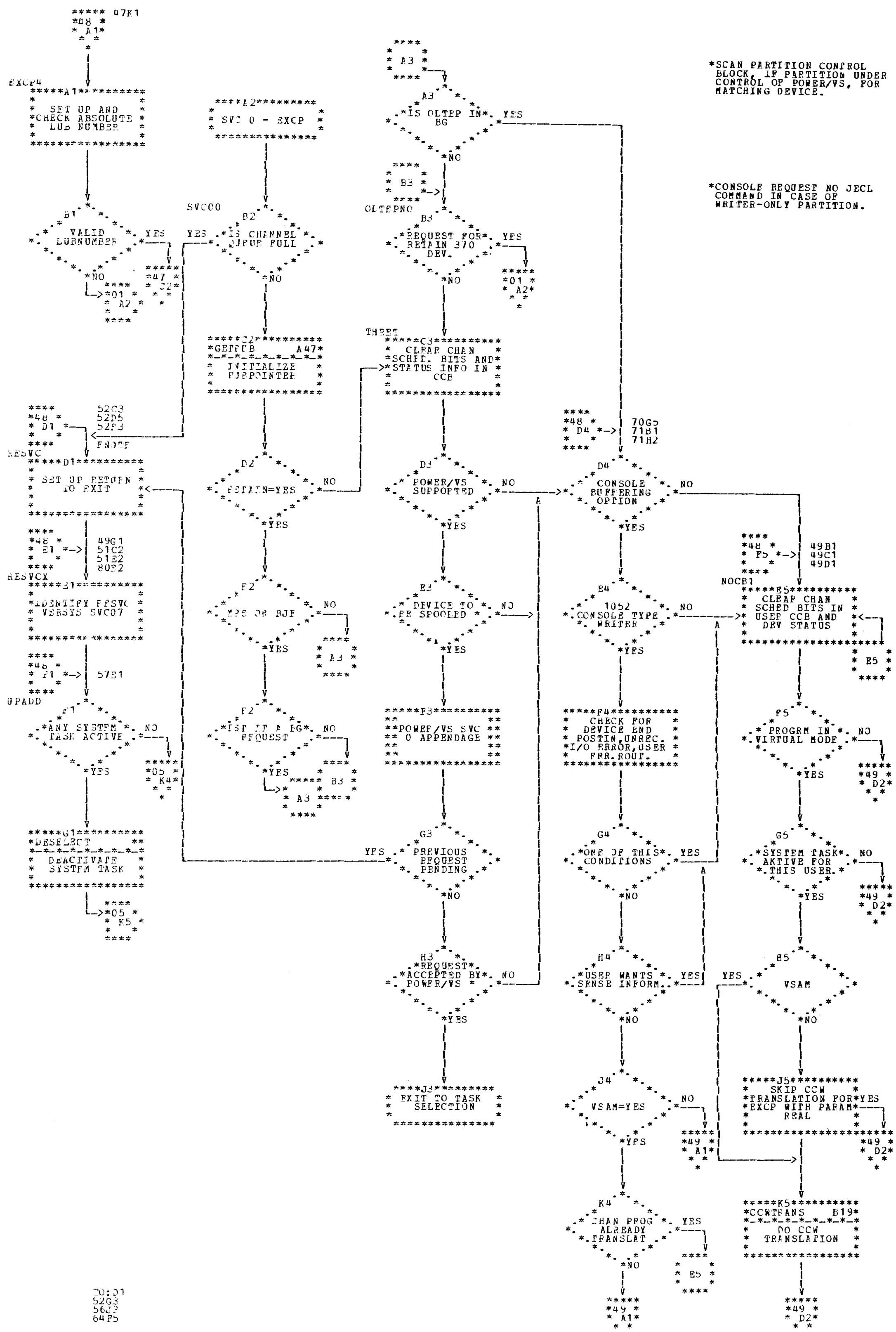


Chart A48. $$A$SUP1 - PIOCS Macro, EXCP (SVC 0)
Refer to Chart 06.3.

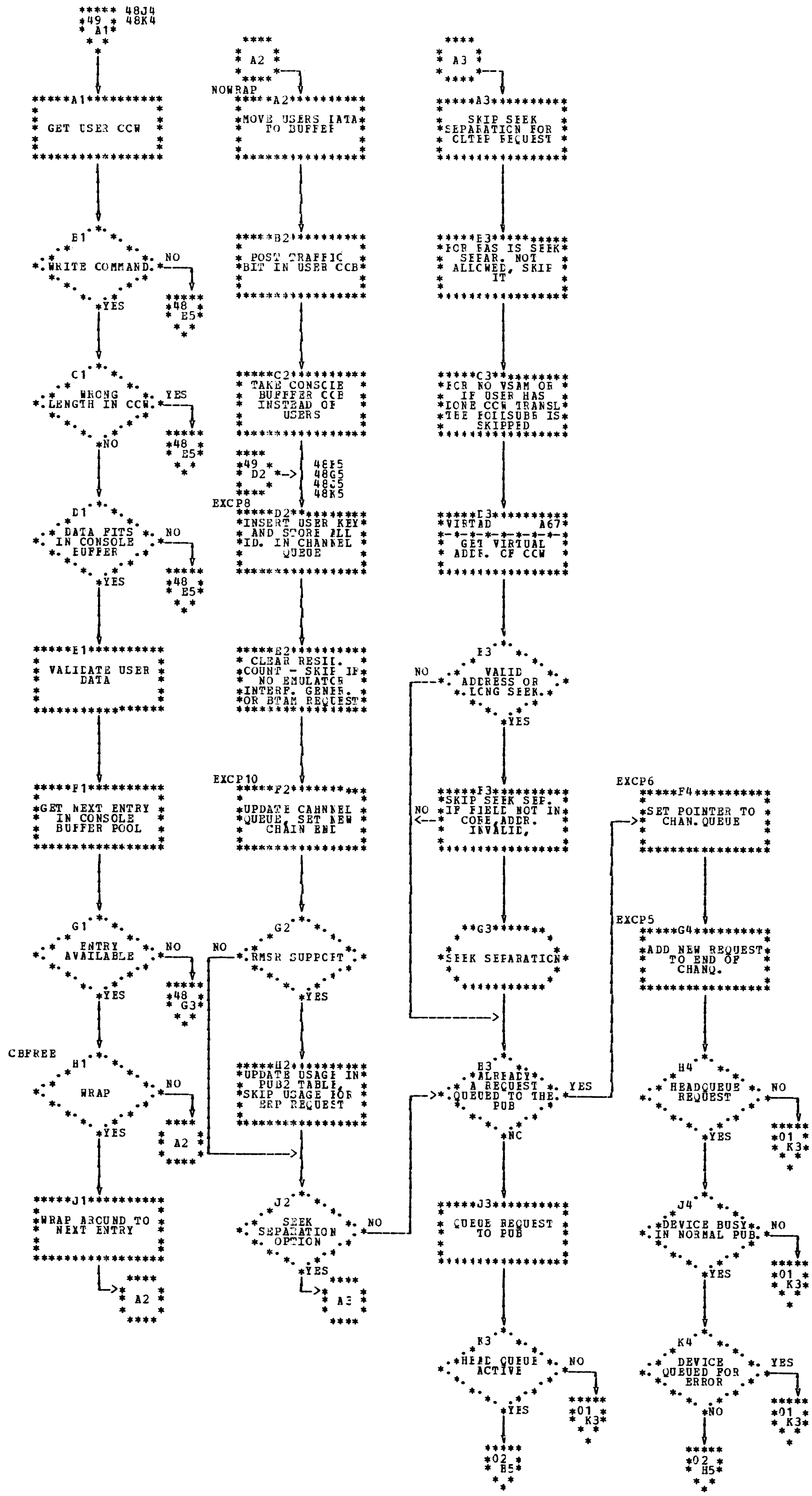


Chart A49. $$A$SUP1 - PIOCS Macro, EXCP (SVC 0)
Refer to Chart 06.3.

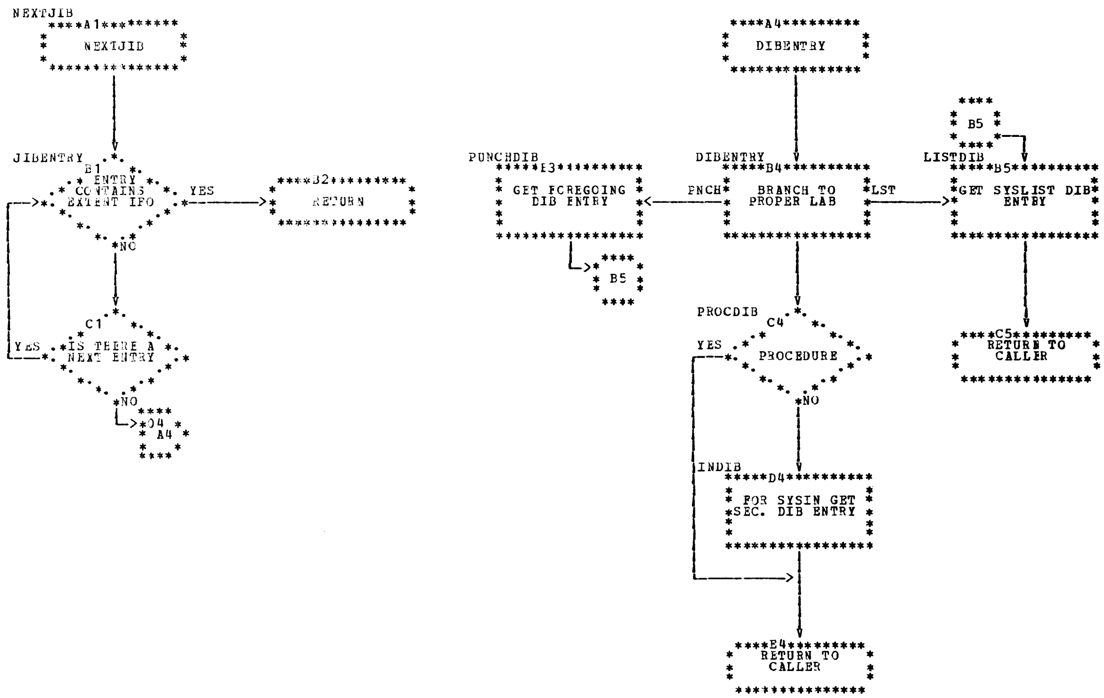


Chart A50. $$A$SUP1 - PIOCS Macro, Subroutines

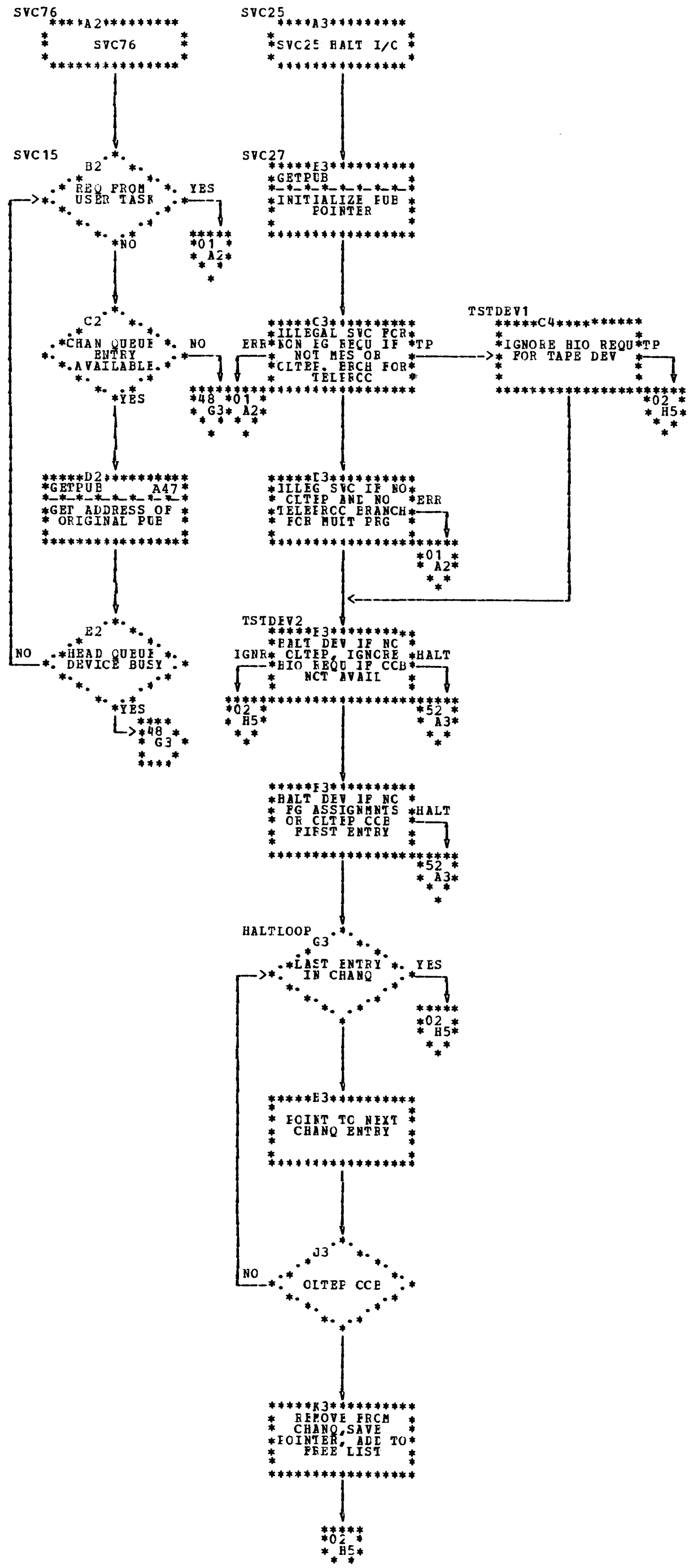


Chart A51. $$A$SUP1 - PIOCS Macro, RMSR Recording EXCP (SVC 76), Halt I/O (SVC 25)

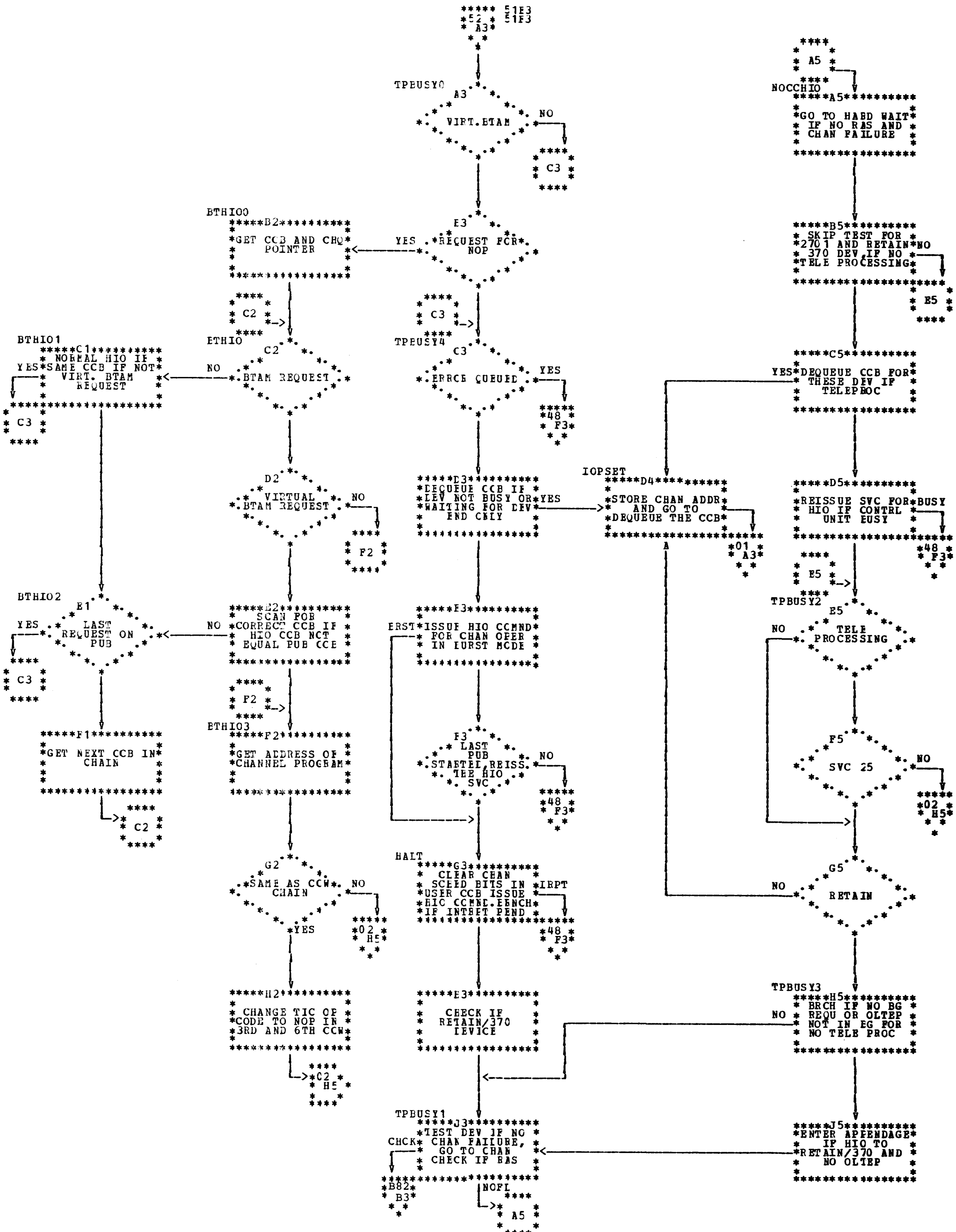


Chart A52. $$A$SUP1 - PIOCS Macro, Halt I/O (SVC 25)

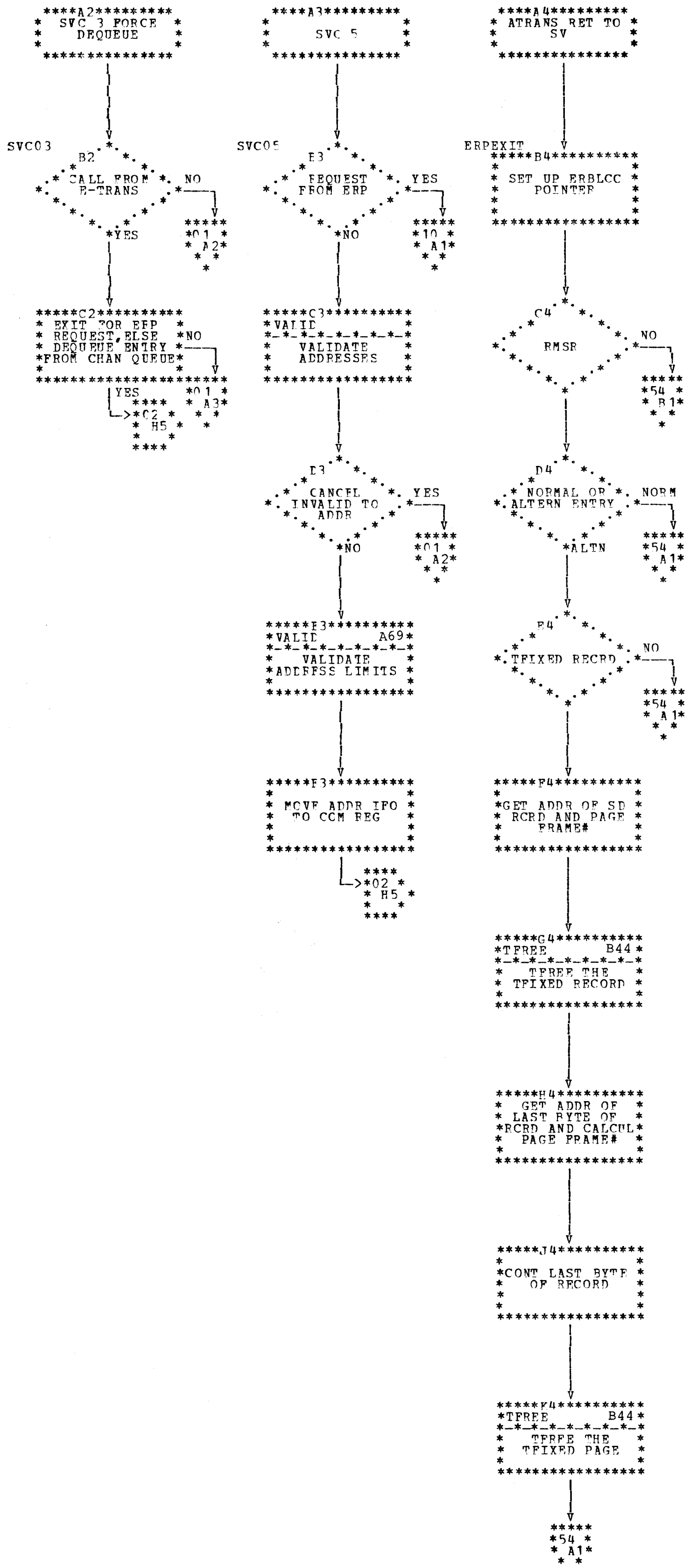


Chart A53. $$A$SUP1 - PIOCS Macro, SVC 3, SVC 5, Return to Supervisor from PTA

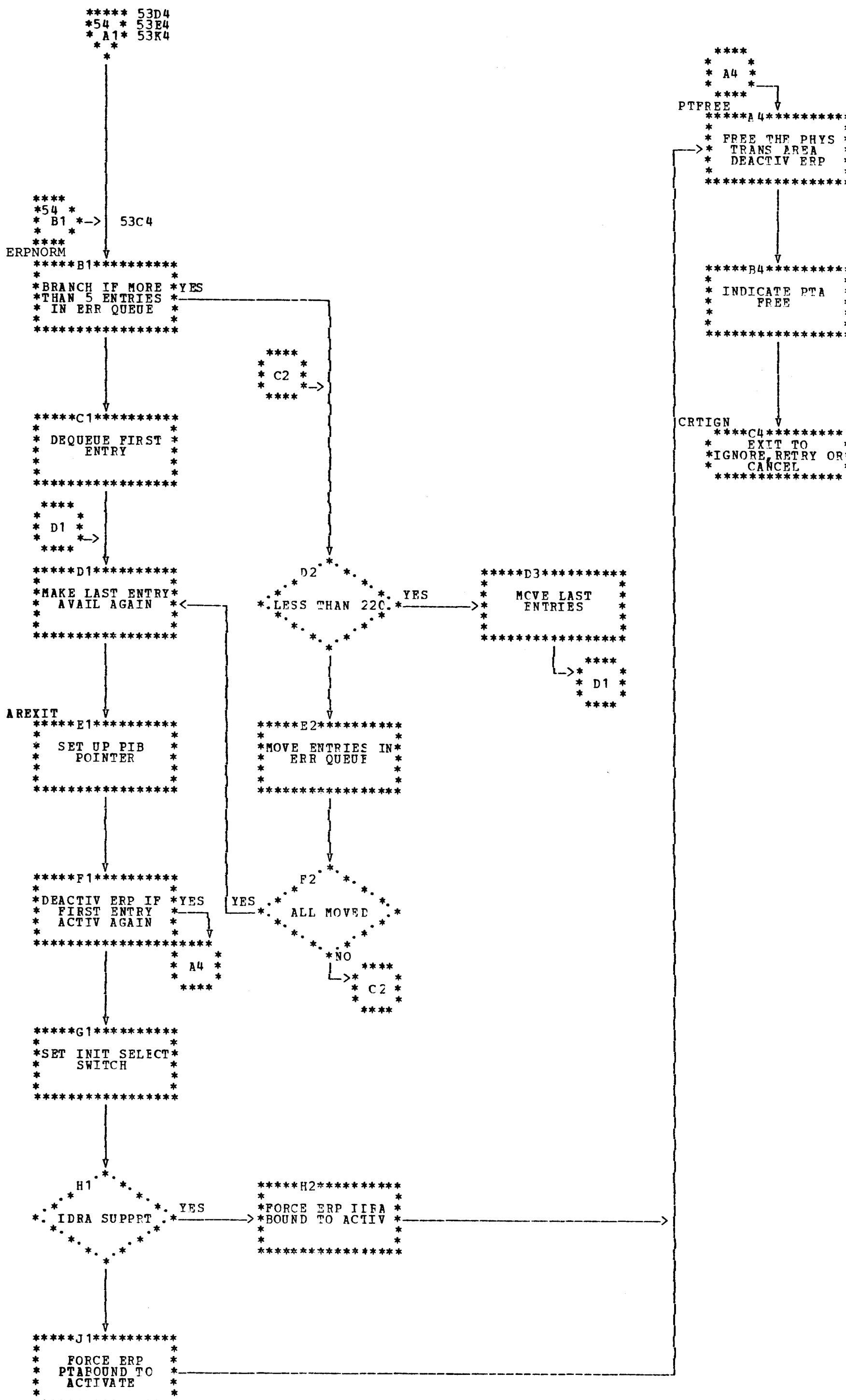


Chart A54. $$A$SUP1 - PIOCS Macro, Return to Supervisor from PTA

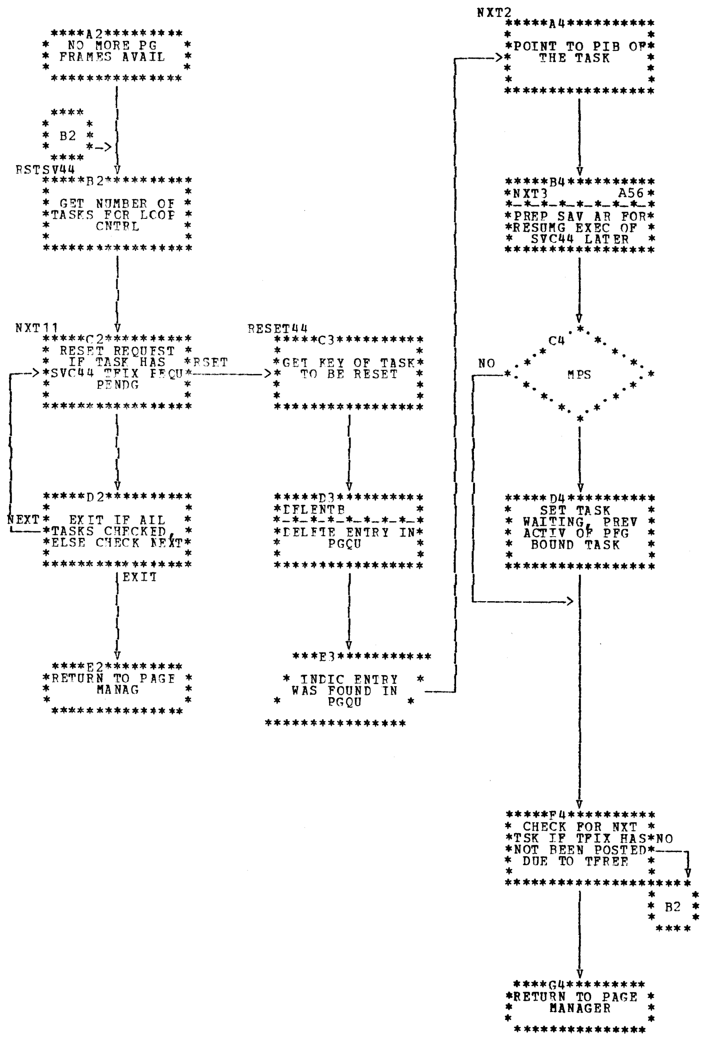


Chart A55. $$A$SUP1 - PIOCS Macro, Reset SVC 44 Request

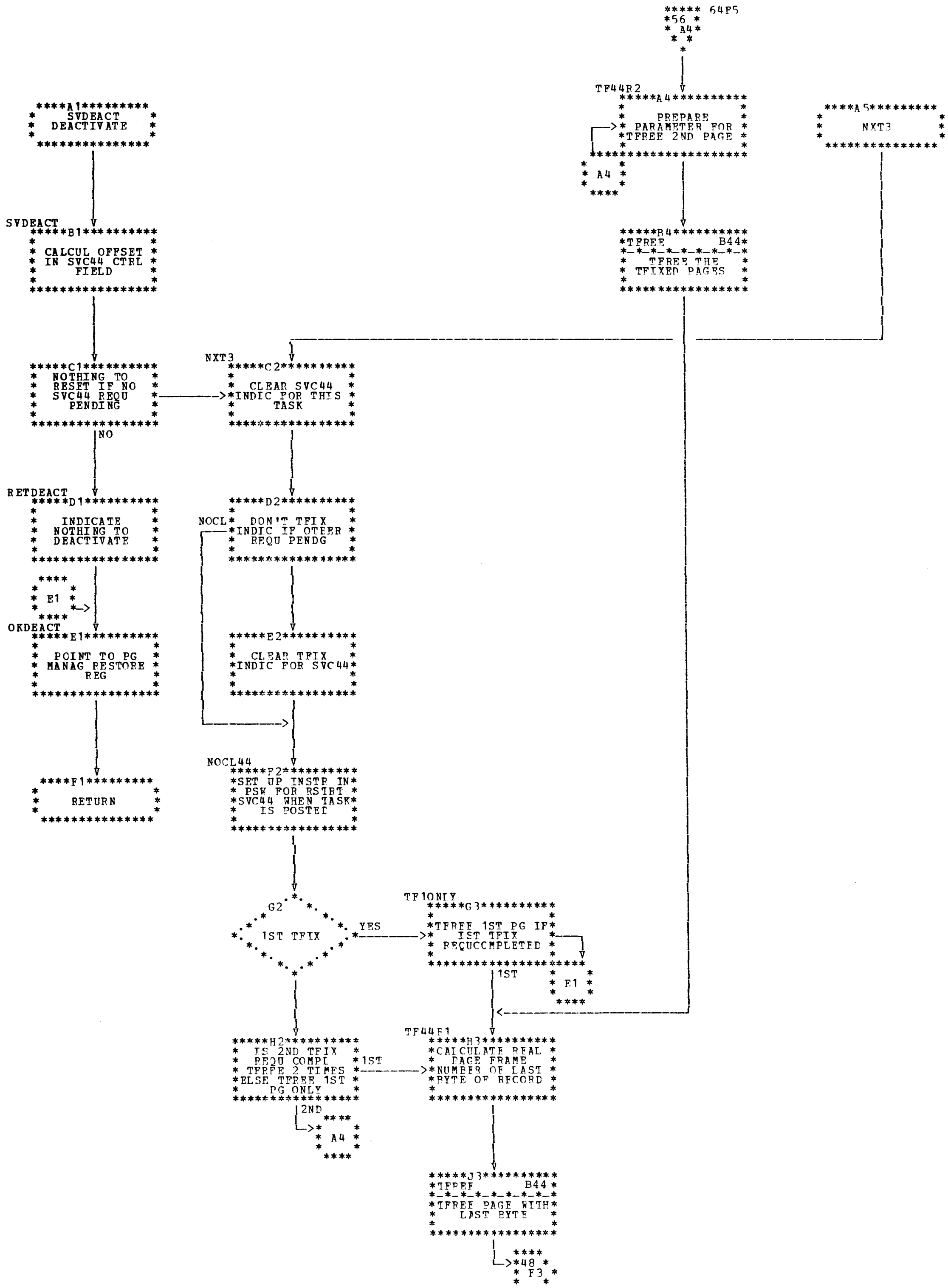


Chart A56. $$A$SUP1 - PIOCS Macrc, Check Whether Deactivated Task has SVC 44 Request with TFIX

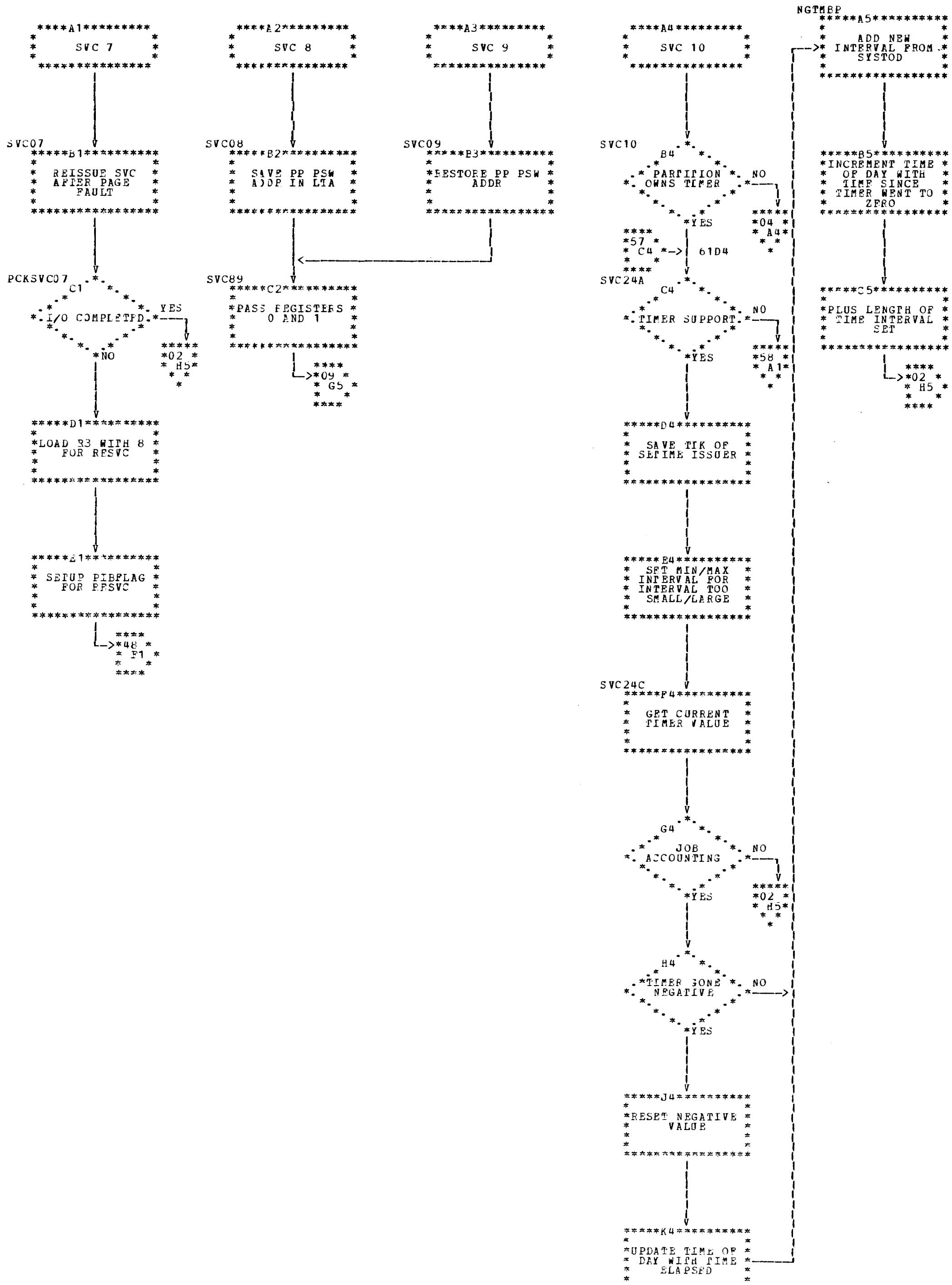


Chart A57. $$A$SUP1 - SGSVC Macro, Entry for SVCs 7, 8, 9, and 10
Refer to Chart 05.1.

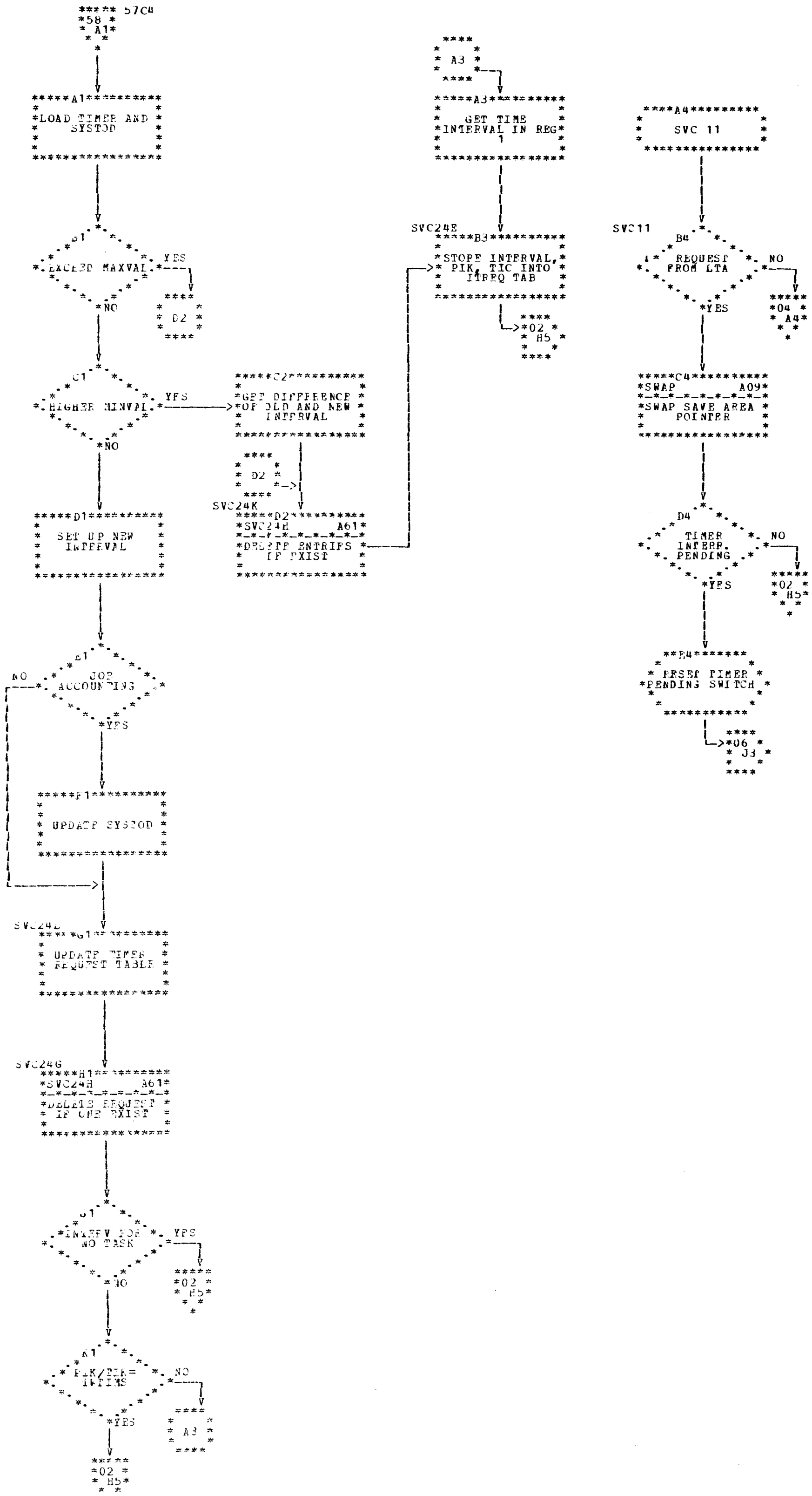


Chart A58. $$A$SUP1 - SGSVC Macro, Set Timer Interval, Provide Linkage (SVC 24)
Refer to Chart 05.1.

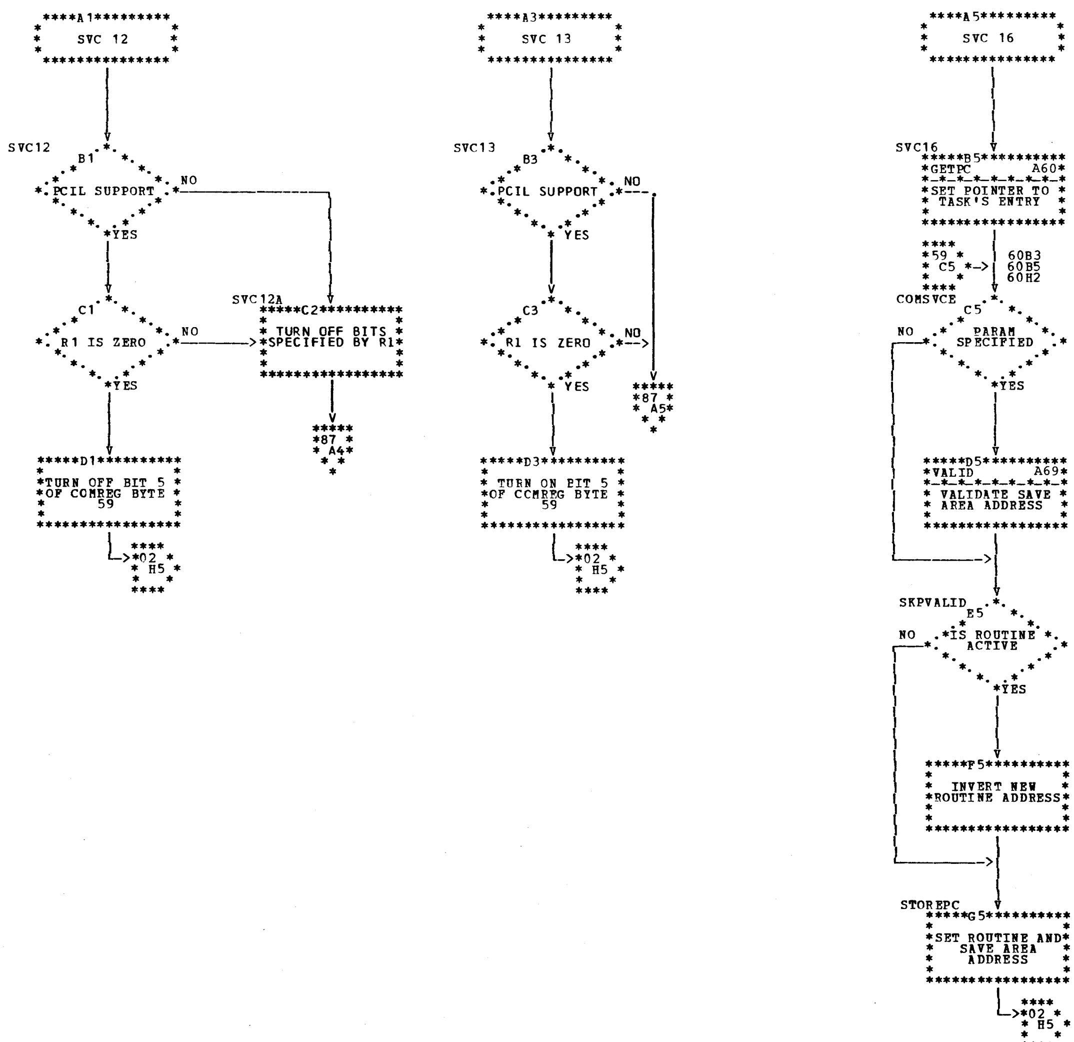


Chart A59. $$A$SUP1 - SGSVC Macro, Entry for SVCs 12, 13, and 16
Refer to Chart 05.1.

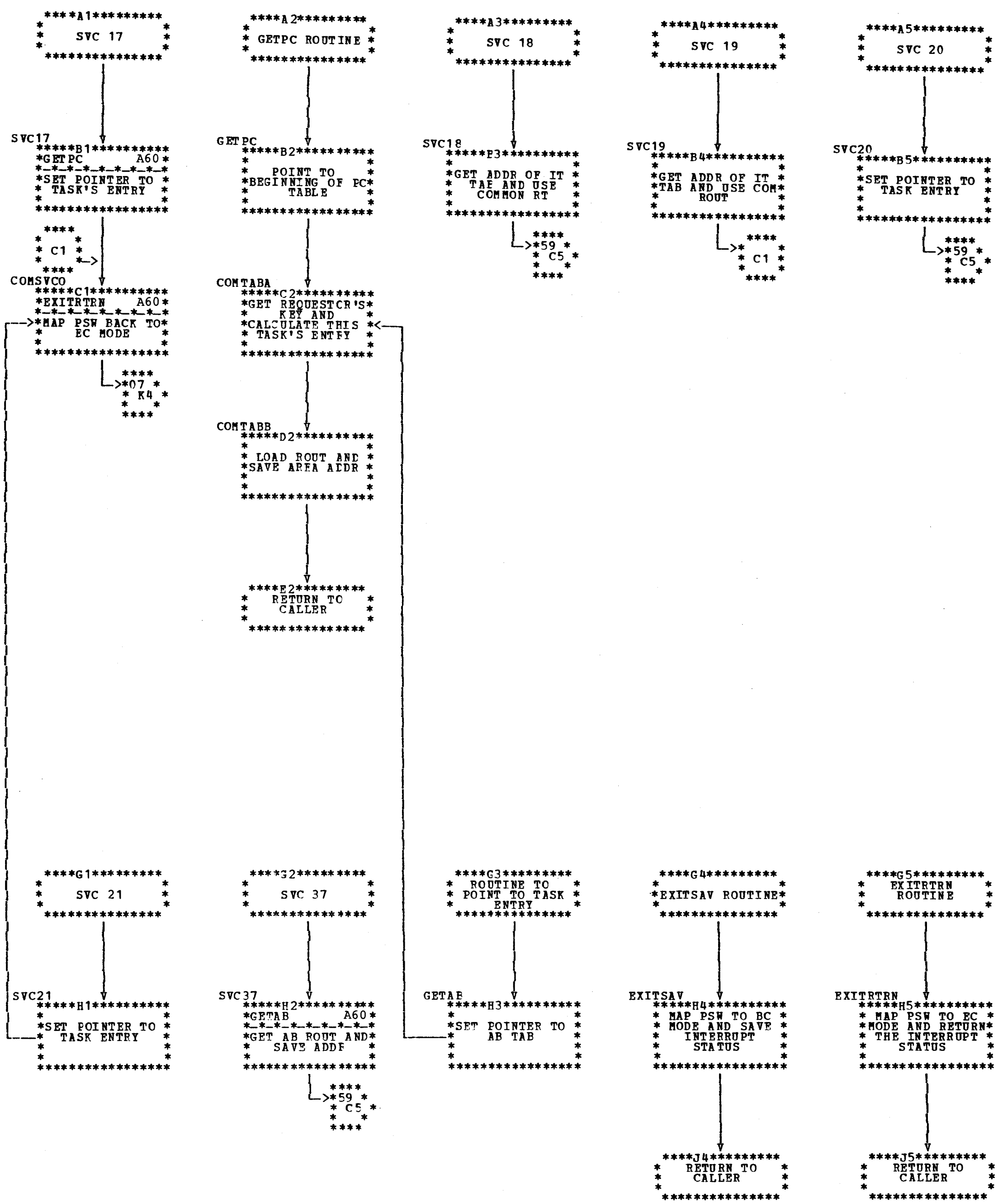


Chart A60. $$A$SUP1 - SGSVC Macro, Entry for SVCs 17-21, 37
Refer to Chart 05.1.

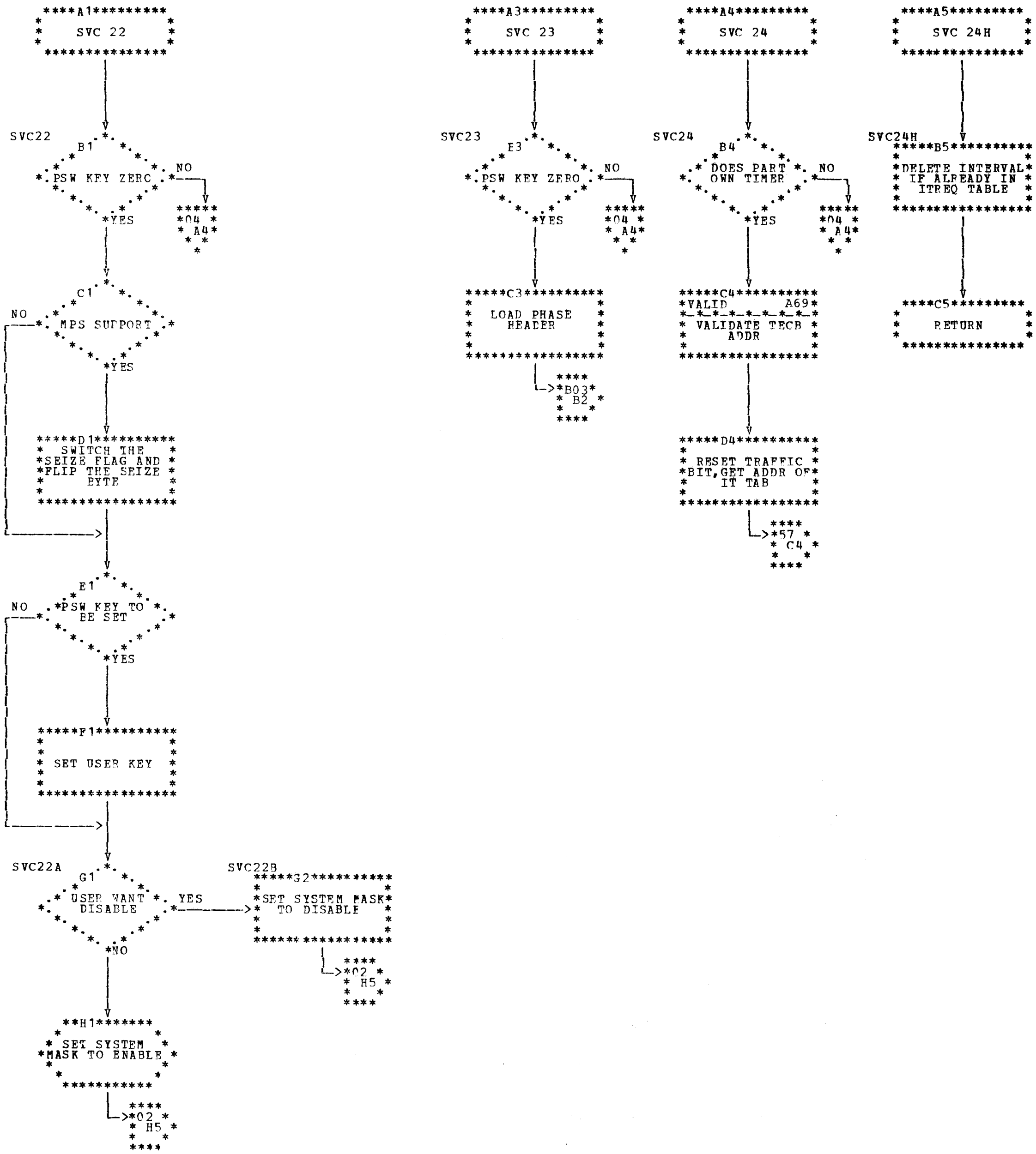


Chart A61. $$A$SUP1 - SGSVC Macro, Entry for SVCs 22, 23, and 24

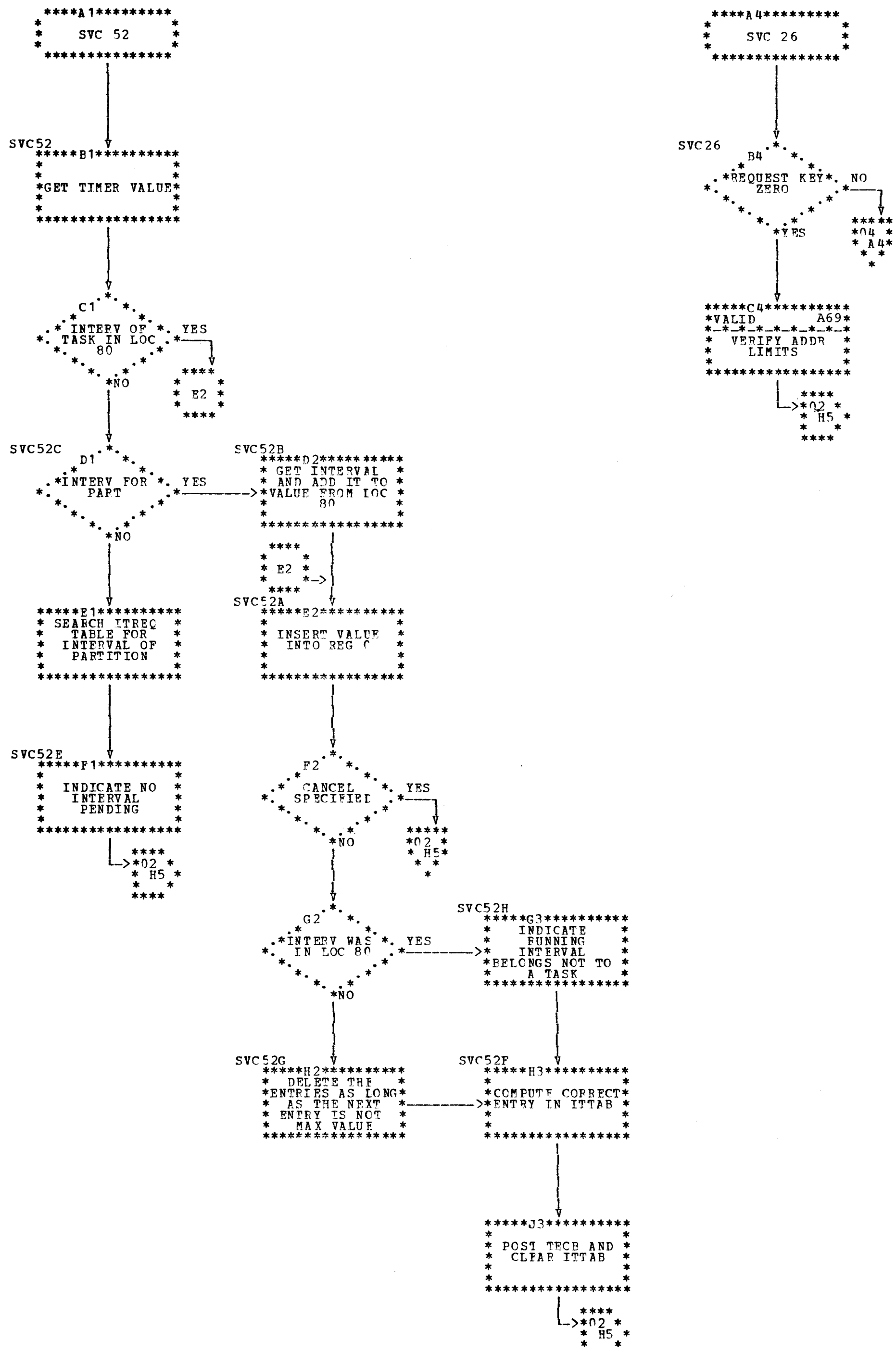


Chart A62. $$A$SUP1 - SGSVC Macrc, Entry fcr SVCs 26 and 52

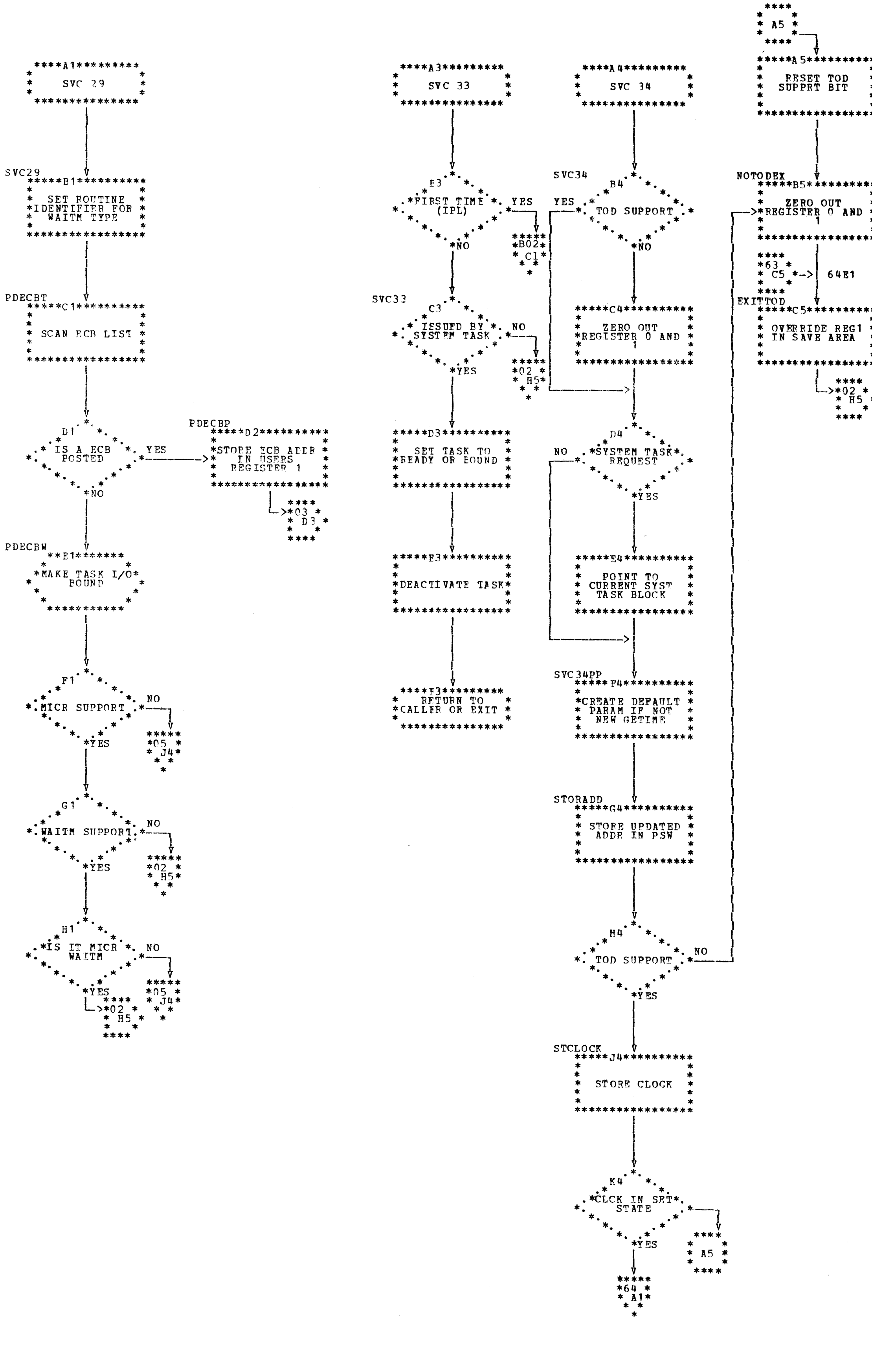


Chart A63. $$A$SUP1 - SGSVC Macrc, Entry fcr SVCs 29, 33, and 34

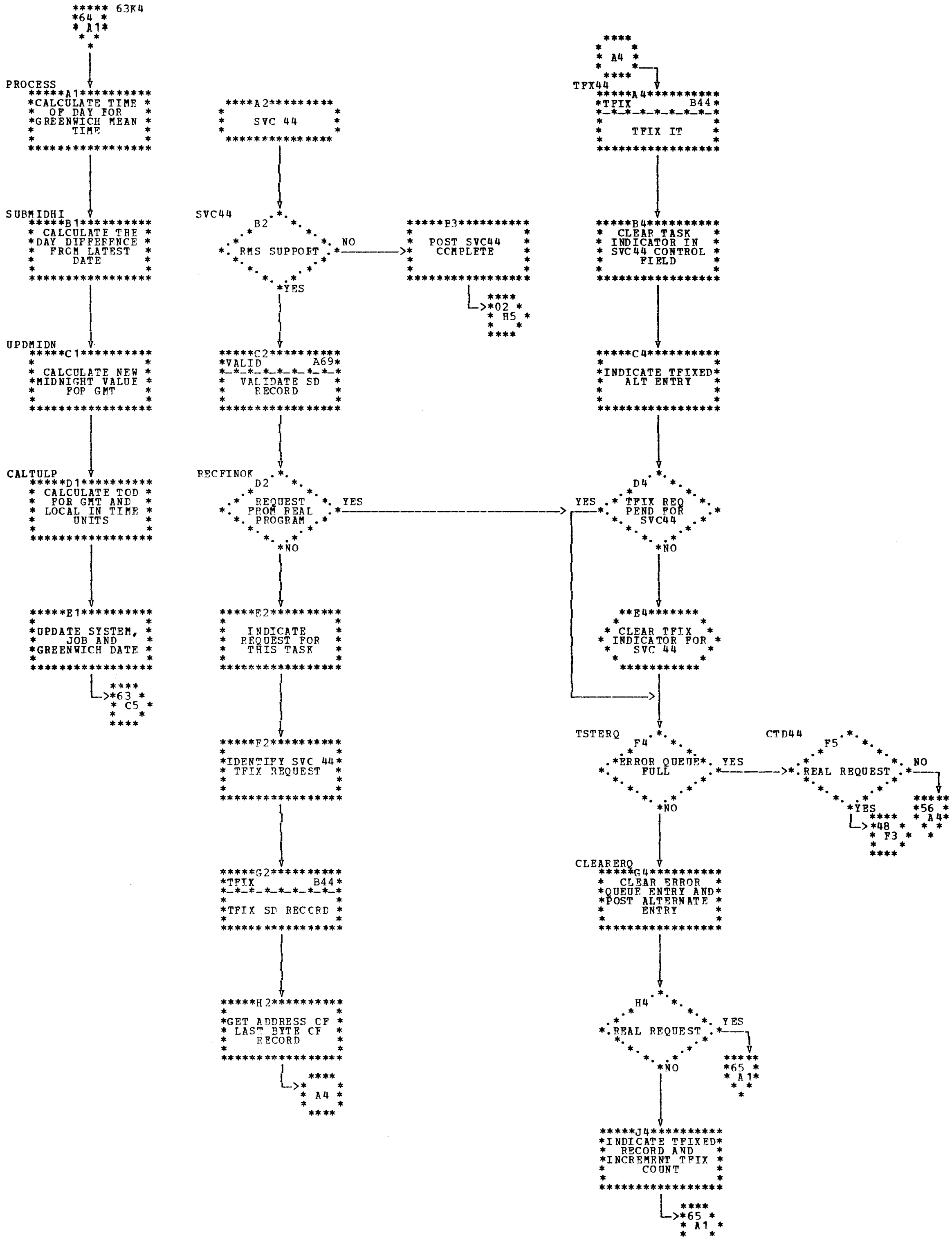


Chart A64. $$A$SUP1 - SGSVC Macro, Entry for SVC 44

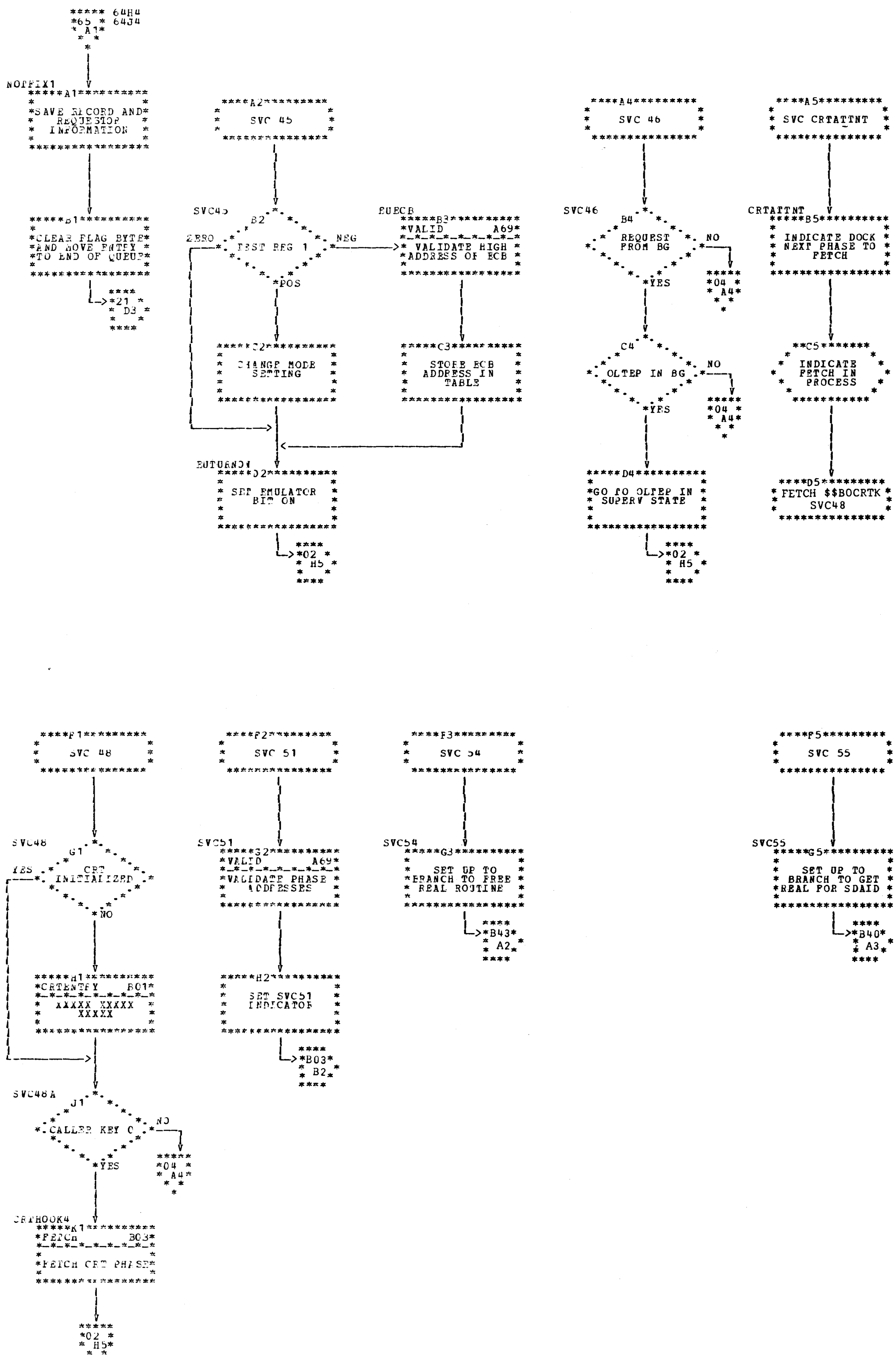


Chart A65. $$A$SUP1 - SGSVC Macro, Entry for SVCs 45, 46, 48, 51, 54, and 56

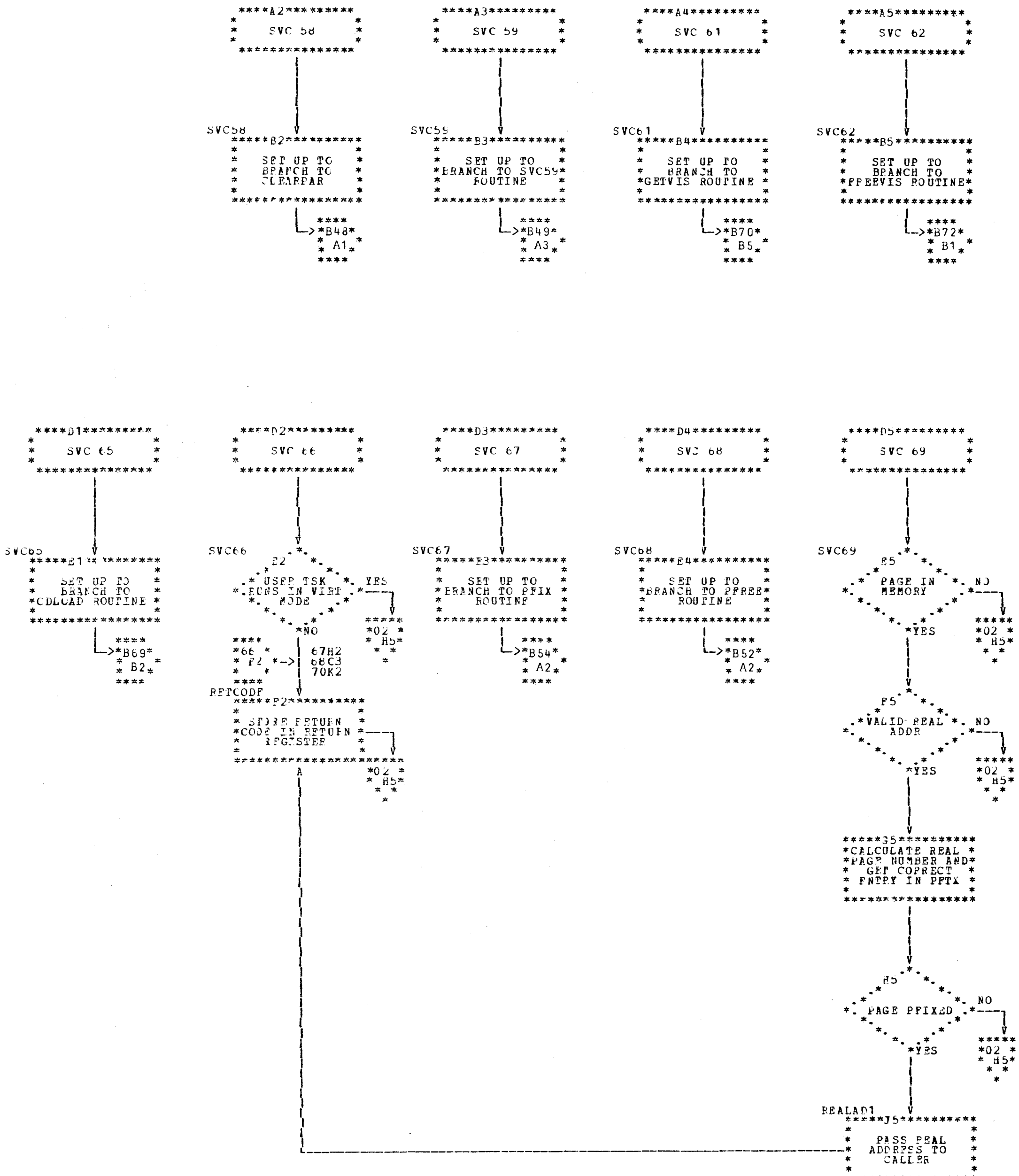


Chart A66. $$A$SUP1 - SGSVC Macro, Entry for SVCs 57-59, 61-62, 65-69

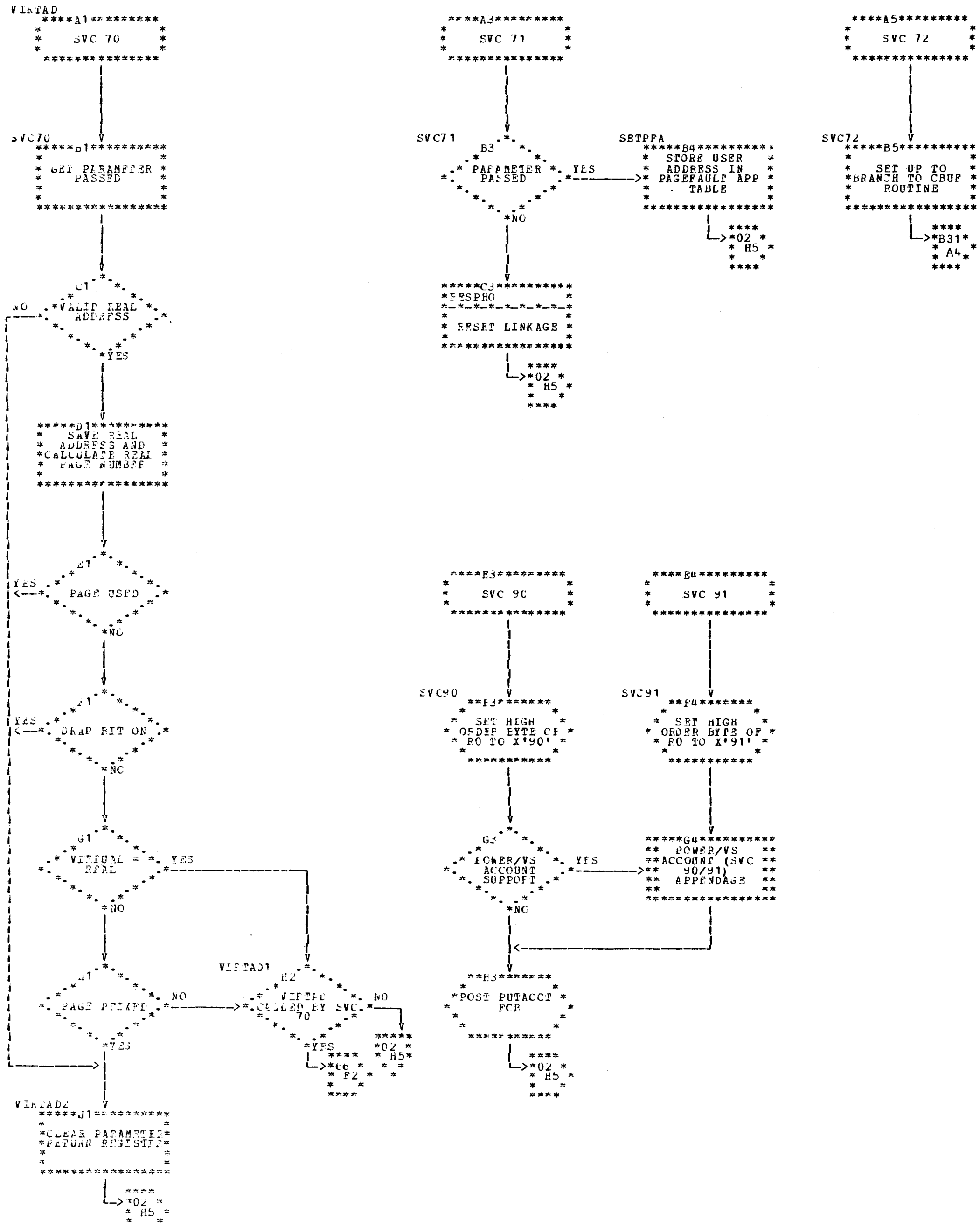


Chart A67. $$A$SUP1 - SGSVC Macro, Entry for SVCs 70-72, 90-91

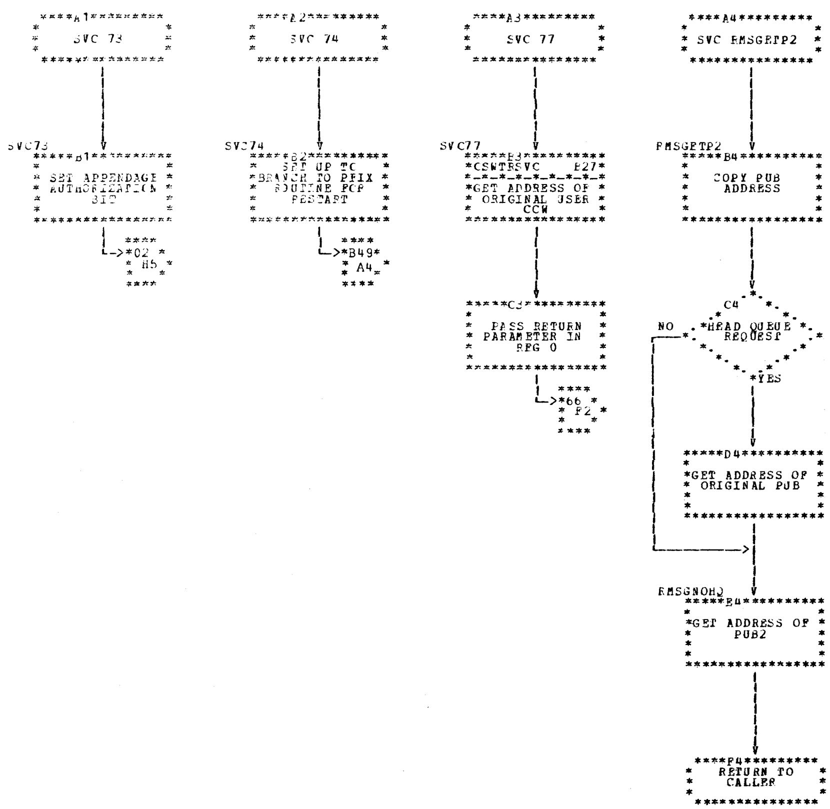


Chart A68. $$A$SUP1 - SGSVC Macrc, Entry for SVCs 73, 74, and 77

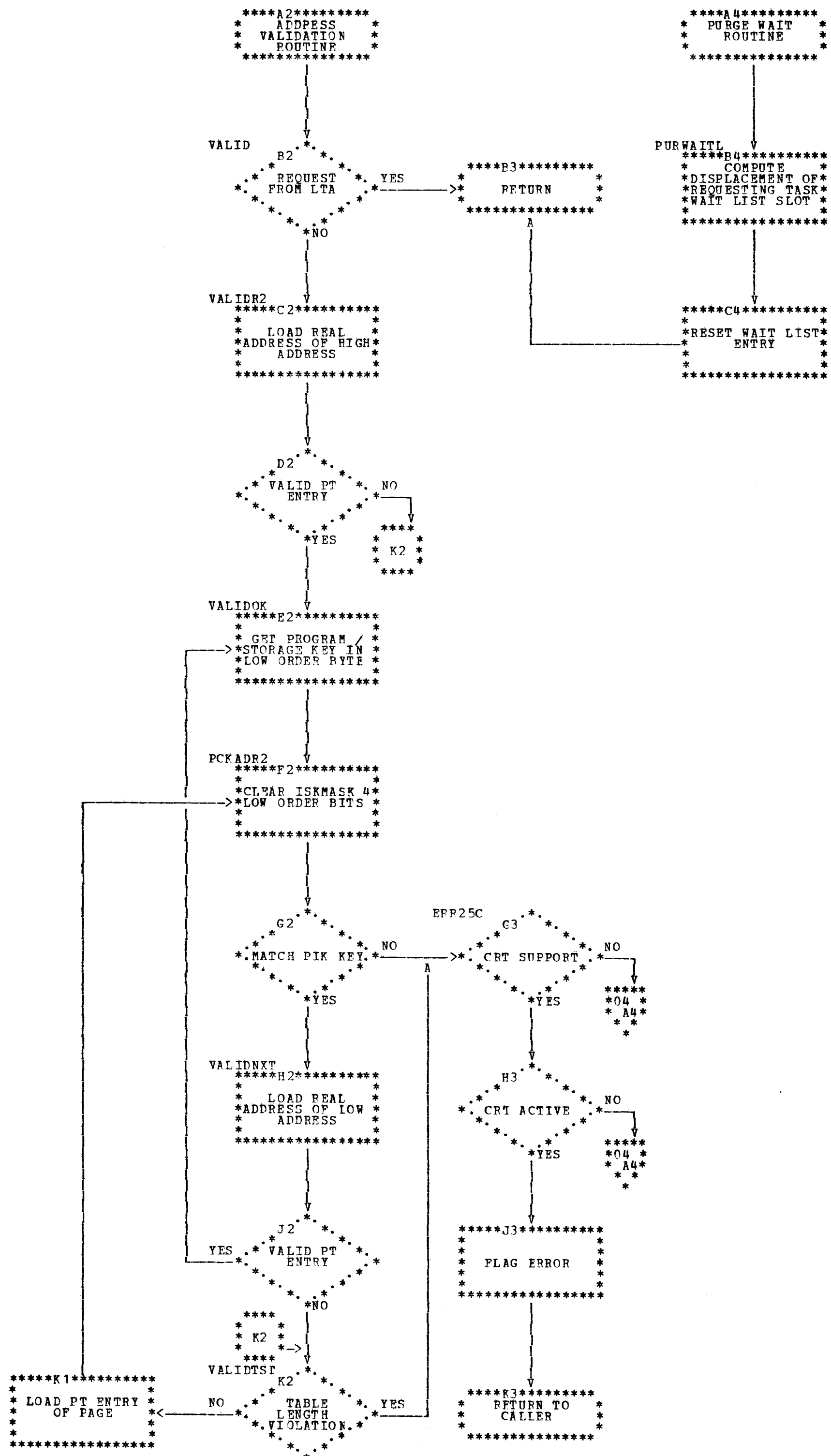


Chart A69. $$A$SUP1 - SGSVC Macrc, Address Validaticn

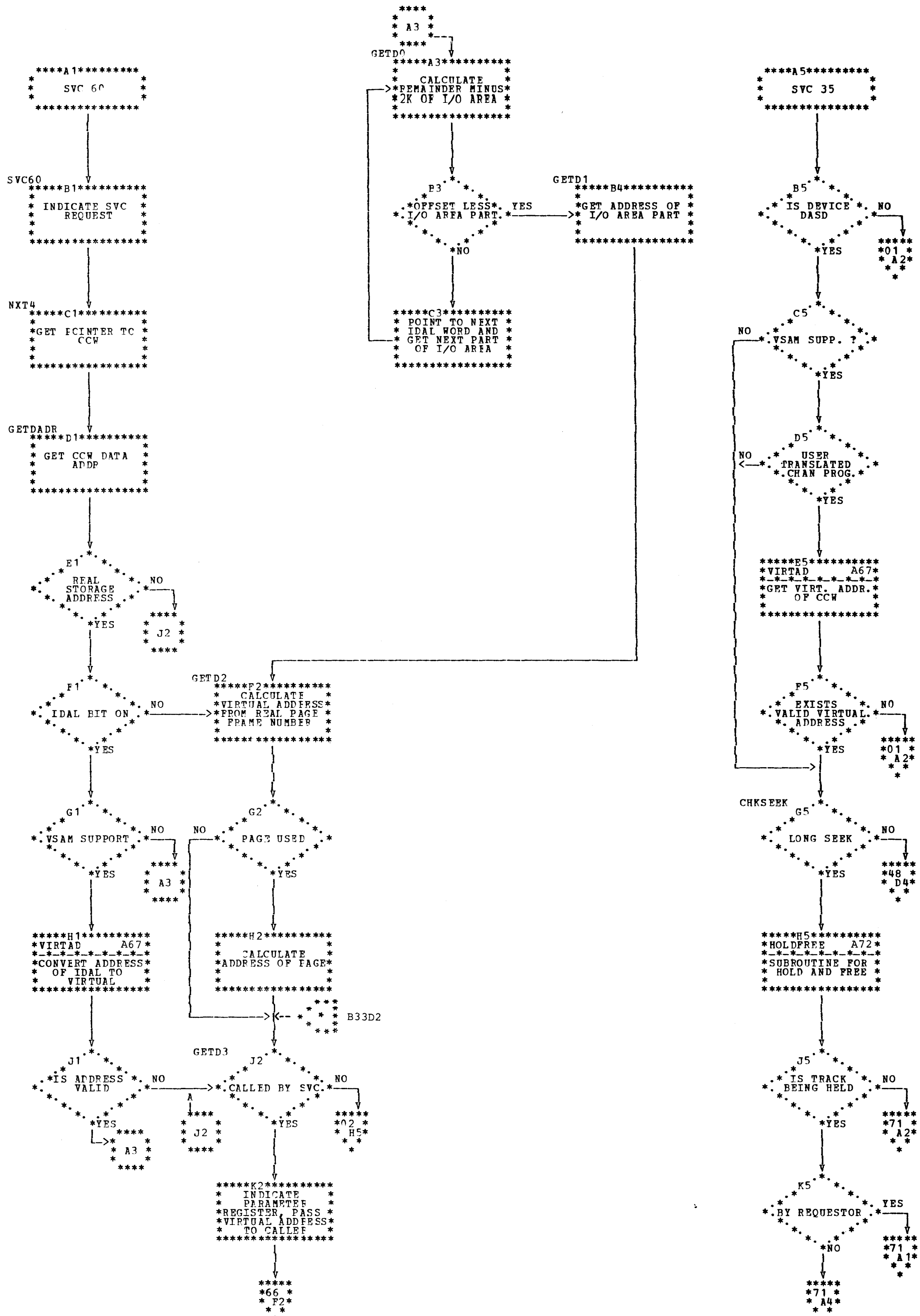


Chart A70. $$A$SUP1 - SGSVC Macro, Provide Virtual Address of Location in I/O Area (SVC 60)

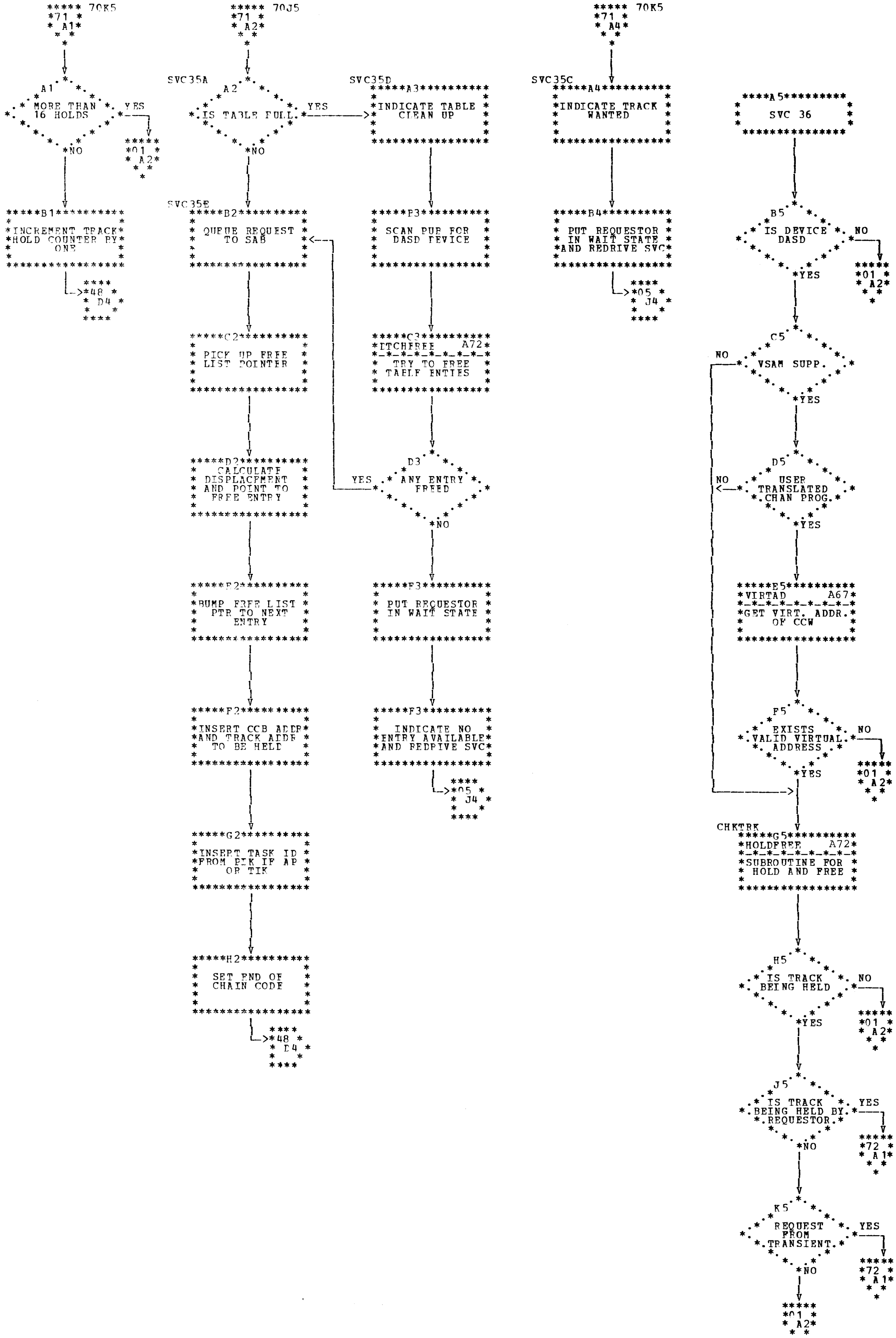


Chart A71. $$A$SUP1 - SGTHAP Macro, Hold Track (SVC 35), Free Track (SVC 36)

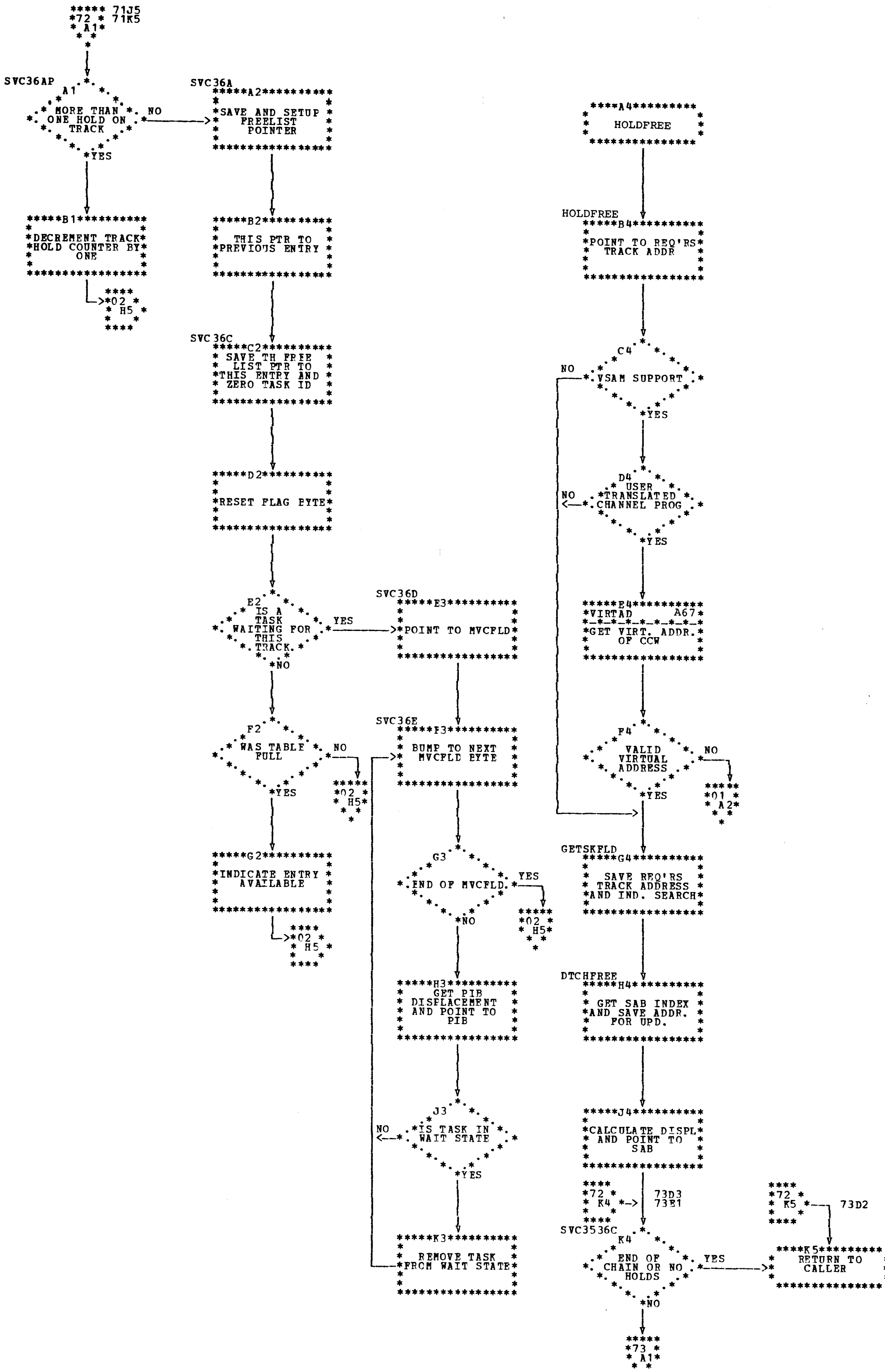


Chart A72. $$A$SUP1 - SGTHAP Macro, Free Track (SVC 36)

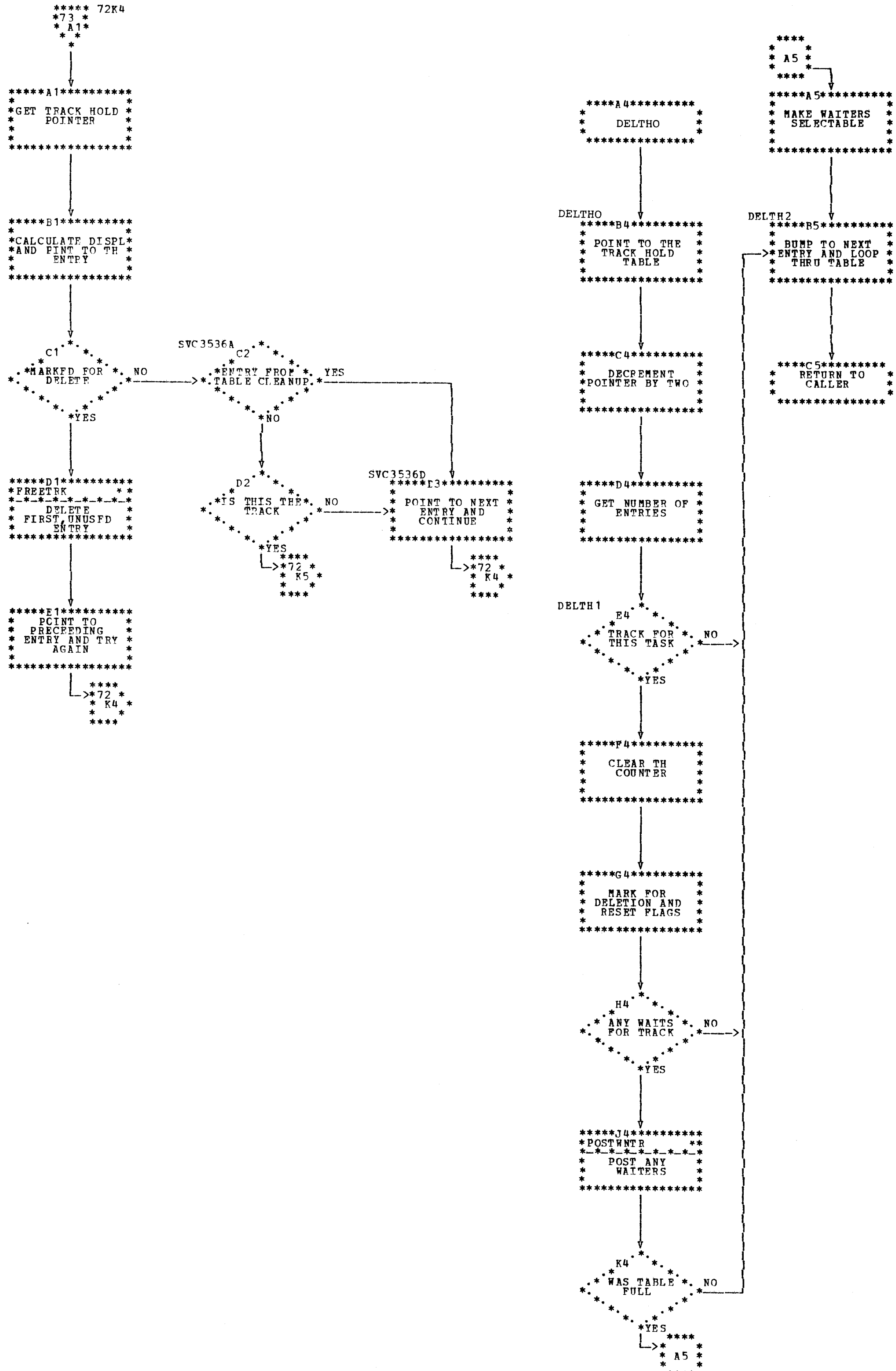


Chart A73. $$A$SUP1 - SGTHAP Macro, Hold Track (SVC 35), Free Track (SVC 36)

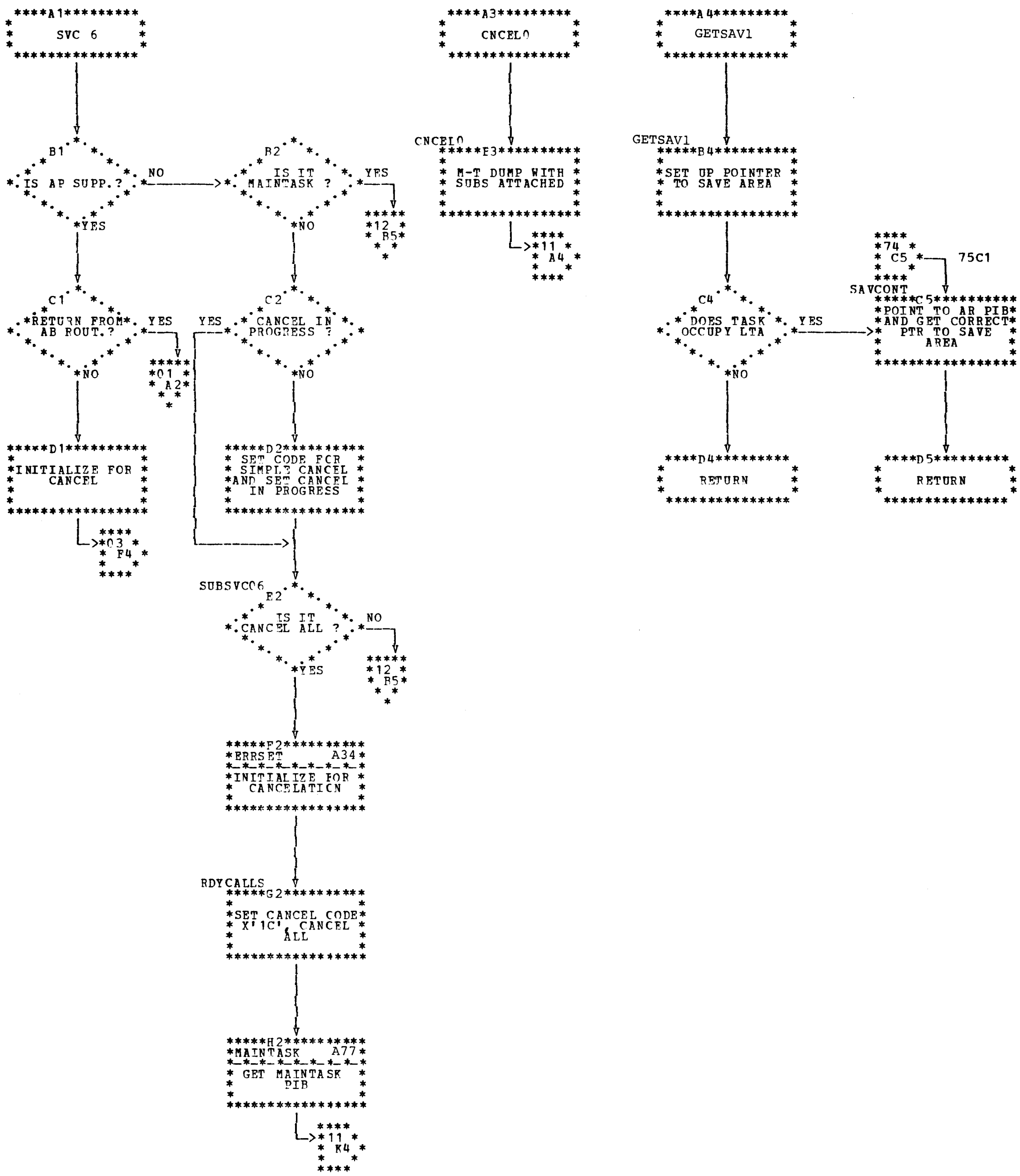


Chart A74. $$A$SUP1 - SGTHAF Macro, Cancel Program or Task (SVC 6)

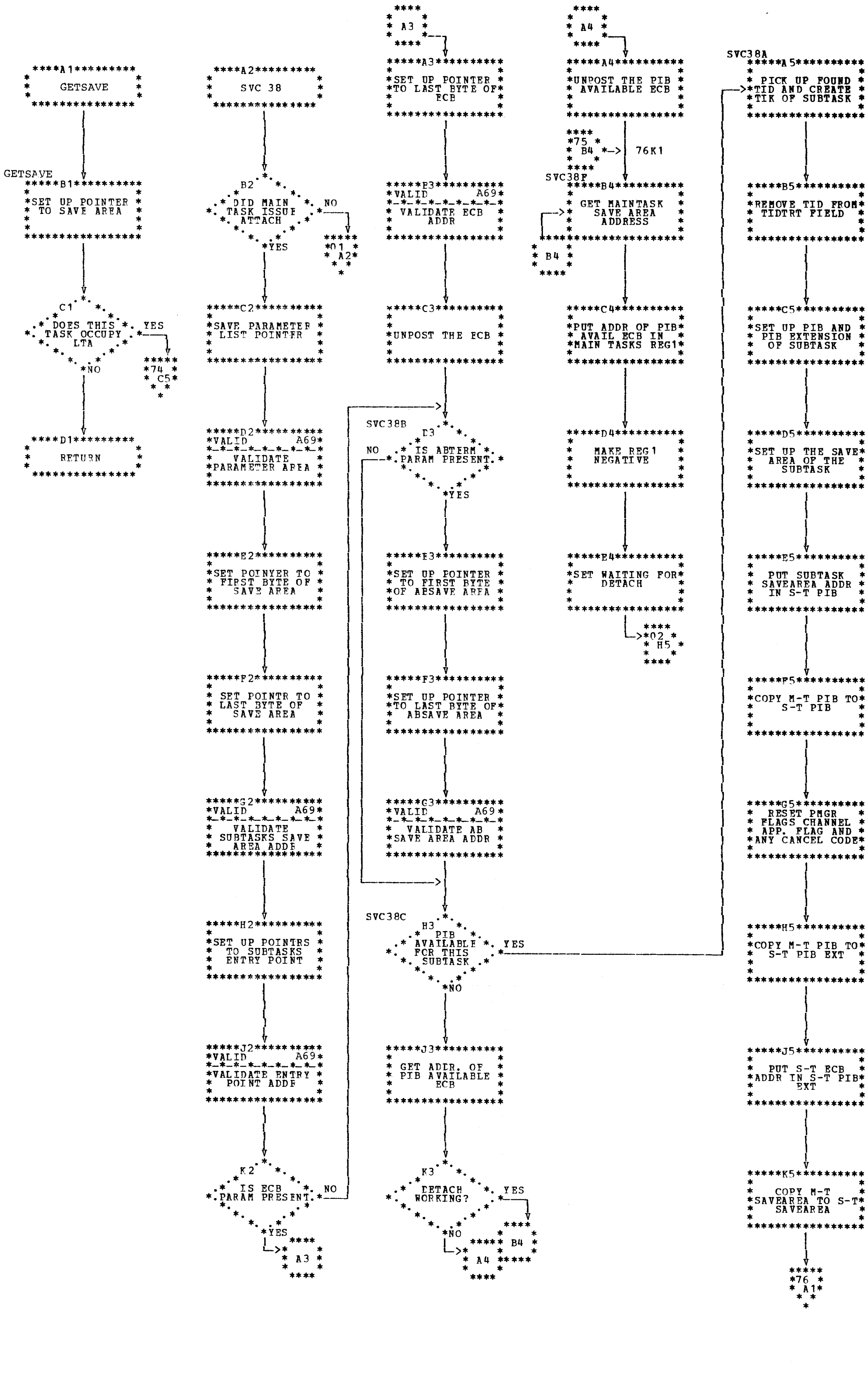


Chart A75. $$A$SUP1 - SGTHAF Macro, Attach Subtask (SVC 38)
Refer to Chart 05.1.

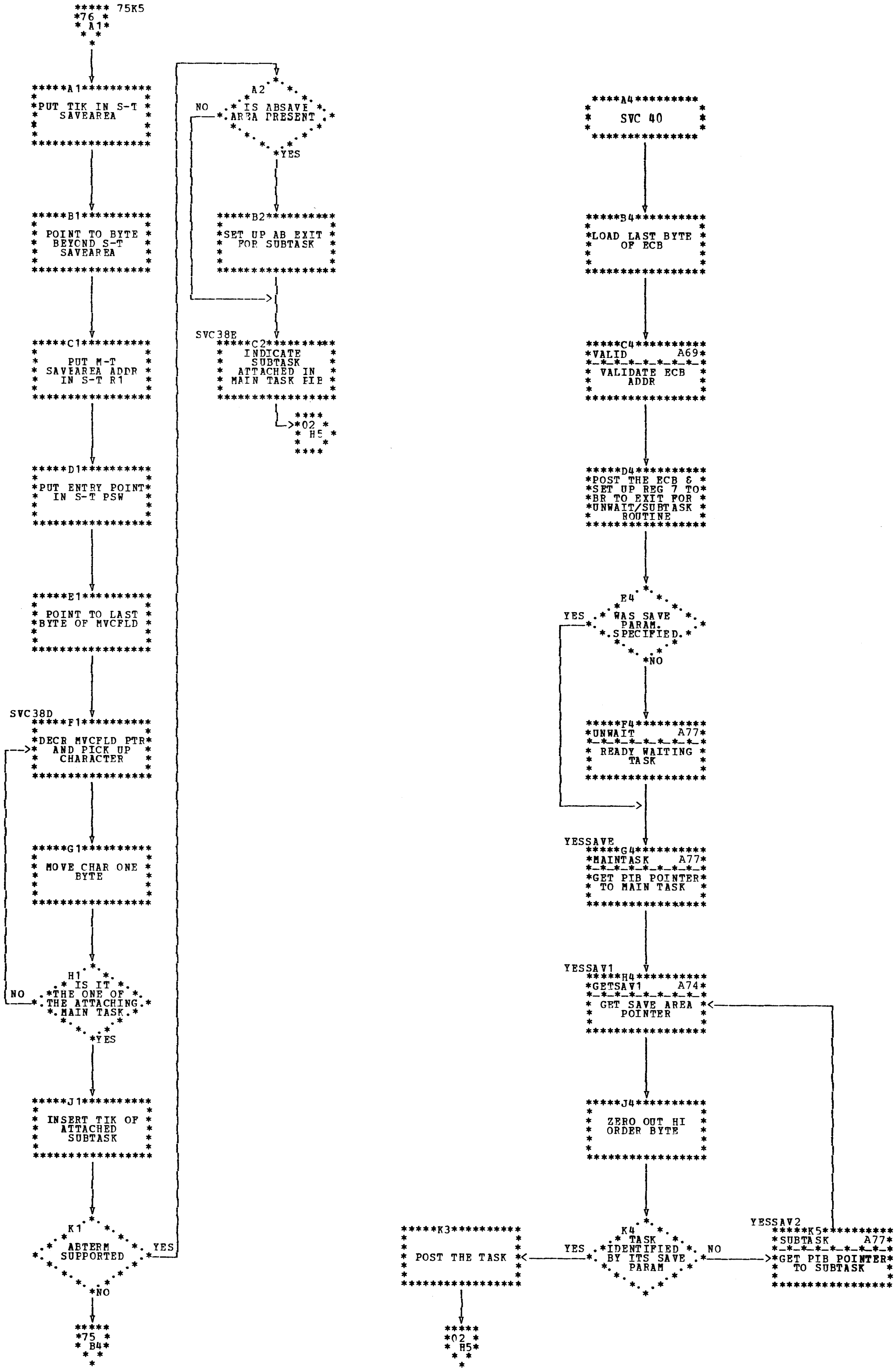


Chart A76. $$A$SUP1 - SGTHAP Macro, Post ECB (SVC 40)
Refer to Chart 05.2.

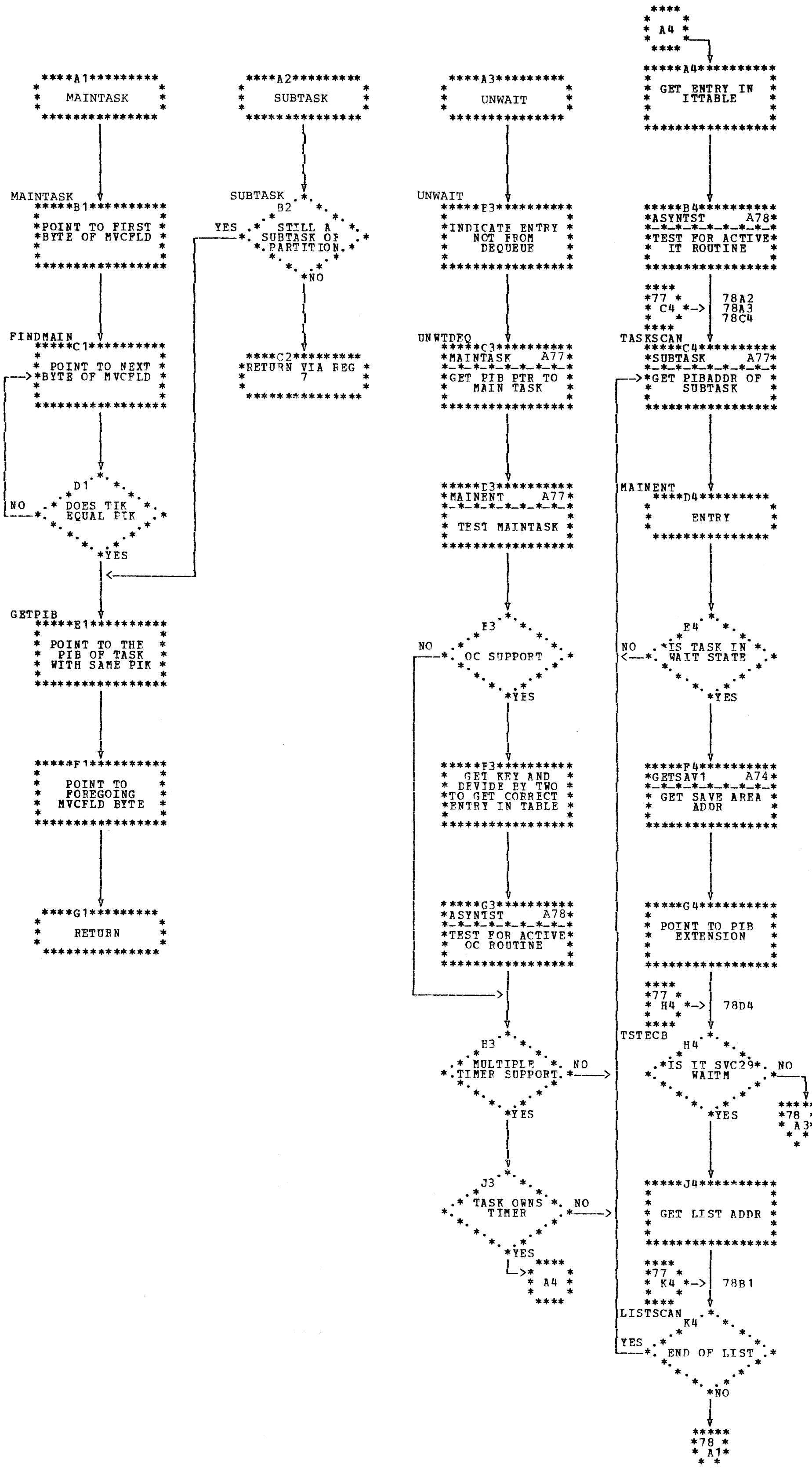


Chart A77. $$A$SUP1 - SGTHAP Macrc, Investigate Waiting Tasks

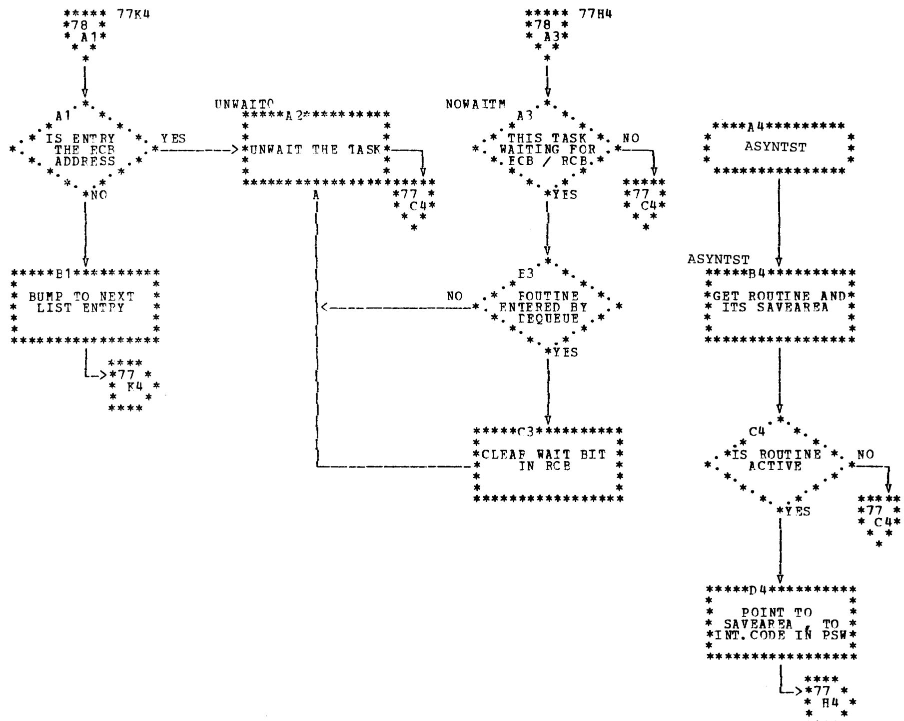


Chart A78. $$A$SUP1 - SGTHAP Macro, Investigate Waiting Tasks

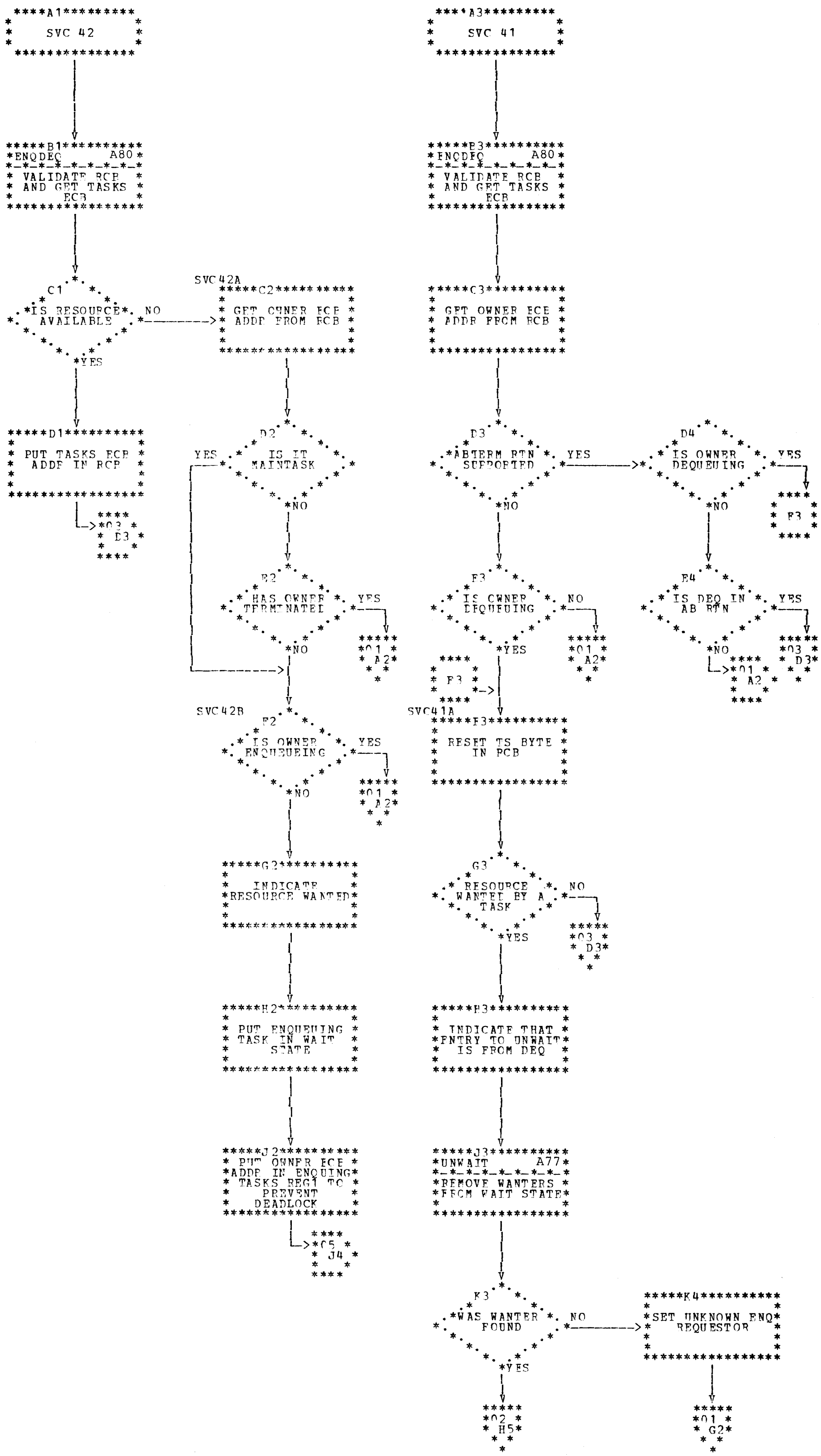


Chart A79. $$A$SUP1 - SGTHAP Macro, Enqueue (SVC 42) and Dequeue (SVC 41) Task from a Resource

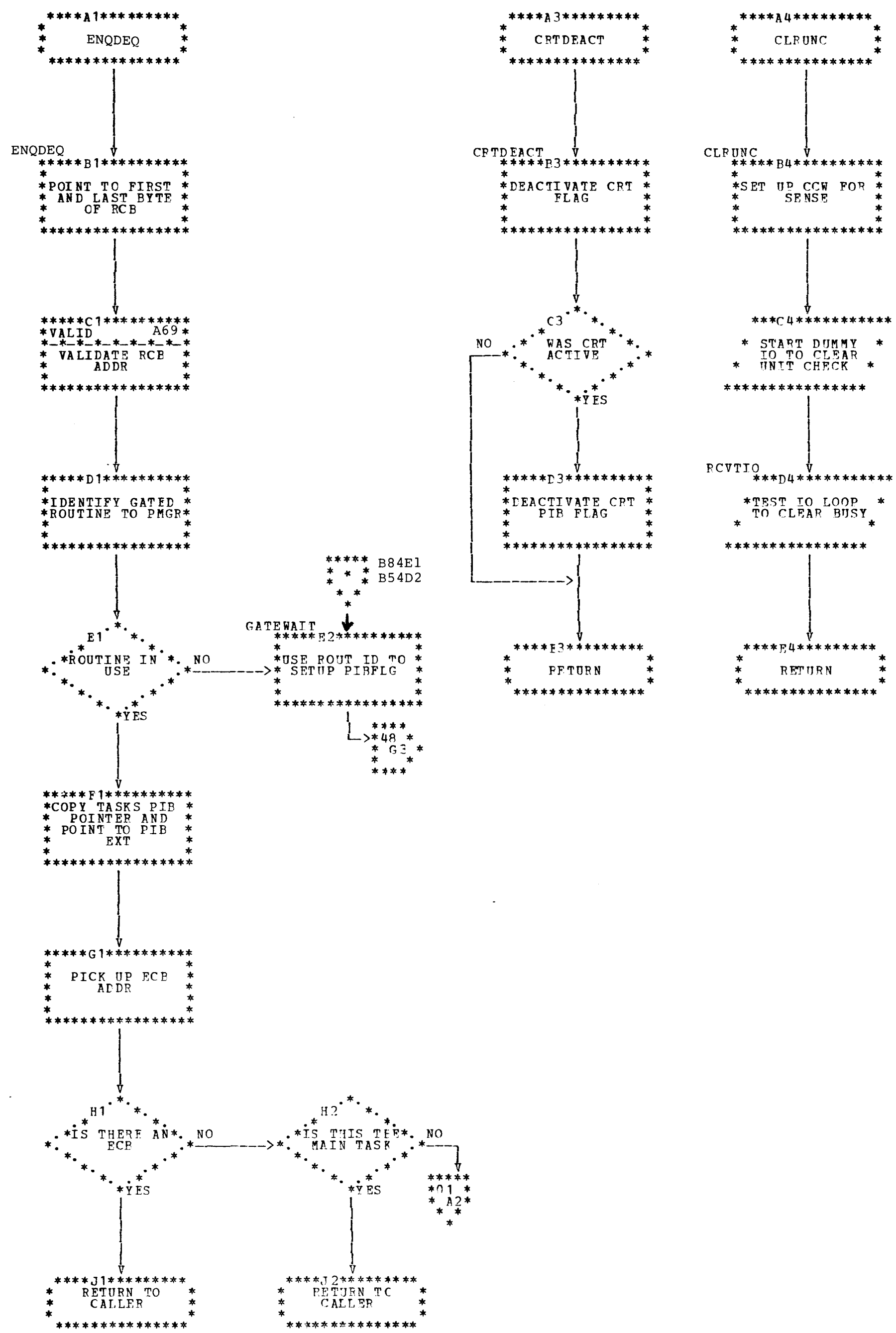


Chart A80. $$A$SUP1 - SGTHAP Macro, Dequeue (SVC 41) Task from a Resource

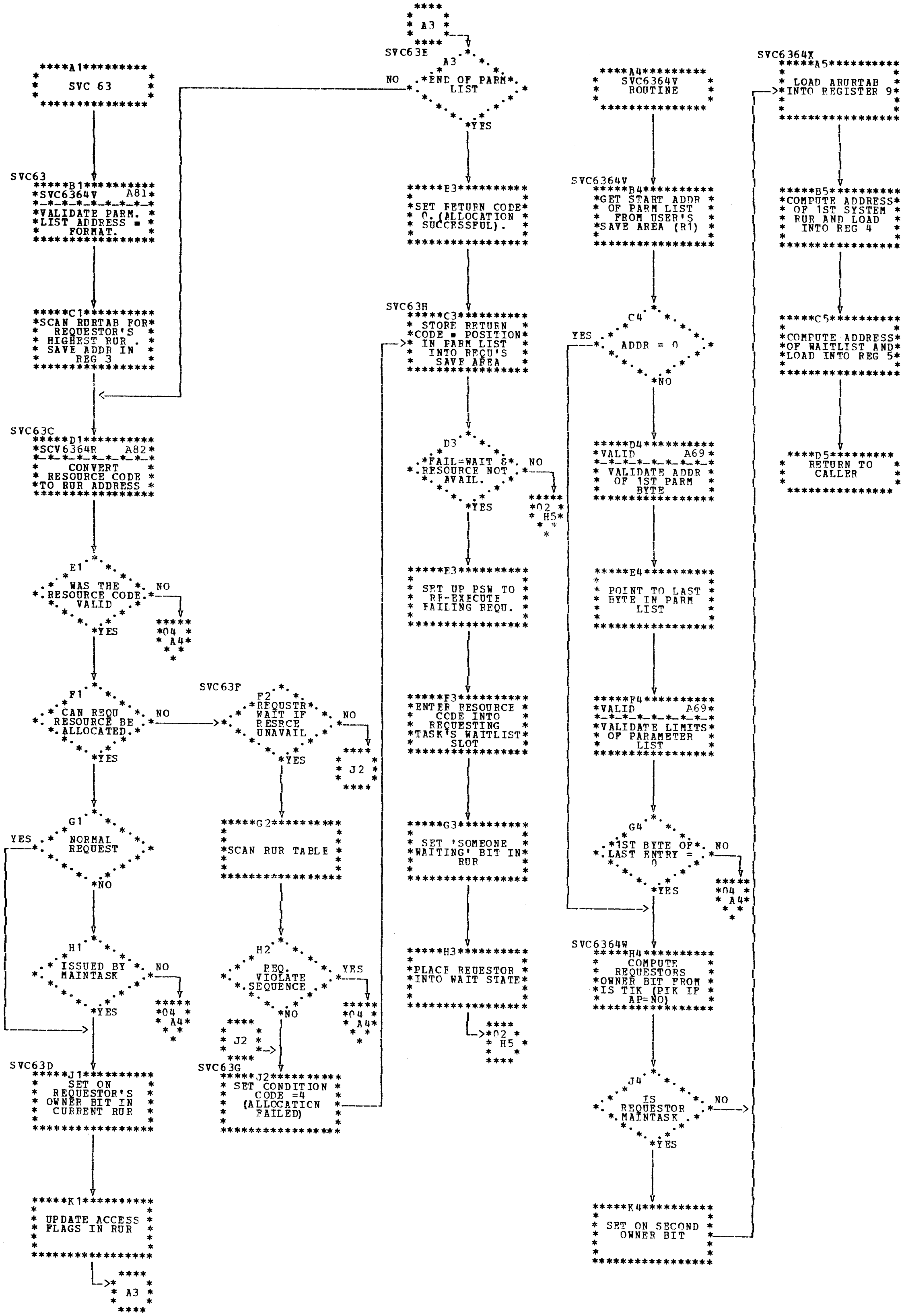


Chart A81. $$A$SUP1 - IOINTER Macro, USE (SVC 63)/RELEASE (SVC 64) Common Subroutines
Refer to Chart 05.2.

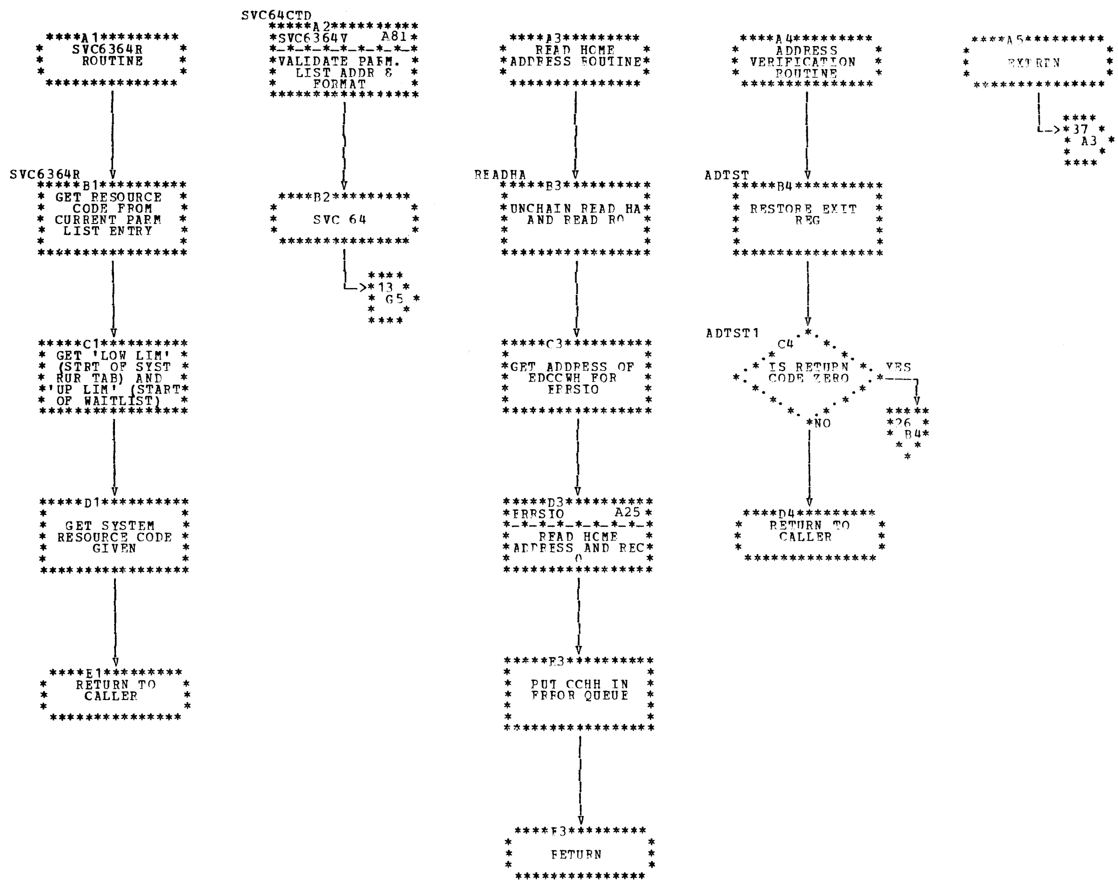


Chart A82. $$A$SUP1 - IOINTER Macro, USE (SVC 63)/RELEASE (SVC 64) Common Subroutines

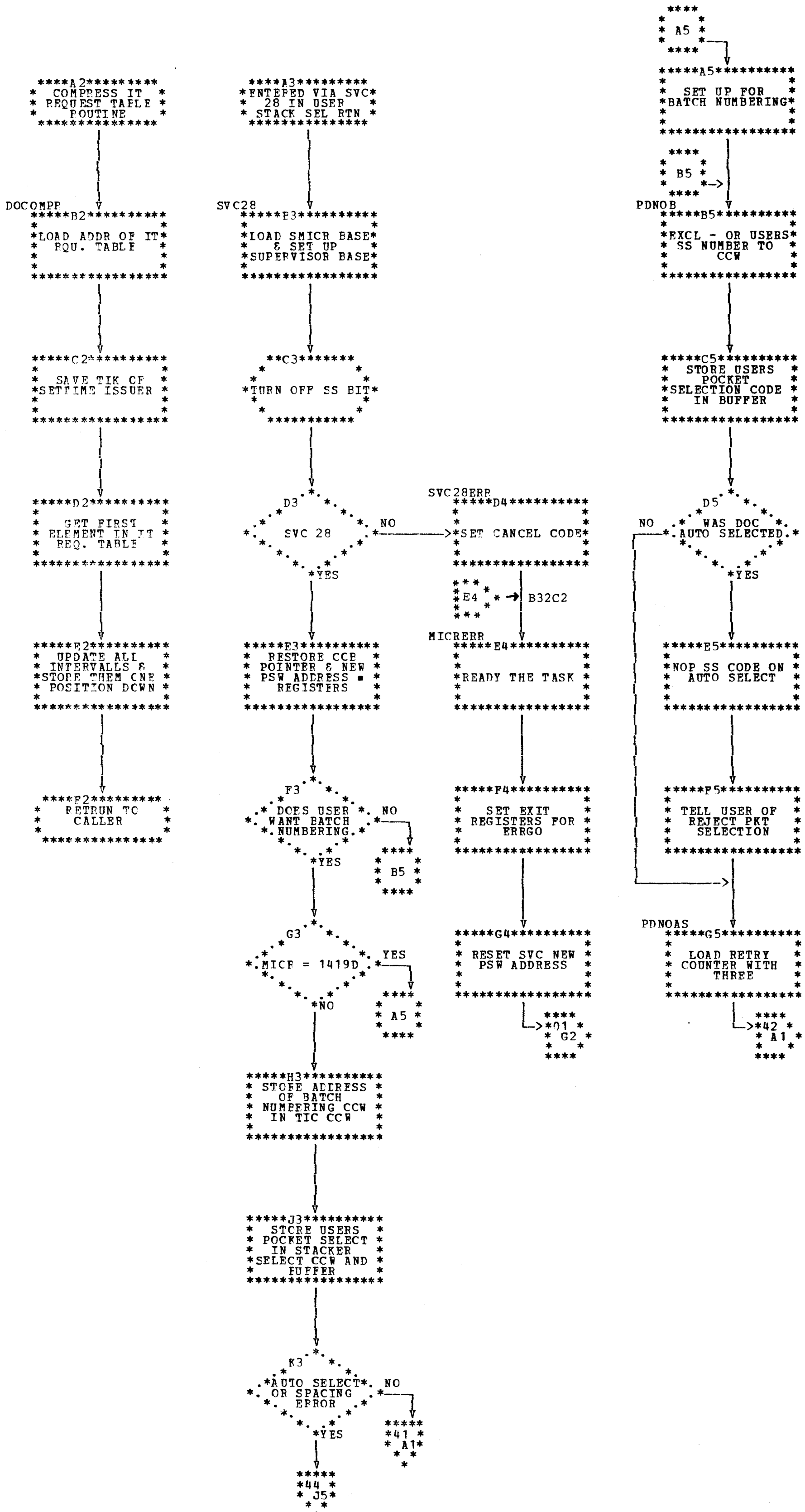


Chart A83. $$A$SUP1 - SMICR Macro, Return from User Stacker Select Routine (SVC 28)
Refer to Chart 05.1.

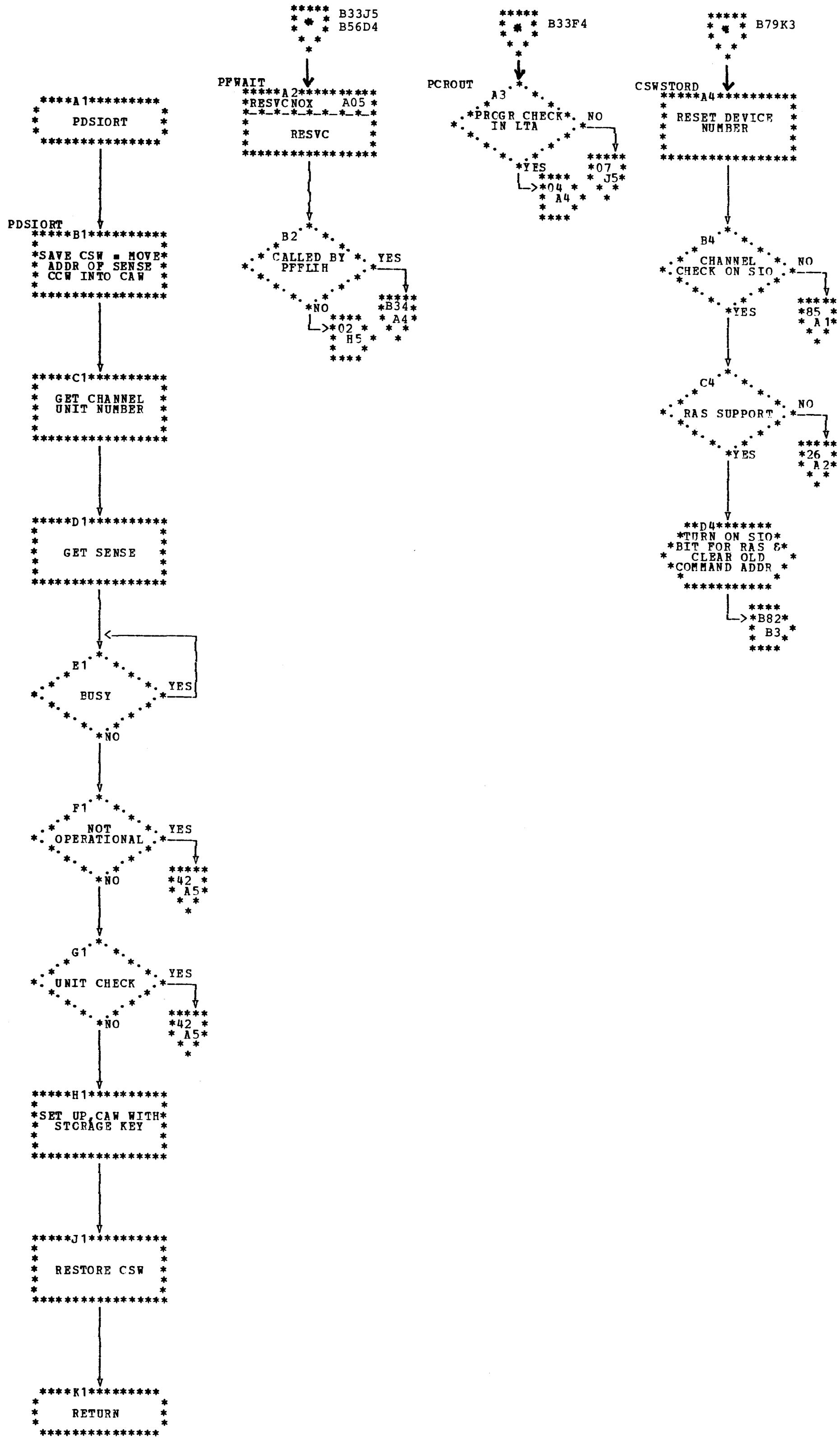


Chart A84. $$A$SUP1 - SMICR Macro, Return from User Stacker Select Routine (SVC 28)
Refer to Chart 05.1.

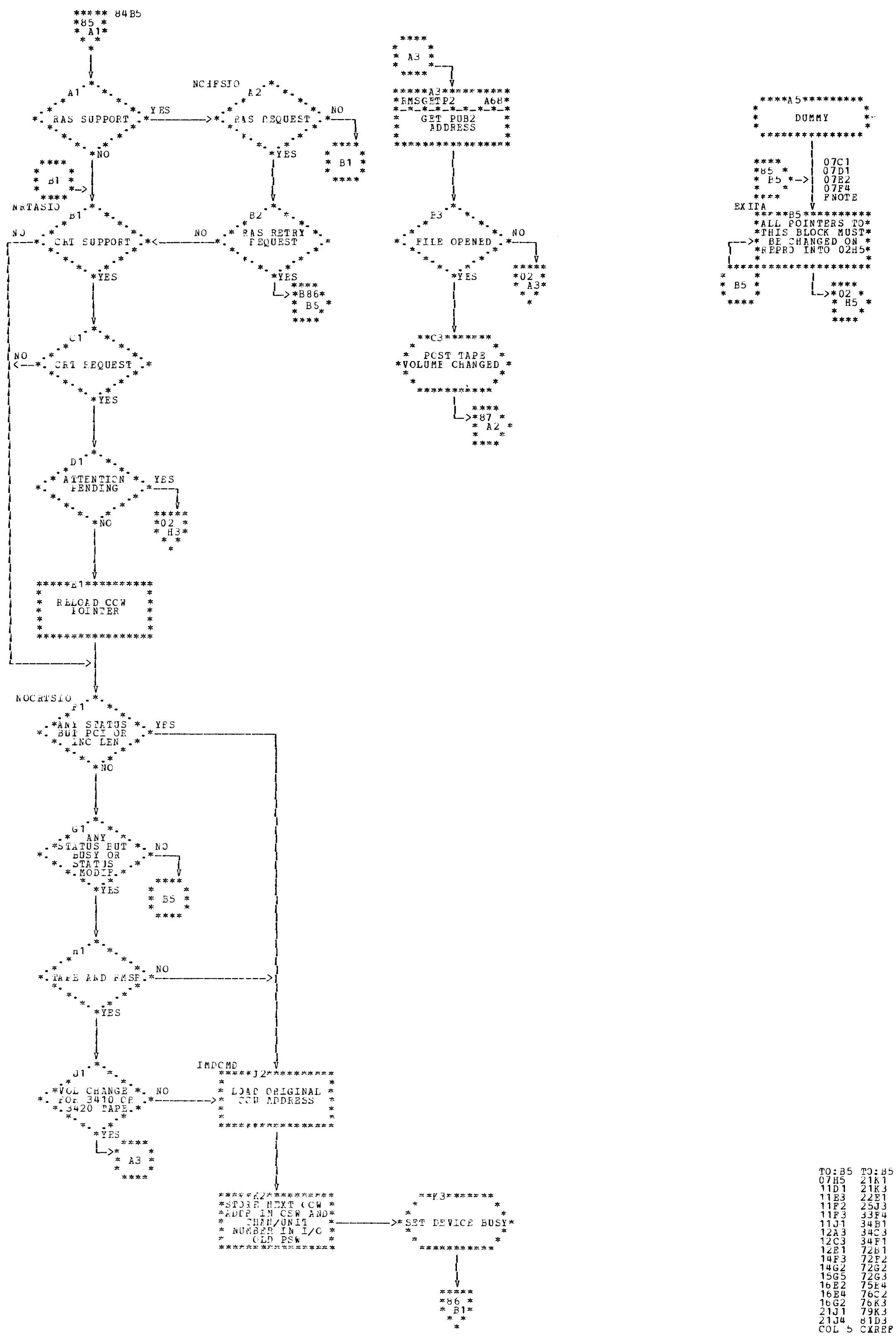


Chart A85. $$A$SUP1 - IOINTER Macro, CSW Stored at SIO Time
Refer to Chart 06.4.

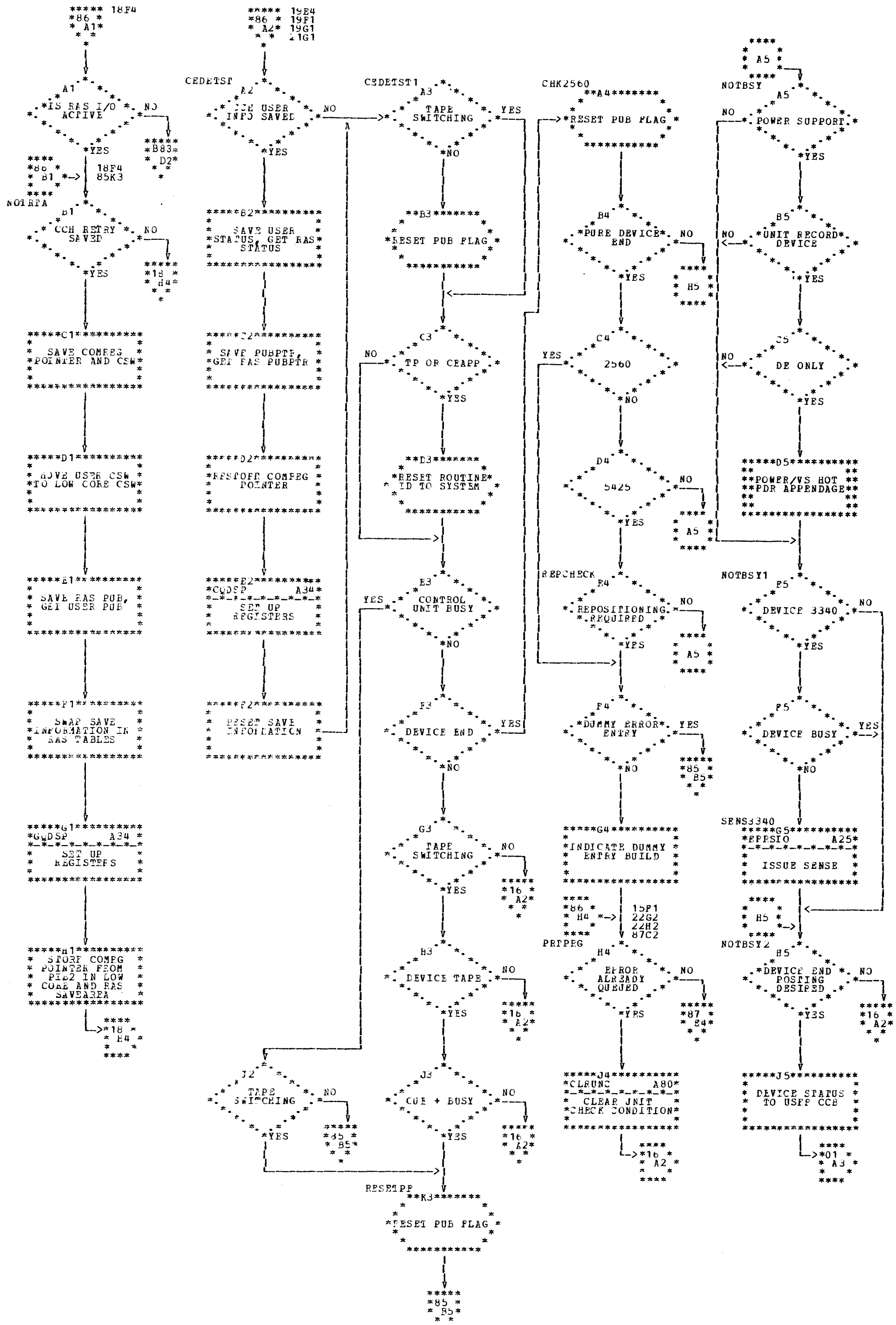


Chart A86. $$A$SUP1 - IOINTER Macro

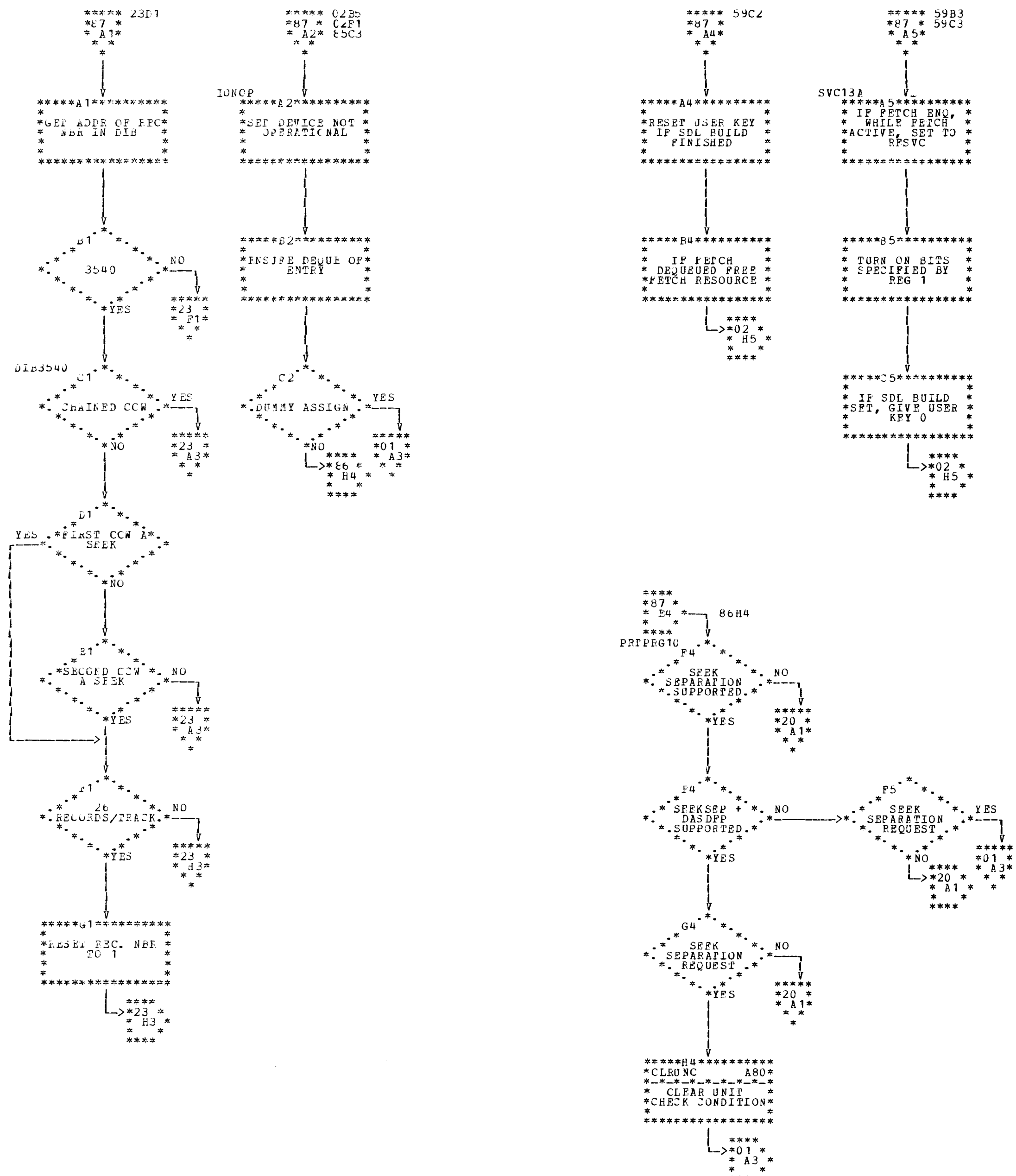


Chart A87. $$A$SUP1 - IOINTER Macro

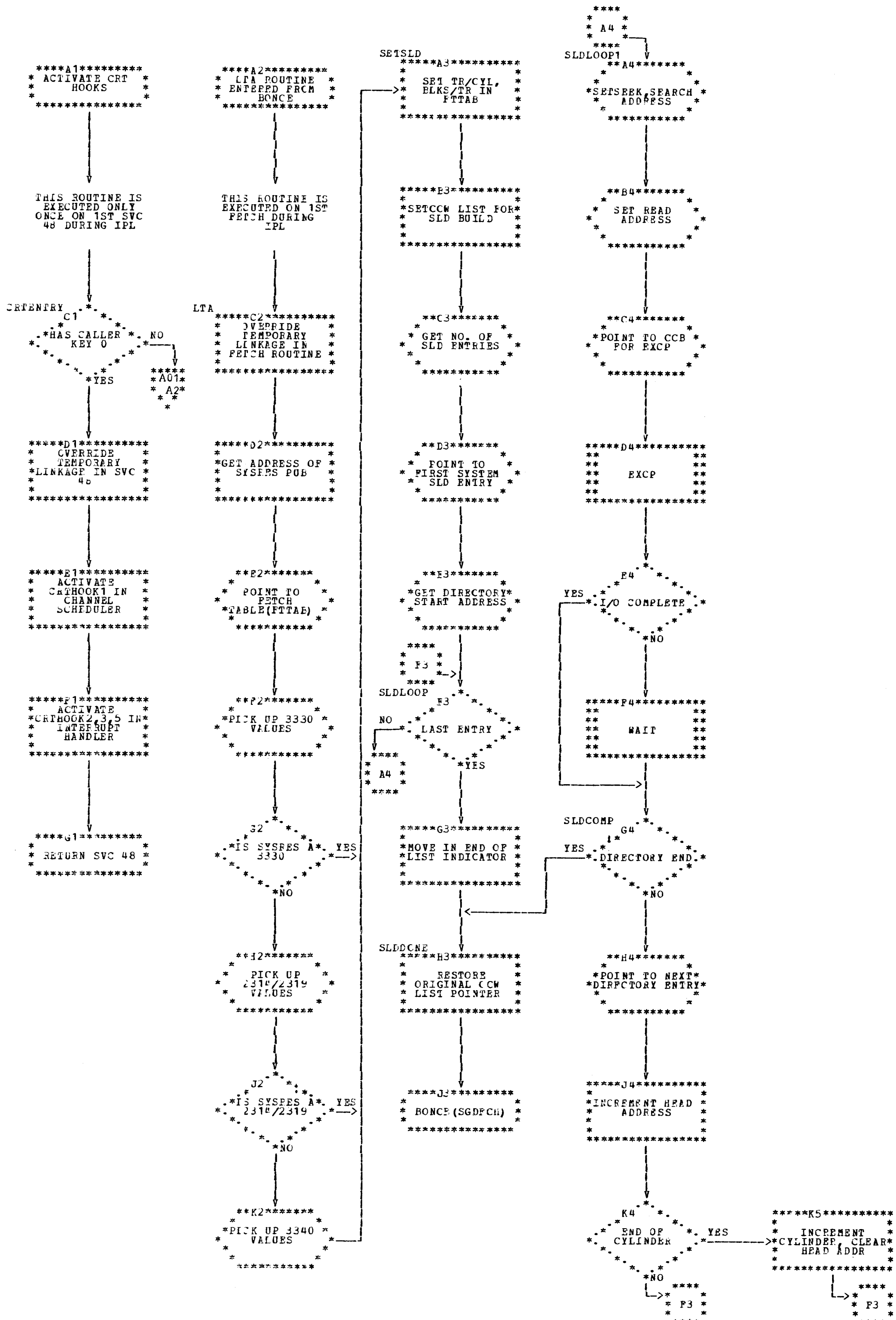


Chart B01. $$A$SUP1 - SGEND Macro, Activate CRT Hooks, LTA Routine Entered from BONCE

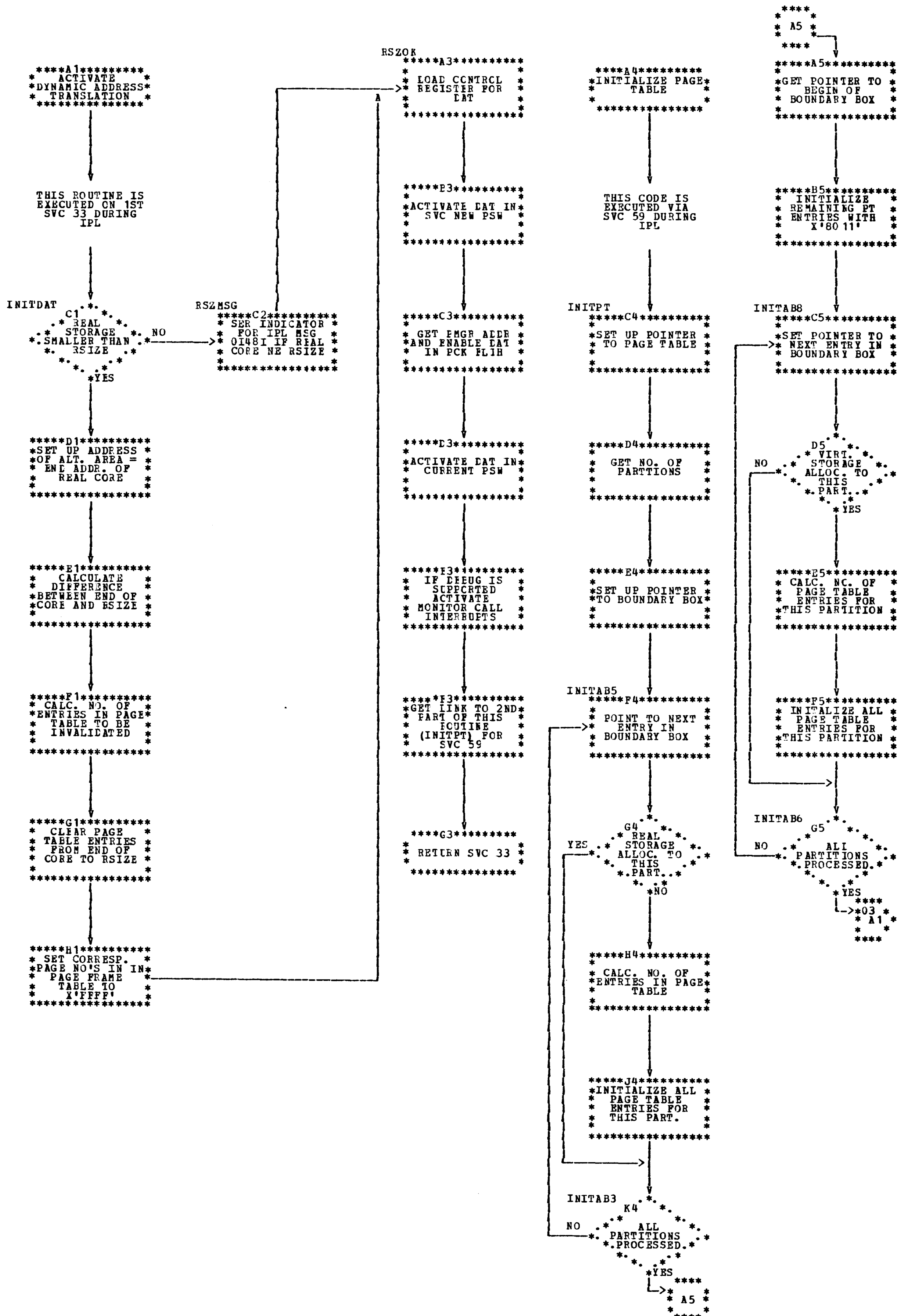


Chart B02. $$A$SUP1 - SGEND Macro, Activate Dynamic Address Translation, Initialize Page Table

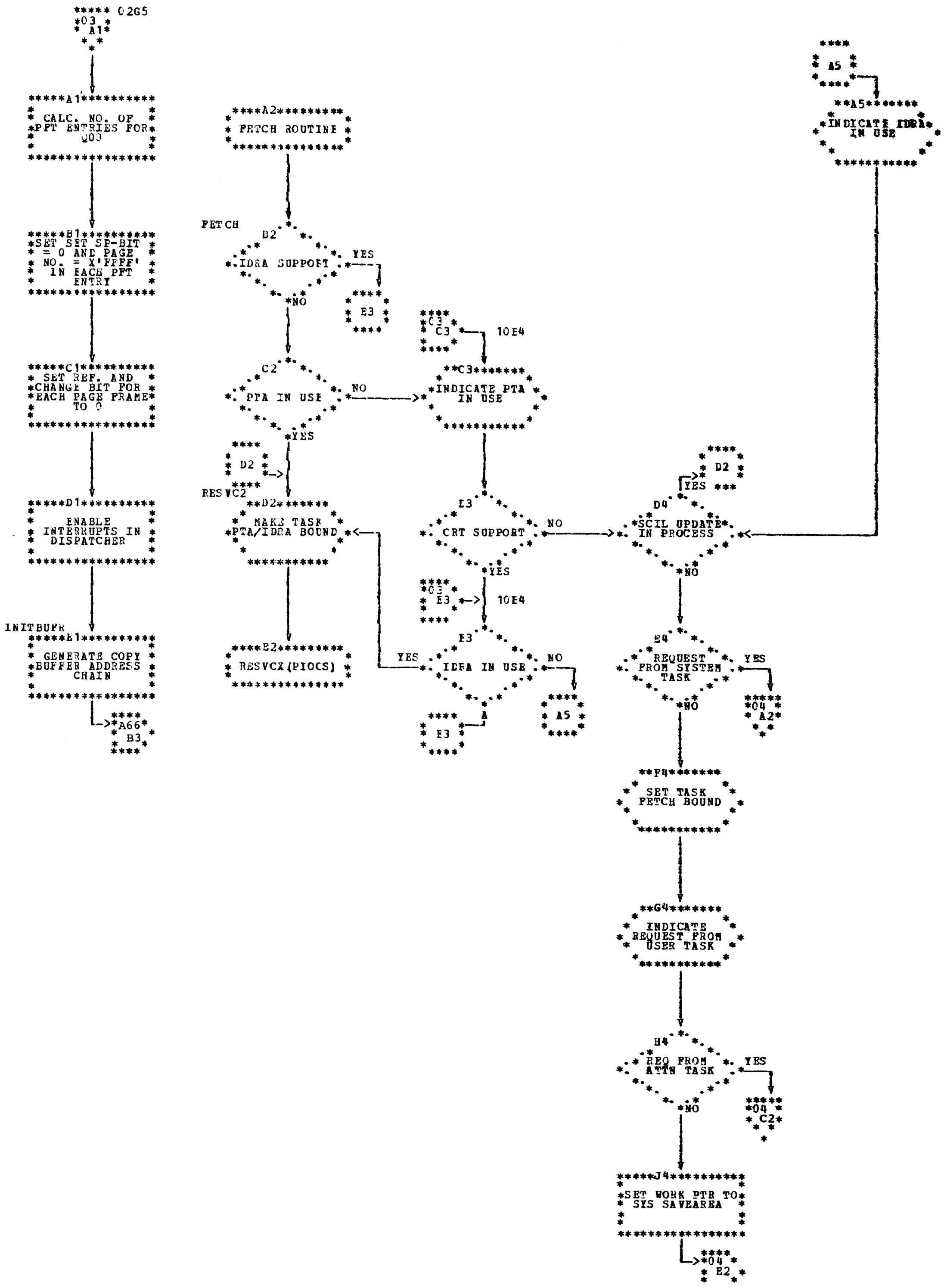


Chart B03. $$A$SUP1 - SGDFCH Macrc, Fetch Routine

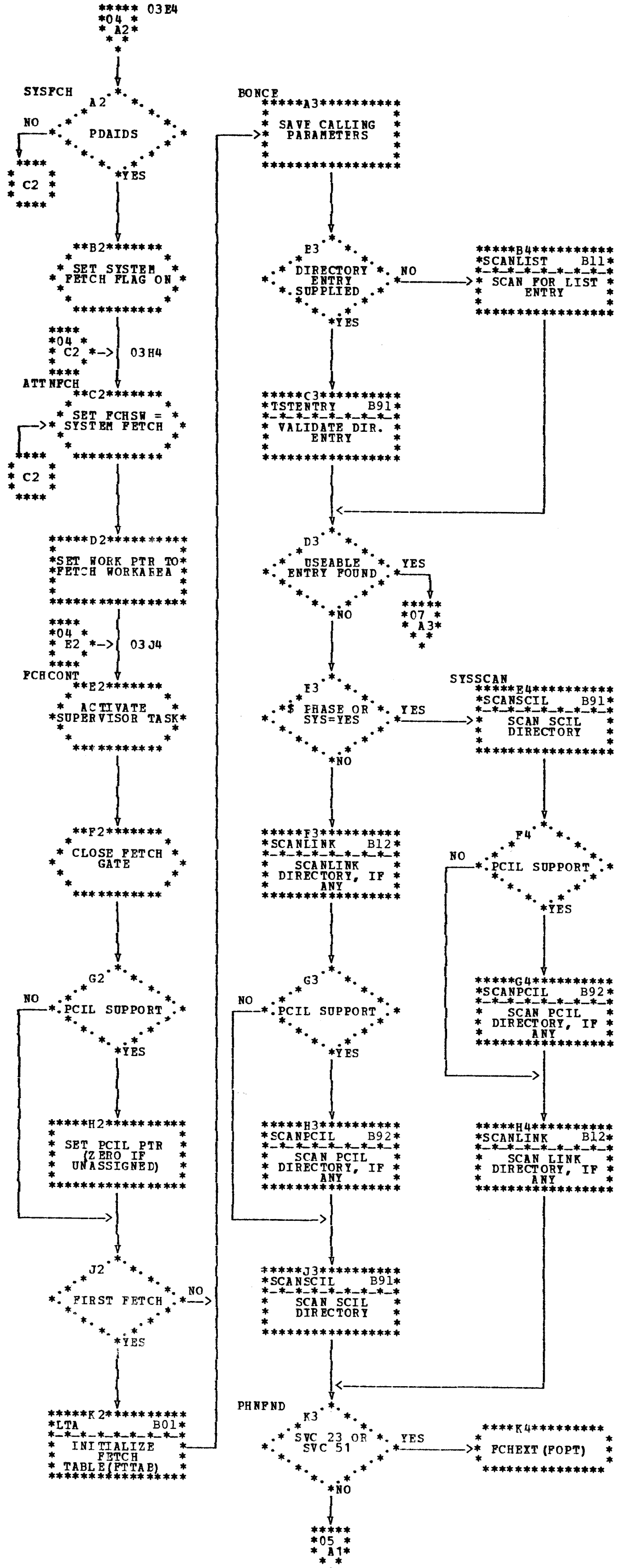


Chart B04. $$A$SUP1 - SGDFCH Macrc, Fetch Rcutine

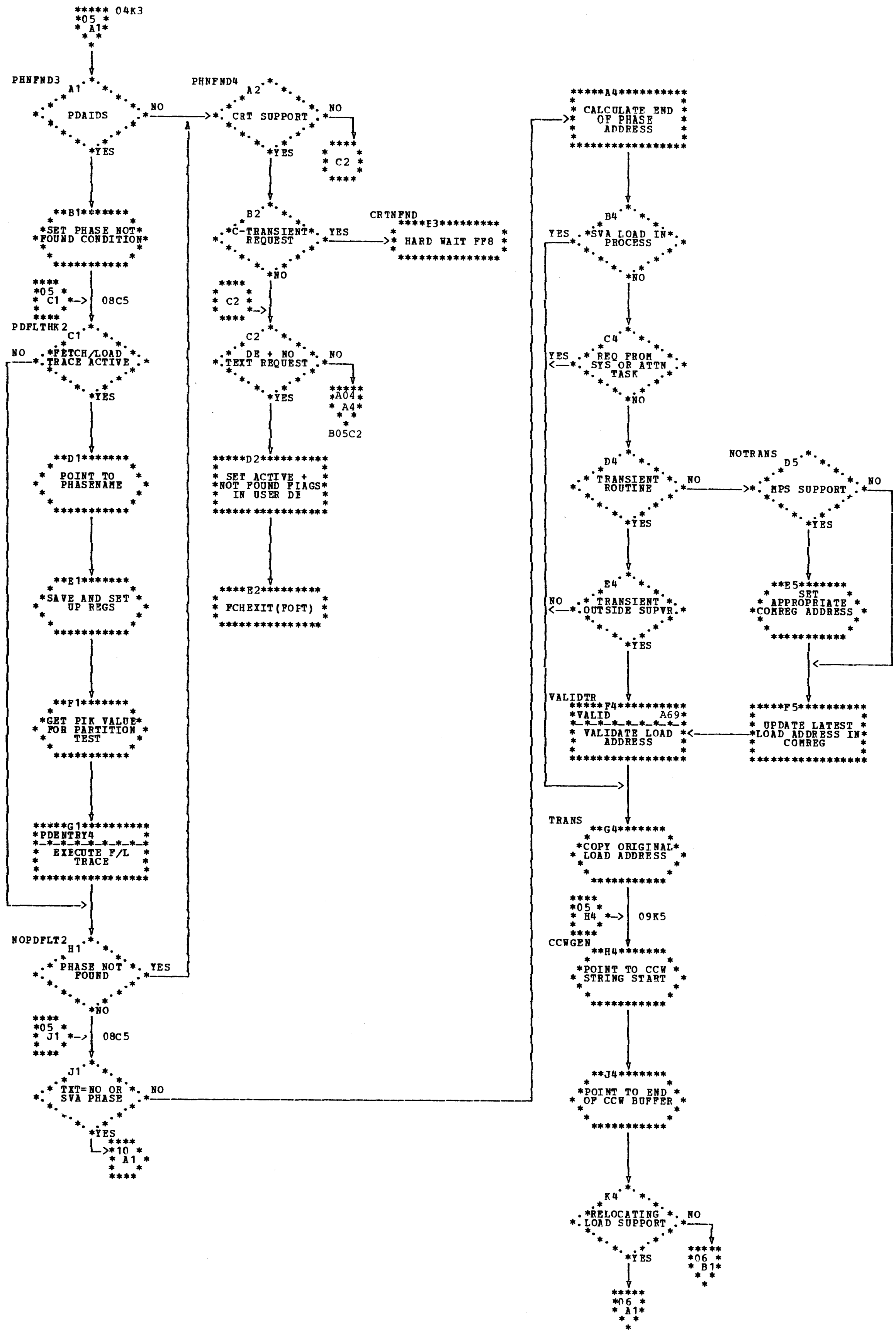


Chart B05. $$A$SUP1 - SGDFCH Macrc, Fetch Routine, Phase Not Found

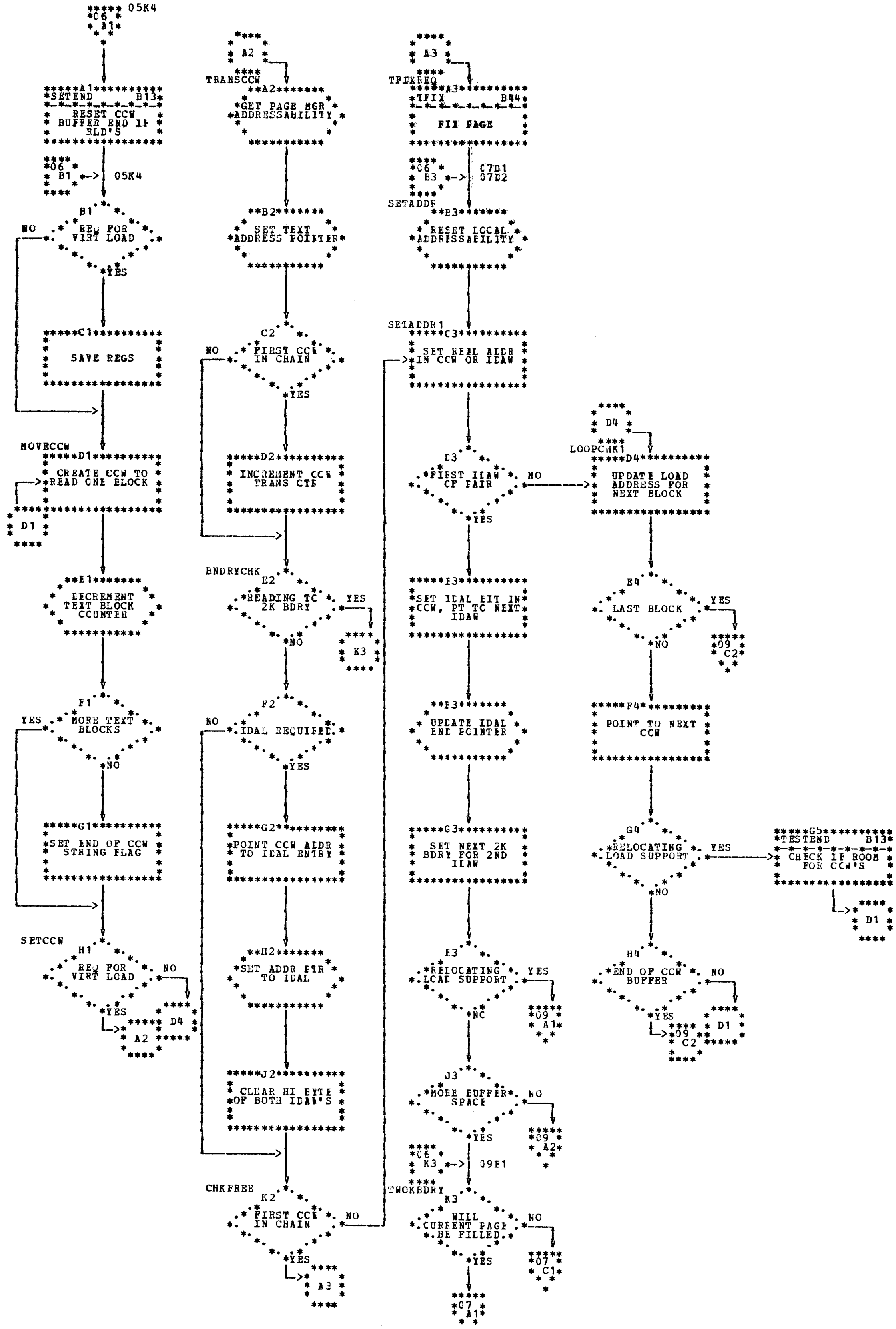


Chart B06. $$A$SUP1 - SGDFCH Macro, Channel Program Translation for Fetch Program

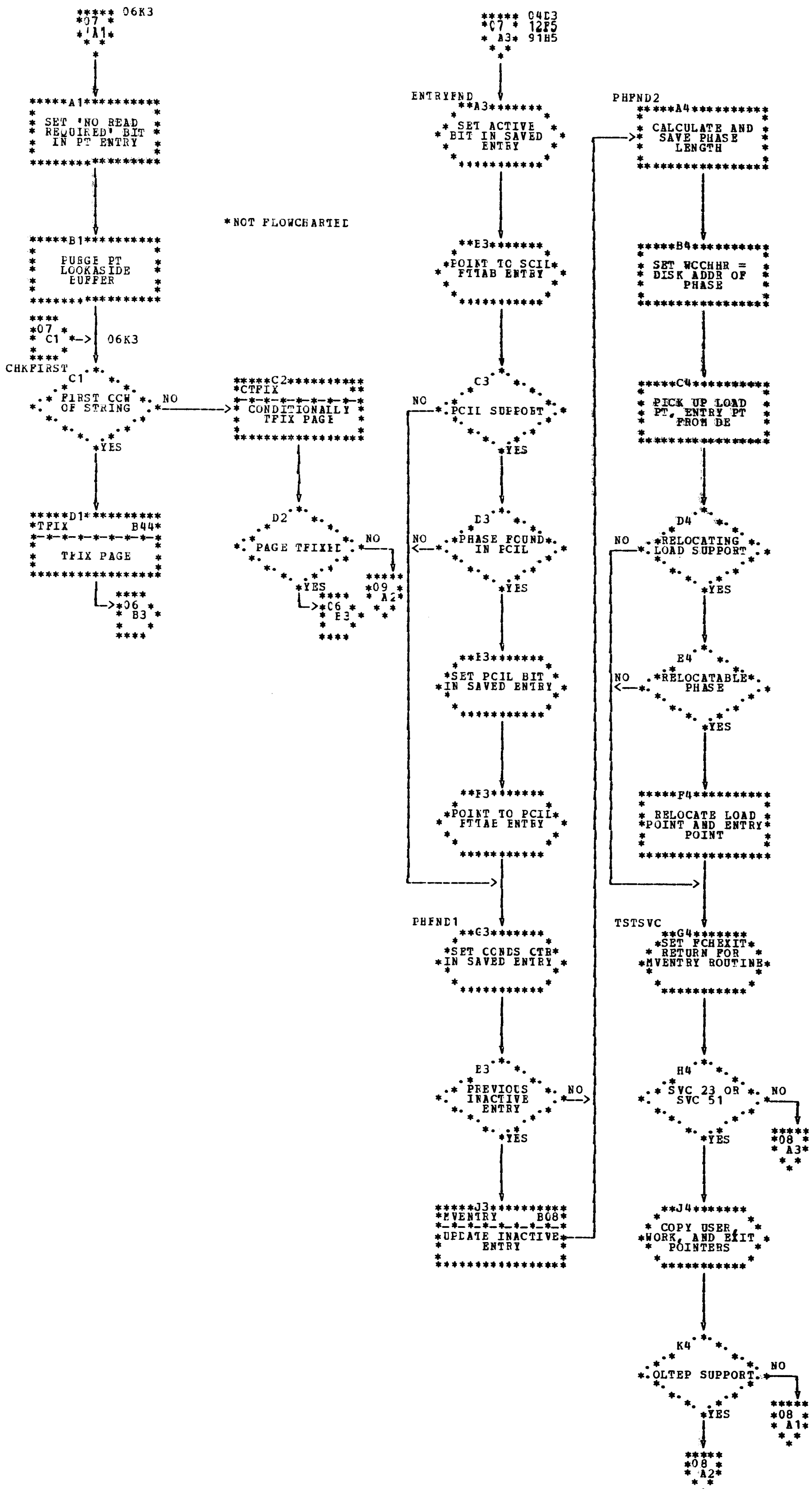


Chart B07. $$A$SUP1 - SGDFCH Macro, Calculate Load Point and Entry Point of Phase

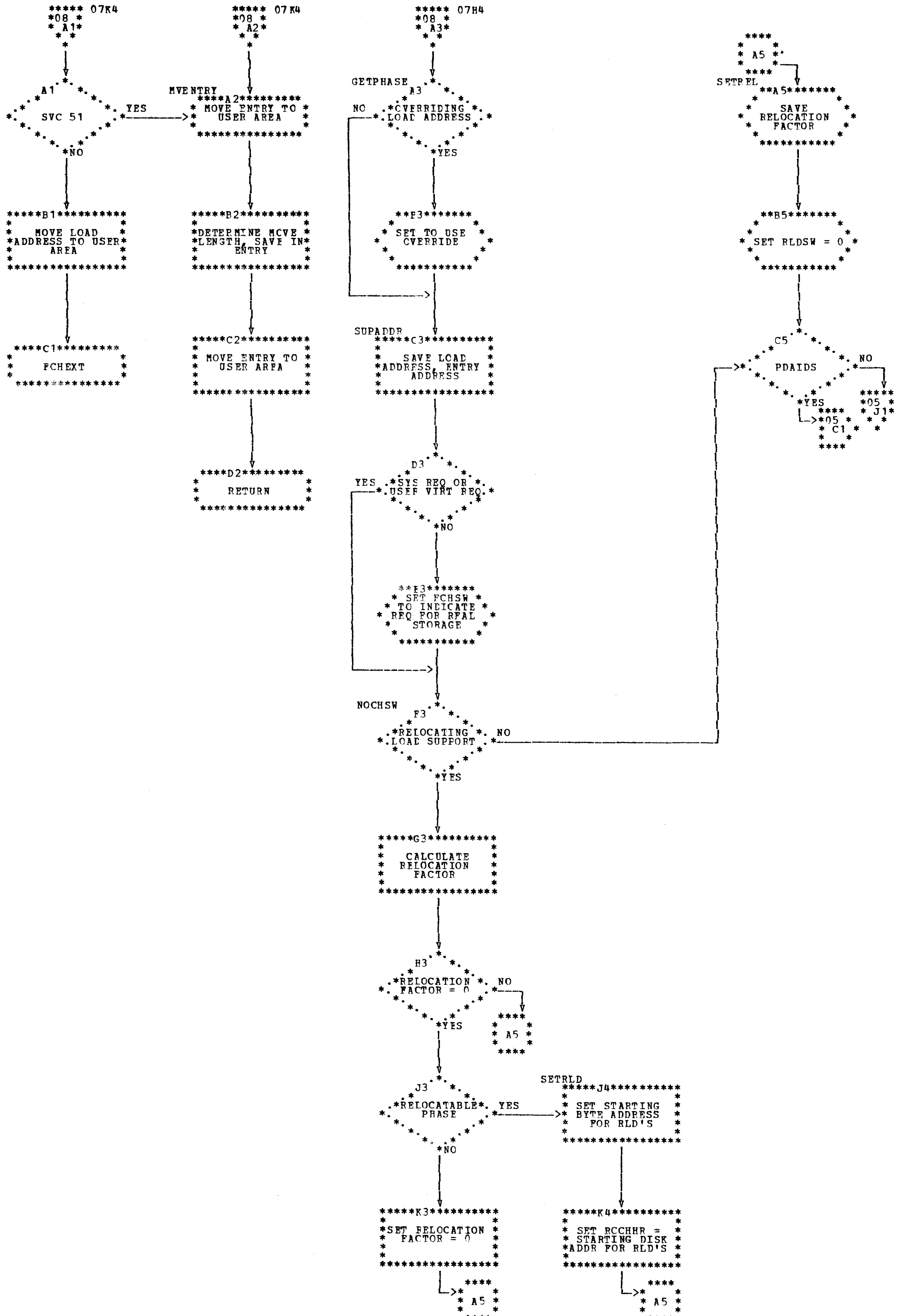


Chart B08. $$A$SUP1 - SGDFCH Macro, Load the Phase

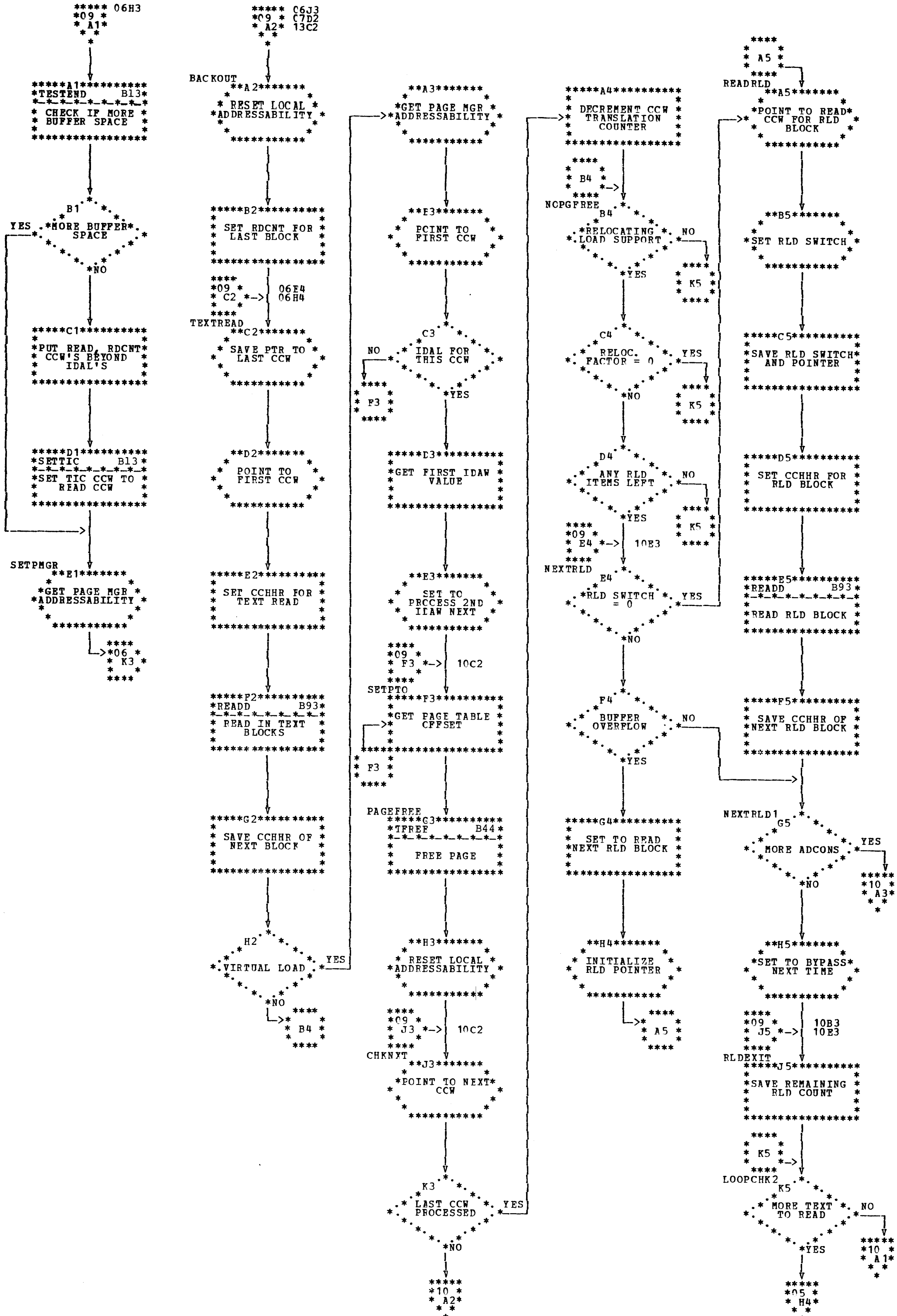


Chart B09. $$A$SUP1 - SGDFCH Macro, Channel Program Retranslation

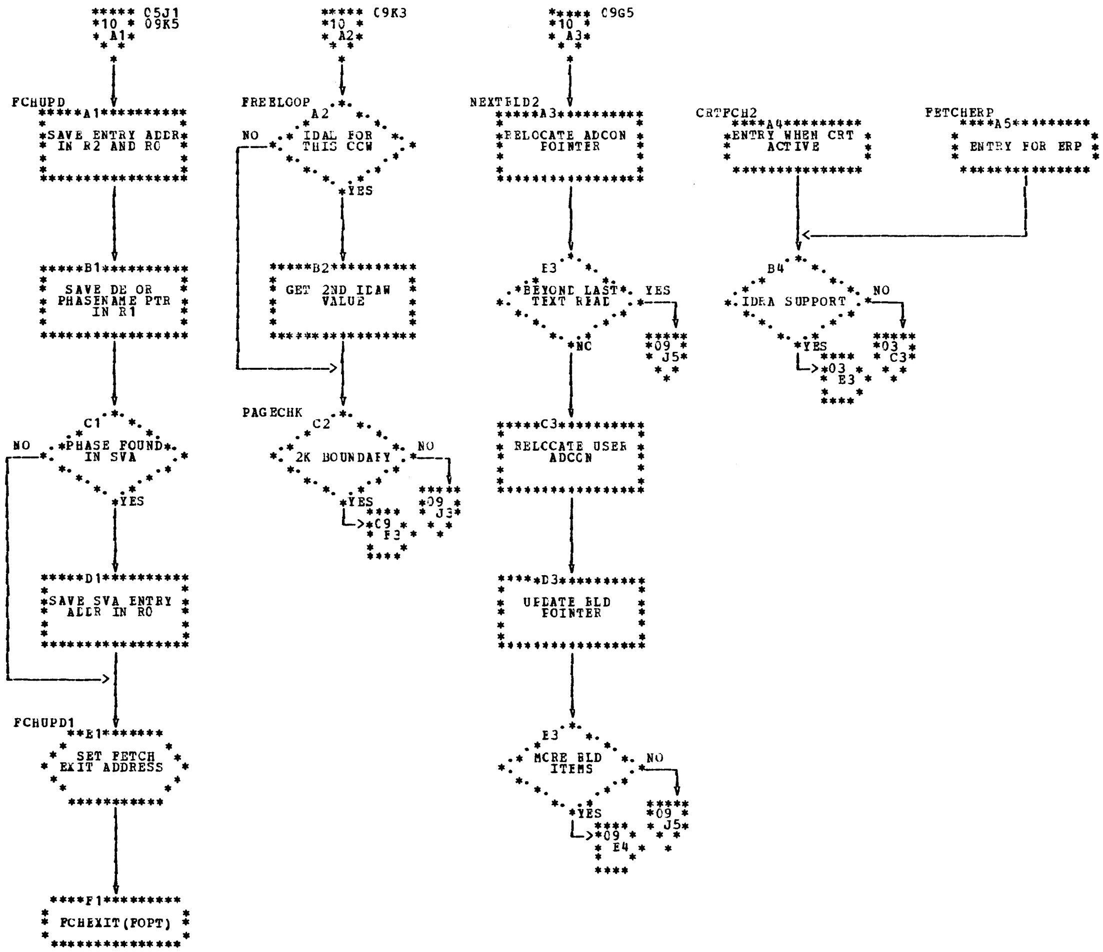


Chart B10. $$A$SUP1 - SGDFCH Macro, Relocate User Adcons

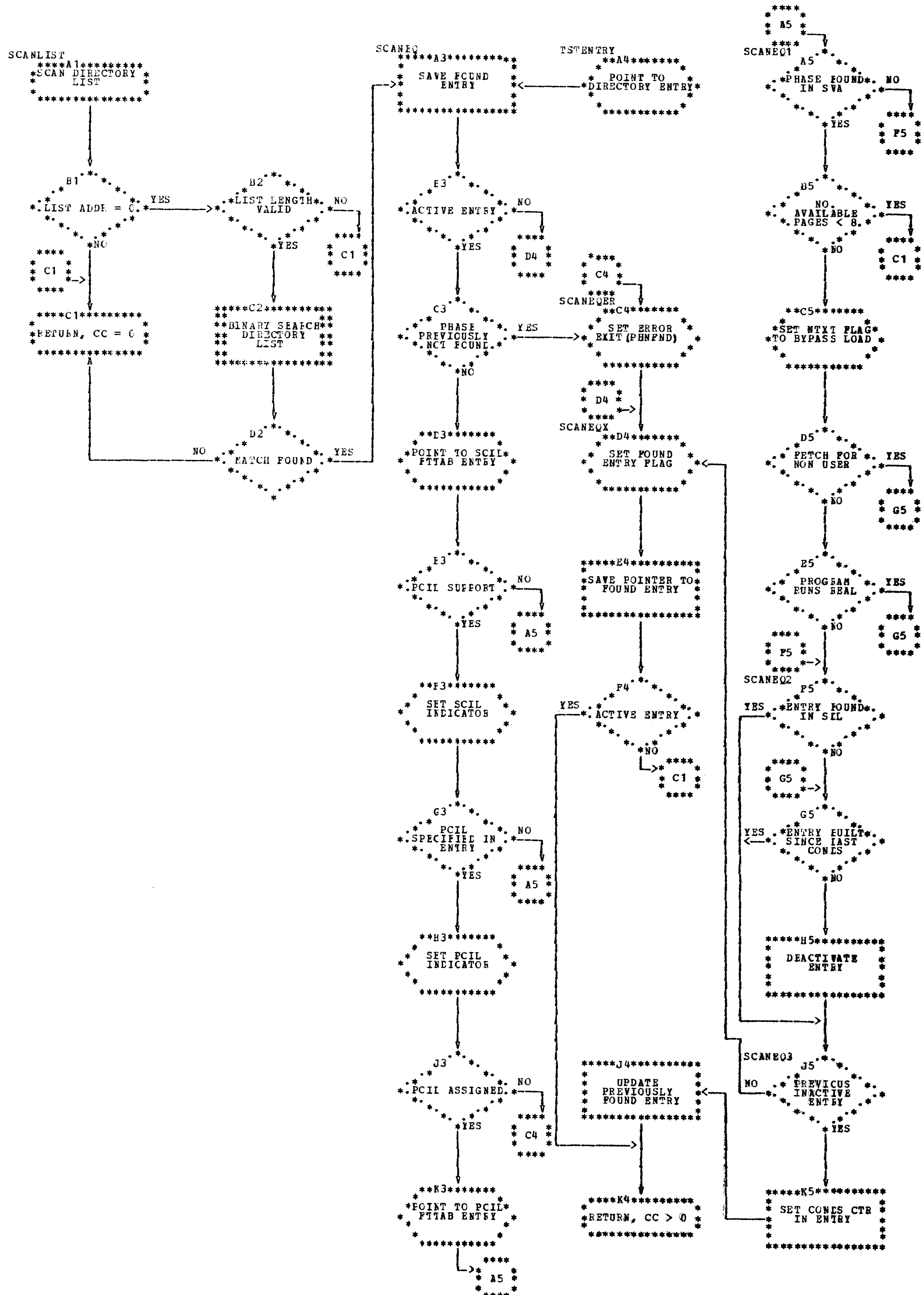


Chart B11. $$A$SUP1 - SGDFCH Macro, Scan Directory List for Equal Phase Name

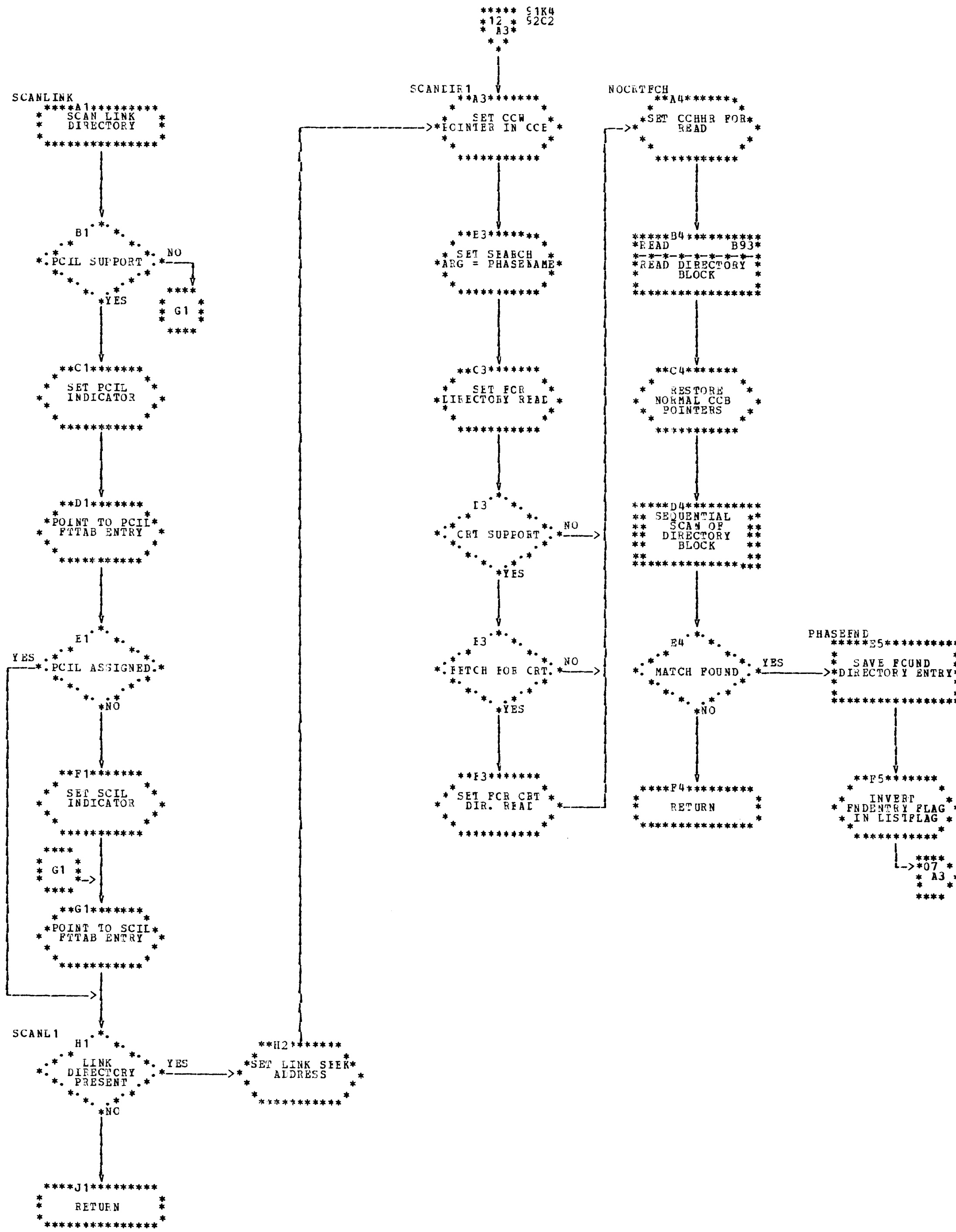


Chart B12. $$A$SUP1 - SGDFCH Macro, Scan Link Directory

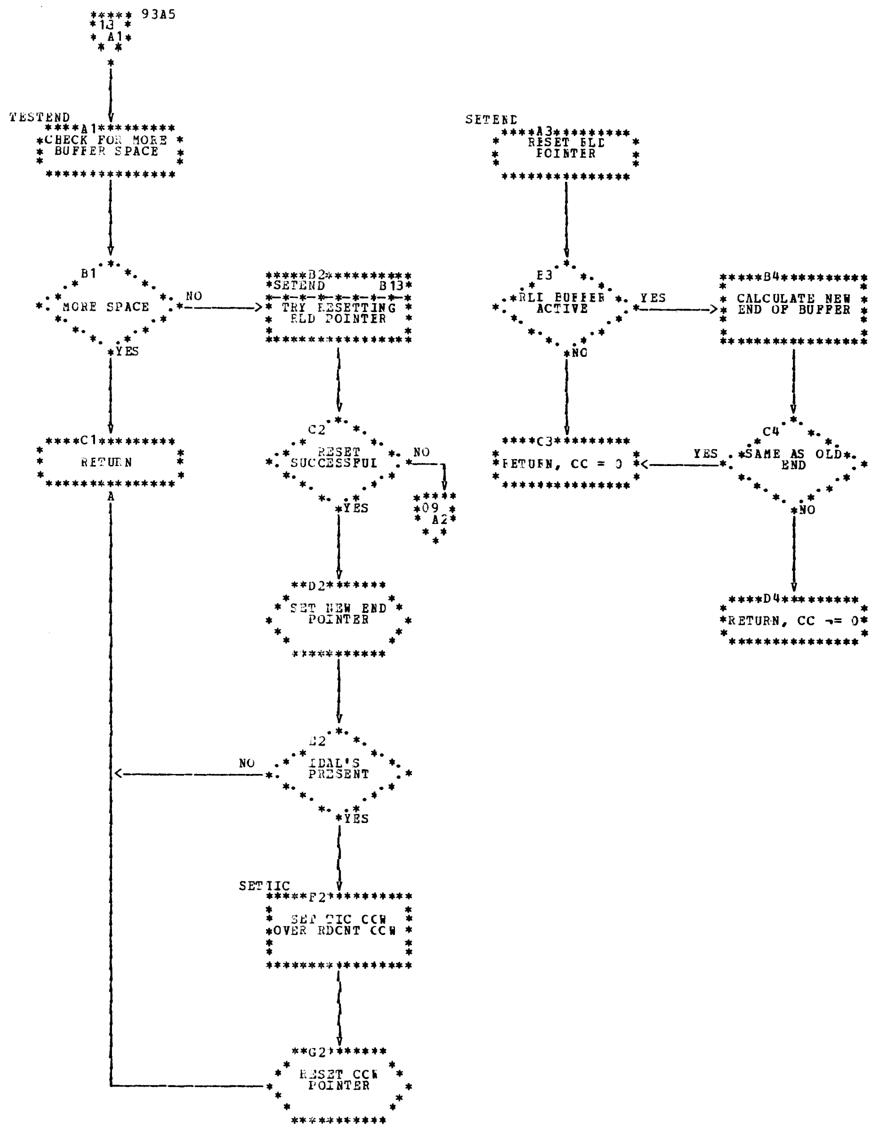


Chart B13. $$A$SUP1 - SGDFCH Macro, Check for More Buffer Space

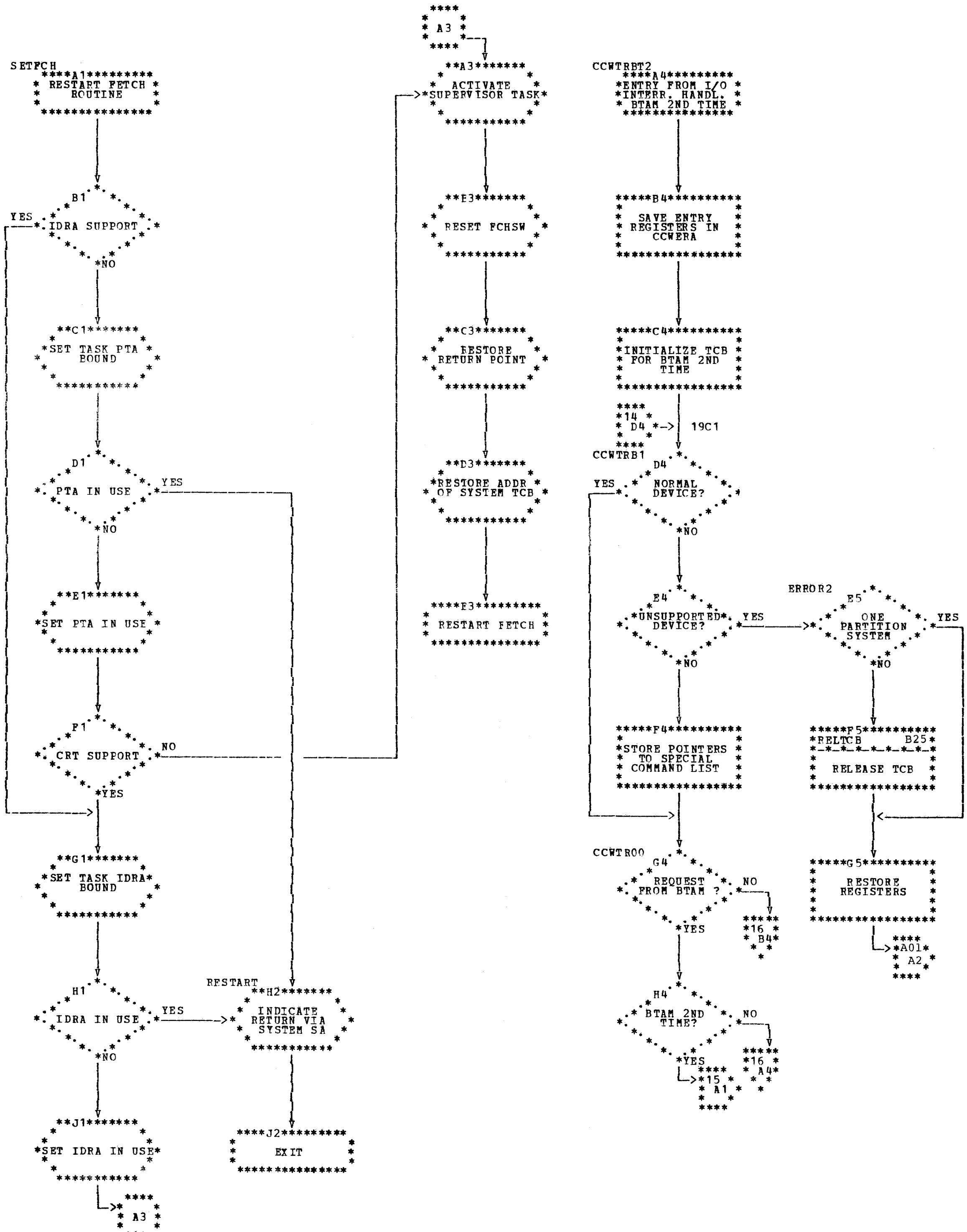


Chart B14. $$A$SUP1 - SGDFCH Macro, Restart Fetch Routine
SGCCWT Macro, Channel Program Translation

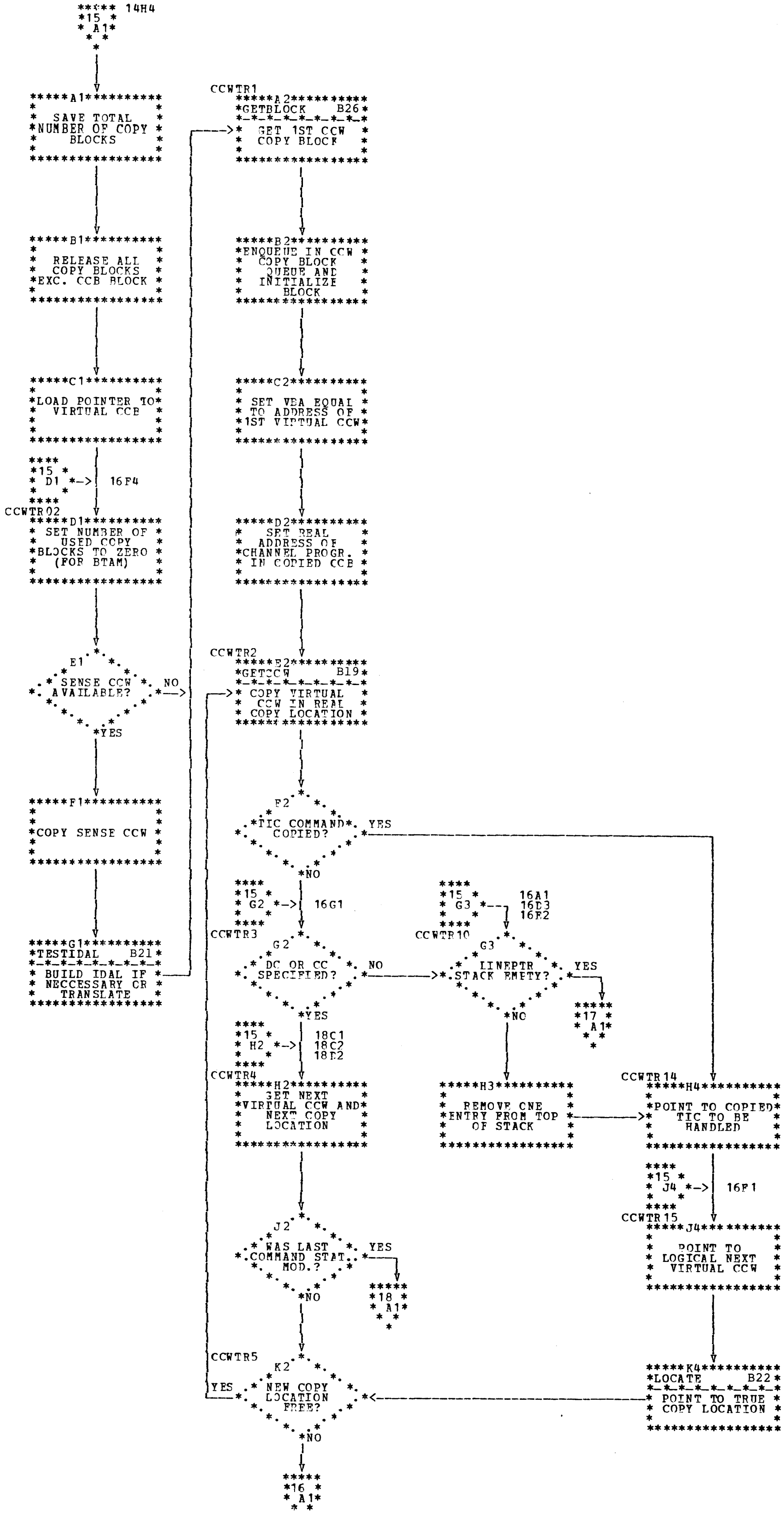


Chart B15. $$A$SUP1 - SGCCWT Macro, Channel Program Translation
Refer to Chart 14.2.

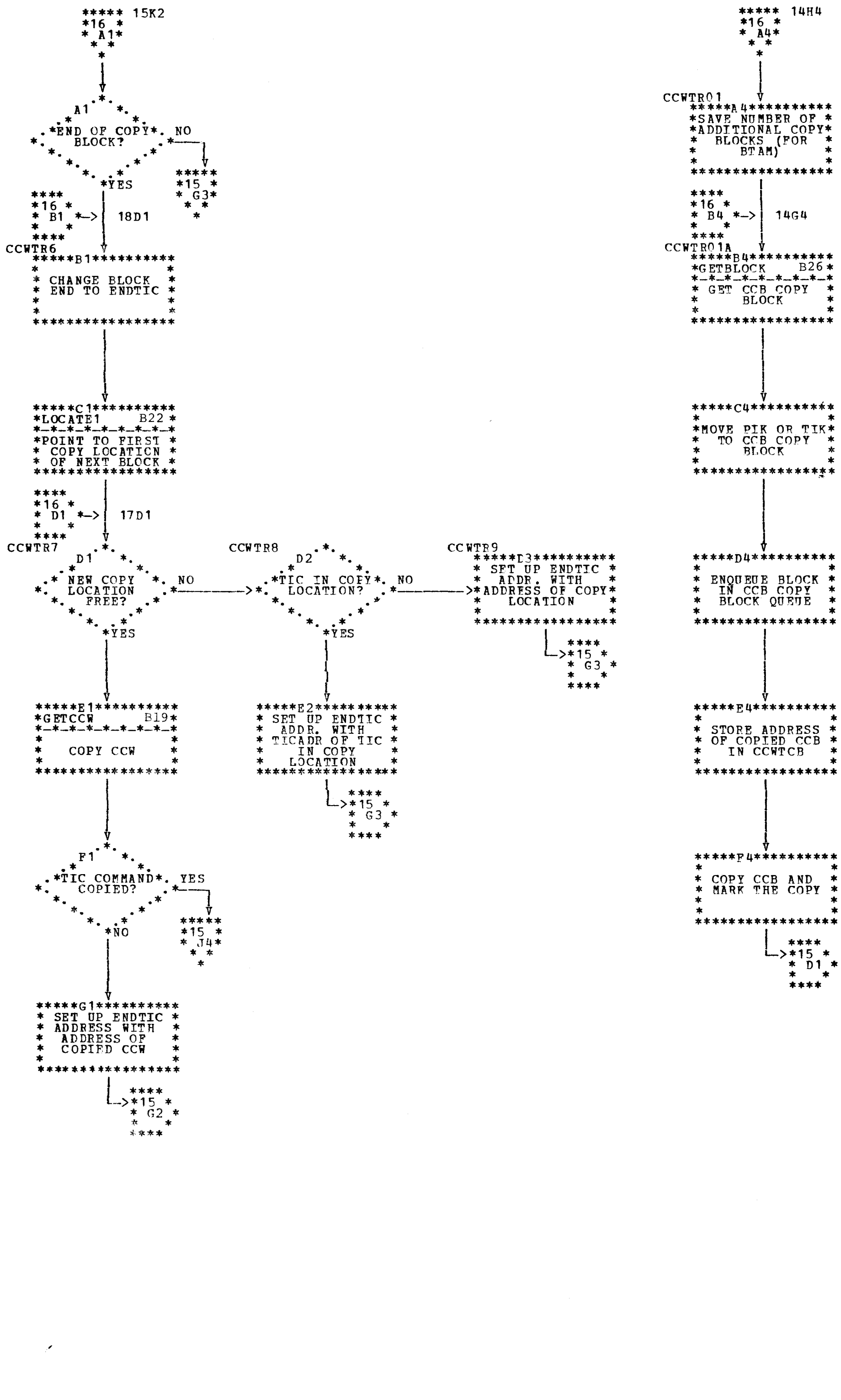


Chart B16. $$A$SUP1 - SGCCWT Macrc, Channel Prcgram Translaticn
Refer to Chart 14.2.

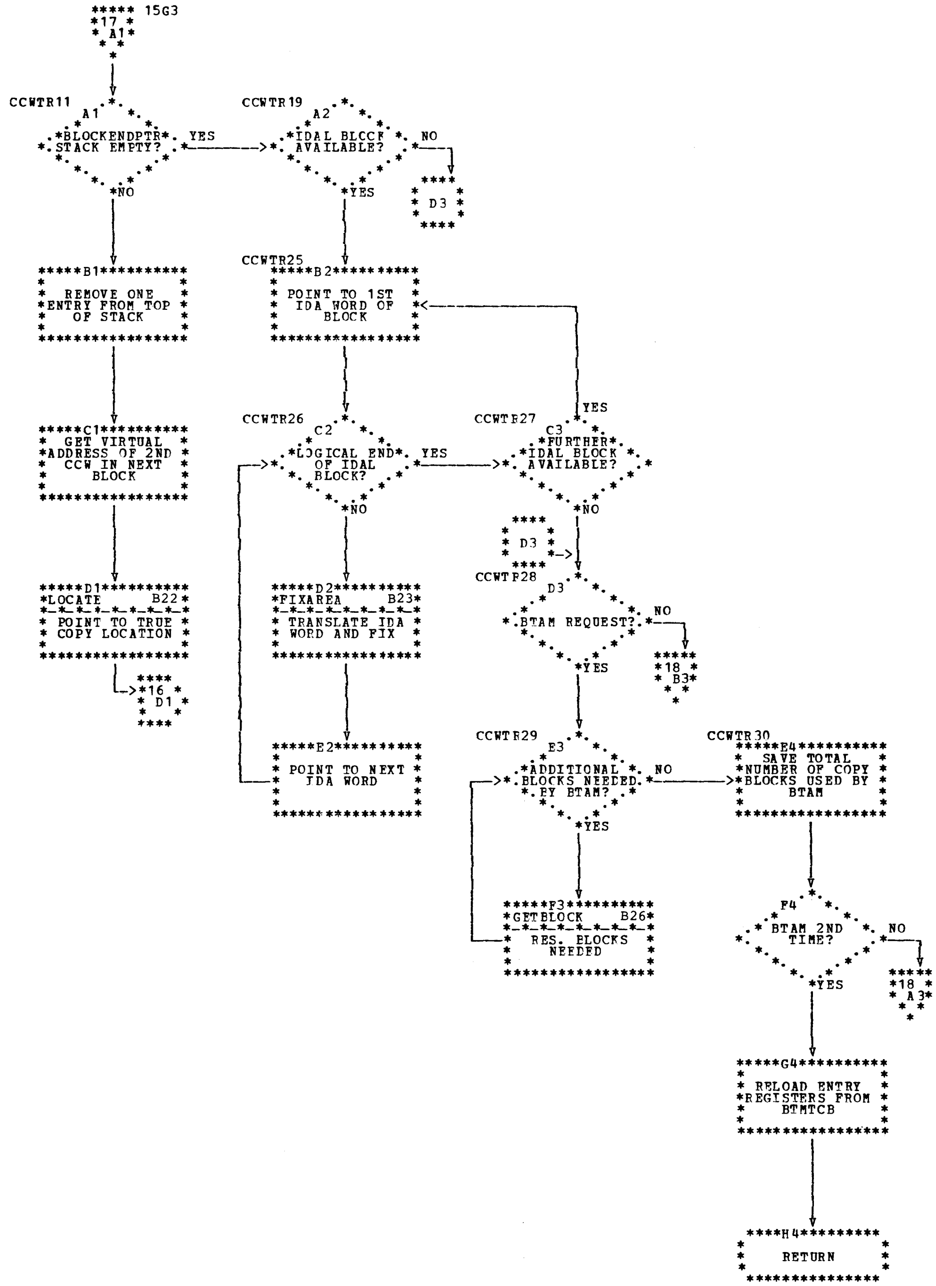


Chart B17. $$A$SUP1 - SGCCWT Macro, Channel Program Translation
Refer to Chart 14.7.

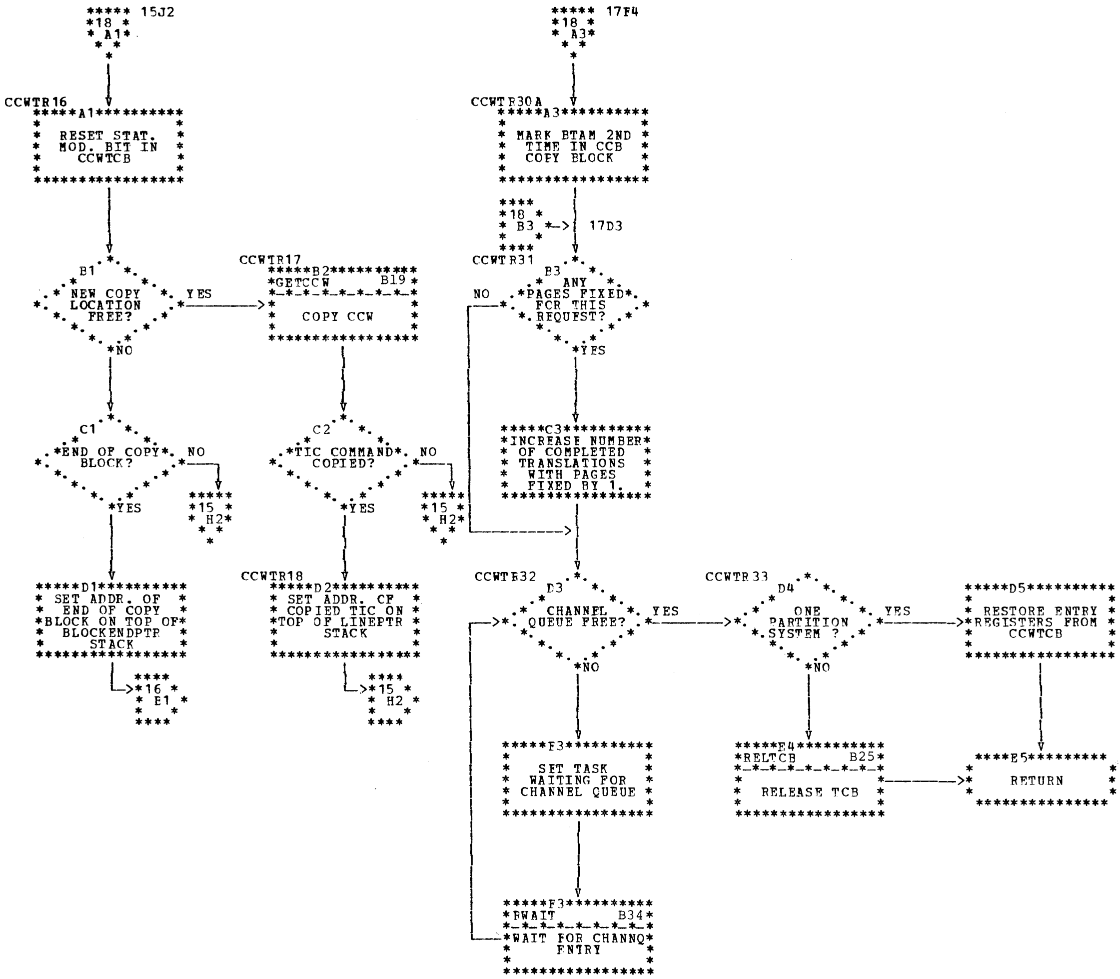


Chart B18. $$A$SUP1 - SGCCWT Macro, Channel Program Translation
Refer to Chart 14.6.

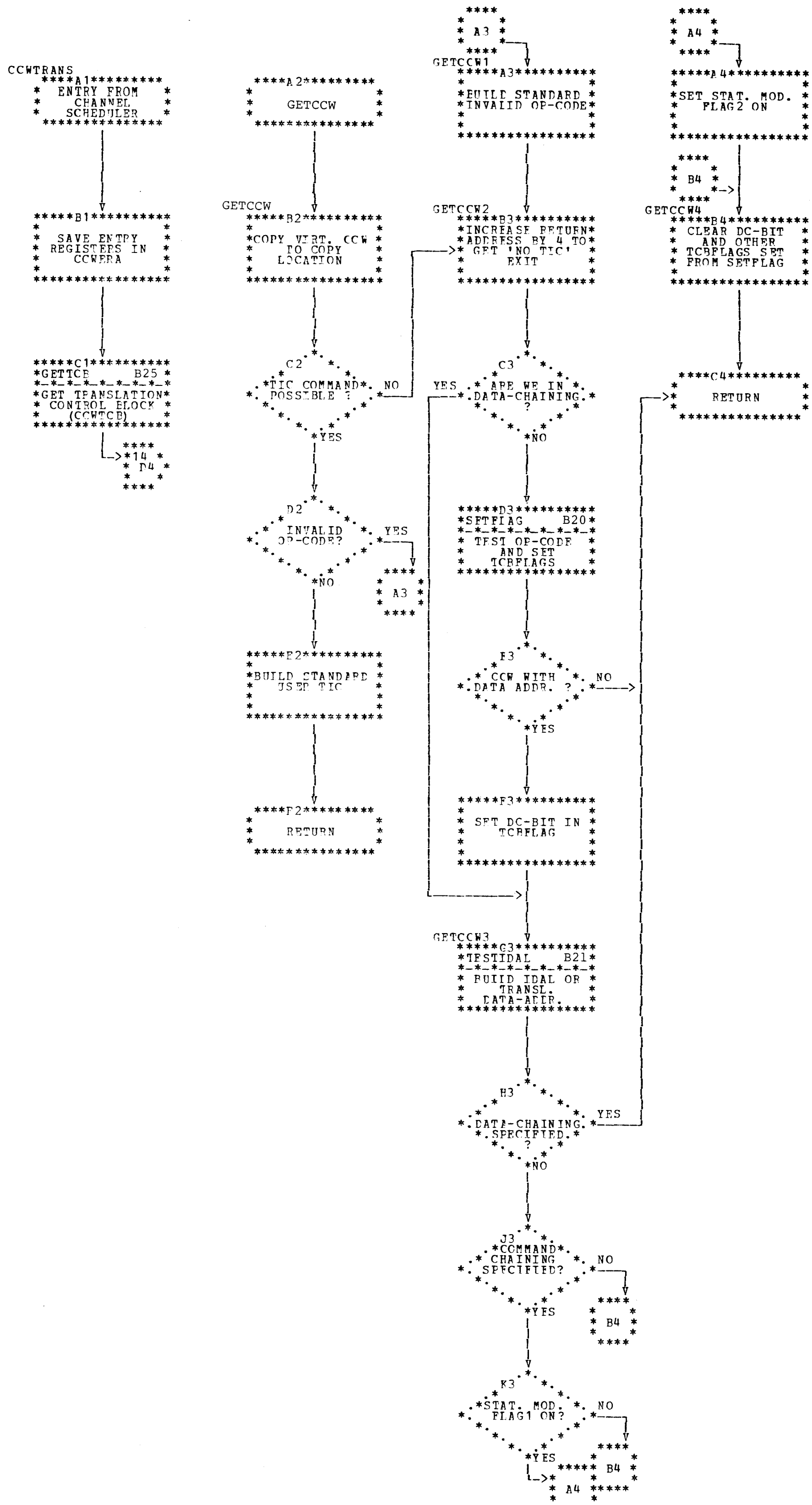


Chart B19. $$A$SUP1 - SGCCWT Macrc, Ccpy CCW
Refer to Chart 14.3.

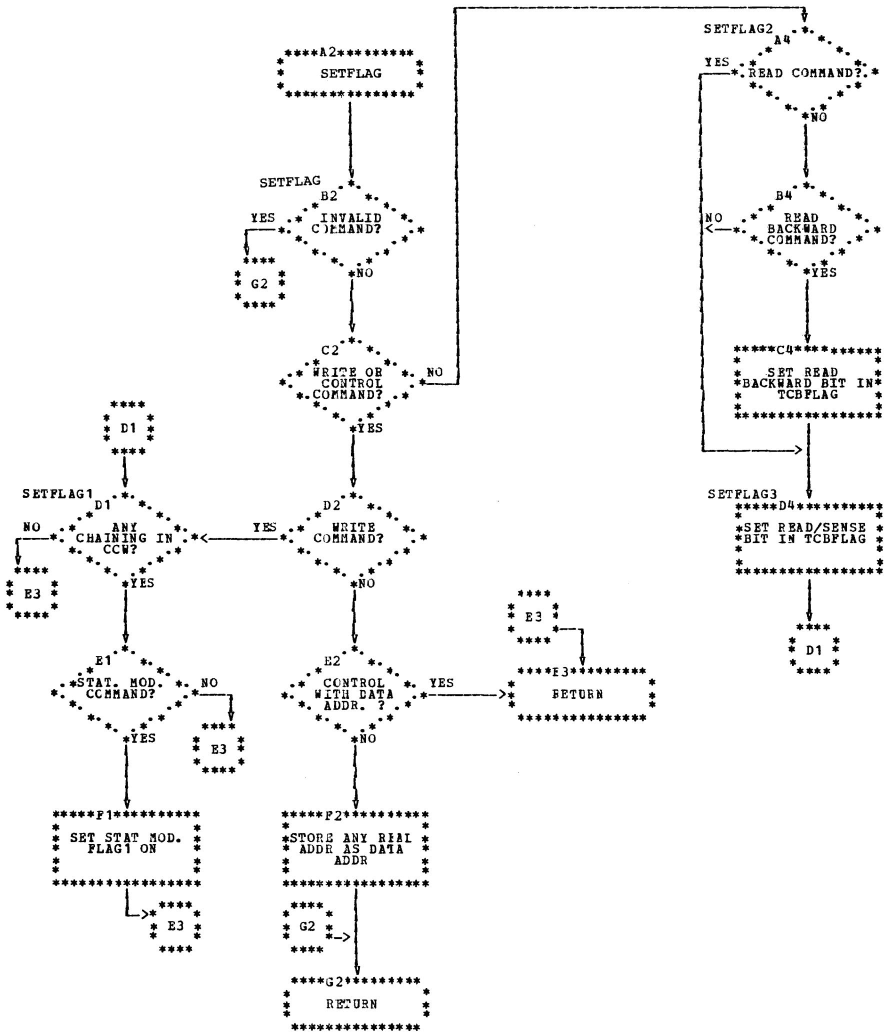


Chart B20. $$A$SUP1 - SGCCWT Macro, Set TCB Flags
Refer to Chart 14.3.

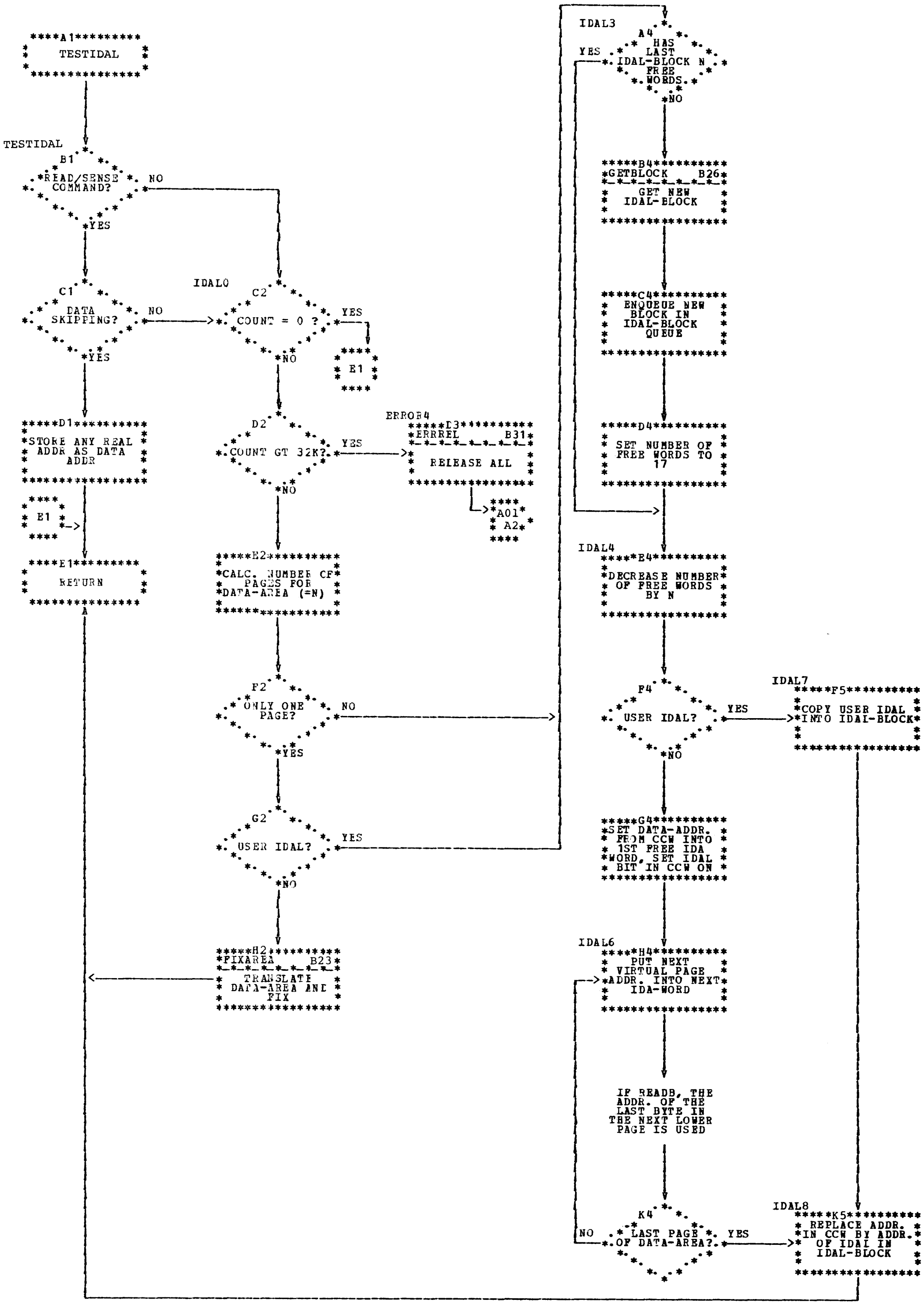


Chart B21. $$A$SUP1 - SGCCWT Macro, Build Indirect Address List
Refer to Chart 14.4.

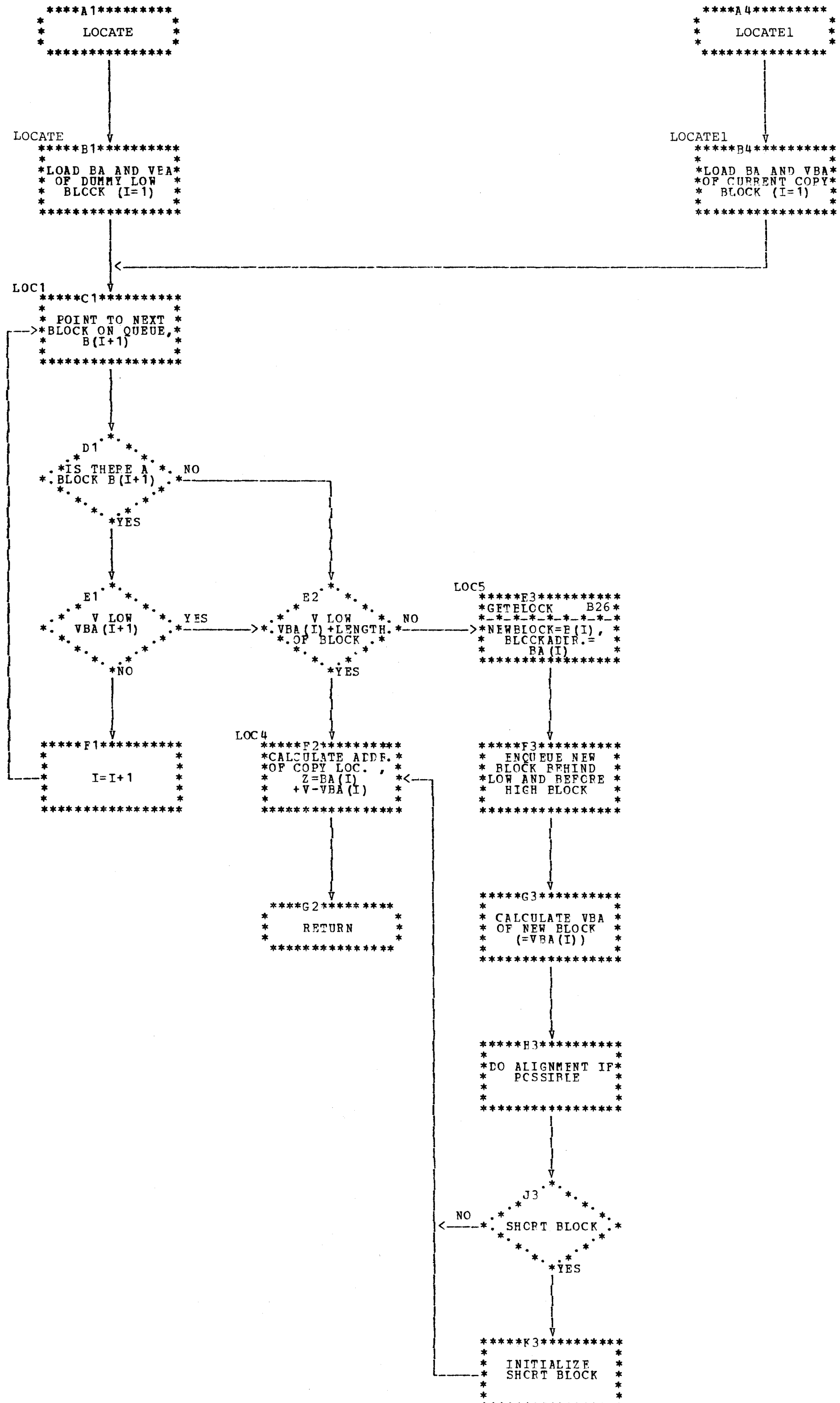


Chart B22. $$A$SUP1 - SGCCWT Macro, Find Address of Copy Location
Refer to Chart 14.5.

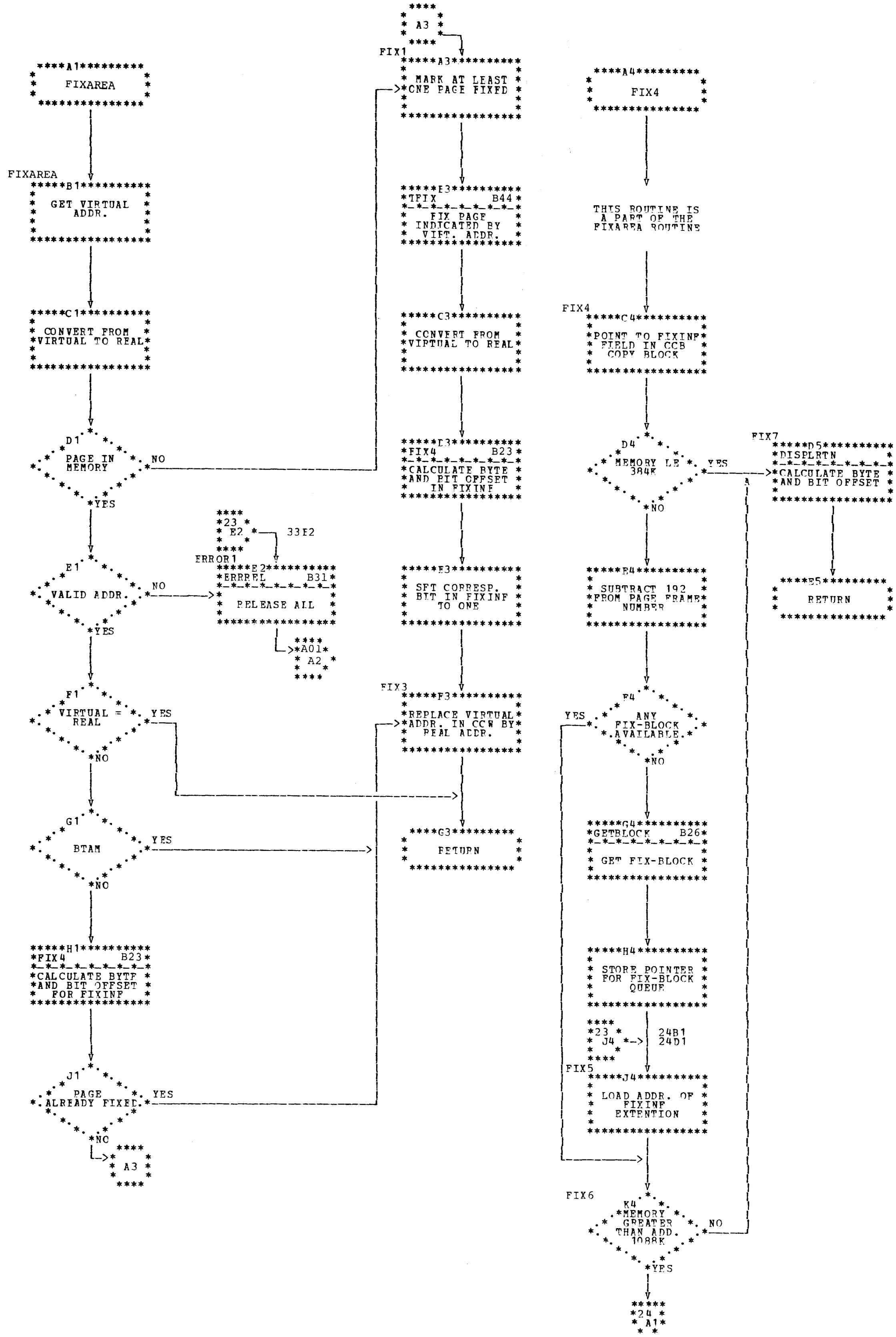


Chart B23. $$A$SUP1 - SGCCWT Macrc, TFIX Data Area
Refer to Chart 14.4.

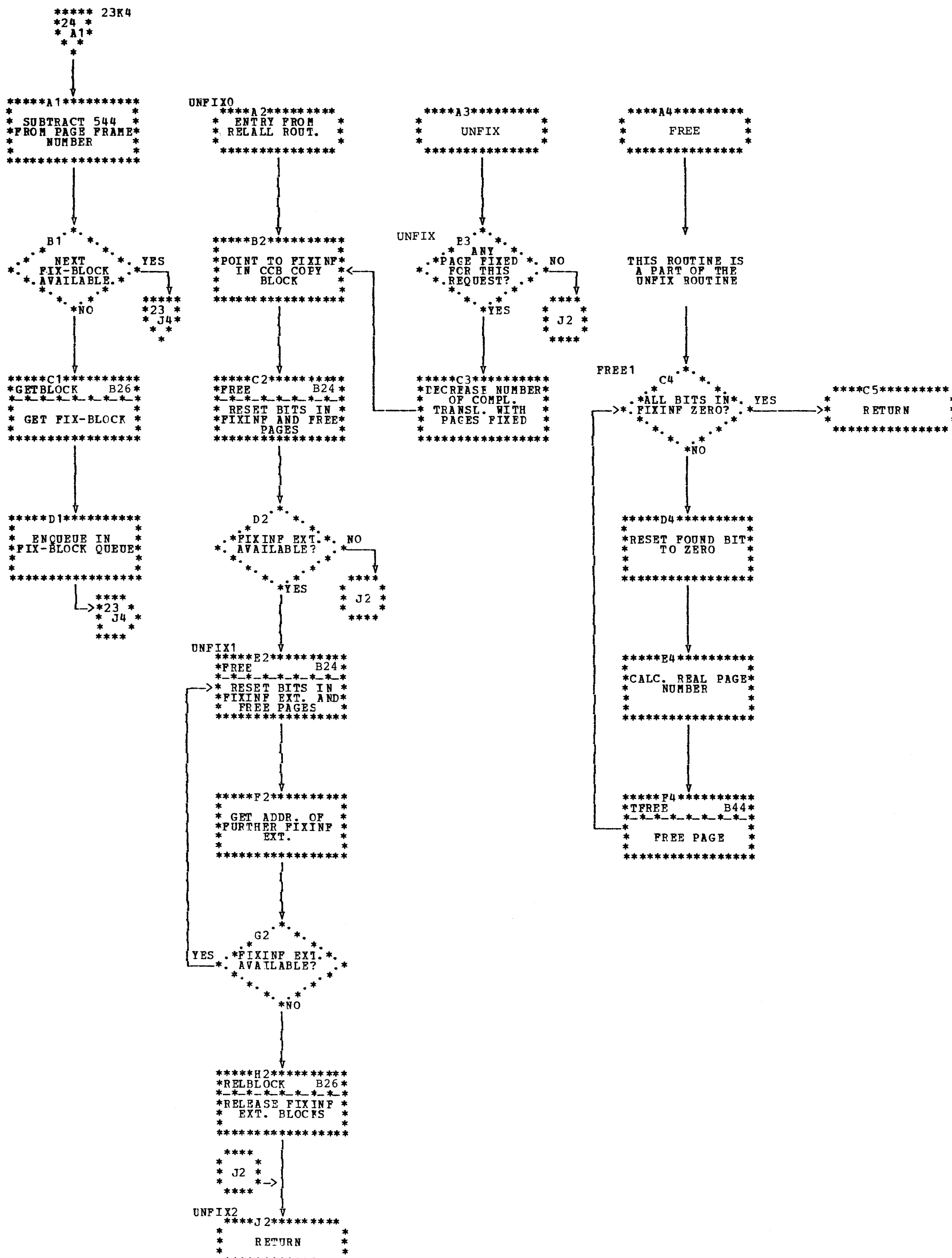


Chart B24. $$A$SUP1 - SGCCWT Macro, TFREE Data Area
Refer to Chart 14.9.

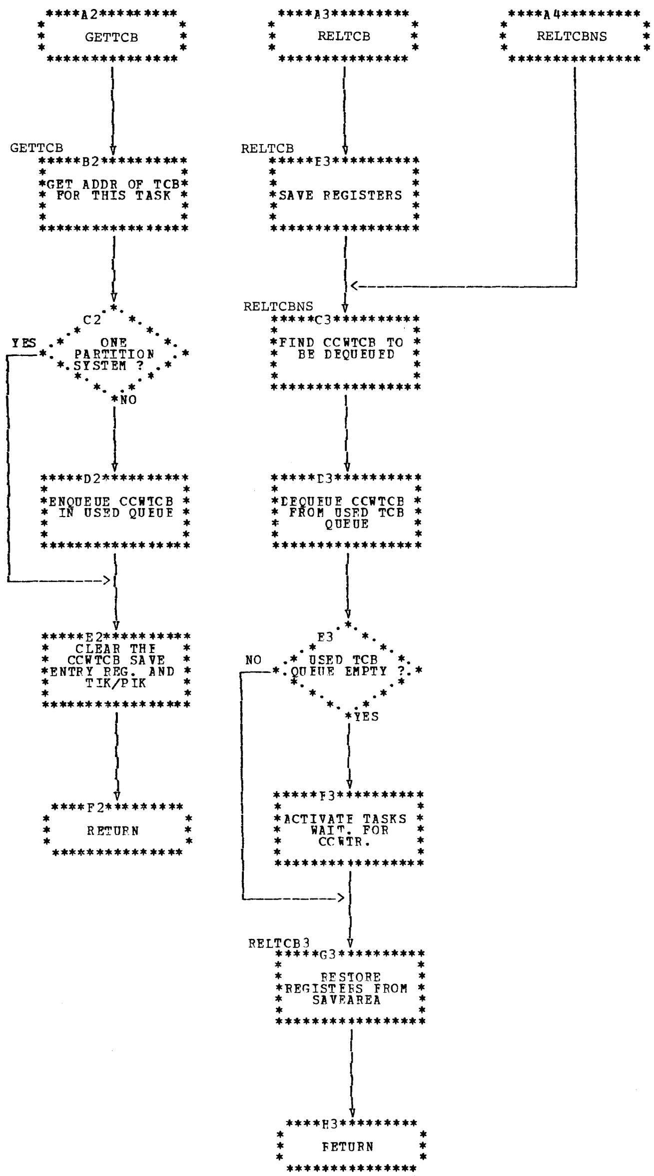


Chart B25. $$A$SUP1 - SGCCWT Macrc, Get/Release TCB
Refer to Chart 14.1.

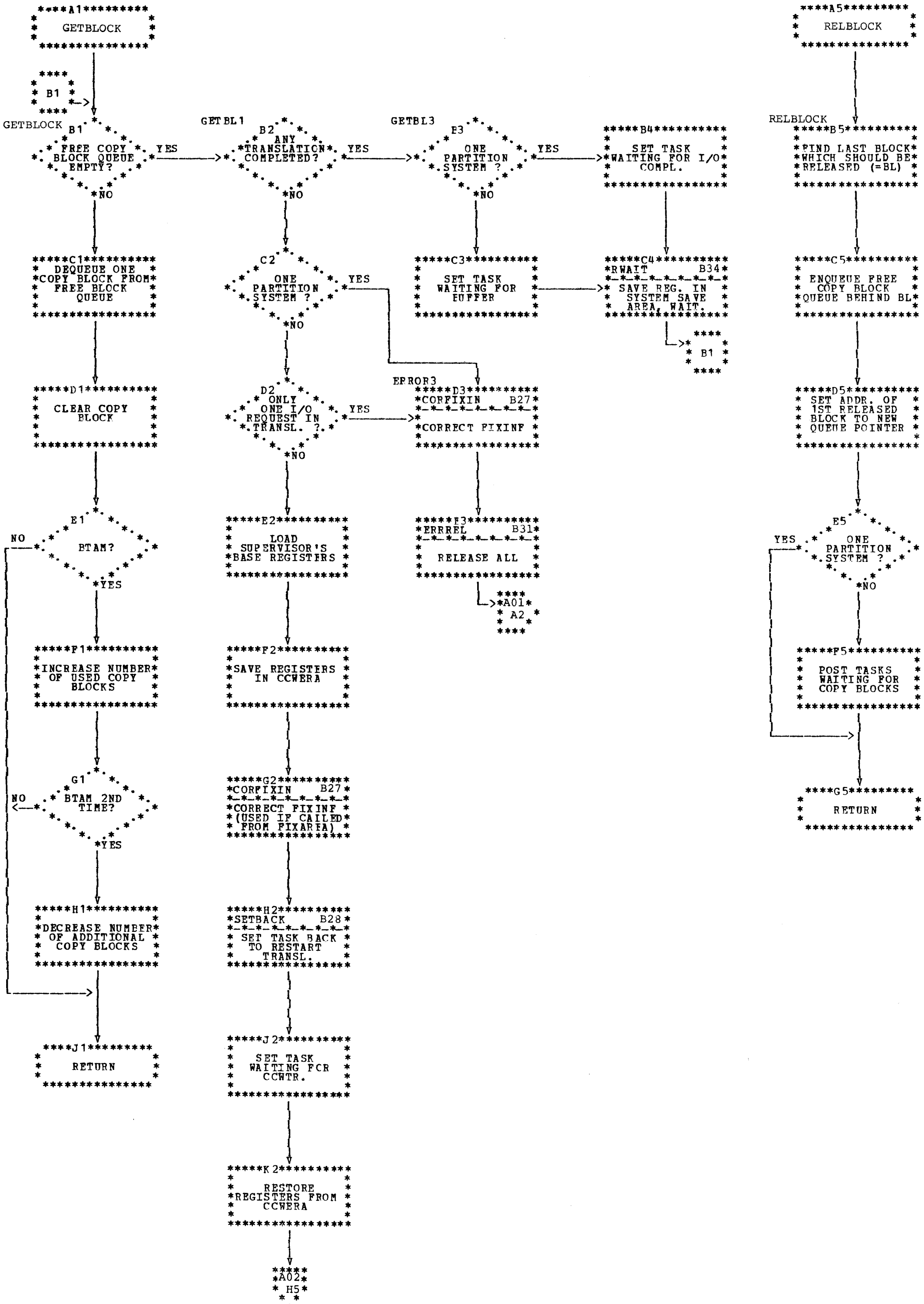


Chart B26. $$A$SUP1 - SGCCWT Macrc, Get/Release Ccpy Blcck
Refer to Chart 14.3.

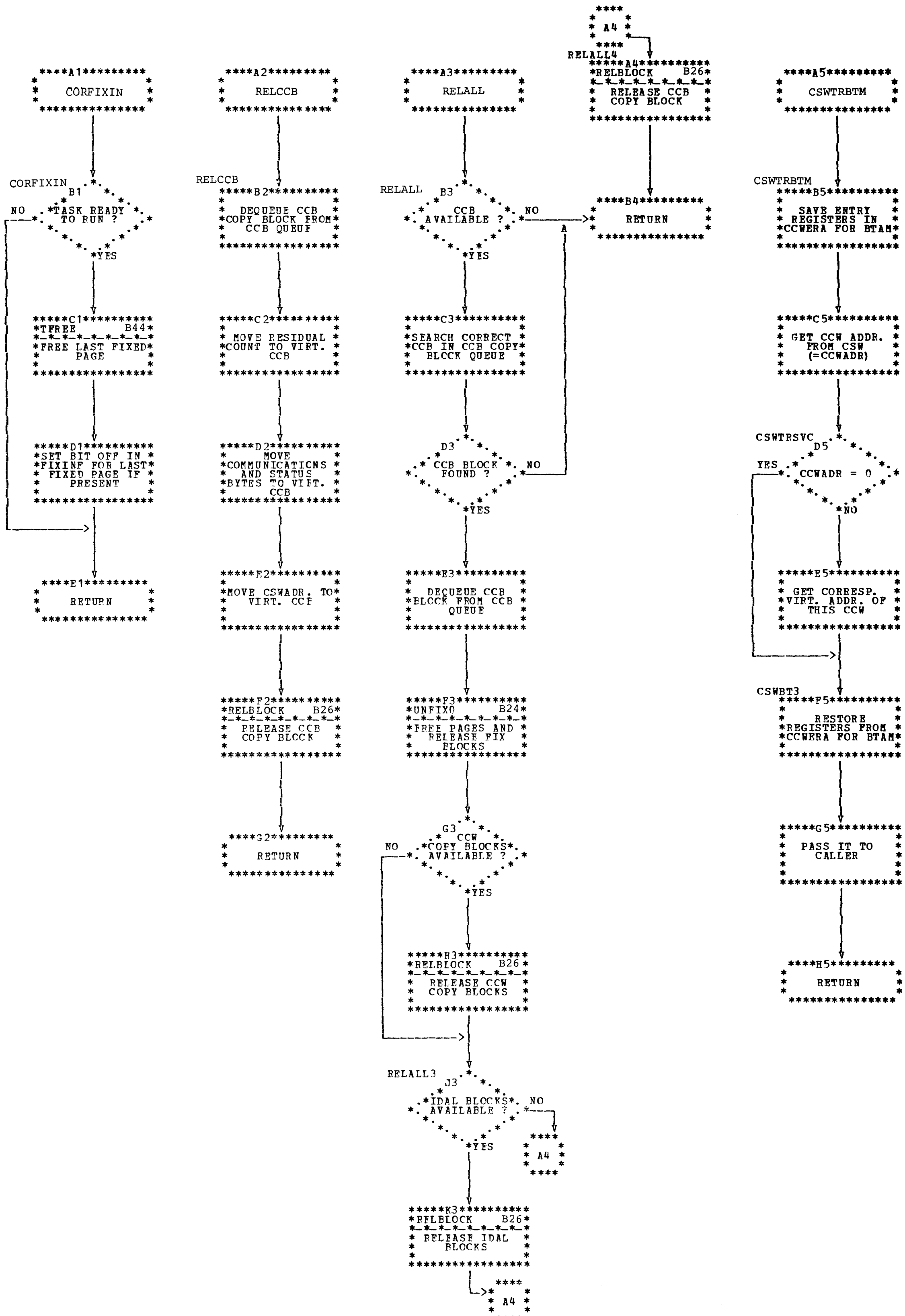


Chart B27. $$A$SUP1 - SGCCWT Macro, Release Blocks, Get Virtual Address of CCW
Refer to Charts 14.8 and 14.11.

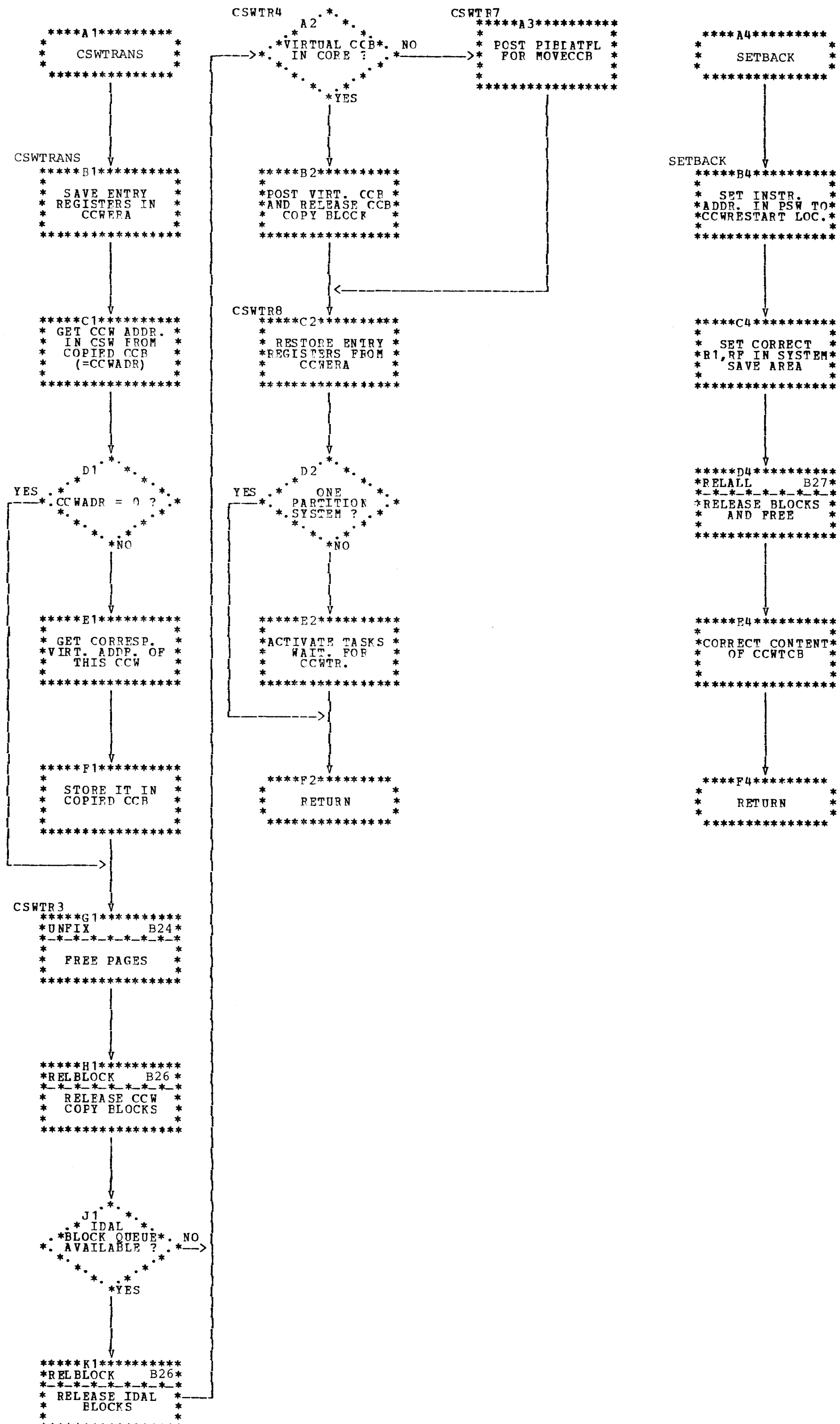


Chart B28. $$A$SUP1 - SGCCWT Macro, Channel Program Retranslation
Refer to Charts 14.8.and 14.9.

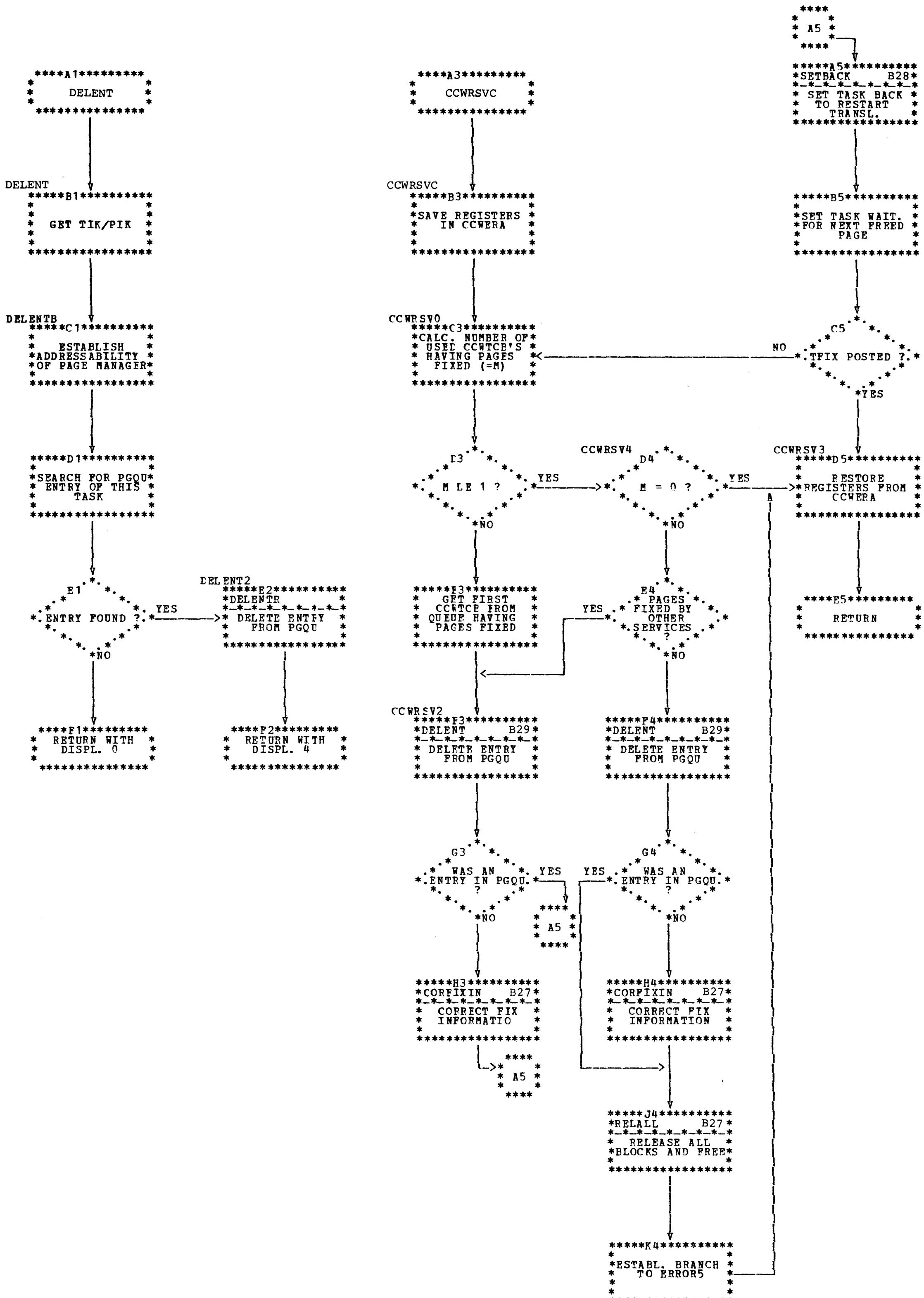


Chart B29. $$A$SUP1 - SGCCWT Macro, Reset Channel Program Translation
Refer to Chart 14.10.

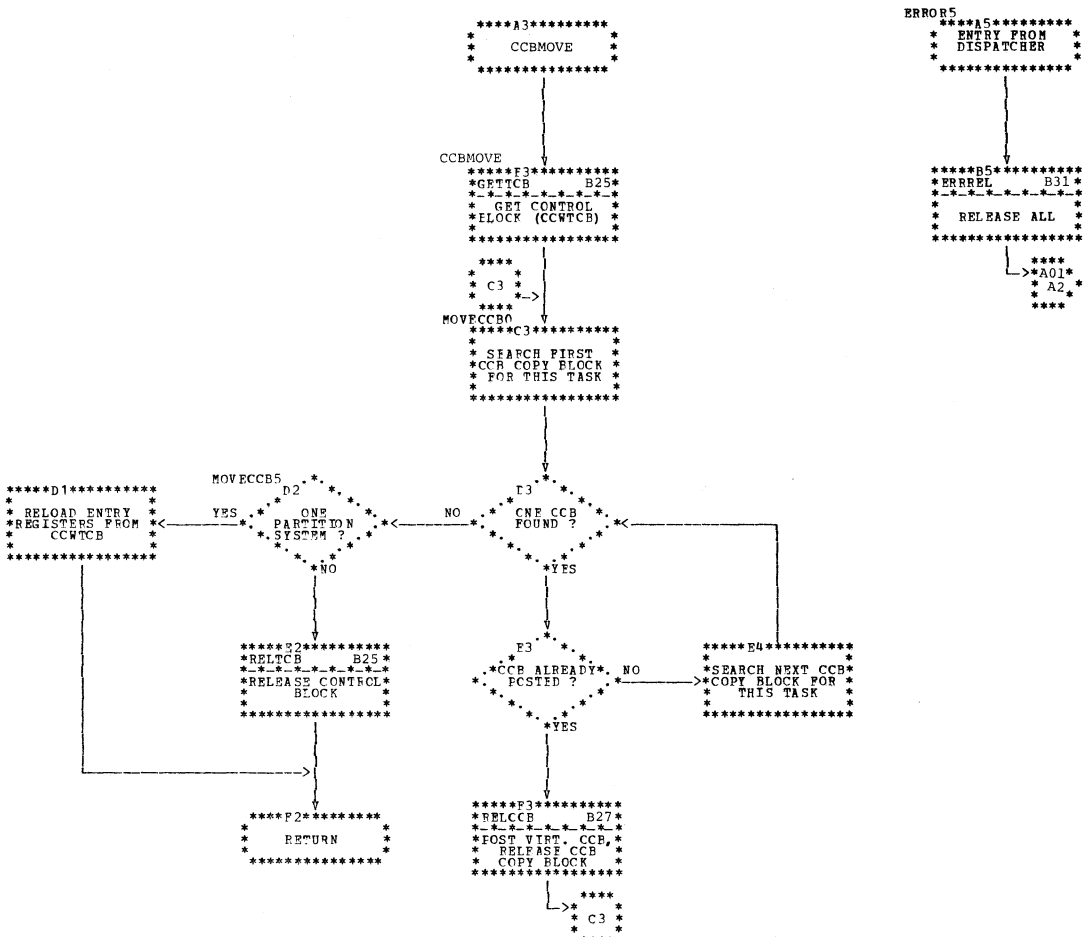


Chart B30. $$A$SUP1 - SGCCWT Macro, Move Copied CCB to Virtual CCB
Refer to Chart 02.3.

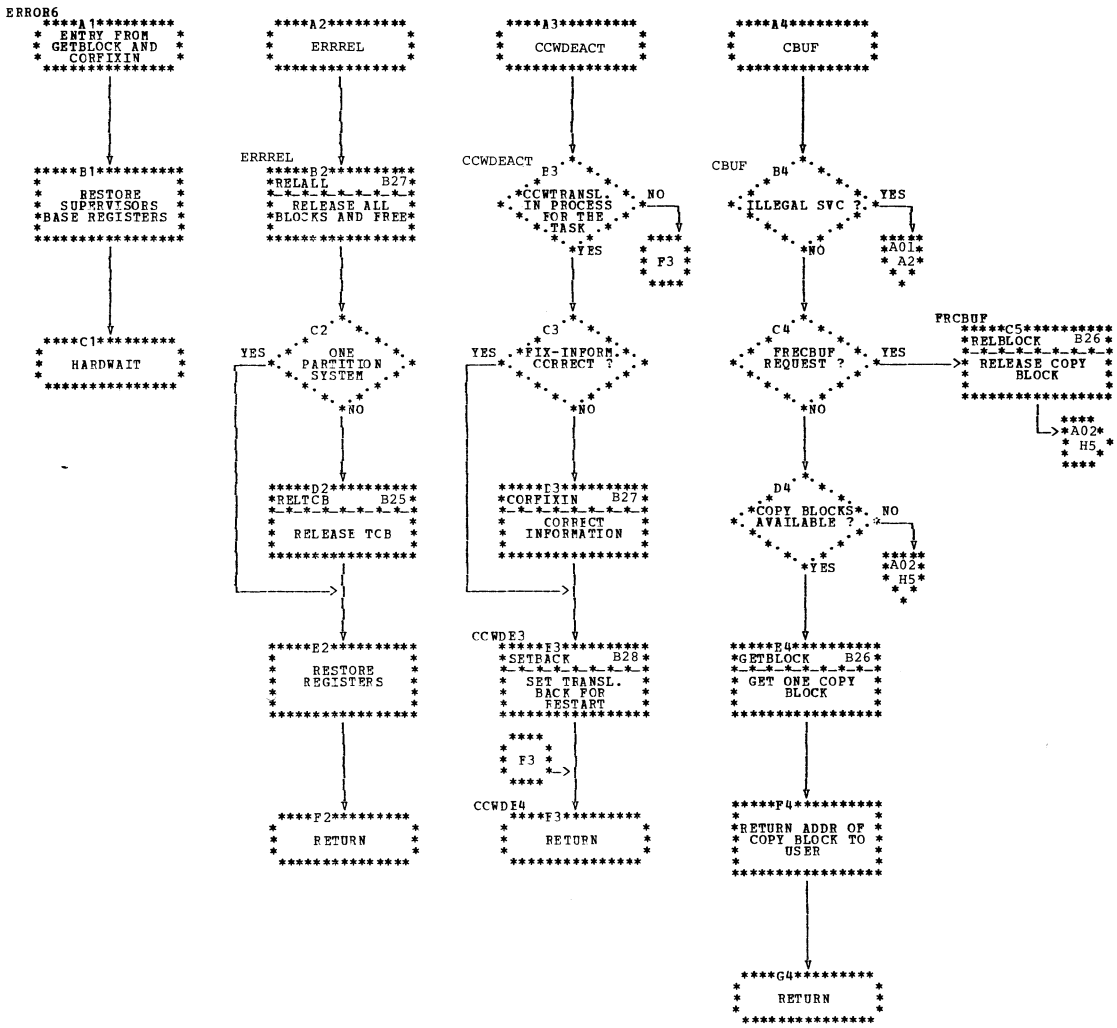


Chart B31. $$A$SUP1 - SGCCWT Macrc, Reset Translaticn, GETCBUF, FREECBUF
Refer to Charts 13.5 and 14.11.

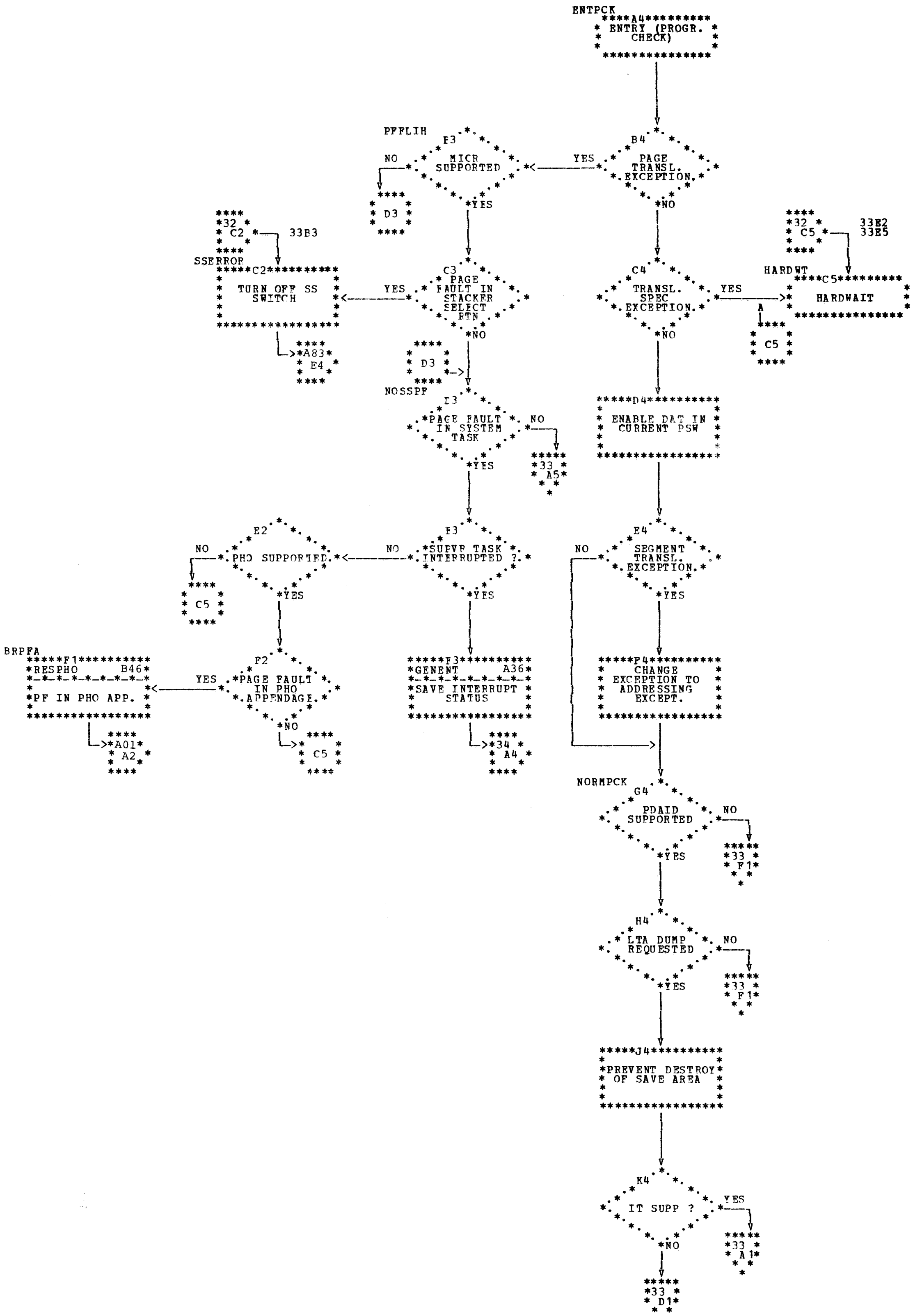


Chart B32. $$A$SUP1 - SGPMGR Macro, Entry Point for Program Check
Refer to Chart 03.1.

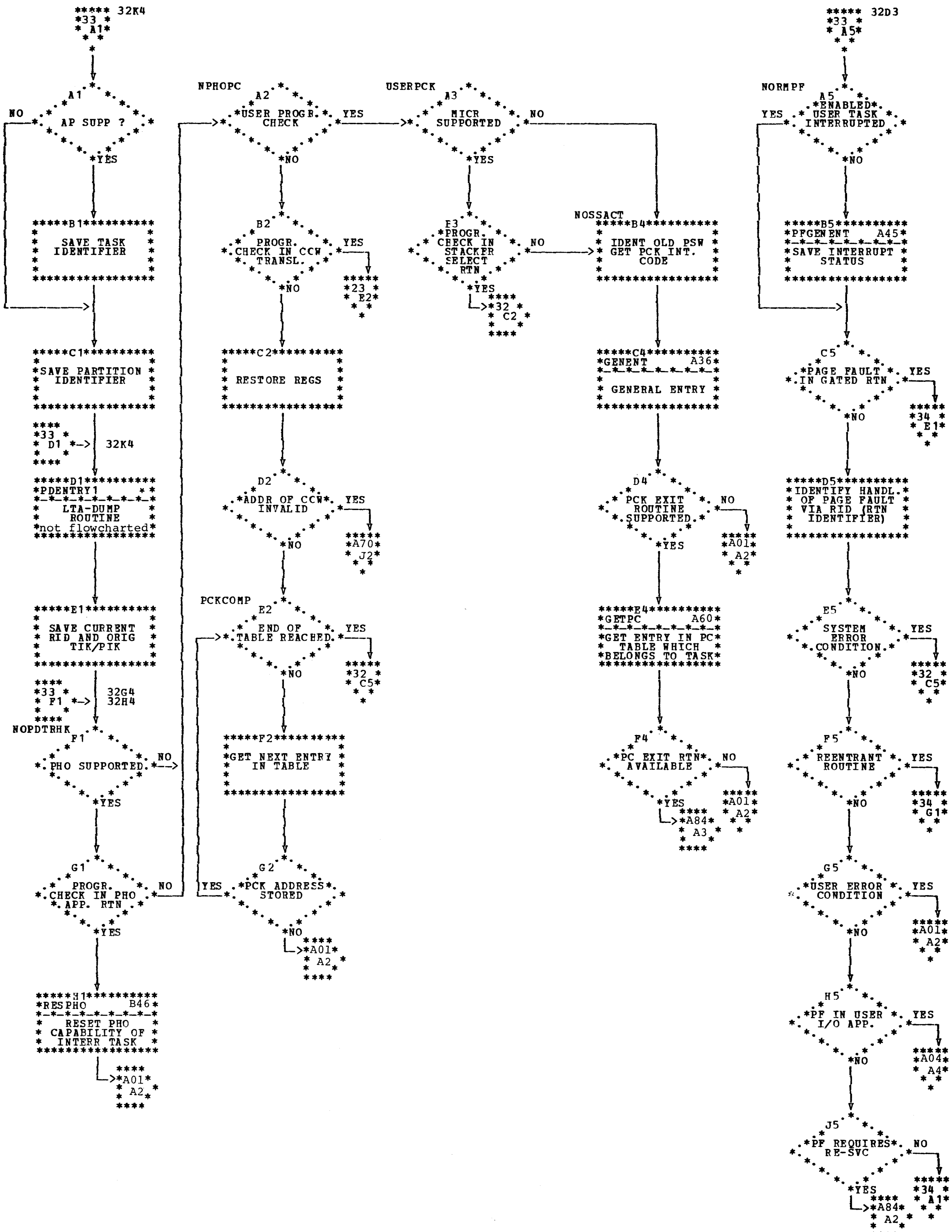


Chart B33. $$A$SUP1 - SGPMGR Macrc, Prograr Check Entry Routine
Refer to Chart 03.1.

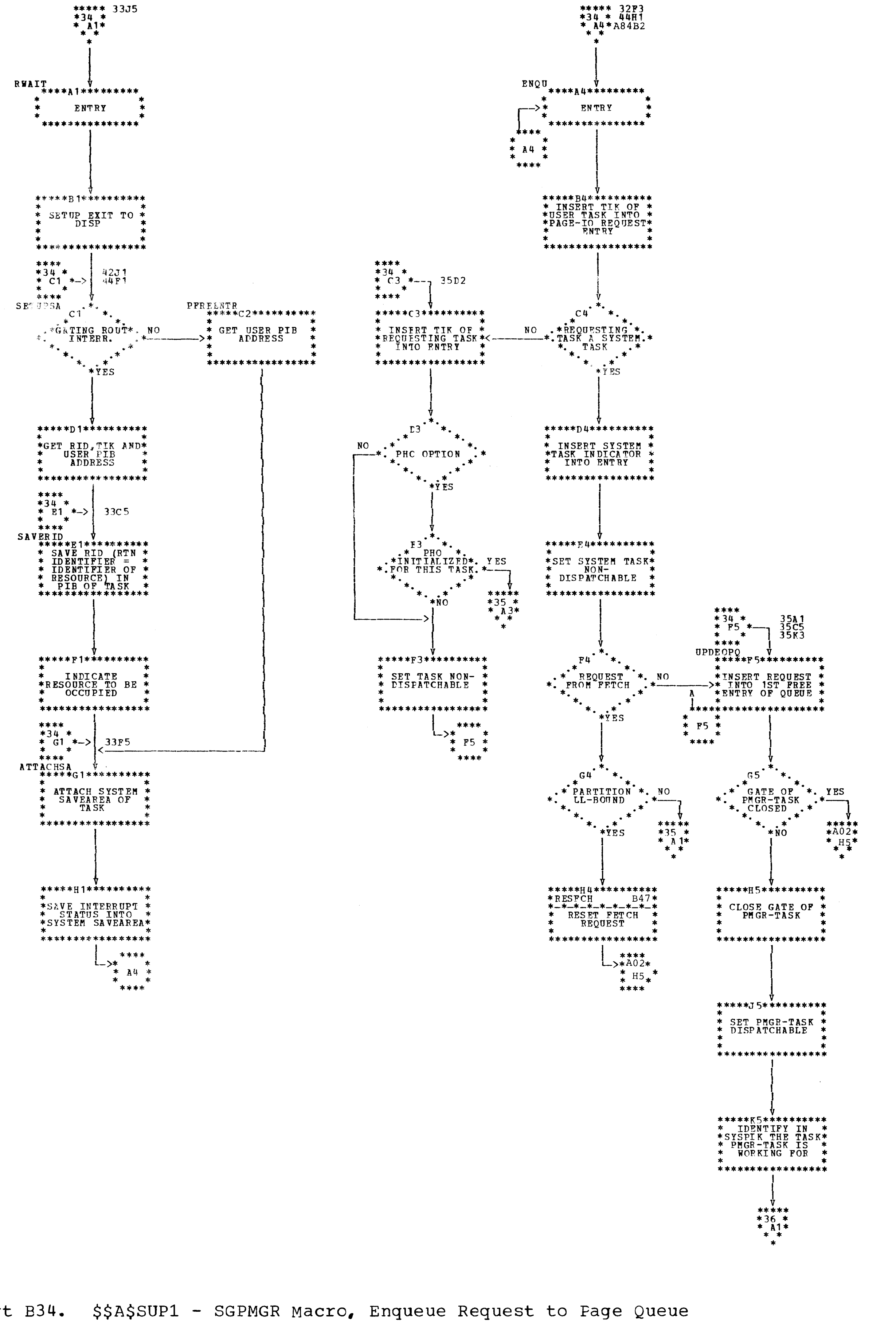


Chart B34. $$A$SUP1 - SGPMGR Macro, Enqueue Request to Page Queue
Refer to Chart 13.1.

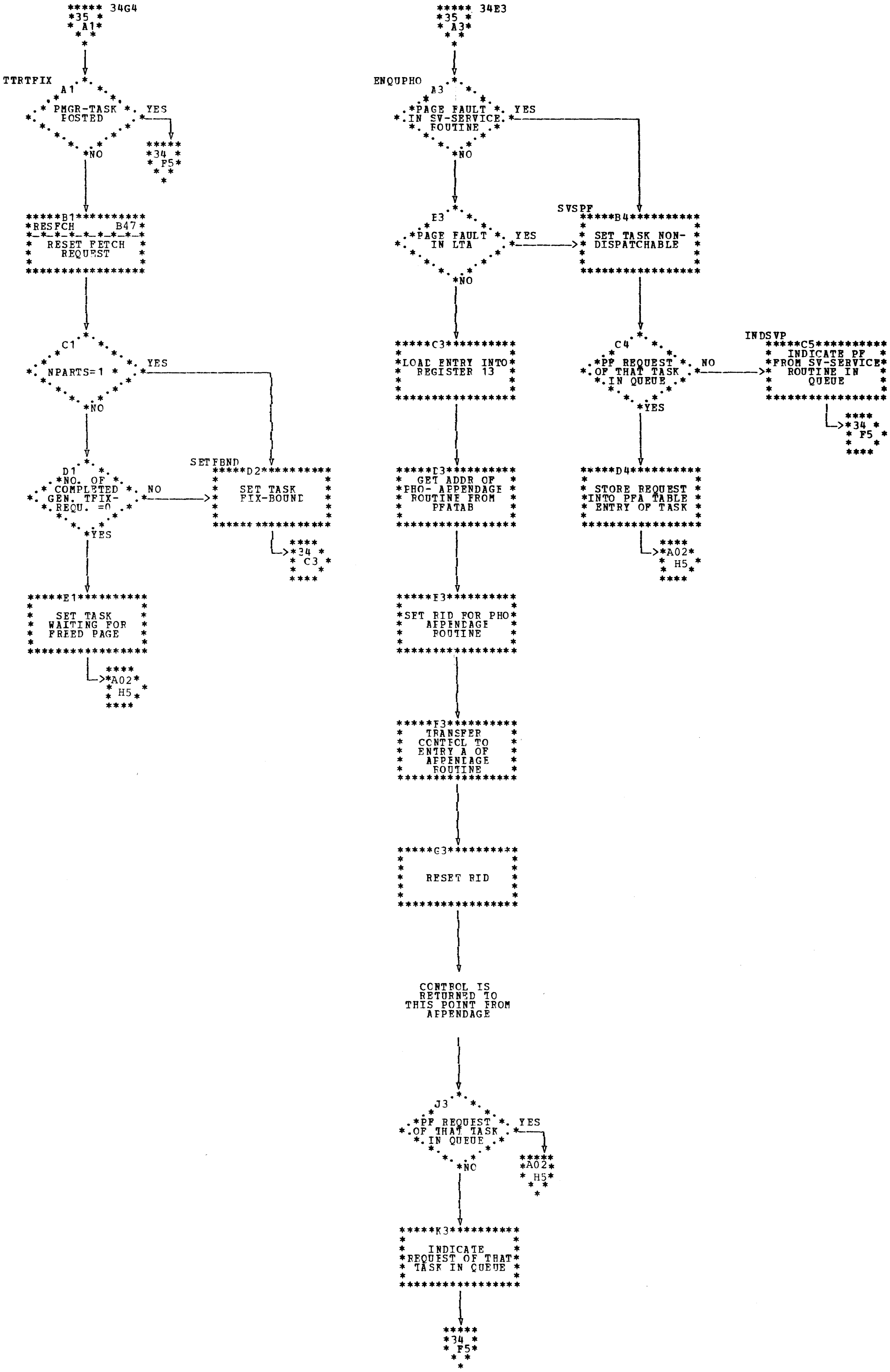


Chart B35. $$A$SUP1 - SGPMGR Macrc, Page Fault Handling Overlap Interface
Refer to Chart 13.1.

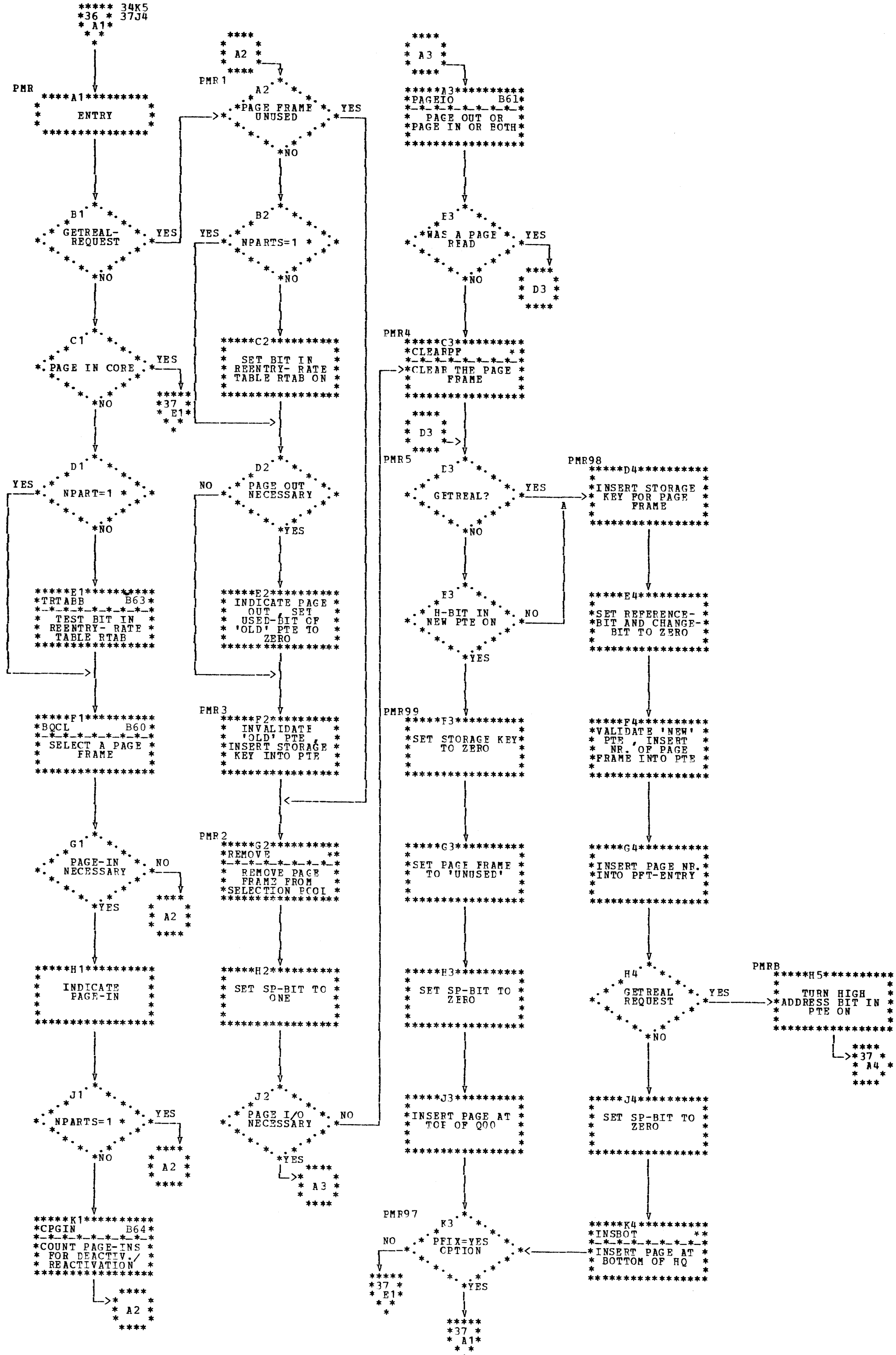


Chart B36. $$A$SUP1 - SGPMGR Macro, Page Manager System Task
Refer to Chart 13.2.

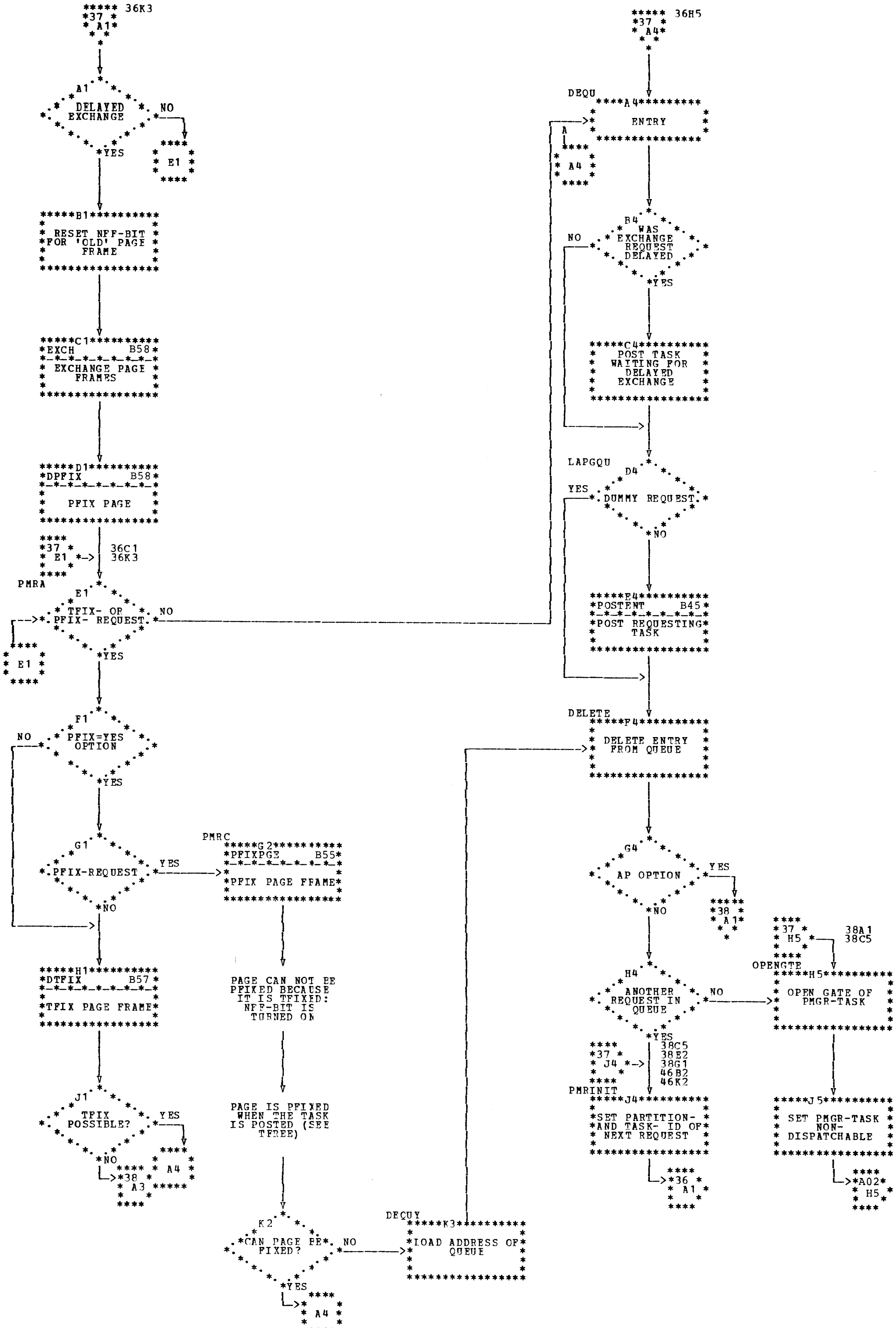


Chart B37. $$A$SUP1 - SGPMGR Macrc, Dequeue Page Queue Entry
Refer to Chart 13.3.

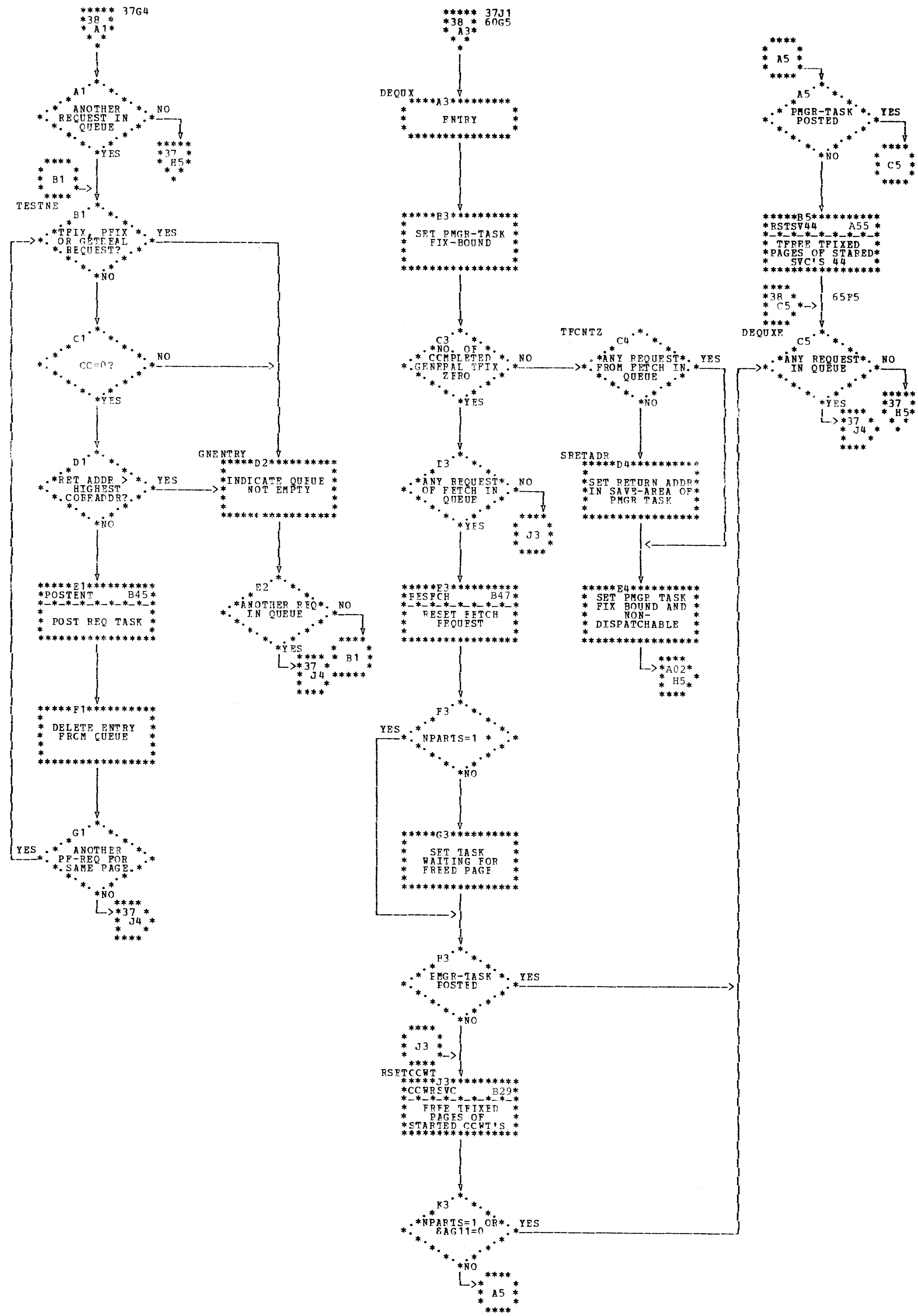


Chart B38. $$A$SUP1 - SGPMGR Macro, Handle No Page Frames Available Condition
Refer to Chart 13.4.

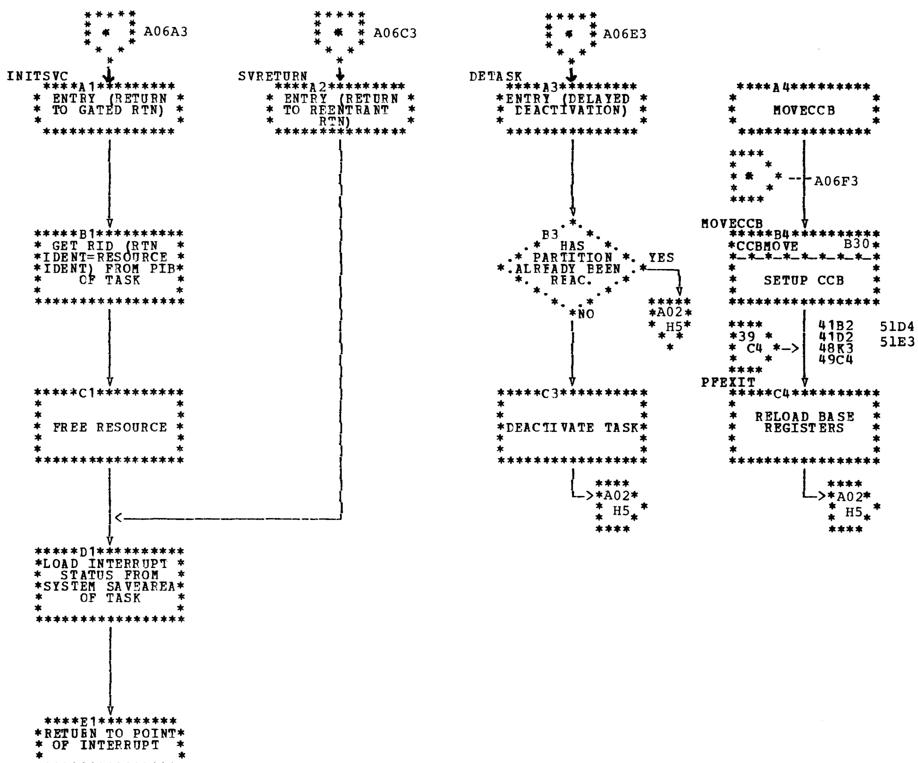


Chart B39. $$A$SUP1 - SGPMGR Macrc, Return to Supervisor Routine
Refer to Charts 02.3 and 13.4.

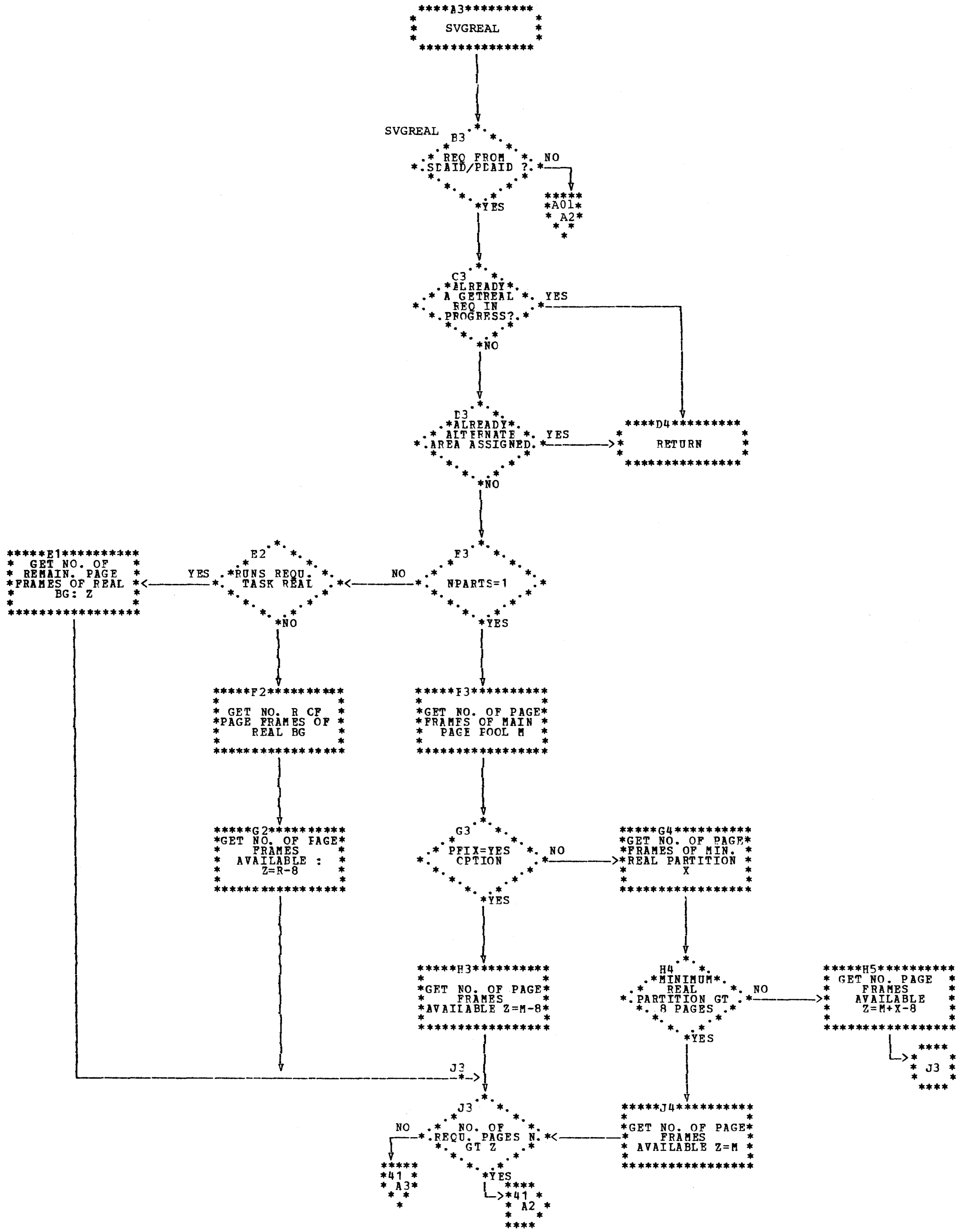


Chart B40. $$A$SUP1 - SGPMGR Macrc, GETREAL (SVC 55) Rcutine
Refer to Chart 13.15.

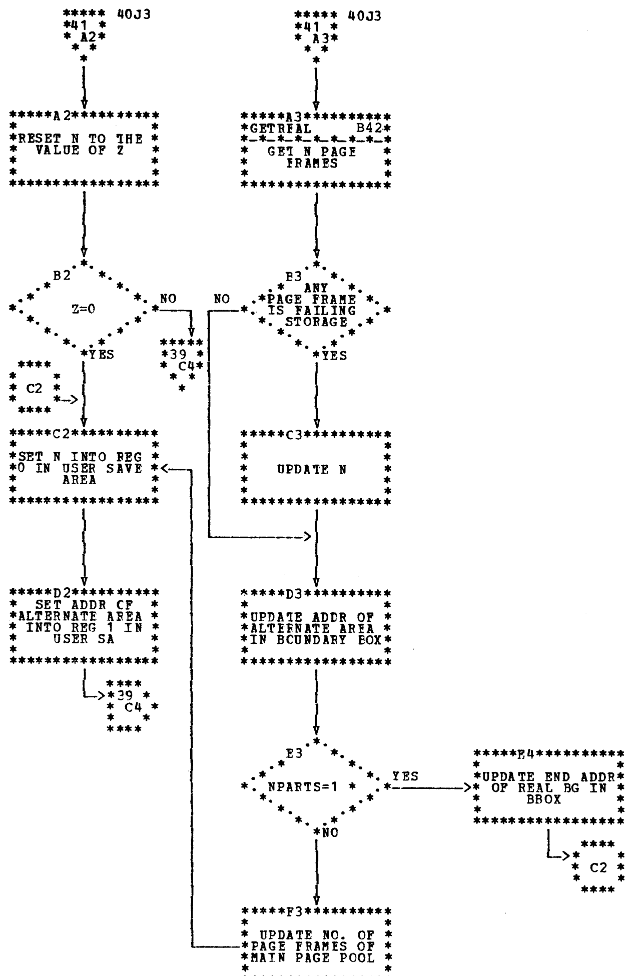


Chart B41. $$A$SUP1 - SGPMGR Macro, GETREAL (SVC 55) Routine
Refer to Chart 13.15.

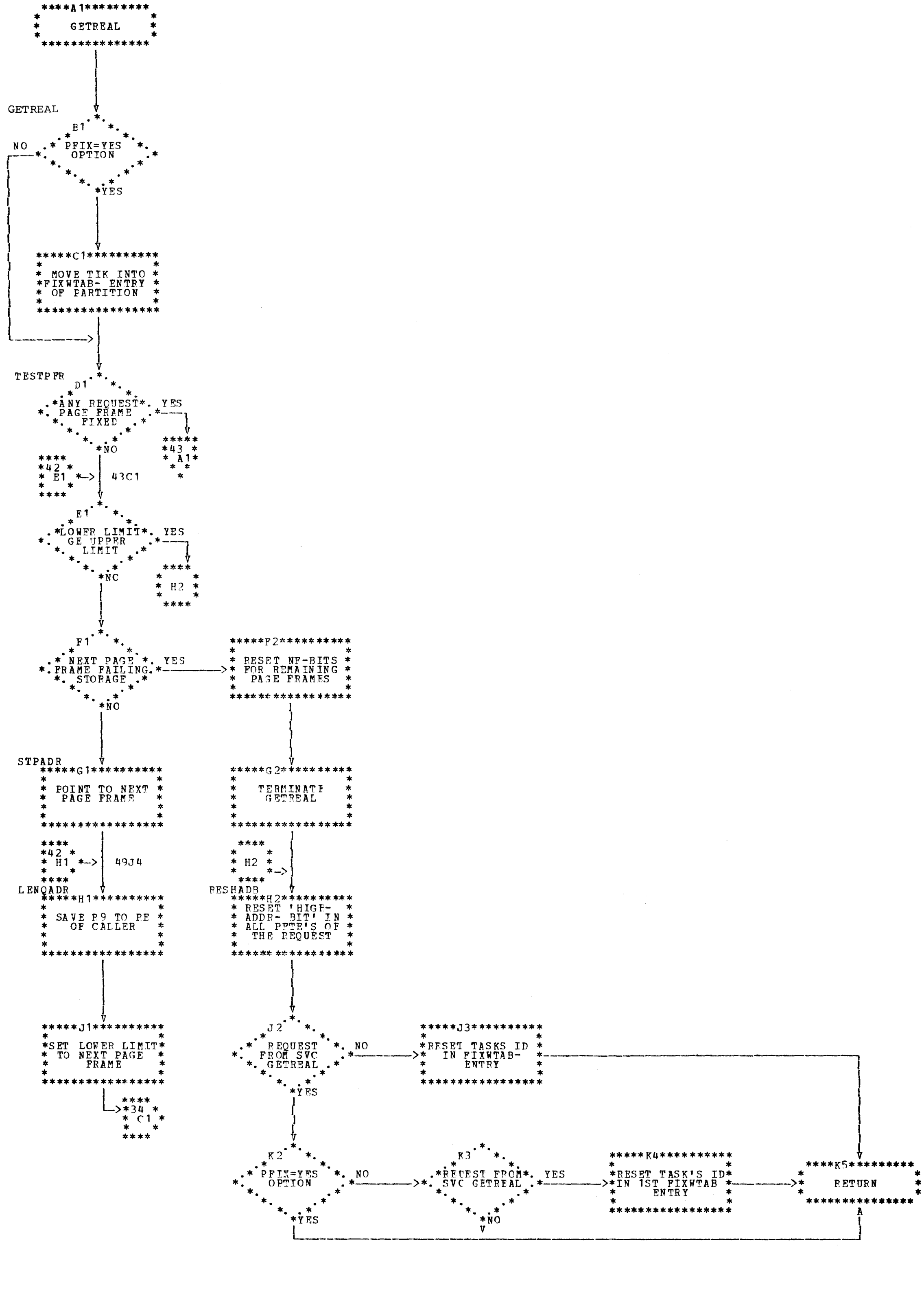


Chart B42. $$A$SUP1 - SGPMGR Macrc, GETREAL Rcutine
Refer to Chart 13.15.

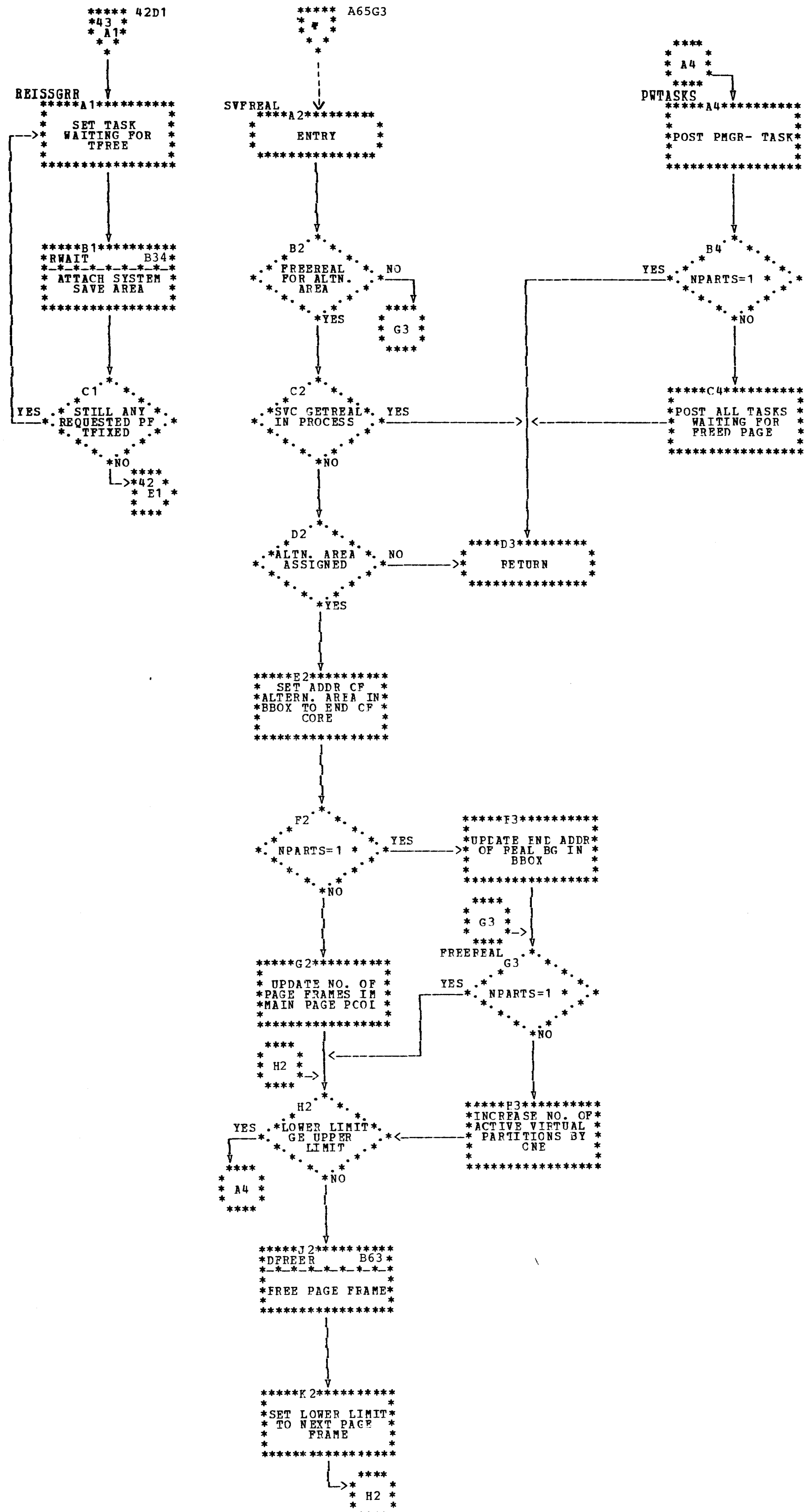


Chart B43. $$A$SUP1 - SGPMGR Macro, FREEREAL (SVC 54) Routine
Refer to Chart 13.17.

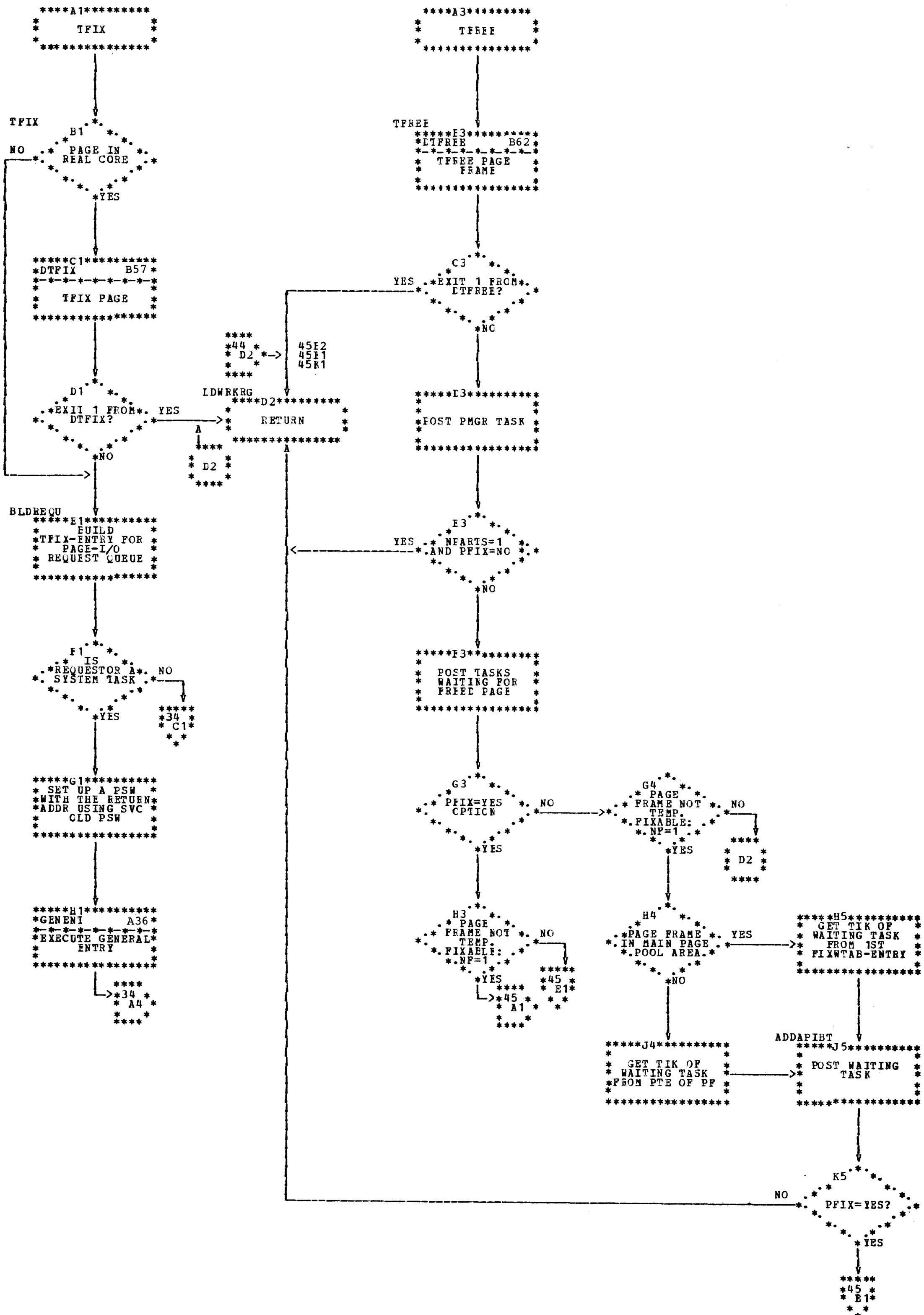


Chart B44. $$A$SUP1 - SGPMGR Macro, TFIX and TFREE Routines
Refer to Charts 13.9 and 13.10.

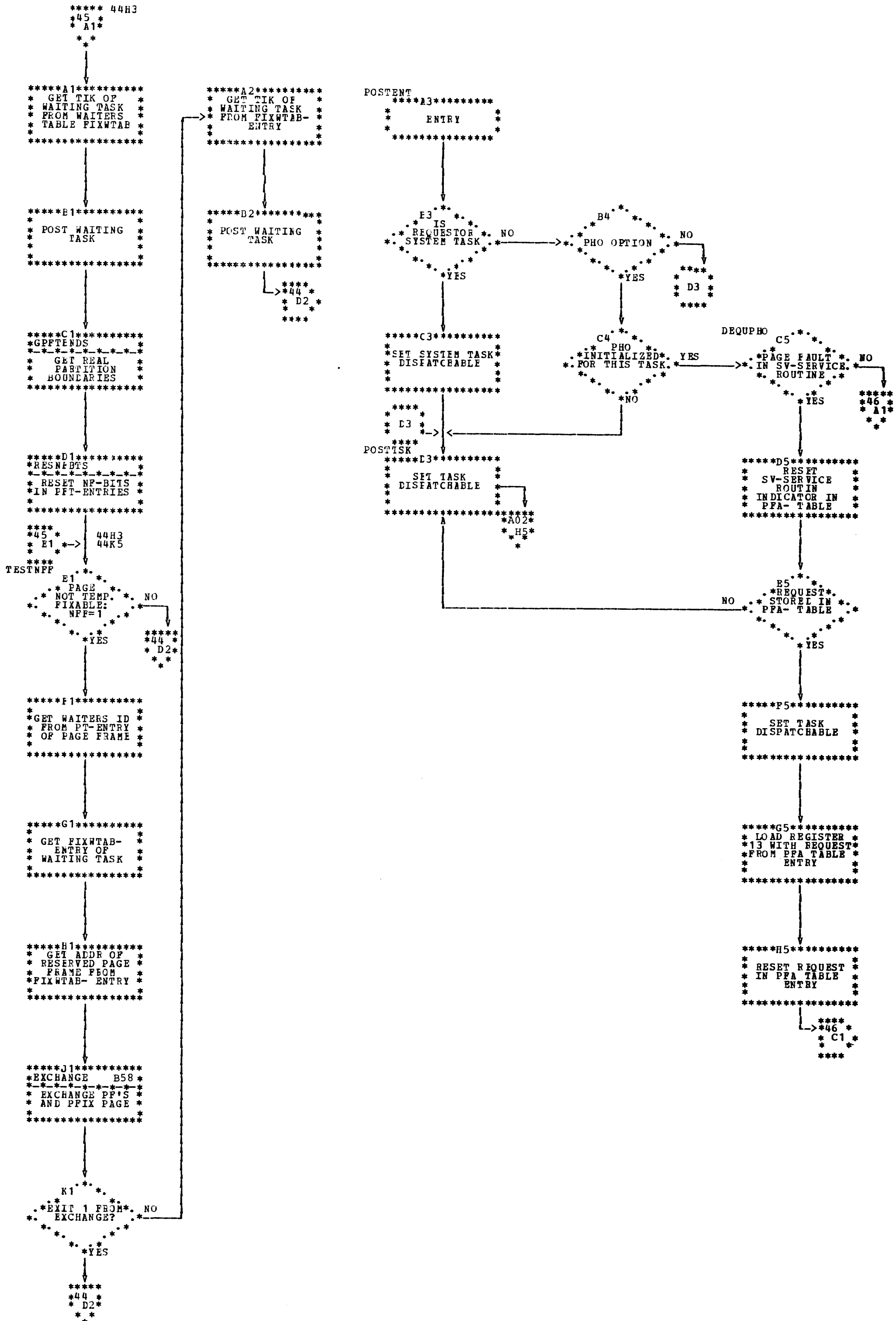


Chart B45. $$A$SUP1 - SGPMGR Macrc, Post Requesting Task Ready to Run
Refer to Charts 13.3 and 13.11.

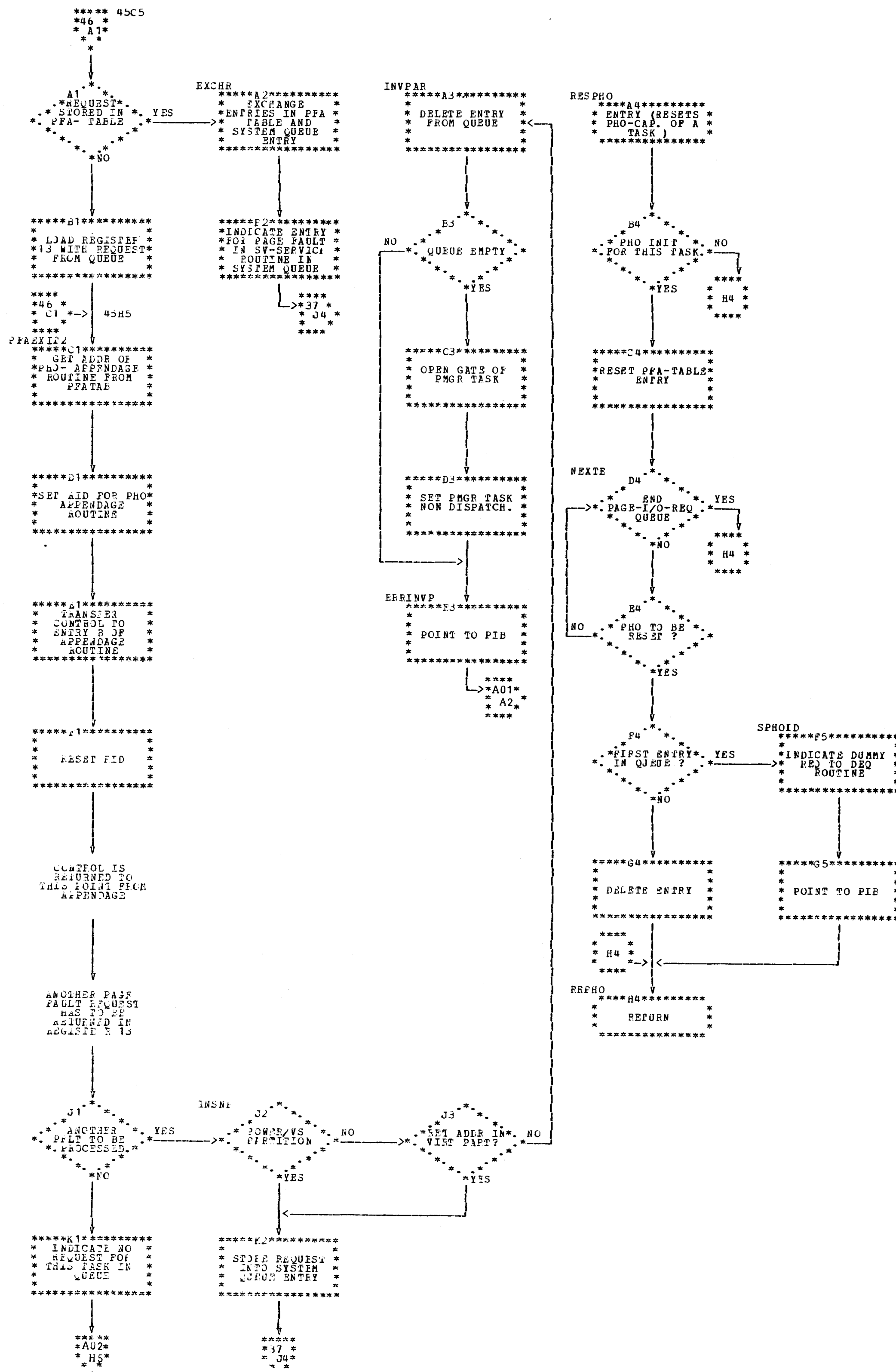


Chart B46. $$A$SUP1 - SGPMGR Macro, Dequeue Page Queue Entry
Refer to Chart 13.3.

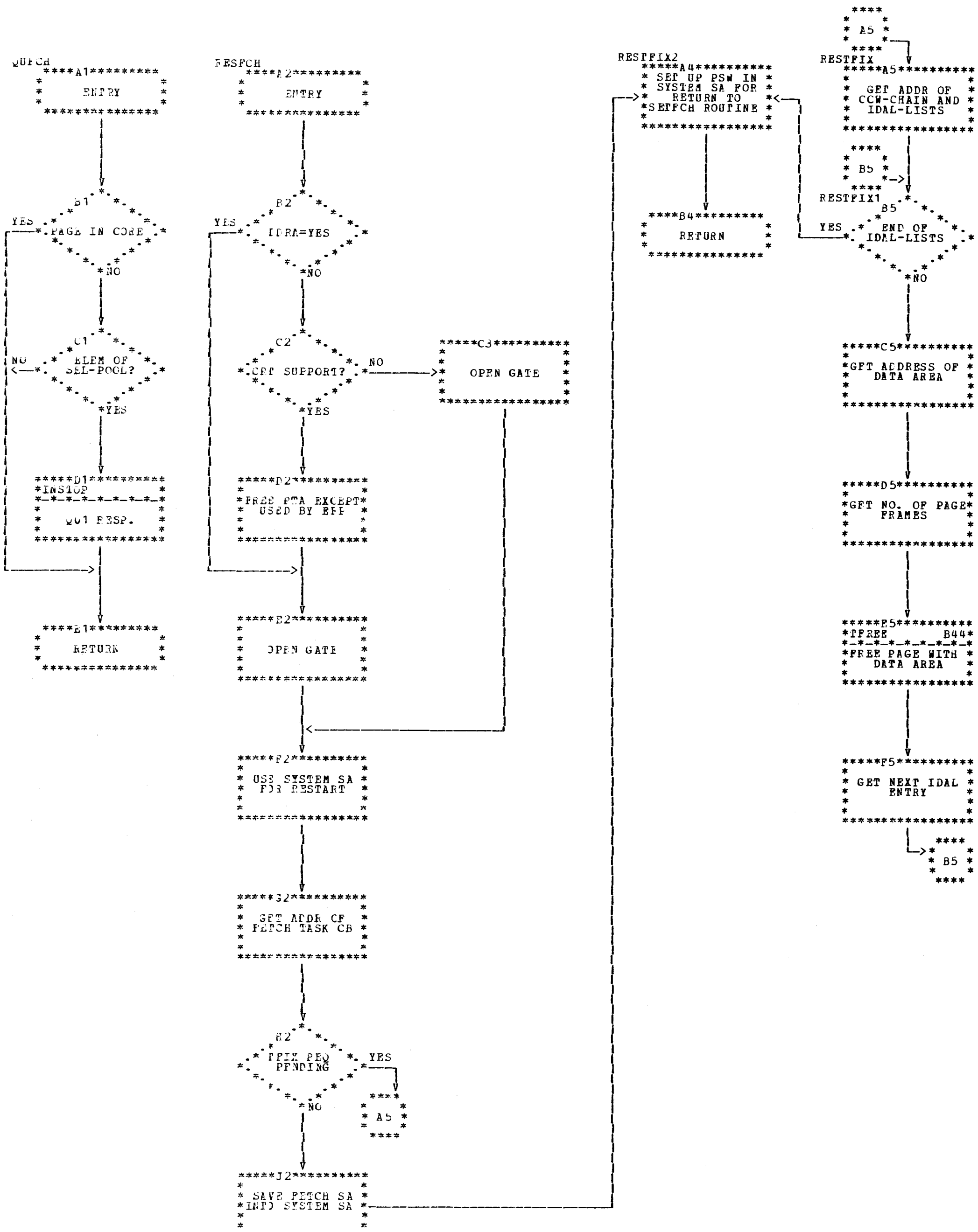


Chart B47. $$A$SUP1 - SGPMGR Macro, Reset Fetch
Refer to Chart 13.5.

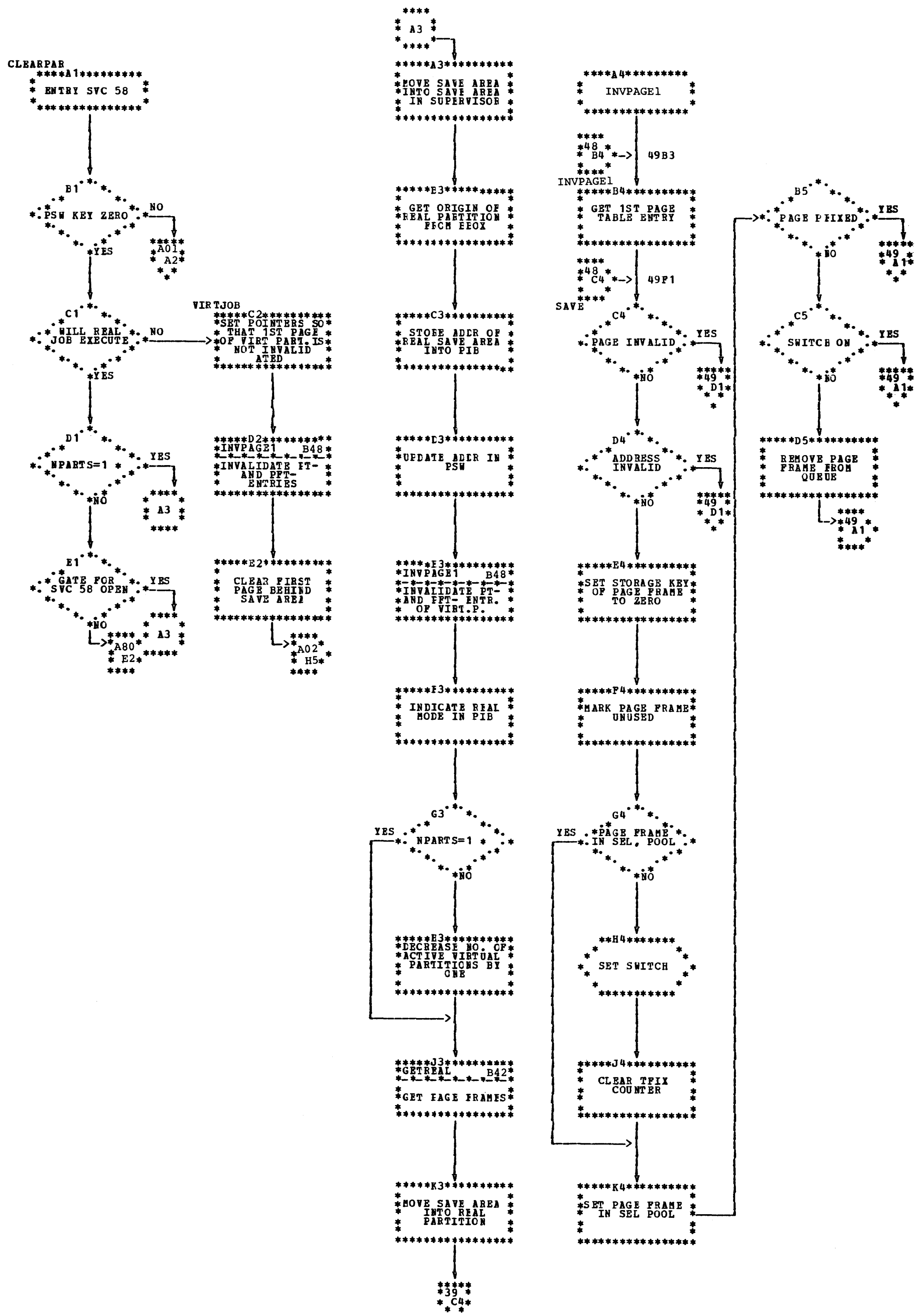


Chart B48. $$A$SUP1 - SGPMGR Macro, Initialize Partition (SVC 58)
Refer to Chart 13.16.

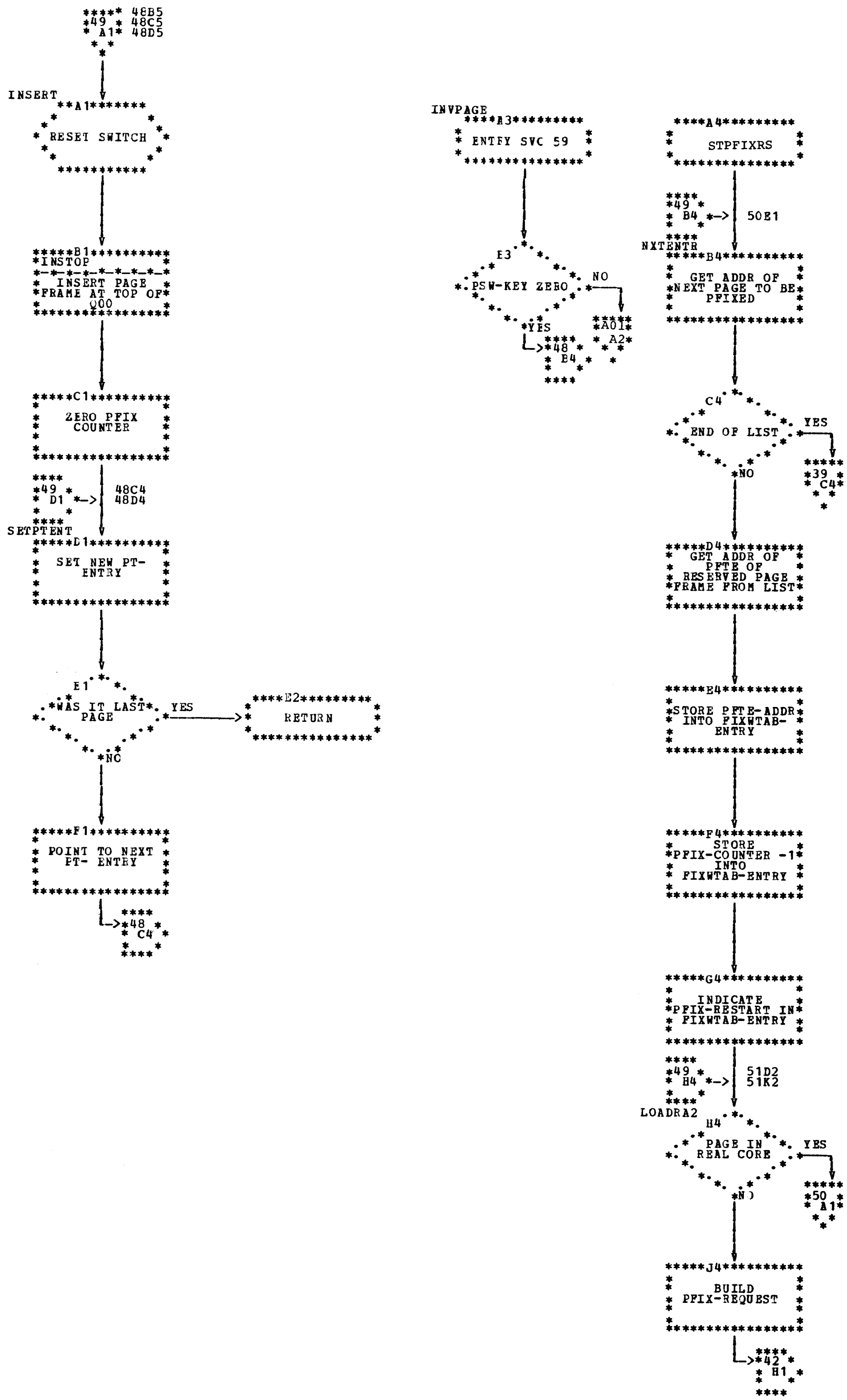


Chart B49. $$A$SUP1 - SGPMGR Macro, PFIX Restart
Refer to Chart 13.13.

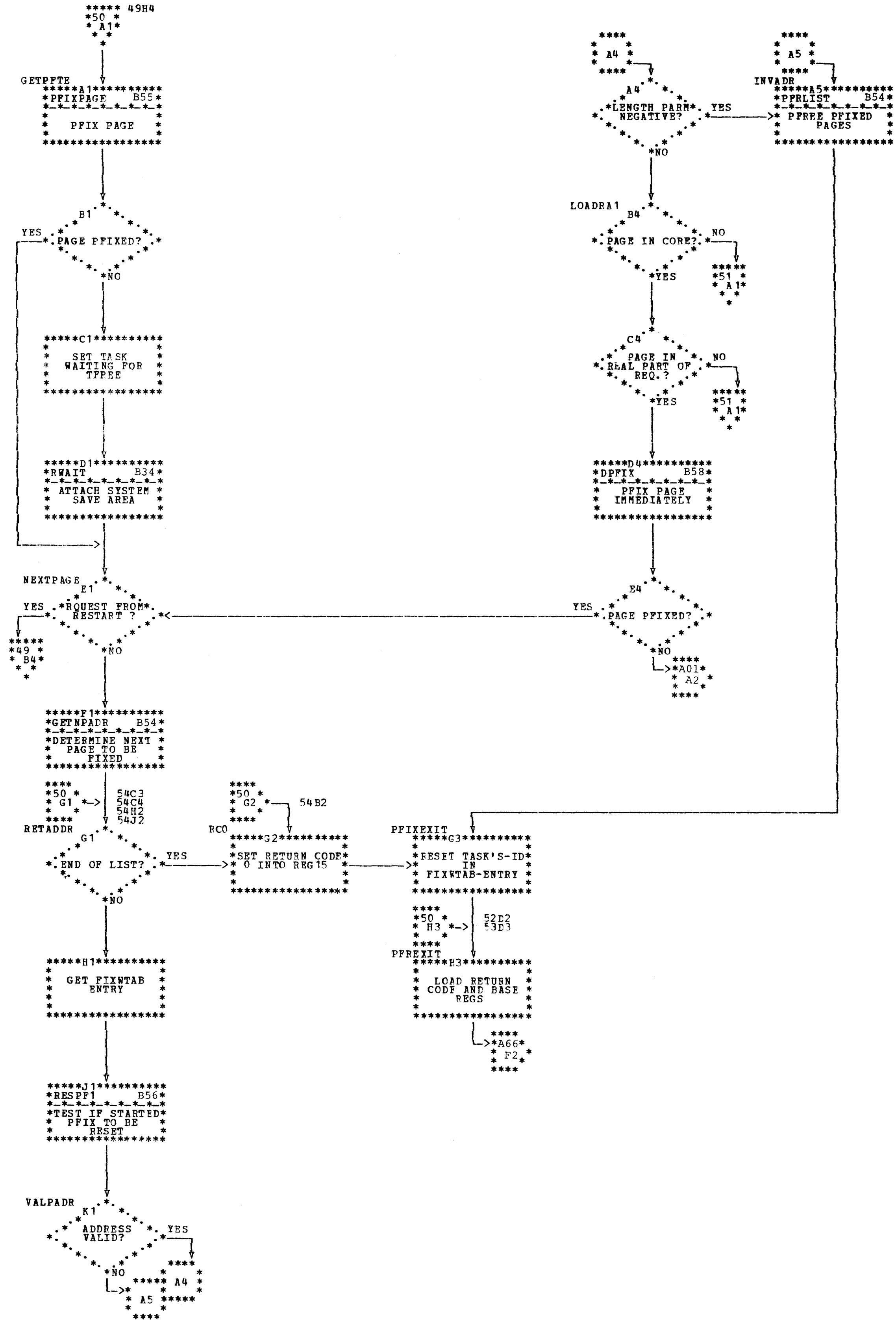


Chart B50. $$A$SUP1 - SGPMGR Macro, PFIX Routine
Refer to Chart 13.11.

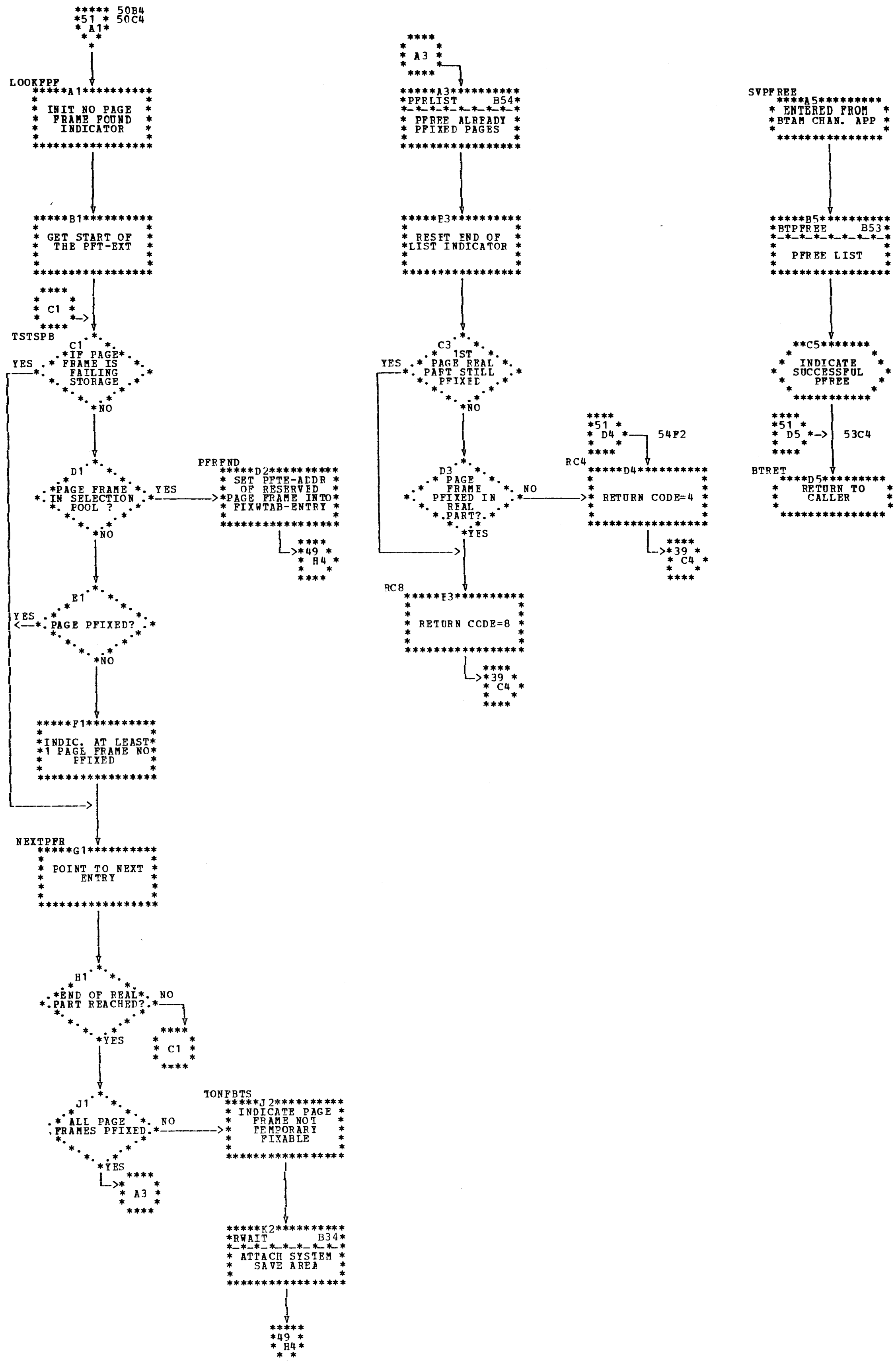


Chart B51. $$A$SUP1 - SGPMGR Macro, PFIX, Find Page Frame
Refer to Chart 13.12.

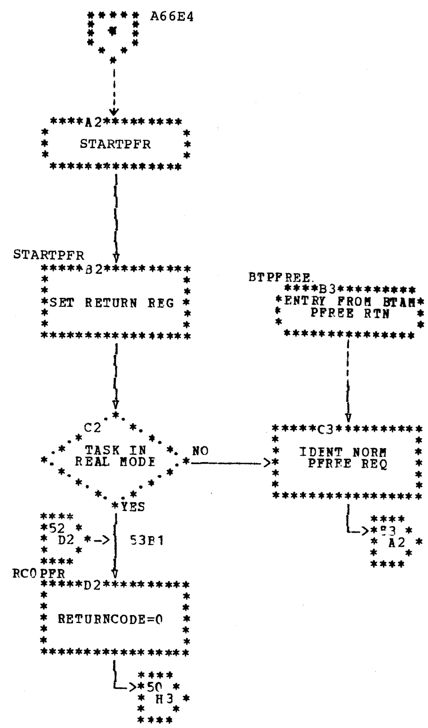


Chart B52. $$A$SUP1 - SGPMGR Macro, PFREE Routine (SVC 68)
Refer to Chart 13.14.

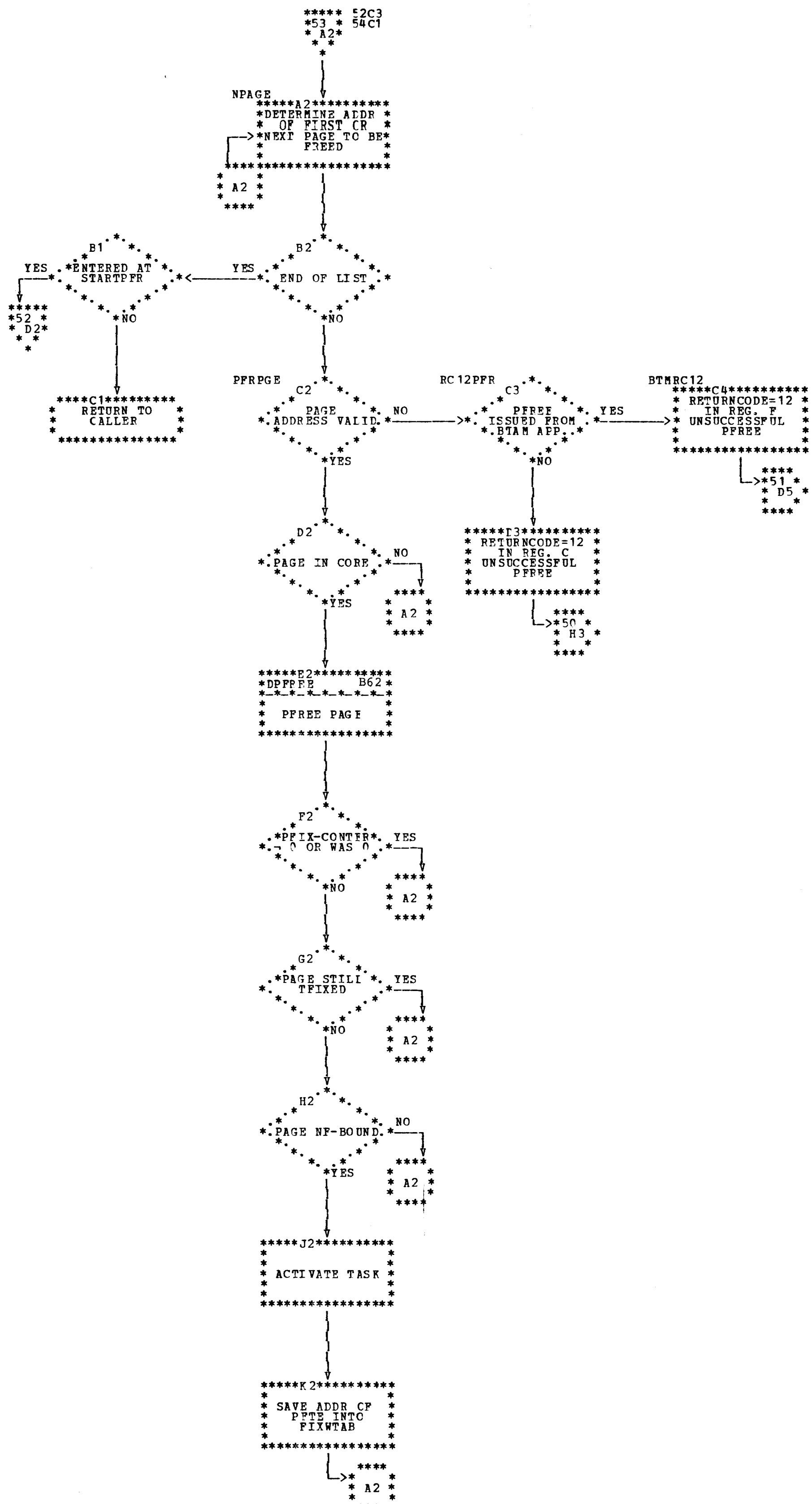


Chart B53. $$A$SUP1 - SGPMGR Macro, PFREE Routine (SVC 68)
Refer to Chart 13.14.

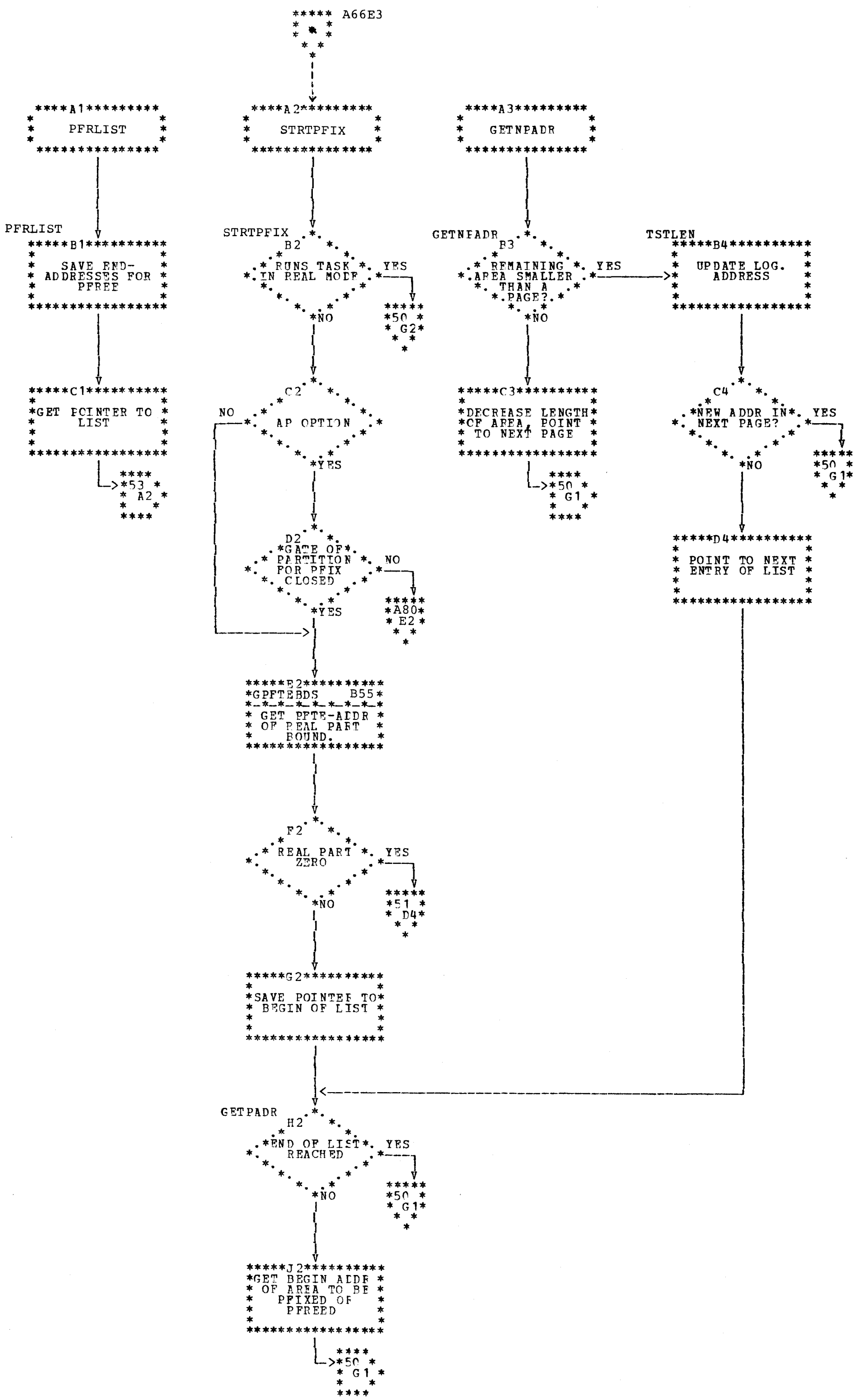


Chart B54. $$A$SUP1 - SGPMGR Macro, PFIX Routine (SVC 67)
Refer to Chart 13.11.

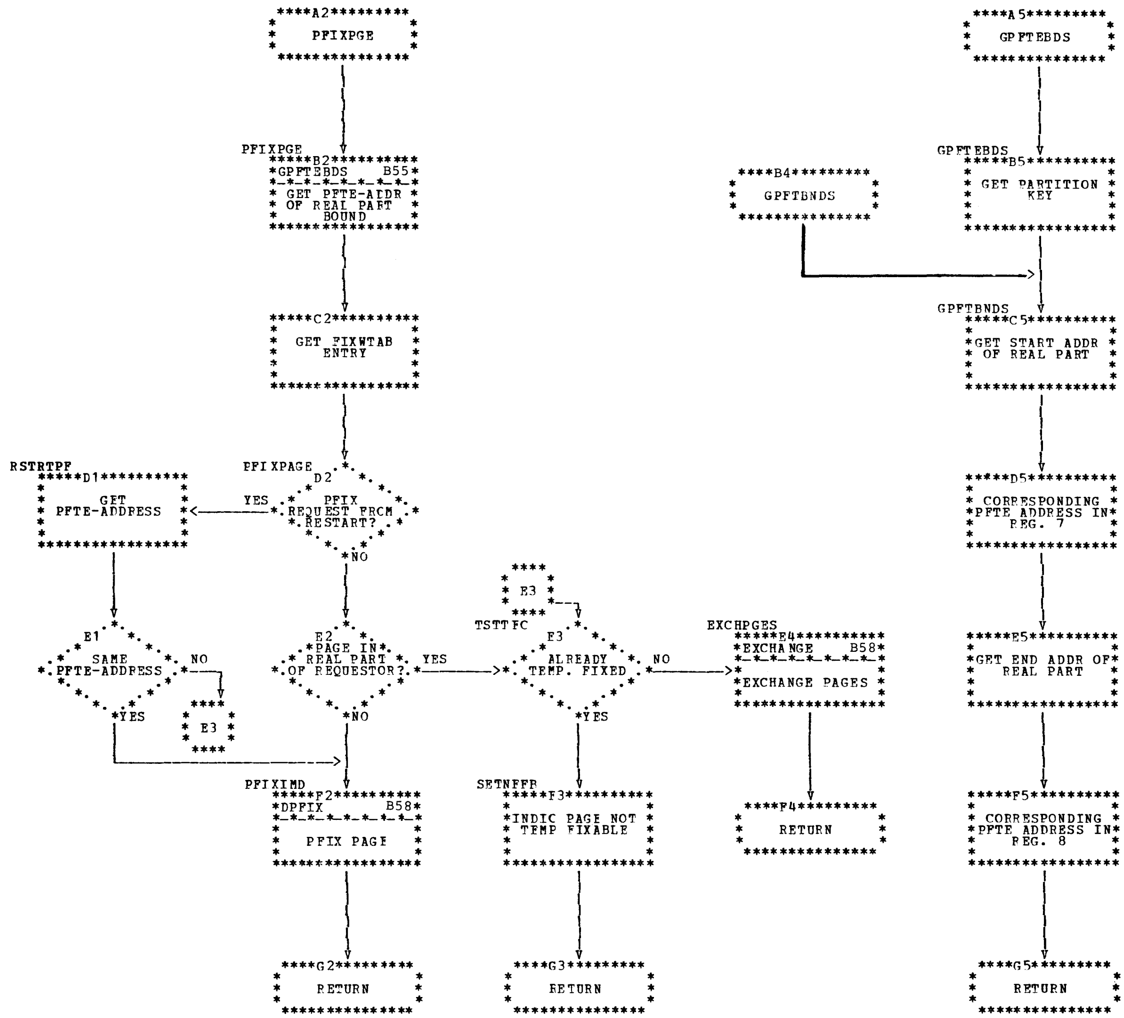


Chart B55. $$A$SUP1 - SGPMGR Macro, PFIX Routine (SVC 67)
Refer to Chart 13.11.

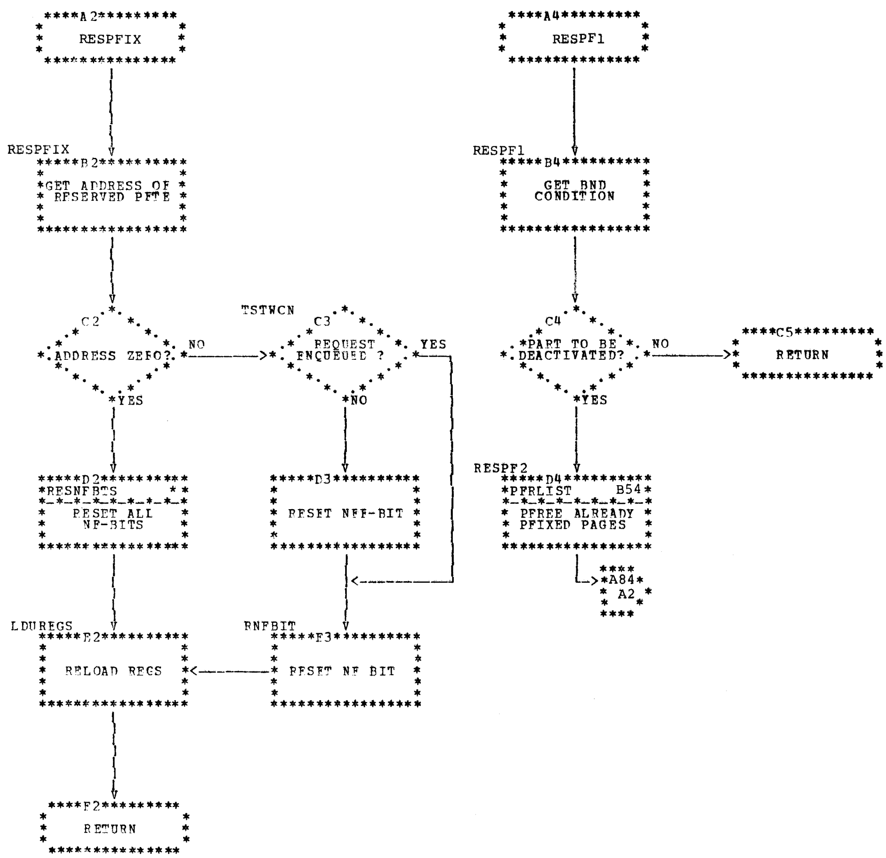


Chart B56. $$A$SUP1 - SGPMGR Macrc, PFIX Rcutine (SVC 67)
Refer to Chart 13.11.

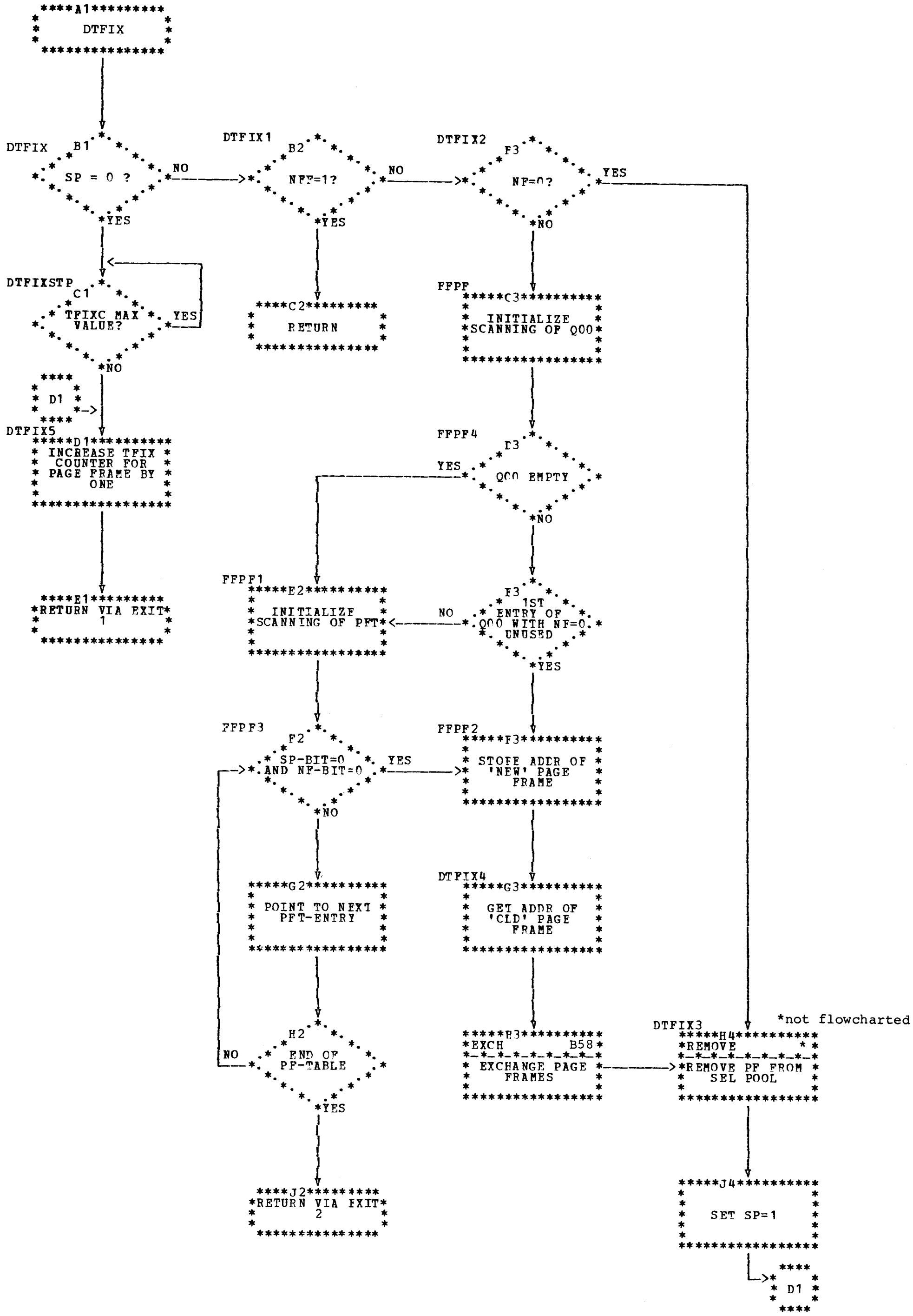


Chart B57. $$A$SUP1 - SGPMGR Macro, DTFIX Subroutine
Refer to Chart 13.9.

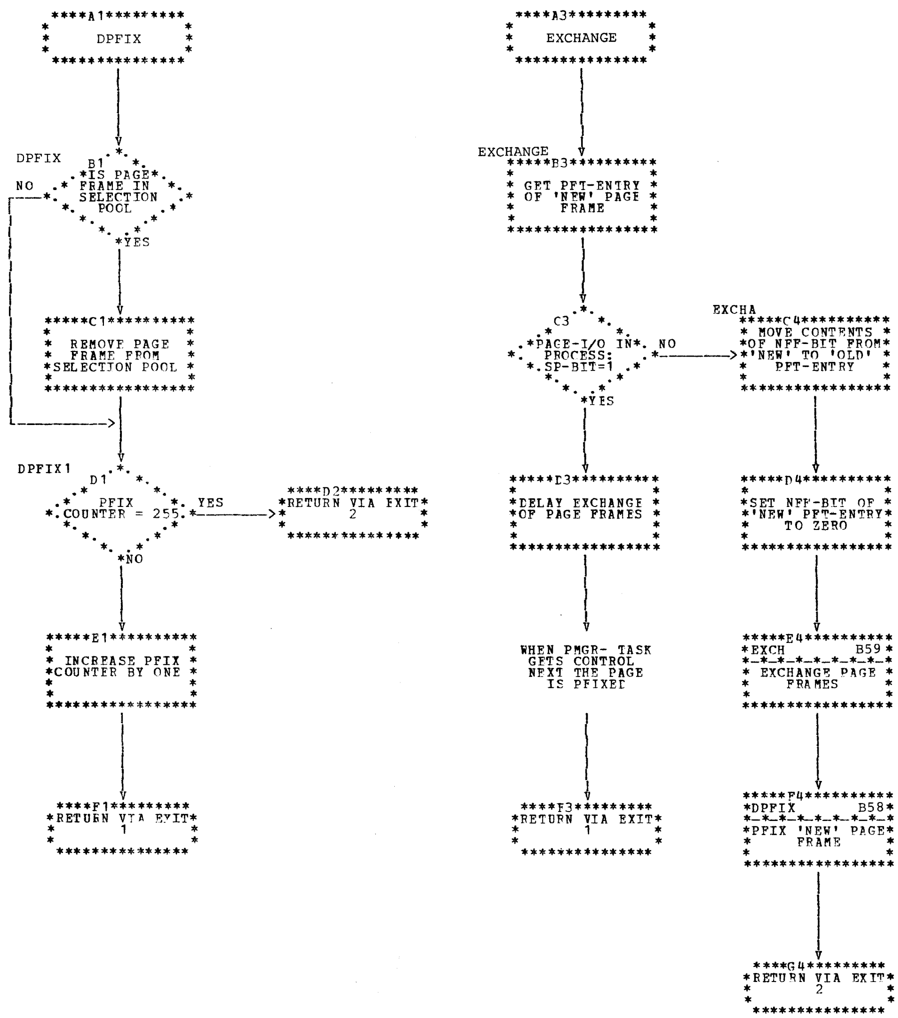


Chart B58. $$A$SUP1 - SGPMGR Macrc, DPFIX Subrcutine, Exchange Pages
Refer to Charts 13.11 and 13.13.

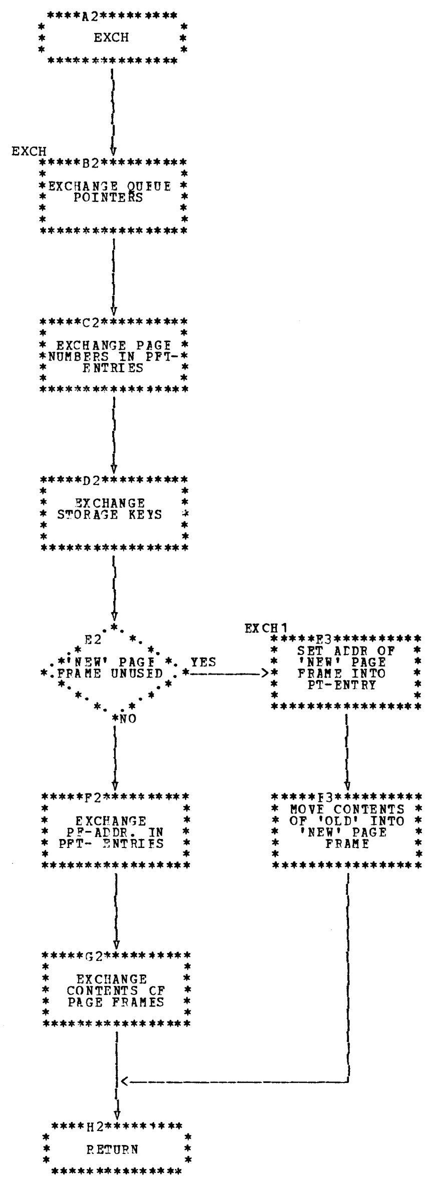


Chart B59. $$A$SUP1 - SGPMGR Macro, Exchange Pages
Refer to Chart 13.13.

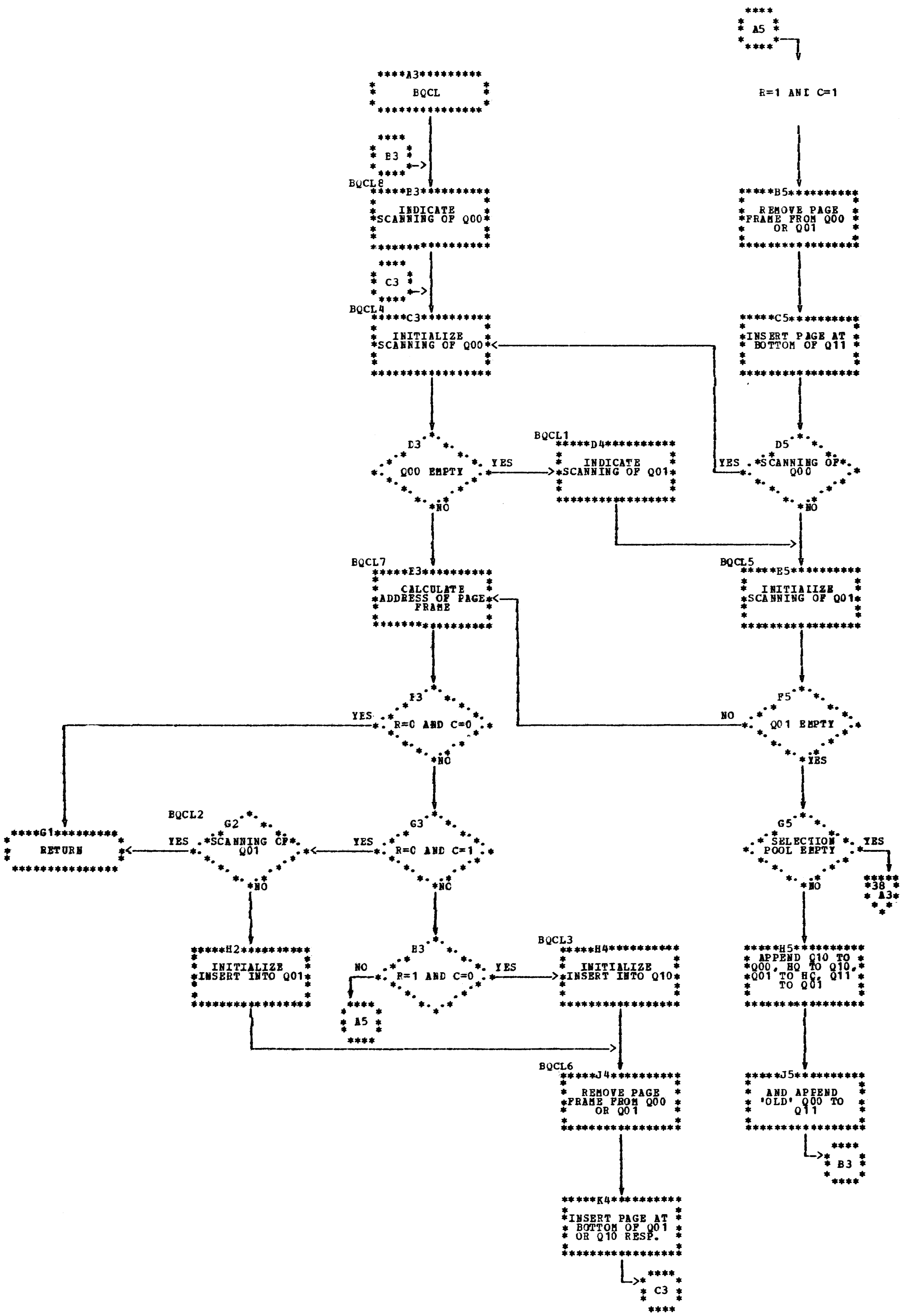


Chart B60. $$A$SUP1 - SGPMGR Macro, Scan Queues for Page Frame
Refer to Chart 13.5.

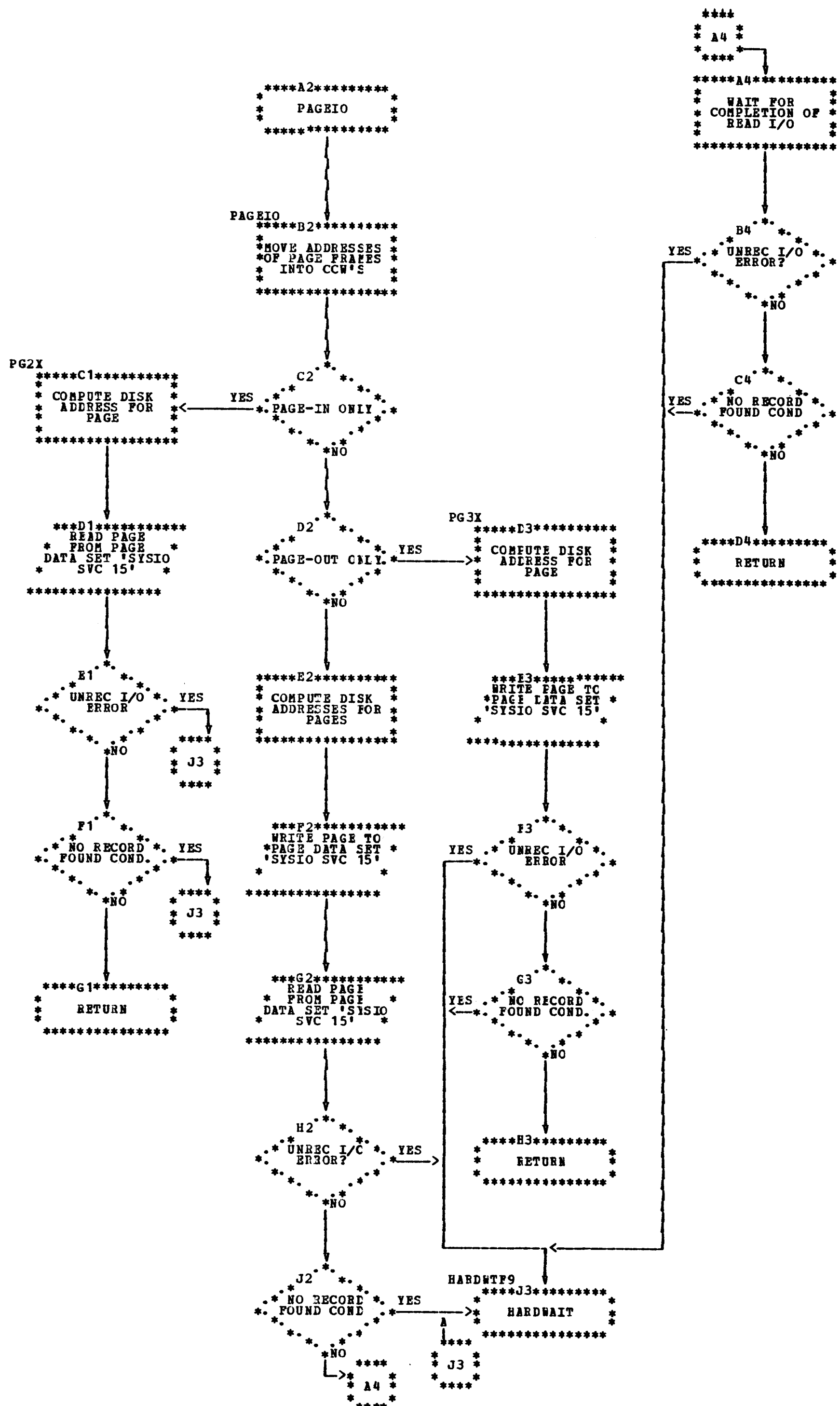


Chart B61. $$A$SUP1 - SGPMGR Macro, Perform Page I/O
Refer to Chart 13.5.

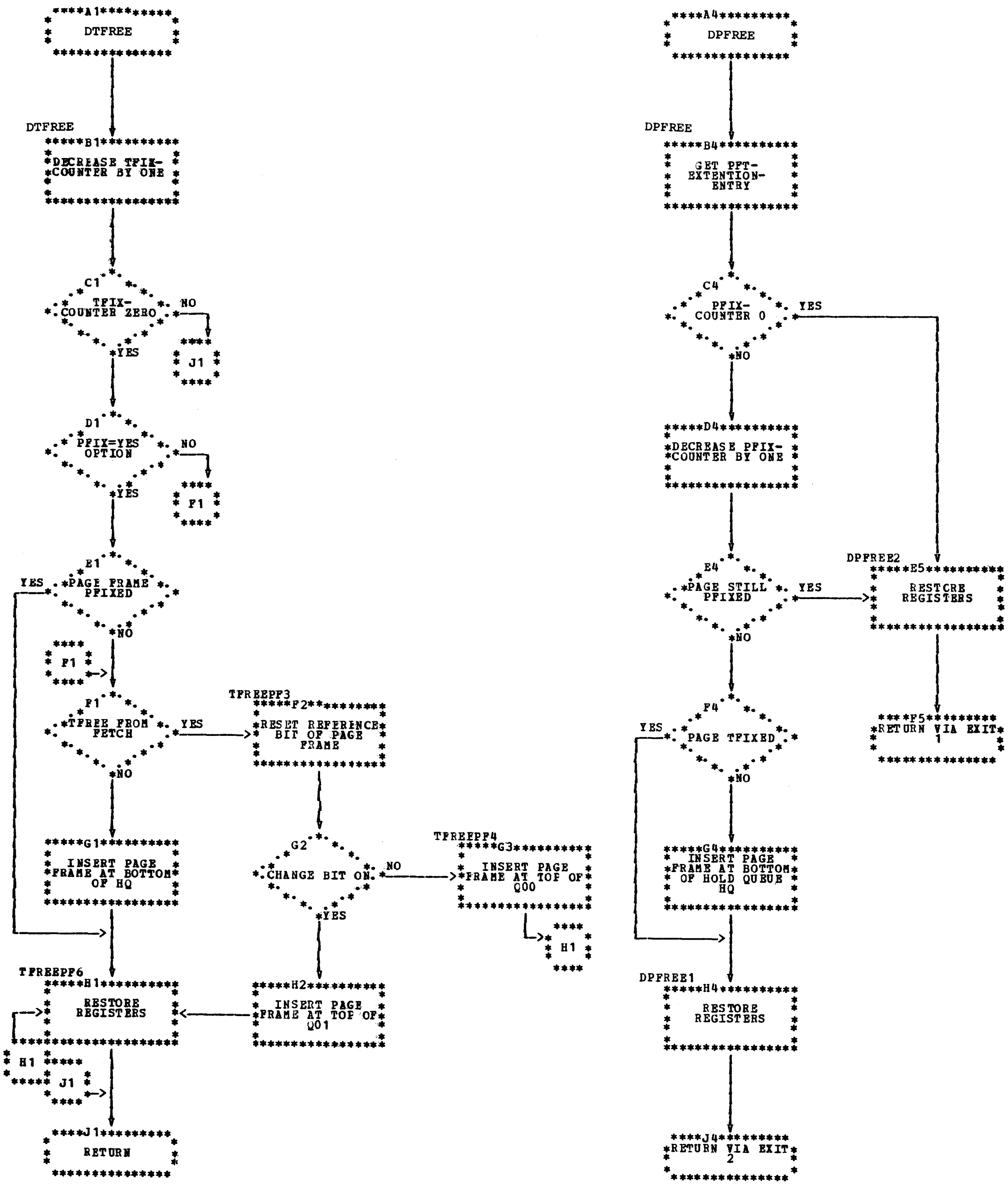


Chart B62. $$A$SUP1 - SGPMGR Macro, DTFREE and DPFREE Subroutines
Refer to Chart 13.10.

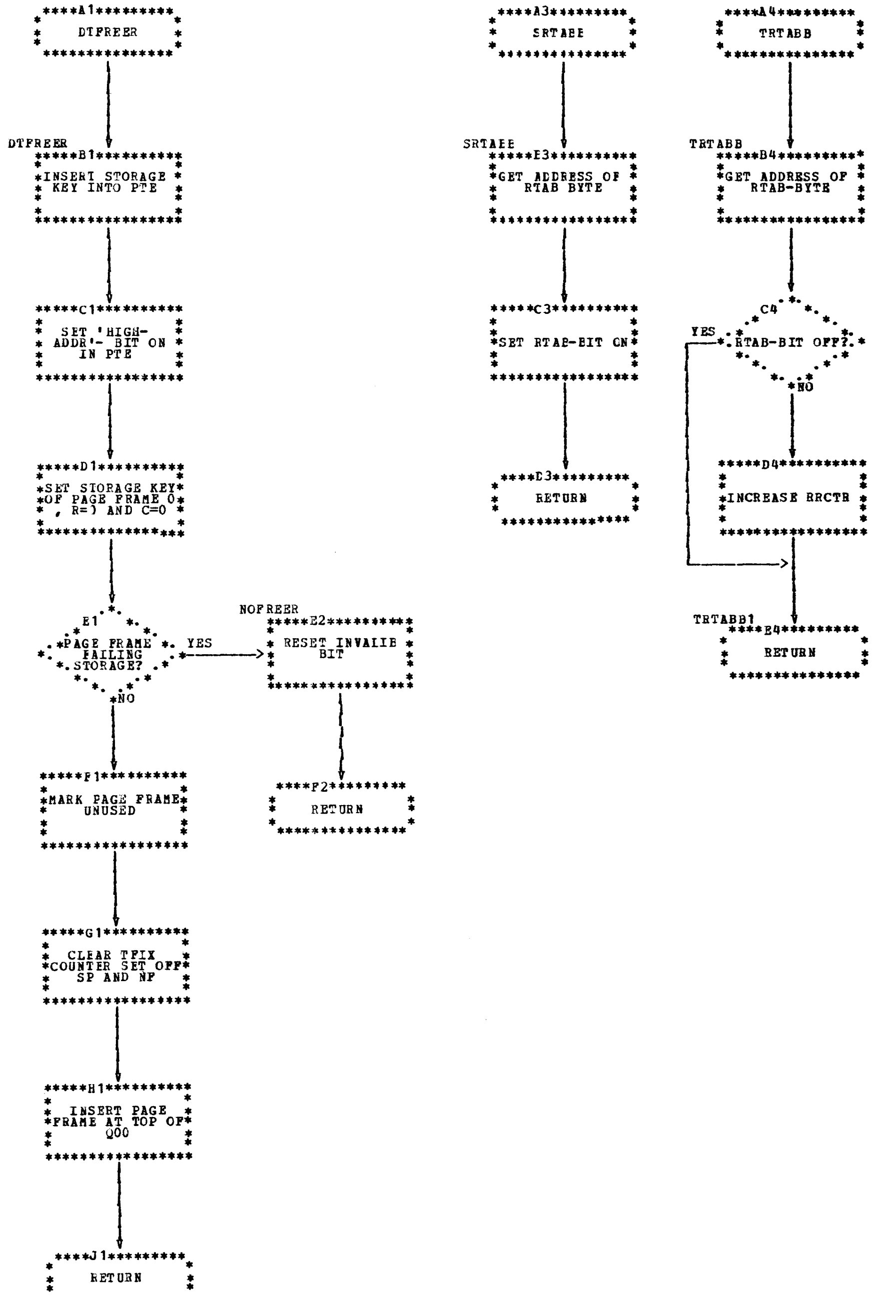


Chart B63. $$A$SUP1 - SGPMGR Macro, Subroutines
Refer to Charts 13.14 and 13.17.

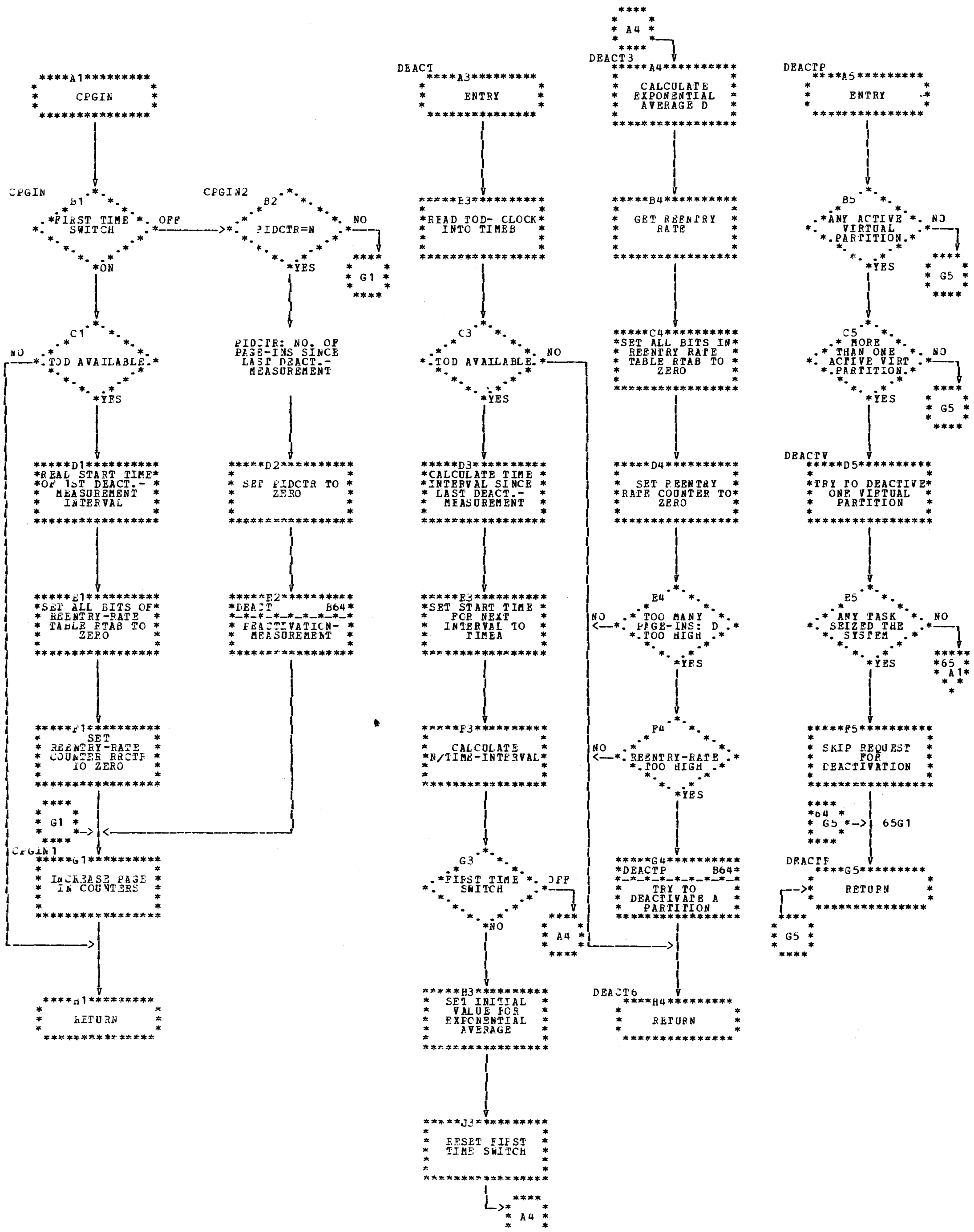


Chart B64. $$A$SUP1 - SGPMGR Macro, Update Page-In Counters
Refer to Chart 13.6.

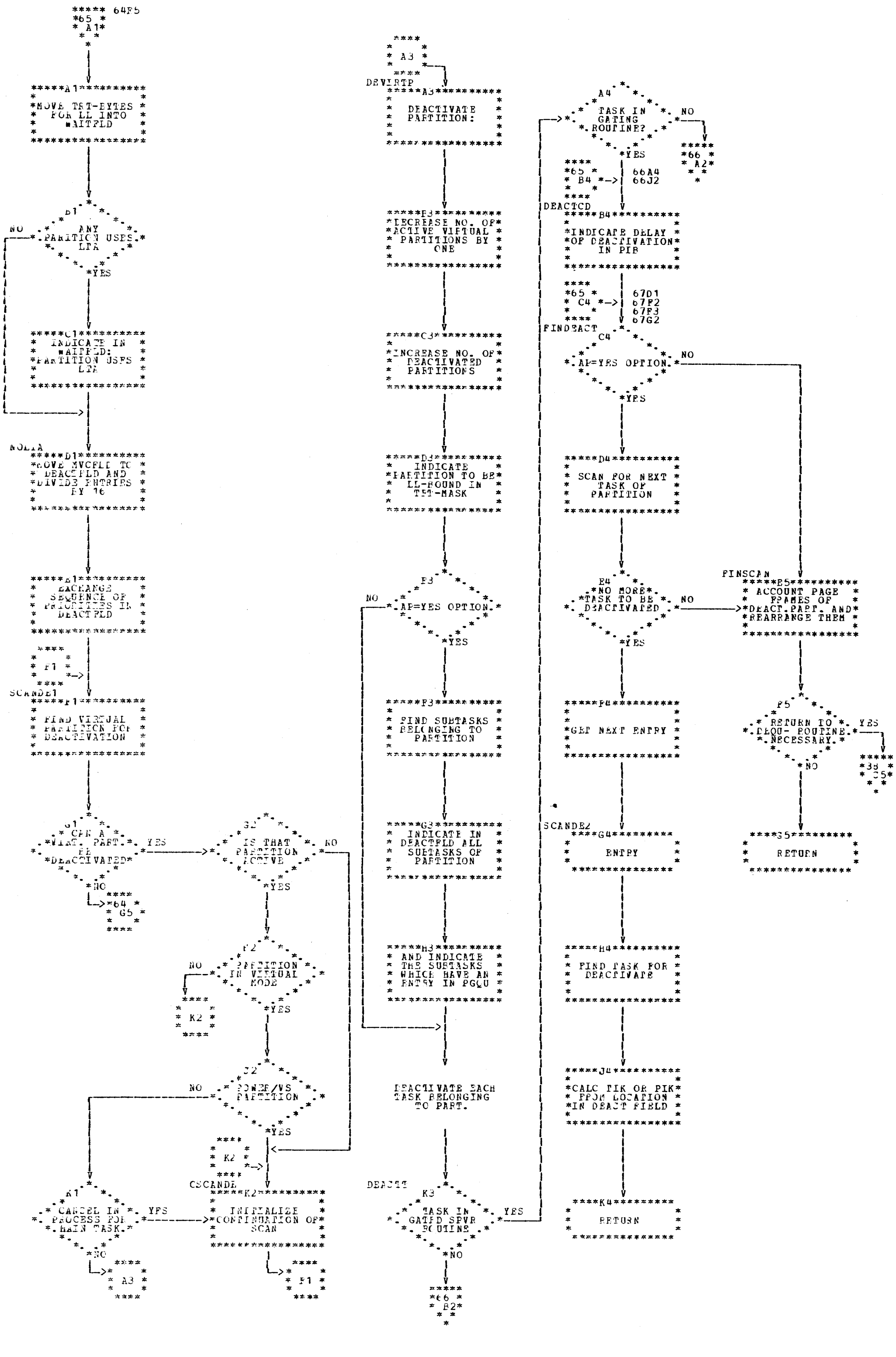


Chart B65. $$A$SUP1 - SGPMGR Macro, Deactivate a Partition
Refer to Chart 13.6.

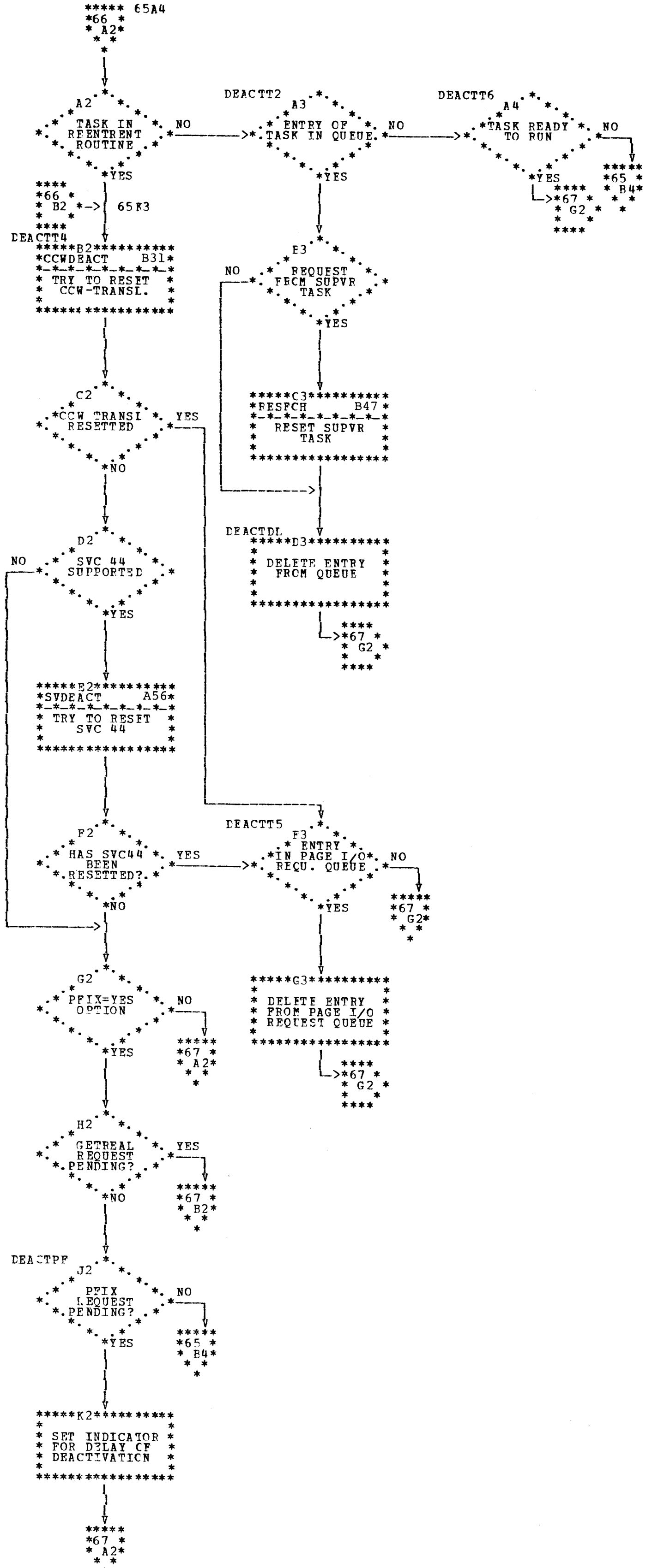


Chart B66. $$A$SUP1 - SGPMGR Macro, Deactivate a Partition
Refer to Chart 13.6.

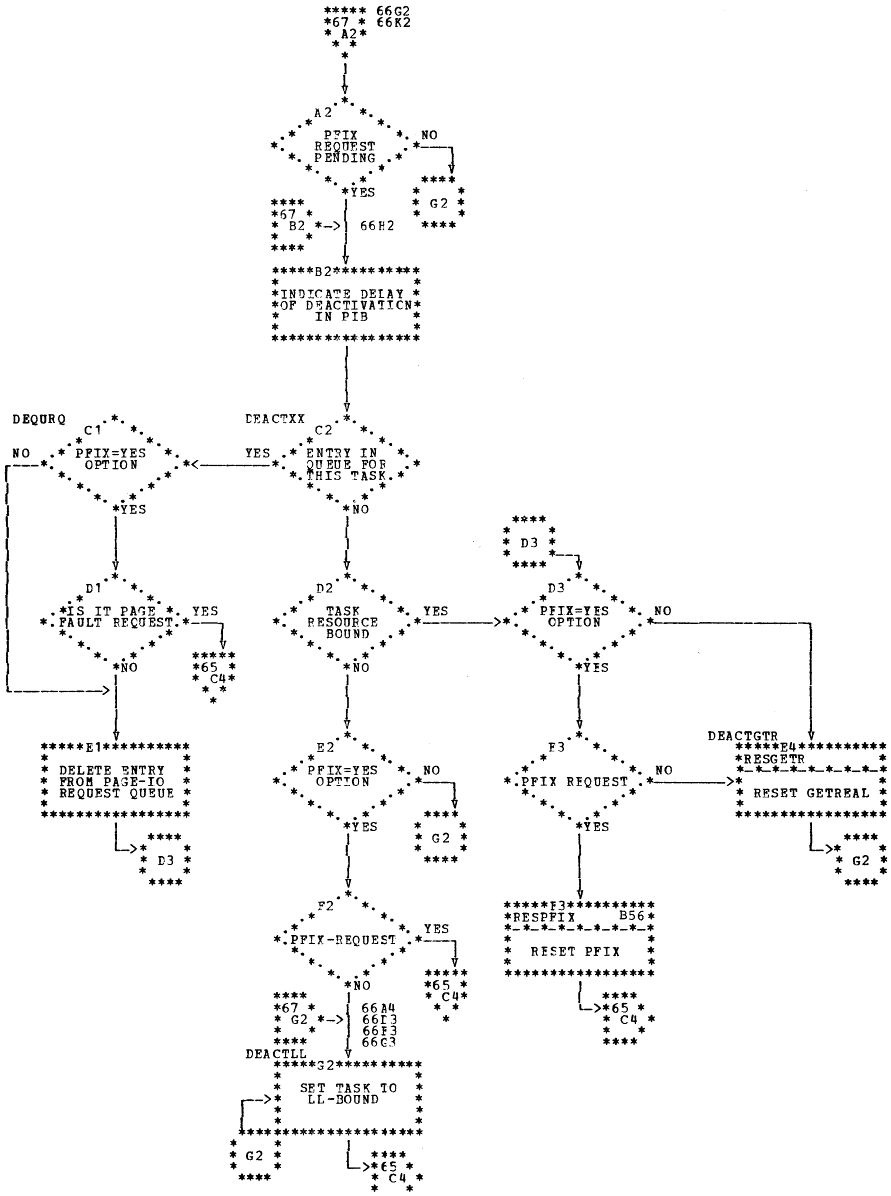


Chart B67. $$A$SUP1 - SGPMGR Macrc, Deactivate a Partition
Refer to Chart 13.6.

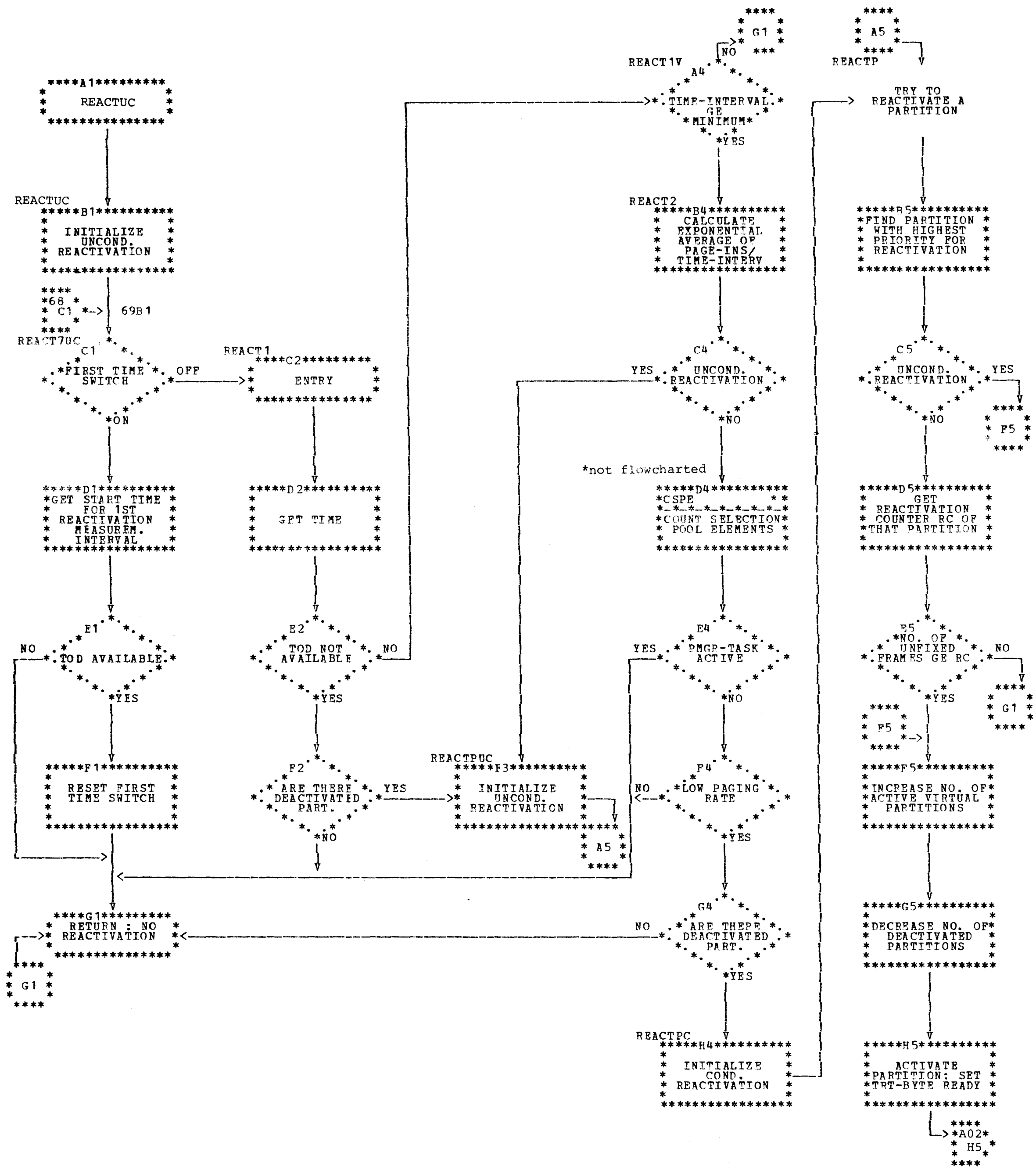


Chart B68. $$A$SUP1 - SGPMGR Macrc, Reactivate a Deactivated Partition
Refer to Chart 13.7.

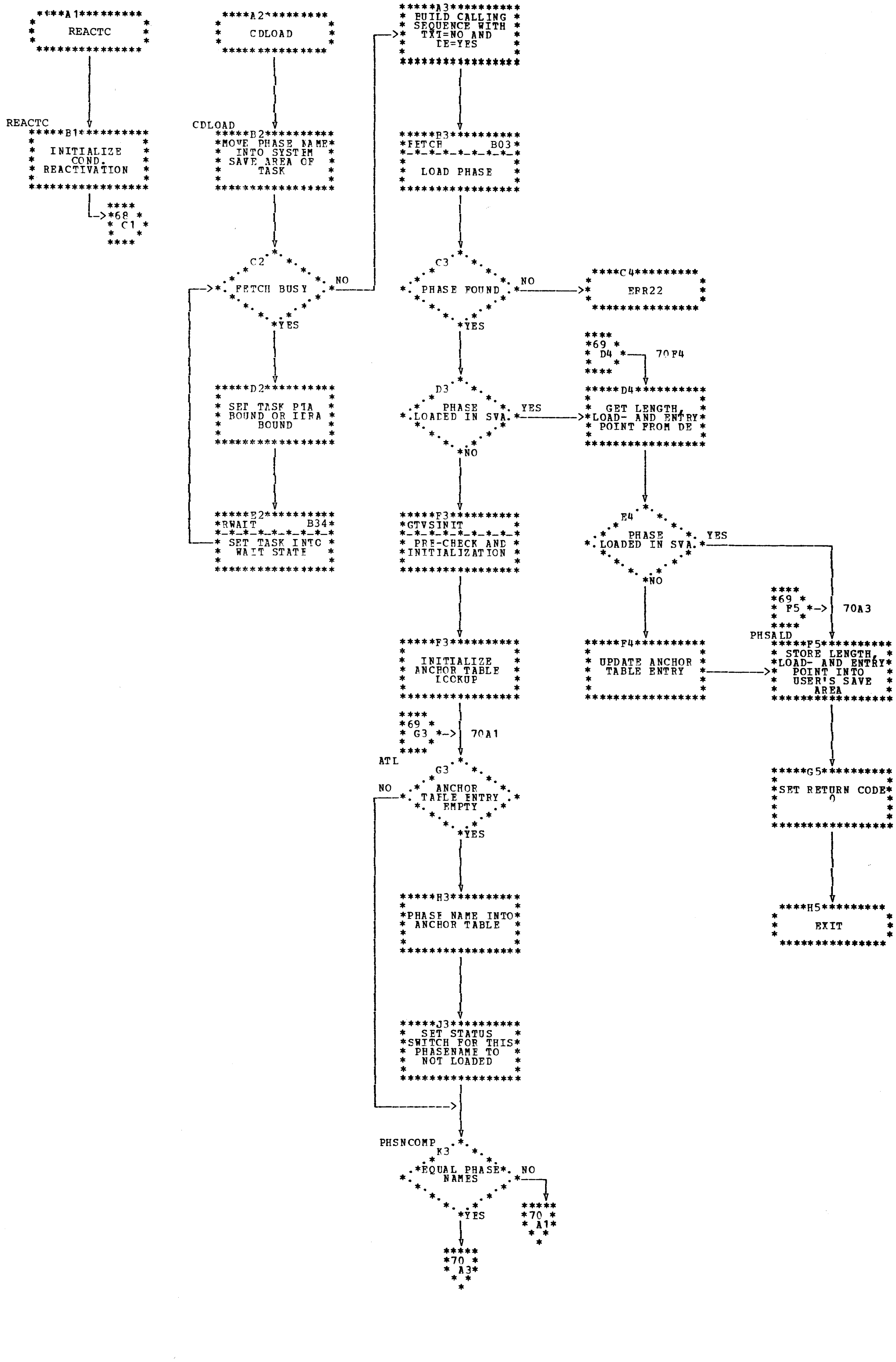


Chart B69. $$A$SUP1 - SGAM Macro, CDLOAD Routine (SVC 65)

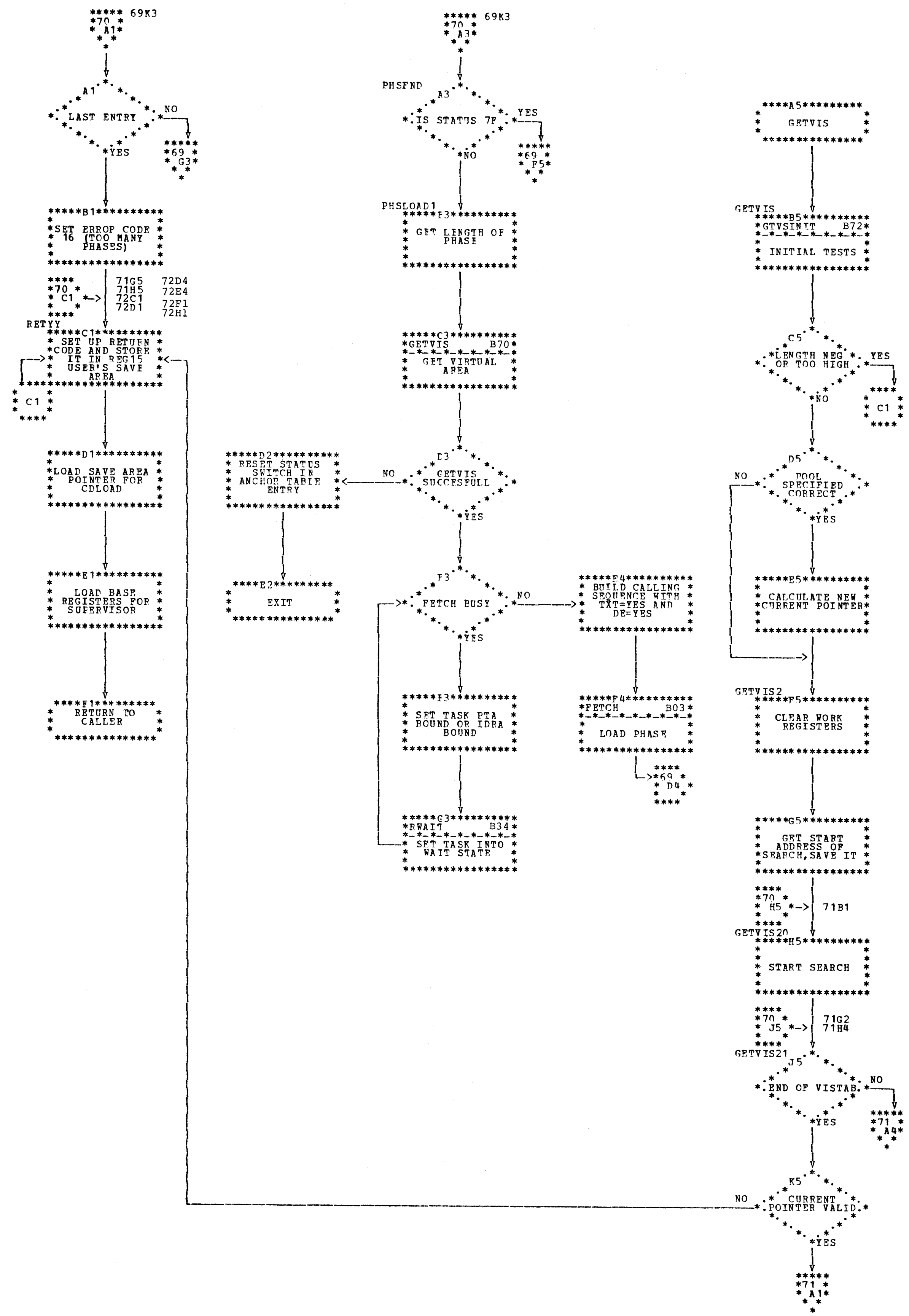


Chart B70. $$A$SUP1 - SGAM Macro, CDLOAD Routine (SVC 65), GETVIS Routine

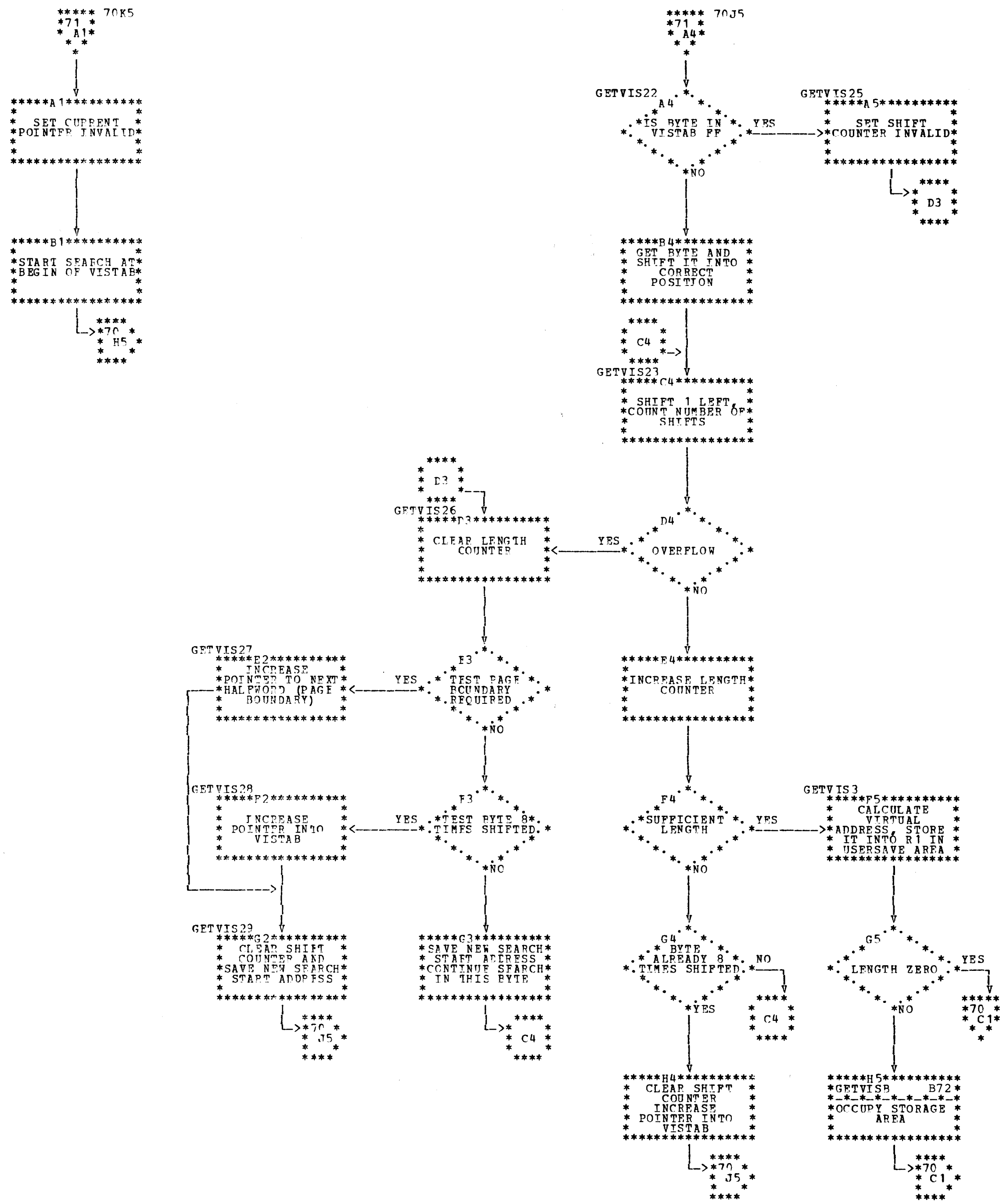


Chart B71. $$A$SUP1 - SGAM Macro, GETVIS Routine

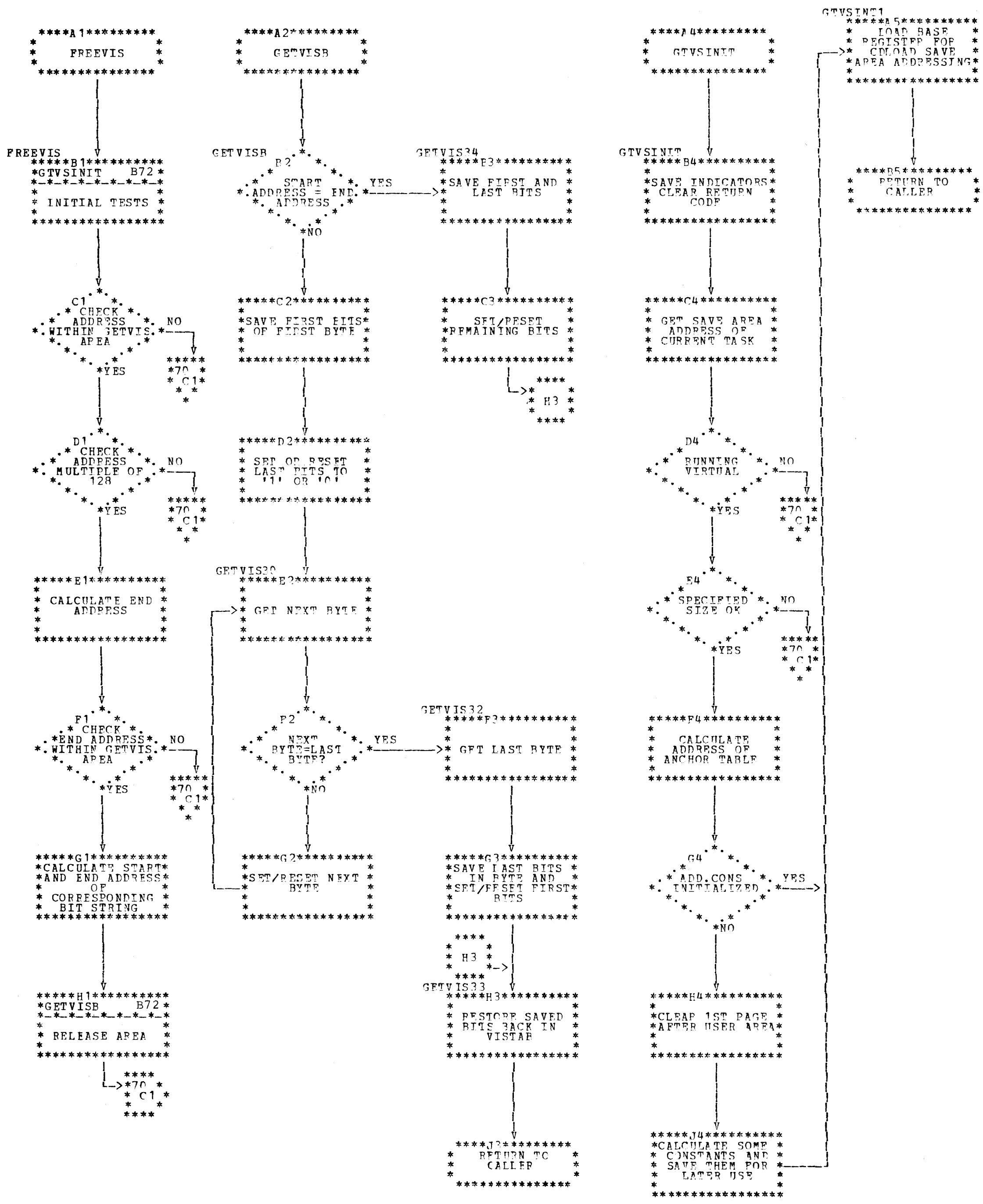


Chart B72. $$A$SUP1 - SGAM Macro, FREEVIS, GETVISB and GTVSINIT Routines

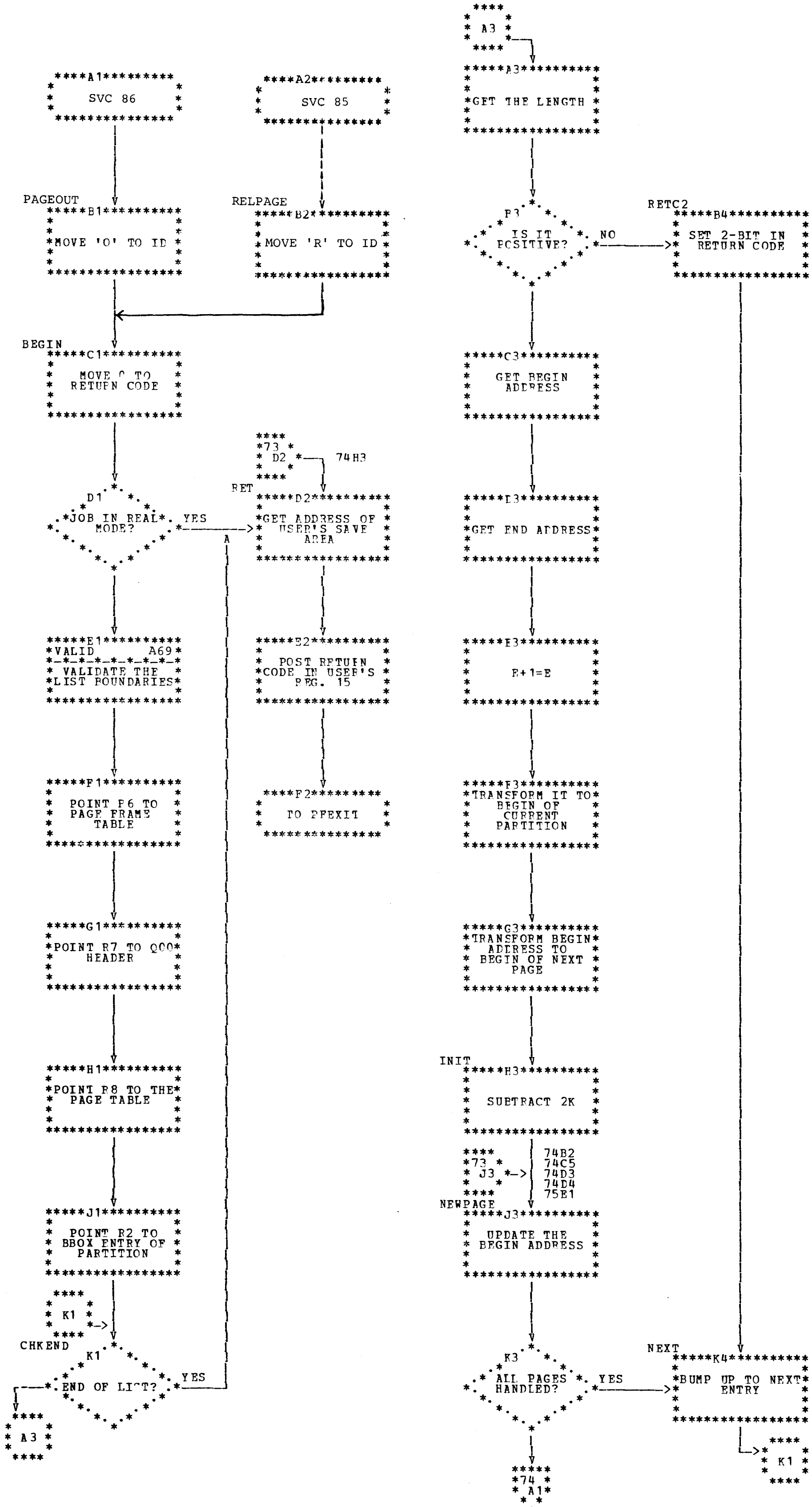


Chart B73. $$A$SUP1 - SGPCPT Macro, PAGEOUT (SVC 85/86) Routine

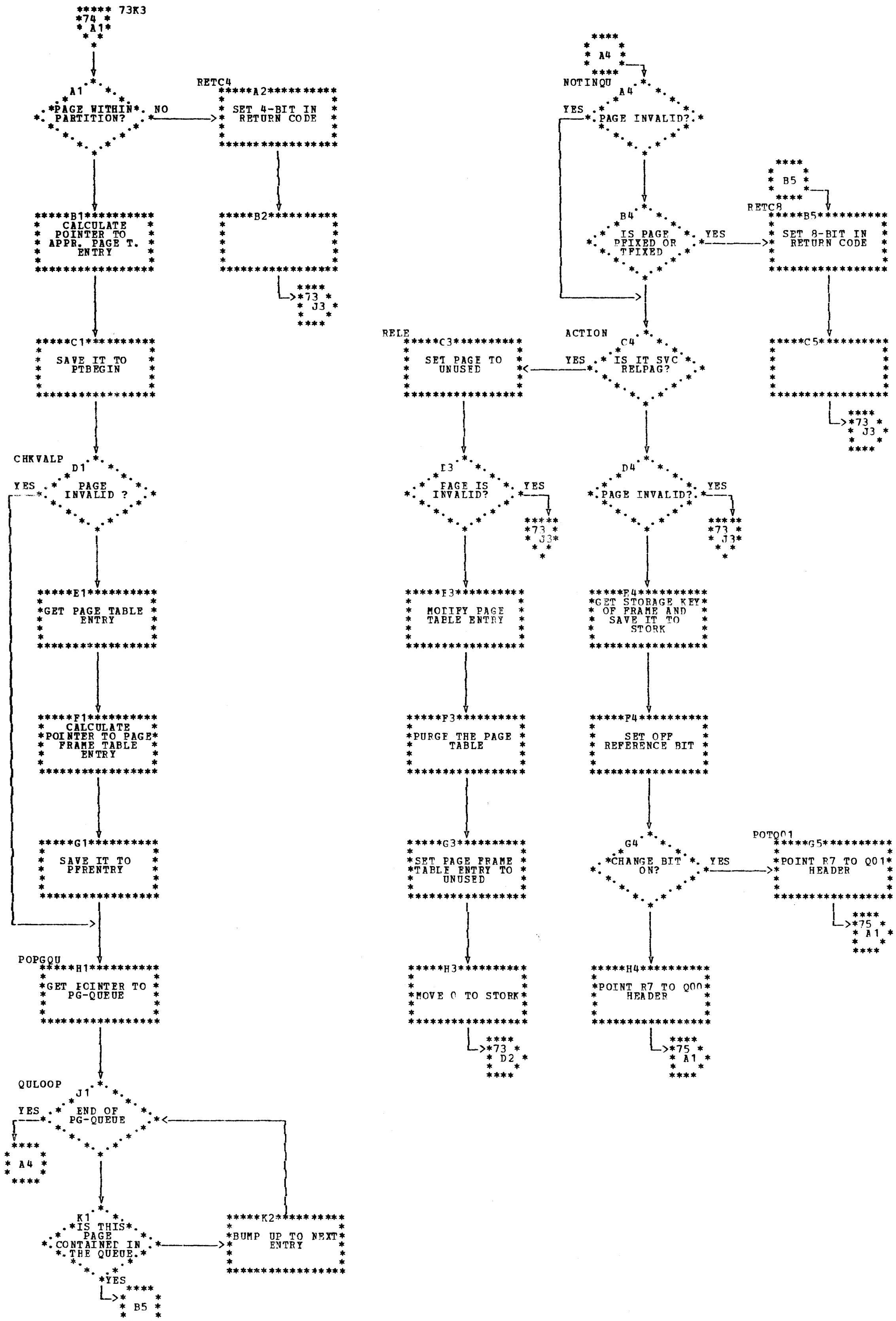


Chart B74. $$A$SUP1 - SGPOPT Macro, PAGEOUT (SVC 85/86) Routine

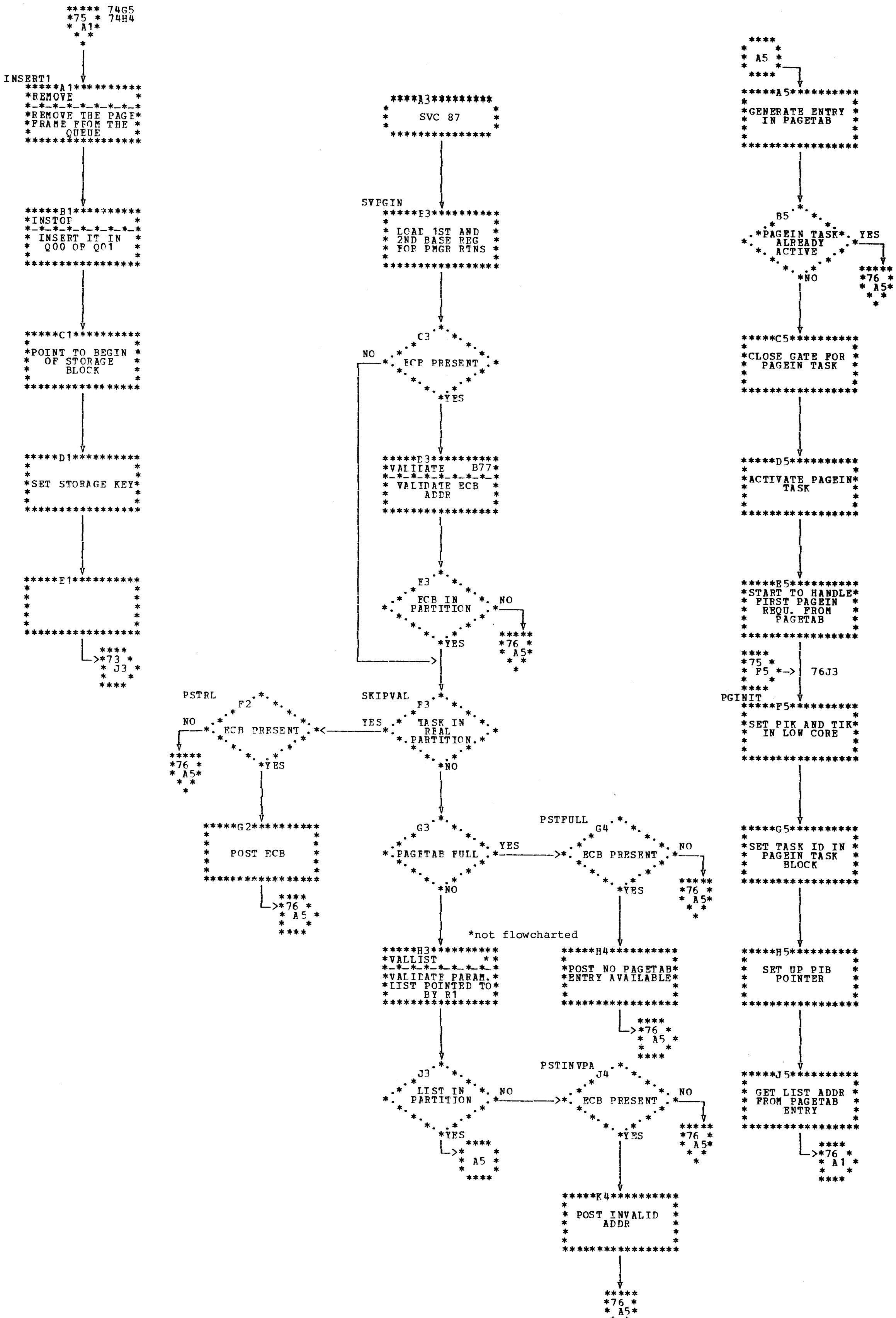


Chart B75. $$A$SUP1 - SGPOPT Macro, PAGEIN (SVC 87) Routine

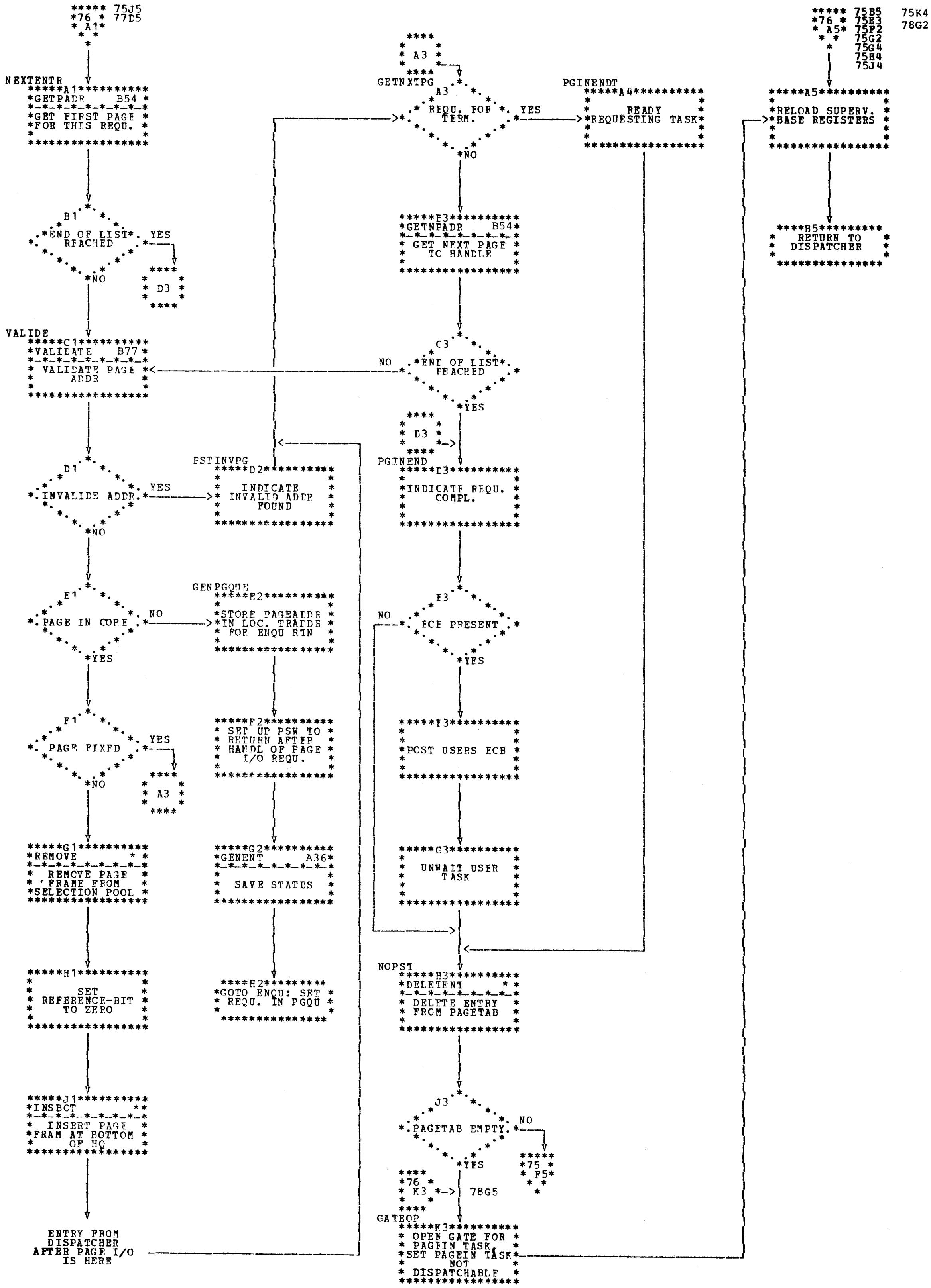


Chart B76. $$A$SUP1 - SGPOPT Macro, PAGEIN (SVC 87) Routine

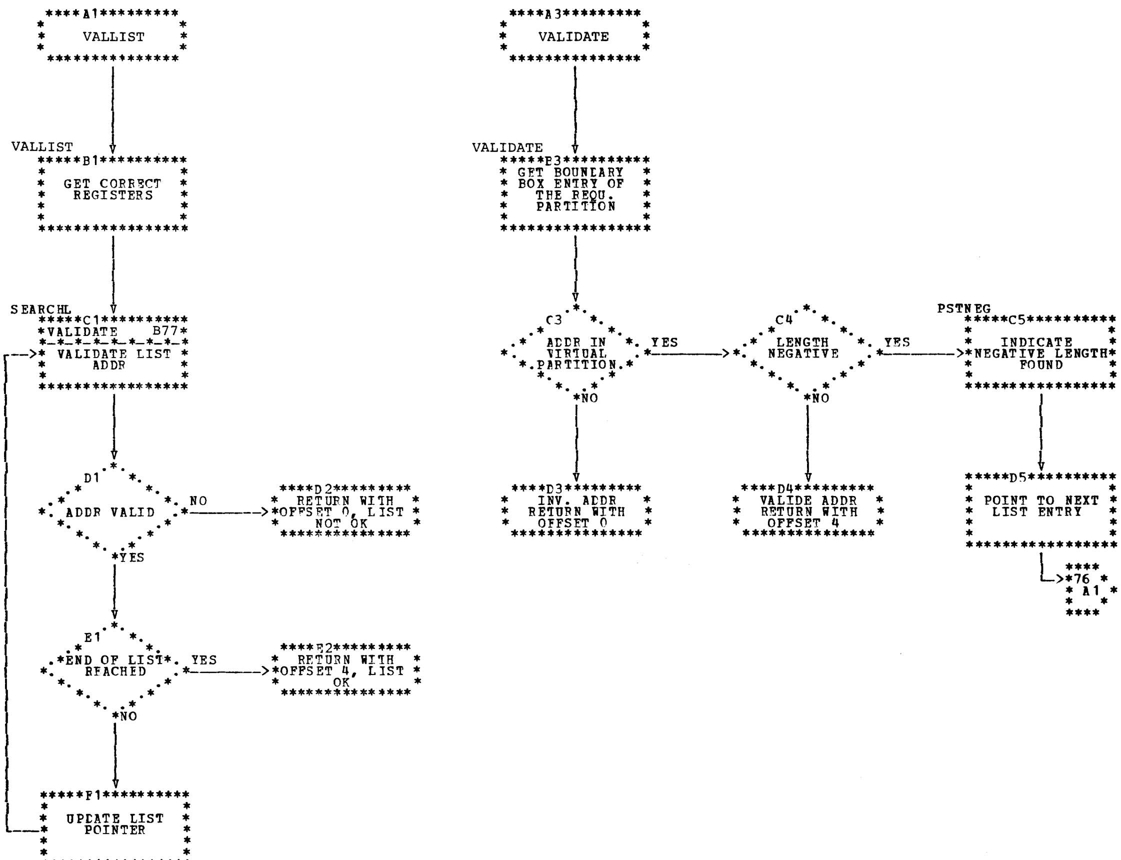


Chart B77. $$A$SUP1 - SGPOPT Macro, Validate Parameter List

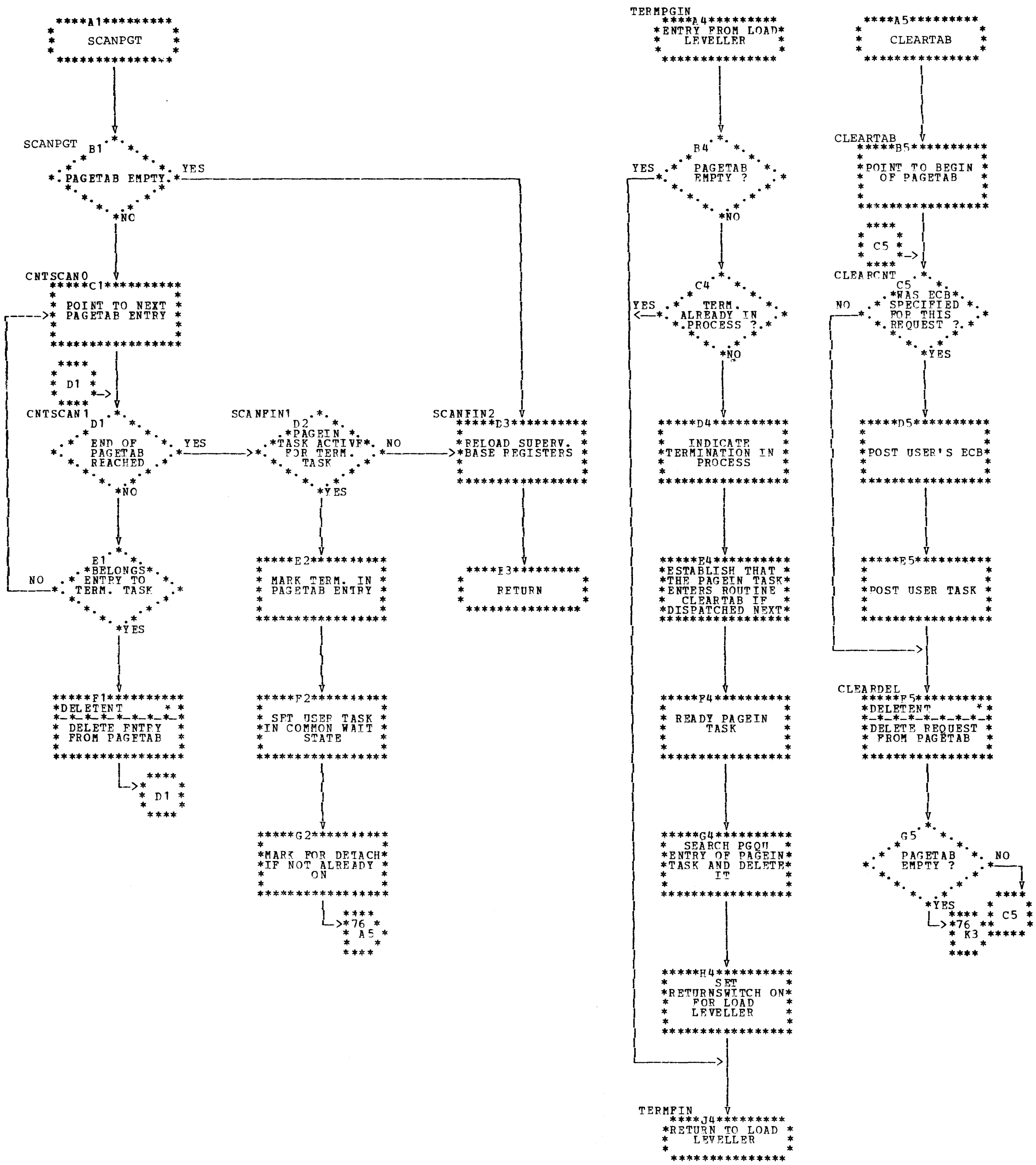


Chart B78. $$A$SUP1 - SGPOPT Macro, Delete PAGETAB Entries if Task Terminates

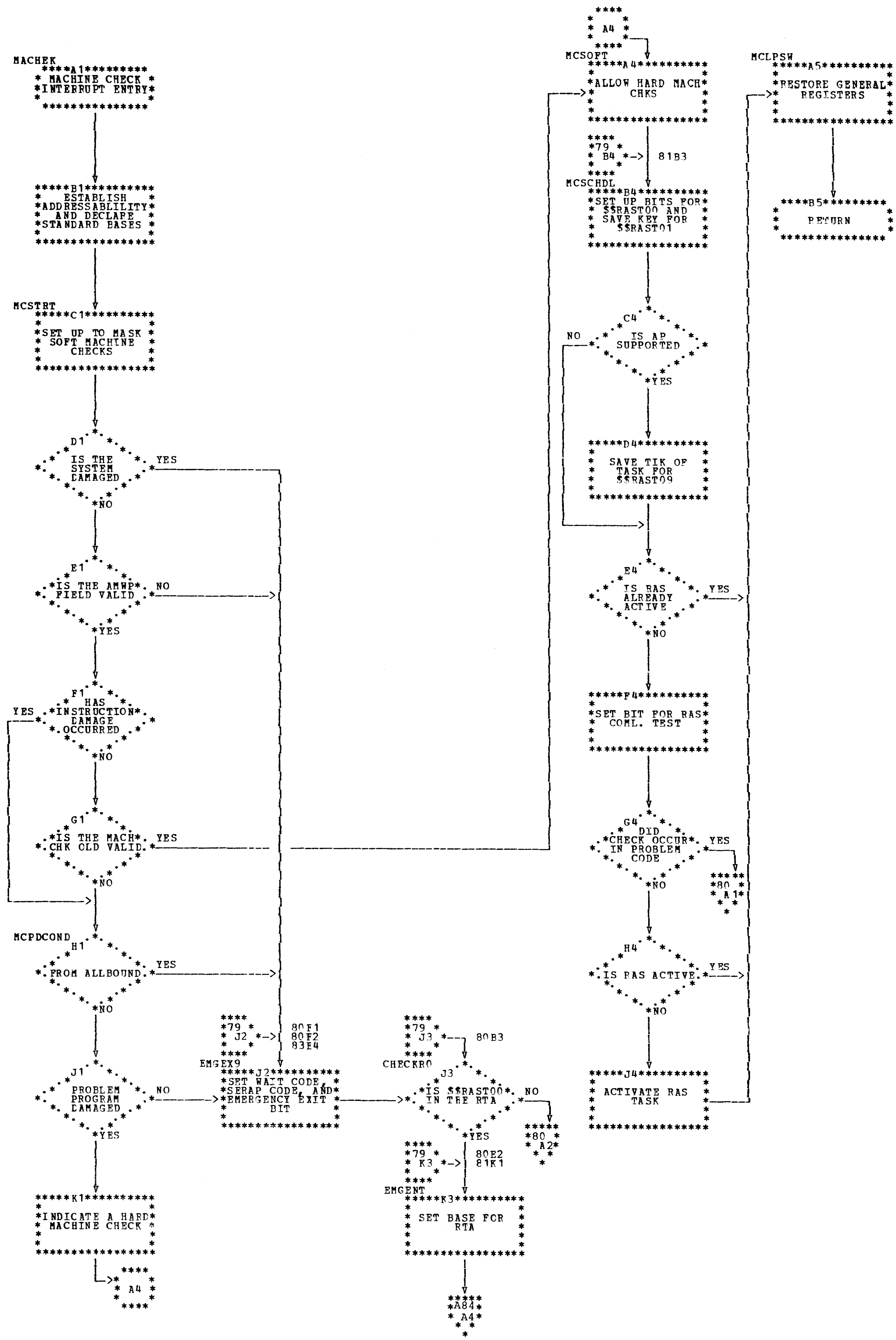


Chart B79. $$A$SUP1 - MCRAS Macro, Machine Check Handler
Refer to Chart 12.1.

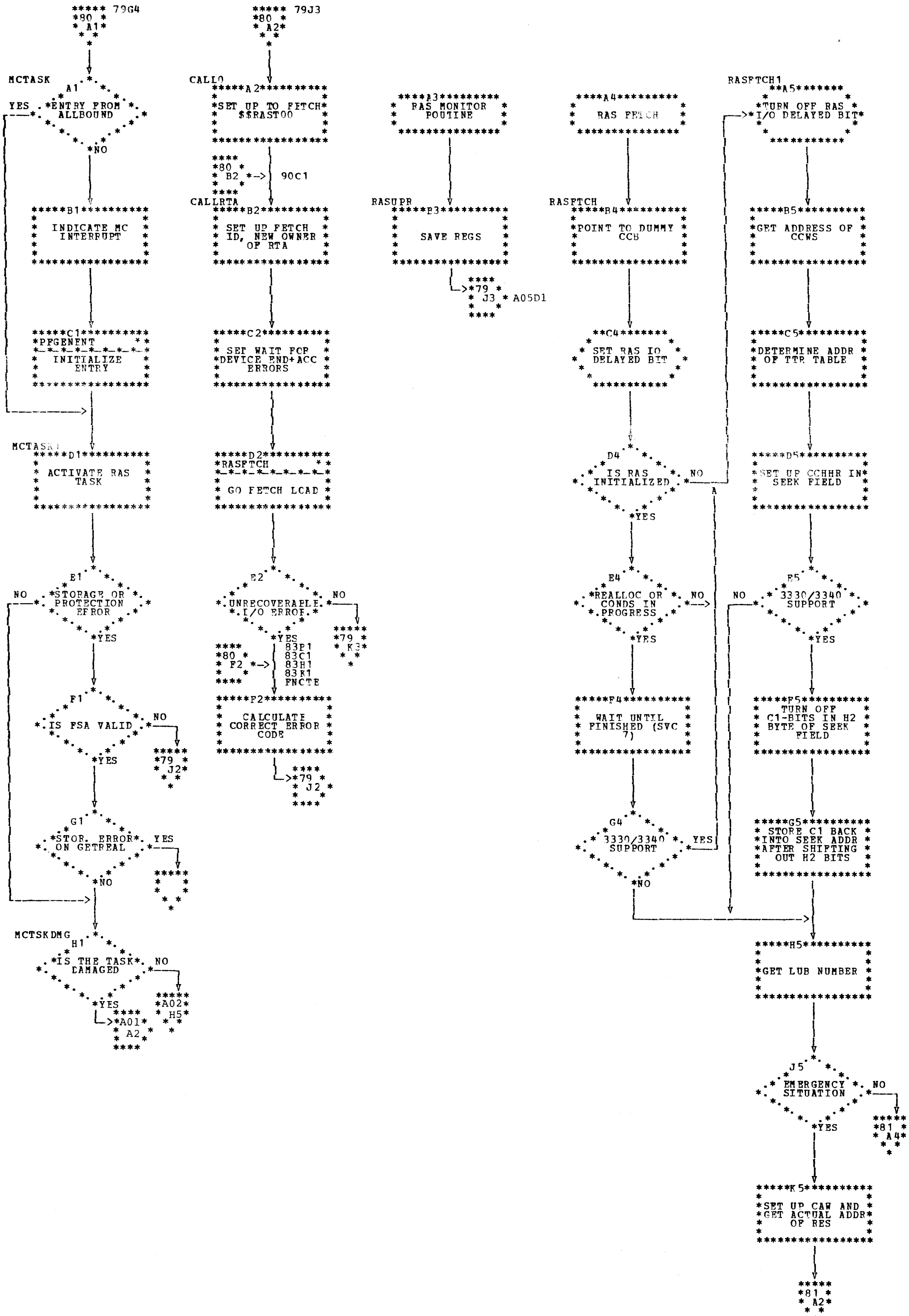


Chart B80. $$A$SUP1 - MCRAS Macro, Determine Machine Check Severity
Refer to Chart 12.1.

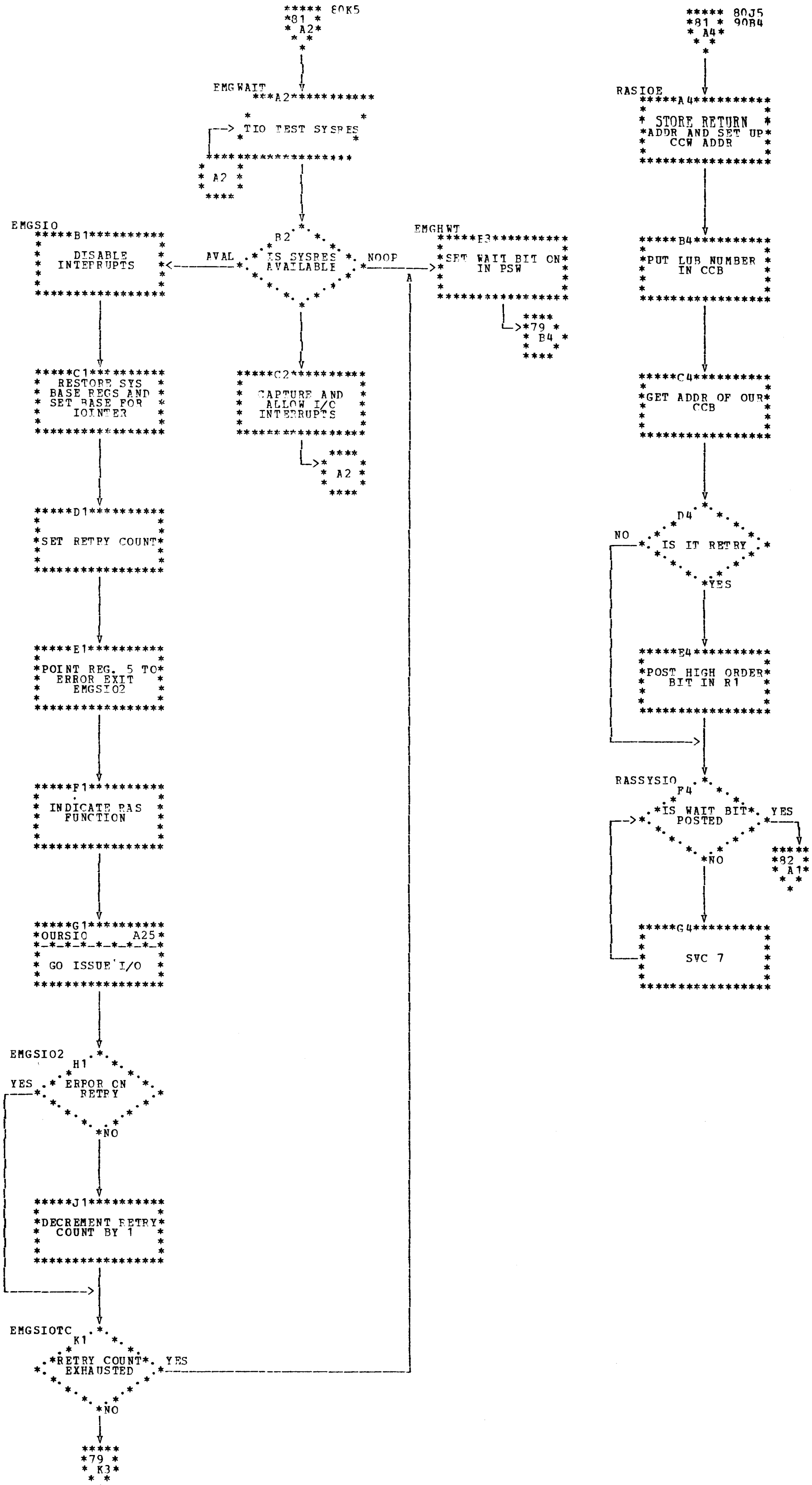


Chart B81. $$A$SUP1 - MCRAS Macro, Emergency Situation Handling
Refer to Chart 12.1.

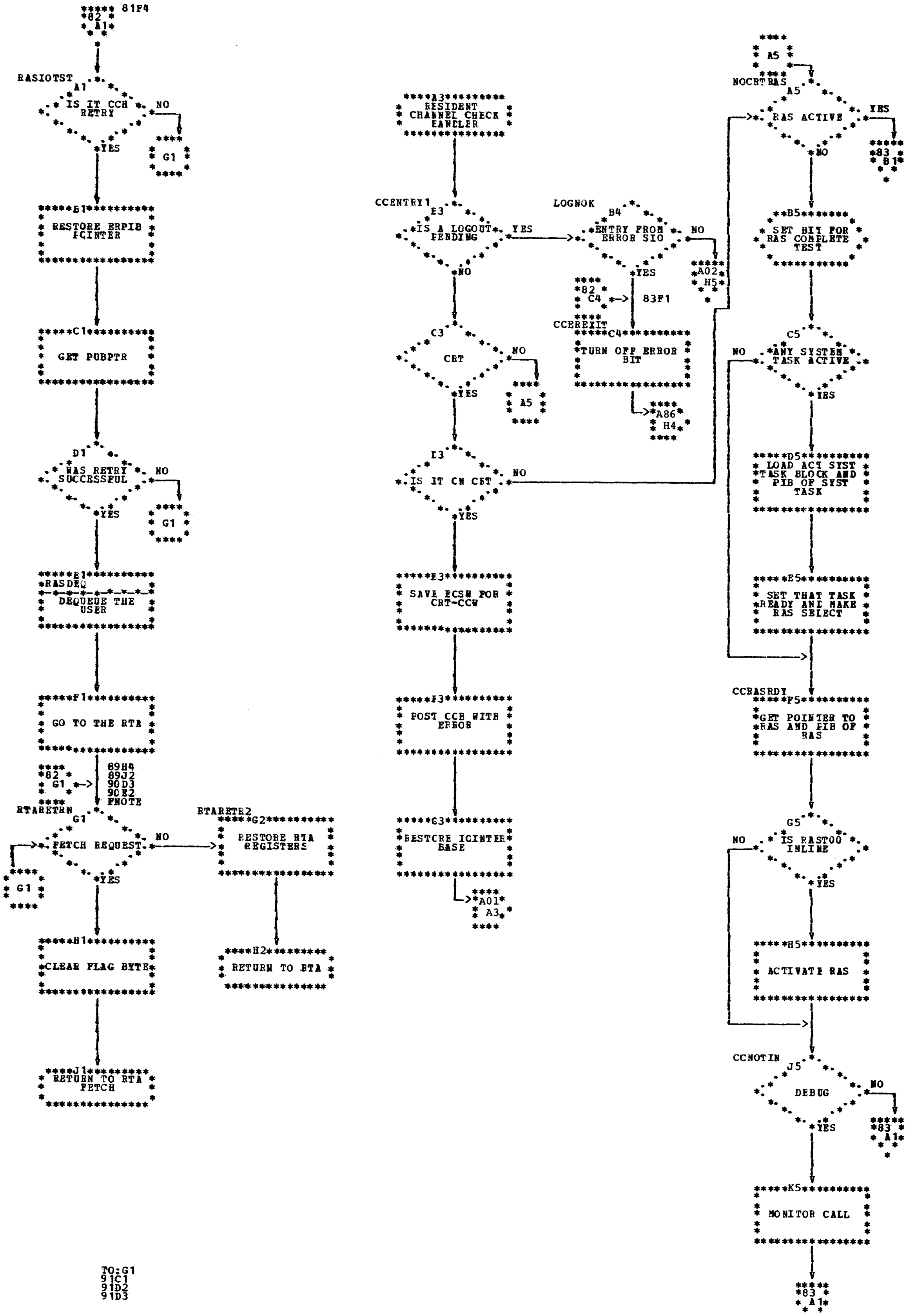


Chart B82. $$A$SUP1 - MCRAS Macro, Channel Check Handler
Refer to Chart 12.2.

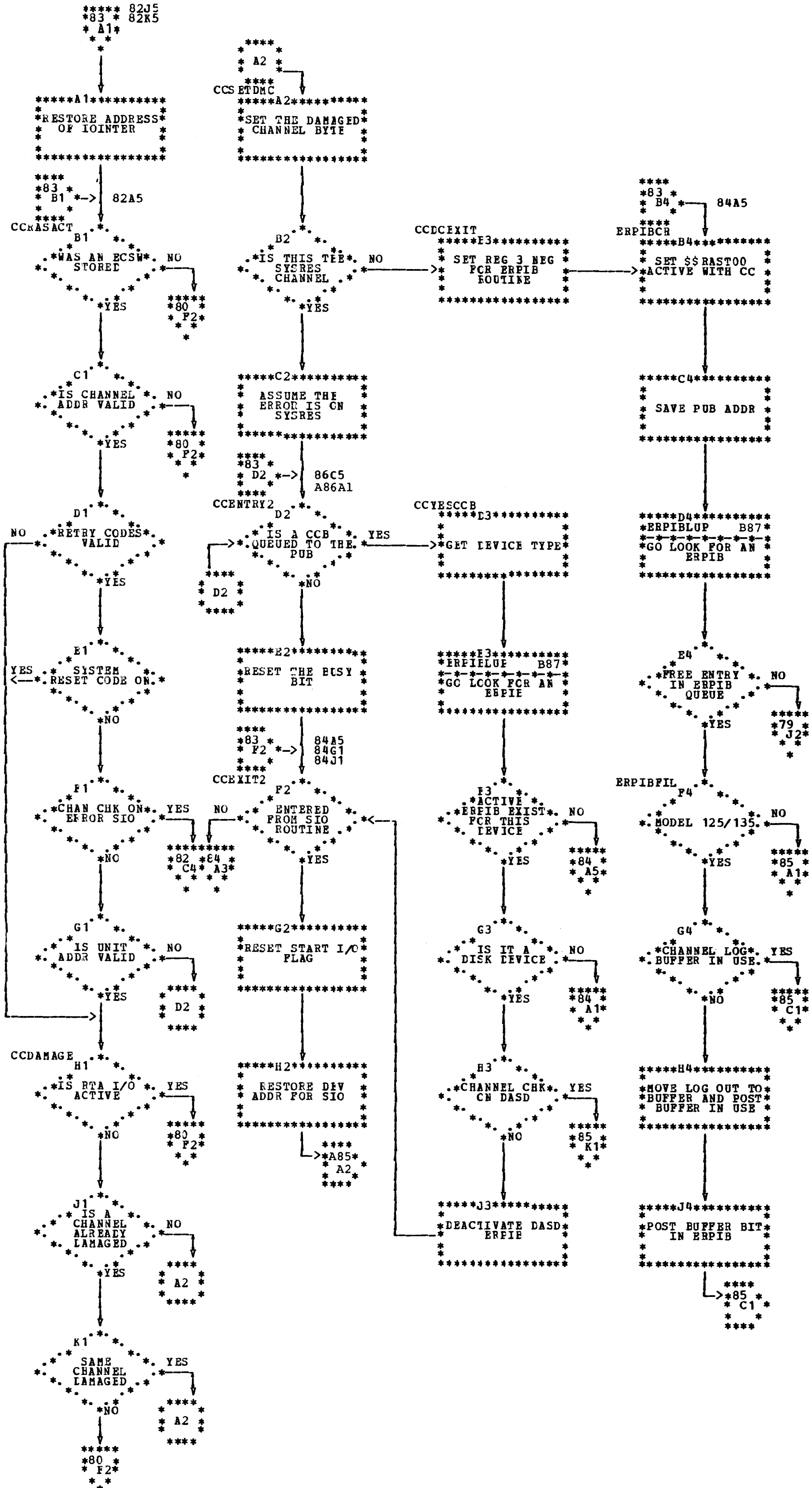


Chart B83. $$A$SUP1 - MCRAS Macro, Channel Check Handler
Refer to Chart 12.2.

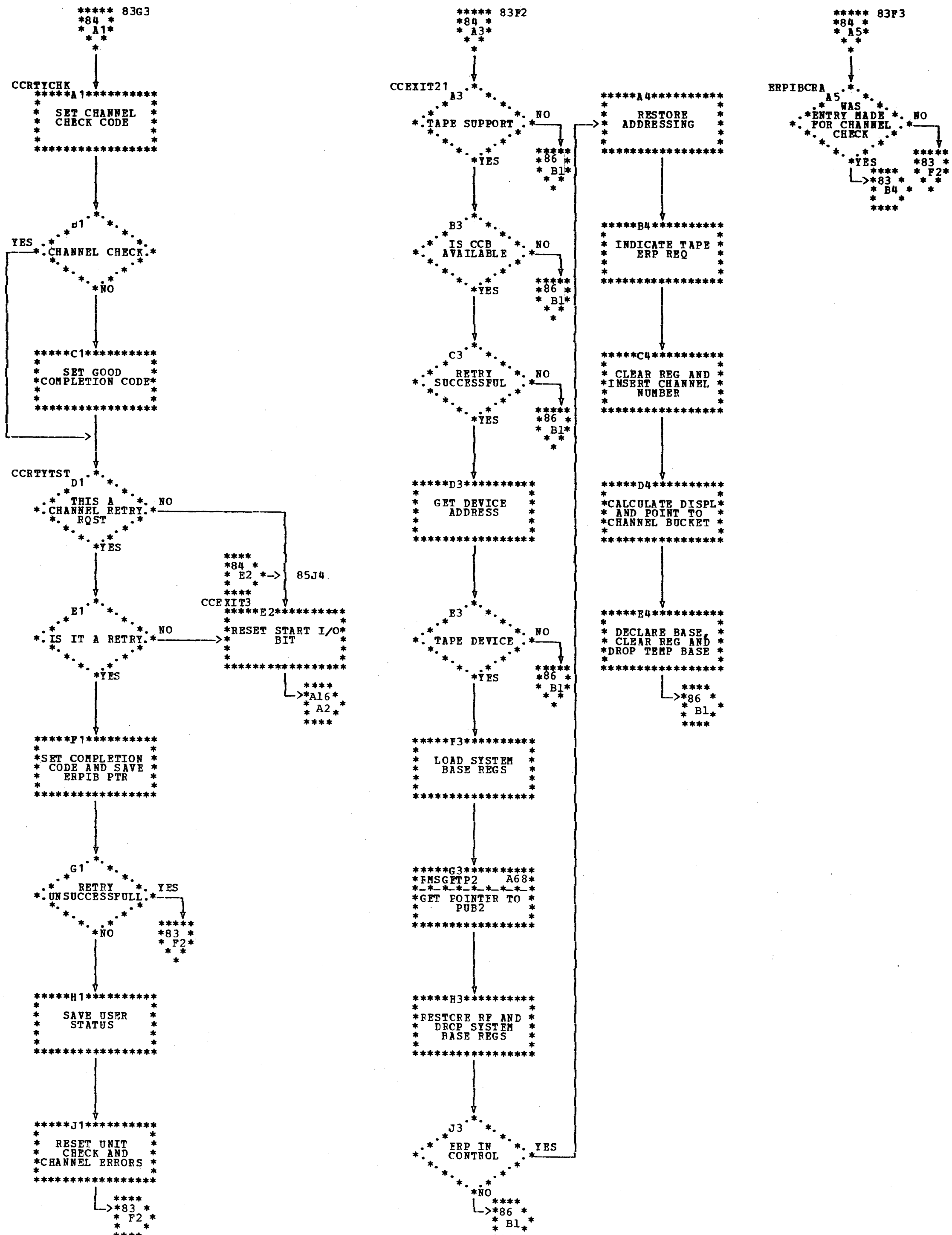


Chart B84. $$A$SUP1 - MCRAS Macro, Channel Check Handler
Refer to Chart 12.2.

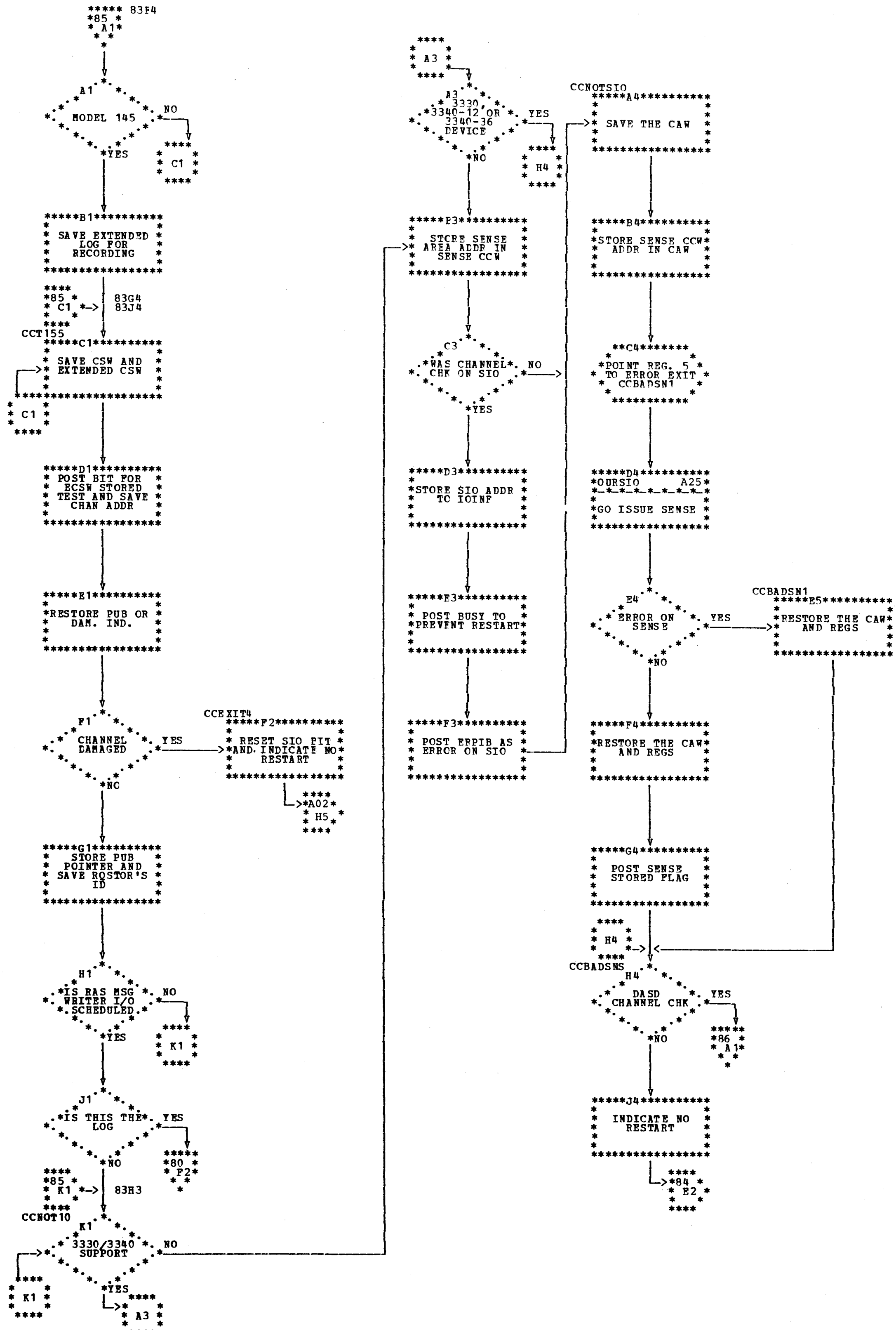


Chart B85. $$A$SUP1 - MCRAS Macrc, Channel Check Handler
Refer to Chart 12.2.

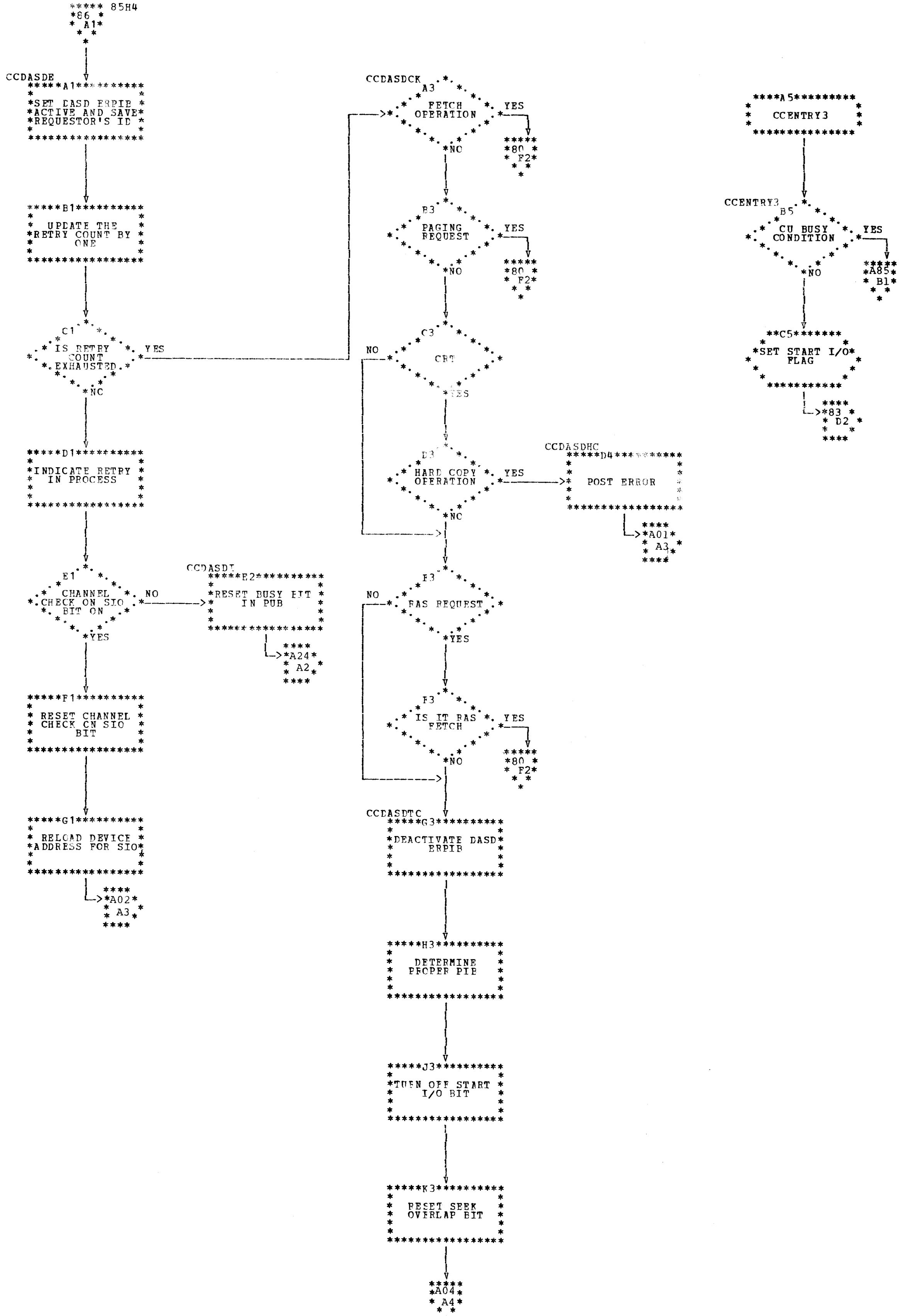


Chart B86. $$A$SUP1 - MCRAS Macro, DASD Channel Check Handler
Refer to Chart 12.2.

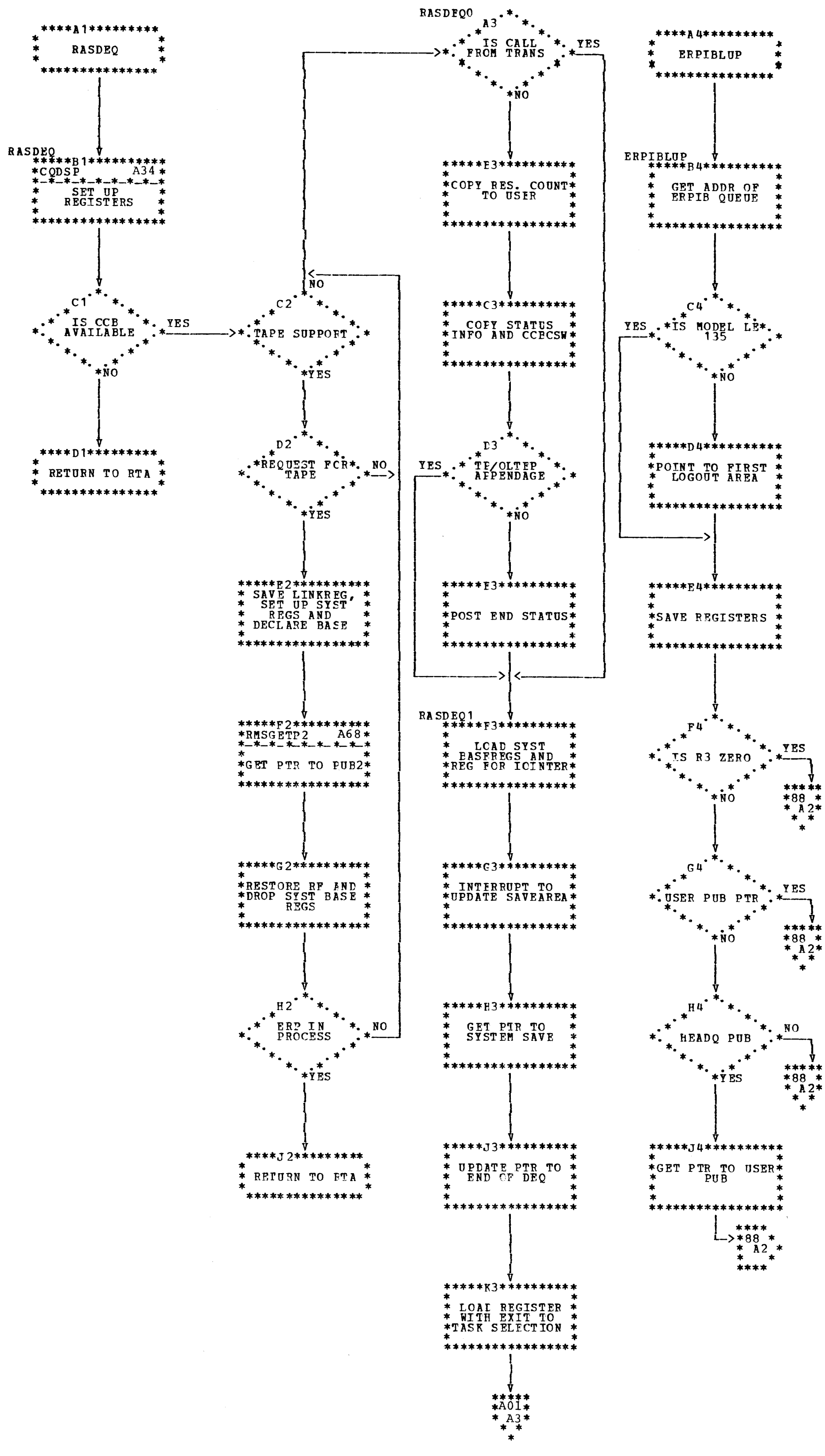


Chart B87. $$A$SUP1 - MCRAS Macro, Dequeue RAS I/O Request, ERPIB Queue Scan

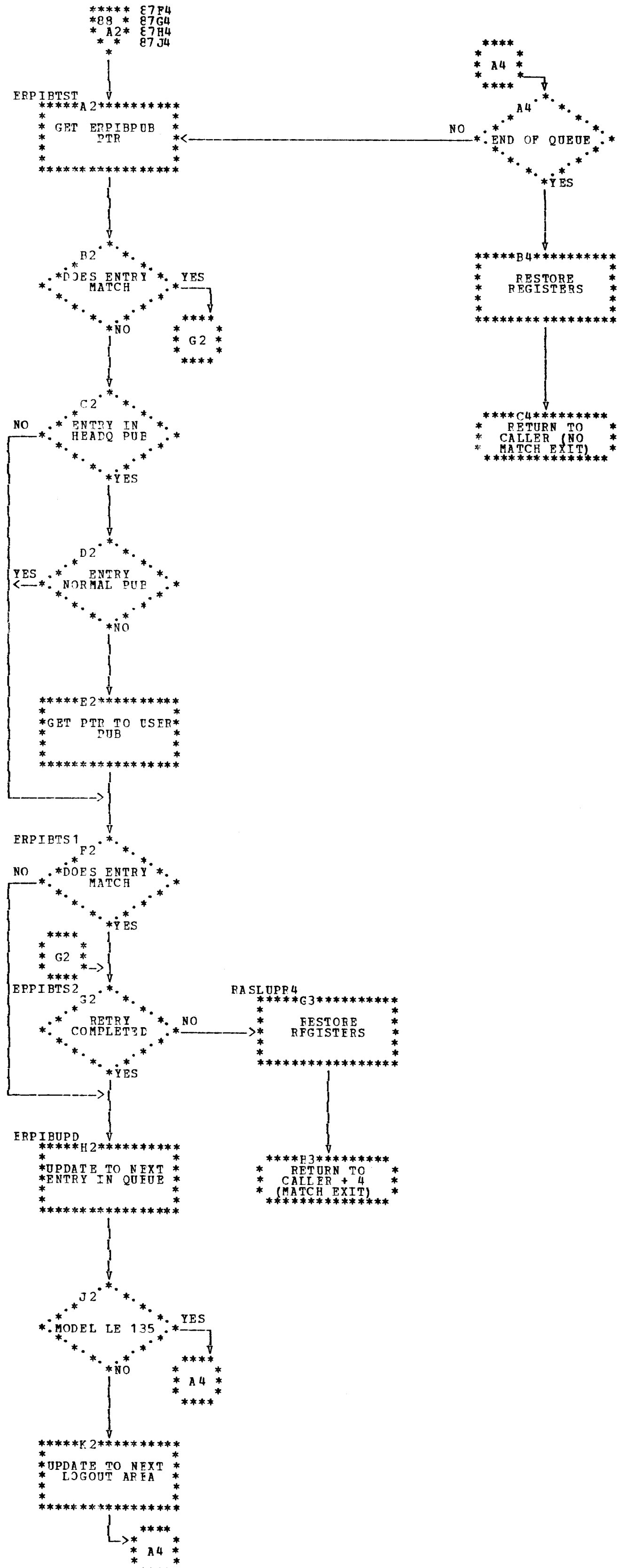


Chart B88. $$A$SUP1 - MCRAS Macro, ERPIB Queue Scan

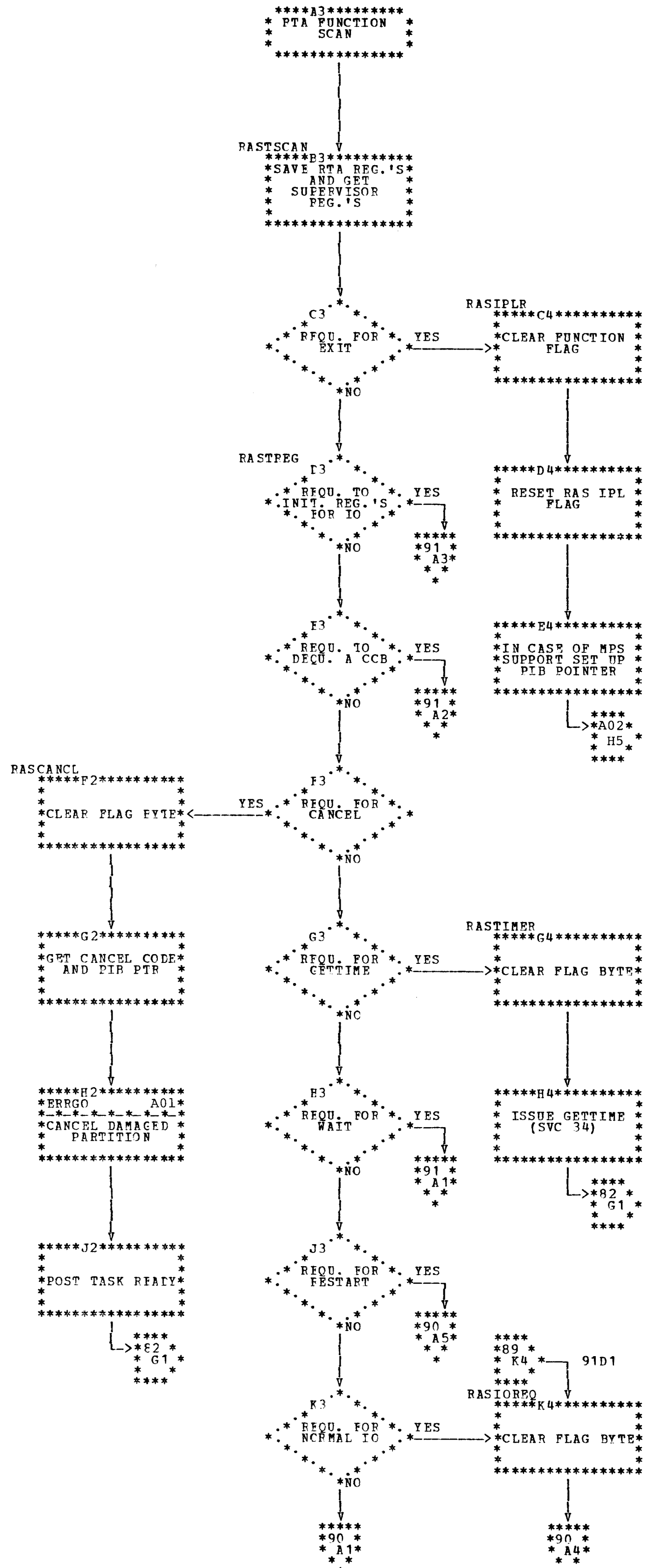


Chart B89. $$A$SUP1 - MCRAS Macro, RTA Function Scan
Refer to Chart 12.3.

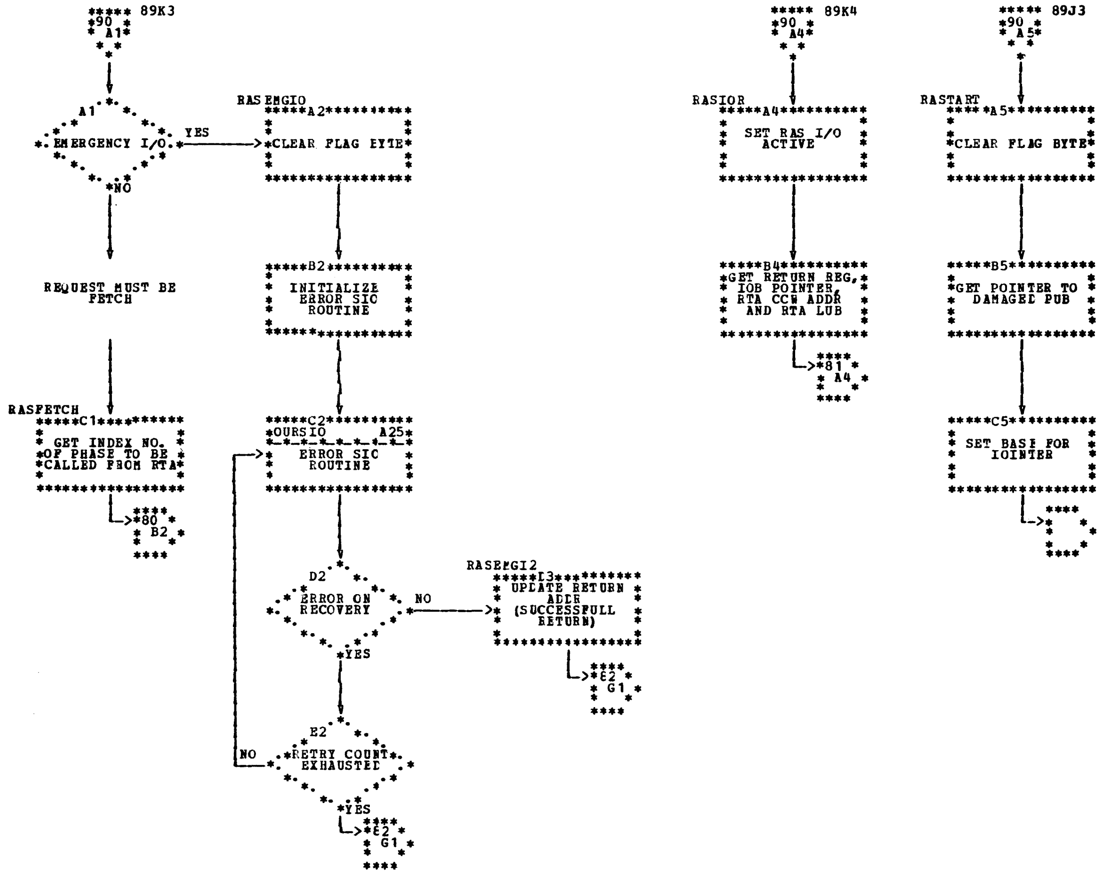


Chart B90. $$A$SUP1 - MCRAS Macro, Perform Functions for RTA
Refer to Chart 12.3.

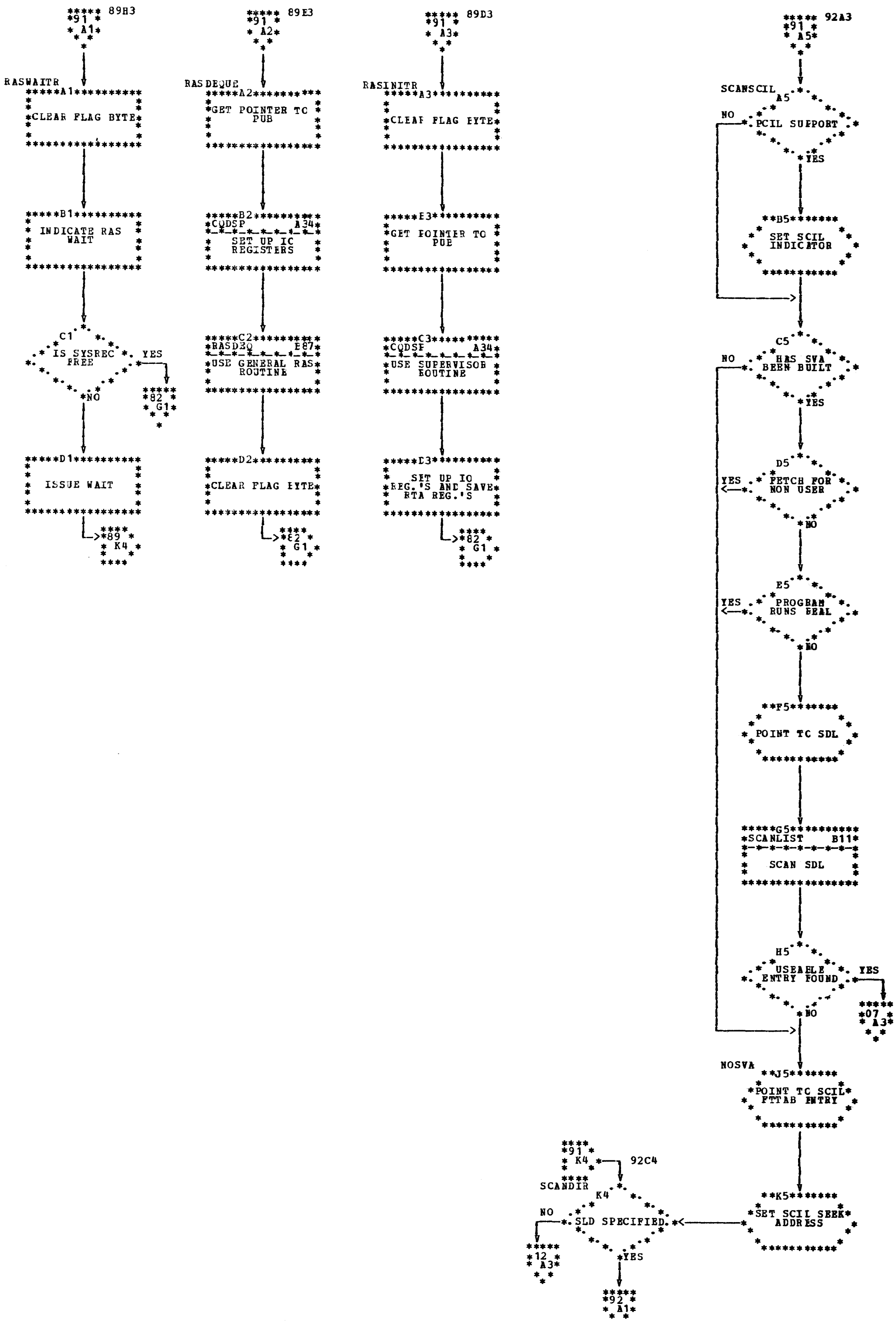


Chart B91. $$A$SUP1 - MCRAS Macro, Perform Function for RTA
SGDFCH Macro, Scan System Directory List (SDL)

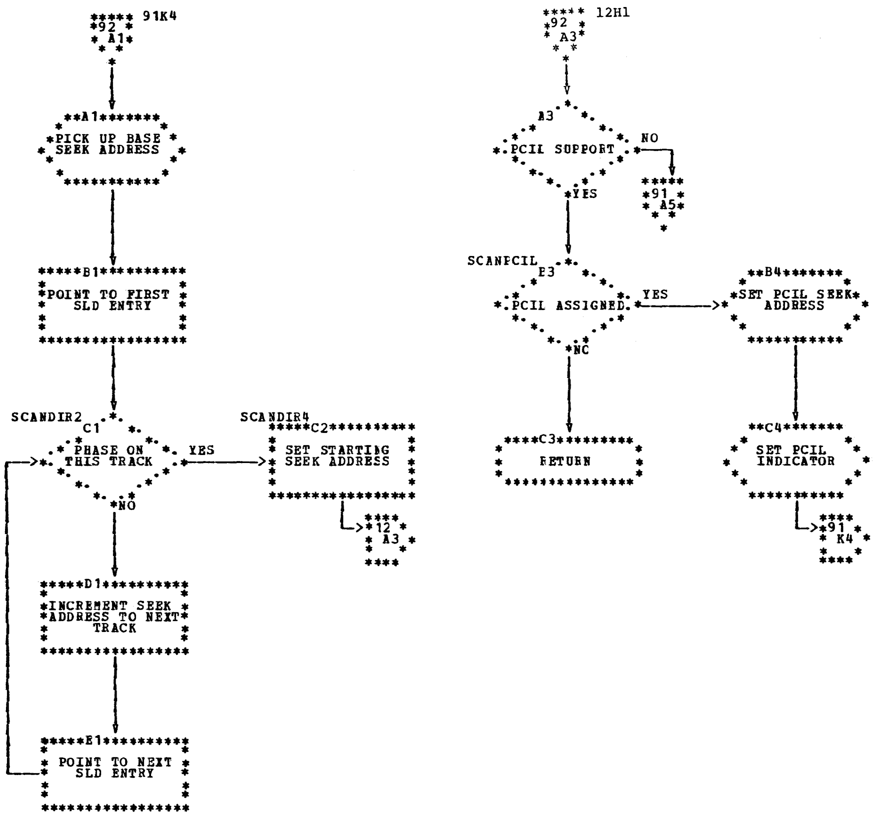


Chart B92. $$A$SUP1 - SGDFCH Macro, Scan System Directory List (SDL)

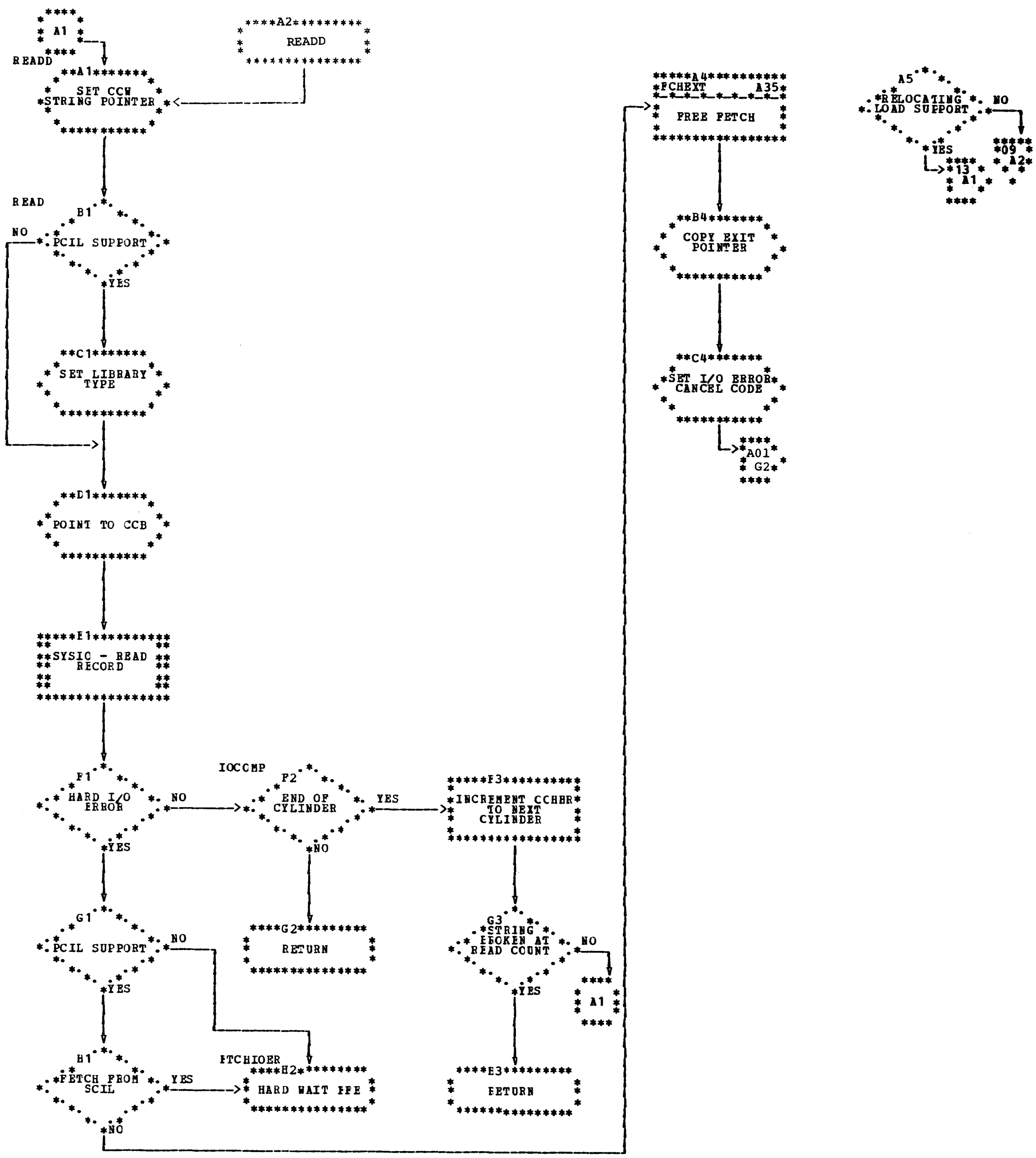


Chart B93. $$A$SUP1 - SGDFCH Macro, Read Directory Blocks

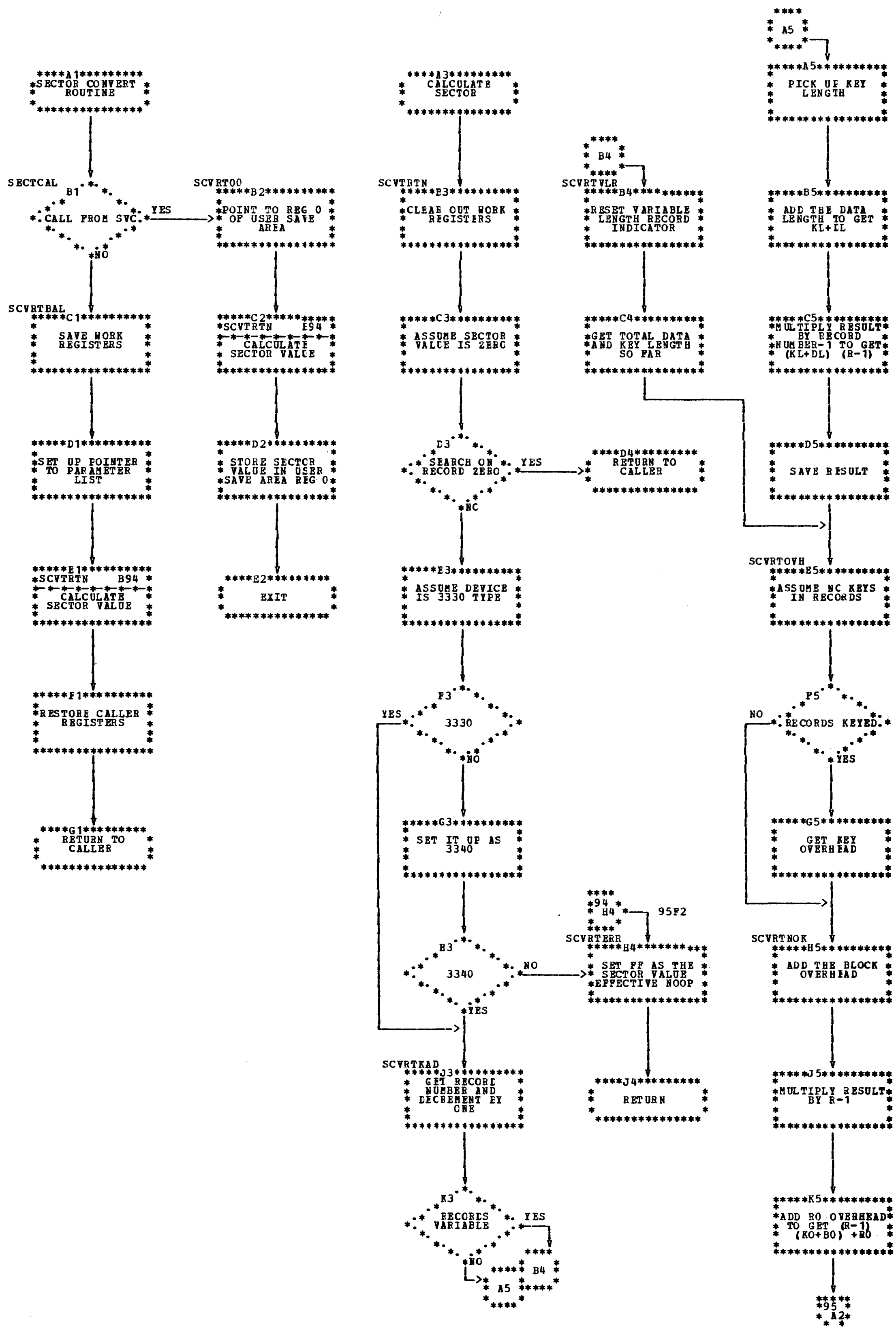


Chart B94. $$A$SUP1 - SGDFCH Macro, Sector Convert Routine (Part 1 of 2)

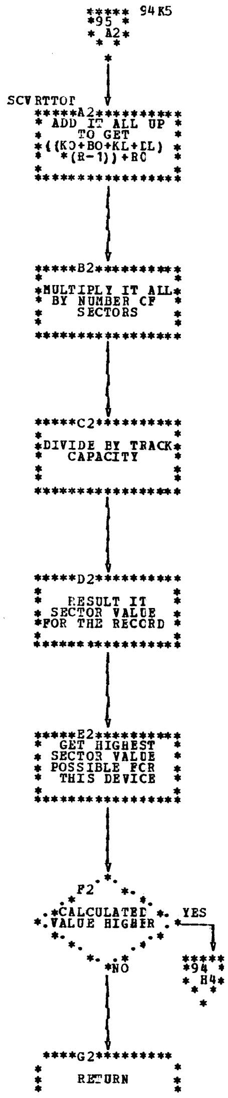


Chart B95. $$A$SUP1 - SGDFCH Macro, Sector Convert Routine (Part 2 of 2)

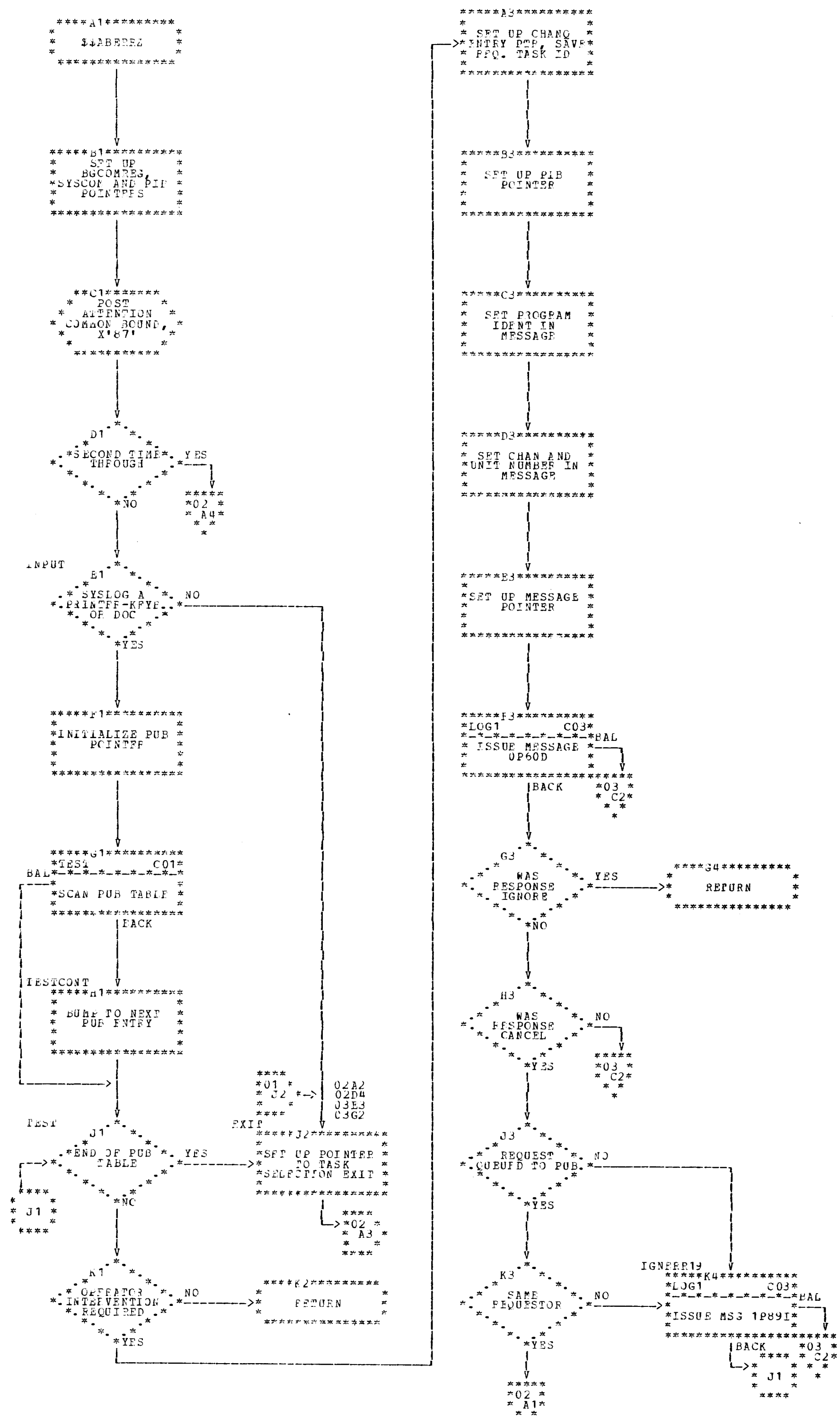


Chart C01. $$ABERRZ - Physical Attention
Refer to Chart 04.1.

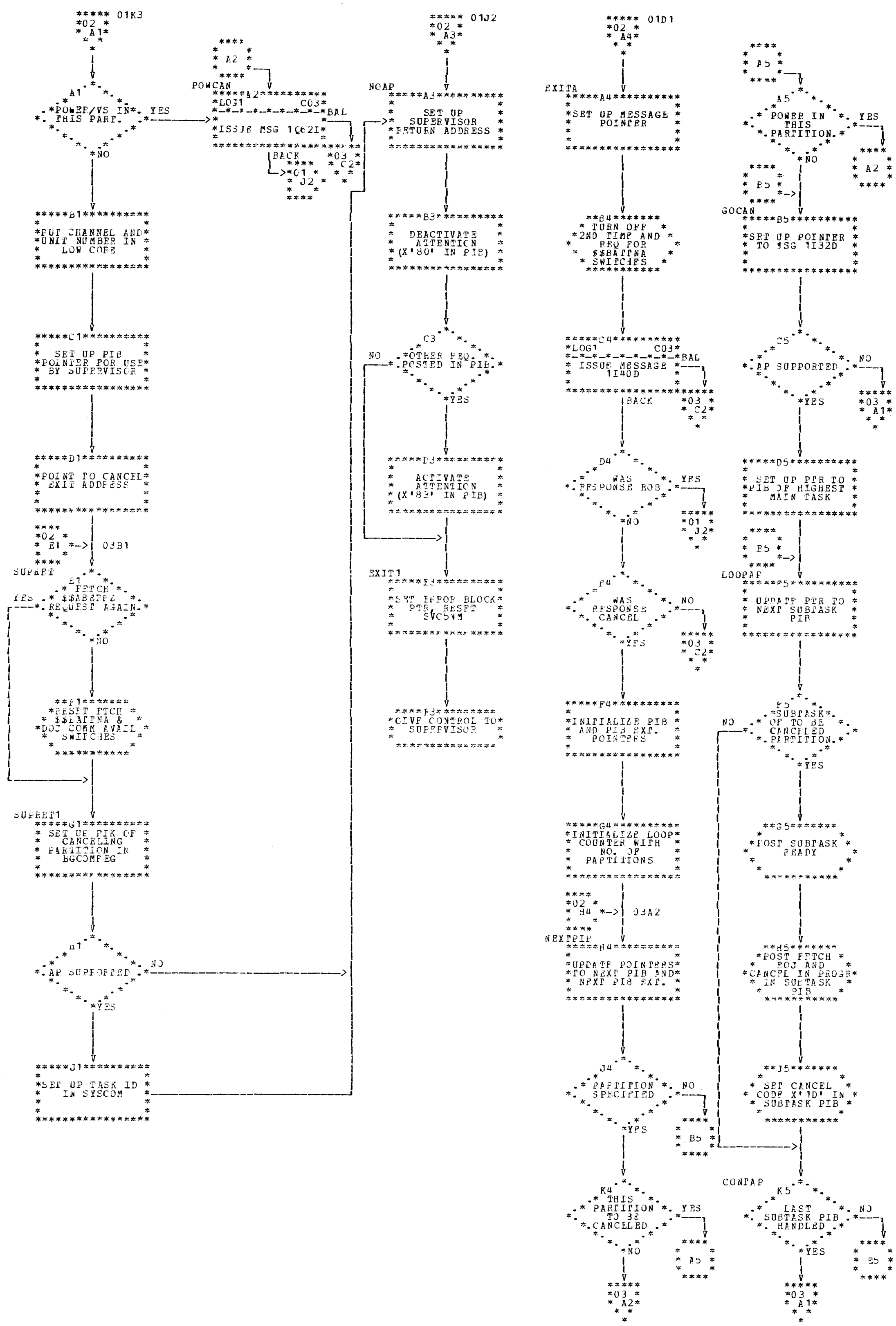


Chart C02. $$ABERRZ - Physical Attention
Refer to Chart 04.1.

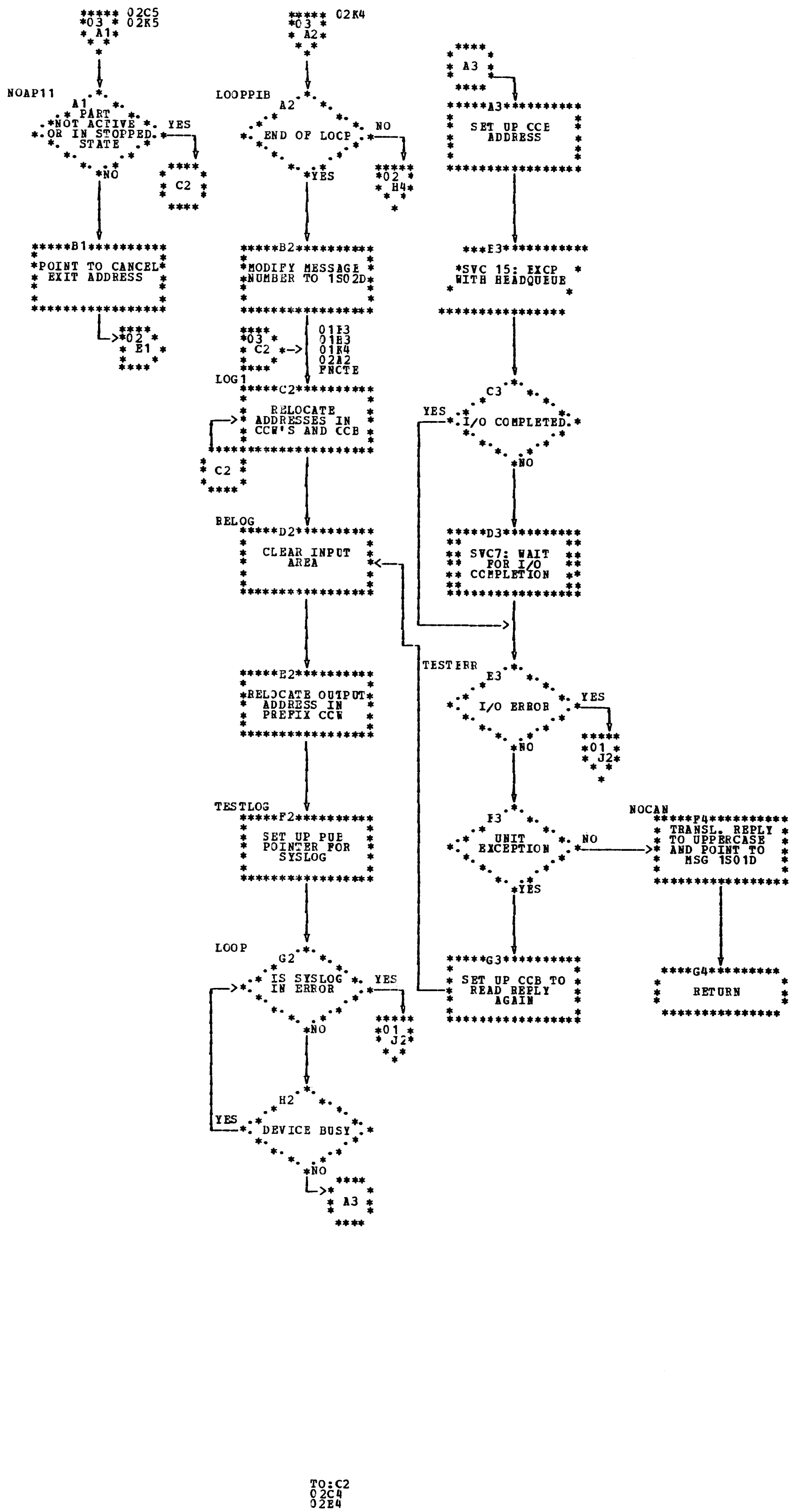


Chart C03. $$ABERRZ - Physical Attention
Refer to Chart 04.1.

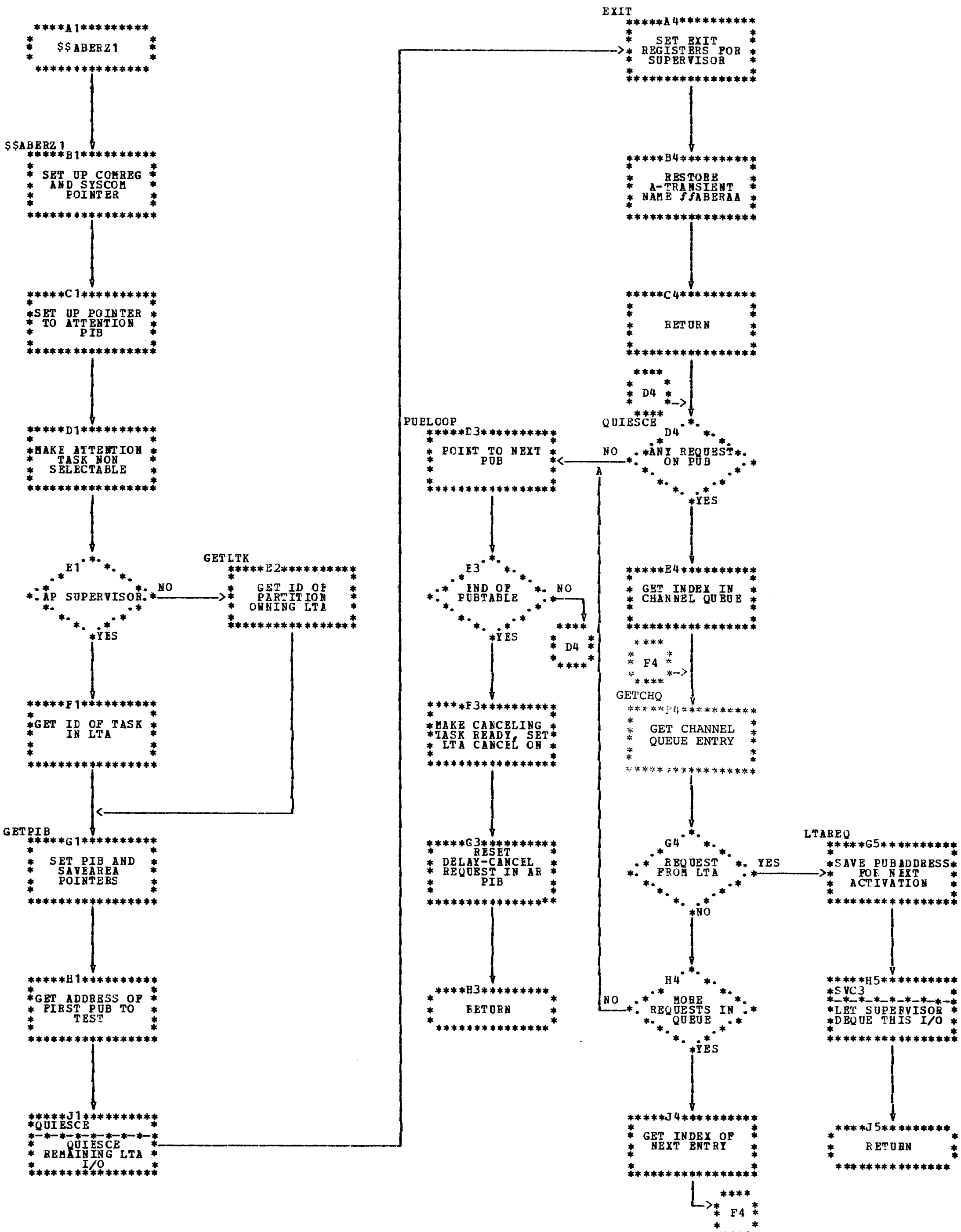


Chart D01. $$ABERZ1 - Quiesce LTA I/O
Refer to Chart 04.1.

Label	Location	Label	Location	Label	Location	Label	Location
ACTION	B74C4	CRTHCOK3	A22F1	CCWTR5	B15K2	CCDASDHC	B86D4
ACTSYS	A17A1	CRTHCOK4	A65K1	CCWTR6	B16B1	CCDASDI	B86E2
ADDAPIBT	B44J5	CRTIGN	A54K3	CCWTR7	B16D1	CCDASDTC	B86G3
ADD1	A23H2	CRTIGN2	A21E1	CCWTR8	B16D2	CCDCEXIT	B83B3
ADTST	A82B4	CRTNFND	B05B3	CCWTR9	B16D3	CCENTRY1	B82B3
ADTST1	A82C4	CSCANDE	B65K2	CCYESCCB	B83D3	CCENTRY2	B83D2
AEXIT	A37A1	CSWBT3	B27F5	CDATROUT	A06A3	CCENTRY3	B86B5
AFTTIO	A25H2	CSWCHK0	A25E1	CDLOAD	B69B2	CCEREXIT	B82C4
ALBEXIT	A36B2	CSWCHK01	A25G1	CEDETST	A86A2	CCEXIT2	B83F2
ALLBND	A04A5	CSWCHK1	A26A3	CEDETST1	A19A2	CCEXIT21	B84A3
ALLBND2	A04B5	CSWCHK2	A26F4	CHANSW	A01D5	CCEXIT3	B84E2
ALLOWAPP	A18F5	CSWSTORD	A84A5	CHCKAPP	A18C5	CCEXIT4	B85F2
AREXIT	A54E1	CSWTRANS	B28A1	CHECKR0	B79J3	CCNOTIN	B82J5
ASYNTST	A78A4	CSWTRBTM	B27A5	CHEND1	A21H2	CCNOTSIO	B85A4
ATL	B69G3	CSWTRSVC	B27D5	CHFAIL	A26A2	CCNOT10	B85K1
ATNRTN	A04B2	CSWTR3	B28G1	CHKEND	B73K1	CCRASACT	B83B1
ATTACHSA	B34G1	CSWTR4	B28A2	CHKFIRST	B07C1	CCRASRDY	B82F5
ATTNFCH	B04C2	CSWTR7	B28A3	CHKFREE	B06K2	CCRTYCHK	B84A1
AVAIL	A01G4	CSWTR8	B28C2	CHKNXT	B09J3	CCRTYTST	B84D1
BACKOUT	B09A2	CTD44	A64F5	CHKSEEK	A70G5	CCSETDMC	B83A2
BEGIN	B73C1	CYLEND	A28J5	CHKTRK	A71G5	CCT155	B85C1
BLDFEQU	B44E1	DASD2311	A08F4	CHKVALP	B74D1	CCWDEACT	B31A3
BNDRYCHK	B06E2	DASD2321	A08D5	CHNDRT	A22A5	CCWDE3	B31E3
BONCE	B04A3	DASD3330	A08E5	CHNOK1	A18A4	CCWDE4	B31F3
BQCL1	B60D4	DATACK	A27A5	CLEARCNT	B78C5	CCWGEN	B05H4
BQCL2	B60G2	DATEXIT	A03E2	CLEARDEL	B78F5	CCWRSVC	B29A3
BQCL3	B60H4	DCK370	A29K4	CLEARERQ	A64G4	CCWRSV0	B29C3
BQCL4	B60C3	DEACT	B64A3	CLEARPAR	B48A1	CCWRSV2	B29F3
BQCL5	B60E5	DEACTCD	B65B4	CLEARTAB	B78A5	CCWRSV3	B29D5
BQCL6	B60J4	DEACTDL	B66D3	CLRTEB	A15H2	CCWRSV4	B29D4
BQCL7	B60E3	DEACTGTR	B67E4	CLRUCUSR	A25C5	CCWTRANS	B19A1
BQCL8	B60B3	DEACTLL	B67G2	CLRUNC	A80B4	CCWTRBT2	B14A4
BRPFA	B32F1	DEACTP	B64A5	CNCEL	A10A2	CCWTRB1	B14D4
BTHIO	A52C2	DEACTPF	B66J2	CNCEL0	A74B3	CCWTR00	B14G4
BTHIO0	A52B2	DEACTR	B64G5	CNCEL2	A10D3	CCWTR01	B16A4
BTHIO1	A52C1	DEACTT	B65K3	CNCEL4	A10G4	CCWTR01A	B16B4
BTHIO2	A52E1	DEACTT2	B66A3	CNCEL5	A11D1	CCWTR02	B15D1
BTHIO3	A52F2	DEACTT4	B66B2	CNCLEXIT	A03F4	CCWTR1	B15A2
BTMRC12	B53C4	DEACTT5	B66F3	CNLRIS	A09B4	CCWTR10	B15G3
BTPFREE	B52B3	DEACTT6	B66A4	CNTSCAN0	B78C1	CCWTR11	B17A1
BTRET	B51D5	DEACTV	B64D5	CNTSCAN1	B78D1	CCWTR14	B15H4
BTRETRNS	A19G1	DEACTXX	B67C2	COMPITRQ	A07D1	CCWTR15	B15J4
CALLRTA	B80B2	DEACT3	B64A4	COMSVCE	A59C5	CCWTR16	B18A1
CALL0	B80A2	DEACT6	B64H4	COMSVCO	A60C1	CCWTR17	B18B2
CALTULP	A64D1	DELENT	B29A1	COMTABA	A60C2	CCWTR18	B18D2
CANCEL	A09C4	DELENTB	B29C1	COMTABB	A60D2	CCWTR19	B17A2
CANCEL0	A09A3	DELENT2	B29E2	CONDREQ	A12C1	CCWTR2	B15E2
CANCEL1	A09A4	DELETE	B37F4	CONTAP	C02K5	CCWTR25	B17B2
CANCEL2	A09E3	DELTHO	A73A4	CONVERT	A07K4	CCWTR26	B17C2
CANCEL2A	A09F3	DELTH1	A73E4	CORFIXIN	B27A1	CCWTR27	B17C3
CBFREE	A49H1	DELTH2	A73B5	CPGIN	B64B1	CCWTR28	B17D3
CBUF	B31A4	DEQU	B37A4	CPGIN1	B64G1	CCWTR29	B17E3
CCBADSNS	B85H4	DEQUPHO	B45C5	CPGIN2	B64B2	CCWTR3	B15G2
CCBADSN1	B85E5	DEQURQ	B67C1	CQDSP	A34A5	CCWTR30	B17E4
CCBMOVE	B30A3	DEQUX	B38A3	CRTATTNT	A65B5	CCWTR30A	B18A3
CCBNOTAV	A18F4	DEQUXE	B38C5	CRTDEACT	A80B3	CCWTR31	B18B3
CCDAMAGE	B83H1	DEQUY	B37K3	CRTENTRY	B01C1	CCWTR32	B18D3
CCDASDCK	B86A3	DETASK	B39A3	CRTFCH2	B10A4	CCWTR33	B18D4
CCDASDE	B86A1	DEVIRTP	B65A3	CRTHCOK2	A22D1	CCWTR4	B15H2

Label	Location
DFREER	B63A1
DIBENTRY	A50B4
DIB3540	A87C1
DISKERR	A26A5
DOCOMPR	A83B2
DPFIX	B58A1
DPFIX1	B58D1
DPFREE	B62A4
DPFREE1	B62H4
DPFREE2	B62E5
DTCHFREE	A72H4
DTFIX	B57A1
DTFIXSTP	B57C1
DTFIX1	B57B2
DTFIX2	B57B3
DTFIX3	B57H4
DTFIX4	B57G3
DTFIX5	B57D1
DTFREE	B62A1
ECFCEDE	A30G5
ECFCPEND	A32E2
ECFDCHN	A32A3
ECFECRCD	A31A4
ECFECSEG	A30H3
ECFINCR	A30F3
ECFLCOP	A30H1
ECFNEWCT	A32H3
ECFNCMT	A32D2
ECFNCTIC	A31J4
ECFNCTPM	A30A4
ECFTRUNC	A31A5
EMGENT	B79K3
EMGEX9	B79J2
EMGHWT	B81B3
EMGSIO	B81B1
EMGSIOTC	B81K1
EMGSIO2	B81H1
EMGWAIT	B81A2
ENQDEQ	A80A1
ENQU	B34A4
ENQUPHO	B35A3
ENTEXT	A46A2
ENTIC	A36A3
ENTPCK	B32A4
ENTRYFND	B07A3
ENTSVC	A46A1
ENVD370	A29F5
EOC370	A31E1
ERPEXIT	A53B4
ERPIBCR	B83B4
ERPIBCRA	B84A5
ERPIBFIL	B83F4
ERPIBLUP	B87B4
ERPIBTST	B88A2
ERPIBTS1	B88F2
ERPIBTS2	B88G2
ERPIBUPD	B88H2
ERPNCRM	A54B1
ERPSVC	A10A1
ERRGC	A01G2
ERRINVP	B46E3
ERRLCG1	A29G3
ERRLCG2	A29H3
ERROR1	B23E2
ERROR2	B14E5
ERROR3	B26D3
ERROR4	B21D3
ERROR5	B30A5
ERROR6	B31A1
ERRREL	B31A2
ERRSET	A34B4
ERRSETA	A34C4
ERRSIO	A25E3
ERRSIO1	A25D2
ERRXX	A01A2
ERRYY	A04A4
ERR10	A01C2
ERR25C	A69G3
ERR3330	A29C3
ERR3340	A29A3
EUECB	A65B3
EUTPUNCK	A20C4
EUTURNON	A65D2
EXCH	B59A2
EXCHA	B58C4
EXCHANGE	B58A3
EXCHPGES	B55E4
EXCHR	B46A2
EXCH1	B59E3
EXCP10	A49F2
EXCP2	A47H1
EXCP3	A47C2
EXCP4	A48A1
EXCP5	A49G4
EXCP6	A49F4
EXCP8	A49D2
EXCP9	A47G3
EXIGN	A21F1
EXIGN1	A21F2
EXIT	A02H5
EXIT	C01J2
EXIT	D01G4
EXITA	A85B5
EXITA	C02A4
EXITRTRN	A60H5
EXITSAV	A60H4
EXITTOD	A63C5
EXIT1	C02E3
EXTERN	A36J1
EXTIC	A29A1
EXTIC0	A29C2
EXTIC1	A29G1
EXTMIC	A37C3
EXTOC	A37D3
EXTRAN	A20H4
EXTRANAE	A21D3
EXTRETRN	A06J3
EXTRTN	A37B3
EXTTIM	A33F4
EXWHY	A29H1
FASTCE	A86B4
FASTCE1	A19G4
FCHCCNT	B04E2
FCHEXIT	A35A1
FCHEXT	A35A3
FCHUPD	B10A1
FCHUPD1	B10E1
FCHZ	A04F2
FETCH	B03B2
FETCHERP	B10A5
FFPF	B57C3
FFPF1	B57E2
FFPF2	B57F3
FFPF3	B57F2
FFPF4	B57D3
FINDEACT	B65C4
FINDMAIN	A77C1
FINDPUB	A18G3
FINSCAN	B65E5
FIXAREA	B23A1
FIX1	B23A3
FIX3	B23F3
FIX4	B23A4
FIX5	B23J4
FIX6	B23K4
FIX7	B23D5
FRCBUF	B31C5
FREDEV1	A16G4
FREDEV2	A16H4
FREE	B24A4
FREELOOP	B10A2
FREEREAL	B43G3
FREEVIS	B72B1
FREE1	B24C4
FTCHIOER	B93H2
GATEOP	B76K3
GATEWAIT	A80E2
GENENT	A36E3
GENPGQUE	B76E2
GEN1	A36E4
GETAB	A60H3
GETADD	A05K5
GETBLOCK	B26A1
GETBL1	B26B2
GETBL3	B26B3
GETCCW	B19A2
GETCCW1	B19A3
GETCCW2	B19B3
GETCCW3	B19G3
GETCCW4	B19B4
GETCHQ	D01F4
GETDADR	A70D1
GETD0	A70A3
GETD1	A70B4
GETD2	A70F2
GETD3	A70J2
GETERP	A05E4
GETLTK	D01E2
GETNPADR	B54B3
GETNXTPG	B76A3
GETNXT1	A08G4
GETNXT3	A08F5
GETPADR	B54H2
GETPC	A60B2
GETPFTE	B50A1
GETPHASE	B08A3
GETPIB	A77E1
GETPIB	D01G1
GETPUB	A47B1
GETPUBOW	A65H5
GETREAL	B42A1
GETSAVE	A75A1
GETSAV1	A74A4
GETSEN	A26A1
GETSKFLD	A72G4
GETTCB	B25A2
GETUTADD	A05K4
GETVIS	B70B5
GETVISB	B72B2
GETVIS2	B70F5
GETVIS20	B70H5
GETVIS21	B70J5
GETVIS22	B71A4
GETVIS23	B71C4
GETVIS25	B71A5
GETVIS26	B71D3
GETVIS27	B71E2
GETVIS28	B71F2
GETVIS29	B71G2
GETVIS3	B71F5
GETVIS30	B72E2
GETVIS32	B72F3
GETVIS33	B72H3
GETVIS34	B72B3
GIOADR	A01K3
GNENTRY	B38D2
GOAPPND	A19D1
GOCAN	C02B5
GOITROUT	A33A1
GOPTA	A05F4
GPFTBNDS	B55C5
GPFTEBDS	B55B5
GTVSINIT	B72B4
GTVSINT1	B72A5
HALT	A52G3
HALTLOOP	A51G3
HARDWT	B32C5
HARDWTF9	B61J3
HOLDFREE	A72A4
HQSCAN	A24A3
HQSCAN1	A24B3
IDAL0	B21C2
IDAL3	B21A4
IDAL4	B21E4
IDAL6	B21H4
IDAL7	B21F5

Label	Location
IDAL8	B21K5
IGNERR19	C01K4
IGNLUB	A47A4
IGNORE	A22H3
IMDCMD	A85J2
INDIB	A50D4
INDSVP	B35C5
INIT	B73H3
INITAB3	B02K4
INITAB5	B02F4
INITAB6	B02G5
INITAB8	B02C5
INITBUFR	B03E1
INITCNCL	A12B5
INITDAT	B02C1
INITHQ	A17A2
INITHQIO	A16G5
INITHQ1	A16H5
INITPT	B02C4
INITRG	A16A2
INITRG1	A16A3
INITRG2	A16G3
INITRG3	A16E4
INITRTN	A17E2
INITSIO	A24A2
INITSVC	B39A1
INPUT	C01E1
INSERT	B49A1
INSERT1	B75A1
INSNR	B46J2
INVADR	B50A5
INVPAGE	B49A3
INVPAGE1	B48A4
INVPAR	B46A3
IOCOMP	B93F2
IOPSET	A52D4
ITF370	A29B5
ITROUT	A07J5
JAENTRY	A37D1
JATIMER	A46A3
JAXIT	A46B4
JAXITS	A46C4
JAXPP	A46E5
JAXRET	A46K4
JIBENTRY	A50B1
LAPGGU	B37D4
LDWRKRG	B44D2
LENQADR	B42H1
LISTDIB	A50B5
LISTSCAN	A77K4
LOADRA1	B50B4
LOADRA2	B49H4
LOCATE	B22A1
LOCATE1	B22A4
LOC1	B22C1
LOC4	B22F2
LOC5	B22E3
LOGNCK	B82B4
LOGPRC	A02B2
LOG1	C03C2
LOOKFPF	B51A1
LOOP	C03G2
LOOPAP	C02E5
LOOPCHK1	B06D4
LOOPCHK2	B09K5
LOOPPIB	C03A2
ITA	B01C2
ITAPROC	A36F5
ITAREQ	D01G5
LUWREGS	B56E2
MACHEK	B79A1
MAINENT	A77D4
MAINTASK	A77A1
MCLPSW	B79A5
MCPDCOND	B79H1
MCSCHDL	B79B4
MCSOFT	B79A4
MCSTRT	B79C1
MCTASK	B80A1
MCTASK1	B80D1
MCTSKDMG	B80H1
MICRERR	A83E4
MODE1052	A22A3
MOVECCB	B39B4
MOVECCB0	B30C3
MOVECCB5	B30D2
MOVECCW	B06D1
MOVECSW	A20C2
MOVERQ	A21H4
MTIEND	A13B5
MTILAB1	A14A4
MTILAB2	A14B5
MTILAB3	A14D4
MVENTRY	B08A2
MVZEX	A20G3
NCHFSIO	A85A2
NEWPAGE	B73J3
NEXT	B73K4
NEXTE	B46D4
NEXTENTR	B76A1
NEXTERQ	A21E4
NEXTJIB	A50A1
NEXTPAGE	B50E1
NEXTPFR	B51G1
NEXTPIB	C02H4
NEXTRLD	B09E4
NEXTRLD1	B09G5
NEXTRLD2	B10A3
NGTMBP	A57A5
NOABCNCL	A10K2
NOAP	C02A3
NOAP11	C03A1
NOATTENT	A21G1
NOBTDEQ	A19B5
NOCAN	C03F4
NOCB1	A48E5
NOCCHIO	A52A5
NOCHSW	B08F3
NOCLEAR	A16D1
NOCL44	A56F2
NOCRT	A02G3
NOCRTCND	A02D4
NOCRTERP	A20E1
NOCRTERR	A21C2
NOCRTFCH	B12A4
NOCRTRAS	B82A5
NOCRTSIO	A85F1
NODIBUPD	A23A3
NOEOF	A23J4
NOFREER	B63E2
NOLOG	A29K3
NOLTA	B65D1
NOMAINT	A23C4
NOMUL1	A12A1
NOMUL2	A12D1
NOPDFLT1	A46F1
NOPDFLT2	B05H1
NOPDINTR	A18A2
NOPDSIO	A02B5
NOPDTRHK	B33F1
NOPGFREE	B09B4
NOPST	B76H3
NORCD	A28A2
NORCD1	A28D2
NOREQCAN	A22E3
NORESERR	A21C3
NORMPCK	B32G4
NORMPF	B33A5
NOSENSE	A20H2
NOSO2	A01H3
NOSSACT	B33B4
NOSSPF	B32D3
NOSVA	B91J5
NOSYSIN	A08A2
NOTBSY	A19E3
NOTBSY1	A19K3
NOTBSY2	A19C3
NOTFIX1	A65A1
NOTFULL	A20H1
NOTIC	A08A4
NOTINQU	B74A4
NOTJAT	A37G1
NOTODEX	A63B5
NOTRANS	B05D5
NOTRTA	A86B1
NOTRTA1	A18H4
NOTSRES	A08B4
NOTTAPE	A02C2
NOTTAPE2	A15G1
NOT3340	A28H4
NOVTAM	A05K1
NOWAITM	A78A3
NOWRAP	A49A2
NPAGE	B53A2
NPHOPC	B33A2
NRTASIO	A85B1
NXTENTR	B49B4
NXT11	A55C2
NXT2	A55A4
NXT3	A56C2
NXT4	A70C1
OCINTR	A33A2
OCINTR1	A33C3
OKDEACT	A56E1
OLTEPNO	A48B4
OLTRET1	A02J3
OLTRET2	A18J2
OLTRET3	A37A5
OPENGTE	B37H5
OURSIO	A25E2
PAGECHK	B10C2
PAGEFREE	B09G3
PAGEIO	B61B2
PAGEOUT	B73B1
PCKADR2	A69F2
PCKCOMP	B33E2
PCKSVC07	A57C1
PCROUT	A84A3
PDAGN	A45B3
PDBOCK	A45A4
PDBUFL	A39G2
PDCHAN	A41F3
PDCOMP	A42E5
PDDFLT	A45A3
PDECBP	A63D2
PDECBT	A63C1
PDECBW	A63E1
PDERXT3	A41H4
PDEXIT	A41F1
PDEXT1	A41G1
PDFLTHK2	B05C1
PDFSTM	A38F1
PDHERE	A40H2
PDIAGN	A38E2
PDIAGN1	A41D2
PDINTR	A37B4
PDINTRKH	A17G2
PDMPI	A37C5
PDMVCSW	A38H4
PDNF	A39H1
PDNOAS	A83G5
PDNOB	A83B5
PDNOBUS	A44G3
PDNONOP	A42A2
PDNOSS	A39H4
PDNOTZR	A40A3
PDPDEQ	A42D5
PDREJDC	A45B1
PDSCAW	A39D1
PDSERR	A44F2
PDSERR1	A44D2
PDSERR2	A44G4
PDSERR3	A44H4

Label	Location
PDSIORT	A84B1
PDSNO1	A42F3
PDSNO2	A44G1
PDSPK	A39F1
PDSS	A41A1
PDSWBP	A41C5
PDSWZ	A41J1
PDTERR5	A42E4
PDTIO	A38D3
PDTIOERR	A38F4
PDTIOK	A38J3
PDUNITCK	A44D1
PDUPDT	A39A1
PFAEXIT2	B46C1
PFEXIT	B39C4
PFFLIH	B32B3
PFGENENT	A36D5
PFIXEXIT	B50G3
PFIXIMD	B55F2
PFIXPAGE	B55D2
PFIXPGE	B55B2
PFREENTR	B34C2
PFREXIT	B50H3
PFRFND	B51D2
PFRLIST	B54A1
PFRPGE	B53C2
PFWAIT	A84A3
PGINEND	B76D3
PGINENDT	B76A4
PGINIT	B75F5
PG2X	B61C1
PG3X	B61D3
PHASEFND	B12E5
PHFND1	B07G3
PHFND2	B07A4
PHNFND	B04K3
PHNFND3	B05A1
PHNFND4	B05A2
PHSALD	B69F5
PHSFND	B70A3
PHSLOAD1	B70B3
PHSNCOMP	B69K3
PMR	B36A1
PMRA	B37E1
PMRB	B36H5
PMRC	B37G2
PMRINIT	B37J4
PMR1	B36A2
PMR2	B36G2
PMR3	B36F2
PMR4	B36C3
PMR5	B36D3
PMR97	B36K3
PMR98	B36D4
PMR99	B36F3
POPGQU	B74H1
POSTENT	B45A3
POSTPUB	A23D5
POSTSTAT	A22H5

Label	Location
POSTTSK	B45D3
POTQ01	B74G5
POWCAN	C02A2
POWSIO	A02D5
PPTEST0	A23H3
PROCDIB	A50C4
PROCESS	A64A1
PROVALRM	A02H4
PRTPRG	A19J5
PRTPRG10	A87E4
PSTCEDE	A47H4
PSTECB	A15G4
PSTEOF	A23H4
PSTEOP	A23G4
PSTFULL	B75G4
PSTINVPA	B75J4
PSTINVPG	B76D2
PSTNEG	B77C5
PSTRL	B75F2
PSTTECB	A07E4
PTFREE	A54H3
PUBFCOND	A24E3
PUBLOOP	D01D3
PUBSCAN	A18F3
PUNCHDIB	A50B3
PURGE	A15A3
PURWAITL	A69B4
PWTASKS	B43A4
QUFCH	B47A1
QUIESCE	D01D4
QUISIO	A05F1
QUISTAPE	A05C4
QULOOP	B74J1
RASCANCL	B89F2
RASDEQ	B87B1
RASDEQUE	B91A2
RASDEQ0	B87A3
RASDEQ1	B87F3
RASEMGIO	B90A2
RASEMGI2	B90D3
RASFETCH	B90C1
RASFTCH	B80B4
RASFTCH1	B80A5
RASINITR	B91A3
RASIOE	B81A4
RASIOR	B90A4
RASIOREQ	B89K4
RASIOTST	B82A1
RASIPLR	B89C4
RASLUPR4	B88G3
RASSYSIO	B81F4
RASTART	B90A5
RASTIMER	B89G4
RASTREG	B89D3
RASTSCAN	B89B3
RASUPR	B80B3
RASWAITR	B91A1
RCVERR	A26B4
RCVTIO	A80D4

Label	Location
RC0	B50G2
RC0PFR	B52D2
RC12PFR	B53C3
RC4	B51D4
RC8	B51E3
RDYCALLS	A74G2
RDYCNL	A11G4
RDYCNLM	A11A4
RDYCNLS	A11B4
RDYCNL0	A11F4
RDYCNL2	A11J4
RDYCNL3	A11K4
RDYCNL5	A11H4
REACTC	B69A1
REACTP	B68A5
REACTPC	B68H4
REACTPUC	B68F3
REACTUC	B68A1
REACT1	B68C2
REACT1V	B68A4
REACT2	B68B4
REACT7UC	B68C1
READ	B93B1
READAR	A22F3
READAR1	A22G3
READD	B93A1
READHA	A82B3
READRLD	B09A5
READYHQ	A17C4
READYHQ1	A17D4
READYHQ2	A17F5
READYHQ3	A17E5
READYHQ4	A17F4
REALAD1	A66J5
RECALIB	A27J2
RECFINOK	A64D2
REISSGRR	B43A1
RELALL	B27A3
RELALL3	B27J3
RELALL4	B27A4
RELBLOCK	B26A5
RELCCB	B27A2
RELE	B74C3
RELOG	C03D2
RELPAGE	B73B2
RELTCB	B25A3
RELTCBNS	B25A4
RELTCB3	B25G3
REPCHECK	A19F5
REQFF	A15D5
REQFF1	A15K5
RESETHQ	A17B4
RESETPF	A19J2
RESET44	A55C3
RESFCH	B47A2
RESHADB	B42H2
RESPFIX	B56A2
RESPF1	B56A4
RESPF2	B56D4

Label	Location
RESPHO	B46A4
RESTART	B14H2
RESTFIX	B47A5
RESTFIX1	B47B5
RESTFIX2	B47A4
RESVC	A48F3
RESVCNOX	A05J4
RESVCX	A48G3
RESVC2	B03D2
RET	B73D2
RETADDR	B50G1
RETCODE	A66F2
RETC2	B73B4
RETC4	B74A2
RETC8	B74B5
RETDEACT	A56D1
RETRY01	A29E5
RETRY10	A28E3
RETURN	A04B3
RETYY	B70C1
RLDEXIT	B09J5
RMSD00	A27E5
RMSD10	A28D1
RMSGETP2	A68B4
RMSGNOHQ	A68E4
RMST02	A02J1
RMST10	A08A1
RMST15	A08B1
RMST20	A08C1
RMST35	A05A4
RNFBIT	B56E3
RRPHO	B46H4
RSETCCWT	B38J3
RSTRTPF	B55D1
RSTSV44	A55B2
RSZMSG	B02C2
RSZOK	B02A3
RTARETRN	B82G1
RTARETR2	B82G2
RTN3330	A31A2
RTN3340	A29A4
RTN3340N	A31A1
RWAIT	B34A1
SAMETASK	A03D3
SAVCONT	A74C5
SAVE	B48C4
SAVERID	B34E1
SCANDE1	B65F1
SCANDE2	B65G4
SCANDIR	B91K4
SCANDIR1	B12A3
SCANDIR2	B92C1
SCANDIR4	B92C2
SCANEQ	B11A3
SCANEQER	B11C4
SCANEQX	B11D4
SCANEQ1	B11A5
SCANEQ2	B11F5
SCANEQ3	B11J5

Label	Location
SCANFIN1	B78D2
SCANFIN2	B78D3
SCANLINK	B12A1
SCANLIST	B11A1
SCANL1	B12H1
SCANPCIL	B92B3
SCANPGT	B78A1
SCANSCIL	B91A5
SCURTKAD	B94J3
SCVRTBAL	B94C1
SCVRTERR	B94H4
SCVRTNOK	B94H5
SCVRTOVH	B94E5
SCVRTTOT	B95A2
SCVRTVLR	B94B4
SCVRT00	B94B2
SCVTRTN	B94B3
SEARCHL	B77C1
SECTCAL	B94B1
SEIZRTRN	A03F3
SEKCHK	A27G2
SEKCHK1	A27H2
SETADDR	B06B3
SETADDR1	B06C3
SETBACK	B28A4
SETCCW	B06H1
SETEND	B13A3
SETFBND	B35D2
SETFCH	B14A1
SETFLAG	B20A2
SETFLAG1	B20D1
SETFLAG2	B20A4
SETFLAG3	B20D4
SETLT	A09G5
SETNFFB	B55F3
SETPFA	A67B4
SETPMGR	B09E1
SETPTENT	B49C1
SETPTO	B09F3
SETREL	B08A5
SETRLD	B08J4
SETSLD	B01A3
SETTIC	B13F2
SETUPSA	B34C1
SHIFTFI	A13D4
SHIFTMVC	A13A4
SIO	A02H3
SKIPVAL	B75F3
SKPPST	A15J3
SKPVALID	A59E5
SLDCOMP	B01G4
SLDDONE	B01H3
SLDLOOP	B01F3
SLDLOOP1	B01A4
SO	A22D5
SPHOID	B46F5
SRETADR	B38D4
SRTABB	B63A3
SSERROR	B32C2
ST$$DET	A12A4
STARTPFR	B52A2
STCLOCK	A63J4
STCMMIC1	A39F3
STCMMIC2	A40F2
STCMMIC3	A40D3
STORADD	A63G4
STOREPC	A59G5
STPADR	B42G1
STPFIXRS	B49A4
STRTIO	A02A3
STRTIO1	A02D2
STRTIO2	A02B3
STRTPFIX	B54A2
SUBMIDHI	A64B1
SUBSVC06	A74E2
SUBTASK	A77A2
SUPADDR	B08C3
SUPRET	C02E1
SUPRET1	C02G1
SVCINDEX	A46J1
SVC00	A48B2
SVC01	A18F1
SVC01A	A18H1
SVC02	A09A5
SVC02A	A09D5
SVC03	A53B2
SVC04	A35B4
SVC04RET	A35C4
SVC05	A53B3
SVC07	A57B1
SVC08	A57B2
SVC09	A57B3
SVC10	A57B4
SVC11	A58B4
SVC12	A59B1
SVC12A	A59C2
SVC13	A59B3
SVC13A	A87A5
SVC15	A51B2
SVC16	A59B5
SVC17	A60B1
SVC18	A60B3
SVC19	A60B4
SVC2WAIT	A05G4
SVC20	A60B5
SVC21	A60H1
SVC22	A61B1
SVC22A	A61G1
SVC22B	A61G2
SVC23	A61B3
SVC24	A61B4
SVC24A	A57C4
SVC24C	A57F4
SVC24E	A58B3
SVC24G	A58H1
SVC24H	A61B5
SVC24K	A58D2
SVC24L	A58G1
SVC25	A51A3
SVC26	A62B4
SVC27	A51B3
SVC28	A83B3
SVC28ERR	A83D4
SVC29	A63B1
SVC33	A63C3
SVC34	A63B4
SVC34PP	A63F4
SVC35A	A71A2
SVC35C	A71A4
SVC35D	A71A3
SVC35E	A71B2
SVC3536A	A73C2
SVC3536C	A72K4
SVC3536D	A73D3
SVC36A	A72A2
SVC36AP	A72A1
SVC36C	A72C2
SVC36D	A72E3
SVC36E	A72F3
SVC37	A60H2
SVC38A	A75A5
SVC38B	A75D3
SVC38C	A75H3
SVC38D	A76F1
SVC38E	A76C2
SVC38F	A75B4
SVC39	A12A2
SVC39A	A13E2
SVC39B	A13K2
SVC39C	A13A3
SVC39E	A13G1
SVC39MVC	A13B3
SVC41A	A79F3
SVC42A	A79C2
SVC42B	A79F2
SVC44	A64B2
SVC45	A65B2
SVC46	A65B4
SVC48	A65G1
SVC48A	A65J1
SVC51	A65G2
SVC52	A62B1
SVC52A	A62E2
SVC52B	A62D2
SVC52C	A62D1
SVC52E	A62F1
SVC52F	A62H3
SVC52G	A62H2
SVC52H	A62G3
SVC54	A65G3
SVC55	A65G5
SVC58	A66B2
SVC59	A66B3
SVC60	A70B1
SVC61	A66B4
SVC90	A67E3
SVC91	A67E4
SVC62	A66B5
SVC63	A81B1
SVC63C	A81D1
SVC63D	A81J1
SVC63E	A81A3
SVC63F	A81F2
SVC63G	A81J2
SVC63H	A81C3
SVC6364R	A82B1
SVC6364V	A81B4
SVC6364W	A81H4
SVC6364X	A81A5
SVC64A	A13J5
SVC64AA	A13G5
SVC64ALL	A13E5
SVC64C	A14B1
SVC64CTD	A82A2
SVC64D	A14C1
SVC64DA	A14F1
SVC64DB	A14G1
SVC64E	A14A2
SVC64F	A14E2
SVC64G	A14F2
SVC64H	A14F3
SVC65	A66E1
SVC66	A66E2
SVC67	A66E3
SVC68	A66E4
SVC69	A66E5
SVC70	A67B1
SVC71	A67B3
SVC72	A67B5
SVC73	A68B1
SVC74	A68B2
SVC76	A51A2
SVC77	A68B3
SVC89	A57C2
SVDEACT	A56B1
SVFREAL	B43A2
SVFRSTRG	A16E3
SVGREAL	B40A3
SVPFREE	B51A5
SVPGIN	B75A3
SVRETURN	B39A2
SVSPF	B35B4
SWAP	A09G5
SYSEXIT	A05A1
SYSFCH	B04A2
SYSFILE	A09A1
SYSINF	A09J4
SYSPROC	A05E2
SYSRTRN	A06A1
SYSSAVE0	A36J3
SYSSAVE1	A36K3
SYSSCAN	B04E4
TASKCLI	A12D4
TASKSCAN	A77C4
TCH	A01D4
TENCHECK	A28F3

Label	Location	Label	Location
TERMFIN	B78J4	UNFIX1	B24E2
TERMPGIN	B78A4	UNFIX2	B24J2
TEST	C01J1	UNITCHK	A22A1
TESTCE	A18G2	UNWAIT	A77A3
TESTCONT	C01H1	UNWAIT0	A78A2
TESTEND	B13A1	UNWTDEQ	A77C3
TESTERR	C03E3	UPADD	A48H3
TESTHQ	A17B3	UPDEOPQ	B34F5
TESTIDAL	B21A1	UPDMIDN	A64C1
TESTLOG	C03F2	USER	A25B3
TESTNE	B38B1	USERPCK	B33A3
TESTNFF	B45E1	VALID	A69B2
TESTPFR	B42D1	VALIDATE	B77A3
TEXTREAD	B09C2	VALIDE	B76C1
TFCNTZ	B38C4	VALIDNXT	A69H2
TFIX	B44A1	VALIDOK	A69E2
TFIXREQ	B06A3	VALIDR2	A69C2
TFREE	B44A3	VALIDTR	B05F4
TFREEPF3	B62F2	VALIDTST	A69K2
TFREEPF4	B62G3	VALLIST	B77A1
TFREEPF6	B62H1	VALPADR	B50K1
TFX44	A64A4	VIRTAD	A67A1
TF1ONLY	A56G3	VIRTAD1	A67H2
TF44R1	A56H3	VIRTAD2	A67J1
TF44R2	A56J2	VIRTJOB	B48C2
THRET	A48D4	XTRESUME	A07D3
TONFBTS	B51J2	YESSAVE	A76G4
TPBUSY0	A52A3	YESSAV1	A76H4
TPBUSY1	A52J3	YESSAV2	A76K5
TPBUSY2	A52E5		
TPBUSY3	A52H5		
TPBUSY4	A52C3		
TRANS	B05G4		
TRANSCCW	B06A2		
TRKCHK	A28A4		
TRKEOC	A27F4		
TRKNO	A28K4		
TRNOFF	A01A3		
TRTABB	B63A4		
TRTABB1	B63E4		
TSTATTN	A22B1		
TSTDEV1	A51C4		
TSTDEV2	A51E3		
TSTECB	A77H4		
TSTENTRY	B11A4		
TSTEOJ	A23K3		
TSTEOP	A23F4		
TSTERF	A15A1		
TSTERQ	A64F4		
TSTLEN	B54B4		
TSTSPB	B51C1		
TSTSVC	B07G4		
TSTTFC	B55E3		
TSTWCN	B56C3		
TTRTFIX	B35A1		
TWOKBDRY	B06K3		
T35405	A09C2		
UNFIX	B24A3		
UNFIX0	B24A2		

PIK (Partition Identification Key)

The halfword PIK at displacement 46 in each partition's communication region has a zero value in the high-order byte and a key value in the low-order byte.

In a foregound communications region, the key value in the PIK is insignificant.

The PIK in the background communications region (BGCOMREG) has the same value as the PIK of the active partition, or X'00' when the Attention Task is active. Depending on the number of partitions supported, the PIK in the BGCOMREG may have any one of the following values:

Active Task or Partition	PIK Value in BGCOMREG NPARTS=5	NPARTS=4	NPARTS=3	NPARTS=2	NPARTS=1
Attention	X'00'	X'00'	X'00'	X'00'	X'00'
BG	X'10'	X'10'	X'10'	X'10'	X'10'
F4	X'20'				
F3	X'30'	X'20'			
F2	X'40'	X'30'	X'20'		
F1	X'50'	X'40'	X'30'	X'20'	

The PIK value also equals the storage protection key multiplied by 16.

TIK (Task Identification Key)

The halfword TIK at displacement 90 in the SCP Communications Region (SYSCOM) has a zero value in the high-order byte and a key value in the low-order byte. This key value is only significant if AP is supported.

The key value in the TIK is the key of the program (task or subtask) that is being serviced. When an interruption occurs, the value of the TIK indicates to the supervisor which program (task or subtask) was interrupted.

The TIK is set by Task Selection in the General Exit Routine and equals the index displacement of the task's Program Information Block (PIB) within the PIB Table.

Depending on the number of partitions supported, the value of the TIK indicates which task was interrupted according to the following table:

NPARTS=2	TIK Value NPARTS=3	NPARTS=4	NPARTS=5	Interrupted Task
X'00'	X'00'	X'00'	X'00'	Attention
X'10'	X'10'	X'10'	X'10'	BG
			X'20'	F4
		X'20'	X'30'	F3
	X'20'	X'30'	X'40'	F2
X'20'	X'30'	X'40'	X'50'	F1
X'30'	X'40'	X'50'	X'60'	Subtasks*
-	-	-	-	
-	-	-	-	
-	-	-	-	
X'F0'	X'F0'	X'F0'	X'F0'	

*Asynchronous Processing option. The number of PIBs initially available for subtasks is 10, 11, 12, or 13, depending on the number of partitions (in an AP supervisor the total number of PIBs is always sixteen).

LIK (Logical Transient Owner Identification Key)

The halfword LIK at displacement 88 in the SCP Communications Region (SYSCOM) is only significant if AP is supported and contains the same value as the TIK when the logical Transient Area (LTA) is in use and therefore identifies the owner of the LTA. When the LTA is free, the halfword LIK contains zeros. The SVC 2 routine sets the LIK, and the SVC 11 routine resets it to zero. If AP is not supported, the LIK contains zeros.

LTK (Logical Transient Key)

The halfword LTK at displacement 110 in each partition's communications region has a zero value in the high-order byte and a key value in the low-order byte.

In a foreground communications region, the key value in the LTK is not significant.

The LTK in the background communications region (BGCOMREG) has the same value as the PIK of the partition of the task that owns the LTA, or contains zeros when the LTA is free. When the LTA is occupied by a task, therefore, the BGCOMREG has the same value in its LTK as in its PIK when the owning task is active.

The SVC 2 routine sets the LTK, and the SVC 11 routine resets it to zero.

REQID (I/O Requestor's Partition or System Task ID)

The REQID is a one-byte identifier in the Channel Queue (CHANQ) entry. (See Figure 6.2.)

When a background or foreground program has requested I/O, the REQID has the same value as the key byte of the PIK for that task's partition. When the Attention Task has requested I/O, the REQID contains X'00'.

When the request for I/O is from a System Task, the REQID has one of the following values:

RAS - X'01'
PMGR - X'02'
SUPVR - X'03'
CRT - X'04'
ERP - X'05'
PAGEIN - X'06'

The REQID is set by the Channel Scheduler Routine.

TKREQID (I/O Requestor's Task Identification)

The TKREQID is a one-byte identifier in the Channel Queue (CHANQ) entry for a task that has requested I/O (see Figure 6.2). In an unused CHANQ entry the TKREQID contains X'FF'.

The TKREQID byte in an active CHANQ entry has the same value as the key byte of the TIK of the task that has requested I/O. If AP is not supported, it has the same value as the PIK of the task that has requested I/O.

TKREQID is set by the Channel Scheduler Routine and reset by the I/O Interrupt Handler.

Card Code	Actual IBM Device	Device-Type X'nn'	Device Type
2400T9	9-track Magnetic Tape units	50	Magnetic Tape
2400T7	7-track Magnetic Tape units	50	
3410T9	9-track 3410 Magnetic Tape units	53	
3410T7	7-track 3410 Magnetic Tape units	53	
3420T9	9-track 3420 Magnetic Tape units	52	
3420T7	7-track 3420 Magnetic Tape units	52	
2495TC	2495 Tape Cartridge Reader	51	Tape Cartridge Reader
1442N1	1442N1 Card Read Punch	30	Card Read Punches
2520B1	2520B1 Card Read Punch	31	
2560	2560 Multifunction Card Machine		
2596	2596 Card Read Punch	30	
3525RP	3525 Card Punch (with optional read feature)	32	
5425	Multifunction Card Unit	34	
2501	2501 Card Reader	10	Card Readers
2540R	2540 Card Reader	11	
3504	3504 Card Reader	12	
3505	3505 Card Reader	12	
2540P	2540 Card Punch	21	Card Punches
2520B2	2520B2 Card Punch	20	
1442N2	1442N2 Card Punch	22	
2520B3	2520B3 Card Punch	20	
3525P	3525 Card Punch	23	
1403	1403 Printer	40	Printers
1403U	1403 Printer with UCS feature	42	
1443	1443 Printer	41	
2260 (local)	1053 Printer with 2548 Control Unit. MODE operand must be entered as X'01'	C0	
3203	3203 Printer	4A	
3211	3211 Printer	43	
3277 (local 3270)	3284 or 3286 Printer with 3272 Control Unit. MODE operand must be entered as X'01'	B0	
3277B (local 3270)	3284 or 3286 Printer with 3272 Control Unit, attached in burst mode to a multiplexer channel. MODE operand must be entered as X'01'	B0	
5203	5203 Printer	4C	
5203U	5203 Printer with UCS feature	4D	
1050A	3210, 3215 Console Printer Keyboards	00	Printer Keyboards
125D	Model 115/125 Integrated Display Operator Console	B2	Display Operator Consoles
125DP	Model 115/125 Integrated Display Operator Console with 5213 Console Printer attached	B2	

Figure 10.1. Device Type Codes (Part 1 of 2)

Card Code	Actual IBM Device	Device-Type X'nn'	Device Type
UNSP	Unsupported device	FF	Unsupported no burst mode on multiplexer channel
UNSPB	Unsupported device	FF	Unsupported with burst mode on multiplexer channel
2311	2311 Disk storage device	60	DASD
2314	2314 Direct-access storage facility	62	
2314	2319 Disk storage facility	62	
2321	2321 Data cell drive	61	
3330	3330-1, 3330-2, or 3333-1 Disk storage	63	
3340	3340 Disk Storage (General)	68	
3340	3340 Disk Storage with 3348 Model 35	69	
3340	3340 Disk Storage with 3348 Model 70	6A	
1419	1255 Magnetic Character Reader	72	MICR-Magnetic Ink Character Recognition devices
1419	1259 Magnetic Character Reader	72	
1419	1419 Magnetic Character Reader	72	
1419P	1419 Dual Address Adapter Primary Control Unit	73	
1419S	1419 Dual Address Adapter Secondary Control Unit	74	
2701	2701/2715 Data Adapter Unit	D0	Teleprocessing lines A=SAD0 command when B=SAD1 command enabling C=SAD2 command the line D=SAD3 command
2702 {A, B, C, D}	2702 Transmission Control Unit	D1	
2703	2703 Transmission Control Unit	D2	
2703	Integrated Communication Adapter (Models 125/135)	D2	
2703	3704/3705 Communication Controller in Emulation Mode	D2	
2955	2955 Data Adapter Unit	D7	Data Link for RETAIN
1017	1017 Paper Tape Reader with 2826 Control Unit	78	Paper Tape Readers
2671	2671 Paper Tape Reader	70	
1018	1018 Paper Tape Punch with 2826 Control Unit	79	Paper Tape Punch
1419	1270 Optical Reader/Sorter	79	Optical Readers
1419P	1275 Optical Reader/Sorter	73	
1287	1287 Optical Reader	77	
1288	1288 Optical Page Reader	77	
3881	3881 Optical Mark Reader	11	
3886	3886 Optical Character Reader	7C	
3540	3540 Diskette Input/Output Unit	80	Diskette
2260	2260 Display Station	C0	Display Station
3277 (local3270)	3277 Display Station; MODE operand need not be entered	B0	
3277B (local 3270)	3277 Display Station; attached in burst mode to a multiplexer channel. MODE operand need not be entered	B0	
7770	7770 Audio Response Unit	D3	Audio Response Unit

Figure 10.1. Device Type Codes (Part 2 of 2)

Density (Bytes per inch)	Parity	Convert Feature	Translate	SS Code*	
200	odd	on	off	10	
200	odd	off	cff	30	
200	odd	off	on	38	
200	even	off	cff	20	
200	even	off	on	28	
556	odd	on	off	50	
556	odd	off	off	70	
556	odd	off	on	78	7-track
556	even	off	off	60	
556	even	off	on	68	
800	odd	on	off	90	
800	odd	off	cff	B0	
800	odd	off	on	B8	
800	even	off	cff	A0	
800	even	off	on	A8	
800	dual density			C8	
1600	dual density			C0	
6250	dual density			D0	9-track
800	single density			C0	
1600	single density			C0	
6250	single density			D0	

* Refer to PUB Table, byte 5.

Figure 10.2. Tape Density Data

Cancel Code (Hex)	Message Code	Descriptive Part of Message (or Condition)
10	----	Normal EOJ
11	0V07I	No channel program translation unsupported device
12	0V06I	Insufficient buffer space for channel program translation
13	0V05I	CCW with count greater than 32K
14	0V04I	Page pool too small
15	0V02I	Page fault in disabled program
16	0V01I	Page fault in MICR stacker select or PHO routine
17	0S02I	(Same as 23 but causes dump because subtasks were attached when maintask issued CANCEL macro)
18	----	(Eliminates cancel message when maintask issues DUMP macro with subtasks attached)
19	0P74I	I/O Operator Option
1A	0P73I	I/O Error
1B	0P82I	Channel Failure
1C	0S14I	CANCEL ALL Macro
1D	0S12I	Maintask Termination
1E	0S13I	Unknown ENQ Requestor
1F	0P81I	CPU Failure
20	0S03I or 0S11I	Program Check
21	0S04I or 0S09I	Illegal SVC
22	0S05I or 0S06I	Phase Not Found
23	0S02I	Program Request
24	0S01I	Operator Intervention
25	0P77I	Invalid address
26*	0P71I	SYSxxx Not Assigned (unassigned LUB code)
27	0P70I	Undefined Logical Unit (invalid LUB code in CCB)
28	----	QTAM cancel in progress
29	0S15I	No relocating loader support (Fetch or load request for relocatable phase while supervisor does not support relocating load).
2A	0V10I	I/O error on page data set
2B	0P84I	I/O error during fetch from private core image library
2C	0V09I	Illegal parameter passed by PHO routine
2D	0P881	Program cannot be executed/restarted due to failing storage block
2E	0S16I	Invalid resource request (possible deadlock)
2F	0V03I	More than 255 PFIX requests for 1 page

Figure 10.3. Cancel Codes and Messages (Part 1 of 2)

Cancel Code (Hex)	Message Code	Descriptive Part of Message (or Condition)
30	0P72I	Reading Past /& Statement (on SYSRDR or SYSIPT)
31	0P75I	Error Queue Overflow
32	0P76I	Invalid DASD Address
33	0P79I	No Long Seek (disk)
34		Reserved
35	0P85I	Job control open failure
36	0V08I	Page fault in I/O appendage routine
37		Reserved
38	0V11I	Wrong privately translated CCW
39		Reserved
40	4T98I	VTAM termination
41	4T99I	VTAM termination
FF	0P78I	Unrecognized Cancel Code
	0P83A**	Supervisor Catalog Failure
	0P87A**	IPL failure

In addition to recognizing the cancel-codes above, the Terminator also recognizes the same codes with the X'80' bit on (cancel occurred in LTA). The X'80' bit is set by $$ABERZ1 when no more I/O requests are pending and the cancel procedure therefore can be continued. Later it is tested for by $$BEOJ and subsequently reset.

*If the CCB is unavailable, the logical unit is SYSxxx.

**This cancel code is not significant in case of a supervisor catalog or IPL failure, because the system is placed in a wait state without any further processing by the Terminator.

Figure 10.3. Cancel Codes and Messages (Part 2 of 2)

Message	Phase	Chart
0P60D	$$ABERRZ (Physical Attention)	C01
0P89I	$$ABERRZ (Physical Attention)	C01
1I32D	$$ABERRZ (Physical Attention)	C02
1I40D	$$ABERRZ (Physical Attention)	C02
1S01D	$$ABERRZ (Physical Attention)	C03
1S02D	$$ABERRZ (Physical Attention)	C03
1Q62I	$$ABERRZ (Physical Attention)	C02

All messages issued by the DOS/VS system control programs are listed in this appendix with a reference to the PLM and issuing phase. For cause and action of each message, see DOS/VS, GC33-5379.

Message	Phase	PLM
0D01A	$$BOCRTV	SY33-8553
0D02I	$$BOCRTM	SY33-8553
0D03A	$$BOCRTX	SY33-8553
0D04A	$$BOCRTV	SY33-8553
0D05A	$$BOCRTW	SY33-8553
0D05A	$$BOCRT1	SY33-8553
0D06A	$$BOCRTY	SY33-8553
0D07D	$$BOCRTS	SY33-8553
0D08A	$$BOCRTS	SY33-8553
0D09D	$$BOCRTR	SY33-8553
0D20E	$$BOCRTB	SY33-8553
0D25E	$$BOCRTB	SY33-8553
0D26A	$$BOCRTG	SY33-8553
0D26E	$$BOCRTB	SY33-8553
0D29E	$$BOCRTB	SY33-8553
0D30A	$$BOCRTV	SY33-8553
0D30A	$$BOCRT1	SY33-8553
0D33A	$$BOCRTE	SY33-8553
0D34E	$$BOCRTF	SY33-8553
0D35E	$$BOCRTF	SY33-8553
0D36E	$$BOCRTF	SY33-8553
0D97A	$$BOCRT1	SY33-8553
0I00A	$$A$IPL2	SY33-8555
0I00I	$IPLRT2	SY33-8555
0I01A	$$A$IPL2	SY33-8555
0I02A	$$A$IPL2	SY33-8555
0I03A	$$A$IPL2	SY33-8555
0I04I	$$A$IPL2	SY33-8555
0I10A	$IPLRT2	SY33-8555
0I11A	$IPLRT2	SY33-8555
0I12A	$IPLRT3	SY33-8555
0I13A	$IPLRT3	SY33-8555
0I14A	$IPLRT3	SY33-8555
0I15A	$IPLRT3	SY33-8555
0I16A	$IPLRT4	SY33-8555
0I17A	$IPLRT4	SY33-8555
0I18A	$IPLRT2	SY33-8555
0I19I	$IPLRT4	SY33-8555
0I20I	$IPLRT5	SY33-8555
0I23A	$IPLRT4	SY33-8555
0I24A	$IPLRT3	SY33-8555
0I25I	$IPLRT4	SY33-8555
0I26I	$$BUFLDR	SY33-8553
0I26I	$IPLRT4	SY33-8555
0I27I	$$BUFLD2	SY33-8553
0I27I	$IPLRT4	SY33-8555
0I28D	$$BUFLD2	SY33-8553
0I28D	$IPLRT4	SY33-8555
0I29I	$IPLRT4	SY33-8555
0I30I	$IPLRT2	SY33-8555
0I31A	$IPLRT2	SY33-8555
0I32I	$IPLRT2	SY33-8555
0I33A	$IPLRT4	SY33-8555
0I34D	$IPLRT4	SY33-8555
0I35I	$IPLRT4	SY33-8555

Message	Phase	PLM
0I37A	$IPLRT5	SY33-8555
0I38A	$IPLRT5	SY33-8555
0I39A	$IPLRT5	SY33-8555
0I40A	$IPLRT5	SY33-8555
0I41A	$IPLRT5	SY33-8555
0I42A	$IPLRT5	SY33-8555
0I43A	$IPLRT5	SY33-8555
0I44A	$IPLRT5	SY33-8555
0I45A	$IPLRT5	SY33-8555
0I46A	$IPLRT5	SY33-8555
0I47I	$IPLRT2	SY33-8555
0I48I	$IPLRT2	SY33-8555
0I49A	$IPLRT5	SY33-8555
0I50A	$IPLRT2	SY33-8555
0I51A	$IPLRT5	SY33-8555
0I52I	$IPLRT5	SY33-8555
0I60A	$IPLRT3	SY33-8555
0I61I	$IPLRT4	SY33-8555
0P08	$$ABERAE	SY33-8552
0P08	$$ABERAH	SY33-8552
0P08	$$ABERAN	SY33-8552
0P08	$$ABERRB	SY33-8552
0P08	$$ABERRF	SY33-8552
0P08	$$ABERRG	SY33-8552
0P08	$$ABERRH	SY33-8552
0P08	$$ABERRI	SY33-8552
0P08	$$ABERRS	SY33-8552
0P08	$$ABERRT	SY33-8552
0P08	$$ABERRU	SY33-8552
0P08	$$ABERRV	SY33-8552
0P08	$$ABERRW	SY33-8552
0P08	$$ABERRY	SY33-8552
0P08	$$ABERR1	SY33-8552
0P08	$$ABERR7	SY33-8552
0P09	$$ABERAE	SY33-8552
0P09	$$ABERAH	SY33-8552
0P09	$$ABERAN	SY33-8552
0P09	$$ABERRB	SY33-8552
0P09	$$ABERRF	SY33-8552
0P09	$$ABERRG	SY33-8552
0P09	$$ABERRH	SY33-8552
0P09	$$ABERRI	SY33-8552
0P09	$$ABERRS	SY33-8552
0P09	$$ABERRT	SY33-8552
0P09	$$ABERRU	SY33-8552
0P09	$$ABERRV	SY33-8552
0P09	$$ABERRW	SY33-8552
0P09	$$ABERRY	SY33-8552
0P09	$$ABERR1	SY33-8552
0P09	$$ABERR7	SY33-8552
0P10	$$ABERAC	SY33-8552
0P10	$$ABERAE	SY33-8552
0P10	$$ABERAF	SY33-8552
0P10	$$ABERAH	SY33-8552
0P10	$$ABERAN	SY33-8552

Message	Phase	PLM
0P10	$$ABERRB	SY33-8552
0P10	$$ABERRF	SY33-8552
0P10	$$ABERRG	SY33-8552
0P10	$$ABERRH	SY33-8552
0P10	$$ABERRI	SY33-8552
0P10	$$ABERRU	SY33-8552
0P10	$$ABERRV	SY33-8552
0P10	$$ABERRW	SY33-8552
0P10	$$ABERRY	SY33-8552
0P10	$$ABERR1	SY33-8552
0P10	$$ABERR7	SY33-8552
0P11	$$ABERAB	SY33-8552
0P11	$$ABERAC	SY33-8552
0P11	$$ABERAF	SY33-8552
0P11	$$ABERAH	SY33-8552
0P11	$$ABERAI	SY33-8552
0P11	$$ABERRB	SY33-8552
0P11	$$ABERRF	SY33-8552
0P11	$$ABERRG	SY33-8552
0P11	$$ABERRH	SY33-8552
0P11	$$ABERRI	SY33-8552
0P11	$$ABERRU	SY33-8552
0P11	$$ABERRV	SY33-8552
0P11	$$ABERRY	SY33-8552
0P11	$$ABERR5	SY33-8552
0P11	$$ABERR7	SY33-8552
0P12	$$ABERRB	SY33-8552
0P12	$$ABERR5	SY33-8552
0P13	$$ABERRB	SY33-8552
0P13	$$ABERR5	SY33-8552
0P13A	$$RAST10	SY33-8552
0P14	$$ABERAB	SY33-8552
0P14	$$ABERAH	SY33-8552
0P14	$$ABERRB	SY33-8552
0P14	$$ABERRT	SY33-8552
0P14	$$ABERRY	SY33-8552
0P14	$$ABERR1	SY33-8552
0P15	$$ABERRB	SY33-8552
0P15	$$ABERR4	SY33-8552
0P16	$$ABERRB	SY33-8552
0P16	$$ABERR5	SY33-8552
0P17	$$ABERAB	SY33-8552
0P17	$$ABERAH	SY33-8552
0P17	$$ABERRB	SY33-8552
0P17	$$ABERR1	SY33-8552
0P18	$$ABERAB	SY33-8552
0P18	$$ABERAH	SY33-8552
0P18	$$ABERAN	SY33-8552
0P18	$$ABERRB	SY33-8552
0P18	$$ABERRF	SY33-8552
0P18	$$ABERRG	SY33-8552
0P18	$$ABERRH	SY33-8552
0P18	$$ABERRI	SY33-8552
0P18	$$ABERRS	SY33-8552
0P18	$$ABERRT	SY33-8552
0P18	$$ABERRU	SY33-8552
0P18	$$ABERRV	SY33-8552
0P18	$$ABERRW	SY33-8552
0P18	$$ABERRY	SY33-8552
0P18	$$ABERR1	SY33-8552
0P18	$$ABERR7	SY33-8552
0P19	$$ABERAB	SY33-8552
0P19	$$ABERAH	SY33-8552
0P19	$$ABERAN	SY33-8552
0P19	$$ABERRB	SY33-8552
0P19	$$ABERRF	SY33-8552
0P19	$$ABERRG	SY33-8552
0P19	$$ABERRH	SY33-8552
0P19	$$ABERRI	SY33-8552
0P19	$$ABERRS	SY33-8552
0P19	$$ABERRT	SY33-8552
0P19	$$ABERRU	SY33-8552
0P19	$$ABERRV	SY33-8552
0P19	$$ABERRW	SY33-8552
0P19	$$ABERRY	SY33-8552
0P19	$$ABERR1	SY33-8552
0P19	$$ABERR7	SY33-8552
0P20	$$A$SUP1	SY33-8552
0P20	$$ABERAC	SY33-8552
0P20	$$ABERAF	SY33-8552
0P20	$$ABERAG	SY33-8552
0P20	$$ABERRD	SY33-8552
0P20	$$ABERRF	SY33-8552
0P20	$$ABERRI	SY33-8552
0P20	$$ABERRU	SY33-8552
0P20	$$ABERR2	SY33-8552
0P20	$$ABERR4	SY33-8552
0P21	$$ABERRB	SY33-8552
0P21	$$ABERR4	SY33-8552
0P22	$$ABERR1	SY33-8552
0P22	$$ABERR4	SY33-8552
0P23	$$ABERR4	SY33-8552
0P24	$$ABERAB	SY33-8552
0P24	$$ABERAH	SY33-8552
0P24	$$ABERRA	SY33-8552
0P24	$$ABERRI	SY33-8552
0P25	$$ABERAB	SY33-8552
0P25	$$ABERAH	SY33-8552
0P25	$$ABERRA	SY33-8552
0P25	$$ABERRI	SY33-8552
0P26	$$ABERRB	SY33-8552
0P26	$$ABERR1	SY33-8552
0P27	$$ABERRA	SY33-8552
0P27	$$ABERRY	SY33-8552
0P28	$$ABERAB	SY33-8552
0P28	$$ABERAH	SY33-8552
0P28	$$ABERAN	SY33-8552
0P28	$$ABERRB	SY33-8552
0P28	$$ABERRF	SY33-8552
0P28	$$ABERRG	SY33-8552
0P28	$$ABERRH	SY33-8552
0P28	$$ABERRI	SY33-8552
0P28	$$ABERRS	SY33-8552
0P28	$$ABERRT	SY33-8552
0P28	$$ABERRU	SY33-8552
0P28	$$ABERRV	SY33-8552
0P28	$$ABERRW	SY33-8552
0P28	$$ABERRY	SY33-8552
0P28	$$ABERR1	SY33-8552
0P28	$$ABERR7	SY33-8552
0P29	$$ABERAB	SY33-8552
0P29	$$ABERAH	SY33-8552
0P30	$$ABERAB	SY33-8552
0P30	$$ABERAH	SY33-8552
0P31	$$ABERAB	SY33-8552
0P31	$$ABERAH	SY33-8552
0P31	$$ABERRA	SY33-8552
0P31	$$ABERRI	SY33-8552
0P32	$$ABERAB	SY33-8552
0P32	$$ABERAH	SY33-8552
0P33	$$ABERRF	SY33-8552
0P33	$$ABERRY	SY33-8552
0P34	$$ABERRS	SY33-8552
0P35	$$ABERAN	SY33-8552

Message	Phase	PLM
0P35	$$ABERRT	SY33-8552
0P36	$$ABERRE	SY33-8552
0P36	$$ABERR4	SY33-8552
0P37	$$ABERRS	SY33-8552
0P38	$$ABERRT	SY33-8552
0P39	$$ABERRI	SY33-8552
0P40	$$ABERRU	SY33-8552
0P41	$$ABERRF	SY33-8552
0P41	$$ABERRH	SY33-8552
0P42	$$ABERAE	SY33-8552
0P42	$$ABERAG	SY33-8552
0P42	$$ABERAH	SY33-8552
0P43	$$ABERAG	SY33-8552
0P44	$$ABERAC	SY33-8552
0P44	$$ABERAH	SY33-8552
0P45	$$ABERRB	SY33-8552
0P46	$$ABERAG	SY33-8552
0P47	$$ABERAE	SY33-8552
0P47	$$ABERAH	SY33-8552
0P48	$$ABERRG	SY33-8552
0P49	$$ABERRG	SY33-8552
0P49	$$ABERR7	SY33-8552
0P50	$$ABERRY	SY33-8552
0P51	$$ABERAN	SY33-8552
0P52	$$ABERAN	SY33-8552
0P53	$$ABERAN	SY33-8552
0P54	$$ABERAN	SY33-8552
0P55	$$ABERR7	SY33-8552
0P60D	$$ABERRZ	SY33-8551
0P70I	$$BEOJ2	SY33-8553
0P71I	$$BEOJ2	SY33-8553
0P72I	$$BEOJ2	SY33-8553
0P73I	$$BEOJ2A	SY33-8553
0P74I	$$BEOJ2A	SY33-8553
0P75I	$$BEOJ2	SY33-8553
0P76I	$$BEOJ2	SY33-8553
0P77I	$$BEOJ2	SY33-8553
0P78I	$$BEOJS2	SY33-8553
0P79I	$$BEOJ2A	SY33-8553
0P81I	$$BEOJ2A	SY33-8553
0P82I	$$BEOJ2A	SY33-8553
0P83A	$$BEOJ2A	SY33-8553
0P84I	$$BEOJ2A	SY33-8553
0P85I	$$BEOJ2A	SY33-8553
0P87A	$$BEOJS2	SY33-8553
0P88I	$$BEOJS2	SY33-8553
0P89I	$$ABERRZ	SY33-8551
0R01I	$JOBCTLB	SY33-8555
0R02I	$JOBCTLB	SY33-8555
0R03I	$JOBCTLB	SY33-8555
0R17I	$JOBCTLB	SY33-8555
0R19I	$JOBCTLB	SY33-8555
0S00I	$$BILSVC	SY33-8553
0S00I	$$BPCHK	SY33-8553
0S01I	$$BEOJ2	SY33-8553

Message	Phase	PLM
0S02I	$$BEOJ2	SY33-8553
0S03I	$$BPCHK	SY33-8553
0S04I	$$BILSVC	SY33-8553
0S05I	$$BILSVC	SY33-8553
0S06I	$$BEOJ1	SY33-8553
0S07I	$$BPSW	SY33-8553
0S08I	$$BEOJ	SY33-8553
0S09I	$$BEOJ1	SY33-8553
0S11I	$$BEOJ1	SY33-8553
0S12I	$$BEOJ2A	SY33-8553
0S13I	$$BEOJ2A	SY33-8553
0S14I	$$BEOJ2A	SY33-8553
0S15I	$$BEOJ2A	SY33-8553
0S16I	$$BEOJ2A	SY33-8553
0S17I	$$BMVKEY	SY33-8553
0T00I	$$ABERA1	SY33-8552
0T00I	$$ABERA2	SY33-8552
0T00I	$$ABERA3	SY33-8552
0T00I	$$RAST08	SY33-8552
0T00I	$$RAST11	SY33-8552
0T03I	$$ABERA1	SY33-8552
0T03I	$$ABERA2	SY33-8552
0T03I	$$ABERA3	SY33-8552
0T03I	$$RAST08	SY33-8552
0T03I	$$RAST11	SY33-8552
0T05E	$$RAST08	SY33-8552
0T05E	$$RAST11	SY33-8552
0T05I	$$ABERA1	SY33-8552
0T05I	$$ABERA2	SY33-8552
0T05I	$$ABERA3	SY33-8552
0T06I	$$RAST03	SY33-8552
0T06I	$$RAST11	SY33-8552
0T07I	$$RAST03	SY33-8552
0T07I	$$RAST11	SY33-8552
0T08I	$$RAST03	SY33-8552
0T08I	$$RAST11	SY33-8552
0T09I	$$RAST03	SY33-8552
0T09I	$$RAST11	SY33-8552
0T10I	$$RAST02	SY33-8552
0T10I	$$RAST04	SY33-8552
0T10I	$$RAST05	SY33-8552
0T10I	$$RAST06	SY33-8552
0T10I	$$RAST07	SY33-8552
0T10I	$$RAST10	SY33-8552
0T10I	$$RAST12	SY33-8552
0T11W	$$RAST00	SY33-8552
0T12I	$$RAST02	SY33-8552
0T12I	$$RAST04	SY33-8552
0T12I	$$RAST05	SY33-8552
0T12I	$$RAST06	SY33-8552
0T12I	$$RAST07	SY33-8552
0T12I	$$RAST10	SY33-8552
0T12I	$$RAST12	SY33-8552
0T13A	$$RAST04	SY33-8552
0T13A	$$RAST05	SY33-8552

Message	Phase	PLM
0T13A	$$RAST06	SY33-8552
0T13A	$$RAST10	SY33-8552
0T14E	$$RAST03	SY33-8552
0T14E	$$RAST11	SY33-8552
0T15E	$$RAST09	SY33-8552
0T15E	$$RAST13	SY33-8552
0T16I	$$RAST03	SY33-8552
0T16I	$$RAST11	SY33-8552
0T17I	$$RAST03	SY33-8552
0T17I	$$RAST11	SY33-8552
0T18E	$$RAST03	SY33-8552
0T18E	$$RAST11	SY33-8552
0T19E	$$RAST09	SY33-8552
0T20E	$$RAST13	SY33-8552
0V01I	$$BEOJS2	SY33-8553
0V02I	$$BEOJS2	SY33-8553
0V03I	$$BEOJS2	SY33-8553
0V04I	$$BEOJS2	SY33-8553
0V05I	$$BEOJS2	SY33-8553
0V06I	$$BEOJS2	SY33-8553
0V07I	$$BEOJS2	SY33-8553
0V08I	$$BEOJS2	SY33-8553
0V09I	$$BEOJS2	SY33-8553
0V10I	$$BEOJ2A	SY33-8553
0V11I	$$BEOJ2	SY33-8553
0V20D	PDSDM	SY33-8554
0V21D	PDSDM	SY33-8554
0V22D	PDSDM	SY33-8554
0V23D	PDSDM	SY33-8554
0V24I	PDSDM	SY33-8554
0V25I	PDSDM	SY33-8554
0V26I	PDSDM	SY33-8554
0V27I	PDSDM	SY33-8554
0V28I	PDSDM	SY33-8554
0V29I	PDSDM	SY33-8554
0V30I	PDSDM	SY33-8554
0V31I	PDSDM	SY33-8554
0V32I	PDSDM	SY33-8554
0V33I	PDSDM	SY33-8554
0V34I	PDSDM	SY33-8554
0V35I	PDSDM	SY33-8554
0V36I	PDSDM	SY33-8554
0V37I	PDSDM	SY33-8554
0V39I	PDSDM	SY33-8554
0V40I	PDSDM	SY33-8554
0V41I	PDSDM	SY33-8554
0V42I	PDSDM	SY33-8554
0V43I	PDSDM	SY33-8554
0V44I	PDSDM	SY33-8554
04E6	SDEHR	SY33-8554
1A0ND	$JOBCTLD	SY33-8555
1A1ND	$JOBCTLD	SY33-8555
1A2ND	$JOBCTLD	SY33-8555
1A2ND	$JOBCTLG	SY33-8555
1A2ND	$JOBCTLJ	SY33-8555
1A3ND	$JOBCTLD	SY33-8555
1A4ND	$JOBCTLD	SY33-8555
1A4ND	$JOBCTLF	SY33-8555
1A4ND	$JOBCTLJ	SY33-8555
1A4ND	$JOBCTLK	SY33-8555
1A5ND	$JOBCTLD	SY33-8555
1A5ND	$JOBCTLF	SY33-8555
1A5ND	$JOBCTLJ	SY33-8555
1A6ND	$JOBCTLD	SY33-8555
1A6ND	$JOBCTLF	SY33-8555
1A7ND	$JOBCTLD	SY33-8555
1A7ND	$JOBCTLF	SY33-8555
1A7ND	$JOBCTLJ	SY33-8555
1A80D	$JOBCTLD	SY33-8555
1A81I	$JOBCTLA	SY33-8555
1A82D	$JOBCTLD	SY33-8555
1A9ND	$JOBCTLD	SY33-8555
1A9ND	$JOBCTLF	SY33-8555
1B01A	SYSBUFLD	SY33-8554
1B02A	SYSBUFLD	SY33-8554
1B03I	SYSBUFLD	SY33-8554
1B10I	SYSBUFLD	SY33-8554
1B11D	SYSBUFLD	SY33-8554
1B12D	$$BATTF1	SY33-8553
1B12D	$$BATTU1	SY33-8553
1B13A	$$BATTF3	SY33-8553
1B13A	$$BATTF4	SY33-8553
1B14A	$$BATTF2	SY33-8553
1B14A	$$BATTF5	SY33-8553
1B15I	$$BATTF4	SY33-8553
1B15I	$$BATTF5	SY33-8553
1B15I	$$BATTU2	SY33-8553
1B16I	$$BATTF4	SY33-8553
1B16I	$$BATTF5	SY33-8553
1B16I	$$BATTU2	SY33-8553
1B17I	$$BATTF5	SY33-8553
1B18A	$$BATTU2	SY33-8553
1B19I	$$BATTF2	SY33-8553
1B19I	$$BATTF3	SY33-8553
1B20A	$$BATTF2	SY33-8553
1B20A	$$BATTF3	SY33-8553
1B20A	$$BATTF4	SY33-8553
1B20A	$$BATTF5	SY33-8553
1B20A	$$BATTU2	SY33-8553
1B21A	SYSBUFF1	SY33-8554
1C00A	$JOBCTLA	SY33-8555
1C10A	$JOBCTLA	SY33-8555
1C10A	$JOBCTLJ	SY33-8555
1C30A	$JOBCTLJ	SY33-8555
1C33I	$JOBCTLE	SY33-8555
1C40I	$$BATTNA	SY33-8553
1C40I	$$BATTNB	SY33-8553
1C50I	$$BATTNA	SY33-8553
1C50I	$$BATTNB	SY33-8553
1C60D	$$BATTNN	SY33-8553
1C70D	$JOBCTLA	SY33-8555
1C80D	$JOBCTLA	SY33-8555
1C90D	$JOBCTLE	SY33-8555
1I00A	$JOBCTLA	SY33-8555
1I00A	$JOBCTLG	SY33-8555
1I10I	$JOBCTLF	SY33-8555
1I30D	$$BATTNC	SY33-8553
1I32D	$$ABERRZ	SY33-8551

Message	Phase	PLM
1I40D	$$ABERRZ	SY33-8551
1I41D	$$BATTNT	SY33-8553
1I41D	$$BATTNU	SY33-8553
1I42D	$$BATTNT	SY33-8553
1I43D	$$BATTNV	SY33-8553
1I44I	$$BATTNX	SY33-8553
1I45D	$$BATTNT	SY33-8553
1I46D	$$BATTNT	SY33-8553
1I46D	$$BATTNW	SY33-8553
1I47I	$$BATTNT	SY33-8553
1I48I	$$BATTNU	SY33-8553
1I50I	$JOBCTLA	SY33-8555
1I60A	$$BATTNA	SY33-8553
1I70I	$JOBCTLA	SY33-8555
1I82I	$JOBCTLM	SY33-8555
1I83A	$JOBCTLM	SY33-8555
1I84A	$JOBCTLM	SY33-8555
1I86A	$JOBCTLM	SY33-8555
1I89A	$JOBCTLM	SY33-8555
1I90A	$JOBCTLM	SY33-8555
1I91A	$JOBCTLM	SY33-8555
1I92I	$JOBCTLM	SY33-8555
1I93I	$JOBCTLM	SY33-8555
1I95A	$JOBCTLM	SY33-8555
1I96A	$JOBCTLM	SY33-8555
1I97I	$JOBCTLM	SY33-8555
1I98I	$JOBCTLJ	SY33-8555
1I99A	$JOBCTLM	SY33-8555
1L0ND	$JOBCTLK	SY33-8555
1L1ND	$JOBCTLE	SY33-8555
1L1ND	$JOBCTLG	SY33-8555
1L1ND	$JOBCTLK	SY33-8555
1M10A	$JOBCTLA	SY33-8555
1M10A	$JOBCTLG	SY33-8555
1M20D	$JOBCTLA	SY33-8555
1M3ND	$JOBCTLA	SY33-8555
1M4ND	$JOBCTLG	SY33-8555
1M5ND	$JOBCTLD	SY33-8555
1M5ND	$JOBCTLF	SY33-8555
1M6ND	$JOBCTLE	SY33-8555
1M7ND	$JOBCTLE	SY33-8555
1M8ND	$JOBCTLE	SY33-8555
1M9ND	$JOBCTLE	SY33-8555
1N00I	$JOBCTLA	SY33-8555
1N10D	$JOBCTLE	SY33-8555
1N2ND	$JOBCTLE	SY33-8555
1N30D	$JOBCTLE	SY33-8555
1N40D	$JOBCTLE	SY33-8555
1N50D	$JOBCTLE	SY33-8555
1N60D	$JOBCTLE	SY33-8555
1N7ND	$JOBCTLG	SY33-8555
1N80I	$JOBCTLG	SY33-8555
1N90I	$JOBCTLG	SY33-8555
1P0ND	$JOBCTLJ	SY33-8555
1P00D	$$BATTNE	SY33-8553
1P00D	$$BATTNF	SY33-8553
1P1ND	$JOBCTLJ	SY33-8555
1P10D	$$BATTNG	SY33-8553
1P20D	$JOBCTLE	SY33-8555
1Q62I	$$ABERRZ	SY33-8551
1Q62I	$$BATTNC	SY33-8553
1S0ND	$JOBCTLE	SY33-8555
1S00D	$$BATTNB	SY33-8553
1S00D	$$BATTNC	SY33-8553
1S00D	$$BATTNE	SY33-8553
1S00D	$$BATTNF	SY33-8553
1S00D	$$BATTNG	SY33-8553
1S00D	$$BATTNH	SY33-8553
1S00D	$$BATTNN	SY33-8553
1S00D	$$BATTNQ	SY33-8553
1S00D	$$BATTNS	SY33-8553
1S00D	$$BATTNT	SY33-8553
1S00D	$$BATTNU	SY33-8553
1S00D	$$BATTNY	SY33-8553
1S00D	$$BATTNZ	SY33-8553
1S00D	$$BATTN2	SY33-8553
1S01D	$$ABERRZ	SY33-8551
1S02D	$$ABERRZ	SY33-8551
1S1ND	$JOBCTLG	SY33-8555
1S1ND	$JOBCTLJ	SY33-8555
1S1ND	$JOBCTLK	SY33-8555
1S1NI	$JOBCTLA	SY33-8555
1S1NI	$JOBCTLB	SY33-8555
1S1NI	$JOBCTLD	SY33-8555
1S1NI	$JOBCTLE	SY33-8555
1S1NI	$JOBCTLF	SY33-8555
1S1NI	$JOBCTLG	SY33-8555
1S1NI	$JOBCTLJ	SY33-8555
1S1NI	$JOBCTLK	SY33-8555
1S1NI	$JOBCTLM	SY33-8555
1T00A	$JOBCTLA	SY33-8555
1T10I	$JOBCTLA	SY33-8555
1T20I	$JOBCTLD	SY33-8555
1T50A	$JOBCTLD	SY33-8555
1T60A	$JOBCTLD	SY33-8555
1T70A	$JOBCTLD	SY33-8555
2100I	$LNKEDT	SY33-8556
2101I	$LNKEDT	SY33-8556
2102I	$LNKEDT	SY33-8556
2110I	$LNKEDT	SY33-8556
2111I	$LNKEDT	SY33-8556
2112I	$LNKEDT	SY33-8556
2113I	$LNKEDT	SY33-8556
2114I	$LNKEDT	SY33-8556
2116I	$LNKEDT	SY33-8556
2120I	$LNKEDT	SY33-8556
2121I	$LNKEDT	SY33-8556
2122I	$LNKEDT	SY33-8556
2123I	$LNKEDT	SY33-8556
2124I	$LNKEDT	SY33-8556
2125I	$LNKEDT	SY33-8556
2130I	$LNKEDT	SY33-8556
2131I	$LNKEDT	SY33-8556
2132I	$LNKEDT	SY33-8556
2133I	$LNKEDT	SY33-8556
2135I	$LNKEDT	SY33-8556
2136I	$LNKEDT	SY33-8556
2140I	$LNKEDT	SY33-8556

Message	Phase	PLM
2141I	$LNKEDT	SY33-8556
2142I	$LNKEDT	SY33-8556
2143I	$LNKEDT	SY33-8556
2144I	$LNKEDT	SY33-8556
2145I	$LNKEDT	SY33-8556
2146I	$LNKEDT	SY33-8556
2147I	$LNKEDT	SY33-8556
2150I	$LNKEDT	SY33-8556
2151I	$LNKEDT	SY33-8556
2155I	$LNKEDT	SY33-8556
2156I	$LNKEDT	SY33-8556
2158I	$LNKEDT	SY33-8556
2170I	$LNKEDT	SY33-8556
2181I	$LNKEDT	SY33-8556
2182I	$LNKEDT	SY33-8556
2184I	$LNKEDT	SY33-8556
2185I	$LNKEDT	SY33-8556
2191I	$LNKEDT	SY33-8556
2192I	$LNKEDT	SY33-8556
2193I	$LNKEDT	SY33-8556
2194I	$LNKEDT	SY33-8556
2195I	$LNKEDT	SY33-8556
2197I	$LNKEDT	SY33-8556
2199I	$LNKEDT	SY33-8556
3C30I	CORGZ	SY33-8557
3C30I	CORGZ1	SY33-8557
3C66I	CORGZ	SY33-8557
3C66I	CORGZ4	SY33-8557
3C66I	CORGZ5	SY33-8557
3C66I	CORGZ6	SY33-8557
3C66I	CORGZ7	SY33-8557
3C67I	CORGZ	SY33-8557
3C67I	CORGZ5	SY33-8557
3C67I	CORGZ6	SY33-8557
3C67I	CORGZ7	SY33-8557
3E01I	EREPMNTR	SY33-8554
3E02I	EREPMNTR	SY33-8554
3E04I	EREPMNTR	SY33-8554
3E06I	EREPMNTR	SY33-8554
3E07D	EREPESWK	SY33-8554
3E07D	EREPTES	SY33-8554
3E08A	EREPESWK	SY33-8554
3E09A	EREPESTR	SY33-8554
3E10I	EREPEDIT	SY33-8554
3E10I	EREPSMCP	SY33-8554
3E11D	EREPMNTR	SY33-8554
3E12D	EREPMNTR	SY33-8554
3E14A	EREPMNTR	SY33-8554
3E15A	EREPESTR	SY33-8554
3E15A	EREPHIST	SY33-8554
3E15A	EREPTES	SY33-8554
3E18A	EREPHIST	SY33-8554
3E20I	EREPESWK	SY33-8554
3E21I	EREPEDIT	SY33-8554
3E22I	EREPEDIT	SY33-8554
3E25I	EREPESTR	SY33-8554
3E25I	EREPHIST	SY33-8554
3E25I	EREPMNTR	SY33-8554
3E25I	EREPTES	SY33-8554

Message	Phase	PLM
3E26I	EREPMNTR	SY33-8554
3E27I	EREPMNTR	SY33-8554
3E28I	EREPMNTR	SY33-8554
3E29I	EREPMNTR	SY33-8554
3E30A	EREPHIST	SY33-8554
3E31A	EREPESTR	SY33-8554
3E31A	EREPHIST	SY33-8554
3E31A	EREPTES	SY33-8554
3E32A	EREPRDE	SY33-8554
3E33A	EREPRDE	SY33-8554
3E34I	EREPRDE	SY33-8554
3E35I	EREPRDE	SY33-8554
3E36I	EREPRDE	SY33-8554
3E37I	EREPRDE	SY33-8554
3E38I	EREPRDE	SY33-8554
3E40I	EREPRDE	SY33-8554
3E41I	EREPRDE	SY33-8554
3E42I	EREPRDE	SY33-8554
3E43I	EREPRDE	SY33-8554
3E67I	CORGZ6	SY33-8557
3H30I	MAINTP2	SY33-8557
3M00I	PSERV	SY33-8557
3M09I	PSERV	SY33-8557
3M10I	CORGZ	SY33-8557
3M10I	CORGZ1	SY33-8557
3M10I	DSERV	SY33-8557
3M10I	MAINT	SY33-8557
3M10I	MAINTA	SY33-8557
3M10I	PSERV	SY33-8557
3M10I	RSERV	SY33-8557
3M10I	SSERV	SY33-8557
3M11I	MAINTR2	SY33-8557
3M20I	PSERV	SY33-8557
3M21I	CORGZ	SY33-8557
3M21I	CORGZ1	SY33-8557
3M21I	CORGZ3	SY33-8557
3M21I	CORGZ4	SY33-8557
3M21I	CORGZ5	SY33-8557
3M21I	CORGZ6	SY33-8557
3M21I	CORGZ7	SY33-8557
3M21I	CORGZ8	SY33-8557
3M21I	DSERV	SY33-8557
3M21I	MAINRTN	SY33-8557
3M21I	MAINTCL	SY33-8557
3M21I	MAINTCN	SY33-8557
3M21I	MAINTDR	SY33-8557
3M21I	MAINTR2	SY33-8557
3M21I	PSERV	SY33-8557
3M21I	RSERV	SY33-8557
3M21I	SSERV	SY33-8557
3M23I	MAINTS2	SY33-8557
3M24I	MAINTS2	SY33-8557
3M25I	MAINTS2	SY33-8557
3M26I	MAINTS2	SY33-8557
3M27I	MAINTP2	SY33-8557
3M27I	MAINTR2	SY33-8557
3M27I	MAINTS2	SY33-8557
3M28I	MAINTS2	SY33-8557
3M29I	MAINTP2	SY33-8557

Message	Phase	PLM
3M32I	MAINTP2	SY33-8557
3M33I	$MAINDIR	SY33-8557
3M33I	CORGZ3	SY33-8557
3M33I	MAINRTN	SY33-8557
3M33I	MAINTUP	SY33-8557
3M33I	PSERV	SY33-8557
3M33I	RSERV	SY33-8557
3M33I	SSERV	SY33-8557
3M34I	MAINT	SY33-8557
3M34I	MAINTP2	SY33-8557
3M35I	DSERV	SY33-8557
3M37I	MAINT	SY33-8557
3M37I	MAINTA	SY33-8557
3M37I	MAINTCL	SY33-8557
3M37I	MAINTCN	SY33-8557
3M37I	MAINTDR	SY33-8557
3M38I	MAINTP2	SY33-8557
3M43I	CORGZ1	SY33-8557
3M43I	CORGZ3	SY33-8557
3M43I	CORGZ4	SY33-8557
3M43I	DSERV	SY33-8557
3M43I	MAINT	SY33-8557
3M43I	MAINTCL	SY33-8557
3M43I	MAINTDR	SY33-8557
3M43I	MAINTP2	SY33-8557
3M43I	MAINTR2	SY33-8557
3M43I	MAINTS2	SY33-8557
3M43I	MAINTUP	SY33-8557
3M43I	PSERV	SY33-8557
3M43I	RSERV	SY33-8557
3M43I	SSERV	SY33-8557
3M44I	CORGZ6	SY33-8557
3M44I	MAINT	SY33-8557
3M44I	MAINTCN	SY33-8557
3M45I	DSERV	SY33-8557
3M52I	$MAINDIR	SY33-8557
3M52I	CORGZ3	SY33-8557
3M52I	MAINTP2	SY33-8557
3M52I	MAINTR2	SY33-8557
3M52I	MAINTS2	SY33-8557
3M52I	MAINTUP	SY33-8557
3M53I	CORGZ3	SY33-8557
3M53I	MAINTP2	SY33-8557
3M53I	MAINTR2	SY33-8557
3M53I	MAINTS2	SY33-8557
3M53I	MAINTUP	SY33-8557
3M54I	$MAINDIR	SY33-8557
3M54I	MAINTDR	SY33-8557
3M54I	MAINTUP	SY33-8557
3M55I	MAINTR2	SY33-8557
3M62I	CORGZ7	SY33-8557
3M62I	MAINTA	SY33-8557
3M63I	CORGZ1	SY33-8557
3M63I	CORGZ7	SY33-8557
3M63I	MAINTA	SY33-8557
3M64I	MAINTA	SY33-8557
3M65I	CORGZ	SY33-8557
3M65I	CORGZ7	SY33-8557
3M65I	MAINTA	SY33-8557
3M66I	CORGZ7	SY33-8557
3M67I	CORGZ7	SY33-8557
3M68I	MAINTA	SY33-8557
3M68I	MAINTCN	SY33-8557
3M70A	MAINTA	SY33-8557
3M70A	MAINTCN	SY33-8557
3M70I	MAINTCN	SY33-8557
3M75I	MAINTCN	SY33-8557

Message	Phase	PLM
3M80I	MAINTA	SY33-8557
3M80I	MAINTCN	SY33-8557
3M81A	MAINTCN	SY33-8557
3M81I	MAINTCN	SY33-8557
3M90I	$MAINDIR	SY33-8557
3M92I	$MAINDIR	SY33-8557
3M93I	$MAINDIR	SY33-8557
3N43I	DSERV	SY33-8557
3U10I	MAINTUP	SY33-8557
3U11I	MAINTUP	SY33-8557
3U20I	MAINTUP	SY33-8557
3U21I	MAINTUP	SY33-8557
3U30I	MAINTUP	SY33-8557
3U31I	MAINTUP	SY33-8557
3U32I	MAINTUP	SY33-8557
3U33I	MAINTUP	SY33-8557
4C10D	PDAID	SY33-8554
4C11D	PDAID	SY33-8554
4C12D	PDAID	SY33-8554
4C13D	PDAID	SY33-8554
4C14D	PDAID	SY33-8554
4C15D	PDAID	SY33-8554
4C16D	PDAID	SY33-8554
4C17D	PDAID	SY33-8554
4C17D	SDAID1	SY33-8554
4C17D	SDPAR	SY33-8554
4C20D	PDAID	SY33-8554
4C21A	PDAID	SY33-8554
4C22A	PDAID	SY33-8554
4C23D	PDAID	SY33-8554
4C24A	$$BPDAID	SY33-8554
4C24A	PDAIDFTT	SY33-8554
4C24A	PDAIDGTP	SY33-8554
4C24A	PDAIDGTT	SY33-8554
4C24A	PDAIDITP	SY33-8554
4C24A	PDAIDITT	SY33-8554
4C24A	PDAIDTDT	SY33-8554
4C26I	PDLIST	SY33-8554
4C27D	PDAID	SY33-8554
4C28D	PDAID	SY33-8554
4C42A	DUMPGEN	SY33-8554
4C43A	DUMPGEN	SY33-8554
4C44A	DUMPGEN	SY33-8554
4C46A	DUMPGEN	SY33-8554
4C50E	PDAID	SY33-8554
4C51D	PDAID	SY33-8554
4C52E	PDAID	SY33-8554
4C53I	SDAID1	SY33-8554
4C54I	SDAID1	SY33-8554
4C55D	SDAID1	SY33-8554
4C56E	SDAID1	SY33-8554
4C57E	SDAID2	SY33-8554
4C58D	SDPAR	SY33-8554
4C59D	SDPAR	SY33-8554
4C60D	SDPAR	SY33-8554
4C61D	SDPAR	SY33-8554
4C62D	SDPAR	SY33-8554
4C63D	SDPAR	SY33-8554
4C64D	SDPAR	SY33-8554
4C65D	SDPAR	SY33-8554
4C66D	SDPAR	SY33-8554
4C67D	SDPAR	SY33-8554
4C68D	SDPAR	SY33-8554
4C69D	SDPAR	SY33-8554
4E10I	$$ABERAA	SY33-8552
4E10I	$$ABERA2	SY33-8552

A

AB parameter 25
AB routine user's save area 97
AB table 96
abnormal termination option table 96
ACANCEL parameter 25
accessing the PUB2 area 63
ACCTCOMN (job accounting interface common table) 55
ACCTxx (job accounting interface partition table 56
ALIGN parameter 25
ALLOC macro 30, 34
ALLOCR macro 30, 34
anchor table (ANCHTAB) 107
ANCHTAB 107
AP parameter 23
ASCII parameter 24
ASCII translation tables 72
ASSGN macro 32, 35
asynchronous processing 83
attaching a subtask 83, 100
attention
  physical 195
  PIB 50
  task PIK 447

B

BBOX (boundary box) 54
BDYBOX macro 36
BGCOMREG (partition communications region) 44
BGPGR parameter 30
BGR parameter 30
BLKMPX parameter 29
BMPX parameter 29
boundary box (BBOX) 54
BSCLNS parameter 31
BUFSIZE parameter 29
burst-mode device on byte mpx channel 152

C

CALLSEQ (fetch parameter save area) 190
cancel codes and messages 453
CAW (channel address word) 150
CBF parameter 26
CBTAB (console buffering table) 181
CCB (command control block) 147
CCB copy block 156
CCH (channel check handler) 198
CCW (channel command word) 150
CCW copy block 158
CCWTCB (translation control block) 154
CHAN parameter 24
channel
  address word (CAW) 150
  appendage routine 151
  bucket 144
  check handler (CCH) 198
  command word (CCW) 150
  control table (CHNTAB) 137
  error conditions 199
  program translation control block (CCWTCB) 154
  queue (CHANQ) table 137
  queue operation 139
  scheduler 149, 216
  status word (CSW) 150
  switching 136
CHANQ parameter 31
CHANQ table (channel queue table) 137
CHANSW parameter 29, 32
CHARSET parameter 25
charts (see phases)
CHNTAB (channel control table) 137
CHUN parameter 32
command chaining for retry 151
command control block (CCB) 147
communications region
    low core pointer 19
    partition 44
    system 42
CONFG macro 24, 34
console buffering 180
console buffering table (CBTAB) 181
copy block 154
core image library block 194
CPU retry 197
CRT constant table (CRTTAB) 70
CRTGEN macro 35
CRTSAV macro 35
CRTTAB (CRT constant table) 70
CRT transient area 15
CSW (channel status word) 150
CSW testing 152
CYL parameter 33

D

DASDFP parameter 26
DATE parameter 25
deactivation of a partition 122
DECK parameter 24
density data 451
detail charts (see phases)
device type codes 449
DIB (disk information block table) 58
directory entry
  core image 190
  second level 191
  system 190

directory list
second level 191
system 189
disk error recovery 183
disk information block (DIB) table 58
DISK parameter 29
DOC parameter 28
DPD macro 33, 35
DPDTAB (page data set table) 70
DUMP parameter 25
DVCGEN macro 32, 35
DVCTYP parameter 31
D2311 parameter 31
D2314 parameter 31
D2321 parameter 31
D2400 parameter 31
D3330 parameter 31
D3340 parameter 31
D3410 parameter 31
D3420 parameter 31
D3540 parameter 31
D3886 parameter 31

E

EBCDIC - ASCII translation table 72
ECB (event control block) 100
ECC (error checking and correction) 199
ECPREAL parameter 28
EDECK parameter 25
EFL (error frequency limit) 199
emulator ECB table (EUECBTAB) 101
ERBLOC (error recovery block) 182
ERPIB (error recovery procedure information block) 202
ERRLOG parameter 23
error
bytes 21
checking and correction (ECC) 199
frequency limit (EFL) 199
queue 182
queue entry 182
recovery block (ERBLOC) 182
recovery procedure information block (ERPIB) 202
ERRQ parameter 27
ERRS parameter 25
EU parameter 23
EUECBTAB (emulator ECB table) 101
EVA parameter 26
event control block (ECB) 100
exponential average
for deactivation 123
for reactivation 126
external interrupt 88

F

FAVP (first available pointer) 130
FCEPGOUT routine 121
fetch
parameter list 190
parameter save area (CALLSEQ) 190
table (FTTAB) 191
FICL (first-in-class-list table) 130
first available pointer (FAVP) 130
first-in-class-list (FICL) table 130
first-on-channel-list (FOCL) 130
fix information block 159
fixing of pages
permanently (see also PFIX) 120
temporarily (see also TFIX) 119
fix waiter table (FIXWTAB) 108
FIXWTAB 108
flowcharts
detail 258
general 205
FLPTR (free list pointer) 133
FnCOMREG (partition communications region) 44
FOCL (first-on-channel-list) 130
FOPT macro 25, 34
FP parameter 24
free list pointer (FLPTR) 133
freeing of pages (PFREE and TFREE) 121
FREEREAL 121
FTTAB (Fetch table) 191
F1 parameter 30, 33
F1PGR parameter 30
F1R parameter 30
F2 parameter 30, 33
F2PGR parameter 30
F2R parameter 30
F3 parameter 30, 33
F3PGR parameter 30
F3R parameter 30
F4 parameter 30, 33
F4PGR parameter 30
F4R parameter 30

G

GETVIS parameter 27
general charts (see phases)
GETREAL 118, 244
global settings 36

H

hard wait error codes 21
headqueueing 94
Headqueue PUB table (HQPUBTAB) 94
HQPUBTAB 94
H1 parameter 33
H2 parameter 33

I

IDAL (indirect data address list) 159
IDAL block 159
IDAW (indirect data address word) 159
identification keys
I/O requestor's partition (REQID) 448
I/O requestor's task (TKREQID) 448
logical transient (LTK) 448
logical transient owner (LTK) 448
partition (PIK) 447
task (TIK) 447

IDRA (independent directory read-in
area 194
IDRA parameter 26
IHBRDWRA macro 36
IJLQDSCT macro 36
IJLQTSVC macro 34
independent directory read-in area
(IDRA) 194
indirect data address list (IDAL) 159
indirect data address word (IDAW) 159
interrupts
external 88, 205
I/O 85, 150, 205
machine check 88, 197, 225
program check 85, 209
supervisor call (SVC) 88, 205
interval timer option table 96
interval timer request table (ITREQ) 97
IODEV parameter 30
IOINTER macro 35
IOTAB macro 35
I/O interrupt 85, 150
I/O table interrelationships 130
ISTAVT macro 36
IT parameter 25
IT routine user's save area 97
ITREQ 97
ITTAB 96

J

JA parameter 27
JALICCS parameter 27
JIB (job information block) 136
JIB parameter 31
job accounting interface common table
(ACCTCOMN) 55
job accounting interface partition
table (ACCTxx) 56
job information block (JIB) 136

L

label list 441
LIK (see identification keys) 448
line mode table (LMT) 60
LINES parameter 25
LIST parameter 24
LISTX parameter 25
LMT (line mode table) 60
load leveler 121
local directory list 189
LOG parameter 25
logical transient area (LTA) 15, 52
logical unit block (LUB) table 138
low real storage 21
LTA (logical transient area) 15, 52
LTK (identification keys) 448
LUB (logical unit block table) 138

M

machine check
analysis and recovery (MCAR) 199
check interrupt 88, 197
check logout area 199
macro functions 34
macros
ALLOC 30, 34
ALLOCR 30, 34
ASSGN 33, 35
BDYBOX 36
CONFG 24, 34
CRTGEN 35
CRTSAV 35
DPD 33, 35
DVCGEN 33, 35
FOPT, text 25, 34
detail charts 261, 291, 304, 326
IHBRDWRA 36
IJLQDSCT 36
IJLQTSVC 34
IOINTER, text 35
detail charts 259, 272, 339, 343
IOTAB 30, 35
ISTAVT 36
MAPPUB2 36
MCRAS, text 35
detail charts 422
PIOCS, text 29, 34
detail charts 260, 266, 305
SEND 33, 35
SGAM, text 35
detail charts 412
SGCCWT, text 35
detail charts 357
SGDFCH, text 35
detail charts 346, 434
SGDSK, text 35
detail charts 285
SGEND, text 35
detail charts 344
SGPMGR, text 35
detail charts 375
SGPOPT, text 35
detail charts 416
SGSCVRT, text 35
detail charts 436.1
SGSVC, text 34
detail charts 315
SGTHAP, text 34
detail charts 268
SMICR, text 35
detail charts 265, 298, 341
STDJC 24, 34
SUPVR 23, 34
VSTAB 29, 34
MAPPUB2 macro 36
MCAR 199
MCH parameter 24
MCRAS macro 35
message cross reference list 455, 457
messages and cancel codes 453
MICR DTF addresses (PDTABB) 54
MICR DTF pointers (PDTABA) 54
MICR parameter 23
MODE parameter 32
MODEL parameter 24
MRSLCH parameter 29

N

next-in-class list (NICL) table 130
NICL 130
NPARTS parameter 23

O

OC parameter 25
OC routine user's save area 97
OC table 95
OLTEP parameter 28
operator communication table 95
option tables
  AB (abnormal termination) 96
  IT (interval timer) 96
  OC (operator communication) 95
  PC (program check) 86
  PHO (page fault handling overlap) 118

P

page
  data set table (DPDTAB) 70
  fault handling overlap option table 118
  frame table (PFT) 114
  queue (PGQU) 117
PAGETAB (see pagein table) 111
page table (PT) 114
PAGEIN parameter 24
PAGEIN routine 111
pagein table (PAGETAB) 111
parameters
  AB 25
  ACANCEL 25
  ALIGN 25
  AP 23
  ASCII 24
  BGPGR 30
  BGR 30
  BLKMPX 29
  BMPX 29
  BSCLNS 31
  BUFSIZE 29
  CBF 26
  CHAN 24
  CHANQ 31
  CHANSW 29, 31
  CHARSET 25
  CHUN 32
  CYL 33
  DASDFP 26
  DATE 25
  DECK 24
  DISK 29
  DOC 28
  DUMP 25
  DVCTYP 32
  D2311 31
  D2314 31
  D2321 31
  D2400 31
  D3330 31
  D3340 31
  D3410 31
  D3420 31
  D3540 31
  D3886 31
  ECPREAL 28
  EDECK 25
  ERRLOG 23
  ERRQ 27
  ERRS 25
  EU 23
  EVA 26
  FP 24
  F1 30, 33
  F1PGR 30
  F1R 30
  F2 30, 33
  F2PGR 30
  F2R 30
  F3 30, 33
  F3PGR 30
  F3R 30
  F4 30, 33
  F4PGR 30
  F4R 30
  GETVIS 27
  H1 33
  H2 33
  IDRA 26
  IODEV 30
  IT 25
  JA 27
  JALIOCS 27
  JIB 31
  LINES 25
  LIST 24
  LISTX 25
  LOG 25
  MCH 24
  MICR 23
  MODE 32
  MODEL 24
  MRSLCH 29
  NPARTS 23
  OC 25
  OLTEP 28
  PAGEIN 24
  PC 25
  PCIL 28
  PD 26
  PFIX 28
  PHO 24
  POWER 23
  PRTY 28
  PSLD 28
  RELLDR 28
  RETAIN 26
  RMS 24
  RPS 26
  RSIZE 29
  SKSEP 26
  SLD 29
  SPARM 25
  SSLNS 32
  SVA 25
  SYM 25
  SYSFIL 27
  SYSxxx 33

TAPE 29
TEB 25
TEBV 26
TCD 28
TP 23
TRKHLD 26
UNIT 33
VCLID 33
VSAM 27
VSIZE 29
WAITM 26
XREF 24
ZONE 28
partition
communications region 44
identification key (PIK) 447
save area 53
PC option table 86
PC parameter 25
PC routine user's save area 97
PCIL parameter 28
PD parameter 26
PDAID 69
PDTABA (MICR DTF pointers) 54
PDTABB (MICR DTF addresses) 54
PFIX 119
PFIX parameter 28
PFREE 121
PFT (page frame table) 114
PGQU (page queue) 117
phases
$$A$SUP1, detail charts 258
general charts 205
$$ABERRZ, text 195
detail charts 437
general chart 211
$$ABERZ1, text 195
detail chart 440
general chart 211
PHO parameter 24
PHO table 118
physical
transient area (PTA) 15
attention 195, 211
unit block (PUB) table 133
unit block ownership (PUBOWNER)
table 135
PIB (program information block) 50
PIB2TAB (PIB extension) 53
PIK (partition identification key) 447
PIOCS macro 29, 34
POWER parameter 23
POWER table (POWERTAB) 59
POWERTAB 59
private second level directory 191
processing, asynchronous 83
program
check option table 86
check interrupt 85
information block (PIB) 50
information block extension
(PIB2TAB) 53
retrieval 189
status word (PSW) 150
PRTY parameter 28
PSLD parameter 28
PSW (program status word) 150
PT (page table) 114
PTA (physical transient area) 15
PUB (physical unit block) 133
PUBOWNER (physical unit block owner
table) 135
PUB2 64
PUB2AREA 63
PUB2 table
accessing the 63
entry format
for DASD 65
for 3540 diskette 66
for tapes 67
for unit record and unsupported
devices 64
for 3886 CCR 64

R

R transients 200
RAS linkage area (RASLINK) 203
RAS monitor table (RASTAB) 201
RAS transients 200
RASLINK (RAS linkage area) 203
RASTAB (RAS monitor table) 201
RCB (resource control block) 101
reactivation of a partition 125
recorder file table (RFTABLE) 61
recovery transients 200
recovery transient area (RTA) 15, 200
reentry rate 123
reentry rate table (RTAB) 123
RELLDR parameter 28
RELPAG routine 110, 121
REQID (identification keys) 448
resource
control block (RCB) 101
table (TRTMASK) 77
usage record table (RURTAB) 106
RETAIN parameter 26
RFTABLE (recorder file table) 61
RID (routine identifier) 87
RMS parameter 24
routine identifier (RID) 87
RPS parameter 26
RSIZE parameter 29
RTA (recovery transient area) 15, 200
RTAB (reentry rate table) 123
RURTAB (resource usage record table) 106

S

SAB (seek address block) 179
save area
user's 53, 97
system 71
SDL (system directory list) 189
second level directory 191
second level directory list 191
seek address block (SAB) 179
segment table (STAB) 114
SEND macro 33, 35
SGAM macro 35
SGCCWT macro 35
SGDFCH macro 35
SGDSK macro 35

SGEND macro 35
SGPMGR macro 35
SGPOPT macro 35
SGSCVRT macro 35
SGSVC macro 34
SGTHAP macro 34
shared virtual area (SVA) 189
SKSEP parameter 26
SLD parameter 29
SMICR macro 35
soft wait error codes 22
SPARM parameter 25
SSLNS parameter 32
STAB (segment table) 114
STDJC macro 23, 34
STID (system task id field) 78
storage map, supervisor 17
storage organization, multiprogramming 16
subtask attaching 83, 100
supervisor call interrupt 85
SUPVR macro 23, 34
SVA 189
SVC interrupt 88
SVC list 89
SYM parameter 25
SYSCOM 42
SYSFIL parameter 27
system
  communications region 42
  directory list (SDL) 189
  generation macros
    functions 34
    globals 36
    SYSGEN options 23
  program flow 14
  save areas 71
  task identification field 78
  task block 79
SYSxxx parameter 33

T

tape error block (TEB) 137
TAPE parameter 29
task identification key (TIK) 447
task selection 77
TCB (translation control block) 154
TEB (tape error block) 137
TEB parameter 25
TEBV parameter 26
TFIX 119
TFREE 121
THFLPTR (track hold free list pointer) 175
TIK (identification keys) 447
THTAB (track hold table) 175
TKREQID (identification keys) 447
TOD parameter 28
total message cross reference list 455, 457
TP parameter 23
track hold free list pointer (THFLPTR) 175
track hold table (THTAB) 175
transient areas 15
translation control block (CCWTCB) 154
translation tables, ASCII 72
TRKHLD parameter 26
TRTMASK (resource table) 77

U

UNIT parameter 33

V

VOLID parameter 33
VSAM parameter 27
VSIZE parameter 29
VSTAB macro 29, 34

W

WAIT error codes 21
WAITM parameter 26

X

XREF parameter 24

Z

ZONE parameter 28

DOS/VS Supervisor Logic

SY33-8551-1

READER'S COMMENT FORM

This sheet is for comments and suggestions about this manual. We would appreciate *your* views, favorable or unfavorable, in order to aid us in improving *this* publication. This form will be sent directly to the author's department. Please include your name and address if you wish a reply. Contact your IBM branch office for answers to technical questions about the system or when requesting additional publications. Thank you.

Your comments* and suggestions:

*** We would especially appreciate your comments on any of the following topics:**

Clarity of the text	Accuracy	Index	Illustrations	Appearance	Paper
Organization of the text	Cross-references	Tables	Examples	Printing	Binding

SY33-8551-1

YOUR COMMENTS, PLEASE . . .

This manual is part of a library that serves as a reference source for systems analysts, programmers and operators of IBM systems. Your answers to the questions on the back of this form, together with your comments, will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

CUT ALONG THIS LINE

Fold   Fold

FIRST CLASS
PERMIT NO. 1359
WHITE PLAINS, N. Y.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

IBM Corporation
1133 Westchester Avenue
White Plains, N.Y. 10604

Attention: Department 813 U

Fold   Fold

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

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

IBM Technical Newsletter

This Newsletter No.	SN33-8770
Date	June 30, 1974
Base Publication No.	SY33-8551-1
File No.	S370-36 (DOS/VS Rel. 30)
Previous Newsletters	SN33-8769

DOS/VS Supervisor
Logic

This Technical Newsletter, a part of release 30 of the IBM Disk Operating System/Virtual Storage, DOS/VS, provides replacement pages for your publication. These replacement pages remain in effect for subsequent DOS/VS releases unless specifically altered. Pages to be inserted and/or removed are:

Cover - 12	77, 78	205, 206	387, 388
19, 20	85, 86	213, 214	391, 392
23 - 26	89 - 92	257 - 260	403 - 406
26.1 (added)	95 - 98	263, 264	425, 426
29, 30	101, 102	273 - 284	433 - 436
33 - 36	107 - 114	293, 294	436.1, 436.2 (added)
36.1 (added	129, 130	305 - 310	437 - 446
37, 38	133, 134	343 - 346	453, 454
41 - 44	151, 152	349, 350	457 - 470
47 - 52	173, 174	353 - 356	
57 - 60	189 - 192	363, 364	

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

Summary of Amendments

Information has been included to reflect the support of the Rotational Position Sensing (RPS) feature, and of Job Accounting Enhancements. Furthermore, various maintenance changes are included.

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

IBM Laboratory, Publications Dept., Uithoorn, The Netherlands

Printed in U.S.A.

Technical Newsletter

This Newsletter No.	SN33-8769
Date	January 2, 1974
Base Publication No.	SY33-8551-1
File No.	S370-36
Previous Newsletters	None

DOS/VS Supervisor Logic

This Technical Newsletter, a part of release 29 of the IBM DOS/VS Disk Operating System, provides replacement pages for your publication. These replacement pages remain in effect for subsequent DOS/VS releases unless specifically altered. Pages to be inserted and/or removed are:

Cover, 2
259 - 262
265 - 268
273 - 282
317, 318
343, 344
344.1, 344.2 (added)
425 - 428
439 - 446

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

Summary of Amendments

Additional changes have been made in the detailed flowcharts to the coverage of DOS/VS System enhancements.

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

IBM Laboratory, Publications Dept., Uithoorn, The Netherlands

Printed in U.S.A.

SY33-8551-1

DOS/VS Supervisor Logic (S370-36) Printed in U.S.A. SY33-8551-1

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

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