Xerox Control Program-Five (CP-V)

Xerox 560 and Sigma 6/7/9 Computers

Data Base
Technical Manual

XEROX

Xerox Control Program-Five (CP-V)

Xerox 560 and Sigma 6/7/9 Computers

Data Base
Technical Manual

90 19 95D

FEBRUARY 1976

File No.: 1X33
XN15, Rev. 0
Printed in U.S.A.

REVISION

This publication documents the D00 version of Control Program-Five (CP-V). Pages dated 2/10/76 denote changes that reflect the D00 version.

TABLE OF CONTENTS

	Section	Page
JIT - Job Information Table	VA	1
Usage	VA	1
Description	VA	2
JIT	VA	3
JIT Usage by Processors	VA.01	19
Scheduler	VC	20
Scheduler Queues	VC	20
Event Transition Tables	VC	20
Resource Sub-Queues	VC	20
Situational Priority Increments	VC	20
Scheduler States	VC	21
Scheduler Events	VC	21
Scheduler/Swapper Tables	VC	22
Scheduler/State Tables/Queues	VC	22
Scheduler Priority and Real-Time Data	VC	23
User Tables	VD	24
Common Multiprocessor Control Tables	VD.01	27
Shared Processor Tables	VE	35
Tables Displaced by Processor Number	VE	35
Processor Table Layout	VE	36
Ghost Job Tables	VE	37
Memory Allocation	VF	38
On-Line Memory Pointers	VF.01	38
Batch Memory Pointers	VF.01	38
Physical Memory Allocation	VF.01	39
CP-V Buffer Linking	VF.01	40
Swap Storage Layout	VF.02	41
Swapping RAD Granule Table	VF.02	42
Input/Output Tables	VG	45
DCT - Device Control	VG.01	45
RAD/PACK Characteristics	VG.01	48
IOQ - I/O Enqueueing	VG.02	49
DOT - Devoce Operations	VG.02	50
CIT - Channel Information	VG.02	50
DTT - Device Type - Class Tables	VG.03	51
AVR Tables	VG.04	57
AVR Table Bit Definitions	VG.04	58
Table Contents and Bit Settings	VG.04	59
COC Tables	VG.05	62

	Section	Page
File Tables	VH	66
Master Index Format for Level 0	VH.01	66
Master Index Format for a Higher Level Index	VH.01	69
Account Directory	VH.02	70
Master Index Format for the Public File Account Directory	VH.02	70
Master Index Format for a Private Volume Set Account Directory	VH.02	71
File Directory	VH.03	72
Master Index Format for Public and Private File Directories	VH.03	73
Free Granule Pool (FSP) Format	VH.03	75
File Information Table (FIT) Format	VH.04	76
Private Volume Set Tables	VH.06	80
VTOC - Volume Table of Contents	VH.06	81
HGP - Allocation Tables	VH.07	83
CFU - Current File Usage Tables	VH.08	85
ACNCFU - Account Directory CFU	VH.08	86
FILCFU - File Directory CFU	VH.08	89
User File CFU	VH.08	91
File DCB	VH.09	93
Labeled Tape DCB	VH.10	104
ANS Labeled Tape DCB	VH.11	112
Device DCB	VH.12	119
CP-V Labeled Tape	VH.16.01	126
General Format of Sentinels	VH.16.01	127
Record Format	VH.16.01	129
ANS Labeled Tape	VH.16.02	132
General Format and Sentinels	VH.16.02	132
Generalized Disk Address	VH.17	135
Consecutive File Format	VH.18	136
Schema for Locating and Referencing Public Files	VH.19	139
Schema for Locating and Referencing Private Files	VH.20	140
Read-Ahead Tables	VH.21	141
Symbionts and Cooperatives	VI.01	144
Symbiont/Cooperative File Block	VI.01	144
Symbiont Tables	VI.01	146
Linkage Between Logical Device Tables	VI.01	149
Input Symbiont Context Block	VI.01	152
Cooperative Context Block	VI.01	153
Output Symbiont Context Block	VI.01	154
Symbiont/Cooperative Context Block Displacement Symbols	VI.01	156
RBBAT Communication Buffers	VI.01	161
Multi-Batch Scheduler Tables	VI.02	171
MBS Tables in RBBAT	VI.02	171
Priority Tables	VI.02	171
Symbiont File Tables	VI.02	172
Serial Number Tables	VI.02	173

Section Page

MBS Tables in Core.. VI.02 174
Resource Allocation Tables................................ VI.02 174
Serial Number Tables for Running Jobs..................... VI.02 175
Partition Tables.. VI.02 176
Partition Table Layout.................................... VI.02 178
GI Tables... VI.02 179
Service Limit Tables...................................... VI.02 180
Peripheral Authorization Tables........................... VI.02 181

Error Log... VK 182
Error Log Format.. VK 182
File Control Pointers..................................... VK 183
Error Log Core Structure.................................. VK 184
Overview.. VK 186
Glossary.. VK 190
Error Log Codes... VK 200
Error Log Entry Formats................................... VK 208
CSE STOP Tables... VK.01 236

Assign/Merge Table.. VL 238
Function.. VL 238
Structure... VL 238
Format of Assign/Merge Table.............................. VL 238
Format of RUN Table....................................... VL.01 241

Error Codes and Error Messages............................ VM 247
Error Message File Format................................. VM 247
Error Messages.. VM 249

:USERS - Logon File....................................... VN.01 261
Structure... VN.01 262
Contents.. VN.01 264
:ACCTLG - Accounting Log Format........................... VN.02 268
Structure... VN.02 269
Contents.. VN.02 270
RATE Record... VN.03 273
Structure... VN.03 273
:RBLOG - Remote Batch Logon File.......................... VN.04 274
Purpose... VN.04 274
Contents.. VN.04 275
Structure of the RBID Record.............................. VN.04 280
:PROCS - Restricted Processor List........................ VN.05 281

	Section	Page
Loader Tables..	VO	283
Task Control Block (TCB)......................................	VO	283
DCB Name Table (DCBTAB).......................................	VO	283
Load Module Layout..	VO	284
HEAD Record...	VO	285
TREE Record...	VO	287
Segment Components..	VO	288
Libraries...	VO	289
REF/BREF Tables...	VO	291
REF/DEF and Expression Stack..................................	VO	292
Relocation Dictionary...	VO	294
LOCCT Table...	VO	297
Real Time..	VP	299
Interrupt Control Block Formats...............................	VP	299
ICB for User Task..	VP	299
ICB for Ghost Job..	VP	300
ICB for CLOCK3...	VP	301
ICB Field Descriptions.......................................	VP	302
ICB Chain Headers..	VP	306
DO List Block Formats...	VP	306
WAIT List Block Format..	VP	308
M:FRGD Module...	VP	309
M:IMC Modifications...	VP	311
ENQ/DEQ..	VQ	313
Description...	VQ	313
Queue Head Entries..	VQ	313
SQ Entries..	VQ	315
QECB Entries..	VQ	316
QNAME Entries...	VQ	316
Remote Processing..	VR	318
Remote Processing Tables in Core..............................	VR	318
Remote Processing Tables in RBBAT.............................	VR	319
RB:FLAG Structure...	VR	320
IRBT and 2780 Context...	VR	321
Remote Processing Handler Structure...........................	VR	324
Screech Codes..	VS	325

	Section	Page
Transaction Processing	VT	347
Physical Work Pages (PWPTABLE)	VT	347
TTP Resident Table - Queue Management	VT	348
Recovery Tables	VU	351
Core Dump Format	VU	351
Dump Tape Format	VU	351
Recovery Buffer	VU	352
Power Fail-Safe Interrupt Status	VU	353
Index		354

```
************************************************************************

     NAME:    JIT                                                                      JIT
     USAGE:   THE LABELS OF THE ITEMS IN JIT ARE CONSTRUCTED                           JIT
              USING THE FOLLOWING CONVENTIONS.  THESE CONVENTIONS                      JIT
              ALLOW THE USER TO DETERMINE FROM THE LABEL ITSELF                        JIT
              WHETHER IT IS AN ADDRESS OR A DISPLACEMENT WITHIN                        JIT
              JIT AND WHETHER IT HAS BYTE, HALFWORD OR WORD                            JIT
              RESOLUTION.  ITEMS IN JIT SHOULD ALWAYS BE                               JIT
              REFERENCED BY LABEL RATHER THAN ABSOLUTELY                               JIT
              BECAUSE THE INTERNAL STRUCTURE OF JIT MAY CHANGE.                        JIT
     JIT LABELING CONVENTIONS:  ALL THE LABELS DEFINED IN JIT                          JIT
               ARE CONSTRUCTED USING THE FOLLOWING                                     JIT
              CONVENTIONS.  THESE CONVENTIONS ALLOW THE USER                           JIT
              TO DETERMINE WHETHER THE LABEL IS AN ADDRESS                             JIT
              OR A DISPLACEMENT FROM J:JIT AND ITS RESOLUTION FROM                     JIT
              THE LABEL ITSELF.                                                        JIT
                                                                                       JIT
              1.)  A COLON (:) INDICATES THAT THE LABEL IS AN                          JIT
                   ADDRESS.                                                            JIT
                   EXAMPLE: J:TCB, JH:PC, JB:VLH                                       JIT
              2.)  NO COLON INDICATES THAT THE LABEL IS A                              JIT
                   DISPLACEMENT FROM J:JIT.                                            JIT
                   EXAMPLE: JTCB, PRDCRM, JBVLH                                        JIT
              3.)  IF THE LABEL STARTS WITH JB:, JB OR BA, IT HAS                      JIT
                   JB OR BA, IT HAS BYTE RESOLUTION.  IF IT STARTS                     JIT
                   WITH JX: OR JX, IT MAY BE EITHER BYTE OR HALF-                      JIT
                   WORD, DEPENDING ON VALUE ASSIGNED BY :BIG DURING                    JIT
                   SYSGEN.  ANY OTHER CHARACTERS INDICATE WORD                         JIT
                   RESOLUTION.                                                         JIT
                   EXAMPLES: WORD - J:UNAME, JUNAME, UNAME                             JIT
                             HALFWORD - JH:PC, JHDA                                    JIT
                             BYTE - JB:VLH, JBVLH, BAABC                               JIT
              4.)  IN SOME CASES SEVERAL LABELS WILL REFERENCE THE                     JIT
                   SAME ITEM GIVING ADDRESS AND DISPLACEMENT WITH                      JIT
                   DIFFERENT RESOLUTION FOR THE SAME ITEM.                             JIT
```

```
                   EXAMPLE: THE I/O ABORT CODE - WORD 17, BYTE 0 -                      JIT
                             J:ABC, JABC, ABC AND BAABC                                 JIT
     DESCRIPTION:    EACH USER RECEIVES AN INITIALIZED JIT WHEN THE                     JIT
              JOB OR TERMINAL SESSION BEGINS.  THIS JIT STAYS WITH                      JIT
              THE JOB UNTIL IT IS LOGGED OFF.  THE JIT IS THE FIRST                     JIT
              PAGE OF A JOB THAT IS SWAPPED IN AND THE LAST TO BE                       JIT
              SWAPPED OUT.                                                              JIT
                                                                                        JIT
              A USER'S JIT CONTAINS HIS ACCOUNTING DATA, RESOURCE                       JIT
              USAGE LIMITATIONS, VARIOUS FLAGS DESCRIBING THE STATUS                    JIT
              OF HIS JOB, SOME LOADER DATA, THE M:UC AND M:XX DCBS,                     JIT
              MEMORY MANAGEMENT DATA, A TEMP STACK FOR MONITOR USE,                     JIT
              POINTERS AND ADDRESSES OF DATA IN HIS CONTEXT BLOCK,                      JIT
              HIS MAP AND ACCESS CODE IMAGES AND SWAPPER DATA, AS                       JIT
              WELL AS MANY OTHER ITEMS TOO NUMEROUS TO MENTION.                         JIT
              THE JIT IS 512 WORDS LONG (1 PAGE) AND IS ALWAYS                          JIT
              LOADED AT .8C00 (VIRTUAL).                                                JIT
                                                                                        JIT
              THE SEEK ADDRESSES (JH:DA) AND THE COMMAND LIST (J:CL)                    JIT
              USED BY THE SWAPPER TO SWAP A USER IN OR OUT ARE                          JIT
              CONTAINED IN JIT.  THERE IS ENOUGH SPACE IN JIT TO                        JIT
              CONTAIN THIS DATA FOR A USER WHOSE SIZE IS NO GREATER                     JIT
              THAN 20 PAGES ON SIGMA 7 AND SMALL MEMORY SIGMA 9/560                     JIT
              SYSTEMS.  IF A USER'S SIZE EXCEEDS 20 PAGES, HE IS                        JIT
              ALLOCATED AN AJIT, "ADDITIONAL JIT", AND THE SWAPPER                      JIT
              COMMAND LIST IS MOVED INTO AJIT.  THE SPACE IN JIT                        JIT
              THAT WAS FORMERLY USED FOR THE COMMAND LIST IS THEN                       JIT
              USED FOR THE SEEK ADDRESSES, I.E., JH:DA SPILLS OVER                      JIT
              INTO J:CL.  ON LARGE MEMORY SYSTEMS (GREATER                              JIT
              THAN 128K), ALL USERS RECEIVE AN AJIT AT THE TIME                         JIT
              THEY RECEIVE THE JIT.  THE AJIT CONTAINS BOTH THE                         JIT
              SWAPPER COMMAND LIST AND THE SEEK ADDRESS TABLE,                          JIT
              JH:DA.                                                                    JIT
```

002

```
DOOJIT     8C00          *** CPV DOO JIT ***                                                       3
           0  1 2 3 4            16                   31                  X                JIT    8C00
           ----------------------------------------------                                  JIT    8C00
J:JIT, JIT \ \ \ \ \ \           \              SYSID  \  SYSID                            JIT    8C00
J:UN       \ \ \ \ \ \           \                     \                                   JIT    8C00
           ----------------------------------------------                                  JIT    8C00
            \ \ \ \ \                                                                      JIT    8C00
            \ \ \ \ - COMMAND PROC, EXEC,                                 X                JIT    8C00
            \ \ \ --- SUB TASK                                            X                JIT    8C00
            \ \ ----- REMOTE ASSIST                                       X                JIT    8C00
            \  ------ GHOST                                                                JIT    8C00
            --------- ON-LINE                                                              JIT    8C00
           ----------------------------------------------                                  JIT    8C01
J:ACCN     \          ACCOUNT NUMBER-- 2 WORDS             \  JACCN                        JIT    8C01
           ----------------------------------------------     ACCN                         JIT    8C01
           \                                               \                               JIT    8C01
           ----------------------------------------------                                  JIT    8C01
           ----------------------------------------------                                  JIT    8C03
J:UNAME    \          USER NAME-- 3 WORDS                  \  JUNAME                       JIT    8C03
           ----------------------------------------------     UNAME                        JIT    8C03
           \                                               \                               JIT    8C03
           ----------------------------------------------                                  JIT    8C03
           \                                               \                               JIT    8C03
           ----------------------------------------------                                  JIT    8C03
           ----------------------------------------------                                  JIT    8C06
J:CTIME    \ # OF I/O OPS * SL:IOTA IN CURRENT QUANTUM \  CEXT                             JIT    8C06
           ----------------------------------------------                                  JIT    8C06
           ----------------------------------------------                                  JIT    8C07
J:OVHTIM   \          CURRENT OVERHEAD TIME                \  OVHTIME                      JIT    8C07
           ----------------------------------------------                                  JIT    8C07
           ----------------------------------------------                                  JIT    8C08
J:CALCNT   \          COUNT OF CAL1 CALLS                  \  IOTIME                       JIT    8C08
           ----------------------------------------------                                  JIT    8C08
```

```
D00JIT    8C09           *** CPV D00 JIT ***                                                       4
            ---------------------------------------------                                   JIT     8C09
J:PTIME     \          PROCESSOR EXECUTION TIME         \   TPEXT                           JIT     8C09
            ---------------------------------------------                                   JIT     8C09
            ---------------------------------------------                                   JIT     8C0A
            \          PROCESSOR OVERHEAD TIME          \  TPOVT                            JIT     8C0A
            ---------------------------------------------                                   JIT     8C0A
            ---------------------------------------------                                   JIT     8C0B
            \  PROCESSOR PAGE * TIME MEMORY USE FACTOR  \  TPIOT                            JIT     8C0B
            ---------------------------------------------                                   JIT     8C0B
            ---------------------------------------------                                   JIT     8C0C
J:UTIME     \          USER EXECUTION TIME              \  TUEXT                            JIT     8C0C
            ---------------------------------------------                                   JIT     8C0C
            ---------------------------------------------                                   JIT     8C0D
            \          USER OVERHEAD TIME               \  TUOVT                            JIT     8C0D
            ---------------------------------------------                                   JIT     8C0D
            ----------------------------------------------                                  JIT     8C0E
            \  USER PAGE * TIME MEMORY USE FACTOR       \  TUIOT                            JIT     8C0E
            ---------------------------------------------                                   JIT     8C0E
            ---------------------------------------------                                   JIT     8C0F
J:DELTAT    \  -- TIME QUANTUM LEFT (INC BY CLOCK4)     \  TIMTMP                           JIT     8C0F
            ---------------------------------------------                                   JIT     8C0F
            ---------------------------------------------                                   JIT     8C10
J:MRT       \  MAX RUN TIME, IF ZERO, NO MAX            \  MRT                              JIT     8C10
            ---------------------------------------------                                   JIT     8C10
            0          8     11              23   26      31                                JIT     8C11
            ---------------------------------------------                                   JIT     8C11
J:ABC       \ ABORT    \ JOB    \            \ \ \ SENSE \  JABC,PRT                        JIT     8C11
            \  CODE    \PRIOR   \            \ \ \SWITCHS\  SS,BAABC                        JIT     8C11
            ---------------------------------------------                                   JIT     8C11
                                             \                                              JIT     8C11
                                             -ENQ'S OUTSTANDING                             JIT     8C11
            0          8   9 10 14 161718 20 24         31                 X                JIT     8C12
            ---------------------------------------------                                   JIT     8C12
J:RNST      \RUN       \  \ \ RUN \  \ \ \\X \ LINK     \   RUNFLAG                         JIT     8C12
            \STATUS    \  \ \ FLAG\  \ \ \\S \          \  JRNST,PUF                        JIT     8C12
            \          \  \ \     \  \ \ \\L \COUNTER   \ CCBEF,RNST                        JIT     8C12
            ---------------------------------------------  BARNST                           JIT     8C12
```

```
DOOJIT    8C12           *** CPV DOO JIT ***                                                        5
                        \  \       \ \ \   \                                                 JIT     8C12
                        \  \       \ \ \   - EXEC. SEV. LEVEL                                JIT     8C12
                        \  \       \ \ - RETURN M:EXIT AFTER M:LINK         X                JIT     8C12
                        \  \       \ - RETURN M:ERR/M:XXX AFTER M:LINK      X                JIT     8C12
                        \  \       - SAVE/RES J:CCBUF ON M:LINK/M:LDTRC     X                JIT     8C12
                        \  -------------COMMAND IN J:CCBUF                                   JIT     8C12
                        ----------------CONTROL CMND BUF FULL                                JIT     8C12
            0                   15                      31                                   JIT     8C13
            ----------------------------------------------                                   JIT     8C13
J:NRS       \ CARD INPUT COUNT  \ # OF SWAPS            \ CIC,NSWAPS                         JIT     8C13
            ----------------------------------------------  HANSWAPS                         JIT     8C13
            0  1 2 3 4 5 6 7 8 9      13 1415           31                                   JIT     8C14
            ----------------------------------------------                                   JIT     8C14
J:ASSIGN    \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ LIMIT       \  JASSIGN                           JIT     8C14
J:EXLY      \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ EXCEEDED FLG\  CPE,PPMD                          JIT     8C14
            ----------------------------------------------                                   JIT     8C14
             \ \ \ \ \ \ \ \ \ \ \ \ \ \ \                                                   JIT     8C14
             \ \ \ \ \ \ \ \ \ \       \ -USER PMDS                                          JIT     8C14
             \ \ \ \ \ \ \ \ \ \       --PROCESSOR PMDS                                      JIT     8C14
             \ \ \ \ \ \ \ \ \ -----------QUE MANAGER CALLED                                 JIT     8C14
             \ \ \ \ \ \ \ \ ------------RESERVED                                            JIT     8C14
             \ \ \ \ \ \ \ --------------RESTRICTED PROC LIST                                JIT     8C14
             \ \ \ \ \ \ ----------------:SYS RESIDENT                                       JIT     8C14
             \ \ \ \ \ ------------------EXECUTE ONLY                                        JIT     8C14
             \ \ \ \ --------------------FLUSH, BAD CONTROL CMND                             JIT     8C14
             \ \ \ ----------------------READ ACCT NONE                                      JIT     8C14
             \ \ ------------------------SWAP PURE PROCEDURE                                 JIT     8C14
             \ --------------------------NO BUFFER CHECK                                     JIT     8C14
             ----------------------------ASSIGNS HAVE BEEN MERGED                            JIT     8C14
            0                   15                      31                                   JIT     8C15
            ----------------------------------------------                                   JIT     8C15
            \ CARD PNCH COUNT   \ ERROR      : ERROR    \                                    JIT     8C15
            \                   \OVRIDE ADDR :SUBCODE   \  CPU,ERO                           JIT     8C15
            ----------------------------------------------                                   JIT     8C15
```

005

```
D00JIT    8C16          *** CPV D00 JIT ***                                                     6
            --------------------------------------------                                 JIT     8C16
            \ MAX PUNCH LIMIT  \ ABNORMAL ADDRESS      \ MPO,ABO                         JIT     8C16
            --------------------------------------------                                 JIT     8C16
            --------------------------------------------                                 JIT     8C17
J:CPPO      \CURRENT PROCESSOR \ FILE EXTENSION BITS   \ CPPO                            JIT     8C17
            \PAGE OUT COUNT    \                       \ JCPPO                           JIT     8C17
            --------------------------------------------                                 JIT     8C17
            --------------------------------------------                                 JIT     8C18
J:TRAP      \ MAX PROC PAGES   \ CC OF LAST \ LAST TRAP\ MPPO                            JIT     8C18
            \      OUT         \    TRAP    \EXECUTED  \                                 JIT     8C18
            --------------------------------------------                                 JIT     8C18
            --------------------------------------------                                 JIT     8C19
            \USER PAGE COUNT   \ADDRESS OF CCI'S LOADER\ ALOCCT                          JIT     8C19
            \                  \COMMAND TABLE          \ CUPO                            JIT     8C19
            --------------------------------------------                                 JIT     8C19
            --------------------------------------------                                 JIT     8C1A
J:JIP       \MAX USER LP PAGES \ JOB IN PROGRESS FLAG  \ MUPO                            JIT     8C1A
            --------------------------------------------                                 JIT     8C1A
            --------------------------------------------                                 JIT     8C1B
J:INTER     \DIAGNOSTIC PAGE CNT\ # OF INTERACTIONS    \ CDPO                            JIT     8C1B
            --------------------------------------------                                 JIT     8C1B
            --------------------------------------------                                 JIT     8C1C
J:RWECB     \ MAX DO PAGES OUT \ ECB ADDRESS           \ MDPO                            JIT     8C1C
            --------------------------------------------                                 JIT     8C1C
            --------------------------------------------                                 JIT     8C1D
JB:STEP     \# OF STEPS\JOB ORIGIN\# TAPE READ & WRITES \ TPACCESS                       JIT     8C1D
JB:ORG      \          \          \                    \                                 JIT     8C1D
            --------------------------------------------                                 JIT     8C1D
            --------------------------------------------                                 JIT     8C1E
J:ASPIN     \          ACTIVE SPINDLES                 \                                 JIT     8C1E
            --------------------------------------------                                 JIT     8C1E
            \                                          \                                 JIT     8C1E
            --------------------------------------------                                 JIT     8C1E
```

```
000JIT    8C20          *** CPV D00 JIT ***                                                  7
                -----------------------------------------------                         JIT      8C20
J:ALB           \  UNUSED              \   CAL3  *J:ALB        \                  X     JIT      8C20
                -----------------------------------------------                         JIT      8C20
                -----------------------------------------------  CCLTFLGS               JIT      8C21
J:TELFLGS       \                                              \  CCLFLAGS               JIT      8C21
                \    CCI AND TEL FLAGS (SEE DEF)               \  MJCFLG                 JIT      8C21
                -----------------------------------------------  JTELFLGS               JIT      8C21
                0          8                     25          31                         JIT      8C22
                -----------------------------------------------                         JIT      8C22
J:CASSIN        \ REMOTE    \  REMOTE   \UNUSED     \ FLAGS    \  ERLFLAGS        X     JIT      8C22
                \BATCH FLAG\ BATCH ID \            \           \  ERRLFLGS               JIT      8C22
                -----------------------------------------------  JRBID                  JIT      8C22
                0                  15                        31                         JIT      8C23
                -----------------------------------------------                         JIT      8C23
J:INTENT        \ \0---------------0 \ USER CONSOLE INTERUPT \  INTENT                 JIT      8C23
                \ \                  \ ENTRY ADDRESS           \                         JIT      8C23
                -----------------------------------------------                         JIT      8C23
                 \                                                                      JIT      8C23
                 \--  COMMAND PROCESSOR BREAK CONTROL                                   JIT      8C23
                -----------------------------------------------                         JIT      8C24
J:TIMENT        \0------------------0\  ENTRY ADDRESS FOR      \  TIMENT                 JIT      8C24
                \                    \  M:STIMER               \                         JIT      8C24
                -----------------------------------------------                         JIT      8C24
                -----------------------------------------------                         JIT      8C25
J:UTIMER        \   INTERVAL SET BY M:STIMER                   \  UTIMER                 JIT      8C25
                -----------------------------------------------                         JIT      8C25
                    7 8 910 1112 14 15                       31                         JIT      8C26
                -----------------------------------------------                         JIT      8C26
                \  \ TRAP FLAGS    \                           \                         JIT      8C26
J:USENT         \  \ \ \ \ \ \ \ \ \ \     ADD.  OF USER TRAP  \                         JIT      8C26
                \  \ \ \ \ \ \ \ \ \ \   \ ROUTINES BY M:TRAP  \  TRPFLAGS               JIT      8C26
                -----------------------------------------------                         JIT      8C26
                    \ \ \ \ \ \ \ \                                                     JIT      8C26
                    \ \ \ \ \ \ \ --BAD CAL TRAP CONTROL                                JIT      8C26
                    \ \ \ \ \ \ ----FIXED POINT TRAP CONTROL                            JIT      8C26
                    \ \ \ \ \ ------DECIMAL TRAP CONTROL                                JIT      8C26
                    \ \ \ \ --------FLOATING POINT TRAP CONTROL                         JIT      8C26
```

```
DOOJIT    8C26          *** CPV DOO JIT ***                                                       8
                \ \ \ ----------STACK TRAP CONTROL                                        JIT     8C26
                \ \ ------------UNIMPLEMENTED INSTR TRAP CONTROL                          JIT     8C26
                \ --------------NON-ALLOWED OPERATION TRAP CNTRL                          JIT     8C26
                ----------------RSVRD, - TRAP CONTROL                                     JIT     8C26
            ---------------------------------------------                                 JIT     8C27
J:TCB       \ 0---------------0 \  TCB ADDRESS         \  TCBADR                          JIT     8C27
            ---------------------------------------------                                 JIT     8C27
            ---------------------------------------------                                 JIT     8C28
J:TREE      \ 0---------------0 \ TREE TABLE ADDRESS    \  JITREE                         JIT     8C28
            ---------------------------------------------                                 JIT     8C28
            ---------------------------------------------                                 JIT     8C29
            \ MIN TEMP PACK SPACE REMAINING             \  TMPDPPK                        JIT     8C29
            ---------------------------------------------                                 JIT     8C29
            ---------------------------------------------                                 JIT     8C2A
J:USCDX     \ 0---------------0 \ ADDR CONTEXT DATA     \                                 JIT     8C2A
            \                   \   BUFFER CHAIN        \                                 JIT     8C2A
            ---------------------------------------------                                 JIT     8C2A
            ---------------------------------------------                                 JIT     8C2B
J:DCBLINK   \ 0----------------0 \ADDR REST OF DCB NAMES\  DCBLINK                        JIT     8C2B
            ---------------------------------------------                                 JIT     8C2B
            ---------------------------------------------                                 JIT     8C2C
M:UC        \                                           \                                 JIT     8C2C
J:COCOPT    :  COC OPTIONS IN M:UC+8                    :                 X               JIT     8C2C
            ---------------------------------------------                                 JIT     8C2C
            ---------------------------------------------                                 JIT     8C2C
J:TITLE     \ IF BATCH, JOB TITLE IN TEXTC (21 WORDS)   \  ATITLE                         JIT     8C2C
            \ IF ONLINE, M:UC (22 WORDS)                \                                 JIT     8C2C
            \  IF SUB TASK, CONTROL INFO(SEE DEF)(RSRVD) \                X               JIT     8C2C
            ---------------------------------------------                                 JIT     8C2C
            0                                          31                 X               JIT     8C42
            ---------------------------------------------                 X               JIT     8C42
J:UPRIV     \  USER PRIVILEGE FLAGS                     \                 X               JIT     8C42
            ---------------------------------------------                 X               JIT     8C42
            ---------------------------------------------                                 JIT     8C42
J:XP        \ DEFAULT FILE EXPIRE,\ MAX FILE EXPIRATION \                                 JIT     8C42
            ---------------------------------------------                                 JIT     8C42
```

```
D00JIT    8C43          *** CPV D00 JIT ***                                                       9
            0          8          16         24         31                                 JIT     8C43
            ---------------------------------------------                                  JIT     8C43
JB:PRIV     \   JOB    \ CUL-(CLL \ RESERVED \ NFPOOL   \  JBNFPOOL                        JIT     8C43
JB:NFPOOL   \PRIVILEGE \       +2)\          \          \                                  JIT     8C43
            \          \  IN PAGES\          \          \                                  JIT     8C43
            ---------------------------------------------                                  JIT     8C43
            0                  15                       31                                 JIT     8C44
            ---------------------------------------------                                  JIT     8C44
J:ABUF      \ 0----------------0\ADR OF ASSIGN-MERGE BUF\                                  JIT     8C44
            ---------------------------------------------                                  JIT     8C44
            0          8          16         24         31                                 JIT     8C45
JB:CCARS    ---------------------------------------------                                  JIT     8C45
JB:DISP     \RECORD SZ \DISPLACE. \ CURRENT  \SPEC SHRD \                                  JIT     8C45
JB:CUN      \ C.C.     \ IN C.C.  \ USER #   \PROC. #   \                                  JIT     8C45
JB:OTEL     ---------------------------------------------                                  JIT     8C45
JB:LPP      ---------------------------------------------                                  JIT     8C46
JB:LC       \# OF LINES\CURRENT # \ PLATEN   \ PROMPT   \                                  JIT     8C46
JB:PCW      \PER PAGE  \OF LINES  \WIDTH     \  BYTE    \                                  JIT     8C46
JB:PROMPT   \DEF=54    \ DEF=49   \ DEF=72   \ DEF='>'  \                                  JIT     8C46
            ---------------------------------------------                                  JIT     8C46
            ---------------------------------------------                                  JIT     8C47
J:IDELTAT   \    INITIAL VALUE OF J:DELTAT              \                                  JIT     8C47
            ---------------------------------------------                                  JIT     8C47
             0 1 2 3 4 5 6 7 8 15                       31                                 JIT     8C48
            ---------------------------------------------                                  JIT     8C48
J:EXTENT    \ \ \ \ \ \ \ \ \ \ \ ADDR OF EXIT CONTROL  \                                  JIT     8C48
            ---------------------------------------------                                  JIT     8C48
             \ \ \ \ \ \ \ \ \                                                             JIT     8C48
             \ \ \ \ \ \ \ \ --'LAST' OP ON EXIT CNTRL                                     JIT     8C48
             \ \ \ \ \ \ \ ----STEP CONDITION CODE                                         JIT     8C48
             \ \ \ \ \ \ ------EXIT CNTRL;C-Y AND QUIT                                     JIT     8C48
             \ \ \ \ \ --------UNUSED                                                      JIT     8C48
             \ \ \ \ ----------M:LINK OR M:LDTRC EXIT CONDITION                            JIT     8C48
             \ \ \ ------------EXIT CNTRL BY CMND PROC                                     JIT     8C48
             \ \ --------------EXIT CNTRL IN PROGRESS                                      JIT     8C48
             \ ----------------SOME LIMIT EXCEEDED                                         JIT     8C48
             ------------------OPERATOR ABORT OR LINE HNGUP                                JIT     8C48
```

```
DOOJIT    8C49          *** CPV D00 JIT ***                                                  10
            ----------------------------------------------                                 JIT     8C49
            \ MIN TEMP RAD SPACE REMAINING               \  TMPDCPK                         JIT     8C49
            ----------------------------------------------                                 JIT     8C49
            ----------------------------------------------                X                JIT     8C49
            \          OLD PSD                           \                X                JIT     8C49
J:XPSD      \                                            \                X                JIT     8C49
            ----------------------------------------------                X                JIT     8C49
            \          CAL3 HANDLER PSD                  \                X                JIT     8C49
            \                                            \                X                JIT     8C49
            ----------------------------------------------                X                JIT     8C49
            ----------------------------------------------                                 JIT     8C4E
TSTACK      \  TEMPORARY STACK DOUBLEWORD (BOUND 8)      \                                 JIT     8C4E
UTS         ----------------------------------------------                                 JIT     8C4E
            \                                            \                                 JIT     8C4E
            ----------------------------------------------                                 JIT     8C4E
            \  TEMPORARY STACK OF SIZE=JTSTACKSZ=121     \                                 JIT     8C4E
            !                                            !                                 JIT     8C4E
            !                                            !                                 JIT     8C4E
            ----------------------------------------------                                 JIT     8C4E
            ----------------------------------------------                                 JIT     8CC9
J:OVRLY     \ OVERLAY ENTRY POINT ADDRESS                \                                 JIT     8CC9
            ----------------------------------------------                                 JIT     8CC9
            0          8          16         24          31                                JIT     8CCA
            ----------------------------------------------                                 JIT     8CCA
J:CPROCS    \ UB:APR   \ UB:APO   \ UB:ASP   \ UB:DB     \                                 JIT     8CCA
            ----------------------------------------------                                 JIT     8CCA
            0                     16                     31                                JIT     8CCB
            ----------------------------------------------                                 JIT     8CCB
J:CFLGS     \  GARBAGE            \  UH:FLG    AT SAVE   \                X                JIT     8CCB
            ----------------------------------------------                                 JIT     8CCB
            ----------------------------------------------                                 JIT     8CCC
J:CCBUF     \          CONTROL COMMAND BUFFER            \                                 JIT     8CCC
            !          FOR CCI AND TEL                   !                                 JIT     8CCC
            !          20 WORDS FOR TEL                  !                                 JIT     8CCC
            !          30 WORDS FOR CCI                  !                                 JIT     8CCC
            !                                            !                                 JIT     8CCC
            !                                            !                                 JIT     8CCC
```

```
DOOJIT    8CCC           *** CPV D00 JIT ***                                                  11
J:DWSK(20)  :                                         :                                 JIT    8CCC
            :                                         :                                 JIT    8CCC
            :                                         :(24)JOPT                         JIT    8CCC
J:LMN(26)   : LAST LM FORMED = 3 WORDS, TEXTC         :                                 JIT    8CCC
J:START(29) :  START ADDRESS OF CURRENT PROGRAM       :                                 JIT    8CCC
            -----------------------------------------                                   JIT    8CCC
            -----------------------------------------                                   JIT    8CEA
M:XX        \         M:XX DCB                        \                                 JIT    8CEA
            :         22 WORDS                        :                                 JIT    8CEA
            :                                         :                                 JIT    8CEA
            -----------------------------------------                                   JIT    8CEA
MXFPL       :         M:XX FUNCTION PARAMETER LIST    :                                 JIT    8CEA
            :         10 WORDS                        :                                 JIT    8CEA
            :                                         :                                 JIT    8CEA
            -----------------------------------------                                   JIT    8CEA
MXKB        :         M:XX KEY BUFFER                 :                                 JIT    8CEA
            :         8 WORDS                         :                                 JIT    8CEA
            :                                         :                                 JIT    8CEA
            -----------------------------------------                                   JIT    8CEA
            0                  24                     31                                JIT    8D12
            -----------------------------------------                                   JIT    8D12
J:BUP       \ 0---------------0 \BEGIN USER PAGE #    \  JBUP                           JIT    8D12
            -----------------------------------------                                   JIT    8D12
            -----------------------------------------                                   JIT    8D13
J:EUP       \ 0---------------0 \ENDING USER PAGE #   \  JEUP                           JIT    8D13
            -----------------------------------------                                   JIT    8D13
            -----------------------------------------                                   JIT    8D14
J:PLL       \ 0---------------0 \PAGE # PROG LOWER LIMIT\  JPLL                         JIT    8D14
            -----------------------------------------                                   JIT    8D14
            -----------------------------------------                                   JIT    8D14
J:PUL       \ 0---------------0 \PAGE # PROG UPPER LIMIT\  JPUL                         JIT    8D14
            -----------------------------------------                                   JIT    8D14
```

```
D00JIT    8D16           *** CPV D00 JIT ***                                                        12
            ---------------------------------------------                                   JIT     8D16
J:DLL       \ 0---------------0 \PAGE # DATA LOWER LIMIT\   JDLL                            JIT     8D16
            ---------------------------------------------                                   JIT     8D16
            ---------------------------------------------                                   JIT     8D17
J:DUL       \ 0---------------0 \PAGE # DATA UPPER LIMIT\ JDUL                              JIT     8D17
            ---------------------------------------------                                   JIT     8D17
            ---------------------------------------------                                   JIT     8D18
J:DDLL      \ 0---------------0 \PAGE # DYNAMIC DATA    \  JDDLL                            JIT     8D18
            \                   \  LOWER LIMIT          \                                   JIT     8D18
            ---------------------------------------------                                   JIT     8D18
            ---------------------------------------------                                   JIT     8D19
J:DDUL      \ 0---------------0 \ PAGE # DYNAMIC DATA   \  JDDUL                            JIT     8D19
            \                   \  UPPER LIMIT          \                                   JIT     8D19
            ---------------------------------------------                                   JIT     8D19
            ---------------------------------------------                                   JIT     8D1A
J:CLL       \ 0---------------0 \PAGE # CONTEXT LOW LIMIT\                                  JIT     8D1A
            ----------------------------------------------                                  JIT     8D1A
            ---------------------------------------------                                   JIT     8D1B
            \ 0---------------0 \PAGE # CONTEXT UP LIMIT\                                   JIT     8D1B
            ---------------------------------------------                                   JIT     8D1B
            ---------------------------------------------                                   JIT     8D1C
J:DCBLL     \ 0---------------0 \ PAGE # DCB LOW LIMIT  \  JDCBLL                           JIT     8D1C
            ---------------------------------------------                                   JIT     8D1C
            ---------------------------------------------                                   JIT     8D1D
J:DCBUL     \ 0------------0    \PAGE # DCB UPPER LIMIT \  JDCBUL                           JIT     8D1D
            ---------------------------------------------                                   JIT     8D1D
JB:PCP      ---------------------------------------------  JPCP                             JIT     8D1E
JB:PCD      \PG CNT OF \PG CNT OF \PG CNT OF \PG CNT OF \  JBPCP                            JIT     8D1E
JB:PCDD     \PROCEDURE \  DATA    \DYNM DATA \ CONTEXT  \  JBPCDD                           JIT     8D1E
JB:PCC      ---------------------------------------------  JBPCC                            JIT     8D1E
            ----------------------------------------------                                  JIT     8D1F
JB:PCDCR    \PG CNT OF \SPARE BUF \ PG CNT   \FILE MNGT \                                   JIT     8D1F
JB:PCPWP    \   DCBS   \SWAP FLAG \ ASSIGNED \SPARE BUF \  JBPBFLG                          JIT     8D1F
JB:FBUC     \          \          \ PHYS,PGS \USE CNT   \  JBFBUC                           JIT     8D1F
            ---------------------------------------------                                   JIT     8D1F
```

```
D00JIT    8D20           *** CPV D00 JIT ***                                                     13
            ---------------------------------------------                                 JIT     8D20
JB:FBUL     \FILE MNGT \FILE MNGT*\ COOP     \ COOP     \  JBFRFP                         JIT     8D20
            \SPARE BUF \FREE BUF  \SPARE BUF \SPARE BUF \  JBCRLL                         JIT     8D20
JB:CBUC     \UPPER LIM \POOL HEAD \LOW LIMIT \USE CNT   \  JBCBUC                         JIT     8D20
            ---------------------------------------------                                 JIT     8D20
                     * INTERRUPT ALTERED                                                  JIT     8D20
            ---------------------------------------------  JBTDP                          JIT     8D21
JB:TDP      \PG # TOP  \PG # BOTTM\FINAL RUN \NEXT AVAIL\  JBBCP                          JIT     8D21
JB:BCP      \DYMN DATA \ COMMON   \ STATUS   \SECTOR POS\  JBNASP                         JIT     8D21
JB:FRS      ---------------------------------------------                                 JIT     8D21
            ---------------------------------------------                                 JIT     8D22
J:USER      \  2 WORDS FOR INSTALLATION USE             \                                 JIT     8D22
            ---------------------------------------------                                 JIT     8D22
            \                                           \                                 JIT     8D22
            ---------------------------------------------                                 JIT     8D22
            ---------------------------------------------                                 JIT     8D24
J:CLS       \  CLOSE STATUS INFO                        \                                 JIT     8D24
            ---------------------------------------------                                 JIT     8D24
            ---------------------------------------------                                 JIT     8D25
            \ NUMBER OF PACK READS AND WRITES           \  DPACCESS                       JIT     8D25
            ---------------------------------------------                                 JIT     8D25
            ---------------------------------------------                                 JIT     8D26
            \  NUMBER OF DISC READS AND WRITES          \  DCACCESS                       JIT     8D26
            ---------------------------------------------                                 JIT     8D26
            0          8          16         24         31                                JIT     8D27
            ---------------------------------------------                                 JIT     8D27
JB:MAX      \  MAXIMUM RESOURCES ALLOWED TO USER        \JBMNPA                           JIT     8D27
            ---------------------------------------------                                 JIT     8D27
JB:MNPA     \     4 WORD BYTE TABLE                     \                                 JIT     8D27
            ---------------------------------------------                                 JIT     8D27
            \                                           \                                 JIT     8D27
            ---------------------------------------------                                 JIT     8D27
            \                                           \                                 JIT     8D27
            ---------------------------------------------                                 JIT     8D27
```

```
D00JIT    8D2B           *** CPV D00 JIT ***                                                        14
            0                     16                    31                                   JIT     8D2B
            ---------------------------------------------                                    JIT     8D2B
            \ PHYSICAL PAGE       \  UNUSED             \  JXPPH            X                JIT     8D2B
JX:PPH      \  CHAIN HEAD         \                     \  JPPH                              JIT     8D2B
            ---------------------------------------------                                    JIT     8D2B
            0                     16         24         31                                   JIT     8D2C
            ---------------------------------------------                                    JIT     8D2C
JX:PPT      \ PHYSICAL PAGE       \ SLNK     \ XLNK     \  JPPT                              JIT     8D2C
JB:SLNK     \  CHAIN TAIL         \          \          \  JXPPT                             JIT     8D2C
JB:XLNK     ---------------------------------------------                                    JIT     8D2C
            0          8          16         24         31                                   JIT     8D2D
            ---------------------------------------------                                    JIT     8D2D
JB:PPC      \PHYS PAGE \PARTITION \# REMAIN  \UNUSED    \  JPPC             X                JIT     8D2D
JB:NRG      \CHAIN CNT \    #     \GRANULES  \          \  JBPPC                             JIT     8D2D
JB:PNR      ---------------------------------------------  JBNRG                             JIT     8D2D
            ---------------------------------------------                                    JIT     8D2E
JB:VLH      \VIRT. PG  \PEAK CORE \  ON-LINE PAGE       \  JBVLH            X                JIT     8D2E
JB:PEAK     \LINK HEAD \          \     COUNT           \  JVLH             X                JIT     8D2E
JH:PC       ---------------------------------------------                                    JIT     8D2E
JB:STEPCC   ---------------------------------------------                                    JIT     8D2F
JB:VLT      \VIRT PG   \STEP COND.\# DISC    \# TAPES   \  JVLT                              JIT     8D2F
JB:PMTS     \LINK TAIL \ CODES    \PACK MOUNT\ MOUNTS   \                                    JIT     8D2F
JB:TMTS     ---------------------------------------------                                    JIT     8D2F
            ---------------------------------------------                                    JIT     8D30
            \ PERMANENT RAD SPACE REMAINING             \  PRDCRM                            JIT     8D30
            ---------------------------------------------                                    JIT     8D30
            ---------------------------------------------                                    JIT     8D31
            \ PERMANENT PACK SPACE REMAINING            \  PRDPRM                            JIT     8D31
            ---------------------------------------------                                    JIT     8D31
            ---------------------------------------------                                    JIT     8D32
            \ TEMPORARY RAD SPACE REMAINING             \  TMDCRM                            JIT     8D32
            ---------------------------------------------                                    JIT     8D32
```

```
D00JIT    8D33          *** CPV D00 JIT ***                                                       15
          ---------------------------------------------                                   JIT    8D33
          \ TEMPORARY PACK SPACE REMAINING             \  TMDPRM                          JIT    8D33
          ---------------------------------------------                                   JIT    8D33
          ---------------------------------------------                                   JIT    8D34
J:VLCS    \   VIRTUAL LINK CHAIN STOP                  \  JVLCS                           JIT    8D34
          ---------------------------------------------                                   JIT    8D34
          ---------------------------------------------                                   JIT    8D35
J:AJ      \   ADDITIONAL JIT PHYSICAL PAGE NUMBER      \  JAJ                             JIT    8D35
          ---------------------------------------------                                   JIT    8D35
          ---------------------------------------------                                   JIT    8D36
J:CLPA    \   COMMAND LIST PHYSICAL ADDRESS            \   JCLPA                          JIT    8D36
          ---------------------------------------------                                   JIT    8D36
          ---------------------------------------------                                   JIT    8D37
J:CLE     \   NUMBER OF WORDS IN COMMAND LIST          \  JCLE                            JIT    8D37
          ---------------------------------------------                                   JIT    8D37
          ---------------------------------------------                                   JIT    8D38
          \  POINTER TO TRANSFER IN CHANNEL IN J:CL    \  JCLP                            JIT    8D38
          ---------------------------------------------                                   JIT    8D38
          ---------------------------------------------                                   JIT    8D39
          \ SAVED WORD OF COMMAND LIST WHERE TIC WENT\  JCLT                              JIT    8D39
          --------------------------------------------                                    JIT    8D39
          --------------------------------------------                                    JIT    8D3A
J:FDDA    \    FILE DIRECTORY DISC ADDRESS             \                                  JIT    8D3A
          ---------------------------------------------                                   JIT    8D3A
          0                     16                     31                                 JIT    8D3B
          ---------------------------------------------                                   JIT    8D3B
J:T       \READ COMP TIME       \ READ CURRENT TIME    \                                  JIT    8D3B
          ---------------------------------------------                                   JIT    8D3B
          ---------------------------------------------                                   JIT    8D3C
J:JAC     \  ACCESS CODES FOR USER                     \  JJAC                            JIT    8D3C
          !     2 BITS PER PAGE - 12 WORDS             !                                  JIT    8D3C
          !                                            !                                  JIT    8D3C
          ---------------------------------------------                                   JIT    8D3C
```

```
D00JIT    8D48         *** CPV D00 JIT ***                                            16
          -------------------------------------------                        JIT      8D48
J:STAR    \   FDA OF *B                                \                     JIT      8D48
          -------------------------------------------                        JIT      8D48
          \          *D                                \                     JIT      8D48
          -------------------------------------------                        JIT      8D48
          \          *G                                \                     JIT      8D48
          -------------------------------------------                        JIT      8D48
          \          *L                                \                     JIT      8D48
          -------------------------------------------                        JIT      8D48
          \          *T                                \                     JIT      8D48
          -------------------------------------------                        JIT      8D48
          \          *N (LNKTRC)                       \                     JIT      8D48
          -------------------------------------------                        JIT      8D48
          -------------------------------------------                        JIT      8D4E
J:BASE    \  SPILL BUFFER FOR INDEX BUFFERS.           \                     JIT      8D4E
          !  ALSO USED BY OTHER MONITOR ROUTINES       !                     JIT      8D4E
          !  AS TEMPORARY STORAGE.    12 WORDS         !                     JIT      8D4E
          -------------------------------------------                        JIT      8D4E
          0                    16                    31                      JIT      8D5A
          -------------------------------------------                        JIT      8D5A
J:TIC     \RESPONSE TIME,2MS OR 0\TURNAROUND TIME OR 0\                      JIT      8D5A
          -------------------------------------------                        JIT      8D5A
          -------------------------------------------                        JIT      8D5B
J:AMR     \ DISC ADDRESS OF ASSIGN-MERGE RECORD        \                     JIT      8D5B
          -------------------------------------------                        JIT      8D5B
          0                  15                      31                      JIT      8D5C
          -------------------------------------------                        JIT      8D5C
J:ICBHDR  \ \ \              \HEAD OF ACTIVE ICB CHAIN\                      JIT      8D5C
          --------------------------------------------                       JIT      8D5C
           \ \                                                               JIT      8D5C
           \ --M:IOEX CAL ISSUED                                             JIT      8D5C
           ----REAL TIME CAL ISSUED                                          JIT      8D5C
```

```
D00JIT    8D5D         *** CPV D00 JIT ***                                                    17
          -------------------------------------------           X                 JIT    8D5D
J:PPRIV   \          PROCESSOR PRIVILEGE FLAGS       \             X                 JIT    8D5D
          ------------------------------------------            X                 JIT    8D5D
          -------------------------------------------                             JIT    8D5E
JB:LMAP   \  VIRTUAL PAGE # CHAIN                     \  JBLMAP                    JIT    8D5E
          :      BYTE TABLE BY VIRTUAL PAGE #         :  JLMAP                     JIT    8D5E
          :      53 WORDS                             :                  X         JIT    8D5E
          -------------------------------------------                             JIT    8D5E
          -------------------------------------------                   X         JIT    8D63
JB:CUR    \  CURRENT RESOURCES ALLOCATED TO USER      \                  X         JIT    8D63
          -------------------------------------------                   X         JIT    8D63
          \      4 WORD BYTE TABLE                    \                  X         JIT    8D63
          -------------------------------------------                   X         JIT    8D63
          \                                           \                  X         JIT    8D63
          -------------------------------------------                   X         JIT    8D63
          \                                           \                  X         JIT    8D63
          -------------------------------------------                   X         JIT    8D63
          -------------------------------------------                             JIT    8D93
JX:CMAP   \  PHYSICAL PAGE # CHAIN                    \  JXCMAP                    JIT    8D93
          :      HALF-WORD OR BYTE                    \  JCMAP                     JIT    8D93
          :       INDEXED BY VIRTUAL PAGE #           :                            JIT    8D93
          :       53 WORDS OR 106 ON BIG 9            :                  X         JIT    8D93
          -------------------------------------------                             JIT    8D93
          0                                           31                 X         JIT    8D93
          -------------------------------------------                   X         JIT    8D93
J:LOCF    \          PERIPHERAL FLAGS                 \                  X         JIT    8D93
JH:LOCF   -------------------------------------------                   X         JIT    8D93
          -------------------------------------------                             JIT    8D93
JH:DA     \  SEEK ADDRESS USED BY J:CL                \  JDA                       JIT    8D93
          :       (SIGMA 7 AND SMALL SIGMA 9)         :                            JIT    8D93
          :       LOCATED AT END OF AJIT IF BIG SYSTEM:                            JIT    8D93
          -------------------------------------------                             JIT    8D93
          -------------------------------------------                             JIT    8D93
J:CL      \  COMMAND LIST USED BY SWAPPER             \  JCL                       JIT    8D93
          :                                           :                            JIT    8D93
          :          SEEK IOCD FA(JH:DA(0))           :                            JIT    8D93
          :          READ/WRITE IOCD                  :                            JIT    8D93
```

```
DOOJIT    8D93          *** CPV DOO JIT ***                                                             18
            !          READ/WRITE IOCD                  !                                   JIT    8D93
            !          READ/WRITE IOCD                  !                                   JIT    8D93
            !          READ/WRITE IOCD                  !                                   JIT    8D93
            !          SEEK BA(JH:DA(1))                !                                   JIT    8D93
            !          READ/WRITE IOCD                  !                                   JIT    8D93
            !            !     !                        !                                   JIT    8D93
            ---------------------------------------------                                   JIT    8D93
            ---------------------------------------------                                   JIT    8E00
J:AJIT      \ADDR OF AJIT IF J:CL TOO BIG FOR JIT       \                                   JIT    8E00
            ---------------------------------------------                                   JIT    8E00
```

LANGUAGE PROCESSOR CHART

		APL	BASIC	EASY	FLAG	FORIV	GPDS	LIB	META	SL/1	TEXT	SORT	EDMS	EDMS Re-structuring	IDP	MANAGE	COBOL	RPG	MERGE
ABSOLUTE	X'2B'	X				X						X	X						
ADDRESSES	X '4F'		X		X	X	X		X	X		X	X						
	JOPT								X										
	CCBUF				X														
	J:ACCN	X		X							X								
	J:CTIME																		
	J:DELTAT	X																	
	J:JIT	X		X							X								
	J:OPT					X										X			
	J:PTIME	X		X															
	J:UTIME	X																	
	J:UNAME										X								
	J:UTIME	X		X															
	JB:LC																		
	JCPPO				X								X						
	M:UC			X		X													
	MRT				X														
	J:CPROCS													X					
	J:IDELTAT	X																	
	J:OVHTIME			X															
	J:RNST			X															
	J:TCB			X															
	J:TELFLGS			X															

SCHEDULER QUEUES

SB:SWP

0 … 7
Ordered list of queues to be searched by swap scheduler

SW
SC10
STI
STOB
SQR
SQFI
SQA
SC9
SC8
SC7
SC6
SC5
SC4
SC3
SC2
SC1
SC0
SRT

SITUATIONAL PRIORITY INCREMENTS

SH:PINC

0 … 15	
Not Used	
3	Special Compute
2	I/O Complete
6	Interactive
4	Terminal Output Cont
3	Resource Unblock

Subtracted from user base priority (UB:PRIOB) to compute current priority (UB:PRIO).

EVENT TRANSITION TABLES

0	1 … 31
continue	Word Table of State Event Transitions Size is a Function of Number of Final States (Displaced by Event)

SB:SET

0 … 7
Destination State (in parallel with S:SET)

RESOURCE SUB-QUEUES FOR SQR/SQRO

RESOURCE INDEX ↓

	SB:RQ		
0	HEAD	R:SYMF	SYMFILE or SGCBUF
1	HEAD	R:SYMD	SYMBIONT DISC
2	HEAD	R:OCR	OPEN/CLOSE
3	HEAD	R:CBA	COC BUFFER
4	HEAD	R:DPA	SWAPPER PAGE
5	HEAD	R:QFAC	QUEUE FOR ALLOCAT
6	HEAD	R:NQW	ENQ WAIT

Chains for sub-states in SQR/SQRO are linked using U:MISC and UB:PRIO as shown.

0 … 7	8 … 23	24 … 31
Resource Index (SB:RQ)	0-----------------------0	FLINK

SB:IOTA - Cell containing I/O time allowance. Remaining quantum is decremented by this value for each I/O.

SL:OPC - Word containing monitor overlay protection counter value.

S:OPC - Word containing working value for overlay protection counter, decremented when unsuccessful at swap scheduling. Refreshed by SL:OPC when successful swap schedule.

SCHEDULER STATES

STATE	#	MEANING
SRT	1	Real Time Compute
SC0	2	Background Compute X'C0' ≤ IB:PRIOB ≤ X'F5'
SC1	3	Background Compute Priority = X'F6'
SC2	4	" " X'F7'
SC3	5	" " X'F8'
SC4	6	" " X'F9'
SC5	7	" " X'FA'
SC6	8	" " X'FB'
SC7	9	" " X'FC'
SC8	A	" " X'FD'
SC9	B	" " X'FE'
SC10	C	" " X'FF'
SCU	D	Current User
STOB	E	Terminal Output Blocked
STOBO	F	Terminal Output Blocked Out of Core
SIOW	10	I/O Wait
SIOMF	11	Master I/O Function Count Too High
SW	12	Wait (Asleep)
SQA	13	Queued for Access (To RBBAT)
SQR	14	Queued for Dynamic Resource
SQRO	15	Queued for Dynamic Resource Out of Core
STI	16	Terminal Inputting
STIO	17	Terminal Inputting Out of Core
SQFI	18	Queued for Real Time Interrupt
SNULL	19	Empty User Slot

SCHEDULER EVENTS

STATE	#	MEANING
E:IIP	0	I/O in Progress
E:QMF, E:IP	1	Queue for Master Function Count Too High
E:CRD	2	Terminal Read
E:CIC	3	Terminal Input Complete
E:CBL	5	Terminal Output Block
E:CUB	6	Terminal Output Unblock
E:CBK	8	User Hit Break
E:CEC	A	User Hit Control-Y
E:ERR	C	User to be Errored
E:ABRT, E:OFF	E	User to be Aborted
E:WU	10	Wake Up Sleeping User
E:SL	12	Begin Wait (Sleep)
E:QA	13	Queue for Access to RBBAT
E:ART	14	Activate Real Time User
E:UQA	16	Unqueue for Access
E:KO	18	Kick Out of Core
E:AP, E:NC	1A	Associate Shared Processor, Need Core Page
E:QE	1B	Quantum End
E:IC	1C	I/O Complete
E:QFI	1D	Queue for Real Time Interrupt
E:NSYMF	1E	No Symbiont File Entries OR RBBAT comm. Buffers
E:SYMF	1F	Symbiont File Entries or RBBAT Comm. Buffers Available
E:NSYMD	20	No Symbiont Disc Space
E:SYMD	21	Symbiont Disc Space Available
E:OCR	22	Open/Close Request
E:NOCR	23	Open/Close Available
E:CFB	24	Need COC Buffer
E:CBA	25	COC Buffer Available
E:ND	26	Need Swapper Page
E:DPA	27	Swapper Page Available
E:QFAC	28	Queue for ALLOCAT
E:UQFAC	29	Unqueue for ALLOCAT
E:NQW	30	ENQ Wait
E:NQR	31	ENQ Release

SCHEDULER/SWAPPER TABLES

SB:OSUL

Out-Swap user Numbers

0 7

First BYTE Contains number of Users in List
Length SMAXOUT

SB:FPL

Processor Numbers of ERETO Processors

0 7

First BYTE Contains number of Procs in List
Length = 2 *SMAXOUT

SH:EDA

Ending Disc Addresses for Swapout

0 15

Length = SMAXOUT

S:BDA

Beginning disc addresses for Swap Out

0 31

Length Smaxout

S:BCL

Pointers to Beginnings of Users' Command - - Lists for Swap Out

0 31

Length Smaxout

S:ECL

Pointers to Ends of Users' Command Lists for Swap Out

0 31

Length Smaxout

S:SCL

Seek Order	BA(SH:SDA) + DISP
X'2E'	COUNT 2 BYTES
Read Order	BA ((PROCESSOR'S JITS)) Physical Page Address) for Swap Out
X'2C'	Count 1 PAGE

0 31

4-Word IOCD Pair for Reading Processor Pages Length 2 *PPROCS

SH:SDA

Processors' Disc Addresses

0 15

Length 2 PPROCS

SB:PNL

In-Swap Processor Number

0 15

Length 4
First BYTE contains number of Processors in List

SCHEDULER/STATE TABLES/QUEUES

SB:HQ

User # of First User in Queue

0 7

Queues displaced by state Number. They are zero if no User in Queue.

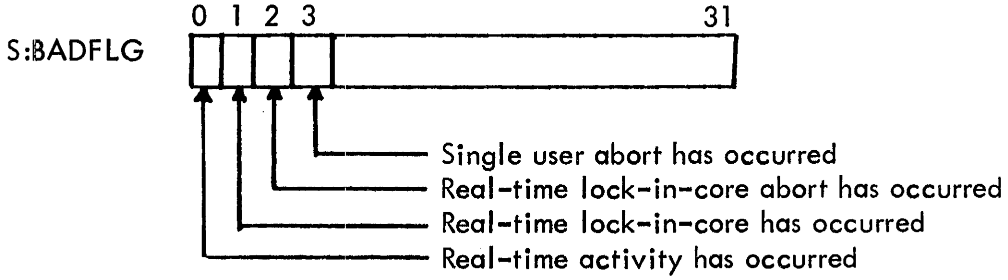
SB:TQ

User # of Last User in Queue

0 7

UB:FL and UB:BL are used to Link Users in Queue.

S:BADFLG is a one-word cell designed to indicate if any real-time activity has occurred. This will be displayed permanently by ANLZ.



S:IRPINC	Cell containing I/O complete priority increment. Default = 4.
S:IOPINC	Cell containing I/O complete priority increment. Default = 3.
S:CPINC	Cell containing special compute priority increment. Default = 1.
SB:RQ	Byte table containing user number for head of resourse subqueue - indexed by resource number which is the integer part of (E:BLK or E:REL)/2. Users in a subqueue are linked through U:MISC. Major queue for a user in any dynamic resource subqueue will be SQR or SQRO.
S:CUT	Cell, current user's priority. Set to X'FF' when idle.
S:PRIODEC	Cell, priority decrement to be applied when interrupting a user for a high-priority user. Applied to UB:PRIO.
S:RTIR	Cell, real-time user in and ready flag.
SB:HQ/SB:TQ	Initialized to have SNULL point to chain of empty user slots.

USER TABLES

UB:DB
Proc. # of Debugger if any (indexed by user #)
0 LENGTH SMUIS+1 7

UB:OV
Proc. # of Mon. Overlay Required by user (indexed by user #)
0 LENGTH SMUIS+1 7

UB:US
User State # (indexed by user #)
0 LENGTH SMUIS+1 7

UB:FL
FLINK in state queue (0 TOP) (indexed by user #)
0 LENGTH SMUIS+1 7

UB:BL
Blink in state queue (0 if Bottom) (indexed by user #)
0 LENGTH SMUIS+1 7

UX:JIT
Physical Page # of JIT (if in Core) (indexed by user #)
0 LENGTH SMUIS+1 7/15

UB:MF
Outstanding I/O Operations
0 LENGTH SMUIS+1 7

UB:ASP
PROC # of special processor except TEL or CCI (indexed by user #)
0 LENGTH SMUIS+1 7

UB:PRIO-*IA
Current user Execution priority
0 Length SMUIS+1 7

UB:PRIOB-*IA
User Base Execution Priority
0 Length SMUIS+1 7

UB:APO
Associated Processor Overlay (by user #)
0 LENGTH SMUIS+1 7

UB:APR
Processor Number or Proc. Overlay (by user #)
0 LENGTH SMUIS+1 7

UB:NECB
Total # of ECB's yet to be posted
0 Length SMUIS+1 7

UH:WL-*IA
Pointer to head of list of ECB's to be posted before re-scheduling
0 Length SMUIS+1 15

NOTE: SMUIS = MAXG+MAXB+MAXOL specified by user in PASS2

UB:PCT
User Page Count (by user #)
0 LENGTH SMUIS+1 7

UB:SWAPI
Index to Swap device for user
0 Length SMUIS+1 7

UB:C#
User cylinder #
0 Length SMUIS+1 7

UB:ACP
Proc. # of Command Processor
0 LENGTH SMUIS+1 7

*IA = Interrupt Altered

U:MISC
Miscellaneous Table - Use depends upon user state (by user #)
0 LENGTH SMUIS+1 31

U:P# (Disk Pack Swapper Only)
Proc # of processors outswapped with user (by user #)
0 LENGTH = SMUIS+1 31

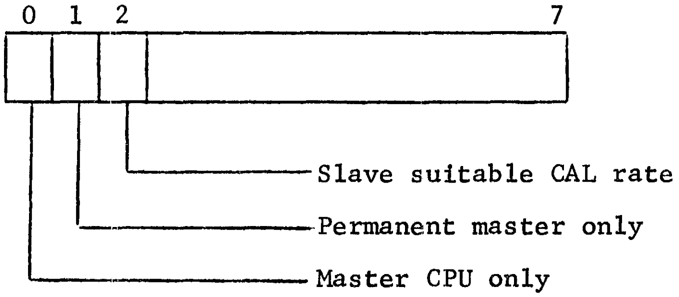
UH:JIT
Users JIT Disc Address, Zero if no JIT (by user #)
0 LENGTH SMUIS+1 15

UH:AJIT
Users Additional JIT Disc Address if any (by user #)
0 LENGTH SMUIS+1 15

SMUIS = Maximum number of users in system (parameter).

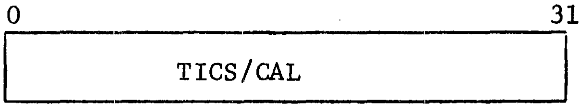
Added User Tables

UB:MPFLG - exists only if NSCPU > 0



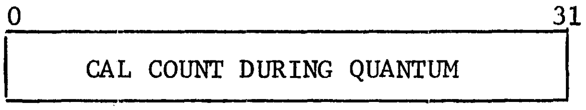
(length = SMUIS)

UB:CALR - exists only if NSCPU > 0



Average compute/cal during recent quantum

U:CALC - exists if NSCPU > 0



UH:FLG

1st Halfword of Users Flags
0 Length SMUIS+1 15

Bit	Meaning if Set
0	Bypass, available core too small now
1	STEP in progress or unblock received before block
2	Initial DCBs being swapped in
3	Special JIT access
4	Initialization must be done
5	DELTA is associated
6	JIT is in core
7	Job is batch
8	TEL in control (or CCI, other CP)
9	DELTA in control
10	Interactive user
11	Pure procedure must be swapped
12	OPNCLS user
13	:ACCTLG or :USERS open
14	Intentry inhibit (real time)
15	Ready to run

UH:FLG2 - *IA

2nd Halfword of Users Flags
0 Length SMUIS+1 15

Bit	Meaning if Set
0	Unused
1	Command processor break
2	Lock in core for RMA (Gentle)
3	COC event for transaction processing
4	Real time lock in core (absolute)
5	System ghost locked out (real time lock in core)
6	Interrupted during a CAL
7	Transaction processing function
8	Concurrent output mode (keyin) - Special Systems
9	Suspended for reconnection - Special Systems
10	COC line hang-up
11	Just swapped in
12	Swap Quantum not satisfied
13	User swap error
14	Context swap error
15	JIT swap error

UH:DL - *IA

DO List Pointer and Flags
0 Length SMUIS+1 15

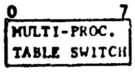
Bit	Meaning if Set
0	Job is to be aborted
1	Job is to be errored
2	Control-Y received
3	Break received
4 - 15	Doubleword address of DO list

Common Multiprocessor Control Tables

SB:INIT - preserved over recovery

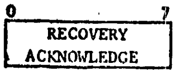
SB:MINT ✻

SB:MPSW



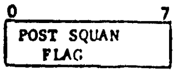
SB:PFLG

SB:RCVA



SB:RCVR

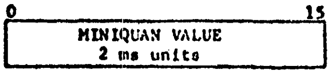
SB:SFLG



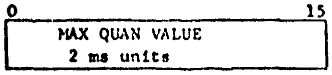
SB:STATE

0 - not active = stopped
1 - INIT+IDLE
2 - user in progress

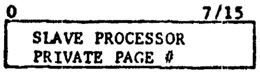
SH:MINQ



SH:MAXQ



SX:SPP



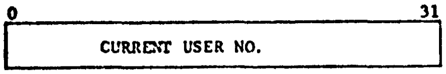
Byte if :BIG=0.

S:ADR

S:CLAIM

0 < CLAIMED
0 ≥ AVAILABLE

S:PCUN



✻ MPIPI - value def specifying presence of interprocessor interrupt pair. Controls interpretation of SB:MINT.

NSCPU - value def declaring number of CPUs which exist other than master.

1. All tables are NSCPU+1 long.
2. Entry 0 is master CPU.
3. Index by processor no. (software).

FB:FLT

0 7
FAULT TYPE INDEX

FB:EFLG

0 7
Error log buffer full flag

FH:SCRCH

0 15
SCREECH CODE

F:EADDR

0 31
Real address of error log buffer for slave CPU fault

Special Multi-Processing Cells

SL:BSTRT

0 31
AUTO START INHIBIT

0 => permit auto start at boot/reboot.
1 => inhibit auto start at boot/reboot.

S:SSCRCH

0 31
SLAVE INITIATED SCREECH CODE

Non-zero => SCREECH

S:MPKYN

0 31
MOOSE RE-ENTRANCY COUNTER (inc by KEYIN)

S:MPDISP

0 31
FLAG SET BY KEYIN TO TRIGGER MOOSE DISPLAY

S:STOUT

0 31
TIME OUT VALUE WHEN WAITING FOR RECOVERY ACKNOWLEDGE

Special Cells - Processor Private

XPSD receivers for the following traps and interrupts are contained in each processors' private page:

NOPPSD	Trap X'40', non-allowed operation
UNIMPSD	Trap X'41', unimplemented instruction
STKLPSD	Trap X'42', stack limit trap
FIXOVPSD	Trap X'43', fixed point arithmetic
FLTFPSD	Trap X'44', floating point arithmetic
DECPSD	Trap X'45', decimal fault
CAL1PSD	Trap X'48', Call 1
CAL2PSD	Trap X'49', Call 2
CAL3PSD	Trap X'4A', Call 3
CAL4PSD	Trap X'4B', Call 4
IPT47	Trap X'47', interprocessor trap (X560)
PSD$T46	Trap X'46', watchdog timer
PSD$T4C	Trap X'4C', parity error
PSD$T4D	Trap X'4D', instruction exception
PSD$I57	Interrupt X'57'
PFSR46PSD	Trap X'46', receiver during power fail safe
POWROFF	Interrupt/Trap X'51' power off

The following XPSD receivers for miscellaneous purposes are also located in the private page:

CTRAPSD	Transfer to central trap handler
RCVPSD	Entry to recovery/T:SCREECHS
BLKPSD	Block user on slave CPU

S:PNO

0 31

PROCESSOR NO. (SOFTWARE)

This cell exists in VPXPSDT, the CPU private page, and hence is unique for each processor. Master is processor 0. Slave numbers are set by MOOSE.

S:CLOCK4

0 31

Clock 4, the subjective counter, will tick indirect this cell.

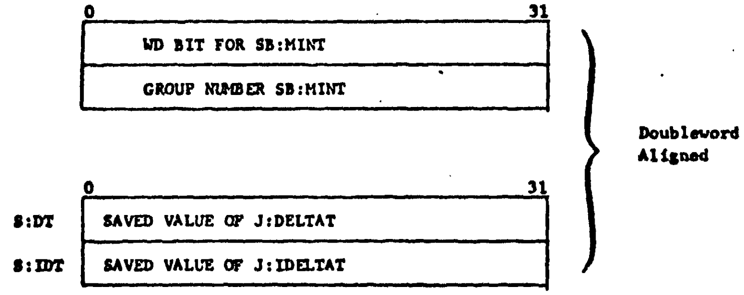
SREGS

0 31

16 words

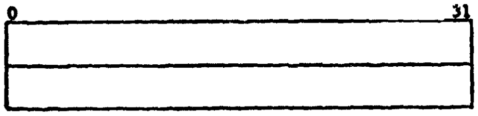
Trap handler temp for register save

S:INTBIT



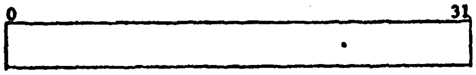
S:CUN

PULLE1



Doubleword temp for T:PULLE and entry.

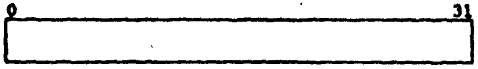
TEMP$BREG



Temp cell for use while acquiring last branch register.

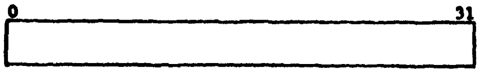
Doubleword aligned temp block for general miscellaneous use.

PFSRSW



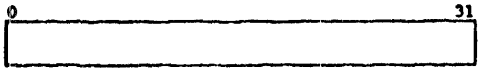
Sense switch settings at power fail safe.

BALANCE



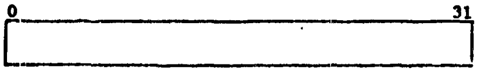
Balance counter for power fail safe.

OFFCNTER



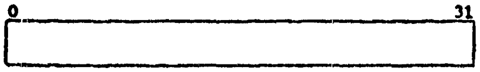
Count of power off interrupts.

ONCNTER



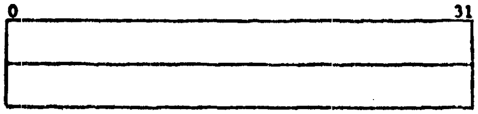
Count of power on interrupts.

NFRST



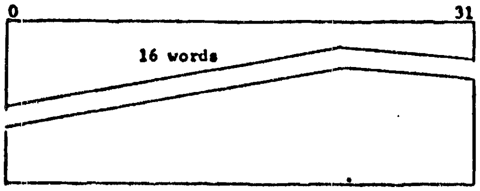
Flag for multiple power off interrupts.

FD:PSD



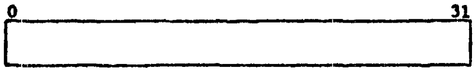
PSD at time of slave CPU fault.

F:REGS
SREG1



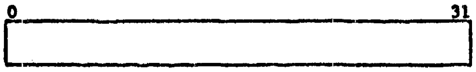
Register block one at PFSR or registers at CPU fault.

FB:CF



Condition codes and floating mode bits at hardware fault.

FB:END



End action driver for CPU fault handler.

F:TEMP1

0 31

Fault handler temp cell.

F:TEMP2

0 31

Fault handler temp cell.

		0 31	
F:ELOGB	0	CODE	
	1	TIME	
FD:EPSD	2	PSD	
	3	PSD	
	4		RESERVED
	5		
FH:IFLG	6	CCS AND FLAGS	
F:IREAL	7	REAL ADDR OF TRAPPED INST	
F:INST	8	TRAPPED INST	
F:EVIRT	9	ANLZ CC AND VIRTUAL ADDR	
F:EREAL	A	REAL EFFECTIVE ADDR	

Error log buffer for slave fault handlers.

Multi-Processing Performance Monitor Cells

C:SIDLE

0 31
SLAVE 1 IDLE
SLAVE 2 IDLE
SLAVE 3 IDLE

C:SUSER

0 31
SLAVE 1 COMPUTE
SLAVE 2 COMPUTE
SLAVE 3 COMPUTE

C:SCHED

0 31
OF SCHEDULES FOR MASTER
OF SCHEDULES FOR SLAVE 1
SLAVE 2
SLAVE 3

Multiprocessing Sysgen Built Tables

If no :SCPU command is detected, PASS2 builds a dummy command. From this command the load module STABLES is built. This module contains the following tables and absolute DEFs.

Absolute DEFs

NSCPU = # of slave CPUs (from NSCPU)*
MPIPI = 1 if MPIPI specified
= 0 if MPIPI not specified

* These are the only entries in STABLES if a non-multi-processing system, i.e., NSCPU = 0, MPIPI = 0, S:ADR is a word long.

Tables

name	entry size	length	contents
S:PCUN	wd	NSCPU+1	0
SB:MPSW	byte	NSCPU+1	0
SB:PFLG	byte	NSCPU+1	0
SX:SPP	byte/hw depending on :BIG	NSCPU+1	0
SB:STATE	byte	NSCPU+1	0
SB:INIT	byte	NSCPU+1	bit 7 = 1 if AUTO specified for entry
S:CLAIM	word	NSCPU+1	0
SB:MINT	byte	NSCPU+1	INTS value specified for entry
SH:MINQ	hw	NSCPU+1	MINQ/2 for entry
SH:MAXQ	hw	NSCPU+1	MAXQ/2 for entry
S:ADR	word	NSCPU+1	0
SB:SFLG	byte	NSCPU+1	0
SB:RCVR	byte	NSCPU+1	0
SB:RCVA	byte	NSCPU+1	0
SL:BSTRT	wd	1	1 if NOAUTO specified
S:MPKYN	wd	1	0
S:MPDISP	wd	1	0
SL:MPCALR	wd	1	10
S:STOUT	wd	1	3000
FB:FLT	byte	NSCPU+1	0
F:EADDR	word	NSCPU+1	0
FB:EFLG	byte	NSCPU+1	0
F:PFSR	word	1	0
FH:SCRCH	halfword	NSCPU+1	0

Tables Displaced by Processor Number (Located in M:SPROCS)

P:NAME
PROCESSOR ROOT OR OVERLAY NAME IN TEXTC FORMAT
0 31 32 63

PB:LNK*
Proc. # of next overlay
0 7

PX:HPP**
Head of physical page chain
0 7/15

PX:TPP
Tail of physical page chain
0 7/15

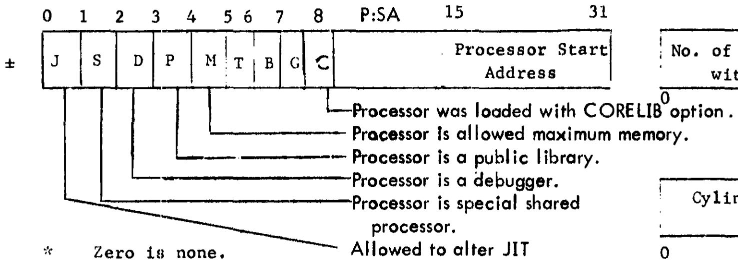
PB:UC
No. of users in core using processor
0 7

PB:PSZ
No. of pages of pure procedure
0 7

PB:DSZ
No. of pages of Data
0 7

PB:DCBSZ
No. of pages of DCBs
0 7

PB:PVA
Virtual page No. of first proc. page
0 7



PB:REP
No. of users associated with Processor
0 7

PB:HVA
Virtual page No. of first page not used
0 7

PB:C#***
Cylinder Number of Procedure
0 7

PB:DC#***
Cylinder Number of Data
0 7

* Zero is none.
** Zero if not in core.
*** Disk Pack Swapping Systems only.

PH:PDA
Disk address of first page of procedure
0 15

PH:DDA
Disk address of first page of data & DCBs
0 15

PH:FRQ
0 15

P:TCB
Processors TCB address (zero implies none)
0 31

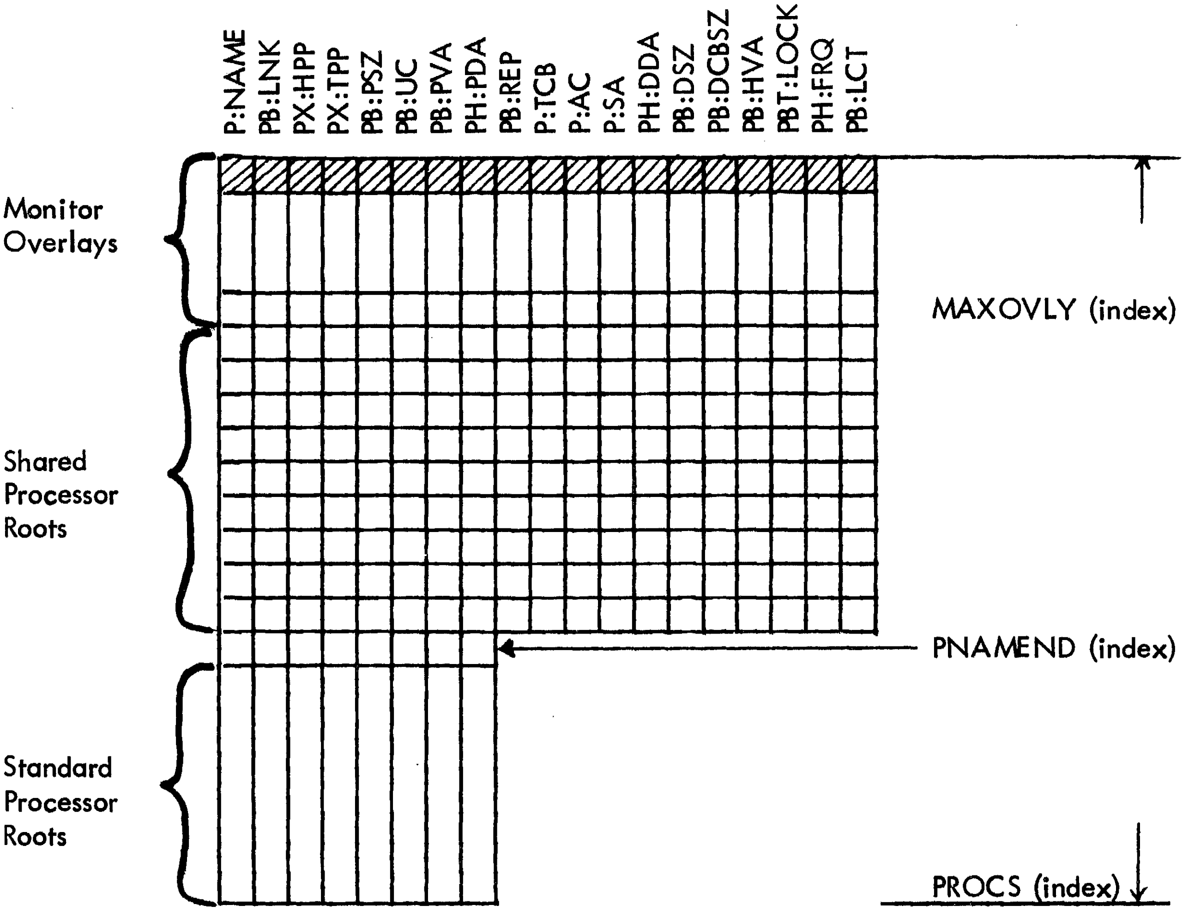
PB:LCT
# of users associated that are real-time locked in core
0 7

P:AC
Access codes for top 16 virtual pages (special processor area)
0 31 32 63

PBT:LOCK
Processor locked in core bit table - length = (PNAMEND + 31)/32 words
0 31 32 63

1 bit/processor - indexed (from left) by processor number.

±T - Teletype Command Processor
B - Batch Command Processor
G - Ghost Command Processor



MAXOLVY	number of monitor overlays plus one (plus one because entry 0 in processor tables is not used)
SPSIZE	total number of spare pages required for one shared processor (a PASS2 parameter)
PNAMEND	index number plus one of last shared processor root entry
PPROCS	index number plus one of last shared processor overlay entry

GHOST JOB TABLES-Interrupt Altered

MAXG - Maximum number of Ghost Jobs

S:GJOBTBL

0

Names of Ghost Jobs (Indexed by Ghost Job Number) (TEXTC)
= 0 if in active slot with index ≥ MING

Length = MAXG-

0 63

SB:GJOBUN

0 7

Ghost Job User #
(Indexed by Ghost Job #)

Length = MAXG+1
= index if under < MING
= 0 if inactive slot with index ≥ MING

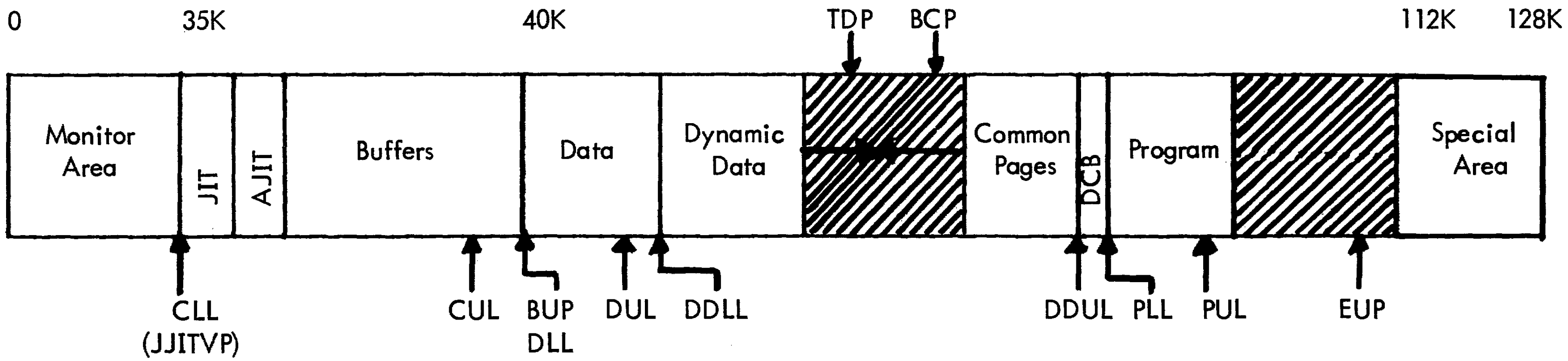
S:GJOBACN

Account for Ghost (Indexed by
Ghost Job Number)

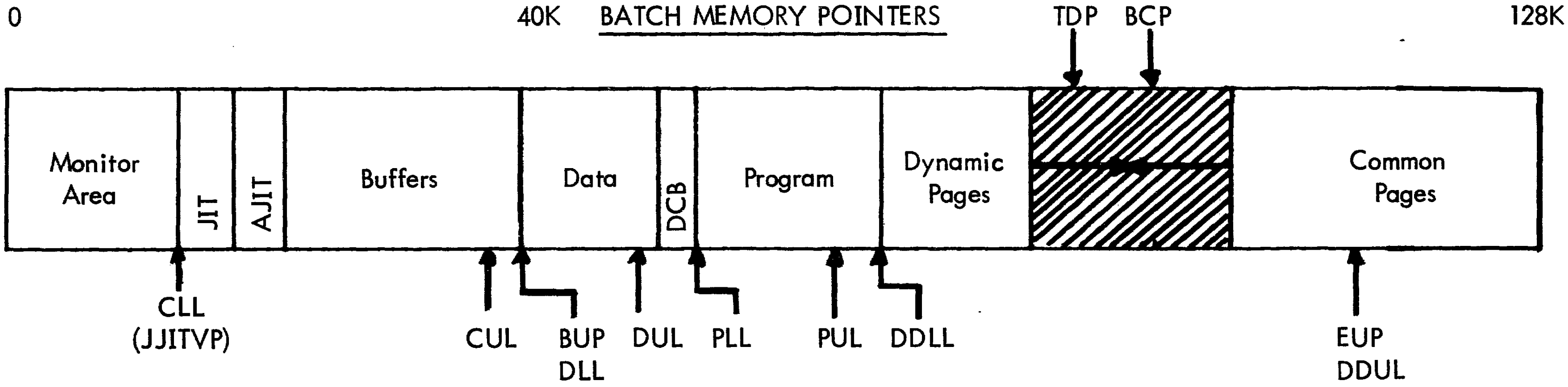
Length = MAXG-

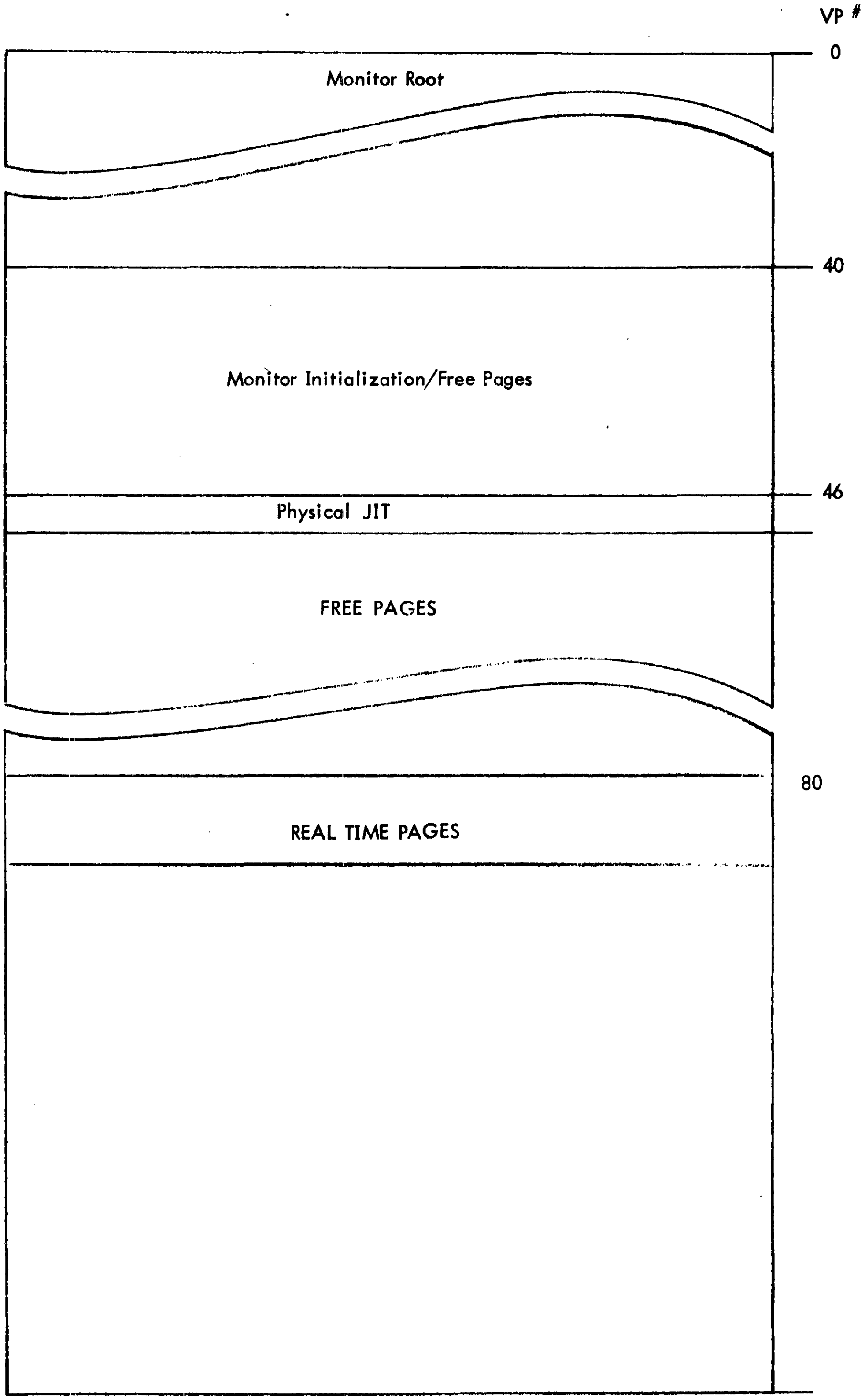
0 63

ON-LINE MEMORY POINTERS

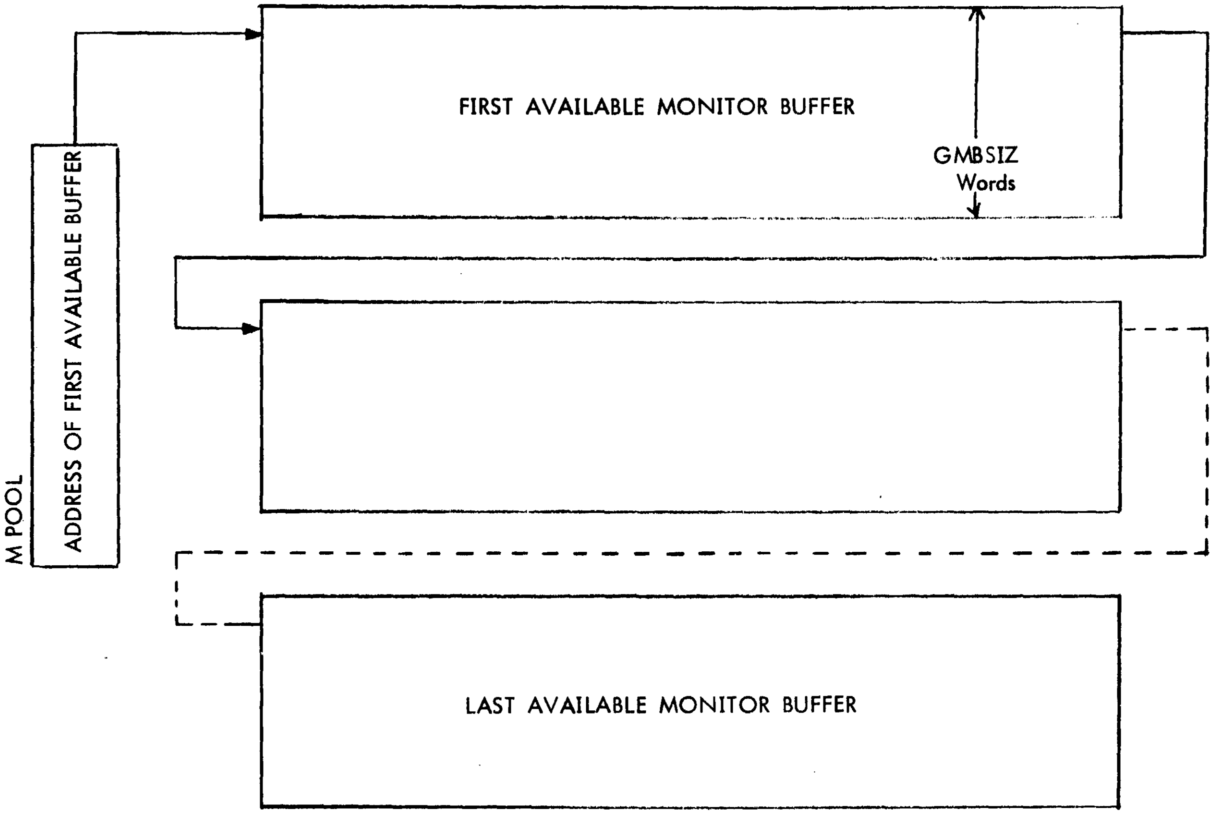


BATCH MEMORY POINTERS

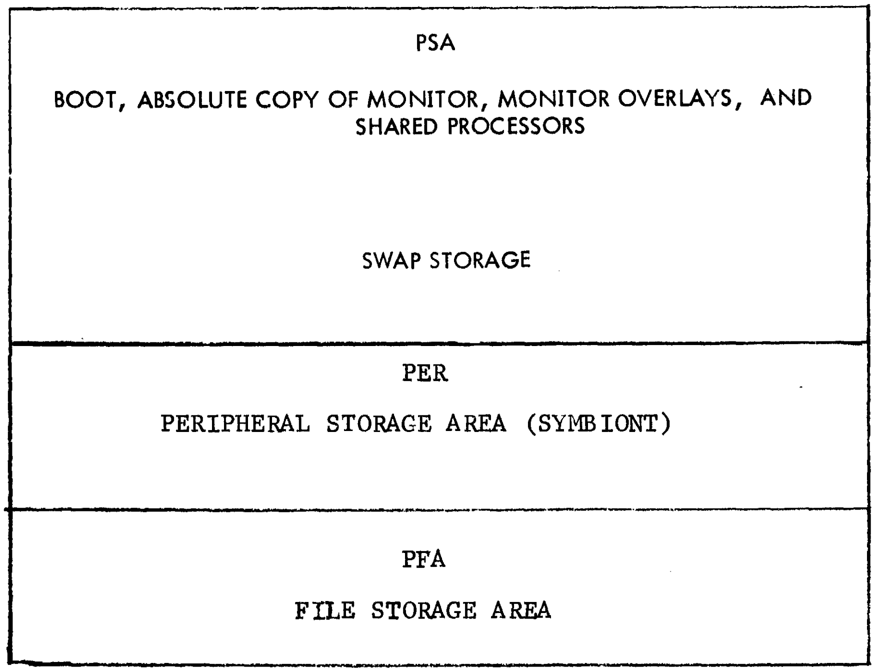



VP #
0
Monitor Root
40
Monitor Initialization/Free Pages
46
Physical JIT
FREE PAGES
80
REAL TIME PAGES

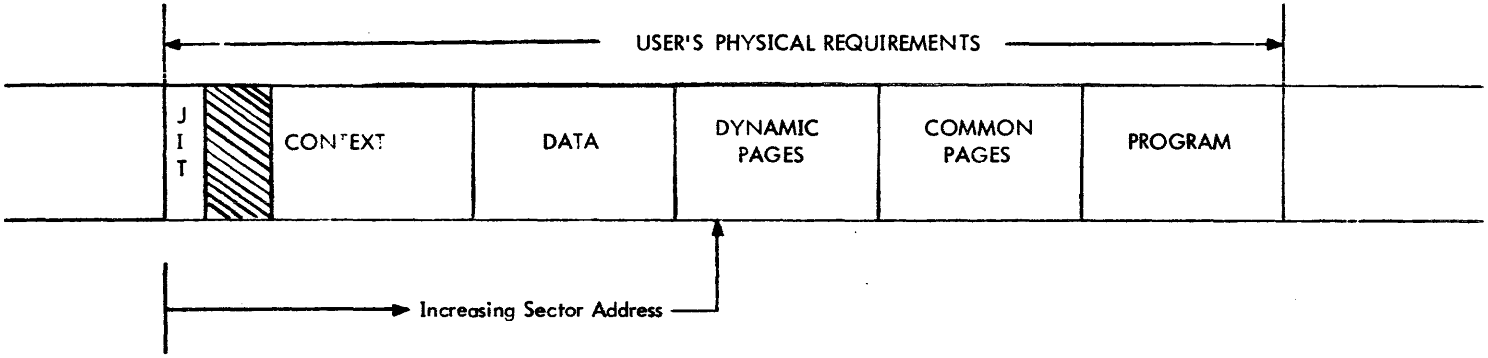
CP-V Buffer Linking



SYSTEM LAYOUT ON RAD



SWAP STORAGE LAYOUT



The disc address of JIT is always maintained in core. The JIT page contains the disc address of the remaining portions of the user's program. The shaded area represents a sufficient number of sector times to allow for initializing the command chains.

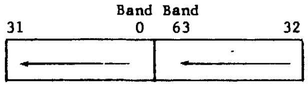
SWAPPING RAD GRANULE TABLE

M:SGP is a word table containing pointers to the swapping granule table for each RAD swap device.

There are four types of granule tables:

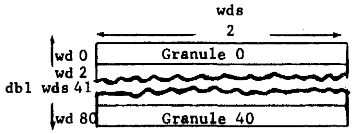
Type	RAD	PSA (Hex)		
0	7212	0	PSA	40
1	7232	0	PSA	80
2	7232	81	PSA	100
3	7232	101	PSA	200
4	3214	0	PSA	80
5	3214	81	PSA	100

Vertical words are granule positions. Horizontal is track or band number. In each word tracks go from R to L. If bit is set, this granule is available.



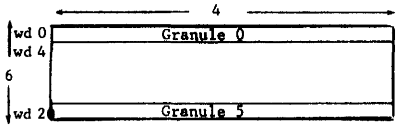
Swapping Granule Table

PSA RAD TYPE 0



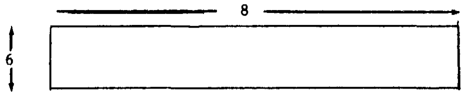
Type 0

PSA RAD TYPE 1



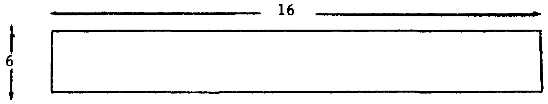
Type 1

PSD RAD TYPE 2



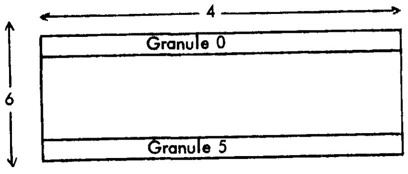
Type 2

PSA RAD TYPE 3



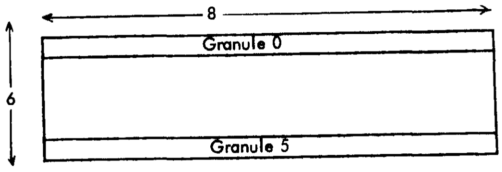
Type 3

PSA RAD TYPE 4



Type 4

PSA RAD TYPE 5



Type 5

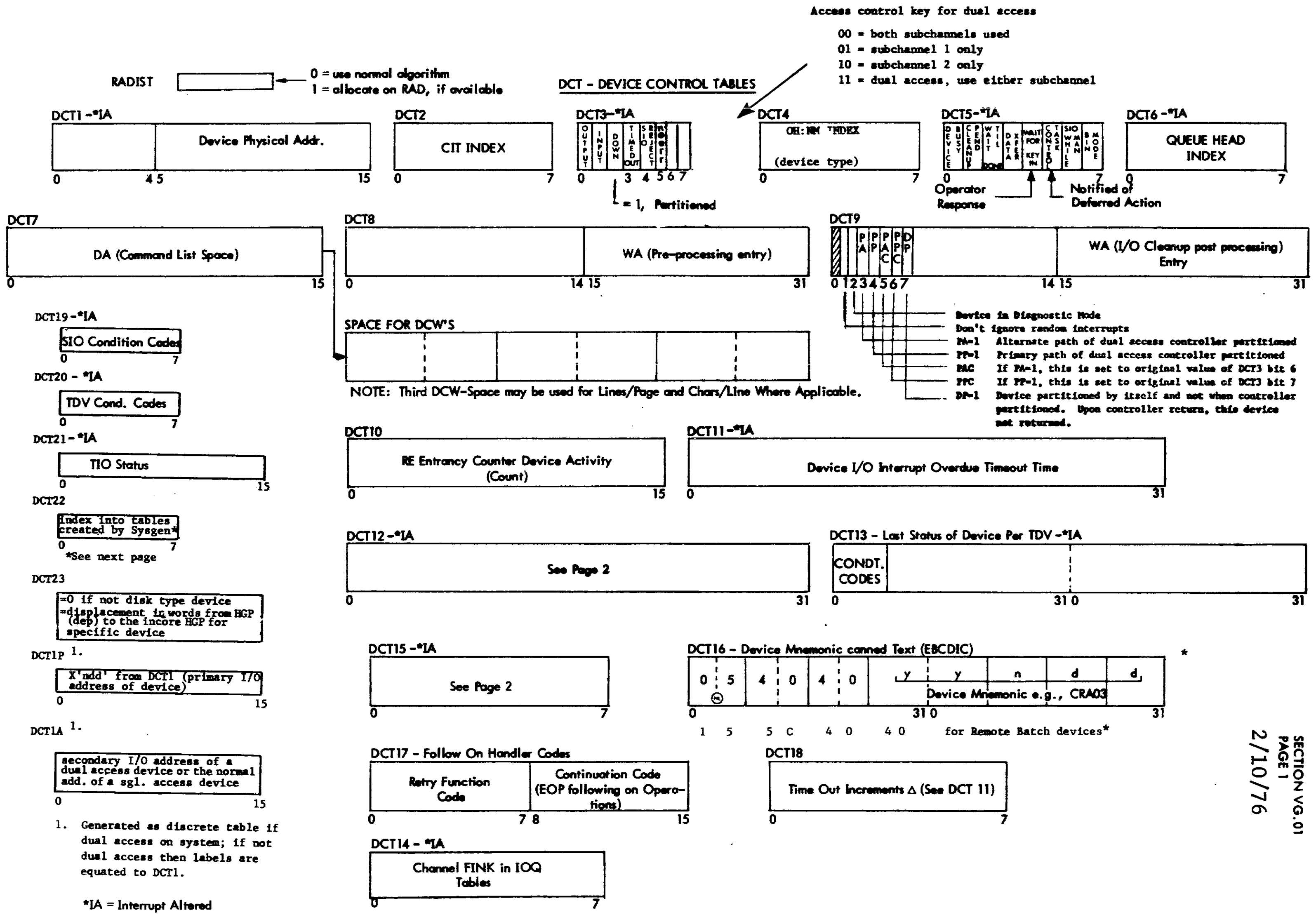
Name	Value						Description
PSA Type	0	1	2	3	4	5	
MB:GAM1	63	7	7	7	7	7	Granule address mask.
MB:GAM2	1	3	7	15	3	7	Mask to extract #SGP words/granule.
MB:GAM3	-1	-2	-3	-4	-2	-3	Shift count to convert SGP index to granule#.
MB:GAM4	6	3	3	3	3	3	Shift count to form track address.
MB:GAM5	7	-4	-4	-4	-4	-4	Shift count to obtain track address.
MB:GAM6	127	15	15	15	15	15	Sector address mask.
MB:GAM7	0	0	0	0	16	16	Low order track bit.
MB:GPT	41	6	6	6	6	6	#Granules per track.
MB:SPT	82	12	12	12	12	12	#Sectors per track.
MB:SWAPS	0	1	2	3	1	2	Shift count to obtain SGP index from granule#.
MB:DWT	41	12	24	48	12	24	SGP size in doublewords.
MB:SPACEJIT	7	1	1	1	1	1	Granule increment to space JITs around RAD.
M:GATLIM	63	127	255	511	127	255	Highest valid track number.
M:GASLIM	80	10	10	10	10	10	Highest valid sector number.
M:ADRINCR	46	4	4	4	5	5	Increment to add to last sector to get first sector on next track.

Note: 1. Swapper related tables contain an entry for each swap device. The total number of entries is defined by LSWAP+1.

2. The user table, UB:SWAPI, contains the index into the swap tables.

M:CLBGN	Beginning of swap command list.
M:FREE#GRAN	Number of available granules.
M:HLTIC	TIC to be inserted at end of command list.
M:JITPAGE	Granule position for next JIT.
M:SGP	Address of SGP table.
M:SNSDA	Buffer for sense information.
M:SWAPD	Device address.
M:SWPEND	Highest possible PSA seek address.
M:WCKBCL	Beginning of command list for write check.
M:WCKECL	End of command list for write check.
MB:SDI	Swapper DCT index.
MB:SFC	Swapper function code.
MB:#RTRY	Number of remaining retries.
MH:CLEND	End of swap command list.

DCT - DEVICE CONTROL TABLES



- Device in Diagnostic Mode
- Don't ignore random interrupts
- PA=1 Alternate path of dual access controller partitioned
- PP=1 Primary path of dual access controller partitioned
- PAC If PA=1, this is set to original value of DCT3 bit 6
- PPC If PP=1, this is set to original value of DCT3 bit 7
- DP=1 Device partitioned by itself and not when controller partitioned. Upon controller return, this device not returned.

NOTE: Third DCW-Space may be used for Lines/Page and Chars/Line Where Applicable.

*See next page

=0 if not disk type device
=displacement in words from HGP (dep) to the incore HGP for specific device

X'ndd' from DCT1 (primary I/O address of device)

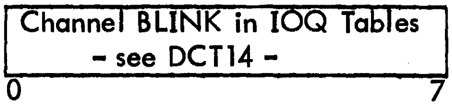
secondary I/O address of a dual access device or the normal add. of a sgl. access device

1. Generated as discrete table if dual access on system; if not dual access then labels are equated to DCT1.

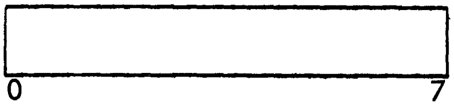
*IA = Interrupt Altered

DCT Tables

DCT15 - *IA



if DCT5 indicates "DEVICE BUSY"

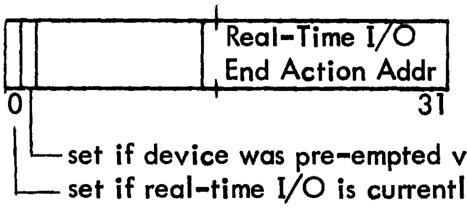


it DCT5 does not indicate "DEVICE BUSY"

0 implies monitor I/O permissible
≠0 implies this is the user number of real-time user who may issue M:IOEX requests

DCT12 - *IA

if DCT15 = 0

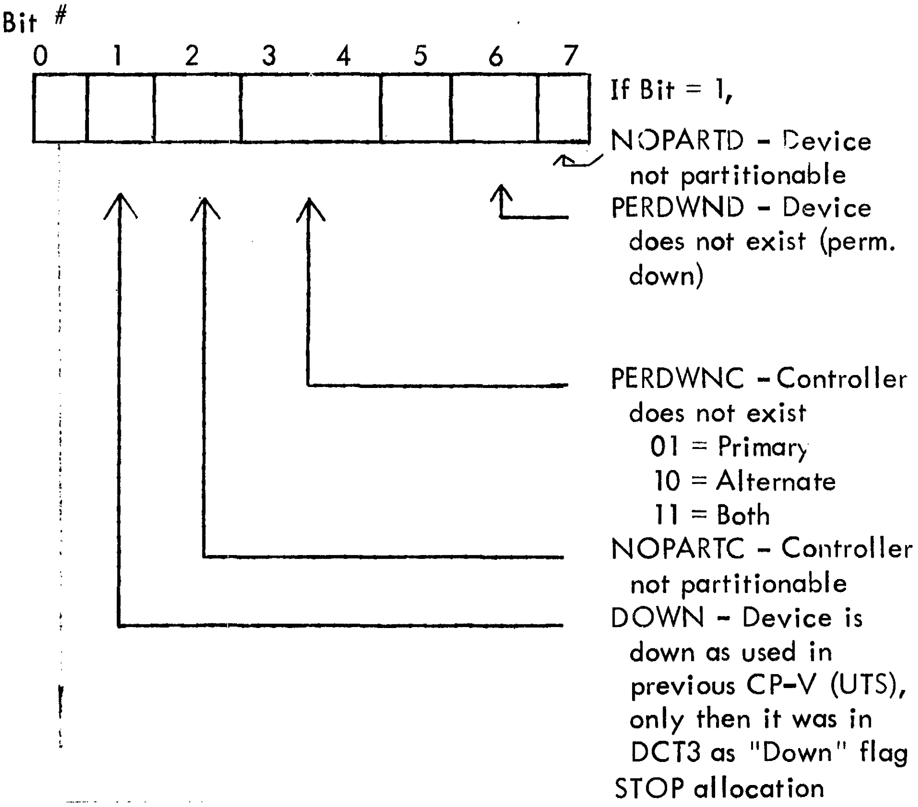


set if device was pre-empted via a DCB call to STOPIO
set if real-time I/O is currently active (i.e. interrupt pending)

if DCT15 ≠ 0

DCT24 - *IA - RMA TABLE

- Byte table
- Parallel to DCT1
- Entry contains:



If Bit = 1,

NOPARTD - Device not partitionable

PERDWND - Device does not exist (perm. down)

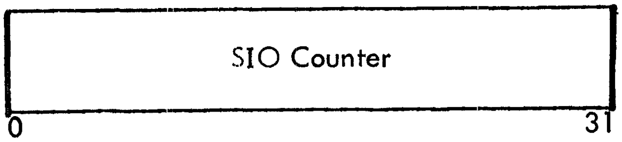
PERDWNC - Controller does not exist
01 = Primary
10 = Alternate
11 = Both

NOPARTC - Controller not partitionable

DOWN - Device is down as used in previous CP-V (UTS), only then it was in DCT3 as "Down" flag

STOP allocation

DCT25



The following tables are idexed by DCT22

Disclims

Largest valid SECTOR ADDRESS + 1

NCYL

Number of cylinders on device. 0 for RADs.

NTPC

Number of tracks/cylinder. For RAD, 1 cylinder assumed.

NSPT

Number of sectors/track

CYL $SHFT

Shift instruction to position CYL portion of seek address of the form SLS, 9 XX or NOP if not present

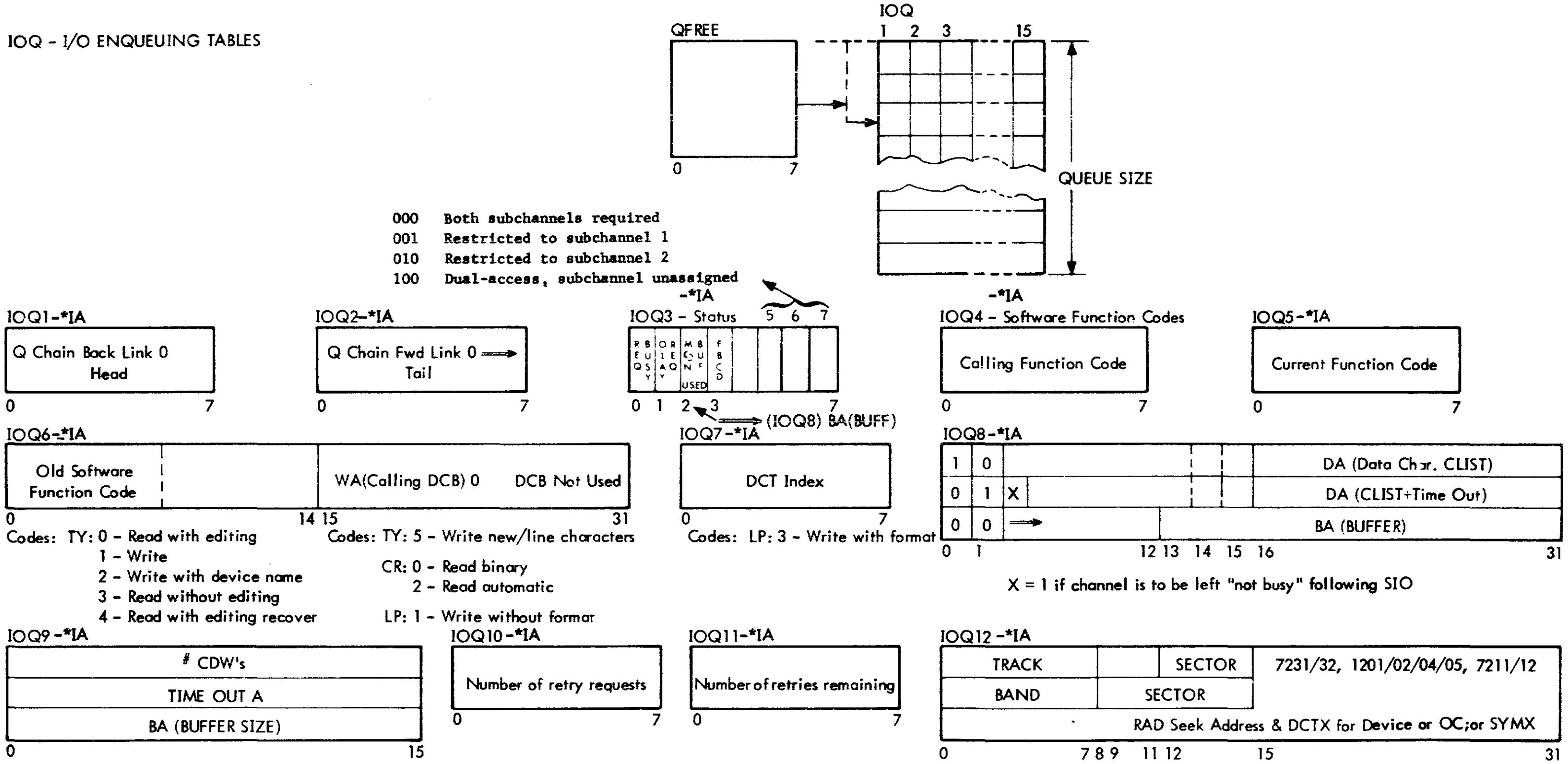
TRK $SHFT

Same as above except for track SLS, 8 XX

SEC $SHFT

Same as above except for SEC SLS, 7

IOQ - I/O ENQUEUING TABLES



000 Both subchannels required
001 Restricted to subchannel 1
010 Restricted to subchannel 2
100 Dual-access, subchannel unassigned

X = 1 if channel is to be left "not busy" following SIO

Codes: TY: 0 - Read with editing
1 - Write
2 - Write with device name
3 - Read without editing
4 - Read with editing recover

Codes: TY: 5 - Write new/line characters
CR: 0 - Read binary
2 - Read automatic
LP: 1 - Write without format

Codes: LP: 3 - Write with format

Codes (continued):

CP: 0 - Punch EBCDIC
1 - Punch binary

DC: 0 - Read
1 - Write
2 - Sense
3 - Check write
4 - Write with check-write

7T, 9T: 0 - Read 9T(7T read packed)
1 - Write 9T (7T write packed)
2 - Read 7T binary
3 - Write 7T binary
4 - Back space record

7T, 9T: 5 - Fore space record
6 - Back space file
7 - Fore space file
8 - Write tape mark
9 - Rewind
A - Rewind off-line
B - 9T sense
C - Read 7T decimal
D - Write 7T decimal
E - Read recovery 9T (7T packed)
F - Write recovery 9T, (7T packed)
10 - Read backward 9T
11 - Read backward 9T recovery
12 - 7T read binary recovery
13 - 7T write binary recovery
14 - 7T read decimal recovery

7T, 9T: 15 - 7T write decimal recovery
16 - Read backward 7T packed
17 - Read backward 7T binary
18 - Read backward 7T decimal

IOQ - I/O ENQUEUING TABLES (continued)

IOQ13 - *IA

TASK (Interrupt number)		WA (End-Action return)	End Action Information
0 – 7	8 – 14	15 – 31	0 – 31

0 ⟹ No End Action

IOQ14 - *IA: Request Priority (0 – 7)

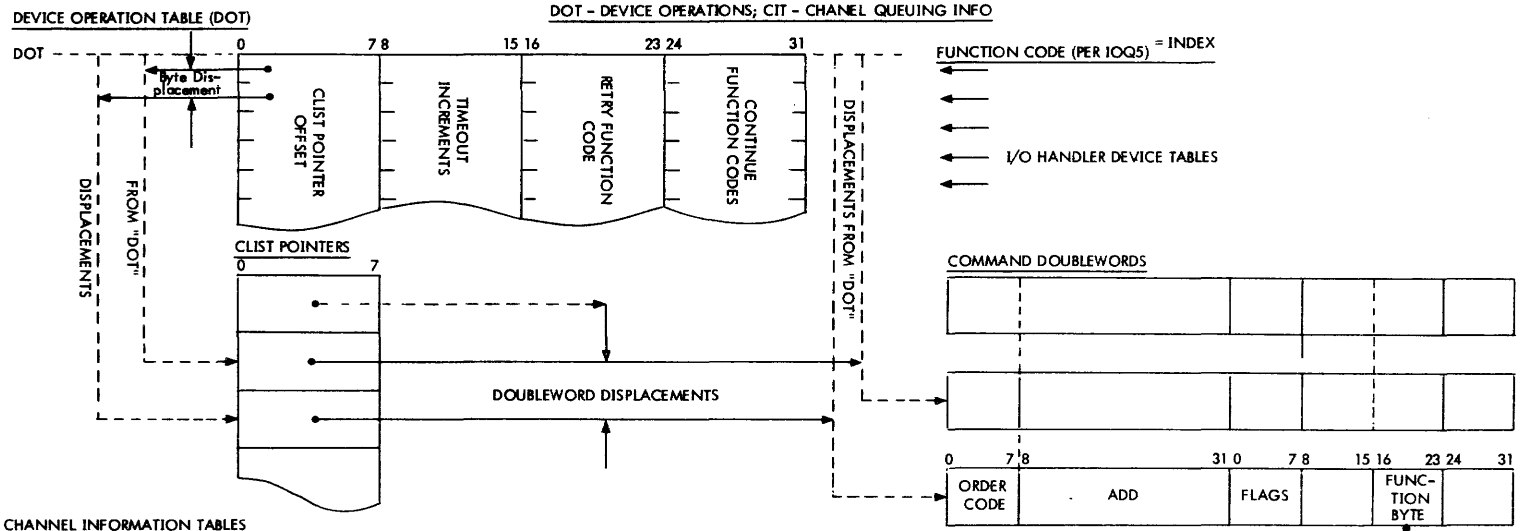
IOQ15 - *IA: User Number (0 – 7)

IOQ16: ECB Address (0 – 31)

CURBQ: Count of Number of Background I/O Requests Queued

OCFLG: Operator's Console Flag - ≠ 0 ⟹ Keyin In Progress

DOT - DEVICE OPERATIONS; CIT - CHANEL QUEUING INFO



0 - Use CDW as is (without function byte)
1 - Seek, seek address in IOQ12
2 - BA(BUFF) ⟹ IOQ8, BC(BUFSIZ) ⟹ IOQ9
3 - Message for OC device (DCTX for device in IOQ12)
4 - Special handler function branch to "ADD"

CHANNEL INFORMATION TABLES

CIT 1: Q - Chain Head Index (0 – 7)

CIT 2: Q - Chain Tail Index (0 – 7)

CIT 3 (bits 0 – 7):

- 0 → Subchannel 1 busy
- 1 → Subchannel 2 busy
- 2 → Subchannel 1 held
- 3 → Subchannel 2 held
- 4 → Dual access channel
- 5 → Preferred subchannel
  (0 if subchannel 2 last selected)
  (1 if subchannel 1 last selected)

CIT 4: bits 0 – 14; bits 15 – 31: 0 ⟹ None, ≠0 ⟹ NA (Queueing Optimization Routine This Channel)

CIT5: Holding request index if subchannel 1 in hold state (0 – 7)

CIT6 1.: Holding request index if subchannel 2 in hold state (0 – 7)

1. Generated for dual access system only; otherwise label is requested to CIT5.

Device Type-Class Tables (DTT)

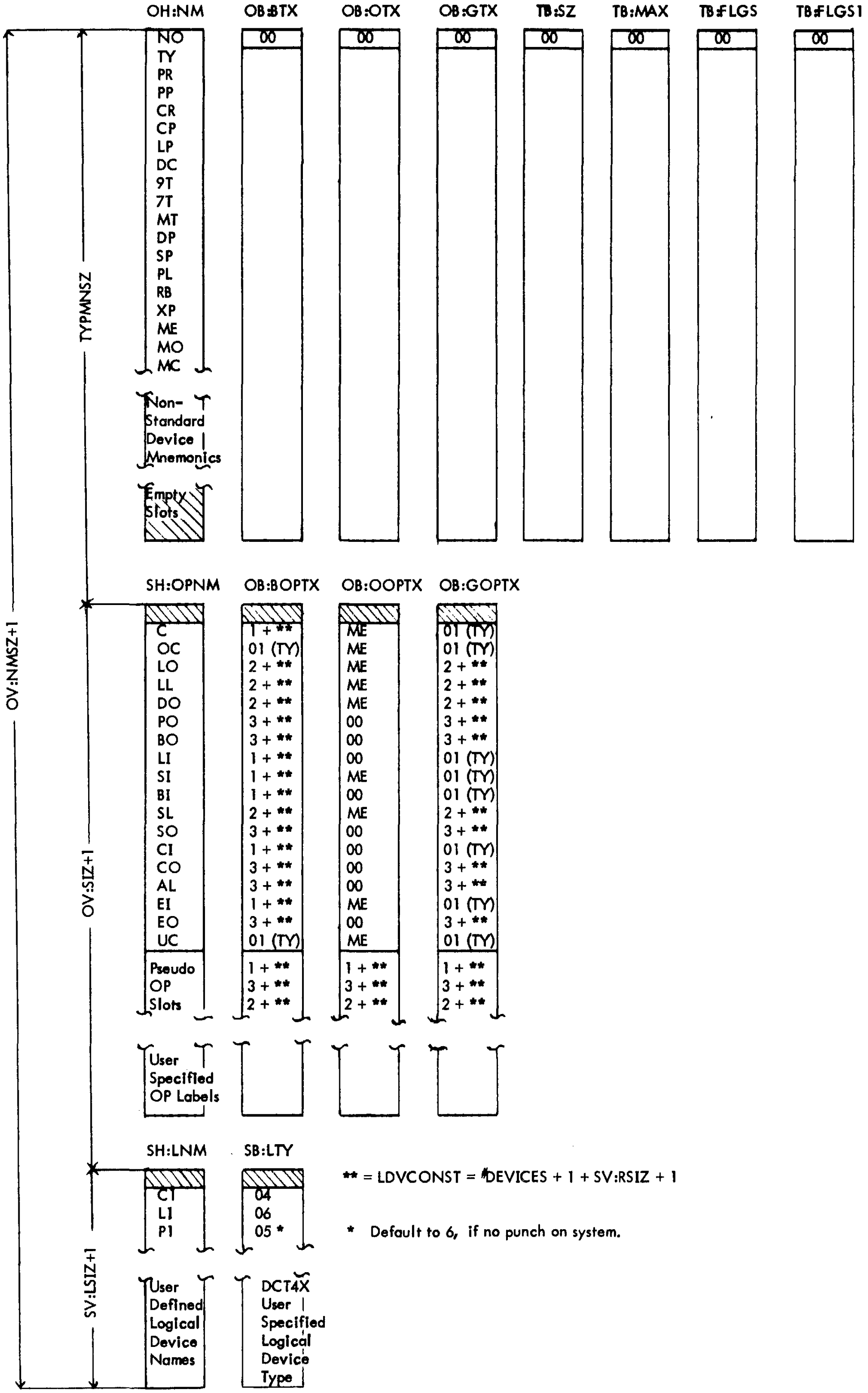
The directing label table exists in three parts.

The first part consists of the device type or mnemonic table, the parallel assignment tables, $\text{OB:}\left\{\begin{matrix}\text{B}\\\text{O}\\\text{G}\end{matrix}\right\}\text{TX}$, and the descriptor tables. If the device appears in the Resource Allocation Table (SH:RNM), then, $\text{OB:}\left\{\begin{matrix}\text{B}\\\text{O}\\\text{G}\end{matrix}\right\}\text{TX}$ contains the index into SH:RNM.

If not, the entries will contain either the appropriate DCT index or the logical stream index i SH:LNM. The TB:SZ, TB:MAX, TB:FLGS, and TB:FLGS1 tables contain descriptors of the physical device attributes. The information is developed from the :DEVICE card in SYSGEN.

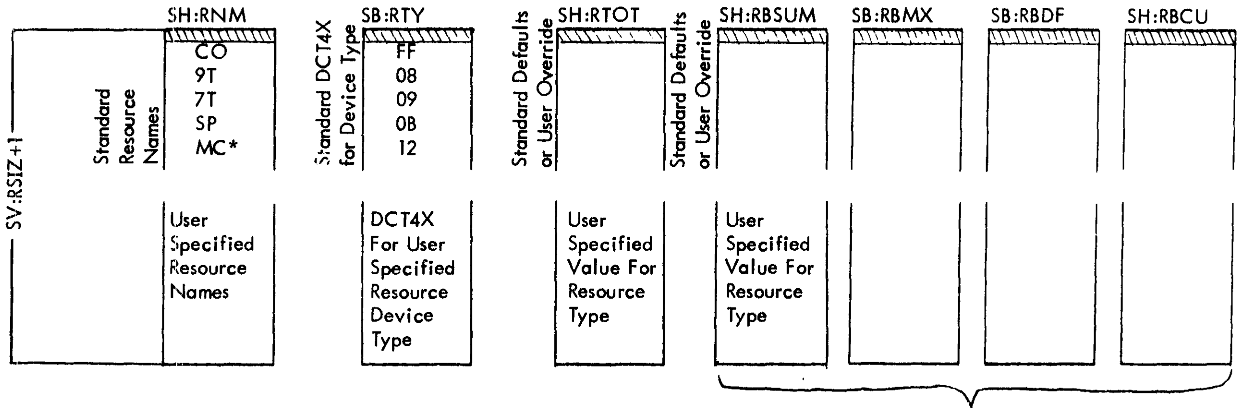
The second part consists of operational labels and the assignment table. This information is developed from the :OPLBLT card in SYSGEN.

The third part consists of logical stream names and the default device type to which they apply. This information comes from :LDEV in SYSGEN.



** = LDVCONST = #DEVICES + 1 + SV:RSIZ + 1

* Default to 6, if no punch on system.



* If MC is present as a device, it is automatically included in the group of standard resources.

These tables are repeated for on-line and ghost with the 'B' replaced by 'O' or 'G' in each of the table names.

Standard Defaults for these Resource Tables

Table Name \ Standard Resource Name	CO (core)	9T (tapes)	7T (tapes)	SP (private disk packs)	MC
SH:RTOT	X'7FFF'	#[1]	#	#	1
SH:RBSUM	X'7FFF'	#	#	#	1
SB:RBMX	X'10'	#	#	#	1
SB:RBDF	X'C'	0	0	0	0
SH:RBCU	0	0	0	0	0
SH:ROSUM	X'7FFF'	#	#	#	1
SB:ROMX	X'10'	#-1[2]	#-1	#-1	1
SB:RODF	X'C'	0	0	0	0
SH:ROCU	0	0	0	0	0
SH:RGSUM	X'7FFF'	#	#	#	1
SB:RGMX	X'FF'	#-1[2]	#-1	#-1	1
SB:RBDF	X'FF'	#-1[2]	#-1	#-1	1
SH:RCU	0	0	0	0	0

1 # is the total number of this type of device specified on :DEVICE commands.

2 If #-1 = 0, then the value is set to 1.

TYPMNSZ	Value definition that specifies number of device type mnemonics and empty slots.
OV:SIZ	Value definition which specifies number of operational labels defined.
OV:NM	TEXT of label as specified on :OPLBLT or type mnemonic as specified on the :DEVICE.
SH:OPNM	Default OP label table and OP label assignments.
OB:BTX	*Index into DCT, RAT or LDEV (Batch).
OB:BOPTX	*Index into DCT or LDEV (Batch).
OB:OTX	*Index into DCT, RAT, or LDEV (On-line).
OB:OOPTX	*Index into DCT or LDEV (On-line).
OB:GTX	*Index into DCT, RAT or LDEV (Ghost).
OB:GODTX	*Index into DCT or LDEV (Ghost).
TB:SZ	Form size or minimum record length.
TB:MAX	Maximum record size.

	TYPE		IN	OUT	BIN	READ REVERSE	COMP	VFC
TB:FLGS	0	1	2	3	4	5	6	7
OTHER	0	0	-	-	-	-	-	-
LISTING	0	1	0	1	-		1	1
TAPE	1	0	1	1	-	-	-	-
PACK (SP)	1	1	1	1	-	-	-	-
RAD (DC)	1	1	0	0				

---

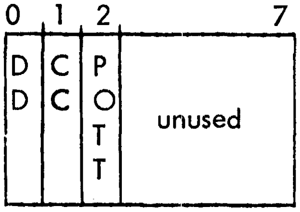
* A DCT index appears as the DCT index into the DCT tables.

A RAT index appears as #DEVICES+1+RAT index; RAT index = index into SH:RNM

An LDEV index appears as #DEVICES+1+SV:RSIZ+1+LDEV index; LDEV index = index into SH:LNM; #DEVICES = DCTSIZE

If TB:FLGS Type = 10(TAPE), then

TB:FLGS1



where

DD is the Dual Density indicator
0 = device does not have dual density capability
1 = device has dual density capability

CC is the Code Conversion indicator
0 = device does not have code conversion capability
1 = device has code conversion capability

POTT is the Potter tape drive indicator
1 = 9T device for which sense and set correction orders are invalid

SV:LSIZ	Value definition defining the number of logical devices defined, 15 maximum.
SV:LSIZP	Value definition determined as follows:

#LDEV	SV:LSIZP
0-3	2
4-5	3
6	4
7-8	5
9-10	6
11	7
12-13	8
14-15	9

SV:LSIZPA	Value definition equal to SV:LSIZP*9.
OV:NMSZ	Value definition equal to TYPMNSZ+OV:SIZ+1+SV:LSIZ+1.
SH:LNM	TEXT of label as specified on :LDEV at SYSGEN time.
SB:LTY	Default device type.

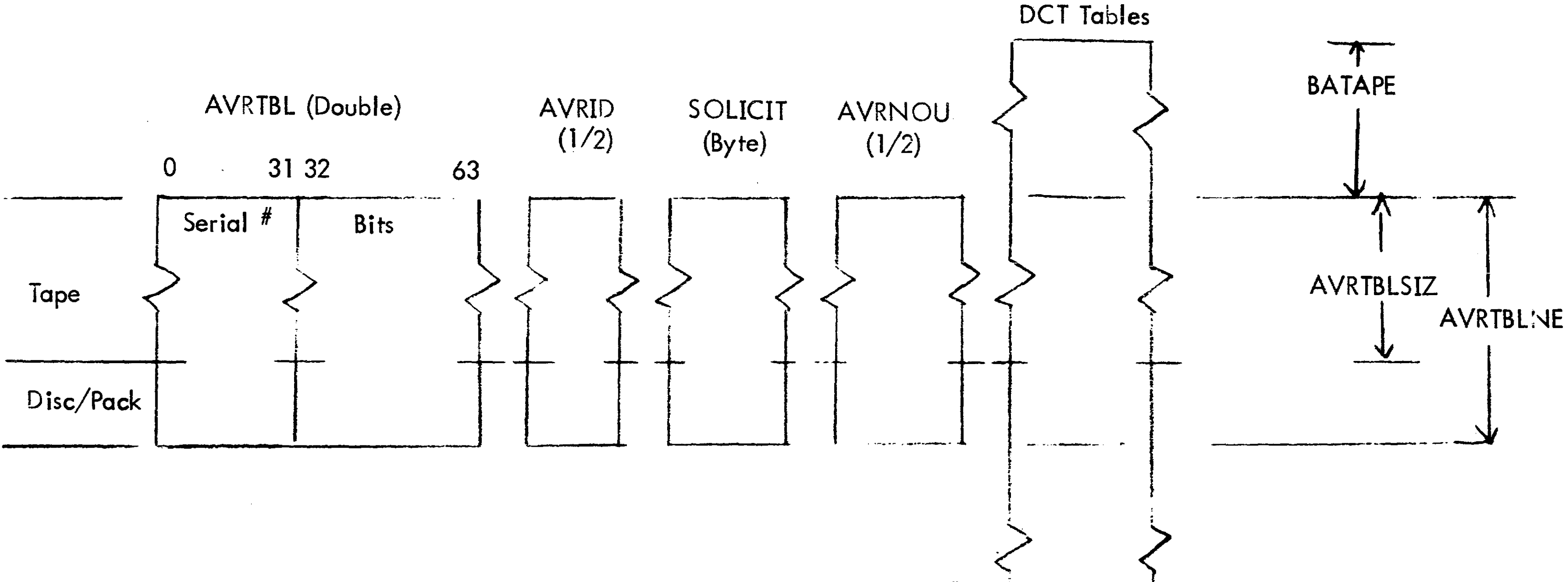
The following logical device filenames will always be present:

C1 - Input stream (card) -required to be device type CR

L1 - Output stream (printer) -default device type LP

P1 - Output stream (punch) -default device type CP or LP (if no punch on system)

AVR TABLES



TABLES (indexed by DCT#
minus BATAPE)

AVRTBL – See bits
AVRID – User #/SID s–1 = Lock
SOLICIT – 0 = No, $\neq 0$ = Yes
AVRNOU – # DCBs + # users for packs
tape spacing operations in progress for tapes

VALUES

BATAPE – first tape DCT index
AVRTBLSIZ – # tape entries
AVRTBLNE – # tape + # packs

AVR Table Bit Definitions

AVRTBL (AVRTBLSIZ=number of tapes, AVRTBLNE=tapes + private packs, BATAPE=first tape DCT index) (EQUed in PASS2)

Entry Format for Labeled Tapes

0 31	32					in use		39	40 47	48 63
Serial Number	PUB	POS	AVR	SCR	HLD	PTL	UPL	OPN	NOU	TPOS

Serial Number	Four-byte EBCDIC serial number of volume mounted.
PUB	Device public (1) or private (0).
POS	Tape at beginning (1) yes (0) no.
AVR	Volume serial number verified, yes (1) or no (0).
SCR	Scratch tape mounted, yes (1) or no (0).
HLD	Volume can be dismounted, yest (1) or no (0).
PTL	PTL option specified in last M:CLOSE, yes (1) or no (0).
UPL	Labeled tape, out or outin.
OPN	Tape position known by system, yes (1) or no (0).
NOU	Number of DCBs open to files on the volume.
TPOS	Number of tape marks between load point and present position of tape.
in use	Set when drive is allocated.

Entry Format for Direct Access Devices

0 31	32							39	40 47	48 63
Serial Number	PUB	▨	AVR	INIT	VER	MTD	PRIM	▨	NOU	HGPDISP

Serial Number	EBCDIC serial of current volume. This field and all other flags are reset to zero when system dismounts volume.
PUB	Device public (1) or private (0).
AVR	Volume serial verified and cylinder BIT MAPS moved to allocation table, yes (1) or no (0).
INIT	Private volume being initialized, yes (1) or no (0).
VER	Verification in progress (1) or not active (0).
MTD	Mount requested (pending), yes (1) or no (0).
PRIM	Volume mounted is primary volume in a private volume set, yes (1) or no (0).
NOU	Number of DCBs open to files on this volume.
HGPDISP	Word displacement from HGP to FWA of allocation table for device.

AVR TABLES

Table Content and Bit Settings

TAPE OR PACK STATUS	PUB	SERIAL	AVR NOU	AVR	AVR ID	SOLICIT	VER	SCR
Premount Public	1	#	1	0	0	0	0	0
Premount	0	#	0	0	#	0	0	0
Available (Empty)	0	0	0	0	0	0	0	0
Solicited	1 0	#	#	0	#	1	0 1	0
Being Verified	1 - 0	#	1	0	#	0	1	0
Dismount (lock)	1	#	#	1	-1	0	0	0
Private non shareable	0	#	#	1	#	0	0	0
Share PRIVATE	0	#	#	1	0	0	0	0
Public (operator)	1	#	#	1	0	0	0	0
System	1	0	0	0	0	0	0	0
Scratch	0	#	0	1	#	0	0	1
Premount SCRATCH	0	#	0	0	0, #	0	0	1
Solicited SCRATCH	0	-1	0	0	#	1	0	0

AVRSID

0 7		16 31
TYC		RBC

Used between end-action and user M:CHECK (or equiv.) for M:PRECORD operations.

TYC - I/O completion type } from register 12
RBC - Remaining record count } on user end action

AVRFNMT - Six Words

0 17	18	19 23
FILENAME	VSN	EXPIRATION

FILENAME	name of the first file on the tape volume (TEXTC).
VSN	volume sequence number (binary)
EXPIRATION	Julian expiration date (five bytes representing YYDDD in EBCDIC.

ANSFLGS - One Byte

0	1	2 3	4	5	6	7
AT	TC	ERR	AK	RO	BLP	MS

AT	tape is ANS (0 = no, 1 = yes).
TO	type of DCB (0 = NON-ANS, 1 = ANS)
ERR	error flags (00 = no error, 01 = NOT ANS, 10 = NOT EXPIRED, 11 = ANS VOL).
AK	ANSSCRATCH key-in processed (0=no, 1=yes).
RO	Access protection (0 = unrestricted, 1 = read only).
BLP	BLP option specified (0 = no, 1 = yes). If CPV labeled tape bit set implies no un-blocked records can be written.
MS	MOUNT OR SCRATCH (0 = Mount, 1 = Scratch)

	0	1	2	3		7
AVRFLGS	DDS	CCS	REW	ATO		RDK

where DDS is the Dual Density Status indicator

0 = density is 1600 bpi (PE)
1 = density is 800 bpi (NRZI)
DDS is only meaningful for output tapes where DD in TB:FLGS1 is 1.

CCS is the Code Conversion Status indicator

0 = no code conversion is performed
1 = code conversion is performed between EBCDIC (in core) and ASCII (on tape).
CCS is only meaningful for tapes where CC in TB:FLGS1 is 1.

REW is the rewinding bit, indicating a REW was initiated but the end action has not yet occurred.

0 = not rewinding
1 = rewinding - no other operations may start on this drive.

ATO is the Asynchronous Tape Operation flag.

1 = a tape spacing operation is in progress and data transfers are not to be initiated - AVRNOU contains the number of operations in progress.
0 = data transfers are OK.

RDK is the READ keyin flag.

1 = a READ keyin has previously been received for this user for this tape.

COC TABLES (indexed by Line Number) Length LNOL 'All are Interrupt Altered

COCOC (0–7): Count of remaining characters to be output

0 inactive
1 last character being output & this COCIO & COCOR are meaningless.

LB:UN (0–7): User No.

CPOS (0–7): Carriage Position

Indicates the current column number of the carriage.

CPI (0–7): Initial Carriage Position for a read

RSZ (0–7): User requested message size if a read is pending

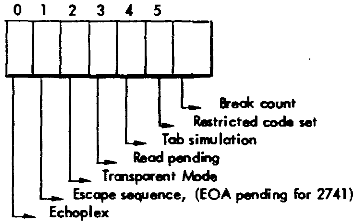
ARSZ (0–7): Current size of records being read (0 echoed) while read is pending

BUFCNT (0–7): Current number of buffers in use by the line

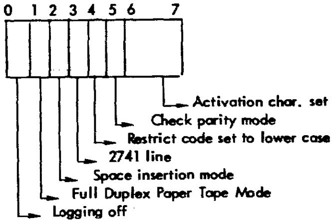
COCTERM (0–7): Terminal Type

0 = 33 teletype
1 = 35 teletype
2 = 37 teletype
3 = XDS Model 7015
4-11 = IBM 2741
12-15 = Installation-defined

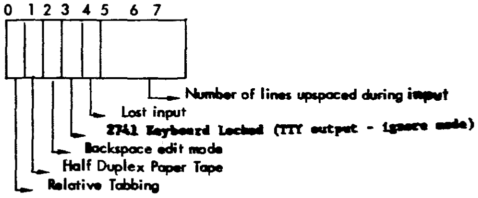
MODE



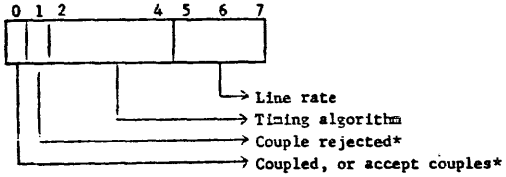
'MODE2



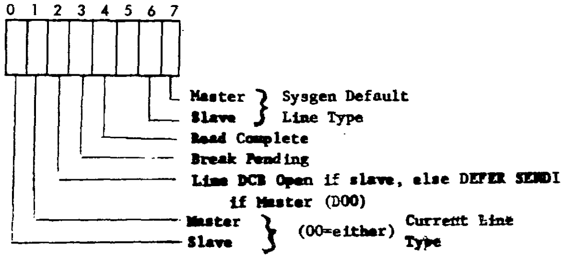
MODE3



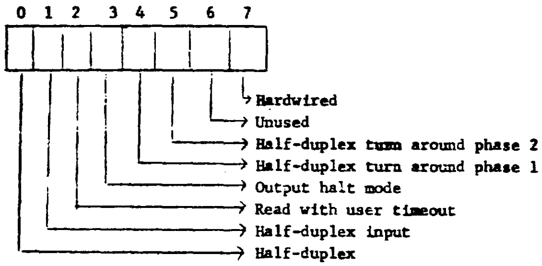
MODE4



MODE5



MODE6



TL (HW) (0–15): Link to the buffer for input tab simulation while a read is pending

A value of 8000 indicates no tabs in effect for the read.

If no read pending, the input buffer address where activation took place.

EOMTIME (HW) (0–15): Time activation was received. If read is pending, number of 1.2 second intervals till timeout.

TIE* (0–7): LINE # COUPLED TO

* Significant only if COUPLE option specified in PASS2

COCOI: Byte pointer to current insertion point into the output stream for the line.

COCOR: Byte pointer to the current removal point from the output stream for the line. 0 = no buffers.

COCII: Byte pointer to current insertion point into the input stream for the line. 0 = no buffers.

COCIR: Byte pointer to current removal point from the input stream for the line. 0 = no buffers.

TRANSLATION TABLES

TTYIN	ASCII to EBCDIC	For TTY
TTYOUT	EBCDIC TO ASCII	
EAPLLC	EAPL Lower Case to EBCDIC	For 2741
EAPLUC	EAPL Upper Case to EBCDIC	With EBCD
EAPL	EBCDIC to EAPL	APL type-ball
ESTDLC	ESTD Lower Case to EBCDIC	For 2741
ESTDUC	ESTD Upper Case to EBCDIC	With EBCD standard
ESTD	EBCDIC to ESTD	Type-ball
SAPLLC	SAPL Lower Case to EBCDIC	For 2741
SAPLUC	SAPL Upper Case to EBCDIC	With selectric
SAPL	EBCDIC to SAPL	APL type-ball
SSTDLC	SSTD Lower Case to EBCDIC	For 2741
SSTDUC	SSTD Upper Case to EBCDIC	With selectric
SSTD	EBCDIC to SSTD	Standard type-ball

DOUBLEWORD TABLES INDEXED BY COC NUMBER

COD:LPC	Word 0 contains the logical line number that corresponds to physical line number 0 on the COC. Word 1 contains the logical line number that corresponds to the last physical line on that COC.

WORD TABLES INDEXED BY COC NUMBER

CO:CMND	DA (command list)
CO:IIL	Input interrupt level select bit for WRITE DIRECT instructions
CO:LNM	Mask for obtaining line number; X'3F' for a COC, X'7F' for an FECP
CO:LST	BA (next character in ring buffer) - BA (first byte after end of ring buffer)
CO:OIL	Output interrupt level select bit for WRITE DIRECT instructions.
CO:OUT	WA (4 word output interrupt PSD block)
CO:RINGE	WA (first word after the end of the ring buffer)
CO:XPSDO	Output interrupt XPSD instruction

HALFWORD TABLES INDEXED BY COC NUMBER

COH:DN	COC device address
COH:II	Input interrupt location
COH:IO	Output interrupt location
COH:RBS	Byte size of ring buffer

BYTE TABLES INDEXED BY COC NUMBER

COB:SIOS	Condition codes from the SIO instruction in COCINIT

READ/WRITE DIRECT INSTRUCTIONS INDEXED BY COC NUMBER

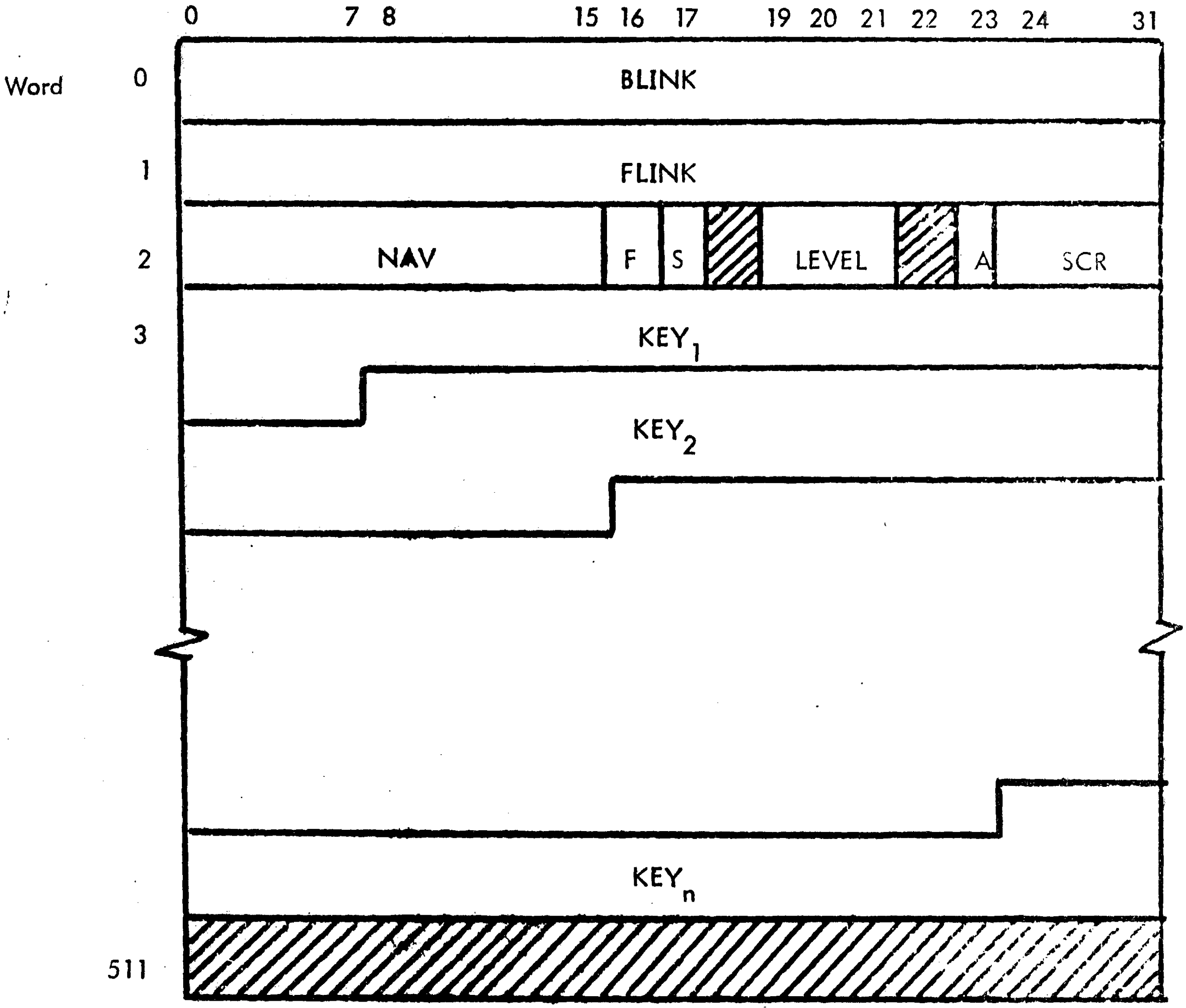
Table	Instruction		Description (d = DIO address)
CO:OUTRS	WD,7	.30d0	Output response
CO:STAT	WD,10	.30d0	Sense receiver status
CO:RCVOFF	WD,7	.30d2	Turn receiver off
CO:RCVON	WD,7	.30d1	Turn receiver on
CO:RCVDOFF	WD,7	.30d3	Turn receiver data set off
CO:XDATA	WD,6	.30d5	Transmit data
CO:TSTAT	WD,7	.30d4	Sense transmitter status
CO:XSTOP	WD,7	.30dE	Stop transmit

COC BUFFER LINKAGES

Free Chain:	COCHPB is the head pointer. FLINKs occupy word 0 of the buffer, and are word displacements from COCBUF to the next buffer.
Input, Output, and Tab Buffer Chains:	COCIR, COCOR, and TL are the respective head pointers. FLINKs occupy halfword 0 of the buffer, and are the byte displacement from COCBUF to byte 2 of the next buffer in the chain.

The level 0 index granule format is detailed below followed by the level 0 key entry

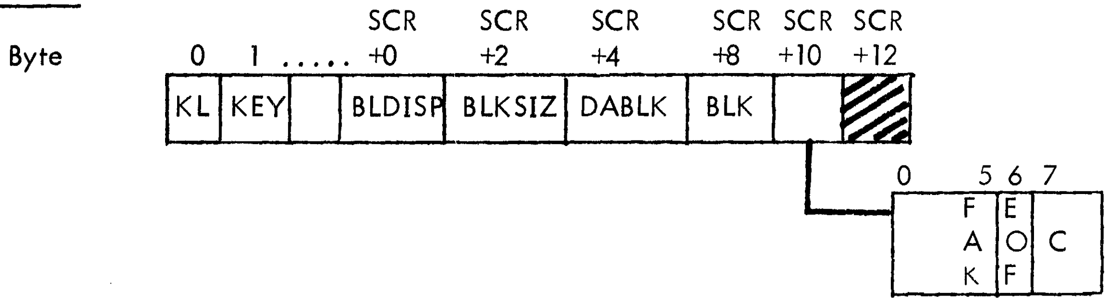
Master Index Format



where:

S =	1 if full-granule size; 0 if half-granule size
BLINK	contains the disk address of the preceding index granule, or zero, if none.
FLINK	contains the disk address of the next index granule, or zero, if none.
NAV	contains the number of significant bytes in the index granule (i.e., points to the next available byte in the index granule).
LEVEL	contains the level of the index entries (i.e., contains 0).
A	is the added flag and indicates whether or not this index half-granule has been added since the current higher level index structure was created (0 means no, 1 means yes).
KEY	is the key entry, shown below
SCR	1 + KEYM (see description on 14-38)
F =	1 if FIT is in this granule; 0, otherwise. The FIT will occupy the final 80 words, if present.

KEY ENTRY



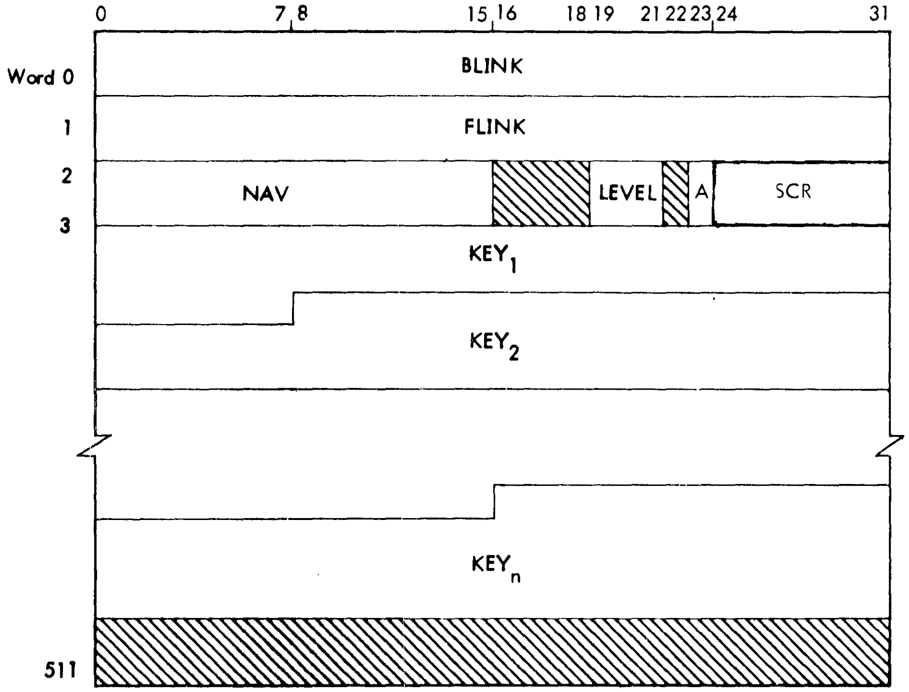
KL	contains the number of bytes in the key
KEY	contains the character string that indentifies the record
BLDISP	contains the byte displacement of the record segment associated with this key entry within the data granule pointed to by DABLK

BLKSIZ	contains the byte count of the initial size of the record segment associated with this key entry. BLKSIZ is never updated.
DABLK	contains the disk address of the data granule that contains the record segment associated with this key entry.
BLK	contains the count of the number of bytes of data currently in the record segment associated with this key entry (BLK $\leq$ BLKSIZ).
FAK	indicates whether or not this entry is the first appearance of this key (0 means no, 1 means yes).
EOF	indicates whether or not this entry is the last undeleted entry in the file (0 means no, 1 means yes).
C	indicates whether or not this record segment is the final segment of the record (0 means yes, 1 means no).

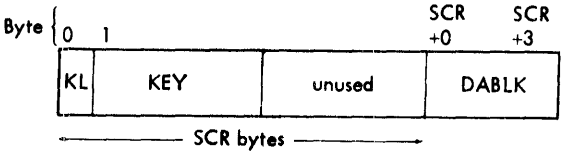
MASTER INDEX FORMAT FOR A HIGHER LEVEL INDEX

The formats of the higher-level granule and the higher-level key entry are detailed below.

Master Index



Key Entry



where

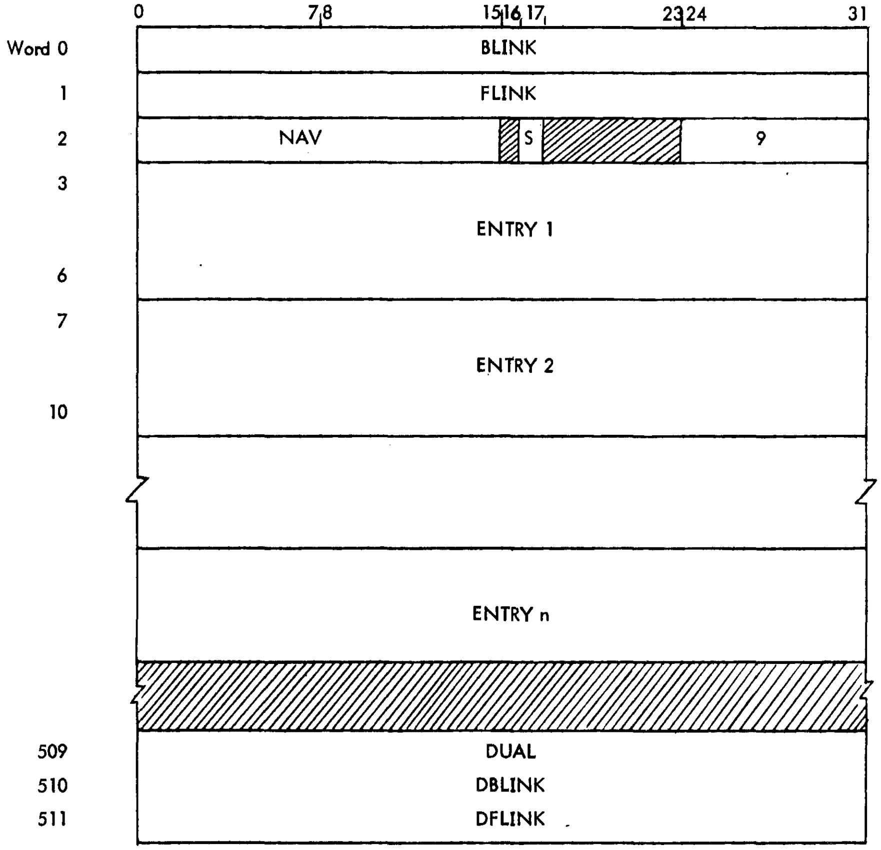
KL	contains the number of bytes in the key. If the key entry in the level 0 half granule pointed to by DABLK is not the first appearance of that key at level 0, the high order bit of KL (at level 1) is set.
KEY	contains the key from the first key entry in an index granule on the lower level.
SCR	is a field in the DCB and equals the maximum key length + 1.
DABLK	contains the disk address of the index granule on the lower level.
BLINK	contains the disk address of the preceding index granule at this level, or zero, if none.
FLINK	contains the disk address of the next index granule at this level or zero, if none.
NAV	contains the number of significant bytes in the index granule.
LEVEL	contains the level of the index entries (the lowest is level 0, the next is level 1, and so forth).
A	only meaningful at level 0; always 0 on higher levels.

ACCOUNT DIRECTORY

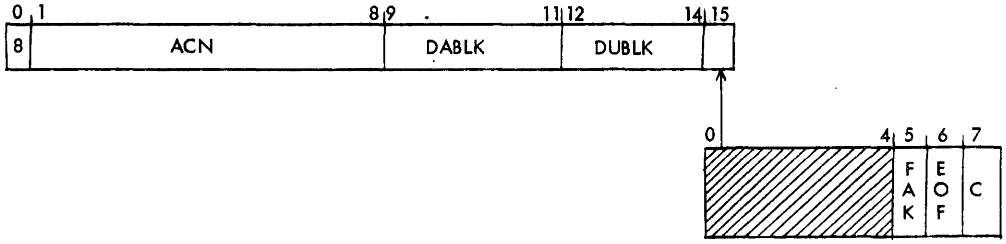
An account directory consists of a master index. An account directory index granule consists of entries that contain an account number and the disk addresses of the file directory associated with the account. There is one account directory for all public files in the system. Each private volume set has its own account directory and, for CP-V, only one account is allowed per volume set. The permanent information about the public file account directory is contained in the ACNCFU. ACNCFU is not used for private account directories. A private account directory is always located on Granule 1 of the primary volume in the set.

FORMAT FOR THE PUBLIC FILE ACCOUNT DIRECTORY

Master Index



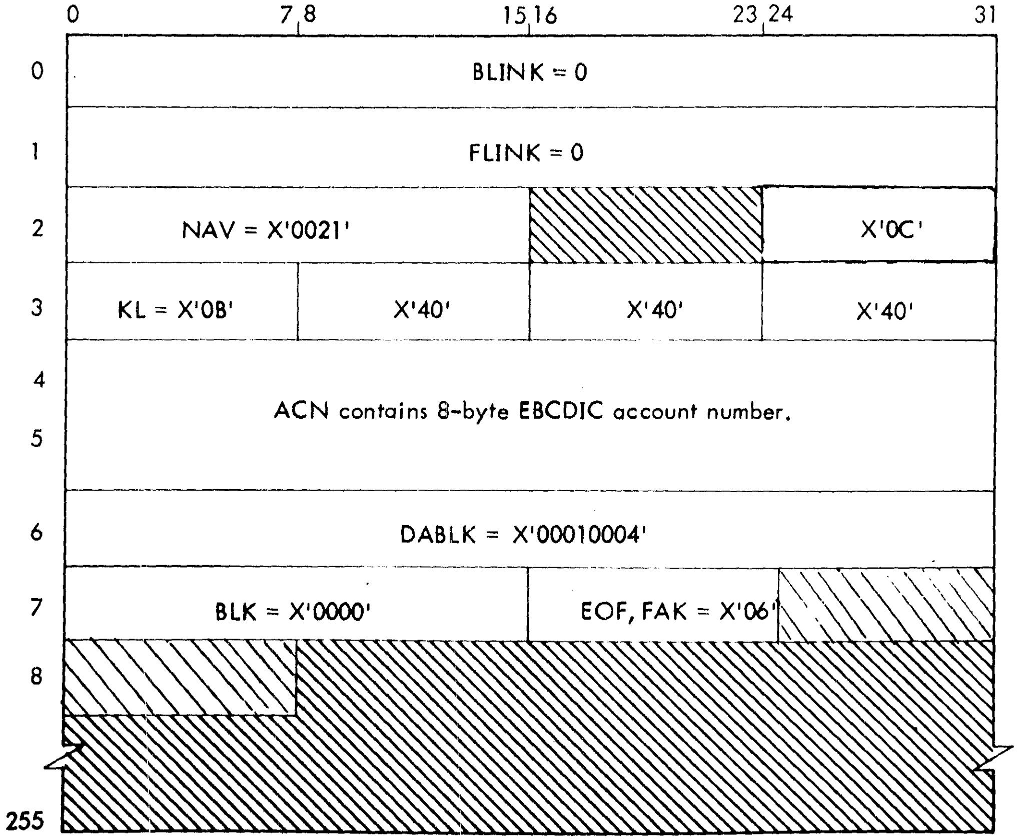
Entry Format



where

S	= 1 for full-granule size; S = 0 for half-granule size.
ACN	contains an eight-byte EBCDIC account number.
DABLK	contains the disk address of the file directory associated with this account.
DUBLK	contains the disk address of the dual of the first granule of the file directory in question.
FAK	= 1; indicates that this entry is the first appearance of this key.
EOF	indicates whether or not this key entry is the last in the file (0 means not EOF, 1 means EOF).
C	= 0; has no meaning for an account directory.
BLINK, FLINK, and NAV	have the same meaning as in the Master Index for Level 0.
DUAL	disk address of the dual location for current granule.
DFLINK, DBLINK	same as FLINK and BLINK except that these disk addresses apply to the dual structure for the directory.

FORMAT FOR A PRIVATE VOLUME SET ACCOUNT DIRECTORY

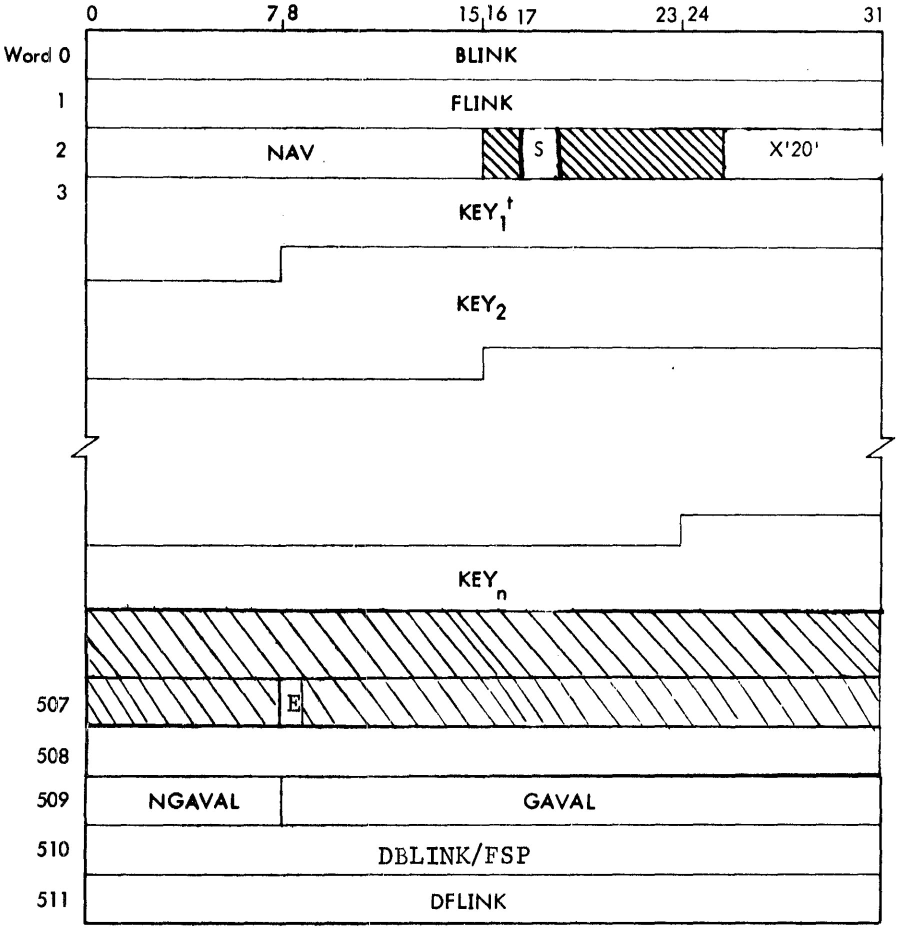


FILE DIRECTORY

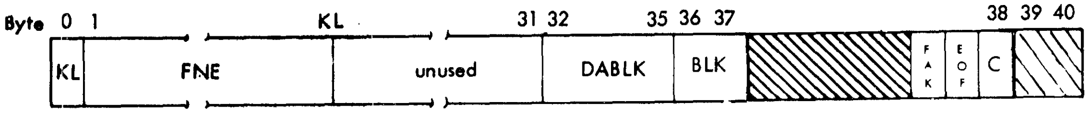
A file directory consists of a Master Index (MI) and a set of File Information Tables (FITs). A file directory index granule consists of key entries that contain the name of a file in the associated account and the disk address of the file's FIT. A FIT is located on a granule allocated to the file and contains all the information necessary to open a file. Information about the file directory itself (its mini-FIT) is contained in the last three words of the first block of its master index. The information from the mini-FIT is maintained in the FILCFU when the file directory is being updated. Public and private file directories have the same formats. However, since there is only one account per private volume set, there is also only one file directory. The file directory always begins on granule 2 of the primary volume in the set.

FORMAT FOR PUBLIC AND PRIVATE FILE DIRECTORIES

Granule Format



Key Entry Format



---

† The first entry in the file directory is a null entry for use with the NXTF option.

E	set if directory is empty.
FSP	disk address of free granule pool (only in first directory granule).

where:

S = 1	for full-granule size; =0, for half-granule size.
KL	contains the number of bytes in the key.
FNE	contains an EBCDIC file name.
DABLK	contains the disk address of the file's FIT.
BLK	Descriptors.

dynamic descriptors

bit 0 is 1
bit 1 is 0 in file directory; but set to 1 in DCB field if we are creating a new synonymous file.
bit 2 is 0 in file directory; but set to 1 in DCB field if a synonymous file is being processed
bit 3 is unused
bit 4 is 1 if the file has been modified since it was backed up by FILL
bit 5 is 1 if the file ahs been modified since it was last backed up by an Increment
bit 6 is 1 if the file has been modified since it was last backed up by a Saveall
bit 7 is 1 if the file has been modified since the last Fill

static descriptors

bit 0 is 1 if the file has a password
bit 1 is 1 if the file is a SYNON file
bit 2 is 1 if the file organization is random
bit 3 is reserved for expansion
bit 4 if 1 if the file is not to be automatically backed up
bit 5 is 1 if the access date is not to be updated
bit 6 is 1 if the file is not to be semi-automatically deleted
bit 7 is unassigned, but see the X'11' VLP, below

entry descriptors (unchanged)
bits 0-4 are not used
bit 5 is 1 (FAK)
bit 6 is 1 if this is the last FD entry for the account (EOF)
bit 7 is 0 (C)
FAK = 1; indicates that this entry is the first appearance of this key.

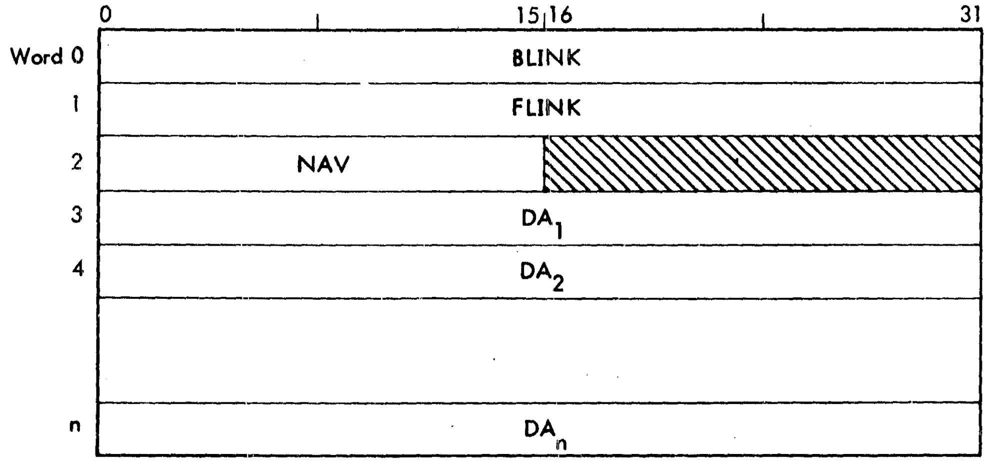
EOF	indicates whether or not this key entry is the last in the file (0 means no, 1 means yes).
C =	0; has no meaning for a file directory.

DBLINK, DFLINK, BLINK, FLINK, and NAV have the same meaning as is previous Account Directory formats. NGVAL,GAVAL, have the same meanings as in the FILCFU (see File Directory CFU (FILCFU) later in this chapter).

Words 509 and 510 as specified only for the initial granule of a directory; otherwise, they contain the same dual information as does an account directory granule.

FREE GRANULE POOL (FSP)[†] FORMAT

A Free granule Pool (FSP) is a collection of granules chained together with backward and forward links. An FSP granule contains the disk address of available granules that have been previously used by the File or Account Directory to which the FSP belongs. The chained granules that contain the disk addresses are also available. When the associated File or Account Directory requires more space, granules will be allocated from the FSP.



where

BLINK contains either the disk address of the preceding granule in the FSP or zero, if none

FLINK contains either the disk address of the next granule in the FSP or zero, if none

NAV contains the word length of the significant data on the granule (i.e., points to the next available word).

DA contains the disk address of a free granule allocated to the directory.

[†] Prior to B00, FSP meant Free Sector Pool because master index blocks were contained in sector units instead of half-granule units.

FILE INFORMATION TABLE (FIT) FORMAT FOR AN ORIGINAL FILE

0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22	23 24 25 26 27 28 29 30 31
(9 words) FNE (in TEXTC format)			
X'03'	LE = 0	NDW = 2	NAW = 2
PASSWORD (2 words)			
X'05'	LE = 0	NDW	NAW
Read ACNs (2 words each)			
X'06'	LE = 0	NDW	NAW
Write ACNs (2 words each)			
X'14'	LE = 0	NDW	NAW
Execute ACNs (2 words each)			
X'15'	LE = 0	NDW	NAW
Execute Vehicles (3 words each, TEXTC)			
X'04'	LE = 0	NDW = 2	NAW = 2
Expiration Date			
X'OF'	LE = 0	NDW = 2	NAW = 2
Last Access Date			
X'10'	LE = 0	NDW = 2	NAW = 2
Backup Date			
X'OE'	LE = 0	NDW = 2	NAW = 2
Creation Date			
X'OA'	LE = 0	NDW = 3	NAW = 3
Modification Date			
X'OD'	LE = 0	NDW = 1	NAW = 1
File Size			
X'OC'	LE =0	NDW = 7	NAW = 7
E	FDA		
TDA			
NGAVAL	GAVAL		
CCBD		0 ▨	SLIDES
▨	▨	▨	▨
SREC			
LDA			
X'09'	LE = 1	NDW = 3	NAW = 3
ORG	KEYM	▨	▨ [illegible]
LSLIDES		LRDLO	SPARE
NSF		10000000	$S_0S_1S_2S_3S_4S_5S_6S_7$

These coded entries are optional; presence of the entry is indicated by the byte 0 hex code.

See CFU format for meaning of fields

See DCB for meaning of fields.

The maximum size of the FIT is 80 words. The only limit on the number of read, write, execute accounts and execute vehicles is the 80 word maximum FIT size.

*The FIT starts in word 4 for consecutive files and in the 80th word from the end for keyed or random files.

Field	**Description**
ACN	is an account number. Each ACN is an eight-byte EBCDIC entry with trailing blanks. If there is no Read ACN entry, and ACN can read the file. If there is no Write ACN, no one can write in the file except the ACN that created the file
CCBD	contains, for keyed tiles, either the byte displacement to the next available byte in the last data granule of the file (SREC), which means that the blocking buffer was truncated; or 0, which means that the last data granule in the file (SREC) contains 512 words.
CYL	specifies whether the file assigned to the DCB is to be allocated by granules or cylinders (0 = granule allocation, 1 = cylinder allocation). It is only meaningful for public files.
Date	is of the form MMDDHHYY, where MM is numerical month DD is day of month HH is hour of day YY is last two digits of the year, all in EBCDIC bytes Expiration date may contain the word NEVER followed by three blanks, which indicates that the file does not have an expiration date. The modification date contains three words. The third word is of the form HHMM, where HH is a repeat of the hour MM is the minute
E = 1	file contains no record
FDA	contains the disk address of the file's first index granule at level 0.
File size	contains the current number of index and data granules allocated to keyed and consecutive files; or the number of data granules allocated for random files.
FNE	is the EBCDIC name of the file in TEXTC format.
GAVAL	contains the disk address of the next available granule in the last cylinder allocated to the file; zero if none.
KEYM	contains 1. the maximumlength, in bytes, of the keys in the file. Applicable to keyed files. Maximum value is 31. 2. the type of device that the random file is to be allocated on (0 = allocate on either RAD or DP, X'7' = allocate on RAD, X'B' = allocate on DP). Applicable to random files.
LDA	contains the disk address of the file's last index granule at level 0.
LRDLO	contains the limiting number of contiguous index granules that can be allocated in level 0 and not be reflected in level 1 before the flag, which signals CLOSE to reconstruct the higher level index structure, is set (i.e., before SLIDES in the CFU is set equal to 255).
LSLIDES	has meaning only if a multilevel index exists and contains 1. the limiting number of index granules that can be allocated in level 0 and not be reflected in level 1 before the flag, which signals CLOSE to reconstruct the higher level index structure, is set. 2. the value 255, which means that once a higher level index structure exists, it is not to be reconstructed. LSLIDES is only used for keyed files.
NAW	is the number of available words in the entry (not including the control word).
NDW	is the number of significant data words in the entry (not including the control word).

NGAVAL	is the number of available granules in the last cylinder allocated to the file
NOSEP	specifies whether or not granules are to be allocated on a specific device type. This flag has no meaning for private or random files.
NSF	is the number of files synonymous with this file.
O	is a level 1 flag indicating whether or not a level 1 index exists in a keyed file (0 = no, 1 = yes).
ORG	is the file organization indicator (0 = non specified and is treated as consecutive, 1 = consecutive, 2 = keyed, 3 = random).
PASSWORD	is an eight-byte EBCDIC entry with trailing blanks.
SLIDES	contains, for keyed files, either 1. a tally of the number of index granules allocated at level 0 since the current multilevel index structure was created, or if non exists, since the file was first opened. 2. a tally of the number of index granules allocated at the current level while the multilevel index structure is being (re) created. 3. the value 255, which means that a new multilevel index structure should be built when the file is closed (unless LSLIDES in the DCB equals 255 and a level-1 index exists).

Field	Description
LE	is the last entry flag and indicates whether or not this parameter is the last entry in the FIT (0 means no, 1 means yes).
SPARE	contains the number of spare byte positions to be left unused in the end of the current index granule in the event that the key to be added is the last key in the file. SPARE is only used for keyed files.
SREC	contains the disk address of the last data granule in the file. It is only used in the output mode.
TDA	contains, for keyed files, either 1. the disk address of the first index granule at the top of the multilevel structure, if one exists. 2. the disk address of the middle index granule, if there are three level-0 index granules and the file is keyed. 3. 0, which means that either the file is consecutive, or that the file is keyed and there are at the most two index half-granules. For consecutive files, TDA contains the number of records in the file.
$S_0 = 1$	if file has a password, = 0 otherwise
$S_1 = 0$	
$S_2 = 1$	if file organization is random
$S_3 = 0$	
$S_4 = 1$	if file is not to be automatically backed up
$S_5 = 1$	if access date is not to be updated
$S_6 = 1$	if file is not to be semi-automatically deleted
$S_7 = 0$	

FILE INFORMATION TABLE (FIT) FORMAT FOR A SYNONYMOUS FILE

0 7	8 15	16 23	24 31	
01	LE = 1	NDW = 8	NAW = 8	
FNE (TEXTC Format)				
0B	LE	NDW	NAW	
SYNON FILE NAME (TEXTC Format)				*

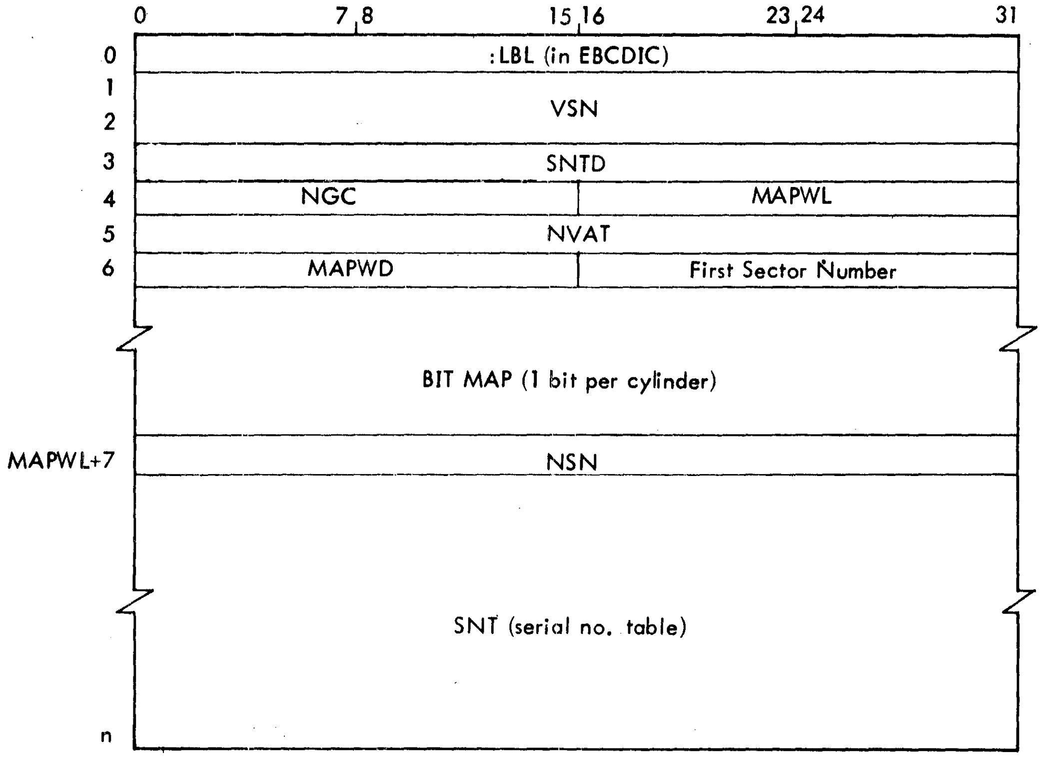
A synonymous file does not have a FIT, but if a synonymous file is accessed on a NXTF open with FPARAM, an X'08' error return is made with the above information passed as the FPARAM.

* This entry replaces the optional entries on the original file.

PRIVATE VOLUME SET TABLES

A private volume set is a collection of disk pack volumes that the user has grouped together, containing any number of files with any type of organization. All files on a private volume set belong to the same account. A private volume set is a self-contained entity and contains its own account directory (with one entry) and file directory (which contains information about all files on the private volume set). The account directory for a private volume set is located on granule 1 of the primary volume, and the file directory begins on granule 2. Because volumes can be mounted serially for consecutive files, the file directory cannot extend beyond the primary volume. The formats for the private account and file directories are the same as for the public directories. Every private volume has a Volume Table of Contents (VTOC) which is built by the VOLINIT processor and is located on granule 0 of the volume. The format of the VOTC is shown on the following page.

VOLUME TABLE OF CONTENTS (VTOC) FORMAT



VOTC Fields

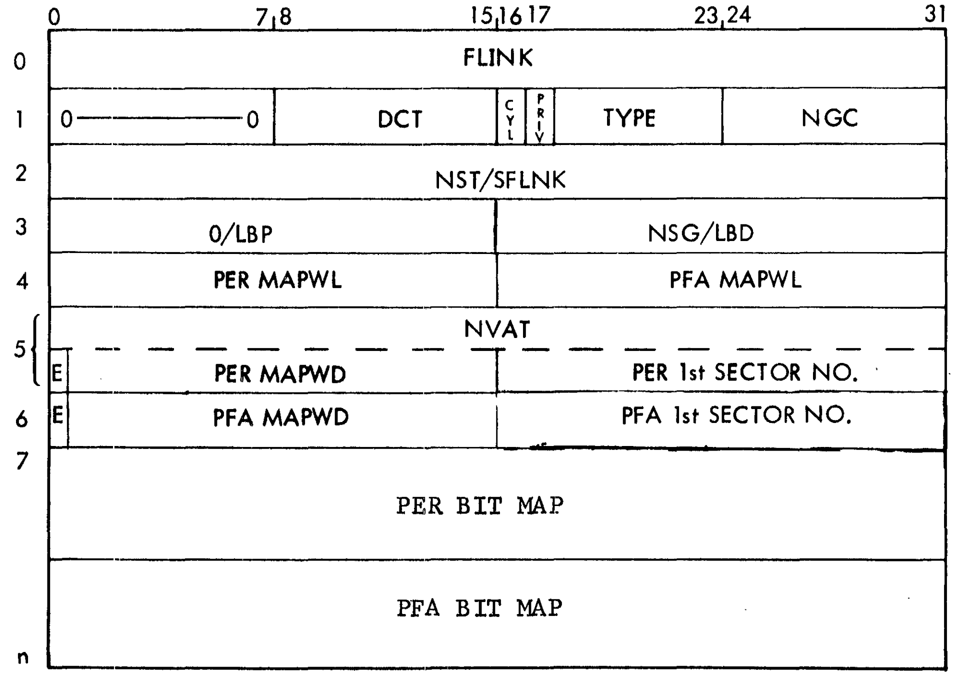
Field Name	Word	Meaning
Bit Map	7+n	contains the cylinder bit map for the volume. If the bit is set, the cylinder is unallocated. If the bit is reset, the cylinder is allocated. For devices where the number of cylinders is not an even multiple of 32, VOLINIT resets the bits in the last word which do not represent cylinders.
First Sector Number	6	contains the sector number for the first sector on the device (i.e., sector 0).
MAPWD	6	contains the word displacement from the start of an allocation table to the bit map. MAPWD is set to X'07' by the VOLINIT processor.
MAPWL	4	contains the number of words in the bit map.
NSN		contains the number of serial numbers in the Serial Number Table.
NGC	4	contains the number of granules per logical cylinder.

Field Name	Word	Meaning
NVAT	5	contains the next volume's cylinder 0 allocation table. Each bit represents a granule. If the bit is set, the granule is unallocated. If the bit is reset, the granule is allocated. The first bit, which represents granule 0 is always set allocated. NVAT is only used when a consecutive file extends beyond volume boundaries.
SNT		contains the serial numbers of all the volumes in the private volume set for the primary volume. The position of a serial number represents its volume number. (The primary volume is always volume 1.) The order of the serial numbers never changes, but new volumes can be added to the set, and will be added at the end. For other volumes, this table contains only the serial number of the primary volume. Only the first four bytes of the serial number are carried in the serial number table.
SNTD	3	contains the word displacement from the start of the VTOC to the start of the Serial Number Table.
VSN	(1&2)	contains the volume serial number (left-justified and blank-filled to eight bytes.)

ALLOCATION TABLES (HGP)

SYSGEN creates an allocation table (called HGP - Heading Granule Pool for each disk device (RAD or disk pack) defined at SYSGEN time. The HGPs are located in ALLOCAT's data (See Section GB) - ALLOCAT) and in CORE starting at the location HGP. (The bit maps for public devices are not in the CORE HGP Tables.) Allocation tables are used to allocate and release granule/cylinder units from symbiont storage areas of a device and the file. An allocation table contains (1) information about the file and symbiont storage areas of a device, (2) the relationship between the device's physical (sector and track) and logical (granule or cylinder) units, and (3) the maps which control the allocation of the granule/cylinder units in the file and in the symbiont storage areas of the device. The system device's allocation table is always the first one and begins at the location HGP. The allocation table format (HGP) is shown below.

Allocation Table Format (HGP)



where

CYL indicates whether the device is allocated by cylinder or granule (0 means granule, 1 means cylinder). All private devices are allocated by cylinder. Disks can only be allocated by granule. Symbiont storage only exists on devices allocated by granule.

DCT	contains the DCT index of the device the allocation table is associated with.
E	if set, indicates that this portion of the bit map (PER or PFA) is all zeroes (ALLOCAT data).
FLINK	contains the address of the next allocation table, or zero if none. HGP equals the address of the first allocation table in the chain.
LBP, LBO	contains the bit position and word displacement of the last single granule allocated from the bit map (ALLOCAT data only).
NGC	contains the number of granules per cylinder, but only has meaning if CYL is set.
NSG	contains the number of sectors per 512 - word granule for this device (CORE headers only).
NST	contains the number of sectors per track for the device (CORE headers only).
NVAT	contains the next volume's cylinder 0 allocation table (if PRIV is set). Each bit represents a granule. If the bit is set, the granule is unallocated. If the bit is reset, the granule is allocated.
PER bit map	is a map in which each bit represents a granule in the symbiont storage area. (Symbionts are only allocated on devices with granule allocation units). The bits are ordered left to right within a word so that bit 0 of the first word represents the first granule in the symbiont storage area, bit 0 of the second word represents the 32nd granule in the storage area, and so forth. If the bit is set, the granule is unallocated, if the bit is reset, the granule is allocated.
PER MAPWD	contains the word displacement from the start of this allocation table to the first word of the bit map for the symbiont storage area.
PER MAPWL	contains the number of words in the bit map for the symbiont storage area.
PER first sector number	contains the sector number of the first track in the symbiont storage area. (Sectors are numbered starting with 0.)
PFA bit map	is a map in which each bit represents a granule or cylinder (depending upon the CYL flag) in the file storage area. The bits are ordered left to right within a word so that bit 0 of the first word represents the first granule/cylinder in the file storage area, but 0 of the second word represents the 32nd granule/cylinder in the storage area, and so forth. If the bit is set, the granule/cylinder is unallocated/ if the bit is reset, the granule/cylinder is allocated.
PER MAPWD	contains the word displacement from the start of this allocation table to the first word of the bit map for the symbiont storage area.
PER MAPWL	contains the number of words in the bit map for the symbiont storage area.
PER first sector number	contains the sector number of the first track in the symbiont storage area. (Sectors are numbered starting with 0.)
PFA bit map	is a map in which each bit represents a granule or cylinder (depending upon the CYL flag) in the file storage area. The bits are ordered left to right within a word so that bit 0 of the first word represents the first granule/cylinder in the file storage area, but 0 of the second word represents the 32nd granule/cylinder in the storage area, and so forth. If the bit is set, the granule/cylinder is unallocated; if the bit is reset, the granule/cylinder is allocated.
PFA MAPWD	contains the word displacement from the start of this allocation table to the first word of the bit map for the file storage area.
PFA MAPWL	contains the number of words in the bit map for the file storage area.
PFA first sector number	contains the sector number of the first track in the file storage area. (Sectors are numbered starting with 0).
PRIV	indicates whether the device is public or private (0 means public, 1 means private).
SFLNK	contains the address of the next allocation table in the circular chain of similar usage devices (ALLOCAT data only).
TYPE	contains the device type (7 means disk; B means disk pack).

CURRENT FILE USAGE (CFU) TABLES

The CFU tables contain information about the currently open disk files. Each open file, whether public or private, has a CFU table. In the case where an old version of a file exists and a new version is being created, the two versions are considered as different files until the new version is closed, at which time it replaces the old version. If the old version is opened in the input mode at the same time the new version is opened in the OUT or OUTIN mode, each will have a different CFU, but the CFUs are linked together and each is called the secondary CFU of the other. SYSGEN creates a pool of user CFU buffers from the number specified in the CFU option of the Monitor's command. In addition, SYSGEN creates two SYSGEN CFUs called ACNCFU and FILCFU, which immediately precede the user CFU pool. ACNCFU contains information about the public file account directory. FILCFU contains information about the currently referenced public or private file directory. The CFU tables are core-resident in the data area of the Monitor's root. The constants ACNCFU and FILCFU equal the addresses of the account and file directory CFUs respectively. The constant BGRCFU equals the address of the first user CFU buffer, and the constant LASTCFU equals the address of the last user CFU buffer.

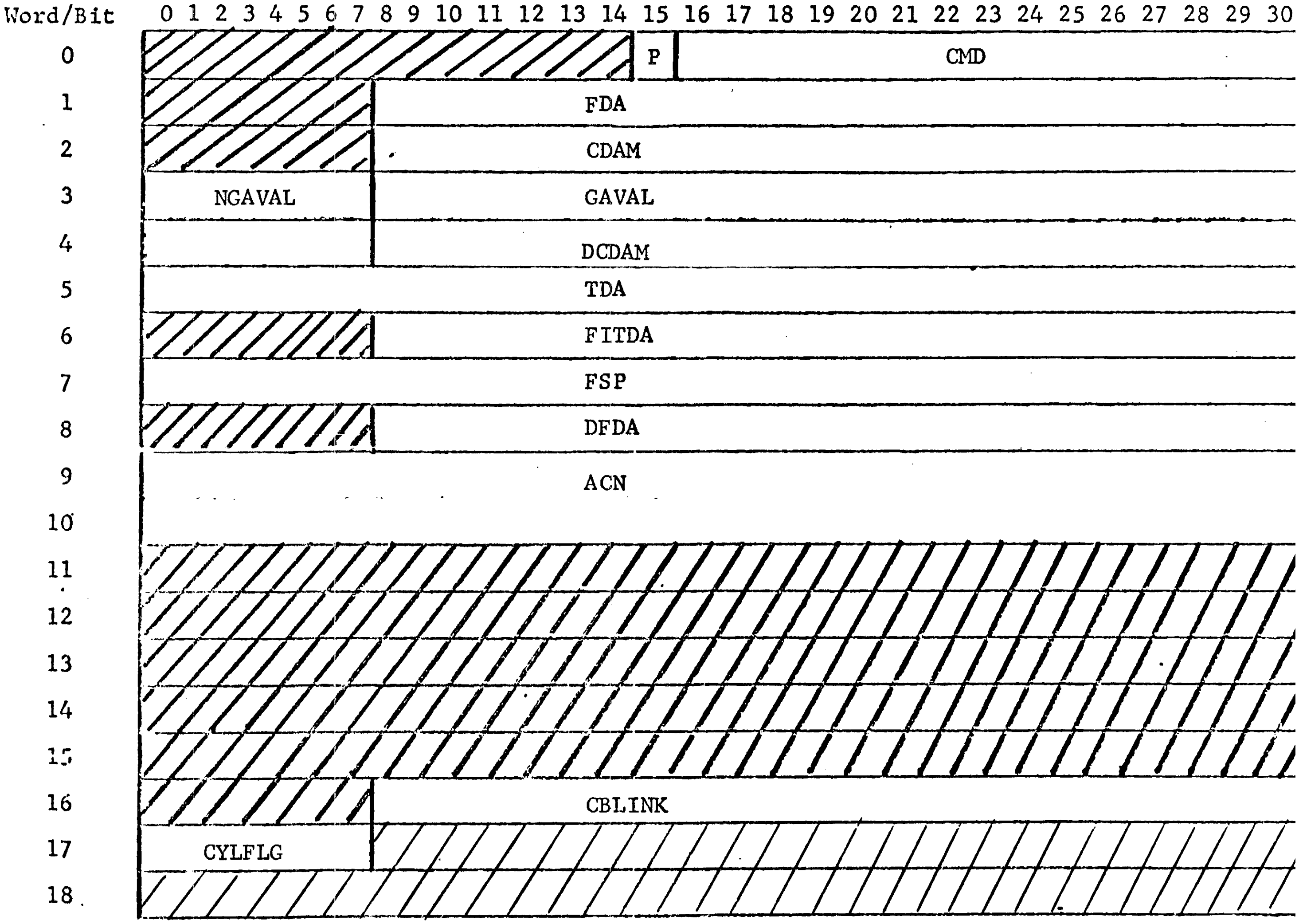
The Account CFU (ACNCFU)

Word/Bit	0	1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
0	NEWFD			
1	E	▨	FDA	
2	▨		CDAM	
3	NGAVAL		GAVAL	
4	▨		DCDAM	
5	TDA			
6	▨		RDA	
7	▨		DRDA	
8	▨		DFDA	
9	▨		DIGRAN	
10	▨		DBUFF3	
11	CC1		▨	
12	CC2		▨	SIGCLS
13	ACCTTBL			
14	ACCTCNT			▨
15	NAMTBL			
16	NAMTBLE			
17	▨ FREE CFU ▨			
18	REDFLGS			

FIELD	WORD	MEANING
ACCTTBL	13	is the word address of the start of the account name entries in the user CFU area.
ACCTCNT	14	is a count of the number of account name entries currently in the user CFU area.
CDAM	2	disc address of the granule of the account directory currently being processed.
DCDAM	4	disc address of the dual of the granule of the account directory currently being processed.
DBUFF3	10	disc address of the dual of the granule currently being processed in BUFF3 in the INST routines in WRTF.
DFDA	8	disc address of the dual of the FDA (see below).
DIGRAN	9	disc address of the dual supplied by the GETIGRAN routine in WRTF.
DRDA	7	disc address of the dual of the RDA (see below).
E	1	is the empty directory flag and is only one for a short period at the start of a cold boot of a PO tape.
CC1	11	set by a STCF in the FINDFIL routines in OPN and CLS after determining whether (equal) or not (not equal) the FILCFU is set up for the directory required.
CC1	12	set by a STCF in the FINDFIL routines in OPN and CLS indicating whether (equal) or not (not equal) the directory to be processed is that of :SYS.
FDA	1	disc address of the start (1st granule) in the account directory.
GAVAL	3	disc address of the next granule available from the cylinder most recently allocated to the account directory. Unless the system contains public cylinder devices, this field and NGAVAL are always zero.
FREECFU	17	address of the last 8-word block set up by CLOSE. Whenever a disk file is closed, the address in FREECFU is decremented by 8. If the CFU to which it now points is not active, the CFU for the file being closed is moved to this CFU and the original CFU is clobbered. If the CFU being pointed to is active, FREECFU is reset to the address of the last 8-word block.

FIELD	WORD	MEANING
NGAVAL	3	the number of granules still unallocated from the cylinder most recently allocated to the account directory.
NAMTBL	15	word address of the start of the file name entries in the user CFU area.
NAMTBLE	16	word address of the next available word in the user CFU area into which a file name can be placed.
NEWFD	0	is a flag used in CLS to indicate whether or not a new file directory is being created.
RDA	6	disc address of the granule requested to be read by the REDSEC routines in RDF.
REDFLGS	18	control flags used by the REDSEC routines in RDF.
SIGCLS	12	flag indicating whether (not zero) or not (zero) a CFU with a name has been released since the last restructuring of the user CFU area.
TDA	5	always zero indicating that no multi-level structure exists.

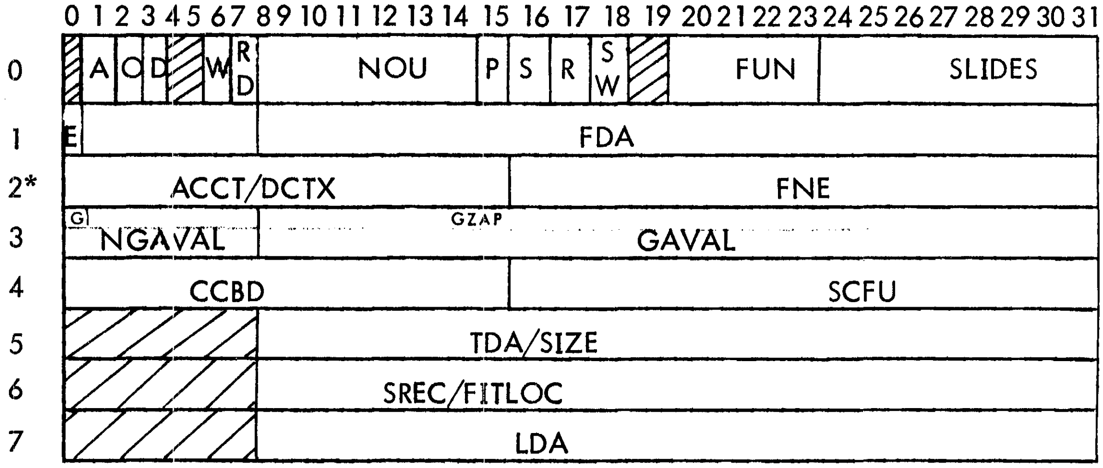
The File CFU (FILCFU)



where:

FIELD	WORD	MEANING
ACN	9,10	the eight characters of the account name of the current file directory.
CBLINK	16	the blink of the granule indicated by CDAM.
CDAM	2	disc address of the granule of the file directory currently being processed.
CMD	0	displacement in the appropriate account directory granule of the entry for the current file directory. This field is only used when a new file directory is being created.
DCDAM	4	dual disk address of the granule pointed to by CDAM.
DFDA	8	disc address of the dual of the FDA (see below).
FDA	1	disc address of the start (1st granule) of the file directory.
GAVAL	3	disc address of the next granule available from the cylinder most recently allocated to the file directory. This field contains zeros if there are none left from the last cylinder or if the directory is being allocated on a granule device.
NGAVAL	3	the number of granules still unallocated from the cylinder most recently allocated to the file directory.
FITDA	6	disc address of the file information table during OPN and CLS.
FSP	7	disc address of the start of the free sector pool or zero.
NFD	17	number of granules in the file directory.
NFIT	17	number of random files in the current account.
NFSP	18	number of granules in the free sector pool.
P	0	is the private flag indicating whether (one) or not (zero) the current file directory is from a private pack set.
TDA	5	always zero indicating that no multi-level structure exist

User CFU



Field Name	Word	Meaning
A	0	Active flag, indicating whether or not a DCB is associated with this CFU (0 means no, 1 yes).
ACCT	2	For non-star public files, this is a doubleword index to a table of account names. The base address of the table is in ACNCFU+13.
CCBD	4	For keyed files only, either the byte displacement to the next available byte in the last data granule (SREC) which means that the blocking buffer was truncated, or 0.
D	0	Indicating whether (=1) or not (=0) to release the granules of the file during close.
DCTX	2	For private pack files, the DCT index of the primary volume as mounted.
E	1	For keyed and consecutive files, =1 if file contains no records, 0, otherwise.
FDA	1	For keyed files, the disk address of the first level 0 Master index granule. For random and consecutive files, the disk address of the first granule of the file.

FNE	2	For non-star files, this is a word address to a table of file names in TEXTC form.
FITLOC	6	For random files, the disk address of the file information table (FIT).
FUN	0	The function of the DCBs associated with this CFU (1 = In, 2 = OUT, 4 = INOUT, 8 = OUTIN).
G	3	For random files, if set indicates a newly allocated file being cleaned.
GAVAL	3	The disk address of the next available granule in the cylinder most recently allocated to this file or zero.
GZAP	3	For random files, the address of the monitor buffer in which the cleaning data is kept. Only meaningful if G is set.
LDA	7	For keyed files, the disk address of the final level zero master index granule. For consecutive files, the disk address of the final granule of the file.
NGAVAL	3	Tally of the number of available granules in the cylinder most recently allocated to this file.
NOU	0	Tally of the number of DCBs currently associated with this CFU.
O	0	For keyed files, a flag indicating whether (= 1) or not (= 0) a level above the level 0 master index exists.
P	0	Private flag, indicating whether (= 1) or not (= 0) the file associated with this CFU is on a private pack.
R	0	Random flag, indicating whether (= 1) or not (= 0) the file associated with this CFU has random organization.
RD	0	Read has occurred flag, indicating whether (= 1) or not (= 0) a read CAL has been executed for the file associated with this CFU during this open.
S	0	Shared flag, indicating whether (= 1) or not (= 0) the DCB (s) associated with this CFU have the share specification.
SCFU	4	Word address of the secondary CFU (if any) associated with the current CFU.
SIZE	5	For consecutive files, a tally of the number of records currently existing in the file. For random files, a tally of the number of granules in the file.
SLIDES	0	For keyed files only, a tally of the number of master index granules at level 0 which are not reflected in a higher level index. If this value is 255, it indicates that a threshold has been exceeded which indicates that the upper level indices should be (re) built.
SREC	6	For keyed files only, the disk address of the final data granule in the file.
SW	0	Shared write flag, indicating whether (= 1) or not (= 0) a modification to the file is currently in progress.
TDA	5	For keyed files only either the disk address of the top of the upper level index structure if the upper exists, or zero.
W	0	Write flag, indicating whether (= 1) or not (= 0) the file contents have been modified during this open.

This appendix contains the formats for the three kinds of DCBs created by the monitor: files, devices, and labeled tape. Following each format, the parameter fields of the DCB are described in alphabetical sequence by their mnemonic. All referenced addresses have word resolution unless otherwise specified.

FILE DCB

Figure A-1 shows the format of the DCB for consecutive, keyed, and random files. All single fields are applicable to the three kinds of files. Fields shown with a heavy border depict differences between consecutive, keyed, and random. Shaded fields are not used by the DCB.

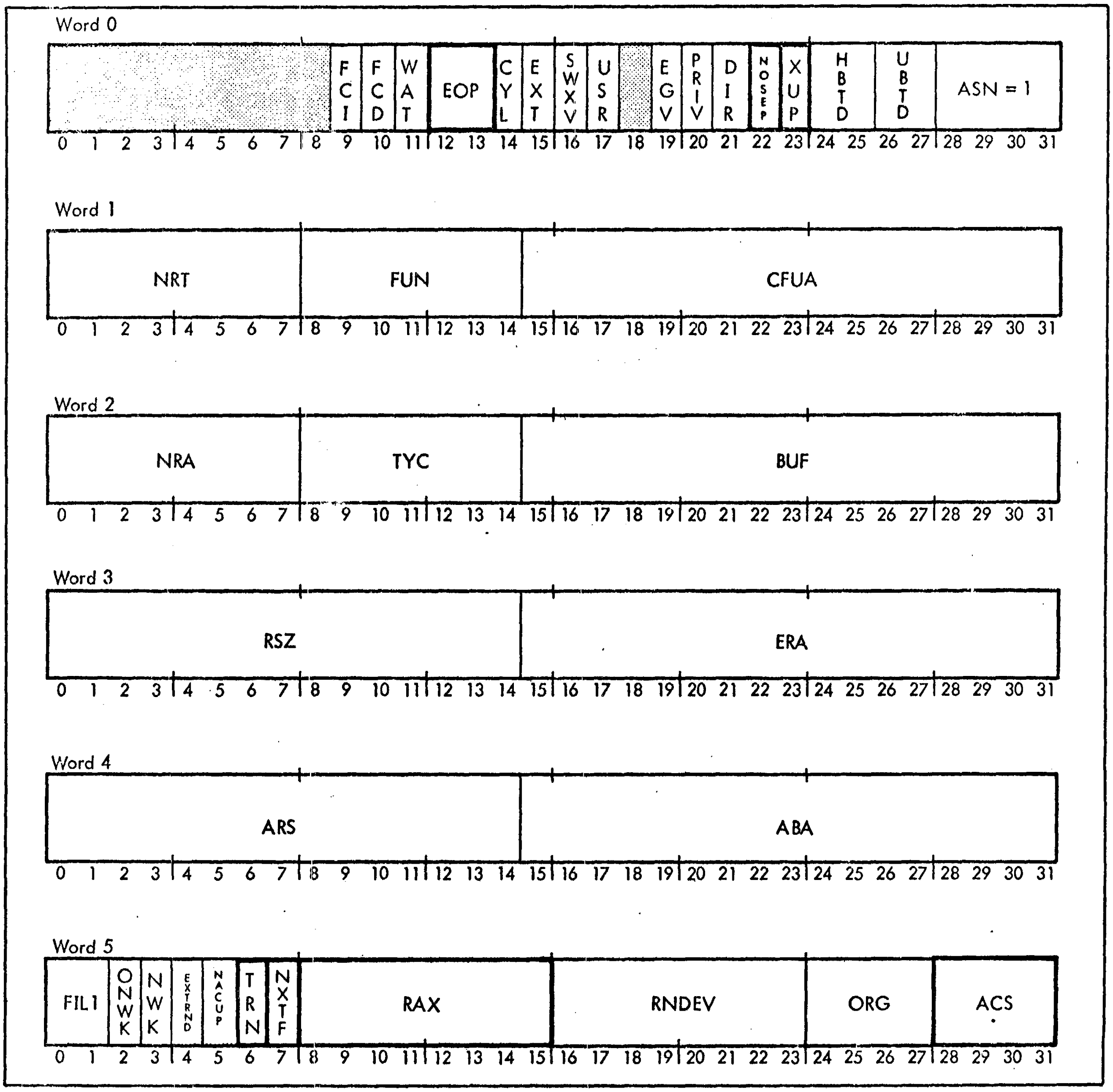


Figure A-1. Format of File DCB

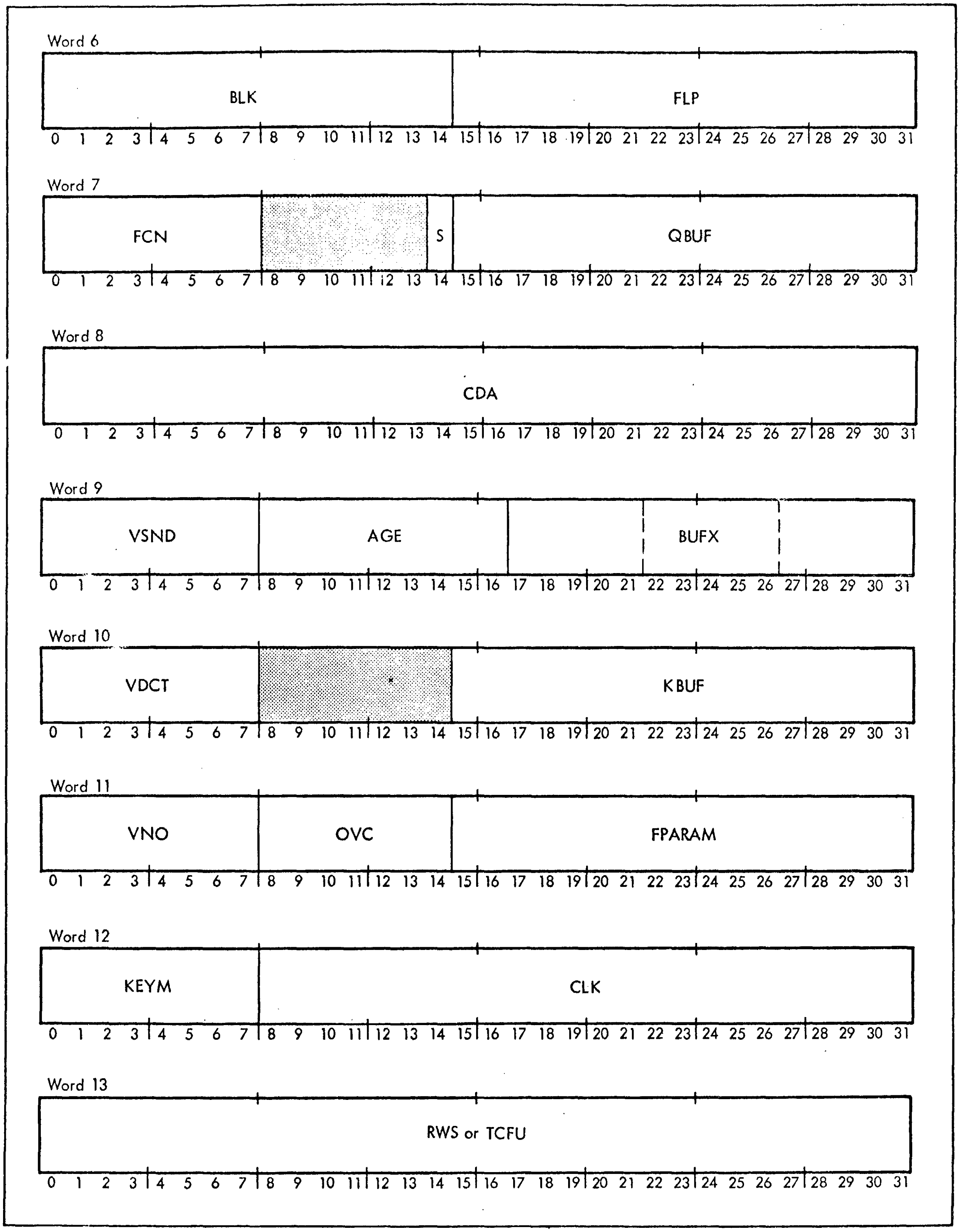
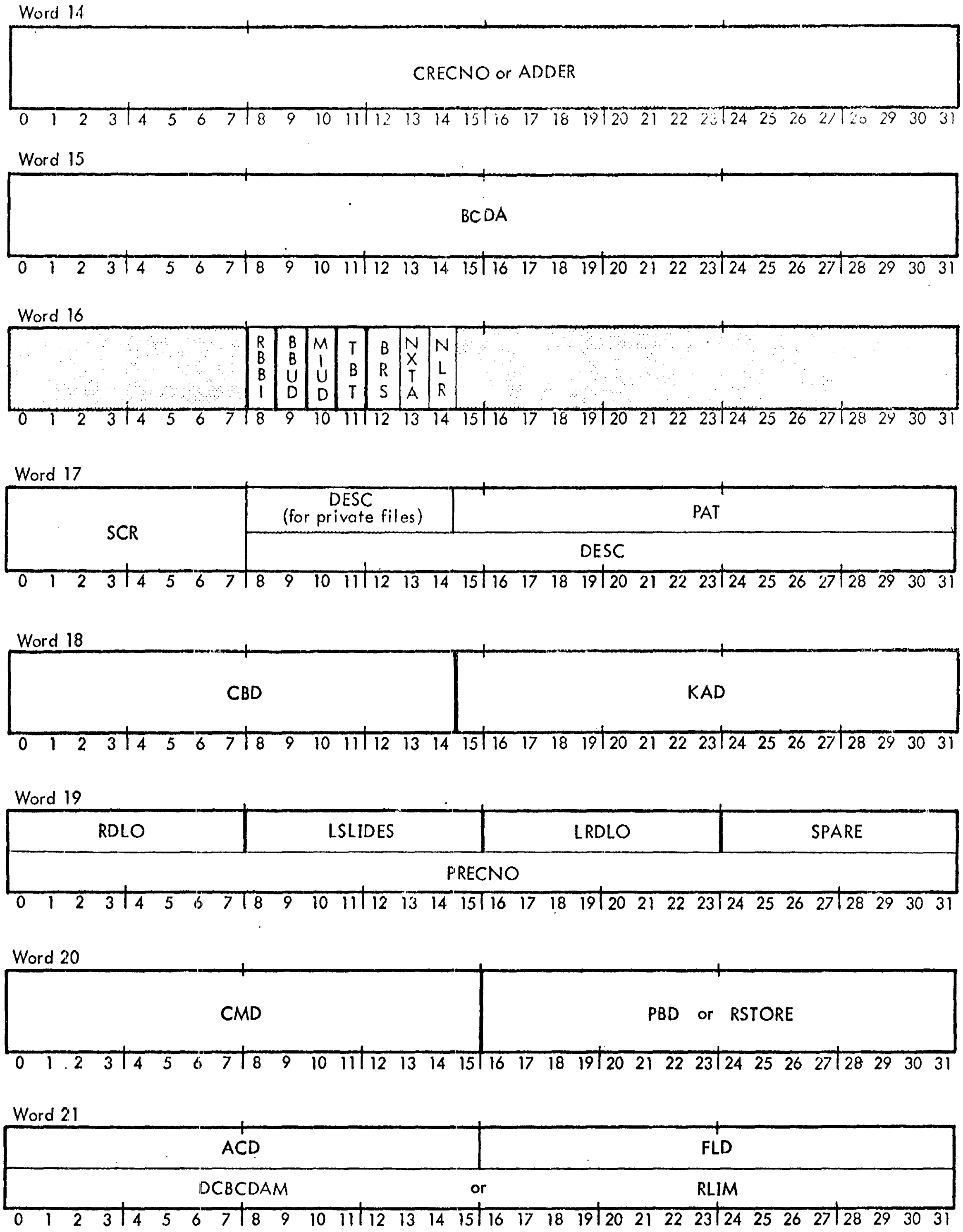


Figure A-1. Format of File DCB (cont.)



Words 22 → n are used for variable length parameters.

Figure A-1. Format of File DCB (cont.)

Field	Description	Word
ABA	Contains the address of the user's routine that will handle abnormal conditions resulting from insufficient or conflicting information. (The monitor returns to ABA in the FPT if the abnormal condition is the result of a device abnormality.)	4
ACD	contains the word displacement to the user's account number in the DCB relative to the start of the variable length parameters. (FLP+ACD = FWA of the EBCDIC account number.)	21
ACS	is the file access indicator (0 = none specified and is treated as sequential, 1 = sequential, 2 = direct). ACS is only meaningful when a file is first written in the OUT or OUTIN mode. If a file has consecutive organization, OPEN always sets ACS to sequential (regardless of the access specified). If a file has keyed organization and access is not specified, OPEN leaves ACS unchanged and the residual ACS value applies. If a file has keyed organization and sequential access is specified, the keys written must be in ascending order. However, if the organization is keyed and direct access is specified, the keys can be written in any order (the monitor sorts them into ascending order). ACS is not used by random files.	5
ADDER	contains the size of a single entry in the master index structure or directory for operations on keyed files or directories.	14
AGE	is used to measure the most recent activity on the DCB so that buffer truncation can be made more efficiently.	9
ARS	contains	4
	1. the actual number of data bytes transferred to or from the user following a read or write.	
	2. the number of records remaining to be skipped following a PRECORD operation that has terminated due to an end-of-file or a beginning-of-file condition.	
ASN	indicates the assignment type currently in effect for the DCB (0 = null, 1 = file, 2 = Xerox labeled tape, 3 = device, X'A' = ANS labeled tape).	0
BBUD	indicates whether or not the blocking buffer (BUF1) has been changed since it was last read or initialized (0 = unchanged, 1 = changed). This flag is used to determine whether or not BUF1 needs to be written out to the data granule specified in BCDA before truncating the buffer. BBUD is not used by random files.	16
BCDA	contains the disk address of the data granule currently in the blocking buffer (BUF1). BCDA is not used by random files.	15
BLK	contains	6
	1. the byte count of the record segment pointed to by either CBD or PBD, depending upon the point in time. Not applicable to random files.	6
	2. the number of bytes to be transferred by the I/O routines whenever called.	
BRS	indicates whether or not the record segment pointed to be CBD or PBD, depending upon the point in time, is blocked (0 = unblocked, 1 = blocked). BRS is not used by random files. During an open BRS, indicates whether the 'TEST' option was indicated in the open FPT (0 = not test, 1 = test).	16

Field	Description	Word
BUF	contains the address of the user's buffer where the data record is to be read or written.	2
BUFX	contains three 5-bit subfields used to index into the table of pooled buffers available to the file management system. These indexes have varying significance depending on the current operation being performed.	9
CBD	contains the current byte displacement within the blocking buffer (BUF1). CBD specifies where the record segment associated with the key pointed to by CMD begins. When writing on the file, CBD = 0 if a data granule other than the last is being updated. CBD is not used by random files.	18
CDA	contains 1. the disk address to be used by the I/O routines whenever called. 2. a counter indicating the number of records to skip. Not applicable to random files.	
CFUA	contains the address of the CFU associated with the file. During open or close operations, CFUA contains the address of the ACNCFU and FILCFU.	1
CLK	contains 1. the net number of data and Master Index granules allocated to or released from the file during this OPEN. Applicable to keyed and consecutive files. The field is a 23-bit signed integer with a guard bit in bit 8 that is used to prevent overflow into the KEYM field. 2. the number of granules allocated to the file. Applicable to random files.	12
CMD	contains 1. the byte displacement to the current key entry in the Master Index Buffer (BUF2) for keyed files. CMD, along with TRN and DCBCDAM, points to the current position in the file. For consecutive files, CMD contains a word position in the granule pointed to by DCBCDAM. None of this is applicable to random files. 2. the byte displacement to the current entry in the Account Directory or File Directory index buffer (BUF2) when the file is being opened or closed.	20
CRECNO	contains the current record number. It is set to 1. 0 if at the beginning of the file. 2. the number of records in the file (obtained from TDA in the CFU) if at the end of the file. 3. the sequential record number of the record most recently read or written. CRECNO is only used for consecutive files.	14
CYL	specifies whether the file assigned to the DCB is to be allocated by granules or cylinders (0 = granule allocation, 1 = cylinder allocation). Only meaningful for public files.	0

Field	Description	Word
DCBCDAM	is used when CFUA points to a user CFU for keyed or random files and contains the disk address of the current index half-granule in the Master Index Buffer (BUF2). If CFUA points to the Account or File Directory CFU, CDAM in FILCFU or ACNCFU contains the disk address of the current index half-granule in BUF2. For consecutive files, DCBCDAM contains a disk address of a granule, reflecting (in conjunction with CMD) the location in the file at which the most recent data transfer operation took place.	21
DESC	is used as temporary storage for file descriptors during open and close. For private files, DESC resides in bits 8-14.	17
DIR	indicates the direction of the read operation (0 = forward, 1 = reverse). DIR is not used by random files.	0
EGV	is the event-given flag and indicates whether or not the completion code posted in the TYC field has been communicated to the user's program by the CHECK routine (0 = no, 1 = yes). The CHECK routine is called either directly by the user or indirectly by the monitor, depending upon the WAIT, ERR, and ABN options in the FPT.	0
EOP	is the ending operation indicator (0 = other, e.g., rewind, 1 = read, 2 = write). Specifies the type of I/O operation currently or last performed. EOP is not used by random files.	0
ERA	contains the address of the user's routine that will handle error conditions resulting from insufficient or conflicting information. (The monitor returns to the ERA in the FPT if the error condition is the result of a device failure.)	3
EXT	is the file extension flag and indicates whether OPEN is to position to the beginning or end of a specified file (0 = beginning-of-file, 1 = end-of-file).	0
EXTRND	is set to one if the NLR field is to be logically appended to the RSTORE field (NLR being the most significant field) for a random file. Otherwise, it is set to zero.	5
FCD	indicates whether the DCB is opened or closed (0 = closed, 1 = opened).	0
FCI	indicates whether the DCB has ever been closed. This flag is set when the DCB is first closed and then never reset (0 = DCB has never been closed, 1 = DCB has been previously opened and closed).	0
FCN	indicates the current number of I/O operations that have been initiated but not completed, for this DCB.	7
FIL1	indicates the file option last specified (0 = none specified and is treated as release, 1 = release, 2 = save).	5
FLD	contains the word displacement to the file name in the DCB relative to the start of the variable length parameters (FLD + FLP = FWA of the EBCDIC file name).	21
FLP	contains the address of the start of the variable length parameters in the DCB (called the file list-pointer).	6
FPARAM	contains the receiving address of the user's 90-word buffer to which the variable length parameters from the file's FIT are to be passed.	11
FUN	indicates the file mode function (0 = null, 1 = IN, 2= OUT, 4 = INOUT, 8 = OUTIN). Since the monitor does not distinguish between INOUT and OUTIN on random files, OUTIN is set the same as INOUT for random files.	1
HBTD	is the I/O handler's byte displacement indicator and is used whenever the I/O routines are called to specify the byte displacement within QBUF into which the data transfer is to begin.	0

Field	Description	Word
IMT	is the image-type flag and indicates the type of key entry in BUF2 (0 = Account or File Directory key, 4 = user's file key).	16
KAD	contains 1. the address of the key specified by the user in the read or write FPT. If a consecutive file is being written, KAD points to the dummy key (in the VLP). If a consecutive file is being read, KAD contains 0. Applicable to keyed and consecutive files. 2. the address of the account number or file name when opening or closing the file.	18
KBUF	contains 1. the address of the buffer containing the key most recently accessed in the Master Index or File Directory. The field is set up by the M:DCB procedure and points to an 8-word buffer following the VLPs. Not applicable to random files except during open. 2. the address of the word buffer containing the relative granule number of the first sector to be used in the I/O transfer. Applicable to random files only. 3. the address of an 8-word buffer in the DCB that contains the TEXTC key or records read sequentially from a keyed file.	10
KEYM	contains 1. the maximum length, in bytes, of the keys in the file pointed to by the DCB. Applicable to keyed files. Maximum value is 31. 2. the type of device that the random file is to be allocated on (0 = allocate on RAD or DP, X'7' = allocate on RAD, X'B' = allocate on DP). Applicable to random files.	12
LRDLO	contains the limiting number of contiguous index half-granules that can be allocated in level 0 and not be reflected in level 1 before the flag, which signals CLOSE to reconstruct the higher level index structure, is set (i.e., before SLIDES in the CFU is set equal to 255). LRDLO is only used for keyed files.	19
LSLIDES	only has meaning if a multilevel index exists and contains 1. the limiting number of index half granules that can be allocated in level 0 and not be reflected in level 1 before the flag, which signals CLOSE to reconstruct the higher level index structure, is set. 2. the value 255, which means that once a higher level index structure exists, it is not to be reconstructed. LSLIDES is only used for keyed files.	19
MIUD	indicates whether or not the Master Index Buffer (BUF2) has been changed since it was last read or initialized (0 = unchanged, 1 = changed). This flag is used to determine whether or not BUF2 needs to be written out to the sector specified in either DCBCDAM or CDAM in FILCFU or ACNCFU before truncating the buffer.	16
NACUP	indicates whether the file's descriptors indicate that the last access date is not to be updated (0 = may be updated, 1 = may not be updated).	5
NLR	indicates whether or not the record segment pointed to by CBD is the first record in a continued data record (0 = second or nth record segment, 1 = first or only record segment). NLR is only meaningful during a WRITE operation.	16

Field	Description	Word
NOSEP	specifies whether or not index granules are to be allocated like data granules (0 = no, 1 = yes). Normally, index granules are allocated on DP. However, if all the devices of the normally allocated type are saturated, the system attempts to allocate on an alternate device. The order of allocation for data granules is DP and RAD regardless of the NOSEP flag. The order of allocation for index granules also is DP and RAD if the NOSEP flag is reset. If the NOSEP flag is set, index granules are allocated like data granules. This flag has no meaning for private files. NOSEP is not used by random files.	0
NRA	indicates the number of recovery tries that may be attempted before a device error message is to be logged.	2
NRT	indicates the number of recovery tries remaining before a device error message is to be logged.	1
NWK	indicates whether or not NEWKEY was specified in the M:WRITE FPT (0 = replace an existing key, if the key does not exist, take an abnormal return; 1 = write a new key, if the key already exists, take an abnormal return). If ONWK is set, the NWK flag is ignored. NWK is only used for keyed files.	5
NXTA	is the next account indicator and specifies whether this account (i.e., the account number in the DCB/JIT) or the next account in the Account Directory (i.e., the one following the account named in the DCB) is to be assigned to the DCB at OPEN (0 = this account, 1 = the next account). If an account number is not specified in the DCB and the NXTA indicator is set, the first account in the Account Directory is put in the DCB and nothing more is done unless NXTF is also set. The previous is not applicable for private volumes. After a file is open, the bit is set to 1 if the DCB is open to a star file (see Glossary); otherwise, it is set to 0.	16
NXTF	is the next file indicator and specifies whether this file (i.e., the file named in the DCB/FPT) or the next file in the File Directory (i.e., the one following the file named in the DCB) is to be assigned to the DCB at OPEN. If a file name is not specified (in either the DCB or FPT), the first name in the File Directory is put in the DCB and assigned (0 = this file, 1 = next file).	5
ONWK	indicates whether or not ONEKEY was specified in the M:WRITE FPT (0 = check NWK flag, 1 = if the key already exists, replace the corresponding record, otherwise write a new record). ONWK is only used for keyed files.	5
ORG	is the file organization indicator (0 = none specified and is treated as consecutive, 1 = consecutive, 2 = keyed, 3 = random).	5
OVC	is the open volume count and only has meaning for private files. 1. for consecutive private files, OVC indicates whether or not the volume pointed to by VNO is opened or not (0 = no, 1 = yes). 2. for keyed or random private files, OVC contains a count of the numbers of volumes that have been opened.	11
PAT	contains the allocation table address of the private volume pointed to by VNO. Only has meaning for private files.	17

Field	Description	Word
PBD	is the previous buffer displacement indicator, specifying at which byte in the blocking buffer (BUF1) the previous record segment begins. PBD is not used by random files.	20
PRECNO	contains the direction (+ or −) and the number of records that must be skipped from the position indicated in CRECNO prior to a data transfer operation (read, write, or delete). PRECNO is only used for consecutive fi.	19
PRIV	indicates whether the file assigned to the DCB is public or private (0 = public, 1 = private). Public files reside on public devices and private files reside on private volume sets.	0
QBUF	contains 1. the buffer address to be used by the I/O routines whenever called. 2. the address within the user's buffer where the next record segment begins. QBUF, 2 is not applicable to random files.	7
RAX	controls read ahead. If set to X'FF', no read ahead is possible. If set to zero, no read ahead is in progress. Otherwise, RAX contains an index into read ahead tables.	5
RBBI	is the release blocking buffer inhibit flag and indicates whether or not the blocking buffer (BUF1) should be released during end-action after the data granule has been read into (BUF1) and the record segment has been transferred to the user's buffer. (0 = release BUF1, 1 = do not release BUF1.) RBBI is not used by random files.	16
RDLO	contains a tally (up to 255) of the number of index half-granules that are read or inserted at level 0 to locate the position of a user-specified key entry at level 0. If RDLO is greater than LRDLO, the flag, which signals CLOSE to reconstruct the higher level index structure, is set. RDLO is only used for keyed files.	19
RLIM	temporarily contains the number of granules specified in the RSTORE option on the ASSIGN control command during the ASSIGN/DCB merge. The first halfword contains X'6E4C' which is used as a flag to indicate that RSTORE was specified. RLIM is used by random files only.	21
RNDEV	contains the type of device requested for file allocation (0 = none specified and for private files gets changed to X'B', 7 = RAD, and X'B' = DP).	5
RSTORE	contains the number of granules to be allocated to the file. RSTORE is used by random files only. If RSTORE value is zero when a random file is created, an abnormal return is made with a code of X'14'. Bits 8-15 of word 5 are used by random files as a high order extension of this field if the EXTRAND bit is set.	20
RSZ	indicates the default record size, in bytes.	3
RWS	indicates 1. the requested number of bytes to be read or written from the user's buffer (BUF). During the I/O operation, RWS is decremented by the value in BLK each time that a record segment is either output or blocked. At the termination of the I/O operation, RWS is set equal to ARS. Applicable to keyed and consecutive files. 2. the requested number of bytes to be read or written from the user's buffer (BUF). At the termination of the I/O operation, RWS is set equal to ARS. Applicable to random files.	13

Field	Description	Word
S	contains the value of the S field from the mode specification in the Open Cal FPT. S = 1 means SHARE; S = 0 means EXCLUSIVE.	7
SCR	indicates 1. the byte length of the key portion of the entries in the Master Index currently referenced by the DCB. This can be the Master Index for the Account Directory, the File Directory, or the user's file. 2. this field is used to temporarily contain the contents of KEYM field. Applicable to random files only.	17
SPARE	contains the number of spare byte positions to be left unused in the end of the current index half-granule in the event that the key to be added is the last key in the file. SPARE is only used for keyed files.	19
SWXV	is the switch volume flag and indicates whether or not the current volume is to be switched to the next volume after all updated buffers have been output to the current volume (0 = no, 1 = yes). Only used for consecutive private files.	0
TBT	not meaningfully used for files; however, the flag does get set and reset.	16
TCFU	contains the address of the user CFU during CLOSE.	13
TRN	indicates, for keyed files, whether the file is positioned before or after the data record whose key entry is pointed to by CMD (0 = after, 1 = before). For consecutive files, this bit is set only if the most recently executed operation on the file was a read backwards.	5
TYC	indicates the type of completion of an I/O operation.	2

TYC Code	Corresponding Error/ Abnormal Code	Meaning
0	0	normal without device I/O transfer
1	0	normal with a device I/O transfer
2	7	lost data
3	10	beginning-of-tape
4	4	beginning-of-file
5	1C	end-of-reel
6	5	end-of-data
7	6	end-of-file
8	41	read error
9	45	write error
A	57	public devices/private volume-set saturated
B	0	SLIDES is 255
C	0	partial higher level index built

UBTD	is the byte displacement indicator, specifying at which byte in the user's buffer (BUF) the data record begins.	0

Field	Description	Word
USR	indicates whether the JOB account number is the same as the account number specified in the DCB (0 = yes, 1 = no).	0
VDCT	contains the DCT index of the device on which the volume (in a private volume set) pointed to by VNO is mounted. Only meaningful for private files.	10
VNO	contains the volume number of the private volume currently being referenced via the DCB. Volume number is the position (starting with one) of a volume within the DCB's SN list. The SN list in the DCB has a fixed order and comes from the serial number table on the primary volume of a private volume set. Only meaningful for private files.	11
VSND	contains the word displacement to the serial number table of the private volume set (i.e., the SN list) in the DCB relative to the start of the Variable Length Parameters (FLP + VSND = the control word of the SN list).	9
WAT	is the wait flag and indicates whether or not WAIT was specified in the FPT (0 = no, 1 = yes).	0
XUP	indicates whether or not a higher level index structure is in the process of being reconstructed or constructed (0 = either that there is no higher level index or that the higher level index is complete, 1 = that the higher level index is being built). Only meaningful for keyed files.	0

VARIABLE LENGTH PARAMETERS 22 → n

Each variable length parameter entry is preceded by a control word of the following form:

Byte 0 = a code number (see Table A-1) identifying the parameter which follows.

Byte 1 = code for the entry position (00 = more parameter entries to follow, 01 = last parameter entry).

Byte 2 = number of significant data words in the parameter entry.

Byte 3 = total number of words reserved for the entry, not including the control word (that is, maximum entry length).

Table A-1. Variable Length Parameter Codes

Code	Parameter Type
01	File name (the first byte of which contains the number of characters in the name).
02	Account number.
03	Password.
04	Expiration date.
05	READ account numbers.
06	WRITE account numbers.
07	SN/INSN serial numbers.
08	OUTSN serial numbers.
09	File information (see Figure A-2).
0A	Modification date.
0B	SYNON name.
0C	File information (see Figure A-2).
0D	File size.
0E	Creation date.
0F	Last access date.
10	Backup date.
11	Descriptors.
12	Search open mask.
13	Reserved for later use.
14	Execute account numbers.
15	Execute vehicle
16	Account list
17	Permission bits corresponding to 16 above.
18	For use by installation.
19	For inclusion of device open prime PLIST.

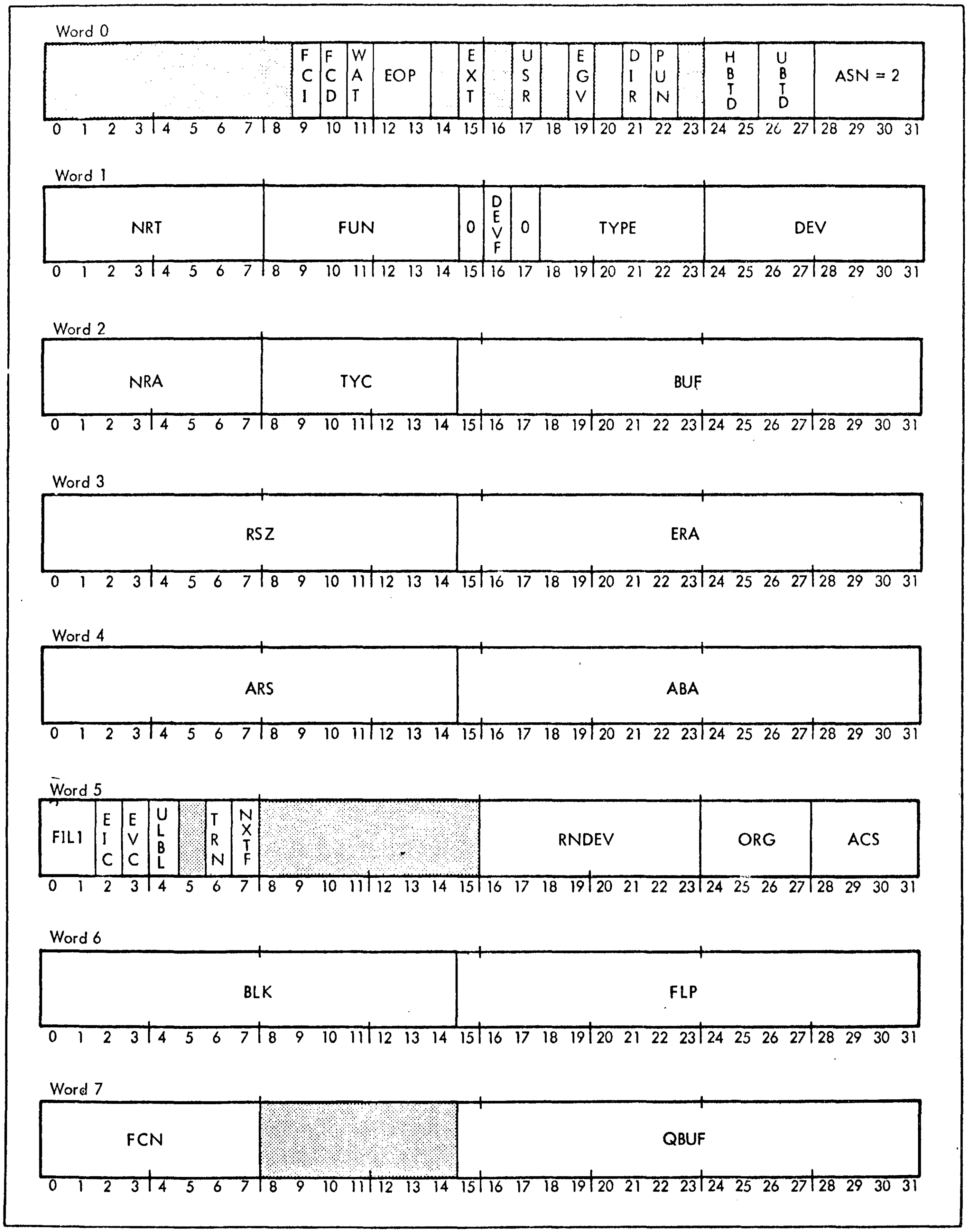


Figure A-4. Format of Xerox Labeled Tape DCB

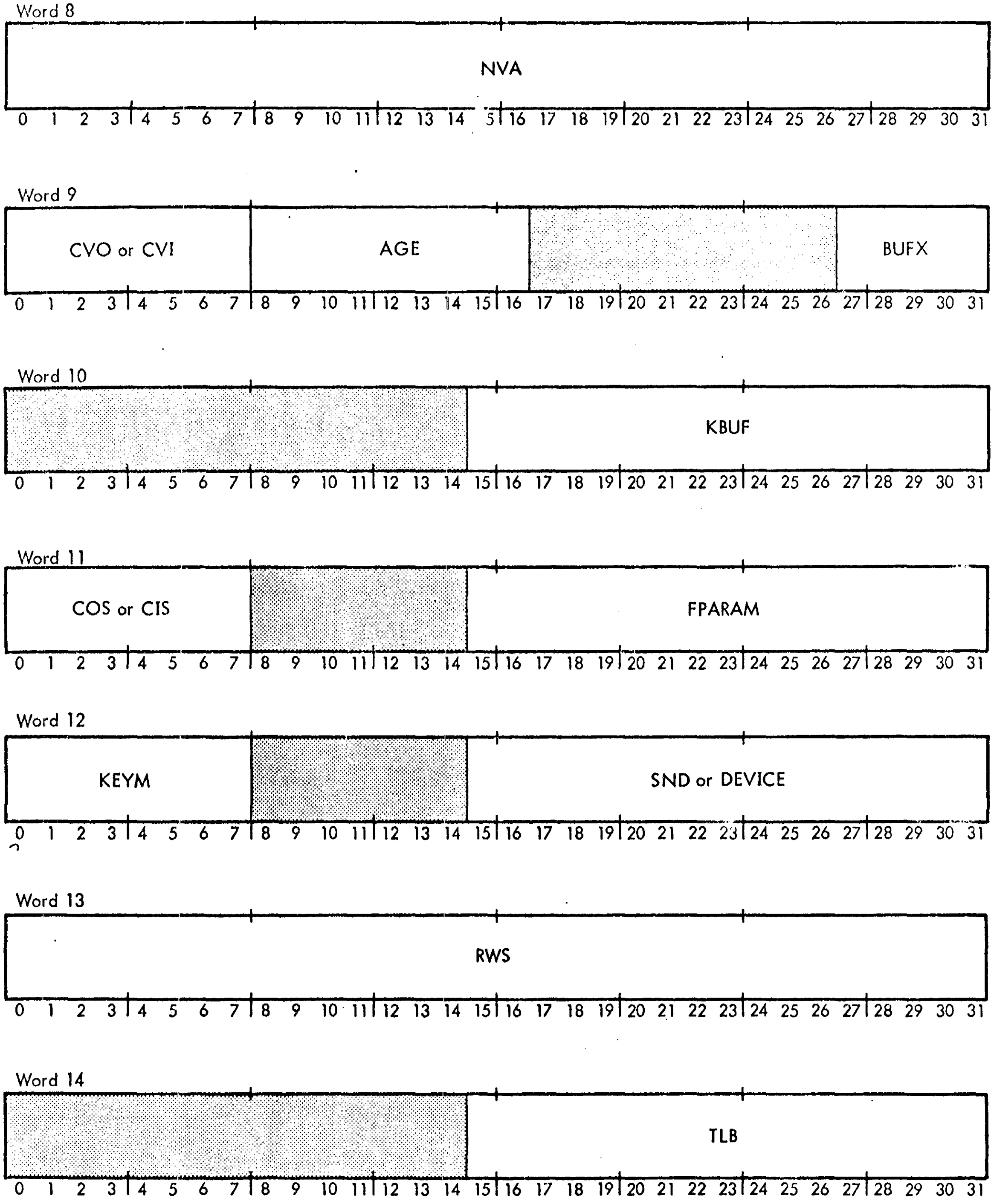
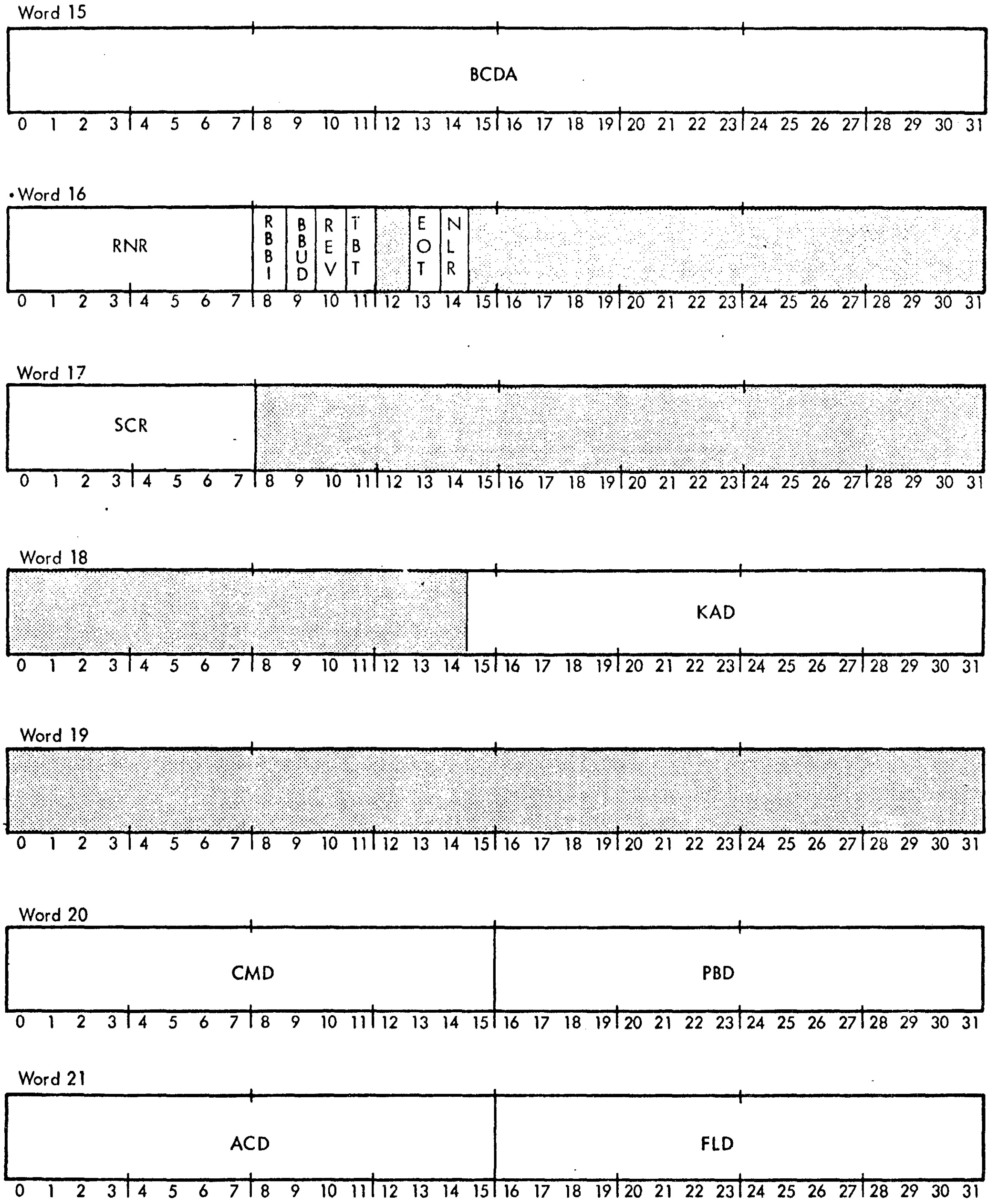


Figure A-4. Format of Xerox Labeled Tape DCB (cont.)



Words 22 → n are used for variable length parameters.

Figure A-4. Format of Xerox Labeled Tape DCB (cont.)

Field	Description	Word
ABA	contains the address of the user's routine that will handle abnormal conditions resulting from insufficient or conflicting information. (The monitor returns to ABA in the FPT if the abnormal condition is the result of a device abnormality.)	4
ACD	contains the word displacement to the users account number in the DCB relative to the start of the variable length parameters. (FLP + ACD = FWA of the EBCDIC account number.)	21
ACS	is the file access indicator (0 = none specified and is treated as sequential, 1 = sequential, 2 = direct). If a file has keyed organization, the keys written must be in ascending order regardless of the access specified.	5
AGE	is used to measure the most recent activity on the DCB so that buffer truncation can be made more efficiently.	9
ARS	contains 1. the actual number of data bytes transferred to or from the user following a read or write. 2. the number of records remaining to be skipped following a PRECORD operation that has terminated due to an end-of-file or a beginning-of-file condition.	4
ASN	indicates the assignment type currently in effect for the DCB (0 = null, 1 = file, 2 = Xerox labeled tape, 3 = device, X'A' = ANS labeled tape).	0
BBUD	indicates whether or not the blocking buffer (BUF1) has been changed since it was last read or initialized (0 = unchanged, 1 = changed). The monitor uses this flag to determine whether or not BUF1 needs to be written out to the data granule specified in BCDA before truncating the buffer.	16
BCDA	contains the number of either the current or last accessed entry in the blocking buffer (BUF1), depending upon the point in time. An entry in a Labeled Tape block consists of a key, control information, and the associated record segment. Entries are numbered from 1 to n.	15
BLK	contains 1. the byte count of the record segment pointed to by either CBD or PBD, depending upon the point in time. 2. the number of bytes to be transferred by the I/O routines whenever called.	6
BUF	contains the address of the user's buffer where the data record is to be read or written, or where user trailer labels are to be read.	2
BUFX	contains the index of the blocking buffer.	9
CIS	contains the relative position of the serial number (in the SN list) of the magnetic tape reel used for current file input.	11
CMD	contains the byte displacement to the current entry in the blocking buffer (BUF1). An entry in a Labeled Tape block consists of a key, control information, and the associated record segment.	20
COS	contains the relative position of the serial number (in the SN list) of the magnetic tape reel used for current file output.	11

Field	Description	Word
CVI	indicates the relative volume number of the current input tape within the current file. CVI is taken from the beginning-of-file sentinel, which appears at the beginning of file and at the beginning of each reel, if the file is continued on more than one reel.	9
CVO	indicates the relative volume number of the current output tape with respect to the current file. CVO is recorded in the beginning-of-file sentinel which is written at the beginning of the file and at the beginning of each reel, if the file is continued on more than one reel.	9
DEV	contains the DCT index of the device assigned to the DCT. DEV is only meaningful if DEVF = 1. When DEVF = 0, the field is defined as OPLB.	1
DEVF	indicates whether the DCB is assigned to a device or an operational label. (0 = operational label, 1 = device.)	1
DEVICE	contains the EBCDIC name specified on the DEVICE option in the M:OPEN call. This use is only transient, and the field is later overlaid by SND.	12
DIR	indicates the direction of the read operations (0 = forward, 1 = reverse).	0
EGV	is the event-given flag and indicates whether or not the completion code posted in the TYC field has been communicated to the user's program by the CHECK routine (0 = no, 1 = yes). The CHECK routine is called either directly by the user or indirectly by the monitor, depending upon the WAIT, ERR, and ABN options in the FPT.	0
EIC	indicates whether or not the last block read from a consecutive file was in error and that a validity check on the control information revealed inconsistencies (0 = no, 1 = yes).	5
EOP	is the ending operation indicator (0 = other, e.g., rewind, 1 = read, 2 = write). Specifies the type of I/O operation currently or last performed.	0
EOT	indicates whether or not the physical end-of-tape mark has been encountered (0 = no, 1 = yes).	16
ERA	contains the address of the user's routine that will handle error conditions resulting from insufficient or conflicting information. (The monitor returns to the ERA in the FPT if the error condition is the result of the device failure.)	3
EVC	indicates whether or not the last block read from a consecutive file was in error but a validity check on control information revealed no inconsistencies (0 = no, 1 = yes).	5
EXT	is the file extension flag and indicates whether OPEN is to position a tape at the beginning or end of a specified file (0 = beginning-of-file, 1 = end-of-file).	0
FCD	indicates whether the DCB is opened or closed (0 = closed, 1 = opened).	0
FCI	indicates whether the DCB has ever been closed. This flag is set when the DCB is first closed and then never reset (0 = DCB has never been closed, 1 = DCB has been previously open and closed).	0
FCN	indicates the current number of I/O operations that have been initiated but not completed, for this DCB.	7
FIL1	indicates the file option specified when the DCB was last opened (0 = none specified, 1 = release, 2 = save).	5
FLD	contains the word displacement to the file name in the DCB relative to the start of the variable length parameters (FLD + FLP = FWA of the EBCDIC file name).	21

Field	Description	Word
FLP	contains the address of the variable length parameters in the DCB (called the file list-pointer).	6
FPARAM	contains the receiving address of the user's 90-word buffer to which the variable length parameters from the file's FIT are to be passed.	11
FUN	indicates the file mode function (0 = null, 1 = IN, 2 = OUT, 4 = INOUT, 8 = OUTIN).	1
HBTD	is the I/O handler's byte indicator and is used whenever the I/O routines are called to specify the byte displacement within QBUF into which the data transfer is to begin.	0
KAD	contains the address of the key specified by the user in the read or write FPT. If a consecutive file is being written, KAD points to the dummy key. If a consecutive file is being read, KAD contains 0.	18
KBUF	contains the address of the buffer containing the key associated with the data record last accessed in the blocking buffer.	10
KEYM	contains the maximum length, in bytes, of the keys in the file pointed to by the DCB. Only meaningful for keyed files. Maximum value is 31.	12
NLR	indicates whether or not the record segment pointed to by CMD is the first record in a continued data record (0 = second or nth record segment, 1 = first or only record segment). NLR is only meaningful during a write and is reset to zero when the first record segment is output.	16
NRA	indicates the number of recovery tries that may be attempted before a device error message is to be logged.	2
NRT	indicates the number of recovery tries remaining before a device error message is to be logged.	1
NXTF	is the next file indicator and specifies whether this file (i.e., the file named in the DCB/FPT) or the next file in the File Directory (i.e., the one following the file named in the DCB) is to be assigned to the DCB at OPEN. If a file name is not specified (in either the DCB or FPT), the first name in the File Directory is put in the DCB and assigned (0 = this file, 1 = next file).	5
NVA	contains a counter indicating the number of records to skip. It is also used as an indicator. If NVA is negative, the last operation performed was a rewind.	8
ORG	is the file organization indicator (0 = none specified, and is treated as consecutive, 1 = consecutive, 2 = keyed).	5

Field	Description	Word
PBD	contains 1. a counter used by M:OPEN to determine how many volumes remain to be searched for the specified file. 2. the number of bytes in the previous labeled tape block. PBD is only meaningful on a read operation and is taken from the PBS field of a labeled block.	20
PUN	indicates whether a 7-track tape is to be read/written in the packed or unpacked mode (0 = unpacked, 1 = packed).	0
QBUF	contains 1. the buffer address to be used by the I/O routines whenever called. 2. the address within the user's buffer where the next record segment begins.	7
RBBI	indicates whether or not the blocking buffer should be released at end-action (0 = release blocking buffer, 1 = do not release blocking buffer because the buffer will be reused to read in the next block). RBBI is set during a read operation when a data record is continued and more than one read request will be initiated.	16
REV	indicates whether the Labeled Tape block currently in the blocking buffer (BUF1) was read in the forward or reverse direction (0 = forward, 1 = reverse).	16
RNDEV	contains the type of device specified (0 = none specified, 8 = 9T, 9 = 7T, X'A' = MT).	5
RNR	is a transient flag used by the system to defer error reporting for a tape block read by the monitor in anticipation of a read not yet requested by the user (0 = user requested read, 1 = user read not requested).	16
RSZ	indicates the default record size, in bytes.	3
RWS	indicates the requested number of bytes to be read or written from the user's buffer (BUF). At the termination of the I/O operation, RWS is set equal to ARS.	13
SCR	indicates the byte length of the key portion of the entries in the Labeled Tape block.	17
SND	contains the word displacement to the tape serial number (SN list) in the DCB relative to the start of the variable length parameters (FLP + SND = FWA of the EBCDIC serial numbers).	12
TBT	indicates whether or not the Labeled Tape blocking buffer has been truncated (0 = no, 1 = yes). Truncation means that monitor has taken the blocking buffer and, if necessary, written the block on tape.	16
TLB	contains the address of a user's label that is to be written on a tape file when the file is output.	14
TRN	indicates whether the file is positioned before or after the data record whose key entry is pointed to by CMD (0 = after, 1 = before).	5

Field	Description	Word
TYC	indicates the type of completion of an I/O operation.	2

TYC Code	Corresponding Error/ Abnormal Code	Meaning
0	0	normal without device I/O transfer
1	0	normal with a device I/O transfer
2	7	lost data
3	1D	beginning-of-tape
4	4	beginning-of-file
5	1C	end-of-reel
6	5	end-of-data
7	6	end-of-file
8	41	read error
9	45	write error

Field	Description	Word
TYPE	contains the device-type code for the tape assigned to this DCB.	1
UBTD	is the byte displacement indicator, specifying at which byte in the user's buffer (BUF) the data record begins.	0
ULBL	indicates whether or not the ULBL option was specified in the FPT of M:READ (0 = no, 1 = yes).	5
USR	indicates whether or not the job account number is the same as the account number specified in the DCB (0 = yes, 1 = no).	0
WAT	is the wait flag and indicates whether or not WAIT was specified in the FPT (0 = no, 1 = yes).	0

VARIABLE LENGTH PARAMETERS 22→n

Each variable length parameter entry is preceded by a control word of the form shown for File DCB and in Table A-1.

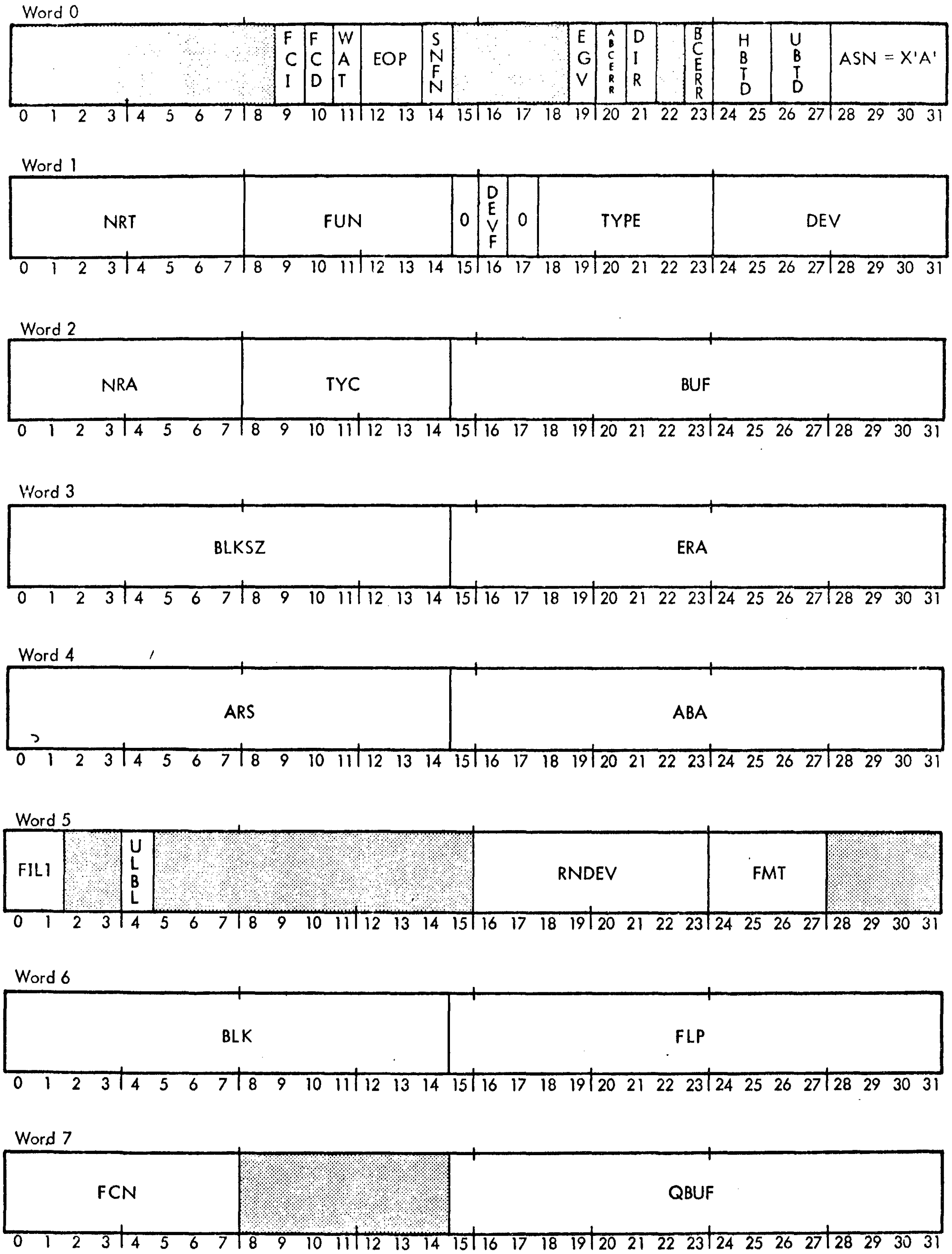


Figure A-5. Format of ANS Labeled Tape DCB

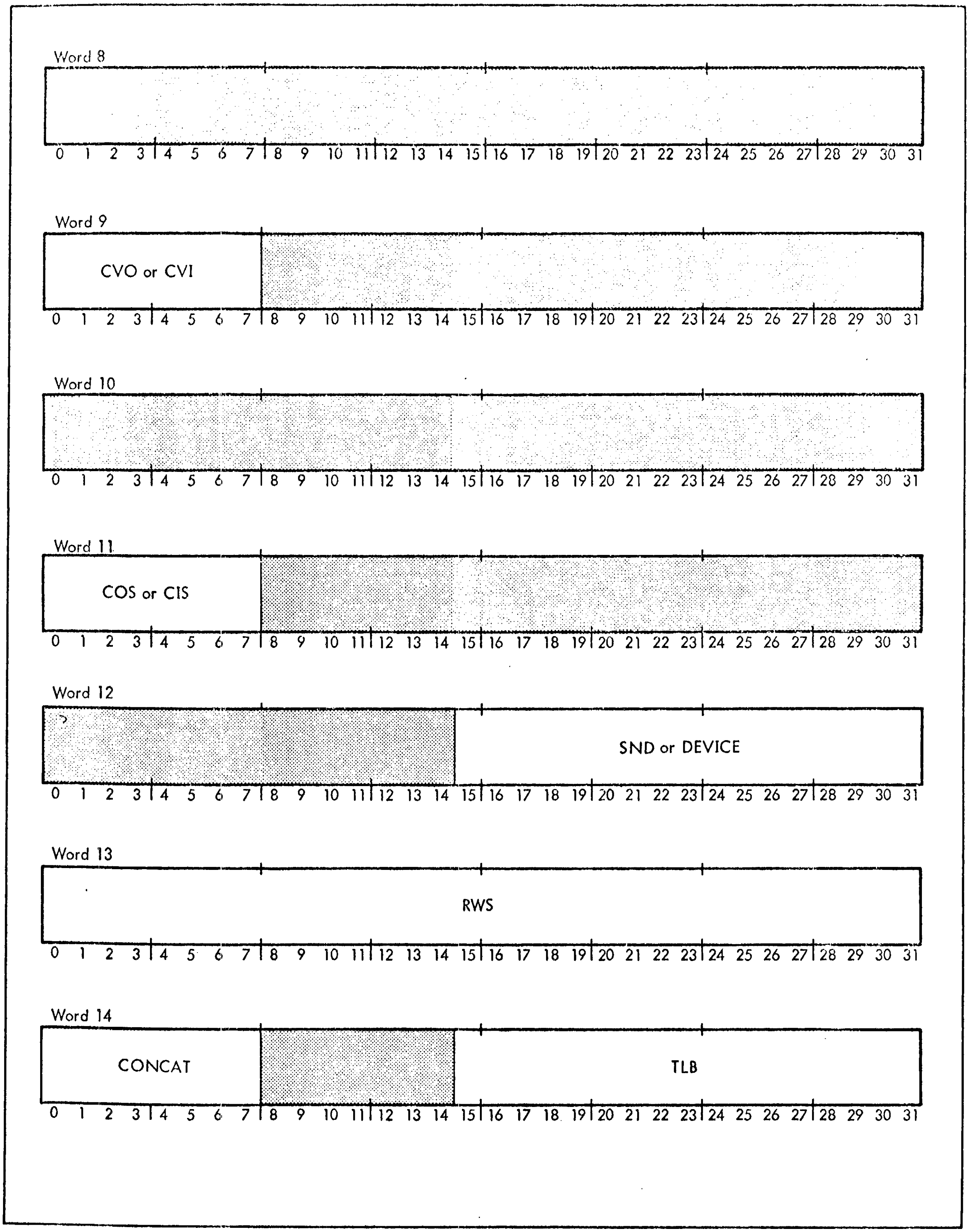
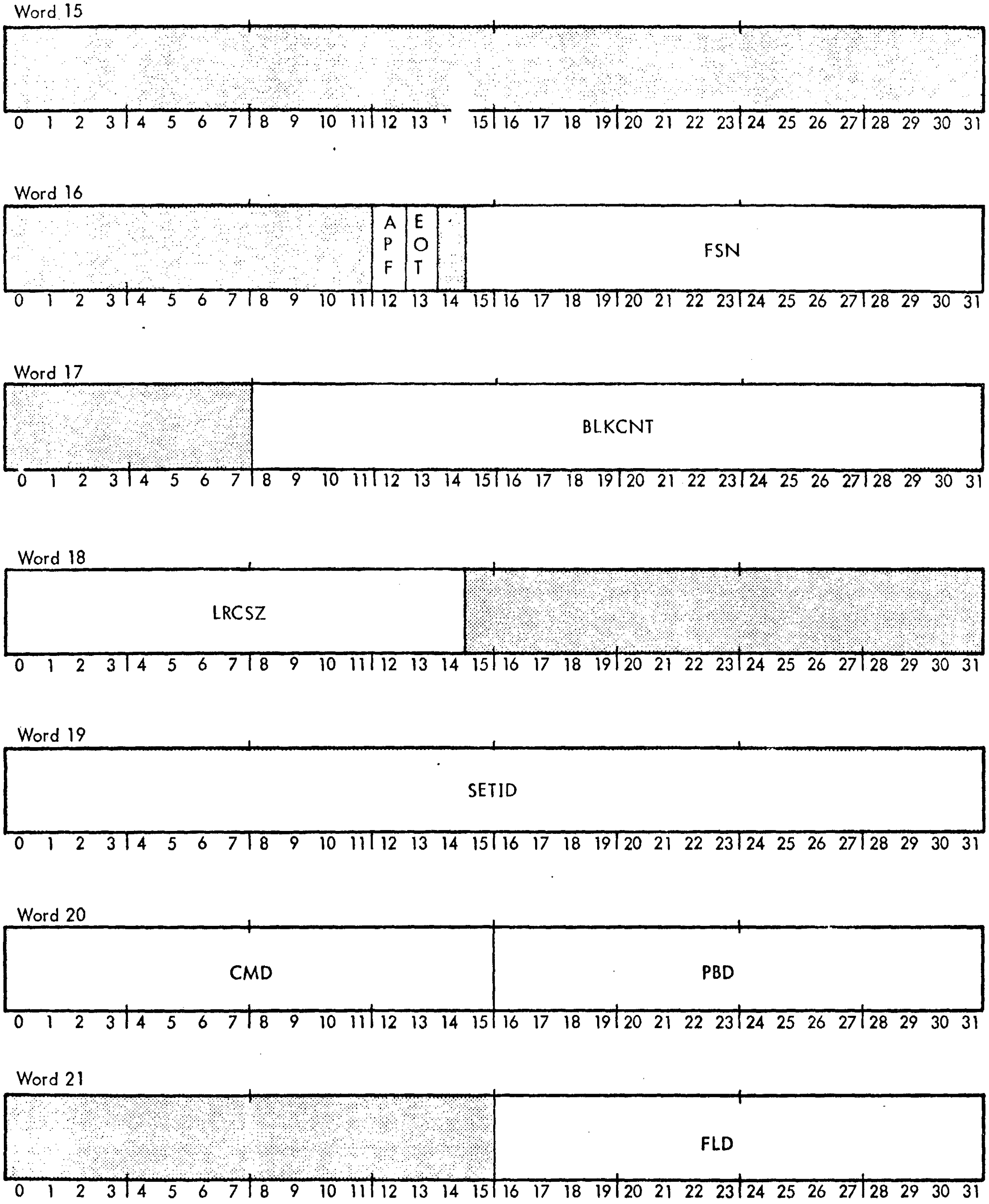


Figure A-5. Format of ANS Labeled Tape DCB (cont.)



Words 22 → n are used for variable length parameters.

Figure A-5. Format of ANS Labeled Tape DCB (cont.)

Field	Description	Word
ABA	contains the address of the user's routine that will handle abnormal conditions resulting from insufficient or conflicing information. (The monitor returns to ABA in the FPT if the abnormal condition is the result of a device abnormality.)	4
ABCERR	indicates whether or not block count errors are to be accepted; i.e., whether or not processing is to continue in the case of inconsistency between the tape-specified and system-accumulated block counts (0 = no, 1 = yes).	0
APF	contains the ANS post flag. If set to 1, it indicates that ANS post-processing of an I/O operation has not yet been done.	16
ARS	contains 1. the actual number of data bytes transferred to or from the user following a read or write. 2. the number of records remaining to be skipped following a PRECORD operation that has terminated due to an end-of-file or a beginning-of-file condition.	4
ASN	indicates the assignment type currently in effect for the DCB (0 = null, 1 = file, 2 = Xerox labeled tape, 3 = device, X'A' = ANS labeled tape).	0
BCERR	indicates whether or not a block count error has been detected during EOF/EOT processing (0 = no, 1 = yes). Always cleared before returning to user.	0
BLK	contains 1. the byte count of the record segment pointed to by either CBD or PBD, depending upon the point in time. 2. the number of bytes to be transferred by the I/O routines whenever called.	6
BLKCNT	specifies the number of blocks in the file.	17
BLKSZ	specifies the block size in bytes.	3
BUF	contains the address of the user's buffer where the data record is to be read or written, or where user trailer labels are to be read.	2
CIS	contains the relative position of the serial number (in the SN list) of the magnetic tape reel used for current file input.	11
CMD	contains the number of tape marks that may be passed during an OPEN while searching the last tape of a set.	20
CONCAT	specifies the number of identically named files that are to be read as one logical file (concatenation).	14
COS	contains the relative position of the serial number (in the SN list) of the magnetic tape reel used for current file output.	11

Field	Description	Word
CVI	indicates the relative volume number of the current input tape within the current file. CVI is taken from the beginning-of-file sentinel, which appears at the beginning of file and at the beginning of each reel, if the file is continued on more than one reel.	9
CVO	indicates the relative volume number of the current output tape with respect to the current file. CVO is recorded in the beginning-of-file sentinel which is written at the beginning of the file and at the beginning of each reel, if the file is continued on more than one reel.	9
DEV	contains the DCT index of the device assigned to the DCT. DEV is only meaningful if DEVF = 1. When DEVF = 0, the field is defined as OPLB.	1
DEVF	indicates whether the DCB is assigned to a device or an operational label. (0 = operational label, 1 = device.)	1
DEVICE	contains the EBCDIC name specified on the DEVICE option in the M:OPEN call. This use is only transient, and the field is later overlaid by SND.	12
DIR	indicates the direction of the read operations (0 = forward, 1 = reverse).	0
EGV	is the event-given flag and indicates whether or not the completion code posted in the TYC field has been communicated to the user's program by the CHECK routine (0 = no, 1 = yes). The CHECK routine is called either directly by the user or indirectly by the monitor, depending upon the WAIT, ERR, and ABN options in the FPT.	0
EOP	is the ending operation indicator (0 = other, e.g., rewind, 1 = read, 2 = write). Specifies the type of I/O operation currently or last performed.	0
EOT	indicates whether or not the physical end-of-tape mark has been encountered (0 = no, 1 = yes).	16
ERA	contains the address of the user's routine that will handle error conditions resulting from insufficient or conflicting information. (The monitor returns to the ERA in the FPT if the error condition is the result of the device failure.)	3
FCD	indicates whether the DCB is opened or closed (0 = closed, 1 = opened).	0
FCI	indicates whether the DCB has ever been closed. This flag is set when the DCB is first closed and then never reset (0 = DCB has never been closed, 1 = DCB has been previously open and closed).	0
FCN	indicates the current number of I/O operations that have been initiated but not completed, for this DCB.	7
FIL1	indicates the file option specified when the DCB was last opened (0 = none specified, 1 = release, 2 = save).	5
FLD	contains the word displacement to the file name in the DCB relative to the start of the variable length parameters (FLD + FLP = FWA of the EBCDIC file name).	21

Field	Description	Word
FLP	contains the address of the variable length parameters in the DCB (called the file list-pointer).	6
FMT	indicates the record format, where 1 = F (fixed length) 2 = D (variable, expressed in decimal) 3 = V (variable, expressed in binary) 4 = U (undefined)	5
FSN	specifies the file sequence number.	16
FUN	indicates the file mode function (0 = null, 1 = IN, 2 = OUT, 4 = INOUT, 8 = OUTIN).	1
HBTD	is the I/O handler's byte indicator and is used whenever the I/O routines are called to specify the byte displacement within QBUF into which the data transfer is to begin.	0
LRCSZ	specifies the logical record size in bytes.	
NRA	indicates the number of recovery tries that may be attempted before a device error message is to be logged.	2
NRT	indicates the number of recovery tries remaining before a device error message is to be logged.	1
PBD	contains 1. a counter used by M:OPEN to determine how many volumes remain to be searched for the specified file. 2. the number of bytes in the previous labeled tape block. PBD is only meaningful on a read operation and is taken from the PBS field of a labeled block.	20
QBUF	contains 1. the buffer address to be used by the I/O routines whenever called. 2. the address within the user's buffer where the next record segment begins.	7
RNDEV	contains the type of device specified (0 = none specified, 8 = 9T, 9 = 7T, X'A' = MT).	5
RWS	indicates the requested number of bytes to be read or written from the user's buffer (BUF). At the termination of the I/O operation, RWS is set equal to ARS.	13
SETID	specifies the file set identification.	19
SND	contains the word displacement to the tape serial number (SN list) in the DCB relative to the start of the variable length parameters (FLP + SND = FWA of the EBCDIC serial numbers).	12
SNFN	indicates the access method (0 = serial number, 1 = filename).	0
TLB	contains the address of a user's label that is to be written on a tape file when the file is output.	14

Field	Description	Word
TYC	indicates the type of completion of an I/O operation.	2

TYC Code	Corresponding Error/ Abnormal Code	Meaning
0	0	normal without device I/O transfer
1	0	normal with a device I/O transfer
2	7	lost data
3	1D	beginning-of-tape
4	4	beginning-of-file
5	1C	end-of-reel
6	5	end-of-data
7	6	end-of-file
8	41	read error
9	45	write error

TYPE	contains the device-type code for the tape assigned to this DCB.	1
UBTD	is the byte displacement indicator, specifying at which byte in the user's buffer (BUF) the data record begins.	0
ULBL	indicates whether or not the ULBL option was specified in the FPT of M:READ (0 = no, 1 = yes).	5
WAT	is the wait flag and indicates whether or not WAIT was specified in the FPT (0 = no, 1 = yes).	0

VARIABLE LENGTH PARAMETERS 22→n

Each variable length parameter entry for ANS labeled tapes is preceded by a control word of the following form:

Byte 0 = a code number (see Table A-2) identifying the parameter which follows.

Byte 1 = code for the entry position (00 = more parameter entries to follow, 01 = last parameter entry).

Byte 2 = number of significant data words in the parameter entry.

Byte 3 = total number of words reserved for the entry, not including the control word (that is, maximum entry length).

Table A-2. Variable Length Parameter Codes for ANS Labeled Tapes

Code	Parameter Type
01	File name (the first byte of which contains the number of characters in the name).
04	Expiration date.
07	SN/INSN serial numbers. (ANS serial numbers are encoded to fit in 32 bits.)
08	OUTSN serial numbers. (ANS serial numbers are encoded to fit in 32 bits.)

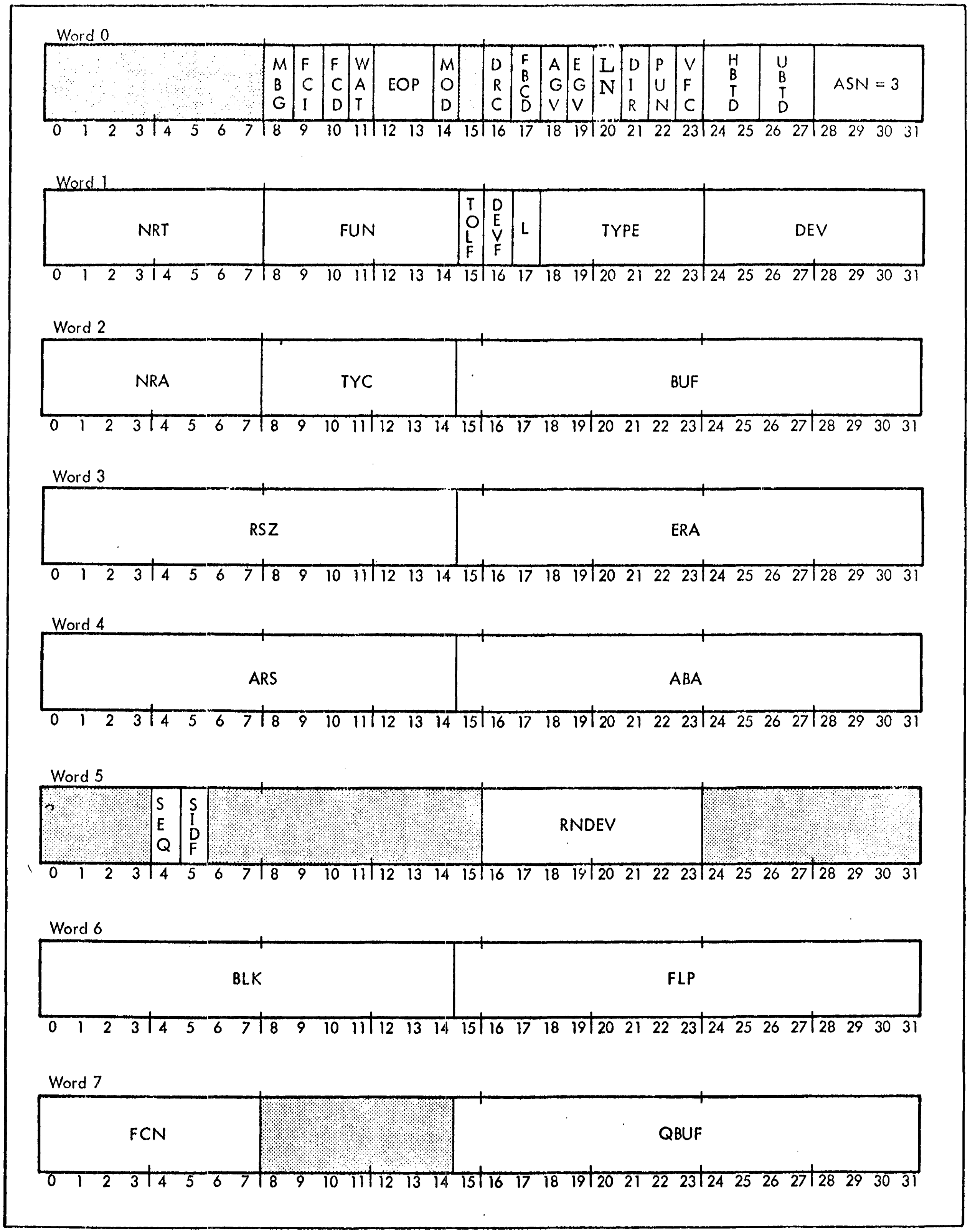


Figure A-3. Format of Device DCB

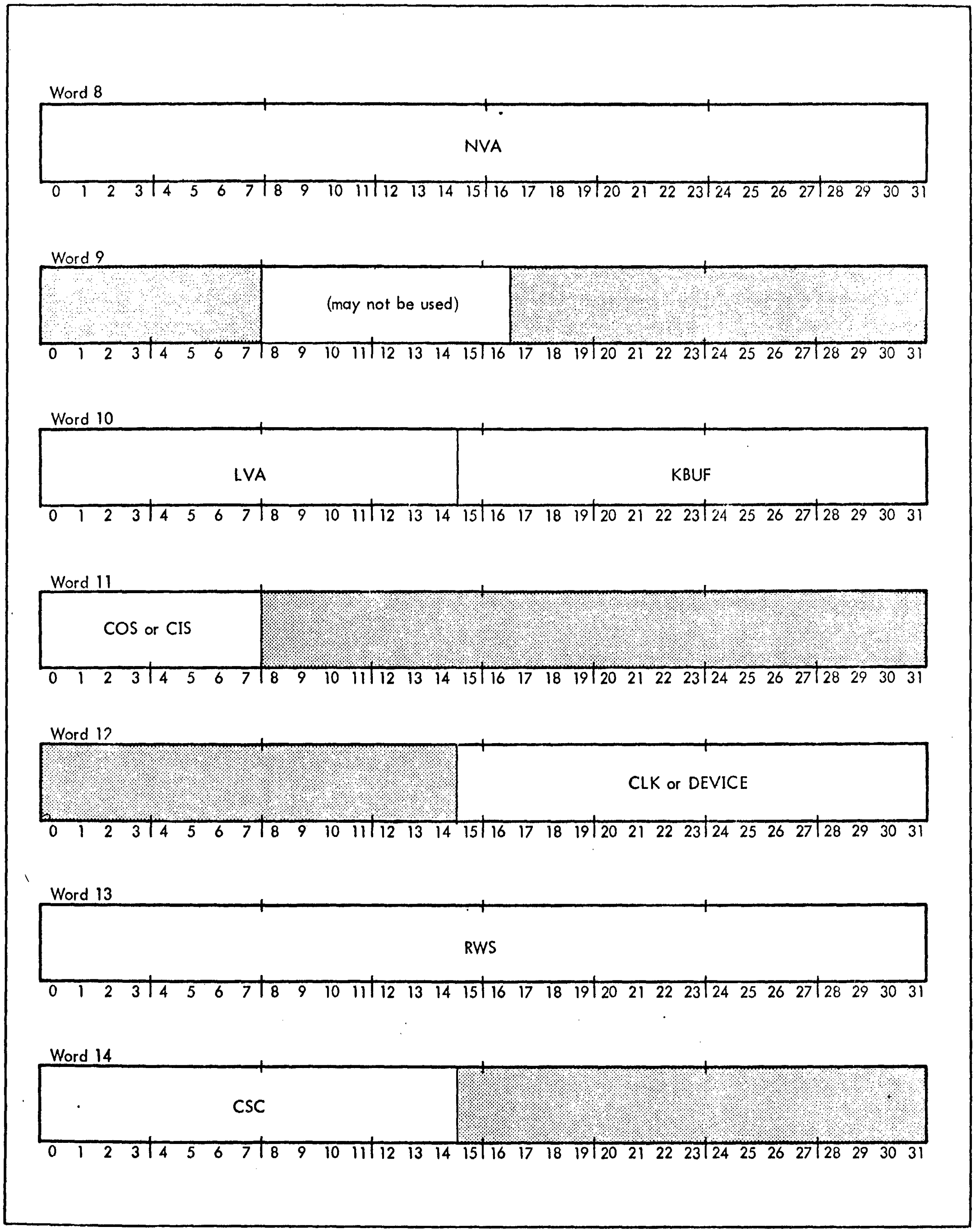
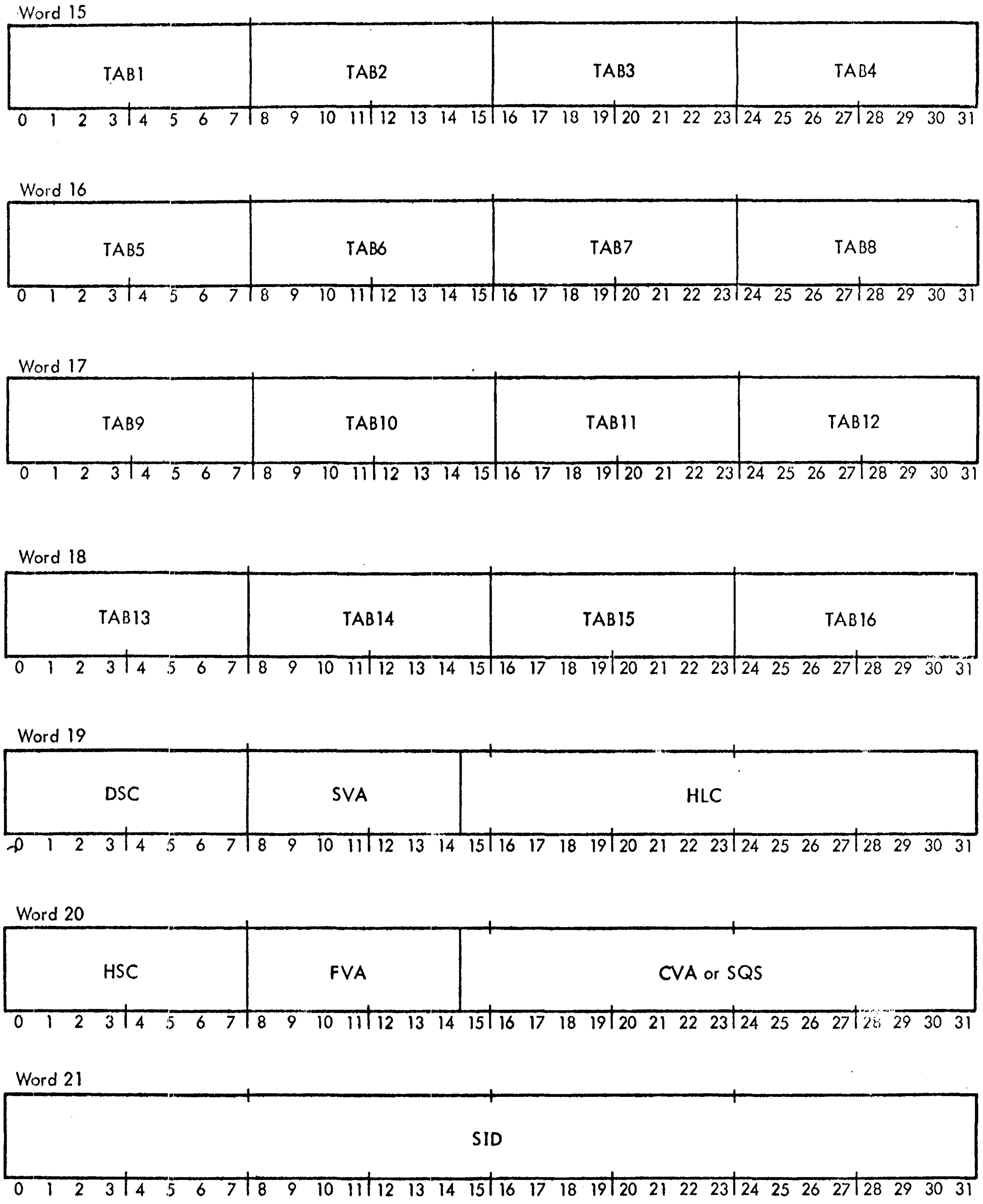


Figure A-3. Format of Device DCB (cont.)



Words 22 → n are used for variable length parameters

Figure A-3. Format of Device DCB (cont.)

Field	Description	Word
ABA	contains the address of the user's routine that will handle abnormal conditions resulting from insufficient or conflicting information. (The monitor returns to ABA in the FPT if the abnormal condition is the result of a device abnormality.)	4
AGV	is the abnormal given flag and indicates whether or not an end-of-file completion code has been returned to the user because a control command was encountered when reading from the C device, (0 = no, 1 = yes).	0
ARS	contains the actual number of data bytes transferred to or from the user in the I/O operation.	4
ASN	indicates the assignment type currently in effect for the DCB (0 = null, 1 = file, 2 = Xerox labeled tape, 3 = device, X'A' = ANS labeled tape).	0
BLK	contains the number of bytes to be transferred by the I/O routines whenever called.	6
BUF	contains the address of the user's buffer where the data record is to be read or written.	2
CIS	contains the relative position of the serial number (in the SN list) of the magnetic tape reel used for current file input.	11
CLK	for a nonsymbiont device, contains 0. For a symbiont device, contains the accounting type in bits 20-23 (0 = none, 1 = DO, 2 = PO, 3 = UO, 4 = LO) and the logical device index in bits 24-31.	12
COS	contains the relative position of the serial number (in the SN list) of the magnetic tape reel used for current file output.	11
CSC	indicates the number of the column at which the page count is to begin (for printer or typewriter). The most significant digit of the count will be printed in this column on the page.	14
CVA	indicates the current value of the page count (for printer or typewriter).	20
DEV	contains the DCT index of the device assigned to the DCB. DEV is only meaningful if DEVF equals 1.	1
DEVF	indicates whether the DCB is assigned to a device or an operational label (0 = operational label, 1 = device).	1
DEVICE	contains the EBCDIC name specified on the DEVICE option in the M:OPEN call. This use is only transient, and the field is later overlaid by CLK.	12
DIR	indicates the direction of the read operation (0 = forward, 1 = reverse).	0
DRC	is the format control flag and indicates whether or not the monitor is to do special formatting of records on read or write operations (0 = yes, 1 = no).	0

Field	Description	Word
DSC	indicates the column number at which the output record is to begin (for a card punch, typewriter, or printer).	19
EGV	is the event-given flag and indicates whether or not the completion code posted in the TYC field has been communicated to the user's program by M:CHECK (1 = yes, 0 = no). M:CHECK is called either directly by the user or indirectly by monitor, depending upon the WAIT, ERR, and ABN options in the FPT.	0
EOP	is the ending operation indicator (0 = other, e.g., rewind, 1 = read, 2 = write). Specifies the type of I/O operation currently or last performed.	0
ERA	contains the address of the user's routine that will handle error conditions resulting from insufficient or conflicting information. (The monitor returns to the ERA in the FPT if the error condition is the result of a device failure.)	3
FBCD	is the FORTRAN BCD flag and indicates whether or not BCD is to be converted to EBCDIC on input, or EBCDIC is to be converted to BCD on output. (0=no conversion, 1=conversion.)	0
FCD	indicates whether the DCB is opened or closed (0 = closed, 1 = opened).	0
FCI	indicates whether the DCB has ever been closed. This flag is set when the DCB is first closed, and then never reset (0 = DCB has never been closed, 1 = DCB has been previously opened and closed).	0
FCN	indicates the current number of I/O operations that have been initiated but not completed, for this DCB.	7
FLP	contains the address of the variable length parameters in the DCB (called the file list-pointer).	6
FUN	contains the file mode function (0=null, 1=IN, 2=OUT, 3=IN and OUT, 4=INOUT, 8=OUTIN).	1
FVA	indicates the first line on which printing is to begin (for printer or typewriter).	20
HBTD	is the I/O handler's byte indicator and is used whenever the I/O routines are called to specify the byte displacement within QBUF into which the data transfer is to begin.	0
HLC	contains the address of the user's page header that is to be output at the beginning of each listing page (the first byte of the page header contains the byte count).	19
HSC	indicates the column number at which the user's page header is to begin (for printer or typewriter).	20
KBUF	not used for devices but because of common program logic, the field contains a meaningless address.	10
L	indicates whether or not the user specified that the DCB was assigned to a listing type device. (0 = no, 1 = yes.) This flag is only used by the FORTRAN I/O routines. The monitor automatically sets this flag when the DCB is assigned to a listing type device (such as the line printer).	1
LN	**used in conjunction with ASN to determine that the DCB is a Transaction Processing slave line DCB.**	

Field	Description	Word
LVA	indicates the number of printable lines per logical page (for printer or typewriter).	10
MBG	is the monitor buffer-flag and indicates whether or not a 34-word output buffer has been allocated to the DCB from the monitor's buffer pool. (0 = the actual I/O operation will take place directly from the user's buffer, = the output record will be transferred from the user's buffer to the monitor's buffer and that the actual I/O operation will take place using the monitor's buffer.)	0
MOD	is the mode flag and indicates the device mode to be used in the I/O operation. (0 = EBCDIC, 1 = binary.) This flag is only used when 1. the DCB is assigned to a card punch or 7-track magnetic tape. 2. the DCB is assigned to a card reader and DRC has been specified.	0
NRA	indicates the number of recovery tries that may be attempted before a device error message is to be logged.	2
NRT	indicates the number of recovery tries remaining before a device error message is to be logged.	1
NVA	contains a counter indicating the number of records to skip on magnetic tape. It is also used as an indicator. If NVA is negative, the last operation performed was a rewind.	8
PUN	indicates whether a 7-track tape is to be read or written in the packed or unpacked mode (0 = unpacked, 1 = packed). PUN is only meaningful when MOD is set.	0
QBUF	contains the buffer address to be used by the I/O routines whenever called.	7
RNDEV	same as TYPE field, except when DCB is a slave line DCB. In this case, RNDEV has the value 0 if the DCB is associated with MOC controller and value = line # + 1 if associated with a COC line.	5
RSZ	indicates the default record size, in bytes.	3
RWS	indicates the requested number of bytes to be read or written from the user's buffer (BUF).	13
SEQ	is the sequence option flag and indicates whether or not punched output is to have sequencing in columns 77-80 (0 = no, 1 = yes).	5
SIDF	is the sequence identification (ID) flag and indicates whether or not punched output is to have sequence identification in columns 73-76 (0 = no, 1 = yes).	5
SID	contains the 4-byte EBCDIC identification to be output in the sequencing identification field (columns 73-76) of punched card output.	21
SQS	indicates the next sequence number to be output in columns 77-80 (for punched card output).	20

Field	Description	Word
SVA	indicates the number of lines to be spaced between printed lines (for typewriter or printer). A 0 means SPACE was not specified; the output will be single spaced.	19
TAB1-16	indicates the column numbers for the tab-stop settings (for output devices).	15-18
TOLF	if 1, bits 16-31 of DCB are TEXT OPLABEL. If 0, DEVF is meaningful.	1
TYC	indicates the type of completion of an I/O operation.	

TYC Code	Corresponding Error/ Abnormal Code	Meaning
0	0	normal without device I/O transfer
1	0	normal with device I/O transfer
2	7	lost data
3	1D	beginning-of-tape
4	4	beginning-of-file
5	1C	end-of-reel
6	5	end-of-data
7	6	end-of-file
8	41	read error
9	45	write error

TYPE	contains the device-type code assigned to the DCB. This field is set whether the DCB is assigned directly to a device or indirectly through an operational label.	1
UBTD	is the type displacement indicator, specifying at which byte in the user's buffer (BUF) the data record begins.	0
VFC	is the vertical format control-flag and indicates whether or not the first byte of the output is a format control character (0 = no, 1 = yes). This flag is only used for printer output.	0
WAT	is the wait flag and indicates whether or not WAIT was specified in the FPT (0 = no, 1 = yes).	0

VARIABLE LENGTH PARAMETERS 22 n

Each variable length parameter entry is preceded by a control word of the form shown for File DCB and in Table A-1.

LABELED TAPE, GENERAL FORMAT AND SENTINELS

Shown below are two labeled tapes containing two volumes of file A, having a total of four records, and the one-record file B. The various sentinels are explained following the tape format sketches. All sentinels begin on a word boundary.

Tape 1
Label Sentinel (:LBL)
Identification Sentinel (:ACN)
Tape Mark
Beginning of File A (:BOF)
User's Label (optional)
Tape Mark
Record 1 of File A
Record 2 of File A
Record 3 of File A
Tape Mark
End of Volume (:EOV)
Trailer Label (optional)
Tape Mark
End of Reel (:EOR)
Tape Mark
Tape Mark

Tape 2
Label Sentinel (:LBL)
Identification Sentinel (:ACN)
Tape Mark
Beginning of File A (:BOF)
User's Label (optional)
Tape Mark
Record 4 of File A
Tape Mark
End of File A (:EOF)
Tape Mark
Beginning of File B (:BOF)
Tape Mark
Record 1 of File B
Tape Mark
End of File B (:EOF)
Trailer Label (optional)
Tape Mark
End of Reel (:EOR)
Tape Mark
Tape Mark

where

:LBL identifies the reel number of the tape (SN). Reel number are four alphanumeric characters in length. Sentinel length is 12 bytes, including four padding bytes. The format is shown below.

:	L	B	L
X	X	X	X
4 PADDING BYTES			

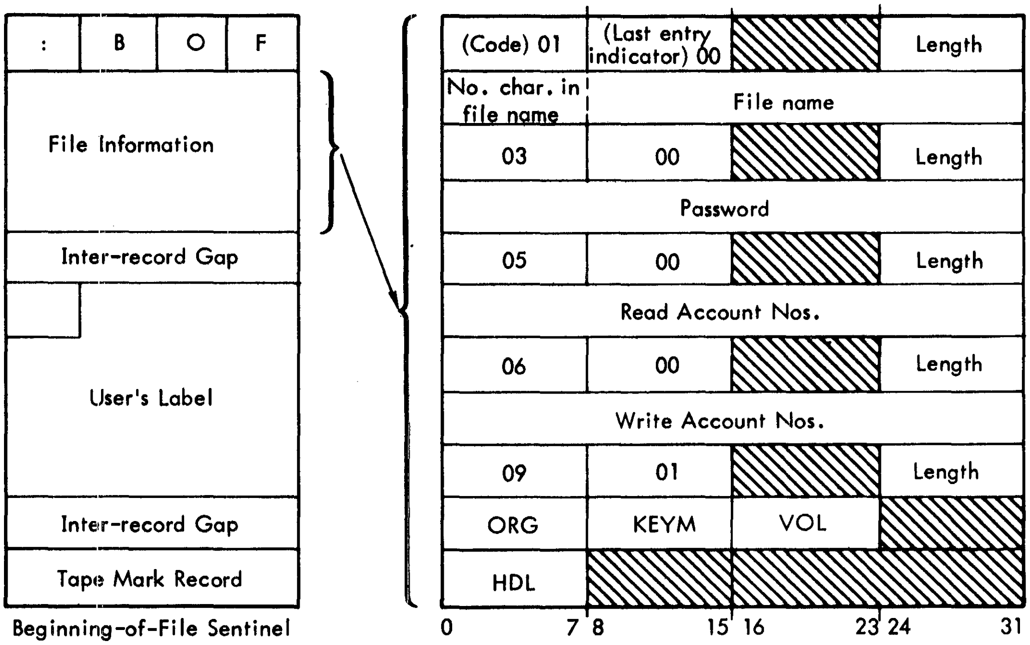
Label Sentinel

:ACN identifies the owner of the tape (ACCNT #), the expiration date, and the creation date, in that order.

The account number is eight alphanumeric characters in length, left-justified and in EBCDIC code. The dates are of the form $m_1m_2d_1d_2$b̸b̸y_1y_2, where m_1m_2 is the numerical representation of the month, d_1d_2, the day, b̸b̸ are blanks, and y_1y_2 are the last two digits of the year. The digits are in EBCDIC and the blanks must appear. Sentinel length is 28 bytes followed by a physical end-of-file (tape mark record). The format of the ACN Sentinel, also referred to as the identification sentinel, is shown below.

:	A	C	N
a_1	a_2	a_3	a_4
a_5	a_6	a_7	a_8
m_1	m_2	d_1	d_2
b̸	b̸	y_1	y_2
m_1	m_2	d_1	d_2
b̸	b̸	y_1	y_2
Inter-record Gap			
Tape Mark Record			

:BOF is the beginning-of-file sentinel consisting of the file-information record, the user's label (if specified), and a physical end of file (tape mark record). The format of the :BOF Sentinel is shown below.



:	B	O	F
File Information			
Inter-record Gap			
User's Label			
Inter-record Gap			
Tape Mark Record			

(Code) 01	(Last entry indicator) 00	▨	Length
No. char. in file name	File name		
03	00	▨	Length
Password			
05	00	▨	Length
Read Account Nos.			
06	00	▨	Length
Write Account Nos.			
09	01	▨	Length
ORG	KEYM	VOL	▨
HDL	▨	▨	▨
0 7	8 15	16 23	24 31

Beginning-of-File Sentinel

The file-information record, as shown, contains several control words and the information associated with each of these. The control words have the following form:

Code	LEI	▨	Length

where

Code identifies the type of information following the control word. The codes are:

01 - file name. The file name may be a maximum of 31 characters. An additional byte is used to state the length of the file name.

03 - password (2 words, left-justified).

05 - Read account numbers.

06 - Write account numbers.

Each account number is left-justified, blank-filled, and two words long. The total number of Read and Write accounts must not exceed 16. Read accounts identify those who may have only read access to the file. Write accounts identify those who may read and write the file. None or All are also allowed.

09 - miscellaneous information, such as:

ORG - gives file organization, i.e., keyed or consecutive.

KEYM - specifies maximum length of the keys. Keys may not be greater than 31 bytes. An additional byte is used to specify the length of the key. On consecutive files, the length of the dummy key is assumed to be three; therefore, KEYM is ignored. On keyed files, if KEYM=0, maximum length is assumed to be 11.

VOL - on multi-reel files, this entry specifies the position of this tape in the file. For example, VOL=2 implies this is the second tape of the multi-reel file. VOL=1 indicates the beginning of the file. Every file begins with VOL=1 (including single-reel files).

HDL - specifies the length of user's label. If HDL=0, then no user's label exists and the following record must be a physical end-of-file.

LEI is the last-entry indicator; this entry in the control word indicates the end of the file information. The control words, along with the information they define, do not have to be in a particular order, but LEI must equal 0 if the file information entry is not the last one and must equal 1 if the entry is the last.

Length specifies the length, in words, of the information associated with a particular entry (i.e., following the code word).

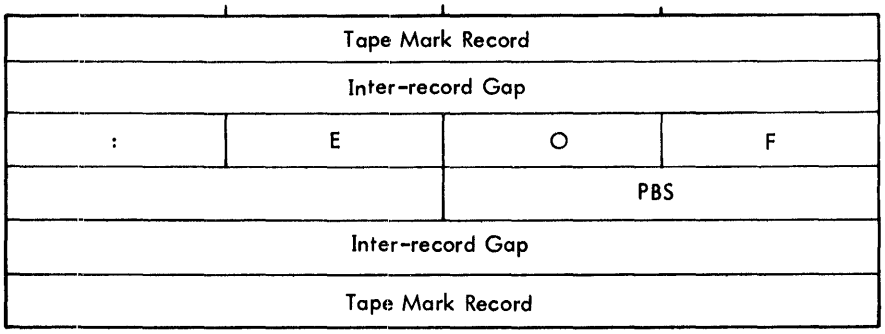
:EOV is the End-of-Volume sentinel:

:	E	O	V
		PBS	
Inter-record Gap			
Tape Mark Record			

:EOR is the End-of-Reel sentinel:

	E	O	R
		PBS	
Inter-record Gap			
Tape Mark Record			

:EOF is the End-of-File sentinel:

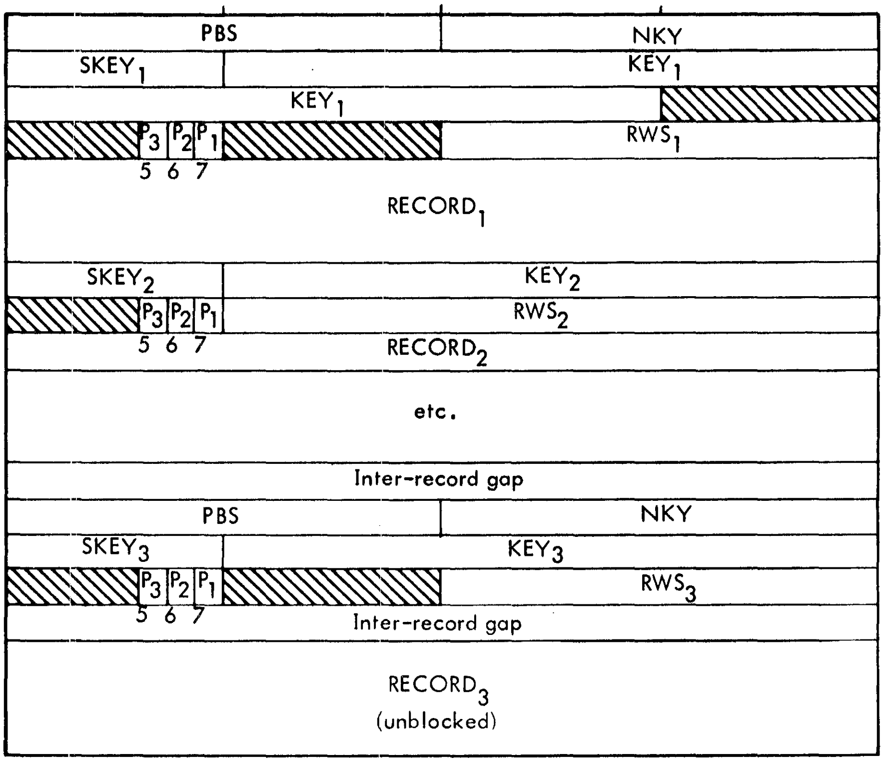


where

PBS is the previous block size, in bytes.

RECORD FORMAT

The labeled-tape record format for blocked records is shown below. The truth table following the diagram shows the possible combinations, of blocked/unblocked continued/not continued records. Figure 16-1 shows a number of examples of various types of records.

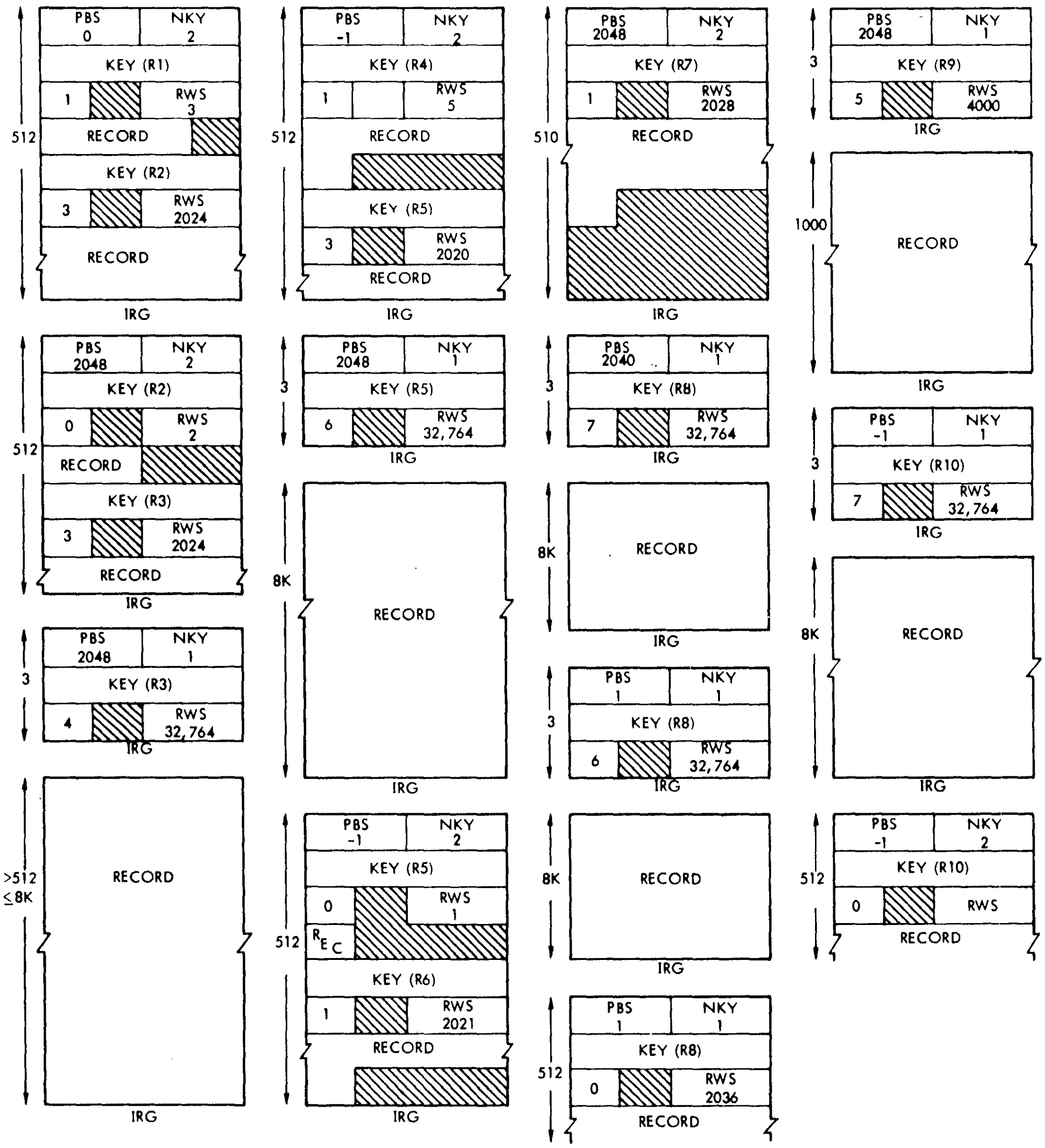


where

NKY is number of entries in block.

PBS is previous block size.

[†] Must start on word boundary.



Notes:
1. All numbers are decimal.
2. A new record will not be started in a block unless there is room for at least one full data word.
3. PBS — -1 whenever prior block is unblocked.
4. All keys are assumed to be three bytes long.

Figure 16.1 Examples of Different Types of Records

SKEY is size of key.

RWS is size of record in block.

P_3 is 1 if record is unblocked, 0 if blocked.

P_2 is 1 if record is continued into next block, 0 if record is not continued.

P_1 is 1 if this is first part of record, 0 if it is not first part.

Truth Table. Combinations of Record Types

Unblocked P3	Continued P2	1st. Segment P1	Meaning of Contol Bits
0	0	1	Record is wholly contained within this 512 word block.
0	1	1	1st record segment of N segments. Continued next block.
0	1	0	Not possible.
0	0	0	Nth record segment (N >1) of N. If N >2, ∴ N-1 was unblocked.
1	0	0	Physical record following is Nth segment of N (N > 1).
1	0	1	Physical record following is complete.
1	1	1	Physical record following is not complete. Size = 8K words.
1	1	0	Physical record following is not complete. Size = 8K words. This is ith segment of N (1 > i < N).

ANS TAPE, GENERAL FORMAT AND SENTINELS

The overall structure for ANS tapes and the sentind format are shown below.

Single file, single volume

VOL1 - HDR1 - HDR2 - UHL1 * file A * EOF1 - EOF2 - UTL1**

Single file, multi-volume

VOL1 - HDR1 - HDR2 - UHL1 * file A * EOV1 - EOV2 **

VOL1 - HDR1 - HDR2 - UHL1 * file A * EOF1 - EOF2 - UTL1 **

Multi-file, single volume

VOL1 - HDR1 - HDR2 - UHL1 * file A * EOF1 - EOF2 - UTL1 *
HDR1 - HDR2 - UHL1 * file B * EOF1 - EOF2- UTL1 **

Multi-file, multi-volume

VOL1 - HDR1 - HDR2 - UHL1 * file A * EOV1 - EOV2 **

VOL1 - HDR1 - HDR2 - UHL1 * file A * EOF1 - EOF2 - UTL1 *
HDR1 - HDR2 - UHL1 * file B * EOF1 - EOF2- UTL1 **

Note: An asterisk denotes a physical tape mark.
A dash denotes an inter-record gap.

ANS TAPES - Block counts, file and volume sequence numbers

Single file, single volume

VOL1 - HDR1 1 1 0 * file A - file A - file A * EOF1 1 1 3 **

Single file, multi-volume

VOL1 - HDR1 1 1 0 * A - A - A * EOV1 1 1 3 **

VOL1 - HDR1 2 1 0 * A - A - A - A * EOF1 2 1 4 **

Multi-file, single volume

VOL1 - HDR1 1 1 0 * A - A * EOF1 1 1 2 * HDR1 1 2 0 * B * EOF1 1 2 1 **

Multi-file, multi volume

VOL1 - HDR1 1 1 0 * A - A * EOF1 1 1 2 * HDR1 1 2 0 * B * EOV1 1 2 1 **

VOL1 - HDR1 2 2 0 * B - B - B * EOF1 2 2 3 **

NOTE:

An asterisk denotes a physical tape mark.

A dash denotes an inter-record gap.

The three numbers in HDR1/EOV1/EOF1 represent volume sequence number, File sequence number and block count respectively.

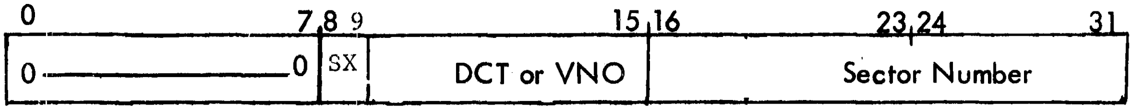
ANS TAPE LABELS FORMAT

HEADER	CHARACTER POSITION	DESCRIPTION	SOURCE/VALUE
VOL1	1-4	VOL1	
	5-10	Volume serial number	AVRTBL
	11	Security	0
	12-80	Not used	Blanks
HDR1/EOV1/EOF1	1-4	HDR1/EOF1/EOV1	
	5-21	Filename	DCB:FN (VLP=01)
	22-27	Serial number first volume	AVRSID
	28-31	Volume sequence number	DCB:CVO
	32-35	File sequence number	DCB:FSN
	36-41	Not used	Blank
	42-47	Julian creation date	date
	48-53	Julian expiration date	DCB:EXPR (VLP=04)
	54	Security	0
	55-60	Block count	DCB:BLKCNT
	61-73	CP-V version number	Core address X'2B'
	74-80	Not used	Blanks
HDR2/EOF2/EOV2	1-4	HDR2/EOF2/EOV2	
	5	Format	DCB:FMT
	6-10	Block size	DCB:BLKSZ
	11-15	Logical record size	DCB:LRCSZ
	16	Recording density	2(800 BPI)
	17	System use	0
	18-50	Not used	Blanks
	51-52	Buffer offset	0(4 if Format =V)
	53-80	Not used	Blanks

GENERALIZED DISK ADDRESS

File Management routines work with generalized disk addresses which are the addresses of granules on disk or DP. IOQ is responsible for correctly converting the disk address to seek address for the specific device.

GENERALIZED DISK ADDRESS FORMAT



where

DCT is the DCT index of the device (DCT indexes start with 1). Used for public files at all times, but for private files only when IOQ is called.

VNO the volume number within a private volume set, if the file is private. VNO is converted to a DCT index immediately before IOQ is called.

Sector number is the relative sector number (starting with 0) of the sector on the device. SX is the sector extension field. Bit 9 is the most significant bit of the sector number, and bit 8 is the next most significant bit.

Consecutive File Format

All information in the consecutive file format is contained in full granule sized records. There are no master index half granules used. Many files consist only of control granules which are flinked and blinked and contain all record data and control information. Each record is broken into segments with length less than or equal to 2048 bytes. Record segments which are shorter than 2033 bytes are blocked into the control granules in monitor blocking buffers for input or output: but record segments of greater length are written from, or read to, the user's buffer directly.

Zero length records are treated as no record on output. Records which are less than 2049 bytes long appear as one segment, unless the record is shorter than 2033 bytes but longer than the amount of space available in the final granule of the file at the time it was written. If such is the case, the record is broken into two segments, the first filling out the current granule and the second starting a new final granule. Records with length greater than 2048 bytes are broken into a sufficient number of 2048 byte segments to reduce the remainder to less than 2049 bytes. This remainder is treated as described above.

Record Segments - Each record is broken into segments $\leq$ 2048 bytes.

Case 1. Record Segment < 2033 bytes

Control Word:

bit 0 = 0 (for blocked)

bit 1 = FAK = 1 for initial segment of record, 0 if not initial.

bit 2 = C = 0 for last segment of record, 1 if otherwise

bit 3 = 0

bit 4-15 = # of data bytes in segment (call it m)

bit 16-31 = word position of previous record segment control word in this granule; or if this is the first segment in the granule, 0.

The next $\left[\frac{m+3}{4}\right]$ words contain the data of the segment.

Case 2. 2033 bytes $\leq$ Record Segment $\leq$ 2048 bytes

Control Word:

bit 0 = 1 (for unblocked)

bit 1 = FAK

bit 2 = C

bit 3 = 1 if preceding word in the granule is a backspace control word (see below).
= 0 if this is the first record segment of the granule or if the preceding record segment in the granule is also unblocked.

bits 4-7 = # of data bytes in the segment less 2033.

bits 8-31 = generalized disk address of the granule containing the data starting at byte 0.

Case 3. Backspace Control Word. (This word is used only when a granule contains a blocked segment followed by an unblocked segment whose control word is in the same granule. The word is inserted following the data of the blocked segment and preceding the control word of the unblocked segment.)

bits 0-15 = 0

bits 16-31 = word position of previous record segment control word in this granule.

Control Information for Control Granules

Word 0 Blink
Word 1 Flink
Word 2

bit 0 = 0 if no case entries appear in the granule, 1 otherwise.

bit 1 = 0 for granule not full, 1 otherwise.

bits 2-15 = 0

bits 16-31 = If bit 1=0, the word position of the next available word in the granule.
If bit 1=1, the word position of the last segment control word in the granule.

The initial granule of a file contains the FIT for that file in words 4-83 of the granule. Word 3 contains a dummy segment control word of type described in Case 1 above with FAK = 0, C = 0, and M = 320. If a record is deleted from a consecutive file, the FAK bit in the first (or only) segment control word for the record is reset to 0.

The following statements apply to the use of granules for consecutive files:

1. If the remainder fo a record to be written is at least 2033 bytes long, up to 2048 bytes will be written unblocked except in one very rare circumstance (see paragraph 4 below).
2. If the remainder of a record to be written is at least 2033 bytes long, it will be written as one or two blocked segments. If the remainder will not fit entirely in the appropriate granule, as much as will fit is placed in that granule, and the remainder is placed in the succeeding granule as a continuation record segment.
3. All bytes of each granule are used except:
   a. some number of bytes in the final granule of the file beyond the end of file;
   b. up to fifteen bytes at the end of a data granule for an unblocked segment;
   c. the final four bytes of a control granule will contain a backspace control word if a blocked segment would have otherwise started there;
   d. up to three bytes per blocked final segment of a record will be unused if the record length is not congurent to zero modulo 4.
4. If the data granule of an unblocked record segment would fall into a different volume of a private volume set from the volume containing the control word for that segment, as many bytes from the start of the segment as will fit into the control granule are written in a blocked fashion to fill out the control granule.

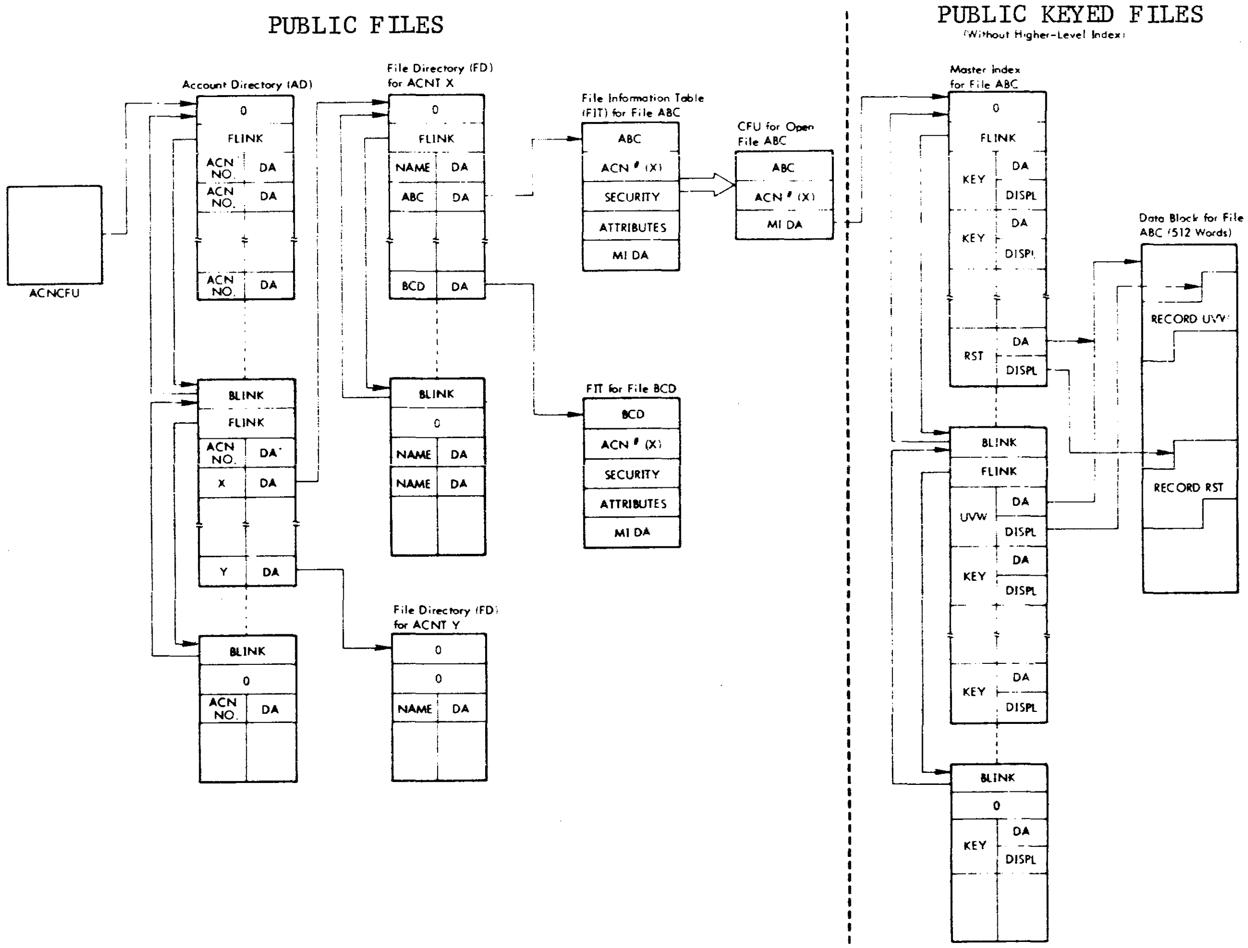
The following changes to the CFU (FIT) and DCB are made:

1. TDA in the CFU (FIT) contains the number of records in the file.
2. FDA and LDA in the CFU (FIT) now contain the appropriate granule addresses as opposed to half-granule addresses.

3. The TRN bit in word 5 of the DCB is 1 only if the most recently executed operation on the file was a read backwards.
4. The fourteenth word of the DCB (W14) contains one of the following:
   a. 0 if at BOF
   b. the contents of TDA, if at EOF
   c. the sequential record number of the record most recently read or written.
5. The nineteenth word of the DCB (W14) contains the direction (+ or −) and the number of records that must be skipped from the position indicated in W14 pripr to a data transfer operation (read, write, or delete).
6. The CMD halfword in word 20 of the DCB contains a word position in the granule pointed to by DCBCDAM (see below).
7. DCBCDAM in word 21 of the DCB contains a disk address of a granule reflecting (in conjunction with CMD and W14) the location in the file at which the most recent data transfer operation took place.

It should be noted that all positioning operations for consecutive files will be done with no I/O. Positioning operations are PRECORD, PFIL, and OPEN with extension. When these operations are encountered, the appropriate modification is made to W19 of the DCB. Only when a data transfer operation is about to take place will the positioning be effected; and at that time, there will be three known points in the file which can be used as a starting point (beginning-of-file, end-of-file, and the position chosen will be the one which requires the fewest record skips to be made.

On a delete forward operation on a consecutive file, all vacated granules will be returned to the availability pool at that time rather than when the file is released.



OPEN

1. First block of AD read into index buffer.
2. If desired ACN not there, second block of AD read into index buffer, etc.
3. When ACN is found, disk address (DA) of account's FD is available.
4. First block of FD read into same index buffer.
5. If desired file not there, second block of FD read into index buffer, etc.
6. When file is found, disk address of file's FIT is available
7. FIT is read into same index buffer.
8. Security checks are made.
9. File's attributes initialize DCB
10. FIT less security information and file attributes transferred to CFU and index buffer released.
11. File is now open

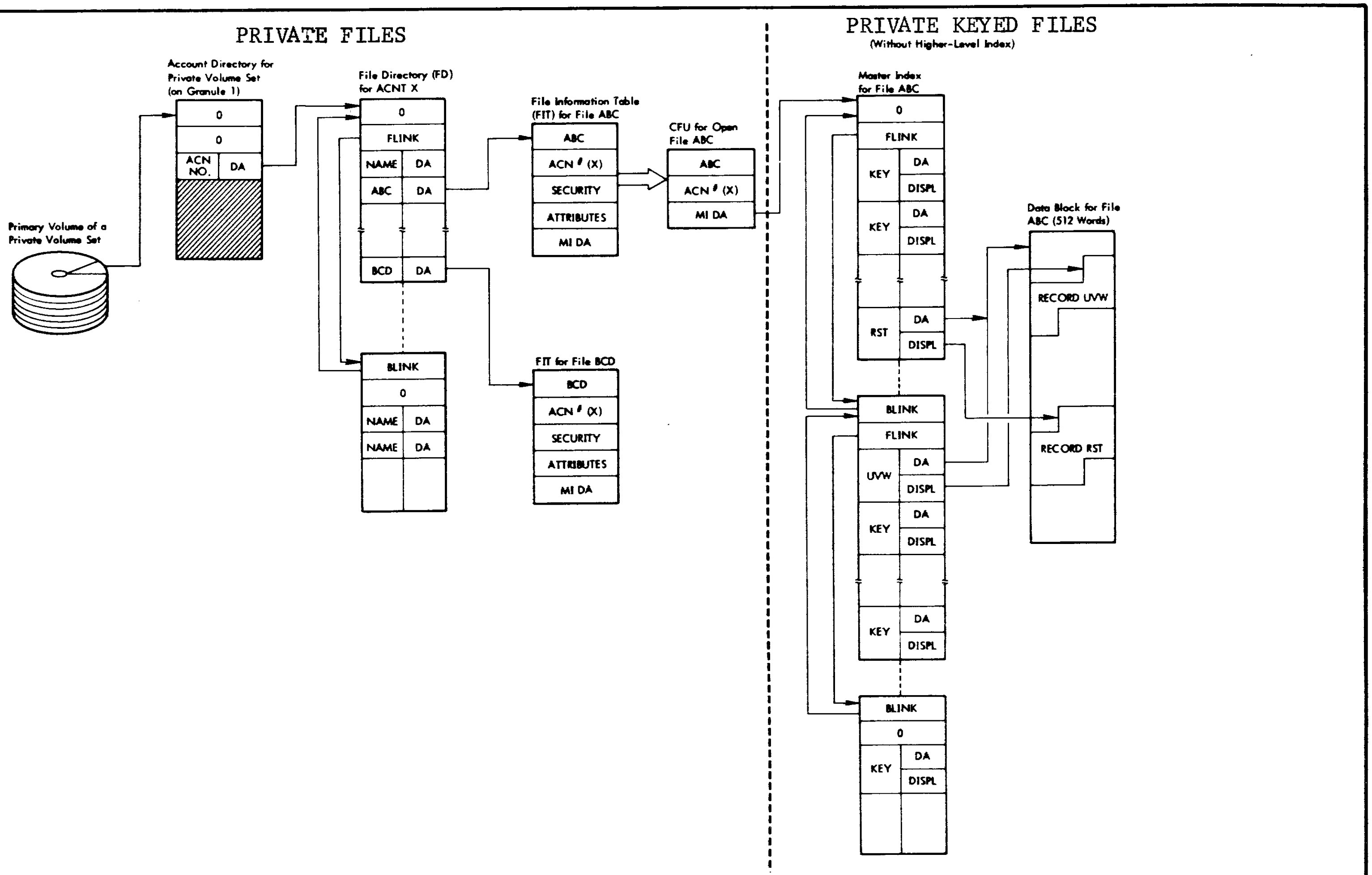
FIRST READ

1. First block of MI read into index buffer.
2. It desired key not there, second block of MI read into index buffer, etc
3. When key is found, DA of data block in which record is contained is available
4. Obtain blocking buffer (BB) and read data block.
5. Transfer record to user's buffer

SUBSEQUENT READS

6. Requested key searched in current MI block.
7. If not there, read next adjacent MI block
8. When key is found, determine if BB contains required data block
9. If not, read required data block into BB.
10. Transfer record to user's buffer.

VH. 19. Schema for Locating and Referencing Public Files



OPEN

1. Private-Volume-Set Account Directory read into index buffer from granule 1 of the set's primary volume.
2. File's account number checked against volume's account number.
3. Account number entry contains disk address of the private volume set's file directory.
4. First block of FD read into same index buffer.
5. If desired file not there, second block of FD read into index buffer, etc.
6. When file is found, disk address of file's FIT is available.
7. FIT is read into same index buffer.
8. Security checks are made.
9. File's attributes initialize DCB.
10. FIT less security information and file attributes transferred to CFU and index buffer released.
11. File is now open.

FIRST READ

1. First block of MI read into index buffer.
2. If desired key not there, second block of MI read into index buffer, etc.
3. When key is found, DA of data block in which record is contained is available.
4. Obtain blocking buffer (BB) and read data block.
5. Transfer record to user's buffer.

SUBSEQUENT READS

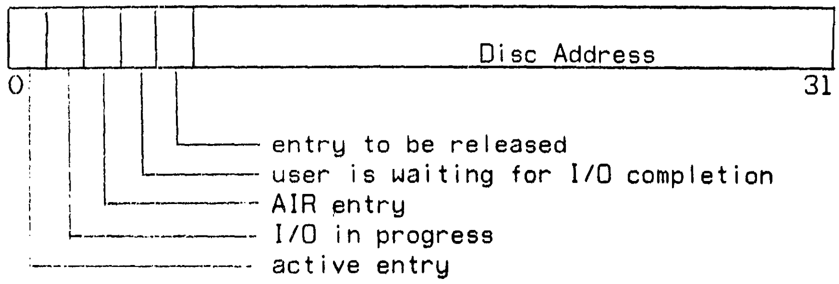
6. Requested key searched in current MI block.
7. If not there, read next adjacent MI block.
8. When key is found, determine if BB contains required data block.
9. If not, read required data block into BB.
10. Transfer record to user's buffer.

VH. 20 Schema for Locating and Referencing Private Files

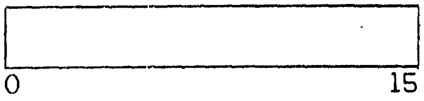
READ-AHEAD TABLES

Parallel Tables - Length = RASIZE

RA:DA

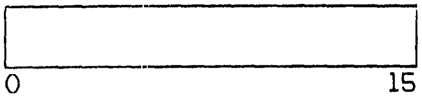


RAH:DCB *



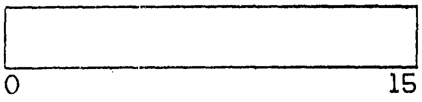
Most significant 16 bits of DCB address

RAH:TIME



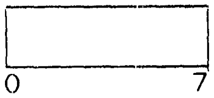
Time at which this entry was gotten (32 msec units)

RAX:PAGE



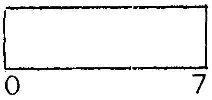
Physical page address of buffer for this entry

RAB:USER *



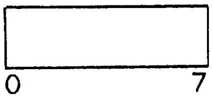
User number of user associated with this ENTRY

RAB:FLINK



Index of next oldest active entry

RAB:BLINK



Index of next newest active entry

The zeroth index into the table is used as follows:

RAB:USER	links the unused entries
RAB:FLINK	links the used entries from oldest to newest
RAB:BLINK	links the used entries from newest to oldest

* UNUSED FOR AIR ENTRIES

Single Cells

RA:CURA	number AIR entries currently active
RA:CURR	number read-ahead entries currently active
RA:CURF	number entries in free pool (inactive but with a buffer page associated
RA:ABNTB	number operations not started because of not enough table entries
RA:ABNCO	number operations not started because no free core page available
RA:ABNTA	number AIR entries timed out
RA:ABNTR	number read-ahead entries timed out
RA:ABNNN	number read-ahead entries not needed
RA:ABNWP	number times a read-ahead buffer is requested but the read-ahead that was done for a different disc address
RA:ASTOR	number attempts to add a file directory (AIR) entry
RA:SSTOR	number AIR entries successfully added
RA:AINIT	number attempts to start a read-ahead
RA:SINIT	number read-aheads started
RA:AGET	number attempts to retrieve an AIR entry

RA:SGET	number AIR entries successfully retrieved
SL:RAMR	maximum number simultaneous read-aheads allowed
SL:AIRM	maximum number simultaneous AIR entries allowed
SL:RAMF	maximum number pages allowed in free pool
SL:RATOR	time-out for read-ahead entries
SL:AIRTO	time-out for AIR entries

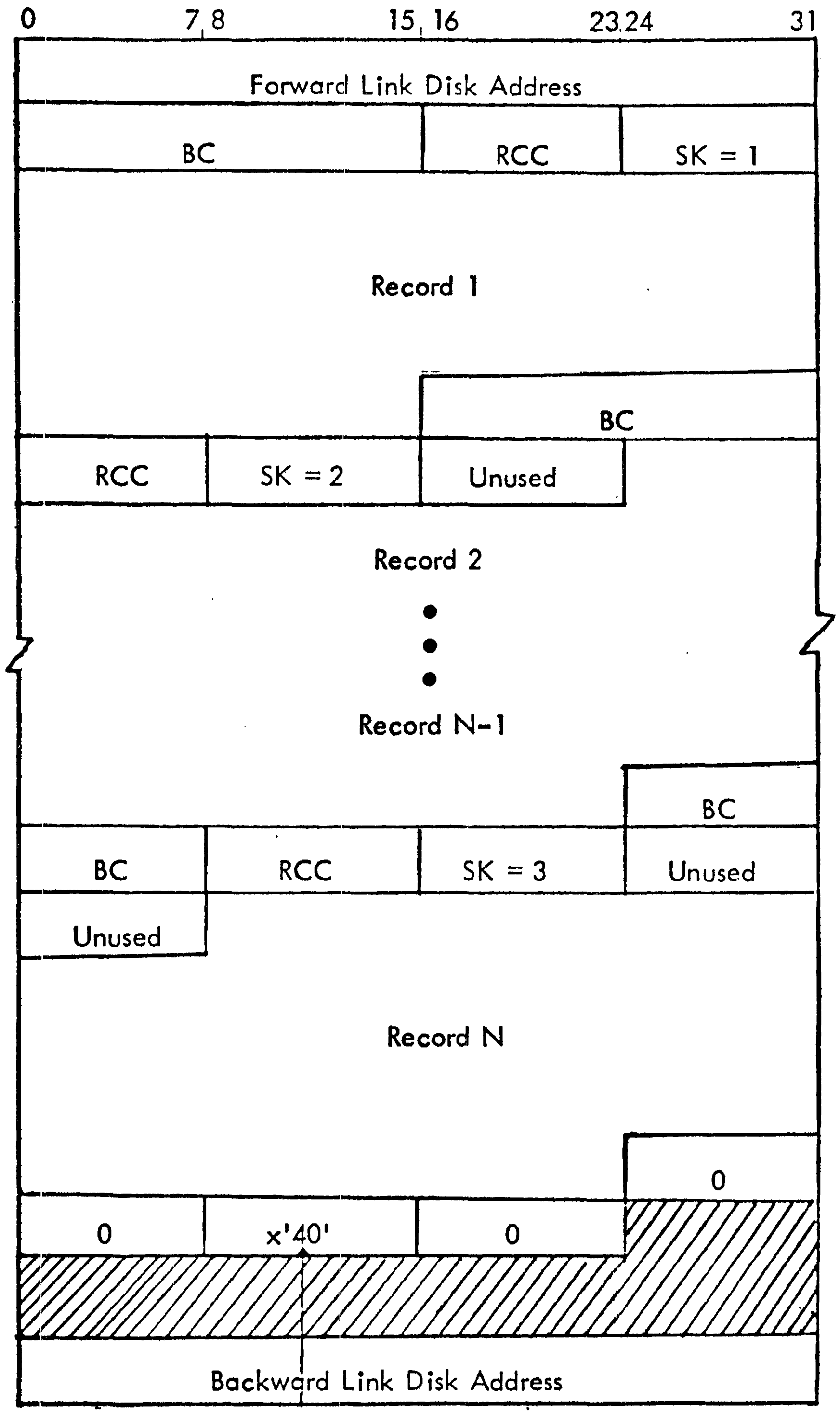
SYMBIONT COOPERATIVE DATA BASES

Symbiont File Block Format - In CP-V both input and output symbiont file blocks have the same format. Each such block contains 256 words, and two blocks reside in a granule of file storage. Word 0 of the block is used for forward link address that is inserted by the system when the file is created. A value of zero implies no forward address (i.e., end of file). Word 255 is used for the backward link address again inserted when the file is created. A value of zero implies no backward address (i.e., beginning of file). Each record in the block is preceded by four bytes of control information. Neither the record nor the control information need start on a word boundary except the first control string. Each control string must immediately follow the preceding record. The first two bytes of a control string are the byte count (BC) of the following record. BC must be greater than zero and less than 1008. No record may be split between blocks. If a block does not have space remaining for a block end control string, a record control string and a record, the next record must begin in a new block. The third byte of a control string is the record control character (RCC) which defines the record.

RCC =	0	a BCD record (e.g., card)
	1	an EOD record (e.g., !EOD)
	2	a binary record (BIN)
	4 or 5	a PRINT record without a VFC character
	6 or 7	a PRINT record, the first byte of which is a VFC character
	x'40'	a block ending control string (i.e., no more records this block)
	X'86'	header line for on-line output.
	X'88'	header line for output repackaged by recovery.
		SK=3 at least for header lines; the first two skipped bytes are the number of times the line is to be printed and the VFC character to be used each not-first time the line is printed.

Other values for RCC are reserved for future enhancements and should not be used. The fourth byte of a control string is the skip byte (SK) defined for the convenience of the block encoder. SK may have the values 1 through 4 inclusive.

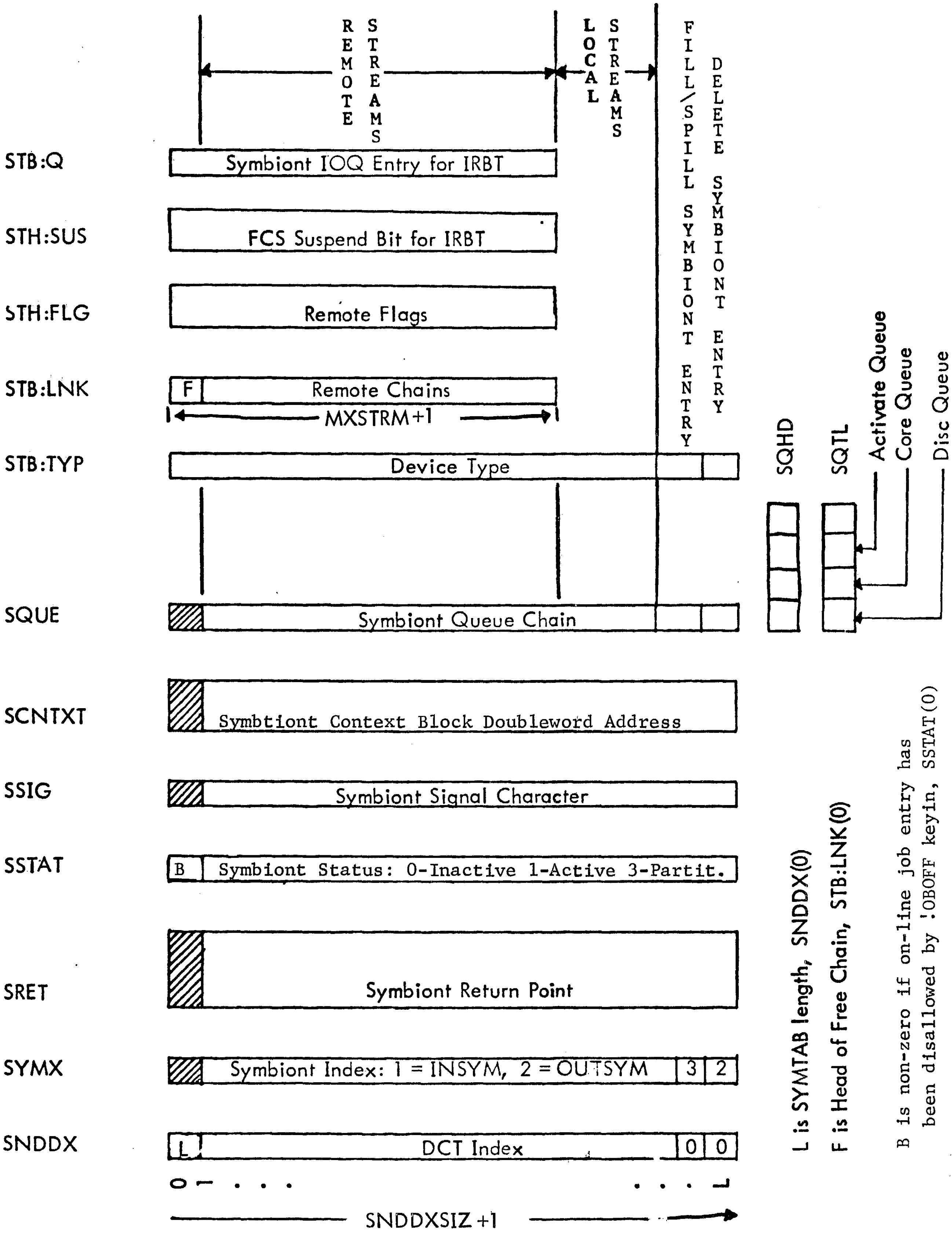
The next SK-1 bytes following the control sequence have no significance and are skipped before the start of data. The skipped bytes are provided to allow a byte aligned MBS instruction (the most efficient execution) to move the bytes into the symbiont block, or to allow placement of the record on a word boundary for record construction ease. The final control string of a block must have RCC = x'40'. BC and SK are not relevant in the final control string.



SYMBIONT FILE BLOCK FORMAT

End of data this buffer. If forward link disk address = 0, this is EOF.
If not, file is continued at forward link.
Records are never split between blocks.

SYMTAB - SYMBIONT TABLES



L is SYMTAB length, SNDDX(0)

F is Head of Free Chain, STB:LNK(0)

B is non-zero if on-line job entry has been disallowed by '!OBOFF keyin, SSTAT(0)

FORMAT OF SYMX ENTRY

0	1	2	3	4	5	6	7
TERM	LIST	OCP	KEEP	IRBT	NC	OUT	IN

IN = 1	signifies input symbiont
OUT = 1	signifies output symbiont
NC = 1	signifies non-control mode if IN signifies write EOF if OUT
IRBT	device is on an IRBT

IN and OUT are mutually exclusive.

KEEP = 1	signifies that granules are kept until done
OCP = 1	signifies OCP device
LIST = 1	signifies Listing type device
TERM = 1	signifies terminate on EOF

DEVICE DCB - WORD 12

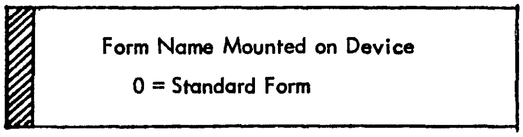
0 7	8 15	16 19	20 23	24 31
		▨	ACCO	LDEVX

where:

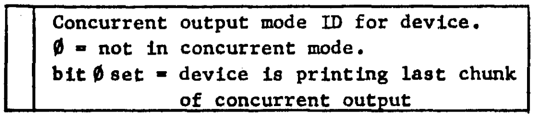
ACCO =	0,	no accounting
	1,	Do-type accounting
	2,	PO-type accounting
	3,	UO-type accounting
	4,	LO-type accounting
LDVEX =		index into SH:LNM

SYMBIONT TABLES IN RBBAT

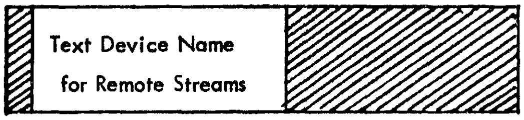
STW:FORM



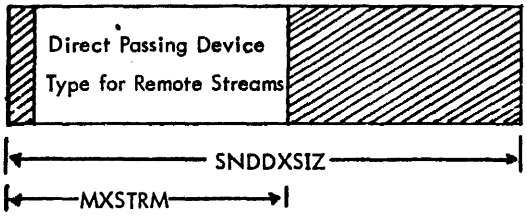
STH:CMDV



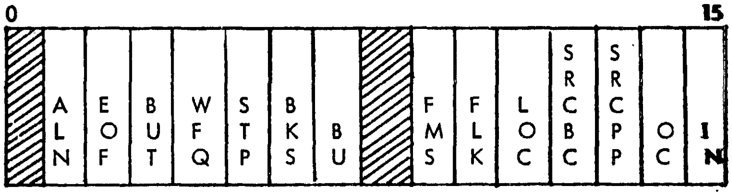
STH:NM



STB:DPD

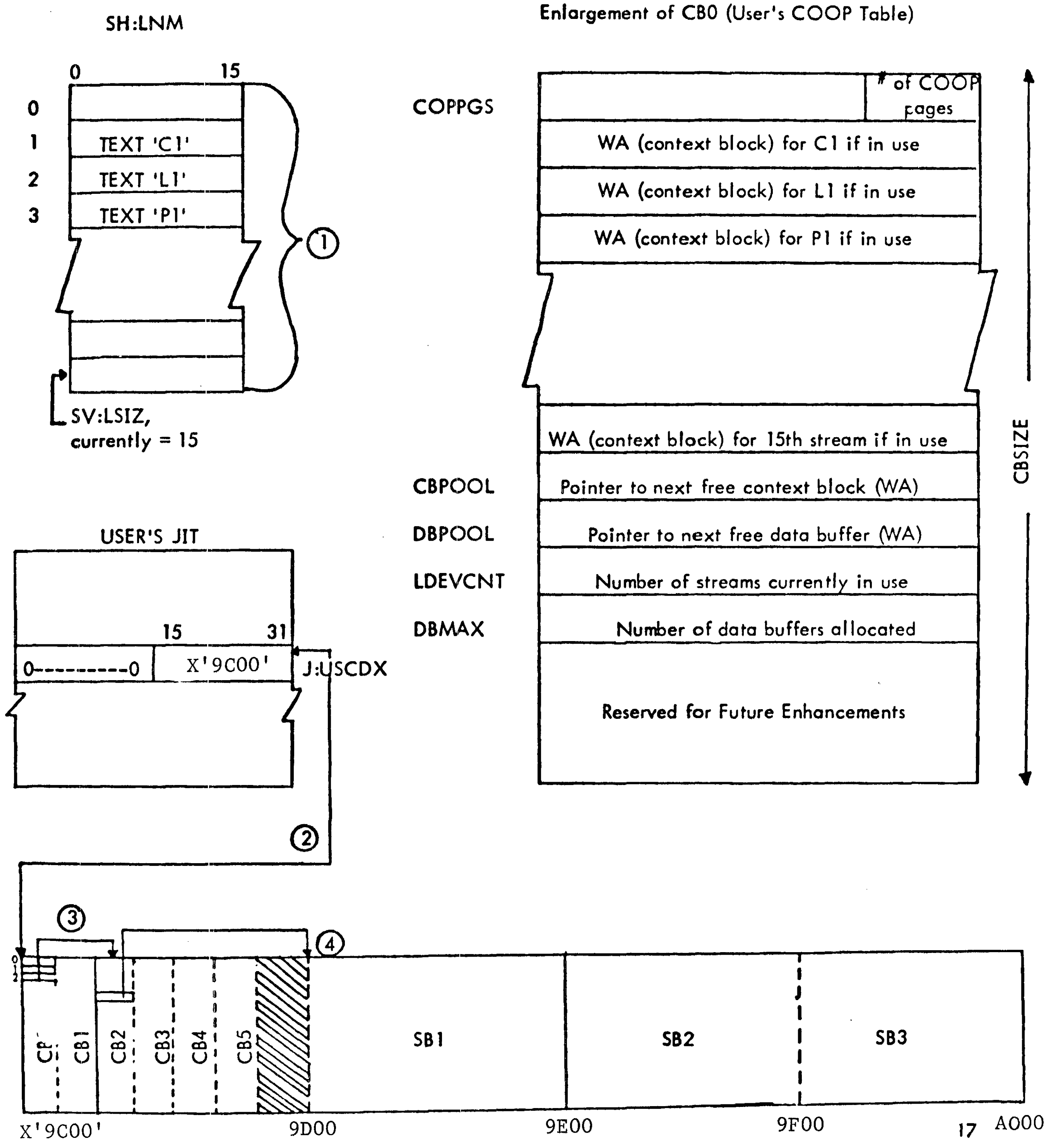


FORMAT OF STH:FLG ENTRY



IN	Input	BU	FCS device
OC	Operator console	BKS BUT	Suspend control
SRCBP SRCPC	SRCB type of device	STP	Output ready
LOC	Device locked if set	WFQ	Waiting for symbiont start
FLK	CP-V to CP-V	EOF	EOF pending
FMS	Forms control	ALN	RBALIGN pending

LINKAGE BETWEEN LOGICAL DEVICE TABLES



Notes on diagram "LINKAGE BETWEEN LOGICAL DEVICE TABLES"

1. Given a stream name in text, find index into SH:LNM (e.g., L1 has index 2).

2. Fetch J:USCDX which points to base of user's COOP table (X'9C00' in example).

3. Use index from Step 1 to fetch context block pointer for this stream (e.g., L1 has context block CB2).

4. Fetch stream's data buffer address from context block using displacement symbol SCFBUF (e.g., L1 has data buffer SB1 as indicated in CB2).

Symbiont/Cooperative Context Block - The context block for symbiont/cooperative operations has been totally redefined for CP-V. The salient features of the new context block format include:

1. Definition of new displacement symbols, conforming to more rigorous conventions than previously (i.e., all start with "SC"; "SCF" signifies a file-related value; "SCD" signifies a device-related value). All current symbols (e.g., SCBINFOX, SCJOBX) have been discarded.

2. Space is provided in the new context block for NEWQ arguments used in performing disc and device I/O operations (SCFQARGS, SCDQARGS). Notice that there is not a DCB in the CP-V context block.

3. Space is provided for "stream attribute" values which are modifiable by the user via the LDEV command (e.g., SCSEQ, SCFORM, SCMISC). All stream attribute fields are used in the cooperative context block; some of them are free for different use by INSYM and OUTSYM (since LDEV acts directly on the COOP's context block only).

4. Space is provided for values maintained internally by the symbiont/cooperative system, including values related to symbiont file construction (SCBLDA, SCFLDA, SCSVDGI) and values used for accounting purposes (SCRCO, SCPCO, SCGCO).

WORD	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
0	0 SYMTABX	SYMBIONT BASE ADDRESS		
1				DEVICE TYPE
2				
3		BLINK DISC ADDRESS		
4		FIRST DISC ADDRESS OF FILE		
5	WRITE DISC FUNCTION CODE	I/O PRIORITY	NRT	0------------------0
6		BLOCKING BUFFER BYTE ADDRESS		
7			BLOCK BYTE COUNT	
8		FILE CURRENT DISC ADDRESS		
9		FILE SIDE:END ACTION ADDRESS		
10	READ FUNCTION CODE	I/O PRIORITY	NRT	DEVICE DCTX
11	CURRENT BLOCK POINTER (BYTES)			
12	MAXIMUM RECORD BYTE COUNT			
13				SYMTABX
14		DEVICE SIDE:END ACTION ADDRESS		
15	DISC WRITE FOLLOWON: 0 READ A CARD, 1 ADD INPUT FILE			
16	4 WORD TEMPORARY SAVE AREA			
17				
18				
19				
20				
21	BLOCKING MODE: 1 RB CC's, 0 NORMAL			
22	(COUNT OF RECORDS READ) - 1			
23	COUNT OF JOB CARDS			
24	# OF GRANULES			
25	BASE OF 3 WORD "NO COMM BUFF" TEMP AREA			
26				
27				
28	FIN FIN = 1 IF FIN CARD HAS BEEN READ			
29				SYMX BYTE FOR THIS STREAM
30	NAME FROM !INCTL CMD			

INSYM CONTEXT BLOCK

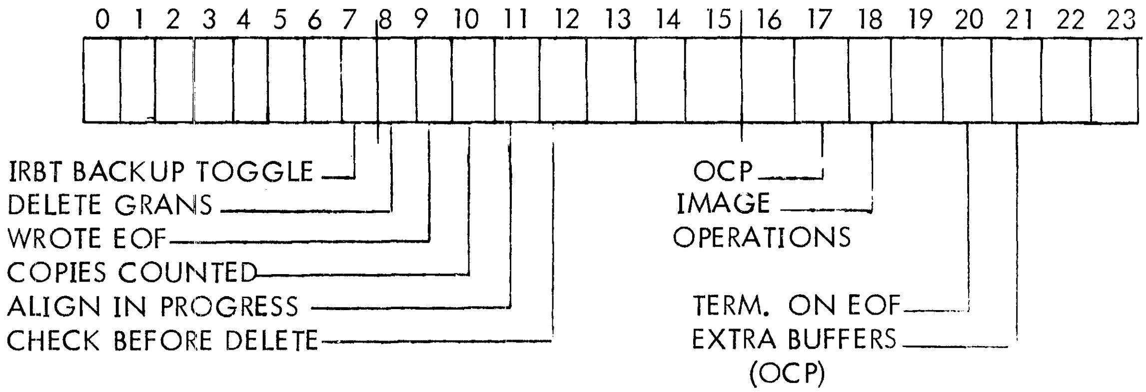
WORD	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
0	0			STREAM NUMBER
1	[illegible]	[illegible]	[illegible]	STREAM TYPE
2				
3		BLINK DISC ADDRESS		
4		FILE STARTING DISC ADDRESS		
5	READ OR WRITE DISC FUNC. CODE	I/O PRIORITY	NRT	0---------------------0
6	0	CORE BLOCK ADDRESS		
7	FCN ≠ IO IN PROGRESS		BLOCK BYTE COUNT	
8		FILE CURRENT DISC ADDRESS		
9	0	0---------0	FILE SIDE END ACTION ADDRESS	
10				
11	CURRENT BLOCK POINTER BYTES			
12				
13				
14				
15		RBID		
16				MINIMUM RECORD SIZE OR LINES PER PAGE
17				MAXIMUM RECORD SIZE OR WIDTH
18	COUNT	SPACE	JDE	COPIES
19	SEQUENCE ID			
20				
21				BLOCKS PROCESSED THIS FILE
22				RECORDS PROCESSED THIS FILE
23				PAGES PROCESSED THIS FILE
24	FORM PROJECTOR CONTROL			
25	FORM NAME (OR ZERO FOR NONE)			
26	FUTURE FORM NAME (OR ZERO FOR NONE)			
27	RELEASE PREVIOUS DISC ADDRESS			
28				CURRENT USER NUMBER
29	CURRENT LINE COUNT THIS PAGE			

COOP CONTEXT BLOCK

Word	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
0	0 SYMTABX	SYMBIONT BASE ADDRESS		
1		Flags (Bit description next page)		DEVICE TYPE
2		FLINK DISC ADDRESS		
3		BLINK DISC ADDRESS		
4	COPY COUNT	FILE OLDEST DISC ADDRESS		
5	READ DISC FUNCTION CODE	I/O PRIORITY	NRT	0--------------------0
6	0	CORE BLOCK ADDRESS		
7			BLOCK BYTE COUNT	
8		FILE CURRENT DISC ADDRESS		
9	0	FILE SIDE:END ACTION ADDRESS		
10	RECORD FUNCTION CODE	I/O PRIORITY	NRT	DEVICE DCTX
11	CURRENT BLOCK POINTER (BYTES)			
12	CURRENT RECORD BYTE COUNT			
13		SYMTABX OR RESTORE DISC ADDRESS		
14	0	DEVICE SIDE:END ACTION ADDRESS		
15	SYMBIONT PRIORITY	RBID	FILE SYSID	
16	VARIOUS TEMP. USES (SCTYC)			
17	I/O CALL ADDRESS			
18	LAST TOP OF FORM DISP.			
19	LAST TOP OF FORM DISK ADDRESS			
20	IRBT BACKUP DISP.			
21	IRBT BACKUP DISK ADDRESS			
22	IRBT TEMP. BACKUP DISP.			
23	IRBT TEMP. BACKUP DISK ADDRESS			
24	# OF GRANULES REMAINING IN FILE			
25	RETRY COUNTER FOR BLINK FAILURE			
26	KEYIN BACKUP DISK			
27	BACKUP LIST			
28	⋮			
39				

OUTPUT SYMBIONT CONTEXT BLOCK

Flags - Word 1 of OUTSYM context block



Symbiont/Cooperative Context Block Displacement Symbols

SCAIF	ADD input file flag Used by INSYM at Disc write end action to decide to read a card (SCAIF = 0) or ADD input file (SCAIF = 1).
SCBESTDA	Best Disk Address File starting disk address used by INSYM to AIF, SYMFILS to delete partial input file, COOP to AOF, SYMFILS to add output, OUTSYM to AOFP. When OUTSYM is deleting or outputing with catchup delete this is the oldest notdeleted granule.
SCBLDA	Backward Link Disc Address Contains SCCDA's predecessor except while updating disk addresses.
SCBSIZ	Block Size Used as the file side size argument to NEWQ and as the boundary during block construction.
SCBUDA	Backup Disk Address Is the Backup point for OUTSYM for the 'R' Keyin (and 'Q' in 'DELETE' mode).
SCBUPPT	Backup Point Used by OUTSYM when IRBT routines request symbiont to backup because of line block reconstruction. This 2 word area contains an appropriate SCDBI and SCCDA.
SCBUPPTS	Backup Point Save Used by OUTSYM to define point of last record output. When IRBT routines send a block down the line this 2 word area is copied over SCBUPPT.
SCCDA	Current Disk Address NEWQ argument defining source or destination of current block on file; used by SYMFILS in output closing.
SCCUN	Current User Number Used by COOP to pass to the end-action routine the number of the user whose I/O is completing so the event may be reported.

SCDALST	OUTSYM Backup Disk Address List Previously output disk addresses used by OUTSYM for the Syyndd, R Keyin.
SCDALSZ	Size of SCDALST Currently 14.
SCDBC	Device Byte Count Symbiont NEWQ argument for record transfer
SCDBI	Data Byte Index Byte displacement to the source or destination of the next record to be processed.
SCDCDA	Device CDA NEWQ argument slot used to pass symbiont SYMTABX to handler or HASPIO.
SCDEVTYP	Device type Byte 3 contains normal device type for operations on local peripherals but is only meaningful to HASPIO on remote operations. Bytes 0-2 are flags.
SCDINFO	Device Information Defines the base of the module specific information. Used by INSYM as the start of a 4-word save area.
SCDQARGS	Device IOQ Arguments Base of a 5-word NEWQ device calling sequence. Contains IO handler function code, IO request priority, number of retries and DCTX in bytes 0-3 respectively.
SCDQFC	Device IOQ Functions Codes Base of an 8-byte (2 word) translation table used to convert generalized system function codes into specific handler function codes.
SCFBUF	File Buffer Contains either a byte address or word address, physical or virtual (as appropriate and convenient to the routine using the context block) which indicates the blocking buffer currently being used.
SCFCO	File Count Used by OUTSYM as a count of files processed since initiation.

SCFFORM	Future Form The name of the FORM to be used when the user issues an M:DEVICE (FORM) CAL without a form name.
SCFINFLG	!FIN Flag The high order bit is used by INSYM to remember to communicate to RBBAT that the current file was terminated by a !FIN control card.
SCFLDA	Forward Link DISC Address Used in symbiont file linkage handling to hold the DISC address of the next block.
SCFORM	FORM Name Used by COOP to contain the name of the paper or card stock to be associated with an output file or to identify the desired non-control input file.
SCFPC	FORM Projector Control Used by COOP to contain the name of the overlay to be used on SHAMROCK FORM projector for the output file
SCFQARGS	File Queue Arguments Base of a 5-word NEWQ calling sequence. Contains handler function code, IO request priority, number of retries in bytes 0, 1, and 2, respectively.
SCGCO	Granule Count Actually the count of the number of symbiont blocks in the file which is two times the number of granules.
SCJCO	JOB Count Used by INSYM to count the number JOB cards encountered since initiation. Also used to recognize and treat properly the case of a lone !IFIN card.
SCLINES	Line Count This Page Used to recognize listing page boundaries for header processing in output file creation.
SCMAXR	Maximum Record Size COOP truncates output records to this size while blocking. Also used in the blocking buffer full test.

SCMINR	Minimum Record Size Used in output file creation to pad short records to an acceptable width. On a listing type device, this cell contains the number of lines per page.
SCMISC	Miscellaneous Contains miscellaneous output file attributes such as column of stream heading page count, default line spacing, job descriptor entry and number of copies.
SCMODE	Blocking Mode Used by INSYM to control file terminations. SCMODE = 1 implies !JOB or !IFIN terminates this file, SCMODE = 0 implies !JOB or !IFIN or !RB terminates. Thus, SCMODE = 1 while blocking RB CCs into a file.
SCNCBT	No Communication Buffer Temp. Defines a 4-word area to hold RBBAT communication arguments while a symbiont is waiting in the activate queue for an RBBAT communication buffer.
SCPCO	Page Count Contains number of print pages processed for the current file.
SCRCO	Record Count Contains the number of records processed for the current file.
SCRCVLST	Recovery List A pointer to and control word for the list of prior disc addresses not released in catchup deletion mode in OUTSYM.
SCRPDA	Release Previous Disc Addresses If nonzero, contains a disc address not released by cooperative end-action because ALLOCATs core stacks were full.
SCSEQ	Sequence ID Contains the sequence ID to be used in stream card sequencing.
SCSVDGI	Saved Ghost Information Contains the file priority, RBID and SYSID of the current file to be used to ADD partial output in the event of a 'Q' signal or system crash during OUTSYM processing.

SCSYMX	SYMX Table Contents Contains for INSYM the contents of the appropriate entry in the SYMTAB table SYMX.
SCTOFDA	Top of Form Disc Address Disc address of the granule containing the record belonging on the tope of the current form.
SCTYC	Type of IO Completion Temporary holding place for TYC at OUTSYM end-action for later examinations.
SCXTRAB	Extra Buffer List A pointer to and control word for the list of buffers used by OUTSYM in multi-core buffer operation.

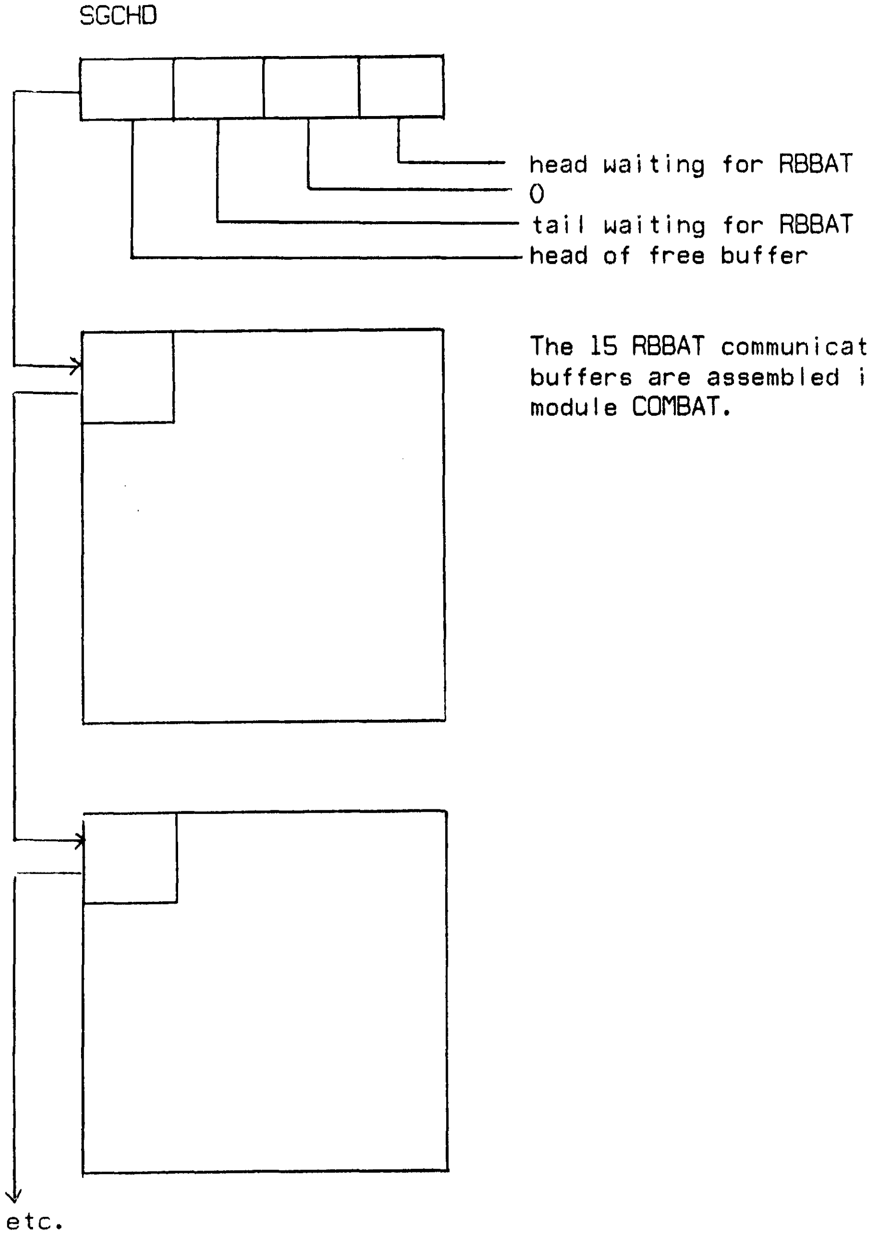
RBBAT COMMUNICATION BUFFERS

In the following descriptions, the numbers in the diagram below are used to describe the fields in a particular RBBAT communication buffer.

LINK	2	3	GFC
5	6	7	8
9	10	11	12
13	14	15	16

Byte 1 is always the link to the next buffer. The buffers are linked to SGCHD as in the diagram below.

Byte 4 is always the Ghost Function Code (GFC) which tells RBBAT what operation to perform.



The 15 RBBAT communication buffers are assembled in the module COMBAT.

GFC: NOPGFC

OPERATION: None - release buffer only

CALLED BY: Those who use two (2) buffers

NOTE: See the two-buffer-calls below.

GFC: AIF

OPERATION: Add Symbiont Control Input File

CALLED BY: INSYM

Byte 2: SYMBX
Byte 3: DCTX
Bytes 5 - 8: Starting disk address
Bytes 15 - 16: Number of granules

GFC: AIFJE

OPERATION: Add JOBENT Control Input File

CALLED BY: T:JOBENT

Bytes 5 - 8: Starting disc address
Bytes 9 - 12: SYSID

GFC: AIFNC

OPERATION: Add Symbiont Non-control Input File

CALLED BY: INSYM

Bytes 2 - 3: SYMBX,DCTX
Bytes 5 - 8: Starting disc address
Bytes 9 - 12: Name from ::NCTL
Bytes 13 - 16: Number of granules

GFC: AOF, AOFL, AOFNB

OPERATION: Add Output File (Last, Non-Batch)

CALLED BY: SUPCLS, T:JOBENT

Byte 2: Link to second buffer if any
Byte 3: Device type
Byte 5: Number of copies
Bytes 6 - 8: Starting disc address
Byte 9: Priority
Byte 10: RBID
Bytes 11 - 12: SYSID
Bytes 13 - 16: Number of granules

SECOND BUFFER

Byte 3: JDE
Bytes 5 - 8: Form name
Bytes 9 - 12: Forms overlay name

GFC: AOFP

OPERATION: Add Partial Output File

CALLED BY: OUTSYM

Byte 2: SYMBX
Byte 3: DCTX
Byte 5: Number of copies
Bytes 6 - 8: Starting disc address
Byte 9: RBSWITCHED, HASP, PARTIAL, Priority (1,1,1,5)
Byte 10: RBID
Bytes 11 - 12: SYSID
Bytes 13 - 16: Number of granules

GFC: MBSGFC

OPERATION: Perform Multi-Batch Schedule

CALLED BY: CLOCKI

GFC:	AOF, AOFL, AOFNB
OPERATION:	Add Output File (Last, Non-Batch)
CALLED BY:	SUPCLS, T:JOBENT
Byte 2:	Link to second buffer if any
Byte 3:	Device type
Byte 5:	Number of copies or 1st byte of COMID (see SECOND BUFFER)
Bytes 6 - 8:	Starting disc address
Byte 9:	Priority
Byte 10:	RBID
Bytes 11 - 12:	SYSID
Bytes 13 - 16:	Number of granules

SECOND BUFFER

Byte 3:	2nd byte of COMID
Byte 3 (bits 1 - 7):	JDE, bit 0 of byte 3: if set, byte 5 of 1st buffer is 1st byte of COMID
Bytes 5 - 8:	Form name
Bytes 9 - 12:	Forms overlay name
Note:	If a COMID is being passed and bit 0 of the first byte of the COMID is set, this indicates the last "chunk" of concurrent mode output.

GFC:	AOFP
OPERATION:	Add Partial Output File
CALLED BY:	OUTSYM
Byte 2:	SYMBX
Byte 3:	DCTX
Byte 5:	Number of copies
Bytes 6 - 8:	Starting disc address
Byte 9:	RBSWITCHED, HASP, PARTIAL, Priority (1,1,1,5)
Byte 10:	RBID
Bytes 11 - 12:	SYSID
Bytes 13 - 16:	Number of granules

GFC:	MBSGFC
OPERATION:	Perform Multi-Batch Schedule
CALLED BY:	CLOCK1

GFC: GOF

OPERATION: Get Output File

CALLED BY: OUTSYM

Byte 2: SYMBX
Byte 3: DCTX
Bytes 5 - 8: -1

ON RETURN

Byte 5: Number of copies
Bytes 6 - 8: Starting disc address (0 = no file)
Byte 9: RBSWITCHED, HASP, ZERO, Priority (1,1,1,5)
Byte 10: RBID
Bytes 11 - 12: SYSID
Bytes 13 - 16: Number of granules

GFC: GIFNC

OPERATION: Get Non-Control Input File

CALLED BY: OPNLD

Byte 3: Device type
Byte 5: User number
Byte 8: RBID
Bytes 9 - 12: Name

ON RETURN

Bytes 5 - 8: Starting disc address (0 = no file found)

GFC: PRIOGFC

OPERATION: PRIO Keyin

CALLED BY: KEYIN

Byte 5: Device type (0 = all)
Bytes 7 - 8: SYSID
Bytes 9 - 12: New PRIO

GFC: KDELGFC

OPERATION: DELE Keyin

CALLED BY: KEYIN

Byte 5: Device type (0 = all)
Bytes 7 - 8: SYSID

GFC: KFRMGFC

OPERATION: (SYYndd,F'XXXX') Type Form Keyin

CALLED BY: KEYIN

Byte 2: SYMBX of SYYndd
Byte 3: JDE (-1 not changed)
Bytes 5 - 8: Form name (-1 not changed)
Bytes 9 - 12: Forms overlay name (-1 not changed)

GFC: KFCGFC

OPERATION: (Form 3B) Type Form Keyin

CALLED BY: KEYIN

Byte 5: Device type
Bytes 7 - 8: SYSID
Bytes 9 - 12: New form name

GFC: KDISPGFC

OPERATION: Display Keyins

CALLED BY: KEYIN

Bytes 5 - 8: A 0 = no option or OC
1 = SYSID
-1 = NORUN
-2 = SYMB
-3 = output
Bytes 9 - 12: A=1 SYSID
A=0 =0 on LP
≠0 on OC

GFC: KCOMGFC, SNDGFC, BCSTGFC

OPERATION: RBCOM, RBSEND, RBBCST Keyins

CALLED BY: KEYIN

Byte 3: DCTX to send to
Bytes 5 - 8: MPOOL word address of TEXTC message

GFC: JESTAT

OPERATION: JOBENT Status Request

CALLED BY: T:JOBENT

Byte 5: Requesting user number
Bytes 7 - 8: SYSID for status

ON RETURN

Bytes 5 - 8:
1 = running
2 = waiting to run
4 = waiting to output
0 = completed
Bytes 9 - 12: number waiting to run if bytes 5 through 8 = 2

GFC: SWITGFC

OPERATION: RBSWITCH Keyin

CALLED BY: KEYIN

Byte 2: Link to second buffer
Byte 3: Byte count of WSN
Bytes 7 - 8: SYSID
Bytes 9 - 12: Device type (TEXT)

SECOND BUFFER

Bytes 5 - 8: 'TO' workstation name, word 1
Bytes 9 - 12: WSN, word 2

GFC:	JEDEL
OPERATION:	Cancel Command
CALLED BY:	T:JOBENT
Byte 2:	Link to second buffer
Byte 4:	Requesting user number
Bytes 7 - 8:	SYSID to delete

SECOND BUFFER

Bytes 5 - 8:	Word 1 of account
Bytes 9 - 12:	Word 2 of account

ON RETURN (BUFF1)

Bytes 5 - 8:	X'3A' no such file X'39' wrong account 0 AOK

GFC:	DUP, HUP
OPERATION:	RB Dial Up, RB Hang Up
CALLED BY:	BSCIO, RBSSS, DSCIO
Byte 3:	DCTX

GFC:	LORR
OPERATION:	RB Logon Record Received
CALLED BY:	BSCIO, DSCIO, RBSSS
Byte 3:	DCTX
Bytes 9 - 12:	MPOOL word address of logon record

GFC:	OCMGFC
OPERATION:	IRBT OC Input Message
CALLED BY:	HASPIO
Byte 3:	DCTX
Bytes 9 - 12:	MPOOL word address of message

GFC: RCVR1GFC

OPERATION: Marks Point of Recovery in COMBUFS

CALLED BY: RECOVER

GFC: RCVRGFC

OPERATION: Perform RBBAT Recovery

CALLED BY: RECOVER

Bytes 5 - 8: Size of dynamic data (-1 says recovery doesn't know)
Bytes 9 S:CUN at time of crash
Bytes 10 - 12: Flags
- 1 = operator recovery
- 2 = data zapped
- 4 = COMBUFS zapped
- 8 = RBBAT wasn't running

GFC: KOSTOP

OPERATION: OUTPUT STOP keyin

CALLED BY: KEYIN

Byte 2: SYMBX

GFC: KFLUSH

OPERATION: FLUSH keyin

CALLED BY: KEYIN

Byte 2: SYMBX
Byte 3: Device Type
Bytes 7 - 8: SYSID

MBS TABLES IN RBBAT: PRIORITY TABLE

		Table Name	
Name	Index	BH:HPRI	BH:TPRI
OIPRI	0	Head of PRIO 0 Input	Tail of PRIO 0 Input
		⋮	⋮
FIPRI	F	Head of PRIO F Input	Tail of PRIO F Input
RUNPRI	10	Head of Running Jobs	Tail of Running Jobs
OOPRI	11	Head of PRIO 0 Output	Tail of PRIO 0 Output
		⋮	⋮
FOPRI	20	Head of PRIO F Output	Tail of PRIO F Output
MFPRI	21	Head of Message File Output	Tail of Message File Output
NCIPRI	22	Head of Non-Control Input	Tail of Non-Control Input
DELPRI	23	Head of Files to be Deleted	Tail of Files to be Deleted
FREI	24	Head of Free Input Slots	Tail of Free Input Slots
FREO	25	Head of Free Output Slots	Tail of Free Output Slots

- Both are halfword tables
- Displacements are hexidecimal

SYMBIONT FILE TABLES IN RBBAT

	INPUT →	← OUTPUT →	
BH:GRAN		# of GRANULES	
BB:COPY		COPY COUNT	BYTE
BH:COMID	Bit 0 set for last chunk of concurrent output →	CONCUR. OUTPUT MODE ID	HALF WORD
BW:FORM		FORM NAME TEXT	WORD
BH:XLNK	HEAD OF EXCLUSIVE CHAIN IN S#H:LNK		HW
BH:SLNK	HEAD OF SHARED CHAIN IN S#H:LNK		HW
BH:PART	BITS IMPLY CANDIDATE PARTITION		HW
B4W:RES	* RESOURCE VALUES -Ordered like SH:RNM -X'80' implies default		Four Words
BH:TIME	JOB RUN-TIME MINUTES		HW
BD:ACCT	JOB ACCOUNT TEXT		DW
BB:RID	JOB OR OUTPUT RBID (0 IMPLIES LOCAL)		
BB:PI/BB:DEV	PRIORITY INCREMENT	DEVICE TYPE	
BW:SDA	FLAGS	FILE STARTING DISK ADDRESS	
BH:SID	FILE SYSID		
BH:LINK	LINK BY INPUT/OUTPUT PRIORITY (0 IMPLIES TAIL)		
index	0 1	INFILE+1	

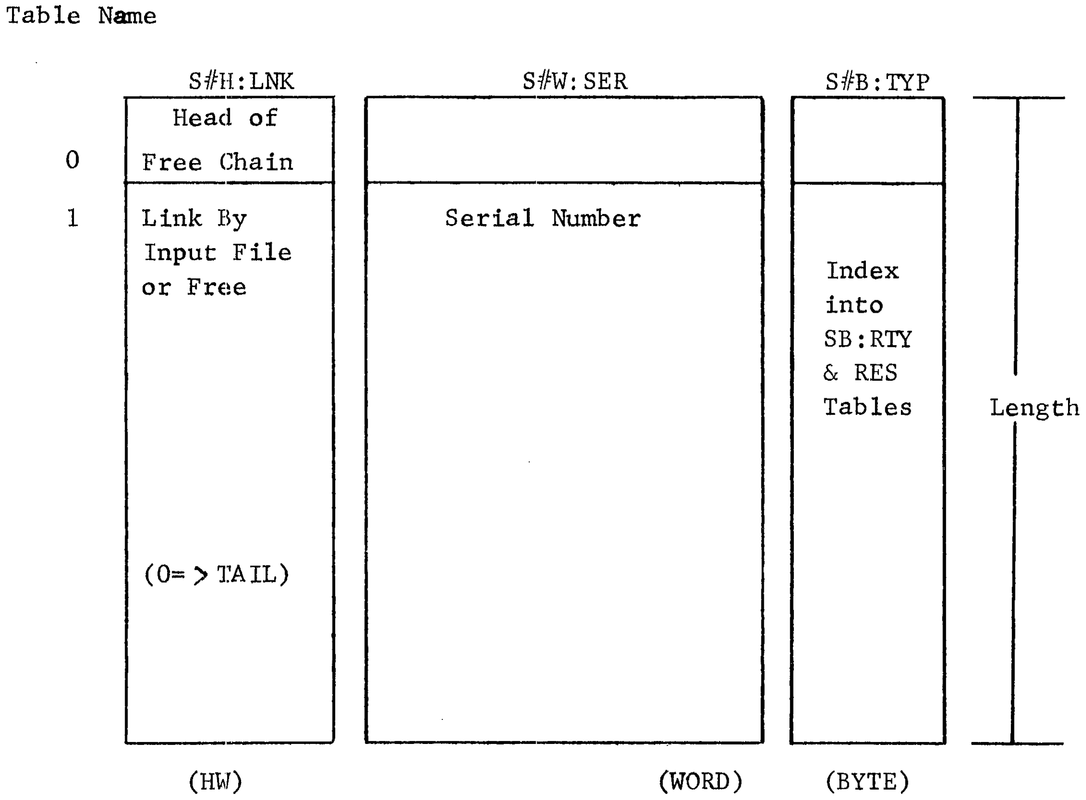
zeroeth entry net used

* job priority

FLAGS

Bit	Out	In
0	----	Order
1	----	Account
2	Delete output	When Added
3	Hold	Hold
4	Using Special Message File	----
5	----	Rerun request
6	HASP & Switched	Second rerun
7	RBSWITCHED	JOBENT sumbitted

MBS TABLES IN RBBAT: SERIAL NUMBER TABLES

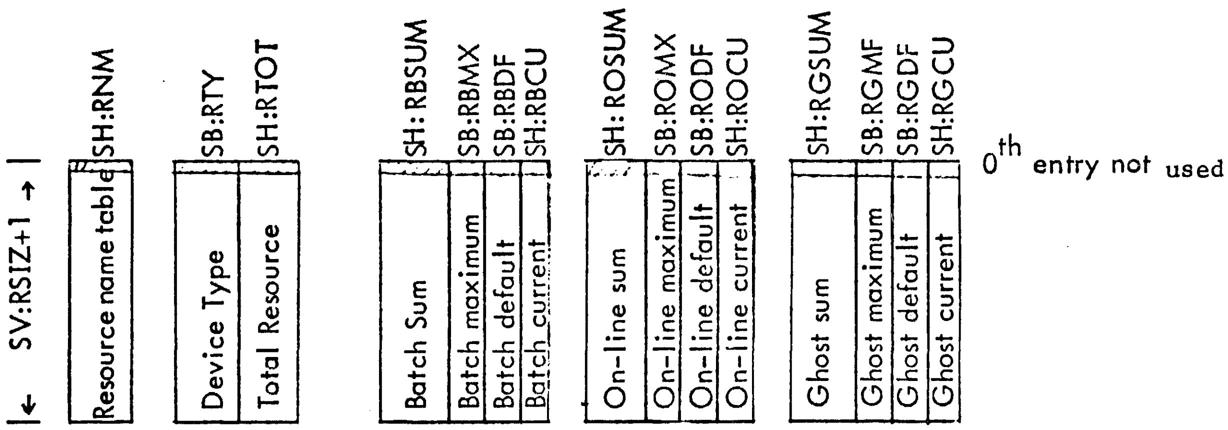


-Heads of serial number chains are in BH:SLNK or BH:XLNK.

-Length = ((INFILE*AVGSER+3)/4)+1

MBS TABLES IN CORE:

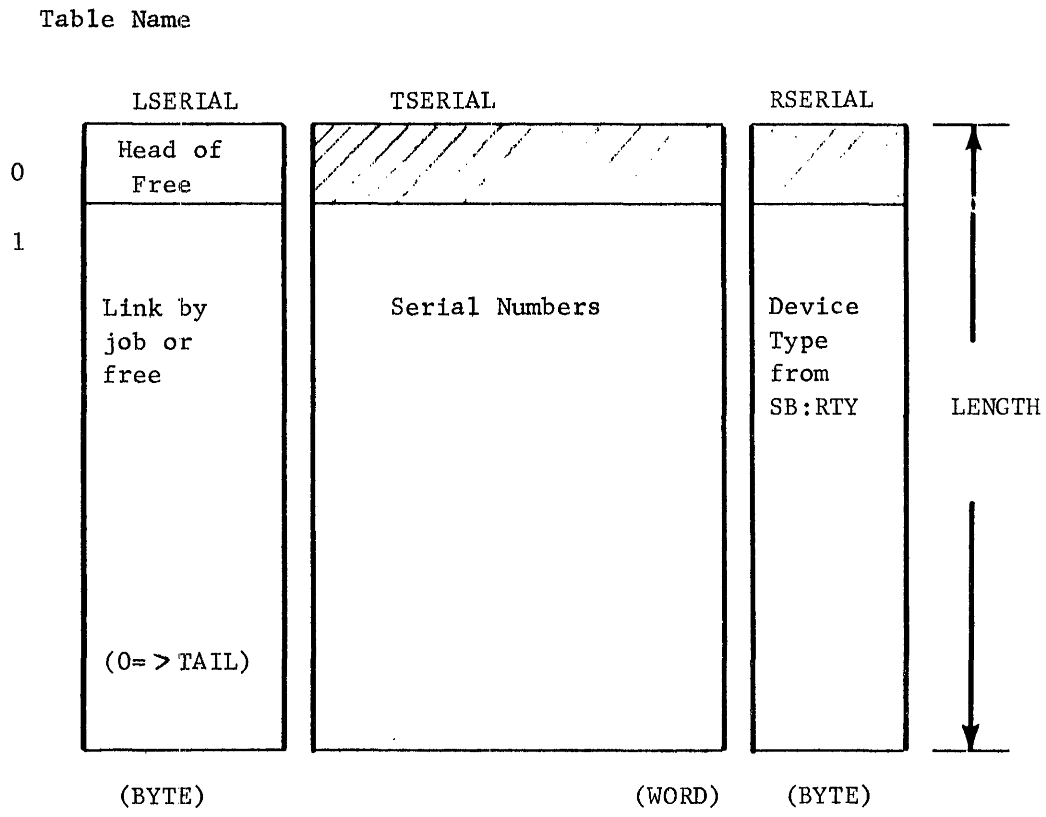
Resource Allocation Tables (RAT)



SH:RNM	Resource namet table in TEXT (halfword)
SB:RTY	Device type (Byte) (entries are unique)
SH:RTOT	Total system resource
SH:R$\left\{\begin{matrix}B\\O\\G\end{matrix}\right\}$SUM	Total resources available for the $\left\{\begin{matrix}\text{batch}\\\text{on-line}\\\text{ghost}\end{matrix}\right\}$ load
SB:R$\left\{\begin{matrix}B\\O\\G\end{matrix}\right\}$MX	Maximum allocation per $\left\{\begin{matrix}\text{batch}\\\text{on-line}\\\text{ghost}\end{matrix}\right\}$ user
SB:R$\left\{\begin{matrix}B\\O\\G\end{matrix}\right\}$DF	Default allocation per $\left\{\begin{matrix}\text{batch}\\\text{on-line}\\\text{ghost}\end{matrix}\right\}$ user
SH:R$\left\{\begin{matrix}B\\O\\G\end{matrix}\right\}$CU	Currently allocated for the $\left\{\begin{matrix}\text{batch}\\\text{on-line}\\\text{ghost}\end{matrix}\right\}$ load

Note: SH:RNM and the value definition SV:RSIZ $\leq$ 15 are in the SYSGEN produced load module SG:RNT. (The value tables are monitor resident, having been initially incorporated into the monitor from the SYSGEN module SG:RCT.) The first four entries are CO, 9T, 7T, and SP.

MBS TABLES IN CORE: SERIAL NUMBER TABLES FOR RUNNING JOBS



-Heads of serial number chains are in users JIT.
-Length = 255 or (((16*AVGSER)+3)/4)+1+16
(whichever is less)

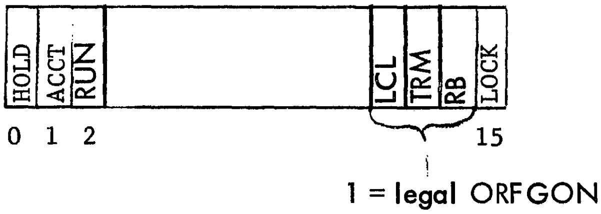
MBS TABLES IN CORE: PARTITION TABLES

SYSTEM NAME	DEFINITION	ENTRY SIZE
PL:LK	Partition table lock control word set if tables currently being altered or used.	1 word
PL:CHG	Control - altered partition status word which flags an occurring change in partition tables and signals which partitions definition has been altered.	1 word

Each of the tables below is of length LPART (number of partitions defined).

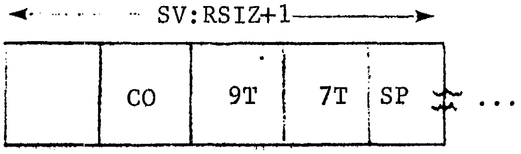
PLH:TL	Lower time limit of partition n.	halfword
PLH:TU	Upper time limit of partition n.	halfword
PLH:QN	Quantum time of partition n.	halfword
PLD:ACT	Current running account of partition n.	doubleword
PLH:CUR	Current number of jobs which have been selected under current partition definition.	halfword
PLH:TOL	Total number of jobs run under this partition.	halfword
PLB:USR	User number of user currently using partition.	byte
PLH:SID	SYSID of job currently using partition.	halfword
PLH:FLG	Partition control flags.	halfword

PLH:FLG



The following two tables of length LPART are (number of defined resources) bytes wide.

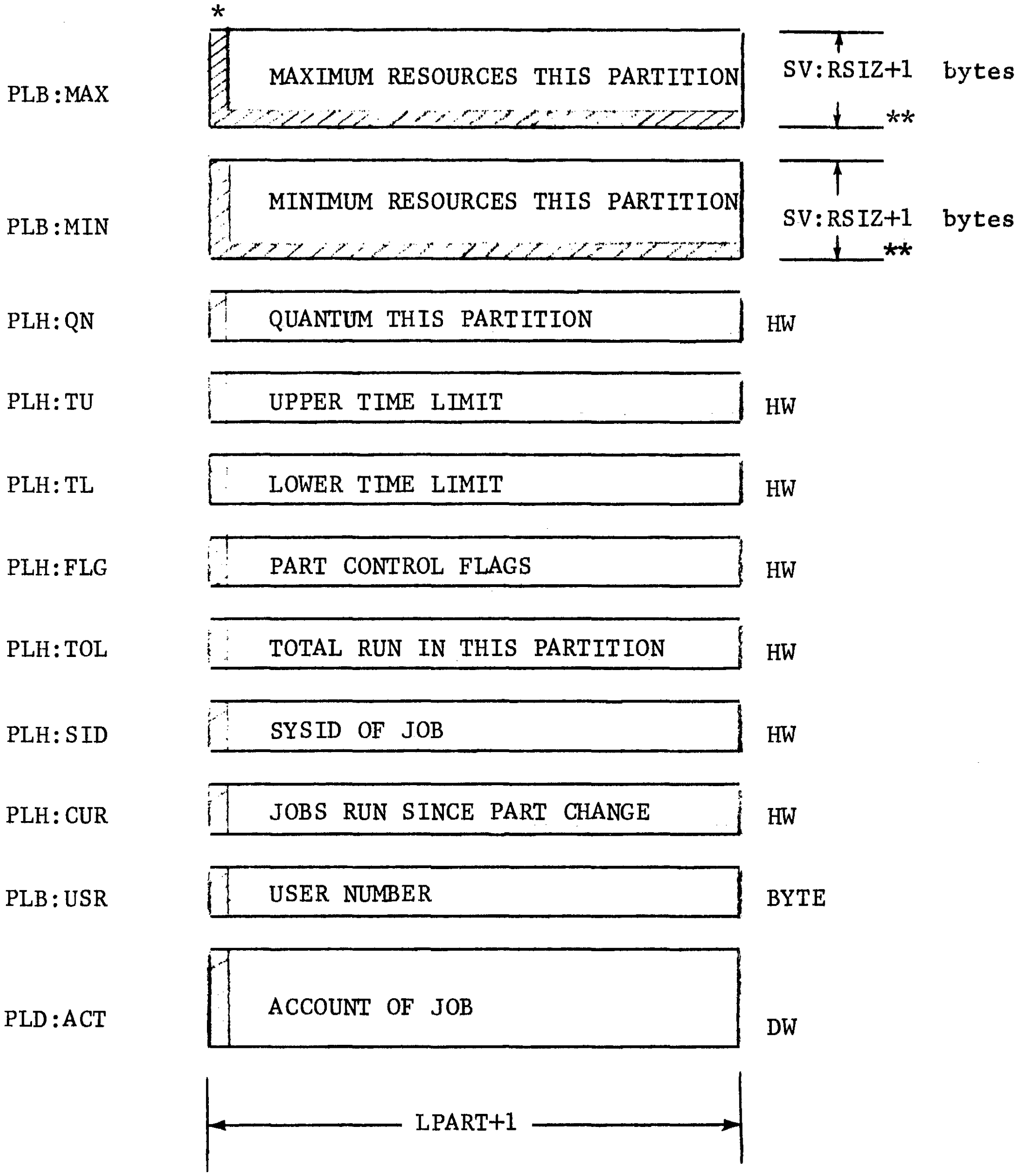
SYSTEM NAME	DEFINITION	ENTRY SIZE
PLB:MAX	Upper resource limits of partition n.	# resources byte
PLB:MIN	Lower resource limits of partition n.	# resources byte



Resource order corresponds to order in SH:RNM.

Byte 0 of PLB:MIN for a partition contains the head of the exclusive serial number chain into LSERIAL for the job running in that partition.

MBS TABLES IN CORE: PARTITION TABLE LAYOUT (PLD:ACT TO PLB:MAX)



* zeroeth entry not used.
** zeroeth sub-entry not used.

GI Tables

These tables (assembled into the module COMBAT) are the communication path from the Multibatch Scheduler to CCI via the input cooperative through the routine GETI. Each table has three entries allowing the MBS to supply three jobs in one scheduling pass by finding three jobs and filling in the information from its internal tables (the Batch Tables) into the GI tables.

GETI, when called in response to a CCI read, finds the proper GI entry by matching the current user number, S:CUN, against the table GIB:UN. Information from other GI tables is moved to JIT as indicated in the table below, the file indicated in GI:SDA is read, and the GI table entry is freed by placing its entry number on the free list GI:FRE.

GIB:UN - User number

GIH:TIM - maximum run time minutes (stored in J:MRT

GI:ASPN - 3 words containing the shareable Drives/Spindles Bit Map (stored in J:ASPIN)

GI:RES - 4 word list of required resources (store in JB:MAX);paralles RAT.

GIB:SLN, GIB:XLN - shared and exclusive serial number table linkages (stored in JB:SLNK, JB:XLNK)

GIB:RID - remote identification (stored in JRBID)

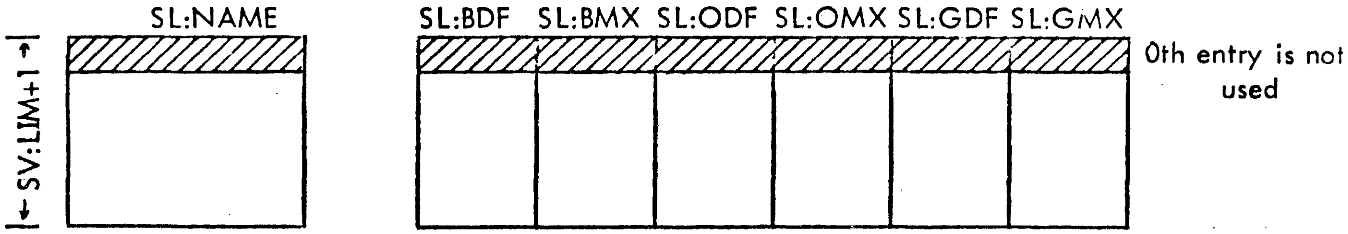
GI:SDA - starting disk address, input file

GIB:PRT - partition number (stored in JB:PNR; SYSID from PLH:SID store in J:JIT)

GI:FRE - A word which lists the free GI entries.

GIB:PRI - PRIO from job card or PRIO keyin; stored in JIT+PRT

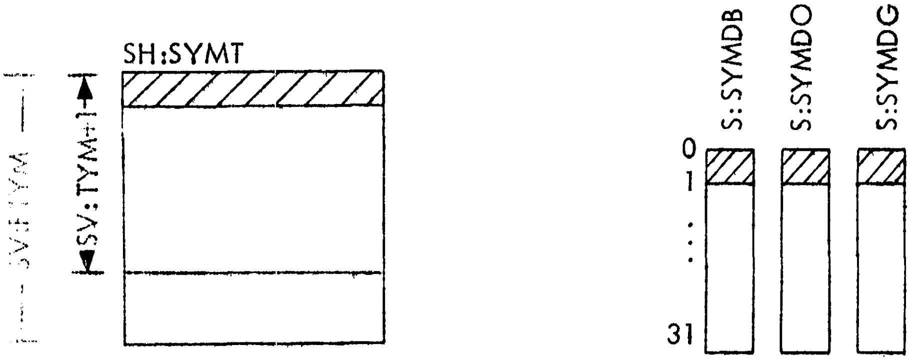
Service Limit Tables (SLT)



SL:NAME	Word table containing the first four characters of the limit name (left justified blank filled).
$SL:\left\{\begin{matrix}B\\O\\G\end{matrix}\right\}DF$	Word tables containing the default limit values for any $\left\{\begin{matrix}\text{batch}\\\text{on-line}\\\text{ghost}\end{matrix}\right\}$ job.
$SL:\left\{\begin{matrix}B\\O\\G\end{matrix}\right\}MX$	Word tables containing the maximum limit values for any $\left\{\begin{matrix}\text{batch}\\\text{on-line}\\\text{ghost}\end{matrix}\right\}$ job.
SV:LIM	Value definition defining the number of entries in the SL:NAME table.

SL:NAME and the value definition SV:LIM are in the SYSGEN produced load module SG:LNT. A value definition is produced for each limit name. The value definition is made up of SV: and the first two characters of the limit name. These value definitions are also included in the SYSGEN module SG:LNT. The value tables are monitor resident, having been initially incorporated into the monitor from the SYSGEN module SG:DLNT.

Peripheral Authorization Tables (PAT)



SH:SYMT	A table in TEXT containing those device types to be associated with symbionts. The table consists of half-word entries in TEXT format left justified and blank filled. The bottom of the table contains FAUTH values as above.
S:SYMD $\begin{Bmatrix}B\\O\\G\end{Bmatrix}$	Words containing Bit tables which are the default authorization flags. Bit 1 corresponds to the 1st entry.
SV:TYM	Is a value defining the table length, for SYMBIONT DEVICES.
SV:FTYM	Is a value defining the table length including FAUTHs, 15 max.

Note; SH:SYMT and the value definition SV:TYM and SV:FTYM are in the SYSGEN produced load module SG:PNT. S:SYMD $\begin{Bmatrix}B\\O\\G\end{Bmatrix}$ are core resident.

ERROR LOG FORMAT

The Error Log File is a special consecutive file on the RAD written by ERRLOG. The file is not accessed by name, but by a special pointer in memory which contains the disk address of the first record in the file. Each record is 64 words long and contains a backward and forward link. The backward link is the disk address of the previous record and the forward link is the disk address of the next record in the file. The backward link is never zero since the file is considered as open and the next record to be written is being constructed in a core buffer as the errors occur. The forward link in the last record written contains the disk address where the next record is to be written.

The record format in the file is as follows:

0	Backward Link
1	Forward Link
2	Number of Words in this Buffer
3	Message 1
	Message 2
	. .
	Message N
63	

Each record may contain several unused words at the end of the record since messages may vary in size, from one to eighteen words each. Therefore, up to nine words may be unused. The maximum number of useful words in a record is 61. The format of the backward and forward link disk address is as follows:

0 15	16 31
DCT Index for Disk	Relative Sector No.

Disk granules for writing the file are obtained by calling the GBG (Get Background Granule) routine. This routine gets one granule at a time. Each granule contains one record of the file.

FILE CONTROL POINTERS

Several pointers in memory provide the necessary control information for constructing and accessing the file. The following is a list of pointers, all of word length, and their definitions:

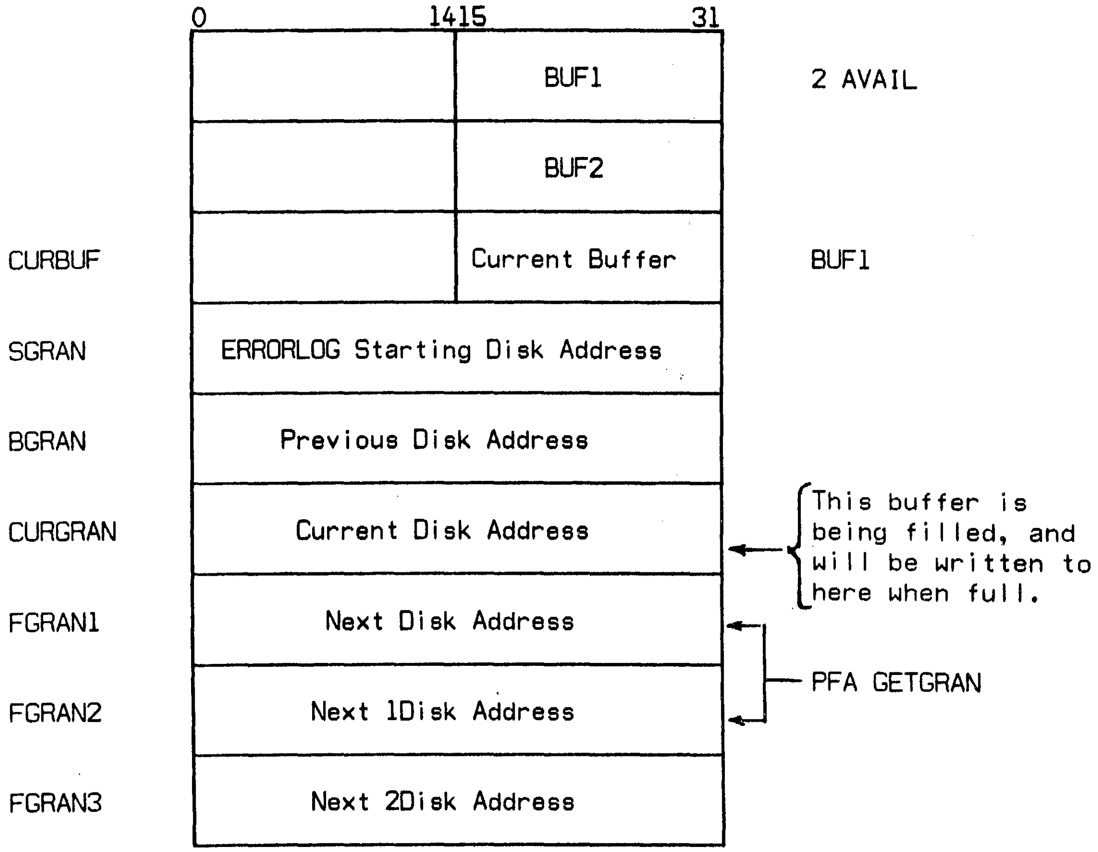
SGRAN	contains the disk address of the first record in the file.
BGRAN	contains the disk address of the last record written.
CURGRAN	contains the disk address of the next record to be written.
FGRAN1	contains the disk address of the record to be written following the current record.
CURBUF	contains address of buffer which is currently being used to pack the messages.

Two 64-word buffers are used for packing the messages. The labels for the two buffers are BUF1 and BUF2. Each buffer is preceded by two control words. The first control word contains the memory address of where the next message will be stored in the buffer. The second control word contains the number of words of space remaining in the buffer. The first three words of the buffer are the same as the first three words of the record on disk. The backward and forward link addresses are put in the buffer before any messages are packed in the buffer.

If one of the buffers is in the process of being written to the disk and tha second buffer fills up before the writing of the first is completed, then all error messages for the errors which occur while this condition exists will be lost. The error logging routine just exits when there is no space available in either of the buffers.

Error Log Core Structure

ERRLGTAB



This buffer is being filled, and will be written to here when full.

	0 31
2	WA (Next Entry)
-1	Number Words Available
0	Back-link Disk Address
1	Forward-link Disk Address
2	Number Words Used (≤61)
255	61 Available (Maximum)

OVERVIEW

The system error log is a standard file of the operating system in question. It may be organized in any access method convenient to the individual operating system but must be accessible sequentially. The name of the file is to be ERRFILE and the account shall be :SYS. Logical records within the file are variable length but may be carried in a fixed length record if convenient. Logical records will never be longer than 20 words.

Each logical record of ERRFILE begins with a byte containing the error log entry code and a byte containing the total number of words in the record.

If a keyed file organization is used to facilitate analysis, the keys will be formatted as follows:

The key contains the Julian date in packed decimal, the time of the error in EBCDIC, and a sequence number for errors with the same time. This sequence number is reset to zero for each entry with a new time. The format of the key is

08	yy	od	dd
h	h	m	m
n			

where:

08	is the number of bytes in the key.
yyoddd	is the Julian date in packed decimal.
hhmm	is the time in hours and minutes (EBCDIC).
n	is the sequence number.

CP-V Error Log Mechanism

In CP-V, the mechansim for creating the error log utilizes a monitor resident recording routine ERRLOG which blocks individual error entries into 64-word blocks using a standard double buffering technique. When full, these blocks may be written into a chain of sectors obtained from the permanent file area (PFA). Only the first 64 words of each 256-word sector is used in order to minimize the core requirement for buffering. Blocks are chained by links carried in the first words of each block and are not part of the standard file system. The core resident head of chain is carried across recoveries by the recovery routines.

Each time six errors are recorded by ERRLOG, the ghost job, ERR:FIL, is triggered. This program uses the privileged error log reading CAL to obtain the blocks of error log entries from the chained file. The read is destructive and secondary storage granules are returned to the system (PFA) as the information is read from them. ERR:FIL deblocks the individual entries, creates a key for the record indicating date and time of the entry, and writes the record into the file, ERRFILE, in account :SYS using the keyed file organization. Each time ERR:FIL is awakened, it continues copying until no more error log entries remain to be copied.

This process is outlined in Figure 1.

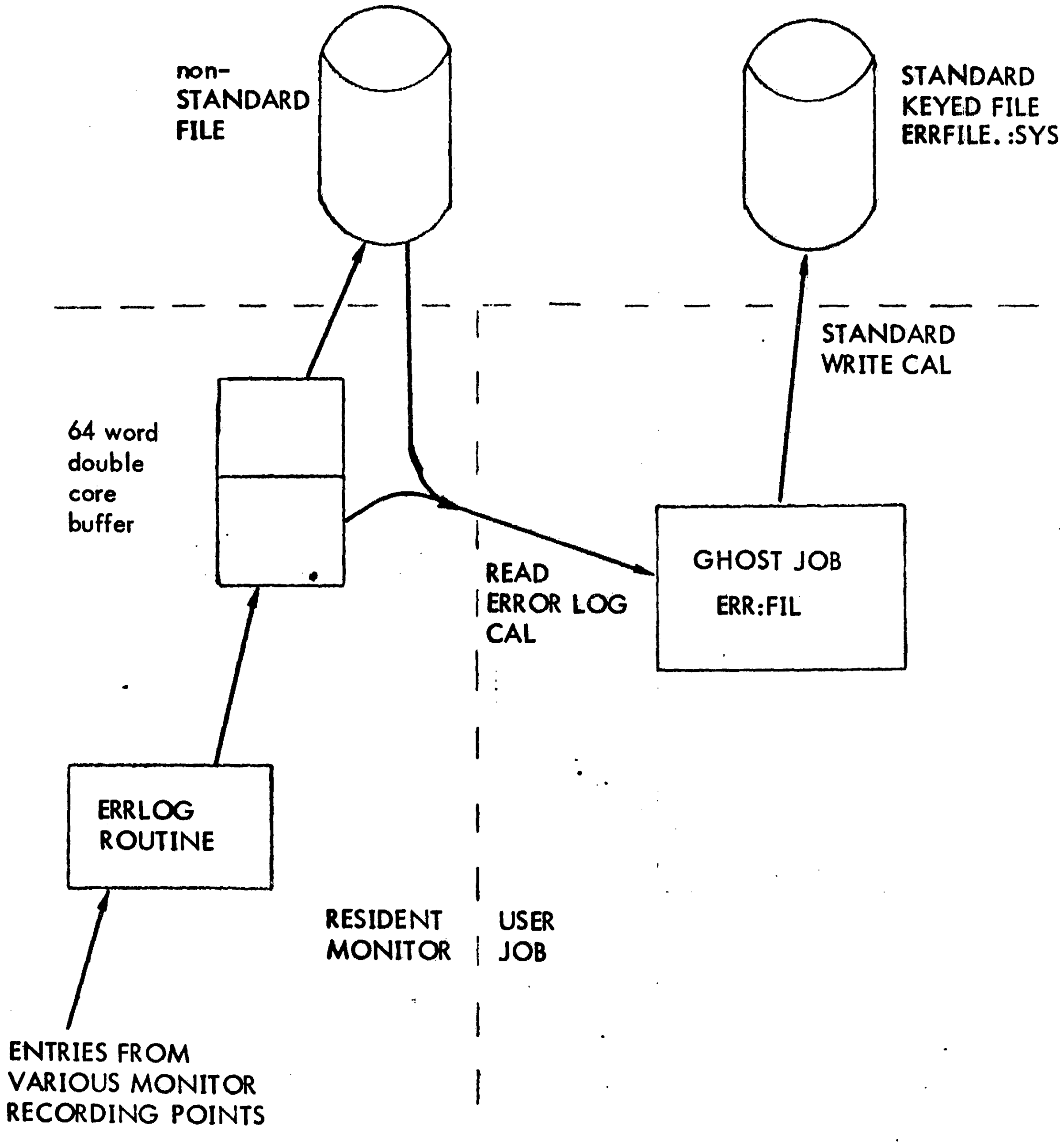


FIGURE 1 - CP-V ERROR LOGGING SCHEME

SUMMARY OF ERROR LOG ENTRIES

Error Codes

0	Null Entry
10	Copy Error (several subtypes)
11	SIO Failure
12	Device Time Out
13	Unexpected Interrupt
14	Reserved
15	Device Error
16	Secondary Device Error
2D	Hardware Error
18	System Startup/Recovery
2E	Watchdog Timer
1A	File Inconsistency Entry
1B	Software-detected Symbiont INconsistency
1C	Reserved
2F	Instruction Exception
1E	Lost Entries
1F	Duplicate Entries
20	Power On
21	Configuration
22	System Identification
23	Time Stamp
24	Bad Granule Release
25	Reserved
26	Remote Processing Error Record
27	Operator Message
28	I/O Activity Count
29-2F	Reserved
30	PFI Primary Record
31	MFI Primary Record
32	Secondary Record for Poll Information
33-40	Reserved
41	560 Processor Configuration
42	560 Memory Parity Secondary Record
43	Sigma 9 Memory Parity Secondary Record
44	Sigma 5 - 7 Memory Parity Secondary Record
45-4F	Reserved
50	ENQUEUE Table Overflow
51	Partitioned Resource
52	Returned Resource

Reserved for Future Enhancements

53 - 5F	For CP-V
60 - 6F	For CP-R
70 - BF	New Features

GLOSSARY OF TERMS

The following pages detail the terms used in the error log formats for each operating system.

CP-R GLOSSARY OF TERMS

AIO CC	A 4-bit value (bits 0-3 of designated byte) representing the condition codes as returned by the hardware in response to an AIO instruction.
AIO Status	A 16-bit value representing the status as returned by the hardware in response to an AIO instruction.
Alternate I/O Address	A 16-bit value representing an alternate physical I/O address by which a dual-access device can be referenced.
Bad Location (Sigma 7 only)	A 32-bit value representing the real memory address which caused a memory parity interrupt.
Code	An 8-bit value in the first byte of an error log message indicating message type.
Core Size (in 8K word blocks)	An 8-bit value representing the number of 8K (8192) blocks of words.
Count	An 8-bit value in the second byte of the error log message representing the number of useful 32-bit words contained in the error log record. Includes the first word in the count.
Current Command Doubleword	A 64-bit value representing the command doubleword currently being processed for a device (as indicated by the TDV status DW).
DCT Index	An 8-bit value indicating the order in which the device is configured into the system (at SYSGEN). The index value for the first device is 1.
Effective Address of Trapped Instruction	A 32-bit value representing the final address computed for the instruction pointed to by the instruction address (IA) in the PSD.

HIO CC	A 4-bit value (bits 0-3 of designated byte) representing the condition codes as returned by the hardware in response to an HIO instruction.
HIO Status	A 16-bit value representing the status as returned by the hardware in response to an HIO instruction.
I/O Address	A 16-bit value representing the physical I/O address.
I/O Count	A 32-bit value representing the number of start input/output (SIO) instructions executed for a device. This value is reset at system boot time.
Julian Day	A 16-bit value representing the julian day of year (e.g., March 1 would be represented as X'3D').
Memory Status Words (only Sigma 9 and 560 series)	Each word is a 32-bit value representing data returned by the hardware in response to a LMS instruction.
MFI (Sigma 6 or 7 only)	A 4-bit value representing the current state of the memory fault indicators returned by the hardware in response to a RD instruction. All memory fault indicators will be reset.
Model Number	A 16-bit value representing the conversion of a number (assigned by Field Engineering to uniquely identify peripheral devices) to a binary value (e.g., 7242 would be represented as X'1C4A').
Number of Parity Errors	A 16-bit value representing the number of bad locations causing memory parity errors (only the first 14 bad locations are entered in the log if the number of errors errors is greater than 14).
Poll CC	A 4-bit value (bits 0-3 of designated byte) representing the condition codes as returned by the hardware in response to a POLP or POLR instruction.
Poll Status (identical to POLR Results)	A 16-bit value representing the processor fault status as returned by the hardware in response to a POLP or POLR Instruction.

Primary I/O Address	A 16-bit value representing the physical I/O address by which a device can be referenced.
PSD	A 64-bit value representing the program status doubleword.
Real Address	A 32-bit value representing the actual memory address. (In an unmapped system, this is the same as the address in the IA field of the PSD.)
Real Time Resolution	An 8-bit value, n, such that actual relative time resolution = 2 milliseconds (e.g., N=1 for a resolution of 500HZ or 2 milliseconds).
Recovery Count	Currently not applicable. Will be 0.
Relative Time	A 32-bit value representing milliseconds since midnight. Resolution is 2 milliseconds.
Retries Remaining	An 8-bit value representing Retry Request minus the number of retries attempted. The range is between Retry Request and -1. A value of -1 indicates the operation was terminated due to retry count rundown.
Retry Request	An 8-bit value representing the maximum number of retries after which device error is returned to the requestor. This value is obtained from the requestor's DCB.
SIO CC	A 4-bit value (bits 0-3 of designated byte) representing the condition codes as returned by the hardware in response to a SIO instruction.
SIO Status	A 16-bit value representing the status as returned by the hardware in response to a SIO instruction.
Site Identification	A 64-bit field. The first 32 bits will contain the EBCDIC representation of the SYSGEN input parameter for version. The second 32 bits will be blanks (X'40').
Startup Type	An 8-bit field indicating type of system initialization. Will be 3 for System device boot.

TDV CC	A 4-bit value (bits 0-3 of designated byte) representing the condition codes as returned by the hardware in response to a TDV instruction.
TDV Status Doubleword	A 64-bit value representing the subchannel status as current command doubleword, device status, and byte count as returned by the hardware in response to a TDV instruction.
TIO CC	A 4-bit value (bits 0-3 of designated byte) representing the condition codes as returned by the hardware in response to a TIO instruction.
TIO Status	A 16-bit value representing the status as returned by the hardware in response to a TIO instruction.
Trap CC	A 4-bit value (bits 0-3 of designated byte) representing the condition codes as returned by the hardware when certain traps occur.
Trapped Instruction	A 32-bit value representing the contents of the location pointed to by the instruction address (IA) in the PSD.
Unit Address	A 6-bit value (bits 2-7 of designated byte) representing the address by which a processor can be referenced. The value is composed of a 3-bit cluster number followed by a 3-bit unit number.
Unit Type	An 8-bit value specifying the type of processor. Bit 0 of the designated byte indicates the presence of the processor in the current operational configuration (0 = present, 1 = not present).
Version	Not currently applicable. Will be 0.
Year	A 16-bit binary value representing the current year minus 1900 (e.g., 1973 will be represented as X'49').

CP-V GLOSSARY OF TERMS

Account	The doubleword used to identify a user's collection of files.
AIO CC	A 4-bit field (bits 0-3 of designated byte) representing the condition codes as returned by the hardware in response to an AIO instruction.
AIO Status	A 16-bit field representing the status as returned by the hardware in response to an AIO instruction.
Alternate I/O Address	A 16-bit value representing an alternate physical I/O address by which a dual-access device can be referenced.
Bytes Remaining	A 16-bit field representing the Remaining Byte Count (RBC) field as returned by the hardware in response to a TDV instruction.
Code	An 8-bit value in the first byte of the error log message indicating message type.
Consecutive, Keyed, Random	Methods of organizing user files in CP-V (refer to the CP-V Batch Reference Manual)
Core Size	An 8-bit value representing the number of 8K (8192) word blocks.
Count	An 8-bit value in the second byte of the error log message representing the number of useful 32-bit words contained in the error log record. Includes the first word in the count.
Count of Entries Identical to Previous Entry	The number of error log records which are identical to one previously logged for identical reasons (excludes time records).
Count of Entries Lost	The number of error log records lost when logging becomes temporarily impossible for any reason.
Current Command Doubleword	A 64-bit value representing the command doubleword currently being processed for a device (indicated by the TDV status DW).
Caller's Address	The address or EBCDIC name of routine back to which the error logging routine will return when logging is complete; used in isolating software faults.

DCT Index	The 8-bit value indicating the order in which the device is configured into the system (at SYSGEN).
DCT Index of Symbiont Device	The 8-bit value indicating the order in which the device associated with the symbiont is configured into the system (at SYSGEN).
Effective Address	A 32-bit value representing the final address computed for the instruction pointed to by the instruction address (IA) in the PSD.
Error Subcode	An 8-bit field indicating which of several types of file inconsistencies has occurred (see CP-V Batch Reference Manual).
File Name	The TEXTC name used to identify a collection of user data on secondary storage.
Granule	The unit of secondary storage allocation equal to 2048 bytes (usually 2 sectors).
Generalized Disk Address	See Section 6.0.
HIO CC	A 4-bit value (bits 0-3 of designated byte) representing the condition codes as returned by the hardware in response to an HIO instruction.
HIO Status	A 16-bit value representing the status as returned by the hardware in response to an HIO instruction.
I/O Address	A 16-bit value representing the physical I/O address.
I/O Count	A 32-bit value representing the number of SIO instructions executed for a device. This field is reset at system boot and recovery time.
Julian Day	A 16-bit value representing the julian day of the year (e.g., March 1 would be represented as X'3D') when the error was logged.

Memory Status Words (only Sigma 9 and 560 Series)	Each word is a 32-bit value representing data returned by the hardware in response to a LMS instruction.
MFI (Sigma 6 or 7 only)	A 4-bit value representing the current state of the memory fault indicators returned by the hardware in response to a RD instruction. All memory fault indicators will be reset.
Mode	A 16-bit value representing the manner in which the file was last referenced (See CP-V Reference Manual).
Model Number	A 16-bit value representing the conversion of a number (assigned by Field Engineering to uniquely identify peripheral devices) to a binary value (e.g., 7242 would be represented as X'7242').
Number of Parity Errors	A 16-bit value representing the number of bad locations causing memory parity errors (only the first 14 bad locations are entered in the log if the number of errors is greater than 14).
Poll CC	A 4-bit value (bits 0-3 of designated byte) representing the condition codes as returned by the hardware in response to a POLP or POLR instruction.
Poll Status (identical to POLR Results)	A 16-bit value representing the processor fault status as returned by the hardware in response to a POLP or POLR instruction (for 560).
Primary I/O Address	A 16-bit value representing the physical I/O address by which a device can be referenced (see Alternate I/O Address).
PSD	A 64-bit value representing the program status doubleword.
Real Address	A 32-bit value representing the actual memory address (in an unmapped system, this is the same as the address in the field of the PSD).
Recovery Count	An 8-bit value initialized to zero at system initialization and incremented by the value one for every system recovery.

Relative Time	A 32-bit value representing milleseconds since midnight. Resolution is 2 msec.
Relative Time Resolution	An 8-bit value, n, such that actual relative time resolution = 2 msec. (e.g., n=1 for a resolution of 500HZ or 2 msec.)
Retries Remaining	An 8-bit value representing Retry Request minus the number of entries attempted. The range is between Retry Request and -1. A value of -1 indicates the operation was terminated due to retry count rundown.
Retry Request	An 8-bit value representing the maximum number of retries after which device error is returned to the requestor. This value is obtained from the requestor's DCB.
Screech Code	The code used by CP-V to identify the system failure which has occurred.
Screech Subcode	An 8-bit field identifying which type of a specific and similar set of system failures has occurred (see CP-V Systems Management Reference Manual).
Seek Address	The physical disc address last used to access this device.
Sense Information	The diagnostic information returned from the device as a result of sending a "sense" order to the device.
SIO CC	A 4-bit value (bits 0-3 of designated (byte) representing the condition codes as returned by the hardware in response to a SIO instruction.
SIO Status	A 16-bit value representing the status as returned by the hardware in response to a SIO instruction.
Site Identification	A 64-bit field which contains the EBCDIC representation of the SYSGEN input parameter for site identification from the :MON command.
Startup Type	An 8-bit field indicating which of several types of system initialization was used.

Subchannel Status	The status of the I/O subchannel received from the hardware as a result of a TDV instruction.
Symbiont File	A CP-V system special file for buffering data between the CPU and slower speed line printers, card punchers, etc.
TDV CC	A 4-bit value (bits 0-3 of designated byte) representing the condition codes as returned by the hardware in response to a TDV instruction.
TDV Current Command DA	A 24-bit field representing the current command doubleword address used in obtaining the device status with a TDV instruction.
TDV Status Doubleword	A 64-bit field representing the subchannel status, as current command doubleword, device status, and byte count as returned by the hardware in response to a TDV instruction.
TIO CC	A 4-bit value (bits 0-3 of designated byte) representing the condition codes as returned by the hardware in response to a TIO instruction.
TIO Status	A 16-bit value representing the status as returned by the hardware in response to a TIO instruction.
Trap CC	A 4-bit value (bits 0-3 of designated byte) representing the condition codes as returned by the hardware when certain traps occur.
Trapped Instruction	A 32-bit value representing the contents of the location pointed to by the instruction address (IA) in the PSD.
Unit Address	A 6-bit value (bits 2-7 of designated byte) representing the address by which a processor can be referenced. The value is composed of a 3-bit cluster number followed by a 3-bit unit number.

Unit Type	An 8-bit value specifying the type of processor. Bit 0 of the designated byte indicates the presence of the processor in the current operational configuration (0 = present, 1 = not present).
User ID	A 16-bit value which is a unique number assigned by the system to the particular job or session.
User Number	An 8-bit value which is the index into internal system tables used to access user-specific information.
Version	The version identifier of the system running (i.e., A00, B00, etc.).
Volume Serial Number	A 4- or 6-byte field supplied by a user to identify either a tape or private pack.
Year	A 16-bit binary value representing the current year minus 1900 (e.g., 1973 will be represented as X'49').

ERROR LOG FORMATS

The following charts detail formats for the error log. Some of the entries will be followed by secondary entries.

All relative times are in milliseconds since midnight. The count is zeroed at midnight, initialized at system startup, and carried over recovery. An operator-initiated time change will also re-establish the correct count. For CP-V and CP-R, the units are 1 millisecond, however, the clock resolution reduces the relative time resolution to 2 milliseconds.

Hachured fields are unused and may have arbitrary contents.

All condition code fields are the result of use of the STCF instruction, that is, a byte with the condition codes in bits 0-3 and the floating controls in bits 5-7.

All error log entries have a code in byte 0 identifying the entry type, a word count in byte 1. The second word contains relative time. Error Log listing and analysis programs which encounter an illegal or unrecognized entry type must be prepared to display that entry in hexadecimal. The code of zero is reserved for null entries; error list and analysis programs are expected to skip these entries.

560 specific formats are noted on the formats involved and are logged only when CP-V or CP-R is running on the 560 machine. Basically, the 560 error reporting consists of two secondary detail records which record the status of each processor with valid status as reported by a POLR instruction and the status of each memory with a set error bit as reported by a LMS instruction. These records are produced by routine which are called from several places in the monitor following a primary error record:

PFI
MFI
I/O With Memory Fault
Watchdog Timer
Hardware Fault Trap (4C)
Instruction Exception

Details of the mechanism are given in Reference 6.

The following codes and causes are utilized by CP-R:

Error Log Code	Cause
11	Recorded when the condition codes returned by the SIO instruction are such that both CC1 and CC2 are true.
12	Recorded when the timeout value specified by DCT11 has been exceeded.
13	Recorded when no match can be found between I/O address returned in the status register by the AIO instruction and any DCT1 I/O address of a device known to be busy. AIO CC=11XX will not be logged.
15	Recorded when an I/O request is not successful upon one of the specified number of retries. May or may not have eventually been successful. Only the status of the last erroneous retry for a given request is logged, not all the retries.
16	Recorded to log specific information returned in response to a sense order issued to a device which has indicated an error.
17	Recorded when program execution traps to hex location 56 due to a memory parity condition.
18	Recorded when system is booted.
19	Recorded when program execution traps to hex location 46 due to a watchdog timer runout condition.
1D	Recorded when program execution traps to location X'4D' on Sigma 9 or Xerox 560 due to an instruction exception condition.
21	Recorded when system is booted.
22	Recorded when system is booted.
23	Recorded whenever the value in K:TIME (seconds since midnight) is a multiple of 3600, i.e., once each hour).
27	Recorded when there is an unsolicited ERRSEND key-in by the operator.

30	Recorded when there is a processor fault interrupt (hex location 56).
31	Recorded when there is a memory fault interrupt (hex location 57).
32	Recorded for each nonzero poll status received by the processor poll subsystem.
41	Recorded when system is booted.
42	Recorded for each memory unit that has recorded an error as determined by the memory poll subsystem (i.e., bits 22-31 of status word zero are nonzero) for 560.
43	Recorded for each memory unit that has recorded an error as determined by the memory poll subsystem (i.e., bits 22-31 of status word zero are nonzero) for Sigma 9.
44	Recorded for each memory unit that has recorded an error as determined by a routine that loops through memory with errors causing an interrupt to hex location 56 for Sigma 7.

The following codes and causes are utilized by CP-V:

CODE	DESCRIPTION	CAUSE
10	Copy Error	Recorded as a result of several possible error conditions in the error logging mechanism. If the record subtypes as 03, 05, or 06, the record is followed by the 64-word buffer in which the error occurred.
11	SIO Failure	Recorded when condition codes returned by an SIO instruction are such that CC1 or CC2 are set.
12	Device Timeout	Recorded when the timeout value specified by DCT11 has been exceeded. The operating system monitors the length of time operations take for a device to perform. If the device exceeds this time, the operating system assumes faulty operation, records this entry and warns the operator.
13	Unexpected Interrupt	Recorded when no match can be found between the I/O address returned in the status register by an AIO instruction and only DCT1 I/O address of a device known to be busy.
15	Device Error	Recorded as a result of examining the status returned in the status register by an AIO, TIO, or TDV instruction and finding an error condition. This record may be followed by 0 to n Memory Parity Secondary Records (42, 43, or 44), 0 to n Secondary Records for Poll Information (32), and Secondary Records for Pack, RAD, and Tape (16), depending on error and machine types.
16	Secondary Record for Pack, RAD, Tape	Recorded to log specific information returned in response to a sense order issued to a device which has indicated an error. This record is preceded (not necessarily contiguously) by the Device Error record (15).

CODE	DESCRIPTION	CAUSE
2D	Hardware Error	Recorded when program execution is interrupted to location X'56' on Sigma 7 or is trapped to location X'4C' on Sigma 9 or Xerox 560. This record may be followed by 0 to n Memory Secondary Records (42, 43, or 44) and 0 to n Secondary Records for Poll Information (32) depending on error and machine type.
18	System Startup	Recorded when the system is booted and at every recovery. This record is followed by System Identification (22), Time Stamp (23), 1 to n Configuration Records (21), or 1 to n Processor Configuration Records (41) depending on machine type.
2E	Watchdog Timer	Recorded when program execution traps to location X'46' due to a hardware watchdog timer runout condition. This record may be followed by 0 to n Memory Secondary Records (42, 43, or 44), and 0 to n Secondary Records for Poll Information (32) depending on error and machine type.
1A	File Inconsistency Entry	Recorded when the operating system finds it cannot, for whatever reason, access a file in the File Management System. Displayed code is described in the BP Reference Manual for CP-V.
1B	Software Detected Symbiont Inconsistency	Recorded when the operating system finds it cannot, for whatever reason, access a symbiont file in the Symbiont File Management System.
2F	Instruction Exception	Recorded when program execution traps to location X'4D' on Sigma 9 or Xerox 560 due to an instruction exception condition. This record may be followed by 0 to n Memory Secondary Records (42, 43, or 44), and 0 to n Secondary Records for Poll Information (32) depending on error and machine type.
1E	Lost Entry Indicator	Recorded when error log buffering constraints, timing considerations, and error detection rates force error logging to be temporarily suspended or otherwise impossible.

CODE	DESCRIPTION	CAUSE
1F	Duplicate Entries	Recorded when the error logging mechanism detects identical consecutive errors. This prevents the error log from becoming saturated with redundant information.
20	Power On	Recorded when the hardware power monitor forces program execution to trap the location X'51' as a result of detecting a restoration of power condition. This normally occurs as a result of a power outage of 500 milliseconds or more in duration.
21	Configuration Recovery	Recorded when the system is booted and at every recovery as part of the set of secondary records associated with System Startup record (18) and as part of the set of secondary records associated with System Identification record (22).
22	System Identification	Recorded when the system is booted and at every recovery as part of the set of secondary records associated with System Startup record (18). It is also recorded when the error log file ERRFILE is empty and the first records transferred to the file were not System Startup (18) and System Identification (22). When this record is the first record in the ERRFILE file, it is followed by Time Stamp (23), 1 to n Configuration Records (21), 1 to n I/O Activity Count (28), or 1 to n Processor Configuration Records (41) depending on machine types.
23	Time Stamp	Recorded when the system is booted at every recovery as part of the set of secondary records associated with System Startup record (18) and part of the set of secondary records associated with System Identification (22). It is also recorded each and every hour on the hour.
24	Bad Granule Release	Recorded when either a bad disk address has been detected or when the granule to be released is already free (dual allocation).

CODE	DESCRIPTION	CAUSE
26	Remote Processing Error Record	Recorded when an error is detected in the transmission of data from a remote processing workstation.
27	Operator Message	Recorded when there is an unsolicited ERRSEND key-in made by the system operator.
28	I/O Activity Count	Recorded at every recovery and every hour on the hour following the Time Stamp (23) and as part of the set of secondary records associated with System Identification (28). I/O activity counts will be reset after this record is logged after recovery.
30	PFI Primary Record	Recorded when program execution is interrupted to location X'56' on the Xerox 560 due to a processor fault interrupt condition. This record may be followed by 0 to n Memory Parity Secondary Record (42), and 0 to n Secondary Records for Poll Information (32) depending on error type.
31	MFI Primary Record	Recorded when program execution is interrupted to location X'56' on the Sigma 9 or Xerox 560 due to a memory fault interrupt condition. This record may be followed by 0 to n Memory Parity Secondary Records (42 or 43), and 0 to n Secondary Records for Poll Information (32) depending on error and machine type.
32	Secondary Record for Poll Information	Recorded to log information obtained by issuing a POLL instruction subsequent to a Device Error (15), Watchdog Timer (19) Hardware Error (17), Instruction Exception (1D), Processor Fault Interrupt (30), and Memory Fault Interrupt (31), only if useful information has been received from the POLL instruction.

CODE	DESCRIPTION	CAUSE
41	Processor Configuration	Recorded when system is booted as part of the set of secondary records associated with System Startup Record (18), and as part of the set of secondary records associated with System Identification Record (22).
42	560 Memory Parity Secondary Record	Recorded to log specific information returned in response to an LMS instruction subsequent to a Device Error (15), Hardware Error (17), Watchdog Timer (19), Instruction Exception (1D), PFI (30), or MFI (31).
43	Sigma 9 Memory Parity Secondary Record	Recorded to log specific information returned in response to an LMS instruction subsequent to a Device Error (15), Hardware Error (17), Watchdog Timer (19), Instruction Exception (1D), or MFI (31).
44	Sigma 5-7 Memory Parity Secondary Record	Recorded to log specific information obtained by scanning memory to attempt to isolate locations which cannot sustain correct parity subsequent to a Device Error (15), Hardware Error (17), Watchdog Timer (19), or Instruction Exception (1D).
50	Enqueue Table Overflow	Recorded to log specific information after the operating system has detected an enqueue table overflow condition.
51	Partitioned Resource	Recorded when a resource has been partitioned from the system.
52	Returned Resource	Recorded when a resource, previously partitioned, has been returned to the system.

10 - COPY ERROR (Several Subtypes)

Read Error (CP-V) subtype

<table>
<tr><td>10</td><td>02</td><td>▨</td><td>01</td></tr>
<tr><td colspan="4">Relative Time</td></tr>
</table>

Read Error End (CP-V)

<table>
<tr><td>10</td><td>02</td><td>▨</td><td>02</td></tr>
<tr><td colspan="4">Relative Time</td></tr>
</table>

Error Log Record Length Error (CP-V)

<table>
<tr><td>10</td><td>03</td><td>▨</td><td>03</td></tr>
<tr><td colspan="4">Relative Time</td></tr>
<tr><td colspan="4">Index to Bad Entry</td></tr>
</table>

Incorrect Time (CP-V)

<table>
<tr><td>10</td><td>03</td><td>▨</td><td>05</td></tr>
<tr><td colspan="4">Relative Time</td></tr>
<tr><td colspan="4">Index to Bad Entry</td></tr>
</table>

Illegal Entry Type (CP-V)

<table>
<tr><td>10</td><td>03</td><td>▨</td><td>06</td></tr>
<tr><td colspan="4">Relative Time</td></tr>
<tr><td colspan="4">Index to Bad Entry</td></tr>
</table>

The above entries occur as a result of an error condition arising in the error log recording mechanism.

11 - SIO FAILURE (CP-R/CP-V)

<table>
<tr><td>11</td><td>COUNT
= 6</td><td colspan="2">Model #</td><td></td></tr>
<tr><td colspan="4">Relative Time</td><td></td></tr>
<tr><td colspan="2">SIO STATUS</td><td colspan="2">I/O Address</td><td>DCT21, DCT1</td></tr>
<tr><td>MFI if
Σ6 or Σ7</td><td>SIO
CC</td><td>TDV
CC</td><td>▨</td><td>-DCT19, DCT20</td></tr>
<tr><td>subchan
status ▨</td><td colspan="3">TDV CURRENT
COMMAND DA</td><td rowspan="2">DCT13</td></tr>
<tr><td colspan="2">TDV Status</td><td colspan="2">Bytes Remaining</td></tr>
</table>

The SIO Failure is emitted when following SIO CC are returned

5-7	8-9 & 56
01xx	010x
10xx **	100x **
11xx	110x

The I/O sequence is SIO, TDV.

** The CC responses which indicate IOP busy may or may not be considered an error condition due to Operating System dependency.

12 - DEVICE TIMEOUT (CP-R/CP-V)

<table>
<tr><td>12</td><td>COUNT = D</td><td colspan="2">Model #</td><td></td></tr>
<tr><td colspan="4">Relative Time</td><td></td></tr>
<tr><td colspan="2">HIO Status</td><td colspan="2">I/O Address</td><td>DCT12</td></tr>
<tr><td>MFI if Σ6 or Σ7</td><td>HIO CC</td><td>TDV CC</td><td>TIO CC</td><td>-, DCT19, DCT20, DCT20</td></tr>
<tr><td>subchan status</td><td></td><td colspan="2">TDV Current Command DA</td><td rowspan="2">DCT13</td></tr>
<tr><td colspan="2">TDV Status</td><td colspan="2">Bytes Remaining</td></tr>
<tr><td colspan="4">Current Command Doubleword</td><td></td></tr>
<tr><td colspan="2">TIO Status</td><td>Retry Request</td><td>Retries Remaining</td><td>DCT21, IOQ10, IOQ11</td></tr>
<tr><td colspan="4">I/O Count</td><td>DCT25</td></tr>
<tr><td colspan="2">Volume Serial Number (6 characters for CP-V ANS Tape)</td><td>Original Function Code</td><td>Current Function Code</td><td></td></tr>
<tr><td colspan="4">Seek Address</td><td>IOQ12</td></tr>
</table>

13 - UNEXPECTED INTERRUPT (CP-R/CP-V)

13	COUNT = 4	Model #		
Relative Time				
AIO Status		I/O Address		DCT12
▨	AIO CC	▨		-, DCT19, -, -

15 - DEVICE ERROR (CP-R/CP-V)

<table>
<tr><td>15</td><td>COUNT = D</td><td colspan="2">Model #</td><td></td></tr>
<tr><td colspan="4">Relative Time</td><td></td></tr>
<tr><td colspan="2">AIO Status</td><td colspan="2">I/O Address</td><td>DCT12</td></tr>
<tr><td>MFI if Σ6 or Σ7</td><td>AIO CC</td><td>TDV CC</td><td>TIO CC</td><td>-, DCT19, DCT20, DCT20A</td></tr>
<tr><td>subchan status</td><td></td><td colspan="2">TDV Current Command DA</td><td rowspan="2">DCT13</td></tr>
<tr><td colspan="2">TDV Status</td><td colspan="2">Bytes Remaining</td></tr>
<tr><td colspan="4">Current Command Doubleword</td><td></td></tr>
<tr><td colspan="2">TIO Status</td><td>Retry Request</td><td>Retries Remaining</td><td>DCT21, IOQ10, IOQ11</td></tr>
<tr><td colspan="4">I/O Count</td><td>DCT25</td></tr>
<tr><td colspan="2">Volume Serial Number
(6 characters for CP-V ANS Tape)</td><td>Original Function Code</td><td>Current Function Code</td><td></td></tr>
<tr><td colspan="4">Seek Address</td><td>IOQ12</td></tr>
</table>

16 - SECONDARY RECORD FOR PACK, RAD, TAPE (CP-R/CP-V)

16	COUNT as Needed	I/O Address
Relative Time		
Sense Information		

7242	10 bytes
Disk A, B	16 bytes
1600 BPI Tape	6 bytes
Nine Track Tape	1 byte
RAD	3 bytes

Note: The I/O Address links the secondary record to the correspondin device error entry.

17 - HARDWARE ERROR (CP-R/CP-V)

<table>
<tr><td>CODE XX1</td><td colspan="4">COUNT=OB</td><td>CPU ADDRESS</td></tr>
<tr><td colspan="6">TIME</td></tr>
<tr><td colspan="6">PSD1</td></tr>
<tr><td colspan="6">PSD2</td></tr>
<tr><td colspan="6">RESERVED</td></tr>
<tr><td colspan="6">RESERVED</td></tr>
<tr><td>TRAP CC</td><td></td><td>RBP</td><td>IAP</td><td>FIP</td><td>0 0 0 0</td></tr>
<tr><td colspan="6">REAL ADDRESS OF TRAPPED INSTRUCTION</td></tr>
<tr><td colspan="6">TRAP INSTRUCTION</td></tr>
<tr><td>ANLZ CC</td><td colspan="5">EFFECTIVE ADDRESS</td></tr>
<tr><td colspan="6">REAL EFFECTIVE ADDRESS</td></tr>
</table>

Generated by interrupt 56 on Sigma 7 and trap 4C on Sigma 9 and 560.

where XX1 = X'2D', X'2E', X'2F'

FIP - When set indicates that a PE occurred while fetching the instruction (causing T4C) or this is an I56 on Sigma 7. In either case, words 4 - 9 will be zero.

IAP - When set indicates that a PE occurred due to an indirect address fetch. Words 8 - 9 will be zero.

RBP - When set indicates that a PE is present in the associated R-block registers (Xerox 560 only).

The Effective Address and the Real Effective Address will be of the addressing type indicated by the ANLZ CC. (b, hw, w, dw, i).

If the instruction is an immediate type, these will be zero.

18 - SYSTEM STARTUP (CP-V)

0 31

<table>
<tr><td>18</td><td>COUNT = 4</td><td>Startup Type</td><td>Recovery Count</td></tr>
<tr><td colspan="4">Relative Time</td></tr>
<tr><td colspan="2">Year - 1900</td><td colspan="2">Julian Day</td></tr>
<tr><td>Screech Code</td><td>Screech Subcode</td><td colspan="2">CPU ADDRESS</td></tr>
</table>

- Screech Code, subcode defined in CP-V Operations Reference Manual
- Recovery Count = 0 for initial startup (values 1, 2, 3 below)
- Startup type codes:

Value	Meaning
1	PO Boot, Initial
2	PO Boot, Under Files
3	System Device Boot, (No Recovery)
4	System Recovery
5	Operator Recovery

- For Multiprocessing CP-V

6	Slave CPU Start Up
7	Slave CPU Shut Down

19 - WATCHDOG TIMER (CP-R/CP-V)

Generated by Trap 46

<table>
<tr><td>CODE XX1</td><td colspan="4">COUNT=OB</td><td>CPU ADDRESS</td></tr>
<tr><td colspan="6">TIME</td></tr>
<tr><td colspan="6">PSD1</td></tr>
<tr><td colspan="6">PSD2</td></tr>
<tr><td colspan="6">RESERVED</td></tr>
<tr><td colspan="6">RESERVED</td></tr>
<tr><td>TRAP CC</td><td></td><td>RBP</td><td>IAP</td><td>FIP</td><td>0 0 0 0</td></tr>
<tr><td colspan="6">REAL ADDRESS OF TRAPPED INSTRUCTION</td></tr>
<tr><td colspan="6">TRAP INSTRUCTION</td></tr>
<tr><td>ANLZ CC</td><td colspan="5">EFFECTIVE ADDRESS</td></tr>
<tr><td colspan="6">REAL EFFECTIVE ADDRESS</td></tr>
</table>

where XX1 = X'2D', X'2E', X'2F'

FIP - When set indicates that a PE occurred while fetching the instruction (causing T4C) or this is an I56 on Sigma 7. In either case, words 4 - 9 will be zero.

IAP - When set indicates that a PE occurred due to an indirect address fetch. Words 8 - 9 will be zero.

RBP - When set indicates that a PE is present in the associated R-block registers (Xerox 560 only).

The Effective Address and the Real Effective Address will be of the addressing type indicated by the ANLZ CC. (b, hw, w, dw, i).

If the instruction is an immediate type, these will be zero.

1A - FILE INCONSISTENCY ENTRY (CP-V)

1A	COUNT = 14	75	Error Subcode	
Relative Time				
File Name (8 words)				From DCB
Account				From DCB
Generalized Disk Address				
Consecutive = 1 ORG Keyed = 2 Random = 3		Mode { in = 1 out = 2 inout = 4 outin = 8		

1B - SOFTWARE-DETECTED SYMBIONT INCONSISTENCIES (CP-V)

0			31
1B	COUNT = 3	00	DCT INDEX of SYMBIONT
Relative Time			
Generalized Disk Address			

1D - INSTRUCTION EXCEPTION (CP-R/CP-V)(560 and SIGMA 9)

Generated by Trap 4D

<table>
<tr><td>CODE XX1</td><td colspan="4">COUNT=OB</td><td>CPU ADDRESS</td></tr>
<tr><td colspan="6">TIME</td></tr>
<tr><td colspan="6">PSD1</td></tr>
<tr><td colspan="6">PSD2</td></tr>
<tr><td colspan="6">RESERVED</td></tr>
<tr><td colspan="6">RESERVED</td></tr>
<tr><td>TRAP CC</td><td></td><td>RBP</td><td>IAP</td><td>FIP</td><td>0 0 0 0</td></tr>
<tr><td colspan="6">REAL ADDRESS OF TRAPPED INSTRUCTION</td></tr>
<tr><td colspan="6">TRAP INSTRUCTION</td></tr>
<tr><td>ANLZ CC</td><td colspan="5">EFFECTIVE ADDRESS</td></tr>
<tr><td colspan="6">REAL EFFECTIVE ADDRESS</td></tr>
</table>

where XX1 = X'2D', X'2E', X'2F'

FIP -	When set indicates that a PE occurred while fetching the instruction (causing T4C) or this is an I56 on Sigma 7. In either case, words 4 - 9 will be zero.
IAP -	When set indicates that a PE occurred due to an indirect address fetch. Words 8 - 9 will be zero.
RBP -	When set indicates that a PE is present in the associated R-block registers (Xerox 560 only).

The Effective Address and the Real Effective Address will be of the addressing type indicated by the ANLZ CC. (b, hw, w, dw, i).

If the instruction is an immediate type, these will be zero.

1E - LOST ENTRY INDICATOR (CP-V)

1E	COUNT = 2	Count of entries lost
Relative time of last lost entry		

The above entry is logged when buffering constraints make error logging temporarily impossible. The newest entries are lost.

1F - DUPLICATE ENTRIES (CP-V)

1F	COUNT = 2	Count of entries identical to previous
Relative Time of last duplicate		

Errorlog routine compares current entry with previous and maintains a count as long as entries are identical, except time. If current an previous are different and count is non-zero, then a duplicate entries entry is made.

20 - POWER ON (CP-V)

20	COUNT = 2	▨
Relative Time		

21 - CONFIGURATION RECORD (CP-R/CP-V)

21	COUNT as needed	▨	▨
Relative Time			
Model #		▨	DCT Index
Alternate I/O Address		Primary I/O Address	

Entered at system STARTUP or RECOVERY

One pair of words per device in DCT order; multiple records may occur -- maximum 5 devices per record

22 - SYSTEM IDENTIFICATION (CP-R/CP-V)

22	COUNT = 5	Core size in 8K word blocks	Relative Time Resolution
Relative Time			
system	▨	version	flags
Site Identification			

Recorded at system STARTUP and RECOVERY

Relative Time Resolution is expressed as a value of n such that actual relative time resolution = 2^n msec. The value of n for the most likely resolutions are

n = 0	when the timing source is supplied by a frequency $\geq$1KHZ
n = 1	500HZ
n = 4	60HZ

For CP-V and CP-R, n = 1.

System, Version, Flags

The format of system, version, flags and of site identification is operating system specific. System, version, and flags are formatted as location X'2B'.

2B	monitor	▨	version		CPU	Parameters
	0 3	4 7	8 15	16 21	22 25	26 31

Location 2B contains three items:

1. Monitor - this field contains the code number of the Monitor. The codes are:

Code	Monitor
0	None or indeterminate
1	BCM
2	RBM
3	RBM-2
4	BPM
5	BTM/BPM
6	UTS
7	CP-V
8	CP-R
9-F	Reserved for future use

2. Version - this is the version code of the Monitor and is coded to correspond to the common designation for versions. The alphabetic count of the version designati is the high-order part of the code and the version number is the low-order part. For example, A00 is coded X'10' and D02 is coded X'42'.

3. Parameters - the bits in this field are used to indicate suboptions of the Monitor. They are meaningful only in relation to a particular Monitor. However, the following assignments have been made for BPM, BTM, and CP-V.

Bit	Meaning if Set
31	Symbiont routines included
30	Remote Batch routines included
29	Real-time routines included
28	Unused
27	Reserved for Data Management System
26	Reserved
22-25	Field defining CPU
18	Multiprocessing
17	Transaction Processing
16	On-line

Bit 22	Bit 23	Bit 24	Bit 25	Meaning
	0	0	0	Sigma 5
	0	0	1	Sigma 6/Sigma 7 (for CP-R 01 defines Sigma 5/
	0	1	0	Sigma 9
	0	1	1	550/560

23 - TIME STAMP (CP-R/CP-V)

23	COUNT = 3	▨
Relative Time		
Year - 1900		Julian Day

For CP-V, this record entered by ERR:FIL once each hour on the hour

Year - 1900, Julian Day: as binary integers

24 - BAD GRANULE RELEASE (CP-V)

24	COUNT = 4	TYPE CODE
Relative Time		
Generalized Disk Address		
Address of Routine Calling ERRLOG or 'ACAT'		

If TYPE CODE = 0

or

Number of Granules Being Released	▨

If TYPE CODE = 1

TYPE CODE

0	Bad Address
1	Dual Allocation

'ACAT' is substituted for the address of routine calling ERRLOG when ALLYCAT detects the error.

26 - REMOTE PROCESSING ERROR RECORD (CP-V)

This record is logged when an error has occurred in the transmission of data from a remote processing workstation.

26	COUNT = 8	I/O Address	
Relative Time			
RB:FLAGS			
Workstation Name			
Current Command Doubleword			
RP1	RP2	RP3	RP4

RP1, RP2, RP3, and RP4 have specific meaning for the type of remote workstation associated with the record as defined by the following tables. Refer to the Remote Processing Reference Manual for definition of terms and codes.

- Xerox 7670 Workstation

RP1	Definition	RP3	RP4
1	First character in record not SOH	Current character position	Offending character
2	Incorrect parity on SEL	"	"
3	Incorrect block protect	"	"
4	Third character in record not STX	"	"
5	RBBAT combuf or MPOOL unavailable for logon	Meaningless	Meaningless
6	Incorrect character parity	Current character position	Offending character
7	Record trailer character not ETX	"	"
8	Incorrect block check parity	"	"
9	Incorrect block check	"	"
A	Communication line time-out	Meaningless	Meaningless
B	NAK received	Response received reading for ACK	
C	Garbled ACK or NAK	Response received reading for ACK	

RP2 represents the current function code associated with the workstation.

RP2	Definition
0	Write card punch
1	Write line printer
2	Send ACK
3	Write TOF (Block protect = 0)
4	Write TOF (Block protect = 1)
5	Write SPACE (Block protect = 0)
6	Write SPACE (Block protect = 1)
7	Read card reader
8	Write TOF (logon)
9	Read card reader (special)
A	Read ACK card punch
B	Read ACK line printer
C	Read ACK TOF (Block protect = 0)
D	Read ACK TOF (Block protect = 1)
E	Read ACK SPACE (Block protect = 0)
F	Read ACK SPACE (Block protect = 1)
10	Write EOT
11	Write DC 1
12	Write ACK (special)
13	Write NAK
14	Write NAK (special)
15	Write BEL (on error)

For Xerox 7670 workstations, the current command doubleword in the error log record contains the second command doubleword used to write the text of an output message and is meaningful only for RP1 = 0, 1, A, or B.

Intelligent Remote Batch Terminal (IRBT)

RP1	Definition	RP4 Definition
0	Recoverable block check error	Expected BCB count (sign extended)
1	Catastrophic block check error (NAK sent in case of line error)	Expected BCB count (sign extended)
2	Communication line time-out	Same as RP2
3	Read for ENQ timed-out (logon)	Same as RP2
4	Received ACKO instead of SIGNON at logon	ACKO
5	Inappropriate line bid (not ENQ-master, not ACKO-slave)	Line bid received.
6	NAK received	NAK
7	Read timed out	Same as RP2
8	Incorrect CRC	Last character CRCed
9	Trailer character not ETB	Offending character
A	Leader character not STX	Offending character
B	Lost data	First character after DLE
C	Garbled ACKO-NAK	First character of message
FF	RP1 is 0 or 1 with RP4 = -1 (result appears as X'FFFFFFFF')	

For IRBT, RP2 is the current function code and RP3 is the calling function code. The following table defines RP2 and RP3.

Value	RP2	RP3
0	Disconnect	Software error - should not occur
1	Write block	Write block - read block
2	Write ACK	Write ACK - read block
3	Write block	Write block (Wait-a-bit) - Read special
4	Write Wait-a-bit	Write Wait-a-bit - Read special
5	Read Block	Software error - should not occur
6	Send NAK	Software error - should not occur
7	Send ENQ	Logon as Slave
8	Read for ENQ	Logon as Master
9	ACKO to ENQ	Logon as Master after ENQ Read
A	Read Logon Record	Software error- should not occur
B	NAK Logon Record	Software error - should not occur

IBM 2780

RP1	Definition	RP4 Definition
1	Disconnect due to;	
	a) EOT on read	EOT
	b) Use of 2780 on IRBT only system	ENQ
2	Line timeout	Same as RP2
3	ENQ not received on logon read	Character received
4	No EOT after EOF sent	Character received
5	a) ENQ in text mode	Character received
	b) No ENQ answering WACK	Character received
	c) ENQ answer to ACK of EOF	Character received
6	NAK received	Character received
8	CRC failed on input	Last character CRCed
9	Unknown response reading for ACK	Character received
A	Trailer character not ETB or ETX	Character received
C	Header character not STX	Character received

For IBM 2780, RP2 is the current function code and RP3 is the calling function code. The following table defines RP2 and RP3.

Value	RP2	RP3
0	Disconnect	Software error - should not occur
1	Write Data	Write
2	Send ENQ	Send ENQ (Wait)
3	Send ACKO	Read
4	Send WACK	Send WACK (Wait)
5	Write Data	Write EOF
6	Send ENQ	Request to Output
7	Read for ACK, ENQ, EOT (depends on RP3)	POL for Input
8	Read for ENQ	Logon
9	Read	Software error - should not occur
A	Send NAK	Software error - should not occur
B	Send ACK1	Software error - should not occur
C	Send EOT	Software error - should not occur

27 - OPERATOR MESSAGE (CP-R/CP-V)

27	COUNT as needed	▨
Relative Time		
TEXTC COUNT	TEXTC Message Max Size = 71 characters (CP-V) Max Size = 56 characters (CP-R)	

A facility will be provided to inject messages from the operator (or diagnostic program) into the error log. These messages might be used to describe unusual conditions surrounding a particular error. The operator may enter these messages from the operator console via the keyin command 'ERRSEND'.

28 - I/O ACTIVITY COUNT (CP-V)

28	COUNT as needed	DCT Index of First Device
Relative Time		
I/O Address$_1$		DCT Index$_1$
I/O COUNT$_1$		
I/O Address$_2$		DCT Index$_2$
I/O COUNT$_2$		
⋮		

Recorded once per hour and at recovery. Maximum of 10 entries per record. Counts are reset to zero after recording them at recovery.

30 - PFI PRIMARY RECORD (CP-R/CP-V) (560)

<table>
<tr><td>30</td><td>COUNT = 2</td><td>▨</td></tr>
<tr><td colspan="3">Relative Time</td></tr>
</table>

31 - MFI PRIMARY RECORD (CP-R/CP-V) (560/Sigma 9)

<table>
<tr><td>31</td><td>COUNT = 2</td><td>▨</td></tr>
<tr><td colspan="3">Relative Time</td></tr>
</table>

32 - SECONDARY RECORD FOR POLL INFORMATION (CP-R/CP-V) (560)

<table>
<tr><td colspan="2">32</td><td colspan="2">COUNT = 3</td><td>▨</td></tr>
<tr><td colspan="5">Relative Time</td></tr>
<tr><td>▨</td><td>Unit Address</td><td>Poll CC</td><td>Unit Type</td><td>Poll Status</td></tr>
</table>

0 1 2 7 8 11 12 15 16

One record produced per valid poll status received.

41 - 560 PROCESSOR CONFIGURATION (CP-R/CP-V) (560)

<table>
<tr><td colspan="2">41</td><td colspan="2">COUNT
as needed</td><td>▨</td></tr>
<tr><td colspan="5">Relative Time</td></tr>
<tr><td>TYPE CODE</td><td>2 ▨</td><td>3
CL</td><td>3
UN</td><td>POLR Results</td></tr>
<tr><td colspan="5">One entry for each unit
in the cluster. Maximum 8</td></tr>
</table>

One record per cluster defined in sysgen.

CL = cluster #
UN = unit #
TYPE = unit type, high bit set if unit does not respond during initialization

TYPE CODE	UNIT NAME
1	CPU
2	MI
3	PI
4	MIOP
5	RMP
6	CT
7	SU

42 - 560 Memory Parity Secondary Record (CP-R/CP-V) (560)

<table>
<tr><td>42</td><td>COUNT = 4</td><td>▨</td></tr>
<tr><td colspan="3">Relative Time</td></tr>
<tr><td colspan="3">Memory Status Word 0</td></tr>
<tr><td colspan="3">Memory Status Word 1</td></tr>
</table>

43 - SIGMA 9 Memory Parity Secondary Record (CP-R/CP-V)

<table>
<tr><td>43</td><td>COUNT = 5</td><td>▨</td></tr>
<tr><td colspan="3">Relative Time</td></tr>
<tr><td colspan="3">Memory Status Word 0</td></tr>
<tr><td colspan="3">Memory Status Word 1</td></tr>
<tr><td colspan="3">Memory Status Word 2</td></tr>
</table>

44 - SIGMA 5-7 Memory Parity Secondary Record (CP-R/CP-V)

<table>
<tr><td>44</td><td>COUNT
as needed</td><td>Number of Parity Errors</td></tr>
<tr><td colspan="3">Relative Time</td></tr>
<tr><td colspan="3">1st Bad Location</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td colspan="3">14th Bad Location</td></tr>
</table>

50 - Enqueue Table Overflow (CP-V)

<table>
<tr><td>50</td><td>COUNT = 3</td><td>▨</td></tr>
<tr><td colspan="3">Relative Time</td></tr>
<tr><td colspan="2">User ID</td><td>Entry Count</td></tr>
</table>

- Binary integers

Entry count is the number of entries in the enqueue table belonging to the specified user at the time the error log entry was made.

51 - Partitioned Resource (CP-V)

<table>
<tr><td colspan="2">51</td><td>COUNT = 3</td><td>Model #</td></tr>
<tr><td colspan="4">Relative Time</td></tr>
<tr><td>F</td><td colspan="2">0-------------------0</td><td>I/O Address</td></tr>
</table>

0 1

F = 0 for device entry
F = 1 for controller entry

This entry will be logged when a resource has been partitioned.

52 - Retumed Resource (CP-V)

<table>
<tr><td colspan="2">52</td><td>COUNT = 3</td><td>Model #</td></tr>
<tr><td colspan="4">Relative Time</td></tr>
<tr><td>F</td><td colspan="2">0------------------0</td><td>I/O Address</td></tr>
</table>

F = 0 for Device Entry
F = 1 for Controller entry

This entry will be logged when a resource is returned from beign partitioned.

GENERALIZED DISK ADDRESS

CP-V utilizes a special format for rotating memory disk addresses and specialized procedures for handling them. The format of the generalized disk address word follows.

0 7	8	9 15	16 31
00	RSE	DCT or VNO	RELATIVE SECTOR NUMBER

RSE	is the relative sector number extended bit which is set when relative sector numbers above 65,535 are addressed.
DCT	is the DCT index of the device on which the sector addressed may be found.
VNO	is the volume number within a private volume set of the devices on which the sector addressed may be found.
Relative Sector Number	is the sector number, of the first sector in a granule which is to be accessed. Sector number progresses from zero through device end.

It is recommended that the following set of procedures be used when any of the above items are desired:

[LABEL]	LDCTX,R	[*]ADDRESS,X	Load DCT index
[LABEL]	STDCTX,R	[*]ADDRESS,X	Store DCT index
[LABEL]	LSECTA,RU1	[*]ADDRESS,X	Load sector number
[LABEL]	STSECTA,RU1	[*]ADDRESS,X	Store sector number

where:

R	register to be loaded or stored.
RU1	odd register to be loaded or stored.
ADDRESS	word address of location containing (or to contain) the disk address.
X	word alignment index register.

These procedures are available in the standard SYSTEM UTS file callable by including the following statement in the appropriate METASYMBOL program:

```
SYSTEM      UTS
```

CSE STOP Tables

In the event of a catastrophic hardware failure in the central system components (CPU, memory, IOPs, and data busses), the fault handling system attempts to log the faults in the in-core error log buffers at BUF1 and BUF2, and in addition, collects additional information into tables in memory. These tables are collectively called the CSE STOP tables. In the event of a "CSE STOP" message being output on the OC, the operator should record the contents of these tables before attempting an operator recovery or turning the machine over to customer engineers.

The following items are contiguous in memory:

LABEL	SIZE	MEANING OF CONTENTS
CSED$CF	1 word	High order byte contains trap condition code
CSED$REGS	16 words	Registers at time of last CSE trap
CSED$BREG	1 word	Meaningful only on Xerox 560 - contents of last branch register
CSED$PSD	2 words	Current PSD at time of trap
CSED$WHY	1 word	Internal code for cause of trap
CSED$MSTAT$ADR	1 word	0 for Sigma 6/7, address of CSED$MSTAT table on Sigma 9 and Xerox 560
CSED$PSTAT$ADR	1 word	0 for Sigma 6/7/9, address of CSED$PSTAT table on Xerox 560
CSED$3STAT$ADR	1 word	Address of memory content polling table
CSED$MSG	7 words	Buffer containing CSE STOP message (reason for stopping)

The following items are contiguous in memory:

LABEL	SIZE	MEANING OF CONTENTS
3COUNT	1 word	Number of bad locations in memory
3ANDADD	1 word	Logical AND of addresses of bad locations
3ORADD	1 word	Logical OR of addresses of bad locations
3ANDCONT	1 word	Logical AND of contents of bad locations
3ORCONT	1 word	Logical OR of contents of bad locations
3FIRST	1 word	Address of lowest cell with bad contents
3FIRSTC	1 word	Contents of lowest cell with bad contents
3LAST	1 word	Address of highest cell with bad contents
3LASTC	1 word	Contents of highest cell with bad contents

Optional Table for Sigma 9, Xerox 560 Only

CSED$MSTAT table (Memory Fault Status Register Polling Table)

This table contains up to 5 entries of the form:

Internal Polling Flags	Real Address Causing Fault	Word 0
Memory Fault Status Word 0		Word 1
Memory Fault Status Word 1		Word 2
Memory Fault Status Word 2 (Sigma 9 Only)		Word 3 (Sigma 9 only)

Optional Table for Xerox 560 Only (ref. 560 OPS Manual)

CSED$PSTAT table (Processor Fault Status Register Polling Table)

Each entry is one word long of the format:

0	1	2 7	8 11	12 15	16 31
*	▨	Proc Addr	POLR CC	Type Code	Status

*	Bit 0, when set indicates presence of valid status
Proc Addr	Processor's cluster/unit address
POLR CC	Condition codes from POLR instruction
Type Code	Processor type: 1 - BP, 2 - MI, 3 - PI, 4 - MIOP, 5 - RMI, 6 - CI, 7 - SU
Status	16-bit contents of fault status registers (result of POLR instruction)

ID

ASSIGN/MERGE Table

FUNCTION

An Assign/Merge Table is the place where information is kept which must be merged into the DCBs that go into the user's context at load or execute time.

An Assign/Merge Table is associated with each job in the system (whether batch or on-line) as a result of a :JOB command in batch or as a result of logging on.

Modifications to the user's Assign/Merge Table result from the ASSIGN command in batch and from the SET command and various other commands in TEL. The Assign/Merge Table is merged into the DCBs placed in the user's context when the user loads or executes a program. The merging results from the monitor calling on File Management to do the actual merging.

Assign/Merge Tables built by TEL are never modified by CCI and vice versa.

STRUCTURE

An Assign/Merge Table is one page long. It consists of a 22-word header, followed by a number of linked entries.

Assign/Merge Table

	BATCH	ON-LINE
0	Index to First Available Area	Index to First Available Area
1	Index to First Entry or 0	Index to First Entry or 0
2	Unused	Unused (4 words)
3	Load Module Name for :RUN (3 words)	
4		
5		
6	Load Module Password for :RUN (2 words)	Image of J:CPPO from User JIT
7		Image of J:ASSIGN from User JIT

8	DATE: YEAR Binary	DAY Julian
9	USER NAME 3 words	
10		
11		
12	LOGON Time Minutes from Midnight	
13	Billing Rate	System ID
14	Extended Accounting Field	
15	6 words	
16		
17		
18		
19		
20	Permanent Remaining RAD Space Allocated	
21	Permanent Remaining Disk Space Allocated	

Assign/Merge Table entries consist of a forward link to the next entry, the DCB name in TEXTC format, and an Adjust DCB (open prime) PLIST whose format is described in the CP-V BP Reference Manual.

DCB names are up to 3 words long for TEL and up to 8 words long for CCI. The Adjust DCB PLIST may be up to 100 words long. The PLIST length for TEL entries is constrained only by the buffer size reserved by TEL to construct the entry. For this reason, TEL entries for COO CP-V may have:

- Maximum of 8 read accounts
- Maximum of 8 write accounts
- Maximum of 8 execute accounts
- Maximum of 3 SNs

word 1	FLINK or 0
	TEXTC DCB Name (zero filled) (Maximum 8 words for CCI) Maximum 3 words for TEL
	Adjust DCB PLIST
	FLINK or 0

.
.
.

Format of RUN Table

Word	0 1 7	8 15	16 23	24 31	
0		N_1	N_2	XSL	
1	0 ——— 0 0 14	15 BIAS 31			
2	F_2				
3	LOAD MODULE NAME				
4	(LMN) IN				
5	TEXTC FORMAT				
6	ACCOUNT				
7	NUMBER				
8	PASSWORD				zero if no password
9					
A	SYMBOLIC START ADDRESS IN TEXTC FORMAT (1 to 19 WORDS)				
1A	± HEX START ADDRESS ADDEND OR ZERO				

where

N_1	is number of SNAP, SNAPC, IF, AND, OR, and COUNT records in System Debug File.
N_2	is number of MODIFY records in System Debug File.
XSL	is value of XSL parameter.
BIAS	is value of BIAS parameter.

Format of Modify Table

The Modify Table is written into the system debug file for the current job.

<table>
<tr><td>2</td><td colspan="2">0 ———————— 0</td><td>res</td><td rowspan="4">location
to
modify</td></tr>
<tr><td>byte count</td><td>C_1</td><td colspan="2">...</td></tr>
<tr><td colspan="4">≈ ≈</td></tr>
<tr><td colspan="4">± hex value or 0</td></tr>
<tr><td>byte count</td><td>C_1</td><td colspan="2">...</td><td rowspan="3">modify value</td></tr>
<tr><td colspan="4">≈ ≈</td></tr>
<tr><td colspan="4">value</td></tr>
</table>

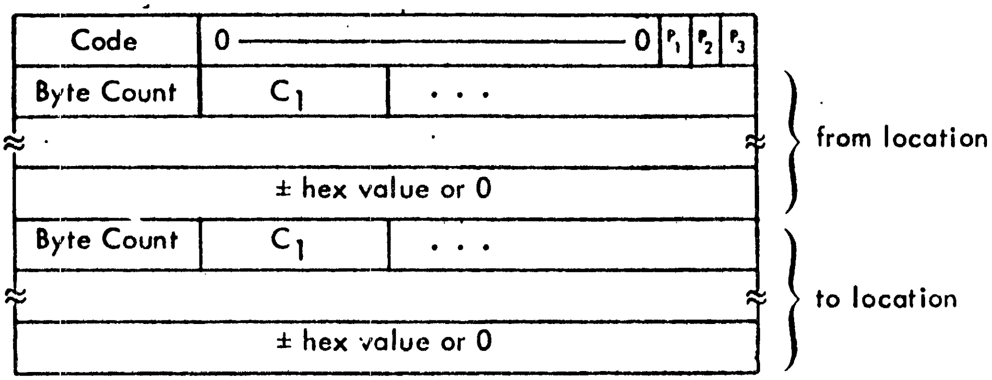
where

res = 0 if byte resolution; = 1 if halfword resolution; = 2 if word resolution; =3 if doubleword resolution. The resolution applies to the name specified with the modify value.

value	is value to be given to the location to be modified. It has the form value + res (name) name is the name of an external definition and value is a 1-digit to 8-digit hexadecimal number.

Format of PMD, PMDE, PMDI Table

The PMD, PMDE, PMDI Table is written into the system debug file for the current job.



where

Code	is 3 for PMD command, 4 for PMDE command, and 5 for PMDI command.
$P_1 = 1$	for dump protection type 00.
$P_2 = 1$	for dump protection type 01.
$P_3 = 1$	for dump protection type 10.
from location	is beginning dump location
to location	is ending dump location

Format of the SNAP, SNAPC Table

The SNAP, SNAPC Table is written into the system debug file for the current job.

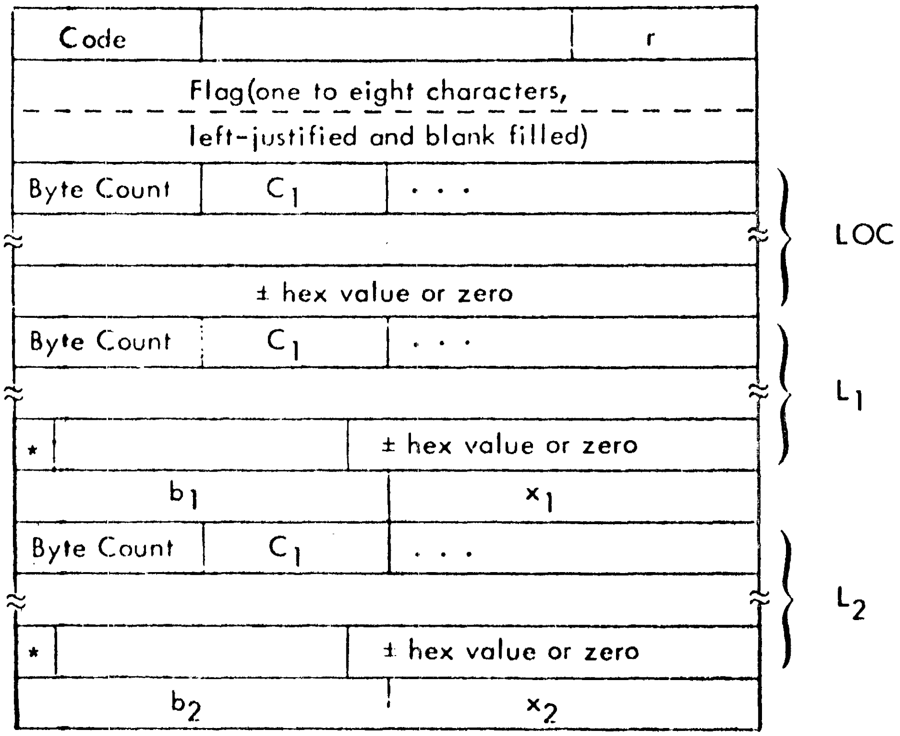
Code	0 —— 0		
Flag (one to eight characters,			
left-justified and blank-filled)			
Byte Count	C_1	. . .	SNAP location
≈		≈	
± hex value or zero			
Comment (one to eight characters,			
left-justified and blank-filled)			
Byte Count	C_1	. . .	from location
≈		≈	
± hex value or zero			
Byte Count	C_1	. . .	to location
≈		≈	
± hex value or zero			

where

Code	is 6 for SNAP command, 7 for SNAPC command
SNAP location	is location at which SNAP is to be done
from location	is beginning dump location
to location	is ending dump location

Format of IF, AND, OR Table

This table is written into the system debug file for the current job.

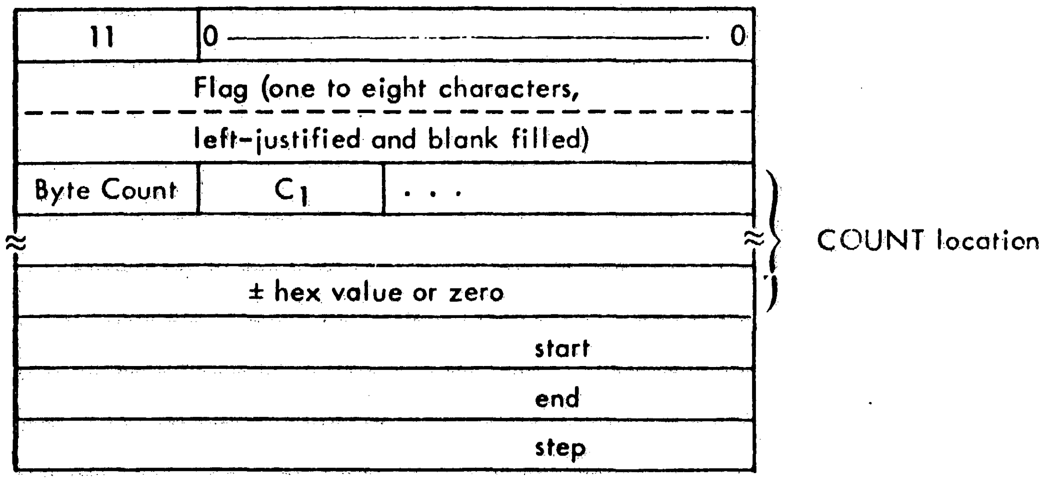


where

Code	is 8 for IF command, 9 for AND command, 10 for OR command.
r	is 0 for GT, 1 for LT, 2 for EQ, 3 for GE, 4 for LE, and 5 for NE.
b_1, b_2	is 0 for byte, 1 for halfword, 2 for word, and 3 for doubleword.
x_1, x_2	is 0 through 7.
*	specifies an indirect address (for L_1 or L_2).
LOC	specifies the location at which the test is to take place.
L_1, L_2	specify the locations to be compared as specified by 'r'.

Format of Count Table

The Count Table is written into the system debug file for the current job.



ID

Error Codes and Error Messages

This section describes the error codes and error message handling in CP-V Subsequent subsections cover 1) the format of the error message file, and the general assignment of codes, 2) the meanings of each of the currently assigned codes, and 3) a list of the messages on file in the current CP-V system.

Codes for various detected error conditions are recorded in JIT. The error code is placed in J:ABC (high order byte, and the subcode in ER0 (right-justified). When PMD, in the case of batch jobs, or TEL, in the case of on-line jobs is entered, a message is printed to correspond to the code and subcode. This message is obtained from the error file via a keyed read using a key constructed from the code and subcode. If either the file or the message record which corresponds to the code is missing, then a message including the error code itself is printed. Processors also use the error message file and are assigned "group codes" for their messages.

For I/O errors the user may gain control in error situations and handle the error himself, by specifying error or abnormal addresses either in the DCB via M:SETDCB or in the M:READ or M:WRITE procs which is described in the UTS Reference Manual, Appendix B.

ERROR MESSAGE FILE FORMAT

The file ERRMSG in the account :SYS contains the error messages of the system, both for Monitor and processor error conditions. The file contains keyed records with keys in the form used by EDIT so that the file may be conveniently changed to suit the individual installation. The record contains the EBCDIC text of the error message.

Keys are one word long and have the form:

03	GC	EC	SC

The first byte always contains 03, the count of bytes in the key, the second byte is the group code, the third, error code, and the fourth is the error subcode.

Group codes presently assigned are:

0 - Monitor
1 - PCL
2 - LOADER
3 - TEL

4 - RUNNER
5 - CCI
6 - DRSP
7 - BATCH
8 - ANLZ

Messages in the file with group codes other than zero are not handled in any way by the Monitor itself.

Error codes currently assigned within the Monitor group are:

0-7F - I/O error and abnormal codes
80-9F - COBOL error codes
A0-BF - Monitor codes
C0-FF - Unused

Contents of the error message file for each of the assigned codes are given in the following pages. The codes follow the format indicated above (with leading zeros suppressed.)

GRP	ERR	SUB	MODULE	MESSAGE
00	1	00	OPEN	INSUFFICIENT INFORMATION TO OPEN DCB.
00	1	0B	OPEN	NUMBER OF REQUESTED GRANULES NOT AVAILABLE.
00	2	00	OPEN	CAN'T OPEN NEXT FILE BECAUSE THERE AREN'T ANY MORE.
00	2	01	OPEN	CAN'T OPEN NEXT ACCOUNT BECAUSE THERE AREN'T ANY MORE.
00	3	00	OPEN	FILE DOES NOT EXIST.
00	4	00	OPEN/READ	BEGINNING OF FILE REACHED.
00	5	00	PRECORD/READ	END OF DATA REACHED.
00	6	00	READ	END OF FILE OR MONITOR CONTROL CARD REACHED.
00	7	00	READ	DATA HAS BEEN LOST. (BECAUSE YOUR BUFFER IS TOO SMALL.)
00	8	00	OPEN	'NEXT' FILE WON'T OPEN BECAUSE IT'S SYNONYMOUS TO SOME OTHER FILE.
00	09	00	RDERLOG	DIAGNOSTIC CLOSE ERROR
00	09	01	RDERLOG	NONEXISTENT DEVICE
00	09	02	RDERLOG	DEVICE BUSY
00	09	03	RDERLOG	SYMBIONT DEVICE BUSY
00	09	04	RDERLOG	NO CLIST IN DCB
00	09	05	RDERLOG	COMMAND LIST DESTROYED BY SWAP
00	09	06	RDERLOG	IOCD'S EXCEED LIMIT
00	09	07	RDERLOG	INVALID COMMAND LIST
00	09	08	RDERLOG	INVALID ADDRESS
00	09	09	RDERLOG	BUFFER CANNOT CROSS PAGE BOUNDARY
00	09	0A	RDERLOG	AUTHORIZATION NOT GRANTED OR < A0 PRIVILEGE
00	09	0B	RDERLOG	INSUFFICIENT CORE TO PERMIT LOCK
00	09	0C	RDERLOG	CONTROLLER NOT PARTITIONED
00	09	0D	RDERLOG	DEVICE REQUESTED NOT PARTITIONED
00	09	0E	RDERLOG	INVALID ADDRESS
00	09	0F	RDERLOG	INSUFFICIENT SPACE TO PROCESS COMMAND LIST
00	A	00	CLOSE	THAT DCB IS ALREADY CLOSED.
00	B	00	OPEN/READ/CVOL	UNRECOGNIZED SENTINEL ON LABELED TAPE.
00	C	00	OPEN	ILLEGAL SYNON OPERATION.
00	D	00	OPEN	NOT ENOUGH ROOM IN VLP OF DCB FOR PRIVATE PACK SN.
00	E	00	OPEN	127 DCB'S OPEN TO FILE - ACCESS DENIED.
00	13	00	DELREC/WRITE	YOU MUST SPECIFY NEWKEY WHEN WRITING A NEW RECORD.
00	14	00	OPEN	FILE ACCESS DENIED-MISSING FILE NAME.ACCOUNT OR PASSWORD.
00	14	01	OPEN	FILE IS BUSY.
00	14	02	OPEN	BAD FPARAM LOCATION.
00	14	03	OPNL	BREAK OR CONTROL-Y DURING MOUNT REQUEST. OPEN NOT PERFORMED.
00	14	04	WRTF	INTERRUPTED M:MOVE OF UNCLEAN RANDOM FILE
00	14	05	OPND	INVALID OP LABEL IN DCB
00	14	11	LBLT	CODE CONVERSION SPECIFIED ON DEVICE NOT HAVING FEATURE
00	14	12	LBLT	800BPI SPECIFIED ON DEVICE NOT HAVING DUAL DENSITY
00	14	13	OPNL	ILLEGAL CODE CONVERSION CHANGE REQUESTED
00	14	14	OPNF	EXECUTE ONLY FILE THE FILE IS IN OPEN STATUS.
00	15	00	DELREC/WRITE	ILLEGAL OPERATION SEQUENCE ON UPDATE FILE
00	15	01	READ/PRECORD	IMPROPER OPERATION SEQUENCE ON SHARED UPDATE FILE
00	16	00	WRITE	CAN'T SAY NEWKEY BECAUSE THE RECORD EXISTS.
00	17	00	WRITE	MUST USE NEWKEY ON OUTPUT FILES
00	18	00	WRITE	KEYS MUST BE GIVEN IN ORDER WHEN WRITING KEYED FILES SEQUENTIALLY.
00	19	00	OPEN/CLOSE	ILLEGAL OPERATION ON M:UC DCB.
00	1A	00	MOVECAL(RDL)	NO ERROR OR ABNORMAL ADDRESS SPECIFIED IN MOVECAL FPT
00	1A	01	MOVECAL(RDL)	SECOND DCB IS MISSING
00	1A	02	MOVECAL(RDL)	ONE OR BOTH DCBS ARE NOT OPEN
00	1A	03	MOVECAL(RDL)	DCB1 NOT OPEN IN OR DCB2 NOT OPEN OUT
00	1A	04	MOVECAL(RDL)	MOVE CAL NOT ALLOWED FOR DEVICE OR ANS DCBS
00	1A	4A	MOVECAL(RDL)	SPECIFIED BUFFER DOES NOT BELONG TO USER
00	1C	00	READ/WRITE/PRECORD	END OF TAPE REACHED.
00	1C	01	WRITE	END OF TAPE ON COMMON JOURNAL
00	1D	00	READ/PRECORD	BEGINNING OF TAPE REACHED

GRP	ERR	SUB	MODULE	MESSAGE
00	1F	00	WRT/IOD/IORT	BIN (OR VFC) NOT VALID FOR THIS DEVICE
00	20	01	READ	PRIVATE PACK IS LOCKED OUT.
00	20	02	READ	EXCLUSIVE USE OF PRIVATE PACK, NOT THIS USER.
00	20	03	READ	PRIVATE PACK NOT PROPERLY ASKED FOR.
00	20	04	RDF	ONLINE USER REQUEST FOR PARTITIONED PRIV. PACK
00	20	05	PV	PACK SET HAS TWO PRIMARY VOLUMES
00	21	00	OPEN/CLOSE	PRIVATE PACK CONSISTENCY CHECK FAILURE.
00	22	00	OPEN	PRIVATE PACK ERROR TRYING TO OPEN EXISTING FILE.
00	2E	00	OPEN	THAT DCB IS ALREADY OPEN!
00	30	01	LBLT	BAD USER LABEL ON ANS TAPE.
00	30	03	LBLT	INVALID FILE NAME LENGTH.
00	30	04	LBLT	CAN'T SPECIFY EXPIRE.NEVER.
00	30	05	LBLT	ILLEGAL FORMAT CODE.
00	31	00	ENQ/ENQO	DEQUEUE ATTEMPTED ON A RESOURCE-ELEMENT NOT QUEUED
00	31	01	ENQ/ENQO	ENQUEUE ATTEMPTED ON A RESOURCE-ELEMENT ALREADY QUEUED
00	31	02	ENQ/ENQO	ENQUEUE(SHARE) ATTEMPTED ON ELEMENT ALREADY QUEUED(EXCL)
00	31	03	ENQ/ENQO	ENQUEUE ATTEMPTED ON RESOURCE ELEMENT NOT PRESENTLY AVAILABLE
00	31	04	ENQ/ENQO	ENQUEUE ABORTED BY CONTROL-Y OR BREAK OR OPERATOR
00	32	00	MOCIOP	MULTI-POINT LINE BUFFER SIZE EXCEEDS 2049 BYTES
00	32	01	MOCIOP	UNABLE TO ASSIGN PHYSICAL WORK PAGE: LIMIT REACHED
00	32	02	MOCIOP	UNABLE TO ASSIGN PHYSICAL WORK PAGE: PAGE NOT OWNED BY USER
00	32	03	OPND	POLLING/SELECTION LIST ALREADY IN USE
00	32	04	OPND/MOCIOP	BAD POLLING SELECTION LIST ADDRESS
00	32	05	OPND	UNABLE TO INITIALIZE SLAVE LINE: BAD SLAVE LINE ID
00	32	06	OPND	ATTEMPTED TO OPEN DCB TO UNOWNED SLAVE LINE
00	32	07	MOCIOP	NUMBER OF POL/SEL LIST COMPONENTS EXCEEDS 254
00	32	08	MOCIOP	END OF LIST HIT ON OPEN LIST DURING AUTOMATIC POLLING OPERATION
00	32	09	OPND	MOCIOP MODULE OR COC SLAVE LINE CODE NOT IN THIS SYSTEM
00	32	70	MOCIOP	STATUS INSTEAD OF DATA IN BUFFER ON MULTI-POINT.
				LINE READ
00	33	00	IOCHEK	READ OR WRITE ON BI-POINT LINE TERMINATED :
				BY PURGE CAL
00	3F	36	JOBENT	JOB ENTRY DISALLOWED BY OPERATOR
00	3F	37	JOBENT	USER NOT ALLOWED TO USE JOB ENTRY SERVICE.
00	3F	38	JOBENT	FUNCTION INCONSISTENCY.
00	3F	39	JOBENT	INVALID ID FOR DELETION.
00	3F	3A	JOBENT	TOO LATE TO DELETE JOB.
00	3F	3B	JOBENT	NO MORE SYMBIONT SPACE IS AVAILABLE OR THE QUEUE IS FULL
00	3F	3C	JOBENT	USER NOT ALLOWED TO USE JOB ENTRY SERVICE.
00	3F	3D	JOBENT	SYSTEM OR LL DEVICE NOT SYMBIONT.
00	3F	3E	JOBENT	THE SPECIFIED DCB IS ALREADY OPEN
00	3F	3F	JOBENT	THE SPECIFIED BUFFER ADDRESS IS NOT IN YOUR PROGRAM.
00	40	00	READ	CAN'T READ AN OUTPUT FILE.
00	41	00	READ	IRRECOVERABLE DEVICE READ ERROR
00	41	01	COOP	INPUT SYMBIONT FILE LOST BAD READ.
00	41	02	READ	LABELED TAPE READ ERROR.
00	41	03	READ	LABELED TAPE READ ERROR RECORD NOT TRANSMITTED.
00	41	04	READ	PARTIAL RECORD TRANSMITTED FOLLOWING ERROR 4103
00	42	00	READ/WRITE/RANDOM	BAD KEY. LENGTH IS ZERO OR GREATER THAN MAX.
00	43	00	READ	CAN'T FIND A RECORD WITH THAT KEY.
00	44	00	WRITE	CAN'T WRITE AN INPUT FILE.
00	45	00	WRITE	IRRECOVERABLE DEVICE WRITE ERROR.
00	45	01	WRITE	WRITE ERROR ON COMMON JOURNAL
00	46	00	READ	CAN'T OPEN FOR READ
00	46	01	READ	CAN'T OPEN FOR READ. DCB HASN'T ENOUGH INFORMATION.
00	46	03	READ	CAN'T OPEN FOR READ FILE DOESN'T EXIST.
00	46	14	READ	CAN'T OPEN FOR READ: PASSWORD, ACCOUNT RESTRICTION, OR ANOTHER USER.

GRP	ERR	SUB	MODULE	MESSAGE
00	46	21	READ/WRITE	PRIVATE PACK LOGIC INCONSISTENCY.
00	46	22	READ	PRIVATE PACK ERROR TRYING TO OPEN EXISTING FILE.
00	46	48	READ	ONLINE USER CANNOT ACCESS CARD READER.
00	47	00	WRITE	CAN'T OPEN FOR WRITE.
00	47	01	WRITE	CAN'T OPEN FOR WRITE: DCB HASN'T ENOUGH INFORMATION.
00	47	03	WRITE	CAN'T OPEN FOR WRITE: FILE DOESN'T EXIST.
00	47	14	WRITE	CAN'T OPEN FOR WRITE: PASSWORD, ACCOUNT RESTRICTION, OR ANOTHER USER.
00	47	2B	OPEN	INVALID OP LABEL IN DCB.
00	47	48	WRITE	SYMBIONT USE FLAG NOT SET FOR ONLINE USER.
00	48	00	OPNLD	PERIPHERAL USE FLAG NOT SET FOR THIS DEVICE
00	48	01	OPEN	ONLINE USER CANNOT ACCESS CARD READER.
00	49	00	OPEN	YOUR FILE IS WRITE PROTECTED.
00	49	01	OPEN	REQUESTED REMOVABLE DEVICE IS NOT AVAILABLE
00	49	02	PV-OPNTP	USER DISMOUNTABLE RESOURCE LIMIT EXCEEDED
00	4A	00	READ/WRITE	SPECIFIED BUFFER DOES NOT BELONG TO THE USER.
00	4A	01	IOCHEK	TIME PARAMETER ON M:CHECKECB TOO BIG
00	4A	02	IOCHEK	ECB IN WRONG STATE ON M:CHECKECB OR M:CHECK W/ECB
00	4A	03	IOCHEK	INFINITE WAIT CONDITION ON M:CHECKECB OR M:CHECK W/ECB
00	4A	04	IOCHEK	NO MONITOR WORK SPACE TO HONOR M:CHECKECB OR M:CHECK W/ECB
00	4A	05	IOCHEK	WRONG ACCESS CODE FOR ECB ADDRESS ON M:CHECKECB OR M:CHECK W/ECB
00	4B	00	READ/WRITE	ATTEMPT TO OPEN A FILE THAT YOU HAVE OPEN ALREADY.
00	4C	00	READ/WRITE	ATTEMPT TO OPEN A FILE THAT SOMEONE ELSE HAS OPEN.
00	4D	00	CLOSE	ATTEMPT TO CLOSE AND RELEASE A FILE THAT SOMEONE ELSE IS READING.
00	4E	00	ARDL	ANS BLOCK COUNT ERROR AND NO ABCERR SPECIFIED.
00	4E	05	READ/CVOL	ANS BLOCK COUNT ERROR PLUS END OF TAPE.
00	4E	07		ANS BLOCK COUNT ERROR PLUS END OF FILE.
00	4F	00	AGTAPE	UNRECOVERABLE ERROR AFTER REFLECTOR ON TAPE.
00	51	00	OPEN	YOU STILL HAVE THIS FILE OPEN IN THROUGH ANOTHER DCB.
00	54	00	READ	YOU MUST NOT READ MONITOR CONTROL CARDS MORE THAN ONCE.
00	55	00	OPEN	TOO MANY FILES OPEN SIMULTANEOUSLY.
00	56	00	CLOSE/CVOL	THE RAD IS FULL OR NO NEXT REEL NUMBER FOR THIS TAPE VOLUME.
00	57	00	READ/WRITE	RAD SATURATED OR NO NEXT REEL NUMBER FOR THIS TAPE VOLUME.
00	57	44	RANDOM	NOT ENOUGH GRANULES FOR WRITE OR COUNT ON READ EXTENDS BEYOND EOF.
00	58	00	ENQ/ENQO	ENQUEUE REQUEST WOULD RESULT IN A DEADLOCK
00	58	01	ENQ/ENQO	ENQUEUE TABLES FULL
00	58	02	ENQ/ENQO	ENQUEUE REQUEST FOR ALL WITH OTHER ELEMENTS ALREADY QUEUED
00	58	03	ENQ/ENQO	UNAUTHORIZED USER ATTEMPTED AN ENQUEUE
00	59	00	IORT	ATTEMPTED READ ON BI-POINT LINE WITH A READ OUTSTANDING
00	59	01	MOCIOP	INVALID POL/SEL LIST INDEX VALUE
00	59	02	MOCIOP	POL/SEL LIST INDEX VALUE POINTS TO EXCLUDED OR DUMMY COMPONENT
00	59	03	MOCIOP	AUTO READ REQUEST CONFLICTS WITH OUTSTANDING I/O REQUESTS
00	59	04	MOCIOP	MULTI-POINT LINE CONTROLLER DOWN
00	59	05	MOCIOP	NO RESPONSE FROM MULTI-POINT LINE TERMINAL(READ)
00	59	06	MOCIOP	MULTI-POINT LINE TERMINAL NOT IN TRANSMIT STATE ON SINGLE TERMINAL READ: REQUEST OR HALT SPECIFIED OR END OF OPEN LIST HIT.
00	59	07	MOCIOP	BAD TRAILER RECORD ON MULTI-POINT TERMINAL READ
00	59	08	MOCIOP	PARITY ERROR IN RECORD RECEIVED FROM MULTI-POINT LINE TERMINAL
00	59	09	MOCIOP	BAD HEADER RECORD ON MULTI-POINT TERMINAL READ
00	59	0A	MOCIOP	NO RESPONSE FROM MULTI-POINT LINE TERMINAL (WRITE)
00	59	0B	MOCIOP	ILLEGAL 1ST CHARACTER IN WRITE SELECTION RESPONSE HEADER FROM: MULTI-POINT LINE TERMINAL
00	59	0C	MOCIOP	MULTI-POINT LINE TERMINAL NOT IN RECEIVE STATE ON WRITE REQUEST
00	59	0D	MOCIOP	INTERNAL TERMINAL ERROR ON MULTI-POINT READ
00	59	0E	MOCIOP	MULTI-POINT LINE TERMINAL NAK RECORD (WRITE DATA RESPONSE)
00	59	0F	MOCIOP	LOST DATA ON MULTI-POINT LINE TERMINAL WRITE
00	59	10	MOCIOP	ILLEGAL RESPONSE ON MULTI-POINT LINE READ
00	5A	00	IORT	NO ECB ADDRESS SUPPLIED WITH SLAVE LINE I/O REQUEST

GRP	ERR	SUB	MODULE	MESSAGE
00	5A	01	IOCHEK	NO ECB ADDRESS SUPPLIED WITH SLAVE LINE M:CHECK
00	5B	00	IOCHEK	LINE HUNG UP ON READ PENDING (BI-POINT LINE)
00	5C	00	IOCHEK	LINE HUNG UP ON INCOMPLETE WRITE (BI-POINT LINE)
00	75	00		BAD DISC ADDRESS IN FREE SECTOR POOL (DIRECTORIES ONLY). ERRLOG ONLY-
00	75	01	READ	DATA RECORDS LOST
00	75	02	READ	FILE INDEX (AND THE ASSOCIATED DATA RECORDS) LOST.
00	75	03	OPEN	ENTIRE FILE LOST.
00	75	04	CLS/OPEN	ERROR IN FILE DIRECTORY. ONE OR MORE FILES ARE INACCESSIBLE.
00	75	05	OPEN	ENTIRE ACCOUNT LOST
00	75	06	OPEN	ACCOUNT DIRECTORY (AND THE ASSOCIATED ACCOUNTS) LOST.
00	75	07		ERROR IN PYRAMID (ERRLOG ONLY)
00	75	40		7500 PLUS HARDWARE ERROR.
00	75	41		7501 PLUS HARDWARE ERROR.
00	75	42		7502 PLUS HARDWARE ERROR.
00	75	43		7503 PLUS HARDWARE ERROR.
00	75	44		7504 PLUS HARDWARE ERROR.
00	75	45		7505 PLUS HARDWARE ERROR.
00	75	46		7506 PLUS HARDWARE ERROR.
00	75	47		7507 PLUS HARDWARE ERROR.
00	7F	1D	INITRCVR	SINGLE USER ABORT. SC 1D.
00	7F	21	INITRCVR	SINGLE USER ABORT. SC 21.
00	7F	22	INITRCVR	SINGLE USER ABORT. SC 22.
00	7F	31	INITRCVR	SINGLE USER ABORT. SC 31.
00	7F	32	INITRCVR	SINGLE USER ABORT. SC 32.
00	7F	49	INITRCVR	SINGLE USER ABORT. SC 49.
00	7F	61	INITRCVR	SINGLE USER ABORT. SC 61.
00	7F	6A	INITRCVR	SINGLE USER ABORT. SC 6A.
00	7F	79	INITRCVR	SINGLE USER ABORT. SC 79.
00	7F	7C	INITRCVR	SINGLE USER ABORT SC 7C.
00	7F	7E	INITRCVR	SINGLE USER ABORT SC 7E.
00	A0	00	STEP(ASP)	THAT'S NO DEBUGGER!
00	A1	00	STEP(ASP)	DON'T TRY TO DEBUG A SHARED PROCESSOR
00	A1	01	STPNR	DON'T TRY TO EXECUTE AN EXECUTE ONLY LOAD MODULE
00	A2	00	STEP(ASP)	ATTEMPT TO ACCESS UNAUTHORIZED PROCESSOR(E.E. CCI ON-LINE).
00	A2	01	STPNR	ACCESS TO NON-SYSTEM PROCESSOR DENIED BY INSTALLATION
00	A2	02	STPNR	ACCESS TO PROCESSOR DENIED BY INSTALLATION
00	A3	00	TRAP	CAN'T GIVE YOU THIS TRAP. YOUR STACK IS FULL OR NOT THERE.
00	A3	01	TRAPC	NO ENVIRONMENT PRESENT FOR RETURN.
00	A3	02	TRAPC	DON'T SIMULATE THAT TRAP
00	A4	00		YOU TRAPPED
00	A4	01	TRAP	NONEXISTENT INSTRUCTION
00	A4	02	TRAP	NONEXISTENT MEMORY REFERENCE
00	A4	03	TRAP	PRIVILEGED INSTRUCTION
00	A4	04	TRAP	MEMORY PROTECT VIOLATION
00	A4	05	TRAP	UNIMPLEMENTED INSTRUCTION
00	A4	06	TRAP	STACK OVERFLOW
00	A4	07	TRAP	FIXED POINT OVERFLOW
00	A4	08	TRAP	FLOATING POINT FAULT
00	A4	09	TRAP	DECIMAL ARITHMETIC FAULT
00	A4	0A	TRAP	WATCHDOG TIMER RUNOUT. STEP ABORTED.
00	A4	0B	TRAP	TRAP 47
00	A4	0C	CSEHAND	HARDWARE ERROR - BUS CHECK FAULT
00	A4	0D	CSEHAND	INSTRUCTION EXCEPTION TRAP - ILLEGAL REGISTER DESIGNATION
00	A4	0E	CSEHAND	HARDWARE ERROR - PARITY ERROR IN [illegible] OR REGISTERS
00	A4	0F	CSEHAND	HARDWARE ERROR - PARITY ERROR IN WRITE LOCK REGISTERS
00	A5	00	STPNR	LOAD MODULE SIZE EXCEEDS USER LIMIT OR AVAILABLE CORE.
00	A5	02	STPNR	VIRTUAL CORE NOT AVAILABLE SPECIAL SHARED PROCESSOR

GRP	ERR	SUB	MODULE	MESSAGE
00	A5	04	STPNR	EXTENDED MEMORY MODE-CURRENT JOB STEP ABORTED
00	A5	06	STPNR	CURRENT SPECIAL SHARED PROCESSOR ABORTED FOR TEL
00	A5	07	STPNR	PROGRAM OVERLAPS CURRENTLY ALLOCATED COMMON PAGES
00	A5	08	STPNR	PHYSICAL CORE NOT AVAILABLE FOR SHARED PROCESSOR
00	A5	09	STPNR	CORE NOT AVAILABLE FOR CONTEXT BUFFER
00	A5	51	STPNR	BAD DATA BIAS FOR CORE LIBRARY
00	A6	03	STPNR	LOAD MODULE DOES NOT EXIST
00	A6	30	STPNR	BAD DCBS OR DCB TABLE
00	A6	31	STPNR	BAD HEAD RECORD
00	A6	32	STPNR	LOAD MODULE BIAS NOT ON PAGE BOUNDARY
00	A6	33	STPNR	PURE PROCEDURE NOT ON PAGE BOUNDARY
00	A6	34	STPNR	DCBS NOT ON PAGE BOUNDARY
00	A6	35	STPNR	HEAD RECORD IS INCOMPLETE
00	A6	36	STPNR	TREE RECORD IS INCOMPLETE
00	A6	37	STPNR	NO DEBUGS ARE ALLOWED WITH LINK-BUILD LMNS
00	A6	38	STPNR	THE PROGRAM IS TOO BIG FOR THE ALLOWED USER AREA
00	A6	39	STPNR	FILE NOT KEYED. NOT A LOAD MODULE
00	A6	3A	STPNR	YOUR DCB CHAIN IS UNLINKED OR CIRCULAR
00	A6	3B	STPNR	TCB ADDRESS NOT WITHIN DATA AREA
00	A6	42	STPNR	THAT'S NO LOAD MODULE
00	A6	43	STPNR	THAT'S NO LOAD MODULE
00	A6	46	STPNR	CAN'T FIND REQUESTED PROGRAM
00	A6	50	STPNR	LOAD MODULE IS PRE-C00
00	A6	51	STPNR	SNAP/MODIFY NOT ALLOWED WITH EXECUTE ONLY LOAD MODULES
00	A7	00	TEL	FINI
00	A8	C0	STPNR	YOU ISSUED AN ERROR OR ABORT CAL
00	A9	00	UCAL	ERROR ON READ OR WRITE OF A/M RECORD
00	AA	00	STPNR	WHAT PUBLIC LIBRARY?
00	AC	00		DON'T TRY TO READ THE CARD READER
00	AD	00	STPNR	EXTENDED PROCESSING LIMITS WERE EXCEEDED.
00	AE	00	CALPROC/ALTCP	THAT CAL HAS AN ILLEGAL OP CODE.
00	AF	00	CALPROC	YOUR CAL1 REFERENCES A NON-EXISTENT DCB.
00	B0	00	DUMP	NO M:DO DCB FOR SNAPSHOT
00	B0	01	DUMP	ATTEMPT TO DUMP AN INACCESSABLE LOCATION
00	B0	02	DUMP	INACCESSIBLE FLAG ADDRESS GIVEN ON CONDITIONAL DEBUG
00	B0	03	DUMP	ILLEGAL PARAMETER IN DEBUG CAL.
00	B1	00	SEGLOAD	CAN'T FIND THAT OVERLAY SEGMENT.
00	B1	01	SEGLOAD	TREE RECORD IS INCOMPLETE
00	B1	02	SEGLOAD	TREE RECORD IS CIRCULAR
00	B1	03	SEGLOAD	DATA AREA IS LARGER THAN RESERVED BY HEAD
00	B1	04	SEGLOAD	PP AREA IS LARGER THAN RESERVED BY HEAD
00	B1	05	SEGLOAD	BUFFER IS BAD IN THE SEGLD DCB
00	B1	06	SEGLOAD	OUT OF PHYSICAL PAGES (SEGLD ERROR)
00	B1	07	SEGLOAD	PAGE ACQUIRED BY M:GVM ENCOUNTERED
00	B1	08	SEGLOAD	PAGED LOAD MODULE GREATER THAN 255 SEGMENTS.
00	B2	00	ENTRY	DON'T USE CAL2,CAL3,OR CAL4 INSTRUCTIONS
00	B3	00	WRTD	LIMIT EXCEEDED.
00	B3	01	WRTD	PUNCH LIMIT
00	B3	02	WRTD	PAGE LIMIT FOR PROCESSOR
00	B3	03	WRTD	PAGE LIMIT FOR USER
00	B3	04	WRTD	PAGE LIMIT FOR DEBUGGING
00	B3	08	WRTD	EXECUTION TIME LIMIT.
00	B4	00	STPNR	EXIT
00	B4	01	STPNR	YOU ISSUED A M:ERR
00	B4	02	STPNR	YOU ISSUED A M:XXX
00	B4	03	STPNR	OPERATOR ERRORED YOU.
00	B4	04	STPNR	JOB ABORTED BY OPERATOR OR USER

GRP	ERR	SUB	MODULE	MESSAGE
00	B5	00	STPNR	LOAD MODULE SIZE EXCEEDS USER LIMIT OR AVAILABLE CORE
00	B5	01	LDLNK	LOAD AND LINK CAN T OPEN YOUR FILE
00	B5	02	STEPNR	VIRTUAL CORE NOT AVAILABLE FOR SPECIAL SHARED PROCESSOR
00	B5	03	LDLNK	LOAD AND LINK CAN T FIND YOUR FILE
00	B5	07	STPNR	PROGRAM OVERLAPS CURRENTLY ALLOCATED COMMON PAGES
00	B5	08	STPNR	PHYSICAL CORE NOT AVAILABLE FOR SHARED PROCESSOR
00	B5	09	STPNR	CORE NOT AVAILABLE FOR CONTEXT BUFFER
00	B5	14	LDLNK	LOAD AND LINK IS DENIED ACCESS TO YOUR FILE
00	B5	46	LDLNK	LOAD AND LINK HAS INSUFFICIENT INFORMATION TO OPEN YOUR FILE.
00	B5	50	STPNR	LOAD MODULE IS PRE-B00
00	B5	51	STPNR	BAD DATA BIAS FOR CORE LIBRARY
00	B5	61	LDLNK	LOAD AND LINK IS NOT PERMITTED WHILE RUNNING UNDER A DEBUGGER
00	B5	62	LDLNK	LOAD AND LINK IS NOT PERMITTED WHEN A SPECIAL PROCESSOR OTHER: THAN A CORE LIBRARY IS ASSOCIATED.
00	B5	63	LDLNK	LOAD AND LINK IS PERMITTED ONLY WITH PROGRAMS CREATED BY OVERLAY: LOADER.
00	B5	64	LDLNK	LOAD AND LINK IS NOT PERMITTED ON PROGRAMS NOT OWNING ALL VIRTUAL: MEMORY FROM DATA THROUGH DYNAMIC DATA.
00	B5	65	LDLNK	LOAD AND LINK FOUND A DCB WHICH IS NOT IN THE DCB PAGES
00	B5	66	LDLNK	LOAD AND LINK COULD NOT FIND OR RELEASE A VALID BLOCKING BUFFER
00	B5	67	LDLNK	INVALID ATTEMPT TO PERFORM LOAD AND LINK EXIT AT JOB STEP
00	B5	68	LDLNK	LOAD AND LINK OPERATION IS ILLEGAL AS SPECIFIED
00	B5	69	LDLNK	LDLNK CLEANUP WITHOUT PREVIOUS LDLNK OPERATION.
00	B5	6B	STPNRN	LOAD AND LINK TO LINKED PROGRAM NOT ALLOWED
00	B5	6C	STPNR	LOAD AND LINK TO SPECIAL SHARED PROCESSOR NOT ALLOWED
00	B5	6D	LNKTRC	PHYSICAL CORE FOR LIBRARY NOT AVAILABLE FOLLOWING LNKTRC
00	B6	00	LDLNK	THAT'S NO SEGLD DCB
00	B6	01	STPNR	THE DCB NAME CHAIN IS NOT IN THE DCB RECORD
00	B6	02	STPNR	THE DCB NAME CHAIN MAY NOT BE LINKED
00	B6	03	STPNR	THE DCB NAME CHAIN IS IRREGULAR
00	B6	04	STPNR	THE DCB HAS NO NAME
00	B6	05	STPNR	USER HAS MORE THAN 509 DCBS
00	B6	06	STPNR	THE DCB IS OUTSIDE THE BUFFER
00	B6	07	STPNR	THE DCB MAY NOT CROSS PAGE BOUNDARIES
00	B6	08	STPNR	THE DCB MUST BE AT LEAST 22 WORDS LONG
00	B6	09	STPNR	KBUF MUST LIE WITHIN THE DCB
00	B6	0A	STPNR	FLP MUST LIE WITHIN THE DCB
00	B6	0B	STEP	FLP OVERLAPS INTO KBUF
00	B6	0C	STEP	M:SEGLD DCB NEEDS 10 WORDS FOR VLPS
00	B7	00	OPNLD	UNRECOGNIZED STREAM ID
00	B7	01	OPNLD	UNRECOGNIZED DEV SPECIFICATION
00	B7	02	OPNLD	FUNCTION NOT LEGAL FOR DEVICE
00	B7	03	OPNLD	UNRECOGNIZED WSN
00	B7	04	OPNLD	PERIPHERAL USE FLAG NOT SET FOR THIS DEVICE
00	B8	00	RTROO	THIS ERROR CODE IS RESERVED FOR REAL-TIME PURPOSES
00	B8	01	RTROOT	CFI ATTTEMPED WITH NO ICB'S ASSOCIATED
00	B8	02	RTROOT	INTRTN ISSUED WHEN NO ICB'S WERE ACTIVE
00	B8	03	RTROOT	RESTRICTED CAL1 ISSUED FOLLOWING M:HOLD (ON)
00	B8	04	RTNR	USER PROVIDED AN ILLEGAL OR NON-EXISTANT INTERRUPT ADDRESS OR TEXT LABEL
00	B8	05	RTNR	ILLEGAL FPT OR MISSING REQUIRED PARAMETER
00	B8	06	RTNR	ZERO TIMER UNITS SPECIFIED VIA M:CLOCK CAL1
00	B8	07	T:JOBENT/GRAN	USER REQUESTED A SERVICE FROM A SYSTEM GHOST AFTER HAVING BLOCKED THAT GHOST VIA M:HOLD (ON)
00	B9	00	RTROOT	THIS ERROR CODE IS RESERVED FOR REAL-TIME PURPOSES
00	B9	01	ALTCP/RTROOT	USER HAS INSUFFICIENT PRIVILEGE TO ISSUE THIS CAL1.5
00	B9	02	RTROOT	INVALID DEVICE ADR. DEVICE NOT PRE-EMPTED, OR SPECIFIED DCB : IS NOT OPENED PROPERLY (M:IOEX)

GRP	ERR	SUB	MODULE	MESSAGE
00	B9	04	ALTCP	EFFECTIVE ADDRESS OF NNEXU IS IN PROTECTED MEMORY
00	B9	05	ALTCP	INSTRUCTION TO BE EXECUTED VIA AN EXU HAD INVALID OP CODE
00	BC	01	TQ	TP SERVICE REQUESTED IS ILLEGAL FOR THIS USER
00	BC	03	TQ	ERROR RETURN FROM GET PHYSICAL WORK PAGE
00	BC	07	TQ	QUEUE SATURATED (INDEX CORE SPACE OR SECONDARY STORAGE UNAVAILABLE)
00	BC	08	TQ	QUEUE LOCK/UNLOCK CALLER DOES NOT HAVE THE REQUIRED PRIVILEGE
00	BC	09	TQ	DCB NOT OPEN FOR A LOCK/UNLOCK REQUEST
00	BC	0A	TQ	SPACE IS NOT AVAILABLE TO DEFINE A LIST
00	BC	10	TQ	BAD MEMORY ADDRESS
00	BC	11	TQ	QUEUE LOCKED
00	BC	12	TQ	QUEUE PHYSICAL PAGE SPACE IS NOT AVAILABLE
00	BC	13	TQ	ERROR IN PRESENTED LIST FOR A PUT, DEFINELIST, OR STATS
00	BC	14	TQ	ENTRY NOT FOUND FOR A QUEUE REQUEST REQUIRING AN EXISTING ENTRY
00	BC	15	TQ	I/O ERROR DURING CONTROL/INDEX TRANSFER FOR AN UNLOCK REQUEST
00	BC	16	TQ	I/O ERROR DURING A DATA BLOCK TRANSFER
00	BC	17	TQ	QUEUE BUSY
00	BC	20	TQ	QUEUE GET OR PURGE REQUEST FOR A NON-EXISTENT GET LIST
01	00	64	PCL	ARGUMENT GREATER THAN 31 CHARACTERS
01	00	C8	PCL	ILLEGAL DEVICE CODE
01	01	2C	PCL	INVALID REEL NUMBER SPECIFICATION
01	01	90	PCL	ILLEGAL FILE NAME SPECIFICATION
01	01	F4	PCL	ILLEGAL ACCOUNT NUMBER SPECIFICATION
01	02	58	PCL	ILLEGAL PASSWORD SPECIFICATION
01	02	BC	PCL	TOO MANY FIELDS IN A FILE IDENTIFICATION SPECIFICATION
01	03	20	PCL	INVALID FILE RANGE SPECIFICATION
01	03	84	PCL	MORE THAN TEN RS FIELDS FOR AN INPUT DEVICE
01	03	E8	PCL	OVERFLOW ON A RS VALUE
01	04	4C	PCL	ERROR ON Y VALUE OF RS OPTION
01	04	B0	PCL	CS ID-FIELD GREATER THAN FOUR CHARACTERS
01	05	14	PCL	ERROR ON N OR K VALUE OF CS OPTION
01	05	78	PCL	IMPROPER TERMINATION WITHIN RS, LN, OR CS OPTION
01	05	DC	PCL	)) MUST TERMINATE RS, LN, OR CS OPTION
01	06	40	PCL	SPECIAL ARGUMENTS MUST HAVE) AS TERMINATION CHARACTER
01	06	A4	PCL	EH?
01	07	08	PCL	UNDEFINED COMMAND
01	07	6C	PCL	ILLEGAL INPUT DEVICE
01	07	D0	PCL	NO DEFINED OUTPUT DEVICE
01	08	34	PCL	ILLEGAL OUTPUT DEVICE
01	08	98	PCL	REEL NUMBER SPECIFICATION NOT VALID FOR THIS DEVICE
01	08	FC	PCL	FILE SPECIFICATION NOT VALID FOR THIS DEVICE
01	09	60	PCL	DATA CODE SPECIFICATION NOT VALID FOR THIS DEVICE
01	09	C4	PCL	MODE SPECIFICATION NOT VALID FOR THIS DEVICE
01	0A	28	PCL	SEQUENCE SPECIFICATION NOT VALID FOR THIS DEVICE
01	0A	8C	PCL	RECORD SELECTION SPECIFICATION NOT VALID FOR THIS DEVICE
01	0A	F0	PCL	PK/BIN/7T COMBINATION NOT VALID
01	0B	54	PCL	NULL ARGUMENT (TWO DELIMITERS IN A ROW)
01	0B	B8	PCL	IMPROPER TERMINATION OF THE COMMAND
01	0C	1C	PCL	ONE REEL NUMBER MUST BE SPECIFIED ON THIS COMMAND
01	0C	80	PCL	'TO' OR 'OVER' NOT SPECIFIED
01	0C	E4	PCL	RECORD SIZE EXCEEDS AVAILABLE MEMORY
01	0D	48	PCL	INVALID DEVICE TYPE FOR THIS COMMAND
01	0D	AC	PCL	TOO MANY REEL NUMBERS SPECIFIED
01	0E	10	PCL	OVERFLOW ON NUMBER OF FILES ON 'SPF' COMMAND
01	0E	74	PCL	INVALID DIRECTION INDICATOR ON 'SPF' COMMAND
01	0E	D8	PCL	INPUT RECORD SIZE LARGER THAN 32767 BYTES
01	0F	3C	PCL	INVALID OPTION FOR COPYALL
01	0F	A0	PCL	ACCOUNT SPECIFICATION NOT VALID ON 'SPF' COMMAND

GRP	ERR	SUB	MODULE	MESSAGE	
01	10	04	PCL	RS SPECIFICATION BEYOND END OF FILE	
01	10	68	PCL	ERROR IN COMPRESSED INPUT	
01	10	CC	PCL	C OPTION INVALID ON BOTH INPUT AND OUTPUT	
01	11	30	PCL	RECORD SELECTION INVALID WITH C OPTION	
01	11	94	PCL	INVALID TAB SPECIFICATION	
01	11	F8	PCL	OVERFLOW ON EDIT LINE NUMBER	
01	12	5C	PCL	ZERO INCREMENT ON CS OR LN OPTION	
01	12	C0	PCL	TX OPTION USED WITHOUT TABS COMMAND	
01	13	24	PCL	INVALID OPTION FOR COPYSTD	
01	13	88	PCL	MORE THAN EIGHT READ WRTIE OR EXECUTE ACCOUNTS	
01	13	EC	PCL	MORE THAN 16 TAB VALUES	
01	14	50	PCL	UNABLE TO DISMOUNT	
01	14	B4	PCL	TOO MANY CHARACTERS IN THE COMMAND	
01	15	18	PCL	INVALID VALUE FOR ANS OPTION	
01	15	7C	PCL	FORMAT CODE INVALID FOR COMMAND	
01	15	E0	PCL	INVALID OPTION FOR ANS TO ANS COPY	
01	16	00	PCL	TAPE DENSITY SPECIFICATION IS IN ERROR	
02	00	01	LOADER	UNEXPECTED EOF	
02	00	02	LOADER	ILLEGAL RECORD I.D.	
02	00	03	LOADER	SEQUENCE ERROR	
02	00	04	LOADER	ILLEGAL RECORD SIZE	
02	00	05	LOADER	CHECKSUM ERROR	
02	00	06	LOADER	ABNORMAL I/O	
02	00	07	LOADER	CANNOT OPEN E.F.	
02	00	08	LOADER	STACK OVERFLOW	
02	00	09	LOADER	BIAS TO LARGE	
02	00	0A	LOADER	ILL. ROM LANGUAGE	
02	00	0B	LOADER	BAD START ADDRESS	
02	00	0C	LOADER	UNEXPECTED ROM END	
02	00	0D	LOADER	REPEAT LOAD IS ZERO	
02	00	0E	LOADER	IMPROPER BOUND	
02	00	0F	LOADER	ILLEGAL ORG	
02	00	10	LOADER	BAD I/O RETURN FROM M:LM DCB	
02	00	11	LOADER	SEV. LEV. EXCEEDED	
02	00	12	LOADER	ILL. LIB. LOAD MOD.	
02	00	13	LOADER	NO ROOM TO ROUND DCBS TO PAGE BOUNDARIES. TRY FORCING XMEM	
02	00	14	LOADER	ILL. DSECT	
02	00	15	LOADER	ROOT SEGMENT TOO LARGE TO LOAD	IN2
02	00	16	LOADER	CANNOT ENTER XMEM NO ROOM FOR CORE IMAGE BUFFER	IN2
02	00	17	LOADER	CANNOT ENTER XMEM. STACKS TOO LARGE.	IN2
02	00	18	LOADER	NOT ENOUGH ROOM TO CONCATENATE XMEM PAGES	IN2
02	00	19	LOADER	NO ROOM TO READ LIBRARY CORE IMAGE	EVL
02	00	1A	LOADER	NO ROOM TO READ LIBRARY RELOCATION DICTIONARY	EVL
02	00	1B	LOADER	NO ROOM FOR NEW EXPRESSION	WRT
02	00	1C	LOADER	NO ROOM TO BUILD DCB TABLE TRY FORCING XMEM	WRT
02	00	1D	LOADER	NO ROOM TO BUILD DCB TABLE	WRT
02	00	1E	LOADER	LIBRARY LOAD MODULE REF/DEF STACK TOO LARGE TO UPDATE	WRT
02	00	1F	LOADER	INSUFFICIENT PHYSICAL MEMORY	
02	00	20	LOADER	BAD ASSIGN MERGE RECORD	
02	00	21	LOADER	NO ROOM TO ADD LIBRARY LOAD MODULE TO ROM TABLE	
02	00	22	LOADER	NO ROOM TO READ LIBRARY REF/DEF STACK	
02	00	23	LOADER	NO ROOM TO UPDATE LIBRARY	
02	00	24	LOADER	INVALID KEY SUPPLIED FOR DELETE RECORD ON M:DIC	
02	00	25	LOADER	I/O ERROR ON M:DIC IN WRITEF.	
02	00	26	LOADER	ILLEGAL LIBRARY LOAD MODULE NAME	
02	00	27	LOADER	ABNORMAL I/O ON OPEN OR READ TO CORE LIBRARY	
02	00	28	LOADER	INVALID DECLARATION NUMBER REFERENCE (BAD ROM).	

GRP	ERR	SUB	MODULE	MESSAGE	
02	00	29	LOADER	INVALID KEY SUPPLIED FOR WRITE RECORD ON M:DIC	
02	00	2A	LOADER	ILLEGAL LOADER TRAP	
02	00	2B	LOADER	ABNORMAL I/O IN WRITELIB	
02	00	2C	LOADER	CANNOT FIND REF/DEF NAME IN STACK	
02	00	2D	LOADER	LIB LOAD MODULE TOO BIG - CANNOT USE EXTENDED MEMORY	
02	00	2E	LOADER	LIB LMN IS NOT ALLOWED ON A PRIVATE VOLUME	
02	00	2F	LOADER	ABNORMAL I/O READING LIB LMN	EVL
02	00	30	LOADER	PAGED LMN MUST NOT HAVE MORE THAN 256 SEGMENTS	EVL
02	00	31	LOADER	EXISTING LMN CANNOT BE MAPPED--X'85'	
04	03	58	RUNNER	BAD FROM OR TO NAME	
04	03	59	RUNNER	BAD IF.AND.OR NAME	
04	03	5A	RUNNER	BAD SNAP NAME	
04	03	5B	RUNNER	BAD PMD NAME	
04	03	5C	RUNNER	BAD MODIFY NAME	
04	03	5D	RUNNER	TOO MANY DEBUG FLAGS	
04	03	5E	RUNNER	TOO MANY DEBUG COMMANDS	
04	03	5F	RUNNER	INVALID BYTE 0 ON DEBUG RECORD	
04	03	60	RUNNER	RUNNER RECEIVED I/O ERROR READING LM HEAD OR TREE	
04	03	61	RUNNER	RUNNER RECEIVED I/O ERROR READING LM RF/DF STACK	
04	03	62	RUNNER	RUNNER RECEIVED I/O ERROR READING THE DEBUG FILE	
04	03	63	RUNNER	BAD SYMBOLIC NAME FOR START ADDRESS.	
04	03	64	RUNNER	MODIFY LOCATION NOT WITHIN LIMITS	
04	03	65	RUNNER	SYMBOLIC START ADDRESS NOT WITHIN LIMITS	
04	03	66	RUNNER	RUNNER FOUND INVALID TREE SIZE IN LOAD MODULE	
04	03	67	RUNNER	RUNNER CAN'T GET PG FOLLOWING PURE P FOR DEBUG AND CLOBBER TABLE	
04	03	68	RUNNER	PMD LOCATION NOT WITHIN LIMITS	
04	03	69	RUNNER	FROM OR TO LOCATION NOT WITHIN LIMITS	
04	03	6A	RUNNER	SNAP LOCATION NOT WITHIN LIMITS	
04	03	6B	RUNNER	RUNNER HAS LOGICAL ERROR IN MOVING DEBUG TABLE	
04	03	6C	RUNNER	PMDS AND DEBUGS ARE NOT ALLOWED FOR LINKED LMNS.	
06	00	00	DRSP	NO ERRORS	
06	00	01	DRSP	INSUFFICIENT PRIVILEGE LEVEL TO PROCESS THIS COMMAND	
06	00	02	DRSP	INSUFFICIENT MEMORY TO READ TREE	
06	00	03	DRSP	DRSP PROGRAM ERROR (SHOULDN'T HAPPEN)	
06	00	04	DRSP	PRONAME REQUIRED	
06	00	05	DRSP	NO SUCH OVERLAY/PROCESSOR	
06	00	06	DRSP	ILLEGAL COMMAND OPTION	
06	00	07	DRSP	ILLEGAL FLAG COMBINATION	
06	00	08	DRSP	DON'T SET FLAGS WITH MONITOR OVERLAY	
06	00	09	DRSP	PROCESSOR/OVERLAY ALREADY EXISTS	
06	00	0A	DRSP	DON'T USE COMMAND ON TEL/CCI	
06	00	0B	DRSP	ILLEGAL PRONAME. NOT :PNN FORMAT	
06	00	0C	DRSP	CANNOT OPEN THE FID	
06	00	0D	DRSP	FID IS NOT A LOAD MODULE	
06	00	0E	DRSP	ONLY ONE LEVEL OF OVERLAYS FOR SHARED PROCESSORS	
06	00	0F	DRSP	OVLY DATA EXCEEDS RANGE 8000-8BFF	
06	00	10	DRSP	ONLY PROCEDURE IS ALLOWED IN A PROCESSOR OVERLAY	
06	00	11	DRSP	NO PRONAME SLOTS AVAILABLE	
06	00	12	DRSP	INSUFFICIENT OVERLAY SLOTS	
06	00	13	DRSP	INSUFFICIENT SPACE ON SWAP RAD	
06	00	14	DRSP	FILE STORAGE LIMIT IN SYSTEM ACCOUNT	
06	00	15	DRSP	INSUFFICIENT VIRTUAL MEMORY TO EXECUTE DRSP	
06	00	16	DRSP	INSUFFICIENT MEMORY TO READ MAX RECORD OF FID	
06	00	17	DRSP	INSUFFICIENT PRIVILEGE FOR DRSP USAGE	
06	00	18	DRSP	DRSP I/O ERR/ABN (CLOSE)	
06	00	19	DRSP	DRSP I/O ERR/ABN (PERM)	

GRP	ERR	SUB	MODULE	MESSAGE
06	00	1B	DRSP	DRSP I/O ERROR IN READING COMMAND
06	00	1C	DRSP	DRSP M:EI ERROR (PERM)
06	00	1D	DRSP	ILLEGAL INDEX RANGE
06	00	1E	DRSP	ILLEGAL PROTECTION TYPE FOR PUBLIC LIBRARY
06	00	1F	DRSP	LM MUST BE ROOT ONLY, PROCEDURE ONLY
06	00	20	DRSP	PRONAME IS ILLEGAL
06	00	21	DRSP	MONITOR OVERLAY CANNOT HAVE OVERLAYS
06	00	22	DRSP	DRSP M:EI ERROR (WRITESWAP)
06	00	23	DRSP	PRONAME NOT FOUND ON RAD
06	00	24	DRSP	PRONAME FOUND ON RAD
06	00	25	DRSP	READ ERROR READING FID (COPY)
06	00	26	DRSP	WRITE ERROR WRITING FID (COPY)
06	00	28	DRSP	DRSP M:BO ERROR (PERM)
06	00	29	DRSP	CAN'T OPEN M.BO (PERM)
06	00	2A	DRSP	CAN'T MAKE PERM, NO RAD SLOTS
06	00	30	DRSP	INCORRECT FID
06	00	31	DRSP	WRITE RAD FILE I/O ERRORS
06	00	32	DRSP	RAD OVERFLOW - DISK SPACE ALLOTTED FOR THE SHARED PROCS IS EXHAUSTED
06	00	34	DRSP	I/O ERROR WHILE WRITING PROCESSOR TO THE SWAP DISK
06	00	35	DRSP	ILLEGAL LMN (LOAD BIAS CHECK)
06	00	36	DRSP	ILLEGAL LOAD MODULE - POSSIBLY BAD BIAS OR CREATED BY LINK LOADER
06	00	39	DRSP	PROCESSOR OVERLAY SLOTS EXHAUSTED .
06	00	50	DRSP	BREAK 50
06	00	51	DRSP	BREAK 51
06	00	52	DRSP	BREAK 52
06	00	53	DRSP	BREAK 53
06	00	54	DRSP	BREAK 54
06	01	00	DRSP	SYNTAX ERROR
06	01	01	DRSP	NO SUCH COMMAND
06	01	02	DRSP	COMMAND NOT IMPLEMENTED
06	01	03	DRSP	TO SPECIFY ALTERNATE ACCOUNT: CR 'FID' FROM 'ACCOUNT'
06	01	04	DRSP	ACCOUNT FIELD ($) > 8 BYTES
06	01	05	DRSP	HIT BREAK TO ENTER DELTA
06	01	06	DRSP	NO ACTION TAKEN
06	01	07	DRSP	NOT ENOUGH MEMORY
06	01	08	DRSP	PROCESSING TERMINATED
06	02	00	DRSP	NO ERRORS
06	02	01	DRSP	NO BOOTFILE
06	02	02	DRSP	THAT'S NOT A BOOTFILE
06	02	03	DRSP	LOCAL SYMBOL TABLES NOT COPIED
06	02	04	DRSP	LM NOT COPIED TO BF
06	02	06	DRSP	$. NO HEAD RECORD
06	02	07	DRSP	NO SUCH SEGMENT
06	02	08	DRSP	$. NO GLOBAL SYMBOL TABLE
06	02	09	DRSP	$ IS NOT A ROM
06	03	00	DRSP	NO BOOTFILE
06	03	01	DRSP	FIELD TOO LONG ($)
06	02	0A	DRSP	BREAK AFTER $
06	02	0B	DRSP	PROCESSING TERMINATED AFTER $
06	03	02	DRSP	BAD CHARACTER ( ) IN FID
06	03	03	DRSP	CANNOT OPEN $
06	03	04	DRSP	BAD FID
06	03	05	DRSP	CANNOT FIND $
06	03	06	DRSP	FILE $: RECORD $
06	03	07	DRSP	ERROR WRITING $ TO BF
06	03	0A	DRSP	$ SYNTAX ERROR
06	03	0B	DRSP	RECORD DOES NOT EXIST

GRP	ERR	SUB	MODULE	MESSAGE
06	03	0C	DRSP	ERROR ' FOR RECORD $
06	03	09	DRSP	CANNOT OPEN $ SEQUENTIAL
06	03	0F	DRSP	ERROR: $ LOCAL SYMBOL TABLE DELETED
06	03	10	DRSP	LOST DATA IN $ LOCAL SYMBOL TABLE
06	03	11	DRSP	ERROR * AFTER $
06	03	15	DRSP	NO LOCCT WITH ELEMENT $
06	03	16	DRSP	ELEMENT $ NOT IN LOCCT $
06	03	18	DRSP	ERROR · SEARCHING FOR LOCCT
06	03	19	DRSP	ERROR · READING LOCCT
06	04	00	DRSP	MONFIX TABLE OVERFLOW FOR $
06	05	00	DRSP	WHAT REEL NUMBER?
06	05	01	DRSP	WHAT MODULE?
06	05	02	DRSP	REEL NUMBER > 4 CHARACTERS
06	07	00	DRSP	LOADER ERROR FOR $
08	FF	00	BOOTSUBR	OPERATOR INITIATED RECOVERY
08	04	04	SCHED	OPERATOR INITIATED SHUTDOWN
08	01	00	SCHED/MM	USERS PAGE CHAIN INCONSISTENT
08	02	00	SCHED	REPORTED EVENT INCONSISTENT WITH USER'S CURRENT STATE
08	0A	00	TSIO/DPSIO	OPCODE IN SWAP COMMAND LIST IS INVALID
08	0B	00	TSIO/DPSIO	INCORRECT ORDER CODE IN SWAP COMMAND LIST
08	0C	00	TSIO/DPSIO	ATTEMPT MADE TO SWAP MONITORS MEMORY
08	0D	00	TSIO	HALT FLAGS MISSING IN SWAP COMMAND LIST
08	0E	00	TSIO	I/O REQUEST WITH NULL COMMAND LIST
08	0F	00	TSIO/DPSIO	INPUT FUNCTION CODE IS INVALID
08	10	00	COC	COC-BAD COC BUF POOL, OR BAD BUF ADR ON RELEASE REQUEST
08	11	00	COC	COC-INVALID INTERNAL CONTROL CODE TRANSLATE REQUEST
08	12	00	COC	COC-BAD INPUT BUF LINKAGE ON RELEASE REQUEST
08	13	00	COC	COC-OUTPUT BUF LINKAGE OR CHARACTER COUNT BAD
08	14	00	THEUNCOC	COC ROUTINE WAS CALLED IN A NON-COC SYSTEM
08	14	01	THEUNCOC	COCIO CALLED IN A NON-COC SYSTEM
08	14	02	THEUNCOC	COCOFF CALLED IN A NON-COC SYSTEM
08	14	03	THEUNCOC	COCSENDX CALLED IN A NON-COC SYSTEM
08	14	04	THEUNCOC	ECHOCR2 CALLED IN A NON-COC SYSTEM
08	17	00	IOQ	INVALID DISC ADDRESS PASSED FOR AN I/O OPERATION
08	19	00	BUFF	INVALID BUFFER ADDRESS PASSED FOR RELEASE
08	1A	00	CLS	ACCOUNT DIRECTORY IS INACCESSIBLE
08	1B	00	SWAPPER	USERS PAGE CHAIN NON-ZERO AT SWAP COMPLETION
08	1D	00	T:OV	REQUESTED OVERLAY NUMBER IS OUT OF RANGE
08	1E	00	SCHED	SYSTEM REBOOTED WITHOUT A PROPER SHUTDOWN
08	1F	00	SWAPPER	NOT ENOUGH PAGES TO PERFORM THIS SWAP
08	21	00	MM	ATTEMPT TO SET ACCESS ON AN NON-EXISTENT VIRTUAL PAGE
08	22	00	TYPR	PRIVATE VOLUME ALLOCATION ERROR
08	23	00	CSEHAND	INVALID ENTRY TO CSE HANDLERS
08	24	00	CSEHAND	INSTRUCTION EXCEPTION TRAP IN MASTER MODE
08	25	00	CSEHAND	UNRECOVERABLE WATCHDOG TIMER TRAP
08	26	00	CSEHAND	CSE TRAP DURING MFI, PFI HANDLING
08	27	00	CSEHAND	PROCESSOR FAULT INTERRUPT
08	28	00	CSEHAND	MEMORY PARITY ERROR - MEMORY ALTERED
08	29	00	CSEHAND	TRAP 4C - BUS CHECK FAULT
08	29	01	CSEHAND	TRAP 4C - MAP PARITY ERROR
08	29	02	CSEHAND	TRAP 4C - REGISTER BLOCK PARITY ERROR
08	29	03	CSEHAND	TRAP 4C - WRITE LOCK REGISTER PARITY ERROR
08	2C	00	ADD	BATCH SCHEDULING ERROR - MBS/CCI ERROR
08	2D	00	COOP	COOPERATIVE BUFFER MANAGEMENT ERROR
08	2D	01	COOP	SYMBIONT/COOP FILE DEVICE INACCESSIBLE
08	2D	02	COOP	USERS COOP CONTEXT BLOCK CHAIN LOST
08	2D	03	SACT	COOP CONTEXT BLOCK POINTERS CLOBBERED

```
GRP ERR SUB MODULE                        MESSAGE
08  2D  04  OUTSYM           COOP SENT OUTSYM BAD DATA
08  2E  00  RDF             POOL BUFFERS LOST - NONE ALLOCATED CURRENTLY
08  2E  01  RA               INCONSISTENCY IN READ-AHEAD TABLES
08  30  00  PFSR            UNBALANCED POWER ON/OFF INTERRUPT PAIRS
08  31  00  IORT            INVALID RESOURCE TYPE
08  32  00  IOQ             DCB DOESNT CONTAIN A VALID DCT INDEX
08  34  00  TPQ1             TRANSACTION PROCESSING FAILURE
08  41  01  RTROOT          FAILED TO FIND USER'S STATE (M:INFSTAT)
08  41  10  RTROOT          ERROR RETURN FROM NEWQNW ON M:IOEX CALL
08  41  11  RTNR            UNABLE TO RE-START SYSTEM I-O VIA M:STARTIO REQUEST
08  43  01  CLOCK4          NO ICBS CHAINED INTO RTICBCLKHDR
08  43  02  CLOCK4          ICBCLK FIELD OF ICB NEGATIVE
08  43  03  CLOCK4/RTNR     NO BACK-LINK FOUND IN DE-CHAINED ICB
08  46  21  RDF             PRIVATE VOLUME LOGIC INCONSISTENCY
08  49  00  TYPR            RESOURCE PRE-ALLOCATION INCONSISTENT WITH REQUESTS
08  56  00  MOCIOP           UNABLE TO RELEASE ASSIGNED PHYSICAL WORK PAGE
08  61  00  INITRCVR        TEL OR CCI HAS TRAPPED
08  61  40  INITRCVR        TEL/CCI SUFFERED A TRAP 40
08  61  42  INITRCVR        TEL/CCI COMMITTED A STACK TRAP
08  62  00  SCHED           USER PROGRAM TOO LARGE FOR PHYISICAL MEMORY
08  63  00  DPSIO           INSUFFICIENT INFORMATION AVAILABLE TO SWAP THIS USER
08  6A  00  MM              ATTEMPT TO RELEASE VIA M:CVM FROM USER W/O PROPER PRIVILEDGE
08  6B  00  SWAPPER         ERROR IN SPARE BUFFER TABLES
08  79  00  ENTRY           MONITOR COMMITTED A STACK TRAP
08  7C  00  ALTCP           ALTCP CALLED TO SERVICE A CAL THAT DOESN'T BELONG TO ALTCP
08  7E  00  INITRCVR        MONITOR HAS SUFFERED A TRAP 40
08  7E  40  INITRCVR        MONITOR HAS TRAPPED
08  7E  41  INITRCVR        MONITOR HAS TRAPPED
08  87  00  ALLYCAT         ALLOCATION BUFFERS CONTAIN INVALID WORD COUNT
08  88  00  SCHED           ALLYCAT CLOBBERED ONE OF THE ALLOCATION BUFFERS
08  89  00  ALLYCAT         ALLYCAT'S HGP CHAIN CLOBBERED
08  93  00  TSIO/DPSIO      TDV ADDRESS DOESNT POINT TO THE COMMAND LIST
08  94  00  TSIO/DPSIO      COMMAND LIST CLOBBERED DURING WRITE-CHECK
08  95  00  TSIO/DPSIO      UNRECOVERABLE I/O ERROR READING USER'S JIT
08  96  00  TSIO/DPSIO      UNRECOVERABLE I/O ERROR READING SHARED PROCESSOR
          END-OF-JOB
```

DATA BASES

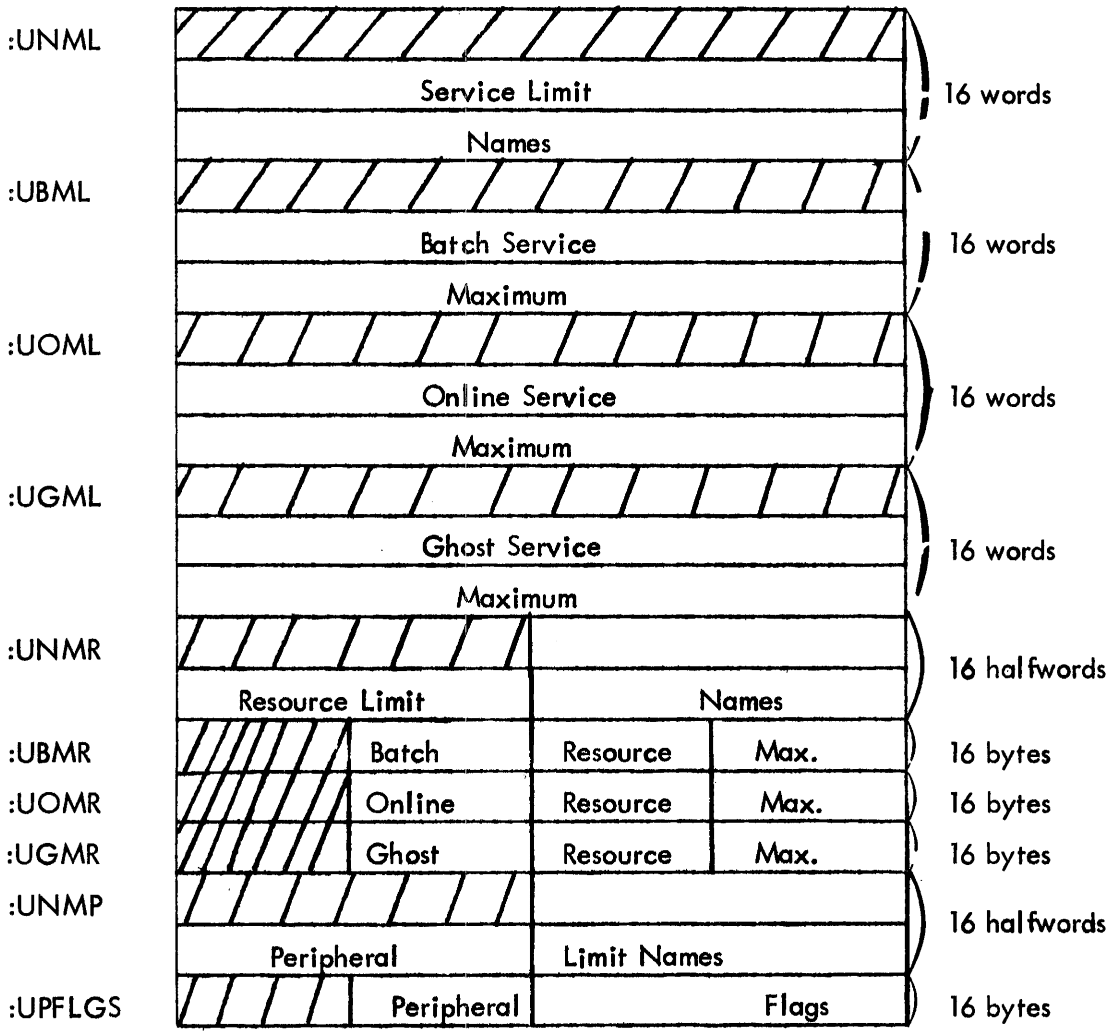
:USER File Record Format

Each 126 word record in the logon file, called :USERS, corresponds to an authorized user. Whenever an on-line or batch user attempts to use the system, the :USERS file is checked to determine if there is a corresponding record. If not, he is denied access to the system, if there is a record associated with that user and if he has logged on with the correct password (if any), he may access the system.

Each record in the :USERS file contains a profile of a user authorization for use of various system resources - such as number of spindles, tape drives, core space RAD and DISK space, etc. At log on time, these various items are placed into corresponding fields in JIT and/or Assign-Merge Table so that information about the user is readily accessible. Controls can then be duly exerted by the monitor. The :USERS file is accessed by SUPER, TEL, CCI, and LOGON.

Word	0					Account	
	1						
	2					Name	
	3						
	4						
	5	0 … 27 MC, 28 XO, 29 RP, 30 SF, 31				Flag Bits	
	6					Password	
	7						
	8					Password	Auto-Call Processor
	9						
	10					Account	
	11						
	12	Count				Name (TEXTC format)	
	13						
	14						
UB:BILL	15	Batch billing	on-line billing	ghost billing	▨		
UB:PRIV	16	Batch privilege	on-line privilege	ghost privilege	▨		
	17	▨					
	18	Accumulated Permanent RAD Space *					
	19	▨					
	20	Default Reten. Per.		Max. Reten. Period			
	21	▨					
	22	Accumulated Permanent DISK Space *					
	23	▨					
	24	Extended Accounting					
	25	Information					
	26	(6 words)					
	27						
	28						
	29						

*Updated by LOG-OFF

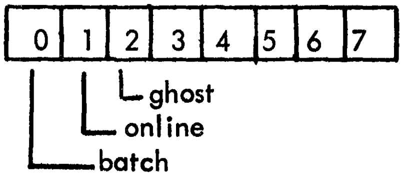

:UNML
Service Limit
Names
16 words
:UBML
Batch Service
Maximum
16 words
:UOML
Online Service
Maximum
16 words
:UGML
Ghost Service
Maximum
16 words
:UNMR
Resource Limit
Names
16 halfwords
:UBMR
Batch
Resource
Max.
16 bytes
:UOMR
Online
Resource
Max.
16 bytes
:UGMR
Ghost
Resource
Max.
16 bytes
:UNMP
Peripheral
Limit Names
16 halfwords
:UPFLGS
Peripheral
Flags
16 bytes

Contents of Users File Record

Name	Description
Account	EBCDIC. Account under which the user is authorized to log in. Value may be 1 to 8 characters in length and is left-justified with trailing blanks.
Name	EBCDIC. Name under which the user is authorized to log in. Value may be 1 to 12 characters in length and is left-justified with trailing blanks.
All	Bit 0 Default read access for user, 0 indicates ALL; 1 indicates NONE.
MC	Word 5, bit 27. Set if user has the MC (Maintenance Console) specified as a resource. Used for internal use by SUPER.
XO	Word 5, bit 28. Set if user may execute processors from :SYS account only. Default = 0.
RP	Word 5, bit 29. Set if user has one or more entries in the :PROCS file specifying this user is allowed/disallowed to execute.
SE	Word 5, bit 30. Set if security check of granule and core cleaning is required.
Password	EBCDIC. Password assigned to this name-account. The contents of the field is the latest value entered by the user himself with a TEL PASSWORD command, or by SUPER via the PASSWORD option of a CREATE or MODIFY command. Value may be 1 to 8 characters in length and is left-justified with trailing blanks. If no password has been specified, these words contain zeroes.

<table>
<tr><td>Auto-Call</td><td>EBCDIC. Password of LMN to be automatically connected to the user's job when he logs on. Value may be 1 to 8 characters in length and is left-justified with trailing blanks. If no auto-call LMN has been specified, auto-call password contains zeroes.</td></tr>
<tr><td>Auto-Call account</td><td>EBCDIC. Account in which resides the LMN to be automatically connected to the user's job when he logs in. Value may be 1 to 8 characters in length and is left-justified with trailing blanks. The value for system processors is ":SYS". If no auto-call LMN has been specified, these words contain zeroes.</td></tr>
<tr><td>Auto-call name</td><td>EBCDIC. Name of LMN to be automatically connected to the user's job when he logs in. The value is in TEXTC format, may be 1 to 11 characters in length, and is left-justified with trailing blanks; the first byte contains a count. If no auto-call LMN has been specified, these words contain zeroes.</td></tr>
<tr><td>Billing</td><td>Binary. Charge class level referencing a charge class table in the RATE file which is used in account processing. The halfword value must lie in the range:
$0 \leq \text{charge class} < \underline{7}$</td></tr>
<tr><td>Privilege</td><td>Binary. Privilege level granted the user. The higher privilege codes include the lower privilege. The range of the privilege codes is X'0' to X'FF'. The currently meaningful codes are as follows:
<table>
<tr><th>hex code</th><th>typical use</th><th>facilities allowed</th></tr>
<tr><td>C0</td><td>systems programmer</td><td>Issue M:SYS CAL
Bypass file security checks.</td></tr>
<tr><td>B0</td><td>systems programmer</td><td>Access and change monitor</td></tr>
<tr><td>A0</td><td>diagnostic programs</td><td>Read and write error file; request devices; invoke diagnostics.</td></tr>
<tr><td>80</td><td>performance displays</td><td>Examine monitor</td></tr>
<tr><td>40</td><td>normal programs (online and batch)</td><td>Default privilege level</td></tr>
</table>
</td></tr>
</table>

Maximum Retention	Binary. Hours of maximum retention period for files created by the user. A zero means the system default value and X'FFFF' means the files are to be retained indefinitely.
Accumulated Permanent DISK Space	Binary. Permanent DISK storage accumulated by the user. This field is not modified by SUPER.
Accumulated Permanent RAD Space	Binary. Permanent RAD storage accumulated by the user. This field is not modified by SUPER.
Extended Accounting Information	EBCDIC. This field specifies installation specific accounting information. A maximum of 24 characters is allowed. The character ; is not allowed.
Service Limit	0-15 four character service limit names. This table gives the serivces for which the maximum values are given in the next three tables.
Batch, Online and Ghost Service Maximum	Three parallel tables of 0-15 full word binary values associated with the service limit names. One table each for batch, online and ghost operation in this account.
Resource Limit Names	0-15 two character resource limit names. This table gives the resources for which maximum usage values are given in the next three tables.
Batch, Online and Ghost Resource Maximum	Three parallel tables of 0-15 byte binary values associated with the resource limit names. One table each for batch, online and ghost operation.

Default Retention Period	Binary. Hours of retention period for files created by the user. A zero means the system default value and X'FFFF' means the files are to be retained indefinitely.
Peripheral Limit Names	0-15 two character peripheral device limit names. This table contains the peripheral device names for which use authorization flags are given in the next table.
Peripheral Flags	A table of 0-15 bytes parallel to peripheral limit names each byte has the form:  A bit value of one means that authorization to use the device is granted.

Job Accounting Record

The following Figure shows the additional structure of the job accounting record. Words 0-37 remain unchanged with respect to previous versions except that the field marked "scratch tapes" in word 32 is now unused.

Four words have been appended to the record, words 38-41 and contain the resource allocation values: the number allocated for batch jobs (regardless of whether or not they were actually used), and, for on-line and ghost jobs it is the number in use at time of logoff. Byte zero of word 38 contains flags which indicate the condition under which the accounting record was generated. The start and end time may be optionally expressed in seconds from midnight by the installation's setting of bits 31 of S:OPTION in the Monitor Root.

Word				
0	Account			
2	Name			
5	Extended Accounting			
11	Charge Units			
12	Line Number	Priority	Final Run Status	Job Steps
13	Job Origin	System Version		
14	Start Date			
15	Start Time			
16	End Time			
17	Console Interactions			
18	Finish Date			
19	SYSID		Pack Mounts	Spindles
20	Cards Read		Cards Punched	
21	Processor Pages		User Pages	
22	Diagnostic Pages		Tape Mounts	Tape Drives
23	Tape Accesses			
24	RAD Accesses			
25	Disk Accesses			
26	I/O CALS			
27	Permanent RAD Granules			
28	Permanent Disk Granules			
29	Core Usage			
30	Processor Execution Time			
31	Processor Service Time			
32	Maximum Core Size	Partition	Save Tapes	▨ ~~Scratch Tapes~~
33	User Execution Time			
34	User Service Time			
35	Peak Temporary RAD Granules			
36	Peak Temporary Disk Granules			
37	Billing Rate		▨	
38	Flags	Resource Allocation	Values	
39				
40				
41				

Contents of Accounting Record

Name	Description
Account	Account number of the user as specified on the job card or log-on message.
Name	Name of the user as specified on the job card or log-on message.
Extended Accounting	Installation-defined accounting information as specified on the job card or log-on message.
Charge Units	Accumulated charge units calculated for the user through use of the rates table.
Line Number	Line number (Data Set Controller Subchannel) to which the user connected. Line Number is set to X'FF' if entry is for a batch job.
Priority	Priority specified on the job card. Unused if entry is for a terminal session.
Final Run Status	Run status at the completion of the job (an eight bit field). X'00' - Job exited normally. X'01' - Job aborted, illegal trap. X'02' - Job aborted, I/O error. X'04' - Job aborted, limit exceeded. X'08' - Reserved for CHKPT. X'10' - Job aborted, 'X' key-in. X'20' - Last job step errored, 'E' key-in. X'40' - Job aborted, M:xxx. X'80' - Last job step errored, M:ERR.
Job Steps	Total number of job steps if batch; total number of subsystem operations if on-line.
Job Origin	Origin of batch job. 0 - From local card reader 1 - From on-line terminal 2 - Reserved for remote batch
System Version	Version of operating system (from cell X'2B') in EBCDIC.
Start Date	Date at job or terminal session start, where the left halfword is the year and the right halfword is the day. Year is a binary value; e.g., 1970 is represented as X'7B2'. Day is the Julian day of the year represented in binary; e.g., September 14 is represented as X'101'.
Start Time	Time of day at start of job of terminal session in minutes from midnight. The value is expressed in binary.
End Time	Time of day at end of job or terminal session. Expressed in the same format as start time.
Console Interactions	Number of interactions during the course of a terminal session (zero for batch).

:ACCTLG - Accounting Log Format

Contents of Accounting Record (cont.)

Name	Description
Finish Date	Date at job or terminal session finish. The format is the same as for Start Date.
Sysid	ID assigned to user job or session.
Pack Mounts	Number of disk packs mounted.
Spindles	Maximum number of disk pack spindles allocated to batch or available to on-line.
Cards Read	Number of cards read, including the job card and any EOD cards, but no FIN cards.
Cards Punched	Number of cards punched, including ID card, JOB card, BIN cards and EOD cards, but not blank cards inserted by the punch symbiont between jobs. If no punched output is produced by the job, the ID and JOB cards are suppressed and the punched card count is reduced to zero.
Processor Pages	Number of pages of printed output generated by shared processors, plus two ID pages at the beginning of the job and the accounting page at the end.
User Pages	The number of pages of printed output generated by user programs only.
Diagnostic Pages	The number of pages of all output to a symbiont file through the M:DO DCB, including core dump snaps and debug output. However, output is not counted if it goes to a user file, even though it goes via M:DO.
Tape Mounts	Number of tapes mounted unless premounted by the operator.
Tape Drives	Maximum number of tape drives allocated to batch or available to on-line.
Tape Accesses RAD Accesses Disk Accesses	Number of read, write, and file positioning accesses on the specified device, but not seek accesses since these are considered part of a read or write. A charageable access is actually a request to a queue.
I/O CALs	Number of CAL1, 1 operations performed.
Permanent RAD Granules	Net change in accumulated RAD storage. This is a signed binary value.
Permanent Disk Granules	Net change in accumulated public disk pack storage. This is a signed binary value.
Core Usage	Product of CPU time times core size in pages (ticks[†] x pages). Includes all core usage by job.
Processor Execution Time	CPU time spent in shared processors in the slave mode, expressed in ticks.[†]
Maximum Core Size	Peak value of core reached, expressed as the number of pages. Does not include shared processors or context.
Partition	Partition number under which the job ran (zero if terminal session).
Save Tapes	Number of save tapes used.
User Execution Time	CPU time spent in other than shared processors expressed in ticks.[†]
User Service Time	Monitor service time spent for other than shared processors expressed in ticks.[†]
Peak Temporary RAD Granules	Peak value of temporary RAD granules used.
Peak Temporary Disk Granules	Peak value of temporary public disk pack granules used.
Billing Rate	Charge class used for accounting for this user. The value is obtained from the user's log-on record and is in the range 0 to 7.

[†] One tick equals two milliseconds.

:ACCTLG - Accounting Log Format

Contents of Accounting Record (cont.)

Name	Description
Flags	Indicate condition under which accounting record was generated 0 = Normal 1 = Recovery condition 2 = Assign-Merge record error 3 = Recovery-but missing or bad A/M record
Resource Allocation	15 byte table containing the number of resources of each type allocated. For batch jobs it is the number allocated (regardless of whether or not they were actually used). For online or ghost jobs it is the number of resources in use at time of log off.

RATE RECORD FORMAT

0	8	Charge class pointer for charge classes zero through seven. Values are relative pointers to the following rate tables.	
1	16		
2	8		
3	8		
4	8		
5	8		
6	8		
7	8		
8	6	CPU time	Batch rate table
9	1	CPU time X core size	
10	0	Terminal Interactions	
11	1000	I/O CALs	
12	0	Console minutes	
13	100000	Tapes and packs mounted	
14	10000	Page - day storage	
15	2000	Peripheral I/O cards + pages	
16	6	CPU time	On-Line rate table
17	1	CPU time X core size	
18	1000	Terminal Interactions	
19	1000	I/O CALs	
20	200	Console minutes	
21	100000	Tapes and packs mounted	
22	10000	Page - day storage	
23	2000	Peripheral I/O cards + pages	

ID

:RBLOG - Remote Batch Logon File

PURPOSE

Each record except the ID record in the remote batch logon file, called :RBLOG, corresponds to an authorized work station. Whenever a user tries to log a work station onto the system via the !RBID command, or uses a work station name on a JOB card, the :RBLOG file is checked to determine if there is a corresponding record. If not, access to the system is denied; if there is a record associated with that work station name, the system may be accessed. The :RBLOG file is accessed by SUPER, RBBAT, and LDEV.

:RBLOG File Record Format

The remote batch ID record (RBID), which is written with the key '...', contains a chain of used RBIDs and a chain of free RBIDs.

The record written with a key of '///' contains the list of WSNs indexed by RBIDs.

The records keyed by WSN are described in three formats:

1. Record layout with associated SUPER internal labels.
2. Record elements keyed to the SUPER internal labels.
3. A bit by bit breakdown of the three flag word formats used in the :RBLOG record.

INTERNAL LABEL	RECORD INDEX	FORMAT/CONTENT		
			0 8 16 24	
RB:RBID	0		RBID |	
R:FLAG	1		Flag Bits	
RD:WSN	2 3		Work station name in text 1-8 characters	
RB:RP	4		Max. Priority |	
R:BUF	5		Buff	
RD:RWSN	6 7		Remote work station name in text (1-8 characters)	
RB:DSM R:RMT	8		DSM | | RMT Number (slave)	
R:INRCB	9		| OC Input RCB	
R:HSX	10		Highest device (stream index)	
R:OUTRCB	11		| SMD output RCB	
RH:DEV	12 19	8 words	Device name table 16 HW Text	*
RB:RCB	20 23	4 words	RCB table - 16 bytes	
RH:F1	24 31	8 words	FLAG 1 Table - 16 HW	
RH:F2	32 39	8 words	FLAG 2 Table - 16 HW	
RH:SUS	40 47	8 words	SUSBIT Table - 16 HW	
RB:MAX	48 51	4 words	Maximum Record Count Table 16 bytes	
RB:MIN	52 55	4 words	Minimum record Count Table 16 bytes	
RB:PRIV	56 59	4 words	Privilege Table 16 Bytes	
RB:TYPX	60 63	4 words	Type Index Table for DC Option 16 Byte.	
RD:GHOST	64 65	2 words	TEXT GHOST NAME	
	66 79	14 words	Reserved for future use	

*These are nine parallel table of 16 elements each, indexed by device.

:RBLOG RECORD (80 WORDS LONG)

Internal Label	Min. Value	Max. Value	Default Value	Data Type	Associated Super Option	Note
R:RBID	0	X'FF'		hex		Assigned by Super
R:FLAG					Type Master/Slave X1/N/ X2/N2 SYS/NSYS	description of R:FLAG follows
RD:WSN				Text 1-8 char.	W	Specified by user when creating new work station
RB:RP	0	X'F'	X'7'	hex	RP	
R:BUFF	1	FF	1	flag	MRB	MRB→FF NMRB→1
RD:WSN				text 1-8 char.	RWSN	
RB:DSM	0	X'FF'	0	hex	DSM	
R:RMT				text 2 char	RMT	
R:INRCB	0	X'FF'	0	hex	IRCB	
R:HSX	0	X'F'			DEV DD	Maintained by SUPER. Indicates the number of devices in record minus 1.
R:OUTRCB					SMD/NSMD	Only one allowed per work station
RH:DEV				Text 2 char.	DEV DD	
RB:RCB	1	X'FF'	must be specified	hex	RCB	

Internal Label	Min. Value	Max. Value	Default Value	Data Type	Associated Super Option	Not
RH:F1					CTL/NCTL SRCB IN/OUT DEV:OC	description of RH:F1 follows
RH:F2					DEV=OC DIR/NDIR LIST SRCB IN/OUT BIN/NBIN	description of RH:F2 follows
RH:SUS	0	X'FFFF'	0	hex	SUS	
RB:MAX	1	255	80 *	decimal	MAX, CP2, LP2	
RB:MIN	1	255	1	decimal	MIN, LPP	
RB:PRIV	0	X'FF'	40	hex	PRIV	
RB:TYPX	0	TYPMNES2	0		DC	Only symb. device types legal
RD:GHOST			0	Text C	GHOST	

* For listing devices, defaults are: Max 132
Min 38 (lines per page)

FLAG NAME	BIT NUMBER (0 - 31)	SET	RESET
R:FLAG	6	TYPE= IRBT	TYPE=7670 or 2780
(word)	7	SIGMA is in SLAVE mode	in MASTER mode
	9	X1 specified	not X1
	10	TYP = 2780	TYPE = IRBT or 7670
	19	SYS specified	NSYS
	24	X2 specified	Not X2
	30	EM specified	NEM
	(remaining bits currently unspecified)		
RH:F1	0	IN and NCTL or OUT and DEV ≠ OC	IN and CTL or OUT and DEV = OC
	12	SRCB=C	not C
	13	SRCB=P	not P
	14	DEV=OC	DEV≠ OC
	15	IN	OUT
RH:F2 (HW)	0	DEV=OC	DEV ≠ OC
	1	DIR	NDIR
	2	LIST=P	LIST≠ P
	3	LIST= S	≠ S
	5	SRCB = U	≠ U
	6	SRCB = C *	≠ C
	7	SRCB = P	≠ P
	9	LIST = Y LIST = S	
	10	IN	OUT
	11	OUT	IN
	12	BIN	NBIN
	15	LIST =Y LIST = S	

* C + P = X

DESCRIPTION OF :RBLOG FLAG WORDS

Note regarding bit values relative to LIST option in RH:F2

	BIT	2	3	9	15
LIST=	N	0	0	0	0
	P	1	0	0	0
	Y	0	0	1	1
	S	0	1	1	1

Structure of the RBID Record

Word 0	HOF	0	TOU	HOU
1	1	2	3	4
2	5	6	7	8
3				
•				
•				
•				
64	253	254	255	0

HOF = Head of Free (initially 1)

TOU = Tail of Used (initially 0)

HOU = Head of Used (initially 0)

Values in words 1 - 64 are changed as work stations are authorized or deleted.

:PROCS File

The setting of the RP bit in the :USERS record (word 5, bit 27) indicates the presence of a record for that user in the :PROCS file. The :PROCS file is a keyed file, keyed by the concatenated user account and name identically to the :USERS file.

The data record will be of variable size up to a full granule (512 words). The first word of the record is reserved for flag bits and a count. The count is in the high order halfword's bits 1 - 7. The low order halfword contains an index to the next free byte in the record.

Bit 0, called PM, indicates the mode of the processor list. PM = 1 indicates that the list is a list of processors which are not allowed for this user. All processors not listed are allowed. When the PM bit is reset, each entry indicates a processor which is allowed in the specified mode (s), all others being disallowed.

(NOTE: The effectiveness of the :PROCS restrictions on executing processors is dependent upon execute only access on these processors so that the user cannot copy them.)

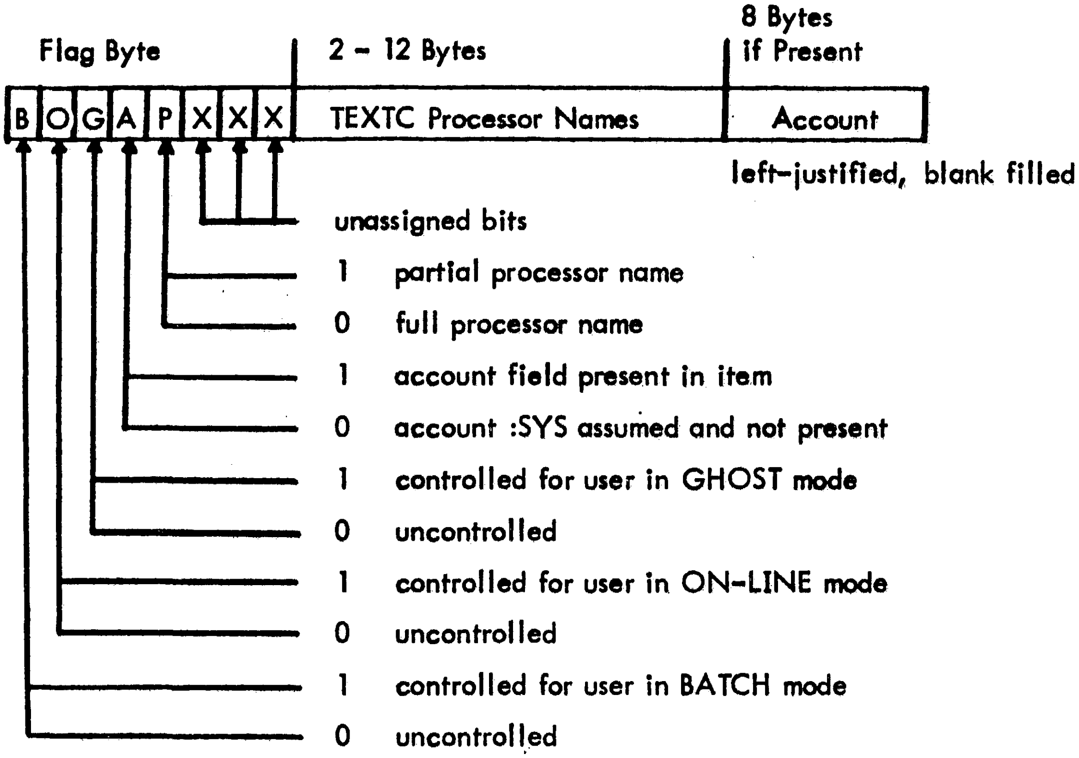
The remainder of the :PROCS record contains variable length items of the form:

Word 0

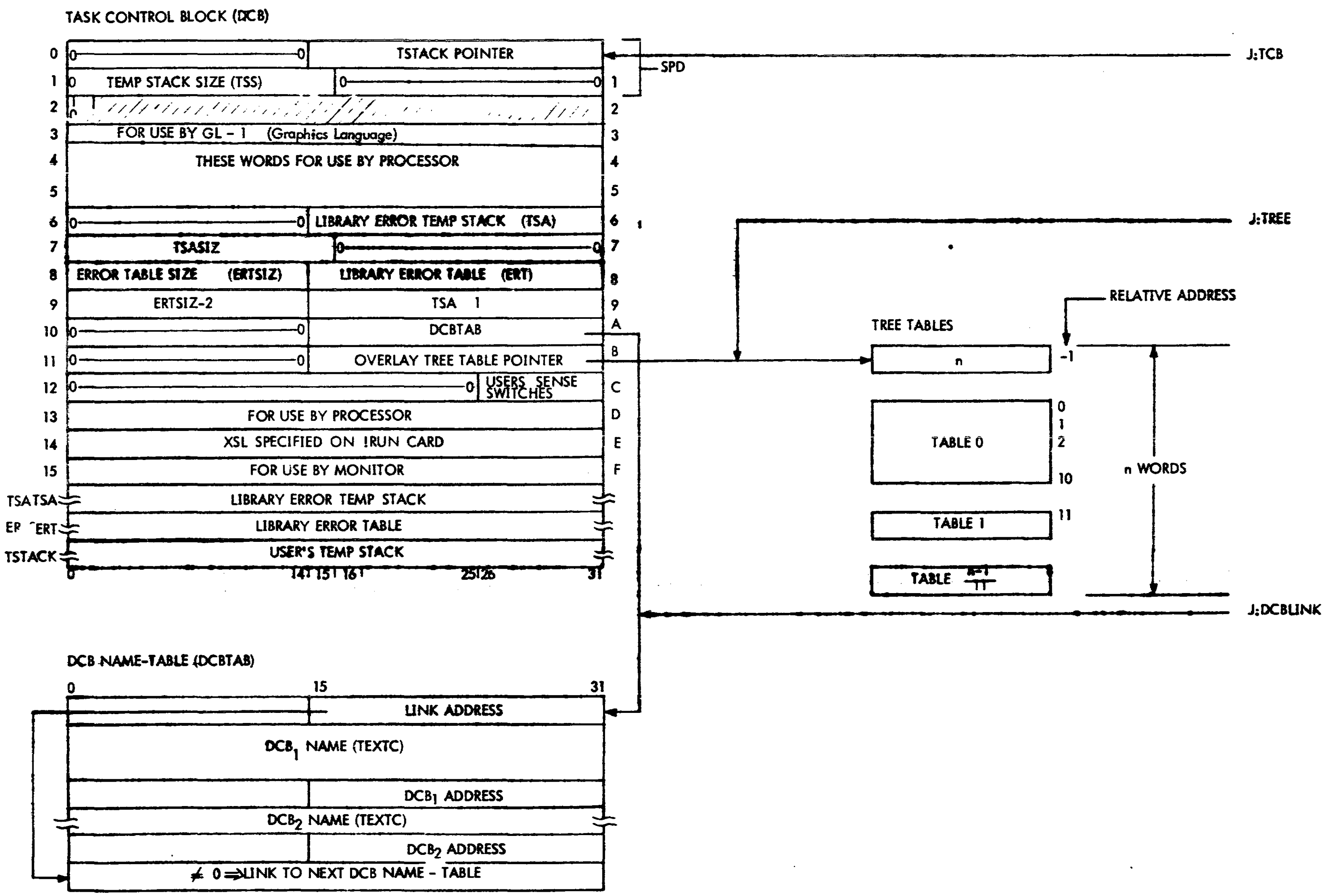
0	1 6	7 15	16 31
PM	Unassigned	Number of Items in Record	Record Byte Count

→ PM bit = 1 record contains names of disallowed processors
→ =0 record contains names of allowed processors

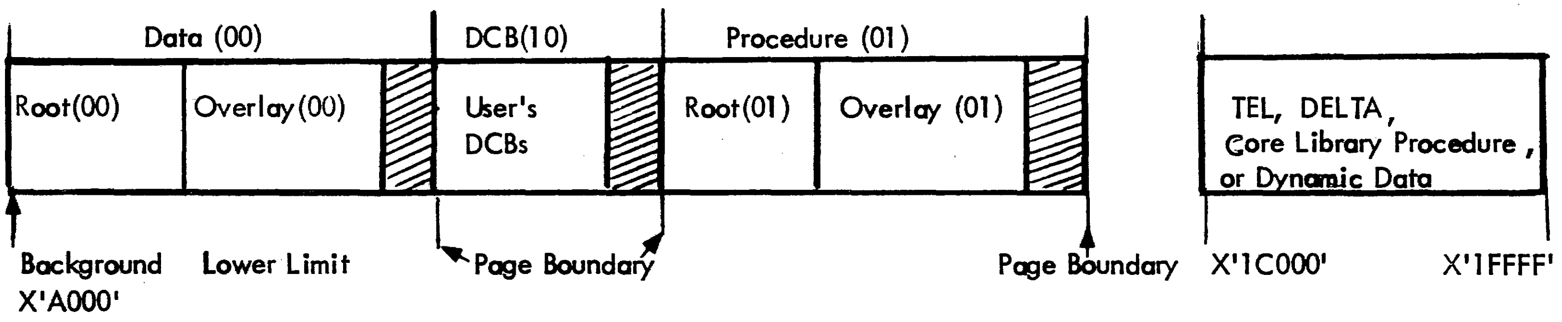
The remainder of the :PROCS record contains variable length items of the form:



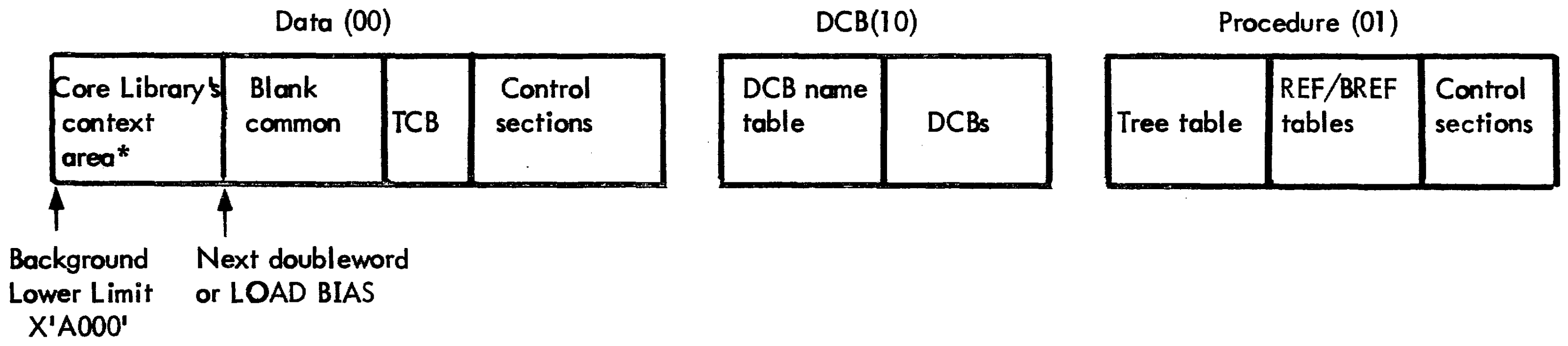
The minimum item size is 3 bytes, the maximum, 21 bytes. There is room in the record for 97 maximum size or 200 'ordinary' sized items.


TASK CONTROL BLOCK (DCB)
TSTACK POINTER
TEMP STACK SIZE (TSS)
SPD
J:TCB
FOR USE BY GL - 1 (Graphics Language)
THESE WORDS FOR USE BY PROCESSOR
LIBRARY ERROR TEMP STACK (TSA)
TSASIZ
ERROR TABLE SIZE (ERTSIZ)
LIBRARY ERROR TABLE (ERT)
ERTSIZ-2
TSA 1
DCBTAB
OVERLAY TREE TABLE POINTER
USERS SENSE SWITCHES
FOR USE BY PROCESSOR
XSL SPECIFIED ON !RUN CARD
FOR USE BY MONITOR
TSATSA
LIBRARY ERROR TEMP STACK
ERTERT
LIBRARY ERROR TABLE
TSTACK
USER'S TEMP STACK
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
0 1 2 3 4 5 6 7 8 9 A B C D E F
0 14 15 16 25 26 31
J:TREE
RELATIVE ADDRESS
TREE TABLES
n
-1
TABLE 0
0 1 2 10
TABLE 1
11
TABLE n-1
n WORDS
J:DCBLINK
DCB NAME-TABLE (DCBTAB)
0 15 31
LINK ADDRESS
DCB₁ NAME (TEXTC)
DCB₁ ADDRESS
DCB₂ NAME (TEXTC)
DCB₂ ADDRESS
≠ 0 ⇒ LINK TO NEXT DCB NAME - TABLE

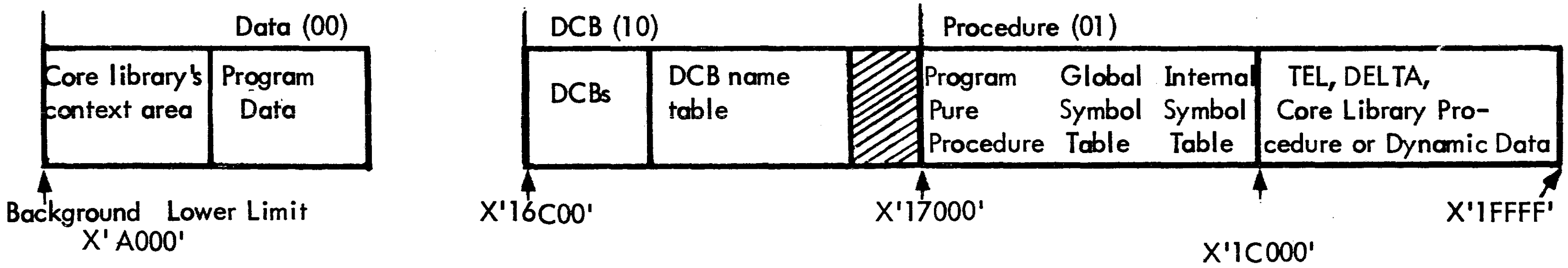
Overlay Loader-Built Load Module Layout at Run Time



Detail of Root



LINK-Built Load Module Layout at Run Time



*Allocated by Loader and initialized at run time by core library procedure routines.

Load Modules, Overall Format

A load module is a keyed file whose name was supplied to the Overlay Loader or LINK (default = idL). The keys and records are as follows:

Record

Key = HEAD	0	8X	00	FF	n	
	1	A B SL	[illegible]	START address		
	2	TCB*		Module Bias*		
	3	DATA Size*		DATA (00) Base*		
	4	PROCEDURE SIZE *		PROCEDURE (01) Base*		
	5	MAX RF/DF Size		TREE Size		
	6	DCB Size*		DCB Base (10)*		
	7		0		0	**
	8		0		0	***
	9		0		0	****
	A		0		0	
	B		0		0	

(Footnotes are on next page.)

Footnotes to keys and records shown on previous page:

*Doubleword address

In byte 0, word 0

X = 0, load module produced by Loader

= 1, load module produced by SYSGEN.

= 2, library load module produced by Loader.

= 3, load module produced by DEFCOM (consists of HEAD, TREE, and REF/DEF (Stack).

= 4, load module produced by LINK.

= 5, paged load module produced by Loader.

n = number of bytes in the HEAD record. For CP-V, n = X'30'.

A = 1, abs module

B = 1, NOTCB

SL= Final Severity Level

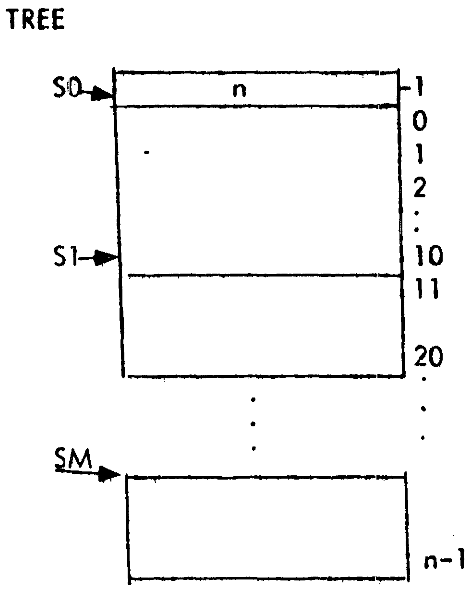
** Word 7	If DEFCOM output, this word = byte size of DATA area.
*** Word 8	If the LMN contains global symbol table (GST) information, Word 7 contains the size in words of the GST in bits 0-14 and location of the GST in bits 15-31. If the LMN internal symbol table (IST) information, Word 8 contains size in words of the IST in bits 0-14 and location of the IST in bits 15-31.
**** Words 9, A, B	If the LMN is associated with a core library, these words are :Pnnn in TEXTC format. If the CORELIB option was specified for a loader-built load module and no core library was associated, bit 0 of word 9 = 1.

Key = **TREE** **Record is the Tree Tables**

Tree Tables

Overall picture for M segments (S0, ... SM)

n = total size of the tables



Tree Table Format (one 11-word Table per Segment)

Displacement from the beginning of the ROM Tables to the first ROM Table for this segment

Tree Size		-1	
Segment Name in		0	
TEXTC Format		1	
		2	
ROM Pointer	Back Link **	3	
Forward Link**	Overlay Link**	4	
00 Size *	00 Loc*	5	Initially Cleared
REF/DEF Size	REF/DEF Loc*	6	
01 Size*	01 Loc*	7	
Expr. Size	Expr. Loc*	8	
10 Size*	10 Loc*	9	
		10	

Tree Tables

Segment name is determined by the name of the first file in the segment. (If the load module has only one segment, i.e., the root, the keys begin with load module name. If no load module name was supplied, the name is idL.)

Words 5-10 of each Tree Table are computed by the Loader or LINK.

Word 10 of the ROOT Tree Table is used by the Loader to monitor the size of the REF/BREF Tables.

*Doubleword address or # of doublewords

** Displacements from TREE

Segment Components - Standard Load Module

For each segment, the following records are built:

	Key	Record
Segment Name Concatenated with:	00 →	REF/DEF stack *
	01 →	EXPR stack *
	02 →	00 REL DICT**
	03 →	00 Control Sections
	04 →	01 REL DICT**
	05 →	01 Control Sections
	06 →	10 REL DICT**
	07 →	10 Control Sections
	09 →	Global Symbol Table(output by SYMCON or LINK)
Input File Name Concatenated with:	10 →	Internal Symbol Table (output by LOADER or LINK)

Segment Components - Paged Load Module Built by the Overlay Loader

For each segment, the expression stack and REF/DEF stack records have the same format as those for the standard load module. Relocation dictionary records are not constructed.

* Output by Loader

** These records are output by Loader for a relocatable load module.

The core images are partitioned into records of at most 512 words in length with 3-byte keys of the following format:

SEG	00	PAGE

where SEG = the TREE segment number of the segment containing the core image.
PAGE = the page number of the virtual page that will contain this record at execution time.

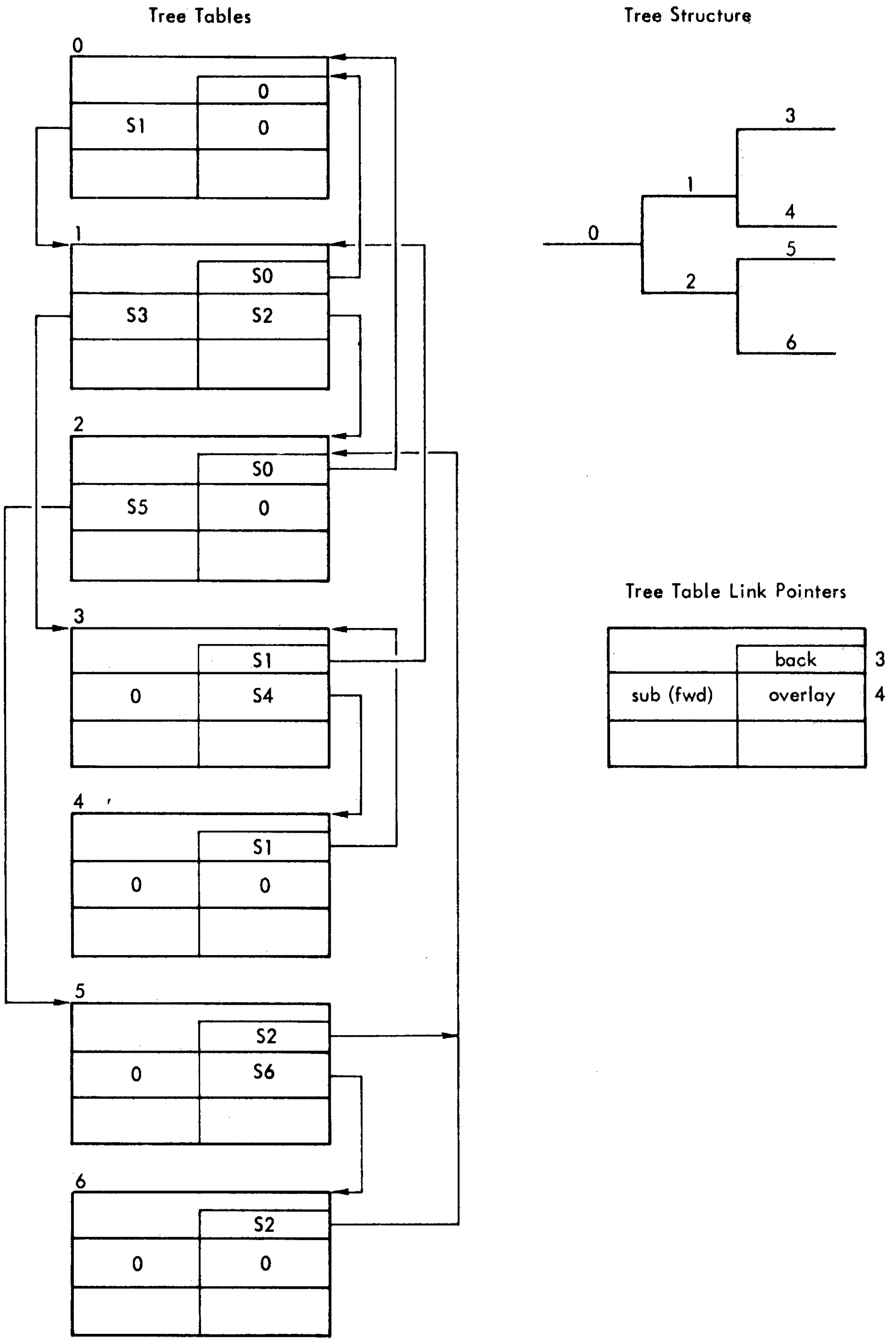
All core image records are one page in length except for the first record of an overlay segment's 00, 01, and 10 areas. The length of this record satisfies the following: at execution time, the record begins at the execution bias for this protection type and ends at the next page boundary.

Library Load Modules **Built by the Overlay Loader**

A library constructed by the Overlay Loader consists of two keyed files, :LIB and :DIC. The library load modules actually reside in one file (:LIB). :DIC is a dictionary whose keys are the text names of DEFs. The record associated with a dictionary key is the text name of the load module (within :LIB) in which that DEF is defined. Thus, in order to locate the unique group of records within :LIB which pertain to a given PREF, the Loader does a keyed READ to :DIC, the key being the PREF which is being satisfied. This keyed READ returns the library load module name within :LIB. With this information the Loader can then read the library load module records into core and merge them with the target load module.

The keys and records in :LIB are identical to those of non-library load modules (see above) except that the keys "HEAD" and "TREE" are concatenated with the TEXT load module name (to keep them unique). Each individual library load module name is "synonymous" (in a file sense) with the name :LIB.

A slight difference also exists in the REF/DEF and expression stack formats. The VALUE word of an entry in the REF/DEF stack is actually the head of a chain through the expression stack of all those entries which involve that REF/DEF. (This expedites subsequent merging of the stacks when the library is included in a user program.)



TREE Table Linking – in Relation to the Overlay Structure

REF/BREF Tables Built by the Overlay Loader

REF Table

An entry is created for every load item involving a REF defined in a higher segment. The load item is replaced by a CAL1,8 X where X is the REF Table entry address (a PLIST for the CAL).

X →

0 1 8 0	0 0 0 0
	SEG
Replaced load item.	
B load item + 1	

SEG = 17 bit address of higher segment name in Tree Table.

BREF Table

An entry is created for every branch type instruction involving a REF to a higher segment. The branch type instruction is replaced by a branch (of the same type) to the BREF entry.

BAL, R0 S:OVRLY			
*	SEG	x	ADDR

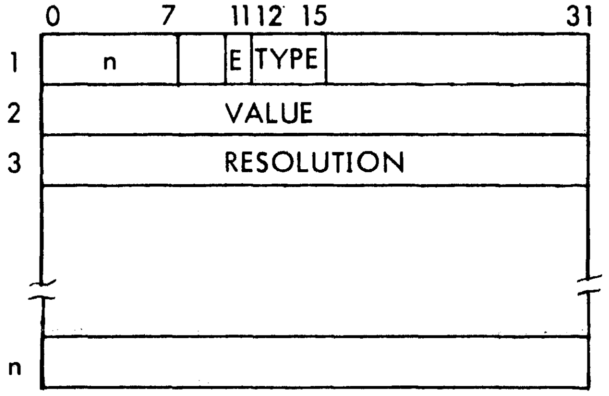
where: S:OVRLY is a system library routine

SEG = segment number (Tree Table displacement/11)

ADDR = address field of replaced instruction

*, x = indirect and index fields from replaced instruction

GENERAL REF/DEF STACK FORMAT USED BY THE OVERLAY LOADER



where:

n = number of words in this entry.

E = 1, if the entry has a VALUE

TYPE =

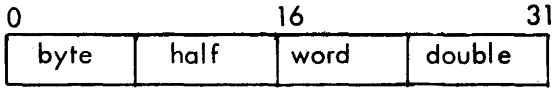
0 or 8	DEF
1	SREF
2	PREF
3 or B	Dummy Section
4 or 6	Control Section
5 or 7	Forward Reference
C or E	Page Boundary Control Section

VALUE = constant or address if the load module is not a library

or

head of a chain in the expression stack if the load module is a library

RESOLUTION = the resolution in which the VALUE is expressed. Resolution is of the form:



If the VALUE is a constant, the RESOLUTION word is 0.

If the VALUE is an address, one and only one byte of the RESOLUTION word is nonzero (viz., the appropriate byte = X'01').

If the RESOLUTION assumes a form different from either of the above, the VALUE is of mixed resolution. (In this case the load module cannot be relocated and is forced ABS.)

GENERAL EXPRESSION STACK ENTRY USED BY THE OVERLAY LOADER

	0 7	8	9	10 15		24 31
1	n	E	C	DISP	CB_1	CB_2
2	CB_3	CB_4			. . .	
"						
"	Destination					
"	Resolution					
"	Word_1					
"	Word_2					
"	⋮					
n	Word_m					

where:

n = number of words in entry

E = 1, this entry has been evaluated.
= 0, this entry has not been evaluated.

C = 0, this entry's Destination is a pointer to the REF/DEF stack.
= 1, this entry's Destination is a core expression.

DISP = number of words to Word 1.

Destination: (where the value of the entry is to be deposited) = one of the following forms, depending upon the value of C.

REF/DEF Pointer

If C = 0

0 15	16 31
Segment's Displacement in Tree Table	Displacement within segment's REF/DEF stack

Core Expression

If C = 1

0 7	8 14	15 31
Field Size	Terminal Bit Position	Word Address

Resolution: Same as REF/DEF stack.

CB_i = a control byte of the expression.

Word_i = is referenced by a control byte and is a constant or pointer to the segment's REF/DEF stack (same form as Destination where C=0).

RELOCATION DICTIONARY

If ABS is not specified on the ILOAD card each segment of Loader-built load modules will have records of relocation dictionaries (one per protection type). One relocation digit is developed for each word in the protection area.

Relocation Dictionary Digits

Digit	Type of Relocation
0	relocate the word at byte resolution.
1	relocate the word at halfword resolution.
2	relocate the word at word resolution.
3	relocate the word at doubleword resolution.
8	relocate the left half of the word at doubleword resolution.
9	relocate the right half of the word at doubleword resolution.
A	relocate both halves of the word at doubleword resolution.
E	absolute.

Notice that relocation digits exist only for items that terminate on halfword boundaries.

A load module which has an item not amenable to one of these digits is set to ABS.
Example:

```
                    BOUND  4
           ZAP      EQU      DA($)
                    GEN,8,16,8   0,ZAP,0
```

or

```
                    BOUND  4
           ZAP      EQU      $
                    GEN,3,17,12  0,ZAP,0
```

Either of these would cause the module to be set ABS since ZAP does not terminate on a halfword boundary.

Symbol Tables

Global Symbols:

A global symbol table is constructed by LINK or by SYMCON. This table is a list of correspondences between symbolic identifiers (labels) used in the original source program and the values of virtual core addresses which have been assigned to them at load time. The global symbols identify object (DEFs) within a module which may be referred to (REFed) in other modules. This table is available to DELTA, for use in debugging.

Internal Symbols:

An internal symbol table is a list of correspondences similar to the global but which applies solely within the module. It is built by LINK or the Loader for each input ROM which contains IST load items. Each internal symbol table is associated with that specific input file (ROM) and identified by its name. The internal as well as the global symbol tables are created for use by the debug processors, such as DELTA. The user has the ability under DELTA to define which set of internal symbols is to be used for specific debugging activities.

Symbol Table Format:

Both global and internal symbol tables consist of three word entries. Symbolic identifiers (labels) are limited to seven (7) characters plus count. Symbols originally longer than seven are truncated leaving the initial characters, although the original character count is retained. Symbols which are identical in their first seven characters and are of equal length occupy one position in the symbol table. The value or definition for such multiply defined symbols is the first one encountered during the linking process. Each symbol entered into the table has a type and internal resolution classification. The internal resolution types are: byte, halfword, word, doubleword, and constant. The following are the symbol types which are supplied by the object language and maintained in the symbol table: instruction, integer, EBCDIC text, short floating point, long floating point, decimal, packed decimal, and hexadecimal.

Location Symbol - code = 01

<table>
<tr><td>01</td><td>CT</td><td colspan="2">S_1</td><td>S_2</td><td>S_3</td></tr>
<tr><td colspan="2">S_4</td><td colspan="2">S_5</td><td>S_6</td><td>S_7</td></tr>
<tr><td>t</td><td>res</td><td></td><td colspan="3">value</td></tr>
</table>

where:

CT is a six-bit field containing the character count of the original symbol.

S_i are the first seven (7) characters of the symbol. Symbols with fewer than seven characters are zero filled.

t is a five-bit field where the values are:

00000 - instruction
00001 - integer
00111 - EBCDIC test (also for unpacked decimal)
00010 - short floating point
00011 - long floating point
00110 - hexadecimal (also for packed decimal)
01001 - integer array
01010 - short floating point array
01011 - long floating complex array
01000 - logical array
10000 - undefined symbol

res is a three-bit field representing the internal resolution. The values are:

000 - byte
001 - halfword
010 - word
011 - doubleword

value location symbols are always represented as a 19-bit byte resolution value.

Constants - code = 10

<table>
<tr><td>10</td><td>CT</td><td>S_1</td><td>S_2</td><td>S_3</td></tr>
<tr><td colspan="2">S_4</td><td>S_5</td><td>S_6</td><td>S_7</td></tr>
<tr><td colspan="5">value</td></tr>
</table>

where: CT and S_I have the same meaning as above

value is the 32-bit value of the constant.

Loader Control Command Table (LOCCT)

	Size of the three tables to follow	(CP-V only)

		0	1	2	3	4	5	6	7	8	9	10	11	12 15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31*
LOCCT	0	R	Q	S	X	F1	F2	F3	F4	T	Y	U	V	SL	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
	1	P1	P2	P3	P4										SYSid															

	2	Displacement from LOCCT to TREET	
	3	Displacement from LOCCT to ROMT	
	4	REF or BREF count (default = 0)	LOAD BIAS (WA) **
	5	Number of Execute Accounts (default = 1)	**
	6	FCOM Size	FCOM (DA) ***
	7	ERSTACK Size (default, X'A')	ERTABLE Size (default, X'A')
	8	TSS Size (default = X'40')	Number of UNSAT accounts (default = 0 if NOSYSLIB; = 1 (:SYS) if not)
	9	Number of READ accounts (default = 1)	Number of WRITE accounts (default = 1)
	10 11 12	Load Module Name, TEXTC followed by blanks. (default = three characters SYSid L)	
	13 14	User Account Number	
	15 16	Load Module Password (default = 0, 0)	
	17 18	EXPIRE date; 'mmdd bbyy' or 'NEVER' (BPM) 0, 0 (CP-V)	
	19 20	Library Password i.e., (PERM, LIB, password) (default = 0, 0)	
8 max.		READ Account Numbers - 2 words per account #, (default = 'ALLbbbbb')	
8 max.		WRITE Account Numbers - 2 words per account #, (default = 'NONEbbbb')	
8 max.		UNSAT Account Numbers and Passwords - 4 words per (acct #, pass). (default password = 0, 0)	
8 max.		EXECUTE Account Numbers - 2 words per acct. #, (default = 'NONEbbbb')	
1 max.		EXECUTE Vehicle - 3 words per vehicle, (default = FETCHbbbbbbb')	
		Total Number of words in TREE Tables	
TREET →		TREE Tables - 1 Table per Segment - 11 words per table (See Figure 12 for format)	
ROMT →		ROM Tables - 1 Table per ROM - 7 words per table (See Figure 11 for format)	

F1 = do not build internal symbol tables

F2 = build global symbol tables

F3 = inhibit output of segment size summary

F4 = DREF specified

P1 = "P" option for privileged processors specified

P2 = "S" option for privileged processors specified

P3 = "M" option for privileged processors specified

P4 = "X" option for privileged processors specified

* NOTE:

A = 1, UDEF specified
B = 1, NOSYSLIB specified
C = 1, REF specified
D = 1, PERM specified
E = 1, LIB specified
F = 1, M10 specified
G = 1, M100 specified
H = 1, FCOM specified if BPM. If CP-V, the OSP specified.
I = 1, ABS specified
J = 1, Assigns Read
K = 1, GO specified
L = 1, BI specified
M = 1, CSEC1 specified
N = 1, NOTCB specified
O = 1, XMEM in effect (set by the Loader in IN2), or PAGE specified
P = 1, LDEF specified
Q = 1, BREF specified
R = 1, EF specified
S = 1, CORELIB specified
T = 1, RDEF specified

bits 10 - 11
0 = no map
1 = map by NAME
2 = map by VALUE
3 = map by NAME and VALUE

X = 1, Execute Vehicle specified
Y = 1, MAPONLY specified
SL = Severity Level (default = 4)

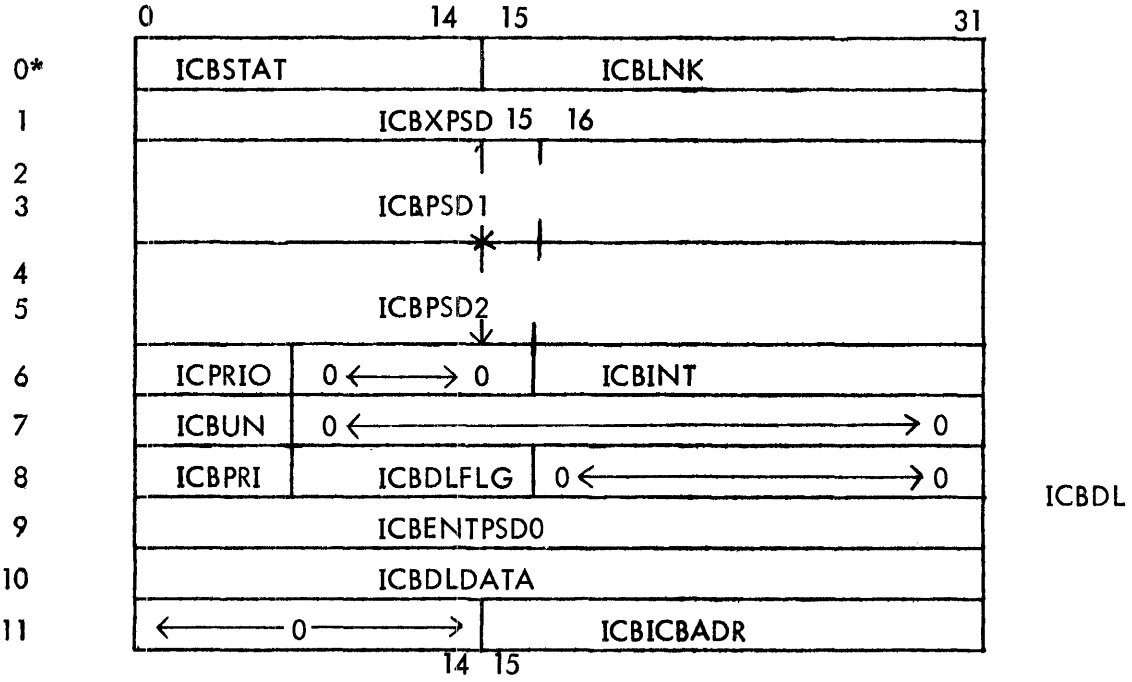
** BPM-CP-V differences in the LOCCT Tables:

	Word	BPM	CP-V
	4	LOAD BIAS field, default = 0	LOAD BIAS, Default = background lower limit WA
	5	Background lower limit	Number of Execute Accounts, Default = 1
***	6	Passed to the Loader in Register D4 (D4) = FCOM size	

REAL TIME

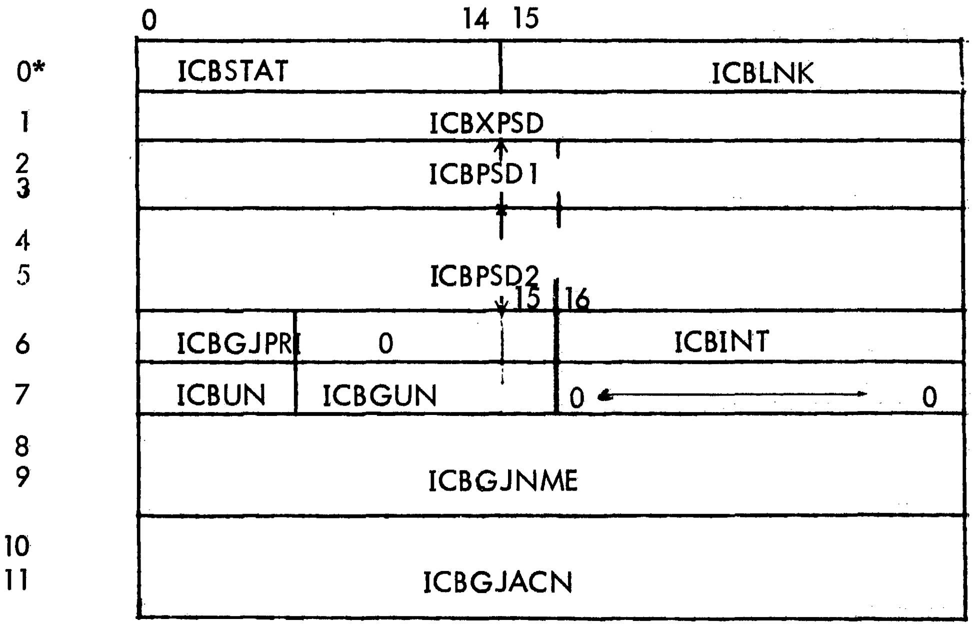
ICB TABLE FORMATS

Type I: ICB Connected to Interrupt Associated with User Task



* doubleword boundary

Type II: ICB Connected To Interrupt Associated With Ghost Job



* doubleword boundary

ICB Connected to CLOCK-3 Interrupt

	0	14	15 31	
0*	ICBSTAT		ICBLNK	
1	ICBTUN			
2	ICBCLK			
3	0 ←→ 0		ICBSYSEP	
4	0 ←→ 0		ICBBLNK	
5	0 ←→ 0			
6	ICBPRIO	0 ←→ 0	ICBINT	
7	ICBUN	0	0	
8	ICBPRI	ICBDFLG	0 ←→ 0	ICBDL
9	ICBENTPSD0			
10	ICBDLDATA			
11	← 0 →		ICBICBADR	

* doubleword boundary

ICB Field Descriptions

WORD	NAME	DESCRIPTION
0	ICBSTAT	Status Flags:

BIT POSITION	NAME	DESCRIPTION
0	ICBSTATA	set if ICB is active
1	ICBSTATC	set if CLEAR was specified via M:CONNECT.
2	ICBSTATDL	set if ICBDL is already chained into UH:DL
3	ICBSTATO	set if ONESHOT was specified via M:CLOCK
4-6	ICBSTATINT	associated interrupt status:
Bit Position		Meaning if set
4		if pseudo interrupt: trigger pending if real interrupt: has been triggered (interrupt is active)
5		enabled
6		armed
7-8	ICBSTATYP	defines ICB type: 00:: user task (Type I) 01:: ghost job (Type II) 10:: clock (Type III)
9	ICBSTATSY	set if ICB is a SYSTEM ICB, in which case, the ICB is only five words in length.

WORD	NAME	DESCRIPTION
0	ICBLNK	Chain link address
1	ICBXPSD	(Types I and II only) this word contains an XPSD instruction; this location is the effective address of the new PSD at location ICBPSD2 which is loaded as a result of the hardware interrupt occurring (in the case of a real interrupt) or the result of an XPSD instruction (in the case of a pseudo-interrupt). This technique allows the interrupt handling routine to determine which ICB is associated with the interrupt that just occurred.
1	ICBTUN	(Type III only) value (in two-millisecond units) from M:CLOCK CAL1.

2-3	ICBPSD1	(Types I and II only) storage area for old (interrupted) PSD upon the occurrence of an interrupt; this location is the effective address of the XPSD instruction in the hardware interrupt location associated with the ICB.
2	ICBCLK	(Type III only) number of clock ticks remaining before an interrupt will occur for the user associated with the ICB.
3	ICBSYSEP	(Type III only) if the ICBSTATSY bit of ICBSTAT is set, then this is a SYSTEM ICB and is only five words in length (0-4); ICBSYSEP contains the address of the system entry point which will be entered upon the expiration of the elapsed time specified in ICBTUN.
4	ICBBLNK	(Type III only) Back link for ICBs currently chained into ICBCLKHDR.
4-5	ICBPSD2	(Types I and II only) contains the new PSD which will be loaded as a result of the execution of the XPSD instruction in the hardware interrupt location associated with the ICB; the instruction address of this PSD will cause the XPSD instruction at ICBXPSD to be executed.
6	ICBPRIO	(Types I and III only) used to save the associated user's priority (his old priority) at the time of the interrupt associated with this ICB.
6	ICBGJPRI	(Type II only) contains the execution priority of the ghost job to be scheduled as the result of the interrupt associated with the ICB.
6	ICBINT	Contains the hardware or pseudo-interrupt location associated with this ICB.
7	ICBUN	The internal number of the user associated with this ICB (the owner).
7	ICBGUN	(Type II only) the internal number of the ghost job itself (user number).
8	ICBPRI	(Types I and III only) contains the execution priority of the task to be scheduled as a result of the interrupt associated with this ICB.

8 ICBDLFLG (Types I and III only) contains the value X'82' which is a code within the DO-LIST table which, in this case, is a subtable of the ICB.

8-9 ICBGJNME (Type II only) contains the TEXTC-formatted name of the ghost job to be placed into execution as a result of the interrupt associated with this ICB.

9 ICBENTPSD0 (Types I and III only) contains an image of word 0 of the PSD to be loaded prior to going to the user upon the occurrence of the interrupt as follows:

0	8	9	10 15	31
0 0	M S	1	0 0	entry address

where:

M, S } reset if MASTER was specified via M:CONNECT/M:CLOCK.

<u>address</u> is the virtual address at which the user is to be given control upon the occurrence of the interrupt; specified via M:CONNECT/M:CLOCK.

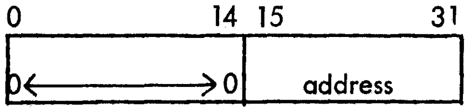
10-11 ICBGJACN (Type II only) contains the TEXT-formatted name (left-justified, trailing blanks) of the account in whose directory may be found the ghost job to be placed into execution as a result of the interrupt associated with this ICB.

10 ICBDLDATA (Type I) contains the interrupt location associated with this ICB (i.e., same as ICBINT). (Type III) same as ICBTUN (i.e., requested elapsed time in two-millisecond units).

11 ICBICBADR (Types I and III) contains the address of word 0 of this ICB.

ICB Chain Headers

ICB Chain Headers are of the form



where address is the word address of word zero of an ICB which in turn points to the next ICB in the chain (ICBLNK). The last ICB in the chain has an ICBLNK of zero.

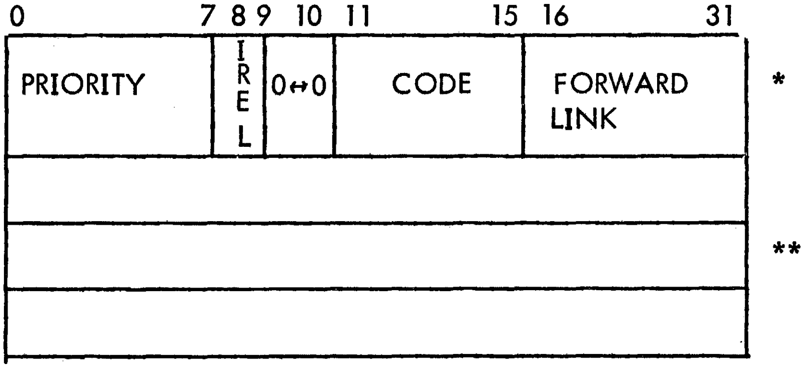
RTICBHDR

This header is built and DEFed by SYSGEN-PASS2 and contained in the M:FRGD module; if this is a real-time system, RTICBHDR will point to the first of a user-specified number of available ICBs. If this is a non-real-time system, RTICBHDR will be set to -1.

RTICBCLKHDR

This header is assembled into and DEFed in the TABLES module; this chain header always points to the five-word SYSTEM ICB associated with the 1.2-second time-of-day routine. RTICBCLKHDR also contains a back link which points to the last ICB chained into RTICBCLKHDR.

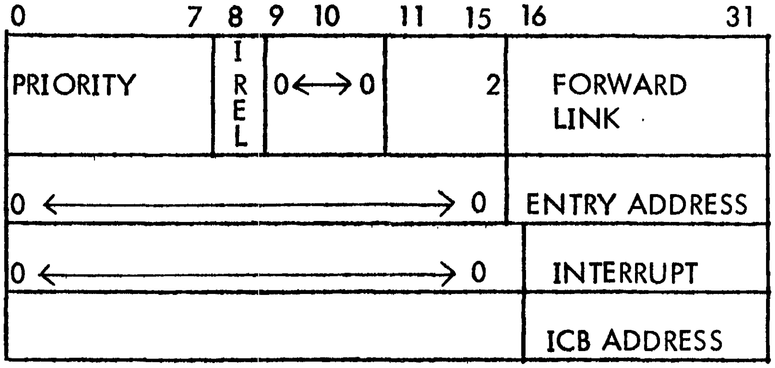
DO LIST BLOCK FORMATS



IREL Do not release block to free chain when done. Set when block is contained in an ICB.

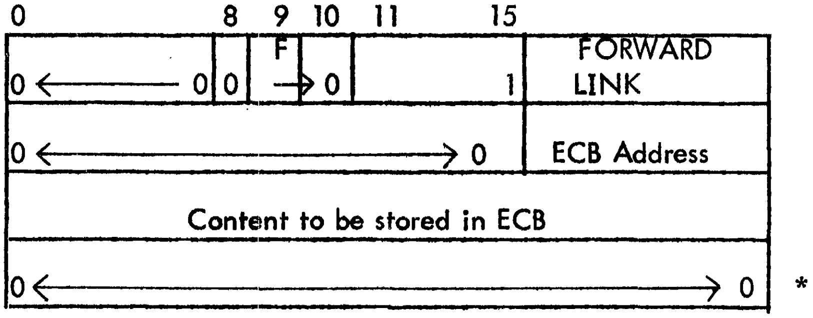
*Header
**Data Words dependent on code value

Do List Format for Interrupt Entry



IREL ⟹ Inhibit release of block. Set if block contained in ICB.

Do List Format for Post-ECB Entry

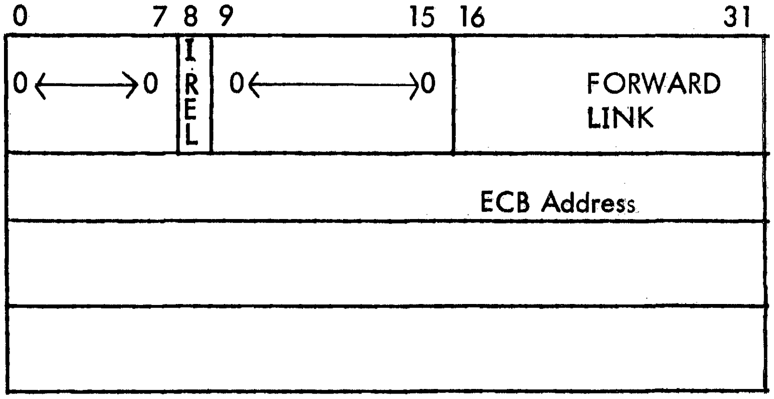


*(unused)

0 = WORD
1 = DW
2 = HW
3 = BYTE

F = posted entry or wait list

Wait List Block Format



IREL	Inhibits release of block. Used when block is contained within a larger block.

M:FRGD MODULE

For a real-time system, one having a :FRGD command present, the following tables will be generated:

```
                BOUND 8
RT:GINTP        DATA        0,0
                GEN,12,20    0,RT:GINT
                GEN,8,1,6,16   X'17',:9,0,:9,0

RT:UINTP        DATA          0,0
                GEN,12,20      0,RT:UINT
                GEN,8,1,6,1,16   X'17':9,0,:9,0

                DEF       RT:GINTP, RT:UINTP
                REF       RT:GIN1, RT:UINT

                DEF       ICBSIZE, ICB, LASTICB, NINTS
ICBSIZE         EQU       12

ICB             EQU       $

I               DO        NINTS                   No. of interrupts

                DO        I=NINTS
                DATA      0,0
                ELSE
                DATA      $+12,0
                FIN
                DATA,8    0,0,0,0,0
                FIN
LASTICB         EQU       $

RTICBHDR        EQU       $
                DO        0=NINTS
                DATA      0
                ELSE
                DATA      ICB

                DEF       RTICBHDR, RESDF
```

```
             DEF      RESDF, CRESDF, RESDFP, DYNRESDF, MDYNRESDF
RESDF        DATA     SIZE  (number of RESDF pages)
CRESDF       DATA     SIZE  (number of RESDF pages)
RESDFP       DATA     ADDRESS (RESDF address)
DYNRESDF     DATA     0
MDYNRESDF    DATA     PAGES (maximum number DYNRESDF pages)
             DEF      PPTABLE, PPTABLSZ
PPTABLE      EQU      $
             DO1      2+(#SEGMENTS-1)
             DATA     0
PPTABLSZ     EQU      $-PPTABLE
             DEF      PP:UPPH, PP:UPPT, PP:UPPC
PP:UPPH      DATA     0
PP:UPPT      DATA     0
PP:UPPC      DATA     0
             DEF      PPTABDSK2
PPTABDSK2    DATA     0
```

```
             DEF          INTLB1, INTLB2, INTLBSIZ, INTLB3
INTLB1       DATA,2       0
             DATA,2       'LA'     )
             DATA,2       'BB'     } text interrupt labels
               :                   )
               .
INTLBSIZ     EQU          HA ($) - HA(INTLB1)

INTLB2       DATA,2       0
             DATA,2       X'65'     \
             DATA,2       X'1022'   ( interrupt addresses
               :                    )
               .
INTLB3       DATA,1       0
             DATA,1       X'65'      default (SYSGEN-specified) execution priority
             DATA,1       X'82'
               :
               .
```

Defaults:

NINTS = 0   Address = X'10000'   #SEGMENTS = 1
SIZE = 0   PAGES = 0

For a non-real-time system, the following subset will be generated:

```
             DEF          RTICBADR
RTICBHDR     DATA         -1
```

M:IMC MODULE

a. Initialize all entries in UB:US to 31

b. Delete UH:TS.

c. ADD UH:DL, a halfword table parallel to other user tables and initiated to zero.

d. Add UH:WL, a halfword table initialized to zero.

e. Add UP:PRIO, a byte table initialized to X'FF'.

f. Add UB:PRIOB, a byte table initialized to zero.

g. Add UB:NECB, a byte table initialized to zero.

h. Generate the cells:

OPTION	DEF	DESCRIPTION	DEFAULT
BPRIO	SL:BPRIO	Batch default priority	X'FC'
OPRIO	SL:OPRIO	On-line default priority	X'FC'
GPRIO	SL:GPRIO	Ghost default priority	X'FC'

ENQ/DEQ TABLES

ENQ/DEG uses the new Queue Table (QT), U:MISC, and the JIT flag ENQ (J:ABC bit 23). U:MISC is set with sleep time whenever a user is put into the sleep state, and JIT:ENQ is set when a user request ENQ and checked by STEP and SSS. The QT is a pool of double words that can be used to contain various kinds of data.

The hierarchy of the Queue Tables is as follows:

a. The first level of entries is the pool of empty entries.

b. The second level of entries is the names of all the queues (qnames) for resources. These are referred to as Q entries.

c. The third level of entries is the user numbers of all the users of a queue (qname). These are referred to as U entries, and the first one is Uhead.

d. The fourth level of entries is the names of all the sub-queues (snames) for elements. The order of the sub-queues is first, NULL, if present then ALL, if present, then those with regular names. These are referred to as S entries.

e. The fifth level is the queues of users for the various resource/ elements. These may be referred to as SQ or user entries, depending on their context of an S chain or a U chain.

The third and fourth levels might be more appropriately thought of as two dimensions at the same level, i.e., a queue can be looked upon as the users of sub-queues or sub-queues of users. Both the third and fourth levels contain the heads of threads through the fifth level.

Double words, or contiguous blocks of double words, are also used as appendages to second and fourth level entries to contain names (qname or sname) that exceed three characters in length. Double words are also used as fifth level appendages to contain the ECB address for pending requests.

The formats of the various types of entries are as follows.

QHEAD ENTRIES

The Q, U, or S entries consist of three fields - the half-word AHVP and OHHP, and the full word QHNAME.

0 15	16 31	32 63
QHHP	QHVP	QHNAME

QHHP

Queue Head Horizontal Pointer

Entry 0 contains the double word index in QT of the first empty entry. Other first level entries contain the double word index of the next empty entry (zero is terminator).

Level 2 (Q) entries contain the double word index in the QT of the next qname (Q) entry (zero is terminator).

Level 3 (U) entries contain the double word index in the QT of the next user number (U) entry for this queue name (zero is terminator).

Level 4 (S) entries contain the double word index in QT of the next sub-queue name (sname) entry for this queue (zero is terminator).

QHVP

Queue Head Vertical Pointer

Entry 0 contains the double word index in QT of the first queue name (qname) entry, i.e., the first entry in level 2.

Level 2 (Q) entries contain the double word index in QT of the first QHEAD level 3 (U) entry, that is, the head of the user lists for this queue (qname).

Level 3 (U) first entry contains the double wrod index in QT of the first sub-queue name (sname) entry. Other level 3 entries contain the double word index in QT of the first SQ entry for each user (user chain).

Level 4 (S) entries contain the double word index in QT of the head of the sub-queue (S) entry (SQ chain).

SQ ENTRIES

The SQ entries consist of five fields - the one-byte QQUN and QQF, and the half-word QQHP, QUVP, and QUHP.

0 15	16 31	32 47	48 55	56 63
QQHP	QUHP	QUVP	QQF	QQUN

QQHP

Queue Queue Horizontal Pointer

Contains the double word index in QT of the next entry in this sname queue (S chain).

QUHP

Queue User Horizontal Pointer

Contains the double word index in QT of the next sname queue entry for this user (U chain). If QQF bit 6 is set, that next entry is an ECB address entry relating to the same sname queue.

QQUN

Queue Queue User Number

Contains the user number of the user at this position in the sname queue.

QUVP

Queue User Vertical Pointer

Contains the double word index in QT of the sname (S or level 4) entry for this queue. During a deadlock check, bit 0 is set to mark the path of checking.

QQF

Queue Queue Flags

Bit 0	1 if request was for SHARE.
Bit 1	1 if access has been allowed for this user (allocated).
Bit 2	1 if the request was NOWAIT.
Bit 3	1 if a JOB entry.
Bit 4	1 if this user is temporarily blocked from access because he was not a user of this queue (qname) when an EXCL or ALL user was denied immediate access.

Bit 5 1 if the user is currently asleep pending receiving access in this sub-queue (sname).

Bit 6 1 if the next entry for this user (object of QUHP) is an ECB address entry.

Bit 7 1 is this is a SHARE entry and there is also an EXCL entry pending upgrade on this sub-queue for this user.

NOTE: If bit 1 is set the queue entry is "allocated"; if bit 4 is set the queue entry is "blocked"; and if neither is set the queue entry is "pending".

QECB ENTRIES

The QECB entries consist of three fields, the half-word QUPP andQUHP, and the 17 bit QUEA.

0 15	16 31	32 46	47 63
QUPP	QUHP		QUEA

QUPP Queue User's Primary Entry Pointer

Contains the double word index in Qt of the primary portion of this entry - points back to the entry whose QUHP points to it.

QUHP Queue User Horizontal Pointer

Same as under SQ entries above.

QUEA Queue User ECB Address

Contains the ECB address for user QQUN for sub-queue QUVP.

QNAME ENTRIES

The QNAME entries consist of a single, variable length field. One to four consecutive QT entries are used to contain queue (qname) or sub-queue (sname) names when the names do not fit in the QHNAME field of the Q or

S entry. One double word is used for names of four to seven characters, two double words for names of eight to fifteen characters, three double words for names of sixteen to twenty three characters, and four double words for names of twenty four to the maximum thirty one characters.

REMOTE PROCESSING TABLES

All the Remote Processing Tables are named RB$\begin{bmatrix}B\\H\\D\end{bmatrix}$:xxxx and are indexed by DCT index. Most exist only for RB device DCT indices although a few (marked with * below) have one additional entry for LOCAL use indexed LCLX a value defined by SYSGEN. Note that this means that the tables usually do not start where their names are defined since the names and EQUed backwards from the tables to make DCT indexing possible. A doubleword, RBLIMS, is defined in the monitor by SYSGEN such that the following test determines whether a device is RB (and thus the tables are meaningful):

```
CLM,DCTX      RBLIMS
BCR,9         ITS$REMOTE
BCS,9         ITS$LOCAL
```

REMOTE PROCESSING TABLES IN CORE

RB:FLAG	A word table containing flag bits which control the basic flow of Remote Processing. See the bit descriptions below.
RB:BUF	A word table containing the address of the context area for IRBT and 2780. IRBTs have a page of context; 2780s one-half page. For 7670, holds number of retries.
RBD:WSN	A doubleword table containing the Workstation Name (WSN) when the line is logged on or if it is set by SYSGEN (WSN option) or KEYIN (IRBLOG).
RBH:ACK	A halfword table used as a buffer to read short messages (ACK, NAK, ENQ, etc.) from the RBT or IRBT.
RBB:ID	A byte table containing the RBID of logged on lines. Often used to tell whether a line is logged on (≠ 0) or not.
RBB:HFE, RBB:HOU RBB:HIN	Byte tables containing the heads of the free, outputting, and inputting chains of symbionts through STB:LNK, respectively for IRBTs. For RBTs (7670, 2780) RBB:HFE is the head of the chains of <u>all</u> symbionts for this line (CR is always first), and whichever of RBB:HIN and RBB:HOU is non-zero is the input or output symbiont that currently is using the line. Note: for IRBTs, the OC device is always the RBB:HOU chain.
RBB:BCB	A byte table that contains the BCB count next expected from an IRBT line. It is used in 7670 to count errors for ERROR MAX, and unused in 2780.

RBB:LPZ, RBB:CPZ	Byte tables used in 7670 and 2780 to keep the current max printer and puch record length. Unused in IRBT.
RBB:SPC	A byte table used in 7670 to help keep track of the number of formats to be performed before a print. Unused in 2780 and IRBT.
RBB:SFC	A byte table used in 7670 to preserve the current function when a warning BELL is being sent. Unused in 2780 and IRBT.

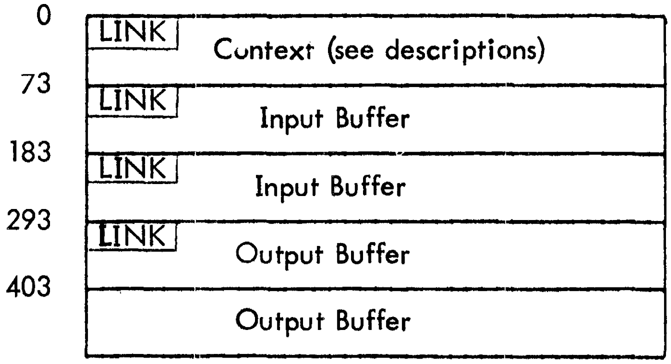
REMOTE PROCESSING TABLES IN RBBAT

*RB:MFAD	A word table containing the address of the current message file buffer. Zero if none.
*RB:SPMF	A word table containing the used byte count of space in the message file buffer.
*RBH:MFX	A halfword table containing the BH:LINK entry index for the message file under construction.
RBB:MXP	A byte table containing the maximum priority for jobs submitted from this WSN. Also used as the priority for any direct passed output.
RBB:DSM	A byte table containing the device selector mask for this WSN.
RBB:SMD	A byte table containing the symbiont index of the system message device for this WSN.

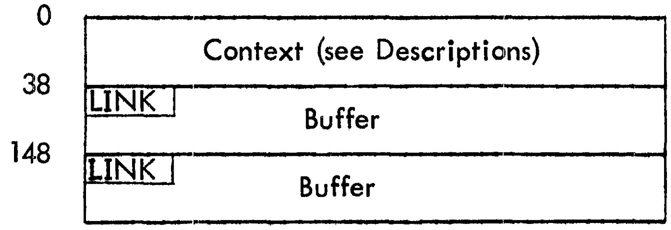
RB:FLAG STRUCTURE

Bit	Name	7670	2780	IRBT	Meaning
0	BPBIT	x	x		Block protect toggle
1	IGBIT	x	x	x	Cards after IFIN were ignored
2	MORBIT	x			Waiting for next chunk of deck
3	HUBIT	x	x	x	Line hung up
4	PUNBIT	x			Punching is allowed
5	DCBIT	x	x	x	WSN specified at SYSGEN
6	HASPBIT	x	x	x	IRBT line
7	SLVBIT			x	We are slave
8	ALBIT	x	x	x	RBLOG keyin done
9	XP1BIT			x	X1 specified in SUPER
10	2780BIT	x	x	x	2780 line (may be changed to IRBT at logon)
11	IBMBIT			x	N3 specified in SUPER
12	DIALBIT	x	x	x	Dial specified at SYSGEN
13	EDISBIT	x	x	x	ERROR MAX on line
14	OFFBIT	x	x	x	Do not connect line (BRX) - Set except at logon for IRBT
15	RBXBIT	x	x	x	Disconnect line now
16	DUPBIT	x	x	x	1 = Full duplex
17	DISCBIT	x	x	x	Disconnect when output done
18	LOFBIT	x	x	x	RBDISC sent (TEMP)
19	SYSBIT	x	x	x	:SYS jobs legal
20	HALBIT	x	x	x	HOLD all flag
21	CLKBIT			x	Wait before ACKO-idle
22	ACTBIT	x	x	x	Line logged on
23	CRTBIT	x			RBBAT disables RBSSS
24	XP2BIT			x	X2 specified in SUPER
25	OADBIT	x	x	x	Set OFFBIT after disconnect
26	FIABIT	x	x	x	RBCC altered stream status
27	SSSBIT	x			Inputting with output suspended
28	LIPBIT	x	x	x	Logging on
29	FINBIT	x			FIN has been read
30	EMBIT	x	x		1 = NOEM specified
31	OBBIT			x	Old BCB was read
31	FRBIT	x			Initial read of file

IRBT CONTEXT PAGE



2780 CONTEXT 1/2 PAGE



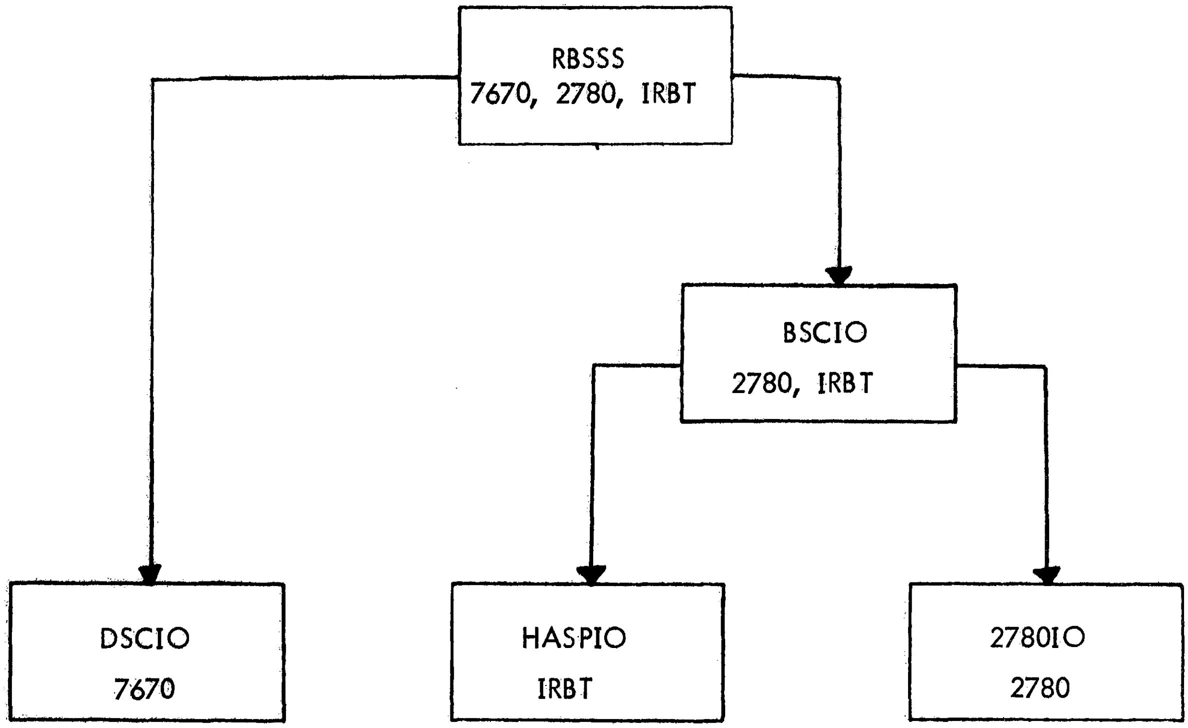
IRBT

```
**************  CONTENT POINTERS *******************
*
FNSV     EQU      0                 PREV INSYMX FOR FIN READ
HPB      EQU      1                 HEAD PARSE BUFFS
HBB      EQU      2                 HEAD BUILD BUFFS
HRB      EQU      3                 HEAD READ BUFFS
HWB      EQU      4                 HEAD WRITE BUFFS
CPB      EQU      5                 CURRENT BSC$PRS BUFF
CBB      EQU      6                 CURRENT BSC$BLD BUFF
CRB      EQU      7                 CURRENT READ BUFF
CWB      EQU      8                 CURRENT WRITE BUFF
POP      EQU      9                 POINTER OF BSC$PRS
POB      EQU      10                POINTER OF BSC$BLD
PIP      EQU      11                BSC$PRS IN PROGRESS
CBCB     EQU      12                CUR. OUTPUT BCB
OCRCB    EQU      13                INPUT RCB FOR OC
RBC      EQU      14                REMAINING BYTE COUNT (OUT)
CCRC     EQU      15                CUR. CRC (OUT)
CURX     EQU      16                CUR. USER (OUT)
BIP      EQU      17                BSC$BLD IN PROGRESS
FBF      EQU      18                FORCE BUFFER FULL
CIP      EQU      19                CONTROL IN PROGRESS
FCSI     EQU      20                COMB. INPUT FCS
CFC      EQU      21                CUR. FUNCTION CODE
BUT      EQU      22                BACKUP TOGGLE
BIN      EQU      23                BINARY REC RCVD.
CURF     EQU      24                LAST FAILING USER (OUT)
LTYC     EQU      25                LINE TYC
RST      EQU      26                SUSPENDED USERS TO RESTART
SAB      EQU      27                SUDCK ABORTS BSC$BLD
PSP      EQU      28                FUNC PERM POINTER
EOF      EQU      29                EOF(S) IN THIS BLOCK (OUT)
CONT     EQU      30                CONTINUATION BUFF ADDR
CONTCK   EQU      31                CONT SYMB INDEX
CONTCNT  EQU      32                CONT BYTE COUNT
SCRC     EQU      33                FUNC PERM CRC
HIP      EQU      34                HANG UP IN PROGRESS
BKD      EQU      35                SYMBIONT ACTUALLY BACKED UP
TTYP     EQU      36                0=IRBT, 1=2780
RTRY     EQU      37                NUMBER OF RETRIES
SPB      EQU      38                SPECIAL BUFFER WHEN WAB SENT
```

2780

```
*****************************************#
*                                                   CONTEXT POINTERS****
FNSV      EQU       0
HPB       EQU       1                               HEAD PARSE BUFFS
HBB       EQU       2                               HEAD BUILD BUFFS
HRB       EQU       3                               HEAD READ BUFFS
HWB       EQU       4                               HEAD WRITE BUFFS
CPB       EQU       5                               CURRENT PARSE BUFF
CBB       EQU       6                               CURRENT BUILD BUFF
CRB       EQU       7                               CURRENT READ BUFF
CWB       EQU       8                               CURRENT WRITE BUFF
POP       EQU       9                               POINTER OF PARSE
POB       EQU       10                              POINTER OF BUILD
PIP       EQU       11                              PARSE IN PROGRESS
OBCB      EQU       12                              *UNUSED IN 2780*
OCRCB     EQU       13                              *UNUSED*
RBC       EQU       14                              REMAINING BYTE COUNT (OUT)
CCRC      EQU       15                              CUR CRC (OUT)
CURX      EQU       16                              HOLDS O.S.UNTIL READY
BIP       EQU       17                              BUILD IN PROGRESS
FBF       EQU       18                              FORCE BUFFER FULL
CIP       EQU       19                              CONTROL IN PROGRESS
FCSI      EQU       20                              TEMP HOLD TO CVT VFC
CFC       EQU       21                              CUR. FUNCTION CODE
BUT       EQU       22                              *UNUSED*
BIN       EQU       23                              *UNUSED*
CURF      EQU       24                              # OF RECS IN CUR BLK
LTYC      EQU       25                              LINE TYC
RST       EQU       26                              *UNUSED*
SAB       EQU       27                              TURNS OFF OUTPUT TIL READY
PSP       EQU       28                              *UNUSED*
EOF       EQU       29                              EOF IN CUR BLK (OUT)
CONT      EQU       30                              SYMB OPERATION IN PROG
CONTCK    EQU       31                              # OF REC ALLOWED/BLK SRB=1MRB=FF
CONTCNT   EQU       32                              *UNUSED*
SCRC      EQU       33                              *UNUSED*
HIP       EQU       34                              HANG UP IN PROGRESS
BKD       EQU       35                              *UNUSED*
TTYP      EQU       36                              2780=1,IRBT=0
RTRY      EQU       37                              NUMBER OF RETRIES
SPB       EQU       38                              *UNUSED*
```

REMOTE PROCESSING MODULE STRUCTURE



SCREECH CODE: #FF-00 CALLED FROM: BOOTSUBR

MESSAGE: OPERATOR INITIATED RECOVERY

REGISTERS:

R0	=	Sigma 6-7	11
		Sigma 9	15
R1	=	Sigma 6-7	Not changed - the contents are the same as when the recovery was initiated.
		Sigma 9	Either not changed or 0.
R2-R15	=	Not changed	

REMARKS: Called from BOOTSUBR on a boot from disk when sense switch 3 is set.

* * *

SCREECH CODE: #404 CALLED BY: SCHED

MESSAGE (ON OC): THAT'S ALL FOLKS!!: OPERATOR INITIATED SHUTDOWN

REGISTERS: Not Displayed.

REMARKS: Called by SSS after the system has quieted down following a ZAP key-in. Recovery then saves some system tables, but does not enter the system restoration phases. No recovery dump file is written.

* * *

SCREECH CODE: #01 CALLED FROM: SCHED, MM

MESSAGE: USERS - PAGE CHAIN INCONSISTENT

REGISTERS: Case 1, called from SCHED

R0 = 0 if circular or unlinked chain; otherwise, the Link # index in chain
R1 = Link register
R2 = Next page chain link
R4 = User Being Scheduled
R7 = Address of Chain Head, Tail, and Count Table
SR4 = Offending page number

Case 2, called from MM (T:XPGVI)

R1 = 0
R3 = Physical page number
R7 = Virtual page number

REMARKS: Can't find requested virtual page in user virtual map chain (JB:LMAP).

1. See T:PGCHK in SCHED.
2. Effective when SS1 set.

* * *

SCREECH CODE: #02 CALLED FROM: SCHED

MESSAGE: REPORTED EVENT INCONSISTENT WITH USER'S CURRENT STATE

REGISTERS:

*R3 = Previous state
*R4 = User number (T:RE,T:RCE)
*R5 = User number (T:RUE)
*R6 = Event number
*R7 = Line number (T:RCE)
SR4 = Rtn. Addr. for reschedule

REMARKS: *Contents dependent upon called entry point; if R4 = S:CU, call was T:RE; if R7 is line number of user in R4, call was T:RCE; if R4 = R5, entry is T:RUE.

* * *

SCREECH CODE: #0A CALLED FROM: DPSIO, TSIO

MESSAGE: OPCODE IN SWAP COMMAND CHAIN IS INVALID

REGISTERS: Case 1, command list security checks - SS4 set.

R1 = Incorrect command list order code if not equal R3
R2 = Incorrect command list entry address (IOCD)
R3 = Order code of first IOCD in command list
R4 = Swap device index
R6 = Command list beginning address
R7 = Swapper function code

Case 2, Unrecoverable read error during inswap

R1 = Inswap user number
R7 = DCT index
SR1 = Incorrect command list entry address (IOCD)
D1 = Order code

* * *

SCREECH CODE: #0B CALLED FROM: DPSIO, TSIO

MESSAGE: INCORRECT ORDER CODE IN SWAP COMMAND LIST

REGISTERS: R1 = Incorrect order code, not seek
R2-R7 = See case 1, S/C #0A above.

REMARKS: SS4 must be on for check,

* * *

SCREECH CODE: #0C CALLED FROM: DPSIO, TSIO

MESSAGE: ATTEMPT TO SWAP MONITOR'S MEMORY

REGISTERS: R1 = Buffer address
R2-R7 = See case 1, S/C #0A above.

REMARKS: SS-4 must be on for check.

* * *

SCREECH CODE #0D CALLED FROM: TS

MESSAGE: HALT FLAGS MISSING IN SWAP COMMAND LIST

REGISTERS:
R0 = FLAGS byte from TIC command
R1 = TIC order code
R2-R7 = See case 1, S/C #0A above.

REMARKS: SS-4 must be set to check. FLAGS must not have command chaining set and must have interrupt on zero byte count or channel end set.

* * *

SCREECH CODE: #0E CALLED FROM: TSIO

MESSAGE: I/O REQUEST WITH NULL COMMAND LIST

REGISTERS:
R4 = Swap device index
R6 = Command list beginning address
R7 = Swapper function code

REMARKS: Not checked for pack-only swappers.

* * *

SCREECH CODE: #0F CALLED FROM: DPSIO, TSIO

MESSAGE: INPUT FUNCTION CODE IS INVALID

REGISTERS:
R2 = Swapper function code
D4 = X'0F

REMARKS: SS-4 must be on to check. Function code not between one and five exclusively.

* * *

SCREECH CODE: #10 CALLED FROM:

MESSAGE: BAD COC BUF POOL, OR BAD BUF ADR ON RELEASE REQUEST

REGISTERS:
R2 Logical line number
R4 Buffer address
R6 Return address from buffer return call

REMARKS:

1. On a COC buffer release, an invalid relative buffer address was specified (address 15 or HRBA*4+15).

2. On a COC buffer GET or RELEASE, an invalid relative buffer address was found in the free pool chain. If the COC module was assembled with the COCGBUG and COCPBUG flags set (normally they're not), and sense switch 4 is set, the entire free pool chain is checked on each PUT and GET operation. (The R4 and R6 contents listed above are valid only at entry and exit times.)

* * *

SCREECH CODE: #11 CALLED FROM: COC

MESSAGE: INVALID INTERNAL CONTROL CODE TRANSLATE REQUEST

REGISTERS:
*R1 = DCB address
R2 = Line number
R5 = Character
*R7 = Byte address of user buffer
*SR2 = Rtrn address
SR3 = Output translation table address

REMARKS: Cause is translate table error (e.g., 2741 N/L on non-2741 line), or bad input buffer chain.

*Not always set.

* * *

SCREECH CODE: #12 CALLED FROM: COC

MESSAGE: COC - BAD INPUT BUF LINKAGE ON RELEASE REQUEST

REGISTERS:
*R0 = Removal point
*R1 = DCB address
R2 = Line number
*R3 = COC #
*R4 = Current release point
SR3 = Output translate table addr.
SR4 = Caller's return, RTN+1 = activation
D3 = Return address

REMARKS: The COC input buffers are being released, and there is a conflict between the insertion and removal points and the chain.

* not always set

* * *

SCREECH CODE: #13 CALLED FROM: COC

MESSAGE: COC - OUTPUT BUF LINKAGE OR CHARACTER COUNT BAD

REGISTERS:
*R1 = DCB address
R2 = Line number
R3 = COC number
R4 = Removal point (usually neg.)
R5 = Character
SR4 = Output count, usually = -1

REMARKS: Inconsistent output count and buffers, may be caused by extended interrupt pulse or clobbered COC tables - usually COCOC, COCOI, or COCOR.

* not always set

* * *

SCREECH CODE: #14 CALLED FROM: THEUNCOC

MESSAGE: COC ROUTINE CALLED IN NON-COC SYSTEM

REGISTERS:
SR2 = BAL adr if 14-03
SR4 = BAL adr if 14-01 or 14-02
D4 = BAL adr if 14-04

REMARKS: The subcode indicates which routine was called:

14-01 COCIO
14-02 COCOFF
14-03 COCSENDX
14-04 ECHOCR2

* * *

SCREECH CODE: #17 CALLED FROM: IOQ

MESSAGE: INVALID DISK ADDRESS PASSED FOR AN I/O OPERATION

REGISTERS:
R1 = IOQ7,3 = DCTX=0
*R2 = DCB address
R3 = Queue index
*SR1 = Seek address from CDA,R2
SR4 = IOQ +.175
D4 = X'17'

REMARKS: Caused by invalid DCT index. If on a RAD/disk, DSCVT will have been called indicating SR1 setup.

* * *

SCREECH CODE: #19 CALLED FROM: BUFF

MESSAGE: INVALID BUFFER ADDRESS PASSED FOR RELEASE

REGISTERS:
R1 = Index to BUFLIMS
R2 = Head of respective buffer pool
R5 = JIT address
SR4 = LINK Return address
D3 = Buffer address
D4 = X'19'

REMARKS: Occurs both on releasing and acquiring buffers of most types (CPOOL, SPOOL, and MPOOL).

* * *

SCREECH CODE: 1A CALLED FROM: CLS

MESSAGE: ACCOUNT DIRECTORY INACCESSIBLE

REGISTERS: No registers significant.

REMARKS: Account directory is bad, and Monitor is unable to reconstruct it. All files are lost.

* * *

SCREECH CODE: #1B CALLED FROM: SWAPPER

MESSAGE: USERS PAGE CHAIN NON ZERO AT SWAP COMPLETION

REGISTERS:
R1 = Inswap user number (S:ISUN)
R2 = Physical byte address of JIT
R3 = UB:US,1 (user state)
R4 = Physical page head
R5 = Physical page tail
R6 = Physical page count
SR4 = Count of swapper free page chain (S:FPPC)

REMARKS: Swappers' free page pool must be nonzero at end of inswap. S:FPPH, S:FPPT contain head and tail of pages just allocated to the inswap user.

* * *

SCREECH CODE: #1D CALLED FROM: T:OV

MESSAGE: REQUESTED OVERLAY NUMBER IS OUT OF RANGE

REGISTERS:
R2 = Overlay name
R3 = Overlay name
R4 = 0
D4 = X'1D'

REMARKS: Requested monitor overlay not in shared processor table.

* * *

SCREECH CODE: #1F CALLED FROM: SWAPPER

MESSAGE: NOT ENOUGH PAGES TO PERFORM THIS SWAP

REGISTERS: R3 = Page to release SR1 = Deficient page count

* * *

SCREECH CODE: #21 CALLED FROM: MM

MESSAGE: ATTEMPT TO SET ACCESS CONTROLS ON NON-EXISTENT VIRTUAL PAGE

REGISTERS: R6 = Number of pages to set R7 = Virtual page number SR4 = Link register

* * *

SCREECH CODE: #22 CALLED FROM: TYPR

MESSAGE: PRIVATE VOLUME ALLOCATION ERROR

REGISTERS:
R2 = SN Count
R3 = DCB volume no.
R4 = SYSID (0 = EXCL. use)
R6 = DCB address
SR4 = RTN. adr.
D2 = DCB:SNT
D4 = X'22'

REMARKS: Error in allocation; specified entry in AVRTAB not found or with bad flags.

* * *

SCREECH CODE: #23 CALLED FROM: CSE57, CSE59, CSEX560,CSECOM

MESSAGE: INVALID ENTRY TO CSE HANDLERS

REGISTERS: None significant

REMARKS: Entry was made to an unused slot of the CSE branch vector for this machine.

* * *

SCREECH CODE: #24 CALLED FROM: CSEHAND

MESSAGE: INSTRUCTION EXCEPTION TRAP IN MASTER MODE

REGISTERS: All relevant information in in-core error log buffers.

REMARKS: Trap X'4D' while in master mode. Slave mode trap causes normal user job step abort.

* * *

SCREECH CODE: #25 CALLED FROM CSEHAND

MESSAGE: UNRECOVERABLE WATCHDOG TIMER TRAP

REGISTERS: All relevant information in in-core error log buffers.

REMARKS: Sigma 9 and Xerox 560 systems will attempt recovery from watchdog timer traps from I/O instructions without SCREECHing.

* * *

SCREECH CODE: #26 CALLED FROM: CSEHAND

MESSAGE: CSE TRAP DURING MFI,PFI HANDLING

REGISTERS: All relevant information is in in-core error log buffers.

REMARKS: On Sigma 9 during MFI handling, or on Xerox 560 during MFI or PFI handling, a CSE trap (X'46', X'4C', X'4D') occurred.

* * *

SCREECH CODE: #27 CALLED FROM: CSEHAND

MESSAGE: PROCESSOR FAULT INTERRUPT

REGISTERS: All relevant information is in in-core error log buffers.

REMARKS: For Xerox 560 systems only, a processor fault interrupt occurred for which continued operation is unlikely.

* * *

SCREECH CODE: #28 CALLED FROM: CSEHAND

MESSAGE: MEMORY PARITY ERROR - MEMORY ALTERED

REGISTERS: All relevant information is in in-core error log buffers.

REMARKS: Memory parity error correction caused memory to be altered, continuation without recovery not possible. Caused by I56 (Sigma 6/7) or T4C (Sigma 9, Xerox 560).

* * *

SCREECH CODE: #29-00 CALLED FROM: CSEHAND

MESSAGE: TRAP 4C - BUS CHECK FAULT

REGISTERS: All relevant information is in in-core error log buffers.

REMARKS: Sigma 9 bus check fault or Xerox 560 miscellaneous X'4C' trap, while in master mode.

* * *

SCREECH CODE: #29-01 CALLED FROM: CSEHAND

MESSAGE: TRAP 4C - MAP PARITY ERROR

REGISTERS: All relevant information is in in-core error log buffers.

REMARKS: Map register parity error on Sigma 9 or Xerox 560, while in master mode.

* * *

SCREECH CODE: #29-02 CALLED FROM: CSEHAND

MESSAGE: TRAP 4C - REGISTER BLOCK PARITY ERROR

REGISTERS: All relevant information is in in-core error log buffers.

REMARKS: Register block parity error or Xerox 560 while in master mode.

* * *

SCREECH CODE: #29-03 CALLED FROM: CSEHAND

MESSAGE: TRAP 4C - WRITE LOCK REGISTER PARITY ERROR

REGISTERS: All relevant information is in in-core error log buffers.

REMARKS: Write lock register parity error on Xerox 560 while in master mode.

* * *

SCREECH CODE: #2C-00 CALLED FROM: ADD

MESSAGE: BATCH SCHEDULING ERROR - MBS/CCI ERROR

REGISTERS:
R1 = (S:CUN) current user #
R2 = Device type
R3 = Context block address
R5 = 0
R6 = User's DCB addr (M:C)
SR2 = OPNLD + .14
SR3 = Context block address
SR4 = OPNLD + .40
D1 = BA(OPNLD+ .1E7)+.28
D2 = BA(CONTXT BLK+SCFQARGS)+.28
D3 = Device type mnemonic text

REMARKS: Register contents significantly different from above indicate the monitor wandered into GETI in ADD. Otherwise, a batch user has been created and has read a card before MBS selected him to be run. Actually all recorded 2C's have been CCI attempting to start a second job. Problem is either CCI read past FIN or MBS/GETI communication (e.g., GIB:UN clobbered).

* * *

SCREECH CODE: #2D-00 CALLED FROM: COOP

MESSAGE: COOPERATIVE BUFFER MANAGEMENT ERROR

REGISTERS:
R1 = BUFLIMS index for S/C 19
R2 = .BC11
R3 = Context block
SR4 = COOP+ .18D
D3 = 0

REMARKS: At context block initialization a buffer was allocated for the context block. This buffer has been lost through core clobbering or mismanagement of a buffer chain. The particular user cannot continue.

* * *

SCREECH CODE: #2D-01 CALLED FROM: COOP

MESSAGE: SYMBIONT/COOP FILE DEVICE INACCESSIBLE

REGISTERS:
R0 = COOP+ .19B
R1 = Context block physical address
R4 = (DCT3(DCTX)).XX1X XXXX
SR4 = COOP+ .15C
D1 = .XXFF0300+DCTX
D2 = BA(COOP BUFFER)
D3 = .400
D4 = Disk address

REMARKS: The symbiont/coop file device containing this user's file is down. If there are many file devices for symbiont/coop only, this user should be aborted. If only one symbiont/coop file device exists, it is pointless to run the system with that device down.

* * *

SCREECH CODE: #2D-02 CALLED FROM: COOP

MESSAGE: USERS COOP CONTEXT BLOCK CHAIN LOST

REGISTERS:
R1 = BUFLIMS index for S/C 19
R2 = .BC10
SR2 = OPNLD+ .137
SR4 = OPNLD+ .139
D3 = 0

REMARKS: Similar to 2D-00 but detected at context block open time. Particularly alarming because this check immediately follows the code which allocates context blocks.

* * *

SCREECH CODE: #2D-03 CALLED FROM: SACT

MESSAGE: COOP CONTEXT BLOCK POINTERS CLOBBERED

REGISTER:
R3 = 0
R6 = User DCB addr
SR1 = FCN,DCB (8,24)
SR4 = Exit from COOP

REMARKS: Either J:USCDX or context block 0 (special pointers) were clobbered.

* * *

SCREECH CODE: #2D-04 CALLED FROM: SUPCLS

MESSAGE: COOP DATA BUFFERS MISALLOCATED

REGISTERS:
D3 = Buffer being released, including spare buffer index in byte 0.
R5 = Context block 0 address and DBPOOL which is the address of the free context buffer list.
R2 = SV:LSIZ
SR4 = Return address of caller of RCBUFF.

REMARKS: An attempt was made to release a COOP data buffer when the free data buffer pool was full. Either the free data buffer pool has been clobbered or too many buffers have been allocated meaning some other COOP data area has been clobbered.

* * *

SCREECH CODE: #2E CALLED FROM: RDF

MESSAGE: POOL BUFFERS LOST - NONE ALLOCATED CURRENTLY

REGISTERS:
SR3 = DCB for which buffer is needed.
D4 = X'2E'

REMARKS: An attempt was made to get an IPOOL or FPOOL buffer, but none were in free pool and no open DCB had any. Probably either DCB chain has been clobbered or one or more DCBs have been clobbered.

* * *

SCREECH CODE: #2E-01 CALLED FROM: RA

MESSAGE: INCONSISTENCY IN READ-AHEAD TABLES

REGISTERS:
R12 = Disc address

REMARKS: An attempt was made to add an AIR block to the tables when it was already there.

* * *

SCREECH CODE: #30 CALLED FROM: PFSR

MESSAGE: UNBALANCED POWER ON/POWER OFF INTERRUPT PAIRS

REGISTERS: Indeterminate

REMARKS: Unbalanced power on/power off interrupt pairs, more of one than another (usually power on, or else system would hang in wait, B $-1).

* * *

SCREECH CODE: #31 CALLED FROM: IORT

MESSAGE: INVALID RESOURCE TYPE

REGISTERS: SR4 = ADDRESS+1 where discovered.

REMARKS: Invalid resource type found.

* * *

SCREECH CODE: #32-00 CALLED FROM: IOQ

MESSAGE: DCB DOESN'T CONTAIN A VALID DCT INDEX

REGISTERS: R2 = Address of DCB

REMARKS: DCT index not present in DCB.

* * *

SCREECH CODE: #34-00 CALLED FROM: TPQ1

MESSAGE: TRANSACTION PROCESSING FAILURE

REGISTERS:

REMARKS: The system queue manager for TP has discovered an unrecoverable state while processing transactions.

* * *

SCREECH CODE: #41-01 CALLED FROM: RTROOT

MESSAGE: FAILED TO FIND USER'S STATE (M:INTSTAT)

REGISTERS: R2 = Address of ICB being checked.

REMARKS: Probably results from a state having been added to SCHED without updating the four masks used by the M:INTSTAT routine (WAIT:MASK, EXU:MASK, IOWAIT:MASK, BLCKD:MASK).

* * *

SCREECH CODE: #41-10 CALLED FROM:

MESSAGE: BAD IOEX CALL TO NEWQ

REGISTERS: Set for BAL,R11 NEWQNW

REMARKS: NEWQNW returned to BAL+1.

* * *

SCREECH CODE: #41-11 CALLED FROM: RTNR

MESSAGE: UNABLE TO RETURN PRE-EMPTED DEVICE

REGISTERS:

REMARKS: RTNR's call to RMAOV was invalid.

* * *

SCREECH CODE: #43-01 CALLED FROM: CLOCK4

MESSAGE: NO ICBs CHAINED INTO RTICBCLKHDR

REGISTERS: Not relevant

REMARKS: Would probably be caused by overwriting lowcore.

* * *

SCREECH CODE: #43-02 CALLED FROM: CLOCK4

MESSAGE: ICBCLK FIELD OF ICB NEGATIVE

REGISTERS:
R2 = Address of bad ICB
R10 = Current timer increment

REMARKS: The ICBCLK field of an ICB should never go negative.

* * *

SCREECH CODE: #43-03 CALLED FROM: RTNR, CLOCK4

MESSAGE: NO BACK-LINK FOUND IN DE-CHAINED ICB

REGISTERS:
R2 = Current ICB (the one being de-chained)
R4 = Forward link (next ICB in chain)

REMARKS: A back-link of zero implies that the current ICB is SYSICB1 (the 1-second CLOCK3 ICB). This ICB should never be de-chained (i.e., de-activated).

* * *

SCREECH CODE: #46-21 CALLED FROM: RDF

MESSAGE: PRIVATE VOLUME LOGIC INCONSISTENCY

REGISTERS: SR4 = Address where error was detected.

REMARKS: Numerous modules call PVERR.

* * *

SCREECH CODE: #49 CALLED FROM: TYPR

MESSAGE: RESOURCE PREALLOCATION INCONSISTENT WITH REQUESTS

REGISTERS:
R2 = 0
R3 = Reel number
D4 = X'49'

REMARKS: Due usually to MBS failure to properly set/reset resource flags or resource (tape or private volume) not properly or fully released back to system.

* * *

SCREECH CODE: #56 CALLED FROM: MOCIOP

MESSAGE: UNABLE TO RELEASE PHYSICAL WORK PAGE

REGISTERS: Registers at time of trap.

REMARKS: Originates in MOCIOP module when unable to release a physical work page locked in core during Transaction Processing I/O on a Message Oriented Controller (e.g., 7605).

* * *

SCREECH CODE: #61 - (TRAP Cell) CALLED FROM: INITRCVR

MESSAGE: TEL OR CCI HAS TRAPPED

REGISTERS: Registers at time of trap.

REMARKS: Trap occurred while operating mapped, slave, and with TEL-in-control set. Subcode is trap location.

* * *

SCREECH CODE: #62 CALLED FROM: SCHED

MESSAGE: USER PROGRAM TOO LARGE FOR PHYSICAL MEMORY

REGISTERS:
R0 = Pages freed
R4 = Inswap user (S:ISUN)

REMARKS: R0 greater than SL:CORE, user got swapped out but now can't fit back in; pages may be released but not reported; JIT-in-core flag = 0 (UH:FLG X'200').

* * *

SCREECH CODE: #63 CALLED FROM: DPSIO

MESSAGE: INSUFFICIENT INFORMATION AVAILABLE TO SWAP THIS USER

REGISTERS:
R2 = IOCD
R6 = Command list address
R7 = Function code
D4 = X'63'

REMARKS: Insufficient data to compute function, follow-on function code invalid, or flags not set properly; disk pack-only swappers.

* * *

SCREECH CODE: #6A CALLED FROM: MM

MESSAGE: ATTEMPT TO RELEASE VIA M:CVM FROM USER W/O PROPER PRIVILEGE.

REGISTERS:
R1 = X'80'
R5 = Address of top of dynamic data or bottom of command
R6 = # pages to release
R7 = Virtual page number
SR1 = # pages released
SR2 = First page to release
SR3 = Incr. or decr. to next page
SR4 = Link
D1 = CC
D2 = CC mask

REMARKS: Virtual page outside of user's area (BUP-EUP) was obtained by an M:CVM CAL but the user lacks required privilege (X'80') to release it.

* * *

SCREECH CODE: #6B CALLED FROM: MM

MESSAGES: ERROR IN SPARE BUFFER TABLES

REGISTERS:
R11 = Address in buffer subroutine within MM (T:GBUF, T:RBUF, etc.) which detected the error.

REMARKS: Usually bad input from the calling routine.

* * *

SCREECH CODE: #6B CALLED FROM: SWAPPER

MESSAGE: ERROR IN SPARE BUFFER TABLES

REGISTERS:
R6 = BA (window page)
R14 = Physical page assigned to window

REMARKS: Page mapped into window is not contained in the spare buffer pool.

* * *

SCREECH CODE: #79 CALLED FROM: ENTRY

MESSAGE: MONITOR COMMITTED A STACK TRAP

REGISTERS: Registers at time of trap.

REMARKS: Master bit on in PSD, overflow, underflow, or pointer to stack lost.

* * *

SCREECH CODE: #79-01 CALLED FROM T:OV

MESSAGE: MONITOR STACK TRAP

REGISTERS: Registers at time of trap

REMARKS: OSTACK overflow

* * *

SCREECH CODE: #7C CALLED FROM: ALTCP

MESSAGE: ALTCP CALLED TO SERVICE A CAL THAT DOESN'T BELONG TO ALTCP

REGISTERS:
R3 = R field of CAL
R6 = First word PLIST
R7 = Addr. PLIST+1
SR1 = Code
SR4 = Exit addr (usually TRAPEXIT)

REMARKS: A CAL1,1 or CAL1,2 was passed to ALTCP but should have been handled by CALPROC.

* * *

SCREECH CODE: #7E-(TRAP Cell) CALLED FROM: INITRCVR

MESSAGE: MONITOR HAS TRAPPED

REGISTERS: Registers at time of trap.

REMARKS: Subcode is trap location. For traps that occur at locations less than X'8000' (JOVVPA), the 15 cells preceding the trap location and the trap location are stored in the monitor JIT AT X'8DF0'-X'8DFF'.

* * *

SCREECH CODE: #87 CALLED FROM: ALLYCAT

MESSAGE: ALLOCATION BUFFERS CONTAIN INVALID WORD COUNT

REGISTERS:
R2 = Stack count
R1 = Stack number

REMARKS: Either low core has been clobbered or some one has changed ALLYCAT's in-core data.

* * *

SCREECH CODE: #88 CALLED FROM: SCHED

MESSAGE: ALLYCAT CLOBBERED ONE OF THE ALLOCATION BUFFERS

REGISTERS:
R1 = Stack index
R3 = Stack count

REMARKS: ALLOCAT end-action has discovered discrepancy in granule/cylinder stacks.

* * *

SCREECH CODE: #89-00 CALLED BY: ALLYCAT

MESSAGE: ALLYCAT'S HGP CHAIN CLOBBERED

REGISTERS:
R7 = Invalid HGP chain address
R9 = ALLOCAT internal link register

REMARKS: ALLOCAT data (HGPs and TABLES) has been destroyed.

* * *

SCREECH CODE: #93 CALLED FROM: DPSIO, TSIO

MESSAGE: TDV COMMAND ADDRESS DOESNT POINT TO COMMAND LIST

REGISTERS:
R1 = 0
SR1 = Command list address from TDV
SR2 = TDV status
D2 = Cmd. list pointer (S:BECL,R1)

REMARKS: IOP/memory failure; extraneous entry to TSIO/DPSIO not generated within CLIST.

* * *

SCREECH CODE: #94 CALLED FROM: DPSIO, TSIO

MESSAGE: COMMAND LIST CLOBBERED DURING WRITE CHECK

REGISTERS:
SR1 = Incorrect command list entry address
SR2 = TDV status
R12 = Order code from incorrect command list entry

REMARKS: Can't find seek or TIC within next five command list entries following error entry, on write or write check.

* * *

SCREECH CODE: #95 CALLED FROM: DPSIO, TSIO

MESSAGE: UNRECOVERABLE I/O ERROR READING USER'S JIT

REGISTERS:
R1 = Inswap user number (S:ISUN)
R7 = DCT Index
SR1 = Command list address from TDV status
SR2 = TDV status

* * *

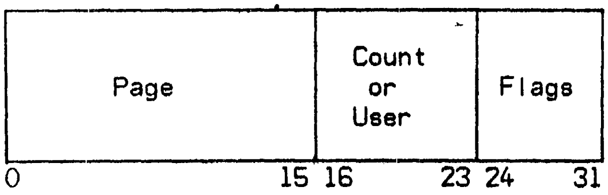
SCREECH CODE: #96 CALLED FROM: DSPIO, TSIO

MESSAGE: UNRECOVERABLE I/O ERROR READING SHARED PROCESSOR

REGISTERS:
R1 = Inswap user number (S:ISUN)
R7 = DCT index
SR1 = Command list address from TDV status
SR2 = TDV status

* * *

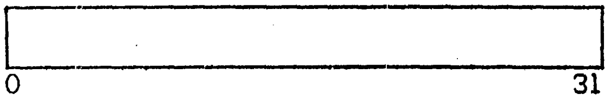
PWPTABLE - Physical Work Pages for TP



Length = PWPEND - PWPTABLE+1

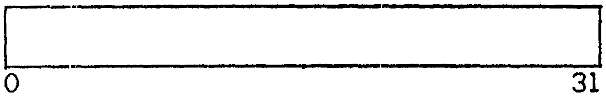
Bit	
0-15	Physical page number (left-justified)
16-23	If bit 26 is 0 - assign count; if bit 26 is 1 - user number
24	Reserved
25	Reserved
26	Release is being accomplished by clock processing due to unavailability of work space
27	Virtual window is being established in user in order to assign a page
28	Reserved
29	Page has been acquired by calling GPWP (Get Physical Work Page)
30	Page has been acquired by APWP (Assign Physical Work Page)
31	0 - this word in PWPTABLE is free for use 1 - this word in PWPTABLE is currently in use

SL:PWP



Bit	
0-31	Current limit of physical work pages

S:PWP#



Bit	
0-31	Current count of physical work pages

TTP RESIDENT TABLE - QUEUE MANAGEMENT ELEMENTS

0 78 1516 2324 31

<table>
<tr><td></td><td colspan="3">Q:MAP</td><td>0</td></tr>
<tr><td>Q:CC</td><td colspan="3">Q:CFU</td><td>1</td></tr>
<tr><td>Q:NSN</td><td colspan="3">Q:SN</td><td>2</td></tr>
<tr><td>Q:MAX</td><td colspan="3">Q:INXCONTROL</td><td>3</td></tr>
<tr><td colspan="4" rowspan="2">Q:DEF</td><td>4</td></tr>
<tr><td>5</td></tr>
<tr><td colspan="4" rowspan="2">Q:GET</td><td>6</td></tr>
<tr><td>7</td></tr>
<tr><td colspan="2">Q:LID</td><td>Q:MIN</td><td>Q:INXNAV</td><td>8</td></tr>
<tr><td></td><td colspan="3">Q:QHEAD</td><td>9</td></tr>
<tr><td></td><td colspan="3">Q:QTAIL</td><td>10</td></tr>
<tr><td colspan="4">QINX</td><td>11</td></tr>
<tr><td colspan="4">Q:INXTAIL</td><td>12</td></tr>
<tr><td></td><td colspan="3">Q:MPOOL</td><td>13</td></tr>
<tr><td colspan="4">Q:TID</td><td>14</td></tr>
<tr><td>Q:USR</td><td>Q:SAT</td><td>Q:PAGES</td><td>Q:OWN</td><td>15</td></tr>
<tr><td colspan="4">Q:TPPP</td><td>16</td></tr>
<tr><td colspan="4">Q:TPPTAIL</td><td>17</td></tr>
<tr><td colspan="4">Q:CONT</td><td>18</td></tr>
<tr><td colspan="4">Q:CONTTAIL</td><td>19</td></tr>
<tr><td colspan="4">Q:DATA</td><td>20</td></tr>
<tr><td colspan="4">Q:DATATAIL</td><td>21</td></tr>
<tr><td colspan="4">Q:PAUSE</td><td>22</td></tr>
</table>

TTP Table Item Descriptions

Item Name	(Bit/Byte) Word	Type	Description
Q:BACK	0 (6)	Bit	Backup option requested during unlock, 1 = yes
Q:CC	1 (0)	Byte	I/O completion code; set by end-action
Q:CFU	1	Address	Real address of the queue CFU table
Q:CONT Q:CONTTAIL	18 - 19	Word	Chain header, queue control blocks
Q:DATA Q:DATATAIL	20 - 21	Word	Chain head/tail, queue data blocks
Q:DEF	4	Doubleword	Chain head/tail, inactive criteria control (points to U:QLIST)
Q:GET	6	Doubleword	Chain head/tail, active criteria control entries (points to U:QLIST)
Q:INXNAV	8 (3)	Byte	Maximum number of keys in an index block
Q:INX Q:INXTAIL	11 - 12	Word	Chain header, queue index blocks
Q:INXCONTROL	3	Address	Address of the index control block
Q:LID	8 (0)	Halfword	Identification number to be assigned to the next GET list; initialized during queue unlock processing
Q:LOCK	0 (2)	Bit	Queue status, 1 = unlocked

Item Name	(Bit/Byte) Word	Type	Description
Q:MAP	0	Address	Address of the Queue Allocation Map
Q:MAX	3 (0)	Byte	Maximum number of queue index blocks retained in core
Q:MIN	8 (2)	Byte	Minimum number of pages required for queue blocks
Q:MPOOL	13	Address	Monitor Buffer Address (MPOOL)
Q:NSN	2 (0)	Byte	Number of volumes if queue is on private storage
Q:OWN	15 (3)	Byte	User number, queue owner
Q:PAUSE	22	Bit	Queue in lock, pause mode
Q:PAGES	15 (2)	Byte	Current number of physical pages allocated for queue usage
Q:QHEAD	9	Address	Word address, head of chain of users queued for access or space (U:QUES)
Q:QTAIL	10	Address	Word address, tail of chain of users queued for access or space (U:QUES)
Q:RCV	0 (7)	Bit	Recovery in progress, 1 = yes
Q:SAT	15 (1)	Byte	Queue saturation percentage to accept high priority PUTs only
Q:SN	2	Address	Address of the queue serial numbers for private storage
Q:TID	14	Word	Highest TID
Q:TPPP Q:TPPPTAIL	16 - 17	Word	Chain header, queue physical pages
Q:USR	15 (0)	Byte	User number, current queue I/O operation

RECOVERY TABLES

Core Dump Format

CORE is converted to a page number and the space required is compared with the RAD space available (RCVRDSZ). If space needed exceeds space available, TAPDMP is called. Otherwise, RCVRAD+2 is entered into the RECOVERY buffer.

CORE is dumped with an SIO and command chain in RCVDMP.

Following system reboot and core initialization, but before swapping RAD initialization, the dump space on RAD is written as a keyed file on the file RAD.

The keys for core pages are `| 03 | 00 | 00 | Page# |`

The user JITs are added to the keyed file with keys of `| 03 | user# | 00 | 00 |`.

Dump Tape Format

The dump tape is a labeled tape and each logical record is one page of core memory.

The label sentinel is :LBL RCVT.

The identification sentinel is :ACN :SYSbbbb Date of the crash in format from DATE and DATE+1.

Tape mark

The beginning of the file sentinel is:

```
:BOF
01    00    02    02
7     T     A     P          file name is TAPDUMP
D     U     M     P
09    01    00    02         ORG is conseq., VOL is 1
01    00    00    00
00    00    00    00
```

Tape mark

Control record

Each data record is 512 words.

The tape mark record is:

```
                        tape mark
                        :EOF
                        3*4         previous block size
   tape mark
                  end of reel sentinel terminates the dump
                           tape mark
                           :EOR
                           NULL
                  tape mark
                  tape mark
```

Recovery Buffer

Each item of information within the Recovery buffer is followed by an identification word. The identification consists of an id code in byte 0 and the word count of the information item in bytes 2 and 3. The items are not necessarily in the buffer in id order.

ID CODE	PROBABLE COUNT	ITEM
01	64+5+2	SGRAN, BGRAN, CURGRAN, FGRAN1, CURBUF and contents of CURBUF and 2 preceding words
02	16	Administrative message (COCMESS)
03	3	Initial and final RCVRAD from RCVDMP and RCVRCNT
04	size	Size of RECOVERY buffer
05	1	Dump tape identification
06	value	Down device number
07	value	Number of locked symbiont devices + 1, RCVRCNT, and SGB
08	logged on users	User #, swap index, seek address (8, 8, 16)
09	149	Partition limits, system limits
0A		Unused
0B	0	HGP reconstruction required
0C		Unused
0D	1	First disc address of granule stack for release
0E		Unused
0F	4	Write symbiont ghost recovery files
10	61 (SGCBUFSZ)	Symbiont ghost communication buffer
11	1	Symbiont ghost error word

BUFFER LAYOUT:

```
. . . . |DATE |DATE+1|TIME |02   3 |04   SIZE|
```

DESCRIPTION:

The Recovery buffer is a RES of 1024 words in the data area of RECOVERY. When RECOVERY processing is completed, the size of the Recovery buffer and its contents are moved to the first 2 granules of RCVRAD on the system RAD. Since the JITs are used by the second phase, any JITs occupying the area of the system swapping RAD where GHOST1 will be swapped must be moved to some other space on the RAD. MVEBUF accomplishes this by investigating the JIT addresses saved in the Recovery buffer and calling RDRAD1 and WRRAD1.

The data (00 protection) and AJIT pages of the symbiont ghost job (RBBAT) also must not be in the area of the system swapping RAD where GHOST1 will be swapped. These pages are moved at the same time as user JITs.

Power Fail-Safe Interrupt Status Tables

PFSRARM - Arm flip-flop status
PFSRTRIG - Trigger flip-flop status
PFSRENAB - Enable flip-flop status

These are halfword tables with a halfword for each interrupt group, including the non-existent group one, SYSGENed in the system.

INDEX

*RB:MFAD, VR-2
*RB:SPMF, VR-2
*RBH:MFX, VR-2
:ACCTLG, VN.02-1
:AMHED, VA.01-1
:LOGSZ, VA.01-1
:PROCS, VN.05-1
:RATES, VN.03-1
:RBLOG, VN.04-1
:UBML, VN.01-3
:UBMR, VN.01-3
:UGML, VN.01-3
:UGMR, VN.01-3
:UNML, VN.01-3
:UNMP, VN.01-3
:UNMR, VN.01-3
:UOML, VN.01-3
:UOMR, VN.01-3
:UPFLGS, VN.01-3
:USER, VN.01-1
2780, VR-4, -7
3COUNT, VK.01-1

A

ABO, VA-4
ACCN, VA-3
ACCO, VI.01-4
ACCOUNT DIRECTORY, VH.02-1
ACCOUNTING RECORD, VN.02-1
ACNCFU, VH.08-2
AIR, VH.21-2
ALLOCATION, VH.07-1
ALOCCT, VA-5
ANS, VH.16.01-2
ANS DCB, VH.11-1
ANSFLGS, VG.04-4
ARSZ, VG.05-1
ASSIGN/MERGE, VA-11,VL-1
ATITLE, VA-6
AVR, VG.04-1
AVRFLGS, VG.04-5
AVRFNMT, VG.04-4
AVRID, VG.04-1
AVRNOU, VG.04-1
AVRTBL, VG.04-1, -2
AVRTBLNE, VG.04-1
AVRTBLSIZ, VG.04-1

B

B4W:RES, VI.02-2
BAABC, VA-4
BARNST, VA-4
BATAPE, VG.04-1, -2
BB:...VI.02-2
BCP, VF.01-1
BD:ACCT, VI.02-2
BGRAN, VK-2
BH:..., VI.02-1, -2
BPRIO, VP-13
BREF, VO-9
BSCIO, VR-7
BUFCNT, VG.05-1
BUFFER, VF.01-3
BUP, VF.01-1
BW:FORM, VI.02-2
BW:SDA, VI.02-2

C

C1, VG.03-6
CBPOOL, VI.01-6
CCBEF, VA-4
CCLFLAGS, VA-5
CCLTFLAGS, VA-5
CDPO, VA-5
CEXT, VA-3
CFU, VH.08-1
CFU, USER, VH.08-7
CIC, VA-4
CIT, VG.01-1,VG.02-2
CLL, VF.01-1
CO:...,VG.05-3, -4
COB:SIOS, VG.05-3
COC, VG.05
COCBUF, VG.05-4
COCHPB, VG.05-4
COCII, VG.05-1
COCIR, VG.05-1, -4
COCO1, VG.05-1
COCOC, VG.05-1
COCOR, VG.05-1, -4
COCTERM, VG.05-1
COD:HWL, VG.05-2
COD:LPC, VG.05-2
COH:..., VG.05-3
CONSECUTIVE, VH.18-1
CONTEXT BLOCK, VI.01-7A
CONTEXT BLOCK DISPLACEMENT SYMBOLS, VI.02-12
COOP CONTEXT, VI.01-9
COOPERATIVE, VI.01-1
COPPGS, VI.01-6
CPE, VA-4
CPI, VG.05-1
CPO, VA-4
CPOS, VG.05-1
CPPO, VA-4

CRESDF, VP-11
CSE STOP, VK.01-1
CSED$, VK.01-1
CUL, VF.01-1
CUPO, VA-5
CURBUF, VK-2
CURGRAN, VK-2

D

DBMAX, VI.01-6
DBPOOL, VI.01-6
DCACCESS, VA-9
DCBLINK, VA-6
DCBTAB, VO-1
DCT, VG.01-1,VG.03-4
DCW, VG.01-1
DDLL, VF.01-1
DDUL, VF.01-1
DEQ, VQ-1
DEVICE DCB, VH.12-1
DISK ADDRESS, VH.17-1
DLL, VF.01-1
DO, VA-5
DO LIST, VP-7
DOT, VG.02-2
DPACCESS, VA-9
DSCIO, VR-7
DTT, VG.03-1
DUL, VF.01-1
DUMP, VU-1
DYNRESDF, VP-11

E

E:..., VC-2
EAPL, VG.05-2
EAPLLC, VG.05-2
EAPLOC, VG.05-2
ECB, VA-5
ENQ, VQ-1
EOMTIME, VG.05-1
ERLFLAGS, VA-5
ERO, VA-4
ERRLFLGS, VA-5
ERROR CODE, VM-1
ERROR LOG, VK-1
ERROR LOG CODES, VK-20
ERROR LOG ENTRIES, VK-8
ERROR MESSAGES, VM-1
ERROR SUBCODE, VM-1
ESTD, VG.05-2
ESTDLC, VG.05-2
ESTDUC, VG.05-2
EUP, VF.01-1
EVENT TRANSITION, VC-1
EVENTS, VC-2
EXPRESSION STACK, VO-11

F

FGRAN1, VK-2
FILCFU, VH.08-5
FILE DCB, VH.09-1
FILE DESCRIPTORS, VH.03-3
FILE DIRECTORY, VH.03-1
FIT, VH.04-1
FSP, VH.03-4

G

GENERALIZED DISK ADD., VK-54
GFC, VI.02-17
GHOST, VE-3
GI TABLES, VI.02-9
GI:..., VI.02-9
GIB:..., VI.02-9
GIH:IIM, VI.02-9
GLOBAL, VO-13
GLOSSARY, VK-9
GPRIO, VP-13
GROUP CODE, VM-1

H

HANSWAPS, VA-4
HASPIO, VR-7
HEAD, VO-3
HGP, VH.07-1
HIGHER INDEX, VH.01-4

I

IA, VD-1
ICB, VP-1,-10
ICBSIZE, VP-10
INSYM CONTEXT, VI.01-8
INTENT, VA-5
INTERNAL SYMBOL, VO-13
INTLB1, VP-12
INTLB2, VP-12
INTLB3, VP-12
INTLBSIZ, VP-12
IOQ, VG.02-1
IOTIME, VA-3
IRBT, VR-4
IREL, VP-7

J

J:ABC, VA-4
J:ABUF, VA-6
J:ACCN, VA-3
J:AJ, VA-10
J:AJIT, VA-12
J:ALB, VA-5
J:AMR, VA-11
J:ASPIN, VA-5
J:ASSIGN, VA-4
J:BASE, VA-11
J:BUP, VA-8
J:CALCNT, VA-3
J:CASSIN, VA-5
J:CCBUF, VA-4
J:CCBUF, VA-8
J:CFLAGS, VA-7
J:CL, VA-12
J:CLE, VA-11
J:CLL, VA-9
J:CLM, VA-8
J:CLPA, VA-10
J:CLS, VA-9
J:CPPO, VA-4
J:CPROCS, VA-7
J:CTIME, VA-3
J:CUP, VA-9
J:DCBLINK, VA-6
J:DCBLINK, VO-1
J:DCBLL, VA-9
J:DCBUL, VA-9
J:DDLL, VA-8
J:DDUL, VA-8
J:DELTAT, VA-3, -7
J:DLL, VA-8
J:DUL, VA-8
J:DWSK, VA-8
J:EUP, VA-8
J:EXLY, VA-4
J:EXTENT, VA-7
J:FDDA, VA-11
J:ICBHDR, VA-11
J:IDELTAT, VA-7
J:INTENT, VA-5
J:INTER, VA-5
J:JAC, VA-11
J:JIP, VA-5
J:JIT, VA-3
J:LMN, VA-8
J:MRT, VA-3
J:NRS, VA-4
J:OVHTIM, VA-3
J:OVRLY, VA-7
J:PLL, VA-8
J:PTIME, VA-3
J:PUF, VA-7
J:PUL, VA-8
J:RNST, VA-4
J:RWECB, VA-5
J:STAR, VA-11
J:START, VA-8
J:T, VA-11
J:TCB, VA-6,VO-1
J:TELBUF, VA-7
J:TELFLGS, VA-5
J:TIC, VA-11
J:TIMENT, VA-5
J:TITLE, VA-6
J:TRAP, VA-4
J:TREE, VA-6,VO-1
J:UN, VA-3
J:UNAME, VA-3
J:USCDX, VA-6
J:USENT, VA-6
J:USER, VA-9
J:UTIME, VA-3
J:UTIMER, VA-5
J:VLCS, VA-10
J:XP, VA-6
JACB, VA-4
JACCN, VA-3
JAJ, VA-10
JASSIGN, VA-4
JB:BCP, VA-9
JB:CBUC, VA-9
JB:COR, VA-10
JB:FBUC, VA-9
JB:FBUL, VA-9
JB:FRS, VA-9
JB:LC, VA-7
JB:LMAP, VA-11
JB:LPP, VA-7
JB:MAX, VA-9
JB:MNPA, VA-9
JB:NFPOOL, VA-6
JB:NRG, VA-10
JB:ORG, VA-5
JB:PCC, VA-9
JB:PCD, VA-9
JB:PCDCB, VA-9
JB:PCDD, VA-9
JB:PCP, VA-9
JB:PCW, VA-7
JB:PECK, VA-10
JB:PMTS, VA-10
JB:PNR, VA-10
JB:PPC, VA-10
JB:PRIV, VA-6
JB:PROMPT, VA-7
JB:SLNK, VA-10
JB:STEP, VA-5
JB:STEPCC, VA-10
JB:TDP, VA-9
JB:TMTS, VA-10
JB:VLH, VA-10
JB:VLT, VA-10
JB:XLNK, VA-10
JBBCP, VA-9
JBCBLL, VA-9

JBCBUC, VA-9
JBFBFP, VA-9
JBFBUC, VA-9
JBLMAP, VA-11
JBMNPA, VA-9
JBNASP, VA-9
JBNFPOOL, VA-6
JBNRG, VA-10
JBPCC, VA-9
JBPCDD, VA-9
JBPCP, VA-9
JBPPC, VA-10
JBTDP, VA-9
JBUP, VA-8
JBVLH, VA-10
JCL, VA-12
JCLE, VA-11
JCLP, VA-11
JCLPA, VA-10
JCLT, VA-11
JCMAP, VA-12
JCPPO, VA-4
JDA, VA-12
JDCBLL, VA-9
JDCBUL, VA-9
JDDLL, VA-8
JDDUL, VA-8
JDLL, VA-8
JDUL, VA-8
JEUP, VA-8
JH:DA, VA-12
JH:LDCF, VA-10
JH:PC, VA-6
JIT, VA-1, -3
JITREE, VA-6
JJAC, VA-11
JJITVP, VF.01-1
JLMAP, VA-11
JOPT, VA-8
JPCP, VA-9
JPLL, VA-8
JPPC, VA-10
JPPH, VA-10
JPPT, VA-10
JPUF, VA-7
JPUL, VA-8
JRBID, VA-5
JRESOPT, VA-8
JRNST, VA-4
JSPBFLG, VA-9
JSTDOPT, VA-8
JTELFLGS, VA-5
JUNAME, VA-3
JVLCS, VA-10
JVLH, VA-10
JVLT, VA-10
JX:, VA-1
JX:CMAP, VA-12
JX:PPH, VA-10
JX:PPT, VA-10
JXCMAP, VA-12
JXPPH, VA-10
JXPPT, VA-10

L

L1, VG.03-6
LABELED TAPE DCB, VH.10-1
LABELED TAPE RECORD FORMAT, VH.16.01-4
LABELED TAPE SENTINELS, VH.16.01-1
LASTICB, VP-10
LB:UN, VG.05-1
LDEV, VG.03-4
LDEVCNT, VI.01-6
LDVCONST, VG.03-2
LDVEX, VI.01-4
LIBRARY, VO-7
LINK, VO-2
LNOL, VG.05-1
LOAD, VO-2
LOCCT, VO-15
LP, VA-5
LPART, VI.02-7
LSERIAL, VI.02-5

M

M:FRGD, VP-10
M:IMC, VP-12
M:SGP, VF.02-2
M:SPROCS, VE-1
M:UC, VA-6
M:XX, VA-8
MASTER INDEX, VH.01-1, -4
MAXG, VE-3
MAXOVLY, VE-2
MBS, VI.02-1, -4
MDPO, VA-5
MDYNRESDF, VP-11
MEMORY, VF.01-1
MEMORY ALLOCATION, VF.01-2
MJCFLG, VA-5
MODE, VG.05-1
MPO, VA-4
MPOOL, VF.01-3
MPPO, VA-4
MRT, VA-3
MUPO, VA-5
MXFPL, VA-8
MXSTRM, VI.01-3

N

NFPOOL, VA-6
NSWAPS, VA-4

O

OB:..., VG.03-1
OB:BOPTX, VG.03-2, -4
OB:BTX, VG.03-2, -4
OB:GODTX, VG.03-4, -2
OB:GTX, VG.03-2, -4
OB:OOPTX, VG.03-2, -4
OB:OTX, VG.03-2, -4
OH:NM, VG.01-1, VG.03-2
OPRIO, VP-13
OV:NM, VG.03-4
OV:NMSZ, VG.03-2, -5
OV:SIZ, VG.03-2, -4
OVHTIME, VA-3

P

P:AC, VE-2
P:NAME, VE-1, -2
P:SA, VE-1, -2
P:TCB, VE-2
P1, VG.03-6
PACK, VG.04-3
PARTITION, VI.02-6
PAT, VI.02-11
PB:..., VE-1, -2
PBT:LOCK, VE-2
PER, VF.01-3
PFA, VF.01-3
PFSRARM, VU-3
PFSRENAB, VU-3
PFSRTRIG, VU-3
PH:DDA, VE-2
PH:FRQ, VE-2
PH:PDA, VE-2
PL:..., VI.02-6
PLB:..., VI.02-6
PLB:MAX, VI.02-7
PLB:MIN, VI.02-7
PLD:ACT, VI.02-6
PLH:..., VI.02-6
PLL, VF.01-1
PNAMEND, VE-2
POWER FAIL-SAFE, VU-3
PP:UPPC, VP-11
PP:UPPH, VP-11
PP:UPPT, VP-11
PPMD, VA-4
PPROCS, VE-2
PPTABDSK2, VP-11
PPTABLE, VP-11
PPTABLSZ, VP-11
PRDCRM, VA-10
PRDPRM, VA-10
PRIVATE, VH.06-1, VH.20-1
PRT, VA-4
PSA, VF.01-3
PUBLIC, VH.19-1
PUF, VA-4
PUL, VF.01-1
PWPENO, VT-1
PWPTABLE, VT-1
PX:..., VE-1
PX:HPP, VE-2
PX:TPP, VE-2

Q

Q:..., VT-3
QECB, VQ-4
QFREE, VG.02-1
QHEAD, VQ-1
QHHP, VQ-2
QHNAME, VQ-2
QHVP, VQ-2
QQF, VQ-2
QQHP, VQ-2
QQUN, VQ-2
QT, VQ-2
QUEA, VQ-4
QUEUE, VQ-2, VT-2
QUEUE HEAD, VG.01-1
QUHP, VQ-2
QUPP, VQ-4
QUVP, VQ-2

R

R:..., VC-1
RA:..., VH.21-2
RA:DA, VH.21-1
RA:SGET, VH.21-3
RAB:BLINK, VH.21-2
RAB:FLINK, VH.21-1
RAB:USER, VH.21-1
RAH:DCB, VH.21-1
RAH:TIME, VH.21-1
RASIZE, VH.21-1
RAT, VG.03-4, VI.02-4
RATE VN.03-1
RAX:PAGE, VH.21-1
RB:BUF, VR-1
RB:FLAG, VR-1, -3
RBB:BCB, VR-1
RBB:CPZ, VR-2
RBB:DSM, VR-2
RBB:HFE, VR-1
RBB:HIN, VR-1
RBB:HOU, VR-1
RBB:ID, VR-1
RBB:LPZ, VR-2
RBB:MXP, VR-2
RBB:SFC, VR-2

RBB:SMD, VR-2
RBB:SPC, VR-2
RBBAT, VI.01-5, VI.02-1
RBBAT COMBUF, VI.02-17
RBD:WSN, VR-1
RBH:ACK, VR-1
RBID, VN.04-1
RBLIMS, VR-1
RBSSS, VR-7
RCVDMP, VU-1
READ-AHEAD, VH.21-1
REAL TIME, VP-1
RECOVERY, VU-1
REF, VO-9
REF/DEF, VO-10
RELOCATION DICTIONARY, VO-12
REMOTE PROCESSING ERROR RECORD, VK-42
RESDF, VP-11
RESDFP, VP-11
RNST, VA-4
RP1, VK-42
RP2, VK-42
RP3, VK-42
RP4, VK-42
RSERIAL, VI.02-5
RSZ, VG.05-1
RT:GINTP, VP-10
RT:UINTP, VP-10
RTICBCLKHDR, VP-7
RTICBHDR, VP-10, -7
RUN STATUS, VA-4
RUNFLAG, VA-4

S

S:BADFLG, VC-4
S:BCL, VC-3
S:BDA, VC-3
S:CPINC, VC-4
S:CUT, VC-4
S:ECL, VC-3
S:GJOBACN, VE-3
S:GJOBTBL, VE-3
S:IOPINC, VC-4
S:IRPINC, VC-4
S:OPC, VC-1
S:PRIODEC, VC-4
S:PWP$^{\#}$, VT-1
S:RTIR, VC-4
S:SCL, VC-3
S:SET, VC-1
S:SYMDB, VI.02-11
S:SYMDG, VI.02-11
S:SYMDO, VI.02-11
S$^{\#}$B:TYP, VI.02-3
S$^{\#}$H:LNK, VI.02-3
S$^{\#}$W:SER, VI.02-3
SAPL, VG.05-2
SAPLLC, VG.05-2
SAPLUC, VG.05-2
SB:FPL, VC-3
SB:GJOBUN, VE-3
SB:HQ, VE-3, -4
SB:IOTA, VC-1
SB:LTY, VG.03-2, -5
SB:OSUL, VC-3
SB:PNL, VC-3
SB:RBDF, VG.03-3, VI.02-4
SB:RBMX, VG.03-3, VI.02-4
SB:RGDF, VI.02-4
SB:RGMF, VI.02-4
SB:RGMX, VG.03-3
SB:RODF, VG.03-3, VI.02-4
SB:ROMX, VG.03-3, VI.02-4
SB:RQ, VC-1, -4
SB:RTY, VG.03-3, VI.02-4
SB:SET, VC-1
SB:SWP, VC-1
SB:TQ, VC-3, -4
SCHEDULER QUEUES, VC-1
SCNTXT, VI.01-3
SCO-10, VC-1, -2
SCREECH, VS-1
SCREECH CODE, VS-1
SCU, VC-2
SERVICE LIMIT, VI.02-10
SGCHD, VI.02-18
SGRAN, VK-2
SH:RTOT, VI.02-4
SH:EDA, VC-3
SH:LNM, VG.03-2, -5, VI.01-6
SH:OPNM, VG.03-2, -4
SH:RBCU, VG.03-3, VI.02-4
SH:RBSUM, VG.03-3, VI.02-4
SH:RGCU, VI.02-4
SH:RGSUM, VG.03-3, VI.02-4
SH:RNM, VG.03-3, VI.02-4
SH:ROCU, VG.03-3, VI.02-4
SH:ROSUM, VG.03-3, VI.02-4
SH:RTOT, VG.03-3
SH:SDA, VC-3
SH:SYMT, VI.02-11
SIO, VG.01-1
SIOMF, VC-2
SIOW, VC-2
SL:AIRM, VH.21-3
SL:AIRTO, VH.21-3
SL:BDF, VI.01-10
SL:BMX, VI.01-10
SL:BPRIO, VP-13
SL:GDF, VI.02-10
SL:GMX, VI.02-10
SL:GPRIO, VP-13
SL:NAME, VI.02-10
SL:ODF, VI.02-10
SL:OMX, VI.02-10
SL:OPC, VC-1
SL:OPRIO, VP-13

SL:PWP, VT-1
SL:RAMF, VH.21-3
SL:RAMR, VH.21-3
SL:RATOR, VH.21-3
SMAXOUT, VC-3
SMUIS, VD-1
SNDDX, VI.01-3
SNDDXSIZ, VI.01-3
SNULL, VC-2
SOFTWARE CHECK, VS-1
SOLICIT, VG.04-1
SPSIZE, VE-2
SQ, VQ-2
SQA, VC-1, -2
SQFI, VC-1, -2
SQHD, VI.01-3
SQR, VC-1, -2
SQRO, VC-1, -2
SQTL, VI.01-3
SQUE, VI.01-3
SRET, VI.01-3
SRT, VC-1, -2
SS, VA-4
SSIG, VI.01-3
SSTAT, VI.01-3
SSTD, VG.05-2
SSTDLC, VG.05-2
SSTDUC, VG.05-2
STAR FILES, VA-11
STATES, VC-2
STB:DPD, VI.01-5
STB:LNK, VI.01-3
STB:Q, VI.01-3
STB:TYP, VI.01-3
STH:FLG, VI.01-3, -5
STH:NM, VI.01-5
STH:SUS, VI.01-3
STI, VC-1, -2
STIO, VC-2
STOB, VC-1, -2
STOBO, VC-2
STORAGE, VF.01-3
STW:FORM, VI.01-5
SUPER, VN.01-1
SV:FTYM, VI.02-11
SV:LIM, VI.02-10
SV:LSIZ, VG.03-5
SV:LSIZP, VG.03-5
SV:LSIZPA, VG.03-5
SV:RSIZ, VG.03-3
SV:RSIZ, VI.02-4
SV:TYM, VI.02-11
SW, VC-1, -2
SWAP STORAGE, VF.02-1
SWAPPER, VC-3
SWAPPER LAYOUT, VF.02-2
SYMBIONT, VI.01-1
SYMBOL, VO-13
SYMTAB, VI.01-3
SYMX, VI.01-3
SYNONYMOUS, VH.04-4
SYSID, VA-3

T

TAPE, VG.04-3
TB:FLGS, VG.03-2, -4, -5
TB:FLGS1, VG.03-2
TB:MAX, VG.03-2, -4
TB:SZ, VG.03-2, -4
TCB, VA-6, VO-1
TCBADR, VA-6
TDP, VF.01-1
TDV, VG.01-1
TIMENT, VA-5
TIMTMP, VA-3
TIO, VG.01-1
TL, VG.05-1, -4
TMDCRM, VA-10
TMDPRM, VA-10
TMPDCPK, VA-7
TMPDPPK, VA-6
TP, VT-1
TPACCESS, VA-5
TPEXT, VA-3
TPIOT, VA-3
TPOVT, VA-3
TRANSLATION, VG.05-2
TREE, VO-5
TREE TABLES, VO-1
TRPFLAGS, VA-6
TSERIAL, VI.02-5
TSTACK, VA-7
TTP, VT-2
TTYIN, VG.05-2
TTYOUT, VG.05-2
TUEXT, VA-3
TUIOT, VA-3
TUOVT, VA-3
TYC, VH.09-10, VH.12-7
TYPMNSZ, VG.03-4

U

U:..., VD-1
UB:..., VD-1
UB:APO, VA.01-2, VA-7
UB:APR, VA.01-2, VA-7
UB:ASP, VA.01-2, VA-7
UB:BL, VC-3

UB:DB, VA-7,VA.01-2
UB:FL, VC-3
UB:PRIO, VC-1
UB:PRIOB, VC-1
UH:...., VD-1
UH:DL, VD-2
UH:FLG, VA-7,VD-2
UH:FLGS, VD-2
UNAME, VA-3
USRENT, VA-6
UTIMER, VA-5
UTS, VA-7

V

VLP, VH.09-11
VTOC, VH.06-2

W

WORK-STATION, VA.01-2

X

XSL, VA-4

XEROX

Reader Comment Form

We would appreciate your comments and suggestions for improving this publication

Publication No.	Rev. Letter	Title	Current Date

How did you use this publication?

☐ Learning ☐ Installing ☐ Sales

☐ Reference ☐ Maintaining ☐ Operating

Is the material presented effectively?

☐ Fully Covered ☐ Well Illustrated ☐ Well organized ☐ Clear

What is your overall rating of this publication?

☐ Very Good ☐ Fair ☐ Very Poor

☐ Good ☐ Poor

What is your occupation?

Your other comments may be entered here. Please be specific and give page, column, and line number references where applicable. To report errors, please use the Xerox Software Improvement or Difficulty Report (1188) instead of this form.

Your name & Return Address

Thank You For Your Interest. (fold & fasten as shown on back, no postage needed if mailed in U S A.)

PLEASE FOLD AND TAPE–
NOTE: U. S. Postal Service will not deliver stapled forms

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 59153 LOS ANGELES,CA 90045

POSTAGE WILL BE PAID BY ADDRESSEE

HONEYWELL INFORMATION SYSTEMS
5250 W. CENTURY BOULEVARD
LOS ANGELES, CA 90045

ATTN: PROGRAMMING PUBLICATIONS

Honeywell

Honeywell Information Systems
In the U.S.A.: 200 Smith Street, MS 486, Waltham, Massachusetts 02154
In Canada: 2025 Sheppard Avenue East, Willowdale, Ontario M2J 1W5
In the U.K.: Great West Road, Brentford, Middlesex TW8 9DH
In Australia: 124 Walker Street, North Sydney, N.S.W. 2060
In Mexico: Avenida Nuevo Leon 250, Mexico 11, D.F.

29294, .75C980, Printed in U.S.A.