HP 3000 Computer Systems

HEWLETT
PACKARD

System Tables
reference manual

HP 3000 Computer Systems

MPE IV
SYSTEM TABLES

Reference Manual

HEWLETT
PACKARD
19447 PRUNERIDGE AVENUE, CUPERTINO, CA 95014

Part No. 32002-90003
E1182

Printed in U.S.A. 1182

LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the date of the current edition, and lists the dates of all pages of that edition and all updates. Within the manual, any page changed since the last edition is indicated by printing the date the changes were made on the bottom of the page. Changes are marked with a vertical bar in the margin. If an update is incorporated when an edition is reprinted, these bars and dates remain. No information is incorporated into a reprinting unless it appears as a prior update.

Third Edition..............November 1982

Effective Pages	Date
ALL	NOV 1982

PRINTING HISTORY

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The date on the title page and back cover of the manual changes only when a new edition is published. When an edition is reprinted, all the prior updates to the edition are incorporated. No information is incorporated into a reprinting unless it appears as a prior update. The edition does not change.

First Edition	JAN 1979
Second Edition	APR 1981
Update No. 1	OCT 1981
Third Edition	NOV 1982

PREFACE

The information included in this manual is provided by Hewlett Packard to describe the internal organization of MPE. It is not intended to be a guide to the modification of MPE.

Any modification of the tables presented in this manual by HP 3000 users is strongly discouraged as serious damage to the operating system may result. Furthermore, Hewlett-Packard will not support, correct, or attend to any resulting modification of the MPE Operating System Software.

It is not the intention of Hewlett-Packard to update this manual on any scheduled basis. Hewlett-Packard is not responsible for problems arising from inaccuracies existing in this manual, nor is it responsible for the correction of any inaccuracies.

MANUAL PLAN

INTRODUCTORY LEVEL:

GENERAL
INFORMATION
Manual
30000-90008

USING THE
HP 3000
03000-90121

USING
FILES
30000-90102

STANDARD USER LEVEL:

COMMANDS
Reference
Manual
30000-90009

ERROR MESSAGES
AND RECOVERY
Reference Manual
30000-90015

SYSTEM
UTILITIES
Manual
30000-90044

INDEX TO MPE
REFERENCE
DOCUMENTS
30000-90045

FILE SYSTEM
Reference
Manual
30000-90236

INTRINSICS
Reference
Manual
30000-90010

DEBUG/
STACK DUMP
Reference Manual
30000-90012

SEGMENTER
Reference
Manual
30000-90011

ADMINISTRATIVE LEVEL:

CONSOLE
OPERATOR'S
GUIDE
32002-90004

SYSTEM MANAGER/
SUPERVISOR
Reference Manual
30000-90014

SUMMARY LEVEL:

SOFTWARE
POCKET GUIDE
30000-90049

TABLE OF CONTENTS

CHAPTER 1 MEMORY LAYOUT

FIXED LOW MEMORY (SERIES II/III) 1-1
FIXED LOW MEMORY (SERIES 30/33) 1-2
FIXED LOW MEMORY (SERIES 44) 1-25
SYSTEM GLOBAL AREA 1-6
SYSGLOB EXTENSION 1-17
SYSDB WORDS 1-19
ALLOW MASK FORMAT 1-20
LOGGING RELATED LOCATIONS 1-21
FLAGX 1-21
PROCESS STOP LIST GENERAL LAYOUT 1-22
  Entry Format 1-22
  Preassigned Entries 1-22
INITIAL MEMORY ALLOCATION 1-23
  Initial Memory Layout 1-25

CHAPTER 2 MEMORY MANAGEMENT TABLES

SEGMENT TABLE STRUCTURE 2-1
  Pointers and DST #'s of Segment Table . . 2-2
    Components
  Standard Segment Identifier Format 2-3
  DST Entry Formats 2-3
    DST Entry 0 Format 2-3
    DST General Entry Format 2-3
  CST Entry Formats 2-5
    CST Entry 0 Format 2-5
    CST General Entry Format 2-5
  CST Entry Field Descriptions 2-6
  Table Format - CSTBLK 2-6
  Program Blocks and the CSTXMAP 2-7
    Entry Format - CST Extension Block . . . 2-7
  Fixed DST Entry Assignments 2-8
SWAP TABLES 2-11
  SWAPTAB 2-11
    SWAPTAB Entry 0 Format 2-11
    SWAPTAB Unassigned Entry Format 2-11
  Segment Locality Lists (SLL) 2-12
    SLL Header Format 2-13
    SLL Entry Format 2-14
    Special Request Table 2-15
MAIN MEMORY REGION HEADERS AND TRAILERS . . 2-16
    Available Region Headers and Trailers 2-17
      Available Region Global Header Format 2-17
      Available Region Subregion Header . . 2-17

Available Region Subregion Trailer . . 2-18
Available Region Global Trailer . . . 2-18
Reserved Region Headers and Trailers . . 2-19
Reserved Region Global Header Format 2-19
Reserved Region Subregion Header . . . 2-20
Reserved Region Subregion Trailer . . 2-20
Reserved Region Global Trailer 2-21
Assigned Region Headers and Trailers . . 2-22
Assigned Region Global Header Format 2-22
Assigned Region Subregion Header . . 2-22
Assigned Region Subregion Trailer . . 2-23
Region Header and Trailer Field Descriptions 2-24
SPACE ALLOCATION STRUCTURES 2-26
Available Region List (ARL) 2-26
Available Region Size Bit Map (ARSBM) 2-27

CHAPTER 3 DISC LAYOUT

SYSTEM DISC LAYOUT 3-1
DISC LABEL 3-4
System Volume 3-4
Serial Volume 3-6
Master Volume 3-8
Slave Volume 3-10
DEFECTIVE TRACKS TABLE 3-12
DEFECTIVE SECTOR TABLE 3-14
RESERVED AREA BIT MAP 3-15
DISC COLD LOAD INFORMATION TABLE 3-16
INITIAL PROGRAM CST MAP 3-20
VIRTUAL DISC SPACE MANAGEMENT STRUCTURES . . 3-22
Virtual Disc Space Management Table . . . 3-22
General Structure 3-22
VDMSTAB Entry 0 Format 3-23
VDMSTAB General Entry Format 3-24
VOLUME TABLE 3-25
Typical Private Volume Entry 3-26
Typical System Volume Entry 3-27

CHAPTER 4 DIRECTORY

OVERVIEW OF DIRECTORY 4-2
DIRECTORY DATA SEGMENT 4-3
DIRECTORY POINTER AREA 4-5
DIRECTORY SPACE DATA SEGMENT 4-6
DIRECTORY STRUCTURE 4-7
DIRECTORY DEFINITIONS 4-8

INDEX BLOCK PREFIX 4-8
INDEX ENTRY 4-9
ACCOUNT ENTRY 4-9
GROUP ENTRY 4-11
FILE ENTRY 4-14
USER ENTRY 4-14
USER ATTRIBUTES/CAPABILITY 4-15
VOLUME SET DEFINITION ENTRY 4-16
GVS LINKAGE, INFO, VOLFLAGS, VOLINFO 4-17
VOLUME SET CLASS ENTRY 4-18
GVC LINKAGE,INFO 4-19
VOLUME MASK FORMAT 4-19
VOLUME TABLE (SEE CHAPTER 3)

CHAPTER 5 LOCK RESOURCES

SIR ALLOCATION 5-1
MULTIPLE SIR ALLOCATION 5-2
SIR TABLE INFORMATION 5-3
SIR ENTRY FORMATS 5-4
RIN TABLE GENERAL LAYOUT 5-5
ALLOCATION AND LOCKING OF LOCAL RINS 5-6
ALLOCATION AND LOCKING OF FILE RINS 5-7
ALLOCATION AND LOCKING OF GLOBAL RINS . . . 5-8

CHAPTER 6 FILE SYSTEM

FILE SYSTEM INTRODUCTION 6-1
FILE SYSTEM OVERVIEW 6-1
  Buffers 6-2
TABLE FORMATS 6-3
  File System Section of PCBX (PXFILE) . . . 6-3
    Overhead 6-4
    PXFILE Control Block Table 6-6
    Available Block 6-7
    Active File Table (AFT) 6-8
  File Control Block Table (CBTAB) 6-10
    Overhead 6-11
    Vector Table 6-13
    Control Block Area 6-16
    Access Control Block (ACB) 6-17
    Logical Access Control Block (LACB) . . 6-18
    Physical Access Control Block (PACB) . . 6-20
    File Control Block 6-36
  File Label (FLAB) 6-42
  File Multi-Access Vector Table (FMAVT) . . 6-50
    Zero Entry Format 6-50

Typical Entry Format 6-51
System Global Area (SYSGLOB) 6-52
SIRs, Locks, and Deadlocks 6-53

CHAPTER 7 PROCESS TABLES

PROCESS TABLES INTRODUCTION 7-1
PROCESS CONTROL BLOCK (PCB) TABLE STRUCTURE AND FORMAT 7-1
Fixed Cells Related to PCB 7-1
PCB Entry 0 Format 7-2
Unassigned PCB Entry Format 7-3
Assigned PCB Entry Format 7-4
PCB Assigned Entry Field Descriptions . . 7-5
PROCESS CONTROL BLOCK EXTENSION (PCBX) STRUCTURE AND FORMAT 7-8
PCBX General Structue 7-8
PXGLOB Format 7-9
PXFIXED Assignments 7-10
PXFIXED Expansion Bitmap 7-14
File System Section of PCBX (PXFILE) 7-15
Overhead (PXFILE) 7-16
Control Block Table (PXFILE) 7-19
Available Block Area (PXFILE) 7-21
Available File Table (AFT) (PXFILE) . . 7-21
PCBX for Core Resident System Process Stacks 7-24
PROCESS TO PROCESS COMMUNICATION TABLE . . 7-25
SUB-SYSTEM RESERVED DL AREA 7-26
FORTRAN LOGICAL UNIT TABLE (FLMT) 7-27

CHAPTER 8 JOB TABLES

JMAT STRUCTURE (JOB MASTER TABLE) 8-1
JMAT ENTRY 8-3
JOB STATES 8-4
JPCNT - JOB PROCESS COUNT TABLE 8-5
GLOBAL RIN FLAG TABLE 8-5
JCUT - JOB CUTOFF TABLE 8-6
JIT - JOB INFORMATION TABLE 8-7
JDT - JOB DIRECTORY TABLE 8-10
JOB DATA SEGMENT DIRECTORY ENTRY 8-11
JOB TEMPORARY FILE ENTRY 8-11
FILE EQUATION FILE ENTRY 8-12
JOB LINE EQUATION ENTRY 8-13
JOB CONTROL WORD TABLE 8-14
AOPTIONS AND FOPTIONS WORD BREAKDOWN 8-15

PMASK WORD BREAKDOWN 8-16
UCOP REQUEST QUEUE (DST#9) 8-17
UCOP ENTRY FORMAT 8-18

CHAPTER 9 RELOCATABLE OBJECT CODE

USL FILES - GENERAL INFO 9-1
RECORD 0 AND OVERALL USL FILE FORMAT 9-1
  Data Descriptors, Passed Parameters . . . 9-5
  Entry Types 0,1 9-6
  Entry Types 2,4-Notes for SPL and FORTRAN 9-7
  Entry Type 2 9-8
  Entry Types 3,4 9-11
  Entry Type 5 9-14
  Entry Type 6 9-15
  Entry Type 7 9-17
  Entry Type 8 9-19
  Entry Header Format 9-21
  Header Types 0,1 9-22
  Header Types 2,3 9-23
  Header Types 4,5 9-24
  Header Types 6,7 9-25
  Header Type 8 9-27
  Header Type 9 9-28
  Header Types 10,11 9-30
RL FILE FORMAT 9-31
  Storage Management 9-32
  Entry Point Directory 9-33
  Typical Directory Entry 9-34
  Procedure Info Block 9-35
  Headers 9-36

CHAPTER 10 PREPARED OBJECT CODE

PROGRAM FILE FORMAT 10-1
  FLAGS 10-3
  FLAGS2 10-4
  CST Remapping Array 10-5
  Segment Descriptor Array 10-5
  Global Area Format 10-5
  External List 10-6
  Entry Point List 10-7
  The Code Segment with Patch 10-8
    Patch Area 10-8
    PMAP Info 10-9
    PMAP Type Table 10-9
    PMAP Records 10-10

Type 0 Segment PMAP Record 10-10
Type 1 Procedure PMAP Record 10-10
Type 2 Secondary Entry PMAP Record 10-11
SL FILE FORMAT 10-12
Storage Management 10-14
Entry Point Directory 10-14
Typical Directory Entry 10-15
Code Segment Linkage Structure 10-16
REFERENCE TABLE STRUCTURE 10-18
Reference Table Entry 10-19
The Code Segment With Patch Area 10-20
Patch Area 10-20
PMAP Info 10-21
PMAP Type Table 10-21
PMAP Records 10-22
Type 0 Segment PMAP Record 10-22
Type 1 Procedure PMAP Record 10-22
Type 2 Secondary Entry PMAP Record 10-23

CHAPTER 11 LOADER

GENERAL INFORMATION 11-1
LOADER SEGMENT ALLOCATION 11-2
LOADER SEGMENT TABLE PRIMARY DB 11-3
REFERENCE COUNT TABLE 11-4
SEGMENT TRANSFORM TABLE 11-4
ENTRY INDEX TABLE 11-5
DIRECTORY 11-5
ENTRY POINTER 11-6
SBUFO 11-7
DIRECTORY ENTRIES 11-8
Definitions 11-10
LOADER CACHE 11-12
LOADER COMMUNICATION TABLE 11-13

CHAPTER 12 PRIVATE VOLUMES/SERIAL DISC

MOUNTED VOLUME TABLE 12-1
PRIVATE VOLUME USER TABLE 12-4
BIND NAMES DATA SEGMENT 12-6
SERIAL DISC TABLES AND DATA STRUCTURES . . 12-8
Data Record Format 12-8
End of File Format 12-9
Contiguous Block Format 12-10
Hole Format 12-10
Gap Table Format 12-11
SDISC Extra Data Segment 12-13

Serial Disc Organization 12-15

CHAPTER 13 I/O

I/O TABLE LINKAGE 13-1
DEVICE REFERENCE TABLE (DRT) 13-2
  Series II/III 13-2
  Series 33/44 13-2
DRIVER LINKAGE TABLE (DLT) 13-3
LOGICAL-PHYSICAL DEVICE TABLE (LPDT) . . . 13-4
  LPDT Entry 13-6
OVERVIEW OF DEVICE TABLES IN DST %16 . . . 13-8
LOGICAL DEVICE TABLE 13-9
  Zero Entry Format 13-9
  Typical Entry Format 13-9
DEVICE CLASS TABLE 13-11
  Typical Entry Format 13-11
LOGICAL DEVICE TABLE EXTENSION (LDTX) . . 13-12
  Zero Entry 13-12
  Typical Entry 13-12
  Terminal Entry 13-13
  Serial Disc Entry 13-14
  CIPER Entry 13-15
  System or Private Volume Disc Entry . . 13-15
INTERRUPT LINKAGE TABLE (ILT) 13-16
  ILT for Series II/III 13-16
  ILT for Series 30/33/44 and
      Series II/III (HP-IB) 13-17
  ILT Terminology 13-18
DEVICE INFORMATION TABLE (DIT) 13-19
  DIT FOR SERIES II/III 13-19
  DIT Terminology (Series II/III) 13-20
  DIT for Series 30/33/44 13-21
  DIT Terminology (Series 30/33) 13-22
  DIT for SIO Devices 13-23
  DIT for Fixed Head Disc 13-25
  DIT for 7900A and 2888A Moving Head Disc 13-26
  DIT for 7905/7906/7920/7925 13-28
  DIT for CS 80 Disc 13-30
  DIT for 7970/71 Magnetic Tape 13-33
  DIT for 7976 Magnetic Tape 13-36
  DIT for Series III Card Reader 13-39
  DIT for HPIB Card Reader 13-41
  Line Printer DIT (Series II/III) 13-43
  2608 Line Printer DIT (HPIB Systems) . . 13-47
  2608S (HIOCIPRO) Line Printer DIT . . . 13-50
  2631 or 2619A Line Printer DIT (for
      HPIB Systems) 13-53
  2680A DIT 13-55
  I/O Status Block 13-57
  30119 Card Reader/Punch DIT 13-60

INP DIT 13-62
DISC REQUEST TABLE AND DISC REQUESTS . . . 13-70
Disc Request Table 13-71
Disc Request Element Format. 13-72
IOQ TABLE LAYOUT 13-75
IOQ Element 13-77
IO System Status Returns 13-79
IOQ Element for 7976A Magnetic Tape . . 13-81
Series II/III Line Printer IOQ Element 13-83
2608 Line Printer IOQ Element - HPIB Systems 13-86
2608S (CIPER) IOQ Element 13-89
2619A and 2631 Line Printer IOQ Element - HPIB Systems 13-92
Series III Card Reader IOQ 13-96
IOQ Element for HPIB Card Reader . . . 13-99
CS 80 Disc IOQ Element 13-101
CS 80 Integrated Cartridge Tape IOQ Element 13-104
SBUF AND TBUF TABLE LAYOUT 13-108
Table Element Allocation (TBUF and SBUF) 13-109
ICS GLOBAL 13-111
ICS Global Cells, with Initial Values 13-113
CS80 Disc Interrupt Linkage Table (CILT) 13-114

CHAPTER 14 SPOOLING

INPUT DEVICE DIRECTORY/OUTPUT DEVICE DIRECTORY 14-1
XDD Subentry Format 14-1
SPOOK OUTPUT TAPE FORMAT 14-5
Label Record 14-5
File Directory 14-6
Device and Class Directory 14-6
Logical Device Entry 14-6
Device Class Entry 14-7
Spoolfile Format 14-7
Spoolfile Block Format 14-7
Spoolfile Record Format 14-8
User Labels Information 14-8

CHAPTER 15 UNIFIED COMMAND LANGUAGE

REPLY INFORMATION TABLE 15-1
MESSAGE SYSTEM, GENERAL DESCRIPTION . . . 15-2
Message Catalog 15-2
Makecat Program 15-3
CATALOG.PUB.SYS 15-5

MESSAGE SET DIRECTORY 15-6
HELP SUBSYSTEM 15-7
  Help Directory 15-7
UDCs 15-8
  UDC Directory 15-8
  COMMAND.PUB.SYS 15-9
CI STACK DEFINITION 15-11
  Field Definitions 15-12
ASSOCIATION DST LAYOUT 15-13

CHAPTER 16 SYSDUMP/INITIAL

CTAB0 (Memory Size Independent
    Configuration Values) 16-1
  Record 0 of CONFDATA File 16-1
CTAB (Memory Size Dependent
    Configuration Values) 16-4
  Records 1-8 of CONFDATA File 16-4
DRIVER TABLE 16-6
SYSDUMP FORMAT 16-7
STORE TABLE FORMAT 16-9
  First Volume 16-9
  Subsequent Volumes 16-11
  End of Volume 16-12

CHAPTER 17 MISCELLANEOUS

LABELED TAPE SUBSYSTEM 17-1
  Format of MPE Tape Labels 17-1
    Vol1 17-1
    UVLn 17-1
    HDR1 17-2
    HDR2 17-3
    IBM 17-4
    EOV1 17-5
    EOV2 17-5
    EOF1 17-5
    EOF2 17-5
TAPE LABEL TABLE 17-6
  Tape Label Table Header Entry 17-6
  Unitialized Table (INITIAL) 17-7
  LCB Entry Format 17-8
  VCB Entry Format 17-10
BREAK POINT TABLE (DST %36) 17-11
  General Layout 17-11
  PCB Breakpoint Extension Table 17-12
  Breakpoint Entry Table 17-13

- Entry(0) 17-13
- Free Entry 17-13
- Last Entry 17-13
- Active Entry 17-13

TIMER REQUEST LIST (TRL) 17-16

MPE USER LOGGING 17-18
- General Design Overview 17-18
  - Hardware Environment 17-18
  - Software Environment 17-18
  - Design Narrative 17-18
  - Error Recovery Description 17-19
- Design Structures 17-20
  - User Logging Table (DST %33) 17-20
  - User Logging Buffer 17-26
  - Logging Identifier Table 17-35
  - Logging Record Format 17-39

MEASUREMENT INFORMATION TABLE (MEASINFOTAB) (DST %41) 17-42

CHAPTER 18 MESSAGE FILES/IPC

FILE STRUCTURE 18-1
- File Label/FCB Extent Map 18-1
- Block Structure 18-2
- Record Format 18-3
- Header Format 18-3
- Message Access Control Block 18-4
- MMSTAT Definitions 18-11

FILE SYSTEM BASIC IPC DEFINITIONS 18-13
- General Behavior 18-13
  - FCPORTOPEN Procedure 18-13
  - FCPORTSEND Procedure 18-13
  - FCPORTRECEIVE 18-13
  - FCPORTCLOSE 18-13
- IPC File's Use of IPC Mechanism 18-13
  - Reader and Writer Wait Queues 18-14
  - Completion Notification for NOWAIT I/O 18-14
  - Timeouts 18-14

PORT DATA STRUCTURES 18-15
- Port Data Segment 18-15
- Port with Two Outstanding Messages . . . 18-15
- Port Number 18-16
- Port Data Segment Array 18-16
- Port Data Segment Global Area 18-17
- Port 18-18
- Message Queue Entry (MQE) 18-19

FILE SYSTEM MESSAGE FILES 18-19
- Timer List Entry (TLE) 18-20
- MMSTAT Definitions 18-21

CHAPTER 19 MPE MEMORY RESIDENT MESSAGE FACILITY

m 505,506. 19-1
MESSAGE INTRINSICS 19-2
  SENDMSG 19-2
  PORTSTATUS 19-2
  RECEIVEMSG 19-3
SUPPORTING DATA STRUCTURES 19-4
  Message Harbor Table 19-4
  Message Tables 19-4
  Message Port Assignments 19-5

CHAPTER 20 MMSTAT EVENTS CATALOG

MMSTATS CATALOG INDEX 20-1
EVENT GROUP 0 - MEMORY MANAGER 20-2
EVENT GROUP 1 - MEMORY MANAGER 20-10
EVENT GROUP 2 - MEMORY MANAGER 20-11
EVENT GROUP 3 20-13
EVENT GROUP 4 - SCHEDULING 20-14
EVENT GROUP 5 - MESSAGE FILES (SEE CHAPTER 18)
EVENT GROUP 6 - FILE SYSTEM 20-16
EVENT GROUP 7 - FILE SYSTEM 20-25
EVENT GROUP 8 - FILE SYSTEM 20-30
EVENT GROUP 9 - DISC I/O TRANSFER 20-31
EVENT GROUP 10 - DISC ERRORS 20-32
EVENT GROUP 11 - SIO 20-33
EVENT GROUP 12 - SOFT'DEATH 20-34
EVENT GROUP 13 20-35
EVENT GROUP 14 - CS/3000 20-35
EVENT GROUP 15 - CS/3000 20-39
EVENT GROUP 16 - CS/3000 20-42
EVENT GROUP 19 - DISC CONTROLLER INTERRUPT 20-43
EVENT GROUP 20 - PRIVATE VOLUMES 20-48
EVENT GROUP 21 - PROCESS CREATION AND TERMINATION 20-49
EVENT GROUP 22 - MONITOR CONFIGURATION INFORMATION 20-50
EVENT GROUP 23 - TERMINAL I/O 20-54

CHAPTER 21 DATA COMMUNICATIONS TABLES

LOW LEVEL DS/3000 TABLES 21-1
  Data Communication IOQ Entry 21-1
  DS Line Control Block 21-3

DSGLOBAL Data Segment 21-4
  DSGLOBAL Information Table (DSGLOBINFO) 21-5
  DSXREF Table 21-6
  DSDEVICE Table 21-10
DS LPDT Entry 21-11
IODSO DIT 21-12
IODSX DIT 21-14
IODSTRMO DIT 21-16
IODSTRMX DIT 21-18
IOPADO DIT (PAD Terminal DIT) 21-20
  Pad Terminal Extra Data Segment . . . 21-34
IODSO IOQ 21-37
IODSX IOQ 21-42
IODSTRMO IOQ 21-50
IODSTRMX IOQ 21-52
IOPADO IOQ 21-54
DS I/O Table Linkage 21-60
Q-Relative Values for Drivers 21-61
DSMON/DSIOM Psuedo Header/Control Block 21-62
DSMON Request Formats 21-63
DSMONX Communication Buffers Format . . 21-74
  DSGETQ and DSPUTQ Buffer Format . . . 21-74
  DSMONG and DMONP Buffer Format 21-79
DSMONX User Channel Information Table 21-82
DSMONX Virtual Circuit Information Table 21-86
DSMONX Layout 21-89
UPPER LEVEL DS/3000 TABLES 21-90
DS Info Data Segment 21-90
DS Available File Table (DS AFT)
    (FSTYPE=1) 21-94
DS Available File Table (DS AFT)
    (FSTYPE=2 or 3) 21-95
Overall Session/Job DS Data Structure 21-96
DS-Related PCBX Structures 21-97
Job DS Extra Data Segment 21-98
Device/Process DS Extra Data Segment . . 21-100
DS Line Control Block (DSLCB) 21-102
DS Line Control Block Extension (DSLCBX) 21-103
DS Control Block (DSCB) with Message
    Header Format 21-104
Pointer Linkage in Device/Process
    DS XDS 21-106
Description of Session/Job DS
    Data Structure 21-108
  Overall Session/Job DS Data Structure 21-108
  AFT and PCBX Structures 21-109
  Job DS Extra Data Segment 21-110
  Device Process DS Extra Data Segment 21-110
  DS Line Control Block (DSLCB) 21-110
  DS Control Block (DSCB) 21-112
CS TABLES 21-113
Binary Synchronous Communication for
    CS DIT 21-113
TERMINAL TABLES 21-121
Terminal IOQ Element 21-121

DIT for ATC/Series II/III 21-126
DIT for ADCC/Series 30/33/40/44 21-142
Multipoint Terminal DIT 21-153
Multipoint Supervisor DIT 21-161

CHAPTER 22 DISC FREE SPACE MAP

DISC RESIDENT DATA STRUCTURES 22-1
Bit Map 22-1
Descriptor Table 22-1
VIRTUAL MEMORY RESIDENT DATA STRUCTURES 22-2
Disc Free Space Data Segment 22-2

CHAPTER 23 CIPER TABLES

CIPER DATA SEGMENT (CDS) OVERVIEW 23-1
Segment Header Area 23-2
Memory Allocation Manager Typical Layout 23-3
Control Table Map (CTM) 23-4
Control Table (CT) 23-5
CIPER Level 'N' Control Block 23-7
Communication Queue Head 23-8
COMQUEUELEMENT 23-10
CONTROL BLOCK INFORMATION AREA 23-11
Level 7 Control Block Information (CBI) 23-11
Level 4 Control Block Information . . . 23-18
CONTROL BLOCK INFORMATION EXTENSION . . . 23-19
General Entry Format 23-19
Level 7 Control Block Information Extension (CBIX) 23-19
Typical Record Buffer Area 23-22
LOGICAL DEVICE TABLE EXTENSION (LDTX) . . 23-24
Zero Entry 23-24
CIPER Entry 23-25
HIOCIPRO DIT (HP2608S) 23-26
CIPER IOQ ELEMENT 23-29
DEVICE REFERENCE TABLE 23-31
INTERRUPT LINKAGE TABLE 23-32

CHAPTER 1 MEMORY LAYOUT

FIXED LOW MEMORY (SERIES II/III)

```
                          -----------------------------
ABSOLUTE MEM LOC   0|  CSTB (BASE OF CST TABLE)  |0
                    |----------------------------|
                   1|          CSTXB             |1 --> CURRENTLY EXE-
                    |----------------------------|      CUTING CST EX-
                   2|          DSTB              |2     TENSION POINTER
                    |----------------------------|
                   3|          PCBB              |3
                    |----------------------------|
                   4| CPCB (CURRENT PCB POINTER) |4
                    |----------------------------|
                   5|  QI (INITIAL Q FOR ICS)    |5
                    |----------------------------|
                   6|  ZI (INITIAL Z FOR ICS)    |6
                    |----------------------------|
                   7|        MASK WORD           |7
                    |----------------------------|
                  10|        DRT BANK            |8
                    |----------------------------|
                  11|        DRT ADDRESS         |9
                    |----------------------------|    \
                  12|        RESERVED            |10  |  RESERVED FOR
                    |----------------------------|     > LOADER MAPPING
                  13|        RESERVED            |11  |  FIRMWARE
                    |----------------------------|    /
                  14|          0                 |
                    |----------------------------|
                  15|P-LABEL FOR INTERRUPT HNDLR |
                    |----------------------------|
                  16| DB SET FOR INTERRUPT HNDLR |
                    |----------------------------|
                  17|U| INTERRUPT INTERVAL VALUE |
                    |----------------------------|
```

U: set if clock interface has been used since coldload

NOTE: ALL POINTERS ARE ABSOLUTE ADDRESSES.

FIXED LOW MEMORY (SERIES 30/33)

```
 %---------------------------------------------DEC
 0|     CSTB (BASE OF CST TABLE)**             |0
   --------------------------------------------
 1|     CSTXP **                               |1 --> CURRENTLY EX-
   --------------------------------------------      ECUTING CST
 2|     DSTB (BASE OF DST TABLE)**             |2     EXTENSION BLOCK
   --------------------------------------------      POINTER
 3|     PCBB (BASE OF PCB TABLE)**             |3
   --------------------------------------------
 4|     CPCB (CURRENT PCB POINTER)**           |4
   --------------------------------------------
 5|     QI (INITIAL Q FOR ICS)**               |5
   --------------------------------------------
 6|     ZI (INITIAL Z FOR ICS)**               |6
   --------------------------------------------
 7|     SYSTEM INTERRUPT MASK WORD**           |7
   --------------------------------------------
10|     DRTBANK (BANK OF THE DRT TABLE)        |8
   --------------------------------------------
11|     DRTADDR (BASE OF DRT TABLE)            |9
   --------------------------------------------
12|     DBBANK (FOR INITIAL'S STACK)*          |10
   --------------------------------------------
13|     DB (FOR INITIAL'S STACK)*              |11
   --------------------------------------------
14|                                            |12
   --------------------------------------------
15|                                            |13
   --------------------------------------------
16|                                            |14
   --------------------------------------------
17|                                            |15
   --------------------------------------------
20|                                            |16
   --------------------------------------------
21|     LR (INTERRUPT INTERVAL)+               |17
   --------------------------------------------
22| TEMPLR (TEMP STOREAGE OF LIMIT REG)+       |18
   --------------------------------------------
23|  PCLC (PROCESS CLOCK LAST COUNT)**         |19
   --------------------------------------------
24|     PCHI (PROCESS TIME - MSW)**            |20
   --------------------------------------------
```

FIXED LOW MEMORY (SERIES 30/33) (CONT)

```
       ----------------------------------------------
    25|     PCLO (PROCESS TIME - LSW)**             |21
       ----------------------------------------------
    26|     SCST (SYSTEM CLOCK STATUS)**            |22
       ----------------------------------------------
    27|     SCLC (SYSTEM CLOCK LAST COUNT)**        |23
       ----------------------------------------------
 30-37|                                             |24-31
       ----------------------------------------------
```

NOTE: ALL POINTERS ARE ABSOLUTE ADDRESSES.

LEGEND: ** NEEDED BY FIRMWARE AND/OR BY SYSTEM, ALWAYS
* NEEDED DURING INITIAL
+ NEEDED BY MPE, SET UP BY INITIAL OR PROGENITOR.

FIXED LOW MEMORY (SERIES 44)

```
 %---------------------------------------------DEC
 0|     CSTB (BASE OF CST TABLE)**             |0
 ---------------------------------------------
 1|     CSTXP **                               |1 --> CURRENTLY EX-
  ---------------------------------------------       ECUTING CST
 2|     DSTB (BASE OF DST TABLE)**             |2     EXTENSION BLOCK
  ---------------------------------------------       POINTER
 3|     PCBB (BASE OF PCB TABLE)**             |3
 ---------------------------------------------
 4|     CPCB (CURRENT PCB POINTER)**           |4
  ---------------------------------------------
 5|     QI (INITIAL Q FOR ICS)**               |5
  ---------------------------------------------
 6|     ZI (INITIAL Z FOR ICS)**               |6
 ---------------------------------------------
 7|     SYSTEM INTERRUPT MASK WORD**           |7
  ---------------------------------------------
10|     DRTBANK (BANK OF DRT TABLE)            |8
  ---------------------------------------------
11|     DRTADDR (BASE OF DRT TABLE)            |9
  ---------------------------------------------
12|     DBBANK (FOR INITIAL'S STACK)           |10
  ---------------------------------------------
13|     DB (FOR INITIAL'S STACK)               |11
  ---------------------------------------------
14|                                            |12
  ---------------------------------------------
15|                                            |13
  ---------------------------------------------
16|                                            |14
  ---------------------------------------------
17|                                            |15
  ---------------------------------------------
20|                                            |16
  ---------------------------------------------
21|     LR (INTERRUPT INTERVAL)+               |17
  ---------------------------------------------
22| TEMPLR (TEMP STOREAGE OF LIMIT REG)+       |18
  ---------------------------------------------
23|  LR   (SYSTEM CLOCK LIMIT REGISTER)        |19
  ---------------------------------------------
24|////////////////////////////////////////////|20
  ---------------------------------------------
```

FIXED LOW MEMORY (SERIES 44)   (CONT)

```
      ----------------------------------------------
   25| TR  (TIME SINCE LAST SOFT TIMER INTERRUPT)  |21
      ----------------------------------------------
   26|     SCST (SYSTEM CLOCK STATUS)**            |22
      ----------------------------------------------
   27|     SCLC (SYSTEM CLOCK LAST COUNT)**        |23
      ----------------------------------------------
30-37|                                             |24-31
      ----------------------------------------------
```

NOTE: ALL POINTERS ARE ABSOLUTE ADDRESSES.

```
LEGEND:  ** NEEDED BY FIRMWARE AND/OR BY SYSTEM, ALWAYS
          * NEEDED DURING INITIAL
          + NEEDED BY MPE, SET UP BY INITIAL OR PROGENITOR.
```

SYSTEM GLOBAL AREA

octal		name
0	SYSGLOB - SYSBASE	
1	CST BASE - SYS BASE	SYSCST
2	DST BASE - SYS BASE	SYSDST
3	PCB BASE - SYS BASE	SYSPCB
4	ARSBM BASE - SYS BASE	SYSARSBM
5	IOQ BASE - SYS BASE	SYSIOQ
6	SBUF BASE - SYS BASE	SYSBUF
7	ICS QI - SYS BASE	SYSICS
10	LPDT BASE - SYS BASE	SYSLPDT
11	STOPS BASE - SYS BASE	SYSBPT
12	TRL BASE - SYS BASE	SYSTRL
13	JCUT BASE - SYS BASE	SYSSIR
14	SIR BASE - SYS BASE	SYSSDCTAB
15	JPCNT BASE - SYS BASE	SYSJPCNT
16	TBUF BASE - SYS BASE	SYSBUF
17	MONBUF BASE - SYS BASE	SMONBUF
20	FIRST FREE MEMORY ADDRESS	
21		
22	TIME OF LAST CYCLE	
23		
24	RESERVED	
25	SWAPTAB BASE - SYSBASE	SYSSWAPTAB

26	VDSMTAB BASE- SYSBASE	VDSMTAB
27		
30	CURRENT CST BLOCK INDEX	CSTBX
31	DISCREQTAB BASE - SYS BASE	SYSDISCREQTAB
32	DISPLACEMENT TO CODE =@CST(0)-@DST(0)	DFC
33	DISPLACEMENT TO SHARABLE = @CST(LAST)-@DST(0)	DFS
34	Not in use	
35	ABS ADDRESS (SYSDIT(8))	SYSDIT8
36	Not in use	
37	Not in use	
40	RESERVED FOR INITIAL (VDSENTRY)	
41	RESERVED FOR INITIAL (VDSMAP)	
42	SRTTAB BASE - SYS BASE	SRTTAB
43	SPECQ HEAD - SYS BASE	SYSSPECQHEAD
44	ARL BASE - SYS BASE	SYSARLD
45	# PAGES IN LARGEST CURRENTLY AVAILABLE REGION	SYSMAXAVAILREG
46	MAKE OVERLAY CANDIDATE INFORMATION	MOCINFO
47	NUMBER OF MEMORY BANKS CONFIGURED -1	SYSNBANKS
50	SCHEDULER TO AWAKE MESSAGE	SCHEDTOAWAKEMSG
51	POINTER TO CSTBLK TABLE	CSTXBLOCKPOINTER
52	AWAKE TO SCHEDULER MESSAGE	AWAKETOSCHEDMSG
53	WAIT TO SCHEDULER MESSAGE	
54	CURRENT ACTIVITY'S PRIORITY	CURACTPRI

SYSTEM GLOBAL AREA (cont)

```
            octal                                                     name
            -----                                                     ----
                 |------------------------------------------------|
             /55|            BUSY TABLE POINTER                   |BUSY
             |  |------------------------------------------------|
             |56|            HEAD TABLE POINTER                   |HEAD
             |  |------------------------------------------------|
             |57|            TAIL TABLE POINTER                   |TAIL
             |  |------------------------------------------------|
             |60|       # OF SIO PROGRAMS EXECUTING               |SIOCOUNT
             |  |------------------------------------------------|
             |61|        PARITY ERROR FLAG (MEM PE)               |PARITY
             |  |------------------------------------------------|
             |62| Impeded queue head for message buffer (PIN)    |IOMSGPIN
             |  |------------------------------------------------|
             |63|      I/O Message system error flags             |IOLOGQX
             |  | (0:1) - No SYSBUF avail for I/O error logging  |
             |  | (1:1) - No SYSBUF for IOMESSAGE (GENMSG)       |
             |  |------------------------------------------------|
reserved     |64|        # OF TERMINALS READING                   |RDCOUNT
for I/O   <     |------------------------------------------------|
system       |65|        # OF TERMINALS WRITING                   |WRTCOUNT
             |  |------------------------------------------------|
             |66|               DSET B                            |CRIO
             |  |------------------------------------------------|
             |67|                                                 |CRIO
             |  |             LAST TIMER                          |
             |70|                                                 |CRIO
             |  |------------------------------------------------|
             |71|          HIGHEST DRT NUMBER                     |HSYSDRT
             |  |------------------------------------------------|
             |72|              POWERFAIL                          |POWERFAIL
             |  |------------------------------------------------|
             |73|            SYSTEM UP FLAG                       |SYSUP
             |  |------------------------------------------------|
             \74|  SYS CONSOLE LOGICAL DEVICE NUMBER              |CONSLDEV
                |------------------------------------------------|
           /  75|           COLD LOAD COUNT                       |CLOADID
           |    |------------------------------------------------|
           |  76|           SHARED FCB DST                        |SHFCBDST
           |    |------------------------------------------------|
           |  77|          MONITORING FLAGS                       |
           |    |------------------------------------------------|
reserved   |100|                                                  |
for file<       |        MAX # OF SPOOL SECTORS                   |MAXSSECT
system     |101|                                                  |
           |    |------------------------------------------------|
```

```
    |----------------------------------------------|
|102|                                              |
|   |        CURRENT # OF SPOOL KILOSECTORS        |NUMSSECT
|103|                                              |
|   |----------------------------------------------|
\104|          # SECTOR/SPOOFLE EXTENT             |EXTSSECT
    |----------------------------------------------|
 105|           MAX CODE SEGMENT SIZE              |
    |----------------------------------------------|
 106|       MAX # OF CODE SEGMENTS/PROCESS         |
    |----------------------------------------------|
 107|          MAX STACK SIZE (MAXDATA)            |
    |----------------------------------------------|
 110|            DEFAULT STACK SIZE                |
    |----------------------------------------------|
 111|        MAX EXTRA DATA SEGMENT SIZE           |
    |----------------------------------------------|
 112|     MAX # EXTRA DATA SEGMENTS/PROCESS        |
    |----------------------------------------------|
 113|      DST number for MESSAGE buffers          |
    |----------------------------------------------|
 114|               UPDATE LEVEL                   |UPDATEL
    |----------------------------------------------|
 115|                FIX LEVEL                     |FIXL
    |----------------------------------------------|
 116|              VERSION LEVEL                   |VERSION
    |----------------------------------------------|
 117|          DEFAULT CPU TIME LIMIT              |
    |----------------------------------------------|
 120|          # OF SECONDS TO LOGON               |
    |----------------------------------------------|
 121|          JOBSYNCH BITS (13:3)                |
    |----------------------------------------------|
 122|        EXTERNAL PLABEL OF INITIATE           |
    |----------------------------------------------|
 123|        INTERNAL PLABEL OF INITIATE           |
    |----------------------------------------------|
 124|              MAXSYSDST                       |
    |----------------------------------------------|
 125|              MAXSYSCST                       |
    |----------------------------------------------|
 126|  SL.PUB.SYS LDEV    |       SL.PUB.SYS       |
    |----------------------------------------------|
 127|               DISC ADDRESS                   |
    |----------------------------------------------|
 130|               (DIRECTORY)                    |
    |                                              |
 131|             (DISC ADDRESS)                   |
    |----------------------------------------------|
```

SYSTEM GLOBAL AREA (cont)

```
        octal                                                     name
        -----                                                     ----
             |----------------------------------------------|
          132|                  SPOOLINDEX                  |
             |----------------------------------------------|
         /133|            EXT LABEL FOR SHOWCOM             |
         |   |----------------------------------------------|
         |134|                                              |
         |   |----------------------------------------------|
         |135|               CS IOWAIT PLABEL               |
reserved<    |----------------------------------------------|
for CS   |136|                     |      CS FIX LEVEL      |
         |   |----------------------------------------------|
         |137|                  CS VERSION                  |
         |   |----------------------------------------------|
         \140|                 CCLOSE PLABEL                |
             |----------------------------------------------|
          141|       LOGICAL PROCESS TABLE (PROGEN)        0|
             |----------------------------------------------|
          142|//////////////////////////////////////////////|
             |----------------------------------------------|
          143|        LOGICAL PROCESS TABLE (UCOP)          2|
             |----------------------------------------------|
          144|        LOGICAL PROCESS TABLE (PFAIL)         3|
             |----------------------------------------------|
          145|       LOGICAL PROCESS TABLE (DEVREC)         4|
             |----------------------------------------------|
          146|        LOGICAL PROCESS TABLE (DRUSG)         5|
             |----------------------------------------------|
          147|        LOGICAL PROCESS TABLE (STMSG)         6|
             |----------------------------------------------|
          150|         LOGICAL PROCESS TABLE (LOG)          7|
             |----------------------------------------------|
          151|        LOGICAL PROCESS TABLE (LOAD)          8|
             |----------------------------------------------|
          152|     LOGICAL PROCESS TABLE (IOMESSPROC)       9|
             |----------------------------------------------|
          153|       LOGICAL PROCESS TABLE SYSIOPRDC       10|
             |----------------------------------------------|

             |----------------------------------------------|
          154|        LOGICAL PROCESS TABLE MEMLOGP        11|
             |----------------------------------------------|
          155|       EXTERNAL PLABEL OF "TERMINATE"         |
             |----------------------------------------------|
```

```
             |------------------------------------------------|
          156|          INTERNAL PLABEL OF "TERMINATE"        |
             |------------------------------------------------|
          157|        EXTERNAL PLABEL OF "COMMANDINTERP"      |
             |------------------------------------------------|
          160|        INTERNAL PLABEL OF "COMMANDINTERP"      |
             |------------------------------------------------|
          161|          EXTERNAL PLABLE OF "SPOOLIN"          |
             |------------------------------------------------|
          162|           INTERNAL PLABLE OF "TRACE0"          |
             |------------------------------------------------|
          163|           EXTERNAL PLABEL OF "TRACE0"          |
             |------------------------------------------------|
          164|          INTERNAL PLABEL OF "SPOOLIN"          |
             |------------------------------------------------|
          165|          EXTERNAL PLABLE OF "SPOOLOUT"         |
             |------------------------------------------------|
          166|          INTERNAL PLABEL OF "SPOOLOUT"         |
  -----      |------------------------------------------------|
    |     167|                     3 WORD                     |
    |        |------------------------------------------------|
    |     170|                     LOGGING                    |
    |        |------------------------------------------------|
    |     171|                      MASK                      |
    |        |------------------------------------------------|
    |     172|///////////|STATE |      DST# - BUFFER 0        | STATE:
    |        |------------------------------------------------|  0 EMPTY
    |     173|///////////|STATE |      DST# - BUFFER 1        |  1 CUR
    |        |------------------------------------------------|  2 FULL
    |     174|            BUFFER LENGTH (SECTORS)             |
    |        |------------------------------------------------|
    |     175|              FREE AREA POINTER                 |
    |        |------------------------------------------------|
    |     176|                     FLAGX                      |
reserved     |------------------------------------------------|
   for    177|          # RECORDS WRITTEN IN BUFFER 0         |
logging      |------------------------------------------------|
    |     200|          # RECORDS WRITTEN IN BUFFER 1         |
    |        |------------------------------------------------|
    |     201|          FILE SIZE (BLOCKS) - 1ST HALF         |
    |        |------------------------------------------------|
    |     202|          FILE SIZE (BLOCKS) - 2ND HALF         |
    |        |------------------------------------------------|
    |     203|                (LOG FILE SIZE)                 |
    |        |------------------------------------------------|
    |     204|                    (BLOCKS)                    |
    |        |------------------------------------------------|
    |     205|              LOG FILE NUMBER   (LOGFILENUM)    |
    |        |------------------------------------------------|
    |     206| NUMBER OF LOGGING [BLOCKS WRITTEN (1ST HALF)]  |
    |        |------------------------------------------------|
    |     207|  BLOCKS WRITTEN [BLOCKS WRITTEN (2ND HALF)]    |
  -----      |------------------------------------------------|
```

SYSTEM GLOBAL AREA (cont)

```
        octal                                                  name
        -----                                                  ----
-----------   |-----------------------------------------------|
     |    210|          (TOTAL # LOG RECORDS MISSED)          |
     |       |-----------------------------------------------|
     |    211|              (DUE TO LOG FAILURE)              |
     |       |-----------------------------------------------|
     |    212| TOTAL# RECORDS MISSED - "JOB INITIATION" LOSS |
     |       |-----------------------------------------------|
logging   213|TOTAL# RECORDS MISSED - "JOB TERMINATION" LOSS |
     |       |-----------------------------------------------|
     |    214|    OPERATOR CONSOLE JOBSESSION # AT STARTUP    |
     |       |-----------------------------------------------|
     |    215|                  GLOBAL                        |
     |       |                                               |
     |    216|                   ALLOW                       |
     |       |                                               |
     |    217|                  MASK                          |
    -----    |-----------------------------------------------|
          220|                                               |
             ~                                               ~
                               LOADER
                               MESSAGE
                                TABLE
             ~                                               ~
    -----    |-----------------------------------------------|
|         250|                                               |
|            |-----------------------------------------------|
reserved  251|                                               |
for          |-----------------------------------------------|
segment   252|                                               |
trace        |-----------------------------------------------|
|         253|                                               |
|            |-----------------------------------------------|
```

```
|            |------------------------------------------------|
reserved  254|                                                |
for          |------------------------------------------------|
segment   255|                                                |
trace        |------------------------------------------------|
|         256|                                                |
|            |------------------------------------------------|
|         257|                                                |
|   ---------|------------------------------------------------|
          260|                     STMON                      |
             |------------------------------------------------|
          261|               MEASINFOTABPTR                   |
             |------------------------------------------------|
          262|    MEASUREMENT STATISTICS CLASS MASK           |GCLASSENABLEDMASI
             |------------------------------------------------|
          263|       CLASS 0 STATISTICS BANK NUMBER           |MEASSTATXDSBANK
             |------------------------------------------------|
          264|         CLASS 0 STATISTICS ADDRESS             |MEASSTSTXDSBASE
             |------------------------------------------------|
          265|                                                |
             |                 SCAN POINT                     |
          266|                                                |
             |------------------------------------------------|
          267|               MEASFLAGS                        | **
             |------------------------------------------------|
          270|             RESERVED                           |
    -----    |------------------------------------------------|
      |   271| Sysbase index of PCB at head of Dispatching Q  |SYSDISQHEAD
      |      |------------------------------------------------|
      |   272| Sysbase index of PCB at tail of Dispatching Q  |SYSDISPQTAIL
      |      |------------------------------------------------|
          273|             RESERVED                           |
      |      |------------------------------------------------|
      |   274|             RESERVED                           |
     misc    |------------------------------------------------|
      |   275|             RESERVED                           |
      |      |------------------------------------------------|
      |   276|          HELP LOGICAL DEVICE NUMBER            |
      |      |------------------------------------------------|
      |   277|                CURRENT LOGON DST               |  DSTLOGON
    -----    |------------------------------------------------|
      |   300|                     (STOP)                     |
      |   301|                     (BITS) (see p. 2-15)       |
      |      |------------------------------------------------|
      |   302|               # PROCESS ENTRIES                |
      |      |------------------------------------------------|
      |   303|                                                |
      |      |------------------------------------------------|
```

```
              |----------------------------------------------|
    |     304|       DEVREC PIN        |          2         |
    |        |----------------------------------------------|
    |     305|                  %20                         |
    |        |----------------------------------------------|
    |     306|        UCOP PIN         |          0         |
    |        |----------------------------------------------|
    |     307|                  %20                         |
process      |----------------------------------------------|
stop      310|         LOG PIN         |          1         |
table        |----------------------------------------------|
    |     311|                  %20                         |
    |        |----------------------------------------------|
    |     312|       IOMESS PIN        |          3         |
    |        |----------------------------------------------|
    |     313|                  %20                         |
    |        |----------------------------------------------|
    |     314|       MEMLOGP PIN       |          4         |
    |        |----------------------------------------------|
    |     315|                  %20                         |
    |        |----------------------------------------------|
    |     316|               RESERVED                       |
    |        |----------------------------------------------|
    |     317|               RESERVED                       |
  -----      |----------------------------------------------|
  ----- 320|       DSGLOBAL DATA SEGMENT DST NUMBER         |
    |        |----------------------------------------------|
    |     321|      RESERVED FOR DS/3000 (SET TO ZERO)      |
    |        |----------------------------------------------|
    |     322|      RESERVED FOR DS/3000 (SET TO ZERO)      |
    |        |----------------------------------------------|
    |     323|               SDSLDEV PLABEL                 |
    |        |----------------------------------------------|
   DS     324|      RESERVED FOR DS/3000 (SET TO ZERO)      |
    |        |----------------------------------------------|
    |     325|      RESERVED FOR DS/3000 (SET TO ZERO)      |
    |        |----------------------------------------------|
    |     326|      RESERVED FOR DS/3000 (SET TO ZERO)      |
    |        |----------------------------------------------|
    |     327|      RESERVED FOR DS/3000 (SET TO ZERO)      |
  -----      |----------------------------------------------|
          330|                DISC STATUS                   | LAST
             |----------------------------------------------| DISC
          331|          LDEV           |        DISC        | SIO
             |----------------------------------------------| ERROR
          332|                  AONESS                      |
             |----------------------------------------------|
          333|                 MAXQUEUE                     |
             |----------------------------------------------| JOBPRI
          334|               DEFAULTQUEUE                   |
             |----------------------------------------------|
```

```
          |-----------------------------------------------|
       335|                DSCHECK PLABEL                 |
          |-----------------------------------------------|
       336|                DSOPEN PLABEL                  |
          |-----------------------------------------------|
       337|                DSCLOSE PLABEL                 |
          |-----------------------------------------------|
       340|           MANAGEWRITE CONV. PLABEL            |
          |-----------------------------------------------|
       341|              CONSDSLINE' PLABEL               |
          |-----------------------------------------------|
       342|               CXREMOTE PLABEL                 |
          |-----------------------------------------------|
       343|               CXDSLINE PLABEL                 |
          |-----------------------------------------------|
       344|                CXRFA PLABEL                   |
          |-----------------------------------------------|
       345|               DSIMAGE PLABEL                  |
          |-----------------------------------------------|
       346| DEFAULT LABEL TYPE    | TAPE LBL AUTO REC FUN |
          |-----------------------------------------------|
       347| SYSDB PTR TO TERM INIT CHNL PGM (S30/33 ONLY) |
          |-----------------------------------------------|
       350|                                           |SD| Softdeath flag
          |-----------------------------------------------|
       351|                                               |
          |             LAST CYCLE DURATION               |
       352|                                               |
          |-----------------------------------------------|
       353|                                               |
          |               CYCLE THRESHOLD                 |
       354|                                               |
          |-----------------------------------------------|
       355|      BUG CATCH ENABLE CELL                    |
          |-----------------------------------------------|
       356|    MONITOR BUFFER     |       TIMESTAMP       | MONBUFT0
          |-----------------------------------------------|
       357|    MONITOR BUFFER     |       TIMESTAMP       | MONBUFT1
          |-----------------------------------------------|
       360|               DSBREAK PLABEL                  |
          |-----------------------------------------------|
       361| Bank of last memory word                      |LAST MEMORY
          |-----------------------------------------------|
       362| Base of last memory word                      |ADDRESS
          |-----------------------------------------------|
      /363|                 PVPROC PIN                    |
      |   |-----------------------------------------------|
      |364|             PV RECOGNITION COUNT              |
Private<  |-----------------------------------------------|
Volumes |365|   VMOUNT FLAGS                  |AUTO|ALL|ON|
      |   |-----------------------------------------------|
```

```
    |-------------------------------------------------|
|366|                                                 |
|   |-------------------------------------------------|
|367|                                                 |
    |-------------------------------------------------|
\370|                                                 |
    |-------------------------------------------------|
 371| MSG CATALOG LDEV   |                            |
    |---------------------                            |
 372|         MESSAGE CATALOG DISC ADDRESS            |
    |-------------------------------------------------|
 373|                  MSG DSTN                       |
    |-------------------------------------------------|
 374|              CONSMPLINE' PLABEL                 |
    |-------------------------------------------------|
 375|                 CONSMRJE PLABEL                 |
    |-------------------------------------------------|
 376| SYSTEM LEVEL UDC FLAG (1 = SYS UDC'S EXIST)     |
    |-------------------------------------------------|
 377|  SYSDB RELATIVE POINTER TO SYSGLOB EXTENSION    |
    |-------------------------------------------------|
 400|  CPU NUMBER ( Set by the firmware )             |
    |-------------------------------------------------|
```

SYSGLOB EXTENSION (%200 LONG; POINTER AT SYSDB+%377)

```
    |----------------------------------------------|
% 0|        Swap Queue Delay (*100ms)             |SWAPQDELAY
    |----------------------------------------------|
   1|  Bank of First Region in Linked Memory       |FIRST
    |----------------------------------------------|MEMORY
   2|  Base of First Region in Linked Memory       |REGION
    |----------------------------------------------|
   3|      Garbage Collection Enable Flag          |GARBCOLLENAB
    |----------------------------------------------|
   4|   Move Threshold (in pages, for garb coll)   |MOVETHRESH
    |----------------------------------------------|
   5|      Main Memory Page Size (in words)        |
    |----------------------------------------------|
   6|              VDS PAGE SIZE                   |
    |----------------------------------------------|
   7|                                              |
    |           LAST MAKE ROOM TIME                |
   8|                                              |
    |----------------------------------------------|
   9|   MEMORY PRESSURE DURATION THRESHOLD         |
    |----------------------------------------------|
    ~                                              ~
    ~                                              ~
```

```
   |-----------------------------------------------|
 57|///////////////////////////////////////////////|
   |-----------------------------------------------|
 60|         PLABEL USERLOG (EXTERNAL)             |
   |-----------------------------------------------|
 61|         PLABEL USERLOG (INTERNAL)             |
   |-----------------------------------------------|
 62|         PLABEL RECLOG  (EXTERNAL)             |
   |-----------------------------------------------|
 63|         PLABEL RECLOG  (INTERNAL)             |
   |-----------------------------------------------|
 64|         PLABEL RESTART (EXTERNAL)             |
   |-----------------------------------------------|
 65|         PLABEL RESTART (INTERNAL)             |
   |-----------------------------------------------|
 66|         PMBC LOW CORE BANK #  (USER)          |
   |-----------------------------------------------|
 67|         PMBC LOW CORE ADDRESS (USER)          |
   |-----------------------------------------------|
 70|         RESERVED FOR IMAGE   DST of cosmic control block
   |-----------------------------------------------|
 71|       RESERVED FOR MEASIO         12| MIOCNT  | *
   |-----------------------------------------------|
 72|   LOADER CACHE SEGMENT NUMBER                 |
   |-----------------------------------------------|
 73|   PLABEL 3270    (EXTERNAL)                   |
   |-----------------------------------------------|
 74|           MIT UPDATE                          |
   |-----------------------------------------------|
 75|           MIT FIX                             |
   |-----------------------------------------------|
 76|           MIT VERSION                         |
   |-----------------------------------------------|
 77|      COUNT OF TAPE CONTROLLERS USING MEASIO   |
   |-----------------------------------------------|
   |                                               |
   ~                                               ~
100|       PORT DATA SEGMENT NUMBER                |
   |-----------------------------------------------|
101|  RESERVED FOR SECOND PORT DATA SEGMENT        |
   |-----------------------------------------------|
102| SYSTEM FPMAP OPTION FLAG                      |SYSFPMAP
   |-----------------------------------------------|
```

```
 * MIOCNT = MEASIOCOUNT  (3 BITS)
** MEASFLAGS   (15:1) = 1 ==> MONITOR ENABLED
               (14:1) = 1 ==> BUFFER FLIP/FLOP
               (13:1) = 1 ==> EOT ON MONITOR TAPE
```

SYSDB WORDS

ADDRESS	NAME	FUNCTION
DB+55	BUSY	- SYSDB relative pointer to BUSY TABLE for I/O resources
DB+56	HEAD	- SYSDB relative pointer to table containing head pointers to I/O resource queues
DB+57	TAIL	- SYSDB relative pointer to table containing head pointers to tail of I/O resource queues
DB+60	SIO COUNT	- Number of I/O Programs currently executing
DB+72	POWER FAIL	- 0-no power fail 1-system disc recovery 2-all other disc recovery 3-all other device recovery
DB+73	SYSUP	- System is up and operable
DB+74	CONSLDEVN	- System console logical device number

JOBSYNCH job synchronization via jobsynch (sysglob+121(8))

(13:1) - JOBSREADY - set by DEVREC & MORGUE (via procedure STARTDEVICE) indicating a ready job. This prevents UCOP from going to a wait state when a job is just made ready.

(15:1) - DEVFREED - set by DEALLOCATE when device count goes to 0.

NOTE - Both bits above used for synchronization of job-made-ready or devicefreed when UCOP is running.

(14:1) - JOBSWAITING- set by UCOP just before waiting if any job is waiting for list device. Signals DEALLOCATE to awake UCOP when a device is freed.

ALLOW MASK FORMAT

	BIT	COMMAND
WORD 1	0	ABORTIO
	1	ACCEPT
	2	DOWN
	3	GIVE
	4	HEADOFF
	5	HEADON
	6	REFUSE
	7	REPLY
	8	STARTSPOOL
	9	TAKE
	10	UP
	11	MPLINE
	12	DSCONTROL
	13	ABORTJOB
	14	ALLOW
	15	ALTSPOOLFILE
WORD 2	0	ALTJOB
	1	BREAKJOB
	2	DELETESPOOLFILE
	3	DISALLOW
	4	JOBFENCE
	5	LIMIT
	6	STOPSPOOL
	7	SUSPENDSPOOL
	8	OUTFENCE
	9	RECALL
	10	RESUMEJOB
	11	RESUMESPOOL
	12	STREAMS
	13	CONSOLE
	14	WARN
	15	WELCOME
WORD 3	0	MON
	1	MOFF
	2	VMOUNT
	3	LMOUNT
	4	LDISMOUNT
	5	MRJECONTROL
	6	JOBSECURITY
	7	DOWNLOAD
	8	MIOENABLE
	9	MIODISABLE
	10	LOG
	11	FOREIGN
	12	IMLCONTROL
	13	SHOWCOM

LOGGING RELATED LOCATIONS

```
SYSDB
      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
 172 |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 or  |/////////|STATE|                DST #       |
 173 |---------------------------------------------|

      STATE = 0 if respective buffer empty
              1 if respective buffer is current
              2 if respective buffer is full
```

FLAGX

```
SYSDB
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     |--|--|--|--|--|--|--|--|--|--|--|--|--|---|--|--|
 176 |/////////////////////////////////|SF|HF|BUF|SL|SD|
     |------------------------------------------------|

 SF = 1 if soft failure
 HF = 1 if hard failure
BUF = 0 if current log buffer is buffer 0
    = 1 if current log buffer is buffer 1
 SL = 1 to indicate a switch in log buffers (from 0 to 1 or from 1
        to 0)
 SD = 1 to indicate shutdown in progress
```

PROCESS STOP LIST GENERAL LAYOUT

```
SYSDB
     |-----------------------------------------------|
 300 |         STOP BITS REPRESENTING WHICH          |
     |       PROCESSES TO STOP ON "SHUTDOWN"         |
     |-----------------------------------------------|
     |               # PROCESS ENTRIES               |
     |-----------------------------------------------|
     |///////////////////////////////////////////////|
     |-----------------------------------------------|
     |               1ST PROCESS ENTRY               |
     |-----------------------------------------------|
     |               2ND PROCESS ENTRY               |
     |-----------------------------------------------|
     |                       .                       |
     |                       .                       |
     |                       .                       |
     |                       .                       |
     |-----------------------------------------------|
 317 |              LAST PROCESS ENTRY               |
     |-----------------------------------------------|
```

ENTRY FORMAT

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|     PROCESS PIN #      |      STOP BIT #       |
|-----------------------------------------------|
|              PROCESS WAIT STATE               |
|-----------------------------------------------|
```

PREASSIGNED ENTRIES

entry #	process	stop bit #
1	devrec	2
2	ucop	0
3	log	1

This section is a description of the method used by INITIAL to allocate memory for MPE tables and code segments in MPE IV. All memory allocated by INITIAL is permanently allocated. All non-core resident code and data is put on disc before exiting INITIAL.

At the most basic level INITIAL will try to build memory to look exactly as diagrammed below. There are, however, several ways in which to deviate from this structure. Before going into the sources of these deviations, it is necessary to point out which portions of memory are used by INITIAL during the restart and therefore cannot be used by MPE until INITIAL has finished. Before INITIAL begins to allocate any memory space, it relocates its core resident code, its code segment swapping area and its stack to the highest configured memory space. Additionally, it uses the last %240 words of bank 0 on a series III and the last %326 words of bank 0 on series 30, 33 and 44 for its I/O buffer area and temporary code segment table. After INITIAL has built all of core resident MPE (tables and code), it builds the disc resident MPE tables. Since some of the disc resident tables may be too large to be built in INITIAL's stack, these tables are built in unused memory space. Therefore, in addition to the memory space required for INITIAL's code, INITIAL's stack and core resident MPE, there must be enough space left in which to build the largest of the disc resident tables.

INITIAL will essentially build memory in the order shown below, however, there may be an unused fragment of memory between the DRT's and the system global area which INITIAL will fill with the smaller tables. Neither the tables marked with an asterisk nor the code segments will ever be put in this area.

Beginning with the B MIT, all bank 0 dependencies have been removed from core resident MPE code. If there is insufficient space in bank 0 for any core resident code segment, INITIAL will put it into bank 1. At the present time core resident MPE is not large enough to occupy more than all of bank 0 and part of bank 1. If the system being built by INITIAL is configured with 128K words or 160K words of memory then INITIAL's stack will be in bank 1 (the code also on a 128K word memory size). If INITIAL is occupying part of bank 1 and the space is needed for a core resident MPE code segment or to build a disc resident table then INITIAL will print the error message "ERROR #350 OUT OF MEMORY".

Except for the exceptions stated above, for every allocation of memory INITIAL will first try to allocate any remaining space between the DRT's and SYSDB. It will then try the next available space in bank 0, then the next available space in bank 1. If it were necessary it could continue searching until all all banks were checked for available space.

Immediately before exiting INITIAL, INITIAL lays down all the memory region headers and trailers as shown below. For any one bank of memory there will only be one block of core resident MPE, regardless of its contents. The only block of core resident MPE that does not have a reserved region global header is in bank 0. It does have the reserved region global trailer though. Before placing any code outside bank 0 the first %23 words of every bank (except bank 0) is reserved for the region global header.

Initial Memory Layout

```
|----------------------------------|
|//////////// BANK 0 //////////////|
|----------------------------------|
|          Low Core memory         |
|----------------------------------|
|                DRT               |
|----------------------------------|
|        System Global area        |
|----------------------------------|
|           Firmware area          |
|----------------------------------|
|         SYSGLOB Extension        |
|----------------------------------|
|               TBUF's             |
|----------------------------------|
|              *DIT's              |
|----------------------------------|
|                DST               |
|----------------------------------|
|                CST               |
|----------------------------------|
|               CSTX               |
|----------------------------------|
|                PCB               |
|----------------------------------|
|                ICS               |
|----------------------------------|
|               *IOQ               |
|----------------------------------|
|        Disc Request Table        |
|----------------------------------|
|             ILT/DLT              |
|----------------------------------|
|        I/O resource Table        |
|----------------------------------|
|          *System Buffers         |
|----------------------------------|
|            Swap Table            |
|----------------------------------|
|          CST Block Table         |
|----------------------------------|
|       Special Request Table      |
|----------------------------------|
|       Message Harbor Table       |
|----------------------------------|
|       Primary Message Table      |
|----------------------------------|
|   Measurement Information Table  |
|----------------------------------|
|              VDSMTAB             |
|----------------------------------|
```

```
|----------------------------------|
|            ARSMB Table           |
|----------------------------------|
|      Available Region List       |
|----------------------------------|
|               LPDT               |
|----------------------------------|
|       Timer Request Queue        |
|----------------------------------|
|        Job Process Count         |
|----------------------------------|
|        Job Cutoff Table          |
|----------------------------------|
|            Sir Table             |
|----------------------------------|
| Memory Management Monitor Buffer |
|----------------------------------|
|                                  |
|                                  |
|                                  |
|                                  |
| Core Resident CST's in CST order |
|                                  |
|                                  |
|                                  |
|                                  |
|----------------------------------|
|  Reserved Region Global Trailer  |
|----------------------------------|
|  Available Region Global Header  |
|----------------------------------|
|                                  |
|                                  |
|                                  |
|                                  |
|        Available Memory          |
|                                  |
|                                  |
|                                  |
|                                  |
|  --------------------------------|
|  Available Region Global Trailer |
|  --------------------------------|
| /////////// BANK 1 //////////////|
|  --------------------------------|
|   Reserved Region Global Header  |
|  --------------------------------|
```

```
|----------------------------------|
|                                  |
|                                  |
|                                  |
|                                  |
| Core Resident CST's that didn't  |
|         fit in BANK 0            |
|                                  |
|                                  |
|                                  |
|                                  |
|----------------------------------|
| Reserved Region Global Trailer   |
|----------------------------------|
| Available Region Global Header   |
|----------------------------------|
|                                  |
|                                  |
|                                  |
|                                  |
|         Available Memory         |
|                                  |
|                                  |
|                                  |
|                                  |
|----------------------------------|
| Available Region Global Trailer  |
|----------------------------------|
|////////// BANK BOUNDRY //////////|
|----------------------------------|
| Available Region Global Header   |
|----------------------------------|
|                                  |
|                                  |
|                                  |
|         Available Memory         |
|                                  |
|                                  |
|                                  |
|                                  |
|                                  |
|                                  |
|----------------------------------|
| Available Region Global Trailer  |
|----------------------------------|
|////////// BANK BOUNDRY //////////|
|----------------------------------|
|               ETC.               |
|                                  |
|                                  |
|                                  |
```

CHAPTER 2 Memory Management Tables

2.1 Segment Table Structure

The current location and state of each data segment and loaded code segme is maintained in the segment table. This table is partitioned into three parts, as shown in Figure 2-1. The partitions are based on the segment classes: a segment is a data segment, a segment is a system sl segment, segment is part of a program. The structure and format of each partition is described in the following.

Overall ST Structure

```
  (%2),(%1002)+SYSBASE -------->   +-------+
                                   |       |
                                   |  DST  |
                                   |       |
  (%0),(%1001)+SYSBASE -------->   +-------+
                                   |       |
                                   |  CST  |
                                   |       |
                                   +-------+
            CSTXMAP     +-------->  | First |
           +-------+    |          |Loaded |
(%1051) -->|       |    |          |Program|
           +-------+    |          |       |
           |       |----+  +------>+-------+<-- (%3), current program pointer
           +-------+       |       |  Next |
           |       |       |       |Loaded |
           |       |       |       |Program|
           |       |       |       +-------+
           +-------+       |       |       |
           |       |-------+       |       |
           +-------+               |       |
           |       |               +-------+
           |       |
           +-------+
```

Figure 2-1

2.1.1 Pointers and DST #'s of Segment Table Components

i. DST

% 2 absolute address of entry 0 of the DST
%1002 sysbase relative index of entry 0 of DST
DST# =2

ii. CST

% 0 absolute address of entry 0 of system sl
%1001 sysbase relative index of entry 0 of system sl
%1032 displacement from DST base of entry 0 of system sl
DST# =1

iii. CSTX

% 1 absolute address of entry 0 of current program
%1033 displacement from DST base to first CSTX entry sl
      = @ CST (LAST) - @ DST (0) = DFS
DST# =4

iv. CSTXMAP

%1051 sysbase relative index of entry 0 of CSTXMAP
DST# =43 (%71)

2.1.2 Standard Segment Identifier Format

```
   SEGIDENTIFIER.(0:1) = 1 ==> SEG IS PART OF A PROGRAM
                           ==> (1:7) = PROGRAM INDEX INTO CSTXBLK,
                               (8:8) = LOGICAL SEG NUMBER (0-63)
   SEGIDENTIFIER.(0:2) = 0 ==> SEG IS A DATA SEGMENT,
                               (2:14) = DST ENTRY NUMBER
   SEGIDENTIFIER.(0:2) = 1 ==> SEG IS AN SL SEGMENT,
                               (2:14) = SL ENTRY NUMBER
      EQUATE SEGIDDATATYPE=0,
```

2.1.3 DST Entry Formats

DST Entry 0 Format

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  Word 0|# OF CONFIGURED ENTRIES                         |
         |-----------------------------------------------|
  Word 1|ENTRY LENGTH                                    |
         |-----------------------------------------------|
  Word 2|# OF AVAILABLE ENTRIES                          |
         |-----------------------------------------------|
  Word 3|TABLE RELATIVE INDEX TO FIRST FREE ENTRY         |
         |-----------------------------------------------|
```

DST General Entry Format

Case (i) DST Entry for a Present Data Segment

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
Word 0   |A |O |R |              SIZE/4                  | FIRMINFO
         |-----------------------------------------------|
Word 1   |D |R |I |S |M |F |S |C |W|                     |
         |C |O |M |T |O |W |Y |O |D|     VMALLOC         |  FLAGS
         |V |C |I |K |D |I |S |R | |                     |
         |  |  |  |  |  |P |  |E | |                     |
         |-----------------------------------------------|
Word 2   |                   BANK                        | MMBANK
         |-----------------------------------------------|
Word 3   |                   BASE                        | MMBASE
         |-----------------------------------------------|
```

Case (ii) DST Entry for an Absent Data Segment

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
        |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
Word 0  |A |O |R |              SIZE/4                  | FIRMINFO
        |-----------------------------------------------|
Word 1  |D |R |I |S |M |F |S |C |W|                     |
        |C |O |M |T |O |W |Y |O |D|      VMALLOC        |  FLAGS
        |V |C |I |K |D |I |S |R | |                     |
        |  |  |  |  |  |P |  |E | |                     |
        |-----------------------------------------------|
Word 2  |    L DEV #            |       HODA            | HODA
        |-----------------------------------------------|
Word 3  |                 LODA                          | LODA
        |-----------------------------------------------|
```

2.1.6 CST Entry Formats

CST Entry 0 Format

```
         0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
        |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 Word 0|# OF CONFIGURED ENTRIES                          |
        |-----------------------------------------------|
 Word 1|ENTRY LENGTH                                     |
        |-----------------------------------------------|
 Word 2|# OF AVAILABLE ENTRIES                           |
        |-----------------------------------------------|
 Word 3|TABLE RELATIVE INDEX TO FIRST FREE ENTRY         |
        |-----------------------------------------------|
```

CSTX ENtry 0 Format

```
         0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
        |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 Word 0|UNUSED                                           |
        |-----------------------------------------------|
 Word 1|UNUSED                                           |
        |-----------------------------------------------|
 Word 2|# OF AVAILABLE ENTRIES                           |
        |-----------------------------------------------|
 Word 3|TABLE RELATIVE INDEX TO FIRST ENTRY              |
        |-----------------------------------------------|
```

CST General Entry Format

Case (i) CST Entry for a Present SL Segment or CSTX Segment

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
Word 0   |A |M |R |T |           SIZE/4                  | FIRMINFO
         |-----------------------------------------------|
Word 1   |/ |R |I |/ |/ |/ |S |C | ///////////////////// |
         |/ |O |M |/ |/ | /|Y |O |////////////////////// | FLAGS
         |/ |C |I |/ |/ |/ |S |R |////////////////////// |
         |/ |  |  |/ |/ |/ |  |E |////////////////////// |
         |-----------------------------------------------|
Word 2   |                  BANK                         | MMBANK
         |-----------------------------------------------|
Word 3   |                  BASE                         | MMBASE
         |-----------------------------------------------|
```

Case (ii) CST Entry for an Absent Segment SL or CSTX Segment

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
Word 0   |A |M |R |T |           SIZE/4                  | FIRMINFO
         |-----------------------------------------------|
Word 1   |/ |R |I |/ |/ |/ |S |C | ///////////////////// |
         |/ |O |M |/ |/ | /|Y |O |////////////////////// | FLAGS
         |/ |C |I |/ |/ |/ |S |R |////////////////////// |
```

```
        |/ |   |  |/ |/ |/ |   |E |////////////////////// |
        |-----------------------------------------------|
Word 2  |   L DEV #          |         HODA             | HODA
        |-----------------------------------------------|
Word 3  |                    LODA                       | LODA
        |-----------------------------------------------|
```

2.1.7 CST Entry Field Descriptions

A = 1 ==> segment absent
M = 1 ==> segment privileged
R = 1 ==> segemnt has been referenced
T = 1 ==> segment is being traced
DCV = 1 ==> disc copy is valid
STK = 1 ==> segment is a stack
MOD = 1 ==> a segment modification (exp., contr.) is pending
FWIP= 1 ==> a forced write of this segment is in progress
VMPAGECNT = # of virtual memory pages allocated to this segment
ROC = 1 ==> segment is recoverable overlay candidate
IMI = 1 ==> segment is in motion in
SYS = 1 ==> segment is a system segment
CORE= 1 ==> segment is core resident
WD= 1 ==> write disabled

TABLE FORMAT-CSTBLK

```
CSTBLK(0)-------------------------------------------
         0                                         *
         *   NUMBER OF ENTRIES IN TABLE            *
         ------------------------------------------
         1                                         *
         *   ANY UNASSIGNED ENTRY = -1             *
         ------------------------------------------
         2                                         *
         *   ANY ASSIGNED ENTRY > 0                *
         ------------------------------------------
         *                                         *
         *   REMAINING CSTBLK TABLE ENTRIES        *
         *                                         *
         ------------------------------------------
```

COMMENTS-

The table is initialized to minus one in each entry. When selected, the entry is replaced by a DST-relative index into the CST extension block.

2.1.8 Program Blocks and the CSTXMAP

Since programs can be dynamically loaded and unloaded, the segment table must be kept packed or fragmentation would occur Thus, the block of ST entries for a program segment begins at an ST entry number that changes if a program which was loaded before it gets unloaded. To manage this dynamic structure, an auxiliary structure, the CSTXMAP is used. A program is identified by its index, CSTXEIX, into this map. The program's current beginning physical ST entry number is equal to CSTXMAP (CSTXEIX).

```
ENTRY FORMAT-CST EXTENSION BLOCK
--------------------------------

CSTXMAP(CSTXEIX)-->-----------------------------
                0  * M = # OF CST'S IN BLOCK   *
                   -----------------------------
                1  * VALIDITY=%125252          *
                   -----------------------------
                2  * # OF USERS SHARING BLOCK  *
                   -----------------------------
                3  *            0              *
                   -----------------------------
%301 ------------> * HAS CST ENTRY FORMAT      *
                   -----------------------------
%302 ------------> * HAS CST ENTRY FORMAT      *
                   -----------------------------
                                 .
                                 .
                   -----------------------------
%300+M ----------> * HAS CST ENTRY FORMAT      *
                   -----------------------------
```

COMMENT
The value of CSTXEIX is established when a CST extension block is allocated. This index into the array CSTXMAP is maintained in the PCB of each process sharing the block.

2.1.9 Fixed DST Entry Assignments

OCTAL		DECIMAL	TABLE NAME
0		0	
1	CST	1	CST
2	DST	2	DST
3	PCB	3	PCB
4	CSTX	4	CSTX
5	SYSTEM GLOBAL AREA	5	SYS
6	CORE	6	CORE
7	ICS	7	ICS
10	SYSTEM BUFFERS	8	SBUF
11	UCOP REQUEST QUEUE	9	UCRQ
12	PROCESS-PROCESS COMMUNICATION TABLE	10	PPCOM
13	I/O QUEUE	11	IOQ
14	TERMINAL BUFFERS	12	TBUF
15	LOGICAL-PHYSICAL DEVICE TABLE	13	LPDT
16	LOGICAL DEVICE AND CLASS TABLE	14	LDT
17	DRIVER LINKAGE TABLE	15	DLT
20	I/O RESOURCE TABLES	16	BUSY, HEAD, TAIL
21	SECONDARY MSG TABLE	17	SECMSGTAB
22	LOADER SEGMENT TABLE	18	LST
23	TIMER REQUEST LIST	19	TRL
24	DIRECTORY	20	DDS

DST ALLOCATION (CONT.)

25	DIRECTORY SPACE	21	
26	RIN TABLE	22	RIN
27	SWAPTABLE	23	SWAPTAB
30	JOB PROCESS COUNT	24	JPCNT
31	JOB MASTER TABLE	25	JMAT
32	TAPE LABEL TABLE	26	VDD
33	LOG TABLE	27	LOGTAB
34	REPLY INFORMATION TABLE	28	RIT
35	VOLUME TABLE	29	VTAB
36	BREAKPOINT TABLE	30	STOP
37	LOG BUFFER1	31	
40	LOG BUFFER2	32	
41	LOG ID TABLE	33	LIDTAB
42	ASSOCIATE TABLE	34	
43	CST BLOCK	35	CSTBLK
44	JOB CUTOFF TABLE	36	JCUT
45	SYSTEM JIT	37	SJIT
46	SPECIAL REQ TABLE	38	SRTTAB
47	VIRTUAL DISC SPACE MANAGEMENT TABLE	39	VDSMTAB
50	//////////////////////	40	
51	ARSBM TABLE	41	ARSBMTAB

DST ALLOCATION (CONT.)

```
      |----------------------|
  52  |  ILT                 |  42           ILT
      |----------------------|
  53  | SIR TABLE            |  43           SIR
      |----------------------|
  54  | FMAVT                |  44           FMAVT
      |----------------------|
  55  | INPUT DEVICE DIRECT  |  45           IDD
      |----------------------|
  56  | OUTPUT DEVICE DIRECT |  46           ODD
      |----------------------|
  57  | WELCOME MESSAGE #1   |  47           LOGONDSTN1
      |----------------------|
  60  | WELCOME MESSAGE #2   |  48           LOGONDSTN2
      |----------------------|
  61  | CS DATA SEGMENT      |  49           CSTAB
      |----------------------|
  62  | PROCESS-JOB          |  50           PJXREF
      | CROSS REFERENCE      |
      |----------------------|
  63  | SYSTEM JDT           |  51           SYSJDT
      |----------------------|
  64  |  COMMAND LOGON DST   |  52           CILOGDST
      |----------------------|
  65  |MOUNTED VOL. SET TABLE|  53           MVTAB
      |----------------------|
  66  | PRI.VOL. USER TABLE  |  54           PVUSER
      ------------------------
  67  | AVAILABLE REGION LIST|  55           ARLDTAB
      |----------------------|
  70  | DISC REQUEST TABLE   |  56           DISCREQTAB
      |----------------------|
  71  |  MSG HARBOR TABLE    |  57           MSGHARBTAB
      |----------------------|
  72  |PRIMARY MESSAGE TABLE |  58           PRIMMSGTAB
      |----------------------|
  73  |MEASUREMENT INFO TABLE|  59           MEASINFOTAB
      |----------------------|
-------------------------------------------------                           !
  74  |  FIRST FREE DST      |  60                                          !
      |----------------------|                                              !
```

2.2 Swap Tables

2.2.1 SWAPTAB

The Swaptab is a core resident memory management table used to ep keep track of the locality lists of the competing processes.

SWAPTAB DST# = 23 (%27)

%1025 Sysbase relative index of SWAPTAB entry 0.

SWAPTAB ENTRY 0 FORMAT

```
            0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
           |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
SWAPTAB00|            # ENTRIES CONFIGURED                  |
           |------------------------------------------------|
SWAPTAB01|                ENTRY SIZE (5)                    |
           |------------------------------------------------|
SWAPTAB02|                # FREE ENTRIES                    |
           |------------------------------------------------|
SWAPTAB03|   TABLE RELATIVE INDEX OF FIRST FREE ENTRY       |
           |------------------------------------------------|
SWAPTAB04|                      0                           |
           |------------------------------------------------|
```

SWAPTAB UNASSIGNED ENTRY FORMAT

```
            0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
           |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
SWAPTAB00|              %100000                             |
           |------------------------------------------------|
SWAPTAB01|   TABLE RELATIVE INDEX OF NEXT FREE ENTRY        |
           |------------------------------------------------|
SWAPTAB02|                    0                             |
           |------------------------------------------------|
SWAPTAB03|                    0                             |
           |------------------------------------------------|
SWAPTAB04|                    0                             |
           |------------------------------------------------|
```

An assigned entry in the swaptab is a process' SLL header or a member of a process' SLL. These formats are now described.

2.2.2 Segment Locality Lists (SLL)

The system maintains for each process a segment locality list (SLL) of the segments belonging to that process' current working set. The process' SLL consists of a header and a list of entries. The header and list entries are taken from the SWAPTAB.

A process' SLL is located via the process' pcbentry. PCB01 contains the sysbase relative index of the process' SLL header.

```
                     SWAPTAB

         |--------------------------------|
         |                .               |
         |                .               |
         |                .               |
         |--------------------------------|
PCB01--> |         SLLHEADER              |
      +--|                                |
      |  |--------------------------------|
      |  |                .               |
      |  |                .               |
      |  |                .               |
      |  |--------------------------------|
      +->|       FIRST SLL ENTRY          |
      +--|                                |
      |  |--------------------------------|
      |  |                .               |
      |  |                .               |
      |  |                .               |
      |  |--------------------------------|
      +->|        NEXT SLL ENTRY          |
      +--|                                |
      |  |--------------------------------|
      |  |                .               |
      |  |                .               |
      v  |                .               |
         |--------------------------------|
```

SLL HEADER FORMAT

```
           0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
          |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
          |S |S |H |I |P |S |     |                       |
          |W |W |A |N |A |T |     |                       |
SLLHEAD00|I |R |S |T |R |R |     |  IOCNT                |SLL
          |P |E |M |L |T |T |     |                       |SCHEDTOIOMSG
          |  |Q |E |O |I |O |     |                       |
          |  |  |M |C |N |V |     |                       |
          |-----------------------------------------------|
SLLHEAD01| SYSBASE RELATIVE INDEX OF FIRST ENTRY IN LIST |SLLFIRSTINX
          |-----------------------------------------------|
SLLHEAD02|   WORD NOT CURRENTLY USED                     |
          |-----------------------------------------------|
SLLHEAD03| SYSBASE RELATIVE INDEX OF MEMORY REQUEST ENTRY|SLLMEMREQINX
          |-----------------------------------------------|
SLLHEAD04|          # ENTRIES IN PROCESS' SLL            |SLLCOUNT
          |-----------------------------------------------|

SLLHEAD00 .(0:1) SWIP, Swap In Progress Flag
          .(1:1) SWREQ, Swap Required Flag
          .(2:1) HASMEM, Has Memory Flag
          .(3:1) INTLOC, Initialize locality list
          .(4:1) PARTIN, Process partially swapped in
          .(5:1) STRTOV, Start swap over flag
          .(6:2) Available
          .(8:8) IOCNT, Segment read completions until awake
```

SLL ENTRY FORMAT

```
           0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
          |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
SLLENTRY00|   PMPQPIN           |  NMPQPIN              |SLLMPQLINK
          |-----------------------------------------------|
SLLENTRY01| SYSBASE RELATIVE INDEX OF NEXT ENTRY IN LIST |SLLNEXTINX
          |-----------------------------------------------|
SLLENTRY02| SYSBASE RELATIVE INDEX OF PREV ENTRY IN LIST |SLLPREVINX
          |-----------------------------------------------|
SLLENTRY03|                 SEGIDENTIFIER                 |SLLSEGIDENT
          |-----------------------------------------------|
SLLENTRY04|/ |S |/ |/ |/ |/ |/ |T |F |L |S|D|//////////// |SLLFLAGS
          |/ |T |/ |/ |/ |/ |/ |O |Z |K |L|I|//////////// |
          |/ |K |/ |/ |/ |/ |/ |S |R |R |L|S|//////////// |
          |/ |  |/ |/ |/ |/ |/ |S |E |E |I|C|//////////// |
          |/ |  |/ |/ |/ |/ |/ |  |Q |Q |M|I|//////////// |
          |/ |  |/ |/ |/ |/ |/ |  |  |  |I|O|//////////// |
          |-----------------------------------------------|
```

```
SLLENTRY00 .(0:8)  PMPQPIN, previous make present deferred queue pin
           .(8:8)  NMPQPIN, next make present deferred queue pin

SLLENTRY01 .(0:16) SYSBASE, relative index of next entry in list (=0=>
                   last entry)

SLLENTRY02 .(0:16) SYSBASE relative index of previous entry in list
                           (=0==> first entry)
SLLENTRY03 Has standard segment identifier format.
SLLENTRY04 .(1:1)  STK ==> process' stack entry
           .(7:1)  TOSS ==> Toss this entry
           .(8:1)  FRZREQ ==> Process requests a freeze on seg
           .(9:1)  LKREQ ==> Process requests a lock on seg
           .(10:1) SLLIMI ==> process is queued for this segment
           .(11:1) DISIOSEG ==> process waiting for disc i/o against
                                this seg
```

SPECIAL REQUEST TABLE

(USED FOR PASSING DATA SEGMENT SIZE CHANGE INFO AND FOR KEEPING A LIST OF DEVICES WAITING FOR A SEGMENT TO ARRIVE IN MEMORY.)

```
               -------------------------------------
 ENTRY 0    0|    # entries in table               |
             |-------------------------------------|
            1|      entry size (5)                 |
             |-------------------------------------|
            2|      # available entries            |
             |-------------------------------------|
            3|     first available entry           |
             |-------------------------------------|
            4|     last available entry            |
             |=====================================|
             |                                     |
             ~                                     ~
             ~                                     ~
             |                                     |
             |=====================================|
first---->0|      next assigned entry            |
assigned     |-------------------------------------|
entry       1|        segidentifier                |
(pointed     |-------------------------------------|
to by       2|          new data seg size          |
%1043)       |-------------------------------------|
            3|      read displacement              |
             |-------------------------------------|
            4|         move count                  |
             |=====================================|
             |                                     |
             |                                     |
             ~                                     ~
```

2.3 Main Memory Region Headers and Trailers

Main memory is partitioned into regions. Each region is in one of three states: available, reserved, or assigned.

An available region is available for consumption by the free space allocation mechanism. An available region consists of neighboring subregions, each of which is either a hole or an overlay candidate. An available region is linked into the available region list of appropriate size.

A reserved region is a main memory region which is in the transition state from available to assigned. A reserved region has been cleaned, and there is a pending disc read of a segment into the region.

Assigned regions are occupied by present segments. Available and reserved regions consist of one or more adjacent subregions. Region headers and trailers are partitioned into global and local components. The global region header/trailer is only valid for the first/last subregion in regions consisting of more than one subregion.

The region headers and trailers of available, reserved, and assigned regions contain the state and control information pertaining to the current or planned contents of the region.

2.3.1 Available Region Headers and Trailers

Available Region Global Header Format
(only valid for first subregion)

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
RB-19    |A |R |A |C |R |R |R |R |/////////////////////|R | RAS
         |S |E |V |L |E |E |E |E |/////////////////////|E |
         |S |S |  |N |S |S |S |S |/////////////////////|S |
         |  |  |  |D |  |  |  |  |/////////////////////|  |
         |-----------------------------------------------|
RB-18    |     REGION SIZE (IN MAIN MEMORY PAGES)        | RS
         |-----------------------------------------------|
RB-17    |                  RESERVED                     |
         |-----------------------------------------------|
RB-16    |                  RESERVED                     |
         |-----------------------------------------------|
RB-15    |  REGION BASE OF PREVIOUS IN THIS AVAILABLE    | PLINK
RB-14    |  REGION LIST                                  |
         |-----------------------------------------------|
RB-13    |  REGION BASE OF NEXT IN THIS AVAILABLE        | NLINK
RB-12    |  REGION LIST                                  |
         |-----------------------------------------------|
RB-11    |                  RESERVED                     |
         |-----------------------------------------------|
```

Available Region Subregion Header
(Valid for All Subregions)

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
RB-10    |    SUBREGION SIZE (IN MAIN MEMORY PAGES)      | SS
         |-----------------------------------------------|
RB-9     |V | SUBREGION DISPLACEMENT (IN MAIN MEM PAGES) | SD
         |-----------------------------------------------|
RB-8     |          WRITE REQUEST POINTER                | WREQP
         |-----------------------------------------------|
RB-7     |          SEGMENT IDENTIFIER                   | SEGIDET
         |-----------------------------------------------|
RB-6     |                  RESERVED                     |
         |-----------------------------------------------|
RB-5     |                  RESERVED                     |
         |-----------------------------------------------|
RB-4     |      LDEV #           |          HODA         | HODA
         |-----------------------------------------------|
RB-3     |          LOW ORDER DISC ADDRESS               | LODA
         |-----------------------------------------------|
RB-2     |///////////////////////////////////////////////|
         |-----------------------------------------------|
RB-1     |///////////////////////////////////////////////|
         |-----------------------------------------------|
```

RB ===>

Available Region Subregion Trailer

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|///////////////////////////////////////////////|
|-----------------------------------------------|
|                SUBREGION SIZE                 | TSS
|-----------------------------------------------|
```

Available Region Global Trailer
(Valid Only for Last Subregion)

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|A |R |A |///////////////////////////////////////| TRAS
|S |E |V |///////////////////////////////////////|
|S |S |  |///////////////////////////////////////|
|-----------------------------------------------|
|                   REGION SIZE                 | TRS
|-----------------------------------------------|
```

2.3.2 Reserved Region Headers and Trailers

Reserved Region Global Header Format
(Only Valid for First Subregion)

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
         |A |R |A |C |S |L |F |I |/////////////////////|M |
         |S |E |V |L |C |K |Z |O |/////////////////////|I |
RB-19    |S |S |  |N |  |D |N |F |/////////////////////|P | RAS
         |  |  |  |D |  |  |  |Z |/////////////////////|  |
         |-----------------------------------------------|
RB-18    |  REGION SIZE (IN MAIN MEMORY PAGES)           | RS
         |-----------------------------------------------|
RB-17    |  ON-GOING I/O COUNT                           | IOCNT
         |-----------------------------------------------|
         |M |E |O |Q |I |E |G |M |R|///////////////////|M | INITMSG
         |S |X |N |S |N |X |A |S |E|///////////////////|S |
RB-16    |G |T |G |E |C |P |R |G |L|///////////////////|G |
         |P |D |I |G |M |R |B |A |R|///////////////////|V |
         |R |I |O |R |V |R |C |B |E|///////////////////|A |
         |O |S |D |E |  |E |O |O |S|///////////////////|L |
         |C |  |I |A |  |Q |L |R |P|///////////////////|I |
         |  |  |S |  |  |  |L |T |G|///////////////////|D |
         -------------------------------------------------
RB-15    |     INITIATION MESSAGE INFORMATION            | INITINFO
         |-----------------------------------------------|
         |M |M |B |S |I |M |/ |/ |///////////////////////| COMPMSG
         |S |O |K |C |O |S |/ |/ |                       |
RB-14    |G |D |D |H |W |G |/ |/ |                       |
         |P |R |L |E |A |A |/ |/ |                       |
         |R |E |K |D |I |B |/ |/ |                       |
         |O |Q |  |M |T |O |/ |/ |                       |
         |C |  |  |S |  |R |/ |/ |                       |
         |  |  |  |G |  |T |/ |/ |                       |
         -------------------------------------------------
RB-13    | PIN OF FIRST PROCESS | PIN OF LAST PROCESS    | MPQLINK
         |-----------------------------------------------|
RB-12    |              RELEASE PAGE COUNT               | PAGECNT
         |-----------------------------------------------|
RB-11    |          SPECIAL REQUEST TABLE POINTER        | SPECREQTABPTR
         |-----------------------------------------------|
```

Reserved Region Subregion Header
  (Valid for all Subregions)

```
           0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
          |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
RB-10     |     SUBREGION SIZE (IN MAIN MEMORY PAGES)     | SS
          |-----------------------------------------------|
          |C |                                            | SD
          |N |                                            |
RB-9      |T |                                            |
          |V |    # PAGES THIS SUBREGION IS DISPLACED     |
          |A |        FROM THE REGION BASE                |
          |L |                                            |
          |I |                                            |
          |D |                                            |
          |-----------------------------------------------|
RB-8      |           WRITE REQUEST TO POINTER            | WREQP
          |-----------------------------------------------|
RB-7      |             SUBSEGMENT IDENTIFIER             | SEGIDENT
          |-----------------------------------------------|
RB-6      |  FREEZE COUNT          |      LOCK COUNT      | LKFZCNTRS
          |-----------------------------------------------|
RB-5      | WRITE DISABLED COUNT |   I/O FROZEN COUNT     | WDIOFZCNT
          |-----------------------------------------------|
RB-4      |        LDEV #          |   HIGH ORDER DA      | HODA
          |-----------------------------------------------|
RB-3      |            LOW ORDER DISC ADDRESS             | LODA
          |-----------------------------------------------|
RB-2      |///////////////////////////////////////////////|
          |-----------------------------------------------|
RB-1      |///////////////////////////////////////////////|
          |-----------------------------------------------|
```

RB ==> First Word of Segment

Reserved Region Subregion Trailer
(Valid for All Subregions)

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
 |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 |///////////////////////////////////////////////|
 |-----------------------------------------------|
 |     SUBREGION SIZE (IN MAIN MEMORY PAGES)     | TSS
 |-----------------------------------------------|
```

Reserved Region Global Trailer

(Valid Only for Last Subregion)

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|A |R |A |//////////////////////////////////////|
|S |E |V |//////////////////////////////////////| TRAS
|S |S |  |//////////////////////////////////////|
|-----------------------------------------------|
|      REGION SIZE (IN MAIN MEMORY PAGES)       | TRS
|-----------------------------------------------|
```

2.3.3 Assigned Region Headers and trailers

Assigned Region Global Header Format

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
RB-19    |A |R |A |C |S |L |F |I |/////////////////////|M | RAS
         |S |E |V |L |C |K |Z |O |/////////////////////|I |
         |S |S |  |N |  |P |N |F |/////////////////////|P |
         |  |  |  |D |  |  |  |Z |/////////////////////|  |
         |  |  |  |  |  |  |  |N |/////////////////////|  |
         |-----------------------------------------------|
RB-18    |     REGION SIZE (IN MAIN MEMORY PAGES)        | RS
         |-----------------------------------------------|
RB-17    |                  RESERVED                     |
         |-----------------------------------------------|
RB-16    |                  RESERVED                     |
         |-----------------------------------------------|
RB-15    |                  RESERVED                     |
         |-----------------------------------------------|
RB-14    |                  RESERVED                     |
         |-----------------------------------------------|
RB-13    |                  RESERVED                     |
         |-----------------------------------------------|
RB-12    |                  RESERVED                     |
         |-----------------------------------------------|
RB-11    |                  RESERVED                     |
         |-----------------------------------------------|
```

Assigned Region Subregion Header

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
RB-10    |       SUB-REGION SIZE                         | SS
         |-----------------------------------------------|
RB-9     |                  RESERVED                     |
         |-----------------------------------------------|
RB-8     |                  RESERVED                     |
         |-----------------------------------------------|
RB-7     |           SEGMENT IDENTIFIER                  | SEGIDENT
         |-----------------------------------------------|
RB-6     |  FREEZE COUNT          |     LOCK COUNT       | LKFZCNTRS
         |-----------------------------------------------|
RB-5     | WRITE DISABLED COUNT   |   I/O FROZEN COUNT   | WDIOFZCNT
         |-----------------------------------------------|
RB-4     |       LDEV#            |         HODA         | HODA
         |-----------------------------------------------|
RB-3     |          LOW ORDER DISC ADDRESS               |
         |-----------------------------------------------|
RB-2     |///////////////////////////////////////////////|
         |-----------------------------------------------|
RB-1     |///////////////////////////////////////////////|
         |-----------------------------------------------|
```

RB==>

Assigned Region Subregion Trailer Format

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|///////////////////////////////////////////////|
|-----------------------------------------------|
|                SUBREGION SIZE                 | TSS
|-----------------------------------------------|

  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|A |R |A |//////////////////////////////////////| TSS
|S |E |V |//////////////////////////////////////|
|S |S |  |//////////////////////////////////////|
|-----------------------------------------------|
|       REGION SIZE (IN MAIN MEMORY PAGES)      | TRS
|-----------------------------------------------|
```

2.3.4 Region Header and Trailer Field Descriptions

RAS, Region Assignment State
.(0:1) Region Assigned Flag
.(1:1) Region Reserved Flag
.(2:1) Region Available Flag
.(3:1) Region Cleaned Flag
.(4:1) Size Change Pending Flag
.(5:1) Region Locked Flag
.(6:1) Region Frozen Flag
.(7:1) Region I/O Frozen Flag
.(8:7) Available
.(15:1) Blocked Lock Migration in Progress Flag
IOCNT, On-Going I/O Count
= # of on-going I/O's in the region which must complete before the initiation message can be processed.

INITMSG, Initiation Message
.(0:1) Message Processed Toggle Switch
.(1:1) Message Externally Disabled Flag
.(2:1) Message On-going I/O Disabled Flag
.(3:1) Queue Segment Read Disc Request Flag
.(4:1) Incore Move Request Flag
.(5:1) Expansion Request Flag
.(6:1) Garbage Collection Flag
.(7:1) Message Aborted Flag
.(8:1) Release Residual Pages Flag
.(9:6) Available
.(15:1) Message Valid Flag
INITINFO, Initaition Message Auxiliary Information
= Sysbase relative index of segment read disc request if INITMSG, QREADREQ=1
or
= +/- Displacement to initiation message for moves and expansions.
COMPMSG, Completion Message

.(0:1) Message Processed Toggle Switch
.(1:1) Segment Modification Required
.(2:1) Block Lock Request
.(3:1) Send Scheduler A Message
.(4:1) Awaken A Device
.(5:1) Message Aborted
.(6:2) Available

MPQLINK,	Make Present Deferred Queue Link
	.(0:8) PIN Of First Process Waiting for this Segment
	.(8:8) PIN of Last Process Waiting for this Segment
PAGECNT,	Release Page Count
	=# of extra pages to release before processing initiation message.
SPECREQTABPTR,	points into special request table to the list of devices queried on this segment.
SS,	Subregion Size
SD,	Subregion Displacement
	.(0:1) Displacement Count Valid Flag
	.(1:15) # Pages to Base of Region
WREQP,	Write Request Pointer
	= Sysbase Relative Index of Disc Write Request when the Data Segment in the Subregion is in Motion Out
SEGIDENT,	Segment Identifier- has standard segment identifier format

Space Allocation Structures

Available regions in main memory are kept track of by multiple free lists. All available regions of the same size are linked into the same available region list (ARL). A bitmap is maintained to indicate which lists are non-empty (ARSBM). A sysglob cell is maintained which contains the size of the largest currently available region. %1045 MAXAVAILREG, number of pages in largest currently available region.

Available Region List (ARL)

%1044 SYSBASE index of base of ARL

ARL DST # = 55 (%67)

```
            0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
           |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
ARLD(0)    |                       0                        |
           |------------------------------------------------|
           |                       0                        |
           |------------------------------------------------|
ARLD(1)    |  BANK  OF FIRST AVAIL REGION OF SIZE = 1 PAGE  |
           |------------------------------------------------|
           |  BASE OF FIRST AVAIL REGION OF SIZE = 1 PAGE   |
           |------------------------------------------------|
ARLD(2)    |  BANK OF FIRST AVAIL REGION OF SIZE = 2 PAGES  |
           |------------------------------------------------|
           |  BASE OF FIRST AVAIL REGION OF SIZE = 2 PAGES  |
           |------------------------------------------------|
           .
           .
           .
           .
           |------------------------------------------------|
ARLD(N)    |  BANK OF FIRST AVAIL REGION OF SIZE = N PAGES  |
           |------------------------------------------------|
           |  BASE OF FIRST AVAIL REGION OF SIZE = N PAGES  |
           |------------------------------------------------|

           Where N = maximum available region size
                   = (2**16/2**pagepower) pages
```

Available Region Size Bit Map (ARSBM)

%1004 SYSBASE index of base of ARSBM

ARSBM DST# = 41 (%51)

```
           0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
          |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
ARSBM(0)  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
          |-----------------------------------------------|
          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
          |-----------------------------------------------|
          .
          .
          .
          |-----------------------------------------------|
ARSBM(M)  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
          |-----------------------------------------------|
```

M = (# of available region sizes/16) +1

ARSBM (J) . (K:1) = 1 ==> the available region list of
size J*16+K Pages is non-empty.

CHAPTER 3 DISC LAYOUT

SYSTEM DISC LAYOUT

```
SECTOR #                                                   SECTOR #
          +---------------------------------------------+
     % 0|                 DISC LABEL                   |0
         |---------------------------------------------|
       1|           Defective Tracks Table or          |1
         |           Defective Sector Table            |
         |---------------------------------------------|
       2| Cold Load Channel Program for /3X, /4X, /6X |2
         | and for discs on Series III HPIB adapter    |
         |---------------------------------------------|
       3| Mem Dump Channel Program for /3X, /4X, /6X  |3
         |---------------------------------------------|
       4|          Reserved Area Bit Map               |4  \
         |---------------------------------------------|    |
       5|                                              |5   |
         |---------------          ------------------|    |
       6|                                              |6   |
         |------------    CODE FOR    ---------------|    |
       7|             INITIAL PROGRAMS                 |    |
         |------------    "BOOTSTRAP"   -------------|    |
      10|                 SEGMENT                      |    |
         |--------------           ------------------|    |
      11|                                              |    |
         |----------------         ------------------|    > Variable
        .|                                             |    | Length
         |---------------------------------------------|    |
        .|                                             |    |
         |---------------------------------------------|    |
        .|                                             |    |
         |---------------------------------------------|    |
         |                                             |    |
         |---------------------------------------------|    |
         |                                             |    /
         |---------------------------------------------|
         | LOW CORE (CST POINTER, QI, ZI, POINTER)     |  <--| Follows
         |---------------------------------------------|     | immediately
         |    TEMPORARY CST (INITIAL PROGRAM)          |     | after
         |---------------------------------------------|     | Bootstrap
         |              Initial's ININ                 |     | Segment
         |---------------------------------------------|
         |              BOOTSTRAP STACK                |
         |---------------------------------------------|
         |   remainder of SIO cold load program or     |
         |         cold load channel program           |
         |---------------------------------------------|
```

SYSTEM DISC LAYOUT (CONT.)

```
SECTOR # |------------------------------------------|SECTOR #
   %     |                                          |
         |------------------------------------------|
         |                                          |
         |------------------------------------------| .
        .|                                          | .
         |------------------------------------------| .
       34|    DISC COLD LOAD INFORMATION TABLE      |28
         |------------------------------------------|
       35|    DISC COLD LOAD INFORMATION TABLE      |29
         |------------------------------------------|
         |                                          |                       !
         ~                                          ~                       !
```

SYSTEM DISC LAYOUT (CONT.)

```
           ~                                              ~
SYSDB     |----------------------------------------------|
----->    |                                              |  ---> Note: Initial
%130/131  |                                              |       tries to allocate
          |              SYSTEM DIRECTORY                |       directly after
          |                                              |       the Free Space
          |----------------------------------------------|       Map.  However,
          |                                              |       this may vary
          |                                              |       depending on
          |                                              |       deleted or
          |                                              |       reassigned tracks
          |             VIRTUAL MEMORY AREA              |
          |                                              |
          |                                              |
          |                                              |
          |----------------------------------------------|
          |          INITIAL PROGRAM SEGMENTS            |
          |           (EXCEPT BOOTSTRAP SEG)             |
          |----------------------------------------------|
          |                SYSTEM FILES                  |
          |           (FROM COLD LOAD TAPE)              |
          |----------------------------------------------|
          |                                              |
          |            SYSTEM TABLES                     |
          |              * LPDT                          |
          |              * LDT                           |
          |              * LDTX                          |
          |              * VOLUME TABLE                  |
          |              * DEVICE CLASS TABLE            |
          |            INITIAL PROGRAM STACK             |
          |                                              |
          |----------------------------------------------|
          |                                              |
          |                                              |
          |                 USER FILES                   |
          |                     .                        |
          |                     .                        |
          |                     .                        |
          |                                              |
          |                                              |
          |                                              |
          |                                              |
          |                                              |
           ----------------------------------------------
```

DISC LABEL (Sector 0 of disc)

```
              SYSTEM VOLUME

        0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
       |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
      0|              CONTROL ORDER                    |0
       |                                               |
      1|              <<CYL/ARC #>>                    |1
       |-----------------------------------------------|   DISC BOOTSTRAP
      2|               READ ORDER                      |2   SIO PROGRAM
       |                                               |   (SYSTEM DISC
      3|             <<MEM ADDRESS>>                   |3   ONLY)
       |-----------------------------------------------|
      4|             SIO JUMP ORDER                    |4   Words 0-5 contain the
       |                                               |    Ascii string
      5|             <<MEM ADDRESS>>                   |5   "SYSTEM DISC" for
       |-----------------------------------------------|    /3X, /4X, /6X
      6|/////////////////|    DISK TYPE   |DISKSUBTYPE|6    (system disc only)
       |-----------------------------------------------|
      7|              COLD LOAD ID                     |7
       |-----------------------------------------------|
     10|        "3"          |          "0"            |8
       |-----------------------------------------------|
     11|        "0"          |          "0"            |9  If word (%11)            !
       |-----------------------------------------------|   contains a "1"           !
     12|                                               |10 a former system          !
       |                                               |   volume has been          !
     13|                                               |11 scratched.               !
       |              VOLUME NAME                      |
     14|                                               |12
       |                                               |
     15|                                               |13
       |-----------------------------------------------|
     16|                                               |
       .                                               .
       .                                               .
       .             UNUSED                            .
       .                                               .
       .                                               .
     24|                                               |
       |-----------------------------------------------|
     25|            CYL                                |   ICF WCS
       |-----------------------------------------------|    IMAGE
     26|   HEAD              |      SECTOR             |   POINTER
       |-----------------------------------------------|
     27|                                               |
       .                                               .
       .                                               .
       .            RESERVED                           .
       .                                               .
    122|                                               |                            !
```

```
   |-----------------------------------------------|
123|                    CYL                        |
   |-----------------------------------------------|
124|     HEAD                |      SECTOR         |
   |-----------------------------------------------|
   .                                               .
   .                                               .
   .                                               .
   .                                               .
   .                                               .
   |-----------------------------------------------|
170|                                               |120
   |-----------------------------------------------|
171|          Disc Free Space map OK flag          |121
   |-----------------------------------------------|
172| Disc Free Space map descriptor table checksum |122
   |-----------------------------------------------|
173|  Disc Free Space descriptor table dirty flag  |123
   |-----------------------------------------------|
174|                                               |124
   |-- Disc Free Space descriptor table address  --|
175|                                               |125
   |-----------------------------------------------|
176|                                               |126
   |------- Disc Free Space bitmap address  -------|
177|                                               |127
   |-----------------------------------------------|
```

DISC LABEL SECTOR 0 (CONT.)

SERIAL VOLUME

```
   |------------------------------------------------|
  0|                                                |0
   |             0  (:STORE)                        |
  1|                                                |1
   |                  or                            |
  2|                                                |2
   |      Cold-load SIO channel program (non-HPIB   |
  3|      machines only).  For HPIB machines, cold  |3
   |      load channel program is in sector 2 and   |
  4|      SOFTDUMP channel program is in sector 3.  |4
   |                                1  1  1  1  1  1|
  5| 0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5|5
   |------------------------------------------------|
  6|SC|MV|SR|         |  TYPE           | SUB-TYPE  |6   SC = 1 ==>
   |------------------------------------------------|       Scratch volume
  7|                                                |7   MV = 1 ==> Master
   |                                                |       Volume of PV set.
 10|                      0                         |8   SR = 1 ==>
   |                                                |       Serial disc
 11|                                                |9
   |------------------------------------------------|
 12|        "S"            |           "E"          |10 \
   |------------------------------------------------|    |
 13|        "R"            |           "D"          |11 | VOL NAME
   |------------------------------------------------|    |
 14|        "I"            |           "S"          |12 | "SERDISC "                     |
   |------------------------------------------------|    |
 15|        "C             |           " "          |13 /
   |------------------------------------------------|
 16| Words Per Sector                               |14 \                                 |
   |------------------------------------------------|    |                                 |
 17| Sectors Per Track (Cartridge tape = 1)         |15 |                                 |
   |------------------------------------------------|    |                                 |
 20| Sector Address of Beginning of Tape (BOT)      |16 |                                 |
   |------------------------------------------------|    | Serial                          |
 21| Double Address of                              |17 > Disc                            |
   |-                                              -|    | Info                            |
 22| End Of Tape (EOT)                              |18 |                                 |
   |------------------------------------------------|    |                                 |
 23| Double Address of                              |19 |                                 |
   |-                                              -|    |                                 |
 24| End Of Data (EOD)                              |20 /                                 |
   |------------------------------------------------|                                      |
 25|                    CYL                         |21  ICF WCS                          |
   |------------------------------------------------|      IMAGE                          |
 26|       HEAD               |       SECTOR        |22  POINTER                          |
   |------------------------------------------------|                                      |
```

Serial volume (continued)

```
   |-----------------------------------------------|
 27|                                               |23
   .                                               .
   .        RESERVED FOR FUTURE WCS                .
122|                                               |82
   |-----------------------------------------------|
123|                     CYL                       |83
   |-----------------------------------------------|
124|       HEAD               |      SECTOR        |84
   |-----------------------------------------------|
```

DISC LABEL (cont)

SECTOR 0

MASTER VOLUME

```
                   ------------------------------------------------
                  0|                                              |0
                  1|                                              |1
                  2|                     0                        |2
                  3|                                              |3
                  4|                                              |4
                  5|                                              |5
                   ------------------------------------------------
SC = SCRATCH      6|SC|MV|SR|      |6   TYPE   11|12  SUB-TYPE 15|6
     VOLUME        ------------------------------------------------
MV = MASTER       7|         GENERATION INDEX                     |7
   VOLUME = 1      ------------------------------------------------
SR = SERIAL      10|                     0                        |8
     VOLUME      11|                                              |9
                   ------------------------------------------------
                 12|                                              |10
                 13|                  VOLUME                      |11
                 14|                   NAME                       |12
                 15|                                              |13
                   ------------------------------------------------
                 16|               INITIAL DATE                   |14
                   ------------------------------------------------
                 17|                  DIRBASE                     |15  0 IF NOT
                   ------------------------------------------------    MASTER
                 20|                  DIRSIZE                     |16  VOLUME
                   ------------------------------------------------
                 21|                                              |17
                 22|                  ACCOUNT                     |18
                 23|                   NAME                       |19
                 24|                                              |20
                   ------------------------------------------------
                 25|                                              |21
                 26|                   GROUP                      |22
                 27|                    NAME                      |23
                 30|                                              |24
                   ------------------------------------------------
```

MASTER VOLUME (CONT.)

```
                    ---------------------------------------------
                 31|                                             |25
                 32|                 VOLUME SET                  |26
                 33|                    NAME                     |27   HEADER
                 34|                                             |28
                    ---------------------------------------------
  VS VTAB        35|                                             |29
  HEADER +          ---------------------------------------------
  8 ENTRIES      36|0 VCOUNT 3|          |          VMASK        |30
  COPIED FROM       ---------------------------------------------
  VSET DEFN      37|                                             |31
  IN SYSTEM      40|                   VOLUME                    |32
  DIRECTORY      41|                    NAME                     |33   VOLUME
                 42|                                             |34   ENTRY 0
                    ---------------------------------------------         .
                 43|                                             |35      .
                    ---------------------------------------------         .
                 44|  SUB-TYPE           |          VTABX        |36      .
                    ---------------------------------------------         .
                 45|                                             |37      .
                   |                     .                       |        .
                   ~                     .                       ~        .
                   ~                     .                       ~     VOLUME
                   |                     .                       |     ENTRY
                116|                                             |78     7
                    ---------------------------------------------
                   .                                             .
                   .                                             .
                   .                                             .
                   .                                             .
                   .                                             .
                   |---------------------------------------------|
                170|                                             |120
                   |---------------------------------------------|
                171|        Disc Free Space map OK flag          |121
                   |---------------------------------------------|
                172|  Disc Free Space descriptor table checksum  |122
                   |---------------------------------------------|
                173| Disc Free Space descriptor table dirty flag |123
                   |---------------------------------------------|
                174|                                             |124
                   |-- Disc Free Space descriptor table address -|
                175|                                             |125
                   |---------------------------------------------|
                176|                                             |126
                   |------- Disc Free Space bitmap address ------|
                177|                                             |127
                   |---------------------------------------------|
```

DISC LABEL (cont)

SECTOR 0

SLAVE VOLUME

```
                  -----------------------------------------------
                 0|                                             |0
                 1|                                             |1
                 2|                    0                        |2
                 3|                                             |3
                 4|                                             |
SC = SCRATCH     5|                                             |
     VOLUME       -----------------------------------------------
MV = MASTER      6|SC|MV|SR|         |6 TYPE 11|12 SUB-TYPE  15|6
  VOLUME = 0      -----------------------------------------------
SR = SERIAL      7|              GENERATION INDEX               |7
     VOLUME       -----------------------------------------------
                10|                    0                        |8
                11|                                             |9
                  -----------------------------------------------
                12|                                             |10
                13|                 VOLUME                      |11
                14|                  NAME                       |12
                15|                                             |13
                  -----------------------------------------------
                16|               INITIAL DATE                  |14
                  -----------------------------------------------
                17|                    0                        |15
                20|                                             |16
                  -----------------------------------------------
                21|                                             |17
                22|               ACCOUNT                       |18
                23|                 NAME                        |19
                24|                                             |20
                  -----------------------------------------------
                25|                                             |21
                26|                 GROUP                       |22
                27|                  NAME                       |23
                30|                                             |24
                  -----------------------------------------------
                31|                                             |25
                32|              VOLUME SET                     |26
                33|                 NAME                        |27
                34|                                             |28
                  -----------------------------------------------
                  .                                             .
                  .                                             .
                  .                                             .
                  .                                             .
```

```
   .                                             .
   |---------------------------------------------|
170|                                             |120
   |---------------------------------------------|
171|         Disc Free Space map OK flag         |121
   |---------------------------------------------|
172|  Disc Free Space descriptor table checksum  |122
   |---------------------------------------------|
173| Disc Free Space descriptor table dirty flag |123
   |---------------------------------------------|
174|                                             |124
   |-- Disc Free Space descriptor table address -|
175|                                             |125
   |---------------------------------------------|
176|                                             |126
   |------- Disc Free Space bitmap address ------|
177|                                             |127
   |---------------------------------------------|
```

DEFECTIVE TRACKS TABLE (DTT -- Sector 1 of Disc)

(the DTT exists on device type 0, 1, & 2 discs)

```
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
    0|       # OF DEFECTIVE TRACK ENTRIES (N)        |0
     |-----------------------------------------------|
    1|          DEFECTIVE TRACK NUMBER        | DTC  |1     120 DEFECTIVE
     |-----------------------------------------------|      TRACKS MAXIMUM
    2|          DEFECTIVE TRACK NUMBER        | DTC  |2
     |-----------------------------------------------|
    3|          DEFECTIVE TRACK NUMBER        | DTC  |3
     |-----------------------------------------------|
    4|          DEFECTIVE TRACK NUMBER        | DTC  |4
     |-----------------------------------------------|
    5|                       .                       |5
     |                                               |
    6|                       .                       |6
     |                                               |
    7|                       .                       |7
     |                                               |
   10|                       .                       |8
     |                                               |
   11|                       .                       |9
     |                                               |
   12|                       .                       |10
     |                                               |
    .|                       .                       |.
    .|                                               |.
    .|                       .                       |.
     |                                               |
     |                       .                       |
     ~                                               ~
                             .
     ~                                               ~
     |                       .                       |
     |                                               |
     |                       .                       |
     |                                               |
     |                       .                       |
     |-----------------------------------------------|
  165|          DEFECTIVE TRACK NUMBER        | DTC  |117
     |-----------------------------------------------|
  166|          DEFECTIVE TRACK NUMBER        | DTC  |118
     |-----------------------------------------------|
  167|          DEFECTIVE TRACK NUMBER        | DTC  |119
     |-----------------------------------------------|
```

DEFECTIVE TRACKS TABLE (CONT.)

```
   |-----------------------------------------------|
170|           DEFECTIVE TRACK NUMBER       | DTC |120
   |-----------------------------------------------|
171|                                               |121
   |                                               |
172|                                               |122
   |                 RESERVED FOR                  |
173|                  FUTURE USE                   |123
   |                                               |
174|                                               |124
   |                                               |
175|                                               |125
   |-----------------------------------------------|
176|       NEXT AVAILABLE ALTERNATE TRACK          |126
   |-----------------------------------------------|
177|     LOGICAL DISC PACK SIZE (CYLINDERS)        |127
   |-----------------------------------------------|
              OR # OF TRACKS IF FH DISC
```

DTC	(DEFECTIVE TRACK CODE)
0	suspect
1	suspect alternate
2	deleted
3	reassigned

NOTE: The situation where there are two entries for the same track, n, one having a DTC of 0 (suspect) and the other having a DTC 3 (reassigned) results from a situation where the disc driver could not "read" (unreadable) the address of the particular track.

```
        DEFECTIVE SECTOR TABLE (DSCT -- sector 1 of disc)
        -------------------------------------------------
         (the DSCT exists on device type 3 (CS/80) discs)


         0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
        +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
     0  |       number of entries in the table          | 0
        |-----------------------------------------------|
    %1  |         index to the first entry (6)          | 1
        |-----------------------------------------------|
    %2  |               entry size (2)                  | 2
        |-----------------------------------------------|
    %3  |       maximum number of entries (61)          | 3
        |-----------------------------------------------|
    %4  |                 0 (reserved)                  | 4
        |-----------------------------------------------|
    %5  |                 0 (reserved)                  | 5
        |-----------------------------------------------|
    %6  |        first defective sector entry           | 6
        |     (double-word logical sector address)      |
        |-----------------------------------------------|
   %10  |               second entry                    | 8
        |                                               |
        |-----------------------------------------------|
   %12  |               third entry                     | 10
        |                                               |
        |-----------------------------------------------|
        |                                               |
        ~                                               ~
                              .
                              .
        ~                     .                         ~
        |                                               |
        |-----------------------------------------------|
  %176  |        maximum defective sector entry         | 126
  %177  |                                               | 127
        +-----------------------------------------------+
```

Unlike the DTT, entries in the DSCT are not permanent. Once a suspect sector is handled by INITIAL or VINIT, its entry is removed from the table. Thus this table contains only unprocessed suspect sectors.

RESERVED AREA BIT MAP (sector 4 of the system disc)

The first 400 sectors of the system disc are reserved for Initial's use. This area contains permanent data structures for the boot. It is also used as a temporary storage area for data during sparing. All other system volumes and private volumes reserve only the first 10 sectors of the disc. They do not have a reserved area bit map.

The bit map contains 1 bit per sector. A '1' means the sector is free.

```
     +-----------------------------------------+
 %0  |                                         |  0
     |                                         |
     |             reserved area               |
     |               bit map                   |
     |                                         |
     |                                         |
     ~                                         ~
                       .
                       .
                       .
     ~                                         ~
     |                                         |
 %30 |                                         |  24
     |-----------------------------------------|
 %31 |                                         |  25
     |           reserved for                  |
     |                future use               |
     |                                         |
     |                                         |
     ~                                         ~
                       .
                       .
                       .
     ~                                         ~
     |                                         |
%177 |                                         |  127
     +-----------------------------------------+
```

DISC COLD LOAD INFORMATION TABLE (SECTORS 28-29)

```
   |------------------------------------------------|
  0|        pointer to table information            | FAEFTR    >-----------|
   |------------------------------------------------|                       |
  1|        pointer to temporary CST info           | TCSTPTR               |
   |------------------------------------------------|                       |
  2|     # of entries to read on disc cold load     | NREAD                 |
   |------------------------------------------------|                       |
  3|     # of code segments in INITIAL              | NVTCST'               |
   |------------------------------------------------|                       |
  4|          INITIAL's DB value                    | INITDB                |
   |------------------------------------------------|                       |
  5|          INITIAL's DL value                    | INITDL                |
   |------------------------------------------------|                       |
  6|          INITIAL's Z value                     | INITZ                 |
   |------------------------------------------------|                       |
  7|          INITIAL's Q value                     | INITQ                 |
   |------------------------------------------------|                       |
  8|          INITIAL's S value                     | INITS                 |
   |------------------------------------------------|                       |
  9|             |  SYSDISC type   |    subtype     | DISCTST               |
   |------------------------------------------------|                       |
 10|                cold load ID                    | COLD'LOAD'ID'         |
   |------------------------------------------------|                       |
 11|              log file number                   | LOG'FILE'NUM'         |
   |------------------------------------------------|                       |
 12|              directory disc                    |                       |
   |                                                | DIRADR                |
 13|                  address                       |                       |
   |------------------------------------------------|                       |
 14|        ldev 1  virtual memory                  |                       |
   |                                                | VIRMEMADDR            |
 15|              disc address                      |                       |
   |------------------------------------------------|                       |
 16|           # LOG PROCS                          |                       |
   |------------------------------------------------|                       |
 17|              LOG ID's                          |                       |
   |------------------------------------------------|                       |
 18|              RIN table                         |                       |
   |                                                | RINADR                |
 19|              disc address                      |                       |
   |------------------------------------------------|                       |
 20|              directory size                    | DIRSECT               |
   |------------------------------------------------|                       |
 21| #sectors in virtual memory region of LDEV 1    | SECTORS IN LDEV1 VM   |
   |------------------------------------------------|                       |
 22|                  UNUSED                        |                       |
   |------------------------------------------------|                       |
 23|              RIN table size                    | RINSECT               |
   |------------------------------------------------|                       |
 24|              # of RINS                         | RINS                  |
   |------------------------------------------------|                       |
```

DISC COLD LOAD INFORMATION TABLE (CONT.)

```
  |-----------------------------------------------|
25|              # of global RINS                 | GRINS
  |-----------------------------------------------|  TL=Tape cold load
26|                                       |TL|RL|RY| LOAD MODE
  |-----------------------------------------------|   RL=Reload
  |                                               |   RY=recovery
27| HIGHEST VOL #          |     # OF VOLUMES     | H'VOL'
  |-----------------------------------------------|
28|              disc cold load entry point       | DISCENTRY
  |-----------------------------------------------|
29|              system  disc DRT number          | SYSDISCDRT
  |-----------------------------------------------|
30|              Job Master Table                 |
  |                                               | JMATLOC
31|              Disc Address                     |
  |-----------------------------------------------|
32|                                               |
  |              IDD Disc Address                 | IDDLOC
33|                                               |
  |-----------------------------------------------|
34|                                               |
  |              ODD Dics Address                 | ODDLOC
35|                                               |
  |-----------------------------------------------|
36|              Welcome Message (DST 47          |
  |                                   10)         | LOGONLOC1
37|              Disc Address                     |
  |-----------------------------------------------|
38|              Welcome Message (DST 48          |
  |                                   10)         | LOGONLOC2
39|              Disc Address                     |
  |-----------------------------------------------|
40|                                               |
  |          LOG ID ADDRESS                       |
41|                                               |
  |-----------------------------------------------|
42|                                               |
  |          LOG TAB ADDRESS                      |
43|                                               |
  |-----------------------------------------------|
44|          LOG ID SIZE                          |
  |-----------------------------------------------|
45|          LOG TAB SIZE                         |
  |-----------------------------------------------|
  |              SIZE IN WORDS                    | FAEFTR+0  <-----------
  |-----------------------------------  *DRIVER  |
  |              MEMORY ADDRESS                   |
  |-----------------------------------   TABLE    |
  |              DISC ADDRESS                     |
  |-----------------------------------------------|
```

DISC COLD LOAD INFORMATION TABLE (CONT.)

```
|-------------------------------------------------|
|              SIZE IN WORDS                       | FAEFTR+4
|------------------------------------              |
|              MEMORY ADDRESS          *CTAB0      |
|------------------------------------              |
|              DISC ADDRESS                        |
|-------------------------------------------------|
|              SIZE IN WORDS                       | FAEFTR+8
|------------------------------------              |
|              MEMORY ADDRESS          *CTAB       |
|------------------------------------              |
|              DISC ADDRESS                        |
|-------------------------------------------------|
|              SIZE IN WORDS           *           | FAEFTR+12
|------------------------------------ COMMUNICA-|
|              MEMORY ADDRESS          TION SUB-   |
|------------------------------------ SYSTEM      |
|                                      DRIVER      |
|              DISC ADDRESS            TABLE       |
|                                                  |
|-------------------------------------------------|
|              SIZE IN WORDS           *           | FAEFTR+16
|------------------------------------ COMMUNICA-|
|              MEMORY ADDRESS          TION SUB-   |
|------------------------------------ SYSTEM      |
|                                      DEFINITION|
|              DISC ADDRESS            TABLE       |
|                                                  |
|-------------------------------------------------|
|              SIZE IN WORDS                       | FAEFTR+20
|------------------------------------ COMMUNICA-|
|              MEMORY ADDRESS          SUBSYSTEM   |
|------------------------------------ TABLE       |
|                                                  |
|              DISC ADDRESS                        |
|                                                  |
|-------------------------------------------------|
|              SIZE IN WORDS                       | FAEFTR+24
|------------------------------------ LOGICAL-    |
|              MEMORY ADDRESS          PHYSICAL    |
|------------------------------------ DEVICE      |
|                                      TABLE       |
|              DISC ADDRESS                        |
|                                                  |
|-------------------------------------------------|
```

```
|-----------------------------------------------|
|             SIZE IN WORDS                     | FAEFTR+28
|------------------------------------ LOGICAL-  |
|             MEMORY ADDRESS           DEVICE    |
|------------------------------------ TABLE     |
|                                               |
|             DISC ADDRESS                      |
|                                               |
|-----------------------------------------------|
|             SIZE IN WORDS                     | FAEFTR+32
|------------------------------------ DEVICE    |
|             MEMORY ADDRESS           CLASS     |
|------------------------------------ TABLE     |
|                                               |
|             DISC ADDRESS                      |
|                                               |
|-----------------------------------------------|
```

DISC COLD LOAD INFORMATION TABLE (CONT.)

```
|-----------------------------------------------|
|             SIZE IN WORDS                      | FAEFTR+36
|-----------------------------------  VOLUME    |
|             MEMORY ADDRESS           TABLE     |
|-----------------------------------             |
|                                                |
|             DISC ADDRESS                       |
|                                                |
|-----------------------------------------------|
|             SIZE IN WORDS                      | FAEFTR+40
|-----------------------------------  LOGICAL   |
|             MEMORY ADDRESS           DEVICE    |
|-----------------------------------  TABLE     |
|                                      EXTENSION |
|             DISC ADDRESS                       |
|                                                |
|-----------------------------------------------|
|             STACK SIZE                         | FAEFTR+44
|-----------------------------------  INITIAL's |
|            MEMORY ADDRESS            STACK     |
|-----------------------------------             |
|                                                |
|             DISC ADDRESS                       |
|                                                |
|-----------------------------------------------|
|             SEGMENT SIZE                       | FAEFTR+48
|-----------------------------------  INITIAL's |
|             MEMORY ADDRESS           SEGMENTS  |
|-----------------------------------             |
|                                                |
|             DISC ADDRESS                       |
|                                                |
|-----------------------------------------------|
|                                                |
.                                                .
.                                                .
.          (MORE SEGMENTS OF INITIAL)            .
.                                                .
```

INITIAL PROGRAM CST MAP

```
LOGICAL           PHYSICAL
 CST#               CST#             SEGMENT NAME
-------           --------           ------------

   0                 1               ININ       \
   1                 2               BOOTSTRAP |----->  core resident
   2                 3               RESIDENT   /
   3                 4               MAINSEG1   \
```

```
 4        5       MAINSEG1A |
 5        6       CONFIGURE |       |non-core resident
 6        7       DEFCTRACKS|       |but present in core
 7       10       SETUP     |------|at completion of
10       11       TAPEIO    |       |cold load
11       12       FILEIO    |
12       13       DISKSPACE /
13       14       DIRECTORY1
14       15       DIRECTORY2
15       16       SL PROGRAM
16       17       PROCESS
17       20       MAINSEG1B
20       21       MAINSEG2
21       22       MAINSEG3
22       23       MAINSEG4
```

*code segment swapping starts at completion of MAINSEG1

Virtual Disc Space Management Structures

Disc space for data segments is allocated from reserved regions of system volumes which have been assigned the virtual memory supporting (VMS) attribute. The data structure used for accounting and management of the virtual disc space of the various VMS volumes is the Virtual Disc Space Management Table (VDSMTAB). This structure consists of a circular list of entries, one for each VMS volume. Each entry contains the information defining the state of the virtual memory region on that volume.

Virtual Disc Space Management Table

VDSMTAB DST# = 39 (%47)
VDSMTABPTR= %1026

General Structure

```
            +------------------+
%1026------>|  # VMS Volumes   |
            |  first to look at|--+
            +------------------+  |
    +------>|                  |  |
    |   +---|  Next in list    |  |
    |   |   |                  |  |
    |   |   +------------------+  |
    |   +-->|                  |<-+
    |   +---|  Next in list    |
    |   |   |                  |
    |   |   +------------------+
    |   +-->|                  |
    +-------|  Next in list    |
            |                  |
            +------------------+
```

VDSMTAB Entry 0 Format

```
             0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
            |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
VDSMTAB00|              #WORDS IN VDSMT                 |TABLELENGTH
            |-----------------------------------------------|
VDSMTAB01| # SYSTEM VOLUMES WHICH HAVE VIRTUAL MEMORY   |VMSVOLUMECNT
            |-----------------------------------------------|
VDSMTAB02|    INDEX OF NEXT ENTRY TO ALLOCATE FROM      |STARTENTRY
            |-----------------------------------------------|
VDSMTAB03|              VM PAGE SIZE (512)              |VMPAGESIZE
            |-----------------------------------------------|
VDSMTAB04|            # SECTORS/VM PAGE (4)             |SECTORSPERVMPAGE
            |-----------------------------------------------|
VDSMTAB05|      OFFSET FROM ENTRY TO BITMAP (%20)       |OFFSETTOBM
            |-----------------------------------------------|
VDSMTAB06|   TOTAL # VM PAGES CONFIGURED IN SYSTEM      |
            |-----------------------------------------------|
VDSMTAB07| LEAST # OF VM PAGES THAT HAVE EVER BEEN AVAIL.|
            |-----------------------------------------------|
            |                                               |
            ~                                               ~

            ~                                               ~
            | VDSMTAB  %10-%17  UNASSIGNED                  |
            |                                               |
            |-----------------------------------------------|
```

VDSMTAB GENERAL ENTRY FORMAT

```
           0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
          |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
Word 0    |   INDEX OF NEXT ENTRY IN CIRCULAR LIST         | NEXTINLIST
          |-----------------------------------------------|
Word 1    |                   LDEV#                        | LDEV
          |-----------------------------------------------|
Word 2    |          STARTING SECTOR OF DEVICE'S           | HOSTARTSECTOR
          |-----------------------------------------------|
Word 3    |             VIRTUAL MEMORY REGION              | LOSTARTSECTOR
          |-----------------------------------------------|
Word 4    |             # SECTORS IN DEVICE'S              | TOTAL SECTOR
          |-----------------------------------------------|
Word 5    |             VIRTUAL MEMORY REGION              | COUNT
          |-----------------------------------------------|
Word 6    | # PAGES IN DEVICE'S VIRTUAL MEMORY REGION      | TOTAL PAGECNT
          |-----------------------------------------------|
Word 7    | # OF PAGES AVAILABLE IN DEVICE'S VM REGION     | PAGESAVAILABLE
          |-----------------------------------------------|
Word %10  |  # OF VALID WORDS IN DEVICE'S BIT MAP          | BMLENGTH
          |-----------------------------------------------|
Word %11  |        SIZE OF SMALLEST RECENT MISS            |SMALLESTMISS
          |-----------------------------------------------|
WORD %12  |  SMALLEST NUMBER OF PAGES EVER AVAILABLE       |
          |-----------------------------------------------|
%13-%20   |            UNASSIGNED                          |
          |-----------------------------------------------|
          |    DEVICE'S VIRTUAL MEMORY BIT MAP             |
          |-----------------------------------------------|
          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
          |-----------------------------------------------|
          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
          |-----------------------------------------------|

           ***COMMENT:  A bit on in a device's VMBIT MAP
                        ==> Corresponding VM page is free.
```

VOLUME TABLE

SIR #22=%26
DST #29=%35

```
                      zero entry
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
word|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
    |                       |                       |
    |     # OF ENTRIES      |                       |
   0|  (NOT COUNTING ZERO)  |   ENTRY SIZE=16(8)    |0
    |-----------------------------------------------|
   1|                 COLD LOAD ID                  |1
    |-----------------------------------------------|.
   2|                   SYSVOLNUM                   |
    |-----------------------------------------------|.
   3|        VIRTUAL MEMORY INTEGRITY NUMBER        |
    |-----------------------------------------------|
    .                                               .
    .                                               .
    |-----------------------------------------------|
  15|///////////////////////////////////////////////|13
    |-----------------------------------------------|
```

TYPICAL PRIVATE VOLUME ENTRY

```
     |-------------------------------------------------|
    0|                                                 |0    indexed by
     |                                                 |     volume #
    1|                                                 |1
     |                     VOLUME                      |
    2|                      NAME                       |2
     |                                                 |
    3|                                                 |3
     |-------------------------------------------------|
    4|                                                 |4
     |                                                 |
    5|                                                 |5
     |                      GROUP                      |
    6|                      NAME                       |6
     |                                                 |
    7|                                                 |7
     |-------------------------------------------------|
   10|                                                 |8
     |                                                 |
   11|                                                 |9
     |                     ACCOUNT                     |
   12|                      NAME                       |10
     |                                                 |
   13|                                                 |11
     |-------------------------------------------------|
     |  LOGICAL DEVICE #     |          |VMS|UN|NS|SC|   NS - NON-SYSTEM
   14| (=0 IF NOT MOUNTED)   |          |   |  |  |  |        DOMAIN
     |-------------------------------------------------|   SC - SCRATCH
     |           |VSET VTABX |       MVTABX            |   UN - UNREADABLE/
   15|           |           |                         |        UNFORMATTED
     |-------------------------------------------------|
```

TYPICAL SYSTEM VOLUME ENTRY

```
  |-----------------------------------------------|
 0|                                               |0    indexed by
  |                                               |     volume #
 1|                                               |1
  |                   VOLUME                      |
 2|                    NAME                       |2
  |                                               |
 3|                                               |3
  |-----------------------------------------------|
 4|                                               |4
  |                                               |
 5|                                               |5
  |                     0                         |
 6|                                               |6
  |                                               |
 7|                                               |7
  |-----------------------------------------------|
10|  STARTING SECTOR OF VOLUME'S VM (0 if none)   |8
  |                                               |
11|                                               |9
  |-----------------------------------------------|
12|                                               |10
  |  NUMBER OF SECTORS RESERVED FOR VM ON VOLUME  |
13|              (0 if none)                      |11
  |-----------------------------------------------|
  |  LOGICAL DEVICE #       |          |VMS|UN|NS|SC|   NS  - NON-SYSTEM
14| (=0 IF NOT MOUNTED)     |          |   |  |  |  |         DOMAIN
  |-----------------------------------------------|   SC  - SCRATCH
  |            |VSET VTABX |      MVTABX          |   UN  - UNREADABLE/
15|            |           |                      |         UNFORMATTED
  |-----------------------------------------------|  VMS  - VIRTUAL MEMORY
                                                            SUPPORTING
```

CHAPTER 4 DIRECTORY

Directory on disc consists of a contiguous area

SYSGLOB cells:

DIRBASE<--------absolute disk addr of base [SYSGLOB+%130 AND %131]

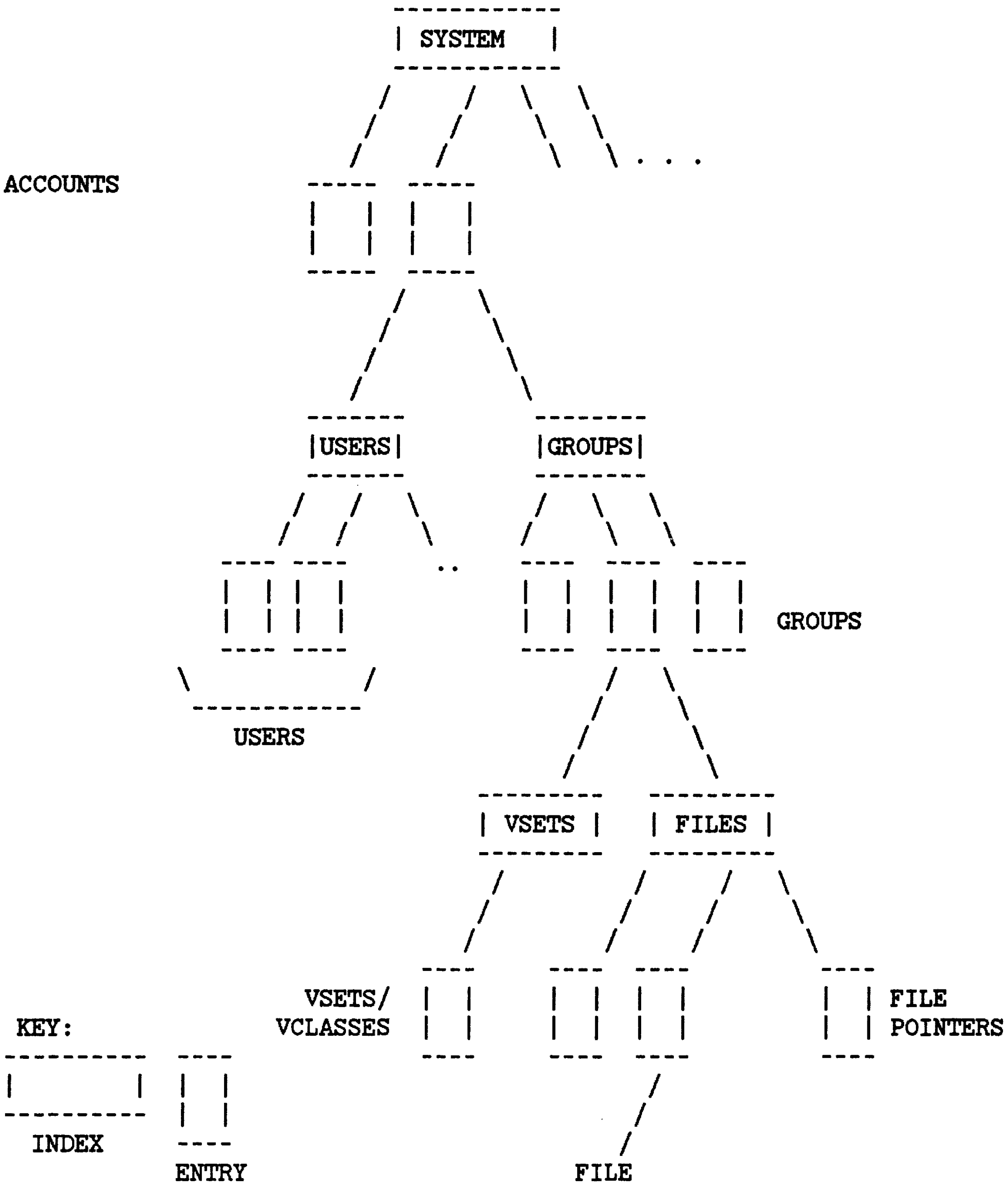
```
DIRBASE------> |------------------|
            ---|     LASTWORD     |
            |  |------------------|
            |  |     FIRSTAVAIL   |--
(WORD ADDR) |  |------------------| |POINTS TO NEXT WORD TO BE EXAMINED
 ----       |  |                  | |
            |  |       BITMAP     <-
            |  ~     (DIRECTORY   ~  The bitmap defines the available/
            |->|    SECTORS USED) |  used sectors in the directory.Bit
               |                  |  0, word 0 corresponds to DIRBASE;
               |------------------|  bit 1 to DIRBASE+1;etc. 1=> avail-
               |         0        |  able 0=> used. [Note: BITMAP (0).
               |------------------|  (0:4)=0 always.]
               |------------------|
               |         .        |
               ~         .        ~
               ~         .        ~
DIRBASE+3----> |------------------|
               ~      DIRECTORY   ~  Directory entries contain pointers
               ~                  ~  which are sector displacements
               |                  |  relative to DIRBASE.  Entries and
               |                  |  indices are grouped into "blocks"
               |------------------|  (block = 3 sectors).
```

The capacities for accounts/groups/users/files are dependent on their block sizes, described in the directory data segment.

*	SYSSAIBSIZE	System acct index block size (sectors)
	SYSAUIBSIZE	Acct. user index block size (sectors)
	SYSAGIBSIZE	Acct. group index block size (sectors)
	SYSGFIBSIZE	Group file index block size (sectors)
	SYSGVSIBSIZE	Group volume set definition ind. blk. size(sectors)
*	SYSAEBSIZE	Acct. entry block size (sectors)
	SYSUEBSIZE	User entry block size (sectors)
	SYSGEBSIZE	Group entry block size (sectors)
	SYSFEBSIZE	File entry block size (sectors)
	SYSMAXBSIZE	Maximum of above. (used to initialize DDS.)
	SYSVSEBSIZE	Volume set definition entry block size (sectors)

*These values are used once for the creation of the (root) system, account index or new systems. This root index is always at address DIRBASE+3.

OVERVIEW OF DIRECTORY



DIRECTORY DATA SEGMENT

```
          0  |-------------------------|  0
          .  |         SECTOR          |  .
          .  ~         BUFFER          ~  .
          .  ~     128(10) WORDS       ~  .
        177  |                         |  127
             |-------------------------|
        200  | ADJUST (DB-DL)          |  128
             |-------------------------|
        201  | XTYPE (INPUT PARM)      |  129
             |-------------------------|
        202  |           :  XMVTABX    |  130
             |-------------------------|
        203  | XINDEXP (FINAL INDEX PRT)|  131
             |-------------------------|
        204  | XANAME (DB REL ADDR)    |  132
             |-------------------------|
        205  | XGUNAME (DB REL ADDR)   |  133
             |-------------------------|
        206  | XFNAME (DB REL ADDR)    |  134
             |-------------------------|
        207  | XASEC (ACCOUNT SECURITY) |  135
             |-------------------------|
        210  |                         |  136
             |-XGSEC (GROUP SECURITY) -|
        211  |                         |  137
             |-------------------------|
        212  | SIRRETURN (FROM GETSIR) |  138
             |-------------------------|
    213-240  ~ DIRECTORY POINTER "A"   ~  139-160 \
             |-------------------------|           > SEE Directory
    241-266  ~ DIRECTORY POINTER "B"   ~  161-182 /  Pointer Area
             |-------------------------|
        267  |/////////////////////////|  183
             |-------------------------|
        270  |   LDEV    :  DIRECTORY  |  184
             |-------------            -|
        271  |   BASE DISC ADDRESS     |  185
             |-------------------------|
SYSSAIBSIZE=3 | SYS.ACCT.INDEX BLK SIZE |  186
             |-------------------------|
      AUI=1  | ACCT.USER INDEX BLK SIZE |  187
             |-------------------------|
      AGI=1  | ACCT.GRP INDEX BLK SIZE |  188
             |-------------------------|
      GFI=2  | GRP FILE INDEX BLK SIZE |  189
             |-------------------------|
     GVSI=1  |GRP VOL DEF INDEX BLK SIZE|  190
             |-------------------------|
      AEB=3  | ACCT ENTRY BLK SIZE     |  191
             |-------------------------|
```

DIRECTORY DATA SEGMENT (CONT.)

```
                  |--------------------------|
       UEB=2      | USER ENTRY BLK SIZE      |  192
                  |--------------------------|
       GEB=2      | GRP ENTRY BLK SIZE       |  193
                  |--------------------------|
       FEB=2      | FILE ENTRY BLK SIZE      |  194
                  |--------------------------|
       VSEB=1     | VOL DEF ENTRY BLK SIZE   |  195
                  |--------------------------|
  DDSBSIZE=3      | MAX.SIZE DIRECTORY BLOCK |  196
                  |--------------------------|
DDSBWSIZE=%600    | DDSBSIZE*128             |  197
                  |--------------------------|
                  |       DISTRIBUTION       |  198
GOODPERCENT=.85   |-                        -|
         307      |          FACTOR          |  199
                  |--------------------------|
         310      |           BASE           |  200
                  |--------------------------|  ---
         311      |                          |  201
                  ~         DA AREA          ~  DDSBWSIZE
                  |                          |
                  |--------------------------|  ---
                  ~                          ~
                  ~                          ~

                  |--------------------------|  ---
                  |        WORK AREA         |
                   (SIZE OF LARGEST ENTRY)      MAX
                  |                          |
                  |--------------------------|  ---
                  ~                          ~
                  ~                          ~

                  |--------------------------|  ---
        1145      |                          |  613
                  ~          DB AREA         ~  DDSBWSIZE
                  |                          |
                  |--------------------------|  ---
```

```
      DIRECTORY POINTER AREA [DA OR DB]    DST=20(10)
     -----------------------------------   SIR=8(10)

  ^ |--------------------------------| 138/
  | |   LDEV          | DIRECTORY BASE|     160 DIRBASE1'
  | |----------------                | 139/
  | |  ADDRESS OF PAGE IN BUFFER     |     161 DIRBASE2'
  | |--------------------------------| 140/
  | | DIRECTORY PAGE IN BUFFER       |     162 CONTENTS
  | |--------------------------------| 141/
  | | DB ADDRESS OF 1ST ELEMENT      |     163 LPNTR
  | |--------------------------------| 142/
  | | STARTING ADDRESS OF BUFFER     |     164 IOPNTR
  | |--------------------------------| 143/
  | | # VALID PAGES IN BUFFER        |     165 NUMVALID
  | |--------------------------------| 144/
  | | DIRTY FLAG                     |     166 DIRTY
  | |--------------------------------| 145/               NOTE:
  | | ELEMENT SIZE                   |     167 XSIZE       -----
  | |--------------------------------| 146/
 ** | # WORDS USED IN BLOCK          |     168 USED     **  INDEXES AND
  | |--------------------------------| 147/                ENTRIES
  | | BLOCK SIZE (SECTORS)           |     169 BSIZE
  | |--------------------------------| 148/             *  INDEXES ONLY
  | | BLOCK SIZE (WORDS)             |     170 BWSIZE
  | |--------------------------------| 149/
  | | MAX # ELEMENTS/BLOCK           |     171 BFACTOR
  | |-|-|-|-|----------|-------------| 150/
  | |I|P| TY|ELEMENT SIZE|BLOCK SIZE|     172 MISCWD
  | | | | | |   (WORDS)  | (SECTORS)|
  | |-|-|-|-|------------|----------| 151/
  | | NUMBER OF ELEMENTS             |     173 XCOUNT
  v |--------------------------------| 152/
  ^ | NUMBER OF ACCESSORS            |     174 PCOUNT
  | |--------------------------------| 153/
  | | ENTRY TOTAL                    |     175 ETOTAL
  | |-|-|-|-|----------|-------------| 154/
  | |0|P| TY|ENTRY SIZE| BLOCK SIZE |     176 EMISCWD
  | |-|-|-|-| (WORDS)  | (SECTORS)  |
  | |-|-|-|-|----------|-------------| 155/
  | | FATHER INDEX POINTER           |     177 PINDEXP
  | |--------------------------------| 156/
  | |  F             |               |     178
  * |----A-----------|---------------| 157/
  | |      T         |   N           |     179 PNAME   TY = 0-FILE
  | |--------H-------|-----A---------| 158/                 1-GROUP
  | |          E     |       M       |     180              2-ACCT
  | |------------R---|---------E-----| 159/                 3-USER
  | |                |               |     181              4-VSD
  v |----------------|---------------|                 I  = 0-ENTRY BLOCK
                                                            1-INDEX BLOCK
                                                       P  = PURGE FLAG
```

DIRECTORY SPACE DATA SEGMENT (DIRSDS)

DST=21_{10}

SIR=8_{10}

```
   |----------------|
   | LDEV  |        |          base address of parent directory
   |--------        |
   |    DIRBASE     |
   |----------------|
---|    LASTWORD    |          defines last word of bit map
|  |----------------|
|  |   FIRSTAVAIL   |--        defines next word to be examined
|  |----------------| |
|  |                | |
|  |                | |
|  |                | |
|  |                | |
|  |                | |
|  |                | |          The bitmap defines the available/used
|  |                | |          sectors in the directory.  Bit 0 word 0
|  |     BITMAP     <--          corresponds to DIRBASE; bit 1 word 0 to
|  |                |            DIRBASE+1 etc. 1=>available 0=>used.
|  |                |            [NOTE:  bitmap(0).(0:4)=0 always.]
|  |                |
|  |                |
|  |                |
|  |                |
-->|                |
   |----------------|
   |                |
   |----------------|
   |       0        |
   |----------------|
```

DIRECTORY STRUCTURE

```
                                        ENTRY BLOCK
                                        -----------

                          --------> |--------------------|
                          |         |                    |
                          |         |--------------------|
                          |         |                    |
                          |         |--------------------|
                          |         |                    |
                          |         |--------------------|
                          |
                          |  -----> |--------------------|
                          |  |      |                    |
                          |  |      |--------------------|
                          |  |      |                    |
                          |  |      |--------------------|
                          |  |      |                    |
                          |  |      |--------------------|
                          |  |
                          |  |  --> |--------------------|
    INDEX BLOCK           |  |  |   |       ENTRY        |
    -----------           |  |  |   |--------------------|
|--------------------|    |  |  |   |       ENTRY        |
|       INDEX        |    |  |  |   |--------------------|
|       BLOCK        |-----  |  |   |       ENTRY        |
|       PREFIX       |       |  |   |--------------------|
|--------------------|       |  |   |       ENTRY        |
|       INDEX        |       |  |   |--------------------|
|       BLOCK        |--------  |   |       ENTRY        |
|       ENTRY        |          |   |--------------------|
|--------------------|          |   |       ENTRY        |
|       INDEX        |          |   |--------------------|
|       BLOCK        |-----------
|       ENTRY        |
|--------------------|
```

DIRECTORY DEFINITIONS

>PAGE - smallest allocatable record ("phys.recd")-currently sector.
>BLOCK - integral# of pages; contains contiguous indices or entries.
>INDEX - pointer to entry block, containing name of 1st entry.
>ENTRY - information-containing "object" may contain pointer to an index block.
>POINTER - 15-bit positive relative page number (relative to directory base).
>DDS - directory data segment.
>ELEMENT - a generic name for index or entry.

INDEX BLOCK PREFIX (10 WORDS)

```
  0-FILE \
  1-GROUP|  3 bits       ------------>| INDEX SIZE (WORDS)
  2-ACCT |<--------      |            | 7 BITS
  3-USER /        |      |
  4-VSET          |      |
                  |      |      ----->| BLOCK SIZE (SECTORS)
PURGE FLAG<----   |      |      |     | 4 BITS
              |   |      |      |
           |-|-|----|----------|-----|
MISCWD    0|1|P| TY |  XSIZE   |BSIZE|0  INDEX BLOCK INFO.
           |-|-|----|----------|-----|
          1|        XCOUNT           |1  NUMBER OF INDEX POINTERS
           |-------------------------|
          2|       IPCOUNT           |2  NUMBER OF ACCESSORS*
           |-------------------------|
          3|        ETOTAL           |3  ENTRY TOTALS
           |-|-|----|---------|------|
EMISCWD   4|0|P| TY | EXSIZE  |EBSIZE|4  ENTRY BLOCK INFO.
           |-|-|----|---------|------|
          5|       PINDEXP           |5  INDEX POINTER OF FATHER
           |-------------------------|
          6|                         |6  \
           |                         |   |
          7|                         |7  |
           |       PNAME             |    >NAME OF FATHER
         10|                         |8  |
           |                         |   |
         11|                         |9  /
           |=========================|
           |                         |
           |~~~~~~~~~~~~~~~~~~~~~~~~~|
```

*The count is incremented by each access that uses and relies upon a pointer to the index block, ie, it is guaranteed not to be purged while the count is not = 0.

INDEX ENTRY (6 WORDS)

```
|--------------------------|
|                          |0  1st NAME OF ENTRY BLOCK
|                          |
|                          |1
|        IE1STNAME         |
|                          |2
|                          |
|                          |3
|                          |
|--------------------------|
|         IEPNTR           |4  POINTER TO ENTRY BLOCK
|--------------------------|
|         IECOUNT          |5  NUMBER OF ENTRIES IN e BLOCK
|--------------------------|
```

ACCOUNT ENTRY (%36 WORDS)

```
   |--------------------|
  0|                    |0
   |                    |
  1|                    |1  ACCT.NAME
   |       ANAME        |
  2|                    |2
   |                    |
  3|                    |3
   |--------------------|
  4|      AGIPNTR       |4  ACCT.GROUP INDEX POINTER
   |--------------------|
  5|      AUIPNTR       |5  ACCT.USER INDEX POINTER
   |--------------------|
  6|                    |6
   |       ACAP         |   CAPABILITY
  7|                    |7
   |--------------------|
 10|                    |8
   |      ALATTR        |   LOCAL ATTRIBUTES
 11|                    |9
   |--------------------|
 12|                    |10 PASSWORD
   |                    |
 13|                    |11
   |       APASS        |
 14|                    |12
   |                    |
 15|                    |13
   |--------------------|
```

ACCOUNT ENTRY (CONT.)

```
     |--------------------|
   16|                    |14
     |     ADFSCOUNT      |   DISC FILE SPACE COUNT (SECTORS)
   17|                    |15
     |--------------------|
   20|                    |16
     |     ADFSLIMIT      |   DISC FILE SPACE LIMIT (SECTORS)
   21|                    |17
     |--------------------|
   22|                    |18
     |     ACPUCOUNT      |   CPU TIME COUNT (SECONDS)
   23|                    |19
     |--------------------|
   24|                    |20
     |     ACPULIMIT      |   CPU TIME LIMIT (SECONDS)
   25|                    |21
     |--------------------|
   26|                    |22
     |   ACONTIMECOUNT    |   CONNECT TIME COUNT (MINUTES)
   27|                    |23
     |--------------------|
   30|                    |24
     |   ACONTIMELIMIT    |   CONNECT TIME LIMIT (MINUTES)
   31|                    |25
     |--------------------|
---32| |/|//| | | | | | | |26 FLAGS (SEE BELOW)
|    |-|-|--|-|-|-|-|-|-|-|
|  33|S|A|/////|          |27 MAX.JOB PRIORITY
|    |---------|----------|
|  34|COMM FILE REC # ACCT|28 command file location of      HARD  CODED
|    |--------------------|   account udc's                   0     1
|  35|COMM FILE REC # SYS |29 command file location of        |     |
|    |--------------------|   system udc's (SYS acct only)    |     |
|                                                               |     |
|                                                               |     |
|                                                               |     |
|       |--|--|--|--|----|--|----|--|----|--|----|--|----|--|----|--|
|       | P|//|/////| R  | R| A  | A| W  | W| L  | L| X  | X| S  | S|
|->ASECW|  |//|/////|ANY |AC|ANY |AC|ANY |AC|ANY |AC|ANY |AC|ANY |AC|
        |--|--|--|--|----|--|----|--|----|--|----|--|----|--|----|--|
                         \---------------- -------------------/
                                          |
    P    PURGE flag                       |
                                     FILE SECURITY


    S    If 1, system level UDC's exist (only in "SYS" account)
    A    If 1, account level UDC's exist for account
```

```
                GROUP ENTRY (%51 WORDS)
                ----------------------


    |--------------------|
   0|                    |0  GROUP NAME
    |                    |
   1|                    |1
    |       GNAME        |
   2|                    |2
    |                    |
   3|                    |3
    |--------------------|
   4|      GFIPNTR       |4  GROUP FILE INDEX POINTER
    |--------------------|
   5|                    |5
    |                    |
   6|                    |6  PASSWORD
    |       GPASS        |
   7|                    |7
    |                    |
  10|                    |8
    |--------------------|
  11|                    |9  DISC FILE SPACE COUNT (SECTORS)
    |     GDFSCOUNT      |
  12|                    |10
    |--------------------|
  13|                    |11 DISC FILE SPACE LIMIT (SECTORS)
    |     GDFSLIMIT      |
  14|                    |12
    |--------------------|
  15|                    |13 CPU TIME COUNT (SECONDS)
    |     GCPUCOUNT      |
  16|                    |14
    |--------------------|
  17|                    |15 CPU TIME LIMIT (SECONDS)
    |     GCPULIMIT      |
  20|                    |16
    |--------------------|
  21|                    |17 CONNECT TIME COUNT (MINUTES)
    |   GCONTIMECOUNT    |
  22|                    |18
    |--------------------|
  23|                    |19 CONNECT TIME LIMIT (MINUTES)
    |   GCONTIMELIMIT    |
  24|                    |20
    |--------------------|
  25|*P|                 |21 GROUP SECURITY (SEE BELOW)
    |--|      GSEC       |
  26|                    |22 *P = PURGE FLAG
    |--------------------|
```

GROUP ENTRY (CONT.)

```
   |--------------------|
 27|     GCAPABILITY    |23 GROUP CAPABILITY
   |--------------------|
 30|     GLINKAGE       |24 GROUP DIR. BASE LINKAGE
   |--------------------|
 31|     GVSDIPNTR      |25 GROUP VOL SET DEFN INDX
   |--------------------|
 32|     GHVSNAME       |26 HOME VOL SET NAME
   |-                  -|
 33|                    |27
   |-                  -|
 34|     GHVSANAME      |28 (Definition's acct name)
   |-                  -|
 35|                    |29
   |--------------------|
 36|                    |30
   |-                  -|
 37|                    |31
   |-    GHVSGNAME     -|   (Definition's group name)
 40|                    |32
   |-                  -|
 41|                    |33
   |--------------------|
 42|                    |34
   |-                  -|
 43|                    |35
   |-    GHVSVSNAME    -|   (Definition's vol set name)
 44|                    |36
   |-                  -|
 45|                    |37
   |--------------------|
 46|    GSAVEFIPNTR     |38 SAVE CELL FOR GFIPNTR
   |--------------------|
 47|   GMOUNTREFCNTR    |39 GROUP BIND COUNTER
   |--------------------|
 50|         0          |40 GSPARE
   |--------------------|
```

GROUP ENTRY (CONT.)

```
GLINKAGE    (0:1) = 0; System Domain
            (0:1) = 1; Private Volumes
            (8:8) = 0; Not Bound
            (8:8) <>0; Bound

              GROUP SECURITY MASK
    |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---
    | P |///| R | R | R | R | R | A | A | A | A | A | W | W | W | W
  25|   |///|ANY|AC |AL |GU |GL |ANY|AC |AL |GU |GL |ANY|AC |AL |GU
    |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---
    | W | L | L | L | L | L | X | X | X | X | X | S | S | S | S | S
  26|GL |ANY|AC |AL |GU |GL |ANY|AC |AL |GU |GL |ANY|AC |AL |GU |GL
    |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---
```

```
          FILE ENTRY (FILE POINTER)(6 WORDS)
          ---------------------------------

          |--------------------|
          |                    |0  FILE NAME
          |                    |
          |                    |1
          |   FNAME            |
          |                    |2
          |                    |
          |                    |3
          |--------------------|
          | FVTABINX |         |4  VOL TABLE INDX / FILE LABEL DISC
          |----------|         |                    ADDRESS
          |     FLABELADDR     |5
          |--------------------|


          USER ENTRY (19 WORDS)
          ----------

          |--------------------|
         0|                    |0  USER NAME
         1|       UNAME        |1
         2|                    |2
         3|                    |3
          |--------------------|
         4|                    |4  CAPABILITY
          |        UCAP        |
         5|                    |5
          |--------------------|
         6|                    |6  LOCAL ATTRIBUTES
          |       ULATTR       |
         7|                    |7
          |--------------------|
        10|                    |8  PASSWORD
        11|       UPASS        |9
        12|                    |10
        13|                    |11
          |--------------------|
        14|                    |12 HOME GROUP (MAY BE NULL)
        15|     UHGROUP        |13
        16|                    |14
        17|                    |15
          |--------------------| LOG CNT (# OF USERS LOGGED ON)
        20|     ULOGCOUNT      |16 INIT TO 1 FOR MANAGER.SYS SO
          |--------------------|   THIS USER CANNOT BE PURGED
UMAXJOBW 21|*P|U|  0  | JOBPRI  |17 MAX.JOB PRI;*P=PURGE FLAG
          |----------|---------|           U=UDC EXIST FLAG
        22|COMM FILE REC #     |18
          |(command file loc of|
          | user udc's)        |
          |--------------------|
```

USER ATTRIBUTES/CAPABILITY

```
                               /        SAVE FILES  ------
       FILE-ACCESS ATTRIBUTES <                          |
                               \ NON-SHARABLE DEVICES--  |
                                 COMMUNICATIONS-----  |  |
                                                   |  |  |
                                                   |  |  |
          / -----------------SYSTEM MGR            |  |  |
         |  |  --------------ACCOUNT MGR           |  |  |
         |  |  |  -----------ACCOUNT LIBRN         |  |  |
  USER   |  |  |  |  --------GROUP LIBRN           |  |  |
         |  |  |  |  |  -----DIAGNOSTICIAN         |  |  |
         |  |  |  |  |  |  --SYSTEM SUPVSR         |  |  |
  ATTR   |  |  |  |  |  |  | CREATE VOLS           |  |  |
         |  |  |  |  |  |  |  |  USE VOLS           |  |  |
          \ |  |  |  |  |  |  |  |  USER LOGGING    |  |  |
            |  |  |  |  |  |  |  |  |              |  |  |
            |  |  |  |  |  |  |  |  |              |  |  |
           |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
           |SM|AM|AL|GL|DI|OP|CV|UV|LG|//|//|//|//|CS|ND|SF|
           |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|


             0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
           |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
           |//|//|//|//|//|//|//|BA|IA|PM|//|//|MR|//|DS|PH|
           |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|


                                  |  |  |        |     |  |
          /      batch access    -|  |  |        |     |  |
          |interactive access    ----|  |        |     |  |
 ACCESS   |   privileged mode    -------|        |     |  |
 TO      <
 GENERAL  |     multiple RINS    ----------------|     |  |
 RESOURCES|extra data segment    ----------------------|  |
          \  process handling    -------------------------|
```

VOLUME SET DEFINITION ENTRY

```
                   |------------------------------------------|
                  0|                                          |0
                  1|                                          |1  VOLUME
                  2|                GVSNAME                   |2  SET
                  3|                                          |3  NAME
                   |------------------------------------------|
     TY = 0       4|TY|1            7|      MVTABX            |4  GVSLINKAGE
                   -------------------------------------------
                  5|VOL COUNT|4     7|       VMASK            |5  GVSINFO
                   -------------------------------------------
               /  6|                                          |6  MEMBER VOLUME
               |  7|                                          |7  NAME (1ST ENTRY
 VOLUME        | 10|              GVSVOLUME                   |8  IS MASTER
 ENTRY 0       < 11|                                          |9  VOLUME)
 (6 WORDS)     |   |------------------------------------------|
               | 12|0                                     14| M|10 GVSVOLFLAGS
                   |------------------------------------------|
               \ 13|  SUB-TYPE         |      VTABX           |11 GVSVOLINFO
                   |------------------------------------------|
               / 14|                                          |12
 VOLUME        | . |                    .                     | .
 ENTRIES       | . ~                    .                     ~ .
 1 - 7         < . ~                    .                     ~ .
               | . |                    .                     | .
               \ 57|                                          |47
                   |------------------------------------------|
                 60|                                          |48
                   |                                          |
                 61|                                          |49
                   |                                          |
                 62|              GVSVOLUME                   |50 MEM. VOL.
                   |                                          |   NAME
                 63|                                          |51
                   |------------------------------------------|
                 64|  GVSVOLFLAGS   (MEMBER VOLUME FLAGS)     |52
                   |------------------------------------------|
                 65|  GVSVOLINFO    (MEMBER VOLUME INFO)      |53
                   |------------------------------------------|
                 66|  GVSDREFCNT    (DEFN. REF. CNTR.)        |54
                   |------------------------------------------|
                 67|                  0                       |55  SPARE
                   |------------------------------------------|
```

TY = 0 VOLUME SET
   = 1 VOLUME CLASS
MVTABX: MOUNTED VOLUME TABLE INDEX (IF MOUNTED)
VOL COUNT: NO. OF VOLUMES
VMASK: VOLUME MASK
M = 0 NOT MOUNTED
  = 1 MOUNTED
VTABX: VOLUME TABLE INDEX

GVSLINKAGE

```
 0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
|-------------------------------------------------------------|
| T | A |           NOT           |           MVTABX          |
|   |   |           USED          |                           |
|-------------------------------------------------------------|
```

T - TYPE
1 = Volume Set Definition
0 = Volume Set Class
A - ALLOCATING FLAG
0 = not initially allocating (not 1st user of set)
1 = 1st user of set allocating resources (transitional)
MVTABX - Mounted Volume Table Index
0 if volume set not logically mounted

GVSINFO

```
 0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
|-------------------------------------------------------------|
|    VOLCNT     |       NOT       |          VSMASK           |
|               |       USED      |                           |
|-------------------------------------------------------------|
```

VOLCNT - Number of members in set
VSMASK - Bit mask of volume member usage
Order is from right to left
i.e. bit 15 is 1st member, bit 14 is 2nd member ...

GVSVOLFLAGS

```
 0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
|-------------------------------------------------------------|
|                       NOT USED                          | M |
|                                                         |   |
|-------------------------------------------------------------|
```

M - Member Mounted Flag
0 = not mounted
1 = mounted

GVSVOLINFO

```
 0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
|-------------------------------------------------------------|
|          DISK                 |           VTABX             |
|        SUB-TYPE               |                             |
|-------------------------------------------------------------|
```

VTABX - Volume Table Index

VOLUME SET CLASS ENTRY

```
                        1 1 1 1 1 1
    0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
   |-------------------------------|
  0|                               | 0   VOLUME CLASS NAME
   |-                             -|
  1|                               | 1
   |-          GVCNAME            -|
  2|                               | 2
   |-                             -|
  3|                               | 3
   |-------------------------------|
  4|         GVCLINKAGE            | 4   VOLUME CLASS IDENTIFICATION
   |-------------------------------|
  5|          GVCINFO              | 5   VOLUME CLASS INFORMATION
   |-------------------------------|
  6|          GVCPNAME             | 6   PARENT VOLUME SET DEFINITION
   |-                              |
  7|                               | 7
   |-         GVCPANAME           -|     ACCOUNT OF PARENT DEFINITION
 10|                               | 8
   |-                             -|
 11|                               | 9
   |-------------------------------|
 12|                               |10
   |-                             -|
 13|                               |11
   |-         GVCPGNAME           -|     GROUP OF PARENT DEFINITION
 14|                               |12
   |-                             -|
 15|                               |13
   |-------------------------------|
 16|                               |14
   |-                             -|
 17|                               |15
   |-         GVCPVSNAME          -|     VSNAME OF PARENT DEFINITION
 20|                               |16
   |-                             -|
 21                                |17
   |-------------------------------|
 22|              0                |18
   |-------------------------------|
 23|              0                |19
   ~                               ~

   ~                               ~
   |-------------------------------|
 67|              0                |55
   |-------------------------------|
```

G V C L I N K A G E

```
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
|-----------------------------------------------------------------|
| T | 0 |        NOT            |               0                 |
|   |   |        USED           |                                 |
|-----------------------------------------------------------------|
```

T - TYPE
      1 = Volume Set Definition
      0 = Volume Set Class

G V C I N F O

```
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
|-----------------------------------------------------------------|
|    VOLCNT     |      NOT      |            VCMASK               |
|               |      USED     |                                 |
|-----------------------------------------------------------------|
```

VOLCNT - Number of members in set
VCMASK - Bit mask of volume member usage (VOLUME CLASS MASK)
         Order is from right to left
          i.e. bit 15 is 1st member, bit 14 is 2nd member ...

VOLUME MASK FORMAT

- USED IN MVTAB, PVUSER, FILE CONTROL BLOCK (FCB), VOLUME SET/CLASS DEFINITION, VOLUME SET VTAB.
- 8-BIT MASK.

```
   -----------------------------------------
   | V7 | V6 | V5 | V4 | V3 | V2 | V1 | V0 |
   -----------------------------------------
     ^    ^    ^    ^    ^    ^    ^    ^
     |    |    |    |    |    |    |    |
     |    |    |    |    |    |    |    -- VOLUME 0 (MASTER)
     |    |    |    |    |    |    |
     |    |    |    |    |    |    ------- VOLUME 1
     |    |    |    |    |    |
     |    |    |    |    |    ------------ VOLUME 2
     |    |    |    |    |
     |    |    |    |    ----------------- VOLUME 3
     |    |    |    |
     |    |    |    ---------------------- VOLUME 4
     |    |    |
     |    |    --------------------------- VOLUME 5
     |    |
     |    -------------------------------- VOLUME 6
     |
     ------------------------------------- VOLUME 7
 0:  NOT MOUNTED OR NON-MEMBER    1:   MOUNTED OR MEMBER
```

CHAPTER 5 LOCK RESOURCES

SIR# ALLOCATION DST %53

decimal SIR #	octal SIR #	SIR NAME
→ 1	1	LOAD PROCESS SIR
2	2	LOCK SEGMENT SIR
3	3	IDD
4	4	ODD
5	5	PROCESS TREE STRUCTURE
6	6	SCHEDULING QUEUE
7	7	CST ENTRIES
8	10	SYSTEM DIRECTORY
9	11	LPDT
10	12	LDT
11	13	STORAGE IN OVERLAY AREA
12	14	DISC FREE SPACE TABLE
13	15	JPCNT
14	16	JCUT
15	17	JMAT
16	20	FMAVT
17	21	LOADER SEGMENT TABLE
18	22	VDD
19	23	SPOOL
20	24	MESSAGE CATALOGUE
21	25	RIT
22	26	VOLUME TABLE
23	27	WELCOME MESSAGE SIR
24	30	ASSOCIATION TABLE
25	31	CS ALLOCATE SIR
26	32	LOGGING BUFFER
27	33	PV MVTAB
28	34	MEASSIR
29	35	PV USER TABLE
30	36	IMAGE
31	37	KSAM
32	40	USER LOGGING
$ 33	41	DEBUG BREAKPOINT TABLE
$ 34	42	PCBSIR
35	43	SUB-QUEUE MAPPING TABLE
36	44	CILOG
37	45	FILE INTEGRITY
38	46	RIN
39	47	TAPE LABELS
40	50	1st JOB
41	51	2nd JOB
.	.	.
.	.	.
.	.	.
.	.	.

MULTIPLE SIR ALLOCATION

The five conventional chains used by MPE for SIR allocation and deallocation are:

LOWER->LOGICAL RANK->HIGHER
1. LDT(10)->LPDT(9)->VDD(18)
2. JMATSIR(15)->LPDT(9)->JPCNT(13)
3. FMAVTSIR(16)->FILESIR(37)->DIRECT.(8)->DISC FREE SPACE TBLE(12)
4. FMAVTSIR(16)->FILESIR(37)->RINTABLE(38)

$ 5. SEGTABSIR (%21)-> BKPTSIR(%41)-> LOCKSIR(2)
6. JMATSIR(15)->LDT(10)->LPDT(9)->ODD(4)

Multiple SIR allocation requires care to avoid process deadlock situations. The rule that should be followed when working with the above SIRs is as follows:

Never attempt a GETSIR of lower rank then the SIR currently held (if any).

For example: suppose two processes, A and B, required the SIRs for the LDT and LPDT. Deadlock would result if done as below due to process A not following the convention order.

```
          incorrect order                    correct order
          ---------------                    -------------

          PROCESS "A"                        PROCESS "B"
              .                                  .
              .                                  .
              .                                  .
              .                                  .
              .                                  .
              .                                  .
          GETSIR(9)  [LPDT]<---      ------->GETSIR(10)  [LDT]
              .               |      |           .
              .               |      |           .
              .        -------|------            .
              .        |      |                  .
              .        |      |                  .
              .        |      |                  .
          GETSIR(10)---       ---------------GETSIR(9)
                            DEADLOCK
```

SIR TABLE INFORMATION

The system internal resource table is located in non-linked memory (resident table). The SIR table is used to protect critical system elements against access by more than one process, i.e., it provides a "lock out" mechanism. Each critical system resource (usually a table) is assigned a specific SIR number. Procedures are provided within MPE to lock (GETSIR) and unlock (RELSIR) the SIR. Processes attempting to obtain a SIR that is not available are impeded by the system. The SIR table entries form the head of a linked list in this case. If more than one process becomes impeded, word 8 of the PCB entry is used to add the "new" process to the growing list. The method of disimpeding the process depends on the SIR type.

A SIR does not respect process priority and operates in a FIFO manner. As processes become impeded on behalf of a SIR the new entries are entered at the tail of the impeded list. When the current holder of the SIR releases it, only the first process in the list (pointed at by the head pointer) is dis-impeded. The linked list head and all pointers are then updated and the newly dis-impeded process will obtain the SIR.

SIR ENTRY FORMATS

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|                        0                       |0      free
|------------------------------------------------|    ------------
|                        0                       |    (not locked)
|------------------------------------------------|

|------------------------------------------------|
|     PIN of holder      |           0           |0     SIR locked
|------------------------------------------------|  --------------------
|                       0                        |  (no impeded processes
|------------------------------------------------|

|------------------------------------------------|
|     PIN of holder      |   SIR QUEUE LENGTH    |0     SIR locked
|------------------------------------------------|   -------------------
|TAIL OF IMPEDED LIST(P)|HEAD OF IMPEDED LIST(P)|1  (impeded processes)
|------------------------------------------------|
```

P = PIN#
PIN = PCB table entry number
SIR QUEUE LENGTH- number of processes queued for this SIR

The SIR table is indexed by SIR#, each SIR# corresponding to a unique, preassigned system internal resource. Entry #0 is not used.
Impeded lists are established by using the SIR table entry (1). (8:8) as the head of the list and PCB(8). (8:8) for elements. Pin numbers are always used as pointers, with 0 indicating end of list.

RIN TABLE GENERAL LAYOUT (Initialized State)

```
  DST=%26      |-----------------------------------------------|--
               |           INDEX OF FIRST FREE ENTRY           | |
               |-----------------------------------------------| |
 --------------|            (# LOCAL+GLOBAL RINS)*2            | |
 |             |-----------------------------------------------| | FIRST
 |             | RT   |       INDEX OF NEXT FREE               |<- FREE
 |             |-----------------------------------------------|-- ENTRY
 |             |                       0                       |  |
 |             |-----------------------------------------------<--
 |             | RT   |       INDEX OF NEXT FREE               |--
 |             |-----------------------------------------------| |
 | RT=RIN TYPE |                       0                       | |
 |  (WHEN      |-----------------------------------------------| |
 |  ALLOWED)   | RT   |       INDEX OF NEXT FREE               |<-
 |             |-----------------------------------------------|--
 |             |                       0                       | |
 | 1-LOCAL RIN |-----------------------------------------------|<-
 | 2-GLOBAL    | RT   |       INDEX OF NEXT FREE               |--
 |   RIN       |-----------------------------------------------| |
 | 3-FILE RIN  |                       0                       |<-
 |             |-----------------------------------------------|
 |             | RT   |       INDEX OF NEXT FREE               |--
 |             |-----------------------------------------------| |
 |             |                       0                       | |
 |             |-----------------------------------------------| |
 |             |                                               | |
 |             ~                                               ~ | .
 |             ~                                               ~ | .
 |             |                                               | |
 |             |-----------------------------------------------| | LAST
 |             | RT   |             0(EOL)                     |<- FREE
 |             |-----------------------------------------------|   ENTRY
 |             |                       0                       |
 |             |-----------------------------------------------|
 ------------->|               FREE LIST POINTER               |--
 SECONDARY     |-----------------------------------------------| |
 TABLE OF 12-  |               TOTAL #OF ENTRIES               | |
 WORD ENTRIES  |-----------------------------------------------| |
 FOR GLOBAL    |              NUMBER FREE ENTRIES              | |
RIN'S ONLY     |-----------------------------------------------| |
               |                   RESERVED                    | |
   ------------|-----------------------------------------------| |
        |      |IF FREE, PTT TO NEXT FREE|                     |0<
        |      |-------------------------|                     |1
        |      ~                                               ~
     LENGTH=
   # ALLOCATED
   GLOBAL RINS
       *12     ~                                               ~
        |      |                                               |10
        |      |                                               |11
   ------------|-----------------------------------------------|
```

ALLOCATION AND LOCKING OF LOCAL RINS

```
                                                      DST22(10)=26(8)
                              RIN TABLE
          |------------------------------------------------|
          |           INDEX OF FIRST FREE ENTRY            |
          |------------------------------------------------|
          |        (NUMBER OF LOCAL+GLOBAL RINS)*2         |
          |------------------------------------------------|
          |                                                |
          |                                                |
          |                                                |
          |                                                |
          |                                                |
          |                                                |
JOB INFOR-|------------------------------------------------|  LOCAL
  MATION  | 0  1|         INDEX OF NEXT RIN                |--  RIN #1
  TABLE   |------------------------------------------------| |(UNLOCKED)
  -----   |                                                | |
  (JIT)   |                                                | |
|------|  |                                                | |
|      |  |                                                | |
|      |  |                                                | |
|      |43|                                                | |
|------|  |------------------------------------------------|<-  LOCAL
| LOCK |->| 0  1|         INDEX OF NEXT RIN                |--RIN #2
|  RIN |  |------------------------------------------------| |(UNLOCKED)
|INDEX |  |                       |                        | |
|------|  |------------------------------------------------| |
          |                                                | |
          |                                                | |
          |                                                | |
          |                                                | |
          |                                                | |
          |------------------------------------------------|<-  LOCAL
          | 0  1|               0 END OF LIST              |  RIN #3
          | |------------------------------------------------|  (LAST)
          | HEAD OF WAITING LIST  |     PIN OF HOLDER (P)  | (LOCKED)
          |------------------------------------------------|
          |       |                          |             |
          |       |                          |             |
           --------                 ---------|             |
           |                         |
           |                         |
           v                         v
          PCB#  waiting processes    PCB#  PCB# of process
                linked through             that "holds" rin
                PCB impeded queue
                (PCB#  pointers)
P = PIN#
```

ALLOCATION AND LOCKING OF FILE RINS

```
                                           DST22(10)=26(8)
                   RIN TABLE
|-----------------------------------------------|
|          INDEX OF FIRST FREE ENTRY            |
|-----------------------------------------------|
|        (NUMBER OF LOCAL+GLOBAL RINS)*2        |
|-----------------------------------------------|
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|-----------------------------------------------|
| 1  1|/////////////////////////////////////////|
|-----------------------------------------------|
|HEAD OF WAITING LIST(P)|     PIN OF HOLDER     |
|-----------------------------------------------|
|       |                        |              |
|       |                        |              |
 --------                 -------
 |                        |
 v                        v
 PCB# waiting processes   PCB#  process that
      linked through PCB       "holds" rin
      impeded queue

 P=pin#
```

ALLOCATION AND LOCKING OF GLOBAL RINS

```
                                                      DST22(10)=26(8)
                              RIN TABLE
          |------------------------------------------------|
          |           INDEX OF FIRST FREE ENTRY            |
          |------------------------------------------------|
          |        (NUMBER OF LOCAL+GLOBAL RINS)*2         |
          |------------------------------------------------|
          |                                                |
          |                                                |
          |                                                |
          |                                                |
          |------------------------------------------------|
          |  1 0|    INDEX OF PASSWORD, USERNAME           |-----------
          |------------------------------------------------|          |
          |HEAD OF WAITING LIST(P)|     PIN OF HOLDER      |          |
          |------------------------------------------------|          |
          |             |                     |            |          |
          |             |                     |            |          |
          |             |                     |            |          |
          |             |                     |            |          |
  PCB#  <-|--------------                     -------------|->PCB#    |
          |                                                |          |
 waiting  |                                                |  process |
processes |                                                |   that   |
          ~                                                ~  'holds' |
          ~                                                ~   RIN    |
          |                                                |          |
          |                                                |          |
          |------------------------------------------------|          |
          |                                                |<----------
          |                                                |
          |                 RIN PASSWORD                   |
          |                                                |
          |                                                |
          |------------------------------------------------|
          |                                                |
          |                   USERNAME                     |
          |            (USER NAME AND ACCOUNT)             |
          |                                                |
          |------------------------------------------------|

           P=pin#
```

CHAPTER 6 FILE SYSTEM

1.0 Introduction

This document describes the MPE-IV file system. Section 2 describes the basic concepts. Section 3 describes the table structures used.

2.0 File System Overview

I/O to files is done by reference to file numbers, which are assigned by calling the FOPEN intrinsic. This establishes an initial "point of attachment", which may be described as a connection between a program (i. e., process) and that particular point in a particular file at which the next FREAD or FWRITE would cause data to be transferred. A point of attachment is described by a control block, of which there are several different kinds (described later). Control blocks may exist in the process's own stack, in an extra data segment assigned by the file system, or (because of file sharing) in some other process' stack. In order to find control blocks quickly, a pointer scheme called vectors is used. A control block is uniquely described by a vector, which consists of one word with the low ten bits containing a segment number, and the upper six containing an index into a table (the "vector table") which describes the location of the control block within that segment. The entire assemblage, consisting of five overhead words, the vector table, and all of the control blocks to which it points, comprises a contiguous piece of storage called the "control block table". If it is in an extra data segment, the control block table comprises the entire segment; if in a stack, it occupies part of the PXFILE part of the PCBX, usually beginning at segment-relative location 106 octal.

The point of attachment is described by a "physical access control block", or PACB, which will exist as a result of an FOPEN to any file (except $NULL). Any required I/O buffers are associated with the PACB; see section 2.1.

All FOPENs specifying "multi-access" for all processes running under a single job use a single PACB for references to a multi-access file. Although all these are attached to a single point in the file, the type of attachment (i. e., AOPTIONS) may be different. So, each FOPEN specifying a multi-access file establishes a "logical access control block", or LACB, which contains the point-of-attachment local values. The use of a single buffer (i. e., PACB) insures that references by various processes or against various FOPENs within one process are dealt with in strict sequential order. Note that references to a file by other jobs, or by other processes not specifying multi-access, will be through other PACBs, whose buffers will be read or written at the pleasure of the file system; in order to insure any sort of coherence to such shared references, the jobs must use global RINS and FLOCK and FUNLOCK the file. $STDIN, $STDLIST, and spoolfiles are opened multi-access automatically.

In the case of disk files, there is another kind of control block: the file control block, or FCB. It contains copies of information read from the file label, such as the end- of-file pointer, the extent map, and the record and block structure. The EOF pointer is updated in the FCB as the file is written, and all changes made to the FCB are posted to the file label when the file is closed. An FCB is shared by all jobs in the system which reference the file.

The file number assigned by an FOPEN is an index into the Available File Table (AFT), a table of four-word entries which is at the end of the PXFILE part of the PCBX. Two of these words are vectors to the PACB and (if it exists) the LACB.

Because control blocks are shared among processes, it is necessary to have a scheme for coordinating access to them. A control block is "locked" by a process which requires exclusive access to it for a time. Other processes which attempt to lock the block will find it already locked, and will be impeded and queued. It may also be necessary to lock an entire control block table so that a process can create or destroy a control block in it, or lock or unlock an existing control block in the table.

Another table used by FOPEN is the File Multi-Access Vector Table (FMAVT). This table exists in a system extra data segment and is used by all jobs and processes in the system. When a file is being FOPENed with multi-access specified, the FMAVT is searched; if the file is already open, the FMAVT gives the PACB vector for the prior reference for each job.

2.1 Buffers

A bit in AOPTIONS specifies, when a file is opened, whether access is to be buffered or unbuffered. If unbuffered, data is transferred directly between the I/O device and the user's buffer (usually in his stack), which will be frozen in memory for the duration of the transfer. If buffered, the data is moved between the user's buffer and a file system buffer to which the I/O is actually done.

Buffers are associated with the PACB, attached to it as an appendage.

3.0 Table Formats

This section gives a detailed discussion of the main tables constructed and used by the file system. The location and overall structure of each table is given, in addition to the table format and a discussion of each field in the table. Table indices at the right of the table are in octal. Index names apply to the entire word; if in parentheses, the names are defined in the file system listing but not explicitly used there.

3.1 File System Section of PCBX (PXFILE)

The PXFILE area is a sub-section of the PCBX. It is a contiguous, expandable and contractable block of storage that is managed by the file system primarily for its own use. Other subsystems, namely CS and DS, also make use of the PXFILE section. In doing so they must conform to the conventions of the file system.

The overall structure of the PXFILE area is:

```
-------------------------
|                       | 66
|       Overhead        |      (fixed)
|                       |
|-----------------------|
|                       | 106
|     Control block     |      (variable)
|         table         |
|                       |
|-----------------------|
|                       |
|       Available       |      (variable)
|                       |
|-----------------------|
|                       |
|      Active File      |      (variable)
|         Table         |
|                       | DL-5
-------------------------
```

3.1.1 Overhead

The part labeled Overhead contains information that pertains to the entire section. It ordinarily begins at segment-relative location 66 octal, but is usually addressed via the pointer at DL-3.

```
  0   1                 7   8                   15
 ----------------------------------------------------
 |               PXFILE size in words               |  0  PXFSIZE
 |--------------------------------------------------|
 |  Last DOPEN error no.   |  Last COPEN error no.  |  1
 |--------------------------------------------------|
 | N |                                              |  2
 |--------------------------------------------------|
 |                 Reserved for DS                  |  3  (PXFDSINFO)
 |--------------------------------------------------|
 | Last KOPEN error number | Last FOPEN error number |  4
 |--------------------------------------------------|
 |                AFT size in words                 |  5  PXAFTSIZE
 |--------------------------------------------------|
 |               CS Trace file info                 |  6  (PXCTRINFO)
 |--------------------------------------------------|
 |  Last responding NO-WAIT I/O AFT entry number    |  7  PXFLEFTOFF
 |--------------------------------------------------|
 |  1st user (NOBUF) control block table DST number | 10  PXFCBT1
 |--------------------------------------------------|
 |  2nd user (NOBUF) control block table DST number | 11  (PXFCBT2)
 |--------------------------------------------------|
 |  3rd user (NOBUF) control block table DST number | 12  (PXFCBT3)
 |--------------------------------------------------|
 |  4th user (NOBUF) control block table DST number | 13  (PXFCBT4)
 |--------------------------------------------------|
 |  5th user (NOBUF) control block table DST number | 14  (PXFCBT5)
 |--------------------------------------------------|
 |  6th user (NOBUF) control block table DST number | 15  (PXFCBT6)
 |--------------------------------------------------|
 |  7th user (NOBUF) control block table DST number | 16  (PXFCBT7)
 |--------------------------------------------------|
 |  8th user (NOBUF) control block table DST number | 17  (PXFCBT8)
 ----------------------------------------------------
```

Partial word field identifiers are:

```
     PXFDOPEN      = PXFILE(1).(0:8)#,   last DOPEN error code
     PXFCOPEN      = PXFILE(1).(8:8)#,   last COPEN error code
     PXFNOCB       = PXFILE(2).(0:1)#,   no CB's in PXFILE CBT?
     PXFKOPEN      = PXFILE(4).(0:8)#,   last KOPEN error code
     PXFFOPEN      = PXFILE(4).(8:8)#,   last FOPEN error code
```

Discussion:

PXFAFTSIZE	This is the size (in words) of the Active File Table (AFT). The size is in words to simplify calculating the size of the available block.

PXFCBT1-8	These are the DST numbers of the user (NOBUF) control block tables. A DST number of 0 indicates that no data segment is allocated.
PXFCOPEN	This contains the last COPEN error number. Not used by the file system.
PXFCTRINFO	This contains information pertinent to the CS trace file. Not used by the file system.
PXFDOPEN	This contains the last DOPEN error number. Not used by the file system.
PXFDSINFO	Reserved for DS. Not used by the file system.
PXFFOPEN	This contains the last FOPEN error number. If it is zero then the last FOPEN completed successfully; otherwise the last FOPEN was unsuccessful and the number is the file system error number.
PXFKOPEN	This contains the last KOPEN error number. KSAM is partly imbedded in the file system, and an FOPEN failure on a KSAM file can be caused by a failure to open either the key file or the data file. This error number is used in conjunction with PXFFOPEN to determine which file caused the KSAM open failure. This error number is not used by the file system.
PXFLEFTOFF	This is the AFT entry number of the last file/line that completed a no-wait I/O; if zero then no no-wait I/O has been completed. This cell is maintained solely by and for the IOWAIT intrinsic.
PXFNOCB	This bit signifies that control blocks are not to be created in the PXFILE control block table. This bit is set by the NOCB parameter to the CREATE intrinsic or the :RUN command. This feature permits the user to have as much stack space as possible; otherwise the file system will take several hundred words of stack for the PXFILE control block table.
PXFSIZE	This is the size (in words) of the complete PXFILE area. It is the sum of the overhead block, the control block table, the active file table and the available block.

3.1.2 PXFILE Control Block Table (PXFCBT)

Addressing within a PXFILE control block table is somewhat more complicated than addressing an extra data segment CBT since the table does not begin at DB+0. As a result all pointers within the table are table relative; the starting address of the table must be added to a pointer to generate a final DB-relative address. This addressing convention is consistently applied to all control block tables.

When the control block table is expanded, space is taken from the AVAILABLE area. If no space is available then the PXFILE area is expanded and the acquired space is added to the AVAILABLE area.

Refer to section 3.2 for a more detailed description of file control block tables.

```
  0   1   2                                      15
 ---------------------------------------------------
 |                Table size in words              | 20  (PXFCBTAB)
 |-------------------------------------------------|
 |            DST number containing table          | 21  PXFDSTX
 |-------------------------------------------------|
 |   0   |         Vector table size in words      | 22  PXFVTSIZE
 |-------------------------------------------------|
 |                    Lock word                    | 23  (PXFLOCK)
 |-------------------------------------------------|
 |                 Impeded queue                   | 24  (PXFQUEUE)
 |-------------------------------------------------|
 |                                                 | 25  PXFVT
 |                                                 |
 |                 Vector table                    |
 |                                                 |
 |                                                 |
 |-------------------------------------------------|
 |                                                 |
 |                                                 |
 |                                                 |
 |                                                 |
 |              Control block area                 |
 |                                                 |
 |                                                 |
 |                                                 |
 |                                                 |
 ---------------------------------------------------
```

The following identifier is also used:

PXFCBTSIZE = PXFILE(16)#, table size in words

Discussion:

PXFCBTAB	This is the first word of the control block table; it is used when referring to the entire table.
PXFCBTSIZE	This is the size in words of the control block table. It is used principally for calculating the size of the available block.
PXFDSTX	This is the DST number of the data segment that contains the control block table. This is the same as the DST number of the stack itself. The common convention of referring to the DST number of the stack as zero is not used, because the file system may refer to a PXFILE control block table in another stack, which would result in an ambiguity since that PXFILE control block table would also have a DST number of zero.
PXFLOCK	This is the lock word for the table and has the same format as the lock word for a control block in the table, i. e. lock bit, break bit, lock count, and locking PIN.
PXFQUEUE	This is the impeded queue for the table and has the same format as the impeded queue for a control block in the table.
PXFVT	This is the first word of the vector table. It is used when referring to the vector table in general.
PXFVTSIZE	This is the size, in words, of the vector table. This is the length of the table and does not reflect the number of entries used or unused.

3.1.3 Available Block

The part labeled Available is used to provide space when the Control Block Table or the Active File Table is expanded. These two tables grow towards each other, and when more space is needed it is simply taken from the Available Block.

When the Available area is exhausted, the PXFILE area is expanded, the AFT is relocated and the new space is added to the Available Block.

Currently the PXFILE area is only expanded; it is never contracted.

3.1.4 Active File Table (AFT)

The part labeled Active File Table contains information used by the file system (or CS, DS, etc.) to grossly characterize the file access and, most importantly, to give the location of the control blocks.

The overall structure of the AFT is:

```
------------------------
|                      |
|       Entry N        |     (fixed, 4 words)
|                      |
|----------------------|
|                      |
|          .           |
|          .           |
|          .           |
|                      |
|----------------------|
|                      |  DL-8
|       Entry 1        |     (fixed)
|                      |  DL-5
------------------------
```

where N = PXFAFTSIZE/4.

The length of the AFT is specified by PXFAFTSIZE. Unused entries are all zeroes. When the table is full it is expanded by taking space from the Available block.

The AFT is negatively indexed by file number: the entry at DL-8 corresponds to file number 1, the entry at DL-12 corresponds to file number 2, etc.

The structure of a file system AFT entry is:

```
   0   1   2   3   4   5                               15
 -----------------------------------------------------
 | Entry type    | N |                               |  0
 |---------------------------------------------------|
 |               Physical ACB Vector                 |  1  AFTPACBV
 |---------------------------------------------------|
 |               Logical ACB Vector                  |  2  AFTLACBV
 |---------------------------------------------------|
 |                NO-WAIT I/O IOQX                   |  3  AFTIOQX
 -----------------------------------------------------
```

The entry format depends on the entry type; the file system uses entry type 0.

The following partial word field identifiers are used:

AFTTYPE	= AFT.(0:4)#,	entry type
AFTNULL	= AFT.(4:1)#,	$NULL file

Discussion:

AFTIOQX	This is the IOQ index of the pending no-wait I/O (if any). This is applicable iff the file was opened with the NOWAIT option specified. Also, CS and DS have the same capability and use this cell in a consistent manner. This is because the IOWAIT intrinsic services the file system as well as CS and DS, and is the principal user of this cell. In the case of a message file the accessor's reply port (file system basic IPC port) is stored in this cell. If this cell is zero there is no no-wait I/O pending.
AFTLACBV	This is the vector of the Logical ACB (LACB) (if any). This is applicable iff the file was opened with the multi-access option specified.
AFTNULL	This bit signifies that the file is $NULL and that there are no control blocks.
AFTPACBV	This is the vector of the Physical ACB (PACB). A PACB exists for all files except $NULL.
AFTTYPE	This is the AFT entry type number. At present the following entry types are defined:

0 - file system
1 - remote file
2 - DS (no-wait I/O disallowed)
3 - DS (no-wait I/O allowed)
4 - CS
5 - CS
6 - KSAM
8 - Message File

3.2 File Control Block Table (CBTAB)

A file control block table can be located in two places: (a) as a sub-part of the PXFILE area, as discussed in section 3.1.2; or (b) in a data segment. Although putting control block tables in PXFILE has the advantage of providing rapid access, it detracts from the space for the user's stack; so the larger control blocks (or optionally, all control blocks) are put into extra data segments. On the other hand, referencing extra data segments may result in an absence trap, which is slow. Extra data segment control block tables are of three kinds: expandable, non-expandable, and shared FCB. Non-expandable CBT's are used for a single PACB with buffers, i. e. where the control block is large, or where the control block can't be local to a single process, i. e., for multi-access. Expandable (or NOBUF) CBT's are used for small control blocks, to wit, LACB's, PACB's with no buffers, and FCB's which are local to a single process. A list of the expandable CBT's associated with a process is kept in the overhead area of PXFILE (cf. section 3.1.1). When a small control block is needed, these CBT's are checked in order to see if one of them has room. Shared FCB CBT's are like expandable CBT's except that they belong to the system rather than to a single process; the system keeps a list of DST's which it has assigned for this purpose.

The overall structure of a control block table is:

```
          ------------------------
          |                      |
          |      Overhead        |     (fixed, 5 words)
          |                      |
          |----------------------|
          |                      |
          |                      |
          |    Vector Table      |     (variable)
          |                      |
          |                      |
          |----------------------|
          |                      |
          |                      |
          |                      |
          |                      |
          |    Control Block     |     (variable)
          |        area          |
          |                      |
          |                      |
          |                      |
          ------------------------
```

3.2.1 Overhead

The part labeled Overhead contains information pertaining to the entire table.

```
  0   1   2       6   7                          15
 -----------------------------------------------------
 |               Table size in words                 |  0  CBTSIZE
 |---------------------------------------------------|
 |            DST Number containing table            |  1  CBTDSTX
 |---------------------------------------------------|
 | Type  |            |  Vector table size in words  |
 |---------------------------------------------------|
 |                    Lock word                      |  3  CBTLOCK
 |---------------------------------------------------|
 |                  Impeded queue                    |  4  (CBTQUEUE)
 -----------------------------------------------------
```

Other identifiers used:

```
     CBTTYPE   = CBTAB(2).(0:2)#;  control block table type
     CBTVTSIZE = CBTAB(2).(7:9)#;  vector table size
```

Discussion:

CBTDSTX	This is the DST number of the data segment that contains the control block table. If the table is contained in a stack, i.e. in the PXFILE area, then this is the DST number of the stack and not 0.
CBTLOCK	This is the lock word for the table and has the same format as the lock word for a control block in the table, i. e. lock bit, break bit, lock count, and locking PIN. The table is locked, thus insuring exclusive access, whenever a control block is being created or destroyed. It isn't necessary to lock the table while locking a control block within it because control block locking is done pseudo-disabled.
CBTQUEUE	This is the impeded queue for the table and has the same format as the impeded queue for a control block in the table. There is no second impeded queue because that facility is used exclusively for BREAK requests against the PACB for $STDIN/$STDLIST.
CBTSIZE	This is the size in words of the table. It is initialized when the table is created and changed when the table is expanded. At present a table is never contracted, even though this is possible.

CBTTYPE	This field is the type of the control block table. Possible values are: 0 - stack [PXFILE] 1 - NOBUF (expandable) 2 - System shared FCB 3 - Buffered (contains a single PACB)
CBTVTSIZE	This is the size, in words, of the vector table area in the control block table. It does not reflect the number of entries used or unused.

3.2.2 Vector Table

The part labeled Vector Table contains information used to locate and lock or unlock control blocks in the control block table.

The overall structure of the vector table is:

```
                    ------------------------
                    |                      |
                    |       Entry 0        |     (fixed, 4 words)
                    |                      |
                    |----------------------|
                    |                      |
                    |          .           |
                    |          .           |
                    |          .           |
                    |                      |
                    |----------------------|
                    |                      |
                    |       Entry N        |     (fixed)
                    |                      |
                    ------------------------
```

where N = (CBTVTSIZE/4)-1. Since only six bits are available for a vector table index, the vector table can contain at most 64 entries.

An unused vector table entry will have zeroes in all the words of the entry. A used vector table entry will have a non-zero value in the first word of the entry (the control block address is necessarily non-zero).

The general structure of a vector table entry is:

```
  0                                            15
 ----------------------------------------------------
 |               Control block address              |  0  VTADR
 |--------------------------------------------------|
 |                   Control word                   |  1  VTCONTROL
 |--------------------------------------------------|
 |            High priority impeded queue           |  2  (VTQUEUE)
 |--------------------------------------------------|
 |            Low priority impeded queue            |  3  (VTSAVEDQUEUE
 ----------------------------------------------------
```

Discussion:

VTADR	Control block address is the table relative address of the control block associated with the vector table

	entry. It is a word displacement from the beginning of the control block table.
VTCONTROL	The control word is used to coordinate access to the control block. It contains a bit which indicates that the control block is being accessed, and therefore "locked", and a byte which contains the PIN of the process which has exclusive access to the control block. Other processes attempting to access the block will be impeded and queued.
VTQUEUE	The high priority impeded queue is a byte pair of PINs that are the head and tail of the impeded queue of processes waiting for access to the control block. Processes are impeded and unimpeded by the file system using the normal mechanisms available under MPE.
VTSAVEDQUEUE	The low priority impeded queue is a byte pair of PINs and has the same format as VTQUEUE. The only time this word is used is when the control block is in BREAK mode, which can only happen to an ACB corresponding to $STDIN/$STDLIST. It is used to save the current VTQUEUE when the control block goes into BREAK mode and to restore VTQUEUE when the control block goes back into non-BREAK mode.

The last three words of a vector table entry comprise a sub-block for the locking system that is used to coordinate access to a particular control block.

The structure of the vector table entry control sub-block is:

```
  0  1  2           7  8                 15
 -----------------------------------------------------
 | L | B |   Lock count   |        Lock PIN          |  0  CBLCONTROL
 |---------------------------------------------------|
 | High priority tail PIN | High priority head PIN   |  1  CBLQUEUE
 |---------------------------------------------------|
 |  Low priority tail PIN |  Low priority head PIN   |  2  CBLSAVEDQUEUE
 -----------------------------------------------------
```

The following partial word field identifiers are used:

```
CBLLOCK         = CBL.(0:1)#,       lock bit
CBLBREAK        = CBL.(1:1)#,       break bit
CBLCOUNT        = CBL.(2:6)#,       lock count
CBLPIN          = CBL.(8:8)#,       PIN holding lock
CBLTAIL         = CBL(1).(0:8)#,    high priority tail PIN
CBLHEAD         = CBL(1).(8:8)#,    high priority head PIN
CBLSAVEDTAIL    = CBL(2).(0:8)#,    low priority tail PIN
```

```
CBLSAVEDHEAD    = CBL(2).(8:8)#;    low priority head PIN
```

Discussion:

CBLBREAK	This is the BREAK bit and is used only for the ACB corresponding to $STDIN/$SDTLIST.
CBLCONTROL	This identifier is used when referring to the first word of the vector table control sub-block.
CBLCOUNT	This is a count of the number of times that the control block is locked by CBLPIN. It is 0 if the control block is not locked and is greater than 0 if the control block is locked.
CBLHEAD	This is the PIN of the process at the head of the high priority impeded queue.
CBLLOCK	This is the lock bit for a control block; 1 denotes locked.
CBLPIN	This is the PIN of the process which has locked the control block and has exclusive access to it. If the control block is not locked then this field is 0.
CBLQUEUE	This is the high priority impeded queue.
CBLSAVEDHEAD	This is the PIN of the process at the head of the low priority impeded queue.
CBLSAVEDQUEUE	This is where CBLQUEUE is saved when creating a break queue.
CBLSAVEDTAIL	This is the PIN of the process at the tail of the low priority impeded queue.
CBLTAIL	This is the PIN of the process at the tail of the high priority impeded queue.

3.2.3 Control Block Area

The part labeled CONTROL BLOCK AREA contains the control blocks used by the file system.

To facilitate storage management, all control blocks have the same overall structure:

```
  0   1   2                                     15
 -----------------------------------------------------
 | Type  |                    Size                    |  0  CBDESCRIP
 |---------------------------------------------------|
 |                                                   |  1
 |                                                   |
 |                                                   |
 |                      Data                         |
 |                                                   |
 |                                                   |
 |                                                   |  N
 ------------------------------------------------------
```

where N = Size-1.

Partial word field identifiers are:

```
     CBTYPE           = CB.(0:2)#,      control block type no.
     CBSIZE           = CB.(2:14)#;     control block size
```

Discussion:

CBDESCRIP	This is the first word of a control block; the format is common for all control blocks.
CBSIZE	This is the size (in words) of the control block. The size includes the descriptor word.
CBTYPE	This is the type number of the control block. There are four types of control blocks: 0 - Garbage 1 - FCB 2 - PACB 3 - LACB

When a control block table is created the initial control block area is completely allocated to a single control block of type garbage. When space is requested for a new control block the control block area is scanned (using a first fit algorithm) for a garbage control block that is as large as the size requested. The space for the new control block is taken from this garbage control block and the space remaining becomes the new garbage control block size.

When space is returned it becomes a new garbage control block. To reduce fragmentation the new garbage control block is combined with either of the two neighboring control blocks if they are of type garbage.

If space is requested and no garbage control block is large enough to contain the new control block then the control block area and control block table are expanded by a sufficient amount. If expansion is not possible, some other control block table must be used.

3.2.4 Access Control Block (ACB)

Virtually every file system intrinsic constructs an ACB as its first action. When using the multi-access option, each accessor shares a single PACB. However each accessor is permitted to view the shared file in a slightly different manner than the other accessors. For example, one accessor may access the file in a read-only mode while the other accessors may access the file in a read-write mode. To do this, each accessor must, during his access, have a slightly different ACB.

The PACB holds information that is global to all accessors of the file. The LACB holds information that is local to each accessor of the file. At the beginning of a particular access, an ACB is constructed by calling LOC'ACB, which copies information from both the LACB and the PACB. At the end of the access, the ACB is released by calling UNLOC'ACB; this updates the PACB and LACB from the ACB since some of the fields may have been modified due to the access. This scheme nearly eliminates EXCHANGEDB's to access the various data segments.

3.2.5 Logical Access Control Block (LACB)

All LACBs have the same structure:

```
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
-----------------------------------------------------------------
|   3   |                  Complete LACB size                   |  0
|---------------------------------------------------------------|
|                               |         File number           |  1
|---------------------------------------------------------------|
|     File name - 1st char.     |     File name - 2nd char.     |  2
|---------------------------------------------------------------|
|     File name - 3rd char.     |     File name - 4th char.     |  3
|---------------------------------------------------------------|
|     File name - 5th char.     |     File name - 6th char.     |  4
|---------------------------------------------------------------|
|     File name - 7th char.     |     File name - 8th char.     |  5
|---------------------------------------------------------------|
|                           FOPTIONS                            |  6
|---------------------------------------------------------------|
|                           AOPTIONS                            |  7
|---------------------------------------------------------------|
|                     Record size in bytes                      | 10
|---------------------------------------------------------------|
|                      Block size in words                      | 11
|---------------------------------------------------------------|
|                     Reserved for PACBV                        | 12
|---------------------------------------------------------------|
|                    Carriage control code                      | 13
|---------------------------------------------------------------|
|   |EOF|Pg |Ln |St |FK |TC |TB |8B |Car|DB | EOF T | EOF M |   | 14
|---------------------------------------------------------------|
|               | TE| IC| Q |   |   Terminal stop character     | 15
|---------------------------------------------------------------|
|                          Error code                           | 16
|---------------------------------------------------------------|
|                   Last I/O transmission log                   | 17
+---------------------------------------------------------------+
```

Partial word field identifiers are:

```
    LACBSIZE        = LACB.(2:14)#,   size in words
    LACBSTOPCHAR    = LACB(2).(0:8)#, terminal stop character
```

Discussion:

LACBAOPTIONS	See ACBAOPTIONS.
LACBBSIZE	See ACBBSIZE.

LACBCTL	See ACBCTL.
LACBERROR	See ACBERROR.
LACBFNUM	See ACBFNUM.
LACBFOPTIONS	See ACBFOPTIONS.
LACBMODE	See ACBMODE.
LACBNAME1-8	See ACBNAME.
LACBPACB	This is the vector of the Physical ACB (PACB) for the file.
LACBRSIZE	See ACBRSIZE.
LACBSIZE	This is the size, in words, of the LACB. All LACBs are sixteen (decimal) words long.
LACBSTATE	See ACBLSTATE.
LACBSTOPCHAR	See ACBSTOPCHAR.
LACBTLOG	See ACBTLOG.

3.2.6 Physical Access Control Block (PACB)

The overall structure of the PACB is:

```
                    -------------------------
                    |                       |
                    |      Basic PACB       |     (fixed)
                    |                       |
                    |-----------------------|
                    |                       |
                    |       Buffering       |
                    |                       |     (variable)
                    |       extension       |
                    |                       |
                    -------------------------
```

The buffering extension is optional; it is present if and only if the file is accessed with buffering. There are thus two possible formats for an ACB:

1. No buffers; the buffering extension is not present.

2. PACB buffers; the buffering extension is present and the buffers are in the buffering extension.

If multiple PACB buffers exist, there will be a buffering extension for each, immediately preceding the buffer. The basic PACB (or NOBUF PACB) is copied into the the ACB as words 0 thru 57 octal; an ACB "extension" is then generated in words 60 thru 67. The resulting ACB thus has the following format:

```
        0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
      ----------------------------------------------------------------
  0  |  2   |                    Complete ACB size                    |  0
     |----------------------------------------------------------------|
  1  |                               |          File number           |  1
     |----------------------------------------------------------------|
  2  |    File name - 1st char.      |    File name - 2nd char.       |  2
     |----------------------------------------------------------------|
  3  |    File name - 3rd char.      |    File name - 4th char.       |  3
     |----------------------------------------------------------------|
  4  |    File name - 5th char.      |    File name - 6th char.       |  4
     |----------------------------------------------------------------|
  5  |    File name - 7th char.      |    File name - 8th char.       |  5
     |----------------------------------------------------------------|
  6  |                            FOPTIONS                            |  6
     |----------------------------------------------------------------|
  7  |                            AOPTIONS                            |  7
     |----------------------------------------------------------------|
  8  |                      Record size in bytes                      | 10
     |----------------------------------------------------------------|
  9  |                      Block size in words                       | 11
     |----------------------------------------------------------------|
 10  |            (Reserved for PACBV, if multi-access)               | 12
     |----------------------------------------------------------------|
 11  |                     Carriage control code                      | 13
     |----------------------------------------------------------------|
 12  |   |EOF|Pg |Ln |St |FK |TC |TB |8B |Car|DB | EOF T | EOF M |    | 14
     |----------------------------------------------------------------|
 13  |               | TE| IC| Q |   |   Terminal stop character      | 15
     |----------------------------------------------------------------|
 14  |                           Error code                           | 16
     |----------------------------------------------------------------|
 15  |                  Last I/O transmission log                     | 17
     +----------------------------------------------------------------+
 16  |                                                                | 20
     |                          File pointer                          |
 17  |                                                                | 21
     |----------------------------------------------------------------|
 18  |                                                                | 22
     |                Current variable block number                   |
 19  |                                                                | 23
     |----------------------------------------------------------------|
 20  |                                                                | 24
     |                     Record transfer count                      |
 21  |                                                                | 25
     |----------------------------------------------------------------|
 22  |                                                                | 26
     |                     Block transfer count                       |
 23  |                                                                | 27
     |----------------------------------------------------------------|
 24  |                                                                | 30
     |                 Highest block number started                   |
 25  |                                                                | 31
     |----------------------------------------------------------------|
 26  |                           FCB Vector                           | 32
```

```
   |---------------------------------------------------------------|
27 |                            Spare                              | 33
   |---------------------------------------------------------------|
28 |       No. input LACB'S         |      Total no. LACB'S         | 34
   |---------------------------------------------------------------|
29 |   |Bk |      Device type       |    Last logical I/O status    | 35
   |---------------------------------------------------------------|
30 |AE |RW |ABR|NE | SEOFS |  EOFS |      Blocking factor          | 36
   |---------------------------------------------------------------|
31 |PF |Hit|       | Current buffer| Tape Displace.|  No. buffers  | 37
   |---------------------------------------------------------------|
32 |                   Current record word index                   | 40
   |---------------------------------------------------------------|
33 |                          Buffer size                          | 41
   |---------------------------------------------------------------|
34 |                            Spare                              | 42
   |---------------------------------------------------------------|
35 |                          FMAVT index                          | 43
   |---------------------------------------------------------------|
36 |                      Volume table index                       | 44
   |---------------------------------------------------------------|
37 |           Name type           |       File disposition        | 45
   |---------------------------------------------------------------|
38 |        Access bit map         |    Logical device number      | 46
   |---------------------------------------------------------------|
39 | S | M | Q | R | D |           | Virtual logical device no.    | 47
   |---------------------------------------------------------------|
40 |  Spooled device type  |     Spooled device record size        | 50
   |---------------------------------------------------------------|
41 |                   Spooled device FOPTIONS                     | 51
   |---------------------------------------------------------------|
42 |                   Spooled device AOPTIONS                     | 52
   |---------------------------------------------------------------|
43 |                       IDD or ODD Index                        | 53
   |---------------------------------------------------------------|
44 |                                                               | 54
   |                      No-Wait disk address                     |
45 |                                                               | 55
   |---------------------------------------------------------------|
46 |                            Spare                              | 56
   |---------------------------------------------------------------|
47 |                            Spare                              | 57
   +---------------------------------------------------------------+
```

The above words, 0-%57, are physically located in the PACB of the file. Below, words %60-%67, are used by file system intrinsics and are placed onto the stack by the procedure LOC'ACB when locking the ACB. Therefore, the buffering extention, if present, will immediately follow word %57 of the actual ACB in the Control Block Table of the file.

```
   +---------------------------------------------------------------+
48 |                         PACB DST nr.                          | 60
   |---------------------------------------------------------------|
49 |                    PACB offset (DST-rel.)                     | 61
```

```
   |----------------------------------------------------------------|
50 |                        LACB DST nr.                            | 62
   |----------------------------------------------------------------|
51 |                   LACB offset (DST-rel.)                       | 63
   |----------------------------------------------------------------|
52 |               ACB offset (Stack-DST-rel.)                      | 64
   |----------------------------------------------------------------|
53 |                DB offset (Stack-DST-rel.)                      | 65
   |----------------------------------------------------------------|
54 |          Stack-DST-rel location of PXFILE CBTAB                | 66
   |----------------------------------------------------------------|
55 |          CBTAB-rel vector table entry address                  | 67
   |----------------------------------------------------------------|
```

The following identifiers are used when referring to an ACB:

```
     (ACBSIZE)     = ACB.(2:14)#,          size in words
     ACBFNUM       = ACB(1).(8:8)#,         file number
     ACBNAME       = ACB(2)#,               file name
     ACBNAME1      = ACBDBL(1)#,            file name - first half
     ACBNAME2      = ACBDBL(2)#,            file name - second half
     ACBFOPTIONS   = ACB(6)#,               FOPTIONS
     ACBAOPTIONS   = ACB(7)#,               AOPTIONS
     ACBRSIZE      = ACB(8)#,               record size (bytes)
     ACBBSIZE      = ACB(9)#,               block size (words)

     ACBCTL        = ACB(11)#,              carriage control word
     ACBLSTATE     = ACB(12)#,              local state flags
     ACBEOF        = ACBLSTATE.(1:1)#,      end of file sensed
     ACBLPCTL      = ACBLSTATE.(2:2)#,      page and line control
     ACBPAGECTL    = ACBLSTATE.(2:1)#,      page control
     ACBLINECTL    = ACBLSTATE.(3:1)#,      line control
     ACBSTREAM     = ACBLSTATE.(4:1)#,      stream I/O
     ACBFKEYS      = ACBLSTATE.(5:1)#,      restore function keys
     ACBXMITCRLF   = ACBLSTATE.(6:1)#,      transmit CR,LF to user
     ACBTBLOCK     = ACBLSTATE.(7:1)#,      disable block mode
     ACBBINARYIO   = ACBLSTATE.(8:1)#,      8-bit terminal transfers
     ACBCARRIAGE   = ACBLSTATE.(9:1)#,      carriage control flag
     (ACBDEFBLOCK) = ACBLSTATE.(10:1)#,     default blocking
     ACBREADCODE   = ACBLSTATE.(11:4)#,     input EOF check
     ACBREADTYPE   = ACBLSTATE.(11:2)#,     input EOF type
     ACBREADMODE   = ACBLSTATE.(13:2)#;     input EOF mode

     ACBMODW       = ACB(13)#,              mode word
     ACBMODE       = ACBMODW.(0:8)#,        mode setting
     ACBTAPEERROR  = ACBMODW.(4:1)#,        report recovered tape error
     ACBINHIBCRLF  = ACBMODW.(5:1)#,        inhibit terminal CR/LF
     ACBQUIESCE    = ACBMODW.(6:1)#,        critical output verify
     ACBSTOPCHAR   = ACBMODW.(8:8)#,        terminal stop character
     ACBERROR      = ACB(14)#,              error code
     ACBTLOG       = ACB(15)#,              last I/O transmission log
     ACBFPTR       = ACBDBL(08)#,           current record number
     ACBBLK        = ACBDBL(09)#,           current variable block
```

```
ACBRTFRCT      = ACBDBL(10)#,           logical record tfr count
ACBBTFRCT      = ACBDBL(11)#,           block transfer count
ACBHIBLK       = ACBDBL(12)#,           highest block started
ACBFCB         = ACB(26)#,              FCB vector

ACBSHCNTS      = ACB(28)#,              LACB counts
ACBSHCNTIN     = ACBSHCNTS.(0:8)#,      # of Read LACB'S
ACBSHCNT       = ACBSHCNTS.(8:8)#,      # of LACB'S
ACBSTATW       = ACB(29)#,              access class, status, etc.
ACBBREAK       = ACBSTATW.(1:1)#,       break ($STDIN/LIST only)
ACBDTYPE       = ACBSTATW.(2:6)#,       device type
ACBACCCL       = ACBSTATW.(2:3)#,       device access class
ACBSUBCL       = ACBSTATW.(5:3)#,       device sub-class
ACBSTATUS      = ACBSTATW.(8:8)#,       last logical I/O status
ACBQSTATUS     = ACBSTATW.(8:5)#,       qualifying status part
ACBGSTATUS     = ACBSTATW.(13:3)#,      general status part
ACBGSTW        = ACB(30)#,              global state flags
ACBNOWAITEOF   = ACBGSTW.(0:1)#,        EOF advanced?
ACBNOWAITMODE   = ACBGSTW.(1:1)#,       last I/O: 0  = read, 1  = write
ACBABORTREAD   = ACBGSTW.(2:1)#,        abort broken re-read?
ACBNEWEOF      = ACBGSTW.(3:1)#,        EOF advanced - tape file
ACBSAVEEOFS    = ACBGSTW.(4:2)#,        for saving ACBEOFS
ACBEOFS        = ACBGSTW.(6:2)#,        EOF flags - :EOD/:
ACBBLKFACT     = ACBGSTW.(8:8)#,        records/block
ACBBUFX        = ACB(31)#,              buffer data & misc. flags
ACBPRIV        = ACBBUFX.(0:1)#,        privileged access only
ACBHIT         = ACBBUFX.(1:1)#,        buffer hit flag
ACBCURRBUF     = ACBBUFX.(4:4)#,        current buffer nr.
ACBTAPEDISP    = ACBBUFX.(8:8)#,        tape displacement
ACBNUMBUFS     = ACBBUFX.(12:4)#,       number of buffers less 1
ACBBUFUSED     = ACB(32)#,              used block word count
ACBBUFSIZE     = ACB(33)#,              buffer size (words)
ACBXXXX        = ACB(34)#,              spare
ACBFMAVTX      = ACB(35)#,              FMAVT index
ACBVDADDR      = ACB(36)#,              volume table index
ACBDNTD        = ACB(37)#,              type & disposition
ACBDNTYPE      = ACBDNTD.(0:8)#,        name type for dir. search
ACBDISP        = ACBDNTD.(8:8)#,        file disposition
ACBAMLD        = ACB(38)#,              access mask & LDEV
ACBACCESS      = ACBAMLD.(0:8)#,        access mask
ACBDADDR       = ACBAMLD.(8:8)#,        logical device number

ACBSPFL        = ACB(39)#,              spool control flags
ACBSPOOLED     = ACBSPFL.(0:1)#,        spooled device flag
ACBSPOOLIO     = ACBSPFL.(0:2)#,        spooled IN/OUT
ACBSPSQ        = ACBSPFL.(2:2)#,        squeeze flags
ACBSPSQZ       = ACBSPFL.(2:1)#,        file squeezed
ACBSPRSQ       = ACBSPFL.(3:1)#,        request to sqz
ACBSPDSQ       = ACBSPFL.(4:1)#,        squeeze just done
ACBSPVDEV      = ACBSPFL.(8:8)#,        spooled virtual device
ACBSPTYRC      = ACB(40)#,              spooled dev type/recsize
ACBSPTYPE      = ACBSPTYRC.(0:6)#,      spooled dev type
ACBSPREC       = ACBSPTYRC.(6:10)#,      spooled dev rec size
ACBSPFOPT      = ACB(41)#,              spooled dev FOPTIONS
ACBSPAOPT      = ACB(42)#,              spooled dev AOPTIONS
```

```
ACBSPXDDX     = ACB(43)#,          IDD/ODD index
ACBNOWAITDA   = ACBDBL(22)#,       No-wait disk address
ACBNOWAITLDEV = ACB(27)#,
```

Discussion:

ACBABORTREAD This flag is used to abort a broken terminal re-read. The flag is set via the ABORT parameter to FUNBREAK. If the flag is set then the READ PENDING message will be aborted along with the re-read. This feature is needed to handle the BREAK...:ABORT, etc. situation.

ACBACCCL This is the access class part of the device type number. The following are legal values:

0 - direct (e.g. disc)
1 - serial input (e.g. card reader)
2 - parallel input/output (e.g. terminal)
3 - serial input/output (e.g. mag tape)
4 - serial output (e.g. line printer)

ACBACCESS This is the access bit map for the file. The following are the bit definitions of this eight-bit field:

(0:1) - unused
(1:1) - unused
(2:1) - read
(3:1) - append
(4:1) - write
(5:1) - lock
(6:1) - execute
(7:1) - save

This access security is determined by the ACCCHECK intrinsic and enforced by the file system.

ACBAOPTIONS This is the AOPTIONS in effect for this file access.

ACBBINARYIO This bit controls full eight bit transfers on the 2644 page mode terminal. It is adjusted by FCONTROL(26) and FCONTROL(27).

ACBBLK This is the block number of the current variable record format block. Applicable iff the record format is variable.

ACBBLKFACT This is the blocking factor for the file. It is the number of records in a block. Legal values range from 1 to 255.

ACBBREAK This is the break mode flag. It is applicable iff the ACB is for $STDIN or $STDLIST. If set it means that the BREAK key has been hit and that the CI should have high priority access to the ACB. The flag will be

	cleared when a RESUME or ABORT is issued.
ACBBSIZE	This is the block size, in words, of the file.
ACBBTFRCT	This is the total number of blocks transferred to and from the file. The initial value is 0D.
ACBBUFUSED	This is the word index, relative to the base of the block, for the selected record within the block. This is applicable iff the file access is buffered.
ACBCARRIAGE	This bit signifies that the file has carriage control. It is the same as the carriage control bit in ACBFOPTIONS if the file is spooled. If not spooled, the bit is zero, and IOMOVE will pass the FWRITE carriage control parameter directly to the driver rather than imbedding it as the first character of the output record.
ACBCTL	This is the CONTROL parameter from the last FWRITE. This value is pertinent iff the file was opened with carriage control.
ACBCURRBUF	This is the buffer number (0-relative) containing the most recently referenced record. Applicable iff the file access is buffered.
ACBDADDR	This is the logical device number of the file. For a disc file this is the logical device number of the first extent.
ACBDEFBLOCK	This bit signifies that the file is to be accessed with default blocking. The bit is initialized from the FOPEN stateword STATE. It does not need to be in the ACB; it is mentioned here only to signify that the bit is effectively used due to the way ACBLSTATE is initialized from STATE.
ACBDISP	This is the file close disposition derived from the FOPEN call. The only way this can be specified is via a file equation. The legal values are the same as those for FCLOSE.
ACBDNTYPE	This is the file reference format type number and is derived from the FOPEN call. The following are legal values: 0 - full name 1 - account name absent 2 - group and account name absent 3 - null name This information is needed by FRENAME.
ACBDTYPE	This is the device type number of the file. The following are legal values (octal):

0 - moving head disc
1 - fixed head disc
7 - foreign disc
10 - card reader
11 - paper tape reader
20 - terminal
24 - card reader/interpreter/punch
26 - SSLC
27 - programmable controller
30 - magnetic tape
31 - serial disc
40 - line printer
41 - card punch
42 - paper tape punch
43 - CALCOMP 500 plotter
44 - CALCOMP 600 plotter
45 - CALCOMP 700 plotter

ACBEOF	This bit is set when EOF has been sensed.
ACBEOFS	This is the type of EOF detected on $STDIN(X). This field consists of two bits: (0:1) - super colon (i.e. EOF for $STDINX) (1:1) - regular colon (i.e. EOF for $STDIN) Applicable for multi-access to $STDIN(X) only.
ACBERROR	This is the error number for the file. It is used by all intrinsics except FOPEN. When an error is detected the error number is placed in this cell. The error number is cleared at the beginning of each callable intrinsic except FCHECK (which reads it).
ACBFCB	This is the FCB vector for the file. Applicable only to disc files.
ACBFKEYS	This bit controls the definition of the f1 and f2 function keys on the 2644 page mode terminal; it is adjusted by FCONTROL(32) and FCONTROL(33). (Obsolete function)
ACBFNUM	File number, range from 1 to 255. Used mostly for calling routines that access things such as labels by file number.
ACBFOPTIONS	This is the FOPTIONS in effect for this file access.
ACBFPTR	This is the sequential access record pointer; it contains the next sequential record number. The initial value is 0D. This value is used only by the FREAD, FWRITE and FUPDATE intrinsics. However the value is maintained by all data transferring file system intrinsics.

ACBFMAVTX	This is the entry index into the file multi-access vector table (FMAVT). This is valid iff the file access is multi-access.
ACBGSTATE	These are miscellaneous state flags. These are "global" in nature in that they are the same for all accessors in a multi-access environment. The constituent bits are described individually.
ACBGSTATUS	This is the general part of the last I/O status for the file. The following are the legal values: 0 - pending 1 - successful 2 - end of file 3 - unusual condition 4 - irrecoverable error
ACBHIBLK	This is the highest block number for which an anticipatory read has been issued, and is applicable iff the file access is buffered. The initial value is -1D.
ACBHIT	This is the buffer hit flag. If set it indicates that the last read or write request was serviced without any physical I/O required. This flag is used only for performance measurement. The code which manipulates it is optional to the file system, and is controlled by compiler toggle X3.
ACBINHIBCRLF	This bit controls the termination of lines written to the terminal. If not set then each line is terminated with a CR and LF; if set then no line termination characters are used. This bit is valid iff the file is a terminal file; it is adjusted by FSETMODE.
ACBLINECTL	This is the line control bit. If not set then each line is post-spaced; if set then each line is pre-spaced. This bit is used by line printers and terminals only. It is adjusted by FCONTROL(1) and FWRITE with the appropriate carriage control.
ACBLPCTL	This are the line and page control bits, which are described separately.
ACBLSTATE	These are miscellaneous state flags. They are "local" in nature in that they may be different for each accessor in a multi-access environment. Bits (9:6) are initialized from the stateword local variable called STATE in FOPEN; the ten remaining bits are initialized individually. The constituent bits are described individually.

ACBMODE	These are miscellaneous mode flags. The constituent bits are described individually.
ACBNAME	This is the local file name. The name is eight bytes in length with trailing blanks added.
ACBNEWEOF	This flag when set indicates that a new tape mark should be written before the tape is rewound or backspaced. Applicable only to mag tape files.
ACBNOWAITEOF	This bit is used to save the value of the local EOF advanced flag NEWEOF in IOMOVE between the I/O initiation and I/O completion calls. This flag is applicable iff the file is accessed in no-wait I/O mode.
ACBNOWAITMODE	This cell is used to save the I/O mode between no-wait I/O initiation and completion calls. If the bit is set then the last I/O request was a write; otherwise it was a read. This cell is pertinent iff the file is accessed in no-wait I/O mode.
ACBNUMBUFS	This is the number of buffers, less one, used for the file access. Applicable iff the file access is buffered.
ACBPAGECTL	This is the page control bit. If not set then a page is assumed to consist of 60 lines (auto page eject); if set then a page is assumed to consist of 66 lines (no auto page eject). This is used primarily for line printers but is also valid for terminals; these are the only devices for which this is valid. This bit is adjusted by FCONTROL(1) and FWRITE with the appropriate carriage control.
ACBPRIV	This flag when set indicates that the file is privileged in that it has a negative file code; the user must be in privileged mode to access it.
ACBQSTATUS	This is the qualifying part of the last I/O status for the file. The values are unique for each general status part. See I/O System IMS for all legal values.
ACBQUIESCE	This bit controls critical output verification. If set, buffered output is guaranteed to have been written to the device when control is returned to the user. This bit is adjusted by FSETMODE.
ACBREADCODE	This field consists of the input EOF checking type and mode, and is used to generate the P1 parameter to ATTACHIO. These fields are described individually.
ACBREADMODE	This field controls the input EOF checking mode. It is 00 for reading $STDIN, 01 for reading $STDINX, and 10 for the command interpreter.

ACBREADTYPE	This field controls the input EOF checking type. It is 01 for JOBs, 10 for SESSIONs, and 00 for DATA.
ACBRSIZE	This is the file's record size in positive bytes.
ACBRTFRCT	This is the total number of records transferred to and from the file. The initial value is 0D.
ACBSAVEEOFS	This field is used to save the contents of ACBEOFS during BREAK mode processing.
ACBSHCNT	This is the total number of LACBs that exist for this PACB. Valid iff the file access is multi-access.
ACBSHCNTIN	This is the total number of input-only LACBs that exist for this PACB. Valid iff the file access is multi-access.
ACBSHCNTS	This is the total LACB and total input-only LACB counts, each of which is described separately.
ACBSIZE	This is the size, in words, of the complete ACB. It includes the buffering extension, if present.
ACBSPAOPT	This is the AOPTIONS for the spooled device. Applicable iff the file access is to a spooled device.
ACBSPFOPT	This is the FOPTIONS for the spooled device. Applicable iff the file access is to a spooled device.
ACBSPOOLED	This is the spooled device flag. If set then the file access is to a spooled device.
ACBSPOOLIO	This field is a combination of the spooled device flag and the input/output mode of the spooled device. Legal values are: 00 - not spooled 01 - illegal 10 - input spooling 11 - output spooling
ACBSPREC	This is the record size, in bytes, of the spooled device. Applicable iff the file access is to a spooled device.
ACBSPTYPE	This is the device type (from the LDT) of the spooled device. Applicable iff the file access is to a spooled device.
ACBSPTYRC	This cell contains the spooled device type and record size, which are described separately.
ACBSPVDEV	This is the logical device number of the spooled device. Applicable iff the file access is to a spooled

device.

ACBSPXDDX This is the index into the IDD or ODD for a spoolfile. Applicable iff the file access is to either a spooled device or a spoolfile.

ACBSTATUS This is the last I/O status for the file. It comes from the I/O status part of the IOCB returned by ATTACHIO. Not all ATTACHIO calls update this cell.

ACBSTOPCHAR This is the record termination character used for terminal reads. This character can be changed via FCONTROL(25).

ACBSTREAM This bit signifies inter-block garbage for disc files. If set, the block size is a multiple of 128 words and therefore there is no garbage data between blocks. This fact is used to improve multi-record I/O by mapping the request into as few ATTACHIOs as possible.

ACBSUBCL This is the sub-class part of the device type number. The sub-class is unique for each access class. The following are the legal sub-class values for each device class:

- 0 - direct
  - 0 - moving head disc
  - 1 - fixed head disc
  - 7 - foreign disc
- 1 - serial input
  - 0 - card reader
  - 1 - paper tape reader
- 2 - parallel input/output
  - 0 - terminal
  - 4 - card reader/punch
  - 6 - SSLC
  - 7 - programmable controller
- 3 - serial input/output
  - 0 - mag tape
  - 7 - serial disc
- 4 - serial output
  - 0 - line printer
  - 1 - card punch
  - 2 - paper tape punch
  - 3 - CALCOMP 500 plotter
  - 4 - CALCOMP 600 plotter
  - 5 - CALCOMP 700 plotter

ACBTAPEDISP This number is used to keep track of the difference or displacement between the physical and logical tape locations. The tape could be mispositioned due to pre-reads and this variable is used to properly backspace the tape before an FWRITE, FSPACE, FCONTROL(6) or FCLOSE(DISP=3).

ACBTAPEERROR	This bit controls the reporting of recovered mag tape errors. If not set the recovered errors are not reported to the user; if set then recovered errors are reported to the user by returning CCL and error number 39. Valid iff the file is a mag tape file. This bit is adjusted by FSETMODE.
ACBTBLOCK	This bit controls block mode transfers on the 2644 page mode terminal. This bit is adjusted by FCONTROL(28) and FCONTROL(29).
ACBTLOG	This is the last I/O transmission log for the file. It comes from the I/O transmission log part of the IOCB returned by ATTACHIO. Not all ATTACHIO calls update this cell.
ACBVDADDR	This is the volume table index for the file. Applicable iff the file is a disc file.
ACBXMITCRLF	This bit controls CR and LF insertion into the user buffer on the 2644 page mode terminal. This bit is adjusted by FCONTROL(30) and FCONTROL(31).

If present, the PACB buffering extension contains from one to sixteen block buffers each having the following format:

```
  0                   7      10  11  12  13  14  15
 ----------------------------------------------------
 |                  IOQ entry index                  |  0  BLKIOQX
 |---------------------------------------------------|
 |                         | U | R | D | W | M | P |  1  BLKFLAGW
 |---------------------------------------------------|
 |                   IOCB - Status                   |  2  BLKLSTAT
 |---------------------------------------------------|
 |              IOCB - Transmission log              |  3  BLKTLOG
 |---------------------------------------------------|
 |                                                   |  4  BLKBLOCK
 |                   Block number                    |
 |                                                   |  5
 |---------------------------------------------------|
 |  Block log. device no.|                           |  6  BLKDADDR
 |-----------------------                            |
 |                Block sector number                |  7
 |---------------------------------------------------|
 |                                                   |  8  BLKEXTBASE
 |                 Block Extent Base                 |
 |                                                   |  9
 |---------------------------------------------------|
 |                 Block Extent Size                 |  10 BLKEXTSIZE
 |---------------------------------------------------|
 |                     Not Used                      |  11
 |---------------------------------------------------|
 |                                                   |  12 BLKBUFFER
 |                                                   |
 |                                                   |
 |                      Buffer                       |
 |                                                   |
 |                                                   |
 |                                                   |
 ----------------------------------------------------
```

Other identifiers used:

```
     BLKIOCB          = BLKDBL(1)#,        IOCB
     (BLKLDEV)        = BLK(6).(0:8)#,     block logical device number
     BLKFLAGS         = BLK(1).( 8:8)#,    block I/O flags
     BLKUNALLOCEXT    = BLK(1).(10:1)#,    block from un-allocated extent
     BLKREVERSE       = BLK(1).(11:1)#,    block for tape FREADBACKWARDS
     BLKDONTWAIT      = BLK(1).(12:1)#,    I/O status not checked.
     BLKIOOUT         = BLK(1).(13:1)#,    last I/O was write?
     BLKDIRTY         = BLK(1).(14:1)#,    buffer modified?
     BLKIOPEND        = BLK(1).(15:1)#,    I/O in progress?
     BLKIOCOMP        = BLK(1).(14:2)#,    I/O complete - not dirty
```

Discussion:

BLKBLOCK	This is the block number of the data contained in the buffer. A value of -1D indicates that the buffer is empty.
BLKBUFFER	If ACB buffering is used, this is the buffer location. When system buffers were used, the buffer location was given by BLKSYSBUFX and BLKSYSBUFDISP.
BLKDADDR	This is the block's logical device and sector number.
BLKDIRTY	This flag is set if the contents of the buffer has been modified. When the block buffer is reused this flag is checked to see if the block needs to be written to the device.
BLKDONTWAIT	This bit is on if the buffer's I/O was completed and the BLKIOQX and pending bits cleared, but the status of the I/O was not checked. This is done to free valuable DRQ entries. If the bit is on, then BLKLSTAT must be checked before using the block.
BLKEXTBASE	This is the sector address of the base of the extents in which the block resides. It is used for I/O disk caching.
BLKEXTSIZE	This is the size, in sectors, of the extent in which the block resides. Also used for I/O disk caching.
BLKFLAGS	These are the miscellaneous flags associated with the block, which are described separately.
BLKIOCB	This is the IOCB returned by the I/O system when the block I/O has completed. On a blocked I/O request this is obtained from the ATTACHIO call; on an unblocked I/O request this is obtained from WAITFORIO.
BLKIOCOMP	This is the buffer modified flag (BLKDIRTY) and the I/O in progress flag (BLKIOPEND), which are described separately. This field is usually interrogated to see if it contains the value 2, which means that the buffer has been modified but not yet written to the device.
BLKIOOUT	This is the mode of the I/O operation for the block. It is set by a write and cleared by a read.
BLKIOPEND	This is the I/O in progress flag. It is set if the I/O is pending; it is cleared when the I/O has completed.
BLKIOQX	This is the IOQ index of the unblocked I/O request for the block. It is used as the argument to WAITFORIO, which insures the completion of the I/O request.
BLKLDEV	This is the logical device number of the block.

BLKLSTAT	The I/O status part of the IOCB consists of the PCB number and the error code for the completed I/O request.
BLKREVERSE	This bit is not currently used but has been reserved f FREADBACKWARDS (reading a tape backwards) to a buffered fil which is not currently supported.
BLKTLOG	The transmission log part of the IOCB is the number of words or bytes transferred by the the I/O request.
BLKUNALLOCEXT	This bit is on if the block in this buffer was read from unallocated extent. In this case, the extent was not allocat and the buffer was simply flushed with fill characters. T block must be allocated before writing to i

3.2.7 File Control Block (FCB)

The FCB coordinates access to a file on a sharable device. At present the only sharable device is a disc, so only disc files have FCBs.

The information contained in an FCB is derived from the file label. The FCB is used to hold this information, rather than the file label, since it can be accessed more quickly.

The FCB can be contained in a stack when first created. If another process opens the file, the FCB will be moved to a system data segment (which will be created if it doesn't already exist) so that the first process' entire stack need not be present when the second process is dealing with the file. The number of a data segment containing a list of numbers of shared file system data segments is kept in system global location 1076 octal. The size of the FCB depends on the maximum number of extents specified at FOPEN; there are 44 (octal) words plus two per extent. There will be at least one extent, since the file label always exists in the first extent. The FCB extent map is in terms of logical device and sector number. The extent map in the file label is in terms of volume rather than logical device; the map is converted by VTABTOLDEV when the label is read, and converted back by LDEVTOVTAB when the label is written to disk.

The FCB has the following format:

```
     0  1  2  3      7  8      12 13 14 15
    --------------------------------------------------
 0 |  1  |            Complete FCB size             | 0
   |------------------------------------------------|
 1 |               New FCB vector                   | 1  FCBNEWFCBV
   |------------------------------------------------|
 2 |                 FOPTIONS                       | 2  FCBFOP-
   |------------------------------------------------|    TIONS
 3 |           Device specification                 | 3  FCBDEVICE
   |------------------------------------------------|
 4 | Prev. lock| Dev. type  | C |     |Device subtype | 4
   |------------------------------------------------|
 5 | No. opens for output  | No. opens for any mode | 5
   |------------------------------------------------|
 6 |            Creator ACB vector                  | 6  FCBACB
   |------------------------------------------------|
 7 |               RIN number                       | 7  FCBRIN
   |------------------------------------------------|
 8 |            Exclusive status                    | 10 FCBEXC-
   |------------------------------------------------|    STAT
 9 |        Private volume information              | 11 FCBPVINFO
   |------------------------------------------------|
10 |                                                | 12 FCBFLIM
   |               File limit                       |
11 |                                                | 13
   |------------------------------------------------|
```

```
12 |                                               | 14  FCBIMAGE
   |              Reserved for IMAGE               |
13 |                                               | 15
   |-----------------------------------------------|
14 |                                               | 16  FCBEOF
   |              End of data pointer              |
15 |                                               | 17
   |-----------------------------------------------|
16 | No. user labels written | No. user labels avail. | 20  FCBUSERLBL
   |-----------------------------------------------|
17 |           Extent size in sectors              | 21  FCBEXTSIZE
   |-----------------------------------------------|
18 |    Blocking factor      |   Sectors per block   | 22
   |-----------------------------------------------|
19 | Sector offset to data   | Disp | No. extents - 1 | 23
   |-----------------------------------------------|
20 |         Last extent size in sectors           | 24  FCBLAST-
   |-----------------------------------------------|     EXTSIZE
21 |                         | No. opens input mode  | 25
   |-----------------------------------------------|
22 | Group name - 1st char.  | Group name - 2nd char. | 26  FCBGN
   |-----------------------------------------------|
23 | Group name - 3rd char.  | Group name - 4th char. | 27
   |-----------------------------------------------|
24 | Group name - 5th char.  | Group name - 6th char. | 30
   |-----------------------------------------------|
25 | Group name - 7th char.  | Group name - 8th char. | 31
   |-----------------------------------------------|
26 |  Acct name - 1st char.  |  Acct name - 2nd char. | 32  FCBAN
   |-----------------------------------------------|
27 |  Acct name - 3rd char.  |  Acct name - 4th char. | 33
   |-----------------------------------------------|
28 |  Acct name - 5th char.  |  Acct name - 6th char. | 34
   |-----------------------------------------------|
29 |  Acct name - 7th char.  |  Acct name - 8th char. | 35
   |-----------------------------------------------|
30 |                                               | 36  FCBSTART
   |           Start of file block number          |
31 |                                               | 37
   |-----------------------------------------------|
32 |                                               | 40  FCBEND
   |    Current number of data blocks in the file  |
33 |                                               | 41
   |-----------------------------------------------|
34 |                                               | 42  FCBNUM-
   |  Number of open and close records (message file) |     OPENCLSREC
35 |                                               | 43
   |-----------------------------------------------|
36 |  Logical device number  |                     | 44  FCBEXTMAP
   |-------------------------                      |
37 |           First extent sector number          | 45
   |-----------------------------------------------|
   |                        .                      |
   |                        .                      |
   |                        .                      |
```

```
|-----------------------------------------------------|
|  Logical device number  |                           |
|-------------------------                            |
|                Last extent sector number            |
-------------------------------------------------------
```

Other identifiers used:

```
FCBSIZE          = FCB.(2:14)#,      size in words
FCBLKST          = FCB(4).(0:2)#,    previous lock state
FCBDTYPE         = FCB(4).(2:6)#,    device type
FCBCRUNCH        = FCB(4).(8:1)#,    pending crunch disposition
FCBSUBTYPE       = FCB(4).(12:4)#,   device subtype
FCBOCNTOUT       = FCB(5).(0:8)#,    no. accessors - output
FCBOCNT          = FCB(5).(8:8)#,    no. accessors
FCBLBLEOF        = FCB(16).(0:8)#,   no. labels written
FCBLBL           = FCB(16).(8:8)#,   no. labels available
FCBBLKFACT       = FCB(18).(0:8)#,   blocking factor
FCBSECTPBLK      = FCB(18).(8:8)#,   sectors per block
FCBSECTOFF       = FCB(19).(0:8)#,   sector offset to data
FCBDISP          = FCB(19).(8:3)#,   pending disposition
FCBNUMEXTS       = FCB(19).(11:5)#,  no. extents less 1
FCBOCNTIN        = FCB(21).(8:8)#,   no. acccessors - input
FCBLABEL         = FCBDBL(18)#,      label LDEV and sector
FCBLDEV          = FCB(36).(0:8)#,   label LDEV
```

Discussion:

FCBACB	This is the vector of the ACB that was created at the same time as the FCB. This is used in conjunction with FCBNEWFCBV when relocating the FCB.
FCBAN	This is the account name of the file. It is eight bytes in length with trailing blanks added.
FCBBLKFACT	This is the blocking factor of the file. It is the number of logical records in a physical block. Legal values range from 1 to 255.
FCBDEVICE	This specifies the device on which the file resides. If it is positive then it represents a logical device number; if negative it represents a (negative) device class index.
FCBDISP	This is the pending FCLOSE disposition for the file. Legal values are: 0 - no change 1 - save permanent 2 - save temporary and rewind 3 - save temporary but do not rewind 4 - release

	7 - invalid file (file label access error)
FCBCRUNCH	This bit governs if space will be returned beyond the EOF u the last FCLOSE of the file. 0 - no change 1 - return space beyond EOF
FCBDTYPE	This is the device type number of the first extent of the file. See ACBDTYPE for a list of legal values.
FCBEND	Block number of the file's EOF, relative to FCBSTART.
FCBEOF	This is the end-of-file pointer for the file. It is a double integer representing the number of records in the file. It can also be viewed as the record number of the next record past EOF.
FCBEXCLSTAT	This is the exclusive status of the file access. If -1 then the file is being accessed exclusively; otherwise it is the number of semi-exclusive accessors.
FCBEXTMAP	This is the extent map of the file. The number of extents is specified by FCBNUMEXTS; a 0D extent descriptor indicates that the extent has not been allocated.
FCBEXTSIZE	This is the extent size, in sectors, of the file. All extents in the file except possibly the last have this size. This is a logical value, and legal values range from 1 to 65535 sectors. This restricts the maximum file size to 2097120 sectors (268,431,360 words).
FCBFLIM	This is the end-of-space pointer for the file. It is a double word integer representing the maximum number of records (fixed length record format) or blocks (undefined or variable length record format) in the file.
FCBFOPTIONS	This is the FOPTIONS in effect for the file.
FCBGN	This is the group name of the file. It is eight bytes long with trailing blanks added.
FCBLABEL	This is the logical device and sector number of the file label, which is the same as the first extent descriptor.
FCBLASTEXTSIZE	This is the size, in sectors, of the last extent in the file. If the file has one extent then this is the same as FCBEXTSIZE; otherwise this value may be different from FCBEXTSIZE. This is the size of the last physical extent for the file; it is not the size of the last allocated extent.

FCBLBL	This is the number of user labels allocated for the file. Since each label is a sector long, this is also the number of sectors allocated for user labels.
FCBLBLEOF	This is the end-of-data pointer for the user labels. It is analogous to FCBEOF in that it represents the number of labels written. The initial value is 0.
FCBLDEV	This is the logical device number of the first extent of the file.
FCBLKST	This is the previous lock state of the file and is derived from the file label. Legal values are: 0 - no accessors 1 - read 2 - write 3 - read/write
FCBNEWFCBV	This is the vector of the new FCB for the file. It is used in conjunction with FCBACB to move the FCB to a system (shared FCB) control block table when the second accessor is established. If this value is zero then there is no new FCB; if non-zero then a new FCB has been created.
FCBNUMEXTS	This is the maximum number of extents, less one, allowed for the file. It is not the number of extents presently allocated, which is always determined by counting non-zero entries in the extent map.
FCBNUMOPENCLSREC	Number of open and close records in the message file.
FCBOCNT	This is the number of accessors for the file. Alternatively it can be viewed as the number of PACBs created for the file.
FCBOCNTIN	This is the number of file accessors having input access.
FCBOCNTOUT	This is the number of file accessors having output access.
FCBRIN	This is the RIN number used to support dynamic locking (i.e. FLOCK and FUNLOCK) for the file. If there is no dynamic locking then this number is zero.
FCBSECTOFF	This is the sector offset from the file label to the first block of the file. This is not necessarily equal to FCBLBL+1 since an integral number of blocks are allocated for the file and user labels.

FCBSECTPBLK	This is the number of sectors in a block for the file.
FCBSIZE	This is the size, in words, of the complete FCB. It includes the extent map.
FCBSTART	Block number of the file's start, excluding the file label block.
FCBSUBTYPE	This is the device sub-type number of the first extent.
FCBUSERLBL	This field describes the user labels for the file. It consists of FCBLBL and FCBLBLEOF, described separately.

3.3 File Label (FLAB)

The file label has the following format:

```
  0  1  2  3        7  8        12  13  14  15
 -----------------------------------------------------
 | File name - 1st char.  | File name - 2nd char.  |  0  FLLOCNAME
 |-------------------------------------------------|
 | File name - 3rd char.  | File name - 4th char.  |  1
 |-------------------------------------------------|
 | File name - 5th char.  | File name - 6th char.  |  2
 |-------------------------------------------------|
 | File name - 7th char.  | File name - 8th char.  |  3
 |-------------------------------------------------|
 | Group name - 1st char. | Group name - 2nd char. |  4  FLGRPNAME
 |-------------------------------------------------|
 | Group name - 3rd char. | Group anme - 4th char. |  5
 |-------------------------------------------------|
 | Group name - 5th char. | Group name - 6th char. |  6
 |-------------------------------------------------|
 | Group name - 7th char. | Group name - 8th char. |  7
 |-------------------------------------------------|
 | Acct name - 1st char.  | Acct name - 2nd char.  | 10  FLACCTNAME
 |-------------------------------------------------|
 | Acct name - 3rd char.  | Acct name - 4th char.  | 11
 |-------------------------------------------------|
 | Acct name - 5th char.  | Acct name - 6th char.  | 12
 |-------------------------------------------------|
 | Acct name - 7th char.  | Acct name - 8th char.  | 13
 |-------------------------------------------------|
 | Creator name - 1st char.| Creator name - 2nd char.| 14  FLUSERID
 |-------------------------------------------------|
 | Creator name - 3rd char.| Creator name - 4th char.| 15
 |-------------------------------------------------|
 | Creator name - 5th char.| Creator name - 6th char.| 16
 |-------------------------------------------------|
 | Creator name - 7th char.| Creator name - 8th char.| 17
 |-------------------------------------------------|
 | Lockword - 1st char.   | Lockword - 2nd char.   | 20  FLLOCKWORD
 |-------------------------------------------------|
 | Lockword - 3rd char.   | Lockword - 4th char.   | 21
 |-------------------------------------------------|
 | Lockword - 5th char.   | Lockword - 6th char.   | 22
 |-------------------------------------------------|
 | Lockword - 7th char.   | Lockword - 8th char.   | 23
 |-------------------------------------------------|
 |                                                 | 24  FLSECMX
 |                 Security matrix                 |
 |                                                 | 25
 |-------------------------------------------------|
 | Reserved               |              | SR | S  | 26
 |-------------------------------------------------|
```

```
|                 Creation date                     | 27  FLCREATE
|---------------------------------------------------|
|               Last access date                    | 30  FLLASTACC
|---------------------------------------------------|
|            Last modification date                 | 31  FLLASTMOD
|---------------------------------------------------|
|                  File code                        | 32  FLFILECODE
|---------------------------------------------------|
|                  FCB vector                       | 33  FLFCBVECT
|---------------------------------------------------|
| S | R | L | X | Subtype |     Disc type    | R/W  | 34  FLLOCK
|---------------------------------------------------|
| No. user labels written | No. user labels avail.  | 35  FLUSERLBL
|---------------------------------------------------|
|                                                   | 36  FLFLIM
|            File limit in blocks                   |
|                                                   | 37
|---------------------------------------------------|
|                                                   | 40
{WNER } PRIVATE VOLUME INFORMATION------------------|
{ohn  }                                             | 41
|---------------------------------------------------|
|                  Checksum                         | 42  FLCHECKSUM
|---------------------------------------------------|
|                  Cold load ID                     | 43  FLCLID
|---------------------------------------------------|
|                  FOPTIONS                         | 44  FLFOPTIONS
|---------------------------------------------------|
|              Record size in bytes                 | 45  FLRECSIZE
|---------------------------------------------------|
|              Block size in words                  | 46  FLBLKSIZE
|---------------------------------------------------|
|      Sector offset      |     |  No. extents -1   | 47
|---------------------------------------------------|
|          Last extent size in sectors              | 50  FLLASTEXT-
|---------------------------------------------------|       SIZE
|            Extent size in sectors                 | 51  FLEXTSIZE
|---------------------------------------------------|
|                                                   | 52  FLEOF
|              End of data pointer                  |
|                                                   | 53
|---------------------------------------------------|
|   Volume table index    |                         | 54  FLEXTMAP
|-------------------------                          |
|            1st extent sector number               | 55
|---------------------------------------------------|
|                        .                          |
|                        .                          |
|                        .                          |
|---------------------------------------------------|
|   Volume table index    |                         |
|-------------------------                          |
|           Last extent sector number               |
|---------------------------------------------------|
|                        .                          |
```

```
|                        .                         |
|                        .                         |
|--------------------------------------------------|
|                                                  |154  FLALLOCTIME
|              File allocation time                |
|                                                  |155
|--------------------------------------------------|
|              File allocation date                |156  FLALLOCDATE
|--------------------------------------------------|
|                        .                         |
|--------------------------------------------------|
|                                                  | 160 FLSTART
|             Start of file block number           |
|                                                  | 161
|--------------------------------------------------|
|                                                  | 162 FLEND
|      Block number of end of file                 |
|                                                  | 163
|--------------------------------------------------|
|                                                  | 164 FLNUMOPENCLSREC
|  Number of open and close records (message file) |
|                                                  | 165
|--------------------------------------------------|
| Device name - 1st char. | Device name - 2nd char. |174  FLDEVNAME
|--------------------------------------------------|
| Device name - 3rd char. | Device name - 4th char. |175
|--------------------------------------------------|
| Device name - 5th char. | Device name - 6th char. |176
|--------------------------------------------------|
| Device name - 7th char. | Device name - 8th char. |177
 ---------------------------------------------------
```

Other identifiers used:

```
     FLSECURE      = FLAB(22).(15:1)#,    file secure bit
     (FLSRRELEASE)= FLAB(22).(14:1)#,    STORE/RESTORE released bit
     (FLSTORE)     = FLAB(28).(0:1)#,     file being stored
     FLRESTORE     = FLAB(28).(1:1)#,     file being restored
     (FLLOAD)      = FLAB(28).(2:1)#,     file loaded
     FLEXCL        = FLAB(28).(3:1)#,     exclusive access
     FLSR          = FLAB(28).(0:2)#,     S & R bits
     FLSRL         = FLAB(28).(0:3)#,     S, R, & L bits
     (FLSRLX)      = FLAB(28).(0:4)#,     S, R, L, & X bits
     FLSUBTYPE     = FLAB(28).(4:4)#,     device sub-type
     FLDTYPE       = FLAB(28).(8:6)#,     device type
     FLSTATUS      = FLAB(28).(14:2)#,    write/read status
     (FLLBLEOF)    = FLAB(29).(0:8)#,     no. labels written
     (FLLBL)       = FLAB(29).(8:8)#,     no. labels available
     FLSECTOFF     = FLAB(39).(0:8)#,     sector offset to data
     FLNUMEXTS     = FLAB(39).(11:5)#,    no. extents less 1
     FLLABEL       = FLABDBL(22)#,        label VTAB and sector
     FLVTAB        = FLAB(44).(0:8)#,     label VTAB index
```

Discussion:

FLACCTNAME	This is the account name of the file. It is eight bytes in length with trailing blanks added.
FLALLOCDATE	Date that the file was allocated on this system.
FLALLOCTIME	Doubleword containing the time that the file was allocated on this system.
FLBLKSIZE	This is the block size, in sectors, of the file.
FLCHECKSUM	This is the exclusive-OR checksum of the file label (excluding words 34, 42, and 43 octal) and is used for error detection. Each time the file label is read from disc the check sum is calculated and compared against the value recorded in the file label. Similarly, each time the file label is written to the disc the check sum is calculated and inserted into the file label.
FLCLID	This is the cold load number in effect the last time that the file was accessed. This should always be the current cold load number. If it is not it means that the system crashed while the file was open and that the data in the file label should be "reset" (principally the FCB vector FLFCBVECT).
FLCREATE	This is the creation date of the file. It is in the format defined by the intrinsic CALENDAR.
FLDEVNAME	This is the FOPEN device specification that was used when the file was created. This information is needed when new extents are allocated.
FLDTYPE	This is the device type number of the first extent of the file; see ACBDTYPE for a list of legal values. This value is determined by configuration.
FLEND	Number of current data blocks (that is, the end of file block number relative to the start of file).
FLEOF	This is the end-of-file pointer for the file. It is a double word integer representing the number of records in the file. It can also be viewed as the record number of the next record past EOF.
FLEXCL	This is the exclusive access flag for the file. If set it means that the file has been opened exclusively by a single accessor. If not set then the file is potentially accessible by others.
FLEXTMAP	This is the extent map of the file. The number of extents is specified by FLNUMEXTS; a OD extent descriptor indicates that the extent has not been allocated.

FLEXTSIZE	This is the extent size, in sectors, of the file. All extents in the file, except the last, have this extent size. This is a logical value, and legal values range from 1 to 65535 sectors. This limits the maximum file size to 2097120 sectors.
FLFCBVECT	If non-zero, this is the vector of the FCB for the file. If zero, the file is not being accessed.
FLFILECODE	This is the file code of the file. Known values are:

-401	IMAGE data set
-400	IMAGE root file
1024	USL file
1025	BASIC data file
1026	BASIC program file
1027	BASIC fast program file
1028	RL file
1029	Program file
1030	STAR file
1031	SL file
1040	Cross Loader ASCII file (SAVE)
1041	Cross Loader relocatable binary file
1042	Cross Loader ASCII file (DISPLAY)
1050	EDITOR KEEPQ file (non-COBOL)
1051	EDITOR KEEPQ file (COBOL)
1060	RJE punch file
1069	RSAM (Bob Strand's ISAM) file
1070	QUERY procedure file
1071	QUERY work file
1072	QUERY work file
1080	KSAM key file
1081 to 1089	Reserved for KSAM
8000 to 8099	Reserved for APL

FLFLIM	This is the end-of-space pointer for the file. It is a double integer representing the maximum number of records (fixed length record format) or blocks (undefined or variable length record format) in the file.
FLFOPTIONS	This is the FOPTIONS of the file.
FLGRPNAME	This is the group name of the file. It is eight bytes long with trailing blanks added.
FLLABEL	This is the volume table index and sector number of the file label, which is the same as the first extent descriptor.
FLLASTACC	This is the last access date of the file. It is in the format defined by the intrinsic CALENDAR.

FLLASTMOD	This is the last modification date of the file. It is in the format defined by the intrinsic CALENDAR.
FLLASTEXTSIZE	This is the size, in sectors, of the last extent in the file. If the file has one extent then this is the same as FLEXTSIZE; if the file has more than one extent then this value may be different from FLEXTSIZE. This is the size of the last physical extent for the file; it is not the size of the last allocated extent.
FLLBL	This is the number of user labels allocated for the file. Since each label is a sector long, this is also the number of sectors allocated for user labels.
FLLBLEOF	This is the end-of-data pointer for the user labels. It is analogous to FLEOF in that it represents the number of labels written.
FLLOAD	This is the LOADED flag for the file. If set it means that the file is a loaded program or SL file and cannot be modified except by a privileged accessor. This flag is set and cleared by the loader, not the file system.
FLLOCK	This identifies the word containing the lock bits, which are described separately.
FLLOCKWORD	This is the lock word of the file. It is eight bytes long with trailing blanks added. If it is all blanks then the file does not have a lockword.
FLLOCNAME	This is the local name of the file. It is eight bytes long with trailing blanks added.
FLNUMEXTS	This is the number of extents, less one, allowed for the file. It is not the number of extents allocated. Legal values range from 0 to 31, i. e., 1 to 32 extents.
FLNUMOPENCLSREC	Number of open and close records in the message file.
FLRECSIZE	This is the record size of the file in negative bytes.
FLRESTORE	This is the RESTORE flag for the file. If set it means that the file is being RESTOREd and cannot be accessed. RESTORE also sets the STORE bit for the file (FLSTORE); see FLSR for a full description of the use of these bits. This flag is set and cleared by STORE/RESTORE, not the file system.
FLSECMX	This is the security matrix of the file. The bits are organized into five groups of six bits each. (Bits 0:2 are not used.) The groups correspond to the access types: READ, APPEND, WRITE, LOCK, and EXECUTE. Within each group, each bit specifies who may have the access: ANY, ACCOUNT MGR, ACCOUNT LIB-

	RARIAN, GROUP, GROUP LIBRARIAN, CREATOR.
FLSECTOFF	This is the sector offset from the file label to the first block of the file. This is not necessarily equal to FLLBL+1 since an integral number of blocks are allocated for the file and user labels.
FLSECURE	This is the file security enforcement flag for the file. If not set then the file has been RELEASEd and the security matrix FLSECMX should be ignored. If set then secured as specified by the security matrix.
FLSR	This is the STORE and RESTORE flags for the file, which are described separately. STORE and RESTORE decode the two-bit field to indicate their operation. Legal values are: 0 - file not in use by either STORE or RESTORE 1 - illegal value 2 - file being STOREd 3 - file being RESTOREd The file system interprets the leftmost bit as indicating that the file is being accessed by either STORE or RESTORE. The rightmost bit is interpreted as indicating what access should be permitted: 0 (file being STOREd) allows read access; 1 (file being RESTOREd) allows no access. This field is set and reset by STORE/RESTORE, not the file system.
FLSRL	This is the STORE, RESTORE and LOADED flags for the file, which are described separately.
FLSRLX	This is the STORE, RESTORE, LOADED and exclusive flags for the file, which are described separately.
FLSRRELEASE	This flag is used by STORE/RESTORE. If a file is STOREd with the ";RELEASE" keyword, STORE will set this flag in the tape copy of the file label. RESTORE will allow any user to access such files, regardless of the file's normal security. If this bit is off in the tape copy of the file label, RESTORE applies normal security checks (as defined by the information in FLSECMX and FLSECURE). This bit is zero for files on disc.
FLSTART	Block number of the file's start, excluding the file label block.
FLSTATUS	This is the read/write status of the file. Legal values are: 0 - no accessors 1 - read 2 - write 3 - read/write

FLSTORE	This is the STORE/RESTORE flag for the file. If set it means that the file is being either STOREd or RESTOREd. The RESTORE bit (FLRESTORE) must be interrogated to determine which operation is taking place; see FLSR for a full description of the use of these bits. This flag is set and cleared by STORE/RESTORE, not the file system.
FLSUBTYPE	This is the device sub-type number of the first extent of the file. This value is determined by configuration.
FLUSERID	This is the creating user name of the file. It is eight bytes long with trailing blanks added.
FLUSERLBL	This field describes the user labels of the file. It consists of FLLBL and FLLBLEOF, which are described separately.
FLVTAB	This is the volume table index of the first extent of the file.

3.4 File Multi-Access Vector Table (FMAVT) DST(%54)

The FMAVT is used to locate shared PACB's for files opened multi-access. Whe an old disc file has been opened multi-access, the FMAVT is searched to determine if the file has previously been opened. The JITDST and the DADDR found i the FMAVT are compared to the JITDST of the job and the DADDR of the device o disc file being opened multi-access. If an entry exists for the file, than th PACB can be easily located for that file. If this is the first process openin the file than an entry is created and inserted into the FMAVT for the file.

Spoolfiles are opened multi-access, therefore, they will have entries in th FMAVT. $STDIN and $STDLIST also have entries in the FMAVT since they too ar opened multi-access.

Zero Entry Format

```
 -----------------------------------------------------
|                Current Table Size                   | 0 FM'CURR'SIZE
|-----------------------------------------------------|
|                  Entry Size = 4                     | 1 FM'ENTRY'SIZE
|-----------------------------------------------------|
|                Maximum Table Size                   | 2 FM'MAX'SIZE
|-----------------------------------------------------|
|                        0                            | 3
 -----------------------------------------------------
```

Descriptions:

FM'CURR'SIZE	The current size of the FMAVT in words. This value increases i increments of %200 words until FM'MAX'SIZE is reached.
FM'MAX'SIZE	The maximum allowable size in words that the FM'CURR'SIZE can get The current value of this is %4000. FM'MAX'SIZE can be change only by changing the code in Initial. The FOPEN fails when the maximum is reached.
FM'ENTRY'SIZE	Size in words of an FMAVT entry, 4 words at present.

Typical Entry Format

```
  0   1   2   3     6   7   8         12  13  14  15
 ----------------------------------------------------
| 1 | G | D |     |              JIT DST             | 0
|----------------------------------------------------|
|     Logical Device      |                          | 1 FM'DADDR
|-------------------------                           |
|                   Disk Address                     | 2
|----------------------------------------------------|
|                   PACB Vector                      | 3 FM'PACBV
 ----------------------------------------------------
```

```
FM'DEVICE    = FMAVT(0).(2:1)#,    Device bit
FM'GLOBAL    = FMAVT(0).(1:1)#,    Global multi-access bit
FM'JITDST    = FMAVT(0).(6:10)#,   JIT DST number of job opening file
FM'LDEV      = FM'DADDR(0).(0:8)#, Logical device number of file
```

Descriptions:

FM'DADDR	The disc address of the file label for disc files. For devic files, the disc address is zero.
FM'DEVICE	This bit is 1 for device files and 0 for disc files.
FM'LDEV	Locical device number of device files or the LDEV of the disc containing the file label for disc files.
FM'JITDST	The DST number of the JIT for the job that has the file open. I this field is non-zero, than only processes in the family tree o this particular job can open the file. This field is zero if th file was open global multi-access.
FM'GLOBAL	This bit is 1 if the file was opened global multiaccess, this allows multi-access to the file between jobs.
FM'PACBV	The PACB vector for this multi-access file. Used to easily fin the Physcial Access Control Block for files opened multi-access

3.5 System Global Area (SYSGLOB)

The file system uses several words in the system global area for its own use.

```
SHFCBDST     = SYSDB+%76,     shared FCB DST no.
MONITOR      = SYSDB+%77,     monitoring flag word
MAXSSECT     = SYSDB+%100,    max # spoolfile sectors
NUMSSECT     = SYSDB+%102,    current # spoolfile sectors
EXTSSECT     = SYSDB+%104,    # sectors/spoolfile extent
SPOOLINDEX   = SYSDB+%132,    class spool index
CSIOWAIT     = SYSDB+%135,    CSIOWAIT PLABEL
CCLOSEPLABL  = SYSDB+%140,    CS CCLOSE PLABEL - FPROCTERM
DSCHKPLABL   = SYSDB+%335,    DSCHECK PLABEL
DSOPENPLABL  = SYSDB+%336,    DSOPEN PLABEL
DSCLOSEPLABL = SYSDB+%337,    DSCLOSE PLABEL
SDSLDEVLABEL = SYSDB+%323,    PLABEL for SDSLDEV
MANWCPLABL   = SYSDB+%340;    MANAGEWRITECONV PLABEL
```

3.6 SIRs, Locks, and Deadlocks

The file system uses two SIRs: the File SIR, which is intended to protect file label integrity, and the FMAVT SIR, which is to guarantee the integrity of the FMAVT. Since the file system locks these resources, and also locks control blocks, deadlocks can occur if locking is done in the wrong order. Not only must the file system handle locking correctly, but the entire ensemble of the file system, its callers, and its callees must do so also. These include KSAM, which has a SIR of its own, and SYSDUMP and STORE, which lock the File SIR because they tweak bits in file labels. The presently accepted order is:

Get FMAVT SIR
Lock ACB
Get File SIR
Lock FCB

It may not be necessary to do all of these things in any particular procedure. In modifying a procedure, you should be sure that any of these locks which you change are consistent not only within your own code, but also with its callers and callees.

CHAPTER 7 PROCESS TABLES

7.1 Introduction

The operating system maintains state, control, and accounting information on each process. The data structures for this purpose are the process co trol block table (PCB; core resident, 1 entry per process) and the proces control block extension (PCBX; contained in the process' stack below DL). Process related information which must be accessible even when the proces stack is not present in main memory is maintained in the process' PCB entry. All other process related information is maintained in the proces PCBX.

A process is identified in the system by its PCB entry number, referred t as its PIN (process identification number), or by its PCBPT=(PIN)*(PCB entry size).

The structure of the PCB table, PCB entry format, PCBX structure, and PCB format are specified in this chapter.

7.2 Process Control Block Table Structure and Format

7.2.1 Fixed Cells Related to PCB

3	Absolute address of base of PCB table
4	Absolute address of current process' PCB entry
%1003	Sysbase relative address ov PCB table base
%1271	Sysbase relative address of head of dispatching queue's PCB entry
%1272	Sysbase relative address of tail of dispatching queue's PCB entry

7.2.2 PCB Entry 0 Format

0	# OF CONFIGURED ENTRIES
1	ENTRY LENGTH (%20)
2	# OF UNASSIGNED ENTRIES
3	TABLE RELATIVE INDEX TO FIRST UNASSIGNED ENTRY
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0

7.2.3 Unassigned PCB Entry Format

0	%100000
1	TABLE RELATIVE INDEX TO NEXT UNASSIGNED ENTRY
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0

```
     7.2.3 Assigned PCB Entry Format
       0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
PCB00|S |B |C |H |P |H |I |P |D |L |S |T |U |H |S |R |                           !
     |A |F |R |S |I |S |P |C |S |W |W |R |S |I |T |I |                           !
     |R |  |I |I |O |P |E |  |O |  |  |W |E |P |O |T | RESABORTINFO              !
     |  |  |T |R |V |R |X |  |F |  |  |  |D |R |V |B |                           !
     |  |  |  |  |R |I |P |  |T |  |  |  |Q |I |A |K |                           !
     |-----------------------------------------------|
PCB01| SYSBASE RELATIVE ADDRESS OF PROCESS' SEGMENT  | SLLPTR
     |               LOCALITY LIST                   |
     |-----------------------------------------------|
     |A |                             |              |
     |D |     XDS       DST#          | RESERVED     |
PCB02|B |                             |              | DBXDSINFO
     |-----------------------------------------------|
     | A|                             | S|           |
PCB03| O|     STK     DST#            | C|RESERVED   | STKINFO
     |-----------------------------------------------|
     |  |  |  |  | B|  | U| J| T| M| S|  | I| S| T| M|
PCB04| M| R| R| M| I| I| C| N| I| S| O|FA| M| I| I| E| WAKEMASK
     |  | G| L| A| O| O| P| K| M| G| N|  | P| R| M| M|
     |-----------------------------------------------|
PCB05|     FATHER'S PIN      |     SON'S PIN         | FATHERSONINFO
     |-----------------------------------------------|
PCB06|   NEXT BROTHER'S PIN  |       BLKIDX          | BROTHERINFO
     |-----------------------------------------------|
PCB07|      PIMP PIN         |       BPTLINK         | PIMPINBREAKLINK
     |-----------------------------------------------|
     |        |W |     | D|  |                       |                           !
PCB08|        |S |     | E|F |                       | PIINFONIMPPIN             !
     |   PSIM |O | OA  | A|A |       NIMPPIN         |                           !
     |        |F |     | D|C |                       |                           !
     |        |T |     |  |  |                       |                           !
     |-----------------------------------------------|                           !
PCB09|L | BMS | PPC |S |  PTYPE |S |HK|SK|ST|HB|CY|BK| PROCSTATE                 !
     |I |     |     |O |        |I |  |  |  |  |  |  |                           !
     |V |     |     |V |        |  |  |  |  |  |  |  |                           !
     |-----------------------------------------------|
PCB10|                 EVENT FLAGS                |WS| EVENTFLAGS                !
     |-----------------------------------------------|
PCB11| SEGIDENTIFIER OF LAST REF. SWAPPABLE SEGMENT  | LASTREFSWAPSEG
     |-----------------------------------------------|
PCB12|            CSTX BLOCK MAP INDEX               | PBX
     |-----------------------------------------------|
PCB13|D |L |C |D |E |I |C |A |                       | QUEUEINGINFO              !
     |I |Q |  |  |  |N |O |S |                       |                           !
     |S |  |  |  |  |T |R |O |      PRIORITY         |                           !
     |P |  |  |  |  |E |E |F |                       |                           !
     |Q |  |  |  |  |R |R |T |                       |                           !
     |-----------------------------------------------|
PCB14|   SYSBASE INDEX OF NEXT PCB ENTRY IN QUEUE    | NQPTR
     |-----------------------------------------------|
PCB15| SYSBASE INDEX OF PREVIOUS PCB ENTRY IN QUEUE  | PQPTR
     |-----------------------------------------------|
```

7.2.4 PCB Assigned Entry Field Descriptions

PCB00	.(0:1)	SAR ==> scheduling attention required
	.(1:1)	Bounds Flag -- Priv mode bounds check
	.(2:1)	CRIT ==> process is critical
	.(3:1)	HSIR ==> process has a sir
	.(4:1)	PIOVR ==> pending PI, process critical
	.(5:1)	HSPRI ==> hold sir priority
	.(6:1)	IPEXP ==> incore protect expired
	.(7:1)	PC ==> prempt capability
	.(8:1)	DSOFT ==> Delayed soft int processing. A pending soft int cannot be processed because of sir or critical state. PSEUDOINT will be invoked when these condition(s) go away.
	.(9:1)	LW ==> long wait
	.(10:1)	SW ==> short wait
	.(11:1)	TRW ==> terminal read wait
	.(12:1)	USEDQ ==> used a quantum since transaction began
	.(13:1)	HIPRI ==> hold impeded priority
	.(14:1)	STOVA ==> processing abort due to stack overflow.
	.(15:1)	RITBK
PCB01	.(0:16)	SLLPTR, SYSBASE relative index to process' segment locality list
PCB02	.(0:1)	ADB, set if db pointing to an absolute address
	.(1:10)	XDS, DST entry number of extra data seg. to which DB is set; zero if none.
	.(11:4)	Reserved for expansion of DST entry number field
PCB03	.(0:1)	STOVRALL FLAG ==> stack overflow is already allocated
	.(1:10)	DST entry number of process' stack
	.(11:1)	SC, set if executing system code
	.(12:3)	Reserved
PCB04	.(0:1)	M, mourning wait.
	.(1:1)	RG, global RIN wait.
	.(2:1)	RL, local RIN wait.
	.(3:1)	MA, mail wait.
	.(4:1)	BIO, blocked I/O wait.
	.(5:1)	IO, I/O wait.
	.(6:1)	UCP, UCOP wait and RIT wait.
	.(7:1)	JNK, junk wait.
	.(8:1)	TIM, timer wait.
	.(9:1)	MSG, file system basic ipc message wait.
	.(10:1)	SON, son wait.
	.(11:1)	FA, father wait.
	.(12:1)	IMP, process waiting to be unimpeded.
	.(13:1)	SIR, process waiting for a sir.
	.(14:1)	TIM, process waiting for a time out.
	.(15:1)	MEM, process waiting for memory.

PCB05	.(0:8)	FPIN, father's PCB entry number	
	.(8:8)	SPIN, son's PCB entry number	
PCB06	.(0:8)	BPIN, brother's PCB entry number	
	.(8:8)	BLKIDX (reserved)	
PCB07	.(0:8)	PIMPPIN, previous impeded pin.	
	.(8:8)	BPTLINK, breakpoint link for process.	
PCB08	.(0:3)	PSIM, pseudo - interrupt mode	
		1: hard kill	
		2: soft kill	
		3: stop	
		4: hibernate	
		5: escape	
		6: break	
		7: normal	
	.(3:1)	ASOFT, OK for soft int to wake process	!
		even though it is waiting on another event.	!
	.(4:2)	OA	
		0: other source	
		1: father	
		2: son	
		3: reply done on RIT wait	
	.(6:1)	DEAD, set during expiration.	
	.(7:1)	FAC, if set, the father is to be activated on process termination.	
	.(8:8)	NIMPPIN, next impeded process' pin	
PCB09	.(0:1)	LIVE, set if process is alive.	
	.(1:2)	BMS, block mail, valid if MA set	
		0: sent to father	
		1: rec from father	
		2: send to son	
		3: rec from son	
	.(3:2)	PPC, process to process communication, set with respect to son.	
		0: null	
		1: son to father	
		2: father to son	
		3: blocked	
	.(5:1)	STOV, stack overflow bit	
	.(6:3)	PTYPE, process type	
		0: user	
		1: user, son of main	
		2: user, main	
		3: user, main, task	
		4: system	
		5:	
		6: system, UCOP	
		7:	
	.(9:1)	SI, set when the Dispatcher (and PSEUDOINT)	!
		should be aware of a pending soft interrupt.	!
	.(10:1)	HK, hard kill pseudo interrupt	
	.(11:1)	SK, soft kill pseudo interrupt	

	.(12:1)	ST, stop pseudo interrupt
	.(13:1)	HB, hibernate pseudo interrupt
	.(14:1)	CY, control-y pseudo interrupt
	.(15:1)	BK, break pseudo interrupt
PCB10	.(0:15)	EVENTFLAGS, one for each wait class in PCB04
	.(15:1)	WS, wake up waiting switch set if an awake is missing.
PCB11	.(0:16)	LASTREFSWAPSEG, segment identifier of last referenced swappable code segment.
PCB12	.(0:16)	PBX, CSTX block map index of process' program.
PCB13		(QUEUEING INFO)
	.(0:1)	DISPQ ==> on dispatching queue
	.(1:1)	L scheduling class
	.(2:1)	C scheduling class
	.(3:1)	D scheduling class
	.(4:1)	E scheduling class
	.(5:1)	INTER ==> process is interactive
	.(6:1)	CORER ==> process is core resident
	.(7:1)	ASOFT, Allow soft interrupt. A value of 1 implies that user soft interrupts will be processed. A zero value inhibits user soft ints (they are queued). This bit is managed by FINTSTATE and FINTEXIT intrinsics.
	.(8:8)	Process' scheduling priority
PCB14	.(0:16)	NQPTR, sysbase index of PCB entry of next process in scheduling queue
PCB15	.(0:16)	PQPTR, sysbase index of PCB entry of previos process in scheduling queue

7.3 PCBX Structure and Format

7.3.1 PCBX General Structure

```
             ------------------------------  ---------------------
     a--->| DL-a = SEG. REL DL VALUE    |                        ^
          |-----------------------------|         |              |
          | DB-a = SEG> REL DB VALUE    |         |              |
          |-----------------------------|         |              |
          |                             |      PXGLOB            |
          |                             |         |              |
          |                             |         |              |
          |-----------------------------|  -------------         |
     b--->|    c-b = PXFIXED LENGTH     |                        |
          |-----------------------------|         |              |
          |                             |         |              |
          |                             |         |              |
          |-----------------------------|         |              |
          | PXFIXED EXPANSION AREA      |      PXFIXED           |
          |        BIT MAP              |         |              |
          |       (4 words)             |         |              |
          |-----------------------------|         |              |
          |                             |         |              |
          ~                             ~         |              |
          ~      PXFIXED EXPANSION      ~         |              |
          |                             |                        |
          |                             |                      PCBX
          |-----------------------------|  --------------      SIZE
     c--->|     d-c = PXFILE LENGTH     |                        |
          |-----------------------------|         |              |
          |                             |         |              |
          |                             |         |              |
          |-----------------------------|         |              |
          |                             |      PXFILE            |
          ~               |             ~         |              |
          ~PXFILE EXPANSION/CONTRACTION~          |              |
|         |               |             |         |              |
          |               |             |         |              |
          |-----------------------------|  -------------         |
     d--->| COUNT OF SECTORS ALLOCATED  |                        |
          |  FOR PXFIXED EXPANSION      |                        |
          |-----------------------------|                        |
          |           DL-c              |                        |
          |-----------------------------|                        |
          |           DL-b              |                        |
          |-----------------------------|                        |
          |           DL-a              |                       \ /
          |-----------------------------|  ---------------------.---
    DL--->|                             |
          ~                             ~
          ~                             ~
          |                             |
          |-----------------------------|
    DB--->|                             |
          ~                             ~
```

7.3.2 PXGLOB FORMAT

The PXGLOB portion of the pcbx is for job information, and contains the same job related information for all processes belonging to the same jo

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|            DL-a=SEG. REL DL VALUE             |0
|-----------------------------------------------|
|            DB-a=SEG. REL DB VALUE             |1
|-----------------------------------------------|
|                USER ATTRIBUTES                |2
|-----------------------------------------------|
|      JMAT INDEX       | ACTUAL JOB INPUT LDN  |3
|-----------------------------------------------|
|JPCNTINDEX(RelByteAddr)| ACTUAL JOB OUTPUT LDN |4
|-----------------------------------------------|
| STACK DUMP FLAGS|        JDT DST INDEX        |5
|-----------------------------------------------|
|//| R|  TY | D| I|        JIT DST INDEX        |6
|-----------------------------------------------|
|      JCUT INDEX       |**|//|*****|///////////|7
|-----------------------------------------------|
```

R = restart bit
I = job in/list interactive
D = job in/list duplicative
TY = job type
0 = undefined
1 = session
2 = job
3 = task
* = reserved:

Stack Dump Flags
Bit 0 = Armed
Bit 1 = Suppress traceback
Bit 2 = Suppress ASCII
Bit 3 = Q-63 to S
Bit 4 = QINIT to S
Bit 5 = DL to QINIT

7.3.3 PXFIXED ASSIGNMENTS

The PXFIXED portion of the pcbx contains specific information and control information.

```
  |----------------------------------------------|
 0|                c-b PXFIXED SIZE              |0
  |----------------------------------------------|
 1|                RELATIVE S(S-DB)              |1
  |----------------------------------------------|
 2|                RELATIVE Z(Z-DB)              |2
  |----------------------------------------------|
 3|                 INITIAL Q(Q-DB)              |3
  |----------------------------------------------|
 4|         INITIAL RELATIVE DL (DB-DL)          |4
  |----------------------------------------------| Trap Modes
 5|   GENERAL RESOURCE CAPABILITY(FROM PROG-FILE) |5 .MAT(12:1)-Arith.
  |----------------------------------------------|  .MLT(13:1)-Library
 6| RESERVED                    |MAT|MLT|MST|MCY|6 .MST(14:1)-System
  |----------------------------------------------|  .MCY(15:1)-Ctl-Y
 7|LINK TO XDS ENTRIES IN EXPANSION AREA| XDS CNT |7 (XDS CNT- 12:4)
  |----------------------------------------------|
10| P| S|   EXTRA DATA SEGMENT DST INDEX         |8
  |----------------------------------------------|
11| P| S|   EXTRA DATA SEGMENT DST INDEX         |9
  |----------------------------------------------|
12| P| S|   EXTRA DATA SEGMENT DST INDEX         |10  / 0:1 RESERVED FOR
  |----------------------------------------------|    |     CST EXPANSION
13| P| S|   EXTRA DATA SEGMENT DST INDEX         |11  | 1:1 = 1 IF ABORT
  |----------------------------------------------|    |       IN PROGRESS
14| X| A|   ABORT Y      |RW|  INITIAL CST INDEX |12  < 7:1 = 0 IF HAVE R/W
  |----------------------------------------------|    |       ACCESS TO
15|       MAXIMUM STACK SIZE(MAXDATA LIMIT)      |13  |       PROG FILE
  |----------------------------------------------|    |     = 1 OTHERWISE
16|             ARITHMETIC TRAP MASK             |14  | 8:8 = CST # OF SEG
  |----------------------------------------------|    |       INITIALLYEXECUTED
17|            ARITHMETIC TRAP PLABEL            |15  \       AT PROC CREATION
  |----------------------------------------------|
20|             LIBRARY TRAP PLABEL              |16
  |----------------------------------------------|
21|             SYSTEM TRAP PLABEL               |17
  |----------------------------------------------|
22|              CONTROL Y PLABEL                |18
  |----------------------------------------------|  JOB TYPE:
  | JOB |                                        |    1=SESSION
23|TYPE |               JOB#                     |19  2=JOB
  |----------------------------------------------|
24|ACTUAL SIZE OF VIRTUAL SPACE ALLOCATED TO STACK|20
  |----------------------------------------------|
25|              USER ABORT PLABEL               |21
  |----------------------------------------------|  U user udcs exist
26|U |L | C|///////////|A |  LOAD PROCEDURE I.D. |22 L logging
  |----------------------------------------------|  A acct udcs exist
27|CUR.MAX STACK SIZE(largest value ever for Z-DL)|23 C process shares clock
  |----------------------------------------------|    1 => clock shared
```

(reserved)

PXFIXED (CONT.)

```
  |-----------------------------------------------|
30|               PROCESS CPU TIME                |24
  |                                               |
31|                    (MSEC)                     |25
  |-----------------------------------------------|
32|    MAXIMUM DATA SEG SIZE USED(IN SECTORS)     |26
  |-----------------------------------------------|
33|   TOTAL VIRTUAL STORAGE USED(IN SECTORS)      |27
  |-----------------------------------------------|
34|       CURRENT EXTRA DATA SEGMENT SPACE        |28
  |-----------------------------------------------|
35|       MAXIMUM EXTRA DATA SEGMENT SPACE        | 29
  |-----------------------------------------------|
36| PRIV MODE BOUNDS FLAGS|       STOV COUNT      | 30
  |-----------------------------------------------|
37|  PROCESS EXECUTION TIME REMAINDER (IN MSEC)   | 31
  |-----------------------------------------------|
40|         SET TO-1 WHEN IN BREAK MODE*          | 32
  |-----------------------------------------------|
41|      CONTINUE FLAG (:CONTINUE COMMAND)**      | 33
  |-----------------------------------------------|
42|               IMAGE PLABL                     | 34
  |-----------------------------------------------|
43|                  ERROR LEVEL                  | 35
  |-----------------------------------------------|
44|               INTRINSIC ERRORS                | 36
  |-----------------------------------------------|
45|               INTRINSIC ERRORS                | 37
  |-----------------------------------------------|
46|               INTRINSIC ERRORS                | 38
  |-----------------------------------------------|
47|               INTRINSIC ERRORS                | 39
  |-----------------------------------------------|
50|               INTRINSIC ERRORS                | 40
  |-----------------------------------------------|
51|               INTRINSIC ERRORS                | 41
  |-----------------------------------------------|
52|TSLR, virtual time since last rescheduled      |42
  |-----------------------------------------------|
53|TSTB, virtual time since transaction began     |43
  |-----------------------------------------------|
54|TSSWAPIN, virtual time since swapin            |44
  |-----------------------------------------------|
55|TSLA, virtual time since last absence          |45
  |-----------------------------------------------|
56|TSLD, virtual time since last deallocation     |46
  |-----------------------------------------------|
57|QCNT, quantums used since transaction began    |47
  |-----------------------------------------------|
60|/|D|/|O| RESERVED FOR FUTURE SOFT INT USE      |48
  |/|C|/|S|                                       |
```

```
    |/|Y|/|I|                                         |                                         !
    |-------------------------------------------------|                                         !
 61| TRLX INDEX FOR KERNEL TIMEOUT PROCEDURE          | 49                                      !
    |-------------------------------------------------|                                         !
 62|       DATACOMM TERMINATION TRAP PLABEL           | 50                                      !
    |-------------------------------------------------|                                         !
 63|                 # SL FAULTS                      | 51                                      !
    |-------------------------------------------------|                                         !
 64|                 # PCB FAULTS                     | 52                                      !
    |-------------------------------------------------|                                         !
 65|                # DATA SEG FAULTS                 | 53                                      !
    |-------------------------------------------------|                                         !
 66|           # BLOCKED DISC I/O's ISSUED            | 54                                      !
    |-------------------------------------------------|                                         !
 67|       # UNBLOCKED DISC I/O's REQUESTED           | 55                                      !
    |-------------------------------------------------|                                         !
 70|       # UNBLOCKED DISC I/O's WAITED ON           | 56                                      !
    |-------------------------------------------------|                                         !
 71|             # IMPEDES (SUBSYSTEM)                | 57                                      !
    |-------------------------------------------------|                                         !
 72|                # IMPEDES (SYSTEM)                | 58                                      !
    |-------------------------------------------------|                                         !
 73|                   # SIR BLOCKS                   | 58                                      !
    |-------------------------------------------------|                                         !
 74|  |CY|  |SI|                                      | 60                                      !
    |-------------------------------------------------|                                         !
 75|             TIMEOUT TRLX                         | 61                                      !
    |-------------------------------------------------|                                         !
 76|                   RESERVED                       | 62                                      !
    |-------------------------------------------------|                                         !
 77|       RESERVED FOR DEBUG                         | 63                                      !
    |-------------------------------------------------|                                         !
100|      PCLASSMASK                                  | 64                                      !
    |-------------------------------------------------|                                         !
101|      PROCQUESTOPWORD                             | 65                                      !
    |-------------------------------------------------|                                         !
102|                                                  | 66                                      !
    |      PROCSTOPTIME                                |                                         !
103|                                                  | 67                                      !
    |-------------------------------------------------|                                         !
NOTES:   P = 1 if opened by priv user                                                           !
         S = 1 if data seg is sharable                                                          !
                                                                                                !
       PCLASSMASK    = BIT MASK OF CLASSES THIS PROCESS HAS ENABLED                             !
       PROCQUESTOPWORD.(0:4) = PROCESS PRIORITY: 7 => L QUEUE                                   !
                                                 6 => C QUEUE                                   !
                                                 2 => D QUEUE                                   !
                                                 1 => E QUEUE                                   !
                      .(4:12)= REASON STOPPED: 1 => STOP SEG FAULT                              !
                                               2 => STOP DISC WAIT                              !
                                               3 => BLOCKED I/O, NON TERMINAL                   !
                                               4 => TERMINAL READ                               !
                                               5 => STOP IMPEDE                                 !
                                               6 => STOP ACTIVE                                 !
       PROCSTOPTIME = DBL WORD TIMESTAMP OF WHEN PROCESS STOPPED FOR                            !
```

	REASON GIVEN IN PROCQUESTOPWORD
DCY	A DELAYED CONTROL Y IS PENDING (THIS BIT IS CHECKED BY ININ ON BOUNDS VIOLATION TO DETERMINE IF GOT: 1) TRUE BOUNDS VIOLATION OR 2) AN INDUCED BOUNDS VIO THAT INDICATES THAT THE CONTROL Y TRAP PROCEDURE MAY NOW BE ENTERED).
OSI	STATE OF THE "ASOFT" PCB BIT WHEN CONTROL Y TRAP WAS ENTERED. ASOFT = 1 ALLOWS USER SOFT INTERRUPTS AGAINST THE PROCESS. IT IS SET TO ZERO WHEN THE CONTROL Y HANDLER IS ENTERED. IT IS SET TO ITS PRIOR STATE WHEN THE USER CALLS RESETCONTROL.

* SET TO COMMAND RECORD LENGTH WHEN COMMAND PENDING (I.E. COMMAND ENTERED DURING BREAK OR ENCOUNTERED DURING FLUSHING).

** CONTINUE FLAG VALUES

0 = NO CONTINUE IN EFFECT
1 = CONTINUE JUST ENCOUNTERED
2 = CONTINUE IN EFFECT FOR THIS COMMAND

CY FLAG

PCBXFIXED(61).(1:1) = SET BY PSEUDOINT WHEN THERE IS A PENDING CONTROL Y WHICH CANNOT BE PROCESSED BECAUSE OF SYSTEM CODE OR PRIVILEGED CODE. ININ CHECKS THIS BIT ON BOUNDS VIOLATION OR TRACE TRAP.

SI FLAG

PCBXFIXED(61).(3:1) = SPECIFIES THE STATE OF THE USER INTERRUPT FLAG WHEN THE CURRENT CONTROL Y WAS PROCESSEI

7.3.4 PXFIXED EXPANSION BITMAP

The PXFIXED bitmap and expansion area is for use in accounting for extra data segments acquired by the process.

The names of extra data segments allocated by and belonging to a process are kept in the PXFIXED part of the PCBX. Up to four such names (DST numbers) can be kept in cells that are permanently allocated for this purpose at PXFIXED locations 8 through 11. If more than four extra data segments are allocated, an expansion of PXFIXED occurs in which it is enlarged by one sector (128 words). Up to three such sectors can be allocated.

The expansion area is managed by a cumbersome scheme in which each sector is divided up into "frames" of eight words. The first word of each frame contains the frame size in the low 4 bits, and a pointer to the next frame (or zero, if none) in the upper 12 bits. The frames are allocated by a bitmap; one bitmap word is needed for each expansion sector, and the words are stored at locations 76,77, and 78 in PXFIXED. Although a procedure exists to de-allocate a frame, it is never called. The original intent presumably was to permit use of frame space by activities other than DST management, but nothing of this sort has been done.

In order to permit the four PXFIXED words to be managed as a frame, they are preceded by a word at PXFIXED(7) which is in the frame header format described above; initially, the frame size field is 4 and the pointer is 0.

Pictorially, a frame looks like this:

```
  0   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15
----------------------------------------------------------------
|           LINK TO NEXT FRAME                    | AVBL WORDS   |
----------------------------------------------------------------
| P | S |                 DST NUMBER OF XDS                      |
----------------------------------------------------------------
| P | S |                 DST NUMBER OF XDS                      |
----------------------------------------------------------------
          . . . (total 4 [first frame] or 7 [all add'l] DST words)
```

P=0 if DST is privileged; i.e., creator was in privileged mode. Non-privileged DSTs are subject to a SYSGLOB limit on the number of such DSTs per process. Also, non-privileged users of the extra data segment intrinsics see only a "logical" index which is basically the negative ordinal position of the PXFIXED slot containing the DST number, but with the sign bit cleared. Privileged callers get the actual DST number to use, so they can do privileged instructions such as MFDS.

S=0 if DST is specified as sharable between processes within the job. There is a list of shared DSTs in the JDT.

File System Section of PCBX (PXFILE)

The PXFILE area is a sub-section of the PCBX. It is a contiguous, expandable and contractable block of storage that is managed by the file system primarily for its own use. Other sybsystems, namely CS and DS, also make use of the PXFILE section. In doing so they must conform to the conventions of the file system.

The overall structure of the PXFILE area is:

```
------------------------
|                      |
|      OVERHEAD        |     (fixed)
|                      |
|----------------------|
|                      |
|    CONTROL BLOCK     |     (variable)
|        TABLE         |
|                      |
|----------------------|
|                      |
|      AVAILABLE       |     (variable)
|                      |
|----------------------|
|                      |
|   AVAILABLE FILE     |     (variable)
|       TABLE          |
|                      |
------------------------
```

VECTOR FORMAT

```
 0            5 6                          15
|----------------------------------------------|
|   ENTRY      |        DST NUMBER             |
|----------------------------------------------|
```

Overhead (PXFILE)

The part labeled OVERHEAD contains information that is pertinent to the entire table.

```
 0  1                    7   8                          15
----------------------------------------------------------------
|                  PXFILE SIZE IN WORDS                        |  0
|--------------------------------------------------------------|
|    LAST DOPEN ERROR NUMBER    |   LAST COPEN ERROR NUMBER    |  1
|--------------------------------------------------------------|
| N |                                                          |  2
|--------------------------------------------------------------|
|     LAST DF AFT               |      SLAVE AFT NUMBER        |  3
|--------------------------------------------------------------|
|    LAST KOPEN ERROR NUMBER    |   LAST FOPEN ERROR NUMBER    |  4
|--------------------------------------------------------------|
|                   AFT SIZE IN WORDS                          |  5
|--------------------------------------------------------------|
|                  CS TRACE FILE INFO                          |  6
|--------------------------------------------------------------|
|         LAST RESPONDING NO-WAIT I/O AFT ENTRY NUMBER         |  7
|--------------------------------------------------------------|
|        1st USER (NOBUF) CONTROL BLOCK TABLE DST NUMBER       |  8
|--------------------------------------------------------------|
|        2nd USER (NOBUF) CONTROL BLOCK TABLE DST NUMBER       |  9
|--------------------------------------------------------------|
|        3rd USER (NOBUF) CONTROL BLOCK TABLE DST NUMBER       | 10
|--------------------------------------------------------------|
|        4th USER (NOBUF) CONTROL BLOCK TABLE DST NUMBER       | 11
|--------------------------------------------------------------|
|        5th USER (NOBUF) CONTROL BLOCK TABLE DST NUMBER       | 12
|--------------------------------------------------------------|
|        6th USER (NOBUF) CONTROL BLOCK TABLE DST NUMBER       | 13
|--------------------------------------------------------------|
|        7th USER (NOBUF) CONTROL BLOCK TABLE DST NUMBER       | 14
|--------------------------------------------------------------|
|        8th USER (NOBUF) CONTROL BLOCK TABLE DST NUMBER       | 15
----------------------------------------------------------------
```

In general the following identifiers are used when referring to this part of the PXFILE area:

```
DEFINE
PXFSIZE         = PXFILE#,        <<PXFILE SIZE>>
PXDSOPENERR     = PXFILE(1).(0:8)#,<<LAST DOPEN ERROR CODE>>
PXCOPENER       = PXFILE(1).(8:8)#,<<LAST COPEN ERROR CODE>>
PXFNOCB         = PXFILE(2).(0:1)#,<<NO CB'S IN PXFILE CBT?>>
PXLASTDSAFT     = PXFILE(3).(0:8)#,<<DSNUM OF LAST DS OPEN>>
PXSLAVEAFT      = PXFILE(3).(8:8)#,<<DSNUM OF SLAVE PTOP DSOPEN>>
PXFKOPEN        = PXFILE(4).(0:8)#,<<LAST KOPEN ERROR CODE>>
PXFFOPEN        = PXFILE(4).(8:8)#,<<LAST FOPEN ERROR CODE>>
PXFAFTSIZE      = PXFILE(5)#,     <<AFT SIZE IN WORDS>>
PXFCTRINFO      = PXFILE(6)#,     <<CS TRACE FILE INFO>>
```

OVERHEAD (CONT.)

```
PXFLEFTOFF       = PXFILE(7)#,      <<LAST RESPONDING AFT NR.>>
PXFCBT1          = PXFILE(8)#,      <<1ST USER CBT DST NR.>>
PXFCBT2          = PXFILE(9)#,      <<2ND USER CBT DST NR.>>
PXFCBT3          = PXFILE(10)#,     <<3RD USER CBT DST NR.>>
PXFCBT4          = PXFILE(11)#,     <<4TH USER CBT DST NR.>>
PXFCBT5          = PXFILE(12)#,     <<5TH USER CBT DST NR.>>
PXFCBT6          = PXFILE(13)#,     <<6TH USER CBT DST NR.>>
PXFCBT7          = PXFILE(14)#,     <<7TH USER CBT DST NR.>>
PXFCBT8          = PXFILE(15)#;     <<8TH USER CBT DST NR.>>
```

The following is an alphabetized list of the above identifiers along with a discussion of their meaning.

PXFAFTSIZE
This is the size (in words) of the Available File Table. Note that the size is in words and not in terms of number of entries. The reason for this is that it simplifies the calculation for the size of the available block.

PXFCBT1-8
These are the DST numbers of the user (NOBUF) control block tables. A DST number of 0 indicates that no data segment is allocated. Note that a DST number is representable with ten bits; a full word is used to simplify the code.

PXFCOPEN
This contains the last COPEN error number. It is not used by the file system; it is included here for completeness only.

PXFCTRINFO
This contains information pertinent to the CS trace file. It is not used by the file system; it is included here for completeness only.

PXFDOPEN
This contains the last DOPEN error number. It is not used by the file system; it is included here for completeness only.

PXFDSINFO
This cell is reserved for DS. It is not used by the file system; it is included here for completeness only.

PXFFOPEN
This contains the last FOPEN error number. If it is zero then the last FOPEN completed successfully; if it is non-zero then the last FOPEN completed unsuccessfully and the number represents the file system error number. Note that only eight bits are needed to hold the error number; a full word is used to simplify the code.

PXFKOPEN
This contains the last "KOPEN" error number. Since KSAM is imbedded in the file system, an FOPEN failure on a KSAM file can be caused by a failure to open either the key file or the data file. This error number is used in conjunction with PXFFOPEN to determine which file caused the KSAM open failure. Note that this error number is not used by the file system; it is included here for completeness only.

PXFLEFTOFF
This is the AFT entry number of the last file/line that completed a no-wait I/O; if zero then no no-wait I/O has been completed. This cell is maintained solely by and for the IOWAIT intrinsic.

PXFNOCB
This bit is used to signify that no control blocks are to be created in the PXFILE control block table. This bit is set by the NOCB parameter to the CREATE intrinsic or the :RUN command. The reason for this feature is to permit the 3000/20 user to have as much stack space as possible; otherwise the MPE/30 file system will take away several hundred words of stack for the PXFILE control block table.

PXFSIZE
This is the size (in words) of the complete PXFILE area. It is the sum of the overhead block, the control block table, the available file table and the available block.

Control Block Table (PXFILE)

The part labeled CONTROL BLOCK TABLE contains a file control block table. This is a new feature with MPE/30; it is not present under MPE/20.

The format of the control block table is the same as any other file control block table. The only difference is that addressing is slightly more complicated since the table does not begin at DB+0. As a result all pointers within the table are table relative; the starting address of the table must be added to a pointer to generate a final DB-relative address. This addressing convention is consistently applied to all file control block tables. When the control block table is expanded, space is taken from

the AVAILABLE area. If no space is available then the PXFILE area is expanded and the acquired space is added to the AVAILABLE area.

The interested reader is referred to section 3.2 for a more detailed description of file control block tables.

```
  0                                                          15
 ----------------------------------------------------------------
 |                    TABLE SIZE IN WORDS                       | 16
 |--------------------------------------------------------------|
 |                DST NUMBER CONTAINING TABLE                   | 17
 |--------------------------------------------------------------|
 |                VECTOR TABLE SIZE IN WORDS                    | 18
 |--------------------------------------------------------------|
 |                         LOCK WORD                            | 19
 |--------------------------------------------------------------|
 |                       IMPEDED QUEUE                          | 20
 |--------------------------------------------------------------|
 |                                                              | 21
 |                                                              |
 |                        VECTOR TABLE                          |
 |                                                              |
 |                                                              |
 |--------------------------------------------------------------|
 |                                                              |
 |                                                              |
 |                                                              |
 |                     CONTROL BLOCK AREA                       |
 |                                                              |
 |                                                              |
 |                                                              |
 ----------------------------------------------------------------
```

In general the following identifiers are used when referring to this part of the PXFILE area:

```
DEFINE
PXFCBTAB        = PXFILE(16)#,    <<CONTROL BLOCK TABLE>>
PXFCBTSIZE      = PXFILE(16)#,    <<TABLE SIZE IN WORDS>>
```

CONTROL BLOCK TABLE (CONT.)

```
-------------------

PXFDSTX          = PXFILE(17)#,    <<TABLE DST NUMBER>>
PXFVTSIZE        = PXFILE(18)#,    <<VECTOR TABLE SIZE IN WORDS>>
PXFLOCK          = PXFILE(19)#,    <<TABLE LOCK WORD>>
PXFQUEUE         = PXFILE(20)#,    <<TABLE IMPEDED QUEUE>>
PXFVT            = PXFILE(21)#;    <<VECTOR TABLE>>
```

The following is an alphabetized list of the above identifiers along with a discussion of their meaning.

PXFCBTAB
This is the first word of the control block table. In general this is used only when referring to the entire control block table.

PXFCBTSIZE
This is the size in words of the control block table. In general this is used only when calculating the size of the available block.

PXFDSTX
This is the DST number of the data segment that contains the control block table. This is the same as the DST number of the stack. Note that the convention of referring to the DST number of the stack as zero is not used. The reason for this is that the file system may refer to a PXFILE control block table in another stack. This would result in an ambiguity since that PXFILE control block table would also have a DST number of zero.

PXFLOCK
This is the lock word for the table and has the same format as the lock word for a control block in the table.

PXFQUEUE
This is the impeded queue for the table and has the same format as the impeded queue for a control block in the table.

PXFVT
This is the first word of the vector table. It is used when referring to the vector table in general.

PXFVTSIZE
This is the size, in words, of the vector table. Note that this is the length of the table and does not reflect the number of entries used or unused.

Available Block Area (PXFILE)

The part labeled AVAILABLE BLOCK is used to provide space when the Control Block Table or the Available File Table is expanded. These two tables grow towards each other, and when more space is needed it is simply taken from the Available Block.

When the Available Block is exhausted, the PXFILE area is expanded, the AFT is relocated and the new space is added to the Available Block.

Note that currently the PXFILE area is only expanded; it is never contracted.

Available File Table, AFT (PXFILE)

The part labeled AVAILABLE FILE TABLE contains information used by the file system (or CS, DS, etc.) to grossly characterize the file access and, most importantly, to give the location of the control blocks.

The overall structure of the AFT is:

```
-------------------------
|                       |
|        ENTRY N        |     (fixed)
|                       |
|-----------------------|
|                       |
|           .           |
|           .           |
|           .           |
|                       |
|-----------------------|
|                       |
|        ENTRY 1        |     (fixed)
|                       |
-------------------------
```

where N = PXFAFTSIZE/4.

The AFT is as long as specified by PXFAFTSIZE. Unused entries are all zero's. When the table is full it is expanded by taking space from the AVAILABLE block.

The AFT is negatively indexed by file number: the entry at DL-8 corresponds to file number 1, the entry at DL-12 corresponds to file number 2, etc.

AFT (CONT.)

The structure of an AFT entry is:

```
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
-----------------------------------------------------------------
| ENTRY TYPE    | N |                                           |  0
|               |   |                                           |
|---------------------------------------------------------------|
|                    PHYSICAL ACB VECTOR                        |  1
|---------------------------------------------------------------|
|                    LOGICAL ACB VECTOR                         |  2
|---------------------------------------------------------------|
|                    NO-WAIT I/O IOQX                           |  3
-----------------------------------------------------------------
```

Note that the entry format is dependent on the entry type. The one shown above is the one used by the file system.

In general the following identifiers are used when referring to an AFT entry:

```
DEFINE
AFTTYPE         = AFT.(0:4)#,      <<ENTRY TYPE>>
AFTNULL         = AFT.(4:1)#,      <<$NULL FILE>>
AFTPACBV        = AFT(1)#,         <<PACB VECTOR>>
AFTLACBV        = AFT(2)#,         <<LACB VECTOR>>
AFTIOQX         = AFT(3)#;         <<NO-WAIT I/O IOQX>>
```

The following is an alphabetized list of the above identifiers along with a discussion of their meaning.

AFTIOQX
This is the IOQ index of the pending no-wait I/O (if any). Note that this is applicable iff the file was opened with the NOWAIT option specified. Also, CS and DS have the same capability and use this cell in a consistent manner. The reason for this is that the IOWAIT intrinsic services the file system as well as CS and DS, and is the principal user of this cell. If the cell is zero then there is no I/O pending; otherwise the cell contains the IOQ index corresponding to the pending I/O.

Exception: a nonzero value for message files specifies the accesors reply port (instead of an IOQ entry).

AFTLACBV
This is the vector of the Logical ACB (LACB) (if any). Note that this is applicable iff the file was opened with the multi-access option specified.

AFTNULL
This bit signifies that the file is $NULL and that there are no control blocks.

AFTPACBV
This is the vector of the Physical ACB (PACB). Note that a PACB exists for all files except $NULL.

AFTTYPE
This is the AFT entry type number. At present the following entry types are defined:

0 - file system
1 - remote file
2 - DS (no-wait I/O disallowed)
3 - DS (no-wait I/O allowed)
4 - CS
5 - CS (AUTO DIAL)
6 - KSAM
8 - message file

PCBX FOR CORE RESIDENT SYSTEM PROCESS STACKS

```
  |-----------------------------------------------| -------------
 0|           DL-a (Seq Rel DL Value)             |0        |
  |-----------------------------------------------|         |
 1|           DB-a (Seq Rel DB Value)             |1        |
  |-----------------------------------------------|         |
 2|         USER ATTRIBUTES (always -1)           |2        |
  |-----------------------------------------------|         |
 3|         0            |    INPUT DEV LDEV      |3        |
  |-----------------------------------------------|    PXGLOB
 4|         0            |    OUTPUT DEV LDEV     |4        |
  |-----------------------------------------------|         |
 5|                      0                        |5        |
  |-----------------------------------------------|         |
 6|   0    | D| I|              0                 |6        |
  |-----------------------------------------------|         |
 7|                      0                        |7        |
  |-----------------------------------------------| ------------
10|             PXFIXED SIZE (c-b)                |8        |
  |-----------------------------------------------|         |
11|              RELATIVE S (S-DB)                |9        |
  |-----------------------------------------------|         |
12|              RELATIVE Z (Z-DB)                |10       |
  |-----------------------------------------------|         |
13|              INITIAL Q (Q-DB)                 |11       |
  |-----------------------------------------------|         |
14|             RELATIVE DL (DB-DL)               |12   PXFIXED
  |-----------------------------------------------|         |
15|       GENERAL RESOURCE CAPABILITY(-1)         |13       |
  |-----------------------------------------------|         |
16|                  RESERVED                     |14       |
  |-----------------------------------------------|         |
17|                     0                         |15       |
  |-----------------------------------------------|         |
20|                    DL-c                       |16       |
  |-----------------------------------------------|         |
21|                    DL-b                       |17       |
  |-----------------------------------------------|
22|                    DL-a                       |18
  |-----------------------------------------------|-------------
```

NOTES: 1. there is no PXFILE area.
       2. the PXFIXED area is much smaller than a normal PCBX.

PROCESS TO PROCESS COMMUNICATION TABLE

This table is used as the communication link by which father and son processes communicate with one another via the mailbox scheme. This table contains two words per entry and is indexed by PCB# (entry index 0 is meaningless). Each two word entry of index N essentially relates where, as well as how much, mail may be found for a process N with respect to communications between N and his father process.

ENTRY FORMAT

```
       |--------------------|
word 0 |     WORD COUNT     |
       |--------------------|
word 1 | MAIL WORD OR DST#  |
       |--------------------|

where word 0 = the # of mail words to
               be transferred.
      word 1 = the only word of mail
               itself if word 0 = 1
                   otherwise
               it contains the DST# of
               the extra data segment
               where "word count" words
               of mail exist.
```

NOTE: Assume process S is the son of process F. Then the process to process communication table index which will be used for mailbox commun cation between son S and father F will be that of the son (i.e. S).

SUB-SYSTEM RESERVED DL AREA

```
        |-----------------------------------------------|
        |                                               |
        ~                                               ~
                       REMAINING DL AREA
        ~                                               ~
        |                                               |
        |-----------------------------------------------|
  DB-12|           RESERVED FOR SORT/MERGE             |DB-10
        |-----------------------------------------------|
  DB-11|         RESERVED FOR TRACE & TOOLBOX          |DB-9
        |-----------------------------------------------|
  DB-10|        EXTERNAL PLABEL OF OUTER BLOCK         |DB-8
        |-----------------------------------------------|
   DB-7|      RESERVED FOR TRACE & SYMBOLIC DEBUG      |DB-7
        |-----------------------------------------------|
   DB-6|             DB ADDRESS OF STLT                |DB-6
        |-----------------------------------------------|
   DB-5|             RESERVED FOR COBOL                |DB-5
        |-----------------------------------------------|
   DB-4|             RESERVED FOR COBOL                |DB-4
        |-----------------------------------------------|
   DB-3|             RESERVED FOR COBOL                |DB-3
        |-----------------------------------------------|
   DB-2|        RESERVED FOR FORMATTER & PASCAL        |DB-2
        |-----------------------------------------------|
   DB-1|             DB ADDRESS OF FLUT                |DB-1
        |-----------------------------------------------|
        |                                               |
        ~                                               ~
                            DB AREA
        ~                                               ~
        |                                               |
        |-----------------------------------------------|
```

FORTRAN LOGICAL UNIT TABLE (FLUT)

The segmenter is responsible for the preparation and initialization of a Fortran logical unit table. This is done when a program is prepared if that program contains at least one program unit that references a logical unit. The location of the FLUT is in the secondary DB area and the address of this location is contained in DB-1.

The FLUT is formatted as per the following example:

```
                         |-------|
                    DB-1 |   X   |
                         |-------|

                         |-------|
                    DB+X | 3 | 0 |
                         |---|---|
                         | 4 | 0 |
                         |---|---|
                         | 5 | 0 |
                         |---|---|
                         | 7 | 0 |
                         |---|---|
                         |10 | 0 |
                         |---|---|
                         |255|///|
                         |-------|
                           ^   ^
              -------------|   |------------------
              |                                  |
          1st BYTE                               2nd BYTE
List of the logical unit numbers     The MPE file number (as returned
referred to in this Fortran-         by FOPEN) used in accessing the
produced program.                    file.  Zero if file not open.
(255 terminates).                    Filled in by formatter as each
                                     l.u. is initially referenced.
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
        |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
        |                                               |
        |-----------------------------------------------|
        |                                               |
        |-----------------------------------------------|
        |                                               |
        |-----------------------------------------------|
        |                                               |
        |-----------------------------------------------|
        |                                               |
        |-----------------------------------------------|
        |                                               |
        |-----------------------------------------------|
```

JMAT - JOB MASTER TABLE STRUCTURE

```
SIR = 15(10) = %17
DST = 25(10) = %31                                 ZEROTH
                                                    ENTRY
                                                      |
                                                      |
                                                      |
       0 1 2 3 4 5 6 7 8 9101112131415                |
      |-------------------------------| ------------------------
     0|   MAXSIZE     |    CURSIZE    |0  max JMAT size (words/128)
      |-------------------------------|   current JMAT size (words/128)
     1| VMOUNT INFO   |  ENTRY SIZE   |1  :VMOUNT state saved for WARMSTARTs
      |-------------------------------|   JMAT entry size (26)
     2|         ENTRY POINTER         |2  DB pointer to first entry (26)
      |-------------------------------|
     3|   SCHEDULING HEAD POINTER     |3  DB pointer to word 0 of head
      |-------------------------------|   entry in scheduling queue
     4|   SCHEDULING TAIL POINTER     |4  DB pointer to word 0 of tail
      |-------------------------------|   entry in scheduling queue
     5| TY|        SCOUNTER           |5  next assignable session #, TY=1
      |-------------------------------|
     6| TY|        JCOUNTER           |6  next assignable batch #, TY=2
      |-------------------------------|
     7|LG|SEC|///////////////JOBFENCE|7  LG=1, logoff in progress
      |-------------------------------|   SEC=0,high;=3,low JOBSECURITY
    10|            SLIMIT             |8  maximum number sessions   C E
      |-------------------------------|                           \ U X
    11|             SNUM              |9  current number sessions | R E
      |-------------------------------|                           | R C
    12|            JLIMIT             |10 maximum # batch jobs    > E U
      |-------------------------------|                           | N T
    13|             JNUM              |11 current # batch jobs    | T I
      |===============================|                           / L N
    14|                               |12                           Y G
      |                               |
    15|           WORKAREA            |13
      |            (14WDS)            |
    16|                               |14
      ~                               ~
      ~                               ~
    31|                               |25
      |===============================| ------------------------------
    32|                               | 26             ^
      |                               |                |
      |                               |                |
      |                               |
      ~                               ~
```

```
JMAT (CONT.)
----

                                                    ENTRY 1
    ~                               ~
    |                               |
    |                               |                  |
    |                               |                  |
  63|                               |51                v
    |===============================| -------------------------------
    |                               |
    |                               |
    |                               |
    ~                               ~
    ~                               ~
    |                               |   LAST
    |                               |   ENTRY
    |===============================|


    SCHEDULING QUEUE
    ----------------
    WAITING SESSIONS
       FIFO WITHIN HIPRI/INPUT PRIORITY
   [ERROR JOBS       ]
   [   FIFO          ]
    WAITING JOBS
       FIFO WITHIN HIPRI/INPUT PRIORITY
```

JMAT - Job Master Table Entry

```
                            1 1 1 1 1 1
    0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
   |-------------------------------|
 0|  state    :D|I:G:A|U:C: INPRI  | 0  state
   |-------------------------------|       0 = free entry
 1| ty:   job/session number       | 1     1 = introduced, in
   |-------------------------------|           STARTDEVICE
 2|4                               | 2  %40 = waiting, job in
 3|         user name              | 3         scheduling queue
 4|                                | 4  %60 = initial, UCOP
 5|                                | 5         has created JSMP
   |-------------------------------|       2 = executing, JSMP
 6|12                              | 6         finished initial.
 7|         account name           | 7     3 = terminating.
10|                                | 8     4 = suspended.
11|                                | 9     D = duplicative
   |-------------------------------|       I = interactive
12|24                              |10     {G = group password
13|         job name               |11     {(QUIET mode, if state=2)
14|                                |12     {A = account password
   |                               |        (STDLIST DELETE, if state=2 or 3)
15|                                |13     {U = user password
   |-------------------------------|       {0 = password validated (STARTDEVICE)
16|32                              |14     {1 = must validate
17|         group logon name       |15     {    password  (INITJSMP)
20|                                |16
21|                                |17
   |-------------------------------|       C = JLIST is device
22| JIN device    : JLIST device   |18         class index
   |-------------------------------|
23|   Julian date (CALENDAR)       |19
   |-------------------------------|       ty = 1 - session
24|   time (CLOCK)                 |20            2 - job
25|                                |21
   |-------------------------------|
26|  main pin      :    XPRI       |22
   |-------------------------------|
27| CPU lim. (0 deflt, -1 no lim.) |23
   |-------------------------------|
30|S|R:N:FT :OUTPRI : NUMCOPIES    |24    ORIGJIN/ORIGJLIST is
   |-------------------------------|      used as a scheduling
31|    ORIGJIN    : ORIGJLIST      |25    link by UCOP (state=
   |-------------------------------|      %40). DB rel. ptr. to
    0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5       next entry. Last entry
                            1 1 1 1 1 1   in list contains 0.
                                        FT = funny terminal
     R = RESTART                             00 - regular term.
     N = SEQUENCED                           01 - regular term.,
     S = ORIGJIN is spooled.                      special logon
                                             10 - APL term.
                                             11 - APL term.
```

JOB STATES

JOB STATES - JMAT ENTRY WORD 0.(0:6)

SHOWJOB - Displays job states by scanning JMAT DST (%31)

LOGON USES ALL STATES EXCEPT "SUSPEND"

STATE NO.	STATE NAME	PROCESS	SEGMENT	PROCEDURE(S)
1	INTRO	DEVREC JSMP SPOOLER	NURSERY	STARTDEVICE ->PUTJMAT ->ALLOCENTRY IN SEGMENT ALLOCUTIL
%40	WAIT	DEVREC JSMP SPOOLER	NURSERY \ SPOOLING /	STARTDEVICE ->SCHEDULEJOB SPOOLSTUFFIN ->SCHEDULEJOB
%60	INITIALIZATION	UCOP	UCOP	LAUNCHJOB
2	EXEC	JSMP	NURSERY	INITJSMP
3	TERMINATING	JSMP	MORQUE	TERMINATE ->EXPIRE -> CLEANUPJOB
0	FREE ENTRY	JSMP	MORQUE	TERMINATE ->EXPIRE -> CLEANUPJOB ->DEALLOCENTRY IN ALLOCUTIL
4	SUSP	JSMP	OPLOW	CXBREAKJOB

For states INTRO and WAIT,

DEVREC	=>	logon command originated on terminal or other unspooled device.
SPOOLER	=>	logon command originated on spooled device.
JSMP	=>	logon command is the result of the execution of a :STREAM command. (This also includes USER processes which have done programmatic :STREAMs.)

JPCNT - JOB PROCESS COUNT TABLE

(1 Entry/Running Job)

CORE RESIDENT

SYSGLOB BASE = DB+13(%15)
DST = 24(10)
SIR = 13(10)

```
          <-----BYTE------>
          |---------------|
          |     MAXI      |<-----maximum # of running jobs (# of bytes-3)
          |---------------|
          |               |<-----total number of free entries
   ----   |---------------|
    ^     |               |
    |     |               |
    |     |---------------|
    |     | 254 (%376)    |<-----free entry
    |     |---------------|
  MAXI    |   0 (%0)      |<------allocated entry
    |     |---------------|
    |     ~               ~
    |     ~               ~
    |     |               |
    v     |               |
  -----   |---------------|
          | 255 (%377)    |<-----free list terminator
          |---------------|
          |  GLOBAL RIN   |
          |  FLAG TABLE   |
          |---------------|
```

A JPCNT entry must be allocated before the main process can be procreated.

The job SIR (PXGJSIR) = some base+JPCNT index.

NOTE: This table is completely byte oriented with each entry consisting of one byte. Entries are taken from available pool on a "first found" basis. 254 (376 octal) in a byte denotes a free entry. 255 (377 octal) denotes the end of table.

GLOBAL RIN FLAG TABLE

This table is a bit table which immediately follows the "free list terminator" byte. It is initialized to 0 and is indexed by JPCNT index for each job. When any process in a job/session locks a global rin, the appropriate bit is turned on.

JCUT - JOB CUTOFF TABLE

1 Entry/ CPU-limited Job

CORE RESIDENT

SYSGLOB BASE = DB+11(%13)
DST = 36(10)
SIR = 14(10)
SYSGLOB + %117 = default
CPU time limit for jobs

```
       0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|   -------------
-----|                   FREE HEAD                     |
|    |-----------------------------------------------|
|    |  # OF REAL ENTRIES   |     ENTRY SIZE (3)     |   HEADER ENTRY
|    |-----------------------------------------------|
| ---|           POINTER TO LAST ENTRY (0)           |
| |  |-----------------------------------------------|   ------------
| |  ~                                               ~
| |
| |  ~                                               ~   TYPICAL ENTRY
| |  |-----------------------------------------------|   ------------
| |  |                   JCUTCPUL                    |    time limit
| |  |-----------------------------------------------|    (seconds)
| |  |                                               |
| |  |                   JCUTCPUC                    |    time count
| |  |                                               |    (msec)
| |  |-----------------------------------------------|   ------------
| |  |                                               |
| |  ~                                               ~
| |
| |  ~                                               ~
| |  |                                               |
| |  |-----------------------------------------------|   ------------
--|->| POINTER TO NEXT FREE ENTRY (END OF LIST = 0)  |
  |  |-----------------------------------------------|
  |  |                                               |    FREE ENTRY
  |  |-----------------------------------------------|
  |  |                                               |
  |  |-----------------------------------------------|   ------------
  -->|                  LAST ENTRY                   |
     |-----------------------------------------------|
     |                                               |
     |-----------------------------------------------|
     |                                               |
     |-----------------------------------------------|
```

```
       JIT -Job Information Table
       --------------------------          JIT DST is word 6 in PXGLOB
                           1 1 1 1 1 1
     0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
    |-------------------------------|
   0|      not used - 0             | 0
    |-------------------------------|
   1|   6       :       JIT DST     | 1
    |-------------------------------|
   2|  pointer to job info        8 | 2
    |-------------------------------|
   3|  pointer to acct info      48 | 3
    |-------------------------------|
   4| pointer to reserved area   59 | 4
    |-------------------------------|
   5|    association table index    | 5
    |-------------------------------|
   6|                             :F| 6   F - Job/Session-wide
    |-------------------------------|         FPMAP option flag
   7|         not used              | 7       (JSFPMAP)
    |-------------------------------|
  10|                             7 | 8   ty - 1 = Session
    |-------------------------------|          2 = Job
  11|ty :    job number             | 9
    |-------------------------------|
  12| JITMAXP        :   JITMPN     |10   JITMAXP - MAXJOBPRI capability
    |-------------------------------|     JITMPN  - Job main PIN.
  13|EOF:       not used            |11   JITEOF  - used by FCLOSE to tell CI
    |-------------------------------|        that a $STDIN(X) file was closed
  14|        DS DATASEG             |12      w/out encountering an EOF.
    |-------------------------------|        (0:1)=$STDIN, (1:1)=$STDINX
  15|           JITASEC             |13
    |-------------------------------|
  16|        JITGSEC (2 words)      |14
    |        group security         |
    |-------------------------------|
  20|        JITHAN (4 words)       |16
    |        account name           |
    |-------------------------------|
  24|        JITHGN (4 words)       |20
    |        home group             |
    |-------------------------------|
  30|        JITLGN (4 words)       |24
    |        log-on group           |
    |-------------------------------|
    +                               +
     0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
                           1 1 1 1 1 1
```

JIT - Continued

```
                      1 1 1 1 1 1
   0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
  |-------------------------------|
34|                               |28
35|            JITUN              |29
36|          user name            |30
37|                               |31
  |-------------------------------|
40| pointer to JITAIP         53  |32
  |-------------------------------|
41|P|M: pointer to JITGIP     55  |33    P - Group's home volume is
  |-------------------------------|          a private volume
42|            LATTR              |34    M - Private volume mounted
43|       local attributes        |35        (i.e. group bound to home
  |-------------------------------|          volume set), JITGIP = 57
44|            PASSF              |36
45|       passed file pointer     |37
  |-------------------------------|
46|            UCAP               |38
47|       user capability *       |39
  |-------------------------------|
50|                               |40    For bit mask definitions, see
51|       allow mask              |41    OPCOMMAND listing or COMSEARCH
52|                               |42    of segment CIINIT.
  |-------------------------------|
53|       local RIN pointer       |43
  |-------------------------------|
54|                               |44
55|            JITJN              |45
56|          job name             |46
57|                               |47
  |-------------------------------|
  +                               +
   0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
                      1 1 1 1 1 1
```

JIT - Continued

```
                       1 1 1 1 1 1
    0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
   |-------------------------------|
60 |                             3 |48   Accounting Info
   |-------------------------------|
61 | JITCREC - # of creations      |49
   |-------------------------------|
62 |          JITCPUC              |50
63 |      cpu milliseconds         |51
   |-------------------------------|
64 |  not used     :    HIPRI      |52     HIPRI - highest job priority
   |-------------------------------|
65 |              0                |53
66 |           JITAIP              |54
   |-------------------------------|
67 |              0                |55
70 |           JITGIP              |56  System volume set
   |-------------------------------|
71 |       0       :    MVTABX     |57
72 |           JITGIP              |58  Mounted private volume set
   |-------------------------------|
73 |                             1 |59
   |-------------------------------|
74 |                             0 |60
   |-------------------------------|
    0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
                       1 1 1 1 1 1
```

* THE FORMAT FOR UCAP (%46-47) IS AS FOLLOWS:

```
       |-----------------------------------------------|
       | 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|
       |-----------------------------------------------|
WORD1  |SM|AM|AL|GL|DI|OP|CV|UV|LG|           |CS|ND|SF|
       |-----------------------------------------------|
WORD2  |                     |BA|IA|PM|     |MR|  |DS|PH|
       |-----------------------------------------------|
```

JDT - JOB DIRECTORY TABLE

```
             |-----------------------|
         0   |   MAX SEG SIZE(WDS)   |            1 entry per job
             |-----------------------|            DST # in PXGLOB
         1   |    POINTER TO JDSD    |
             |-----------------------|
         2   |    POINTER TO JTFD    |
             |-----------------------|
         3   |    POINTER TO JFEQ    |
             |-----------------------|
         4   |    POINTER TO JLEQ    |
             |-----------------------|
         5   |  POINTER TO JJCW      |
             |-----------------------|
         6   |POINTER TO FREE SPACE  |
             |-----------------------|
             ~      WORK   AREA      ~
             ~        15 words       ~
             |-----------------------|
 JDSJNUM     |TY|       NUM          |    job number
             |-----------------------|
             |///////////|   JSMPIN  |    main process number
             |-----------------------|
             ~       JOB DATA        ~
 JDSD        ~   SEGMENT DIRECTORY   ~
             |-----------------------|
             |                       |
             ~     JOB TEMPORARY     ~     |-----------------------|
 JTFD              FILE DIRECTORY          |ENTRY      |NAME       |
             ~                       ~     | SIZE (WDS)| SIZE (WDS)|
             |                       |     |-----------------------|
             |-----------------------|     |    C1     |    C2     |
             |                       |     |-----------------------|
             |                       |  ---
             ~       JOB FILE        ~  |  |-----------------------|
 JFEQ              EQUATION TABLE       |  |    CN     |   (%40)   |
             ~                       ~  |  |-----------------------|
             |-----------------------|  |  |                       |
             ~       JOB LINE        ~  |  |        ENTRY          |
 JLEQ        ~   EQUATION TABLE      ~  |  |     INFORMATION       |
             |-----------------------|  |  |                       |
             |                       |  |  |-----------------------|
             ~   JOB CONTROL WORD    ~  |
             ~   TABLE    (JJCW)     ~  The name is a
             |-----------------------|  concatenation of up to 3 subnames.
             |                       |  Bit 0 of the 1st character of each
             |      FREE SPACE       |  subname is 1.
             |-----------------------|
```

JOB DATA SEGMENT DIRECTORY ENTRY - (IN JDT)

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|           4           |           1           |
|-----------------------------------------------|
|                  SEGMENT ID                   |
|-----------------------------------------------|
|         EXTRA DATA SEGMENT DST INDEX          |
|-----------------------------------------------|
|          # OF PROCESSES ACCESSING             |
|-----------------------------------------------|
```

JOB TEMPORARY FILE ENTRY - (IN JDT)

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| ENTRY SIZE (WORDS)    |  NAME SIZE (WORDS)    |
|-----------------------------------------------|
~                                               ~
           NAME-ACTUAL FILE DESIGNATOR         --------- Name is a
~                                               ~  concatenation of up
|-----------------------------------------------|  to three subnames.
|   VOLUME POINTER      |                       |  Bit 0 of the first
|-----------------------|                       |  character of each
|               FILE LABEL POINTER              |  subname is 1.
|-----------------------------------------------|
```

FILE EQUATION TABLE ENTRY - (IN JDT)

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| ENTRY SIZE (WORDS)    |   NAME SIZE (WORDS)    |
|-----------------------------------------------|
~                                               ~

                     NAME
~             (FORMAL DESIGNATOR)               ~
|-----------------------------------------------|
|                    PMASK                      |
|                                               |*
|                                               |
|-----------------------------------------------|
| NAME LENGTH (BYTES)   | DEVICE LENGTH (BYTES) |
|-----------------------------------------------|
~                                               ~

           NAME-ACTUAL DESIGNATOR
~           (may not be present)                ~
|-----------------------------------------------|
~                                               ~

              DEVICE/CLASS NAME
~           (may not be present)                ~
|-----------------------------------------------|
|                  FOPTIONS                     |*
|-----------------------------------------------|
|                  AOPTIONS                     |*
|-----------------------------------------------|
|       #BUFFERS         |  INIT ALLOC  |D |T |S |<---disposition
|-----------------------------------------------|     BIT13 DEL
|                 RECORD SIZE                   |     BIT14 TEMP
|-----------------------------------------------|     BIT15 SAVE
| # EXTENTS    |/////////|      BLOCK FACTOR     |
|-----------------------------------------------|
|                    FILE                       |
|-----------------------------------------------|
|                    SIZE                       |
|-----------------------------------------------|
|                  FILE CODE                    |
|-----------------------------------------------|
|  OUTPRI  |     NUMCOPIES        |             |
|-----------------------------------------------|
|  REF COUNT     |     # OF USER LABELS         |
|-----------------------------------------------|
|         LENGTH  FORMS=/LABEL=                 |
|-----------------------------------------------|
|                FORMS/LABEL                    |
~                  ARRAY                        ~
~                                               ~
|-----------------------------------------------|
```

JOB LINE EQUATION ENTRY

```
  |-----------------------------------------------|
  |   ENTRY SIZE (WORDS)  |   DESIG. SIZE (WORDS) |
  |-----------------------------------------------|
  |                     FORMAL                    |
  ~                 LINE DESIGNATOR               ~
  ~                  (1-4 WORDS)                  ~
  |-----------------------------------------------|
 0|                     PMASK1                    |0
  |-----------------------------------------------|
 1| REF CNT        5|P |          PMASK2          |1 P=FLAG
  |-----------------------------------------------|
 2|     NAME LENGTH       |        DEV LENGTH     |2
  |-----------------------------------------------|
 3|                                               |3
  |                                               |
 4|                      NAME                     |4
  |                                               |
 5|      ( END OF LEQ ENTRY IF NON-BLANK )        |5
  |                                               |
 6|                                               |6
  |-----------------------------------------------|
 7|                                               |
  |                                               |
10|                                               |8
  |                     DEVICE                    |
11|                                               |9
  |                                               |
12|                                               |10
  |-----------------------------------------------|
13|                     PMASK3                    |11
  |-----------------------------------------------|
14| DRIVER NAME LENGTH    |                       |12
  |-----------------------------------------------|
15|                                               |13
  |                                               |
16|                                               |14
  |                  DRIVER NAME                  |
17|                                               |15
  |                                               |
20|                                               |16
  |-----------------------------------------------|
21|                   LIST PNTR                   |17
  |-----------------------------------------------|
22|                    COPTIONS                   |18
  |-----------------------------------------------|
23|                    AOPTIONS                   |19
  |-----------------------------------------------|
24|                    DOPTIONS                   |20
  |-----------------------------------------------|
```

JLEQ ENTRY (CONT.)

```
   |------------------------------------------------|
 25|               NUMBER OF BUFFERS                |21
   |------------------------------------------------|
 26|             BUFFER SIZE IN WORDS               |22
   |------------------------------------------------|
 27|              INSPEED (2 words)                 |23
   |------------------------------------------------|
 31|              OUTSPEED (2 words)                |25
   |------------------------------------------------|
 33|                 POLL REPEAT                    |27
   |------------------------------------------------|
 34|                  POLL DELAY                    |28
   |------------------------------------------------|
 35|                 C TRACE INFO                   |29
   |------------------------------------------------|
 36|                LOCAL ID PNTR                   |30 \
   |------------------------------------------------|    |
 37|               REMOTE ID PNTR                   |31  |
   |------------------------------------------------|    |
 40|                SUPLIST PNTR                    |32  | REL TO ORIG
   |------------------------------------------------|    | OF LEQ ENTRY
 41|               PHONE LIST PNTR                  |33  |
   |------------------------------------------------|    |
 42|                POLLIST PNTR                    |34  |
   |------------------------------------------------|    |
 43|               MISC ARRAY PNTR                  |35 /
   |------------------------------------------------|
```

```
      JJCW     JOB CONTROL WORD TABLE
      ----     --- ------- ---- -----

      |---------------------------------|   Name may be any alpha-
      | NAME SIZE (BYTES)|              |   numeric string, begin-
      |------------------|              |   ning with an alpha,
      |                                 |   between 1 and 255 char-
      ~                                 ~   acters long.
      ~              NAME               ~
      |                                 |   TY  00 = OK
      |---------------------------------|       01 = WARN
      | TY |        MODIFIER            |       10 = FATAL
      |---------------------------------|       11 = SYSTEM

      MODIFIER = VALUE FROM 0 TO %377777
```

AOPTIONS AND FOPTIONS WORD BREAKDOWN

```
OPTION WORD 2           OPTION WORD 1
 (AOPTIONS)              (FOPTIONS)

 0|---|                  0|---|
  | 0 |                   | 0 |
  |   |                   |   |
  | 0 |                   | 0 |
  |   |                   |---|
  | 0 |                  2|   |
  |---|                   |   |file type
 3|   |copy              3|   |
  |---|                   |---|
 4|   |no-wait            | 0 |
  |---|                   |---|
 5|   |                  5| 0 |disallow files
  |   |multi-             |---|
 6|   |access            6|   |labelled tape
  |---|                   |---|carriage
 7|   |inhibit buff.     7|   |control
  |---|                   |---|
 8|   |                  8|   |
  |   |exclusive          |   |record format
 9|   |                  9|   |
  |---|                   |---|
10|   |dynamic locking  10|   |
  |---|multi-             |   |default
11|   |record             |   |designator
  |---|                   |   |
12|   |                 12|   |
  |   |                   |---|
  |   |access type      13|   |ascii/binary
  |   |                   |---|
  |   |                 14|   |
  |   |                   |   |domain
15|   |                 15|   |
  |---|                   |---|
```

PMASK WORD BREAKDOWN

```
                     --------------------

                  --------- PMASK WORD 2
                  |    ----- PMASK WORD 1
                  |    |
               |---|---|0
FILE TYPE      |   |   |BLOCK FACTOR
               |---|---|
LABELLED TAPE  |   |   |RECSIZE
               |---|---|
FRMS MESSAGE   |   |   |DISPOSITION
               |---|---|
USER LABELS    |   |   |NUMBUFFERS
               |---|---|
              4|   |   |INHIBIT BUFFERING
               |---|---|
              5|   |   |EXCLUSIVE
               |---|---|
POINTER ENTRY  |   |   |MULTI-RECORD
               |---|---|
DYN.LOCKING    |   |   |ACCESS TYPE
               |---|---|
WAIT,NOWAIT    |   |   |COPY,NOCOPY
               |---|---|
MULTI ACCESS   |   |   |CARRIAGE CONTROL
               |---|---|
NUMCOP         |   |   |RECORD FORMAT
               |---|---|
OUTPRI         |   |   |DEFAULT DESIGNATOR
               |---|---|
FILECODE       |   |   |ASCII/BINARY
               |---|---|
FILESIZE       |   |   |DOMAIN
               |---|---|
NUMEXTS        |   |   |DEVICE
               |---|---|
INIT ALLOC     |   |   |NAME
               |---|---|
                       15

      1->info present
      0->info absent
```

UCOP REQUEST QUEUE (DST#9)

```
                   |------------------------------------------------|
                  0|              MAX# REQ ENTRIES N/2              |
                   |------------------------------------------------|
                  1|          DOUBLE POINTER TO NEXT AVIL           |------
                   |------------------------------------------------|      |
                  2|          DOUBLE POINTER TO NEXT REQ            |---   |
                   |------------------------------------------------|   |  |
                  3|                       0                        |   |  |
                -- |------------------------------------------------|   |  |
               |   |                                                |   |  |
               |   |                                                |   |  |
               |   |                                                |   |  |
               |   |                                                |   |  |
               |   |                                                |   |  |
               |   |                                                |   |  |
               |   |                                                |   |  |
               |   |                                                |   |  |
               |   |                                                |   |  |
               |   |------------------------------------------------|   |  |
               |   |                     REQ 1                      |<--   |
               |   |------------------------------------------------|      |
               |   |                     REQ 2                      |      |
               |   |------------------------------------------------|      |
               |   |                       .                        |      |
               |   |                       .                        |      |
               N   |                       .                        |      |
               WRDS|                       .                        |      |
               |   |                       .                        |      |
               |   |                       .                        |      |
               |   |------------------------------------------------|      |
               |   |                     REQ N                      |      |
               |   |------------------------------------------------|      |
               |   |                                                |      |
               |   |                                                |<-----
               |   |                                                |
               |   |                                                |
               |   |                                                |
               |   |                                                |
               |   |                                                |
               |   |                                                |
               |   |                                                |
                -- |------------------------------------------------|
```

```
             UCOP ENTRY FORMAT                      Request Codes
             -----------------                      -------------
                                                   0 null
                                                   1 null
 0                                      12-15
|-----------------------------------------------|
|////////////////////////////////////|    2     |
|-----------------------------------------------|  2  process deletion
|///////////////////////|          PIN          |
|-----------------------------------------------|
           0-7                     8-15
```

CHAPTER 9 RELOCATABLE OBJECT CODE

USL FILES - GENERAL INFO

- USL record length 128 words always.
- Layout of doubleword disc addresses

```
|-------------------------------------------------|
|                           |      WORD #         |
|      25-BIT RECORD #      |   WITHIN RECORD     |
|-------------------------------------------------|
0                        24 25                  31
```

- Hash links join all entries with the same hash key regardless of type.
- Linear lists terminate with a zero link
- Circular lists containing only the list head point directly to themselves.
- Single-word disc addresses

```
|-------------------------------------------------|
|                           |      WORD #         |
|      9-BIT RECORD #       |   WITHIN RECORD     |
|-------------------------------------------------|
0                         8 9                   15
```

Uninitialized fields are reserved for future use and should be set to zero.

RECORD 0 AND OVERALL USL FILE FORMAT

```
  -----------                              NOTE:
 0|   LID    | 0   LOADER ID           S.A. = Starting Address
  |----------|
 1|   NE     | 1   NR. DIRECTORY ENTRIES
  |----------|
 2|   DL     | 2   DIR. LENGTH
  |----------|
 3|  SUMDG   | 3   TOTAL DIR. GARBAGE
  |----------|
 4|   NDG    | 4   NR. DIR. GARB. ENTRIES
  |----------|
 5|  SABDL   | 5   S.A. BLOCK DATA LIST
  |----------|
 6|  SAIPL   | 6   S.A. INTERRUPT PROC. LIST
  |----------|
 7|  SASL    | 7   S.A. SEGMENT LIST
  |----------|
  |----------|                       USL FILE FORMAT (CONT.)
```

```
  10|   FL     | 8   FILE LENGTH          ---------------
  11|          | 9
    |----------|
  12|   SAAD   | 10  S.A. AVAIL. DIR.
    |----------|
  13|   ADL    | 11  AVAIL. DIR. LENGTH
    |----------|
  14|   SAI    | 12  S.A. INFO BLOCK
  15|          | 13
    |----------|
  16|    IL    | 14  INFO BLOCK LENGTH
  17|          | 15
    |----------|
  20|   SAAI   | 16  S.A. AVAIL. INFO
  21|          | 17
    |----------|
  22|    AIL   | 18  AVAIL. INFO LENGTH
  23|          | 19
    |----------|
  24| TOTAL    | 20  TOTAL INFO GARBAGE
  25|  I.G.    | 21
    |----------|
  26|   NIG    | 22  NR. INFO  GARB. ENTRIES
    |----------|
  27|          | 23
    |          |
  30|          | 24
    |          |
  31|          | 25
    |          |
  32|          | 26
    |          |
  33|          | 27
    |          |
  34|          | 28
    |          |
  35|          | 29
    |          |
  36|          | 30
    |          |
  37|          | 31
    |          |
  40|          | 32
    |----------|
  41|   HL     | 33  HASH LINKS
    |     0    |
    |----------|
    |    .     |
    |    .     |
    |----------|
 177|  HL      | 127
    |    94    |
    -----------
```

USL FILES - GENERAL INFO (CONT.)

```
                     0-------------0      ^
                      |           |       |
                      | RECORD 0  |       |
                      |           |       |
                      |           |       |
                   177-------------127    |
                                          |
                                          |
      ---          200-------------128    |
       |              |           |       |
       |              | DIRECTORY |      32K
       |              |           |      MAX
       DL             |  ENTRIES  |       |
       |              |           |       |
       |              |           |       |
      ---             -------------       |
       |                    |             |
      ---             -------------       |
       |         SAAD |           |       |
       |              | AVAILABLE |       |
      ADL             | DIRECTORY |       |
       |              |           |       |
      ---             -------------       |

                 ____________________________

      ---        SAI* -------------
       |              |           |
       |              |   INFO    |
       IL             | (HEADERS) |
       |              |  (CODE)   |
       |              |           |
       |              |           |
      ---             -------------
                            |
      ---        SAAI -------------
       |              |           |
      AIL             | AVAILABLE |
       |              |   INFO    |
       |              |           |
      ---        FL-1 -------------
```

*SAI MUST BE ON A RECORD BOUNDRY

NOTE: ALL ADDRESSES IN RECORD 0 ARE WORD ADDRESSES.

USL FILES - GENERAL INFO (CONT.)

```
SASL--->----------          -----------          ----------
        |  SEGL  | -------->|    SEGL |--------->|   0    |
        |        |          |         |          |        |
        | SEG.A  |          | SEG.K   |          | SEG.B  |
        |        |          |         |          |        |
     ---|        |---     --|SUBL     |<--    ---|        |---
        ----------  |     | -----------  |    |  ----------
                    |     |              |    |
                    |     |              |    |
---------------------------              --------------------------
|                                                                 |
|                                                                 |
|                                                                 |
|                                                                 |
|       ----------          -----------          ----------       |
|------>|  SUBL  |--------->|     SUBL|--------->|    SUBL|-------|
        |        |          |         |          |        |
        | PROC.C |          | PROC.A  |          | MAIN   |
        |        |          |         |          |        |
     ---|        |---    ---|SECL     |<--    ---|        |---
        ----------  |    |  -----------  |    |  ----------
                    |    |               |    |
                    |    |               |    |
--------------------------               ----------------------------
|                                                                  |
|                   CIRCULAR LINK POINTS TO ITSELF                 |
|                   IF LIST IS EMPTY                               |
|                                                                  |
|                                                                  |
|       ----------          -----------          ---------         |
------->|  SECL  |--------->|     SECL|--------->|    SECL|----------
        |        |          |         |          |        |
        |PROC.A  |          |PROC.A   |          |PROC.A  |
        |      3 |          |      1  |          |      5 |
        |        |          |         |          |        |
        ----------          -----------          ----------


A \                                   PROC C \
K  >SEGMENT NAME ENTRIES              PROC A  >SUBPROGRAM
B /                                   MAIN    / ENTRIES

        A  \
         3 |
        A  |
         1  } SECONDRY ENTRY POINT ENTRIES
        A  |
         5 /
```

DATA DESCRIPTORS, PASSED PARAMETERS

```
  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
---|-|-|-|-|-|-|-|-|-|-|--|--|--|--|--|----
| MODE   | STRUCTURE |       TYPE        |
-----------------------------------------
```

TYPE	WORDS	CODE
NULL		0
LOGICAL	1	1
INTEGER	1	2
BYTE	1/2	3
REAL	2	4
DOUBLE	2	5
LONG	3	6
COMPLEX	4	7
LABEL (SPL)		10
CHARACTER	N/2	11
LABEL (FORTRAN)		12
UNIVERSAL (MATCHES ANY TYPE)		13

STRUCTURE	
SIMPLE VARIABLE	0
POINTER	1
ARRAY	2
PROCEDURE	3

MODE	
NULL	0
VALUE	1
REFERENCE	2
NAME	3

NOTE: A descriptor of 0 results in an automatic match.

ENTRY TYPE 0

GARBAGE

```
 0  1                 10 11     15
----------------------------------
|///|        NW         |   0    |
----------------------------------
|                                |
|                                |
|           GARBAGE              |
|                                |
|                                |
----------------------------------
```

NW - Number of words in this
     block

ENTRY TYPE 1

SEGMENT NAME

```
 0  1            7 8   10 11    15
----------------------------------
|//|         NW          |  1    |
----------------------------------
|             H L                |
----------------------------------
|A |///////| NC          CHAR1   |
----------------------------------
|               .                |
|   (VARIABLE # CHAR. SEE NC)    |
|               .                |
|               .                |
----------------------------------
|  CHAR. NC     |////////////////|
----------------------------------
|             SEGL               |
----------------------------------
| L |          SUBL              |
----------------------------------
```

NW - Number of words in entry
     block

HL - Hash link - points to next
     entry having the same
     hash code

A - Activity bit
    0 if active
    1 if inactive
    (initialize to 0)

Note: An inactive segment
      implies that all entry
      points are inactive

NC - Number of characters in
     name. Max is 16

CHAR. 1 - First character in
          variable field

CHAR. NC - Last character in
           variable field

SEGL - Segment link - points to
       next segment name
       entry

SUBL - Subprogram link - points
       to next entry having
       the same segment name

L - Last entry in list
    0 if not last
    1 if last

CLARIFICATION NOTES ON ENTRY TYPES 2 AND 4

WITH RESPECT TO SPL AND FORTRAN

*ENTRY TYPE 2 SPL O.B.	**ENTRY TYPE 4 SPL PROC	*ENTRY TYPE 2 FORTRAN MAIN	**ENTRY TYPE 4 FORTRAN SUB.
TPDB	0	0	0
TSDB[1,5]	TSDB[1]	TSDB[1,2,3,4]	TSDB[1,2,3,4]
NWPUST	NWPUST	NWPUST	NWPUST
NWSDB[5]	NWO	NWD	NWD

WHERE:
TPDB = Total primary DB length in words
TSDB = Total secondary DB length in words
NWPUST = Number of words in "TRACE" array
NWSDB = Number of words in secondary DB array
NWO = Number of words in own array
NWD = Number of words in data array

Notes:
1. Does not include the length of the STLT
2. Does not include the length of the FLUT
3. Does not include the length of any common array
4. Includes the length of any DB-allocated format array array
5. Are not necessarily equal

In general TPDB and TSDB are summations of storage allocated in the global area of the program's data segment. They are not, however, complete since the compilers are not aware of all storage actually allocated! The STLT and FLUT are examples of this since these tables are constructed by the segmenter. Common arrays also present a problem since their inclusion in TPDB and TSDB might cause their storage requirements to be counted more than once.

ENTRY TYPE 2

```
OUTER BLOCK
 0   1  2  3  4 5 6 7  8     10 11     15
-----------------------------------------
|//|          NW               |    2   |
-----------------------------------------
|                 HL                    |
-----------------------------------------
| A | C | I |///| NC |     CHAR 1       |
-----------------------------------------
|                  .                    |
|       (VARIABLE # CHAR.SEE NC)        |
|                  .                    |
-----------------------------------------
|      CHAR NC      |///////////////////|
-----------------------------------------
| L |              SUBL                 |
-----------------------------------------
| L |              SECL                 |
-----------------------------------------
|                  SSA                  |
-----------------------------------------
|                  SAC                  |
|    RELATIVE TO SAI (SEE RECORD 0)     |
|                                       |
-----------------------------------------
| F | W |          NWC                  |
-----------------------------------------
|                   SE                  |
-----------------------------------------
|                  TPDB                 |
-----------------------------------------
|                  TSDB                 |
-----------------------------------------
|                 NWPUST                |
-----------------------------------------
|               NWD/NWSDB               |
-----------------------------------------
| T |               NH                  |
-----------------------------------------
|                  SAH                  |
|     RELATIVE TO SAI (SEE RECORD 0)    |
-----------------------------------------
|                  HDW                  |
-----------------------------------------
```

ENTRY TYPE 2 (CONT.)

```
-----------------------------------------
|                  .                    |
|                  .                    |
|                  .                    |
-----------------------------------------
|                 HDW                   |
-----------------------------------------
|                  .                    |
|                  .                    |
|                  .                    |
-----------------------------------------
| T |              NH                   |
-----------------------------------------
|                                       |
|                 SAH                   |
|                                       |
-----------------------------------------
|                 HDW                   |
-----------------------------------------
|                  .                    |
|                  .                    |
|                  .                    |
-----------------------------------------
|                 HDW                   |
-----------------------------------------
```

NW - Number of words in entry block.

HL - Hash link - points to next entry with same hash code.

A - Activity bit. 0 if active, 1 if inactive outer block.

C - Callability bit set if entry point is uncallable.

I - Priv mode bit - set if program unit is to be executed in priv mode..

NC - Number of characters in name. Max is 16.

CHAR. 1 - First character in variable field.

CHAR. NC - Last character in variable field.

L - Last entry in list.
0 if not last
1 if last

ENTRY TYPE 2 (CONT.)

SUBL - Subprogram link - points to next entry Entry having the same segment name.

SECL - Secondary entry point list link.

SSA - Program unit starting PB address.

SAC - Starting 8FILE9 address of code module

F - Set if fatal error

W - Set if non-fatal error

NWC - Number of words in code module.

SE - Stack size estimate

TPDB - Total number of words of primary DB to be allocated

TSDB - Total number of words of secondary DB to be allocated.

NWPUST - Number of words in trace array (PUST)

NWD - Number of words in data array (FORTRAN)

NWSDB - Number of words in secondary DB array (SPL)

T - Terminating bit - set if last set of headers in entry

NH - Number of headers

SAH - Starting address of header (relative to SAI)

HDW - Header (pointer)

ENTRY TYPE 3

OUTER BLOCK - SECONDARY ENTRY POINT

```
 0   1  2  3  4 5 6 7  8     10 11     15
-----------------------------------------
|//|            NW              |  3    |
----------------------------------------|
|                 HL                    |
-----------------------------------------
| A | C |//|//|  NC  |      CHAR 1      |
-----------------------------------------
|                 .                     |
|      (VARIABLE # CHAR.SEE NC)         |
|                 .                     |
-----------------------------------------
|     CHAR NC     |/////////////////////|
-----------------------------------------
| L |             SECL                  |
-----------------------------------------
|                 SSA                   |
-----------------------------------------
```

ENTRY TYPE 4

PROCEDURE

```
 0   1  2  3 4567 8                  10 11        15
---|--|--|--|----|-------------------|------------
|//|              NW                 |     4      |
--------------------------------------------------
|                        HL                      |
--------------------------------------------------
|A | C| I| H| NC |          CHAR.1               |
--------------------------------------------------
|                        .                       |
|           (VARIABLE # CHAR. SEE NC)            |
|                        .                       |
--------------------------------------------------
|  CHAR.NC      |////////////////////////////////|
--------------------------------------------------
|L |                SUBL                         |
--------------------------------------------------
|L |                SECL                         |
--------------------------------------------------
|                   SSA                          |
--------------------------------------------------
```

```
ENTRY TYPE 4 (CONT.)
------------

-------------------------------------------------
|                                               |
|                SAC                            |
|                                               |
-------------------------------------------------
|F | W|           NWC                           |
-------------------------------------------------
|                  SE                           |
-------------------------------------------------
|                TPDB                           |
-------------------------------------------------
|                TSDB                           |
-------------------------------------------------
|               NWPUST                          |
-------------------------------------------------
|                NWD/NWO                        |
-------------------------------------------------
|  P  |     NP   |          CN                  |
-------------------------------------------------
|                  TN                           |
-------------------------------------------------
|                  PARM.1                       |
-------------------------------------------------
|                     .                         |
|      (VARIABLE # OF PARMS. SEE CN)            |
|                     .                         |
-------------------------------------------------
|                PARM. NP                       |
-------------------------------------------------
| T|                NH                          |
-------------------------------------------------
|                                               |
|                   SAH                         |
|                                               |
-------------------------------------------------
|                    HDW                        |
-------------------------------------------------
|                     .                         |
|                     .                         |
|                     .                         |
-------------------------------------------------
|                    HDW                        |
-------------------------------------------------
|                     .                         |
|                     .                         |
|                     .                         |
-------------------------------------------------
|                    ETC                        |
-------------------------------------------------
```

ENTRY TYPE 4 (CONT.)

NW - Number of words in entry block
HL - Hash link - points to next entry with same hash code
A - Activity bit. 0 if active, 1 if inactive entry point
C - Callability bit set if entry point is uncallable
I - Priv mode bit. Set if procedure is to be executed in priv mode.
H - Hidden entry point. Set if entry point will not be in
    library directory.
NC - Number of characters in name. Max is 16.
CHAR1 - First character in variable field.
CHAR NC - Last character in variable field.
L - Last entry in list
    0 if not last
    1 if last
SUBL - Subprogram link. Points to next entry having the same segment
    Name
SECL - Secondary entry point list link.
SSA - Unit starting PB address
SAC - Starting (file) address of code module
F - Set if fatal error
W - Set if non-fatal error
NWC - Number of words in code module
SE - Stack size estimate
TPDB - Total number of words of primary DB to be allocated.
TSDB - Total number of words of secondard DB to be allocated.
NWPUST - Number of words in trace array (PUST)
NWD - Number of words in data array (FORTRAN)
NWO - Number of words in own array (SPL)
P - Parm checker
    00 no checking. (Implies NP undefined, FN and PARM's absent)
    01 check procedure type. (Implies NP is undefined and PARM's absent)
    10 check procedure type and number of PARM's (implies PARM's absent)
    11 check procedure type, number of PARM 's and type of each PARM.
NP - Number of PARM's
CN - Character count of PARM's
TN - Terminating bit. Set if last set of headers in entry.
NH - Number of headers
SAH - Starting address of header
HDW - Header (pointer)

ENTRY TYPE 5

PROCEDURE - SECONDARY ENTRY POINT

```
 0  1  2  3 4 5 6 7 8   10 11     15
|--|-----------------|--------|-------|
|//|        NW               |   5   |
|-------------------------------------|
|                   HL                |
|-------------------------------------|
| A| C |//|H |  NC   |     CHAR. 1    |
|-------------------------------------|
|                  .                  |
|  (VARIABLE #CHAR. SEE NC)           |
|                  .                  |
|-------------------------------------|
|     CHAR. NC        |////////////////|
|-------------------------------------|
|L |          SECL                    |
|-------------------------------------|
|             SSA                     |
|-------------------------------------|
```

NW - Number of words in entry block

HL - Hash link - points to next entry with same hash code

A - Activity bit. 0 if active, 1 if inactive entry point

C - Callability bit set if entry point is uncallable.

H - Hidden entry point set if entry point will not be in library directory

NC - number of characters in name, max is 16

CHAR 1 - First character in variable field.

L - Last entry in list
0 if not last
1 if last

SECL - Secondary entry point list link

SSA - Unit starting PB' address

ENTRY TYPE 6

INTERRUPT PROCEDURE

```
| 0|1 |2 |3 |4567|8      10|11    15|
-------------------------------------
|//|       NW             |   6     |
-------------------------------------
|                HL                 |
|                                   |
-------------------------------------
|A | IT  |//| NC |       CHAR.1     |
-------------------------------------
|                 .                 |
|      (VARIABLE # CHAR. SEE NC)    |
|                 .                 |
-------------------------------------
|A | IT  |//| NC |       CHAR.1     |
-------------------------------------
|                 .                 |
|      (VARIABLE # CHAR. SEE NC)    |
|                 .                 |
-------------------------------------
|  CHAR. NC       |//////////////////|
-------------------------------------
|               IPL                 |
-------------------------------------
|               DBS                 |
-------------------------------------
|               SSA                 |
-------------------------------------
|                                   |
|               SAC                 |
|                                   |
-------------------------------------
|F | W|        NWC                  |
-------------------------------------
|T |            NH                  |
-------------------------------------
|                                   |
|               SAH                 |
|                                   |
-------------------------------------
|               HDW                 |
-------------------------------------
|                .                  |
|                .                  |
|                .                  |
-------------------------------------
|               HDW                 |
-------------------------------------
```

ENTRY TYPE 6 (CONT.)

NW - Number of words in entry block

HL - Hash link. Points to next entry with same hash code

A - Activity bit. 0 if active, 1 if inactive entry.

IT - Interrupt procedure type number

NC - Number of characters in name (maximum is 16)

CHAR 1 - First character in variable field.

CHAR NC	Last Character in variable field
IPL	Interrupt procedure link
DBS	Number of words of DB storage required.
SSA	Unit starting PB' address
SAC	Starting (file) address of code module.
F	Set if fatal error
W	Set if non-fatal error
NWC	Number of words in code module
T	Terminating bit. Set if last set of headers in entry.
NH	Number of headers
SAH	Starting address of header.
HDW	Header (pointer)

ENTRY TYPE 7

BLOCK DATA

```
| 0 | 1 | 2 | 3 |4567|8       10|11        15|
----|---|---|---|----|----------|-------------
|///|             NW            |      7     |
---------------------------------------------
|                    HL                      |
---------------------------------------------
| A | F | W |///| NC |         CHAR.1        |
---------------------------------------------
|                    .                       |
|            BLOCK DATA NAME                 |
|                    .                       |
---------------------------------------------
|     CHAR.NC        |///////////////////////|
---------------------------------------------
|                    BDL                     |
---------------------------------------------
|                    CAL                     |
---------------------------------------------
|///////////////| NC |        CHAR.1         |
---------------------------------------------
|                    .                       |
|            COMMON ARRAY NAME               |
|                    .                       |
---------------------------------------------
|      CHAR.NC       |///////////////////////|
---------------------------------------------
| T |                NH                      |
---------------------------------------------
|                    SAH                     |
|                                            |
---------------------------------------------
|                    HDW                     |
---------------------------------------------
|                    .                       |
|                    .                       |
|                    .                       |
---------------------------------------------
|                    HDW                     |
--------------------------------------------|
|                    .                       |
|                    .                       |
|                    .                       |
---------------------------------------------
```

ENTRY TYPE 7 (CONT.)

```
----------------------------------------------
|                    CAL                     |
----------------------------------------------
|///////////////| NC |       CHAR.1          |
----------------------------------------------
|                     .                      |
|            COMMON ARRAY NAME               |
|                     .                      |
----------------------------------------------
|      CHAR.NC        |///////////////////////|
----------------------------------------------
| T |                NH                      |
----------------------------------------------
|                    SAH                     |
|                                            |
----------------------------------------------
|                    HDW                     |
----------------------------------------------
|                    ETC                     |
|                                            |
```

NW	Number of words in block
HL	Hash link. Points to next entry with same hash code.
A	Activity bit. 0 if active, 1 if inactive block.
F	Set if fatal error.
W	Set if non-fatal error.
CHAR 1	First character in variable field.
CHAR NC	Last character in variable field.
BDL	Block data link
CAL	Common array length
T	Terminating bit. Set if last set of headers in entry.
NH	Number of headers.
SAH	Starting address of headers.
HDW	Header (pointer)

ENTRY TYPE 8

PROCEDURE - SECONDARY ENTRY POINT

```
 0  1  2  3  4 5 6 7  8     10 11     15
|---|--|--|--|--------|--------|----------|
|///|          NW              |    8     |
|-----------------------------------------|
|                  HL                     |
|-----------------------------------------|
| A | C|//| H|   NC   |      CHAR. 1      |
|-----------------------------------------|
|                  .                      |
|       (VARIABLE #CHAR. SEE NC)          |
|                  .                      |
|-----------------------------------------|
|     CHAR. NC      |/////////////////////|
|-----------------------------------------|
| L |              SECL                   |
|-----------------------------------------|
|                 SSA                     |
|-|------|------|---------|---------|------|
|   P |      NP    |        CH            |
|-----------------------------------------|
|                  TN                     |
|-----------------------------------------|
|                PARM. 1                  |
|-----------------------------------------|
|                  .                      |
|                  .                      |
|                  .                      |
|-----------------------------------------|
|                PARM. NP                 |
 -----------------------------------------
```

NW - NUMBER OF WORDS IN ENTRY BLOCK

HL - HASH LINK - POINTS TO NEXT ENTRY WITH SAME HASH CODE

A - ACTIVITY BIT. 0 IF ACTIVE, 1 IF INACTIVE ENTRY

C - CALLABILITY BIT SET IF ENTRY POINT IS UNCALLABLE

H - HIDDEN ENTRY POINT. SET IF ENTRY POINT WILL NOT BE IN LIBRARY DIRECTORY

NC - NUMBER OF CHARACTERS IN NAME. MAX IS 16

ENTRY TYPE 8 (CONT.)

CHAR 1 - FIRST CHARACTER IN VARIABLE LIST

CHAR NC - LAST CHARTACTER IN VARIABLE
LIST

L - LAST ENTRY IN LIST
0 IF NOT LAST
1 IF LAST

SECL - SECONDARY ENTRY POINT LIST LINK

SSA - UNIT STARTING PB' ADDRESS

P - PARM CHECKER
- 00 NO CHECKING (IMPLIES NP UNDEFINED, TN AND PARMS ABSENT)
- 01 CHECK PROCEDURE TYPE (IMPLIES NP IS UNDEFINED AND PARMS ABSENT)
- 10 CHECK PROCEDURE TYPE AND NUMBER OF PARMS. (IMPLIES PARMS ABSENT)
- 11 CHECK PROCEDURE TYPE, NUMBER OF PARMS AND TYPE OF PARM.

NP - NUMBER OF PARMS

CN - CHARACTER COUNT OF PARMS

TN - PROCEDURE TYPE

ENTRY HEADER FORMAT

```
              --------------
SAH -------->|    HEADER    |
             |--------------|
             |       .      |
             |       .      |
             |       .      |
             |--------------|
             |    HEADER    |
              --------------
              --------------
SAH -------->|    HEADER    |
             |--------------|
             |       .      |
             |       .      |
             |       .      |
             |--------------|
SAC -------->|     CODE     |
             |--------------|
             |       .      |
             |              |
             |       .      |
             |--------------|
             |    HEADER    |
              --------------


              --------------
SAH -------->|    HEADER    |
             |--------------|
             |       .      |
             |       .      |
             |       .      |
             |--------------|
             |    HEADER    |
              --------------
```

EACH ENTRY (EXCEPT SECONDARY ENTRY POINT ENTRIES) MAY DESCRIBE N> 0 SETS OF HEADERS. THE HEADERS IN EACH SET MUST BE CONTINUOUS AND IN THE SAME ORDER AS THE HOW LIST DESCRIBING THE SET.

THE CODE MODULE MAY BE PLACED IN ANY POSITION IN A HEADER SET. NOTE THAT IF THE CODE MODULE IS AT THE BEGINNING OF A SET, SAC = SAH.

IF THE ENTRY HAS NO HEADER SET, THEN NH, SAH SEQUENCE IS ABSENT.

HEADER TYPE 0

GARBAGE

```
  0  1                          10 11     15
|---|---------------------------|---------|
|///|            NW             |    0    |
|-----------------------------------------|
|                                         |
|                                         |
|               GARBAGE                   |
|                                         |
|                                         |
-------------------------------------------
```

PCALs

HEADER TYPE 1

```
 0  1  2  3  4 5 6 7 8     10 11      15
|--|--|--|--|---------|----------|---------|
|//|                            |    1    |
|-----------------------------------------|
|                  PBA                    |
|-----------------------------------------|
|///////////|   NC     |    CHAR. 1       |
|-----------------------------------------|
|                   .                     |
|                   .                     |
|                   .                     |
|-----------------------------------------|
|      CHAR. NC     |////////////////////|
|-----------------------------------------|
|  P  |     NP     |          CN          |
|-----------------------------------------|
|                   TN                    |
|-----------------------------------------|
|                PARM. 1                  |
|-----------------------------------------|
|                   .                     |
|                   .                     |
|                   .                     |
|-----------------------------------------|
|                PARM. NP                 |
-------------------------------------------
```

PBA - PB' ADDRESS OF LINKED LIST OF PCAL INSTRUCTIONS TO BE REPAIRED - LOWER 14 BITS USED AS NEGATIVE DISP. - BIT 0 SET MEANS THAT WORD IS NOT A PCAL INSTRUCTION BUT A POINTER TO A SST LABEL OF ''EXTERNAL'' FORMAT - A LINK OF 0 TERMINATES THE LIST - BIT 1 SET MEANS THAT THE WORD IS TO BE INITIALIZED WITH THE PB ADDRESS OF THE PROCEDURE.

HEADER TYPE 2

PB ADDRESSES

```
  0  1                  10 11       15
|--|--------------------|----------|
|//|        NW          |     2    |
|----------------------------------|
|                PBA               |
|----------------------------------|
|                 .                |
|                 .                |
|                 .                |
|----------------------------------|
|                PBA               |
------------------------------------
```

PBA - PB' ADDRESS OF PB ADDRESS
      TO BE CORRECTED

HEADER TYPE 3

OWN/DATA VARIABLES

```
  0  1                  10 11       15
|--|--------------------|----------|
|//|        W           |    3     |
|----------------------------------|
| B|               PBA             |
|----------------------------------|
|                  .               |
|                  .               |
|                  .               |
|----------------------------------|
| B|               PBA             |
------------------------------------
```

PBA - PB' ADDRESS OF OWN VARIABLE
      POINTER TO BE CORRECTED

HEADER TYPE 4

DSDB/OWN/DATA/VALUES

```
  0  1             10 11        15
|---|----------------|-----------|
|///|      NW        |     4     |
|--------------------------------|
|              LD                |
|--------------------------------|
| B |            IN              |
|--------------------------------|
|                                |
|         INITIAL VALUES         |
|                                |
----------------------------------
```

LD - LOGICAL WORD DISPLACEMENT
     IN OWN ARRAY FOR INITIAL VALUES
B - BYTE BIT-SET IMPLIES THAT LD IS
     TYPE BYTE AND THAT THE FIRST
     WORD OF THE INITIAL VALUE BLOCK
     IS A COUNT OF THE NUMBER OF BYTES
     IN THE INITIAL VALUE BLOCK
IN - INTERATION NUMBER - NUMBER OF
     TIMES THE BLOCK OF INITIAL VALUE
     IS TO APPEAR IN THE SECONDARY BD -
     1->NO DUPLICATION,
     2->DUPLICATION, ETC

HEADER TYPE 5

PUST

```
 0  1               10 11      15
|--|-----------------|---------|
|//|       NW        |    5    |
|-------------------------------|
|              PBA              |
|-------------------------------|
|                               |
|        INITIAL VALUES         |
|                               |
---------------------------------
```

PBA - PB' ADDRESS OF LINKED LIST OF
      POINTERS TO BE INITIALIZED WITH
      DB ADDRESS OF PUST (SAME LIST
      FORMAT AS FOR FORMAT STRINGS)
      A PBA of -1 INDICATES NO FIX-UPS.

NOTE: ALL REFERENCES TO THE PUST INCLUDE THE FOUR-WORD HEADER THAT IS APPENDED BY THE SEGMENTER. THESE WORDS ARE NOT PRESENT IN THE HEADER; THEY ARE AUTOMATICALLY ALLOCATED AND INITIALIZED BY THE SEGMENTER.

HEADER TYPE 6

GLOBAL VARIABLES

```
 0 1          7 8     10 11    15
|--|------------|-------|-------|
|//|       NW          |   6   |
|-------------------------------|
|              TN               |
|-------------------------------|
|     DBA      |/////////|  NC |
|-------------------------------|
|   CHAR.1     |    CHAR. 2     |
|-------------------------------|
|               .               |
|               .               |
|               .               |
|-------------------------------|
|   CHAR. NC   |///////////////|
---------------------------------
```

HEADER TYPE 7

EXTERNAL VARIABLES

```
 0 1 2 3 4 5 6 7 8     10 11    15
|--|-|-|-|-|-|-|-|---------|-------|
|//|           NW          |   7   |
|----------------------------------|
|               TN                 |
|----------------------------------|
| M|/////|  NC    |   CHAR. 1      |
|----------------------------------|
|                .                 |
|                .                 |
|                .                 |
|----------------------------------|
~                                  ~
```

PBA-PB' address of linked lists of instructions to be repaired;lower 8 bits of inst. used as neg. displacement to next instruction;a link of 0 terminates the list.

M -Monitored variable bit;set

itored by debug.

DA -Logical word disp. in PUST;

EXTERNAL VARIABLES (Continued)

```
~                                   ~
|    CHAR. NC    |/////////////////|     lower 8 bits of word will be
|----------------------------------|     init. with prim.DB address
|                DA                |     of variable;DA is present
|----------------------------------|     if M=1.
|                PBA               |
|----------------------------------|  NOTE:PBA of -1 implies null list
|                 .                |
|                 .                |
|                 .                |
|----------------------------------|
|                PBA               |
------------------------------------
```

HEADER TYPE 8

PRIMARY DB

```
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
|-|-|-|-|-|-|-|-|-|-|--|--|--|--|--|--|
|/|         NW         |      8      |
|------------------------------------|
|U | U  | U  | U  | U  | U  | U  | U  |
| 0| 1  |  2 |  3 |  4 |  5 |  6 |  7 |
|------------------------------------|
|                 .                  |
|                 .                  |
|                 .                  |
|------------------------------------|
|U   |U   |U   |U   |U   |////////////|
| N-5| N-4| N-3| N-2| N-1|////////////|
|------------------------------------|
|                                    |
|                                    |
|           INITIAL VALUES           |
|                                    |
|                                    |
--------------------------------------
```

U - ADDRESS BITS
00 IF NO ADDRESS
01 IF NO ADDRESS
10 IF WORD ADDRESS IN SECONDARY DB
11 IF BYTE ADDRESS IN SECONDARY DB

N - NWPDB

NOTE: INITIAL ADDRESSES THAT ARE SECONDARY DB ADDRESSES ARE 0 RELATIVE (I.E., THEY ARE LOGICAL DISPLACEMENTS IN SECONDARY DB).

HEADER TYPE 9

COMMON VARIABLES

```
 0 1 2 3 4 5 6 7 8     10 11   15
|--|-|-|-|-|-|-|-|-|--------|-------|
|//|          NW            |   9   |
|-----------------------------------|
|               NWC                 |
|-----------------------------------|
|////////|  NC    |    CHAR. 1      |
|-----------------------------------|
|                 .                 |
|                 .                 |
|                 .                 |
|-----------------------------------|
|    CHAR. NC     |////////////////|
|-----------------------------------|
| B| M|           NL                |
|-----------------------------------|
|                 LD                |
|-----------------------------------|
|                 DA                |
|-----------------------------------|  -----
|                 PBA               |    |
|-----------------------------------|    |
|                 .                 |    |
|                 .                 |   NL
|                 .                 |    |
|-----------------------------------|    |
|                 PBA               |    |
|-----------------------------------|  -----
|                                   |
|                 .                 |
|                 .                 |
|                 .                 |
|                                   |
|-----------------------------------|
| B| M|           NL                |
|-----------------------------------|
|                 LD                |
|-----------------------------------|
|                 DA                |
|-----------------------------------|
|                 PBA               |
|-----------------------------------|
|                 .                 |
|                 .                 |
|                 .                 |
|-----------------------------------|
|                 PBA               |
 -----------------------------------
```

HEADER TYPE 9 (CONT.)

NWC - NUMBER OF WORDS IN COMMON ARRAY

NC - NUMBER OF CHARACTERS IN COMMON NAME- IF BLANK COMMON 4 COM'

DA - LOGICAL WORD DISP. IN PUST - LOWER 8 BITS OF WORD WILL BE INIT. WITH PRIM. DB ADDRESS OF VARIABLE - NOTE DA IS PRESENT IF M = 1

B - BYTE BIT
0 IF THE PRIMARY DB POINTER TO BE ALLOCATED AND INITIALIZED AND LD ARE OF TYPE WORD
1 IF TYPE BYTE

M - MONITORED VARIABLE BIT - SET IF VARIABLE IS BEING MONITORED BY DEBUG

NL - NUMBER OF ADDRESS LISTS FOR VARIABLE

LD - LOGICAL DISPLACEMENT OF VARIABLE IN COMMON ARRAY

PBA - PB' ADDRESS OF LINKED LISTS OF INSTRUCTIONS TO BE REPAIRED
LOWER 8 BITS USED AS NEGATIVE DISPLACEMENT TO NEXT INSTRUCTION
A LINK OF 0 TERMINATES THE LIST

PBA = -1 INDICATES NO FIX-UPS

HEADER TYPE 10

LOGICAL UNITS

```
  0                    10 11   15
|--|---------------------|-------|
|//|          8          |  10   |
|--------------------------------|
|                                |
|                                |
|                                |
|             BIT MAP            |
|                                |
|                                |
|                                |
----------------------------------
```

```
BIT MAP - BIT MAP OF LOGICAL UNITS
          REFERENCED; BIT 0
          CORRESPONDS TO LU 0, ETC.
          (1 LESS THAN OR EQUAL TO LU
 LESS THAN OR EQUAL TO 99)
```

HEADER TYPE 11

FORMAT STRING

```
  0                    10 11   15
|--|---------------------|-------|
|//|         NW          |   11  |
|--------------------------------|
|               PBA              |
|--------------------------------|
|               NC               |
|--------------------------------|
|    CHAR. 1     |    CHAR. 2    |
|--------------------------------|
|                .               |
|                .               |
|                .               |
|--------------------------------|
|    CHAR. NC    |///////////////|
----------------------------------
```

```
PBA - PB' ADDRESS OF LINKED LIST OF
      POINTERS TO BE INITIALIZED
      LOWER 14 BITS OF WORD USED
      AS NEGATIVE DISPLACEMENT TO
      NEXT POINTER - BIT 0 SET
      MEANS THAT THE POINTER IS TO
      BE TYPE BYTE - A LINK OF 0
      TERMINATES THE LIST.
```

RL FILE FORMAT

```
   ------------                                    ------------
  0|   LID     |0  LOADER ID                      0|          |
   |-----------|                                   |          |
  1|   FL      |1  FILE LENGTH (IN RECORDS)        |  RECORD  |
   |-----------|                                   |    0     |
  2|   NS      |2  NR. SECTIONS                    |          |
   |-----------|                                   ------------
  3|           |3
   |-----------|
  4|           |4
   |   SAXL    |   S.A. EXTERNAL SET LIST
  5|           |5
   |-----------|                                   ------------
  6|           |6                                 1|          |
   |           |                                   |          |
  7|           |7                                  | FREE MAP |
   |           |                                   |    0     |
 10|           |8                                  |          |
   |           |                                   ------------
 11|           |9
   |           |
 12|           |10
   |           |
   |           |                                   ------------
   |           |                                NS |          |
   |           |                                   |          |
   |           |                                   | FREE MAP |
   |           |                                   |  NS-1    |
   |           |                                   |          |
   |           |                                   ------------
   |           |   NOTE: UNINITIALIZED FIELDS ARE
   |           |         RESERVED FOR FUTURE USE AND
   |           |         SHOULD BE ZERO.
   |           |                                   ------------
   |           |                               NS+1|          |
   |           |                                   |          |
   |           |                                   |          |
   |           |                                   |          |
   |           |                                   |AVAILABLE |
   |           |                                   |          |
   |-----------|                                   |          |
 41|   HL      |33 S.A. HASH LIST 0                |          |
   |     0     |                                   |          |
   |-----------|                                   |          |
   |    .      |                                   |          |
   |    .      |                                   |          |
   |    .      |                                   |          |
   |-----------|                                   |          |
177|   HL      |127 S.A. HASH LIST 94              |          |
   |     94    |                                   |          |
   ------------                                    ------------
```

STORAGE MANAGEMENT

FILE SPACE IS MANAGED IN TERMS OF 32 WORDS BLOCKS (4 BLOCKS PER 128 WORD RECORD).

FREE SPACE (BLOCKS) IS ACCOUNTED FOR IN A BIT MAP, WHICH IS PARTITIONED INTO RECORDS (2K BLOCKS PER SECTION). A 0 INDICATES THAT A BLOCK IS USED, A 1 INDICATES THAT IT IS FREE.

FILE SPACE IS ALSO PARTITIONED INTO 512 RECORD SECTIONS (64 MAX. SECTIONS, 2K BLOCKS PER SECTION, 1 MAP PER SECTION). THE NUMBER OF SECTIONS IN A FILE IS NS=(FL+511) & LSR(9). THE FIRST NS RECORDS FOLLOWING RECORD 0 (RECORDS 1 TO NS) ARE RESERVED FOR THE SECTION MAPS.

A COMPLETE FILE ADDRESS WOULD HAVE THE FOLLOWING CONFIGURATION:

```
 0 1 2 3 4 5 6                15 16                26 27        31
|------------|--------------------|--------------------|-----------|
|            |      SECTION       |       BLOCK        | DISPLCMT  |
------------------------------------------------------------------
```

FILE (WORD) ADDRESS
DOUBLE WORD

ENTRY POINT DIRECTORY

```
----------       ----------          ----------          ----------
|   HL   |------>|  LINK  |-->...-->|  LINK  |-->...-->|    0
----------       |--------|          |--------|          |--------|
                 |  USED  |          |  USED  |          |  USED  |
                 |--------|          |--------|          |--------|
                 |        |          |        |          |        |
                 |        |          |        |          |        |
                 |        |          |        |          |        |
                 |        |          |        |          |        |
                 |        |          |        |          |        |
                 |--------|          |--------|          |--------|
                 |////////|          |////////|          |////////|
                 |////////|          |////////|          |////////|
                 ----------          ----------          ----------
```

THE DIRECTORY IS PARTITIONED INTO 95 HASH LISTS (SAME HASH FUNCTION AS USL); EACH HASH LIST IS A LINKED LIST OF RECORDS.

EACH RECORD CONTAINS A SUCCESSOR LINK (RECORD #) AND A USED SPACE COUNT. A LINK OF 0 TERMINATES A LIST. WHEN A RECORD IS VOID OF ENTRIES (USED=2), ITS SPACE IS RETURNED TO THE FREE STORAGE AREA.

TYPICAL DIRECTORY ENTRY

```
   0   1   2   3   4567  8                    15

|---|---|---|---|------|----------------------|
| S | U | I |///|  NC  |        CHAR. 1       |
|---------------------------------------------|
|                      .                      |
|                      .                      |
|                      .                      |
|---------------------------------------------|
|       CHAR. NC       |//////////////////////|
|---------------------------------------------|
|                                             |
|                S.A. INFO BLOCK              |
|                                             |
|---------------------------------------------|
|                  S.A. ENTRY                 |
|---------------------------------------------|
| F | W |           NW CODE                   |
|---------------------------------------------|
|  LC   |      NP      |          CN          |
|---------------------------------------------|
|                      TN                     |
|---------------------------------------------|
|                   PARM. 1.                  |
|---------------------------------------------|
|                      .                      |
|                      .                      |
|                      .                      |
|---------------------------------------------|
|                   PARM. NP                  |
-----------------------------------------------
```

S - SECONDARY ENTRY POINT BIT - SET IF
THE ENTRY POINT WAS ORIGINALLY A
SECONDARY ENTRY POINT.

U - UNCALLABLE BIT - SET IF ENTRY POINT
IS UNCALLABLE.

I - PRIVILEGED MODE BIT - SET IF CODE
MODULE IS TO BE RUN IN PRIV. MODE.

LC is (0:2)...Level of Checking
    0 = No checking
    1 >= Check for procedure type
    2 >= Check for # parameters
    3 >= Check for parameter type
NP is (2:6) is # parameters

PROCEDURE INFO BLOCK

```
 0                             15
|--------------------------------| ---------------
|            NW INFO             |
|--------------------------------|              |
|            NW CODE             |              |
|--------------------------------|              |
|        # ENTRY POINTS          |              |
|--------------------------------| -------      |
|                                |              |
|                                |    |         |
|          CODE MODULE           |   NWC        |
|                                |    |         |
|                                |              |
|--------------------------------| -------      |
|                                |              |
|           EXTN LINK            |              |
|                                |              |
|--------------------------------|              |
|              TPDB              |              |
|--------------------------------|              |
|              TSDB              |              |
|--------------------------------|              |
|             NWSDB              |            NWI
|--------------------------------|              |
|             HEADER             |              |
|--------------------------------|              |
|                                |              |
|             HEADER             |              |
|                                |              |
|--------------------------------|              |
|                                |              |
|                                |              |
|               .                |              |
|                                |              |
|               .                |              |
|                                |              |
|                                |              |
|--------------------------------|              |
|                                |              |
|             HEADER             |              |
|                                |              |
|--------------------------------|              |
|               -1               |
 --------------------------------  ---------------
```

ALL HEADERS FOR THE PROCEDURE ARE APPENDED TO THE INFO BLOCK. THE HEADER SETS (EXTERNAL LISTS) ARE LINKED BY INCREASING FILE ADDRESS; A LINK OF %17777777777D TERMINATES THE LIST.

HEADERS

```
 0   1   2   3   4567  8         10 11        15
|---|---|---|---|------|----------|-----------|
|///|          NW                 |     1     |
|---------------------------------------------|
| F | W |        NW CODE                      |
|---------------------------------------------|
|                                             |
|           S.A. INFO BLOCK                   |
|                                             |
|---------------------------------------------|
|              S.A. ENTRY                     |
|---------------------------------------------|
|                  PBA                        |
|---------------------------------------------|
| S | U | I |///| NC   |       CHAR. 1        |
|---------------------------------------------|
|                      .                      |
|                      .                      |
|                      .                      |
|---------------------------------------------|
|       CHAR. NC       |//////////////////////|
|---------------------------------------------|
|   P   |     NP       |         CN           |
|---------------------------------------------|
|                      TN                     |
|---------------------------------------------|
|                    PARM. 1                  |
|---------------------------------------------|
|                      .                      |
|                      .                      |
|                      .                      |
|---------------------------------------------|
|                   PARM. NP                  |
 ---------------------------------------------
```

F - SET IF FATAL ERROR
W - SET IF NON-FATAL ERROR
S - SATISFIED BIT - SET IF EXTERNAL IS
    SATISFIED WITHIN RL.
U - UNCALLABLE BIT
I - PRIVELEGED BIT

ALL HEADERS ARE THE SAME AS IN A USL EXCEPT FOR THE PCAL HEADER.

CHAPTER 10 PREPARED OBJECT CODE

PROGRAM FILE FORMAT

0	FLAGS	0	
1	NS	1	NUMBER OF CODE SEGMENTS
2	GS	2	GLOBAL SIZE (DB TO QI) IN WORDS
3	SAG	3	GLOBAL AREA RECORD #
4	SAS		SEGMENT SET RECORD # (EACH SEG. STARTS IN NEW RECORD)
5	ISS	5	INITIAL STACK SIZE IN WORDS
6	IDLS	6	INITIAL DL SIZE IN WORDS
7	MAXD	7	MAX. DATA SEGMENT SIZE (DL TO Z) IN WORDS
10	SAE	8	ENTRY POINT LIST RECORD #
11	SSEG	9	STARTING SEGMENT #
12	SADR	10	PRIN. ENTRY PT PB ADDRESS
13	SASTLT	11	DB ADR. OF STLT (-1 IF NO STLT) (STLT=Segment Length Table)
14	SAFLUT	12	DB ADR. OF FLUT (-1 IF NO FLUT)
15	SAX	13	EXTERNAL LIST RECORD #
16	SSTT	14	PRIN. ENTRY PT SST #
17	SATC	15	STARTING ADDRESS OF TRAPCOM'
20	SAPMAP	16	STARTING RECORD OF PMAP INFO
21	SASI	17	STARTING RECORD OF SYMBOLIC ITEMS
22	FLAGS2	19	
23	CKSUM	19	TOTAL CHECKSUM OF ALL SEGMENTS
24		20	NOTE : ALL UNUSED WORD ARE RESERVED FOR FUTURE USE AND SHOULD BE SET TO ZERO.
25		21	
26		22	

PROGRAM FILE FORMAT (CONT.)

```
     |             |
     ~             ~
     ~             ~
     |-------------|
   27|             |23
     |-------------|
   30|             |24
     |-------------|
   31|             |25
     |-------------|
   32|             |26
     |-------------|
   33|             |27
     |-------------|
     | CST  | CST  |
   34|    o |    1 |28 \
     |-------------|    |
     |             |    |
     |-------------|    >   CST REMAPPIMG ARRAY
     | CST  |//////|    |
     |    n |//////|    |
     |-------------|    /
     |P|S|  SL     |    \
     | | |     o   |K   |
     |-------------|    |
     |      .      |    |
     |      .      |    >   SEGMENT DESCRIPTER ARRAY
     |      .      |    |
     |-------------|    |
     |P|S|  SL     |    |
     | | |     n   |    |
     |-------------|    /
     |             |
     |             |
     |             |
     |             |
     |             |
     |             |
     |             |
     |             |
     |             |
     |             |
     |             |
     |             |
     |             |
    L|             |  L
     ---------------
```

P-PRIVILEGED MODE
S-Segment STT format: 0=> old format, 1=> new (extended) format
N=NS -1
K=28 + (NS +1) & LSR (1)
L=((28 + NS + (NS + 1)&LSR(1) + 127)/128)128 - 1

FLAGS

```
 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
|-|--|--|--|--|--|--|--|--|--|---|---|---|---|---|---|
|F|W |Z |P |//|//|  |BA|IA|PM|   |   | MR|///| DS| PH
------------------------------------------------------
```

```
        F - FATAL ERROR IN PROGRAM
        W - NON-FATAL ERROR IN PROGRAM
        Z - ZERO UNIT DL AREA
        P - SET IF ANY SEG IS PRIV. (IF NOT SET NORMAL=
            NONPRIV MODE)
```

CAPABILITIES

```
              /                  BATCH ACCESS (9)    [BA]
              |
              |            INTERACTIVE ACCESS (8)    [IA]
              |
              |               PRIVILEGED MODE (7)   [PM]
              |
ACCESS TO     |
GENERAL      <
RESOURCES     |
              |
              |                  MULTIPLE RINS (4)   [MR]
              |
              |
              |             EXTRA DATA SEGMENT (2)   [DS]
              |
              \               PROCESS HANDLING (1)   [PH]
```

FLAGS2

```
0 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
|-|--|--|--|--|--|--|--|--|--|---|---|---|---|---|---|
|T|K |              RESERVED                        |
------------------------------------------------------
```

T - PATCH AREA EXISTED IN ALL CODE SEGMENTS
K - CHECKSUM VALID

CST REMAPPING ARRAY

CONTAINS THE LAST CST NUMBERS ASSIGNED TO THE SEGMENTS; INDEXED BY SEGMENT NUMBER. WHEN A PROGRAM FILE IS PREPARED, THE ARRAY IS INITIALIZED TO 0, 1...,N. THIS ARRAY IS USED TO RE-ESTABLISH INTRA-PROGRAM LINKAGE WHEN THE PROGRAM IS LOADED.

SEGMENT DESCRIPTER ARRAY

CONTAINS THE SEGMENT LENGTH AND A FLAG INDICATING IF THE SEGMENT IS TO BE LOADED IN PRIV. MODE. INDEXED BY SEGMENT NUMBER. ALL SEGMENTS BEGIN ON A RECORD BOUNDARY. THE NUMBER OF RECORDS FOR A GIVEN SEGMENT IS (SL + 127) & LSR(7). THE RECORD NUMBER, SAS, OF SEGMENT N IS

```
SAS:=0
FOR I=0 TO N-1
 BEGIN
 SAS:=SAS + (SL(I) + 127)&LSR(7)
 END
```

GLOBAL AREA FORMAT

A SET OF RECORDS CONTAINING THE INITIAL VALUES FOR THE GLOBAL AREA OF THE DATA SEGMENT. THIS SET BEGINS AT RECORD SAG (WORD 3) AND CONSISTS OF (GS + 127) & LSR(7) RECORDS.

EXTERNAL LIST

```
0      7 8      15
|---|---|--------|                  ---      ---      ---
|///|NC | CHAR 1 | TYPICAL ENTRY      |        |        |
|----------------|                    |        |        |
|       .        |                    |        |        |
|       .        |                CHECK 0      |        |
|       .        |                    |        |        |
|----------------|                    |        |        |
|CHAR NC|////////|                    |   CHECK 1&2     |
|----------------|                    |        |        |
|     NR         |                    |        |        |
|----------------| \                  |        |        |
| STT # | SEG #  | |                  |        |        |
|----------------| |                  |        |        |
|       .        | |                  |        |        |
|       .        | >  NP              |        |        |
|       .        | |                  |        |   CHECK 3
|----------------| |                  |        |        |
| STT # | SEG #  | |                  |        |        |
|-|--------------| /                  |        |        |
|LC| NP |  CN    |                    |        |        |
|-|--------------|                  ---        |        |
|      TN        |                             |        |
|----------------|                           ---       |
|    PARM 1      |                                      |
|----------------|                                      |
|       .        |                                      |
|       .        |                                      |
|       .        |                                      |
|----------------|                                      |
|                |                                      |
|----------------|                                      |
|    PARM NP     |                                      |
------------------                                    ---
        .
        .
        .
------------------
|       0        |    LIST TERMINATER
------------------


LC (0:2) = LEVEL OF CHECKING
           0 = NO CHECKING
           1 >= CHECK FOR PROCEDURE TYPE
           2 >= CHECK FOR # PARAMETERS
           3 >= CHECK FOR PARAMETER TYPE
NP (2:6)   IS # PARAMETERS
```

ENTRY POINT LIST

```
 ---------------------
 |////| NC | CHAR 1  |
 |-------------------|
 |         .         |
 |         .         |
 |         .         |
 |-------------------|
 | CHAR NC |/////////|
 |-------------------|
 |     P.B. ADR      |
 |-------------------|
 |       STT #       |
 ---------------------
           .
           .
           .
 ---------------------
 |////| NC | CHAR 1  |
 |-------------------|
 |         .         |
 |         .         |
 |         .         |
 |-------------------|
 | CHAR NC |/////////|
 |-------------------|
 |     P.B. ADR      |
 |-------------------|
 |       STT #       |
 |-------------------|
 |         0         |  LIST TERMINATER
 ---------------------
```

NOTE THAT THE ENTRY POINT LIST MUST IMMEDIATELY FOLLOW THE EXTERNAL LIST.

THE CODE SEGMENT WITH PATCH AREA

```
 --------
 |      |
 | CODE |
 |      |
 |------|
 |      |
 | PATCH|
 | AREA |
 |      |
 |------|
 |      |
 | STT  |
 |      |
 --------
```

PATCH AREA

```
------------
| PROGRAM  |     4-WORD PROGRAM NAME
| NAME     |
|----------|
| SEGMENT  |     8-WORD SEGMENT NAME
| NAME     |
|----------|
|   //     |     1-WORD UNUSED
|----------|
| CHECKSUM |     1-WORD CHECKSUM
|----------|
| PREP TIME|     2-WORD PREP TIME
|----------|
|PATCH TIME|     2-WORD PATCH TIME
|----------|
|  PATCH   |
|  AREA    |
|----------|
| PALEN    |     1-WORD PATCH AREA LENGTH
|----------|
|          |
|  STT     |
|          |
```

PMAP INFO

```
 ------------------
|                  |
|      PTT         |    PMAP TYPE TABLE
|                  |
|------------------|
|                  |
|      SPP         |    SEGMENT PMAP POINTERS
|                  |
|------------------|
|                  |
|      APD         |    ACTUAL PMAP DATUM
|                  |
 ------------------
```

PMAP TYPE TABLE

```
 ------------------
|     PTTL         |    TYPE TABLE LENGTH
|------------------|
|     LPR0         |    LENGTH OF PMAP RECORD TYPE 0
|------------------|
|     LPR1         |    LENGTH OF PMAP RECORD TYPE 1
|------------------|
|       :          |
|       :          |
|------------------|
|     LPRn         |    LENGTH OF PMAP RECORD TYPE n
 ------------------
```

NOTE : n = PTTL - 2

PMAP RECORDS

TYPE 0   SEGMENT PMAP RECORD

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
--------------------------------
|        0|  NC   |   char 1    |
|-------------------------------|
|              .                |
|              .                |
|              .                |
|-------------------------------|
|  char NC      |///////////////|
|-------------------------------|
|   STT LEN     |   SEG NUM     |
|-------------------------------|
|       SEG LENGTH              |
--------------------------------
```

TYPE 1   PROCEDURE PMAP RECORD

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
--------------------------------
|        1|  NC   |   char 1    |
|-------------------------------|
|              .                |
|              .                |
|              .                |
|-------------------------------|
|  char NC      |///////////////|
|-------------------------------|
|H|/////////////////////////////|
|-------------------------------|
|       SA OF CODE              |
|-------------------------------|
|        CODE LENGTH            |
|-------------------------------|
|    PRIMARY ENTRY POINT ADDR   |
|-------------------------------|
|     COBOL TOOL BOX ID         |
|          LINK                 |
|-------------------------------|
|   TOOL BOX PROCEDURE ID       |
--------------------------------
```

```
TYPE 2   SECONDARY ENTRY PMAP RECORD
------   ---------------------------

          0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
         ---------------------------------
         |        2|  NC   |   char 1    |
         |-------------------------------|
         |               .               |
         |               .               |
         |               .               |
         |-------------------------------|
         |  char NC      |///////////////|
         |-------------------------------|
         |H|/////////////////////////////|
         |-------------------------------|
         |  SECONDARY ENTRY POINT ADDR   |
         |-------------------------------|
         |     NUMBER OF ENTRY POINTS    |
         ---------------------------------

         H : HIDDEN ENTRY FLAG
```

SL FILE FORMAT

```
   --------
  0| LID  |0
   |------|
  1|  FL  |1 FILE LENGTH (IN RECORDS)
   |------|
  2|  EL  |2 EXTENT LENGTH (IN RECORDS)
   |------|
  3|      |3
   |------|
  4| NSEG |4 # SEGMENTS
   |------|
  5|      |5
   |------|
  6|      |6
   |------|
 7 | FRTL |7 S.A. OF FREE R.T. ENTRY LIST (-1 IF NONE)
   |------|
 10|      |8
   |------|
 11| NRT  |9 # REFERENCE TABLE ENTRIES
   |------|
 12|      |10
   |------|
 13|  NS  |11 # SECTIONS
   |------|
 14|      |12
   |      |
   |      |
   |      |
   |      |
   |      |
   |      |
   |      |
   |      |
   |      |
   |      |
   |      |
   |      |
   |      |
   |      |
   |------|
 41| HL0  |33
   |------|
   |  .   |
   |  .   |
   |  .   |    NOTE:
   |------|    SHADED AND UNITIALIZED FIELDS ARE
177|HL94  |127 RESERVED FOR FUTURE USE AND
   --------    SHOULD BE ZERO. HL = HASH LIST.
```

SL FILE FORMAT (CONT.)

```
   0------------
    |          |
    |  RECORD  |
    |    0     |
    |          |
    ------------


   1------------
    |          |
    |  RECORD  |  <--- REFERENCE TABLE POINTERS
    |    1     |
    |          |
    ------------


   2------------
    |          |
    | FREE MAP |
    |    0     |
    |          |
    ------------
         .
         .
         .
NS+1------------
    |          |
    | FREE MAP |
    |   NS-1   |
    |          |
    ------------


NS+2-------------
    |          |
    |          |
    |AVAILABLE |
    |          |
    |          |
    |          |
    |          |
    |          |
    |          |
    |          |
    |          |
    ------------
```

STORAGE MANAGEMENT

FILE SPACE IS MANAGED IN TERMS OF 128 WORD BLOCKS (1 BLOCK PER 128 WORD RECORD).

FREE SPACE (BLOCKS) IS ACCOUNTED FOR IN A BIT MAP, WHICH IS PARTITIONED INTO RECORDS (2K BLOCKS PER SECTION). A 0 INDICATES THAT A BLOCK IS USED; A 1 INDICATES THAT IT IS FREE.

FILE SPACE IS ALSO PARTITIONED INTO 2048 RECORD SECTIONS (16 MAX. SECTIONS, 2K BLOCKS PER SECTION 1 MAP PER SECTION). THE NUMBER OF SECTIONS IN A FILE IS NS=(FL + 2047) & LSR(7). THE FIRST NS RECORDS FOLLOWING RECORDS 0, 1 (RECORDS 2 TO NS+1) ARE RESERVED FOR THE SECTION MAPS.

IF THE SECTION MAPS SPECIFY MORE SPACE THAN IS POTENTIALLY AVAILABLE, THOSE RECORDS BEYOND FLIMIT ARE MARKED AS "USED".

ENTRY POINT DIRECTORY

```
----------       -----------       -----------       -----------
|  HL    |------>|  LINK   |->---->|  LINK   |->---->|    0    |
----------       |---------|       |---------|       |---------|
                 |  USED   |       |  USED   |       |  USED   |
                 |---------|       |---------|       |---------|
                 |         |       |         |       |         |
                 |         |       |         |       |         |
                 |---------|       |---------|       |---------|
                 |/////////|       |/////////|       |/////////|
                 |/////////|       |/////////|       |/////////|
                 -----------       -----------       -----------
```

THE DIRECTORY IS PARTITIONED INTO 95 HASH LISTS (SAME HASH FUNCTION AS USL); EACH HASH LIST IS A LINKED LIST OF RECORDS.

EACH RECORD CONTAINS A SUCCESSOR LINK (RECORD #) AND A USED SPACE COUNT. A LINK OF 0 TERMINATES A LIST. WHEN A RECORD IS VOID OF ENTRIES (USED=2), ITS SPACE IS RETURNED TO THE FREE STORAGE AREA.

THE HASH LIST HEAD POINTERS (HL IN THE DIAGRAM ABOVE) ARE IN RECORD 0 WORDS %41 TO %177.

TYPICAL DIRECTORY ENTRY

```
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
-----------------------------------------------------------------
|///| U |///| P |       NC       |            CHAR 1            |
|---------------------------------------------------------------|
|                                .                              |
|                                .                              |
|                                .                              |
|---------------------------------------------------------------|
|        CHAR NC                 |//////////////////////////////|
|---------------------------------------------------------------|
|        STT #                   |           SEG #              |
|---------------------------------------------------------------|
|  LC   |          NP            |            CN                |
|---------------------------------------------------------------|
|                               TN                              |
|---------------------------------------------------------------|
|                             PARM 1                            |
|---------------------------------------------------------------|
|                                .                              |
|                                .                              |
|                                .                              |
|---------------------------------------------------------------|
|                             PARM NP                           |
|---------------------------------------------------------------|
```

LC is (0:2)...Level of Checking
      0 = No checking
      1 >= Check for procedure type
      2 >= Check for # parameters
      3 >= Check for parameter type
NP is (2:6) is # parameters

P - 0= Not permanently allocated
    1= Permanently allocated

U - Uncallable bit - set if entry
    point is uncallable.

CODE SEGMENT LINKAGE STRUCTURE

```
-----------------
|               |
|               |
| CODE SEGMENT  |
|               |
|               |
|---------------|
|               |
| STT MAP ARRAY |
|               |
|---------------|
|               |
|               |
| EXTERNAL LIST |
|               |
|               |
-----------------
```

EACH CODE SEGMENT OCCUPIES AN INTEGRAL NUMBER OF RECORDS. THIS BLOCK OF INFORMATION CAN BE SUB-DIVIDED INTO THREE TABLES: THE CODE SEGMENT PROPER, AN STT SEGMENT MAP ARRAY, AND AN EXTERNAL LIST.

STT MAP ARRAY

A 1 BYTE X 256 BYTE ARRAY. IT IS INDEXED BY STT NUMBER AND RETURNS (IF THE STT CORRESPONDS TO AN EXTERNAL OF THE SEGMENT) THE SEGMENT NUMBER OF THE EXTERNAL AND 255 OTHERWISE. THIS ARRAY IS USED WHENEVER THE SEGMENT IS LOADED AND IS UPDATED WHENEVER THE SL IS BOUND BY THE SEGMENTER.

EXTERNAL LIST

A SYMBOLIC LIST OF THE EXTERNALS OF THE SEGMENT. EACH ENTRY CONTAINS INFORMATION ABOUT THE EXTERNAL: PARAMETER CHECKING LEVEL AND PARAMETER MATCHING INFORMATION, AND THE SEGMENT NUMBER AND STT NUMBER IF THE EXTERNAL IS SATISFIED WITHIN THE SL.

CODE SEGMENT LINKAGE STRUCTURE (CONT.)

```
 0 1 2 3 4567 8         15

|-|-|-|-|----|------------|
|                         |
|                         |
|                         |
|                         |
|                         |
|                         |
|                         |
|                         |
|      CODE SEGMENT       |
|                         |
|                         |
|                         |
|                         |
|                         |
|                         |
|-------------------------|
|                         |
|                         |
|      STT MAP ARRAY      |
|                         |
|-------------------------|
|S|/|/|/| NC |   CHAR. 1  | S - SATISFIED BIT - SET IF EXTERNAL
|-------------------------|     IS SATISFIED WITHIN SL
|            .            |
|            .            |
|            .            |
|-------------------------|
| CHAR. NC   |////////////|
|-------------------------|
|  STT #     |   SEG. #   |
|-------------------------|
| P |   NP   |     CN     |
|-------------------------|
|           TN            |
|-------------------------|
|         PARM. 1         |
|-------------------------|
|                         |
|                         |
|-------------------------|
|         PARM. NP        |
|-------------------------|
|            .            |
|-------------------------|
|            0            |  EXTERNAL LIST TERMINATOR
 -------------------------
```

REFERENCE TABLE STRUCTURE

FOR EACH SEGMENT THERE IS A REFERENCE TABLE ENTRY OF 32 WORDS. THE REFERENCE TABLE ENTRIES ARE PACKED FOUR TO A RECORD. THE RECORDS CONTAINING THE REFERENCE TABLE ENTRIES ARE LISTED IN RECORD 1. THE RECORD CONTAINING REFERENCE TABLE ENTRY N IS REC 1 (N.(0 : 14)); THE FIRST WORD OF THE ENTRY IS REFTAB (N.(14 : 2) & LSL (5)).

WHEN A SEGMENT IS DELETED, THE REFERENCE TABLE ENTRY CORRESPONDING TO THE SEGMENT IS RELEASED. THESE FREE ENTRIES ARE LINKED TOGETHER IN A LIST; THE SEGMENT # IS USED AS A LINK AND IS PLACED IN THE FIRST WORD OF THE ENTRY.

WHEN A SEGMENT IS ADDED IT IS ASSIGNED A SEGMENT NUMBER (0 LESS THAN/EQUAL TO N LESS THAN/EQUAL TO 254); THE NUMBER IS THAT OF THE FIRST FREE REFERENCE TABLE ENTRY, OR, IF NONE ARE FREE, THE NEXT AVAILABLE REFERENCE TABLE ENTRY (CAUSING SPACE ALLOCATION FOR THE ENTRY).

```
                        REFERENCE TABLE    (256 MAX. ENTRIES)
                        ---------------
                                    TYPICAL ENTRY

 DREC. 1      R.T. REC.      0 1 2 3 4 5 6 7 8 9          15 %
----------   ------------   |-|-|-|-|-|-|-|-|-|-|-----------|
|  RL    |-->|    E     |-->|P|N|      SEGMENT LENGTH       | 0
|   0    |   |    0     |   |-------------------------------|
|--------|   |----------|   |    SEGMENT ADDRESS (REC. #)   | 1
|        |   |    E     |   |-------------------------------|
|   .    |   |    1     |   | # REC'S FOR SEG. & EXTN. LIST | 2
|   .    |   |----------|   |-------------------------------|
|   .    |   |    E     |   |F|S|/|/|A|C|X|/|/| # ENTRY PTS.| 3
|        |   |    2     |   |-------------------------------|
|--------|   |----------|   |            SAPMAP             | 4
|  RL    |   |    E     |   |-------------------------------|
|   63   |   |    3     |   |           SASI                | 5
----------   ------------   |-------------------------------|
(FILE REC1)  (1 SECTOR)     |T|K|                           | 6
                            |-------------------------------|
SEG.NAME -16 BYTE ARRAY     |                               | 7
         WITH NO CHARAC-    |-------------------------------|
         TER COUNT AND      |                               | 10
         TRAILING BLANKS    |                               |
         ADDED.             |                               |
                            |          SEGMENT NAME         |
REF.MAP -256 BIT ARRAY      |                               |
        (INDEXED BY SEG#);  |                               |
        BIT SET IF SEG IS   |-------------------------------|
        REFERENCED DIRECT-  |                               | 20
        LY OR INDIRECTLY.   |                               |
                            |                               |
F   SEGMENT DELETED         |                               |
S   EXTERNAL SATISFIED      |                               |
A   PERMANENTLY ALLOCATED   |                               |
C   CORE RESIDENT SEGMENT   |                               |
X   MPE SEGMENT             |                               |
P   PRIV.INST. IN SEGMENT   |       REFERENCED SEGMENTS     |
N   SLSEGFLAG               |            BIT MAP            |
T   PATCH FLAG              |                               |
K   CHECKSUM FLAG           |                               |
                            |                               |
SLSEGFLAG:                  |                               |
      = 0 => SEG STT IS IN  |                               |
             OLD FORMAT     |                               |
      = 1 => SEG STT IS IN  |                               |
             NEW FORMAT --  |                               |
             EXTENDED CSTS  |                               |
                            |                               |
                            ---------------------------------
```

THE CODE SEGMENT WITH PATCH AREA

```
--------
|      |
| CODE |
|      |
|------|
|      |
| PATCH|
| AREA |
|      |
|------|
|      |
| STT  |
|      |
--------
```

PATCH AREA

```
------------
| SEGMENT  |    8-WORD SEGMENT NAME
| NAME     |
|----------|
|   //     |    1-WORD UNUSED
|----------|
| CHECKSUM |    1-WORD CHECKSUM
|----------|
| PREP TIME|    2-WORD PREP TIME
|----------|
|PATCH TIME|    2-WORD PATCH TIME
|----------|
|  PATCH   |
|  AREA    |
|----------|
| PALEN    |    1-WORD PATCH AREA LENGTH
|----------|
|          |
|  STT     |
|          |
```

PMAP INFO

```
-----------------
|               |
|      PTT      |    PMAP TYPE TABLE
|               |
|---------------|
|               |
|               |
|               |
|      APD      |    ACTUAL PMAP DATUM
|               |
-----------------
```

PMAP TYPE TABLE

```
-----------------
|     PTTL      |    TYPE TABLE LENGTH
|---------------|
|     LPR0      |    LENGTH OF PMAP RECORD TYPE 0
|---------------|
|     LPR1      |    LENGTH OF PMAP RECORD TYPE 1
|---------------|
|       :       |
|       :       |
|---------------|
|     LPRn      |    LENGTH OF PMAP RECORD TYPE n
-----------------
```

NOTE : n = PTTL - 2

PMAP RECORDS

TYPE 0 SEGMENT PMAP RECORD

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
---------------------------------
|        0|  NC   |   char 1    |
|-------------------------------|
|              .                |
|              .                |
|              .                |
|-------------------------------|
|  char NC      |///////////////|
|-------------------------------|
|   STT LEN     |   SEG NUM     |
|-------------------------------|
|       SEG LENGTH              |
---------------------------------
```

TYPE 1 PROCEDURE PMAP RECORD

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
---------------------------------
|        1|  NC   |   char 1    |
|-------------------------------|
|              .                |
|              .                |
|              .                |
|-------------------------------|
|  char NC      |///////////////|
|-------------------------------|
|H|/////////////////////////////|
|-------------------------------|
|       SA OF CODE              |
|-------------------------------|
|        CODE LENGTH            |
|-------------------------------|
|    PRIMARY ENTRY POINT ADDR   |
|-------------------------------|
|     COBOL TOOL BOX ID         |
|          LINK                 |
|-------------------------------|
|   TOOL BOX PROCEDURE ID       |
---------------------------------
```

TYPE 2   SECONDARY ENTRY PMAP RECORD

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
 ---------------------------------
 |        2|  NC    |   char 1    |
 |-------------------------------|
 |              .                |
 |              .                |
 |              .                |
 |-------------------------------|
 | char NC        |///////////////|
 |-------------------------------|
 |H|/////////////////////////////|
 |-------------------------------|
 | SECONDARY ENTRY POINT ADDR    |
 |-------------------------------|
 |    NUMBER OF ENTRY POINTS     |
 ---------------------------------
```

H : HIDDEN ENTRY FLAG

CHAPTER 11 LOADER

GENERAL INFORMATION

The first area of the CST, pointed to by absolute 0, contains system and library segments. Its size is configurable but it may not contain more than 191 entries. This area is assigned CST numbers 1-%277. The second area is used for programs. The total number of entries in this area is not hardware limited. This area is allocated a block at a time with one program per block. A block may contain from 1 to 63 segments, which will be assigned CST entry numbers %301-%377. The maximum number of segments in a program file is 63 and segments of different programs will have the same CST number. Thus both a block number and a CST# are required to uniquely identify a program segment. A fallout of this is that logical segment=physical CST-%301.

The loader is a system process which will do loads sequentially. If a process needs code to be loaded, it will get the load process' SIR, fill a communication data segment and then awake the loader. Upon completion, the loader will return its status through the communication data segment and then activate the waiting process.

LOADER SEGMENT ALLOCATION

The order in which storage is allocated for arrays is arbitrary, with one exception: The storage for array DIR must be last in the data segment. This allows the data segment expansion/contraction intrinsics to be applied so that DIR storage may be dynamically allocated.

```
               ---------------------------
               |     DATA LABELS         |  PRIMARY DB
               |-------------------------|
(DB+3)------>  |     REF TAB             |  REFERENCE COUNT TABLE
               |    (ref count)          |
               |-------------------------|
(DB+4)------>  |       XFORM             |  SEGMENT TRANSFORM TABLE
               |-------------------------|
(DB+5)------>  |       ENT TAB           |  ENTRY INDEX TABLE
               |-------------------------|
               |        SBUF 0           | \
               |-------------------------| |
               |        SBUF 1           | |
               |-------------------------|  >  UTILITY BUFFERS
               |        SBUF 2           | |   (128 words each)
               |-------------------------| |
               |        SBUF 3           | |
               |-------------------------| |
               |        SBUF 4           | /
               |-------------------------|
               |                         |
(DB+2)------>  |                         |
               |                         |
               ~          DIR            ~
               ~                         ~     DIRECTORY
               |                         |
               |                         |
               ---------------------------
```

LOADER SEGMENT TABLE PRIMARY DB (DST %22)

0	UTILITY INTEGER	S0
1	DIRECTORY LENGTH	DIRLEN
2	ENTRY TABLE POINTER	DIR
3	REFERENCE COUNT TABLE POINTER	REFCOUNT
4	CST TO LCST AND FLAG TABLE POINTER	XFORM
5	CST TO ENTRY INDEX TABLE POINTER	ENTTAB
6	SECONDARY ENTTAB POINTER	ENTP2
7	ENTRY POINTER	ENTP
10	SECONDARY ENTRY POINTER	ENTP1
11	SECOND RECORD DISC BUFFER POINTER	SBUF0
12	"	SBUF1
13	"	SBUF2
14	"	SBUF3
15	"	SBUF4
16	UTILITY INTEGER	SI
17	"	SJ
20	"	SK
21	"	SL
22	"	SM
23	"	SN
24	"	SP
25	"	SQ
26	"	SR
27	"	SS
30	"	ST

REFERENCE COUNT TABLE

(DB + 3)

```
|-----------------|
|                 |
|                 |
|      REFTAB     |
|                 |
|                 |
|-----------------|
```

Indexed by CST number; contains the reference count for each code segment. Contains -1 if the CST entry is not allocated.

SEGMENT TRANSFORM TABLE

(DB + 4)

```
  LEFT     RIGHT
  BYTE     BYTE
|--------|--------|
|LOG CST#| FLAGS  |
|--------|--------|
|                 |
|      XFORM      |
|                 |
|-----------------|
```

Indexed by CST number; contains the file-relative (logical) segment number and segment attributes.

```
| 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|         SEG#          |  T  | A| C| X|////////|
|-----------------------|-----|--|--|--|--------|
```

T-Segment Type: System SL =0
Public SL =1
Group SL =2
Program Seg =3

A-Perm.Allocated Segment (1/0)
C-Core Resident Segment (1/0)
X-System (MPE) Segment (1/0)

ENTRY INDEX TABLE

(DB + 5)

```
|-----------------|
|                 |
|                 |      Indexed by CST number; contains the
|     ENT TAB     |      directory index of the file entry
|                 |      corresponding to the CST number.
|                 |
|-----------------|
```

DIRECTORY

(DB + 2)

```
|-----------------|
|                 |
|                 |      Accessed by entry key - contains vari-
|      DIR        |      able length entries, each entry describ-
|                 |      ing a set of CST numbers.
|                 |
|-----------------|
```

The directory is completely filled with variable length entries. The empty state is represented by a single garbage entry. It is accessed by a sequential search using a double word entry key, or by direct indexing using ENTTAB.

The first word of each entry has the same format and includes an entry type number. In addition, most entries (all entries except type garbage) have an implicit double word entry key. Those entries that have an explicit single word key have an additional word that is implicitly 0. The entry key immediately follows the entry descriptor (first) word.

For file entries, the key is the double word sector number of the file label with the first byte of the double word replaced with the logical device number. For process entries, the key is the single word PIN with the first byte of the single word replaced with the extension number (LOADPROC id number).

ENTRY POINTER (ENTP)

(DB + 7)

```
     | 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|
     |-----------------------------------------------|
   0 |                    | A|  LS | F| P|  |   ET   |
     |-----------------------------------------------|
   1 |       #wds in garbage entry/process id        |  ENWG,EPID*,EF
ID1
     |-----------------------------------------------|
   2 |             Second word of file ID            |  EF102
     |-----------------------------------------------|
   3 |              Working set pointer              |  EWSP
     |-----------------------------------------------|
   4 |                CST block index                |  ECST
     |-----------------------------------------------|
   5 |           Prog file reference count           |  ESHR
     |-----------------------------------------------|
   6 |               #Segments in file               |  ESEG
     |-----------------------------------------------|
```

A = Program Allocated
LS = Library Search
F = File Mode
P = Program Mode
ET = Entry Type

*EPID

```
| 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|
|-----------------------------------------------|
|  EXTENSION NUMBER      |       PIN NUMBER     |
|-----------------------------------------------|
```

EFID1 = First word of file ID

```
                    SBUF0 (DB + 9)
                    -----

     | 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|
     |-----------------------------------------------|
  0  | F| N| Z|/////////|        CAP LIST             | SFAGS
     |-----------------------------------------------|
  1  |              Number segments                  | SNRSEGS
     |-----------------------------------------------|
  2  |              Global area size                 | SGLOBALSIZE
     |-----------------------------------------------|
  3  |            REC. NR. of global area            | SGLOBALRECD
     |-----------------------------------------------|
  4  |           Rec. nr. of segment list            | SSEGMENTRECD
     |-----------------------------------------------|
  5  |                 Stack size                    | SSTACKSIZE
     |-----------------------------------------------|
  6  |                  DL size                      | SDLSIZE
     |-----------------------------------------------|
  7  |             Max. data seg. size               | SMAXDATA
     |-----------------------------------------------|
 10  |          Rec. nr. of entry point list         | SENTRYRECD
     |-----------------------------------------------|
 11  |            Starting segment nr.               | SSTARTINGSEG
     |-----------------------------------------------|
 12  |            Starting PB address                | SSTARTINGADR
     |-----------------------------------------------|
 13  |         Starting address of STLT              | SSASTLT
     |-----------------------------------------------|
 14  |         Starting address of FLUT              | SSAFLUT
     |-----------------------------------------------|
 15  |          Rec. Nr. of external list            | SEXTERNALRECD
     |-----------------------------------------------|
 16  |              Starting SST Nr.                 | SSTARTINGSST
     |-----------------------------------------------|
 17  |         Starting address of trapcom.          | SSATRAPCOM
     |-----------------------------------------------|
```

F = Fatal Error
N = Non-Fatal Error
Z = Zero DB

DIRECTORY ENTRIES

```
   | 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--------|  GARBAGE(0)
|  |                       |     |  |  |  |   0    |0 -------
|  |-----------------------|-----|--|--|--|--------|
|NWG                 NWG                           |1 FREE SPACE
|  |-----------------------------------------------|
|  |                                               |
|  |                  GARBAGE                      |
|  |                                               |
-- |-----------------------------------------------|

   |                                |10|11|12|13|14|15|
-- |--------------------|--------|--|--|--|--------|  SL FILE(1)
|  |                    |        | M|  |  |   1    |0 --------
|  |--------------------|--------|--|--|--|--------|  Indicates which
|  |                                               |1 CSTs are being
|  |------------------FID--------------------------|  used for the
16 |                                               |2 segments of the
|  |-----------------------------------------------|  SL file.
|  |               PVINFO                          |3 PVINFO:     $
|  |-----------------------------------------------|  0:4- unused  $
|  |               CSTARRAY                        |4 4:4- MVTAB inx $
|  |                                               |. 8:8- vols mtd. $
|  |                                               |. (master=bit 15)$
-- |-----------------------------------------------|.

   |                         7| 9   |10|11|12|13|14|15|
-- |-----------------------------|--|--|--|--------|
|  |                    | A| LIB | M| P|  |   2    |0 Program File
                                                     Directory (2)
|  |--------------------|--|-----|--|--|--|--------|  ----------------
|  |                                               |1 Indicates which
|  |------------------FID--------------------------|  CST's are being
|  |                                               |2 used for the
|  |-----------------------------------------------|  segments of the
|  |                                               |3 program file and
|  |-----------------------------------------------|  its internals.
|  |  CST block index                              |4
|  |-----------------------------------------------|
|  |  #process sharing                             |5
|  |-----------------------------------------------|
20 |  #segments in prog. file                      |6
|  |-----------------------------------------------|
21 |            PVINFO                             |.          $
   |-----------------------------------------------|           $
|  |///////////////////////////////////////////////|
|  |///////////////////////////////////////////////|
|  |  CST bit map                                  |.
|  |///////////////////////////////////////////////|
|  |///////////////////////////////////////////////|
-- |-----------------------------------------------|
```

A: set if program file is allocated

DIRECTORY ENTRIES (CONT.)

```
    |                            |10|11|12|13|14|15|
 -- |----------------------------|--|--|--|--------|   LOADING (3)
 |  |                            | M| P|  |   3    |0  -------
 |  |----------------------------|--|--|--|--------|
 3  |                                              |1  Indicates that
 |  |-------------------FID------------------------|   the program
 |  |                                              |2  file is being
 -- |----------------------------------------------|   loaded.

    |                            |10|11|12|13|14|15|
 -- |----------------------------|--|--|--|--------|   WAITER(4)
 |  |                            | M| P|  |   4    |0  ------
 |  |----------------------------|--|--|--|--------|
 |  |                                              |1  Indicates that
 |  |-------------------FID------------------------|   a process is
 5  |                                              |2  waiting for the
 |  |----------------------------------------------|   program file to
 |  |              CREATER PIN                     |3  be loaded.
 |  |----------------------------------------------|
 |  |               (NOT USED)                     |4
 -- |----------------------------------------------|


    | 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|
 -- |--|--------|-----------|-----|--|--|--|--------|   LOADED(5)
 |  |                            | M| P|  |   5    |0  ------
 |  |----------------------------|--|--|--|--------|
 |  |                                              |1  Indicates that
 |  |-------------------FID------------------------|   a program file
 5  |                                              |2  has been loaded
 |  |----------------------------------------------|
 |  |                                              |3
 |  |            LOAD PROCESS STATUS               |
 |  |                                              |4
 -- |----------------------------------------------|

    |                            |10|11|12|13|14|15|
 -- |----------------------------|--|--|--|--------|   SHARER(6)
 |  |                            | M| P|  |   6    |0  ------
 |  |----------------------------|--|--|--|--------|
 |  |           0           |         PIN          |   Indicates that
 4  |-----------------------|----------------------|   a process is
 |  |                                              |2  running the
 |  |---------------------FID----------------------|   program file.
 |  |                                              |3
 -- |----------------------------------------------|
```

DIRECTORY ENTRIES (CONT.)

```
   |                                |11|12|13|14|15|
-- |--------------------------------|--|--|--------|   EXTENSION(7)
|  |                       |        |  | F|   7    |0  ---------
|  |-----------------------|--------|--|--|--------|   Indicates that
|  |       EXT             |           PIN         |1  a process has
|  |-----------------------|-----------------------|   loadproced a
14 |                                               |2  procedure.
|  |                                               |
|  |                 CST ARRAY                     |
|  |                                               |
|  |                                               |
-- |-----------------------------------------------|


-- |--|--|--|------------------------------------|  CST ARRAY(BIT MAP)
|  | 0| 1| 2| .  .  .  .  .                       |  ---------
|  |-----------------------------------------------|
|  |                                               |
|  |                       .                       |
|  |                       .                       |
12 |                       .                       |
|  |                       .                       |
|  |                       .                       |
|  |                                               |
|  |                                               |
|  |-----------------------------------------|--|--|
|  |                            .  .  .  .  .|76|77|
-- |-----------------------------------------|--|--|
```

DEFINITIONS

```
NWG  - #words in garbage entry.
---

FID  - file ID.
---
        word 1-(0:8)=log dev#
        word 1-(8:8)=msb of disc address
        word 2-=lsb of disc address
LIB  - 0=SSL, 1=PSL, 2=GSL.
F    - CST array format (0=list, 1=bit map)
-

M    - executing mode.  indicates whether the segments for the file
-      have been copied onto the system disc (1=fast) or not (slow).
```

DEFINITIONS (CONT.)

T - entry type

0	GARBAGE	self explanatory
1	SL	indicates which CST's are being used for segments of the file. Currently F=1 and M=0 for all SL entries.
2	PROGRAM	indicates which CST's are being used for segments of the file and all its externals. Currently M=0 for all program entries.
3	LOADING	indicates that a program file (FID) is being loaded on behalf of a process (PIN).
4	WAITING	indicates that a process (PIN) is waiting for a program file (FID) to be loaded.
5	LOADED	transformed entry of type 4 that is used to return status of load.
6	SHARER	indicates that a process (PIN) is currently running a program file (FID).
7	EXTENSION	indicates that a process (PIN) has LOADPROCed a procedure (1<=EXT<=225).

P - program mode bit=0 (normal) everything that should be in priv is in priv mode and likewise for non-
=1 (NOPRIV) everything in non-priv mode.

$$$$$ LOADER CACHE

SYGLOB extension area + %72 contains DST number of cache
BUCKETSIZE = %52

```
                         CACHE DATA SEGMENT FORMAT
                        --------------------------
                        0|                        |
                        1|   HIT COUNTER          |
                        --------------------------
                        2|                        |
                        3|   MISS COUNTER         |
                        --------------------------
                        4|      BUCKET 0          |
                        --------------------------
      4+ BUCKETSIZE     |       BUCKET1           |
                        --------------------------
                                    .
                                    .
                                    .
                        --------------------------
4+94* BUCKETSIZE        |                         |
                        |       BUCKET 94         |
4+95* BUCKETSIZE -1     |                         |
                        --------------------------


      BUCKET FORMAT

      -----------------
    0 |   Length of   |
      |   SLDIR1 +1   |
      -----------------
    1 | SLDIR 1       | Most recently referenced system SL
      |               | directory entry from this SL directory
      |---------------| bucket
      |  LENGTH OF    |
      |  SLDIR2 + 1   |
      -----------------
      | SLDIR 2       | Second most recently referenced entry
      -----------------
      .               .
      .               .
      -----------------
      |  LENGTH OF    |
      | SLDIRN + 1    |
      -----------------
BUCKET| SLDIRN        | Nth most recently referenced entry;if
SIZE-1|---------------| not complete then indicates end of
                        bucket
```

All bucket words are initalized to BUCKETSIZE +1, indicating no entries.

LOADER COMMUNICATION TABLE (LCT) SYSDB + %220

Form incoming to Loader

```
    | 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|
    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 0  | CMD | LIB | M|  | L|  | PROG    PIN           |   COMMAND
    |-----|-----|--|--|--|--|-----------------------|
 1  | LOGICAL DEVICE #      |         DISC          |   PROGRAM FILE
    |-----------------------|-----------------------|
 2  |       ADDRESS                                 |   DESCRIPTOR
    |-----------------------------------------------|
 3  |  # CHARS IN NAME      |                       |   CMD=loader cmd
    |-----------------------|                       |       0=load prgm
 4  |                                               |       1=load proc
    |                                               |       2=alloc prog
 5  |     PROCEDURE                                 |       3=alloc proc
    |                                               |   LIB=library
 6  |                                               |       search
    |                                               |       0=SYS
 7  |        NAME                                   |       1=PUB
    |                                               |       2=GROUP
 8  |                                               |
    |                                               |
 9  |                                               |   M=NONPRIV MODE
    |                                               |
10  |                                               |   L=LOAD MAP REQ.
    |-----------------------------------------------|
11  |    WAITER   PCB   INDEX                       |
    |-----------------------------------------------|
12  |                    |BA|IA|PM|     |MR|  |DS|PH|   USER CAPABILITY
    |--------------------|--|--|--|-----|--|--|--|--|
13  |                                               |
    |                                               |
14  |           GROUP                               |
    |                                               |
15  |                   NAME                        |
    |                                               |
16  |                                               |
    |-----------------------------------------------|
17  |                                               |
    |                                               |
18  |           ACCOUNT                             |
    |                                               |
19  |                   NAME                        |
    |                                               |
20  |                                               |
    |-----------------------------------------------|
21  |     PVINFO (see "DIRECTORY ENTRIES")          |         $
    |-----------------------------------------------|         $
```

LCT (CONT.)

Form returned to WAITER

```
   |-----------------------------------------------|
 0 |  F.S. ERROR OR STARTING CST #                  |
   |-----------------------------------------------|
 1 |    LOAD PROCESS ERROR NUMBER                   |
   |-----------------------------------------------|
 2 |    LOAD MAP FLAG                              |TRUE IF LMAP
   |-----------------------------------------------|PROVIDED
 3 |                       |    LDEV               | \
   |-----------------------|-----------------------| |
 4 |          DISC                                 | |   LOAD MAP DISC
   |                                               | |   FILE DISCRIPTOR
 5 |                 ADDRESS                       | |
   |-----------------------------------------------| /
```

MVTAB (Mounted Volume Table)

```
                       1 1 1 1 1 1
   0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
  |-------------------------------| -------------------
 0|  entry size   :  max entries  | 0                 |
  |-------------------------------|                   |
 1|  # of mounted volume sets     | 1                 |
  |-------------------------------|                   |
 2|    ldev       :   DIRBASE     | 2 master volume of|
  |----------------              -|   SYS VS is always|
 3|     of SYSTEM volume set      | 3 ldev = 1.       |
  |-------------------------------|                   |
 4|               0               | 4                 |
  |-------------------------------|                   |
 5|               0               | 5                 |
  |-------------------------------|                   |-- entry 0
  |                               |                   |  (MVTABX = 0)
                                                      |
                                                      |
                                                      |
  |                               |                   |
  |-------------------------------|                   |
17|               0               |21                 |
  |-------------------------------|                   |
18|               0               |22                 |
  |-------------------------------|                   |
19|               0               |23                 |
  |-------------------------------|                   |
20|               0               |24                 |
  |-------------------------------| -------------------
```

MVTAB (CONT.)

```
  |--------------------------------| ------------------
 0|0:cycl://///////////////////////| 0                 |
  |--------------------------------|                   |
 1|hvol:nvol:       ucnt           | 1                 |
  |--------------------------------|                   |
 2|    ldev       :   DIRBASE      | 2 master volume   |
  |----------------                |   of volume set   |
 3|        of volume set           | 3 is on this ldev |
  |--------------------------------|                   |
 4|       generation number        | 4                 |
  |--------------------------------| ----              |
 5|     ldev       :      VTABX    | 5  |              |
  |--------------------------------|    |- vol entry 0 |-- entry 1
 6|///////////////:      vcnt      | 6  |  (double)    |  (MVTABX = 1)
  |--------------------------------| ----              |
  |                                |                   |
                 .                                     |
                 .                                     |
  |                                |                   |
  |--------------------------------| ----              |
19|     ldev       :      VTABX    |23  |              |
  |--------------------------------|    |- vol entry 7 |
20|///////////////:      vcnt      |24  |  (double)    |
  |--------------------------------| ---- -------------|
  |                                |
  |                                |
  |                                |


  |                                |
  |                                |
  |                                |
  |--------------------------------| ------------------|
  |                                |                   |
  |                                |                   |
  |                                |                   |
                                                       |
                                                       |-- entry n-1
                                                       |  (MVTABX = n-1)
  |                                |                   |
  |                                |                   |
  |                                |                   |
  |--------------------------------| ------------------
```

MVTAB (CONT.)

```
  |--------------------------------| -----------------
 0|0:cycl:////////////////////////| 0                |
  |--------------------------------|                  |
 1|hvol:nvol:       ucnt           | 1                |
  |--------------------------------|                  |
 2|    ldev       :    DIRBASE     | 2                |
  |----------------                |                  |
 3|        of volume set           | 3                |
  |--------------------------------|                  |
 4|      generation number         | 4                |
  |--------------------------------| ----             |
 5|     ldev      :     VTABX      | 5  |             |
  |--------------------------------|    |- vol entry 0 |-- entry n
 6|////////////////:     vcnt      | 6  | (double)     |  (MVTABX = n)
  |--------------------------------| ----             |
  |                                |                  |
                  .                                   |
                  .                                   |
  |                                |                  |
  |--------------------------------| ----             |
19|     ldev      :     VTABX      |23  |             |
  |--------------------------------|    |- vol entry 7 |
20|////////////////:     vcnt      |24  | (double)     |
  |--------------------------------| ---- ------------|
```

cycl - cyclical volume index (local VTABX) for disc space allocation

hvol - highest (ordinal) volume index (volume index being the volume set's local VTABX) of a mounted member of the volume set(class).

nvol - # of volumes mounted for the volume set(class).

ucnt - # of users having mounted the volume set.

vcnt - # of users having mounted the volume.

```
        PVUSER (Private Volume User Table)
        ----------------------------------
                        1 1 1 1 1 1
    0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
   |-------------------------------| -------------------
  0|      table size (words)       | 0                 |
   |-------------------------------|                   |
  1|///////////////:  # of entries | 1                 |
   |-------------------------------|                   |
  2|bitmask of MVTABX's represented| 2                 |
   |-------------------------------|                   |-- table head
 $3| maximum table size ( words )  | 3                 |   (5 words)
   |-------------------------------|                   |
  4|      available pointer        | 4                 |
   |-------------------------------| ------------------|
   |    op mask    :    MVTABX     |    |              |
   |-------------------------------|    |              |
   |   max users    :    # pins    |    |              |
   |-------------------------------|    |- entry head  |
   |   current size of entry       |    |  (4 words)   |
   |-------------------------------|    |              |
 $ |      PV flags             |OP |    |              |
   |-------------------------------| ----              |
   |    vmask      :     pin       |    |              |
   |-------------------------------|    |              |
   |       user bind count         |    |              |
   |-------------------------------|    |              |
   |      user mount count         |    |              |
   |-------------------------------|    |              |
   |      system bind count        |    |- user entry 1|
   |-------------------------------|    |              |
   |      system mount count       |    |              |
   |-------------------------------|    |              |
   |       bind names count        |    |              |
   |-------------------------------|    |              |
   |  DST # of bind names segment  |    |              |
   |-------------------------------| ----              |
   |    vmask      :     pin       |    |              |
   |-------------------------------|    |              |-- volume set
   |       user bind count         |    |              |   entry 1
   |-------------------------------|    |              |   (MVTABX = j)
   |      user mount count         |    |              |
   |-------------------------------|    |              |
   |      system bind count        |    |- user entry 2|
   |-------------------------------|    |              |
   |      system mount count       |    |              |
   |-------------------------------|    |              |
   |       bind names count        |    |              |
   |-------------------------------|    |              |
   |  DST # of bind names segment  |    |              |
   |-------------------------------| ----              |
                   .                                   |
                   .                                   |
   |                               |                   |
 PVUSER (CONT.)
```

```
------

  |-------------------------------| ----              |
  |    vmask      :     pin       |    |              |
  |-------------------------------|    |              |
  |       user bind count         |    |              |
  |-------------------------------|    |              |
  |      user mount count         |    |              |
  |-------------------------------|    |              |
  |      system bind count        |    |- user entry n|
  |-------------------------------|    |              |
  |      system mount count       |    |              |
  |-------------------------------|    |              |
  |       bind names count        |    |              |
  |-------------------------------|    |              |
  |  DST # of bind names segment  |    |              |
  |-------------------------------| ------------------|
  |                               |
  |                               |
  |                               |



  |                               |
  |                               |
  |                               |
  |-------------------------------| ------------------|
  |    op mask    :    MVTABX     |                   |
  |-------------------------------|                   |
  |                               |                   |
  |                               |                   |
                                                      |-- volume set
                                                      |   entry n
                                                      |   (MVTABX = k)
  |                               |                   |
  |                               |                   |
  |-------------------------------| ------------------|
  |                               |
  |               a               |
  |               v               |
  |               a               |
                  i
                  l
                  a
  |               b               |
  |               l               |
  |               e               |
  |                               |
  |-------------------------------|
```

Bind Names Data Segment

(Created and managed via PVUSER Table)

```
                       1 1 1 1 1 1
   0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
  |-------------------------------|  -----------
 0|      max segment length       | 0          |
  |-------------------------------|            |
 1|    current segment length     | 1          |
  |-------------------------------|            |
 2|               0               | 2          |
  |-------------------------------|            |
  |                               |            |
  |                               |            |-- entry 0
                                               |
                                               |
                                               |
  |                               |            |
  |                               |            |
  |-------------------------------|            |
 8|               0               |10          |
  |-------------------------------|  ----------|
 0|          bind count           | 0          |
  |-------------------------------|            |
 1|                               | 1          |
  |-          G R O U P          -|            |
 2|                               | 2          |
  |-           N A M E           -|            |
 3|                               | 3          |
  |-                             -|            |
 4|                               | 4          |-- entry 1
  |-------------------------------|            |
 5|                               | 5          |
  |-        A C C O U N T        -|            |
 6|                               | 6          |
  |-           N A M E           -|            |
 7|                               | 7          |
  |-                             -|            |
 8|                               |10          |
  |-------------------------------|  -----------
  |                               |
  |                               |
  |                               |
```

BIND NAMES DATA SEGMENT (CONT.)

```
  |                               |
  |                               |
  |-------------------------------| ------------
 0|         bind count            | 0          |
  |-------------------------------|            |
 1|                               | 1          |
  |-         G R O U P           -|            |
 2|                               | 2          |
  |-          N A M E            -|            |
 3|                               | 3          |
  |-                             -|            |
 4|                               | 4          |-- entry n
  |-------------------------------|            |
 5|                               | 5          |
  |-        A C C O U N T        -|            |
 6|                               | 6          |
  |-           N A M E           -|            |
 7|                               | 7          |
  |-                             -|            |
 8|                               |10          |
  |-------------------------------| ------------
  |              a                |
  |              v                |
  |              a                |
                 i
                 l
                 a
                 b
  |              l                |
  |              e                |
  |                               |
  |-------------------------------|
```

Data Record format

The primary purpose of the Serial Disc Interface (SDISC) is to adapt the undefined length transfers characteristic of mag tape to the fixed length environment of a disc or integrated cartridge tape (ICT). To accomplish this, data is buffered within SDISC. The buffer is an integral number of sectors (blocks for the ICT) long. Files always start on a sector boundary, but data records within files may start anywhere and straddle sector boundaries. A record in the buffer is structured as follows:

```
     +---------+-----------------------------+---------+
     | record  |                             | record  |
     | length  |             data            | length  |
     | (bytes) |                             | (bytes) |
     +---------+-----------------------------+---------+
```

The record length is always a one-word positive byte count which includes only the data portion of the record, not the length words themselves. Records within a file might be stored on the disc as follows:

```
     +----+----------------------------+---       -----
     | RL |///////////////////////////|             ^
     +----+------+----+----+----------+             |
     |///////////| RL | RL |//////////|             |
     +-----------+----+--+-+--+----+--|         Sector N-1
     |///////////////////| RL | RL |//|             |
     +-------------------+----+----+--|             |
     |////////////////////////////////|             v
     +---+----+----+-----------------+---       -----
     |///| RL | RL |//////////////////|             ^
     +---+----+--+-+--+----+----------+             |
     |///////////| RL | RL |//////////|             |
     +-----------+----+----++----+----+         Sector N
     |//////////////////////| RL | RL |             |
     +----------------------+----+----+             |
     |////////////////////////////////|             v
     +-----+----+----+----------------+---       -----
     |/////| RL | RL | .....          |
     +-----+----+----+----------------+
```

The reason for the trailing byte count is to implement an easy way to backspace records.

End of File format

Since files always start on a sector boundary, it follows that they also end on one. End of files consist of a 0 record length and 0-fill to the end of the current sector as follows:

```
+-------------------------------+
|/////////////////////// RL RL /|
|///////////////////////////////|
|//////// RL RL ////////////////|        Sector N
|                    +----------+
|///////////////| RL | 0        |
+--------------------+          |
|                               |
|          Zero fill            |
+-------------------------------+---
```

In addition, an End-of-File entry is made in the Gap Table, so that files may be skipped by scanning Gap Table entries instead of serially scanning the data area. The Gap Table is described a few pages from now.

Contiguous Block format

A serial disc, if it can do everything a mag tape can do, must also be a cold-load device. This means that machine microcode must be able to read a bootstrap channel program and the resident segments of INITIAL from the disc into memory. The microcode and channel programs cannot deal with the record length words which surround standard data records, so for them we have a structure, called a CONTIGUOUS BLOCK, which has the data without the length words. Information as to the length of each contiguous block must therefore be kept elsewhere, so there are Gap Table entries which hold the beginning and ending sector addresses of each contiguous block. This implies that each block must begin and end on a sector boundary. In this way they are similar to data files. To set contiguous blocks off from normal data, and to reach a sector boundary, a record length and fill character = %177777 is used, as follows:

```
+-------------------------------+---       -----
|/////// Previous records //////|            ^
|///////////////////////////////|            |
|             +-----------------+            |
|////////| RL | -1              |        Sector N-1
+-------------+                 |            |
|                               |            |
|            -1 fill            |            |
|                               |            v
+-------------------------------+---       -----
|                               |            ^
|         Contiguous block      |        Sector N
|                               |            v
|                               +---       -----
|                               |            ^
|                +--------------+            |
|                |              |        Sector N+1
+----------------+              |            |
|            -1 fill            |            v
+-------------------------------+---       -----
```

Hole format

Holes on the serial disc have the same format as contiguous blocks (that is, they start and end on sector boundaries with -1 fill characters as required). They are generated during write error processing on the HP7920 or HP7925 large discs, or the HP7902 or HP9895 floppy discs. They are at least one track long. Write errors rarely occur in actual use, so holes are similarly rare. Further details may be found in the Serial Disc IMS.

Gap Table format

The Gap Table is a four-word header followed by a series of two-word device address entries. A permanent copy lives on the device, starting in sector 4, while a working copy lives in main memory. The copy in memory is posted to the disc only when a backspace or rewind operation occurs after writing (in other words, when the copy in main memory has changed). The length of the Gap Table is device-dependent according to the table below:

Device	Number of sectors (or ICT blocks)
HP7920	44
HP7925	106
HP7935	219
HP7902/9895	26
ICT	4 blocks ("S" cartridge)
	5 blocks ("L" cartridge)

The Gap Table looks like this:

```
   +---------------------------+
 0 | sector addr of load point |\
 1 |          unused           | \
 2 |          unused           | /-Gap Table header
 3 |          unused           |/
   +------+--------------------+
 4 | type |                    |
   +------+   Sector address   |   Entry (two words)
 5 |                           |
   +------+--------------------+
 6 | type |                    |
   +------+   Sector address   |   Entry (two words)
 7 |                           |
   +---------------------------+
                  .
                  .
                  .
```

The type field is bits 0, 1 and 2 of the first word. The eight possible types are:

0. End of File. The associated sector address contains one or more end of file fill characters (0) to fill out that sector. In the worst case (the previous record ended exactly at the end of the previous sector), the end of file sector contains all zeros.
1. End of data. The associated sector address is the last address of valid data plus 1, in other words, the next available address. In practice, such an entry is usually preceded by an end-of-file entry, since the EOD entry is written when you stop writing, and the file system will not let you backspace or rewind after writing without sending a Write End of File. An EOD entry is also written at the beginning of the Gap Table when new (unwritten) media is inserted. This prevents erroneous reading of blank media.

2. Beginning of Hole. The starting address of a "defective" area of the disc. Usually on a track boundary, but may be in mid-track if a contiguous block was being written when the "defect" was encountered.
3. End of Hole. The corresponding ending address of the "defective" area. Always at a track boundary.
4. Beginning of (contiguous) Block. The starting address of a contiguous block, exclusive of the -1 fill characters which may have been required to get us to a sector boundary. Unlike the End of File fill characters, there need not be any -1 characters if the previous record or contiguous block (with or without the trailing length word) ended exactly on a sector boundary.
5. End of (contiguous) Block. The address of the last sector containing contiguous block data. The sector may also contain -1 fill characters to get us to a sector boundary, but as with the beginning of block they are not required if the contiguous block ends exactly on a sector boundary.
6. End of Tape mark. The sector address of the simulated End of Tape reflector. This type is now written only to floppy discs for use by INITIAL's serial disc interface. When read by MPE's SDISC, it will be skipped no matter what device it is found on. This ensures compatibility with older serial discs.
7. End of Gap Table. No associated sector address. If you hit this while scanning the Gap Table, you've gone too far. In practice, this type is created whenever the Gap Table is cleared, by the simple device of initializing the table to -1.

SDISC Extra Data Segment

With insignificant exceptions, SDISC operates entirely in split-stack mode, that is, using an extra data segment for its working storage. Since SDISCI runs on the user's stack (under the File System and ATTACHIO), it reall wouldn't do to have the user support a 16K RECBUFF (for an ICT) or a 13.6K Ga Table (for a 7925) on his stack.

The extra data segment (XDS) is usually acquired by the external procedur ALLOCATE when the serial disc device is first assigned to a user as part of a FOPEN. The external procedure DEALLOCATE makes the XDS go away as part of it processing of the final FCLOSE against the device. The system program PVPRO may also acquire and release an XDS so that the tape label routines in LABSE may also use SDISC for their work when DEVREC processes a device on-line interrupt.

In addition to the Gap Table already described, the XDS contains a data buffer (RECBUFF), SDISC's global storage area and a small buffer (called WORKTABLE) used to hold data while moving it from a "defective" disc area to its ne location as part of the process of creating a hole. WORKTABLE also holds th contents of the Serial Disc label sector when SDISC reads it in as part of it self-configuration.

The three arrays in the XDS (WORKTABLE, RECBUFF and GPT (Gap Table)) are al dynamically configured by SDISC as vanilla indirect arrays, such as might hav been constructed by SPL. This is done by declaring the array names as pointers, then inserting appropriately computed element-0 addresses in them.

The extra data segment is organized as follows:

```
   +-------------------+   These twelve words are reserved
 0 | WORDSPERSECTR     |   for use by ALLOCATE when the data
   |. . . . . . . . . .|   segment is created.  However, AL-
 1 | SECTORSPERTRAK    |   LOCATE only stuffs the last five
   |. . . . . . . . . .|   of them.  We fill the first seven
 2 | STARTADDRESS (BOT)|   ourselves with information we get
   |. . . . . . . . . .|   from the label sector.
 3 | EOTSECTR (disc    |
   | address of simu-  |
 4 | lated end of tape)|
   |. . . . . . . . . .|
 5 | EODSECTR (last    |
   | sector of disc)   |   Simulates tape runoff.
 6 |                   |
   |. . . . . . . . . .|
 7 | JUSTALLOCATED     |   Tells us to initialize SDISC
   |. . . . . . . . . .|     parameters to BOT if true.
 8 | WRITERING         |   Simulation of tape write ring.
   |. . . . . . . . . .|
 9 | FATALERROR        |   Disables SDISC when true.
   |. . . . . . . . . .|
10 | LPERRORLOG        |   Dumps XDS and user stack to LP
   |                   |     if true and FATALERROR occurs.
   |                   |     Currently may be set only in
   |. . . . . . . . . .|     DEBUG.
```

```
   |. . . . . . . . . .|
11 | MAX'DSEG'SIZE     |   Max size of our XDS, so we can
   +-------------------+     check that it's big enough.
   | SDISC global vari-|
   |   ables, including|
   |   array pointers. |
   +-------------------+
   | W                 |
   |   O               |   Length is WORDSPERSECTR.
   |     R             |
   |       K           |
   |         T         |
   |           A       |
   |             B     |
   |               L   |
   |                 E |
   +-------------------+
   | R                 |
   |   E               |   Length is calculated as 32 *
   |     C             |     WORDSPERSECTR.
   |       B           |
   |         U         |
   |           F       |
   |             F     |
   +-------------------+
   | G                 |   Length varies with device, and is
   |   A               |     calculated by SDISC as part of its
   |     P             |     self-configuration.
   |                   |
   |         T         |
   |           A       |
   |             B     |
   |               L   |
   |                 E |
   +-------------------+
```

Serial disc organization

The disc is organized as follows:

```
+--------------------+
| Label sector       |   0   See expanded view in Chapter 3.
+--------------------+
| Defective Trk Tbl  |   1   Maintained by disc driver, not
+--------------------+         used by SDISC.
| Cold load          |   2   HP-IB cold load channel prog.
+--------------------+
| Soft dump          |   3   SOFTDUMP channel program.
+--------------------+
| Gap Table          |   4 to STARTADDRESS - 1.
|      .             |
|      .             |
+--------------------+
| Data               |   STARTADDRESS
|      .             |        .
|      .             |        .
|      .             |        to
|. . . . . . . . . . |        .
|      .             |   EOTSECTR
|. . . . . . . . . . |        .
|      .             |        to
|. . . . . . . . . . |        .
| Last data sector   |   EODSECTR
+--------------------+
```

CHAPTER 13 I/O

```
                                   ----------
|-------------------------------->/ LOGICAL   \  I/O TABLE LINKAGE
|                                 \ DEVICE   /   -----------------
|                                  -----------
|                                       |
|                                       |
|                                    ----------
|                              --- |  DITP    |
|                              |    ----------  LPDT
|                              |   | FLAGS    |
|                              |    ----------
|                              |    ----------
|                              --->| FLAGS    |<-------------
|                                   ----------    DIT       |
|                                  | DITP     |             |
|                                   ----------              |
|   -------------------------------| IOQP*    |             |
|   |                               ----------              |
|   |                              |UNIT|LDEV |             |
|   |                               ----------              |
|   |            IOQ*       -------| DLTP     |             |
|   |      ----------       |       ----------              |
|   ----->| FLAGS    |      |      | ILTP     |---          |
|          ----------       |       ----------   |  ILT     |
|         | IOQP     |      |         ------------          |
|          ----------       |         |---> -----------     |
----------|    |LDEV |      |      ------->|Ch|   |DRT |    |
           ----------       |      |    --- -----------     |
          |          |      |      |    |  |  SIOP     |    |
                            |      |    |   -----------     |
                            |      |    |  |UNIT EXTRACT    |
                            |      |    |   -----------     |
                            |      |    |  |SIOP | Q # |    |
                            |      |    |  |SIZE |     |    |
           ----------       |      |    |   -----------     |
          |          |      |      |    |  |           |    |
          |  DLT     |<------      |    |  |           |    |
          |          |             |    |   -----------     |+UNIT
          |          |             |    |  | DITP      |<----
           ----------              |    |  |-----------|
                                   |    |
                                   |    |
                                   |    |  |----------|
                                   v    -->|SIO PROG AREA
                                   |        -----------
                                   |       |   DRT     |
                                   |        -----------
                                   |       |  SIOP     |
                                   |        -----------
                                   |       |  PI       |
                                   |        -----------
                                   ------->|  ILTP     |
                                            -----------
```

```
   * DRQ for disc requests                      |          |
                                                 -----------
```

DEVICE REFERENCE TABLE (DRT)

(SERIES II/III)

```
|--------------------------------------------|
|                   SIOP                     |
|--------------------------------------------|
|                    PI                      |
|--------------------------------------------|
|                    DBI                     |
|--------------------------------------------|
|                 RESERVED                   |
|--------------------------------------------|
```

SIOP - absolute address of SIO program
PI   - interrupt handler plabel
DBI - this is the absolute address of the ILT

```
                 (/33, /44)
   ABS   ---------------------------
     8   |       Bank of DRT       |
         |-------------------------|    >----|
     9   |  Offset of DRT in Bank  |         |
         ---------------------------         |
                                             |
                                             |
            DRT ENTRY ON /33, /44            |
         ---------------------------         |
         |        SIOP             |    <----|
         |-------------------------|
         |        DBI              |
         |-------------------------|
         |        PI               |
         |-------------------------|
         |        Channel Flags    |
         ---------------------------
```

DRIVER LINKAGE TABLE (DLT)

```
   0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
 |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 |                       |DF|MC|CR|     |0 |MTYP |
0|     QUEUE NUMBER      |   (SEE BELOW)        |     DPROC
 |----------------------------------------------|
1|               MONITOR PLABEL                 |     DMNTR
 |----------------------------------------------|
2|              INITIATOR PLABEL                |     DINIT
 |----------------------------------------------|
3|              COMPLETOR PLABEL                |     DCOMP
 |----------------------------------------------|
4|              INTERRUPT PLABEL                |     DINTP
 |----------------------------------------------|
5|        DIT SIZE       |      DEVICE TYPE     |     DTYPE
 |----------------------------------------------|
6|          CS DRIVER EDITOR PLABEL             |
 |----------------------------------------------|
7|            INITIALIZATION PLABEL             |
 |----------------------------------------------|
```

There is one DLT for each type of driver. A pointer in the DIT allows different devices on a controller to have different drivers and interrupt handlers.

DPROC.QNUMB - This field contains the I/O process request queue number for type 2 drivers. Zero for all other types.

.(8:1).DRVRFRZN (DF) - Driver code frozen. Set by MAM when then the driver code segment has been made present and frozen from a request from SIODM.

.(9:1).MAMERRORC (MC) - MAM Error on Code Makepresent

.(10:1).CORERES (CR) - If set both initiator and completor code are core resident.

.(14:2).DRVRTYPE (MTVP) - DRIVER/MONITOR TYPE

- 0 - not used
- 1 - driver can be executed on any stack
- 2 - driver can be executed in the user process or in the I/O process identified by IDNUMB
- 3 - run only in process whose PCB number is in IDNUMB

DMNTR - I/O Monitor Plabel.

DINIT - Driver Initiator Procedure Plabel.

DCOMP - Driver Completor Procedure Plabel.

DINTP - Special interrupt hanler Plabel. This procedure is called by GIP if ISPEC is set DFLAG. No other action is taken by GIP except to set the Interupt Status in DSTAT.

DTYPE.DITSIZE - The length of the DIT in words for this driver.

LOGICAL-PHYSICAL DEVICE TABLE (LPDT)

The system uses the Logical-Physical Device Table (LPDT) for many purposes. For every physical device on the system, there is an entry which is used to communicate to the system the various states the device may be in. Included in the entry is the DRSTATE (device recognition state) used by DEVREC and the I/O drivers for the handling of unexpected interrupts (e.g., a tape mount, a carriage return on an available terminal). Also in the LPDT is an entry for every open spoolfile, which allows a large part of the operating system to treat open spoolfiles in much the same way as physical devices are treated.

Much of the low-level operating system software accesses the LPDT. Specifically, DEVREC (the device recognition system process) and many I/O drivers modify both the LPDT header and the DRSTATE for specific devices. Although there is an LPDT SIR, these low-level modules don't use it: we do not wish an I/O driver to impede while waiting for a SIR. Thus, whenever either the LPDT header or the second word of an LPDT entry are modified, the modifying software first DISABLEs in order to lock the LPDT. Software that accesses the spooling information in the LPDT typically uses the SIR mechanism to lock the table.

Although it would seem that SIR locking is the proper method for locking the LPDT, not all software that modifies the LPDT uses it. As a result, improper LPDT locking could lead to incorrect information in the LPDT header. Included in the header is the service request count for DEVREC. Occasionally, this count is decremented once too often; thus, the highest numbered logical real device requesting DEVREC service will never get serviced.

In summary, if you are modifying the LPDT for a real device, you must first DISABLE, do all your modifications, and then ENABLE. This is also the case if you want to be sure of the LPDT data you are reading.

If you are modifying the LPDT for a virtual device (an open spoolfile), you must lock the LPDT SIR before the table can be safely modified.

It is easy to determine that the LPDT was improperly locked after the fact, but it is impossible to determine who was improperly locking the table. To avoid improper LPDT locking, it is imperative that you eye-check all code for improper locking.

LPDTDST=%15
LPDTSIR=%11

```
                  LDEV            =2
|-----------------/---------------|
| HIGH ENTRY # |   ENTRY SIZE     |
|---------------------------------|
|        SERV. REQ INT            |
|---------------------------------|
|                                 |


|                                 |
|---------------------------------| \
|0|                               | |
|---------------------------------| | NORMAL DEVICE ENTRY
|       -- same --                | |
|---------------------------------| /
|                                 |


|                                 |
|---------------------------------| \
| |I|                             | |
|1|0|    XDD INDEX                | |    VIRTUAL DEVICE ENTRY-ASSIGNED
|---------------------------------| |      IO = 0  IDD
|       -- same --                | |         = 1  ODD
|---------------------------------| /
|                                 |
|                                 |


|                                 |
|                                 |
|---------------------------------| \
|1|       0                       | |    VIRTUAL DEVICE FREE ENTRY
|---------------------------------| |
|       -- same --                | |
|---------------------------------| /
|                                 |
|                                 |
```

LPDT (CONT.)

```
                 LPDT ENTRY
                 ----------
    0    1    2   3   4    5    6    7    8    9  10 11 12 13 14 15
   ------------------------------------------------------------------
 0 | V  |          DITP/VIRTUAL DEVICE INFORMATION                   |
   |FLAG|                                                            |
   |------------------------------------------------------------------
   |        |    |    | CY|   |    |          |BR|LG|              |          |
   |        |    |    |---|DUP|INTR|          |--|--|              |
 1 |DRSTATE |JOBS|DATA|BOT|   |    |   EOF    |  |DR|   SUBTYPE    |
   |        |    |    |---|---|----|          |--|--|              |
   |        |    |    |NSD| M | RV |          |SF|FS|              |          |
   ------------------------------------------------------------------
```

There is one two-word entry in the LPDT for each Logical Device.

The base of the entry for a given Logical Device is equal to the Logical Device number multiplied by the entry size (word 0.(8:8)), currently two. The physical device characteristics are maintained in the DIT and ILT.

The field definitions for each entry are:

WORD 0 --

```
   VFLAG    - Virtual device flag
   DITP     - When VFLAG = 0, SYSDB relative pointer to the DIT
                           1, Virtual device information
```

WORD 1 --

The following fields are defined for all devices:

```
   DRSTATE  - Device Recognition State
              0-Not owned
              1-Owned or recognized
              2-Service requested - set by driver upon unexpected
                                    interrupt and awake DEVREC
              3-Service granted - set by DEVREC
                (sequence for logon:0-2-3-1)
   JOBS     - Accepting Jobs or Sessions
   DATA     - Accepting Data
   EOF      - End of File condition
              0-No EOF
              1-HARDWARE EOF
              2-:DATA
              3-:EOD
              4-:HELLO
              5-:BYE
              6-:JOB
              7-:EOJ
```

LPDT (CONT.)

SUBTYPE - Device subtype. For tapes, the SUBTYPE is divided into two subfields as follows:

WORD1.(13:3) - actual device subtype
WORD1.(12:1) - 0 = operator allocation
1 = automatic allocation

The definitions for bits 4,5,6,10, and 11 in word 1 are device dependent.

For terminal-like devices only,
CY - Control Y is allowed and has been detected
BR - Break detected or ignore break if main running
LG - The terminal is logging on. This bit is set by PROGEN and DEVREC when the logon sequence starts. If the bit is off when polled by INITJSMP, the terminal has disconnected. For now, only IOTERM0 and HIOTERM0 support the use of this bit. MULTIPOINT and DS pseudo-terminals do not.

For tape drives only,
BOT - Tape is at load point or no tape mounted
DR - DEVREC is performing Automatic Volume Recognition (AVR) on tape drive or suppress AVR on job/data-accepting tapes

For all devices except disc drives,
DUP - Duplicative
INTR - Interactive

For disc drives only,
NSD - The disc is a non-system domain disc drive

For non-system domain disc drives (NSD=1) only,
M - Mounted private volume
RV - Reserved volume for multiple pack mount requirement
SF - Serial or foreign disc physically and logically mounted
FS - If SF = 1, then: FS = 0, Serial disc
FS = 1, Foreign disc

LPDT (CONT.)

```
   0
 |-----------------------------------------------|
 |  |   V=0 then DITPOINTER                      |
0| V|   V=1 then Virtual Device Entry Info.      |
 |-----------------------------------------------|
 |                                               |
1|                 -- as before --               |
 |-----------------------------------------------|
```

The first word of each entry in the LPDT has changed to reflect the addition of Virtual Devices.

A "real" logical device (ie. one on which an ATTACHIO call may be performed) has the sign bit set to "zero".

A "virtual" logical device has the sign bit set to "one". Thus anyone who loads the DIT pointer for use must check this sign bit.

OVERVIEW OF DEVICE TABLES IN DST %16

```
|---------------------------|<-----DST %16
|                           |
|   LOGICAL DEVICE TABLE    |
|                           |
|           LDT             |
|                           |
|---------------------------|
|                           |
|                           |
|     DEVICE CLASS TABLE    |
|                           |
|                           |
|---------------------------|
|                           |
|   LOGICAL DEVICE TABLE    |
|        EXTENSION          |
|          LDTX             |
|                           |
|---------------------------|
```

LOGICAL DEVICE TABLE

(Indexed by Log Dev#)

DST 16(8) = 14(10)
SIR 12(8) = 10(10)

ZERO ENTRY FORMAT

```
   0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
  |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 0|   HIGHEST ENTRY #     |     ENTRY SIZE=5       |
  |------------------------------------------------|
  |      POINTER TO FIRST DEVICE CLASS ENTRY       |
 1|           (RELATIVE TO TABLE BASE)             |
  |------------------------------------------------|
 2|        NUMBER OF DEVICE CLASS ENTRIES          |
  |------------------------------------------------|
 3|           SIZE OF DEVICE CLASS TABLE           |
  |------------------------------------------------|
 4|////////////////////////| STREAMS DEVICE NUMBER |
  |------------------------------------------------|
```

TYPICAL ENTRY FORMAT

```
   0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
  |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  |                FILE USE COUNT                  |0
  |------------------------------------------------|
  |VOL TABLE INDEX IF DEV |                        |  *or process # of
  |  TYPE<8 OTHERWISE   * |      CONTROL Y PIN     |1  I/O spooler for
  | MAIN PROCESS PIN #    |                        |   this device
  |------------------------------------------------|
  |     RECORD WIDTH      |CS|FO|  DEVICE TYPE     |2
  |------------------------------------------------|
  |     |  |  |  |  |  |  | DEFAULT OUTPUT DEVICE  |
  |  SS | F| M| R|  |HT| C|  OR CLASS INDEX(C=1)  |3
  |------------------------------------------------|
  |                    |S |                        |
  |       MISC         | Q|       VDD INDEX        |4
  |------------------------------------------------|
```

LDT (CONT.)

```
SS. . . spool state
        0 not spooled           reserved
        1 spooled input           for
        2 spooled output        spooling
SQ = 1  SPOOLING ENABLED
C . . . default device is class index     CS . . . CS device
F . . . avail to system                   FO . . . Special Forms
M . . . avail to diagnostics              HT . . . 0 = Header/Trailer on
R . . . down requested                             1 = Header/Trailer off
MISC. . . miscellaneous information, device dependent:

   1) For terminal-like devices, default terminal type to be used when
      not specified in HELLO command.
   2) For variable density tape drives, contains density information.
      WORD4.(1:3) -- actual tape density
                     0 = density not yet determined
                     1 = 1600 BPI
                     2 = 6250 BPI
      WORD4.(4:3) -- density requested in FOPEN for writes to tape,
                     unlabelled tapes only
                     0 = no FOPEN with write access yet
                     1 = 1600 BPI
                     2 = 6250 BPI
```

DEVICE CLASS TABLE

(Sequentially Organized)

TYPICAL ENTRY FORMAT

```
             0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
            |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
           0|                                               |
            |                                               |
           1|                                               |
            |                  CLASS NAME                   |
           2|                                               |
            |                                               |   T=TERMINAL
           3|                                               |     ACCESS
            |-----------------------------------------------|     BIT
           4|//|  CYCLICAL POINTER  |SQ| T|CLASS ACCESS TYPE|
            |-----------------------------------------------|
           5|#OF DEVICES IN CLASS(N)|        DEVICE #1      |
            |-----------------------------------------------|
           6|       DEVICE #2        |        DEVICE #3      |
            |-----------------------------------------------|
                       .                        .
                       .                        .
                       .                        .
                       .                        .
                       .                        .
                       .                        .
                       .                        .
                       .                        .
            |-----------------------------------------------|
N/2+6-->    |       DEVICE #N-1      |        DEVICE #N      |
            |-----------------------------------------------|

           SQ = Spool Queue bit
```

5

Same as devType, except

Serial disc = 4

Foreign disc =

NOTE: The device class table is in the same data segment (DST 16(8) as the LDT. ie., the LDT consists of three separate tables.

1. logical device table and
2. device class table
3. LDT Extension

Logical Device Table Extension (LDTX)

DST %16 = #14
SIR %12 = #10

Zero Entry

```
       0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   0 |    Highest Entry #     |       Entry Size      |
     |-----------------------------------------------|
   1 |                                               |
     |-----------------------------------------------|
   2 |                                               |
     |-----------------------------------------------|
   3 |                                               |
     |-----------------------------------------------|
   4 |                                               |
     +-----------------------------------------------+
```

Typical Entry

```
       0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   0 | S|SD|CP|NS|  reserved   |DB|      TBRC         |
     |-----------------------------------------------|
   1 |              device specific                  |
     |-------                                 -------|
   2 |            information fields                 |
     |-------                                 -------|
   3 |          See the following examples           |
     |-------                                 -------|
   4 |             LDTX descriptor                   |
     +-----------------------------------------------+
```

Legend for all entries:

S......Seek ahead enable/disable flag (system or PV disc only).
SD.....This logical device is a Serial Disc.
CP.....This logical device uses the CIPER protocol.
NS.....This is a non-shareable (system or PV) disc device.
DB.....If set to 1, then debugging in effect (CIPER calls DEBUG)
TBRC...Terminal's baud rate code.

Logical Device Table Extension (LDTX)

Terminal Entry

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
   +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 0 | 0| 0| 0| 0|  reserved    |  |       TBRC      |
   |-----------------------------------------------|
 1 |                      0                        |
   |-----------------------------------------------|
 2 |TB|            reserved for ATP                |
   |-----------------------------------------------|
 3 |                      0                        |
   |-----------------------------------------------|
 4 |                      0                        |
   +-----------------------------------------------+
```

TB.....used only on Series 3X, 4X, 6X
       1 = terminal connected to ATP
       0 = terminal connected to ADCC
TBRC...Terminal's baud rate code.

Series III (ATC)

TBRC	chars/second
1	240
2	120
3	60
4	30
5	15
6	10
7	14

Series 3X, 4X, 6X (ATP or ADCC)

TBRC		chars/second
%6	6	60
%7	7	240
%10	8	960
%11	9	480
%12	10	unused
%13	11	120
%14	12	unused
%15	13	30
%16	14	15
%17	15	10
%20	16	1920
%21	17	3840
%22	18	180

Serial Disc Entry

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
  +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
0 |  | 1| 0| 0|  reserved    |  |        0        |
  |-----------------------------------------------|
1 |       Serial disc extra data segment #        |
  |-----------------------------------------------|
2 |                      0                        |
  |-----------------------------------------------|
3 |                      0                        |
  |-----------------------------------------------|
4 |                      0                        |
  +-----------------------------------------------+
```

Logical Device Table Extension (LDTX)

CIPER Entry

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
   +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 0 | 0| 0| 1| 0|  reserved    |DB|       0         |
   |-----------------------------------------------|
 1 |  CIPER Device Control Data Segment # (CDCDS)  |
   |-----------------------------------------------|
 2 |DN|  CTM Index for this device (CTMI)          |
   |-----------------------------------------------|
 3 |                     0                         |
   |-----------------------------------------------|
 4 |                     0                         |
   +-----------------------------------------------+
```

DB.....If set to 1, then debugging in effect

DN.....If 1, the CIPER facility has been de-activated for this device because of error.

CTMI...Control Table Map Index (an index into the Control Table Map (CTM) which is located in the CIPER Data Segment (CDCDS)

System or Private Vol. Disc Entry

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
   +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 0 | S| 0| 0| 1|  reserved    |  |        0        |
   |-----------------------------------------------|
 1 |                     0                         |
   |-----------------------------------------------|
 2 |Disc Free Space DST number (DFSDST)            |
   |-----------------------------------------------|
 3 |Disc Free Space error status (DFSERR)          |
   |-----------------------------------------------|
 4 |                     0                         |
   +-----------------------------------------------+
```

S......Seek ahead enable/disable flag.

INTERRUPT LINKAGE TABLE (ILT)

ILT FOR SERIES II/III

```
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
0   |                      0                        |   ICPVA0
    +-----------------------------------------------+
1   |                      0                        |   ICPVA01
    +-----------------------------------------------+
2   |                      0                        |   ICPVA02
    +-----------------------------------------------+
3   |                      0                        |   ICPVA03
    +-----------------------------------------------+
4   |                      0                        |   ICPVA04
    +-----------------------------------------------+
5   |                      0                        |   ICPVA05
    +-----------------------------------------------+
6   |                      0                        |   ISRQL/ICPGM
    +-----------------------------------------------+
7   | M|     CHANQUE     |        |   DRT NUMBER    |   IDRTN
    +-----------------------------------------------+
%10 |SYSDB relative pointer to I/O program area.    |   ISIOP
    +-----------------------------------------------+
%11 |                      0                        |   ISTAP
    +-----------------------------------------------+
%12 |single instruction that is executed to extract |   IUNIT
    |the device unit number from the status.        |
    +-----------------------------------------------+
%13 |                      0                        |   ICDP
    +-----------------------------------------------+
%14 |        SIOPSIZE       |         CQUEN         |   IQUEUE
    +-----------------------------------------------+
%15 |  |  |  |  |          0                        |   IFLAG
    +-----------------------------------------------+
%16 | SYSDB relative DIT pointer for unit 0         |   IDITP0
    +-----------------------------------------------+
                          .
                          .
                          .
    +-----------------------------------------------+
    |SYSDB relative DIT pointer for unit n          |   IDITPN
    +-----------------------------------------------+
    |            Seekmask  (Disc only)              |
    +-----------------------------------------------+
    |              I/O                              |
    |               Program                         |
    |                Area                           |
    +-----------------------------------------------+
SIOPSIZE - SIO PROGRAM SIZE / 2.
```

ILT FOR SERIES 30/33/44 & SERIES II/III (HP-IB)

```
           -----------------------------------------

            0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
          +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 0        |         Channel                              |   ICPVA0
 1        |          Program                             |   ICPVA01
 2        |           Variable                           |   ICPVA02
 3        |            Area   (ICPVA)                    |   ICPVA03
          +----------------------------------------------+
 4        |         DMA Abort                            |   ICPVA04
 5        |          Address                             |   ICPVA05
          +----------------------------------------------+
 6        |                     0                        |   ISRQL/ICPGM
          +----------------------------------------------+
 7        | M|     CHANQUE     |     |    CHAN    | DEV  |   ICNTRL
          +----------------------------------------------+
%10       |SYSDB relative pointer to channel program area.|   ISIOP
          +----------------------------------------------+
%11       |SYSDB relative pointer to status return area. |   ISTAP
          +----------------------------------------------+
%12       |single instruction that is executed to extract |   IUNIT
          |the device unit number from the status pointed |
          |to by ISTAP.                                   |
          +----------------------------------------------+
%13       |SYSDB relative DIT pointer of the device       |   ICDP
          |currently using the channel to perform a data  |
          |operation.                                     |
          +----------------------------------------------+
%14       |        SIOPSIZE        |        CQUEN        |   IQUEUE
          +----------------------------------------------+
%15       |RW|WP|IG|SC|SQ|                    |  HCUNIT  |   IFLAG
          +----------------------------------------------+
%16       | SYSDB relative DIT pointer for unit 0         |   IDITP0
          +----------------------------------------------+
                                .
                                .
                                .
          +----------------------------------------------+
          |SYSDB relative DIT pointer for unit n          |   IDITPN
          +----------------------------------------------+
          |        Program status return area             |
          |         pointed to by ISTAP                   |
          +----------------------------------------------+
          |            Seekmask  (Disc only)              |
          +----------------------------------------------+
          |              I/O                              |
          |               Program                         |
          |                Area                           |
          +----------------------------------------------+
```

ILT TERMINOLOGY

IPCVA - These four words comprise the channel program variable area where information is stored concerning a channel program Interrupt instruction or abort. CPVA0 should be used only for channel program aborts.

ICPVA4 - Words 4 and 5 contain DMA address, when channel program aborts during DMA transfer.

ISRQL - Serial poll request queue length. Series 33 currently does not support any serial poll devices. This should always be zero.

ICPGM - This is the SYSDB relative address of the channel program to be started for this device after receiving a HIOP interrupt in GIP. GIP will call STARTIO when the flags word indicates "ignore halt interrupt" and "start channel program" bits are set.

ICNTRL - Contains controller information.

  .M If set, the controller is sharing a software channel resource in order to limit bandwidth.

  .CHNQ The software channel resource number.

  .DRTN The DRT number for a Series 33 device is equivalent to:
    .CHAN - channel number (4 most significant bits of DRTN)
    .DEV - device number (3 least significant bits of DRTN)

IFLAG - Used for controller flags.

  .RW Runwait flag. An idle channel program should be started when there are no active requests to process.

  .WP Waitprog flag. An idle channel program has been started for this controller. This bit is reset by an interrupt.

  .IG Ignorehi flag. An HIOP instruction has been issued against this controller, but the channel program was not in a wait statement. Therefore, ignore the interrupt generated by the channel code when this program halts.

  .SC Start channel program flag. When set along with the IG flag, GIP will start a previously attempted SIOP on this device.

  .SQ Start channel program "queued" flag. When bit SC is set, this bit will determine if the call to START'HPIB will have logical parameter QUEUED true or false.

  .HCUNIT Highest configured unit number for this controller.

DEVICE INFORMATION TABLE (DIT)

There is one DIT per physical device. If a physical device represents represents more than one logical device, the logical device number is obtained from the I/O queue element.
Although details of DIT's vary with device, the following structure is common to all:

```
                DIT  for Series II/III

   0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
  +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
0 |T |D |AC|RQ|CE|MU|SP|IO|IA|NO|ST|NS|  STATE    |   DFLAG
  +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
1 |SYSDB relative pointer to the DIT for the next |   DLINK
  |   device requesting this resource or service  |
  +-----------------------------------------------+
2 |SYSDB relative pointer to the first IOQ in     |   DIOQP
  | request list for this device                  |
  +-----------------------------------------------+
3 | IOT |  Phys. unit #   | Logical device number |   DLDEV
  +-----------------------------------------------+
4 |SYSDB relative pointer to Driver Linkage Table |   DDLTP
  +-----------------------------------------------+
5 |SYSDB relative pntr to Interrupt Linkage Table |   DILTP
  +-----------------------------------------------+
6 |Controller hardware status                     |   DSTAT
  +-----------------------------------------------+
7 |Hardware error status.  Set when the driver    |   DSERR
  | detects an error.  Whenever <>0, the driver   |
  | monitor logs an I/O error and clears this word|
  +-----------------------------------------------+
  |        Device Dependent Area                  |   (DTIME)
  +-----------------------------------------------+
```

DIT TERMINOLOGY (SERIES II/III)

DFLAG - DEVICE RELATIVE FLAGS

T	SET IF DEVICE IS A TERMINAL.
D	SET IF DEVICE IS A DISC.
AC	ACTIVE BIT. 1 IMPLIES A MONITOR CURRENTLY SERVICING THIS DEVICE.
RQ	REQUEST BIT. 1 IMPLIES SERVICE REQUESTED WHILE MONITOR IS ACTIVE.
MU	IF SET, MULTIPLE UNIT CONTROLLER.
IO	IF SET, THEN A CHANNEL PROGRAM IS CURRENTLY EXECUTING.
IA	IF SET, AN INTERRUPT OR RESPONSE HAS OCCURRED.
NO	IF SET, DEVICE IS IN A NOT READY OR OPERATOR WAIT.
CE	CACHING ENABLED ON THIS DEVICE (MASS STORAGE ONLY)
SP	SIO PREEMPTION
ST	START WAIT CHANNEL PROGRAM
NS	DO NOT SHORT WAIT THIS DISC
STATE	CURRENT DRIVER STATE AS DEFINED BY THE MONITOR. ALLOWABLE STATES ARE: 0 - START REQUEST 1 - NOT USED (BUT RESERVED) 2 - CALL DRIVER INITIATOR 3 - CALL DRIVER COMPLETOR 4 - NOT USED (BUT RESERVED) 5 - COMPLETE REQUEST 6 - UNEXPECTED INTERRUPT OCCURED 7 - START OPERATOR INTERVENTION WAIT %10 - WAITING (ON OPERATOR). RESTART AT 0 %11 - WAITING (DATA MAKEPRESENT/FREEZING) %12 - WAITING (INITIATOR CODE MAKEPRESENT/FREEZE) %13 - WAITING (FOR COMPLETION INTERRUPT) %14 - WAITING (FOR DEVICE CONTROLLER AVAILABILITY) %15 - NOT USED (BUT RESERVED) %16 - WAITING (INITIATOR CODE MAKEPRESENT) %17 - WAITING (COMPLETOR CODE MAKEPRESENT)

IOT - I/O System type	0-Series II/III I/O System 1-HP-IB 2-unused 3-unused

DIT FOR SERIES 30/33/44

```
    0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
   +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 0 |T |D |AC|RQ|CE|MU| 0|IO|IA|NO|ST|NS|  STATE    |   DFLAG
   +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 1 |SYSDB relative pointer to the DIT for the next |   DLINK
   |   device requesting this resource or service  |
   +-----------------------------------------------+
 2 |SYSDB relative pointer to the first IOQ in     |   DIOQP
   | request list for this device                  |
   +-----------------------------------------------+
 3 | IOT |  Phys. unit #   | Logical device number |   DLDEV
   +-----------------------------------------------+
 4 |SYSDB relative pointer to Driver Linkage Table |   DDLTP
   +-----------------------------------------------+
 5 |SYSDB relative pntr to Interrupt Linkage Table |   DILTP
   +-----------------------------------------------+
 6 |      Controller Hardware Status               |   DSTAT
   +-----------------------------------------------+
 7 |Hardware error status.  Set when the driver    |   DSERR
   | detects an error.  Whenever <>0, the driver   |
   | monitor logs an I/O error and clears this word|
   +-----------------------------------------------+
   |        Device Dependent Area                  |   (DTRQX)
   +-----------------------------------------------+
```

DTRQX	Used by some device drivers, it denotes timer request index.

DIT TERMINOLOGY (SERIES 30/33)

DFLAG - DEVICE RELATIVE FLAGS

T	SET IF DEVICE IS A TERMINAL.
D	SET IF DEVICE IS A DISC.
AC	ACTIVE BIT. 1 IMPLIES A MONITOR CURRENTLY SERVICING THIS DEVICE.
RQ	REQUEST BIT. 1 IMPLIES SERVICE REQUESTED WHILE MONITOR IS ACTIVE.
MU	IF SET, MULTIPLE UNIT CONTROLLER.
IO	IF SET, THEN A CHANNEL PROGRAM IS CURRENTLY EXECUTING.
IA	IF SET, AN INTERRUPT OR RESPONSE HAS OCCURRED.
NO	IF SET, DEVICE IS IN A NOT READY OR OPERATOR WAIT.
ST	IF SET, AN IDLE CHANNEL PROGRAM SHOULD BE STARTED FOR THIS DEVICE.
CE	CACHING ENABLED ON THIS DEVICE (MASS STORAGE ONLY)
NS	DO NOT SHORT WAIT THIS DISC
STATE	CURRENT DRIVER STATE AS DEFINED BY THE MONITOR. ALLOWABLE STATES ARE: 0 - START REQUEST 1 - NOT USED (BUT RESERVED) 2 - CALL DRIVER INITIATOR 3 - CALL DRIVER COMPLETOR 4 - NOT USED (BUT RESERVED) 5 - COMPLETE REQUEST 6 - UNEXPECTED INTERRUPT OCCURED 7 - START OPERATOR INTERVENTION WAIT %10 - WAITING (ON OPERATOR). RESTART AT 0 %11 - WAITING (DATA MAKEPRESENT/FREEZING) %12 - WAITING (INITIATOR CODE MAKEPRESENT/FREEZE) %13 - WAITING (FOR COMPLETION INTERRUPT) %14 - WAITING (FOR DEVICE CONTROLLER AVAILABILITY) %15 - NOT USED (BUT RESERVED) %16 - WAITING (INITIATOR CODE MAKEPRESENT) %17 - WAITING (COMPLETOR CODE MAKEPRESENT)

IOT - I/O System type 0-Series II/III I/O System
1-HP-IB
2-unused
3-unused

DIT for SIO Devices

```
    0    1    2   3   4    5    6     7    8    9  10 11 12 13 14 15
 |-----------------------------------------------------------------|
0|TERM|DISC|ACT|REQ|CE  |  M  | SIO  | IO |IAK|  M  |NT|   |   STATE   |DFLAG
 |    |    |   |   |    |UNIT| PREMP|PROG|   |HEAD|RY|   |           |
 |-----------------------------------------------------------------|
1|                          NEXT DITP                              |DLINK
 |-----------------------------------------------------------------|
2|                            IOQP                                 |DIOQP
 |-----------------------------------------------------------------|
3| IOT     |    UNIT                    |          LDEVN            |DLDEV
 |-----------------------------------------------------------------|
4|                            DLTP                                 |DLTP
 |-----------------------------------------------------------------|
5|                            ILTP                                 |DILTP
 |-----------------------------------------------------------------|
6|                 Controller Hardware Status                      |DSTAT
 |-----------------------------------------------------------------|
7|                   Hardware Error Status                         |DSERR
 |-----------------------------------------------------------------|
8|                                                                 |DTRQX
 |-----------------------------------------------------------------|
 |                                                                 |


 |                  DRIVER DEPENDENT DIT AREA                      |
 |                                                                 |
 |                                                                 |
 |                                                                 |
  -----------------------------------------------------------------
```

```
DFLAG.TERMINAL  - Device is a terminal
     .DISC      - Device is a Disc (Bit 0 = 0)
     .ACTIVE    - A monitor is currently servicing this device
     .REQUEST   - Service requested while monitor was active

     .MUNIT     - device controller servicing multiple units
     .SIOPREMPT- If set then a preemptive request has been queued for
                  this device.  Preempt code is set in IOQ.
     .IOPROG    - I/O program in progress.  Decrement SIOCOUNT and
                  check for multi-channel when complete
     .IAK       - Interrupt or Response has occurred.
     .M HEAD    -Moving head disc
     .NT RDY    -Not ready for SIO. SIODM holds off next SIO until
                 ALLOWPOLL is done.
     .CE        - Caching is enabled on this device (mass storage only)

DTRQX        - Used by some device drivers, it denotes timer
               request index.
```

DIT FOR SIO DEVICES (CONT.)

DFLAG.STATE - this quantity specifies the next action to be taken in servicing the request.

0-new - start request.
1-not used.
2-call Driver Initiator Procedure
3-call Driver Completor Procedure
5-complete request
6-device recognition
7-start operator intervention wait (%10)
%10-restart request on interrupt
%11-wait for data to be frozen then state 2
%12-wait for driver code to be frozen then state 2
%13-call completor on interrupt
%14-wait for device controller
%15-not used
%16-wait for initiator make present then state 2
%17-wait for completor make present then state 3

DLINK - SYSDB relative pointer to the DIT for the next device requesting this resource or service.
DIOQP - SYSDB relative pointer to the first IOQ in the request list for this device
DLDEV.LDEVN - Logical Device Number
.UNIT - unit number of the physical device.
.IOT - IO type 0=> Series III I/O, 1=> HPIB I/O
DDLTP - SYSDB relative pointer to the DLT.
DILTP - SYSDB relative pointer to the ILT.
DSTAT - interrupt status for this device. Set each time the device interrupts.
DSERR - Hardware Device Controller Status. Set when the driver detects an error. whenever not zero SIODB logges an I/O error and clears this word.
DTIME - time out completed flags. If a timeout occurs in response to a timer request type %20 (I/O request), the sign bit is set in this word. The IA bit in DFLAG is also set, and the monitor for this device is awakened. (Only used if timer services are requested. Must be word #8 if timer services are requested.)

DIT FOR FIXED HEAD DISK

```
    0  1  2   3   4   5  6   7  8   9  10 11 12      15
  |--|--|---|---|---|--|--|---|---|---|--|--|----------|
  | 0| 1|ACT|REQ|CE | 0| 0|I/O|IAK| 0 | 0| 0|  STATE   |    DFLAG
  |----------------------------------------------------|
 1|                     NEXT DITP                      |    DLINK
  |----------------------------------------------------|
 2|           CURRENT REQUEST SYSBASE INDEX            |    DCURRREQP
  |----------------------------------------------------|
 3| IOT |                     |           LDEVN        |    DLDEV
  |----------------------------------------------------|
 4|                        DLTP                        |    DDLTP
  |----------------------------------------------------|
 5|                        ILTP                        |    DILTP
  |----------------------------------------------------|
 6|                    DEVICE STATUS                   |    DSTAT
  |----------------------------------------------------|
 7|                DEVICE STATUS (ERROR)               |    DSERR
  |----------------------------------------------------|
 8| SYSBASE INDEX OF FIRST REQUEST IN QUEUE            |    DQHEAD
  |----------------------------------------------------|
 9| SYSBASE INDEX OF LAST REQUEST IN QUEUE             |    DQTAIL
  |----------------------------------------------------|
10|                     XFER COUNT                     |    DXFER
  |----------------------------------------------------|
11|                  LOGICAL DISK ADDR                 |    DDADR
  |----------------------------------------------------|
12|                   SYSBUF ADDRESS                   |    DSYSBA
  |----------------------------------------------------|
  |             ERROR & RETRY INFORMATION              |
  |--|--|-----------------------------------|----------|
  |  |  |                                   |  RETRY   |    QMISC
  | B| W|                                   |  COUNT   |    OF IOQ
  |--|--|-----------------------------------|----------|
```

IOT - I/O Devices
      0 - Series II/III
      1 - HP-IB
      3 - unused
      4 - unused

B - modify bad track table
W - write bad track table

DIT FOR 7900A & 2888A MOVING HEAD DISC

```
     0  1  2   3   4   5    6   7    8   9  10 11 12       15
    |--|--|---|---|---|-----|--|----|---|---|--|--|----------|
    | 0| 1|ACT|REQ|CE |  M  | 0| I/O|IAK| 1 | 0| 0|  STATE   |    DFLAG            |
    |  |  |   |   |   |UNIT |  |PROG|   |   |  |  |          |
    |--------------------------------------------------------|
   1|                       NEXT DITP                        |    DLINK
    |--------------------------------------------------------|
   2|             CURRENT REQUEST SYSBASE INDEX              |    DCURRREQP
    |--------------------------------------------------------|
   3| IOT |          UNIT          |          LDEVN          |    DLDEV
    |--------------------------------------------------------|
   4|                          DLTP                          |    DDLTP
    |--------------------------------------------------------|
   5|                          ILTP                          |    DILTP
    |--------------------------------------------------------|
   6|                 CURRENT DEVICE STATUS                  |    DSTAT
    |--------------------------------------------------------|
   7|                  DEVICE ERROR STATUS                   |    DSERR
    |--------------------------------------------------------|
   8| SYSBASE INDEX OF FIRST REQUEST IN QUEUE                |    DQHEAD
    |--------------------------------------------------------|
   9| SYSBASE INDEX OF LAST REQUEST IN QUEUE                 |    DQTAIL
    |--------------------------------------------------------|
  10|                      CURRENT DISC                      |
  11|                        ADDRESS                         |     DADR
    |--------------------------------------------------------|
  12|                    ALTERNATE TRACK                     |
  13|                     DISC ADDRESS                       |    DALTADR
    |--------------------------------------------------------|
  14|                   CURRENT CYLINDER                     |    CURCYL
    |--------------------------------------------------------|
  15|              CURRENT DATA BUFFER ADDRESS               |    DBUFF
    |--------------------------------------------------------|
  16|               NEXT DATA BUFFER ADDRESS                 |    DNXTBUFF
    |--------------------------------------------------------|
  17|                 WORD COUNT REMAINING                   |    WCR
    |--------------------------------------------------------|
  18|                  CURRENT WORD COUNT                    |    CWC
    |--------------------------------------------------------|
  19|                    SYSBUF ADDRESS                      |    DSYSBA
    |--------------------------------------------------------|
```

```
IOT - I/O Devices
      0 - Series II/III
      1 - HP-IB
      3 - unused
      4 - unused
```

```
                     ERROR & RETRY INFORMATION
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |           |  RETRY    |
| M| R| W| T| A| X| C| S| 0  0  0  0|  COUNT    |     QMISC OF IOQ
|--|--|--|--|--|--|--|--|-----------|-----------|
```

M - handling defective track map
R - read defective track map
W - write defective track map
T - track to track xfer

A - reading alternate track
X - xfer from alt. track
C - recalibration done
S - seek or recal in progress

DIT FOR 7905/7906/7920/7925

```
     0  1  2   3   4   5     6   7    8   9  10 11 12       15
   |--|--|---|---|---|-----|--|----|---|---|--|--|----------|
  0| 0| 1|ACT|REQ|CE | M   | 0| I/O|IAK| 1 | 0| 0|  STATE   | 0  DFLAG        |
   |  |  |   |   |   |UNIT |  |PROG|   |   |  |  |          |
   |--------------------------------------------------------|
  1|                      NEXT DITP                         | 1  DLINK
   |--------------------------------------------------------|
  2|            CURRENT REQUEST SYSBASE INDEX               |    DCURRREQP
   |--------------------------------------------------------|
  3| IOT |          UNIT           |          LDEVN         | 3  DLDEV
   |--------------------------------------------------------|
  4|                         DLTP                           | 4  DDLTP
   |--------------------------------------------------------|
  5|                         ILTP                           | 5  DILTP
   |--------------------------------------------------------|
  6|                 CURRENT DEVICE STATUS                  | 6  DSTAT
   |--------------------------------------------------------|
  7|                  ERROR DEVICE STATUS                   | 7  DSERR
   |--------------------------------------------------------|
  8| SYSBASE INDEX OF FIRST REQUEST IN QUEUE                |    DQHEAD
   |--------------------------------------------------------|
  9| SYSBASE INDEX OF LAST REQUEST IN QUEUE                 |    DQTAIL
   |--------------------------------------------------------|
 10|                   CURRENT LOGICAL                      |12
 11|                    DISK ADDRESS                        |13  CLDA
   |--------------------------------------------------------|
 12|                  CURRENT PHYSICAL                      |14  CURCUL
 13|                    DISK ADDRESS                        |15   CPDA
   |--------------------------------------------------------|
 14|             CURRENT DATA BUFFER ADDRESS                |16  CDBA
   |--------------------------------------------------------|
 15|                 WORD COUNT REMAINING                   |17  WCR
   |--------------------------------------------------------|
 16|                  CURRENT WORD COUNT                    |20  CWC
   |--------------------------------------------------------|
 17|                   SYSBUF ADDRESS                       |21  SYSBUFA
   |--------------------------------------------------------|
 18|                   STATUS 1 RETURN                      |22  STAT1
   |--------------------------------------------------------|
 19|                   STATUS 2 RETURN                      |23  STAT2
   |--------------------------------------------------------|
 20|                                                        |24
   |                         CYL                            |    CEDA
 21|                                                        |25
   |--------------------------------------------------------|
 22|           HEAD              |          SECTOR          |26
   |--------------------------------------------------------|  \
 23|                   STATUS 1 RETURN                      |27|
   |--------------------------------------------------------|  |
 24|                         CYL                            |30|
   |--------------------------------------------------------|  |
```

DIT FOR 7905/7906/7920/7925 (CONT.)

```
  |----------------------------------------------------------|  |
25|             HEAD              |            SECTOR        |31|
  |----------------------------------------------------------|   REQUEST
26|                        DISPLACEMENT                      |32 SYNDROME
  |----------------------------------------------------------|  |
27|                           PATT 1                         |33|
  |----------------------------------------------------------|  |
28|                           PATT 2                         |34|
  |----------------------------------------------------------|  |
29|                           PATT 3                         |35|
  |----------------------------------------------------------|  /
30|                    SCOUNT (SECTOR COUNT)                 |36
  |----------------------------------------------------------|
31|                                                          |37
  |                  INITIALIZE ADDRESS                      |
32|                                                          |40
  |----------------------------------------------------------|
  |          POINTER TO THIS DIT'S STATTAB WORD              |41
  |----------------------------------------------------------|
```

IOT - I/O Devices
      0 - Series II/III
      1 - HP-IB
      3 - unused
      4 - unused

```
            ERROR & RETRY INFORMATION
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| D| S| E| M| W| 0| 0| C| 0  0  0  0  0  0  0  0|     QMISC OF IOQ
|--|--|--|--|--|--|--|--|-----------------------|
```

D - retry determination
S - request syndrome
E - request error info
M - update track map
W - writing track map
C - issued a recalibration

DIT for CS 80 Disc

There is one DIT per physical device. If a physical device represents more than one logical device, the logical device number is obtained from the IOQ element. For the CS'80 disc controller, there will only be one device. The following diagram shows the DIT used by the CS'80 disc driver.

NOTE: Integrated Cartridge Tape's DIT has the same format.

```
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15  MNEMONIC
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   0|TM|DS|AC|RQ|CE| 0| 0|IO|IA|NO|ST| 0|   STATE   | DFLAG
    +--+--+--+--+--+--+--+--+--+--+--+--+-----------+
   1| SYSDB relative pointer to the DIT for the next| DLINK
    | device requesting this resource or service    |
    +-----------------------------------------------+
   2| Current request sysbase index                 | DCURREQP
    +-----+-----------------+-----------------------+
   3|IOT  |     Phys. unit # | Logical device number | DLDEV
    +-----+-----------------+-----------------------+
   4| SYSDB relative pointer to Driver Linkage Table| DDLTP
    +-----------------------------------------------+
   5| SYSDB relative pointer to Intrp Linkage Table | DILTP
    +-----------------------------------------------+
   6| DSTAT is -1 when a system powerfail occurred  | DSTAT
    +-----------------------------------------------+
   7| Hardware error status.  Set when the driver   | DSERR
    | detects an error.  Whenever <>0, the driver   |
    | monitor logs an I/O error and clears this word|
    +-----------------------------------------------+
 %10| Sysbase index of first request in queue       | DQHEAD *
    +-----------------------------------------------+
 %11| Sysbase index of last request in queue        | DQTAIL *
    +--+--+-------------------------------+---------+
 %12|LK|IF|                               | SUBSTATE | DMISC
    +--+--+-------------------------------+---------+
 %13| SYSDB relative ptr to system buffer element   | DSBUFADDR
    +-----------------------------------------------+
 %14| High order logical sector address of bad blk  | DBADBLK1
    +-----------------------------------------------+
 %15| Low order logical sector address of bad blk   | DBADBLK2
    +-----------------------------------------------+
 %16| Byte transfer left when bad block occurred    | DBADXFER
    +-----------------------------------------------+
 %17| Hardware logged error status - CPVA (0)       | DLOGERROR
    +-----------------------------------------------+
 %20| Channel program aborted relative offset       | DSIOPSTOP
    +-----------------------------------------------+
 %21| Disc status (20 bytes)-Logged on status error | DSTATUS
    +-----------------------------------------------+
  . |                       .                       |
    +-----------------------------------------------+
  . |                       .                       |
    +-----------------------------------------------+
```

```
%32|                                                 |
   +-------------------------------------------------+
```

DFLAG - Flags and request state

TM TERM - Set if device is a terminal.
DS DISC - If TM = 0 and this bit is set then the device is a disc, otherwise device dependent.
CE - Caching is enabled.
AC ACTIVE - A monitor is currently servicing this device.
RQ REQUEST - A service request is pending while the monitor is active.
IO IOPROG - An I/O Channel Program is running for this device.
IA IAK - An interrupt or response has occurred for this device.
NO NOTRDY - Go to state %10 after Idle Channel Program is started.
ST STWAIT - The device monitor is starting an Idle Channel Program for this device. There is no IOQ associated with this type of request.
STATE - State of the device monitor. Specifies the next action to be taken in SIODM in servicing the request:

0 - start new request
1 - not used
2 - call driver initiator procedure
3 - call driver completor procedure
4 - not used
5 - process request completed
6 - initiate device recognition sequence
7 - start operator intervention wait
%10 - wait for interrupt (operator intervention) restart at state 0
%11 - wait for data segment freeze, then state 2
%12 - wait for driver initiator to be frozen, then allocate controller (state 2)
%13 - wait for I/O completion interrupt, then state 3
%14 - wait for controller, then call driver initiator
%15 - not used
%16 - wait for initiator make present, then state 2
%17 - wait for completor make present, then state 3

DLINK - A SYSDB relative pointer to the next DIT requesting this resource or service.

DCURREQP - A current request sysbase index.

DLDEV.(0:2) - I/O system type

0 - HP3000 Series 2/3
1 - HP3000 Series 33 (HPIB)
2 - Unused
3 - Unused

DLDEV.(2:6) - Unit number of this device. Zero if a single unit.

DLDEV.(8:8) - Logical device number of this device.

DSTAT - Set to a -1 when a system powerfail has occurred.

DSERR - Pointer to status to be logged.

Bits(0:7) - Number of words to be logged.
Bits(8:15) - Offset relative to DITP(0).

DMISC - Device dependent processing flags

LOCK'FLG - Lock flag denoting unload status of the disc volume.

0 - Allow operator unload to the volume.
1 - Deny operator unload to the volume.

IGNORE'INT'FLG - Ignore unexpected interrupt flag.

SUBSTATE - Indicates state of the idle channel program:

0 - Normal idle channel program wait
1 - Idle request being serviced wait

DSBUFADDR - SYSDB relative pointer to the system buffer element used to read the DSCT. Zero, if no element gotten.

DBADBLK1 - High order logical sector address of the bad block for the Defective Sector Table (DSCT) entry.

DBADBLK2 - Low order logical sector address of the bad block for the DSCT entry.

DBADXFER - Byte transfer left when bad block occurred.

DLOGERROR - CPVA(0) logged on hardware error status.

DSIOPSTOP - Stopped channel program relative offset location due to an error in CPVA(0).

DSTATUS - 20 bytes disc status logged on status error. (See CS'80 Disc Drive Status).

Caution: * Since the "C" MIT, word %10 and %11 of the DIT for disc devices have been used for DQHEAD and DQTAIL pointers for disc request queues. Word %10 is also used by the timer procedure to hold a timer request index (DTRLX). Unless word %10 of the DIT is freed up in a future MIT, timers cannot be implemented on any disc drivers.

DIT for 7970/71 Magnetic Tape

Device Information Table (DIT)

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   0| 0| 0|AC|RQ| 0|MU| 0|IO|IA| 0| 0| 0|   STATE   |   DFLAG
    +--+--+--+--+--+--+--+--+--+--+--+--+-----------+
   1| SYSDB relative pointer to the DIT for the next|   DLINK
    | device requesting this resource or service    |
    +-----------------------------------------------+
   2| SYSDB relative pointer to the first IOQ in    |   DIOQP
    | request list for this device                  |
    +---------+-------------+-----------------------+
   3|IOT   |    Phys. unit # | Logical device number |   DLDEV
    +---------+-------------+-----------------------+
   4| SYSDB relative pointer to Device Linkage Table|   DDLTP
    +-----------------------------------------------+
   5| SYSDB relative pntr to Interrupt Linkage Table|   DILTP
    +-----------------------------------------------+
   6|RW|RU|SH|CE|BO|                             |AA|   DSAVE
    +-----------------------------------------------+
   7| Hardware error pointer. Set when the driver   |   DSERR
    | detects an error.  Whenever <>0, the driver   |
    | monitor logs an I/O error and clears this word|
    +--+--+--+--+--+--------------------------------+
 %10| Bit 0 is set at completion of timer           |   DTIME
    +--+--+--+--+--+--------------------------------+
 %11| Interrupt status for this unit.  Set by the   |   DSTAT
    | driver each time it processes an interrupt.   |
    +-----------------------------------------------+
 %12| Holds the time out request entry index while  |   DRQST
    | a timer is active.                            |
    +-----------------------------------------------+
 %13|          Hardware logged error status         | DLOGERROR
    +-----------------------------------------------+
```

DFLAG - Flags and request state

AC	ACTIVE	- A monitor is currently servicing this device.
RQ	REQUEST	- A service request is pending while the monitor is active.
MU	MUNIT	- This device is on a multi-unit controller.
IO	IOPROG	- An I/O Channel Program is running for this device.
IA	IAK	- An interrupt or response has occurred for this device.
NO	NOTRDY	- Go to state %10 after Idle Channel Program is started.
ST	STWAIT	- The device monitor is starting an Idle Channel Program for this device. There is no IOQ associated with this type of request.

```
  STATE        - State of the device monitor.  Specifies the next action
                 to be taken in SIODM in servicing the request:
                   0 - start new request
                   1 - not used
                   2 - call driver initiator procedure
                   3 - call driver completor procedure
                   4 - not used
                   5 - process request completed
                   6 - initiate device recognition sequence
                   7 - start operator intervention wait
                 %10 - wait for interrupt (operator intervention)
                       restart at state 0
                 %11 - wait for data segment freeze, then state 2
                 %12 - wait for driver initiator to be frozen, then
                       allocate controller (state 2)
                 %13 - wait for I/O completion interrupt, then state 3
                 %14 - wait for controller, then call driver initiator
                 %15 - not used
                 %16 - wait for initiator make present, then state 2
                 %17 - wait for completor make present, then state 3

DLDEV - I/O system type, unit and logical device number
  IOT I/O TYPE- Type of I/O system
                 0 - HP3000 Series II/III
                 1 - HP3000 Series 33 (HP-IB)
                 2 - unused
                 3 - unsused

DSAVE - Device processing flags
  RW  RWBIT  - Indicates tape has been rewound.
  RU  RWUNLD - Indicates that a rewind/unload was performed to allow a
               write-ring mount.
  SH  SHORT  - A short read is in progress.  After completion of read,
               EOF is checked for and if not present, the requested
               bytes are transfered from the short-read buffer to the
               user's buffer.
  CE  CESTAT - Channel parity error processing is in progress.
  BO  BODEOF - Backspace record due to a data EOF processing is in
               progress.
  AA  AB'ACK - Abort Channel Program is executing.
$PAGE
DSTAT - Mag tape controller status

  BITS         USE

    0     END OF FILE

    1     BEGINNING OF TAPE
    2     END OF TAPE
    3     SINGLE TRACK ERROR (NOT LOGGED FOR READS)

    4     COMMAND REJECT
    5     FILE PROTECT
    6     MULTIPLE TRACK ERROR
```

7	UNIT ONLINE		
8	(NOT USED)		
9	UNIT NUMBER (MSB)		
10	UNIT NUMBER (LSB)		
11	TIMING ERROR		
12	TAPE RUNAWAY		
13	REWINDING	*	
14	UNIT BUSY	**	(REPORTED AS UNIT NOT READY)
15	INTERFACE BUSY	*	

FOR STATUS READ (3RD BYTE STATUS) DENOTES:

BITS ----	USE ---
0	0 (NOT USED)
1	0 (NOT USED)
2	POWER ON
3	COMMAND PARITY ERROR
4	*UNIT 3 PLACED ON LINE
5	*UNIT 2 PLACED ON LINE
6	*UNIT 1 PLACED ON LINE
7	*UNIT 0 PLACED ON LINE

*NOTE: BITS 4,5,6,7 NOT USED BY DRIVER.

DIT for 7976 Magnetic Tape

There is one DIT per physical device. If a physical device represents more than one logical device, the logical device number is obtained from the IOQ element. The following diagram shows the DIT used for the mag tape driver.

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
   +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  0| 0| 0|AC|RQ| 0|MU| 0|IO|IA| 0| 0| 0|   STATE   |   DFLAG
   +--+--+--+--+--+--+--+--+--+--+--+--+-----------+
  1| SYSDB relative pointer to the DIT for the next|   DLINK
   | device requesting this resource or service    |
   +-----------------------------------------------+
  2| SYSDB relative pointer to the first IOQ in    |   DIOQP
   | request list for this device                  |
   +--------+--------------+-----------------------+
  3|     |    Phys. unit # | Logical device number |   DLDEV
   +--------+--------------+-----------------------+
  4| SYSDB relative pointer to Driver Linkage Table|   DDLTP
   +-----------------------------------------------+
  5| SYSDB relative pntr to Interrupt Linkage Table|   DILTP
   +-----------------------------------------------+
  6|RW|RU|SH|  |DC|PF|                             |   DSAVE
   +-----------------------------------------------+
  7| Hardware error status.  Set when the driver   |   DSERR
   | detects an error.  Whenever <>0, the driver   |
   | monitor logs an I/O error and clears this word|
   +--+--+--+--+--+--------------------------------+
%10| Bit 0 is set at completion of timer           |   DTIME
   +--+--+--+--+--+--------------------------------+
%11| Interrupt status for this unit.  Set by the   |   DSTAT
   | driver each time it processes an interrupt.   |
   +-----------------------------------------------+
%12| Holds the time out request entry index while  |   DRQST
   | a timer is active.                            |
   +-----------------------------------------------+
%13| Error log. Contains 5 valid bytes of status   |   DLOGERROR
   +-----------------------------------------------+
```

DFLAG - Flags and request state

AC ACTIVE - A monitor is currently servicing this device.
RQ REQUEST - A service request is pending while the monitor is active.
MU MUNIT - This device is on a multi-unit controller.
IO IOPROG - An I/O Channel Program is running for this device.
IA IAK - An interrupt or response has occurred for this device.
NO NOTRDY - Go to state %10 after Idle Channel Program is started.
ST STWAIT - The device monitor is starting an Idle Channel Program for this device. There is no IOQ associated with this type of request.

STATE - State of the device monitor. Specifies the next action to be taken in SIODM in servicing the request:

- 0 - start new request
- 1 - not used
- 2 - call driver initiator procedure
- 3 - call driver completor procedure
- 4 - not used
- 5 - process request completed
- 6 - initiate device recognition sequence
- 7 - start operator intervention wait
- %10 - wait for interrupt (operator intervention) restart at state 0
- %11 - wait for data segment freeze, then state 2
- %12 - wait for driver initiator to be frozen, then allocate controller (state 2)
- %13 - wait for I/O completion interrupt, then state 3
- %14 - wait for controller, then call driver initiator
- %15 - not used
- %16 - wait for initiator make present, then state 2
- %17 - wait for completor make present, then state 3

DSAVE - Device processing flags

RW RWBIT - Indicates tape has been rewound.

RU RWUNLD - Indicates that a rewind/unload was performed to allow a write-ring mount.

SH SHORT - A short read is in progress. After completion of read, EOF is checked for and if not present, the requested bytes are transfered from the short-read buffer to the user's buffer.

DC DSFLAG - Transfer used data chaining - used for computing the transmission log.

PF POWER - Device power up indication.

DSTAT - Mag tape controller status

BITS	USE
0	END OF FILE (EOF)
1	BEGINNING OF TAPE (BOT) / LOAD POINT (LP)
2	END OF TAPE (EOT)
3	SINGLE TRACK ERROR (NOT LOGGED FOR READS)
4	COMMAND REJECT (REJECT)
5	FILE PROTECT (NOT WRITE ENABLED; NO WRITE RING)
6	MULTIPLE TRACK ERROR (MTE)
7	UNIT ONLINE
8	GCR (6250 BPI DENSITY)
9	UNIT NUMBER (MSB)

10	UNIT NUMBER (LSB)		
11	TIMING ERROR		
12	TAPE RUNAWAY		
13	REWINDING	*	
14	UNIT BUSY	**	(REPORTED AS UNIT NOT READY)
15	INTERFACE BUSY	*	

DIT for Series III Card Reader

```
      0  1  2   3   4   5    6   7   8   9   10 11  12       15
     |--|--|---|---|---|-----|--|----|---|----|---|---|----------|
     | 0| 0|ACT|REQ| 0 |  0  |  | I/O|IAK|READ| NR|   | MSTATE   |  DFLAG
     |  |  |   |   |   |     |  |PROG|   |DONE|MSG|   |          |
     |-----------------------------------------------------------|
    1|                 DITP LINK TO NEXT DIT                     |  DLINK
     |-----------------------------------------------------------|
    2|                 IOQP POINTER TO 1st REQUEST               |  DIOQP
     |-----------------------------------------------------------|
    3|          UNIT #             |        LOGICAL DEVICE #     |  DLDEV
     |-----------------------------------------------------------|
    4|                 DRIVER LINKAGE TABLE POINTER              |  DDLTP
     |-----------------------------------------------------------|
    5|                 INTERRUPT LINKAGE TABLE POINTER           |  DILTP
     |-----------------------------------------------------------|
    6|                       (SEE BELOW)                         |  DSTAT
     |-----------------------------------------------------------|
    7|                 ERROR STATUS IF NOT 0                     |  DSERR
     |-----------------------------------------------------------|
  %10|                 REQUESTED WORD COUNT                      |  DWCNT
     |-----------------------------------------------------------|
```

DSTAT bits:

```
BIT0=SIO OK
BIT1=0
BIT2=INT PENDING
BIT3=TIMING ERROR
BIT4=LIGHT DARK CHECK
BITS 5-6 =    00 COLUMN BINARY MODE
              01 UNUSED
              10 PACKED BINARY MODE
              11 HOLLERITH-TO-ASCII MODE
BIT7=COMPARE ERROR
BIT8=EOF DETECTED
BITS 9-10 =   00 NORMAL
              01 HOPPER EMPTY
              10 UNUSED
              11 STACKER FULL
BIT11=INVALID HOLLERITH
BIT12=PICK FAIL OR MOTION CHECK
BIT13=TEST
BIT14=TROUBLE
BIT15=NOT READY
```

CARD READER DIT FIELD DEFINITIONS

DFLAG - Flags and device state

ACTIVE	Monitor is currently active servicing this device.
REQUEST	Service for this device was requested while the monitor was active.
IOPROG	SIO program in progress.
IAK	Interrupt occurred or request aborted or preempted.
READDONE	Previous read resulted in an EOF with a backup save requested. The data has been saved in an auxiliary buffer and will be passed back on the next read request.
NRMESSAGE	Set when a not ready message has been issued, and cleared when the reader is found ready. Used to prevent multiple Not Ready messages when power is turned on.
MSTATE	Monitor State. See SIODM specifications for details.

DLINK - SYSDB relative ponter to the DIT for the next device requesting service for this resource.

DIOQP - SYSDB relative pointer to the first IOQ element in the request list for this device.

DLDEV - Logical device number and unit number.

UNIT	Unit number of device.
LDEVN	Logical device number.

DDLTP - SYSDB relative pointer to driver linkage table (DLT).

DILTP - SYSDB relative pointer to interrupt linkage table (ILT).

DSTAT - Device interrupt status. Contains the device interrupt status at the last interrupt. See hardware ERS for details.

DSERR - Device interrupt error status. If not zero, then holds the device interrupt status from an operation with an erroneous completion status. Causes SIODM to log an error.

DWCNT - Holds the requested transfer count in words.

DIT for HPIB Card Reader

There is one DIT per physical device. If a physical device represents more than one logical device, the logical device number is obtained from the IOQ element. The following diagram shows the DIT used for the card reader driver.

```
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15    MNEMONIC
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   0| 0| 0|AC|RQ| 0|MU| 0|IO|IA|NO|ST| 0|   STATE   |    DFLAG
    +--+--+--+--+--+--+--+--+--+--+--+--+-----------+
   1| SYSDB relative pointer to the DIT for the next|    DLINK
    | device requesting this resource or service    |
    +-----------------------------------------------+
   2| SYSDB relative pointer to the first IOQ in    |    DIOQP
    | request list for this device                  |
    +--------+-------------+------------------------+
   3|  IOT   | Phys. unit # | Logical device number |    DLDEV
    +--------+-------------+------------------------+
   4| SYSDB relative pointer to Driver Linkage Table|    DDLTP
    +-----------------------------------------------+
   5| SYSDB relative pntr to Interrupt Linkage Table|    DILTP
    +-----------------------------------------------+
   6|RD|AF|                                         |    DSAVE
    +-----------------------------------------------+
   7| Hardware error status.  Set when the driver   |    DSERR
    | detects an error.  Whenever <>0, the driver   |
    | monitor logs an I/O error and clears this word|
    +--+--+--+--+-----------------------------------+
 %10| Not Used                                      |    DTIME
    +--+--+--+--+-----------------------------------+
 %11|               Request word count              |    DWCNT
    +-----------------------------------------------+
 %12| Device Status.  Read from device during       |    DSTAT
    | each execution of the channel program.        |
    +-----------------------------------------------+
 %13! Logging will be done from here.               !    DLOGERROR
    +-----------------------------------------------+
```

DFLAG - Flags and request state
  AC ACTIVE - A monitor is currently servicing this device.
  RQ REQUEST - A service request is pending while the monitor is active.
  MU MUNIT - This device is on a multi-unit controller.
  IO IOPROG - An I/O Channel Program is running for this device.
  IA IAK - An interrupt or response has occurred for this device.
  NO NOTRDY - Go to state %10 after Idle Channel Program is started.
  ST STWAIT - The device monitor is starting an Idle Channel Program for this device. There is no IOQ associated with this type of request.

STATE - State of the device monitor. Specifies the next action to be taken in SIODM in servicing the request:

- 0 - start new request
- 1 - not used
- 2 - call driver initiator procedure
- 3 - call driver completor procedure
- 4 - not used
- 5 - process request completed
- 6 - initiate device recognition sequence
- 7 - start operator intervention wait
- %10 - wait for interrupt (operator intervention) restart at state 0
- %11 - wait for data segment freeze, then state 2
- %12 - wait for driver initiator to be frozen, then allocate controller (state 2)
- %13 - wait for I/O completion interrupt, then state 3
- %14 - wait for controller, then call driver initiator
- %15 - not used
- %16 - wait for initiator make present, then state 2
- %17 - wait for completor make present, then state 3

DLDEV - Device logical device number

IOT I/O TYPE - I/O System type
- 0 = Series II / III I/O system
- 1 = HP-IB
- 2 = unused
- 3 = unused

DSAVE - Device processing flags

RD READDONE - A card has already been read.

AF ABORTFLAG - A device clear has already been sent for this series of aborted IOQs.

LINE PRINTER DIT (SERIES II/III)

There is one DIT per physical device. If a physical device represents more than one logical device, the logical device number is obtained from the IOQ element. The following diagram shows the DIT used for IOLPRT0.

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15     MNEMONIC
   +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  0| 0| 0|AC|RQ| 0| 0| 0|IO|AK|PS|NE|TF|   STATE   |    DFLAG
   +--+--+--+--+--+--+--+--+--+--+--+--+-----------+
  1| SYSDB relative pointer to the DIT for the next|    DLINK
   | device requesting this resource or service    |
   +-----------------------------------------------+
  2| SYSDB relative pointer to the first IOQ in    |    DIOQP
   | request list for this device                  |
   +--------+--------------------------------------+
  3|        | Phys unit #  | Logical device number |    DLDEV
   +--------+--------------------------------------+
  4| SYSDB relative pointer to Driver Linkage Table|    DDLTP
   +-----------------------------------------------+
  5| SYSDB relative ptr to Interrupt Linkage Table |    DILTP
   +-----------------------------------------------+
  6| Controller interrupt status.  Set by GIP each |    DSTAT
   | time it processes an interrupt for this DIT.  |
   | See individual field descriptions on nxt page.|
   +-----------------------------------------------+
  7| Hardware error pointer. Set when the driver   |    DSERR
   | detects an error.  Whenever <> 0, the driver  |
   | monitor logs an I/O error and clears this word|
   +-----------------------------------------------+
%10| Bit 0 is set at completion of 2-second timer. |    DTIME
   +-----------------------------------------------+
%11| Timer Request List Index.  Not used except to |    DTRLX
   | clear the request after timing out.           |
   +-----------------------+-----------------------+
%12|                       |Last byte if odd bytcnt|    DLAST
   |                       | Data byte for VFC or  |
   |                       | left margin download  |
   +--+--+--+--+-----------+-----------------------+
%13|VF|PF|BT|TL|Left margin| Vertical Format Code  |    DVFC1
   |MD|RS|JB|NR|           |                       |
   +--+--+--+--+-----------+-----------------------+
%14| Lines left to skip    | %202 (2608) or %102.  |    DVFC2
   | (subtypes 1, 2, >15   | Skip to channel 3 pre-|
   | line slew request)    | to postspace print.   |
   +-----------------------+-----------------------+
%15|        HARDWARE ERROR LOGGING STATUS          | DLOGERROR
   +-----------------------------------------------+
%16| DVR DEPENDENT FLAGS =>   |PS|NE|TF|           |    DDF
   +--+--+--+--+--+--+--+--+--+--+--+--+-----------+
```

DFLAG.AC - Active. A monitor is currently servicing this device. *
DFLAG.RQ - Request. A service request is pending while the monitor is active. *
DFLAG.IO - An I/O channel program is in progress. Decrement SIOCOUNT and check for multiple channels when complete. *
DFLAG.AK - Interrupt Acknowledge. An interrupt has occurred. *
DDF .PS - Prespace. The last request was a prespace (space then fill buffer) operation.
DDF .NE - Not Empty. The print buffer is not empty. Causes a print when changing from pre- to postspace or before ejecting a page for a File Open, File Close or Device Close.
DDF .TF - Top of Form. The last request ended with a skip to channel 1 (page eject).

* Not examined or modified by IOLPRT0.

DFLAG.STATE - State of the device monitor. Specifies the next action to be taken by SIODM in servicing the request. Not used within IOLPRT0.

DSTAT.(0:1) - SIO OK. Set when no SIO channel program is in progress, that is, it is OK to start one.
.(1:1) - WIO OK. Set when it is OK to execute a WIO instruction or a doubleword WRITE channel order. If clear, indicates that a one word transfer is in progress.
.(2:1) - Interrupt Pending. If set, indicates one or more bits of the Interrupt Status Byte (DSTAT.(8:8) are set.
.(3:2) - U.I. Transfer State. Used mostly for hardware maintenance. See U.I. card manual (30051-90001) for details.
.(5:1) - Device Flag. Indicates a print-and-advance-paper sequence in progress. Since the 2608 buffers such commands, this signal may be shorter than with other printers.
.(6:1) - Always 0. DSTAT.(8:8) always contains the Interrupt Status Byte.
.(7:1) - Not used. Always 0.
.(8:3) - Varies among HP-supported line printers according to the table below:

SUBTYPE	MODEL(S)	BIT 8	BIT 9	BIT 10
0	2610, 2614	LINE PRINTED	READY	NOT READY
1	2607	Not used	READY	NOT READY
2	2613, 2617, 2618, 2619	Not used	READY	NOT READY
3	2617J (KATAKANA)	Not used	READY	NOT READY
4	2608	ON LINE	NOT READY	VFC CHAN 9

.(11:1) - Data Transfer Interrupt bit. Always 0.

.(12:1) - Not used. Always 0.

.(13:1) - Programmed Interrupt bit. True if interrupt request was generated by:

a) SIN machine instruction,

b) INTERRUPT channel order, or

c) END-WITH-INTERRUPT channel order.

.(14:1) - Transfer Error Interrupt bit. True if interrupt was generated by:

a) an illegal memory address,

b) a memory parity error, or

c) a multiplexer parity error during data xfr to U.I.

.(15:1) - Time-out Interrupt bit. Set if 5-second timer on U.I. card is enabled, then times out without being cleared.

DLAST.(8:8) - Request dependent. If a print request has an odd number of bytes, this word holds the final byte. For VFC downloads, contains the associated data byte (6 or 8 lines per inch and number of lines in VFC). For left margin downloads, also contains the associated data byte (the number of columns to offset).

DVFC1.(0:1) - VFC Modified. 2608 only. Indicates that an external VFC has been downloaded into the 2608.

DVFC1.(1:1) - Power Fail/Reset. 2608 only. The 2608 has suffered a Power Failure or someone has pressed the front panel Reset button. In either case, the printer's operating environment has been destroyed, and must be reloaded by the operator.

DVFC1.(2:1) - Between Jobs. Set when a Device Close is executed, cleared when an FOPEN is performed. 2608 Power Fail/ Master Reset's will be cleared but not reported while this bit is set (thus avoiding an extraneous console message when the printer is powered up).

DVFC1.(3:1) - TALLY'NOT'READY. Set when an off-line condition is detected on a 2607. Causes a three-second delay when the 2607 comes back on-line.

DVFC1.(4:4) - Left margin offset (2608 only). Stored during each :DOWNLOAD which specifies a left margin and restored to printer following a 2608 power fail or reset. Set to 0 when system is initialized.

DVFC1.(8:8) - Request dependent. Contains the carriage control byte sent to the printer during a print request.

DVFC2.(0:8) - LINES'LEFT'OVER. Has two functions:

1) The 2607/13/17/18/19 can only slew (skip) a maximum of 15 lines per print command (not counting VFC skips, which can be of any length). Slew requests > 15 lines must be broken up. This byte holds the number of lines (greater than 15) which remain to be slewed at any point of a request to such a printer, or 0 if the number of lines to skip is <= 15. This mechanism is not needed (and this field is therefore 0) for CDC and 2608 line printers, which can slew up to 63 lines at a time.
2) The carriage control characters "0" and "-" specify double and triple spacing, respectively. But if you use the equivalent channel skip, you get skips to the next odd and third lines, respectively, which is not the same as double and triple spacing. If you slew (advance paper) 2 or 3 lines, you can easily print over the paper perforations unless your program watches out for such things. We avoid this by examining the NO'AUTO'PAGE eject bit (IOQ(QPAR2).(14:1)). If it is set, then the request is treated like a normal slew and LINES'LEFT'OVER is not used. If it is clear (auto eject desired), then we simulate the multiple line skip by doing two ("0") or three ("-") skips to channel 3 (single spaces with auto page eject for the standard VFC). In this case, LINES'LEFT'OVER holds the number of such single spaces remaining in the request.

a 1990

DVFC2.(8:8) - %202 for 2608, %102 otherwise. Causes skip to channel 3 (single space with auto page eject). Used when last request left data in print buffer (prespace) and current operation is postspace. Buffer is dumped first, using this byte as carriage control.

2608 LINE PRINTER DIT (HPIB SYSTEMS)

There is one DIT per physical device. If a physical device represents more than one logical device, the logical device number is obtained from the IOQ element (however, there is only one device per 2608 controller.) The following diagram shows the DIT used for the 2608 line printer driver.

```
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
     +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    0| 0| 0|AC|RQ| 0| 0| 0|IO|IA|NO|ST| 0|   STATE   |   DFLAG
     +--+--+--+--+--+--+--+--+--+--+--+--+-----------+
    1| SYSDB relative pointer to the DIT for the next|   DLINK
     | device requesting this resource or service    |
     +-----------------------------------------------+
    2| SYSDB relative pointer to the first IOQ in    |   DIOQP
     | request list for this device                  |
     +--------+--------------+-----------------------+
    3|IOT     |  Phys. unit # | Logical device number |   DLDEV
     +--------+--------------+-----------------------+
    4| SYSDB relative pointer to Driver Linkage Table|   DDLTP
     +-----------------------------------------------+
    5| SYSDB relative pntr to Interrupt Linkage Table|   DILTP
     +-----------------------------------------------+
    6|VM|     |    TAB     |                |PS|FL|TP|   DSAVE
     +-----------------------------------------------+
    7| Hardware error pointer. Set when the driver   |   DSERR
     | detects an error.  Whenever <>0, the driver   |
     | monitor logs an I/O error and clears this word|
     +--+-----+-----------+-----------------+--+--+--+
  %10| Bit 0 is set at completion of timer           |   DTIME
     +--+-----+-----------+-----------------+--+--+--+
  %11| Holds the time out request entry index while  |   DRQST
     | a timer is active.                            |
     +-----------------------------------------------+
  %12|          Hardware logged error status         | DLOGERROR
     +-----------------------------------------------+
```

DFLAG - Flags and request state

AC ACTIVE - A monitor is currently servicing this device.

RQ REQUEST - A service request is pending while the monitor is active.

IO IOPROG - An I/O Channel Program is running for this device.

IA IAK - An interrupt or response has occurred for this device.

NO NOTRDY - Go to state %10 after Idle Channel Program is started.

ST STWAIT - The device monitor is starting an Idle Channel Program for this device. There is no IOQ associated with this type of request.

STATE - State of the device monitor. Specifies the next action to be taken in SIODM in servicing the request:
0 - start new request
1 - not used
2 - call driver initiator procedure
3 - call driver completor procedure
4 - not used
5 - process request completed
6 - initiate device recognition sequence
7 - start operator intervention wait
%10 - wait for interrupt (operator intervention) restart at state 0
%11 - wait for data segment freeze, then state 2
%12 - wait for driver initiator to be frozen, then allocate controller (state 2)
%13 - wait for I/O completion interrupt, then state 3
%14 - wait for controller, then call driver initiator
%15 - not used
%16 - wait for initiator make present, then state 2
%17 - wait for completor make present, then state 3

DLDEV - I/O system type, unit and logical device number
IOT I/O TYPE- Type of I/O system
0 - HP3000 Series II/III
1 - HP3000 Series 33 (HP-IB)
2 - unused
3 - unsused

DSAVE - Device processing flags
VM VFCMOD - VFC has been modified.
TAB TABDFAULT - System tab default.
PS PRESPACE - Last request used prespacing.
FL FULL - Line printer buffer is full.
TP TOP - Printer is at top of form

2608 Line Printer Status

BYTE 1 & BYTE 2:

BITS	USE
0	ON LINE
1	NOT READY
2	VFC CHANNEL 9 (BOTTOM OF FORM)
3	VFC CHANNEL 12 (TOP OF FORM)
4	VFC INITIALIZED
5	6/8 LINES PER INCH
6	(NOT USED)
7	POWER RESTORED/UNIT RESET
8	ON LINE
9	PRINT MECH ERROR
10	SELF TEST FAILURE
11	PAPER ERROR
12	SELF TEST MODE
13	6/8 LPI
14	PLATEN/RIBBON ERROR
15	(NOT USED)

- BYTE 3: PRINT MODE
  - BITS 0-7 MODE NUMBER
- BYTE 4: PRIMARY/SECONDARY
  - BITS 0-3 SECONDARY CHARACTER SET CODE
  - BITS 4-7 PRIMARY CHARACTER SET CODE
- BYTE 5: SELF TEST
  - BITS 0 PASS FAIL
  - BITS 1-7 SUBTEST NUMBER
- BYTE 6: 6 LPI DOT ROW COUNT
- BYTE 7: 6 LPI FORM LINE NUMBER
- BYTE 8: 6 LPI FORM LENGTH IN LINES
- BYTE 9: 8 LPI DOT ROW COUNT
- BYTE 10: 8 LPI FORM LINE NUMBER
- BYTE 11: 8 LPI FORM LENGTH IN LINES
- BYTE 12: FIRMWARE IDENTIFICATION CODE
- BYTE 20: POWER-UP LANGUAGE
  - BITS 0-3 SECONDARY CHARACTER SET CODE
  - BITS 4-7 PRIMARY CHARACTER SET CODE

HIOCIPRO DIT (HP2608S)

There is one DIT per physical device. If a physical device represents more than one logical device, the logical device number is obtained from the IOQ element (however, this driver only supports one device per controller.) The following diagram shows the DIT used for the HP-IB CIPER physical driver.

<table>
<tr><th>Word #</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>MNEMONIC</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>AC</td><td>RQ</td><td>0</td><td>0</td><td>0</td><td>IO</td><td>IA</td><td>NO</td><td>ST</td><td>0</td><td colspan="4"></td><td></td></tr>
<tr><td>1</td><td colspan="16">SYSDB relative pointer to the DIT for the next device requesting this resource or service</td><td>DLINK</td></tr>
<tr><td>2</td><td colspan="16">SYSDB relative pointer to the first IOQ in request list for this device</td><td>DIOQP</td></tr>
<tr><td>3</td><td colspan="2">IOT</td><td colspan="6">Phys. unit #</td><td colspan="8">Logical device number</td><td>DLDEV</td></tr>
<tr><td>4</td><td colspan="16">SYSDB relative pointer to Driver Linkage Table</td><td>DDLTP</td></tr>
<tr><td>5</td><td colspan="16">SYSDB relative pointer to Intrp Linkage Table</td><td>DILTP</td></tr>
<tr><td>6</td><td>VS</td><td>AB</td><td>RE</td><td>TP</td><td>NR</td><td colspan="3">NR CNT</td><td colspan="8">DEVICE STATUS</td><td></td></tr>
<tr><td>7</td><td colspan="16">Hardware error status. Set when the driver detects an error. Whenever <0, the driver monitor logs an I/O error and clears this word</td><td>DSERR</td></tr>
<tr><td>8</td><td colspan="16">Bit 0 is set at completion of timer</td><td>DTIME</td></tr>
<tr><td>9</td><td colspan="16">Holds the time out request entry index while a timer is active.</td><td>DRQST</td></tr>
<tr><td>10</td><td>RF</td><td>UE</td><td>DE</td><td>TO</td><td colspan="3">UNIT CNT</td><td colspan="3">DATA CNT</td><td colspan="3">TO CNT</td><td colspan="3">PRTY CNT</td><td></td></tr>
<tr><td>11</td><td colspan="16">Error logging location #1</td><td>DLOGERROR</td></tr>
<tr><td>12</td><td colspan="16">Error logging location #2</td><td>DLOGCOUNT</td></tr>
</table>

DFLAG - Flags and request state

AC	ACTIVE	- A monitor is currently servicing this device.
RQ	REQUEST	- A service request is pending while the monitor is active.

IO IOPROG - An I/O Channel Program is running for this device.

IA IAK - An interrupt or response has occurred for this device.

NO NOTRDY - Go to state %10 after Idle Channel Program is started.

ST STWAIT - The device monitor is starting an Idle Channel Program for this device. There is no IOQ associated with this type of request.

STATE - State of the device monitor. Specifies the next action to be taken in SIODM in servicing the request:

0 - start new request
1 - not used
2 - call driver initiator procedure
3 - call driver completor procedure
4 - not used
5 - process request completed
6 - initiate device recognition sequence
7 - start operator intervention wait
%10 - wait for interrupt (operator intervention) restart at state 0
%11 - wait for data segment freeze, then state 2
%12 - wait for driver initiator to be frozen, then allocate controller (state 2)
%13 - wait for I/O completion interrupt, then state 3
%14 - wait for controller, then call driver initiator
%15 - not used
%16 - wait for initiator make present, then state 2
%17 - wait for completor make present, then state 3

DLDEV - I/O system type, unit and logical device number

0 - HP3000 Series 2/3
1 - HP3000 Series 33 (HPIB)
2 - Unused
3 - Unused

DSAVE - Device processing flags

VS - VALID STATUS - Set to indicate Device Status has been updated.

AB - DVRABFLAG - Sequence Abort in progress due to ABORT request.

RE - RETRYFLAG - Sequence Abort in progress due to an error.

TP - TIMERPOPPED - Current error is due to software timer popping.

NR - NOTRDYFLAG - Not Ready Wait in progress.

NR CNT - Number of Not Ready Waits during this request.

DEVICE STATUS - Device status returned during a Sequence Abort.

BIT 8 - CRC available and enabled.
BIT 9 - Reserved.
BIT 10 - Reserved.
BIT 11 - Reserved.
BIT 12 - Power fail or reset has occurred.
BIT 13 - A protocol error has been detected.
BIT 14 - A parity error has been detected.
BIT 15 - The peripheral has data to send.

DSERR - Pointer to status to be logged.

Bits.(0:8) - Number of words to be logged.
Bits.(8:8) - Offset relative to DITP(0).

DCOUNTS - Error flags and error counts (4).

RF - REQ FAILED - An error has forced this request to be aborted.

UE - UNIT ERROR - The current error is a Unit Error.

DE - DATA ERROR - The current error is a Data Error.

TO - TIME OUT - The current error is a GIC Time Out Error.

UNIT CNT - Number of Unit Errors during this request.

DATA CNT - Number of Data Errors during this request.

TO CNT - Number of GIC Time Outs during this request.

PRTY CNT - Number of HP-IB Parity Errors during this request.

2631 OR 2619A LINE PRINTER DIT (FOR HPIB SYSTEMS)

There is one DIT per physical device. If a physical device represents more than one logical device, the logical device number is obtained from the IOQ element (however, there is only one device per 2631 controller.) The following diagram shows the DIT used for the 2631 line printer driver.

```
       0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
     +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    0| 0| 0|AC|RQ| 0| 0| 0|IO|IA|NO|ST| 0|    STATE   |   DFLAG
     +--+--+--+--+--+--+--+--+--+--+--+--+-----------+
    1| SYSDB relative pointer to the DIT for the next|   DLINK
     | device requesting this resource or service    |
     +-----------------------------------------------+
    2| SYSDB relative pointer to the first IOQ in    |   DIOQP
     | request list for this device                  |
     +-----+---------------+-------------------------+
    3|IOT  |    Phys. unit # | Logical device number |   DLDEV
     +-----+---------------+-------------------------+
    4| SYSDB relative pointer to Driver Linkage Table|   DDLTP
     +-----------------------------------------------+
    5| SYSDB relative pntr to Interrupt Linkage Table|   DILTP
     +-----------------------------------------------+
    6|                              |BJ|AB|PS|FL|TP|   DSAVE
     +-----------------------------------------------+
    7| Hardware error status.  Set when the driver   |   DSERR
     | detects an error.  Whenever <>0, the driver   |
     | monitor logs an I/O error and clears this word|
     +-----------------------------------------------+
  %10| Bit 0 is set at completion of timer           |   DTIME
     +-----------------------------------------------+
  %11| Holds the time out request entry index while  |   DRQST
     | a timer is active.                            |
     +-----------------------------------------------+
  %12|          Hardware logged error status         | DLOGERROR
     +-----------------------------------------------+
```

DFLAG - Flags and request state

AC	ACTIVE	- A monitor is currently servicing this device.
RQ	REQUEST	- A service request is pending while the monitor is active.
IO	IOPROG	- An I/O Channel Program is running for this device.
IA	IAK	- An interrupt or response has occurred for this device.
NO	NOTRDY	- Go to state %10 after Idle Channel Program is started.
ST	STWAIT	- The device monitor is starting an Idle Channel Program for this device. There is no IOQ associated with this type of request.

STATE - State of the device monitor. Specifies the next action to be taken in SIODM in servicing the request:

0 - start new request
1 - not used
2 - call driver initiator procedure
3 - call driver completor procedure
4 - not used
5 - process request completed
6 - initiate device recognition sequence
7 - start operator intervention wait
%10 - wait for interrupt (operator intervention) restart at state 0
%11 - wait for data segment freeze, then state 2
%12 - wait for driver initiator to be frozen, then allocate controller (state 2)
%13 - wait for I/O completion interrupt, then state 3
%14 - wait for controller, then call driver initiator
%15 - not used
%16 - wait for initiator make present, then state 2
%17 - wait for completor make present, then state 3

DLDEV - I/O system type, unit and logical device number
IOT I/O TYPE - Type of I/O system
0 - HP3000 Series 2/3
1 - HP3000 Series 33 (HPIB)
2 - Unused
3 - Unused

DSAVE - Device processing flags

BJ	BETJOB	- Between jobs flag. If set, suppress Powerfail message.
AB	ABORT	- Abort (caused by Powerfail or Operator) has occurred.
PS	PRESPACE	- Last request used prespacing.
FL	FULL	- Line printer buffer is full.
TP	TOP	- Printer is at top of form

2680A DIT

```
          0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
         +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  DIT0   !0 !0 !AC!RQ!0 !0 !SP!CP!IA!NR!SW!  !   STATE   !  DFLAG
         -------------------------------------------------
     1   !            POINTER TO NEXT DIT                !  DLINK
         -------------------------------------------------
     2   !         POINTER TO ACTIVE IOQ OR ZERO         !  DIOQP
         -------------------------------------------------
     3   ! IOT !   UNIT NUMBER   ! LOGICAL DEVICE NUMBER !  DLDEV
         -------------------------------------------------
     4   !       DRIVER LINKAGE TABLE POINTER            !  DDLTP
         -------------------------------------------------
     5   !       INTERRUPT LINKAGE TABLE POINTER         !  DILTP
         -------------------------------------------------
     6   !      SPECIAL ERROR CONDITIONS TO BE LOGGED    !  DSTAT
         -------------------------------------------------
     7   !            ERROR LOGGING INFORMATION          !  DSERR
         -------------------------------------------------
     8   !T !     TIMEOUT INDICATION IN BIT 0            !  DTIME
         -------------------------------------------------
     9   !        TIMER REQUEST INDEX (TRL) OR ZERO      !  DTRLX
         -------------------------------------------------
    10   !          CURRENT DATA WRITE BYTE COUNT        !  DCBCNT
         -------------------------------------------------
    11   !            CURRENT DATA WORD COUNT            !  DCWCNT
         -------------------------------------------------
    12   !         # OF WORDS LEFT TO TRANSFER           !  DRCNT
         -------------------------------------------------
    13   !  BUFFER OFFSET FOR NEXT # OF WORDS TO XFER.   !  DOFFSET
.FLAG=ON
         -------------------------------------------------
    14   !                                             !D!  DDEBUG
         -------------------------------------------------
    15   ! I/O STATUS BLOCK WORD 1 GETS LOGGED FROM HERE !  DLOGBUFFER
         -------------------------------------------------
    16   ! I/O STATUS BLOCK WORD 3 GETS LOGGED FROM HERE !
         -------------------------------------------------
 17/32   ! I/O STATUS AREA (16 WORDS, SEE DEFINITION)    !  DIOSTAT
         +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

DFLAG - DEVICE RELATIVE FLAGS.

AC	ACTIVE BIT. 1 IMPLIES A MONITOR CURRENTLY SERVICING THIS DEVICE.
RQ	REQUEST BIT. 1 IMPLIES SERVICE REQUESTED WHILE MONITOR IS ACTIVE.
SP	SIO PREEMPTION. IF SET THEN A PREEMPTIVE REQUEST HAS BEEN QUEUED FOR THIS DEVICE. PREEMPT CODE IS SET IN IOQ ELEMENT.
CP	CHANNEL PROGRAM IN PROGRESS. IF SET, THEN A CHANNEL PROGRAM IS CURRENTLY EXECUTING.
IA	IF SET, AN INTERRUPT OR RESPONSE HAS OCCURED.
NR	IF SET, DEVICE IS IN A NOT READY OR OPERATOR WAIT.

SW IF SET, AN IDLE CHANNEL PROGRAM SHOULD BE STARTED FOR THIS DEVICE.

MSTATE CURRENT DRIVER STATE AS DEFINED BY THE MONITOR. ALLOWABLE STATES ARE:
0 - START REQUEST
1 - NOT USED(BUT RESERVED)
2 - CALL DRIVER INITIATOR
3 - CALL DRIVER COMPLETOR
4 - UNUSED(BUT RESERVED)
5 - COMPLETE REQUEST..PERHAPS RETURN TO USER.
6 - UNEXPECTED INTERRUPT OCCURRED.
7 - START OPERATOR INTERVENTION WAIT.
%10 - WAITING (ON OPERATOR). RESTART AT 0.
11 - WAITING (DATA MAKEPRESENT/FREEZING)
12 - WAITING (INITIATOR CODE MAKEPRESENT/FREEZE)
13 - WAITING (FOR COMPLETION INTERRUPT)
14 - WAITING (FOR DEVICE CONTROLLER AVAILABILITY)
15 - UNUSED(BUT RESERVED)
16 - WAITING (INITIATOR CODE MAKEPRESENT)
17 - WAITING (COMPLETOR CODE MAKEPRESENT)

DLDEV - I/O SYSTEM TYPE, UNIT AND LOGICAL DEVICE NUMBER.
IOT I/O SYSTEM TYPE.
0 - HP3000 SERIES II/III (SIO/DIO)
1 - HP-IB
2 - RESERVED
3 - RESERVED

DCBCNT - CURRENT BYTE COUNT TO BE TRANSFERRED.

DCWCNT - CURRENT WORD COUNT TO BE TRANSFERRED.

DRCNT - REMAINING WORD COUNT TO TRANSFER.

DOFFSET - OFFSET IN BUFFER OF NEXT # WORDS TO TRANSFER.

DDEBUG - IF BIT 15=1 THEN DEBUGGING INFO WILL BE SENT TO CONSOLE

DLOGBUFFER - STATUS WORDS 1 & 3 ARE MOVED HERE TO BE LOGGED IF THEY WERE LOGGED FROM THE I/O STATUS BLOCK THEIR CONTENTS MIGHT BE CHANGED BEFORE THEY WERE LOGGED.

DIOSTAT - I/O STATUS AREA 16 WORDS, SEE I/O STATUS BLOCK DEFINITION.

I/O STATUS BLOCK

```
       0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   0  !0 !--THE "OR" OF WORDS 1/15 IS LOCATED HERE----! DIT 17
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   1  !OF!MS!PW!PE!TE!  !  !  !  !  !  !  !  !  !  !  !      18
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   2  !  !  !  !  !  !  (RESERVED)  !  !  !  !  !  !  !      19
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   3  !              MCS FAULT NUMBER                 !      20
     +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   4  !CL!FL!VL!CU!FU!VU!IL!IP!ST!SB!IR!MP!NJ!NM!TL!NC!      21
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   5  !LP!PS!NC!  !  !  (RESERVED)  !  !  !  !  !  !  !      22
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   6  !PL!OP!IP!  !  !  (RESERVED)  !  !  !  !  !  !  !      23
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   7  !  !  !  !  !  !  (RESERVED)  !  !  !  !  !  !  !      24
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   8  !  !  !  !  !  !  (RESERVED)  !  !  !  !  !  !  !      25
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   9  !  !  !  !  !  !  (RESERVED)  !  !  !  !  !  !  !      26
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  10  !  !  !  !  !  !  (RESERVED)  !  !  !  !  !  !  !      27
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  11  !  !  !  !  !  !  (RESERVED)  !  !  !  !  !  !  !      28
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  12  !          RECORD NUMBER OF ERROR               !      29
      +--             IF WORD 4 TO 6 <> 0           --+
  13  !                NON-ZERO                       !      30
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  14  !  SHEET NUMBER OF ERROR IF WORD 4 TO 6 <> 0   !     31
      +--                     OR                    --+
  15  !  LAST SHEET TRANSFERRED IF "JOB" & POWER-ON   !      32
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

WORD 0 - EACH BIT IS THE 'OR' OF ONE WORD IN THE TABLE (EXCEPT BIT 0 WHICH IS NOT USED). THEREFORE, BIT .(1:1) IS SET IF WORD 1 IN THE TABLE IS NON-ZERO.

```
WORD 1 - BIT= 0 - (OF)  ONLINE/OFFLINE BIT.
              1 - (MS)  MESSAGE BEING DISPLAYED ON THE 2680A CONSOLE.
              2 - (PW)  POWER UP COMPLETED SINCE LAST I/O STATUS READ.
              3 - (PE)  PARITY ERROR DETECTED ON PHI COMMAND.
              4 - (TE)  TRANSMISSION ERROR DETECTED IN THE PRINTER.
           5/15 -       RESERVED.  UNUSED.
```

WORD 2 - NOT USED. RESERVED.

WORD 3 - MCS FAULT NUMBER. CONTAINS AN INTEGER DESCRIBING THE LAST FAULT TO OCCUR SINCE THE LAST TIME THE I/O STATUS WAS READ OR THE HP2680A WAS POWERED DOWN. IF THE WORD IS ZERO THERE IS NO MCS FAULT. SEE DCS ERS FOR A DESCRIPTION OF THE MCS FAULT NUMBERS.

WORD 4 - BIT= 0 - (CL) NO ROOM FOR ATTEMPTED CHARACTER SET LOAD.
1 - (FL) NO ROOM FOR ATTEMPTED FORM LOAD.
2 - (VL) NO ROOM FOR ATTEMPTED VFC LOAD.
3 - (CU) ATTEMPT TO PRINT DATA AND THERE IS NO CURRENTLY SELECTED CHARACTER SET.
4 - (FU) ATTEMPT TO SELECT AN UNDEFINED FORM SET.
5 - (VU) ATTEMPT TO PRINT DATA AND THERE IS NO CURRENTLY SELECTED VFC SET.
6 - (IL) ATTEMPT TO PRINT DATA AND THERE IS NO CURRENTLY SELECTED LOGICAL PAGE TABLE (LPT) ENTRY.
7 - (IP) ATTEMPT TO MOVE PEN OFF THE LOGICAL PAGE.
8 - (ST) THE 2680A COULD NOT PROCESS ALL OF THE DATA BEFORE IT WAS SUPPOSED TO BE TRANSFERRED TO THE DRUM/PAPER. DATA WAS LOST!
9 - (SB) SPOOLER BLOCK CONTAINS FORMAT ERROR.
10 - (IR) INVALID RECOVERY BLOCK RECEIVED FROM SPOOLER.
11 - (MP) MAXIMUM NUMBER OF COPIES PER PHYSICAL PAGE HAS BEEN EXCEEDED. THIS IS A RESULT OF THE SPOOLER PROCESS SETTING THE MAXIMUM COPIES PER PAGE WITH FUNCTION CODE 132.
12 - (NJ) A COMMAND OR FUNCTION CODE WAS RECEIVED WHEN NO "JOB" WAS IN PROGRESS. THE COMMAND OR FUNCTION WAS IGNORED BY THE DCS.
13 - (NM) NO MEMORY. 2680A DYNAMIC MEMORY ALLOCATION HAS DETECED THAT MAIN MEMORY IS COMPLETELY OCCUPIED WITH CHARACTER SETS, VFC'S, FORMS AND DATA SUCH THAT THE 2680A CANNOT PROCESS THE CURRENT INPUT DATA. DATA WILL BE LOST!
14 - (TL) ATTEMPT TO PRINT DATA AND THERE ARE MORE THAN THE MAXIMUM ALLOWABLE LOGICAL PAGE TABLE (LPT) ENTRIES SELECTED.
15 - (NC) A NON-EXISTENT VFC CHANNEL WAS SKIPPED TO.

WORD 5 - BIT= 0 - (LP) LOGICAL PAGE TRUNCATED TO FIT PHYSICAL PAGE.
1 - (PF) PAGE SIZE PEQUIRED BY PROGRAMMER DID NOT MATCH PAGE SIZE SET BY OPERATOR. OPERATOR PAGE SIZE PREVAILS.
2 - (NC) NO CHARACTER SET SELECTED.
WORD 6 - BIT= 0 - (PL) NOT ENOUGH MEMORY FOR PICTURE DOWNLOAD.
1 - (OP) ATTEMPT TO PRINT MORE THAN 64 PICTURES ON A PHYSICAL PAGE.
2 - (IP) ATTEMPT TO PRINT A PICTURE WHICH IS NOT PRESENT.

WORDS 7/11 NOT USED. RESERVED FOR FUTURE USE.

WORDS 12/13 - THE RECORD NUMBER WHICH CONTAINS THE OFFENDING ERROR AS DEFINED BY WORD FOUR. IF A POWER FAIL OCCURS DURING A "JOB", THE POWER FAIL BIT IS SET AND A SHEET NUMBER IS MADE AVAILABLE IN WORDS FOURTEEN AND FIFTEEN. HOWEVER, THE RECORD NUMBER IS LOST AND CANNOT BE REPORTED. THESE WORDS OCCUR IN A "JOB" ONLY.

WORDS 14/15 - THE SHEET NUMBER ON WHICH THE ERROR OCCURED AS DEFINED BY WORD FOUR. IF AN ERROR OCCURS IN THE ENVIRONMENT FILE AT THE START OF A "JOB", THEN THIS NUMBER WILL BE ZERO. IN ADDITION, WHEN A POWER FAIL OCCURS DURING A "JOB", THE POWER ON BIT IS SET IN WORD ONE AND THE SHEET NUMBER OF THE LAST SUCCESSFULLY TRANSFERRED PAGE IS PLACED HERE. THIS INFORMATION IS FOR USE BY THE SPOOLER SHOULD A RECOVERY OF A "JOB" BE DETERMINED. THESE WORDS OCCUR IN "JOB" ONLY.

ALL WORDS OF THE I/O STATUS ARE CLEARED WHENEVER THE STATUS BLOCK IS RETURNED TO THE HOST. IT IS UP TO THE HOST CPU TO RETAIN ANY ON-GOING STATUS BITS REQUIRED.

QMISC -

```
        0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
       +--+--+--+--+-----+--+--+--+--+--------+--------+--+
IOQ3   !MB!RB!AB!IO!TO!             !  XFER  ! PARITY !  !  QMISC
       +--+--+--+--+-----+--+--+--+--+--------+--------+--+
```

WHERE:

.(0:1) - MB	USER REQUESTED TRANSFER IN EXCESS OF 4096 WORDS. THE DRIVER CAN WRITE UP TO 4096 WORDS TO THE 2680A. IN ORDER TO HANDLE UP TO 32K WORDS, MULTIPLE WRITES ARE USED WITHOUT A RETURN TO THE USER WHO CALLED THE DRIVER. THIS BIT INDICATES THAT MULTIPLE WRITES ARE BEING DONE TO THE 2680A.
.(1:1) - RB	THE CURRENT WRITE BLOCK MUST BE RETRIED.
.(2:1) - AB	USER REQUESTED ABORT IN PROGRESS FLAG.
.(3:1) - IO	I/O STATUS HAS BEEN READ AND IS AVAILABLE.
.(4:1) - TO	GENERAL I/O CONTROLLER TIMED OUT.
.(5:4) - RESERVED	NOT CURRENTLY USED.
.(9:3) - XFER	2680A TRANSFER ERROR COUNTER.
.(12:3)- PARITY	CHANNEL PROGRAM COMMAND PARITY ERROR COUNTER.
.(15:1)- RESERVED	NOT CURRENTLY USED.

NOTE IN THE ABOVE, SINGLE BIT FIELDS ARE AS DEFINED WHEN THE BIT IS A LOGIC "1".

30119 CARD READER/PUNCH DIT

Everything is the same as the SIO DIT
and standard IOQ except as noted below:

1. DIT (9)

```
       0   1   2   3   4   5  6  7  8  9   10 11 12,13 14  15
     -----------------------------------------------------------
     |C  |S  |S  |H  |I  |E  |I |P |P |S  |E |S |M     |C  |T  |
%11  | B |   |   |   | I | O |  |  |  | P |  |  | O    | O |   |  DACCP
     |  F|  C|  S|  S|  F|  F| B| R| N|  D| C| m|  D   |  N|  R|
     |   |   |   |   |   |   |  |  |  |   |  |  |     E|   |   |
     -----------------------------------------------------------
```

DIT(9).(0:1)	CBF	Clear Buffer Full - 0= the next card leaving the hopper will be read by the device. 1= the read buffer will be cleared when next card leaves the hopper.
DIT(9).(1:1)	SC	Stacker Control - 0=all cards are stacked in right hopper until device goes not ready. 1= cards are stacked per bit 2.
DIT(9).(2:1)	SS	Stacker Select - 0=Right stacker (stacker 1) 1= Left Stacker (stacker 2).
DIT(9).(3:1)	HS	Hopper Select - 0= Pick from rear hopper (primary hopper). 1= Pick from front hopper (secondary hopper).
DIT(9).(4:1)	IIF	Inhibit Input Feed - Inhibit picking a card when card currently in wait station is eject to a hopper.
DIT(9).(5:1)	EOF	End Of File has been detected on a read oper
DIT(9).(6:1)	IB	Internal Buffer -An internal buffer is being used. The buffer is the SIO area in the ILT.
DIT(9).(7:1)	PR	Print - Print on the next card to pass the print station.
DIT(9).(8:1)	PN	Punch - Punch 80 columns of data on the next card to pass the punch station.
DIT(9).(9:1)	SPD	Separate Print Data - Print data other than that being punched on the next card to pass the punch and print station.
DIT(9).(10:1)	EC	Eject Card - Eject on a write after a read. Used when reading one card then punching one card (last card was read).

30119 CARD READER/PUNCH (CONT.)

DIT(9).(11:1)	Sm	Stacker Mode -Saved staker mode on last read
DIT(9).(12:2)	MODE	Access Mode - 0= File opened for Read only 1= File opened for Write only 2= File opened for Read/Write
DIT(9).(14:1)	CON	Control - 0= no FCONTRL has occured for this file (use default settings). 1= FCONTROL has been done on this file (use settings in this DIT word for controlling this device).
DIT(9).(15:1)	TR	Timer Request - A timer request is pending. Timer request index is in word %12.

2. DIT(10) Timer request index (see DIT(9).(15:1)).

3. QMISC{IOQ(4)}

```
  0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
--------------------------------------------------
|I  |N  |W  |                                    |
|   | R |   |              UNUSED                |
|  O|  I|  R|                                    |
--------------------------------------------------
```

IOQ(4).(0:1)	IO	I/O initiated - waiting for completion interrupt.
IOQ(4).(1:1)	NRI	Waiting for a "Not Ready Interrupt" to bring the device back online.
IOQ(4).(2:1)	WR	Write - current operation is a write operation.
IOQ(4).(3:13)		Not Used

```
                     INP DIT
                     -------
        0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
       +--+--------+--------+--------+--------+--------+
DIT0   |  0   |AC|RQ|TI| 0|PR|IO|IN|SM|MAMSTATE| IOSTATE|  DFLAG
       -------------------------------------------------
   1   |          POINTER TO NEXT DIT                  |  DLINK
       -------------------------------------------------
   2   |         INPUT REQUEST QUEUE                   |  DIOQP
       -------------------------------------------------
   3   |                       | LOGICAL DEVICE NUMBER |  DLDEV
       -------------------------------------------------
   4   |     DRIVER LINKAGE TABLE POINTER              |  DDLTP
       -------------------------------------------------
   5   |     INTERRUPT LINKAGE TABLE POINTER           |  DILTP
       -------------------------------------------------
   6   |    INTERRUPT STATUS                           |  DSTATUS
       -------------------------------------------------
   7   |        SOFTWARE TIMER REQUEST INDEX           |  DTRLX
       --------------------------------------------------------
   8   |TO|                                            |  DTIME
       -------------------------------------------------
   9   |          READY QUEUE HEAD POINTER             |  READYQ
       -------------------------------------------------
  10   |          READY QUEUE TAIL POINTER             |  READYQTL
       -------------------------------------------------
  11   |         ACTIVE QUEUE HEAD POINTER             |  ACTIVEQ
       -------------------------------------------------
  12   |         ACTIVE QUEUE TAIL POINTER             |  ACTIVEQTL
       -------------------------------------------------
  13   |         WAITED QUEUE HEAD POINTER             |  WAITEDQ
       -------------------------------------------------
  14   |         WAITED QUEUE TAIL POINTER             |
       -------------------------------------------------
  15   |EO|WP|TR|  |PFSTATE|UF|PR|NR|SD| OS    |     |AB|  DSTATE
       -------------------------------------------------
  16   |    RESERVED           | MESSAGE TO INP TYPE   |  DOUTMSG
       -------------------------------------------------
  17   |         REQUEST IDENTIFIER (@IOQP)            |  DOUTID
       -------------------------------------------------
  18   |             PARAMETER 1 (QMISC)               |  DOUTP1
       -------------------------------------------------
  19   |                OUT COUNT                      |  DOUTCNT
       -------------------------------------------------
  20   |             PARAMETER 2 (QPAR2)               |  DOUTP2
       -------------------------------------------------
  21   |          SEND DIALOGUE COUNTER                |  DSEND
       -------------------------------------------------
  22   |         RECEIVE DIALOGUE COUNTER              |  DRECV
       -------------------------------------------------
  23   |         "MESSAGE SENT" EOT BUFFER             |  DEOT
       -------------------------------------------------
```

INP DIT (cont)

```
   -------------------------------------------------
24 |     RESERVED         | MESSAGE FROM INP TYPE |  DINMSG
   -------------------------------------------------
25 |          REQUEST IDENTIFIER (@IOQP)          |  DINID
   -------------------------------------------------
26 |     ERROR CODE       |LS|           | CSTATUS |  DRSTATUS
   -------------------------------------------------
27 |                 IN COUNT                     |  DINCNT
   -------------------------------------------------
28 |             TRANSMISSION LOG                 |  DXLOG
   -------------------------------------------------
29 |                PARAMETER                     |  DINPARM
   -------------------------------------------------
30 |       TRACE READY REQUESTS COUNT             |  DTRCNT
   -------------------------------------------------
31 |   EXTERNAL TRACE EXTRA DATA SEGMENT NUMBER   |  DDSTN
   --------------------------------------------------------
32 |     RESERVED        |  OUT MSG TYPE AT ERROR |  DERROR
   -------------------------------------------------
33 |          REQUEST IDENTIFIER (@IOQP)          |
   -------------------------------------------------
34 |            PARAMETER 1 (QMISC)               |
   -------------------------------------------------
35 |                 OUT COUNT                    |
   -------------------------------------------------
36 |            PARAMETER 2 (QPAR2)               |
   -------------------------------------------------
37 |           LAST CS ERROR CODE                 |  DCSERR
   -------------------------------------------------
38 |       IOQP POINTER AT TIME OF ERROR          |  DSAVE
   -------------------------------------------------
39 !TP!PHY DRVR VERSN # ! LOGICAL DRIVER VERSION # !  DVERSION
   --------------------------------------------------------
40 !     RESERVED         !  IN MSG TYPE AT ERROR !  DERRORI
   -------------------------------------------------
41 !          REQUEST IDENTIFIER (@IOQP)          !
   -------------------------------------------------
42 !     ERROR CODE      !LS!           !  STATUS !
   -------------------------------------------------
43 !                 IN COUNT                     !
   -------------------------------------------------
44 !             TRANSMISSION LOG                 !
   -------------------------------------------------
45 !                PARAMETER                     !
   -------------------------------------------------
46 !           DRIVER ERROR CODE                  !  DDRVRERR
   -------------------------------------------------
47 !           MONITOR ERROR CODE                 !  DMNTRERR
   -------------------------------------------------
```

INP DIT (cont)

```
   --------------------------------------------------
48 !HARDWARE ERROR STATUS !   SIO PROGRAM INDEX    !  DSERR
   --------------------------------------------------
49 !     TOOTHPICK HARDWARE ERROR STATUS           !  DTP'ERROR
   --------------------------------------------------
50 |  ADDITIONAL TOOTHPICK HARDWARE ERROR STATUS   |
   --------------------------------------------------
51 !      DRIVER TRACE READ IOQ POINTER            !  DTR'IOQP
   --------------------------------------------------
```

DFLAG - Flags, IOSTATE and MAMSTATE

ACTIVE - If set, the Driver is active servicing this device

REQUEST - If set, service for this device was requested while the Driver was active. The Driver is run again to insure servicing of the condition which caused REQUEST to be set.

DO'TIMING - If set, the hardware and software timers are started in the normal manner when performing an operation. If clear, no timing is done.

SIOPREEMPT- Preemptive request queued by ATTACHIO. Not used by this Driver.

IOPROG - If set, an I/O program is in progress. Set by STARTIO and cleared by GIP. Not used by the Driver.

IAK - Interrupt Acknowledge If set, an interrupt has occurred or a software timeout has completed.

SIMULATOR - If set, all I/O is to be simulated. The Driver will set flags in the DRT instead of calling STARTIO.

MAMSTATE - Memory Manager State

0 - Null, no Memory Management requests or condition
1 - Not used
2 - Data segment associated with the first request in the Active Queue is being made present and frozen.
3 - Data segment associated with the first request in the Active Queue is frozen in memory.
4 - Data segment associated with the second request in the Active Queue is being made present and frozen. Implies the data segment associated with the first request is frozen.
5 - Data segments associated with the first and second requests on the Active Queue are frozen in memory.
6 - Not used
7 - Not used

IOSTATE - Current I/O program operation being performed

- 0 - Inactive No I/O in progress
- 1 - Idle Read The Idle Read I/O program has been started.
- 2 - Sending message An I/O program which sends a message without data and then goes to the Idle Read section of the I/O program has been started.
- 3 - Sending data An I/O program which sends a message and data and then goes to the Idle Read section has been started.
- 4 - Send message and interrupt An I/O program which sends a message without data then interrupts and halts when the message is sent has been started.
- 5 - Send data and interrupt An I/O program which sends a message with data then interrupts and halts has been started.
- 6 - Receive data An I/O program which sends a message and receives data then interrupts and halts has been started.
- 7 - Do not start I/O Used to hold off requesting any I/O activity during a power on reset or when an error occurs.

DLINK - Link word for the linked list of devices waiting to be serviced by the I/O process associated with this device

DIOQP - System DB relative pointer to the first element in the requests to be processed list for this device The requests are queued to this list by ATTACHIO but in processing, the are moved to other queues depending of the state of the request The Driver always attempts to keep this list empty.

DLDEV - Logical Device Number of this device

DDLTP - System DB relative pointer to the Driver Linkage Table. (DLT)

DILTP - System DB relative pointer to the Interrupt Linkage Table. (ILT)

DSTATUS - Controller hardware status Set by GIP on interrupt and the Physical Driver during certain service operations See INP ERS for description. For the Toothpick version, this word contains the software timeout flags as described for the word DTIME below.

DTRLX - Timer request index for software timeouts as returned by the MPE procedure TIMEREQ

DTIME - Timed out flags and type 3 driver process PCB Number

TIMED - If set, a software timeout has completed

READYQ - System DB relative pointer to the IOQ for the first request in the Ready Queue. If zero, the Ready Queue is empty.

READYQTL - System DB relative pointer to the last IOQ in the Ready Queue. When the queue is empty, this word points to the word preceding the queue head pointer in the DIT.

ACTIVEQ - System DB relative pointer to the IOQ for the first request in the Active Queue. If zero, the Active Queue is empty.

ACTIVEQTL - System DB relative pointer to the last IOQ in the Active Queue. When the queue is empty, this word points to the word preceding then queue head pointer in the DIT.

WAITEDQ - System DB relative pointer to the IOQ for the first request in the Waited Queue. If zero, the Waited Queue is empty.

WAITEDQTL - System DB relative pointer to the last IOQ in the Waited Queue. When the queue is empty, this word points to the word preceding then queue head pointer in the DIT.

DSTATE - Driver state and control flags

ERRORONLY - If set, the Driver trace record is to be returned to the Trace Process only when an error occurs.

WRAP - If set, the Driver will overlay the oldest trace entry when a trace record overflow occurs. If clear, entries are lost when an overflow occurs.

TRACEON - If set the Driver trace facility is enabled and the Driver generates trace entries for most of its local subroutine calls.

PFSTATE - Power failure recovery state
- 0 - No power failure recovery in progress
- 1 - Powerfailure detected on the mainframe before INP indication. Check for completion of any pending I/O and then wait in PFSTATE 2 for INP to pfail.
- 2 - Power failure detected on the Mainframe before INP has indicated a power failure. Wait for INP to indicate a power failure.
- 3 - Power failure indicated by INP before being informed by the Mainframe power failure routines. Wait for the Mainframe power failed request.
- 4 - Power failure indicated both on the Mainframe and by INP. Power failure recovery may be started.
- 5 - Send Redo The Mainframe receive count was less than INP's send count so the dialogue must be restarted. The Driver is sending the Redo message.
- 6 - Send Ignore The Mainframe send count was greater than INP's receive count so any part of a dialogue so far received is to be ignored and the entire dialogue will retransmitted. The Driver is sending Ignore message.
- 7 - Recovered. The Mainframe and INP dialogue counters agree or mainframe not sending, so no recovery is necessary. The Driver is sending the recovered message informing INP to go back to its normal mode.

UNFRZ - If set, the source data segment is to be unfrozen when the data has been transmitted to the INP. If clear, the source data segment remains frozen until a reuquest complete indication is returned by the INP.

PASSREADS - If set, then read requests are to be passed around other requests which have been impeded because no buffers are available on the INP.

NOTRDYWAIT- If set, then a request has been impeded because no buffers were available on the INP.

SENDING - If set, an I/O program which send sends a message, with or without associated data has been started but not completed.

OPENSTATE - Operational state of the Driver and INP
- 0 - Not opened or closed

1 - In ROM The device has been opened but the RAM Operating System has not been entered
2 - Crashed Some catastrophic error has occured
3 - In RAM. The device has been opened, down loaded and is in the RAM Operating System.

ABORT - If set one or more requests have been aborted but the abort was not done because the aborted request was in the process of doing a Memory Management function or I/O when when request to abort was processed. The actual abort will take place when the Memory Management function completes.

The following five words hold the message block which is sent to INP when the Physical Driver is called to send a message with or without associated data. The Logical Driver sets the message contents into this area and calls the Physical Driver to send the message.

DOUTMSG - Message type code for messages sent to INP

DOUTID - Request identifier associated with the message being sent.

DOUTP1 - Parameter one of the message being sent to INP

DOUTCNT - Count parameter of the message being sent to INP

DOUTP2 - Parameter two of the message being sent to INP

DSEND - Messages sent counter. This word contains the number of messages sent since the RAM Operating System was entered. It is used for power failure recovery.

DRECV - Messages received counter. This word contains the number of messages received from INP since the RAM Operating System was entered. It is used for power failure recovery.

DEOT - End of dialogue flag. When a message has been sent and the EOT indicating INP has received the message is transmitted, it is received into this word. This flag is used to indicate to the Logical Driver that a transmission has been completed and that the Physical Driver should be called to check the completion status and update the IOSTATE.

INP DIT (CONT)

The following six words are the data area into which messages from INP are received. The Physical Driver constructs I/O programs which reference this area.

DINMSG - Message type code of message from INP

DINID - Request Identifier associated with message from INP

DRSTATUS - Request Completion status

DINCNT - Number of bytes of data to be received associated with the completion of a request which results in data being sent from INP.

DXLOG - Transmission log to be returned when the request identified by DINID is completed.

DINPARM - Parameter associated with the completion of this request. This word is return in the X register by IOSTATUSX.

DTRCNT - Trace ready pending count. This word contains the number of Trace Ready messages recieved but not satisfied by Trace Read requests.

DDSTN - If not zero then internal Driver extra data segment tracing is enabled and this is the data segment number into which the trace entries are to be set.

DERROR - Driver Error block. The following sixteen words are used to store information describing the current operations being performed when a catastrophic Driver error occurred. A catastrophic error occurres on illogical Driver control data, MPE errors or when INP does not respond in an expected manner. The first five word block is used to hold the current or last message transmitted to INP when a catastrophic error condition was detected. It contains the data in the same format as message to INP block.

DCSERR - CS Error Code associated with a catastrophic Driver error

DSAVE - Request Identifier of the request being processed when a catastrophic Driver error was detected

INP DIT (CONT)

DVERSION - Version numbers of the Physical and Logical Drivers

TP - If set, the Physical Driver is for the Toothpick System

PVERSION - Physical Driver version number

LVERSION - Logical Driver version number

DERROR1 - The six word block beginning here is used to hold the last message received from INP before a catastrophic Driver error was detected. It contains the data in the same format as the message from INP block.

DDRVRERR - Holds the code specifying the catastrophic error detected by the Physical Driver. See ERRORS under the PHYSICAL DRIVER INTERNAL SPECIFICATIONS for the definition.

DMNTRERR - Holds the code specifying the catastrophic error detected by the Logical Driver. See ERRORS under the LOGICAL DRIVER INTERNAL SPECIFICATIONS for the definition.

DSERR - Hardware Controller status when a catastrophic Driver error was detected.

HSTATUS - Left byte of the DSTATUS word at time of error

SIOPX - SIO program area relative index to the last order executed or current order being executed at time of error.

DTP'ERROR - Toothpick hardware error status. To be defined.

DTR'IOQP - If not zero then an IOQP pointer to the Trace Read request which is supplying the locked and frozen buffer into which the Driver places trace entries to generate a trace record.

DLOGX - Driver local trace buffer index. This is the index relative to the Driver local trace buffer to place the next trace entry.

DLOGBUF - Driver local trace buffer. This buffer extends from here to the end of the DIT.

DISC REQUEST TABLE AND DISC REQUESTS

Requests for disc transfers are effected by acquiring an entry from the Dis Request Table (DISCREQTAB), filling the proper information, and calling the DISCQMANAGER to link the request into the device's doubly linked request qu The head and tail of a device's request queue are contained in the devices's DIT.

```
  DISCREQTAB
                   +------------+
                   |            |
                   |            |
                   |            |
            +----->+------------+
+------+    |      |  Device's  |
|      |----+      |  Current   |
| DIT  |-------+   |  Request   |
|      |----+  |   +------------+
+------+    |  |   |            |
            |  |   |            |
            |  |   |            |
            |  |   |            |
            |  |   |            |
            |  |   |            |
            |  |   +------------+
            |  +-->|   First    |
            |      |  Request   |
            |      | in Queue   |<>----+
            |      +------------+      |
            |      |            |      |
            |      |            |      |
            |      |            |      |
            |      |            |      |
            |      |------------|      |
            |      |   Next     |<>----+
            |      |  Request   |
            |      | in Queue   |<>----+
            |      |------------|      |
            |      |            |      |
            |      +------------+      |
            |      |   Last     |<>----+
            +----->|  Request   |
                   | in Queue   |
                   +------------+
                   |            |
                   |            |
                   |            |
                   +------------+
```

DISC REQUEST TABLE

DISCREQTAB DST ENTRY# = 56 (%70)
DISCREQTAB PRT = %1031

DISC REQUEST TABLE ENTRY 0 FORMAT

```
                 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
                |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
DISCREQTAB00|    TOTAL ENTRIES       |   PRIMARY ENTRIES       |
                |-----------------------------------------------|
DISCREQTAB01|  IMPEDED PROCESS PCB  |   ENTRY SIZE (%20)      |
                |-----------------------------------------------|
DISCREQTAB02| TABLE INDEX OF HEAD OF AVAILABLE ENTRY LIST   |
                |-----------------------------------------------|
DISCREQTAB03| TABLE INDEX OF TAIL OF AVAILABLE ENTRY LIST   |
                |-----------------------------------------------|
DISCREQTAB04| MAX ENTRIES IN USE    |CURRENT ENTRIES IN USE |
                |-----------------------------------------------|
DISCREQTAB05|                   OVERFLOWS                     |
                |-----------------------------------------------|
DISCREQTAB06|                                                 |
            |                 TOTAL REQUESTS                  |
DISCREQTAB07|                                                 |
                |-----------------------------------------------|
DISCREQTAB08|SYSBASE INDEX OF HEAD OF DISABLED REQ Q          | DISCQHEAD
                |-----------------------------------------------|
DISCREQTAB09|SYSBASE INDEX OF TAIL OF DISABLED REQ Q          | DISCQTAIL
                |-----------------------------------------------|
DISCREQTAB10|///////////////////////////////////////////////|
                |-----------------------------------------------|
            .
            .
DISCREQTAB15|///////////////////////////////////////////////|
                |-----------------------------------------------|
```

DISC REQUEST ELEMENT FORMAT

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
        |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
Word 00 |A |M |D |S |I |B |C |D |M |Q |S |P |C |D |D |M |
        |B |M |I |B |O |K |O |A |M |U |I |F |U |I |I |S |
        |O |R |A |U |W |D |M |T |E |E |O |A |R |S |S |G |
        |R |E |G |F |A |  |P |A |R |U |F |I |R |A |A |D |
        |T |Q |  |  |K |  |  |F |R |E |A |L |E |B |T |O |
        |  |  |  |  |E |  |  |  |  |  |I |  |Q |  |  |N |
        |  |  |  |  |  |  |  |  |  |  |L |  |  |  |  |  |
        |-----------------------------------------------|
Word 01 |             REQUEST URGENCY CLASS             | URGCLASS
        |-----------------------------------------------|
Word 02 |       UNIT #          |        LDEV #         | LDEVN
        |-----------------------------------------------|
Word 03 |                 MISCELLANEOUS                 | MISC
        |-----------------------------------------------|
Word 04 | S|        DST      (IF PROCESS DISC I/O)      | DSTN
        |- - - - - - - - - - - - - - - - - - - - - - - -| S=STACK
        |          BANK      (IF SEGMENT TRANSFER)      |
        |-----------------------------------------------|
Word 05 | OFFSET INTO DATA SEG  (IF PROCESS DISC I/O)   | ADDR
        |- - - - - - - - - - - - - - - - - - - - - - - -|
        |     ADDRESS IN BANK   (IF SEGMENT TRANSFER)   |
        |-----------------------------------------------|
Word 06 |                       |       FUNCTION        | FUNC
        |-----------------------------------------------|
Word 07 |        COUNT/XLOG/CONTROL RETURNS             | XFERCNT
        |-----------------------------------------------|
Word 08 |         P1 (HODA IF SEGMENT TRANSFER          | PAR1
        |-----------------------------------------------|
Word 09 |         P2 (LODA IF SEGMENT TRANSFER          | PAR2
        |-----------------------------------------------|
Word 10 |         PCBN             | QUALIFIER | STATUS | STAT
        |-----------------------------------------------|
Word 11 |SYSBASE RELATIVE INDEX OF PREV REQUEST IN QUEUE| PREVREQP
        |-----------------------------------------------|
Word 12 |SYSBASE RELATIVE INDEX OF NEXT REQUEST IN QUEUE| NEXTREQP
        |-----------------------------------------------|
Word 13 |       SEGIDENTIFIER  (IF SEG TRANSFER)        |SEGIDENT
        |-----------------------------------------------|
Word 14 |DISPLACEMENT OF READ OR WRITE FROM SEG BASE(MM)|SEGDISP
        |-----------------------------------------------|
Word 15 |///////////////////////////////////////////////| AUXREQFLAGS
        |///////////////////////////////////////////////|
        |///////////////////////////////////////////////|
        |///////////////////////////////////////////////|
        |-----------------------------------------------|
Note: Upon return to free list, word (#1) becomes index of next  EE
 free entry.
```

Word 0 - QFLAG - Request dependent flags

Bit 0	.ABORT	Request has been aborted externally.
Bit 1	.MMREQ	Request is for a segment transfer.
Bit 2	.DIAG	Diagnostic request (not used).
Bit 3	.SBUF	System Buffer. Target is a system buffer whose index is relative to the start of the SBUF table.
Bit 4	.IOWAKE	Wake caller on completion of request.
Bit 5	.BLOCKED	Blocked I/O. Caller is waited in ATTACHIO until request is completed.
Bit 6	.COMPLETED	Request has been completed and caller woken if he had specified.
Bit 7	.DATAFRZN	Data segment has been made present and is frozen.
Bit 8	.MAMERRORD	MAM error on data segment make present.
Bit 9	.PREQQUEUED	Request is queued into disc's req queue
Bit 10	.SFAIL	Start SIO failure in GIP.
Bit 11	.PFAIL	The I/O has been aborted because of a powerfail.
Bit 12	.CURREQ	Request is device's current request.
Bit 13	.DISABLED	Request is disabled.
Bit 14	.DISATMPT	Attempted to disable this request.
Bit 15	.MSGDONE	A message request reply has completed.

Word 2 - QLDEV.QLDEVN - Logical Device Number
Word 3 - QMISC - Device dependent.

Word 4
QDSTN - If SYSBUFRs is clear then this is the DST number of the target data segment.If bit 0 is set then buffer address is a DB offset value instead of segment relative offset (implemented for NOWAIT IO and NOBUFF).
Word 5
QADDR - Offset in data segment or sys buff table to target data buffer.
Word 6
QFUNC.FUNC - Function code and qualifiers as specified by driver.

Word 7

QXFERCNT-On initiation specifies the word count if positive or byte count if negative. At completion of the request this location contains the actual transmission count in the same units as the call. Certain control requests return data through this location.

Word 8

QPAR1 - Parameter one, defined by driver

Word 9

QPAR2 - Parameter two, defined by driver

QMISC - Miscellaneous request dependent storage available to driver.

Word 10

QSTAT.PCBN - PCB Number of process which made this request. Zero if not associated with any process and IOQ is to be returned by the system.

.QUALIFIER - A code which further defies or qualifies the general status. Defined by driver.

.STATUS - General Status. Indicates current and result state of the request according to the following codes.

0 - not started or awaiting completion.
1 - successful completion.
2 - end of file detected.
3 - unusual condition.
4 - irrecoverable error.

NOTE: See I/O System Status Returns.

```
                  IOQ TABLE LAYOUT
                  ----------------


             |------------------------|
             | TOTAL #   |  PRIMARY # |
             |------------------------|
             | IMPEDED   | ENTRY SIZE |
             |PROCESS PCB|            |  TSIZE
             |------------------------|
      -------|      HEAD INDEX        |  THEAD
      |      |------------------------|
-------------|      TAIL INDEX        |  TTAIL
|     |      |------------------------|
|     |      |MAXIMUM OF |  CURRENT   |  TUSE
|     |      | IN USE    |  IN USE    |
|     |      |------------------------|
|     |      |      OVERFLOWS         |  TOVRFL
|     |      |------------------------|
|     |      |                        |
|     |      |    TOTAL REQUESTS      |  TRQSTS
|     |      |                        |
|     |      |------------------------|
|     |   -->|                        |
|     |   |  |------------------------|
|     |   |  |      INDEX OF 5        |-----
|     |   |  |------------------------|    |
|     |   |  |                        |    |
|     |   |  |                        |    |
|     |   |  |       ENTRY 1          |    |
|     |   |  |                        |    |
|     |   |  |                        |    |
|     |   |  |------------------------|    |
|-----|---|->|                        |<---|----
      |   |  |------------------------|    |   |
      |   |  |           0            |    |   |
      |   |  |------------------------|    |   |
      |   |  |                        |    |   |
      |   |  |                        |    |   |
      |   |  |        ENTRY 2         |    |   |
      |   |  |                        |    |   |
      |   |  |                        |    |   |
      |   |  |------------------------|    |   |
      |---|->|                        |    |   |
          |  |------------------------|    |   |
          |--|       INDEX OF 1       |    |   |
             |------------------------|    |   |
             |                        |    |   |
             |                        |    |   |
             |         ENTRY 3        |    |   |
             |                        |    |   |
             |                        |    |   |
             |------------------------|    |   |
             |                        |    |   |
```

IOQ (CONT.)

```
|-------------------------|     |    |
|      Indeterminate      |     |    |
|-------------------------|     |    |
|                         |     |    |
|                         |     |    |
|         ENTRY 4         |     |    |
|        (IN USE)         |     |    |
|                         |     |    |
|                         |     |    |
|-------------------------|     |    |
|                         |<----|    |
|-------------------------|          |
|        INDEX OF 2       |----------|
|-------------------------|
|                         |
|                         |
|         ENTRY 5         |
|                         |
|                         |
|-------------------------|
```

IOQ ELEMENT

```
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
   |        REQUEST DEPENDENT FLAGS                |
  0|                                               |  QFLAG
   |-----------------------------------------------|
  1|                  IOQ POINTER                  |  QLINK
   |-----------------------------------------------|
  2|        UNIT #          |        QLDEVN        |  QLDEV
   |-----------------------------------------------|
  3|                 MISCELLANEOUS                 |  QMISC
   |-----------------------------------------------|
  4|S |          DATA SEGMENT DST NUMBER           |  QDSTN S(Word 4(0:1)
   |                                               |  Stackflag  If set
   |-----------------------------------------------|  QADDR is DB rel.
   |                                               |
  5|                    ADDRESS                    |  QADDR
   |-----------------------------------------------|
  6|                        |       FUNCTION       |  QFUNC
   |-----------------------------------------------|
  7|          COUNT/XLOG/CONTROL RETURNS           |  QWBCT
   |-----------------------------------------------|
  8|                      P1                       |  QPAR1
   |-----------------------------------------------|
  9|                      P2                       |  QPAR2
   |-----------------------------------------------|
 10|          PCBN          |  QUALIFIER   | STATUS |  QSTAT
   |-----------------------------------------------|
```

QFLAG - Request dependent flags

Bit 0	.ABORT	Request has been aborted externally.
Bit 1	.SPECIAL	Special handling is to be applied to this request. For disc, indicates a memory management request.
Bit 2	.DIAG	Diagnostic request (not used).
Bit 3	.SBUF	System Buffer. Target is a system buffer whose index is relative to the start of the SBUF table.
Bit 4	.IOWAKE	Wake caller on completion of request.
Bit 5	.BLOCKED	Blocked I/O. Caller is waited in ATTACHIO until request is completed.
Bit 6	.COMPLETED	Request has been completed and caller woken if he had specified.

IOQ ELEMENT (CONT.)

Bit 7	.DATAFRZN	Data segment has been made present and is frozen.
Bit 8	.MAMERRORD	MAM error on data segment make present.
Bit 9	.PREQ	This request has been started but was preempted by a MAM request.
Bit 10	.SFAIL	Start SIO failure in GIP.
Bit 11	.PFAIL	The I/O has been aborted because of a powerfail.
Bits12-13	.PREMPT	Premptive type code: 1-soft, 2-hard.
Bit 15	.MSGDONE	A message request reply has completed.

QLINK - SYSDB relative pointer to next IOQ element. Points to first word of element.

QLDEV.QLDEVN - Logical Device Number

QMISC - Device dependent.

QDSTN - If SYSBUFRs is clear then this is the DST number of the target data segment.If bit 0 is set then buffer address is a DB offset value instead of segment relative offset (implemented for NOWAIT IO and NOBUFF).

QADDR - Offset in data segment or sys buff table to target data buffer.

QFUNC.FUNC - Function code and qualifiers as specified by driver.

QWBCT - On initiation specifies the word count if positive or byte count if negative. At completion of the request this location contains the actual transmission count in the same units as the call. Certain control requests return data through this location.

QPAR1 - Parameter one, defined by driver

QPAR2 - Parameter two, defined by driver

QMISC - Miscellaneous request dependent storage available to driver.

QSTAT.PCBN - PCB Number of process which made this request. Zero if not associated with any process and IOQ is to be returned by the system.

.QUALIFIER - A code which further defies or qualifies the general status. Defined by driver.

.STATUS - General Status. Indicates current and result state of the request according to the following codes.

0 - not started or awaiting completion.
1 - successful completion.
2 - end of file detected.
3 - unusual condition.
4 - irrecoverable error.

I/O SYSTEM STATUS RETURNS

	STATUS %
0 - PENDING	
1 - WAITING FOR COMPLETION	10
2 - DOING ERROR RECOVERY	20
3 - NOT READY WAIT	30
4 - NO WRITE RING WAIT	40
5 - NEW PAPER TAPE WAIT	50
1 - SUCCESSFUL	
0 - NORMAL	1
1 - READ TERMINATED WITH SPECIAL CHARACTER	11
2 - TAPE RETRY FOR SUCCESS REQUIRED	21
3 - LOW TAPE OR END OF TAPE AFTER WRITE	31
2 - END OF FILE	
1 - PHYSICAL END OF FILE	12
2 - DATA	22
3 - END OF DATA	32
4 - HELLO	42
5 - BYE	52
6 - JOB	62
7 - END OF JOB	72
3 - UNUSUAL CONDITION	
1 - TERMINAL PARITY ERROR	13
2 - TERMINAL READ TIMED OUT	23
3 - I/O ABORTED EXTERNALLY	33
4 - DATA LOST	43
5 - DATA SET NOT READY OR DISCONNECT OR UNIT NOT ON LINE	53
6 - ABORTED BECAUSE OF POWER FAIL	63
7 - BOT AND BSR, BSF REQUEST	73
10 - TAPE RUNAWAY	103
11 - EOT AND WRITE REQUEST	113
12 - NO WRITE RING AFTER REQUEST TO OPERATOR	123
13 - END OF TAPE (PAPER TAPE LOW)	133
14 - PLOTTER LIMIT SWITCH REACHED	143
15 - ENABLE SUBSYSTEM BREAK AND NO CONTROL Y PIN	153
16 - READ TIME RETURNED OVERFLOW	163
17 - BREAK STOPPED READ	173
20 - WRITE AND NO CARD IN WAIT STATION	203
21 - DEVICE POWERED ON - OPERATING ENVIRONMENT LOST	213
27 - VFC HAS BEEN RESET	273

I/O SYSTEM STATUS RETURNS (CONT.)

4 - IRRECOVERABLE ERROR

0 - INVALID REQUEST	4
1 - TRANSMISSION ERROR	14
2 - I/O TIME OUT	24
3 - TIMING ERROR	34
4 - SIO FAILURE	44
5 - UNIT FAILURE	54
6 - INVALID DISC ADDRESS	64
7 - TAPE PARITY ERROR	74
11 - PAPER TAPE TAPE ERROR	114
12 - SYSTEM ERROR	124
13 - INVALID SBUF INDEX	134
14 - CHANNEL FAILURE, TIMEOUT OR NO RESPONSE FROM CONTROLLER	144
15 - UNINITIALIZED MEDIA (LINUS)	154
16 - NO SPARE BLOCKS AVAILABLE	164
17 - DELETED RECORD DETECTED ON IBM FLOPPY DISC	174
20 - LABELED DEVICE UNAVAILABLE AFTER REELSWITCH	204
21 - PARITY ERROR DETECTED ON PHI COMMAND (EPOC)	214

IOQ ELEMENT FOR 7976A MAGNETIC TAPE

```
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   0|     Request dependent flags (see below)       |   QFLAG
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   1| SYSDB relative pointer to next IOQ element.   |   QLINK
    | Points to first word of element.              |
    +-----------------------+-----------------------+
   2|                       | Logical device number |   QLDEV
    +--+--+--+--+--+-----+--+-----+--------+--------+
   3| R| B| F| G|BO| TOUT| FSCNTR | BSCNTR | RTCNTR |   QMISC
    +--+--+--+--+--+-----+--------+--------+--------+
   4| S| If QFLAG.(3:1) is clear then this is the   |   QDSTN
    |  | DST number of the target data segment.  If |
    |  | S is set, QADDR is DB relative.            |
    +--+--------------------------------------------+
   5| Offset in the data segment or system buffer   |   QADDR
    | table to the target data buffer.              |
    +-----------------------+-----------------------+
   6|                       | Function code for     |   QFUNC
    |                       | this request.  (See   |
    |                       | next section.)        |
    +-----------------------+-----------------------+
   7| On initiation, specifies the word count (>0)  |   QWBCT
    | or byte count (<0).  At completion of the     |
    | request this location contains the actual    |
    | transmission count in the same units (bytes   |
    | or words) as in the request.                  |
    +-----------------------------------------------+
 %10| Parameter 1.  Used only for reads.  Contains  |   QPAR1
    | the EOF specification in bits (13:3).         |
    +-----------------------------------------------+
 %11| Parameter 2.  Used only for writes.  If bit   |   QPAR2
    | (13:1) is set, writing past EOT is allowed.   |
    +-----------------------+--------------+--------+
 %12|         PCBN          |  QUALIFIER   | STATUS |   QSTAT
    +-----------------------+--------------+--------+
```

QFLAG - Request dependent flags

Bit 0 ABORT - Abort this request and return an error indication to the caller.
Bit 1 SPECIAL - Apply special handling to this request. (Not used)
Bit 2 DIAG - This is a request from the diagnostic subsystem. (Not used)
Bit 3 SYSBUFF - Target is an index relative to the SBUF Table of the data buffer.
Bit 4 IOWAKE - Wake caller on completion of request.
Bit 5 BLOCKED - Blocked I/O. The caller is waited in ATTACHIO until the request is completed. Implies IOWAKE.
Bit 6 COMPLETED - The request has been completed and the caller

awakened if he had requested (with IOWAKE).

Bit 7 DATAFRZN - Set by the memory management routines (MAM) when a MAKEPRESENT request is successfully completed and indicates the data segment is frozen in memory.
Bit 8 MAMERRORD - An error has occurred while MAM was trying to make the target data segment present and freeze it in memory.
Bit 9 PREQ - (Not used)
Bit 10 SFAIL - Delayed failure of SIO instruction. If a call to START'HPIB resulted in the request being added to the channel queue, this bit indicates that the SIO instruction failed when the request was selected for execution.
Bit 11 PFAIL - The request was aborted because of a system power failure.

QMISC - Driver request dependent flags and counters. Used mostly for error retries.

RETRY - Indicates an error retry is in progress.
BACK - Backspace record processing for an error retry is in progress.
FORWARD - Forward space record processing for an error retry is in progress.
GAP - Gap processing for an error retry is in progress.
BODEOF - Backspace record due to a data EOF processing is in progress.
TOUTCNTR - GIC timed-out counter.
FSCNTR - Forward space record counter.
BSCNTR - Backspace record counter.
RTCNTR - Error retry counter.

QSTAT - PCB number and request completion status.

PCBN - The Process Control Block (PCB) number of the process which made this request. If zero, the request is not associated with any process and the IOQ element is to be returned by the system when the request has completed.
STATUS - General status indicating the final state of the request. The following codes are used:
  0 - Not started or awaiting completion.
  1 - Successful completion.
  2 - End-of-file detected.
  3 - Unusual, but recoverable, condition detected.
  4 - Irrecoverable error has occurred.
QUALIFIER - A code which further defines or qualifies the general status. (See the section Driver Return Status Codes.)

SERIES II/III LINE PRINTER IOQ ELEMENT

```
       0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15     MNEMONIC
      +--+--+--+--+--+--+--+--+--+--+--+--+-----+--+--+
     0|     Request dependent flags (see below)       |   QFLAG
      +--+--+--+--+--+--+--+--+--+--+--+--+-----+--+--+
     1| SYSDB relative pointer to next IOQ element.   |   QLINK
      | Points to first word of element.              |
      +-----------------------+-----------------------+
     2| Physical unit number  | Logical device number |   QLDEV
      +--------------------+--+-----+-----+--+--+--+--+
     3|                    | WAITFLD|     |RT|MC|PS|PP|   QMISC
      +--+-----------------+--------+-----+--+--+--+--+
     4| S| If QFLAG.(3:1) is clear then this is the   |   QDSTN
      |  | DST number of the target data segment.  If |
      |  | S is set, QADDR is DB relative.            |
      +--+--------------------------------------------+
     5| Offset in the data segment or system buffer   |   QADDR
      | table to the target data buffer.              |
      +-----------------------+-----------------------+
     6|       Not used        | Function code for     |   QFUNC
      |                       | this request.  See    |
      |                       | next section.         |
      +-----------------------------------------------+
     7| On initiation, specifies the word count (>0)  |   QWBCT
      | or byte count (<0).  At completion of the     |
      | request this location contains the actual     |
      | transmission count in the same units (bytes   |
      | or words) as in the request.  The count is    |
      | truncated to produce a max of 256 characters. |
      +-----------------------------------------------+
   %10|   Parameter 1 of QFUNC.  See next section.    |   QPAR1
      +-----------------------------------------------+
   %11|   Parameter 2 of QFUNC.  See next section.    |   QPAR2
      +-----------------------+--------------+--------+
   %12|         PCBN          |  QUALIFIER   | STATUS |   QSTAT
      +-----------------------+--------------+--------+
```

QFLAG - Request dependent flags

Bit 0 .ABORT - Request has been aborted externally, either by the operator or a system intrinsic.
Bit 1 .SPECIAL - Not used.
Bit 2 .DIAG - Not used.
Bit 3 .SYSBUFRS - Target is a system-buffer-relative index to the data buffer. *
Bit 4 .IOWAKE - Wake caller on completion of request. *
Bit 5 .BLOCKED - Blocked I/O. The caller is waited in ATTACHIO until the request is completed. Implies IOWAKE. *
Bit 6 .COMPLETED - Request has been completed, and the caller awakened if s/he had requested (with IOWAKE). *

Bit 7 .DATAFRZN - If set, then the data segment has been made present and frozen in memory. Set by the memory management routines (MAM) when a MAKEPRESENT request is successfully completed. *

Bit 8 .MAMERRORD - An error has occurred while MAM was trying to make the target data segment present and freeze it in memory. *

Bit 9 .PREQ - Not used.

Bit 10 .SFAIL - Delayed failure of SIO instruction. If a call to STARTIO resulted in the request being added to the channel queue, this bit indicates that the SIO instruction failed when the request was selected for execution.

Bit 11 .PFAIL - The request was aborted because of a system power failure.

Bits12-13 .PREMPT - Not used.

Bit 14 . - Not used.

Bit 15 .MSGDONE - Not used.

QMISC.WAITFLD - This field contains a code describing the current idle state of the driver. The driver orients itself at each entry, based on the state of this field.

0 - The current entry is the start of a new request.

1 - The normal state while waiting for a completion interrupt of a print, fill or control operation.

2 - An SIO channel program was in progress when an asynchronous interrupt (usually an external abort) occurred, or a 2607 printer was placed on-line after going off-line while printing. The driver enters this state and waits for three seconds for the channel program or 2607 printer to complete, so as not to pose control conflicts to the U.I. card between the driver and the program.

3 - A Not Ready, Off Line or Paper Out (or Jammed) condition has been detected. The request will be continued or retried when the operator has corrected the condition and placed the printer on line.

4 - A 2607 (Tally) printer has come on-line after going off-line while printing. One line of data is buffered in the printer. This state causes the driver to shift to state 2 to allow the 2607 to print and space the buffered line before sending it the next line.

QMISC.(12:1) - RETRY (RT). Kludge to catch an LDEV configured as a 2608 when the physical device is a different subtype. Prevents Master Clear'ing and retrying a request more than once when the Power Fail/Reset device status bit is "set" by a non-2608.

QMISC.(13:1) - MASTER'CLEAR (MC). Set when a 2608 Master Reset, required because of a printer Power Fail/Reset, is configured and executed.

QMISC.(14:1) - PRESPACE (PS). The current operation is a prespace (space then print) request. This bit alerts the continuation section to fill the print buffer after spacing.

QMISC.(15:1) - PRE'TO'POST (PP). The previous request was a prespace operation while the current operation is a postspace.

QSTAT.PCBN - The Process Control Block (PCB) number of the process which made this request. If zero, the request is not associated with any process, and the IOQ element is to be returned by the system when the request has completed. *

QSTAT.STATUS - General status. Indicates the final state of the request. The following codes are used:

- 0 - Not started, or awaiting completion.
- 1 - Successful completion.
- 2 - Not used.
- 3 - Unusual, but recoverable, condition (such as Request Aborted Externally).
- 4 - Irrecoverable error (such as SIO failure, memory parity error, etc.).

QSTAT.QUALIFIER - A code which further defines or qualifies the general status.

2608 LINE PRINTER IOQ ELEMENT -- HPIB SYSTEMS

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
   +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  0|      Request dependent flags (see below)      |   QFLAG
   +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  1| SYSDB relative pointer to next IOQ element.   |   QLINK
   | Points to first word of element.              |
   +-----------------------+-----------------------+
  2|                       | Logical device number |   QLDEV
   +--+--+--+--------+-----+-----------+-----------+
  3|PP|PE|MC|TOUTCNTR|                 | WAITCODE  |   QMISC
   +--+--+--+--------+-----------------+-----------+
  4| S| If QFLAG.(3:1) is clear then this is the   |   QDSTN
   |  | DST number of the target data segment.  If |
   |  | S is set, QADDR is DB relative.            |
   +--+--------------------------------------------+
  5| Offset in the data segment or system buffer   |   QADDR
   | table to the target data buffer.              |
   +-----------------------+-----------------------+
  6|                       | Function code for     |   QFUNC
   |                       | this request.  (See   |
   |                       | next section.)        |
   +-----------------------+-----------------------+
  7| On initiation, specifies the word count (>0)  |   QWBCT
   | or byte count (<0).  At completion of the     |
   | request this location contains the actual     |
   | transmission count in the same units (bytes   |
   | or words) as in the request.                  |
   +-----------------------------------------------+
%10| Parameter 1.  Vertical Format specification.  |   QPAR1
   | (See next section for detail.)                |
   +-----------------------------------------------+
%11| Parameter 2.  Space Mode Flags.  (See next    |   QPAR2
   | section for details.)                         |
   +-----------------------+--------------+--------+
%12|         PCBN          |  QUALIFIER   | STATUS |   QSTAT
   +-----------------------+--------------+--------+
```

QFLAG - Request dependent flags

Bit 0 ABORT - Abort this request and return an error indication to the caller.
Bit 1 SPECIAL - Apply special handling to this request. (Not used)
Bit 2 DIAG - This is a request from the diagnostic subsystem. (Not used)
Bit 3 SYSBUFF - Target is an index relative to the SBUF Table of the data buffer.
Bit 4 IOWAKE - Wake caller on completion of request.
Bit 5 BLOCKED - Blocked I/O. The caller is waited in ATTACHIO until the request is completed. Implies IOWAKE.
Bit 6 COMPLETED - The request has been completed and the caller

awakened if he had requested (with IOWAKE).

Bit 7 DATAFRZN - Set by the memory management routines (MAM) when a MAKEPRESENT request is successfully completed and indicates the data segment is frozen in memory.

Bit 8 MAMERRORD - An error has occurred while MAM was trying to make the target data segment present and freeze it in memory.

Bit 9 PREQ - (Not used)

Bit 10 SFAIL - Delayed failure of SIO instruction. If a call to STARTIO resulted in the request being added to the channel queue, this bit indicates that the SIO instruction failed when the request was selected for execution.

Bit 11 PFAIL - The request was aborted because of a system power failure.

QMISC - Driver request dependent flags and counters.

PRE'TO'POST - Pre to post spacing change flag.

PEJECT - Last operation was a page eject.

MASTERCLR - Master clear done to clear powerfail bit in status. Master clear needs to be done from not ready conditon.

TOUTCNTR - Channel time-out retry counter.

WAITCODE - Indicates type of wait:
- 0 - new request
- 1 - completion wait
- 2 - not ready wait

QSTAT - PCB number and request completion status.

PCBN - The Process Control Block (PCB) number of the process which made this request. If zero, the request is not associated with any process and the IOQ element is to be returned by the system when the request has completed.

STATUS - General status indicating the final state of the request. The following codes are used:
- 0 - Not started or awaiting completion.
- 1 - Successful completion.
- 2 - End-of-file detected.
- 3 - Unusual, but recoverable, condition detected.
- 4 - Irrecoverable error has occurred.

QUALIFIER - A code which further defines or qualifies the general status. (See the section Driver Return Status Codes.)

2608 Line Printer Request Codes

Operation	Function	Parameters
WRITE	1	P1 - Vertical Format Specification 1 - use 1st data char as format spec %53 - "+", print and suppress spacing %55 - "-", print and triple space %60 - "0", print and double space %61 - "1", print and top of form %200-%277, print and space N-%200 lines %300-%377, print with channel N-%277 All others, print and single space. P2 - Space Mode Flags (15:1) - Prespace flag if set, print then fill buffer if clear, fill buffer then print (14:1) - No page stepover flag if set, single and double space without stepover (66 lines/page) if clear, single and double space with stepover (60 lines/page)
FILE OPEN	2	Page eject if not at top of form
FILE CLOSE	3	Page eject if not at top of form
DEVICE CLOSE	4	Page eject if not at top of form
READ STATUS	%17	Read I/O status Count - buffer must be at least 2 bytes
VFC SET	%100	Load VFC RAM Count - form length in words (0 loads RAM form internal ROM) P1 - 6 for 6 LPI or 8 for 8 LPI any other value defaults to 6 LPI
TAB SET	%101	Sets logical column definition P1 - 0 to 15, any other value defaults to 15

CIPER IOQ Element

<table>
<tr><th>Word #</th><th>0</th><th>1 2 3 4 5 6 7</th><th>8 9 10 11 12</th><th>13 14 15</th><th>MNEMONIC</th></tr>
<tr><td>0</td><td colspan="4">Request dependent flags (see below)</td><td>QFLAG</td></tr>
<tr><td>1</td><td colspan="4">SYSDB relative pointer to next IOQ element. Points to first word of element.</td><td>QLINK</td></tr>
<tr><td>2</td><td colspan="2"></td><td colspan="2">Logical device number</td><td>QLDEV</td></tr>
<tr><td>3</td><td colspan="4"></td><td>QMISC</td></tr>
<tr><td>4</td><td>S</td><td colspan="3">If QFLAG.(3:1) is clear then this is the DST number of the target data segment. If S is set, QADDR is DB relative.</td><td>QDSTN</td></tr>
<tr><td>5</td><td colspan="4">Offset in the data segment or system buffer table to the target data buffer.</td><td>QADDR</td></tr>
<tr><td>6</td><td colspan="2">Used by the new Disc routines for special status returns.</td><td colspan="2">Function code for this request. (See next section.)</td><td>QFUNC</td></tr>
<tr><td>7</td><td colspan="4">On initiation, specifies the word count (0) or byte count (<0). At completion of the request this location contains the actual transmission count in the same units (bytes or words) as in the request.</td><td>QWBCT</td></tr>
<tr><td>8</td><td colspan="4">Parameter 1.</td><td>QPAR1</td></tr>
<tr><td>9</td><td colspan="4">Parameter 2.</td><td>QPAR2</td></tr>
<tr><td>10</td><td colspan="2">PCBN</td><td>QUALIFIER</td><td>RSTATUS</td><td>QSTAT</td></tr>
</table>

QFLAG - Request dependent flags

Bit 0	ABORT	- Abort this request and return an error indication to the caller.
Bit 1	SPECIAL	- Apply special handling to this request. (Not used)
Bit 2	DIAG	- This is a request from the diagnostic subsystem.
Bit 3	SYSBUFF	- Target is an index relative to the SBUF Table of the data buffer.
Bit 4	IOWAKE	- Wake caller on completion of request.
Bit 5	BLOCKED	- Blocked I/O. The caller is waited in ATTACHIO until the request is completed. Implies IOWAKE.

Bit 6 COMPLETED - The request has been completed and the caller awakened if he had requested (with IOWAKE).

Bit 7 DATAFRZN - Set by the memory management routines (MAM) when a MAKEPRESENT request is successfully completed and indicates the data segment is frozen in memory.

Bit 8 MAMERRORD - An error has occurred while MAM was trying to make the target data segment present and freeze it in memory.

Bit 9 PREQ - (Not used)

Bit 10 SFAIL - Delayed failure of SIO instruction. If a call to STARTIO resulted in the request being added to the channel queue, this bit indicates that the SIO instruction failed when the request was selected for execution.

Bit 11 PFAIL - The request was aborted because of a system power failure.

QSTAT - PCB number and request completion status.

PCBN - The Process Control Block (PCB) number of the process which made this request. If zero, the request is not associated with any process and the IOQ element is to be returned by the system when the request has completed.

RSTATUS - General status indicating the final state of the request. The following codes are used:

0 - Not started or awaiting completion.
1 - Successful completion.
2 - End-of-file detected.
3 - Unusual, but recoverable, condition detected.
4 - Irrecoverable error has occurred.

QUALIFIER - A code which further defines or qualifies the general status.

General Status (13:3)	Qualifying Status (8:5)	Overall (8:8)
0 - Pending	1 - Waiting For Completion	%10
	3 - Not Ready Wait	%30
1 - Successful	0 - No Errors	%1
2 - End of File	(Not Used)	
3 - Unusual Condition	3 - Request Aborted	%33
	6 - Powerfail Abort	%63
	%21 - Device Powered Up	%213

4 - Irrecoverable Error	0 - Invalid Request	%4
	1 - Transfer Error	%14
	2 - I/O Timed Out Before Complete	%24
	4 - SIO Failure	%44
	5 - Unit Failure	%54
	%12 - System Error	%124
	%14 - Channel Failure	%144
	%21 - Parity Error	%214

2619A AND 2631 LINE PRINTER IOQ ELEMENT -- HPIB SYSTEMS

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   0|      Request dependent flags (see below)      |   QFLAG
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   1| SYSDB relative pointer to next IOQ element.   |   QLINK
    | Points to first word of element.              |
    +-----------------------+-----------------------+
   2|                       | Logical device number |   QLDEV
    +--+--+--+--------+-----+-----------+-----------+
   3|PP|PE|PF|TOUTCNTR|                 | WAITCODE  |   QMISC
    +--+--+--+--------+-----------------------------+
   4| S| If QFLAG.(3:1) is clear then this is the   |   QDSTN
    |  | DST number of the target data segment.  If |
    |  | S is set, QADDR is DB relative.            |
    +--+--------------------------------------------+
   5| Offset in the data segment or system buffer   |   QADDR
    | table to the target data buffer.              |
    +-----------------------+-----------------------+
   6|                       | Function code for     |   QFUNC
    |                       | this request.  (See   |
    |                       | next section.)        |
    +-----------------------+-----------------------+
   7| On initiation, specifies the word count (>0)  |   QWBCT
    | or byte count (<0).  At completion of the     |
    | request this location contains the actual    |
    | transmission count in the same units (bytes   |
    | or words) as in the request.                  |
    +-----------------------------------------------+
 %10| Parameter 1.  Vertical Format specification.  |   QPAR1
    | (See next section for detail.)                |
    +-----------------------------------------------+
 %11| Parameter 2.  Space Mode Flags.  (See next    |   QPAR2
    | section for details.)                         |
    +-----------------------+--------------+--------+
 %12|         PCBN          |  QUALIFIER   | STATUS |   QSTAT
    +-----------------------+--------------+--------+
```

QFLAG - Request dependent flags

Bit 0 ABORT - Abort this request and return an error indication to the caller.
Bit 1 SPECIAL - Apply special handling to this request. (Not used)
Bit 2 DIAG - This is a request from the diagnostic subsystem. (Not used)
Bit 3 SYSBUFF - Target is an index relative to the SBUF Table of the data buffer.
Bit 4 IOWAKE - Wake caller on completion of request.
Bit 5 BLOCKED - Blocked I/O. The caller is waited in ATTACHIO until the request is completed. Implies IOWAKE.
Bit 6 COMPLETED - The request has been completed and the caller

awakened if he had requested (with IOWAKE).

Bit 7 DATAFRZN - Set by the memory management routines (MAM) when a MAKEPRESENT request is successfully completed and indicates the data segment is frozen in memory.

Bit 8 MAMERRORD - An error has occurred while MAM was trying to make the target data segment present and freeze it in memory.

Bit 9 PREQ - (Not used)

Bit 10 SFAIL - Delayed failure of SIO instruction. If a call to STARTIO resulted in the request being added to the channel queue, this bit indicates that the SIO instruction failed when the request was selected for execution.

Bit 11 PFAIL - The request was aborted because of a system power failure.

QMISC - Driver request dependent flags and counters for 2631.

PRE'TO'POST - Pre to post spacing change flag.

PEJECT - Last operation was a page eject.

TOUTCNTR - Channel time-out retry counter.

POWERFAIL - Power fail flag indicates power fail occurred.

WAITCODE - Indicates type of wait:
  0 - new request
  1 - completion wait
  2 - not ready wait

Format for 2619A

```
 0  1  2  3  4                     12       15
------------------------------------------------
|PP|PE|PF|TO|BF|                   | WAITCODE  |
------------------------------------------------
```

TOUT - Channel timed out flag

BUF'FILL - Buffer fill operation in progress

QSTAT - PCB number and request completion status.

PCBN - The Process Control Block (PCB) number of the process which made this request. If zero, the request is not associated with any process and the IOQ element is to be returned by the system when the request has completed.

STATUS - General status indicating the final state of the request. The following codes are used:
  0 - Not started or awaiting completion.
  1 - Successful completion.
  2 - End-of-file detected.
  3 - Unusual, but recoverable, condition detected.
  4 - Irrecoverable error has occurred.

QUALIFIER - A code which further defines or qualifies the general status. (See the section Driver Return Status Codes.)

2619 Line Printer Request Codes

Operation	Function	Parameters
WRITE	1	P1 - Vertical Format Specification 1 - Use 1st data char as format specification. %53 - "+", print and suppress spacing %55 - "-", print and triple space %60 - "0", print and double space %61 - "1", print and top of form %200-%277, print and space N-%200 lines %300-%312, print with channel N-%277 %320 - Fill Line Printer Buffer Only All others, print and single space. P2 - Space Mode Flags (15:1) - Prespace flag if set, print then fill buffer if clear, fill buffer then print (14:1) - No page stepover flag if set, single and double space without stepover (66 lines/page) if clear, single and double space with stepover (60 lines/page)
FILE OPEN	2	Page eject if not at top of form
FILE CLOSE	3	Page eject if not at top of form
DEVICE CLOSE	4	Page eject if not at top of form
READ STATUS	%17	Read I/O status Count - buffer size
*IDENTIFY	%110	Return ID value in Bank & Buffaddr
*SELF TEST:		
INITIATE	%111	Subtest number to execute in Bank and Buffaddr (subtest number ranges from 0 to 7)
STATUS	%112	Subtest result returned in Bank & Buffaddr
*LOOPBACK TEST:		
WRT DATA	%113	Data to LP in Bank & Buffaddr [PING]
READ DATA	%114	Data from LP read into Bank & Buffaddr [PONG] Count - Buffer Size (256 bytes max)

2631 Line Printer Request Codes - HPIB

Operation	Function	Parameters
WRITE	1	P1 - Vertical Format Specification 1 - Use 1st data char as format specification. %53 - "+", print and suppress spacing %55 - "-", print and triple space %60 - "0", print and double space %61 - "1", print and top of form %200-%277, print and space N-%200 lines %300-%307, print with channel N-%277 %320 - Fill Line Printer Buffer Only All others, print and single space. P2 - Space Mode Flags (15:1) - Prespace flag if set, print then fill buffer if clear, fill buffer then print (14:1) - No page stepover flag if set, single and double space without stepover (66 lines/page) if clear, single and double space with stepover (60 lines/page)
FILE OPEN	2	Page eject if not at top of form
FILE CLOSE	3	Page eject if not at top of form
DEVICE CLOSE	4	Page eject if not at top of form
READ STATUS	%17	Read I/O status Count - 1 byte minumum required
VFC SET	%100	LOADS VFC RAM P1 - 1 - 1 LPI (lines per inch) 2 - 2 LPI 3 - 3 LPI 4 - 4 LPI 5 - 5 LPI 6 - 6 LPI 8 - 8 LPI 12 - 12 LPI Any other value defaults to 6 LPI.

SERIES III CARD READER IOQ

```
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
   0|                  (SEE BELOW)                  |     QFLAGS
    |-----------------------------------------------|
   1|         IOQP POINTER TO NEXT REQUEST          |     QLINK
    |-----------------------------------------------|
   2|      UNIT #           |   LOGICAL DEVICE #    |     QLDEV
    |-----------------------------------------------|
   3|             AUXILIARY BUFFER FLAG             |     QMISC
    |-----------------------------------------------|
   4|                DST NUMBER OR 0                |     QDSTN
    |-----------------------------------------------|
   5|            OFFSET IN DST OR BANK 0            |     QADDR
    |-----------------------------------------------|
   6|                       |     FUNCTION CODE     |     QFUNC
    |-----------------------------------------------|
   7|           WORD(+) OR BYTE(-) COUNT            |     QWBCT
    |-----------------------------------------------|
 %10|                                      |  EOF   |     QPAR1
    |-----------------------------------------------|
 %11|                             | BINARY |        |     QPAR2
    |-----------------------------------------------|
 %12|     PCB NUMBER        |  QUALIFIER   | STATUS |     QSTAT
    |-----------------------------------------------|

BIT0 ABORT
BIT1 SPECIAL
BIT2 DIAGNOSTIC
BIT3 SYS BUFFER
BIT4 IO WAKE
BIT5 BLOCKED
BIT6 COMPLETED
BIT7 DATA FREEZE
BIT8 MAM ERROR
BIT9 0
BIT10 SFAIL
BIT11 PFAIL
```

CARD READER IOQ (CONT.)

QFLAG - Flags and request state.

ABORT	Abort this request and return an error indication to the caller.
SPECIAL	Special handling is to be applied to this request. Has no meaning for card reader requests.
DIAGNOSTIC	This is a request from a diagnostic subsystem. Not used by card reader driver.
SYSBUFRS	Target is an index relative to the SBUF table of the data buffer.
IOWAKE	Wake caller on completion of request.
BLOCKED	Blocked I/O. The caller is waited in ATTACHIO until the request is completed. Implies wake.
COMPLETED	Request has been completed and caller woken if requested.
DATAFRZN	If set then the data segment has been frozen in memory. Set by MAM when a MAKEPRESENT request is successfully completed.
MAMERRD	An error has occurred in trying to make the target data segment present and freeze it in core.
SFAIL	SIO program failed to start because a) device didn't respond, or b) request has queued because device was busy.
PFAIL	This request has been aborted because of a power failure.

QLINK - SYSDB relative pointer to the next IOQ element. Points to the first word of the next element.

QLDEV - Logical device number.

QLDEVN Logical device number.

QMISC - Auxiliary buffer flag. When odd. Data is being read into an auxiliary buffer because the requested count is less than 40 words.

QDSTN - Contains the data segment number of the target data area.

QADDR - Offset to the target data area in the data segment or bank.

CARD READER IOQ (CONT.)

QFUNC - Function code. See ATTACHIO description for details.

FUNC Function code field.

0 - read
2 - file open (no operation)
3 - file close (no operation)
4 - device close (clear EOF field in LPDT)

QWBCT - Word or byte count and control returns. On initiation specifies a word count if positive or a byte count if negative. At completion of the request this location contains the actual transmission count in the same units as the call specified. Odd counts are rounded up to produce reads of an even number of bytes. All counts are truncated to produce maximum reads of 40 words for ASCII or 80 words for column binary.

QPAR1 - End of file specification. See EOFCHECK write up for details.

QPAR2 - Binary/ASCII specification.

BINARY If 0 then ASCII code conversion; 40 words maximum read.
If not 0 then column binary read; 80 words maximum read.

QSTAT - Request completion status and PCB number associated with this request.

PCBN PCB number associated with request. If zero this IOQ element is returned by the system when the request is completed.

STATUS General Status. See general IOQ entry for specifications.

QUALIFIER Driver specific status. See general IOQ entry.

IOQ ELEMENT FOR HPIB CARD READER

```
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
   +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  0|      Request dependent flags (see below)      |    QFLAG
   +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  1| SYSDB relative pointer to next IOQ element.   |    QLINK
   | Points to first word of element.              |
   +-----------------------+-----------------------+
  2|                       | Logical device number |    QLDEV
   +--+--+--+--+--+-----+--+-----+--------+--------+
  3|       Auxillary buffer flag.                  |    QMISC
   +--+--+--+--+--+-----+--------+--------+--------+
  4| S| If QFLAG.(3:1) is clear then this is the   |    QDSTN
   |  | DST number of the target data segment.  If |
   |  | S is set, QADDR is DB relative.            |
   +--+--------------------------------------------+
  5| Offset in the data segment or system buffer   |    QADDR
   | table to the target data buffer.              |
   +-----------------------+-----------------------+
  6|                       | Function code for     |    QFUNC
   |                       | this request.  (See   |
   |                       | next section.)        |
   +-----------------------+-----------------------+
  7| On initiation, specifies the word count (>0)  |    QWBCT
   | or byte count (<0).  At completion of the     |
   | request this location contains the actual     |
   | transmission count in the same units (bytes   |
   | or words) as in the request.                  |
   +-----------------------------------------------+
%10| Parameter 1.  Contains the EOF specification  |    QPAR1
   +-----------------------------------------------+
%11| Parameter 2.  Contains the data mode          |    QPAR2
   | specification in bits (11:2). (See below card |
   | reader request codes for detail information)  |
   +-----------------------+--------------+--------+
%12|         PCBN          |  QUALIFIER   | STATUS |    QSTAT
   +-----------------------+--------------+--------+
```

QFLAG - Request dependent flags

Bit 0	ABORT	- Abort this request and return an error indication to the caller.
Bit 1	SPECIAL	- Apply special handling to this request. (Not used)
Bit 2	DIAG	- This is a request from the diagnostic subsystem.
Bit 3	SYSBUFF	- Target is an index relative to the SBUF Table of the data buffer.
Bit 4	IOWAKE	- Wake caller on completion of request.
Bit 5	BLOCKED	- Blocked I/O. The caller is waited in ATTACHIO until the request is completed. Implies IOWAKE.
Bit 6	COMPLETED	- The request has been completed and the caller awakened if he had requested (with IOWAKE).

Bit 7 DATAFRZN - Set by the memory management routines (MAM) when a MAKEPRESENT request is successfully completed and indicates the data segment is frozen in memory.

Bit 8 MAMERRORD - An error has occurred while MAM was trying to make the target data segment present and freeze it in memory.

Bit 9 PREQ - (Not used)

Bit 10 SFAIL - Delayed failure of SIO instruction. If a call to STARTIO resulted in the request being added to the channel queue, this bit indicates that the SIO instruction failed when the request was selected for execution.

Bit 11 PFAIL - The request was aborted because of a system power failure.

QMISC - Auxillary buffer flag used to indicated a read into the driver's buffer and not the user's buffer.

QSTAT - PCB number and request completion status.

PCBN - The Process Control Block (PCB) number of the process which made this request. If zero, the request is not associated with any process and the IOQ element is to be returned by the system when the request has completed.

STATUS - General status indicating the final state of the request. The following codes are used:

0 - Not started or awaiting completion.
1 - Successful completion.
2 - End-of-file detected.
3 - Unusual, but recoverable, condition detected.
4 - Irrecoverable error has occurred.

QUALIFIER - A code which further defines or qualifies the general status. (See the section Driver Return Status Codes.)

CS 80 DISC IOQ ELEMENT (IOQ)

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
   +-----------------------------------------------+
  0|      Request dependent flags (see below)      |   QFLAG
   +-----------------------------------------------+
  1|      Request urgency class                    | QURGCLASS
   +-----------------------+-----------------------+
  2|      Unit#            | Logical device number |   QLDEV
   +-----+--+--+--+--+-----+--+--+-----+-----------+
  3|CHANF|RS|OP|IM|SR|RTRAN|LF|SP|     | WAITCODE  |   QMISC
   +-----+--+--+--+--+-----+--+--+-----+-----------+
  4| S|   DST   (If process disc I/O)              |   QDSCTN
   |- - - - - - - - - - - - - - - - - - - - - - - -|
   |      DST   (If segment transfer) [S=Stack]    |
   +--+--------------------------------------------+
  5| Offset in the data seg (If process disc I/O)  |   QADDR
   |- - - - - - - - - - - - - - - - - - - - - - - -|
   | Address in Bank (If segment transfer)         |
   +-----------------------+-----------------------+
  6|                       | Function code for     |   QFUNC
   |                       | this request.         |
   +-----------------------+-----------------------+
  7| On initiation, specifies the word count (>0)  |   QWBCT
   | or byte count (<0).  At completion of the     |
   | request this location contains the actual     |
   | transmission count in the same units (bytes   |
   | or words) as in the request.                  |
   +-----------------------------------------------+
%10| P1 - Parameter 1 (Usually High Order of       |   QPAR1
   | Current Logical Disc Address [CLDA1])         |
   +-----------------------------------------------+
%11| P2 - Parameter 2 (Usually Low Order of        |   QPAR2
   | Current Logical Disc Address [CLDA2])         |
   +-----------------------+--------------+--------+
%12|         PCBN          |  QUALIFIER   | STATUS |   QSTAT
   +-----------------------+--------------+--------+
%13| Sysbase relative indx of previous req in queue| QPREVREQP
   +-----------------------------------------------+
%14| Sysbase relative indx of next req in queue    | QNEXTREQP
   +-----------------------------------------------+
%15|        Segidentifier (If segment transfer     | QSEGIDENT
   +-----------------------------------------------+
%16| Displacement of read or wrt from seg base (MM)| QSEGDISP
   +-----------------------------------------------+
%17|S |////////////////////////////////////////////|
   |W |////////////////////////////////////////////|
   |A |////////////////////////////////////////////|
   |P |////////////////////////////////////////////|
   +-----------------------------------------------+
```

QFLAG - Request dependent flags

Bit 0 ABORT - Request has been aborted externally.
Bit 1 MMREQ - Request is for a segment transfer.
Bit 2 DIAG - This is a request from the diagnostic subsystem.
Bit 3 SBUF - Target is an index relative to the SBUF Table of the data buffer.
Bit 4 IOWAKE - Wake caller on completion of request.
Bit 5 BLOCKED - Blocked I/O. The caller is waited in ATTACHIO until the request is completed. Implies IOWAKE.
Bit 6 COMPLETED - The request has been completed and the caller awakened if he had requested (with IOWAKE).
Bit 7 DATAFRZN - Data segment has been present and is frozen.
Bit 8 MAMERRORD - An error has occurred while MAM was trying to make the target data segment present and freeze it in memory.
Bit 9 PREQUEUED - Request is queued into disc's request queue
Bit 10 SFAIL - Delayed failure of SIO instruction. If a call to STARTIO resulted in the request being added to the channel queue, this bit indicates that the SIO instruction failed when the request was selected for execution.
Bit 11 PFAIL - The request was aborted because of a system power failure.
Bit 12 CURREQ - Request is device's current request.
Bit 13 DISABLED - Request is disabled.
Bit 14 DISATMPT - Attempt to disable this request.
Bit 15 MSGDONE - A message request reply has completed.

QLDEV.QLDEVN - Logical Device Number

QMISC - Driver request dependent flags and counters.

CHAN'ERR'FLG - Channel error retry flag.
RSTAT'FAIL'FLG - Request status failed flag.
OPER'REQ'FLG - Operator requested release flag.
IM'FAULT'FLG - Internal maintenance fault flag.
STAT'RTRY'FLG - Status error single retry flag.
RTRANS'FLG - Retransmit required flag.
LOAD'FLG - Media load flag.
SYS'PFAIL'FLG - System powerfail flag.

WAITCODE - Indicates type of wait:

0 - new request
1 - completion wait
2 - not ready wait
3 - release/release deny wait
4 - IOQ defer wait
5 - DSCT read wait
6 - DSCT write wait
7 - synchronization wait

QDSTN - If system buffer is clear then this is the DST number of the target data segment. If bit 0 is set then buffer address is a DB offset value instead of segment relative offset (implemented for NOWAIT I/O and NOBUFF).

QADDR - Offset in data segment or system buffer table to target data buffer.

QFUNC - Function code and qualifiers as specified by driver.

QSTAT - PCB number and request completion status.

PCBN - The Process Control Block (PCB) number of the process which made this request. If zero, the request is not associated with any process and the IOQ element is to be returned by the system when the request has completed.

STATUS - General status indicating the final state of the request.

0 - Not started or awaiting completion.
1 - Successful completion.
2 - End-of-file detected.
3 - Unusual, but recoverable, condition detected.
4 - Irrecoverable error has occurred.

QUALIFIER - A code which further defines or qualifies the general status. (See the section Driver Return Status Codes.)

CS 80 INTEGRATED CARTRIDGE TAPE REQUEST IOQ ELEMENT

```
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
   +-----------------------------------------------+
  0|      Request dependent flags (see below)      |   QFLAG
   +-----------------------------------------------+
  1|      Request urgency class                    | QURGCLASS
   +-----------------------+-----------------------+
  2|      Unit#            | Logical device number |   QLDEV
   +-----+--+--+--+--------+--+--+-----+-----------+
  3|CHANF|RS|OP|IM| RETRY  |LF|SP|     | WAITCODE  |   QMISC
   +-----+--+--+--+--------+--+--+-----+-----------+
  4| S|   DST   (If process disc I/O)              |   QDSCTN
   |- - - - - - - - - - - - - - - - - - - - - - - -|
   |      DST   (If segment transfer) [S=Stack]    |
   +--+--------------------------------------------+
  5| Offset in the data seg (If process disc I/O)  |   QADDR
   |- - - - - - - - - - - - - - - - - - - - - - - -|
   | Address in Bank (If segment transfer)         |
   +-----------------------+-----------------------+
  6|                       | Function code for     |   QFUNC
   |                       | this request.         |
   +-----------------------+-----------------------+
  7| On initiation, specifies the word count (>0)  |   QWBCT
   | or byte count (<0).  At completion of the     |
   | request this location contains the actual     |
   | transmission count in the same units (bytes   |
   | or words) as in the request.                  |
   +-----------------------------------------------+
%10| P1 - Parameter 1 (Usually High Order of       |   QPAR1
   | Current Logical Disc Address [CLDA1])         |
   +-----------------------------------------------+
%11| P2 - Parameter 2 (Usually Low Order of        |   QPAR2
   | Current Logical Disc Address [CLDA2])         |
   +-----------------------+--------------+--------+
%12|         PCBN          |  QUALIFIER   | STATUS |   QSTAT
   +-----------------------+--------------+--------+
%13| Sysbase relative indx of previous req in queue| QPREVREQP
   +-----------------------------------------------+
%14| Sysbase relative indx of next req in queue    | QNEXTREQP
   +-----------------------------------------------+
%15|        Segidentifier (If segment transfer     | QSEGIDENT
   +-----------------------------------------------+
%16| Displacement of read or wrt from seg base (MM)| QSEGDISP
   +-----------------------------------------------+
%17|S |////////////////////////////////////////////|
   |W |////////////////////////////////////////////|
   |A |////////////////////////////////////////////|
   |P |////////////////////////////////////////////|
   +-----------------------------------------------+
```

QFLAG - Request dependent flags

Bit 0 ABORT - Request has been aborted externally.
Bit 1 MMREQ - Request is for a segment transfer.
Bit 2 DIAG - This is a request from the diagnostic subsystem.
Bit 3 SBUF - Target is an index relative to the SBUF Table of the data buffer.
Bit 4 IOWAKE - Wake caller on completion of request.
Bit 5 BLOCKED - Blocked I/O. The caller is waited in ATTACHIO until the request is completed. Implies IOWAKE.
Bit 6 COMPLETED - The request has been completed and the caller awakened if he had requested (with IOWAKE).
Bit 7 DATAFRZN - Data segment has been present and is frozen.
Bit 8 MAMERRORD - An error has occurred while MAM was trying to make the target data segment present and freeze it in memory.
Bit 9 PREQUEUED - Request is queued into disc's request queue
Bit 10 SFAIL - Delayed failure of SIO instruction. If a call to STARTIO resulted in the request being added to the channel queue, this bit indicates that the SIO instruction failed when the request was selected for execution.
Bit 11 PFAIL - The request was aborted because of a system power failure.
Bit 12 CURREQ - Request is device's current request.
Bit 13 DISABLED - Request is disabled.
Bit 14 DISATMPT - Attempt to disable this request.
Bit 15 MSGDONE - A message request reply has completed.

QLDEV.QLDEVN - Logical Device Number

QMISC - Driver request dependent flags and counters.

CHAN'ERR'FLG - Channel error retry flag.
RSTAT'FAIL'FLG - Request status failed flag.
OPER'REQ'FLG - Operator requested release flag.
IM'FAULT'FLG - Internal maintenance fault flag.
RETRY'COUNT - Retry count area.
LOAD'FLG - Media load flag.
SYS'PFAIL'FLG - System powerfail flag.

WAITCODE - Indicates type of wait:

0 - new request
1 - completion wait
2 - not ready wait
3 - release/release deny wait
4 - IOQ defer wait
5 - DSCT read wait
6 - DSCT write wait
7 - synchronization wait

QDSTN - If system buffer is clear then this is the DST number of the target data segment. If bit 0 is set then buffer address is a DB offset value instead of segment relative offset (implemented for NOWAIT I/O and NOBUFF).

QADDR - Offset in data segment or system buffer table to target data buffer.

QFUNC - Function code and qualifiers as specified by driver.

QSTAT - PCB number and request completion status.

PCBN - The Process Control Block (PCB) number of the process which made this request. If zero, the request is not associated with any process and the IOQ element is to be returned by the system when the request has completed.

STATUS - General status indicating the final state of the request.

0 - Not started or awaiting completion.
1 - Successful completion.
2 - End-of-file detected.
3 - Unusual, but recoverable, condition detected.
4 - Irrecoverable error has occurred.

QUALIFIER - A code which further defines or qualifies the general status. (See the section Driver Return Status Codes.)

```
SBUF AND TBUF TABLE LAYOUT
--------------------------

     |-----------------------------------------------|
     |      SECONDARY #      |       PRIMARY #       |
     |-----------------------------------------------|
     |  IMPEDED PROCESS PCB  |      ENTRY SIZE       |    TSIZE
     |-----------------------------------------------|
-----|                  HEAD INDEX                   |    THEAD
|    |-----------------------------------------------|
| ---|                  TAIL INDEX                   |    TTAIL
| |  |-----------------------------------------------|
| |  |   MAXIMUM OF IN USE   |    CURRENT IN USE     |    TUSE
| |  |-----------------------------------------------|
| |  |                   OVERFLOWS                   |    TOVRFL
| |  |-----------------------------------------------|
| |  |                     TOTAL                     |
| |  |                    REQUESTS                   |    TRQSTS
| |  |-----------------------------------------------|
| |  |                  INDEX OF 5                   |---------
| |  |-----------------------------------------------|        |
| | ->                                               |        |
| | ||                    ENTRY 1                    |        |
| | ||                                               |        |
| | ||-----------------------------------------------|        |
| | ||                       0                       |        |
| | ||-----------------------------------------------|        |
| -->|                                               |<---    |
|   ||                    ENTRY 2                    |   |    |
|   ||                                               |   |    |
|   ||-----------------------------------------------|   |    |
|   -|                  INDEX OF 1                   |   |    |
|    |-----------------------------------------------|   |    |
---->|                                               |   |    |
     |                    ENTRY 3                    |   |    |
     |                                               |   |    |
     |-----------------------------------------------|   |    |
     |                  INDEX OF 2                   |----    |
     |-----------------------------------------------|        |
     |                                               |        |
     |                    ENTRY 4                    |<---    |
     |                   (IN USE )                   |   |    |
     |-----------------------------------------------|   |    |
     |                  INDEX OF 4                   |----    |
     |-----------------------------------------------|        |
     |                                               |<--------
     |                    ENTRY 5                    |
     |                                               |
     |-----------------------------------------------|

3 - 1 - 5 - 4 - 2
```

TABLE ELEMENT ALLOCATION (TBUF AND SBUF)

The allocation of the elements in the IOQ terminal buffer (TBUF) and system buffer (SBUF) tables is of concern to the I/O system.

FREE LIST OF TABLE ELEMENTS

These tables are in the form of a free-linked list of the free elements. For the SBUF's the -1 word of entry is the link to the next element. For the TBUF's, word zero is the link and word 1 is the link for the IOQ elements.

Each word has an 8-word header beginning at the base of the table. The first four words of the header are for managing the table and the second four are for monitoring table activity.

The entries follow the header at word eight.

ELEMENT ALLOCATION

Elements are obtained from the beginning of the free list, pointed to by the head and returned to the end of the free list pointed by the tail.

When the free list is empty, the head index is zero and the tail index is set to point at the head index.

The tables are divided into two areas: a primary and a secondary area. Most requests are obtained from the primary area. The secondary area is used only for critical requirements when the primary area is exhausted. These areas are logical areas determined by parameters in the header.

The utility of the core resident tables is seriously reduced if their use is not restricted to dynamic situations.

One of three responses must be specified to the routines which allocate elements from the I/O system tables.

1. Impede caller if primary is empty.

2. Get from primary area only.

3. Get from secondary area if primary area is empty.

TABLE ELEMENT ALLOCATION (CONT.)

Request types 2 and 3 return an indication to the caller if the request could not be satisfied. The following table specifies the types of calls for element allocation and the action if an element is not activated.

BUFFER USER	CALL TYPE	FINAL ACTION
SBUF's		
File system	Impede	---
Ptape	Impede	---
Bad track	Primary	Forget request
TBUF's		
Terminal write (impedable)	Impede	---
Terminal write (not impedable)	Primary	I/O error
Terminal read on ICS	Secondary	I/O error
Log error	Primary	Forget request
IOQ's		
ATTACHIO (not impedable)	Primary	Return IOQX-0
ATTACHIO (impedable)	Impede	---
SIODM (memory management)	Secondary	Sudden death
IOMESSAGE	Secondary	I/O error

HEADER DEFINITION

Primary #	- Number of elements in the primary area.
Total #	- Total number of elements in the table.
Size	- Size in words of each element.
Impeded PCB	- If not zero then contains the PCB number of the first process waiting for an element in this table.
Head index	- Index of first free element.
Tail index	- Index of last free element.
In use	- Current number not in free list.
Overflows	- Number of requests made for an element.
Total requests	- Total number of elements requested.

ICS GLOBAL

```
QI -
      |--------------------|
    63.                    .
      .  RESERVED          .
    50.                    .
      |--------------------|
    49|  CANDPIN           |
      |--------------------|
    48|  LAST WEIGHT       |
      |--------------------|
    47|                    |
      |  PAUSETIME         |
    46|                    |
      |--------------------|
    45|  LISTSTATE         |
      |--------------------|
    44|  CUREFILTER        |
      |--------------------|
    43|  CURDFILTER        |
      |--------------------|
    42|  CWTNUM            |
      |--------------------|
    41|  CWTDENOM          |
      |--------------------|
    40|  CURCFILTER        |
      |--------------------|
    39|  MAXCFILTER        |
      |--------------------|
    38|  MINCFILTER        |
      |--------------------|
    37|  ESCHEDBASE        |
      |--------------------|
    36|  DSCHEDBASE        |
      |--------------------|
    35|  CSCHEDBASE        |
      |--------------------|
    34|  WORSTEPRI         |
      |--------------------|
    33|  WORSTDPRI         |
      |--------------------|
    32|  WORSTCPRI         |
      |--------------------|
    31| XDSEG Bank for PMBC|
      |--------------------|
    30| XDSEG Addr for PMBC|
      |--------------------|
    29| XDSEG lim  for PMBC|
      |--------------------|
    28| Status     for PMBC|
      |--------------------|
```

```
       |--------------------|
     27|                    |
       . RESERVED           .
     22|                    |
       |--------------------|      ----------
     21| PAUSETIME          | MPE III ONLY!
     20|                    | MPE III ONLY!
       |--------------------|
     19| PAUSECODE          | MPE III ONLY!
       |--------------------|
     18|       DISAP        |    PSEN, PSDB counter
       |--------------------|
     17|      Reserved      |
       |--------------------|
     16|        SDST        |    process' stack DST#
       |--------------------|
     15|        PSTA        |    pseudo-interrupt status
       |--------------------|
     14|       PADDR        |    pseudo-interrupt address
       |--------------------|
     13|     TRACE FLAG     |    flag set non-zero on IXIT away from ICS
       |--------------------|
     12|       PFAIL        |    PTR to powerfail PCB
       |--------------------|
     11|        JCUT        |    absolute JCUT address
       |--------------------|
     10|         XP         |    pointer to executing process PCB
       |--------------------|
      9|        PCBX        |    absolute stack address
       |--------------------|
      8|         Z          |    stack DB relative Z
       |--------------------|
      7|         DL         |    stack DB relative DL
       |--------------------|
      6|         S          |    stack DB relative S
       |--------------------|
      5|       SBANK        |    stack bank
       |--------------------|
      4|        STDB        |    absolute stack DB
       |--------------------|\
      3|         0          ||
       |--------------------||
      2|         P          ||
       |--------------------| >  DISPATCH stack marker
      1|       STATUS       ||
       |--------------------||
QI    0|P |      0          ||
       |--------------------|/
     +1|  DB BANK RETURN    |\
       |--------------------| > FOR DISPATCH
       |     DB RETURN      ||
       |--------------------|/
       |D |     PARM        |
       |--------------------|
```

P=PSEUDO-DISABLED AND DISP INSTRUCTION EXECUTED.
D=DISPATCHER INTERRUPTED.

-19 PAUSECODE(MPE III ONLY): 0 = system not paused 1 = paused for dis
2 = paused for swap 3 = system idle

ICS GLOBAL CELLS, with initial values

STDB - absolute address of the currently running process's stack.
SBANK - bank address for process' stack.
S - stack DB relativeS
DL - stack DB relative DL
Z - stack DB relative Z
PCBX - absolute stack address
XP - PCB table relative pointer to word 0 of the running process' PCB.

The above cells are to be initialized for the PROGENITOR.

CPCB - absolute 4, is an absolute version of XP. If CPCB is zero, then the above cells are invalid. This will never be the case in a process. CPCB should also be set by INITIAL.
SDST - DST# for running process' stack.
JCUT - the bank zero absolute address of the JCUT table.
PADDR - PB relative address for the procedure PSEUDOINT.
PSTA - status value for PSEUDOINT, %140000+CST#.
DISAP - PSDB counter, initially 0.

INITIAL sets the above as described.

CS 80 DISC Interrupt Linkage Table (ILT)

There is one ILT for each device controller configured on the system.
A controller may support more than one unit, however the CS'80 disc
driver will only concern itself with the single unit controller.

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
   +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  0|        Channel                                |   ICPVA0
  1|             Program                           |   ICPVA1
  2|                 Variable                      |   ICPVA2
  3|                      Area (ICPVA)             |   ICPVA3
   +-----------------------------------------------+
  4|        DMA Abort                              |   ICPVA4
  5|             Address                           |   ICPVA5
   +-----------------------------------------------+
  6|                      0                        |   ISRQL
   +--+--------------------+-----+-----------+--------+
  7|LI|      CHANQUE        |     |   CHAN    |  DEV   |   ICNTRL
   +--+--------------------+-----+-----------+--------+
%10| SYSDB relative pointer to channel program area|   ISIOP
   +-----------------------------------------------+
%11| SYSDB relative pointer to idle status area    |   ISTAP
   +-----------------------------------------------+
%12| single instruction that is executed to extract|   IUNIT
   | the device unit number from the status pointed|
   | to by ISTAP. [Since only Unit 0 exists on the |
   | CS'80 discs, ANDI 0 is used to return Unit 0] |
   +-----------------------------------------------+
%13| SYSDB relative DIT pointer of the device      |   ICDP
   | currently using the channel to perform a      |
   | data operation.                               |
   +-----------------------+-----------------------+
%14|       SIOPSIZE        |        CQUEN          |   IQUEUE
   +--+--+--+--------------+-----------+-----------+
%15|RW|WP|IG|                          |  HCUNIT   |   IFLAG
   +--+--+--+--------------------------+-----------+
%16| SYSDB relative DIT pointer for unit 0         |   IDITP0
   +-----------------------------------------------+
%17| 20 bytes status area for idle channel program |   ISTAT
   +-----------------------------------------------+
  .|                      .                        |
   +-----------------------------------------------+
  .|                      .                        |
   +-----------------------------------------------+
%31|             CS'80 Discs                       |
   .               Channel                         .
   |               Program                         |
   +-----------------------------------------------+
```

ICPVA0 - Channel Program Variable Area

The first word is used by the channel program processor to store status information after I/O channel aborts. The next word is used by the driver to indicate if status should be examined for special conditions or errors. The other two words are not used.

ICPVA4 - DMA abort address

If a DMA abort occurs, the absolute address where the abort occurred is stored in this area.

ICNTRL - Contains controller information

LIM - If this bit is set, the controller is sharing a software channel resource in order to limit bandwidth.
CHANQUE - The software channel resource number.
CHAN - Channel number (four most significant bits of DRTN).
DEV - Device number (three least significant bits of DRTN).

IQUEUE -

SIOPSIZE - (number of words + 1)/2 in the channel program area.
CQUEN - For a multi-unit controller this field contains the software controller resource number.

IFLAG - Controller and Channel Program state flags

RUNWAIT - An Idle Channel Program should be started when there are no active requests to process.
WAITPROG - An Idle Channel Program has been started for this controller. This bit is reset by an interrupt.
IGNOREHI - An HIOP instruction has been issued against this controller but the channel program was not in a wait statement. Therefore ignore the interrupt generated by the channel code when this program halts.
HCUNIT - Highest configured unit number for this controller.

ISTAT - 20 bytes of status from the idle channel program.

CHAPTER 14 SPOOLING

INPUT DEVICE DIRECTORY/OUTPUT DEVICE DIRECTORY

IDD/ODD (Common attributes referred to as XDD)

```
IDD:  DST = 45 (= %55)          ODD:  DST = 46 (= %56)
      SIR =  3                        SIR =  4
```

Overview of table structure

```
         +--------------------------------------+
         |   Entry 0  (8 words)                 |
         | . . . . . . . . . . . . . . . . . .  |
       2 |   Subentry area pointer              |  ---
         | . . . . . . . . . . . . . . . . . .  |    |
         |                                      |    |
         +--------------------------------------+    |
         |                                      |    |
         |   Head entries (4 words each)        |    |
         |                                      |    |
         +--------------------------------------+    |
         |                                      |  <--
         |   Subentries (%36 words each)        |
         |                                      |
         +--------------------------------------+
```

Entry 0 (overall table definitions)

```
   0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
 +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
0|      Maximum size     |     Current size      |0 (sectors)
 +-----------------------+-----------------------+
1| Head entry size = 4   | Subentry size = %36   |1 ( words )
 +-----------------------+-----------------------+
2|  Subentry area pointer (segment relative)     |2
 +--+--------------------------------------------+
3|DD|      Next avail device file ID (DFID)      |3
 +--+--------------------------------+-----------+
4|///////////////////////////////////|   Fence   |4
 +-----------------------------------+-----------+
5|///////////////////////////////////////////////|5
 +-----------------------------------------------+
6|///////////////////////////////////////////////|6
 +-----------------------------------------------+
7|///////////////////////////////////////////////|7
 +-----------------------------------------------+
```

DD: 0 ==> This is the IDD,
1 ==> This is the ODD.

Fence: For spooled output devices (ODD), the system-wide outfence. For spooled input devices (IDD), the jobfence.

Typical head entry (4 words)

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
+--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
|    Device outfence    |    Logical device     |
+-----------------------+-----------------------+
|                  Head pointer                 |
+-----------------------------------------------+
|                  Tail pointer                 |
+-----------------------------------------------+
|///////////////////////////////////////////////|
+-----------------------------------------------+
```

There are two types of head entry, a class entry and a logical device entry. There is only one class entry, if it exists at all, and it is the first head entry in the XDD. All spoofles opened by class (e.g., LP, SLOWLP, EPOC, PP, etc.) are linked to this entry. There is one logical device entry for each real (physical, as opposed to virtual) device on the system. Output devices appear in the ODD, input devices in the IDD. AC/DC devices such as terminals appear in both directories.

Each head entry is linked to 0 or more subentries (a typical subentry is shown in the next table). A null chain (0 subentries) consists of head pointer = 0 and tail pointer = segment-relative address of the associated head pointer. If one or more subentries exists, the pointers are segment-relative addresses of the first word of the first and last subentries of the chain. Any intermediate subentries are linked through the subentries. The tail subentry always contains a 0-link.

The Device Outfence and LDEV# fields are meaningless for the class entry. For logical device entries (non-0 Logical Device field), a non-0 Device Outfence means that this outfence overrides the system-wide outfence in word 4 of entry 0, but only for this device.

Typical subentry (%36 words)

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
   +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
  0|TL|State|  Outpri   |CL|         Device         |0
   +--+--+--+-----------+--+------------------------+
  1| Type|               Job number                 |1
   +-----+------------------------------------------+
  2|                                                |2
  3|                 User name                      |3
  4|                                                |4
  5|                                                |5
   +------------------------------------------------+
  6|                                                |6
  7|               Account name                     |7
 10|                                                |8
 11|                                                |9
   +------------------------------------------------+
 12|                                                |10
 13|                 Job name                       |11
 14|                                                |12
 15|                                                |13
   +------------------------------------------------+
 16|                                                |14
 17|                 File name                      |15
 20|                                                |16
 21|                                                |17
   +--+---------------------------------------------+
 22|IO|              Device file ID                 |18
   +--+--+-----------------+------------------------+
 23|FS|DA|/////////////////| Head index (see expl) |19
   +--+--+-----------------+------------------------+
 24|    Logical device     |  Sector address...     |20
   +-----------------------+                        |
 25|                          of spoofle label.     |21
   +------------------------------------------------+
 26|  Number of extents    |     Virtual LDEV       |22
   +-----------------------+------------------------+
 27|          Last extent size (sectors)            |23
   +--+--+--+--+--+--+--+---------------------------+
 30|SQ|//|RS|FD|SO|AB|//|     Number of copies      |24
   +--+--+--+--+--+--+--+---------------------------+
 31|   Segment-relative link to next subentry,      |25
   |   this device or class. 0 ==> last subentry.  |
   +------------------------------------------------+
 32|  Number of records in spoofle (doubleword)     |26
 33|                                                |27
   +------------------------------------------------+
 34|  Time (CLOCK intrinsic format) at which        |28
   |  spoofle was made READY, or 0D if not closed   |
 35|  properly (say, because of a system failure)   |29
   +------------------------------------------------+
```

not
MPE V

MPE V : 3:13

slightly diff in MPE

Word 0:	TL	-- A bit reserved for tape labels.
	State	-- State of subentry: 0 ==> Active 1 ==> Ready 2 ==> Open 3 ==> Locked
	CL	-- 1 ==> DEVICE field is a class index into the Device Class Table. 0 ==> DEVICE field is an LDEV number.
Word 1:	Type	-- Describes which environment created the subentry: 0 ==> Session' (SPOOK) 1 ==> Session 2 ==> Job 3 ==> Job' (SPOOK)
Word %22:	IO	-- 1 ==> Output DFID 0 ==> Input DFID
Word %23:	FS	-- There are one or more forms message requests in the spoofle.
	DA	-- The spoofle was created via a :DATA record (input spooling only).
Word %24:	LDEV	-- The logical device in class SPOOL where the file label (first extent) of the spoofle lives.
Word %26:	LDEV	-- LPDT index of virtual device LDEV. Simulates the properties of a real LDEV to the process which FOPENs a new (previously non-existing) spool file (State field (XDD(0).(1:2)) = 2 (Open)).
Word %30:	SQ	-- 1 ==> Squeeze (purge) spoofle extents as the final copy is printed. 0 ==> Purge only when final copy printed.
	RS	-- 1 ==> Restart job when warmstarting (input spooling only).
	FD	-- 1 ==> There are non-standard forms on the device.
	SO	-- Spaced Out bit. File System could not acquire a new extent when creating spoofle.
	AB	-- This is the $STDLIST of an aborted job.

Head index: The (segment-relative address)/4 of the head entry with which this subentry is linked. Since head entries are four words long, this can be thought of as an index into the head entry portion of the XDD -- if you disallow values of 0 and 1. Cute, huh?

SPOOK Output Tape Format

The overall format of output tapes produced by the SPOOK "OUTPUT" command is shown below. The various components of the tape are then described in detail. The format described here is subject to change as MPE evolves. Also, there may be errors in SPOOK which would cause the actual tape format to differ from the one described here in some cases. All numeric information is in integer format unless otherwise specified.

EOF

EOF

Label Record

EOF

File Directory Records

Device and Class Directory Record

EOF

Spoolfile

EOF

Spoolfile

EOF

.....

Mechanisms for End-of-tape and tape switching are the same as for STORE/RESTORE tapes.

Label Record

Words 0-13: "SPOOLFILETAPE LABEL-HP3000."

Word 23: reel number (first reel is number 1)

Word 24: date (from CALENDAR intrinsic)

Words 25&26: time (from CLOCK intrinsic)

30+31 "MPE V" if from MPE V

All other words are zero.

File Directory

The File Directory has one entry for each spoolfile on the tape. Each entry is 12 words, and entries are packed into as many 1020-word records as needed. The last record will be padded with zeros if necessary. The entry format is:

Word 0: Device file id number (bit 0 is on to indicate that the file is an output spoolfile)

Words 1-3: zero

Words 4-7: User name

Words 8-11: Account Name

Device and Class Directory

The Device and Class Directory is contained in one 1024-word record. There is no EOF separating this record from the File Directory. This directory contains one entry for each logical device or device class linked to the spoolfiles on the tape. Also, there is an entry for each logical device in each class in the directory, whether or not that logical device was directly referenced by a spoolfile. The entries are packed into the tape record one after another in no particular order. The entry formats are shown below.

Logical Device Entry

Word 0: logical device number

Word 1: Bits 0:8 : device subtype
Bits 8:8 : 3 (=length of this entry in words)

Word 2: device type

Device Class Entry

Word 0: Device class number (negated). This is the number of the entry of this device class in the system's Device Class Table.

Word 1: Total number of words in this entry.

Words 2 on: The entire contents of the Device Class Table entry for this device class.

There is one known bug in the Device and Class Directory. The last logical device in each class will be skipped when generating device entries for the members of the class. Unless that logical device is entered into the directory for some other reason, it will not be present.

apparently fixed in MPE V

Spoolfile Format

32 in MPE V

ODD entry (30-word tape record)

Spoolfile block ---> Two spoolfile blocks packed into one
Spoolfile block 1024-word tape record.

Two spoolfile blocks

Two spoolfile blocks

.....

The first few spoolfile blocks have been modified to contain user label information from the spoolfile. This is explained later.

Spoolfile Block Format

A spoolfile block is a 512-word block that contains variable length records in spooler format. The 2680 is intimately familiar with this structure. Any effort to change this format should be cleared with the 2680 project in Boise first! Spoolfile records start at the first word of the block. The last record is followed by a -1 to indicate that no more records follow. The last two words of the block contain a doubleword which is the record number of the first record in the block.

Spoolfile Record Format

Word 0: Byte count of record - 2

Word 1: Byte count of data portion of record. Note that this count includes trailing blanks. However, trailing blanks are truncated in the actual record, so this count may be more than the number of bytes actually present in the data portion.

Word 2: Function Code: 1=Fwrite
2=Fcontrol
3=Fopen
4=Fclose
%200 and beyond=Fdevicecontrol

Word 3: P1 -- ATTACHIO parameter

Word 4: P2 -- ATTACHIO parameter

Words 5 on: Data Portion of Record

User Labels Information

In the C-Mit and newer MPE versions, spoolfiles have a number of user labels with several kinds of information. These are:

1. Master: user label 0.
2. FOPEN entry catalog: user labels 1-10.
3. Circular queue for restart checkpointing: user labels 11-27.

Since older versions of MPE did not use user labels, a way was needed to incorporate them into the SPOOK tape format without losing forward and backward compatibility. The method used is to add several special spoolfile blocks to the beginning of the spoolfile on tape. Each of these blocks has exactly one FOPEN record at its beginning. This record is followed by a -1. Thus old versions of MPE will assume that the rest of the block is garbage. However, the rest of the block is actually used to contain user label information. The first two spoolfile blocks (i.e. the first tape record of the spoolfile proper) contain only the FOPEN records. The next 5 tape records actually contain user labels in addition to the FOPEN records. The user labels are packed 3 to a spoolfile block, 6 to a tape record. Each spoolfile block of 512 words has the following format:

Words 0-4: FOPEN record

Word 5: -1 (to "terminate" the block)

Words %200-%377: user label

Words %400-%577: user label

Words %600-%777: user label

Following this special group of blocks, the spoolfile resumes a normal format. The special FOPEN records all have the number of user labels in P2.

It is often the case that some of the 27 user labels have not been initialized before the tape is written. In that case, their places will be filled with garbage. There is no easy way of detecting this except by careful inspection.

Since user labels are written 6 to a tape record and there are 28 user labels, the last %400 words of the final user label tape record are always uninitialized.

REPLY INFORMATION TABLE (RIT)

DST %34; SIR %25

```
%|----------------------------------------------|
0|             NUMBER OF ENTRIES                | \
 |----------------------------------------------| |                    !
1|            MAX NUMBER OF ENTRIES             | |                    !
 |----------------------------------------------| |                    !
2| POSITION OF NEXT FREE ENTRY SPACE IN QUEUE   | | TABLE  57          !
 |----------------------------------------------| | HEADER wd          !
3|          NUMBER OF QUEUED ENTRIES            | |                    !
 |----------------------------------------------| |                    !
 | (52 WORDS TO HOLD PIN#'s OF QUEUED ENTRIES)  | |                    !
 |----------------------------------------------| |                    !
 |                  UNUSED                      | |                    !
 |==============================================| /
0|            PROCESS NUMBER (PIN)              |           \
 |----------------------------------------------|           |
1|              DST# (FOR REPLY)                |           |
 |----------------------------------------------|           |
2|        BUFFER ADDRESS (DST RELATIVE)         |           |
 |----------------------------------------------|           |
3| MAX LENGTH OF STRING | REPLY TYPE EXPECTED   |           |
 |----------------------------------------------|           |
4|                                              |           |
 |----------------------------------------------|           |
5|                                              |           |
 |----------------------------------------------|           |
6|                                              |           | ENTRY
 |----------------------------------------------|           | (51    !
7|             # BYTES IN MESSAGE               |           |  wds)
 |----------------------------------------------|           |
 |                                              |           |
 |                                              |           |
 |               MESSAGE IN ASCII               |           |
 |                                              |           |
 |               (UP TO 86 CHARS.)              |           |        !
 |                                              |           |
 |                                              |           |
 |                                              |           /
 |----------------------------------------------|
```

NOTE: Process Number = 0 means entry is empty
  Reply Type = 0 for number (num)
             = 1 for yes or no (y/n)
             = 2 for string (sxx)
             = 3 for yes, no, or STRING
             = 4 for string   !
  TABLE SIZE = 2046 words   !
  MAX # OF ACTIVE ENTRIES = 39   !
  MAX # OF QUEUED ENTRIES = 52   !

The message system consists of the following parts:
- Callable intrinsic GENMESSAGE.
- Uncallable procedure GENMSG which is used by MPE.
- System message catalog (CATALOG.PUB.SYS) and any number of user catalogs.
- Program MAKECAT which builds message catalogs.
- MESSAGE SIR %24
- MESSAGE SYSGLOB CELLS %371-373
- MESSAGE DATA SEGMENT

The message system is used by calling GENMESSAGE (or GENMSG) with a message number. The message system fetches the message from a message catalog, inserts parameters, then routes the message to a file or returns the message in a buffer to the caller.

A message catalog is a numbered editor-type file containing sets of messages. The sets serve to break a catalog into managable portions. A message system user may call GENMESSAGE using either his own message catalog or using MPE's catalog (CATALOG.PUB.SYS).

After creating a message file, run the program MAKECAT in order to build a catalog that is readable by the message system. This file is still readable by the editor (it can be "texted") but it contains a directory (written as a userlabel).

In order to use the message catalog, the program must first open the message catalog, then call GENMESSAGE with the file number, set number and message number. (MPE users don't need to open the catalog, GENMSG automatically uses CATALOG.PUB.SYS.) The file must be opened with the aoptions "NOBUF" and "MULTI" -record access.

MESSAGE CATALOG

Messages in the catalog can be of any length and can contain up to five parameters. Continuation of a message is indicated by "%" or "&" at the end of a line. The "%" symbol indicates that the message is continued and that a carriage return, line feed be issued the terminal. The "&" symbol indicates that the message is continued on the same line with no carriage return, line feed.

Parameters may be inserted into the message fetched from the catalog. The parameters are passed in the GENMESSAGE (or GENMSG) call and inserted wherever a "!" is found. Message sets are indicated by "$SET n" starting in column 1 (the rest of the line is treated as a comment). Maximum value for n is 63. Comments can be inserted in the catalog by placing "$" in column 1. Message numbers are positive integers, need not be contiguous, but must be in ascending order. After processing by the program MAKECAT, the catalog file contains records of 80 bytes, blocked 16, in 32 extents. (The system message catalog is only one extent, however). The format of the message catalog is as follows:

MESSAGE SYSTEM (CONT.)

```
$SET 1    SYSTEM MESSAGES
1 LDEV #! IN USE BY FILE SYSTEM
2 LDEV #! IN USE BY DIAGNOSTICS
3 LDEV IN USE, DOWN PENDING
5 IS "!" ON LDEV#! (Y/N)?
    .
    .
    .
$ MESSAGE 35 IS TWO LINES LONG, A PARAMETER STARTS THE
$ FIRST LINE AND THE SECOND LINE IS "HP32002"
35 !%
HP32002B.00.!
    .
    .
    .
276 LDEV # FOR "!" ON ! (NUM)!
$
$SET 2 CIERROR MESSAGES
82 STREAM FACILITY NOT ENABLED:  SEE OPERATOR.  (CIERR 82)
200 MORE THAN 30 PARAMETERS TO BUILD COMMAND.  (CIERR 200)
    .
    .
    .
204 FILE COMMAND REQUIRES AT LEAST TWO PARAMETERS, INCLUDING
THE
 FORMAL NAME OF THE FILE  (CIERR 204)
    .
    .
```

MAKECAT PROGRAM

The program MAKECAT.PUB.SYS is used to build message catalogs (and also HELP catalogs). The program's input file has the formaldesignator INPUT, which must be used for all entry points. The program has the following entry points:

(no entry point)	- Reads from input file and builds a temporary file (formaldesignator CATALOG). Also renames any old temporary CATALOG, CATnn, using an archival numbering scheme (i.e., CAT1, CAT2, etc.).
BUILD	- (Must log on under MANAGER.SYS.) Reads from input file, build the system message catalog (formaldesignator CATALOG), and installs the message system. Existing catalog is renamed CATnnnn according to the same scheme as for no entry point (above). Installation of the message system means moving the directory contained in the userlabel of the catalog into a data segment. The DST number and the disc address of CATALOG are placed in system global area. The message system may be installed while the system is running.

DIR	- (Must have PM or OP capability.) Installs the system message catalog (does not build a new one). Opens input file, moves the directory in the CATALOG into a data segment, and places the DST number and disc address of CATALOG in system global area. This may be done when the message system seems to be "broken", but the catalog is intact. (MPE is issuing "MISSING MSG. SET=mm. MSG=nn" at terminals and at the console.) This may be done while the system is running.
HELP	- Used to build the HELP catalog. Reads input file and builds a HELP catalog (formaldesignator HELPCAT).

```
MESSAGE SYSTEM              CATALOG.PUB.SYS
--------------              ---------------
```

$SET 1 - System messages.
$SET 2 - CI errors and warnings messages.
$SET 3 - Miscellaneous ABORT messages.
$SET 4 - Program error abort messages.
$SET 5 - Intrinsics abort messages.
$SET 6 - Run-time abort messages.
$SET 7 - CI general messages.
$SET 8 - File System error messages.
$SET 9 - Loader error messages.
$SET 10 - CREATE error messages.
$SET 11 - ACTIVATE error messages.
$SET 12 - SUSPEND error messages.
$SET 13 - MYCOMMAND error messages.
$SET 14 - LOCKGLORIN error messages.
$SET 15 - Private Volumes error messages.
$SET 16 - DS/3000 messages.
$SET 17 - HELP facility error messages.
$SET 18 - Graphic devices messages.
$SET 19 - Serial Disc error messages.
$SET 20 - User Logging error messages.
$SET 21 - Association Utility (ASOCTABL) messages.
$SET 22 - 2680A Page Printer messages.
$SET 25 - 2680A Page Printer error file messages.
$SET 26 - Disc Free Space messages.
$SET 27 - System Internal Error messages.
$SET 28 - CIPER Device Error messages.

MESSAGE SET DIRECTORY

```
   DST # IN SYSGLOB %373

   CAT DISC ADDR IN SYSGLOB %371-372

CREATED BY RUNNING MAKECAT.PUB.SYS.
KEPT IN A DATA SEGMENT AND IN A USER LABEL.

%              DATA SEGMENT                 #
  |-------------------------------------|
0 |         MAX. SET #                  |0 \          \
  |-------------------------------------|  | HEADER   |
1 |     # OF MESSAGE RECORDS            |1 /          |
  |-------------------------------------|             |
2 |  RECORD OFFSET TO FIRST MESSAGE     |2 \          |
  |-------------------------------------|  | SET 1    | USER
3 |        FIRST MESSAGE #              |3 /          | LABEL
  |-------------------------------------|             |
4 |  RECORD OFFSET TO FIRST MESSAGE     |4 \          |
  |-------------------------------------|  | SET 2    |
5 |        FIRST MESSAGE #              |5 /          |
  |-------------------------------------|             |
  |                                     |             |
  ~        EMPTY ENTRY                  ~             |
  ~                                     ~             |
  |                                     |             |
  |-------------------------------------|             |
50|  RECORD OFFSET TO FIRST MESSAGE     |40\          |
  |-------------------------------------|  | SET 63   |
51|        FIRST MESSAGE #              |41/          |
  |-------------------------------------|             |
52|              0                      |42\          |
  |-------------------------------------|  | CUR MSG  |
53| RECORD OFFSET TO CURRENT MESSAGE    |43/          |
  |-------------------------------------|            /
54|           MESSAGE                   |44
  |            BUFFER                   |
  ~          (640 WORDS)                ~
  ~                                     ~
  |                                     |
  |-------------------------------------|
1253                                   683

  EMPTY ENTRY:
  |-------------------------------------|
  | RECORD OFFSET OF NEXT IN-USE SET    |
  |-------------------------------------|
  |               -1                    |
  |-------------------------------------|
```

```
HELP SUBSYSTEM          HELP DIRECTORY
--------------          --------------

KEPT AS USER LABEL
READ ONTO USER'S STACK
USES SEARCH INTRINSIC FORMAT
VARIABLE ENTRY SIZE


%
 |--------------------------------------------|
0|       DIRECTORY SIZE (WORDS)               |
 |--------------------------------------------|
1| ENTRY LGTH (BYTES) | KEYWORD LGTH (BYTES) | \
 |--------------------------------------------| |
2|               ENTRY                        | |
 |              KEYWORD                       | |  ENTRY
 ~                                            ~ |
 ~              1-255 BYTES                   ~ |
 |                                            | |
 |--------------------------------------------| |
 |     ENTRY RECORD # IN CICAT                | |
 |   LEFT BYTE          |     RIGHT BYTE      | /
 |--------------------------------------------|
 | ENTRY LGTH (BYTES) | KEYWORD LGTH (BYTES) | \
 |--------------------------------------------| |
 |               ENTRY                        | |
 ~              KEYWORD                       ~ |  ENTRY
 ~             1-255 BYTES                    ~ |
 |                      |---------------------| |
 |                      | ENTRY REC # LEFT BYTE| |
 |--------------------------------------------| |
 |ENTRY REC # R. BYTE | ENTRY LGTH (BYTES)    | / \
 |--------------------------------------------|    |
 |KEYWORD LGTH (BYTES)|                       |    |
 |---------------------                       |    |
 |               ENTRY                        |    |
 ~              KEYWORD                       ~    |  ENTRY
 ~             1-255 BYTES                    ~    |
 |                                            |    |
 |--------------------------------------------|    |
 |             ENTRY REC #                    |    |
 |   LEFT BYTE          |     RIGHT BYTE      |    /
 |--------------------------------------------|
 |                                            |
 ~                                            ~
 ~                                            ~
 |                                            |
 |--------------------------------------------|
```

*EXTRA DATA SEGMENT - DST # IN DB+%250 OF UMAIN STACK

*BUILT BY INITUDC

```
 0  1  2  3      6 7 8               15
|---------------------------------------|
|LT|LN|NH|NB|    |TY |    ENTRY SIZE    | \ LT-OPTION LIST
|---------------------------------------| | LN-OPTION LOGON
|      HEADER RECORD NUMBER             | | NH-OPTION NOHELP
|---------------------------------------| | NB-OPTION NOBREAK
|        BODY RECORD NUMBER             | | TY- 00=USER UDC
|---------------------------------------| |     01=ACCOUNT UDC
|    FILE NUMBER      |  COMMAND LENGTH | |     10=SYSTEM UDC
|---------------------------------------| |
|                                       | > ENTRY
|                 COMMAND               | |
~                   NAME                ~ |
~                (1-16 BYTES)           ~ |
|                                       | |
|                                       | /
|---------------------------------------|
|                                       |
|                                       |
~                 ENTRIES               ~
~                                       ~
|                                       |
|                                       |
|---------------------------------------|
|                    |         0        |  ENTRY SIZE=0 ENDS
|---------------------------------------|         DIRECTORY
```

UDC'S   COMMAND.PUB.SYS

*RECORD SIZE = 20(10) WORDS, 6 RECORDS/BLOCK

*KEEPS TRACK OF WHO IS USING WHAT UDC CATALOG

*CAN BE PURGED TO DISABLE UDC'S

*CAN BE REBUILT TO REENABLE UDC'S

```
 %      RECORD 0      #          %     FREE ENTRY      #
  |-----------------|             |-----------------|
 0|1st FREE ENTRY # |0           0|NEXT FREE ENTRY #|0
  |-----------------|             |-----------------|
 1|    not used     |1           1|  ENTRY TYPE=0   |1
  |-----------------|             |-----------------|
 2|   MAX IN USE    |2           2|                 |2
  |-----------------|             |                 |
 3|    # IN USE     |3            ~     not used    ~
  |-----------------|             ~                 ~
 4|                 |4            |                 |
  |    not used     |             |                 |
  ~                 ~             |                 |
  ~                 ~             |                 |
  |                 |             |                 |
23|                 |19         23|                 |19
  |-----------------|             |-----------------|
```

COMMAND.PUB.SYS (cont.)

```
 %   USER ENTRY       #          %   FILE ENTRY       #
   |-----------------|             |-----------------|
  0| CATALOG ENTRY # |0           0|NEXT CAT. ENTRY #|0
   |-----------------|             |-----------------|
  1|   ENTRY TYPE=0  |1           1|  ENTRY TYPE = 2 |1
   |-----------------|             |-----------------|
  2|                 |2           2|                 |2
   |      USER*      |             |   FILE NAME     |
  3|                 |3           3|                 |3
   |                 |             |  FOPEN FORMAT:  |
  4|                 |4           4|                 |4
   |                 |             |                 |
  5|                 |5           5|                 |5
   |-----------------|             |                 |
  6|                 |6           6|      FILE       |6
   |                 |             |                 |
  7|     ACCOUNT*    |7           7|   [/LOCKWORD]   |7
   |                 |             |                 |
 10|                 |8          10|      GROUP      |8
   |                 |             |                 |
 11|                 |9          11|     ACCOUNT     |9
   |-----------------|             |                 |
 12|                 |10         12|        0        |10
   |                 |             |                 |
 13|    not used     |11         13|                 |11
   |                 |             |                 |
 14|                 |12         14|                 |12
   |                 |             | (UP TO 36 BYTES)|
 15|                 |13         15|                 |13
   |                 |             |                 |
 16|                 |14         16|                 |14
   |                 |             |                 |
 17|                 |15         17|                 |15
   |                 |             |                 |
 20|                 |16         20|                 |16
   |                 |             |                 |
 21|                 |17         21|                 |17
   |                 |             |                 |
 22|                 |18         22|                 |18
   |                 |             |                 |
 23|                 |19         23|                 |19
   |-----------------|             |-----------------|
```

* IF THE USER FIELD AND THE ACCOUNT FIELD CONTAIN "@_______", THIS INDICATES SYSTEM LEVEL UDC'S.

IF ONLY THE USER FIELD CONTAINS @ AND 7 SPACES, THIS INDICATES ACCOUNT LEVEL UDC'S.

CI STACK DEFINITION

```
                 ---------------------------------------
DB+%0           |  BCOMIMAGE (Byte Ptr. To Command)     |
                |---------------------------------------|
DB+%1           |           COMMAND IMAGE               |
               \|             (280 bytes)              \|
                \                                        \
                |\                                       |\
                |---------------------------------------|
DB+%215         |           LINELENSTACK                |
               \|            (30 words)                \|
                \                                        \
                |\                                       |\
                |---------------------------------------|
DB+%253         |    NEXTMSG (Not currently used)       |
                |---------------------------------------|
DB+%254         |          THIS IS SPARE                |
                |---------------------------------------|
DB+%255         |              UDC0                     |
                |---------------------------------------|
DB+%256         |              UDC1                     |
                |---------------------------------------|
DB+%257         |              UDC2                     |
                |---------------------------------------|
DB+%260         |              UDC3                     |
                |---------------------------------------|
DB+%261         |              UDC4                     |
                |---------------------------------------|
DB+%262         |            IFNESTING                  |
                |---------------------------------------|
DB+%263         |             IFSKIP                    |
                |---------------------------------------|
DB+%264         |            ELSESEEN                   |
                |---------------------------------------|
DB+%265         |            CIFLAGS                    |
                |---------------------------------------|
DB+%266         |      CONTINUE STATE STACK             |
                |           (2 words)                   |
                |---------------------------------------|
DB+%270         |          PENDINGCOMLEN                |
                |---------------------------------------|
DB+%271         |    BLASTCOMIMAGE (Byte Ptr.)          |
                |---------------------------------------|
DB+%272         |       LAST COMMAND IMAGE              |
               \|          (280 bytes)                 \|
                \                                        \
                |\                                       |\
                |---------------------------------------|
```

Field Definitions

BCOMIMAGE: Byte pointer to COMIMAGE (sometimes called WCOMIMAGE) in the CI stack.

COMMAND IMAGE: Command character string currently being executed.

LINELENSTACK: A CI command can span up to 30 input lines. This stack holds the length of each input line.

NEXTMSG: Used to be used to link messages together. No longer being used.

THIS IS SPARE: Not used.

UDC0: Holds the DST number of the UDC definitions.

UDC1: Holds the old S register value for UDCs.

UDC2: (0:1)--FLUSHUDC, used by :SETCATALOG

UDC3: UDC options for current UDC.

UDC4: (0:1)--UDC Fatal Ci Error
(1:1)--UDC EXITBREAK
(2:1)--UDC BREAKDETECTED
(3:1)--UDC NOPRINT
(4:1)--UDC IMAGEADJUST
(10:6)--UDC NESTLEVEL

IFNESTING: Level of nesting of :IF commands.

IFSKIP: Whether the current commands are being skipped as the false part of a :IF command.

ELSESEEN: Level of the :ELSE commands.

CIFLAGS: (13:1)--Sequenced: line numbers at rear.
(15:1)--Not REDOable (last command).

CONTINUE STATE STACK: History of the :CONTINUE commands.
= 0--no :CONTINUE
= 1--just seen
= 2--in effect.

PENDINGCOMLEN: If <> 0, command is already in stack and this word is the command string length.

BLASTCOMIMAGE: Byte pointer to last command image.

LAST COMMAND IMAGE: When a command completes execution, the command string is copied here for use by the :REDO command.

ASSOCIATION DST LAYOUT

```
|==========================================|
|                                          |   0     DST %42
|                                          |   1
|                  Not                     |   2     SIR %30
|                                          |   3
|                  Used                    |   4
|                                          |   5     One entry/
|                                          |   6     system ldev
|==========================================|
|     Not Used       | JMAT Index (8 bits) |   7    \
|------------------------------------------|        |
|   Not Used     |   JIT Index (10 bits)   |   8    |
|------------------------------------------|        |
| DST rel. index to user's next entry.     |   9    |- Ldev 1
|------------------------------------------|        |
                                                    (Associated)
|   Class name under which this ldev is    |  10    |
|   associated.  Left justified and        |  11    |
|   padded with blanks.  8 bytes.          |  12    |
|                                          |  13    /
|==========================================|
|                   0                      |  14    \
|------------------------------------------|        |
|                   0                      |  15    |
|------------------------------------------|        |
|                   0                      |  16    |- Ldev 2
|------------------------------------------|        |
                                                    (Unassociated
                                                    )
|                                          |  17    |
|                 Don't                    |  18    |
|                 Care                     |  19    |
|                                          |  20    /
|==========================================|
|                                          |

.                   .                      .
.                   .                      .
.                   .                      .

|                                          |
|==========================================|
|            JMAT Index or 0               |  7*n   \
|------------------------------------------|        |
|             JIT Index or 0               |        |
|------------------------------------------|        |
|          Next Entry Pointer or 0         |        |- Ldev n
|------------------------------------------|        |
|                                          |        |
|                Classname                 |        |
|              or Don't Care               |        |
|                                          |        /
|==========================================|
```

CTABO (Memory Size Independent Configuration Values)

RECORD 0 OF CONFDATA FILE

```
    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
   0|              MEMORY SIZE IN K WORDS            |0
    |------------------------------------------------|
   1|                 CORE SIZE INDEX                |1
    |------------------------------------------------|
   2|               STANDARD STACK SIZE              |2
    |------------------------------------------------|
   3|                  HIGHEST DRT #                 |3
    |------------------------------------------------|
   4|            TERMINAL BOUND PRIORITY             |4
    |------------------------------------------------|
   5|                 NORMAL PRIORITY                |5
    |------------------------------------------------|
   6|               CPU BOUND PRIORITY               |6
    |------------------------------------------------|
   7|            # OF SECONDS TO LOG-ON              |7
    |------------------------------------------------|
  10|        LOG FILE RECORD SIZE (SECTORS)          |8
    |------------------------------------------------|
  11|            LOG FILE SIZE (RECORDS)             |9
    |------------------------------------------------|
  12|                  LOG FILE #                    |10
    |------------------------------------------------|
  13|            LOG BITS (ONLY 11 USED)             |~~11~~—— % רוףף
    |                                                |
  14|                                                |12
  15|        <<DEFINES WHAT IS BEING LOGGED>>        |13
  16|                                                |14
    |                                                |
  17|                                                |15
    |------------------------------------------------|
  20|      DEFAULT JOB/SESSION CPU TIME LIMIT        |16
    |------------------------------------------------|
  21|                 FILES DUMPED                   |17
    |------------------------------------------------|
  22|           HIGHEST LOGICAL DEVICE #             |18
    |------------------------------------------------|
  23| HIGHEST VOLUME #    | HIGHEST SYS. VOLUME NO.  |19
    |------------------------------------------------|
  24|            DEVICE CLASS TABLE SIZE             |20
    |------------------------------------------------|
```

RECORD 0 (CONT.)

```
  |---------------------------------------------|
25|                  FIX LEVEL                  |21
  |---------------------------------------------|
26|               COLD LOAD COUNT               |22
  |---------------------------------------------|
27|          MAX INITIAL SEGMENT SIZE           |23
  |---------------------------------------------|
30|         DISC COLD LOAD ENTRY POINT          |24
  |---------------------------------------------|
31|          SIZE OF OLD VOLUME TABLE           |25
  |---------------------------------------------|
32|   SIZE OF OLD COLD LOAD INFORMATION TABLE   |26
  |---------------------------------------------|
33|         TIME QUANTUM  (unused)              |27
  |---------------------------------------------|
34|          MAXIMUM OPEN SPOOL FILES           |28
  |---------------------------------------------|
35|                 CSTAB SIZE                  |29
  |---------------------------------------------|
36|                                             |30
  |   MAXIMUM # OF SPOOL FILES (KILO SECTORS)   |
37|                                             |31
  |---------------------------------------------|
40|         # OF ADDITIONAL CS DRIVERS          |32
  |---------------------------------------------|
41|         # SECTORS PER SPOOL EXTENT          |33
  |---------------------------------------------|
42|                UPDATE LEVEL                 |34
  |---------------------------------------------|
43|                  VERSION                    |35
  |---------------------------------------------|
44|              SERIAL DISC LOAD         |FD|SD|36
  |---------------------------------------------|
45|                 MIT VERSION                 |37
  |---------------------------------------------|
46|                 MIT UPDATE                  |38
  |---------------------------------------------|
47|                  MIT FIX                    |39
  |---------------------------------------------|
50|DR|TP|      reserved for system             |40
  |------        conversion id's.               |
51|          All bits currently unused          |41
  |               must be zero.                 |
52|                                             |42
  |                                             |
53|                                             |43
  |---------------------------------------------|
```

DR   0 = 7-bit DRT system   TP   0 = word 4 of CTAB is no. of TBUFS
     1 = 9-bit DRT system   1 = word 4 of CTAB is TBUFS
                                       per port

RECORD 0 (CONT.)

```
   |------------------------------------------------|
54|           TAPE RECORD SIZE (WORDS)               |44
   |------------------------------------------------|
  ~////////////////////////////////////////////////~.
  ~////////////////////////////////////////////////~.
  ~////////////////////////////////////////////////~.
   |------------------------------------------------|
177|                   RESERVED 124(10)-127(10)      |127
   |------------------------------------------------|
```

SERIAL DISC LOAD (Word %44)
   FD - Date given for sysdump was future date
   SD - Sysdump was to serial disc

CTAB (Memory Size Dependent Configuration Values)

RECORDS 1-8 OF CONFDATA FILE

record		memory size k words
1	-	64
2	-	80
3	-	96
4	-	128
5	-	160
6	-	192
7	-	224
8	-	256 and larger

This table describes the CTAB format in detail and is typical of any record (1-8)

```
  |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 0|             # OF CST ENTRIES                  |0
  |-----------------------------------------------|
 1|             # OF DST ENTRIES                  |1
  |-----------------------------------------------|
 2|             # OF PCB ENTRIES                  |2
  |-----------------------------------------------|
 3|             # OF IOQ ENTRIES                  |3
  |-----------------------------------------------|
 4|     # OF TERMINAL BUFFERS PER PORT            |4
  |-----------------------------------------------|
 5|         # OF CST ENTENSION ENTRIES            |5
  |-----------------------------------------------|
 6|    INTERRUPT CONTROL STACK SIZE (Q1 to Z1)    |6
  |-----------------------------------------------|
 7|        # UCOP REQUEST QUEUE ENTIRES           |7
  |-----------------------------------------------|
10|           # BREAKPOINT ENTRIES                |8
  |-----------------------------------------------|
11|               # TRL ENTRIES                   |9
  |-----------------------------------------------|
12|               # LOCAL RINS                    |10
  |-----------------------------------------------|
13|               # GLOBAL RINS                   |11
  |-----------------------------------------------|
14|               # OF SYSTEM BUFFERS             |12
  |-----------------------------------------------|
15|               # OF CONCURRENT PROGS           |13
  |-----------------------------------------------|
16|               # OF MAM TABLE ENTRIES          |14
  |-----------------------------------------------|
17|     reserved for type-ahead buffer size       |15
  |-----------------------------------------------|
```

RECORDS 1-8 (CONT.)

```
  |-----------------------------------------------|
20|///////////////////////////////////////////////|16
  |-----------------------------------------------|
  ~///////////////////////////////////////////////~.
  ~///////////////////////////////////////////////~.
  |-----------------------------------------------|
24|///////////////////////////////////////////////|20
  |-----------------------------------------------|
25|           DIRECTORY SIZE (SECTORS)            |21
  |-----------------------------------------------|
  ~///////////////////////////////////////////////~.
  ~///////////////////////////////////////////////~.
  ~///////////////////////////////////////////////~.
  |-----------------------------------------------|
36|           MAXIMUM CODE SEGMENT SIZE           |30
  |-----------------------------------------------|
37|      MAXIMUM # OF CODE SEGMENTS/PROCESS       |31
  |-----------------------------------------------|
40|         MAXIMUM STACK SIZE (MAXDATA)          |32
  |-----------------------------------------------|
41|       MAXIMUM EXTRA DATA SEGMENT SIZE         |33
  |-----------------------------------------------|
42|   MAXIMUM # OF EXTRA DATA SEGMENTS/PROCESS    |34
  ~///////////////////////////////////////////////~.
  ~///////////////////////////////////////////////~.
  ~///////////////////////////////////////////////~.
  |-----------------------------------------------|
50|          MAXIMUM # RUNNING SESSIONS           |40
  |-----------------------------------------------|
51|          MAXIMUM # OF RUNNING JOBS            |41
  |-----------------------------------------------|
52|                # LOG PROCS                    |42
  |-----------------------------------------------|
53|                  LOG ID's                     |43
  |-----------------------------------------------|
54|         # DISC REQUEST TABLE ENTRIES          |44
  |-----------------------------------------------|
55|         # SPECIAL REQUEST TABLE ENTRIES       |45
  |-----------------------------------------------|
56|         # PRIMARY MESSAGE TABLE ENTRIES       |46
  |-----------------------------------------------|
57|         # SWAP TABLE ENTRIES                  |47
  |-----------------------------------------------|
60|         # SECONDARY MESSAGE TABLE ENTRIES     |48
  |-----------------------------------------------|
```

DRIVER TABLE

The Driver Table consists of 6 word entries, in correspondence to the LDEV entries, up to the highest LDEV used, entry zero is a dummy entry.

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|          DRT #           |      UNIT #        |
|--|--------|--------|--|--|--------------------|
|CR| CHAN # |        |DS|       MASTER LDEV     |
|--|--------|--------|--|-----------------------|  TYPICAL ENTRY
|          D            |           R           |  FORMAT
|-----------------------|-----------------------|
|          I            |           V           |
|-----------------------|-----------------------|
|          N            |           A           |
|-----------------------|-----------------------|
|          M            |           E           |
|-----------------------|-----------------------|
```

DS	DS DEVICE (if set DRT is zero)
CR	CORE RESIDENT
CHAN #	CHANNEL #
MASTER LDEV	LDEV of device which this DS device is linked to.

Words 2-6 contain the driver name.

SYSDUMP FORMAT

```
|-----------------------------------------| <---TAPE LOAD POINT
|   READ - SIO - PROGRAM  PROGRAM         |
|-----------------------------------------| <---SERIAL DISC LOAD POINT
|            SIO PROGRAM                  |
|-----------------------------------------|
|                    ICS                  |
|-----------------------------------------|
|                LOW CORE                 |
|-----------------------------------------|
|                   TCST                  |
|-----------------------------------------|
|                CS TABLE                 |
|-----------------------------------------|
|            DRIVER TABLE                 |
|-----------------------------------------|
|                   LPDT                  |
|-----------------------------------------|
|                    LDT                  |
|-----------------------------------------|
|              DEVICE CLASS TABLE         |
|-----------------------------------------|
|                   LDTX                  |
|-----------------------------------------|
|                   VTAB                  |
|-----------------------------------------|
|                  OLDVTAB                | *
|-----------------------------------------|
|  DISC COLD LOAD INFORMATION TABLE       | *
|-----------------------------------------|
|                   CTAB                  |
|-----------------------------------------|
|                  CTAB0                  |
|-----------------------------------------|
|                  CSDVR                  |
|-----------------------------------------|
|                  CSDEF                  |
|-----------------------------------------|
|            INITIAL'S DB AREA            |
|-----------------------------------------|
|               STACK MARKER              |
|-----------------------------------------|
|           INITIAL'S SEGMENTS            |
|-----------------------------------------|
|                RIN TABLE                | *
|-----------------------------------------|
* NOT DUMPED IF DATE =CARRIAGE RETURN
```

SYSDUMP FORMAT (CONT.)

```
|----------------------------------------|
|      LOGGING IDENTIFIER TABLE          |  *
|----------------------------------------|
|          DIRECTORY HEADER              |  *
|----------------------------------------|
|             DIRECTORY                  |  *
|----------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXX|
|----------------------------------------|
| SYSTEM PROGRAMS, SL, NON-STD. DRIVERS  |
|----------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXX|
|----------------------------------------|
|        STORE/RESTORE HEADER            |
|----------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXX|
|----------------------------------------|
|       STORE/RESTORE DIRECTORY          |  *
|----------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXX|
|----------------------------------------|
|   USER FILES (SEPARATED BY "EOF's"     |  *
|----------------------------------------|
|        STORE/RESTORE TRAILER           |
|----------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXX|
|----------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXX|
|----------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXX|
|----------------------------------------|
```

* NOT DUMPED IF DATE = CARRIAGE RETURN

NOTE: ON DISC, READ-SIO-PROGRAM KEPT IN DISC LABEL.

STORE TAPE FORMAT

FIRST VOLUME

```
|-------------------------------------------|
|XXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXXXXXX|
|-------------------------------------------|
|XXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXXXXXX|
|-------------------------------------------|
|           "STORE/RESTORE LABEL -          |0     \
|                 HP/3000."                 |13    |
|-------------------------------------------|      |
|                  "VIIB"                   |14    |
|                                           |15    |
|-------------------------------------------|      |
|        PARTIAL FIRST FILE FLAG            |16    |
|-------------------------------------------|      |
|                 CHECKSUM                  |17    |
|-------------------------------------------|      |
|        DIRECTORY INDEX OF FIRST FILE      |18    |
|-------------------------------------------|      | HEADER
|                                           |19    |  40 WORDS
|                                           |      |
|                                           |      |
|                                           |22    |
|-------------------------------------------|      |
|               VOLUME NUMBER               |23    |
|-------------------------------------------|      |
|                   DATE                    |24    |  DATE:
|-------------------------------------------|      |   0:7 last 2 digits
|                   TIME                    |25    |       of year
|                                           |26    |   7:9 Julian date
|-------------------------------------------|      |
|  TAPEBLOCKSIZE (#WORDS/BLOCK;def=4096)    |27    |  TIME:
|-------------------------------------------|      |   25.(0:8) hours
|                                           |28    |      (8:8) minutes
|                                           |      |   26.(0:8) seconds
|                                           |      |      (8:8) .1 secs.
|                                           |      |
|                                           |39    /
|-------------------------------------------|
```

FIRST VOLUME (CONT.)

```
|-----------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXXX|
|-----------------------------------------|
|                    .                    |             \
|                    .                    |             |
|                    .                    |             |
|-----------------------------------------|             |
|                FILE NAME                |\            |
|-----------------------------------------||TYP FILE    |
|               GROUP NAME                || ENTRY      | VOLUME
|-----------------------------------------||(12 WDS.)   | DIRECTORY:
|               ACCT. NAME                |/            |  # ENTRIES
|-----------------------------------------|             | DETERMINED
|                    .                    |             | BY TAPEBLOCK-
|                    .                    |             | SIZE
|                    .                    |             /
|-----------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXXX|
|-----------------------------------------|
|                                         |             \
|      FILES  (separated by "EOF's")      |             | FILES
|                                         |             |
~                                         ~             /
```

STORE FORMAT

SUBSEQUENT VOLUMES

```
|----------------------------------------|
|        "STORE/RESTORE LABEL-           |0             \
|              HP/3000."                 |13            |
|----------------------------------------|              |
|                "VIIB"                  |14            |
|                                        |15            |
|----------------------------------------|              |
|         PARTIAL FIRST FILE FLAG        |16 FLAG=1:    |
|----------------------------------------|   1st FILE   |
|                CHECKSUM                |17 ON THIS    |
|----------------------------------------|   VOL IS A   |
|      DIRECTORY INDEX OF FIRST FILE     |18 PARTIAL.   |   HEADER
|----------------------------------------|              |    40 WDS.
|                                        |19            |
|                                        |22            |
|----------------------------------------|              |
|             VOLUME NUMBER              |23            |
|----------------------------------------|              |
|                 DATE                   |24            |
|----------------------------------------|              |
|                 TIME                   |25            |
|                                        |26            |
|----------------------------------------|              |
|             TAPEBLOCKSIZE              |27            |
|----------------------------------------|              |
|                                        |28            |
|                                        |39            / NOTE: NO EOF.
|========================================|
|                   .                    |              \
|                   .                    |              |
|                   .                    |              |
|----------------------------------------|              |
|               FILE NAME                |\             |
|----------------------------------------||TYPICAL      |
|               GROUP NAME               || FILE         | VOLUME
|----------------------------------------|| ENTRY        | DIRECTORY
|               ACCT NAME                |/             |
|----------------------------------------|              |
|                   .                    |              |
|                   .                    |              |
|                   .                    |              /
|----------------------------------------|
|XXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXXXXX|
|----------------------------------------|
|                                        |              \
|                <FILES>                 |              | FILES
|         (separated by "EOF's)          |              |
~                                        ~              /
```

STORE FORMAT

```
            ------------
            END OF VOLUME
            -------------

~                                         ~             \
|                                         |             |
|              <FILES>                    |             |
|        (separated by "EOF's)            |             | FILES
|                                         |             |
|                                         |             /
|-----------------------------------------|
XXXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXXXX|
|-----------------------------------------|
|     "STORE/RESTORE LABEL-HP/3000."      |0            \
|                                         |13           |
|-----------------------------------------|             |
|                                         |14           |
|                                         |             |
|                                         |20           |
|-----------------------------------------|             |
|   FLAG: PRECEDING EOF MARKS FILE ENDED  |21           | TRAILER
|-----------------------------------------|             |
|  FLAG: PRECEDING EOF MARKS TAPESET ENDED|22           |  40 WDS.
|-----------------------------------------|             |
|              VOLUME NO.                 |23           |
|-----------------------------------------|             |
|                  DATE                   |24           |
|-----------------------------------------|             |
|                  TIME                   |25           |
|                                         |26           |
|-----------------------------------------|             |
|                                         |27           |
|                                         |             |
|                                         |39           /
|-----------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXXXX|
|-----------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXXXX|
|-----------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXXXX|
|-----------------------------------------|
```

LABELED TAPE SUBSYSTEM

The MPE labeled tape subsystem permits convenient access to tapes labeled to either ANSI or IBM standards. It operates as a set of subprocedures to the file system.

A labeled tape consists of one or more logical files. Each logical file consists of three physical files, i. e. tape areas delimited by tapemarks. The first physical file contains header labels, the second contains the data, and the third contains trailer labels which are (except for minor differences) copies of the header labels. The tape mark following trailer labels will be followed either by header labels for the next file, or by another tapemark if there is no next file.

Format of MPE Tape Labels

Labels are 80 bytes long, and conventionally are identified by their first four characters (three letters and a digit) and contain information as follows (CP := character position; L:= length):

VOL1: Present only on the first file of a volume, the volume label contains the volume identifier, which is usually the number on the tape strap, and is thus not expected to be changed.

CP	Field Name	L	Content
1/3	Label identifier	3	"VOL"
4	Label Number	1	"1"
5/10	Volume Identifier	6	Vol ID
11	Accessibility	1	"0" if IBM, else " "
12/79	Not used	62	Blanks
80	Label-Standard Version	1	"1" if H-P ANSI else " "

UVLn: User volume labels. May be present on tapes from foreign shops, but are not written by MPE. If encountered, they are ignored.

HDR1: First header label. Required for each file. Specifies:

CP	Field Name	L	Content
1/3	Label identifier	3	"HDR"
4	Label Number	1	"1"
5/21	File Identifier	17	File name, if tape was not written by MPE, only the first eight are significant.
22/27	Volume Set Ientifier	6	Names the volume on which the set of files begins
28/31	Reel Number	4	Counts the reels that contain this file (1 starts)
32/35	File sequence number	4	Counts the files in the set of files (1 starts)
36/41	Not Used	6	MPE writes blanks
42/47	Creation Date	6	Year and day within year when the file was written.
48/53	Expiration Date	6	Year and day within year when the file may be overwritten without permission.
54	Accessibility	1	%230 if Lockword, "0" if IBM
55/60	Block count	6	Number of blocks if IBM.
61/73	System Code	13	"HP MPE 3000 "
74/80	Not Used	7	Blanks

HDR2: Second header label. Although defined by the standard, may be missing on foreign tapes. Contains:

CP	Field Name	L	Content
1/3	Label identifier	3	"HDR"
4	Label Number	1	"2"
5	Record Format	1	"F" = Fixed "V" = Variable "U" = Undefined Others treated as Undefined
6/10	Block Length	5	Block length (in character format).
11/15	Record Length	5	Record length (adhering to to MPE rules) in characters.
16/23	Lockword	8	MPE File Lockword.
24/36	Not Used	13	MPE writes blanks
37	Record Type	1	"A" = ASCII "B" = Binary.
38	Carriage Control	1	"C" = control " " = no control.
39/80	Not Used	42	Blanks

IBM: IBM has a slightly different format. It is:

CP	Field Name	L	Content
1/3	Label identifier	3	"HDR"
4	Label Number	1	"2"
5	Record Format	1	"F" = Fixed "V" = Variable "U" = Undefined Others treated as Undefined
6/10	Block Length	5	Block length (in character format).
11/15	Record Length	5	Record length (adhering to to MPE rules) in characters.
16	Not Used	1	Blank.
17	IBM Position	1	"0" = no volume switch "1" = a switch has occurred.
18/38	Not Used	11	Blanks.
39	IBM Block Attribute.	1	"B" = Blocked records. "S" = Spanned records. "R" = Blocked and Spanned. " " = No blocked or spanned.
40/80	Not Used	41	Blanks

User header labels: optional. Standard prescribes UHLn in the first four characters, but MPE doesn't care.

EOV1: End of Volume; used as first trailer label. Required if the logical file is continued onto another reel. Identical to HDR1, except contains the number of physical blocks of data in the data area.

CP	Field Name	L	Content
1/3	Label identifier	3	"EOV"
4	Label Number	1	"1"
5/54	Same as HDR1	50	
55/60	Block Count	6	Number of data blocks since last beginning of file section label group.
61/80	Same as HDR1	20	

EOV2: Defined by the standard, but may be missing on foreign tapes. Follows EOV1; format same as HDR2.

EOF1: End of File; used as first trailer label. Required if this is the end of the logical file. Format same as EOV1.

EOF2: Same as EOV2 except used after EOF1.

User trailer labels: optional. Standard prescribes UTLn in the first four characters, but MPE again doesn't care.

TAPE LABEL TABLE

The tape label table is the private playground of the tape label subsystem. It consists of two parts: LDEV Control Blocks (LCBs) and Volume Control Blocks (VCBs). The LDEV area is set up at system initialization and contains one entry for each magnetic tape LDEV and serial disc device in the system. As is common in MPE, the first entry is a dummy which tells where the other things in the table are. The volume area contains one entry for each labeled tape volume requested or active on the system.

Although table entries are stored in an extra data segment, they are generally manipulated via local copies on the stack. The procedures GETLDEV and GETFNUM look for LDEV and volume entries as specified; they copy them to stack buffers and return the DST address for use in copying them back. POSTVTENT copies the entries back, and in the case of a new volume entry, allocates space for it in the volume section of the tape label table.

Tape Label Table Header Entry

During PROGEN, SETUP'TAPES is called to initialize the table. The overall structure of the initialized TLT is:

```
TLTDST -- %32,#26                 TLTSIR -- %47,#39

  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
-----------------------------------------------------------------
|      Table initialization word (=1 when initialized)         |  0
|--------------------------------------------------------------|
|      Entry size (ESIZE) = %32,#26                            |  1
|--------------------------------------------------------------|
|   Table relative pointer to base of LCB entries (LTBASE) (1) |  2
|--------------------------------------------------------------|
|   Table relative pointer to base of VCB entries (VTBASE) (2) |  3
|--------------------------------------------------------------|
|   Table relative pointer to top of Volume table (VTTOP) (3)  |  4
|--------------------------------------------------------------|
|      Size of Tape Label Table, in words (VTMAX)              |  5
|--------------------------------------------------------------|
|                                                              |  6
|                                                              |
|                                                              |  7
|                                                              |
|                                                              | 10
|                                                              |
~                         not used                             ~
~                                                              ~
|                                                              |
|                                                              | 30
|                                                              |
|                                                              | 31
|--------------------------------------------------------------|
~                                                              ~
```

```
|                                                                 | 32
|                                                                 | <-(1)
~      LDEV Control Block area -- one entry/mag tape drive        ~
~                                                                 ~
|                                                                 |
|                                                                 |
|-----------------------------------------------------------------|
|                                                                 | <-(2)
|                                                                 |
~      Volume Control Block table -- contains VCB entries         ~
~                                        and free entries         ~
|                                                                 |
|                                                                 |
|-----------------------------------------------------------------|
~                                                                 ~
|                                                                 |
|                                                                 |
|-----------------------------------------------------------------|
```

Unintialized Table (INITIAL)

INITIAL will build the "uninitialized" TLT as follows:

```
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
----------------------------------------------------------------
|     Size of the table, in words (always > 1)                 |  0
|--------------------------------------------------------------|
|     Number of LDEVS in the table = X                         |  1
|--------------------------------------------------------------|
|         LDEV#                  |                         |T  |  2
|--------------------------------------------------------------|
|                                                              |
~             Total of LDEVS (X) entries of above              ~
~                                                              ~
|                                                              |
|--------------------------------------------------------------|
|         LDEV#                  |                         |T  |  X+2
|--------------------------------------------------------------|
|                                                              |
~           Expansion area                                     ~
~             during SETUP'TAPES                               ~
|                                                              |
----------------------------------------------------------------
```

T: 1 if Tape drive
   0 if not Tape drive (ie. serial disc)

LCB Entry Format

The LCB entries have the following structure:

```
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
----------------------------------------------------------------
|               | Type  | T | L | B | HP|                      |  0
|--------------------------------------------------------------|
|                    Logical device number                     |  1
|--------------------------------------------------------------|
|                         VCB address                          |  2
|--------------------------------------------------------------|
|                         Reel number                          |  3
|--------------------------------------------------------------|
|                    File sequence number                      |  4
|--------------------------------------------------------------|
|                        Creation date                         |  5
|--------------------------------------------------------------|
|                       Expiration date                        |  6
|--------------------------------------------------------------|
|                                                              |  7
|                                                              |
|                          File name                           | 10
|                                                              |
~                                                              ~
~                                                              ~
|                                                              |
|                                                              | 16
|                               +------------------------------|
|                               |                              | 17
+--------------------------------------------------------------+
|                                                              | 20
|                                                              |
|                                                              | 21
|                          (not used)                          |
|                                                              | 22
|                                                              |
|                                                              | 23
|--------------------------------------------------------------|
|                                                              | 24
|                                                              |
|                    Volume set identifier                     | 25
|                                                              |
|                                                              | 26
|--------------------------------------------------------------|
|                                                              | 27
|                                                              |
|                      Volume identifier                       | 30
|                                                              |
|                                                              | 31
|--------------------------------------------------------------|
```

Type: 00 = no tape mounted
      01 = unlabelled
      10 = ANSI
      11 = IBM
L: 1 if file has lockword.
T: 1 if device is a tape drive.
B: 1 if tape is from Burroughs, which has incorrect block/record size in the HDR2 label. Code can be patched to correct the size.
HP: 1 if tape is Hewlett-Packard ANSI format.

VCB address: Pointer to VCB entry describing volume mounted on tape drive, only if linked. Otherwise, 0.

VCB Entry Format

The VCB format is:

```
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
----------------------------------------------------------------
| A | F | D |    Position    | W | SeqTyp| LblTyp| L | M | R | B |  0
|---------------------------------------------------------------|
|                            LDEV #                             |  1
|---------------------------------------------------------------|
|                             PIN                               |  2
|---------------------------------------------------------------|
|                   File number (AFT index)                     |  3
|---------------------------------------------------------------|
|                    File sequence number                       |  4
|---------------------------------------------------------------|
| S | R | D | C |  Density   | V |          Reel number         |  5
|---------------------------------------------------------------|
|                       Expiration date                         |  6
|---------------------------------------------------------------|
|                                                               |  7
|                                                               |
|                         File name                             | 10
|                                                               |
~                                                               ~
~                                                               ~
|                                                               |
|                                                               | 16
|                              +--------------------------------|
|                              |                                | 17
+---------------------------------------------------------------+
|                                                               | 20
|                                                               |
|                                                               | 21
|                           Lockword                            |
|                                                               | 22
|                                                               |
|                                                               | 23
|---------------------------------------------------------------|
|                                                               | 24
|                                                               |
|                    Volume set identifier                      | 25
|                                                               |
|                                                               | 26
|---------------------------------------------------------------|
|                                                               | 27
|                                                               |
|                        Volume name                            | 30
|                                                               |
|                                                               | 31
|---------------------------------------------------------------|
```

BREAK POINT TABLE

DST = 30(10) = %36

The break point table is divided into 2 sections:

1) PCB BREAKPOINT EXTENSION TABLE (PCB'BKPT'EXT)
   This table contains the heads of the breakpoint chains

2) BREAKPOINT ENTRY TABLE (BKPT'ENTRY'TAB)
   This table contains the actual entries

General Layout

```
                           ----------------------
                           |                    |
                           |   PCB'BKPT'EXT     |
                           |                    |
   PCB(7).(8:8)            |                    |
      ----                 |                    |
      |  | --------------> |--------------------|
      ----              ------                  |
                        |  |--------------------|
                        |  |                    |
                        |  |                    |
                        |  ~                    ~
                        |  ~                    ~
                        |  |                    |
                        |  |-\-\-\-\-\-\-\-\-\-\-
                        |  |                    |
                        |  |   BKPT'ENTRY'TAB   |
                        |  |                    |
                        |  |--------------------|
     SYS GLOBAL         ----->                  |
             14:15      ------                  |
    --------------      |  |--------------------|
%11 |        :L:S|      |  |                    |
    --------------      |  |                    |
                           ~                    ~
    L = Table locked
    S = System break    |  ~                    ~
        points exist    |  |--------------------|
                        ----->                  |
                           |                    |
                           |--------------------|
                           |                    |
                           |                    |
                           |                    |
                           ----------------------
```

PCB BREAKPOINT EXTENSION TABLE

```
-------------------------
|    # ENTRIES          |     ENTRY SIZE =   1
-------------------------
|  HEAD SYSTEM LIST     |     FREE ENTRY =   0
-------------------------
|  # USED USER ENTRIES  |     ACTIVE ENTRY = Index 1st Entry
-------------------------                    in breakpoint
|                       |                    chain
~      USER ENTRIES     ~
~                       ~
|                       |
|                       |
-------------------------
```

BREAKPOINT ENTRY TABLE

```
          ENTRY (0)                      FREE ENTRY
    -------------------------     -------------------------
0   |# WORDS BREAKPOINT TAB |     |1:      SIZE           |
    -------------------------     -------------------------
1   |    HEAD FREE LIST     |     |  FORWARD LINK         |
    -------------------------     -------------------------
2   |                       |     |  BACKWARD LINK        |
    |                       |     -------------------------
3   |      UNUSED           |     |                       |
    |                       |     |                       |
4   |                       |     ~                       ~
    -------------------------     ~                       ~
                                  |                       |
         LAST ENTRY               |                       |
    -------------------------     -------------------------
0   |                      1|
    -------------------------
```

The breakpoint entry table consists of variable length entries
The minimum entry size is 5.

```
                ACTIVE ENTRY

          0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
          ------------------------------
         |0|P:L:V|D:F:T|U:P:C|U: SIZE    |
   0     | | : : | : : | :M: |P:         |
         ---------------------------------
   1     |       BLOCKLABEL              |
         ---------------------------------
   2     |          PLOC                 |
         ---------------------------------
   3     |       INSTRUCTION             |
         ---------------------------------
   4     |          LINK                 |
         ---------------------------------
         |       USERLABEL               |.
         --------------------------------- .
         |                               |  .
         |                               |   .
         ~      CONDITION/COUNT          ~    variable
         ~                               ~     .
         |                               |    .
         ---------------------------------   .
         |   COND DESCRIPTOR             | .
         ---------------------------------
```

BREAKPOINT ENTRY TABLE (CONT.)

ENTRY(0).(0:1) = FR: FREE ENTRY
1 = FREE
0 = USED
ENTRY(0).(1:1) = P: PRIVILEGED MODE BREAKPOINT
1 = PRIV.
0 = NON-PRIV
ENTRY(0).(2:1) = L: PROCESS-LOCAL BREAKPOINT
1 = PROCESS-LOCAL
0 = SYSTEM
ENTRY(0).(3:1) = V: VALIDATION BIT
1 = INSTRUCTION IN ENTRY(3)
0 = INSTRUCTION NOT IN TAB.
ENTRY(0).(4:1) = D: DOUBLE TRAP
1 = BREAKPOINT OSCILLATES BETWEEN P/P+1
0 = NOT DOUBLE TRAP
ENTRY(0).(5:1) = F: FAKE 'DUMMY' TRAP
1 = BREAKPOINT AT P+1
0 = BREAKPOINT AT P (ORIG. LOC)
ENTRY(0).(6:1) = T: TWO WORD INSTRUCTION
1 = TWO WORD INSTRUCTION
0 = NOT TWO WORD INSTRUCTION
ENTRY(0).(7:1) = U: USER LABEL PRESENT
1 = TRAP TO USER SUPPLIED LABEL
0 = TRAP TO DEBUG
ENTRY(0).(8:1) = PM: PERMANENT BREAKPOINT
1 = PERM
0 = TEMPORARY
ENTRY(0).(9:1) = C: CONDITION/COUNT
1 = CONDITION/COUNT SPECIFIED
0 = NO COND/COUNT
ENTRY(0).(10:1) = UP: UPDATING
1 = ENTRY IN PROCESS OF BEING UPDATED/REMOVED
0 = NOT BEING UPDATED/REMOVED
ENTRY(4) = LINK: LINK
0 = END OF CHAIN
>0= INDEX NEXT ENTRY

BREAKPOINT ENTRY TABLE (CONT.)

```
          COUNT                        CONDITION
   --------------------         --------------------
1) |  ORIGINAL CNT.   |      2) |  OPERAND1        |
   --------------------         --------------------
   |  # OF HITS       |         |  OPERAND2        |
   --------------------         --------------------
   |                1 |         |OPT1|OPt2| RELOP  |
   --------------------         --------------------

RELOP -> (8:8) RELOP NUMBER:
               3  = LT     9  = LTE
               4  = GT     10 = GTE
               5  = EQ     11 = NEQ
OPT1  -> (0:2) OPERAND1'S TYPE
OPT2  -> (2:2) OPERAND2'S TYPE

OPERAND TYPES:
   0 ->  CONSTANT (SINGLE WORD)
   1 ->  ADDRESS (DOUBLE WORD)
   3 ->  INDIRECT ADDRESS (TRIPLE WORD)

OPERAND FORMS:
   CONSTANT ->  -------------
                | CONST     |
                -------------

   ADDRESS  ->  -------------
                | REG | BASE|
                -------------
                |  OFFSET   |
                -------------
                |IND. OFFSET|   (TYPE 3 ONLY)
                -------------

   REG      ->  (0:6) CORRESPONDING INDEX INTO 'REGY':
                      3  = A      10 = DL
                      4  = SY     11 = Q
                      7  = DA     12 = S
                      8  = DX     17 = EA
                      9  = DB
   BASE     ->  (6:10) SEG #/BANK #
```

TIMER REQUEST LIST (TRL)

The system clock interrupts every 100 ms, with the CR being automatically cleared. An exception is the Shared Clock Interface measurement service which allows rates as fast as 5 ms. The interrupt handler is the procedure TICK. On entry, DB is pointing to the base of timer request list. Besides timeout requests, the clock also controls time slicing.

```
        -------------------------------|
    / 0|         FREE LIST PTR         |
    |  |-------------------------------|
    | 1|  NR ENTRIES   | ENTRY SIZE(4) |
ENT0|  |-------------------------------|
    | 2|          TRACE WORD           |
    |  |-------------------------------|
    \ 3|  # of days since last start   |  Series 30/33 only
       |-------------------------------|
    / 4|        QUANTUM/100 ms         |  QTIME
    |  |-------------------------------|
    | 5|                               |
    |  |          TIME OF DAY*         |  DTIME*
ENT1| 6|                               |
    |  |-------------------------------|
    \ 7|     YEAR      |  JULIAN DAY   |
       |-------------------------------|
    / 8|  PTR TO MOST ACTIVE REQUEST   |  HEAD
    |  |-------------------------------|
    | 9|               0               |
ENT2|  |-------------------------------|
    |10|               0               |
    |  |                               |  dummy time
    \11|               0               |  ----------
       |-------------------------------|       |
    /12|A|   CODE   |   PTR TO NEXT    |       |
    |  |-------------------------------|       |
    |13|              REQ              |       |
ENT3|  |-------------------------------|  assignable
    |  |     TIME TO SERVICE AFTER     |  entries
    \  | REQUEST IN FRONT (UNIT= 100ms)|       |
       |-------------------------------|       |
       |                               |       |
       |                               |       |
     A:  0 if inactive request            ----------
         1 if active request
```

TRL (CONT.)

CODE & REQ indicate the type of request.

CODE:	REQ:	TYPE:
0	DITP	Hangup
1	DITP	Carrier failure
2	DITP	202 turnaround
3	DITP	Read
4	DITP	Logon
5	PCBB index to process	Delay
6	DITP	LP not ready
7	DITP	2640
%10	Port mask	Msg port timeout
%11	DITP	Block mode read timeout (30 secs)
%12	PCBB index to process	Watchdog timer for process

The list of pending requests is kept ordered by time with later entries at the tail.

%20-%37	DITP	SIO device timeout: DIT8. (code_1 on expiration, cleared on Timereq.
%5/%6	*DTIME	For Series 30/33, DTIME is # of TICS (0.091457 ms) since last midnight.

MPE USER LOGGING enables users and subsystems to log changes to data sets on disc or serial files. This "change" file can later be used to recover data lost due to a system or program failure. The log file can itself be used for auditing purposes.

I. GENERAL DESIGN OVERVIEW

A. Hardware Environment

No special hardware is required to operate the system . However, if logging to a tape file is desired, the hardware configuration must include a tape drive.
If there is no tape drive, then may log to a serial disc class device.

B. Software Environment

MPE USER LOGGING is an integral part of MPE.
No other special software is required.

C. Design Narrative

User Logging enables users and subsystems to journalize additions and modifications to MPE and subsystem files. The journal can reside on either disc or serial logfiles.

User Logging consists of a logging process, a memory buffer, a disc resident logging buffer (for serial logging) and a user defined destination log file on disc or serial media.

The logging process has two functions depending on whether the destination file resides on disc or serial media. If the destination file is serial, the logging process performs all output to the destination file. If the destination file is on disc, the logging process allocates additional space (extents) as it is required by the user.

The logging buffer is divided into communication and buffer areas. The communication area is used to pass information among the users and the logging process. This information includes status of the logging process and logging file, space remaining in the logging file and error information important to users or the logging process. The buffer portion of the logging data segment blocks inputs into the logging file before the data is actually posted. The buffer is flushed any time a user requests to close a log file or when a logging process is terminated. (The buffer is also flushed by the begin/end transaction or buffer flush requests).

D. Error Recovery Description

The error recovery mechanisms provided by User Logging are: power fail recovery and recovery from system failure.

Power failure recovery applies only to tape log files since MPE provides adequate recovery for disc files during power fail. When a power failure is detected, a message will be printed on the console asking the operator to place the tape drive back on-line. (If the operator places the tape on-line before the message valid data may be overwritten). (To reset the tape drive the operator must hit the load button until the tension returns to the drive. Then hit the reset button followed by placing the tape drive back on-line).
At this time the log process will recover the file by rewinding to the load point and then forward spacing to the point where the power fail occured. Writing to the log file will continue at that point.

In the event of a system failure, the warm start load option initiates recovery of User Logging files. In the case of a serial file, the file is read and compared to the disc logging buffer. All records found in the disc buffer that are not on the serial log file are posted and a proper end of file written. If the destination file is a disc file, all records are read and verified and an end of file posted to the file. In order to continue logging to a User Logging file that has been recovered in this manner, the logging process for the file must be restarted using the console command :LOG.

NOTE:

Any records in the buffer area of the logging buffer will be lost.

User logging has been enhanced to work with labeled serial discs. Internally the log process handles serial disc (or cartidge tape) log files the same as for tape files.

II. DESIGN STRUCTURES

A. USER LOGGING TABLE

ENTRY SIZE = #38 words
DST %33

Table containing an entry for each activated user logging process. Each entry is created when the process is started, and deleted when the process terminates. (Via :LOG command). The information is extracted from the Logging Identifier Table (LIDTAB).

ENTRY 0

#		%
0	NUMBER OF ENTRIES	0
1	FREE ENTRY HEAD PT.	1
2	INUSE ENTRY HEAD PT.	2
3	NEXT BUFFER NUMBER	3
4	MAX # PROCESSES	4
5	MAX # USERS/PROCESS	5
6		6
7	ENTRY SIZE	7
	.	
	.	
37	.	45

WORD ENTRIES

NUMENTRIES = LOGTAB
FREE = LOGTAB(1)
INUSE = LOGTAB(2)
BUFNUM = LOGTAB(3)
MAXLOGPROC = LOGTAB(4)
MAX'USR'PROC = LOGTAB(5)
LOGTAB'ESIZE = LOGTAB(7)

NUMENTRIES
The number of entries in the logging table.

FREE
A table relative pointer to the first free entry in the logging table. (-1 = table full).

INUSE
A table relative pointer to the first entry in the logging table that is being used (-1 = no entries in use).

BUFNUM
The number of the buffer associated with this logging process. Used to create the name of buffer file if serial logfile. (i.e. ULOGxxxx.PUB.SYS).

MAXLOGPROC
The maximum number of user logging processes allowed.

MAX'USR'PROC
The maximum number of users per logging process.

LOGTAB'ESIZE
The size (in words) of each entry in the table.

#	TYPICAL ENTRY	%
0	LOGGING IDENTIFIER	0
4	BUFFER NAME	4
8	FILE NAME	10
12	LOCK WORD	14
16	GROUP	20
20	ACCT	24
24	NUMBER OF USERS	30
25	BUFFER DST NO	31
26	LOG STATUS	32

```
27    |CURR AUTO | CURR TYPE|   33
      |______________________|
28    |      LOG  DEV        |   34
      |______________________|
29    |      LOG  PCB #      |   35
      |______________________|
30    |     SWITCH FLAG      |   36
      |______________________|
31    |NEW AUTO   | NEW TYPE |   37
      |______________________|
32    |     ADDRESS OF       |   40
      |_                    _|
      | LOGGING BUFFER       |
      |______________________|
34    |      SIZE OF         |   42
      |_                    _|
      | LOGGING BUFFER       |
      |______________________|
36    |   FWRD ENTRY PT      |   44
      |______________________|
37    |   BWRD ENTRY PT      |   45
      |______________________|
```

TABINDEX = WORD INDEX TO CURRENT ENTRY
BTABINDEX = BYTE INDEX TO CURRENT ENTRY
DTABINDEX = DOUBLE INDEX TO CURRENT ENTRY

LGNAME = BTABINDEX
BNAME = BTABINDEX+8
LFNAME = BTABINDEX+16
LFLOCKW = BTABINDEX+24
LFGROUP = BTABINDEX+32
LFACCT = BTABINDEX+40

NUMUSERS = TABINDEX+24
DST = TABINDEX+25
STATUS = TABINDEX+26
LGAUTO = TABINDEX+27.(0:8)
LGTYPE = TABINDEX+27.(8:8)
LGDEV = TABINDEX+28
PIN = TABINDEX+29
LGSWITCH = TABINDEX+30
LGNEWAUTO = TABINDEX+31.(0:8)
LGNEWTYPE = TABINDEX+31.(8:8)

LGADDR = DTABINDEX+16
BSIZE = DTABINDEX+17

NEXT = TABINDEX+36
PREV = TABINDEX+37

LGNAME
The name of the logging process (logging identifier).

BNAME
The name of the disc buffer used if the logging process destination file is a serial file. This is a file that resides in PUB.SYS. The format of the name is ULOGxxxx where xxxx is the buffer number padded on the left with zeroes.

If the switch flag is true, the following will be the fully qualified file name of the new log file.

LFNAME
The name of the logging file.

LFLOCKW
The lockword of the disc logging file.

LFGROUP
The group that the destination logging file resides in if the file is a disc file.

LFACCT
The account that the destination logging file resides in if the file is a disc file.

NUMUSERS
The number of users currently accessing the logging file.

DST
The dst number of the logging data segment (LOGBUFF).
(-1 = LOGBUFF not created yet)

STATUS
The status of the logging process.
ACT = 1, INACT = 0, RECOVERING = 2, INITIALIZING = -1.

LGAUTO
True if the automatic changelog facility was enabled.

LGTYPE
The type of destination file of the logging process.
DISC = 0, TAPE = 1, SDISC = 2, CTAPE = 3

LGDEV
The logical device number of the disc logging file or the disc logging buffer.

PIN
The PCB number for the logging process.

LGSWITCH
Flag indicating a CHANGELOG is pending (if true).

LGNEWAUTO

True if the automatic changelog facility was requested for the new log file.

LGNEWTYPE
If a switch is pending, this will be the type of the new log process. (-1 = no switch pending)

LGADDR
Sector number of the current extent in the disc logging file or the disc buffer file. (Disc buffer file has only 1 extent)

BSIZE
The number of records in the current extent (for disc logging) or the number available in the disc logging buffer.

NEXT
A table relative pointer to the next entry in the logging table. (-1 = this is last entry)

PREV
A table relative pointer to the previous entry in the logging table. (-1 = this is first entry)

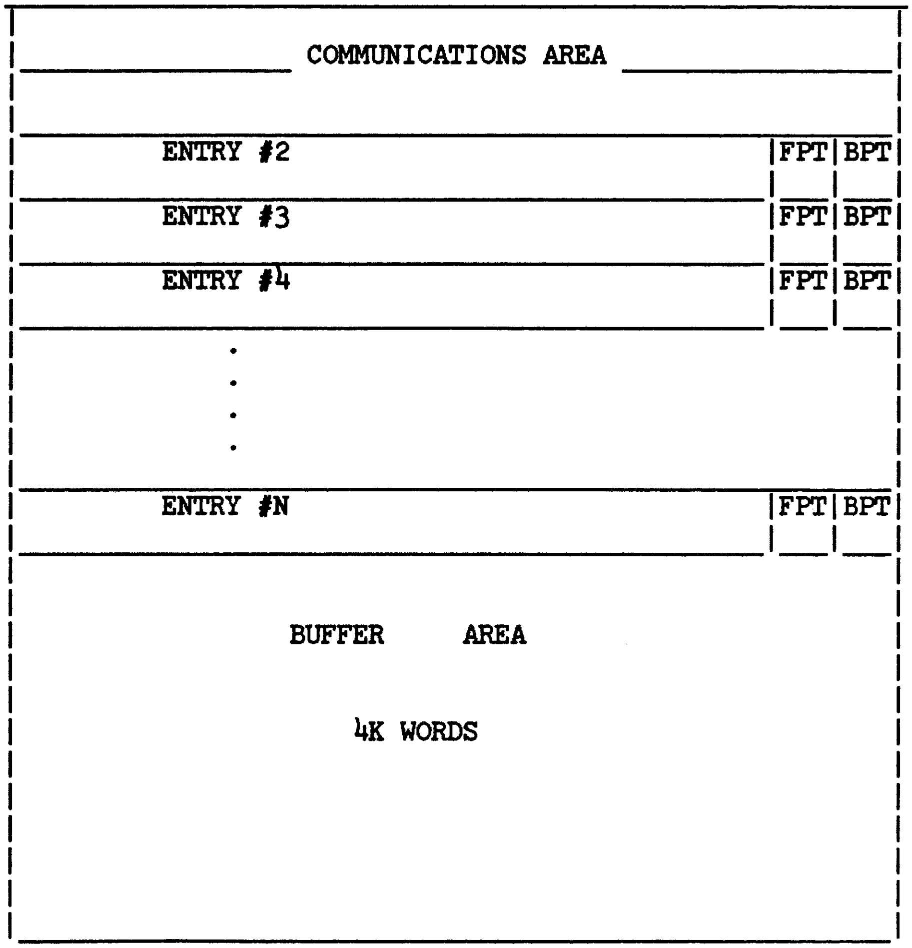
B. USER LOGGING BUFFER

There will be one of these tables around for the life of any active user loggging process. The table consists of three parts:

COMMUNICATIONS AREA - Info about status of the process, etc. that is common to all users of the process. Also the cells for messages to/from the process.

USER ENTRIES - Info for a specific user of the process. One of these for every user of a process (Setup by OPENLOG, released by CLOSELOG).

BUFFER AREA - Buffer used to hold logging records from all users before writing to the log file.



COMMUNICATIONS AREA

#		%
0	LOGGING IDENTIFIER	0
4	SWITCH FLAG	4
5	NEW AUTO | NEW TYPE	5
6	AUTO | TYPE	6
7	BUFFER DST	7
8	LOG PIN	10
9	NUMBER OF USERS	11
10	MAX NUMBER OF USERS	12
11	NEXT USER NUMBER	13
12	SLEEP COUNT	14
13	STATE	15
14	MSG	16
15	LOG MSG	17
16	USER MSG	20
17	LOG ERROR	21
18	LOG DEVICE	22
19	BUFFER SPACE	23
20	USED SPACE IN BUFFER	24
21	FILE SET NUMBER	25
22	LOG ADDRESS	26
24	INPUT RECORD	30
26	FILE	32

USER LOGGING BUFFER (CONTINUED)

	SIZE	
28	FILE SPACE	34
30	TOTAL RECORDS	36
32	MAX SIZE	40
34	LAST EXTENT	42
35	EXTENT	43
36	RESOURCE	44
38		46
48	IN USE HEAD PTR	60
49	FREE HEAD PTR	61

```
LOGID             =     BLOGBUFF(0)

SWITCH'           =     LOGBUFF(4)
NEWAUTO           =     LOGBUFF(5).(0:8)
NEWTYPE           =     LOGBUFF(5).(8:8)
AUTO              =     LOGBUFF(6).(0:8)
LOGTYPE           =     LOGBUFF(6).(8:8)
BDST              =     LOGBUFF(7)
LOGPIN            =     LOGBUFF(8)
NUMUSER           =     LOGBUFF(9)
MAXUSER'          =     LOGBUFF(10)
USERNO            =     LOGBUFF(11)
SLPCT             =     LOGBUFF(12)
STATE             =     LOGBUFF(13)
MSG               =     LOGBUFF(14)
LOGMSG            =     LOGBUFF(15)
USERMSG           =     LOGBUFF(16)
LOGERR            =     LOGBUFF(17)
LOGDEV            =     LOGBUFF(18)
BSPACE            =     LOGBUFF(19)
BUFUSED           =     LOGBUFF(20)
VSETNO            =     LOGBUFF(21)

LOGADDR           =     DLOGBUFF(11)
INBUFREC          =     DLOGBUFF(12)
FSIZE             =     DLOGBUFF(13)
FSPACE'           =     DLOGBUFF(14)
TRECS             =     DLOGBUFF(15)
MAXFSPACE         =     DLOGBUFF(16)

LASTEXT'          =     LOGBUFF(34)
EXTENT            =     LOGBUFF(35)

RESOURCE          =     DLOGBUFF(18)

UHEAD             =     LOGBUFF(48)
FHEAD             =     LOGBUFF(49)
```

LOGID
The name of the logging process.

SWITCH'
True if log file switch is pending.

NEWAUTO
True if the automatic changelog option has been specified for the new log file.

NEWTYPE
If a switch was requested, this will be the type of the new logging file. (-1 = no switch pending)

AUTO
True if the automatic changelog option was specified for the current log file.

LOGTYPE
The type of destination file for the logging process.
DISC = 0, TAPE = 1, SDISC = 2, CTAPE = 3

BDST
The data segment number of this table.

LOGPIN
This is the PCB number for the logging process (PIN*16).

NUMUSER
The number of users currently accessing the logging file.

MAXUSER'
The maximum number of users allowed to access the logging file.

USERNO
The next sequential number to be assigned users accessing the system. It will get incremented for every unique OPENLOG - used as the log # in the logging record format.

SLPCT
The number of users currently waiting for activization by the logging process.

STATE
The state of the user logging process.
ACTIVE = 1, INACTIVE = 0.

MSG
An internal messge word used to indicate an error or operator request.

```
    6 - Continue processing, all o.k.
    2 - Suspend - error reading buffer file or writing to serial file
    3 - Stop - set when issue :LOG logid,STOP  or when an EOF
               condition is found on the disc log file.
```

LOGMSG

A messages from the logging process.

6 - Continue processing, all O.K.
15 - EOF - if there are no more extents available to be allocated.
12 - Disc space - could not allocate the new extent because no space left in the group.
9 - Write error - error occurred while writing to log file

USERMSG

A messages from the user process.

6 - Continue processing, all O.K.
12 - Disc space - user process needs another extent allocated for disc logging.

LOGERR

Last error found.
After changelog:

+N - File System error number encounterd
0 - No error
-1 - New disc log file was not empty
-2 - New disc log file did not have file code LOG
-3 - New disc file is too small

LOGDEV

The logical device number of the current extent of the disc log file or the disc buffer file (buffer file has only 1 extent).

BSPACE

The amount of space, in records, that are currently available to the users. On the last block of the last extent, one record will be saved by the logging process so that the proper close information can be posted to the file - either the trailer record (if the log logging process is stoppped) or the change'to'new record because of an EOF condition (and the AUTO option had been specified).

BUFUSED

The number of records currently in the buffer. On all extents, except the last extent BUFSPACE+BUFUSED = 32 (number of records in a complete block). However, on the last block of the last extent this will NOT be true since one record is always held in reserve by the logging process.

VSETNO

This shows the order in the log file "set" of the currently opened log file.

LOGADDR

The disc address of the current extent of the disc log file. If it's a serial file, this is the disc address of the disc buffer for the file.

INBUFREC

The record number of the next block to be written to the logging destination file or the disc logging buffer for serial files. (Used as an offset into the current extent for the writes - since each record is one sector in length).

FSIZE
The current extent size of the logging destination file or disc logging buffer file for serial destination files.
(on the last extent this will be the last extent size minus 1).

FSPACE'
The space in records that remains in the current extent of the disc logging destination file or disc buffer for tape destination files. (On the last extent of the disc log file, this is the amount of space minus 1).

TRECS
The total number of records written to the logging destination file (including those records currently in the buffer).

MAXFSPACE
The total file size, in records, minus 1. (Need that last record to post close information).

LASTEXT'
The extent number of the final extent in the disc logging file or disc buffer file.

EXTENT
The current extent number of the disc logging file or disc logging buffer.

RESOURCE
Used for resource management (i.e. locking the LOGBUFF).
Format is:

```
    RESOURCE.(0:8) = Owner's pin,  RESOURCE.(8:8) = Queue length,
    RESOURCE1.(0:8)= Q tail pin,   RESOURCE1.(8:8)= Q head pin.
```

UHEAD
A table relative pointer to the first entry into the logging data segment. (-1 = no entries currently in use)

FHEAD
A table relative pointer to the first free entry in the logging data segment. (-1 = no free entries)

TYPICAL LOGBUFF ENTRY

#		%
0	USER NAME	0
4	GROUP NAME	4
8	ACCOUNT NAME	10
12	USER PCB #	14
13	OPENLOG COUNT	15
14	WAIT STATE	16
15	ERROR CODE	17
16	LOG NUMBER	20
17	SUBSYSTEM CODE	21
18	TOTAL RECORDS	22
23	FRWD ENTRY PTR	27
24	BKWRD ENTRY PTR	30

```
BINDEX            =     BYTE INDEX TO CURRENT ENTRY
INDEX             =     WORD INDEX TO CURRENT ENTRY
DINDEX            =     DOUBLE INDEX TO CURRENT ENTRY

USER              =     BINDEX
GROUP             =     BINDEX+8
ACCT              =     BINDEX+16

UPIN              =     INDEX+12
OPENCNT           =     INDEX+13
WSTATE            =     INDEX+14
ERROR             =     INDEX+15
LGNUM             =     INDEX+16
SCODE             =     INDEX+17

RECS              =     DINDEX+9

NENTRY            =     INDEX+23
PENTRY            =     INDEX+24
```

USER
The name of the user who opened the logging file through this entry.

GROUP
The group of the user who opened the logging file.

ACCT
The account of the user who opened the logging file.

UPIN
The process identification number for the user's process.

OPENCNT
Counter of how many times this user called OPENLOG. (Incremented for every OPENLOG, decremented for every CLOSELOG).

WSTATE
The wait status of the users process.
ACTIVE = 1, INACTIVE = 0.

ERROR
Used to hold error information for this user.
0 = O.K.   -1 = no room in disc (or disc buffer) and NOWAIT.

LGNUM
The logging number assigned to the user.
(From USERNO in global area to be used as log # in the log record).

SCODE
The subsystem code for the caller. This applies only to privleged callers.

RECS
The number of records written by this user.

NENTRY
A table relative pointer to the next entry in the logging data segment. (-1 = this is the last entry)

PENTRY
A table relative pointer to the previous entry in the logging data segment. (-1 = this is the first entry)

C. LOGGING IDENTIFIER TABLE

ENTRY SIZE = #33 words
DST %41

Table containing an entry for each potential logging process. Entries are added via :GETLOG and released via :RELLOG.

```
                  ENTRY #0
  #                                        %
          _______________________
  0      |                       |         0
         |_______________________|
  1      |MAX NUMBER OF ENTRIES  |         1
         |_______________________|
  2      |                       |         2
         |_______________________|
  3      |                       |         3
         |_______________________|
  4      |      ENTRY SIZE       |         4
         |_______________________|
         |           .           |
         |           .           |
 32      |           .           |        40
         |_______________________|
```

ENTRIES

MENTRIES = LIDTAB(1)
ENTRYSIZE = LIDTAB(4)

MENTRIES
The maximum number of entries in the table. (i.e. maximum number of user logging processes. 1 entry for every process - activated or not).

ENTRYSIZE
The size of each entry in the table.

TYPICAL ENTRY

#		%
0	LOGGING IDENTIFIER	0
4	PASSWORD	4
8	FILE NAME	10
12	FILE LOCK WORD	14
16	FILE GROUP	20
20	FILE ACCOUNT	24
24	USER'S	30

```
             ~                     ~
             |        NAME         |
             |_                   _|
             |                     |
             |_____________________|
     28      |                     |     34
             |_                   _|
             |       USER'S        |
             |_                   _|
             |       ACCOUNT       |
             |_                   _|
             |                     |
             |_____________________|
     32      |      LOG TYPE       |     40
             |_____________________|
```

BYTE ENTRIES

LID = BLIDTAB
PW = BLIDTAB(8)
FNAME' = BLIDTAB(16)
LW = BLIDTAB(24)
FGROUP = BLIDTAB(32)
FACCT = BLIDTAB(40)
UNAME = BLIDTAB(48)
UACCT = BLIDTAB(56)

WORD ENTRIES

TYP = LIDTAB(32)

LID
The logging identifier name. This is a maximum of eight characters long.

PW
The pass word for the logging identifier. This is a maximun of eight characters long.

The following is the fully qualified file name of the current log file.

FNAME'
The name of the destination file.

LW
The lock word on the destination file if the file is on disc.

FGROUP
The group that the file resides in.

FACCT
The account that the destination file resides in.

UNAME
The name of the user who created the logging identifier.

UACCT
The account of the user who created the logging identifier.

TYP
The status of the entry.

-1 = null entry
0 = disc logging file
1 = tape logging file
2 = serial disc logging file
3 = cartridge tape logging file

D. LOGGING RECORD FORMAT

RECORD SIZE = 128 words
USER AREA = 119 words

LOG RECORD AT OPENLOG

```
0     2    3     4    6     7     11   12        24   25           127
 ___________________________________________________________________
|     |     |     |    |    |      |    |         |    |            |
| rec#|cksum|code |time|date| logid|log#|  creator|pcb |            |
|_____|_____|_____|____|____|______|____|_________|____|____________|
```

USER OR SUBSYSTEM/CONTINUATION LOG RECORD (from WRITELOG)

```
0     2     3     4    6    7    8   9                              127
 ___________________________________________________________________
|     |     |     |    |    |    |   |                              |
| rec#|cksum|code |time|date|log#|len|        user area             |
|_____|_____|_____|____|____|____|___|______________________________|
```

LOG RECORD AT CLOSELOG

```
0     2    3     4    6     7     11   12        24   25           127   128
 ___________________________________________________________________   __
|     |     |     |    |    |      |    |         |    |            |
| rec#|cksum|code |time|date| logid|log#|  creator|pcb |            |
|_____|_____|_____|____|____|______|____|_________|____|____________|
```

CRASH MARKER

```
0     2     3     4    6    7                                       127  128
____________________________________________________________________
|     |     |     |    |    |                                       |
| rec#|cksum|code |time|date|                                       |
|_____|_____|_____|____|____|_______________________________________|  _|
```

HEADER RECORD (START/RESTART)

```
0     2    3     4    6     7       11                              127
____________________________________________________________________
|     |     |     |    |    |       |                               |
| rec#|cksum|code |time|date| logid |                               |
|_____|_____|_____|____|____|_______|_______________________________|
```

TRAILER RECORD (STOP)

```
0     2     3     4    6     7      11                                127
 ______________________________________________________________________
|     |     |     |    |    |       |                                 |
| rec#|cksum|code |time|date| logid |                                 |
|_____|_____|_____|____|____|_______|_________________________________|
```

NULL RECORD

```
0     2     3     4    6    7                                         127  128
 ______________________________________________________________________   _
|     |     |     |    |    |                                         |  |
| rec#|cksum|code |time|date|                                         |  |
|_____|_____|_____|____|____|_________________________________________|  _|
```

BEGIN TRANSACTION MARKER

```
0     2     3     4    6    7    8   9                                127
 ______________________________________________________________________
|     |     |     |    |    |    |   |                                |
| rec#|cksum|code |time|date|log#|len|        user area               |
|_____|_____|_____|____|____|____|___|________________________________|
```

END TRANSACTION MARKER

```
0     2     3     4    6    7    8   9                                127
 ______________________________________________________________________
|     |     |     |    |    |    |   |                                |
| rec#|cksum|code |time|date|log#|len|        user area               |
|_____|_____|_____|____|____|____|___|________________________________|
```

CODE DEFINITION

```
  CODE.(8:8) =
        1   Open log record
        2   User/subsystem record (writelog)
        3   Close log record
        4   Header record
        5   Trailer record
        6   Restart record
        7   Continuation of a user or subsystem record
        9   Crash marker
       10   End transaction record
       11   Begin transaction record
    SPACE   NULL record
```

DATA FIELDS OF LOG RECORDS

REC# = DOUBLE INTEGER
CKSUM = INTEGER
CODE = INTEGER
TIME = DOUBLE (from intrinsic CLOCK)
DATE = INTEGER (from intrinsic CALENDAR)
LOGID = ASCII
LOG# = INTEGER
LEN = INTEGER
USERAREA = ASCII
CREATOR = ASCII
PCB = INTEGER

NOTE:

1. The checksum algorithm uses the exclusive or (XOR) function against a base of negative one.

2. Null record is used for filler.

3. The code word of the logging record can contain a subsystem code defined by the user in the first half of the word (0:8). User logging allows privileged users to pass this code in the index parameter of the Openlog intrinsic.

4. The "len" field will contain the entire length of the data in the transaction (i.e. the length passed to WRITELOG, BEGINLOG, ENDLOG). If a continuation record is part of the transaction, it will also contain the entire length of the data. For example, a length of 140 was passed to the intrinsic. The "len" field of the first record will be 140, the "len" field of its continuation record will also be 140 - even though the actual amount of data found in the first record will be 119 and the data found in the continuation record will be 21.
   (Positive length = # words, negative length = # bytes)

MEASUREMENT INFORMATION TABLE (MEASINFOTAB) DST=59 (%73)

```
        ------  --------------------------------------
           |    0 | LDEV # OF MEASIO                   | MEASLDEV
           |      --------------------------------------
           |    1 | MEASIO PLABEL                      | MEASPLAB
           |      --------------------------------------
           |    2 | MEASIO DST #                       | MEASDSTN
           |      --------------------------------------
Reserved        3 |                                    |
for MEASIO        --------------------------------------
control         4 |                                    |
           |      --------------------------------------
           |    5 |                                    |
           |      --------------------------------------
           |    6 |                                    |
           |      --------------------------------------
           |    7 |                                    |
        -----     --------------------------------------
           |   10 |                                    |
           |      --------------------------------------
           |   11 |                                    |
           |      --------------------------------------
           |   12 |                                    |
Reserved          --------------------------------------
for            13 |                                    |
performance       --------------------------------------
tunning        14 |                                    |
parameters        --------------------------------------
           |   15 |                                    |
           |      --------------------------------------
           |   16 |                                    |
           |      --------------------------------------
           |   17 |                                    |
        -----     --------------------------------------
           |   20 | GLOBAL STATISTICS XDS NUMBER       | MEASSTATX-
           |      |                                    | DSNUM
           |      --------------------------------------
           |   21 | PROCESS STATISTICS XDS BANK        | MEASPROC-
           |      |                                    | XDSBANK
           |      --------------------------------------
           |   22 | PROCESS STATISTICS XDS BASE        | MEASPROC-
           |      |                                    | XDSBASE
           |      --------------------------------------
           |   23 | PROCESS STATISTICS XDS NUMBER      | MEASPROC-
           |      |                                    | XDSNUM
           |      --------------------------------------
           |   24 |    CLASS 14 STATISTICS XDS BANK    |
           |      --------------------------------------
           |   25 |    CLASS 14 STATISTICS XDS BASE    |
           |      --------------------------------------
```

MEASINFOTAB (CONTINUED)

```
             |     -------------------------------------
             | 26 |    CLASS 14 STATISTICS XDS NUM.     |
             |     -------------------------------------
             | 27 |    CLASS 13 STATISTICS XDS BANK     |
             |     -------------------------------------
             | 30 |    CLASS 13 STATISTICS XDS BASE     |
             |     -------------------------------------
             | 31 |    CLASS 13 STATISTICS XDS NUM.     |
             |     -------------------------------------
             | 32 |    CLASS 12 STATISTICS XDS BANK     |
             |     -------------------------------------
             | 33 |    CLASS 12 STATISTICS XDS BASE     |
             |     -------------------------------------
             | 34 |    CLASS 12 STATISTICS XDS NUM.     |
             |     -------------------------------------
             | 35 |    CLASS 11 STATISTICS XDS BANK     |
             |     -------------------------------------
             | 36 |    CLASS 11 STATISTICS XDS BASE     |
             |     -------------------------------------
             | 37 |    CLASS 11 STATISTICS XDS NUM.     |
             |     -------------------------------------
             | 40 |    CLASS 10 STATISTICS XDS BANK     |
             |     -------------------------------------
             | 41 |    CLASS 10 STATISTICS XDS BASE     |
             |     -------------------------------------
             | 42 |    CLASS 10 STATISTICS XDS NUM.     |
             |     -------------------------------------
             | 43 |    CLASS 09 STATISTICS XDS BANK     |
             |     -------------------------------------
             | 44 |    CLASS 09 STATISTICS XDS BASE     |
             |     -------------------------------------
             | 45 |    CLASS 09 STATISTICS XDS NUM.     |
             |     -------------------------------------
  -----------

             |     -------------------------------------
  reserved   . |                                       |
       for     -------------------------------------
measurement  . |                                       |
  interface    -------------------------------------
             |  . |                                     |
             |     -------------------------------------
             | 50 |CLASS 0 ENABLED   |CLASS 1 ENABLED   |
             |    |COUNT             |COUNT             |
             |     -------------------------------------
             | 51 | CLASS 2 EN.CNT. | CLASS 3 EN.CNT.   |
             |     -------------------------------------
             | 52 | CLASS 4 EN.CNT. | CLASS 5 EN.CNT.   |
             |     -------------------------------------
             | 53 | CLASS 6 EN.CNT. | CLASS 7 EN.CNT.   |
             |     -------------------------------------
             | 54 | CLASS 8 EN.CNT. | CLASS 9 EN.CNT.   |
             |     -------------------------------------
```

MEASINFOTAB (CONTINUED)

```
        | 55 | CLASS 10 EN.CNT.| CLASS 11 EN.CNT.  |
        |    ---------------------------------------
        | 56 | CLASS 12 EN.CNT.| CLASS 13 EN.CNT.  |
             ---------------------------------------
        | 57 | CLASS 14 EN.CNT.| CLASS 15 EN.CNT.  |
      ----   ---------------------------------------
        | 60 |                                     |
        |    ---------------------------------------
        | 61 |                                     |
  reserved   ---------------------------------------
    for   62 |                                     |
  shared     ---------------------------------------
  clock   63 |                                     |
  interface  ---------------------------------------
  user    64 |                                     |
        |    ---------------------------------------
        | 65 |                                     |
        |    ---------------------------------------
        | 66 |                                     |
        |    ---------------------------------------
        | 67 |                                     |
      -----  ---------------------------------------
```

CHAPTER 18 MESSAGE FILES

This chapter contains the data structures necessary to support message files. files. The first section details the message file's version of the familiar file system data structure; ie, the file label, file control block, access control block, etc..

The second section show the tables used by the basic ipc mechanism which is a set of internal, MPE procedures designed to support the "boundary conditions" of ipc files. For example, signalling a no wait reader that its record has arrived. See the section's introduction for a detailed description.

{File Structure}

{File label/FCB extent map}

```
............................ End of file block    Start of file block
: Disc addr of extent 0     :   .                    .
:...........................:   .                    .
: Disc addr of extent 1     :   v                    .
............................:   -                    .
: Disc addr of extent 2     :                        .
:...........................:                        .
: Disc addr of extent 3     :                        .
:...........................:                        .
z                           z                        .
:...........................:                        .
: Disc addr of extent n-1   :                        v
:...........................:                        -
: Disc addr of extent n     :
:...........................:
```

The EOF and SOF are examples only, meant to show that 1) the start of file moves into the extent map as records are read and 2) that the file can wrap around and, hence, cause the SOF to be greater than the EOF.

When a file becomes empty the SOF and EOF are reset to the first block of extent zero.

Each extent is composed of a number of blocks. Extents all have the same number of blocks. Extent zero also contains space for the file label and user labels in the exact same format as standard files. Starting with block zero, sufficient blocks are allocated to the file label/user labels to satisfy their space requirements.

Extents outside of the SOF/EOF range may not exist. They are deleted at close time when there are no more writers accessing the file.

```
     {Block Structure}

.............................   **********************************
: First data record         :
:...........................:   Exact same format as standard
: Second data record        :   variable length blocks.
:...........................:
z                           z
:...........................:
: Last data record          :
:...........................:
: Record delimeter (-1)     :
:...........................:   **********************************
:                           :
: Empty space (next record  :
: would not fit)            :
:                           :
:...........................:
: Header delimiter (%77)    :
:...........................:
: Last header record        :
:...........................:
z                           z
:...........................:
: Second header record      :
:...........................:
: First header record       :
:...........................:
```

Separating the data portion of the records from their header enables the standard file system access procedures to read the records with no knowledge that they are msg file records.

{Record Format}

```
.............................
: Number of bytes in record :
:...........................:
: First data word of record :
:...........................:
z                           z
:...........................:
: Last data word of record  :
:...........................:
```

Length word's value does not include itself.

{Header Format}

```
.............................
: C:LC:       :  Header Type: 0
:...........................:
: Writer's ID               : -1
:...........................:
```

C (0:1) - Set on if this was the last record written before the system crashed. This bit is set on by the first open on the file after the crash.

LC (1:1)- Valid only for close headers. Set to one if this is the last writer to close the file.

Type(8:8)- 0 data
1 open
2 close

{Message Access Control Block}

Notes:

1. Words/fields that do not pertain to message files are left blank.

2. This diagram shows the "combined" ACB as it appears to the message access procedures (the procedures in IPC). Thus it is a combination of the LACB and the PACB.

```
    ...................................................
 0 :   : Size of the ACB including buffers (words)    : 0
    ...................................................
 1 :                        : File number             : 1  *
    ...................................................
 2 : File name                                        : 2  *
    .....................            ...................
    z                                                 z    *
    ...................................................
 6 : Foptions                                         : 6  *
    ...................................................
 7 : Aoptions                                         : 7  *
    ...................................................
 8 : Record size (bytes)                              : 10 *
    ...................................................
 9 : Block size (words)                               : 11 *
    ...................................................
    z                                                 z    *
    ...................................................
11 : Carriage control code (writers)                  : 13 *
    ...................................................
    z                                                 z    *
    ...................................................
14 : Error code                                       : 16 *
    ...................................................
15 : Transmission log (units same as last read/write) : 17 *
    ...................................................
16 : Total number of unread records (includes opens   : 20
    ...............                    ................
17 : and closes)                                      : 21
    ...................................................
18 : Block number of the file's tail (relative to the : 22
    ..................               ..................
19 : start of file block)                             : 23
    ...................................................
20 : Logical record transfer count                    : 24
    ..................               ..................
21 :                                                  : 25
    ...................................................
```

```
    .....................................................
22 : Physical block transfer count                     : 26
    ..................                ...................
23 :                                                   : 27
    .....................................................
24 : Address of the head record's header               : 30
    .....................................................
25 : Address of the next write header                  : 31
    .....................................................
26 : FCB control block vector                          : 32
    .....................................................
   z                                                   z
    .....................................................
28 : Number readers         : Number readers & writers : 34
    .....................................................
29 z                                                   z
    .....................................................
30 :                        : Records per block        : 36
    .....................................................
31           :Wrt buf indx:                : # buf - 1 : 37
    .....................................................
32 : Address of the head record's data                 : 40
    .....................................................
33 : Size of the buffer (words)                        : 41
    .....................................................
   z                                                   z
    .....................................................
38 :                        : Logical device number    : 46
    .....................................................
39 :0:# rd buf  : # wt buf   :er :qw :m :c :d  :s :f   : 47
    .....................................................
40 : Number of max sized free records                  : 50
    .....................................................
41 :                                                   : 51
    .....................................................
42 : Number of free words in the current free record   : 52
    .....................................................
43 : Address of the next write record                  : 53
    .....................................................
44 : Number of nondata records in the file             : 54
    .....................................................
45 :                                                   : 55
    .....................................................
46 : # of read requests that have a claim on file      : 56
    .....................................................
47 : Last read error     : Last write error            : 57
    .....................................................
48 : DST number of the physical ACB                    : 60
    .....................................................
49 : Address of the physical ACB                       : 61
    .....................................................
```

```
   ..................................................
50 : DST number of the logical ACB                   : 62
   ..................................................
51 : Address of the logical ACB                      : 63
   ..................................................
52 : DST rel address of the stack access control blk : 64
   ..................................................
53 : DST rel address of the DB area                  : 65
   ..................................................
54 : PACB vector table entry address                 : 66
   ..................................................
55 : PACB control block vector table address         : 67
   ..................................................
56 : Target area's DST number                        : 70
   ..................................................
57 : Reserved for calling parameters                 : 71
   ...............                         ..........
58 :                                                 : 72
   ...............                         ..........
59 :                                                 : 73
   ..................................................
60 : Reserved for the stack marker from file system  : 74
   ...............                         ..........
61 : intrinsics                                      : 75
   ..................................................
   z                                                 z
   ..................................................
64 : User's soft interrupt plabel                    : 100*
   ..................................................
65 : Number of seconds to wait on boundary condition : 101*
   ..................................................
66 : O:Ex:Nd:Vr:Bt:Cls  :C : Carriage control        : 102*
   :.................................................:
67 : Reply Port (basic IPC port)                     : 103*
   ..................................................
68 : Writer ID                                       : 104*
   ..................................................
69 : Control block index for nowait writer record buf : 105*
   ..................................................
70 : DST relative addr of nowait writer record buffer : 106*
   ..................................................
71 : No wait I/O resultant error code                : 107*
   ..................................................
72 : No wait I/O resultant transmission log          : 110*
   ..................................................
73 : Write wait queue (basic IPC port)               : 111
   ..................................................
74 : Read wait queue (basic IPC port)                : 112
   ..................................................
75 : Head record's length (bytes)                    : 113
   ..................................................
76 : Head record's record type (same values as header): 114
   ..................................................
```

```
     .....................................................
 77 : Head record's writer ID                            : 115
     .....................................................
 78 : Head record's header word value                    : 116
     .....................................................
 79 : Max size record plus its overhead (words)          : 117
     .....................................................
 80 : ACB wait queue message - contains same info  as   : 120
     ...................               ...................
 81 : the wait queue message in the Message Queue        : 121
     ...................               ...................
 82 : Entry                                              : 122
     ...................               ...................
 83 :                                                    : 123
     .....................................................
 84 :                                                    : 124
 85 : Waiter's reply port, 0 if using ACB compltn area : 125
     .....................................................
 86 : ACB completion message area - see Message Queue   : 126
     ..................                ...................
 87 : Entry for completion message format               : 127
     .....................................................
 88 : Waiting process's pin                              : 130
    :.....................................................:
 89 : Waiting process's file number                      : 131
    :.....................................................:
 90 : Waiting process's soft interrupt plabel            : 132
     .....................................................
 91 : DST rel address of buffer one                      : 133
     .....................................................
 92 : DST rel address of buffer two                      : 134
     .....................................................
 93 : Etc.                                               : 135
     .....................................................
```

* Value is private to a particular accessor.

Word	Field	Description
66		Accessor's local flags.
	(0:1)	O 1 - have not yet issued an FREAD/FWRITE against the file.
	(1:1)	ex 1 - extended wait mode.
	(2:1)	nd 1 - do not destroy the next record read.
	(3:1)	vr 1 - writer has not yet written his first record (ie., he is a virgin).
	(4:1)	bt 0 - transmission log should be expressed in words. 1 - " " " " " " bytes.
	(5:1)	cls Not currently used (reserved for group IPC standard).
	(6:1)	C No wait completion message is in LACB area.
	(8:8)	car ctl carriage control character to be used for the writer's record (a value of one indicates no carriage control character).

Word	Field	Description
39		File's global flags.
	(9:1)	er 1 - extended read
	(10:1)	qw 1 - one or more writers has been queued on the wait queue.
	(11:1)	m 1 - wait msg is located in the ACB
	(12:1)	c 1 - completion msg is located in the ACB
	(13:1)	d 1 - the current write buffer has dirty bit set
	(14:1)	s 1 - the start of file is block zero
	(15:1)	f 0 - the ACB buffers have not been filled

{MMSTAT Definitions}

Octal Value	Event Type	Parameter 1	Parameter 2
72/0	Read init	# free rec	
72/1	Read compl	(0:8) error, (8:8) ID	Number of records
72/2	Write init	(0:8) # rec, (8:8) ID	Number of free records
72/3	Write compl	(0:8) error, (8:8) ID	Number of free records
72/4	Control	(0:8) error, (8:8) ID	(0:4) func, (4:12) parm
72/5	EOF	(0:8) error, (8:8) ID	Number of records
72/6	Open	(0:8) error, (8:8) ID	Number of records
72/7	Close	(8:8) #free, (8:8) ID	Number of records
72/10	Initiation	0	(0:8) fix, (8:8) update
73/0	Put record	(0:8) error, (8:8) ID	(0:3) rec type, (3:13) number of records
73/1	Delete rec	(0:8) error, (8:8) ID	(0:3) rec type (3:13) number of records
73/2	Delete blk	Start of file block #	End of file block #

Notes:

1. The aa/bb notation in the "octal value" column denotes type/subtype. Type is the actual MMSTAT event number. Subtype is (0:4) of parameter 0.

2. Several items can possibly exceed their fields, in that case the bits beyond the field are lost. These items are number of records, number of free records, start of file, and end of file.

3. Parameter word zero has a common format for all the MMSTAT events.

Field	Description
(0:4)	Event's subtype.
(4:2)	File's state 0 - empty 1 - partially full 2 - only a fraction of a free record is left 3 - completely full
(6:1)	Nonzero indicates that there is one or more waiting readers.
(7:1)	Nonzero indicates that there is one or more waiting writers.
(11:1)	Nonzero indicates that the write has a carriage control character.
(12:4)	Flags local to the accessor. (12:1) - the accessor has done no FREADs/FWRITEs (13:1) - extended wait (14:1) - nondestructive read (15:1) - writer has not written any records

{File System Basic IPC Definitions}

The objective of this set of uncallable procedures is to provide a simple ipc mechanism to support the ipc file access procedures. It enables one process to send short, control messages to another process.

{General behavior}

{FCPORTOPEN procedure}

The heart of this mechanism is the port. A process desiring to receive messages would first open (create) a port. This process is termed the "port manager." When the port is created, a port number is returned to the opener. Since the port number value cannot be known in advance, potential senders need some method of obtaining the port number from the port manager.

Both the ports and the messages are contained in a single disc resident data segment. There can be a total of over thiry-five hundred open ports and outstanding messages Thus neither ports nor message blocks are scarce resources.

{FCPORTSEND procedure}

This procedure sends a 0 to 5 word message to a port. Optionally a timeout value may be specified which will limit the duration the message will remain attached to the port. Expiration of the timeout causes the message to be deleted from the target port's queue and placed on the sender's reply port (specified by the sender in the FCPORTSEND procedure call).

{FCPORTRECEIVE}

Reads and deletes the head message from a port. The sender's return port number is also given to the receiver, enabling him to send a reply message.

{FCPORTCLOSE}

Demolishes the port.

{IPC File's Use of the IPC Mechanism}

All open message files have two ports open for the file (read wait queue and write wait queue), plus one port per accessor (reply port). Their use is described in the following.

{Reader and writer wait queues}

When an empty message file is accessed by more than one reader (share), then there must be a way of having the readers' FREADs satisfied in the same order that they were issued. That is, there must be queue of waiting readers. The ipc access procedures accomplish this by dedicating a basic ipc port as a "read wait queue." Whenever a reader's request is stalled because the file is empty, a message is sent to the read wait queue. Subsequent FREADs by other processes will queue up behind the first reader in a FIFO manner. An FWRITE will take the first entry from the wait queue and send a "read may be done" message to the reader's reply port.

In a like manner multiple writers will queue on the write wait queue when the file is full.

{Completion notification for nowait I/O}

The IOWAIT intrinsic waits for a message to be sent to the reply port (s) of the specified user files.

{Timeouts}

When an accessor encounters a boundary condition (ex, a reader accesses an empty file), it may specify that the condition must be satisfied in x seconds (FCONTROL 4). To this end the ipc access procedures merely issue the FCPORTSEND to the wait queue with the user's timeout value specified. The timeout will tear the message from the wait queue and place it on the accessor's reply port.

{Port Data Structures}

{Port data segment}

```
                       ............
  System DB extension  :Port DST #: .......
  + %100               ............       :
                                          :
                                          :
                                          :
                    ................      :
                    :              :<-----:
  Port data segment : Global area  :
                    :              :
                    z              z
                    :..............:
                    :              :
                    : Remainder is :
                    : composed of  :
                    : "block size" :
                    : chunks.      :
                    :..............:
```

The chunks are a combination of free entries, ports, message queue entries, and timer list entries.

{Port with two outstanding messages}

```
          ..............      .............      .............
          :            :----->:           :----->:           :.......
          : Port       :      : MQE 1     :      : MQE 2     :      .
          :            :      :           :      :           :    -----
          ..............      .............      .............     ---
                                                                    -
```

{Port number}

```
 0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16
...:..:..:..:..:..:..:..:..:..:..:..:..:..:..:..:...
:Port index :  Port data segment relative addr/8   :
....................................................
```

Port index   Index into the port DST number array

{Port DST Number Array}

Located in System DB Extension Area.

```
   ....................................................
64 : Port data segment number                         : 64
   ....................................................
65 : Reserved for a second port segment               : 65
   ....................................................
```

{Port Data Segment Global Area}

```
    .....................................................
 0 :  Data segment number of this port data segment   : 0
    .....................................................
 1 :  Block size in words                             : 1
    .....................................................
 2 :  Total number of blocks                          : 2
    .....................................................
 3 :  Maximum number of blocks                        : 3
    .....................................................
 4 :  Current number of free blocks                   : 4
    .....................................................
 5 :  Number of open ports                            : 5
    .....................................................
 6 :  Head of free list                               : 6
    .....................................................
 7 :  Tail of free list                               : 7
    .....................................................
10 :  Head of impeded process list                    : 8
    .....................................................
11 :  Tail of impeded process list                    : 9
    .....................................................
12 :  Head of timeout thread (TQE address)            : 10
    .....................................................
13 :  TRLX of timeout                                 : 11
    .....................................................
14 :  Value returned by TIMER intrinsic when          : 12
    ...................             .....................
15 :  Timeout was initiated.                          : 13
    .....................................................
16 :  Head of port list (in units of port numbers).   : 14
    .....................................................
17 :  Not used.                                       : 15
    .....................................................
```

```
   {Port}

   0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
  ...:..:..:..:..:..:..:..:..:..:..:..:..:..:..:...
0 :  Head MQE address                             : 0
  .................................................
1 :  Tail MQE address                             : 1
  .................................................
2 :E :  W  :  Next port number in port list thread: 2
  .................................................
3 :  Soft int subtype     :  Pin of port's owner  : 3
  .................................................
4 :  Soft interrupt parameter one                 : 4
  .................................................
5 :  Number of MQEs in the port's queue           : 5
  .................................................
6 :  Number of sends to this port                 : 6
  .................................................
7 :  Soft interrupt plabel                        : 7
  .................................................
  :0 :1 :2 :3 :4 :5 :6 :7 :8 :9 :10:11:12:13:14:15:
```

E	Enable wake up bit 0 - Do not awaken the process 1 - Awaken the process
W type	Action to be taken on an enabled port when a message is received. 0 - Awaken the process on a message wait bit. 1 - Generate user software interrupt 2 - Generate system software interrupt

{Message Queue Entry (MQE)}

```
     0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16
  ...:..:..:..:..:..:..:..:..:..:..:..:..:..:..:..:...
0 :  Next MQE entry; if last, (port addr) LOR 7   : 0
  .................................................
1 :  Port number of return port                   : 1
  .................................................
2 :Time List Entry (TLE),0=no timeout,-1=timed out: 2
  .................................................
3 :  Parameter zero                               : 3
  .................................................
4 :  Parameter one                                : 4
  .................................................
5 :  Parameter two                                : 5
  .................................................
6 :  Parameter three                              : 6
  .................................................
7 :  Parameter four                               : 7
  .................................................
  :0 :1 :2 :3 :4 :5 :6 :7 :8 :9 :10:11:12:13:14:15:
```

Timer entry definitions - 0 - no timeout
1 - timeout expired
2 - TLE address for a pending timeout

File System Message Files

Wait Message

parm#
0 - WRITER ID
1 - LOCAL FLAGS (differ with each accessor)
  (0:1) - accessor just opened file
  (1:1) - will wait on boundary condition if no symbiotic process
  (3:1) - writer has not written a record
  (4:1) - transmission log in bytes
  (8:1) - carriage control code
2 - DST# of data buffer
3 - Address of data buffer (DST relative)
4 - Length of data buffer in bytes

Completion Message

0 - Resultant error code
1 - Resultant transmission log in bytes

{Timer List Entry (TLE)}

```
     0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
   ...:..:..:..:..:..:..:..:..:..:..:..:..:..:..:..:...
0 :  Next TLE (sorted in incr time val), 0 if last: 0
   ................................................
1 :  Preceding TLE entry (0 if first entry)       : 1
   ................................................
2 :  Number of milliseconds the timeout value     : 2
   ...............              ...................
3 :  of this TLE is beyond the previous TLE.      : 3
   ................................................
4 :  Address of the affected MQE                  : 4
   ................................................
5 :  Address of the MQE's port                    : 5
   ................................................
6 :  Value of TIMER when this timeout expires     : 5
   ............                   .................
7 :  (Milliseconds)                               : 7
   ................................................
   :0 :1 :2 :3 :4 :5 :6 :7 :8 :9 :10:11:12:13:14:15:
```

{MMSTAT Definitions}

Octal Value	Event Type	Parameter 0	Parameter 1	Parameter 2
62	Open	Port number	Port DST num	Flags parameter
63	Receive completion	Port number	MQE address 15:1 Waitspc	Return port
64	Send	Port number	MQE address 15:1 Q type	Return port
65	Change status	Port number	0 = enable 1 = disable	Head MQE address
66	Abort	Port number	Parameter zero	Return port
67	Close	Port number	Port DST	# open ports left
70	Expand	Port DST num	# expand blks	Total # blocks
71	Timeout expired	Port num	MQE address	Return port

CHAPTER 19 MPE Memory Resident Message Facility

I. Overview

The memory resident message facility of MPE IV addresses the need for an efficient, simple, and uniform method for system code to send short status-type messages to processes.

Each process is created with a message harbor which supports a set of message ports which are private to that process. There is a maximum of four ports per harbor in the initial implementation. This limit can be easily extended when new ports are required.

Any system code, even code running on the ICS, can send a message to any port of any process. The destination process' PIN must be known, and a priori conventions on portnumber and message formats must be established. The caller of SENDMSG may optionally specify that the destination process be awakened from a message wait.

The caller of SENDMSG specifies whether the message is to be buffered in the primary message table or the secondary message table. When the secondary table is specified, if the pool of secondary message entries is exhausted, the calling process is queued for a message table entry and blocked until one becomes available. Use of the primary message table is reserved for code running on the ICS or during critical sections (Pdisabled or Disabled intervals) in which it is not possible to release control of the processor to queue for a free message table entry. If the primary table is specified and no free entries are available, the SENDMSG crashes the system.

Messages can be of any length up to the configured maximum. Message length is specified in the call to SENDMSG and RECEIVEMSG. In the initial implementation, messages are limited to 4 words in length. This maximum can be easily increased if the need arises.

By calling PORTSTATUS, a process may at any time determine whether a specified port is non-empty or obtain the portnumber of his most urgent non-empty port (lowest numerical port number =most urgent port).

By calling RECEIVEMSG, a process may receive the message at the head of his specified message port. This receive is optionally non-destructive.

A process can wait on a message wait, or on a combination of message wait and other wait types.

II. Message Intrinsics

A. Procedure SENDMSG(Destpin,Destport,Msglength,Flags);
Value Destpin,Destport,Msglength,Flags;
Integer Destpin,Destport,Msglength;
Option Privileged,Uncallable;
Logical Flags;

Destpin, Destport, and Msglength had better be within range and reasonable (process and port exist), since SENDMSG checks and will crash if the parameters are bad.

The caller of SENDMSG stacks the message contents before calling the procedure. SENDMSG expects the first msg word to be at Q-7-Msglength, and the last msg word at Q-8. The message contents at Q-8 to Q-7-Msglength are deleted from top of stack by the exit from SENDMSG to the caller.

```
Flags.(1:1)=1 ==> Wake-up destination process from a
                  message wait
     .(0:1)=1 ==> place message in secondary message
                  table
```

Return CC=CCG if process was already awake else CC=CCE.

B. Logical Procedure PORTSTATUS(Portnumber);
Value Portnumber;
Integer Portnumber;
Option Privileged,Uncallable;

When supplied a valid port number, PORTSTATUS returns a true value if the port is non-empty and a false value if the port is empty.

When passed a -1 as portnumber parameter, PORTSTATUS returns the portnumber of the process' most urgent non-empty port (the smaller the number, the more urgent the port).

If all ports are empty, PORTSTATUS returns CC=CCE. If at least one port is non-empty, PORTSTATUS returns CC=CCG.

C.

```
Procedure RECEIVEMSG(Portnum,Msglength,Flags);
Value Portnum,Msglength,Flags;
Integer Portnum,Msglength;
Option Privileged,Uncallable;
Logical Flags;
```

Portnum and Msglength had better be within range or else its Suddendeath time.

The caller of RECEIVEMSG does an ADD S Msglength to make space for the message contents. RECEIVEMSG stores the message contents into Q-8,Q-9,...,Q-7-Msglength. Q-7-Msglength contains the first word of the message.

Flags.(0:1)=1 ==> do not release message from head of port's message queue (non-destructive read)

Return CC=CCG if port was empty, else CC:=CCE.

III. Supporting Data Structures

A. Message Harbor Table [DST #57 (%71)]

The message facility is presently used only by the Dispatcher and should not be used by any process. The Message Harbor Table is created during system generation. It is a resident structure, though needn't reside in bank 0. Its base is located through the DST entry which describes it.

```
*................................*
*  LINK TO FIRST MSG PORT 0      *       MESSAGE HARBOR
*--------------------------------*
*  LINK TO FIRST MSG PORT 1      *         TABLE ENTRY
*--------------------------------*
*  LINK TO FIRST MSG PORT 2      *           FORMAT
*--------------------------------*
*  LINK TO FIRST MSG PORT 3      *
*--------------------------------*
*  NON-EMPTY PORT MASK           *
*................................*
```

```
FIRST MSG QUEUE LINK .(0:1) =1 ==> NEXT MESSAGE IN SECONDARY
                                    MESSAGE TABLE
                     .(1:15) = INDEX OF NEXT ENTRY IN
                               APPROPRIATE TABLE
```

B. Message Tables

Prim Msg Tab DST = #58 (%72)
Sec Msg Tab DST = #17 (%21)

There are two types of tables which are used to buffer sent messages, the primary and secondary message tables. The tables are identical in format, but independently configurable with respect to size. Both tables are resident structures, though they needn't be located in bank 0. The bases of the message tables are located by looking up their addresses in the DST entry describing them.

```
**********************************
*  # OF CONFIGURED ENTRIES       *
*--------------------------------*
*  # ENTRY SIZE (5)              *       MESSAGE TABLE
*--------------------------------*
*  # ENTRIES AVAILABLE           *         ENTRY ZERO
```

```
           *-------------------------------*
           *  INDEX OF FIRST FREE ENTRY    *           FORMAT
           *-------------------------------*
           *  PIN OF FIRST IMPEDED PROCESS *
           *...............................*

           *...............................*
           *  NEXT MSG IN QUEUE LINK       *       MESSAGE TABLE
           *-------------------------------*
           *  MSG WORD 1                   *       ASSIGNED ENTRY
           *-------------------------------*
           *  MSG WORD 2                   *           FORMAT
           *-------------------------------*
           *  MSG WORD 3                   *
           *-------------------------------*
           *  MSG WORD 4                   *
           *...............................*


           *...............................*
           *    %100000                    *
           *...............................*
           *  INDEX NEXT FREE ENTRY        *
           *-------------------------------*       FREE ENTRY
           *  Don't Care                   *
           *-------------------------------*          FORMAT
           *  Don't Care                   *
           *-------------------------------*
           *  Don't Care                   *
           *...............................*

NEXT MSG IN QUEUE LINK .(0:1) =1 ==> NEXT MESSAGE IN SECONDARY
                                     MESSAGE TABLE
                       .(1:15) = INDEX OF NEXT ENTRY IN
                                 APPROPRIATE TABLE
```

C. Message Port Assignments

Message Port 0 : Junk Port (to be used when no message
                            interference can occur.)
Message Port 1 : Reserved (for message facility)
Message Port 2 : Reserved (for message facility)
Message Port 3 : Image Port / deferred IOMESSPROC task

CHAPTER 20 MMSTATS EVENT CATALOG

MMSTATS CATALOG INDEX

EVENT NAME	EVENT NO. DEC.	%		EVENT NAME	EVENT NO. DEC.	%
ALCSTBLK	20	024 (-)	*	FREAD	62	076 (-)
ALLOCMEM	12	014	*	FREADDIR	64	100 (-)
BINREAD	233	351 (-)	*	FREADLABEL	76	114 (-)
BREAK	237	355 (-)	*	FREADSEEK	68	104 (-)
CABORTIO	142	216	*	FRENAME	80	120 (-)
CCLOSE	146	222	*	FSETMODE	72	110 (-)
CCLOSETRACEFILE	154	232	*	FSPACE	69	105 (-)
CCONTROL	152	230	*	FUNLOCK	79	117 (-)
CGARBAGE	7	007	*	FUPDATE	66	102 (-)
CONFIG-INFO	221	335 (-)	*	FWRITE	63	077 (-)
CONFIG-INFO	222	336 (-)	*	FWRITEDIR	65	101 (-)
CONFIG-INFO	223	337 (-)	*	FWRITELABEL	77	115 (-)
COPEN	140	214	*	GIPINTERRUPT	192	300
COPENTRACEFILE	153	231	*	IOBUFTRAP	125	175
CPOLLIST	155	233	*	I/O COMPLETION	111	157 (-)
CREAD	147	223	*	IOWAIT	67	103 (-)
CREAD1	147	240	*	MAKEOC	1	001
CSDRIVER	150	226	*	MONINIT	228	344 (-)
CSIOWAIT	144	220	*	MONOFF	229	345 (-)
CWRITE	149	225	*	PROCESS COMPLETE	211	323 (-)
DC1DC2ACK	231	347 (-)	*	QONSEG	0	000
DEALLOCM	13	015	*	QUIESCE	40	050
DEALCSTBLK	21	025 (-)	*	RELRESOURCES	23	027 (-)
DISKBUGCATCHER	200	310	*	SEGIOINIT	5	005
			*	SIODM-ENTRY	194	302
DISKBUGCATCHER	201	311	*	SIODM-EXIT	195	303
DISKERROR	100	144 (-)	*	SIODONE	6	006
DISKERROR	101	145 (-)	*	SPECCHAR	236	354 (-)
DISKINTRPT	191	277	*	SPECIALRQ	2	002
SOFT'DEATH	120	170	*	SPECREAD	238	356 (-)
			*	START I/O	193	301
DISK TRAFFIC	98	142 (-)	*	SWAPIN	8	010
FCHECK	74	112 (-)	*	SYSPINS	224	340 (-)
FCLOSE	81	121 (-)	*	SYSPINS	225	341 (-)
FCONTROL	71	107 (-)	*	SYSPINS	226	342 (-)
FETCHSEG	4	004	*	SYSPINS	227	343 (-)
FGETINFO	75	113 (-)	*	TERMLOGOFF	235	353 (-)
FLOCK	78	116 (-)	*	TERMLOGON	234	352 (-)
FOPEN/(DA)	60	074 (-)	*	TERMREAD	230	346 (-)
FOPEN/(DA)	61	075 (-)	*	TERMWRITE	232	350 (-)
FPOINT	70	106 (-)	*			

MMSTAT EVENT GROUP 0

MEMORY MANAGER

EVENT 0

EVENT NAME: QONSEG
DESCRIPTION: ABSENCE TRAP ON CODE/DATA SEGMENT

CALLING MODULE: KERNELC
CALLING PROCEDURE(S): QUEUEONSEGMENT

PARAMETER DESCRIPTION

```
P1 = SEGIDENTIFIER.(0:2) = SEG TYPE FIELD
                         = 0   => SEG IS A DATA SEGMENT,
                                  .(2:14) = DST ENTRY NUMBER
                         = 1   => SEG IS AN SL SEGMENT,
                                  .(2:14) = SL ENTRY NUMBER
                         = 2,3 => SEG IS PART OF A PROGRAM,
                                  .(1:7) = PROGRAM INDEX
                                           INTO CSTBLK
                                  .(8:8) = LOGICAL SEGMENT
                                           NUMBER (0-255)
P2 = PCB01(CPCB) - SLL POINTER
P3 = STATUS (IN STACK MARKER) OF CALLING (TRAPPING) SEGMENT
```

EVENT 1

EVENT NAME: MAKEOC
DESCRIPTION: MAKE SEGMENT AN OVERLAY CANDIDATE - RELEASE SEGMENT TO THE POOL OF AVAILABLE SPACE

CALLING MODULE: KERNELC
CALLING PROCEDURE: MAKEOC

PARAMETER DESCRIPTION

```
P1 = SEGIDENTIFIER.(0:2) = SEG TYPE FIELD
                         = 0   => SEG IS A DATA SEGMENT
                                .(2:14) = DST ENTRY NUMBER
                         = 1   => SEG IS AN SL SEGMENT
                                 .(2:14) = SL ENTRY NUMBER
                         = 2,3 => SEG IS PART OF A PROGRAM,
                                 .(1:7) = PROGRAM INDEX
                                 .(8:8) = LOGICAL SEGMENT NUMBER
                                          (0-255)
P2 = 0 (UNUSED)
P3 = 0 (UNUSED)
```

EVENT 2

EVENT NAME: SPECIALRO

DESCRIPTION: REQUEST OF SEGMENT EXPANSION/CONTRACTION, UNLOCK, UNFREEZE, IOUNFREEZE, LOCK, IOFREEZE, FREEZE

CALLING MODULE: KERNELC, KERNELD, ININ

```
CALLING PROCEDURES:  UNLOCKSEG', IOFREEZE', FETCHSEGMENT-(KERNELC)
                     DLSIZE, ZSIZE, GETPXSEG, ALTDSEGSIZE,
                     ALTPXFILESIZE                    -(KERNELD)
                     STACKOVERFLOW                    -(ININ)
```

PARAMETER DESCRIPTION

```
     P1 = SEGIDENTIFIER.(0:2) = SEG TYPE FIELD
                              = 0   => SEG IS A DATA SEGMENT,
                                       .(2:14) = DST ENTRY NUMBER
                               =1   => SEG IS AN SL SEGMENT,
                                       .(2:14) = SL ENTRY NUMBER
                               =2,3 => SEG IS PART OF A PROGRAM,
                                       .(1:7) = PROGRAM INDEX
                                                INTO CSTBLK
                                       .(8:8) = LOGICAL SEGMENT
                                                NUMBER (0-255)
P2 = .(0:1)   =1 => REQUEST IS THROUGH FETCHSEGMENT (TYPES
                    0,1,2)
     .(12:4)    TYPE OF REQUEST
              =  0=> IOFREEZE
              =  1=> FREEZE
              =  2=> LOCK
              =  3=> IOUNFREEZE
              =  4=> UNFREEZE
              =  5=> UNLOCK
              =  6=> DLSIZE EXPANSION
              =  7=> DLSIZE CONTRACTION
              =  8=> PXFIXED EXPANSION
              =  9=> PXFILE EXPANSION
              = 10=> PXFILE CONTRACTION
              = 11=> XDS EXPANSION
              = 12=> XDS CONTRACTION
              = 13=> ZSIZE EXPANSION
              = 14=> ZSIZE CONTRACTION
              = 15=> STACKOVERFLOW
P3 = FOR TYPES (P2.(12:4))
              = 0,2,3,5 => P3.(8:8) = LOCK OR IOFREEZE COUNT
              = 1,4 => P3.(0:8) = FREEZE COUNT
              = 6-15 => REQUESTED SIZE OF AREA IN WORDS
```

EVENT 4

EVENT NAME: FETCHSEG
DESCRIPTION: SEGMENT REQUEST (FOR I/O SYSTEM OR PROCESS)

CALLING MODULE: KERNELC
CALLING PROCEDURE: FETCHSEGMENT

```
     PARAMETER DESCRIPTION
     ---------------------
     P1 = SEGIDENTIFIER.(0:2) = SEG TYPE FIELD
                              = 0   => SEG IS A DATA SEGMENT,
                                       .(2:14) = DST ENTRY NUMBER
                              = 1   => SEG IS AN SL SEGMENT,
                                       .(2:14) = SL ENTRY NUMBER
                              = 2,3=> SEG IS PART OF A PROGRAM,
                                       .(1:7) = PROGRAM INDEX
                                                INTO CSTBLK
                                       .(8:8) = LOGICAL SEGMENT
                                                NUMBER (0-255)
     P2 = REQUESTORID
          .(0:1) = 1    => I/O SYSTEM REQUEST
                           .(8:8) = LDEV #
          .(0:1) = 0    => PROCESS REQUEST
                           .(8:8) = PIN # OF REQUESTING PROCESS
          .(1:1) = 1    => IOFREEZE REQUEST
          .(2:1) = 1    => BLOCKED LOCK REQUEST
          .(3:1) = 1    => LOCK REQUEST
          .(4:1) = 1    => FREEZE REQUEST
     P3=  .(13:3)= 0    => SEGMENT ALREADY PRESENT
                 = 1    => SEGMENT IS RECOVERABLE OVERLAY CANDIDATE
                 = 2    => SEGMENT ALREADY ON ITS WAY IN FOR SOMEONE
                 = 3    => SEGMENT NOT PRESENT -- MUST FETCH
```

EVENT 5

EVENT NAME: SEGIOINIT
DESCRIPTION: MEMORY MANAGEMENT READ/WRITE OF SEGMENT FROM/TO DISC QUEUED

CALLING MODULE: KERNELC
CALLING PROCEDURES: PROCESSINITMSG, STARTSEGWRITE

```
PARAMETER DESCRIPTION
---------------------
P1 = SEGIDENTIFIER.(0:2) = SEG TYPE FIELD
                         = 0   => SEG IS A DATA SEGMENT,
                                  .(2:14) = DST ENTRY NUMBER
                         = 1   => SEG IS AN SL SEGMENT,
                                  .(2:14) = SL ENTRY NUMBER
                         = 2,3 => SEG IS PART OF A PROGRAM,
                                  .(1:7)  = PROGRAM INDEX
                                            INTO CSTBLK
                                  .(8:8)  = LOGICAL SEGMENT
                                            NUMBER (0-255)
P2 = DISCREQUEST INDEX - INDEX INTO THE DISC REQUEST TABLE
                         (SYSDB RELATIVE)
P3 = .(0:1) = 1  => WRITE START
            = 0  => READ START
     .(2:15) = LDEV #
```

EVENT 6

EVENT NAME: SIODONE
DESCRIPTION: MEMORY MANAGEMENT SEGMENT READ/WRITE FROM/TO DISC COMPLETE

CALLING MODULE: KERNELC
CALLING PROCEDURES: SEGREADCOMPLETOR, SEGWRITECOMPLETOR

PARAMETER DESCRIPTION

```
P1 = SEGIDENTIFIER.(0:2) = SEG TYPE FIELD
                         = 0   => SEG IS A DATA SEGMENT,
                                  .(2:14) = DST ENTRY NUMBER
                         = 1   => SEG IS AN SL SEGMENT,
                                  .(2:14) = SL ENTRY NUMBER
                         = 2,3=> SEG IS PART OF A PROGRAM,
                                  .(1:7)  = PROGRAM INDEX
                                            INTO CSTBLK
                                  .(8:8) = LOGICAL SEGMENT
                                            NUMBER (0-255)
P2 = DISCREQUEST INDEX - INDEX INTO THE DISC REQUEST TABLE
                         (SYSDB RELATIVE)
P3 = .(0.1) = 1  => WRITE COMPLETE
            = 0  => READ COMPLETE
```

EVENT 7 (%7)

EVENT NAME: CGARBAGE
EVENT DESCRIPTION: GARBAGE COLLECTION HAS JUST TAKEN PLACE

CALLING MODULE: KERNELC
CALLING PROCEDURE: COLLECTGARBAGE

PARAMETER DESCRIPTION

P1 = BANK OF SOURCE JUST MOVED FROM
P2 = ADDR OF SOURCE JUST MOVED FROM
P3 = MOVEPAGECNT, NUMBER OF PAGES JUST MOVED FROM

EVENT 8 (%10)

EVENT NAME: SWAPIN
DESCRIPTION: SWAP IN A PROCESS

CALLING MODULE: KERNELC
CALLING PROCEDURE: SWAPIN

PARAMETER DESCRIPTION

```
P1 = PIN OF PROCESS BEING SWAPPED IN
P2 = .(0:1) =  0 => BEING SWAP
            =  1 => END SWAP
     .(1:1) =  0 => NORMAL (PARTIAL SWAP OK)
            =  1 => SWAP REQUIRED
     .(12:4)=  0 => PROCESS SWAPIN COMPLETE
               2 => NO ROOM, HARD REQ MAY SUCCEED
               3 => NO ROOM, HARD REQ FAILED
               4 => SWAPIN STOPPED - MORE URGENT ACTIVITY
               8 => NO LOCK SPACE
P3 = HARDREQUEST = TRUE => HARD REQUEST ON SWAPIN
                   FALSE=> NORMAL
```

MMSTAT EVENT GROUP 1

MEMORY MANAGER

EVENT 12 (%14)

EVENT NAME: ALLOCMEM
DESCRIPTION: FOUND A HOLE FOR A SEGMENT REPLACEMENT REQUEST

CALLING MODULE: KERNELC
CALLING PROCEDURE: RESERVEREGION

PARAMETER DESCRIPTION

P1 = REQUESTED SIZE IN PAGES
P2 = BANK OF SELECTED REGION
P3 = ADDRESS OF SELECTED REGION

EVENT 13 (%15)

EVENT NAME: DEALLOCM
DESCRIPTION: RELEASE REGION OF MEMORY TO AVAILABLE STATUS

CALLING MODULE: KERNELC
CALLING PROCEDURE: RELEASEREGION

PARAMETER DESCRIPTION

P1 = SIZE RELEASED IN PAGES
P2 = BANK OF RELEASED REGION BASE
P3 = ADDRESS OF RELEASED REGION BASE

```
*****************************************************************
*                                                               *
*                                                               *
*                  MMSTAT EVENT GROUP 2                         *

                   MEMORY MANAGER
*                                                               *
*                                                               *
*****************************************************************
```

EVENT -20 (-%24)

EVENT NAME: ALCSTBLK
DESCRIPTION: REQUEST TO RESERVE A BLOCK OF ENTRIES IN THE CSTX

CALLING MODULE: KERNELD
CALLING PROCEDURE: ALCSTBLOCK

PARAMETER DESCRIPTION

P1=EIX	CST BLOCK INDEX ASSIGNED
P2=CSTX	DST RELATIVE INDEX OF WORD 0 OF THE FIRST RESERVED CSTX ENTRY
P3=N	NUMBER OF CSTX ENTRIES RESERVED

EVENT -21 (%25)

EVENT NAME: DEALCSTBLK
DESCRIPTION: INDICATES THAT A CST EXTENSION BLOCK HAS BEEN DEALLOCATED

CALLING MODULE: KERNELD
CALLING PROCEDURE: DEALCSTBLOCK

PARAMETERS	PARAMETER DESCRIPTION
P1=EIX	CST BLOCK INDEX ASSIGNED TO THE BLOCK OF CST ENTRIES
P2=CSTX	DST RELATIVE INDEX OF WORD 0 OF THE FIRST CST ENTRY TO BE RELEASED
P3=MCNT	=(#ALLOCATED CSTX ENTRIES- #ENTRIES BEING RELEASED)*4

EVENT -23 (-%27)

EVENT NAME:RELRESOURCES
DESCRIPTION: RESOURCES (VDS,MAIN MEMORY, ST ENTRY) RESERVED FOR THE
FOR THE SEGMENT HAVE BEEN RELEASED

CALLING MODULE: KERNELD

CALLING PROCEDURE: RELDATASEG

PARAMETERS PARAMETER DESCRIPTION

P1=NEW DB DST NUMBER
P2=DELTA P AT EXCHANGEDB CALL

P3=STATUS AT EXCHANGEDB CALL

MMMSTAT EVENT GROUP 3

(NOT CURRENTLY ASSIGNED)

MMSTAT EVENT GROUP 4

SCHEDULING

EVENT 40 (%50)

EVENT NAME: QUIESCE
DESCRIPTION: PROCESS SWITCH - STATE OF PROCESS SAVED

CALLING MODULE: KERNELC
CALLING PROCEDURE: DSP

PARAMETER DESCRIPTION

```
P1 = PCB00(CPCB)
     .(0:1) =  1   => SAR   - SCHEDULING ATTENTION REQUIRED
     .(2:1) =  1   => CRIT  - PROCESS IS CRITICAL
     .(3:1) =  1   => HSIR  - PROCESS HAS SIR
     .(4:1) =  1   => PIOVR - PENDING PI, PROCESS CRITICAL
     .(5:1) =  1   => HSPRI - HOLD SIR PRIORITY
     .(6:1) =  1   => IPEXP - INCORE PROTECT EXPIRED
     .(7:1) =  1   => PC    - PREMPT CAPABILITY
     .(8:1) =  1   => MP    - MUST PREMPT
     .(9:1) =  1   => LW    - LONG WAIT
     .(10:1)=  1   => SW    - SHORT WAIT
     .(11:1)=  1   => TRW   - TERMINAL READ WAIT
     .(12:1) =1   =>  USEQD - USED A QUANTUM SINCE TRANSACTION
                              BEGAN
     .(13:1)=  1   => HIPRI - HOLD IMPEDED PRIORITY
     .(14:1)=  1   => ALLOW SOFT INTERRUPTS EVEN THOUGH IN
                      SYSTEM CODE
     .(15:1)=  1   => RITBK - PROCESS IN RIT BREAK
```

```
P2 = PCB04(CPCB)
          .(0:1) =  1  => M     - MOURNING WAIT
          .(1:1) =  1  => RG    - GLOBAL RIN WAIT
          .(2:1) =  1  => RL    - LOCAL RIN WAIT
          .(3:1) =  1  => MA    - MAIL WAIT
          .(4:1) =  1  => BIO   - BLOCKED IO WAIT
          .(5:1) =  1  => IO    - IO WAIT
          .(6:1) =  1  => UCP   - UCOP WAIT, RIT WAIT
          .(7:1) =  1  => JNK   - JUNK WAIT
          .(8:1) =  1  => TIM   - TIMER WAIT
          .(9:1) =  1  => INT   - INTERRUPT WAIT
          .(10:1)=  1  => SON   - SON WAIT
          .(11:1)=  1  => FA    - FATHER WAIT
          .(12:1)=  1  => IMP   - PROCESS WAITING TO UNIMPEDED
          .(13:1)=  1  => SIR   - PROCESS WAITING FOR SIR
          .(14:1)=  1  => TIM   - PROCESS WAITING FOR TIME OUT
          .(15:1)=  1  => MEM   - PROCESS WAITING FOR MEMORY

P3 = PCB13(CPCB)
          .(0:1) =  1  => DISPQ - PROCESS ON DISPATCHING QUEUE

          .(1:1) =  1  => L SCHEDULING CLASS
          .(2:1) =  1  => C SCHEDULING CLASS
          .(3:1) =  1  => D SCHEDULING CLASS
          .(4:1) =  1  => E SCHEDULING CLASS
          .(5:1) =  1  => INTER- PROCESS IS INTERACTIVE
          .(6:1) =  1  => CORER- PROCESS IS CORE-RESIDENT
          .(8:8) =  PROCESS' SCHEDULING PRIORITY
```

```
***************************************************************
*                                                             *
*                  MMSTAT EVENT GROUP 6                       *

*                          FILESYS                            *

*        THESE EVENTS ARE FOR DEVELOPMENT USE ONLY            *
***************************************************************
```

EVENT -60(%74)

EVENT NAME: FOPEN
DESCRIPTION: OLD FILE OPEN

CALLING MODULE: FILEACC

CALLING PROCEDURE: FOPENDA

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	(0:2)=2 -> NON-SPOOLER ACCESS (0:2).NE.2 ->
P2= AOPTIONS	SEE INTRINSICS MANUAL
P3= FILE LABEL FOPTIONS	SEE INTRINSICS MANUAL

EVENT -61(%75)

EVENT NAME: FOPEN'
DESCRIPTION: OLD DISC FILE OPEN (CONTINUATION OF EVENT -60)

CALLING MODULE: FILEACC

CALLING PROCEDURE: FOPENDA

PARAMETERS	PARAMETER DESCRIPTION
P1= RECORD SIZE	
P2= FILE LABEL BLOCK SIZE	
P3= # OF BUFFERS	

EVENT -61(%75)

EVENT NAME: FOPEN'
DESCRIPTION: OLD FILE OPEN (CONTINUATION OF EVENTS -60 & -61)

CALLING MODULE: FILEACC

CALLING PROCEDURE: FOPENDA

PARAMETERS	PARAMETER DESCRIPTION	
P1=	FILE LABEL FILE LIMIT	MSW
P2=	FILE LABEL FILE LIMIT	LSW
P3=	FILE LABEL # OF EXTENTS	

EVENT -60(%74)

EVENT NAME: FOPEN
DESCRIPTION: NEW DISC FILE OPEN

CALLING MODULE: FILEACC

CALLING PROCEDURE: FOPEN

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	(0:2)=2 -> NON-SPOOLER ACCESS (0:2).NE.2 ->
P2= AOPTIONS	SEE INTRINSICS MANUAL
P3= FOPTIONS	SEE INTRINICS MANUAL

EVENT -61(%75)

EVENT NAME: FOPEN'
DESCRIPTION: NEW DISC FILE OPEN (CONTINUATION OF EVENT -60)

CALLING MODULE: FILEACC

CALLING PROCEDURE: FOPEN

PARAMETERS	PARAMETER DESCRIPTION
P1= RECORD SIZE	
P2= BLOCK SIZE	
P3= # OF BUFFERS	

EVENT -61(%75)

EVENT NAME: FOPEN'
DESCRIPTION: NEW DISC FILE OPEN (CONTINUATION OF EVENT -60 & -61)

CALLING MODULE: FILEACC

CALLING PROCEDURE: FOPEN

PARAMETERS PARAMETER DESCRIPTION

P1= FCB FILE LIMIT

P2= FCB MAX # EXTENTS

P3= (0:8)= INITIAL ALLOCATION EXTENTS

EVENT -62(%76)

EVENT NAME: FREAD
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FREAD

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	(0:1) BUFFER HIT FLAG
P2= ACBTLOG	TRANSFER COUNT
P3= NOT USED	

EVENT -63(%77)

EVENT NAME: FWRITE
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FWRITE

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	(0:1) BUFFER HIT FLAG
P2= TCOUNT	SEE INTRINSIC MANUAL
P3= NOT USED	

EVENT -64(%100)

EVENT NAME: FREADDIR
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FREADDIR

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	(0:1) BUFFER HIT FLAG
P2= ACBTLOG	TRANSFER COUNT
P3= NOT USED	

EVENT -64(%100)

EVENT NAME: FREADDIR'
DESCRIPTION: CONTINUATION OF EVENT -64 FREADDIR

CALLING MODULE: FILEIO

CALLING PROCEDURE: FREADDIR

PARAMETERS	PARAMETER DESCRIPTION
P1= REC #	MSW
P2= REC #	LSW
P3= NOT USED	

EVENT -65(%101)

EVENT NAME: FWRITEDIR
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING MODULE: FWRITEDIR

PARAMETERS	PARAMETER DESCRIPTION
P1= FILENUM	(0:1) BUFFER HIT FLAG
P2= TCOUNT	SEE INTRINSIC MANUAL
P3= NOT USED	

EVENT -65(%101)

EVENT NAME: FWRITEDIR'
DESCRIPTION: CONTINUATION OF EVENT -65 FWRITEDIR

CALLING MODULE: FILEIO

CALLING PROCEDURE: FWRITEDIR

PARAMETERS	PARAMETER DESCRIPTION
P1= REC #	MSW
P2= REC #	LSW
P3= NOT USED	

EVENT -66(%102)

EVENT NAME: FUPDATE
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FUPDATE

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	(0:1) BUFFER HIT FLAG
P2= TCOUNT	SEE INTRINSIC MANUAL
P3= NOT USED	

EVENT -67(%103)

EVENT NAME: IOWAIT
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: IOWAIT

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	(0:1) BUFFER HIT FLAG
P2= ACBTLOG	TRANSFER COUNT
P3= NOT USED	

EVENT -68(%104)

EVENT NAME: FREADSEEK
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FREADSEEK

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	(0:1) BUFFER HIT FLAG
P2= REC #	MSW
P3= REC #	LSW

EVENT -69(%105)

EVENT NAME: FSPACE
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FSPACE

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= DISPLACEMENT	SEE INTRINSIC MANUAL
P3= NOT USED	

MMSTAT EVENT GROUP 7

FILESYS

THESE EVENTS ARE FOR DEVELOPMENT USE ONLY

EVENT -70(%106)

EVENT NAME: FPOINT
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FPOINT

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= REC #	MSW
P3= LSW	LSW

EVENT -71(%107)

EVENT NAME: FCONTROL
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FCONTROL

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= CODE	SEE INTRINSIC MANUAL
P3= NOT USED	

EVENT -72(%110)

EVENT NAME: FSETMODE
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FSETMODE

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= MODEFLAGS	SEE INTRINSIC MANUAL
P3=	

EVENT -74(%112)

EVENT NAME: FCHECK
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FCHECK

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= ERRORCODE	SEE INTRINSIC MANUAL
P3= 0	

EVENT -75(%113)

EVENT NAME: FGETINFO
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FGETINFO

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= FOPTIONS	SEE INTRINSIC MANUAL
P3= AOPTIONS	SEE INTRINSIC MANUAL

EVENT -76(%114)

EVENT NAME: FREADLABEL
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE:

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= TCOUNT	SEE INTRINSIC MANUAL
P3= 0	

EVENT -77(%115)

EVENT NAME: FWRITELABEL
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FWRITELABEL

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= TCOUNT	SEE INTRINSIC MANUAL
P3= 0	

EVENT -78(%116)

EVENT NAME: FLOCK
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FLOCK

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= LOCKCOND	SEE INTRINSIC MANUAL
P3= COND CODE	SEE INTRINSSIC MANUAL

EVENT -79(%117)

EVENT NAME: FUNLOCK
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FUNLOCK

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= 0	
P3= 0	

```
***********************************************************
*                                                         *
*                 MMSTAT EVENT GROUP 8                    *
*                                                         *
*        THESE EVENTS ARE FOR DEVELOPMENT USE ONLY        *
*                                                         *
***********************************************************
```

EVENT -80(%120)

EVENT NAME: FRENAME
DESCRIPTION:

CALLING MODULE: FILEACC

CALLING PROCEDURE: FRENAME

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= 0	
P3= 0	

EVENT -81(%121)

EVENT NAME: FCLOSE
DESCRIPTION:

CALLING MODULE: FILEACC

CALLING PROCEDURE: FCLOSE

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= DISP	SEE INTRINSIC MANUAL
P3= SECCODE	SEE INTRINSIC CODE

```
**************************************************************
*                                                            *
*                  MMSTAT EVENT GROUP 9                      *
*                                                            *
*                  DISC I/O TRANSFER REQUESTS                *

*         THESE EVENTS ARE FOR DEVELOPMENT USE ONLY          *
**************************************************************
```

EVENT -98(%142)

EVENT NAME: DISK TRAFFIC
DESCRIPTION: DISC I/O REQUEST HAS BEEN QUEUED

CALLING MODULE: HARDRES

CALLING PROCEDURE: ATTACHIO

PARAMETERS	PARAMETER DESCRIPTION
P1=CNT	DATA TRANSFER COUNT:WORDS IF >0; BYTES IF <0
P2=FLAGS.(0:4)	
P3=FNCT	=0 ==>READ =1 ==>WRITE =2 ==>OPEN FILE =3 ==>CLOSE FILE =4 ==>CLOSE DEVICE

MMSTAT EVENT GROUP 10

DISC ERRORS

EVENT 100(%144)

EVENT NAME: DISK ERROR
DESCRIPTION: RECORD DISC ERROR

CALLING MODULE: IOFDISC1

CALLING PROCEDURE: FHDDVR

PARAMETERS	PARAMETER DESCRIPTION
P1=DIPT(DSTAT)	HARDWARE STATUS
P2=S0	QMISC
P3=IOQP(QLDEV).QLDEVN LOR STOCOUNT&LSL(8))	=LDEV/SIO PROGRAM COUNTER

EVENT 101(%145)

EVENT NAME: DISK ERROR
DESCRIPTION: RECORD DISC ERROR

CALLING MODULE: IOMDISC0

CALLING PROCEDURE: MHDDVR

PARAMETERS	PARAMETER DESCRIPTION
P1=DIPT(DSTAT)	HARDWARE STATUS
P2=S0	QMISC
P3=IOQP(QLDEV).QLDEVN LOR STOCOUNT&LSL(8))	=LDEV/SIO PROGRAM COUNTER

MMSTAT EVENT GROUP 11

SIO

EVENT -110(%156)

EVENT NAME: START I/O
DESCRIPTION:DRIVER INITIATOR FOR SIO DEVICE HAS BEEN CALLED

CALLING MODULE: HARDRES

CALLING PROCEDURE: SIODM

PARAMETERS   PARAMETER DESCRIPTION

P1=IOQPL(QSTAT) LOR IOQPL(QLDEV).LDEVN
=(0:8) PCB ENTRY # OF PROCESS MAKING REQUEST
(8:8) LOGICAL DEVICE NUMBER OF DEVICE FOR I/O
P2=IOQP(QWBCT)=WORD COUNT IF>0;BYTE COUNT IF<0
P3=(0:2) = FUNCTION CODE SPECIFIED BY DRIVER

= 0 => READ
= 1 => WRITE
= 2 => CONTROL

=(6:10)= DSTN OF TARGET DATA SEG

EVENT -111(%157)

EVENT NAME: I/O COMPLETION
DESCRIPTION: SIO COMPLETION

CALLING MODULE: HARDRES

CALLING PROCEDURE: SIODM

PARAMETERS   PARAMETER DESCRIPTION

P1=IOQP(QLDEV).LDEVN=LOGICAL DEVICE NUMBER OF DISC INVOLVED IN TRANSFER
P2=IOQP(QPAR1)   (DEFINED BY DRIVER)
P3=IOQP(QPAR2)   (DEFINED BY DRIVER)

```
***************************************************************
*                                                             *
*                                                             *
*                 MMSTAT EVENT GROUP 12                       *

                  SOFT DEATH
*                                                             *
*                                                             *
***************************************************************
```

EVENT 120(%170)

EVENT NAME: SOFT'DEATH
DESCRIPTION: BUG CATCHER

CALLING MODULE: HARDRES

CALLING PROCEDURE: SOFT'DEATH

PARAMETERS	PARAMETER DESCRIPTION
P1	SOFT'DEATH I.D. NUMBER
P2	CALLERS STATUS REGISTER
P3	CALLERS DELTA P

EVENT 125 (%175)

EVENT NAME: IOBUFTRP
EVENT DESCRIPTION: IOSYSTEM BUFFER TRAP

CALLING MODULE: HARDRES
CALLING PROCEDURE: SIODM

PARAMETER DESCRIPTION

P1 = IOQP
P2 = IOQP(QDSTN).DSTN = DST NUMBER OF BUFFER
P3 = 0

MMSTAT EVENT GROUP 13

MPE I/O DISC ATTACHIO INFO

EVENT -130 (-%202)

EVENT NAME: ATTACHIO disc
DESCRIPTION: Additional ATTACHIO disc info to supplement group 9.

CALLING MODULE: Unknown

CALLING PROCEDURE: Unknown

PARAMETERS	PARAMETER DESCRIPTION
P1	LDEV# of disc
P2	P-offset of calling code segment
P3	STATUS register of caller

EVENT -131 (-%203)

EVENT NAME: ATTACHIO disc
DESCRIPTION:

CALLING MODULE: Unknown

CALLING PROCEDURE: Unknown

PARAMETERS	PARAMETER DESCRIPTION
P1	High-order file extent base sector address (if instrumented).
P2	Low-order file extent base sector address (if instrumented).
P3	Extent size in sectors (if instrumented).

EVENT -132 (-%204)

EVENT NAME: ATTACHIO disc
DESCRIPTION:

CALLING MODULE: Unknown

CALLING PROCEDURE: Unknown

PARAMETERS	PARAMETER DESCRIPTION
P1	P1 or high-order sector address of requested transfer.
P2	P2 or low-order sector address of requested transfer.
P3	FLAGS word of ATTACHIO call, where
.(0:4) =	0 Unknown I/O requestor 1 general file sys, no instrumentation 2 spooler, no instrumentation 3 directory I/O 4-7 unassigned as of Q-MIT 8 GENMESSAGE, where extent base is is message-set base and extent size is message-set sectors. 9 File sys, BUF, FQUIESCEIO. 10 File sys, NOBUF, sequential 11 File sys, NOBUF, direct access 12 File sys, BUF, sequential 13 File sys, BUF, direct access 14 File sys, KSAM 15 File sys, IMAGE

MMSTAT EVENT GROUP 14

CS/3000

EVENT 140 (%214)

EVENT NAME: COPEN
DESCRIPTION:

CALLING MODULE: COMSYS2

CALLING PROCEDURE: COPEN

PARAMETERS PARAMETER DESCRIPTION

P1 (0:8) = CS ERROR CODE
(8:8) = LOGICAL DEVICE NUMBER

P2 PMAP1

P3 PMAP2

EVENT 142 (%216)

EVENT NAME: CABORTIO
DESCRIPTION:

CALLING MODULE: COMSYS1

CALLING PROCEDURE: CABORTIO

PARAMETERS	PARAMETER DESCRIPTION
P1	LOGICAL DEVICE
P2	IOQINDEX
P3	0

EVENT 144 (%220)

EVENT NAME: CSIOWAIT
DESCRIPTION:

CALLING MODULE: COMSYS1

CALLING PROCEDURE: CSIOWAIT

PARAMETERS	PARAMETER DESCRIPTION
P1	(0:8) = CS ERROR CODE (8:8) = LOGICAL DEVICE NUMBER
P2	TRANSMISSION LOG
P3	

EVENT 146 (%222)

EVENT NAME: CCLOSE
DESCRIPTION:

CALLING MODULE: COMSYS3

CALLING PROCEDURE: CCLOSE

PARAMETERS	PARAMETER DESCRIPTION
P1	(0:8) = CS ERROR CODE (8:8) = LOGICAL DEVICE NUMBER
P2	LINE NUMBER
P3	0

EVENT 147 (%223)

EVENT NAME: CREAD
DESCRIPTION:

CALLING MODULE: COMSYS4

CALLING PROCEDURE: CREAD

PARAMETERS	PARAMETER DESCRIPTION
P1	(0:8) = CS ERROR CODE (8:8) = LOGICAL DEVICE NUMBER
P2	INCOUNT
P3	STATION

EVENT 149 (%225)

EVENT NAME: CWRITE
DESCRIPTION:

CALLING MODULE: COMSYS4

CALLING PROCEDURE: CWRITE

PARAMETERS	PARAMETER DESCRIPTION
P1	(0:8) = CS ERROR CODE (8:8) = LOGICAL DEVICE NUMBER
P2	OUTCOUNT
P3	INCOUNT

```
******************************************************************
*                                                                *
*                                                                *
*                  MMSTAT EVENT GROUP 15                         *
*                                                                *
*                  CS/3000                                       *
******************************************************************
```

EVENT 150 (%226)

EVENT NAME: CSDRIVER
DESCRIPTION:

CALLING MODULE: BSCLCM

CALLING PROCEDURE: CSDRIVER

PARAMETERS		PARAMETER DESCRIPTION
P1	TIMER	LSW
P2	CURRENTSTATE	WHERE THE DRIVER IS IN THE STATE TRANSITION TABLE
P3	CURRENTEVENT	(0:8) = CURRENT EVENT (8:8) = LOGICAL DEVICE WHAT CAUSED THE DRIVER TO BECOME ACTIVE

EVENT 152 (%230)

EVENT NAME: CCONTROL
DESCRIPTION

CALLING MODULE: COMSYS5

CALLING PROCEDURE: CCONTROL

PARAMETERS	PARAMETER DESCRIPTION
P1	(0:8) = CS ERROR CODE (8:8) = LOGICAL DEVICE NUMBER
P2	CONTROL CODE
P3	PARAMETER

EVENT 153 (%231)

EVENT NAME: COPENTRACEFILE
DESCRIPTION:

CALLING MODULE:

CALLING PROCEDURE: COPENTRACEFILE

PARAMETERS	PARAMETER DESCRIPTION
P1	(0:8) = CS ERROR CODE (8:8) = LOGICAL DEVICE NUMBER
P2	CTRACEINFO
P3	0

EVENT 154 (%232)

EVENT NAME: CCLOSETRACEFILE
DESCRIPTION:

CALLING MODULE:

CALLING PROCEDURE: CCLOSETRACEFILE

PARAMETERS	PARAMETER DESCRIPTION
P1	(0:8) = CS ERROR CODE (8:8) = LOGICAL DEVICE NUMBER
P2	0
P3	0

EVENT 155 (%233)

EVENT NAME: CPOLLIST
DESCRIPTION:

CALLING MODULE:

CALLING PROCEDURE: CPOLLIST

PARAMETERS	PARAMETER DESCRIPTION
P1	LOGICAL DEVICE
P2	CS ERROR CODE
P3	PMAP

```
**************************************************************
*                                                            *
*                                                            *
*                  MMSTAT EVENT GROUP 16                     *

                   CS/3000
*                                                            *
*                                                            *
**************************************************************
```

EVENT 160(%240)

EVENT NAME: CREAD
DESCRIPTION:

CALLING MODULE: DSMON

CALLING PROCEDURE:

```
PARAMETERS      PARAMETER DESCRIPTION

    P1= TIME STAMP

    P2= (0:4) NOT USED
        (4:1) BLOCK
        (5:2) STATE
        (7:3) NEXT
        (10:1) :=0  INITIALIZATION EVENT
               :=1  COMPLETION EVENT
        (11:5) SUB EVENT NUMBER

    P3= DEPENDS ON THE SUB EVENT NUMBER AND
        IF ITS A INTIALIZATION OR COMPLETION EVENT.
        MSG: (0:4)   STRMTYPX
             (4:6)   MSG CLS
             (10:16) STRMTYP
```

SUB EVENT NO.	SUB EVENT NAME	INIT PARM	COMP PARM
0	CREAD	0	LEN
1	CWRITE	X MSG	LEN
2	IOWAIT	0	LEN
3	CCHECK	0	ERRCOD
4	DSATTN	0	0
5	DSWC	X MSG	R MSG
6	CHNGEWAIT	PARM	0
7	MONREQ	REQ	0
10	CABORT	0	T/F
11	CRESET	0	0
12	CSDATA	R MSG	
13	CSREREAD		

MMSTAT EVENT GROUP 19

DISC CONTROLLER INTERRUPT

EVENT 191(%277)

EVENT NAME: DISKINTRPT
DESCRIPTION: A 7905/7920 CONTROLLER IS PROCESSING AN ATTENTION INTERRUPT (ONLINE/OFFLINE)
CALLING MODULE: HARDRES

CALLING PROCEDURE: SIODM

PARAMETERS	PARAMETER DESCRIPTION
P1= @DITP	(US)--ie.WHO GOT THE INTERRUPT
P2= @DITP	(THEM)--ie. WHO RAN THE POLL PROGRAM
P3= DITP	"OUR" DIT FLAGS WORD

THERE SHOULD BE AT LEAST AN %300 AND AN %303 FOR EACH SIO PRGM. A SINGLE ISOLATED (IN TIME) REQUEST WILL GENERATE AT LEAST A %303, %300, %303. IF THE QUEUE OF IOQE'S ON A DIT NEVER EMPTIES, THERE WOULD BE ONE %300 AND ONE %303 PER SIO PRGM.

EVENT 192(%300)

EVENT NAME: GIPINTERUPT
DESCRIPTION: INTERRUPT JUST PROCESSED

CALLING MODULE: HARDRES

CALLING PROCEDURE: GIP

PARAMETERS	PARAMETER DESCRIPTION
P1= (0:7)	LDEV note a) its easy to read in octal b) ldevs > 127 will be recorded mod 128
(8:9)	ADDRESS CONTAINED IN DRT WORD 0 RELATIVE TO SIO PROGRAM AREA (ie where did it stop?) ABS(DRTN*4)-(ILTP(ISIOP)+SYSDB))
P2= DEVICE STATUS	(the TIO GIP just did)
P3= LSW of a call to TIMER	

EVENT 193(%301)

EVENT NAME: STARTIO
DESCRIPTION: Issuing SIOP machine instruction.

CALLING MODULE: HARDRES

CALLING PROCEDURE: START'HPIB, STARTIO

PARAMETERS	PARAMETER DESCRIPTION
P1=	DRT number.
P2=	Absolute address of SIO program to start.
P3=	LSW of TIMER

EVENT 194(%302)

EVENT NAME: SIODM-ENTRY
DESCRIPTION: Entering SIODM

CALLING MODULE: HARDRES

CALLING PROCEDURE: SIODM

```
PARAMETERS      PARAMETER DESCRIPTION

    P1= (0:7) LDEV  --  SAME AS 192(%300)

        (8:9)    a IOQ table relative index
                 to convert this into the number that
                 is formated by DPAN2, multiply this num-
                 ber by %13 and add %10,that will be the
                 number in the left column of returned
                 IOQ'S-- add the table base to get the
                 DPAN number for "in-use" enries.


    P2= DIT WORD 0 (DIT FLAGS) -- note that P2.(12:4)
                   contains the state we are "leaving"

    P3= (0:4)  THE CONTENTS OF DIT0.(12:4)
               ie, the state we entered in

        (4:12) LSW OF TIMER -- note the difference
                               between P3 of %300 and
                               P3 of %303, these 12 bits
                               will hold ~4.1 seconds which
                               is enough for 30229 con-
                               trollers purpose and DS
                               timeouts (some types).
```

EVENT 195(%303)

EVENT NAME: SIODM-EXIT
DESCRIPTION: Leaving SIODM main loop.

CALLING MODULE: HARDRES

CALLING PROCEDURE: SIODM

PARAMETERS PARAMETER DESCRIPTION

P1= (0:7) LDEV -- SAME AS 192(%300)

(8:9) a IOQ table relative index to convert this into the number that is formated by DPAN2, multiply this number by %13 and add %10,that will be the number in the left column of returned IOQ'S-- add the table base to get the DPAN number for "in-use" enries.

P2= DIT WORD 0 (DIT FLAGS) -- note that P2.(12:4) contains the state we are "leaving"

P3= (0:4) THE CONTENTS OF DIT0.(12:4) ie, the state we entered in

(4:12) LSW OF TIMER -- note the difference between P3 of %300 and P3 of %303, these 12 bits will hold ~4.1 seconds which is enough for 30229 controllers purpose and DS timeouts (some types).

```
***************************************************************
*                                                             *
*                                                             *
*                 MMSTAT EVENT GROUP 20                       *

                  PRIVATE VOLUMES
*                                                             *
*        THESE EVENTS ARE FOR DEVELOPMENT USE ONLY            *
*                                                             *
***************************************************************
```

EVENT 200(%310)

EVENT NAME: DISKBUGCATCHER
DESCRIPTION:

CALLING MODULE: PVSYS

CALLING PROCEDURE: MVTABLE

PARAMETERS	PARAMETER DESCRIPTION
P1= FUNCT	
P2= MVTABX	
P3= DELTAP	

EVENT 201(%311)

EVENT NAME: DISKBUGCATCHER
DESCRIPTION:

CALLING MODULE: PVSYS

CALLING PROCEDURE: USERTABLE

PARAMETERS	PARAMETER DESCRIPTION
P1= FUNCT	
P2= MVTABX	
P3= DELTAP	

```
****************************************************************
*                                                              *
*                                                              *
               MMSTAT EVENT GROUP 21                      *

*              PROCESS CREATIONS AND TERMINATIONS              *

*              LOGICAL PROCESS TABLE                           *
*                                                              *
****************************************************************

                             EVENT -211(%323)

EVENT NAME: PROCESS COMPLETION
DESCRIPTION: PROCESS HAS TERMINATED

  CALLING MODULE: MORGUE

      CALLING PROCEDURE: TERMINATE

                 PARAMETERS      PARAMETER DESCRIPTION

                     P1=0
                     P2=0
                     P3=0
```

MMSTAT EVENT GROUP 22

MONITOR CONFIGURATION INFORMATION

TIME STAMP OF EVENT TRACE ENABLE AND DISABLE

EVENT 221(%335)

EVENT NAME: CONFIGURATION INFORMATION
DESCRIPTION: EVENT GROUP MASK

CALLING MODULE: CRIO

CALLING PROCEDURE: CONSMON

PARAMETERS PARAMETER DESCRIPTION

P1= MEASMSK0

P2= MEASMSK1

P3=

EVENT 222(%336)

EVENT NAME: CONFIGURATION INFORMATION
DESCRIPTION: MPE VERSION FIX UPDATE

CALLING MODULE: OPCOMMAND

CALLING PROCEDURE: CXMON

PARAMETERS	PARAMETER DESCRIPTION
P1= VERSION	
P2= FIXL	
P3= UPDATEL	

EVENT -223 (-%337)

EVENT NAME: CONFIGURATION INFORMATION
DESCRIPTION: SYSTEM TABLE LOCATIONS AND AVAILABLE LINKED MEMORY INFORMATION
CALLING MODULE: OPCOMMAND

CALLING PROCEDURE: CXMON

PARAMETERS   PARAMETER DESCRIPTION

```
P1=F(%1032)=@CST(0)-@DST(0)
           =DISPLACEMENT TO CODE
P2=F(%1033)=@CST(LAST)-@DST(0)
           =DISPLACEMENT TO SHARABLE
P3=LOGICAL(TOTAL&DLSK(4))=LINKED MEMORY SIZE
```

EVENT -224 -(%340)

EVENT NAME: SYSPINS
DESCRIPTION: LOGICAL PROCESS TABLE

CALLING MODULE: OPCOMMAND

CALLING PROCEDURE: CXMON

PARAMETERS PARAMETER DESCRIPTION

P1=ABSOLUTE(%1141)=PROGEN'S PCBENTRY NUMBER
P2=ABSOLUTE(%1142)=MAM'S PCB ENTRY NUMBER
P3=ABSOLUTE(%1143)=UCOP'S PCB ENTRY NUMBER

EVENT -225 (-%341)

EVENT NAME: SYSPINS(CNTD.)
DESCRIPTION: LOGICAL PROCESS TABLE

CALLING MODULE: OPCOMMAND

CALLING PROCEDURE: CXMON

PARAMETERS PARAMETER DESCRIPTION

P1=ABSOLUTE(%1144)=PFAIL'S PCB ENTRY NUMBER
P2=ABSOLUTE(%1145)=DEVREC'S PCB ENTRY #
P3=ABSOLUTE(%1146)=PRMSG'S PCB ENTRY #

EVENT -226 (-%342)

EVENT NAME: SYSPINS(CNTD.)
DESCRIPTION: LOGICAL PROCESS TABLE

CALLING MODULE: OPCOMMAND

CALLING PROCEDURE: CXMON

PARAMETERS PARAMETER DESCRIPTION

P1=ABSOLUTE(%1147)=STMSG'S PCB ENTRY #
P2=ABSOLUTE(%1150)=LOG'S PCB ENTRY #
P3=ABSOLUTE(%1151)=LOAD'S PCB ENTRY #

EVENT -227 (-%343)

EVENT NAME: SYSPINS(CNTD.)
DESCRIPTION: LOGICAL PROCESS TABLE

CALLING MODULE: OPCOMMAND

CALLING PROCEDURE: CXMON

PARAMETERS PARAMETER DESCRIPTION

P1=ABSOLUTE(%1152)=IOMESSPROC'S PCB ENTRY #
P2=ABSOLUTE(%1153)=SYSIOPROC'S PCB ENTRY #
P3=ABSOLUTE(%1154)=MEMLOGP'S PCB ENTRY #

EVENT -228 (%344)

EVENT NAME: TIMESTAMP
DESCRIPTION: TIMESTAMP

CALLING MODULE: OPCOMMAND

CALLING PROCEDURE: CXMON

PARAMETERS PARAMETER DESCRIPTION

P1=CALENDER (0:7)=YEAR OF CENTURY
(7:9)=DAY OF YEAR
P2=CLOCK(WORD1).(0:7)=HOUR OF DAY
(8:8)=MINUTE OF HOUR
P3=CLOCK(WORD2).(0:7)=SECONDS INTO MINUTE
.(8:8)=TENTHS OF SECONDS

EVENT -229 (-%345)

EVENT NAME: MONOFF
DESCRIPTION: END EVENT TRACING

CALLING MODULE: OPCOMMAND

CALLING PROCEDURE: CXMON

PARAMETERS PARAMETER DESCRIPTION

P1=0
P2=0
P3=0

MMSTAT EVENT GROUP 23

TERMINAL I/O

EVENT 230 (%346)

EVENT NAME: TERMREAD
DESCRIPTION: TERMINAL READ COMPLETION

CALLING MODULE: HARDRES
CALLING PROCEDURE: TIP

PARAMETERS PARAMETER DESCRIPTION

P1 = LDEV
P2 = READ DURATION
P3 = BYTES READ

EVENT 231 (%347)

EVENT NAME: DC1DC2ACK
DESCRIPTION: DC1/DC2 HAS BEEN SATISFIED

CALLING MODULE: HARDRES
CALLING PROCEDURE: TIP

PARAMETERS PARAMETER DESCRIPTION

P1 = LDEV
P2 = DURATION (BETWEEN START AND DC2)
P3 = BYTES READ (EXCLUDING DC2)

EVENT 232 (%350)

EVENT NAME: TERMWRITE
DESCRIPTION: WRITE COMPLETION

CALLING MODULE: IOTERMO
CALLING PROCEDURE: TERMIOM

PARAMETERS PARAMETER DESCRIPTION

P1 = LDEV
P2 = 0
P3 = BYTE COUNT OF TRANSFER

EVENT 233 (%351)

EVENT NAME: BINREAD
DESCRIPTION: BINARY READ COMPLETED

CALLING MODULE: HARDRES
CALLING PROCEDURE: TIP

PARAMETERS PARAMETER DESCRIPTION
P1 = LDEV
P2 = DURATION
P3 = BYTES READ

EVENT 234 (%352)

EVENT NAME: TERMLOGON
DESCRIPTION: TERMINAL JUST LOGGING ON

CALLING MODULE: IOTERMO
CALLING PROCEDURE: TERMIOM

PARAMETERS	PARAMETER DESCRIPTION
P1 = LDEV	
P2 = 0	
P3 = 0	

EVENT 235 (%353)

EVENT NAME: TERMLOGOFF
DESCRIPTION: TERMINAL JUST LOGGED OFF

CALLING MODULE: IOTERMO
CALLING PROCEDURE: TERMIOM

PARAMETERS	PARAMETER DESCRIPTION
P1 = LDEV	
P2 = 0	
P3 = 0	

EVENT 236 (%354)

EVENT NAME: SPECCHAR
DESCRIPTION: PROCESSED SPECIAL CHARACTER

CALLING MODULE: HARDRES
CALLING PROCEDURE: TIP

PARAMETERS	PARAMETER DESCRIPTION
P1 = LDEV	
P2 = SPECIAL CHARACTGER PROCESSED	
P3 = 0	

EVENT 237 (%355)

EVENT NAME: BREAK
DESCRIPTION: PROCESSED BREAK

CALLING MODULE: HARDRESS
CALLING PROCEDURE: TIP

PARAMETERS	PARAMETER DESCRIPTION
P1 = LDEV	
P2 = DSTATE	
P3 = 0	

EVENT 238 (%356)

EVENT NAME: SPECREAD
DESCRIPTION: SPECIAL READ TERMINATION CHARACTER DETECTED

CALLING MODULE: HARDRES
CALLING PROCEDURE: TIP

PARAMETERS	PARAMETER DESCRIPTION
P1 = LDEV	
P2 = DURATION	
P3 = BCNT	

CHAPTER 21 DATA COMMUNICATIONS TABLES

LOWER LEVEL DS/3000 TABLES

DATA COMMUNICATIONS IOQ ENTRY

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
   0|                 (SEE BELOW)                   |
    |-----------------------------------------------|
   1|                   NEXT IOQP                   |
    |-----------------------------------------------|
   2|        UNIT #          |         QLDEVN       |
    |-----------------------------------------------|
   3|           IOQ STATN/LAST STATN REF            |
    |-----------------------------------------------|
   4|S |                BUFFER DST                  | S=STACKFLAG
    |-----------------------------------------------|
   5|                 BUFFER1 ADDR                  |
    |-----------------------------------------------|
   6|                        |        FUNCTION      |
    |-----------------------------------------------|
   7|                  COUNT/TLOG                   |
    |-----------------------------------------------|
   8|          BUFFER2 ADDR/CONTROL CODE            |
    |-----------------------------------------------|
   9|              TCOUNT2/PARAMETER                |
    |-----------------------------------------------|
  10|       USER PCBN        |    I/O QS    | I/O GS | QS=QUALIFIED STATUS
    |-----------------------------------------------| GS=GENERAL STATUS
```

BIT0 ABORT
BIT3 SYS BUFFER
BIT4 IO WAKE
BIT5 BLOCKED
BIT6 COMPLETED
BIT7 DATA FROZEN
BIT8 MAM ERROR
BIT 10 SFAIL
BIT11 PFAIL
BIT14 TIMER
BIT15 MSG ERROR

GS 0=PENDING
GS 1=SUCCESSFUL
GS 2=END OF TRANSMISSION
RECEIVED
GS 3=UNUSUAL CONDITION

Word 0 .ABORT - Abort this I/O request
.SYSBUF - Data is in system buffers
.IOWAKE - Wake caller upon completion
.BLOCKED - Blocked I/O, do blocked AWAKE when I/O is complete
.COMPLETE - Request has been completed
.DATAFRZN - The DST has been frozen

.MAMERRORD - MAM failed to freeze the DST
.SFAIL - The I/O program failed to start due to no SIO OK
.PFAIL - The Abort bit was set because of a power failure
.TIMED - An I/O timeout request has completed
.MSGDONE - A message reply has been completed

DS Line Control Block (DSLCB)

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
    |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
  0 |                      0                        |
    |-----------------------------------------------|
    |                                               |
    |                                               |
    |                                               |
    |                                               |
    |                                               |
    |                                               |
    |                                               |
    |                                               |
    |                                               |
    |                                               |
    |                                               |
    |                                               |
    |                                               |
    |                                               |
    |                                               |
    |                                               |
    |                                               |
    |                                               |
    |                                               |
    |                                               |
    |===============================================|
```

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
    -------------------------------------------------
    |AB| R| L|              | $STDLIST or ERRCODE   |
    -------------------------------------------------
```

ENTRY FORMAT

```
       AB - Set when a line error occurs (ABORT bit)
        R - Request is in progress
        L - Line opened
    (8:8) - If AB set then ERRCODE, else if session
            then $STDLIST LDEV, else a zero.
```

NOTES

Contained in DSMON'S DL-DB area Line Control Block. One entry is created for each DSOPEN main process, otherwise the entry is set to zero. DSLCB table size (in words) is the number of PCB's in system+1. Indexed by PIN.

DSGLOBAL DATA SEGMENT

```
    |================================================|
 0  |   DSGLOBINFO TABLE BASE OFFSET  (SEG. REL)     |
    |------------------------------------------------|
 1  |   DSXREF TABLE BASE OFFSET      (SEG. REL)     |
    |------------------------------------------------|
 2  |   DSDEVICE TABLE BASE OFFSET    (SEG. REL)     |
    |------------------------------------------------|
    |                                                | (-1)
    |                                                | ( 0)
    |          DS GLOBAL INFORMATION TABLE           |
    |                                                |
    |                                                |
    |------------------------------------------------|
    |                                                | (-1)
    |                                                | ( 0)
    |                                                |
    |                                                |
    |                                                |
    |                 DSXREF TABLE                   |
    |                                                |
    |                                                |
    |                                                |
    |------------------------------------------------|
    |                                                | (-1)
    |                                                | ( 0)
    |                                                |
    |                                                |
    |                DSDEVICE TABLE                  |
    |                                                |
    |                                                |
    |                                                |
    |                                                |
    |                                                |
    |================================================|
```

NOTES

The DSGLOBAL data segment is referenced by a DST number stored at SYSDB+%320. All tables in this segment have a standard format which require the negative oneth and zeroth entries to contain the number of entries and an entry size respectively. Segment relative table bases point to entry zero.

DSGLOBINFO

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
|             NUMBER OF REAL ENTRIES            | (-1)
|-----------------------------------------------|
|                   ENTRY SIZE                  | ( 0)
|-----------------------------------------------|
|    NUMBER OF CONFIGURED DS LINES IN SYSTEM    | ( 1)
|-----------------------------------------------|
|//|RS|RE|RM|//|  DS CAPABILITY MASK (WORD)     | ( 2)
|-----------------------------------------------|
|          DS INFO DATA SEGMENT NUMBER          | ( 3)
|===============================================|
```

NOTES

This table is used to hold information which is global in respect to the DS/3000 software within the system.

        CURRENT NUMBER OF ENTRIES: 3
        CURRENT ENTRY SIZE (WORDS): 1

RS - Remote can use sequence numbers.
RE - Remote can use exclusive mode w/o excl. protocol.
RM - Remote supports multi-packet algorithm.

DSXREF

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
|              NUMBER OF REAL ENTRIES           | (-1)
|-----------------------------------------------|
|                    ENTRY SIZE                 | ( 0)
|-----------------------------------------------|
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|===============================================|
```

DSXREF TABLE NOTES (DS1)

The DSXREF table will contain an entry for each process in the system. For pseudo terminals for which a session exists, the entry corresponding to the main PIN contains one of the preceding single word entry formats. If this is a master request, the Request Control Word is contained as the DSXREF entry. If the current request is initiated by a slave, the RCW is contained in word 9 of the IODSTRM0 (pseudo terminal) DIT.

Request state:
   0 = Command out
   1 = PTOP out
   2 = PTOP in break mode; reply pending
   3 = PTOP in break mode.

SLAVE ENTRY FORMAT (DS1)

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
-------------------------------------------------
| 1| P|                 | SLAVE PSEUDOTERM LDEV |
|-----------------------------------------------|
|                      0                        !
!-----------------------------------------------!
!                      0                        !
-------------------------------------------------
```

MASTER ENTRY FORMAT (DS1)

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
-------------------------------------------------
| 0| P|  |PB|REQ. STATE | CURR MASTER REQ. LDEV | RCW
|-----------------------------------------------|
|                      0                        |
|-----------------------------------------------|
|                      0                        |
-------------------------------------------------
```

P   - Inhibit next breakmode request across line.
PB - Current BREAK request issued by PCLOSE.
RCW - Request control word.

DSXREF table notes (X.25)

The DSXREF table will contain an entry for each process in the system. The current entry size is 3 words, which contains Request Control Word (RCW), Line Control Word (LCW) and Saved IOQ Word (SIOQW).

The format for master (or slave master) entries is:

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
        |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
 RCW    | 0|  |  |PB| REQ. STATE| Curr master req LDEV. |
        |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|

        |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
 LCW    |  | R| Num. son proc.  | Curr master req UC.   |
        |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|

        |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
 SIOQW  |         save IOQ index   ( PTOP break )       |
        |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
```

If the current request is initiated by a slave process (slave (acting as a master), and the process number is equal to the mainpin then the RCW, LCW and SIOQW are contained in IODSTRMX DIT.

```
                  IODDSTRMX DIT

          ------------------------------
          |              .             |
          |              .             |
          |----------------------------|
          |             RCW            | %11 TMSTR
          |----------------------------|
          |              .             |
          |----------------------------|
          |             LCW            |  17 TLCW
          |----------------------------|
          |              .             |
          |----------------------------|
          |            SIOQW           |  20 TSIOQW
          |----------------------------|
          |              .             |
          |              .             |
          ------------------------------
```

For pseudo terminals for which a session exists, the entry corresponding to the mainpin in DSXREF table contains the following:

```
        0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
      |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
RCW   | 0|  |  |PB| REQ. STATE| Slave Pseudo Term LDEV|
      |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|

      |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
LCW   |  | R|                 |       UC number       |
      |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|

      |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
SIOQW |         save IOQ index  ( PTOP break )        |
      |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
```

Notes:

R - Request in progress
PB - Current break request issued by PCLOSE break
Num. son proc. - Current number of sons processing DSbreak (Break, Abort, Resume).

REQ. STATE -

Master or Slave Master States

0 = Command Out
1 = PTOP Out
2 = PTOP in break mode reply pending
3 = PTOP in break mode, flow issued
4 = PTOP in break mode
5 = PTOP in break, resume issued
6 = PTOP in break, saved IOQ on line
7 = PTOP in break, abort issued

Slave States

0 = Null
1 = PTOP in break
2 = PTOP in break flow not issued
3 = PTOP in break flow issued
4 = PTOP in break Resume/Abort received
5 = PTOP in break PCLOSE BREAK received

DSDEVICE

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
|             NUMBER OF REAL ENTRIES            | (-1)
|-----------------------------------------------|
|                   ENTRY SIZE                  | ( 0)
|-----------------------------------------------|
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|                                               |
|===============================================|
```

DSDEVICE TABLE ENTRY

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
-------------------------------------------------
|          0           | DEV. SUB. |DEVICE TYPE|
-------------------------------------------------
```

NOTES

The DSDEVICE table is initialized during system startup to contain a device type corresponding to the device's relationship to DS/3000, and a device subtype (DS1 or DS1.5). The number of real entries corresponds to the number of LDEVs configured for the system. Entry size is one word.

DEVICE TYPES:
  0 = Non DS/3000 related device.
  1 = DS/3000 related CS device.
  2 = DS device.
  3 = DS pseudo terminal.
  4 = DS PAD pseudo device.

DEV. SUB. = DEVICE SUBTYPES:
  0 = DS1
  1 = DS1.5 (X25)

DS LPDT ENTRY

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
|//|        DIT pointer (SYSDB-relative)        |
|-----------------------------------------------|
|STATE|JA|DA|CY|DU|IN|   EOF    |BR| C|  SUBTYPE |
|===============================================|
```

STATE - Device recognition state

	(For DS device)	(For virtual terminal)
0	Available (for use)	Available (not owned)
1	Not available	Owned or recognized
2	:DSCONTROL device lock	DEVREC service request
3	not used	DEVREC service granted

(For virtual terminals only)

J - Job accepting

DA - Data accepting

CY - Control Y detected

DU - Duplicative

IN - Interactive
(For DS device) Remote side can compress data

EOF - End of file condition
0 - No :EOF
1 - Hardware EOF
2 - :DATA
3 - :EOD
4 - :HELLO
5 - :BYE
6 - :JOB
7 - :EOJ

BR - (For virtual terminal) Break detected
(For DS device) DSMON not created

C - Default is data compression.

SUBTYPE - (For DS device)
0 - default is no data compression
1 - default is data compression

IODS0 DIT

```
        0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
       |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
    #0 | 0| 0| A|NR| Q|  | P|  | D| N|PS|STATE| X| M| S| %0
       |-----------------------------------------------|
     1 |             NEXT DITP (ALWAYS 0)              |  1
       |-----------------------------------------------|
     2 |                     IOQP                      |  2
       |-----------------------------------------------|
     3 |      UNIT NUMBER      |         LDEV          |  3
       |-----------------------------------------------|
     4 |                     DLTP                      |  4
       |-----------------------------------------------|
     5 |                    SAVEIOQ                    |  5
       |-----------------------------------------------|
     6 |                   USECOUNT                    |  6
       |-----------------------------------------------|
     7 |        CS LDEV        |        FWDLDEV        |  7
       |-----------------------------------------------|
     8 |         HDLEN         |       MSG CLASS       | 10
       |-----------------------------------------------|
     9 |                       0                       | 11
       |-----------------------------------------------|
    10 | R| E| C| B|           |      STREAM TYPE      | 12
       |-----------------------------------------------|
    11 |                   SUBSTREAM                   | 13
       |-----------------------------------------------|
    12 |     FROM PROCESS      |      TO PROCESS       | 14
       |-----------------------------------------------|
    13 |                RTE TIME STAMP                 | 15
       |-----------------------------------------------|
    14 | MULTI-PACK ACTL RECV CNT (APND+DATA/+BYTES)   | 16
       |-----------------------------------------------|
    15 |            DATA LENGTH  (+ BYTES)             | 17
       |-----------------------------------------------|
    16 |                  HDX1/TEMP1                   | 20
       |-----------------------------------------------|
    17 |                  HDX2/TEMP2                   | 21
       |-----------------------------------------------|
    18 |                  HDX3/TEMP3                   | 22
       |-----------------------------------------------|
    19 |               DSIOM DELAY PARMS               | 23
       |-----------------------------------------------|
    20 |            MORE DSIOM DELAY PARMS             | 24
       |-----------------------------------------------|
    21 |                  DSXREF TEMP                  | 25
       |-----------------------------------------------|
    22 |CR|RS|RE|RM|//| REMOTE'S DS CAP. MASK (WORD)   | 26
       |-----------------------------------------------|
```

IODS0 DIT NOTES

A - Monitor is currently executing
NR- New request has occurred while processing
Q - X21 Queued flag
P - Pre-emptive
R - Reply
E - Reject
C - Continuation
B - Break mode
D - DSMON request bit
N - Keeps DSIOM delay in effect (NULLF)
PS- Primary/Secondary CSline
CR- Capability mask reply bit
RS- Remote can use sequence numbers
RE- Remote can use exclusive mode w/o excl. protocol
RM- Remote supports multi-packet algorithm

STATE - CSline state
  0 = unconnected
  1 = control
  2 = text

X - Exclusive mode valid if set.
M - Master mode valid if set.
S - Slave mode valid if set.

Note: Bits 0 and 1 of the IODS0 DIT must be 0 to fit MPE IO system conventions.

IODSX DIT

```
        0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
      |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
  # 0| 0| 0| A| N|  |  | P|  |  |  |  |  |  | X|  |  |%0 DFLAG
     |------------------------------------------------|
    1|                   NEXT DITP                    | 1 DLINK
     |------------------------------------------------|
    2|                     IOQP                       | 2 DIOQ
     |------------------------------------------------|
    3|      UNIT NUMBER        |         LDEV          | 3 DLDEV
     |------------------------------------------------|
    4|                     DLTP                       | 4 DDLTP
     |------------------------------------------------|
    5|                   SAVEIOQ                      | 5 SAVEIOQ
     |------------------------------------------------|
    6|                   USECOUNT                     | 6 DUSECOUNT
     |------------------------------------------------|
    7|       CS LDEV           |        FWDLDEV        | 7 DLINKDEV
     |------------------------------------------------|
    8| T|        TIME OUT COMPLETED FLAGS             |10 DTIMWD
     |------------------------------------------------|
    9|              TIMER REQUEST INDEX               |11 DTIMINDEX
     |------------------------------------------------|
   10|             DSX DATA SEGMENT DST #             |12 DXDST
     |------------------------------------------------|
   11|         HDLEN           |       MSG CLASS      |13 DXHEADR
     |------------------------------------------------|
   12|              USER CHANNEL NUMBER               |14  1-HDUC
     |------------------------------------------------|
   13| R| E| C| B|             |      STREAM TYPE     |15  2-HDSTRMTYP
     |------------------------------------------------|
   14|                 SUBSTREAM TYPE                 |16  3-HDSUBSTRM
     |------------------------------------------------|
   15|     FROM PROCESS        |      TO PROCESS      |17  4-HDPROCN
     |------------------------------------------------|
   16|                 RTE TIME STAMP                 |20  5-HDRTE
     |------------------------------------------------|
   17|                       0                        |21  6-
     |------------------------------------------------|
   18|            DATA + APPENDAGE LENGTH             |22  7-HDMSGLEN
     |------------------------------------------------|
   19|                  HDX1/TEMP1                    |23 10-HDX1
     |------------------------------------------------|
   20|                  HDX2/TEMP2                    |24 11-HDX2
     |------------------------------------------------|
   21|                  HDX3/TEMP3                    |25 12-HDX3
     |------------------------------------------------|
   22|            REMOTE DS LEVEL NUMBER              |26
     |------------------------------------------------|
   23|  MONG/ MONP/ GETQ/ PUTQ   BUFFER AREA          |27
     |------------------------------------------------|
   24|                BUFFER AREA (1)                 |30
     |------------------------------------------------|
     |                                                |
```

```
  |                       .                       |
  |                       .                       |
  |                       .                       |
  |-----------------------------------------------|
44|                BUFFER AREA (21)               |54
  |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
```

NOTES

A => MONITOR CURRENTLY EXECUTING IF = 1
N => A NEW REQUEST HAS BEEN RECEIVED WHILE EXECUTING IF = 1
P => PRE-EMPTIVE
X => EXCLUSIVE MODE IN EFFECT IF = 1
R => REPLY
E => REJECT
C => CONTINUATION
B => BREAK MODE

HDLEN = HEADER + APPENDAGE LENGTH IN WORDS
HDMSGLEN = DATA + APPENDAGE LENGTH IN BYTES

```
DFLAG.(15:1) = 1 , indicating a IODSX DIT.
                   This bit is reset when the first
                   DSINIT does its work.
```

Bit 0 and 1 of the IODSX DIT must be 0 to fit MPE IO system convention.

IODSTRM0 DIT

```
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
 #0  |T | B|     |SY| A|MC|CB| D| F| S| L| E| R| P| O|  %0
     |-----------------------------------------------|
  1  |                   NEXT DITP                   |  1
     |-----------------------------------------------|
  2  |                     IOQP                      |  2
     |-----------------------------------------------|
  3  |     UNIT NUMBER       |         LDEV          |  3
     |-----------------------------------------------|
  4  |                     DLTP                      |  4
     |-----------------------------------------------|
  5  |                    SAVEIOQ                    |  5
     |-----------------------------------------------|
  6  |                   SYSBUFADR                   |  6
     |-----------------------------------------------|
  7  |       DS LDEV         |        FWDLDEV        |  7
     |-----------------------------------------------|
  8  | M|                 |           TDSTN          | 10
     |-----------------------------------------------|
  9  |           |MAST. STATE|     MASTER LDEV       | 11
     |-----------------------------------------------|
 10  |       MESSAGE CLASS         | STREAM TYPE     | 12
     |-----------------------------------------------|
 11  |      FROM PROC        |        TO PROC        | 13
     |-----------------------------------------------|
 12  |                RTE TIME STAMP                 | 14
     |-----------------------------------------------|
 13  |              RESERVED FOR X.25                | 15
     |-----------------------------------------------|
 14  |              RESERVED FOR X.25                | 16
     |-----------------------------------------------|
 15  |              RESERVED FOR X.25                | 17
     |-----------------------------------------------|
 16  |              RESERVED FOR X.25                | 20
     |-----------------------------------------------|
 17  |       PRINT BUFFER SIZE ON MASTER SYSTEM      | 21
     |===============================================|
```

NOTES

T - Terminal
B - Break
SY - System Read
A - Session being aborted
MC - Master is compressing
CB - Clear break (CLRBRK)
D - Read abort
F - Flush
S - Session
L - LOGF (Set during logon process)
E - Tells terminal driver that line error occurred
R - SYSLOADF (RTE down load in progress)
P - Prompted
O - Pending
M - Terminal pre-empt
TDSTN - Slave DS extra data segment

IODSTRMX DIT

```
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
    |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
# 0| T|TB|EX|RT| O| A|MC|CB|  | F| S|  |TE|CL| P|TT|%0 DFLAG
   |-----------------------------------------------|
  1|                   NEXT DITP                   | 1 DLINK
   |-----------------------------------------------|
  2|                     IOQP                      | 2 DIOQ
   |-----------------------------------------------|
  3|      UNIT NUMBER      |         LDEV          | 3 DLDEV
   |-----------------------------------------------|
  4|                     DLTP                      | 4 DDLTP
   |-----------------------------------------------|
  5|                    SAVEIOQ                    | 5 SAVEIOQ
   |-----------------------------------------------|
  6|                   SYSBUFADR                   | 6 SYSBUFAD
   |-----------------------------------------------|
  7|      DS LDEV          |        FWDLDEV        | 7 DLINKDEV
   |-----------------------------------------------|
  8| M|             |           TDSTN             |10 TDSTN
   |-----------------------------------------------|
  9|               RCW (SLAVE MASTER)              |11 TMSTR
   |-----------------------------------------------|
 10| R| E| C| B| MESSAGE CALSS   | STREAM TYPE     |12 TCLSTYP
   |-----------------------------------------------|
 11|    FROM PROC          |        TO PROC        |13 TPROCN
   |-----------------------------------------------|
 12|                RTE TIME STAMP                 |14 TRTE
   |-----------------------------------------------|
 13| USER CHANNEL TYPE     | USER CHANNEL I.D.     |15 TCHANID
   |-----------------------------------------------|
 14| X.25 XDS OFFSET TO TERMINAL DIT EXTENSION     |16 TXDSTOFF
   |-----------------------------------------------|
 15|             LCW (SLAVE MASTER)                |17 TLCW
   |-----------------------------------------------|
 16|             SIOQW (SLAVE MASTER)              |20 TSIOQW
   |-----------------------------------------------|
 17|             PRINT BUFFER SIZE                 |21 TPBUFSZ
   |-----------------------------------------------|
 18|                PRINT IOQ                      |22  TPIOQ
   |-----------------------------------------------|
 19|              PRINT BUFFER POINTER             |23 TPBUFPNTR
   |-----------------------------------------------|
 20|                 FCLOSE IOQ                    |24 TFCLOSEIOQ
   |-----------------------------------------------|
```

IODSTRMX DIT NOTES

T => TERMINAL = 1
TB => TERMIANL BREAK
EX => EXPANSION REQUEST IN PROGRESS (PRINT)
RT => INCOMING RESET REPLY PENDING
A => SESSION BEING ABORTED (ABORT JOB)
MC => MASTER IS COMPRESSING (not used)
CB => CLEAR BREAK (CLRBRK)
F => FLUSH
S => SESSION
TE => TERMINAL ERROR (LINE ERROR)
CL => INCOMING CLEAR REPLY PENDING
P => PROMPTED
TT => PS TERMINAL TYPE
= 0 IF DS PSEUDO TERMINAL
= 1 IF DS PAD PSEUDO TREMINAL
O => PPENDING
M => TERMINAL PRE-EMPT
R => REPLY
E => REJECT
C => CONTINUATION
B => BREAK MODE
TDSTN - SLAVE DS EXTRA DATA SEGMENT

```
<<*************************************************************>>
<<          PAD TERMINAL DEVICE INFORMATION TABLE (DIT)        >>
<<*************************************************************>>



|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
|1 |UP|AC|  |  |AB|  |IC|SL|RS|IP|E!|Xc|LD|P |1 |  0
|-----------------------------------------------|
|            NEXT DIT POINTER (SYSDB-RELATIVE)  |  1
|-----------------------------------------------|
| SYSDB-REL PTR TO FIRST IOQ ELEMENT ON DIT     |  2
|-----------------------------------------------|
|Fl|NE|NP| UNIT NUMBER  |      LDEV             |  3
|-----------------------------------------------|
|DLTP: SYSDB-REL PTR TO DS LOGICAL DEVICE TABLE |  4
|-----------------------------------------------|
|                     0                         |  5
|-----------------------------------------------|
|  |  |  |  |  |  |  |  |RT|  |DC|LT|Br|Es|EF|  |  6 DRQST
|-----------------------------------------------|
|   DS LDEV             |   FWD LDEV            |  7
|-----------------------------------------------|
|PT|Tr|PC|PF| PTY |LO|RB|Edit in |TMode|LP lev  |  8 DMODEM
|-----------------------------------------------|
| TERMINAL TYPE         |Ec|LogTp|EO|BO|TR|Ti|Bl|  9
|-----------------------------------------------|
|  PAD TERMINAL EXTRA DATA SEGMENT DST#         |  10
|-----------------------------------------------|
|  Utility/temporary storage                    |  11
|-----------------------------------------------|
|  CURRENT PAD TERMINAL XDS SIZE                |  12
|-----------------------------------------------|
|  USER CHANNEL NUMBER (0-255)                  |  13
|-----------------------------------------------|
|  NO. WORDS AVAILABLE IN READ BUFFER           |  14
|-----------------------------------------------|
|  EOFCHECK BUFFER WORD 1                       |  15
|-----------------------------------------------|
|  EOFCHECK BUFFER WORD 2                       |  16
|-----------------------------------------------|
|  EOFCHECK BUFFER WORD 3                       |  17
|-----------------------------------------------|
|                                               |  18
|-----------------------------------------------|
|  DST# PREMPTIVE PRINT PARITY CHANGE BUFFER    |  19
|-----------------------------------------------|
|  MAXIMUM READ TIME  (in seconds)              |  20
|-----------------------------------------------|
|  READ TIME (1st word of double timers)        |  21
|-----------------------------------------------|
|            (2nd word of double timers)        |  22
|-----------------------------------------------|
|DEF TERMTYPE        |             |DSPEED      |  23
```

```
|------------------------------------------------|
|     USER CHANNEL NUMBER                        | 24
|------------------------------------------------|
| REQUEST CODE/STATUS                            | 25
|------------------------------------------------|
|                 IOQ INDEX                      | 26
|------------------------------------------------|
| OUTGOING HEADER DST NUMBER                     | 27
|------------------------------------------------|
| OUTGOING HEADER OFFSET                         | 28
|------------------------------------------------|
| OUTGOING BYTE COUNT (+BYTES)                   | 29
|------------------------------------------------|
| INCOMING HEADER DST NUMBER                     | 30
|------------------------------------------------|
| INCOMING HEADER OFFSET                         | 31
|------------------------------------------------|
| INCOMING HEADER BYTE COUNT (+BYTES)            | 32
|------------------------------------------------|
| OUTGOING APPENDAGE DST NUMBER                  | 33
|------------------------------------------------|
| OUTGOING APPENDAGE OFFSET                      | 34
|------------------------------------------------|
| OUTGOING APPENDAGE + BYTE COUNT                | 35
|------------------------------------------------|
| INCOMING APPENDAGE DST NUMBER                  | 36
|------------------------------------------------|
| INCOMING APPENDAGE OFFSET                      | 37
|------------------------------------------------|
| INCOMING + BYTE COUNT                          | 38
|------------------------------------------------|
|           OUTGOING DATA DST#                   | 39
|------------------------------------------------|
|           OUTGOING DATA OFFSET                 | 40
|------------------------------------------------|
|           OUTGOING + BYTE COUNT                | 41
|------------------------------------------------|
|           INCOMING DATA DST#                   | 42
|------------------------------------------------|
|           INCOMING DATA OFFSET                 | 43
|------------------------------------------------|
|           INCOMING + BYTE COUNT                | 44
|------------------------------------------------|
|           USER MAIN PROCESS ID#                | 45
|------------------------------------------------|
| MULTIPAGE WRITE BUFFER 0 DST#                  | 46
|------------------------------------------------|
| MULTIPAGE WRITE BUFFER 0 WORD LENGTH           | 47
|------------------------------------------------|
| MULTIPAGE WRITE BUFFER 1 DST#                  | 48
|------------------------------------------------|
| MULTIPAGE WRITE BUFFER 1 WORD LENGTH           | 49
|------------------------------------------------|
|BW|        |WBUF0|WBUF1|NOB |F1|A1|F0|A0|R1|R0| 50
|------------------------------------------------|
```

```
|BYTES OF WRITE DATA IN BUFF0 READ XDS             | 51
|-------------------------------------------------|
|BYTES OF WRITE DATA IN BUFF1 READ XDS             | 52
|-------------------------------------------------|
|BYTES OF WRITE DATA IN MULTIPAGE WRITE BUFF0     | 53
|-------------------------------------------------|
|BYTES OF WRITE DATA IN MULTIPAGE WRITE BUFF1     | 54
|-------------------------------------------------|
|SI|                     | Prev. Req. Func. Code| 55
|-------------------------------------------------|
| Previous request IOQ QPAR1 (usually for write)| 56
|-------------------------------------------------|
| Previous request IOQ QPAR2 (usually for write)| 57
|-------------------------------------------------|
|  Timer request list number                       | 58
|-------------------------------------------------|
|  SS Break Character   | Record End Character    | 59
|-------------------------------------------------|
|  STORED READ DST NUMBER                          | 60
|-------------------------------------------------|
|  STORED READ OFFSET                              | 61
|-------------------------------------------------|
|  STORED READ POSITIVE BYTE COUNT                 | 62
|-------------------------------------------------|
```

DIT FIELDS AND THEIR MEANINGS

```
EQUATE GETQ'OFFSET'IN'DIT = 24;

EQUATE DIT'STATUS'WORD = 0;

         <<The 0'th  DIT word contains the following flags and>>
         <<status bits:                                       >>
         <<                                                   >>
         <<  0:1  =  1, indicating a pseudoterminal           >>
         <<  2:1  =  1, indicating driver is active           >>
         <<  5:1  =  1, set by DSKILLJOB to mean ABORTJOB done>>
         <<  7:1  =  1, interrupt received, owe 'resume output>>
         <<  8:1  =  1, suppress CR/LF after a read           >>
         <<  9:1  =  1, reset request received, not processed >>
         << 10:1  =  1, set initial PAD parameters for logon  >>
         << 11:1  =  0, echo !!! after CTRL X                 >>
         << 12:1  =  1, means a read was cancelled by CTRL X  >>
         <<             and must be restarted                 >>
         << 13:1  =  1  if a line disconnect has been sent    >>
         <<             from the network.                     >>
         <<                                                   >>
         << 14:1  =  1, if prespace carriage control in write >>
         <<       =  0, if postspace carriage conts. in write >>
         << 15:1  =  1, indicating a PAD pseudoterminal       >>

  DEFINE DIT'DRIVER'IS'ACTIVE   = PAD'DIT'ARRAY(DIT'STATUS'WORD).(2:1)#;
         <<If set, the driver is working on another request when called.>>

  DEFINE INTERRUPT'NEEDS'ACTION= PAD'DIT'ARRAY.(7:1)#;

  DEFINE SUPPRESS'CR'LF'AFTER'READ
         = PAD'DIT'ARRAY( DIT'STATUS'WORD ).(8:1)#;
         <<If set, suppress CR/LF after a read.  This is passed to the  >>
         <<driver via a read IOQ.  This is set using FSETMODE intrinsic.>>

  DEFINE LINE'RESET'REQUESTED   = PAD'DIT'ARRAY(DIT'STATUS'WORD).(9:1)#;
         <<Set when the completor finds a request for a line reset.     >>
         <<The initiator responds to the request.                       >>

  DEFINE SET'INITIAL'PAD'PARMS = PAD'DIT'ARRAY(DIT'STATUS'WORD).(10:1)#;
         <<Set when the completor detects a terminal logging on. The    >>
         <<initiator then sends a PAD message to initialize the PAD     >>
         <<parameters.                                                  >>

  DEFINE CONTROL'X'ECHO'ON      = PAD'DIT'ARRAY(DIT'STATUS'WORD).(11:1)#;
         <<If set, the driver initiator will send a !!! to the terminal >>
         <<after a control X is sent from the terminal.                 >>


  DEFINE CONTROL'X'RESTART'READ
                               = PAD'DIT'ARRAY(DIT'STATUS'WORD).(12:1)#;

          <<If set, the initiator will restart a read after a CTRL X.   >>
```

```
  DEFINE DIT'PRESPACE'BIT      = PAD'DIT'ARRAY(DIT'STATUS'WORD).(14:1)#;
        <<This bit is set when carriage control characters are to be>>
        <<sent before the write data itself.                        >>

  DEFINE LINE'DISCONNECTED
                               = PAD'DIT'ARRAY(DIT'STATUS'WORD).(13:1)#;


EQUATE SYSDB'REL'NEXT'DIT'POINTER = 1;

         <<This DIT word contains the SYSDB-relative offset to>>
         <<the next DIT.  This is usually 0. It is only used  >>
         <<when several DIT's are using the same system       >>
         <<resource.                                          >>

EQUATE SYSDB'REL'IOQ'POINTER = 2;

         <<The first DIT words contains the SYSDB- relative   >>
         <<offset to the first IOQ element associated with    >>
         <<this DIT.                                          >>

EQUATE UNIT'NUMBER'LDEV = 3;

         <<The third word of the DIT contains the unit number >>
         <<and the logical device number of this terminal. The>>
         <<unit number is assigned by SYSDUMP.                >>

  DEFINE FLUSH = (0:1)#;<<When set, flush writes and return will >>
                        <<have IOQ status 0; reads completed with>>
                        <<IOQ status = %173.  Keep doing this for>>
                        <<all requests until a request 25, clear >>
                        <<flush and write is received.           >>

  DEFINE LDEV = (8:8)#;

EQUATE SYSDB'REL'DLTP = 4;

         <<The fourth word of the DIT contains the SYSDB      >>
         <<relative offset to the DS logical device table.    >>

         <<The fifth word of the DIT contains a 0.  This is   >>
         <<expected by POWERFAIL.                             >>


EQUATE  DRQST  = 6;

  DEFINE READ'TIME'OUT  = PAD'DIT'ARRAY(DRQST).(8:1)#;
       <<This bit is set by TIMERREQ if a timer has expired   >>
       <<on a timed read request.  It is reset by driver      >>
       <<if the read was completed in time.                   >>

  DEFINE DATASEG'CLEANUP = PAD'DIT'ARRAY(DRQST).(10:1)#;
       <<This bit is set after a logon timeout has expired    >>
       <<following a :BYE - we can finally release the PAD    >>
       <<XDS when we get to the completion section.           >>
```

```
  DEFINE LOGON'TIME'OUT = PAD'DIT'ARRAY(DRQST).(11:1)#;
       <<This bit is set by TIMERREQ if a timer has expired    >>
       <<for a logon timeout.  It is reset by the driver if the>>
       <<logon came in time.                                   >>

  DEFINE IN'BREAK'STATE = PAD'DIT'ARRAY(DRQST).(13:1)#;
       <<This bit, if set, indicates that break was allowed    >>
       <<and was found.  This bit is only set by an IOQ with   >>
       <<a function code of 30.  If QPAR1 of this IOQ is odd,  >>
       <<the break state is set.  If QPAR1 of this IOQ is even,>>
       <<the break state is reset, i.e., end of break state.   >>
       <<This is issued via a direct ATTACHIO from the command >>
       <<interpreter.                                          >>


EQUATE OTHER'LDEVS = 7;

      <<The left eight bits are the ldev of the DS device. The right>>
      <<eight bits are the forward pointer to the next DS logical   >>
      <<device.  DSINIT fills these fields.                         >>


EQUATE DMODEM = 8;

  DEFINE DIT'PREEMPT'BIT = PAD'DIT'ARRAY(%10).(0:1)#;
       <<This bit is set when a pre-emptive print is >>
       <<posted to this DIT.  The IOQ is reordered.  >>


  DEFINE TIME'UP = PAD'DIT'ARRAY(DMODEM).(1:1)#;

        <<This bit is set when a timer request expires and the >>
        <<driver is awakened.  This usually happens when a     >>
        <<conversational write is waiting for the next request >>
        <<to actually send the write data.  The timer request  >>
        <<is made to be sure that the data is sent even if no  >>
        <<further request is made or the system is too slow.   >>
        <<It is reset either when the driver is called for a   >>
        <<new write or read request or the driver sends out the>>
        <<write data anyway.                                   >>
        <<The procedure TIMEREQ requires this bit be in this   >>
        <<word.  This bit position is specified by the %21     >>
        <<value of the first parameter of TIMEREQ.             >>

  DEFINE PARITY'CHECK = PAD'DIT'ARRAY(DMODEM).(2:1)#;
        <<If set, this indicates that driver will check incoming>>
        <<parity and generate outgoing parity.                  >>

  DEFINE PARITY'FOUND = PAD'DIT'ARRAY(DMODEM).(3:1)#;
        <<If set, this indicates that the driver has determined >>
        <<the parity of the remote terminal.                    >>
```

```
  DEFINE PARITY = PAD'DIT'ARRAY(DMODEM).(4:2)#;
        <<This holds the value of the parity which was determined>>
        <<by the driver procedure FIND'PARITY.                   >>
        <<                                                       >>
        <<  0:  Parity bit is always 0;                          >>
        <<  1:  Parity bit is always 1;                          >>
        <<  2:  Parity is even;                                  >>
        <<  3:  Parity is odd;                                   >>
        <<                                                       >>
        <<  Note that this driver only supports 0 and 3.         >>

  DEFINE TERMINAL'LOGGED'ON = PAD'DIT'ARRAY(DMODEM).(6:1)#;
        <<This bit is set when the terminal is logged on.  It is >>
        <<initially set to 0 and is reset to 0 whenever the      >>
        <<there is a log off.                                    >>

  DEFINE READ'DATA'BUFFERED = PAD'DIT'ARRAY(DMODEM).(7:1)#;
        <<This bit is set whenever there is read data which has  >>
        <<been buffered awaiting a read.  This happens when the  >>
        <<terminal has received a logon message, but the terminal>>
        <<has not yet been logged on.  It also occurs when an    >>
        <<end of file was found and data must be buffered for the>>
        <<the next read.  This is changed by the procedure       >>
        <<called STORE'INCOMING'MESSAGE.  The address of the     >>
        <<stored data and its byte length are stored in the DIT. >>

  DEFINE EDIT'INPUT = PAD'DIT'ARRAY(DMODEM).(8:3)#;
        <<If 0, don't edit any incoming data characters.         >>
        <<   1, edit incoming data.                              >>
        <<   possibly other cases                                >>


EQUATE TERM'INFO = 9;

  DEFINE USER'TERM'TYPE = PAD'DIT'ARRAY(TERM'INFO).(0:8)#;

  DEFINE LOGON'TYPE     = PAD'DIT'ARRAY(TERM'INFO).(9:2)#;
      <<This contains the logon type.  It is determined      >>
      <<by an IOQ with function code 21.  It values are as   >>
      <<follows:                                             >>
      <<                                                     >>
      <<   0:  DATA, break not enabled.                      >>
      <<   1:  SESSION, break enabled.                       >>
      <<   2:  JOB, break not enabled.                       >>
      <<                                                     >>
      <<When these are set, the logon timeout is stopped.    >>

  DEFINE ECHO'SETTING   = PAD'DIT'ARRAY(TERM'INFO).(8:1)#;
      <<If 0, allow PAD to echo terminal input.              >>
      <<If 1, don't allow PAD to echo.                       >>

  DEFINE CONTROLY'OK    = PAD'DIT'ARRAY(TERM'INFO).(11:1)#;
       <<If this bit is set, subsystem break is enabled.    >>
       <<Note that the subsystem break character need not  >>
       <<be EM (control Y).  This bit is set by an IOQ with >>
```

```
       <<function code 13 (fcontrol 17).  The bit is reset   >>
       <<by an IOQ with function code 12 (fcontrol 16).      >>
       <<These bits are changed in the initiator but provide >>
       <<control information to the completor.               >>

  DEFINE BREAK'OK       = PAD'DIT'ARRAY(TERM'INFO).(12:1)#;
       <<If this bit is set, the driver will allow system    >>
       <<break to take place.  This will be signalled to the >>
       <<the driver by a level 0 interrupt packet to a level >>
       <<1 indication of break PAD message.                  >>
       <<This bit is set by an IOQ with function code 11     >>
       <<(fcontrol 15).  It is reset by IOQ function 10,     >>
       <<(fcontrol 16).                                      >>

  DEFINE READ'TIMER'ON = PAD'DIT'ARRAY(TERM'INFO).(14:1)#;
       <<If this bit is set by an IOQ with function code 17  >>
       <<the time since the last read is being stored in the >>
       <<DIT to be returned when an IOQ with a function 18   >>
       <<is found.  This bit is reset by function codes 16   >>
       <<and 18.                                             >>

  DEFINE READ'TIMEOUT'ENABLED = PAD'DIT'ARRAY(TERM'INFO).(13:1)#;
       <<If this bit is set by an IOQ with function code 5,  >>
       <<all reads are to be timed out.  The time out value  >>
       <<is stored (as seconds) in word 20 of the DIT.  If   >>
       <<QPAR1 of an IOQ with function code 20 is 0, this bit>>
       <<will be reset, indicating that there is no timeout  >>
       <<interval.                                           >>

  DEFINE BLOCK'MODE = (15:1)#;

        <<If this is 1, the terminal is in block mode; if 0 the>>
        <<terminal is in character mode.                       >>


EQUATE PAD'TERMINAL'XDS'DST = 10;

        <<This word contains the DST number of the PAD terminal>>
        <<extra data segment.                                  >>

EQUATE UTILITY = 11;

  DEFINE DIT'UTILITY'WORD = PAD'DIT'ARRAY(UTILITY)#;
        <<This word is a utility word used to transfer single  >>
        <<words between data segments.  It is in the DIT so    >>
        <<that the DB register does not have to be set to the  >>
        <<stack to use a local variable.  In general, when a   >>
        <<single byte must be transferred from one data area   >>
        <<area to another, a whole word is moved to this place >>
        <<and the unwanted byte is removed before the remaining>>
        <<byte plus a null byte is moved to the final location.>>
```

```
EQUATE CURRENT'PAD'TERMINAL'XDS'SIZE = 12;

        <<This word contains the current number of words which >>
        <<are available in the PAD terminal extra data segment.>>


EQUATE USER'CHANNEL'NUMBER = 13;

        <<This word contains the user channel number for the   >>
        <<the current session.  It maximum value is 255.       >>
<<DSIOMX assumes that the word in this position has the UC.    >>

  DEFINE USER'CHANNEL = PAD'DIT'ARRAY(USER'CHANNEL'NUMBER)#;


EQUATE NO'WORDS'IN'FREE'R'W'BUFFER     =  14;

        <<This word contains the CURRENT number of words that  >>
        <<can be placed in the free read/write buffer of the   >>
        <<the PAD terminal extra data segment.                 >>


EQUATE EOF'CHECK'1'WORD = 15;
EQUATE EOF'CHECK'2'WORD = 16;
EQUATE EOF'CHECK'3'WORD = 17;

        <<These three words will hold copies of the first three>>
        <<words of incoming data.  They will be examined by    >>
        <<the procedure EOFCHECK to determine if they contain  >>
        <<an end-of-file indication.                           >>


<<WORD  18   IS   UNUSED PRESENTLY.  ITS VALUE IS     0.       >>

EQUATE PARITY'CHANGE'BUFF = 19;

  DEFINE PRE'EMPT'PARITY'CHANGE'BUFFER =
         PAD'DIT'ARRAY( PARITY'CHANGE'BUFF )#;

        <<This word contains the DST number of an extra data   >>
        <<segment obtained to buffer data while changing the   >>
        <<parity of pre-emptive prints.  This is only acquired >>
        <<when even parity has been detected in incoming logon >>
        <<data. It is released with a device close IOQ.        >>


EQUATE READ'MAX'SECONDS = 20;
  DEFINE MAXIMUM'READ'SECONDS = PAD'DIT'ARRAY(READ'MAX'SECONDS)#;

        <<This word contains the number of seconds allowed for a>>
        <<timed read.  The value is derived from the IOQ(QPAR1) >>
        <<field of a request with function code 5, i.e., set    >>
        <<read time out.                                        >>
```

```
EQUATE READ'TIME'1 = 21;
EQUATE READ'TIME'2 = 22;

          <<These two words are used to hold the timer value    >>
          <<for the time taken to complete a read request.      >>
          <<The double word logical value is determined by the  >>
          <<TIMER intrinsic.  This value is read by an IOQ      >>
          <<request with function code 18.                      >>

EQUATE INITIAL'TERMTYPE'AND'SPEED = 23;

          <<This word contains the terminal type and speed the  >>
          <<is set by INITIAL.  The user terminal type is held  >>
          <<in word labelled TERMINFO.                          >>


<<****************************************************************>>
<<   WORDS 24 THROUGH 45 ARE USED FOR FORMATTING THE GETQ          >>
<<   ELEMENT.  THESE ARE DEFINED EARLIER IN THIS LISTING.          >>
<<****************************************************************>>


  GETQ'ELEMENT'LENGTH   = 22, Length of a GETQ element in words

 The following offsets  refer to the GETQ formatting
area in the PAD terminal DIT.
The values below are DIT relative offsets.

  GETQ'UC'WORD          = 24, <<This holds the user channel #>>
  GETQ'REQ'STATUS'WORD  = 25, <<This holds the request/status>>
                              <<word for the current request >>
                              <<in bits (8:15) for requests  >>
                              <<with IOQ index:              >>
                              <<                             >>
                              << Code  Meaning     Data      >>
                              <<                             >>
                              <<  3    UC clear  clear parms.>>
                              <<  4    UC IO  level 0 data   >>
                              <<  5    UC IO  level 1 data   >>
                              <<                             >>
                              <<Bit positions (0:8) are used >>
                              <<to specify the EOR character >>
                              <<for DSMONX:                  >>
                              <<                             >>
                              <<  CR: 0                      >>
                              <<  RS: 1                      >>
                              <<  any non-printing           >>
                              <<    character : 2            >>
                              <<                             >>
                              <<For responses to requests    >>
                              <<that don't have an IOQ, the  >>
                              <<status to the request is     >>
                              <<coded as follows:            >>
                              <<                             >>
                              << 3: incoming clear completed >>
```

```
                            << 4: incoming interrupt       >>
                            <<    completed                >>
                            << 5: incoming reset completed >>

GETQ’IOQ’INDEX’WORD   = 26, <<The IOQ index of the current >>
                            <<request being formatted      >>

GETQ’O’H’D’1          = 27, <<This holds the first header  >>
                            <<descriptor;                  >>

GETQ’O’H’D’2          = 28, <<The second header            >>
                            <<descriptor                   >>
GETQ’O’H’D’3          = 29, <<The third header             >>
                            <<descriptor                   >>

GETQ’I’H’D’1          = 30, <<Holds the incoming descriptor>>
                            <<for the header               >>

GETQ’I’H’D’2          = 31, <<Holds incoming header        >>
                            <<descriptor                   >>

GETQ’I’H’D’3          = 32, <<Holds incoming header        >>
                            <<descriptor                   >>

GETQ’O’A’D’1          = 33, <<Holds the outgoing appendage >>
                            <<descriptor                   >>

GETQ’O’A’D’2          = 34, <<Holds the outgoing appendage >>
                            <<descriptor                   >>

GETQ’O’A’D’3          = 35, <<Holds the outgoing appendage >>
                            <<descriptor                   >>

GETQ’I’A’D’1          = 36, <<Holds the incoming appendage >>
                            <<descriptor                   >>

GETQ’I’A’D’2          = 37, <<Holds the incoming appendage >>
                            <<descriptor                   >>

GETQ’I’A’D’3          = 38, <<Holds the incoming appendage >>
                            <<descriptor                   >>
GETQ’OUT’DST          = 39, <<Holds the source DST# for    >>
                            <<outgoing data                >>
GETQ’OUT’OFFSET       = 40, <<Holds the offset for outgoing>>
                            <<data                         >>
GETQ’OUT’DATA’COUNT   = 41, <<Holds the data count for     >>
                            <<outgoing data in + bytes     >>

GETQ’IN’DST           = 42, <<Holds the target DST# for    >>
                            <<incoming data                >>
GETQ’IN’OFFSET        = 43, <<Holds the target offset      >>
                            <<for incoming data            >>
GETQ’IN’DATA’COUNT    = 44, <<Holds expected data          >>
                            <<count in + bytes             >>
                            <<for incoming data            >>
```

```
  GETQ'MAINPIN           = 45  <<User main process number     >>

                               <<This is not needed now, but  >>
                               <<may have to be used later.   >>
                               <<Its value is returned in the >>
                               <<corresponding PUTQ.  It will >>
                               <<be set to zero for now.      >>
EQUATE BUFF0'DST'MULTIPAGE'WRITE = 46;

         <<This word holds the DST# of the buffer 0 used to store>>
         <<outgoing multipage VIEW writes.                       >>

EQUATE BUFF0'LENGTH'MULTIPAGE'WRITE = 47;

         <<This word holds the word length of the buffer 0 for   >>
         <<outgoing multipage VIEW writes.                       >>

EQUATE BUFF1'DST'MULTIPAGE'WRITE = 48;

         <<This word holds the DST# of the buffer 1 used to store>>
         <<outgoing multipage VIEW writes.                       >>

EQUATE BUFF1'LENGTH'MULTIPAGE'WRITE = 49;

         <<This word holds the word length of the buffer 1 for   >>
         <<outgoing multipage  VIEW writes.                      >>


EQUATE BUFFER'STATUS'MULTIPAGE = 50;

         <<This word holds status information concerning the use >>
         <<of the multipage write buffers and the read buffer    >>
         <<extra data segment with the fixed length write buffers>>
         <<                                                      >>
         <<Meaning of the fields:                                >>
         <<                                                      >>

         <<  8:2  Next outgoing write buffer                     >>
         <<       Values:                                        >>
         <<                                                      >>
         <<       0: buffer 0 in read extra data segment         >>
         <<       1: buffer 1 in read extra data segment         >>
         <<       2: multipage write buffer 0                    >>
         <<       3: multipage write buffer 1                    >>

         <<  4:2  Write buffer 0 status                          >>
         <<  6:2  Write buffer 1 status                          >>
         <<       Values:                                        >>
         <<                                                      >>
         <<       0: buffer empty (available)                    >>
         <<       1: data buffered - waiting for next req.       >>
         <<       2: data shipped  - waiting for completion      >>
```

```
  DEFINE WR'BUF0'STAT =
         PAD'DIT'ARRAY(BUFFER'STATUS'MULTIPAGE).(4:2)#;

  DEFINE WR'BUF1'STAT =
         PAD'DIT'ARRAY(BUFFER'STATUS'MULTIPAGE).(6:2)#;

  DEFINE NEXT'OUTGOING'WRITE'BUFFER =
         PAD'DIT'ARRAY(BUFFER'STATUS'MULTIPAGE).(9:1)#;

EQUATE WR'BUFF0'XDS = 51;
         <<This word holds the number of bytes of outgoing write >>
         <<data stored in buffer 0.                              >>

  DEFINE WRITE'BUFF0'XDS = PAD'DIT'ARRAY(WR'BUFF0'XDS)#;

EQUATE WR'BUFF1'XDS = 52;
         <<This word holds the number of bytes of outgoing write >>
         <<data stored in buffer 1.                              >>

  DEFINE WRITE'BUFF1'XDS = PAD'DIT'ARRAY(WR'BUFF1'XDS)#;

EQUATE WR'BUFF0'BYTES = 53;
         <<This word holds the number of bytes of outgoing write >>
         <<data stored in write buffer 0.                        >>

  DEFINE WRITE'BUFF0'BYTES = PAD'DIT'ARRAY(WR'BUFF0'BYTES)#;

EQUATE WR'BUFF1'BYTES = 54;
         <<This word holds the number of bytes of outgoing write >>
         <<data stored in write buffer 1.                        >>

  DEFINE WRITE'BUFF1'BYTES = PAD'DIT'ARRAY(WR'BUFF1'BYTES)#;

EQUATE PREV'QFUNC = 55;
          <<This word holds the previous request function code in>>
          <<its left 8 bits.  This is used for conversational    >>
          <<write/read requests.  Usually it is a previous read  >>
          << function = 1.                                       >>

  DEFINE PREVIOUS'FUNCTION'CODE = PAD'DIT'ARRAY(PREV'QFUNC).(8:8)#;


EQUATE PREV'QPAR1 = 56;
          <<This word holds the IOQ QPAR1 parameter value from   >>
          <<the previous request, usually a write.  This is also >>
          <<used to make conversational GETQ requests.  It       >>
          <<usually holds formatting information for the previous>>
          <<write request that has already had its IOQ returned. >>

  DEFINE PREVIOUS'QPAR1 = PAD'DIT'ARRAY(PREV'QPAR1)#;

EQUATE PREV'QPAR2 = 57;
          <<Like the word above, it holds the value of IOQ QPAR2 >>
          <<parameter for the same purpose.                      >>
```

```
  DEFINE PREVIOUS'QPAR2 = PAD'DIT'ARRAY(PREV'QPAR2)#;

EQUATE DIT'TIMER = 58;

  DEFINE TIMER'REQUEST'LIST'NUMBER =
         PAD'DIT'ARRAY(DIT'TIMER)#;

           <<This word holds the value of the timer request list >>
           <<entry returned from the procedure TIMERREQ. It is   >>
           <<stored here so that the timer request can be aborted>>
           <<using this value as a reference.                    >>

EQUATE DSTOP = 59;
  DEFINE SS'BREAK'CHARACTER = PAD'DIT'ARRAY(DSTOP).(0:8)#;
           <<This byte is the character used as a subsystem break>>
           <<character by the completor.  Default is control Y   >>
           <<which is ASCII EM.  This is set by an IOQ with a    >>
           <<function code 37.                                   >>

  DEFINE EOR'CHARACTER      = PAD'DIT'ARRAY(DSTOP).(8:8)#;
           <<This byte is the character used as the end-of-record>>
           <<character for use by the completor.  It is set by   >>
           <<an IOQ with function code 37.                       >>
           <<Default is Carriage Return in character mode and    >>
           <<Record Separator in block mode.                     >>

EQUATE CARRIAGE'RETURN  = 13;
EQUATE RECORD'SEPARATOR = 30;
EQUATE CONTROL'Y        = 25;

EQUATE STORE'READ'DST   = 60;

  DEFINE BUFFERED'READ'DST = PAD'DIT'ARRAY(STORE'READ'DST)#;
           <<This contains the DST # of a temporary extra data   >>
           <<segment used to temporarily buffer incoming data    >>
           <<that was not expected, e.g., logon hello requests,  >>
           <<data held when EOF found.                           >>


EQUATE STORE'READ'OFF   = 61;
  DEFINE BUFFERED'READ'OFFSET = PAD'DIT'ARRAY(STORE'READ'OFF)#;
            <<This holds the offset in the extra data segment for>>
            <<buffered read data.                                >>


EQUATE STORE'READ'B'C   = 62;
  DEFINE BUFFERED'READ'BYTE'COUNT = PAD'DIT'ARRAY(STORE'READ'B'C)#;
            <<This holds the positive byte count of the number   >>
            <<characters stored in the temporary read buffer.    >>
```

PAD TERMINAL EXTRA DATA SEGMENT

The PAD terminal extra data segment is a buffer for storing both write and read data as well as carriage control characters. This extra data segment is acquired at logon time and is released when the terminal logs off. It is never frozen in memory.

Its areas include:

(1) Five words giving the segment relative offsets to tables and buffers. The offsets are byte offsets. Note that at the present time, the parity conversion tables are not used. These offsets are used only for parity conversion, but not implemented now.

(2) A table of carriage control characters to be sent with the user write data.

(3) Two areas for parity conversion tables, unused at the present time.

(4) The initial PAD parameters set for the terminal user when he logs on to the host.

(5) The values of the PAD parameters that are to be set by this driver, sent as a SET or SET AND READ PAD command.

(6) The area in which to read in the values of the PAD parameters following a SET AND READ level-1 data message to the PAD.

(7) Two write buffers, each of length defined below.

(8) The buffer area for level-0 reads, of length defined below. This buffer area is expandable when larger reads are needed. The maximum size allowable for the entire extra data segment (XDS) is 31232 words.

PAD TERMINAL DIT EXTENSION IN EXTRA DATA SEGMENT

```
  0  1  2  3  4  5  6  7 10 11 12 13 14 15 16 17%
|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|

+-----------------------------------------------+
| Byte offset of even incoming parity conv. tab.|  0
|-----------------------------------------------|
| Byte offset of even outgoing partiy conv. tab.|  1
|-----------------------------------------------|
| Byte offset of write buffer 0                 |  2
|-----------------------------------------------|
| Byte offset of write buffer 1                 |  3
|-----------------------------------------------|
| Byte offset of read buffer                    |  4
|-----------------------------------------------|
|   Formfeed = %14    |   Null                  |  5
|-----------------------------------------------|
|   Formfeed          |   Carriage Return = %15 |  6
|-----------------------------------------------|
|   Carriage Return   |   Line feed =  %12      |  7
|-----------------------------------------------|
|   Line feed         |   Line feed             |  8
|-----------------------------------------------|
                      .
                      .
                      .
                      V
        (total of 31 words of 2 line feeds)
|-----------------------------------------------|
|   Line feed         |   Line feed             |  38
|-----------------------------------------------|
| Exclamation Point! | Exclamation Point = !    |  39
|-----------------------------------------------|
| Exclamation Point! |   Carriage Return = %15  |
|-----------------------------------------------|
|   Line feed =  %12  |                         |
|-----------------------------------------------|
| Table to convert incoming even parity to 0    |  42
                      |
  (Unused now)        |   (128 words)
                      V
|-----------------------------------------------|
| Table to convert outgoing data to 0 parity    |  170
                      |
  (Unused now)        |   (064 words)
                      V
|                                               |
|-----------------------------------------------|
| 0  0  0  0|   MC      | Parm # = 1            |  234
|-----------------------------------------------|
| Parm 1 initial value  | Parm # = 2            |
|-----------------------------------------------|
| Parm 2 initial value  | Parm # = 3            |
|===============================================|
```

```
                         .
                         .
                         .
|-----------------------|-----------------------|
| Parm 12 initial value |    Null               |
|-----------------------------------------------|
| 0  0  0  0|    MC     | Parmeter# a           |  246
|-----------------------------------------------|
| SET value for parm. a | Parmeter # b          |
|-----------------------------------------------|
| SET value for parm. b | Parmeter # c          |
|-----------------------------------------------|
                         .
                         .
                         .
|-----------------------------------------------|
| SET value for parm. n |    Null               |
|-----------------------------------------------|
| 0  0  0  0|    MC     | Parmeter # a          |  258
|-----------------------------------------------|
| READ area for parm. a | Parmeter # b          |
|-----------------------------------------------|
| READ area for parm. b | Parmeter # c          |
|-----------------------------------------------|
                         .
                         .
                         .
|-----------------------------------------------|
| READ area for parm. n | Null                  |
|-----------------------------------------------|
|                                               |  272
|          LEVEL-0 WRITE BUFFER  0              |
                     |    Length can be
                     |    changed by a new
                     V    equate of
                          LENGTH'READ'WRITE'BUFFERS
|-----------------------------------------------|
|          LEVEL-0 WRITE BUFFER 1               |  1232
|                    |                          |
                     |    Length =
                     V    LENGTH'READ'WRITE'BUFFERS
|                                               |
|-----------------------------------------------|
|          LEVEL-0 READ BUFFER                  |  2192
|                    |                          |
                     | Initial length =
                     V LENGTH'READ'WRITE'BUFFERS
|                                               |
|                                               |
|===============================================|  3152  (initially)
```

The MC is the message code. For a description of this and the PAD parameters and their values, see CCITT X.3 and X.29 specifications.

```
                                              <<End of Comment>>;

                              IODS0 IOQ

          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
         |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
     #0  |AB|SP|B |BC|IO|BL|CO|CF|CI|     |PF|HA|SO|ST|DO|  .QFLAG
         |-----------------------------------------------|
      1  |                   NEXT IOQP                   |  .QLINK
         |-----------------------------------------------|
      2  |MULTI-PACK HEAD+APP LN |          LDEV         |  .QLDEV
         |-----------------------------------------------|
      3  |          HEADER ADDRESS  (DSDS REL)           |  .QMISC
         |-----------------------------------------------|
      4  |            REQUESTING DST NUMBER              |  .QDSTN
         |-----------------------------------------------|
      5  |           XMIT ADDRESS  (DSDS REL)            |  .QADDR
         |-----------------------------------------------|
      6  |           DSDST            |    FUNC CODE     |  .QFUNC
         |-----------------------------------------------|
      7  |          XMIT COUNT  (+WORDS/-BYTES)          |  .QWBCT
         |-----------------------------------------------|
      8  |     RECEIVE ADDRESS  (REQUESTING DS REL)      |  .QPAR1
         |-----------------------------------------------|
      9  |                 RECEIVE COUNT                 |  .QPAR2
         |-----------------------------------------------|
     10  |        PCBN          |       ERROR CODE       |  .QSTAT
         |===============================================|
```

IODS0 IOQ NOTES

.QFLAG

AB - Abort request and return error to caller.
SP - :DSCONTROL operator request from CONSDSLINE'.
B - Broken.
BC - Sense of request is in bytes.
IO - Wake caller on completion of request.
BL - Blocked I/O. Wait in attachio until completion.
CO - Request is completed and caller waken if requested.
CF - Continuation record flag (are processing cont recds)
CI - Continuator record initiator IOQ.
PF - IOQ aborted because of power failure.
HA - Hard Pre-empt. This is a DSMON request.
SO - Soft Pre-empt. This is a non-DSMON request.
ST - Request started.
DO - DSMON request is complete. (Two part requests only)

.QLINK - SYSDB relative pointer to first word of next IOQ.
.QLDEV - Logical Device number.
.QMISC - If DSMON request, then DSLCB address. (DSMON DST relative) Else, the offset in DS data segment of the header address.
.QDSTN - Contains the DST number of the target data area.
.QADDR - Transmit address.
.QFUNC
DSDST - DS DST number in AFT(1) for non-DSMON requests. Set to zero for DSMON requests.
FUNC - 0 = Reserved for DSWRITECONV.
1 = DSWRITE.
2 = DSOPEN.
3 = DSCONTROL.
4 = DSCLOSE.
5 = DSWRITECONV.
.QWBCT - Transmit count.
.QPAR1 - Receive address. (DSWRITECONV only)
.QPAR2 - Receive count. (DSWRITECONV only)
.QSTAT - Request completion status and PCB number which is associated with this request.
.QPCBN - PCB number associated with this request.
If zero, this IOQ element will be returned by the system when the request is completed.

DS DEVICE (IODSO) ATTACHIO CALLS

	QMISC	DSTX	ADDR	FUNC	COUNT	P1	P2	FLAGS
BREAK (DSBREAK)	0	0	0	0	0	Break Type	Pin	%203
REJECT (DSREJECT)	0	0	0	0	0	Reject Stuff	From/To Process	%203
WRITE (DSWRITE)	Header	Data DST	Output Buffer	1	Output Length	0	0	DSparm
OPEN (DSOPEN)	0	Stack DST	Info Buffer	2	Info Length	DSoptions	0	%201
CLOSE (DSCLOSE)	0	Stack DST	Dummy Buffer	3	0	Close Type	0	%201
CONSREQ (CONDSLINE')	0	Stack DST	as below	4	---------	as below	------	%241
TRACEON			TraceFile Name		TraceFile Name Length	0	Trace Options	
TRACEOFF			Dummy Buffer		0	1	0	
OPEN/SHUT			LineSpeed (double)		2	3	Open Options	
MON			Dummy Buffer		0	7	Monitor Options	
DEBUG			Dummy Buffer		0	8	Debug Options	
RETRIES			RetryCount (integer)		1	9	0	
WRITECONV	Header	Data DST	Output Buffer	5	Output Length	Input Buffer	Input Length	DSparm
WRITECONV (DSMON)	LCB	DSMON Stack	BUF2 (output)	5	Output Length	BUF1 (input)	Input Length	

DS DEVICE (IODSO) ATTACHIO CALLS

RETURN := ATTACHIO(LDEV, QMISC, DSTX, ADDR, FUNC, COUNT, P1, P2, FLAGS)

LDEV - logical device number of DS device

QMISC - miscellaneous request-dependent parameter
HEADER - address of DSCB header area in device-process DS XDS
LCB - address of Line Control Block in DSMON's stack
RTE TIMESTAMP - ???

DSTX - DST number for data segment containing ADDR buffer
Data DST - DST number for stack or extra data segment with data
Stack DST - DST number for stack (of course)

ADDR - address of data or other request information (in DSTX)
Output Buffer - address of buffer holding outgoing data
Info Buffer - address of buffer holding open info (see OPEN MON REQ)
BUF2 - address of DSMON's BUF2, holding incoming messages
Trace file name - character string name of trace file (optional)
Line Speed - double value for CS line speed
Retry Count - integer value for CS error retry count
Dummy Buffer - address not used by request

FUNC - function code identifying request

COUNT - length of data in ADDR buffer
Output Length - length of data in Output Buffer (+words/-bytes)
Info Length - length of data in Info Buffer (+words)
Trace File Length - length of Trace File Name (+words)

P1 - request-dependent parameter
Break Type - ???
Reject Stuff - ???
DS Options - master/slave enabled (see OPEN MON REQ)
Close Type - if bit 14 = 1 then final close else not final close
Input Buffer - address of buffer to holding incoming data
BUF1 - address of DSMON's BUF1, to hold outgoing messages
Request Type - code selecting type of CONSREQ

P2 - request-dependent parameter
Pin - ???
From/to process - ???
Input Length - length of Input Buffer (+words/-bytes??)
Trace Options - see TRACE DSMON REQUEST
Open Options - see OPEN/SHUT DSMON REQUEST
Monitor Options - see MON DSMON REQUEST
Debug Options - see DEBUG DSMON REQUEST

FLAGS - flags specifying wait/no wait IO, preemption, etc.
%200 - Soft preempt, no wait (slave PTOP DSWRITECONV)
%201 - Soft preempt, wait (other user DSWRITECONVs)
%203 - Soft preempt, nowait, no PCB (DSREJECT and DSBREAK)
%241 - Soft preempt, wait, special (CONSREQ)

%400 - Hard preempt, no wait (DSMON DSWRITECONV)
%401 - Hard preempt, wait (DSMON DSWRITECONV)
DSparm - %200, %201, %400, or %401

DS DEVICE (IODS0) ATTACHIO RETURN VALUES

```
                           Word 1                    Word 2
                      (0:8)       (8:8)
                   +------------+-----------+-----------------------+
REJECT             |////////////|///////////|///////////////////////|
                   +------------+-----------+-----------------------+
BREAK              |  main pin  |///////////|///////////////////////|
                   +------------+-----------+-----------------------+
DSWRITE - nowait   |        IOQ index       |///////////////////////|
                   +------------+-----------+-----------------------+
DSWRITE - wait     |////////////| comp code |///////////////////////|
                   +------------+-----------+-----------------------+
OPEN               |////////////| comp code |  actual buffer size   |
                   +------------+-----------+-----------------------+
CLOSE              |////////////| comp code |///////////////////////|
                   +------------+-----------+-----------------------+
CONSOLE REQ - all  |////////////| comp code |///////////////////////|
                   +------------+-----------+-----------------------+
DSWRITECONV - nowait|       IOQ index       |///////////////////////|
                   +------------+-----------+-----------------------+
DSWRITECONV - wait |////////////| comp code | received data length  |
                   +------------+-----------+-----------------------+
```

Word 1:

main pin - apparently not used in DSBREAK??

comp code - completion code:

0 = IO request not yet completed

1 = IO request successfully completed

> 1 = IO request failed; CS error code

IOQ index - pointer to IOQ for nowait IO request

Word 2:

actual buffer size - actual size of DSMON buffer

received data length - actual length of data moved into INBUF

IODSX IOQ

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
   |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
# 0|AB|SP|  |  |IO|BL|CO|  |RE|  |  |  |HA|SO|ST|DO|% 0 QFLAG
   |-----------------------------------------------|
  1|                   NEXT IOQP                   |  1 QLINK
   |-----------------------------------------------|
  2|         0             |        LDEV           |  2 QLDEV
   |-----------------------------------------------|
  3|                                               |  3 QMISC
   |-----------------------------------------------|
  4|                                               |  4 QDSTN
   |-----------------------------------------------|
  5|                                               |  5 QADDR
   |-----------------------------------------------|
  6|           DSDST             | FUNC CODE       |  6 QFUNC
   |-----------------------------------------------|
  7|                                               |  7 QWBCT
   |-----------------------------------------------|
  8|                                               | 10 QPAR1
   |-----------------------------------------------|
  9|                                               | 11 QPAR2
   |-----------------------------------------------|
 10|       PCBN            |   ERROR CODE          | 12 QSTAT
   |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
```

IODSX IOQ NOTES

.QFLAG

AB - ABORT REQUEST AND RETURN ERROR TO CALLER
SP - :DSCONTROL OPERATOR REQUEST FROM CONDSLINE’
IO - WAKE CALLER ON COMPLETION OF REQUEST
BL - BLOCKED I/O. WAIT IN ATTACHIO UNTIL COMPLETION
CO - REQUEST IS COMPLETED AND CALLER WAKEN IF REQUESTED
RE - RESET REQUEST ISSUED TO DSMONX.
HA - HARD PRE-EMPT. THIS IS A DSMONX REQUEST.
SO - SOFT PRE-EMPT. THIS IS A NON DSMONX REQUEST.
ST - REQUEST STARTED.
DO - MESSAGE DONE.

.QLINK

SYSDB RELATIVE POINTER TO FIRST WORD OF NEXT IOQ.

.QLDEV

LOGICAL DEVICE NUMBER OF DSDEVICE.

.QMISC

MISCELLANEOUS REQUEST-DEPENDENT PARAMETER:

.QDSTN

DST NUMBER FOR DATA SEQMENT CONTAINING ADDR BUFFER

.QADDR

ADDRESS OF DATA OR OTHER REQUEST INFORMATION IN DSTN

.QFUNC

DSDST - DS DST NUMBER IN AFT(1) FOR NON-DSMONX REQUESTS.
FUNC - FUNCTION CODE IDENTIFYING REQUEST

FUNC 0	DSBREAK DSREJECT
FUNC 1	DSWRITE
FUNC 2	DSOPEN
FUNC 3	DSCLOSE
FUNC 4	DSCONSREQ
FUNC 5	DSWRITECONV
FUNC 6	ABORT READ
FUNC 7	INCOMING REPLY
FUNC 8	TIMER
FUNC 9	ABORT TIMER
FUNC 10	RESET
FUNC 11	FLOW CONTROL
FUNC 12	PAD CLEAR REQUEST

.QWBCT

LENGTH OF DATA IN ADDR BUFFER

.QPAR1
.QPAR2

REQUEST DEPENDENT PARAMETERS

.QSTAT

REQUEST COMPLETION STATUS AND PCB NUMBER WHICH IS ASSOCIATED WITH THIS REQUEST.

```
PCBN - PCBN ASSOCIATED WITH THIS REQUEST. IF ZERO,
       THIS IOQ ELEMENT WILL BE RETURNED BY THE
       SYSTEM WHEN THE REQUEST IS COMPLETED.
```

DS DEVICE (IODSX) ATTACHIO CALLS

	QMISC	DSTN	ADDR	FUNC	COUNT	P1	P2	FLAGS
BREAK (DSBREAK)	0	0	0	0	0	Break Type	Pin	%203
REJECT (DSREJECT)	RTE TIME	0	0	0	UC number	Reject Stuff	From/To process	%203
WRITE (DSWRITE)	Header	Data DST	Output Buffer	1	Output length	0	0	DSparm
OPEN (DSOPEN)	0	Stack DST	Info Buffer	2	Info Length	DSoptions	0	%201
CLOSE (DSCLOSE)	UC NUM.	Stack DST	Dummy Buffer	3	0	Close Type	0	%201
CONSREQ (CONDSLINE')	0	Stack DST	as below	4	---- as below --------			%241
TRACEON			Trace file Name		Trace file Name Length	0	Trace Option	
TRACEOFF			Dummy Buffer		0	1	0	
OPEN/SHUT			LineSpeed (double)		2	3	Open Options	
DEBUG			Dummy Buffer		0	8	Debug Options	
WRITE CONV	Header	Data DST	Output Buffer	5	Output Length	Input Buffer	Input Length	DSparm
ABORT READ	Termianl DIT PTR	0	0	6	0	UC Number	MainPin	%203
INCOMING REPLY	Request Code	0	0	7	0	UC Number	0	%203
TIMER	Time Requested	0	0	8	0	0	0	

ABORT TIMER	0	0	0	9	0	0	0	
RESET	0	0	0	10	0	UC Number	MainPin	%203
FLOW CONTROL	0	0	0	11	0	UC Number	From/To Process	%203
PAD CLEAR	0	0	0	12	0	UC Number	0	%203

DS DEVICE (IODSX) ATTACHIO CALLS (NOTES)

RETURN := ATTACHIO(LDEV,QMISC,DSTN,ADDR,FUNC,COUNT,P1,P2,FLAGS)

LDEV - logical device number of DS device.

QMISC -
  RTE Time - RTE time stamp used by 1000 systems.
  HEADER - Address of DSCB header area in device-process DS XDS. The DST number for this XDS is given in DSDST field of QFUNC.
  UC Num - UC number associated with the close request.
  Terminal DITP - Terminal DIT pointer.
  Request Code - Request code associated with incoming request. (Clear or Reset)
  Time Requested- Time requested by DSMONX for Timer request.

DSTN -
  Data DST - DST number for stack or extra data data seqment.
  Stack DST - DST number for stack.

ADDR - Address of data or other request information in DSTN.
  Output Buffer - Address of buffer holding outging data.
  Info Buffer - Address of buffer holding call information to be used by DSMONX or passed to high levels.
  Dummy Buffer - Address not used by request.
  Line Speed - Double value for CS line speed.
  TraceFile name- Character string name of trace file (optional).

FUNC - Function code identifying request.

COUNT - Length of data in ADDR buffer.
  Output length - Length of data in Output buffer (+words/-bytes).
  Info length - Length of data in Info buffer (+words/-bytes).
  Trace File Len.- Length of trace file name (+words).

P1 - Request-dependent parameter.
  Break Type - Break = -1
    Control Y = 0
    Resume = 1
    Abort = 2
  Reject Stuff - This word contains the following:
    (0: 4) := R.E.C.B
    (4: 6) := Message Class
    (10: 6) := Stream Type
  DSoptions - Master/Slave enabled. If bit 10=1 then slave first DSOPEN.
  Close Type - If bit 14=1 the final close else not final close.
  Input Buffer - Address of buffer holding incoming data.
  Console code - Code selecting type of CONSREQ.

- P2 - Request-dependent parameter
  - Pin - Current Pin.
  - From/To process- From and To process number.
  - Input Length - Length of input buffer (+words/-bytes).
  - Trace Options - see Trace DSMON REQUEST.
  - Open Options - see OPEN/SHUT DSMON REQUEST.
  - Debug Options - see DEBUG DSMON REQUEST.
  - MainPin - Mainpin associated with this request.

- FLAGS - Flags specifying wait/no wait IO, preemption, etc.
  - %200 -Soft preempt, no wait (slave PTOP DSWRITECONV)
  - %201 -Soft preempt, wait (other user DSWRITECONV)
  - %203 -Soft preempt,no wait,no PCB
  - %241 -Soft preempt,wait,special (CONSREQ)
  - DSPARM - %200, %201

DS DEVICE (IODSX) ATTACHIO RETURN VALUES

	WORD 1		WORD 2
	(0:8)	(8:8)	
REJECT	/////////	1	//////////////////
BREAK	/////////	1	//////////////////
DSWRITE - no wait	IOQ INDEX		//////////////////
DSWRITE - wait	/////////	COMPCODE	//////////////////
OPEN	/////////	COMPCODE	UC NUMBER
CLOSE	/////////	COMPCODE	//////////////////
CONSOLE REQ - ALL	/////////	COMPCODE	//////////////////
DSWITECONV - wait	/////////	COMPCODE	received data ln
ABORT READ	/////////	1	//////////////////
INCOMING REPLY	/////////	1	//////////////////
TIMER REQUEST	/////////	1	//////////////////
ABORT TIMER REQ	/////////	1	//////////////////
RESET	/////////	1	//////////////////
FLOW CONTROL	/////////	1	//////////////////
PAD CLEAR	/////////	1	//////////////////

WORD 1:

COMPCODE - Completion code:
 0 = IO request not yet completed.
 1 = IO request successfully completed.
 >1 = IO request failed, CS error code

IOQ INDEX - Pointer to IOQ for nowait IO request

WORD 2:

Received data ln - Acctual length of data moved into INBUF

```
                     IODSTRM0 IOQ

      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
    |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
#0  |AB|CI|XR|BC|IO|BL|CO|CR|CF|MO|     |HA|SO|ST|DO|  .QFLAG
    |-----------------------------------------------|
1   |                   NEXT IOQP                   |  .QLINK
    |-----------------------------------------------|
2   |      UNIT NUMBER      |         LDEV          |  .QLDEV
    |-----------------------------------------------|
3   |    RESERVED FOR FUTURE USE  (SET TO ZERO)     |
    |-----------------------------------------------|
4   |D |             DATA SEGMENT NUMBER            |  .QDSTN
    |-----------------------------------------------|
5   |             TARGET ADDRESS OFFSET             |  .QADDR
    |-----------------------------------------------|
6   |         UNUSED        |     FUNCTION CODE     |  .QFUNC
    |-----------------------------------------------|
7   |             COUNT  (+WORDS/-BYTES)            |  .QWBCT
    |-----------------------------------------------|
8   |       PARAMETER 1  (FUNCTION DEPENDENT)       |  .QPAR1
    |-----------------------------------------------|
9   |       PARAMETER 2  (FUNCTION DEPENDENT)       |  .QPAR2
    |-----------------------------------------------|
10  |         PCBN          | QUAL STAT | GEN STAT  |  .QSTAT
    |===============================================|
```

IODSTRMO IOQ NOTES

.QFLAG

AB - Abort request and return error to caller.
CI - Currently in CI prompt/read sequence.
XR - Print buffer expansion requested.
BC - Request is in bytes.
IO - Wake caller on completion of request.
BL - Blocked I/O. Wait in attachio until completion.
CO - Request is completed and caller waken if requested.
CF - Continuation record flag (are processing cont recds).
CI - Continuation record initiator IOQ.
MO - IOQ modified by driver
HA - Hard Pre-empt. This is a DSMON request.
SO - Soft Pre-empt. This is a non-DSMON request.
ST - Request started.
DO - DSMON request is complete. (Two part requests only)

.QLINK - SYSDB relative pointer to first word of next IOQ.
.QLDEV - Logical Device number.
.UNIT - Logical unit number.
.QDSTN - Contains the DST number of the target data area.
D - 1 = DB relative offset.
0 = Segment relative offset.
.QADDR - Offset to the target area data segment.
.QFUNC - Function code field.
.QWBCT - Word count or byte count and control returns.
.QPAR1 - Parameter one. (Function dependent use)
.QPAR2 - Parameter two. (Function dependent use)
.QSTAT - Request completion status and PCB number which is associated with this request.
.QPCBN - PCB number associated with this request.
If zero, this IOQ element will be returned by the system when the request is completed.
.QUAL - A code that further defines the general status.
.GEN - General status. Indicates the current status according to the following codes:
0 = Not started or awaiting completion.
1 = Successfully completed.
2 = End of file detected.
3 = Unusual conditon encountered.
4 = Irrecoverable error encountered.

```
                  IODSTRMX IOQ

     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
   |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
# 0|AB|CI|  |  |IO|BL|CO|  |RE|MO|  |  |HA|SO|ST|DO|%  0 QFLAG
   |-----------------------------------------------|
  1|                   NEXT IOQP                   |  1 QLINK
   |-----------------------------------------------|
  2|    UNIT NUMBER        |          LDEV         |  2 QLDEV
   |-----------------------------------------------|
  3|   RESERVED FOR FUTURE USE ( SET TO ZERO )     |  3 QMISC
   |-----------------------------------------------|
  4|           DATA SEGMENT NUMBER                 |  4 QDSTN
   |-----------------------------------------------|
  5|         TARGET ADDRESS OFFSET                 |  5 QADDR
   |-----------------------------------------------|
  6|         UNUSED        |       FUNC CODE       |  6 QFUNC
   |-----------------------------------------------|
  7|           COUNT (+WORDS/-BYTES)               |  7 QWBCT
   |-----------------------------------------------|
  8|     PARAMETER 1 (FUNCTION DEPENDENT)          | 10 QPAR1
   |-----------------------------------------------|
  9|     PARAMETER 2 (FUNCTION DEPENDENT)          | 11 QPAR2
   |-----------------------------------------------|
 10|       PCBN            | QUAL STAT | GEN STAT  | 12 QSTAT
   |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
```

IODSTRMX IOQ NOTES

.QFLAG

AB - ABORT REQUEST AND RETURN ERROR TO CALLER.
CI - CURRENTLY IN CI PROMPT/READ SEQUENCE.
IO - WAKE CALLER ON COMPLETION OF REQUEST.
BL - BLOCKED I/O. WAIT IN ATTACHIO UNTIL COMPLETION.
CO - REQUEST IS COMPLETED AND CALLER WAKEN IF REQUESTED.
RE - RESET REQUEST ISSUED TO DSMONX.
MO - IOQ MODIFIED BY DRIVER.
HA - HARD PRE-EMPT.
SO - SOFT PRE-EMTP.
ST - REQUEST STARTED.
DO - MESSAGE DONE.

.QLINK - SYSDB RELATIVE POINTER TO FIRST WORD OF NEXT IOQ.
.QLDEV - LOGICAL DEVICE NUMBER.
.UNIT - LOGICAL UNIT NUMBER.
.QDSTN - CONTAINS THE DST NUMBER OF THE TARGET DATA AREA.
.QADDR - OFFSET TO THE TARGET AREA IN DATA SEGMENT.
.QFUNC - FUNCTION CODE FIELD.

0. READ
1. WRITE
2. FILE OPEN
3. FILE CLOSE
4. DEVICE CLOSE
5. SET TIMEOUT
6. SET INSPEED
7. SET OUTSPEED
%10. ECHO ON
11. ECHO OFF
12. DISABLE BREAK
13. ENABLE BREAK
14. DISABLE ESCAPE
15. ENABLE ESCAPE
16. DISABLE TAPEMODE
17. ENABLE TAPEMODE
%20. DISABLE TIMER
%21. ENABLE TIMER
22. READ TIMER
23. DIABLE PARITY
24. ENABLE PARITY
25. LOGGED ON
26. SET PARITY
27. SET TERMINAL TYPE
30. ALLOCATE TERMINAL
31. CLEAR FLUSH AND WRITE
32. ENABLE CONTROL X !!! ECHO
33. DISABLE CONTROL X !!! ECHO
34. NOT USED
35. PTAPE READ
36. SET/RESET BREAK MODE
37. SET/RESET CONSOLE MODE

.QWBCT - WORD OR BYTE COUNT AND CONTROL RETURNS.
.PCBN - PCB NUMBER ASSOCIATED WITH THIS REQUST. IF ZERO THIS IOQ ELEMENT IS RETURNED BY THE SYSTEM WHEN THE REQUEST IS COMPLETED.
.QUAL - A CODE WHICH FURTHER DEFINES OR QUALIFIES THE GENERAL STATUS.
.STATUS - GENERAL STATUS. INDICATE THE CURRENT OR RESULTANT STATUS OF THE REQUEST ACCORDING TO THE FOLLOWING CODES:
0 - NOT STARTED OR AWAITING COMPLETION.
1 - SUCCESSFULLY COMPLETED.
2 - END OF FILE DETECTED.
3 - UNUSUAL CONDITION.
4 - IRRECOVERABLE ERROR.

```
                  IOPADO IOQ

     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15

   |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
0  |AB|        |IO|BL|CO|              |HP|SP|ST|DN|  .QFLAG
   |-----------------------------------------------|
1  |           POINTER TO NEXT IOQ POINTER         |  .QLINK
   |-----------------------------------------------|
2  |  UNIT NUMBER          |       LDEV            |  .QLDEV
   |-----------------------------------------------|
3  |FL|                 |READSTOP| RSTATE          |
   |-----------------------------------------------|
4  |ST|      DATA SEGMENT NUMBER                   |  .QDSTN
   |-----------------------------------------------|
5  |        TARGET ADDRESS OFFSET                  |  .QADDR
   |-----------------------------------------------|
6  |     UNUSED            |  FUNCTION CODE        |  .QFUNC
   |-----------------------------------------------|
7  |      DATA COUNT (+WORDS/-BYTES)               |  .QWBCT
   |-----------------------------------------------|
8  |      PARAMETER 1 (FUNCTION DEPENDENT)         |  .QPAR1
   |-----------------------------------------------|
9  |       PARAMETER 2 (FUNCTION DEPENDENT)        |  .QPAR2
   |-----------------------------------------------|
10 |      PCBN             | QUAL STAT | GEN STAT  |  .QSTAT
   |===============================================|
```

IOPADO IOQ ELEMENT FIELD NAMES

```
EQUATE QFLAG            =  0; <<Flags field of IOQ element   >>

      <<   Flags and Request State Information              >>
      <<                                                    >>
      <<   Bit 0:1  ABORTED bit.  When set, abort this      >>
      <<            request and return an error condition   >>
      <<            to the caller.                          >>
      <<                                                    >>
      <<   Bits 1:3 Unused.                                 >>
      <<                                                    >>
      <<   Bit 4:1  Wake caller on completion of request.   >>
      <<                                                    >>
      <<   Bit 5:1  Blocked I/O.  The caller is waited in   >>
      <<            ATTACHIO until request is completed.    >>
      <<            Implies wake.                           >>
      <<                                                    >>
      <<   Bit 6:1  Completed.  The request has been        >>
      <<            completed and callwaking caller is requested>>
      <<                                                    >>
      <<   Bit 7:5  Unused.                                 >>
      <<                                                    >>
      <<   Bit 12:1 Hard pre-empt write request.            >>
      <<                                                    >>
      <<   Bit 13:1 Soft pre-empt write request.            >>
      <<                                                    >>
      <<   Bit 14:1 This request has been started.          >>
      <<                                                    >>
      <<   Bit 15:1 This request is done. Used for 2-step   >>
      <<            request operations.                     >>

  ABORTED         =  (0:1)
  WAKE            =  (4:1)
  BLOCKED         =  (5:1)
  COMPLETED       =  (6:1)
  PRE'EMPT'LEVEL = (12:2)
  SOFT'PRE'EMPT  = (12:1)
  HARD'PRE'EMPT  = (13:1)
  STARTED         = (14:1)
  MSG'DONE        = (15:1)
  STARTED'MSGDONE = (14:2)

 QLINK            = 1; <<Pointer to next IOQ element  >>
                       <<on PAD terminal DIT if <> 0  >>
 QLDEV            = 2; <<The left byte is the logical >>
                       <<unit number, the right byte  >>
                       <<byte is the logical device   >>
                       <<number.                      >>
 QMISC            = 3; <<Unused, =0                   >>
 QDSTN            = 4; <<The source DST# (if outgoing)>>
                       <<or target DST#  (if incoming)>>
                       <<of data to be transferred    >>
 QADDR            = 5; <<The source offset (if        >>
```

```
                            <<outgoing) or target offset   >>
                            <<(if incoming) of data to be  >>
                            <<transferred                  >>

                            <<The first bit, ST, is set to >>
                            <<1 if the address is DB       >>
                            <<relative; it is 0 if it is   >>
                            <<segment relative.            >>

QFUNC                 = 6;  <<The right byte of this word  >>
                            <<is the function code:        >>
                            <<                             >>
                            <<  0:  read                   >>
                            <<  1:  write                  >>
                            <<  2:  file open              >>
                            <<  3:  file close             >>
                            <<  4:  device close           >>
                            <<  5:  set read timeout       >>
                            <<  6:  set input speed        >>
                            <<  7:  set output speed       >>
                            <<  8:  enable echo            >>
                            <<  9:  disable echo           >>
                            << 10:  disable break          >>
                            << 11:  enable break           >>
                            << 12:  disable subsystem break>>
                            << 13:  enable subsystem break >>
                            << 14:  disable tape mode      >>
                            << 15:  enable tape mode       >>
                            << 16:  disable read timer     >>
                            << 17:  enable read timer      >>
                            << 18:  return timed read      >>
                            << 19:  disable parity check   >>
                            << 20:  enable parity check    >>
                            << 21:  set logon type         >>
                            << 22:  unused                 >>
                            << 23:  set terminal type      >>
                            << 24:  allocate terminal      >>
                            << 25:  clear flush & write    >>
                            << 26:  ctrl X echo on         >>
                            << 27:  ctrl X echo off        >>
                            << 28:  unused                 >>
                            << 29:  ptape read             >>
                            << 30:  set break mode         >>
                            << 31:  set console mode       >>
                            << 32:  set parity             >>
                            << 33:  allocate terminal      >>
                            << 34:  set terminal type      >>
                            << 35:  return terminal type   >>
                            << 36:  return terminal speed  >>
                            << 37:  set new stop and       >>
                            <<      subsystem break chars. >>
```

```
QWBCT            =  7; <<The + word or - byte count of>>
                       <<the data to be transferred   >>
QPAR1            =%10; <<The first parameter, function>>
                       <<dependent; see table below.  >>
QPAR2           = %11; <<The second parameter,function>>
                       <<dependent; see table below.  >>


QSTAT           = %12; <<The status of the IOQ request>>
                       <<the left byte is the process >>
                       <<control block number (if it  >>
                       <<is zero, this IOQ element is >>
                       <<returned by the system when  >>
                       <<I/O is complete, the right   >>
                       <<byte contains status         >>
                       <<status information           >>

        IOQ'PCBN = (0:7) ;  <<PCB number associated with this   >>
                            <<request.  If zero, this IOQ       >>
                            <<element is returned by the system>>
                            <<when the request is complete. It >>
                            <<will be zero if driver does an    >>
                            <<ATTACHIO itself.                  >>

        IOQ'QUALIFIER = (8:4) ;  <<Qualifying status, see       >>
                                 <<ATTACHIO for details.        >>

        IOQ'STATUS  = (12:4) ; <<General status of the request:>>
                               << 0:  Not started               >>
                               <<                               >>
                               << 1:  Sucessfully completed;    >>
                               << 2:  End-of-file detected;     >>
                               << 3:  Unusual condition;        >>
                               << 4:  Irrecoverable error;      >>
                               <<                                  >>
                               << 7: This request, presumably a    >>
                               <<    print is to be sent as soon   >>
                               <<    as there is a GETQ available;>>
                               <<    used for pre-emptive prints.  >>
```

IOPAD0 IOQ PARAMETERS 1 AND 2 AND IOQ FUNCTIONS

IOQ FUNCTION	MEANING OF QPAR1 AND QPAR2		
0: Read	P1.(0:1)	1:	Suppress line feed following read.
	P1.(13:3)		
		0:	reset EOF and read
		1:	detect :EOF:
		2:	detect all data EOF's
		3:	detect all session EOF's
		4:	detect all job EOF's
		5:	no EOF check; don't return saved EOF data on this read.
	P2.(0:8)		Special end-of-read character if not 0.
	P2.(9:1)		V/3000 read if set.
	P2.(10:1)		User block mode read if set.
	P2.(11:2)		Binary read.
1: Write	P1		Vertical format specification
		%01 :	Use first character of user data as vertical format specification.
		%53 :	Carriage return only.
		%55 :	Triple space and carriage return.
		%60 :	Double space.
		%61 :	Formfeed.
		%200-277 :	(n-%200) LF's then carriage return.
		%320 :	No vertical formatting.
		others :	carriage return and line feed.
	P2.(15:1)	:	If set, prespace vertical formatting characters, e.g. in Fortran.
2: File open	P1	:	Terminal type.
	P2	:	Terminal speed.
3: File close	P1,P2	:	Unused.
4: Device close	P1,P2	:	Unused.
5: Set read timeout	P1	:	Read time out in seconds; if 0 disable read timeout.
6: Set input speed	P1,P2	:	This function is done by PAD, not driver.
7: Set output speed	P1,P2	:	This function is done by PAD, not driver.
8: Enable echo	P1,P2	:	Unused. Old echo setting returned in returned byte count.

9: Disable echo	P1,P2	Unused. Old echo setting returned in returned byte count.
10-20:	P1,P2	The parameters in these function requests are not used.
21: Set logon type	P1	0 data, break not enabled 1 session, break enabled 2 job, break not enabled.
22: Unused.		
23: Set terminal type	P1	Terminal type.
24: Allocate terminal	P1	Terminal type.
	P2	Line speed. (unused).
25: Clear flush and write		Same parameters as write.
26-29:		No parameter values checked.
30: Set break mode	P1	Odd, terminal in break; Even, terminal not in break.
31: Set console mode, unused.		
32: Set parity, unused presently.		
33: Same as function 33.		
34: Same as function 23.		
35: Return terminal type		Terminal type returned in byte count.
36: Return output speed		Return speed in CPS in byte count.
37: Set new stop and subsystem break characters	P1	if 0, disable special character;
	.(0:8)	subsystem break character
	.(8:15)	stop character

DS I/O TABLE LINKAGE

```
                         +--------------------------------+
                         |            ^                   ^
                         v            |                   |
+=========+         +=========+       | +=========+       | +=========+
|         |<------+ |         | +------>|         | ------->|         |
|         |       | |---------| |     | |---------| |     | |---------|
|         |   (2) | |     ---------+  | |     ---------+  | |    0    |
|         |       | |---------| |  |  | |---------| |  |  | |---------|
|         |       | |         | |  |  | |         | |  |  | |         |
|         |       | |         | |  |  | |         | |  |  | |         |
|         |       | |---------| |  |  | |---------| |  |  | |---------|
|         |   (7) +----- | -----+  |  +----- | -----+  |  +----- |  0 |
|         |         |---------|    |    |---------|    |    |---------|
|         |         |         |    |    |         |    |    |         |
|         |         |         |    |    |         |    |    |         |
|         |         |         |    |    |         |    |    |         |
+=========+         +=========+    |    +=========+    |    +=========+
 CS Device           DS Device     |      Virtual      |      Virtual
    DIT                 DIT        |    Terminal DIT   |    Terminal DIT
                                   |                   |
                                   |                   |
                         +---------+         +---------+
                         |                   |
                         v                   v
                    +=========+         +=========+
[user request IOQ]  |         |         |         | [virtual terminal
                    |---------|         |---------|  request IOQ]
              (1)   |    ----------+    |    0    |
                    |---------|    |    |---------|
                    |         |    |    |         |
                    |         |    |    |         |
                    +=========+    |    +=========+
                                   |
                         +---------+
                         |
                         v
                    +=========+
[DSMON request IOQ] |         |
                    |---------|
                    |    0    |
                    |---------|
                    |         |
                    |         |
                    +=========+
```

Q-Relative values for drivers IODS0,IODSTRM0 and procedures DSKILLALL, DSLOGON, DSXIO, and DSREJECT.

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
|            XLDEV   (DSABORT ENTRY)            | Q-5
|-----------------------------------------------|
|                     FLAGS                     | Q-4
|-----------------------------------------------|
|                       X                       | Q-3
|-----------------------------------------------|
|                    DELTA P                    | Q-2
|-----------------------------------------------|
|                    STATUS                     | Q-1
|-----------------------------------------------|
|                    DELTA Q                    | Q
|-----------------------------------------------|
|                    HEADER                     | Q+1
|-----------------------------------------------|
|                DSINFO, DSDITP                 | Q+2
|-----------------------------------------------|
|                DSIOQP, DSIOQPD                | Q+3
|-----------------------------------------------|
|                 DSDST, DSLOC                  | Q+4
|-----------------------------------------------|
|                    DSADDR                     | Q+5
|-----------------------------------------------|
|              DSLCB  (DSDST REL)               | Q+6
|-----------------------------------------------|
|              DSBUFSZ, DSBUFSZD                | Q+7
|-----------------------------------------------|
|                    DSCOUNT                    | Q+8
|-----------------------------------------------|
|                     DITP                      | Q+9
|-----------------------------------------------|
|                     IOQP                      | Q+10
|-----------------------------------------------|
|                    TOPROC                     | Q+11
|-----------------------------------------------|
|                    MSGCLS                     | Q+12
|-----------------------------------------------|
|       STRMTYP, ERROCD  (IF HEADER = -1)       | Q+13
|-----------------------------------------------|
|                    MSGLEN                     | Q+14
|===============================================|
```

DSMON <---> DSIOM
PSEUDO HEADER/CONTROL BLOCK

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
    |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
 0  |                      - 1                      |
    |-----------------------------------------------|
 1  |          - 1 or DSMON REQ. NUMBER             |  .REQ
    |-----------------------------------------------|
 2  |                  ERROR CODE                   |  .ERRCOD
    |-----------------------------------------------|
 3  |                  BUFFER SIZE                  |  .BUFSIZE
    |-----------------------------------------------|
 4  |        UNUSED         | N| Q| L|STATE| X| M| S|  .FLAGS
    |-----------------------------------------------|
 5  |                       0                       |
    |-----------------------------------------------|
 6  |                       0                       |
    |-----------------------------------------------|
 7  |                       0                       |
    |===============================================|
```

NOTES

This pseudo-header may at times be seen in words 8-15 (%10-%17) of the IODS0 DIT.

.REQ - If DSMON request number is still the same when returned from DSIOM, then the request was processed. If a -1 is returned, and if ERRCOD = 0, then request was discarded due to a CSWRITE in progress and DSIOM has to resubmit the request. If a -1 returned and ERRCOD <> 0, then a CSERROR occurred on this request, and ERRCOD contains the CSERROR code.

.FLAGS - Same as in the IODS0 DIT. (DIT0.(9:7))
- N - DSMON has a null CWRITE outstanding
- Q - DSMON has a CREAD ENQ outstanding
- L - CS line is a secondary
- STATE - CS line state
  - 0 - unconnected
  - 1 - control
  - 2 - text
- X - exclusive mode enabled
- M - master mode enabled
- S - slave mode enabled

DSMON REQUEST FORMATS

DSMON requests are internal messages sent from a user to a DSMON process to request some service (opening or closing a CS line, turning on CS tracing, etc.) The request is sent and executed by the following mechanism:

1. The user calls ATTACHIO to initiate an IO request. (See DS DEVICE ATTACHIO PARAMETERS)
2. ATTACHIO calls DSIOM, which calls the IODS0 driver to execute the IO request.
3. IODS0 calls DSMONREQ to transmit the request to DSMON.
4. DSMONREQ formats a DSMON request message, and moves it to DSMON's incoming message buffer BUF1.
5. DSIOM completes a pending DSWRITECONV from DSMON.
6. DSMON awakes and notices the DSMON request in its BUF1. It executes the appropriate MONxxx procedure based on the request type, then calls DSWRITECONV to return the results of the request in a DSMON-DSIOM communications block.
7. DSWRITECONV calls ATTACHIO which calls DSIOM.
8. DSIOM processes the DSMON-DSIOM communications block and notices that a DSMON request has been completed. It calls IODS0 again to finish completion of the IO request.
9. IODS0 does associated bookkeeping (incrementing usecounts, etc) sets the ATTACHIO status returns to reflect the success or failure of the request, and calls DSCOMPLETE to complete the pending IO request.
10. The user's ATTACHIO call completes, giving the user the status of its request. (See DS DEVICE ATTACHIO RETURNS).

DSMON REQUEST FORMATS

The general format of a DSMON request is

```
+==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==+
|                     - 1                       |
|-----------------------------------------------|
|                 Request type                  |
|-----------------------------------------------|
|               Request parameter               |
|-----------------------------------------------|
|                                               |
                 Request buffer
|                                               |
+===============================================+
```

-1	- identifies the message as a DSMON request
Request type	- selects specific request
Request parameter	- one word of request-specific information
Request buffer	- more request-specific information (may be omitted)

DSMON REQUEST FORMATS

Request type 0 - MONTRCON - turns on CS tracing

```
                                   1  1  1  1  1  1
        0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
       +==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==+
    0  |                      - 1                      |
       |-----------------------------------------------|
    1  |                       0                       |
       |-----------------------------------------------|
    2  |AL|WR|           MASK            |   ENTRIES   | Trace Options
       |-----------------------------------------------|
    3  |                                               |
                        TRACE FILE NAME
  2+n  |                                               |
       +===============================================+
```

Request type = 0

```
Request parameter - Trace Options
      AL     (0:1) - 0 = trace I/O errors only
                     1 = trace all activity
      WR     (1:1) - 0 = no wrap on trace entries
                     1 = wrap trace entries if table is full
      MASK   (2:8) - 0 = use default trace mask
                    >0 = mask indicating driver actions to trace
                         bit  2 - STN
                         bit  3 - OPR and EDT
                         bit  4 - RCT
                         bit  5 - RTX
                         bit  6 - SCS, POL, SEL
                         bit  7 - STX
                         bit  8 - 3270 STN
                         bit  9 - not used
                         bit 10 - mainframe IC
      ENTRIES (11:5) - 0 = use driver default for max entries per record
                      >0 = (max number of entries per record)/8

Request buffer     - Trace file name, a string of 2n ascii characters
                     with one or two trailing blanks
```

Request type 1 - MONTRCOFF - turns off CS tracing

```
                                    1  1  1  1  1  1
      0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
    +==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==+
  0 |                     - 1                       |
    |-----------------------------------------------|
  1 |                       1                       |
    |-----------------------------------------------|
  2 |                       0                       |
    +===============================================+
```

Request type   - 1

Request parameter - unused

Request buffer   - none

DSMON REQUEST FORMATS

Request type 2 - MONOPEN - first open to DS device; causes DSMON
to COPEN CS device

```
                                      1  1  1  1  1  1
        0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
      +==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==+
    0 |                      - 1                      |
      |-----------------------------------------------|
    1 |                       2                       |
      |-----------------------------------------------|
    2 |  STDLIST LDEV (X.21)  |  |Q |SL|QU|NC|CO|CI|EX| DS options
      |-----------------------------------------------|
    3 |                    BUFSIZE                    | Open Info
      |-----------------------------------------------|
    4 |          IDLIST LENGTH (+BYTES)               |
      |-----------------------------------------------|
    5 | total number of IDs   |    local ID length    |
      |...............................................|
      |                                               |
                           local  ID
      |                                               |
      |...............................................|
      |   remote ID length    |                       |
      |.......................                        |
      |                                               |
                           remote ID 1
      |                                               |
      |...............................................|
                            .  .  .
      |...............................................|
      |   remote ID n length  |                       |
      |.......................                        |
      |                                               |
                           remote ID n
      |                                               |
      |-----------------------------------------------|
      |               PHONELIST Length                |
      |...............................................|
      |total phone numbers = 1|  phone number 1 length|
      |...............................................|
      |                                               |
                          phone number
      |                                               |
      +===============================================+
```

Request type - 2

Request parameter - DS (open) options
Q (9:1) - X.21 queued flag
SL (10:1) - first slave DSOPEN
QU (11:1) - QUIET
NC (12:1) - NOCOMP
CO (13:1) - COMP
CI (14:1) - open from DSLINE or REMOTE HELLO
EX (15:1) - EXCLUSIVE mode

Request buffer - COPEN related parameters
BUFSIZE - size of DS line buffer (from :DSLINE LINEBUF)
IDLIST - local and remote IDs (from :DSLINE LOCID and REMID)
IDLIST length = 0 - configured default id sequences
PHONELIST - remote phone number (from :DSLINE PHNUM)
PHONELIST length = 0 - configured default phone list

Request type 3 - MONCONSCMD - opens or shuts master and slave access

```
                                      1  1  1  1  1  1
        0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
      +==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==+
    0 |                     - 1                       |
      |-----------------------------------------------|
    1 |                       3                       |
      |-----------------------------------------------|
    2 | NETWORK ID (X.21)  |              |SL| X| M| S| Open options
      +===============================================+
    3 |                     LINE                      |
      |                    SPEED                      |
      +===============================================+
```

Request type - 3

Request parameter - Open options (from :DSCONTROL command)
SL (12:1) - Dev is X.21 switched line
X (13:1) - Dev is X.21 related
M (14:1) - MASTER mode enabled
S (15:1) - SLAVE mode enabled

Request buffer - Line speed (double word value, from :DSCONTROL)
(Line speed = 0 - use configured default)

DSMON REQUEST FORMAT

Request type 4 - MONCLOSE - last DSCLOSE to DS device; causes DSMON
to CCLOSE the CS line

```
                                    1  1  1  1  1  1
      0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
    +==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==+
  0 |                    - 1                        |
    |-----------------------------------------------|
  1 |                     4                         |
    |-----------------------------------------------|
  2 |                     0                         |
    +===============================================+
```

Request type - 4

Request parameter - not used

Request buffer - none

NOTE: Request types 5 (MONSYSOPEN) and 6 (MONSYSREAD) no longer exist. They appear to have been a planned feature of DS that fell through the cracks and was lost and forgotten. Remains of this prehistoric code can be found in DSMON.

DSMON REQUEST FORMATS

Request type 7 - MONMON - turns on (off) CS and DS MMSTAT monitoring

```
                                  1  1  1  1  1  1
    0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
   +==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==+
 0 |                     - 1                       |
   |-----------------------------------------------|
 1 |                      7                        |
   |-----------------------------------------------|
 2 |//////////////////////////////////////|CS|DS|ON|Monitor options
   +===============================================+
```

Request type - 7

Request parameter - Monitor options
CS (14:1) - 0 = doesn't effect CS monitoring
1 = turn on (off) CS monitoring
DS (15:1) - 0 = doesn't effect DS monitoring
1 = turn on (off) DS monitoring
ON (15:1) - 0 = turn off monitoring
1 = turn on monitoring

Request buffer - none

Request type 8 - MONDEBUG - activates or deactives DSMON breakpoints

```
                                    1  1  1  1  1  1
      0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
    +==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==+
  0 |                      - 1                      |
    |-----------------------------------------------|
  1 |                       8                       |
    |-----------------------------------------------|
  2 |                 DEBUG OPTION                  |
    +===============================================+
```

Request type - 8

Request parameter - Debug option
0 = deactivate DSMON breakpoint
1 = activate DSMON breakpoint
2 = activate fatal error traps:
if DS error, cause System Failure 915
if CS error, cause System Failure 916

Request buffer - none

Request type 9 - MONRETRIES - changes number of CWRITE error retries

```
                                   1  1  1  1  1  1
     0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
    +==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==+
  0 |                     - 1                       |
    |-----------------------------------------------|
  1 |                       9                       |
    |-----------------------------------------------|
  2 |                       0                       |
    |-----------------------------------------------|
  3 |               NUMBER OF RETRIES               |
    +===============================================+
```

Request type   - 9

Request parameter - not used

Request buffer   - number of retries (single word value)

DSMONX Communication Buffers Format
DSGETQ & DSPUTQ buffer format

```
      DSGETQ BUFFER                       DSPUTQ BUFFER

+-------------------------+         +--------------------------+
|      # OF ENTRIES       |         |       # OF ENTRIES       |
|-------------------------|         |--------------------------|
|                         |         |                          |
|       ENTRY 1           |         |        ENTRY 1           |
|                         |         |                          |
|-------------------------|         |--------------------------|
|                         |         |                          |
|       ENTRY 2           |         |        ENTRY 2           |
|                         |         |                          |
|-------------------------|         |--------------------------|
.                         .         .                          .
.                         .         .                          .
|-------------------------|         |--------------------------|
|                         |         |                          |
|       ENTRY N           |         |        ENTRY N           |
|                         |         |                          |
+-------------------------+         +--------------------------+
```

ENTRY FORMAT

```
+-------------------------+         +--------------------------+
|   USER CHANNEL NUMBER   |         |    USER CHANNEL NUMBER   |
|-------------------------|         |--------------------------|
|   REQUEST/REPLY CODE    |         |    IOQINDEX  /  ZERO     |
|-------------------------|         |--------------------------|
|  IOQINDEX   / ZERO      |         |     R-DATA DESCRIPTOR    |
|-------------------------|         |--------------------------|
|  T-DATA DESCRIPTOR 1    |         |    REQUEST/REPLY CODE    |
|-------------------------|         |--------------------------|
|  R-DATA DESCRIPTOR 1    |         |     REQUEST STATUS       |
|-------------------------|         |--------------------------|
|  T-DATA DESCRIPTOR 2    |         | CHAN. TYPE  | MAINPIN    |
|-------------------------|         +--------------------------+
|  R-DATA DESCRIPTOR 2    |
|-------------------------|
|  T-DATA DESCRIPTOR 3    |
|-------------------------|
|  R-DATA DESCRIPTOR 3    |
|-------------------------|
|       MAINPIN           |
+-------------------------+
```

```
dsgetq'fc=%20,

<< ATTACHIO function code for DSGETQ request                 >>

dsgetq'entlen=22,

<< each request entry size in DSGETQ buffer                  >>

dsgetq'buflen=dsgetq'entlen*max'num'of'entries+1;

<< the buffer size for DSGETQ request                        >>


dsgetq'num'of'entries=dsgetq'bufptr#,

<< the first word of DSGETQ indicates number of requests in>>
<< DSGETQ buffer                                             >>

dsgetq'ucno=dsgetq'entptr#,

<< user channel number of this request associated with       >>

dsgetq'req'code=dsgetq'entptr(1).(8:8)#,

<< if dsgetq'ioqindex <> 0, this word has the following      >>
<< definition:                                               >>
<<                                                           >>
<< 1 - DS call request                                       >>
<< 2 - reserved for PAD call (PAD emulator)                  >>
<< 3 - call clear request                                    >>
<< 4 - level-0 I/O request                                   >>
<< 5 - level-1 I/O request (PAD messages)                    >>
<< 6 - interrupt request                                     >>
<< 7 - restart request                                       >>
<< 8 - reset request                                         >>
<< 9 - info request                                          >>

dsgetq'status=dsgetq'entptr(1).(8:8)#,

<< if dsgetq'ioqindex = 0, this word has the following       >>
<< defintion:                                                >>
<<                                                           >>
<< 3 - incoming clear has been completed                     >>
<< 5 - incoming reset has been completed                     >>

dsgetq'subcode=dsgetq'entptr(1).(0:8)#,

<< if dsgetq'ioqindex <> 0 and requestor is a PAD driver,    >>
<< this byte has the following definition:                   >>
<<                                                           >>
<< 0 - read terminator is a carriage return                  >>
<< 1 - read terminator is a record seperator                 >>
<< 2 - read terminator is a non-printing character           >>
<<                                                           >>
<< otherwise, this byte should be zero                       >>
```

```
dsgetq'ioqindex=dsgetq'entptr(2)#,

<< the ioqindex associated with this request                  >>

dsgetq'odes1=dsgetq'entptr(3)#,
dsgetq'odst1=dsgetq'entptr(3)#,
dsgetq'oaddr1=dsgetq'entptr(4)#,
dsgetq'olen1=dsgetq'entptr(5)#,

<< the first piece of the outgoing message                    >>

dsgetq'ides1=dsgetq'entptr(6)#,
dsgetq'idst1=dsgetq'entptr(6)#,
dsgetq'iaddr1=dsgetq'entptr(7)#,
dsgetq'ilen1=dsgetq'entptr(8)#,

<< the first buffer for the incoming message                  >>

dsgetq'odes2=dsgetq'entptr(9)#,
dsgetq'odst2=dsgetq'entptr(9)#,
dsgetq'oaddr2=dsgetq'entptr(10)#,
dsgetq'olen2=dsgetq'entptr(11)#,

<< the second piece of the outgoing message                   >>

dsgetq'ides2=dsgetq'entptr(12)#,
dsgetq'idst2=dsgetq'entptr(12)#,
dsgetq'iaddr2=dsgetq'entptr(13)#,
dsgetq'ilen2=dsgetq'entptr(14)#,

<< the second buffer for the incoming message                 >>

dsgetq'odes3=dsgetq'entptr(15)#,
dsgetq'odst3=dsgetq'entptr(15)#,
dsgetq'oaddr3=dsgetq'entptr(16)#,
dsgetq'olen3=dsgetq'entptr(17)#,

<< the third piece of the outgoing message                    >>

dsgetq'ides3=dsgetq'entptr(18)#,
dsgetq'idst3=dsgetq'entptr(18)#,
dsgetq'iaddr3=dsgetq'entptr(19)#,
dsgetq'ilen3=dsgetq'entptr(20)#,

<< the third buffer for the incoming message                  >>

dsgetq'mainpin=dsgetq'entptr(21)#;

<< the main pin associated with this request                  >>
```

```
dsputq'fc=%21,

<< ATTACHIO function code for DSPUTQ request               >>

dsputq'entlen=8,

<< each request entry size in DSPUTQ buffer                >>

dsputq'buflen=dsputq'entlen*max'num'of'entries+1;

<< the total buffer size for the DSPUTQ request            >>


dsputq'num'of'entries=dsputq'bufptr#,

<< number of request/reply entries in DSPUTQ buffer        >>

dsputq'ucno=dsputq'entptr#,

<< user channel number associated with this entry          >>

dsputq'ioqindex=dsputq'entptr(1)#,

<< ioqindex associated with this entry                     >>

dsputq'dst = dsputq'entptr(2)#,
dsputq'addr = dsputq'entptr(3)#,
dsputq'len = dsputq'entptr(4)#,

<< the descriptor for the unsolicit incoming message       >>
<< or in DS message case, this will be the descriptor for  >>
<< the actual user data portion. the length shows the      >>
<< actual data received even if a truncation happened due  >>
<< to insufficient buffer size                             >>

dsputq'req'code=dsputq'entptr(5)#,

<< if dsputq'ioqindex = 0, this word has the following     >>
<< definition:                                             >>
<<                                                         >>
<< 1 - unsolicit incoming data ( level-0 )                 >>
<< 2 - unsolicit incoming data ( level-1 )                 >>
<< 3 - incoming clear                                      >>
<< 4 - incoming interrupt                                  >>
<< 5 - incoming reset                                      >>
<<                                                         >>
<< otherwise, the request code will be the same as given   >>
<< in the DSGETQ                                           >>

dsputq'status=dsputq'entptr(6)#,

<< if dsputq'ioqindex <> 0,  this is the completion status >>
<< for the given ioqindex, otherwise should be zero        >>
```

```
dsputq'uctype=dsputq'entptr(7).(0:8)#,

<< there are three different types of user channels:        >>
<<                                                          >>
<< 0 - HP3000 to HP3000 channel                             >>
<< 1 - HP3000 to PAD channel                                >>
<< 2 - HP3000 to HP1000 channel                             >>

dsputq'mainpin=dsputq'entptr(7).(8:8)#;

<< the main pin associated with the given user channel     >>
```

```
                  DSMONG & DSMONP buffer format
             DSMONG BUFFER                      DSMONP BUFFER

    +-------------------------+      +---------------------------+
    |       # OF ENTRIES      |      |        # OF ENTRIES       |
    |-------------------------|      |---------------------------|
    |                         |      |                           |
    |         ENTRY 1         |      |          ENTRY 1          |
    |                         |      |                           |
    |-------------------------|      |---------------------------|
    |                         |      |                           |
    |         ENTRY 2         |      |          ENTRY 2          |
    |                         |      |                           |
    |-------------------------|      |---------------------------|
    .                         .      .                           .
    .                         .      .                           .
    |-------------------------|      |---------------------------|
    |                         |      |                           |
    |         ENTRY N         |      |          ENTRY N          |
    |                         |      |                           |
    +-------------------------+      +---------------------------+

                           ENTRY FORMAT

    +-------------------------+      +---------------------------+
    |     REQUEST CODE        |      |         IOQINDEX          |
    |-------------------------|      |---------------------------|
    |        PARAMETER        |      |          STATUS           |
    |-------------------------|      +---------------------------+
    |        IOQINDEX         |
    |-------------------------|
    | PARAMETER DESCRIPTOR    |
    |-------------------------|
    |                         |
    +-------------------------+
```

```
dsmong'fc=%22,

<< ATTACHIO function code for DSMONG request                      >>

dsmong'entlen = 6,

<< the entry size for each request in DSMONG buffer               >>

dsmong'buflen = dsmong'entlen * max'monreq'entries + 1;

<< the buffer size for DSMONG request                             >>


dsmong'num'of'entries  = dsmong'bufptr#,
dsmong'req'code=dsmong'entptr#,
dsmong'parm = dsmong'entptr(1)#,
dsmong'ioqindex = dsmong'entptr(2)#,
dsmong'des = dsmong'entptr(3)#,
dsmong'dst = dsmong'entptr(3)#,
dsmong'addr = dsmong'entptr(4)#,
dsmong'len = dsmong'entptr(5)#;

<< the requests in DSMONG buffer are defined in the               >>
<< following table:                                               >>
<<                                                                >>
<< COMMAND      REQ'CODE      PARM       DESCRIPTOR               >>
<< -------      --------      ----       ----------               >>
<< OPEN            3        %(2)11       LINE'SPEED               >>
<< OPEN MASTER     3        %(2)10       LINE'SPEED               >>
<< OPEN SLAVE      3        %(2)01       LINE'SPEED               >>
<< SHUT            3        %(2)00       LINE'SPEED               >>
<< TRACE ON        0        CTRACEINFO   TRACE'FILE               >>
<< TRACE OFF       1                                              >>
<< MON             7        %(2)111                               >>
<< MON DS          7        %(2)011                               >>
<< MON CS          7        %(2)101                               >>
<< MOFF            7        %(2)000                               >>
<< DEBUG ON       %10          1                                  >>
<< DEBUG OFF      %10          0                                  >>
<< DEBUG N        %10          N                                  >>
```

```
dsmonp'fc=%23,

<< ATTACHIO function code for DSMONP request                 >>

dsmonp'entlen=2,

<< the entry size for each request in DSMONP buffer           >>

dsmonp'buflen=dsmonp'entlen*max'monreq'entries+1;

<< DSMONP buffer size                                         >>


dsmonp'num'of'entries=dsmonp'bufptr#,

<< number of entries in DSMONP buffer                         >>

dsmonp'ioqindex = dsmonp'entptr#,

<< the completed ioqindex                                     >>

dsmonp'status = dsmonp'entptr(1)#;

<< the completion status associated with the ioqindex         >>
```

DSMONX
USER CHANNEL INFORMATION TABLE

```
      +----------------------------------------+
      |       POINTER TO NEXT UCIT             |
      |----------------------------------------|
      |A|M|L|W|L'|                      | TYPE |
      |----------------------------------------|
      |          REQUEST QUEUE POINTER         |
---   |========================================|
      |  T-REQUEST CODE    |      T-MAINPIN    |
      |----------------------------------------|
 T    |               T-IOQINDEX               |
 R    |----------------------------------------|
 A    |          T-DATA DESCRIPTOR 1           |
 N    |----------------------------------------|
 S    |          T-DATA DESCRIPTOR 2           |
 M    |----------------------------------------|
 I    |          T-DATA DESCRIPTOR 3           |
 T    |----------------------------------------|
 T    |   TOTAL MESSAGE LENGTH (IN PACKETS)    |
 E    |----------------------------------------|
 R    |            PACKETS ACKNOWLEDGED        |
      |----------------------------------------|
      |            PACKETS LEFT                |
---   |========================================|
      |                R-IOQINDEX              |
      |----------------------------------------|
 R    |          R-DATA DESCRIPTOR 1           |
 E    |----------------------------------------|
 C    |          R-DATA DESCRIPTOR 2           |
 E    |----------------------------------------|
 I    |          R-DATA DESCRIPTOR 3           |
 V    |----------------------------------------|
 E    |       UNSOLICIT DATA DESCRIPTOR        |
 R    |----------------------------------------|
      |   TOTAL MESSAGE RECEIVED IN BYTES      |
      |----------------------------------------|
      |   TOTAL UNSOLICIT MSG RCVD IN BYTES    |
      |----------------------------------------|
      | R-REQUEST CODE     |    R-MAINPIN      |
      |----------------------------------------|
      |              READ TYPE                 |
---   |========================================|
      |        SPECIAL REQUEST IOQINDEX        |
      |----------------------------------------|
      |        CURRENT VCIT CONNECTED          |
      |----------------------------------------|
      |               STATUS                   |
      |----------------------------------------|
      |  REMOTE NODE NAME IN ASCII (8 BYTES)   |
      |----------------------------------------|
      |          HIGH LEVEL BUFFSIZE           |
      +----------------------------------------+
```

```
ucit'entlen = 41;

<< user channel information table size                        >>


ucit'nextent          = ucit'entptr#,

<< pointer to next free UCIT, used only in free list          >>

ucit'a'bit            = ucit'entptr(1).(0:1)#,

<< indicate the UCIT is currently allocated if set            >>

ucit'master           = ucit'entptr(1).(1:1)#,

<< indicate the local end is the call originator if set       >>

ucit'msglevel         = ucit'entptr(1).(2:1)#,

<< indicate a level-1 message is sent if set                  >>

ucit'wait             = ucit'entptr(1).(3:1)#,

<< indicate this UCIT is dequeued from active UC queue        >>
<< because the associated VC is busy if set                   >>

ucit'msglevel'        = ucit'entptr(1).(4:1)#,

<< indicate the message currently received is a level-1       >>
<< pad message if set                                         >>

ucit'type             = ucit'entptr(1).(13:3)#,

<< 0 - remote is DS/3000                                      >>
<< 1 - remote is PDN PAD                                      >>
<< 2 - remote is DS/1000                                      >>

ucit'next'ucioqp      = ucit'entptr(2)#,

<< point to next request queued onto this UCIT                >>

ucit'wreqcode         = ucit'entptr(3).(0:8)#,

<< write request code, same definition as in DSGETQ           >>

ucit'wmainpin         = ucit'entptr(3).(8:8)#,

<< the mainpin associated with this write request             >>

ucit'wioq             = ucit'entptr(4)#,

<< the ioqindex associated with this write request            >>

ucit'odes1            = ucit'entptr(5)#,
ucit'odst1            = ucit'entptr(5)#,
```

```
ucit'oaddr1          = ucit'entptr(6)#,
ucit'olen1           = ucit'entptr(7)#,

<< the first piece of the outgoing message ( DS HEADER )  >>

ucit'odes2           = ucit'entptr(8)#,
ucit'odst2           = ucit'entptr(8)#,
ucit'oaddr2          = ucit'entptr(9)#,
ucit'olen2           = ucit'entptr(10)#,

<< the second piece of the outgong message (DS APPENDAGE) >>

ucit'odes3           = ucit'entptr(11)#,
ucit'odst3           = ucit'entptr(11)#,
ucit'oaddr3          = ucit'entptr(12)#,
ucit'olen3           = ucit'entptr(13)#,

<< the third piece of the outgoing message (USER DATA)    >>

ucit'no'of'pckts     = ucit'entptr(14)#,

<< total number of packets worth of the outgoing message  >>

ucit'pckts'acked     = ucit'entptr(15)#,

<< total number of packets being acknowledged so far      >>

ucit'pckts'left      = ucit'entptr(16)#,

<< number of packets left to be transmitted               >>

ucit'rioq            = ucit'entptr(17)#,

<< the ioqindex associated with read, maybe the same value>>
<< as write ioqindex if a writeconversational request     >>

ucit'ides1           = ucit'entptr(18)#,
ucit'idst1           = ucit'entptr(18)#,
ucit'iaddr1          = ucit'entptr(19)#,
ucit'ilen1           = ucit'entptr(20)#,

<< the first buffer for incoming message (DS HEADER)      >>

ucit'ides2           = ucit'entptr(21)#,
ucit'idst2           = ucit'entptr(21)#,
ucit'iaddr2          = ucit'entptr(22)#,
ucit'ilen2           = ucit'entptr(23)#,

<< the second buffer for incoming message (DS APPENDAGE)  >>

ucit'ides3           = ucit'entptr(24)#,
ucit'idst3           = ucit'entptr(24)#,
ucit'iaddr3          = ucit'entptr(25)#,
ucit'ilen3           = ucit'entptr(26)#,
```

```
<< the third buffer for incoming message (USER DATA)      >>

ucit'ides4           = ucit'entptr(27)#,
ucit'idst4           = ucit'entptr(27)#,
ucit'iaddr4          = ucit'entptr(28)#,
ucit'ilen4           = ucit'entptr(29)#,

<< the temporary buffer used by DSMONX to keep unsolicit  >>
<< messages                                               >>

ucit'rlen123         = ucit'entptr(30)#,

<< the total length (in bytes) of the incoming message    >>

ucit'rlen4           = ucit'entptr(31)#,

<< the total length of the unsolicit incoming message     >>

ucit'rreqcode        = ucit'entptr(32).(0:8)#,

<< the read request code , same definition as in DSGETQ   >>

ucit'rmainpin        = ucit'entptr(32).(8:8)#,

<< the mainpin associated with this read request          >>

ucit'intioq          = ucit'entptr(33)#,

<< the ioqindex associated with interrupt request         >>

ucit'vcit'entptr     = ucit'entptr(34)#,

<< the VCIT associated with this UCIT                     >>

ucit'status          = ucit'entptr(35)#,

<< the completion status of the completd ioq              >>

ucit'lnode           = ucit'entptr(36)#,

<< the remote logical node name in ASCII form             >>

ucit'buffsize        = ucit'entptr(40)#;

<< the buffer size used by high level DS software         >>
```

DSMONX
VIRTUAL CIRCUIT INFORMATION TABLE

```
+------------------------------------------+
|       POINTER TO NEXT VCIT               |
|------------------------------------------|
|       CURRENTLY CONNECTED UCIT           |
|------------------------------------------|
|        VIRTUAL CIRCUIT NUMBER            |
|------------------------------------------|
|            RETRY COUNT                   |
|------------------------------------------|
|      P'(S)         |        P(S)         |
|------------------------------------------|
|      LWE           |        W            |
|------------------------------------------|
|      LWE'          |        W'           |
|------------------------------------------|
|                    |        WA           |
|------------------------------------------|
|Q|Q'|I|I'|         |M|M'|              |F|
|------------------------------------------|
|              TIMER LENGTH                |
|------------------------------------------|
|            TIMER ENTRY POINTER           |
+------------------------------------------+
```

```
vcit'entlen = 11;

<< the size of virtual circuit information table in words >>


vcit'nextent         = vcit'entptr#,

<< pointer to next available VCIT in free list            >>

vcit'ucit'entptr     = vcit'entptr(1)#,

<< pointer to the associated UCIT which connected to      >>

vcit'vcno            = vcit'entptr(2)#,

<< the virtual circuit number relative to LOW'VC          >>

vcit'retry'cnt       = vcit'entptr(3)#,

<< the retry count, for clear and reset requests          >>

vcit'p's             = vcit'entptr(4).(0:8)#,

<< send packet sequence number of last received data pckt >>

vcit'ps              = vcit'entptr(4).(8:8)#,

<< send packet sequence number of the ready-to-be-send    >>
<< data packet                                            >>

vcit'lwe             = vcit'entptr(5).(0:8)#,

<< local receiving buffers' lower window edge,            >>
<< LWE <= data packet <= LWE+W-1 are legal if not out of  >>
<< sequence                                               >>

vcit'w               = vcit'entptr(5).(8:8)#,

<< local window size                                      >>

vcit'lwe'            = vcit'entptr(6).(0:8)#,

<< remote receiving buffers' lower window edge, local site>>
<< should send data packet only within                    >>
<< LWE' <= data packet <= LWE'+W'-1                       >>

vcit'w'              = vcit'entptr(6).(8:8)#,

<< remote window size                                     >>

vcit'wa              = vcit'entptr(7).(8:8)#,

<< the number of data packets outstanding allowed before  >>
<< RR to remote                                           >>
```

```
vcit'qbit             = vcit'entptr(8).(0:1)#,

<< indicate if Q-bit should be set or not when sending    >>

vcit'qbit'            = vcit'entptr(8).(1:1)#,

<< indicate if Q-bit is set or not when data received     >>

vcit'ibit             = vcit'entptr(8).(2:1)#,

<< indicate there is a interrupt request outstanding      >>

vcit'ibit'            = vcit'entptr(8).(3:1)#,

<< indicate an interrupt indication has been received but >>
<< has not been confirmed yet                             >>

vcit'mbit             = vcit'entptr(8).(8:1)#,

<< indicate M-bit should be set when sending              >>

vcit'mbit'            = vcit'entptr(8).(9:1)#,

<< indicate M-bit is set when data received               >>

vcit'f'bit            = vcit'entptr(8).(15:1)#,

<< indicate the remote window is temporarily closed       >>

vcit'timer            = vcit'entptr(9)#,

<< indicate there is a timer outstanding if <> -1         >>

vcit'tmr'entptr       = vcit'entptr(10)#;

<< index to associate the outstanding timer in TRL        >>
```

            DSMONX Layout

```
       +----------------------+
       |                      |
       |                      |
       |                      |
       |                      |
       |                      |
       |                      |
       |                      |
DL --> |----------------------|          +-----------+
       |                      |          |           |
       |                      |          |  UCIOQ    |
       |        VCIT          |    ----->|           |
       |                      |    |     |   DST     |
       |                      |    |     |           |
DB --> |----------------------|    |     +-----------+
       |                      |====|
       |   GLOBAL VARIABLES   |    |
       |----------------------|    |
       |                      |    |     +-----------+
       |   TIMER REQ LIST     |    |     |           |
       |----------------------|    |     |  TEMPBUF  |
       |                      |    |---->|           |
       |   ACTIVE UC QUEUE    |    |     |    DST    |
       |----------------------|    |     |           |
       |  VC ALLOCATION TABLE |    |     +-----------+
       |----------------------|    |
       |                      |    |
       |         UCIT         |    |
       |                      |    |
 Q --> |----------------------|    |
       |                      |    |     +-----------+
       |                      |    |     |           |
       |                      |    |     |  LINEBUF  |
       |                      |    ----->|           |
       |                      |          |    DST    |
 S --> +----------------------+          |           |
                                         +-----------+
```

DS INFO DATA SEGMENT

The data segment no. for this XDS is in DSGLOBINFO(3).
DSGLOBINFO is a table in DSGLOBAL data segment.

```
                 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
                +==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==+
TABLE'SIZE      |             Number of real entries           |  %0   0
                |----------------------------------------------|
ENTRY'SIZE      |                  Entry size                  |  %1   1
                |----------------------------------------------|
NUMLDEVS        |             Number of IODSO ldevs            |  %2   2
                |----------------------------------------------|
LDEVSLISTSTART  |                First IODSO ldev              |  %3   3
                |----------------------------------------------|
                |               Second IODSO ldev              |  %4   4
                |----------------------------------------------|
                |                      .                       |
                |                      .                       |
                |----------------------------------------------|
                |                Last IODSO ldev               |
                |----------------------------------------------|
                |                      0                       |
                |----------------------------------------------|
                |                      0                       |
                |----------------------------------------------|
                |                      .                       |
                |                      .                       |
                |                      .                       |
                |----------------------------------------------|
                |                      0                       |
                |==============================================|
                |              Entry for first PIN             |
                |                      .                       |
                |                      .                       |
                |==============================================|
                |             Entry for second PIN             |
                |                      .                       |
                |                      .                       |
                |==============================================|
                |                      .                       |


                |                      .                       |
                |==============================================|
                |              Entry for last PIN              |
                |                      .                       |
                |                      .                       |
                +==============================================+
```

FORMAT FOR AN ENTRY IN THE DS INFO DATA SEGMENT

```
                0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
               |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
JOBNOX         | S/J |               Job number                 |  %0   0
               |------------------------------------------------|
TERMNOX        |            0         | Terminal number         |  %1   1
               |------------------------------------------------|
NAMEX          |                                                |  %2   2
               |         User name  (four words)                |
               |                                                |
               |------------------------------------------------|
ACCTX          |                                                |  %6   6
               |         User account (four words)              |
               |                                                |
               |------------------------------------------------|
FIRSTLDEVX     |         Info on first IODS0 ldev               | %12  10
               |               (two words)                      |
               |------------------------------------------------|
               |         Info on second IODS0 ldev              | %14  12
               |               (two words)                      |
               |------------------------------------------------|
               |                      .                         |
               |                      .                         |
               |                      .                         |
               |------------------------------------------------|
               |         Info on last IODS0 ldev                |
               |                                                |
               |================================================|
```

NOTES

S - Session
J - Job
The code for S or J is the same as in the Job Information Table. (JIT)

FORMAT FOR INFO ON AN IODSO LDEV
(THIS INFORMATION IS FOR EACH PIN)

```
                 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
                |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
ENTRYLDEVX      |   Number of DSOPENs   |       IODSO ldev      | %0   0
                |-----------------------------------------------|
ENTRYLDEVX1     | L| N| M| S| X| C| Q|//////////////////////////| %1   1
                |-----------------------------------------------|
```

NOTES

ENTRYLDEVX
No of DSOPENS - This corresponds to the PIN and IODSO ldev under consideration.
IODSO ldev - The logical device number of the IODSO device under consideration.

ENTRYLDEVX1
L - On if the PIN is on the remote (slave) side of the DS line.
N - On if a session exists for the PIN.
M - On if Master access is opened for this DS line.
S - On if Slave access is opened for this DS line.
X - On if the user has exclusive access over the DS line.
C - On if the user has set the compress option for the DS line.
Q - On if the user has set the quiet option for the DS line.

NOTES ON THE DS INFO DATA SEGMENT

This XDS contains information that is global to the system. Entry zero contains some header information . The number of real entries is the no of PINs allowed on the system. The ENTRY'SIZE is variable with each system and is:

ENTRY'SIZE = FIRSTLDEVX + NUMLDEVS*2

This allows two words of information for each IODSO ldev. Real entries contain information only if the corresponding PIN is alive and at least one DSOPEN has been executed by that PIN. If the first word of the entry is non-zero then it has information according to the above format else all the other words are also zero. An entry contains information on IODSO ldevs in the same order as in the header entry. A non-zero entry has non-zero information on all IODSO ldevs for which at least one DSOPEN has been executed. This XDS is initialized as a fixed size data segment at system startup time.

DS AFT
AVAILABLE FILE TABLE

(FSTYPE = 1 ONLY)

```
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
  |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
0 | FSTYPE    |          UNUSED                |MR| .AFT0
  |-----------------------------------------------|
1 |         RFNUM         |         LINENUM       | .AFT1
  |-----------------------------------------------|
2 |   RESERVED FOR FUTURE USE    (SET TO ZERO)    | .AFT2
  |-----------------------------------------------|
3 |                    IOQX                       | .AFT3
  |===============================================|
```

NOTES

AFT0

- FSTYPE - 0 = Local File
  - 1 = Remote File
  - 2 = DSNUM
  - 3 = DSNUM (No Wait)
  - 4 = CS File
  - 5 = CS File (With Auto Dial)
  - 6 = KSAM
- MR - Multi-record access

AFT1

- RFNUM - Remote file number
- LINENUM - Local line number of remote file

AFT2

Not currently used.

AFT3

- IOQX - No wait IOQX

DS AFT
AVAILABLE FILE TABLE
(FSTYPE = 2 OR 3)

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
    |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
 0  | FSTYPE    | C| M| P| R|        LDEV NUM       |  .AFT0
    |-----------------------------------------------|
 1  | DSDSCB INDEX    |          DSDST NUM          |  .AFT1
    |-----------------------------------------------|
 2  | PREVIOUS AFT POINTER  |      DS ERROR NUM     |  .AFT2
    |-----------------------------------------------|
 3  |                    IOQX                       |  .AFT3
    |===============================================|
```

NOTES

AFT0
  FSTYPE - 0 = Local File
           1 = Remote File
           2 = DSNUM
           3 = DSNUM (No Wait)
           4 = CS File
           5 = CS File (With Auto Dial)
           6 = KSAM
  C - On if DSOPEN called by CXDSLINE or REMOTE'HELLO.
  M - On if Master PTOP AFT.
  P - On if PTOP related.
  R - On if remote main process.
  LDEV NUM - Logical device number.

AFT1
  DSDST NUM - DS Data segment table pointer.
  DSDSCB INDEX - DS Dataseg control block index.

AFT2
  PREVIOUS AFT POINTER - Preceding DS open AFT Pointer.
  DS ERROR NUMBER - DS error number.

AFT3
  IOQX - No wait IOQX

OVERALL SESSION/JOB DS DATA STRUCTURE

```
   +==============+        +===============+
   |              |   +--->|               |<-----------------------+
   |Job Info Table|   |    |               |                        |
   |              |   |    |               |                        |
   |--------------|   |    |  Job DS XDS   |                        |
%14| (Job DS XDS)-----+    |               |                        |
   |--------------|        |               |                        |
   |              |        |               |                        |
   +==============+        +===============+                        |
                                                                    |
                     +-----------------------+                      |
   CI process stack  |                       |                      |
   +==============+  |     +==============+  |   +==============+   |
   |              |  | +-->|              |  +-->|              |   |
   |              |  | |   |--------------|      |--------------|   |
   |--------------|  | |   | (Job DS XDS)-----+  | (Job DS XDS)---->|
   | AFT (ldev b)----+ |   |--------------|   |  |--------------|   |
   |--------------|    |   |              |   |  |              |   |
   | AFT (ldev a)------+   |Device Process|   |  |Device Process|   |
   |--------------|        |    DS XDS    |   |  |    DS XDS    |   |
   |              |        | (ldev a, CI) |   |  | (ldev b, CI) |   |
   |              |        |              |   |  |              |   |
   +==============+        +==============+   |  +==============+   |
                                              |                     |
                     +----------------------+ +-------------------->|
   SON process stack |                      |                       |
   +==============+  |     +==============+ |    +==============+   |
   |              |  | +-->|              | +--->|              |   |
   |              |  | |   |--------------|      |--------------|   |
   |--------------|  | |   | (Job DS XDS)-----+  | (Job DS XDS)---->|
   | AFT (ldev b)----+ |   |--------------|   |  |--------------|   |
   |--------------|    |   |              |   |  |              |   |
   | AFT (ldev a)------+   |Device Process|   |  |Device Process|   |
   |--------------|    |   |    DS XDS    |   |  |    DS XDS    |   |
   | AFT (ldev a)------+   | (ldev a,SON) |   |  | (ldev b,SON) |   |
   |--------------|        |              |   |  |              |   |
   |              |        +==============+   |  +==============+   |
   |              |                           |                     |
   +==============+                           +-------------------->|
```

```
                  DS-RELATED PCBX STRUCTURES
             +================================+
PXGLOBAL     |                                |
             |================================|
PXFIXED      |                                |
             |================================|
PXFILE       |                                | 0
             |--------------------------------|
             |  DSOPEN Error |  CSOPEN Error  | 1
             |--------------------------------|
             |                                | 2
             |--------------------------------|
          +---- Last DS AFT  |  Slave DS AFT-----+
          |  |--------------------------------|  |
          |  |            . . . .             |  |
          |  |================================|  |
AFT       |  |            . . . .             |  |
          |  |================================|  |
DSNUM z-->+->|0010| |M|P|R|  |  DS Ldev       |  |
             |--------------------------------|  |
             | DSCB index | DevProc DS XDS ----------> DSCB in Dev-
             |--------------------------------|  |   Proc DS XDS
          +---- Previous AFT |  DS Error Num  |  |
          |  |--------------------------------|  |
          |  |             IOQX               |  |
          |  |================================|  |
          |  |            . . . .             |  |
          |  |================================|  |
DSNUM y-->+->|0010| |M|P|R|  |  DS ldev       |  |
             |--------------------------------|  |
             | DSCB index | DevProc DS XDS ----------> DSCB in Dev-
             |--------------------------------|  |   Proc DS XDS
          +---- Previous AFT |  DS Error Num  |  |
          |  |--------------------------------|  |
          |  |             IOQX               |  |
          |  |================================|  |
          |  |            . . . .             |  |
          |  |================================|  |
DSNUM x-->+->|0010| |M|P|R|  |  DS ldev       |<-+
             |--------------------------------|
             | DSCB index | DevProc DS XDS ----------> DSCB in Dev-
             |--------------------------------|      Proc DS XDS
          0<--- Previous AFT |  DS Error Num  |
             |--------------------------------|
             |             IOQX               |
             |================================|
             |            . . . .             |
             |================================|
DL           |                                |
```

JOB DS EXTRA DATA SEGMENT

```
                0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
               +==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==+
DSDSMAXSIZE    |   Maximum size of Job DS XDS (16K Max.)        |  %0    0
               |------------------------------------------------|
DSDSALLOC      |   Present size of Job DS XDS                   |  %1    1
               |------------------------------------------------|
DSDSFREESPPT   |   Pointer to free space area                   |  %2    2
               |------------------------------------------------|
DSDSOPENS      |   Number of DSOPENs active in this job         |  %3    3
               |------------------------------------------------|
               |   Print Buffer Pointer (SEG REL)               |  %4    4
               |------------------------------------------------|
               |   Print Buffer Size (+WORDS)                   |  %5    5
               |------------------------------------------------|
               |                       .                        |
               |                       .                        |
               |                       .                        |
               |                     Unused                     |
               |                       .                        |
               |                       .                        |
               |                       .                        |
               |------------------------------------------------|
DSLCBX area    |                                                !%14   12
               |     Job DS Line Control Block Extension        |
 (one DSLCBX   |                for open device                 |
  for each     |                                                !
  network      |------------------------------------------------|
  node.        |                       .                        !
  max. 8    )  |                       .                        !
               |                       .                        !
               |------------------------------------------------|
               |                                                |
               |     Job DS Line Control Bolck Extension        |
               |                for open device                 |
               |------------------------------------------------|
DSLCB area     |                                                |%114  76
               |     Job DSLCB for first opened device          |
(one DSLCB     |                                                |
 for each      |------------------------------------------------|
 configured    |                                                |%120  80
 DS device)    |     Job DSLCB for second opened device         |
               |                                                |
               |------------------------------------------------|
               |                       .                        |
               |                       .                        |
               |                       .                        |
               |------------------------------------------------|
               |                                                |
               |     Job DSLCB for last opened device           |
               |                                                |
               |------------------------------------------------|
               |                       .                        |
```

```
|                       .                        |
|                  Print Buffer                  |
|                       .                        |
|                       .                        |
|------------------------------------------------|
|           Size of free area (+words)           |
+================================================+
```

DEVICE/PROCESS DS EXTRA DATA SEGMENT

```
               0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
              +==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==+
DSDSMAXSIZE   |      Maximum size of this DS XDS (=16K)       |  %0   0
              |-----------------------------------------------|
DSDSALLOC     |          Present size of this DS XDS          |  %1   1
              |-----------------------------------------------|
DSDSFREESPT   |          Pointer to free space area           |  %2   2
              |-----------------------------------------------|
DSDSOPENS     |     No. of DSOPENs in process for device      |  %3   3
              |-----------------------------------------------|
DSDSRFAPT     |               RFA buffer pointer              |  %4   4
              |-----------------------------------------------|
DSDSRFASIZE   |           RFA buffer size (+words?)           |  %5   5
              |-----------------------------------------------|
DSDSIMAGE'PT  |          IMAGE control block pointer          |  %6   6
              |-----------------------------------------------|
DSDSCOMPBUFP  |          Compression buffer pointer           |  %7   7
              |-----------------------------------------------|
DSDSCOMPBUFSZ |     Compression buffer size (+words?/2)       | %10   8
              |-----------------------------------------------|
DSDSJOBXDS    |        Job DS XDS data segment number         | %11   9
              |-----------------------------------------------|
              |     Reserved as a temporary sratchpad area    | %12  10
              |-----------------------------------------------|
              |   Multi-Packet total message length (+bytes)  | %13  11
              |===============================================|
DSCB Pointer  |       Pointer to DSCB for first DSOPEN        | %14  12
  Area        |-----------------------------------------------|
              |       Pointer to DSCB for second DSOPEN       | %15  13
(one for      |-----------------------------------------------|
 each DSOPEN  |                       .                       |
 in process   |                       .                       |
 for device;  |                       .                       |
 64 maximum)  |-----------------------------------------------|
              |       Pointer to DSCB for last DSOPEN         |%113  75
              |===============================================|
              |                                               |%114  76
DSLCB Area    |         DSLCB for device and process          |
              |                                               |
(one for each |-----------------------------------------------|
 configured   |                       .                       |%120  80
 DS device;   |                       .                       |
 but only one |                       .                       |
 used)        |                                               |
```

```
                |===============================================|
                |                                               |
DSCB Area       |                                               |
                |          DS Control Blocks (DSCBs)            |
(elements in    |          (one for each DSOPEN)                |
 any order)     |                                               |
                |                                               |
                |-----------------------------------------------|
                |                                               |
                |                                               |
                |      Remote File Access (RFA) Buffer          |
                |        (one for slave side only)              |
                |                                               |
                |                                               |
                |-----------------------------------------------|
                |                                               |
                |                                               |
                |      Program-to-program (PTOP) Buffer         |
                |       (one for slave side PTOP only)          |
                |                                               |
                |                                               |
                |-----------------------------------------------|
                |                                               |
                |                                               |
                |          IMAGE Control Block                  |
                |       (one for remote IMAGE only)             |
                |                                               |
                |                                               |
                |-----------------------------------------------|
                |                                               |
                |                                               |
                |          Compression Buffers                  |
                |       (two for comp, READ/PRINTs)??           |
                |                                               |
                |                                               |
                |-----------------------------------------------|
                |                       .                       |
                |                       .                       |
                |                       .                       |
                |===============================================|
                |     Size of free space area (+words?)         |
                |-----------------------------------------------|
                |                                               |
                |                                               |
                |             Free Space Area                   |
                |                                               |
                |                                               |
                +===============================================+
```

DS LINE CONTROL BLOCK (DSLCB)

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
| No. DSOPENS for device| Logical device number |
|-----------------------------------------------|
|  User Channel Number  | From process number   |
|-----------------------------------------------|
|///////////|UC|QQ|Q1|Q0|C1|UP|RF|HI|PM|PG|PS|RS|
|-----------------------------------------------|
|          DS device buffer size (+words)       |
|===============================================|
```

			Job DSLCB	Device-process DSLCB
	DSLCBCOUNT	- No of DSOPENs for device	x	x
	DSLCBLDEV	- Logical device number	x	x
	DSLCBRMPNUM	- From process number		x
	DSLCBFRMNUM	- From process number		x
UC -	DSLCBUCF	- User Channel on this line		x
QQ -	DSLCBQTOQ	- QTOQ flag		
Q1 -	DSLCBQUIET1	- Suppress next output		x
Q0 -	DSLCBQUIET0	- QUIET mode specified	x	
C1 -	DSLCBCOMP	- Compress on this line	x	x
UP -	DSLCBSESSION	- Remote session up	x	
RF -	DSLCBFOPEN	- Remote FOPEN in progress		x
HI -	DSLCBHELLOOP	- DSOPEN on REMOTE HELLO	x	
PM -	DSLCBPTOPMSTR	- PTOP master on this line	x	x
PG -	DSLCBPTOPGET	- Slave PTOP was GET		x
PS -	DSLCBPTOP	- PTOP slave on this line		x
RS -	DSLCBRMPFLAG	- Process is remote slave	x	x
	DSLCBBUFSIZE	- DS device buffer size	x	x

DS Line Control Block Extension (DSLCBX)

```
        0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
       |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
   # 0| Logical Device Number |  User Channel Number  |%0
      |-----------------------------------------------|
     1|                                               | 1
      |---------         Logical          -----------|
     2|                                               | 2
      |---------          Node            -----------|
     3|                                               | 3
      |---------          Name            -----------|
     4|                                               | 4
      |-----------------------------------------------|
     5|             Virtual Buffer Size               | 5
      |-----------------------------------------------|
     6|                                            | S| 6
      |-----------------------------------------------|
     7|                                               | 7
      |-----------------------------------------------|
```

NOTES

S => DSLCBX entry is remote slave

DS CONTROL BLOCK (DSCB) WITH MESSAGE HEADER FORMAT

```
                0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
               |==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|==|
               |                       0                       |  0   %0
               |-----------------------------------------------|
DSLCBPT        |   Pointer to DSLCB for opened DS device       |  1   %1
               |-----------------------------------------------|
DSPROGNUM      |   Program number   (only for remote RTE)      |  2   %2
               |-----------------------------------------------|
DSIOCLASSNUM   |   I/O class number (only for remote RTE)      |  3   %3
               |-----------------------------------------------|
DSTERMNUM      |   Terminal number  (only for remote RTE)      |  4   %4
               |-----------------------------------------------|
DSPTOPBUFPT    |   PTOP buffer pointer                         |  5   %5
               |-----------------------------------------------|
DSPTOPBUFL     |   PTOP buffer size (+words??)                 |  6   %6
               |-----------------------------------------------|
DSRMTLENGTH    |   PTOP transfer length (+words/-bytes??)      |  7   %7
               |-----------------------------------------------|
DSPTOPFUNCT    |   PTOP function code   (from last GET)        |  8  %10
               |-----------------------------------------------|
               |///////////////////////////////////////////////|  9  %11
               |===============================================|
HEADBUF(0)     | Headlength (+words)    |  Message class       | 10  %12
               |-----------------------------------------------|
HEADBUF(1)     |         Remote computer id (always 0)         | 11  %13
               |-----------------------------------------------|
HEADBUF(2)     | R| E| C| B|CO| P|/////|  Stream type          | 12  %14
               |-----------------------------------------------|
HEADBUF(3)     |         Substream type (always 0)             | 13  %15
               |-----------------------------------------------|
HEADBUF(4)     | From process number    |  To process number   | 14  %16
               |-----------------------------------------------|
HEADBUF(5)     |///////////////////////////////////////////////| 15  %17
               |-----------------------------------------------|
HEADBUF(6)     |///////////////////////////////////////////////| 16  %20
               |-----------------------------------------------|
HEADBUF(7)     | Dsdatal (Appendage + data length, +bytes)     | 17  %21
               |===============================================|
               |                                               | 18  %22
               |                                               |
               |              Appendage Section                |
               |      (see message formats for specifics)      |
               |                                               |
               |                                               |
               |                                               |
               |===============================================|
               |                                               |
               |                                               |
               |           Unused (for appendage)              |
               |                                               |
               |                                               |153 %231
               |===============================================|
```

HEADER Definitions:

HEADLENGTH - length of header and appendage (+words)
MESSAGETYP - classifies message
COMPUTERID - unused
R - REPLYRECORD - on if message is a reply
E - REJECTRECORD - on if message has been rejected
C - CONTUFOLLOW - on if continuation record to follow
CO - COMPREC - on if data in message is compressed
B - ??? - on if in break mode
P - NOT'IN'PTOP - on if in master PTOP mode
STRMTYP - identifies message within message class
SUBSTRMTYPE - unused
FROMPROCESS - PIN of process from which message transmitted
TOPROCESS - PIN of process to which message sent
DSDATAL - length of appendage and data (+bytes)

POINTER LINKAGE IN DEVICE/PROCESS DS XDS

```
            +==================================+
            |        Maximum DS XDS size       |
            |----------------------------------|
            |        Present DS XDS size       |
            |----------------------------------|
            |        Free space pointer  ------------------------+
            |----------------------------------|                 |
            |        Number of DSOPENs         |                 |
            |----------------------------------|                 |
            |        RFA buffer pointer  --------------------+   |
            |----------------------------------|             |   |
            |        RFA buffer size           |             |   |
            |----------------------------------|             |   |
            |        IMAGE CB pointer    ----------------+   |   |
            |----------------------------------|         |   |   |
            |        Comp. buffer pointer------------+   |   |   |
            |----------------------------------|     |   |   |   |
            |        Comp. buffer size/2       |     |   |   |   |
            |----------------------------------|     |   |   |   |
            |        Job DS XDS number         |     |   |   |   |
            |----------------------------------|     |   |   |   |
            |                                  |     |   |   |   |
            |                                  |     |   |   |   |
            |                                  |     |   |   |   |
            |==================================|     |   |   |   |
    +--------------- First DSCB pointer        |     |   |   |   |
    |       |----------------------------------|     |   |   |   |
+---|--------------- Second DSCB pointer       |     |   |   |   |
|   |       |----------------------------------|     |   |   |   |
|   |       |                .                 |     |   |   |   |
|   |       |                .                 |     |   |   |   |
|   |       |                .                 |     |   |   |   |
|   |       |==================================|     |   |   |   |
|   |   +-->|                                  |     |   |   |   |
|   |   |   |       DS Line Control Block      |     |   |   |   |
|   |   |   |                                  |     |   |   |   |
|   |   |   |==================================|     |   |   |   |
|   |   |   |                                  |<----+   |   |   |
|   |   |   |       Compression Buffer(s)      |         |   |   |
|   |   |   |                                  |         |   |   |
|   |   |   |==================================|         |   |   |
|   +------>|                0                 |         |   |   |
|       |   |----------------------------------|         |   |   |
|       +---------  DSLCB pointer              |         |   |   |
|       |   |----------------------------------|         |   |   |
|       |   |                                  |         |   |   |
|       |   |       First DS Control Block     |         |   |   |
|       |   |                                  |         |   |   |
|       |   |==================================|         |   |   |
|       |   |                                  |<--------+   |   |
|       |   |       IMAGE Control Block        |             |   |
```

```
|       |   |                                  |             |   |
|       |   |==================================|             |   |
|       |   |                                  |<------------+   |
|       |   |           RFA Buffer             |                 |
|       |   |                                  |                 |
|       |   |==================================|                 |
+---------->|                0                 |                 |
        |   |----------------------------------|                 |
        +------------  DSLCB pointer           |                 |
            |----------------------------------|                 |
            |                .                 |                 |
            |                .                 |                 |
            |----------------------------------|                 |
            |       PTOP buffer pointer --------------+          |
            |----------------------------------|      |          |
            |                                  |      |          |
            |                                  |      |          |
            |      Second DS Control Block     |      |          |
            |                                  |      |          |
            |==================================|      |          |
            |                                  |<-----+          |
            |            PTOP Buffer           |                 |
            |                                  |                 |
            |==================================|                 |
            |                 .                |                 |
            |                 .                |                 |
            |                 .                |                 |
            |==================================|                 |
            |      Size of free space area     |<----------------+
            |----------------------------------|
            |                                  |
            |                                  |
            |            Free space            |
            |                                  |
            |                                  |
            +==================================+
```

DESCRIPTION OF SESSION/JOB DS DATA STRUCTURE

OVERALL SESSION/JOB DS DATA STRUCTURE

Each session and job which is using DS has its own data structure consisting of a set of extra data segments and areas within MPE tables. The MPE structures include:

- . Job Information Table (JIT): one word with the data segment number of the job's DS extra data segment.
- . Available File Table (AFT) entries: one for each active DS service within each process.

The DS extra data segments provide space for control information and data buffers. There are two types:

- . Job DS XDS: one for the entire job/session, with global DS information.
- . Device-process DS XDS: one for each remote system (i. e., DS device) being accessed by each process.

The following is a sample list of actions and the DS data structures created by those actions :

ACTIONS	CREATED STRUCTURES
:DSLINE to DS device a	Job DS XDS; sets JIT pointer DeviceProcess DS XDS for ldev a, CI AFT (to ldev a) in CI stack
:DSLINE to DS device b	DeviceProcess DS XDS for ldev b, CI AFT (to ldev b) in CI stack
:RUN son process	
.	
.	
.	
POPEN to DS device a	DeviceProcess DS XDS for ldev a, SON AFT (to ldev a) in SON stack
FOPEN to DS device a	AFT (to ldev a) in SON stack
POPEN to DS device b	DeviceProcess DS XDS for ldev b, SON AFT (to ldev b) in SON stack

AFT AND PCBX STRUCTURES

The PCBX area, at the beginning of each process stack below the DL address, is used by MPE, the File System, and datacomm subsystems to hold process-related information.

The File System's Available File Table (AFT) is the primary structure used by DS. Each DS service (remote commands, remote files and data bases, PTOP, etc.) in use by the process has assigned an AFT entry. An active DS service is identified by a DS number, which is equivalent to an opened file's file number. The DS number is used to index into the AFT to select the proper entry. Since the AFT starts at the end of the PCBX and grows back towards the beginning of the segment, DS numbers are used to compute a DB-minus address with the formula

AFT address (relative to DB) := DL address - AFTsize * DSnumber

There are two types of AFTs associated with DS:

A DS AFT supplies limited information about the DS service, including the ldev of the DS device to the remote. The DSCB Vector (word 2 of the AFT) specifies the DS Control Block in the Device-process DS XDS associated with the service. Each successful call to DSOPEN creates a new DS AFT. The AFT is deleted by DSCLOSE when the service is terminated.

A REMOTE FILE AFT is created by the File System when a remote file is FOPENed in a local process. It supplies the file number to be used by the remote File System (RFNUM) and the DS number of the DS AFT created for the remote file.

The active DS AFTs are linked in a chain, with the Last DS AFT field (PXFILE(3).(0:8)) giving the DS number of the most recently created DS AFT, and each Previous AFT field supplying the DS number of the preceding DS AFT. If the process is a remote CI, the slave DS AFT field (PXFILE(3).(8:8)) indicates the DS number of the DS AFT opened to reply to the master. Otherwise, the slave DS AFT field is zero.

Finally, error numbers for the last DSOPEN and COPEN executed are held in PXFILE(1). DSCHECK and CCHECK can look there for errors on opens.

JOB DS EXTRA DATA SEGMENT

The job DS XDS contains information that is global to the job, and is known to all processes using DS within the job. This includes the total number of active DS services (DSOPENs) and some control information for each DS device (line to a remote) access by the session. The job DS XDS has the same general format as the Device-Process DS XDS, with the unused data structures deleted.

DEVICE-PROCESS DS EXTRA DATA SEGMENT

A device-process DS XDS is created for each DS device (remote system) in use within the process. It holds control information and data buffers to be used for the process' communication with the remote. Originally (see above) there was only one DS XDS for all processes, but it was discovered that certain types of concurrent DS activity required separate sets of buffers and control blocks. The current data segment per device per process scheme solves these problems. Each of these segments has essentially the same format as the old DS XDS, so changes to DS code have been minimized.

Elements of the device-process DS XDS:

- DSCB pointer area: holds up to 64 pointers that link AFTs to DSCBs
- DSLCBs: see below; only one DSLCB in the segment
- DSCBs: see below
- RFA buffer: used for intermediate buffering of remote FREAD and FWRITE data
- PTOP buffer: used for intermediate buffering of PWRITE data on the slave side
- IMAGE control block: holds plabels for IMAGE intrinsics used in processing remote data base access requests; dynamically loaded via LOADPROC when the first remote DBOPEN executed. (When DS was released, both DS and IMAGE were optional products, so calls to IMAGE from DS could not be coded directly.)
- Compression buffers: one or two (depending on your point of view) buffers used during compression and decompression of data; also used to hold READ and PRINT data to and from the remote pseudo terminal.

DS LINE CONTROL BLOCK (DSLCB)

The DS Line Control Block holds control information pertaining to the use of a DS device, that is, access to a remote system. There are two types of DSLCBs:

> There is a JOB DSLCB (in the Job DS XDS) for each DS device being accessed by any process within the job. This DSLCB holds control information that is global throughout the job and/or must be available to all processes in the job. Some of this information is used for occasional processing, like the establishment and termination of the remote session (DSLCBHELLOP, DSLCBSESSION). A job DSLCB is created when a

:DSLINE DSdevice;OPEN is executed (the first DSOPEN for the DSdevice), and is destroyed when the :DSLINE DSdevice;CLOSE is performed.

There is one DEVICE-PROCESS DSLCB in each Device-process DS XDS. This DSLCB contains information relating to a particular process' access to a DS device. Some of the information is copied from the corresponding job DSLCB (e.g. DSLCBBUFSIZE, DSLCBLDEV). Other fields are used in DS activity local to the particular process (e.g. DSLCBFOPEN, DSLCBPTOPGET). The device-process DSLCB is created when its associated DS XDS is created (on the first DSOPEN in the process for the DS device), and is deleted when the DS XDS is released.

These two types of DSLCBs resulted when the original single DS data segment was split for the Moulinex fix. The format of the original DSLCB was retained, but certain fields are maintained only in the job or the device-process copies, and some fields are present in both. This was done on a functional basis -- those fields that are job-specific in nature and are rarely accessed are in the job DSLCB; those fields that are process-specific and/or frequently accessed are in the device-process DSLCB. Hopefully this minimizes the data segment switches, with most of the DS processing done with DB pointing at a device-process DS XDS.

NOTE: There are, unfortunately, TWO data structures called DS Line Control Blocks within DS. One (this one) is found in the DS XDSs and is used by the user services ("higher") level (DSSEG1-DSSEG5). The other is found in the DL-DB area of the DSMON process stack and is used by the DS IO ("lower") level (DSIOM, DSMISC, IODSO, IODSTRMO). Do not confuse them!

DS CONTROL BLOCK (DSCB)

A DS Control Block exists for each DS service in use by a process. Each DSOPEN (for a :DSLINE, FOPEN of a remote file, POPEN, etc.) creates a DSCB, and the corresponding DSCLOSE deletes the DSCB. The DSCB holds control information specific to a service and provides space for the header and appendage of messages relating to the service. There are three sections in the DSCB:

- Miscellaneous control information
    - a pointer to the DSLCB for the remote system
    - three words used for messages to RTE (HP1000) remotes
    - four words used for PTOP slave processing
- Message header - built by MANAGEWRITECONV for each request
    - message identification (class and stream type)
    - routing information (from and to processes)
    - various lengths (headlength, dsdatalength)
    - various status flags (rejection, continuation, etc.)
- Message appendage - supplied by the caller of MANAGEWRITECONV contains request specific information (such as FOPEN parameters). See Message Formats for details.

The DSCB is always allocated as a 153 word block to hold the largest appendage. Normally there will be unused space after the appendage.

BINARY SYNCHRONOUS COMMUNICATION FOR CS DIT

```
       0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     |--|--|--|--|--|--|-----------------------------|
     |     |AC|           |SI|IN|              |DR|ON|  ACTV=ACTIVE
    0|     |TV|           |O-|T-|              |V-|LI|0 INT-AC=INTERRUPT
     |     |  |           |ON|AC|              |EN|NE|          ACKNOWLEDGE
     |-----------------------------------------------|  DRV-EN=DRIVER
    1|                   NEXT DITP                   |1         ENTERED
     |-----------------------------------------------|
    2|                     IOQP                      |2
     |-----------------------------------------------|
    3|         UNIT          |       DLDEVN          |3
     |-----------------------------------------------|
    4|                     DLTP                      |4
     |-----------------------------------------------|
    5|                     ILTP                      |5
     |-----------------------------------------------|
     |     |TO|LO|                                   |  TO=TIMEOUT
    6|     |  |C-|                                   |6 LOC=LOCAL
     |     |  |TO|                                   |
     |-----------------------------------------------|
    7|       HARDWARE STATUS                         |7
     |-----------------------------------------------|
   10|            RESERVED                           |8
     |-----------------------------------------------|
   11|                   CONTROL P                   |9
     |-----------------------------------------------|
   12|                   LCM' DITP                   |10
     |-----------------------------------------------|
   13|                  EDIT' DITP                   |11
     |-----------------------------------------------|
   14|                   PD' DITP                    |12
     |-----------------------------------------------|
     |CM|  |PW|  |HD|SF|  |TO|  |BF|ID|US|     |LO|  |
(0)15|P-|  |R-|  |AB|AB|  |  |  |FZ|FZ|ER|     |C-|  |13(0) MASK
     |IN|  |FL|  |T |T |  |  |  |  |  |RQ|     |TO|  |
     |-----------------------------------------------|
     |CM|  |PW|  |HD|SF|  |TO|  |BF|ID|US|     |LO|  |
(1)16|P-|  |R-|  |AB|AB|  |  |  |FZ|FZ|ER|     |C-|  |14(1) FLAG
     |IN|  |FL|  |T |T |  |  |  |  |  |RQ|     |TO|  |
     |-----------------------------------------------|
(2)17|SUBTYPE |     DEV. TYPE      |       LCN       |15(2) LINE INFO
     |-----------------------------------------------|
(3)20|      TRANSFER LENGTH                          |16(3)
     |-----------------------------------------------|
(4)21|LAST RECOVERABLE ERROR |      ERROR CODE       |17(4)
     |-----------------------------------------------|
(5)22|        WAIT QUEUE                             |18(5)
     |-----------------------------------------------|
```

CS DIT (CONT.)

```
      |IN|IN|TR|IN|SPEED|    CS      |       CS          |
 (6)23|H-|H-|C-|H-|SELEC|   MODE     |      CODE         |19(6) COPTIONS
      |TO|ID|SP|CL|     |            |                   |
      |-----------------------------------------------|

      |-----------------------------------------------|
      |                        |           |DIAL |IN|CO|
 (7)24|      PROTOCOL          |           |TYPE |H-|N-|20(7) AOPTIONS
      |                        |           |     |BF|IO|
      |-----------------------------------------------|
      (  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15)
      |  |RE|NO|DS|END  |LD|AS|DB|DB|EX|MFW  |CH|NUM  |
(10)25|  |M-|-R|B-|SEQN |-G|-B|WK|T |-I|TYPE |A-|SYNCS|21(8) DOPTIONS
      |  |WT|VI|CT|     |PH|CC|  |TD|TB|     |WR|     |
      |-----------------------------------------------|
      |     |CO|           |AB|DU|HA|XMSN |SP|        | MISC
      |     |DE|           |T-|AL|LF|     |D-|        |
(11)26|     |SN| HSI CHAN  |AK|SP|SP|MODE |CH|        |22(9)
      |-----------------------------------------------|
      |               |IO|                            |  DSTINFO
(12)27|               |PR|        CS MISC DSTN        |23(10)
      |               |ES|                            |
      |-----------------------------------------------|
(13)30|               RECEIVE TIMEOUT                 |24(11)
      |-----------------------------------------------|
(14)31|                LOCAL TIMEOUT                  |25(12)
      |-----------------------------------------------|
(15)32|               CONNECT TIMEOUT                 |26(13)
      |-----------------------------------------------|
(16)33|                                               |27(14)
      |              INSPEED CHRS/SEC                 |
(17)34|                                               |28(15)
      |-----------------------------------------------|
(20)35|                                               |29(16)
      |              OUTSPEED CHRS/SEC                |
(21)36|                                               |30(17)
      |-----------------------------------------------|
      (0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15)
(22)37|RE|RE|TR|TR|IN|DI|ID|ID|ID|1 |2 |MS|AB|FI|PA|AB|
      |QU|CV|CO|PD|HN|RT|BI|FR|ER|ST|ND|TA|TL|ND|DA|PO|31(18) FLAGS
      |SD|ER|MP|RV|DL|BF|TS|ZN|R |IN|IN|TR|AT|ID|DD|LL|
      |-----------------------------------------------|
      |                                               |
(23)40|              MISC ARRAY                       |32(19)
      |                                               |
      |-----------------------------------------------|
      |                     TIME                      |
      |         (CHRONOS TIME OF LAST                 |33(20)
(24)41|      CONNECTION)(CALENDAR&CLOCK)              |
(25)42|                                               |34(21)
      |-----------------------------------------------|
(25)43|                                               |35(22)
      |               # MESSAGE SENT                  |
```

CS DIT (CONT.)

```
 (27)44|                                                |36(23)
       |------------------------------------------------|
 (30)45|                                                |37(24)
       |               # MESSAGES RECV'D                |
 (31)46|                                                |38(25)
       |------------------------------------------------|
 ------

       |------------------------------------------------|
 (32)47|               RECOVERABLE ERRORS               |39(26)
       |------------------------------------------------|
 (33)50|              IRRECOVERABLE ERRORS              |40(27)
       |------------------------------------------------|
 (34)51|    COMPLETION CODE     |      TIMEOUT CODE     |41(28)
       |------------------------------------------------|
 (35)52|               LOCAL TRLX TIMEOUT               |42(29)
       |------------------------------------------------|
 (36)53|               TIMEOUT TRLX                     |43(30)
       (1  2                 8                          )
       |------------------------------------------------| TRWR=TRACEWRAP
       |TR|TR|             |TR|                         | TRAL=TRACE ALL
 (37)54|AL|WR|  TRACE MASK |DR|    TRACE ENTRY NUM      |44(31)TRDR=TRACE
       |------------------------------------------------|          DRIVER
 (40)55|      MAX ENTRYS        |    CURRENT RETRYS     |45(32)
       |------------------------------------------------|
 (41)56|      LINE STATE        |                       |46(33)
       |------------------------------------------------|
 (42)57|                  XMSN LOG                      |47(34)
       |------------------------------------------------|
 (43)60|      CTS DELAY         |    PREEMP ERROR       |48(35)
       |------------------------------------------------|
 (44)61|                                                |49(36) DRIVER
       |------------------------------------------------|
 (45)62|                  CNTRLSEQ                      |50(37) PARM1
       |------------------------------------------------|
 (49)63|                  TIMEOUT                       |51(38) DRIVER
 (47)64|                  VALUE                         |52(39) PARM2
       |------------------------------------------------|
 (50)65|               OUTPUT BUFFER BANK               |53(40) DRIVER
       |------------------------------------------------|
 (51)66|               OUTPUT BUFFER                    |54(41) PARM3
       |------------------------------------------------|
 (52)67|               OUTPUT BUFFER LENGTH             |55(42) DRIVER
       |------------------------------------------------|
 (53)70|               INPUT BUFFER BANK                |56(43) PARM4
       |------------------------------------------------|
 (54)71|               INPUT BUFFER                     |57(44) DRIVER
       |------------------------------------------------|
 (55)72|               INPUT BUFFER LENGTH              |58(45) PARM 5
       |------------------------------------------------|
       (0  1  2         6  7  8                         )
       |TR|TR|TR|       |IN|DS|  RESPONSE               |
 (56)73|CE|CC|CF|       |& |R |  TIMEOUT                |54(46)
```

```
      |RR|OM|LH|       |PL|DL|                        |
      |-----------------------------------------------|
(57)74|                 BID TIMEOUT                   |60(47)
      |-----------------------------------------------|
```

CS DIT (CONT.)

```
          |-----------------------------------------------|
          (0 |        |4                                  )
   (60)75|PO|         |    BLOCK SIZE                     |61(48)
         |LC|         |                                   |
         |HG|         |                                   |
         |------------------------------------------------|
   (61)76|                 SEND MFW                        |62(49)
         |------------------------------------------------|
   (62)77|                 AGGREGATE XLOG                  |63(50)
         |------------------------------------------------|
         (                         |8                     )
  (63)100|   REQ STATION           |  CURRENT STATION    |64(51)
         |------------------------------------------------|
  (64)101|   # POLL ENTRIES        |  POLL LIST INDEX    |65(52)
         |------------------------------------------------|
  (65)102|                 TRACE IOQ                       |66(53)
         |------------------------------------------------|
  (66)103|                 POLL ENTRY DELAY                |67(54)
         |------------------------------------------------|
  (67)104|                 POLL REPEAT                     |68(55)
         |------------------------------------------------|
  (70)105|                 POLL LOOP DELAY                 |69(56)
         |------------------------------------------------|
  (71)106|                 CONFIG BUFFER SIZE              |70(57)
         |------------------------------------------------|
  (72)107|                 REQUEST IOQ                     |71(58)
         |------------------------------------------------|
  (73)110|                 HARD ABORT IOQ                  |72(54)
         |------------------------------------------------|
  (74)111|                 SOFT ABORT IOQ                  |73(60)
         |------------------------------------------------|
  (75)112|                 RETRANSMISSIONS                 |74(61)
         |------------------------------------------------|
  (76)113|                 # RESPONSE TIMEOUTS             |75(62)
         |------------------------------------------------|
  (77)114|                 # BCC ERRORS                    |76(63)
         |------------------------------------------------|
  100)115|                 # RECV TIMEOUTS                 |77(64)
         |------------------------------------------------|
  101)116|                 # OVERRUNS                      |78(65)
         |------------------------------------------------|
  102)117|                 PREVIOUS RECOV ERROR            |79(66)
         |------------------------------------------------|
  103)120|                 BUF 1 BYTES LEFT                |80(67)
         |------------------------------------------------|
  104)121|                 BUF 2 BYTES RIGHT               |81(68)
         |------------------------------------------------|
  105)122|                 RECV MFW                        |82(69)
         |------------------------------------------------|
```

CS DIT (CONT.)

```
|----------------------------------------------|
|                                              |
|         LINE CONTROL MONITOR (LCM)           |
|            SECTION OF THE DIT                |
|                                              |
|                                              |
|                                              |

|                                              |
(0  1  2  3        6  7  8     10 11 12        )
|----------------------------------------------|
|RC|SE|RE|RE|     |SD|RD|RD|   |DW|SV|TE|      |
|AK|AK|SP|SP|     |WA|RE|IN|   |N |AB|XT|      |
|CT|CT|TO|FG|     |CK|PT|TR|   |LD|RT|  |      |LCMP(0) LCMPFLAGS
|----------------------------------------------|
|                USER REQUEST                  |LCMP(1)
|----------------------------------------------|
|                CURRENT STATE                 |LCMP(2)
|----------------------------------------------|
|                TRACE STATE                   |LCMP(3)
|----------------------------------------------|
|                MRJE BUF 0                    |LCMP(4)
|----------------------------------------------|
|                MRJE BUF 1                    |LCMP(5)
|----------------------------------------------|
|                MRJE BUF 2                    |LCMP(6)
|----------------------------------------------|
|                                              |
|                LCM BUFFER (8 words)          |LCMP(7)-LCMP(14)

|                                              |
|----------------------------------------------|
|                                              |
|                EDITOR SECTION                |
|                  OF THE DIT                  |
               (DRIVER DEFINED)
|                                              |
|----------------------------------------------|
|                                              |
|               PHYSICAL DRIVER                |
              SECTION OF THE DIT
|              (DRIVER DEFINED)                |
|                                              |
|----------------------------------------------|
```

CS DIT FIELDS AND DEFINITIONS

MASK and FLAG Words 13 and 14	CMP-IN	Completion Interrupt
	PWR-FL	Power Fail
	HD-ABT	Hard Abort
	SF-ABT	Soft Abort
	TO	Timeout
	BF FZ	Buffer Frozen
	ID FZ	ID Frozen
	USER RQ	User Request
	LOC-TO	Local Timeout
COPTIONS Word 19	INH-TO	Inhibit Timeout
	INH-ID	Inhibit ID
	TRC-SP	CS Trace
	INH-CL	Inhibit :CLINE
AOPTIONS Word 20	INH-BF	Inhibit Buffering Override
	CON-IO	Concurrent IO
DOPTIONS Word 21	REM-WT	Delay Sequence Wait
	NO-RVI	Poll Termination Sequence
	DSB-CT	Disable Control Read
	END-SEQN	Ending Sequence
	LD-GPH	Leading Graphics
	AS-BCC	Value of US ASCII BCC
	DB WK	Disable WACK
	DB-TTD	Disable TTD
	EX ITB	Expect ITB
	MWF TYPE	Message Format Word
	CHA-WR	Chain Writes
	NUM-SYNCS	Number of Leading SYNCS
MISC Word 22	CODE SN	Code Sensing
	ABT-AK	Abort ACK
	DUAL SP	Dual Speed
	HALF SP	Half Speed
	XMSN MODE	Transmission Mode
	SPD-CH	Speed Changeable

CS DIT (CONT.)

DST INFO
Word 23

ID PRES	ID Present

FLAGS
Word 31

REQ USD	Request Used
RECV ER	Recoverable Error
TR COMP	Trace Out Completion
TR PDRV	Trace Out Physical Driver
IN HNDL	Interrupt Handler
DIRT BF	Dirty Buffer
ID BITS	ID Frozen Bits
ID FRZN	ID Frozen
ID ERR	ID MAM Error
1ST IN	First Interrupt
2ND IN	Second Interupt
MSTA TR	MMSTAT Trace
ABT LAT	Abort Later
FIND ID	Find Station ID
PAD ADD	Pad Added
AB POLL	Abort Poll

STANDARD (46)
Word 54

TRC ERR	Trace Error Toggle
TRC COM	Trace Complete
TRC FLH	Trace Flush
IN & PL	Increment and Poll
DSR DL	Date Set Ready Delay

LCMFLAGS
LCMP(0)

RC AKCT	Received ACK Counter
SE AK CT	Send ACK Counter
RESP TO	Response Timeout
RESP FG	Response Flag
SD WACK	Send WACK
RD REPT	Read Repeat
RD INTR	Read Interrupt
DWN LD	Download
SV ABRT	Save Abort
TEXT	Text

TERMINAL IOQ ELEMENT

```
       0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
      |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
      |            REQUEST DEPENDENT FLAGS             |
     0|                                                |     QFLAG
      |------------------------------------------------|
     1|                   NEXT IOQP                    |     QLINK
      |------------------------------------------------|
     2|        UNIT #          | LOGICAL DEVICE NUMB.  |     QLDEV
      |------------------------------------------------|
      |FL|                    |READSTOP |REQUEST STATE |     QMISC
      |------------------------------------------------|
     4|SF|        DATA SEGMENT NUMBER                  |     QDSTN
      |------------------------------------------------|
     5|            TARGET ADDRESS OFFSET               |     QADDR
      |------------------------------------------------|
     6|                       |     FUNCTION CODE      |     QFUNC
      |------------------------------------------------|
     7|          COUNT/XLOG/CONTROL RETURNS            |     QWBCT
      |------------------------------------------------|
   %10|       PARAMETER 1 (FUNCTION DEPENDENT)         |     QPAR1
      |------------------------------------------------|
   %11|       PARAMETER 2 (FUNCTION DEPENDENT)         |     QPAR2
      |------------------------------------------------|
      |                       |  QUALIFYING   |GENERAL |
   %12|          PCBN         |    STATUS     | STATUS |     QSTAT
      |------------------------------------------------|
```

BIT0 ABORT
BIT1 SPECIAL
BIT2 DIAGNOSTIC
BIT3 SYS BUFFER
BIT4 IO WAKE
BIT5 BLOCKED
BIT6 COMPLETED
BIT7 DATA FREEZE
BIT8 MAM ERROR
BIT9
BIT10-12 READ ERRORS
BIT13-15 RPLEVEL

TERMINAL IOQ FIELDS AND DEFINITIONS

QFLAG - Flags and request state.

ABORT	Abort this request and return an error indication to the caller.
SPECIAL	Special handling is to be applied to this request. Has no meaning for terminal requests.
DIAGNOSTIC	This is a request from a diagnostic subsystem. Not used by terminal system.
SYSBUFRS	Target is an index relative to the SBUF table of the data buffer.
IOWAKE	Wake caller on completion of request.
BLOCKED	Blocked I/O. The caller is waited in ATTACHIO until the request is completed. Implies wake.
COMPLETED	Request has been completed and caller woken if requested.
DATAFRZN	If set then the data segment has been frozen in memory. Set by MAM when a MAKEPRESENT request is successfully completed.
MAMERRD	An error has occurred in trying to make the target data segment present and freeze it in core.
READERRORS	This field contains a code specifying the resulting status on a read termination. 0 - no error 1 - read terminated on special read termination character 2 - read completed because break was enabled and detected and allowed. 3 - read data lost because of no TBUFS available, PTAPE swing buffer write not completed in time or term=11 and char following DC2 was not a CR. 4 - character lost because interrupt not service before next character was input 5 - read parity error occurred and parity checking enabled 6 - read timed out 7 - block mode read timed out

TERMINAL IOQ (CONT.)

RPLEVEL Request preempt level. If the preempt type of the request was zero then this is the value of TMODE when the request was queued, otherwise it is the preempt type of the request.

0 - terminal in normal mode and non preemptive request
2 - normal request, terminal was in console mode when the request was queued
3 - soft preemptive, preempt reads with no data input
4 - hard preemptive, preempt all non preemptive requests

QLINK - SYSDB relative pointer to the next IOQ element. Points to the first word of the next element.

QLDEV - Logical device number.

QLDEVN Logical device number

QMISC - Request state and flags

FLUSH (FL) This flag is set when a control Y is detected and accepted while this request was waiting or being processed. Causes reads and writes to be successfully completed, although no I/O takes place.

READSTOP Stop read operation if not zero.

0 - null or no stop
1 - break has been detected and is allowed
2 - subsystem break has been detected and is allowed
3 - request has been prempted
4 - read operation has been timed out
5 - request has been aborted
6 - block mode read has timed out

NOTE: BIT 10 is NO STOP bit;suppresses aborts and prompts

RSTATE Request state. Any codes not described below are unused.

0 - Request not started or new.
1 - Request has been started. Reads or writes may be waiting for the current write to finish to be continued.
2 - A read operation is in progress.
%43 - A read operation has been completed but the data has not been transferred to the callers buffer.
%44 - A read operation has been stopped. The cause and corresponding action to be taken is identified by the STOPREAD field in QMISC.
5 - Read initiation conditions have been checked and the read can be started as soon as the current operation (usually a write) is completed.
%30 - Waiting (because 270 bytes tanked or no TBUFs) to enter a CRLF because a post space write follows a previous prespace write.

%31 - Waiting (because 270 bytes tanked or no TBUFs) to enter prespace carriage control bytes.
%32 - Waiting (because 270 bytes tanked or no TBUFs9 to enter callers data into terminal buffers.
%33 - Waiting (because 270 bytes tanked or no TBUFs) to enter post space carriage control bytes.
%34 - %37 Correspond to states %30 - %33 but waiting to enter an ENQ for the 2640/44. When the ENQ has been entered into the TBUF, the state reverts to the current state -4.

STACKFLAG(SF) If the QADDR is the offset from DB to target address, otherwise QADDR is offset from DST base.

QDSTN - Contains the data segment number of the target data area.

QADDR - Offset to the target data area in the data segment or bank. For PTAPE reads, this word contains an SBUF index to the first of a pair of SBUFs used to read the data into.

QFUNC - Function code. See ATTACHIO description for details.

FUNC Function code field.

0 - read
1 - write
2 - file open
3 - file close
4 - device close
5 - set timeout
6 - set inspeed
7 - set outspeed
%10 - echo on
%11 - echo off
%12 - disable break
%13 - enable break
%14 - disable escape
%15 - enable escape
%16 - disable tapemode
%17 - enable tapemode
%20 - disable timer
%21 - enable timer
%22 - read timer
%23 - disable parity
%24 - enable parity
%25 - logged on
%26 - set parity
%27 - set terminal type
%30 - allocate terminal
%31 - clear flush and write
%32 - enable control X !!! echo
%33 - disable control X !!! echo
%34 - not used
%35 - PTAPE read
%36 - set/reset break mode
%37 - set/reset console mode
%40 - set parity
%41 - allocate terminal
%42 - set terminal type
%43 - return terminal type
%44 - return outspeed
%45 - set stop characters
%46 - change console interrupt
%47 - speed sense
%50 - powerfail recovery

QWBCT - Word or byte count and control returns. On initiation specifies a word count if positive or a byte count if negative. At completion of the request this location contains the actual transmission count in the same units as the call specified. Certain control requests return information through this location.

TERMINAL IOQ (CONT.)

QPAR1 - Parameter one. See first page of driver listing for details.

QPAR2 - Parameter two. See first page of driver listing for details.

> NOTE: During PTAPE reads, QPAR1 and QPAR2 contain a double word disc base address of the virtual memory area where the spooled data is saved temporarily.

QSTAT - Request completion status and PCB number associated with this request.

PCBN	PCB number associated with request. If zero this IOQ element is returned by the system when the request is completed.
QUALIFIER	A code which further defines or qualifies the general status. See ATTACHIO description for details.
STATUS	General status. Indicates the current or resultant status of the request accorddng to the following codes. 0 - not started or awaiting completion 1 - successfully completed 2 - end of file detected 3 - unusual condition 4 - irrecoverable error

```
DIT for ATC/SERIES II,III
-------------------------


      0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15

              DFLAG FOR A READ:
    -----------------------------------------------------------------
   0|TRM|UP |ACT|REQ|SIH|SPG|WWT|PR |NWL|PTY|TCH|BRD|    DSTATE      |
    |---------------------------------------------------------------|
              DFLAG FOR A WRITE:
    |---------------------------------------------------------------|
   0|TRM|UP |ACT|REQ|SIH|   |WWT| 1 |NWL|       |AWT|    DSTATE      |
    |---------------------------------------------------------------|
    |---------------------------------------------------------------|
   1|           SYS I/O PROC NEXT DIT POINTER                       | DLINK
    |---------------------------------------------------------------|
   2|           FIRST REQUEST IOQ POINTER                           | DIOQP
    |---------------------------------------------------------------|
   3|FLU|NCE|NPT|      UNIT         |        LOGICAL DEVICE #       | DLDEV
    |---------------------------------------------------------------|
   4|                        DLT POINTER                            | DDLTP
    |---------------------------------------------------------------|
   5|                        ILT POINTER                            | DILTP
    |---------------------------------------------------------------|
   6|HGU|DSC|CFT|TTO|HTO|   |SPE|SPW|RDT|ONL|DSY|LGO|BRK|ESC|BTO|STD| DRQST
    |---------------------------------------------------------------|
   7|TIM|TMR|DELECHO|FFD|    TTYPE        |EXS|CNP|   | PAIRCODE    | DTYPE
    |---------------------------------------------------------------|
 %10|PEM|  MTYPE    |CF |CB |SB |NSY|RCT|WCT|PMD| TMODE |  LPLEVEL  | DMODE
    |---------------------------------------------------------------|
  11|TPM|RES|SYN|ECH|SPS|ESC|   | OUTSPEED  |FIL|BOK|   INSPEED     | DSPEE
    |---------------------------------------------------------------|
  12| 0 | 0 |       UNIT        |PCL|PTY|  NEXT DSTATE  |PSL| 1 | 0 | DCNTR
    |---------------------------------------------------------------|
  13|                     REQUESTED COUNT IN BYTES                   | DRBC
    |---------------------------------------------------------------|
  14|                    READ/WRITE BYTE COUNT                      | DBCNT
    |---------------------------------------------------------------|
  15| WAITED STATE  |  HSTATE   |TTW|           TURN CHAR           | DSAVE
    |---------------------------------------------------------------|
  16|         SUB SYS BREAK CHAR      |          EOR CHAR           | DSTOP
    |---------------------------------------------------------------|
  17|       NEXT DITP OF BANDWIDTH WAITED DEVICE                    | DWAIT
    |---------------------------------------------------------------|
 %20| WRITE BYTES TANKED SO FAR / TIMEOUT LENGTH FOR BLOCK MODE READ| DXCNT
    |---------------------------------------------------------------|  DBTI
  21|         BYTE COUNT OF EOF SAVED READ                          | DRCNT
    |---------------------------------------------------------------|
  22|         COUNT TO END OF READ/WRITE TBUF                       | DCNT
    |---------------------------------------------------------------|
  23|         HEAD POINTER TO READ/WRITE TBUF's                     | DHEAD
    |---------------------------------------------------------------|
  24|         TAIL POINTER TO READ/WRITE TBUF's                     | DTAIL
    |---------------------------------------------------------------|
```

DIT FOR ATC/SERIES II/III (CONT.)

```
    |----------------------------------------------------------------|
 25|          BYTE POINTER TO NEXT READ/WRITE BYTE                  | DPNTR
    |----------------------------------------------------------------|
 26|          HEAD POINTER TO EOF SAVED READ TBUF's                 | DRPTR
    |----------------------------------------------------------------|
 27|      TERMINAL TYPE        |BWR|PTY SV|   |NFM|    DSPEED      | DLAST
    |----------------------------------------------------------------|
%30|          POINTER TO NEXT DIT IN TBUF WAIT LIST                 | DTBLK
    |----------------------------------------------------------------|
 31|          POINTER TO SAVED TBUF AFTER TBUF WAIT                 | DNXTB
    |----------------------------------------------------------------|
 32|          READ TIME/FIRST WORD OF DOUBLE TIMERS                 | DRTIM
    |----------------------------------------------------------------|  DRTI
 33|          2ND WORD OF DOUBLE READ START TIMER READING           |
    |----------------------------------------------------------------|
 34|          MAXIMUM READ TIME IN SECONDS                          | DRTMA
    |----------------------------------------------------------------|
 35|  LF SYNCS    |        CR SYNCS        |        SYNC COUNT      | DSYNC
    |----------------------------------------------------------------|
 36|          IOQP TO BROKEN READ SAVED DATE                        | DBREA
    |----------------------------------------------------------------|
 37|      2640/SPEED TRLX          |     LOGON/HANGUP/READ TRLX     | DTRLX
    |----------------------------------------------------------------|
%40|      CFAIL TRLX               |     TURN TRLX                  | DDSET
    |----------------------------------------------------------------|
 41|LOGONTY|XOW|AEJ|     CFAIL CNT     |       MCODE                | DMONI
    |----------------------------------------------------------------|


    |----------------------------------------------------------------|
 42|                                                                |
    |                MMSTAT TIMING INFO                              | DMMTI
 43|                                                                |
    |----------------------------------------------------------------|
 44|                                             |RQS| ESCSEQCNT  | DMISC
    |----------------------------------------------------------------|
```

DFLAG - FLAGS AND DEVICE STATE

TERMINAL	Device is a terminal
UP	If set, device is on line, has been speed sensed or has been initialized and can do I/O. If clear then in speed sense mode.
ACTIVE	If set, monitor is currently active servicing this device.
REQUEST	Service for this device was requested while the monitor was active.
SPECIH	Use special interrupt handler.
SPOOLING	Input has been requested through the PTAPE procedure.
WRTWAIT	A character or sync is in the process of being output and a completion interrupt is expected.
PAIR	Pair is set whenever no read is in progress or when the action on the next character is dependent on the previous character input or the previous state. See paircode for details on the various pair conditions.
NEWLINE	A linefeed was the last character input or output. Used to determine if a CR/LF is necessary on mode changes or at FOPEN time.
PTYCHK/ 2645K FLAG	Read data is to be checked for correct parity, and if incorrect a parity error indication is to be returned to the caller.
TERMCHAR	A special read termination character has been specified. The read data is to be checked and if the termination character is found the read will be terminated and the character set in the buffer. If the binaryread bit is set then this bit indicates a "transparent" read is in progress with sub system break and EOR characters in DSTOP. Both a termchar and a transparent read may be in progress simultanously if the termchar field of QPAR2 is not zero.

DIT for ATC/SERIES II,III (CONT.)

BINARYREAD A binary or transparent read was specified. If TERMCHAR is clear then a binary read is in progress. All 8 bits are transferred and no editing takes place. A binary read is teminated only when the count is satisfied. If termchar is set, then a transparent read is in progress. No editing takes place but only 7 bits are transferred. An EOR and sub system break character are held in DSTOP.

ACKWAIT An ENQ was sent to a 2640/44. Waiting for an ack or time out before continuing the write. Has this meaning during write operations only.

DSTATE Device state. Specifies the current device activity and is used to detemine the next state.

0 - null or no activity.
1 - writing.
2 - reading.
3 - XON write, reading next.
4 - turning 202 modem to write state, next state in NXTD STATE.
5 - wait for less terminal activity to start read/write
6 - end of record (EOR) LF in progress, null state next.
7 - EOR CR in progress, EOR CR state next.
%10 - EOR sync in progress, EOR CR state next.
%11 - write being waited for a break allowed check by term.
%12 - delete LF or delete echo character being written or start read next. Send XON to start read next.
%13 - delete CR being written, delete LF state next.
%14 - "!!!" or syncs being written. Next state is delete CR or saved in WAITEDSTATE if sync set.
%15 - 1st character of a termtype 11 read is being echoed.
%16 - have TIP start a read operation.
%17 - finish up read then do DSTATE operation held in NXTDSTATE.

DLINK - Link word for linked list of the devices waiting for service by the system IO process. If not zero or -1 (end of list) then a DIT pointer to the next device waiting.

DIOQP - SYSDB relative pointer to the first IOQ element in the request list for this device.

DLDEV - Logical device number and unit number.

FLUSH This flag is set whenever a break has been detected and accepted. While it is set, writes are returned completed without any I/O being done. Reads are returned with an unusual condition status, %173.
It also holds off any further break service requests.
It is reset with a function code 25 operation.

DIT for ATC/SERIES II,III (CONT.)

NO'CX'ECHO	if set, then "!!!" is not to be echoed when a control X is detected to delete a line.
.NO PTY	Termtype is 8 bit in nature.(no pty set or check allowed)
UNIT	unit number of device.
LDEVN	Logical device number.

DDLTP - SYSDB relative pointer to driver linkage table (DLT).

DILTP - SYSDB relative pointer to interrupt linkage table (ILT).

DRQST - Monitor service request flags. The requests are serviced in a left to right order. The bit position determines the priority with which the request is serviced.

HANGUPTO	Hangup timeout has been completed.
DISCNCT	Dataset has disconnected (dataset ready has dropped).
CFAILTO	Timeout started when carrier failed has completed. If 103 then hangup else try to turn 202 around again.
TURNTO	CB or SB is not true 5 seconds after starting the read to write turnaround on the 202. Hangup device.
HP2640TO	An ACKWAIT from an ENQ to 2640/44 has timed out. The ACKWAIT is terminated and the write restarted.
SPOOLEND	A control Y has been detected terminating PTAPE input.
SPOOLSW	Switch PTAPE input buffers.
READTO	A read operation has been timed out.
ONLINE	A colon has been input and the device speed sensed. If not connected through a dataset, initiate a log on time out.
DSETRDY	Dataset ready has been detected. Initiate a log on time out.
LOGONTO	A log on time out has occurred. The caller has not logged on. The device is hungup.
BRK	A break has been detected or SB has dropped while writing.

ESC A control Y has been detected.

BLOCK TO Block mode read has timed out before completion. Read is returned with IO timeout code.

STAT DONE Logical write and associated status request have been completed for 2631B.

DTYPE - Terminal type and other flags.

TIMING A request to measure the time taken to complete a read operation has occurred and the time at the initiation of the read has been saved in DRTIMED. When the read is completed, the time taken will be saved in DRTIME.

TIMEREAD The time required to complete a read operation is to be monitored and saved in DRTIME.

DELECHO This field contains a code which specifies the character to be output when a delete character (control H) is input. Different characters are output if the word count is zero to keep the carriage at the proper place.

CODE	INPUT<>0	INPUT=0	COMMENT
0	nothing	space	terminal backspaces
1	"/"	nothing	hard copy no backspace
2	line feed	space	hard copy backspaces
3	control Y	nothing	2600 control Y backspaces

FORMFEED If set then a form feed is output when the form feed character (%14) is to be output. If clear a LF is output in place of the form feed character. In either case, the character is preceeded by an XOFF and carriage return. Usually clear for terminals which do not respond to a form feed.

TTYPE terminal type as specified in the MPE ERS.

0 - ASR 33
1 - ASR 35
2 - ASR 37
3 - execuport
4 - datapoint
5 - Memorex
6 - terminet
7 - 2741 call 360
8 - 2741 PTTC/EBCDIC
9 - mini bee (HP2615)
10 - HP2640/44
11 - HP2640/44 & auto enter cap
12 - HP2645K Katakana/Roman data
13 - term connected to packet switching network or other computer
15 - HP2635A print term (8 bit)
16 - HP2635A print term (7 bit)
18 - Generic CRT
19 - HP2631B (7 bit)
20 - HP2631B (8 bit)
21 - HP2631B (7 bit)
22 - HP2631B (8 bit)

ETXSENT	End of Text (ETX) character has been sent to a 2640X on a 202 to stop the terminal from listening. Carrier may now be dropped.
CONSTRNTRPT.(11:1)	If set then Control A on the Console will cause PROGEN to be awoken. If clear, then Control A is ignored.

PAIRCODE when the action to be taken on the next character is dependant on the previous state or character input then this field contains a code specifying the previous character or condition.

0 - no read in progress

1 - XOFFPAIR. Last character input was an XOFF during a tapemode read on a terminet. EOR has been returned and if the next char is a CR then ignore it.

2 - DELETEPAIR. A LF was echoed on a char delete. No LF echo is needed if next char is a control H.

3 - ESCPAIR. Last character was an escape. Check next character for an escape sequence.

4 - NODATAYET. A "NONSYNC" terminal read has been started with echo on but no data has been input yet. If the first character is a DC2 then paircode is set to enter (the DC2 is not saved) othewise process as a regular character.

5 - NOECHO. A termtype 11 read has been started with echo off. If first char is a DC2 then set paircode to enter (1st char not saved) otherwise write character.

6 - CRWAIT. A 2640/44 block mode read has been satisfied and stopped and waiting for a CR to complete the read. No Control X checks are made to restart read.

7 - CRWAITLF. Same as CRWAIT but an LF is to be echoed if requested after the CR is detected. Continue read with echo on.

8 - ENTER. First character of a noecho read was a DC2. If next character not a CR then set Data Lost status, else set PRIMED and if Reading then restart read to input data.

9 - DC2PAIR. Last character read was a DC2 from a 2640/44. If the next character is a CR then set primed, delete all data input and restart read.

DMODEM - Modem state and control flags

PREMPT When set indicates that at least one request is preemptive. In this case a scan of the request list is made to determine which request should be processed first and if the current request is to be stopped.

DIT FOR ATC/SERIES II,III (CONT.)

MTYPE	Modem Type 0 - hardwired 2 - 202S 1 - 103 3 - 2002 4-7 -- Same as 0-3 except no speed sensing is done.
CF	Carrier detected status from dataset.
CB	Clear to send status from dataset. Request to send delayed.
SB	Secondary receive status. Senders CB when writing.
NOSYNC	If set specifies that no delays are used by this teminal. Instead an ENQ is sent after 80 characters and the write doesn't continue until an ACK is received or a timeout occurs. Set for 2640/44 terminals.
RDCOUNTED	When set, indicates the "number of terminals doing block mode reads counter" has been incremented and when this operation completes the counter is to be decremented.
WRTCOUNTED	When set, indicates that the "number of terminals doing writes" has been incremented and when this unit completes its operation the counter is to be decremented.
PRIMED	When set indicates an "ESC D" sequence has been written or a DC2 has been received by a NOSYNC terminal. Before any read operation is initiated to a primed terminal to do a block mode read, the number of terminals doing I/O must be less than 13. If it is greater then a request to start the read is queued.
TMODE	Terminal Mode. 0 - normal 1 - break mode 2 - console mode 3 - console mode and return to break mode
LPLEVEL	Preempt level of last request. If preempt level of new request is higher then generate a CR/LF. 0 - normal request 1 - Not Used 2 - normal request with terminal in console mode 3 - soft prempt (preempt reads with no input yet) 4 - hard preempt (preempt all requests)

DSPEED - Multiplexor speed and other flags.

TAPEMODE	Input from paper tape. No characters are emitted in response to delete commands or at end of record.

DIT FOR ATC/SERIES II,III (CONT.)

RESTART	If set indicates that a write completion interrupt has occurred while the terminal buffers were being filled. The filling procedure restarts the write by issuing a SYNC. During a read if this bit is set, the read is to be restarted when a CR is detected because a control X deleting the line was detected.

DIT for ATC/SERIES II,III (CONT.)

SYNC	If set and DSTATE=Repeating then SCOUNT contains the number of SYNC characters to be output after the completion of the current operation. If clear and DSTATE =Repeating, then SCOUNT contains the number of "!" remaining to be output in response to a Control X.
ECHO	If set specifies that characters read during input are to be echoed if the device is operating full duplex.
SPDSENSING	If set indicates that the device is in the speed sensing mode. When in the speed sensing mode a control has been sent to the multiplexor connecting the main channel to the diagnostic channels.
ESC	Control Y breaks have been enabled through an FCONTROL call.
OUTSPEED	A code used to determine the baud rate and character size of the data output. 0 - 240 CPS or not determined 1 - 240 characters per second (CPS) 2 - 120 CPS 3 - 60 CPS 4 - 30 CPS 5 - 15 CPS 6 - 10 CPS 7 - 14 CPS
FILLING	Set when IOTERMO is putting data into TBUFS. If the last TBUF is to be returned by TIP when this flag is set then the write is waited and DCNT is set to -2 by TIP to indicate TIP is waiting.
BRKOK	If set then break is allowed otherwise break is ignored. Set and cleared through FCONTROL calls.
INSPEED	A code used to determine the baud rate and character size to be used to input data. The codes have the same meaning as those specified in outspeed above.

DCNTRL - This is a control word output to the multiplexor board to send control and data to the particular channel. It also contains other information in the unused areas.

PCL - Parity Control bit. If set, parity is enabled. If it is zero, parity is disabled.

PARITY	This bit is ORED into the eighth bit position on all characters output. If the eighth bit is zero it represents the parity of the character output if the parity control option is selected, otherwise it represents the sense of the eighth bit output. Also represents the parity expected during a read. Set when speed sensed or by function 21.

NXTDSTATE This is the next DSTATE to be set after a 202 modem turnaround is completed. Also contains the next DSTATE after a FINISHREAD (DSTATE=%17) operation is completed.

PRESPLAST If set then the last write operation was a PRESPACE. If next write is a postspace and newline is not set then a CR/LF is output to clean up the carriage.

DRBCT - For read and write request, this word holds the requested transfer count in bytes.

DBCNT - During reads this word contains the number of characters input.

During writes it contains the number of characters remaining to be written, including any already written from the current TBUF.

DSAVE - Holds next DSTATE after waiting and repeating DSTATEs and also the next byte to be output after a 202 turnaround is completed.

WAITEDS Holds the current DSTATE when a break is detected and an operation is suspended in order that term may check that break is allowed. It also holds the next DSTATE after "SYNC's" are output in the repeating DSTATE.

HSTATE Hangup state.
0 - null or hungup
1 - on line or normal operating condition
2 - log on time out in progress
3 & 5 - INITWAIT. speed sense failed, disconnected speed
4 - DCLOSE issued, disconnect next.
6 - hangup turn to read is in progress. the 202 dataset needed to be put in read state before hanging up.
7 - hang up settling timeout is in progress. sensing delay, then reinitialize channel.

TURNTOWRT If DSTATE is TURN202, then if set indicates a turn to write else the turn is a turn to read.

TURNCHAR Holds the character to be output after the 202 is turned around from read to write.

DSTOP - Holds the subsystem break and end of record characters if not zero indicating no editing is to be applied to a read. If not zero then no editing is to be applied to the characters input except for the following characters.

BREAKCHAR Detection of this character causes the same action as the detection of control Y for a normal read.

EORCHAR Detection of this character terminates input. if the device is in tapemode or 264X doing block mode input, the read is not terminated until a CR is detected.

DWAIT - Link word for a liked list of the devices waiting to be started when the terminal activity decreases. If not zero then a DIT pointer of the next device waiting. If -1 then signifies that this device is the last one in the list.

DXCNT - Holds the number of bytes transferred so far to the TBUFs during a spacing or user's data transfer operation. Used to restart the TBUF fill operation after a wait because more than 270 bytes have already been tanked. (Valid for write.)

DBTIME- Contains the timeout length for block mode read. (Valid for read. This is the same word of the DIT as DXCNT.)

DRCNT - When read data has been saved because an EOF was returned this word contains the byte count of the saved data.

DCNT - During a write, this word contains the number of characters remaining to be written from the current TBUF. During a read it contains the number of characters remianing to fill the current TBUF or to satisfy the read count. Set to -2 to indicate a write completed during a fill operation. When -1 then new TBUF need to get next byte from.

DHEAD - A SYSDB relative pointer to the current TBUF being written from or the first TBUF of a linked list during a read.

DTAIL - A SYSDB relative pointer to the current TBUF being read into or the last TBUF of a linked list during a write.

DPNTR - A SYSDB relative byte index to the last byte written or to last byte read. During a read if a new buffer is to be gotten to save the current byte input then this pointer is set to -1.

DIT FOR ATC/SERIES II,III (CONT.)

DRPTR - When not zero, this word points to a linked list of TBUFs which contain the data saved from a read which returned an EOF requesting the read to be saved.

DLAST - Holds the default terminal type, parity save data and preconfigured speed code.

TERMT	Default terminal type. The terminal is set to this type when it is speed sensed.
BWRITE	If set the last write was in binary mode and PTYSAVE contains the original parity control and sense bits.
PTYSAVE	Holds the PTYCNTRL and parity bits during a binary write when parity generation is disabled and the parity sense is set to zero.
NEWFORM	Last carriage control was a form feed.
DSPEED	Preconfigured default speed code. See OUTSPEED for definition.

DTBLK - Link word for a linked list of the devices waiting for a TBUF to be available. If not zero or -1 (end of list) then a DITP pointer of the next device waiting.

DNXTB - Holds the pointer to a TBUF allocated to a device which has been waiting. Used to insure that a waiting device gets at least one TBUF when it comes to the top of the TBUF waiting list.

DRTIME- During a times read, this is the reading of the timer at the initiation of the read. After a timed read is completed, the time in 1/100 of a second is saved in DRTIME as a single word. If it is -1 then the time was greater than 32K.

DRTMAX- When a read operation time out is requested, this quantity represents the maximum time in seconds allowed for the read to be completed.

DSYNC - CR and LF SYNC counts and the current SYNC count

LFSYNC	Contains the number of SYNCs to be issued after a carriage return is output. If >7, then actual count will be (N-6)*5

DIT FOR ATC/SERIES II,II (CONT.)

CRSYNC	Contains the number of SYNCs to be issued after a carriage return is output. If >7, then actual count will be (N-6)*5.
SCOUNT	SYNC COUNTER. Represents the number of SYNCs remaining to be issued after the current SYNC character is completed. This field also holds the number of "!"'s remaining to be echoed after a control X is input.

NOTE - Holds 80 minus the number of characters written since the last read or ENQ for 2640/44 terminals. When this count goes to zero, an ENQ is inserted in the write stream.

DBREAK- On broken reads, this word holds a pointer to an IOQ element which contains the count, head, tail and DPNTR pointers used to restart the broken read.

DTRLX - Holds read and data set time out request indexes.

2640TRLX	holds the timer request index for 2640/44 block mode reads and ENQ/ACK time outs.
READTRLX	holds logon, hangup and read time out request indexes.

DDSET - Holds the TRLX indexes for the timeouts associated with the data set control operations.

CFAILTRLX	Holds the TRLX index to time out loss of carrier detect
TURNTRLX	Holds the TRLX index to time out turn the 202 to write

.LOGONTYPE- indicates type of logon type to this terminal
- 0= :DATA
- 1= :JOB
- 2= :HELLO

.XONWAIT - XOFF has been received during write, waiting for XON to continue. This bit is set when a write is paused by a CONTROL S.

.AUTOEJECT- 2631B will skip over perforations.

.CFAILCNT - carrier fail detect count

.MCODE - Monitor function and control code.
- .(13:5) - Function
  - 0 - Null or no monitoring
  - 1 - Call help
  - 2 - Monitor activity
  - 3 - Form Delta time histogram
  - 7 - Monitor calls/counts/initiations

DIT for ATC/SERIES II,III (CONT.)

.(10:1) - Apply above to DSET1,DSET2 and DSETCONTROL
.(11:1) - Apply above to TIP
.(12:1) - Apply above to TERM

DMMTIM - 2 words used for timing statistics

DMISC - miscellaneous bit fields:

.REQSTAT - requesting 2631B status
.ESCSEQCNT- index into excape sequence for 2631B and VIEW

During PTAPE reads, several of the DITP words are used for different purposes than those in a normal read. The words and their use are listed below.

DBCNT - A 16 bit logical quantity representing the total number of characters input during this PTAPE read.

DCNT - SYSDB relative pointer to the base of the SBUF currently being used to hold the data as it is input.

DHEAD - SYSDB relative pointer to the base of the SBUF to be written to virtual, memory or the pointer to the buffer to be used when the current one is full.

DTAIL/
DPNTR- Double word logical disc address to the area where the next SBUF is to be written in virtual memory when it is full or the PTAPE read is terminated.

DIT for ATC/SERIES II,III (CONT.)

TERMINAL SPEED ENCODING

The default speed code set in the DIT will be used to initialize both the input and output speeds. This parameter will be used to determine the speed when an FCONTROL 37 (Allocate Terminal) is issued which does not specify a speed.

CODE (Future rel)	SPEED (Baud)	CODE (SERIES II/III)
0	Undefined	0
1	Externally Clocked	
2	50	
3	75	
4	110	6
5	134.5	7
6	150	5
7	200	
8	300	4
9	600	3
10	1200	2
11	4800	1
13	7200	
14	9600	
15-63	Reserved for future expansion	

The default speed code will be set in word %27 bits 10 thru 15 of the terminal DIT.

```
      |                                                   |
      |0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15|
       ---------------------------------------------------
  %27 |   Terminal Type   |BWT|PTYSV|///|NFM| DEFAULT SPEED| 23 DLAST
       ---------------------------------------------------
      |                                                   |
```

ADCC DIT/SERIES 30,33,40,44

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
   |-----------------------------------------------|
  0|TM|UP|AC|RQ|SH|SP|MA|PR|NL|PC|TC|BR|   DSTATE   | 0
   |-----------------------------------------------|
  1|   SYSIO PROCESS NEXT DIT POINTER              | 1 DLINK
   |-----------------------------------------------|
  2|   FIRST REQUEST IOQ POINTER                   | 2 DIOQP
   |-----------------------------------------------|
  3|FL|NE|RF|DO|   SYNCSTATE|   LOGICAL DEVICE #   | 3 DLDEV
            STAT
            REQ
   |-----------------------------------------------|
  4|        DLT POINTER                            | 4 DDLTP
   |-----------------------------------------------|
  5|        ILT POINTER                            | 5 DILTP
   |-----------------------------------------------|
  6|HU|DC|CF|TT|TO|AW|SW|SE|RT|OL|DR|LO|BK|SK|BT|SD| 6 DRQST
   |-----------------------------------------------|
  7|TM|TR|DLECH|FF|     TTYPE     |WX|CI| PAIRCODE | 7 DTYPE
   |-----------------------------------------------|
%10|PM| MTYPE  |CF|CB|SB|NS|RC|WD|PR|TMODE|LP LEVEL| 8 DMODEM
   |-----------------------------------------------|
 11|TM|RS|EO|EC|SS|SB| OUTSPEED  |RT|BO| INSPEED   | 9 DSPEED
   |-----------------------------------------------|
 12|HW|LL|SS|DONXTMOD|DM|PO|OP|NEXTDSTATE |PS|FL|AE|10 DCNTRL
   |-----------------------------------------------|
 13|      REQUESTED BYTE COUNT                     |11 DRBCT
   |-----------------------------------------------|
 14|RD CHAR ALREADY INPUT/CHARS LEFT TO WRITE      |12 DBCNT
   |-----------------------------------------------|
 15|WAITEDSTATE| HSTATE|TW|DA|CC|BC|PE|NOT|SR|II|CO|13 DSAVE
                                      LOGON
   |-----------------------------------------------|
 16| SUBSYS BREAK CHAR      |   EOR CHAR           |14 DSTOP
   |-----------------------------------------------|
 17| DITP OF NEXT DEV WAITING FOR BANDWIDTH        |15 DWAIT
   |-----------------------------------------------|
%20| WRITE BYTES TRANSFERRED SO FAR                |16 DXCNT/DBTIME
   |-----------------------------------------------|
 21| BYTE COUNT OF EOF SAVED DATA                  |17 DRCNT
   |-----------------------------------------------|
 22| READ/WRITE COUNT TO END OF CURRENT TBUF       |18 DCNT
   |-----------------------------------------------|
 23| HEAD POINTER TO READ/WRITE TBUFS              |19 DHEAD
   |-----------------------------------------------|
 24| TAIL POINTER TO READ/WRITE TBUFS              |20 DTAIL
   |-----------------------------------------------|
 25| BYTE OFFSET IN TBUF TO START CHANNEL PROGRAM  |21 DPNTR
   |-----------------------------------------------|
 26| HEAD POINTER TO EOF SAVED READ TBUFS          |22 DRPTR
   |-----------------------------------------------|
 27|   TERM TYPE        |BW|EB|NF| DEFAULT SPEED   |23 DLAST
```

ADCC DIT/SERIES 30,33,40,44 (CONT.)

```
    |------------------------------------------------|
%30  POINTER TO NEXT DIT IN TBUF WAIT LIST           |24 DTBL
    |------------------------------------------------|
 31| POINTER TO SAVED TBUF AFTER TBUF WAIT           |25 DNXTB
    |------------------------------------------------|
 32| TOTAL READ TIME / 1ST WORD OF TIMER READING     |26 DRTIME/DRTIMED
    |------------------------------------------------|
 33| 2ND WORD OF TIMER READING                       |27
    |------------------------------------------------|
 34| MAX READ TIME IN SECONDS                        |28 DRTMAX
    |------------------------------------------------|
 35| LF SYNC    | CR SYNC    | SYNC COUNT            |29 DSYNC
    |------------------------------------------------|
 36| IOQP TO INFO ON SAVED BROKEN READ DATA          |30 DBREAK
    |------------------------------------------------|
 37| 2640 TRLX               |LGON/HNGUP/RDTIMR TRLX |31 DTRLX
    |------------------------------------------------|
%40| CFAIL TRLX              | TURN TRLX             |32
    |------------------------------------------------|
 41| NUMBER OF BYTES IN OUTSTANDING TANKS            |33 DTANKB
    |------------------------------------------------|
 42|LGNTY|SYNST| CFAIL COUNT      |  LF COUNT        |34 DMONTR
    |------------------------------------------------|
 43| POINTER TO BEGINNING OF SIO PROGRAM             |35 DSIOPC
    |------------------------------------------------|
 44| POINTER TO SECOND TBUF USED FOR READ            |36 DBLKTAIL
    |------------------------------------------------|
```

DIT INFORMATION

0 - DFLAG

.TERM (0:1) - SET IF DEVICE IS A TERMINAL

.UP (1:1) - SET IF DEVICE IS ON LINE AND HAS BEEN SPEED SENSED, OR HAS BEEN INITIALIZED (BY ALLOCATING TERMINAL) AND READY TO DO IO

.ACTIVE (2:1)- SET IF IOTERMO IS CURRENTLY ACTIVE SERVICING THIS TERMINAL

.REQUEST (3:1)- SET IF SERVICE FOR THIS TERMINAL IS REQUESTED WHILE IOTERMO IS ACTIVE

.SPECIH (4:1) - SET IF SPECIAL INTERUPT HANDLER IS USED, NOT APPLICABLE

.SPOOLING (5:1) - A READ OERATION TO USE SYSBUF HAS BEEN REQUESTED THROUGH THE PTAPE PROCEDURE

.MODACTIVE (6:1) - SET IF SIO PROGRAM TO CONTROL MODEMS IS CURRENTLY ACTIVE

.PAIR (7:1) - SET (1) WHEN NO READ IS IN PROGRESS, OR (2) DURING READING, THE NEXT CHARACTER INPUT MAY REQUIRE SOME SPECIAL ACTION, SEE PAIRCODE FOR DETAILS

.NEWLINE (8:1) - SET IF THE LAST CHARACTER OUTPUT IS A LF, USED TO DETERMINE IF A CR/LF IS NECESSARY DURING MODE CHANGES OR AT FOPEN TIME

.PTYCHK (9:1) - SET IF PARITY CHECKING/GENERATION IS ENABLED, ODD/EVEN PARITY IS DETERMINED BY ODDPTY IN DCNTRL

.TERMCHAR (10:1) - SEE BINREAD

.BINREAD (11:1) --

TERMCHAR	BINREAD	
0	0	REGULAR READ
0	1	BINARY READ IN PROGRESS, THE READ IS ONLY TERMINATED WHEN THE REQUESTED BYTE COUNT IS SATISFIED
1	0	SPECIAL EOR CHARACTER IS SPECIFIED IN QP2 TO TERMINATE READ
1	1	TRANSPARENT READ IN PROGRESS, NO EDITING IS PERFORMED ON INPUT DATA, READ IS TERMINATED BY EOR CHARACTER SPECIFIED IN DSTOP OR QP2 OR SUBSYS BREAK CHARACTER IN DSTOP

.ENQACKWAIT (11:1) - DURING WRITE, BIT 11 IS SET WHEN THE CURRENT CHANNEL PROGRAM SUSPENDS THE WRITE BY SENDING AN ENQ AND THEN WAITS FOR AN ACK FROM THE TERMINAL

.DSTATE(12:4) - DEVICE STATE OF THE TERMINAL, SPECIFIES THE CURRENT ACTIVITY AND DETERMINES THE NEXT STATE

- 1 - WRITING
- 2 - READING
- 4 - TURN202; CURRENTLY TURNING AROUND THE 202 MODEM TO DO READ OR WRITE, NEXT DSTATE IS IN DCNTRL.NXTDSTATE
- 6 - EORLF; END OF RECORD CARRIAGE CONTROL IN PROGRESS, NULL STATE NEXT
- 7 - SPDSENSW -- SPEED SENSE SIO IN PROGRESS
- %10 - EORSYNC
- %11 - WAITED; READ OR WRITE OPERATION BEING SUSPENDED, WAITING FOR IOTERMO TO CHECK IF BREAK IS ALLOWED
- %14 - REPEATING; "!!!" BEING WRITTEN AFTER CONTROL X IS DETECTED, EORLF NEXT TO OUTPUT CR/LF
- %16 - MODEMSIO; CHANNEL PROGRAM CURRENTLY ACTIVE IN SETTING UP THE ADCC MODEM CONTROL LOGIC. WHEN THE CHANNEL PROGRAM COMPLETES, IF DCNTRL.DOMOD IS SET, A NEW CHANNEL PROGRAM IS STARTED TO SET THE MODEM LOGIC TO A NEW SET OF CONDITIONS. THE NEXT DSTATE IS IN NXTDSTATE.
- %17 - FINREAD; FINISH UP READ OPERATION AND PERFORM THE DSTATE INDICATEDIN NXTDSTATE.

1 - DLINK

LINK WORD FOR A LINKED LIST OF DEVICES WAITING FOR SERVICE BY THE SYSTEM I/O PROCESS.
0 => NONE WAITING
-1 => LAST DEVICE ON LINKED LIST
DITP -- A POINTER TO THE DIT OF THE NEXT WAITING DEVICE

2 - DIOQP

SYSDB RELATIVE POINTER TO THE 1ST IOQ ELEMENT IN THE SERVICE REQUEST LIST FOR THIS DEVICE

3 - DLDEV

.FLUSH (0:1) - SET WHEN A BREAK HAS BEEN DETECTED AND ACCEPTED. AS LONG AS IT REMAINS SET, ALL WRITE REQUESTS ARE RETURNED AS COMPLETED WITHOUT ANY ACTUAL I/O BEING PERFORMED. READS ARE RETURNED WITH AN UNUSUAL CONDITION STATUS, %173.
.NOCXECHO (1:1) - IF SET, THEN "!!!" IS NOT ECHOED WHEN A CONTROL X TO DELETE A LINE HAS BEEN DETECTED
.RDFLUSH (2:1) - NO TBUFS; FLUSH READ, WAIT FOR EOR
.LDEVN (8:8) - LOGICAL DEVICE NUMBER
DO STAT REQ (3:1) - SET WHEN A STATUS REQUEST IS NEEDED FROM A 2631B REMOTE SPOOLED PRINTER.
SYNCSTATE (4:4) - SAVES SYNC CHARACTER INTERRUPT CODE FOR HALF DUPLEX MODES.
.ABORWRT (5:1) - WRITE SIO HAS BEEN ABORTED

4 - DDLTP

SYSDB RELATIVE POINTER TO THE DRIVER LINKAGE TABLE (DLT)

5 - DILTP

SYSDB RELATIVE POINTER TO INTERRUPT LINKAGE TABLE (ILT)

6 - DRQST

REQUESTS FOR IOTERMO SERVICE THAT HAVE BEEN GENERATED BY TIP. THE REQUESTS ARE SERVICED IN A LEFT TO RIGHT ORDER, SO THE BIT POSITION DETERMINES THE REQUEST PRIORITY.

.HANGUP (0:1) - DATASET HANGUP TIMEOUT HAS BEEN COMPLETED
.DISCNCT (1:1) - DATASET HAS BEEN DISCONNECTED (CC HAS DROPPED)
.CFAILTO (2:1) - TIMEOUT FOR CARRIER FAIL HAS BEEN COMPLETED, HANGUP A 103 MODEM OR TRY TO TURNAROUND A 202.
.TURNTO (3:1) - CB OR SB FROM THE 202 MODEM DID NOT RISE 5 SECONDS AFTER STARTING THE "READ TO WRITE TURNAROUND", HANG UP THE DATASET.
.2640TO (4:1) - A 10 SECOND TIMEOUT TO WAIT FOR AN ACK FROM THE TERMINAL HAS EXPIRED, TERMINATE THE WAIT AND RESTART THE WRITE OPERATION
.SPOOLSW (6:1) - ONE OF THE TWO SYSBUFS USED FOR PTAPE READ HAS BEEN FILLED, SWITCH THEM SO THAT IT CAN BE EMPTIED ONTO DISC.
.SPOOLEND (7:1) - A CONTROL Y TO TERMINATE PTAPE READHAS BEEN DETECTED
.READTO (8:1) - A READ OPERATION HAS BEEN TIMED OUT
.ONLINE (9:1) - ALSO SPFOUND, A CR HAS BEEN INPUT AND SPEED SENSED, INITIATE A LOG ON TIMEOUT
.DSETRDY (10:1) - DATASET READY (CC) HAS BEEN DETECTED, INITIATE A LOGON TIMEOUT
.LOGONTO (11:1) - A LOGON TIMEOUT HAS EXPIRED AND THE CALLER STILL HAS NOT LOGGED ON; HANGUP THE DEVICE
.BRK (12:1) - A BREAK HAS BEEN DETECTED, OR SB FROM THE DATASET HAS DROPPED DURING A WRITE OPERATION
.SSBRK (13:1) - A SUBSYSTEM BREAK HAS BEEN DETECTED
.BLOCKTO (14:1) - BLOCK MODE READ HAS TIMED OUT
.STATDONE (15:1) -

7 - DTYPE

.TIMING (0:1) - SET IF THE TIME REQUIRED TO DOMPLETE THE CURRENT READ OPERATION IS TO BE RECORDED, THE STARTING TIME HAS BEEN RECORDED IN DRTIME, WHEN THE READ IS COMPLETED, THE ELAPSED TIME WILL BE SAVED IN DRTIME
.TIMEREAD (1:1) - SET WHEN THERE IS A REQUEST TO MEASURE THE TIME REQUIRED TO COMPLETE A READ OPERATION, CAUSES TIMING TO GET SET WHEN THE READ IS INITIATED.
.DELECHO (2:2) - THIS FIELD CONTAINS A CODE WHICH SPECIFIES THE REQUIRED ACTION WHEN A CONTROL H IS DETECTED
.FORMFEED (4:1) - SET FOR TERMINALS THAT RESPOND TO A FORMFEED, IF CLEAR, A LF IS SENT IN PLACE OF THE FF CHARACTER; THE CHARACTER TO BE OUTPUT (FF OR LF) IS PRECEDED BY A XOFF AND CR.
.TTYPE (5:5) - TERMINAL TYPE, A SUBSET OF THE SERIES III TERM TYPES
.WAITXON (10:1) - WAITING FOR XON
.CONSINTRPT (11:1) - SET IF CONTROL A CAN BE ACKNOWLEGED WHEN THE TERMINAL IS USED AS A SYSTEM CONSOLE

.PAIRCODE (12:4) - WHEN THE NEXT INCOMING CHARACTER MAY REQUIRE SPECIAL ACTION, THIS FIELD CONTAINS A SPECIAL CODE SPECIFYING THE CONDITIONS AND ACTIONS TO BE TAKEN:

- 0 - NO READ IN PROGRESS
- 1 - CRWAIT; A BLOCK MODE READ HAS BEEN SATISFIED AND STOPPED, NOW WAITING FOR A CR TO COMPLETE THE READ
- 2 - CRWAITLF; SAME AS CRWAIT BUT AFTRE THE CR IS DETECTED, A LF IS TO BE ECHOED IF REQUESTED
- 3 - NOECHO; A TERMTYPE 11 READ HAS BEEN STARTED WITH ECHO OFF, IF THE FIRST INCOMING CHARACTER IS A DC2, THEN A BLOCK MODE READ IS ABOUT TO BEGIN, OTHERWISE THE CHARACTER IS TO BE ECHOED BACK TO THE TERMINAL AND ECHO TO BE TURNED BACK ON.
- 4 - DC2PAIR; THE LAST CHARACTER READ WAS A DC2, IF THE NEXT CHARACTER IS A CR AND IF OWN DC1/DC2 HANDSHAKE IS ENABLED, THE READ OPERATION WILL BE COMPLETE; IF THE NEXT CHARACTER IS A CR AND OWN DC1/DC2 HANDSHAKE DISABLED, THEN THE CR IS IGNORED AND READ WILL CONTINUE.
- 5 - NODATAYET; A REGULAR READ HAS BEEN STARTED WITH ECHO ON.

8 - DMODEM

.PREMPT (0:1) - WHEN SET BY ATTACHIO, AT LEAST ONE PENDING REQUEST IS PREEMPTIVE

.MTYPE (1:3) - MODEM TYPE:

- 0 - HARDWIRED TERMINAL
- 1 - 103 MODEM
- 2 - 202C MODEM
- 3 - 2002 MODEM
- 4-7 => SAME AS 0-3, BUT NO SPEED SENSING (6&7 NOT CURRENTLY SUPPORTED)

.CF (4:1) - CURRENT CARRIER DETECT STATUS FROM MODEM

.CB (5:1) - CURRENT CLEAR TO SEND STATUS FROM MODEM

.SB (6:1) - CURRENT SECONDARY RECEIVE STATUS FROM MODEM

.NOSYNC (7:1) - SET FOR HP263X, HP264X TERMINALS; INDICATES THAT NO DELAYS BETWEEN CHARACTERS ARE NECESSARY FOR THIS TERMINAL, INSTEAD, AN ENQ IS SENT AFTER EVERY 80 CHARACTERS AND THE WRITE OPERATION IS SUSPENDED UNTIL AN ACK IS RECEIVED OR A 10 SECOND TIMEOUT OCCURS.

.PRIMED (10:1) - INDICATES THAT A DC2 HAS BEEN RECEIVED FROM THE TERMINAL DOING A FAST READ. A BLOCK MODE READ IS IN PROGRESS.

.TMODE (11:2) - TERMINAL MODE:

- 0 - NORMAL
- 1 - BREAK MODE
- 2 - CONSOLE MODE
- 3 - CONSOLE MODE AND RETURN TO BREAK MODE

.LPLEVEL (13:3) - PREEMPT LEVEL OF LAST REQUEST, IF PREEMPT LEVEL OF THE NEW REQUEST IS HIGHER, CR/LF IS TO BE OUTPUT TO THE TERMINAL:

- 0 - NORMAL REQUEST
- 2 - NORMAL REQUEST WITH TERMINAL IN CONSOLE MODE
- 3 - SOFT PREEMPT (PREEMPT READ OPERATION THAT HAS NOT INPUT ANY DATA YET)

4 - HARD PREEMPT (PREEPMT ALL READS)

9 - DSPEED

.TAPEMODE (0:1) - CURRENT INPUT IS FROM PAPER TAPE, INCOMING CHARACTERS ARE TRANSPARENT

.RESTART (1:1) - WHEN THE TERMINAL IS IN TAPEMODE OR BLOCK MODE READ AND A CONTROL X HAS BEEN DETECTED, PAIRCODE IS SET TO CRWAIT TO WAIT FOR A CR T TERMINATE THE READ, AT WHICH TIME THE READ IS TO BE RESTARTED

.ECHOON (2:1) - ECHO WAS TURNED OFF, REENABLE IT FOR CURRENT OPERATION

.ECHO (3:1) - IF SET, ALL INCOMING CHARACTERS ARE TO BE ECHOED IF OPERATING IF FULL DUPLEX MODE

.SPDSENSING (4:1) - SET IF CURRENTLY IN SPEED SENSE MODE, THE FIRST PORTION OF A POSSIBLE CR HAS BEEN IDENTIFIED AND WAITING TO RECEIVE THE REST OF THE CHARACTER.

.SSBRKOK (5:1) - SUBSYSTEM BREAKS HAVE BEEN ENABLED VIA A FCONTROL CALL.

.OUTSPEED (6:4) - CONTAINS AN ADCC CODE FOR THE CURRENT OUTPUT BAUDRATE; ADCC CODES FOR DIFFERENT BAUDRATES:

% 7 - 240 CPS
%10 - 960 CPS
%11 - 480 CPS
%13 - 120 CPS
%15 - 30 CPS
%16 - 15 CPS
%17 - 10 CPS

.RESTARTSPDS (10:1) - RESTART IDLE WAIT OR SPEEDSENSE AFTER CURRENT CHANNEL PROGRAM COMPLETES.

.BRKOK (11:1) - BREAK IS ALLOWED IF SET, OTHERWISE IGNORED. SET AND CLEARED VIA FCONTROL CALLS.

.INSPEED (12:4) - CANTAINS AN ADCC CODE FOR THE CURRENT INPUT BAUDRATE

10 - DCNTRL

.HIOPWAIT (0:1) - THE ACTIVE CHANNEL PROGRAM CANNOT BE HALTED IMMEDIATELY WHEN AN HIOP INSTRUCTION WAS EXECUTED; A SUBSEQUENT INTERRUPT WILL OCCUR AND SOFTWARE IS TO IGNORE IT.

.LFLAST (1:1) - A POSTSPACE LF HAS BEEN TANKED INTHE WRITE TBUF'S

.SPDSIO (2:1) - SET WHEN AN IDLE WAIT CHANNEL PROGRAM IS ACTIVE, WHEN THE TERMINAL IS NOT ACTIVE DOING READ/WRITE, AN IDLE WAIT PROGRAM IS STARTED TO LISTEN TO THE KEYBOARD.

.DONXTMOD (3:3) - AN ATTEMPT TO START A CHANNEL PROGRAM TO CONTROL THE ADCC MODEM LINES FAILED BECAUSE A PREVIOIUS MODEM CONTROL PROGRAM IS STILL ACTIVE. THIS FIELD CONTAINS A CODE SPECIFYING THE CONTROL TO BE DONE WHEN THE PREVIOUS CHANNEL PROGRAM COMPLETES AND THE NEW ONE CAN BE STARTED.

.DOMOD (6:1) - ATTEMPT TO START A MODEM CONTROL CHANNEL PROGRAM FAILED BECAUSE A PREVIOUS ONE IS STILL ACTIVE; WHEN IT COMPLETES, START THE MODEM CONTROL CHANNEL PROGRAM AS SPECIFIED IN DONXTMOD

.PTYON (7:1) - SPECIFIES PARITY GENERATION ON WRITE DATA AND PARITY CHECKING ON READ DATA

.ODDPTY (8:1) - IF SET, ODD PARITY IS USED FOR GENERATION AND CHECKING, OTHERWISE EVEN PARITY IS USED.

.NXTDSTATE (9:4) - CONTAINS THE NEXT DSTATE TO BE USED WHEN A 202 MODEM TURNAROUND IS COMPLETED, ALSO CONTAINS THE NEXT DSTATE WHEN A FINISHREAD (DSTATE=%17) OPERATION IS COMPLETED.

.PRESPLAST (13:1) - INDICATES THAT THE LAST WRITE OPERATION WAS A PRESPACE WRITE, IF THE NEXT WRITE IS POSTSPACE AND NEWLINE IS NOT SET THEN A CR/LF IS OUTPUT TO START WRITING A NEW LINE.

.FILLING (14:1) - INDICATES THAT IOTERMO IS CURRENTLY TRANSFERRING WRITE DATA FROM THE CALLER'S STACK INTO A TBUF.

.ADDENQ (15:1) - IOTERMO IS CURRENTLY PUTTING AN ENQ INTO THE TBUF AFTER 80 BYTES OF WRITE DATA HAVE BEEN TANKED.

11 - DRBCT

HOLDS THE REQUESTED READ/WRITE BYTE COUNT

12 - DBCNT

DURING A READ OPERATION, IT SPECIFIES THE NUMBER OF BYTES THAT HAVE BEEN READ. DURING A WRITE OPERATION, IT SPECIFIES THE NUMBER OF BYTES REMAINING TO BE WRITTEN.

13 - DSAVE

.WAITEDSTATE (0:4) - HOLDS THE CURRENT DSTATE WHEN A BREAK IS DETECTED AND THE CURRENT OPERATION SUSPENDED SO THAT IOTERMO MAY CHECK THAT BREAK IS ALLOWED, IF DISALLOWED, THE CURRENT DSTATE WILL BE RESUMED.

.HSTATE (4:3) - THE MODEM HANGUP STATE:

- 0 - NULL OR HUNGUP
- 1 - ON LINE OR NORMAL OPERATION
- 2 - LOGGINGON; LOG ON TIMEOUT IN PROGRESS
- 4 - DCLOSE ISSUED, DISCONNECT NEXT
- 6 - HANGUPTURN; HANGUP TURNAROUND TO READ IN PROGRESS, THE 202 MODEM NEEDS TO BE IN A READING STATE BEFORE HANGUP
- 7 - HANGUP SETTLING TIMEOUT IN PROGRESS

.TURNTOWRT (7:1) - WHEN THE 202 MODEM IS BEING TURNAROUND (DSTATE= TURN202), A 1 INDICATES TURNAROUND TO WRITE, A 0 INDICATES TURNAROUND TO READ.

.DELACK (8:1) - AN ENQ HAS JUST BEEN SENT DURING A WRITE WHEN A BREAK WAS DETECTED, DELAY THE NEXT WRITE FOR 0.5 SECOND TO AVOID OVERRUNNING THE TERMINAL.

.CC (9:1) - THE CURRENT DATASET READY STATUS FROM MODEM

.BLOCKRD (10:1) - DURING A READ OPERATION, 2 CHANNEL PROGRAMS, EACH WITH ITS OWN TBUF, ARE USED TO SERVICE INCOMING DATA; THIS BIT IS SET IF THE 2ND CHANNEL PROGRAM IS CURRENTLY ACTIVE RECEIVING DATA.

.AUTOEJECT (11:1) - 2631B WILL SKIP OVER PERFORATIONS

.NOTLOGON (12:1) - IF CLEAR AND THERE IS A LOGON TIMER GOING, THEN YOU ARE IN A SPEEDSENSE MODE. IF SET AND

THERE IS A LOGON TIMER GOING, YOU THEN ARE IN
TIMING SEQUENCE FOR A MODEM.
.REQSTAT (13:1) - REQUESTING 2631B STATUS
.ININ (14:1) - INITIALIZING TERMINAL PORT
.CCON (15:1) - CC ALWAYS ON

14 - DSTOP
IF NOT ZERO, CONTAINS THE USER SPECIFIED SUBSYSTEM BREAK AND END OF RECORD CHARACTERS. IF THEY ARE SPECIFIED, THEN NO EDITING IS DONE TO THE INCOMING DATA DURING A READ.
.BRKCHAR (0:8) - DETECTION OF THIS CHARACTER DURING READING CAUSES THE SAME ACTION AS THAT OF A CONTROL Y.
.EORCHAR (8:8) - DETECTION OF THIS CHARACTER TERMINATES THE READ AND IS INCLUDED WITH THE REST OF THE READ DATA TO BE TRANSFERED TO THE CALLERS STACK

15 - DWAIT
LINK WORD FOR A LINKED LIST OF DIT'S WAITING TO DO I/O WHEN THE TERMINAL ACTIVITY DECREASES,
0 - NONE WAITING
-1 - THIS DIT IS THE LAST ONE ON THE LIST
OTHER - A DIT POINTER TO THE NEXT DEVICE WAITING

16 - DXCNT(WRITE)/DBTIME(READ)
DXCNT (VALID DURING WRITES) INDICATES THE NUMBER OF BYTES TRANSFERRED SO FAR INTO TBUF'S WHEN CARRIAGE CONTROL BYTES OR DATA BYTES ARE BEING TANKED. USED TO RESTART THE FILL TBUF OPERATION WHEN 540 BYTES HAVE ALREADY BEEN TANKED AND THE FILL OPERATION HAS TO BE SUSPENDED.
DBTIME (VALID DURING READ) - TIMEOUT PERIOD FOR BLOCKMODE READ.

17 - DRCNT
CONTAINS THE BYTE COUNT OF THE READ DATA SAVED WHEN AN EOF WAS DETECTED.

18 - DCNT
DURING A WRITE, IT INDECATES THE NUMBER OF CHARACTERS TO BE WRITTEN BY THE CURRENT EXECUTION OF THE CHANNEL PROGRAM. DURING A READ, IT INDECATES THE NUMBER OF CHARACTERS TO BE READ BY THE CURRENT CHANNEL PROGRAM. WHEN=-2, IT INDECATES THAT ALL TANKED DATA HAS BEEN WRITTEN OUT AND THAT IOTERM0 IS INTHE MIDDLE OF FILLING A TBUF.   FILLING A TBUF.

19 - DHEAD
A SYSDB RELATIVE POINTER TO
(1) DURING WRITE, THE CURRENT TBUF CONTAINING DATA TO BE WRITTEN,
(2) DURING READ, THE 1ST TBUF ON THE LINKED LIST OF INPUT DATA.

20 - DTAIL
A SYSDB RELATIVE POINTER TO
(1) DURING WRITE, THE LAST TBUF ON THE LINKED LIST OF TANKED DATA,
(2) DURING READ, THE CURRENT TBUF USED FOR RECEIVING DATA.

21 - DPNTR

A WORD POINTER USED DURING WRITES TO INDICATE THE OFFSET WITHIN A TBUF OF THE 1ST BYTE OF DATA TO BE WRITTEN BY THE CURRENT CHANNEL PROGRAM.

22 - DPNTR

A SYSDB RELATIVE POINTER TO A LINKED LISTOF TBUF'S CONTAINING THE DATA SAVED WHEN AN EOF WAS DETECTED.

23 - DLAST

.TERMTYPE (0:7) - THE DEFAULT OR CONFIGURED TERM TYPE. WHEN THE TERMINAL IS SPEED SENSED, THIS IS THE TERM TYPE USED.

.BINWRT (7:1) - SET IF THE LAST WRITE OPERATION WAS IN BINARY MODE.

.EIGHTBITS (8:1) - SET IF THE 8-BIT PROTOCOL IS USED AND PARITY GENERATION/CHECKING IS DISALLOWED. USED FOR TERM TYPES 12 AND 15.

.NEWFORM (9:1) - LAST CARRIAGE CONTROL WAS A FORM FEED.

.DEFAULTSPEED (10:6) - THE ADCC CODE OF THE DEFAULT OR CONFIGURED TERMINAL BAUDRATE.

24 - DTBLK

A DIT POINTER TO THE NEXT TERMINAL WAITING FOR A TBUF.

25 - DNXTB

A POINTER TO A TBUF ALLOCATED TO A TERMINAL WHICH HAS BEEN WAITING; THIS IS TO INSURE THAT A WAITING TERMINAL GETS AT LEAST ONE TBUF WHEN IT COMES TO THE TOP OF THE TBUF WAITING LIST.

26, 27 - DRTIME

DURING A TIMED READ OPERATION, THIS IS THE READING OF THE TIMER AT THE INITIATION OF THE READ. AFTER THE READ IS COMPLETED, THE TOTAL ELAPSED TIME IN 1/100 OF A SECOND IS SAVED INDRTIME AS A SINGLE WORD. IF IT IS -1 THEN THE ELAPSED TIME WAS GREATER THAN 32K.

28 - DRTMAX

WHEN A TIME LIMIT ON A READ OPERATIONIS REQUESTED, THIS QUANTITY REPRESENTS THE MAXIMUM TIME (SECONDS) ALLOWED FOR THE READ OPERATION TO COMPLETE; IF THIS LIMIT IS EXCEEDED, THE READ OPERATION WILL BE TERMINATED.

29 - DSYNC

.LFSYNC (0:4) - CONTAINS THE NUMBER OF SYNC CHARACTERS TO BE SENT AFTER A LF IS OUTPUT

.CRSYNC (4:4) - CONTAINS THE NUMBER OF SYNC CHARACTERS TO BE SENT AFTER A CR IS OUTPUT

.SYNBCCOUNT (8:8) - SPECIFIED THE NUMBER OF DATA CHARACTERS THAT CAN BE TANKED BEFORE AN ENQ HAS TO BE INSERTED IN THE TBUF. FOR WRITE OPERATIONS TO A 264X TERMINAL, AFTER 80 CHARACTERS HAVE BEEN SENT SINCE THE LAST ENQ OR THE LAST READ OPERATION, AN ENQ HAS TO BE SENT AND THE WRITE SUSPENDED UNTIL AN ACK IS RECEIVED.

30 - DBREAK

WHEN A BREAK WAS DETECTED DURING A READ OPERATION, THE DATA ALREADY INPUT IS SAVED AND THIS WORD CONTAINS A POINTER TO AN IOQ USED TO STORE THE BYTE COUNT, TBUF HEAD AND TAIL OF THE SAVED DATA.

31,32 - DTRLS

HOLDS TIMEOUT REQUEST INDICES
.2640TRLX (0:8) - HOLDS THEN INDEX OF A 10 SECOND TIMEOUT REQUEST
                  FOR THE ENQ/ACK HANDSHAKE/BLOCK MODE TIMEOUT
.RREADTRLX (8:8) - HOLDS THE LOGON, HANGUP AND TIMED READ TIME-F
                   OUT REQUEST INDICES
.CFAILTRLX (0:8) - HOLDS THE INDEX OF A TIMEOUT REQUEST DUE TO
                   LOSS OF CARRIER DETECT FROM THE DATASET.
.TURNTRLX (8:8) - HOLDS THE INDEX OF A TIMEOUT REQUEST FOR A LINE
                  TURNAROUND ON A 202 DATASET.

33 - DTANKB

A COUNT OF THE BYTES TANKED IN THE LINKED TBUF'S; THIS COUNT IS USUALLY GREATER THAN DBCNT, THE COUNT OF BYTES REMAINING TO BE OUTPUT, BECAUSE THE DATA IN A TBUF IS SENT OUT IN BLOCKS SEPARATED BY AN ENQ.

34 - DMONTR

.LOGONTYPE (0:2)
.SYNCSTATE (2:2) STATE OF TANKING LF/SYNC
                 0 => TANK XOFF/CR; 1=> DETERMINE LF'S TO TANK
                 2 => TANK LF/SYNC
.CFAILCNT (4:6) - A COUNT OF THE TIMES WHEN LOSS OF CARRIER
                  DETECT FROM THE DATASET IS DETECTED DURING A
                  READ OPERATION; WHEN THE COUNT EXCEEDS 50, THE
                  USER IS HUNG UP AND THE DATASET DISCONNECTED
.LFCOUNT (10:6) - NUMBER OF LF'S FOR %2NN CARRIAGE CONTROL

35 - DSIOPC

STORES THE POINTER TO THE CHANNEL PROGRAM WHICH IS TO BE STARTED WHEN A DATASET LINE TURNAROUND IS COMPLETE; THE CHANNEL PROGRAM TO BE STARTED IS EITHER FOR A READ OR WRITE OPERATION.

36 - DBLKTAIL

POINTER TO THE SECOND TBUF SEF FOR A READ OPERATION; 2 READ CHANNELPROGRAMS, EACH WITH ONE TBUF, ARE USED TO INSURE AGAINST DATA OVERRUNS DURING FAST BLOCK MODE READS.

```
 MULTIPOINT TERMINAL DIT
 ----------------------

 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
 ----------------------------------------------------
|0 |0 |AC|RQ|0 |0 |PM|0 |IA|0 |0 |  0|  STATE       |0 DFLAG
-----------------------------------------------------
|               NEXT DITP                          |1 DLINK
-----------------------------------------------------
|                  IOQP                            |2 DIOQP
-----------------------------------------------------
|      UNIT           |          LDEVNT            |3 DLDEVT
-----------------------------------------------------
|                  DLTP                            |4 DDLTP
-----------------------------------------------------
|                  ILTP                            |5 DILTP
-----------------------------------------------------
|             RESERVED                             |6
-----------------------------------------------------
|             RESERVED                             |7
-----------------------------------------------------
|RT|LG|                  0                         |8 DTIME
-----------------------------------------------------
|GS|RE|CR|FC|MR|WP|RP|DR|UP|PS|RTR|TIM|BR|SSR|FLU|LP|9 DMISCT
-----------------------------------------------------
|LG TY|WA|RJ|DW|DR|UR|EOD|      LDEVNL             |10 DLDEVL
-----------------------------------------------------
|     DSTN    of   terminal   buffer               |11 DDSBUF
-----------------------------------------------------
|     Write   limit   Counter                      |12 DWLIM
-----------------------------------------------------
|     FORMATF            |  Reserved               |13 DFRMAT
------------------------------------------------------
|   Dit Pointer For Next Unit                      |14 DNEXT
-----------------------------------------------------
| Pointer to next Dit with postponed write         |15 DNWRT
-----------------------------------------------------
|LF|DR|BM|AT|SM|WQ|DJST |  STATIONINDIT            |16 DSTA
-----------------------------------------------------
| FIRST WORD FOR ASCII WRITES (if par 1=1)         |17 DFIRST
-----------------------------------------------------
|   ACTUAL BYTE COUNT FOR READS                    |18 DBCNT
-----------------------------------------------------
|   READTINDEXF          |  LOGONTINDEXF           |19 DTIND
-----------------------------------------------------
| READTIME - 1ST WORD OF DOUBLE READTIMER READING  |20 DRTMD
-----------------------------------------------------
| 2nd WORD OF DOUBLE READTIMER READING (start)     |21
-----------------------------------------------------
|         MAXIMUM READ TIME IN SECONDS             |22 DRTMAX
-----------------------------------------------------
```

MULTIPOINT TERMINAL DIT (CONT.)

```
 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
 ----------------------------------------------------
|  TERMINALTYPE      |   0    |   SPEED              |23 DTYPE
----------------------------------------------------
| LOGICAL/PHYSICAL WRITE COUNTER                     |24 DWCNT
----------------------------------------------------
| HOLDS UNEDITED MODE CHARS, WHILE IN BREAK MODE     |25 DBUNM
----------------------------------------------------
| DSTN OF DATA SEGMENT HOLDING "HELLO" MESSAGE       |26 DDSHEL
----------------------------------------------------
|  BYTE COUNT FOR "HELLO MESSAGE"                    |27 DHBCNT
----------------------------------------------------
| POINTER TO NEXT DIT IN WACK Q                      |28 DWACK
----------------------------------------------------
| POINTER TO NEXT DIT IN REJECT Q                    |29 DREJT
----------------------------------------------------
| CURRENT VERSION NO. OF IOMPTRM0 (MODULE 1)         |30 DMOD
----------------------------------------------------
|    ATTENCHAR          |    ENDCHAR                 |31 DUNMD
----------------------------------------------------
|  DSTN OF SECONDARY TERMINAL BUFFER                 |32 DDSB2
----------------------------------------------------
|  BYTE COUNT (READS), BUFFER LENGTH (WACK or reject|33 DBCNT
----------------------------------------------------
|LW|ED|OB|2W|WD         |     GROUPINDIT             |34 DGRP
----------------------------------------------------
```

MULTIPOINT TERMINAL DIT (CONT.)

DFLAG - Flags and SIODM state.

- .ACTIVE - SIODM is currently active servicing this device.
- .REQUEST - Service for this device was requested while SIODM was active.
- .PREMPT - Peemptive request flag.
- .IAK - Response has occured (interrupt acknowledge flag).
- .STATE - SIODM state.

DLINK - SYSDB relative pointer to the DIT for the next device requesting service or this resource.

DIOQP - SYSDB relative pointer to the DIT for the next device requesting service or this resource.

DLDEVT - Logical device number and unit number.

- .LDEVNT - Logical device number of the multipoint terminal.
- .UNIT - Unit number representing terminal address (group and device ID).

DDLTP - SYSDB relative pointer to Driver Linkage Table (DLT).

DILTP - SYSDB relative pointer to dummy Interrupt Linkage Table (ILT) to satisfy SIODM requirements (no reaal ILT is associated with multipoint terminals).

DTIME - Timer flags.

- .READTOF - Read timeout has occurred.
- .LOGONTOF - Log on timeout has occurred.

DMISCT - Miscellaneous flags.

- .GSIN - Last character received from the terminal was the GS character.
- .READEROR - Read error has occurred.
- .CRITICAL - If set, IOMPTRM0 will not attempt to release extra data segments previously acquired by MPMON

MULTIPOINT TERMINAL DIT (CONT.)

.FILTERCRLFOK - Proper editing of input data with respect to CR and LF characters has already been made.

.MARKED - This DITT has already been processed during construction of SUPLIST.

.WPOSTP - Current write request has been postponed.

.READPEND - Read request is pending against this terminal.

.DATAREADY - Input data has been received and is ready in the terminal read buffer.

.UP - Device has been initialized through the log on procedure or has been allocated.

.PRESPACEF - Last write operation was with a prespace request. If the next write operation is with a post space request, output CR and LF before data.

.READTIMERF - Read timing requested and not yet in progress.

.TIMING - Current read request is being timed.

.BRKOK - System break is enabled.

.SSBRKOK - Subsystem break is enabled.

.FLUSH - This flag i set whenever break has been detected and accepted. While it is set, writes are returned completed without any I/O being done. Reads are returned with an unusual condition status %173. It also holds off any further break service requests. It is reset with a function code 25 operation.

.LASTPREMEPT - Last request was a preemptive request.

DLDEVL

. LOGONTYPE - 0: JOB
1: SESSION
2: DATA

. WACK - If set then WACK or EOT condition has been detected and the terminal was placed in the WACK queue.

. REJECT - If set then a terminal error has been detected and the terminal was placed in the REJECT queue.

. DOWN - If set then this terminal was declared down through the console operator command or the configuration file.

. DOWNREQ - If set then a request is pending to declare the terminal down.

. UPREQ - If set then a request is pending to declare the terminal up.

.ECHO'OFF'D - For 3270 terminals, set true if no echo print to the terminal is wanted.

. LDEVNL - Logical device number of the controller servicing the multipoint line.

DDSBUF - Data segment number of the terminal read buffer.

DWLIM - Write limit counter.

DFORMAT

.FORMATF - This field holds information about vertical format specification for writes obtained from P1 parameter of the IOQ element or from the first data byte.

DNEXT - SYSDB relative pointer to the DITT for the next terminal on the same line.

DNWRITE - SYSDB relative pointer to the DITT for the next terminal with postponed write.

DSTATION - Flags and station number.

.LFLUSH - This flag is set to indicate that data for this terminal already scheduled to be written from the output buffer should not be physically sent to the terminal (break or subsystem break environment).

MULTIPOINT TERMINAL DIT (CONT.)

.DISCONREQ - Request to disconnect the terminal.

.BREAKMODE - Terminal is in break mode.

.ATTENTERM - Terminal is in attention mode.

.SSBMODE - Terminal is in subsystem break mode.

.WLQUEUE - A write request was forced to be queued by MPE I/O system.

.DJSTATE - State of terminal straps D and J.
- 0 - Initial state.
- 1 - Straps D and J are open or will be open before the next write.
- 2 - Undefined D and J setting.

.STATIONINDIT - Station number assigned to this terminal by CS.

DFIRST - Storage for first word for ASCII writes if vertical format is specified by first data byte.

DBCNT - Actual byte count for reads.

DTIND - Timer indexes.

.READTINDEXF - Read timer index.

.LOGONTINDEXF - Log on timer index.

DRTIME (DRTIMED) - During a timed read, this is the reading of the timer at the initiation of the read. After a timed read is completed, the time in 1/100 of a second is saved in DRTIME as a single word. If it is -1 then the time was greater than 32K.

DRTMAX - When a read operation timeout is requested, this quantity represents the maximum time in seconds allowed for the read to be completed.

MULTIPOINT TERMINAL DIT (CONT.)

DTYPE - Terminal type and speed.

.TERMINALTYPE - Configured terminal type. Multipoint terminal is type 14.

.SPEED - Reserved field for configured terminal speed (not used for multipoint terminals).

DWCNT - Logical/physical write counter.

DBUNMODE - Holds unedited mode characters while in break mode.

DDSHEL - DST number of data segment holding "HELLO" message (or backspaced data).

DHBCNT - Byte count for "HELLO" message (or backspaced data).

DWACK - Pointer to next DIT in WACK queue.

DREJECT - Pointer to next DIT in REJECT queue.

DMOD1VER - Current version number of the multipoint terminal driver (IOMPTRMO).

DUNMODE - Unedited mode characters.

.ATTENCHAR - Attention character.

.ENDCHAR - End-of-character. (Effective as a control character is set to %137, otherwise not used).

DDSBUF2 - Data segment number of secondary read buffer.

DBCNT2 - Byte count for read if secondary read buffer is used

DGROUP

.L'WRITE'D - Set true if last I/O request was a write.

.EOS'D - Set true if a write to a 3270 terminal reaches end of screen.

MULTIPOINT TERMINAL DIT (CONT.)

```
  .ODD'BYTE'3270 - Set true if there is an odd number of bytes in
                   a write.

  .WRITEPEND  - Reserved.

  .ZERO'WRITE - Set true if no byte is transmitted in a write
                because of an error other than a conversation write.

  .WRITEDONE  - Set true after a write issuded is completed.
      GROUPINDIT - Logical group number assigned to this terminal
                   by CS.
```

MULTIPOINT SUPERVISOR DIT

```
 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
-------------------------------------------------------
|0   |0  |AC|RQ|0  |0  |PR|0  |IA|0  |0   |0   | STATE        |0 DFLAG
------------------------------------------------------
|                NEXT     DITP                   |1 DLINK
------------------------------------------------------
|                    IOQP                        |2 DIOQP
------------------------------------------------------
|       UNIT             |           LDEVNS      |3 DLDEVS
------------------------------------------------------
|                    DLTP                        |4 DDLTP
------------------------------------------------------
|                    ILTP                        |5 DILTP
------------------------------------------------------
|                  RESERVED                      |6
------------------------------------------------------
|                  RESERVED                      |7
------------------------------------------------------
|WA|RJ|                   0                      |8 DTIME
------------------------------------------------------
|MP|DU|DE|TO|TOR|TR|SN|SR|BH|MA|MU |GP |GD| GW |GR|CR |9 DMISCS
------------------------------------------------------
|      RESERVED           |           LDEVNL      |10 DLDEVL
------------------------------------------------------
|  DIT POINTER FOR MP SUPERVISOR                 |11 DDITSP
------------------------------------------------------
|  OFFSET TO TRACE BUFFER IN MPMON STACK         |12 DTBOFF
------------------------------------------------------
|       WRITE LIMIT CONSTANT                     |13 DWLCON
------------------------------------------------------
|     DIT POINTER FOR FIRST UNIT                 |14 DNEXT
------------------------------------------------------
| POINTER TO FIRST DITT WITH POSTPONED WRITE     |15 DNWRIT
------------------------------------------------------
| CURRENT VERSION NO. OF IOMPS0 (MODULE 2)       |16 DMOD2V
------------------------------------------------------
| ADDRESS OF LINE READ BUFFER IN MPMON STACK     |17 DINBA
------------------------------------------------------
| ADDRESS OF LINE WRITE BUFFER IN MPMON STACK    |18 DOUTBA
------------------------------------------------------
|              OUTPUT SPEED                      |19 DOSPD
------------------------------------------------------
```

MULTIPOINT SUPERVISOR DIT (CONT.)

```
  0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
-------------------------------------------------------
| INDEX OF HEAD ENTRY IN LINE WRITE BUFFER            |20 DHEADI
-------------------------------------------------------
| INDEX OF TAIL ENTRY IN LINE WRITE BUFFER            |21 DTAILI
-------------------------------------------------------
| INDEX OF LAST AVAILABLE WORD IN LINE WRITE BUFFER   |22 DENDI
-------------------------------------------------------
|    TERMINAL TYPE   |SP|DO|ID|       SPEED           |23 DTYPE
-------------------------------------------------------
| CURRENT VERSION NO. OF MPMONCMD (Module 3)          |24 DMOD3V
-------------------------------------------------------
|  DSTN OF MPMON  STACK                               |25 DMDSTN
-------------------------------------------------------
|  LINE  SPEED  - 1st WORD                            |26
   DLSPEED
-------------------------------------------------------
|  LINE SPEED  - 2nd     WORD                         |27
-------------------------------------------------------
|  POINTER TO FIRST DIT IN WACK Q                     |28 DWACK
-------------------------------------------------------
|  POINTER TO FIRST DIT IN REJECT Q                   |29 DREJ
-------------------------------------------------------
|     WACKTINDEX         |      REJECTTINDEX          |30 DWRTI
-------------------------------------------------------
|     CFCHAR0            |       CFCHAR1              |31 DCF01
-------------------------------------------------------
|     CFCHAR2            |       CFCHAR3              |32 DCF23
-------------------------------------------------------
|     CFCHAR 4           |       CFCHAR5              |33 DCF45
-------------------------------------------------------
|     CFCHAR6            |       CFCHAR7              |34 DCF67
-------------------------------------------------------
|RO|UDR|FB|CD|FS|DI|      0          |DUS|MM| UD|MO|35
-------------------------------------------------------
```

MULTIPOINT SUPERVISOR DIT (CONT.)

DFLAG -
DLINK - Same as for DITT
DIOQP -

DLDEVS - Logical device number and unit number.

.LDEVNS - Logical device number of the Multipoint Supervisor.

.UNIT - Unit number (always 0).

DDLTP - Same as for DITT
DILTP

DTIME -
.WACKTO If set, then WACK timeout has expired
.REJECTIO If set, then REJECT timeout expired.

DMISCS - Miscellaneous flags.

.MPOK - If set, then IOMPSO is allowed to process I/O requests against the Multipoint Supervisor.

.DUPLEX - Reserved.

.DEBUGON - If set, then DEBUG will be called from MPMON. This flag is set through the MPLINE command.

.TRACEON - Trace facility is enabled.

.TRACEOFFREQ - Trace facility is to be disabled.

.TRACEONREQ - Trace facility is to be enabled.

.SHUTNOW - Request to shut the line immediately.

.SHUTREQ - Request to shut line after all terminals are released
New sessions are not allowed to be initiated.

.BUSYHEAD - The line write buffer contains data to be written to a terminal on the line.

MULTIPOINT SUPERVISOR DIT (CONT.)

.MPONACT - MPMON process is active.

.MPMONUP - MPMON process has been created and activated.

.GENWPOSTP - A write request for one or more terminals on the line has been postponed.

.GENDISCON - Request to disconnect the line.

.GENWACK - If set then there is a terminal in the WACK queue.

.GENREJECT - If set then there is a terminal in the REJECT queue.

.COMPLREQ - Request to complete dummy read pending against the Multipoint Supervisor.

DLDEVL

.LDEVNL - Logical device number of the controller servicing the multipoint line.

DDITSP - SYSDB relative pointer to the DIT for the Multipoint Supervisor (DITS).

DTBUFOFFS - Offset to the trace buffer in MPMON stack.

DWLCON - Write limit constant.

DNEXT - SYSDB relative pointer to the DITT for the first terminal on the line (the terminal with the lowest logical device number).

DNWRITE - SYSDB relative pointer to the DITT for the first terminal with postpond write.

DMOD2VER - Current version number of the Multipoint Supervisor driver (IOMPSO).

DINBUFA - Address of the line read buffer in MPMON stack.

DOUTBUFA - Address of the line write buffer in MPMON stack.

DOSPEED - Output speed.

DHEADI - Index of head entry in the line write buffer.

DTAILI - Index of tail entry in the line write buffer.

DENDI - Index of last available word in the line write buffer.

DTYPE

.TERMINALTYPE - Configured terminal type. Multipoint Supervisor is type 14 (same type as multipoint terminals).

.SUPER - This device is a Multipoint Supervisor.

.DITSOK - DIT's for the multipoint terminals and the Multipoint Supervisor on this line have been rearranged and their format corresponds to standard DIT format for SIO devices.

.SPEED - Reserved field for configured terminal speed (not used for Multipoint Supervisor).

.INITDONE - If set then all multipoint terminals belonging to the same multipoint supervisor have been linked.

DMOD3VER - Current version number of the MPLINE command processor (MPMONCMD).

DMONDSTN - Data segment number of MPMON stack.

DLSPEED (DLSPEEDD) - If not equal to 0, then the line is opened with speed specified in this double word.

DWACK - Pointer to the first terminal DIT in the WACK queue.

DREJECT - Pointer to the first terminal DIT in the REJECT queue.

DWRT1
.WACKTINDEX - WACK timer index.
.REJECTTINDEX - REJECT timer index.

DCF01 through DCF67 - String of characters representing:
a) the name of the configuration file, or
b) the logical device number of the terminal, or
c) terminal group and device ID.

DCONFL
.REOPEN - If set then a request for line reopening has been made.

.UPDOWNREQ - If set then a request to set the terminal UP or DOWN has been made.

.FALLBACK - Reserved.

.CHDUPL - Reserved.

.FORCE'SHUT - If set then a request has been made to shut the line immediately.

.DUMP'INP - Reserved.

.DUPLEX'SPEC - Reserved.

.MON'MODE - Reserved.

.UP'DOWN - If true then the terminal is to be set UP else the terminal is to be set DOWN. This flag is used in conjunction with .UPDOWNREQ flag.

.MSGOFF - If set then certain MTS messages are not displayed on the operator console.

CHAPTER 22 DISC FREE SPACE MAP

22.1) Disc Resident Data Structures

There are two disc resident free space data structures, the bit map and the descriptor table, for each disc volume that has a free space map, i.e. system discs and private volumes. The addresses of these data structures are kept in the disc label. The symbols that define the descriptor table and bit map are in the include file INCLDFS2.

22.1.1) Bit map

The bit map is divided up into pages, which is the physical block of the map that is read or written. At the moment, a page is defined to be one sector (128 words) long, this may be changed by changing a compile time constant. The last word of the page is a checksum for that page, all other words are data. There is a one to one correspondence between bits in the map and sectors of the disc. A one bit represents a free sector and a zero bit represents an allocated sector. The bit map is a contiguous set of pages, enough to represent the entire disc, excluding spare tracks and spare sectors.

22.1.2) Descriptor table (DT)

The descriptor table is an array of three word entries, one entry for each page of the bit map. Each entry looks like this:

```
                ====================
                =                  =
     word 0     =  largest space   =
                =                  =
                ====================
                =                  =
     word 1     =  starting space  =
                =                  =
                ====================
                =                  =
     word 2     =  ending space    =
                =                  =
                ====================
```

Thus the descriptor table looks like this:

```
   ------------
   =          =  entry for page 0
   ------------
   =          =  entry for page 1
   ------------
   =          =  entry for page 2
   ------------
   =          =  entry for page 3
   ------------
        .
        .
        .
   ------------
   =          =  entry for last page
   ------------
```

Each entry describes the free space on the corresponding page of the bit map. The largest space word is the size of the largest contiguous block of free space on the page, which is not at the very beginning or very end of the page. That is, the first bit physically representing the space is not the first bit of data on the page or the last bit representing the space is not the last bit of data on the page. Starting space is the number of sectors of contiguous space represented by the set of bits whose first bit is the first bit of data on the page. Ending space is the number of sectors of contiguous space represented by the set of bits whose last bit is the last bit of data on the page. The starting space and ending space fields allow looking across page boundries, thus preventing fragmentation on page boundries. Thus, if all sectors represented on a page are free, then starting and ending space will be the same and have the total number of free sectors represented on the page. Largest space will be zero, as there is no block of space that is not at the beginning or end of the page. A value of -1 for all the fields in an entry indicates the corresponding page is bad, either from a checksum or I/O error.

22.2) Virtual Memory Resident Data Structures

For each system disc or physically mounted private volume there is a data segment which has information about the disc free space map, the current copy of the descriptor table, some work space for the procedures while in spilt stack mode and buffers for pages of the bitmap. The DST number of the data segment for a given disc is found in the LDTX entry for that disc.

22.2.1) Disc Free Space Data Segment

For each system disc or physically mounted private volume in the up and running system there is a DST which contains info about the disc free space map for that disc, some work area, a copy of the descriptor table and buffers for the pages of the bit map. All symbols that define these data segments are in the include file INCLDFS1, and they are prefixed with "ds'". The structure of the data segment is as follows:

```
                 ===================================
      0 (%0)     =              ds'ldev            =
                 =---------------------------------=
      1 (%1)     =              ds'dst             =
                 =---------------------------------=
      2 (%2)     =                                 =
                 =--------- ds'disc'size ----------=
      3 (%3)     =                                 =
                 =---------------------------------=
      4 (%4)     =        ds'last'page'of'map      =
                 =---------------------------------=
      5 (%5)     =       ds'last'buffer'index      =
                 =---------------------------------=
      6 (%6)     =                                 =
                 =-------- ds'map'address ---------=
      7 (%7)     =                                 =
                 =---------------------------------=
      8 (%10)    =              ds'lock            =
                 =---------------------------------=
      9 (%11)    =           ds'lock'count         =
                 =---------------------------------=
     10 (%12)    =           ds'queue'head         =
                 =---------------------------------=
     11 (%13)    =           ds'queue'tail         =
                 =---------------------------------=
     12 (%14)    =        ds'descriptor'table      =
                 =---------------------------------=
     13 (%15)    =       ds'buffer'page'number     =
                 =---------------------------------=
     14 (%16)    =          ds'buffer'dirty        =
                 =---------------------------------=
     15 (%17)    =          ds'buffer'area         =
                 =---------------------------------=
     16 (%18)    =     ds'first'threshold'page     =
                 =---------------------------------=
     17 (%21)    =                                 =
                 =-- ds'size'of'last'allocation  --=
     18 (%22)    =                                 =
                 =---------------------------------=
                 =---------------------------------=
     19 (%23)    =   ds'last'page'allocated'from   =
```

```
                 =---------------------------------=
     20 (%24)    =      ds'next'buffer'index       =
                 =---------------------------------=
     21 (%25)    =         ds'page'number          =
                 =---------------------------------=
     22 (%26)    =         ds'word'number          =
                 =---------------------------------=
     23 (%27)    =          ds'bit'number          =
                 =---------------------------------=
     24 (%30)    =         ds'page'pointer         =
                 =---------------------------------=
     25 (%31)    =    ds'starting'word'number      =
                 =---------------------------------=
     26 (%32)    =    ds'starting'bit'number       =
                 =---------------------------------=
     27 (%33)    =                                 =
                 =----- ds'number'of'sectors  -----=
     28 (%34)    =                                 =
                 =---------------------------------=
     29 (%35)    =          ds'bit'count           =
                 =---------------------------------=
     30 (%36)    =          ds'entry'type          =
                 =---------------------------------=
     31 (%37)    =         ds'buffer'index         =
                 =---------------------------------=
     32 (%40)    =                                 =
                 =-------- ds'disc'address --------=
     33 (%41)    =                                 =
                 =---------------------------------=
     34 (%42)    =         ds'error'status         =
                 =---------------------------------=
```

The rest of the data segment contains tables whose size and location is dependent on the size of the disc and or the number of buffers in the data segment. They are shown below just to demonstarate there relation to one another, for there actual location, the pointers should be examined. The symbol "ds'array'area" defines the start of the area.

The first table is the descriptor table, it is in the same format as the disc copy, but a dummy entry of all zeros is added before and after the table, these are needed by procedures "Find'Page" and "Build'Descriptor'Entry". The pointer to this table is "ds'descriptor'table", it points to the entry for page zero, not the dummy entry.

```
                 ===================================
                 =                0                =
                 =---------------------------------=   dummy
                 =                0                =
                 =---------------------------------=   entry
```

```
=                 0                 =
=====================================
=           largest space           =
=-----------------------------------=  entry for
=          starting space           =
=-----------------------------------=    page 0
=           ending space            =
=====================================
=           largest space           =
=-----------------------------------=  entry for
=          starting space           =
=-----------------------------------=    page 1
=           ending space            =
=====================================
                  :
                  :
                  :
=====================================
=           largest space           =
=-----------------------------------=  entry for
=          starting space           =
=-----------------------------------=  last page
=           ending space            =
=====================================
=                 0                 =
=-----------------------------------=    dummy
=                 0                 =
=-----------------------------------=    entry
=                 0                 =
=====================================
```

The next table is ds'buffer'page'number table, it has a one word entry for each buffer in the data segment. Each entry contains the page number of the page currently in the corresponding buffer or -1 if the buffer is empty. This is pointed to by "ds'buffer'page'number".

```
=====================================
=          buffer 0 entry           =
=====================================
=          buffer 1 entry           =
=====================================
                  :
                  :
                  :
=====================================
=         last buffer entry         =
=====================================
```

The next table is the ds'buffer'dirty table, which has a one word entry for each buffer. A TRUE indicates the page in the correspnding buffer is dirty, i.e. the disc copy is not uptodate. A FALSE indicates that the buffer is clean. If DFS was compiled with dirty buffer management turned off, this table is not present and the ds'buffer'dirty pointer is zero.

```
====================================
=         buffer 0 entry           =
====================================
=         buffer 1 entry           =
====================================
                  :
                  :
                  :
====================================
=        last buffer entry         =
====================================
```

The remainder of the data segment contains the buffers, each buffer is the size of one page of the bit map, which is currently one sector (128 words). The beginning of the buffer area is pointed to by "ds'buffer'area" and the number of buffers is the value in "ds'last'buffer'index" plus one.

```
====================================
=                                  =
=                                  =
=                                  =
=            buffer 0              =
=                                  =
=                                  =
=                                  =
====================================
=                                  =
=                                  =
=                                  =
=            buffer 1              =
=                                  =
=                                  =
=                                  =
====================================
                  :
                  :
                  :
====================================
=                                  =
=                                  =
=                                  =
=           last buffer            =
```

```
             =                                 =
             =                                 =
             =                                 =
             ===================================
```

Each of the fields of the data segment is described in the include file INCLDFS1, where they are defined. It should be noted that the following fields are just workspace, used to pass information between procedures while in spilt stack mode and have no meaning between calls to the disc free space management subsystem:

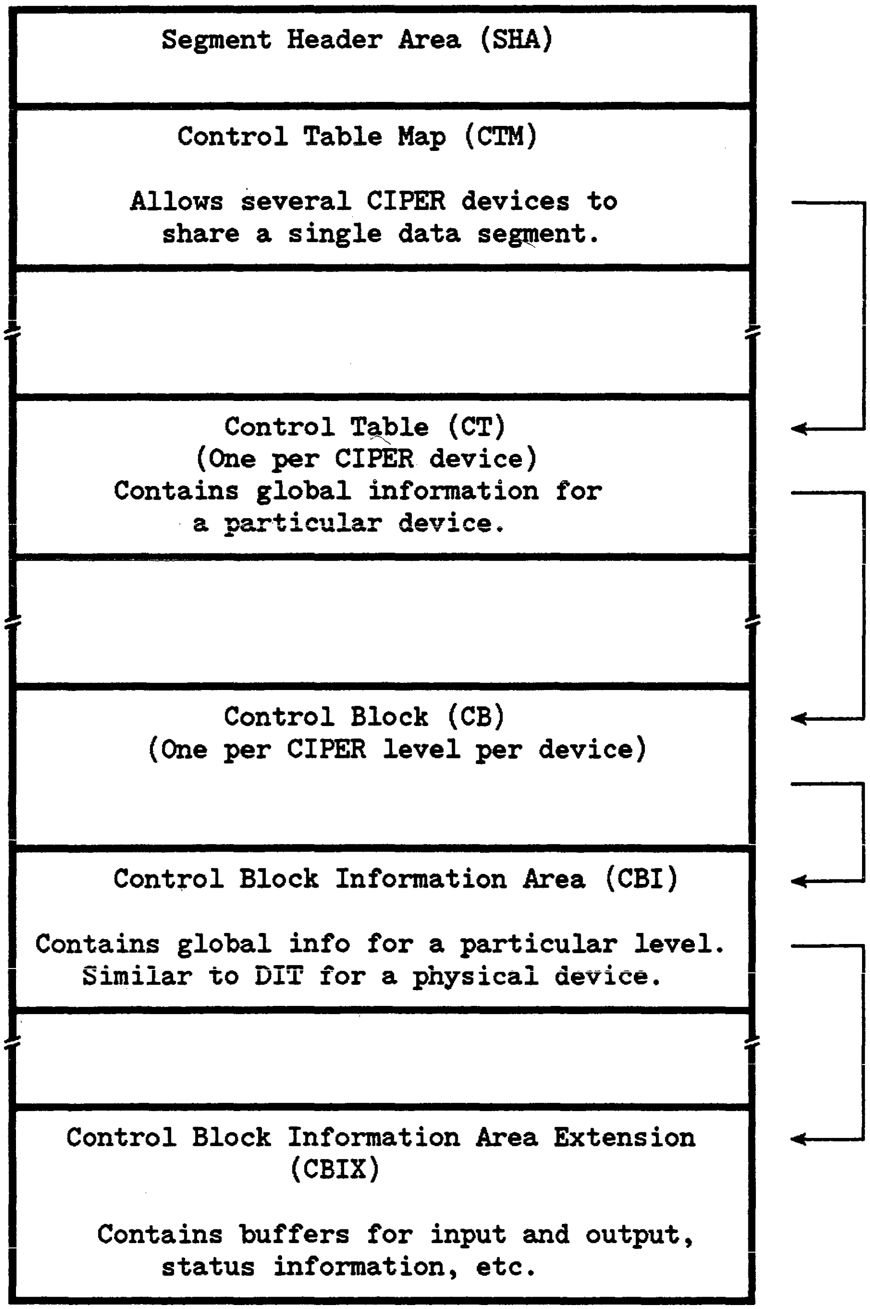
ds'page'number
ds'bit'number
ds'starting'word'number
ds'number'of'sectors
ds'bit'count
ds'disc'address
ds'word'number
ds'page'ptr
ds'starting'bit'number
ds'entry'type
ds'buffer'index

The field ds'error'status normally has no meaning between calls unless the error'type field has a value greater than "fatal'dfs'error", in which case it means that disc space may nolonger be allocated on this disc.

CHAPTER 23 CIPER TABLES

CIPER Data Segment (CDS) Overview

The CIPER data segment (CDS) is the primary data structure accessed by SOFTIO. The general format of the segment is illustrated below. The following data structures are expansions of the general format and are self explanatory within each structure detailed.



Segment Header Area (SHA)

The SHA is the first data structure encountered within the data segment. There is only one Segment Header Area per segment.

Word	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
0	SHA'FREE'SPACE'TBL'PTR
1	SHA'CDS'DST'NUM
2	SHA'MAX'SEG'SIZE
3	SHA'SEG'SIZE
4	SHA'CTM'PTR
5	SHA'LIOQ'LIST'PTR

Discussion:

SHA'FREE'SPAC'TBL'PTR - Data segment base relative address of the upper stop boundary of the dynamically managed memory area.

SHA'CDS'DST'NUM - The number of this data segment.

SHA'MAX'SEG'SIZE - The maximum size this data segment is configured for in virtual memory (ie, maximum possible size).

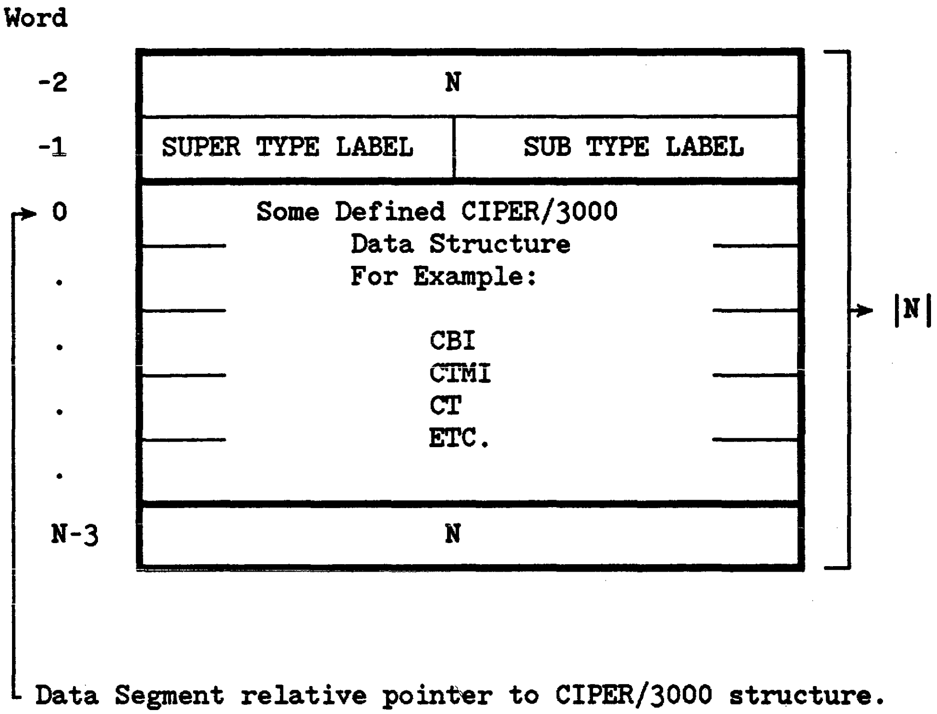
SHA'SEG'SIZE - The current size of this data segment (either main memory or disc).

SHA'CTM'PTR - Data segment base relative address of the Control Table Map.

SHA'LIOQ'LIST'PTR - Data segment base relative address of the Logical IO Queue list (not used at this time).

Memory Allocation Manager Typical Layout

The SOFTIO module contains a Memory Manager which manages all data structures within the CIPER/3000 data segment (CDS). The structures used are ambles (Pre & Post). The Preamble is two words in length, while the postamble is 1 word in length. The MAM preamble and postamble surround each portion of the CDS allocated.



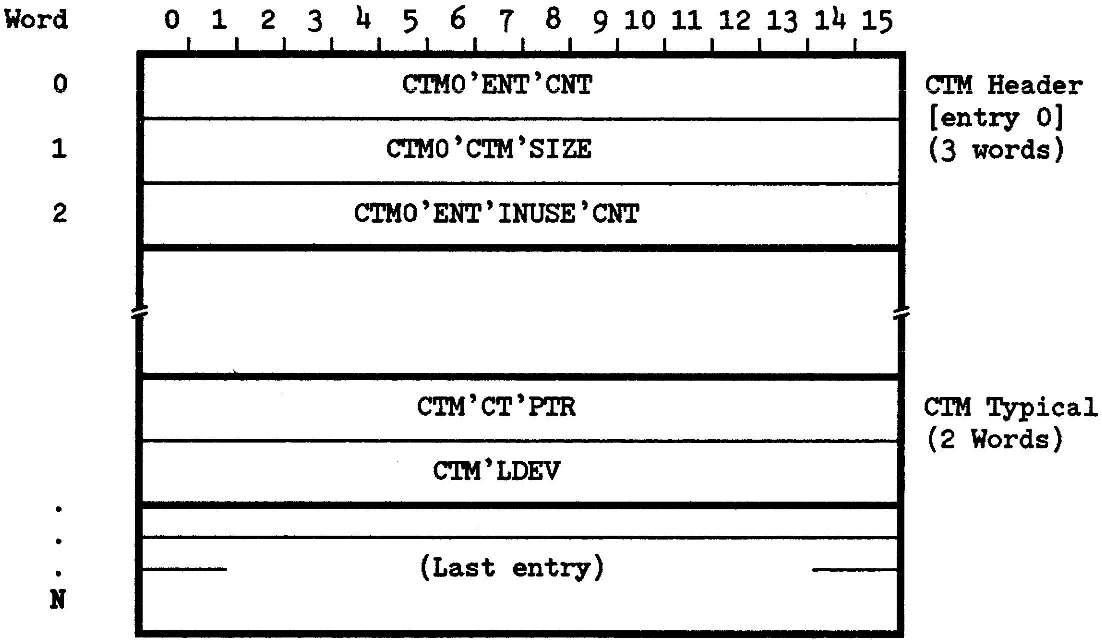
Data Segment relative pointer to CIPER/3000 structure.

NOTE: If N is less than or equal to zero then the area is currently deallocated.

If N is greater than zero then the area is currently allocated.

Control Table Map (CTM)

This table is a series of entries, one for each logical device. The LDTX contains an index to the logical device Control Table Map. There is only one logical device per data segment. Therefore, the CTM is comprized of only one entry.



Discussion:

CTMO'ENT'CNT - The number of entries in the CTM (not counting the header entry).

CTMO'CTM'SIZE - The size of each entry in the CTM (disregarding the size of entry 0, the size of each entry is currently 2).

CTMO'ENT'INUSE'CNT - The number of entries in the CTM (not counting the head entry) currently in use.

CTM'CT'PTR - Data segment base relative address of the Control Table (CT).

CTM'LDEV - The logical device number for which this entry is associated.

Control Table (CT)

The Control Table Map points to the Control Table where specific device, caller, and level control information is stored.

Word	
0	CT’SIR
1	CT’SIR’SAVE
2	CT’CDS’DST’NUM
3	CT’CTMI
4	CT’MSW’CALLERS’DB
	(CT’D’CALLERS’DB)
5	CT’LSW’CALLERS’DB
6	CT’CALLERS’STK
7	CT’CALLERS’STK’DB
8	CT’LVL’CNT
9	CT’LVL’ACTIVE
10	CT’LVL’ACTIVE’PTR
11	CT’VDT’PTR
.	CT’LVL1’CB’PTR
.	CT’LVL2’CB’PTR
.	CT’LVL3’CB’PTR
.	CT’LVL4’CB’PTR
.	CT’LVL5’CB’PTR
.	CT’LVL6’CB’PTR
N	CT’LVL7’CB’PTR

Discussion:

CT'SIR - Not currently used.

CT'SIR'SAVE - Not currently used.

CT'CDS'DST'NUM - The number of this data segment.

CT'CTMI - The control table map index used to reach this control table.

CT'D'CALLERS'DB - The result of the call to CHANGEDB which moved DB to the CIPER data segment (CDS). It is used to return DB to the same spot the caller had it at.

CT'CALLERS'STK - The dst number of the calling processes' stack.

CT'CALLERS'STK'DB - The offset from the data segment base in the calling processes' DB.

CT'LVL'CNT - The number of levels currently loaded into this control table.

CT'LVL'ACTIVE - The level which is currently within this control table.

CT'LVL1'CB'PTR - Not currently used.

CT'LVL2'CB'PTR - Not currently used.

CT'LVL3'CB'PTR - Not currently used.

CT'LVL4'CB'PTR - The pointer to the control block of level four (network protocol).

CT'LVL5'CB'PTR - Not currently used.

CT'LVL6'CB'PTR - Not currently used.

CT'LVL7'CB'PTR - The pointer to the control block of level seven (the logical driver).

CIPER Level 'N' Control Block

For every level that exists in the CIPER Protocal model, there is a control block which contains specific control information for the level at which the operation is being accomplished. This implementation of CIPER contains level 7, 6, 4, 2, and 1. Therefore, it would seem to follow that there are seven control blocks. However, that is a false assumption. Level 7 is really the user interface in which MPE is the file system, SPOOLER, ATTACHIO and the Logical Driver. Thus, level 7 resides only partially in SOFTIO. Level 6 is the CIPER translator (procedure CPR'XLATOR). In this implementation, it is not a user-callable intrinsic and hence does not need a CB of its own. However, should it ever become a user-callable intrinsic, it will then require a level 'N' CB. Levels 7 and 4 do require control blocks since they use the data segment extensively. Levels 5 and 3 do not exist and do not currently need space. Levels 2 and 1 refer specifically to the physical driver. In the case of the HBIB driver, the DIT and IOQ hold the information that would ordinarily be in the Level 'N' CB. Thus, the HPIB driver does not need this structure. For the Multi-Point Terminal System (MTS), the process and physical driver do not require space in the data segment for control information. However, MTS does access the CIPER data segment for the data being read or written from/to the device.

In summary: there are two level 'N' CBs for this implementation. There is one for Level 7 and one for Level 4. The format for the control block is as follows:

Word		
0	CB'PLABEL	→ CB'SIZE
1	CB'QH'PTR	
2	CB'CBI'PTR (CB'INFO'PTR)	

Discussion:

CB'PLABEL - Control Block Program label. The PLABEL of the module which will be called for this level. Allows multiple modules for any level. Not currently used.

CB'QH'PTR - Control Block Queue Head pointer.
Data segment relative pointer to the Communication Queue Head.

CB'INFO'PTR - Data segment base relative address to the control block information area. This information area is level dependent. The information within the 'INFO' block pointed to by CB'INFO'PTR contains variable length information which only the level module called 'knows' about.

CB'SIZE - currently the CB size is set to 3 words.

NOTE: Since there are only two levels (level 4 and level 7), there are only two level 'N' control blocks and two 'INFO' areas at this time.

Communication Queue Head

Communication Queue Heads are used for passing internal messages within CIPER. CIPER runs on the caller's stack. The Queue Head mechanism is useful for passing messages between procedures at the same level and to the level above and below the current level. This helps to synchronize all of the events occuring within CIPER. The level 'N' CB contains a data segment relative pointer to the queue head. The queue head mechanism is logically similar to the message harbor table mechanism in the MPE internal message system.

When a message is to be passed by some procedure at some level, it merely calls one of the T'LINK'XXX procedures to do so. The memory manager acquires chunks of free memory in the data segment to hold the data. The pointers in the following table are data segment relative pointers to that chunk (or those chunks) of memory that are reserved for the message data.

Head'entry
Tail'entry
Entry'size
Free'list'ptr
Free'count
In'use'count
Max'used'count
Back'pointer

Discussion:

Head'entry: this is a pointer to the first entry in the queue. Items are typically removed from the head.

Tail'entry: this is a pointer to the last entry in the queue.

Entry'size: specifies the queue entry size of all entries in this particular queue. Size is in words.

Free'list'ptr: pointer to a queue of available entries. Elements are added and removed from the head.

Free'count: the number of queue elements available in the freelist.

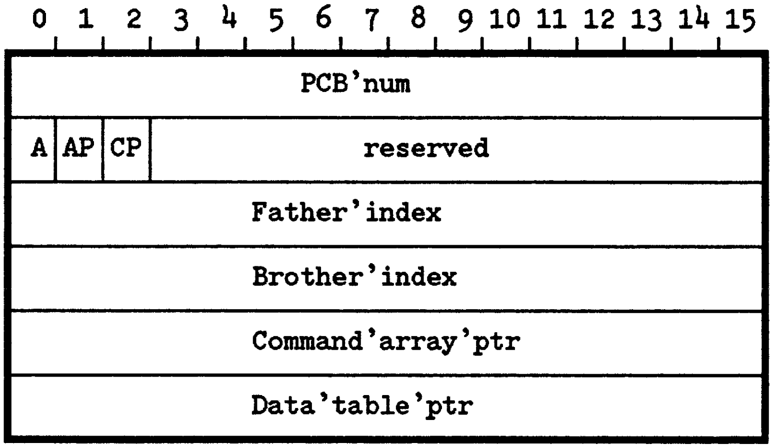
In'use'count: the number of queue elements currently linked in the request queue.

Max'used'count: a high-water mark which tallies the maximum value that In'use'count ever assumes.

Back'pointer: an optional backward reference pointer to the request queue. Could be used for a two-way linked list.

In general, a given level (n) has four request queues associated with it. There is a command queue from the level above (n+1), and a response queue back to that level. There is also a command queue to the next lower level (n-1), and a response queue from that level.

COMQUELEMENT (Communications Queue Element)



Discussion:

PCB'num: the Process Control Block number of the process issuing the request. Maintained at all levels to facilitate ABORTPROCIO requests.

A: abort bit. Set in responce to an ABORTIO(LDEV) command. Allows Ciper'IO'Process to clean up the request as soon as possible.

AP: process abort bit. Set in responce to an ABORTPROCIO command. Allows Ciper'IO'Process to clean up the request when it is convenient for it to do so.

CP: Ciper'IO'Process flag, which is set if CIP issued this particular request, either on its own behalf or for a user process.

Father'index: a request queue pointer to level (n+1)'s queue element that caused this request to be generated.

Brother'index: a request queue pointer to a level n queue element that is related to this element, by virtue of having the same level (n+1) queue element generating the request.

Command'array'ptr: a pointer to a command array specified by the calling level. Contents of the command array depend on the interface established between levels. The first word will always be the request flags passed from the user (or CIP) which must be maintained to the lowest level.

Data'table'ptr: a pointer to the virtual data table associated with this request.

Control Block Information Area

For every level 'N' Control Block, there is a Control Block Information Area (CBI). In this implementation there are 2 CBI's since there is a level 7 and a level 4 Control Block. The CBI is a variable length extension to the CB. It can be different lengths for different levels. The level 7 and level 4 CBI7s are outlined below.

LEVEL 7 CONTROL BLOCK INFORMATION (CBI)

Word	
0	CDS'AREA'BASE
1	INITIALIZED
2	JOB'ACTIVE
3	FREE'BUFF'LIST
4	O'R'BASE
5	I'R'BASE
6	DEV'STATUS'BASE
7	COMPOSITE'STATUS'BASE
8	ENV'STATUS'BASE
9	JOB'REPORT'BASE
10	EXPANDED'FEATURES
11	INPUT'SEQUENCE'COUNT
12	OUTPUT'SEQUENCE'COUNT
13	RECEIVE'READY'COUNT
14 15	CPR'XLATOR FLAGS
16	SEQUENCE'1'BUFFER
17	O'R'DATA'TYPE
18	I'R'DATA'TYPE
19	FILE'OPEN'COUNT

LEVEL 7 CONTROL BLOCK INFORMATION (CBI) CONT.

20	DEVICE'ALLOCATED
21	LOGICAL'DEVICE
22	CIPER'DST
23	OUT'RECS'OVERWRITTEN
24	IN'RECS'OVERWRITTEN
25	DEVICE'BUFFER'SIZE
26	DEVICE'ENV'STATUS'SIZE
27	PRODUCT'NUMBER
28	STORAGE'REQUIREMENTS
29	TEMP'AREA
30	CT'PTR
31	PACKET'HEADER'SIZE
32	PACKET'TRAILER'SIZE
33	PACKET'SIZE
34	DEV'CLR'COUNT
35	DEV'CLR'IN'PROGRESS
36	SR'ENABLE
37	ESB'FREQUENCY
38	LOGGING'DST
39	LOGGING'BUFFER
40	EVENT'MAP
41	STATUS'ENABLED
42	STATUS'RECEIVED
43	STATUS'REPORTED

44	DEFAULT'ACCESS'MODE
45	COMP'STAT'AVAILABLE

Discussion:

CDS'AREA'BASE - contains the CDS relative address of the Control Block Information eXtension (CBIX). The CBIX is the area that contains all record buffer, status tanks, and other arrays used by the logical driver.

INITIALIZED - a logical flag set to true when the entire CBIX and all other information areas have been completely initialized.

JOB'ACTIVE - a logical flag set to true when a command has been sent to start a job, and that command has been passed to the device.

FREE'BUFF'LIST - a word pointer which contains the CBIX relative address of the first entry in a linked list of record buffer areas that are currently not in use.

O'R'BASE - a word pointer which contains the CBIX relative address of the base of the output record buffer area. This buffer is normally contained within the region pointed to by cds'area'base, but it does not have to be.

I'R'BASE - a word pointer which contains the CBIX relative address of the base of the input record buffer area. Like the output record buffer area, this is typically contained within the region pointed to by cds'area'base.

COMPOSITE'STATUS'BASE - word pointer to the area that contains composite status. Composite status is the logical "OR" of any device status reports that are received during any one call to the logical driver.

DEV'STATUS'BASE - a word pointer which contains the CBIX relative address of the base of the device status buffer area, which is used to store incoming device status reports.

ENV'STATUS'BASE - a word pointer which contains the CBIX relative address of the base of the environmental status buffer, which is used to store incoming device environmental status reports.

JOB'REPORT'BASE - a word pointer which contains the CBIX relative address of the base of the job report buffer area, which is used to store incoming job reports.

EXPANDED'FEATURES - a logical flag which is set to true when a driver call is performed requesting access to the extended features of the peripheral. This access may not be granted if the caller has insufficient capability. The default is that the user is not in expanded features mode.

INPUT'SEQUENCE'COUNT - an integer counter which contains the input record sequence count. This value is used in error checking to determine if the protocol at the logical level has been violated (such as an entire record lost). This counter is set to zero upon completion of a device clear sequence, and increments by one after reception of an input record.

OUTPUT'SEQUENCE'COUNT - an integer counter which contains the output record sequence count. Each time a record is sent to the peripheral, this value is incremented by one. The peripheral maintains a similar count, which it uses to perform error checking on the records it receives. This counter is set to zero upon completion of a device clear sequence.

RECEIVE'READY'COUNT - an integer counter which maintains the number of available buffers in the peripheral. This count is increased by the value the peripheral sends in its RECEIVE READY report, and is decremented by one each time a record is sent to the peripheral. If the count ever reaches zero, then the logical driver must wait for a RECEIVE READY before it can send any more records.

CPR'XLATE'FLAGS - a double integer which is used by the CIPER function code translator during its process of translating MPE function codes into device recognizable commands.

SEQUENCE'1'BUFFER - a word pointer which contains the CBIX relative address of an array used by the CIPER function code translator to buffer any escape sequences which must be placed ahead of the user's data.

O'R'DATA'TYPE - an integer which contains a code signifying the type of data being currently sent to the peripheral. This is initially set to zero (specifies user data with the control mask invoked), but it may be changed by an appropriate call to the logical driver.

I'R'DATA'TYPE - an integer which contains a code specifying the type of data requested from the peripheral by the user. This is initially set to zero (specifies responces to user escape sequences) but may be changed by an appropriate call to the logical driver.

FILE'OPEN'COUNT - an integer which counts the number of nested file open calls that have currently been made against the device. In the final version of CIPER, this count will be used to determine if the user is finished with the device so resources used by the logical driver may be returned to the system.

DEVICE'ALLOCATED - set TRUE when the first FOPEN is requested. Set FALSE upon completion of device close request.

LOGICAL'DEVICE - an integer which is used to store the logical device number of the device for which this data segment has been allocated. The logical driver will pass this value on to lower levels, as it must reach the physical driver.

CIPER'DST - an integer which is used to store the data segment number of this data segment. The logical driver will pass this down to lower levels, as it must reach the physical driver.

OUT'RECS'OVERWRITTEN - an integer counter which tallies the number of times a device clear command had to be written over an output record buffer of user data. This is used for internal debugging and protocol validation only.

IN'RECS'OVERWRITTEN - an integer counter which tallies the number of times a CLEAR RESPONCE has overwritten user's data in the input record buffer area. This is used for internal debugging and protocol validation only.

DEVICE'BUFFER'SIZE - an integer which contains the size, in bytes, of the peripheral's record maximum record size. This information is returned in the peripheral's CLEAR RESPONCE.

DEVICE'ENV'STATUS'SIZE - an integer which contains the size, in bytes, of the peripheral's largest environmental status report. This information is returned in the peripheral's CLEAR RESPONCE.

PRODUCT'NUMBER - a word pointer which contains the CBIX relative address of a buffer area used to store the ASCII encoded product number of the peripheral. This information is returned in the peripheral's CLEAR RESPONCE.

STORAGE'REQUIREMENTS - an integer which contains the size in words, of the region in the CIPER data segment that the logical driver requires for its buffer areas and other storage. The value contained does not include the size of the CBIX.

TEMP'AREA - a word pointer which contains the DB relative address of a small region of the CIPER data segment which is allocated only during the initialization phases, then later released.

CT'PTR - a word pointer which contains the DB relative address of the control table for the logical device. This is a backward pointer.

PACKET'HEADER'SIZE - an integer which contains the size, in bytes, of the Level 2 packet header. This value is used by the logical driver to reserve space at the front of the record for use by the network protocol level.

PACKET'TRAILER'SIZE - an integer which contains the size, in bytes, of the Level 2 packet trailer. This value is used by the logical driver to reserve space at the front of the record for use by the network protocol level.

PACKET'SIZE - indicates size, in bytes, of level 4 packet.

DEV'CLR'COUNT - count of current recursion level in B08'DEVICE'CLR. If preset limit exceeded, we give up.

DEV'CLR'IN'PROGRESS - a count of how many times the DEVICE CLEAR procedure has been recursively entered. If this count exceeds a preset level, then the DEVICE CLEAR has been unable to restore normal communications with the device, probably due to a catastrophic hardware malfunction.

SR'ENABLE - configuration information used to construct a CONFIGURE record in the event the device powerfails and must be initialized.

ESB'FREQUENCY - configuration information that tells the device how many checkpoints can occur before the transmission of an environmental status block becomes mandatory.

LOGGING'DST - the data segment number of a DST used for performance evaluation. This DST will not be allocated when CIPER is released.

LOGGING'BUFFER - contains the CBIX relative address of an area used for construction of log entries for performance logging.

EVENT'MAP - a bit map that describes which performance events (currently there is only one type defined) are to be logged.

STATUS'ENABLED - a bit map set by the caller (spooler) which defines the types of peripheral status reports the caller is interested in receiving. When (if) any of the enabled types is received, the caller will be notified via a special return code (%41).

STATUS'RECEIVED - bit map of which status types have been received since the last time the caller read those status reports.

STATUS'REPORTED - bit map of which status types that have been received have been reported to the caller via the %41 status return code.

DEFAULT'ACCESS'MODE - during initialization, set TRUE if device subtype=9, otherwise, set FALSE. Indicates whether access mode is FEATURE or TRANSPARENT after a start of job request.

COMP'STAT'AVAILABLE - set to TRUE whenever a new version of composite status becomes available. Set to FALSE whenever composite status is either read or cleared.

Level 4 Control Block Information (CBI)

Word	
0	LVL'2'HEADER'SIZE
1	LVL'2'TRAILER'SIZE
2	LVL'2'PACKET'SIZE
3	HEADER'MOVE'SIZE
4	TRAILER'MOVE'SIZE
5	INITIALIZED

Discussion:

LVL'2'HEADER'SIZE - Contains the number of words required by the physical driver (CIPER level 2) for frame headers. Returned by the physical driver during initialization.

LVL'2'TRAILER'SIZE - Contains the number of words required by the physical driver (CIPER level 2) for frame trailers. Returned by the physical driver during initialization.

LVL'2'PACKET'SIZE - Contains the size (in bytes) of the largest frame the physical driver can accept in one call. Returned by the physical driver during initialization.

HEADER'MOVE'SIZE - Contains the combined number of words (level 4 and level 2) that must be moved to make room for packet and frame trailers.

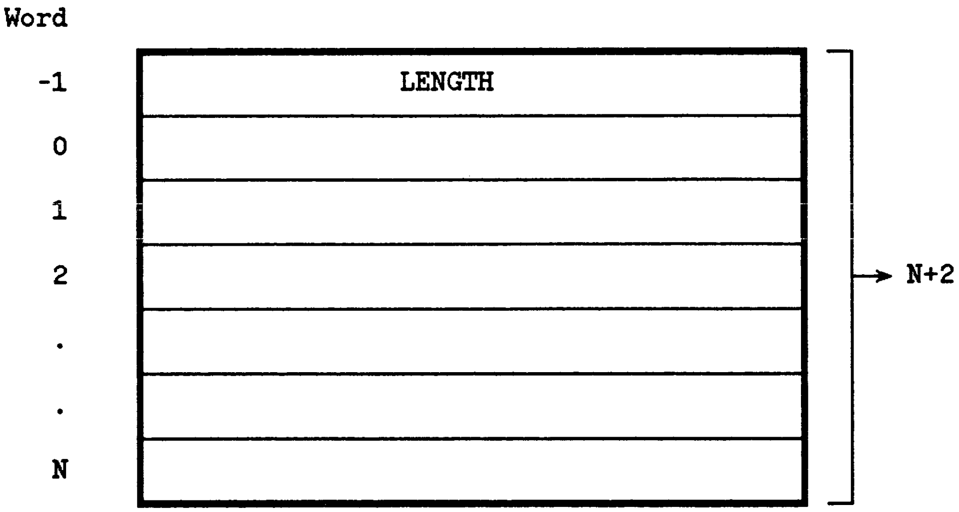
TRAILER'MOVE'SIZE - Contains the combined number of words (level 4 and level 2) that must be moved to make room for packet and frame trailers.

INITIALIZED - Set to TRUE if the CBI has been successfully initialized. Otherwise, set to FALSE.

Control Block Information Extension

The first table is a typical sub-area within the CBIX, such as is used for status tanks, buffer areas, etc. The second table actually shows the order of the different sub-areas within the CBIX. This CBIX is for Level 7, the Logical Driver.

General Entry Format



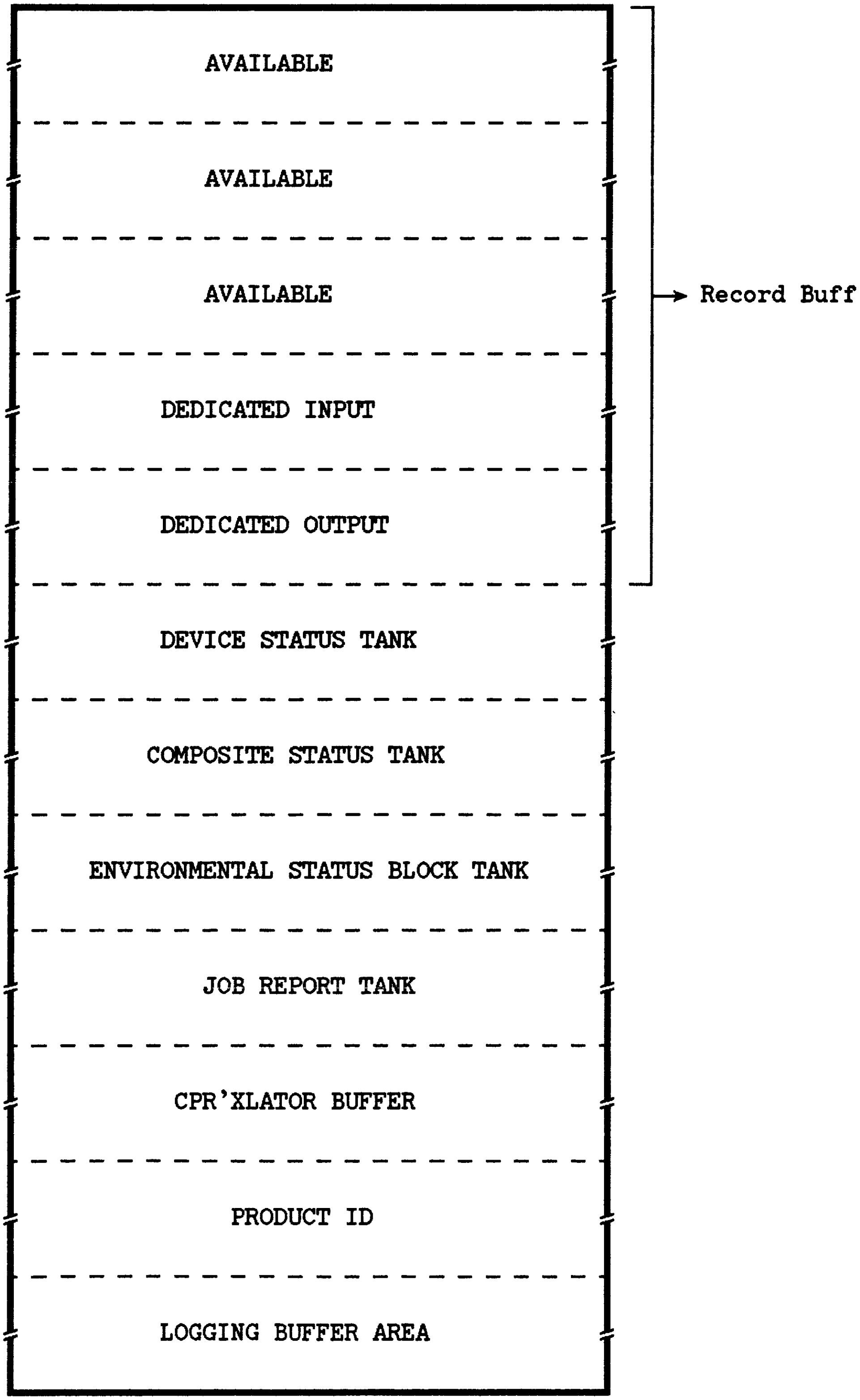
NOTE: LENGTH - Is the size of the sub-area, including the length word itself. Thus, length ALWAYS contains N+2.

Each entry in the CBIX has the general from of the above entry.

Level Seven Control Block Info Extension (CBIX)

The following describes the current CBIX form for level seven. Note that in this layout, the drawing is not to any scale.

Level Seven CBIX (Continued)



NOTES:

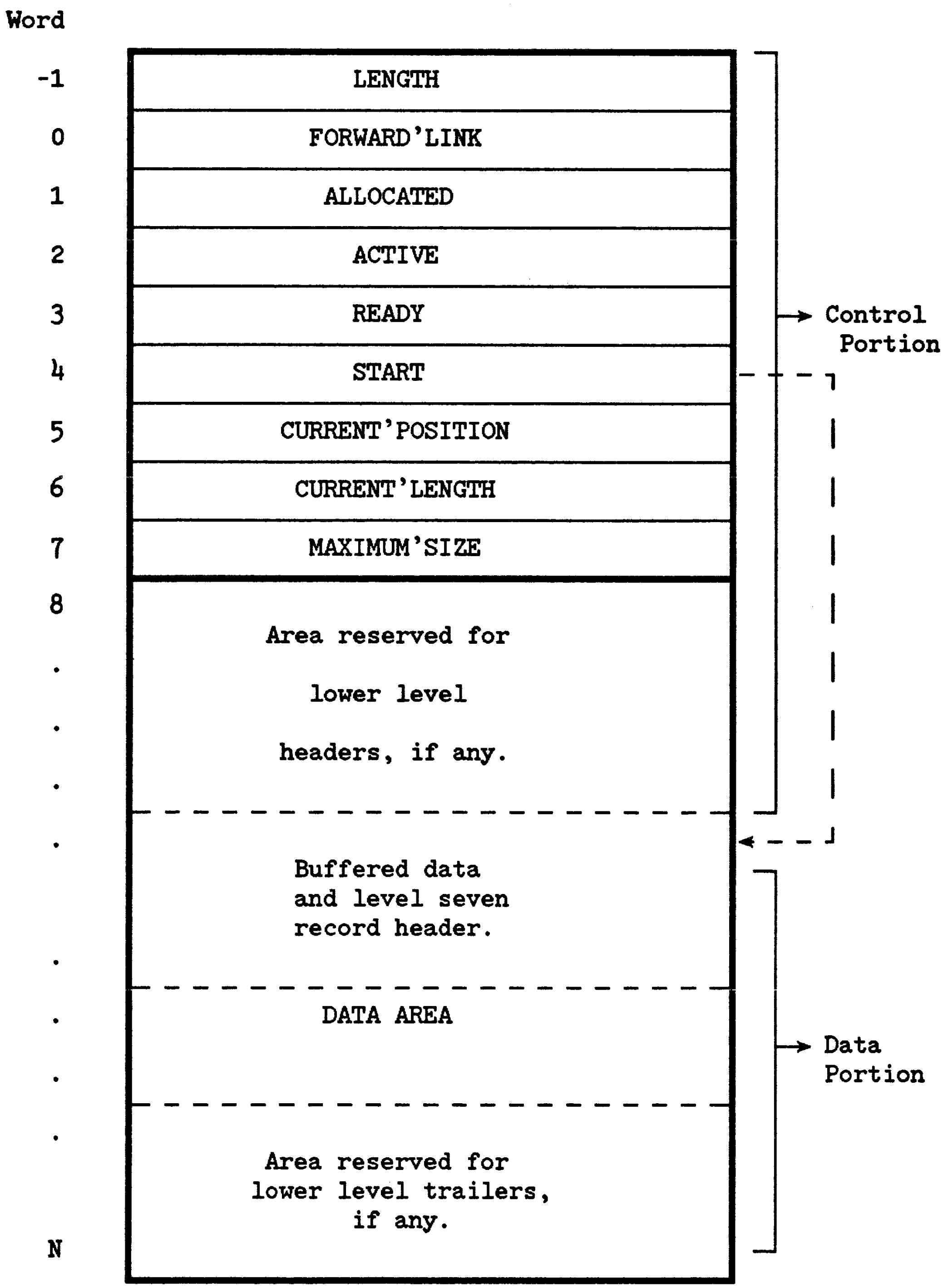
- Device status tank actually holds two copies - one copy of the previous report is used to compare against a new copy to see if any states have changed (such as going from on-line to offline).

- Composite status is the logical "OR" of all device status reports received during a particular call to the logical driver. This was done to reduce the possibility of the calling program missing an error condition due to multiple device status reports over-writing themselves. The area is cleared out at the start of most calls to B08'LOGICAL'DVR, so only those status reports which are received during the call will be returned.

- The logging buffer area is allocated all of the time, but the code to use it is not in place unless SOFTIO is compiled with the X7 toggle set "ON." When logging, this area is used to construct a log record before writing it to a logging data segment. The head entry of the current logging DST is kept in the logging buffer.

- Five record buffer areas are allocated during initialization. One is used as a dedicated output buffer, one is a dedicated input buffer, and the other three are linked into a free-list. The free-list buffers are used to send asynchronous requests (e.g. ESB Immediate) without disturbing a record under construction (such as a write data record).

Typical Record Buffer Area



Discussion:

Control Portion

LENGTH - Is the number of words, including the length word, allocated to a particular buffer area.

FORWARD'LINK - Is used if the buffer area is linked in a free-list. If so, FORWARD'LINK contains the CBIX relative address of the next buffer in the list. Otherwise, contains a zero.

ALLOCATED - Is set FALSE if the buffer area is in the free-list. It is set to TRUE when not in the free-list.

ACTIVE - Is set to TRUE if the buffer area contains any pending data.

READY - Is set to TRUE when a buffer area being used for output is ready for transmission (currently not being used for this release).

START - Is an offset (in words) to the start of the data portion of the buffer area. The offset is relative to the zeroth word of the control portion (not the -1 word!!!).

CURRENT'POSITION - Is an offset (in bytes) to the next available byte in the data portion. The offset is relative to the zeroth word of the control portion (not the -1 word!!!).

CURRENT'LENGTH - Contains a count (in bytes) of the data currently contained in the data portion.

MAXIMUM'SIZE - Contains the maximum number of bytes that a record may contain. This quantity is a device dependent value.

Data Portion

This is where a record going to or coming from the peripheral is assembled or interpreted. In the case of the HP 2608S, the first four (4) bytes are always the record header.

The amount of space required by lower levels for headers and trailers is determined at initialization and the appropriate number of words allocated when the record buffer area is set up.

Logical Device Table Extension (LDTX)

The LDTX is the last of three tables in the LDT data segment. Refer to Chapter 13 for a full description of these tables.
The procedure B08 'Logical' Driver uses the CIPER entry to locate and access the CIPER data segment.

DST %16 = 14

SIR %12 = 10

Zero Entry

	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15
0	Highest Entry	Entry Size
1		
2		
3		
4		

CIPER Entry

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	1	0	reserved					DB	0					
1	CIPER Device Control Data Segment (CDCDS)															
2	DN	CTM Index for this device (CTMI)														
3	0															
4	0															

Discussion:

0.(2:1): This logical device uses the CIPER protocol.

CTMI: Control Table Map Index (an index into the Control Table Map (CTM) which is located in the Ciper Device Control Data Segment (CDCDS)).

DN: Ciper is shutdown. If set, an internal data integrity error has occurred and the device has been locked out from user access.

DB: If set to 1, then debugging in effect.

HIOCIPRO DIT (HP2608S)

There is one DIT per physical device. If a physical device represents more than one logical device, the logical device number is obtained from the IOQ element (however, this driver only supports one device per controller.) The following diagram shows the DIT used for the HP-IB CIPER physical driver.

<table>
<tr><th>Word</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>MNEMONIC</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>AC</td><td>RQ</td><td>0</td><td>0</td><td>0</td><td>IO</td><td>IA</td><td>NO</td><td>ST</td><td>0</td><td colspan="4"></td><td></td></tr>
<tr><td>1</td><td colspan="16">SYSDB relative pointer to the DIT for the next device requesting this resource or service</td><td>DLINK</td></tr>
<tr><td>2</td><td colspan="16">SYSDB relative pointer to the first IOQ in request list for this device</td><td>DIOQP</td></tr>
<tr><td>3</td><td colspan="2">IOT</td><td colspan="6">Phys. unit</td><td colspan="8">Logical device number</td><td>DLDEV</td></tr>
<tr><td>4</td><td colspan="16">SYSDB relative pointer to Device Linkage Table</td><td>DDLTP</td></tr>
<tr><td>5</td><td colspan="16">SYSDB relative pointer to Intrp Linkage Table</td><td>DILTP</td></tr>
<tr><td>6</td><td>VS</td><td>AB</td><td>RE</td><td>TP</td><td>NR</td><td colspan="3">NR CNT</td><td colspan="8">DEVICE STATUS</td><td></td></tr>
<tr><td>7</td><td colspan="16">Hardware error status. Set when the driver detects an error. Whenever <0, the driver monitor logs an I/O error and clears this word</td><td>DSERR</td></tr>
<tr><td>8</td><td colspan="16">Bit 0 is set at completion of timer</td><td>DTIME</td></tr>
<tr><td>9</td><td colspan="16">Holds the time out request entry index while a timer is active.</td><td>DRQST</td></tr>
<tr><td>10</td><td>RF</td><td>UE</td><td>DE</td><td>TO</td><td colspan="3">UNIT CNT</td><td colspan="3">DATA CNT</td><td colspan="3">TO CNT</td><td colspan="3">PRTY CNT</td><td></td></tr>
<tr><td>11</td><td colspan="16">Error logging location 1</td><td>DLOGERROR</td></tr>
<tr><td>12</td><td colspan="16">Error logging location 2</td><td>DLOGCOUNT</td></tr>
</table>

DFLAG - Flags and request state

AC	ACTIVE	- A monitor is currently servicing this device.
RQ	REQUEST	- A service request is pending while the monitor is active.

IO IOPROG - An I/O Channel Program is running for this device.

IA IAK - An interrupt or response has occurred for this device.

NO NOTRDY - Go to state %10 after Idle Channel Program is started.

ST STWAIT - The device monitor is starting an Idle Channel Program for this device. There is no IOQ associated with this type of request.

STATE - State of the device monitor. Specifies the next action to be taken in SIODM in servicing the request:

0 - start new request
1 - not used
2 - call driver initiator procedure
3 - call driver completor procedure
4 - not used
5 - process request completed
6 - initiate device recognition sequence
7 - start operator intervention wait
%10 - wait for interrupt (operator intervention) restart at state 0
%11 - wait for data segment freeze, then state 2
%12 - wait for driver initiator to be frozen, then allocate controller (state 2)
%13 - wait for I/O completion interrupt, then state 3
%14 - wait for controller, then call driver initiator
%15 - not used
%16 - wait for initiator make present, then state 2
%17 - wait for completor make present, then state 3

DLDEV - I/O system type, unit and logical device number

0 - HP3000 Series 2/3
1 - HP3000 Series 33 (HPIB)
2 - Unused
3 - Unused

DSAVE - Device processing flags

VS - VALID STATUS - Set to indicate Device Status has been updated.

AB - DVRABFLAG - Sequence Abort in progress due to ABORT request.

RE - RETRYFLAG - Sequence Abort in progress due to an error.

TP - TIMERPOPPED - Current error is due to software timer popping.

NR - NOTRDYFLAG - Not Ready Wait in progress.

NR CNT	- Number of Not Ready Waits during this request.
DEVICE STATUS	- Device status returned during a Sequence Abort.

BIT 8	- CRC available and enabled.
BIT 9	- Reserved.
BIT 10	- Reserved.
BIT 11	- Reserved.
BIT 12	- Power fail or reset has occurred.
BIT 13	- A protocol error has been detected.
BIT 14	- A parity error has been detected.
BIT 15	- The peripheral has data to send.

DSERR - Pointer to status to be logged.

Bits.(0:8)	- Number of words to be logged.
Bits.(8:8)	- Offset relative to DITP(0).

DCOUNTS	- Error flags and error counts (4).
RF - REQ FAILED	- An error has forced this request to be aborted.
UE - UNIT ERROR	- The current error is a Unit Error.
DE - DATA ERROR	- The current error is a Data Error.
TO - TIME OUT	- The current error is a GIC Time Out Error.
UNIT CNT	- Number of Unit Errors during this request.
DATA CNT	- Number of Data Errors during this request.
TO CNT	- Number of GIC Time Outs during this request.
PRTY CNT	- Number of HP-IB Parity Errors during this request.

CIPER IOQ Element

Word	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15			MNEMONIC
0	Request dependent flags (see below)			QFLAG
1	SYSDB relative pointer to next IOQ element. Points to first word of element.			QLINK
2		Logical device number		QLDEV
3				QMISC
4	S	If QFLAG.(3:1) is clear then this is the DST number of the target data segment. If S is set, QADDR is DB relative.		QDSTN
5	Offset in the data segment or system buffer table to the target data buffer.			QADDR
6	Used by the new Disc routines for special status returns.	Function code for this request. (See next section.)		QFUNC
7	On initiation, specifies the word count (0) or byte count (<0). At completion of the request this location contains the actual transmission count in the same units (bytes or words) as in the request.			QWBCT
8	Parameter 1.			QPAR1
9	Parameter 2.			QPAR2
10	PCBN	QUALIFIER	RSTATUS	QSTAT

QFLAG - Request dependent flags

Bit 0 ABORT - Abort this request and return an error indication to the caller.

Bit 1 SPECIAL - Apply special handling to this request. (Not used)

Bit 2 DIAG - This is a request from the diagnostic subsystem.

Bit 3 SYSBUFF - Target is an index relative to the SBUF Table of the data buffer.

Bit 4 IOWAKE - Wake caller on completion of request.

Bit 5 BLOCKED - Blocked I/O. The caller is waited in ATTACHIO until the request is completed. Implies IOWAKE.

Bit 6 COMPLETED - The request has been completed and the caller awakened if he had requested (with IOWAKE).

Bit 7 DATAFRZN - Set by the memory management routines (MAM) when a MAKEPRESENT request is successfully completed and indicates the data segment is frozen in memory.

Bit 8 MAMERRORD - An error has occurred while MAM was trying to make the target data segment present and freeze it in memory.

Bit 9 PREQ - (Not used)

Bit 10 SFAIL - Delayed failure of SIO instruction. If a call to STARTIO resulted in the request being added to the channel queue, this bit indicates that the SIO instruction failed when the request was selected for execution.

Bit 11 PFAIL - The request was aborted because of a system power failure.

QSTAT - PCB number and request completion status.

PCBN - The Process Control Block (PCB) number of the process which made this request. If zero, the request is not associated with any process and the IOQ element is to be returned by the system when the request has completed.

RSTATUS - General status indicating the final state of the request. The following codes are used:

0 - Not started or awaiting completion.
1 - Successful completion.
2 - End-of-file detected.
3 - Unusual, but recoverable, condition detected.
4 - Irrecoverable error has occurred.

QUALIFIER - A code which further defines or qualifies the general status.

General Status (13:3)	Qualifying Status (8:5)	Overall (8:8)
0 - Pending	1 - Waiting For Completion	%10
	3 - Not Ready Wait	%30
1 - Successful	0 - No Errors	%1

2 - End of File	(Not Used)	
3 - Unusual Condition	3 - Request Aborted	%33
	6 - Powerfail Abort	%63
	%21 - Device Powered Up	%213
4 - Irrecoverable Error	0 - Invalid Request	%4
	1 - Transfer Error	%14
	2 - I/O Timed Out Before Complete	%24
	4 - SIO Failure	%44
	5 - Unit Failure	%54
	%12 - System Error	%124
	%14 - Channel Failure	%144
	%21 - Parity Error	%214

Device Reference Table

There is one DRT per device controller. The contents of this table are used for processing interrupts.

Word	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	MNEMONIC
0	Channel Program Pointer (SIOP)																DRT0
1	Channel Program Variable Area pointer (CPVA)																DRT1
2	Interrupt Handler Program Label																DRT2
3	ST	SH	PF	(status)									WS	GF	DT	WT	DRT3

DRT3

Bit 0 - ST, Channel Program Status; 0 - halted, 1 - running
1 - SH, SIOP or HIOP instruction pending
2 - PF, Power Fail recovery in progress
12 - WS, Waiting for device status request
13 - GF, GIC FIFO buffer not empty
14 - DT, DMA transfer active
15 - WT, Channel Program in Wait state

Interrupt Linkage Table

There is one ILT for each device controller configured on the system. A controller may support more than one unit, however the HP-IB CIPER physical driver currently only supports one unit.

<table>
<tr><th>Word</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>MNEMONIC</th></tr>
<tr><td>0</td><td colspan="16">Channel</td><td>ICPVA0</td></tr>
<tr><td>1</td><td colspan="16">Program</td><td>ICPVA1</td></tr>
<tr><td>2</td><td colspan="16">Variable</td><td>ICPVA2</td></tr>
<tr><td>3</td><td colspan="16">Area (ICPVA)</td><td>ICPVA3</td></tr>
<tr><td>4</td><td colspan="16">DMA
Abort</td><td>ICPVA4</td></tr>
<tr><td>5</td><td colspan="16">Address</td><td>ICPVA5</td></tr>
<tr><td>6</td><td colspan="16">0</td><td>ISRQL</td></tr>
<tr><td>7</td><td>LI</td><td colspan="6">CHANQUE</td><td colspan="2"></td><td colspan="4">CHAN</td><td colspan="3">DEV</td><td>ICNTRL</td></tr>
<tr><td>8</td><td colspan="16">SYSDB relative pointer to Channel Program area</td><td>ISIOP</td></tr>
<tr><td>9</td><td colspan="16">SYSDB relative pointer to Status Return area. (Always zero for this driver.)</td><td>ISTAP</td></tr>
<tr><td>10</td><td colspan="16">single instruction that is executed to extract the device unit number from the status pointed to by ISTAP. (Since there is only one unit on the controller, this entry is not used.)</td><td>IUNIT</td></tr>
<tr><td>11</td><td colspan="16">SYSDB relative DIT pointer of the device currently using the channel to perform a data operation.</td><td>ICDP</td></tr>
<tr><td>12</td><td colspan="8">SIOPSIZE</td><td colspan="8">CQUEN</td><td>IQUEUE</td></tr>
<tr><td>13</td><td>RW</td><td>WP</td><td>IG</td><td colspan="9"></td><td colspan="4">HCUNIT</td><td>IFLAG</td></tr>
<tr><td>14</td><td colspan="16">SYSDB relative DIT pointer for unit 0</td><td>IDITP0</td></tr>
<tr><td>15
.
.</td><td colspan="16">Peripheral
Channel
Program</td><td></td></tr>
<tr><td>N</td><td colspan="16">(Variable length)</td><td></td></tr>
</table>

ICPVA0/3 - Channel Program Variable Area

The first word is used by the channel program processor to store status information after I/O channel aborts. The next word is used by the driver to indicate if status should be examined for special conditions or errors. The other two words are not used.

ICPVA4/5 - DMA abort address

If a DMA abort occurs, the absolute address where the abort occurred is stored in this area.

ICNTRL - Contains controller information

LIM - If this bit is set, the controller is sharing a software channel resource in order to limit bandwidth.

CHANQUE - The software channel resource number.

CHAN - Channel number (four most significant bits of DRTN).

DEV - Device number (three least significant bits of DRTN).

IQUEUE -

SIOPSIZE - (number of words + 1)/2 in the channel program area.

CQUEN - For a multi-unit controller this field contains the software controller resource number.

IFLAG - Controller and Channel Program state flags

RUNWAIT - An Idle Channel Program should be started when there are no active requests to process. This flag is always 0 for this version of the driver.

WAITPROG - An Idle Channel Program has been started for this controller. This bit is reset by an interrupt.

IGNOREHI - An HIOP instruction has been issued against this controller but the channel program was not in a wait statement. Therefore ignore the interrupt generated by the channel code when this program halts.

HCUNIT - Highest configured unit number for this controller.

Part No. 32002-90003
Printed in U.S.A. 11/82
E1182

HEWLETT
PACKARD