# TOPS-10 Tables Descriptions

April 1986

ABSTRACT

This document contains illustrated descriptions of the tables used by the TOPS-10 monitor and the ANF-10 networking softare to allocate memory, control jobs and resources, and provide information.

Operating System: TOPS-10 Version 7.03
GALAXY Version 5.1
DECnet-10 Version 4.0

.lm0

Printed: April 1986



The postage-prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

CONTENTS

## *FOREWORD*

TOPS-10 Monitor Tables Descriptions are the result of the effort to document and illustrate information that system analysts and programmers might need to repair or modify TOPS-10 monitor source code. This document is provided by the Software Publications department for LCG Software Engineering without guarantee of technical accuracy.

The tables are simply an extension and summary of the monitor source code. To the inexperienced user, they may seem bewilderingly complex. However, to the user with experience in TOPS-10 monitor internals, and in the TOPS-10 data structures, this document can be a useful reference for a conceptual view of the monitor.

The monitor source code itself is complex and lengthy, since it allows many types of hardware and many software applications. The code does not regard the support status of any product, and this document follows suit. For complete information about product support, refer to the current TOPS-10 Software Product Description.

This document is only a summary of the information written and processed by the monitor. It is an effort to record the information critical in developing and maintaining TOPS-10. Do not assume this document is correct in all cases. Please refer to source code listings of the appropriate modules whenever using the information in these tables.

The monitor tables are listed to reflect a TOPS-10 operating system based on one or more KL processors. The information will be different for any other type of system.

The data structures for the ANF-10 network software are listed at the end of this document, in Appendix A.

## Conventions

The TOPS-10 tables are described in a consistent format that is used to illustrate actual assembly code. The tables are described as follows.

Table Name -- Descriptive title of table

Description:    A description of the table, including how and when it is used.

Defined in:    Module where table is defined.

Used by:    Modules that access the table.

GETTAB Tables: GETTAB UUO symbolic index, followed by the GETTAB table number in parentheses.

See also:    Where to look for more information.

```
           +-------------------------------------------------------------+
Word name  |          General description of word's contents             |
           |-------------------------------------------------------------|
           |         Left half             |          Right half         |
           |-------------------------------------------------------------|
 Symbol*   |    * means "see following pages" for more information       |
           |-------------------------------------------------------------|
 Symbol(2) |     (2) means "see Note no. 2" following the table          |
           +-------------------------------------------------------------+
```

* Special information about one or more words in the table.

Notes:

1. Notes contain more information about about the monitor table and the way the data is used by the monitor.

2. Numbered notes refer to specific words or items in the table. These items are flagged by (n), where n is the number of note.

3. Notes also describe any restrictions or warnings in using the data in the table.

4. Data words that require more detail are listed on the subsequent pages, as shown below.

Following the monitor table, a word-by-word description of the information stored in table is sometimes necessary. Each complex word is illustrated as shown here. The contents of bytes and the meaning of certain bit settings are listed after the illustration, if necessary.

```
+-------------------------------------------------------------+
|Breakdown of bytes   | X|  and bits    | X|  UN1 | X| X| X| X |
+-------------------------------------------------------------+
 0                  17 18 19        26 27 28   31 32 33 34  35
```

| Word Symbol | Bits | Mask Symbol | Description |
|---|---|---|---|
| WORD | 0-17 | WR.LH | Description of contents. |
| WORD | 18 | WR.MB | Meaning of bit settings. |
| WORD | 19-26 | WR.FL | Contents of bytes (fields). |
| WORD | 27 | WR.X | Flag settings. |
| WORD | 28-31 | WR.UN1 | Use of the data. |
| WORD | 32 | WR.FL1 | Results of setting bits. |
| WORD | 33 | WR.FL2 | Setting flags. |
| WORD | 34 | WR.FL3 | Storing codes. |
| WORD | 35 | WR.FL4 | Reading data. |

## 1 ACC -- ACCESS TABLE

Description: Contains information needed in order to gain access to a specific version of a specific file.

One table for each existing version of each open file.

The access table for the current and possibly a superseding version of the file are linked to the corresponding NMB. Also, each disk device data block contains a pointer to the access table for the file currently being accessed.

Defined in: COMMOD

Used by: FILFND, FILIO, FILUUO, IPCSER, SEGCON, SYSINI, UUOCON

See also: NMB, PPB

```
            +------------------------------------------------------------+
ACCALC (2)  |           Highest relative block number allocated          |
            |------------------------------------------------------------|
ACCNMB*     |    NMB or next ACC           |  See following page         |
            |------------------------------------------------------------|
ACCPT1      |               First retrieval pointer to file              |
            |------------------------------------------------------------|
ACCDOR (1)  |    Next dormant ACC          |  Previous dormant ACC       |
            |------------------------------------------------------------|
ACCPPB*     |  See following pages         |            PPB              |
            |------------------------------------------------------------|
ACCADT*     |                    See following pages                     |
            |------------------------------------------------------------|
ACCWRT (2)  |           Highest relative block number written            |
            |------------------------------------------------------------|
ACCPRV*     | Privileges  |Mode|  Creation time  |Low creation date      |
            +------------------------------------------------------------+
```

* Details on following pages.

Notes:

1. When all users have closed a file, its access table is considered dormant. Dormant access tables are linked into a doubly linked list through ACCDOR, and are not deleted until their core space is needed. If a table is not dormant, ACCDOR contains zero.

2. ACCWRT and ACCALC do not include the second RIB.

1.1 ACCNMB -- Next ACC Byte

```
+------------------------------------------------------------+
|  NMB or Next ACC             | A  | LBS |1| UN1 | D| N| S| G|
+------------------------------------------------------------+
 0                          17  18 19-26 27 28-31 32 33 34 35
```

| Word Symbol | Bits | Mask Symbol | Description |
|---|---|---|---|
| ACCNMB | 0-17 | | Address of next ACC for same name and PPN if there is one. |
| ACCABC | 18 | ACPABC | This file always has bad checksum. |
| ACCLBS | 19-26 | ACYLBS<br>ACZLBS | Number of words in last data block of file. |
| ACC1PT | 27 | ACP1PT | File has only one retrieval pointer. |
| ACCUN1 | 28-31 | COYUN1 | Logical unit number, within structure, where file begins. |
| ACCDIR | 32 | ACPDIR | This file is a directory. |
| ACCNDL | 33 | ACPNDL | This file cannot be deleted. |
| ACCSBC | 34 | ACPSBC | Sometimes bad checksum. |
| ACCGRB | 35 | ACPGRB | Don't grab access table. |

Notes:

1. Normally there is only one ACC linked to an NMB. But while a file is being superseded, both the old and the new versions of the file have ACCs linked to the NMB.

2. The low order two bits of a pointer to another ACC will be zero. The pointer back to the NMB points to Word 2 of the NMB, NMBACC. Hence, its low order bits will not be zero.

3. There may be an arbitrary number of ACCs for older versions of a file, which are still being read. However, these ACCs are not linked to the NMB. The only pointers to them are in the DDBs of the readers.

1.2 ACCPPB -- PPB Address

```
+-------------------------------------------------------------+
|   FSN    |Sim. updates|Z|   |         PPB                    |
+-------------------------------------------------------------+
 0--------5 6---------13 14    18--------------------------35
```

| Word Symbol | Bits | Mask Symbol | Description |
|---|---|---|---|
| ACCFSN | 0-5 | ACYFSN<br>ACZFSN | File structure number of structure to which this file belongs. |
| ACCWCT | 6-13 | ACYWCT<br>ACZWCT | Write count for simultaneous updates |
| ACCZRB | 14 | ACPZRB | If SFD, the SFD has empty data blocks. |
| ACCPPB | 18-35 | | Core address of project-programmer data block (PPB) |

1.3 ACCADT -- Access Status

```
+-------------------------------------------------------------+
| A | Last Access Date    | N| Count/Use| R| D| STS | S| P| N|
+-------------------------------------------------------------+
 0-2 3------------------17  18 19-----27 28  29 30-32 33 34 35
```

| Word Symbol | Bits | Mask Symbol | Description |
|---|---|---|---|
| ACCADT | 0-2 | | High-order 3 bits of creation date. Low-order part is in ACCCDT. |
| ACCADT | 3-17 | | Date this file was last accessed for more than just LOOKUP. |
| ACCNIU | 18 | ACPNIU | File not in UFD. |
| ACCCNT | 19-27 | ACYCNT<br>ACZCNT | Count of user channels with LOOKUP in force for this version of this file or SFD use count. |
| ACCREN | 28 | ACPREN | Rename in progress. |
| ACCDEL | 29 | ACPDEL | File to be deleted when all readers finished. |
| ACCSTS | 30-32 | ACYSTS | Access table status code |
| | 30 | ACPCRE | File being created. |
| | 31 | ACPSUP | File being superseded. |
| | 32 | ACPUPD | File being updated. |
| ACCSMU | 33 | ACPSMU | This file being simultaneously updated. |
| ACCPAL | 34 | ACPPAL | Pre-allocated file. |
| ACPSTS | 35 | ACPNDR | QUESER "don't delete on reset" bit. |

Notes:

1. Access Table state codes are:

   | Code | Symbol | Meaning |
   |---|---|---|
   | 4 | ACRCRE | File being created. |
   | 2 | ACRSUP | File superseding another. |
   | 1 | ACRUPD | File being updated. |

2. The ACCADT word is called ACCUSE when the file is stored in an SFD.

1.4 ACCPRV -- Privilege Code

```
+-----------------------------------------------------------+
| Privilege | Mode  |   Creation time    |    Creation date  |
+-----------------------------------------------------------+
 0---------8 9----12 13---------------23 24----------------35
```

| Word Symbol | Bits | Mask Symbol | Description |
|---|---|---|---|
| ACCPRV | 0-8 | ACYPRV | Privileges (described below). |
| ACCMOD | 9-12 | | Data mode of file. |
| ACCCTM | 13-23 | | File creation time. |
| ACCCDT | 24-35 | | Low-order 12 bits of file creation date. Upper 3 bits in ACCADT. |

Notes:

Privilege codes are stored in the following format:

| Bits | Meaning |
|---|---|
| 0 | FILDAE called on protection failure. |
| 1-2 | Owner's protection. |
| 4-7 | Not used. |
| 3-5 | Apply to any job with matching project number. |
| 6-8 | Apply to all other jobs. |

A job is considered the owner of a file if one of the following is true:

1. INDPPN set to 0 at MONGEN (default) and programmer number matches.

2. INDPPN set to -1 at MONGEN and both project and programmer number match.

Privilege codes for user files:

| Code | Highest Privileges |
|---|---|
| 7 | None |
| 6 | Execute |
| 5 | Read |
| 4 | Append (allocate, deallocate) |
| 3 | Update |
| 2 | Write (supersede, truncate) |
| 1 | Rename (change attributes) |
| 0 | Change privileges |

Privilege codes for directories:

| Code | Privilege Given by Bit Being Set |
| --- | --- |
| 4 | Allows LOOKUPs in this directory |
| 2 | Allows creates |
| 1 | Allow directory to be read as a data file |

Any combination of these bits may be set.

## 2 AVALTB -- AVAILABLE RESOURCE TABLE

Description: Contains flags to indicate that a sharable device has become available. Each entry referenced by its own label.

Entry is -1 if the corresponding sharable resource has become available since the last scheduling and some job is waiting for it.

Defined in: COMMON

Used by: CLOCK1, COMMON, CPNSER, SCHED1, SYSINI

See also: REQTAB

The words in AVALTB are stored in the following order. However, not all words may be present on all systems.

| Word | Symbol | Contents |
|---|---|---|
| 0 | AUAVAL | Alter disk UFD quota |
| 1 | DAAVAL | Disk storage allocation wait |
| 2 | CBAVAL | Disk core block scan wait |
| 3 | DTAVAL | DECtape control wait |
| 4 | IPAVAL | IPCF interlock wait |
| 5 | CXAVAL | Context save wait |
| 6 | DCAVAL | Data control wait (magtape and DECtape) |

The following words are conditional and depend on a feature test option to be included (see Note 2):

| Word | Symbol | Cond. | Contents |
|---|---|---|---|
| 7 | CAAVAL | FTLOCK | Semi-permanent core allocation wait |
| 15 | MMAVAL | FTMP | Memory management wait |
| 16 | EVAVAL | FTMP | Exec virtual memory wait |
| 17 | EQAVAL | FTEQDQ | ENQ/DEQ wait |
| 20 | MCAVAL | FTMP | Monitor I/O disk cache wait |

Special Notes:

1. Table REQTAB has entries corresponding to the AVALTB entries.

2. The AVALTB entries are built by the conditionally assembled RWAITS macro in S.MAC; therefore, some of the above listed entries may not be present in all systems.

3. This table is initialized to zero.

## 3 BAF -- BAD ALLOCATION FILE

Description: Disk block in which all known bad regions in a structure are recorded.

The BAF is always the next block after the home block in HOME.SYS, with a copy in Block 11 (decimal). The containing the BAF will be marked in the file HOME.SYS; information from the BAF is copied into the file BADBLK.SYS by the monitor.

Defined in: COMMOD

Used by: FILIO, FILUUO, ONCMOD, REFSTR

```
Symbol       Map
            +-----------------------------------------------------------+
BAFNAM      |        SIXBIT         / BAT  /                            |
            |-----------------------------------------------------------|
BAFFIR (3)  |  No. of words in             |        Rel adr of first    |
            |  in BAFREG area              |        bad region pair  (1)|
            |-----------------------------------------------------------|
BAFNBS*     |    NBS    |    NBR           | KDC  |                     |
            |-----------------------------------------------------------|
BAFCNT      |           No. pairs added to BAF by monitor               |
            |-----------------------------------------------------------|
BAFREG*(1)  |                  Bad region pair                          |
            |-----------------------------------------------------------|
            |                  Bad region pair                          |
            |-----------------------------------------------------------|
            /                            .                              /
            /                            .                              /
            /                            .                              /
            |-----------------------------------------------------------|
BAFCOD      |        0                     |   Unlikely code (606060)   |
            |                              |                            |
            |-----------------------------------------------------------|
BAFSLF      |        0                     |    This block # in unit    |
            |                              |                            |
            +-----------------------------------------------------------+
```

Notes:

1. The label BAFREG should not be used by programs that look at this block. They should use the right half of BAFFIR to determine location of the first bad region pair.

2. The Map Program is a stand-alone program that checks all disk blocks and writes an initial BAF. As the monitor finds bad blocks, it makes additional entries in the BAF.

3. Both halves of BAFFIR are written by the mapping program and are never changed by the monitor. The left half is always an even number.

3.1 BAFNBS -- Bad Blocks

```
+-------------------------------------------------------------+
|   NBS    |     NBR            |   KDC    |                   |
+-------------------------------------------------------------+
 0------8  9---------------17  18-----24 25                   35
```

| Word Symbol | Bits | Mask Symbol | Description |
|---|---|---|---|
| BAFNBS | 0-8 | | Number of bad blocks found by the map program. |
| BAFNBR | 9-17 | BAYNBR | Number of bad regions found by the map program. (Number of entries in BAFREG table.) |
| BAFKDC | 18-24 | BAYKDC | Controller device code used by map program. |

## 3.2 BAFREG -- Bad Region Word Pair

First Word of Each Bad Region Pair

```
+------------------------------------------------------------+
|     NBB     | 0 |      PUB      | KNM | |          APN      |
+------------------------------------------------------------+
 0--------8    9  10---------17 18-20    22-----------------35
```

| Word Symbol | Bits | Mask Symbol | Description |
|---|---|---|---|
| BAFNBB | 0-8 | BAYNBB | Number of bad blocks -1 in this bad region (not clusters). |
| BAFOTH | 9 | BAPOTH | Non-zero if this bad region is detected on another controller or processor than the one that originally added the entry. |
| BAFPUB | 10-17 | BAPPUB | Physical unit within controller. Bit 17-n represents unit n, where n = 0-7. |
| BAFKNM | 18-20 | BAYKNM | Logical controller number, of this type. From UNIKNM. |
| | 21 | BAPNTP | Non-zero if new-style BAT block entry. |
| BAFAPN | 22-35 | BAYAPN | Serial number of APR running when error was detected. |

Second Word of Bad Region Pair

```
+------------------------------------------------------------+
| Ver  |OTR|DTR|HDR|      |  LBN of 1st bad block in region    |
+------------------------------------------------------------+
 0---2  3   4   5  6---8 9---------------------------------35
```

| Word Symbol | Bits | Mask Symbol | Description |
|---|---|---|---|
| BAFVER | 0-2 | | Version number of entry (presently 0) |
| BAFERR | 3 | BAPOTR | Other error (not data or search error) |
| | 4 | BAPDTR | Data error (parity or ECC hard error) |
| | 5 | BAPHDR | Search error or header compare error |
| BAFELB | 9-35 | | LBN of first bad block in region |

Notes:

Values of the word labels are relative to the beginning of the bad region pair, that is, 0 and 1.

## 4 USER I/O BUFFER

Description: Contains pointers and storage area for I/O data. Set up by user programs as needed.

Defined in: Status bits defined in S.MAC

```
+-------------------------------------------------------------+
|                             |        File status bits       |
|-------------------------------------------------------------|
| U |      Buffer size         |        Next buffer address    |
|-------------------------------------------------------------|
|          Bookkeeping         |        Word count, n          |
|-------------------------------------------------------------|
/                 n = number of data words                    /
|-------------------------------------------------------------|
/                         Unused area                         /
/                        (S-n)-1 words                        /
+-------------------------------------------------------------+
```

### 4.1 First Word -- I/O Status Bits

| Bit | Meaning |
|---|---|
| 18 | Improper mode. |
| 19 | Device detected error. |
| 20 | Data error, for example, checksum failure. |
| 21 | Block too large. |
| 22 | End of file. |
| 23 | Device is actively transmitting or receiving data. |
| 24-29 | Device-dependent parameters. |
| 30 | Synchronous input. Stop device after each buffer is filled. |
| 31 | Word count supplied by user for output. (Monitor normally computes word count.) |
| 32-35 | Data mode. |

### 4.2 Second Word -- Use Bit

| Bits | Symbol | Description |
|---|---|---|
| 0 | IOUSE | "Use" bit is set as follows: |

| I/O | In Progress | Finished |
|---|---|---|
| Input | 0 | 1 |
| Output | 1 | 0 |

Note that 0 means the buffer is available to filler; 1 means buffer is available to emptier.

| Bits | Symbol | Description |
|---|---|---|
| 1 | IOIBC | Inhibit zeroing output buffers at completion of output |
| 2-17 | IOSIZ | Size of buffer, not counting first two words |
| 18-35 | | Address of second word of next buffer of ring. |

### 4.3 Third Word -- Word Count

| Bit | Meaning |
|---|---|
| 0-17 | Depends on type of device and mode |
| 18-35 | Word count. (Normally computed by monitor for the device being used.) |

5 BUFFER RING HEADER

Description: Contains information that is used by programs to access their I/O buffers.

Defined in: S.MAC

```
Symbol      Map
           +-------------------------------------------------------------+
JBFADR     | V |                                | Current buffer address  |
           |-------------------------------------------------------------|
JBFPTR(3)  |                        Byte pointer                          |
           |-------------------------------------------------------------|
JBFCTR(3)  |                         Byte count                           |
           |-------------------------------------------------------------|
JBFUDX     |                Universal Device Index (UDX)                  |
           +-------------------------------------------------------------+
```

Label values are relative to the beginning of the buffer ring header.

| Word Symbol | Bits | Description |
|---|---|---|
| JBFADR | 0 | Virgin buffer ring bit (will be 1 if buffer ring has been set up but not referenced. |
| | 18-35 | Address of second word of the buffer currently available to the user program. |
| JBFPTR | 0-35 | Byte pointer for user to access next byte of buffer with ILDB or IDPB. |
| JBFCTR | 0-35 | Number of remaining bytes available to user. |
| JBFUDX | 0-35 | UDX for MSGSER (MPX-controlled devices only) |

Notes:

1. Ring header must be set up within user program.

2. Values are initialized by each INPUT or OUTPUT UUO, for the buffer made available to the user.

3. User program must keep JBFPTR and JBFCTR up to date as it uses the buffer.

## 6 CDB -- CPU DATA BLOCK

Description: There is one CPU data block for each CPU in the system. The block contains both a constant and a variable data area, in contiguous address space. Routine addresses, pointers, bit masks, and processor-dependent instructions are representative of the data found in the CPU Constants Area. The CPU Variables Area is cleared at initialiation and on 403 restarts. This area contains current status words for the CPU, including current job information, protected job data locations, scheduler and swapper parameters, and so forth.

References can be made to a CDB location without indexing. The symbols used in this case will be in the form of .Cnxxx, where n is the CPU number (such as .C0xxx, .C1xxx, .C2xxx, and so on).

A "C" macro is used to generate labels and data for the Constants Area. A "V" macro is used to define symbols and allocate space for the Variables Area. No data is generated by a "V" macro call. The "C" and "V" macro calls are contained in the CDB macro, which is expanded once for each CPU.

Defined in: COMMON

Used by: CLOCK1, CP1SER, ERRCON, KLSER, KSSER, PSISER, SCHED1, SYSINI, UUOCON

GETTAB Tables:

```
.GTC0C (55), .GTC0V (56)
.GTC1C (57), .GTC1V (60)
.GTC2C (61), .GTC2V (62)
.GTC3C (63), .GTC3V (64)
.GTC4C (65), .GTC4V (66)
```

NOTE

All CDB symbols in the following tables require a prefix of .CP or .Cn, where n is the CPU number.

The following tables represent a KL-SMP (multiprocessor) system, assembled with KL-paging enabled. Entries will vary for other types of systems.

CPU Data Block -- Constants Area

| Symbol | Map |
| --- | --- |
| CDB | Address of next CDB,,0 |
| ASN | APR serial number |
| OK | Number of jiffies since this CPU stopped. (if the value of this location is less than or equal to 0, this CPU is running.) |
| EPT | Physical address of EPT |
| LOG | Logical CPU name as SIXBIT/CPUn/ (n=CPU number) |
| PHY | Physical CPU name as SIXBIT/CPUxn (x=S or L,n=CPU number) |
| TYP | CPU type, 4=KL10, 5=KS10 |
| MPT | Pointer to bad address subtable bits. Bits 0-8=length \| RH=offset into CDB |
| RTC | Real time clock (DK10) DDB adr |
| RTD | DK10 DDB adr if high prec. time, 0 if low (APR clock) |
| PAR | Pointer to parity subtable Bits 0-8=length \| RH=offset into CDB |
| RSP | Pointer to response subtable Bits 0-8=length \| RH=offset into CDB |
| DKX | Number of DK10s on this CPU |
| EBS | EBOX ticks per second |
| MBS | MBOX ticks per second |
| NMT | Pointer to NXM subtable Bits 0-8=length \| RH=offset into CDB |
| CSB | Pointer to CPU status block Bits 0-8=length \| RH=offset into CDB |

| | |
|---|---|
| DSB | Pointer to device status block<br>Bits 0-8=length \| RH=offset into CDB |
| SDP | Pointer to SBDIAG subtable<br>Bits 0-8=length \| RH=offset into CDB |
| BPA | Performance analysis subtable ptr<br>Bits 0-8=length \| RH=offset into CDB |
| MAP | Addr of CPU's exec map |
| SPT | Special pages table |
| XPT | Temporary storage for SPT |
| CHX | This CPU's bit in TKBSTS word of MTA KDB, indicating a sweep needs to be done. |
| CPN | CPU number |
| SK0 | Generate SKPCPU(0) (Instruction to skip only on CPU0) |
| SK1 | Generate SKPCPU(1) (Instruction to skip only on CPU1) |
| OK1 | Address of policy CPU's OK word |
| SLF | Pointer to start of CDB |
| SCN | Scheduler run queue scan list addr (SSCAN or SSCAN1) |
| SST | Address of subqueue scanning table |
| NPD | Null PDL pointer |
| EPD | Error PDL pointer |
| NJD | Address of null job data area (offset by 20) |
| STO | Scanner once-a-tick routine address |
| ISR | Scanner once-a-second routine address |
| DLK | Calls DSKLOK ownership flag (0 = currently owns DSKLOK) |

| | |
|---|---|
| SCD | Scheduler interlock flag (-1=doesn't own interlock, 0=owns interlock, n>0=has owned interlock) |
| RES | Address of power fail restart return |
| NBI | Number interrupts broken by BRKLOK |
| ABK | Return address for CPNBPT |
| KAF | Keep-alive-failure dispatch address |
| EPL | Address of this CPU's PDL for processing PDL overflows |
| NAP | CONSO, CONI PI bits for all PIs except APR PI in progress |
| APP | CONSO, CONI PI bits for APR PI in progress |
| API | APR PI channel for this CPU |
| ACO | CONSO PI bit for APR PI on |
| CHL | Address of interrupt PC for interrupt level (APnCHL) |
| CKL | Address of interrupt PC for clock level (CKnCHL) |
| CON | APR CONSO mask for currently enabled condition |
| EEB | Standard EXEC enabled CONSO bits |
| IEF | Mask to clear all interrupting APR error flags except parity and clock errors and sweep done |
| CCF | Clear clock flag instruction |
| HCT | Instruction that skips if clock has ticked. |
| MPI | Address of parity sweep instruction (CPLMPI or CPSMPI) |
| MPS | Bits to request parity sweep \| Addr of sweep subroutine |
| NXM | Mask to test/clear APR NXM bit |
| MPE | Mask to test/clear APR parity err |

```
            |-------------------------------------------------------------|
SCS         |  Scheduler doorbell bits for all CPUs except this CPU       |
            |-------------------------------------------------------------|
SCC         |  Scheduler doorbell bit for this CPU                        |
            |-------------------------------------------------------------|
QPS         | Queued protocol doorbell bits for all CPUs except this CPU|
            |-------------------------------------------------------------|
QPC         |  Queued protocol doorbell bit for this CPU                  |
            |-------------------------------------------------------------|
DBM         |  Mask of all doorbell bits for this CPU                     |
            |-------------------------------------------------------------|
EBR         |              Exec base register on this CPU                 |
            |-------------------------------------------------------------|
CTN         |  CTY number for this CPU      |  DLS line number on DTE     |
            |-------------------------------------------------------------|
DTN         |                 Number of DTEs on this CPU                  |
            |-------------------------------------------------------------|
CAC         /  Saved AC set 0 at start of stopcode processing             /
            |-------------------------------------------------------------|
STT         |              K?SER temp for trap processing                 |
            |-------------------------------------------------------------|
ST1         |                  Saved T1 on page traps                     |
            |-------------------------------------------------------------|
ST2         |                  Saved T2 on page traps                     |
            |-------------------------------------------------------------|
EJ1         |               K?SER temp for IME processing                 |
            |-------------------------------------------------------------|
EJ2         |               K?SER temp for IME processing                 |
            |-------------------------------------------------------------|
EJ3         |               K?SER temp for IME processing                 |
            |-------------------------------------------------------------|
EJ4         |               K?SER temp for IME processing                 |
            |-------------------------------------------------------------|
RCT         /                Real time PI channel table                   /
            /                       (6 words)                             /
            |-------------------------------------------------------------|
RDT         /                 Real time dismiss table                     /
            /                       (6 words)                             /
            |-------------------------------------------------------------|
CPI         |          CONI PI at start of stopcode processing            |
            |-------------------------------------------------------------|
```

| | |
|---|---|
| SVA | Stopcode processing JSRs here to save all AC sets |
| SVB | Instruction that jumps to routine to save all AC sets (JRST SVSETS) |
| TRP | Current MUUO saved at start of stopcode processing |
| RTS | Temporarily used during RTTRP error procedure |
| RTT | "RTTRP in progress" Flag |
| TML | Value of time at last clock tick |
| OCB | 0 if this CDB isn't owned by a CPU |
| AID | -1 if restart (not initial startup) |
| DWD | Recursion interlock for DIE |
| SAV | PI save routines<br>Tape PI Save routine addr \| Disk PI Save routine addr |
| ACD | Zero if AUTCON has run on this CPU |
| TIL | AUTCON tape interlock word |
| NUM | Starting Controller for AUTCON (RPx, RNx, MTx) |
| CML | Address of start of tape channel's interrupt routine |
| TCH | Used for vectored tape interrupts |
| SPR | Instruction to cause parity error interrupt |
| CPR | Instruction to clear parity error flag |
| SBO | SBDIAG function 0 argument |
| SOA | SBDIAG function 0 answer |
| SB1 | SBDIAG function 1 argument |
| S1A | SBDIAG function 1 answer |

| | |
|---|---|
| TOA | Addr of character typeout routine |
| TIV | Addr of vector with input routines |
| NLD | DX20 auto-reload flag: 0 = enable, non-zero = disabled |
| DDT | Instruction for this CPU to enter EDDT |
| EDV | Code "EDV" \| Length of EXEC data vector |
| ED1 | Adr of address swapping block |
| ED2 | Relocated contents of .JBSYM |
| ED3 | Relocated contents of .JBUSY |
| ED4 | Address of word for DDT to use |
| ED5 | CPU/paging hardware data |
| ED6 | Physical address of this CPU's EPT |
| ED7 | Physical address of this CPU's SPT |
| ED0 | Physical address of this CPU's CST |
| HSF | Word for DDT to use |
| SYB | Length of address swapping block |
| SY1 | Number of words to swap |
| SY2 | Address of first word to swap |
| SY3 | Place where new map may be found |
| SY4 | Place to save old contents |
| SPC | Stopcode PC flags are saved here during DIE routine |
| SP1 | Stopcode PC |
| SP2 | New PC flags |
| SP3 | JSR entry point into DIE routine |

CPU Data Block -- Variables Area

The variables area of the CDB has traditional, six-character symbols, used only in the CDB for CPU0. Some of these symbols are obsolete, and have been superseded by symbols in the form .Cnxxx (where n is the CPU number). For words that have both types of symbols, the traditional symbol is listed below the .Cnxxx symbol, in parentheses.

| Symbol | Map |
|---|---|
| VBG (CORMAX) | Size of largest user program (in words) |
| CORLST | Pointer to last free block possible |
| CORTAL | Free + dormant + idle core blocks |
| SHFWAT | Obsolete |
| HOLEF | Absolute address of lowest hole in core |
| UPT | Uptime for this CPU in clock ticks |
| SHFWRD | Obsolete |
| STUSER | Obsolete |
| HIGHJB | Highest job number currently assigned |
| CLRWRD | Number of words cleared by CLRCOR |
| LST (LSTWRD) | Lost time on this CPU in jiffies |
| MEMSIZ | Size of physical memory in words |
| TPE | Total parity errors detected on this CPU |
| SPE | Total spurious parity errors (did not recur) on this CPU |

| | |
|---|---|
| MPC | Total number of times this CPU continued after a parity error |
| MPA | Memory parity address of first bad address |
| MPW | Contents of first bad word found |
| MPP | Memory parity PC exclusive of parity sweep |
| EPOCNT | Number of PDL overflows at UUO level not recovered |
| EPOREC | Number of PDL overflows at UUO level recorded |
| MAXMAX | Highest legal value of CORMAX |
| SYSKTM | Count-down timer for SET KSYS command |
| CORMIN | Lower bound on CORMAX |
| ABC | Address break count on this CPU |
| ABA | Address break address on this CPU |
| LJR | Last job run on this CPU |
| ODA | Obsolete (3 words) |
| STS | Stop timesharing on this CPU. Contains job no. that did TRPSET or RECON. UUO. |
| RUN | Operator-controlled scheduling bit for this CPU. (Refer to GETTAB word %CVRUN for bit definitions) |
| NUL | Null time for this CPU in jiffies |
| EDI | No. of Exec "don't care" interrupts |
| JOB | Current job on this CPU |
| OHT | Overhead time for this CPU in jiffies |
| EVM | Max. amount of exec space for jobs mapped in exec mode by LOCK UUO. |

| | |
|---|---|
| EVU | Total exec virt addr. space currently being used to map user segments |
| LLC | Number of times this CPU has looped waiting for other CPU |
| TUC | Total number of UUOs on this CPU |
| TJC | Total job context-switches |
| TNE | Total NXMs |
| SNE | Total non-reproducible NXMs |
| NJA | Total jobs crashed this NXM |
| MNA | First address found with NXM |
| EBJ | EBOX ticks/jiffy |
| MBJ | MBOX ticks/jiffy |
| PBA | Physical address with bad parity on last parity trap |
| TBD | Contents of bad word on last AR/ARX parity trap |
| TGD | Good contents of word after recovery from parity trap |
| NPT | Total no. of AR/ARX parity traps |
| AER | Results of RDERA on last parity/NXM interrupt |
| PEF | Results of CONI APR, on parity interrupt |
| PSB | Obsolete (4 words) |
| PPC | PC on last AR/ARX parity trap |
| PFW | Page fail word on last AR/ARX parity trap |
| HPT | No. hard AR/ARX parity traps |
| SAR | No. soft AR/ARX parity traps |
| PTP | No. page table parity traps |
| CSN | No. cache sweeps started |

| | |
|---|---|
| CLN | No. of times scheduler skipped a job because the job needed a cache sweep on another CPU |
| CLT | No. of jiffies CPU ran null job waiting for cache sweep |
| CSD | No. of times swap-out had to wait for cache sweep |
| CRN | Cache sweep request sweep count |
| CEC | No. non-recoverable AR/ARX parity errors involving cache |
| PTR | Retry word for AR/ARX parity trap |
| TSD | Obsolete |
| REP | Parity error/NXM reporting flag<br>0 = report NXM error, -1 = report parity error |
| NDB | Number of times this CPU's doorbell was rung |
| SBR | Status blocks read on this CPU (see SR.xxx in S.MAC)<br>Unused \| Bit settings |
| BPF | Background performance analysis timer.<br>If meter is running, contains negative of the number of clock ticks to next performance analysis update. |
| FBI | File blocks input (read) |
| FBO | File blocks output (written) |
| SBI | Swapping blocks input (read) |
| SBO | Swapping blocks output (written) |
| SNC | Number of CPU stopcodes |
| SND | Number of DEBUG stopcodes |
| SNJ | Number of JOB stopcodes |
| SJN | Job number as last stopcode |
| SNM | Name of last stopcode \| PC+1 of last stopcode |
| SPN | Program running at last stopcode |
| SPP | PPN of user at last stopcode |

| | |
|---|---|
| STN | TTY name at last stopcode |
| SUP | User PC at time of last stopcode |
| SUU | UUO at time of last stopcode |
| EJN | Job number at last parity/NXM error |
| EPN | Program at last parity/NXM error |
| PPI | Results of CONI PI, on parity/NXM interrupt |
| TPI | Results of CONI PI, on page fail trap |
| RQS | Number of times scheduler interlock was requested when not owned. |
| TFI | Number of tape frames read on this CPU |
| TFO | Number of tape frames written on this CPU |
| SNI | Number of stopcodes that did not dump (Events) |
| | / Response subtable (1) / |
| | / Memory parity subtable (1) / |
| | / Memory NXM subtable (1) / |
| | / CPU status block subtable (1) / |
| | / SBDIAG status block subtable (1) / |
| | / Device status block subtable (1) / |
| | / KL background performance analysis subtable (1) / |
| ADR (JOBADR) | Same as JBTADR (J) for current job |
| REL | Highest rel addr for current user |
| PC | / Job PC when scheduler is called (2 words) / |
| XTM | Time of last switch from monitor cycle to user job or back |
| LS2 | Additional lost time (fractional jiffy) |
| NL2 | Additional null time (fractional jiffy) |

| | |
|---|---|
| OH2 | Additional overhead time (fractional jiffy) |
| TNT | Time interval since last at clock level |
| HTM | Hung device time check |
| SEC | Seconds left before doing once-a-minute code |
| RCU | / Count of realtime CONSOs of skip chain (6 words) / |
| RIT | / Realtime initialization table (6 words) / |
| DMI | Realtime dismiss instruction |
| CKF (2)<br>(CLKFLG) | Non-zero when CLK interrupt (PI 7) requested |
| TMF (3)<br>(TIMEF) | Non-zero when APR clock ticked |
| SCF (4) | Force scheduling from exec mode |
| CHT | Flag to remember clock has ticked.<br>Used to call queued I/O protocol routines. |
| RTF<br>SCDRTF | Non-zero when realtime reschedule is required |
| ISF | "In-scheduler" flag |
| SUD | Address of scan table used during last scheduler scan |
| HQU | Non-zero if current job needs to be rescheduled for HPQ UUO |
| PLT | Flag set if current clock tick is potentially lost time |
| CLF | Flag set if current clock tick is potentially lost time<br>due to the state of the stack. |
| CL2 | Low order cache lost time |
| SDA | Number cache sweeps for core deallocation |

| | |
|---|---|
| CSR | Cache sweep request flag for this CPU |
| AEF | APR error flag |
| SAC | Saved copy of .CPAEF |
| APC | Current PC on detecting APR error (2 words) |
| MDP | Memory parity error double-word PC |
| PPD | AR/ARX double-word PC |
| NJE | Error in null job if non-zero |
| SFC | Scheduler fairness count |
| SQF | Non-zero if current job from subqueues |
| APR | Current user address break conditions |
| IPI | Interval timer PI assignment |
| CN1 | CONSO mask for APR interrupts user wants to handle. |
| DTO | Last DATAO PAG done |
| SP | Place to save P on APR interrupt (2 words) |
| S17 | Place to save AC17 on CLK interrupt (2 words) |
| A17 | 20th (octal) word for storing ACs on parity trap |
| LPP | Last memory parity PC |
| LSB | Obsolete |
| LCI | Time of last parity/NXM interrupt caused by channel reference |

| | |
|---|---|
| PIP | Pointer to real interrupt PC |
| PSP | Parity/NXM sweep in progress on this CPU |
| CHE | Channel error reporting in progress on this CPU |
| TCX | Results of DATAI PAG, on error trap |
| TCT | Triad counter for 60Hz leap jiffies |
| PJB | Owner of performance meter (job no.) |
| MJB | Measured job of PERF. UUO. |
| MJ1 | Job enable condition |
| PMR | Non-zero means PERF. meter is running |
| PAE | Used to store PERF. analysis enables |
| PRQ | Flag used in testing and giving PERF. meter away |
| APS | Non-zero means ACCT and PERF. meters should be kept in sync |
| MMO | "Virtual PERF meter's" high order memory reference count |
| MM1 | Low order mem ref count |
| BPC | Background performance analysis (B.P.A.) sample interval (in ticks) |
| BPT | /B.P.A. saved RDTIME at start of current interval (2 words) / |
| TIM | Clock interrupt flag for KL10s |
| ETM | SOSN done on this location every minute, to make sure KL error chunks don't remain allocated |
| EAD | KL error chunk addresses<br>Addr. of last KL error chunk \| Addr. of first KL error chunk |

```
              |------------------------------------------------------------|
KPB           /            KLINIK parameter buffer (6 words)               /
              |------------------------------------------------------------|
20F           | Flags about RSX20F front end | Count of characters         |
              |                              | being output                |
              |------------------------------------------------------------|
20S           /        Space for incoming line speeds (2 words)            /
              |------------------------------------------------------------|
20B           / Buffer for 16-bit data to RSX20F (30 words)                /
              |------------------------------------------------------------|
PTH           |       Parity/NXM trap occurred during cache sweep          |
              |------------------------------------------------------------|
STE           |               RDERA contents on sweep trap                 |
              |------------------------------------------------------------|
PTF           | No. of page table parity traps allowed between clock ticks|
              | (Used to crash system if too many.)                        |
              |------------------------------------------------------------|
CA1           /              Power fail AC block 1 save area               /
              |------------------------------------------------------------|
CA2           /              Power fail AC block 2 save area               /
              |------------------------------------------------------------|
CA3           |              Power fail AC block 3 save area               /
              |------------------------------------------------------------|
CA4           /              Power fail AC block 4 save area               /
              |------------------------------------------------------------|
CTQ           |        SCNSER output queue header for CTY on this CPU      |
              |        (for KL10, is header for all RSX20F lines)          |
              |------------------------------------------------------------|
QUE           |     Queue of DDBs for I/O requests on other CPUs.          |
              |     Emptied into CPUDSQ once per tick                      |
              |------------------------------------------------------------|
SWP           |     Non-zero if swap request from another CPU              |
              |------------------------------------------------------------|
QND           |             Address of last DDB in .CPQUE                  |
              |------------------------------------------------------------|
SWD           |               Flag for FILIO cache sweeps                  |
              |------------------------------------------------------------|
DRQ           |            Disks on this CPU need requeuing.               |
              |  (-1 if disk requests for this CPU need to be requeued     |
              |        to another CPU because this one is dead.)           |
              |------------------------------------------------------------|
```

| | |
|---|---|
| TAP | Tape waiting for cache sweep.<br>0 if no tape waiting for sweep<br>-1 if tape I/O waiting for sweep<br>0,,-1 if tape waiting, DSKTIC did sweep |
| PIB | Save PI state for NBFOFF |
| PIS | Save PI state for SYSPIN |
| DPI | Save PI state for DEVPIN |
| BTI | Save PI state for BTSOFF |
| IUT | Uptime a second ago |
| NTF | NETSER software interrupt flag |
| QTS | / QUESER variables (23 words) / |
| CPG | Result of DATAI PAG, done by SVSETS with bits set<br>so that DATAO will restore current AC set |
| ACA | Address of 20 word block where SVSETS saved the AC set |
| KPM | PM.KPM is set if MCA25 |
| JCH | Job/context handle for the current job on this CPU |
| CHN | / Addr. of channel data block (CHN) for internal channels /<br>/ KL10 only (8 words) / |
| PAT | / Patch space (here to next page boundary) / |

Notes:

1. The subtables are defined by GETTAB symbols; their contents are listed in the TOPS-10 Monitor Calls Manual.

2. CKF-CLKFLG takes the place of a hardware interrupt flag.

3. TMF-TIMFF is set so clock interrupt routines will know another jiffy has passed.

4. Normally, clock interrupts are dismissed when they interrupt exec mode. SCF is set after monitor detects an error and desires to force rescheduling.

5. K?SER refers to the processor-specific module (KLSER, KSSER, KISER).

## 7 CB -- CONNECTION BLOCK

Description: Contains information describing a connection to an application on a CI node.

Connection blocks are created and destroyed by SCASER as connections to applications on CI nodes are opened and closed.

Defined in: SCAPRM

Used by: KLPSER, MSCCOM, RAXKON, SCASER, SCSUUO

See also: PB, SB, PCB

| Symbol | Map |
| --- | --- |
| .CBANB | Address of next connection block |
| .CBAPB | Address of previous connection block |
| .CBPBK | Address of path block (PB) |
| .CBSTS | Status information (1) |
| .CBFLG | Flags (2) |
| .CBSCI | Source connect I.D. |
| .CBDCI | Destination connect I.D. |
| .CBADR | Address of routine to call when condition changes |
| .CBBUF | Number of message and datagram buffers to queue |
| .CBNWQ | Address of next entry on work queue |
| .CBSPN | Source process name |
| .CBDPN | Destination process name |
| .CBDTA | / Connection data (varying length) / |
| .CBREA | Destination and source disconnect reasons |
| .CBRSP | Expected response |
| .CBMCD | Minimum send and receive credits |
| .CBSCD | Send credit |
| .CBRCD | Receive credit |

| | |
|---|---|
| .CBPRC | Pending receive credit |
| .CBRQC | Credits outstanding |
| .CBRTC | Return credit |
| .CBNPO | Number of packets in port command queue |
| .CBDGR | Number of datagram buffers on hardware queue |
| .CBCDD | Number of dropped datagrams |
| .CBLCK | Interlock word |
| .CBPND | Flag word for credit requests in progress |
| .CBJNB | Address of next CB in the job list |
| .CBJPB | Address of previous CB in the job list |
| .CBMGJ | Number of UUO message receive buffers queued |
| .CBDGJ | Number of datagram buffers in user space |
| .CBJCH | Job number and JCH of job owning the connection |
| .CBTMQ | Pointer to top of message pending queue |
| .CBBMQ | Pointer to bottom of message pending queue |
| .CBTDQ | Pointer to top of datagram available queue |
| .CBBDQ | Pointer to bottom of datagram available queue |
| .CBTXQ | Pointer to top of data request complete queue |
| .CBBXQ | Pointer to bottom of data request complete queue |
| .CBTEQ | Pointer to top of event queue |
| .CBBEQ | Pointer to bottom of event queue |
| .CBTBQ | Pointer to first buffer list descriptor block |
| .CBBBQ | Pointer to last buffer list descriptor block |

Notes:

1. .CBSTS contains the connections block state in the left half, and the connection state in the right half.

2. The contents of .CBFLG consist of bits with the following meanings:

| Bit | Symbol | Meaning |
|---|---|---|
| 0 | CB.NCC | Needs credit notify |
| 1 | CB.JSY | CB is for UUO connection |
| 2 | CB.ABT | CB has been aborted |
| 3 | CB.RAP | CB has to be reaped |
| 4 | CB.DCL | This was a "don't-care" listener |
| 5 | CB.KIL | Fork has been killed |
| 6 | CB.MDC | Maintenance data CB |
| 7 | CB.CVC | VC was closed |
| 8 | CB.SOB | Stuck on buffers |
| 9 | CB.PTC | Protocol complete |
| 10 | CB.ERR | SC.ERR deferred |
| 11 | CB.DIS | SC.DIS deferred |
| 12 | CB.DRQ | SC.DRQ deferred |
| 13 | CB.SNM | SC.SNM deferred |
| 10-13 | CB.DEF | All deferred bits |

## 8 CDT -- CONNECTED DEVICE TABLE

Description: Used to point to DDBs of devices connected to the Multiplexed I/O Facility (MPX). This table contains the connected device's UDX number and DDB address and is sorted by UDX number.

The left half of the DEVXTR word of the MPX DDB points to the CDT table.

Defined in: MSGSER

Used by: MSGSER

Map

| | |
|---|---|
| Size of table | No. of free slots |
| UDX 1 | DDB addr. |
| UDX 2 | DDB addr. |
| UDX 2 | DDB addr. |

. . .

## 9 CHKTAB -- UUO CHECK BIT TABLE

Description: Contains bits for checking UUOs that can be executed on any CPU (UU.CP1), for checking effective address (UU.EA) and for flagging LOOKUP, ENTER, and RENAME UUOs (UU.LER).

The table is in the same order as UUOTAB so the UUOTAB index in AC T2 serves to get the CHKTAB entry.

If a check bit is specified for the requested UUO, UUOCON goes to the UUOCHK routine in VMSER.

Defined in: UUOCON

Map

| | |
|---|---|
| Check bit for 40 | Check bit for 41 |
| 0 | 0 |
| 0 | 0 |
| 0 | Check bit for 47 |

. . .

Because this table is accessed in half words, the bit definitions are equivalent for the right half and left half words. The following bit definitions are based on the number of bits in the half word.

| Bits | Symbol | Description |
|---|---|---|
| 0<br>18 | UU.CP1 | UUO can be executed on any CPU. |
| 7<br>25 | UU.LER | Argument list is a LOOKUP/ENTER/RENAME block. |
| 11<br>29 | UU.EA | Check effective address. |

## 10 CHKTBC -- CALL AND CALLI UUO CHECK BIT TABLE

Description: Contains validity check bits for CALL and CALLI UUOs. This table is in the same order as UCLJMP, so the UCLJMP index in AC T2 serves to get the CHKTBC entry. If a check bit is specified for the requested UUO, UUOCON goes to the UUOCHK routine in VMSER.

Defined in: UUOCON

Used by: UUOCON, CPNSER

```
Map
+-------------------------------------------------------------+
|    Check bits for CALLI 0    |      Check bits for CALLI 1  |
|-------------------------------------------------------------|
|    Check bits for CALLI 2    |      Check bits for CALLI 3  |
|-------------------------------------------------------------|
|    Check bits for CALLI 4    |      Check bits for CALLI 5  |
|-------------------------------------------------------------|
                                   .   .   .
```

Bit Definitions:

Because the table is organized as halfwords, the following bit definitions apply for both left halfwords and right halfwords

```
+----------------------------------------------------------------+
|CP1|   |   |NCL|WCC|NAL|LFT|LER|MNS|CEA|CAC| EA| Arg. list length  |
+----------------------------------------------------------------+
```

| Bits | Symbol | Description |
|---|---|---|
| 0<br>18 | UU.CP1 | UUO can be executed on any CPU. |
| 1<br>19 | UU.SE1 | UUO is executed in Section 1. |
| 3<br>21 | UU.NCL | Negative argument (repeat) count is legal. |
| 4<br>22 | UU.WCC | Working set can change. |
| 5<br>23 | UU.NAL | Not allowed if locked. |
| 6<br>24 | UU.LFT | Use left half of UUO for argument count. |
| 7 | UU.LER | Argument list is a LOOKUP/ENTER/RENAME block. |

| | | |
|---|---|---|
| 25 | | |
| 8 | UU.MNS | User-supplied argument list length is a negative value. |
| 26 | | |
| 9 | UU.CEA | Use contents of effective address as list length. |
| 27 | | |
| 10 | UU.CAC | Use contents of UUO ac as list length (modifiable using UU.LFT). |
| 28 | | |
| 11 | UU.EA | Check effective address. |
| 29 | | |
| 30-35 | | Length of user's argument list. |

If neither UUO.CEA nor UU.CAC is set, the list length is taken from the CHKTBC table.

11 CHN -- CHANNEL DATA BLOCK

Description: Contains information pertaining to all devices on one data channel.

One Channel Data Block is generated dynamically by AUTCON for each channel when the system is started and when a new channel comes on-line.

Defined in: COMMOD

Used by: AUTCON, ERRCON, FILFND, FILIO, KLSER, RNXKON, RPXKON, SYSINI, T78KON, TAPUUO, TDZKON, TMXKON, TMZKON, TX1KON

The format of the prototype CHN is illustrated in the following table. Where a different symbol is defined for right and left halfwords, the left half symbol is followed by two commas (,,) and the right half symbol is listed below it.

| Symbol | Map | |
|---|---|---|
| .CHNBSY | Number pending requests on this channel ( -1 is idle) | |
| .CHSYS,,<br>.CHLUE | ADR of next CHN<br>0 if last | Last UDB with error |
| .CHICW | Initial control word on last error | |
| .CHFCW | Final control word after last error | |
| .CHCW2 | Command word -2 on last error | |
| .CHCW1 | Command word -1 on last error | |
| .CHCW0 | Command word on last error | |
| .CHDWZ | Data word -2 on last error | |
| .CHDW1 | Data word -1 on last error | |
| .CHDW0 | Data word on last error | |
| .CHMPE | Number of memory parity errors | |
| .CHDPE | Number of data parity errors (from device) | |
| .CHNXM | Number of non-existent memory errors or data | |

| | | |
|---|---|---|
| .CHCSR,, .CHLDE | Bits to request CPU to sweep core | Last DDB Address |
| .CHTCW | Expected termination control word of last error | |

(The remaining words apply only to disk channels.)

| | | |
|---|---|---|
| CHNECT | Error count on current data transfer | |
| CHNRCT | Current recalibrate count | |
| CHNQUE | First DDB in TWQ (2) | Job number of DDB |
| CHNIFP * | Initial fairness count for positioning | |
| CHNCFP | Current fairness count for positioning | |
| CHNIFT | Initial fairness count for transfers | |
| CHNCFT | Current fairness count for transfers | |
| CHNIFS | Initial swapping fairness count | |
| CHNCFS | Current swapping fairness count | |
| CHNNUM | Number of blocks currently being transferred | |
| CHNQUL | Length of XFER wait queue (TWQ) | |
| CHNTCW | Expected termination control word | |
| CHNCUA | Current unit active on channel | |
| CHNRSC (3) | Number of time DX20 on this channel was restarted | |
| CHNPCB | Addr. of Port Control Block for IPA-20 type channel | |

* CHNIFP bytes are described on the next page.

Notes:

1. The errors referred to in the Channel Data Block are both soft and hard errors.

2. TWQ = Transfer Wait Queue

3. CHNRSC applies only to internal channels.

CHNIFP -- Initial Fairness Count

```
        |-----------------------------------------------------------|
CHNIFP  |22B|DX1|RH2|R11|LP2|CD2|MX|KLP|KNI|      |SWF|     IFC    |
        |-----------------------------------------------------------|
          0   1   2   3   4   5  6   7   8  9-16  17  18--------35
```

| Word Symbol | Bit | Byte Symbol | Meaning |
|---|---|---|---|
| CHB22B | 0 | CP.22B | This is a 22-bit channel. |
| | 1 | CP.DX1 | This is a DX10 channel |
| | 2 | CP.RH2 | This is an RH20 channel. |
| | 3 | CP.R11 | This is a RH11. |
| | 4 | CP.LP2 | This is a LP20. |
| | 5 | CP.CD2 | This is a CD20. |
| | 6 | CP.MX | This channel can start multiple transfers at the same time (such as, CI disks). |
| | 7 | CP.KLP | This is a CI20 channel. |
| | 8 | CP.KNI | This is an NIA20 channel. |
| | 17 | CP.SWF | Cache sweep must be done before interrupt exit in FILIO. |
| CHNIFP | 18-35 | | Initial fairness count for positioning. |

12 CHTABL -- SPECIAL CHARACTER TABLE

Description: One table entry for each ASCII character, specifying characteristics and, in some cases, the address of a special action routine for processing the received character.

Indexed by the ASCII value of the character.

Defined in: SCNSER

Used by: SCNSER, PTYSER

Format of left half of each word:

```
|-------------------------------------------------------------------|
|SPO|   |PUNC|2PC|EPAR|VPOS|CRE|FILO|INVL|CNC|CRET|RIA|ALT|FIL|UAE|BRK|
|-------------------------------------------------------------------|
  0  1-3  4    5    6    7    8    9   10   11   12  13  14  15  16  17
```

| Bit | Label | Meaning |
|---|---|---|
| 0 | CHSPO | Requires special checking on output |
| 1-3 | | Undefined |
| 4 | CHPUNC | Punctuation character |
| 5 | CH2PC | 8-bit character that has a multi-character 7-bit expansion |
| 6 | CHEPAR | Character is even parity (see Note 1) |
| 7 | CHVPOS | Vertical positioning simulated with line feeds |
| 8 | CHCRE | Gets CRLF after its <CTRL/x> echo |
| 9 | CHFILO | Bit for output filler routine (not in table) |
| 10 | CHINVL | Reserved 9-bit ASCII character (should never be received) |
| 11 | CHCNC | This is <CTRL/C> |
| 12 | CHCRET | This is carriage return |
| 13 | CHRIA | RCV interrupt level action required (See Note 2) |
| 14 | CHALT | This is an altmode |
| 15 | CHFIL | Requires fillers at some speeds |
| 16 | CHUAE | Echoes as <CTRL/x> |
| 17 | CHBRK | This is a break character |

Right half of each word may contain the address of a routine to process the character upon receipt.

The format of CHTABL, the special character table, is described below. The character's ASCII code is followed by the printable characters (Char), the name of the character used in the code (Name), and the bit definitions for the character (Bits). The meanings of the bit settings are described on the previous page. Note that Bits 1-3 of each word are undefined.

| ASCII Code | Char | Name | 0 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18---35 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 000 | | NUL | 1 | 1 | | 1 | 1 | | | | | | 1 | | | | | RINUL |
| 001 | ^A | SOA | 1 | 1 | | | | | | | | | 1 | | | 1 | | RICA |
| 002 (2) | B | STX | 1 | 1 | | | | | | | | | 1 | | | 1 | | RICB |
| 003 | ^C | ETX | 1 | 1 | | 1 | | 1 | | | 1 | | 1 | | | 1 | 1 | RICC |
| 004 | ^D | EOT | 1 | 1 | | | | | | | | | 1 | | | 1 | | RICD |
| 005 | ^E | ENQ | 1 | 1 | | 1 | | | | | | | | | | 1 | | |
| 006 | ^F | ACK | 1 | 1 | | 1 | | | | | | | | | | 1 | | |
| 007 | ^G | BEL | 1 | 1 | | | | | | | | | | | | | 1 | |
| 010 | ^H | BS | 1 | 1 | | | | | | | | | 1 | | | | 1 | RIBSP |
| 011 | ^T | HT | 1 | 1 | | 1 | | | | | | | | | 1 | | | |
| 012 | ^J | LF | 1 | 1 | | 1 | | | | | | | | | 1 | | 1 | |
| 013 | ^K | VT | 1 | 1 | | | 1 | | | | | | | | 1 | | 1 | |
| 014 | ^L | FF | 1 | 1 | | 1 | 1 | | | | | | | | 1 | | 1 | |
| 015 | ^M | CR | 1 | 1 | | | | | | | | 1 | 1 | | 1 | | | RICM |
| 016 | ^N | SO | 1 | 1 | | | | | | | | | | | | 1 | | |
| 017 | ^O | IS | 1 | 1 | | 1 | | 1 | | | | | 1 | | | 1 | | RICO |
| 020 (2) | ^P | DLE | 1 | 1 | | | | | | | | | 1 | | | 1 | | RICP |
| 021 | ^Q | DC1 | 1 | 1 | | 1 | | | | | | | 1 | | 1 | 1 | | RICQ |
| 022 | ^R | DC2 | 1 | 1 | | 1 | | | | | | | 1 | | 1 | 1 | 1 | RICR |
| 023 | ^S | DC3 | 1 | 1 | | | | | | | | | 1 | | 1 | 1 | | RICS |
| 024 | ^T | DC4 | 1 | 1 | | 1 | | | | | | | 1 | | 1 | 1 | 1 | RICT |
| 025 | ^U | NAK | 1 | 1 | | | | 1 | | | | | 1 | | | 1 | 1 | RICU |
| 026 | ^V | SYN | 1 | 1 | | | | | | | | | | | | 1 | | |

| ASCII Code | Char | Name | 0 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18---35 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 027 | ^W | ETB | 1 | 1 | | 1 | | | | | | | 1 | | | 1 | 1 | RIDEL |
| 030 | ^X | CAN | 1 | 1 | | 1 | | | | | | | | | | 1 | | |
| 031 | ^Y | EM | 1 | 1 | | | | | | | | | | | | 1 | | |
| 032 | ^Z | SUB | 1 | 1 | | | | 1 | | | | | | | | 1 | 1 | |
| 033 | $ | ESC | 1 | 1 | | 1 | | | | | | | 1 | 1 | | | 1 | RIALT |
| 034 | ^\ | FS | 1 | 1 | | | | | | | | | | | | 1 | | |
| 035 | ^] | GS | 1 | 1 | | 1 | | | | | | | | | | 1 | | |
| 036 | ^^ | RS | 1 | 1 | | 1 | | | | | | | | | | 1 | | |
| 037 | ^_ | US | 1 | 1 | | | | | | | | | | | | 1 | | |
| 040 | (Space) | | | 1 | | | | | | | | | | | | | | |
| 041 | ! | | | 1 | | 1 | | | | | | | | | | | | |
| 042 | " | | | 1 | | 1 | | | | | | | | | | | | |
| 043 | # | | | 1 | | | | | | | | | | | | | | |
| 044 | $ | | | 1 | | 1 | | | | | | | | | | | | |
| 045 | % | | | 1 | | | | | | | | | | | | | | |
| 046 | & | | | 1 | | | | | | | | | | | | | | |
| 047 | ' | | | 1 | | 1 | | | | | | | | | | | | |
| 050 | ( | | | 1 | | 1 | | | | | | | | | | | | |
| 051 | ) | | | 1 | | | | | | | | | | | | | | |
| 052 | * | | | 1 | | | | | | | | | | | | | | |
| 053 | + | | | 1 | | 1 | | | | | | | | | | | | |
| 054 | , | | | 1 | | | | | | | | | | | | | | |
| 055 | - | | | 1 | | 1 | | | | | | | | | | | | |

| ASCII Code | Char | Name | Map (Bits) | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 0 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18---35 |
| 056 | . | | | 1 | | 1 | | | | | | | | | | | | |
| 057 | / | | | 1 | | | | | | | | | | | | | | |
| 060 | 0 | | | | | 1 | | | | | | | | | | | | |
| 061 | 1 | | | | | | | | | | | | | | | | | |
| 062 | 2 | | | | | | | | | | | | | | | | | |
| 063 | 3 | | | | | 1 | | | | | | | | | | | | |
| 064 | 4 | | | | | | | | | | | | | | | | | |
| 065 | 5 | | | | | 1 | | | | | | | | | | | | |
| 066 | 6 | | | | | 1 | | | | | | | | | | | | |
| 067 | 7 | | | | | | | | | | | | | | | | | |
| 070 | 8 | | | | | | | | | | | | | | | | | |
| 071 | 9 | | | | | 1 | | | | | | | | | | | | |
| 072 | : | | | 1 | | 1 | | | | | | | | | | | | |
| 073 | ; | | | 1 | | | | | | | | | | | | | | |
| 074 | < | | | 1 | | 1 | | | | | | | | | | | | |
| 075 | = | | | 1 | | | | | | | | | | | | | | |
| 076 | > | | | 1 | | | | | | | | | | | | | | |
| 077 | ? | | | 1 | | 1 | | | | | | | | | | | | |
| 100 | @ | | | 1 | | | | | | | | | | | | | | |
| 101 | A | | | | | 1 | | | | | | | | | | | | |
| 102 | B | | | | | 1 | | | | | | | | | | | | |
| 103 | C | | | | | | | | | | | | | | | | | |
| 104 | D | | | | | 1 | | | | | | | | | | | | |
| 105 | E | | | | | | | | | | | | | | | | | |

106 F | | | | | | | | | | | | | | | | |
|--------------------------------------------------|

| ASCII Code | Char | Name | Map (Bits) | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 0 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18---35 |
| 107 | G | | | | | 1 | | | | | | | | | | | | |
| 110 | H | | | | | 1 | | | | | | | | | | | | |
| 111 | I | | | | | | | | | | | | | | | | | |
| 112 | J | | | | | | | | | | | | | | | | | |
| 113 | K | | | | | 1 | | | | | | | | | | | | |
| 114 | L | | | | | | | | | | | | | | | | | |
| 115 | M | | | | | 1 | | | | | | | | | | | | |
| 116 | N | | | | | 1 | | | | | | | | | | | | |
| 117 | O | | | | | | | | | | | | | | | | | |
| 120 | P | | | | | 1 | | | | | | | | | | | | |
| 121 | Q | | | | | | | | | | | | | | | | | |
| 122 | R | | | | | | | | | | | | | | | | | |
| 123 | S | | | | | 1 | | | | | | | | | | | | |
| 124 | T | | | | | | | | | | | | | | | | | |
| 125 | U | | | | | 1 | | | | | | | | | | | | |
| 126 | V | | | | | 1 | | | | | | | | | | | | |
| 127 | W | | | | | | | | | | | | | | | | | |
| 130 | X | | | | | | | | | | | | | | | | | |
| 131 | Y | | | | | 1 | | | | | | | | | | | | |
| 132 | Z | | | | | 1 | | | | | | | | | | | | |
| 133 | [ | | | 1 | | | | | | | | | | | | | | |
| 134 | \ | | | 1 | | 1 | | | | | | | | | | | | |
| 135 | ] | | | 1 | | | | | | | | | | | | | | |

| ASCII Code | Char | Name | 0 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18---35 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 136 | ^ | | | 1 | | | | | | | | | | | | | | |
| 137 | _ | | | 1 | | 1 | | | | | | | | | | | | |
| 140 | ` | | | 1 | | 1 | | | | | | | | | | | | |
| 141 | a | | | | | | | | | | | | | | | | | |
| 142 | b | | | | | | | | | | | | | | | | | |
| 143 | c | | | | | 1 | | | | | | | | | | | | |
| 144 | d | | | | | | | | | | | | | | | | | |
| 145 | e | | | | | 1 | | | | | | | | | | | | |
| 146 | f | | | | | 1 | | | | | | | | | | | | |
| 147 | g | | | | | | | | | | | | | | | | | |
| 150 | h | | | | | | | | | | | | | | | | | |
| 151 | i | | | | | 1 | | | | | | | | | | | | |
| 152 | j | | | | | 1 | | | | | | | | | | | | |
| 153 | k | | | | | | | | | | | | | | | | | |
| 154 | l | | | | | 1 | | | | | | | | | | | | |
| 155 | m | | | | | | | | | | | | | | | | | |
| 156 | n | | | | | | | | | | | | | | | | | |
| 157 | o | | | | | 1 | | | | | | | | | | | | |
| 160 | p | | | | | | | | | | | | | | | | | |
| 161 | q | | | | | 1 | | | | | | | | | | | | |
| 162 | r | | | | | 1 | | | | | | | | | | | | |
| 163 | s | | | | | | | | | | | | | | | | | |
| 164 | t | | | | | 1 | | | | | | | | | | | | |
| 165 | u | | | | | | | | | | | | | | | | | |

| ASCII Code | Char | Name | 0 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18---35 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 166 | v | | | | | | | | | | | | | | | | | |
| 167 | w | | | | | 1 | | | | | | | | | | | | |
| 170 | x | | | | | 1 | | | | | | | | | | | | |
| 171 | y | | | | | | | | | | | | | | | | | |
| 172 | z | | | | | | | | | | | | | | | | | |
| 173 | { | | | 1 | | 1 | | | | | | | | | | | | |
| 174 | \| | | | 1 | | | | | | | | | | | | | | |
| 175 (3) | } | | 1 | 1 | | 1 | | | | | | | 1 | 1 | | | | RIALT |
| 176 (3) | ~ | | 1 | 1 | | 1 | | | | | | | 1 | 1 | | | | RIALT |
| 177 | | DEL | 1 | 1 | | | | | | | | | 1 | | | | 1 | RIDEL |
| 200 | (reserved) | | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 201 | (reserved) | | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 202 | (reserved) | | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 203 | (reserved) | | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 204 | | IND | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 205 | | NEL | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 206 | | SSA | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 207 | | ESA | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 210 | | HTS | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 211 | | HTJ | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 212 | | VTS | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 213 | | PLD | 1 | 1 | 1 | | | | | | | | | | | | 1 | |

| ASCII Code | Char | Name | Map (Bits) | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 0 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18---35 |
| 214 | | PLU | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 215 | | RI | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 216 | | SS2 | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 217 | | SS3 | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 220 | | DCS | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 221 | | PU1 | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 222 | | PU2 | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 223 | | STS | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 224 | | CCH | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 225 | | MW | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 226 | | SPA | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 227 | | EPA | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 230 | (reserved) | | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 231 | (reserved) | | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 232 | (reserved) | | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 233 | | CSI | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 234 | | ST | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 235 | | OSC | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 236 | | PM | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 237 | | APC | 1 | 1 | 1 | | | | | | | | | | | | 1 | |
| 240 | (reserved) | | 1 | 1 | 1 | | | | | 1 | | | | | | | 1 | |
| 241 | | SP03 | | 1 | | | | | | | | | | | | | | |
| 242 | | SC04 | | 1 | 1 | | | | | | | | | | | | | |

| ASCII Code | Char | Name | 0 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18---35 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 243 | | SC02 | | 1 | 1 | | | | | | | | | | | | | |
| 244 | (reserved) | | 1 | 1 | 1 | | | | | 1 | | | | | | | 1 | |
| 245 | | SC05 | | 1 | 1 | | | | | | | | | | | | | |
| 246 | (reserved) | | 1 | 1 | 1 | | | | | 1 | | | | | | | 1 | |
| 247 | | SM24 | | 1 | 1 | | | | | | | | | | | | | |
| 250 | | SC01 | | 1 | 1 | | | | | | | | | | | | | |
| 251 | | SM52 | | 1 | 1 | | | | | | | | | | | | | |
| 252 | | SM21 | | 1 | 1 | | | | | | | | | | | | | |
| 253 | | SP17 | | 1 | 1 | | | | | | | | | | | | | |
| 254 | (reserved) | | 1 | 1 | 1 | | | | | 1 | | | | | | | 1 | |
| 255 | (reserved) | | 1 | 1 | 1 | | | | | 1 | | | | | | | 1 | |
| 256 | (reserved) | | 1 | 1 | 1 | | | | | 1 | | | | | | | 1 | |
| 257 | (reserved) | | 1 | 1 | 1 | | | | | 1 | | | | | | | 1 | |
| 260 | | SM19 | | 1 | | | | | | | | | | | | | | |
| 261 | | SA02 | | 1 | 1 | | | | | | | | | | | | | |
| 262 | | NS02 | | 1 | | | | | | | | | | | | | | |
| 263 | | NS03 | | 1 | | | | | | | | | | | | | | |
| 264 | (reserved) | | 1 | 1 | 1 | | | | | 1 | | | | | | | 1 | |
| 265 | | SM17 | | 1 | | | | | | | | | | | | | | |
| 266 | | SM25 | | 1 | 1 | | | | | | | | | | | | | |
| 267 | | SM26 | | 1 | | | | | | | | | | | | | | |
| 270 | (reserved) | | 1 | 1 | 1 | | | | | 1 | | | | | | | 1 | |

| ASCII Code | Char | Name | 0 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18---35 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 271 | | NS01 | | 1 | | | | | | | | | | | | | | |
| 272 | | SM20 | | 1 | 1 | | | | | | | | | | | | | |
| 273 | | SP18 | | 1 | 1 | | | | | | | | | | | | | |
| 274 | | NF04 | | 1 | 1 | | | | | | | | | | | | | |
| 275 | | NF01 | | 1 | 1 | | | | | | | | | | | | | |
| 276 | (reserved) | | 1 | 1 | 1 | | | | | 1 | | | | | | | 1 | |
| 277 | | SP16 | | 1 | | | | | | | | | | | | | | |
| 300 | | LA14 | | | | | | | | | | | | | | | | |
| 301 | | LA12 | | | | | | | | | | | | | | | | |
| 302 | | LA16 | | | | | | | | | | | | | | | | |
| 303 | | LA20 | | | | | | | | | | | | | | | | |
| 304 | | LA18 | | | | | | | | | | | | | | | | |
| 305 | | LA28 | | | | | | | | | | | | | | | | |
| 306 | | LA52 | | | 1 | | | | | | | | | | | | | |
| 307 | | LC42 | | | 1 | | | | | | | | | | | | | |
| 310 | | LE14 | | | | | | | | | | | | | | | | |
| 311 | | LE12 | | | | | | | | | | | | | | | | |
| 312 | | LE16 | | | | | | | | | | | | | | | | |
| 313 | | LE18 | | | | | | | | | | | | | | | | |
| 314 | | LI14 | | | | | | | | | | | | | | | | |
| 315 | | LI12 | | | | | | | | | | | | | | | | |
| 316 | | LI16 | | | | | | | | | | | | | | | | |
| 317 | | LI18 | | | | | | | | | | | | | | | | |
| 320 | (reserved) | | 1 | 1 | 1 | | | | | 1 | | | | | | | 1 | |

```
321          LN20  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |
                   |------------------------------------------------|
```

| ASCII Code | Char | Name | 0 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18---35 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 322 | | L014 | | | | | | | | | | | | | | | | |
| 323 | | L012 | | | | | | | | | | | | | | | | |
| 324 | | L016 | | | | | | | | | | | | | | | | |
| 325 | | L020 | | | | | | | | | | | | | | | | |
| 326 | | L018 | | | | | | | | | | | | | | | | |
| 327 | | L052 | | | 1 | | | | | | | | | | | | | |
| 330 | | L062 | | | 1 | | | | | | | | | | | | | |
| 331 | | LU14 | | | | | | | | | | | | | | | | |
| 332 | | LU12 | | | | | | | | | | | | | | | | |
| 333 | | LU16 | | | | | | | | | | | | | | | | |
| 334 | | LU18 | | | | | | | | | | | | | | | | |
| 335 | | LY18 | | | | | | | | | | | | | | | | |
| 336 | (reserved) | | 1 | 1 | 1 | | | | | 1 | | | | | | | 1 | |
| 337 | | LS61 | | | 1 | | | | | | | | | | | | | |
| 340 | | LA13 | | | 1 | | | | | | | | | | | | | |
| 341 | | LA11 | | | 1 | | | | | | | | | | | | | |
| 342 | | LA15 | | | 1 | | | | | | | | | | | | | |
| 343 | | LA19 | | | 1 | | | | | | | | | | | | | |
| 344 | | LA17 | | | 1 | | | | | | | | | | | | | |
| 345 | | LA27 | | | | | | | | | | | | | | | | |
| 346 | | LA51 | | | 1 | | | | | | | | | | | | | |
| 347 | | LC41 | | | 1 | | | | | | | | | | | | | |
| 350 | | LE13 | | | 1 | | | | | | | | | | | | | |
| 351 | | LE11 | | | 1 | | | | | | | | | | | | | |

| ASCII Code | Char | Name | 0 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18---35 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 352 | | LE15 | | | 1 | | | | | | | | | | | | | |
| 353 | | LE17 | | | 1 | | | | | | | | | | | | | |
| 354 | | LI13 | | | 1 | | | | | | | | | | | | | |
| 355 | | LI11 | | | 1 | | | | | | | | | | | | | |
| 356 | | LI15 | | | 1 | | | | | | | | | | | | | |
| 357 | | LI17 | | | 1 | | | | | | | | | | | | | |
| 360 | (reserved) | | 1 | 1 | 1 | | | | | 1 | | | | | | | 1 | |
| 361 | | LN19 | | | 1 | | | | | | | | | | | | | |
| 362 | | LO13 | | | 1 | | | | | | | | | | | | | |
| 363 | | LO11 | | | 1 | | | | | | | | | | | | | |
| 364 | | LO15 | | | 1 | | | | | | | | | | | | | |
| 365 | | LO19 | | | 1 | | | | | | | | | | | | | |
| 366 | | LO17 | | | 1 | | | | | | | | | | | | | |
| 367 | | LO51 | | | 1 | | | | | | | | | | | | | |
| 370 | | LO61 | | | 1 | | | | | | | | | | | | | |
| 371 | | LU13 | | | 1 | | | | | | | | | | | | | |
| 372 | | LU11 | | | 1 | | | | | | | | | | | | | |
| 373 | | LU15 | | | 1 | | | | | | | | | | | | | |
| 374 | | LU17 | | | 1 | | | | | | | | | | | | | |
| 375 | | LY17 | | | 1 | | | | | | | | | | | | | |
| 376 | (reserved) | | 1 | 1 | 1 | | | | | 1 | | | | | | | 1 | |
| 377 | (reserved) | | 1 | 1 | 1 | | | | | 1 | | | | | | | 1 | |

Notes:

1. CHEPAR is set for entries whose ASCII character code contains an even number of 1 bits.

2. Entries 2 and 20 will have CHRIA and the dispatch address set only if FTMIC is non-zero.

3. Character codes 175 (}) and 176 (~) are converted to ASCII code 33 (ESC) only if SET TTY ALTMODE is in effect.

## 13 CHREQV -- CHARACTER EQUIVALENCE TABLE

Description: Used by SCNSER to translate 8-bit ASCII characters to 7-bit ASCII. This table is indexed by character, where the offset is obtained from CHTABL, and is used for translation only when the octal code is between 200-377.

The Equivalence Tables (CHREQV and METEQV) are formatted in 9-bit bytes, each byte containing an octal code for a 7-bit ASCII characters. The bytes are read in reverse. Therefore, they are listed from left to right in the following table.

Defined in: SCNSER

Used by: SCNSER

| ASCII Code | Name | Byte 4 | Byte 3 | Byte 2 | Byte 1 |
|---|---|---|---|---|---|
| 200 | (reserved) | 033 | 100 | 000 | 000 |
| 201 | (reserved) | 033 | 101 | 000 | 000 |
| 202 | (reserved) | 033 | 102 | 000 | 000 |
| 203 | (reserved) | 033 | 103 | 000 | 000 |
| 204 | IND | 033 | 104 | 000 | 000 |
| 205 | NEL | 033 | 105 | 000 | 000 |
| 206 | SSA | 033 | 106 | 000 | 000 |
| 207 | ESA | 033 | 107 | 000 | 000 |
| 210 | HTS | 033 | 110 | 000 | 000 |
| 211 | HTJ | 033 | 111 | 000 | 000 |
| 212 | VTS | 033 | 112 | 000 | 000 |
| 213 | PLD | 033 | 113 | 000 | 000 |
| 214 | PLU | 033 | 114 | 000 | 000 |
| 215 | RI | 033 | 115 | 000 | 000 |
| 216 | SS2 | 033 | 116 | 000 | 000 |

| ASCII Code | Name | Byte 4 | Byte 3 | Byte 2 | Byte 1 |
|---|---|---|---|---|---|
| 217 | SS3 | 033 | 117 | 000 | 000 |
| 220 | DCS | 033 | 120 | 000 | 000 |
| 221 | PU1 | 033 | 121 | 000 | 000 |
| 222 | PU2 | 033 | 122 | 000 | 000 |
| 223 | STS | 033 | 123 | 000 | 000 |
| 224 | CCH | 033 | 124 | 000 | 000 |
| 225 | MW | 033 | 125 | 000 | 000 |
| 226 | SPA | 033 | 126 | 000 | 000 |
| 227 | EPA | 033 | 127 | 000 | 000 |
| 230 | (reserved) | 033 | 130 | 000 | 000 |
| 231 | (reserved) | 033 | 131 | 000 | 000 |
| 232 | (reserved) | 033 | 132 | 000 | 000 |
| 233 | CSI | 033 | 133 | 000 | 000 |
| 234 | ST | 033 | 134 | 000 | 000 |
| 235 | OSC | 033 | 135 | 000 | 000 |
| 236 | PM | 033 | 136 | 000 | 000 |
| 237 | APC | 033 | 137 | 000 | 000 |
| 240 | (reserved) | 137 | 000 | 000 | 000 |
| 241 | SP03 | 041 | 000 | 000 | 000 |
| 242 | SC04 | 174 | 010 | 143 | 000 |
| 243 | SC02 | 075 | 010 | 114 | 000 |
| 244 | (reserved) | 137 | 000 | 000 | 000 |

| ASCII Code | Name | Byte 4 | Byte 3 | Byte 2 | Byte 1 |
|---|---|---|---|---|---|
| 245 | SC05 | 075 | 010 | 131 | 000 |
| 246 | (reserved) | 137 | 000 | 000 | 000 |
| 247 | SM24 | 123 | 143 | 000 | 000 |
| 250 | SC01 | 170 | 010 | 117 | 000 |
| 251 | SM52 | 050 | 103 | 051 | 000 |
| 252 | SM21 | 137 | 010 | 141 | 000 |
| 253 | SP17 | 074 | 074 | 000 | 000 |
| 254 | (reserved) | 137 | 000 | 000 | 000 |
| 255 | (reserved) | 137 | 000 | 000 | 000 |
| 256 | (reserved) | 137 | 000 | 000 | 000 |
| 257 | (reserved) | 137 | 000 | 000 | 000 |
| 260 | SM19 | 157 | 000 | 000 | 000 |
| 261 | SA02 | 137 | 010 | 053 | 000 |
| 262 | NS02 | 062 | 000 | 000 | 000 |
| 263 | NS03 | 063 | 000 | 000 | 000 |
| 264 | (reserved) | 137 | 000 | 000 | 000 |
| 265 | SM17 | 165 | 000 | 000 | 000 |
| 266 | SM25 | 120 | 162 | 000 | 000 |
| 267 | SM26 | 056 | 000 | 000 | 000 |
| 270 | (reserved) | 137 | 000 | 000 | 000 |
| 271 | NS01 | 061 | 000 | 000 | 000 |
| 272 | SM20 | 137 | 010 | 157 | 000 |
| 273 | SP18 | 076 | 076 | 000 | 000 |
| 274 | NF04 | 061 | 057 | 064 | 000 |
| 275 | NF01 | 061 | 057 | 062 | 000 |

| ASCII Code | Name | Byte 4 | Byte 3 | Byte 2 | Byte 1 |
|---|---|---|---|---|---|
| 276 | (reserved) | 137 | 000 | 000 | 000 |
| 277 | SP16 | 077 | 000 | 000 | 000 |
| 300 | LA14 | 101 | 000 | 000 | 000 |
| 301 | LA12 | 101 | 000 | 000 | 000 |
| 302 | LA16 | 101 | 000 | 000 | 000 |
| 303 | LA20 | 101 | 000 | 000 | 000 |
| 304 | LA18 | 101 | 000 | 000 | 000 |
| 305 | LA28 | 101 | 000 | 000 | 000 |
| 306 | LA52 | 101 | 105 | 000 | 000 |
| 307 | LC42 | 054 | 010 | 103 | 000 |
| 310 | LE14 | 105 | 000 | 000 | 000 |
| 311 | LE12 | 105 | 000 | 000 | 000 |
| 312 | LE16 | 105 | 000 | 000 | 000 |
| 313 | LE18 | 105 | 000 | 000 | 000 |
| 314 | LI14 | 111 | 000 | 000 | 000 |
| 315 | LI12 | 111 | 000 | 000 | 000 |
| 316 | LI16 | 111 | 000 | 000 | 000 |
| 317 | LI18 | 111 | 000 | 000 | 000 |
| 320 | (reserved) | 137 | 000 | 000 | 000 |
| 321 | LN20 | 116 | 000 | 000 | 000 |
| 322 | LO14 | 117 | 000 | 000 | 000 |
| 323 | LO12 | 117 | 000 | 000 | 000 |
| 324 | LO16 | 117 | 000 | 000 | 000 |
| 325 | LO20 | 117 | 000 | 000 | 000 |
| 326 | LO18 | 117 | 000 | 000 | 000 |

| | | | | | |
|---|---|---|---|---|---|
| 327 | L052 | 117 | 105 | 000 | 000 |

| ASCII Code | Name | Byte 4 | Byte 3 | Byte 2 | Byte 1 |
|---|---|---|---|---|---|
| 330 | L062 | 057 | 010 | 117 | 000 |
| 331 | LU14 | 125 | 000 | 000 | 000 |
| 332 | LU12 | 125 | 000 | 000 | 000 |
| 333 | LU16 | 125 | 000 | 000 | 000 |
| 334 | LU18 | 125 | 000 | 000 | 000 |
| 335 | LY18 | 131 | 000 | 000 | 000 |
| 336 | (reserved) | 137 | 000 | 000 | 000 |
| 337 | LS61 | 163 | 163 | 000 | 000 |
| 340 | LA13 | 140 | 010 | 141 | 000 |
| 341 | LA11 | 047 | 010 | 141 | 000 |
| 342 | LA15 | 136 | 010 | 141 | 000 |
| 343 | LA19 | 176 | 010 | 141 | 000 |
| 344 | LA17 | 042 | 010 | 141 | 000 |
| 345 | LA27 | 141 | 000 | 000 | 000 |
| 346 | LA51 | 141 | 145 | 000 | 000 |
| 347 | LC41 | 054 | 010 | 143 | 000 |
| 350 | LE13 | 140 | 010 | 145 | 000 |
| 351 | LE11 | 047 | 010 | 145 | 000 |
| 352 | LE15 | 136 | 010 | 145 | 000 |
| 353 | LE17 | 042 | 010 | 145 | 000 |
| 354 | LI13 | 140 | 010 | 151 | 000 |
| 355 | LI11 | 047 | 010 | 151 | 000 |
| 356 | LI15 | 136 | 010 | 151 | 000 |
| 357 | LI17 | 042 | 010 | 151 | 000 |
| 360 | (reserved) | 137 | 000 | 000 | 000 |

| | | | | | |
|---|---|---|---|---|---|
| 361 | LN19 | 176 | 010 | 156 | 000 |

| ASCII Code | Name | Byte 4 | Byte 3 | Byte 2 | Byte 1 |
|---|---|---|---|---|---|
| 362 | L013 | 140 | 010 | 157 | 000 |
| 363 | L011 | 047 | 010 | 157 | 000 |
| 364 | L015 | 136 | 010 | 157 | 000 |
| 365 | L019 | 176 | 010 | 157 | 000 |
| 366 | L017 | 042 | 010 | 157 | 000 |
| 367 | L051 | 157 | 145 | 000 | 000 |
| 370 | L061 | 057 | 010 | 157 | 000 |
| 371 | LU13 | 140 | 010 | 165 | 000 |
| 372 | LU11 | 047 | 010 | 165 | 000 |
| 373 | LU15 | 136 | 010 | 165 | 000 |
| 374 | LU17 | 042 | 010 | 165 | 000 |
| 375 | LY17 | 042 | 010 | 171 | 000 |
| 376 | (reserved) | 137 | 000 | 000 | 000 |
| 377 | (reserved) | 137 | 000 | 000 | 000 |

## 14 CIPWT -- CLOCK REQUEST QUEUE

Description: This table allows a monitor routine to be run after a specific time interval. One entry for each job number, plus three more.

Position of entry in table is of no importance. Each entry occupies 2 words.

Each count is decremented by clock-level service in the monitor. When any ountdown reaches 0, (or goes negative), the contents of the second word of the entry are put into T1 and a PUSHJ is done to the address in the left half of the first word.

CLOCK is a 36 bit byte pointer to the highest in-use entry. Routines which make requests to be stored in this table will reference CLOCK. When an entry is eliminated from the table, the last entry is copied into its place, and CLOCK is decremented.

Defined in: COMMON

Used by: CLOCK1, DISSER, DLSINT, ERRCON, FILIO, PSISER, RPXKON, SCNSER, SYSINI, UUOCON, VBCSER

Format of each two-word entry:

```
|-------------------------------------------------------------|
|     Routine address          |          Count down timer    |
|-------------------------------------------------------------|
| 0|1-3| 4| 5-------------------------------------------35    |
|-------------------------------------------------------------|
```

The bits in the second word of each entry are defined as follows:

| Bit | Meaning |
|---|---|
| 0 | Request is CPU-specific |
| 1-3 | CPU number |
| 4 | Scanned by CLOCK1 |
| 5-35 | Data |

## 15 COMTAB -- COMMAND TABLE

Description: Specifies legal command names. There are corresponding entries in DISP that give routine address and legality conditions for each command.

There is one entry for each monitor command.

Position in the table is of no significance, but COMTAB and DISP entries must be in corresponding positions.

Defined in: COMMON

Used by: UUOCON, COMCON

GETTAB Table: .GTCOM (30)

Format:

```
+-------------------------------------------------------------+
|                   SIXBIT   /   CMD1     /                   |
|-------------------------------------------------------------|
|                   SIXBIT   /   CMD2     /                   |
|-------------------------------------------------------------|
|                   SIXBIT   /   CMD3     /                   |
|-------------------------------------------------------------|
                         .   .   .
```

## 16 CSRTAB -- CONTEXT SAVE/RESTORE TRANSLATION TABLE

Description: CSRTAB is used by the context service module (CTXSER) to save and restore certain job parameters.

Each word in the CSRTAB corresponds to information that must be saved and restored with each context switch for every job. When the information is contained in a single word (for example, the user PC for <CTRL/T>), the information is pointed to by the address in CSRTAB.

Other types of information (such as the TTY DDB) cannot be saved and restored in a single word. The CSRTAB entry for this kind of information contains a flag (Bit 0 is set) to indicate that the effective address of the entry is the location of a subroutine to save/restore that information.

Entries in CSRTAB are defined using the CTX macro. Using CSRTAB, data is copied to and from the job parameter portion of a context block, starting at offset .CTBPR (Beginning of the Parameter Block). Each offset in the parameter block is assigned a name by the CTX macro.

Defined in: CTXSER

Used by: CTXSER

Format of CSRTAB Entry:

```
+-------------------------------------------------------------+
| S | Z |                   | @ | (AC) |       Address         |
+-------------------------------------------------------------+
```

Every CSRTAB entry contains the following fields:

| Bit | Symbol | Meaning |
|---|---|---|
| 0 | S | A subroutine is used to save/restore the data. |
| 1 | Z | Bit 0 is off, and the data word is zeroed after the save/restore. |
| 13 | @ | Bit 0 is off, and hardware indirection is used to find the data word. |
| 14-17 | (AC) | Bit 0 is off, and hardware indexing is used to find the data word. |
| 18-35 | Address | If Bit 0 is off, this address is used to calculate the effective address of the data word. If Bit 0 is on, this is the location of the subroutine to save/restore data. |

Format of the CSRTAB table is illustrated below. Each word is described by:

- o Symbol is the symbol associated with the offset.
- o The setting of Bit 0, the S flag described above.
- o The setting of Bit 1, the Z flag described above.
- o Address is the symbol associated with the word where job parameters are saved.
- o Job information saved in the word at Address.

| Symbol | Bit 0 | Bit 1 | Address | Job Information |
|---|---|---|---|---|
| .CXSYS | 1 | 0 | CSRSYS | "From SYS" bit |
| .CXMON | 1 | 0 | CSRMON | Monitor mode bit, and others |
| .CXSCX | 0 | 1 | .PDSCX (W) | SAVCTX word |
| .CXBKM | 1 | 0 | CSRBKM | Terminal break mask (saves 40 words) |
| .CXPIA | 0 | 1 | JBTPIA (J) | PSI data base (PIT) |
| .CXIPC | 1 | 0 | CSRIPC | IPCF data base (saves 11 words) |
| .CXENQ | 1 | 0 | CSRENQ | ENQ/DEQ queue chain address |
| .CXTTY | 1 | 0 | CSRTTY | TTY DDB (save 13 words) |
| .CXSTS | 1 | 0 | CSRSTS | Job status |
| .CXST2 | 0 | 0 | JBTST2 (J) | Second job status word |
| .CXSWP | 0 | 0 | JBTSWP (J) | Swapped-out disk address |
| .CXIMI | 0 | 0 | JBTIMI (J) | Swapped-in image size |
| .CXIMO | 0 | 0 | JBTIMO (J) | Swapped-out image size |
| .CXSGN | 0 | 0 | JBTSGN (J) | High segment |

| Symbol | Bits 0 | Bits 1 | Address | Job Information |
|---|---|---|---|---|
| .CXAD2 | 1 | 0 | CSRAD2 | JBTAD2 |
| .CXPDB | 0 | 0 | JBTPDB (J) | Number of funny pages |
| .CXCHK | 0 | 0 | JBTCHK (J) | Swapped-out checksum |
| .CXPRG | 0 | 0 | JBTNAM (J) | Name of program to run |
| .CXPC | 0 | 0 | JBTPC (J) | User PC for <CTRL/T> |
| .CXDDB | 0 | 0 | JBTDDB (J) | I/O wait DDB |
| .CXNAM | 0 | 0 | .PDNAM (W) | Program file name |
| .CXSTR | 0 | 0 | .PDSTR (W) | Program's file structure |
| .CXDIR | 0 | 0 | .PDDIR (W) | Program's PPN |
| .CXSFD | 0 | 0 | .PDSFD (W) | Program's SFD(s) |
| .CXSTM | 0 | 0 | .PDSTM (W) | Time of last RESET |
| .CXCMN | 0 | 0 | .PDCMN (W) | Ptr to user-defined commands |
| .CXUNQ | 0 | 0 | .PDSJB (W) | Ptr to UNQTAB for user commands |
| .CXSJB | 0 | 1 | .PDSJB (W) | DECnet data base |
| .CXABS | 0 | 0 | .PDABS (W) | Address break settings |
| .CXTMI | 0 | 0 | .PDTMI (W) | Virtual timer trap interval |
| .CXTMC | 0 | 0 | .PDTMC (W) | Virtual timer counter |
| .CXSPS | 1 | 0 | CSRSPS | SET CPU command bits |
| .CXVRT | 0 | 0 | JBTVRT (J) | Program size for <CTRL/T> |
| .CXSG2 | 0 | 0 | JBTSG2 (J) | Section no. for high segments |
| .CXCVL | 0 | 0 | .PDCVL (W) | Current phys. and virt. limits |
| .CXLBS | 1 | 0 | CSRLBS | UUO setting for BIGBUF |
| .CXRTD | 1 | 0 | CSRRTD | HPQ and HIBER settings |
| .CXPAT | 1 | 0 | CSRPAT | Patch space |

In the table illustrated above, the fields are:

- Symbol is the symbol name associated with the entry in the table. They are formatted as .CXxxx, where xxx is the unique, three-character name, and is added to .CX by the CTX macro.

- Bit 0 is the flag, indicating whether a data word is saved/restored, or a subroutine is called to perform the save/restore operation. If Bit 0 is on, the subroutine is pointed to by the symbolic address.

- Bit 1 indicates whether a data word is to be preserved or zeroed after the save/restore operation. If Bit 0 is on, Bit 1 must be off. If Bit 1 is on, the data word referred to by the symbolic address will be zeroed after the save/restore operation.

- The address in the table is the symbolic location of the data word or subroutine (depending on Bit 0). Data words are subject to indirection and indexing, as indicated by the presence of (J) or (W) after the symbol above. A (J) indicates the address is the location of a job table, and the exact data word to be saved/restored found by indexing into the table using the job number ("J" = Job). A (W) indicates the data word is in the job's Process Data Block (PDB); it is found using the location of the data word in the PDB.

- The job information column in the above table contains a description of the type of job information to be saved or restored.

  Most of the entries in CSRTAB save one data word. However, certain subroutines save more that one word, as indicated in the description of the job information in the table shown above.

## 17 CSTTAB -- CUSTOMER-DEFINED COMMANDS TABLE

Description: Specifies legal command names that have been defined in the monitor by customer. There are corresponding entries in DISPC that give routine address and legality conditions for each command.

There is one entry for each monitor command.

Position in the table is of no significance, but CSTTAB and DISPC entries must be in corresponding positions.

Defined in: COMMON

Used by: UUOCON, COMCON

Format:

```
+-------------------------------------------------------------+
|                    SIXBIT   /   CMD1     /                  |
|-------------------------------------------------------------|
|                    SIXBIT   /   CMD2     /                  |
|-------------------------------------------------------------|
|                    SIXBIT   /   CMD3     /                  |
|-------------------------------------------------------------|
                           .   .   .
```

## 18 COMTB2 -- SET COMMAND TABLE

Description: When the command interpreter has determined that a SET command is to be executed, it does a table lookup on the SET command argument to determine the dispatch address. The SET command argument table is named COMTB2 and is in the same format as COMTAB. The dispatch table for the SET commands is named DISP2. COMTB2 and DISP2 entries must be in corresponding positions.

Defined in: COMMON

Used by: UUOCON,COMCON

GETTAB Table: .GTCM2 (43)

Format:

```
+-------------------------------------------------------------+
|                    SIXBIT   /   CMD1     /                  |
|-------------------------------------------------------------|
|                    SIXBIT   /   CMD2     /                  |
|-------------------------------------------------------------|
|                    SIXBIT   /   CMD3     /                  |
|-------------------------------------------------------------|
                         .     .     .
```

## 19 DDB -- DEVICE DATA BLOCK

Description: Contains information needed to perform I/O operations. One such block exists for each device or, in the case of disk, one for each INIT or ASSIGN.

Number of entries in the DDB varies with the device.

Labels for DDBs are defined in S.MAC. Each device service routine contains a DDB for that device. For devices with more than one DDB, the DDBs are set dynamically as needed or at system initialization time. The new DDBs are modeled after the one contained in the device service routine.

The label of a DDB entry is defined as the location of that entry relative to the beginning of the block. Such labels must be indexed by an AC containing the address of the beginning of the DDB. AC F is normally used for this purpose in the monitor.

The start of the DDB chain is accessible as GETTAB item %CNDEV from a user program, or in location DEVLST in the monitor. DDBs for some peripherals have their own tags in the monitor and some sub-chains are also tagged:

TTYLST - Start of TTY sub-chain
PTYLST - Start of PTY sub-chain
LPxDDB - Tag for LPTx on I/O bus
CRxDDB - Tag for CDRx on I/O bus
FLxDDB - Tag for LPTx off console front end
FCxDDB - Tag for CDRx off console front end

Defined in: Device service routines and in S.MAC

Used by: Device service routines

```
Symbol      Map
           +------------------------------------------------------------+
DEVNAM     |                   SIXBIT device name                       |
           |------------------------------------------------------------|
DEVCHR*    |                   See bit definitions                      |
           |------------------------------------------------------------|
DEVIOS*    |                   See bit definitions                      |
           |------------------------------------------------------------|
DEVSER     |  Adr of next DDB           |Dispatch table address         |
           |------------------------------------------------------------|
DEVMOD*    |                   See bit definitions                      |
           |------------------------------------------------------------|
DEVLOG     |                  SIXBIT logical device name                |
           |------------------------------------------------------------|
DEVBUF     |   Address of user's 3-word | Address of user's 3-word      |
           |   output buffer header     |   input buffer header         |
           |------------------------------------------------------------|
DEVIAD*    |                   See bit definitions                      |
           |------------------------------------------------------------|
DEVOAD*    |                   See bit definitions                      |
           |------------------------------------------------------------|
DEVSTS     |             Word for device CONI                           |
           |------------------------------------------------------------|
DEVSTA*    |              See bit definitions                           |
           |------------------------------------------------------------|
DEVXTR*    |                 See bit definitions                        |
           |------------------------------------------------------------|
DEVEVM*    |                 See bit definitions                        |
           |------------------------------------------------------------|
DEVPSI     |   Enabled PSI conditions  |   Pending PSI interrupts       |
           |------------------------------------------------------------|
DEVESE*    |                 See bit definitions                        |
           |------------------------------------------------------------|
DEVHCW*    |                 See bit definitions                        |
           |------------------------------------------------------------|
DEVCPU*    |                 See bit definitions                        |
           |------------------------------------------------------------|
DEVJOB*    |                 See bit definitions                        |
           |------------------------------------------------------------|
```

The remainder of the DDB is different for different types of devices. The TTY DDB is continued after the following description of the magtape and disk DDBs.

The Disk and Magtape DDB contain the following words, following DEVJOB:

| | | |
|---|---|---|
| DEVFIL | File name on last LOOKUP/ENTER (arg to FNDFIL) | |
| DEVEXT,, DEVLFT | File extension on last LOOKUP/ENTER | Number blocks left in current group |
| DEVPPN | PPN on last LOOKUP/ENTER | |
| DEVNBF | Number buffers swept for | Buffers not swept for |
| DEVSBF | Saved value of DEVNBF | |
| DEVCSN | Cache sweep serial number | |
| DEVISN | | Section no. for I/O |

The disk DDB follows. The magtape DDB is continued after the remainder of the disk DDB.

The disk DDB continues from offset DEVISN, as follows:

| | | |
|---|---|---|
| DEVDMP | Current IOWD for dump mode | |
| DEVRET | 0 | Addr of current retrieval pointer in DDB |
| DEVREL DEVDIA | Relative block number in file to read or write next | |
| DEVUNI | Addr of original UDB | Addr of current UDB |
| DEVUFB | | Addr of UFD data block |
| DEVSFD | | Addr of NMB for father SFD |
| DEVBLK | Logical block number in unit to read or write next | |
| DEVRSU DEVACC | -Num of unused pointer positions in RIB | Addr of access table entry for user channel |
| DEVFLR | Block number of file which first in-core ptr points to | |
| DEVFUN* | See bit definitions | |
| DEVQUE | Addr of next DDB in queue | Job number of DDB in LH |

| | |
|---|---|
| DEVELB* | See bit definitions |
| DEVLRL<br>DEVPRI | Last DEVREL \| Disk priority |
| DEVSPN | SIXBIT spooled file name |
| DEVSPM | Pointer to spooling parameter block |
| DEVRIB* | See bit definitions |
| DEVUPP | "In-your-behalf" PPN |
| DEVCUR* | See bit definitions |
| DEVGEN | Generation number of UDB \| Addr of core copy of RIBs |
| DEVLNM | Current SFD looking for \| Logical name spec being used |
| DEVCFS* | See bit definitions |
| DEVRB1* | First retrieval pointer (Same format as RIB) |
| DEVRB2* | Second retrieval pointer |
| | . . . |
| DEVRBn* | Last retrieval pointer in core |
| DEVDRB | Addr. of I/O Request Block for CI disks |

The Magtape DDB has different words following DEVISN, as follows:

| | |
|---|---|
| TDVUDB<br>TDVKDB | UDB pointer \| KDB pointer (prime) |
| TDVSTS* | See bit definitions |
| TDVIOR | IORB to wait for |
| TDVSUL | Saved user upper limit |
| TDVSLL | Saved user lower limit |
| TDVSVM | Saved M for dump mode |
| TDVREM | Remainder for mode 16 |

The Terminal DDB (TTY DDB) differs from the disk and magtape DDBs. It contains the following information after the DEVJOB word:

```
          |----------------------------------------------------------|
DDBLDB    |        Unused           |        Address of attached LDB |
          +----------------------------------------------------------+
```

* These words are described in more detail on the following pages.

## 19.1 DEVCHR -- Device Characteristics Word

```
|-----------------------------------------------------------------|
|LPTL|CMDA|DATJ|DIBP|CNET|OFLN|C2IO|PDVTIM| PDVCNT |      | PBUFSZ  |
|-----------------------------------------------------------------|
0      1    2    3    4    5    6   7---9 10----16 17-18 19-------35
```

| Bits | Value | Byte Pointer | Meaning |
|---|---|---|---|
| 0 | 400000 | DVLPTL | Lowercase line printer |
| 1 | 200000 | DVCMDA | Device controlled by mountable device allocator |
| 2 | 100000 | DVDATJ | Device allocated to job in DEVJOB |
| 3 | 40000 | DVDIBP | Device is a batch PTY |
| 4 | 20000 | DVCNET | Device controlled by NETSER |
| 5 | 10000 | DVOFLN | Device off-line last time service routine polled |
| 6 | 4000 | PDV2IO | Device can do simultaneous input and output |
| 7-9 | 3400 | PDVTIM | Code for hung device timeout<br>0 means device cannot be hung,<br>n means hung time is 2**n-1 seconds |
| 10-16 | 376 | PDVCNT | Countdown timer for the hung device. PDVCNT contains the number of seconds to go before considering device hung. This value is initialized every time the device is serviced, using the hung constant to determine the value. |
| 17-18 | | | Unused |
| 19-35 | 377777 | PBUFSZ | Buffer size |

19.2 DEVIOS -- Device Input/Output Status Word

```
|-----------------------------------------------------------------------|
| DEV1 | E| I| S| F| B| W| I| E|DT|BK|DE| A| DEV2 | C|WC| PIOMOD       |
|-----------------------------------------------------------------------|
 0--11  12 13 14 15 16 17 18 19 20 21 22 23  24-29 30 31 32--------35
```

| Bits | Value | Byte Pointer | Meaning |
|---|---|---|---|
| 0-11 | | | Device-dependent bits (see Notes, below) |
| 12 | 40 | IOEND | Service routine has transmitted last data |
| 13 | 20 | I/O | 1 for output; 0 for input |
| 14 | 10 | IOSTBL | Device error flag |
| 15 | 4 | IOFST | Next item will be the first item of a buffer |
| 16 | 2 | IOBEG | Virgin device |
| 17 | 1 | IOW | Input/output wait |
| 18 | 400000 | IOIMPM | Improper mode |
| 19 | 200000 | IODERR | Device error |
| 20 | 100000 | IODTER | Data error |
| 21 | 40000 | IOBKTL | Block too large |
| 22 | 20000 | IODEND | Data end encountered |
| 23 | 10000 | IOACT | Device active |
| 24-29 | | | Device-dependent bits (listed below) |
| 30 | | IOCON | Continuous |
| 31 | | IOWC | Don't compute word count |
| 32-35 | | PIOMOD | Data mode codes: |

| Code | Symbol | Mode |
|---|---|---|
| 0 | A | ASCII |
| 1 | AL | ASCII line |
| 2 | PIMMOD | Packed image |
| 3 | BYTMOD | Byte |
| 10 | I | Image |
| 13 | IB | Image binary |
| 14 | B | Binary |
| 15 | SD | Scope dump |
| 16 | DR | Dump by records |
| 17 | D | Dump across records |

Notes:

Bits 0-11 and 24-29 of the DEVIOS word are defined differently for the type of device DDB. Those bits are defined in the following lists:

Disk DDB:

| Bits | Value | Symbol | Meaning |
|---|---|---|---|
| 0 | 400000 | IOSMON | Monitor I/O request (such as reading a RIB) |
| 1 | 200000 | IOSAU | File has AU resource |
| 2 | 100000 | IOSUPR | Super-USETI/USETO being used |
| 3 | 40000 | IOSDA | File has DA resource |
| 4 | 20000 | IOSRIB | RIB is in monitor buffer |
| 5 | 10000 | IOSRDC | File has read count up |
| 6 | 4000 | IOSWLK | File structure is software write-locked |
| 7 | 2000 | IOSPBF | Partial buffer done |
| 8 | 1000 | IOSFIR | First block of group being accessed (Compute or check retrieval pointer checksum) |
| 9 | 400 | IOSSCE | Software checksum error encountered |
| 10 | 200 | IOSHWE | Hardware write error encountered |
| 11 | 100 | IOSHRE | Hardware read error encountered |
| 16 | 2 | IOSHMS | Hung message already typed |
| 17 | 1 | IOSRST | RESET or RELEAS done on spooled device |
| 28 | 200 | UDSX | Super-USETO is formatting disk |

Magtape DDB:

| Bits | Value | Symbol | Meaning |
|---|---|---|---|
| 1 | 200000 | OFFLIN | Unit is off-line |
| 2 | 100000 | OFLUNH | Off-line unit is not ready |
| 3 | 40000 | FINP | First input operation |
| 4 | 20000 | LBLNED | Labelling action needed |
| 5 | 10000 | LBLWAT | Waiting for labelling process |
| 6 | 4000 | LBLSTP | Stop I/O because of error |
| 7 | 2000 | FOUT | First output operation |
| 8 | 1000 | LBLEOF | EOF encountered |
| 24 | 4000 | IOBOT | Beginning of tape |
| 25 | 2000 | IOTEND | End of tape |
| 26 | 1000 | IOPAR | Write even parity if 1 on magtape |
| 27-28 | 600 | PDENS | Density of magtape: |
| | 00 | | Installation default |
| | 01 | | 200 BPI |
| | 10 | | 556 BPI |
| | 11 | | 800 BPI |
| 29 | 100 | IONRCK | Read with no reread check |

TTY DDB:

| Bits | Value | Symbol | Meaning |
|---|---|---|---|
| 0 | 400000 | TTYOUW | I/O wait is for output |
| 1 | 200000 | FRCEND | Force EOF due to image mode timeout |
| 2 | 100000 | IOLBKA | Saved value of IOSBKA when HALTed |
| 25 | 2000 | IOSABS | Break on characters in break mask table |
| 26 | 1000 | IOSBKA | Break on all characters |
| 27 | 400 | IOSTEC | "Truth in echoing" mode |
| 28 | 200 | IOSNEC | "No echo" mode |
| 29 | 100 | IOSFCS | User wants all characters |

TSK DDB:

| Bits | Value | Symbol | Meaning |
|---|---|---|---|
| 4 | 20000 | IOSUSO | UUOCON stopped output (no output buffers available) |
| 5 | 10000 | IOSUSI | UUOCON stopped input (no input buffers available) |
| 6 | 4000 | IOSERR | I/O can't continue due to an error |
| 7 | 2000 | IOSZAP | Device no longer owned by job |
| 8 | 1000 | IOSCON | Device is connected |
| 9 | 400 | IOSREL | Device has been released |
| 11 | 100 | IOSDMR | Disable message reassembly |

## 19.3 DEVMOD -- Device Modes Word

```
|-----------------------------------------------------------------|
|DC|DS|CD|LP|AT|TF|TU|DI|LN|PT|PR|DT|AV|MT|TT|DR|IN|OU|AC|AP|I/O mode|
|-----------------------------------------------------------------|
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20----35
```

| Bits | Value | Pointer | Meaning |
|---|---|---|---|
| 0 | 400000 | DVDIRI | DECtape directory is in core |
| 1 | 200000 | DVDSK | Disk |
| 2 | 100000 | DVCDR | Card reader |
| 3 | 40000 | DVLPT | Line printer |
| 4 | 20000 | TTYATC | TTY attached to job if 1 |
| 5 | 10000 | TTYUSE | TTYDDB in use flag |
| 6 | 4000 | TTYBIU | TTYDDB in use |
| 7 | 2000 | DVDIS | Display (DIS) device |
| 8 | 1000 | DVLNG | Device has long dispatch table |
| 9 | 400 | DVPTP | Paper tape punch |
| 10 | 200 | DVPTR | Paper tape reader |
| 11 | 100 | DVDTA | DECtape |
| 12 | 40 | DVAVAL | Device is available to this job |
| 13 | 20 | DVMTA | Magnetic tape (rewind) |
| 14 | 10 | DVTTY | Terminal |
| 15 | 4 | DVDIR | Directory-oriented device |
| 16 | 2 | DVIN | Device can do input |
| 17 | 1 | DVOUT | Device can do output |
| 18 | 400000 | ASSCON | Device assigned by ASSIGN command |
| 19 | 200000 | ASSPRG | Device assigned by program, using a monitor call |
| 20-35 | 177777 | | If data mode n is legal, bit 35-n is set. |

## 19.4 DEVIAD -- Input Buffer Word

```
|-------------------------------------------------------------------|
|  Device-dependent bits       |    Current input buffer address    |
|-------------------------------------------------------------------|
 0------------------------13 14----------------------------------35
```

| Bits | Value | Symbol | Meaning |
|---|---|---|---|
| 0-13 | 777760 | | Device-dependent bits (listed below) |
| 14-35 | | PDVIAD | Address of current user's input buffer (exec virtual address if EVM, user virtual address if not in EVM). |

Disk DDB:

| Bits | Value | Symbol | Meaning |
|---|---|---|---|
| 0-2 | 700000 | DEYCOD | File status code (from UNISTS): |

| Code | Symbol | Meaning |
|---|---|---|
| 0 | ICOD | Idle |
| 3 | PWCOD | Position wait |
| 4 | PCOD | Positioning |
| 5 | TWCOD | Transfer wait |
| 6 | TCOD | Transferring |

| Bits | Value | Symbol | Meaning |
|---|---|---|---|
| 3 | 400000 | DEYSCN | SFD scanning (/SCAN) in effect |
| 4 | 200000 | DEPLPC | Last RIB pointer is in core |
| 5-8 | 17000 | DEYFNC | Highest allowed function with file: |

| Code | Symbol | Meaning |
|---|---|---|
| 1 | FNCEXC | Execute only |
| 2 | FNCRED | Read |
| 3 | FNCALL | Allocate |
| 4 | FNCDLL | Deallocate |
| 5 | FNCAPP | Append |
| 6 | FNCUDP | Update |
| 7 | FNCCRE | Create |
| 10 | FNCSUP | Supersede |
| 11 | FNCTRN | Truncate |
| 12 | FNCCAT | Change attributes except name, directory, and privileges. |
| 13 | FNCDEL | Delete |
| 14 | FNCCNM | Change name |
| 15 | FNCCPR | Change privileges |

| | | | |
|---|---|---|---|
| 9-12 | 740 | DEYEUN | Logical unit, within structure, of error |

DECtape DDB:

| Bits | Value | Symbol | Meaning |
|---|---|---|---|
| 1-2 | 300000 | IADPTR | Number of channels on which the device is initiated. |

Magtape DDB:

| Bits | Value | Symbol | Meaning |
|---|---|---|---|
| 0 | 400000 | OFLHNG | Hung device |
| 9 | 400 | MTSNAR | Set if user disabled RETRY |
| 10 | 200 | IOSRTY | No retry on error |
| 11 | 100 | IOSCPZ | I/O being started on a queued I/O request |

### 19.5 DEVOAD -- Output Buffer Word

```
|-----------------------------------------------------------------|
| Device-dependant bits   |        Current output buffer address   |
|-----------------------------------------------------------------|
 0--------------------13  14-------------------------------------35
```

| Bits | Value | Symbol | Meaning |
|---|---|---|---|
| 0-13 | 777760 | | Device-dependent bits (listed below) |
| 14-35 | | PDVOAD | Address of current user's output buffer (exec virtual address if in EVM, user virtual address if not in EVM). |

Disk DDB:

| Bits | Value | Symbol | Meaning |
|---|---|---|---|
| 0 | 400000 | DEPSWP | SWPSER DDB |
| 1 | 200000 | DEPLIB | LOOKUP from LIB/SYS |
| 2-8 | 177000 | DEYRLC | Offset into RIB of first retrieval pointer stored in DEVRB1 |
| 9 | 400 | DEPUWZ | USETO writing zeros to extend file |
| 10 | 200 | DEPPPO | Zero PPN was specified in UUO |
| 11 | 100 | DEPFDA | FILDAE should be called on CLOSE |

Magtape DDB:

| Bits | Value | Symbol | Contents |
|---|---|---|---|
| 0-13 | 777760 | PBUFRM | Maximum frame count from TAPOP. UUO function |

TTY DDB:

| Bits | Value | Symbol | Contents |
|---|---|---|---|
| 1-12 | 377740 | BYTCNT | Remaining byte count for asynchronous (non-blocking) output. |

DECtape DDB:

| Bits | Value | Symbol | Contents |
|---|---|---|---|
| 0-6 | 774000 | SLPNTR | Dead-reckoning sleep time, in seconds. |

19.6 DEVSTA -- Device Station Word

```
|-------------------------------------------------------------------------|
|SP|AD|LE|RA|TYP|IB|  SPL  |AI||IN|ON|EC|MS|EV|DR|DL|       |  PDVSTA    |
|-------------------------------------------------------------------------|
 0  1  2  3 4-9 10 11---17 18 19 20 21 22 23 24 25 26-29 30-------35
```

| Bits | Value | Symbol | Meaning |
|---|---|---|---|
| 0 | 400000 | DEPSPL | DDB is for a spooled device |
| 1 | 200000 | DEPADY | Disk in 10/11 compatibility mode |
| 2 | 100000 | DEPLEN | Variable length buffers |
| 3 | 40000 | DEPRAS | Device has restricted assignment |
| 4-9 | 37400 | PDVTYP | Device type |
| 10 | 200 | DEPIBC | Inhibit clearing output buffers |
| 11-17 | 177 | | Spool bits: |

| Mask | Symbol | Device |
|---|---|---|
| 37 | .SPALL | Bit mask for defined spool bits |
| 20 | .SPCDR | Card reader spool bit |
| 10 | .SPCDP | Card punch spool bit |
| 4 | .SPPTP | Paper tape punch spool bit |
| 2 | .SPPLT | Plotter spool bit |
| 1 | .SPLPT | Line printer spool bit |

| Bits | Value | Symbol | Meaning |
|---|---|---|---|
| 18 | 400000 | DEPAIO | Doing asynchronous input/output |
| 19 | 200000 | DEPIND | Input not yet done (asynch I/O) |
| 20 | 100000 | DEPOND | Output not yet done (asynch I/O) |
| 21 | 40000 | DEPECS | On if a non-superceding enter |
| 22 | 20000 | DEPMSG | This device controlled by MSGSER |
| 23 | 10000 | DEPEVM | Device doesn't need EVM |
| 24 | 4000 | DEPDER | Disable error recovery on this device |
| 25 | 2000 | DEPDEL | Disable error logging on this device |
| 26-29 | 1700 | | Unused |
| 30-35 | 77 | PDVSTA | Station number (node) of device |

## 19.7 DEVXTR -- Extra Word

The DEVXTR contains different information for different types of DDBs. The format for each type of DDB is listed below:

MPX DDB:

| Bits | Value | Contents |
|---|---|---|
| 0-17 | 777777 | Address of connected device table |
| 18-35 | 777777 | Number of connected devices |

MPX-Controlled Device DDB:

| Bits | Value | Contents |
|---|---|---|
| 0-17 | 777777 | I/O flags (same as left half of USRJDA) |
| 18-35 | 777777 | Address of MPX: DDB to which device is connected |

Magtape DDB:

| Bits | Value | Symbol | Contents |
|---|---|---|---|
| 0-2 | 700000 | TDYHNI | Initial value of queued/asynch I/O hung timer. |
| 3-5 | 70000 | TDYHNG | Current value of queued/asynch I/O hung timer. The value of TDYHNG is the number of times a queued I/O or asynchronous I/O request can get hung device errors if it has not been actually started, before a hung device condition actually occurs. |
| 6-35 | | | Unused |

19.8 DEVEVM -- Exec Virtual Memory Word

```
|----------------------------------------------------------------------|
|  PIEVM         |    PIEVMS      |       POEVM       |    POEVMS      |
|----------------------------------------------------------------------|
0-----------8  9-----------17  18------------26  27---------------35
```

| Bits | Value | Symbol | Contents |
|---|---|---|---|
| 0-8 | 777000 | PIEVM | Page number of this device's EVM for buffered input |
| 9-17 | 777 | PIEVMS | Number of pages of EVM allocated for buffered input |
| 18-26 | 777000 | POEVM | Page number of this device's EVM for buffered output |
| 27-35 | 777 | POEVMS | Number of pages of EVM allocated for buffered output |

19.9 DEVESE -- Extended Software Error Word

```
|----------------------------------------------------------------------|
|      Next DDB        | PI Level |  CNDSIZ         |  PDVESE          |
|----------------------------------------------------------------------|
0-----------------17 18-------19  20 ---------26 27---------------35
```

| Bit | Value | Symbol | Meaning |
|---|---|---|---|
| 0-17 | 777777 | | Address of next DDB in pending PSI chain |
| 18-19 | 600000 | | PSI interrupt level |
| 20-26 | 177000 | | PSI vector offset |
| 27-35 | 777 | PDVESE | Extended software error status |

## 19.10 DEVHCW -- Device Hardware Characteristics Word

```
|-------------------------------------------------------------------|
| L | P | VFT | CST | TYP |  TYU   |       |     Character set name  |
|-------------------------------------------------------------------|
 0    1   2--5  6--8  9--11 12--14 15-17 18-----------------------35
```

| Bits | Symbol | Contents |
|---|---|---|
| 0 | HC.LCP | Lowercase printer |
| 1 | HC.PGC | Printer has page counter |
| 2-5 | HC.VFT | VFU type, one of the following codes: |

| Code | Symbol | Meaning |
|---|---|---|
| 0 | .HCVTO | Optical (paper tape) VFU |
| 1 | .HCVTD | Direct access VFU (DAVFU) |
| 2 | .HCVTN | No VFU (handled by hardware) |

| | | |
|---|---|---|
| 6-8 | HC.CST | Character set type, one of the following codes: |

| Code | Symbol | Meaning |
|---|---|---|
| 0 | .HCC64 | 64-character set |
| 1 | .HCC95 | 95-character set |
| 2 | .HCC28 | 128-character set |
| 3 | .HCCVR | Variable character set |

| | | |
|---|---|---|
| 9-11 | HC.TYP | Line printer type, one of the following codes: |

| Code | Symbol | Meaning |
|---|---|---|
| 0 | .HCTUK | Unknown |
| 1 | .HCTBX | BA10 |
| 2 | .HCTLC | LP100 |
| 3 | .HCT20 | LP20 |

| | | |
|---|---|---|
| 12-14 | .HCTYU | Line printer class, one of the following codes: |

| Code | Symbol | Meaning |
|---|---|---|
| 0 | .HCUUK | Unknown or unspecified |
| 1 | .HCULP | LP05-type |
| 2 | .HCULN | LN01-type |

| | | |
|---|---|---|
| 18-35 | HC.CSN | Character set name (SIXBIT/nnn/) |

19.11 DEVCPU -- CPU Word

```
|-------------------------------------------------------------------|
|CPF|CPS|PCL| CPU|                | Addr of interrupt interlock     |
|-------------------------------------------------------------------|
 0-2 3-5 6-8 9-11  12---------17  18-------------------------------35
```

| Bits | Value | Symbol | Contents |
|---|---|---|---|
| 0-2 | 700000 | DEYCPF | CPU number of primary CPU owning device. DEYCPF=7 when any CPU can do I/O to the device. In this case, DEYPCL will contain 0. |
| 3-5 | 70000 | DEYCPS | CPU number of secondary CPU owning device |
| 6-8 | 7000 | DEYPCL | CPU number of CPU doing I/O to device |
| 9-11 | 700 | DEYCPU | CPU number of CPU that did last IN or OUT UUO |
| 18-35 | | | Address of UUO/interrupt level interlock word |

19.12 DEVJOB -- Job Word

```
|-------------------------------------------------------------------|
|               |      PUNIT       |      PCTXN      |      PJOBN    |
|-------------------------------------------------------------------|
 0-------------8 9---------------17 18------------26 27------------35
```

| Bits | Symbol | Contents |
|---|---|---|
| 0-8 | | Reserved |
| 9-17 | PUNIT | Unit number. |
| 18-35 | PJCHN | Byte pointer to job context handle for INITed devices (includes both context number and job number). |
| 18-26 | PCTXN | Byte pointer to context number for INITed devices. |
| 27-35 | PJOBN | Job number of device owner |

19.13 DEVFUN -- UDB Pointer (Disk DDB Only)

```
|-------------------------------------------------------------------|
| S | L |                       | UDB address to which first         |
|   |   |                       | retrieval pointer points           |
|-------------------------------------------------------------------|
 0   1   2-----------------17    18-----------------------------35
```

| Bits | Value | Symbol | Contents |
|---|---|---|---|
| 0 | 400000 | DEPFFS | File found by scanning (/SCAN) |
| 1 | 200000 | DEPFFL | File found in LIB or SYS |
| 2-17 | 177777 | | Unused |
| 18-35 | 777777 | | Address of UDB for unit, to which first retrieval pointer in DDB points (DEVRB1) |

19.14 DEVELB -- Error Information (Disk DDBs Only)

```
|-------------------------------------------------------------------|
| Error bits |        First logical block number of bad region      |
|-------------------------------------------------------------------|
0---------8 9---------------------------------------------------35
```

| Bits | Value | Symbol | Contents |
|---|---|---|---|
| 3 | 400000 | BAPOTR | Other error (neither BAPDTR nor BAPHDR) |
| 4 | 200000 | BAPDTR | Data error |
| 5 | 100000 | BAPHDR | Search or header compare error |
| 9-35 | | | Logical block number, within structure, of start of bad region |

## 19.15 DEVRIB -- Current RIB Information (Disk DDBs Only)

```
|-----------------------------------------------------------------|
| E |    Count    |    Unit |           Cluster address           |
|-----------------------------------------------------------------|
 0   1--------8 9-----12 13--------------------------------------35
```

| Bits | Value | Symbol | Contents |
|---|---|---|---|
| 0 | 400000 | | Extended RIB |
| 1-8 | 377000 | DEYRBC | Count of RIBs |
| 9-12 | 740 | DEYRBU | Unit within structure |
| 13-35 | | DEYRBA | Cluster address within unit |

## 19.16 DEVCUR -- Current Unit Information (Disk DDBs Only)

```
|-----------------------------------------------------------------|
|RAD|PRV|RRC|RHC|PHO|LBF| DEYNB1  | DEYNBB   | Address of current UDB |
|-----------------------------------------------------------------|
 0   1   2   3   4   5   6----11 12-----17 18--------------------35
```

| Bits | Value | Symbol | Contents |
|---|---|---|---|
| 0 | 400000 | DEPRAD | Rename in progress |
| 1 | 200000 | DEPPRV | Don't check privileges on LOOKUP |
| 2 | 100000 | DEPRRC | Auto-rewrite of RIB on change |
| 3 | 40000 | DEPRHC | RIB had changed |
| 4 | 20000 | DEPPHO | Physical-only set in INIT |
| 5 | 10000 | DEPLBF | Use large buffers |
| 6-11 | 7700 | DEYNB1 | Number of blocks in first buffer |
| 12-17 | 77 | DEYNBB | Number of blocks per buffer |
| 18-35 | 777777 | | Address of current UDB doing I/O |

## 19.17 DEVFCS -- CFP-Supplied Word (Disk DDBs Only)

```
|-------------------------------------------------------------------|
|                               | P |  CFP-supplied on LOOKUP       |
|-------------------------------------------------------------------|
 0---------------------------16   17 18---------------------------35
```

| Bits | Value | Symbol | Contents |
|---|---|---|---|
| 17 | 1 | DEPCFS | CFP can be supplied by program |
| 18-35 | 777777 | | CFP from LOOKUP UUO |

## 19.18 DEVRBn -- Retrieval Pointer (Disk DDBs Only)

```
|-------------------------------------------------------------------|
|     Cluster count    |        Checksum       |     Cluster address |
|-------------------------------------------------------------------|
```

Widths of these fields are defined symbolically, and may be different for each file structure. Byte pointer is in structure data block.

| Field | Byte Pointer |
|---|---|
| Cluster count | STYCNP |
| Checksum | STYCKP |
| Cluster address | STYCLP (23 bits maximum) |

If cluster count = 0, the word actually is one of the following:

- o Pointer to new unit, if bit 18 = 1. Bits 19-35 specify logical unit number within file structure.
- o EOF flag, if whole word is zero.

19.19 TDVSTS -- Status Information Word (Magtape DDBs Only)

```
|-------------------------------------------------------------------|
|0|1| |3|TDYDEN | TDYMOD |                          | TDYMD1 | TDYDN1 |
|-------------------------------------------------------------------|
 0 1 2 3 4-----7  8----10 11--------------------28  29---31  32---35
```

| Bits | Value | Symbol | Contents |
|---|---|---|---|
| 0 | 400000 | D.RDBK | Read backwards |
| 1 | 200000 | D.NRLT | Read next record at low threshold |
| 2 | | | Unused |
| 3 | 40000 | D.EPAR | Even parity |
| 4-7 | 36000 | TDYDEN | Tape density used |
| 8-10 | 1600 | TDYMOD | Tape data format from TAPOP. UUO function .TFMOD |
| 11-28 | | | Unused |
| 29-31 | 160 | TDYMD1 | Tape data format from SET FORMAT command |
| 32-35 | 17 | TDYDN1 | Tape density from TAPOP. UUO function .TFDEN |

## 20 DEVDSP -- DEVICE DISPATCH TABLE

Description: Table of JRSTs to routines that perform various device-dependent functions. For each device, corresponding table entries go to routines to perform a specific function.

The first nine entries are present in each device service routine. Tables that contain the additional entries are referred to as long dispatch tables.

Defined in: Device Service Routines

Used by: COMMON, COMNET, MSGSER, SYSINI, UUOCON

The format of each word in this table is a JRST instruction, where the subroutine will perform one of the following functions:

| Label of entry (Relative to DEVDSP) | Relative Address | Function |
|---|---|---|
| DOFL | -5 | Is device offline |
| DDVO | -4 | DEVOP. UUO |
| DSZ | -3 | Return buffer size |
| DINI | -2 | Device and service routine initialization |
| DHNG | -1 | "Hung device" action |
| DRL | 0 | Release (table base adr - DEVDSP) |
| DCL | 1 | CLOSE, CLOSE output |
| DOU | 2 | OUTPUT operation |
| DIN | 3 | INPUT operation |
| DEN (1) | 4 | ENTER operation |

| | | |
|---|---|---|
| DLK (1) | 5 | LOOKUP operation |
| DDO (1) | 6 | DUMP mode output |
| DDI (1) | 7 | DUMP mode input |
| DSO (1) | 10 | USETO operation |
| DSI (1) | 11 | USETI operation |
| DGF (1) | 12 | UGETF operation |
| DRN (1) | 13 | RENAME operation |
| DCLI (1) | 14 | CLOSE input - dump mode |
| DCLR (1) | 15 | UTPCLR UUO |
| DMT (1) | 16 | MTAPE operation |

Labels for table entries, relative to the base address, are defined in S.MAC.

Notes:

1. Only a "long dispatch table" contains entries DEN through DMT. The long dispatch table is used for directory-oriented devices (DECtape, disk, and labelled magtape).

2. The actual tables are defined in the device service routines. The base address of the device dispatch table is contained in the corresponding Device Data Block, in the right half of DEVSER.

3. The device dispatch table labels are normally indexed by AC T4, which must contain the base address of the appropriate table. The dispatch is usually performed by a PUSHJ P, Dxxx(T4) where Dxxx represents the label of the appropriate table entry.

4. Before attempting to dispatch to any of the long dispatch table entries, the monitor checks the DVLNG bit of DEVMOD in the Device Data Block for that device.

5. The actual names of the tables are xyzDSP where xyz is the three-letter name of the device (for example, PTRDSP for the paper tape reader).

## 21 DISK CACHE DATA STRUCTURE

Description: The data structures described in this section are used by the monitor to implement a software disk cache.

The basic data structure consists of two doubly-linked lists, a list header, and a hash table. Each node in the list contains forward and backward pointers for each of the two lists to which it is linked (the pointers are .CBNHB, .CBPHB, .CBNAB, and .CBPAB), a UDB address (.CBUDB), a block number (.CBBLK), and a pointer to the address in FRECOR (.CBDAT) where the block is stored. For statistics purposes, the node also contains a count of the number of times this block has been hit since it was in the cache, named (.CBHIT).

The list header points to the two linked lists. The first linked list is the access list. The most recently accessed block is at the head of the list; the least recently accessed block is at the bottom of the list. This list is linked through the .CBNAB/.CBPAB words. The second linked list is the free list. It contains a list of all blocks that are not currently in use, and as such do not appear in the hash list described below. This list is linked through the .CBNHB/.CBPHB words.

The hash table consists of pointers into the .CBNHB/.CBPHB list for the corresponding list for blocks that hash to the same position. Thus, the hash table is really a number of separate list headers for the lists of blocks that hash to that position in the hash table.

At initialization time (CSHINI), all the blocks are allocated and linked into the free list. They are also linked into the access list. The hash table entries are linked to themselves because the table is empty.

To find an entry, given its UDB and block number, hash the block into the hash table, and, using that entry as a list head, follow the list until you find a match or return to the header. This is done with the routine CSHFND. In general, these lists are one or two blocks in length.

The main cache handling routine is CSHIO, which simulates I/O from the cache, doing the necessary physical I/O to fill and write the cache. Note that this is write-through cache, so no sweeps are required and the data in the cache always reflects the blocks on disk.

Format of CBHEAD List Header:

| Symbol | Map |
|---|---|
| .CBNHB | Pointer to first block in free list |
| .CBPHB | Pointer to last block in free list |
| .CBNAB | Pointer to first block in access list |
| .CBPAB | Pointer to last block in access list |

The following items are cached: RIBs, UDF data, SFD data, and SATs.

Format of Two-Word CBHSHT Hash Table Entry:

| Symbol | Map |
|---|---|
| .CBNHB | Pointer to first hash block in this chain |
| .CBPHB | Pointer to last hash block in this chain |

Format of Each List Entry:

| Symbol | Map |
|---|---|
| .CBNHB | Pointer to next hash block in this chain |
| .CBPHB | Pointer to previous hash block in this chain |
| .CBNAB | Pointer to next accessed block |
| .CBPAB | Pointer to previous accessed block |
| .CBUDB | UDB of unit containing this block |
| .CBBLK | Block number |
| .CBDAT | Pointer to 128(8) words for this disk block |
| .CBHIT | Count of hits for this disk block |

## 22 DISP AND DISP2 -- COMMAND DISPATCH TABLES

Description: Specify dispatch routine addresses for each monitor command. The DISP table contains addresses for monitor commands, and DISP2 contains addresses for monitor SET commands. Entries correspond to command names in COMTAB; the offset of the command in COMTAB is the index into DISP and DISP2.

Defined in: COMMON

Used by: COMCON

Map

```
+--------------------------------------------------------------+
|          Dispatch address for command 1                      |
|--------------------------------------------------------------|
|          Dispatch address for command 2                      |
|--------------------------------------------------------------|
|          Dispatch address for command 3                      |
|--------------------------------------------------------------|
                            .   .   .
```

## 23 DISPC -- CUSTOMER-DEFINED COMMAND DISPATCH TABLE

Description: Specify dispatch routine addresses for each customer-defined monitor command. Entries correspond to command names in CSTTAB; the offset of the command in CSTTAB is the index into DISPC.

Defined in: COMMON

Used by: COMCON

Map

```
+-------------------------------------------------------------+
|        Dispatch address for command 1                       |
|-------------------------------------------------------------|
|        Dispatch address for command 2                       |
|-------------------------------------------------------------|
|        Dispatch address for command 3                       |
|-------------------------------------------------------------|
                         .   .   .
```

## 24 DSCTAB -- DATASET CONTROL TABLE

Description: Contains information required for timing function on datasets. One entry for each data set that needs timing. Index is contained in Line Data Block, LDBBY2 word.

Defined in: COMDEV, COMMON, SCNSER

Each word in DSCTAB is formatted as follows:

```
|-------------------------------------------------------------------|
|HWC|SWC|FAI|NCR|BLI|DLW|DLF|DLC|EON| Res.| Time |  LINTAB Index   |
|-------------------------------------------------------------------|
  0   1   2   3   4   5   6   7   8  9--11 12--17  18------------35
```

| Bits | Symbol | Meaning |
|---|---|---|
| 0 | DSCHWC | When last heard from, the hardware carrier was on |
| 1 | DSCSWC | Software considers carrier to be on |
| 2 | DSCFAI | Carrier went off, but may be brief failure |
| 3 | DSCNCR | New carrier flag; on briefly for clock sync |
| 4 | DSCBLI | Blind flag; ignore everything for 1 second |
| 5 | DSCDLW | Dialer wait; waiting for results from dialer |
| 6 | DSCDLF | Dialer fail; unsuccessful dialer attempt |
| 7 | DSCDLC | Dialer complete; successful dialer action |
| 8 | DSCEON | End of number; sent all digits to dialer |
| 9-11 | | Unused. |
| 12-17 | DSTIMP | Time field for functions that require timing |
| 18-35 | | Line number for this dataset (LINTAB index) |

## 25 ECB -- ETHERNET CHANNEL BLOCK

Description: Contains information needed to control a system's access to the ethernet.

Defined in: ETHPRM

Used by: ETHSER

See also: EPB, EMB, EKB

| | |
|---|---|
| ECBSYS | Address of next ECB in system |
| ECBCID | Ethernet channel id |
| ECBSTS* | Ethernet channel status |
| ECBEAD | Ethernet channel address (2 words) |
| ECBEPB | Address of first EPB on this channel |
| ECBEKB | Address of first EKB on this channel |
| ECBCTR* | Ethernet channel counters block |

Notes:

1. The two words reserved for ECBEAD contain the channel's ethernet address stored as six 8-bit bytes left justified.

2. The ECBSTS word is described below.

25.1 ECBSTS -- Ethernet Channel Status Word

```
|--------------------------------------------------------------------|
| 0|                          Unused                                 |
|--------------------------------------------------------------------|
 0 1---------------------------------------------------------------35
```

| Bits | Symbol | Meaning |
|---|---|---|
| 0 | ECSONL | Channel is on-line |

25.2 ECBCTR -- Ethernet Channel Counters Block

Contains channel wide ethernet counters. Only updated on read channel counters or read kontroller counters function calls to ETHSER.

| | |
|---|---|
| CC.SLZ | Seconds since counters zeroed |
| CC.BYR | Bytes received |
| CC.BYX | Bytes transmitted |
| CC.DGR | Datagrams received |
| CC.DGX | Datagrams transmitted |
| CC.MBR | Multi-cast bytes received |
| CC.MDR | Multi-cast datagrams received |
| CC.DXD | Datagrams transmitted, initially deferred |
| CC.DX1 | Datagrams transmitted, single collision |
| CC.DXM | Datagrams transmitted, multiple collisions |
| CC.XMF | Transmit failures |
| CC.XFM* | Transmit failure bit mask |
| CC.RCF | Receive failures |
| CC.RFM* | Receive failure bit mask |
| CC.UFD | Unrecognized frame destination |
| CC.DOV | Data overrun |
| CC.SBU | System buffer unavailable |

| | |
|---|---|
| CC.UBU | User buffer unavailable |

### 25.2.1 CC.XFM -- Ethernet Channel Counters Transmit Failure Bit Mask

```
|-----------------------------------------------------------------------|
|                Unused                |XCL|XBP|XFD|XFL|XOC|XSC|XCC|XEC|
|-----------------------------------------------------------------------|
 0-----------------------------------27  28  29  30  31  32  33  34  35
```

| Bits | Symbol | Meaning |
|---|---|---|
| 28 | CCXCL | Carrier lost |
| 29 | CCXBP | Transmit buffer parity error |
| 30 | CCXFD | Remote failure to defer |
| 31 | CCXFL | Frame too long |
| 32 | CCXOC | Open circuit |
| 33 | CCXSC | Short circuit |
| 34 | CCXCC | Carrier check failed |
| 35 | CCXEC | Excessive collisions |

### 25.2.2 CC.RFM -- Ethernet Channel Counters Receive Failure Bit Mask

```
|-----------------------------------------------------------------------|
|                       Unused                      |RFP|RNB|RFL|RFE|RBC|
|-----------------------------------------------------------------------|
 0-----------------------------------------------30  31  32  33  34  35
```

| Bits | Symbol | Meaning |
|---|---|---|
| 31 | CCRFP | Free list parity error |
| 32 | CCRNB | No free buffers |
| 33 | CCRFL | Frame too long |
| 34 | CCRFE | Framing error |
| 35 | CCRBC | Block check error |

## 26 EKB -- ETHERNET KONTROLLER BLOCK

Description: Contains information needed to multiplex individual ethernet kontrollers into a single ethernet channel.

Defined in: ETHPRM

Used by: ETHSER

See also: ECB, EPB, EMB

| | |
| --- | --- |
| EKBSYS | Address of next EKB in system |
| EKBECB | Address of ECB for this EKB |
| EKBNXT | Address of next EKB on same ECB |
| EKBKTY* | Ethernet kontroller type |
| EKBKID | Ethernet kontroller id |
| EKBSTS* | Ethernet kontroller status |
| EKBKKB(1) | Address of kontroller's kontroller block |
| EKBKDA | Address of kontroller's function dispatch routine |
| EKBHEA(1) | Ethernet hardware address (2 words) |
| EKBCTR* | Ethernet kontroller counters block |

Notes:

1. The kontroller's kontroller block address and function dispatch routine address are supplied on the call to ETKINI in ETHSER by the kontroller service module when a kontroller is configured.

2. These two words contain the kontroller's physical ethernet address stored as six 8-bit bytes left justified.

## 26.1 EKBKTY -- Ethernet Kontroller Type Word

```
|----------------------------------------------------------------------|
| CPU | KTY | KNO |                    Unused                          |
|----------------------------------------------------------------------|
 0---2 3---5 6---8 9------------------------------------------------35
```

| Bits | Symbol | Meaning |
|---|---|---|
| 0-2 | EKYCPU | CPU number of kontroller |
| 3-5 | EKYKTY | Kontroller type code: |
| | .KTKNI | 1 = NIA20 |
| | .KTUNA | 2 = DEUNA (not implemented) |
| 6-8 | EKYKNO | Kontroller number |

## 26.2 EKBSTS -- Ethernet Kontroller Status Word

```
|----------------------------------------------------------------------|
|0| Unused | KST | PST |  Unused   | S|            Unused              |
|----------------------------------------------------------------------|
 0 1------2 3---5 6---8 9-------17 18 19---------------------------35
```

| Bits | Symbol | Meaning |
|---|---|---|
| 0 | EKSONL | Kontroller is online |
| 3-5 | EKSSTS | Current kontroller state (unused) |
| 6-8 | EKSPST | Previous kontroller state (unused) |
| 18 | EKSSEA | Need to set ethernet address |

## 26.3 EKBCTR -- Ethernet Kontroller Counters Block

Contains channel ethernet counters for a kontroller. Only updated on read channel counters or read kontroller counters function calls to ETHSER.

| | |
|---|---|
| KC.SLZ | Seconds since counters zeroed |
| KC.BYR | Bytes received |
| KC.BYX | Bytes transmitted |
| KC.DGR | Datagrams received |
| KC.DGX | Datagrams transmitted |
| KC.MBR | Multi-cast bytes received |
| KC.MDR | Multi-cast datagrams received |
| KC.DXD | Datagrams transmitted, initially deferred |
| KC.DX1 | Datagrams transmitted, single collision |
| KC.DXM | Datagrams transmitted, multiple collisions |
| KC.XMF | Transmit failures |
| KC.XFM* | Transmit failure bit mask |
| KC.RCF | Receive failures |
| KC.RFM* | Receive failure bit mask |
| KC.UFD | Unrecognized frame destination |
| KC.DOV | Data overrun |
| KC.SBU | System buffer unavailable |
| KC.UBU | User buffer unavailable |

### 26.3.1 KC.XFM -- Ethernet Kontroller Counters Transmit Failure Bit Mask

```
|-------------------------------------------------------------------|
|                 Unused                |XCL|XBP|XFD|XFL|XOC|XSC|XCC|XEC|
|-------------------------------------------------------------------|
0-----------------------------------27  28  29  30  31  32  33  34  35
```

| Bits | Symbol | Meaning |
|---|---|---|
| 28 | KCXCL | Carrier lost |
| 29 | KCXBP | Transmit buffer parity error |
| 30 | KCXFD | Remote failure to defer |
| 31 | KCXFL | Frame too long |
| 32 | KCXOC | Open circuit |
| 33 | KCXSC | Short circuit |
| 34 | KCXCC | Carrier check failed |
| 35 | KCXEC | Excessive collisions |

### 26.3.2 KC.RFM -- Ethernet Kontroller Counters Receive Failure Bit Mask

```
|-------------------------------------------------------------------|
|                       Unused                     |RFP|RNB|RFL|RFE|RBC|
|-------------------------------------------------------------------|
0-----------------------------------------------30  31  32  33  34  35
```

| Bits | Symbol | Meaning |
|---|---|---|
| 31 | KCRFP | Free list parity error |
| 32 | KCRNB | No free buffers |
| 33 | KCRFL | Frame too long |
| 34 | KCRFE | Framing error |
| 35 | KCRBC | Block check error |

## 27 EMB -- ETHERNET MULTICAST ADDRESS BLOCK

Description: Contains a multicast address enabled for datagram reception by an ethernet portal.

Defined in: ETHPRM

Used by: ETHSER

See also: ECB, EPB, EKB

```
          +-------------------------------------------------------------+
EMBNXT    |                Address of next EMB on this EPB              |
          |-------------------------------------------------------------|
EMBMCA    |                  Ethernet multicast address                 |
          |                          (2 words)                          |
          +-------------------------------------------------------------+
```

EMBMCA -- These two words contain an ethernet multicast address stored as six 8-bit bytes left-justified.

## 28 ENQ/DEQ BLOCKS

Description: Contains information about the ENQ/DEQ facility for a job. The Q-Block describes the user request for ENQ resources. The Lock Block describes the specific locks requested.

In the following block descriptions, the symbol for the left half of the word is listed first, and below it is the symbol for the right half of the word.

The format of the Q-block is:

| Symbol | Map | |
|---|---|---|
| .QBLJQ<br>.QBNJQ | Back ptr to last Q-block for this job | Forward ptr to next Q-block for this job |
| .QBJCH | Job context handle | Channel no. (.QBCHN) \| Flags (.QBFLG) |
| .QBLQ<br>.QBNQ | Ptr to last Queue Block for this lock | Ptr to next Queue Block for this lock |
| .QBLQR<br>.QBNQR | Ptr to last multiple Q-Block | Ptr to next multiple Q-Block |
| .QBRID<br>.QBNRP | Request-id | No. requested from pool |
| .QBGRP<br>.QBLB | Group no. | Ptr to Lock Block |
| .QBCHK<br>.QBMSK | Ptr to next Q-block to be checked for deadlock | Mask Block |

The format of the Lock Block is:

```
          +-------------------------------------------------------------+
.LBLHS    | Back ptr to last              |  Ptr to next                |
.LBNHS    | Lock Block in hash chain      |  Lock Block in hash chain   |
          |-------------------------------------------------------------|
.LBLEN    | Length of Lock Block          |  Flags                      |
.LBFLG    |                               |                             |
          |-------------------------------------------------------------|
.LBLQ     | Ptr to last Q-Block           | Ptr to next Q-Block         |
.LBNQ     | in chain                      | in chain                    |
          |-------------------------------------------------------------|
.LBLVL    |     Level number              | Addr of Access Table        |
.LBACC    |                               | (-2, -3, or 400000+jobn     |
          |-------------------------------------------------------------|
.LBPUL    |     Number in pool            | Number available            |
.LBAVL    |                               |                             |
          |-------------------------------------------------------------|
.LBTIM    |                        Time stamp                           |
          |-------------------------------------------------------------|
.LBTLN    |  Length of Table Block        |  Lock-associated Table      |
.LBTBL    |                               |                             |
          |-------------------------------------------------------------|
.LBNMS    |  Number of words in           |       Timer                 |
.LBPLT    |  Mask Block                   |                             |
          |-------------------------------------------------------------|
.LBTXT    |                    ASCIZ text, or                           |
          |                    500000,,0+user-code, or                  |
          |                    36-bit user code                         |
          +-------------------------------------------------------------+
```

## 29 EPB -- ETHERNET PORTAL BLOCK

Description: Contains ethernet protocol specific information for ethernet protocol users.

Defined in: ETHPRM

Used by: ETHSER

See also: ECB, EMB, EKB

| | |
|---|---|
| EPBNXT | Address of next EPB on same ECB |
| EPBEKB | Address of EKB assigned to this EPB |
| EPBPTY(1) | Ethernet Protocol Type |
| EPBJCH | JCH of Portal Owner |
| EPBKPB | Address of Kontroller's Protocol Block |
| EPBPID | Ethernet Portal Id |
| EPBSTS* | Ethernet Portal Status |
| EPBEMB | Address of first EMB on the EPB |
| EPBCBI(2) | Portal user's callback id |
| EPBCBA | Portal user's callback routine address |
| EPBCBU(3) | Portal user's callback UN block (23 words) |
| EPBBSZ | Receive datagram buffer size |
| EPBXBC | Current transmit datagram buffer count |
| EPBRBC | Current receive datagram buffer count |
| EPBCTR* | Ethernet portal counters block |

Notes:

1. The ethernet protocol type is stored as a 16-bit code right justified.

2. The portal user's callback id and callback routine address are supplied on the open portal function call to ETHSER.

3. The callback UN block is used at interrupt level by ETHSER to post interrupt level callbacks to the portal's user.

## 29.1 EPBSTS -- Ethernet Portal Status Word

```
|-------------------------------------------------------------------|
|0|P|I| CST | PST |                  Unused                         |
|-------------------------------------------------------------------|
 0 1 2 3---5 6---8 9-----------------------------------------------35
```

| Bits | Symbol | Meaning |
|---|---|---|
| 0 | EPSOPN | Portal is open |
| 1 | EPSPAD | Protocol uses padding |
| 2 | EPSINF | Portal is an information portal |
| 3-5 | EPSSTS | Current protocol state: |
| | .PSDIS | 0 = Disabled |
| | .PSDWE | 1 = Disabled, want to enable |
| | .PSEIP | 2 = Enable in progress |
| | .PSEPD | 3 = Enable in progress, want to disable |
| | .PSENA | 4 = Enabled |
| | .PSEWD | 5 = Enabled, want to disable |
| | .PSDIP | 6 = Disable in progress |
| 6-8 | EPSPST | Previous protocol state |

29.2 EPBCTR -- Ethernet Portal Counters Block

Contains portal specific ethernet counters. Only updated on read portal counters function call to ETHSER.

| | |
|---|---|
| PC.SLZ | Seconds since counters zeroed |
| PC.BYR | Bytes received |
| PC.DGR | Datagrams received |
| PC.BYX | Datagrams received |
| PC.DGX | Datagrams transmitted |
| PC.UBU | User buffer unavailable |

## 30 EPT -- EXEC PROCESS TABLE

Description: Contains information about the executive process and points to important addresses, like page maps. This table has been called the Exec Page Map Page (EPMP) in the past. However, the EPT is not a page map. It points to the appropriate page maps and other hardware-related instructions.

Each CPU constants GETTAB table (.GTCnC) contains a word (%CCTOS) that points to that CPU's EPT.

Defined in: COMMON.MAC

```
         +-----------------------------------------------------------+
  0      /        Eight channel logout areas (4 words each)          /
         |-----------------------------------------------------------|
 40      |                  I/O Page Fail trap                       |
         |-----------------------------------------------------------|
 41      |                       Unused                              |
         |-----------------------------------------------------------|
 42      /            Priority interrupt instructions                /
         /                     (16 words)                            /
         |-----------------------------------------------------------|
 60      /                  Channel fill words                       /
         /                      (4 words)                            /
         |-----------------------------------------------------------|
         /                        Unused                             /
         |-----------------------------------------------------------|
140      /                 4  DTE control blocks                     /
         /                    (10 words each)                        /
         |-----------------------------------------------------------|
200      /                       Unused                              /
         |-----------------------------------------------------------|
421      |         Arithmetic overflow trap instruction              |
         |-----------------------------------------------------------|
422      |        Push down list overflow trap instruction           |
         |-----------------------------------------------------------|
423      |               Trap 3 trap instruction                     |
         |-----------------------------------------------------------|
         /                        Unused                             /
         |-----------------------------------------------------------|
510      |         Time base for high-precision runtime              |
         |                      (2 words)                            |
         |-----------------------------------------------------------|
512      |             Performance analysis counter                  |
         |                      (2 words)                            |
         |-----------------------------------------------------------|
```

```
          |------------------------------------------------------------|
514       |            Interval timer interrupt instruction             |
          |------------------------------------------------------------|
         /                         Unused                              /
          |------------------------------------------------------------|
540       |            Address of Section 0 exec page map               |
          |------------------------------------------------------------|
541      /             Address of Sections 1-37 exec page maps         /
         /                 One word for each section                   /
          |------------------------------------------------------------|
600-777  /                         Unused                              /
         +-------------------------------------------------------------+
```

## 31 EXE FORMAT SAVE FILES

Defined in: S.MAC

Used by: COMCON, SEGCON

Description: The old .SAV, .LOW, .SHR, .HGH and .XPN files have been replaced by the .EXE type in order to provide a unified format for saved core image files. This type of file consists of two distinct but related portions:

1. Information about the structure of the file

2. The data in the file

The format of an executable file created by the monitor can have the following sections:

```
+--------------------------------------------------------------+
|                  Directory Section                           |
|--------------------------------------------------------------|
|                  Entry Vector Section (optional)             |
|--------------------------------------------------------------|
|                  End Vector Section                          |
|--------------------------------------------------------------|
|                  File data                                   |
+--------------------------------------------------------------+
```

Each section is one or more words describing the data in the file, introduced by an identifier code in the left half and the length of the chunk in the right half. At this time, the defined codes are:

1. Directory: 1776

2. Entry Vector: 1775

3. End: 1777

Other sections may be added later as they become necessary. Each section is described below:

The save file starts with a directory of the pages of data. All pages of data are stored on page boundaries in exactly the form they will be loaded into the virtual address space. There is no zero compression within pages; only entire pages will be compressed.

The format of the directory is:

```
Map
+-------------------------------------------------------------+
|          1776              |         Size of directory      |
|-------------------------------------------------------------|
|          Flags             |        File page number        |
|-------------------------------------------------------------|
|          Repeat            |      Process page number       |
|-------------------------------------------------------------|
|          Flags             |        File page number        |
|-------------------------------------------------------------|
|          Repeat            |      Process page number       |
|-------------------------------------------------------------|
                           .   .   .
|-------------------------------------------------------------|
|                        Entry Vector                         |
|-------------------------------------------------------------|
```

Word 0 has 1776 in the left half and the size of the directory in words in the right half.

Word 1 has flag bits in the leftmost nine bits and a file page number in the right 27 bits. If the file page number is zero, then the page is allocated but zero. The flag bits are:

| Bits | Symbol | Description |
|---|---|---|
| 0 | SV%HIS | Set if this is part of the hiseg |
| 1 | SV%SHR | Set if this page is sharable |
| 2 | SV%WRT | Set if the page may be written |
| 3 | SV%CON | Page is concealed (access by PORTAL only) |
| 4 | SV%SYM | Page is part of symbol table (unused) |
| 5 | SV%ABZ | Page is allocated but zero (internal flag) |

Word 2 has a repeat count in its leftmost nine bits for one less than the number of consecutive file and memory pages described by this pointer. The rightmost nine bits describe the process address into which this page should be loaded. Words 1 and 2 are repeated for each contiguous portion of the process space that has identical access bits.

The directory section is followed by the entry vector section. The entry vector is described by a 3-word entry vector block. Word 0 has the 1775 in the left half, and the word count of the entry vector block in the right half.

```
+-------------------------------------------------------------+
|                     Directory Section                       |
|-------------------------------------------------------------|
|       1775                  |          Word count           |
|-------------------------------------------------------------|
```

```
|                            (JRST)                            |
|--------------------------------------------------------------|
|                    30-bit Start Address                      |
|--------------------------------------------------------------|
|                         End Vector                           |
|--------------------------------------------------------------|
```

The entry vector is included only for multi-section programs. It is written in the following format by default:

Word 0 contains the code (1775) in the left half and the word count of the entry vector block (always 3) in the right half).

Word 1 contains the value 254000, which is the value of JRST with the halves reversed.

Word 2 contains the 30-bit starting address for the program.

Alternate forms of the entry vector block can be written. The entry vector block then points to the entry vector. In this case, Word 1 contains the length of the entry vector, and Word 2 contains the address of the entry vector. The address in Word 2 points to an Entry Vector, in the following format:

```
+--------------------------------------------------------------+
|                    30-bit Start Address                      |
|--------------------------------------------------------------|
|                    30-bit Reenter Address                    |
|--------------------------------------------------------------|
|                 Version number of program                    |
+--------------------------------------------------------------+
```

The entry vector block is followed by the End Block. The End Block word has the value 1777,,1.

The End Block is followed by the file data.

## 32 FETTBL -- TABLE OF BITS FOR MONITOR FEATURES

Description: Contains bits for feature test switches.

Defined in: UUOCON (local symbol)

Used by: UUOCON

GETTAB Table: .GTFET (71)

Map

| |
|---|
| Miscellaneous UUO features |
| Real time and scheduling features |
| Command features |
| Accounting features |
| Error control and options |
| Non-I/O debugging features |
| File system features |
| Internal disk features |
| Scanner features |
| Miscellaneous I/O features |
| Miscellaneous I/O features (second word) |
| Internal disk features (second word) |
| File system features (second word) |
| Miscellaneous UUO features (second word) |

## 33 FRAGMENT TABLE

Description: Contains data needed in order to swap in a fragmented core image. Set up as required when segments are swapped out. JBTSWP points to first fragment table for a given segment. This table is used only for high segments, and for low segments that are in the process of being swapped out.

Used by: SCHED1, SEGCON, SWPSER, VMSER

```
Map
+-------------------------------------------------------------+
|      Size                  |     |  U   |      Address      |
|-------------------------------------------------------------|
 0----------------------17  18-21 23----24 25-------------35
```

The bits in the fragment entry are defined below:

| Bits | Contents |
|---|---|
| 0-17 | Number of pages in fragment |
| 20-22 | Unit index into SWPTAB |
| 23-35 | Logical page within unit where fragment starts |

Notes:

1. A zero word indicates end of table.

2. Fragment tables are built from four-word blocks. If more than one block is required, and they are not contiguous, the last word of the preceding block contains -1 in the left half and the address of the next table in the right half.

## 34 HOM -- HOME BLOCK

Description: Block on each disk unit or pack that contains vital statistics that cannot be "built in" when a monitor is generated. These are primarily parameters of the unit or pack and the structure to which it belongs.

Defined in: COMMOD

Used by: MONBTS, ONCMOD, REFSTR

| Symbol | Map | |
|---|---|---|
| HOMNAM | SIXBIT /HOM/ (Written by MAP program.) | |
| HOMHID | SIXBIT unit ID (Written by MAP program.) | |
| HOMPHY* | Physical disk address of this block on this unit | Physical disk address of the other Home blk on this unit |
| HOMSRC | Position of this structure in System Search List<br>-1 means not in System Search List | |
| HOMSNM (2,3) | SIXBIT structure name 0 = not in file structure | |
| HOMNXT(3) | Unit ID for next unit in STR 0 = last or only unit | |
| HOMPRV(3) | Unit ID for previous unit in STR (0=last or only unit) | |
| HOMLOG | SIXBIT logical unit name within structure | |
| HOMLUN(3) | Logical unit number within structure | |
| HOMPPN | PPN that refreshed structure under timesharing, or 0 | |
| HOMHOM | Logical block # for Home block within unit | Logical block # for extra Home block within unit |
| HOMGRP | No. blocks per group to try for on output (not clusters) | |
| HOMBSC | No. blocks per supercluster on this unit | |
| HOMSCU | No. superclusters per unit | |
| HOMCNP | Byte pointer for cluster count in retrieval pointers | |
| HOMCKP | Byte pointer for checksum in retrieval pointers | |
| HOMCLP | Byte pointer for cluster address in retrieval pointers | |

| | |
|---|---|
| HOMBPC | No. blocks per cluster for this structure |
| HOMK4S | No. K words for swapping on this unit<br>0 means no swapping |
| HOMREF (7) | Non-zero if file structure must be refreshed |
| HOMSIC | No. SAT blocks in core |
| HOMSID | Unit ID of next unit in active swapping list<br>0 if last or not in active swapping list |
| HOMSUN | Logical unit # in active swapping list<br>-1 if not in swapping list |
| HOMSLB | First logical block # for swapping on this unit |
| HOMCFS | Swapping class for unit |
| HOMSPU | No. SAT blocks per unit |
| HOMOVR | Overdraw limit per user on this structure |
| HOMGAR | Upper bound on total reserved blocks guaranteed to user |
| HOMSAT | SAT.SYS (1) |
| HOMHMS (5) | HOME.SYS (1) |
| HOMSWP | SWAP.SYS (1) |
| HOMMNT | MAINT.SYS (1) |
| HOMBAD | BADBLK.SYS (1) |
| HOMCRS | CRASH.EXE (1) |
| HOMSNP | SNAP.SYS (1) |
| HOMRCV | RECOV.SYS (1) |
| HOMSUF | SYS [1,4].UFD (1) |
| HOMPUF | PRINTR [3,3].UFD (1) |
| HOMMFD | MFD [1,1].UFD (1) |

```
         |-------------------------------------------------------------|
HOMPT1   | First retrieval pointer for MFD for this unit's structure   |
         |-------------------------------------------------------------|
HOMUN1   | Logical unit # where MFD starts                             |
         |-------------------------------------------------------------|
HOMLEN   /Table of lengths of files created by refresh (6 words)       /
         / Needed by CRS, SNP, RCV, and UFDs (in that order)           /
         |-------------------------------------------------------------|
HOMUTP   | Unit type on which Home block was written                   |
         | (see UNYUTP)                                                |
         |-------------------------------------------------------------|
HOMRIP   | Reserved for use by RIPOFF                                  |
         |-------------------------------------------------------------|
HOMKLB   / 20 words used by console front end in KL10 systems          /
         |-------------------------------------------------------------|
HOMFEA   | FE-file address for KS10                                    |
         |-------------------------------------------------------------|
HOMFES   | FE-file size for KS10                                       |
         |-------------------------------------------------------------|
HOMTCS   | Tracks/cylinder/sector for KS10                             |
         |-------------------------------------------------------------|
HOMKLE   | Used to find files for bootstrap/dump                       |
         |-------------------------------------------------------------|
HOMK4C   | K for CRASH.EXE                                             |
         |-------------------------------------------------------------|
HOMPVS   | Word containing bit which says private structure            |
         |-------------------------------------------------------------|
HOMSDL   | Position of structure in system dump list                   |
         |-------------------------------------------------------------|
HOMOPP   | PPN of structure owner                                      |
         |-------------------------------------------------------------|
HOMMSU   | Reserved for use by DIGITAL                                 |
         |-------------------------------------------------------------|
HOMCUS   / Reserved for customer definition (4 words)                  /
         |-------------------------------------------------------------|
HOMVID   / Volume-ID (12 PDP-11 bytes = 3 words)                       /
         |-------------------------------------------------------------|
HOMOWN   | Owner name                                                  |
         |-------------------------------------------------------------|
HOMVSY   | System type (TOPS-10)                                       |
         |-------------------------------------------------------------|
HOMCOD   |          0             |  (Unlikely code) 707070            |
         |-------------------------------------------------------------|
HOMSLF   |          0             | This blk # in unit                 |
         |                        | (not cluster)                      |
         +-------------------------------------------------------------+
```

Notes:

1. This value is the logical block number of first RIB for the file.

2. HOMSNM (structure name) is ignored by the monitor when the structure is mounted.

3. HOMSNM, HOMNXT, HOMPRV, and HOMLUN are checked by the monitor at system startup; and by PULSAR when the unit comes on line, to determine whether a complete structure is available that can be mounted.

4. An extra copy of the Home block is maintained on each unit in case the original becomes unreadable. These two Home blocks which must be at specific places known to the software. Currently, they are blocks 1 and 10 (decimal) of each unit.

5. Each file structure has one Home file HOME.SYS. It is a "Sparse File" with retrieval information in first block of each group. However, only the second block of each group (1 cluster) has data, the remaining blocks (if any) are 0. The Home block is constructed so that each group is on a separate unit.

6. Home blocks are limited to the first 262,000 blocks on a unit. Knowing the logical block address of Home blocks is useful only to recovery programs when file structure has been clobbered. Most programs will read Home blocks using HOME.SYS.

7. HOMREF is set by ONCE-only code when some parameter for this unit has been changed. It is checked when the system is started and by PULSAR when the pack is mounted.

* The bytes in HOMPHY are described below:

HOMPHY -- Physical Address

```
|-------------------------------------------------------------------|
|        C        |  S  |  A  |          C          |  S  |  A  |
|-------------------------------------------------------------------|
0---------------7 8---12 13--17 18------------------25 26--30 31--35
```

| Byte | Meaning |
|---|---|
| C | Cylinder address |
| S | Surface |
| A | Sector address |

## 35 INTTAB -- INTERRUPT ROUTINE TABLE

Description: Contains descriptive information about each interrupt routine. One entry, of two words, for each interrupt routine.

Position of an entry in the table is of no significance.

Defined in: COMMON

Used by: COMMON, ONCE

```
           +------------------------------------------------------------+
Device 1   | D | DDB count | CPU# | PI channel |Interrupt routine adr(4)|
           |------------------------------------------------------------|
           |     | Station # |  DDB Length    | Prototype DDB address    |
           |------------------------------------------------------------|
Device 2   | D | DDB count | CPU# | PI channel |Interrupt routine adr(4)|
           |------------------------------------------------------------|
           |     | Station # |  DDB Length    | Prototype DDB address    |
           |------------------------------------------------------------|
                                   .   .   .
           |------------------------------------------------------------|
Device n   | D | DDB count | CPU# | PI channel |Interrupt routine adr(4)|
           |------------------------------------------------------------|
           |     | Station # |  DDB Length    | Prototype DDB address    |
           +------------------------------------------------------------+
```

Format of each two-word entry:

Word 0

```
|--------------------------------------------------------------------|
| D |  DDB count   |  CPU#  |  PI channel  | Interrupt routine adr (4)|
|--------------------------------------------------------------------|
 0  1----------11 12----14 15----------17 18----------------------35
```

Bit 0 = 1 for any type of DECtape routine.

Word 1

```
|--------------------------------------------------------------------|
|      | Station # |   DDB Length |        Prototype DDB address      |
|--------------------------------------------------------------------|
 0---2 3----------8   9-----------17 18--------------------------35
```

Notes:

1. INTTAB entries are set up by the ASGINT and ASGSV1 macros in COMMON according to parameters specified in HDWCNF.MAC.

2. Table is used by INTLOP in ONCE to build the interrupt routine chain for each channel.

3. Used by ONCE-only code in COMMON to set up multiple Device Data Blocks.

4. The interrupt routine address is the address of the CONSO in the skip chain.

## 36 JBTADR -- JOB ADDRESS TABLE

Description: Contains the core address and length for each segment in core. One entry for each job number and each high segment number.

Defined in: COMMON

Used by: CDPSER, CLOCK1, COMCON, CORE1, CPNSER, DISSER, DTASRN, ERRCON, FILIO, KLSER, LOKCON, METCON, REFSTR, SCHED1, SEGCON, SWPSER, SYSINI, UUOCON, VMSER

GETTAB Table: .GTADR (1)

| | | |
|---|---|---|
| Job 0 | Length -1 | Relocation address |
| Job 1 | Length -1 | Relocation address |
| | . . . | |
| Job n | Length -1 | Relocation address |
| High Segment n+1 | 0 | Relocation address |
| n+2 | 0 | Relocation address |
| | . . . | |

Notes:

1. A job that is not in core has zero in its entry.

2. This table is the same as JBTDAT.

3. The left half of JBTADR contains the highest page number in the job's Section 0 page map.

4. For low segments, the right half of JBTADR contains .JDAT. For high segments, the symbolic contents of the right half is .VJDT.

   If the segment is locked into contiguous memory locations, JBTADR contains the first physical page number where the job resides.

5. The Length field applies only to the job's low segment in Section 0.

## 37 JBTAD2 -- SECOND JOB ADDRESS TABLE

Description: This table contains the first physical page number for each job's low segment. This table is kept so that the monitor can obtain the page number when the job is not currently mapped or the UPT is in error. There is one entry per job.

Defined in: COMMON

Used by: CORE1, ERRCON, KLSER, LOKCON, VMSER

```
          +-------------------------------------------------------+
Job 0     |                         |    First physical page      |
          |-------------------------------------------------------|
Job 1     |                         |    First physical page      |
          |-------------------------------------------------------|
                                 .    .    .
          |-------------------------------------------------------|
Job n     |                         |    First physical page      |
          +-------------------------------------------------------+
```

38 JBTCHK -- SEGMENT CHECKSUM TABLE

Description: Used to assure correct read-in of swapped segments. One entry for each job number and high seg number. Indexed by job number or high segment number.

Each entry contains the first word of the corresponding segment when that segment is swapped out.

Defined in: COMMON

Used by: SWPSER

```
             +-------------------------------------------------------------+
Job 0        |                     Checksum 0                              |
             |-------------------------------------------------------------|
Job 1        |                     Checksum 1                              |
             |-------------------------------------------------------------|
Job 2        |                     Checksum 2                              |
             |-------------------------------------------------------------|
                                 .    .    .
             |-------------------------------------------------------------|
Job n        |                     Checksum n                              |
             |-------------------------------------------------------------|
High         |                                                             |
Segment      |                     Checksum n+1                            |
n+1          |                                                             |
             |-------------------------------------------------------------|
n+2          |                     Checksum n+2                            |
             |-------------------------------------------------------------|
                                 .    .    .
```

## 39 JBTCLM -- JOB CORE LIMITS

Description: Contains the core limit for each job. Right half is JBTDDB.

Defined in: COMMON

Used by: COMCON, CORE1, VMSER

See also: JBTDDB

```
          +------------------------------------------------------------+
Job 0     |    Core limit for job       |     JBTDDB                   |
          |------------------------------------------------------------|
Job 1     |    Core limit for job       |     JBTDDB                   |
          |------------------------------------------------------------|
                                  .    .    .
          |------------------------------------------------------------|
Job n     |    Core limit for job       |     JBTDDB                   |
          +------------------------------------------------------------+
```

## 40 JBTCQ -- JOB QUEUES TABLE

Description: Contains the master job queues. Each queue is an ordered list of job numbers corresponding to all the jobs in some particular status. There are two sets of queues, one for jobs that are in core and one for jobs that are swapped out.

The index in the positive direction relative to JBTCQ is by job number. The index in the negative direction is the queue number for the in-core master queues and the queue number offset by the maximum number of queues for the out-of-core master queues.

Each queue is an ordered list of job numbers. In JBTCQ, each queue is represented by a forward and backward linked ring of table entries. Each ring begins and ends with a "queue header" entry at the position equal to the negative queue number. A job number in the queue is represented by the entry at the position equal to the job number. This entry contains pointers to the preceding entry and the following entry, thus establishing a unique position for that job number within the queue.

Defined in: COMMON

Used by: SCHED1, SYSINI

See also: JBTSTS

```
           +------------------------------------------------------------+
 -MAXQ -3  |     Last job in queue      |        First job in queue       |
           |------------------------------------------------------------|
 -MAXQ -2  |     Last job in queue      |        First job in queue       |
           |------------------------------------------------------------|
 -MAXQ -1  |     Last job in queue      |        First job in queue       |
           |------------------------------------------------------------|
 -MAXQ                              . . .
           |------------------------------------------------------------|
       -3  |     Last job in queue      |        First job in queue       |
           |------------------------------------------------------------|
Queue  -2  |     Last job in queue      |        First job in queue       |
Number     |------------------------------------------------------------|
       -1  |     Last job in queue      |        First job in queue       |
           |------------------------------------------------------------|
 JBTCQ  0  |     Last job in queue      |        First job in queue       |
           |------------------------------------------------------------|
        1  |  Number of previous job    |        Number of next job       |
           |------------------------------------------------------------|
Job     2  |  Number of previous job    |        Number of next job       |
Number     |------------------------------------------------------------|
```

```
3 |  Number of previous job   |       Number of next job        |
  |-------------------------------------------------------------|
                          .   .   .
```

Notes:

1. Every job number will be in one and only one queue.

2. It is possible for a queue to contain no job numbers. In this case, the queue header entry contains a pointer to itself, the negative queue number, in both halves.

## 41 JBTCSQ -- SUBQUEUES TABLE

Description: Contains the PQ2 subqueues that partition PQ2 into a number of ordered classes. Each subqueue (or subclass) is an ordered subset of PQ2 job numbers, all of the same class.

Defined in: COMMON

Used by: SCHED1, SYSINI

```
                                +-----------------------------------------------+
                       -SQn     | Last job in subqueue |First job in subqueue   |
                                |-----------------------------------------------|
                       -SQ1     | Last job in subqueue |First job in subqueue   |
Subqueue                        |-----------------------------------------------|
Number                 -SQ0     | Last job in subqueue |First job in subqueue   |
                                |-----------------------------------------------|
JBTCSQ                          |                                               |
                                |-----------------------------------------------|
                          1     |Number of previous job| Number of next job     |
Job                             |-----------------------------------------------|
Number                    2     |Number of previous job| Number of next job     |
                                |-----------------------------------------------|
                          3     |Number of previous job| Number of next job     |
                                +-----------------------------------------------+
                                          .    .    .
```

Notes:

1. Every job number in PQ2 will be in one and only one subqueue.

2. It is possible for a subqueue to contain no job numbers. In this case, the subqueue header entry contains a pointer to itself, the negative subqueue number, in both halves.

3. The number of classes partitioning PQ2 is specified at MONGEN time.

4. When the scheduler scans these subqueues, it does so in order, starting with SQ0 (class 0).

## 42 JBTDDB -- DDB REQUESTED BY JOB

Description: One entry for each job, containing JBTCLM (job core limit) in the left half, and the DDB that the job is waiting for, in the right half.

Defined in: COMMON

Used by: COMCON

See also: JBTCLM

```
        +--------------------------------------------------------------+
Job 0   |    JBTCLM                      |     DDB requested            |
        |--------------------------------------------------------------|
Job 1   |    JBTCLM                      |     DDB requested            |
        |--------------------------------------------------------------|
                                     .   .   .
        |--------------------------------------------------------------|
Job n   |    JBTCLM                      |     DDB requested            |
        +--------------------------------------------------------------+
```

## 43 JBTDEV -- HIGH SEGMENT DEVICE TABLE

Description: One entry for each high segment.

The entry is the high segment's physical device name or file structure name. (Job number entries are not used. This table overlaps the high end of the JBTPPN table.)

Defined in: COMMON

Used by: SEGCON, UUOCON

GETTAB Table: .GTDEV (24)

```
High          +-------------------------------------------------------+
Segment       |              Physical device name                     |
         n+1  |              or file structure name                   |
              |-------------------------------------------------------|
         n+2  |              Physical device name                     |
              |              or file structure name                   |
              |-------------------------------------------------------|
                             .     .     .
```

## 44 JBTDTC -- JOB REQUESTING DECTAPE DDB

Description: Contains one entry for each job, with the address of the master DECtape DDB that contains the DT resource the job desires in the left half, and the right half contains JBTIPC. JBTDTC is non-zero when the job is waiting for a DECtape DDB (in DT resource wait state), or owns the DT resource for the specified master DECtape DDB.

Defined in: COMMON

Used by: DTASER, SCHED1

See also: JBTIND

```
         +--------------------------------------------------------------+
Job 0    | Master DECtape DDB address   |    JBTIPC                     |
         |--------------------------------------------------------------|
    1    | Master DECtape DDB address   |    JBTIPC                     |
         |--------------------------------------------------------------|
                                       .   .   .
         |--------------------------------------------------------------|
    n    | Master DECtape DDB address   |    JBTIPC                     |
         +--------------------------------------------------------------+
```

## 45 JBTIMI -- JOB PAGE COUNT

Description: One entry for each job containing the number of physical pages in the user portion of the job, referenced by byte pointer IMGIN.

Defined in: COMMON

Used by: CORE1, SCHED1, SEGCON, SWPSER, VMSER

See also: JBTIMO

```
          +---------------------------------------------------------+
Job 0     | | NZSICN | NZSSCN |   Physical page count of user area |
          |---------------------------------------------------------|
Job 1     | | NZSICN | NZSSCN |   Physical page count of user area |
          |---------------------------------------------------------|
                                  .   .   .
          |---------------------------------------------------------|
Job n     | | NZSICN | NZSSCN |   Physical page count of user area |
          +---------------------------------------------------------+
```

Bit definitions:

| Bits | Byte Pointer | Contents |
|---|---|---|
| 0-2 | | Reserved |
| 3-8 | NZSICN | Byte pointer to number of pages to allocate on swap-in for Non-Zero Section (NZS) maps. |
| 9-14 | NZSSCN | Byte pointer to number of pages allocated to NZS page maps. |
| 15-35 | IMGIN | Number of physical pages in user portion of job. |

46 JBTIMO -- SWAPPED-OUT PAGE COUNT

Description: Contains the number of physical pages in swapped-out job (that is, the number of pages on disk). This table is referenced by byte pointers IMGOUT.

Defined in: COMMON

Used by: CORE1, SEGCON, SWPSER, VMSER

See also: JBTIMI

```
         +-------------------------------------------------------------+
Job 0    |                              | No. of pages on disk         |
         |-------------------------------------------------------------|
Job 1    |                              | No. of pages on disk         |
         |-------------------------------------------------------------|
                                             .   .   .
         |-------------------------------------------------------------|
Job n    |                              | No. of pages on disk         |
         +-------------------------------------------------------------+
```

| Bits | Symbol | Contents |
|---|---|---|
| 0-14 | | Reserved |
| 15-35 | IMGOUT | Page count |

## 47 JBTIPC -- REQUESTED IPCF INTERLOCK

Description: Contains one entry for each job. Each entry contains JBTDTC in the left half, and the address of the Exec IPCF interlock that the job is waiting for (or owns) in the right half. The right half is non-zero when the job is in IP resource wait state or owns the IPCF interlock.

Defined in: COMMON

Used by: SCHED1, IPCSER

See also: JBTDTC

```
          +-------------------------------------------------------------+
Job 0     |    JBTDTC                      |     Interlock address      |
          |-------------------------------------------------------------|
Job 1     |    JBTDTC                      |     Interlock address      |
          |-------------------------------------------------------------|
                                        .   .   .
          |-------------------------------------------------------------|
Job n     |    JBTDTC                      |     Interlock address      |
          +-------------------------------------------------------------+
```

## 48 JBTJIL -- JUST-SWAPPED-IN LIST

Description: Special queue containing jobs in PQ2 that have just been swapped in and have not expired their time slice. The format of the queue table is similar to JBTCQ, that is indexed in the positive direction by job number and the negative direction by a queue number (not the same queue number that is used in JBTCQ).

The three queue headers are regular (timesharing), background batch, and jobs whose low segments are in core and are waiting for swapping I/O to be finished for a high segment.

Defined in: COMMON

Used by: SCHED1, SYSINI

See also: JBTOLS

| | | |
|---|---|---|
| JBTJIL | Last job in queue | First job in queue |
| Job 1 | No. of previous job | No. of next job |
| 2 | No. of previous job | No. of next job |
| 3 | No. of previous job | No. of next job |
| | . . . | |
| n | No. of previous job | No. of next job |

Notes:

The scheduler will search the timesharing queue for job selection ahead of PQ2 a certain percentage of the time. This percentage is called the response fairness factor and may be modified from the default value of 10% by the SCHED. UUO.

## 49 JBTJRQ -- REQUEUE LIST

Description: A singly linked first in last out list of jobs waiting to be requeued. The right half of JBTJRQ points to the first entry in the queue. The remainder of the table is indexed by job number with each entry containing the job number of the next job in the queue.

Defined in: COMMON

Used by: CLOCK1, SCHED1

```
              +--------------------------------------------------------+
JBTJRQ        |                          |  No. of 1st job in the list |
              |--------------------------------------------------------|
Job  1        |              No. of next job in the list               |
              |--------------------------------------------------------|
     2        |              No. of next job in the list               |
              |--------------------------------------------------------|
     3        |              No. of next job in the list               |
              |--------------------------------------------------------|
                                     .   .   .
              |--------------------------------------------------------|
     n        |              No. of next job in the list               |
              +--------------------------------------------------------+
```

## 50 JBTLIM -- JOB TIME LIMIT TABLE

Description: One entry per job, indexed by job number. Bit definitions are found in S.MAC. JBTLIM is assembled only if FTTLIM is non-zero.

Defined in: COMMON

Used by: CLOCK1, COMCON, CORE1, FILFND, IPCSER, SCNSER, UUOCON

GETTAB Table: .GTLIM (40)

```
              +-----------------------------------------------------------+
Job 0         |LTL|   LCR    | LBT | LSY |        JB.LTM                  |
              |-----------------------------------------------------------|
    1         |LTL|   LCR    | LBT | LSY |        JB.LTM                  |
              |-----------------------------------------------------------|
    2         |LTL|   LCR    | LBT | LSY |        JB.LTM                  |
              |-----------------------------------------------------------|
                                       .   .   .
              |-----------------------------------------------------------|
    n         |LTL|   LCR    | LBT | LSY |        JB.LTM                  |
              +-----------------------------------------------------------+
```

| Bit | Mask | Pointer | Content |
|---|---|---|---|
| 0 | JB.LTL | | Set if time limit set by forced DETACH. |
| 1-9 | JB.LCR | JBYLCR | User core limit (in pages). |
| 10 | JB.LBT | | Batch job. |
| 11 | JB.LSY | | Set when program came from SYS. |
| 12-35 | JB.LTM | JBLTM | Time limit in jiffies (0 = infinite). |

## 51 JBTLOC -- JOB LOCATION TABLE

Description: One entry per job number; indexed by job number.

The first entry in the table is the central station number.

The table is used by the LOCATE command and by the NEWJOB routine in COMMON to locate a job at the station number of its controlling terminal.

JBTLOC is assembled only if FTNET is non-zero.

Defined in: COMMON

Used by: AUTCON, COMCON, COMDEV, IPCSER, NETSER, PTYSER, SYSINI, UUOCON

GETTAB Table: .GTLOC (26)

```
              +-----------------------------------------------------+
              |             Central site station number             |
              |-----------------------------------------------------|
Job  1        |                    Job location                     |
              |-----------------------------------------------------|
     2        |                    Job location                     |
              |-----------------------------------------------------|
     3        |                    Job location                     |
              |-----------------------------------------------------|
                                    .   .   .
              |-----------------------------------------------------|
     n        |                    Job location                     |
              +-----------------------------------------------------+
```

52 JBTNAM -- JOB NAME TABLE

Description: Tells SIXBIT name of each segment; typically, this is the file it came from. One entry for each job number or high segment number. Indexed by job number or high segment number. JBTPRG is the same table.

Defined in: COMMON

Used by: CLOCK1, COMCON, ERRCON, FILFND, IPCSER, KLSER, LPTSER, NETSER, PTYSER, SCNSER, SEGCON, TAPUUO, TSKSER, UUOCON

GETTAB Table: .GTPRG (3)

```
                 +-------------------------------------------------------+
  Job     0      |                                                       |
                 |-------------------------------------------------------|
          1      |                  SIXBIT/segment1/                     |
                 |-------------------------------------------------------|
          2      |                  SIXBIT/segment2/                     |
                 |-------------------------------------------------------|
          3      |                  SIXBIT/segment3/                     |
                 |-------------------------------------------------------|
                                     .    .    .
                 |-------------------------------------------------------|
          n      |                  SIXBIT/segmentn/                     |
                 |-------------------------------------------------------|
High             |                  SIXBIT/segmentn+1/                   |
Segment n+1      |                                                       |
                 |-------------------------------------------------------|
        n+2      |                  SIXBIT/segmentn+2/                   |
                 |-------------------------------------------------------|
                                    .    .    .
```

## 53 JBTOLS -- JOB OUTPUT LIST

Description: Special queues containing jobs in PQ2 which have exceeded their time slice and consequently have become eligible for swap out. The format of the queue table is similar to that of JBTCQ, in that the table is indexed in the positive direction by job number, and the negative direction by queue number. The two queue headers are regular output (timesharing) and background batch.

Defined in: COMMON

Used by: SCHED1, SYSINI

See also: JBTJIL

| | | |
|---|---|---|
| JBTOBQ | Last job in queue | First job in queue |
| JBTOLQ | Last job in queue | First job in queue |
| JBTOLS | Last job in queue | First job in queue |
| Job 1 | No. of previous job | No. of next job |
| 2 | No. of previous job | No. of next job |
| 3 | No. of previous job | No. of next job |
| | . . . | |
| n | No. of previous job | No. of next job |

Notes:

Once a job enters JBTOLS it has a higher priority for swap out.

54 JBTPC -- USER MODE PC

```
Description:    Contains user program counter for each job in the
                system.

Defined in:     COMMON

                Used by:  CLOCK1, COMCON, ERRCON, UUOCON

GETTAB Table:   .GTPC (152)

               +-----------------------------------------------------------+
Job     0      |                        Full-word PC                       |
               |-----------------------------------------------------------|
        1      |                        Full-word PC                       |
               |-----------------------------------------------------------|
        2      |                        Full-word PC                       |
               |-----------------------------------------------------------|
                                          .   .   .
               |-----------------------------------------------------------|
        n      |                        Full-word PC                       |
               +-----------------------------------------------------------+
```

## 55 JBTPDB -- PROCESS DATA BLOCK TABLE

Description: One entry per job, indexed by job number. The right half contains the address of this job's Process Data Block.

Defined in: COMMON

Used by: CLOCK1, COMCON, DATMAN, ERRCON, IPCSER, KLSER, MOSSER, NETSER, SCHED1, SCMUUO, UUOCON, VMSER

GETTAB Table: .GTPDB (162)

```
          +--------------------------------------------------------+
Job   0   |      | IFYPGS | NFYPGS |         PDB address            |
          |--------------------------------------------------------|
      1   |      | IFYPGS | NFYPGS |         PDB address            |
          |--------------------------------------------------------|
      2   |      | IFYPGS | NFYPGS |         PDB address            |
          |--------------------------------------------------------|
                                .    .    .
          |--------------------------------------------------------|
      n   |      | IFYPGS | NFYPGS |         PDB address            |
          +--------------------------------------------------------+
```

Bit definitions:

| Bit | Symbol | Contents |
|---|---|---|
| 0-5 | | Reserved |
| 6-11 | IFYPGS | Byte pointer to the number of funny pages allocated on swap-in. |
| 12-17 | NFYPGS | Byte pointer to the number of funny pages allotted to the user. |
| 18-35 | | PDB address. |

## 56 JBTPIA -- SOFTWARE PROGRAM INTERRUPT TABLE

Description: Contains flags and pointers to program interrupt table for each job that is using the programmed software interrupt facility.

Defined in: COMMON

Used by: IPCSER, PSISER, UUOCON

```
                +---------------------------------------------------+
Job      0      | 0  | 1  |              | Interrupt table address   |
                |---------------------------------------------------|
         1      | 0  | 1  |              | Interrupt table address   |
                |---------------------------------------------------|
         2      | 0  | 1  |              | Interrupt table address   |
                |---------------------------------------------------|
                                   .    .    .
                |---------------------------------------------------|
         n      | 0  | 1  |              | Interrupt table address   |
                +---------------------------------------------------+
```

Bit definitions:

Bit 0 is set if the PI system is on for this job.

Bit 1 is set if the PI system is turned off until this job issues a DEBRK.

## 57 JBTPPN -- PROJECT PROGRAMMER NUMBER TABLE

| | |
|---|---|
| Description: | Contains the project programmer number for each logged in job. if the high segment came from disk, this table contains the PPN for path pointer of the directory from which the high segment came. |
| Defined in: | COMMON |
| Used by: | CLOCK1, COMCON, ERRCON, FILFND, FILIO, FILUUO, IPCSER, KLSER, LPTSER, NETSER, SCLINK, SCNSER, SEGCON, TAPUUO, TSKSER, UUOCON, |
| GETTAB Table: | .GTPPN (2) |

```
                 +-------------------------------------------------------+
  Job     0      |                                                       |
                 |-------------------------------------------------------|
          1      |               Project-programmer number               |
                 |-------------------------------------------------------|
          2      |               Project-programmer number               |
                 |-------------------------------------------------------|
          3      |               Project-programmer number               |
                 |-------------------------------------------------------|
                                      .    .    .
                 |-------------------------------------------------------|
High      n      |               Project-programmer number               |
Segment          |-------------------------------------------------------|
         n+1     |                   0 or path pointer                   |
                 |-------------------------------------------------------|
         n+2     |                   0 or path pointer                   |
                 |-------------------------------------------------------|
                                     .    .    .
```

58 JBTPRV -- JOB PRIVILEGE BITS

Description: Tells privileges allowed each job. One entry for each job number, indexed by job number. Bits are set by LOGIN from ACCT.SYS File as modified by customer for his particular installation.

Defined in: COMMON. Bits defined in S.UNV.

Used by: CLOCK1, COMCON, COMMOD, CPNSER, FILFND, IPCSER, RTTRP, UUOCON

GETTAB Table: .GTPRV (6)

Each entry in JBTPRV is formatted as follows:

```
|--------------------------------------------------------------------|
|IPC|DPR|MET|POK|CCC| HPQ |NSPL|ENQ|   |RTT|LOCK|TRPS|SPYA|SPYM|      |
|--------------------------------------------------------------------|
  0  1-2  3   4   5  6-9    10    11 12  13  14    15   16   17  18--35
```

| Bits | Symbol | Meaning |
|---|---|---|
| 0 | JP.IPC | Job allowed to use IPCF priv. functions |
| 1-2 | JP.DPR | Highest disk priority allowed to the job |
| 3 | JP.MET | Job allowed to use METER UUO |
| 4 | JP.POK | Job allowed to use POKE UUO |
| 5 | JP.CCC | Job allowed to change CPU specifications |
| 6-9 | PVHPQ | Largest HPQ run queue for this job |
| 10 | PVNSPL | Job allowed to unspool devices |
| 11 | JP.ENQ | Job allowed to use ENQ/DEQ |
| 12 | JP.ADM | Job has system administrator privileges and may create user accounts with REACT |
| 13 | PVRTT | Job allowed to use RTTRP UUO |
| 14 | PVLOCK | Job allowed to use LOCK UUO |
| 15 | PVTRPS | Job allowed to use TRPSET UUO |
| 16 | PVSPYA | Job allowed to spy at all of core using SPY/PEEK UUOs |
| 17 | PVSPYM | Job allowed to spy at monitor using SPY/PEEK UUOs |
| 18-35 | | Reserved for customer-defined privileges |

## 59 JBTRQT -- JOB RUN-QUEUE TIME TABLE

Description: One entry per job number. Contains a count of the number of ticks a job was in PQ1 or PQ2.

Defined in: COMMON

Used by: CLOCK1, SCHED1, UUOCON

GETTAB Table: .GTTRQ (53)

```
                +---------------------------------------------------------+
Job      0      |                  "Want to run" time                     |
                |---------------------------------------------------------|
         1      |                  "Want to run" time                     |
                |---------------------------------------------------------|
         2      |                  "Want to run" time                     |
                |---------------------------------------------------------|
                                     .    .    .
                |---------------------------------------------------------|
         n      |                  "Want to run" time                     |
                +---------------------------------------------------------+
```

Notes:

JBTRQT is assembled only if FTRSP is non-zero, and is also updated only if the monitor is patched at location RQTPAT to enable it.

60 JBTRSP -- RESPONSE TIME TABLE

Description: This table contains one entry per job number, including the null job. The null job entry is not referenced.

Defined in: COMMON. Bits are defined in S.UNV.

Used by: CLOCK1, UUOCON

GETTAB Table: .GTRSP (50)

```
                 +-------------------------------------------------------+
Job      0       | C | O | I | R | X |       Time job started waiting    |
                 |-------------------------------------------------------|
         1       | C | O | I | R | X |       Time job started waiting    |
                 |-------------------------------------------------------|
         2       | C | O | I | R | X |       Time job started waiting    |
                 |-------------------------------------------------------|
                                      .   .   .
                 |-------------------------------------------------------|
         n       | C | O | I | R | X |       Time job started waiting    |
                 +-------------------------------------------------------+
```

| Bit | Name | Description |
|---|---|---|
| 0 | JR.RCR | Recorded first CPU use |
| 1 | JR.ROR | Recorded TTY output UUO |
| 2 | JR.RIR | Recorded TTY input UUO |
| 3 | JR.RRR | Recorded CPU quantum exceeded requeue |
| 4 | JR.RXR | Recorded first of above 3 responses |

The bits are set to 1 when the type of response is recorded; set to 0 when user types in.

Notes:

1. The entries in the right half of this table are the uptimes at which the user began to wait for system response. If zero, the user is not waiting. Entries are made in the table when the user's job comes out of TTY Input Wait or types a command that runs a program. The entries are used by the scheduler to calculate entries in each CPU's response subtable.

2. JBTRSP is assembled only if FTRSP is non-zero.

## 61 JBTRTD -- REAL TIME DEVICES

| | |
|---|---|
| Description: | Contains real time status bits in the left half for HPQs and hibernate-wake. The right half contains a count of the real time devices owned by the job. |
| Defined in: | COMMON |
| Used by: | CLOCK1, IPCSER, PTYSER, RTTRP, SCMUUO, SCNSER, SYSINI, UUOCON |
| GETTAB Table: | .GTRTD (37) |

```
                +---------------------------------------------------+
Job       0     |           Status bits            |  Count         |
                |---------------------------------------------------|
          1     |           Status bits            |  Count         |
                |---------------------------------------------------|
          2     |           Status bits            |  Count         |
                |---------------------------------------------------|
                                    .     .     .
                |---------------------------------------------------|
          n     |           Status bits            |  Count         |
                +---------------------------------------------------+
```

| Bit | Symbol | Meaning |
|---|---|---|
| 0 | MONHBR | Only an exec process can wake job |
| 1 | IPCACE | IPCF event enable |
| 2-5 | HPQSPT | Console command setting of HPQ for job |
| 6-9 | HPQPNT | Current HPQ position of job |
| 10 | WAKEB | Wake bit - set if wake job by HIBER |
| 11 | IOACE | I/O activity enable |
| 12 | PTYWUE | PTY activity enable |
| 13 | TTIALE | TTY activity enable - line mode |
| 14 | TTIACE | TTY activity enable - character mode |
| 15-17 | HIBPRT | Hibernate protection code for job |
| 27-35 | | Count of number of realtime devices owned by job |

62 JBTSCD -- SUBQUEUE (CLASS) TABLE

Description: Contains each job's class type, indexed by job number; one entry for each job.

Defined in: COMMON

Used by: CLOCK1, SCHED1, UUOCON

GETTAB Table: .GTJTC (120)

```
            +--------------------------------------------------------+
Job     0   |X|              | Class # |          JS.TYP             |
            |--------------------------------------------------------|
        1   |X|              | Class # |          JS.TYP             |
            |--------------------------------------------------------|
        2   |X|              | Class # |          JS.TYP             |
            |--------------------------------------------------------|
                                 .    .    .
            |--------------------------------------------------------|
        n   |X|              | Class # |          JS.TYP             |
            +--------------------------------------------------------+
```

| Bits | Symbol | Description |
|---|---|---|
| 0 | JS.PQ2 | Job is in PQ2 |
| 13-17 | JS.CLS | Job's scheduler class |
| | JBYCLS | |
| 27-35 | JS.TYP | Job's scheduler type |

63 JBTSFD -- SUB-FILE DIRECTORY TABLE

Description: One entry per job holding search list and SFD information.

Defined in: COMMOD

Used by: COMMON, FILFND, FILUUO

```
         +-------------------------------------------------------+
Job   0  | LIB: PPB addr     | X | S |   NMB addr        | U | S |
         |-------------------------------------------------------|
      1  | LIB: PPB addr     | X | S |   NMB addr        | U | S |
         |-------------------------------------------------------|
      2  | LIB: PPB addr     | X | S |   NMB addr        | U | S |
         |-------------------------------------------------------|
                                 .   .   .
         |-------------------------------------------------------|
      n  | LIB: PPB addr     | X | S |   NMB addr        | U | S |
         +-------------------------------------------------------+
```

| Bits | Label | Meaning |
|---|---|---|
| 0-15 | | Address of library PPN PPB |
| 16 | JBPXSY | Search NEW before SYS |
| 17 | JBPSYS | Search SYS after DSK |
| 18-33 | | High-order 16 bits of address of default SFD NMB or 0. The NMB must be on a four-word boundary; therefore, bits 34-35 of the address must be zero. |
| 34 | JBPUFB | Bits 18-33 point to a PPB. |
| 35 | JBPSCN | Scanning is on. |

## 64 JBTSGN -- SEGMENT TABLE

Description: Tells which high segment, if any, each job is using, or which job that high segment was last or is being swapped in for. One entry for each segment number. Indexed by job number or segment number, JBTSGN is also referenced by the symbol JBTSWF.

Defined in: COMMON

Used by: CLOCK1, COMCON, CORE1, CPNSER, ERRCON, IPCSER, KLSER, LOKCON, METCON, SCHED1, SEGCON, UUOCON, VMSER

GETTAB Table: .GTSGN (14)

```
                +-------------------------------------------------------+
Job        0    |SP|SH|UW|ME|CO|LO|NC|SE|NO|RE|GT|    0     | High seg # |
                |-------------------------------------------------------|
           1    |SP|SH|UW|ME|CO|LO|NC|SE|NO|RE|GT|    0     | High seg # |
                |-------------------------------------------------------|
           2    |SP|SH|UW|ME|CO|LO|NC|SE|NO|RE|GT|    0     | High seg # |
                |-------------------------------------------------------|
           3    |SP|SH|UW|ME|CO|LO|NC|SE|NO|RE|GT|    0     | High seg # |
                |-------------------------------------------------------|
                                      .   .   .
                |-------------------------------------------------------|
High       n    |SP|SH|UW|ME|CO|LO|NC|SE|NO|RE|GT|    0     | High seg # |
Segment         |-------------------------------------------------------|
         n+1    |            0             |      Low Segment            |
                |-------------------------------------------------------|
         n+2    |            0             |      Low Segment            |
                |-------------------------------------------------------|
                                     .   .   .
```

The bit definitions that may be set for low segment entries are listed below:

| Bits | Symbol | Description |
|---|---|---|
| 0 | SPYSEG | High segment is physical core (see SPY UUO). |
| 1 | SHRSEG | High segment is sharable. The SHRSEG bit is also kept in the JBTSTS entry for that high segment. |
| 2 | UWPOFF | User mode write protect is off. |
| 3 | MEDDLE | User has meddled with sharable program. |
| 4 | CORCNT | High segment's in-core count has been incremented |
| 5 | LOKSEG | The high segment this job is sharing is locked in core. |
| 6 | NCSH | High segment is not cached. |
| 7 | SEGMB | UPT needs updating because the high segment |

| | | |
|---|---|---|
| | | locked by another job. |
| 8 | NOCSH | High segment is not cached because it is not writeable. |
| 9 | REDOMP | High segment is part of UPT needs to be rewritten |
| 10 | GTSSEG | High segment was obtained with a GETSEG UUO. |
| 13-17 | | Zero, so can read "@JBTSGN" to compare right half. |
| 18-35 | | If SPYSEG set, highest physical address user may see. If no high segment, zero. Otherwise, high segment number associated with this job. |

For high segment entries, the left half is 0 and bits 18-35 contain the low segment for which the high segment was/is being swapped in.

## 65 JBTSG2 -- HIGH SEGMENT SECTION NUMBERS

Description: One word for each high segment, containing the section numbers where the high segment is stored.

Defined in: COMMON

Used by: K?SER, SEGCON, VMSER

```
                  +--------------------------------------------------------+
High        1     | Section # |                                            |
Segment           |--------------------------------------------------------|
Number      2     | Section # |                                            |
                  |--------------------------------------------------------|
            3     | Section # |                                            |
                  |--------------------------------------------------------|
                                   .     .     .
                  |--------------------------------------------------------|
            n     | Section # |                                            |
                  +--------------------------------------------------------+
```

Bits 0-5 contain the section number where the high segment is stored.

66 JBTSHR -- HIGH SEGMENT SHARER COUNT

Description: Contains one word for each high segment. Indexed by high segment number, JBTSHR contains the total number of jobs (sharers) using the high segment.

Defined in: COMMON

Used by: K?SER, SEGCON, VMSER

```
                        +-------------------------------------------------------+
High     JOBMAX         |                  Total Sharer Count                   |
Segment                 |-------------------------------------------------------|
Number   JOBMAX+1       |                  Total Sharer Count                   |
                        |-------------------------------------------------------|
         JOBMAX+2       |                  Total Sharer Count                   |
                        |-------------------------------------------------------|
                                              .    .    .
                        |-------------------------------------------------------|
         JOBMAX+n       |                  Total Sharer Count                   |
                        +-------------------------------------------------------+
```

## 67 JBTSPL -- SPOOL CONTROL TABLE

Description: Contains input file name counter, devices being spooled, and disk priority for the job.

Defined in: COMMON

Used by: CLOCK1, COMCON, COMMOD, FILUUO, IPCSER, UUOCON

GETTAB Table: .GTSPL (36)

```
             +-------------------------------------------------------+
Job      1   | Input file name counter | DSK priority | Spool devices|
Number       |-------------------------------------------------------|
         2   | Input file name counter | DSK priority | Spool devices|
             |-------------------------------------------------------|
         3   | Input file name counter | DSK priority | Spool devices|
             |-------------------------------------------------------|
                                     .    .    .
             |-------------------------------------------------------|
         n   | Input file name counter | DSK priority | Spool devices|
             +-------------------------------------------------------+
```

| Bits | Symbol | Meaning |
|---|---|---|
| 0-17 | JB.SIN | Spooled input name counter. |
| 24-26 | JS.DPR<br>JBYPRI | Current disk priority |
| 27-35 | JB.SPL | Spooling bits: |
| 27 | JB.DFR | Deferred spooling |
| 31 | JB.CDR | Card reader spooling |
| 32 | JB.CDP | Card punch spooling |
| 33 | JB.PTP | Papertape punch spooling |
| 34 | JB.PLT | Plotter spooling |
| 35 | JB.LPT | Line printer spooling |

68 JBTSPS -- SECOND PROCESSOR STATUS

Description: Assembled in multiprocessor systems to indicate second processor status. One entry for each job number.

Defined in: COMMON

Used by: COMDEV, CPNSER, UUOCON

GETTAB Table: .GTSPS (54)

```
                  +--------------------------------------------------------+
Job         0     |    Bits set by monitor    | GETTAB bits for 6 CPUs     |
Number            |--------------------------------------------------------|
            1     |    Bits set by monitor    | GETTAB bits for 6 CPUs     |
                  |--------------------------------------------------------|
            2     |    Bits set by monitor    | GETTAB bits for 6 CPUs     |
                  |--------------------------------------------------------|
                                        .    .    .
                  |--------------------------------------------------------|
            n     |    Bits set by monitor    | GETTAB bits for 6 CPUs     |
                  +--------------------------------------------------------+
```

| Bit | Label | Meaning |
|---|---|---|
| 0 | | Not runnable on CPU5 |
| 1 | | Not runnable on CPU4 |
| 2 | | Not runnable on CPU3 |
| 3 | | Not runnable on CPU2 |
| 4 | SP.NR1 | Not runnable on CPU1 |
| 5 | SP.NR0 | Not runnable on CPU0 |
| 6 | | Current job on CPU5 |
| 7 | | Current job on CPU4 |
| 8 | | Current job on CPU3 |
| 9 | | Current job on CPU2 |
| 10 | | Current job on CPU1 |
| 11 | | Current job on CPU0 |
| 12 | SP.ROP | Job forced to policy CPU when policy CPU dies |
| 14 | SP.CC1 | Set if <CTRL/C> was typed or if <CTRL/D> breakpoints are enabled, and job is running on another CPU to force context switch to policy CPU. |
| 24 | | SET CPU command bit for CPU5 |
| 25 | | SET CPU command bit for CPU4 |
| 26 | | SET CPU command bit for CPU3 |
| 27 | | SET CPU command bit for CPU2 |
| 28 | SP.SC1 | SET CPU command bit for CPU1 (OK as far as user is concerned to run on this CPU even if stopped or not scheduling) |
| 29 | SP.SC0 | SET CPU command bit for CPU0 |
| 30 | | Can run on CPU5 |
| 31 | | Can run on CPU4 |

| | | |
|---|---|---|
| 32 | | Can run on CPU3 |
| 33 | | Can run on CPU2 |
| 34 | SP.CR1 | Can run on CPU1 |
| 35 | SP.CR0 | Can run on CPU0 |

## 69 JBTSTS -- STATUS TABLE

Description: Contains status information about each job and high segment. One entry for each job number and each high segment number. Indexed by job number or high segment number.

See also: AVALTB, JBTST2, JBTST3, QBITS

Defined in: COMMON. Bits defined in S.UNV.

Used by: CLOCK1, COMCOM, CORE1, CP1SER, DISSER, ERRCON, FILFND, FILIO, FILUUO, IPCSER, KILOCK, KISER, KLSER, KSSER, METCON, MTXSER, NULSEG, PSISER, PTYSER, REMDSX, RTTRP, SCHED1, SCNSER, SEGCON, SYSINI, UUOCON, VBCSER, VMSER

GETTAB Table: .GTSTS (0)

```
                +-------------------------------------------------------+
  Job     0     |                    Job status bits                    |
                |-------------------------------------------------------|
          1     |                    Job status bits                    |
                |-------------------------------------------------------|
          2     |                    Job status bits                    |
                |-------------------------------------------------------|
                                       .    .    .
                |-------------------------------------------------------|
  Job     n     |                    Job status bits                    |
                |-------------------------------------------------------|
High     n+1    |                High segment status bits               |
Segment         |-------------------------------------------------------|
         n+2    |                High segment status bits               |
                |-------------------------------------------------------|
                                       .    .    .
```

Notes:

Left Half of Job Number Entries:

| Bits | Label | Meaning |
|---|---|---|
| 0 | RUN | Set if user wants job to run. |
| 1 | CMWB | Job in Command Wait |
| 2 | JXPN | Job must be swapped out because it is expanding, and there is not enough room in core. |
| 3 | JNA | This job number is assigned. |
| 4 | JERR | Monitor detected error has occurred. |
| 5 | NSWP | Job is not to be swapped. |
| 6 | SHF | Monitor is waiting to shuffle or swap out this job. |
| 7 | SWP | 1 if job swapped out or in transit. |
| 8 | NSHF | Job is not to be shuffled. |

| | | |
|---|---|---|
| 9 | CNTRLC | <CTRL/C> typed while in monitor mode and not in TTY input wait - delay stopping job. |
| 10-14 | BSTS | Specifies transfer table to requeue job for current conditions. See AVLQTB, QBITS for the names of the tables. These are system dependent values as not all queues will be assembled for all systems. The value here may be used to count 12-bit bytes into STSTBL to get the SIXBIT queue name. See below for a list of job queues and wait state codes. |
| 15 | JLOG | Job logged in. |
| 16 | JRQ | Job has changed state and must be queued at clock level before rescheduling can take place. |
| 17 | JACCT | Privileged system CUSP which cannot be interrupted. |
| 18 | CLKR | Job has clock request in. |
| 19 | LOK | Job is being locked in core. |
| 20 | JDC | Job has typed "DCORE." |
| 21 | UTRP | Trap to USER on UUO exit (reenter DDT). |
| 22 | JDCON | Job in <CTRL/C> state waiting to continue from error. |
| 23 | JS.DEP | DAEMON error pause. |
| 24 | | Reserved for use by DIGITAL. |
| 25 | JS.XO | Execute-only core image. |
| 26 | JS.RUU | RUN UUO or command in progress. |
| 27 | JS.MPE | Memory parity error for job. |
| 28 | JS.BPT | <CTRL/D> DDT breakpoint is enabled. |
| 29 | JS.DPM | DAEMON problem message needed. |
| 30 | JS.ASA | Use shadow ACs for UUO args instead of user ACs. |
| 31 | JS.XOR | Run UUO or command in progress on execute-only file. |
| 32 | JS.RQR | Reset quantum run time. |
| 33 | JS.SFL | Stop job if disk is full. |
| 34 | JS.NXM | Some page in user's address space is contained in non-existent memory. |
| 35 | JS.NTO | Non-blocking terminal output. |

Left half of High Seg Entries:

| Bit | Symbol | Meaning |
|---|---|---|
| 0 | SNA | This high seg number assigned. |
| 1 | SHRSEG | Sharable segment (also kept in JBTSGN). |
| 2 | JXPN | High segment is expanding and must be swapped out. |
| 3 | SS.SYS | Hiseg came from SYS. |
| 4 | SERR | High segment swap read error. |
| 5 | NSWP | This high seg is not to be swapped. |
| 7 | SWP | 1 if segment swapped out or in transit. |
| 8 | NSHF | High segment cannot be shuffled. |

| | | |
|---|---|---|
| 9-17 | | Segment access privilege bits - same as disk file. |
| 18 | JS.SFE | Hiseg came from an EXE file. |
| 19-35 | JBYICC | In-core count of jobs using this high segment. |

Job Queues and Wait State Codes:

Job queues keep track of a job's priority to run and to be in core and are maintained in table JBTCQ. Wait State Codes keep track of the runnability of a job and are defined in parallel so that a queue number and a wait state code that have the same meaning will also have the same numeric value. Values will differ at different sites due to conditional assembly.

The Usage key is described following the list of queues.

| Symbol | Value | Usage | Feature | Meaning |
|---|---|---|---|---|
| RNQ | 00 | N | | Ready to run |
| WSQ | 01 | U | | I/O wait satisfied |
| TSQ | 02 | U | | TTY I/O wait satisfied |
| DSQ | 03 | U | | Disk I/O wait satisfied |
| PSQ | 04 | U | | Paging I/O wait satisfied |
| AUQ | 05 | R | | Alter UFD wait |
| DAQ | 06 | R | | Disk space allocation wait |
| CBQ | 07 | R | | Disk core block scan wait |
| DTQ | 10 | R | | DECtape controller wait |
| IPQ | 11 | R | | IPCF interlock wait |
| CXQ | 12 | R | | Context save wait |
| DCQ | 13 | R | | DECtape/magtape control wait |
| CAQ | 14 | R | FTLOCK | Semi-permanent core wait (LOCK) |
| MMQ | 15 | R | FTMP | Memory management wait |
| EVQ | 16 | R | FTKI\|FTKL | Exec virtual memory wait |
| EQQ | 17 | R | FTEQDQ | Enqueue-dequeue wait |
| MCQ | 20 | R | FTMP | Monitor I/O disk cache wait |
| IOWQ | 21 | C | | I/O wait |
| TIOWQ | 22 | CQ | | TTY I/O wait |
| DIOWQ | 23 | C | | Disk I/O wait |
| PIOWQ | 24 | C | | Paging I/O wait |
| PQIOQ | 25 | C | | Paging queue wait |
| SLPQ | 26 | CQ | | Sleeping (>= 1 second) |
| EWQ | 27 | CQ | | Event wait (see JBTST2) |
| NAPQ | 30 | C | | Napping (sleep < 1 second) |
| NULQ | 31 | CQ | | Unassigned jobs |
| JDCQ | 32 | CQ | FTDAEM | Job waiting for DAEMON |
| STOPQ | 33 | CQ | | Job stopped (C state) |
| CMQ | 34 | Q | | Command wait for swapper |
| PQ1 | 35 | Q | | Jobs starting up or coming out of TTY I/O wait |

| | | | | |
|---|---|---|---|---|
| PQ2 | 36 | Q | | Non interactive jobs |
| HPQ1 | 37 | Q | FTHPQ | High priority (realtime) jobs |
| HPQ2 | 40 | Q | FTHPQ | Higher priority (realtime) jobs |
| HPQ3 | 41 | Q | FTHPQ | Highest priority (realtime) jobs |

The number of HPQs is determined in the MONGEN dialogue.

Key to Usage:

N No queue header for this WSC. (JBTCQ+0 contains 0). Defined in QUEUES macro.

U Wait State Codes for jobs that have become unblocked but need to be processed by QREQ in SCHED1 before being considered to run. Defined in QUEUES macro.

R Wait State Codes for jobs that blocked at UUO level requiring a sharable resource that was unavailable (see REQTAB). Jobs will be unblocked by the scheduler when the resource becomes available (see AVALTB). Defined in RWAITS macro.

C Wait State Codes for jobs that are blocked waiting for a monitor event, such as I/O complete, clock ticks, command input, or DAEMON activity, before they will be runnable. Defined in CODES macro.

Q These queues in JBTCQ actually hold jobs. Other symbols have queue headers reserved for them, but jobs are never placed in them. RNQ has no queue header at all.

Notes:

1. RNQ, IOWQ, DIOWQ, PIOWQ, WSQ, TSQ, DSQ, and PSQ never actually hold jobs. The queues are defined only to define the corresponding Wait State Codes.

2. The values of PQ1, PQ2, CMQ, and STOPQ are never used as wait state codes. Jobs in any of the PQs have wait state codes of 000. When jobs are put into CMQ or STOPQ they retain their previous codes, so that they can be returned to their previous queues.

## 70 JBTST2 -- JOB STATUS TABLE 2

Description: Contains status information about each job. (Extension to the JBTSTS table.) Indexed by job number.

Defined in: COMMON. Bits defined in S.MAC.

Used by: CLOCK1, COMCON, FILFND KASER, KISER, KLSER, KSSER, SCHED1, SEGCON, UUOCON

GETTAB Table: .GTST2 (117)

```
                 +--------------------------------------------------------+
Job          0   |     Status bits     | E| M| A| Code| EW| PC| Q#| 0| R| S|
Number           |--------------------------------------------------------|
             1   |     Status bits     | E| M| A| Code| EW| PC| Q#| 0| R| S|
                 |--------------------------------------------------------|
             2   |     Status bits     | E| M| A| Code| EW| PC| Q#| 0| R| S|
                 |--------------------------------------------------------|
                                         .     .     .
                 |--------------------------------------------------------|
             n   |     Status bits     | E| M| A| Code| EW| PC| Q#| 0| R| S|
                 +--------------------------------------------------------+
```

| Bit | Label | Meaning |
|---|---|---|
| 0 | JS.IPQ | On if job is in a processor queue. |
| 1-2 | JS.DEB | Deferred echo bits |
| 3 | JS.SAC | Context auto-save requested by COMCON |
| 4 | JS.OLS | Job owns locked structure |
| 5 | JS.SIP | On if swapping I/O is in progress for this job |
| 6 | JS.FPS | On if long KA-10 floating point instructions should be simulated (FORCE was JXPN to avoid forgetting FORCE was cleared) |
| 7 | JS.NNQ | On if not to assign new quantum value on swap-in. Set when job GETSEGs a swapped segment. |
| 8 | JS.BBJ | On if job is from background batch. |
| 9 | JS.CSQ | On if job is changing subqueue due to SCHED. UUO. |
| 10 | JS.IGS | In GETSEG (doing a GETSEG UUO). |
| 11 | JS.HNG | Job stayed in FORCE too long, causing temporary or permanent system hang. Cleared when swapped in or out. |
| 12 | JS.BPR | Bypass program to run checking if set. The current RUN command is allowed despite .PDPGM (used for KJOB). |
| 13 | JS.FXO | File daemon made this core image execute-only. |
| 14 | JS.CFX | Call file daemon on program exit. |
| 15 | JS.HIB | This job is hibernating |

| | | |
|---|---|---|
| 16 | JS.NCS | On if the cache bits are not to be turned on in this job's map. |
| 17 | JS.EXE | A new save (EXE file) is underway, or on a GET, the LOOKUP on the EXE file failed. (Prevents EXE file from being read in again for the low segment.) |
| 18 | JS.MIG | Job has migrated to another swapping unit when a disk controller is being taken off-line (FTDHIA). |
| 19 | JS.ABP | An address break page fault occurred because of a reference to this user's virtual address space during UUO processing. |
| 20-24 | JBYEWT | ESLEEP reason: |

| Code | Symbol | Reason |
|---|---|---|
| 1 | EV.TKW | Tape controller |
| 2 | EV.REW | Rewind |
| 3 | EV.LBL | Label processing |
| 4 | EV.NET | Network device |
| 5 | EV.NTC | Network terminal connect |
| 6 | EV.STC | Network station control |
| 7 | EV.DTE | DTE I/O |
| 10 | EV.KDP | KDP I/O |
| 11 | EV.IPC | IPCF system process receive |
| 12 | EV.FEI | Front end device input |
| 13 | EV.FEO | Front end device output |
| 14 | EV.D60 | DN60 device |
| 15 | EV.DCN | DECnet connect I/O |
| 16 | EV.DMR | DMR I/O |

| | | |
|---|---|---|
| 25 | EWAKEB | EWAKE called (wakeup waiting) |
| 26 | JS.RPC | If = 1, run program in .PDPGM on <CTRL/C> |
| 27-32 | PJBST2 | Queue number. Also called PJBST1 (for indexing by T1). |
| 33 | JS.OOO | User ran out of order. Stop when gives up last resource. |
| 34 | JS.TFO | Job forced out by timer. |
| 35 | JS.SCN | Job was scanned to run by at least one CPU during last tick. |

71 JBTST3 -- JOB STATUS TABLE 3

Description: Contains the address of the CPU data block for the CPU on which the job ran last. This table applies to multi-CPU KL systems only.

Defined in: COMMON

```
                 +-------------------------------------------------------+
Job         0    |                          |       CDB Address          |
Number           |-------------------------------------------------------|
            1    |                          |       CDB Address          |
                 |-------------------------------------------------------|
            2    |                          |       CDB Address          |
                 |-------------------------------------------------------|
                                       .    .    .
                 |-------------------------------------------------------|
            n    |                          |       CDB Address          |
                 +-------------------------------------------------------+
```

Bits 18 through 35 contain the address of the CPU Data Block for the CPU that last ran the job.

## 72 JBTSWP -- JOB SEGMENT SWAP AREA

| | |
|---|---|
| Description: | Contains information used when swapping segments. There is one entry for each job number and high segment number. It is indexed by the job number or high segment number. The fragment table is documented separately in Section 29. |
| Defined by: | COMMON |
| Used by: | COMCON, KISER, KLSER, KSSER, SCHED1, SEGCON, SWPSER, UUOCON, VMSER |
| GETTAB Table: | .GTSWP (7) |

Each word in the JBTSWP table is formatted depending on the setting of Bit 15. If Bit 15 is set, the core address is in bits 18-35. If Bit 15 is clear, bits 15-17 contains the index to the unit number and bits 22-35 contain the first logical K on the unit.

```
                +-------------------------------------------------------+
Job      0      |F|          | 1|    |    Core address                  |
Number          |-------------------------------------------------------|
         1      |F|          | 1|    |    Core address                  |
                |-------------------------------------------------------|
         2      |F|          | 1|    |    Core address                  |
                |-------------------------------------------------------|
                                               . . .
                |-------------------------------------------------------|
         n      |F|          | 1|    |    Core address                  |
                |-------------------------------------------------------|
High     n+1    |F|          | 1|    |    Core address                  |
Segment         |-------------------------------------------------------|
Number   n+2    |F|          | 1|    |    Core address                  |
                |-------------------------------------------------------|
                                               . . .
```

| Bit | Name | Description |
|---|---|---|
| 0 | FRGSEG | 1 if low or high segment is fragmented on the swapping device. |
| 15-35 | | Disk address, if Bit 0 is off. Core address in Bits 18-35 of fragment table if Bit 0 is set. |
| 15-17 | JBYSUN | Index to unit number in SWPTAB. |
| 22-35 | JBYLKN | First logical K on the unit. |

73 JBTUPM -- UPT ADDRESS TABLE

Description: Contains the address for the user page map page for each job and high segment that is in core. This table is also called JBTHSA.

Defined in: COMMON

Used by: KILOCK, KISER, KLSER, SCHED1, SEGCON, SWPSER, SYSINI UUOCON, VMSER

GETTAB Table: .GTUPM (100)

```
Job            +-----------------------------------------------------+
Number     0   |    HSO     |     0      |     HSS    |   UPT addr    |
               |-----------------------------------------------------|
           1   |    HSO     |     0      |     HSS    |   UPT addr    |
               |-----------------------------------------------------|
           2   |    HSO     |     0      |     HSS    |   UPT addr    |
               |-----------------------------------------------------|
                                   .   .   .
               |-----------------------------------------------------|
           n   |    HSO     |     0      |     HSS    |   UPT addr    |
               |-----------------------------------------------------|
High     n+1   |    HSO     |            |            |  HSA          |
Segment        |-----------------------------------------------------|
Number   n+2   |    HSO     |            |            |  HSA          |
               |-----------------------------------------------------|
                                   .   .   .
```

Description for Low Segment Entry:

| Bit | Pointer | Meaning |
|---|---|---|
| 0-8 | JBYHSO | Virtual page number of hi-seg. origin |
| 18-22 | JBYHSS | Number of pages less one in high segment |
| 23-35 | | Physical page number of UPT |

Description for High Segment Entry:

| | | |
|---|---|---|
| 0-8 | JBYHSO | Virtual page number of high-seg origin |
| 23-35 | JBYHSA | Physical page number of first page of high segment |

74 JBTVIR -- VIRTUAL SIZE TABLE

Description: Virtual size of program

Defined in: COMMON

Related Tables: JBTIMI

```
                 0---5  6-----------14 15------------------------35
Job              +---------------------------------------------------+
Number       0   |      | HIVSIZ        |          LOVSIZ            |
                 |---------------------------------------------------|
             1   |      | HIVSIZ        |          LOVSIZ            |
                 |---------------------------------------------------|
             2   |      | HIVSIZ        |          LOVSIZ            |
                 |---------------------------------------------------|
                                        .   .   .
                 |---------------------------------------------------|
             n   |      | HIVSIZ        |          LOVSIZ            |
                 +---------------------------------------------------+
```

| Bits | Symbol | Contents |
|---|---|---|
| 0-5 | | Reserved |
| 6-14 | HIVSIZ | High segment size (non-sharable). If this field is 0, the high segment is sharable. |
| 15-35 | LOVSIZ | Low segment size. |

## 75 JBTWCH -- WATCH TABLE

Description: This table is assembled if FTWATCH is assigned a non-zero value in S.MAC. It determines the specific watch parameters to be displayed. There is one entry per job number including the null job. The null job entry is not referenced.

Defined in: COMMON

Used by: COMCON, MTXSER, UUOCON

GETTAB Table: .GTWCH (35)

```
Job          +-------------------------------------------------------+
Number    0  |      Conditions          |          Time of day       |
             |-------------------------------------------------------|
          1  |      Conditions          |          Time of day       |
             |-------------------------------------------------------|
          2  |      Conditions          |          Time of day       |
             |-------------------------------------------------------|
                                      . . .
             |-------------------------------------------------------|
          n  |      Conditions          |          Time of day       |
             +-------------------------------------------------------+
```

| Bit | Name | Description |
|---|---|---|
| 1 | JB.WDY | Watch time of day started to wait. |
| 2 | JB.WRN | Watch runtime when return to command level. |
| 3 | JB.WWT | Watch waiting time when return to command level. |
| 4 | JB.WDR | Watch number of 128 word disk block read. |
| 5 | JB.WDW | Watch number of 128 word disk blocks written. |
| 6 | JB.WVR | Watch versions. |
| 7 | JB.WMT | Watch number MTA performace statistics. |
| 8 | JB.WFL | Watch file activity. |
| 9 | JB.WLM | Long error messages. |
| 10 | JB.WNM | Normal error messages. |
| 11 | JB.WPM | Prefix error messages. |
| 13-35 | | Time of day (in jiffies) user started to wait. |

## 76 JDA -- DEVICE ASSIGNMENT TABLE

Description: Associates a device or file with each active channel in a user job, and tells which UUOs have been done on that channel.

The JDA is part of the job's UPT. Each JDA contains 16 entries corresponding to the 16 software channels of a user job.

Indexed by channel number.

Defined in: COMMON

Used by: COMCON, CORE1, DTASRN, FILIO, FILUUO, MSGSER, PTYSER, SEGCON, UUOCON, VMSER

```
           +--------------------------------------------------------+
Channel  0 |          UUO bits*        |   Device Data Block address  |
           |--------------------------------------------------------|
         1 |          UUO bits*        |   Device Data Block address  |
           |--------------------------------------------------------|
         2 |          UUO bits*        |   Device Data Block address  |
           |--------------------------------------------------------|
                                    .    .    .
           |--------------------------------------------------------|
        17 |          UUO bits*        |   Device Data Block address  |
           +--------------------------------------------------------+
```

* The UUO bits are set for the following reasons:

| Bit | Label | Meaning |
|---|---|---|
| 0 | INITB | INIT or OPEN has been done |
| 1 | IBUFB | Input ring header specified (by INIT) |
| 2 | OBUFB | Output ring header specified (by INIT) |
| 3 | LOOKB | A LOOKUP has been done |
| 4 | ENTRB | An ENTER has been done |
| 5 | INPB | An INPUT has been done |
| 6 | OUTPB | An OUTPUT has been done |
| 7 | ICLOSB | An input CLOSE has been done |
| 8 | OCLOSB | An output CLOSE has been done |
| 9 | INBFB | An input buffer ring has been set up |
| 10 | OUTBFB | An output buffer ring has been set up |
| 11 | SYSDEV | This is the system tape device or SYSPPN on DSK |
| 12 | RENMB | RENAME UUO in progress |
| 13 | RESETB | RESET UUO in progress |
| 18-35 | | Address of Device Data Block for I/O on this software channel |

Notes:

1. If both LOOKB and ENTRB are on, the file is being accessed in update mode.

2. Extended channel information is kept in a 64-word table in funny space pointed to by the .UPCTA word in the UPT. If extended channels are not in use, the word is zero. Each entry in the table is formatted the same as entries in the JDA table.

## 77 JOBDAT -- DATA AREA

Description: Storage area for items of interest to both the monitor and the user.

There is one Job Data Area for each job that has a non-zero core allocation. It occupies the first 140 locations of the job's core area, and is swapped out along with the job.

Defined in: COMMON

Used by: COMMON

| Offset | Symbol | Map | |
|---|---|---|---|
| 0 - 17 | .JBBAC | User ACs during UUO (16 words) | |
| 20 - 37 | .JBBAC | Hardware ACs while job inactive (16 words) | |
| 40 | .JBUUO | User UUO stored here | |
| 41 | .JB41 | User UUO branch instruction | |
| 42 | .JBERR | Unused | Error cnt for RPG |
| 43 | .JBENB | Unused | User APR trap flags |
| 44 | .JBREL | 0 | Length of low seg |
| 45 | .JBPD1 | Push down list (21 words) | |
| 72 | .JBHCU | Highest I/O channel in use | |
| 73 | .JBPC | Job PC when job inactive | |
| 74 | .JBDDT | Unused | Start addr of DDT |
| 76 | .JBBPT | Address of unsolicited breakpoint entry into DDT | |

| | | | |
|---|---|---|---|
| 114 | .JBSDD | JOBDDT here on SAVE/Protected from I/O | |
| 115 | .JBHRL | First free loc in high seg | Length of high seg |
| 116 | .JBSYM | Symbol table pointer | |
| 117 | .JBUSY | Undefined symbol table pointer | |
| 120 | .JBSA | First free loc in low seg- when loaded | Program start address |
| 121 | .JBFF | Current first free location in low seg. | |
| 122 | .JBS41 | .JB41 here on SAVE | |
| 123 | .JBEXM | Address of last D or E command | |
| 124 | .JBREN | Address for REENTER command | |
| 125 | .JBAPR | Branch loc on user enabled APR error | |
| 126 | .JBCNI | APR conditions on APR trap | |
| 127 | .JBTPC | PC stored here on APR trap | |
| 130 | .JBOPC | Old PC stored here on START, DDT, REENTER, & CSTART commands | |
| 131 | (3) .JBCHN | Used for FORTRAN job chaining (root link) | |
| 132 | .JBFDV | DDB addr for FINISH command | |
| 133 | .JBCOR | Highest loc in low seg actually loaded | Low seg core assignment |
| 134 | .JBINT | | Data block adr for error intercept |
| 135 | .JBOPS | Reserved for runtime operating system | |
| 136 | .JBCST | Reserved for customer | |
| 137 | .JBVER | Job version number | |
| 140 | | First loc in user's program area | |

Notes:

1. The actual tables are included at the beginning of each user's area.

2. Many of these words contain different values while a SAVE or GET is in progress, and therefore have several different labels.

3. .JBCHN is also symbolized by .JBOVL.

4. System-sensitive locations for all machines are stored in the UPT instead of the user's core image. If .JBxxx is stored in location yyy, VJBDAT will define the value of the symbol as -1000+yyy (octal).

## 78 KON -- KONTROLLER DATA BLOCK

Description: There is one KON per disk controller. It contains information specific to that controller, for example, dispatch addresses into the controller dependent routines. Controller data blocks are generated dynamically by AUTCON when the system is started.

Defined in: COMMOD

Used by: FILIO, ONCE, ONCMOD, RPXKON, RNXKON, SYSINI, VMSER

```
               +-----------------------------------------------------+
KONBSY,, (2)   | K | Reserved |     | B |    Max. unit #              |
KONMUN         |   |          |     |   |                             |
               |-----------------------------------------------------|
KONTBP         |           AOBJN pointer to table of UDBs            |
               |-----------------------------------------------------|
KONCAM (3)     |                                        |5|4|3|2|1|0|
               |-----------------------------------------------------|
KONIIO         |  Offset into the KDB of the first I/O instruction   |
               |-----------------------------------------------------|
KONCOM         |  Negative of CCWMAX      |          KONLST          |
               |-----------------------------------------------------|
KONRED (4)     | Table of controller-dependent dispatch instructions |
               |-----------------------------------------------------|
KONCUA         | UDB address of unit doing transfer (or last one)    |
               |-----------------------------------------------------|
KONIOC         | Address of controller-channel pair in low core      |
               |-----------------------------------------------------|
KONCHN         | Address of channel data block for this controller   |
               |-----------------------------------------------------|
KONPTR         | Indirect pointer to index KONTAB    (P3)            |
               |-----------------------------------------------------|
KONDMP         | Holds DEVDMP during dump-mode I/O                   |
               |-----------------------------------------------------|
KONERR         | Dispatch to controller-dependent error-recovery prog. |
               |-----------------------------------------------------|
KONECC         | Used for ECC mask and position                      |
               |-----------------------------------------------------|
KONRRG         | Used to hold drive registers                        |
               |-----------------------------------------------------|
KONECR         | Control register on error                           |
               |-----------------------------------------------------|
KONEDB         | Data buffer register on error                       |
               |-----------------------------------------------------|
KONREG         | Length of KONEBK                                    |
               |-----------------------------------------------------|
KONEBK         | Place to save drive registers on error              |
               +-----------------------------------------------------+
```

Notes:

1. The CONSO skip chain entry for the controller immediately precedes the Kontroller Data Block for the controller. These skip chain entries differ for each type of controller.

2. KONBSY contains the following bits:

   Bit 0 (KOPBSY) is set if the controller is busy.

   Bits 1-6 are reserved.

   Bit 17 (KOPBND) is set for CI disk controllers when disk units on the controller are bound.

   Bits 18-35 (KONMUN) contain the maximum unit number on the controller.

3. KONTBP contains the AOBJN pointer to the the table of UDB addresses stored in the KDB. The offset of this table within the KDB varies dpending on the type of controller. The offset to the table of UDB addresses is defined by the symbol xxKNTB, where xx is the controller type (RP, RN, RA,...). This table supersedes the KONTAB table.

4. KONCAM contains a bit mask that tells which CPU(s) can access the controller. Bit 35 is set for CPU0, Bit 34 is set for CPU1, Bit 33 for CPU2, and so forth.

5. KONRED, the table of controller-dependant dispatch addresses, is described below.

KONRED (Controller-Dependent Dispatch Addresses)

```
                  +-------------------------------------------------------+
KONRED (1)        | PI   |  Address of read program                      |
                  |-------------------------------------------------------|
KONRDS            |         Address of read, stop-on-error program        |
                  |-------------------------------------------------------|
KONRDF            |         Read header and data                          |
                  |-------------------------------------------------------|
KONRDC            |         Read in 10/11 compatibility mode              |
                  |-------------------------------------------------------|
KONWRT            |         Entry point to write program                  |
                  |-------------------------------------------------------|
KONWTS (2)        |         Entry point to write, stop on error           |
                  |-------------------------------------------------------|
KONWTF            |         Write format                                  |
                  |-------------------------------------------------------|
KONWTC            |         Write in 10/11 compatibility mode             |
                  |-------------------------------------------------------|
KONUNL            |         Unload drive                                  |
                  |-------------------------------------------------------|
KONPOS (3)        | F |     Position heads-entry zero for fixed-head disk |
                  |-------------------------------------------------------|
KONLTM (4)        |         Compute rotational latency time               |
                  |-------------------------------------------------------|
KONUPA (5)        | U | S |  Test if controller is on-line                |
                  |-------------------------------------------------------|
KONCPY            |         Determine type and capacity of unit           |
                  |-------------------------------------------------------|
KONRCL            |         Recalibrate                                   |
                  |-------------------------------------------------------|
KONSTP            |         Stop on hung unit                             |
                  +-------------------------------------------------------+
```

Notes:

1. Bits 0-2 (KOBPI) of KONRED contain the PI level for the controller.

2. For CI disks, KONWTS is equivalent to KONCNA, where Bit 0 (KOPCNA), if set, indicates credits are not available.

3. Bit 0 (KOPPWX) of KONPOS, if set, indicates fixed-head devices.

4. For CI disks, KONLTM is equivalent to KONMX, where Bit 0 (KOPMX) indicates the disk can do multiple transfers at one time.

5. KONUPA contains two flags. Bit 0 (KONDWN) is set when the controller is down. Bit 1 (KONMPS) is set when the controller can seek while doing a transfer.

## 79 LDB -- LINE DATA BLOCK

Description: Contains data pertaining to one terminal line. One LDB per line; including scanner lines, PTYs, and CTY. LINTAB serves as directory.

Defined in: SCNSER

Used by: CLOCK1, COMCON, COMMON, D76INT, MSGSER, NETSER, PSISER, PTYSER, SYSINI, UUOCON, XTCSER

| | |
|---|---|
| LDBDDB | Address of line's attached DDB |
| LDBCOM* | Forced command bits (Section 76.1) |
| LDBATR*<br>LDICLR | Line attribute bits Section 76.2)<br>Start clearing here on restart |
| LDBOST* | Terminal output bits (Section 76.3) |
| LDBIST* | Input state word (Section 76.4) |
| LDBBKU | Copy of LDBTIP at last break XMTECH |
| LDBBKI | Copy of LDBTIP at last break RECINT |
| LDBTIP | T2 to put characters in input buffer |
| LDBTIT | T2 to take characters from input buffer |
| LDBTIC | Count of echoed characters in input buffer |
| LDBBKC | Count of break characters in input buffer |
| LDBTOP | T3 to put characters in output buffer |
| LDBTOT | T2 to take characters from output buffer |
| LDBTOC | Count of characters from output buffer |
| LDBECT | T2 to take characters from input for echoing |
| LDBECC | Count of characters to echo |
| LDBIEC | Count of invisible characters in echo stream |
| LDBIIC | Count of invisible characters in input stream |
| LDBEOP | T3 to put characters in echo buffer |

| | | |
|---|---|---|
| LDBEOT | T3 to take characters from echo buffer | |
| LDBEOC | Count of characters in echo buffer | |
| LDBOOP (1) | Byte pointer to enqueue out-of-band characters | |
| LDBOOT (1) | Byte pointer to dequeue out-of-band characters | |
| LDBOOC (1) | Count of enqueued out-of-band characters | |
| LDBCLP | Command line pointer (for COMCON) | |
| LDBXNP | XON class character pointer for output | |
| LDBFLP | Filler character pointer for output | |
| LDBNNP | "Not now" character pointer for output | |
| LDBPBK | Up to 4 break characters for Packed Image Mode (PIM) | |
| LDBHPS | Horizontal Position Counter | |
| LDBBCT | Total command line count for this line | Total break characters for this line |
| LDBICT | Total input character count | |
| LDBOCT<br>LDICLE (2) | Total output character count<br>(Clear through here on restart) | |
| LDBDCH* | Hardware status bits (Section 76.5) | |
| LDBBYT* | First word of software status bits (Section 76.6) | |
| LDBBY2* | Second word of software status bits (Section 76.7) | |
| LDBBY3* | Third word of software status bits (Section 76.8) | |
| LDBLSW* | Page length bits (Section 76.9) | |
| LDBPAG* | Terminal page bits (Section 76.10) | |
| LDBISR* | Interrupt service routine addr (Section 76.11) | |
| LDBISB* | Line speed word (Section 76.12) | |
| LDBQUE | Global address of the next line in the queue | |
| LDBQUH | Queue header | Reserved |
| LDBTTW* | Type of line bits (Section 76.13) | |

|-----------------------------------------------------|

| | |
|---|---|
| LDBREM+0* (4) | Remote bits (Section 76.14) |
| LDBREM+1 (4) | Last characteristics message sent |
| LDBREM+2* (4) | Remote line number (Section 76.15) |
| LDBREM+3* (4) | Remote node number (Section 76.16) |
| LDBREM+4* (4) | MCR/VTM word (Section 76.17) |
| LDBLAT (2)<br>LDBNRT (3)<br>LDBTTD* (5) | Global address of LAT service data block, or<br>Global address of NRB (NRTSER data block), or<br>Line information for support of RSX-20F terminals<br>(Section 76.18) |
| LDBMIC* (6) | MIC bits (Section 76.19) |
| LDBLOT (7) | T2 to take characters for logging |
| LDBLOC (7) | Count of characters to log |
| LDBBKB* | Break mask field width (Section 76.20) |
| LDBCSB | / Reserved space for special character coding / |
| LDBCC1 | Clear OOB flags for low-order control characters |
| LDBCC2 | Clear OOB flags for high-order control characters |
| LDBCHM* | Characters mapped by RECMAP (Section 76.21) |
| LDBLEN | Size of a data block for a line |

Notes:

1. If FTIP (Programmable Software Interrupt System) support is included.

2. If LAT (LAT-11 terminal) support is included.

3. If FTDECN (include DECnet support).

4. If FTNET (include NCL network software).

5. If FTKL10 (include KL10 support).

6. If FTMIC (include Macro Command Processor).

7. If FTMLOG (include MIC Log File Support).

79.1 LDBCOM -- Forced Command Word

```
+----------------------------------------------------------------------+
|L|L|L|L|L|         |    L    |                                         |
|D|D|D|D|D|         |    D    |                                         |
|B|B|B|B|B|         |    P    |                                         |
|C|C|C|D|F|         |    C    |                                         |
|M|M|M|E|D|         |    M    |                                         |
|R|F|K|T|X|         |    X    |                                         |
|----------------------------------------------------------------------|
|0|1|2|3|4|5----8|9-----12|13------------------------------------35|
+----------------------------------------------------------------------+
```

| Word Symbol | Bits | Mask Symbol | Description |
|---|---|---|---|
| 0 | 400000 | LDBCMR | Command request bit |
| 1 | 200000 | LDBCMF | Command forced |
| 2 | 100000 | LDBCMK | Forcing KJOB command |
| 3 | 40000 | LDBDET | Job detached from this line during command processing |
| 4 | 20000 | LDBFDX | Processing a FILDAE exit message |
| 5-8 | | | Reserved |
| 9-12 | | LDPCMX | Pointer index for forced command |
| 13-35 | | | Reserved |

## 79.2 LDBATR -- Line Attributes Word

```
+-----------------------------------------------------------------------+
|L|L|L|                                                                  |
|A|A|A|                                                                  |
|L|L|L|                                                                  |
|8|C|D|                                                                  |
|B|O|I|                                                                  |
|T|S|S|                                                                  |
|-----------------------------------------------------------------------|
|0|1|2|3-------------------------------------------------------------35|
+-----------------------------------------------------------------------+
```

| Word Symbol | Bits | Mask Symbol | Description |
|---|---|---|---|
| 0 | 400000 | LAL8BT | Line is associated with an eight-bit terminal |
| 1 | 200000 | LALCOS | Line can do overstriking |
| 2 | 100000 | LALDIS | Line is associated with a display terminal |
| 3-35 | | | Reserved |

## 79.3 LDBOST -- Output Bits Word

```
+-----------------------------------------------------------------------+
|L|L|L|L|L|L|L|L|L|L| L| L|                                           |
|0|0|0|0|0|0|0|0|0|0| 0| 0|                                           |
|L|L|L|L|L|L|L|L|L|L| L| L|                                           |
|X|N|E|F|S|S|S|N|R|R| P| M|                                           |
|F|B|S|S|A|T|S|N|E|E| I| I|                                           |
|P|S|P|P|P|P|O|P|O|E| M| C|                                           |
|-----------------------------------------------------------------------|
|0|1|2|3|4|5|6|7|8|9|10|11|12-------------------------------------35|
+-----------------------------------------------------------------------+
```

| Word Symbol | Bits | Mask Symbol | Description |
|---|---|---|---|
| 0 | 400000 | LOLXFP | XOFF fill pointer to be sent |
| 1 | 200000 | LOLNBS | Need bell sent |
| 2 | 100000 | LOLESP | Echo stream pointer to be serviced |
| 3 | 40000 | LOLFSP | Fill string pointer to be serviced (SEND ALL) |
| 4 | 20000 | LOLSAP | SEND ALL pending |
| 5 | 10000 | LOLSTP | Output stopped by XOFF |
| 6 | 4000 | LOLSSO | SCNSER stopped output (for page stop) |
| 7 | 2000 | LOLNNP | Not-now pointer to be serviced |
| 8 | 1000 | LOLREO | Re-eat output after free CRLF |
| 9 | 400 | LOLREE | Re-eat echo after free CRLF |
| 10 | 200 | LOLPIM | Terminal was opened in packed image mode |
| 11 | 100 | LOLMIC | Line is controlled by MIC |
| 12-35 | | | Reserved |

79.4 LDBIST -- Input Status Word

```
+-----------------------------------------------------------------+
|      Character being deferred    |        Reason for deferring  |
+-----------------------------------------------------------------+
```

The reason codes are:

| Code | Symbol | Meaning |
|---|---|---|
| 1 | LISDCI | Deferred clear interrupt |
| 2 | LISQOT | Quoting a character |
| 3 | LISSWI | Evaluating a possible switch sequence |

## 79.5 LDBDCH -- Hardware Status Word

```
+---------------------------------------------------------------------+
|L|L|L|L|  |L|L|L|L|L|  L|  L|          | L|  L|   L   |  | L|          |
|D|D|D|D|  |D|D|D|D|D|  D|  D|          | D|  D|   D   |  | D|          |
|L|L|L|L|  |L|L|L|L|L|  L|  L|  LDPVR1  | R|  R|   P   |  | R|  LDPLNO  |
|I|P|C|F|  |8|D|N|F|I|  C|  B|          | P|  C|   V   |  | S|          |
|D|P|R|C|  |B|L|E|C|M|  O|  K|          | T|  T|   R   |  | H|          |
|L|S|P|R|  |I|R|C|S|I|  M|  A|          | Y|  Y|   2   |  | C|          |
|---------------------------------------------------------------------|
|0|1|2|3|4|5|6|7|8|9|10|11|12------17|18|19|20---24|25|26|27--------35|
+---------------------------------------------------------------------+
```

| Word Symbol | Bits | Mask Symbol | Description |
|---|---|---|---|
| 0 | 400000 | LDLIDL | Line is idle. If clear, we are expecting a transmit interrupt |
| 1 | 200000 | LDLPPS | Prompt position has been set for this line |
| 2 | 100000 | LDLCRP | Control-R pending (XMTECH synch plug) |
| 3 | 40000 | LDLFCR | Forcing control-R (XMTECH synch plug) |
| 4 | 20000 | | Free bit |
| 5 | 10000 | LDL8BI | Eight-bit input mode by program (reserved) |
| 6 | 4000 | LDLDLR | Suppress dollar sign |
| 7 | 2000 | LDLNEC | No echo, due to program |
| 8 | 1000 | LDLFCS | Line inited in full character set mode |
| 9 | 400 | LDLIMI | Image input |
| 10 | 200 | LDLCOM | Line is at command level |
| 11 | 100 | LDLBKA | Break on all characters (DDTIN, TTCALL) |
| 12-17 | | LDPVR1 | 4 of 6 bits pointed to by GETLP1 for GETLIN: (*) |
| 12 | 40 | LDLSLV | Slave; this terminal may be assigned |
| 13 | 20 | LDLLCT(1) | Lowercase translate to upper |
| 14 | 10 | LDLTAB(2) | Line accepts tabs, not spaces |
| 15 | 4 | LDLLCP | Local copy (no echo) |
| 16 | 2 | LDLFRM(3) | Line accepts FF and VT (else use LFs) |
| 17 | 1 | LDLNFC | No free carriage return at 72 columns |
| 18 | 400000 | LDRPTY | Pseudo-terminal |
| 19 | 200000 | LDRCTY | Console terminal |
| 20-24 | | LDPVR2 | Bits pointed to by LDPVR2: |
| 20 | 100000 | LDROSU(4) | Output suppress (<CTRL/O>) |
| 21 | 40000 | LDRDSD | Dataset line |
| 22 | 20000 | LDR274 | Line is a 2741 |

| | | | |
|---|---|---|---|
| 23 | 10000 | LDRHLF | Half duplex line (TWX or DC10C) |
| 24 | 4000 | LDRRMT | Remote non-dataset line |
| 25 | 2000 | LDRREM | Obsolete |
| 26 | 1000 | LDRSHC | Suppress hung check |
| 27-35 | | LDPLNO | Pointer to hardware line number |

* Bits for GETLIN UUO:

| Value | Label | Description |
|---|---|---|
| 100 | GTLRDY | Bit for GETLIN to indicate waiting break character |
| 20 | GTLT37 | Model 37 bit (copy of LDLLCT) |
| 10 | GTLT35 | Model 35 bit (copy of LDLTAB) |
| 4 | GTLLCP | Local copy (copy of LDLLCP) |
| 2 | GTLXON | XON is true |

(1) LDPLCT Pointer to lower case bit
(2) LDPTAB Pointer to hardware tabs bit
(3) LDPFRM Pointer to hardware form feed bit
(4) LDPOSU Pointer to output suppression bit

79.6 LDBBYT -- First Word For Software Status

The LDBBYT byte word is also referenced by the byte pointer stored in LDBOFL.

```
+-------------------------------------------------------------------------+
|L| L | P |     |L| L| L| L| L| L|  L  | L| L|  L  | L| L| L|          |
|1| D | O |     |1| 1| 1| 1| 1| 1|  D  | 1| 1|  D  | 1| 1| 1|          |
|L| P | H |     |L| L| L| L| L| L|  P  | R| R|  P  | R| R| R|          |
|O| F | P |     |U| Q| Q| Q| Q| D|  T  | M| C|  C  | D| D| D|          |
|F| L | O |     |N| C| N| T| O| E|  I  | I| H|  P  | E| E| E|          |
|L| C | S |     |R| C| C| C| T| M|  M  | F| P|  U  | L| C| M|          |
|-------------------------------------------------------------------------|
|0|1 2|3--5|6---8|9|10|11|12|13|14|15-19|20|21|22-24|25|26|27|28----35|
+-------------------------------------------------------------------------+
```

| Bits | Value | Mask Symbol | Description |
|---|---|---|---|
| 0 | | L1LOFL | Set to 1 if front end for this line is down |
| 1-2 | | LDPFLC | Count of number of fillers by class |
| 3-5 | | POHPOS | Old horizontal position. Needed for tab simulation |
| 6-8 | | | Reserved |
| 9 | 400 | L1LUNR | Unread in progress |
| 10 | 200 | L1LQCC | Quote next character, for CCTYI |
| 11 | 100 | L1LQNC | Quote next character, for TYICCY |
| 12 | 40 | L1LQTC | Quote next character, set/cleared by XMTECH |
| 13 | 20 | L1LQOT | TTY Quote Enabled flag |
| 14 | 10 | L1LDEM | Deferred Echo flag, set/cleared by XMTECH |
| 15-19 | | LDPTIM | Timeout on image input |
| 20 | | L1RMIF | MIC interlock flag |
| 21 | | L1RCHP | Change hardware parameters queue bit |
| 22-24 | | LDPCPU | CPU number |
| 25 | | L1RDEL | Echo may echo 1 line if deferred |
| 26 | | L1RDEC | Echo may echo 1 character if deferred |
| 27 | | L1RDEM | Deferred echo bit. Set by SET TERMINAL DEFER |
| 28-35 | | | Reserved |

79.7 LDBBY2 -- Second Word For Software Status

```
+-------------------------------------------------------------------------+
|L|L|L|L|L|L|L|L|L|                  | L| L|              |   L   |       |
|2|2|2|2|2|2|2|2|2|                  | 2| 2|              |   D   |       |
|L|L|L|L|L|L|L|L|L|      LDPDSC      | R| R|    LDPWID    |   P   |       |
|D|C|H|H|H|H|H|S|T|                  | X| E|              |   A   |       |
|E|C|D|D|D|D|D|N|A|                  | 0| C|              |   P   |       |
|L|S|1|2|3|4|5|D|P|                  | N| S|              |   C   |       |
|-------------------------------------------------------------------------|
|0|1|2|3|4|5|6|7|8|9--------------17|18|19|20---------27|28---31|32-35|
+-------------------------------------------------------------------------+
```

| Bits | Value | Mask Symbol | Description |
|---|---|---|---|
| 0 | 400000 | L2LDEL | Last character in was a delete |
| 1 | 200000 | L2LCCS | Last character in was a <CTRL/C> |
| 2 | 100000 | L2LHD1 | XMT done flag seen this character on HDX line |
| 3 | 40000 | 2LHD2 | RCV done flag seen this character on HDX line |
| 4 | 20000 | 2LHD3 | Ignoring RCV interrupts due to echo check error on HDX line |
| 5 | 10000 | L2LHD4 | Next RCV interrupt will be queued after echo check |
| 6 | 4000 | L2LHD5 | Receive echo was in fact not same as transmitted character |
| 7 | 2000 | L2LSND | Send allowed while busy |
| 8 | 1000 | L2LTAP | <CTRL/Q> from keyboard turns on L2RXON |
| 9-17 | | LDPDSC | Dataset Control Table Index back pointer |
| 18 | 400000 | L2RXON | XON is true (paper tape input) |
| 19 | 200000 | L2RECS | Eat command sync |
| 20-27 | | LDPWID | Width of terminal carriage |
| 28-31 | | LDPAPC (1) | Asynchronous port characteristic (type) |
| 32-35 | | | Reserved |

(1) Fields in LDPAPC:

| | | |
|---|---|---|
| 0 | APCUNK | Unknown type or not yet set up |
| 1 | APCHWD | Hardwired terminal |
| 2 | APCDSD | Dataset line |
| 3 | APCTSN | Reserved |
| 4 | APCGAN | Reserved |
| 5 | APCADL | Autodialler |
| 6 | APCMCM | Reserved |
| 7 | APCNRT | DECnet NRTSER line |
| 10 | APCLAT | LAT-11 terminal |
| 11 | APCCTM | DECnet CTERM line |

79.8 LDBBY3 -- Third Word For Software Status

```
+--------------------------------------------------------------------+
|L|L|L|L|L|L|L|L|L|L| L|     R     |        L         | L|    L    |
|3|3|3|3|3|3|3|3|3|3| 3|     e     |        D         | 3|    D    |
|L|L|L|L|L|L|L|L|L|L| L|     s     |        P         | R|    P    |
|D|F|O|E|I|C|F|O|E|I| C|     r     |        M         | T|    T    |
|M|P|P|P|P|P|H|H|H|H| H|     v'    |        X         | M|    M    |
|C|D|D|D|D|D|D|D|D|D| D|     d     |        T         | O|    R    |
+--------------------------------------------------------------------+
|0|1|2|3|4|5|6|7|8|9|10|11--------18|19---------------26|27|28-----35|
+--------------------------------------------------------------------+
```

| <u>Bits</u> | <u>Symbol</u> | <u>Meaning</u> |
|---|---|---|
| 0 | L3LDMC | Deferred echo mode changed (sign bit) |
| 1 | L3LFPD | Fill partly done (3 characters) |
| 2 | L3LOPD | Output partly done (3 characters) |
| 3 | L3LEPD | Echo partly done (3-part character to be completed) |
| 4 | L3LIPD | Input partly done (3-part character) |
| 5 | L3LCPD | Command partly done (3-part character) |
| 6 | L3LFHD | Fill half-done (2-part character) |
| 7 | L3LOHD | Output half-done (2-part character) |
| 8 | L3LEHD | Echo half-done (2-part character) |
| 9 | L3LIHD | Input half-done (2-part character) |
| 10 | L3LCHD | Command half-done (2-part character) |
| 11-18 | | Reserved |
| 19-26 | LDPMXT | Maximum idle time for auto-disconnect |
| 27 | L3RTMO | Timeout flag: overflow for LDPTMR (auto-disconnect timer expired) |
| 27-35 | LDPTMR | Count up timer for auto-disconnect |

79.9 LDBLSW -- Page Length Word

```
+----------------------------------------------------------------------+
|                  |                |                 |                |
|      LDPLNB      |     LDPSTB     |     LDPLNC      |     LDPSTC     |
|                  |                |                 |                |
|----------------------------------------------------------------------|
|0---------------8|9------------17|18-------------26|27-----------35|
+----------------------------------------------------------------------+
```

| Bits | Value | Mask Symbol | Description |
|---|---|---|---|
| 0-8 | | LDPLNB | Page or "forms" length (base value) |
| 9-17 | | LDPSTB | Stop (after n lines) value (base) |
| 18-26 | | LDPLNC | Page or "forms" counter (counted up to 0) |
| 18 | 400000 | LPRLCO | Length counter overflowed flag |
| 27-35 | | LDPSTC | Stop counter (counted up to 0) |
| 27 | 400 | LPRSCO | Stop counter overflowed flag |

## 79.10 LDBPAG -- Page Bits

```
+-----------------------------------------------------------------------+
|L|L|L|L|L|L|L|  |L|L| L| L| L| L| L|      |             |             |
|P|P|P|P|P|P|P|  |P|P| P| P| P| P| P|      |             |             |
|L|L|L|L|L|L|L|  |L|L| L| L| L| L| P|      |   LDPPFF    |   LDPACR    |
|I|P|X|A|B|S|S|  |P|S| S| F| S| F| F|      |             |             |
|R|A|O|L|L|L|A|  |O|T| S| F| B| F| F|      |             |             |
|M|G|F|T|K|F|P|  |K|P| T| S| L| F| H|      |             |             |
|-----------------------------------------------------------------------|
|0|1|2|3|4|5|6|7|8|9|10|11|12|13|14|15--17|18---------26|27---------35|
+-----------------------------------------------------------------------+
```

| Bits | Value | Mask Symbol | | Description |
|---|---|---|---|---|
| 0 | 400000 | LPLIRM | | Terminal not heard from this second |
| 1 | 200000 | LPLPAG | | Set terminal page command was executed |
| 2 | 100000 | LPLXOF | | Sent XOFF, always send XON later |
| 3 | 40000 | LPLALT | (1) | Altmode conversion bit |
| 4 | 20000 | LPLBLK | | Suppress blank lines |
| 5 | 10000 | LPLSLF | | Suppress line feeds |
| 6-7 | | | | Reserved |
| 8 | 1000 | LPLPOK | | Forcing XMIT start using TOPOKE |
| 9 | 400 | LPLSTP | | Automatically stop every (LDPSTB) lines |
| 10 | 200 | LPLSST | | Don't clear page counter for free on XON |
| 11 | 100 | LPLFFS | | Stop on form feeds (reserved) |
| 12 | 40 | LPLSBL | | Ring bell on auto-stop |
| 13 | 20 | LPLFFF | | Simulate form feed with linefeeds |
| 14 | 10 | LPLFFH | | Simulate form feed with clear-screen |
| 15-17 | | | | Reserved |
| 18-26 | | LDPPFF | | Number LFs remaining on VT and FF simulation |
| 27-35 | | LDPACR | | Auto-CRLF column counter |

(1) LDPALT    Pointer to altmode conversion bit

79.11 LDBISR -- Interrupt Service Routine

```
+-----------------------------------------------------------------+
|@| Index   | Section Number      | Address of Service Routine    |
|-----------------------------------------------------------------|
|0|1|2-------5|6-----------------17|18---------------------------35|
+-----------------------------------------------------------------+
```

| <u>Bits</u> | <u>Description</u> |
|---|---|
| 0 | Clear for global indirect word |
| 1 | Indirecion (clear) |
| 2-5 | Index register (T1) |
| 6-17 | Section number of routine |
| 18-35 | Address of interrupt service routine |

## 79.12 LDBISB -- Line Speed Word

```
+---------------------------------------------------------------------+
|L|  L   |  L   |L| L| L| L|                                          |
|I|  D   |  D   |D| D| D| D|                                          |
|L|  P   |  P   |P| P| P| P|                                          |
|C|  T   |  R   |A| D| R| T|                                          |
|F|  S   |  S   |P| B| T| D|                                          |
|E|  P   |  P   |L| K| C| Y|                                          |
|---------------------------------------------------------------------|
|0|1---4|5---8|9|10|11|12|13---------------------------------------35|
+---------------------------------------------------------------------+
```

| Bits | Symbol | Meaning |
|---|---|---|
| 0 | LILCFE | 1 if front end is clever, 0 if dump |
| 1-4 | LDPTSP | Transmit speed |
| 5-8 | LDPRSP | Receive speed |
| 9 | LDPAPL | APL mode |
| 10 | LDPDBK | Line has debreak |
| 11 | LDPRTC | <CTRL/R>/<CTRL/T> compatibility |
| 12 | LDPTDY | User said SET TERMINAL TIDY |
| 13-35 | | Reserved |

The byte pointers defined for this word are:

| Value | Byte Pointer |
|---|---|
| 400000 | LILCFE |
| 360000 | LILTSP |
| 017000 | LILRSP |
| 000400 | LILAPL |
| 000200 | LILDBK |
| 000100 | LILRTC |
| 000040 | LILTDY |

79.13 LDBTTW -- Line Type Word

```
+-----------------------------------------------------------------------+
|ANF|NRT|LAT|USE|Reserved |FSP|VDC|IDC|Res.|    LCH    |  PRP  |  TTT    |
|-----------------------------------------------------------------------|
| 0 | 1 | 2 | 3 |4-------8| 9 | 10| 11|  12| 13---20|21--28|29-----35 |
+-----------------------------------------------------------------------+
```

| Bits | Symbol | Description |
|---|---|---|
| 0 | LTLANF | ANF-10 network virtual terminal |
| 1 | LTLNRT | DECnet NRT or CTERM virtual terminal |
| 2 | LTLLAT | LAT server terminal line |
| 3 | LTLUSE | Allocatable LDB in use (always on for local LDBs, such as the CTY) |
| 4-8 | | Reserved |
| 9 | LDLFSP | Line is a full-SCNSER PTY |
| 10 | LDLVDC | Visible deletion of character has happened |
| 11 | LDLIDC | Invisible deletion of character has happened |
| 12 | | Reserved |
| 13-20 | LDPLCH | Last character read by COMCON |
| 21-28 | LDPPRP | Position of prompt |
| 29-35 | LDPTTT | Terminal type, as specified by SET TTY TYPE command |

## 79.14 LDBREM+0 -- Remote Bits

```
+-----------------------------------------------------------------------+
| | | | | | | | | | | |  |  |  |  |  |  |  |  |  |                      |
| | | | | | | | | | | |  |  |  |  |  |  |  |  |  |        LDPSTS        |
| | | | | | | | | | | |  |  |  |  |  |  |  |  |  |                      |
|-----------------------------------------------------------------------|
|0|1|2|3|4|5|6|7|8|9|10|11|12|13|14|15|16|17|18|19|20----------------35|
+-----------------------------------------------------------------------+
```

The bits in this word are defined different for different applications (NETVTM and NETMCR).

| Bits | Value | Mask Symbol | | Description |
|---|---|---|---|---|
| 0 | 400000 | LRLVTM | (1) | If set, then this is a "local terminal" that has been "SET HOSTed" to another host |
| 1 | 200000 | LRLCON | (1) | If set, then terminal is "connected" (NCL connect sequence is complete) |
| 2 | 100000 | LRLSTS | (1) | If set, then a "STATUS" message is required. Same bit, but different messages for VTM and MCR |
| 3 | 40000 | LRLSCH | (2) | If set, then a "CHARACTERISTICS" message is required (works like LRLSTS) |
| 4 | 20000 | LRLDST | (2) | A "delayed" status message is required (used to optimize traffic. Has priority over LRLSTS) |
| 5 | 10000 | LRLQED | (2) | If set, then VTM line has been "queued" by VTMENQ |
| 6 | 4000 | LRLDIP | (2) | Set to initiate a disconnect |
| 7 | 2000 | LRLVTF | (2) | VTM terminal needs to be freed by FRELDB |
| 8 | 1000 | LRLVTZ | (2) | VTM terminal has been zapped |
| 3 | 40000 | LRLTTO | (3) | LDPCHR has the next character to output |
| 4 | 20000 | LRLTTW | (3) | Line is waiting for a data request |
| 5 | 10000 | LRLSCG | (3) | <CTRL/O> action requested (send character gobbler) |
| 6 | 4000 | LRLEPW | (3) | Echo pipeline marker waiting to go |
| 7 | 2000 | LRLIMO | (3) | Indicates that remote is in image mode output |
| 8 | 1000 | LRLADR | (3) | Use of the auto-dialer has been requested |
| 9 | 400 | LRLXOF | (3) | An XOFF (<CTRL/S>) message has |

| | | | |
|---|---|---|---|
| | | | been requested |
| 10 | 200 | LRLCHR (3) | This terminal has received at least 1 Characteristics message |
| 11 | 100 | LRLHUR (3) | Hang-up phone requested |
| 12 | 40 | LRLDSR (3) | The -10's copy of what it thinks LRLDTR should be |
| 13 | 20 | LRLGRT (3) | Greet the terminal (with INITIA) |
| 14 | 10 | LRLATO (3) | Indicates this line possesses the auto-baud capability (set or cleared by the attribute field of the CONNECT message) |
| 15 | 4 | LRLADL (3) | Indicates this line possesses an auto-dialer (also set by CONNECT message) |
| 16 | 2 | LRLTMO (3) | Auto-disconnect requested hang-up phone |
| 17 | | | Reserved |
| 18 | 400000 | LRRSHC (3) | The line at the other end has "SET HOST capability." (it can respond to DISCONNECT messages) |
| 19 | 200000 | LRRXFF (3) | Send XON/XOFF in status message |

Notes:

1. Bits used by both NETVTM (local SET HOST) and NETMCR (remote terminals)

2. Bits used only by NETVTM (local SET HOST)

3. Bits used only by NETMCR ("normal" remote terminals, like DN87)

79.15 LDBREM+2 -- Remote Line Number

```
+------------------------------------------------------------------+
|                        |                          |              |
|        LDPSLA          |          LDPDLA          |     LDPRLN   |
|                        |                          |              |
|------------------------------------------------------------------|
|0--------------------12|13--------------------25|26----------35|
+------------------------------------------------------------------+
```

| Bits | Byte Pointer | Meaning |
|---|---|---|
| 0-12 | LDPSLA | Our source link address |
| 13-25 | LDPDLA | Our destination link address |
| 26-35 | LDPRLN | Line number at remote station |

79.16 LDBREM+3 -- Remote Node Number

```
+-----------------------------------------------------------------+
|  L  |                |                  |                        |
|  D  |                |                  |                        |
|  P  |     LDPDRQ     |     LDPEPM       |         LDPRNN         |
|  E  |                |                  |         LDPRNF         |
|  L  |                |                  |                        |
|  E  |                |                  |                        |
|-----------------------------------------------------------------|
|0---3|4------------11|12-------------19|20--------------------35|
+-----------------------------------------------------------------+
```

| Bits | Byte Pointer | Description |
|---|---|---|
| 0-3 | LDPELE | 2741 element number |
| 4-11 | LDPDRQ | Number of data requests from remote |
| 12-19 | LDPEPM | Serial number of last EPM from remote |
| 20-35 | LDPRNN | Number of node owning this TTY |
| 20-35 | LDPRNF | Same as LDPRNN except indexed by "F" |

### 79.17 LDBREM+4 -- MCR/VTM Word

The bytes in LDBREM+4 are different for MCR and VTM applications. The following byte definitions are used in MCR applications:

```
+-----------------------------------------------------------------------+
|                 |                 |                 |                 |
|     LDPCHR      |     LDPADT      |     LDPJOB      |                 |
|                 |                 |                 |                 |
|-----------------------------------------------------------------------|
|0-------------8|9-------------17|18------------26|27-------------35|
+-----------------------------------------------------------------------+
```

| Bits | Byte Pointer | Description |
|---|---|---|
| 0-8 | LDPCHR | If LRLTTO=1, contains the next output character |
| 9-17 | LDPADT | Auto-disconnect timer |
| 18-26 | LDPJOB | Pointer to job (only for connects) |
| 27-35 | | Unused |

The following byte definitions are used in LDBREM+4 by VTM applications:

```
+-------------------------------------------------------------------------+
|                                    |                                    |
|              LDPDST                |              LDBVTQ                |
|                                    |                                    |
|-------------------------------------------------------------------------|
|0-------------------------------17|18--------------------------------35|
+-------------------------------------------------------------------------+
```

| Bits | Byte Pointer | Description |
|---|---|---|
| 0-17 | LDPDST | Delayed-status-message for VTM |
| 18-35 | LDBVTQ | VTM-queue-link |

## 79.18 LDBTTD -- RSX-20F Word

```
+--------------------------------------------------------------------+
|                                                                    |
| Line information for support of RSX-20F terminals                  |
|                                                                    |
|--------------------------------------------------------------------|
|0-----------------------------------------------------------------35|
+--------------------------------------------------------------------+
```

| Value | Byte Pointer | Description |
|---|---|---|
| 740000 | | Remembered transmit speed |
| 36000 | | Remembered receive speed |
| 1000 | LTLXOF | Sent XOFF to -20F |
| 400 | LTLRBS | Remote bit sent for -20F datasets |
| 200 | LTLCTO | Need to send flush output to -20F |

79.19 LDBMIC -- MIC Bits

```
+------------------------------------------------------------------+
| | | | | | | | | | |  |  |  |  |                    |           |           |
| | | | | | | | | | |  |  |  |  |                    |           |           |
|------------------------------------------------------------------|
|0|1|2|3|4|5|6|7|8|9|10|11|12|13|14|15-------21|22-------28|29-------35|
+------------------------------------------------------------------+
```

| Bits | Value | Mask Symbol | Description |
|---|---|---|---|
| 0 | 400000 | LDLCHK | Set if any of Bits 1-14 are set |
| 1 | 200000 | LDLMCC | Set if <CTRL/C> has been typed |
| 2 | 100000 | LDLOPC | Set if operator character seen in column 1 |
| 3 | 40000 | LDLERC | Set if error character seen in column 1 |
| 4 | 20000 | LDLMCP | Set if <CTRL/P> has been typed |
| 5 | 10000 | LDLMCB | Set if <CTRL/B> has been typed |
| 6 | 4000 | LDLSIL | Silence this line |
| 7 | 2000 | LDLMMM | Line in monitor mode |
| 8 | 1000 | LDLMTI | Line in user mode and TI wait or in monitor mode and can accept commands |
| 9 | 400 | LDLCL1 | Line is in Column on output |
| 10 | 200 | LDLMCA | Set if <CTRL/A> has been typed (abort) |
| 11 | 100 | LDLRSP | Set if error output is available |
| 12 | 40 | LDLRSY | Set if error output is being accepted |
| 13 | 20 | LDLLOG | MIC is logging |
| 14 | | | Reserved |
| 15-21 | | | ASCII character to be treated as operator character |
| 22-28 | | | ASCII character to be treated as error character |
| 29-35 | | | MIC master job numbers -- allows more than one MIC to run |

79.20 LDBBKB -- Break Mask Field Width

```
+-----------------------------------------------------------------------+
|   | Reserved   | Field Width | Reserved                               |
|-----------------------------------------------------------------------|
|0  |1---------2|3----------11|12------------------------------------35|
+-----------------------------------------------------------------------+
```

| Bits | Byte Pointer | Description |
|---|---|---|
| 0 | LDLBKM | Break masks are enabled. |
| 1-2 | | Reserved |
| 3-11 | LDPFWD | Field width |
| 12-35 | | Reserved |

79.21 LDBCHM -- RECMAP Characters

```
+-----------------------------------------------------------------------+
| |     LDPUNP    |      LDPESC       |      LDPSW1      |      LDPSW2    |
|-----------------------------------------------------------------------|
| 0| 1----------9 |10------------17 | 18-----------25| 26-----------33|
+-----------------------------------------------------------------------+
```

| Bits | Byte Pointer | Description |
|---|---|---|
| 0 | LDBBKM | Break masks are enabled |
| 1-9 | LDPUNP | TTY unpause character |
| 10-17 | LDPESC | TTY escape character |
| 18-25 | LDPSW1 | Switch sequence number one |
| 26-35 | LDPSW2 | Switch sequence number two |

## 80 LINTAB -- LINE TABLE

Description: Contains the address of each Line Data Block. One entry per line (including scanner, CTY, and PTY lines.) Indexed by line number.

Defined: COMDEV

Used in: CLOCK1, COMCON, COMMON, CPNSER, D76INT, DLOINT DL1INT, DTESER, ERRCON, MSGSER, NETMCR, ONCE, PTYSER SCNSER, SYSINI, TTDINT, UUOCON

```
         +--------------------------------------------------------+
Line 0   |                 Fullword LDB address                   |
         |--------------------------------------------------------|
Line 1   |                                                        |
         |--------------------------------------------------------|
Line 2   |                                                        |
         |--------------------------------------------------------|
   -     |                                                        |
   -     |                                                        |
   -     |                                                        |
         |                                                        |
         |                                                        |
         |--------------------------------------------------------|
Line N   |                                                        |
         +--------------------------------------------------------+
          0                                                    35
```

## 81 LOGTAB -- LOGIN JBTTAB POINTER TABLE

Description: Table of pointers to those job tables in which the LOGIN UUO stores statistics.

Defined in: UUOCON (local symbol)

Used by: UUOCON

```
+--------------------------------------+
|    AC J     |        JBTPPN##        | ; PPN
|--------------------------------------|
|    AC J     |        JBTPRV##        | ; Privilege bits
|--------------------------------------|
|    AC W     |         PDNM1##        | ; 1st half user name
|--------------------------------------|
|    AC W     |         PDNM2##        | ; 2nd half user name
|--------------------------------------|
|    AC W     |         PDCNO##        | ; Charge no. this job
+--------------------------------------+
  0         17  18                  35
```

Notes:

1. LOGIN UUO used only by LOGIN and LOGOUT programs.

2. User program does:

```
LOGIN AC,
    or
CALLI AC,15
```

   where AC contains:

```
XWD no.-of-entries,location
```

   which is a list of job statistics to be stored in the monitor tables listed in LOGTAB.

82 LVDTBL -- LEVEL D DISK PARAMETER TABLE

Description: Contains parameters for the Level D disk routines. Each entry accessed by its own label.

Defined in: COMMOD

Used by: CLOCK1, COMCON, COMMON, FILFND, FILIO, FILUUO, ONCMOD, REFSTR, SCNSER, SYSINI, UUOCON

GETTAB Table: .GTLVD (16)

| | | |
|---|---|---|
| MFDPPN | PPN for MFDS [1,1] | |
| SYSPPN | PPN for device SYS [1,4] | |
| FSFPPN | PPN for FAILSAFE [1,2] | |
| HELPPN | PPN for HELP and SYSTAT [2,5] | |
| PNTPPN | PPN for printer spooling [3,3] | |
| SYSPPB | First PPB in system | PPB to start scan for obtainable NMB* |
| SYSSTR | First STR in system | Index in structure data block for pointer to next structure data block |
| SYSUNI | First UDB in system | Index in unit data block for pointer to next unit data block |
| SWPUNI | First swapping UDB | Index in unit data block for pointer to next swapping unit data block |
| CORNUM | Number of 4 word blocks | |
| STNPRT | Standard privilege | |
| UFDPRT | Standard UFD privilege | |
| MBFNUM | Number of monitor buffers | |
| QUESTR | SIXBIT name of structure for queuing programs | |
| CRUPPN | UFD for dumping crashes | |

| | |
|---|---|
| SFDLVL | Number of nested SFD levels allowed |
| SPLPRT | Protection for spooled output \| |
| SYSPRT | Protection for most system files \| |
| SYSPRY | Protection for system .SYS files \| |
| MUSTMX | Negative maximum extended RIB argument for USETI |
| MAXTRN | Maximum number of blocks to transfer in one operation |
| XSYPPN | PPN of experimental system (NEW) [1,5] |
| OLDPPN | PPN of old system (OLD) [1,3] |
| UMDPPN | User-mode diagnostics PPN [6,10] |
| NUMBF | Default number of disk buffers |
| MAXSWP | Maximum number of units in active swapping list |
| ALGPPN | ALGOL library PPN [5,4] |
| BLIPPN | BLISS library PPN [5,5] |
| FORPPN | FORTRAN library PPN [5,6] |
| MACPPN | MACRO library PPN [5,7] |
| UNVPPN | Universal library PPN [5,17] |
| PUBPPN | User-maintained SYS [1,6] |
| TEDPPN | Text editor library PPN [5,10] |
| RELPPN | REL file library PPN [5,11] |
| RNOPPN | RUNOFF library PPN [5,11] |
| SNOPPN | SNOBOL library PPN [5,13] |
| DOCPPN | DOC file library PPN [5,14] |
| FAIPPN | FAIL library PPN [5,15] |
| MUSPPN | Music library PPN [5,16] |
| DECPPN | Library for DEC-distributed software PPN (DEC) [10,7] |

| | |
|---|---|
| TABSWP | Pointer to active swap list |
| BASPPN | BASIC library PPN [5,1] |
| COBPPN | COBOL library PPN [5,2] |
| MXIPPN | PDP-11 library PPN [5,3] |
| NELPPN | NELIAC library PPN [5,20] |
| DMPPPN | Dump library PPN [5,21] |
| POPPPN | POP2 library PPN [5,22] |
| TSTPPN | Test library PPN [5,23] |
| ALLOVR | Non-zero: log soft errors if recover from overrun on 1 try |
| MASERR | DAEMON's pointers to massbuss error locations |
| BATCHN | DAEMON's pointers to BAT block and channel error info |
| DBSPPN | DBMS library PPN [5,24] |
| EXPCHN | Offset of the expected channel terminal word in CHN |
| MICPPN | MIC library PPN [5,25] |
| TPSPPN | Text processing system library PPN [5,26] |
| CTLPPN | CTL file library PPN [5,27] |
| GAMPPN | Game library PPN [5,30] |
| ACTPPN | System accounting library PPN [1,7] |
| APLPPN | APL library PPN [5,31] |
| RIBECT | RIB error threshold |
| RIBTOT | Total RIB errors |
| SYSDOR | Dormant access table pointer |
| SYSCOR | Free core pointer |
| INTFNC | Number of times front end (RSX20F) had disk |
| D60PPN | DAS60 log file area [5,32] |
| ERTLOC | Starting location of queue table for DAEMON error reports |

|----------------------------------------------------------|

| | |
|---|---|
| ERTPT1 | Starting pointer for DAEMON error extract |
| ERTPT2 | Starting pointer for DAEMON error insert |
| ERTLTH | Length of DAEMON error table |
| ERTCDA | Offset of UNICDA in UDB |
| ERTDES | Offset of UNIDES in UDB |
| SYSPTR | Pointer to in-core copies of retrieval ptrs |
| MAXSSL | Max. # in system search list \| MAX # in job search list |
| ERTSLB | Offset of UINSLB into UDBs |
| UTPPPN | UETP area |
| INIPPN | Initialization area [5,34] |
| ERPSIZ | Length of entry in Daemon error report table |
| SYSKON | Core addr of first KDB \| Offset in KDB of next KDB |
| NUMLBF | Default number of large disk buffers |
| | Offset into DEVUNI for SYSTAT |
| %LDCSZ | Size of disk cache, in blocks |
| %LDRDC | Monitor cache read calls |
| %LDRDH | Monitor cache read hits |
| %LDWRC | Monitor cache write calls |
| %LDWRH | Monitor cache write hits |
| %LDHSF | CSHFND calls |
| %LDHSC | CSHFND collisions in hash table |
| %LDHSL | Length of cache hash table |
| %LDHST | Addr of cache hash table |
| %LDCHD | Addr of cache list header |
| | Offset for spooled file name |
| | Offset for spooled parameter block pointer |

| | |
|---|---|
| | Offset for I/O block number |
| | Offset to retrieval/ALL blocks |
| | Offset to NMB for father SFD |
| UPSPPN | Area for mailers (UPS) [5,35]<br>PPN for library for mailers |
| | Address of pointer to first system error block |
| ROODRB | Number of times we ran out of DRBs |

Note:

If the right half of SYSPPB = 0, the core grabber starts over at the beginning of the PPB list.

83 MAGTAPE CONTROLLER DATA BLOCK

Description: Contains controller-dependent information. The Magtape KDB is a prototype data block used by AUTCON to configure the tape controllers at system startup and when they come on-line. Tags for magtape KDB's are MTxKDB, where x is the controller number (A, B, C,...). Tags for the CONSO instructions are MTxINT.

See also: Magtape Device Data Block, Magtape Unit Data Block

Defined in: COMDEV

Used by: AUTCON

| | | |
|---|---|---|
| TKBCSO | CONSO MTxS,0 | |
| | JRST .-1 | |
| | JSR MT'x'SAV | |
| | JSP W,TAPINT | |
| TKBNAM | SIXBIT Controller Name | |
| TKBCNT | Number of records done on this operation | |
| TKBKDB | Link to next magtape KDB, 0 if none | |
| TKBIUN | Initial Unit AOBJN Pointer | |
| TKBCUN | Current Unit AOBJN Pointer | |
| TKBDSP | Controller dispatch location (1) | |
| TKBSTS | Controller status bits (2) | Consecutive ops remaining |
| TKBICP | Pointer to ICPC/CHL info | |
| TKBTIM | Timer for spacing operations | |
| TKBCDB | Pointer to channel (0 if TM11A) | |
| TKBJOB | Job # of maintenance mode owner | |
| TKBCCL | Channel command list (5 words long) | |
| TKBERB | IORB for error recovery (4 words long) | |

|--------------------------------------------------|

```
        |-----------------------------------------------------|
TKBFCT  |  Fairness count for queued I/O                      |
        |-----------------------------------------------------|
TT2CO1  | CONO MTxS,(T1)      (TM02 only)                     |
        |-----------------------------------------------------|
TT2CI2  | CONI MTxS,T2        (TM02 only)                     |
        |-----------------------------------------------------|
TT2CO3  | CONO MTxS,(T3)      (TM02 only)                     |
        |-----------------------------------------------------|
TT2DI2  | DATAI MTxS,T2       (TM02 only)                     |
        |-----------------------------------------------------|
TT2DO2  | DATAO MTxS,T2       (TM02 only)                     |
        |-----------------------------------------------------|
TKBUDB  |        Pointers to units on this Controller         |
        |          (8 words long, found by TKBIUN)            |
        +-----------------------------------------------------+
```

Notes:

1. TKBDSP points to the controller-dependent data block in TAPSER. The data block contains offsets into the dispatch table. Each word in the dispatch table points to code to perform the following operations:

| Offset | Symbol | Operation |
|---|---|---|
| 0 | TPKINI | Initialization code |
| 1 | TPKRES | Reset active transfer |
| 2 | TPKSIO | Start I/O |
| 3 | TPKINT | Interrupt routine |
| 4 | TPKCMD | Set device command in list (DX10 only) |
| 5 | TPKIDL | Set device idle |
| 6 | TPKONL | Skip if controller online |
| 7 | TPKSCH | Cause schedule cycle |
| 10 | TPKINX | Initialization code after system startup |
| 11 | TPKLOD | Load microcode |
| 12 | TPKEDL | Enable/disable microcode loading |
| 13 | TPKCFG | Auto-configuration only needed for tapes with sub-units, such as TM02 and TM78. |

2. TKBSTS Bits:

(These bits are also defined for TUBSTS in the UDB.)

| Bit | Symbol | Meaning |
|---|---|---|
| 10 | TKSCHX | Not yet swept for CPU0. |
| 11 | TKSCHE | Job has swept cache for queued requests |
| 12 | TKSMNT | Controller is in maintenance mode |
| 13 | TKSSIL | Request silence about offline condition |
| 14 | TKSSCH | Requested scheduled interrupt |
| 15 | TKSSTD | Started |

| | | |
|---|---|---|
| 16 | TKSSEL | Selected |
| 17 | TKSOFL | Offline |

## 84 MAGTAPE UNIT DATA BLOCK

Description: Tape unit dependent information. The UDB is a prototype data block that contains information used by AUTCON to configure tape drives at system startup and when they come on-line. UDBs can be found by tracing them through the pointers in their KDB's (TKBUDB pointed to by TKBIUN) or by tags formed by concatenating "..U" with controller number and unit number, for example, ..U12 for the third unit on the second controller (MTB2).

See also: Magtape Device Data Block, Magtape Kontroller Data Block

Defined in: COMDEV

Used by: COMMON, TAPUUO

```
        +---------------------------------------------------+
TUBNAM  |                 SIXBIT Unit Name                  |
        |---------------------------------------------------|
TUBKDB  |                         | Pointer to controller   |
        |---------------------------------------------------|
TUBADR  |                         | Unit addr on controller |
        |---------------------------------------------------|
TUBAKA  | Current unit address    |   Current controller    |
        |---------------------------------------------------|
TUBDDB  |                 Pointer to DDBs                   |
        |---------------------------------------------------|
TUBCUR  |              Pointer to current DDB               |
        |---------------------------------------------------|
TUBSTS* |      Unit status        |   Configuration info    | TUBCNF
        |---------------------------------------------------|
TUBQUE  |              Queue pointer for IORBs              |
        |---------------------------------------------------|
TUBERR  |               Error recovery info                 |
        |---------------------------------------------------|
TUBRID  |                      Reelid                       |
        |---------------------------------------------------|
TUBFIL  |               # of files from BOT                 |
        |---------------------------------------------------|
TUBREC  |              # of records from EOF                |
        |---------------------------------------------------|
TUBCRD  |        # of characters read since unload          |
        |---------------------------------------------------|
TUBCWR  |       # of characters written since unload        |
        |---------------------------------------------------|
TUBSRE  |               # of soft read errors               |
        |---------------------------------------------------|
TUBHRE  |               # of hard read errors               |
        |---------------------------------------------------|
```

```
        |-----------------------------------------------------|
TUBSWE  |              # of soft write errors                 |
        |-----------------------------------------------------|
TUBHWE  |              # of hard write errors                 |
        |-----------------------------------------------------|
TUBTME  |          Total media errors since unload            |
        |-----------------------------------------------------|
TUBTDE  |          Total device errors since reload           |
        |-----------------------------------------------------|
TUBTUN  |                  Total unloads                      |
        |-----------------------------------------------------|
TUBTRY  |              Retries to resolve error               |
        |-----------------------------------------------------|
TUBCCR  |           Character count on last record            |
        |-----------------------------------------------------|
        |           File            |          Record         |
TUBPBE  |               position before error                 |
        |-----------------------------------------------------|
TUBFES  |               Final error state word                |
        |-----------------------------------------------------|
TUBCHR  |             Statistics for MTCHR. UUO               |
        |-----------------------------------------------------|
TUBDDA  |               Shadow area for DAEMON                |
        |                12 (octal) words                     | TUBDDE
        |-----------------------------------------------------|
TUBPGM  |   Program name on error                             |
        |-----------------------------------------------------|
TUBUID  |   PPN using drive on error                          |
        |-----------------------------------------------------|
TUBCNI  |            Error status for next record             | FTRDBA
        |-----------------------------------------------------|
TUBMSG  | Label message length | Label message function code| FTTLAB
        |-----------------------------------------------------|
TUBPHY  |             Physical name for label PCS             | FTTLAB
        |-----------------------------------------------------|
TUBLBL  |                 Label status word                   | FTTLAB
        |-----------------------------------------------------|
TUBRFM  |          Record format (for labels only)            | FTTLAB
        |-----------------------------------------------------|
TUBRCC  |           Record size (for labels only)             | FTTLAB
        |-----------------------------------------------------|
TUBBKL  |            Block size (for lables only)             | FTTLAB
        |-----------------------------------------------------|
TUBEXP  |         Expiration date (for labels only)           | FTTLAB
        |-----------------------------------------------------|
TUBPRT  |            Protection (for labels only)             | FTTLAB
        +-----------------------------------------------------+
```

TUBSTS Bits:

| Bit | Symbol | Meaning |
|---|---|---|
| 0 | TUSNS | Do not schedule this unit (sign bit) |
| 1 | TUSBOT | Beginning of tape |
| 2 | TUSWTL | Write locked |
| 3 | TUSREW | Tape rewinding |
| 12 | TKSMNT | Controller is in maintenance mode (also in TKBSTS) |
| 13 | TKSSIL | Request silence about offline condition (also in TKBSTS) |
| 14 | TKSSCH | Requested scheduled interrupt (also in TKBSTS) |
| 15 | TKSSTD | Started (also in TKBSTS) |
| 16 | TKSSEL | Selected (also in TKBSTS) |
| 17 | TKSOFL | Offline (also in TKBSTS) |

TUBCNF Bits:

| Bit | Symbol | Meaning |
|---|---|---|
| 18 | TUC7TK | 7-track |
| 19 | TUCIRD | Interrupts when rewind done |
| 20 | TUCDMS | Diagnostic mode set |
| 21 | TUCSNS | Force sense |
| 23 | TUCD62 | Drive can do 6250 BPI |
| 24 | TUCD16 | Drive can do 1600 BPI |
| 25 | TUCD80 | Drive can do 800 BPI |
| 26 | TUCD55 | Drive can do 556 BPI |
| 27 | TUCD20 | Drive can do 200 BPI |
| 28 | TUCDIG | DIAG. UUO has been done |

## 85 MEMTAB -- VIRTUAL MEMORY PAGE TABLE

| | |
|---|---|
| Description: | Contains one word per page of core. It is used during swap/page requests in conjunction with the UPT to keep track of where pages end up on the swapping area, and which page to transmit next.<br><br>Under KL-paging, MEMTAB is in Section 3. |
| Defined in: | S.MAC |
| Used by: | VMSER |
| See also: | PT2TAB |

```
+------------------------------------------------------------+
| L | G | I |            See below                           |
+------------------------------------------------------------+
```

The first three bits of a MEMTAB entry are flags to indicate the following:

| Bit | Symbol | Meaning |
|---|---|---|
| 0 | MT.LEF | Last entry in fragment chain |
| 1 | MT.GPB | Return swapping space when I/O done in IP queue |
| 2 | MT.IPC | IPCF page,,addr of packet+.ICPFI in MEMTAB |

The format of bits 3 through 35 of the MEMTAB table entries differs for the status of the page. For a page that is being transmitted to/from disk, the entry contains the disk address in bits 15-35.

The MEMTAB entry for a page in a paging queue contains the job number in Bits 5-14 (MT.JOB). For an IPCF page when the page is in the IP queue, the high-order 3 bits of MT.JOB contain the IPCF header address (remaining 15 bits of address of IPCF header are stored in PT2TAB).

For a page in one of the IN paging queues, the remainder of the word is formatted as follows:

| Bits | Symbol | Meaning |
|---|---|---|
| 24-35 | MT.VPN | Section-relative virtual page number (page is a job page) |
| 22-26 | MT.VSN | Section number. |
| 18-35 | MT.IPA | Address of IPCF header block for this IPCF page. |

## 86 METABL -- META-CHARACTER TABLE

Description: One table entry for each meta-character (function character), specifying characteristics and, in some cases, address of a special action routine for processing the received character.

Indexed by the function code of the Meta character.

Uses the same bits as CHTABL.

Defined in: SCNSER

Used by: SCNSER

Format:

| Code | Name | CHSPO | | CHPUNC | CH2PC | CHEPAR | CHVPOS | CHCRE | CHFILO | CHINVL | CHCNC | CHCRET | CHRIA | CHALT | CHFIL | CHUAE | CHBRK | Disp addr on RECINT or XMTCHR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4000 | ACR | 1 | | 1 | 1 | | | | | | | 1 | | | 1 | | 1 | |
| 4001 | ^U | 1 | | 1 | | | | | | | | | | | | 1 | | METDL |
| 4002 | ^W | 1 | | 1 | | | | | | | | | | | | 1 | | METDW |
| 4003 | DEL | 1 | | 1 | | | | | | | | | | | | | | METDC |
| 4004 | ^H | 1 | | 1 | | | | | | | | | | | 1 | | | METBS |
| 4005 | .TONFC | 1 | | 1 | 1 | | | | | | | | | | | | | METNFC |
| 4006 | .TOHPS | 1 | | 1 | 1 | | | | | | | | | | | | | METHPS |
| 4007 | .TOFLM | 1 | | 1 | | | | | | | | 1 | | | | | | METFLM |

## 87 MFD -- MASTER FILE DIRECTORY

Description: Disk file which tells location of each UFD (User File Directory) in a file structure. There is one MFD included in each file structure.

There is one entry in the MFD for each UFD that has files in the structure.

Position of an entry is of no significance.

Defined in: COMMOD

Used by: FILFND, FILIO, FILUUO

```
+--------------------------------------+
|           Entry for user A           |
|--------------------------------------|
|           Entry for user B           |
|--------------------------------------|
|                  .                   |
|                  .                   |
|                  .                   |
|--------------------------------------|
|           Entry for user Z           |
+--------------------------------------+
```

Format of each entry

```
+--------------------------------------+
|     Project       |   Programmer     |
|--------------------------------------|
|   SIXBIT/UFD/     |       CFP        |
+--------------------------------------+
```

Notes:

1. The MFD is actually the UFD for [1,1]. PPN [1,1] is used only for this purpose.

2. The CFP (Compressed File Pointer) specifies the relative supercluster within the structure where the RIB of the corresponding file can be found.

3. The first entry in the MFD is a pointer to itself (to [1,1].UFD).

```
        Relative      Number of       Relative       Number
CFP=    Unit #     x  Superclusters + Block #    .   Blocks
        within        per             within     -   per
        STR           Unit            that unit  .   Supercluster
```

The number of clusters in a "supercluster" is determined by:

```
number-clusters-in-structure
-----------------------------
          2(18)
```

rounded up by 1. Hence, every structure will have fewer than 2**18 superclusters, and the CFP will fit in 18 bits. The concept of a supercluster is used only in connection with Compressed File Pointers.

## 88 NMB -- FILE NAME BLOCK

Description: Used to remember file name in a project programmer number across all file structures. There is one NMB for each open file of each PPN regardless of how many versions of the files are in existence. This table is linked into the list for each PPN.

Defined in: COMMOD

Used by: FILIO, FILFND, FILUUO

```
             +-----------------------------------------------+
NMBNAM       |File name in SIXBIT                            |
             |-----------------------------------------------|
NMBPPB*      |Next NMB          |X|X|                        |
             |-----------------------------------------------|
NMBRNG(3)    | Ptr to SFD NMB lst  |  Compressed File Pointer | NMBCFP
             |-----------------------------------------------|
NMBACC*(6)   |  First ACC block    | File ext in SIXBIT (2)   | NMBEXT
             |-----------------------------------------------|
NMBFSN(7)    |FSN  |                                         |
             |-----------------------------------------------|
NMBKNO(4)    |            Know bits for this file            |
             |-----------------------------------------------|
NMBYES(5)    |            Yes bits for this file             |
             |-----------------------------------------------|
NMBCNT       |                  Use count                    |
             +-----------------------------------------------+
```

Notes:

1. If a file is being superseded, there are Access Blocks corresponding to each existing version of the file, but only one NMB.

2. The compressed file pointer in NMBEXT is copied from the second word of the UFD entry for this file. Its value is the logical super-cluster number within the structure of the file's RIB (see UFD).

3. If this NMB is for an SFD (NMPSFD (bit 34) set in NMBSFD), NMBRNG is a pointer to a NMB list of files under the SFD and the extension is assumed to be 'SFD'.

4. NMBKNO - Bit 36-n set if we know whether or not file exists in structure n (can be either way, and NMBYES tells which is true).

5. NMBYES - Bits 36-n set if file definitely exists in structure n.

6. Access tables are linked into a ring for each file, starting and ending at NMBACC. All addresses that point to ACC blocks are even multiples of 4; the link back to NMBACC is not. If there are no ACC blocks in the ring at some time, the left half of NMBACC does not contain zero, but rather its own address.

7. NMBFSN holds the file structure number in bits 0-5.

88.1 NMBPPB -- Next NMB

```
+-----------------------------------------------------+
| Addr of next NMB      |x|y|     Reserved            |
+-----------------------------------------------------+
0                      15   17                       35
```

| Bits | Byte Pointer | Description |
|---|---|---|
| 0-15 | | Address of next NMB for this directory (low order 2 bits=0). |
| 16 | NMPUPT | If this bit is on, NMBPPB is the location of the father SFD for this list of NMBs (only set in the last NMB in the list). |
| 17 | NMPSFU | Bit on in AC, never in core, if the location returned (by UFORSF) is an SFD ACC. Off if the location is a UFB. |

88.2 NMBACC -- First ACC

```
+-----------------------------------------------------+
|      NMBACC                |         NMBEXT         |
+-----------------------------------------------------+
0                           17                       35
```

| Word Label | Bits | Contents |
|---|---|---|
| NMBACC | 0-17 | First ACC in access ring with this filename/PPN. If ring is empty, the byte is not 0, instead it points to itself. In this way no special checking is needed to add or delete access blocks from ring. |
| NMBEXT | 18-35 | Holds the left-justified SIXBIT file extension. |

## 89 NUMTAB -- TABLE OF GETTAB TABLES

Description: Contains monitor table address and bits that indicate whether there may be segment data or process data.

This is the table that is referenced by the GETTAB UUO to pass back the information requested.

Defined in: UUOCON

Used by: COMMON, UUOCON

GETTAB Table: .GTSLF (23)

```
        Bits*          GETTAB Table Address
+-------------------------------------+
|                  |       JBTSTS     |
|-------------------------------------|
|                  |       JBTADR     |
|-------------------------------------|
|                  |       JBTPPN     |
|-------------------------------------|
|        "         |         "        |
+-------------------------------------+
```

Notes:

1. ABSTAB (loc 410) in COMMON contains the absolute address of NUMTAB.

2. Bits 0-8 hold the maximum size of the table if it is a regular table

3. Bits 9-11 contain one of the following codes:

| Code | Meaning |
|---|---|
| 0 | undefined in this monitor |
| 1 | index by item type |
| 2 | index by job number |
| 3 | index by job or segment |
| 4 | index by job data in PDB |
| 5 | index by negative and positive offsets |

## 90 PAGTAB -- PAGE TABLE

Description: This table is used to keep track of user core. It contains one word for each page of physical core. PAGPTR, defined in COMMON, contains the starting address for the linked list of free pages. In addition, each segment in core has its own linked list of pages and is addressed through the EPT at location 412.

Under KL-paging, PAGTAB is in Section 3.

Defined in: S.MAC

Used by: KLSER, SYSINI, VMSER

See also: PT2TAB

```
Physical
 Page    0 1 2 3 4 5 6 7 8          17 18                         35
        +-------------------------------------------------------+
   0    | Page bits      |          |Ptr to next phys pg of list|
        |-------------------------------------------------------|
   1    |                |                                      |
        |-------------------------------------------------------|
   2    |                |                                      |
        |-------------------------------------------------------|
                                 .    .    .
```

Bit definitions:

| Bits | Label | Meaning |
|---|---|---|
| 0 | FREPAG | On if page is not in some job's addressing space. |
| 1 | LOKPHB | On if this page is contained in a segment that is locked in physically contiguous memory. |
| 2 | LOKEVB | On if this page is contained in segment that is locked virtually contiguous in the exec addressing space. |
| 3 | LOKIPB | On if this page is contained in a segment that is locked in place. |
| 4 | NXMBIT | On if this page is below MEMSIZ but is non-existent. |
| 5 | MONTRB | On if this page is contained in the monitor. |
| 6 | IPCBIT | On if this page is owned by IPCF. |
| 7 | TNCSHB | On if this page is temporarily uncached. |
| 8 | CONTGB | On if this range of pages must be physically contiguous (for DX20 microcode, CI20 disks,...). |

91 PB -- PATH BLOCK

Description: Contains information describing a specific path to a specific node on a CI network, status of the virtual circuit to that node, and a linked list of Connection Blocks for connections to applications on that node.

Path Blocks are created by KLPSER when a START datagram is received from a previously unknown CI node.

Defined in: SCAPRM

Used by: KLPSER, MSCCOM, RAXKON, SCASER, SCSUUO

See also: CB, PCB, SB

| Symbol | Map |
|---|---|
| .PBPCB | Port Control Block address |
| .PBDPN | Destination port number (a CI node address) |
| .PBIDX | Indexes (1) |
| .PBVCS | Status of virtual circuit (2) |
| .PBFLG | Flags (3) |
| .PBSST | Start sequence timer |
| .PBTIM | Time last message received |
| .PBTWQ | Forward link for SCA work queue for this path block |
| .PBBWQ | Backward link for SCA work queue for this path block |
| .PBCLC | Total number of locked connections |
| .PBOBB | Saved buffer to use in SC.SNM |
| .PBFCB | Pointer to first connection block |
| .PBLCB | Pointer to last connection block |
| .PBDPC | Destination port characteristics |
| .PBDCR | Destination code revision |
| .PBDPF | Destination port functionality |
| .PBDPS | Destination port state |

```
.PBCPU   |  CPU that owns this path block (SMP systems only)      |
         +--------------------------------------------------------+
```

Notes:

1. The indexes stored in .PBIDX are contained in the following fields of the word:

| Bits | Symbol | Contents |
|---|---|---|
| 0-11 | PBPBI | This path block index |
| 12-23 | PBNPI | Next path block index |
| 25-35 | PBSBI | System block index |

2. The state of the virtual circuit is stored in .PBVCS. The codes stored in the word have the following meanings:

| Code | Meaning |
|---|---|
| 0 | Closed |
| 1 | Start sent |
| 2 | Start received |
| 3 | Opened |

3. The flags that are stored in .PBFLG indicate the following states:

| Bit | Symbol | Meaning |
|---|---|---|
| 0 | PB.TMG | Timed message |
| 1 | PB.OVC | Virtual circuit needs to be opened |
| 2 | PB.OFL | Node is offline |
| 18 | PB.NTC | Virtual circuit needs to be closed |
| 19 | PB.OKO | Virtual circuit is ready to be opened |
| 20 | PB.WFI | Waiting for new Identification-received message |

## 92 PCB -- PORT CONTROL BLOCK

Description: Contains queue structures and other data shared by the CI20 microcode and the CI20 driver in TOPS-10. The Port Control Block is the interface between a CI20 and the KL10.

One PCB is created for each CI20 detected by AUTCON during system initialization.

Defined in: KLPPRM

Used by: KLPSER

See also: CB, PB, SB

The pointers stored in the beginning of the PCB are defined by the CI20 microcode; therefore, they are stored as physical addresses. Starting at .PCKCT, pointers are stored by software as virtual addresses.

| Symbol | Map |
|---|---|
| .PCBDT | Address of Buffer Descriptor Table (BDT) |
| .PBMQE | Message queue entry length |
| .PBDQE | Datagram queue entry length |
| .PBRQE | Reserved queue entry length |
| .PCCQ3* | Command queue 3 |
| .PCCQ2* | Command queue 2 |
| .PCCQ1* | Command queue 1 |
| .PCCQ0* | Command queue 0 |
| .PCRSQ* | Response queue |
| .PCMFQ* | Message free queue |
| .PCDFQ* | Datagram free queue |
| .PCRFQ* | Reserved free queue |
| .PCRSV | Reserved for this port |
| .PCER0 | Error word 0 (1) |
| .PCER1 | Error word 1 (API function word) |

| | |
|---|---|
| .PCER2 | Error word 2 (register data) |
| .PCER3 | Error word 3 (channel logout word 1) |
| .PCER4 | Error word 4 (channel logout word 2) |
| .PCPBA | Physical address of start of PCB |
| .PCPIA | Priority interrupt level |
| .PCIVA | Interrupt vector address (not used) |
| .PCCCW | Port's channel command word |
| .PCRSP | Reserved for this port |
| .PCVPO | Virtual to physical offset for references into the PCB (use XMOVEI AC,.PCxxx/SUB AC,.PCVPO) |
| .PCSTS | Status flags (2) |
| .PCFQC | Number of datagrams and messages to put back on the free queue when KLIPA restarts after a SET MEMORY OFFLINE command |
| .PCONN | The CI node number for this CPU |
| .PCSBK | System block address, indexed by CI node number |
| .PCPBK | Path block address, indexed by CI node number |
| .PCRIS | Request-id status and flags, indexed by CI node no.(3) |
| .PCRIT | Request-id timer, indexed by CI node number |
| .PCRIN | Next node for request-id poller |
| .PCCPU | CPU that owns this CI (SMP systems only) |
| .PCKCT | CPU uptime when last command was queued |
| .PCKRT | CPU uptime when last response was received |
| .PCKAC | Total number of Keep-Alive Failures |
| .PCKCI | CONI at last Keep-Alive Failure |
| .PCKAT | System uptime at last Keep-Alive Failure |
| .PCCSR | CONI at last interrupt |

```
.PCCRA  |      CRAM address                                           |
        |-------------------------------------------------------------|
.PCCDL  |      Left half CRAM data                                    |
        |-------------------------------------------------------------|
```

```
        |---------------------------------------------------------|
.PCCDR  |     Right half CRAM data                                |
        |---------------------------------------------------------|
.PCLGO  |     Channel logout word 0                               |
        |---------------------------------------------------------|
.PCLG1  |     Channel logout word 1                               |
        |---------------------------------------------------------|
.PCLG2  |     Channel logout word 2                               |
        |---------------------------------------------------------|
.PCECW  |     Port's CCW at time of error                         |
        |---------------------------------------------------------|
.PCLKE  |     Date and time of last KLIPA error                   |
        |---------------------------------------------------------|
.PCCTM  |     Timer for next periodic read-counters               |
        |---------------------------------------------------------|
.PCCJB  |     Job number of job that owns the counters            |
        |---------------------------------------------------------|
.PCCTR  /     Statistics counters (date/time when last read),     /
        /     Followed by variable-length counters data           /
        |---------------------------------------------------------|
.PCMJB  |     Job number of job doing a mainenance function       |
        |---------------------------------------------------------|
.PCMTI  |     Maintenance mode message timer                      |
        |---------------------------------------------------------|
.PCMFL  |     Maintenance mode message flag (4)                   |
        |---------------------------------------------------------|
.PCMCN  | Buffer name that Close Buffer command is for            |
        |---------------------------------------------------------|
.PCMCF  | -1 if a Close Buffer response was received for the      |
        |          buffer in .PCMCN                               |
        |---------------------------------------------------------|
.PCCDB  | Address of simulated CDB for DIAG. UUO                  |
        |---------------------------------------------------------|
.PCULB  /          Header of variable-length microcode            /
        /          loader parameter block (5)                     /
        +---------------------------------------------------------+
```

The blocks designated by * in the PCB are variable-length queues. The Port Control Block queues are formatted as follows:

| Offset | Symbol | Contents |
|---|---|---|
| 0 | .PQIWD | Interlock word. The interlock is a simple AOS-style interlock. This word contains -1 if the queue is available to be manipulated, and a zero or positive value if the queue is locked against access. The interlock word is set and tested using a AOSx instruction. |
| 1 | .PQFLI | Forward link (FLINK) word. This word contains the physical memory address of the |

| | | |
|---|---|---|
| | | first entry on that queue. If the queue is empty, the FLINK word contains its own physical memory address. |
| 2 | .PQBLI | Backward link (BLINK) word. This word contains the physical memory address of the last entry in the queue. If the queue is empty, the previous word (.PQFLI) contains its own address, and the contents of this word (.PQBLI) are indeterminate. |

Notes:

1. The error word .PCERO contains the following fields for the error data:

| Bits | Symbol | Contents |
|---|---|---|
| 0 | EO.CMD | Error occurred while reading a command queue |
| 1-2 | EOQUE | Command queue number (if Bit 0 is set) |
| 3 | EO.RES | Error occurred while building a response |
| 4-11 | EOMBZ | Must be zero |
| 12-35 | EOFLI | Forward link of entry that is in error |

2. The status word .PCSTS contains flags designating the following states:

| Bit | Symbol | Meaning |
|---|---|---|
| 0 | ST.STP | KLIPA was stopped last second |
| 1 | ST.MFL | Memory is being set offline. KLIPA should be shut down and restarted when ST.RES is set. |
| 2 | ST.RES | Restart the KLIPA after SET MEMORY OFFLINE |
| 3 | ST.MAI | Maintenance mode enabled |
| 4 | ST.WAB | Wire A is bad |
| 5 | ST.WBB | Wire B is bad |
| 6 | ST.DED | KLIPA is dead |
| 7 | ST.PTH | Last path for loopback packet |
| 8 | ST.CQA | Queued I/O for this KLIPA |
| 9 | ST.RDY | KLIPA initialization complete |

3. The status and flags for the request-id are stored in .PCRIS, and have the following meanings:

| Bit | Symbol | Meaning |
|---|---|---|
| 0 | RI.PTH | Path last ID sent on (off = A, on = B) |
| 1 | RI.PAO | Path A is open |
| 2 | RI.PBO | Path B is open |
| 3 | RI.NRA | No response on Path A |
| 4 | RI.NRB | No response on Path B |
| 5 | RI.TRY | Request-id attempt (off=first, on=second) |

| | | |
|---|---|---|
| 6 | RI.WFR | Waiting for response |
| 12-17 | | Total number times there was no response |

4. The maintenance mode message status flag stored in .PCMFL is initialized to -1. The word contains 0 if the packet was received with no errors. The flag is set to 1 if the packet was received with an error.

5. The microcode parameter block (.PCULB) is a variable-length block that contains information used by BOOT when the KLIPA microcode is (re)loaded. This block is defined in S.MAC and is also used for other devices, such as TX01, TX02, DX20, NIA20, and, of course, CI20.

93 PDB -- PROCESS DATA BLOCK

Description: One PDB for each active job. Set up by CREPDB routine in DATMAN when the job is initialized.

Defined in: COMMON (prototype)
DATMAN (modify and find routines)

Used by: CLOCK1, COMCON, COMMOD, COMMON, CORE1, FILFND, IPCSER, KLSER, NETSER, QUESER, SCHED1, SEGCON, UUOCON, VMSER

```
                                                                  Conditional
                                                                    Assembly
              +-----------------------------------------------+
.PDIPT,,      | (1) | ICPT (MCU)    |Quantum run time          |
.PDQNT        |     |               |                          |
              |-----------------------------------------------|
.PDCNO        |User's charge number                           |   FTCNO
              |-----------------------------------------------|
.PDKCT        |Kilo-core ticks for the job                    |   FTKCT
              |-----------------------------------------------|
.PDNM1        |First half of user's name in SIXBIT            |   FTUNAME
              |-----------------------------------------------|
.PDNM2        |Second half of user's name in SIXBIT           |   FTUNAME
              |-----------------------------------------------|
.PDRTM        |Job's incremental runtime                      |   FTTIME
              |-----------------------------------------------|
.PDTTM        |Job's total runtime                            |   FTTIME
              |-----------------------------------------------|
.PDTT2        |Additional runtime in fractional jiffies       |   FTTIME
              |-----------------------------------------------|
.PDEBT        |Total EBOX time in jiffies                     |FTKL10,FTTIME
              |-----------------------------------------------|
.PDEB2        |Remainder in EBOX counts                       |FTKL10,FTTIME
              |-----------------------------------------------|
.PDMBT        |Total MBOX time in jiffies                     |FTKL10,FTTIME
              |-----------------------------------------------|
.PDMB2        |Remainder in MBOX counts                       |FTKL10,FTTIME
              |-----------------------------------------------|
.PDPGM        |Program to run on Control-C or RUN             |   FTSET
              |-----------------------------------------------|
.PDABS  (2)   | Addr. break settings | Break address          |
              |-----------------------------------------------|
.PDCVL  (3)   |      CVPL            | CPPL                   |
              |-----------------------------------------------|
.PDMVL        |      MVPL            | MPPL                   |
              |-----------------------------------------------|
```

| Symbol | Note | Contents (left half) | Contents (right half) | Feature test |
| --- | --- | --- | --- | --- |
| .PDDVL | (4) | Pointer to table of DDBs with log. names | | FTHSLN |
| .PDIPC | (5) | First packet | Send/receive ctrs | FTIPCF |
| .PDIPA | (6) | IPCF statistics | | FTIPCF |
| .PDIPQ | (7) | Flags and quotas | | FTIPCF |
| .PDIPL | | Interlock word | | |
| .PDPID | | PID for PID-specific receives | | |
| .PDIPI | | PID of this job's [SYSTEM]INFO | | FTIPCF |
| .PDIPN | | Last entry in IPCFQ | | FTIPCF |
| .PDEQJ | | 0 | Pointer to job queue | FTEQDQ |
| .PDQSN | | FILDAE seq. # | QUEUE. UUO seq. # | |
| .PDEPA | | 0 | Addr of packet response to pseudo-process msg | |
| .PDEQQ | (8) | Flags | ENQ quota | FTEQDQ |
| .PDJSL | (9) | Job search list | | FTSTR |
| .PDSCX | | Job's saved context word | | |
| .PDDIA | | Location of DIAG. DDB for job | | FTDHIA |
| .PDSTR | | Structure the program came from | | |
| .PDNAM | | Name of the program | | |
| .PDDIR | | Directory the program came from | | |
| .PDSFD | | Path to program | | |
| .PDDFL | (10) | Word containing user-defined defaults | | FTSET |
| .PDCAP | (12) | Maximum privileges allowed | | FTPRV |
| .PDACS | | Account string Eight words | | FTACCT |
| .PDVKC | | Virtual time-core interval | | FTKCT,FTACCT |
| .PDUUC | | Count of UUOs done by this job | | FTACCT |
| .PDHZF | | HPQ fit flag | | FTHPQ |

|----------------------------------------|

| Symbol | | Contents | Condition |
|---|---|---|---|
| .PDPST | | Negative of swapout time | FTPSCD |
| .PDOBI | (11) | Operator/batch information | |
| .PDSTM | | Time of last reset | |
| .PDLBS | | Default size of large disk buffer<br>LH is set by UUO, RH is set by command | |
| .PDOSL | | Old-style LIB PPN | |
| .PDCMN | | AOBJN pointer to user-defined command list | |
| .PDUNQ | | User-defined command pointers<br>LH is user UNQTAB<br>RH is address of user command block | |
| .PDSAC | (13) | Address of first context block | M.CTX |
| .PDCTC | (13) | Address of current context block | M.CTX |
| .PDCTQ | (13) | Context quota word | M.CTX |
| .PDCTU | (13) | Context use word | M.CTX |
| .PDCTX | (14) | Context flag word | |
| .PDTMI | | Initial value for virtual timer traps | |
| .PDTMC | | Countdown value for vir timer traps, or old PC | |
| .PDVRT | | Virtual memory paging rate | |
| .PDSCS | | Addr. of process queue block for SCS. UUO | M.SCA |
| .PDEJB | | Addr. of Ethernet job block | M.ENET |
| .PDCST | (15) | Reserved for customer definition | |

Notes:

1. Bit 0 of .PDIPT is the PDMSWP bit that is set to indicate the expiration of the MCU (minimal care utilization).

2. .PDABS contains address break settings:

| Bit | Symbol | Meaning |
|---|---|---|
| 0 | OC.BCI | Break on instruction fetch |
| 1 | OC.BCD | Break on data fetch |

| | | |
|---|---|---|
| 2 | OC.BCW | Break on write |
| 3 | OC.BCM | Break on MUUO reference to address (software) |
| 4 | OC.ABE | Address break enabled |
| 5 | OC.FEP | Follow exec paging |
| 6 | OC.FUP | Follow user paging |
| 7 | OC.BSU | Break address and conditions set by UUO. |

3. .PDCVL - Bit 18 is set if the CPPL is a limit rather than a guideline.

4. .PDDVL - Pointer to table of pointers to DDBs owned by this job and having logical names, or zero if no such DDBs, or -1 if too many to fit in the table (4 words = 8 DDBs).

5. .PDIPC

| Bits | Contents |
|---|---|
| 0-17 | Pointer to first packet |
| 18-26 | Packets sent and not received |
| 27-35 | Packets waiting to be received |

6. .PDIPA - Left half contains the count of sends since LOGIN. Right half contains the count of receives since LOGIN.

7. .PDIPQ

| Bits | Symbol | Meaning |
|---|---|---|
| 0 | IP.DBS | Receiver is disabled |
| 1 | IP.HBS | Quotas have been set |
| 2 | IP.DPR | At least 1 PID dropped on RESET |
| 3 | IP.DPL | At least 1 PID dropped on LOGOUT |
| 4 | IP.LOK | Interlock bit for this job's IPCF receive queue |
| 9-17 | IP.JOB | Job whose IPCF queue has been locked |
| 18-26 | | Send quota |
| 27-35 | | Receive quota |

8. .PDEQQ - Bit 0, EQ.HBS, indicates the quota has been set

9. .PDJSL - The number of words in the PDB is a function of the maximum number of file structures in a search list (.SLMXJ= 10 (decimal)). For each file structure there will be a 9-bit byte plus an additional two bytes for the fence and stop markers. For file structure bytes, the following definitions exist:

| Value | Symbol | Use |
|---|---|---|
| 400 | | Spare bit |
| 200 | FS.NCR | No-create |
| 100 | FS.WLK | Software write-lock |

| | |
|---|---|
| 77 | File structure number |

System search list is maintained in COMMOD starting at location SYSSL and also consists of 9-bit bytes.

The fence marker will have a value 1 greater than the maximum file structure number. The stop marker will have a value 1 greater than the fence marker.

10. .PDDFL contains the following fields:

| Bits | Description |
|---|---|
| 0-8 | Default file protection |
| 9 | Non-zero if default protection was specified |
| 10 | Non-zero if file daemon specified protection |
| 11 | Use default file specification on RUN and GET |
| 12 | Don't ask about detached jobs on LOGIN |
| 18-26 | File protection from FILDAE |
| 27-35 | Default number of disk buffers |

11. Bits for .PDOBI are:

| Bits | Description |
|---|---|
| 0-1 | Write to operator values |
| 2-4 | Operator privilege type |
| 10 | Batch stream number set |
| 12-17 | Batch stream number |

12. For the values for .PDCAP, see JBTPRV.

13. These context words are conditionally assembled depending on the value of M.CTX, which causes CTXSER to be loaded if necessary. This word is usd only by CTXSER.

14. This context word is always assembled, independant of state of M.CTX and the presence of CTXSER. .PDCTX contains flags and fields used for context creation and by the scheduler. In general, left half bits are of a transient nature affecting only the current context, while right half bits apply to those operations which are of a job-wide nature. .PDCTX is referenced by CTXSER and SCHED1 although SCHED1 only reads the state of the scheduler bit (CT.SCDE).

| Bits | Symbol | Meaning |
|---|---|---|
| 0 | CT.SCD | Scheduler requesting context save |
| 1 | CT.ATO | Auto-save in progress |
| 3 | CT.TOP | Creating a new top-level context |
| 4 | CT.SWT | Switch to an existing context |
| 5 | CT.UUO | Context saved with CTX. UUO |

| | | |
|---|---|---|
| 6 | CT.PRN | Physical device search on RUN UUO |
| 18 | CT.LGO | Job is logging out |
| 19 | CT.MTJ | Job is migrating |
| 27-35 | CT.MFC | Migrate's first context number |

15. .PDCST is always at the end of the PDB, and is equivalent to one or more words reserved for customer definition. The symbol M.PCST is equivalent to the number of words reserved here.

94 PPB -- PROJECT PROGRAMMER NUMBER DATA BLOCK

Description: Contains information pertaining to all files belonging to one PPN. There is one PPB for each PPN with any active files; it is linked into a list for the system, starting at SYSPPB, and is also available through JBTPPB.

Defined in: COMMOD

Used by: FILFND, FILUUO

```
            +-------------------------------------------+
PPBNAM      | Project Number       | Programmer number  |
            |-------------------------------------------|
PPBSYS      | Next PPB in system   |                    |
            |-------------------------------------------|
PPBUFB      | First UFB            |                    |
            |-------------------------------------------|
PPBNMB(1)   | First NMB            |                    | PPBNLG
            |-------------------------------------------|
PPBCNT      |        Use count for the PPB              |
            |-------------------------------------------|
PPBKNO(2)   | KNO bits for UFD                          |
            |-------------------------------------------|
PPBYES(3)   | YES bits for UFD                          |
            |-------------------------------------------|
PPBLOK      | Bits  n+1=1 if UFD for FSN is interlocked|
            +-------------------------------------------+
```

Notes:

1. Bit 35 of PPBNMB is the PPPNLG bit; PPN is not logged in. This bit, when set, indicates the PPN is logged in. Used to flush PPB immediately when last file becomes dormant in PPB.

2. PPBKNO - Bit 36-n set if monitor knows whether or not UFD for this PPN exists in structure n.

3. PPBYES - Bit 36-n is set if the UFD for this PPN definitely exists in structure n.

## 95 PTYTAB -- PSEUDO-TERMINAL DDB TABLE

Description: Table of pseudo-terminal (PTY) DDBs. There is one entry for each PTY in the system. (See TTYTAB also.)

Defined in: COMMON

Used by: PTYSER, SCNSER, SYSINI

```
         +--------------------------------------+
PTYTAB:  |                  |   DDB address     | PTY0
         |--------------------------------------|
         |                  |     "     "       | PTY1
         |--------------------------------------|
         |                  |     "     "       | PTY2
         |--------------------------------------|
         /                    . . .             /
         |--------------------------------------|
         |                  |   DDB address     | PTYn
         +--------------------------------------+
```

## 96 PT2TAB -- PAGE SECTION NUMBERS

Description: Used to keep track of user core, this table contains one word for each page of physical core. Indexed by page number, this table is complementary to PAGTAB.

Defined in: S.MAC

Used By: KLSER, SYSINI, VMSER

See also; PAGTAB

```
Physical
 Page

        +--------------------------------------------------------+
   0    | | L | Virtual section no.   | Backward link addr       |
        |--------------------------------------------------------|
   1    | | L | Virtual section no.   | Backward link addr       |
        |--------------------------------------------------------|
   2    | | L | Virtual section no.   | Backward link addr       |
        |--------------------------------------------------------|
                                 .    .    .
```

| Bits | Symbol | Meaning |
|---|---|---|
| 0-17 | P2.VPN | Virtual section number for swapout (see below) |
| 1 | P2.LIP | Lock In Progress (page to be returned) |
| 18-35 | P2.BLK | Backward link to previous page |

P2.VPN contains the next virtual page number for a swap request. For a paging queue request, this field stores the virtual page number for this page for job pages. For IPCF pages, this field contains the low-order 15 bits of the address of the IPCF queue header. The size of the field is defined by the symbol P2.SPN, which has a value of 15 (decimal).

## 97 QBITS -- WAIT STATE CODE REQUEUE TABLE

Description: Specifies a requeue dispatch address and a transfer table address for requeueing a job as a function of its wait state code. There is one entry for each wait state code, and the value of the wait state code is the index factor of the table. The wait state code is the queue number, offset by the first sharable resource code.

Defined in: SCHED1

Used by: SCHED1

See also: JBTSTS

| Wait State | Dispatch Address | Transfer Table Address |
|---|---|---|
| 0 | QRNT | QRNW |
| 1 | QWST | QWSW (-1) |
| 2 | QTST | QTSW |
| 3 | QDST | QDSW (-1) |
| 4 | QPST | QPSW (-1) |
| 5 | QAUT | QAUW (-1) |
| 6 | QDAT | QDAW (-1) |
| 7 | QCBT | QCBW (-1) |
| 10 | QDTT | QDTW (-1) |
| 11 | QIPT | QIPW (-1) |
| 12 | QCXT | QCXW (-1) |
| 13 | QDCT | QDCW (-1) |
| 14 | QCAT | QCAW (-1) |
| 15 | QMMT | QMMW |
| 16 | QEVT | QEVW |

| | | |
|---|---|---|
| 17 | QEQT | QEQW |
| 20 | QMCT | QMCW (-1) |
| 21 | QIOWT | QIOWW (-1) |
| 22 | QTIOWT | QTIOWW |
| 23 | QDIOWT | QDIOWW (-1) |
| 24 | QPIOWT | QPIOWW (-1) |
| 25 | QPQIOT | QPQIOW (-1) |
| 26 | QSLPT | QSLPW |
| 27 | QEWT | QEWW |
| 30 | QNAPT | QNAPW (-1) |
| 31 | QNULT | QNULW |
| 32 | QJDCT | QJDCW |
| 33 | QSTOPT | QSTOPW |

Notes:

A transfer table address of -1 implies that the code at the dispatch address will only modify the wait state code, and no physical queue transfer will take place (that is, a short term state change).

## 98 QUEUE TRANSFER TABLE

Description: Contains input parameters for Queue Transfer routine. There is one such table for each different type of queue transfer.

Defined in: SCHED1

```
+-------------------------------------+
|        Place       |    Function    |
|-------------------------------------|
|        Quant       |      Dest      |
+-------------------------------------+
```

Notes:

Place>0 Transfer to beginning of destination queues.

Place<0 Transfer to end of destination queue.

Function is the address of the QXFER routine to be used. It defines the type of transfer. There are three possible values:

1. QFIX - Destination queue specified in this table

2. QLINK - Destination queue is a function of source queue

3. QJSIZ - Destination queue is a function of job size

Dest specifies the destination queue.

1. If Function is QFIX, Dest is the destination queue number.

2. If Function is QLINK, Dest is the address of a Job Size-Queues Progression Table that specifies destination queue as a function of source queue.

3. If Function is QJSIZ, Dest is the address of a Job-Size-Queue Table which specifies destination queue as a function of job size.

Quant specifies the change to the job's quantum run time as follows:

If Quant<0, no change to quantum runtime.

If Quant>0, reset the quantum run time as follows:

1. If Function is QFIX, set quantum run time to the value of Quant.

2. If Function is QLINK or QJSIZ, Quant is the address of a Quantum Time Table, which has entries corresponding to the entries in the table used to determine the destination queue. The entry in the same position as the selected destination queue is used to reset the quantum run time.

See BQFIX and following in SCHED1.

99 QUEUE TABLE FOR JOB SCANNING

Description: Determines manner in which job queues are scanned by routine QSCAN in SCHED1. There is one entry for each queue to be considered, and entries are in the order that the corresponding queues are to be considered.

Defined in: COMMON

Used by: SCHED1

```
+--------------------------------------+
|      Queue        |    Scan code     |
|--------------------------------------|
|        "          |       "          |
|--------------------------------------|
|        "          |       "          |
|--------------------------------------|
|        "          |       "          |
|--------------------------------------|
|                   0                  |
+--------------------------------------+
```

The Scan Code is the address of a scanning routine in QSCAN. The routines are:

| Routine | Function |
|---|---|
| QFOR | Scans whole queue forward, first in-core then out-core |
| QBAK | Scans whole queue backward, first out-core then in-core |
| IQFOR | Scans in-core queue forward |
| IQBAK | Scans in-core queue backward |
| IQFOR1 | Scans in-core queue for first member |
| IQBAK1 | Scans in-core queue backward (all but first member) |
| OQFOR | Scans out-core queue forward |
| OQBAK | Scans out-core queue backward |
| OQFOR1 | Scans out-core queue for first member |
| OQBAK1 | Scans out-core queue backward (all but first member) |
| SQFOR | Scans out-core subqueues (PQ2 class swap-in scan) |
| BGFOR | Scans out-core background batch subqueue (PQ2 class swap-in) |
| ISSFOR | Scans in-core subqueues (PQ2 class scheduling scan) |
| IBBFOR | Scans in-core background batch subqueue (PQ2 class scheduling) |
| OSSFOR | Scans out-core subqueues (PQ2 class lost-time scan) |
| IRRFOR | Scans just swapped in queue, then QP2 in-core queue |
| IGFOR | Scans just swapped in queue and jobs waiting for high segment |
| OLFOR | Scans background batch, B.B. JIL, regular output queue, PQ2 in-core |

Queues are scanned, in specified manner, in the order in which their entries appear in the table, and a zero entry terminates the table. See SSCAN, SSCAN1, ISCAN, and OSCAN, in COMMON.

100 QQSTAB -- QUANTUM TIME QUEUE TABLE

Description: Specifies value to which a job's quantum run time is reset on certain types of queue transfers.

Defined in: COMMON

Used by: SCHED1

```
+---------------------------------------+
|           Quantum time 1              |
|---------------------------------------|
|           Quantum time 2              |
|---------------------------------------|
|                 -                     |
|---------------------------------------|
|                 -                     |
|---------------------------------------|
|                 -                     |
|---------------------------------------|
|                 0                     |
+---------------------------------------+
```

Quantum time is in jiffies (power line frequency).

A 0 entry indicates end of table.

## 101 QTTAB -- QUEUE PROGRESSION QUEUE TABLE

Description: Specifies the queue for a job to be put into as a function of the queue it is in.

A Queue Progression Table is specified in the Transfer Table for link-type queue transfers. (Refer to Transfer Table.)

Defined in: COMMON

Used by: SCHED1

| | |
|---|---|
| Source queue 1 | Destination queue 1 |
| Source queue 1 | Destination queue 2 |
| - | - |
| - | - |
| - | - |
| 0 | 0 |

Each entry is a queue number.

A 0,,0 entry indicates end of table.

## 102 REQTAB -- SHARABLE DEVICE REQUEST TABLE

Description: Tells how many jobs require use of each sharable device. Each entry is referenced by its own label.

An entry contains -1 if no job wants that resource. If a job uses the device, the entry is incremented to 0. Each additional job that asks for the device while it is in use increments the entry by one, and must be requeued to the corresponding sharable resource wait queue.

Defined in: SCHED1

Used by: CLOCK1, SYSINI

See also: AVALTB

The words in REQTAB are stored in the following order. However, some of the words may not be included in all systems.

| Word | Symbol | Resource |
|---|---|---|
| 0 | AUREQ | Alter disk UFD quota |
| 1 | DAREQ | Disk storage allocation |
| 2 | CBREQ | Disk core block scan |
| 3 | DTREQ | DECtape control |
| 4 | IPREQ | IPCF interlock |
| 5 | CXREQ | Context save |
| 7 | DCREQ | Data control (magtape and DECtape) |
| 10 | CAREQ | Semi-permanent core allocation |
| 11 | MMREQ | Memory management |
| 12 | EVREQ | Exec virtual memory |
| 13 | EQREQ | ENQ/DEQ |
| 14 | MCREQ | Monitor I/O disk cache |

Notes:

1. Entries in this table may be tested and incremented simultaneously, as follows:

```
AOSE XXREQ
PUSHJ P,     XXWAIT
  -
  -
  -
```

   If the resource was available, the routine may continue. Otherwise, the job must be requeued to wait for it.

2. Table AVALTB has entries corresponding to the entries in REQTAB.

3. The AVALTB entries are built by the conditionally assembled RWAITS MACRO entries in S.MAC; therefore, some of the above listed entries will not be present in all systems.

103 RIB -- RETRIEVAL INFORMATION BLOCK

Description: Disk block containing descriptive information about a file. There is one prime RIB for each file. If a file needs more retrieval pointers than can fit in a single RIB, a second (extended) RIB block is allocated to hold the additional pointers (and so on). The last block(s) of a file is (are) a copy of the prime RIB, called the redundant RIB.

Defined in: COMMOD

Used by: FILFND, FILIO, FILUUO, ONCMOD, REFSTR

| | | | | |
|---|---|---|---|---|
| RIBFIR | -Number of retrieval pointers | | | First pointer address |
| RIBPPN | Project # | | | Programmer # |
| RIBNAM | File name in SIXBIT | | | |
| RIBEXT* | File extension | | | Access date |
| RIBPRV* | Access | Mode | Creation time | Creation date |
| RIBSIZ | File length in words | | | |
| RIBVER | Version number (as in .JBVER) | | | |
| RIBSPL | Possible user file name when spooled | | | |
| RIBEST | Estimated length of file in blocks | | | |
| RIBALC | Number of blocks allocated for file (Including RIB's) | | | |
| RIBPOS | Logical block # in structure | | | |
| RIBFT1 | Word for future use by DEC | | | |
| RIBNCA | Non-privileged word for customer to define | | | |
| RIBMTA | Tape label if file on magtape | | | |
| RIBDEV | Name of structure containing file | | | |
| RIBSTS* | Status bits | | | |

| | | |
|---|---|---|
| RIBELB* | Logical block # where bad region begins | |
| RIBEUN | Err unit # in structure | Number bad blks in region |
| RIBQTF* | FCFS quota for this PPN in this structure (UFD only) | |
| RIBQTO* | Logged out quota this PPN in this STR (UFD only) | |
| RIBQTR* | Reserved quota this PPN in this STR (UFD only) | |
| RIBUSD* | No. of blocks used when job was logged out (UFD only) | |
| RIBAUT | Author PPN writing the file | |
| RIBNXT | Next STR for this file | |
| RIBPRD | Prev STR for this file | |
| RIBPCA | Privileged argument for customer use | |
| RIBUFD | Blk # in STR of UFD Data Block with ptr to this RIB | |
| RIBFLR | Relative block # of 1st block in RIB | |
| RIBXRA* | Address of next RIB in chain | |
| RIBTIM | Creation date and time in new format | |
| RIBLAD | Last accounting date (UFD only) | |
| RIBDED | Directory expiration date (UFD only) | |
| RIBACT | AOBJN pointer for accounting string | |
| | Retrieval Pointers (details on following pages) | |
| RIBACS | Account string (pointer in RIBACT) | |
| RIBCOD | 0 | 777777 |
| RIBSLF | 0 | Self block number |

103.1 RIBEXT -- File Extension

```
+-------------------------------------------------------+
|      Extension          |       |   Access Date       |
+-------------------------------------------------------+
0                       17 18   23 24                  35
```

| Bits | Contents |
|---|---|
| 0-17 | File extension in SIXBIT |
| 24-35 | Last access date |

103.2 RIBPRV -- Access Privilege

```
+-------------------------------------------------------+
|  Access      |Mode | Creation time  |  Creation date  |
+-------------------------------------------------------+
0             8    12                23                35
```

| Bits | Contents |
|---|---|
| 0-8 | Access code |
| 9-12 | Data mode of file |
| 13-23 | File creation time |
| 24-35 | Low-order twelve bits of file creation date |

Access Codes:

| Bits | Meaning |
|---|---|
| 0-2 | Apply to any job with matching programmer number. |
| 3-5 | Apply to any job with matching project number. |
| 6-8 | Apply to all other jobs. |

Privilege Codes for User Files:

| Code | Highest Privileges |
|---|---|
| 7 | None (but owner may read) |
| 6 | Execute-only (but owner may read) |
| 5 | Read |
| 4 | Append (allocate, deallocate) |
| 3 | Update |
| 2 | Write (supersede, truncate) |
| 1 | Rename (change attributes) |
| 0 | Change privileges |

Privilege Codes for Directories:

| Code | Privilege |
|---|---|
| 4 | Allow LOOKUPs in this directory |
| 2 | Allow creates |
| 1 | Allow directory to be read as a data file |

Any combination of these bits may be set.

Notes:

1. RIBPRV is maintained in ACYPRV while the file is being accessed.

2. If the monitor is assembled for File Daemon, and FILDAE is running, an owner privilege greater than or equal to 4 invokes FILDAE, giving extended access protection modes. See the TOPS-10 Monitor Calls Manual for details.

### 103.3 RIBSTS -- Status Word

```
+--------------------------------------+
|             Status bits              |
+--------------------------------------+
```

Left half bits apply to the UFD, right half bits apply to this file.

| Bits | Label | Meaning |
|---|---|---|
| 0 | RIPLOG | (LH only) User logged in |
| 1 | RIPCHG | Set to 1 by FILSER if any file is written or renamed |
| 9,27 | RIPSCE | File has had checksum error |
| 7,24 | RIPABU | Always BACKUP this UFD/file |
| 10,28 | RIPHWE | File has had hard write error |
| 11,29 | RIPHRE | File has had hard read error |
| 14,32 | RIPBFA | File found bad by FAILSAFE during restore operation |
| 15,33 | RIPCHR | File closed after crash |
| 17,35 | RIPBDA | File found bad by assessment CUSP |
| 18 | RIPDIR | This is a directory |
| 19 | RIPNDL | This file cannot be deleted by any user |
| 20 | RIPDMP | Dump file not yet processed by CRSCPY |
| 21 | RIPNFS | Not to be dumped by FAILSAFE |
| 22 | RIPABC | Always bad checksum (SWAP.SYS, SAT.SYS) |
| 23 | RIPCBS | Compress bit set |
| 25 | RIPNQC | This file is not checked for quota |
| 31 | RIPPAL | Preallocated file |
| 32 | RIPRMS | This is an RMS file |

103.4 RIBELB -- Data Error Location

```
+--------------------------------------------------------+
|  Error bits  |       LBN where bad region starts        |
+--------------------------------------------------------+
```

Bits 0 through 8 give the type of error that occurred. Bits 9 through 35 give the logical block number on the unit where the error occurred.

| Bit | Meaning |
|---|---|
| 3 | Error other than listed below |
| 4 | Data error (parity or ECC hard) |
| 5 | Search or header compare error |

103.5 File-Specific Definitions

The following words in the RIB are defined differently for file header blocks. The symbols for the UFD word and the file header word are:

| UFD | File | Contents for File Header |
|---|---|---|
| RIBQTF | .RBTYP | File type and flags |
| RIBQTO | .RBBSZ | Byte sizes |
| RIBQTR | .RBRSZ | Record and block sizes |
| RIBUSD | .RBFFB | FFB and ACW fields. |

The file header words are each described in more detail in the following sections.

103.5.1 .RBTYP -- File Type

.RBTYP contains the following:

| Bits | Symbol | Meaning |
|---|---|---|
| 0 | RB.DEC | File formatted by DIGITAL |
| 1 | RB.RMS | RMS-10 formatted file |
| 2 | RB.MCY | MACY11 format |
| 3 | RB.CTG | File is contiguously allocated |
| 4 | RB.WSB | Records do not cross section boundaries |
| 5-14 | | Reserved for use by DIGITAL. |
| 15-17 | RB.CRY | File is encrypted (field contains code indicating the type of encryption algorithm) |
| 18-23 | RB.DTY | File data type (codes listed below) |

Code Symbol Meaning

| Code | Symbol | Meaning |
|---|---|---|
| 0 | .RBDUN | Undefined (none specified) |
| 1 | .RBDAS | ASCII character data |
| 2 | .RBDBI | Binary (image) data |
| 60-77 | | Reserved for customer definition |

24-29 RB.DTO Data OTS type (codes listed below)

| Code | Symbol | Meaning |
|---|---|---|
| 0 | .RBOUN | Undefined (none specified) |
| 1 | .RBOCO | COBOL |
| 2 | .RBOFO | FORTRAN |
| 3 | .RBOMS | Mail file (MS program) |
| 60-77 | | Reserved for customer definition |

30-35 RB.DCC Data carriage-control formatting (codes listed below)

| Code | Symbol | Meaning |
|---|---|---|
| 0 | RB.CUN | Undefined (none specified) |
| 1 | RB.CFO | FORTRAN carriage control |
| 2 | RB.CAS | ANSI space carriage control |
| 60-77 | | Reserved for customer definition. |

### 103.5.2 .RBBSZ -- Byte Sizes

.RBBSZ contains the following:

| Bits | Symbol | Meaning |
|---|---|---|
| 0-7 | RB.BSZ | Logical data byte size |
| 8-15 | RB.FSZ | Physical data frame size |
| 16-23 | RB.HSZ | Fixed header size |
| 24-39 | RB.RFM | Record format (for variable-length records (codes are listed below) |

| Code | Symbol | Meaning |
|---|---|---|
| 0 | .RBRUN | Undefined (no specified record structure) |
| 1 | .RBRFX | Fixed-length records |

| | | |
|---|---|---|
| 2 | .RBRVR | Variable-length records |
| 3 | .RBRVF | Variable-length records with fixed-length header |
| 4 | .RBRSP | Spanned records (ANSI labelled tapes) |
| 60-77 | | Reserved for customer definition |

| | | |
|---|---|---|
| 30-35 | RB.RFO | Record organization (codes are listed below) |

| Code | Symbol | Meaning |
|---|---|---|
| 0 | .RBRSQ | Sequential records |
| 1 | .RBRRL | Relative records |
| 2 | .RBRID | Indexed records |
| 3 | .RBRHS | Hashed records |
| 60-77 | | Reserved for customer definition |

### 103.5.3 .RBRSZ -- Record Sizes

.RBRSZ contains:

The left half (RB.RSZ) specifies the record size, in bytes.

The right half (RB.BLS) specifies the block size, in bytes.

### 103.5.4 .RBFFD -- FFB And ACW

.RBFFB contains:

The left half (RB.FFB) specifies the first free byte within the last block of the file.

The right half (RB.ACW) is the application-specific field.

## 103.6 RIBXRA -- Next RIB

(Same format as DEVRIB)

| Bits | Byte Pointer | Contents |
|---|---|---|
| 0 | | Set to 1 |
| 1-8 | DEYRBC | Number of RIB (first extended RIB is 1, and so forth) |

| | | |
|---|---|---|
| 9-12 | DEYRBU | Logical unit on which extended RIB exists |
| 13-35 | DEYRPA | Cluster address on unit of extended RIB |

## 103.7 Retrieval Pointer Format

```
+--------------------------------------------------------+
| Cluster count  |      Checksum      |  Cluster addr    |
+--------------------------------------------------------+
```

Widths of these fields are defined symbolically, and may be different for each file structure. Byte pointer is defined in the HOM block, kept in Structure Data Block while the structure is mounted.

| Field | Byte Pointer |
|---|---|
| Cluster Count | STYCNP |
| Checksum | STYCKP |
| Cluster Address | STYCLP (23 bits maximum) |

If cluster count = 0, the word actually is one of the following:

- o Pointer to new unit, if bit 18 = 1. Bits 19-35 specify logical number within file structure.
- o EOF flag, if whole word is zero.
- o Cluster count is number of clusters in group.

104 SAB -- STORAGE ALLOCATION BLOCK

Description: Table describing allocation of clusters of blocks for a file structure.

Each allocation bit represents a corresponding cluster of physical blocks within the structure.

(See also SAT.SYS and SPT.)

Defined in: COMMOD

Used by: FILFND, FILIO, ONCMOD, REFSTR

```
         +-------------------------------------------------------+
SABRNG   |   Core adr of next        |     No free clusters      | SABTAL
         |   SAB for unit            |     in this SAT           |
         |-------------------------------------------------------|
SABSCN   |   No words in             |     Adr to start scan     |
         |   SAT Buffer              |     for free clusters     |
         |-------------------------------------------------------|
SABNDX*  |  A     | B | NDX   |            First cluster         | SABCLA
         |        |   |       |                                  | SABFIR
         |-------------------------------------------------------|
SABHOL   |           Number of blocks in largest hole            |
         |-------------------------------------------------------|
SABBIT   |                                                       |
         |                   One data block of                   |
         |                      SAT.SYS                          |
         |                                                       |
         +-------------------------------------------------------+
```

* Details on following page.

SABNDX -- NDX and CLA

```
+---------------------------------------------------------+
|  A   |  B   |        NDX        |     First cluster     |
+---------------------------------------------------------+
   0       1    2                12   13                  35
```

| Word Label | Bits | Mask Symbol | Contents |
|---|---|---|---|
| SABFIR | 0 | SAPDIF | Set if table in core different from disk. |
| SABFIR | 1 | SAPBAD | Set if SAT block is on a bad block in disk. |
| SABNDX | 2-12 | SAYNDX | Index value for SPT entry representing this SAT. |
| SABCLA | 13-35 | SAYCLA | Cluster address within unit of first cluster represented in this SAT. |

Notes:

1. The NDX and CLA fields are related by the formula.

$$CLA = \begin{array}{l}\text{Number}\\ \text{Cluster}\\ \text{per SAT}\end{array} \times (NDX)$$

2. Bits 2-35 are set -1 when the file structure is created to force SAT to be read into SAB.

3. Under KL-paging, the SAT is usually in Section 7.

## 105 SAT.SYS -- CLUSTER ALLOCATION FILE

Description: Disk file describing the allocation of all clusters of blocks on the file structure.

Contains one bit for each cluster of the file structure.

Bits are in the same order as the clusters which they represent.

| | |
|---|---|
| Cluster 0-35 | One bit for each cluster (clusters 0 through 35) |
| 36-71 | Clusters 36 through 71 |
| . . . | And so forth... |

Notes:

1. SAT blocks are always on the same unit as the clusters that they represent.

2. If more than one SAT block is needed for a single physical unit, each block will be near the clusters that it represents. Hence, only the first block in each group (1 cluster) of SAT.SYS contains data.

3. Programs should not look at bits corresponding to nonexistent clusters. They may or may not be set. (Function of hardware sector length.)

4. Each SAT block has a corresponding entry in the Storage Allocation Pointer Table (SPT) for that unit.

5. The unused bits in the final word of each SAT must be set to 1, as the monitor depends on this condition when searching for holes.

## 106 SB -- SYSTEM BLOCK

Description: Contains information describing a specific node in the CI network. System blocks are created by KLPSER when a START datagram is received from a previously unknown CI node.

Defined in: SCAPRM

Used by: KLPSER, MSCCOM, RAXKON, SCASER, SCSUUO

See also: CB, PB, PCB

```
Symbol   Map
        +--------------------------------------------------------+
.SBDPN  |        Destination port number (CI node number)        |
        |--------------------------------------------------------|
.SBIDX  |        This system's System Block index                |
        |--------------------------------------------------------|
.SBPIN  |  Index of first Path Block,,no. of Path Blocks         |
        |--------------------------------------------------------|
.SBDSA  /        Destination system address (2 words)            /
        |--------------------------------------------------------|
.SBMMS  |  Maximum message and datagram length values            |
        |--------------------------------------------------------|
.SBDST  |        Destination software type                       |
        |--------------------------------------------------------|
.SBDSV  |        Destination software version                    |
        |--------------------------------------------------------|
.SBDSE  /        Destination software edit no. (2 words)         |
        |--------------------------------------------------------|
.SBDHT  |        Destination hardware type                       |
        |--------------------------------------------------------|
.SBDHV  /        Destination hardware version (3 words           /
        |--------------------------------------------------------|
.SBNNM  /        Destination node name (2 words)                 /
        |--------------------------------------------------------|
.SBDTD  /        Destination time of day from Start Packet       /
        /                 (2 words)                              /
        +--------------------------------------------------------+
```

The last portion of the System Block (from .SBDSA to .SBDTD) is used to store the BLT data from the Start Datagram.

## 107 SCHEDULER SCAN TABLES

Description: Used by the system scheduler when selecting a job to run.

Defined in: COMMON

Used by: SCHED1

See also: Queue Transfer Table

SSCAN -- Used by policy CPU for selecting a job to run:

```
        +---------------------------------------+
SSCAN   |      -HPQn         |    IQFOR         |
        |      -HPQ1         |    IQFOR         |
        |      -PQ1          |    IQFOR         |
        |      -PQ2          |    IRRFOR (RR)   |
        |      -PQ2          |    ISSFOR (class)|
        |      -PQ2          |    IBBFOR (class)|
        |                    |                  |
        +---------------------------------------+
```

SSCAN1 -- Used by second processor for selecting a job to run:

```
        +---------------------------------------+
SSCAN   |      -HPQn         |    IQFOR         |
        |      -HPQ1         |    IQFOR         |
        |      -PQ2          |    IRRFOR (RR)   |
        |      -PQ2          |    ISSFOR (class)|
        |      -PQ1          |    IQFOR         |
        |      -PQ2          |    IBBFOR (class)|
        |                    |                  |
        +---------------------------------------+
```

SQSCAN -- Used by SQFOR code:

```
              Subqueue #        Ptrs to Quota Left
        +-------------------------------------+
SQSCAN  |       -SQ0         |    CLSQTA      |
        |       -SQ1         |    CLSQTA+1    |
        |       -SQ2         |    CLSQTA+2    |
        |                    |                |
        |       -SQn         |    CLSQTA+n    |
        +-------------------------------------+
```

SQFOR scans subqueues forward according to SQSCAN table if RRFLAG = 0, (count of classes with non-zero quotas), otherwise it scans PQ2 with QFOR routine.

DCSCAN is used by the scheduler for selecting jobs for IPCT decrementing. This table is not processed by QSCAN but by specific code in the IPCT maintenance routines.

DCSCAN:

```
        +--------------------------------+
        |                           -EWQ |
        |--------------------------------|
        |                          -SLPQ |
        |--------------------------------|
        |                           -PQ2 |
        |--------------------------------|
        |                           -PQ1 |
        |--------------------------------|
        |                          -HPQn |
        +--------------------------------+
```

Notes:

The items flagged by (RR) are used in a system built with the round-robin scheduler (FTNSCHED=0). Items flagged by (class) are used in a system built with the class scheduler (FTNSCHED=1).

## 108 SPT -- STORAGE ALLOCATION POINTER TABLE

Description: Contains pointers to all SAT blocks for a unit, whether in core or not. There is one entry for each SAT block on a unit, in order of the cluster address which they represent, and zero entry indicates end of table.

Defined in: COMMOD

Used by: FILFND, ONCMOD

```
         +-------------------------------------------------------+
SPTFIR   |     Free cluster cnt       |    Cluster adr in unit   |
         |-------------------------------------------------------|
         |     Free cluster cnt       |    Cluster adr in unit   |
         |-------------------------------------------------------|
                                    . . .
         |-------------------------------------------------------|
         |            0               |            0             |
         +-------------------------------------------------------+
          0                         12 13                      35
```

| Bits | Byte Pointer | Contents |
|---|---|---|
| 0-12 | SPYTAL | Number free clusters represented in this SAT Block. |
| 13-35 | SPYCLA | Cluster address within unit for this SAT Block. |

Notes:

- o Each Unit Data Block, UDB, contains a pointer to its SPT.
- o Each SAT block that is in core is in a SAB, Storage Allocation Block. The SAB contains the index value for the entry in this table corresponding to the SAT block that it currently contains.
- o The last word in the SPT table will always be zero.
- o Under KL-paging, the SPT is usually in Section 2.

109 STR -- FILE STRUCTURE DATA BLOCK

Description: Contains descriptive information about a file structure.

There is a Structure Data Block for each structure defined in the system.

Defined in: COMMOD

Used by: COMMON, FILFND, FILIO, FILUUO, IPCSER, ONCMOD

```
        +--------------------------------------------------------+
STRNAM  |                 SIXBIT structure name                  |
        |--------------------------------------------------------|
STRSYS  |   Next STR in system        |      This STR number     | STRFSN
        |--------------------------------------------------------|
STRUNI  |  First UDB for this STR     |      K for CRASH.EXE     | STRK4C
        |--------------------------------------------------------|
STRREF  |   Nonzero if STR needs      | No. of units in this STR | STRUNM
        |      to be refreshed        |                          |
        |--------------------------------------------------------|
STRHGH  |            Highest logical blk in structure            |
        |--------------------------------------------------------|
STRSIZ  |              Size of STR in 128 word blocks            |
        |--------------------------------------------------------|
STRGAR  |          Limit on total blocks reserved in STR         |
        |--------------------------------------------------------|
STRRES  |          No. of reserved blocks remaining free         |
        |--------------------------------------------------------|
STRALT  | Alter number for this structure                        |
        |--------------------------------------------------------|
STRTAL  | No. first-come-first-serve free blocks on structre     |
        |--------------------------------------------------------|
STROVR  |               Overdraw limit - per user                |
        |--------------------------------------------------------|
STRMNT  |               Mount count for this STR                 |
        |--------------------------------------------------------|
STRPT1  |              First retrieval ptr for MFD               |
        |--------------------------------------------------------|
STRTRY* | TRY | RETRY | RECAL |     |       |  X  | UNIT|   | X | STRUN1
        |--------------------------------------------------------|
STRBPU  |            (Maximum) No. of blocks per unit            |
        |--------------------------------------------------------|
STRBSC  |     No. of blocks per       |  (Maximum) No. of super- |
        |       supercluster          |     clusters per unit    |
        |--------------------------------------------------------|
```

| | | |
|---|---|---|
| STRJOB | Access | Job # having access or 0 |
| STYCNP | Byte ptr to RIB cluster count in AC T2 | |
| STYCKP | Byte ptr to RIB checksum field in AC T2 | |
| STYCLP | Byte ptr for cluster adr in AC T2 | |
| STRPPN | PPN of the structure owner | |
| STRSDL | Position of STR in system dump list (-1= not in list) | |
| STRCRS | LBN of RIB for CRASH.EXE | |

* Details following.

Notes:

1. All STR Data Blocks are set up by the ONCE-Only code, according to information found in the Home Blocks. No information pertaining to structures is coded into the monitor.

2. STRALT is incremented each time a SAT block is written for this structure.

3. Access is -1 if the job in right half is the only job with the structure mounted, and it is not single-access structure.

STRTRY Byte Definitions

```
+-------------------------------------------------------+
| TRY   |RETRY |RECAL |       |  X   | UNIT |        |X|
+-------------------------------------------------------+
0      6      12     18     27           31      34 35
```

| Word Label | Bits | Byte Label | Content |
|---|---|---|---|
| STRTRY | 0-5 | STYTRY | Number of times to retry before error considered hard. |
| STRTRY | 6-11 | STYSER | Number of times to retry on search and data errors. |
| STRTRY | 12-17 | STYRCL | Number of recalibrates for search and data errors. |
| STR1PT | 27 | STP1PT | Set if STRPT1 is only retrieval pointer for MFD. |
| STRUN1 | 28-31 | STYUN1 | Logical unit number within this file structure where MFD begins. |
| STRPVS | 35 | STYPVS | Non-zero if this is a private structure. |

## 110 SWPLST -- SWAPPING LIST TABLE

Description: Table used by the VM swapper in conjunction with MEMTAB to keep track of jobs being swapped or having paging I/O in progress.

Defined in: COMMON

Used by: SCHED1, SWPSER, VMSER

```
 0                 13 14              26 27           35
+-------------------------------------------------------+
|         Bits         | Start phys. pg. | Number of pages|
|-------------------------------------------------------|
|                      |                 |                |
|-------------------------------------------------------|
                            . . .
```

Notes:

| Bit | Symbol | Meaning |
|---|---|---|
| 0 | SL.FRG | Fragmented entry |
| 1 | SL.DIO | Direction of I/O (1 = out) |
| 2 | SL.SIO | Swapping/paging (1 = swapping) |
| 3 | SL.IOP | I/O in progress |
| 4 | SL.IOD | I/O done (this swap list entry is done) |
| 5 | SL.IPC | On if an IPCF page |
| 6 | SL.DFM | Don't find me (used to keep FNDSLE from finding this entry) |
| 11 | SL.CHK | Swapping checksum error |
| 12 | SL.ERR | I/O error (IODTER, IODERR, or IOIMPM) |
| 13 | SL.CHN | Channel error (IOCHMP or IOCHNX) |

For a contiguous entry, the data is in the following fields:

Bits 14-26 — Contains the starting physical page number (used as an index into MEMTAB).

Bits 27-35 — Contains the number of pages.

For a fragmented entry, the following field contains:

Bits 18-35 — Contains the address of the fragment table. The fragment table is linked the same way the JBTSWP entry is, but the entries are as above.

## 111 SW2LST -- SECONDARY SWPLST

Description: The original SWPLST entry is stored here for cleanup purposes, since SWPLST is modified while I/O is progressing.

Defined in: COMMON

Used by: VMSER

## 112 SW3LST -- THIRD SWPLST

Description: The right half of this table is used to store the job number of the job being swapped or doing paging I/O, and left half holds the contents of SWPOUT at the time the original related SWPLST was created.

Defined in: COMMON

Used by: SCHED1, SWPSER, VMSER

```
        +------------------------------------+
SW3LST  |                 .                  |
        |                 .                  |
        |------------------------------------|
        |     (SWPOUT)    |    Job Number    |
        |------------------------------------|
        /              .  .  .               /
```

## 113 SWPTAB -- SWAPPING TABLE

Description: Contains addresses of the Unit Data Blocks of all units available for swapping. This table specifies the active swapping list.

Defined in: COMMOD

Used by: CPNSER, FILFND, FILIO, ONCMOD, SEGCON, SWPSER, SYSINI, VMSER

```
+--------------------------------------+
|                UDB 1                 |
|--------------------------------------|
|                UDB 2                 |
|--------------------------------------|
|                UDB 3                 |
|--------------------------------------|
|                                      |
|                                      |
|--------------------------------------|
|                UDB n                 |
+--------------------------------------+
```

## 114 TABSTR -- STRUCTURE TABLE

Description: Contains addresses of all File Structure Data Blocks in the system. File structures are in order of access speed, fastest to slowest.

Index value for each entry is the File Structure Number, FSN.

Defined in: COMMOD

Used by: COMMON, CPNSER, FILFND, FILIO, FILUUO, ONCMOD, SYSINI

```
         +--------------------------------------+
STRAOB:  |  -.SLMAX         |     .FSMIN        |
         |--------------------------------------|
STRTAB:  |                STR 1                 |
         |--------------------------------------|
         |                STR 2                 |
         |--------------------------------------|
         |                                      |
         |                                      |
         |--------------------------------------|
         |                STR n                 |
         +--------------------------------------+
```

Notes:

1. Number of entries is .SLMAX. This value cannot exceed 36.

2. The first word is an AOBJN pointer to the rest of the table.

## 115 TRANSFER TABLES

Description: These tables are used in the requeuing process to determine the destination queue and quantum runtime for jobs being requeued by the scheduler.

Defined in: SCHED1

See also: JBTSTS, QBITS

Although these tables are defined by macros in COMMON they are represented here in their expanded form:

```
EQFIX== 400000,,QFIX        ;Specifies fix transfer to end of queue
EQLNKX==400000,,QLNKZ       ;Specifies requeing and quantum run
                            ; time based on current queue and
                            ; job size

QNULW:  EXP      EQFIX      ;Null queue
        XWD      -1,-NULQ

QSTOP::
QSTOPW: EXP EQFIX           ;Stop queue
        XWD -1,-STOPQ

QJDCW:  EXP EQFIX           ;DAEMON wait queue
        XWD -1,-JDCQ
QCMW::  EXP EQFIX           ;Command wait queue
        XWD -1,-CMQ

QTSW:                       ;TTY I/O Wait satisfied
QRNW:   EXP EQFIX           ;Jobs just became runnable
        XWD QADTAB##,-PQ1

QRNW1:  EXP EQFIX           ;Back of QP1, no quantum change
        XWD  -1,-PQ1

QRNW2:  EXP EQFIX           ;Back of PQ2, no quantum change
        XWD  -1,-PQ1

QTIOWW: EXP EQFIX           ;TTY I/O Wait satisfied
        XWD -1,TIOWQ

QSLPW:  EXP  EQFIX          ;Sleep for greater than/equal to
        XWD  -1,-SLPQ       ;1 second

QEWW:   EXP EQFIX           ;Event wait
        XWD -1,-EWQ

QTIME:  EXP EQLNKZ          ;When quantum time exceeded
        XWD 0,QRQTBL
```

## 116 TTFCOM -- FORCED COMMANDS TABLE

Description: Allows SCNSER to force a specified command to be executed for a job without having to put the command into the terminal buffer.

The TTFCOM table contains one entry, in SIXBIT format, for each command that SCNSER might want to force. Each entry is conditionally assembled.

Each symbol in TTFCOM is associated with a value that equals its offset within TTFCOM. The LDB DDB contains this offset for forced commands.

Defined in: SCNSER

Used in: CLOCK1, COMCON, NETMCR, NETSER, NETVTM, ONCE, SYSINI, UUOCON

| Label | Content | Command |
|---|---|---|
| TTFCXC | HALT | Control-C |
| TTFCXD | .BYE | Dataset disconnect |
| TTFCXH | .HELLO | Dataset connect |
| TTFCXR | .RESTA | System restart |
| TTFCXK | KJOB | Kill job |
| TTFCXI | INITIA | Call initializing CUSP |
| TTFCXJ | .FCONT | Forced continue |
| TTFCXT | .TYPE | Retype line |
| TTFCXW | USESTA | <CTRL/T> |
| TTFCXL | .NETLD | Network reload |
| TTFCXS | .HALT | <CTRL/C> with no trapping |
| TTFCXB | .BPT | <CTRL/D> breakpoint |
| TTFCXX | CTEST | (For patching) |

## 117 TERMINAL CHUNKS

Description: Used to hold characters that need to be typed on a terminal, or characters received from a terminal and not yet read by a program. The first word of each chunk is a link word containing the addresses of previous and following chunks. The remainder of the terminal chunk is 3 words consisting of 3 12-bit bytes. Each byte contains an ASCII character or null.

Buffers are set up dynamically, as needed, from a pool of monitor free core reserved for that purpose.

The association between a buffer and a line depends on pointers in the Line Data Block.

Buffers are built from four word "chunks," which are linked together as necessary.

Defined in: SCNSER

See also: LDB

```
+-------------------------------------+
| Prev chunk addr  | Next chunk addr  |
|-------------------------------------|
|  Byte 0 |    Byte 1   |   Byte 2    |
|-------------------------------------|
|  Byte 3 |    Byte 4   |   Byte 5    |
|-------------------------------------|
|  Byte 6 |    Byte 7   |   Byte 8    |
+-------------------------------------+
```

Notes:

1. The bits in each byte are defined as:

| Bits | Meaning |
|---|---|
| 9 | Image mode |
| 10 | Character has been echoed |
| 11 | Character has been logically deleted from the character stream. |
| 12 | Current byte is a special function character (meta-character), rather than a normal data character. |

2. If there is not another chunk in a given direction, the corresponding linkage will be zero.

3. All chunks that are not part of a buffer are linked together to form the "free list". The word TTFTAK points to the oldest chunk in the free list, and word TTFPUT points to the newest chunk in

the free list.

4. The size of an individual buffer is limited by program action. An output buffer cannot exceed 80 characters. If an input buffer exceeds 172 characters (value of symbol TTIWRN) each receive interrupt will force the output of an XOFF. If an input buffer contains 300 characters (symbol TTIMAX) no additional characters will be accepted from that line. A bell will be substituted for the echo of a lost character.

118 TTUUOT -- TTCALL DISPATCH TABLE

Description: This table contains pre-check and dispatch information for TTCALL UUOs. The bits in the left half are checked before dispatching. There is one entry for each TTCALL UUO.

Defined in: SCNSER

```
         +-----------------------------------------------------+
TTUUOT:  | Check bits  |              |   Dispatch address    | TTCALL 0
         |-----------------------------------------------------|
         |             |              |                       | TTCALL 1
         |-----------------------------------------------------|
         |             |              |                       | TTCALL 2
         |-----------------------------------------------------|
         |             |              |                       |
         |             |              |                       |
         |-----------------------------------------------------|
         |             |              |   Dispatch address    | TTCALL 17
         +-----------------------------------------------------+
           0          4              18                     35
```

| Bit | Label | Meaning |
|---|---|---|
| 0 | TC.ADC | This function must be address-checked |
| 1 | TC.USR | This function must be at user level, else return |
| 2 | TC.USW | This function must be at user level, else wait |
| 3 | TC.ATW | This function must be attached, else wait |
| 4 | TC.ATR | This function must be attached, else return |
| 5 | TC.ECS | This function releases the previous input line, causing a subsequent RESCAN to fail |

## 119 TTYTAB -- TTY TABLE

Description: One entry per job, indexed by job number. This is the table of controlling (attached) terminals for each job.

Defined in: COMMON

Used by: CLOCK1, COMCON, COMDEV, CPNSER, ERRCON, FILIO, IPCSER, PSISER, PTYSER, SCNSER, UUOCON

```
         +--------------------------------------------------------+
TTYTAB:  |                          |                             | Job 0
         |--------------------------------------------------------|
         |                          |        DDB address          | Job 1
         |--------------------------------------------------------|
         |                          |        DDB address          | Job 2
         |--------------------------------------------------------|
         |                          |        DDB address          | Job 3
         |--------------------------------------------------------|
         |                          |                             |
         |                          |                             |
         |                          |                             |
         |                          |                             |
         |                          |                             |
         |--------------------------------------------------------|
         |                          |        DDB address          | Job n
         +--------------------------------------------------------+
```

A zero entry indicates no attached terminal, otherwise, right half is controlling DDB for the job. There is always a TTY DDB for every job, even though no TTY need be attached. Thus, UUOs look through TTYTAB(n) for a DDB because UUOs come from jobs.

## 120 TYPTAB -- DEVICE TYPES TABLE

Description: Contains three letter generic device type prefix in SIXBIT format for all possible devices in the system.

Used by UUOCON subroutine which finds a DDB given its Universal Device Index.

Defined in: UUOCON

Used by: UUOCON

```
+--------------------------------------+
|              SIXBIT/DSK/             |
|--------------------------------------|
|              SIXBIT/DTA/             |
|--------------------------------------|
|              SIXBIT/MTA/             |
|--------------------------------------|
/                 . . .                /
+--------------------------------------+
```

121 UCLJMP -- CALL AND CALLI UUO DISPATCH TABLE

Description: Contains dispatch addresses for CALL and CALLI UUOs. There is one entry for each two routines. Left half contains address for even numbered routines; right half for odd numbered routines

Indexed by one half the CALLI argument.

For CALL UUOs, a table lookup is done in UCLTAB to get routine number; for CALLI UUOs the number is supplied directly. One half of this routine number is used as the table index. The left half is taken for even numbers; right half for odd. UUOCON then dispatches to that address.

Defined in: UUOCON

Used by: UUOCON

```
+-------------------------------------+
/                . . .                /
|-------------------------------------|
| Customer adr -4  | Customer adr -3  |
|                                     |
| Customer adr -2  | Customer adr -1  |
|                                     |
|-------------------------------------|
|   DEC    adr 0   |  DEC     adr 1   |
|-------------------------------------|
|   DEC    adr 2   |  DEC     adr 2   |
|-------------------------------------|
/                . . .                /
+-------------------------------------+
```

Notes:

UCLJMP has entries corresponding to entries in UCLTAB. Table entries may be added in the negative direction by customers, and in the positive direction by DIGITAL. Once a table entry is established, its position can never be changed without invalidating those programs that use the corresponding CALLI.

122 UCLTAB -- CALL UUO NAMES TABLE

Description: Contains names of the CALL UUOs. There is one entry for each CALL function.

Indexed by corresponding CALLI value. Customer defined CALLs have negative index values; DEC CALLs have positive values. Table entries are SIXBIT expressions of the CALL names. There are corresponding dispatch addresses in the UCLJMP table.

```
         +-------------------------------------+
CCLTAB:  |          Customer CALL -m           |
         |-------------------------------------|
         |          Customer CALL -2           |
         |-------------------------------------|
         |          Customer CALL -1           |
         |-------------------------------------|
UCLTAB:  |             DEC CALL 0              |
         |-------------------------------------|
         |             DEC CALL 1              |
         |-------------------------------------|
         |             DEC CALL 2              |
         |-------------------------------------|
         |             DEC CALL n              |
         +-------------------------------------+
```

Notes:

1. Customers may extend the table in the negative direction with as many of their own CALLs as desired.

2. The value specified in a CALLI UUO corresponds to the position of the CALL UUO name in this table. Hence, once an entry is established, its position in the table can never be changed without invalidating any existing programs that use that CALLI.

3. All CALLs above CALLI AC,55 do not have a corresponding CALL with a SIXBIT argument. In the future, only CALLIs will be added by DIGITAL.

123 UDB -- UNIT DATA BLOCK

Description: One UDB for each physical disk drive on the system (two if the drive is dual ported). Unit Data Blocks are generated dynamically by AUTCON when the system is started and when units come on line.

Defined in: COMMOD

Used by: COMMON, CPNSER, DPXKON, ERRCON, FHXKON, FILFND, FILIO, FILUUO, FSXKON, KLSER, ONCMOD, RPXKON, SYSINI, VMSER

```
        +-------------------------------------------------+
 UNINAM | SIXBIT physical unit name                       |
        |-------------------------------------------------|
 UNILOG | SIXBIT logical name within structure (HOMLOG)   |
        |-------------------------------------------------|
 UNIHID | SIXBIT Home Block ID name (HOMHID)              |
        |-------------------------------------------------|
*UNISYS | Next UDB in system    |    SIC     |       LUN  |
        |-------------------------------------------------|
 UNISTR | Next UDB for STR      |    STR Data Block       |
        |-------------------------------------------------|
 UNICHN | Next UDB on channel   |    CHN Data Block       |
        |-------------------------------------------------|
 UNIKON | Next UDB on controller|    KON Data Block       |
        |-------------------------------------------------|
*UNISWP | Next UDB for swapping |         CFS             |
        |-------------------------------------------------|
 UNIHCT | Hard disk error statistics                      |
        |-------------------------------------------------|
 UNISCT | Soft and hard error statistics                  |
        |-------------------------------------------------|
 UNIMCT | Monitor detected error statistics               |
        |-------------------------------------------------|
 UNIERR | Device CONI at time of last hard error          |
        |-------------------------------------------------|
 UNISOF | CONI at time of last error before recovery      |
        |-------------------------------------------------|
 UNIHBN | Last logical block number on hard or soft error |
        |-------------------------------------------------|
 UNIBRC | Number of buffered mode blocks read on unit     |
        |-------------------------------------------------|
 UNIBWC | Number of buffered mode blocks written on unit  |
        |-------------------------------------------------|
```

| | | | |
|---|---|---|---|
| UNIDRC | Number of dump mode blocks read on unit | | |
| UNIDWC | Number of dump mode blocks written on unit | | |
| UNIMRC | Number of monitor blocks read on unit | | |
| UNIMWC | Number of monitor blocks written on unit | | |
| UNIICT | Number of blocks swapped in from unit | | |
| UNIOCT | Number of blocks swapped out to unit | | |
| UNIMSC | Number of monitor + swap seeks on this unit | | |
| UNIUSC | Number of user mode seeks on this unit | | |
| UNIPCT | Number of positioning failures | Number of soft+hard seek incomplete failures | |
| UNIFKS | Free K for swapping on this unit | | |
| UNISDI | Last DATAI status before recovery attempted | | |
| UNIHDI | Last DATAI status after first recovery failed | | |
| UNIECT | # times error status returned for last operation | | |
| UNIHNG | Hung timeout counters | | |
| *UNISTS | Status code for unit | | |
| *UNICCT | Section# for swapping SAT | BCT | No. of channel termination errors |
| UNIHOM | 1st home block address | redundant home blk adr | |
| UNIQUE | addr. of 1st PWQ DDB | Job no. of PWQ DDB | |
| *UNIGRP | # blocks to try for on output | Last disk position (RP20) | |
| UNIBPU | # logical blocks per unit (returned by DSKCHR) | | |
| UNIBPM | # of logical blocks/unit incl. maint. cyls. | | |
| UNIPCI | # of blocks paged in from unit | | |

```
        |----------------------------------------------------|
UNIPCO  | # of blocks paged out to unit                      |
        |----------------------------------------------------|
*UNICHR |    BPC     |    BPT     |    Blocks per cylinder   |
        |----------------------------------------------------|
*UNICPS |    WPS     | |   SPU    |            CPS           | UNIWPS
        |----------------------------------------------------|
UNICYL  | Current physical cylinder number                   |
        |----------------------------------------------------|
UNIBLK  | Logical block number within unit                   |
        |----------------------------------------------------|
UNISAB  | Address of first SAB in ring                       |
        |----------------------------------------------------|
UNITAL  | # of free blocks on unit (reserved + FCFS)         |
        |----------------------------------------------------|
*UNIDES | Unit description bits for DSKCHR                   |
        |----------------------------------------------------|
UNIPTR  | -length swap SAT table| Addr of swap SAT table     |
        |----------------------------------------------------|
UNISLB  | 1st logical block for swapping on unit             |
        |----------------------------------------------------|
UNIXRA  |  # blks read using      |  # blks written using    |
        |  extended ribs          |  extended ribs           |
        |----------------------------------------------------|
UNICDA  | Previous cont. of RH | Addr of active DDB          |
        |----------------------------------------------------|
UNIGEN  | Generation number of UDB                           |
        | (AOSed when unit is dismounted)                    |
        |----------------------------------------------------|
UNIRCV  | # of hung unit retries without success             |
        |----------------------------------------------------|
UNISWA  | addr of current         |  distance to swap block  |
        | SWPLST entry            |                          |
        |----------------------------------------------------|
UNISWD  | Distance to swap cylinder                          |
        |----------------------------------------------------|
UNIQUL  | Length of position wait queue                      |
        |----------------------------------------------------|
UNIBUC  | # of blocks in 10/11 compatibility mode            |
        |----------------------------------------------------|
UNIDIA  | Job # of job shutting| Addr of DDB of job          |
        | down I/O (DIAG. UUO) | shutting down I/O (DIAG.)   |
        |----------------------------------------------------|
*UNIALT |   Bit mask of CPUs      |  Alternate port addr     |
        |----------------------------------------------------|
*UNI2ND | A |  Reserved           |  Alternate port addr     |
        |----------------------------------------------------|
UNISER  |  Drive serial number (double-word)                 |
        |----------------------------------------------------|
```

| | |
|---|---|
| UNITIM | Hung-timer |
| UNIJOB | Previous RH UNIJOB \| Job no. of RH UNICDA |
| *UNIAJB | DA resource status word |
| *UNIDS2 | N \| P \| res. \| KOF \| PUN |
| UNILTM | Universal date/time of lock on structure |
| UNISPT | RH is address of storage allocation pointers (SAT) table |
| UNIPGT | Page quarter turns \| Page turns (No. of times RIB was reread to get new pointers) |
| UNICRC | No. of monitor cache read calls |
| UNICRH | No. of monitor cache read hits |
| UNICWC | No. of monitor cache write calls |
| UNICWH | No. of monitor cache write hits |
| UNICBK | No. of monitor blocks cached for this unit |
| UNIK4S | Word addr of K for swapping on this unit |
| UNILAS | Last command issued to massbus device |
| UNISCR | Contents of control register at first error |
| UNIHCR | Contents of control register at end |
| UNISDR | Contents of data register at first error |
| UNIHDR | Contents of data register at end |
| UNIEBK | Drive registers saved here on error. LH has last error, RH has first error. Last word in block is command which cause the error. (0-16 words, determined by X'ERNO) |

* Indicates that details are shown on following pages.

123.1 UNISYS -- Next UDB In System

```
+-----------------------------------------------------------+
|   Next UDB in system  |  SIC   |          |      LUN      |
+-----------------------------------------------------------+
 0                    17 18       25         30          35
```

| Word | Bits | Byte | Description |
|---|---|---|---|
| UNISYS | 0-17 | | Core address of next UDB in system. Zero indicates last unit. |
| UNISIC | 18-25 | UNYSIC | Number of SAT blocks in core for this unit. |
| UNILUN | 30-35 | UNYLUN | Logical unit number within file structure for unit |

123.2 UNISWP -- Next UDB For Swapping

```
+---------------------------------------------------------+
|Next UDB for swapping | |  CFS  |  K for swapping        |
+---------------------------------------------------------+
 0                    17    20   22 23                  35
```

| Word | Bits | Byte | Description |
|---|---|---|---|
| UNISWP | 0-17 | | Address of next UDB for swapping |
| UNIFCS | 20-22 | UNYCFS | Swapping class of unit |
| UNIK4S | 23-35 | UNYK4S | Number of K for swapping on the unit |

### 123.3 UNISTS -- Unit Status

```
+------------------------------------------------------+
|                         Code                         |
+------------------------------------------------------+
 0                                                   35
```

| Word | Code | Description |
|---|---|---|
| ICOD | 0 | Unit idle |
| PWCOD | 2 | Position wait |
| PCOD | 3 | Positioning |
| TWCOD | 4 | Waiting to transfer data |
| TCOD | 5 | Transferring data |
| MDACOD | 6 | Unit useable only by MDA (mountable device allocator) |
| OWCOD | 7 | Obsolete |
| OCOD | 10 | Operator wait, no file active |
| OW2COD | 11 | Obsolete |
| O2COD | 12 | Same as OCOD, but no message once a minute |

### 123.4 UNICCT -- Channel Error

```
+------------------------------------------------------+
| Section# |   BCT   | _# of channel termination errors |
+------------------------------------------------------+
 0        8 9      17 18                             35
```

| Word Label | Bits | Byte Pointer | Description |
|---|---|---|---|
| UNISNS | 0-8 | UNYSNS | Section number of swapping SAT table |
| UNIBCT | 9-17 | UNYBCT | Number of slots left in BAT block for unit |
| UNICCT | 18-35 | | Number of channel termination errors on this unit |

### 123.5 UNIGRP -- Output Word

```
+------------------------------------------------------+
| Blocks to try on output  | P | N | R | M |   |  LKP  |
+------------------------------------------------------+
```

| Bits | Symbol | Meaning |
|---|---|---|
| 18 | UNIPWQ | Clock request outstanding of another CPU to process the Position Wait Queue. |

| | | |
|---|---|---|
| 19 | UNINDU | Disk cache needs sweeping |
| 20 | UNIRHP | HOM block reread in progress (CI disks) |
| 21 | UNIMSG | "Offline" message has been given for this minute. |
| 22-26 | | Reserved |
| 27-35 | UNILKP | Last known position of disk (RP20 disks) |

## 123.6 UNICHR -- Block Counts

```
+------------------------------------------------------+
|      BPC      |      BPT      |  Blocks per cylinder  |
+------------------------------------------------------+
 0            8 9           17 18                     35
```

| Word Label | Bits | Byte Pointer | Description |
|---|---|---|---|
| UNIBPC | 0-8 | UNYBPC | Number of blocks per cluster |
| UNIBPT | 9-17 | UNYBPT | Number of blocks per track |
| UNIBPY | 18-35 | UNYBPY | Number of blocks per cylinder |

## 123.7 UNICPS -- SAT Word

```
+------------------------------------------------------+
|      WPS      | |      SPU      |         CPS         |
+------------------------------------------------------+
 0            8 9  10          17 18                  35
```

| Word Label | Bits | Byte Pointer | Description |
|---|---|---|---|
| UNIWPS | 0-8 | UNYWPS | Number of words per SAT block |
| UNISPU | 9-17 | UNYSPU | Number of SAT blocks on the unit |
| UNICPS | 18-35 | UNYCPS | Number of clusters per SAT |

123.8 UNIDES -- Unit Description

```
+-----------------------------------------------------------------------+
| Bits | Status code | Bits | | Channel# | Kon-type | No. | U-type |    |
+-----------------------------------------------------------------------+
 0---6   7---------8  9--16    18------20  21-----26 27-29 30---32 33-35
```

| Word Label | Bit | Mask Symbol | Content |
|---|---|---|---|
| UNIDES | 0 | UNPRHB | Monitor must reread HOME block to ensure pack ID is correct. Set when a pack goes offline. |
| UNIDES | 1 | UNPOFL | Unit is offline. |
| UNIDES | 2 | UNPHWP | Unit is hardware write-protected. |
| UNIDES | 3 | UNPSWP | Unit is in a structure that is software write-protected. |
| UNIDES | 4 | UNPSAF | Unit is in a single access structure. |
| UNIDES | 5 | UNPZMT | Structure mount count is zero. |
| UNIPRF | 6 | UNPPRF | Unit is in a private structure. |
| UNIUST | 7-8 | | Unit status code, as follows: |
| | | UNVPIM | 0 - unit is up and pack mounted. |
| | | UNVPBM | 1 - unit is up and pack is being mounted. |
| | | UNVNPM | 2 - unit is up, but pack is not mounted. |
| | | UNVDWN | 3 - unit is down. |
| UNIDES | 9 | UNPMSB | Unit has more than one SAT block. |
| UNIDES | 10 | UNPNNA | No new access on structure. |
| UNIAWL | 11 | UNPAWL | Structure is write-protected for all jobs. |
| UNIDES | 12 | UNPFUS | Unit got a file-unsafe. |
| UNIWMD | 13 | UNPWMD | Unit waiting for MDA to do something. |
| UNIDES | 14 | UNPALT | Unit is dual-ported. |
| | 15 | UNPUSI | Unit status is inconsistent. |
| | 16 | UNPRSS | Removing swapping space from unit. |
| UNISCN | 18-20 | | Data channel number |
| UNIKTP | 21-26 | UNYKTP | Controller type, as follows: |
| | | TYPDR | 0 - DR (Future drum, if any) |
| | | TYPFH | 1 - FH RC10 (Burroughs disk or Bryant drum) |
| | | TYPDP | 2 - DP RP10 (RP01-03 disks) |
| | | TYPMD | 3 - MD Bryant mass disk |
| | | TYPFS | 4 - FS RH10 with RS04 |
| | | TYPRP | 5 - RP RH10 with RP04-06 |
| | | TYPRN | 6 - RH20/RP20 |
| UNIKNM | 27-29 | UNYKNM | Controller number within type. |
| UNIUTP | 30-32 | UNYUTP | Unit type |
| UNIPUN | 33-35 | UNYPUN | Obsolete (see UNIDS2) |

Notes:

1. This word is returned by the DSKCHR UUO. Those items marked with an asterisk are returned by the DSKCHR UUO.

2. Controller type starts at zero (for example, DPA=0, DPB=1, and so on).

## 123.9 UNIALT -- First Word For Alternate Port

```
+--------------------------------------------------------+
| Bit mask for CPU        |     Alternate port address   |
+--------------------------------------------------------+
```

Notes:

1. The left half of UNIALT contains a bit mask representing the CPU(s) that can access the disk (for CI disks only). The bit is on, if the HSC50 has been initiated on the device. Bit 17 = CPU0, Bit 16 = CPU1, Bit 15 = CPU2,...

2. The right half contains the address of the other port, if the drive is dual- or alternate-ported.

### 123.10 UNI2ND -- Second Word For Alternate Port

```
+-----------------------------------------+
| A |                 |    UDB address    |
+-----------------------------------------+
 0                  17 18                35
```

Entire word is zero if this unit is not being accessed through dual ports. For units that are dual-ported, UNI2ND will be one of the following:

| | |
|---|---|
| XWD 0,UDB-addr-of-alternate | If this is the main port for the unit |
| XWD -1,UDB-addr-of-main | If this is the alternate port |

### 123.11 UNIAJB -- DA Status

1. Is -1 if no DA in progress on this unit
2. Is +n if job n is allocating but no other jobs are waiting.
3. Is n,,n is job n is allocating and others are waiting to use the DA resource.

## 123.12 UNIDS2 -- Lap Plug Number

```
+--------------------------------------------------------+
| N | P | Res. |     KOF     |     KNM     |     PUN      |
+--------------------------------------------------------+
```

| Bit | Symbol | Meaning |
|---|---|---|
| 0 | U2PNRM | Set if the unit has non-removable media. |
| 1 | U2PPGA | Set if port became inaccessible without giving an off-line interrupt. |
| 2-8 | | Reserved. |
| 9-17 | UNIKOF | Contains the offset of the unit into KONTAB. |
| 18-26 | UNIKNM | Contains the controller number. |
| 27-35 | UNIPUN | Contains the physical unit number (lap plug number). |

124 UFB -- UFD DATA BLOCK

Description: One data block for every UFD/file structure pair which has an active file. All blocks for a file structure are linked together.

Defined in: COMMOD

Used by: FILFND, FILIO, FILUUO

```
        +-----------------------------------------------------------+
UFBTAL  |  Total of reserved + free blocks left in this UFD (1)|
        |-----------------------------------------------------------|
UFBPPB* |Next UFB, this user |  privileges   |  UN1 |              |
        |-----------------------------------------------------------|
UFBPT1  |              First retrieval pointer to UFD               |
        |-----------------------------------------------------------|
UFBWRT  |              FCFS quota             |No. of blocks (2)    |
        |-----------------------------------------------------------|
UFBFSN  | FSN  |                                                    |
        |-----------------------------------------------------------|
UFBAUJ  | Equals n is job n owns the AU for this UFB                |
        |-----------------------------------------------------------|
UFBWAT  |          -1 if AU is available (3)                        |
        +-----------------------------------------------------------+
```

* UFBPPB is described on the next page.

Notes:

1. UFBTAL will go negative if the user has exceeded quota and is using overdraw. No new ENTERs allowed if this is 0 or negative. Total includes RIBs.

2. In UFBWRT, Bits 0-26 contain the logged-in first-come/first-served quota. This is never decremented. Bits 27-35 contain the number of blocks written in the UFD itself.

3. The value of UFBWAT reflects the status of the AU resource. If the contents of this word is -1, the AU is available. If the word is 0, the AU is in use. If the word contains a non-negative number, that number reflects the number of jobs waiting for the resource.

UFBPPB -- Next UFB

```
+-----------------------------------------------------------+
|      UFBPPB                 |      PRV      | 1| UN1 |     |
+-----------------------------------------------------------+
 0                         17 18           26   28     32  35
```

| Word Label | Bits | Symbol | Content |
|---|---|---|---|
| UFBPPB | 0-17 | | Core address of next UFD data block for this PPN (in another file structure). |
| UFBPRV | 18-26 | UFYPRV | Access privileges for this UFD in this structure. Byte pointer UFYPRV is used to load this byte into AC. The codes that can be stored in UFBPRV are: |

| Code | Symbol | Meaning |
|---|---|---|
| 1 | UFRXRD | Can read directory |
| 2 | UFRXCR | Can create files |
| 4 | UFRXLK | Can do LOOKUPS |

| Word Label | Bits | Symbol | Content |
|---|---|---|---|
| UFBIPT | 27 | UFPIPT | Set if UFBPT1 is the only retrieval pointer for this UFD. |
| UFBUN1 | 28-31 | COYUN1 | Logical unit number within file structure associated with first retrieval pointer. |

## 125 UN BLOCK -- USER NI BLOCK

Description: Used to communication function specific data between the ethernet driver (ETHSER) and its users.

Defined in: ETHPRM

Used by: D8EINT, DNADLL, ETHSER, ETHUUO, LATSER, LLMOP

The Ethernet functions are described below the UN block.

| | |
|---|---|
| UN.PID | Portal id |
| UN.SID | Secondary id |
| UN.RID | Request id |
| UN.STA* | Portal status word |
| UN.JCH | JCH of portal owner |
| UN.UID | User id |
| UN.CBA | User callback address |
| UN.PTY* | Protocol identification word |
| UN.DAD(1,4) | Destination ethernet address (2 words) |
| UN.SAD(1,4) | Source ethernet address (2 words) |
| UN.BSZ(2) | Datagram buffer size in bytes |
| UN.BFA(2) | Datagram buffer byte pointer (2 words) |
| UN.CAR(3,4) | Current ethernet address (2 words) |
| UN.HAD(3,4) | Hardware ethernet address (2 words) |

Notes:

1. The destination and source ethernet addresses are only used by the NU.RCV (receive datagram) and NU.XMT (transmit datagram) functions. Additionally, the destination ethernet

address is used by the NU.EMA (enable multi-cast address) and NU.DMA (disable multi-cast address) functions to specify the multi-cast address.

2. The datagram buffer size and byte pointer are used by the NU.RCV (receive datagram) and NU.XMT (transmit datagram) functions. If the datagram size is specified as zero, then UN.BFA is assumed to be the address of an MSD chain for the datagram. Additionally, these words are used to specify auxiliary buffers for several other functions.

3. The current ethernet address and the hardware ethernet address are used by the NU.RCI (read channel information) function.

4. The ethernet address is stored as six 8-bit bytes left justified in two words.

Ethernet User Functions:

All calls to the ethernet driver (ETHSER) are made by specifying a function code and the address of an UN block where arguments for the function are stored.

Portals implement a specific protocol on the ethernet. A protocol user creates a new portal with the NU.OPN function. A portal is closed by the NU.CLO function. Porstal user queue receive and transmit datagrams via the NU.RCV and NU.XMT functions. Individual multi-cast ethernet addresses can be enabled or disabled by the NU.EMA and NU.DMA functions. Functions exist for getting information about ethernet channels, portals, and kontrollers.

| Value | Symbol | Description |
|---|---|---|
| 1 | NU.OPN | Open portal |
| 2 | NU.CLO | Close portal |
| 3 | NU.RCV | Queue receive datagram buffer |
| 4 | NU.XMT | Queue transmit datagram buffer |
| 5 | NU.EMA | Enable multi-cast address |
| 6 | NU.DMA | Disable multi-cast address |
| 7 | NU.RCL | Read ethernet channel list |
| 10 | NU.RCI | Read ethernet channel information |
| 11 | NU.RCC | Read ethernet channel counters |
| 12 | NU.SCA | Set ethernet channel address |
| 13 | NU.RPL | Read ethernet portal list |
| 14 | NU.RPI | Read ethernet portal information |
| 15 | NU.RPC | Read ethernet portal counters |
| 16 | NU.RKL | Read ethernet kontroller list |
| 17 | NU.RKI | Read etherent kontroller information |
| 20 | NU.RKC | Read ethernet kontroller counters |

125.1 UN.STA -- Portal Status Word

```
|-----------------------------------------------------------------|
|RUN|ZRO|   |ADS|                    |Time domain reflectometry value|
|-----------------------------------------------------------------|
 0   1   2  3-4 5-----------------17 18--------------------------35
```

| Bits | Symbol | Meaning |
|---|---|---|
| 0 | UNRUN | Portal is in run state (online) |
| 1 | UNZRO | Zero counters flag |
| 3-4 | UNADS | Address space of datagram buffer: |
| | UNA.EV | 0 = Exec virtual |
| | UNA.UV | 1 = User virtual |
| | UNA.PH | 2 = Physical |
| 18-35 | UNTDR | Time domain reflectometry value |

125.2 UN.PTY -- Protocol Identification Word

```
|-----------------------------------------------------------------|
|PAD|   | CHN |        Unused        |           Protocol type     |
|-----------------------------------------------------------------|
 0  1-2 3---5 6----------------17 18-----------------------------35
```

| Bits | Symbol | Meaning |
|---|---|---|
| 0 | UNPAD | Protocol uses padding |
| 3-5 | UNCHN | Ethernet channel number |
| 18-35 | UNPRO | Protocol type: |
| | PT%INF | -1 = Information protocol |
| | PT%PRM | -2 = Promiscuous receiver type |
| | PT%UNK | -3 = Unknown protocol receiver type |

## 126 UNQTAB -- COMMAND TABLE

Description: Contains command characteristics bits for all monitor commands. Indexed by command name offset in COMTAB. SET commands are described by UNQTB2.

Defined in: COMMON

Used by: COMCON, UUOCON

See also: COMTAB, COMTB2, DISP, DISP2, UNQTB2

```
+-------------------------------------------------------+
|            Command bits (described below)             |
+-------------------------------------------------------+
```

| Bits | Symbol | Meaning |
|---|---|---|
| 0 | NOINCK | No core needed for command |
| 1 | NOCRLF | No automatic <CRLF> |
| 2 | NOPER | No printing monitor prompt (period) |
| 3 | TTYRNU | Set terminal to user mode and start program |
| 4 | TTYRNC | Keep terminal in monitor mode and start program. |
| 5 | TTYRNW | Set terminal to user level and restore I/O status. |
| 6 | CMWRQ | Requeue job after command wait. |
| 7 | NOMESS | No command response, ever. |
| 8 | ERRFLG | Command error |
| 9 | SACFLG | Command was executed in an alternate context. |
| 10 | NOFLM | Super-noCRLF. |
| 18 | NOCORE | No core needed for the command. |
| 19 | NOJOBN | No job number needed for command. |
| 20 | NOLOGN | A job does not need to be logged in to use this command. |
| 21 | NOACT | Command must wait until job's devices are not active. |
| 22 | NORUN | The job must not be running. <CTRL/C> required. |
| 23 | INCORE | Job must be in core, if it has core. |
| 24 | NXONLY | Not legal for execute-only program |
| 25 | NBATCH | Not legal for batch job. |
| 26 | CMDERR | Error encountered in command processing. |
| 27 | NORCMP | Allow use by job that is not logged in, on a remote terminal, even with M.RCMP set. |
| 28 | | Reserved. |
| 29 | CUSTMR | Reserved for customer definition. |
| 30-31 | | Reserved. |
| 32 | UNIQ.1 | Command is unique to one character. |
| 33 | UNIQ.2 | Command is unique to two characters. |
| 34 | UNIQ.3 | Command is unique to three characters. |
| 35 | UNIQ.4 | Command is unique to four characters. |

## 127 UNQTB2 -- SET COMMAND TABLE

Description: Contains bits describing characteristics of the SET commands. UNQTAB contains descriptions of the remainder of the monitor commands. The tables are formatted identically. UNQTB2 is indexed by command name offset into COMTB2.

Defined in: COMMON

Used by: COMCON, UUOCON

See also: COMTAB, COMTB2, DISP, DISP2, UNQTAB

```
+--------------------------------------------------------+
|          Command characteristics bits                  |
+--------------------------------------------------------+
```

Refer to UNQTAB for definitions of bits.

128 UNQTBC -- CUSTOMER-DEFINED COMMAND TABLE

Description: Contains command characteristics bits for all customer-defined monitor commands. Indexed by command name offset in CSTTAB.

Defined in: COMMON

Used by: COMCON, UUOCON

See also: CSTTAB, DISPC

```
+--------------------------------------------------------+
|           Command bits (described for UNQTAB)          |
+--------------------------------------------------------+
```

## 129 UNWTAB -- UNWIND RESOURCE TABLE

Description: Contains the names of routines for unwinding scheduler interlocks and resource waits. This table is equivalent to AVALTB and REQTAB. The default unwinding routine is UNWRES. The default scheduler routine is SCDRES.

Defined in: S

Used by: SCHED1

Each word in this table appears as:

```
+-----------------------------------------------------------+
| Ptr to unwind routine     | Ptr to scheduler routine      |
+-----------------------------------------------------------+
```

Where the left half contains the address of the routine to use to unwind the resource, and the right half contains the address of the scheduler level routine to get the resource.

## 130 UFD -- USER FILE DIRECTORY

Description: Contains the locations for all files in the structure belonging to a particular project programmer number. One UFD in each structure for each project programmer number having any files in that structure.

Defined in: COMMOD

Used by: FILFND, FILIO, FILUUO

```
+---------------------------------------+
|             File 1 entry              |
|---------------------------------------|
|             File 2 entry              |
|---------------------------------------|
|             File 3 entry              |
|---------------------------------------|
|                                       |
+---------------------------------------+
```

Format of each entry:

```
        +---------------------------------------+
UFDNAM  |         File name in SIXBIT           |
        |---------------------------------------|
UFDEXT  |    Extension        |       CFP       | UFDCFP
        +---------------------------------------+
```

Note:

See MFD for discussion of compressed file pointers (CFPs).

131 UPT -- USER PROCESS TABLE

Description: Contains information about each job, and is used by the monitor to control memory mapping, scheduling, and I/O, and contains "scratch space" for dynamically changing variables about the job.

The UPT is pointed to by the GETTAB table .GTUPM (100), and has been called the User Page Map Page (UPMP) for many years. The UPT points to the page maps for each user section that has been created, but contains no page mapping information itself.

The offsets into the UPT are often called by the symbol .USxxx, which is equivalent to .UPxxx (listed below) plus .UPMP (the start of the UPT). Offset values are shown for items that are used by hardware.

Defined in: S.MAC

| Symbol | Map | |
|---|---|---|
| JOBPDO | Push down list (156 words) | |
| JOBPRO | Protected job data area (24 words) | |
| .UPLPS | Current first virtual page numbers on swapout, or pointer to table of same for fragmented, low segment swapout | |
| .UPSLX | | SWPLST index (paging I/O) |
| .UPTMP | Temporary locations used for swapping (6 words) | |
| .UPFFT | Virtual time of first page fault | |
| .UPLFT | Virtual time of last fault | |
| .UPVCT | Real page faults | Faults when page is in core |
| .UPREL | Highest location gotten by CORE UUO or command | |
| .UPNXP | Page range as specified in PAGE. arg. list, or current page being processed by PAGE. | |
| .UPJOB | Job number | |
| .UPMEM | Total virtual memory a job has | |
| .UPHSE | Virtual address of the end of the high segment | |
| .UPHSS | Virtual address of the start of the high segment | |

|--------------------------------------------------------|

<table>
<tr><th>Symbol</th><th colspan="2">Map</th></tr>
<tr><td>.UPVRT</td><td colspan="2">Non-zero if job is virtual (LH=high seg.,RH=low seg.)</td></tr>
<tr><td>.UPBTS*</td><td colspan="2">Random collection of bits (see Notes)</td></tr>
<tr><td>.UPANA</td><td colspan="2">Count of non-accessible pages</td></tr>
<tr><td>.UPICT</td><td colspan="2">Incremental count of page faults</td></tr>
<tr><td>.UPPFH</td><td colspan="2">Copy of .JBPFH on swap-out</td></tr>
<tr><td>.UPFOP</td><td colspan="2">Used by FILOP. to recover from a page fail</td></tr>
<tr><td>.UPHVA</td><td colspan="2">Used for address checking at interrupt level</td></tr>
<tr><td>.UPLST</td><td>Pointer to swappable DDBs</td><td>Ptr to saved context blk</td></tr>
<tr><td>.UPFCC</td><td colspan="2">Header for cached free space</td></tr>
<tr><td>.UPFCU</td><td colspan="2">Header for uncached free space (KL only)</td></tr>
<tr><td>.UPFCD</td><td>Core loc. of SWITCH.INI</td><td>Header for restricted free space</td></tr>
<tr><td>.UPLNM</td><td>T4</td><td>Ptr. to logical name space</td></tr>
<tr><td>.UPCTA</td><td></td><td>Extended channel table loc.</td></tr>
<tr><td>.UPMBF</td><td colspan="2">Address of monitor buffer</td></tr>
<tr><td>.UPLBF</td><td colspan="2">Flag to indicate use of extra page of directory<br>0=don't get, -n=IOWD for it, n=can get</td></tr>
<tr><td>.UPSPT</td><td colspan="2">Current section pointer (swapping I/O, and so forth)</td></tr>
<tr><td>.UPNCR</td><td colspan="2">No. of core page in NZS</td></tr>
<tr><td>.UPSCT</td><td colspan="2">/ Array of counters for no. of pages (9 bits) /</td></tr>
<tr><td>.UPPFF</td><td colspan="2">PFH page fault PC flags</td></tr>
<tr><td>.UPPFC</td><td colspan="2">PFH page fault PC counter</td></tr>
<tr><td>.UPPFL</td><td colspan="2">Last page paged out by PFH</td></tr>
</table>

| Symbol | Offset | Map |
|---|---|---|
| .UPPFU | | Count of page faults for current UUO |
| .UPUSN | | Section no. read as argument from /USE on R, RUN, GET, MERGE commands |
| .UPCDB | | Mapping pointers for current CPU's CDB |
| .UPUSA | | Program start address |
| .UPEPL | | Address of extended pushdown list |
| .UPUAC | | Block of 20 words to hold job's ACs while getting PFH |
| .UPSBF | | Saved .UPMBF when doing 4-block read |
| .UPEND | | Last word allocated to UPMP |
| .UPPFT | | Addr of user page fault trap instruction |
| .UPAOT | 421 | Addr of user arithmetic trap instruction |
| .UPPDT | 422 | Addr of user push down list overflow instruction |
| .UPO3T | 423 | Addr of user trap 3 instruction |
| .UPMUO | 424 | MUUO flags \| MUUO opcode, AC |
| .UPMUP | 425 | MUUO old PC |
| .UPMUE | 426 | MUUO effective address |
| .UPUPF | 427 | Addr of MUU Process Context word |
| .UPMTS<br>.UPMTE | | MUUO trap vector (10 words)<br>End of trap vector |
| WSBTAB | | Working set bit table for Section 0 (17 words) |
| | | Reserved |
| .LMPFW | 500 | Page fail word |
| .LMPFP | 501 | Page fail old PC word (2 words) |
| | 503 | Page fail new PC word |

| Symbol | Offset | Map |
| --- | --- | --- |
| .LMEBH | 504 | EBOX cycle meter count (high-order bits) |
| .LMEBL | 505 | EBOX cycle meter count (low-order bits) |
| .LMMBH | 506 | MBOX cycle meter count (high-order bits) |
| .LMMBL | 507 | MBOX cycle meter count (low-order bits) |
| | | Unused |
| SECTAB | | Address of user's section page map pages |
| | | Reserved for software |
| .UMORG | | Map slots for funny space follow this word |
| .UMWSB | | Map slot for .WSBNZ |
| .UMUPM | | Map slot for current section map |
| .UMJDT | | Map slot for .JDAT |
| .UMVJD | | Map slot for .VJDT |
| .UMTMP | | Map slot for temporary use |
| .UMJBK | | Map slot used by JOBPEK |
| .UMNZM | | Reserved |
| .UMUUP | | Cached UPT map slot |

Notes:

The bits defined in .UPBTS are:

| Bits | Symbol | Meaning |
|---|---|---|
| 0 | UP.BIG | User has created extended sections |
| 1 | UP.WHC | Working set has changed. |
| 2 | UP.MGP | Monitor got PFH |
| 3 | UP.GET | Running GET to get program that is to big to fit in core |
| 4 | UP.SAA | Set access allowed immediately |
| 5 | UP.CSP | Core image may contain SPY pages |
| 6 | UP.MPF | Merging PFH |
| 7 | UP.MMO | Job owned and released MM resource over scheduler call |
| 8 | UP.PGB | Paging I/O pages have not been returned |
| 9 | UP.IYB | "In-your-behalf" PPN |
| 10 | UP.WSS | Working set is scrambled |
| 11 | UP.DST | Don't put TTY at monitor level |
| 12 | UP.CXO | Core image (not just high segment) is execute-only |
| 13 | UP.FIP | FILOP. in progress |
| 14 | UP.DDW | Don't diddle working set |
| 15 | | Reserved |
| 16 | UP.NZS | Need to swap in non-zero section |
| 16 | UP.MAP | Current SWPLST entry has map information |
| 17 | UP.CTX | Context save/restore in progress. Used for RUN error recovery and high segment manipulation. |
| 18 | UP.SWF | SET WATCH FILES has been set |
| 19 | UP.JXP | MAPBAK. Call XPANDH when done. |
| 20 | UP.EPL | Count of non-accessible pages. |

## 132 USER PAGE MAP (SECTION MAP)

Description: The page map contains the physical page number that corresponds to each virtual page for the user. Indexed by virtual page number, this page contains one word for each virtual page. That word contains the physical address for that page, and the accessibility bits associated with the page. The monitor maintains one Section Map for each user section.

Each page pointer is formatted as follows:

```
+--------------------------------------------------------+
|Cd|P|W|K|C|A|SP|N|C|O|SS|     Addr                      |
+--------------------------------------------------------+
```

| Bits | Symbol | Meaning |
|---|---|---|
| 0-2 | | Accessibility code (see below) |
| 3 | PM.PUB | Public page |
| 4 | PM.WRT | Writable page |
| 5 | PM.KPM | "Keep Me" bit (page should not be cleared on sweeps). |
| 6 | PM.CSH | Page has been cached. |
| 7 | PM.AAB | Access allowed bit, for swapped-out pages. |
| 8 | PM.SPY | Spy privileges are required to access this page. |
| 9 | PM.NIA | No I/O allowed (usually set for high segment pages). |
| 10 | PM.COR | Page is in core. |
| 11 | PM.OIQ | On for in-progress queues. |
| 12 | PM.SSP | Slow swapping space. |
| 15-35 | PM.ADR | Disk or memory address field. This address is essentially an effective address. The accessibility code in Bits 0-2 (described below) is used to determine the handling of the address, whether immediate or indirect. If indirect, the address is mapped through the SPT (Special Pages Table). |

The accessibility codes stored in Bits 0-2 are:

| Code | Symbol | Meaning |
|---|---|---|
| 0 | PM.NCD | No access is allowed |
| 1 | PM.DCD | Immediate page pointer |
| 2 | PM.SCD | Shared page pointer |
| 3 | PM.ICD | Indirect page pointer |
| 4 | PM.ACD | Bit mask for all codes |

## 133 UUOTAB -- UUO DISPATCH ADDRESS TABLE

Description: Contains address of operator-dependent UUO routines.

Table is in order of UUO op code, with two addresses per entry. Entry n contains entries corresponding to op codes 40 + 2n, 41 + 2n.

Entries corresponding to invalid op codes contain the address of UUOERR. Some of these are reserved for future use by DIGITAL, others for customers. See current listing for specific examples.

Defined in: UUOCON

Used by: UUOCON

```
+---------------------------------------------------------+
|    Adr for op code 40        |    Adr for op code 41    |
|---------------------------------------------------------|
|    Adr for op code 42        |    Adr for op code 43    |
|---------------------------------------------------------|
/                              .                          /
/                              .                          /
|---------------------------------------------------------|
|    Adr for op code 76        |    Adr for op code 77    |
+---------------------------------------------------------+
```

134 WSBTAB -- WORKING SET BIT TABLE

Description: This bit table is found in the UPT from location 440 to 456. If a bit is on in this table, then the relative page is in core.

Defined in: S.MAC

Used by: VMSER

```
       +-------------------------------------+
       |                                     |
       |                                     |
       |                                     |
       |-------------------------------------|
 440   | | | | | |                           |
       |-------------------------------------|
       |                                     | WSBTAB
       |                                     |
       |-------------------------------------|
 456   |                           | | | | | |
       |-------------------------------------|
       |                                     |
```

# APPENDIX A

# ANF-10 FRONT END TABLES

## A.1 CHUNK WORDS

Description: Describes the format of chunks for messages.

Defined in: DNCNFG.P11

| | |
|---|---|
| 0 | Link to next chunk in current message |
| CN.MLK | Link to next message (or 0 if none) |
| CN.LEN | Message length, including NCL header, but excluding BCC |
| CN.TIM | DDCMP timer |
| CN.DDB | Address of DDB sending this message (depends on DEVN+FT.DTE NE 0) |
| CN.SCB | Pointer to SCB window for message |
| CN.ADR<br>CN.DDC | Address of next byte to use (start of DDCMP header: CN.DDC) |
| CN.CNT | Count of bytes left in message |
| CN.NCN | NCN saved here |
| | DDCMP header BCC |
| CN.NCT | NCT byte of NCL message (the NCL message begins here) |
| DATA | This is the message data |
| CNKSIZ==> | |

Notes:

FIRFRE  Address of first free chunk.

LSTFRE  Address of last free chunk.

FRECNT  Count of free chunks.

| | |
|---|---|
| FREMAX | Max number of chunks. |
| CNKLN1 | Max amount of data in first chunk of message. (CNKSIZ - CN.NCT) |
| CN.DT2 | Length of header in succeeding chunks of message. |
| CNKLN2 | Max amount of data in succeeding chunks of message. (CNKSIZ - CN.DT2) |

## A.2 DEVICE DATA BLOCKS

Description: Contains information needed to perform I/O operations. One such block for each device.

Defined in: DNDEV.P11

```
          +-----------------------------------------------------------------+
DB.STS*   |                          Status bits                            |
          |                                                                 |
          |DS.|DS.|DS.|DS.|DS.|DS.|DS.|DS.|DS.| 0 |DS.|DS.|DS.|DS.|DS.|DS.|
          |CON|XDS|TTY|IQU|EPL|XCH|COR|Q10|PAU|   |DIE|ACT|OUT|QUE|DSC|CAC|
          |-----------------------------------------------------------------|
DB.LNK    |                       Link address to                           |
          |                           next DDB                              |
          |-----------------------------------------------------------------|
DB.HDW    |                 Hardware address for device                     |
DB.DHB    |                                                                 |
          |-----------------------------------------------------------------|
DB.RPC    |                   Default starting address                      |
          |                (moved to DB.OPC by CLRDDB)                      |
          |-----------------------------------------------------------------|
DB.TPC    |                Timer-runout dispatch address                    |
          |                                                                 |
          |-----------------------------------------------------------------|
DB.DVT    |                       Device attributes                         |
          |                                                                 |
          |-----------------------------------------------------------------|
DB.DVU    |     Device controller type      |        Device unit type       |
DB.DVV    |                                 |                               |
          |-----------------------------------------------------------------|
DB.WID    |                Carriage width for terminals                     |
DB.RLN    |              Record length for other devices                    |
          |-----------------------------------------------------------------|
DB.ACR    |          Unit number            |       Auto-crlf point         |
DB.UNI    |                                 |                               |
          |-----------------------------------------------------------------|
DB.OBJ    |       NCL remote object         |       NCL object type         |
DB.ROT    |             type                |                               |
          |-----------------------------------------------------------------|
DB.MDR    |Maximum number of chunks device| Maximum number of output        |
DB.CHK    |  can have before sending DRQ  |   data requests (DRQ)           |
          |-----------------------------------------------------------------|
DB.TYP    |     Restricted node number      |         Type of device        |
DB.RNN    |    (Depends on FT.RNN = 1)      |                               |
          |-----------------------------------------------------------------|
```

```
            |--------------------------------------------------------------|
DB.PFH      | Node number to reconnect to |Preferred host to connect to   |
DB.RCN      |    (Depends on FTHOST)      |      (Depends on FT.PFH)      |
            |--------------------------------------------------------------|
DB.OLA      |                      Our link address                        |
            |                                                              |
            |--------------------------------------------------------------|
DB.RDT*     |                Remote data type (see below)                  |
            |            depends on FT.RDM, FT.RDP, or FT.RDA              |
            |--------------------------------------------------------------|
DB.TSK      |    Address of task for this device:  Printer get task       |
            |                 (depends on FT.TSK EQ 1)                     |
            |--------------------------------------------------------------|
DB.TSK+2    |                     Keyboard get task                        |
            |                 (depends on FT.TSK EQ 1)                     |
            |--------------------------------------------------------------|
DB.TSK+4    |                     Printer put task                         |
            |                 (depends on FT.TSK EQ 1)                     |
            |--------------------------------------------------------------|
DB.TSK+6    |                     Keyboard put task                        |
            |                 (depends on FT.TSK EQ 1)                     |
            |--------------------------------------------------------------|
DB.DCS*     |                   Device control status                      |
DB.ZER      |                                                              |
            |TS.|TS.|RNG|TS.|TS.|TS.|TS.|TS.|TS.|TS.|TS.|TS.|TS.|TS.|TS.|TS.|
            |ADL|DSR|CAR|DTR|CRL|LMD|TIW|FRM|TAB|TAP|PAG|IMO|IMI|FRZ|.LC|DFE|
            |--------------------------------------------------------------|
DB.MML      |                  Maximum message length                      |
            |                        for device                            |
            |--------------------------------------------------------------|
DB.DCM*     |                             |       Data code and mode      |
            |                             |          (see below)          |
            |   |   |   |   |   |   |   |   |DCM|DCM|DCM|DCM|DCM|DCM|DCM  |
            |   |   |   |   |   |   |   |   |.CF|.XX|.DI|.HO|.IM|.EB|.AS  |
            |--------------------------------------------------------------|
DB.RLA      |                    Remote link address                       |
            |                                                              |
            |--------------------------------------------------------------|
DB.SCB      |              SCB address for user of this device             |
            |                                                              |
            |--------------------------------------------------------------|
DB.OBF      |            Pointer to from-10 (output) buffer                |
            |                                                              |
            |--------------------------------------------------------------|
DB.OLN      |                Length of current message                     |
            |                                                              |
            |--------------------------------------------------------------|
DB.OCN      |              Count for current sub-message                   |
            |                                                              |
            |--------------------------------------------------------------|
```

```
            |----------------------------------------------------------------|
DB.OAD      |                     Current byte pointer                       |
            |                                                                |
            |----------------------------------------------------------------|
DB.OPC      |                PC to run at when in run queue                  |
            |                                                                |
            |----------------------------------------------------------------|
DB.ODR      |    Current column number     |       Number of output          |
DB.COL      |                              |        data requests            |
            |----------------------------------------------------------------|
DB.IDR      |     Compressed character     |        Number of input          |
DB.CCN      |            Count             |        data requests            |
            |----------------------------------------------------------------|
DB.TIM      |       Timer type code        |          Timer value            |
            |                              |negative=seconds,positive=jiff   |
            |----------------------------------------------------------------|
DB.HLD      |                    Character being held                        |
            |            (used when outputing free CRLF, ...)                |
            |----------------------------------------------------------------|
DB.VFU      |                Pointer to line-printer's VFU                   |
            |                                                                |
            |----------------------------------------------------------------|
DB.CHR      |                 Character being uncompressed                   |
            |                                                                |
            |----------------------------------------------------------------|
DB.IBF      |              Pointer to to-ten (input) buffers                 |
            |                                                                |
            |----------------------------------------------------------------|
DB.ICC      |                              |     Input character count       |
            |                              |                                 |
            |----------------------------------------------------------------|
DB.ICN      |              Input message count:  total message               |
            |                                                                |
            |----------------------------------------------------------------|
DB.ICN+2    |           Incremental count for current sub message            |
            |                                                                |
            |----------------------------------------------------------------|
DB.ICN+4    |               Address of field for byte count                  |
            |                                                                |
            |----------------------------------------------------------------|
DB.IAD      |                  Input character address                       |
            |                                                                |
DB.SIZ=>    +----------------------------------------------------------------+
```

End of standard DDB.

The following pages contain the terminal-dependent data.

```
             +-------------------------------------------------------------+
DB.BIT       |                    Line number mask:                        |
             |       PDP-11 bit number "N" is set for line number "N"      |
             |-------------------------------------------------------------|
DB..LN       |   Fill timer for <^H> <010>   |   4 bit binary line number  |
DB.FIL       |        (backspace)            |                             |
             |-------------------------------------------------------------|
DB.FIL+2     |   Fill timer for <LF> <012>   |  Fill timer for <^I> <011>  |
             |        (line feed)            |           (tab)             |
             |-------------------------------------------------------------|
DB.FIL+4     |   Fill timer for <FF> <014>   |  Fill timer for <VT> <013>  |
             |        (form feed)            |       (vertical tab)        |
             |-------------------------------------------------------------|
DB.EPL       |      Serial number for        |   Fill time for <CR> <015>  |
             |   Echo pipeline marker (EPL)  |     (carriage return)       |
             |-------------------------------------------------------------|
DB.LCB       |                      Pointer to LCB                         |
             |          for physical line to user of this device           |
             |-------------------------------------------------------------|
DB.DNS       |     DN-11 timer (seconds)     |  DN-11 table displacement   |
DB.DNT       |     (depends on FTDN11)       |  & stats (depends on FTDN11)|
             |-------------------------------------------------------------|
DB.TTS*      |                   TTY status (see below)                    |
             |                                                             |
             |   |   |   |   |   |   |   |   |   |   |   |   |   |CHR|TT.|
             |   |   |   |   |   |   |   |   |   |   |   |   |   |APL|APL|
             |-------------------------------------------------------------|
DB.DNR       |               DN-11 request word (see below)                |
DB.TZR       |                    (depends on FTDN11)                      |
             |-------------------------------------------------------------|
DB.BCD*      |             BCD terminal status  (see below)                |
             |                    (depends on FT2741)                      |
             |BCD|BCD|BCD|BCD|BCD| <==| BCD|==> |BCD|BCD|BCD|BCD|BCD|BCD|BCD|
             |274|XRB|KBL|PRL|CDB|    | COD |BRK|UPS|OCR|RCR|CON|TDY|APL|HDB|
             |-------------------------------------------------------------|
DB.STR       |                  Pointer to string to type                  |
             |                                                             |
             |-------------------------------------------------------------|
DB.TOC       |            Number of output characters in chunk             |
             |                                                             |
             |-------------------------------------------------------------|
DB.TOB       |                     TTY output buffer:                      |
             |                 pointer to first character                  |
             |-------------------------------------------------------------|
DB.TOB+2     |                  Pointer to last character                  |
             |                                                             |
             |-------------------------------------------------------------|
```

```
           |-------------------------------------------------------------|
DB.ASP     |        Character for         |         ASAP character       |
DB.BUF     |        DH-11 to type         |            (^G, ...)         |
           |-------------------------------------------------------------|
DB.FTM     |              Fill time for current character                |
           |                                                             |
           |-------------------------------------------------------------|
DB.PCN     |       Printer count (number of characters from NCL)         |
           |    (this and the following words depend on FT.TSK EQ 1)     |
           |-------------------------------------------------------------|
DB.PPT     |                   Printer putter pointer                    |
           |                                                             |
           |-------------------------------------------------------------|
DB.PTK     |                   Printer taker pointer                     |
           |                                                             |
           |-------------------------------------------------------------|
DB.KPT     |                  Keyboard putter pointer                    |
           |                                                             |
           |-------------------------------------------------------------|
DB.KTK     |                   Keyboard taker pointer                    |
           |                                                             |
           |-------------------------------------------------------------|
DB.KQU     /                      Keyboard queue                         /
           |                  (length is "TQS" words)                    |
DB.TSZ=>   +-------------------------------------------------------------+
```

A.2.1 DB.STS -- Status Bits

| Symbol | Value | Description |
|---|---|---|
| DS.CAC | 000001 | Send out CONNECT ACCEPT |
| DS.DSC | 000002 | Send out DISCONNECT CONFIRM |
| DS.QUE* | 000004 | Device has a RUN request in queue |
| DS.OUT* | 000010 | Device does output |
| DS.ACT* | 000020 | Device is ACTIVE |
| DS.DIE | 000040 | Abort, other end of connection died |
| DS.IST | 000100 | Input stopped by XOFF |
| DS.PAU | 000200 | Task is using TTY for input (FT.TSK=1) |
| DS.Q10 | 000400 | Task has queued characters to ten (FT.TSK=1) |
| DS.COR | 001000 | Device wants core to run |
| DS.XCH | 002000 | Send CHARACTERISTICS message |
| DS.EPL | 004000 | Send echo pipeline marker |
| DS.IQU | 010000 | Input has been queued to NCL |
| DS.TTY* | 020000 | Device is a terminal |
| DS.XDS | 040000 | Send DB.DCS to other node |
| DS.CON | 100000 | Device is CONNECTED |

* = Bits cleared on DDB initialization (CLRDDB)

A.2.2 DB.RDT -- Remote Data Type

| Symbol | Value | Description |
|---|---|---|
| RDEMPT | 000001 | Multipoint |
| RDEPTP | 000002 | Point-to-Point |
| RDEASC | 000004 | ASCII |
| RDEBRK | 100000 | ASCII break was seen |

A.2.3 DB.DCS -- Device Control Status

Bits For Terminals:

| Symbol | Value | Description |
|---|---|---|
| TS.DFE | 000001 | Deferred echo mode |
| TS..LC* | 000002 | Lower case mode |
| TS.FRZ | 000004 | Output frozen by XOFF |
| TS.IMI | 000010 | Input image mode |
| TS.IMO | 000020 | Output image mode |
| TS.PAG | 000040 | TTY paging enabled (XON/XOFF) |
| TS.TAP | 000100 | Paper tape mode |
| TS.TAB* | 000200 | Hardware tabs |
| TS.FRM* | 000400 | Hardware form feeds |
| TS.TIW | 001000 | Terminal is in input wait |
| TS.LMD | 002000 | Terminal is in line mode |
| TS.CRL | 004000 | No free CRLF |
| TS.DTR* | 010000 | DTR is present on line |
| TS.RNG* | 020000 | RING is present on line |
| TS.CAR* | 020000 | CARRIER is present on line |
| TS.DSR* | 040000 | DSR is present on line |
| TS.ADL | 100000 | Line is an auto-dial line (BELL 801) |

* = Preserved on a system restart

DB.DCS -- Bits For Line Printers:

| Symbol | Value | Description |
|---|---|---|
| LPT.FE | 000001 | Fatal error |
| LPT.FL | 000002 | Offline |
| LPT.PZ | 000004 | Page count zero |
| LPT.VE | 000010 | VFU error |
| LPT.RE | 000020 | RAM error |
| LPT.IC | 000040 | Illegal |
| LPT.OV | 000100 | Optical VFU |
| LPT.PE | 000200 | Parity |
| LPT.DE | 000400 | Demand |
| LPT.ME | 001000 | Master synch error |
| LPT.RV | 002000 | Receiving VFU data from the 10 |

| | | |
|---|---|---|
| LPT.RR | 004000 | Receiving RAM data from the 10 |

### A.2.4 DB.DCM -- Data Code And Mode

| Symbol | Value | Description |
|---|---|---|
| DCM.AS | 001 | ASCII |
| DCM.EB | 002 | EBCDIC |
| DCM.IM | 004 | Image mode |
| DCM.HO | 010 | Hollerith mode (card-reader only) |
| DCM.DI | 020 | DEC Image mode (card-reader only) |
| DCM.XX | 040 | Reserved |
| DCM.CF | 100 | Compressed data mode |

### A.2.5 DB.TTS -- TTY Status

| Symbol | Value | Description |
|---|---|---|
| TT.APL | 000001 | Terminal is in APL mode |
| CHRAPL | 000002 | This is an APL character |

### A.2.6 DB.BCD -- BCD Terminal Status

| Symbol | Value | Description |
|---|---|---|
| BCD274 | 100000 | This line is a 2741 |
| BCDXRB | 040000 | Sending a reverse break |
| BCDKBL | 040000 | Keyboard is currently locked |
| BCDPRL | 010000 | Printer is currently locked |
| BCDCDB | 004000 | Last time line was reversed was to get input |
| BCDCOD | 003400 | Code for current golf ball |
| BCDBRK | 000200 | Currently processing a receive BREAK |
| BCDUPS | 000100 | Set if in upper shift mode |
| BCDOCR | 000040 | Set if last character xmitted was a <CR> |
| BCDRCR | 000020 | Set if last character received was a <CR> |
| BCDCON | 000010 | Set if last character was a control fan (^) |
| BCDTDY | 000004 | Set if terminal is in TTY TIDY mode |
| BCDAPL | 000002 | Set if terminal is in APL mode |
| BCDHDB | 000001 | Set if terminal has DEBREAK feature |

Notes:

The symbol FIRDDB points to the first DDB in the system.

Start zeroing at DB.ZER on a restart.

Value of TQS (number of words in keyboard queue) is 20 octal.

## A.3 DH-11 BLOCK

Description: Contain device dependent information pertaining to each DH11. One such block for each DH11. Referenced by DH#BLK where "#" is the DH-11 number (0,1,2,...)

Defined in: DNDH11.P11

```
            +-----------------------------------------------------------------+
0           |               Hardware address of this DH-11                    |
            |                    (0 if not present)                           |
            |-----------------------------------------------------------------|
DHBBAR      |                    Active lines mask                            |
            |          PDP-11 bit "N" is on if line "N" is active             |
            |-----------------------------------------------------------------|
DHB.BN      |             Line number on node of first line                   |
            |                       on this DH-11                             |
            |-----------------------------------------------------------------|
DHB.DM      |              Address of DM-11BB for this DH-11                  |
            |                        (0 if none)                              |
            |-----------------------------------------------------------------|
DHB.VC      |               Vector address of this DH-11                      |
            |                                                                 |
            |-----------------------------------------------------------------|
DHB.LC      /       This space contains the line control blocks               /
            |                   (LC..SZ X 20 words)                           |
DHB.SZ==>   +-----------------------------------------------------------------+
```

## A.4 DMC11 BASE TABLE

Description: Contains device dependent information for each DMC11. One such block for each DMC11. Referenced by DMBAS# where "#" is the DMC11 number (0 - DMCN).

Defined in: DNCDMC.P11

```
           +--------------------------------------------------------------+
           |                                |             Fill            |
           |                                |           (5 bytes)         |
           |--------------------------------------------------------------|
           |                                |                             |
           |                                |                             |
           |--------------------------------------------------------------|
           |         Received NAK's         |                             |
 MB.RNK    |                                |                             |
           |--------------------------------------------------------------|
 MB.XNB    |           Sent NAK's           |          Sent NAK's         |
 MB.XNH    |           bad header           |          no buffer          |
           |--------------------------------------------------------------|
 MB.XND    |           Sent REP's           |          Sent NAK's         |
 MB.XRP    |                                |           Data BCC          |
           |--------------------------------------------------------------|
 MB.RRP    |          Fill to end           |        Received REP's       |
 MB.XX     |     (255. - MB.RRP bytes)      |                             |
           |--------------------------------------------------------------|
           /                                /                             /
           |                                |                             |
 MB.LEN==> +--------------------------------------------------------------+
```

Notes:

MB.LEN = 256 (decimal).

DMCLBO is the address of the first DMC11 line.

The first five bytes and the remaining bytes after MB.RRP are used for other counters by the DMC11, but are not used by the DMC11 driver. For further information regarding the base table, refer to the DMR11 Synchronous Controller User's Guide.

## A.5 DMC11 MESSAGE BUFFERS

Description: Blocks of physically contiguous memory used by the DMC11 to send and receive data. Buffers are shared by all DMC11's on the system. Referenced by CB..# where "#" is the buffer number (0 - CB.NUM).

Defined in: DNCDMC.P11

```
           +-------------------------------------------------------------+
 CB.LNK    |                    Link to next buffer                      |
           |                                                             |
           |-------------------------------------------------------------|
 CB.CNT    |                  Number of bytes in buffer                  |
           |                                                             |
           |-------------------------------------------------------------|
 CB.DAT    /                         Data area                           /
           |                      (MSGMAX bytes)                         |
 CB.LEN==> +-------------------------------------------------------------+
```

Notes:

MSGMAX defaults to 512 (decimal) bytes.

CB.NUM is the total number of buffers in the system:
CB.NUM = DMCN times <DMCIBF + DMCOBF> where:
  DMCN is the number of DMC11s on the system
  DMCIBF is the number of input buffers
  DMCOBF in the number of output buffers.

DMCIBF and DMCOBF default to 4.

CBFRST is the address of the first buffer.
CBFSTF is the address of the first free buffer.
CBFREC is the count of free buffers.

## A.6 DMC11 MESSAGE BUFFER QUEUES

Description: Contain pointers to DMC-11 buffers. Allocated within each Line Block on systems with DMC-11s.

Defined in: DNCDMC.P11

```
          +-------------------------------------------------------------+
 CBQ.CT   |                        Queue length                         |
          |                                                             |
          |-------------------------------------------------------------|
 CBQ.FS   |                  Address of first buffer                    |
          |                                                             |
          |-------------------------------------------------------------|
 CBQ.LS   |                   Address of last buffer                    |
          |                                                             |
 CBQ.LN==> +-------------------------------------------------------------+
```

## A.7 DN11 BLOCK

Description: Contain status and device information for each line of an automatic calling unit. Referenced by DN#BLK where "#" is the DN11 number (0 - DN11N).

Defined in: DNDN11.P11

```
            +---------------------------------------------------------------+
            |                   Hardware status register                    |
            |                                                               |
            |---------------------------------------------------------------|
            |               DDB address for associated ACU                  |
            |                                                               |
            |---------------------------------------------------------------|
 DNPTR      |                Pointer to last digit dialed                   |
            |                                                               |
            |---------------------------------------------------------------|
            /                                /        Digits to dial        /
            |                                |  (^D18 bytes, first is NUL)  |
            |---------------------------------------------------------------|
            /        Last digit is 017       /                              /
            |                                |                              |
 DNTABL==>  +---------------------------------------------------------------+
```

Notes:

The first digit is always NUL.
The last digit is always 017.
One DN11 can handle up to 4 lines.

## A.8 DZ11 BLOCK

Description: Contain device dependent information pertaining to each DZ11. One such block for each DZ11. Referenced by DZ#BLK where "#" is the DZ11 number (0 - NDZ11).

Defined in: DNDZ11.P11

```
           +------------------------------------------------------------+
 DZADDR    |               Hardware address for this DZ11               |
           |                                                            |
           |------------------------------------------------------------|
 DZB.BN    |            Line number on node of first line               |
           |                     of this DZ11                           |
           |------------------------------------------------------------|
 DZB.VC    |                Vector address of this DZ11                 |
           |                                                            |
           |------------------------------------------------------------|
 DZB.BR    |                        Break bits                          |
           |                                                            |
           |------------------------------------------------------------|
 DZB.LC    /        This space contains the line control blocks         /
           |                (LC..SZ times 10 words)                     |
 DZB.SZ==> +------------------------------------------------------------+
```

## A.9 LINE BLOCK

Description: Contain information for DDCMP lines. One block for each DDCMP line. Referenced by LBLK# where "#" is the line number (0,1,2,...NTLINE).

Defined in: DNCNFG.P11

```
         +---------------------------------------------------------------+
LB.STS*  |                       Line status bits                        |
         |                         (see below)                           |
         |LS.|MPT|LS.|LS.|LS.|LS.|LS.|LS.|LS.|LS.|LS.|LS.|LS.|LS.|LS.|LS.|
         |.SS|NSP|SSY|XDT|XCT|.RN|.XG|.RG|.XQ|.RQ|NRP|XRP|XAK|XNK|STK|.ST|
         |---------------------------------------------------------------|
LB.ST5*  |                      Second status word                       |
         |                         (see below)                           |
         |L2.|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
         |DDP|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
         |---------------------------------------------------------------|
LB.LNK   |                    Link to next line block                    |
         |                                                               |
         |---------------------------------------------------------------|
LB.DDB   |                Addr. of associated DDP device                 |
         |                  (depends on FT.DDP NE 0)                     |
         |---------------------------------------------------------------|
LB.BIT   |               Bit corresponding to line number                |
         |     (PDP-11 bit number "N" is on for line number "N")         |
         |---------------------------------------------------------------|
LB.LNU   | Device service routine code |          Line number            |
LB.DVS*  |        (see below)          |             binary              |
         |---------------------------------------------------------------|
LB.LVL   |                        Level for link                         |
         |                                                               |
         |---------------------------------------------------------------|
LB.DHB   |          Asynch:  DH11/DZ11 device control block              |
LB.SLA   |              Synch:  line hardware address                    |
         |---------------------------------------------------------------|
LB..LN   |                 Asynch:  4 bit line number                    |
LB.SLV   |          Synch:  synchronous line vector address              |
         |---------------------------------------------------------------|
LB.LCB   |          DH11/DZ11 only:  line control block address          |
         |               (Depends on DH.MAX+DZ.MAX > 0)                  |
         |---------------------------------------------------------------|
LB.MPL   |           Link to next line drop for multipoint               |
         |                    (depends on FT.MPT)                        |
         |---------------------------------------------------------------|
```

```
          |---------------------------------------------------------------|
LB.MPN    |          Multipoint next station selection control            |
          |                     (depends on FT.MPT)                       |
          |---------------------------------------------------------------|
LB.MPA    |   Multipoint select timer     |  Multipoint station address   |
LB.MPT    |    (depends on FT.MPT)        |     (depends on FT.MPT)       |
          |---------------------------------------------------------------|
LB.MPS*   |  Additional control status    |    Multipoint node status     |
          |    (depends on FT.MPT)        |     (depends on FT.MPT)       |
          |   |   |   |   |   |   |   |   |MP.|MP.|MP.|MP.|   |MP.|   |MT.|
          |   |   |   |   |   |   |   |   |SEL|OFF|SOL|SNM|   |RTS|   |CMS|
          |---------------------------------------------------------------|
LB.NSS*   |                   NSP status (see below)                      |
          |                     (Depends on FTDCP1)                       |
          |   |NS.|NS.|NS.|   |NS.|   |   |   |   |   |   |   |   |   |   |
          |   |STR|EDS|CNF|   |NSQ|   |   |   |   |   |   |   |   |   |   |
          |---------------------------------------------------------------|
LB.NNM    |      Node number he           |        Node number for        |
LB.DNA    |        talks to               |           NSP node            |
          |---------------------------------------------------------------|
LB.MML    |                   Maximum message length                      |
          |                                                               |
          |---------------------------------------------------------------|
LB.SNM    /        Software ID            /         Station name          /
LB.SID    |      (SIDSIZ bytes)           |        (SNMSIZ bytes)         |
          |---------------------------------------------------------------|
LB.DAT    /  Configuration information    /             Date              /
LB.CNF    |      (NSPCFS bytes)           |        (DATESZ bytes)         |
          |---------------------------------------------------------------|
LB.VNN    |   ANF node name for DECNET    |   ANF node number for DECNET  |
LB.VNM    | (depends on FTDCP4) (6 bytes) |   (depends on FTDCP3!FTDCP4)  |
          |---------------------------------------------------------------|
          |                               |                               |
          |                               |                               |
          |---------------------------------------------------------------|
          |                               |                               |
          |                               |                               |
          |---------------------------------------------------------------|
          |  DECNET node number for ANF   |                               |
LB.HNN    |                               |                               |
          |---------------------------------------------------------------|
LB.HNM    |                               |   DECNET node name for ANF    |
          |                               |          (6 bytes)            |
          |---------------------------------------------------------------|
          |                               |                               |
          |                               |                               |
          |---------------------------------------------------------------|
          |                               |                               |
          |                               |                               |
          |---------------------------------------------------------------|
LB.CNN    /         Options list          /      ANF node to recieve      |
LB.CNF    |      (NSP$MX+1 bytes)         |        DECNET connects        |
```

|---------------------------------------------------------------|

| | | |
|---|---|---|
| LB.OCN | Count of ACKs received for messages sent | |
| LB.OCN+2 | Total number of NAKs received | |
| LB.OCN+4 | NAKs received for REP responses | |
| LB.OCN+6 | NAKs received because of bad BCC | |
| LB.OCN+1 | NAKs received for no room | |
| LB.ICN | Number of messages received OK | |
| LB.ICN+2 | Total bad messages | |
| LB.ICN+4 | Total transmitted REPs which won NAK responses | |
| LB.BNN | Bootstrapping timer | Bootstrapping node number |
| LB.FB | First DMC-11 buffer (depends on DMCN NE 0) | |
| LB.BAS | Address of DMC-11 base (depends on DMCN NE 0) | |
| LB.WHA | Address of caller of L.DOWN (depends on FTWHYD) | |
| LB.WHS<br>LB.WHN | Copy of LB.OBF+4 (depends on FTWHYD) | # of last ACKed message (depends on FTWHYD) |
| LB.STX* | For DMC-11: status word<br>For DUP-11: dispatch address for driver<br>bits: LS2 RUN, LS2 WAI, LS2 MAI | |
| LB.STY* | Status word for DUP-11 (depends on FTDUP11)<br>bits: UP$ RCC | |

| | | |
|---|---|---|
| LB.LCT | Active link count (depends on FTDCP1) | |
| LB.LKT | Pointer to link table (depends on FTDCP1) | |
| LB.LLE | Last link entry (depends on FTDCP1) | |
| LB.NSS* | Status word (see below) (depends on FTDCP3 or FTDCP4) | |
| | \| \| \| \| \| \| \| \| \| \| \| \| LBS.NQ \| LBS.L1 \| LBS.IC | |
| LB.LEB | Pointer to LEB chain (depends on FTDCP3 or FTDCP4) | |
| LB.SCB | SCB address for node at other end of this physical link | |
| LB.2ND | Pointer to next line block in the event of parallel lines | |
| LB.REP | REP timer counted down once a second | |
| LB.ROK<br>LB.LAR | Last message number ACK received | Last message number received OK |
| LB.LAP<br>LB.HSN | Highest message number sent | Last message number ACK processed |
| LB.RDC<br>LB.NCD | Last NAK code sent | REP timer (incr once a second) |
| LB.TRY | | Count of BCC NAKS rcvd for firs message in queue (under FT.BIG) |
| LB.XDN | Routine to JSR to on synchronous line transmit done interrupts | |
| LB.CTL | / Next control message to transmit (10 bytes) / | |

| | |
|---|---|
| LB.COB | Current output buffer<br>Word 1 = Pointer to 1st chunk of current message |
| LB.COB | Word 2 = Pointer to current chunk |
| LB.COB | Word 3 = Number of bytes left |
| LB.BOO | Pointer to bootstrap message to send |
| LB.OBF | Output buffers<br>Word 1 = Address of first buffer |
| LB.OBF+2 | Word 2 = Address of last buffer (or 0) |
| LB.OBF+4 | Word 3 = Number of messages in queue |
| LB.RDN | Dispatch address for receive done |
| LB.CIB | Current input buffer (current chunk address) |
| LB.CIB+2 | Number of characters left in message |
| LB.SXR | Transmitter address #1 |
| LB.SXR+2 | Transmitter word count #1 |
| LB.SXR+4 | Transmitter address #2 |
| LB.SXR+6 | Transmitter word count #2 |

```
          |-----------------------------------------------------------------|
LB.SRR    |                       Receiver address #1                       |
          |                                                                 |
          |-----------------------------------------------------------------|
LB.SRR+2  |                      Receiver word count #1                     |
          |                                                                 |
          |-----------------------------------------------------------------|
LB.SRR+4  |                       Receiver address #2                       |
          |                                                                 |
          |-----------------------------------------------------------------|
LB.SRR+6  |                      Receiver word count #2                     |
          |                                                                 |
          |-----------------------------------------------------------------|
LB.SLE    |             Count of synchronous line error interrupts          |
          |   (depends on FT.SLB - synchronous line error reporting)        |
          |-----------------------------------------------------------------|
LB.SLE+2  |              Hardware status on last error interrupt            |
          |   (depends on FT.SLB - synchronous line error reporting)        |
          |-----------------------------------------------------------------|
LB.SLE+4  |              Count of synchronous line xmit timeouts            |
          |   (depends on FT.SLB - synchronous line error reporting)        |
          |-----------------------------------------------------------------|
LB.CTY    |             Address of string to type on CTY for line           |
          |                  (depends on FT.HLP and FT.SLB)                 |
          |-----------------------------------------------------------------|
LB.CRS    |        Synchronous interface status at crash - word 0           |
          |                  (depends on DEBUG and FT.SLB)                  |
          |-----------------------------------------------------------------|
LB.CRS+2  |        Synchronous interface status at crash - word 1           |
          |                  (depends on DEBUG and FT.SLB)                  |
          |-----------------------------------------------------------------|
LB.CRS+4  |        Synchronous interface status at crash - word 2           |
          |                  (depends on DEBUG and FT.SLB)                  |
          |-----------------------------------------------------------------|
LB.CRS+6  |        Synchronous interface status at crash - word 3           |
          |                  (depends on DEBUG and FT.SLB)                  |
          |-----------------------------------------------------------------|
LB.IPT    |                          Input putter                           |
          |               relative to beginning of line block               |
          |-----------------------------------------------------------------|
LB.ITK    |                          Input taker                            |
          |               relative to beginning of line block               |
          |-----------------------------------------------------------------|
LB.IBF    /  Input buffers:  First 8 bytes are DDCMP ctrl msg or header.    /
          |     Word 5 is link to rest of message.  length=5_*NINBUF        |
LB.SIZ=>+-----------------------------------------------------------------+
```

Notes:

This is the end of the standard Line Block.

Optional sections are described on the following pages.

This is the DMC-11 specific section of the Line Block, beginning at LB.IBF.

```
              +-------------------------------------------------------------+
LB.INQ        /                  Buffers given to DMC to fill               /
              |                       (CBQ.LN words)                        |
              |-------------------------------------------------------------|
LB.IND        /                Filled buffers returned by DMC               /
              |                       (CBQ.LN words)                        |
              |-------------------------------------------------------------|
LB.OUT        /                  Buffers queued to be sent                  /
              |                       (CBQ.LN words)                        |
              |-------------------------------------------------------------|
LB.TMO        |                  Timer to time out DMC-11                   |
              |                                                             |
              +-------------------------------------------------------------+
```

## A.9.1 LB.STS -- First Word For Line Status

| Symbol | Value | Description |
|---|---|---|
| LS..ST | 000001 | Send a START |
| LS.STK | 000002 | Send a STACK |
| LS.XNK | 000004 | Send a NAK |
| LS.XAK | 000010 | Send an ACK |
| LS.XRP | 000020 | Send a REP message |
| LS.NRP | 000040 | Need response to REP message |
| LS..RQ | 000100 | RCV interrupt queued |
| LS..XQ | 000200 | XMIT done interrupt queued |
| LS..RG | 000400 | Sync receiver active |
| LS..XG | 001000 | Sync transmitter active |
| LS..RN | 002000 | Received NAK |
| LS.XCT | 004000 | Transmitting a CONTROL message |
| LS.XDT | 010000 | Transmitting a DATA message |
| LS.SSY | 020000 | Strip sync before next message |
| LS.MPT | 040000 | Multi-point line |
| LS.NSP | 040000 | NSP line (depends on FTDCP1, FTDCP3 or FTDCP4) |
| LS..SS | 100000 | Stripping sync now |

## A.9.2 LB.ST5 -- Second Word For Line Status

| Symbol | Value | Description |
|---|---|---|
| L2.DDP | 100000 | Line block in use as a DDP device |

## A.9.3 LB.DVS -- Interrupt Service Routine Codes

| Code | Symbol | Meaning |
|---|---|---|
| 0 | LS..DP | DP11 Line |
| 2 | LS..DS | DS11 Line |
| 4 | LS..DU | DU11 Line |
| 6 | LS..DV | DV11 Line |
| 10 | LS..DQ | DQ11 Line |
| 12 | LS..UP | DUP11 Line |
| 14 | LS..DM | DMC11 Line |
| 16 | LS..DH | DH11 Line |
| 20 | LS..DZ | DZ11 Line |

## A.9.4 LB.MPS -- Node Status

| Symbol | Value | Description |
|---|---|---|
| MP.SFF | 017 | Selection failure count |
| MP.SEL | 200 | Station selected |
| MP.OFF | 100 | Station offline |
| MP.SOL | 040 | Set offline when deselected |
| MP.SNM | 020 | Set select bit in next message transmitted |
| MP.RTS | 004 | Inhibits setting of RTS |
| MP.CMS | 001 | Set to enable MP.RTS transition |

## A.9.5 LB.NSS Bits -- NSP Status

Requires FTDCP1.

| Symbol | Value | Description |
|---|---|---|
| NS.STR | 040000 | Strip always |
| NS.EDS | 020000 | Enable dynamic stripping |
| NS.NSQ | 020000 | Something in queue for this line |
| NS.CNF | 010000 | Send CONFIGURATION message |

## A.9.6 LB.STX -- DMC-11 Status

| Symbol | Value | Description |
|---|---|---|
| LS2MAI | 000001 | Maintenance mode |
| LS2WAI | 000002 | Waiting for memory or buffers |
| LS2RUN | 000004 | DMC is running |

### A.9.7 LB.STY -- DUP11 Status

| Symbol | Value | Description |
|---|---|---|
| UP$RCC | 000001 | Set if BCC is good when service is called |

### A.9.8 LB.NSS Bits -- Status Word

Depends on FTDCP3 or FTDCP4.

| Symbol | Value | Description |
|---|---|---|
| LBS.IC | 000001 | In contact with NSP |
| LBS.L1 | 000002 | NSP node is level 1 (intercept node) |
| LBS.NQ | 000004 | Node is in NSP queue |

Notes:

Words LB.NNM through LB.DAT depend on FTDCP1 being set.

Words LB.VNN through LB.CNN depend on FTDCP3 or FTDCP4 being set.

LB.SXR and LB.SRR depend on at least one of the following being non-negative (asynchronous DDCMP or no DQ-11s):

FTDP11, FTDS11, FTDU11, FTDUP11, (NTLINE-NLINES)

Start zeroing from LB.ZER on a restart.

Default value of NINBUF is 4.

## A.10 LINE CONTROL BLOCK

Description: Contain control information pertaining to each terminal line. One block for each terminal line. Referenced by "LCB#" where "#" is the line number.

Defined in: DNCNFG.P11

```
          +-----------------------------------------------------------------+
LC.CAR*   |                      DM-11BB control word                        |
          |                  bits  0-9 are timer (LCB.TM)                    |
          |   |   |LCB|LCB|   |   |   |   |   |   |   |   |   |   |   |     |
          |   |   |.AB|.DS|   |   |   |   |   |   |   |   |   |   |   |     |
          |-----------------------------------------------------------------|
LC.STA*   |      Number of times LPR       |      State of modem control     |
LC.MOD    |          is modified           |  (see next page for values)     |
          |-----------------------------------------------------------------|
LC.XSP*   |      Coded receive speed       |       Coded transmit speed      |
LC.RSP    |   (whole word is "LC.SPD")     |                                 |
          |-----------------------------------------------------------------|
LC.PXS    |Saved RCV speed for non auto-bd|Saved xmit speed for non auto-b|
LC.PRS    |   (whole word is "LC.PSP")     |          data-set lines         |
          |-----------------------------------------------------------------|
LC.BLK    |                       Link to DDB or LB                          |
          |                                                                  |
          |-----------------------------------------------------------------|
LC.INS    |                Address of input service processor                |
          |                                                                  |
          |-----------------------------------------------------------------|
LC.OUS    |               Address of output service processor                |
          |                                                                  |
          |-----------------------------------------------------------------|
LC.CNT    |                          Count of bytes                          |
          |                 (DZ-11 lines only, NDZ11 NE 0)                   |
          |-----------------------------------------------------------------|
LC.BUF    |                     Buffer pointer for bytes                     |
          |                 (DZ-11 lines only, NDZ11 NE 0)                   |
LC..SZ=>+-----------------------------------------------------------------+
```

### A.10.1 LC.CAR -- DM-11BB Control Word

| Symbol | Value | Description |
|---|---|---|
| LCB.BK | 100000 | Break character flag (framing error) |
| LCB.IG | 040000 | Ignore line for one character |
| LCB.AB | 020000 | Autobaud Line |
| LCB.DS | 010000 | Data Set Line |
| LCB.LS | 004000 | Low speed auto baud detect |
| LCB.TM | 00177 | Timer |

### A.10.2 LC.STA -- State Of Modem Control

| Symbol | Value | Description |
|---|---|---|
| LCS.VG | | Virgin state |
| LCS.RG | 002 | Ringing |
| LCS.CD | 004 | Carrier Detect |
| LCS.CS | 006 | Carrier detect staisfied (2 second wait) |
| LCS.AB | 010 | Auto Bauding |
| LCS.RU | 012 | Running unconnected (No -10 connected) |
| LCS.RW | 014 | Running waiting for -10 connection to finish |
| LCS.RC | 016 | Running and connected to -10 |
| LCS.LC | 020 | Carrier lost |
| LCS.HA | 022 | Want to hang up |
| LCS.HG | 024 | Hung up, carrier off for 18 seconds |
| LCS.DL | 026 | Dialer is running |
| LCS.DS | 030 | Dialout succeeded |
| LCS.DF | 032 | Dialout failed |
| LCS.MX | 032 | Maximum state number for LC.STA |

### A.10.3 LC.SPD -- Codes For Transmit And Receive Speeds

| Speed | Code | Speed | Code |
|---|---|---|---|
| 50 | 1 | 1800 | 12 |
| 75 | 2 | 2200 | 13 |
| 110 | 3 | 2400 | 14 |
| 134.5 | 4 | 3600 | 15 |
| 150 | 5 | 4800 | 16 |
| 200 | 6 | 7200 | 17 |
| 300 | 7 | 9600 | 20 |
| 600 | 10 | 19200 | 21 |
| 1200 | 11 | EXTERNAL-A | 22 |
| | | EXTERNAL-B | 23 |

A.11 PHASE II LINK ENTRY BLOCK

Description: Contains status and control information for the DECNET compatible port connections. Used with DECNET version 2.

Defined in: DNNSP3.P11

```
        +----------------------------------------------------------------+
LE.STS* |                          Status bits                            |
        |                   (see descriptions below)                      |
        |LES|LES|LES|LES|LES|LES|LES|LES|LES|LES|LES|LES|   |   |   |    |
        |.NR|.MR|.MD|.DR|.LS|.DN|.DA|.LN|.LA|.DS|.DV|.DC|   |   |   |    |
        |----------------------------------------------------------------|
LE.LNK  |                   Next LE block for this node                  |
        |                                                                |
        |----------------------------------------------------------------|
LE.SCB  |                   Pointer to SCB for this node                 |
        |                                                                |
        |----------------------------------------------------------------|
LE.NCL  |                    Link address NCL is using                   |
        |                                                                |
        |----------------------------------------------------------------|
LE.DCP  |         Link address NCL and NSP think is destination          |
        |odd byte is incremental number            even byte is LEB numb|
        |----------------------------------------------------------------|
LE.NSP  |                    Link address NSP is using                   |
        |                                                                |
        |----------------------------------------------------------------|
LE.DPN  |           Destination device that NCL is asking about          |
        |                                                                |
        |----------------------------------------------------------------|
LE.LIL  |                    Last input LS/INT message                   |
        |                                                                |
        |----------------------------------------------------------------|
LE.LID  |                     Last input data message                    |
        |                                                                |
        |----------------------------------------------------------------|
LE.LOL  |                    Last output LS/INT message                  |
        |                                                                |
        |----------------------------------------------------------------|
LE.LOD  |                    Last  output data message                   |
        |                                                                |
        |----------------------------------------------------------------|
LE.OQL  |                  List header of LS/INT message                 |
        |               which have been output but not ACKed             |
        |----------------------------------------------------------------|
LE.OQD  |                  List header of data messages                  |
        |               which have been output  but not ACKed            |
        |----------------------------------------------------------------|
```

```
          |---------------------------------------------------------------|
LE.BUF    |                     Buffer for this link                      |
          |                                                               |
          |---------------------------------------------------------------|
LE.IIK    |                 Last input intercept ACKed                    |
          |                                                               |
          |---------------------------------------------------------------|
LE.IDK    |                   Last input data ACKed                       |
          |                                                               |
          |---------------------------------------------------------------|
LE.ODK    |                  Last output data ACKed                       |
          |                                                               |
          |---------------------------------------------------------------|
LE.ODS    |                   Last output data sent                       |
          |                                                               |
          |---------------------------------------------------------------|
LE.STT*   |     Reason to send to NSP     |      Current link state       |
LE.RSN    |        for disconnect         |         (see below)           |
          |---------------------------------------------------------------|
LE.MDR    | Outstanding data requests for |Maximum number of data requests|
LE.ODR    |          this link            |         for this link         |
          |---------------------------------------------------------------|
LE.TIM    |                               |      Logical link timer       |
          |                               |                               |
LE.SIZ=>+-----------------------------------------------------------------+
```

A.11.1 LE.STS -- Status Bits

| Symbol | Value | Description |
|---|---|---|
| LES.DC | 000020 | Have to send DISCONNECT |
| LES.DV | 000040 | Connection is to a device, not a task |
| LES.DS | 000100 | Connection is being broken |
| LES.LA | 000200 | Have to send LS/INT ACK |
| LES.LN | 000400 | Have to send LS/INK NAK |
| LES.DA | 001000 | Have to send DATA ACK |
| LES.DN | 002000 | Have to send DATA NAK |
| LES.LS | 004000 | Have to send LS to request an intr. message |
| LES.DR | 010000 | Have to send extra DATA REQUEST message |
| LES.MD | 020000 | Set if in middle of a dialog messsage |
| LES.MR | 040000 | Set if other side is requesting a message |
| LES.NR | 100000 | Set if other side is doing no requesting |

A.11.2 LE.STT -- Link State Code

| Symbol | Value | Description |
|---|---|---|
| LES.ID | 000 | Idle |
| LES.LI | 002 | NCL is trying to initialize a logical link |
| LES.PI | 004 | NSP is trying to initialize a logical link |
| LES.RN | 006 | Link is setup |
| LES.DS | 010 | Trying to disconnect |

## A.12 STATION CONTROL BLOCK

Description: Contain status information pertaining to each node in the network. One such block for each node. Referenced by "SCB#" where "#" is the station number (0,1,2,...).

Defined in: DNCNFG.P11

```
        +---------------------------------------------------------------+
SB.FLG* |                        Station flags                          |
        |                         (See below)                           |
        |SF.|SF.|SF.|SF.|SF.|SBF|SBF|SBF|SBF|SF.|SBF|SBF|SBF|SF.|SBF|SBF|
        |NSP|FAK|MCR|XRC|XCN|.NQ|.SQ|.SK|.NK|XAK|.RR|.RP|.NB|HID|.IC|.IU|
        |---------------------------------------------------------------|
SB.HXN  |     Last ACK processed       |       Highest NCL message      |
SB.LAP  |                              |        number transmitted      |
        |---------------------------------------------------------------|
SB.HAR  |   Receive message number     |       Highest ACK received     |
SB.RMN  |                              |                                |
        |---------------------------------------------------------------|
SB.TIM  |                              |              Timer             |
        |                              |       for REPs and STARTs      |
        |---------------------------------------------------------------|
SB.IMQ  |                 Input message queue address                   |
        |                 (messages not yet in order)                   |
        |---------------------------------------------------------------|
SB.OMQ  |                 Output message queue address                  |
        |            (ACKed by DDCMP, but not yet by NCL)               |
        |---------------------------------------------------------------|
SB.SQS  /                 Sequential node control area                  /
        |             (SEQLIM-1 blocks of SB.SQS bytes each)            |
        |---------------------------------------------------------------|
SB.LBA  |                  Address of LB for station                    |
        |                                                               |
        |---------------------------------------------------------------|
SB.LVL  |                 Cost of best path to this node                |
        |                                                               |
        |---------------------------------------------------------------|
SB.RTN  |               Return address over call to "MARK"              |
        |                                                               |
        |---------------------------------------------------------------|
SB.RRO  |                  Saved R0 over call to "MARK"                 |
        |                                                               |
        |---------------------------------------------------------------|
SB.RSB  |                  Saved SB over call to "MARK"                 |
        |                                                               |
        |---------------------------------------------------------------|
```

```
           |---------------------------------------------------------------|
SB.WOQ     |               Queue of messages waiting for                   |
           |                message number assignment                      |
           |---------------------------------------------------------------|
SB.NGH     /                     Neighbors list                            /
           |                  ((NGHMAX*2)+1 words)                         |
           |---------------------------------------------------------------|
SB.SNM     /                   Station name (ASCII)                        /
           |                     (SNMSIZ bytes)                            |
           |---------------------------------------------------------------|
SB.SID     /                   Software id (ASCII)                         /
           |                     (SIDSIZ bytes)                            |
           |---------------------------------------------------------------|
SB.DAT     /                  Software date (ASCII)                        /
           |                     (DATESZ bytes)                            |
           |---------------------------------------------------------------|
SB.NNM     |                      Node number                              |
           |                        binary                                 |
SB.SIZ=>+---------------------------------------------------------------+
```

## A.12.1 SB.FLG -- Station Flags

| Symbol | Value | Description |
|---|---|---|
| SBF.IU | 000001 | SCB is in use (for TENSCB means port enabled) |
| SBF.IC | 000002 | Station in contact (exchanged NCL START/STACK) |
| SF.HID | 000004 | We have the node ID for station |
| SBF.NB | 000010 | Need to send a NEIGHBORS message to node |
| SBF.RP | 000020 | A REP to this station is outstanding |
| SBF.RR | 000040 | We owe a response to this station's REP |
| SF.XAK | 000100 | We need to send an NCL-ACK to this station |
| SBF.NK | 000200 | We need to send an NCL-NAK to this station |
| SBF.SK | 000400 | We need to send an NCL-STACK to this station |
| SBF.SQ | 001000 | This station is a sequential node |
| SBF.NQ | 002000 | A request is in the NCL queue for this station |
| SF.XCN | 004000 | We need to send a CONFIGURATION message to node |
| SF.XRC | 010000 | We need to request a CONFIG message from node |
| SF.MCR | 020000 | This station has a command decoder |
| SF.FAK | 040000 | Future ACK - ACK received but DDCMP not done |
| SF.NSP | 100000 | NSP line (depends on FT.DCP) |

Notes:

SB.NGH format for each neighbor's entry is the SCB address followed by the Link Level.

## A.13 TASK BLOCK

Description: Contain pointers and status information pertaining to tasks using the DECNET compatible port. One such block for each task. Used for DECNET Phase II only.

Defined in: DNTSK.P11

```
            +----------------------------------------------------------------+
TK.STS*     |                   Task status information                      |
            |                        (see below)                             |
            |TK.|TK.|TK.|TK.|TK.|TK.|   |   |   |   |   |   |   |   |   |   |
            |RUN|LGI|TRG|WAK|NOP|SLP|   |   |   |   |   |   |   |   |   |   |
            |----------------------------------------------------------------|
TK.LNK      |                    Link to next TK block                       |
            |                                                                |
            |----------------------------------------------------------------|
TK.RQL      |                        Run queue link                          |
            |                                                                |
            |----------------------------------------------------------------|
TK.PRI      |                  Pointer to priority queue                     |
            |                                                                |
            |----------------------------------------------------------------|
TK.JSA      |                  Address to continue task                      |
            |                                                                |
            |----------------------------------------------------------------|
TK.RSA      |                  Address to start task at                      |
            |                    on a system restart                         |
            |----------------------------------------------------------------|
TK.PDL      |                 Address of push down list                      |
            |                                                                |
            |----------------------------------------------------------------|
TK.PDL+2    |               Current push down list pointer                   |
            |                                                                |
            |----------------------------------------------------------------|
TK.TIM      |                       Seconds timer                            |
            |                                                                |
            |----------------------------------------------------------------|
TK.QTM      |                               |          Quantum time          |
            |                               |           in jiffies           |
            |----------------------------------------------------------------|
TK.TPC      |                Address to goto when clock                      |
            |                         goes off                               |
            |----------------------------------------------------------------|
TK.DTK      |               Copy of DB.TSK for current SVC                   |
            |                                                                |
            |----------------------------------------------------------------|
```

```
          |-------------------------------------------------------------|
TK.SPT    |                     Send queue putter                       |
          |                                                             |
          |-------------------------------------------------------------|
TK.STK    |                     Send queue taker                        |
          |                                                             |
          |-------------------------------------------------------------|
TK.SQU    /                         Send queue                          /
          |                      (TKSQSZ words)                         |
          |-------------------------------------------------------------|
TK.ARG    |               Save value to return to caller                |
          |                                                             |
TK.SIZ=>+-------------------------------------------------------------+
```

A.13.1 TK.STS -- Task Status

| Symbol | Value | Description |
|---|---|---|
| TK.RUN | 100000 | Task is runnable |
| TK.LGI | 040000 | Task is "Logged In" (has not EXITed yet) |
| TK.TRG | 020000 | Task has been triggered by another task |
| TK.WAK | 010000 | Some even has woken this task |
| TK.NOP | 004000 | Device was not OPENed on this call |
| TK.SLP | 002000 | Task is sleeping (in timer queue) |

## A.14 TO-11 BLOCK

Description: Contain pointers and control information pertaining to messages from the 10 to the 11. Used only in DN87S and DN20 systems. Note that these reside in chunks, and that the following definitions redefine the meanings of some entries in the standard chunk.

Defined in: DNDTE.P11

```
         +----------------------------------------------------------------+
TE.LNK   |                Address of next chunk in message                |
         |                                                                |
         |----------------------------------------------------------------|
TE.QPR   |             Number of bytes left to transfer in                |
         |                   queued protocol message                      |
         |----------------------------------------------------------------|
TE.LEN   |                   Total length of message                      |
         |                                                                |
         |----------------------------------------------------------------|
TE.FFW   |          Copy of first word for current QPR message            |
TE.LIN*  |              (line number and flags in hi byte)                |
         |   |   |   |   |   |   |   |FW.|   |   |   |   |   |   |   |    |
         |   |   |   |   |   |   |   |MOR|   |   |   |   |   |   |   |    |
         |----------------------------------------------------------------|
TE.CNK   |                  Space left in current chunk                   |
         |                                                                |
         |----------------------------------------------------------------|
TE.QHD   |               Queued protocol message header:                  |
         |                   length of first message                      |
         |----------------------------------------------------------------|
TE.QFN   |                     QPR word:  function                        |
         |                                                                |
         |----------------------------------------------------------------|
TE.QDV   |                QPR word:  device  (will be NCL)                |
         |                                                                |
         |----------------------------------------------------------------|
TE.QSP   |                            Spare                               |
         |            (start of user data for compression)                |
         |----------------------------------------------------------------|
TE.ADR   |               Address of where to put data of                  |
         |                       next fragement                           |
TE..SZ=>+----------------------------------------------------------------+
```

### A.14.1 TE.LIN -- QPR Message

| Symbol | Value | Description |
|---|---|---|
| FW.MOR | 000400 | There will be another QPR message in NCL message |

## A.15 TO-10 BLOCK

Description: Contain pointers and control information pertaining to messages from the 11 to the 10. Used only on DN87S and DN20 systems. Note that these reside in chunks, and that the following definitions redefine the meanings of some entries in the standard chunk.

Defined in: DNDTE.P11

| | | |
|---|---|---|
| TT.FLK | Forward link | |
| TT.RLK | Backward link | |
| TT.ALC | Space allocated for this block | |
| TT.HDL | Length of header | |
| TT.QHD | First queued protocol (QPR) word: length of header | |
| TT.QFN | QPR word: function | |
| TT.QDV | QPR word: device (will be NCL) | |
| TT.QSP | Spare (unused) | |
| TT.QFW | Line number (first word) | Indirect message length |
| TT.ADR | Address of real data | |
| TT.USR | User supplied data | |
| TT.EFN | Event flag number | |