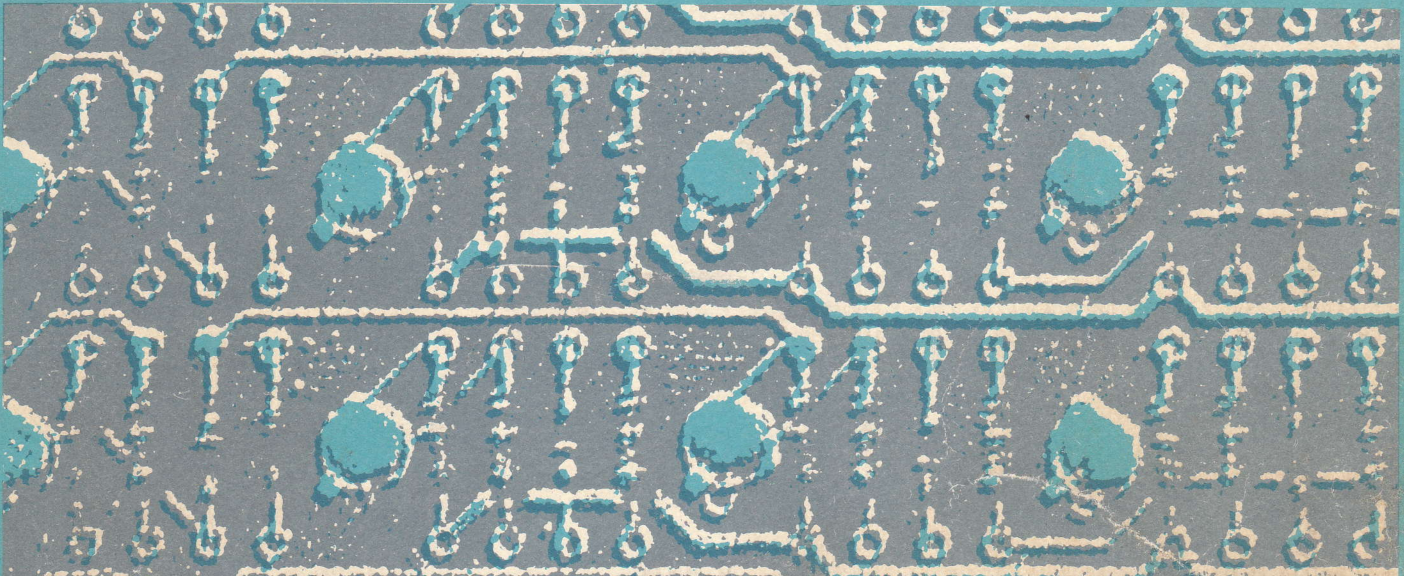
P800M Programmer's Guide 3

Volume IV : Trouble Shooting Guide



PHILIPS Data Systems

PHILIPS

P800M Programmer's Guide 3

Volume IV: Trouble Shooting Guide

A Publication of:
Philips Data Systems,
Department SSS, T & D,
P.O. Box 245, 7300 AE Apeldoorn, The Netherlands.

Publication Number 5122 991 28475

January 1983

PREFACE

While every care has been taken in the preparation of this book, some errors may remain. Should the reader find an error or omission, or have any other comment to make, he is invited to contact:

SSS, Training and Documentation,

at the address on the opposite page. A form is provided at the end of this book, for the user`s convenience.

TABLE OF CONTENTS

page

INTRODUCTION

This trouble shooting guide is intended for use by support programmers and systems managers in conjunction with, and after a fair appreciation of, the associated manuals in this set. These include:

P800M Programmer's Guide 3 Vol. I (MAS Manual)
P800M Programmer's Guide 3 Vol. III (Software Processors)
P800M Programmer's Guide 1, 2 & 3 Vol II (Instruction Set)
P800M Data Communication User Manual

Its main purpose is to speed up the process of identifying and rectifying errors in system implementation and operation. It does not cover hardware errors (apart from giving a list of device hardware error codes); these should be referred to an engineer.

Definition of terms

It will be assumed that terms in common use in the other manuals will not need re-definition here, but any which are unusual or quite new will be defined as they arise.

Syntax

_ indicates one or more spaces.

<n> indicates that an item is to be substituted for n; n describes the item.

[n] indicates that 'n' is an optional item.

a|b The vertical bar is an exclusive OR and indicates that a or b, but not both, should be entered.

a an underlined parameter indicates a default value.

Chapter 1

MONITOR CONTROL TABLES

T:CVT The Communication Vector Table

This table points to the major system tables and some important system subroutines which are addressed directly by transient area routines using an offset from the beginning of the table.

The structure of the CVT is as follows:

LABEL	LOCATION	CONTENTS
T:CTIM	0	T:RTC : REAL TIME CLOCK TABLE ADDR.
T:CPLS	2	1 : CLOCK
	4	M:DISP : DISPATCHER ADDR.
T:CRST	6	R:RSET : NON-STANDARD CLOCK RESET VALUE
	8	M:LAB : SCHED.LABEL MANAGEMENT ROUTINE
	/A	T:PWT : DWT CHAIN POINTER
	/C	T:SLT : SLT ADDRESS
	/E	T:MCT : T:MCT ADDRESS
	/10	T:PCT : FIRST PCT ADDRESS IN SYSTEM MACHINE
CVTSS1	/12	: SYSTEM STATUS WORD
CVTSS2	/14	: Initial System Dynamic Area Size (reset to zero by INIMON, sysgen dependent)
	/16	B:POIN : B TIMER CHAIN POINTER ADDR.
	/18	C:POIN : C TIMER CHAIN POINTER ADDR.
	/1A	R:ABRT : ADDR. OF R:ABRT ROUTINE
CVTARS	/1C	M:ARES : AUTO RESTART ROUTINE ADDR.
	/1E	T:CORE : CORE ALLOCATION TABLE
CVTNBP	/20	: MAXIMUM No. OF PROGRAMS (Sysgen dependant)
	/22	T:DAD : DAD TABLE
	/24	R:DMAS : GET DYN AREA IN SYST MACHINE

LABEL	LOCATION	CONTENTS
	/26	R:DMLS : REL DYN AREA IN SYST MACHINE
	/28	S:FCL : FCL START ADDRESS
	/2A	R:ASY1 : ACTIVATE A PROGR.
	/2C	R:SEV : SET EVENT
	/2E	R:ALOC : ALLOC MEMORY
	/30	R:DLOC : DE - ALLOC MEMORY
	/32	7 : MAX. No. OF SCHED LAB (DEFAULT VALUE IN DCF)
	/34	9 : MAX No. OF SEGMENTS
	/36	-50 :-(MAX No. OF FILE CODES)
	/38	-10 :-(MAX No. OF BLOCK BUFFERS)
	/3A	R:UNSP : ADDRESS OF UNSUSPEND ROUTINE
	/3C	R:DEEV : ADDRESS OF DECREMENT EVENT COUNTS ROUTINE
	/3E	R:DESW : ADDRESS OF DECREMENT SWAP EVENT COUNTS ROUTINE
CVTBAT	/40	: PCT ADDR OF BATCH PCT
CVTSWN	/42	T:SWIN : ADDR. OF SWAP-IN TABLE
CVTQIN	/44	T:QIN : ADDR. OF QUEUE-IN TABLE
	/46	T:ELIG : ADDR. OF T:ELIG
CVTRMN	/48	: ADDR. 1ST PCT LOADED IN DLA
	/4A	-3 : DEFAULT VALUE OF MIN RES TIME IN DLA
	/4C	R:SPND : ADDR. OF R:SPND, SUSPEND A PROGRAM
CVTSUP	/4E	0 : LAST ADDRESS OF SUPERVISOR +2
	/50	R:PUTW : ADDR. OF R:PUTW,PUT A PROG IN WAIT
CVTSWP	/52	0 : TIMER INTERRUPT COUNT
	/54	R:FSWP : ADDRESS OF FORCE SWAP-OUT ROUTINE

LABEL	LOCATION	CONTENTS
	/56	R:RKLM : ADDRESS OF LKM INIT. ROUTINE
	/58	R:REAC : ADDRESS OF REACTIVATION ROUTINE
	/5A	R:DUMP : DUMP SYSTEM ROUTINE
	/5C	T:DCT : ADDR. OF FIRST DCT
	/5E	T:LCB : ADDR. OF FIRST LCB
	/60	T:SCT : ADDR. OF FIRST SCT
	/62	20 : DEFAULT FOR CONS LC
CVTFDC	/64	ADDR. OF FDC CHAIN
	/66	ADDR. OF TRT CHAIN
	/68	ADDR. OF WORK STORE FOR TDFM
	/6A	: MAXIMUM NUMBER OF BUFFERS FOR TDFM (SYSGEN DEPENDENT)
	/6C	R:INSW : INCR. EXIT AND EVENT COUNT
	/6E	R:SEV2 : 2ND ENTRY FOR R:SEV
	/70	R:INEV : INCR EVENT COUNT
	/72	R:SEV1 : 1ST ENTRY FOR R:SEV
CVTDLK	/74	TRC110 : ADDR. OF DATACOM RTC INT. ROUTINE
	/76	20 : DEFAULT No. OF DTC LINE CODES
CVTSPT	/78	T:SPT : SPOOL TABLE
CVTBPA	/7A	0 : TDFM DISC. BUFF. QUEUE
CVTBOQ	/7C	0 : BACK-OUT QUEUE
CVTLOA	/7E	0 : LOCK QUEUE ADDRESS
	/80	R:HALT : SYSTEM ERROR ROUTINE
CVTMBX	/82	MAIL BOX : ADDRESS
CVTSLM	/84	0 : SEC LOAD MODULES

LABEL	LOCATION	CONTENTS
CVTDTO	/86	0 : POINTER TO DWT FOR TIME-OUT (/2C in DWT)
	/88	R:EXIT : EXIT ROUTINE
	/8A	0 :
CVTSDA	/8C	0 :
	/8E	R:EL : EL SIMULATION ROUTINE
	/90	R:ES : ES SIMULATION ROUTINE
	/92	R:MVSU : MVSU SIMULATION ROUTINE
	/94	R:MVUS : MVUS SIMULATION ROUTINE
	/96	R:MCHU : MOVE CHARACTER IN A BUFFER
	/98	R:ACTD : R:ACTD ACTIVATION ROUTINE
	/9A	R:TRCE : TRACE ROUTINE
CVTSEG	/9C	T:SEG : SEGMENT TABLE IN EXTENDED AREA
CPUTYP	/9E	0=857, 2=858, 4=859, 6=854, 8=871
	/A0	0 : SIZE OF ESDA (NB.PAGES)
	/A2	R:GESA : GET BUF IN ESDA
	/A4	R:RESA : REL BUF "
	/A6	R:MODE : SET EXTENDED MODE
	/A8	R:MODA : SET ABSOLUTE MODE
CVTERL	/AA	0 : ERLOG ANCHOR
	/AC	R:ACT
	/AE	R:AMPN
	/B0	R:ACPA
	/B2	R:APME

The format of the more important of these locations is as follows:-

CVTTS1, The System Status Word:

bit 0 = 1 Dispatcher has to save Flt.pt registers.
= 0 Dispatcher does not have to save Flt.pt. registers.

bit 2 = 1 HD command received

bit 3 = 1 Interrupt control panel is being processed, refuse further interrupt.
= 0 Control panel interrupt can be accepted.

bit 11 = 1 Hardware floating point provided with CPU
= 0 No hardware fl.pt option in CPU.
(bit 0 set to 1 when bit 11 = 1 and FON [only 1 or no program uses flt.pt] bit 0 set to 0 when either bit 11 = 0 or FOF)

bit 15 = 1 Automatic restart routine has to be called by the dispatcher (a power failure occured, next interrupt at level zero will be Auto Restart).

bit 14 = 1 Auto Restart routine selected at Sysgen.

bit 13 = 1 A halt requested at Auto Restart, giving the operator enough time to switch on all the devices.

CVTSS2 contains the system dynamic area size. Reset to zero after initialization.

T:MCT The Machine Control Table

One of these exists for each machine declared and gives all the necessary information about a machine, particularly, the addresses of the first PCT in the chain of PCT's for the machine, the first FCT and the start of the dynamic area.

Th MCT's for each machine declared are forward chained, the first in the chain being for the machine called SYSTEM. This is pointed to by location /E of the CVT (CVTMCT).
They are also pointed to by each PCT in the machine (location PCTMCT of the program control table).

The layout of the MCT is as follows:

LOCATION	LABEL	CONTENTS
0		Address of next MCT in the Chain, 0 if last
2	MCTNAM	
4		MACHINE NAME
6		
8	MCTPCT	Address of the first PCT of the machine (0 if none)
/A	MCTFCT	Address of first entry in FCT chain
/C	MCTSLM	Max. No. of sched. labels (default value for FCL)
/E	MCTDYN	Base (lowest) address of Dyn.area (virtual address)
/10	MCTSEG	Core resident seg. table address
/12	MCTSP	0 or address of the first PCT suspended because of dyn. area overflow
/14	MCTKIN	QCB Address (0 if none)
/16	MCTSTA	machine status
/18	MCTMCT	MCT address

LOCATION	LABEL	CONTENTS
/1A	MCTECB	0 | 0 | 1 | File code (E0, EE, 1)
/1C	MCTEBF	@MWA MACHINE WORKING AREA ECB used to read FCL
/1E	MCTERL	Commands
/20	MCTEEL	
/22	MCTEST	
/24		0
/26	MCTFCW	Address of event on which FCL waits
/28	MCTENT	entry point FCL has to activate
/2A	MCTSPW	address of first PCTLNK in wait, 0 if none
/2C	MCTMFC	-(maximum number of user file codes)
/2E	MCTRTM	-(minimum resident time in dyn.load area) default value
/30	MCTMBF	-(maximum number of blocking buffers on the machine)
/32	MCTLCT	Address of the first LCT; zero if none
/34	MCTSEM	Semaphore Address (Not in system machine)
/36	MCTDWD	Data Window Address (Not in system machine)
/38	MCTMDG	@ of first entry ~~Not used~~ in T:MDG (Not in system machine)

The labels used have the following meanings:

MCTLNK Points at the next entry in the chain, 0 if it is the last one.

MCTNAM 3 words, containing the machine name, left justified, filled with spaces.

MCTPCT Address of the first PCT in the machine.

MCTFCT Address of the first entry in file code table of the machine.

MCTSLM Defines the default value for number of scheduled labels of a program.

MCTDYN Points at the base address of the dynamic area of the machine.

MCTSP Address of the first program suspended because of dynamic area overflow, 0 if none. For the system machine, this is the PCT address of a system program suspended because of a DA overflow.

MCTSTA Defines the status of the machine as follows:

MCTSTA	Ev	0	Sm	Fg	Bg	Mr	Eo	Pc	Gm				Mg	RTN		
bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Ev = 0 Event occured
Sm = 1 System machine
Fg = 1 Foreground machine
Bg = 1 Backgrnd machine
Mr = 0 FCL is running (SM rejected)
(set by SM, reset by BYE)
Eo = 1 if /EO is assigned to device like TY, display i.e. print 'FCL:'
before reading FCL command
Pc = FCL under a catalogued procedure
Gm = Machine being defined (DCF DCB) refuse DCF or DCB on that machine
RTN = Return code, used by FCL and its commands
0 Read next command, activate /040B or MCTENT entry
1 Exit
2 Wait for MCTFCW, then activate MCTENT entry point
3 Activate a program, wait for sched.lab. (RUN)
4 Read correction on 01
5 Read correction in a subcommand (DCF, DCB)
6 Read a subcommand (in DCF or DCB)
Mg = 0 Middle ground programs are allowed
Mg = 1 No middle ground program

MCTMCT Address of the MCT, used for activation or I/O request.

MCTECB A block of 6 words used as ECB for I/O requests.

MCTEBF Is the buffer address, or to be exact, the address of the first word of the machine work area MWA, a zone of 140 characters, allocated in the dynamic area to read and process FCL for that machine.

MCTFCW Address of the event on which FCL has to wait for completion before either reading next command (if MCTENT=0) or before activating X:MASG, entering MCTENT, to process the event.

MCTENT Defines the entry points that FCL has to activate.

MCTSPW Is the address of the first PCT waiting for an event in the machine. 1 if no program is in wait state.

MCTMFC Is the maximum number of User file codes that can be assigned in the machine. It is a negative value, used to control assign request. In fact, this limit is a protection for other machines because an excessive assignment may cause an overflow in the System Dynamic Area.

MCTRTM Defines the default value for the minimum resident time of a disc-resident program.

MCTMBF Gives the maximum number of blocking buffers usable in the machine. It is a protection for other machines. Indeed, an excessive use of blocking buffers may cause an overflow in the system dynamic area, and the whole system hangs up.

MCTSEM Address of first Semaphore Block.

MCTDWD Address of first Data Window (obtained by LKM 56).

MCTSEG Address of the core-resident segment table of the machine which is described on the following page.

MCTKIN Address of the first queue control block, 0 if none.

MCTMDG Address of the first entry in the T:MDG chain of the machine

The Memory-Resident Segment Table

This is created in the system dynamic area when the foreground machine is declared. It is used to define the memory allocation of core-resident program. The location MCTSEG (/10) of each foreground machine control table points to the first word of its core-resident segment table.

The layout of the core-resident segment table can be shown diagrammatically thus:

```
                                   ---------------------------
(MCTSEG) ------>                  | Highest segment number    |
                                  |---------------------------|
                                  | Highest virtual address   |
                                  | in seg.0 (always /FFFE)   |
                ------>         T |---------------------------|
(PCTMMU)                        | |                           |
of programs                     | | These words define the    |
loaded in                       | | page allocated to         |
Seg. 0                          | | segment 0.                |
                16 words        | |                           |
                                v |                           |
                                - |---------------------------|
                                  | Highest virtual address   |
                                  | in seg. 1.                |
                ------>         T |---------------------------|
(PCTMMU)                        | |                           |
of programs                     | | These words define the    |
loaded in                       | | pages allocated to        |
Seg. 1                          | | segments 0 and 1.         |
                16 words        | |                           |
                                v |                           |
                                - |---------------------------|
                                  | Highest virtual address   |
                                  | in seg. 1.                |
                ------>         T |---------------------------|
(PCTMMU)                        | |                           |
of programs                     | | These words define the    |
loaded in                       | | pages allocated to        |
Seg. 2                          | | segments 0 and 2          |
                16 words        | |                           |
                                v |                           |
                                -  ---------------------------
```

Remarks:

- The cost in core can thus be calculated as follows:

  C = 18 + 17/n

  where n = the No. of segments

- For the batch machine, MCTSEG = 0, however, an MMU save area is created for the batch program. If the machine is core-resident, 16 words are created and contain the addresses of pages allocated to the machine. If the machine is disc-resident, then the table (18 words) is created as for any disc-resident program.

- For a disc-resident program, when the program is loaded, an MMU table is created (in the dynamic area) and pointed to by PCTMMU.

T:PCT Program Control Table

In a foreground machine, one of these exists for each program declared using a LOD, SWP, RON, REP command, for each active REP task and for an active middleground program. The FCL task in a foreground machine is chained in the system machine.

In the background machine, there is one PCT and it is followed by the JPT. Location /40 (CVTBAT) of the CVT points to the PCT of the batch program.

The PCT's are chained in various ways e.g.:

Location /48 of the CVT points to the first PCT of a chain of PCT's for all programs loaded in the dynamic loading area. Location /42 of each PCT points to the next in this chain.

Location /10 of the CVT points to the chain of all PCT's in the system machine.

Location /8 of an MCT points to the chain of all PCT's for that machine. Location 0 of each PCT points to the next PCT.

Location /12 of an MCT points to the chain of PCT's within that machine which are suspended because of dynamic area overflow.
Location /24 of each PCT points to the next in this chain.

Location /2A of an MCT points to the chain of PCT waiting for an event.
Location /26 of each PCT points to the next PCT in this chain.

Location /28 of the DWT points to the chain of PTC's awaiting attachment to a device. Location /24 of each points to the next in this chain.

The entry P:CUR in the system MAP contains a pointer to the current PCT.

LOCATION	LABEL	CONTENTS
0	MCTPCT PCTLNK	Address of next PCT in the chain, 0 if last
2	PCTNAM	Program name
4		
6		
8	PCTSAD	Start address of the Program (virtual)
/A	PCTSAV	Register's save area address
/C	PCTMMU	MMU save area address
/E	PCTSTA	Program Status. If ≠ 0, program is not eligible
/10	PCTMOD	Program characteristics
/12	PCTLEV	Software level of program (bit 0 = 1 if not connected)
/14	PCTMSE	ECB main sequence waits (virtual)
/16	PCTACT	ECB address of activating program (virtual)
/18	PCTPCT	PCT addr. of activating program
/1A	PCTLAB	Sched. label Address
/1C	PCTRQQ PCTMOT	Activation request queue Mother PCT for re-entrant programs
/1E	PCTEVC	Event count
/20	PCTSEC	swap event count
/22	PCTMCT	address of MCT of the program
/24	PCTSP	0, or address of next PCT suspended on the same lack of resource
/26	PCTWT	Adress of next PCTLNK in wait, 0 if last or not in wait

The following words of PCT are used only with user programs:

LOCATION	LABEL	CONTENTS
/28	PCTLMD	NR | | Load module : DAD file code
/2A	PCTLMS	N|Sector No. of load module in DAD
/2C	PCTREG	Core region size (No. of pages)/ or ending address if core resident
/2E	PCTSW1	Initial image of swappable (or read only) program on D : CI
/30	PCTSWN PCTDAU	Current swap area address in D : CI (swappable, Mid., Bg program) Daughter PCT for reentrant programs
/32	PCTLAD	Program load address (beginning)
/34	PCTSSA	Scheduled label save area address
/36	PCTSLE	ECB on which sched lab waits
/38	PCTKAB	0 or address of keep control on abort or Flt. point interrupt or read key in parameter block
/3A	PCTREC	Scheduled label save area address or (– # of reentrances for REE prog.)
/3C	PCTMOV	Address of records to be moved to swapped-in user program
/3E	PCTIRT	- (initial value of minimum resident time)
/40	PCTCRT	- (current value of resident time)
	PCTMAC	MCT address of activating program
/42	PCTRNX	Next PCT address connected to resident time chain (disc resident program), 0 if last

A detailed explanation of these entries follows:

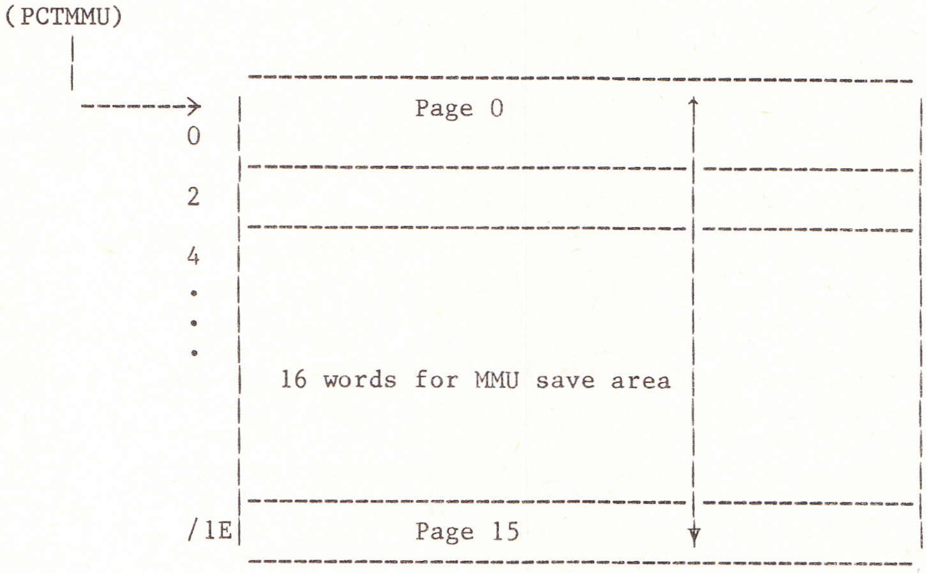
PCTLNK The address of the next PCT in the machine. 0 if it is the last entry.

PCTNAM 6 characters, identifying the name of the program.

PCTSAD Program start address. It is a virtual address to be loaded into the P register when the program is started. The bit 15 is reset to zero.

PCTSAV Points at the save area of the main sequence.

PCTMMU Points at the save area for MMU registers. For core-resident User programs, it points at the core-resident segment table entry SEGMMU. For disc-resident user programs, it points at a save area created dynamically in the system dynamic area and is constructed as follows:



PCTSTA Program status: Program not eligible if non-zero.

A	Ab	L	P	W	Nl	Ex	Mg	Wsl			Mw	Sp	Sa	C	Sr
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

A = 1 Inactive
A = 0 Active (set by activate)
Ab = 1 Program is aborted
Ab = 0 not aborted
L = 0 Program loaded
L = 1 Program not loaded
P = 1 Program in pause (main or Sch. Lab.)
W = 1 Program in wait (main sequence)
Nl = 1 Program not loadable (I/O err. on disc).
WSl= 1 Sched. Lab. in wait
Sa = 1 Program suspended, being swapped
C = 1 Suspended because a supervisor call (LKM) is being processed.
Sr = 1 Suspended because of resource default (attach, get buffer ...)
Sp = 1 Spool bit (background)
Mw = 1 Program in multiple wait
Ex = 1 Fatal exit (exit code = -1); ignore all scheduled labels
Mg = 1 Middleground program has exited.

PCTMOD:

S	E	Sl	RO	SW	C	Md	Re	SP	B	L	Ts	SC	Rb	Bl	Bs
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

S = 1 System program, no check on PCTEVC for exit
E = 1 Program has issued exit macro but not effective because event count> 0. When PCTEVC = 0, the module which decrements PCTEVC has to activate exit module
SL = 1 Sched. Labels to be dispatched
RO = 1 Read only program
SW = 1 Swappable program
C = 1 Core-resident program
Md = 1 Middle-ground program
Re = 1 Re-entrant program (set by REP command)
Rb = 1 PCT to be deleted, this PCT is created by activate, for a re-entrant program (only one of these 2 bits set)
Sp = 1 Program swapped
B = 1 Background program (which can be swappable or core-resident)
L = 1 Limits sets (No. of cards, lines, sec...)
(cf. JPT table)
Ts = 1 Program to be swapped
Set to 1 when the system decides to swap out this program but the PCTSEC > 0. When the PCTSEC reaches zero, the modules which decrement PCTSEC must activate swap out program when Ts = 1.
Sc = 1 Sched. lab. is running. Set by dispatcher
Reset when sched. label exits.
Bl = 1 being loaded
Bs = 1 being swapped out

PCTLEV Software level of the program. If the bit 0 is set, the program is disconnected or not connected to a level.

PCTMSE Address of the event on which the main sequence waits.

PCTACT, PCTLAB are respectively the activation block (A8), PCT address, and scheduled label of the activating program when this one performs the activation request. These 3 words allow the system to start the scheduled label of the activating program when the current program issues the exit macro.

If PCTPCT = 0, then the program is a system program; and if PCTACT # 0, then it contains the system ECB.

PCTRQQ Activation request queue, 0 if none, as follows:

```
(PCTRQQ)
   |
   |      ---------------------------------------------------
   -->   |    Address of next request block, zero if none    |
         |---------------------------------------------------|
         |         A2                                        |
         |---------------------------------------------------|
         |         A3 of calling program                     |
         |---------------------------------------------------|
         |         A4                                        |
         |---------------------------------------------------|
         |         A5 (PCT address of activating program)    |
         |---------------------------------------------------|
         |         A6 (Scheduled label address)              |
         |---------------------------------------------------|
         |         A7 (PCT of activated program)             |
         |---------------------------------------------------|
         |         A8 of calling program                     |
         |---------------------------------------------------|
         |         A9 = ((A8)+2)  i.e. parameter word placed |
         |                        in A4 at activating time   |
          ---------------------------------------------------
```

PCTLMS Sector address of load module in the DAD (PCTLMD + 1).

N = 0 Non-consecutive granules
N = 1 Consecutive granules

PCTLMS points at the second sector of the first granule of the file, i.e. for non consecutive file, it points at the sector GRANTB.

The first sector of the program is the (PCTLMS) + 1.

PCTKAB   points at the Keep Control on Abort, Keep Control on Flt.Pt. Error, or Read Key-In requests for the program.

```
(PCTKAB)
   |
   |              -----------------------------------------------
   '-->      0   |   Address of next request block, 0 if last    |
                 |-----------------------------------------------|
             2   |   Block type                  |               |
                 |-----------------------------------------------|
             -   |                                               |
                 |                                               |
             -   |                                               |
             -   |                                               |
             -   |                                               |
             -   |                                               |
                 |                                               |
                 |                                               |
                 |                                               |
                 |                                               |
                  -----------------------------------------------
```

These request blocks obtained from the dynamic area of the system machine are chained together.
The second word defines the type of the request.

Block Type 4 = Keep control on abort
Block Type 5 = Keep control on floating point error
Block Type 6 = Read unsolicited key in
Block Type 7 = Set Event

For Abort / Flt. block, next 5 words are:

```
                 |                                               |
                 |-----------------------------------------------|
          4      |  User Abort / error recovery address (user A7) |
                 |-----------------------------------------------|
          6      |  User Abort / error block address (user A8)   |
                 |-----------------------------------------------|
          8      |  Status of Floating Unit / Abort code         |
                 |-----------------------------------------------|
         10      |                      IC                       |
                 |-----------------------------------------------|
         12      |                      PSW                      |
                 |-----------------------------------------------|
         14      |                     (A8)                      |
                  -----------------------------------------------
```

The last 4 words are used only for disc-resident programs (only for Abort) to save the context of the Abort when the program is swapped out. They will be restored to the program when the program is reloaded into core. (This block will be linked to the PCTMOV chain, and removed from PCTKAB).

For read in requests, the next words are:

```
          |                                                        |
          |--------------------------------------------------------|
    4     |  User ECB address (A8)                                 |
          |--------------------------------------------------------|
    6     |  Sched. label                                          |
          |--------------------------------------------------------|
    8     |  Effective length (set by OCOM) or special chars.      |
          |--------------------------------------------------------|
   10     | ↑                                                      |
          | |                                                      |
          | |                                                      |
          | |                                                      |
          | | Key-in message, only for disc-                       |
          | | resident program (used to record                     |
          | | the operator key-in, in order to                     |
          | | transfer it to the User program when                 |
          | ↓ it is reloaded into core)                            |
           --------------------------------------------------------
```

The words 10 and onwards are reserved only for disc-resident programs, in order to save the operator message when the program is swapped out. As with the Abort / Flt.block, this block can be linked to PCTMOV when the key-in is recorded in order to transfer it to the program when the latter is reloaded into core.

PCTMOV   is the chain of "events" to be set when the disc-resident program is reloaded into memory. It is used especially for an Abort-Key-in request which occurs when the program is swapped out, or for read/write requests when the ECB and/or buffer are in the CMA.

These blocks are to be released from the System Dynamic Area after setting the user's event.

```
(PCTMOV)
   |
   |          ----------------------------------------------
   -->    0  |   Address of next block, 0 if last           |
             |----------------------------------------------|
          2  |   Block type                |                |
             |----------------------------------------------|
          4  |                           7    8             |
             |----------------------------------------------|
          6  |                                              |
             |                                              |
             |            Other words (see PCTKAB)          |
             |                                              |
             |                                              |
             |                                              |
             |                                              |
             |                                              |
             |                                              |
              ----------------------------------------------
```

For block type 3 (Set Event), word 4 contains the event address.

PCTSAV and PCTSSA point to important save areas, which have the following layout:

```
                 Main sequence Save Area Address

(PCTSAV)
    |
    |             ---------------------------------------------
    |-->         |          IC                                  |
                 |---------------------------------------------|
                 |          PSW                                 |
                 |---------------------------------------------|
                 |          A1                                  |
                 |---------------------------------------------|
                 |          A2                                  |
                 |---------------------------------------------|
                 |          A3                                  |
                 |---------------------------------------------|
                 |          A4                                  |
                 |---------------------------------------------|
                 |          A5                                  |
                 |---------------------------------------------|
                 |          A6                                  |
                 |---------------------------------------------|
                 |          A7                                  |
                 |---------------------------------------------|
                 |          A8                                  |
                 |---------------------------------------------|
                 |          A9                                  |
                 |---------------------------------------------|
                 |          A10                                 |
                 |---------------------------------------------|
                 |          A11                                 |
                 |---------------------------------------------|
                 |          A12                                 |
                 |---------------------------------------------|
                 |          A13                                 |
                 |---------------------------------------------|
                 |          A14                                 |
                 |---------------------------------------------|
                 |          Floating Reg. 1                     |
                 |---------------------------------------------|
                 |          Floating Reg. 2                     |
                 |---------------------------------------------|
                 |          Floating Reg. 3                     |
                  ---------------------------------------------
```

```
              Sched. Lab. Area Address

(PCTSSA)
   |
   |            ---------------------------------------------
   -->        |        IC                                   |
              |---------------------------------------------|
              |        PSW                                  |
              |---------------------------------------------|
              |        A1                                   |
              |---------------------------------------------|
              |        A2                                   |
              |---------------------------------------------|
              |        A3                                   |
              |---------------------------------------------|
              |        A4                                   |
              |---------------------------------------------|
              |        A5                                   |
              |---------------------------------------------|
              |        A6                                   |
              |---------------------------------------------|
              |        A7                                   |
              |---------------------------------------------|
              |        A8                                   |
              |---------------------------------------------|
              |        A9                                   |
              |---------------------------------------------|
              |        A10                                  |
              |---------------------------------------------|
              |        A11                                  |
              |---------------------------------------------|
              |        A12                                  |
              |---------------------------------------------|
              |        A13                                  |
              |---------------------------------------------|
              |        A14                                  |
              |---------------------------------------------|
              |        Floating Reg. 1                      |
              |---------------------------------------------|
  122         |        Floating Reg. 2                      |
              |---------------------------------------------|
              |        Floating Reg. 3                      |
              |---------------------------------------------|
              |     Max. No. of sched. labels to be dispatched  |
              |---------------------------------------------|
  128         |  Current No. of sched. labels to be dispatched  |
               ---------------------------------------------
```

This table only exists if the max. No. of scheduled labels has been defined by means of the LAB or LOD commands, followed by 2 x (max. number of Sch. Lab to be dispatched) words. 1 entry = 2 words. These entries are upward shifted when a sched. lab. exits.

first sched. label to be dispatched
(A8)
second sched. label to be dispatched
(A8)
--
--
etc.

PCTLMD bits 0-3 (NR): No. of pages of the root segment for a swappable overlaid program.

bit 4-7: Unused.

bits 8-15: DAD file code Load Module.

T:JPT The Job Parameter Table

This immediately follows the PCT for the background machine and contains all the information necessary to control the running of the background job, including the default values for the BCP control commands.

The layout is as follows:

LOCATION	LABEL	CONTENTS	
0	JPTUID	USERID	
2		8 characters, left justified,	
4		filled with spaces	
6			
8	JPTLMT	Exec. time limit seconds (TIME) - Zero if none	
/A	JPTLML	Max. of printed lines (PRNT) - Zero if no limit	
/C	JPTLMR	Max.of punched records (PNCH) - Zero if no limit	
/E	JPTCNT	Curr. Elapsed time	
/10	JPTCNL	Current number of printed lines	
/12	JPTCNR	Current number of punched records	
/14	JPTDSK	DAD of current program	User JOB DAD logical address
/16	JPTDIR	User directory address within the JOB DAD	
/18	JPTMOD	Job characteristics (see later explanation)	
/1A	JPTMD2	Undefined	Value of ABCD in : STP
/1C	JPTPST	Address of current, BCL command in the CCT (Command Control Table) of BCP	
/1E	JPTBCP	N	Disc address of BCP (address in directory + 1 = GRANTB)
/20	JPTROT	N	Disc address of the program to be loaded (address in directory + 1 = GRANTB)
/22	JPTCOD	Current abort code	current exit code
/24	JPTMCD	Input file code of last command	Maximum error /exit code of the step (authorized)
/26	JPTFCE	File code used to re-read an erroneous command	Highest exit/error code encountered

LOCATION	LABEL	CONTENTS
/28	JPTSDI	System Directory Address in DAD/FO
/2A	JPTBRE	No. of pages required to load BCP
/2C	JPTPRE	No. of pages required to load User program

The next words comprise the save registers area, used to transmit the register contents of the previous program to the BCP when an abort occurs.

/2E	JPTSAV	IC
/30		PSW
/32		A1
/34		A2
/36		A3
/38		A4
/3A		A5
/3C		A6
/3E		A7
/40		A8
/42		A9
/44		A10
/46		A11
/48		A12
/4A		A13
/4C		A14
/4E		Floating point register 1
/50		Floating point register 2
/52		Floating point register 3

Explanation of the Labels

JPTUID 4 words containing the name of the user in the JOB command. They are used to define the default value of the USID parameter for BCL of LIB, UPD etc. commands and user requests.

JPTLMT 6 words used to control the execution of the JOB. 3 words define the limits of the execution of user program, used only when bit L of PCTMOD is set to 1 (prog. not loaded).
The other 3 words give the current values of these counters.

JPTDSK The left byte gives the file code of the DAD on which the program is stored as a load module.
The right byte is the file code of the DAD containing the USERID of the JOB. It is obtained from the :JOB command and used for default value.

JPTDIR Is the address in the DAD, of the first sector of user directory of the userid specified in JOB command.

S	I	E	Ig	Ty	LL	LC	UC	L	B	C	Lb	Cp	Jm	P	Dm
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

S = 1 System User

I = 1 JOB card received and processed.

I = 0 JOB card expected. BCP ignores all cards except JOB, END, EOJ.
Reset by EOJ, Abort etc...

C = 1 BCP processor is running (set by monitor).

Lb = 1 Librarian processor running.
Set by monitor on request by BCP, reset automatically at exit of processor Lib.

E = 1 EOJ received and processed.
Initialized = 1, reset on JOB card.
Set after EOJ processed or simulated.

Ty = 1 /EO assigned to a device such as TY, DY; the BCP has to prompt 'BCP': before reading a control command.

LL = 1 File code /02 is assigned to the same device as /EO. Thus the BCP does not have to print the command.

LC = 1 File code 02 is assigned to the same device as /01. This error message is set only once. (Otherwise sent twice in interactive mode).

UC = 1 File code /EO assigned to the same device as /01. The error message routine does not have to write the command on /01 before printing error message.

UC = 0 The error message routine has to print the command before writing the error message, afterward this routine sets UC = 1. The error message has to be sent once or twice according to LC bit and whether in interactive or batch mode.

L = 0 Exit and link.

L = 1 Exit no link, EOB.

B = 1 Batch processing mode.

CP = 1 Catalogued procedure.

JM = 1 JOB MISSING message printed (reset to zero at JOB command).

Ig = 1 Ignore all commands until :EOJ, :EOB, :JOB or :STP.

Dm = 1 Postmortem dump required in this case: P = ALL (monitor + batch)

Dm = 0 No postmortem dump.

JPTMD2 Right byte contains the value of ABCD in the :STP command, (initialized to /7F) i.e. the value of the exit / error code when the program is aborted : it will be compared with the current value of the error / exit code. If it is lower than the current code, this code remains unchanged when the program is aborted, else, it is replaced by the value of ABCD parameter. Left byte bit 7 set to 1 at Start Batch (SB) command.

JPTPST Contains the address of the current command in the command control table (CCT) of the BCP processor; used only by the BCP.

JPTBCP Is the disc sector address of the BCP processor.
N = 0 if the load module is a non-consecutive file.
N = 1 if the load module is a consecutive file.

JPTROT Is the disc sector address of the program to be executed.
N = 0 if the load module is non-consecutive file.
N = 1 if the load module is a consecutive file.

JPTCOD Left byte: the current abort code when the program is aborted. Transmitted from the Monitor to BCP in order to print an abort message to the User.
Right byte: the current exit code or error code of a program transmitted to BCP in order to check with maximum exit/error code authorized for the JOB step.

JPTMCD Left byte: the input file code of the current command used to read continuation lines.
Right byte: the maximum error/exit code authorized in the current step (the CODE = parameter on the :STP command).

JPTFCE Left byte: the file code from which the BCP reads correction for syntax errors. It is the file code defined in the ERR command.
Right byte: the highest error or exit code encountered from the beginning of the :JOB, used to check whether the step has to be executed or skipped.

JPTSDI the directory of the system library (USID = SYSTEM)

JPTBRE No. of pages required to load the BCP.

JPTPRE No. of pages required to load the program.

T:SLT The Software Level Table

This table is a list of PCT addresses; its position in the list defines the software level of a program. When a program is connected to a level, its PCT address is entered in the table at the appropriate position. The highest priority program is connected to level 0. A zero entry in T:SLT indicates that no program has been connected to that level.

The address of T:SLT is contained in location /C of the CVT.

The length of the table is specified at SYSGEN, and determines the total number of software levels, and hence the maximum number of programs which can be connected simultaneously.
The maximum number of entries is 240.

The layout of the SLT can be represented schematically as follows:

LOCATION	CONTENTS
0	Address of PCT for the program connected to level 0 (0 if none)
2	Address of PCT for the program connected to level 1 (0 if none)
2i	-- -- Address of PCT for program connected to level i (0 if none) -- -- --
n-6	Address of PCT of batch program (if anny)
n-4	Reserved
n-2	Idle time statistics routine
n	Idle task PCT address

'n' is the highest software level (lowest priority) and has a maximum value of 478, corresponding to level 239.

The table is updated by:

- The CNL and DSL commands
- The DCB command
- LKM 20 and LKM 21 (connect to and disconnected from a level)
- The activation of middleground or re-entrant programs.
- The DCF command, when the FCL task is connected to a level.

T:FCT The Filecode Control Table

A chain of one or more filecode control tables exists for each machine in the system in order to describe the assigment of a filecode to each device in use by that particular machine. They are created in the system dynamic area, the start of the chain being pointed to by the location MCTFCT (/A) of the MCT. Each table contains in word 0 the address of the next FCT in the chain; a zero entry indicates the end of the chain. Thus the same filecode may be assigned to two different files or devices in two different machines.

The layout of these 4, 5 or 7 word tables is as follows:

m

LOCATION	LABEL	CONTENTS					(MCTFCT)----
	bits		8	9	10	11 15	
0	FCTLNK	Address of next FCT in the chain, 0 if last					<----
2	FCTYP	see below	P	M	D1	<file type>	
4						<file CODE ~~type~~>	
6	FCTADR	DWT / LFT / FDC / DAD / FCTLNK					
8	FCTACN	Assign count (only present if M=1)					

FCTYP: <File type>
0=physical device (DWT address in FCTADR)
2=DMF logical file (LFT address in FCTADR)
4=DAD device
6=TDFM file (FDC address in FCTADR)
8=the file code is equivalent to another one
/A=the file code is assigned to a Data Com. device

When File type = 0, the FCTYP is used as follows:

bits	0										10	11				15
FCTYP	S	Ip	Op	On								0	0	0	0	0

S = 1 File code is assigned to a spooled device
Then Ip = Input spooled device = CR
Op = Output spooled device LP, PL, PP
On = 1 The file is opened i.e

- for the C.R, assign has been done to a disc file containing card image.
- for LP, an assign has been done to a disc logical file ready to receive the print-line. When end of file is encountered on the disc file, this assign is removed, and the file code is considered as "not opened".

S = 0 File code is assigned to a non-spooled device.

File type = 8; the file code in FCTCOD is equivalent to another file code, which is pointed to by the contents of FCTADR.

M = 1 the file code has several equivalences. When this bit is set, FCT has 5 words, the fifth one contains the number of file codes equivalent to this file code + 1 i.e. the total number of file codes assigned to the same file/device.
Initially when the "Assign File Code A to a File or Device" command is given, an entry of 4 words is created. When B is assigned "equiv. to A", then the old entry of A is deleted, another one of 5 words created; and a entry of 4 words for B is also created.
If C is assigned equiv. to B, then 1 entry of 4 words is created for C, and FCTACN of A is incremented.
When a file code is deleted and M=1, or if file type = 8, FCTACN of the initial file code table is decremented.
If FCTACN=0, this entry in FCT is released.
When deleting a file code if M = 0, the entry in the FCT is released.

D1 When this entry has bit M set, and if FCTACN is not zero after decrementing, the bit D1 is set to 1 but the entry is not released from the FCT chain (unless P = 1).

P = 1 The file code is a permanent one of the batch machine; this table has 7 words:

FCTLNK	0	
FCTYP	/2	
FCTCOD	/4	
FCTADR	/6	
FCTACN	/8	
FCTPTY	/A	Contents of FCTYP at machine declaration (DCB)
FCTPAD	/C	Contents of FCTADR at machine declaration (DCB)

The FCTPTY and FCTPAD are restored into FCTCOD, FCTADR respectively at the beginning of the JOB.

- this assignment is never freed.
- the file code of the table cannot be assigned as equivalent to other file codes (Other files codes can be assigned equivalent to this one).

T:DWT The Device Work Table

DWT describes the characteristics of physical devices and parameters of I/O requests. One DWT is created for each device when the configuration is declared at SYSGEN. They are chained via byte -2 and the first in the chain is addressed via location /A (CVTDWT) of the CVT. They are also pointed to by location /6 (FCTADR) of the related FCT. However, new entries can be added to the chain, e.g. by means of FCL commands.

The format of the table is as follows:

```
     --(FCTADR)            -----------------------------------------------
     |        (CVTDWT)    |                                               |
     |           |        |  Address of next DWTLNK in the system,        |
-2   |  DWTLNK    ------->|  0 if none                                    |
     |                    |-----------------------------------------------|
 0   ->DWTDN              |  Zero or Device Name                          |
                          |-----------------------------------------------|
 2      DWTDA             |  See explanation |<device address> (6 bits)   |
                          |-----------------------------------------------|
 4      DWTBLG            |              Best length                      |
                          |-----------------------------------------------|
 6      DWTDRV            |              Driver address                   |
                          |-----------------------------------------------|
 8      DWTSTS            |              Software status                  |
                          |-----------------------------------------------|
/A      DWTECB            |              ECB address                      |
                          |-----------------------------------------------|
/C      DWTBUF            | Char.address/buffer addr. via prog.channel    |
                          |-----------------------------------------------|
/E      DWTRLG            |              Request length                   |
                          |-----------------------------------------------|
/10     DWTELG            |              Effective length                 |
                          |-----------------------------------------------|
/12     DWTORD            |                    Order                      |
                          |-----------------------------------------------|
                          |RY/ |  |TO |   |   |   |   |   |  |            |
/14     DWTRY             |  RD|  |   | CR| L1| P1|   | R |  Device type|
                          | 0  |  | 4 | 5 | 6 | 7 | 8 | 9 |10         15|
                          |-----------------------------------------------|
/16     DWTTAB            |  Word to be output    / Tabulation address    |
                          |-----------------------------------------------|
/18     DWTCSM            |  check-sum (object order) / save char.        |
                          |                        for LP                 |
                          |-----------------------------------------------|
/1A     DWTCTL            |  4 x 4 right or left indicator / LP :         |
                          |                          save control code    |
                          |-----------------------------------------------|
/1C     DWTA5             |  (A5) PCT address of the program which        |
        DWTPCL            |             uses the device                   |
                          |-----------------------------------------------|
/1E     DWTA6             |  (A6) Sched. label                            |
        DWTSLB            |                                               |
                          |-----------------------------------------------|
/20     DWTC:N            | Controller status address C:Nxx or DCTHD      |
                          |  address                                      |
                          |-----------------------------------------------|
```

/22	DWTATT	Address of attached PCT/ or 0	
/24	DWTSST	SST sequence	
/26	DWTDET DWTMCL	Address of first PCT waiting for detach, 0 if none MCT address of program requesting the LKM	
/28	DWTUEC DWTMCB	User ECB MCT address of program containing the buffer	
/2A	DWTURO	User request order	
/2C	DWTNT DWTPCB	Address of next DWT in the timer chain. User PCT address if direct transfer into the user area.	
/2E	DWTIME	Timer value	anc DTO
/30	DWTQUE	Address of first request in queue /0 if none	
/32	DWTFLG	~~X~~ | ~~X~~ |RW|Ib|W |S |Mx|B	|IB|AM| DY | TY| I
/34	DWTSNH DWTIOB	Cyl. No. Cyl. No.	8 most sign, bits for RSN File code (if ECB is in use)
/36		ECB used	Buffer address
/38			Request length
/3A			Eff. length
/3C			Status
/3E	DWTSEC DWTSNL	 least sign for RSN	Head & Sector No./0/ Tab. address
/40	DWTFCT	FCT entry address	
/42	DWTRA	LFT, FDC, DAD address	
/44	DWTMCF DWTATK	MCT containing FCT MCT of program containing buffer	
/46	DWTPSH		
/4~~6~~8	DWTRST DWTPSH		
/4~~8~~A	DWTDTO	Device time out	
/4~~A~~C	DWTCIO	CIO information for error logging	
	DWTLNG	length of DWT	IS:CLK

(DWTC:N) MT
TK

2e byte hardware order
8000 free 0100 retry active /200 EOT
0000 busy 1C00 retry count

DWTECB MT TK /8000 Indpt expected

NB: Locations DWTFCT and DWTRA are used with X1215/6 disc devices only.

A fuller description of some of these locations follows:

DWTDA	St	X	S	Ip	Op	NO	Ti	Ta	D	DI	Device address
bits	0	1	2	3	4	5	6	7	8	9	10 15

Ip = 1 Input
Op = 1 Output
Used only for spoolable device

No = 1 Device not available
set and reset by operator command or system "IN" intialization. Usable in request command (it may be assign) where the user does not have to specify the device address --> the system allocates an operable device then informs the user about its address.

Ta = 1 Tabulation accepted with this device
Ti = 1 Time out accepted
D = 1 Disc device
DI = 1

DWTDN to DWTS$S are used especially by the drivers. Thus, they cannot be used by new modules of MAM, especially X:IO.

The remaining words of DWT are used by X:IO to perform the LKM/50 and to record user's parameters.

St = 1 Start Spooling received
X = Unused
S = Spooled Device

DWTRCB
0-7 line nr.
8-15

DWTC:N for ANA8

DWTCSM +1 = order

DWTC:N

0	STATUS 8000 = free 0000 = busy
2	— current line count
4	— max line count sysgen value
6	Character count for message
8	message block address

FF = ANSWER ON MESSAGE EXPECTED

ANA!

DWTSTS

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	I	Sk							B						

Sk = skip mode

I = interrupt expected
B = break detection

DWTRY:

Bit 0 = 1 : Retry or Release required by operator
Bit 1 = 1 : Disk is UPL type
Bit 4 (TO) = 1 : Time-out has occurred
Bit 5 (CR) = 1 : The card reader is assigned to filecode /EO of the batch machine. It is used to initiate a check on whether the current card is :JOB, :STP, :EOJ, or :EOB.
Bit 6 (L1) = 1 : The device is a line printer used in the batch machine.
Bit 7 (P1) = 1 : The device is a paper tape punch used in the batch machine.
Bit 9 :
R = 1 if no standard retry is requested.
R = 0 if standard retries are requested.

It is set according to bit 9 of the request order given by the calling program.

Device type (bits 10-15 of DWTRY):

0	=	TY
2	=	DY
4	=	CR
8	=	PR
/C	=	PP
/10	=	LP
/14	=	PL
/18	=	MT
/1C	=	TK
/20	=	1215 removable
/21	=	1215 fixed
/22	=	CDC 400 cylinders, 5 heads
/23	=	CDC 800 cylinders, 5 heads
/24	=	CDC 400 cylinders, 19 heads
/25	=	CDC 800 cylinders, 19 heads
/26	=	1216 Removable disc
/27	=	1216 Fixed disc
/28	=	X1250 Fixed Head Disc
/29	=	CMD 16M removable
/2A	=	CMD 16M fixed
/2B	=	CMD 48M fixed
/2C	=	CMD 80M fixed
/2D	=	PRIAM 8M
/2E	=	PRIAM 24M
/2F	=	Floppy
/33	=	PRIAM 35M

TY DY
BIT 0,1: 11 = AMA4 : /C442 (A)
10 = AMA8 : /8442
01 = ASCU4 : /4402

DWTFLG:

AM	=	1 AMA8 channel (DY connected to AMA8) OR AMA42
IB	=	1 Intermediary buffer
RW	=	1 Rewind on Mag. tape
I	=	1 ECB and buffer of the current operation belong to the CMA
Ib	=	1 Internal buffer provided (e.g. CR)
W	=	1 Transfer per word
W	=	0 Transfer per character
S	=	1 Single device controller
S	=	0 Multiple device controller
M	=	1 I/O processor
DY	=	1 FOR DY ON AMA, ASCU4
TY	=	1 FOR DY OR TY ON "

M = 0 Programmed channel
B = 1 Direct transfer to users area; = 0 for system area.
DWTIOB : "ECB" used to make the physical request to the driver.

DWTIOB : Cylinder No. (disc only)
DWTIOB + 2 : Buffer adress
DWTIOB + 4 : Requested length
DWTIOB + 6 : Effective length
DWTIOB + 8 : Status
DWTIOB + 10 : Head and sector No., disc only.

At the end of the physical I/O, DWTIOB + 4 is used to activate the entry 4 of X:IO.

DWTFCT : Used to save the FCT entry address for which the request is being processed.

DWTRA : Used to save the address of LFT, FDC, DAD, etc...., for which the request is being processed. This parameter is used especially by the access method in order to know which file is being processed.

Request Block structure of 'request in queue' function:-

```
( DWTQUE) or ( PWQUE)
  |
  |      ----------------------------------------------------------------
   ---->| Address of next request in queue (0 if last)                  |
        |                                                               |
  0     |                                                               |
        |---------------------------------------------------------------|
  2     | A2 cyl. No.                      or time out value.           |
        |---------------------------------------------------------------|
  4     | A3 Head & sector No.                  or 0                    |
        |---------------------------------------------------------------|
  6     | A4 FCT                                                        |
        |---------------------------------------------------------------|
  8     | A5 DWT                                                        |
        |---------------------------------------------------------------|
  /A    | A6 Sched. label                                               |
        |---------------------------------------------------------------|
  /C    | A7 Order                                                      |
        |---------------------------------------------------------------|
  /E    | A8 User ECB                                                   |
        |---------------------------------------------------------------|
  /10   | A9 File code                                                  |
        |---------------------------------------------------------------|
  /12   | A10 Buffer address in system dynamic area                     |
        |---------------------------------------------------------------|
  /14   | A11 Requ. length in system dynamic area                       |
        |---------------------------------------------------------------|
  /16   | A12 PCT address                                               |
        |---------------------------------------------------------------|
  /18   | A13 Buffer address in user area                               |
        |---------------------------------------------------------------|
  /1A   | A14                                                           |
         ---------------------------------------------------------------
```

T:LFT Logical File Table

Describes the characteristics of a DFM file and any access request parameters; is addressed from location /6 (FCTADR) of an Assign-Type 2 FCT.

The layout is as follows:-

```
-----------(FCTADR)
|
| LOCATION    LABEL                   CONTENTS
|            ------------------------------------------------------------
---0---->| LFTORD   |                      | User request order      |
         |----------|------------------------------------------------|
   2     | LFTEAD   |             User ECB address                   |
         |----------|------------------------------------------------|
   4     | LFTREC   |             User record area address           |
         |----------|------------------------------------------------|
   6     | LFTLGT   |             User request length                |
         |----------|------------------------------------------------|
   8     | LFTPCT   |             PCT Address (A5)                   |
         |----------|------------------------------------------------|
  /A     | LFTLAB   |             Sched. lab. address (A6)           |
         |----------|------------------------------------------------|
  /C     | LFTMD1   |A| P| S| O| U| C| T| R|Se| Fm|   | W|Co|Pr|Pw|Re|
         |----------|-----------|--|---------|--------------------|--|
  /E     | LFTMD2   |           |C1|         |                    |B |
         |----------|------------------------------------------------|
  /10    | LFTDCT   |             DAD control table address          |
         |----------|------------------------------------------------|
  /12    | LFTBOT   |  Address of GRANTB of the file within DAD      |
         |----------|------------------------------------------------|
  /14    | LFTSRC   |  Relative current sector number                |
         |----------|------------------------------------------------|
  /16    | LFTSAC   |  Address of current sector -in DAD-            |
         |----------|------------------------------------------------|
  /18    | LFTBAD   |  Blocking buffer address, 0 if none            |
         |----------|------------------------------------------------|
  /1A    | LFTBDS   | Displacement of next record in blocking buffer|
         |----------|------------------------------------------------|
  /1C    | LFTBUF   | Current buffer address (for current operation)|
         |----------|------------------------------------------------|
  /1E    | LFTSEC   | Current sector to be read or written           |
         |----------|------------------------------------------------|
  /20    | LFTORC   | Current order to be performed                  |
         |----------|------------------------------------------------|
  /22    | LFTSTC   | Current status                                 |
         |----------|------------------------------------------------|
  /24    | LFTSVD   | Save field for buffer pointer                  |
         |----------|------------------------------------------------|
  /26    | LFTSVS   | Save field for relative sector number          |
         |----------|------------------------------------------------|
  /28    | LFTSLU   | Save field for effective length                |
         |----------|------------------------------------------------|
  /2A    | LFTSLB   | Save field for number of blanks                |
         |----------|------------------------------------------------|
  /2C    | LFTSLC   | Character counter                              |
          -----------------------------------------------------------
```

LOCATION	LABEL	CONTENTS
/2E	LFTSLT	Total number of characters to transfer
/30	LFTSLR	Save field for real record length
/32	LFTLK1	Internally used as return address
/34	LFTLK2	Internally used as return address
/36	LFTATT	Address of attached PCT, /0 if not attached
/37	LFTDET	Address of the first PCT waiting for detach the LFT
/3A	LFTEOT	Relative highest sector No. of the file (data = file - 2, first two sectors not incl.) (5 to /7FFD)
/3C	LFTRQQ	Request queue address
/3E	LFTDFC	DAD file code (of the file)
/40	LFTRET	Return address after a physical I/O
/42	LFTFCT	FCT address

LFTMD1	A	P	S	O	U	C	T	R	Se	Fm		W	Co	Pr	Pw	Re

A = 0 LFT is busy. A request from the user program has already been recorded and not yet terminated. Thus the file is busy.

P = 1 The file is write protected.

P = 0 Not write protected, user can write to the file.

S = 1 Source file, set by assign command.

O = 1 Object file, set by assign command.

U = 1 Undefined type file (user file), contains user data.

C = 1 Load module (core image file), contains an executable
program. Only one bit over SOUC is set to 1 by assign command
or
implicit assign.

T = 1 Temporary file. Set at the time the file code is assigned. Used to know whether the file can be extended or not, i.e. if the Data Management has to read GRANTB to find out the next granule address in sequential access.

T = 0 Catalogued File.

R = 1 Random access is used on the file.

Se = 1 Sequential access is used.
R and Se are reset to 0 each time the BCP processes a control command in the batch machine.

Fm = 1 The last write operation on the file is a write file mark (the file is closed).
The rewind and read do not modify this bit.

W = 1 Write request.

= 0 Read request.

Co = 0 Non-consecutive file.

Pr = 1 Previous access on file was read.

Pw = 1 Previous access on file was write.

Re= 1 An EOF mark has been read on the file.

LFTMD2

B = 1 Intermediate buffer is available in system dynamic area.

T:DAD The DAD Control Table

All DADs used by the system must be declared, using the FCD command in the DCF and DCB commands, or using the ASG command in FCL, before they can be accessed. A T:DAD table is created for each DAD. They are chained together by location /0 (DADLNK), the first in the chain being pointed to by location /22 (CVTDAD) in the CVT. In addition, they are addressed from location 10 (LFTDCT) of the LFT, and from location /06 (FCTADR) of an Assign Type 4 FCT.

The layout is as follows:-

--(CVTDAD)/(DADLNK)/(LFTDCT)/(FCTADR)

LOCATION	LABEL	CONTENTS
---0---->	DADLNK	Address of next entry in the chain, 0 if last
2	DADFC	Disc file code /CX
4 6 /8	DADNAM	↕ DAD name
/A	DADSPT	No. of physical records/tracks
/C	DADSLG	(sector length) physical length in charac.
/E	DADNBC	No. of cylinders of the DAD
/10	DADBOT	Address of first cylinder of DAD
/12	DADSPG	No. of sectors/granules
/14	DADTPC	No. of tracks per cylinder
/16	DADPWT	DWT address
/18	DADSTA	Assign count
/1A	DADINT	No. of interlaces

The rest of the table describing the DAD allocations is as follows:

/1C	DADBTB	Length of DAD allocation table in char. (this word excluded)
		DAD ALLOCATION TABLE

Note: The table is ordered; 1 bit corresponds to 1 granule, and thus this table is equivalent to a BITAB description.

A bit set to zero means that the corresponding granule is allocated to a file or does not exist in this DAD.

A bit set to 1 means that the corresponding granule is free.

T:CORE The Core Allocation Status Table

This is used to describe the allocation status of all pages within a machine. It consists of 32 words (on a P859), or 8 words (other models). The first word contains the count of free pages within the machine, and the next 30 (P859) or 6 words define the page status. In each of these words bit 0 is set to zero, but the remaining bits represent page numbers in order: bit 1 of the first word represents page 0. bit 2 page 1, etc. If a bit is set to zero, the page is allocated or non-existent; if set to 1, the page can be allocated.

The last word (PAGOFF) contains the address of the Page-Off Table, which is created in the Dynamic Area when an FCL or Operator 'Page-Off' command is received. Like T:CORE, it contains 6 words in which a bit set indicates that the corresponding page is declared temporarily 'off' and cannot be used by the monitor until it has been set 'on' again.

The following diagram is a schematic representation of T:CORE:

```
                          -------------------------------------------
                         |           No. of Free Pages               |
                         |-------------------------------------------|
   (TVCCCR)------------->|  0   |  Page status bits (pages 0 - 14)   |
                         |------|------------------------------------|
                         |  0   |  Page status bits (pages 15 - 29)  |
                         |------|------------------------------------|
                         |  0   |             etc.                   |
                         |------|------------------------------------|
                         |  0   |                                    |
                         |------|------------------------------------|
                         |  0   |                                    |
                         |------|------------------------------------|
                         |  0   |                                    |
                         |------|------------------------------------|
            PAGOFF       |   Address of 'Page-Off' Table, or Zero    |
                         |                                           |
                          -------------------------------------------
                 bits       0                                      15
```

Location /1E of the CVT (TVCCCR) contains the address of T:CORE.

T:ELIG Table of Eligible Programs

This table is a set of flag bits, each one representing a level; thus its length is dependent on the number of software levels declared at SYSGEN. If a program is eligible to run (i.e. the status word (PCTSTA) in its PCT = 0) the corresponding bit is set to one; if it is not eligible, the bit is set to zero.

The actual length of the table is given by dividing the total number of levels by 15. In each word bit 0 is set to zero, but the remaining bits are in level order; bit 1 of word 0 corresponds to level 0, bit 2 to level 1, etc.

The following example shows the layout of a table for 240 levels:

```
(TVCELI, location /46 of the CVT)
 |
 |         ------------------------------------------------
  -->0    |  0  |                                          |
          |--|--|------------------------------------------|
   2      |  |  |                                          |
          |--|--|------------------------------------------|
   4      |  |  |                                          |
          |--|--|------------------------------------------|
   6      |  |  |                                          |
          |--|--|------------------------------------------|
   8      |  |  |                                          |
          |--|--|------------------------------------------|
  /A      |  |  |                                          |
          |--|--|------------------------------------------|
  /C      |  |  |                                          |
          |--|--|------------------------------------------|
  /E      |  |  |                                          |
          |--|--|------------------------------------------|
  /10     |  |  |                                          |
          |--|--|------------------------------------------|
  /12     |  |  |                                          |
          |--|--|------------------------------------------|
  /14     |  |  |                                          |
          |--|--|------------------------------------------|
  /16     |  |  |                                          |
          |--|--|------------------------------------------|
  /18     |  |  |                                          |
          |--|--|------------------------------------------|
  /1A     |  |  |                                          |
          |--|--|------------------------------------------|
  /1C     |  |  |                                          |
          |--|--|------------------------------------------|
  /1E     |  v  |                                    | 1   |
           ------------------------------------------------
           0                                           15
                                                        ^
                                                        |
                                              Idle task state
```

T:SWIN Table of programs loaded into the dynamic loading area

A chain is created in the dynamic area of the system machine defining the disc-resident swappable programs which have been swapped in. The start of the chain is addressed from /42 of the CVT and the chain structure can be represented diagrammatically as follows:

```
                     --------------            --------------
(/42 of CVT)------>|   Next        |---------->|      0       |
                   |--------------|           |--------------|
                   | PCT Address  |---        | PCT Address  |---
                    --------------   |         --------------    |
                                     |                           |
                                     |                           |
                                     |                           |
                                     |                           |
                                     |                           |
                                     |                           |
                                     |                           |
                                     |                           |
                    --------------   |         --------------    |
                   |    PCT       |<--        |    PCT       |<--
                   |              |           |              |
                   |    of        |           |    of        |
                   |              |           |              |
                   |loaded program|           |loaded program|
                   |              |           |              |
                   |              |           |              |
                   |              |           |              |
                    --------------             --------------
```

T:QIN Queue of program waiting to be swapped-in

T:QIN is structured in a similar way to T:SWIN and defines the programs waiting to be swapped-in. The initial order of this chain is the order in which the programs were declared. Location 44 of the CVT points to the start of the chain.

The following diagram illustrates the structure:

```
                     ---------------               ---------------
(/44 of CVT)----->|      Next      |---------->|       0        |
                  |----------------|           |----------------|
                  |  PCT Address   |---        |  PCT Address   |---
                     ---------------   |           ---------------    |
                                       |                              |
                                       |                              |
                                       |                              |
                                       |                              |
                                       |                              |
                                       |                              |
                     ---------------   |           ---------------    |
                  |       PCT      |<--        |       PCT      |<--
                  |                |           |                |
                  |       of       |           |       of       |
                  |                |           |                |
                  |     waiting    |           |     waiting    |
                  |                |           |                |
                  |     program    |           |     program    |
                  |                |           |                |
                  |                |           |                |
                     ---------------               ---------------
```

Swappable background programs are always placed at the end of the T:QIN chain so that they are only swapped-in when no foreground programs are waiting to run.

T:RTC The Real-Time Clock Table

This is used to hold the current values of program timers, the time and the date. while the machine is running.

A chain of blocks, each containing the PCT of a program, is formed for all programs connected to a particular timer. Thus there can be a chain for each timer. The beginning of these chains is pointed to from within T:RTC.
T:RTC itself is addressed from location 0 of the CVT.

The following diagram shows the layout of the Real-Time Clock Table:

	LOCATION	LABEL	CONTENTS	(T:CTIM)----
	0	TBDAY	Day (ASCII)	<-
	2	TBMON	Month (ASCII)	
	4	TBYEAR	Year (ASCII)	
H:TIME	6	TBHOUR	Hour Timer (binary hours - 24)	
	8	TBMIN	Min. Timer (binary mins - 60)	
	/A	TBSEC	Sec. Timer (binary secs - 60)	
	/C	TBTEN	1/10 of sec (binary) - 10	
	/E	TBFIF	1/50 of sec (binary) - 5	
	/10	TBPUL	0 - No. of pulses of non-standard clock	
H:POIN	/12	FSHOUR	first timer block address (conn. to hour timer)	
	/14	FSMIN	first timer block address (connected to minute)	
	/16	FSSEC	first timer block address (connected to second)	
	/18	FSTEN	first timer block address (conn. to 1/10 sec.)	
	/1A	FSFIF	first timer block address (conn. to 1/50 sec.)	
	/1C	FSPUL	first timer block add. (conn.to non-std clock)	
	/1E	FSABS	first timer block address (absolute time)	
V:FLAG	/20	SCHOUR	flags used for scanning -	
	/22	SCMIN	the chain of blocks -	
	/24	SCSEC	connected to the corresponding -	
	/25	SCTEN	timer	
	/28	SCFIF	if 0, then scan	

MONITOR CONTROL TABLES

LOCATION	LABEL	CONTENTS
/2A	SCPUL	if = 0 do not scan
/2C	V:ABS	Abslute Time flagword
/2E	V:SCAN	If ≠ 0 X:RTC is running
/30	RSHOUR	- 24
/32	RSMIN	- 60
/34	RSSEC	- 60
/36	RSTEN	- 10
/38	RSFIF	- 5
/3A	RSPUL	- 1 if standard clock - No. of pulse if not
/3C	T:SHT	First short timer block address

Timer Blocks

The general format of blocks connected to a timer is:

0	Address of next block, /0 if last one		
2	S	Block type	Timer No. (8 bits)
4	PCT address		
6			
8			

/3E	IO:CUR	Ø	minutes Seconds 100 ms.
/44	IO:AWK	/FFF	
/4A	IO:LNK	Ø	

The actual format depends on the block type, as follows:

Block type 0. Connect a program to a timer (format 1, or format 2 after the first activation):

LOCATION	CONTENTS		
0	Address of next block		
2	Zero 0 7	Timer No. given by user 8 15	bits
4	PCT Address		
6	No. of cycles of the timer No. immediately lower than the current one (first activation)		
8	- NC of the current timer		
/A	- PR (0 if one activation)		

Format 2 before first activation
(absolute time):

6	Hour	Min.
8	Sec.	

NC = Number of Cycles
PR = Pulse Rate

Initialisation of a BLOCK TYPE 0 (connect to a timer)

The block is inserted in the chain of blocks of programs connected to the same timer, between T:RTC1 and the first block. The block locations are initialised as follows:

Byte 0 contains the address of the next block in the chain.

Byte 2 is zero.

Byte 3 contains the Timer No. given by the user.

Byte 4 contains the PCT address of the program connected.

Byte 6 and Byte 8 -

a) User block format 1 (iterative timer)

NC is divided by the 'reset value' of the given timer. If the remainder = 0, then its negative value is placed in byte 8 and the quotient in byte 6.

If the remainder ≠ 0, then the quotient is divided by the 'reset value' of the next higher timer. If the remainder of this division is ≠ 0, the remainder is negated and placed in byte 8 and the quotient in byte 6. If the remainder = 0, the process of division is repeated until either the quotient becomes zero (in which case the negated remainder is placed in byte 8 and zero in byte 6), or the hour timer is reactivated (i.e. the hour No. is placed in byte 8 and zero in byte 6).

The block is then inserted in the chain of blocks belonging to the timer corresponding to the last division.

b) User block format 2 (absolute timer)

The time delay until the first activation is computed in seconds, then the process corresponding to that of format 1 is started with 'NC' set to the computed delay. The Timer No. is the seconds timer.

Byte 10 contains the negative value of 'PR' as given by the user.

Initialisation of a block type (Wait for a given time)

The block is initialised as follows:

```
          ------------------------------------------------------------
   0     |     Address of the next block, Zero if last                |
         |------------------------------------------------------------|
         | S |    1 (Block type)            |  Timer No. given by user |
   2     | 0 | 1                          7 | 8                     15 |
  bits   |------------------------------------------------------------|
   4     |                     PCT Address                            |
         |------------------------------------------------------------|
         |   No. of cycles of the timer No. immediately lower         |
   6     |            (first activation = 0)                          |
         |------------------------------------------------------------|
   8     |   - NC of the current timer                                |
         |------------------------------------------------------------|
  /A     |   User ECB address                                         |
          ------------------------------------------------------------
```

S = 0 for block type 0

Note: If the user program has been swapped out when the specified time expires, the block must be linked to the PCTMOV chain of events for swapped programs.

Initialisation of a block type 2 (Set a Timer Block)

The block is set up as follows:

```
          ------------------------------------------------------------
   0     |     Address of the next block, Zero if last                |
         |------------------------------------------------------------|
         | S |    2 (Block type)            |  Timer No. given by user |
   2     | 0 | 1                          7 | 8                     15 |
  bits   |------------------------------------------------------------|
   4     |                     PCT Address                            |
         |------------------------------------------------------------|
         |   No. of cycles of the timer No. immediately lower         |
   6     |            (first activation = 0)                          |
         |------------------------------------------------------------|
   8     |   - NC of the current timer                                |
         |------------------------------------------------------------|
  /A     |   User ECB address                                         |
         |------------------------------------------------------------|
  /C     |   Sched. label address                                     |
         |------------------------------------------------------------|
  /E     |   Only if S = 1; used to save the value of TMB3 for user   |
         |                                                     block  |
          ------------------------------------------------------------
```

TMB3 is the last word of the monitor control block used with the LKM connect to timer requests. It is reset to zero when the time has elapsed.

QCB The Queue Control Block

One of these control blocks is created for each queue initiated within a machine. They are automatically generated when a 'Put In Queue' request names a new queue. These control blocks are chained together via word 0, the first one being pointed to by the location MCTKIN (/14) of the MCT and the last in the chain having zero in word 0. The format of a queue control block is as follows:

LOCATION	LABEL	CONTENTS
QBLNK	0	Address of next QCB of the machine, 0 if last
QBNAM	2	↑ QNAM, in 6 character,
	4	left justified,
	6	↓ filled with spaces.
QBCUR	8	Current element in queue being processed
QBNXT	/A	Next element in the queue
QBLST	/C	Last element in the queue
QBREQ	/E	Address of the queue for "get next element in queue" requests

Where - QBLNK is the pointer to the next QCB of the machine, 0 if the last.

- QBNAM 3 words, containing the name of the queue.

- QBCUR points at the current entry of the queue, the one being processed by User program, 0 if none.

- QBNXT points at the next element to be delivered by "get next" element in queue" request, 0 if none.

- QBLST points at the last element in Q, 0 if none.

- QBREQ is the queue of the requests "get next element in Q". These requests are recorded when the queue is empty.

The format of a queue entry (which is just a memory area having a particular structure) is as follows:

```
                 ---------------------------------------------
                | F |   Entry length  L                        |           ^
                |---|------------------------------------------|           |
                |   First data word                            |<-------   |
                |                                              |  |  |  | |
                |                                              |  |  |  | |
                |   Data words                                 |  |  |  | |
                |                                              |  |  |  | |
                |                                              |  |  |  | |
                |                                              |  |  |  | |
                |   Last data word                             |  |  |  | |
                |----------------------------------------------|  |  |  | |
QBCUR           |                                              |  |  |  | |
  or            |   Next element in this queue                 |  |  |  | |
QBNXT--->       |                                              |  |  |  | |
  or            |----------------------------- Queue 1 ------- |  |  |  | L
QBLST           |   First data word address                    |--   |  | |
                |----------------------------------------------|     |  | |
                |   Next element Queue 2                       |     |  | |
                |----------------------------- Queue 2 ------- |     |  | |
                |   First data word address                    |-----   | |
                |----------------------------------------------|        | |
                |   Next element Queue 3                       |        | |
                |----------------------------- Queue 3 ------- |        | |
                |   First data word address                    |--------  |
                |----------------------------------------------|          |
                |   No. of queues to which the message         |          |
                |             belongs   (QCNT)                 |          v
                 ----------------------------------------------
```

Where - F = 0: the queue entry (memory area) is to be released as soon as 'Get next area' request has been issued for the next area in all the queues in which this area has been placed.

- F = 1: The area is not to be released

- L is the length of this queue entry.

The next words are user data; these are followed by pairs of words (one pair for each queue into which this area has been entered), the first pointing to a similar pair of words in the next entry in the queue and the second to the beginning of the user's data in this area. Pointers QBCUR, QBNXT and QBLST always point to the first of these two words.

At the end of the queue entry is a word (QCNT) containing a count of the number of queues in which this queue entry has been placed. Thus data areas belonging to the same queue are chained. When a 'Get next queue entry' request is received, this count is decremented for the current queue entry (pointed to by QBCUR in the QCB); if QCNT becomes zero, the memory area occupied by this queue entry buffer is freed, provided that F = 0.

When a 'Get next' request is received and there is no queue having the given name, the system creates a QCB for the issuing machine with QBCUR = QBNXT = QBREQ = 0, and the request is recorded in the queue of requests pointed to by QBREQ. Entries in this queue have the following form:

```
QBREQ
   |
   |             -------------------------------------------------
   |
   -------->  |   Address of next request recorded, 0 if last   |
              |-------------------------------------------------|
              |   ECB address                                   |
              |-------------------------------------------------|
              |                                                 |
```

T:DCT The Disc Control Table

DCT describes the characteristics of a disc and current status. They are addressed from location /20 (DWTC:N) of the associated DWTs, and are chained together via word 0 to facilitate scanning when a disc interrupt disc interrupt is received.
The layout of the DCT depends on the disc type. (X1215/16 or fixed head disc, CDC BIGD, CDC BIGD2, 250K floppy, 1M floppy, UPL disc)

X1215/1216

LOCATION	LABEL	CONTENTS
0	DCTLNK	Address of next entry in chain or zero
2	DCTHD	See below
4	DCTDWT	Address of DWT of the disc
6	DCTCUR	See below
8	DCTSK	Contents of register for Seek
/A	DCTRD	Contents of register for Read
/C	DCTRM1	First multiplex word for Read (see below)
/E	DCTRM2	Second multiplex word for Read (address)
/10	DCTW	Contents of register for Write
/12	DCTWM1	First multiplex word for Write (see below)
/14	DCTWM2	Second multiplex word for Write (address)
/16	DCTSLG	Sector length (VTOC)
/18	DCTVTC	Sector address of VTOC
/1A	DCTNBT	No. of tracks per cylinder
/1C	DCTBAD	Sector address of bad track list
/1E	DCTSPT	No. of sectors per track (first DAD)
/20	DCTINT	No. of interlaces (first DAD)
/22	DCTNBR	Pack number (volume serial number)
/24	DCTBTR	Address of bad track list or 0
/26	DCTREP	Address of first replacing (spare) cylinder
/28	DCTCYL	No. of cylinders of the disc
/2A	DCTVCH	Current virtual cylinder no. (11 bits Current virtual head no. (5 bits)

/2C		Not used
/2E	DCTRM3	Bits 4-11 contain most significant address bits when the address > 18 bits (For Read)
/30	DCTWM3	Bits 4-11 contain most significant address bits when the address > 18 bits (For Write)
/32 /34 /36 /38 /3A /3C /3E		Not used

DCTHD:

A	Sy	DDA	Current head position	
0	1	2	8	15

A = 1 device free
A = 0 device busy

Sy = 1 system disc or disc being premarked
DDA Disc Device Address

DCTCUR:

N	Rd	I	Sz		S	R	W	B	D	RR	Pm		Curr.Retry No	
0	1	2	3	4	5	6	7	8	9	101	11	12	13	15

N = 1 Device not operable
Rd = 1 Disc becomes ready (just mounted, not yet initialised)
I = 1 Interrupt pending
Sz = 1 Seek to zero to be performed
S = 1 Seek to be performed
R = 1 Read to be performed
W = 1 Write to be performed
B = 1 Position in DCTHD is wrong: seek to zero is required in the next command
D = 1 connected to DMAC (not used)
RR = 1 The controller is busy when the request is received, thus it cannot be performed (Read or Write being processed on the other unit)
Pm = 1 The disc is being premarked.

DCTRM1/DCTWM1	bit 0 =	0	Char. mode (length in chars)
		1	Word mode (length in words)
	bit 1 =	0	Input
		1	Output
	bits 2, 3		Bits 64/128 of physical address
	bits 4-15		Length

CDC discs with BIGD or BIGD2 controller

LOCATION	LABEL	CONTENTS
0	DCTLNK	Address of next entry in chain or zero
2	DCTHD	See below
4	DCTDWT	Address of DWT of the disc
6	DCTCUR	See below
8	DCTCU	bit 0 = 0 BIGD controller bit 0 = 1 BIGD2 controller
/A /C /E /10 /12 /14		Not used
/16	DCTSLG	SECTOR LENGTH (VTOC)
/18	DCTVTC	Sector address of VTOC
/1A	DCTNBT	No. of tracks per cylinder
/1C	DCTBAD	Sector address of bad track list
/1E	DCTSPT	No. of sectors per track (first DAD)
/20	DCTINT	No. of interlaces (first DAD)
/22	DCTNBR	Pack number (volume serial number)
/24	DCTBTR	Address of bad track list or 0
/26	DCTREP	Address of first replacing (spare) cylinder
/28	DCTCYL	No. of cylinders of the disc
/2A	DCTVCH	Current virtual cylinder no. (11 bits) Current virtual head no. (5 bits)
/2C	DCTCMD	Address of command sequence
/2E	DCTSKB	Seek or seek to zero command
/30	DCTRDB	Other commands (see below)

/32	DCTCW1	See below
/34	DCTCW2	
/36	DCTCW3	
/38	DCTCW4	
/3A	DCTCW5	
/3C	DCTCW6	
/3E	DCTCW7	

DCTHD:

```
 ------------------------------------------------------
| A | Sy |        DDA        |  Current head position  |
 ------------------------------------------------------
  0   1    2                 8                       15
```

A = 1 device free
A = 0 device busy

Sy = 1 system disc or disc being premarked
DDA   Disc Device Address

DCTCUR:

```
 ----------------------------------------------------------------------
| N | Rd | I | Sz |   | S | R | W | B | D | RR | Pm |   | Curr.Retry No|
 ----------------------------------------------------------------------
  0   1    2   3    4   5   6   7   8   9  101   11  12  13          15
```

N = 1 Device not operable
Rd = 1 Disc becomes ready (just mounted, not yet initialised)
I = 1 Interrupt pending
Sz = 1 Seek to zero to be performed
S = 1 Seek to be performed
R = 1 Read to be performed
W = 1 Write to be performed
B = 1 Position in DCTHD is wrong: seek to zero is required in the next command
D = 1 connected to DMAC (not used)
RR = 1 The controller is busy when the request is received, thus it cannot be performed (Read or Write being processed on the other unit)
Pm = 1 The disc is being premarked.

CDC discs on BIGD controller (DCTCU bit 0 = 0)

DCTCW1 total length (in words)

DCTCW2 length of first block (in words)
bit 1 = 1 data chaining
bits 2, 3 most significant bits of physical address of first block

DCTCW3 least significant bits of physical address of first block

DCTCW4 length of second block (in words)
bits 2, 3 most significant bits of physical address of second block

DCTCW5 least significant bits of physical address of second block

DCTCW6
and
DCTCW7 are not used

CDC discs on BIGD2 controller (DCTCU bit 0 = 1)

DCTCW1 total length (in words)

DCTCW2 length of first block (in words)
bit 1 = 1 data chaining
bit 0 = 1 Read, but no transfer into memory (READ only)

DCTCW3 8 most significant bits of physical address of first block

DCTCW4 16 least significant bits of physical address of the first block

DCTCW5 length of second block (in words)
bit 0 = 1 Read, but no transfer into memory (READ only)

DCTCW6 8 most significant bits of physical address of second block

DCTCW7 16 most significant bits of physical address of second block

Data fault error recovery on BIGD and BIGD2 controller (READ only)

Maximum 27 retries are performed with all possible combinations of early/late strobe and carriage forewards/backwards.
(3 retries per combination)

DCTRDB bit 0 = 1 early strobe
bit 1 = 1 late strobe
DCTCW1 bit 0 = 1 carriage backwards
bit 1 = 1 carriage forwards

Floppy on FLDB

LOCATION	LABEL	CONTENTS
0	DCTLNK	Address of next entry in chain or zero
2	DCTHD	See below
4	DCTDWT	Address of DWT of the disc
6	DCTCUR	See below
8	TST	TST instruction
/A	WER1	WER1 instruction (length)
/C	WER2	WER2 instruction (eff. address)
/E	C10	C10 start instruction
/10	C10H	C10 halt instruction
/12	1NR	1NR instruction
/14	OTR	OTR instruction
/16	RER	RER instruction
/18	DCTSST	SST instruction
/1A	RWER1	WER1 instruction
/1C	RWER2	WER2 instruction
/1E	BIOLI1	Content;s of register for CIO
/20	RWER12	WER1 instruction
/22	RWER22	WER2 instruction
/24	BIOLI2	Contents of register for CIO
/26	PRWER	Pointer to WER instruction
/28	DCEBUF DCTBUF	Buffer address
/2A	DCELGH DCTLGH	Length in words
/2C	DCESTA DCTSTA	Soft status
/2E	DCTNXT	
/30	DCEEFL	Effective length in characters

/32	DCTSEC	Sector address
/34	DCTSCH	Max. sector number in track
/36	DCTSAV	Save address for initial length
/38 /3A /3C /3E		Not used

DCTHD:

A	Sy	DDA	Current head position	
0	1	2	8	15

A = 1   device free
A = 0   device busy

Sy = 1 system disc or disc being premarked
DDA   Disc Device Address

DCTCUR:

N	Rd	I	Sz		S	R	W	B	D	RR	Pm		Curr.Retry No	
0	1	2	3	4	5	6	7	8	9	101	11	12	13	15

N = 1   Device not operable
Rd = 1   Disc becomes ready (just mounted, not yet initialised)
I = 1   Interrupt pending
Sz = 1   Seek to zero to be performed
S = 1   Seek to be performed
R = 1   Read to be performed
W = 1   Write to be performed
B = 1   Position in DCTHD is wrong: seek to zero is required in the next command
D = 1   connected to DMAC (not used)
RR = 1   The controller is busy when the request is received, thus it cannot be performed (Read or Write being processed on the other unit)
Pm = 1   The disc is being premarked.

Floppy on FL1MB/FL1MZ

LOCATION	LABEL	CONTENTS
0	DCTLNK	Address of next entry in chain or zero
2	DCTHD	See below
4	DCTDWT	Address of DWT of the disc
6	DCTCUR	See below
8	DCTSK	Contents of register for Seek
/A	DCTRD	Contents of register for Read
/C	DCTRM1	Multiplex word 1 for Read
/E	DCTRM2	Multiplex word 2 for Read
/10	DCTW	Contents of register for Write
/12	DCTWM1	Multiplex word 1 for Write
/14	DCTWM2	Multiplex word 2 for Write
/16	DCTSLG	Sector length (VTOC)
/18	DCTVTC	Sector address of VTOC
/1A	DCTNBT	No. of tracks per cylinder
/1C	DCTBAD	Sector address of bad track list
/1E	DCTSPT	No. of sectors per track
/20	DCTINT	No. of interfaces (first DAD)
/22	DCTNBR	Pack number (volume number)
/24	DCTBTR	Address of bad track list or zero
/26	DCTREP	Address of first replacing (spare) cylinder
/28	DCTCYL	No. of cylinders of the disc
/2A	DCTVCH	Current virtual cylinder number (11 bits Current virtual head number (5 bits)
/2C	DCTCMD	Command order
/2E	DCTRM3	Multiplex word 3 for Read
/30	DCTWM3	Multiplex word 3 for Write
/32	DCTTYP	See below

/34	DCTRTY	Number of retries
/36	DCTSEC	Sector number
/38 /3A /3C /3E		Not used

DCTHD:

A	Sy	DDA	Current head position
0	1	2	8 15

A = 1 device free
A = 0 device busy

Sy = 1 system disc or disc being premarked
DDA Disc Device Address

DCTCUR:

N	Rd	I	Sz		S	R	W	B	D	RR	Pm		Curr.Retry No
0	1	2	3	4	5	6	7	8	9	101	11	12	13 15

N = 1 Device not operable
Rd = 1 Disc becomes ready (just mounted, not yet initialised)
I = 1 Interrupt pending
Sz = 1 Seek to zero to be performed
S = 1 Seek to be performed
R = 1 Read to be performed
W = 1 Write to be performed
B = 1 Position in DCTHD is wrong: seek to zero is required in the next command
D = 1 connected to DMAC (not used)
RR = 1 The controller is busy when the request is received, thus it cannot be performed (Read or Write being processed on the other unit)
Pm = 1 The disc is being premarked.

DCTTYP Type of Floppy

0	type = F1	250K floppy disc (DAD structured)
1	type = F2	not supported
2	type = F3	1M floppy disc (DAD structured)
3	type = F4	1M floppy disc (Data disk)
4	type = F5	1M floppy disc (Data disk)
5	type = F6	250K floppy disc (Data disk)

UPL discs

LOCATION	LABEL	CONTENTS
0	DCTLNK	Address of next entry in chain or zero
2	DCTHD	See below
4	DCTDWT	Address of DWT of the disc
6	DCTCUR	See below
8	DCTIOD	Address of I/O Descriptor
/A	DCTCWT	Address of Control unit Work Table
/C	DCTFLG	See below
/E	DCTQUE	Address of next DCT queued for same controller
/10	DCTCOM	Command code
/12	DCTIOL	Diminished requested length (req. length - 1)
/14	DCTRBF	Buffer address
/16	DCTSLG	Sector length (VTOC)
/18	DCTVTC	Sector address of VTOC
/1A	DCTNBT	No. of tracks per cylinder
/1C	DCTBAD	Not used
/1E	DCTSPT	No. of sectors per track
/20	DCTINT	No. of interlaces of the disc (= 1)
/22	DCTNBR	Pack number (volume serial number)
/24	DCTBTR	Not used
/26	DCTREP	Not used
/28	DCTCYL	No. of cylinders of the disc
/2A	DCTVCH	Not used

/2C	DCTNSC	Number of sectors
/2E	DCTRML	Length of last record
/30		Not used
/32		
/34		
/36		
/38		
/3A		
/3C		
/3E		

DCTHD:

A	Sy	DDA	Current head position
0	1	2	8 15

A = 1 device free
A = 0 device busy

Sy = 1 system disc or disc being premarked
DDA Disc Device Address

DCTCUR:

N	Rd	I	Sz		S	R	W	B	D	RR	Pm		Curr.Retry No
0	1	2	3	4	5	6	7	8	9	101	11	12	13 15

N = 1 Device not operable
Rd = 1 Disc becomes ready (just mounted, not yet initialised)
I = 1 Interrupt pending
Sz = 1 Seek to zero to be performed
S = 1 Seek to be performed
R = 1 Read to be performed
W = 1 Write to be performed
B = 1 Position in DCTHD is wrong: seek to zero is required in the next command
D = 1 connected to DMAC (not used)
RR = 1 The controller is busy when the request is received, thus it cannot be performed (Read or Write being processed on the other unit)
Pm = 1 The disc is being premarked.

DCTFLG bit 0 = 0 Removable disc
bit 0 = 1 Fixed disc

bits 14/15 Relative drive number (0 - 3)

CWT Control unit Work Table

One table exists for each control unit for the UPL discs. The anchor for this table is location /A of the DCTs of the discs connected to this control unit.
The layout of this table is as follows:

LOCATION	LABEL	CONTENTS
0	CWTSTA CWTLEV	Busy free indicator (bit 0 = 1 indicates busy) Interrupt level (bits 10 = 15)
2	CWTCIO	CIO start instruction
4	CWTSST	SST instruction
6	CWTQUE	Address of first DCT waiting for the controller to be free
8	CWTDWT	Address of first DWT in DWT chain
/A /C /E /10	CWTRDQ	See below
/12	CWTCID CWTIDO	Control unit interrupt descriptor IOD address Describing I/O to be performed (3 bytes)
/15	CWTIDI	IOD address describing complete I/O (3 bytes)
/18	CWTINT	Interrupt type (see below)
/19	CWTRDN	Interrupting relative drive number
/1A	CWTPAI	PAI code (peripheral attention interrupt)

CWTRDQ One word is reserved for each drive connected to the control unit, corresponding to the relative drive number.

Bit 15 of such word is set at start of I/O and reset at completion.

If an I/O request is submitted to the same drive before I/O completion, then the DWT address of the target disc is recorded in the corresponding word.

CWTINT Describes the interrupt type.

/20 I/O completion, the previous I/O request has just ended, the real address of the IOD can be found in the CID (CWTIDI).

/40 Release interrupt, controller is able to accept an I/O request.

/80 Peripheral Attention interrupt, the CID (CWTRDN) contains the relative drive number of the interrupting drive.

IOD I/O Descriptor

This block is used to give all information necessary to perform a command. The anchor is location /8 of the DCT.

The layout of this table is as follows:

BYTE NO.	LABEL	CONTENTS
-2	IDDCT	Corresponding DCT address
0	IDRDN	Relative drive number
1	IDCOM	See below
2	IDCONO	Control information
3	IDLST	Logical status
4-5	IDPST	Physical status
6-7	IDIOAL IDRML IDNOS IDIOEA	Buffer length Remaining length Number of sectors Buffer extension address
8	IDSD	Displacement of record in sector
9	IDNST	Number of sectors per track
/A	IDNTC	Number of tracks per cylinder
/B-/C	IDNOR	Number of retries (in case of automatic retry)
/E-/F	IDRSN IDRRSN	Real sector number where command is started Real sector number containing the searched record
/13	IDKD IDRSNF IDPAB IDFBN	Number of octads of beginning of key ignored in the comparison Real sector number (used for format tracks command) Program area beginning Block number where search begins
/14 /15 /16-18	IDKL IDDL IDADL IDPAE IDKA	Record key length Record data length Data length to return Program area end Key address (SMI command)
/19-1B	IDFIB	File beginning (SMI command)
/1C-1E	IDFIE	File end (SMI command)
/1F	IDKLNG	Key length (SMI command)

IDCOM command code

0	Read data
1	Write data
6	Format track
7	Format defective track
8	Format alternate track

T:LKM LKM Processing Control Table (U:LKM for User LKMs)

This is used by the LKM interrupt routine (I:LKM), when processing a link-to-monitor request. It contains the address of the routine which processes the request.

The table has the following layout:

```
                  ----------------------------------------------
(T:LKM)---->     | Max. no. of LKM entries                      |
                 |    (No. of entries + 1)                      |
                 |----------------------------------------------|
First 2-word     | F |                                          | LKMFG
                 |----------------------------------------------|
   entry         |                                              | LKMPTR
                 |----------------------------------------------|
2nd 2-word       | F |                                          |
                 |----------------------------------------------|
   entry         |                                              |
                 |----------------------------------------------|
                 |                                              |
   etc.          |                                              |
                 |                                              |
                 |                                              |
                 |                                              |
```

Where -

LKMFG is a flag word, in which if:

F = 0 the request is processed by a program running at hardware level 63 (X:IO or X:MASG or X:USVC); LKMFG, bits 1 to 15, contain the parameter to be passed to the processing program in register A3. It is used to identify the entry point in the called program. In this case, LKMPTR points to the PCT of the program to be activated.

F = 1 the macro is processed by a module running at level 62. In this case, LKMPTR contains the start address of the module.

Notes:

a) F = 1, the processing routine runs at the interrupt level 62. It must respect the interfaces of interrupt routines, especially when it uses the A15 stack or it branches to the dispatcher.

b) F = 0, the processing program runs at the software level. It can be either:

- X:IO program, if the request to be processed is an LKM 1.

- X:USVC program, if the request is chosen at sysgen to be core resident instead of disc resident. In this case, the entry point LKMFG contains the entry number in the T:RMAC table. This table contains the start addresses of the core resident processing modules.

- X:MASG program, if the macro is processed by a transient module. In this case, the entry point in LKMFG identifies the segment and entry point as follows:

```
 ---------------------------------------------------------
| 0 |     Entry No.        |           Segment             |
 ---------------------------------------------------------
      1                 5    6                          15
```

bits 1 to 5 Entry point No. starting with 1
bits 6 to 15 Segment No.

The current or last entry processed is stored in the word T:SCUR, whose address can be found from the MAP of the system.

T:SPT The Spooling Table

This table contains the characteristics of physical spooled devices and their related disc files. One table is created at SYSGEN for each spooled device and the tables are chained together via location zero, the first one being pointed to by location /78 (CVTSPT) of the CVT. The structure of the table is as follows:

```
LOCATION     LABEL                    CONTENTS

(CVTSPT)            0
 |          -----------------------------------------------------------------
 |--0->| SPTLNK | Address of next table in the chain-zero if last           |
       |--------|---------------------------------------------------------- |
   2   | SPTDVA | E  | O | C | D | R | N | B | S |     |EB | Device Addr.|
       |        |  o |   |                                                  |
       |--------|----|---|--------------------------------------------------|
   4   | SPTNBF | E  | O | No. of jobs entries in the queue                 |
       |        |  1 |   |                                                  |
       |--------|-------------------------------------------------------------|
   6   | SPTDWT |          DWT Address of the spooled device                |
       |--------|-------------------------------------------------------------|
   8   | SPTDAD |          DAD Control Table Address                        |
       |--------|-------------------------------------------------------------|
  /A   | SPTSTA | E  | O | JB| EJ| BCP| ESJ| EOB|            | Dev. type    |
       |        |  2 |   |                                                  |
       |--------|-------------------------------------------------------------|
ud /C | SPTCUR | Current file pointer being unspooled                  |
       |        | (word address, sector 5)                                  |
       |--------|-------------------------------------------------------------|
  /E   | SPTNXT | Word address of next queue entry                          |
       |--------|-------------------------------------------------------------|
  /10  | SPTLST | Word address of the last word of the queue                |
       |--------|-------------------------------------------------------------|
  /12  | SPTFC1 |  |                      | Spooled device filecode         |
       |--------|-------------------------------------------------------------|
  /14  | SPTRC1 |  Record area address                                      |
       |--------|-------------------------------------------------------------|
  /16  | SPTRL1 |  Requested length                                         |
       |--------|-------------------------------------------------------------|
  /18  | SPTEL1 |  Effective length                                         |
       |--------|-------------------------------------------------------------|
  /1A  | SPTST1 |  Status                                                   |
       |--------|-------------------------------------------------------------|
  /1C  | SPTSC1 |  Zero                                                     |
       |--------|-------------------------------------------------------------|
  /1E  | SPTDFC |                         | DAD filecode                    |
       |--------|-------------------------|-----------------------------------|
  /20  | SPTFC2 |  |                      | Spooled filecode                |
       |--------|-------------------------------------------------------------|
  /22  | SPTRC2 |  Record area address                                      |
       |--------|-------------------------------------------------------------|
  /24  | SPTRL2 |  (Spool in) Requested length                              |
       | SPTRNB |  (Spool out) Remaining no. of files of the current        |
       |        |  entry being processed                                    |
       |--------|-------------------------------------------------------------|
  /26  | SPTEL2 |  Effective length                                         |
       |--------|-------------------------------------------------------------|
```

/28	SPTST2	Status
/2A	SPTSC2	Zero or sector number if accessed to a DAD
/2C	SPTRNB SPTA10	(Spool in) Remaining no. of files of the current entry being processed FCT address of output device
/2E	SPTLFT	LFT address of current file to be output
/30	SPTSRP	Relative sector number of beginning of page of current output file
/32	SPTSAP	Sector number in DAD of beginning of page of current output file
/34	SPTSBP	Displacement of the Record of beginning of page of current output file
/36	SPTSRC	Current relative sector number of output file
/38	SPTSAC	Current sector number in DAD of output file
/3A	SPTSBC	Displacement of next Record of output file in blocking buffer

An explanation of some of these locations follows:

SPTDVA

0										10 15
E_o	O_p	C	D	R	N	B	S		EB	Device Address

E_o= 1 The event has occurred (Used for synchronisation, e.g. when waiting for an operator response

C = 1 Resume current I/O operations

D = 1 Cancel current output file

R = 1 Rewind current output file

N = 1 No format for the current output file

B = 1 Backspace to the beginning of the last page - for LP only (format character = /31)

S = 1 Start Spooling

O_p = 1 Operator intervention is required

EB = 1 End of Batch card read

SPTSTA

E_2	O_p	JB	EJ	BCP	ESJ	EOB		Dev. Type

0 15

E_2= 1 Operator commands for the spooling device (e.g. DM, DB or CR) are suspended until E_2 = 1.

JB = 1 A job card has been read.

EJ = 1 EOJ card has been read.

BCP = 1 BCP suspended. This occurs when attempting to unspool an empty file; it is reset to zero when a new job is spooled.

ESJ = 1 Submitted jobs in queue.

EOB = 1 End of batch received. Dev. type = /F Card reader
= /0 Line printer

SPTNBF Number of files already in the spooling queue. For CR it is the number of jobs to be processed. This value is incremented as each job is spooled in, and decremented by the BCP when a job is unspooled. For the LP, it is the number of files to be unspooled. E_1 is set to 1 if this number is non- zero, allowing synchronisation between spooled and unspooled programs.

SPTDAD DAD table address on which the device is spooled.

SPTCUR Pointer of the current job in the queue of jobs to be output (on the LP for example), or read by the BCP.

SPTNXT The address of the next free entry in the queue. Its initial value is 2, i.e. the first free word in sector 5 of the DAD D:SPCR. When the values of SPTNXT and SPTCUR arre equal, thereare no files in the queue and SPTNBF = zero. When either SPTCUR
or SPTNXT become equal to SPTLST, they are reinitialised.

SPTLST This is the highest value attainable by SPTCUR or SPTNXT. For the CR, it is the word address of the last location of sector 5 of the DAD D:SPCR.

SPTFC1 to SPTSC1

These locations constitute the ECB used either for reading a card or writing a record to the LP, PP or PL.

Bit 0 of SPTFC1 is the event bit.

SPTFC2 to SPTSC2
These locations constitute the ECB used by the program S:SPxx to access the disc file, where xx is the device name.
Bit 0 of SPTFC2 is the event bit.

SPTDFC DAD filecode: used to update the spool queue.

SPTRNB Remaining number of files in the entry of the queue being processed. E.g. in the case of the LP, it is the remaining number of files to be output for the current job.

T:MBX The Mailbox Table

A mailbox table is created in the system machine dynamic area whenever an LKM 52 request naming a new mailbox is received by MAS. The layout is is follows:

LOCATION	LABEL	CONTENTS
0 (CVTMBX)->	MBXLNK	Address of next mailbox in the chain; zero if this is the last
2	MBXNAM	Mailbox name
4	MBXLRN	Number of letters in the mailbox
6	MBXLTR	Address of the first letter
8	MBXRQN	Number of requests for a letter
UU /A	MBXRQA	Address of the first request

Each request for a letter is stored in a chain of request definition boxes of the following format:

(MBXRQA)-0->	Address of next request; zero if none
2	PCT address of user requesting program
4	Address of user request block
6	Scheduled Label address

Whenever a letter is sent, a letter definition box is created and chained to any other letter definition boxes created for letters directed at the same mailbox. The layout of these boxes is as follows:

(MBXLTR)-0->	Address of next letter; zero if none
2	Length of the letter
4	First word of the letter
6	

T:SEM The Semaphore Table

One of these semaphore tables is created in the system machine dynamic area each time a semaphore is declared by a user. They are added to the chain of all semaphore tables for that machine.
The layout is is follows:

LOCATION	LABEL	CONTENTS
0 (MCTSEM)->	SEMLNK	Address of next semaphore; zero if none
2	SEMNAM	Semaphore name
4	SEMVAL	Semaphore value
6	SEMPCT	PCT Address of chain of suspended programs - zero if none

The chain of suspended programs addressed by the location SEMPCT is linked via the location PCTSP (/24), thus:

```
                     PCT1                            PCT2
                --------------                  --------------
               |              |                |              |
               |              |                |              |
(SEMPCT)---->  |    PCTLNK    |                |    PLTLNK    |
               |--------------|                |--------------|
               |              |                |              |
               |              |                |              |
               |              |                |              |
               |--------------|                |--------------|
               |    PCTSP     |-------------->|    PCTSP     |     Zero if last
               |--------------|                |--------------|
               |              |                |              |
               |              |                |              |
               |              |                |              |
                --------------                  --------------
```

Secondary Load Module Tables

One of these tables is created within the dynamic area of the system machine and added to the chain of such tables whenver a secondary load module is declared. The start of the chain is pointed to by location CVTSLM (/84) of the CVT.

The format of these table is as follows:

LOCATION	LABEL	CONTENTS
0	SLMLNK	Next SLM table address, zero if none
2	SLMNAM	Sec. load module name
4		
6	SLMADR	Load address of SLM
8	SLMNBP	No. of pages of the secondary load module
/A	SLMREL	2 x relative page No. of the first page in MMU
/C	SLMMMU	Contents of MMU registers corresponding to these pages (length = (SLMNBP))

Data Window Tables

These define data areas obtained or released by the LKM 56 request, thus allowing the sharing of data areas between the programs of a machine. They are chained in the usual manner via location zero, the first one being addressed from location /36 (MCTDWD) of the MCT.
The layout is as follows:

LOCATION	LABEL	CONTENTS
0 (MCTDWD)->	DWDLNK	Address of next data window; zero if none
2	DWDNAM	Name of data window
4	DWDNBP	No. of pages of data window
6	DWDREL	2 x relative page No. of the first page in the MMU
8	DWDMMU	↑ Contents of the MMU registers, (DWDNBP) words ↓

where:
the length of the area starting at, and including DWDMMU, is equal to DWDNBP in length.

Short Timer Tables

These tables are created/deleted by an LKM 64 request. They are chained in the usual manner via location zero, the first one being addressed from location /3C (T:SHT) of the Real time clock table (T:RTC).

LOCATION	LABEL	CONTENTS
0 (T:SHT)->	SHTLNK	Next short timer block, or zero
2	SHTNAM	Name of short timer
4	SHTECB	ECB address
6	SHTPCT	PCT address
8	SHTLAB	Scheduled label address or zero
/A	SHTINI	Initial value of timer
/C	SHTCUR	Current value of timer
/E	SHTMCT	MCT address

Error Recording blocks

The anchor for these blocks is location /A0 of the CVT (CVTERL). The error logging facility is active, when in the system machine filecode /21 is assigned. This filecode should be assigned to the D:ERLG file, to which these blocks are written every minute (when avalaible).

All hardware errors on magnetic memory devices are recorded, except "Not Operable" and "Wrong Length".

At the same time, only one block can exist in memory for the specific device.

The layout of these blocks is as follows:

LOCATION	LABEL	CONTENTS
/0	ERLLNK	Link to the next block
/2 /4 /6	ERLDAT	Date
/8 /A /C	ERLTIM	Time
/E /10 /12	ERLPRO	Program name
/14 /16 /18	ERLMCT	Machine name
/1A	ERLDN	Device name
/1C /1D	ERLDA ERLURO	Device address User order
/1E	ERLCIO	CIO control word
/20	ERLCC	Cylinder number (DK only)
/22 /23	ERLH ERLSEC ERLFLS	Head number (DK only) Sector number (DK only) Sector number (FL only)
/24	ERLSTA	Status
/26	ERLRET	Number of retries
/28	ERLRQL	Requested length

The I/O ECB structure (System Routines)

A description of the other I/O ECBs will be found in the MAS Manual under the heading 'LKM 1 - I/O Requests'; the following diagram shows the structure of ECB used by system routines:

```
LOCATION    LABEL                    CONTENTS

            ------------------------------------------------------------
 -4        | ECBPCT |  PCT Address (only if bit 3 in ECBFC = 1)        |
           |--------|--------------------------------------------------|
 -2        | ECBMCT |  MCT Address (only if bit 2 in ECBFC = 1)        |
           |--------|--------------------------------------------------|
  0        | ECBFC  | E | O | F | U |        |  Filecode               |
            ------------------------------------------------------------
bits                 0   1   2   3   4      7  8                      15
            ------------------------------------------------------------
  2        | ECBBF  |  Record Area                                     |
           |--------|--------------------------------------------------|
  4        | ECBRL  |  Requested length                                |
           |--------|--------------------------------------------------|
  6        | ECBEL  |  Effective length                                |
           |--------|--------------------------------------------------|
  8        | ECBST  |  Returned Status                                 |
           |--------|--------------------------------------------------|
 10        | ECBST  |  a)  Tabulation Table Address                    |
           |        |  b)  Sector Number of disc                       |
           |        |  c)  Filecode /Cx: cylinder number               |
           |--------|--------------------------------------------------|
 12        | ECBHD  |  a)  Timeout   or b)  Head and Sector number     |
            ------------------------------------------------------------
```

Where:

- ECBFC is used for event handling:

bit 0 = 1: Event has occurred (I/O complete.
bit 0 = 0: Event has not yet occurred.

bit 1 = 0.

bit 2 = 1: File code is a user filecode; ECBMCT points to the MCT of the machine in which the filecode is assigned.
bit 2 = 0: Filecode is a system filecode; ECBMCT not used.

bit 3 = 1: The ECB is a system ECB, but the buffer is located in the user area; ECBPCT points to the PCT of the user program.
bit 3 = 0: The ECB is a system ECB; ECBPCT not used.

bits 8 to 15: Filecode.

- ECBST is the returned status:

a) Zero : The operation terminated satisfactorily.
b) Positive : The operation was completed but the following conditions were encountered:

1	EOF encountered (Read)
2	EOS encountered (Read)
4	Data Error
8	Incorrect Length
/10	End of tape, end of media, request done
/20	Beginning of tape
/40	End of tape reached but the current record has been read or written (warning signal)
/80	EOV mark detected.

c) Negative (bit 0 set)

Bit 1 = 0: bits 2-15 indicate the hardware status

Bit 1 = 1:

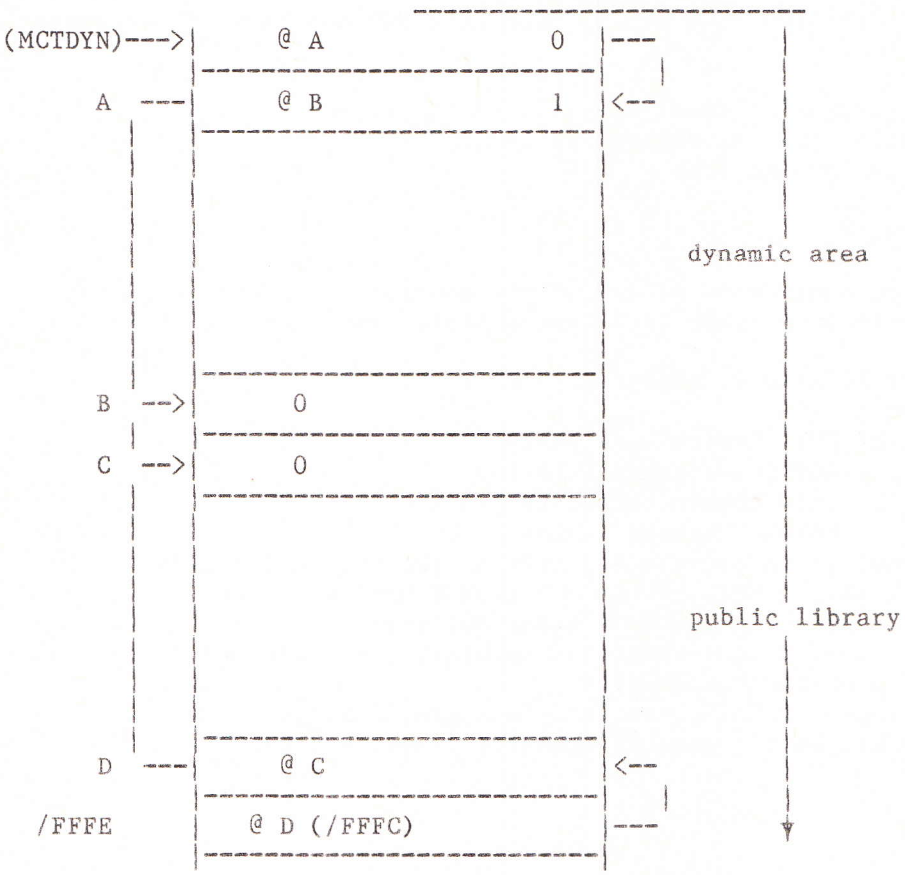
/C001	Illegal File Code or File Code not assigned
/C002	Device attached to other programs
/C008	Buffer address, or requested length invalid
/C010	Function unknown or incompatible with the Device or File
/C020	Write protection on Disc File
/C040	End of media: current operation aborted
/C080	Time-out
/C100	Disc queue overflow
/C200	Dynamic Buffer overflow; no disc blocking buffer free
/C400	Blocking overflow (No free granule).
/C800	Sector address out of dad (grantb overwritten)

The Dynamic Area

The dynamic area of a foreground machine is defined by means of the SCL CMA command. For the batch machine the dynamic area is implicitly defined and initialised after loading a batch program, and consists of the remainder of the machine unoccupied by the program.

When the system allocates memory blocks, one word more is provided than is required; this word is used to address the next block. Bit 15 of this word is used to denote either that the block is free (bit 15=1) or that it is already allocated (bit 15=0).

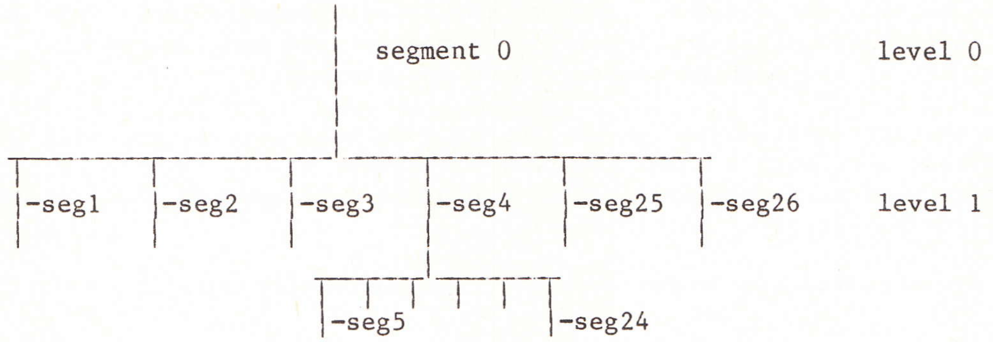
The dynamic area can be represented diagrammatically thus:



Remarks:

@ D always = /FFFC

The supervisor overlay tree is as follows:



When the system is running in a system task, the MMU consists of the pages of:

- segment 0
- current segment of level 1
- (eventually) current segment of level 2
- the System Dynamic Area

segment 0

Segment 0 is the communication window of the monitor. Its contents are all permanent information, such as hardware tables and common routines. It contains:

- interrupt locations, address 0-/7E
- System stack
- Tables like DWT (device work table)
DCT (disc control table)
Data communication tables
Network access tables
- MCT of the system machine and PCTs of the system programs
- T:CORE, T:SLT, T:ELIG, T:LKM and T:RMAC tables
- Tables for managing multiple transient areas
- Transient area 0 into which the segments are loaded from the D:MSEG DAD or the D:MASG file
- System program X:MASG, the transient area 0 loader
- Common routines, in general, modules starting with 'R':

The Layout of Permanent Memory Allocation

/00	Addresses of Interrupt Routines and of I:PARA	
/7E	@ I:TRAP	
/80	/0000	
/82	@ CVT	
/84	CPU type	CPUTYP = 0 : P857 2 : P858 non ? 4 : P859 6 : P854
/86	RESERVED	
/88	MCT of the System Machine	
/BC	P:CUR (@ PCT of Current Program)	
/BE	T:SGDK (@ First Sector of Segment 0)	
/C0	T:SCUR (Segment number in memory)	
/C2	CNTOVL (Number of programs suspended on System Dynamic Area overflow)	
/C4	I:PARA routine: STR A1,A15 LDK A1.0 CF A15,R:HALT	(unknown interrupt)
/CC	R:HALT routine: INH ABL DMSYSR	If at SYSGEN DUMPSA=YES, put /12F0 in A0 and RUN.
/D2	Idle Task: S:IDLE DLC 31 RB S:IDLE	
/D4	~~Not used~~ T: SCUR	
/100	Stack	
/300	Transient area of supervisor	
/12F0	Routines and tables	

Segment 1

Segment 1 is the nucleus of the monitor.
It contains:

- Internal interrupt processing routines (LKM, RTC, etc)
- Dispatcher
- LKM execution routines like:
  - activate
  - exit
  - wait
  - set event
  - get/free buffer
- Drivers
- Datacommunication handler (LKM 8)
- Network access routines

Segment 2

Segment 2 contains some monitor tasks.
These tasks are:

- Allocate granule (X:ALGR)
- Swap handler (X:SWIO)
- Timer handler (X:RTC)
- Spool handlers (like X:LP07)
- Operator communication handler (X:OCOM)
- Dump handler (X:DUMP)
- Stand-alone dump routine (DUMPSA)

Segment 3

Segment 3 contains the I/O handler (X:IO), the disk file management handler and a part of TDFM.

Segment 4, and 5-24

Segment 4 contains TDFM routines and the system task X:TDFM. This task is the supervisor of the multiple transient areas. It handles the loading of the level 2 segments 5-24. These segments can be loaded into 4 transient areas, each having a length of 1 page.
The segments in these areas are overwritten via the LRU (least recently used) algorithm. Level 2 segments all contain TDFM routines. The segments are loaded from the MASR load module, so not from D:MSEG or D:MASG like X:MASG does. Due to this loading, the MASR must be a consecutive file!

Segment 25

In segment 25, the LKM routines are located, which can be chosen as core resident during sysgen. These LKMs run under the system task X:USVC. The LKMs are:

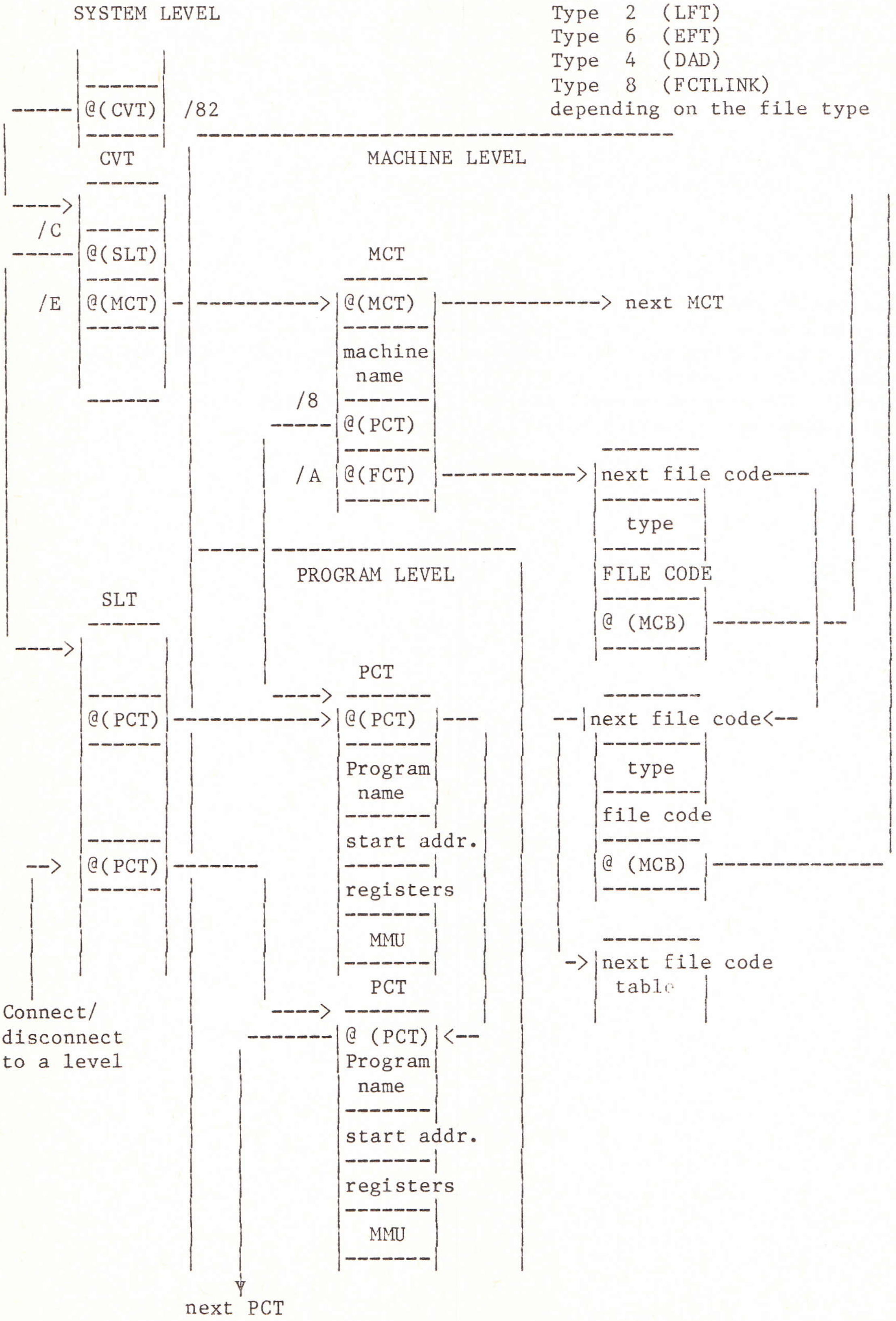
- Get date and time
- Semaphore
- Get page
- Connect a secondary load module
- Send/receive letter

Segment 26

Segment 26 contains the monitor initialisation routines, like INIMON. It gets control from the monitor loader and returns to segment 0 after monitor initialisaton. After initialisation, the pages used by this segment are released.
REMARK: The segment numbers are not fixed. They are dependent on the System Generation parameters!

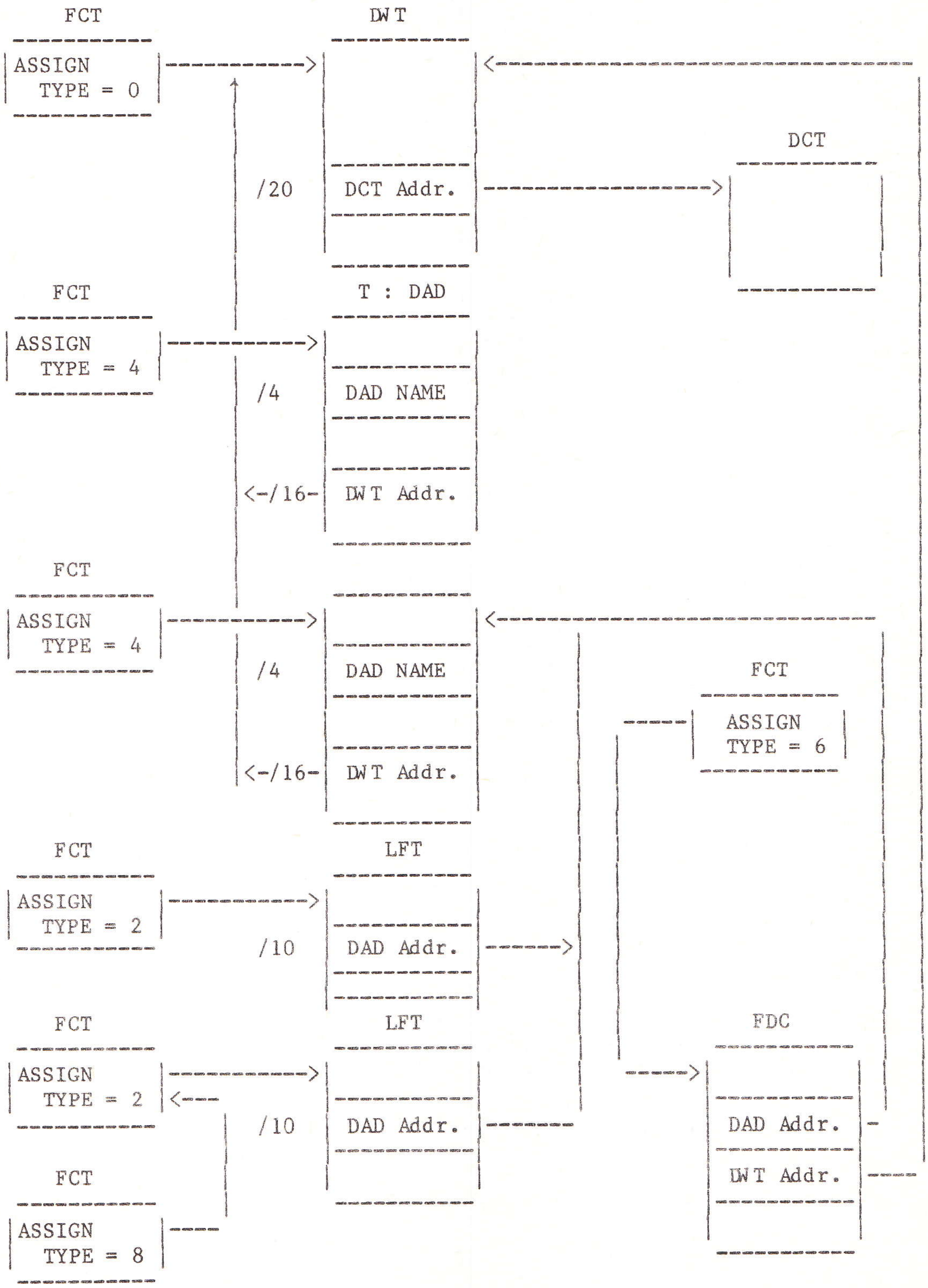
The Chaining of Supervisor Blocks

The MCBs can be: (I/O Control Blocks)
Type 0 (DWT)
Type 2 (LFT)
Type 6 (EFT)
Type 4 (DAD)
Type 8 (FCTLINK)
depending on the file type



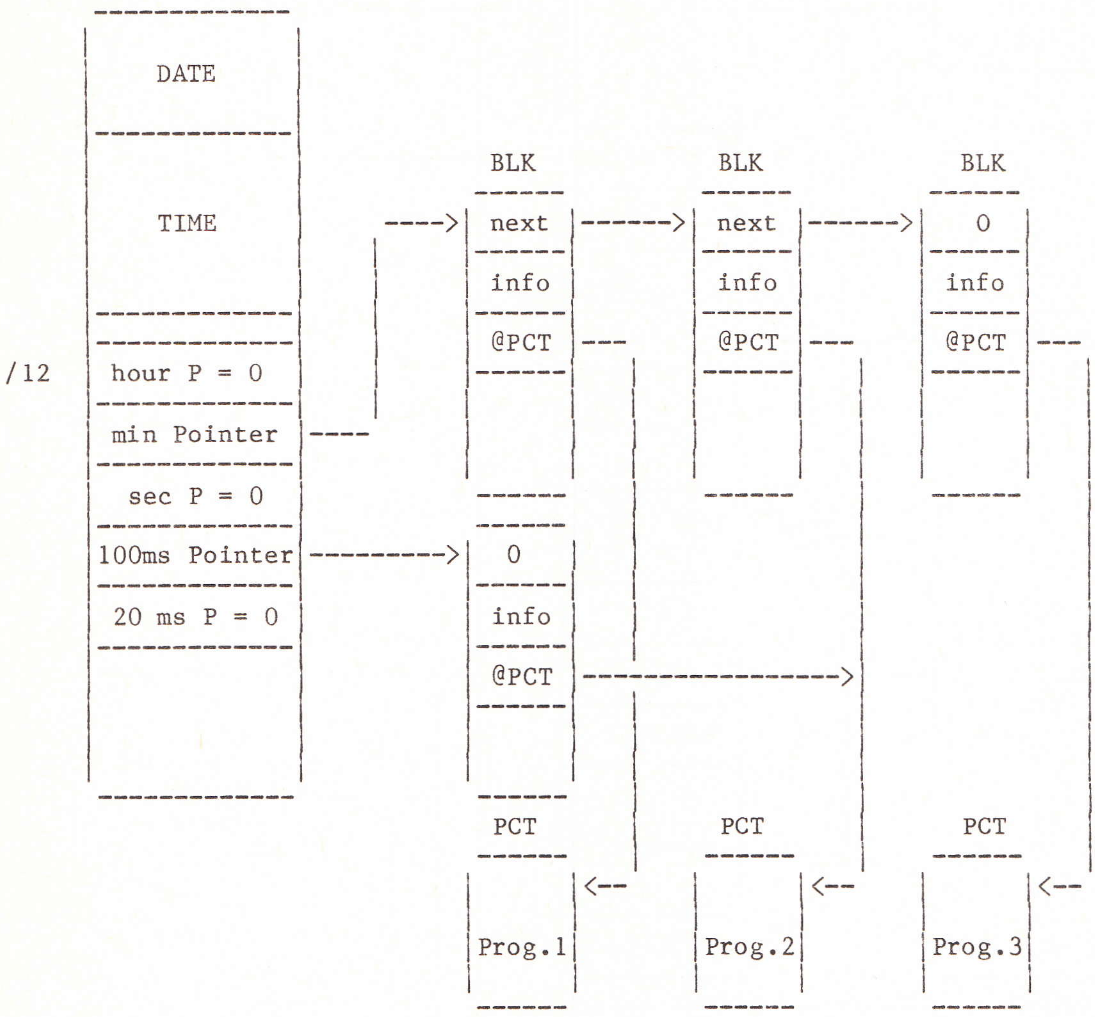
Note: @ = 'address of'

The Chaining of Disc Control Blocks



Chaining of Timer Blocks

Address of T:RTC is contained in the first word of the CVT



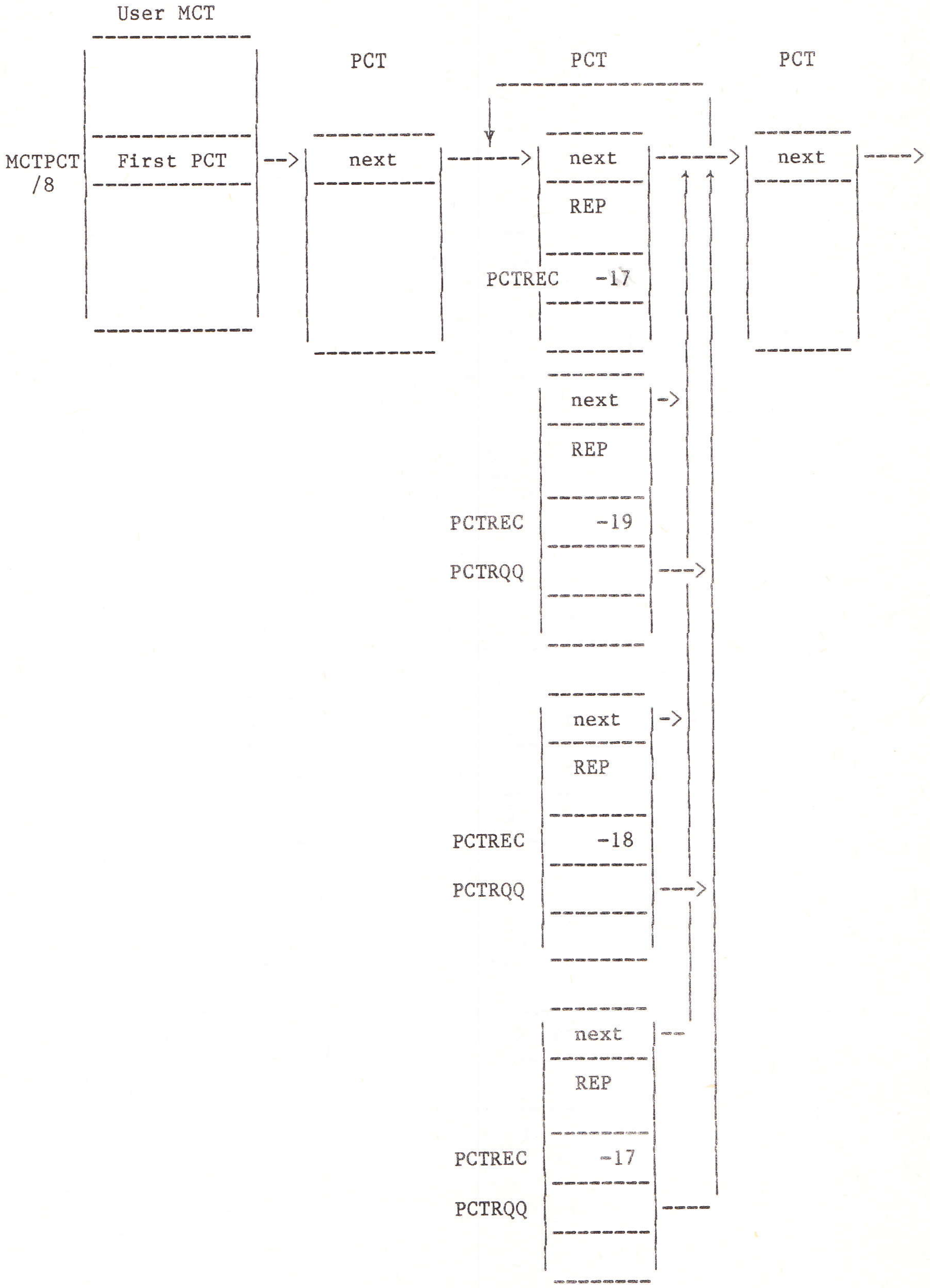
In this example:

Prog 1 is connected to timer 1 (minutes)
Prog 2 is connected to timer 1 and 3 (100ms)
Prog 3 is connected to timer 1

No programs are connected to other timers.

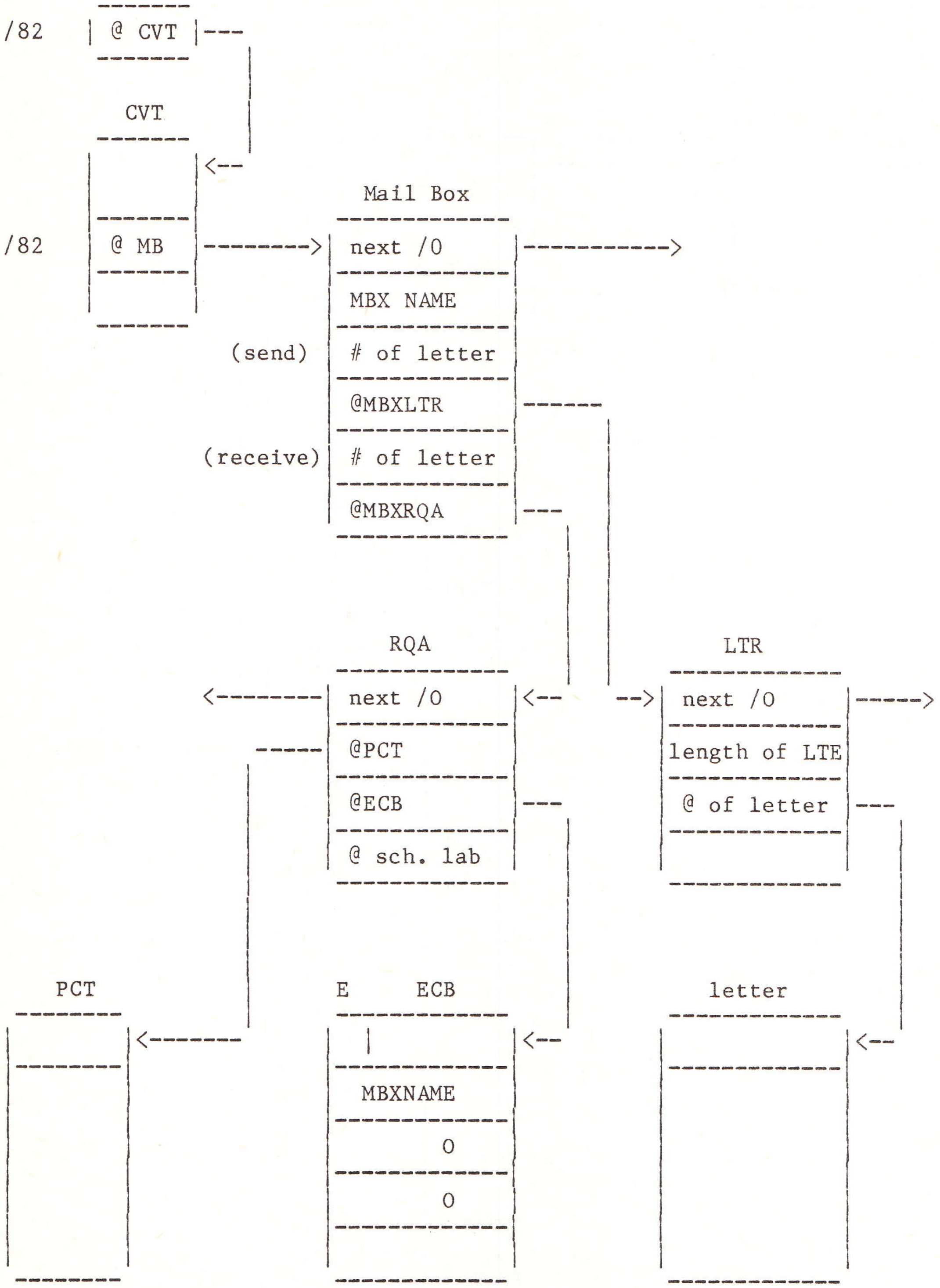
Note: @ = "address of".

Block Chaining for Re-entrant Programs



Note: The initial value of PCTREC is -20

Block Chaining for Letters



Block Chaining for Semaphore

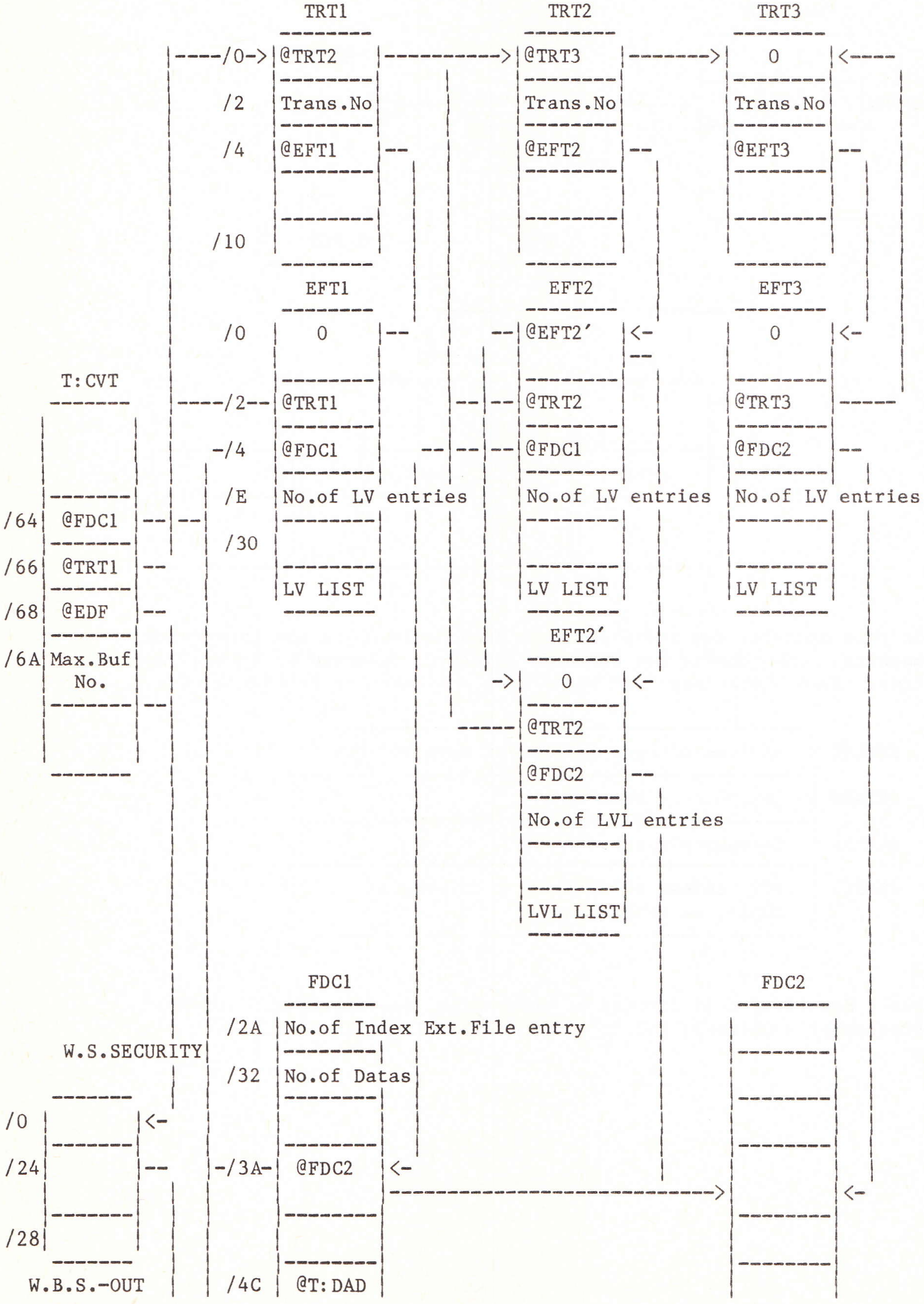
```
                 User MCT
              --------------
             |              |           SEMBLK              SEMBLK
    /34      |--------------|        ------------        -----------
    MCTSEM   |   @ first    | ------>|    next    |----->|    0      |
             |--------------|        |------------|      |-----------|
             |              |        |   S   1    |      |   S   2   |
             |              |        |------------|      |-----------|
             |              |        |     1      |      |    -3     |
              --------------         |------------|      |-----------|
                                     |     0      |   ---|  @ PCT    |
                                      ------------    |   -----------
                                                      |
               ---------------------------------------
              |
              |          PCT              PCT               PCT
              |      ------------     ------------      ------------
               ---->|            |   |            |    |            |
                    |------------|   |------------|    |------------|
          PCTSP     |    next    |-->|    next    |--->|     0      |
                    |------------|   |------------|    |------------|
                    |            |   |            |    |            |
                     ------------     ------------      ------------
```

In this example, two semaphore have been declared in the foreground machine, consequently two SEMBLOKs have been reserved in System Dynamic Area. Such blocks are four words long and have the following layout:

SEMLNK	Address of next SEMBLK, or zero if none
SEMNAM	Semaphore's name
SEMVAL	Semaphore's value
SEMPCT	PCT address of the suspended program chain, or zero if none

For a description of the use of semaphores, see Appendix C of P800 Programmer's Guide 3, Vol I, under the heading 'LKM 55'.

Example of Control Block Chaining



TRT1
/0 @TRT2
/2 Trans.No
/4 @EFT1
/10

TRT2
@TRT3
Trans.No
@EFT2

TRT3
0
Trans.No
@EFT3

EFT1
/0 0
/2 @TRT1
/4 @FDC1
/E No.of LV entries
/30
LV LIST

EFT2
@EFT2'
@TRT2
@FDC1
No.of LV entries
LV LIST

EFT3
0
@TRT3
@FDC2
No.of LV entries
LV LIST

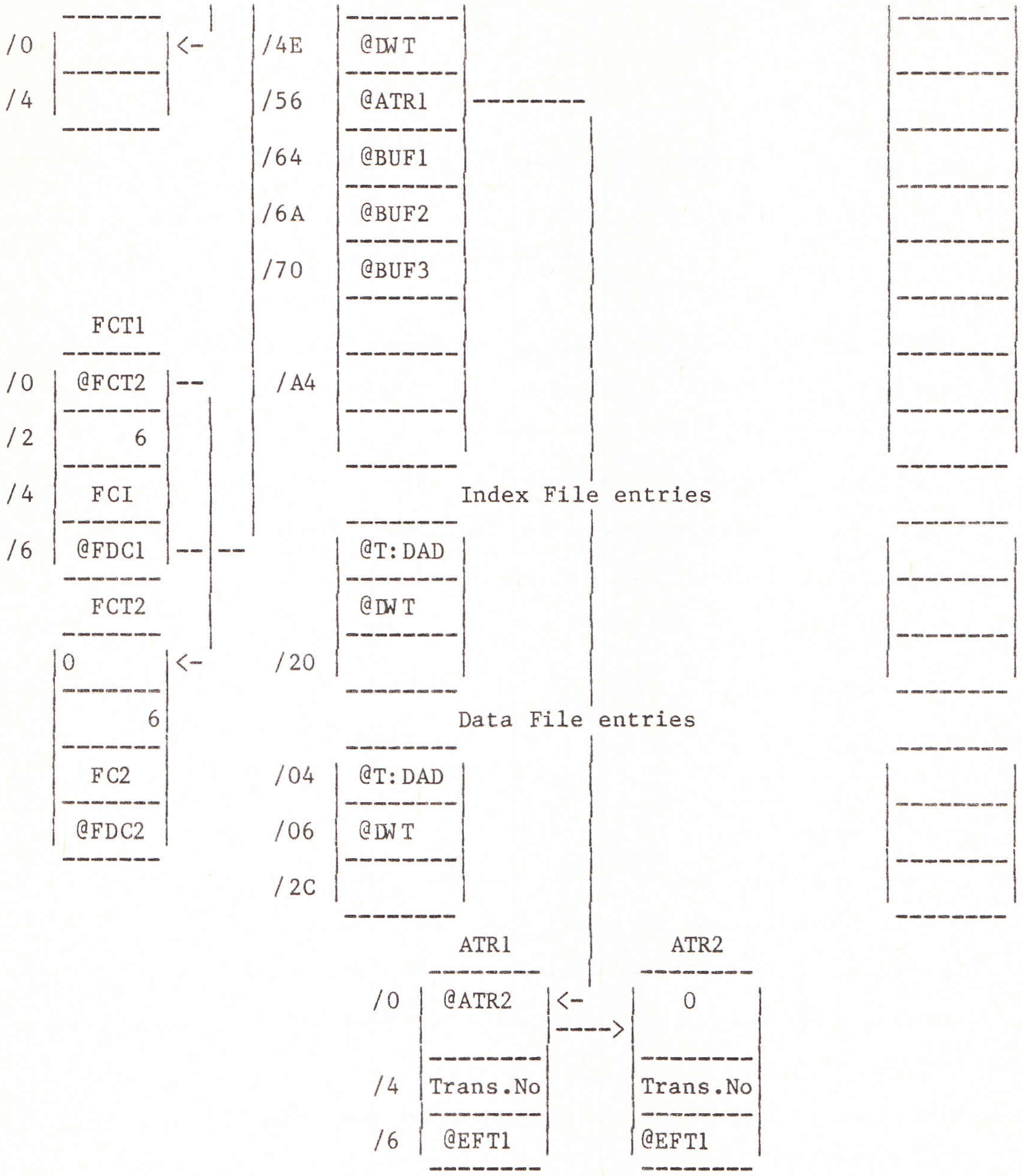
EFT2'
0
@TRT2
@FDC2
No.of LVL entries
LVL LIST

T:CVT
/64 @FDC1
/66 @TRT1
/68 @EDF
/6A Max.Buf No.

W.S.SECURITY
/0
/24
/28
W.B.S.-OUT

FDC1
/2A No.of Index Ext.File entry
/32 No.of Datas
/3A @FDC2
/4C @T:DAD

FDC2


/0
/4
/4E
@DWT
/56
@ATR1
/64
@BUF1
/6A
@BUF2
/70
@BUF3
FCT1
/0
@FCT2
/A4
/2
6
/4
FCI
Index File entries
/6
@FDC1
@T:DAD
@DWT
FCT2
0
/20
6
Data File entries
FC2
/04
@T:DAD
@FDC2
/06
@DWT
/2C
ATR1
ATR2
/0
@ATR2
0
/4
Trans.No
Trans.No
/6
@EFT1
@EFT1

T:MDG The declared middleground programs table

The MDG describes the characteristics of the load module (segmented or not) to be activated as a middleground program.

The layout of this table is as follows,

MCTMDG

LOCATION	LABEL	CONTENTS
0	MDGLNK	Address of next MDG in chain, 0 if last
2	MDGNAM	
4		/ Program name /
6		
8	MDGSW1	Initial image address in D:CI
/A	MDGLMD	DAD file code of load module
/C	MDGDAD	T:DAD address
/E	MDGSAD	Start address of the program
/10	MDGLMS	Sector address in the DAD
/12	MDGREG	Core Region size

The table is updated by:

The MDG command (creation of the block)

Explanation of the labels,

MDGLNK address of the next entry in the chain
a new block is chained on the last one

MDGNAM the name of the program

MDGSW1 address of the initial image of the middleground program on the Core Image DAD (D:CI)

MDGLMD bits 0-7 contain the number of pages ~~for~~ of the root for an overlaid middleground program.
bits 8-15 contain the DAD file code of the load module

MDGDAD address of the T:DAD corresponding to the DAD from which the load module has been loaded

MDGSAD start address of the program (virtual)

MDGLMS sector address of load module in the DAD
bit 0=1 ~~non~~ consecutive granules. It points at the second sector of the first granule of the file
bit 0=0 for non consecutive file, it points at the sector GRANTB

MDGREG Contains the core region size (number of pages) of the program

1.09.98

Chapter 2

DEBUGGING USER DUMPS

Using the tables described in section 1 in conjunction with the error codes listed in Appendix D, it is usually a straightforward matter to determine the cause of program or system errors. Starting from the CVT address at /82, the other monitor control tables and their contents can be traced. Almost without exception, the first step is to determine the location within the coredump of the instruction causing the error. This can be found by first finding the PCT address for the program concerned, and from this the MMU save area address and the program's register save area address.

The register save area contains, in addition to the other registers, the program status word (PSW) and the location counter (or P register), which contains the address of the next instruction to be executed.

Converting Logical to Absolute Addresses

The addresses in the monitor control tables and in the P register are logical addresses which have the following format:

```
                -------------------------------------------
                | page number | displacement within the page |
                -------------------------------------------
         bits     0         3   4                          15
```

The page number can be used as a displacement pointer for the 16 word MMU table, each word of which contains, in bits 0-5 (+ 14-15 for P854, P859), the address of the corresponding page (e.g. word 0 contains the address of page 0, word 1 that of page 1, etc.).

The actual format of an MMU entry is as follows:

Bit	Contents
0-5	Physical page address
6	Set = page fault
7	Set = read only page
8	Set = page that has been modified
9-13	Not used
14&15	Leading bits of the physical page address (P854, P859)

Example 1

Suppose we wish to find the absolute address corresponding to the logical address /2CA6.

Bits 0-3 contain the value 2, and this is therefore the page number. The re mainder, /CA6, is the displacement within the page.

Now, suppose that word 3 of the MMU had the following contents:

```
             --------------------------------------------
            | 0 1 1 0 0 1 |               |    |  0  |  0 |
             --------------------------------------------
   bits       0         5   6               13   14    15
```

(ignoring bits 6-13 as irrelevant)

The value in bits 0-5 is /19 (= 011001 and 00011001 fo P854/P859), and since absolute addresses are 18 bits (as for P854/P859) in length the page address is thus /19000.

To this we now add the page displacement and find that the absolute address corresponding to /2CA6 is /19CA6.

Example 2

Question: Why does my program remain in a wait state? I must find the last instruction executed in the program and the corresponding Event Control Block.

Explanation: - MAIN is the name of my program, its PCT address is /B71A (see the example of a dump given below). Its PCTSTA (/0800) means program in wait state. Its save registers field address is /B75E. At this address we get instruction counter (P reg), PSW and registers A1-A14. P reg (=/F77A) points to the next instruction to be executed in the program. A8 (=F4A0) points to an event control block. Because these addresses are given as virtual addresses, we have to convert them into absolute addresses; therefore, we need the saved MMU area.

The address of this is /b468 and the contents are:

page number	value of corresponding registers
14 (/E)	/4024
15 (/F)	/4480

The address of the next instruction is

/F77A meaning : /F page number
/77A displacement

/F corresponds to /4480 *

This value can be written:

$$\underbrace{0100\quad 0100\quad 1000\quad 0000}_{A}$$

A gives physical address of the page: /1100 =(00)010001
displacement / 77A

absolute address = $/\overline{1177A}$

Suppose for P854, P859 A would be /4481

This value can be written:

0100 0100 1000 0001

A gives physical address of the page: /5100 0(01)10001

displacement / 77A

absolute address =$/\overline{5177A}$

```
 CORE DUMP*** FROM 0B000 TO  20000***
 0B000  3963 8144 B00E 8141  AC0A 8F20 AC78 0004  0003 0002 0000 BB4A  0006 1118 123C 133C
 0B020  140A 1505 9E19 D143  0000 D243 0002 D343  0004 D443 0006 D543  0008 D643 000A 5F4A
 0B040  8720 FFFF 8741 AC0A  5F50 8441 B068 8420  B070 0306 1C01 C820  000A 1130 E131 0100
 0B060  1B01 5910 B9B0 8F20  AFEA 3030 3030 3031  B072 B07C B07E 0004  00F1 7A50 B086 B088
 0B080  0000 0001 7202 B090  B092 0000 0002 7338  B09A B09C 0000 0003  73D2 B0A4 B0A6 0000
 0B0A0  0004 724E B0AE B0B0  0000 0005 7388 B0B8  B0BA 0000 00E0 724E  B0C2 B0C4 0000 00EF
 0B0C0  7202 B0CC B0CE 0000  00C3 741C B0D6 B0D8  0000 00C1 7462 B0E0  B0E2 0000 00C2 74A8
 0B0E0  B0EA B0EC 0000 00C0  74EE B0F4 B0F6 0000  00C4 7534 B0FE B100  0000 00C5 757A B108
 0B100  B10A 0000 00C6 764C  B112 B114 0000 00C7  7692 B11C B11E 0000  00C8 75C0 B126 B128
 0B120  0000 00C9 7606 B130  B132 0004 00F0 7A00  B13A 0000 0004 00F6  7A00 B2CE 0190 2020
 0B140  2020 4C41 4245 4C20  3D20 444F 5352 454C  2030 2020 2020 2020  2020 2020 4441 5445
 0B160  203D 2020 3238 2F30  372F 3738 2020 2020  2020 2020 5041 434B  204E 4252 203D 2020
 0B180  4632 3220 2020 2020  2020 2020 2020 0066  0000 0000 0000 0000  0000 0000 0000 0000
 0B1A0  0000 0000 0000 0000  0000 03FF FFFF FFFF  FFFF FFFF FFFF FFFF  FFFF FFFF FFFF FFFF
** 0B1C0  TO   0B1DE  CONTAIN X'FFFF ' **
 0B1E0  FFFF FFFF FFFF FFFF  FFFF FFFF FFFF FFFF  FFFF FFFF FFF0 0000  2020 2020 2020 2020
** 0B200  TO   0B2BE  CONTAIN X'2020 ' **
 0B2C0  2020 2020 2020 2020  2020 2020 2020 B2E0  0000 0400 FC46 F8D4  5020 0000 0000 0000
 0B2E0  B2EA B2EC 0002 002F  B2FA B2F4 0000 0002  0034 B57E B2F9 09A8  B33E 0086 F476 F536
 0B300  003E B71A 0000 8A90  0000 7A00 09A9 0000  09AA B930 0048 8400  00E0 B574 0000 0004
 0B320  0000 B444 20E0 B58E  003C B372 68C6 6F58  0000 0000 0005 0000  00F0 2020 B2E2 B34E
 0B340  B350 0500 DBC6 DBB8  B2DA 0000 0000 B360  0000 0400 D248 CED6  0001 0220 B371 0000
 0B360  B371 300D 463A 2020  2020 0000 0000 B371  B4AA 0000 464F 5254  2020 9456 B3B6 B3DC
 0B380  8000 8400 0074 B440  B440 0000 0000 0000  0000 0000 0094 0000  0000 0000 0000 0000
 0B3A0  0000 0000 0000 B3FC  0000 0000 0000 0000  0000 0000 0000 953C  FEC4 59A0 59C8 FC40
 0B3C0  9538 59A0 0000 0000  B440 3FF8 B58E 0000  B42A 0000 0000 0000  0000 0000 0000 0400
 0B3E0  0800 0C00 1000 1400  1800 1C00 2000 2400  2800 2C00 3000 3400  3800 3C00 0000 FC40
** 0B400  TO   0B41E  CONTAIN X'0000 ' **
 0B420  0000 0001 0000 0000  0000 0000 464F 5254  2020 B71A B4AC 0007  E000 B464 0000 0000
 0B440  9201 B42A A002 B58E  0005 0003 0000 0000  0000 0000 B71A FFD9  FFFD FFF6 0000 0000
 0B460  0000 0000 0001 FFFE  0222 0237 0236 0235  0234 0234 022C 022A  0229 0226 0226 0225
 0B480  0224 0224 4024 4480  DFFE 0240 0240 0240  0240 4840 4C40 5040  5440 5840 5C40 6040
 0B4A0  6440 6881 6C80 4080  4440 B4B4 B4B6 0000  0001 7202 B4BE B4C0  0000 0002 7338 B4C8
 0B4C0  B4CA 0000 0050 724E  B4D2 B4D4 0000 0052  7338 B4DC B4DE 0000  00C0 74EE B4E6 B4E8
 0B4E0  0000 00C1 7462 B4F0  B4F2 0000 00C2 74A8  B4FA B4FC 0004 00F0  7A00 B504 B506 0004
 0B500  00F2 B6C8 B50E B53A  0004 00F3 B510 B538  B544 00C3 5254 4C46  494C 0010 019A 0014
 0B520  0064 0008 0002 741C  0001 0005 000A 01FF  FFFF FFFF FFFF FFFF  B542 B574 0004 00F4
 0B540  B544 B572 0000 00C3  5254 4C56 4552 0010  019A 001E 0078 0008  0002 741C 0001 0005
 0B560  0010 3FFF FFFF FFFF  FFFF FFFF FFFF FFFF  FF00 B57C B2E2 0000  00E0 7202 B5C2 0086
 0B580  CD32 E046 003E B87E  0000 8A90 0000 B6C8  0221 0001 0223 BACC  0018 BACC 0222 0015
 0B5A0  0000 0162 0000 2020  2020 2020 0010 2020  6FDC 6F58 0000 2020  0005 0000 00F2 6C0E
 0B5C0  B2EC B6C7 80EF B5D0  0026 0027 0000 D900  3030 0D0A 4D41 433A  464F 5254 2020 2C50
 0B5E0  524F 3A20 5254 464C  3420 2050 5345 0D0A  5041 5553 4520 3120  3D2F B6C7 3030 2C41
 0B600  3220 3D2F 4530 3436  2C41 3320 3D2F 3030  3030 2C41 3420 3D2F  4433 3638 2C41 3520
 0B620  3D2F 3030 3230 2020  2041 3620 3D2F 3030  3030 2C41 3720 3D2F  3030 3836 2C41 3820
 0B640  3D2F 4344 3332 2C41  3920 3D2F 3030 3030  2C41 3130 3D2F 3030  3030 2C41 3131 3D2F
 0B660  3030 3030 2C41 3132  3D2F 3030 3030 2020  2041 3133 3D2F 4341  4132 2C41 3134 3D2F
 0B680  4342 3838 2C46 5231  3D2F 3030 3030 2C46  5232 3D2F 3030 3030  2C46 5233 3D2F 3030
 0B6A0  3030 0000 B6C7 0000  8400 0002 B4B6 7338  0000 0086 A2EA 2002  B5DE 0048 5C90 0000
 0B6C0  0000 B6C7 0000 B718  B510 00C3 5254 4C50  524F 0010 019A 0064  0000 0008 0002 741C
 0B6E0  0001 0005 0032 0000  0000 0000 0000 0000  1000 0000 0000 1000  5C00 3FFF FFFF FFFF
 0B700  FFFF FFFF FFFF FFFF  FFFF FFFF FFFF FFFF  FFFF FFFF FFFF FFFF  B7CA B7CC 4D41 494E
 0B720  5020 F628 B75E B468  0800 0400 002C F4A0  0000 0000 0000 0000  0000 0078 B42A 0000
 0B740  0000 00F2 0259 FFFC  2020 0000 F3EA B784  0096 B2D0 0000 4C50  524F 0510 019A F77A
 0B760  FEC5 0000 F530 0000  3132 0000 0000 0000  F4A0 002D 0001 F560  F74E 0000 F626 0000
 0B780  0000 0000 0000 FC41  0000 0000 0000 0000  0000 0000 0000 0000  0000 0000 0000 0000
 0B7A0  0000 0000 0000 0000  0000 0007 0000 0000  0000 0000 0000 0000  0000 0000 0000 0000
 0B7C0  0000 0000 0000 0000  0000 B87C B87E 5254  464C 3430 DD42 B810  B48A 8000 0400 002E
 0B7E0  DF0A C56E B87E 0000  0000 0000 0000 B42A  0000 0000 00F2 02C9  DFFC 0000 0000 DD26
 11400   0008 0000 0000 8001   F422 0008 0009 0000   0001 F41A 0008 0000   0000 2020 4D41 3039
 11420   3030 2020 4D41 3034   3030 2020 4D41 3035   3430 0001 F43C 0008   0000 0000 2020 4D41
 11440   3038 3030 8001 F458   0008 0009 0000 0001   F460 0008 0000 0000   2020 4D41 3032 3230
 11460   2020 4D41 3036 3030   3A45 4F50 8050 F496   0004 0004 0000 802F   F4FA 003E 003E 0000
 11480   0000 012F 00F0 8000   0000 0000 5546 0000   0000 0000 0000 3A45   4F50 5254 464C 3420
 114A0   0000 F4A4 3404 0A2D   0000 0000 0000 0000   0000 0000 0000 0000   0000 0000 0000 0000
** 114C0   TO    114DE   CONTAIN X'0000 '  **
 114E0   0000 0000 0000 0000   0000 0000 0000 0000   0000 0000 0000 0000   F4FA 204C 4B4D 3132
 11500   3135 3038 3738 3033   3031 3030 0000 4F04   F4A0 3132 0000 0000   0000 F4A0 002D 0001
 11520   5254 464C 3420 0000   F4A4 0000 0000 0000   4D41 3032 3230 2020   0001 F542 000C 0000
 11540   0000 2020 4649 4E20   4445 2052 554E 2020   444E 2020 4D41 494E   2020 3132 3230 2020
 11560   4D41 3032 3230 2020   4D41 3032 3130 4142   4D41 3032 3230 4142   4D41 3036 3030 4142
 11580   B7CC 4D41 494E 5020   F628 B75E B468 0002   0400 002C 0000 0000   0000 0000 0000 0001
 115A0   0079 B42A 0000 0000   00F2 0259 FFFC 2020   0000 F3EA B784 0096   B2D0 0000 4C50 524F
 115C0   0510 019A 0000 464F   5254 2020 B71A B4AC   0007 E000 B464 0000   0000 9600 B42A 20E0
 115E0   B58E 0048 0006 0000   0000 0000 0000 0000   FFD7 FFFD FFF6 0000   0000 0000 0000 0000
** 11600   TO    1161F   CONTAIN X'0000 '  **
 11620   0000 0000 FEC1 F74C   8620 FFFC 86A0 F626   F6A1 FC30 80A0 F580   2804 0023 871C 5002
 11640   207F 81C2 0012 91A0   0001 0500 82A0 0000   80A0 F46C 0782 2804   0001 8240 F496 EA40
 11660   F468 540C 8240 F498   EA40 F46A 5046 207F   1604 8240 F496 A220   FF03 8259 F4A4 8240
 11680   F498 EA20 2020 5014   8241 F692 B9A0 3030   3030 3130 8420 F69C   8F20 FD4C 0203 E259
 116A0   F4A6 EA20 0001 5C58   E258 F4A5 2A02 E259   F4A5 5F64 80A0 F482   2804 0017 EF20 0000
 116C0   5002 207F 8641 F4F6   8641 F4F4 8706 E759   F4A7 80A0 F49A E258   F4A4 E241 F49E E258
 116E0   F4A5 2A04 E259 F4A5   2804 0014 EF20 0000   8C20 F848 0786 80A0   F444 2804 0001 8720
 11700   F49A 80A0 F4A0 8320   F4A0 2804 000C EF20   0000 8C20 F868 8440   F55A 8441 F4FE 92A0
 11720   0001 83A0 F560 84A0   F74E BD41 F50C 8120   F49A 8220 F520 030A   F6A1 F8AE 810E 8220
 11740   F530 0308 F6A1 F8AE   F6A1 FF70 8F12 ED20   0003 5038 EE20 0000   5008 91A0 0007 1E04
 11760   5F96 EAA0 0000 50B4   0786 80A0 F406 2804   0001 80A0 F4A0 2804   0002 0200 8241 F4A0
 11780   8640 F4F4 E258 F4A5   2280 5430 1E04 EE40   F4F4 500C EE20 FFFC   5C16 8640 F4F6 5F12
 117A0   8140 F4F4 500A 1904   8141 F4F4 8F20 F768   8140 F4F6 8141 F4F4   8F20 F768 9AA0 0001
 117C0   E258 F4A4 E241 F49E   0786 80A0 F3FC 2804   0001 E258 F4A5 2203   E259 F4A5 2202 5424
 117E0   E258 F4A6 9A20 0001   E259 F4A6 5016 EA20   0001 510A E258 F4A5   2A02 E259 F4A5 0503
 11800   8F20 F6FE 80A0 F49A   E758 F4A7 2804 0015   EF20 0000 5470 EAA0   0000 5C90 8220 F542
 11820   8241 F53A 0786 80A0   F538 2804 0001 F6A1   FE34 8220 F54E 8241   F53A 0786 80A0 F538
 11840   2804 0001 2804 0003   0786 80A0 F432 2804   0001 8440 F55C 8441   F4FE 83A0 F568 0508
 11860   84A0 F74E 8F20 F72A   0786 80A0 F44E 2804   0001 8440 F55A 8441   F4FE 83A0 F570 84A0
 11880   F74E 8F20 F72A 0786   80A0 F410 2804 0001   8440 F55E 8441 F4FE   83A0 F578 84A0 F816
 118A0   8F20 F72A 4649 4E20   4D41 494E 2020 8441   F8C4 910C 920C 1901   1A01 E424 E429 1B01
 118C0   5C0C 8420 3132 F03A   FD02 0001 0000 0008   0000 0000 0000 0000   0000 0000 0000 0000
 118E0   0000 0000 FFFC 0000   0000 0000 0000 0000   0000 0000 0000 0000   0000 0000 0000 0000
** 11900   TO    1197E   CONTAIN X'0000 '  **
 11980   0000 0000 0000 0000   0000 0000 0000 0000   0000 0000 0000 0000   0000 0000 0000 0001
 119A0   F9BC 0008 0000 0000   0001 F9C4 0008 0000   0000 0001 F9CC 0008   0000 0000 2020 4B45
 119C0   3030 3230 2020 4B45   3032 3030 2020 4B45   3032 3230 2020 5052   4F47 5220 4142 4F52
 119E0   5445 4420 4154 2020   5858 5858 2020 2050   5357 3D2F 2020 2020   2C41 3120 3D2F 2020
 11A00   2020 2C41 3220 3D2F   2020 2020 2C41 3320   3D2F 2020 2020 202C   4134 203D 2F20 2020
 11A20   202C 4135 203D 2F20   2020 2020 2020 2041   3620 3D2F 2020 2020   2C41 3720 3D2F 2020
 11A40   2020 2C41 3820 3D2F   2020 2020 2C41 3920   3D2F 2020 2020 202C   4131 303D 2F20 2020
 11A60   202C 4131 313D 2F20   2020 2020 2020 2041   3132 3D2F 2020 2020   2C41 3133 3D2F 2020
 11A80   2020 2C41 3134 3D2F   2020 2020 2C46 5231   3D2F 2020 2020 2C46   5232 3D2F 2020 2020
 11AA0   2C46 5233 3D2F 2020   2020 2020 4558 4954   2043 4F44 4520 3D20   5858 5858 F9E8 F9F4
 11AC0   F9FE FA08 FA12 FA1C   FA26 FA34 FA3E FA48   FA52 FA5C FA66 FA74   FA7E FA88 FA92 FA9C
 11AE0   FAA6 FAAA 0012 F9EC   0040 FA2C 0040 FA6C   003E FB2E 0010 FB3E   0018 FB56 0010 FB66
 11B00   0014 FB7A 0018 FB92   0012 FBA4 0018 FBBC   000E FBCA 0010 FBDA   0012 FBEC 0012 FBFE
 11B20   000E FC0C 000E FC1A   0016 FC30 0000 2020   504F 5745 5220 4641   494C 5552 4520 2020
 11B40   4E4F 5420 5749 5245   4420 494E 5354 5255   4354 494F 4E20 2020   4D45 4D4F 5259 2050
 11B60   524F 5445 4354 2020   4459 4E20 4152 4541   2044 4553 5452 4F59   4544 2020 544F 4F20
 11B80   4D41 4E59 2053 4348   4544 2E20 4C41 422E   2020 2020 4F50 4552   4154 4F52 2041 424F
 11BA0   5254 4544 2020 424C   4F43 4B2E 2042 5546   2E20 4F56 4552 464C   4F57 2020 2020 4441
 11BC0   4420 4F56 4552 464C   4F57 2020 4441 4420   5120 4F56 4552 464C   4F57 2020 4D45 4D4F
```

Chapter 3

INITIALISING THE SYSTEM

Declaring a Foreground Machine

Using the DCF Command

The general form of a DCF command is:

```
DCF_<machine ID>, n
```

where n is the number of segments to be reserved for this foreground machine.

In executing this command, the system creates in the dynamic area:

1 MCT for the new machine
1 PCT for the FCL task (linked to the system machine PCTs)
1 MMU save area (SEGMMU) in the core-resident segment table.
One of these is created for each segment, and therefore for n segments, 17(n+1)+1 words will be reserved.

The memory cost of this command is, therefore:

1 MCT	30 words
1 PCT	92 words
SEGO	18 words

and for each further segment, 17 words.

The general form is:

```
CMA_<No. of Pages>|,Public Lib.Size|
```

The system initialises:

1 MMU save area (SEGMMU)
a communication area.

Assuming, for example, that the command:

```
CMA 2,2000
```

had been given, the system would create a communication area having the following structure:

```
          ---------
(MCT)--->|    0    |
         |---------|
         | M/C ID  |
         |         |           ---
         |---------|            ↑       ----------       low address
/E       |  @CMA   |------------|------>| @  A    |---                 ↑
         | virtual |            |       |---------|   |                |
          ---------             |     A | @  B    |<--                 |
             MCT                |       |         |---                 |
                             dynamic    |---------|   |                |
                               area     |         |   |                |
                                |       |---------|   |                |
                                |     B |    0    |<--                 |
                                *       |---------|                    |
                                |     C |    0    |<--   ↑             |
                                |       |---------|   |  |        2 pages
                             public     |         |   |  |             |
                             lib.area   |         |   |  |             |
                                |       |         |   |  2000          |
                                |       |---------|   |  chars.        |
                                |     D | @  C    |---   |             |
                                |       |---------|      |             |
                                |       |@D(/FFFC)|      ↓             |
                                ↓        ----------                    ↓
                               ---         CMA           high address
```

Note: - @ means 'address of'

- A is the second word of the dynamic area and contains a pointer to B, the last free word in the dynamic area.

- D contains a pointer to C, the last free word in the public library area (this area is allocated from the top down).

Note: CMA 0 is a useless command, because one page is always reserved for the CMA (segment 0).

Using the SEG Command (Define a Segment)

General form:

```
SEG_n,<No. of Pages>
```

where - n is the segment number being defined.

Note: This command is only given if the machine has had core-resident segments (apart from segment 0) declared.

The initialisation of SEGMMU is performed for each segment.

Using the LAB, MFC and MBF Commands

The general form of these commands is as follows:

```
LAB_<No. of Scheduled Labels>
MFC_<Max. No. of Filecode Table Entries>
MBF_<Max. No. of Blocking Buffers>
```

The relevant fields of the MCT are updated.

Using the FCD Command

There are three forms of this command:

a) FCD <disc filecode>
b) FCD <device name><device address>
c) FCD <DAD filecode>,<disc filecode>,<DADname>

a) is used to define a physical disc file code
b) is used to define filecodes for non-disc devices
c) is used to define filecodes for DADs

One FCD command is given for each filecode for this foreground machine which will not be assigned using the ASG command or LKM 23. All discs, however, must be declared with FCD commands.

For command types a) and b), the system creates one FCT for each device, the DWT having been created at SYSGEN. The cost is five words for each device.

For command type c), the system:

- creates one FCT per DAD,
- reads the VTOC of the disc,
- creates a DAD control table (T:DAD) for each DAD,
- reads the BITAB in order to initialise T:DAD and creates the DADBTB (which defines the granules allocated to the DAD).

The cost in memory is:

8 words per FCT
14 words for each T:DAD entry
$\frac{G}{16}$ + 1 words for the DADBTB
where G is the number of granules in the DAD.

Note: The buffer required for reading the VTOC is freed when this operation is complete.

The T:DAD entries for SUPERV and D:CI, and D:MSEG if required, are created at SYSGEN.

Other SCL Commands

Using the NDV Command (Introduce a new device)

The format of this command is:

```
NDV_<dev.name>,<dev.address>,<interrupt level>
```

The system creates a 45 word entry in the system machine communication area containing:

- a DWT for the device, and
- the interrupt routine.

The address of the interrupt routine is then entered in the level location table LOCAT.

Using the POF Command (Page Off)

General format:

```
POF_<Page No.>
```

A page table is created in the dynamic area, of the same form as T:CORE and consisting of 6 words. Initially zero, each bit in the table is considered to represent a page number, these being set in a 1:1 correspondence in ascending order. A bit set to one indicates that the corresponding page is temporarily 'OFF' and cannot be allocated.

The page table is addressed from the last location in T:CORE.

The effect can be reversed using the 'PON' command.

Using the DOF Command (Device Off)

This command sets bit 5 in location DWTDA (/2) of the DWT.

The effect can be reversed using the 'DON command.

Using the FOF Command

If hardware floating point is provided, INIMON sets bits 0 and 11 in location CVTSSI (/12) of the CVT. On receipt of an 'FOF' command, the system turns off bit 0, which has the effect of preventing the despatcher from saving the floating point registers.

The effect can be reversed by means of the 'FON' command.

Initialisation carried out by the System while processing FCL commands

Using the LOD Command (Load a Memory-Resident Program)

The general format is:

```
LOD_<Segment No.>,<Programm ID>,<DAD filecode>[,max No.of Sch.Labs.]
```

The system creates a PCT in the dynamic buffer area (74 words) and loads the program from the specified DAD into the pages previously reserved for it.

Using the REP Command (Load a Re-Entrant Program)

The general format is:

```
REP_<max No.of activations>,<Seg.No.>,<Prog.ID>,<DAD filecode>[,
            <max No. of sched. labels>]
```

The No. of activations is entered into the PCT which, of course, must be created in the dynamic area of the machine. For each subsequent activation a new PCT is created, and thus the maximum cost in words of memory is:

74 x Max No. of activations

Using the SWP and RON Commands
(Declare A-Swappable or a Read-only Program)

General formats:

```
SWP    <program name>,<DAD filecode>[,<max no.sched.labels>][,<min.
       memory-resident time>]
RON    <program name>,<DAD filecode>[,<max no.sched.labels>][,<min.
       memory-resident time>][,R,max no.of activations>]
```

The system creates a PCT (with PCTMMU of 92 words) and initialises the locations PCTSWI (/2E) and PCTSWN (/30) with the addresses in the DAD D:CI of the initial program core image and swap area. The load module is converted to a core image program and copied to the granules which have been allocated in D:CI. In the case of the SWP command, additional granules are reserved in D:CI for the swap area.

Using the CNL command (Connect a Program to a Level)

General format:

```
CNL_<program name>,<level>
```

The system updates SLT with the PCT address, and the PCT with the level number.

Using the ACT Command (Activate a Program)

The general format is:

```
ACT_<program name>[,<A3 value>][,<A4 value>]
```

The program is started at its specified start address but, in the case of swappable or read-only programs, these are first loaded from DAD D:CI.

Notes:

1) This must be taken as a minimum figure; other buffers must be reserved for handling non-disc devices. These buffers are allocated at the start of an I/0 and freed when it ends, so several buffers may be reserved at one time, for queued I/0 requests on one device. The user should leave one extra page in the System Dynamic Area for these requests.

2) If the command LAB 0 is entered, the scheduled label save area is not created, and this saves 21 words.

Using the SCR command (Remove a Filecode Entry)

General format:

```
SCR_<filecode>
```

The system frees the relevant FCT and blocking buffers.

Using the ASG Command (Assign a Filecode)

There are four basic types of ASG command:

a) Assign a filecode to a non-disc device
b) Assign a filecode to a catalogued or temporary disc file
c) Assign a filecode to another filecode
d) Assign a filecode to a DAD.

In the case of:

a) The system creates a new FCT of 5 words.
b) The system creates a new FCT of 5 words and an LFT of 35 words.
c) The system creates a new 6-word FCT for the old filecode (the old 5-word FCT is freed) and a new 5-word FCT for the new filecode.
d) See the assignment of a DAD via an FCD command.

Using the LSM Command (Load a Secondary Load Module)

The general format is:

```
LSM_<module name>,<DAD fc>,<userid>[,R|W]
```

The system creates a table in the dynamic area of the system machine consisting of five words + a number of words equal to the number of pages occupied by the module (these are, in fact, a copy of the MMU registers).

<u>Example</u>: Calculating the memory size of the system tables created during the declaration of a machine.

```
DCF TEST, 2                         1 MCT + 1 PCT + SEGMMU      174 words
CMA 2, 4096
SEG 1, 4
SEG 2, 3
FCD 01, TY10
FCD 02, LP07
FCD 03, PR20
FCD 04, CR06                        10 FCT (normal)              50 words
FCD 05, PP30
FCD /C0
FCD /C2
FCD /C3
FCD /E0 TY10
FCD /E1 CR06
FCD /F0, /C0, SUPERV   mandatory, to declare filecode /F0
FCD /F2, /C3, LKM2
FCD /F8, /C2, OBJECT
FCD /F9, /C2, UPDATE
FCD /FA, /C2, ECRIDA                5 FCT (for DAD)              40 words
DEN                                 4 T:DAD                      56 words
                                    4 DADBTB (alloc.table)       54 words
                                                                ---------
                                                                374 words
```

Remarks

- In this example, the sum of granules for all the declared DADs is 800 granules (50 cyl. per DAD).

- All the lengths given include the chain word of the buffer dynamic area.

- After a DCF command, the system returns 'D:' until the 'DEN' command is entered, when it returns FCL again.

Chapter 4

THE SYSTEM DYNAMIC AREA

The length of the System Dynamic Area is given in the CVT, location /14 (CVTSS2). It is mainly used by the system to create all control blocks required to perform the tasks requested by users. The control blocks are:

- File Code Table (FCT) for all machines declared
- Program Control Table (PCT) for all programs declared
- Logical Disc File Table (LFT) for all user files assigned
- DAD Control Table (T:DAD) for all user DADs assigned.
- Machine Control Table (MCT) for all machines declared
- Queues, mail boxes, semaphores, secondary load module tables, data window tables, short time tables.

In addition, the system reserves blocking buffers in the Dynamic Area in the following cases:

- for processing sequential files
- for processing direct access non-consecutive files (for the GRANTB)
- for processing direct access files on disc, when the user buffer spreads out over two non-consecutive pages.

The length of such a buffer is equal to the sector length of the DAD containing the file. For example, if the system must handle N files simultaneously, the cost in memory is equal to the sum of the sector lengths of the DADs concerned, plus N chaining words. See the sections on initialising the system with SCL and FCL commands, for the memory cost of the tables created.

Chapter 5

OBJECT LIBRARY STRUCTURE

An object library is composed of 4 parts:

- One sector containing the GRANTB
- Others containing the Data (object modules)
- Others containing the Directory
- Another containing the Directory Header.

GRANTB Sector

GRANTB is located on library sector 1 and indicates the addresses of the granules used for this library.

A listing of this sector can be obtained in the following way:

```
LIB
PRD USID = QUALIF, DAD = /F4
```

The listing thus obtained gives, for each file catalogued under this user name, the following information:

```
FILE NAME   TYPE  VERS  DATE    ADDR  ORG
SYSLIB      OB          020177  07C8  NC
```

In this example, /07C8 is the address of the first sector of SYSLIB library; the operator may then send the following command:

```
DUF  FNAM = /F4,  FROM = /7C9,  TO = /7C9
```

OBJECT LIBRARY STRUCTURE

Remarks

FNAM is followed by the filecode of the DAD containing the library. This method for obtaining the GRANTB listing is valid for all types of files.

```
                 |Length of GRANTB
                 |
                 |             |Address of data granules
SECTOR / RECORD                |
                 ↓             ↓
0000   003E 0190 07C8 0978 0988 0990 0998 09A0 09A8 09B8 09C0 09C8 09D0 09D8
0020   09E8 09F0 0A00 CA08 0A20 0A18 0A20 0000                          0000
0180   0000                     0000 09F8 09B0 0970 0000
                                                ↑
                                                |
                              Address of directory granules
```

Sector Containing Data

All object modules are separated by "end of segment" records.

A listing of these sectors can be obtained by the DUF command:

```
Example: list the first 50 data sectors of SYSLIB library
         LIB
         DUF FNAM = SYSLIB, TYPE=OB, USID = QUALIF;
         DAD = /F4, FROM = 0, TO = 50
```

The Sector Containing the Directory Header

The logic address of this sector is 1597; the description of this sector is as follows:

```
                          |Creation Date
                          |
                          |       |No. of sectors in Directory
Sector / Record 063D   LENGTH 019A |
                          ↓        ↓
0000     0036 000C 3032 3031 3734 0014 00A1 00A0 0000 ------ 0000
                                         ↓    ↓
                                         |    |
                                Initial No.   |
                                of Sectors    |
                                           Actual No.
                                           of Sectors
```

The sum of the number of sectors occupied by the Directory and the number of sectors occupied by the Data indicates the fill-up rate of this library.

This sum must be less than 1596; if this limit is reached, insertion of new modules will be rejected (library overflow).

A listing of this sector can be obtained by the DUF command; for example:

```
LIB
DUF  FNAM=SYSLIB, TYPE=OB, USID=QUALIF, DAD=/F4
FROM=1597, TO=1597
```

Sectors Containing the Directory

The Directory is created on the sectors preceding the "Header Directory" sector.

While insertions are being carried out, sectors 1596 then 1595 etc., will be allocated to this Directory.

The number of assigned sectors is indicated in the Directory Header (see previous paragraph).

The list of these sectors can be obtained by the DUF command.

Example: List the SYSLIB Library Directory.
No. of sectors in Directory=/14 i.e. 20 sectors the address of the last sector of the Directory is 1596 - 20 = 1576.

a) The following DUF command may then be sent:

```
LIB
DUF FNAM=SYSLIB, TYPE=OB, USID=QUALIF,
DAD=/F4, FROM=1597, TO=1597
```

b) The POD command may be used:

```
POD_[PRNT=<fc>][,EXTN=YES|NO][,LIBR=<filename>]
       [,USID=<userid>][,DAD=<dad fc>]
```

For example:

```
POD LIBR=SYSLIB,USID=QUALIF,DAD=/F4
```
where the default values are /02 for the print filecode and 'NO' for EXTN.

Chapter 6

BLOCK DIRECTORY STRUCTURE

Numbers refer to the example on the next page

1 - No. of characters in the block, this word being excluded
2 - module name
3 - relative sector number of the 1st sector of the module
4 - initial No. of sectors in the module
5 - actual No. of sectors of the module
6 - date (last catalogue day)
7 - two words not used
8 - clusters type 5, 2, 6 (object code records)
9 - cluster end type 7.

Information in a cluster:

10 - type of cluster
11 - length of this cluster in words
12 - bit 0-2 shows the length in characters of the name
13 - name of the external reference (if type = 2)
14 - 8 if relocatable (for entry point)
15 - name of the entry point (if type = 5)
16 - logical address of the entry point in the object module
17 - start address if main routine
18 - length of object module in characters
19 - number of errors found during compilation by ASM

Chapter 7

LOAD MODULE GENERAL STRUCTURE

On disc, the load module has the following structure (the numbers refer to the diagram on the following page):

1 - Start address of the program (bit 15 is set for an overlaid program).
2 - Number of sectors to be loaded when starting the program (for an overlaid program it is the number of sectors of the root).
3 - Effective length of the program in bytes (including these 4 words).
4 - Length of the program region, i.e. the space required for loading and execution.

The following words exist only for an overlaid program:

5.1 - The numer of segments (root excepted) in the program.
5 2 - Load segment block; one of these exists for each segment and is used for segment loading by the overlay control routine. They are pointed to by A8 before the LKM 27 Command is issued.

The format of an overlay control block is as follows:

event byte	ascendant seg. number
load address	
disc address	
effective length	

- Bit 0 of event byte is set when the corresponding segment is loaded in core, which means that it is very easy to find out, from a dump of the program after an abort, which segments were in core.

6 - Behind this segment table we find the overlay control routine, which requests by LKM 27 the loading of one segment before it exits.

```
              START = 019E      LENGTH = 046C      REGION = 053C
                     ***     OVERLAY STRUCTURE      ***

                        ***       LEVEL       0    ***

   SEGMENT      00     ADDRESS = 00DC     SECTOR     0000     ASCENDANT     FF
OLEIDA    00DC      DEBI       01A2    R:EXAS   01FC

                        ***       LEVEL       1    ***

   SEGMENT      01     ADDRESS = 02C6     SECTOR     0002     ASCENDANT     00
ECR000    02C6

   SEGMENT      02     ADDRESS = 02C6     SECTOR     0003     ASCENDANT     00
ECR005    02C6

   SEGMENT      06     ADDRESS = 02C6     SECTOR     0007     ASCENDANT     00
ECR020    02C6

   SEGMENT      09     ADDRESS = 02C6     SECTOR     000A     ASCENDANT     00
ECR180    02C6

                        ***       LEVEL       2    ***

   SEGMENT      03     ADDRESS = 03A6     SECTOR     0004     ASCENDANT     02
BIDE      03A6

   SEGMENT      04     ADDRESS = 03A6     SECTOR     0005     ASCENDANT     02
MUST      03A6

   SEGMENT      05     ADDRESS = 03A6     SECTOR     0006     ASCENDANT     02
SUBLED    03A6

   SEGMENT      07     ADDRESS = 0376     SECTOR     0008     ASCENDANT     06
BIDE      0376

   SEGMENT      08     ADDRESS = 0376     SECTOR     0009     ASCENDANT     06
MUST      0376
```

LOAD MODULE GENERAL STRUCTURE

When loaded into core, the start of the program area has the following structure:

1) For all types of program -

word	contents
0	Start address (bit 15 is set for overlaid programs)
1	Number of sectors loaded (for overlaid programs this is the sector length of the root)
2	Program length in bytes (length of root for overlaid programs)
3	Load address (i.e. address of word 0).

2) Overlaid programs have, in addition, the following:

word	contents
4	Length of the program region (space required for loading and execution)
5	Number of segments (root excepted)
6-9	Overlay control block; one of these exists for each overlay segment in the program. The format of these blocks is as follows:

```
 ------------------------------------------------
| Event byte      |     Ascendant Seg. number    |
|------------------------------------------------|
|                 Load address                   |
|------------------------------------------------|
|                 Disc address                   |
|------------------------------------------------|
|                 Effective length               |
 ------------------------------------------------
```

Bit 0 of the Event Byte is set to 1 when the segment is loaded.

Load Module General Structure:

This is a random file, each sector of which contains 188 code words and a 12 word relocation table.

In this table, each bit corresponds to one code word, the bit is set to one if a relocation must be made during loading. The first sectors of the load module contain the root. If the user wants to know where a particular segment begins on the file, he just has to look up the value of the "disc address" in the corresponding control block in the segment table.

Example: Segment 4 begins at the disc address 5.

Chapter 8

DATA COMMUNICATION INTERNAL STRUCTURE

Control Blocks for DATEM

LCB - Line Control Block Contains the characteristics of the line and Datem request parameters .

T:SCT - Special Characters Table
Contains a set of editing characters and a set of terminator characters.

DTCTIM - Timers Table
contains the values of the timers initialised by the DATEM requests with time control.

SYNTAB - SYN table
This table is used for synchronous lines; contains up to 8 SYN values.

LCT - Line code table (one per machine)
Contains the LCB address for each line code.

ALT - Additional line code table (one per line)
This is an extension of the LCT and has the same function.

ECB - Event Control Block
Contains the parameters for DATEM requests.

DATA COMMUNICATION TABLE DESCRIPTIONS

LCB Line Control Block

```
Location
  |       0   1   2   3   4   5   6   7  8   9   10  11  12  13  14  15
  V      ---------------------------------------------------------------
 -2     |             NEXT LCB ADDRESS 0 IF LAST                         |
        |----------------------------------------------------------------|
  0     |BSY| ASG  |CRN|FLD|  DVT  |IOP|WRD|ECH|CAR|TCA|CTT|GVM|WFD|CNT|
        |----------------------------------------------------------------|
  2     |    SYN   |WFE|  INDEX       |ALC|WFC|         DVA              |
        |----------------------------------------------------------------|
  4     |                  TIME CONTROL VALUE                            |
        |----------------------------------------------------------------|
  6     |SCB| CRC  |HFR|  PAR  |INH|INP|ASY|  CRC  |HFR|  PAR  |  OUT   |
        |----------------------------------------------------------------|
  8     |                     ECB ADDRESS                                |
        |----------------------------------------------------------------|
 /A     |               SCHEDULED LABEL ADDRESS                          |
        |----------------------------------------------------------------|
 /C     |         SYSTEM BUFFER POINTER TO NEXT CHARACTER                |
        |----------------------------------------------------------------|
 /E     |ONE|              REMAINING LENGTH                              |
        |----------------------------------------------------------------|
/10     |                  REQUESTED LENGTH                              |
        |----------------------------------------------------------------|
/12     |                     PCT ADDRESS                                |
        |----------------------------------------------------------------|
/14     |EBC|  LNB      |     NBC       |       INTERRUPT LEVEL          |
        |----------------------------------------------------------------|
/16     |                  SYSTEM BUFFER ORIGIN                          |
        |----------------------------------------------------------------|
/18     |                    USER BUFFER ORIGIN                          |
        |----------------------------------------------------------------|
/1A     |           CHECK TABLE ADDRESS / ENQ CHARACTER                  |
         ----------------------------------------------------------------
```

A more detailed description of these fields follows:

Location 0

BSY - 1 The line is busy
0 If not busy

CRN - 1 The carrier is always on
0 If not always on

FLD - 1 Full duplex connection
0 If not full duplex

DVT - Device Type
00 SLCU2
01 SLCU4
10 ALCU4
11 AMA8A/C

IOP - 1 The control unit is connected to the IOP
0 If not so connected

WRD - 1 A write process has been started
0 If not

ECH - 1 A "Read echo" has been started
0 If not started

CAR - 1 Opposite carrier on (set if a read process has been started)
0 If not

TCA - 1 Time control pending
0 If not

CTT - 1 A terminator table is to be check
0 If not

GVM - 1 One or two extra BCC characters must be read after a terminator character. Set in all synchronous read request and if a negative request number is specified for an asynchronous read
0 If not

WFD - 1 A "Wait for Data" request has been started
0 If not

CNT - 1 The modem is connected to the line. Always set for a leased line
0 If not (switched line disconnected)

Location 2

SYN - SYNTAB index which points to a SYN value. Used only with synchronous lines

WFE - 1 A "Search pattern" request has been issued
0 If not

ALC	-	1	The line is always connected (leased line)
		0	If switched line
WFC	-	1	A "Wait for Call" request has been started
		0	If not or if a "calling interrupt" is received
DVA	-		Device address of the control unit handling the line
INDEX	-		Index entry in special characters table

Location 6

This word contains the information sent by a CIO Start. The first character is used in input, the second in output.

SCB	-	1	Synchronisation mode (only for SLCU2)
		0	If not
CRC	-	00	No CRC
		01	CRC CCITT
		11	CRC IBM
		10	LRC
HRC	-		Frequency selection of modem
		1	Higher frequency
		0	Lower frequency
PAR	-	00	No parity
		01	Even parity
		11	Odd parity
INH	-	1	Inhibit control character check (SLCU2)
		0	If not
INP	-	1	For input
ASY	-	1	Auto SYN generation (SLCU2)
		0	If not
OUT	-	0	For output

Location /E

ONE	-	1	One character at least has been transmitteed

Location /14

EBC	-	1	The EBCDIC code is used (SLCU2)
		0	If not
LNB	-		Line number for AMA8
NBC	-		Number of bits per character

Location /1A

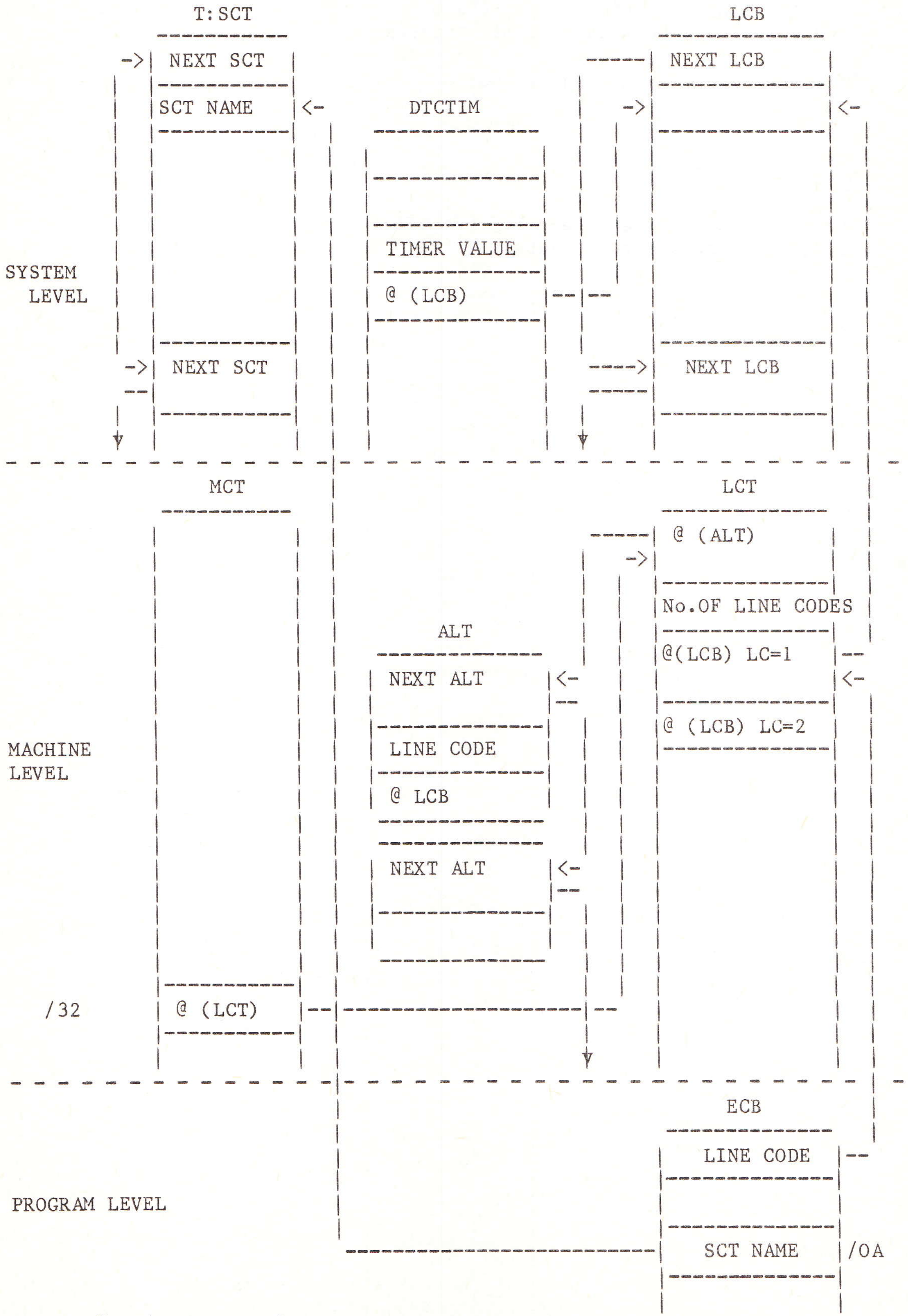
During a read operation and if the bit 12 (CTT) of word 0 is set, this word contains a terminator table address.

If a "search pattern" request has been started, this word contains the ENQ pattern to be checked.

Note

For full duplex lines there are two LCBs: the first is for the input line, the second for the output line.

Chaining of Datem Blocks



Note: @ = 'address of'

T:SCT Special Characters Table

Location	
0	NEXT SCT ADDRESS
2	NAME OF THE TABLE (2ASCII CHAR)
4	TERMINATOR TABLE LENGTH
6	EDITING TABLE LENGTH
8	EDITING TABLE
	TERMINATOR TABLE

Editing Table

The function of this table is to define the action to perform after the detection of a special character.

Each word of this table is divided as follows:

0 7	8 15
ACTION	character

ACTION
00 Delete the previous character and this one
02 Delete all previous characters and this one
04 Ignore this character if others are following
06 Ignore this character always

Terminator Table

This table is used to check the end of an input. Each word has the following layout:

- Right byte Terminator character
- Left byte If zero, the terminator consists of only one character (found in right byte).
  Else the terminator consists of two characters, the first one checked on line being provided from left byte, the second one from right byte.

DTCTIM The Time Control Table

```
             --------------------------------------------------
Location 0  |          NUMBER OF PENDING TIME-OUTS              |
            |---------------------------------------------------|
         2  |          MAXIMUM NUMBER OF TIME-OUTS (n)          |
            |---------------------------------------------------|
         4  |                TIME-OUT VALUE                     |)
            |---------------------------------------------------|)1 ENTRY
         6  |          LCB ADDRESS FOR THIS TIME-OUT            |)
            |---------------------------------------------------|
         8  |        ↑                                          |
            |        |                                          |
            |        |                                          |
            |        |                                          |
            |        |     n-1 ENTRIES                          |
            |        |                                          |
            |        |                                          |
            |        ↓                                          |
             --------------------------------------------------
```

SYNTAB

```
 -------------------------------------------------------------
|    FIRST "SYN" VALUE         |      SECOND "SYN" VALUE      |
|------------------------------|------------------------------|
|                              |                              |
|------------------------------|------------------------------|
|                              |                              |
|------------------------------|------------------------------|
|                              |      EIGHTH "SYN" VALUE      |
 -------------------------------------------------------------
```

A SYNTAB index from 0 to 7 is given by the word 1 (0-2) of the LCB.

Before a write operation on a synchronous line, the first word of the Buffer will be filled with the SYN pattern found in the SYNTAB.
The buffer address is held in location 2 of the ECB.

LCT The Line Code Table

```
           bit 0              7   8                         15
               -----------------------------------------------
Location 0    |           ADDRESS OF FIRST ALT                |
              |-----------------------------------------------|
         2    |  0  |           |  NUMBER OF ENTRIES (n)      |
              |-----------------------------------------------|
         4    |  LCB ADDRESS FOR LINE CODE No.1 (0 IF NOT ASSIGNED) |
              |-----------------------------------------------|
         6    |  LCB ADDRESS FOR LINE CODE No.2 (0 IF NOT ASSIGNED) |
              |-----------------------------------------------|
              |                                               |
              |                                               |
              |                                               |
              |                                               |
              |-----------------------------------------------|
Loc'n 2n+2    |  LCB ADDRESS FOR LINE CODE = n                |
               -----------------------------------------------
```

ALT The Additional Line Code Table

```
           bit 0              7   8                         15
               -----------------------------------------------
Location 0    |           ADDRESS OF NEXT ALT                 |
              |-----------------------------------------------|
         2    |  1  |           |         LINE CODE           |
              |-----------------------------------------------|
         4    |       LCB ADDRESS FOR THIS LINE CODE          |
               -----------------------------------------------
```

ECB The Event Control Block

```
               0              7   8                         15
               -----------------------------------------------
Location 0    |  E  |           |               LINE CODE     |
              |-----------------------------------------------|
         2    |           BUFFER ADDRESS                      |
              |-----------------------------------------------|
         4    |           BUFFER LENGTH                       |
              |-----------------------------------------------|
         6    |          TRANSMITTED LENGTH                   |
              |-----------------------------------------------|
         8    |           SERVICE STATUS                      |
              |-----------------------------------------------|
        /A    |     SPECIAL CHARACTER TABLE NAME, IN ASCII    |
              |-----------------------------------------------|
        /C    |     TIME-OUT VALUE IN TENTHS OF SECS.         |
               -----------------------------------------------
```

Datem Request Calling Sequence

Any service must be requested using the following sequence:

```
      LDK    A7,    service number
      LDKL   A8,    ECB address
      LKM
      DATA   8      (-8 if scheduled label used)
      DATA          scheduled label address
```

The following services are provided by DATEM:

(A7)	1 (A) +		Read with echo
	2		Read without echo
	3 (A) +		Read with echo with time control
	4		Read without echo with time control
	5		Write
	6		Write with time control
	/D		Change line definition
	/E		Get line definition
	/10		Stop the exchange
	/11	*	Disconnect the line
	/12 (S)		Search pattern
	/13	*	Wait for a call
	/14(A)-		Accept data
	/15		Set time control

(A)	Asynchronous lines only
(S)	Synchronous lines only
*	Switched lines only
+	Full duplex lines only
-	Not possible with AMA8 with input under IOP

Results of Datem Commands

SLC cmmands

DLN command: DLN_[No. of line codes]

- Reserves the necessary field for the LCT and initialises the second word of it.
- Initialises the word MCTLCT (MCT + /32) which is the link between the MCT and the associated LCT.

The area required for the LCT is reserved in the system dynamic area.
If n line codes are declared, the cost in memory will be n+2 words.

DLC command: DLC_<Line code>,<dev name>[<dev address>]

Either:

- The line code declared is not greater than the line number declared by DLN command.
  In this case, the line code is used as a pointer in the LCT. The LCB address corresponding to the line given by the device description is then stored in the LCT indexed by the line code + 2. No field is reserved.

- The line code declared is greater than the line number declared by DLN command.
  In this case, the DLC command causes the creation of an Additional Line Code Table (ALT). The link with the LCT or the previous ALT is then initialised and the LCB address is stored in the ALT location 4.

FCL commands

DAS command: DAS_<Line code>,<dev name>[<dev address>]
The processing of this command is the same as for the DLC command.
DDL command: DDL_<Line code>
The LCT entry or the ALT corresponding to this line code is deleted.

Datem Status Codes (ECB location /8)

A one word status reply is given after the DATEM request has been completed. No bit set: no error detected.

Bit	0 set	Logical line busy
	1	Non-connected line
	2	Illegal line code
	3	Illegal request
	4	Character(s) lost
	5	End of carrier detection
	6	Time-out request may not be served
	7	Buffer overflow
	8	Transmission stopped
	9	Power failure
	10	Time over
	11	Break detection (Asyn only)
	12	Command refused
	13	Parity error (hardware detection)
	14	Throughput error
	15	Modem not operable.

Chapter 9

IPL PROCEDURE

Organisation

The option consists of a 64-word ROM mounted on the CPU card, holding a bootstrap program, and the necessary control circuits to load and run the bootstrap using parameters previously set onto the 16 data switches.

The parameters set on the data switches are:

bit 0 = 0: character exchange on Programmed Channel
      = 1: word exchange on Programmed Channel

bit 1 = 0: IPL not loaded from disc or loaded from CDC disc
      or loaded from CMD
      = 1: IPL loaded from disc

bit 2 is used only if bit 1 = 1:
      = 0 fixed head disc, flexible disc or CDC disc (BIGD)
      = 1 moving head disc (X1215 or X1216) or CDC disc (BIGD2)
      or CMD

bit 3 = 0: IPL input device connected to I/O Processor
      = 1: IPL input device connected to Programmed Channel

bits 4 to 7 contain control information for the control unit:
TY = 0001 MT = 0010 FL = 0000 FHD = 0000
TK = 0111 PR = 0000 CDC = 0001 X1215/16 = 0011
DFPC fixed = 0001 DFPC Removable = 0000

bit 8 = 0: a single device control unit is involved
      = 1: a multiple device control unit is involved

```
bit 9 = 1:  X1215 or X1216 used for IPL
      = 0:  other device or disc type used

bits 10 to 15 contain the device address (see new Appendix on
            device addresses contained in this supplement).
```

Remarks:

Bits 4 to 7 are given above as 0011 for X1215/16, but since this is the interlace number of the first DAD, 0101 could be used.

Where a device has not specific setting requirements on the data switches, it is sufficient to set the switches to define "Other devices", the correct channel, and the device address and qualification required for the CIO start command.

Operation

The operation of the initial program loader consists of 4 main steps:

1. The bootstrap is copied from the ROM into the first words of the central memory.

2. The contents of the 16 data switches are copied into register A15.

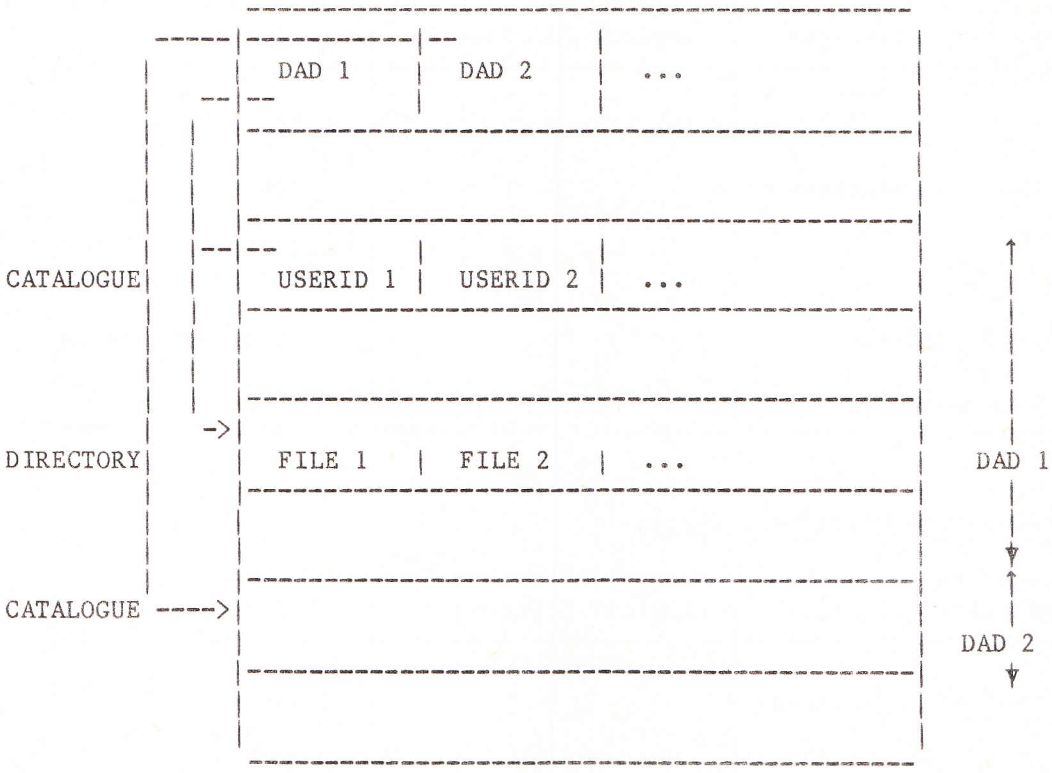
3. The CPU is put into the INHIBIT INTERRUPT state.

4. The P register is loaded with zero and the CPU started in run mode.

Chapter 10

DISC ORGANISATION

The logical structure of a disc, from a user's point of view, can be shown diagrammatically as follows:



Logical Disc Structure

Logical Disc Organisation

The first DAD of the disc begins at the cylinder 0 (zero), where the first granule (at least 8 sectors) is used to describe the disc organisation characteristics as follow:

- VOLAB, volume label, is located at:
  - * sector zero for X1215/6 discs.
  - * sector i (where i is the number of interlaces of the first DAD) for CDC disc and Fixed Head disc.
- IPL (System disc only), initial program loader, used to load the supervisor, is located at:
  - * sector i (i = number of interlaces in the first DAD of the disc) for X1215 disc.
  - * sector zero for CDC disc and Fixed Head disc.

The logical sector addresses (logical = relative address in the first DAD) of these sectors are:

X1215/6 discs

Physical Addresses	Logical Addresses	Contents
0	0	VOLAB
i (No. of interlaces)	1	IPL
	2 to 5	Catalogue of the first DAD
6 i (modulo 16)	6	List of bad tracks
7 i (modulo 16)	7	VTOC

CDC Discs and Fixed Head Discs

Physical Addresses	Logical Addresses	Contents
i (No. of interlaces)	0	VOLAB
0	1	IPL
	2 to 5	Catalogue of the first DAD
	6	List of bad tracks
	7	VTOC

CMD disks

The CMD (Cartridge Module Drive) disks are physically structured as a set of sectors with a fixed length (256 data octads). These sectors can all be accessed physically by giving the Real Sector Number (RSN), but at DAD level a disk is organised as a set of logical sectors. These logical sectors, the length being defined at Premark or Declare DAD time, are mapped on 1 or several consecutive logical sectors. Access to these logical sectors is done at DAD level by giving the logical sector number within the DAD. The physical coordinates are calculated as follows:

RSN=(log.sectnr x N) + RSN(begin DAD)
N= Number of physical sectors on which the logical sectors are mapped.

The organisation of the first DAD is:

sector 0	IPL
N	VOLAB+BITTAB
2N	CATALOG
3N	CATALOG
4N	CATALOG
5N	CATALOG
6N	Bad track list
7N	VTOC

Note: For CDC discs and Fixed Head discs, the LKM I/O request orders /01, /05, /11, /15 are not identical; they are slightly different when cylinder and head numbers are both 0 and sector no.is 0 or 1.

Order	Sector	Operation
/11	0	Read the physical sector = <number of interlaces> of the first DAD (sector which contains the VOLAB)
/11	1	Read the physical sector 0 of the first DAD (IPL)
/15	0	Write the physical sector = <number of interlaces> of the first DAD, i.e. the VOLAB sector.
/15	1	Write the physical sector 0 of the first DAD (IPL).
1	0	Read the physical sector zero
1	1	Read the physical sector 1
5	0	Write the physical sector zero
5	1	Write the physical sector 1

Thus, depending on the nature of the discs, the sector zero is used in different ways. However, in order to be able to initialise a disc pack when it is mounted, 5 words in sector zero must always be set as follows, whatever the disc model:

Decimal Byte Address	Hexadecimal Byte Address	Contents
74	/4A	Physical sector number of the bad track list (Logical address = 6)
76	/4C	No. of sectors per track in the first DAD
78	/4E	No. of interlaces of the first DAD
80	/50	Sector size (in characters) of the first DAD
82	/52	Physical sector number of the VTOC (logical addr = 7)

VOLAB format

The first 86 characters of the sector are used to contain:

- the label of the volume,
- Premark date,
- Volume number.
- Characteristics of the first DAD as follows:

/0	Sector id. used only for X1215, set to zero	
2	(not used)	
4	␣	␣
6	␣	␣
8	L	A
/A	B	E
/C	L	␣
/E	=	␣
/10	↑	
/12	16 characters of volume label	
/14		
/16		
/18		
/1A		
/1C		
/1E	↓	
/20	␣	␣
/22	D	A
/24	T	E
/26	␣	=
/28	␣	␣
/2A	Day in ASCII	
/2C	␣	␣
/2E	Month in ASCII	

```
     |--------------------------------------------------------|
/30  |  |__|     |__|                                         |
     |--------------------------------------------------------|
/32  |  Year in ASCII                                         |
     |--------------------------------------------------------|
/34  |  |__|     |__|                                         |
     |--------------------------------------------------------|
/36  |  |__|     |__|                                         |
     |--------------------------------------------------------|
/38  |  |__|     |__|                                         |
     |--------------------------------------------------------|
/3A  |  P          A                                          |
     |--------------------------------------------------------|
/3C  |  C          K                                          |
     |--------------------------------------------------------|
/3E  |  |__|       N                                          |
     |--------------------------------------------------------|
/40  |  B          R                                          |
     |--------------------------------------------------------|
/42  |  |__|       =                                          |
     |--------------------------------------------------------|
/44  |  |__|     |__|                                         |
     |--------------------------------------------------------|
/46  |      ↑    4 hexadecimal ASCII characters               |
     |------|-------------------------------------------------|
/48  |      ↓    Containing the Volume number                 |
     |--------------------------------------------------------|
/4A  |  Physical Sector Number of Bad Track List              |
     |--------------------------------------------------------|
/4C  |  Number of Sectors per Track                           |
     |--------------------------------------------------------|
/4E  |  Number of Interlaces of the First DAD                 |
     |--------------------------------------------------------|
/50  |  Sector Size in Characters of the First DAD            |
     |--------------------------------------------------------|
/52  |  Physical Sector Number of the VTOC                    |
      --------------------------------------------------------
```

Bad Track List

The logical sector 6 contains the addresses of bad tracks on the disc. These addresses are written at the disc premark or DAD premark. The defective track address is stored in a word as follows, for either X1215/6 or CDC disc:

```
 ------------------------------------------------------------
|    | Cyl No.                       |     Head No.          |
 ------------------------------------------------------------
 0   3                               11   12   13   14   15
```

At premark operation:

When a defective track is detected, its address is added into the list, and the length of the table is updated. Its relative position from the beginning of the table defines the relative position of its replacing track from the first spare cylinder of the disc. The format of defective track sector follows:

0	Sector ID (used only for X1215/6 disc)
2	Length in characters of defective track list, this word excluded
4	Address of first defective track
6	Address of second defective track
. . .	etc.

VTOC format

VTOC is used to contain the coordinates and characteristics of DADs. Entries are of 8 words. They are written in the order in which the DADs are defined on the disc.
Thus, when the address of the first cylinder of a DAD is not equal to the address of the last cylinder + 1 of the previous DAD, that means a certain number of free cylinders exist between these 2 DADs. They will be used at the next DAD allocation, if possible before acquiring cylinders from the first free cylinder pointer.

The format of the VTOC is as follows:

```
     ----------------------------------------------------------
 0  | Sector ID (used only for X1215/6)                        |
    |----------------------------------------------------------|
 2  | No. of used characters, this word and the last word      |
    | (FFFF) included                                          |
    |----------------------------------------------------------|
 4  |    V     T                                               |
    |----------------------------------------------------------|
 6  |    O     C                                               |
    |----------------------------------------------------------|
 8  |   |__|  |__|                                             |
    |----------------------------------------------------------|
/A  |        0                                                 |
    |----------------------------------------------------------|
/C  | No. of tracks per cylinder                               |
    |----------------------------------------------------------|
/E  | No. of cylinders of the disc                             |
    |----------------------------------------------------------|
/10 | Address of the first free cylinder after the last DAD    |
    |----------------------------------------------------------|
/12 | Address of replacing cylinder (first spare)              |
    |----------------------------------------------------------|
/14 |    DAD name ↑                                            |
    |--------------|-------------------------------------------|
/16 |              |                                           |
    |--------------|-------------------------------------------|
/18 |              ↓                                           |
    |----------------------------------------------------------|
/1A | No. of interlaces     |    No. of sectors per track      |
    |----------------------------------------------------------|
/1C | Sector size (in characters)                              |
    |----------------------------------------------------------|
/1E | No. of cylinders per DAD                                 |
    |----------------------------------------------------------|
/20 | Address of the first cylinder of the DAD                 |
    |----------------------------------------------------------|
/22 | No. of sectors per granule                               |
     ----------------------------------------------------------
```

The format of the 'First Entry' (dummy) is as follows:

Location /A Set to zero, not used yet.
Location /C No. of tracks per cylinder, set at disc premark.
/E No. of cylinders of the disc. It is used to check for I/O requests in order to refuse any operation causing a disc fault status. It must be set at disc premark.
Location /10 Is the first free cylinder following the last DAD in the VTOC.
Location /12 Address of the first cylinder of the spare tracks, used to replace bad ones detected on the disc.

Note: Address of a cylinder is the relative position of the cylinder from the cylinder zero of the disc (e.g. 0, 1, 2...).

Current entry consists of 8 words as follows:

/14 DAD name : 6 ASCII characters, left justified and space-filled.

/1A bits 0 - 7: No. of interlaces
bits 8 - 15: No. of sectors per track

Used to compute the physical sector No , thus:

SN = LSxNI

where:

SN is the physical sector No.
LS is the logical sector No.
NI is the No.of interlaces modulo No.of sectors per track.

/1C Sector size in characters.

/1E No. of cylinders in the DAD.

/20 Address of the first cylinder in the DAD, i.e. the cylinder No. (0, 1, 2, etc) of the first cylinder of the DAD.

/22 No. of sectors per granule.

These 8 words are repeated for each DAD.

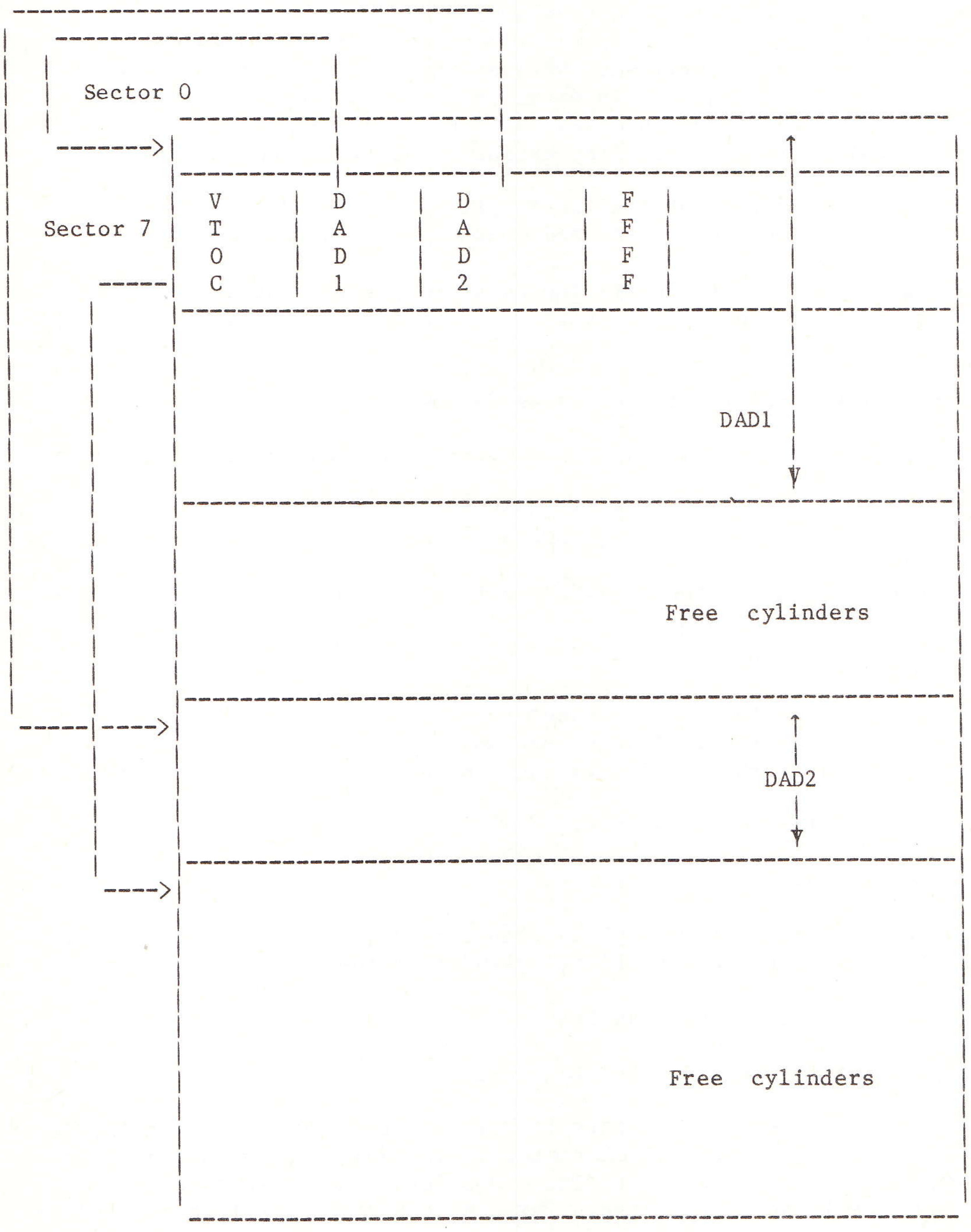
When a DAD is deleted, the entry is removed from VTOC and the following entries are moved upward 16 characters. The number of used characters is updated. Thus, entries in VTOC are ordered in the same sequence as the DADs are found on the disc. This is used to detect freed cylinders on the disc when a DAD is created.

The last entry is followed by FFFF when the sector is not full.

The first cylinder of the first DAD is always the cylinder 0 of the disc. Since the first granule contains not only DAD information but also disc structure characteristics, it cannot be deleted by Librarian processor. When the User desires to remove the first DAD, then he has to re-premark the disc.

DISC ORGANISATION

The following diagram shows the physical disc layout:



Structure of a DAD

The sectors of the first granule of the DAD are used as follows:

a) Sector Zero

This contains the BITTAB for the DAD, i.e. the table giving the allocation status of all the granules of the DAD. It starts at location /54 of the DAD:

Location No.

dec	hex	
0	/0	used only with X1215, for sector ID
2	/2	not used
4	/4	Locations 4 to /52 of the DAD are not used, except in the first DAD where they contain the VOLAB of the disc.
82	/52	
84	/54	Number of characters of BITTAB, this word excluded
86	/56	First word of BITTAB
88	/58	Second word of BITTAB
		etc.

Each bit of the BITTAB represents the status of the associated granule; a bit set to 0 means allocated (or not existing); set to 1 means free. The length of the BITTAB depends on the size of the DAD and the number of sectors per track in the DAD.
The remaining words of the sector, if any, are not used; they are reset to zero.
The BITTAB is created when the DAD is initialised (e.g. by Librarian processor).

b) Sector 1: logical sector No. 1 of the DAD is not used, except for the first DAD of the disc, where it is used to contain the IPL.

c) Sectors 2 to 5: are used to contain a catalogue of Users in the DAD. Entries consist of 8 words as follows:

Sectors 2 - 5: CATALOGUE

Locations		
0	Reserved for sector ID (X1215/6)	
2	Number of used characters in the sector	
4	C	A
6	T	A
8	L	Ø
/A	G	␣
/C	0	
/E	0	
/10	0	
/12	0	
Current Entry	User identifer up to 8 characters left justified filled with spaces	
	Pass Word	
	Pass Word	
	Address of the first sector of directory	
	Account number	

Remarks

For a deleted entry:

The first word of the USERID is reset to zero and the last entry of the catalogue is followed by a word = /FFFF (-1).

When a new USERID is declared, the system tries to use a deleted entry before taking a free entry at the end of the catalogue. Number of used characters includes the first two words and the /FFFF entry if any.

When the last entry of the current sector is filled, the flag /FFFF is written at the beginning of the next sector, if any.

- The remaining records 6, 7, ..., up to the end of the first granule of the DAD are unused.

Note: For the first DAD, 6, 7, are used to describe the physical structure of the disc.

d) Other sectors of the first granule of the DAD are not used. (They can be used for the physical disc unit if it is the first DAD of the disc.)

The remaining granules can be used either to contain the User Directory (one granule per directory), or be allocated to the User files. Their statuses are indicated in the BITTAB, which is loaded into memory when the DAD is declared in a machine. The BITTAB on the sector 0 of the DAD is updated each time a file is catalogued or removed from the directory.

Directory Format

The directory (1 granule) is used to contain the names and characteristics of all catalogues files belonging to one User (there can be several Users in a DAD, up to the limit of the catalogue).

One entry is catalogued for each version of a file. They are created when the file is catalogued and removed when the file is deleted. the first entry of the directory is used to contain the name of the User. Current entries are of 8 words.

The layout of the directory is as follows:

Locations

	(0	Sector ID, used only with X1215/6 discs	
	(2	Number of used characters (first two words included)	
	(4	USERID	
	(6	USERID	1st entry of directory
First	(8	USERID	
Entry	(10	USERID	
	(12	0	
	(14	0	
	(16	0	
	(18	0	
	(	File name (6 characters)	
	(	File name	
	(	File name	
Current	(	File type	
Entry	(	YEAR 6 | 7 MONTH 10 | 11 DAY	
	(	Number of sectors (first two included)	
	(	Address of the first sector of the file	
	(	S | P | Sy | In |	

Explanation

- The first 3 words of a current entry contain the file name of up to 6 ASCII characters, left-justified and space-filled to the right.
- The fourth word is the file type. in ASCII characters as follows:

OB	Object file
SC	Source file
LM	Load module
UF	User data file
EF	Extended file

- The fifth word is the catalogued date:

```
 ----------------------------------------------------------------
|            YEAR              |     MONTH      |      DAY       |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
 ----------------------------------------------------------------
  0   1   2   3   4   5   6   7   8   9  10  11  1213  14   15
```

Year = 7 least significant bits of the year
Month and day are in binary.

The catalogued date is the date when the file is physically "kept", i.e. its name is introduced to the directory. If this operation is done when the day is updated, it can differ from the date the job is started.

- The sixth word is reset to zero for non-consecutive files. For consecutive files, it contains the number of sectors in the file (including the first two); it is equal to the product of the number of granules in the file by the number of sectors per granule.
- The seventh word points at the first sector of the first granule of the file.
- The eigth word contains various flags and the version Number as follows:

```
 ----------------------------------------------------------------
| S   P   Sy  In                                 |  Version No.   |
|   |   |   |   |   |   |                    |   |                |
 ----------------------------------------------------------------
  0                                           11  12            15
```

S	=	0 Shared file
S	=	1 Unshared file
P	=	1 Write protected
P	=	0 Not protected
Sy	=	0 Not system file
Sy	=	1 System file
In	=	1 Invisible file
In	=	0 Visible file

- The last entry of the directory is followed by an /FFFF flag, unless it is the last sector of the directory.
- The deleted entry has its 1st word reset to zero. It can then be re-used for a new file.

File Formats

A file is composed of a set of granules which may or may not be consecutive. For consecutive files, all the granules must be requested at assign time, before the file is accessed. For non-consecutive files,granules are allocated dynamically when a sector is written, whether the file is catalogued or temporary. The sectors of a file are used as follows:

- The first sector: reserved. Not used in the present system, but can be used to contain all User names authorised to access the file.

- Second sector: used only for non-consecutive files. It is the granule table (GRANTB), i.e. the table containing the addresses of al granules belonging to the file.

The layout of the GRANTB sector is as follows:

0	Sector identifier
2	Length in characters of GRANTB
4	Address of the first granule
6	Address of the second granule
8	.
/A	.
	etc.
	(unused words are reset to zero)
Lst - 4	0
Last - 2	0
Last word	0

The third sector is the first sector for the data of the file. In direct access, it corresponds to sector zero of the file. The sectors following are used for file data.

Note: For non-consecutive files, data space = the maximum number of granules - sector size (in words) - 5.

DFM Files

The disc basic access method is called DFM (Disc File Management). Files can be consecutive or not. They have the standard structure as described in the section 'File Formats'. The access mode can be either direct or sequential (variable length).

Direct access:

The file is accessed directly on the sector level. The calling program has to specify the sector number, relative from the beginning of the file.

If the file is consecutive, the maximum number of sectors that can be accessed is limited either to 32K-2, or to the maximum size of the DAD.For non-consecutive files, it is determined by the sector size.

Sequential Files

Variable Length Records are blocked within sequential files.

Sector format:

```
Word    0    1                                3 words reserved
         --------------------------------------------------
        | ID | L |             data              | 0 | 0 | 0 |
         --------------------------------------------------
```

* ID = cyl identifier (X1215/6 only)
* L = length of used area (from 0 to sector size - 10 char) as follows:

Sector status indicators

```
        |---------
        |_________|_________________________________________
        | D | S | F |                  Length               |
         ---------------------------------------------------
```

* D = 1 Sector is deleted from the file
* S = 1 Sector contains a segment mark (:EØS)
* F = 1 Sector contains a file mark (:EØF)

Logical record format:

Logical records used in sequential access are compressed, trailing blanks removed, and the records are blocked as follows:

```
Word    0    1    2
         --------------------------------------------------
        | L1 | L2 |             Data                | S | D |
         --------------------------------------------------
                   <--------------------------------------->
                                   L1
```

- S is the relative sector address within the file containing the first word of the record.

- D is the displacement in the sector S of the first word of the record as a No. of characters.

- L1 = length, in characters, of the written record,
  including S and D,
  excluding L1 and L2.

  as follows:

```
Bits     0    1    2                                              15
         ----------------------------------------------------------
        | V  | S  | F  |                     L1                    |
         ----------------------------------------------------------
```

- V = 1 Current record is deleted from the file
- S = 1 Current record is a segment mark (EOS)
- F = 1 Current record is a file mark (EOF)

L2 is the length of the record, in characters, given in the User ECB when the record was written.

S and D are always stored on the same sector. So are L1 and L2.

Chapter 11

TDFM

General

TDFM comprises:

1) The EDF Standard Processor

   This is activated in the background machine by a BCL Standard Processor Call.
   EDF allocates disc space for the TDFM files and intialises the EDF system data on these files.
   It also provides facilities to copy, restore, delete, reorganise and perform backup operations for TDFM files.

2) Various extensions to the MAS I/O LKM (such as LKM 1) and MAS assignment facilities.
   These enable the user to perform input and output operations on TDFM files in either foreground or background programs.

3) The TDFM system control program is memory resident. It uses, in an overlay structure, disc resident segments, of which four can be in memory at the same time. When a new segment has to be loaded, the least recently used segment is released from memory.

T:SEG

This table is used for the segments running in extended system mode. It indicates the resident segments and gives the SCT address for each segment.

Layout:

Location	Contents
/0 - /E	Each bit of these 8 words indicates if corresponding segment is memory resident.
/10	Physical sector number of the MAS monitor load module.
/12	First SCT (S:TDFM)
/14-/114	SCT addresses for each segment

S:TDFM

This table is the segment control table (SCT) for TDFM.

Layout:

Location	Contents
0	Next SCT or zero
2	PCT address of X:TDFM
4	Physical page number of transient area 1
6	Physical page number of transient area 2
4	Physical page number of transient area 3
/A	Physical page number of transient area 4
/C	Segment number in transient area 1
/E	Segment number in transient area 2
/10	Segment number in transient area 3
/12	Segment number in transient area 4
/14	Stack
/16	For allocation
/18	of transient
/1A	arrays
/1C	Logical loading address of segment
/1E	Chaining word for segment block

TDFM System Initialisation

This section describes the initialisations carried out by the system on receipt of the following commands:

1) Assign a Filecode to an Extended File.

This is given, either by the ASG command (under the BCP or FCL processor), or by LKM 23 (Assign a Filecode).

The system creates the following control blocks in its dynamic area:

one FCT (type 6)	6 words
one FDC which is composed of:	
a general part	83 words
an index part	17 words x No. of index files
a data part	30 words x No. of data files
Total minimum length	130 words

2) Request 'Transaction Ready'

The system creates tables in the system dynamic area:

one TRT (Transacton Table)...	10 words
one or more EFT (extended File Table)...	26 words x No.of extended files
Total minimum memory used	36 words

Note: If the transaction operates on 'n' extended files, 'n' EFT will be created.

3) Request 'Transaction Finished'

The system releases the TRT and all EFTs involved in this transaction.

4) Requests with 'Attach'

For each 'Attach' request an 'AT' block is created (8 words). All AT blocks are chained together from the relevant FDC.

These blocks are freed following requests with 'detach', or following the transaction finished request.

In addition to the File Descriptor Control Block (FDC), the Transaction Table (TRT), the Extended File Table (EFT) and the Attach Blocks (AT), other reserved areas are required, as follows:

Working storage security area (only provided if security was requested): the length of this area is 20 words.

Buffer Pool; the number of buffers is declared at system generation time and can be changed by the following EDF command:

```
SBUF_MBUF = <Max. No. of Buffers>
```

The length of such buffers is equal to the sector size of the DAD containing the index and data files.

Layout of the Control Tables

The following diagrams show the layout of these tables:

FDC Table Layout

1) Extended File Entry:

/00	0	PACK NUMBER (DESCR. FILE)
/02	1 2 3	DAD NAME (DESCR. FILE)
/08	4 5 6 7	USERID (DESCR. FILE)
/10	8 9 10	EXTENDED FILE NAME
/16	11	DESCR. FILE SECTOR SIZE
/18	12	REL. SEC. NO. OF LOGGED SECTION IN DESCR. FILE
/18	13	REL. SEC. NO. OF SCRATCH AREA IN DESCR. FILE
/1C	14	CURR. SCRATCH POINTER
/1E	15	NO. OF ENTRIES /SEC. IN UNLOGGED SECT. OF DESCR.
/20	16	NO. OF ENTRIES /SEC. IN LOGGED SECT. OF DESCR.
/22	17 18	MAX. NO. OF RECORDS IN EXT. FILE
/26	19 20	DATE OF CREATION OR LATEST RESTORATION
/2A	21	NO. OF INDEX LINKS
/2C	22	MAX. NO. OF INDEX LEVELS
/2E	23	MAX. KEY LENGTH
/30	24	/8000 OR NO. OF KEY USED FOR DISPATCING (CRITERION KEY)
/32	25	NO. OF DATA FILES
/34	26	MAX. SEC. SIZE (OF ALL SUBFILES)
/36	27	PASSWORD IN READ MODE
/38	28	PASSWORD IN UPDATE MODE

/3A	29	LINK
/3C	30	STATUS OF EXTENDED FILE
/3E	31 32	CURRENT NUMBER OF RECORDS
/42	33 34	CURRENT NUMBER OF DELETED RECORDS
/46	35	FREE WORD
/48	36	FILE CODE OF DAD OF DESCR. FILE
/4A	37	REL. SECT. NO. IN DAD OF SECT. 0 OF DESCR.
/4C	38	ADDR. OF DAD CONT. TABLE FOR DESCR. FILE
/4E	39	DWT ADDR. FOR DESCR. FILE
/50	40	ASSIGN COUNT
/52	41	OPEN COUNT
/54	42	ADDR. OF Q ON FILE FREE
/56	43	ADD. OF LIST OF ATTACHED RECORDS
/58	44	ADDR. OF MAC. DEF. BLOCK OF TR. READY WAITING FOR EXCL. ACCESS ON THIS FILE
/5A	45	TRT ADDR. OF TR. HAVING EXCL. ACCESS
/5C	46	FREE WORD
/5E	47	POST PROCESSING WORD
/60	48	REQUESTS'S EFT ADD.
/62	49	EFFECTIVE LENGTH
/64	50	BUF1 ADDR.
/66	51	ADDR. OF FDC FILE ENTRY OWNING SECTOR IN BUF1
/68	52	SECT. NO. IN BUF1
/6A	53	BUF2 ADDR.
/6C	54	ADDR. OF FDC FILE ENTRY OWNING SECTOR IN BUF2
/6E	55	SECTOR NO. IN BUF2
/40	56	BUF3 ADDR.

/72	57	ADDR. OF FDC FILE ENTRY OWNING SECTOR IN BUF3
/74	58	SECTOR NO. IN BUF2
/76	59 66	8 WORD SAVE AREA FOR REGS. DURING PHYSICAL I/O
/86	67 74	ADDITIONAL 8 WORDS REG. SAVE AREA
/96 /A4	75 . . . 82	ADDITIONAL 8 WORDS REG. SAVE AREA

A more detailed description of some of these locations follows:

- Date of creation or latest restoration (Words 19-20, locations /26 - /28)

```
 0        4 5      8 9      13 14 15     0  1   3 4       9 10       15
 ------------------------------------    ------------------------------
| Year /77 | Month | day    |      | | | 0 |     | Minutes | Seconds  |
 ------------------------------------    ------------------------------
                            |________________|
                                  Hours
```

- Status of extended file (Word 30, location /3C):

```
  0   1   2   3   4   5   6   7    8     9   10   11    12   13  14   15
 -------------------------------------------------------------------------
| F |   | IOV|   |   |   |    |SCR |DOV | EX| BA | BMD | CR | L | BO | BU |
 -------------------------------------------------------------------------
```

F = 1 : File free
IOV = 1 : Index overflow
E = 1 : File empty
SCR = 1 : If file protected in BO mode and scratches cleared at 1st open
DOV = 1 : Data overlow
EX = 1 : File under exclusive access
BA = 1 : Buffers allocated
BMD = 1 : Buffers may be allocated
CR = 1 : File crashed
L = 1 : File locked for back-out recovery
BO = 1 : File protected in back-out mode
BU = 1 : File protected in back-up mode

- Post-processing Word (Word 47):

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
D	A	H	C												

D = 1 : Current record to be detached (co-ords. in EFT)
A = 1 : Current record to be attached
H = 1 : H. Count to be returned to user
C = 1 : Q.Coord. to be returned to user

2) Index File Entry:

/00	0	DAD FILE CODE
/02	1	REL. SECT. NO. OF SECT. 0 OF FILE IN DAD
/04	2	DAD CONT. TABLE ADDRESS
/06	3	DWT. DAD
/08	4 5 6	INDEX FILE NAME
/0E	7	INDEX FILE SECT. SIZE
/10	8	NO. OF LEVELS
/12	9	KEY POSITION
/14	10	KEY LENGTH
/16	11	PADDING WORD
/18	12	REL. SECT. NO. IN INDEX FILE OF 1st SECT. OF SCRATCH
/1A	13	CURRENT SCRATCH POINTER
/1C	14	STATUS OF INDEX FILE
/1E	15	REL. SECT. NO. IN INDEX FILE OF NEXT FREE SECTOR
/20	16	HIGHEST SPLIT SECT. NO.

The format of the status word is as follows:

- Status of index files (Word 14, location /1C):

```
  0   1   2   3   4  5   6   7   8                                           15
 -------------------------------------------------------------------------------
| L |    | OV| B |   | P |    |     |                 KN                        |
 -------------------------------------------------------------------------------
```

```
L      = 1 : Last index file
OV     = 1 : Index file in overflow
B      = 1 : Bijective key
P      = 1 : Primary key
KN     = 1 : Key number (0, 1, ...)
```

3) Data File Entry:

```
             -----------------------------------------------------------
/00    0    |  DAD FILE CODE                                            |
            |-----------------------------------------------------------|
/02    1    |  REL. SECT. NO. OF SECT. 0 OF FILE IN DAD                 |
            |-----------------------------------------------------------|
/04    2    |  DAD CONT. TABLE ADDRESS                                  |
            |-----------------------------------------------------------|
/06    3    |  DWT. ADDR.                                               |
            |-----------------------------------------------------------|
/08    4    |  DATA FILE SECT. SIZE                                     |
            |-----------------------------------------------------------|
/0A    5    |  HIGHEST REL. SEC. NO. IN DATA FILE                       |
            |-----------------------------------------------------------|
/0C    6    |                                                           |
       .    |                                                           |
       .    |  HIGHEST VALUE OF CRIT. KEY IN THIS DATA FILE             |
       .    |                                                           |
      15    |                                                           |
            |-----------------------------------------------------------|
/20   16    |  STATUS OF DATA FILE                                      |
            |-----------------------------------------------------------|
/22   17    |                                                           |
      18    | CUR. NO. OF RECORDS                                       |
            |-----------------------------------------------------------|
/26   19    |                                                           |
      20    | CUR. NO. OF DELETED RECORDS                               |
            |-----------------------------------------------------------|
/2A   21    |  REL. SECT. NO. OF NEXT FREE RECORD                       |
            |-----------------------------------------------------------|
/2C   22    |  DISPL. OF NEXT FREE REC. IN SECTOR                       |
             -----------------------------------------------------------
```

- The 'Status of Data File' Entry (Word 16, location /20):

```
  0   1                                  8                                    15
 -------------------------------------------------------------------------------
| L | OV|                                |                DN                    |
 -------------------------------------------------------------------------------
```

```
L      = 1 : Last data file
OV     = 1 : Data file in overflow
BN     = 1 : Data file number (0, 1, ...)
```

TRT Table Description:

/00	0	LINK WORD
/02	1	TRANSACTION NUMBER
/04	2	ADDR. OF EFT CHAIN FOR THIS TRANSACTION
/06	3	FILE COUNT (TOT. OP. COUNT FOR TRANS. READY)
/08	4	MACRO COUNT
/0A	5	ADDR. OF TR. FINISHED/CANCEL MACRO. DEF. BLOCK WAITING FOR MAC. COUNT=0
/0C	6	STATUS
/0E	7	CUR. SECT. NO. IN BO. SUBFILE
/10	8	PCT ADD. OF PROG WHICH ISSUED THE TR. READY

- The Status Word (Word 6, location /C) has the following format:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
TFR	BU	BO	FF		AB										

TFR = 1 : When macro 'Transac. Finished' or 'Finish + Cancel' received.
BU = 1 : At least one file opened in Update or exclusive mode and protected in B-Up mode (BU logging is possible).
BO = 1 : No file opened in Update or exclusive mode and only protected in B-Up mode (BO logging is possible).
FF = 1 : 'Finish' or 'Finish + Cancel' forbidden
AB = 1 : Transaction has been aborted.

EFT Table Description:

/00	0	LINK WORD
/02	1	TRT ADDRESS
/04	2	FDC ADDRESS
/06	3	INTERNAL FILE NO. FOR LOGGING FOR THIS TRANSAC. OR /8000
/08	4	STATUS
/0A	5	PTR. TO DATA FILE ENTRY IN FDC (WRITE)
/0C	6	PTR TO INDEX FILE ENTRY IN FDC
/0E	7	NO. OF ENTR. IN LVL. LIST (MAX. NO. OF LEVELS)
/10	8	KEY LENGTH
/12	9	H. COUNT TO BE RETURNED TO USER
/14	10	DATA FILE NUMBER
/16	11	REL. SEC. NO. OF REC. IN DATA FILE
/18	12	DISPL. OF REC. BEGIN. IN SEC.
/1A	13	RECORD'S LENGTH
/1C	14 . . . 23	KEY VALUE
/30	24	TOTAL H. COUNT FOR CUR. CHAIN

Level List

/00	0	STATUS
/02	1	SEC. NO. OF KEY ENTRY
/04	2	DISPLACEMENT OF ENTRY IN THIS SECTOR
/06	3	INDIR. ADDR. OF BUF. CONTAINING THIS ENTRY
/08	4	DISPL. OF PREVIOUS ENTRY/BUSY CHAIN PTR.

- Status of EFT Table (Word 4):

```
 0   1   2   3   4   5   6   7   8   9   10   11   12   13  14  15
 -------------------------------------------------------------------
|   | R | U | E |   |   |SPR|    | SP| SQ| SPL|LVLM|    |   |    |   |
 -------------------------------------------------------------------
```

R   = 1 : Access opened in read mode
U   = 1 : Access opened in update mode
E   = 1 : Access opened in exclusive mode
SPR = 1 : LVL set by macro 'Read Previous Entry'
SP   = 1 : LVL set by 'Posit^{n}' macro. In this case "Key value" contains the value used in this 'Posit^{n}' macro.
SO   = 1 : LVL set by macro 'Read' and 'Read Next'
SPL = 1 : If splitting into one or more indices for current Write macro.
LVLM = 1 : LVL to be updated at the end of the current Write macro.

- Status of Level List, Entry (Word 0, location 0):

```
 0   1   2   3   4   5   6   7   8   9   10   11   12   13  14  15
 -------------------------------------------------------------------
| L | F |   | NF| E | G | L | D |EOF|    |    |    |    |   |    |   |
 -------------------------------------------------------------------
```

L   = 1 : Last entry in level list
F   = 1 : First entry in level list
NF   = 1 : A free entry exists in the sector pointed by Word 1.
E   = 1 : The key searched for is equal to the one pointed to by Word 2
G   = 1 : The key searched for is greater than the one pointed to by Word 2
L   = 1 : The key searched for is lower than the one pointed to by Word 2
D   = 1 : The entry pointed to by Word 2 is a deleted entry
EOF = 1 : The entry pointed to by Word 2 is the last one of the index.

ATTACHED RECORD BLOCK

/00	0	LINK WORD
/02	1	PTR. TO 'AT' QUEUE ON THIS RECORD
/04	2	TRANSACTION NUMBER
/06	3	EFT ADDRESS
/08	4	DATA FILE NUMBER
/0A	5	REL. SEC. NO. IN DATA FILE OF REC. BEG.
/0C	6	DISPL. OF REC. BEG. IN SECTOR
/0E	7	RECORD'S LENGTH

WORKING STORAGE SECURITY

/00	0	SECURITY STATUS
/02	1	NEXT REQUEST NUMBER
/04	2	ACTIVE TRANSACTION COUNT
/06	3	B-OUTQ ADDRESS /0
/08	4	LOCKQ ADDRESS /0
/0A	5	B-UP. F.CODE IN SYSTEM MACHINE
/0C	6	B-UP. IDENT (DATE/HOUR)
	7	
/10	8	DATE AND START TIME OF RUN
/12	9	
/14	10	SECT. SIZE IN B-UP. FILE
/16	11	HIGHEST SECT. NO IN B-UP. FILE
/18	12	NEXT FREE REC. SEC. IN B-UP. FILE
/1A	13	DISPLAY OF NEXT FREE REC. IN THIS SEC.
/1C	14	B-OUT F.CODE IN SYSTEM MACHINE
/1E	15	NO. OF SECTS. PER GR. IN B.O. FILE
/20	16	SECT. SIZE IN B.O. FILE
/22	17	ADDR. OF B-OUT GRANTB
/24	18	ADDR. OF W.S. FOR B-OUT RECOVERY
/26	19	B-OUT IDENT (DIH OF B.O. CREATION)
/28	20	B-OUT IDENT (DIH OF B.O. CREATION)

- Working storage security status (Word 0):

```
 0    1    2    3    4     5     6    7    8    9    10    11    12    13   14   15
-------------------------------------------------------------------------------------
| L | WR|     |BOC|BOF|BOFOR|    |    |BUR|BUC|BUOV|      |    |    |     |     |
-------------------------------------------------------------------------------------
```

L = 1 : TDFM locked during B-Out initialisation
WR = 1 : Wait for recovery; TDFM only accepts LKM B-Out or requests from B-Up procedure.
BOC = 1 : Back-Out file crashed
BOF = 1 : Back-Out recovery failed
BOFOR = 1 : Back-Out recovery forbidden
BUR = 1 : Back-Up recovery running
BUC = 1 : Back-Up file crashed
BUOV = 1 : Back-Up file overflow

Working-storage Back-out recovery

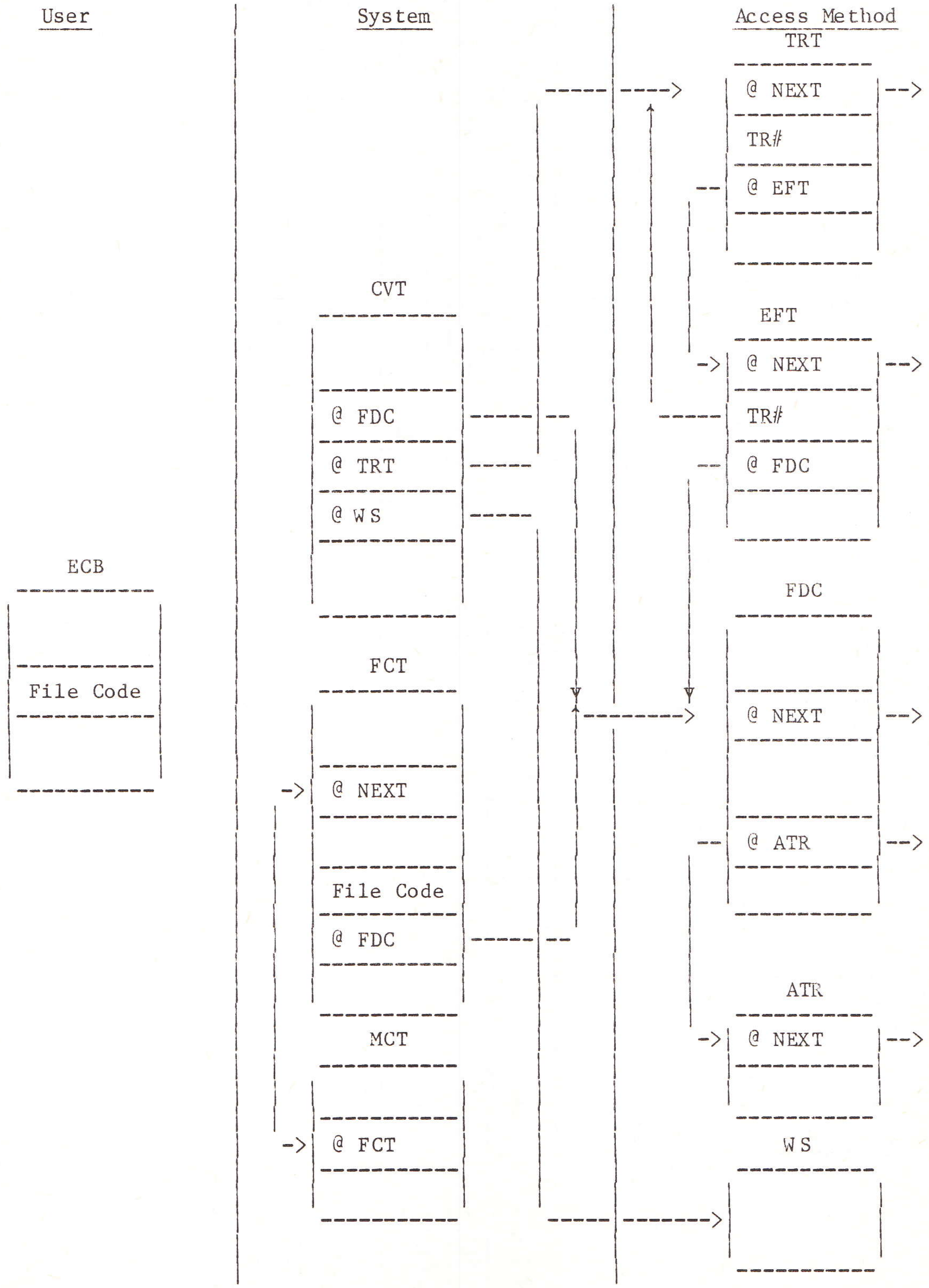
```
              -----------------------------------------------------------
/00     0    |  UNDO MACRO COUNT                                         |
             |-----------------------------------------------------------|
/02     1    |  ADDRESS OF RUN CHAIN                                     |
             |-----------------------------------------------------------|
/04     2    |  ADDRESS OF FILE CHAIN                                    |
             |-----------------------------------------------------------|
```

Chaining of the Control Blocks

The linkages which unite these control tables into a TDFM system can be shown diagrammatically as follows:

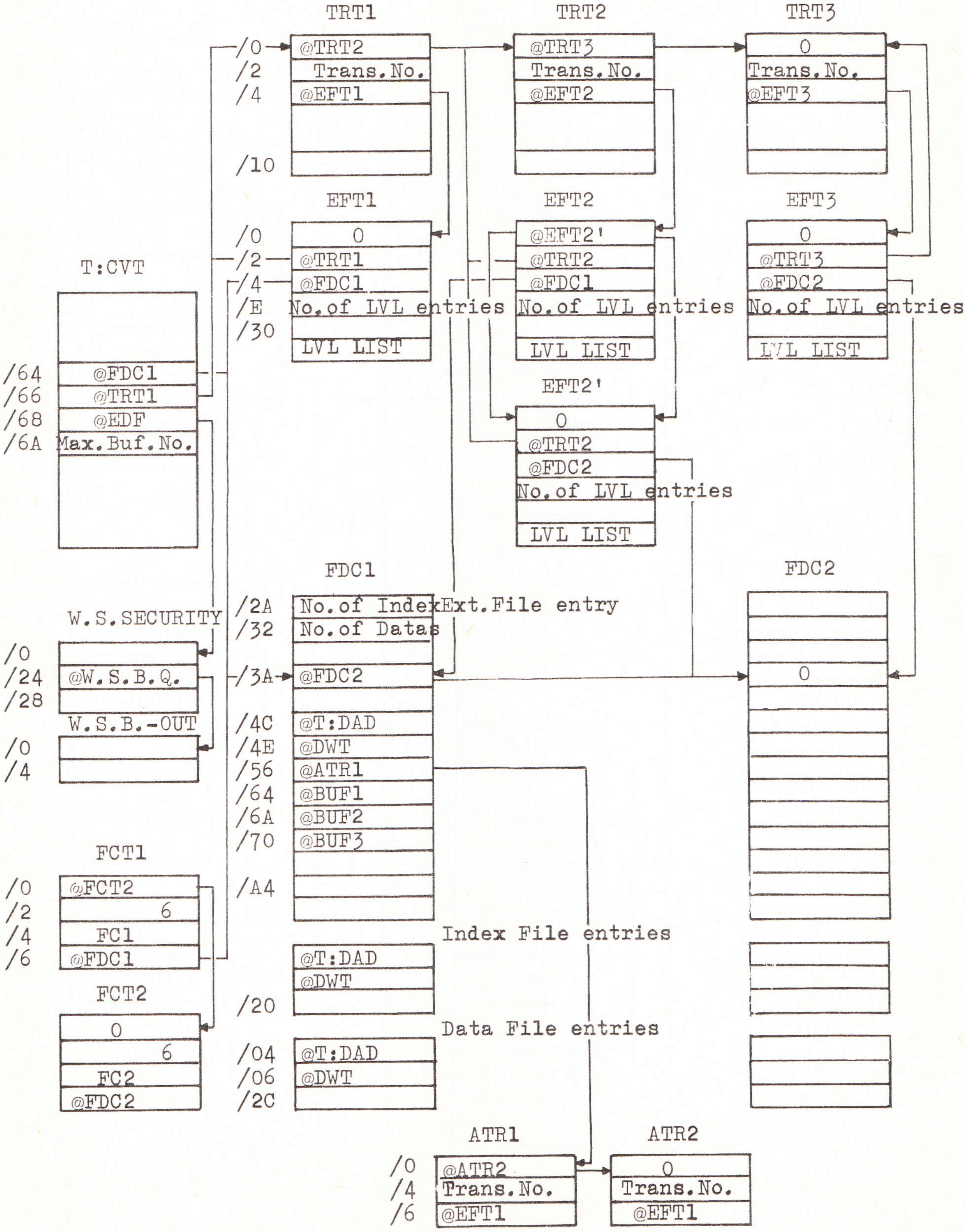
T.D.F.M. CONTROL BLOCKS DESCRIPTION

CHAINING OF CONTROL BLOCK



Note: @ = 'address of'

EXAMPLE OF CONTROL BLOCK CHAINING.



TDFM I/O REQUESTS

List of Requests	Value in A7
Request operations on a File/Files	
Transaction Ready	/25
Transaction Finished	/3B
Finish and Cancel Transaction	/28
Abort Transaction	/27
Position	/3C
Requests on a Record	
a) Read:	
Read on Key	(/0A)
Read on Key and Attach	(/1A)
Read Next	(/02)
Read Next and Attach	(/12)
Read Previous	(/3F)
Read Previous and Attach	(/3A)
Read on Physical Co-ordinates	(/1B)
Read on Physical Co-ordinates and Attach	(/1C)
b) Modify:	
Replace	(/2E)
Replace and Detach	(/3D)
Delete	(/2D)
Delete and Detach	(/3E)
Write	(/0B)
Write and Attach	(/2C)
c) Detachment:	
Detach a Record	(/2A)
detach all Records attached to a transaction	(/2B)
Security Requests	
Back-out Recovery	(/29)

RETURNED STATUS

Apart from the general LKM 1 error codes, the user may receive the following codes which are specific to TDFM:

Warning Status

Value	Meaning
/0008	The requested length for a read operation is less than the record length: the transferred length = the requested length; the effective length = the record length.
/1000	The record just read is the start of a homonymous chain.
/1008	Both the warnings for /1000 and /0008 apply.
/1001	The record to be deleted has already been deleted.
/1002	Position on Key request with key higher than all existing keys; file is positioned at EOF.
/1003	No transactions for back-out recovery.
/D000	Internal system status for back-up recovery - ignore.

Error Status

Value	Meaning
/A000	Requested file not opened by requesting transaction.
/A001	Detach forbidden because requested file is back-out protected.
/A002	Detach forbidden, file was damaged by incorrectly performed modifying operation.
/A004	Dynamic area overfow in the system machine; cannot allocate buffer for Transaction Table, EFT or file buffer. If this error occurs with Cancel, the Cancel may be retried.
/A005	Back-out recovery refused because run without security.
/A006	Back-out recovery not first request of run.
/A007	Unknown order code in A7.
/A008	Buffer address not in user's area.
/A009	Requested length is zero.
/A00C	Back-out recovery compulsory; any other request refused.
/A00D	Unknown transaction number.
/A00F	Transaction Ready refused because back-out recovery failed.
/A010	Transaction Ready refused because transaction already exists.
/A011	Transaction Finished already received.
/A012	Overflow of system request number for logging.
/A013	Transaction already aborted.
/A024	Transaction cannot be cancelled because none of the filesopened in update mode is protected in back-out mode.
/A025	Transaction Finished forbidden because one modifying request issued by the transaction on a protected file was incorrectly performed.
/A026	Cancel Transaction forbidden (see /A025)
/A027	Abort refused because normal end of transaction allowed.
/A060	Transaction Ready for zero or negative number of files.
/A061	Two entries for the same file in Transaction Ready ECB.
/A062	Transaction Ready uses unknown filecode.
/A063	Transaction Ready on a file under exclusive access for another transaction.
/A064	Date of one protected file greater than run date.

/A06C Transaction Ready in Read mode for an empty file.
/A06D File still locked after back-out recovery failure.
/A06E Transaction Ready with unknown opening mode.
/A070 Transaction Ready refused because back-out file not declared, or not-on-line.
/A071 Transaction Ready refused because back-up file not declared, or not-on-line.
/A072 Transaction Ready on a file damaged by previous modification incorrectly performed.
/A120 Unknown key name.
/A123 File empty.
/A150 Erroneous coordinates or wrong requested length.
/A180 Sequential read on a file not previously positioned.
/A181 Read and Attach forbidden on a file opened in Read mode; file positioning not destroyed.
/A183 Read Next refused because file is positioned at EOF.
/A185 Record attached to aborted transaction; queuing forbidden.
/A186 Record attached to unknown transaction.
/A187 Record to be read attached to another transaction; queuing refused to avoid deadlock.
/A188 Read without queuing on attached record.
/A243 Read on Key value less than or equal to padding key value.
/A244 Read on non-existent Key value; file positioned.
/A245 Read on Key value greater than all existing key values - file positioned at EOF.
/A246 Record deleted.
/A2A0 Write without Attach attempted on a protected file.
/A2A1 Modification attempted on a file opened in Read mode.
/A2A2 Index overflow; Write not performed.
/A2A3 One key of written record less than or equal to padding key.
/A2A4 Requested length for Write greater than 4095 bytes.
/A2A5 data overflow - Write not performed.
/A2A7 No criterion key defined for Write on requested file.
/A2A8 Written criterion key value greater than highest key declared at generation for last data file.
/A2B0 Write performed, but secondary bijective key becomes multiform (key bumber returned in ECB Word 10).
/A2B1 Written primary key value already exists.
/A2B6 Write uses an invalid data file number.
/A2B7 One of the written keys lies outside the user's buffer.
/A2B8 Written primary key value exists in deleted record attached to a still running transaction which may be cancelled.
/A301 Modification of a non-attached record.
/A302 Modification of a record attached to another transaction.
/A304 Detach forbidden on a protected file.
/A305 Length of replacing record not equal to record length on disc.
/A306 Not all key values are identical in the replaced and replacing records.
/A36A Delete Record attempted using a key value not in the index.
/A36B Record coordinates not found in index; Delete undone.
/A371 The record in data file is already deleted; file is probably corrupted.
/A372 Key value in data record not found in index.
/A373 Record coordinates not found in index; I/O error when releasing.

/A420 Detach uses wrong record coordinates in the ECB.

/A482 Read Previous attempted on a file positioned on the first key value; file positioning is destroyed.

/A4F0 Previous logging error forbids further use of back-up file.

/A4F4 Back-up file overflow.

/A520 Previous logging error forbids further use of back-out file.

/A525 Back-out file overflow (Disc GRANTB).

/A526 Overflow of DAD containing the back-out file.

/A532 Transaction Ready refused because too many simulataneous transactions.

/A580 One file of a transaction to be undone is not assigned.

/A590 One of the files involved in the undoing of a transaction has been damaged after the Cancel request.

/A596 Undo Write/Delete failed because a key value in the data record was not found in the corresponding index.

/A5F3 Back-out impossible because end of back-out file met before end of one transaction.

/A602 Back-out recovery stopped because inconsistency detected in back-out file.

/A603 Result block for back-out recovery is too small.

/A620 In ECB for back-out, user's buffer address not word-aligned.

/FFFF Inconsisten~~y~~cy detected in one index sector.

Disc I/O Errors

All the status code values relevant to disc I/O errors are of the form /Bxxx:

/B0xx Disc not damaged.

/B1xx Requested file(s) damaged (back-up logging not performed if file protected).

/B2xx Back-up logging incorrectly performed.

/B3xx Back-out logging incorrectly performed.

/B4xx I/O error during Cancel or Back-out recovery.

Note: In the case of status codes /B441 or /B443, the back-out recovery is successfully performed but the back-out file is damaged and a new one must be generated.

Error Code Cross Reference Table

The table below contains a list of error codes, together with the requests which may give rise to them. The requests are in coded form and these codes are listed here for convenience.

Code	Request
a	Back-out
b	Detach All Records
c	Detach One Record
d	Write and Attach
e	Write
f	Delete and Attach
g	Delete
h	Replace and Detach
i	Replace
j	Read on Physical Co-ordinates and Attach
k	Read on Physical Co-ordinates
l	Read Previous and Attach
m	Read Previous
n	Read Next and Attach
o	Read Next
p	Read on Key and Attach
q	Read on Key
r	Position
s	Abort
t	Cancel
u	Transaction Finished
v	Transaction Ready

Cross Reference Table

Error Code	Affected Request
/A000	c to r inclusive
/A001	c
/A002	d to r inclusive
/A004	a to r and t, u, v
/A005	a
/A006	a
/A00C	b to r and t, u, v
/A00D	b to u inclusive
/A00F	v
/A010	v
/A011	b to u
/A012	d to i and t, u, v
/A013	b to v
/A024	t
/A025	u
/A026	t
/A027	s
/A060	v
/A061	v
/A062	v
/A063	v
/A064	v
/A06C	v
/A06D	v
/A06E	v
/A06F	v
/A070	a, v
/A071	v
/A072	v
/A120	p, q, r
/A123	p, q, r
/A180	j, k
/A181	l, m, n, o
/A183	j, l. n, p
/A2B6	d, e
/A2B7	d, e
/A2B8	d, e
/A301	f, g, h, i
/A302	f, g, h, i
/A304	f, h
/A305	h, i
/A306	h, i
/A36A	f, g
/A371	f, g
/A420	c
/A482	l, m
/A4F0	d to i and t, u, v
/A4F4	d to i and t, u, v
/A520	d to i and t, u, v
/A525	d to i and v
/A526	d to i and v
/A532	v

/A580	a and t
/A590	t
/A596	a, t
/A5F3	a
/A602	a
/A603	a
/A620	a
/FFFF	a, d, e, f, g, t

APPENDIX A

COMMAND LIST SUMMARY

GENERAL

The commands summarised here include:

Operator Commands
Processor Call Commands
BCP Commands
LIB Commands
UPDATE Commands
EDF Commands
FCL Commands
Miscellaneous Commands

The command structure and syntax are summarised but no explanatory notes are included, since these have been given in the appropriate manual.

COMMAND LIST SUMMARY

1. Operator Commands

```
RY_<device address>
RD_<device address>
ON_<device address>
OF_<device address>
AS_<file code>, <device name> <device address>
SD_<day>, <month>, <year>
SC_[<hour> [,<minites>[,<seconds>]]]
      default value = 0
PS_<machid>, <prog name>                            (foreground only)
PS_                                                 (batch only)
RS_<machid>, <prog name>[ ,<contents of A7>]        (foreground only)
RS_                                                 (batch only)
AB_<machid>, <prog name>                            (foreground only)
AB_                                                 (batch only)
DM_<absolute address 1>, <absolute address 2>
CR_<file code>
WM_<location>, <value 1> [,<value i>]
PK_<disc unit file code> [,V|W]
DB_[<1st locat>[,<last locat>]]
SB
SM_<machid>
KI_<machid>, <prog name>, <spe char>, <mess>
KL_BATCH, <spe char>, <mess>
SP_<device name><device address>[,<function code>]
LD_<prog name>,<DAD filecode>[,<load address>]
```

2. Processors Call Commands (ASM, FRT, LKE, MAC, RTL)

General form:

```
<Processor name> [SIZE = <n pages.|MAX]
[,DUMP = ALL|PROG|NO]
```

The OPT(ions) command follows immediately after the call command:

```
ASM:
OPT|_|[PROG= <file code>|<prog name>]
[,VERS = <number>|0][,USID = <userid>]
[,DAD = <dad fc>][,LIST = YES|NO][,COND = YES|NO]

LKE:
OPT|_|[STAD = Name| last start address]
      [,CBLK = hexa value| last region address]
      [,CREF = YES|NO][,MAP = YES|NO]
      [,SLIB = YES|NO|NAME]
      [,ULIB =YES|NO|NAME]
      [,CATL = <name>]
      [,DBUG = ENTR|STAB| NONE][,GENE = LM|OB]
      [,KEEP = <ident. list>][,FRGT = <ident. list>]
      [,ONAM = <name>|NONAME][,DLST = <name>]
      [,INTC = <ident list>][,ROVP = <name>]
      [,CROV = <ident list>]

FRT:
OPT|_|[PROG = <file code>|<progr name>]
      [,VERS = <NUMBER>|0],USID = <USERID>[,DAD = <DAD F.C.>]]]
      [,OPTM = YES|NO][,RNTR = YES|NO]
      [,ACTV =YES|NO][,GNRC = YES|NO]
      [,WALL = YES|NO][,C = YES|NO]
      [,FPP =S|W][,LIST = YES|NO]
      [,X = YES|NO]
      [,D = YES|NO|FULL|BASIC|NONE]
      [,M = YES|NO]

MAC:
OPT|_|[IN|PRIMINFC=filecode][,OUT|OUTPUTFC=filecode][,SC|SCANCHAR
    ="|#|$|%][,ST|STATEMAR=<start column>],continuation col]]
    [,SO|SINGOPTS=I|M|N|O|P|S][,SF|SUMMARFC=<filecode>]
    [MF|MACROFLS=[fdlname=filecode][,fd 2 name=filecode]---]
```

LIB and UPD processor calls have no parameters, but commands given to these processors once loaded, are listed below:

BCP is loaded whenever the SB (start batch machine) command is given.

COMMAND LIST SUMMARY

```
RTL:OPT [LIST= YES|NO][,PARLIST=YES|NO][,MODE=SYST|APP]
        [,FORT=YES|NO][,SYMB=YES|NO][,RES= YES|NO]
                                         [TEST= YES|NO]
        [,CARDFMT=(n:m)| (1:72)][,XREF=<filecode>|<filename|/D8]
        [,OP1=option][,OP2=option].....[,OP5=option][,TA1=table spec]
        [,TA2=table spec].....[TA7=table spec][,STACK= 512]
        [,OBJLIST=YES|NO],PROG=<filecode>|<filename>|E1
        [,DAD=/Fx|<current dad>][,USID=<userid>|<current user>]
        [,VERS=<n>|0]
```

3. BCP Commands

These commands may be entered either with keyword or positional parameters, except the ASG command. Positional parameters are only allowed here to assign a device.
The underline value means that the default value is the value indicated in the job command.

```
 :JOB_[USID = <userid>][,DAD = <dad f.c.>]
[,ACNT = <acc>][,PASW = <password>][,VOLN = <volume number>]
                                                       [,DNAM = <DAD name>]
SKP_[PAGE = number of pages|1]
ERR_[FCOD = <recovery f.c.|01>]
INC_[FCOD = <f.c./E2>]
INC_LIBR = <lib name>[,MNAM = <module name>|ALL]
[,USID = <userid>][,DAD = <dad f.c.>]
SCR_[FCOD = <f.c.>
ASG_[FCOD = <f.c.>,DVCE = <dev name>[<dev addr>]
```

For a temporary file:

```
ASG[T]_FCOD = <f.c.>,DAD = <dad f.c.>[,TYPE = OB|SC|LM|UF][,NBGR =
<NO. of granules>|1]
[,CONS = YES|NO]
```

For a catalogued file:

```
ASG[F]_FCOD = <f.c.>[,DAD = <dad f.c.>]FNAM = <file name>[,TYPE =
OB|SC|LM|UF|EF]
[,USID = <userid>][,WPRO = YES| NO][,VERS = <version No.>|0]
```

Equivalence:

```
ASG[E]_FCOD = <new f.c.>,ECOD = <old f.c.>
```

For a DAD:

```
ASD_FCOD = <DAD f.c.>, DISK = <disk f.c.> ,DNAM = <DAD name>
ASD_FCOD = <DAD f.c.>, VOLN = <volume number> ,DNAM = <DAD name>
PSE_<message to be sent to the operator>
MES_<message to be sent to the operator>
NOD_<name>[rovs| * [,a]] rovs = core-resident overlay segment name
                            a = absolute load address
ROV_<name>,<address>
HLP_CMND = <command name][,FCOD = <print f.c.>]
RUN_PROG = <name>[,VERS = <version No.>|o]]
[,USID <userid>][,DAD = <dad f.c.>]
[,SIZE = <n pages>|MAX]
[,PNCH = <max No. of rec to be punched>|1000|NO]
[,PRNT = <max No. of lines to be printed>|1000|NO]
[,TIME = <execution time limit>|300|NO]
[,DUMP = ALL|PROG||NO[,A1=<value>][,A2=<value>]...[,FR1=<value>]...
```

COMMAND LIST SUMMARY

```
REQ_DVCE = <dev. name>, FCOD = <f.c.>,MESS = '<message>'
message sent to the operator for this command:
MOUNT_ON_<dev. name><dev.addr.><message>
   THEN RS, PLEASE

REL_FCOD = <f.c.>,MESS = '<message>'
message sent to the operator for this command:
DISMOUNT_<dev. name><dev.addr.><message>,PLEASE

:EOJ
:EOB
REW_FCOD = <f.c.>
ULD_FCOD = <f.c.>
FFS_FCOD = <f.c.>[,NUMB = <No. of tape mark>|ALL|1]
FBS_FCOD = <f.c.>[,NUMB = <No. of records>|1]
RFS_FCOD = <f.c.>[,NUMB = <No. of records>|1]
PLB_FCOD = <f.c.>
WEV_FCOD = <f.c.>
WLB_FCOD = <f.c.>[,SNUM = '<voll serial No.>']
[,SCOD = 'X'][OWNE = '<owner code>']

WES_FCOD = <f.c.>[,NUMB = <No. of EOS records>|1]
WEF_FCOD = <f.c.>[,NUMB = <No. of EOF records>|1]
REF_FCOD = <f.c.>

:STP_[CODE = <highest err. code>|0][,ABCD = <code>]
RBS_FCOD = <filecode>[,NUMB = <No. of records>]
ROI_FCOD = <filecode>[,MESS = <message text>]

DAS_LCOD = <line code>,DVCE = <dev. name><dev. addr.><line nbr.>
DDC_LCOD = <line code>
DDL_LCOD = <line code>
DHD_LCOD = <line code>
DHL_LCOD = <line code>
```

4. LIB Commands

The commands may be entered either with keyword or positional parameters. Keyword parameters are shown here, but if positional parameters are to be used they should be entered in the same order as the keyboard parameters but in the form:

```
    COM x,y,z
```

where COM is the command mnemonic, and x,y,z are parameter values. If a positional parameter is omitted, a comma must be substituted in its place.

Commands reserved for the SYSTEM user:

```
DCD_DNAM=<dad name>,NCYL=<No. of cylinders>,
NINT=<No. of interlaces>[,NSPT=<No. of sectors per track>]
NSPG=<No. of sectors per granule>[,SLNG=<No. of chars. per sector>]
DISK=<disc f.c.>
```

For X'1215', NSPT and SLNG are not used.

```
DLD_DNAM=<dad name>,DISK = <disc f.c.>
DCU_USID=<userid>,DAD = <dad f.c.>
[,PASW = <pass word>][,ACNT = <account>]
DLU_USID=<userid>,DAD = <dad f.c.>
SDM_DISK=<disc f.c.>,ONTO = <Mag tape f.c.>
[,CK = YES|NO]
```

To restore a disc from the tape, operator starts IPL from the tape and answers the message:

```
DISC PHYSICAL ADDRESS : <address>
WRITING THE VOLUME LABEL ? YES|NO
```

NO means VOLAB of the disc remains unaltered.
YES means VOLAB recorded on tape is written on disc.

```
CDD_DISK=<disc f.c.>
SDD_DISK=<disc f.c.>,ONTO = <disc f.c.>
```

Commands not reserved for System use:

The underlined value means that the default value is the value indicated in the JOB command.

```
PRV_DISK = <disc f.c.>
PRC_DAD = <dad f.c.>
SVU_IUSI = <userid>[,IDAD = <dad f.c.>][,OUSI = <userid>]
                                         [,ODAD = <dad f.c.>]
```

COMMAND LIST SUMMARY

```
PRF_FNAM = <f.c.>|
<file name>,TYPE = UF|SC[,USID=<userid>]
[,VERS = <vers No.>[,DAD=<dad f.c.>]]
[,PRNT = <f.c.>|02][,FROM = <f. line>]
[,TO = <l.line>]
DUF_FNAM = <f.c.>|
<file name>,TYPE = SC|UF|OB|LM
[,USID = <userid>[,VERS = <vers, No.>|0]
[,DAD = <f.c.>][,PRNT = <f.c.>|02]
[,FROM = <f.line>][,TO = <l.line.>]
PRD_[USID=<userid> [,DAD=<dad f.c.>]][,PRNT=<f.c.>|02]

POD_[LIBR=<filename>|USRLIB][,EXTN=YES|NO][,USID=<userid>
[,DAD=<dad f.c.>]][,PRNT=<f.c.>|02]

KPF_FCOD = <f.c.>[,TYPE = SC|LM|UF|OB]
[,FNAM = <file name>][,USID = <userid>]
```

default values:

TYPE = the file type with which it is assigned
FNAM = the source file <ident> is taken into account.

```
DLF_FNAM = <file name>[,TYPE = SC|LM|UF|OB]
[,USID = <userid>[,DAD = <dad f.c.>]]
[,VERS = <vers No.>|ALL|OLD|0]
```

ALL means all versions are deleted.
OLD means all versions except version 0 are deleted.

```
KOM_[LIBR=<lib name>|USRLIB][,MNAM=<module name>|ALL]
      [,USID=<userid>][,DAD=<dad f.c.>][,FCOD=<f.c.>|/D5]
DOB_[LIBR = <lib name>|USRLIB]
[,MNAM = <module name>|ALL]
COB_LIBR = <lib name>|USRLIB]
[,USID = <userid>[,DAD = < f.c.>]] (Default = JOB userid, JOB DAD
SMV_VERS = <max vers No.>(Default value is 0)
```

CDF commands:

- Output file is a temporary disc file and already assigned:
  ```
  CDF_INAM = <fc1>,ONAM=<fc2>
  ```

- Output filecode to be assigned to a temporary file by LIB:
  ```
  CDF_INAM = <filename>[,ITYP = SC|OB|UF|LM]
     [,IUSI=<userid>][,IVER=<ver. No.>|0]
     [,IDAD=<dad f.c.>]
     ,ONAM = <fc2>,ODAD = <dad fc>
     [,OTYP = SC|OB|UF|LM]
  ```

- Output file is a catalogued file:
  ```
  ,ONAM = <filename>[,OTYP = SC|OB|UF|LM]
  [,OUSI=<userid>][,OVER=<version No.>]
  [,ODAD=<dad f.c.>]
  ```

COMMAND LIST SUMMARY

CSF commands:

1. Output file is a temporary disc file already assigned:

```
   CSF_INAM = <fc 1>,ONAM = <fc 2>
```

2. Output file code assigned to a temporary file by LIB:

```
   CSF_INAM = <filename>[,ITYP = SC|OB|UF]
   [,IUSI=<userid>][,IVER=<vers No.>|0][IDAD=<fc 1>]
   ,ONAM = <fc2>[,OTYP = SC|OB|UF],ODAD = <dad fc>
```

3. Output file is a catalogued file:

```
   ,ONAM = <file name>[,OTYP = SC|OB|UF]
   [,OUSI = <userid>][,OVER=<vers>]0][,ODAD = <dad fc>]
```

```
LTO_LNAM = <f.c.>|/D6|<file name>[,LUSI = <userid>
[,LDAD = <f.c.>]][,LVER = <versnb>]|0]
,ONAM = <f.c.>|/03|
          <f.c.>,ODAD = <dad f.c.>
   [,IDEN = <ident>]
```

Set and reset flags command:

```
   SSH_FNAM = <file name>,TYPE = UF|OB|SC|LM
   [,VERS = <vers nb>|0]
```

   same parameters for:

```
   RSH SPF RPF SSF RSF SIF RIF
```

Magnetic/Cassette tape positioning commands: (see BCP commands page)

```
LEN
SRD_IVOL = <input f.c.>,OVOL = <output f.c.>,DAD = <dad name>
   [,NSPT = <No. of sectors per track>]][,NINT = <Interlace No.>]
   [,OPT = ALL|ONLY]
```

NSPT and NINT are not required for X1215/6 discs, and, if omitted for disc to disc transfers, the values of the source disc are taken. OPT specifies that either ALL or ONLY occupied granules are transferred.

- HLP command

```
   HLP_[CMND = <LIB command mnemonic | ALL>]
```

- REC command

   1) Filecode already assigned:

```
      REC_[LCOD = <linecode>,FNAM = <filecode>]
```

   2) Filecode to be assigned by LIB:

```
      REC_[LCOD = <linecode>,FNAM = <filecode>]
      [,TYPE = SC|=OB|UF|LM][,DAD = <dad fc>
```

```
   3) Input filecode already catalogued:
      REC_[LCOD = <linecode>,FNAM = <filecode>]
      [,TYPE = SC|=OB|UF|LM][,USID = <userid>][,VERS = <versn No.>]
      [,DAD = <dad fc>]
            default values are:
               DAD       =  JOB DAD
               Userid    =  JOB userid
               VERS      =  0
               TYPE      =  UF-
```

SEN command

```
   1) Filecode already assigned:
      SEN_[LCOD = <linecode>,FNAM = <filecode>]

   2) Filecode is a catalogued disc file:
      SEN_[LCOD = <linecode>,FNAM = <filename>[,TYPE= SC|OB|UF|LM]
      [,USID = <userid>][,VERS = <version No.>][,DAD = <dad fc>]

            default values are:
               DAD       =  JOB DAD
               Userid    =  JOB userid
               VERS      =  0
               TYPE      =  UF
```

5. Update Commands:

Definition Phase:

```
!!IN_FNAM = <file code>
            |<file name>[,TYPE = SC|UF]
            [,USID = <userid>[,DAD = <dad f.c.>]]
            [,VERS = <vers nb>|0]
```

or, with positional parameters:

```
!!IN_<file code>
      |<file name>[,SC|UF][,<userid>
      [,<dad f.c.>]][,<vers.No.>]

!!OU_FNAM = <file code>
            |<file code>[,DAD = <dad f.c.>
            [,TYPE =SC|UF]|
            <file name>[,TYPE =SC|UF]|
            [,USID = <userid>[,DAD = <dad ff.c.>]]
            [,VERS = <vers nb>|0]
            [,RSIZ = <output rec. size>]

!!OU <file code>|
      <file code>[,SC|UF],<dad f.c.>|
      <file name>[,SC|UF][,<dad f.c.>]
      [,<userid>]][,<vers.No.>][,<output rec size>]
```

!!IN is optional if the input file is: /D4, SC
!!OU is optional if the output file is: /D3, UF or /D4, SC.

```
Default values:
!!IN command:   userid = JOB userid
                DAD fc = JOB DAD
                TYPE   = SC                 VERS   = 0

!!OU command:   as for !!IN except:
                TYPE    = that specified or assumed by the !!IN command.
```

```
!!RS_STNG = 'charac.string', BY = 'charac.string'
      [,FROM = <line>|0]
      [,TO = <line>|last line of input file]
```

or:

```
!!RS_'repl.char.string', 'new char.string'[,<line>]
      [,<line>]
!!DS_STNG = 'char.string'[,FROM = <line>|0]
      [,TO = <line>]     (Default value = EOF)
```

or:

```
!!DS_'string to delete'[,<line>][,<line>]
      [,<line>]
!!DE_STNG = 'charac.string'[,FROM = <line>|0]
      [,TO = <line>]
```

```
or:

!!DE_'charac.string'[,<line>][,<line>]
!!IS_STNG = 'charac.string'[,FROM = <line>|0]
      [,TO = <line>]

or:

!!IS_'charac.string'[,<line>][,<line>]
!!CC_SPEC = <sp charac>
!!CC_<sp charac>
```

Execution Phase

```
!!RE_LINE = <line>,STNG = 'character string',
      BY = 'replacement char. string',

or:

!!RE_<line>,<character string>,<replacement character string>
!!DL_FROM = <line>[,TO = <line>|EOF]

or:

!!DL_<line>[,<line>|EOF]
!!IL_[AFTR = <line>]

or:

!!IL_<line>
!!JN_[AFTR = <line>],AUXI )= <file code>|
       <file name>[,TYPE = SC|UF][,USID = <userid>]
       [,VERS = <vers>|0][,DAD = <dad f.c.]
       [,FROM = <line>][,TO = <line>]

or:

!!JN_[<line>],<file code>|
       <file name>[SC|UF][,<userid>]
       [,<vers>][,<dad f.c.][,<line>]
       [,<line>]

!!EN  (normal termination)
!!EX  (immediate exit)

!!KF_[FNAM = <file name>][,TYPE = SC|UF
      [,USID = <userid>]

or:

!!KF_[<file name>][,SC|UF][,<userid>]
!!LF_FNAM = <file code>[,STNG = <char.string>][,FROM = <line 1>]
               [,TO = <line 2>]
```

or:

```
!!LF_FNAM = <file name>],TYPE = SC|UF]
               [,TO = <line 2>]
               [,USID = <userid>][,FDAD = dad f.c.>]
               [STNG = ’char.string’[,FROM = <line>]
               [,TO = <line>]
```

or:

```
!!LF_<file code>[,<char.string][,<line 1>][,line 2]
```

or:

```
!!LF_<file name>[,SC|UF][<userid>]
       [,dad f.c.>][,’char.string’][.<line 1>]
       [,line 2]

!!CI_[FCOD = <filecode>]           default = original source
```

or:

```
!!CI_[<filecode>]
!!HL_[CMND = <command mnemonic>]    default = ALL
```

or:

```
!!HL_[<command mnemonic>]
```

6. EDF Commands:

Disc space allocation and initialisation:

```
FILE_FNAM = <file name>,DAD = <dad f.c.>,
     USID = <usid>,NKEY = <# of index files>,
     NDAT = <# of data files>,MREC = <max # of records>,
   [,SECU = BU|FULL]
```

BU means back up only
FULL means back up and back out

Index file description:

One key command per index (see NKEY)

```
KEY_FNAM = <index file name>,DAD = <dad file code>,
    USID = <usid>,KLGT = <key length>,KLGT = <integer length>,
    KPAD = <padding char.>,KNAM = <key name>,
      KPOS = <key position>[,BIJ][,DISP]
      KLGT = Key length in characters
      KPOS = Displacement to first character of key field in data
             record (from 0)
      KPAD = /<hexad digit><hexa digit>
```

used for initialisation of the key field. The key filled by such character must be lower than any value of the key.
BIJ means that the key is bijective. At least one bijective key defined in the first key command.
DISP means that the key is the dispatching one.
The default value for FNAM is that used for KNAM.

Data file reservation:

One DAT command per data file (see NDAT)

```
DATA_FNAM = <data file name>,DAD = <dad f.c.>,
     USID = <usid>,NBGR = <# of granules>,
     [,CRIK = <max key value in file>]       (criterion value of file)
```

Syntax of CRIK:
CRIK = <number of bytes><mode><value> $$ etc ...

The values available for <mode> are:

```
A      ASCII characters      Note:  For the last DATA command, CRIK
D      decimal number               value is > highest key value in
H      hexadecimal number           the file.
Example:  CRIK = 2 A 57 $$ 3 H 1 B A 3 C 7 $$ 2 D 314
```

Reorganisation of an Index file:

```
IDRG_FNAM = <extended file name>,FDAD = <dad f.c. of descriptor>,
     KDAD = <dad f.c. of index file>,
     KNAM = <key name>,USID = <usid>,
     [FREE = <% of free space in index sector>|0]
```

Save and restore an extended file:

```
COPY_FNAM = <subfile name>,IDAD = <dad f.c.>,
     IUSI = <userid>,ODAD = <dad f.c.>,OUSI = <userid>
```

The sequence must be:

```
COPY_file description
COPY_index file 1
COPY_index file n
COPY_data file 1
COPY_data file n
```

COPY is only used for making the first COPY of an extended file. If the user wants to save his extended file again, at the same place on the disc, he must use SAVE. The current date and time is copied in the file descriptor.

```
SAVE_FNAM = <extended file name>,IDAD = <dad f.c.>,
     IUSI = <userid>,ODAD = <dad f.c.>,OUSI = <userid>
```

The restoration of such a file is made by using:

```
REST|_|FNAM = <extended file name>,IDAD = <dad f.c.>,
      IUSI = <userid>,ODAD = <dad f.c.>,OUSI = <userid>
```

The data available in the new file descriptor will be the date of the last saving of the file.

```
REPL_FNAM = <subfile name>,IDAD = <dad f.c.>,
     IUSI = <userid>,ODAD = <dad f.c.>,OUSI = <userid>
```

Same utilisation as COPY but the date written in the file description in output is not the current date but the date indicated in the file description in input.

Other commands

```
DEL_FNAM = <extended file name>,DAD = <dad f.c.>,
    USID = <userid>
```

used to delete an extended file, the file descriptor of which is still valid. Otherwise it is possible to delete all subfiles by the librarian command: DLF

```
DLF_FNAM = <subfile name>,DAD = <dad f.c.>,
    USID = <userid>
```

```
ABT    abort the EDF processor
EFEN   end of EDF (rejected if any activity exists)
```

```
DUMP_FNAM = <EDF filename>,USID = <userid of descrptr. file>
     ,DAD = <dad fc of descrptr file>,KNAM = <key name>
   [,FROM = <lowest key value|BCF>]
     [,TO = <highest key value|EOF]
   [,PRNT = <print fc>|/02]
```

```
STAT_FNAM = <EDF filename>,USID = <userid of descrptr. file>
     ,DAD = <dad fc of descrptr file>,PRNT = <additional print fc>]
   print fc is normally that of the printer.
SBUF_NBUF = <No. of buffers>
DKMT_FNAM = <filename>,IDAD = <dadfc>,IUSI = <userid>,OCOD = <fcod>
MTDK_FNAM = <filename>,ODAD = <dadfc>,OUSI = <userid>,ICOD = <fcod>
```

Loading and Unloading a file

```
LOAD_ONAM = <EDF filename>,ODAD = <dad fc descrptr. file>
     OUSI = <userid>,ICOD = <input fc>,
   TYPE = {CONT|CRIK|File No.|SAME}[,{N = <fixed blocking factor>
   |SEP = <separator>}][,FREE=<% free space kept in index sectors>]
     [,IGEN = YES|NO]

     'separator' is a 2ASCII character record blocking separator
     Default value for [{N = |SEP = } ] is N=1

UNLD_FNAM = <EDF filename>,DAD = <dad fc of descrptr. file>
     USID = <userid of descrptr file>,OCOD = <output filecode>
     [,DATA = <subfile No.>][,N = <fixed deblocking factor>|1]
```

COMMAND LIST SUMMARY

Back up commands:

```
BUGN_FNAM = <back-up file>,USID = <userid>,
   DAD = <dad f.c.>,NBGR = <nb of granules>
```

The userid must be the first userid of the dad if a back-up file with the same name already exists in the same dad, this command must be entered under USID = SYSTEM.

```
INSE_IDEN = 'run ident',BUFC = <back-up f.c.>[,BOFC = <back-out fc>]
            back-up f.c. and back-out f.c. are previously assigned
            in the system machine by the ASG FCL command.
```

```
RBUP_BUFC = <back-up f.c.>[,NRUN = <nb of runs to be redone>]
   [,BOFC = <back-out f.c. in BATCH machine>]
```

```
BOGN_FNAM = <back-out filename>,USID = <userid>]
   DAD = <dad fc for cataloguing>],NBGR = <No. of granules>]
```

```
SPRO_FNAM = <filename>,USID = <userid>,DAD = <dad fc for descriptor
            file>, SECU = NO|BU|FULL
```

7. FCL Commands

Machine declaration commands

```
DCF_<machid>[,No. of core resident segments 0]
DCB_[<size>][,level]
CMA_<No.of pages>[,<pub.lib.size>]
SEG_<n>,<no.of pages>
LAB_<No.of scheduled labels>
FCD_<file code>,<device name><device address>
FCD_<disc unit file code>
FCD_<dad file code>,<disc unit file code><dad name>
MFC_<max number of spare entries>
MBF_<max number of blocking buffers>
DEN - Definition End
```

for data communication :

```
DLN[<No. of line codes>|system generation value]
DLC<line code>,<dev. name>[dev address]
DLC_<new line code>,<old line code>
```

System commands

```
DAT_<day>,<month>,<year>
CLK_<hour>,<minutes>,<seconds>
KIM_<machid>|BATCH
DON_<dev. name> <dev. address>
DOF_<dev. name> <dev. address>
NDV_<dev. name> <dev. address>[,<int. level>]
[,<No. of lines per page>]
PON_<page nb>
POF_<page nb>
FON
FOF
WRD  <dad file code>,<sector Nb>,
<displacement>,<value 0>[,<value I>] ...
```

COMMAND LIST SUMMARY

User Commands

```
LOD_<segment No.>,<program name>,<dad file code>
     [,No. of scheduled labels>]
SWP_<progr name>,<dad f.c.>
[,No. of scheduled labels>]
RON_<program name>,<DADfc>[,<max.number of scheduled labels>]
     [,R,<max.number of activations>]
CNL_<progr.name>,<level>
DSL_<progr.name>
CNT_<progr.name>,<n tim>,<format No.>,<reactivation param>,
     [,<nc>|hh,mm,ss]
DST_<progr.name>[,<timer number>]
ACT_<progr.name>[,<contents of A3>]
     [,<cntents of A4>]
RUN_<progr.name>[,<contents of A3>]
     [,<cntents of A4>]
KIL_<progr.name>
KIF_<file code>,<file name>
     [,file type>|UF]
FLD_<dad f.c.>,<file name>
     [,file type>|UF][,<version No.>|0]
RAB_<progr.name>
REP_<Max.No.of activations>,<Seg.No.>,<program>,<dad fc>,[,<max.No.
                                   of scheduled labels>]
DEB_<progr.name>
DLP_<device name> <device address>[,<number of lines per page>]
```

The following are either user or system commands:

```
KIS_<segment number>
KLM_<secondary load module name>
LSM_<secondary load module name>,<DADfc>,<userid>[,{R | W}]
PCM_<print filecode>
```

The following are communication commands:

```
DAS_<linecode>,{<device name>[<device address>] | NO}
DAS_<new linecode>,<old linecode>
DDL_<linecode>
DDC_<linecode>
DHL_<linecode>
DHD_<linecode>
```

COMMAND LIST SUMMARY

User of System Commands

```
BYE_[<machid>][,<machid>] ... (one <machid> may be BATCH)
SCR_<file code>
ASG_<new file code>,<old file code>
ASG_<file code>,<dev. name>[<dev. addr>]
ASG_<new file code>,DDFX,<file type>
       [,<No. of granules>[,NC]]
ASG_<file code>,DDFX,<file type>,<file name>
ASG_<file code>,<disc unit file code>,<dad name>
DUF_<file code>,<dad file code>,<first addr>
       [,<last addr>]
DTO_<device name>[<device addr>], <timeout value>
CLS_[<machine 1>,><machine 2>, ......]
```

In the system machine:

```
WRM_<location>,<value 0>[,<value i>]
DUM_<first abs.addr>,<last abs.addr>]
```

In the user machine:

```
DUM_<seg.No.>,<first ver.addr>[,<last ver.addr>]
WRM_<seg.No.>,<location>,<value 0>[,<value i>] ...
TIM
PRG <program name>,<print filecode>,{A | M | S}
PFC [{<print filecode> | /01}]
PLV [{<print filecode> | /01}]
MAP [{<print filecode> | /01}]
PLC [{<print filecode> | /01}]
PRS <program name>[,{<print filecode> | /01}]
```

In the system machine: (for batch programs)

```
PSE
ABT
RST_[<contents of A7>]
```

In the user machine:

```
PSE_<progr. name>
RST_<progr. name>,<contents of A7>
ABT_<progr. name>
RYD_<dev. addr>
RDV_<dev. addr>
```

COMMAND LIST SUMMARY

Background machine

```
:EOB
:EOJ
ERR_FCOD = ][filecode of recovery device|/03>]
:JOB_[USID = <userid>][,DAD = <dad fc>][,ACNT = <account No.>
               [,PASW = <password>]
:STP_[CODE = <error code>|0][,ABCD = <error code>|/7F][NCOD = <new code>]
SCR_FCOD = <filecode>
FBS_FCOD = <filecode>[,NUMB = <No.of tapemarks>|1]
FFS_FCOD = <filecode>[,NUMB = <No.of tapemarks>|ALL|1]
PLB_FCOD = <filecode>
RBS_FCOD = <filecode>[,NUMB = <No.of records>|1]
REF_FCOD = <filecode>
REL_FCOD = <filecode>[,MESS = <'message'>
REQ_FCOD = <filecode>[,DVCE = <device mnemonic>,MESS = <'message'>
REW_FCOD = <filecode>
RFS_FCOD = <filecode>[,NUMB = <No.of records>]
ULD_FCOD = <filecode>
WEF_FCOD = <filecode>[,NUMB = <No.of EOF marks>]
WES_FCOD = <filecode>[,NUMB = <No.of EOS marks>]
WEV_FCOD = <filecode>
WLB_FCOD = <filecode>[,SNUM = <volume serial No.>]
                       [,SCOD = <security code>][,OWNE = <owner code>]
```

Background Interactive Command

```
MES_<message>
PSE_<message>
ROI_FCOD = <filecode>,MESS = <'message'>
```

Background Task Initiation Command

Standard Processor Call:

```
   XXX_[SIZE = MAX|<No. of pages>][,DUMP = ALL|PROG|NO]
```

Non-Standard Processor Call:

```
   XXX_[SIZE ={MAX|<No. of pages>}][,DUMP ={ALL | PROG | NO}]
         [,USID=<userid>][,DAD=<DAD filecode>]
```

where XXX is the three character processor mnemonic (ASM, FRT, etc)

```
RUN_[PROG = <program name>|/D6][,USID = <userid>|<:JOB Userid>]
      [,DAD = <dad fc>|<:JOB DAD fc>][,VERS = <version No.>|0]
      [,SIZE = <No.of pages>|0][DUMP = ALL|PROG|NO]
      [PNCH = <max.punch cards>|NO|1000][,PRNT = <max.print records>
                                                         |NO|1000]
      [,TIME = <max.No.of seconds>|NO|300][A1 = <value 1>,
                           <A2 = <value 2>,---A14 = <value 14>|0]
```

Miscellaneous Background Commands

INC commands:

a) disc device

```
INC_LIBR = <filename>,MNAM = <module name>|ALL[,USID = <userid>]
      [,DAD = <dad fc>]
```

a) non-disc device

```
INC_FCOD = <filecode>          default value = /E2
```

NOD command:

```
NOD_<node name>[,{<rov segment name>|*}[,<absolute address>]]
```

APPENDIX B

BATCH CATALOGUED PROCEDURES

Batch catalogued procedures are seen by the BCP as a set of BCP commands or data, as follows:

```
%% <procedure name>
...
...   commands/data
...
PEND
```

All the procedures of a <userid> must be kept in a catalogued file (FNAM = B:PROC, TYPE = UF).

Creation of a catalogued procedure

1. If B:PROC does not exist yet:

```
ASG FCOD = /20, DVCE = <input dev>
LIB
CSF INAM = /20, ONAM = /40, ODAD = Fx
KPF FCOD = /40, FNAM = B:PROC
LEN
```

2. B:PROC already exists:

```
UPD
!!IN  B:PROC, UF
!!IL
% % <procedure name>
...
...
PEND
!!KF  B:PROC, UF
```

FCL CATALOGUED PROCEDURES

FCL catalogued procedures are seen by the FCL as a set of commands and data as follows:

```
% % <procedure name>
...
...  commands/data
...
PEND
```

All the user procedures are kept in a file
(FNAM = F:PROC, TYPE=UF) catalogued in the first <usid> of the DAD /FOof the user machine.

System procedures are kept in a file (FNAM=S:PROC, TYPE=UF) of the DAD /F6 of the System Machine.

Creation of FCL catalogued procedures

1. If F:PROC does not exist yet:

```
   ASG FCOD = /20, DVCE = <input device type code>
   LIB
   CSF INAM = /20, ONAM = /40, ODAD = /FO
   KPF FCOD = /40, FNAM = F:PROC
   LEN
```

2. F:PROC already exists:

```
   UPD
   !!IN  F:PROC, UF
   !!IL
   % % <procedure name>
   ...
   ...
   PEND
   !!KF  F:PROC, UF
```

EXAMPLE OF A BATCH CATALOGUED PROCEDURE

```
000000    %%CATALS
000001    ASG FCOD=/30.DAD=@DAD./F2,CONS=NO,TYPE=SC
000002    LIB   DUMP=@DUMP.ALL
000003    CSF INAM=/E0,ONAM=/30
000004    KPF FCOD=/30,TYPE=SC,FNAM=@1
000005    PRF FNAM=/@1,TYPE=SC
000006    LEN
000007    SCR FCOD=/30
000008    PEND
000009    %%OLE
000010    ASM DUMP=@DUMP.ALL
000011    OPT PROG=OLEIDA,LIST=@LIST.NO
000012    INC LIBR=OBJLIB,DAD=/F2,USID=BATCH,MNAM=DEBI
000013    INC LIBR=OBJLIB,DAD=/F2,USID=BATCH,MNAM=R:EXAS
000014    NOD N1
000015    ASM DUMP=@DUMP.ALL
000016    OPT PROG=ECR000,LIST=@LIST.NO
000017    NOD N1
000018    ASM DUMP=@DUMP.ALL
000019    OPT PROG=ECR005,LIST=@LIST.NO
000020    NOD N2
000021    INC LIBR=OBJLIB,DAD=/F2,USID=BATCH,MNAM=BIDE
000022    NOD N2
000023    INC LIBR=OBJLIB,DAD=/F2,USID=BATCH,MNAM=MUST
000024    NOD N2
000025    INC LIBR=OBJLIB,DAD=/F2,USID=BATCH,MNAM=SUBLED
000026    NOD N1
000027    ASM DUMP=@DUMP.ALL
000028    OPT PROG=ECR020,LIST=@LIST.NO
000029    NOD N3
000030    INC LIBR=OBJLIB,DAD=/F2,USID=BATCH,MNAM=BIDE
000031    NOD N3
000032    INC LIBR=OBJLIB,DAD=/F2,USID=BATCH,MNAM=MUST
000033    NOD N1
000034    ASM DUMP=@DUMP.ALL
000035    OPT PROG=ECR180,LIST=@LIST.NO
000036    LKE   DUMP=@DUMP.ALL
000037    OPT MAP=YES,CREF=YES,DBUG=ENTR,SLIB=NO,CATL=OLEIDA
000038          :STP CODE=/10
000039    ASG FCOD=/05,DVCE=CR06
000040    ASG FCOD=/3C,DAD=/FB,TYPE=UF,CONS=NO
000041    RUN PROG=OLEIDA,DUMP=@DUMP.ALL
000042    LIB   DUMP=@DUMP.ALL
000043    PRF FNAM=/3C
000044    LEN
000045    PEND
```

Parameters

Catalogued procedures contain a variety of parameters:

Key word parameter without default value: DAD = @DAD
Key word parameter with default value: DAD = @DAD./F2
Positional parameter without default value: FNAM = @3
Positional parameter with default value: FNAM = @3.B:PROC
Key word parameter with condition: DAD = @DAD?
Positional parameter with condition: FNAM = @3?

The reading of data inserted in the catalogued procedure

If, for example, PROG1 reads a data file from /AO, it is possible to insert this data file in the catalogued procedure; the input file code (/AO in this example) will have to be assigned by equivalence with /EE in the catalogued procuedure.

```
%% EXAMPL
...  Other commands
...
ASG FCOD = /AO, ECOD = /EE
RUN PROG1, USID = @1.QUALIF, DAD = @2./F2
...
...
...    Data file read from /AO by PROG1
...
<special EOF>      : EOF May not be present
...
...  Other commands
...
PEND
```

APPENDIX C

FILE CODES USED

1. In the System Machine

The assignation of these file codes is done in the module INFCT (depending on system generation).

/01	error messages
/02	listing output
/Cx	physical discs
/E0	FCL input
/EF	operator commands; BCP commands PSE, MES; LKM 6
/F0	System DAD
/F1	D:CI DAD
/FF	D:MSEG DAD for CDC discs
/21	D:ERLG file for error logging (must be assigned by the user)

2. By Processors

General

/E0	control command input
/02	listing output
/01	error messages and recovery
/E1	ASCII data input
/E2	binary data input
/03	binary output

from /D0 up to /DF disc temporary files

/D0	input file used by INC command
/D4	source file
/D5	object module file
/D6	load module file

LIST OF FILE CODES USED BY PROCESSORS

ASM

/E0	for reading of OPT command
/02	listing output
/01	operator communication

input data

/E1 or /D4 or a file code specified in OPT command
output data
/D5

FRT

Same file codes as ASM

LKE

/E0	for reading of OPT command
/02	listing output
/01	operator communication
/D5	input data (object modules)
/D6	output data (load modules)
/D7	/D8, /D9, /DA work files

UPD

/E0	for command input
/02	listing output
/01	or file code shown in ERR command for error recovery.

Input files:	all available file codes declared in !!IN command or /D0
Ounput files:	all available file codes declared in !!OU command or /D4 if TYPE = SC and /D3 if TYPE = UF

/D0	working input file code
/D1	working output file code
/D7	working auxiliary file code

LIB

/E0	for command input
/E1	ASCII data input
/E2	binary data input
/02	listing output
/01	for file code declared in ERR command for error recovery operator communication
/D4	source file
/D5	object file
/D6	load module file

/D0 to /D3 temporary work files
/F0 to /FF DAD file codes
/C0 to /CF disc physical units

LIST OF FILE CODES USED BY PROCESSORS

RTL

/D4	RTL source input
/D7	work file
/D8	cross reference information
/D9	assembler source output
/E0	for command input
/02	for listing output
/01	for operator communication

3. By the BCP

/E0	reading of commands
/02	writing of commands
/01	operator communication

File code indicated in ERR command.
The peripheral corresponding to these file codes are declared in the FCD commands

4. By the FCL Processor

/E0	reading of commands
/02	writing of commands
/01	operator communication

5. By catalogued procedure processing

/EC
/ED
/EE

APPENDIX D

SYSTEM ERROR CODES

The following message is output:

```
FATAL ERROR ...SYSERR CODE = <error code>
```

The error code is also contain in Reg. Al and can have one of the following values:

0	Unrecognised Interrupt
1	Too many Scheduled Labels
2	Power Failure
3	Not enough free pages and no disc resident program is running, so remaining pages not enough to run any more disc-resident programs.
4	a) Error in DAD search b) Bad activation of a swap program (at system level)
5	Not used
6	Not used
7	Stack overflow
8	Not used
9	Bad calling sequence in system dynamic area buffer request
/A	Not used
/B	a) I/O Error in loading a segment b) Unpremarked CDC disc loaded during user application
/F	DWT not found
/12	Ft. pt. interrupt from system program interrupt routine
	Trap interrupt from system program or interrupt routine for
format 0	instructions (P854).
/13	System program aborted
/14	Page fault from system program or interrupt routine

/15	Trap interrupt from system program or interrupt routine
/16	Unknown device : device in DWT chain unknown (i.e. interrupt unrecognised).
/17	Swap queue overflow
/18	Element requested by LKM 30 (queue handling) not in queue.
/1A	Inconsistent Index Sector (TDFM)
/1B	Inconsistent BITTAB
/1C	Unknown block type in PCTMOV queue
/20	Trap interrupt for format 1 instructions in system program or interrupt routine (P854).

2. Floating point Status Codes

These are returned in A7 following an LKM request:

/24	Division by zero
/27	Overflow in IFIX function

3. I/O Error Code

These are returned in location 8 (ECBST) of the ECB, following an LKM 1 request:

a) Zero : The operation terminated satisfactorily.

b) Positive : The operation was completed but the following conditions were encountered:-

1	EOF encountered (Read)
2	EOS encountered (Read)
4	Data Error
8	Incorrect Length
/10	End of tape, end of media, request done
/20	Beginning of tape
/40	End of tape reached, but the current record has been read or written (warning signal)
/80	EOV mark detected.

c) Negative (bit 0 set)

Bit 1 = 0:- Bits 2-15 indicate the hardware status

Bit 1 = 1:-

/C001	Illegal File Code or File Code not assigned.
/C002	Device attached to other programs.
/C008	Buffer address, or requested length invalid.
/C010	Function unkown or incompatible with the Device or File.
/C020	Write protection on Disc File.
/C040	End of media: current operation aborted.
/C080	Time-out.
/C100	Disc queue overflow.
/C200	Dynamic Buffer overflow; no disc blocking buffer free.
/C400	Blocking overflow (No free granule).
/C800	Sector address out of DAD (GRANTB overwritten)
/E000	PFAR

4. Hardware Error

X1215/6 Discs

STATUS SET CONDITIONS

- Bit 15 is set if it is attempted to execute an I/O program on a non-operational drive.
- Bit 14 is set if the channel does not answer an exchange request in 12.8 microsec. during a read or write operation.
- Bit 13 is set if a check word error is detected.
- Bit 12 is set if the end of a sector is found before the end of the exchange when writing or reading.
- Bit 11 is set if:

  A channel-out (bit 1 of the 1st control word = 0) is executed after a read command;
  A channel-in (bit 1 of the 1st control word = 1) is executed after a write command;
- Bit 9 gives the drive number concerned.
- Bit 8 is zero if the information concerns the cartridge 0, one for the fixed disc.
- Bit 7 : reserved.
- Bit 6 is set if the seek operation is finished but was impossible to execute correctly (cf. seek command), or if the drive becomes inoperable during a seek operation.
- Bit 5 is set if the seek operation is correctly executed.
- Bits 4, 3, 2 : reserved.
- Bit 1 is set when a drive becomes operable.

CDC Discs

After an accepted SST (status request), the bits of R3 have the following status:

Bit	15	Not operable drive
	14	Throughput error
	13	Parity error
	12	Incorrect length
	11	Nul
	10	Nul
	09	Drive number
	08	Nul
	07	Nul
	06	Seek error
	05	Nul
	04	Record not found
	03	Nul
	02	Flag bad track
	01	Drive ready after unready
	00	Nul

REMARKS

- Bit 15 is set if a CIO is attempted on a not operable drive.

- Bit 14 is set if the CU is not able to access the memory within 100 microsec. during a write, read or verify operation.

- Bit 13 is set if a word comparison is wrong, or if the rest of the CRC accumulaton is different from zero.

- Bit 12 is set if the specified length is different from the real length, or if the read or write is not finished before the end of the track.

- Bit 09 is zero if the information concerns the drive zero, and one if the information concerns drive one.

- Bit 06 is set if, during a Seek operation, the drive is not able to access the addressed cylinder, or if the cylinder number comparison is wrong.

- Bit 04 is set during a Write or Read of Verify operation if the CU does not find the addressed record.

- Bit 02 is set if the Home Address of the track where the record has been read has a flag bit set.

- Bit 01 is set when a drive becomes operable.

ERROR CODES

Floppy 250K

Bit	0	nul
	1	drive ready after not ready
	2	key not found
	3	nul
	4	deleted data address mark
	5	record not found
	6	write protected
	7	nul
	8	drive number
	9	drive number
	10	retry
	11	program error
	12	incorrect length
	13	data fault
	14	nul
	15	not operable

Floppy 1M

Bit	0	nul
	1	nul
	2	nul
	3	nul
	4	deleted data address mark found
	5	sector not found
	6	seek error
	7	write protected
	8	nul
	9	nul
	10	retry procedure necessary to read an identifier
	11	program error
	12	full track processed
	13	data fault
	14	throughput error
	15	drive not operable

UPL disks

Bit	0	nul
	1	nul
	2	data part time out
	3	write protection or fault
	4	disk has become ready
	5	nul
	6	seek error
	7	no identifier detection
	8	no correct sector number
	9	identifier error
	10	flag error
	11	no correct head or track set number
	12	nul
	13	read data error
	14	throughput error
	15	drive not operable

ERROR CODES

Mag Tape Status Codes

During an accepted SST the BIO lines have the following meaning:

- 15	Devices inoperable (formatter or tape unit)
- 14	Throughput error
- 13	Data fault
- 12	Incorrect length
- 11	Program error
- 10	End of tape
- 9)	Tape number decoding
- 8)	
- 7	No identification burst
- 6	Write unable
- 5	Load point
- 4	No data
- 3	FM detected
- 2	Rewinding
- 1	Was not ready
- 0	Reserved

Tape No. decoding:

8	9	Tape No.
0	0	0
0	1	1
1	0	2
1	1	3

Inoperable devices

Bit 15 is set if the tape unit is not ready to receive a command except when the tape unit is engaged in a rewind operation in "on line" condition (in this case the bit 2 is set).

- Inter locks are not made
- Initial load is not complete
- The transport is not on line
- The formatter is power-off.

Throughput error

This bit (14) is set during a read or write command if the multiplex channel does not answer an exchange request coming from the CU in the allowed time.
The data exchange is stopped.

Data fault

This bit (13) is set during a read or write command or search command if a read error has been detected by the formatter for one or more of the following reasons:

In NRZ1 mode

- Vertical parity error on data character
- Longitudinal parity error
- CRC parity error (detected during any execution of a read forward command only).

In PE mode

- False preamble detection
- False postamble detection
- Buffer overflow
- Multichannel dropout
- Parity error without associated channel dropout.

Incorrect length

This bit (12) is set during a read command wherever the tape block length is different from the channel block length.

Program error

This bit (11) is set wherever the control unit receives:

- from the Multiplex channel an invalid code
  - * Output exchange from the channel with the command read.
  - * Input exchange from the channel with the command write.

  A program error stops any data exchange.

- from the central processor a CIO start with an invalid command on the BIO lines.

End of tape (Bit 10)

The end of the tape area has been sensed during the command performed. It is not stored in the CU for the next command forward.

Tape Unit Address

Bits 8 and 9 give the tape unit concerned by the status word.

Write Unable

This bit (6) is set wherever the control unit receives a write, write file mark, or erase gap command, while the tape unit is file protected. A write unable declaration stops any data exchange.

ERROR CODES

Load Point (Bits)

The selected tape unit is at load point.

No data

This bit (4) is set if any data block has not been found within 20 seconds after having sent the command to the transport.

File mark (Bit 3)

This bit is set if a file mark has been sensed during a read, a space block or search FM command.

Rewinding

This bit (2) is set if the tape unit selected is engaged in a rewind operation.

Was not ready

This bit (1) is set when the status of a tape unit has changed from not ready to ready.

- After a complete rewind operation.
- After a not operable status when a tape unit becomes operable.

No identification burst

This bit (08) is set if the identification burst is not found with the Phase Encoded Transport during a forward command from the load point.

Cassette Tape Status Codes

Following a successful SST or TST request status bits are set as follows:

	After an SST	After a TST
bit No.15	Not operable	CU busy
14	Throughput error	0
13	Parity error	0
12	Incorrect length	0
11	Program error	0
10	End of tape	0
09	Device number	0
08	Device number	0
07	1=A side; 0=B side	0
06	Write unable	0
05	Beginning of tape	0
04	No data	0
03	Tape mark	0
02	0	0
01	Has been inoperable	0
00	0	0

Bit 15

Following an SST request, this bit is set when a command cannot be executed because the drive is inoperable.

This may be due to:

- cassette not inserted,
- no power,
- cassette not loaded.

Following a TST request, this bit is set if the CU is not in the Inactive state.

Throughput error (Bit 14)

This bit is set if the CU receives no answer to an exchange request within the allowed time.
When a throughput error occurs:

- when reading, the data exchange requests are inhibited, the tape is stopped in the Interblock Gap and the CU switches to the Wait State.

- when writing, the data exchange requests are inhibited, the CU creates the Interblock Gap, stops the tape and switches into the Wait State.

ERROR CODES

Parity error (Bit 13)

This bit is set when a CRC check fails at the end of a Read or Write command.

Incorrect length (Bit 12)

This bit is set during a Read command whenever the physical data block length is different from the channel length.

Program error (Bit 11)

This bit is set when the CU receives an INR instead of an OTR, or an OTR instead of an INR, or when it receives an invalid CIO start.
A program error stops any data exchange.

End of tape (Bit 10)

This bit is set when the End of tape hole is passed in the forward direction and as long as it is not passed in the reverse direction.

Device number (Bits 8 & 9)

	bits	08	09
Drive	0	0	0
-	1	0	1
-	2	1	0

A or B side (Bit 7)

This bit is set when the A side of the cassette is up, and reset for the B side.

Write unable (Bit 6)

This bit is set when a Write or Erase command has been attempted on a write protected track.

Beginning of tape (Bit 5)

This bit is set when the beginning of tape hole is passed in a backward operation.
This bit is also set if the device number and A side bits indicate the end of a tape rewind.

No data (Bit 4)

When the tape is read and no data is encountered for 400 mm, this bit is set.

Tape mark (Bit 3)

This bit is set when a control block of 2 characters (Preamble - 2 characters - postamble) is encountered during Read, Space or Search commands.

Was inoperable (Bit 1)

This bit is set during a scanning operation when a drive state has changed from inoperable to operable. (After a power on and each time a cassette is inserted.)

Possible combinations of status bits

Bit No.	Bit values							
15	0	0	X	X	0	1	0	1
14	0	0	X	X	0	0	0	0
13	0	0	X	X	0	0	0	0
12	0	0	X	0	0	0	0	0
11	0	0	X	X	0	0	0	0
10	X	X	X	X	0	0	0	X
09)	Drive number							
08)								
07	X	X	X	X	X	X	X	0
06	0	0	0	X	0	0	0	0
05	0	X	X	X	1	0	1	X
04	0	0	X	0	0	0	0	0
03	0	X	X	0	0	0	0	0
01	0	0	0	0	0	0	1	0
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)

X : At least one of the bits may be different from zero.

(1) End of Write, Write Tape mark, Erase correctly performed.
(2) End of Read, Backspace, Search correctly performed.
(3) End of Read, Backspace, Seearch not correctly performed.
(4) End of Write, Write Tape mark, Erase not correctly performed.
(5) End of Rewind operation correctly performed.
(6) End of Rewind not correctly performed.
(7) The drive becomes operable.
(8) End of Unlock command correctly performed.

ERROR CODES

Program Abort Codes

/01	No scheduled label save area for scheduled label.
/02	Invalid instruction
/03	Memory protect violation
/04	Dynamic area destroyed
/05	Too many schedules labels
/06	Abort by operator
/07	Too many blocking buffers requested
/08	Disc overflow
/09	Not used - Disc queue overflow
/0A	Memory overflow during program load
/0B	Time limit expired (Batch only)
/0C	Print limit exceeded (Batch only)
/0D	Punch limit exceeded (Batch only)
/0E	Floating point error
/0F	Aborted by LKM 46
/10	:JOB and/or :EOJ cards read by batch program
/11	Error in loading program root
/12	I/O error in loading overlay segment
/13	Fatal error in spooling the I/O request
/14	Debug error
/15	Debug fatal error exit.

ERROR CODES

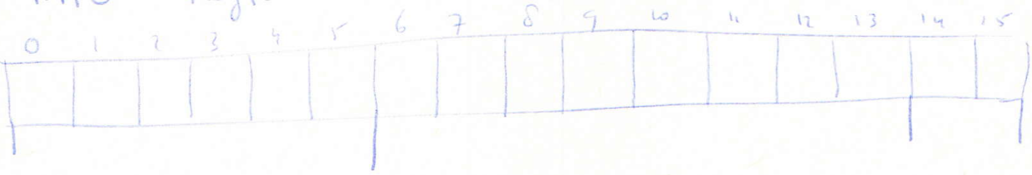
Program Status Word (PSW)

```
bits   0                 5   6   7   8   9   10  11  12  13  14   15
      ----------------------------------------------------------------
      |                  |   |   |   |   |   |   |   |   |   |   |   |
      ----------------------------------------------------------------
                             |___|   |________________________________|
                                                   v
          priority level          CR            GF register
             register
```

Priority level register   bits 0-5   Priority of the current program
Condition register   bits 6-7
General function register   bits 8-15:

```
   bit 8    CPU Stop/Run state
       9    Masked Interrupt (0) or Unmasked Interrupt (INH,ENB)
      10    Control Panel Interrupt
      11    Power Failure
      12    Real-time Clock (Control Panel Key set to RTC/ON)
      1~~3~~4   LKM Stack Overflow, Illegal Code
      1~~4~~3   Extended System Mode
      15    User (1) or System Mode (0)
```

MMU Register



0 - 5 least significant bits of physical page address
14 - 15 most significant bits of physical page address
6 page connected(0)/ not connected (1)
8 ? page modified

APPENDIX E

LKM INFORMATION

SUMMARY LIST

LKM Instruction	Function
1	I/O Request
2	Await event
3	Exit
4	Get Dynamic Buffer
5	Release Buffer
6	Pause
7	Retain Control on Abort
8	Datem II request
9	Reserved function
10	Connect program to Timer or Clock
11	Disconnect program from Timer or Clock
12	Switch program levels
13	Unknown function
14	Attach a Device or File
15	Detach a Device or File
16	Not Known
17	Get Time
18	Set Event
19	Reserved
20	Connect Program to a Level
21	Disconnect Program from a Level
22	Wait for a given time
23	Assign a File Code
24	Delete a File Code
25	Read Unsolicited Key-In
26	Cancel Request to Read Unsolicited Key-In
27	Load an Overlay Segment
28	Set Timer

SUMMARY LIST

LKM Instruction	Function
29	Reset Timer
30	Queue Handling Requests
31	Cancel Retain Control on Abort
32	Set and Reset File Attributes
33	Check and Assign a File Code
34	Check and Write EOF on a File
35	Get Program′s Characteristics
36	Begin Job
37	Retain Control on Floating Point Error
38	Cancel Retain Control on Floating Point Error
39	Get Machine Options
40	Keep File
41	Delete File from Library Directory
42	Begin a BCP Command
43	Allocate Permanent Granules
44	Release Permanent Granules
45	Dump Memory
46	Abort the Program
47	Call User-written LKMs
48	Assign a Linecode
49	Delete a Linecode
50	Internal I/O Request
51	Initiate Spooling for a BCP command
52	Send/Receive a Letter
53	Conditional Dump
54	Request or Release a Device
55	Semaphore
56	Page Control
57	Connect Secondary Load Module
58	Wait Multiple
59	Reserved
60	Load/Delete Secondary Load Module
61	Reserved
62	Check if DAD Filecode is Assigned
63	Set Date and Clock
64	Short timer for AMSNET
65	Access to AMSNET
66	LKM driver for AMSNET
67	Not known
68	Not known
69	Not known
70	Interface FCL and middleground processor
71	Assign DAD
72	Not known
73	Not known
74	Not known

SUMMARY LIST

LKM 1 - I/O Requests

Purpose

To initiate an action on, or retrieve information about, a device or file:

Calling Sequence

```
LDK  A7,L
LDKL A8,M
LKM
DATA [-]1
[DATA N]
```

Where L	=	Request Order Code (see below)
M	=	Event Control Block (ECB) Address
N	=	Scheduled Label Address

Request Order Codes -

bit 6 = 1	Time out period is specified in location ECBHD of the ECB (for teletypes etc).
bit 6 = 0	Default value of time-out defined at Sysgen.
bit 8 = 1	Implicit Wait: The requesting program will be put into a wait state until the operation is terminated.
bit 8 = 0	No implicit Wait: Control is returned to the calling program as soon as the request is recorded.
bit 9 = 1	User Error Action: The requesting program will process all abnormal or error conditions. The hardware status is returned in this case.
bit 9 = 0	System Error Action: The system performs the standard error actions and returns an error status to the calling program.

bits 10-15 of A7 are used to define the function required -

/00	Get device/file description
/01	Basic Read
/02	Standard Read
/05	Basic Write
/06	Standard Write
/0A	Direct Read (Disc File)
/0B	Direct Write (Disc File)
/11	Direct Read (DAD or Disc Unit)
/15	Direct Write (DAD or Disc Unit)
/10	Replace a bad track
/12	Replace a bad track (CDC disc)
/13	Seek to track zero (CDC disc)
/14	Write home address and premark the track (CDC disc)

/30	Get information about a File Code
/22	Write EOF Mark
/26	Write EOS Mark

For Magnetic Tape Cassettes the following extra codes are available:

/16	Skip forward to EOF Mark
/24	Write EOV
/31	Rewind (DFM file also)
/32	Fast search forward to tape mark (TC only)
/33	Skip 1 block backwards
/34	Skip 1 block forwards (MT only)
/35	Fast search backward to tape mark (TC only)
/36	Skip backward to EOF mark
/37	Lock (cassette)
/38	Unlock

For Flexible Disc the following request order codes are available:

/11	Read Sector
/15	Write Sector
/2D	Door Lock
/2E	Door Unlock
/2F	Write Deleted Data Address Mark
/3A	Compound Read
/3B	Compound Write
/3C	Search Key with Mask
/3D	Write Deleted Data Address Mark and Verify
/3E	Search Key
/3F	Write Sector and Verify

For all these functions MAS is compatible with present systems, the DFM/EDFM functions remain unchanged. DADs can be accessed exactly like the file codes /FO-/FF of the present systems, i.e. they can be read by any user program in the batch machine, but written only by the BCP, EDF and Librarian processors. Foreground users can access their DADs directly.

The physical disc units (file codes /CO-/CF) are not accessible by any program but the Librarian processors. The orders /11 and /15 are still used to read and write a sector of the disc.

SUMMARY LIST

ECB for LKM 1

```
          -----------------------------------------------------------
ECBFC    | EV |  L  |                                | file code      |
         |----------------------------------------------------------|
ECBBF    |               buffer address                              |
         |----------------------------------------------------------|
ECBRC    |               requested length                            |
         |----------------------------------------------------------|
ECBEL    |               effective length                            |
         |----------------------------------------------------------|
ECBST    |                   status                                  |
          -----------------------------------------------------------
```

on typewriter access

```
          -----------------------------------------------------------
ECBSC    |               Tabulation address                          |
         |----------------------------------------------------------|
ECBHD    |                                    | Time out values      |
         |                                    | in minutes           |
          -----------------------------------------------------------
```

on DFM or DAD access

```
          -----------------------------------------------------------
ECBSC    |               Relative sector number                      |
         |----------------------------------------------------------|
ECBHD    |                   not used                                |
          -----------------------------------------------------------
```

on Disc access

```
          -----------------------------------------------------------
ECBSC    |               Cylinder  number                            |
         |----------------------------------------------------------|
ECBHD    |         Head number                | Sector number        |
          -----------------------------------------------------------
```

On access to Floppy Disc the layout is as follows:

0	E	File code
2	Buffer address (including delete pattern (if order /3D is used)	
4	Requested length 1-128, for orders /3C and /3E 1-512 (even No.) for orders /11, /15, /3D and /3F 1-3328 (even No.) for orders /3A and /3B	
6	Effective length	
8	Status word	
/A	Absolute sector No. of the first sector to be read/written (except for orders /3C and /3E)	
/C	Timeout Value (bit 6 of A7=0)	

Explanation of Table

ECBFC (Byte 0) : E and L (bits 0 and 1) refer to event handling (See 'Wait on Event' request, LKM 2).

(Byte 1) : Bytes 8-15 contain the file code.

ECBBF (Bytes 2 and 3) : Start Address of the record buffer area.

Note : For Floppy disc the following apply:

- For request code /3D (delete), the buffer should contain the delete pattern.
- For request codes /3C and /3E, the buffer should contain the search key block.

ECBSC (Bytes /A and /B) : Absolute sector number except in the case of order codes /3C and /3E, in which case this location is left blank by the user and MAS returns the sector number following a successful search.

P800 MAS Vol IV: Trouble Shooting Guide

12NC: 5122 991 28475

Name ______________________________ Date ________

Company ______________________________________

Address ______________________________________

____

Telephone Number ____________________ Extension ________

Comments or Suggestions: