APPENDICES

To standardize the system generation procedure for all systems, a set of system generation processors has been developed which provides great flexibility, extensive logging of the process and improved efficiency.
The three steps inherent in any system generation process, i.e. monitor tables generation, monitor body generation and system medium generation are handled by a number of generation processors which are loaded and started successively.

For the generation of a Disc Real Time Monitor these are:

- GENMON, a generation monitor used only during the system generation process to run the generation processors.
- DRMGEN, which generates the monitor tables from the answers it receives from the user in a conversational process with a standard list of questions.
- PREMDK, which is used to premark the system disc and write an Initial Program Loader on it, as the first module on the system disc. PREMDK runs under any monitor.
- GENLKE, which runs under GENMON and scans the library of system modules (DRTMLIB), to select the ones requested during the DRMGEN phase and link them with the tables generated during DRMGEN.
- DISLOD, which, running under GENMON, is used to record the system processors on disc in load module format.

Depending on his configuration, the user may receive his sysgen tools, i.e. the above-mentioned processors, on punched tape or on cassette. In the first case, each processor is contained on a separate punched tape, in the second case all the processors are contained on one cassette.
In the description which follows in the paragraphs below, the cassette case is the basis, as this is the assumed standard for this sysgen process. It is very easy for the user, however, to redefine these standards (under GENMON) in case he works with punched tape. Apart from the redefinition of the standards, the main difference in the description is that from cassette the successive processors can be loaded and started without any manual

operation, whereas with punched tape, for each following step a new tape with the following sysgen processor must be put on the tape reader and then loaded and started.

Note: For DataCom SysGen, see P800M Data Communication User Manual.

In the following paragraphs, the whole set of operations necessary for the generation of a DRTM is described in a number of sections corresponding to the system generation processors listed above. At the end of each section a number of notes and rema remarks is given, which the user must carefully read before starting the operation.

OPERATION

The minimum configuration required for generating a DRTM is:
- CPU with 16k memory
- typewriter with address 10
- paper tape reader and punch, or
  two magnetic tape cassette drives on 1 control unit
- one X1215 disc unit

If the configuration is paper tape-oriented, the user receives 18 tapes, containing:
- IPL + GENMON
- DRMGEN
- PREMDK
- GENLKE
- DRMLIB 1
- DRMLIB 2
- DISLOD
- one tape for each of the monitor segments:
  D:USV1, D:USV2, D:USV3, D:USV4, D:OCOM, D:DUMP, D:ROOT,
  D:SEG1, D:SEG2, D:SEG3, D:SEG4.

If the configuration is cassette tape-oriented, the user receives two so called generation cassettes (G1 and G2), containing:

- cassette G1: side A: IPL
  GENMON
  DRMGEN
  PREMDK
  GENLKE
  side B: DRTMLIB 1
- cassette G2: side A: DRTMLIB 2
  DISLOD
  side B: monitor segments:
  D:USV1
  D:USV2
  D:USV3
  D:USV4
  D:OCOM
  D:DUMP
  D:ROOT
  D:SEG1
  D:SEG2
  D:SEG3
  D:SEG4

The user needs two cassettes or paper tapes of his own, to be used for intermediate storage of sysgen output.

Note:
Throughout this chapter, user replies typed in response to questions output by one of the sysgen processors, are underlined.

To start the process:
- switch on the CPU
- for cassette:
  - load generation cassette 1 (hereafter called cassette G1) in cassette drive TK05 with side A up
  - set the data switches on the CPU control panel to allow the bootstrap to load from TK05:

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

(hexa 0785)

(for the significance of the bits, see page 1-64 ;
suffice it to mention here that bit 3 is 0 if the cassette drives are connected to the I/O processor and 1 if they are connected to the programmed channel, and bits 10 to 15 contain the device address.)

- for paper tape:
  - put the tape containing IPL ▸GENMON on the tape reader and make it operable
  - switch on the paper tape punch and feed tape
  - set the data switches on the CPU control panel to allow the bootstrap to load from the tape reader:

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

(hexa 1020)

(for the significance of the bits, see page 1-64 ;
suffice it to mention here that bit 3 is 1 because the reader is connected to Programmed channel and that bits 10 to 15 contain the device address which is here assumed to be 20).

Then:
- press the MC button
- press the IPL button

Now the bootstrap is loaded which loads the first sysgen program from the cassette in TK05 into memory:

GENMON

GENMON is a special monitor used only during the sysgen process. To be able to do this, it must know the system configuration, the device addresses, interrupt levels and file codes. It is here that this system generation procedure shows its flexibility, for the great number of definition possibilitieswhich the user has at this point. GENMON outputs two questions, allowing the user to give his definitions and assignments. However, the system generation will handle a set of

built-in standards if these are acceptable to the user. In this case he does not have to define anything, but if one of the user's assignments or definitions is different from the standard ones as listed below, he must redefine under GENMON.

When GENMON is loaded its identification is output on the typewriter:

GENMON

When loading is terminated,

:EOS

:EOF

is output on the typewriter, followed by the question

STANDARD CONFIGURATION?

The reply to this question can be Y[ES] or N[O].

If the user replies Y or YES followed by (LF) (CR), GENMON assumes the following standard configuration definition:

- typewriter : TY10 at level /6
- tape reader : PR20 at level /4
- tape punch : PP30 at level /5
- line printer : LP07 at level /17
- card reader : CR06 at level /15
- cassette tape : TK05 at level /14
  TK15 at level /14
  TK25 at level /14
- magnetic tape : MT04 at level /13
  MT14 at level /13
  MT24 at level /13
- X1215 disc : BM02 at level /10 (removable cartridge).

(It is not necessary for the user to have all these devices in his configuration to be able to answer YES; but the ones he does have must then correspond to these standards.)

If the user replies N or NO, i.e. if one or more of the device addresses or levels is different from the standards above, GENMON outputs the following list on the typewriter, thereby allowing the user to define the configuration himself:

```
TY:
PP:
PR:
LP:
TK:
MT:
CR:
DK:
DISK TYPE:     (type in B)
LKM LEVEL:                  (standard = 1)
RTC LEVEL:                  (standard = 2)
PANEL INTERRUPT LEVEL:      (standard = 7)
```

For each of the devices listed, the user can reply as follows:

- (CR) if he wants the standard address and level (see above);
- <address>,<level> if one of these is different from the standards, followed by (CR)
- N or NO followed by (CR) if he wants the device excluded from the system.

When the user has terminated his reply to the first question, GENMON types out:

```
STANDARD FILE CODE ASSIGNMENT?
```

The procedure here is the same as for the first question: the reply may be either Y[ES] or N[O].
If the user replies Y or YES followed by (LF) (CR) GENMON assumes the following standard file code assignments:

- file code 1 : TY10 (system keyboard)
- file code 2 : LP07 (listing output)
- file code 3 : TK15 (object output)
- file code 4 : TK05 (object input)
- file code 5 : TY10 (system keyboard)
- file code 6 : TK05 (object input)
- file code 7 : TK25 (object input)
- file code 8 : PR20 (object input)
- file code A : TY10 (sysgen source input)

- file code B : TK15 (sysgen object output)
- file code E0: TY10 (system keyboard)
- file code E2: TK05 (Disload object input)
- file code EF: TY10 (system keyboard)
- file code F0: BM02 (Disload disc output;i.e. X1215)

If the user replies N or NO to this second GENMON question, i.e. if one or more of his file code assignments is going to be different from the standard list above GENMON outputs the following list on the typewriter, thereby allowing the user to give his own file code assignments (since the GENMON program is the same for all monitors, some of the file codes given in this list may be irrelevant to the generation of a DRTM and the user must type in NO after those):

```
LOAD INPUT DEV. AND MAIN LKE INPUT DEV. F.C./4: (standard = TK05)
SYSGEN INPUT DEV.                       F.C./A: (standard = TY10)
SYSGEN OUTPUT DEV.                      F.C./B: (standard = TK15)
AUX. LKE INPUT DEV 1                    F.C./6: (standard = TK05)
AUX. LKE INPUT DEV 2                    F.C./7: (standard = TK25)
AUX. LKE INPUT DEV 3                    F.C./8: (standard = PR20)
AUX. LKE INPUT DEV 4                    F.C./9: (no standard)
IPLGEN/LKE/CASLOAD OUTPUT DEV.          F.C./3: (standard = TK15)
LISTING OUTPUT DEV                      F.C./2: (standard = LP07)
CASLOAD INPUT DEV.                      F.C./C: (type in NO)
DISLOAD INPUT DEV.                      F.C./E2:(standard = TK05)
```

For each of the file codes listed, the user can reply as follows:
- (CR) if he wants the standard assignment (see above)
- <dev.name><dev.address> if one of these is different from the standards, followed by (LF) (CR)
- N or NO followed by (CR) if he does not want this file code taken into account
- if there is only one device of its kind, e.g. one PP, or in case a device which must be taken is the first of a series encountered in the standard list above, e.g. TK05, it suffices to specify only the device name, i.e. PP or TK.

When the user has terminated his reply to this question, GENMON types out:

END OF GENMON INITIALIZATION
READY TO LOAD PROGRAMS

Now the user can proceed to the next phase: DRMGEN.

Notes on GENMON:

For the question STANDARD CONFIGURATION:

- From the time the MC button has been pushed up to the end of GENMON initialization, no ready interrupts should occur.
- If the user has answered N or NO to this question, in the list typed out by GENMON, specification of a level is mandatory for LKM LEVEL. For RTC LEVEL it is mandatory if the CPU key is in the RTC/ON or LOCK position. For PANEL LEVEL it is also mandatory.
- The standards imply that the system disc will be the removable cartridge of the X1215 disc unit; therefore, if the user wants his system on the fixed cartridge, he must redefine DK under this question, e.g. as BM22.

For the question STANDARD FILE CODE ASSIGNMENT:

- File code /4: all programs will be loaded from this file code. During the syslink phase it is used as GENLKE object input, so it must be cassette or punched tape.
- File code /A: from this file code the parameters for table generation will be read. This may be done in interactive mode (e.g. TY) or not (e.g. PTR, cassette or magtape or card reader).
- File code /B: this may be cassette, punched tape or magnetic tape.
- AUX. INPUT DEV.: these may be assigned in advance, especially for syslink if libraries are to be scanned on various devices.
- File code /3: this is the main output device (sequential), i.e. cassette tape, punched tape or magnetic tape.
- File code /2: for logging of the sysgen operation: LP.
- File codes /4, /A, /B and /3 are mandatory; file codes /6 up to /9 and /2 are optional.

When answers to GENMON questions are given on an ASR type-

writer, they must not be typed in before the bell signal because of the low speed of the GENMON I/O module. This is not necessary for devices on V-24 interface such as the matrix typewriter P842, because they work in echo mode.

DRMGEN

When GENMON initialization is terminated and the message READY TO LOAD PROGRAMS has been output the user can start the second phase, DRMGEN, the building of the DRTM tables. This is done by typing replies to questions output on the typewriter by DRMGEN. From these replies DRMGEN builds the tables and records them on the medium with file code /B, i.e. in standard cases the cassette in drive TK15.
When the questions and answers are handled via the typewriter, this phase is done in conversational mode. It is also possible however, to do it in non-conversational mode, for example by having the questions and answers pre-recorded on punched tape. In such a case, file code /A must have been assigned to tape reader during the GENMON phase under the question STANDARD FILE CODE ASSIGNMENT?

The following actions must now be taken:
- when working with paper tape, take the IPL+GENMON tape from the reader, put the DRMGEN tape on it and make it operable; prepare the tape punch for output by switching it on and feeding tape
- when working with cassettes, put the first working cassette (hereafter called cassette W1) in cassette drive TK15; this cassette will receive the DRMGEN output.
- load a disc cartridge in the X1215 disc drive (this is the disc which will later on receive the generated system, depending on the disc defined under DK in the question STANDARD CONFIGURATION during GENMON)
- push the INT button on the CPU control panel
- on output of M: type in
  LD
- now the following sysgen processor, i.e. DRMGEN is loaded from the cassette or paper tape and its identification is output:
  IDENT DRMGEN

- when loading is terminated,
  :EOS
  :EOF are output on the typewriter

(At this point, if file codes /A and/or /B have been redefined under GENMON, i.e. if they are not assigned to TY10 or to a cassette respectively, prepare the relevant device. Normally, the cassette in drive TK15 should now be ready to receive the tables output by DRMGEN according to the replies given on TY10).

- push the INT button
- on output of M: type in
  ST
- now DRMGEN is started and outputs the following message on the typewriter:
  TABGEN INITIALIZATION
  IDENT SYSTEM DEFINITION
- then, if the user works in conversational mode, a series of questions is output, to which the user must type in the replies:

IDENT

Reply: Specify a character string of up to 6 alphanumeric characters, to be punched at the beginning of the module. This may be followed by a characters string of up to 73 characters, containing comments. The comment field must be separated from the ident field by a blank.

Error Message: TG03: the first character is not alphabetic.

STACK SIZE

Reply: Specify 4 hexadecimal characters, giving the size, in words, of the stack which is used by the system to save registers when an interrupt occurs.
Note that 10 decimal words are required in the stack for each accepted interrupt.

Error Message: TG03: the reply is not a hexadecimal number.

DYN AREA SIZE

Reply: Specify up to 4 hexadecimal characters, giving the size, in characters, of the dynamic allocation area in memory, shared by the system and user programs.
The minimum size to be declared is /580.
Note that, in addition to user requests for memory space, the system also requires memory space to process some of the user requests and external events, i.e.

- to activate a background program the system requires 205_{10} words during program loading;
- to assign a disc logical file, the system requires 205_{10} words during the assign process;
- when a disc sequential file is used the system requires 205_{10} words as long as that file remains opened;
- for the requests Connect a Level to a Clock or to a Timer and Wait for a given Time the system requires 4_{10} words from the time the request is issued until the program is disconnected from the time or until the given time has elapsed;
- for an activate request (from user as well as from system) 6_{10} words are required if the program to be activated is busy, until its activation becomes effective;
- for a request to Read an Unsolicited Key-In, 7_{10} words are required until the character string is input or until the request is cancelled.

Note: In addition to the size requested, the system reserves one extra word for block management in this partition.

Error Message: TC03: the reply is not a hexadecimal number.

READ ONLY LENGTH

Reply: Specify up to 4 hexadecimal characters giving the length, in characters, of the Read Only Area. This length must be equal to the size of the largest Read Only program used. It may not be smaller than /980, i.e. the size of the system read only program.

Error Message: TG03: the reply is not a hexadecimal number.

SWAP AREA LENGTH

Reply: Specify up to 4 hexadecimal characters giving the length, in characters, of the Swap Area. (Minimum: /196 characters). This area is optional, so the reply may be 0.

Error Message: TG03: the reply is not a hexadecimal number.

SWAPPING FILE CODE

Reply: Specify the file code of the disc (from /F0 to /FF) on which the core image file D:CI will be stored when swapping occurs. This file is used for read only as well as for swappable programs.

Error Message: TG03: invalid reply.

SWAPPING NUMBER OF GRANULES

Reply: Specify the size of the D:CI file, in granules. The minimum size must be 8 granules, due to the size of the system read only programs.
For user programs, the size must be defined as follows:

- Read Only Programs:
  if L = program length, then the number of

sectors required for this program is:

$$\frac{L - 1}{200_{10}} + 1$$

- Swappable Programs:
  if S = length of the swap area, the number of sectors required for each swappable program is:

$$\frac{S - 1}{200_{10}} + 1$$

After the requirement for each program has been calculated, add all the results, divide by 8 and add 9 (upper value rounded + 8 granules for the system read only programs) to obtain the definite reply, i.e. the number of swapping granules required.

Error Message: TG03: invalid answer.

SWAPPING TIME SLICE

Reply: Specify the standard time slice for swapping, in tenths of a second.
Because the System Loader cannot be interrupted by other software level programs (its standard level is 49), the time slice must be defined according to the type of disc on which the D:CI file is stored and so the length of the Swap Area.

Error Message: TG03: invalid answer.

CORE RESIDENT AREA LENGTH

Reply: Specify up to 4 hexadecimal characters giving the length, in characters, of the Core Resident Area. The size must be equal to the total length of the user programs declared as core resident at initialization time. If the area is not used, 0 must be specified.

Error Message: TG03: invalid reply.

PROG SAVE AREA LENGTH

Reply: Specify up to 4 hexadecimal characters giving the length, in characters, of the Program Save Area.

Note:

- Since the 10 system read only programs also use this area the minimum size declared must be /154.
- For user programs it must be extended as follows:
  - for each core resident, read only or swappable user program, add:
    17_{10} words if no scheduled label is used,
    34_{10} + 2n words if n scheduled labels will be declared in the program.
  - for a background program, a save area is allocated only when the program is in memory:
    34_{10} + 2m words, where m is the same value which is given in reply to the question MAX NUMBER OF SCHEDULED LABLES

Error Message: TG03: invalid reply.

USER INTERRUPT ROUTINES

Reply: Specify all user interrupt routines (usually drivers for special devices) which must be link-edited with the monitor for inclusion in the system. as follows:

```
<L>,<ENTRYi>
  :
  :
END
```

L, ENTRYi specifies the name (up to 6 characters) of a routine connected to the single level L.

END indicates that all routines have been declared or is used if none are entered.

Error Messages: TG03: syntax error
TG06: the first character of the name (ENTRYi/j) is not alphabetic
TG09: error in the level declaration.

POWER FAILURE

Reply: Specify the level of power failure option as follows:
<L> giving the level number 1 or 2 hexadecimal digits.

Specify N if this feature is not available in the system.

Note: The standard system power failure routine performs only a HALT.

Error Message: TG03: parameter error.

HALT WHEN POWER ON

Reply: Type in Y or N.
If the reply is Y, the system will, when power is on again after a failure, execute a Halt instruction to give the operator some time to attend to any devices that might need his intervention before the system is started again.
To select the user routine processing power failure and automatic restart, the user must include it during GENLKE, before entering the standard library. If he does not, the standard routine will be selected.

REAL TIME CLOCK LEVEL

Reply: <L>
where L is the level to which the clock is connected

ABS TIME MANAGEMENT NEEDED

Reply: Specify Y if connection to absolute time is wanted, or N if it is not wanted.
This question will appear only if the answer to the question REAL TIME CLOCK was neither N⌈O⌉ nor EN⌈D⌉ . If the reply is N, the connection to a timer remains available, as well as the Get Date and Get Time features (monitor requests).

PANEL LEVEL

Reply: When operator communication (OCOM) modules are selected, the user must specify the level to which the control panel interrupt is connected.
The level is specified as:
<L> a level number of 1 or 2 hexadecimal digits

Specify N if no operator communication feature is used.

Error Message: TGO3

LKM LEVEL

Reply: First specify the level to which the LKM interrupt is connected in the same manner as for the previous question. Then, define the optional monitor requests derived for your system (other

monitor requests provided in the software are standard). DRMGEN will print the names of the optional ones, after which the user may type Y if he wants it to be included or N if he does not.

DRTM lists only two optional functions:
DFM (Disc File Management) (required for SCL)

Note: The user must not issue a request in his programs for a function he has not included. For example, if he has typed N after DFM at DRMGEN, an I/O monitor request (LKM1) on a logical disc file may cause a system hang-up. Therefore users should define exactly the required functions.

Error Message: TG06: the reply is neither N nor Y. Try again.

RESIDENT MACRO

Reply: Some of the monitor requests which are standard to the DRTM may be core resident or disc resident. DRMGEN prints all the names of these requests one by one and the user replies by typing behind each one a Y if he wants it to be core resident or an N if he wants it to be disc resident.
TABGEN prints the following list:

CNTM (Connect a Timer)
DNTM (Disconnect a Timer)
ATDV (Attach a device/file)
DTDV (Detach a device/file)
GTIM (Get Timer)
CNLV (Connect A Level)
DNLV (Disconnect a Level)
WFGT (Wait for a Given Time)
ASSG (Assign a File Code)
DEFC (Delete a File Code)
KEYN (Read Unsolicited Key-in)
CNSL (Cancel Read Unsolicited Key-in)

Error Message: TG06: the reply was neither N or Y. Try again.

USER LKM

Reply: The user may specify his own set of monitor requests for inclusion in the monitor, as follows:

<N1>,<ENTRY1>
<Ni>,<ENTRYi>
<Nn>,<ENTRYn>

END

where Ni consists of two hexadecimal digits defining the DATA number which follows the LKM instruction, and ENTRY is the symbolic entry point of the routine processing the LKM DATA Ni function. The system will extend the monitor request table in which word i contains the address of ENTRYi. Therefore, during SYSLINK the user must provide the Linkage Editor with the module containing all entry points ENTRYi specified here.
END indicates that all user LKM functions have been declared or that none are wanted.
All user LKMs are processed by core resident programs working at level 48.

Note: Ni must not be equal to any of the standard LKM DATA numbers.

Error Message: TG03: the first parameter is not hexadecimal
TG07: the first character of the second parameter is not alphabetic
TG08: syntax error

PCT STANDARD LEVELS

Reply: DRMGEN will print a list of system programs, behind each of which the user may type Y or N to indicate whether he wants the standard level and assignations for that program or not.
If he types N, he must also type the level and

assignations he wants, as follows:

<level>, [DR | CR]

where level must be a software level between /31 and /3F exclusive.

DR must be specified if the program should not be core resident.

CR must be specified if the program is to be core resident.

Note:

The program names printed are:

TIMER (clock management) : M:DCK2 module

LOADER: D:LDER module

PNRLKM (possibly non-resident monitor requests): D:USV1, D:USV2, D:USV3 and D:RMAC modules.

GR ALLOC (granule allocation): D:ALGR and D:CTPN modules

OCOM (control panel interrupt): D:OCOM module

SWAP: D:SWP module

DUMP: D:DUMP module

SCL (system command language segments): D:ROOT, D:SEG1, D:SEG2, D:SEG3, D:SEG4

If Y is specified as the reply, DRMGEN prints and implements the following list:

TIMER	: 31	CR
LOADER	: 31	CR
PNRLKM	: 31	DR
GR ALLOC	: 31	CR
OCOM	: 31	CR
SWAP	: 31	CR
DUMP	: 32	DR
SCL	: 3D	DR

The Idle Task (IDLTSK module) will always be level /3F and core resident.

NB OF PROG CONTROL TABLES

Reply: Specify the number of program control tables which must be reserved for your system. For each user

program one entry in the program control table is allocated. Each entry occupies 18_{10} words. The user need not reserve entries for the system programs.

Error Message: TG03: invalid reply.

DEVICES ON PROGRAMMED CHANNEL

(Note: devices on channels must be declared according to Release Bulletin)

Reply: Specify, which devices are connected to the programmed channel, as follows:

<DNDA>,<L>

!

END

where:

DN is the device name, in two ASCII characters.

DA is the device address, in two hexadecimal digits.

<L> specifies the level to which the device is connected.

END indicates that all devices have been declared.

Note: When the configuration contains only 2 cassette drives (TK), the physical addresses (DA) of these drives must be /0X and /1X, X being the control unit address.

Note: For the operator's typewriter, three devices(TY, TP, TR) may be declared with the same address, if they are all used. No check is made on device declaration.

Error Messages: TG01: the device specified is not supported by the system.

TG02: device address error

TG09: level error

TG10: device not declared.

Device names used:

TR : ASR tape reader

TP : ASR tape punch

PR : high-speed tape reader

PP : high-speed tape punch

TY : operator's typewriter
CR : card reader
LP : line printer
TK : cassette tape unit
MT : magnetic tape unit

DEVICES ON MULTIPLEX

Reply: Specify, which devices are connected to the multiplex in the same manner as for the devices on the programmed channel.
Note: For line printer there are additional parameters:
- Line printer:
  LDPA,L,LG, number
  where LG=[S|L]: S = 80-column printer
  L = 132-column printer.
  <number> : a hexadecimal number specifying the number of lines per page wanted.

No check is made on the device declaration.

DISK UNITS

Reply: Specify all disc units connected to the I/O processor as follows:
DNDA,L,Q
END
where:
DN is the device name, giving the type of disc on 2 ASCII characters as:
X1: X1215 X2: X1216
D1: CDC 40M D2: CDC 80M
Note: For the X1215, each disc pack, not disc unit must be declared, for there are 2 packs to one unit.

DA is the device address, 2 hexa digits (see App.).
L specifies the interrupt level to which the disc is connected (1 or 2 hexadecimal digits)
Q defines, on 2 hexadecimal digits, the length of the disc queue table, in number of entries

Note:

- The disc queue table is used only if DFM has been selected. If it has not, this parameter must be 0. The number of entries in the disc queue table must be equal, for each disc unit, to the maximum number of disc logical I/O which may be processed simultaneously on the same unit. The answer must therefore be equal to the one given under DISC LOGICAL FILES. given

NBR OF DAD CONTROL TABLES

Reply: A 2-digit hexadecimal number, indicating the number of DAD file codes (from CF downwards as far as C0) which has to be assigned internally.

TOTAL LENGTH OF BITABS

Reply: A 2-digit hexadecimal number, giving the total length of the required BITAB tables (see Disc Organisation).

SPECIFY FILE CODES

Reply: Specify all file codes used by the programs. If system processors are used, their standard file codes must be declared.
Declaration is done as follows:

```
<FC>,<DNDA>
     ¦
<FC>¦,<DNDA>
    END
```

where FC is a file code (2 hexadecimal digits) assigned to the device indicated by DN with address DA.

Note:

File codes /EF and 01 are used by the system to read/write to/from the operator's typewriter. It is therefore necessary to declare at least these file codes, if the typewriter is going to be used. Other file codes may be assigned by using the command AS.
Some file codes, i.e. /02, /E0, /F0 to /FF are used by the system, but they have to be assigned, too.

FILE CODES ASG TO USER DEVICES

Reply: The user may declare all file codes assigned to his non-standard devices. The related I/O drivers (including Device Work Tables) and the interrupt routines for these devices must be written by the user and be incorporated in the system during the GENLKE phase. (The entry points for the interrupt routines must be declared under USER INTERRUPT ROUTINES). These file codes must be declared here, as they cannot be assigned by AS operator message or ASG control command. The reply must be as follows:

```
<FC>,<DWTi>
    ↓
   END
```

where:
FC is a two-digit hexadecimal file code which will be generated in the File Code Table.
DWTi is the entry point (a name of up to six characters) of the Device Work Table (DWT) associated with this device.
END indicates that all file codes have been declared.

Any number of file codes can be assigned to the same DWTi. The system checks only if the file code is a two-digit hexadecimal number, but not if it has already been used or is one of the standard file codes used by the system processors.

Note

The user devices can be assigned by means of the SCL command AS only with the following syntax:

AS␣<file code1>,<file code2>

It is therefore necessary to define <file code2> during DRMGEN for each user device and it must not be reassigned under the DRTM.

Error Message: TG03: syntax error, e.g. the file code has not been specified as hexadecimal.

TG07: the first character of the DWTi is not alphanumeric.

SPARE ENTRIES IN FCT

Reply: Specify the number of spare entries reserved in the File Code Table (FCT), on one or two hexadecimal digits.
These entries are required for assignment of user file codes at execution time or for assignment of temporary system file codes.
Each entry takes up two words.

DISC LOGICAL FILES

Reply: This message is printed only in case the user has selected the Disc File Management (DFM) option (see under LKM LEVEL), to find out the number of disc logical files used in system.
Specify two hexadecimal digits, giving the maximum number of logical files which can be assigned (and not yet opened) at the same time.
Each entry takes up 28_{10} words, as the system will

reserve a Logical File Table for each disc logical file, to store all the information about it.

SIMULATED INSTRUCTIONS

Reply: Y or N, depending on whether the simulation package must be included or not. If the reply was Y, the following list is output:
MULTIPLY:
DIVIDE:
D ADD:
D SUB:
D SHIFT:
MLR-MSR:
After each item the user must type in Y or N to indicate whether he wants that instruction simulation routine included or not. For P856/857 the answer should be N, for the instructions are included in the instruction set.

SIMULATED ROUTINES SAVING AREAS NB

Reply: This question is output only if the reply to the previous question was Y.
The user must type in the number of save areas required by the simulation package.
For DRTM, it is the maximum number of programs, main sequences or scheduled labels which might be simulated at the same time.

MAX NUMBER OF SCHEDULED LABELS

Reply: Type in a two-digit hexadecimal number, specifying the maximum number of scheduled labels which may be in queue at the same time. This will be the length of the FILLAB table described in the paragraph on Scheduled Labels in Chapter 4.

Note: This is not the maximum number of scheduled labels used in the program, which may be a higher number.

TABGEN ENDED

PREMDK

This is a program which checks the disc on which the generated system must be written for defective tracks, writes sector identifiers on it and an Initial Program Loader (IPL) in the second sector.
PREMDK outputs a number of questions on file code /EF (i.e. TY10) to which the user must type in the replies. Output is done onto file code /F0, i.e. the system disc.
Operation is as follows:

- with paper tape, put the PREMDK tape on the reader; make it operable
- push the INT button on the CPU control panel
- on output of M: type in

  LD
- now PREMDK is loaded as the next processor from the cassette C1 in drive TK05 or the tape reader and its identification is output:

  IDENT PREMDK
- when loading is terminated,

  :EOS

  :EOF

  is output on the typewriter.
- push the INT button
- on output of M: type in

  ST
- now PREMDK is started and the following messages and questions are output on the typewriter:

  INITALISATION OF PREMRK

  COMPLETE PREMARK: Reply: VV is typed to have PREMRK write only the VTOC and label.
  WW if the user wants PREMRK to only write the sector identifiers.
  (CR) if the user wants PREMRK to write and verify the sector identifiers.

  DISK TYPE: Reply: Enter 2 characters identifying the type of disc to be premarked: X1 for X1215, X2 for X1216, D1 for CDC 40M, D2 for CDC 80M.

  LABEL: Reply: Type in no more than 16 characters for the volume label. If it is less, it will be left-justified and padded with blanks.

DATE:	Reply:	Type in 6 decimal characters, giving day, month and year.
PACK NBR:	Reply:	Type in no more than 4 characters for the pack number. If it is less, it will be right-justified and zero-padded.
DISC UNIT PHYSICAL ADDRESS:	Reply:	Type in 2 hexa characters giving the device address.
DAD NAME:	Reply:	Type in no more than 6 characters, specifying the name of the first DAD. If it is less, it will be left-justified and padded with blanks.
NBR OF CYL OF dadname:	Reply:	Specify the number of cylinders of this DAD. (Max. per disc type: X1215: 200; X1216: 400; CDC40M/80M: 103).
NBR OF INT OF dadname:	(CDC discs only) Reply:	Specify the interlace factor of this DAD (not the number 13, no multiples of 3).
SYST. USERID:	Reply:	Type in the name of the userid; no more than 8 characters. If less, it will be left-justified and filled with blanks.
ANOTHER DAD:	Reply:	OK if more DADs are to be specified for this disc. The sequence will then restart at DAD NAME. NO if this is the only or last DAD.

END OF PREMARK

Error Messages:

- BAD GRANULE 0: the disc pack is not usable
- NO SYSTEM USER POSSIBLE: granule 1 is bad and therefore no system can be stored on this disc pack.

Now the disc is premarked and ready to receive the generated system. But first we have the GENLKE phase, during which the tables generated under DRMGEN are linked with the modules required from the DRM Library and, possibly, user and/or extension libraries to generate the user's monitor.

Note: If sysgen is to be done under control of the Disc Operating Monitor, the user must reload the system by means of IPL procedure after Premark.

GENLKE

During this phase the final user monitor is obtained by linking the tables generated under DRMGEN with the monitor modules required from the DRM Library and/or any user library or extension libraries (see note at the end of this section).
The input to GENLKE is done from file code /4, i.e. cassette or paper tape. The output is done onto file code /3, in standard cases working cassette W2 in drive TK15, but it may also be punched tape or magnetic tape.
In any case, the GENLKE processor must now be loaded:

- when using paper tape, take the PREMDK tape from the reader, put on the GENLKE tape and make the reader operable
- push the INT button on the CPU control panel
- on output of M: type in LD
- now GENLKE is loaded from the cassette tape or the tape reader and its identification is output:
  IDENT GENLKE
- when loading is terminated,
  :EOS
  :EOF
  is output on the typewriter.

With cassettes:

On the basis of the availability of two cassette drives, the cassettes are now handled as follows (with three drives, see below):

- take cassette G1 from drive TK05 after GENLKE has been loaded
- take cassette W1 (containing the tables generated under DRMGEN) from drive TK15 and put it into drive TK05 and wait for it to be rewound
- put the second working cassette W2 into drive TK15
- now start GENLKE as follows:
- push the INT button
- on output of M: type in
  ST
- GENLKE outputs
  L:
  and the user must type in the link-edit command as follows:

E [:<decimal number>] ,<module name> [,[8|4]]

where <decimal number> consists of three digits:

- the first is the file code for the object output device
- the second is the file code for the listing device
- the third is the file code for the object input device.

<module name> is the name of the user's monitor

8 or 4 is used if the monitor is punched on paper tape to indicate whether it must be punched in 4-track ot 8-track format.

If the standard file codes are used (see under GENMON, i.e. /3,/2 and /4 respectively), they need not be specified and the command can be given as:

E,<module name>

- then GENLKE outputs

  L:

  to which the user must reply with

  P

  The tables generated during DRMGEN are now recorded from the cassette W1 in drive TK05 onto cassette W2 in drive TK15.
- when this is finished take cassette W1 from drive TK05
- put generation cassette G1 into drive TK05 with side B up (DRMLIB1) and wait for it to be rewound, so that it is positioned correctly for the scanning of the first part of the DRTM Library.
- in response to the

  L:

  output by GENLKE, now type in

  L

  upon which GENLKE will start scanning part 1 of the DRTM Library, select the required modules and record them onto the cassette W2 in drive TK15. The names of the selected modules are output on the listing device, together with their base addresses and any comments included in the identifiers.
- when this is finished, GENLKE again types out

  L:

  The user must now type in

  U

  to check if there are any unsatisfied references. This will appear to be so, for at least INIMON must remain as an unsolved reference, so:

- GENLKE now lists the references remaining to be solved in the second part of the DRTM Library
- take cassette G1 from drive TK05
- put cassette G2 in drive TK05 with side A up and wait for it to be rewound
- after the
  L:
  typed out by GENLKE, the user must now type
  L
  upon which GENLKE will start scanning the second part of the DRTM Library, selecting the required modules and recording them on cassette W2 in drive TK15. The names of the selected modules are output on the listing device, together with their base addresses and any comments included in the identifiers.
- when this is finished, GENLKE again types out
  L:
  The user again types in
  U
  to check for unsatisfied references. The last one to be solved must be INIMON, so if GENLKE types
  INIMON
  (possibly followed by ASEX, a module referenced by INIMON)
  after the user has typed in U, it is correct.
- GENLKE then types out
  L:
  and the user must type
  L
  to solve this last unsatisfied reference.
  (If there were more unsatisfied references, the user must repeat this L:L process until INIMON is the last unsolved reference and then give his last L command)
- now, after all modules have been included, GENLKE again types:
  L:
- the user once more types
  U
  to make sure that all references have been solved. Then, on the next
  L:
  the user types
  T
  to indicate the end of the GENLKE phase
- GENLKE then outputs the symbol table of the generated DRTM

on the listing device and on the typewriter it outputs monito length (L = XXXX), monitor start address (S = XXXX) and the first free location after the monitor in memory (E = XXXX).

Note: If the user has three cassette drives and wants to use them all during this phase, the procedure is as follows:

- after loading GENLKE, take cassette G1 from drive TK05 and put it back in with side B up. Wait for it to be rewound.
- put cassette W1 (tables output by DRMGEN) into drive TK25 and wait for it to be rewound
- put the second working cassette W2 in drive TK15
- start GENLKE by pushing the INT button and, on output of M: typing in ST
- on output of L: type in the option message as follows:
  E:327,<module name>,8
  where 3 and 2 are the normal object output and listing file codes and /7 is now assigned to the input file code, so not /4 = TK05, but /7 = TK25. This is because TK25 contains the cassette W1 with the tables generated under DRMGEN. See also under GENMON for the file code assignments.
- when this is accepted, GENLKE outputs L: and the user types
  H
  Upon this command, GENLKE scans the input file (i.e. the cassette W1 with the tables) up to EOF, then goes into Pause state.
- now GENLKE is restarted with an RS command with a new file code, switching it back from /7 to /4, the normal input file code and the one assigned to TK05, which contains the DRTM Library part 1 which must now be scanned. So:
- push the INT button and on output of M: type
  RS 04
- GENLKE now starts scanning and selecting the required modules from the DRTM Library 1 and the rest of the procedure is the same as described above, starting after the user's first L command.

With paper tape:

- in this case the procedure is basically the same as with cassettes (see description above), but input is done from the tape reader and output onto tape punch. On the tape punch an IPL has already been generated and the paper tape should be left as it is.

- first the GENLKE tape must be put on the reader and GENLKE is loaded into memory by LD command. Then the tape containing the tables generated under DRMGEN is put on the reader and GENLKEBis started with an ST command. Having entered the E: option command, the user types P and the tables are processed.
- then the DRMLIB 1 tape is put on the reader and the user types L after which this library is scanned and the selected modules are output on the punched tape.
- if the user types in U, the unsolved references are listed. They should all be contained on the DRMLIB 2 tape. This tape is now put on the reader and also processed with an L command.
- having then checked if INIMON (+ ASEX) is the last unsolved reference with a U command and having typed L to solve it, the user then types T to terminate the process.
- on the tape punch a DRTM has now been generated.

Note:
If modules from other (paper or cassette) tapes beside the DRMLIB tape must be link-edited during this phase, the tapes must be scanned in a defined order, which is:
- User Library tape(s), if any
- Extension tape(s)
- DRMLIB tape

When more references than INIMON remain unsatisfied, rescanning must start at the first step in this sequence.

DISLOD

DISLOD is the last system generation processor. It is used to record the generated monitor and monitor modules onto the system disc specified under GENMON, in load module format accepted by the DRTM.
The standard object input file code for DISLOD is /E2, (TK05), the output disc file code is F0 (BM02), the listing file code is /2 (LP07) and all conversational processes are done through file code /EF (TY10).

- when working with paper tape, put the DISLOD tape on the tape reader and make it operable
- with cassette, the cassette tape in file code /E2 is positioned after the DRMLIB 2, i.e. before DISLOD)
- push the INT button on the CPU control panel
- on output of M: type in LD
- now DISLOD is loaded from the tape reader or the cassette and its identification is output:
  IDENT DISLOD
- when loading is terminated,
  :EOS
  :EOF is output on the typewriter.
- with paper tape, DISLOD tape must now be taken from the reader and the newly generated DRTM tape must be placed on it, ready to be read
- with cassettes, take cassette G2 from file code /E2 (normally TK05) and replace it by cassette W2 (containing the new DRTM) which must first be taken from file code /3 (normally TK15).
- push the INT button
- on output of M: type
  ST
  to start the DISLOD processor.
- DISLOD then outputs the message
  SYSLOAD XX F852
  on the typewriter, followed by the question
  NEXT ACTION:
- at this point, the user has three possibilities for action:
  - if he types (LF) (CR) the next program or module on the input file (/E2) will be recorded onto the system disc. DISLOD will output the program name onto the typewriter (PROG.NAME - XXXXXX) and list name, length in sectors,

start address and program length on the listing device (/2).

After this, DISLOD will again output the message NEXT ACTION:,

- if he types PS, DISLOD goes into Pause state, enabling the user to modify an assignment, operate on the cassettes by manual control, etc. To restart DISLOD, the message RS must be typed in (after having pushed the INT button), which may be followed by a parameter containing a new object input file code.
- if he types HT, DISLOD performs an Exit and the process is stopped.

- so, after the first NEXT ACTION: message output by DISLOD, the user types (LF) (CR) and his newly generated monitor is recorded from cassette W2 or paper tape onto the system disc specified by the user under GENMON and premarked under PREMDK.
- when this is finished, DISLOD again outputs
  NEXT ACTION:
- with paper tape, now put the first of the monitor segment and system processor tapes on the reader, i.e. the D:USVI tape (see list of sysgen tapes at beginning of chapter)
- with cassettes, now take cassette W2 out of the drive and replace it by the second generation cassette (G2), with side A up, containing the disc-resident parts of the monitor

- now, after each NEXT ACTION: message, the user must type (LF) (CR) to include successively the disc-resident monitor parts and the system processors. With cassettes, this requires no further manual operations, but with paper tape, the user must put the following tape in the reader each time before typing in (LF) (CR)
- when the last one (IPLGEN) has been recorded onto the system disc and DISLOD again types
  NEXT ACTION:
  the user must type in
  HT
  to terminate the DISLOD operation.

The user now has a Disc Real Time Monitor on disc.

DISC PREMARK

PREMRK is a stand-alone program to be used for formatting a disc pack before it will actually be used. It divides the disc into sectors writes identifiers in them and checks for bad tracks.
PREMRK is in absolute format, so it is loaded by IPL.
After it has been loaded, it starts to type out the following questions on the operator's console and the user can type in his answers:

INITIALISATION OF PREMRK

COMPLETE PREMARK:	Reply:	VV is typed to have PREMRK write only the VTOC and label. WW if the user wants PREMRK to only write the sector identifiers. (CR) if the user wants PREMRK to write and verify the sector identifiers.
DISK TYPE:	Reply:	Enter 2 characters identifying the type of disc to be premarked: X1 for X1215, X2 for X1216, D1 for CDC 40M, D2 for CDC 80M.
LABEL:	Reply:	Type in no more than 16 characters for the volume label. If it is less, it will be left-justified and padded with blanks.
DATE:	Reply:	Type in 6 decimal characters, giving day, month and year.
PACK NBR:	Reply:	Type in no more than 4 characters for the pack number. If it is less, it will be right-justified and zero-padded.
DISC UNIT PHYSICAL ADDRESS:	Reply:	Type in 2 hexa characters, giving the device address.
DAD NAME:	Reply:	Type in no more than 6 characters,

		specifying the name of the first DAD. If it is less, it will be left-justified and padded with blanks.
NBR OF CYL OF dadname:	Reply:	Specify the number of cylinders of this DAD. (Max. per disc type: X1215: 200; X1216: 400; CDC40/80M: 103)
NBR OF INT OF dadname:	(CDC disc only) Reply:	Specify the interlace factor of this DAD (not the number 13, no multiples of 3).
SYST. USERID:	Reply:	Type in the name of the userid; no more than 8 characters. If less, it will be left-justified and filled with blanks.
ANOTHER DAD:	Reply:	OK if more DADs are to be specified for this disc. The sequence will then restart at DAD NAME. NO if this is the only or last DAD for this disc.

END OF PREMARK

Appendix C Peripheral Input/Output

Below, device driver imformation is given for each I/O order, except for the order for the driver for the video display terminal, which are grouped together at the end of this Appendix.

This information applies to the order, which the user must specify in an I/O monitor request.

BASIC READ (/01)

Operator's Typewriter
All characters are entered on 8 bits until the requested length is reached.

ASR Tape Reader
All characters are entered on 8 bits. The reader stops one character after an Xoff code has been read.

High-speed Tape Reader
All characters are entered on 8 bits, without checking or special features, until the requested length is reached.
Basic Read does not ensure a stop on character.

Card Reader
All the words are entered and stored in Hollerith code on 12 bits (4 to 15). In each word the column image is right-justified. The words are stored until the requested length is reached. The length is given in words.

Disc
With the aid of Data Management all the sector words are entered in the memory buffer.

Magnetic Tape Cassette:
All Read/Write operations (Basic, Standard, Object) are the same, with the following characteristics:
- maximum record length: 256 characters
- required length: block length.

- effective length: block length (without control character).

- all read/write operations are done on the requested length
- incorrect length after read operation: no error, if requested length is greater than block length and the returned status is correct.
- throughput error or data fault: retry is made automatically, up to five times:
  - after read: backspace - read
  - after write: backspace - erase - write.

Magnetic Tape:
Same as for cassette tape, with the following differences:
- maximum record length: 4095 characters; minimum 12 characters.
- required length: block length without control word.

- physical block length: required length + 2.
- effective length: block length.
- 12 characters are always transferred, in any case.
- incorrect length: see cassette tape above.

BASIC WRITE (/05)

Operator's Typewriter
All characters are output without checking or special features. This order can be used to print something and have the answer on the same line.

ASR Tape Punch
All characters are output without checking or special features.

Line Printer
All characters are output without checking. There is no control character.

High-speed Tape Punch
All characters are output without checking or special features.

Disc
With the aid of Data Management all the sector words are output onto the disc.

Cassette and Magnetic Tape
See under Basic Read (/01).

STANDARD READ (/02)

Operator's Typewriter
ASCII characters are entered on 8 bits, with the following special features:

- the special characters, coded from /0 to /1F, are ignored.
- code /7F (Rub-out or Delete character) is ignored.
- code /5F (←) can be used to delete the preceding character. If several ← are used consecutively, an equal number of preceding characters will be deleted.
- code /5E (↑) is used to delete the line preceding it, up to the next carriage-return.
- code /0D (carriage return) indicates end of block. It is the last character to be entered. It is not transmitted to the user's buffer.
- code /0A means 'line feed'.
- code /5C (\) is used as a tabulation symbol (see ECB word 5). If the address of the tabulation table is zero, or if the number of tackets is zero, or if the storage address is greater than the last tacket, the code /5C is stored in the buffer. In other cases, /5C is not stored and replaced by spaces, as indicated by the tackets in the tabulation table.

ASR Tape Reader
For ASCII characters, the same features apply as for the keyboard: the code for carriage return must be preceded by the code for Xoff.
For object code in 4+4+4+4 tape format, the first character identifies the object format. It must be in the range from /18 to /1F and is converted to a number from /0 to /7 and stored on one character. The second character contains the word-count of the input block, excluding the first word and the checksum. Each punched row (4 bits) entered after this identifier is stored on one half-character up to the checksum. When the checksum has been read, input is stopped. The 8+8 tape format cannot be read on the ASR tape reader. To start the reader, an Xon code is sent by the system before entering the characters.

High-speed Tape Reader
Same as for the ASR tape reader. In addition: for object code in 8+8 format, the first character, identifying the object code format, must have one of the following values: /10, /1 to /4 or /15 to /17. It is converted to a number from /0 to /7. Each punched row (8 bits) entered after this identifier is stored on one character up to the checksum. The second character is the length of the block, in words, excluding the first word and the checksum.

Card Reader
All words are read in Hollerith code, on 12 bits, converted and stored in ASCII code, on 8 bits, until the requested length is reached. Words which are not in Hollerith are converted into the ASCII code for /20 and a 'data fault' status is returned in the software status (ECB word 4: bit 13 is 1). There is no special code. However, EOS and EOF marks are detected (bits 14 and 15 in the software status).

Cassette and Magnetic Tape
See under Basic Read (/01).

STANDARD WRITE (/06)

Operator's Typewriter
All characters, except /0 to /1F (special code characters) are output without checking. At the end of a line, a carriage return and line feed are output. The first word in the buffer contains a right-justified control character, as for the line printer (see below). If it equals /30 or /31, it is output as line feed; if it is different, it is not output.

ASR Tape Punch
Same features as for the keyboard. At the end of a line, the following character sequence is output: LF - Xoff - CR - Rubout.

High-speed Tape Punch
Same as for ASR tape punch.

Line Printer
All characters are output without checking, except for the control code. The first word in the buffer contains a right-justified control char. This control code may have one of the following three values:
+ (/2B): print the line without advancing the paper (superposition).
0 (/30): advance two lines before printing.
1 (/31): skip to top of page before printing.
All other control codes are used as normally: advance one line and print. At the end of the buffer, after the requested length, one character must follow to be used by the system for a print code.
If the requested length is more than one line, the system puts a print code after the maximum length and the buffer will be printed on two or more lines.

Cassette and Magnetic Tape
See under Basic Read (/01).

OBJECT WRITE 4+4+4+4 TAPE FORMAT (/07)

ASR Tape Punch
The first character is output on one row, converted from /0 - /7 to /18 - /1F. Each following character is output on two rows; to avoid special (ASCII) code each row is converted. The second character contains the length of the block in characters, excluding the first character. At the end an 8-bit checksum is performed and punched, followed by an Xoff code.

High-speed Tape Punch
Same as for ASR tape punch, except that the second character contains the length in words.

OBJECT WRITE 8+8 TAPE FORMAT (/08)

High-speed Tape Punch
The standard object code is output in 8+8 format, where the first character is a format character and is output on one row, converted as follows:
/0 → /10
/1 to /4 → /01 to /04
/5 to /7 → /15 to /17
The second character contains the length in words, excluding the first word. An 8-bit checksum is performed and punched.

Cassette and Magnetic Tape
See under Basic Read (/01).

READ SECTOR (/11)

Flexible Disc

Any sector on a flexible disc can be read at random with this order.

The requested length, as expressed in characters in ECB word 2, determines the record length (1,2,3 or 4 sectors). The length must be an even number with a maximum of 512 characters. If an odd number is given (2n+1), the driver transmits an even number (2n). ECB5 must contain the absolute sector number (0 - 2001) of the sector to be read. If more than one sector must be read (128 requested length 512), only the number of the first sector must be specified.

WRITE SECTOR (/15)

Flexible Disc

With this order any sector on a flexible disc can be written at random.

The requested length, as expressed in characters in ECB word 2, dtermines the record length (1, 2, 3 or 4 sectors). The length must be an even number with a maximum of 512 characters. If an odd number is given (2n+1), the driver transmits an even number (2n). ECB5 must contain the absolute sector number (0 - 2001) of the sector to be written. If more than one sector must be written (128 requested length 512), only the number of the first sector must be specified.

READ UP TO END-OF-SEGMENT (/14)

High-speed Tape Reader
The tape is read until an :EOS statement has been read.

Card Reader
The cards are read until an :EOS statement has been read.

READ UP TO END-OF-FILE (/16)

High-speed Tape Reader
The tape is read until an :EOF statement has been read.

Card Reader
The cards are read until an :EOF statement has been read.

WRITE EOF MARK (/22)

Operator's Typewriter
An end-of-file mark is output as follows: :EOF LF Xoff CR Rub-out

ASR Tape Punch
An end-of-file mark is output as follows: :EOF LF Xoff CR Rub-out

High-speed Tape Punch
An end-of-file mark is output as follows: :EOF LF Xoff CR Rub-out

Line Printer
An end-of-file mark is output as follows: :EOF

WRITE END-OF-VOLUME (/24)

End-of-tape management for Magnetic and Cassette tapes is a user program responsibility.
When the physical end of a tape is encountered during a write operation, a status is returned in ECB word 4 with the EOT bit set. The user may then issue a Write EOV request (/24; Write End-Of-Volume; see under I/O monitor requests), before requesting the operator to mount a new tape. When a new tape is mounted, for magnetic tape the unit must first be switched off by pressing the OFF LINE button, while for cassette tape a Manual Control (MC) operator command 'Unlock' must be given to enable the operator to remove the cassette.
Then the operator can mount a new tape reel or cassette and restart the program.
To ensure that all records will be retrieved when the file is read, the EOT (end-of-tape) status also returned in the status word of the ECB should be ignored and only the EOV status must be taken into account.

Note:
In case the EOT is detected while reading an EOV, only the EOV status is returned.

WRITE EOS MARK (/26)

Operator's Typewriter
An end-of-segment mark is output as follows: :EOS LF Xoff CR Rub-out

ASR Tape Punch
An end-of-segment mark is output as follows: :EOS LF Xoff CR Rub-out

High-speed Tape Punch
An end-of-segment mark is output as follows: :EOS LF Xoff CR Rub-out

Line Printer
An end-of-segment mark is output as follows: :EOS

Magnetic Tape Cassette
An end-of-segment mark is written as /6F6F.

Magnetic Tape
An end-of-segment mark is written as :EOS + 8 blank characters.

DOOR UNLOCK (/2D)

Flexible Disc

With this order the door of the flexible disc drive unit identified in ECB word 0 is unlocked. The other words in the ECB must be filled with dummy parameters:

ECB0: file code
ECB1: dummy value
ECB2: dummy value
ECB3: updated by the flexible disc driver
ECB4: updated by the flexible disc driver
ECB5: dummy value

DOOR LOCK (/2E)

Flexible Disc

With this order the door of the flexible disc drive unit identified in ECB word 0 is locked. The other words in the ECB must be filled with dummy values:

ECB0: file code
ECB1: dummy value
ECB2: dummy value
ECB3: updated by the flexible disc driver
ECB4: updated by the flexible disc driver
ECB5: dummy value

WRITE DELETED DATA ADDRESS MARK (/2F)

Flexible Disc

With this order a special mark, providing IBM compatibility, is written on the sector. The length of the record on which the deleted data address mark is written is that specified as the requested length in the ECB, which must be expressed in characters.

The buffer of the calling program must contain the information to be written on the sector(s) with a deleted data address mark. This information may be anything the user chooses to write.

ECB5 must contain the number of the sector in which the deleted data address mark is to be written.

If the deleted data address amrk is to be written in more than one sector, only the number of the first sector is to be given.

See also order /3D.

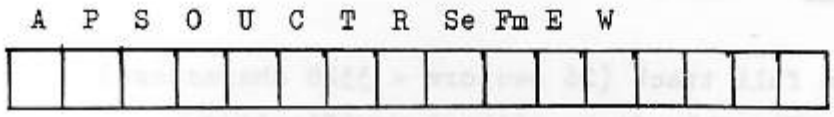
RETURN INFORMATION ABOUT A FILE CODE (/30)

By means of this order it is possible to find out the assignment of a file code. The information will be returned in the Event Control Block:

ECB - word 0: File Code
ECB - word 1: Device Name (2 ASCII characters):
TY = operator's typewriter (listing)
TR = ASR tape reader
TP = ASR tape punch
PR = tape reader
PP = tape punch
LP = line printer
CR = card reader
MT = magnetic tape
DK = disc
TK = cassette tape

If NO device is assigned, the ECB contents are set to zero.
ECB - word 2: maximum record size.
ECB - word 3: left character: unused.
ECB - word 3: right character: device address.
ECB - word 4: status = 0. For line printer, it contains the number of lines specified for this printer at system generation time.

If this order is used for a logical disc file code, the system will return the device name DL in ECB word 1, and word LFTMD1 from the Logical File Description Table (LFT) in ECB word 4 and fill the other words of the ECB with zeroes. LFTMD1 contains the following information:



A=0: LFT busy
1: LFT not active
P=1: file write-protected
S=1: source file
O=1: object file
U=1: undefined file
C=1: load module
T=1: temporary file
R=1: random access
Se=1: sequential access
Fm=1: file closed (file mark written)
E=1: current entry already in queue
W=0: write request
1: read request

OFF LINE (/38)

Magnetic tape:
This order switches the machine off.

Cassette Tape:
This order unlocks the cassette from the drive unit.

COMPOUND READ (/3A)

Flexible Disc

With this order one full flexible disc track (26 sectors = 3328 characters) is read into the buffer in one revolution of the disc. This function is performed correctly only if the flexible disc unit is connected to the I/O Processor. If it is connected to Programmed Channel, the order is also accepted and executed, but then more than one revolution of the disc is required.
ECB1 must contain the buffer address.
ECB2 must contain the requested length: any even-numbered value from 2 to 3328. If an odd number is given (2n+1), the driver transmits an even number of characters (2n).
ECB5 gives the absolute sector number from which the read operation must be started. There is no automatic chaining of sectors on the next track.

COMPOUND WRITE (/3B)

Flexible Disc

With this order one full track (26 sectors = 3328 characters) is written onto the flexible disc, without verification.
This function is performed correctly only if the flexible disc unit is connected to the I/O Processor. If it is connected to Programmed Channel, the order is also accepted and executed, but then more than one revolution of the disc is required.
ECB1 must contain the buffer address.
ECB2 must contain the requested length: any even-numbered value from 2 to 3328. If an odd number is given (2n+1), the driver transmits an even number of characters (2n).
ECB5 gives the absolute sector number from which the write operation must be started. There is no automatic chaining of sectors on the next track.

SEARCH KEY WITH MASK (/3C)

Flexible Disc

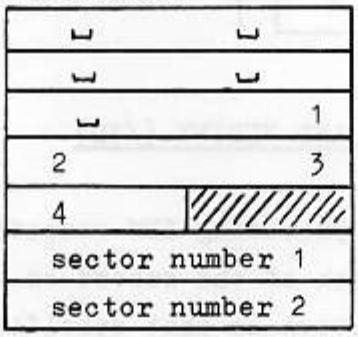
With this order a given character configuration can be searched for anywhere in a sector.

This is done by introducing dummy characters to mask the characters preceding the key inside the sector in which it is supposed to be located.

The dummy characters are given as blanks and are not significant in the search operation.

To set up the mask and indicate in which sector the search must be done, a searched key block must be made, the address of which is to be put in ECB1.

For example, if the key (character configuration) to be searched for begins at the 6th character in a sector, the preceding 5 characters should be given as blanks:



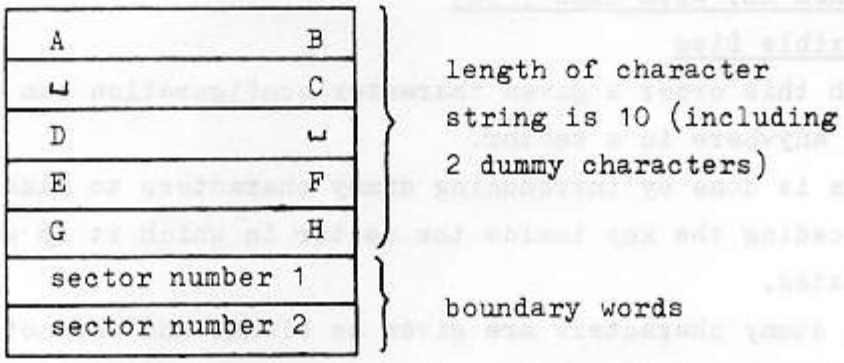
The boundary words contain the absolute sector numbers of the sectors between which the sector with the searched key is expected to be found.

ECB2 specifies the length of the character string (dummy characters plus key characters) which must be scanned for the key, excluding boundary words (maximum length: 128 characters, minimum length: 1 character).

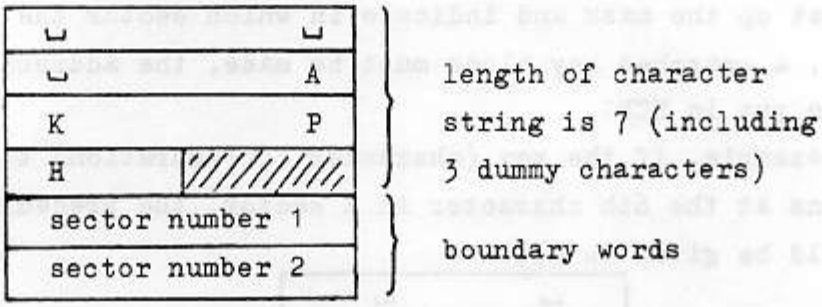
ECB5 is not significant, but on completion of the search, if the key has been found, ECB5 contains the address of the sector containing the key.

The layout of the searched key block is as follows:

- for even number of characters:



- for odd number of characters:



WRITE DELETED DATA ADDRESS MARK AND VERIFY (/3D)

Flexible Disc

With this order a special mark, providing IBM compatibility, is written on the sector. The length of the record on which the deleted data address mark is written is that specified as the requested langth in the ECB, which must be expressed in characters.

The buffer of the calling program must contain the information to be written on the sector(s) with a deleted data address mark.

ECB5 must contain the number of the sector in which the deleted data address mark is to be written.

If the deleted data address mark is to be written on more then one sector, only the number of the first sector is to be given.

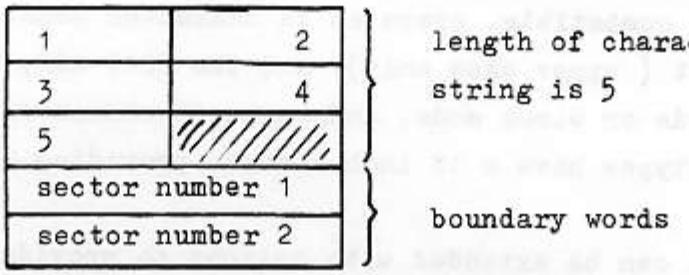
After the record with the deleted data address mark is written, the control unit will use the next revolution of the disc to read and verify the mark.

The Deleted Data Address Mark may contain any information. It is a hardware recognized character implemented for IBM compatibility.

SEARCH KEY (/3E)

Flexible Disc

With this order a given character configuration is searched for within a sector, starting from the beginning of the sector. A searched key block must be made, containing the searched key (character configuration) and the numbers of the sectors between which the sector with the searched key is expected to be found:



The address of this block must be given in ECB word 1.
ECB2 specifies the length of the character string to be searched for, boundary words excluded. Maximum length is 128 characters, minimum length is 1 character.
ECB5 is not significant, but on successful completion of the search this word will contain the address of the sector containing the key.

WRITE SECTOR AND VERIFY (/3F)

Flexible Disc

With this order any sector on a flexible disc can be written at random. After the record is written, the Flexible Disc Control Unit uses the following disc revolution to read and verify the written data.
The requested length, as expressed in characters in ECB2 determines the record length (1, 2, 3 or 4 sectors). The length must be an even number with a maximum of 512 characters. If an odd number is given (2n+1), the driver transmits an even number of characters (2n).
ECB5 must contain the absolute sector number (0 - 2001) of the sector to be written. If more than one sector is to be written (128 requested length 512), only the number of the first sector must be given.

The P817 driver provides the interface between the DRTM and a P817 video display terminal, connected to the system via a data communication control unit (AMA8A, AMA8C, ALCU2 or ALCU4). The driver must be regarded as a user-written driver, i.e. it must be linked to the monitor as user interrupt routine during system generation.

P817 VIDEO DISPLAY TERMINAL

The P817 video display terminal is available in two types: the P817-001, which is TTY compatible, operates in character mode, and has a 64 character set (upper case only), and the P817-002, which can operate in character mode or block mode, and has a 96 character set (upper/lower case). Both types have a 12 inch screen, providing 24 lines of 80 characters.
The P817-001 can be extended with options to provide a 96 character set (upper/lower case), display attributes (blinking, blanking, underlining and low intensity, and hard copy facility via an auxiliary device. For the P817-002 options are available for display attributes, hard copy via an auxiliary device, and split screen operation.
The options, including the roll-up function can be selected by hardware straps; in P817-002 terminals with split screen option some strap selection can be changed via the software.

Character/Block Mode

In character mode, each character is transmitted to the system when it is entered on the keyboard. In block mode, the terminal transmits the line in which the cursor is present when the ETB key is pressed, or all lines up to and including the one in which the cursor is present, when the ETB key is pressed together with the Shift key.

Display Attributes

Characters can be displayed with or without display attributes. The attributes are: blinking (characters are displayed/not displayed at a rate of 1.5 Hz), blanking (characters are replaced by a blank on the screen, e.g.passwords), underlining (a continuous line underneath the characters), and low intensity (characters displayed at 2/3

of normal intensity). In P817-002 terminals, characters displayed with the low intensity attribute are also protected against overwriting.

Split Screen Operation

In P817-002 terminals with split screen option, the screen area can be divided into two independent parts; the number of lines for the upper screen part can be specified by the user. All control functions not specifying an explicit line number, apply to the screen part in which the cursor is present; control functions specifying an explicit line number apply to that line.

Hard Copy Facility

If a hard copy device is connected to the display terminal, one or more lines from the screen can be printed. The user can request a copy of the line in which the cursor is present, or a copy of the screen image from the first character position on the first line, up to the current cursor position.

I/O REQUESTS

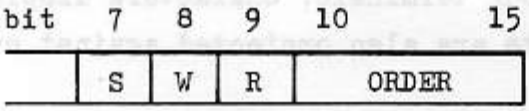
A user program can perform I/O operations via the monitor by means of LKM-requests. Data transfer requests take the same form as the standard I/O-requests for other devices; for special P817 functions the function code and parameters are specified in a separate Function Control Block.

Calling Sequence

The calling sequence for all P817 functions is

```
LDK     A7, CODE
LDKL    A8, ECBADR
LKM
DATA    1
```

Register A7 is loaded with CODE, specifying details of the I/O function as follows:



S, W and R specify the mode of operation:

S=1: (W must also be 1): used by a swappable program; this program can then be swapped out immediately when its swap event count value becomes zero.

W=1: the requesting program waits for the completion of the I/O operation

W=0: the requesting program does not wait for completion of the I/O operation; the program will issue a Wait request later on, for synchronization

R=1: the program itself will process any abnormal condition concerning the requested operation. The system will return the hardware status in ECB word 4 (see below). No retry is possible.

R=0: any abnormal conditions will be processed by the system. The software status is returned in ECB word 4.

ORDER specifies which I/O function is required, by giving one of the following values:

/01: receive (read); characters are read until the required length is reached (see description of ECB below)

/05: send (write); characters are sent until the required length is reached

/12: disconnect; the terminal is disconnected from the line, the screen is cleared, and the cursor is put in the 'home' position

/13: connect; the terminal is connected to the line, the screen is cleared, and the cursor is put in the 'home' position

/1A: wait for incoming call; if the terminal is connected to a switched line, the program is put in wait state until data are received from the terminal; if the terminal is not connected to a switched line, this order is treated as an illegal order.

ECBADR is loaded into register A8, specifying the address of the Event Block.

Event Control Block

The layout of the ECB is:

Y/X	EVENT CHARACTER	LINE CODE	WORD 0
X	BUFFER ADDRESS		WORD 1
X	REQUIRED LENGTH		WORD 2
Y	EFFECTIVE LENGTH		WORD 3
Y	STATUS WORD		WORD 4
X	TIME OUT VALUE		WORD 5
X	FCB ADDRESS		WORD 6

X: these words must be filled by the user

Y: these words are filled by the monitor

WORD 0: event character / line code

bit 0 : if '1', end of operation has occurred for this ECB

bit 1-7 : not used

bit 8-15: line code (file code of the required line)

WORD 1: address of the I/O buffer area

WORD 2: required length, in characters

WORD 3: number of characters which have been transferred

WORD 4: status word

bit 0 = 0 no error

= 1 error

bit 1 = 0 hardware status in bit 2-15

= 1 software status in bit 2-15

hardware status:

bit 2-4 not used

bit 5-7 line number (AMA 8 only)

bit 8 break detection

bit 9 calling indicator

bit 10 end of carrier detection

bit 11 time out error, or I/O command not accepted

bit 12 time out elapsed

bit 13 parity error

bit 14 throughput error
bit 15 modem not operable

software status:
bit 2-6 not used
bit 7 illegal FCB code
bit 8 device busy
bit 9 address not required
bit 10 illegal function specification
bit 11 illegal order
bit 12 illegal buffer size
bit 13 illegal buffer address
bit 14 device not attached
bit 15 illegal file code

WORD 5: time out value
the time out value specifies, in units of o.1 seconds, the maximum waiting time between the I/O request and the setting of bit 0 in the event character. If the time out value is $\leq$ 0, the monitor does not check the time out.

WORD 6: FCB (Function Control Block) address

Function Control Block

The Function Control Block is a 4-word block, containing a function control code and (where applicable) function parameters. The layout of the FCB is:

WORD 0	function code
WORD 1	parameter 1
WORD 2	parameter 2
WORD 3	parameter 3

WORD 0 : function code (hexadecimal)
if =0, a normal write operation (order /05) is carried out

WORD 1-3: function parameters, only used if applicable for the specified function. (see note on next page)

Note: Function parameters must be specified in ASCII code (e.g. decimal 12 must be specified as /3132, not as /C)

FUNCTION CODES

The table below shows for all P817 function codes:

- function name
- function code
- number of parameters (indicated by 'X', or by the fixed parameter value)
- the P817 type for which the function is available

For all function codes, order /05 (send) must be specified in the I/O request.

function name	code	p1	p2	p3	P817 -001	P817 -002
bell	07				x	x
backspace	08				x	x
line feed	0A				x	x
cursor home	0B				x	x
clear screen	0C				x	x
cursor return	0D				x	x
disable manual input	60				x	x
enable manual input	62				x	x
media copy page	69	0			x	x
media copy line	69	6			x	x
set attributes	30	x			x	x
reset attributes	31	x			x	x
cursor position	48	x	x		x	x
cursor horizontal tabulation	49	x				x
cursor horizontal back tabulation	6A	x				x
cursor forward	43	x				x
cursor backward	44	x				x

function name	code	p1	p2	p3	P817 -001	P817 -002
cursor up	41	x				x
cursor down	42	x				x
insert line	4C	x				x
delete line	4D	x				x
erase	4A	x				x
fast output	70	x				x
write line	71	x	x	x		x
read line	72	x	x	x		x
write straps	34	x	x	x		x
set indicator	32	x				x
reset indicator	33	x				x

Function Description

The description of the P817 control functions is divided in two parts: the functions available for P817-001 and P817-002, and the functions available for P817-002 only. With each function, the relevant words of the Function Control Block are given.

Common Functions P817-001/002

FCB — function name / function description

/07 — BELL

The buzzer in the display unit sounds for 0.1 seconds

/08 — BACKSPACE

The cursor is moved one position to the left. If the cursor already is on the first character position of the first line, nothing happens; if the cursor was at the first position of any other line, it is moved to last position on the previous line.

/0A

LINE FEED

The cursor is moved to the same character position on the next line. If the cursor already is on the bottom line and the roll up function is not wired, nothing happens.
If the roll up function is available, all lines move one position up, and the bottom line is filled with null-characters.

/0B

CURSOR HOME

The cursor is moved to the first character position on the top line.

/0C

CLEAR SCREEN

The screen image is cleared, and the cursor is moved to the cursor home position. The Keyboard becomes unlocked (see function EMI below).

/0D

CURSOR RETURN

The cursor is moved to the first character position of the current line.

/60

DISABLE MANUAL INPUT

The keyboard is locked, and all keyboard input is ignored (except for the Clear Screen Key). If a key is pressed, the buzzer sounds for 0.1 second.

/62

ENABLE MANUAL INPUT

The keyboard is unlocked, and the Disable Manual Input state is reset.

/69
/3330

MEDIA COPY (PAGE)

The contents of the screen, from the first character on the top line up to and including the current cursor position, are copied on the auxiliary device. At the beginning of each line 'Carriage Return/Line Feed' is generated for the auxiliary device.

/69
/3336

MEDIA COPY (LINE)

The line on which the cursor is present is copied on the auxiliary device, from the first character position of the line up to and including the current cursor position. At the beginning of each line 'Carriage Return/Line Feed' is generated for the auxiliary device.

Note: After the execution of the Media Copy function, the terminal returns '0' to the first parameter word if the copy is ready, or '3' if the copy is not ready (e.g. device inoperable).

/30
/XX

SET ATTRIBUTES

The SAT function sets the format attributes, which control the way in which data are displayed on the screen. The format attributes are:

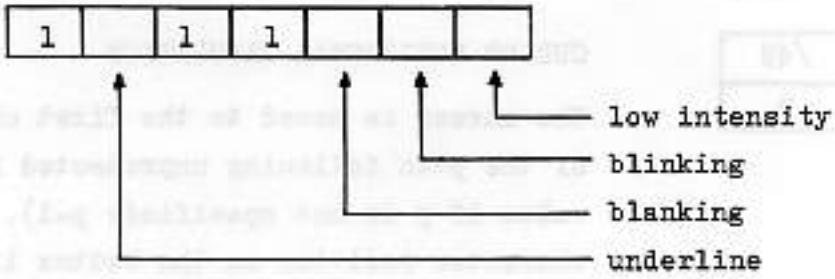
underline - a continues line under all characters

blinking - characters displayed/not displayed at a rate of ca. 1.5 Hz

low intensity - characters are displayed at about 2/3 of normal intensity (in P817-002 terminals operating in block mode, these characters are also protected against overwriting from the keyboard, and are not transmitted after a transmit command)

blanking - characters are stored in the screen memory, but not displayed on the screen

The required attributes are specified by /XX as the hexadecimal value of the 7-bit character shown below:



A '0' bit sets the required attribute.

/31
/XX

RESET ATTRIBUTES

The RAT function resets the format attributes, specified by '0' in /XX in the same way as described for the SAT function.

Note: all format attributes are reset automatically for null codes resulting from the Clear Screen, Insert Line and Delete Line function or keyboard entry, and for the line filled with null codes, which appears on the screen after a roll up.

/48
p
n

CURSOR POSITION

The cursor is moved to the p-th character position on the n-th line. If p or n exceeds its maximum value (i.e. p>80 or n>24), it is assumed to be 0.

Control Functions for P817-002

The following control functions are available only for P817-002 terminals:

/49
p

CURSOR HORIZONTAL TABULATION

The cursor is moved to the first character position of the p-th following unprotected field (default value if p is not specified: p=1), or to the last character position on the bottom line, whichever comes first.
If p=0, the function request is ignored.

/6A
p

CURSOR HORIZONTAL BACK TABULATION

If p=1, the cursor is moved to the first position of the current unprotected field (or the first position in the preceding unprotected field if it already was in the first position of the current unprotected field).
If p>1, the cursor is moved to the first position of the p-th preceding unprotected field (default value: p=1), or to the home position, whichever comes first.
If p=0, the function request is ignored.

/43
p

CURSOR FORWARD

The cursor is moved to the p-th next character position, or to the end of the line, whichever comes first.
Default value p=1; if p=0 the function request is ignored.

/44
p

CURSOR BACK

The cursor is moved to the p-th preceding character position, or to the first character position of the line, whichever comes first.

Default value p=1. If p=0, the function request is ignored.

/41
p

CURSOR UP

The cursor is moved p lines up in the same character position, or to the same character position on the top line, whichever comes first.
Default value p=1. If p=0, the function request is ignored.

/42
p

CURSOR DOWN

The cursor is moved p lines down in the same character position.
If the p-th line is beyond the bottom line, the cursor remains on the bottom line; if the roll up function is strapped, roll ups are executed until the p-th line becomes the bottom line.
Default value p=1; if p=0 the function request is ignored.

/4C
p

INSERT LINE

p lines are inserted at the line on which the cursor is present. The contents of the current line and all following lines are shifted p lines down, and the lower p lines are lost.
Default value p=1; if p=0 the function request is ignored.

/4D
p

DELETE LINE

The p lines from the current cursor position down are deleted; all following lines move p lines up, and p lines with null-characters are added at the bottom of the screen.
Default value p=1; if p=0 the function request is ignored.

/70
p

FAST OUTPUT

The first character from the I/O buffer is repeated p times from the current cursor position, or up to the end of the line, whichever comes first. The cursor remains in it's old position. If the specified character is non-dispalyable (hexadecimal value </20 or =/7F) p white squares are displayed.
Note: for this function, required length =1 must be specified in the ECB.

/71
p
n
x

WRITE LINE

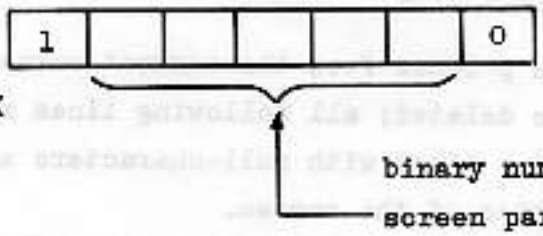
The string of x characters from the I/O buffer is written on line n, starting from character position p.

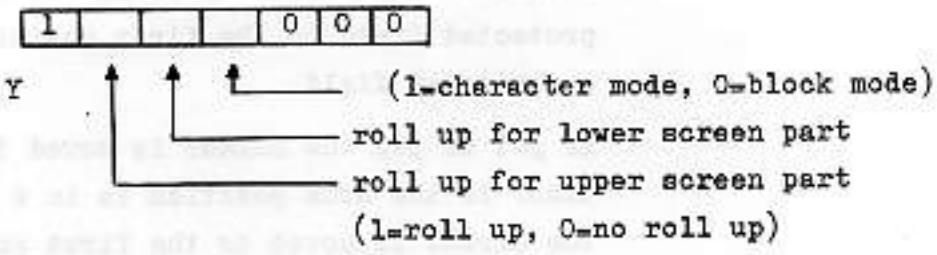
If the string is longer than the available line length, all remaining characters are written on the last character position of the line.
If x=0, it is assumed to be 1. If one of the parameters p or n is not specified, specified as 0, or too high (p>80 or n>24), it is assumed to be 1.
All non-displayable characters in the string are replaced by white squares.

/34
x
y

WRITE STRAPS

The WST function is used to change the strap-selectable split screen function. The new function specification is contained in hexadecimal code in the characters x and y, of which the 7-bit binary value is shown below:



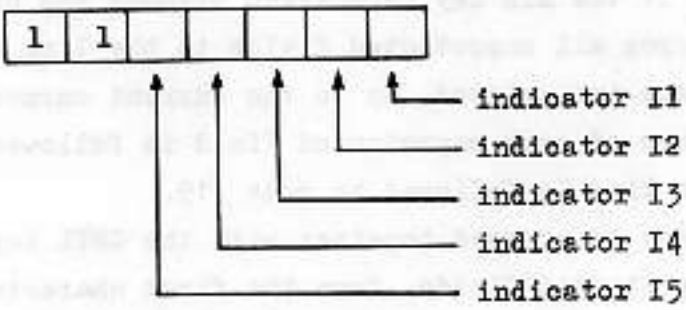


If X or Y is specified with an illegal value, the function request is ignored.

/32
x

SET INDICATOR

The indicators specified by x are lit. The binary format of x is



A '0' bit specifies that the indicator must be lit.

/33
x

RESET INDICATOR

The indicators, specified by a '0' bit in x, in the format described for the SIND function are extinguished.

/4A

ERASE IN DISPLAY

All unprotected characters in the screen area specified by p are replaced by null-characters.
p=0 specifies the screen area from the current cursor position to the end of the screen.
p=1 specifies the screen area from the first character on the top line up to the current cursor position

p=2 specifies the complete screen.

If p=0 the cursor remains in the current unprotected field, or the cursor is moved out of the current protected field to the first position in the first unprotected field.

If p=1 or p=2 the cursor is moved to the home position. If the home position is in a protected field, the cursor is moved to the first position in the next unprotected field.

KEYBOARD FUNCTIONS

Transmit Line / Transmit Page

In P817-002 terminals, the ETB key is used to transmit a block of characters. If the ETB key is pressed without the CNTL key, all characters from all unprotected fields in the line where the cursor is present are transmitted, up to the current cursor position. The last character of each unprotected field is followed by code /09, the complete line is followed by code /19.
If the ETB key is pressed together with the CNTL key, all characters from all unprotected fields, from the first character position on the first line, up to the current cursor position, are transmitted. However, if the ETB key is pressed when the cursor is on the last position of a line or an unprotected field, and the character in that position is the last typed in character, that character is also transmitted. The last character of each unprotected field is followed by code /09, each line is followed by the codes for Cursor Return (/0D) and Line Feed (/0A), and the completely transmitted page is followed by the code /19.
To receive a block of characters, the user program must issue an I/O request with order /01 (receive). The number of characters in the block must be equal to the required length specified in the ECB.

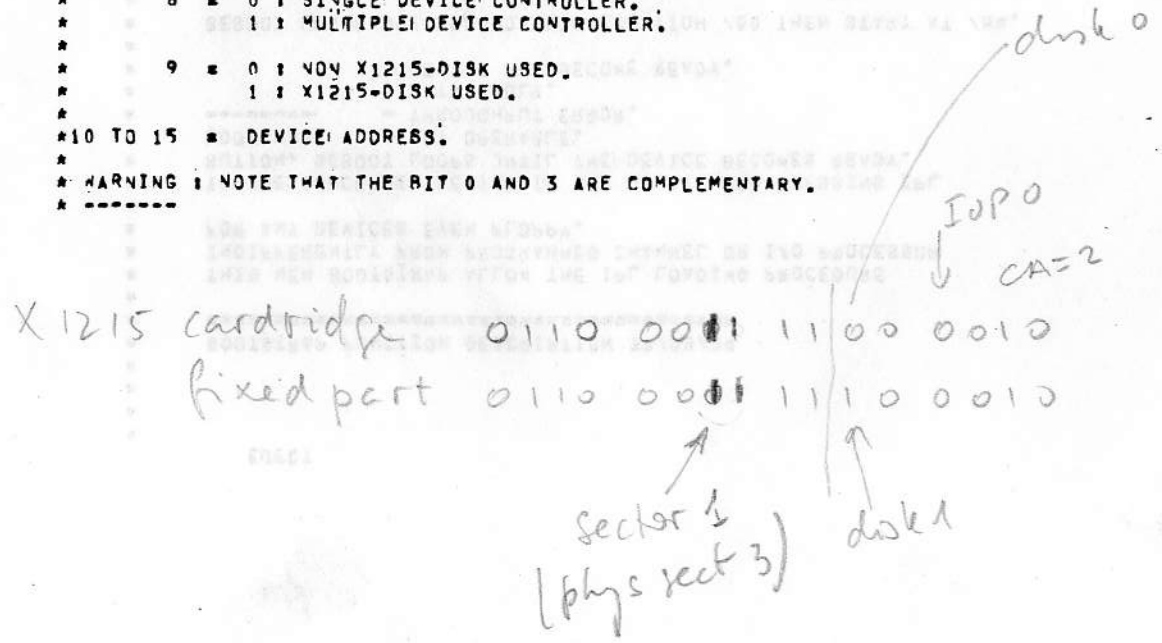
Program Keys

If one of the program keys on the keyboard of the P817-002 is pressed, a program key report is sent to FCB parameter word 1, with value /303X if key X is pressed without the CNTL key, or with value /313X if key X is pressed together with the CNTL key.

Appendix D Control Unit Status Word Configuration

Bit	Description	ASR	PTR	PTP	S.CU	CR	LP	CAS	MT	DK1215	DK40MB	DKFL	SCUZ	SALCU	HDLC
15	Not operable	X	X	X	X	X	X	X	X	X	X	X		X	X
14	Throughput error	X	X		Xin	X		X	X	X	X			X	X
13	Parity error				X			X			X		X	X	
	Data fault								X	X		X			
	FCS error														X
12	Incorrect length					X		X	X	X	X	X			X
11	'Break detection'				X										
	Program error							X	X	X		X			
10	End of tape		X					X	X						
	Tape low			X											
	End of reception				X										
	Hopp./Stack. empty/full					X									
	End of carr. detection													X	X
	Retry											X			
9	Calling indicator													X	X
	Dev/tape/drive no.							X	X	X	X	X			
8	Dev/tape/drive no.							X	X	X		X			
	Device ready				X										
	Break detection												X	X	
7	1=A side/0=B side							X							
	4–7 No. of true data bits														X
6	Write protected							X	X			X			
	Seek error									X	X				
	4–7 No. of true data bits														X
	Begin of tape/load point							X	X						
5	Seek completed									X					
	4–7 No. of true data bits														X
4	Record not found							X			X	X			
	Del. data addr. mark											X			
	4–7 No. of true data bits														X
3	Tape mark detection							X							
	File mark detection								X						
	Idle detect														X
2	Rewind								X						
	Abort detect														X
	Flag bad data track										X				
	Key not found											X			
1	Unit rdy after not rdy							X	X	X	X	X			
0															

```
00000                IDENT   BEBOOT                                  ASM 04
00001  *
00002  *
00003  *       MEANING OF THE KEYS DISPLAYED ON THE CONTROL PANNEL
00004  *       ***************************************************
00005  *
00006  *
00007  *   BIT  0  =  0 : CHARACTER EXCHANGE ON PROGRAMMED CHANNEL.
00008  *              1 : WORD       "      "    "       "
00009  *
00010  *         1  =  0 : IPLI FROM A NON-DISK DEVICE.
00011  *              1 : IPLI FROM DISK.
00012  *
00013  *         2  =  0 : FIXED HEADS DISK.
00014  *              1 : MOVING HEADS DISK.
00015  *
00016  *         3  =  0 : DEVICE ON I/O PROCESSOR (MX).
00017  *              1 : DEVICE ON PROGRAMMED CHANNEL (CP).
00018  *
00019  *   4 TO 7  =  BOU LINES (USED TO QUALIFY AN INSTRUCTION I/O).
00020  *
00021  *         8  =  0 : SINGLE DEVICE CONTROLLER.
00022  *              1 : MULTIPLE DEVICE CONTROLLER.
00023  *
00024  *         9  =  0 : NON X1215-DISK USED.
00025  *              1 : X1215-DISK USED.
00026  *
00027  *10 TO 15  =  DEVICE ADDRESS.
00028  *
00029  * WARNING : NOTE THAT THE BIT 0 AND 3 ARE COMPLEMENTARY.
00030  * -------
```



```
00031                   EJECT
00032          *
00033          *
00034          *
00035          *
00036          *     BOOTSTRAP FUNCTION DESCRIPTION 77/08/24
00037          *     =======================================
00038          *
00039          *     THIS NEW BOOTSTRAP ALLOW THE IPL LOADING PROCEDURE
00040          *     INDIFFERENTLY FROM PROGRAMMED CHANNEL OR I/O PROCESSOR
00041          *     FOR ANY DEVICES EVEN FLOPPY.
00042          *
00043          *     IF THE CONCERNED DEVICE IS NOT READY WHEN PRESSING IPL
00044          *     BUTTON, BEBOOT LOOPS UNTIL THE DEVICE BECOMES READY.
00045          *     LOOP MASK :    = NOT OPERABLE.
00046          *     --------       = THROUGHPUT ERROR.
00047          *                    = DATA FAULT.
00048          *                    = DEVICE HAS BECOME READY.
00049          *
00050          *     BEBOOT LOADS ONE RECORD ONTO LOCATION /80 THEN START AT /84.
00051          *
00052          *     THE RECORD IS THE SECTOR NUMBER 1 ON DISK, OR 254 CHARACTERS
00053          *     FROM THE INPUT DEVICE (LEADING NULL CHAR. IS IGNORED) OR SECTOR
00054          *     NUMBER 0 IF FLOPPY DEVICE USED.
00055          *
```

```
00056                    EJECT
00057          *
00058          *
00059          *
00060          *
00061          *              USED REGISTERS.
00062          *              ***************
00063          *
00064          *     A1      =:     BOU LINES.
00065          *     A2      =:     A(INR INST),
00066          *     A3      =:     A(CIO INST),
00067          *     A4      =:     A(SST INST).
00068          *     A5      =:     IF MX : FIRST WORD TO BE SENT TO EXTERNAL REGISTER.
00069          *                    IF CP : CHARACTER COUNT.
00070          *     A6      =:     IF MX : SECOND WORD FOR THE MULTIPLEX.
00071          *                    IF CP : A (NEXT CHAR. TO BE LOADED).
00072          *     A7      =:     )
00073          *     A8      =:     )  WORK REGISTERS.
00074          *     A9      =:     )
00075          *     A10 TO A14 : NOT USED.
00076          *     A15     =:     CONTAINS THE KEYS VALUE FROM THE CONTROL PANNEL.
00077          *
00078          *
00079          * WARNING : A1 MUST NOT BE DESTROYED IF BOOT IS CALLED AGAIN.
00080          * --------   A3 MUST BE RESTORED          "       "       "
00081          *            A5 CHAR. COUNT INITIALIZED TO 254 AND DECREMENTED.
00082          *            A6 ON CP, INITIALIZED TO /80 AND INCREMENTED.
00083          *
00084          *
00085          *
```

```
00086                           EJECT
00087                           ORG     0
00088                   *
00089                   * BASE REGISTERS INITIALIZATION.
00090                   *
00091  0000  0200  F    BOOT    LDK     A2,INR          A2 = A(INR INST),
00092  0002  0300  F            LDK     A3,CIO          A3 = A(CIO INST),
00093  0004  0400  F            LDK     A4,SST          A4 = A(SST INST).
00094                   *
00095                   * GET DEVICE ADDRESS TO INITIALIZE THE I/O COMMANDS.
00096                   *
00097  0006  861E               LDR     A6,A15          A6 = KEYS VALUE.
00098  0008  263F               ANK     A6,/3F          KEEP JUST THE DEVICE ADDRESS.
00099  000A  AE29               ORRS    A6,A2           INIT INR INST WITH D.A.
00100  000C  AE2D               ORRS    A6,A3           INIT CIO INST..........
00101  000E  AE41               ORS     A6,HIO          INIT HIO..............
       0010  0000  F
00102                   *
00103                   * GET CONTROLLER TYPE AND ADDRESS.
00104                   *
00105  0012  871E               LDR     A7,A15          MULTI OR SINGLE
00106  0014  3FC8               SLC     A7,8            DEVICE CONTROLLER ?
00107  0016  5600  F            RF(6)   INIT20          IF SINGLE --> BRANCH,
00108  0018  260F               ANK     A6,/F           KEEP THE DEVICE CONT. ADDRESS (CA).
00109                   *
00110                   * INITIALIZE THE WER INSTRUCTIONS AND SST ONE.
00111                   *
00112  001A  AE31       INIT20  ORRS    A6,A4           INIT SST INST WITH D.A OR C.A
00113  001C  3E41               SLL     A6,1            COMPUTE EXTERNAL REGISTER ADDRESS.
00114  001E  AE41               ORS     A6,WER1         UPDATE WER INSTRUCTION
       0020  0000  F
00115  0022  AE41               ORS     A6,WER2         WITH THIS EXTERNAL REG.
       0024  0000  F
00116  0026  811C               LDR     A1,A7           A1 = BOU LINES.
00117  0028  0550               LDK     A5,/50          BLOCK LENGTH TO BE LOADED.
00118  002A  0680               LDK     A6,/80          IN LOCATION /80.......
00119                   *
00120                   * CHECK DEVICE TYPE.
00121                   *
00122  002C  3FE7               SRC     A7,7            IS THE DEVICE A DISK OR NOT ?
00123  002E  5600  F            RF(6)   WER1            NO.
00124  0030  3FC1               SLC     A7,1            IS IT A FIXED HEADS DISK ?
00125  0032  5600  F            RF(6)   NOSEEK          YES (NO SEEK TO ZERO).
00126  0034  0103               LDK     A1,3            EXECUTE A SEEK TO ZERO OPERATION
00127  0036  41C0       CIO     CIO     A1,1,0          ON MOVING HEADS DISK.
```

```
00128                          EJECT
00129                    *
00130                    * PREPARE THE NEXT LOGICAL I/O FOR A DISK.
00131                    *
00132  0038  811E        NOSEEK  LDR     A1,A15      COMPUTE THE
00133  003A  3966               SRL     A1,6        SECTOR
00134  003C  213C               ANK     A1,/3C      NUMBER.
00135  003E  6520               LDKL    A5,/80CD    FIRST WORD ON MX FOR A DISK.
       0040  80CD
00136                    *
00137                    * EXECUTE THE LOGICAL I/O FOR ANY DEVICE ON MX OR CP INDIFFERENTLY.
00138                    *
00139  0042  7500        WER1    WER     A5,0        FOR ANY DEVICE (DISC OR NOT) EXECUTE
00140  0044  7601        WER2    WER     A6,1        THE WER WHATEVER THE CHANNEL IS.
00141  0046  F031               EXR*    A0          EXECUTE A SST. . . . .
00142  0048  F02D               EXR*    A3          "   CIO.........
00143  004A  5C06               RB(4)   *-4         REFUSED ! TRY AGAIN.
00144                    *
00145                    * CHECK CHANNEL WANTED AND TAKE CORRESPONDING ACTION.
00146                    *
00147  004C  8194               LDR     A9,A5               A9 = NBR OF CHAR TO READ.
00148  004E  871E        MX##CP  LDR     A7,A15              IS THE DEVICE CONNECTED
00149  0050  3F43               SLL     A7,3        ON MULTIPLEX CHANNEL ?
00150  0052  5600  F            RF(6)   SST         YES. SEND A SST.
00151                    *
00152                    * INPUT ONE CHARACTER FROM THE DEVICE.
00153                    *
00154  0054  4F00        INR     INR     A7,0,0      READ ONE CHAR. IN A7.
00155  0056  5400  F            RF(4)   SST         REFUSED ! SEND A SST TO KNOW WHY.
00156  0058  E994               CWR     A9,A5       LEADING CHAR. ?
00157  005A  5400  F            RF(4)   INR10       NO.
00158  005C  871C               LDR     A7,A7       YES. NO CHECK WETHER
00159  005E  580C               RB(0)   INR         WORD OR CHARACTER EXCHANGE.
00160  0060  879E        INR10   LDR     A15,A15     IS IT IN CHARACTER MODE ?
00161  0062  5600  F            RF(6)   STORE       YES.
00162  0064  8739               STR     A7,A6       STORE THE WORD JUST READ ON CP.
00163  0066  1601               ADK     A6,1                USE SAME ROUTINE (WORD OR CHAR. ON CP)
00164                    *
00165                    * STORE CHARACTER JUST READ AND CHECK END OF TRANSFER.
00166                    *
00167  0068  E739        STORE   SCR     A7,A6       STORE ONE CHAR. TO /80 + (A6).
00168  006A  1601               ADK     A6,1        NEXT LOCATION TO BE ERASED.
00169  006C  1D01               SUK     A5,1        IS THERE ANOTHER CHAR. TO READ ?
00170  006E  5C1C               RB(4)   INR         YES. LOOP ON INPUT.
00171                    *
00172                    * END OF TRANSFER - EXECUTE A CIO HALT.
00173                    *
00174  0070  4180        HIO     CIO     A1,0,0      CIO HALT TO STOP THE EXCHANGE.
```

```
00175                          EJECT
00176                    *
00177                    * EXECUTE A SST AND ANALYZE THE STATUS TO LOOP FLE : A READY IS SET.
00178                    *
00179  0072  9FC0       SST      SST     A7,0          SEND STATUS.
00180  0074  5C28               RB(4)   MX::CP                REFUSED ! THIS ALLOW TO HANDLE THE
00181                    *                                     BOTH CASES (WAIT ON CP OR MX).
00182  0076  A720               ANKL    A7,/8007      TEST : BIT 1 = DEVICE HAS BECOME READY.
       0078  8007
00183                    *                                     13 = DATA FAULT.
00184                    *                                     14 = THROUGHPUT ERROR.
00185                    *                                     15 = NOT OPERABLE.
00186  007A  0C00               AB(4)   /00           ONE OF THESE CONDITIONS THEN
00187                    *                             START BOOTSTRAP AGAIN FROM THE
00188                    *                             VERY BEGINNING.
00189  007C  0F84               AB      /84           NORMAL END BRANCH TO IPL ROOT.
00190                            END     BOOT
```

SYMBOL TABLE

```
BOOT    0000  A   INR     0054  A   CIO     0036  A   SST     0072  A
MIO     0070  A   INIT20  001A  A   WER1    0042  A   WER2    0044  A
NOSEEK  0038  A   MX::CP  004E  A   INR10   0060  A   STORE   0068  A

       ASS.ERR.    00000
:EOF
PROG ELAPSED TIME:  00H-00M-00S-000MS-
```

Command	Meaning	Page
AS	Assign a file code	1-96
BF	Space file backwards	1-99
BR	Space record backwards	1-99
CN	Connect a program to a software level	1-91
CT	Connect a program to a timer	1-92
DF	Delete a file	1-99
DL	Delete a file code	1-97
DN	Disconnect a program from a level	1-93
DT	Disconnect a program from a timer	1-93
EN	End of commands	1-99
FF	Space file forward	1-98
FR	Space record forward	1-98
KF	Keep file	1-95
LD	Load a memory resident program	1-88
Magnetic Tape Control Commands		1-98
RO	Declare a read only program	1-89
RW	Rewind tape	1-98
SC	Set clock	1-94
SD	Set date	1-94
ST	Start a program	1-94
SW	Declare a swappable program	1-90
TS	Define time slice	1-91
UN	Unload tape	1-98
WF	Write End-Of-File mark	1-99
WS	Write End-Of-Segment mark	1-99
WV	Write End-Of-Volume mark	1-99

Message	Meaning	Page
CC	Request SCL	1-102
CR	Correction	1-102
DD	Dump Disc	1-102
DM	Dump Memory	1-102
HD	Halt Dump	1-103
HT	Stop CPU	1-103
RD	Release Device	1-103
RY	Retry I/O Operation	1-103
WM	Write Memory	1-104

Request	LKM	Page
Input/Output	1	1-107
Wait for an Event	2	1-112
Exit	3	1-114
Get Buffer	4	1-115
Release Buffer	5	1-117
Connect Program to Timer	10	1-118
Disconnect Program from Timer	11	1-120
Activate a Program	12	1-121
Switch Inside a Software Level	13	1-123
Attach Device to Program	14	1-124
Detach Device from Program	15	1-126
Get Time	17	1-127
Set an Event	18	1-129
Connect Program to Software Level	20	1-130
Disconnect Program from Level	21	1-131
Wait for a Given Time	22	1-132
Assign a File Code	23	1-134
Delete a File Code	24	1-137
Read Unsolicited Operator Message	25	1-138
Cancel Unsolicited Message	26	1-140

Appendix G Device Addresses

It is recommended to give device addresses according to the following table:

ADDRESS FIELD
(6 bits)

3 leftmost bits / 3 rightmost bits	/00	/08	/10	/18	/20	/28	/30	/38
/00		(SLCU2)	System Consol	(ALCU2/4)	PTR		PTP	
/01	Disc*: X1216 X1215 Floppy	(SLCU2)	Disc*: X1216 X1215 Floppy	(ALCU2/4)	Disc*: X1216 X1215 Floppy		Disc*: X1216 X1215 Floppy	
/02	Disc*: X1215 X1216 Floppy	(SLCU4)	Disc*: X1215 X1216 Floppy	(ALCU4)	Disc*: X1215 X1216 Floppy		Disc*: X1215 X1216 Floppy	
/03	Disc*: Floppy X1216 X1215	(SLCU4)	Disc*: Floppy X1216 X1215	(ALCU4)	Disc*: Floppy X1216 X1215		Disc*: Floppy X1216 X1215	
/04	MT*	(SLCU4)	MT*		MT*		MT*	
/05	TK*	(SLCU4)	TK*		TK*			
/06	CR	(AMAB) Input	Disc*: CDC40 CDC80				Disc*: CDC40 CDC80	
/07	LP	(AMAB) Output	Disc*: CDC80 CDC40				Disc*: CDC80 CDC40	

Explanatory Notes:

- All addresses on one line marked with an asterisk are reserved for a multiple device control unit, even if only one device is attached.
- Where several disc types are given for one address, they are listed in order of priority for that address. Only the first one is to be considered standard.
- Columns 1, 3 and 5 are reserved for Data Communication control units; because of the wide variety of configurations and exchange rates, no no fixed allocations can be defined and no standard is assumed. (cont.)

- Columns 1, 3 and 5 are reserved for Data Communication control units; because of the wide variety of configurations and exchange rates, no no fixed allocations can be defined and no standard is assumed.
- For Data Communication control units the address is of importance only when using the I/O Processor for:
  - throughput possibility
  - optimizing the number of IOPs.
- In some configurations, the first I/O Processor may be used to connect Data Communication control units at addresses which are free; this is in contradiction with the recommended standard, but it saves one IOP.
- An example of a device address on 6 bits is: /38 = 111000.