APPENDICES

To standardize the system generation procedure for all systems, a set of system generation processors has been developed which provides great flexeibility, 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 Operating Monitor these are:

- GENMON, a generation monitor used only during the system generation process to run the generation processors.
- DOMGEN, 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 (DOMLIB), to select the ones requested during the DOMGEN phase and link them with the tables generated during DOMGEN.
- 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 Data Communication User Manual
In the following paragraphs, the whole set of operations necessary for the generation of a DOM 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 DOM 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-oriemted, the user receives 25 tapes, containing:
- IPL + GENMON
- DOMGEN
- PREMDK
- GENLKE
- DOMLIB (the DOM Standard Library)
- DISLOD
- one tape for each of the monitor segments and system processors:
  CCI, CSEG1, CSEG2, CSEG3, CSEG4, CSEG5, LED (Line Editor), CSEG7, CSEG8, CSEG9, CSEGA, CSEGB, CSEGC, CSEGD, CSEGE, ASM (Assembler), LKE (Linkage Editor), DEB (Debugging Package), IPLGEN (IPL Generator).

If the configuration is cassette tape-oriented, the user receives two so called generation cassettes, containing:
- cassette 1: side A: IPL
  GENMON
  DOMGEN

PREMDK
GENLKE
side B: DOMLIB
DISLOD
- cassette 2: side A: monitor segments and system processors:
CCI
CSEG1
CSEG2
CSEG3
CSEG4
CSEG5
LED
CSEG7
CSEG8
CSEG9
CSEGA
CSEGB
CSEGC
CSEGD
CSEGE
ASM
LKE
DEB
IPLGEN
side B: ------

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 your Programmer's Guide, Vol. 1; 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 68 ; 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 processor from either the cassette in drive TK05 or the paper tape reader 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 possibili-

tieswhich 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, which is the case when the user works with paper tape, GENMON outputs the following list on the typewriter, thereby allowing the user to give his own file code assignments (since the GENMON processor is the same for all monitors, some of the file codes given in this list are irrelevant to the generation of a DOM 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: DOMGEN.

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 (PTR, cassette, magnetic tape, 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.

DOMGEN

When GENMON initialization is terminated and the message READY TO LOAD PROGRAMS has been output the user can start the second phase, DOMGEN, the building of the DOM tables. This is done by typing replies to questions output on the typewriter by DOMGEN. From these replies DOMGEN 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 a 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 DOMGEN 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 DOMGEN 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. DOMGEN is loaded from the cassette or paper tape and its identification is output:
  IDENT DOMGEN

- when loading is terminated,
  :EOS
  :EOF
  is 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 DOMGEN according to the replies given on TY10.)

- push the INT button
- on output of M: type in
  ST
- now DOMGEN is started and outputs the following messages 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 blank and a comment field of up to 73 characters.

Error Message: TG03: the first charactersis 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.

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.

REAL TIME CLOCK LEVEL

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

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). DOMGEN 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.
For each of the monitors a different list appears:

IORM (I/O Requests)
WAIT (Wait for Event) } necessary for standard system
EXIT (Exit)
BUFF (Get/Release Buffer): required for FORTRAN
PSE (Pause): required for cassette and magnetic tape handling and for the PSE control command.
ABRT (Abort Control): required for Debug
SLD (load a Segment): } necessary for
DFM (Disc File Management) } standard system

Note: If the user gives, later on in a running program, a request for a function which he has not included at this stage, the value -1 is returned in the A7 register.

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 DOMGEN, 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.

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 GENLKE 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

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: TG03

OCOM OPTIONS

Reply: Specify, if the operator communication package must be included in the system by typing:
Y if it must be included, or:
N if it must not be included.
If the reply was Y, DOMGEN will print, one by one, the operator commands available. By typing Y or N after each one, the user indicates whether he wants it to be included or not.
The following list is output by DOMGEN:

DM (Dump Memory)
MC (Manual Control) (required for magnetic and cassette tape handling)
HD (Halt Dump) (recommended if DM has been selected)
WM (Write into Memory)
RY (Retry Device) } (see Note below)
RD (Release Device) }
AS (Assign a File Code)
AB (Abort a Program)
PS (Pause)
RS (Restart) (necessary with cassette or magnetic tape)

Note: - When the commands RY and RD are not selected, the system will stop when an I/O error occurs which requires operator intervention. In a configuration with only one typewriter these two commands are needed, but if more devices are used, they are highly recommended.
- RS is required if PS or PSE (LKM) has been selected.
- If N has been replied to the question OCOM OPTIONS, the abort option may be selected as an LKM option or under the question ABORT MODULE. (Pressing the INT button will then abort a running program). If, in add-

ition, Y has been replied under DUMP IN ABORT, pressing the INT button during the memory dump output will simulate an HD command.

USER COMMANDS

Reply: Specify any user-made operator commands which must be included in the system, as follows:

```
<CM1>,<ENTRY1>
        :
<CMn>,<ENTRYn>
      END
```

Where CMi consists of 2 characters defining the operator command and ENTRYi specifies the entry point of the module which processes command CMi. END indicates that all commands have been declared or that none are required.

Note: CMi must not be equal to any of the standard operator commands.

Error Message: TG03: syntax error.

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

END indicates that all devices have been declared.

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:

BM: X1215 moving-head disc (model 1) .

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, on 2 hexadecimal digits.
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.

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 Code EF and 01 are used by the system for input/output to/from the operator's typewriter. It cannot be assigned or reassigned by AS command. So, if used, it is necessary to declare at least this file code. All other file codes may be assigned at execution time by means of the operator command AS or the ASG control command.

Some file codes, such as 02, 03, E0, E1, E2 are used by the system. These must also be specified, if any of the standard processors are used.
At least one file code between F0 and FF should be defined for each disc.

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.

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.

Apart from those already defined, the system requires at least five entries, and one for the Line Editor JN command and one for the catalogued procedures, if used.

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.

The system itself requires at least five files, and one for the Line Editor JN command and one for catalogued procedures, if used.

DISK BUFFER POOL

Reply: This message is printed only in case the user has selected the Disc File Management (DFM) option (see under LKM LEVEL). Specify the number of blocking buffers used for blocking and deblocking logical records of sequential files or for the granule table in case of random access files. The system requires at least two and one for the Line Editor JN command and one for the catalogued procedures, if used. (Not more buffers than the maximum number of logical files opened at the same time. (See Data Management)).

DISK ALLOCATION TABLE

Reply: Specify the length, in characters, of the disc allocation table on two hexadecimal digits. (For X1215: /66).

DUMP IN ABORT

Reply: Specify Y if a post-mortem dump is wanted in case of abort, or N if it is not wanted.

MAXIMUM NUMBER OF SEGMENTS

Reply: Specify the maximum number of segments used by any of the programs, on two hexadecimal digits.

Note:
The Control Command Interpreter (CCI) uses 14 segments.

ABORT MODULE (This question is printed only if N was replied for AB under OCOM OPTIONS and ABRT under LKM LEVEL)

Reply: If neither the monitor request 'Keep Control on Abort' nor the operator command AB have been selected the user may include the System Abort Module by typing Y.
If the reply is N, the abort module will not be included. Abort of a user program will, if the module has been selected, result in an output message and the user may enter a new command to run the next job. If the module has not been selected, an abort will stop the machine, through a HLT instruction.

Error Message: TG06: invalid reply.

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:
For P856/P857 the answer is N, for these instructions are included in the instruction set.

After each item the user must type in Y or N to indicate whether he wants that instruction simulation routine included or not.

Note: If the CCI package is to be used, the reply for this question must be Y.

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.
This is 1, or, if scheduled labels are used, 2.

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 G1 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:
  INITIALIZATION OF PREMRK
  NBR OF CYLINDERS = (type in 4 decimal digits giving the number of cylinders on the disc, followed by (LF) (CR) Example: 0203)
NBR OF TRACKS = (type in 4 decimal digits giving the number of tracks per cylinder, followed by (LF) (CR) Example: 0002)
NBR OF SECTORS/TRACK = (type in 4 decimal digits giving the number of sectors per track, followed by (LF) (CR) Example: 0016)
DISK TYPE = (for the X1215, this must be CM followed by (LF) (CR))
DISK UNIT PHYSICAL ADDRESS = (type in two hexadecimal digits giving the physical address of the disc, followed by (LF) (CR) Example: 32)

LABEL = (type in a volume label, consisting of up to 8 characters, followed by (LF) (CR) Example: TONTO)

DATE = (type in 6 decimal characters, separated by delimiters, e.g. spaces or slashes, followed by (LF) (CR) Example: 01/01/84)

PACK NBR = (type in 3 characters giving the number of the disc pack, followed by (LF) (CR) Example: F32)

SYSTEM USERID = (type in up to 8 ASCII characters, specifying the first user of this disc pack, i.e. the user identification of the system. This is to enable the pack to be loaded by DISLOD. It must be followed by (LF) (CR) Example: SILVER) It is mandatory to answer this question.

When the user has answered these questions, PREMDK starts premarking the disc and outputs the messages:

WRITING THE IDENTIFIERS

CHECKING THE IDENTIFIERS

END OF CHECK

NBR OF BAD GRANULES = XXXX

RUN AGAIN? (to this last question the user must type in NO if he wants no other disc premarked; if he types in OK, the above procedure is repeated; after NO PREMDK types out its last message:)

END OF PREMRK

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 DOMGEN are linked with the modules required from the DOM Library and, possibly, user and/or extension libraries to generate the user's monitor.

GENLKE

During this phase the final user monitor is obtained by linking the tables generated under DOMGEN with the monitor modules required from the DOM 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 DOMGEN) 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 system

8 or 4 is used if the monitor is punched on paper tape to indicate whether it must be punched in 4-track or 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 DOMGEN 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 (DOMLIB) and wait for it to be rewound, so that it is positioned correctly for the scanning of the DOM Library.
- in response to the

L:

output by GENLKE, now type in

L

upon which GENLKE will start scanning the DOM 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. The last module to be included must be INIMON, so if GENLKE types

INIMON

after the user has typed in his first U, it is correct.
The GENLKE processor then types out
L:
and the user can type
L
to solve his 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 DOM on the listing device and on the typewriter it outputs monitor 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 (DOMLIB). Wait for it to be rewound.
- put cassette W1 (tables output by DOMGEN) 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 but 7 is assigned now 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 DOMGEN. 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 and 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 DOM Library 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 DOM Library 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 DOMGEN is put on the reader and GENLKE is started with an ST command. Having entered the E: option command, the user types P and the tables are processed.

- then the DOMLIB is put on the tape reader and the user types L after which this library is scanned and the selected modules are output on the punched tape.

- having checked if INIMON 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 DOM has now been generated.

Note:

If modules from other (paper or cassette) tapes beside the DOMLIB 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)
- DOMLIB tape

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

Now the user has generated his Disc Operating Monitor on the device with file code /3.
The only thing that remains to be done now is to store the monitor together with the system processors on the system disc. This is done during the following phase, DISLOD.

DISLOD

DISLOD is the last system generation processor. It is used to record the generated monitor, monitor modules and system processors onto the system disc specified under GENMON, in load module format accepted by the DOM.
The standard object input file code for DISLOD is /E2 (standard 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 DOMLIB, 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 DOM tape must be placed on it, ready to be read
- with cassettes, take cassette G1 from file code /E2 (normally TK05) and replace it by cassette W2 (containing the new DOM) 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 P852
  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 in Pause state, enabling the user to modify an assignment, operate on the cassettes or tapes manually, 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 CCI 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 (CCI segments) and the system processors LED, ASM, LKE, DEB and an IPL generator IPLGEN.
- 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 complete system on disc. If he wants to record any other processors, such as FORTRAN and its Library, on it, this can be done with the relevant CCI commands of the generated Disc Operating Monitor. For this purpose, press the MC button and load the DOM.

System Generation under DOM Control

If the user already has a disc system and a Disc Operating Monitor, he can use this DOM to generate any system : BOS, DOS, BRTM or DRTM. If his DOM includes the Cassette File Management Package, it can also be used to generate a Cassette Operating System.

The necessary elements are on punched tape or cassette tape:
- the XXXGEN processor (BOMGEN, DOMGEN etc.)
- the Monitor Library

These must be recorded and arranged on disc.
For each system to be generated a special user identification must be declared (userid).

Let us assume such a userid is declared as GENUSER. This userid must then include the XXXGEN processor and the monitor library, except for INIMON (+ ASEX for DRTM):
- the tape containing XXXGEN is put on the tape reader
  (if it is on cassette tape, the procedure is:
  - load cassette G1, side A
  - assign file code /E2
  - give the CCI command FFS /E2
    This will position the cassette tape after the tape mark following GENMON, i.e. just before the XXXGEN processor. The monitor library always starts at the beginning of a cassette side.)
- give the CCI command RDO
  followed by LKE
  and KPF /L,XXXGEN
  to store the XXXGEN processor on the disc.
- then the library tape is put on the reader (or cassette)
- give the command RDO for each tape (if there are more tapes or more than one cassette side)
- then the command KPF /O . Keep the last tape on the reader.
- give the command BYE
  followed by the userid SYSTEM to declare a system session
- then the command RDO to read INIMON (+ ASEX for DRTM) as the last

modules and keep them with the command

KPF /0

Note: If systems including extensions such as Data Communication or Cassette File Management must be generated, these extensions must be read at this point:

- give the command BYE
  followed by the userid GENUSER
  followed optionally by RDO <extension> commands to include the extension or user library modules. This is done in this way because the extension may contain modules with the same names as processed during the previous phase. Under GENLKE only the first module encountered would then be taken.
  They are kept by KPF /0 command.
- now we have the XXXGEN processor and the monitor library on disc and we can generate the monitor tables and link them with the required modules. First we must assign the correct file codes (see GENMON: all input devices plus disc sequential file may be used):
- the following commands must be given:

ASG /A,DK,<file name>	assuming that the XXXGEN commands have been stored in a permanent disc file
ASG /B,DK	(a temporary file)
RUN XXXGEN	(object code will be output on /B)
REF /B	(rewind /B, i.e. the generated tables)
RDO /B	(into the /0 file)
LKE M	(there are 2 libraries: a user library with the monitor modules and a system library consisting of INIMON (+ ASEX), so INIMON will always be the last module selected)

KPF /L,<file name>

- now the monitor has been generated and kept in the /L file under <file name>.

Note: If an error occurs during the XXXGEN process, an exit will be made with bit 8 set in register A7, so in batch processing mode an abort will occur.

- to generate a complete system then, a user must be created under which the other modules, e.g. for a DOS system the disc-resident monitor segments and the system processors, must be kept.
- declare a userid, for example DOSGEN
- premark a scratch disc pack and declare a system userid for it, for example WHOME.

- reload the System by means of IPL procedure.
- enter this userid WHOME
- then:

  BYE

  WHOME

  MOV <file name>,/L,GENUSER (see above)

  KPF /L,SUP (there must of course be no module of this name among the system parts to be included.
- the first module will now be catalogued on disc starting from sector address /10.
- then give the command

  SVU DOSGEN,/FO

  to implement all the other system modules and processors.

Note: It is also possible to use Catalogued Procedures for users who must do sysgens regularly. See page 23

Note: When the DOS Linkage Editor is used instead of GENLKE during the linking phase, the addresses given by the DOS LKE in the MAP output must be subtracted by 8 to get the exact memory addresses, whereas the addresses output by GENLKE are correct.

Note: IPLGEN is a DOS module which is required when generating a BOM or BRTM, because the monitor which is then generated on tape must be preceded by an IPL. The sequence of operations in such a case is:

ASG /3,<output device>

RUN IPLGEN

PLD <monitor>

The generated IPL can be used for any device.

DISK 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 and writes identifiers in them and it 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 typewriter and the user can type in his answers:

NBR OF CYLINDERS =
Type in 4 decimal characters, specifying the number of cylinders on the disc, followed by LF CR

NBR OF TRACKS =
Type in 4 decimal characters, giving the number of tracks per cylinder, followed by LF CR

NBR OF SECTORS TRACK =
Type in 4 decimal characters, specifying the number of sectors per track, followed by LF CR

DISK TYPE =
Type in 2 characters, specifying the type of disc unit, and the type of channel used, as follows:

CM: X1215 on I/O processor
FM: fixed-head disc on I/O processor

then type in LF CR

DISK UNIT PHYSICAL ADDRESS =
Type in two hexadecimal characters, followed by LF CR

LABEL =
Type in 8 characters, giving the volume label, followed by LF CR

DATE =
Type in 6 decimal characters, specifying the date, followed by LF CR

PACK NBR =
Type in 3 characters to specify the pack number, followed by LF CR

SYSTEM USERID =
Type in the identification of the system generated, followed by LF CR.

RUN AGAIN?:
Type in NO if no other discs are to be formatted, or OK if another disc must be done. In the second case, the above procedure is repeated.

Example:

```
INITIALISATION OF PREMRK
NBR. OF CYLINDERS = 0203
NBR. OF TRACKS = 0002
NBR. OF SECTORS/TRACK = 0016
DISK TYPE: CM
DISK UNIT PHYSICAL ADDRESS = 02
LABEL = EXAMPLE
DATE = 12/02/74
PACK NBR. = 001
SYSTEM USERID = SAG
- WRITING THE IDENTIFIERS
- CHECKING THE IDENTIFIERS
- END OF CHECK
- NBR. OF BAD GRANULES = 0000

RUN AGAIN?: NO
END OF PREMRK
```

The user replies are given in boldface.

Error Messages:
BAD GRANULE ZERO (granule zero of the pack is bad and therefore this pack is not usable)
NO SYSTEM USER POSSIBLE (granule one of the pack is bad and therefore no system catalogue can be opened).

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 character. 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).

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 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.

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 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.

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
- 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.

Note: (physical files; F0-FF)

If this order is used for a disc file code, the system will return the device name DK or DL (logical files) in ECB word 1 and fill the other words of the ECB with zeros.

OFF LINE (/38)

Magnetic tape:
This order switches the machine off.

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

Appendix D Control Unit Status Word Configuration

Bit	Description	Control unit								
		ASR	CR	c	DM	LP	PTP	PTR	CASS Tape	MT
0	–									
1	has become ready				x				x	x
2	rewinding									x
3	tape mark has been read								x	x
4	no data								x	
5	on cylinder				x					
	beginning of tape								x	
6	seek error				x					
	write unable								x	x
7	A or B side								x	
8	Device Address								x	x
9	Device Address				x				x	x
10	EOT							x	x	x
	tape low						x			
11	Program error				x				x	x
12	incorrect length		x		x				x	x
13	Parity error								x	x
	data fault		x		x					
14	throughput error	x	x		x			x	x	x
15	not operable (only significant bit for TST)	x	x		x	x	x	x	x	x

DM = moving-head disc

```
00000                    IDENT    BEBOOT
00001          *
00002          *
00003          *        DISPLAY THE KEYS AS FOLLOWS:
00004          *
00005          *        BITS     MEANING
00006          *        0=1      IPL LOADED FROM ASR , FORMAT 4*4
00007          *        1=1      DISK,THEN
00008          *        2=1      MOVING HEADS
00009          *        2=0      FIXED HEADS
00010          *        3=1      PROGRAMMED CHANNEL
00011          *        3=0      I/O PROCESSOR
00012          *        4 TO 7   BOU LINES ( 4 RIGHTMOST BITS )
00013          *        8=1      MULTI DEVICE CONTROLLER
00014          *        8=0      SINGLE DEVICE CONTROLLER
00015          *        9=1      X1215 DISK
00016          *        10 TO 15 DEVICE ADDRESS
00017          *
00018          *
00019          *
00020          *        DESCRIPTION
00021          *
00022          *        BEBOOT LOADS ONE RECORD ONTO LOCATION /80 THEN START AT /84
00023          *        THE RECORD IS THE SECTOR # 1 IF DISK,OR 254 CHARACTERS OF THE
00024          *        INPUT DEVICE,LEADING NULL CHARACTERS IGNORED
00025          *
00026          *
00027          *
00028          *        USED REGISTERS :
00029          *
00030          *        A1 BOU LINES,NOT TO BE DESTROYED IF BOOT IS CALLED AGAIN
00031          *        A2 ADDR OF INR INSTRUCTION
00032          *        A3 ADDR OF CIO INSTRUCTION(WHICH IS DESTROYED AND NEEDS TO BE
00033          *                                   RESTORED IF BOOT IS CALLED AGAIN)
00034          *        A4 ADDR OF SST INSTRUCTION
00035          *        A5 MULTIPLEX : CONTENTS OF 1ST WORD TO BE SENT TO EXT REGISTER
00036          *           IO BUS : CHARACTER COUNT,INITIALIZED AT 254 AND DECREMENTED
00037          *        A6 MULTIPLEX : CONTENTS OF 2ND WORD TO BE SENT TO EXT REGISTER
00038          *                                   (LOADING ADDR)
00039          *           IO BUS : ADDR OF NEXT CHAR TO BE LOADED,INIT AT /80 AND
00040          *                                   INCREMENTED
00041          *        A7,A8 WORK REGISTERS
00042          *        A9 WORK REGISTER
00043          *        A10 TO A14 NOT USED
00044          *        A15 CONTAINS THE KEYS'VALUE
00045          *
00046          *
00047          *
00048          *
00049          *
```

```
00050  *
00051  *
```

```
00052                          EJECT
00053                          AORG    0
00054                  *
00055                  *
00056                  BOOT    EQU     *
00057                  *       INITIALIZE REGISTERS
00058  0000  0200  F           LDK     A2,INR          ADDR OF INR INSTRUCTION
00059  0002  0300  F           LDK     A3,CIO          ADDR OF CIO INSTRUCTION
00060  0004  0400  F           LDK     A4,SST          ADDR OF SST INSTRUCTION
00061                  *                               EXTRACT DEVICE ADDR AND INIT I/O COMMANDS
00062  0006  861E              LDR     A6,A15
00063  0008  263F              ANK     A6,/3F          DEVICE ADDR
00064  000A  9629              ADRS    A6,A2           INITIALIZE I/O INSTRUCTIONS
00065  000C  962D              ADRS    A6,A3
00066  000E  9641              ADS     A6,HIO
       0010  0000  F
00067                  *                               EXTRACT CONTROLLER ADDR AND INIT WER INST
00068  0012  871E              LDR     A7,A15
00069  0014  3FC6              SLC     A7,6            MULTI OR SINGLE DEVICE CONTROLLER
00070  0016  5600  F           RF(6)   INIT20          SINGLE ONE
00071  0018  260F              ANK     A6,/F           MULTIPLE ONE
00072                  INIT20  EQU     *               INITIALIZE MULTIPLEX DBLE WORD:
00073  001A  9631              ADRS    A6,A4           SST INSTRUCTION
00074  001C  3E41              SLL     A6,1
00075  001E  9641              ADS     A6,WER1         SET UP WER INSTRUCTIONS
       0020  0000  F
00076  0022  9641              ADS     A6,WER2
       0024  0000  F
00077                  *                               LOAD A1 WITH BOU CONTENTS
00078  0026  811C              LDR     A1,A7
00079  0028  0550              LDK     A5,80           MULTIPLEX DOUBLEWORD: LOAD 80 CHAR INTO
00080                  *                               LOCATION /80
00081  002A  0680              LDK     A6,/80
00082                  *                               CHECK IF DISK
00083  002C  3FE7              SRC     A7,7
00084  002E  5600  F           RF(6)   NODISK          NO
00085                  *                               FIXED HEADS ?
00086  0030  3FC1              SLC     A7,1
00087  0032  5600  F           RF(6)   NOSEEK          YES
00088  0034  0103              LDK     A1,3            SEEK ZERO
00089  0036  41C0      CIO     CIO     A1,1,0          CIO SEEK ZERO
00090                  NOSEEK  EQU     *
00091  0038  811E              LDR     A1,A15
00092  003A  3966              SRL     A1,6            SECTOR NUMBER
00093  003C  213C              ANK     A1,/3C
00094  003E  8520              LDRL    A5,/80CD        1ST WORD OF MULTIPLEX FOR DISK DEVICE
       0040  80CD
00095                  NODISK  EQU     *
00096                  *                               EXECUTE WER,WHATEVER THE CHANNEL IS
00097                  WER1    EQU     *
```

```
00098  0042  7500          WER     A5,0
00099                WER2    EQU     *
00100  0044  7601          WER     A6,1
00101  0046  F031          EXR*    A4          SST
00102  0048  F02D          EXR*    A3
00103  004A  5C06          RB(4)   *-4
00104  004C  871E          LDR     A7,A15
00105  004E  3F43          SLL     A7,3
00106  0050  5600  F       RF(6)   SST         MULTIPLEX
00107                *
00108                *               IO BUS
00109                *
00110  0052  8194          LDR     A9,A5       IF A9=A5 IGNORE LEADING CHAR,
00111                INR     EQU     *
00112  0054  4F00          INR     A7,0,0      READ ONE CHAR
00113  0056  5C04          RB(4)   *-2
00114  0058  E994          CWR     A9,A5       LEADING CHAR?
00115  005A  5400  F       RF(4)   INR10       NO
00116  005C  27FF          ANK     A7,/FF      CHECK IF NULL
00117  005E  580C          RB(0)   INR         YES,IGNORE
00118                *               NO,CHECK IF 4*4
00119                *
00120                INR10   EQU     *
00121  0060  879E          LDR     A15,A15     4*4
00122  0062  5600  F       RF(6)   STORE       NO 8*8
00123  0064  3F44          SLL     A7,4        YES,4*4
00124  0066  809C          LDR     A8,A7       SAVE LEFT BITS
00125  0068  F029          EXR*    A2          READ NEXT CHARACTER
00126  006A  5C04          RB(4)   *-2
00127  006C  270F          ANK     A7,/F       GET 4 RIGHT MOST BITS
00128  006E  9702          ADR     A7,A8
00129                STORE   EQU     *
00130  0070  E739          SCH     A7,A6       STORE CHAR
00131  0072  1601          ADK     A6,1        NEXT CHAR ADDR
00132  0074  1D01          SUK     A5,1        COUNT DONE ?
00133  0076  5924          RB(1)   INR         NO
00134                *               YES
00135                *
00136  0078  4180          HIO     CIO     A1,0,0
00137                *
00138                STATUS  EQU     *
00139  007A  4FC0          SST     SST     A7,0
00140  007C  5C04                  RB(4)   *-2
00141  007E  0F84                  A8      /84
00142                *
00143                *
00144                *
00145                       END     BOOT
```

```
          SYMBOL TABLE

  BOOT    0000  A   INR     0054  A   CIO     0036  A   SST     007A  A
  HIO     0078  A   INIT20  001A  A   WER1    0042  A   WER2    0044  A
  NODISK  0042  A   NOSEEK  0038  A   INR10   0060  A   STORE   0070  A
  STATUS  007A  A

          ASS,ERR,    00000
!EOF
PROG ELAPSED TIME:  00H-00M-00S-000MS-
BEA IPL525
DATE     /   /     TIME 24H-60M-60S-
```

Comment Sheet

P800M Programmer's Guide 2, Volume I (5122 991 27373)

Name _____

Company _____

Address _____

Telephone Number _____ Ext. _____

Comments or suggestions