EMAS 2900 SYSTEM NOTE

No: 1
Date: 6/5/77

Fixed Sites on EMAS System Discs

EPAGE NUMBER	LENGTH (EPAGES)	USE
0	X40	IPL SUPERVISOR (CHOPSUPE)
X40	X40	SUP VSN 0
X80	X40	1
XC0	X40	2
X100	X40	3
X140	X40	4
X180	X40	5
X1C0	X40	6
X200	X40	DIRECTOR VSN 0 (default)
X240	X40	1 (DIRVSN 1)
X280	X40	2
X2C0	X40	3
X300	X100	Default BASEFILE (or room to get 2 versions if required and neither exceeds 512K). IMP Compiler? No, just enough to RELOAD into the file system. (Process 1 DIRECTOR to look after the file system initialisation and maintenance)
X400	X100	Local controller stack, DIRECTOR stack, and gla for Process 1 (sub-sites to be assigned by Dave).

File system to start at X800 (or lower). (Currently it starts at X2000).

J.K. Yarwood

EMAS 2900 SYSTEM NOTE

No: 2 (revised)
Date: 10/5/79

Configuration and Grope

EMAS 2900 attempts to define the configuration on which it is running by the "grope" feature and to adapt (with limits) accordingly. This Note defines the current limits and the grope procedure.

The grope procedure has the great advantage of allowing one Supervisor to be used for all sites, thus minimising maintenance. It has the minor operational disadvantage that devices powered down or physically disconnected are missed during the grope and cannot be used until the next IPL. Devices in manual or marked not available are discovered and incorporated into the configuration.

Configurations

* OCP and SACs

  EMAS 2900 is currently restricted to one OCP and one SAC. It is simple to extend the software to multiple SACs but there is a real difficulty in the smaller machines where the SAC selection logic is omitted and the same SAC appears on all vacant ports. Since a single SAC has so much more usable transfer capacity than a 4-75, extensions to multiple SACs have a very low priority.

* Core and SMACs

  EMAS 2900 can accept these without limit. There must be at least 0.5 Megabyte and there must be at least 256K on SMAC 0. The advantages of having more than the minimum number of SMACs for any given core size are unquantifiable but probably marginal. The amount of core require depends on the OCP speed - probably 1 Mb on 2960, 1.5 Mb on 2970 and 2 Mb on 2976 or 2980 is a reasonable minimum.

* SFCs or Drums

  EMAS 2900 can run without a paging device but this would not give satisfactory multi-access except on a 2960 and smaller machines. For larger machines a ratio of between 5 and 10 Mb of paging storage per megabyte of core store will give a balanced configuration - the lower figure when substantial batch capacity is envisaged, the larger for 100% MAC machines.

* Discs

  EMAS 2900 supports only the larger discs, EDS100 and EDS200, with each disc being treated as a logical file system; EDS200s are preferable to EDS100s. EDS30s and 60s are too small. The number of drives required depends on the file space to be allowed to each user and on the number of accredited users. EMAS 4-75 allows each user over 10 Mb but the average is a little over 1 Mb. This figure may reflect the Edinburgh teaching load, and other installations will probably wish to allow 2-5 Mb/user. The accredited users should be spread as evenly as possible over the

available logical file systems so that each file system has its share of serious and casual users.

* GPCs

EMAS 2900 will support any number of GPCs, although the advantages of having more than one seem marginal. Each EMAS 2900 configuration must have at least one OPER (and preferably a two-screen OPER) and one tape cluster of two decks. Larger configurations will require two to four clusters for back-up and archiving. All other slow peripherals are best provided on the Front End Processor, although one local card reader and one local line printer (for diagnostics) are supported.

* FEP

A Front End Processor is required to handle all terminal and RJE traffic.

Groping

This process takes place in several steps and takes about 15 seconds as there have to be a lot of software timeouts. Before starting to grope it is essential to ensure that:

a) time-out is enabled on the Engineer's panel.

b) any line printers have the right cartridge no. set (see Step 3).

After this an IPL load will load CHOPSUPE, which starts the grope steps.

Step 1

CHOPSUPE tries to access all 16 possible SMACs - non-existent SMACs time out. Any SMACs present are then asked to send their configuration registers. From these CHOPSUPE builds the core tables and clears all core (except the first 256K of SMAC 0).

Step 2

CHOPSUPE determines the port(s) containing SACs. All trunks on the SAC(s) are then put into direct control mode and the controllers told to identify themselves. A list is made of the controllers relevant to EMAS 2900 - the others are ignored.

Step 3

All GPCs are initialised and their microprograms are loaded. Property codes are read for all streams. All OPERs are cleared and the the OPER with the largest number of screens is selected as the principal OPER. Reps are loaded into all printers according to the following table:

Cartridge no. (set on handswitch)	Rep
0	The currently loaded rep, if any; otherwise the 64 character rep
2	The standard 48 character rep
3	The non-standard 384 character rep for the Bush 2980
4	The standard 64 character rep
5	The standard 96 character rep

The first printer found is selected as the diagnostic printer. Any operable tapes are rewound and the labels read.

Step 4

All SFCs are initialised and their microprograms loaded. Any operable devices are formatted and the available number of page frames calculated.

Step 5

All DFCs are initialised and the devices examined. Any EDS100s and EDS200s are entered into a table and a label-reading sequence initiated. EDS30s and 60s are ignored.

Step 6

A communication area (Public Segment 48 - further details in EMAS 2900 Supervisor Note 15) is constructed with the information gleaned ready for handing to the Supervisor. A summary of the configuration is displayed on the principal OPER and the first Operator Command awaited.

P.D. Stephens

OPER

Each 2900 OPER station comprises a keyboard and one or more screens, up to a maximum of four. Each screen can display twenty-four lines (numbered from 0 at the top to 23 at the bottom), each containing forty characters (numbered from 0 on the left). One screen is connected to the keyboard and is referred to as the "right-hand screen". The hardware echoes the characters typed at the console on line 22 of this screen.

To simplify the software, EMAS 2900 regards each OPER station as a number of virtual displays. This number may vary - currently it is six - but it is independent of the number of actual screens on the OPER station.

The six virtual displays are as follows:

0 Last 21 messages from the System (scrolled) plus interactive facilities for entering Commands from keyboard.

1 Status of 1st 60 processes plus System queue information.

2 Status of 2nd 68 processes.

3 Reserved for VOLUMS and DIRECT.

4 Reserved for SPOOLR (see separate documentation).

5 The 24 messages from the System immediately previous to those on screen 0.

For normal multi-screen OPERs the right hand screen is locked onto virtual display 0 and the left hand screen starts with virtual display 1. Use of the "page forward" and "page back" keys on the OPER keyboard increases or decreases the number of the display on the left hand screen; the list is treated as cyclic - going past the last virtual display switches back to the first.

For single screen OPERs the only screen starts on virtual display 0. The page forward and back keys map all six virtual displays onto the screen. An additional feature is that pressing the "ENTER" key causes an immediate switch to virtual display 0, as does the arrival of a message or prompt for the interactive screen (i.e. virtual display 0).

The twenty-four lines of the interactive screen are used as follows. The "activities" referred to are explained below, in the description of externalroutine OPER.

lines 0-20 - scrolled output (from activity 7 on OPER)

line 21 - prompt line (from activity 8)

line 22 - input line (hardware)

line 23 - input echo (software)

Two routines are provided for interfacing:

1) DISPLAY TEXT for updating virtual screens 1-6, and

2) OPER for interactive facilities (i.e. virtual screen 0)

```
1) externalroutine DISPLAY TEXT (integer WHICH, LINE, CHAR, c
                                 string (41) TEXT)
```

The text on line LINE of OPER number WHICH starting from character position CHAR is set to TEXT.

```
 Lines  0-23 are on virtual screen 1
       24-47 are on virtual screen 2
       48-71 are on virtual screen 3
         .
         .
```

The third and fourth screens, when present, display virtual displays n+1 and n+2 respectively, where virtual display n is on screen 2. The addition is cyclic as for "page forward".

```
2) externalroutine OPER(recordname P)

   recordformat PFM(integer DEST, SRCE, c
                    integer P1,P2,P3,P4,P5,P6
                    string(23) TEXT
                    byteinteger LINE,CLEAR,ZERO, string(20) DTEXT

P_DEST = X'32'<<16 + operno<<4 + activity
```

operno: (0 <= operno <= 15) specifies which OPER is required; zero selects the principal OPER

activity:

1 - set monitoring level from P_P1; 0 (no monitoring) is the default

2 - normal termination (from GPC)

3 - attention (from GPC)

4 - initialize (from GPC)

5 - abnormal termination (from GPC)

6 - call DISPLAY TEXT as follows:

DISPLAY TEXT(WHICH, P_LINE, P_CHAR, P_DTEXT)

If P_ZERO=255, first clear the whole of the relevant virtual screen.

If P_ZERO=1, operate "page forward" by software until the virtual screen containing P_LINE is visible.

7 - output request, P_TEXT = message. The message is scrolled onto the right hand screen, preceded by "SRCE/" with SRCE in Hex.

8 - input request, P_TEXT = prompt string. The input is ponned back to "SRCE" in P_TEXT (two messages will be sent if the input line is longer than 23 characters).

9 - update left hand screen (kicked by the clock).

10 - nominate main OPER
P_P1 = new main OPER "operno". The specified OPER becomes the principal OPER. P_P1 = -1 makes SRCE (i.e. this) the current OPER.

Operational notes

1) The OPER accepts input in two modes:

   * following an input request (activity 8)

   * after "COMMAND" has been struck

   In both cases the appropriate prompt is displayed and the input line may be typed (and terminated by "ENTER").

   Requested input is sent to the service which made the request.

   Command input is sent to the resident command interpreter (routine PARSE COM).

2) Striking "COMMAND" will force the OPER into command input mode. Any input request which is pre-empted will be re-established once the command line has been input.

3) A command input line may have continuations. An ampersand (&) at the end of a line is the continuation mark.

P.S. Robertson
P.D. Stephens

EMAS 2900 System Calls

EMAS supervisors from Sup. 11F onwards have support code for the system call mechanism. The larger 2900s have a hardware mechanism for inward calls but not outward calls. The software support for inward calls has two purposes:

1) To appear identical to the hardware mechanism in the case of the smaller machines and also emulating machines where pressure on the micro-program store has led to the omission of the hardware mechanism.

2) To provide more information if the hardware call fails.

Inward calls are completely defined by section 5.3.15 of the PLI SD 2.5.1) and are not considered further in this note.

Outward calls are to enable the Subsystem to run user programs at a lower level of privilege and on a separate stack. Software intervention is needed to copy any parameters and to allow for an inward return - since the EXIT instruction cannot lower the ACR level.

Before making an outward call the Subsystem must

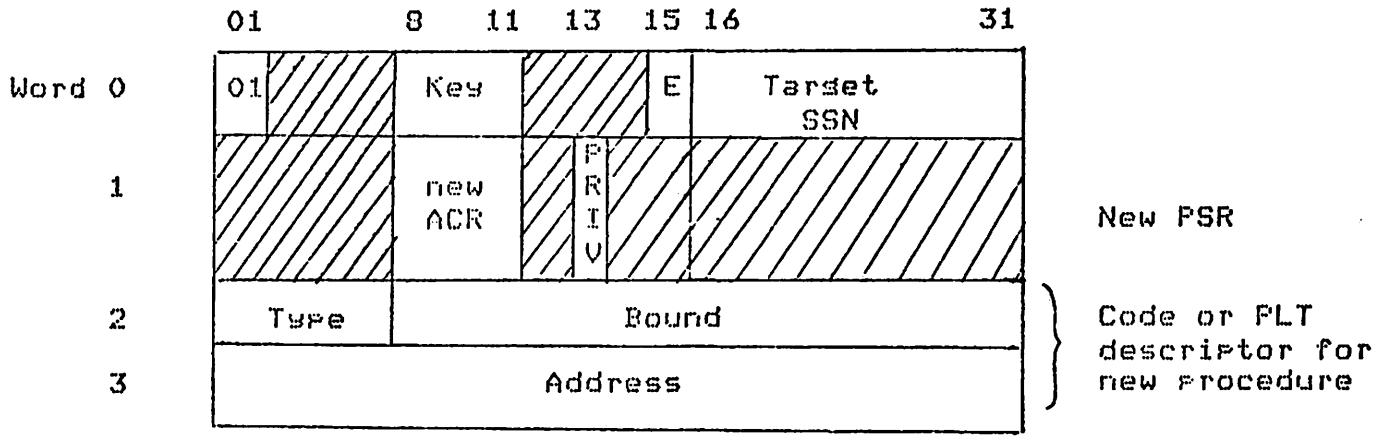
1) Create the new (Target) stack.

2) Notify the Local Controller of the stack so that it can provide the locked-down "SSN+1" segment.

3) Ensure that its current stack will not be accessible to the user at his higher ACR level.

4) Make a suitable entry in the system call tables.

The system call table entry for an outward call is as follows:



The Key is checked against the caller's ACR. If satisfactory, the procedure defined by the descriptor is entered on the new stack with B, C and the parameters set up on the old stack. A special inward return descriptor is set in the LINK field of the new stack which does the converse. ACC and B are preserved across an inward return so that it is possible to outward call Function and Maps if desired.

Bit 15 of word 0 of the system call table entry distinguishes ICL and EMAS conventions. If this bit is 0 the local name space of the called procedure starts at word 0 of the new stack (ICL convention). If Bit 15 is not 0 then the local name space starts after the "useful" information in the file, as defined by word 0 of an EMAS standard header. This permits "preloaded" Fortran stacks if these are going to be supported.

Error Codes

All system call failures give a system call error (16). The subclass signifies as follows:

^ = "I" Value out of range.

↓ = "J" value out of range.

2 = ACR check fails - user insufficiently privileged to use this system call.

3 = New stack invalid - validate fails on stack, or SSN+1 not resident, or new stack = current stack.

4 = Task calls not implemented.

5 = Inward return from procedures not activated by an outward call.

33 = ACR fails: outward call new ACR < current ACR, or inward call new ACR > current ACR.

34 = Invalid Code/PLT descriptor in SCT entry.

P.D. Stephens

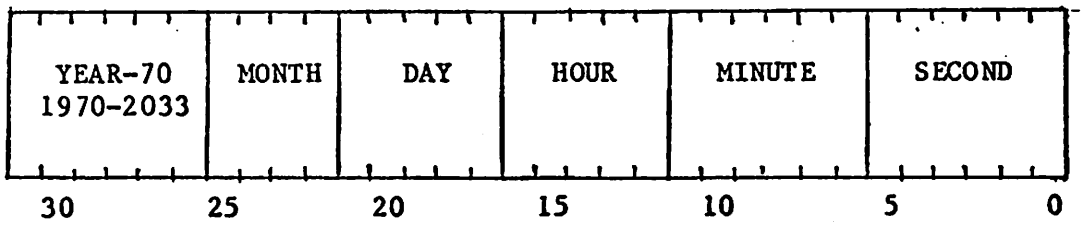
EMAS 2900 SYSTEM NOTE

No: 5
Date: 17/4/78

Proposal for a standard format for packed date and time

There are a number of places in the Subsystem, Director, and System Processes where it is useful to store the date or the date and time, in a packed format. I suggest that the following format, although inelegant, has the merits of being decoded with a trivial function, keeps both date and time in 32 bits, and can be used for sorting into chronological order. For dates beyond 2001 it will be necessary to sort on all 32 bits, treating the most significant bit as 2**31.

The format is:



When used for Date alone the 17 least significant bits should be all 0. The file

ERCC06.NR_PACKDTS

contains the source of the following functions, which operate with respect to the format.

stringfn UNPACKDATE(integer PACKED)

stringfn UNPACKTIME(integer PACKED)

These return strings with the standard formats using "/" as separators in the date and "." as separators in the time.

integerfn PACKDATE(string (8) DATE)

integerfn PACKDATEANDTIME(string (8) DATE,TIME)

Roderick McLeod

EMAS 2900 SYSTEM NOTE

No: 6
Date: 24/1/80

Bootstrapping the EMAS 2900 System

These instructions apply to the P2, P3, P4 and P4/1 processors. Where a processor-specific operation is required in the load sequence the processor type or types for which the action is appropriate is indicated in brackets.

1. By inspection at the Engineer's panel (P3, P4, P4/1) or Engineer's control processor (P2), ensure that:

   a) timouts are enabled

   b) the handkey switches for image store data are set to zero

   c) Stop on SMAC failure is set (P3)

   d) μprog load is inhibited (P4)

   e) all other switches are in the central position (P3, P4, P4/1)

2. Configure the store as required, but note that:

   a) there must be at least 256K of contiguous store on SMAC0 starting at real address 0 to allow CHOPSUPE (the Supervisor loader) to load

   b) apart from requirement (a) the store need not be contiguous

   c) the store may be configured as being interleaved (P4, P4/1)

3. Ensure that any unused SAC ports are <u>not</u> connected back-to-back.

4. The printer repertoire to be loaded is determined from the cartridge number set on the printer thumbwheel, viz:

   0 - leave the currently loaded rep loaded, or load the 64 character rep if no rep is loaded (this rep comprises the first 64 characters of the 96 character rep)
   2 - load the 48 character rep for the PODPS Craiglockhart 2970
   3 - load the 384 character rep for the Bush Estate 2980
   4 - load the standard 64 character rep
   5 - load the standard 96 character rep (as for the ERCC KB 2970)

5. If it is not already loaded, load the standard OCP microprogram.

6. Mount the System IPL tape and perform the standard IPL sequence (P3, P4, P4/1) or autoload (P2) sequence <u>twice</u> - the first time to skip over the tape label, and the second time to load and enter the bootstrap program.

7. The bootstrap reads CHOPSUPE from the tape, rewinds the tape to BT and enters CHOPSUPE. The configuration is groped and after a short delay (-10 seconds) configuration details are displayed on the control Oper and the main line printer.

General note: The remaining instructions entail the use of the Oper console. Press COMMAND before typing a line, and press ENTER to enter it after it has been typed.

8. The EDS100 selected to contain the file system should now be formatted, thus:

```
FORMAT EDnn -1 -1
```

This will format the whole of the disc, indicating on the Oper if there are any write failures. These failures should be noted for use subsequently in section 13. Provided that there are no failures below cylinder number 19, continue with the system set-up; otherwise another disc will have to be used.

When the format is complete (after -8 minutes) an IPL label should be written to the disc, thus:

```
ILABEL EDnn <label>
```

See Director Note 4 for label conventions.

9. Now prime down the System from the IPL tape to disc:

```
PRIME <tape> <disc> X400
```

10. Inform the System of the current date and time:

```
DT ddmmyy hhmm
```

This may not be necessary on machines where the clock is preserved through IPLS (P4 now; others in future).

11. Director version 1 should be selected:

```
DIRVSN 1      (see Director Note 9)
```

This prevents an automatic consistency check being applied to the file system by the DIRECT process when the Supervisor is loaded, nor will the VOLUMS and SPOOLR processes be started.

12. The main Supervisor can now be loaded:

```
SLOAD <fsys> <site>
```

where fsys is the file system number - that is, the numeric part of the disc label; e.g. EMAS00 would be FSYS 0.

site is the start page number of the required Supervisor; e.g. X80.

13. The Supervisor creates the DIRECT process. A new file system can now be created by using the following Director commands (see Director Note 10). In each case n is the decimal FSYS number.

    a) D/CLEAR FSYS n

       The responses are: 1/BAD PGS CT=0
                          1/VOL/SPLR IND CRE 0 SUCC
                          1/DONE

    b) If there were any bad tracks discovered during formatting, enter them in the bad pages list:

       D/BADFSYSCYLTRK <fsys> <cyl> <track>

       The response is 1/DONE.

14. Next perform the disc consistency check:

       D/CCK n

    The response is

       1/<disc> 0% FULL

15. Re-IPL the System from disc (similar to step 6, but done only once), and again do steps 10 and 12 (but not 11). The DIRECT process is created; this performs another file consistency check and then starts the VOLUMS and SPOOLR processes.

16. The essential Subsystem and other user files can be transferred from a back-up tape (see VOLUMS command TRANSFER) after process SUBSYS and any prospective users have been accredited (see DIRECT command NEWUSER).

17. Log-on may now commence. Note that the SPOOLR process will at this stage be configured for only JOURNAL and LP queues. It will be necessary to specify a CONFIG file using the S/CONFIG command, and to start and re-start SPOOLR before normal service can be offered.

Appendix: 2960 IPL

IPL is done from the ECP:

autoload	DO ALD pts	(Kent 2960)	send twice if
	DO AL pts	(Dalkeith 2960)	IPL from tape
register load	RT		
run	GO		

To set the handkey register:

SW <hex value>

To halt the OCP (and display current PC)

HA (use GO to restart)

Hardware dump:

DO DUMP pts	(Kent)
DU pts	(Dalkeith)

EMAS 2900 dump:

Halt	HA	
Reset	RT	
Set handkey	SW ptsm	port trunk stream mechanism (of dump tape)
Run	GO	

The image store may be read and written to (if appropriate) from the ECP if you know the equivalent "working store" word numbers (engineers will tell you):

RWS <word>
WWS <word> <value>

The OCP control register #6011 is WS #27.

J. Maddock
(revised by P. Stephens
and J.K. Yarwood)

No: 7
Date: 11/12/79

DBOOT

DBOOT is the piece of code entered immediately after a disc IPL. It is concerned with:

- re-arranging and tidying main store
- on dual systems, IPL'ing the other OCP
- entering CHOPSUPE

When the IPL button is pressed, a transfer into store is initiated from the device whose address is set up on the console.

1. the transfer starts at real address 0.
2. at the end of the transfer:
   - the OCP loads its microprogram from real address 64 (words) onwards, unless there is a 'stop' word there.
   - the image store locations in block 0 lines 0-15 are loaded from real words 32-47 and LSTB and PSTB are loaded somehow from words 48-63.
   - execution commences at the instruction addressed by PC.
   - whether PC and subsequent addresses are treated as real or virtual depends on the RAM bit in the SSR. If virtual, the PAGE and SEGMENT tables must already be set up and must also be able to address the controller communications area in words 0-7.
3. the communications area for the controller involved in the IPL is in real words 0-7, the response area being in words 6 and 7. There are no streams.
4. the OCP can establish SAC/TRUNK/STREAM of the input device by:

   SAC from the OCP interrupt flags in image store
   TRUNK from SAC interrupt flags
   STREAM from bits 24-31 of real word 6

The main store layout when DBOOT is entered, which it is vital to understand, is:

X'00000' communications area

X'00028' start of first hole available for code

X'00080' initial machine state, PC etc.

X'000D0' start of second hole for code

X'00100' OCP microprogram

X'13E00' start of third hole for code

X'14000' CHOPSUPE - Part 1

X'1C000' space for OCP microprogram overlays(P3 only)

X'20000' CHOPSUPE - Part 2

X'37000' limit of store referenced by the IPL, beyond this point the store may have parity errors and can be cleared only by writing zero quads

Start by determining the OCP type. The various image store addresses differ for P2/3, P4/0 and P4/>0. CTB is loaded with the address of the appropriate set.

If OCP is not an S1, pick up real word 6 (CRESP0) and check that the Normal Termination bit is set (*** else IDLE 'F000', not a normal termination) and save STREAM in MARK.

On a P4, set image store location 4012 bits 26-31 to allow SAC interrupts when the System is running.

Pick up the System Interrupt Register and extract the interrupting port number P, form the address of the trunk flags, X'44P00000', and read and clear the SAC interrupt flags. Extract the trunk number, T (*** else IDLE 'F001', no interrupt flags). Set PAW to X'PT0000'. After a tape IPL, MARK and PAW are both set to -1 (see SYSTEM NOTE 8).

If it's a dual OCP, do the remote IPL by sending a special message to the other OCP having first, if it's a P2/3 dual, NEQ image store location X'600A' with X'80' >> HISPORT !!!! Execute a busy loop for a while to give the other OCP time to do its IPL.

Clear the top end of SMAC 0 with long long zero words (X'37000'- X'40000')

If a P3, the OCP microprogram overlays are moved from wherever they are to X'4000'-X'8000' and the OCP's descriptor is updated appropriately. If it's a dual P3, the other OCP's descriptor is updated also.

The final step is to move the two parts of CHOPSUPE down to X'8000' - X'10000' - X'27000' and ACTIVATE it. (CHOPSUPE has to be split to accommodate the OCP microprogram)

A. Gibbons

No: 8
Date: 12/12/79

IPL'ing from Tape

Tapes to be used for IPL are written by VOLUMS and consist of:

- an extended label block; this contains the special 'stop' word at byte X'100' so that, if this block is read, the OCP stops in an orderly fashion
- a TIPLBLOCK
- a tape mark
- several 4K blocks (not written by VOLUMS but by BULK MOVER) comprising the 'system' to be IPL'd

This set of 4K blocks contains the same information as would have been read from disc in a disc IPL. The function of the TIPLBLOCK therefore is to read in these blocks and position them in store as if they had been read by a disc IPL, and then to start execution of this code, once again as though after a disc IPL.

The function of the TIPLBLOCK program in more detail is:

- clear mainstore (this program occupies the first X'1000' bytes)
- if it's a P3, move the OCP microprogram overlays to X'3A000' - X'3E000'
- move this program to the 'top' of store (X'3E000'), re-enter it there and initialise variables
- initialise a trace area
- compute PORT, TRUNK and STREAM of the IPL device
- load GPC microprogram and connect the IPL stream
- locate the first operable deck on the IPL stream
- read blocks from the tape, the first block going temporarily to X'38000', until either a tape mark is read or the next block read would overwrite the first block in its temporary home
- rewind the tape
- move the first block down to the bottom of store
- ACTIVATE the program that has been read in

Program errors

If a fatal error is detected, the program IDLEs:

F001 CRESP0 zero on entry

F002 not a normal termination

F003 SAC interrupt flags, which give the TRUNK, zero

F004 first four decks on the IPL stream are inoperable

F005 GPC microprogram load failed

F006 CONNECT IPL stream failed

F007 Controller Detected Error when reading tape

F008 more than 10 attempts to read a block

F009 backspace failed when retrying read block

F010 rewind fails

F011 sense fails

F012 MARK < -1

F013 non attention response received when an attention expected

Background notes

TIPLBLOCK is produced from a MAPLE program, source TBOOT1S. When this has been compiled and converted to EMAS object format, with the name ERCC08.SIPLF, it may be converted to a

```
%CONSTINTEGERARRAY TIPLBLOCK(0:n)
```

in file ERCC08.TIPLBLOCK by calling the command MAKETIPLBLOCK, source MAKETBS. The OBEY file TBOOT does all this.

A. Gibbons

No: 9
Date: 12/12/79

Using MAPLE

ICL's MAPLE (or SFL) compiler has been made available on EMAS by Bob Eager of Kent University. Although the use of an assembler, such as MAPLE, is not normally recommended, there are times when it is necessary (e.g. IPL routines) and MAPLE provides a good framework for writing 2900 machine code (e.g. 'natural' expression of the various operand formats, pseudo OP codes for versions of jumps and macros - very reminiscent of System/360 Assembler in fact).

To use the MAPLE compiler, type the following:

```
INSERT(ERCC08.MAPLE)
```

You need only ever do this once; it will be remembered in subsequent logons.

To enter the compiler, type:

```
MAPLE(sourcefile, OMF=Z.omffile, PLT=YES, KEYS=NO)
```

Note that the placement field in the OMF parameter must appear but is ignored.

There are some subsidiary commands, used to set up options before running MAPLE:

MAPLENOLIST	- prevents a listing file being produced.
MAPLELIST	- causes a listing file, called T#LIST, to be produced. This is a convenient name for many commands; see SEND, LOOK, etc. This is the default setting.
MAPLELIST(name)	- nominates a file to receive the listing.
MAPLELSIZE(size)	- sets maximum size of listing file (bytes). Default is 1000000.
MAPLEEDIT(file)	- nominates a file to be used for compile-time editing commands.
MAPLENEW(file)	- nominates a file to receive the edited source file.

The default is for no editing, and no source output.

All defaults may be restored by the command:

```
MAPLEDEFAULTS
```

The OMF output from MAPLE may be converted to EMAS load format by the command:

```
COMF(omffile,emasfile)
```

The MAPLE components are held in a PD file MAPLE0108. To rebuild the EMAS version of the MAPLE assembler, simply:

```
OBEY(MAPLE0108_MAKEITUP)
```

To make a traced version, use:

```
OBEY(MAPLE0108_MAKETRACE)
```

Summary of pdfile members

COMF2S	- source of OMF to EMAS load format converter
HELP	- this file
KDIAG3S	- source of diagnostic support routines for traced version
KMAPL	- object code of MAPLE assembler (EMAS format)
MAKEITUP	- OBEY file to build production version
MAKETRACE	- OBEY file to build traced version
MBASE3S	- source code of interface program
OMFSQ2S	- source of support routines for OMF converter

A. Gibbons

EMAS 2900 SYSTEM NOTE

No: 10
Date: 27/05/80

IDLEs at IPL

If certain errors occur before an Operator console has been found CHOPSUPE cannot tell the Operator. Instead it halts with an IDLE instruction. These signify as follows:

a) IDLEs during Booting (from Disc or Tape)

F001	CRESP0 zero on entry
F002	Not normal termination
F003	SAC interrupt flags, which give trunk zero on entry
F004	First 4 decks on IPL stream inoperable
F005	GPC microprogram load failed
F006	Connect IPL stream fails
F007	Controller detected error when reading tape
F008	More than 10 attempts to read block
F009	Backspace failed when retrying read block
F010	Rewind fails
F011	Sense fails
F012	MARK < -1
F013	Non attention response received when attention expected

b) Errors detected during GROPEing for devices

B00B	OCP not 2960, 2970, 2980, 2972 or 2976
00DD	No GPC found in configuration
0DDD	No Controllers found at all
FF00	Too many GPC's (>8)
FF01	Too many slots (>256) or supplied table too small
FF02	Too many entries in RESPONSE array
FF03	Too many magtape streams (>32) (in FORM TABLES)
FF04	Too many Operator console streams (>7) (in FORM TABLES)
FF05	Supplied table too small (in CHECKLIM)

c) Errors detected after GROPE completed

AAAA	Normal CHOPSUPE idle (awaiting command from Operator console)
3333	IMP %STOP executed (software error)
E00E	Dump to tape completed successfully
12121	Dump to tape failed (deck not known or faulty)
CCCC	Attempt to return from procedure invoked by Activate

d) Unexpected interrupts in CHOPSUPE

00F0	System error interrupt occurred (probable OCP fault)
00F1	External interrupt occurred (none ever expected)
00F2	Multi-processor interrupt occurred (none ever expected)
00F4	Virtual Store interrupt occurred (none ever expected)
00F6	Program error interrupt occurred (software or hardware fault)
00F7	System Call interrupt occurred (no System Calls ever made!)
00F8	OUT interrupt occurred (no OUTs in code!)
00F9	Extracode interrupt occurred (none ever expected)
00FA	Event pending interrupt occurred (none ever expected)
00FB	Instruction Counter interrupt occurred (always masked)

P.D. Stephens