

ACR <user> <acr> <fsys>*

Sets the ACR field in the file index for <user> on <fsys>. To be used only exceptionally. Normally, use command SETACR in MANAGR.

BAD FSYS CYL TRK <fsys> <cyl> <trk>

Makes entries in the list of "bad" (unreadable, unwriteable) pages on <fsys>. <cyl> and <trk> define 3 EMAS pages on EDS100s and 200s, and so 3 entries are created by this command. The list comprises integers holding the logical page numbers of the bad pages. The list can be printed using the command LIST BAD PAGES in ENGINR. The command BAD FSYS PAGE (q.v.) to DIRECT can be used to make a single entry of a single logical page number.

BAD FSYS PAGE <fsys> <page>

Enters (logical) page number <page> into the list of bad pages on <fsys>. See BADFSYSCYLTRK above.

BASEF <user> <fsys>*

Reports the Subsystem "basefile" identifier for <users> on <fsys>. The identifier is recorded in the user's file index. The basefile is the file which is loaded and entered by Director at process start-up. The default basefile identifier is normally null, when a file which has been "PRGed" to a fixed site, currently X'380', on the SLOAD disc is used. The basefile to be used may be specified by the DIRECT command SETBASEF (q.v.), or by a call by a user process to its Director.

BROADCAST

Broadcasts a message, set up using DIRECT command SETMSG, to each interactive process.

CCK <fsys>*

Initiates the file system consistency check for <fsys> (or for all on-line discs if <fsys> is omitted).

CCK DONE <fsys>

Paged processes cannot normally reference any <fsys> for which the file system consistency check has not been done. Exceptionally it may be necessary to allow such access, however, and giving this command enables access as though the consistency check had been done. The file system should be regarded as dangerously vulnerable unless a consistency check has been performed, and an early IPL is desirable after this command.

CIND <user> <fsys>*

Re-initialises the file index for <user> on <fsys>. To be used only exceptionally; the MANAGR commands NEW USER and DEL USER are infinitely preferable, allowing the index owners surname to be entered and an archive index to be created.

CLEAR BAD PAGES LIST <fsys>

Clears (to zero) the list of recorded "bad" (unreadable, unwriteable) pages on <fsys>.

Note: it is necessary to omit intermediate spaces in this command (except the one delimiting <fsys>) unless the "prompt" mode of DIRECT input is being used, because of the unusual length of the command line.

CLEAR FSYS <fsys>

Re-initialises the file system on disc <fsys>.

CLOSE <time>

Specifies the System close-down time (0001 to 2400) at which the service is to be automatically closed. If <time> is specified as 0 (zero), any previously specified close-down time is cancelled.

CLOSE DOWN

Causes the service to be closed (within about 30 seconds).

CLOSE USERS <time>*

Operates analogously to CLOSE <time>, except that the three System processes DIRECT, VOLUMS and SPOOLR are not stopped, allowing (for example) system maintenance work to be performed after user processes have been stopped without the need for an IPL. If <time> is omitted, the System continues to run, but new log-ons are rejected with the message "NO USER SERVICE". The service may be re-opened using command OPEN USERS (q.v.).

CONNECT FE <fe no>

May be used after a re-load of FEP software to re-establish the "log-on control stream", without an IPL of the mainframe.

DDUMP <fsys> <index no>

Used exceptionally to dump a page from the file system index area on <fsys>. Frequently command DUMPI can be more profitably used.

DELIVER

Used to set delivery information (by default equal to "Machine Room") for main log files sent to SPOOLR's LP queue as a result of DIRECT command PRINT. DIRECT prompts for an input string of delivery information, a null string causing the default to be reinstated. Delivery information remains set until a further call of DELIVERY, or until IPL.

DEL USER <user> <fsys>

Deletes a user's file index. To be used exceptionally, MANAGR command DELUSER being preferred.

DIRVSN <user> <fsys>* <vsn>*

Used either to report the Director version currently set in the user's file index (in this case one or both optional parameters are omitted), or to set the Director version (0, 1, 2 or 3) for the user to <vsn>. To be used exceptionally, for Director testing. (Supervisor command DIRVSN should also be noted in this context.)

DISCONNECT FE <fe no>

Disconnects the "log-on control stream" for the specified FEP.

DUMPI <user> <fsys>*

Used to dump out the file index for <user> on <fsys>, both in hexadecimal and partially formatted.

FSYS <user> <fsys>*

Reports the file system number on which <user> resides, or "user not known". If the optional parameter <fsys> is supplied, the search for <user> is restricted to that file system. Only discs for which the file system consistency check has been performed are searched.

INT: <user> <character>

Relays a "single-character" interrupt message to <user>, exactly as though the (Escape) INT: mechanism had been invoked from an interactive terminal. Note that if two or more processes currently exist belonging to <user> it is not possible to determine which process will receive the interrupt message.

MAIN LP

Used to direct main log output from the system to the main line printer (determined at IPL), instead of to files which are ultimately to be spooled to the LP or JOURNAL queues of the SPOOLR process. DIRECT command PRINT re-instates the spooling of main log data.

MSG <user>

Used to send a one-line text message (an operator message) to <user>. DIRECT prompts for the one-line message. Note that if two processes currently exist belonging to <user> it is not possible to determine which will receive the message. If there is no process belonging to <user> currently active, the message is stored for the next invocation of an interactive process by <user>.

NEW USER <user> <fsys> <nkb>

Creates a file index of size <nkb> for <user> on <fsys>. To be used only exceptionally, MANAGR command NEWUSER being preferred, since the latter allows for attachment of the user's surname and the creation of an archive index.

OPEN TO <usergroup>

Allows a username (6 characters) or a user group (6 characters including some '?' characters) to be appended to a list of usernames and usergroups for which log-ons are allowed even if the service is generally closed to users. Currently a maximum of 10 separate entries may be stored. Individual items cannot be removed from this list, but the whole list is destroyed by the DIRECT commands OPEN USERS and CLOSE USERS.

OPEN USERS

Allows users to (re-)commence logging on after the service has been closed (e.g. by command CLOSE USERS, q.v.).

PASS <user> <pass>

Used to specify a password by which <user> may log-on to the System (even if the service is generally closed to users). Only one <user>/<pass> combination is stored; subsequent PASS commands cause replacement of the current pair. The special password is otherwise cancelled by the DIRECT command PASS OFF.

PASS OFF

Cancels any stored special password specified earlier by DIRECT command PASS (q.v.).

PRG

Used to move System programs (supervisors, Directors, subsystems) from a file to a chosen fixed disc site. DIRECT prompts for two input lines, the first giving the <file> and <fsys>*, the second the disc <label> and disc <site>. <site> must be a X'40'-aligned numeric value.

PRINT <n>*

A value of 1 is implied if the parameter is omitted. For n>0, the current main log file is closed and copied to SPOOLR's LP queue (as well as to the JOURNAL queue).

For n>1, the subsequent n-1 main log files are additionally spooled to the LP queue (as well as to the JOURNAL queue), unless a subsequent PRINT command overrides this effect. For n=0, the current main log file is closed and spooled to the JOURNAL queue (only). Any PRINT command to DIRECT (re-)initiates spooling of main log data to disc if it was currently being printed directly on the main line printer.

Note: main log files belong to VOLUMS's file index, and commence "M#" (destined for JOURNAL) or "L#" (destined for LP). Just before any file is spooled, DIRECT renames the file (first two characters remain unchanged) so that the last four characters give the current time of day. This may help to identify files in SPOOLR's queues, for exceptional copying/printing.

PRM <file> <fsys>*

Attaches read, write and execute access permissions to <file>, specified as <user>.<11-char filename>. To be used only exceptionally.

PROMPT OFF

Cause DIRECT to remove the prompt "Direct" from the OPER screen so that the COMMAND key and "D/<command>" may subsequently be used to input commands to DIRECT.

PROMPT ON

Causes DIRECT to place the prompt "Direct:" on the OPER screen so that subsequent commands to DIRECT can be entered without the "D/" prefix and without using the COMMAND key.

S <file> <fsys>*

Reports (on the OPER screen) the logical page number of the first page of each disc section of <file> on <fsys>. <file> must be specified as <user>.<11-char filename>.

SEND MSG <user>

Provided that a message has been stored using command SET MSG (q.v.), the message is sent to <user>. Note that if two processes currently exist belonging to <user> it is not possible to determine which will receive the message. If there is no process belonging to <user> currently active, the message is stored for the next invocation of an interactive process by <user>.

SET BASEF <user> <fsys>*

Sets the Subsystem "basefile" identifier to be used by <user>, whose file index is on <fsys>. The default identifier is normally null. See DIRECT command BASEF.

SET MSG

Used to store a text message to be dispatched subsequently to one or all interactive processes by command SEND MSG or BROADCAST. DIRECT prompts (with a ":") for text input; several lines may be typed, up to a maximum of 255 characters or until a line containing only ":" is typed. Implied newline characters are replaced by spaces, in building up the stored string. When the message is complete, it is printed out in full on the OPER screen for checking before possible dispatch.

SNOS

Reports the following data on the OPER screen: the disc number from which the main supervisor was loaded, the current setting of DIRVSN (supervisor command) and the "sync1", "sync2" and "async" service number bases for local processes.

START <user> <fsys>* <dirvsn>*

Starts a process for <user>, whose file index is on <fsys> using Director version <dirvsn>. (The latter parameter is to be used only when the implications are fully understood.)

STOP

Disconnects all FEP control streams and stops the DIRECT process. (Used only exceptionally - CLOSEDOWN preferred).

TRANSFER

May be used to move a file from ownership of one user to another. DIRECT prompts for two lines of input, each containing two parameters:

<file1> <fsys1>

<file2> <fsys2>

The first determines the identity of the file to be moved; the second determines its destination. <file1> and <file2> must be specified in the form <user>.<11-char filename>.

UNPRG

Used to move System programs (supervisors, Directors, subsystems) from a specified disc site into a file (which must not already exist). Parameters are specified exactly as for DIRECT command PRG (q.v.).

USERS <usergroup> <n> or
USERS <n>*

This command is used to set limits on the numbers of interactive processes belonging to various groups of users, or to display current numbers and limits. <usergroup> is specified as a 6-character string in which '?' characters (if any) are interpreted as standing for any alphanumeric character.

USERS <n> sets the maximum number of interactive processes.

USERS <usergroup> <n> sets the maximum number of interactive processes whose owners belong to <usergroup>.

USERS displays the current numbers in the usergroups which have been specified, with the corresponding current limits, on virtual OPER screen number 2. The corresponding display is also generated after input of the other forms of the USERS command.

VSN

Reports on the OPER screen the Director version.

J.K. Yarwood

Proposed Director facilities for Testing GPC Peripherals1. Introduction

In EMAS, all GPC peripherals are driven by interfacing to a resident routine called GPC, whose functions are:

- a) allocation of devices (described in mnemonic form, e.g. CRO) to device service routines.
- b) initiation of command chains expressed in RCB/LB/ALs supplied by the service routines.
- c) notification to the service routines of termination, attention, program-controlled and request initiation interrupts.

Service routines may be resident or non-resident, but in either case communicate with the GPC routine by sending and receiving 32-byte messages to and from it. In addition to the 32-byte messages, work areas may be provided at device allocation time; these areas contain the RCBs, LBs and ALs and perhaps also the data areas for transfers.

The provisional specification of the GPC routine (as implemented at October 1977) is given in EMAS 2900 Supervisor Note no 5. Changes are expected in detail rather than in structure.

2. Testing of peripherals

Provision has not yet been made for general testing of peripherals, nor is design for such complete. However, the following guidelines will obtain:

- a) Facilities will be provided to give one or more paged processes the level of privilege necessary to establish communication with the GPC routine.
- b) Provision will be made for pages of such processes to be locked down as necessary (in particular the pages containing RCBs/LBs and ALs and the data area will clearly need to be made resident for the duration of transfers).
- c) A set of procedures (part of the so-called DIRECTOR interface, and implemented as system calls) will be provided, to be used by the paged process:
 - i) to establish communication with the GPC routine;
 - ii) to allow messages to be sent to and received from the GPC routine, possibly with a modicum of validity checking of associated data;
 - iii) to specify the extent of necessary locking down of pages, as described above.

A provisional specification for these procedures is attached.

- d) It is expected that all programs which are to run in these privileged processes will be coded in IMP (see reference), so as to afford compatible parameter passing with the procedures provided, and achieve fast compilation and good run-time diagnostics using existing, proven software.

3. Detailed specification of procedures (provisional)

(This section is subject to change.)

externalintegerfn LOCK (integer PAGE ADDRESS)
external UNLOCK (integer PAGE ADDRESS)

Causes the specified "logical page" (currently 4K bytes) of the calling process's virtual memory to remain resident (cease being resident) in main store.

Possible error results:

- 56 PAGE ADDRESS is not connected in the virtual memory or is invalid.
-

externalintegerfn GET EPAGE (integername VIRTUAL ADDRESS)
externalintegerfn RETURN EPAGE (integer VIRTUAL ADDRESS)

Allocates (de-allocates) a page of main store to (from) the process. The (local) virtual address of the page is returned in the parameter.

Possible error results:

- 57 (for GET EPAGE) Page not allocated to this process.
-

externalintegerfn ALLOCATE (integer MNEMONIC, integername IDEN)
externalintegerfn DEALLOCATE (integer MNEMONIC)

Makes a request on the GPC routine to allocate (de-allocate) the GPC peripheral denoted by MNEMONIC (as a multi-character IMP value, e.g. M'CRO') to (from) the calling process.

IDEN is set to an identifier to be used in execute chain requests (see below).

Possible error results:

- 1 Parameter invalid or device not known.
- 2 Device already allocated (unallocated).

externalintegerfn EXECUTE CHAIN (integer RCBA, IDEN, PAWFN, SAWFLAGS)

A 'SEND CHANNEL FLAG' is issued for the device specified by IDEN (as returned by the ALLOCATE request). RCBA gives the address of the RCB to be used. PAWFN and AWFLAGS are the PAW function and SAWFLAGS fields (right-justified in these parameters) for the controller PAW word and stream SAWO word.

Possible error results:

58 Invalid parameter or RCB/LB/ALE not accessible.

externalintegerfn AWAIT RESPONSE (integername IDEN, RESPO, RESP1)

The process is suspended until a response is available from the GPC. IDEN, RESPO and RESP1 are then set with the device identifier obtained from the ALLOCATE and the stream response words 0 and 1, respectively.

Possible error results:

None.

4. Conclusion

It is intended that the facilities broadly described above will enable any desired operation to be carried out "on-line" on peripheral devices not currently allocated for other purposes.

Reference

"Edinburgh IMP Language Manual", edited by R.R. McLeod (2nd Edition, 1974).
Edinburgh Regional Computing Centre.

J.K. Yarwood

Operation of file system consistency check

Following IPL and the SLOAD command, a process 'DIRECT' is created, which performs the following functions:

1. file system maintenance
2. checking of log-in messages from the network and creation of other processes
3. dispatch of operator messages and broadcast messages to other processes.

'DIRECT' comprises some special routines set aside for these purposes, although it is still (currently) part of the DIRECTOR, whose code is shared by all processes.

There are four versions (0-3) of DIRECTOR on a system disc. Version 0 is the normal 'service' version, and is used by default. The DIRVSN command (e.g. DIRVSN 1) causes the DIRECTOR version specified to be used by processes started subsequently.

If the DIRVSN is 0 (the default), the DIRECT process performs the file system consistency check, starts the VOLUMS and SPOOLR processes, and connects the 'control streams' to front-end processors so that log-on messages can be received. If the DIRVSN is 1, 2 or 3 the consistency check is not done, nor are VOLUMS and SPOOLR started.

DIRVSN 1 will normally be an adequate 'back-up' for the current service version (frequently the 'previous' version); versions 2 and 3 are normally test versions.

Date and time

When the consistency check is invoked (either automatically or by command 1/CCK, which should be given only when no other process is running), the date currently set, by the DT command, is checked against the date of the previous IPL. If the current date is not within 6 days of the previous IPL, the consistency check does not proceed.

If the 'within 6 days' rule is to be overridden, either because of a previous error in 'DT' or because more than 6 days have elapsed, a specific CCK command, 1/CCK n, where n specifies the 'SLOAD' disc, must be used; in this case the date of the previous IPL will be over-written regardless. Otherwise the system should be IPLed again and a correct 'DT' given.

Consistency check error messages

A log of each consistency check is maintained on each disc, in file VOLUMS.CCKMESS. Many lines in the file commence

DT dd/mm/yy hh.mm.ss

Each username process is mentioned. Each file destroyed is mentioned. Names of files destroyed are preceded by 'FILE='.

The CCKMESS files are 'circular' text files, of size 256 Kbytes. The files may be copied to ordinary text files by the MANAGR command

CCKOUT (f/filename)

where f is the number of the disc concerned and filename is the file to contain the messages extracted. The 'f/' may be omitted, in which case all file systems are processed. 'filename' may be '.LP'.

The CCKMESS file header contains a 'READ TO' (past tense) pointer, which is updated by the CCKOUT program. If the CCKOUT program is not run before new consistency messages overrun the 'READ TO' pointer, the latter is updated with the 'START OF TEXT' pointer of the circular file header.

J.K.Yarwood

Creating a new file system

On a newly-formatted disc, or where a complete file-system is to be reprimed, the following procedures should be applied. Spaces are not significant in these commands except as delimiters.

1/CLEAR FSYS n

where n is the decimal number which is the last two digits of the disc label. n may not be omitted. The response is 1/DONE. Indexes are created on the disc for VOLUMS and SPOOLR.

1/CCK DONE n

This has the effect of marking the disc table entry for the disc as being "consistency-checked". It is currently necessary only to allow the CLEAR BAD PAGES LIST command to be effected (see below). Hence it may be omitted if that command is not to be executed. The response is 1/DONE.

1/CLEAR BAD PAGESLIST n

This is necessary for a newly-formatted disc, because the count of bad pages is uninitialised. For a reprime to a disc which has not been reformatted, this may be omitted if the bad pages list is to be retained (it is not affected by CLEAR FSYS). The response is 1/SUCCESSFUL.

1/CCK n

A file system consistency check should now be performed on the disc.

1/NEWUSER ERCC00 n 4

should be given for as many usernames as are required. n is the FSYS number. ERCC00 is an example of the username. 4 is the number of Kbytes to be allocated to the index. Please only type 4 for the time being. You can start the reprime after 2 or 3 of these, if you start with the first users on the tape: you will be able to keep ahead! (This will not be necessary in later versions of the system - a magnetic tape will be used.)

After the reprime, an IPL and new consistency check may be a good idea, though it is not logically necessary.

Temporary

To reprime the biggest files, you may need to have done the command

```
SETMAXFILE (VOLUMS,4000)  
ERCC10.DIRTESTY
```

This program will need modifying, too, to take an FSYS other than the "SLOAD" disc. You will also need file ERCC10.NEWSERVY to go with DIRTESTY.

J.K. Yarwood

Director Error Messages at the Main OPER Console

This note describes messages of the form

****<username> ERROR xxx <param>**

which Director sends to the main OPER console on certain error conditions, together with action to be taken (if any).

<username>	gives the owner of the process from which the message originated
xxx	are 3 hexadecimal digits identifying the error condition
<param>	If present is a decimal or hexadecimal number further qualifying the error condition

In general, these messages are of a non-urgent, non-critical nature. Special action, except where otherwise stated below, is not normally required unless the condition is persistent, when the author of this Note should be advised. In addition, notification would be appreciated of occurrences of errors marked *, which "should not occur".

<u>Error</u>	<u>Condition</u>
--------------	------------------

- | | |
|------|---|
| 001* | A file is being created with the (unusual) option of being zeroed at time of creation. In the event of failure to write zeroes to the file, a re-allocation takes place. If this procedure fails 10 times, this error is given and the create operation fails. |
| 002* | Director failed to fill stack DA entries. |
| 003* | Director failed to find a VM gap in which to connect a segment of file indexes. |
| 004* | Stack of ASYNC INHIBits in Director is in error. |
| 005 | Director failed to find an expected index at its correct location on disc.

Action: Notify Systems staff. An early IPL and file system consistency check is indicated. |
| 006 | At DDISCONNECT, Director failed to find the correct generation of a file in the file owner's index. This can occur if a user has been deleted (DELUSER), or his file index moved to another disc, while one or more of his files were in use by another process. No adverse side effects should result. |

- 007* Director failed to fill stack DA entries.
- 008 Director's file system claim-semaphore routine was called before the free-semaphore routine had been called to release the semaphore previously held.
- Action: An early IPL is probably desirable to avoid possible file index corruption.
- 009 Director's file system release-semaphore routine was called without a preceding corresponding call of the claim-semaphore routine.
- 00A Accounts overflow: 1=instrs; 2=OCP; 3=pturns; 4=connect.
- 00B DIRECT failed to destroy a workfile belonging to a user whose process has just terminated.
PARAM = Director flag from DDESTROY.
- 00C DIRECT failed to find the username in its process list for a process which has just terminated.
- 00D DIRECT's list of active processes is full. There is currently space for 255 active processes.
- 00E An FCHECK file system consistency check process failed to create the consistency-check record file VOLUMS.CCKMESS.
PARAM = Director DCREATE flag.
- Action: Systems staff to scrutinise VOLUMS index to discover the reason.
- 00F* At process start-up, the process was unable to connect a relevant disc-site (e.g. the default BASEFILE) into its virtual memory.
PARAM = flag from Director's DISC SEG CONNECT routine.
- 010* At process start-up, the process was unable to connect its System call table segment (for access to Director routines). PARAM = flag from Director's DISC SEG CONNECT routine.
- 011* At batch process start-up, the process failed to disconnect-and-destroy the file passed to it by SPOOLR giving the batch job details.
PARAM = Director's flag from DDISCONNECT.
- 012* At interactive process start-up, the process failed to connect its interactive terminal data streams.
PARAM = flag from Director routine CONNECT STREAM.
- 013* DIRECT failed to find the file index for the owner of a process just terminated.
PARAM = flag from Director's routine CONNECT INDEX.
- Action: An early IPL is indicated.
- 014* DIRECT failed to find the file index for a valid username for whom a process was to be started.
PARAM = flag from Director's routine CONNECT INDEX.

- 015 At completion of the consistency check for an FSYS, the FCHECK process was unable to create a file VOLUMS.LOSTFILES in which to record filenames destroyed by the consistency check.
PARAM = flag from Director routine DCREATE.
- Action: Systems staff to scrutinise VOLUMS file index to ascertain reason.
- 016* At completion of the consistency check for an FSYS, the FCHECK process was unable to connect the VOLUMS.LOSTFILES file in which to record filenames destroyed by the consistency check.
PARAM = flag from Director's routine DCONNECT.
- Action: Systems staff to scrutinise VOLUMS file index to ascertain reason.
- 017* At completion of the consistency check for an FSYS, the FCHECK process was unable to connect the VOLUMS.CCKMESS file, in which a record of the consistency check is to be placed.
PARAM = flag from Director routine DCONNECT
- Action: Systems staff to scrutinise VOLUMS file index to ascertain reason.
- 018* At completion of the consistency check for an FSYS, the FCHECK process was unable to permit the VOLUMS.CCKMESS file to DIRECT.
PARAM = flag from Director's routine DPERMISSION
- Action: Systems staff to scrutinise VOLUMS index to ascertain reason.
- 019 DSEGMENT fails.
- 01A DCHSIZE new SCONNECT fails.
- 01B DIRECT stack extend fails.
- 01C REF NEW BLOCKS not zero.
- 01D-01F Spare.
- 020-035 These relate to maintenance by DIRECT of a pool of 64K-sized files for use as mainlog files by the Supervisor routine PRINTER. In each case, PARAM is a flag from the relevant Director routine. The error message is preceded by a message "LOG TROUBLE" on the main OPER console.
- 020 Create logfile failed.
- 021 Get (logfile) Disc Address failed.
- 022 Permit logfile to DIRECT failed.
- 023 Connect logfile failed.
- 024 Re-connect logfile failed.
- 025 Set file status=temporary failed.
- 026 Disconnect logfile failed.
- 027 Rename logfile failed.
- 028 Connect logfile failed.
- 029 SPOOLR logfile copy to LP failed.
- 02A Create and/or connect logfile copy for LP failed.
- 02B Disconnect logfile failed.
- 02C SPOOLR logfile to JOURNAL failed.

02D Failed to create spare logfiles.
02E Spare.
02F Disconnect logfile failed.
030 Rename logfile failed.
031 Connect logfile failed.
032 No ready file in DIRECT's logfile pool.
033 File disc address offered by PRINTER for spooling does not belong to
a file in DIRECT's logfile pool.
034* Create VOLUMS.LOGMAP file (holding DIRECT's map of logfiles and its
process list) failed.
035* Connect VOLUMS.LOGMAP file failed.

036-03F Spare.

040* DIRECT fails to start VOLUMS at System start-up.
PARAM = flag from Director routine STARTP

041* DIRECT fails to start SPOOLR at System start-up.
PARAM = flag from Director routine STARTP.

042 At batch process startup, the process failed to connect the file
passed to it by SPOOLR giving the batch job details.
PARAM = Director's flag from DCONNECT.

J.K. Yarwood

Format of File Index Header

```

%RECORDFORMAT HDRF(%STRING (6) OWNER, %BYTEINTEGER %C
FLAG, SECTSI, ALNK, USE, IMARK, %C
IUSE, BUSE, ISESSM, SURNAME, %C
IMAX, TMAX, BMAX, STKKB, %C
%INTEGER LNKSTART, CELSTART, FREEBYTES, TRYING, %C
NAMSTART, NEXNAM, LOFDAD, TOP, %C
DWSP, BWSP, SEMA, MSGSEMA, %C
SEMANO, IINSTRS, BINSTRS, NKBOU, %C
DINSTRS, IPTRNS, BPTRNS, NKBIN, %C
IMSECS, BMSECS, CHKSUM, ASEMA, %C
%BYTEINTEGER CODES, SIGMON, DEPTH, CONCURR, %C
ACR, DIRVSN, MSGSPRIV, BATCHSS, %C
BASEF, DELY, STARTF, ADRTELE, %C
LOGFILE, SPECIALSS, SPFF1, GPOLDR, %C
%INTEGER USED AFDS, AFILES, AKB, USED FDS, %C
FILES, TOTKB, ATOP, SPA0 , DATE, %C
MAXFILE, MAXKB, DIRMON, %C
CHERFILES, CHERKB, CONNECTT, ARESTORES, %C
ZASL, ZFREEC, ZNCELLS, TEMPKB, %C
FILES0, FILES1, FILES2, FILES3, %C
BNUTS, INUTS, XLNKST, XCELST, %C
XTOP, XNAMST, ISTOP)

```

Note: This format applies both to the on-line and the archive indexes. Each entry in the description is marked 1 if used in the on-line index and 2 if used in the archive index.

<u>Field</u>	<u>Note</u>	<u>Description</u>
OWNER	12	The index owner's username.
FLAG	—	Not used; previous function was to be set non-zero when an error condition (inconsistency) in the index was found. Subsequently no routine would alter or look at the index until cleared "manually".
SECTSI	1-	The (maximum) size in epages of disc sections for files described by the indexes, set up (permanently) at index creation. Currently envisaged values: 16 or 32.
ALNK	12	List-head of whole-index-permission cells in list-pool zero.
USE	—	Not used; formerly current number of processes active in this username (i.e. OWNER).
IMARK	—	Not used; intended to assist with forward/backward compatibility considerations during development.

<u>Field</u>	<u>Note</u>	<u>Description</u>
IUSE	1-	Current number of interactive processes active in this username.
BUSE	1-	Current number of batch processes active in this username.
ISESSM	1-	Length of current session in minutes. Zero represents the default value, currently "very large".
SURNAME	1-	Listhead of cells in list-pool zero containing the initials and surname of the index owner.
IMAX	1-	maximum number of interactive processes which may be concurrently active for OWNER. Zero represents the default, currently 1.
TMAX	1-	Maximum number of interactive and batch processes which may be concurrently active for OWNER. Zero represents the default, currently 1.
BMAX	1-	Maximum number of batch processes which may be concurrently active for OWNER. Zero represents the default, currently 1.
STKKB	1-	Number of Kbytes of the processor stack (temporary) file for invocations of the process. (This file is connected as local segment 4 during process start-up.) Zero represents the default, currently 64.
LNKSTART	12	The offset, in bytes, from the start of the index of the link bytes for pool zero of list cells. (If any of pools 1, 2 or 3 exist they commence 256, 512 and 768 bytes beyond this point respectively.)
CELSTART	12	The offset, in bytes, from the start of the index of the list cells for pool zero. (If any of pools 1, 2 or 3 exist they commence 1024, 2048 and 3072 bytes beyond this point respectively.)
FREEBYTES	12	The number of bytes between the most recent (highest) file-name (identifier) and the corresponding (and hence lowest) file-descriptor.
TRYING	—	Not used.
NAMSTART	12	The offset in bytes from the start of the index of the area for storing file-names.
NEXNAM	12	The offset in bytes from the start of the index of the position where the identifier will be put for the next-created file.
LOFDAD	12	The offset in bytes from the start of the index of the lowest (most recent) file-descriptor.
TOP	12	The size of the index in bytes.
DWSP	1-	Reserved.

<u>Field</u>	<u>Note</u>	<u>Description</u>
BWSP	1-	Reserved.
SEMA	1-	The semaphore word for the index (set to -1 at consistency check).
MSGSEMA	1-	The semaphore word for the owner's message-file (#MSG), set zero at consistency check.
SEMANO	1-	The semaphore number for the index, being the number of kbytes between FSYSTART (the start of the file-system area on the disc) and the start of the index.
IINSTRS	1-	Number of thousands of machine instructions executed in interactive sessions for OWNER (cumulative over sessions).
BINSTRS	1-	Number of thousands of machine instructions executed in interactive sessions for OWNER (cumulative over sessions).
NKBOUT	1-	Number of kbytes of data "spooled out of the index" (cumulative).
DINSTRS	1-	Number of thousands of machine instructions executed in Director procedures. (Cumulative, but currently not updated).
IPTRNS	1-	Number of pageturns made on behalf of OWNER in interactive sessions (cumulative over sessions).
BPTRNS	1-	Number of pageturns made on behalf on OWNER in batch sessions (cumulative over sessions).
NKBIN	1-	Number of kbytes "spooled into the index" (cumulative).
IMSECS	1-	Number of milliseconds of OCP time attributed to OWNER in interactive sessions (cumulative over sessions).
BMSECS	1-	Number of milliseconds of OCP time attributed to OWNER in batch sessions (cumulative over sessions).
CHKSUM	-2	Checksum for the archive index.
ASEMA	1-	Semaphore word for accesses to the archive index (set -1 at consistency check).
CODES	—	Not used.
SIGMON	1-	Signal monitoring level to apply from next start-up of a process on behalf of OWNER (set zero on consistency check).
DEPTH	1-	Depth of monitor to be applied when DIRMON is non-zero (set zero at consistency check).
CONCURR	—	Not used.
ACR	1-	The ACR level at which the subsystem is to be started when a process is next created for OWNER. Zero represents the default, currently 6.

<u>Field</u>	<u>Note</u>	<u>Description</u>
DIRVSN	1-	The Director version (0, 1, 2 or 3) to be used at the next invocation of a process for OWNER (unless a D/START command specifies a Director version). A value of 255 represents the current default Director version (zero or as set by the DIRVSN command). When this value has been read at process start-up, it is then set to 255.
MSGSPRIV	--	Not used.
BATCHSS	1-	List-head of cells in list-pool zero to contain the name of a subsystem ("basefile") to be used for batch processes for OWNER. Currently unused.
BASEF	1-	List-head of cells in list-pool zero to contain the name of a subsystem ("basefile") to be used for interactive processes for OWNER. Currently used also for batch processes. A null filename implies use of the fixed site on the System disc. (See EMAS 2900 System Note No. 1).
DELY	1-	List-head of cells in list-pool zero to contain the default delivery information for data "spooled out of the index".
STARTF	1-	List-head of cells in list-pool zero to contain the identifier of a file of commands to be executed at process start-up. Currently unused, as the Edinburgh Subsystem keeps the identifier in the "option file".
ADRTELE	1-	List-head of cells in list-pool zero to contain the address and telephone number of the index owner.
LOGFILE	1-	List-head of cells in list-pool zero to contain a filename to be used as a Director process-monitoring file. (See EMAS 2900 Subsystem Writer's Manual).
SPECIALSS	1-	List-head of cells in list-pool zero to contain a filename to be used as a test subsystem. Reset to null filename after a process start-up using that subsystem. (Not yet implemented).
SPFF1	--	Spare list-head, currently set to ENDLIST at consistency check.
GPHOLDR	1-	List-head of cells in list-pool zero to contain a username to be used as the group-holder of OWNER'S scarce-resource units.
USED AFDS	1-	Number of file-descriptors in the archive index.
AFILES	1-	Number of files in the archive index.
AKB	1-	Number of epages in the archive index (perhaps to become the number of kbytes at a future date).
USED_FDS	12	Number of file-descriptors in the on-line index.
FILES	12	Number of files in the on-line index.
TOTKB	12	Number of kbytes of data described by the index.

<u>Field</u>	<u>Note</u>	<u>Description</u>
ATOP	1-	Size of the archive index in bytes (currently not maintained).
SPAO	—	Spare.
DATE	1-	Compressed date and time at which a process belonging to OWNER was most recently started.
MAXFILE	1-	Maximum size (kbytes) of non-temporary files belonging to OWNER. Zero represents the default value, currently 1024 kbytes.
MAXKB	1-	Maximum total file-space (kbytes) belonging to OWNER. Zero represents the default value (currently 32768 kbytes).
DIRMON	1-	A bit-mask specifying which Director calls are to be monitored (see also the DEPTH field of this format). Set to zero at consistency check. More detail in the EMAS 2900 Subsystem Writer's Manual.
CHERFILES	1-	Number of cherished files in the on-line index (not yet implemented - the field is zero).
CHERKB	1-	Number of kbytes of cherished filespace in the on-line index (not yet implemented - the field is zero).
CONNECTT	1-	Number of seconds of terminal connect-time for interactive processes belonging to OWNER (cumulative over sessions).
ARESTORES	1-	Number of files restored from archive (cumulative over sessions).
ZASL	12	Free-list pointers for list-pools 0, 1, 2 and 3.
ZFREEC	12	Numbers of free cells in list pools 0, 1, 2 and 3.
ZNCCELLS	12	Numbers of cells in list-pools 0, 1, 2 and 3.
FILES 0-3	-2	Number of types 0-3 files described by the archive index. Type 0 is archive files, type 1 is back-up files. Types 2 and 3 are unassigned.
INUTS	1-	Number of scarce-resource units assigned to the index.
BNUTS	—	Reserved for possible use as batch-resource.
XLNKST XCELST XTOP XNAMST	12	Copies of LNKSTART, CELSTART, TOP and NAMSTART (copied at consistency check), present as a prelude to moving the semaphore fields (SEMA, MSGSEMA, ASEMA) nearer to the front of the index.
ISTOP	—	Dummy entry marking the end of the record format.

J.K. Yarwood

The EMAS 2900 Front End ProcessorContents

1. Introduction
2. Software Components
 - 2.1 AMI Handler
 - 2.1.1 "Internal" transactions with the 2900
 - 2.1.2 Data transactions with the 2900
 - 2.1.3 Interface to the high-level protocol tasks
 - 2.1.4 AMI handler fault messages
3. Operation of FEP
 - 3.1 Loading procedure
 - 3.2 Powering off procedure
 - 3.3 Dumping the system
 - 3.4 Processing a Front End dump
 - 3.5 Responding to crashes

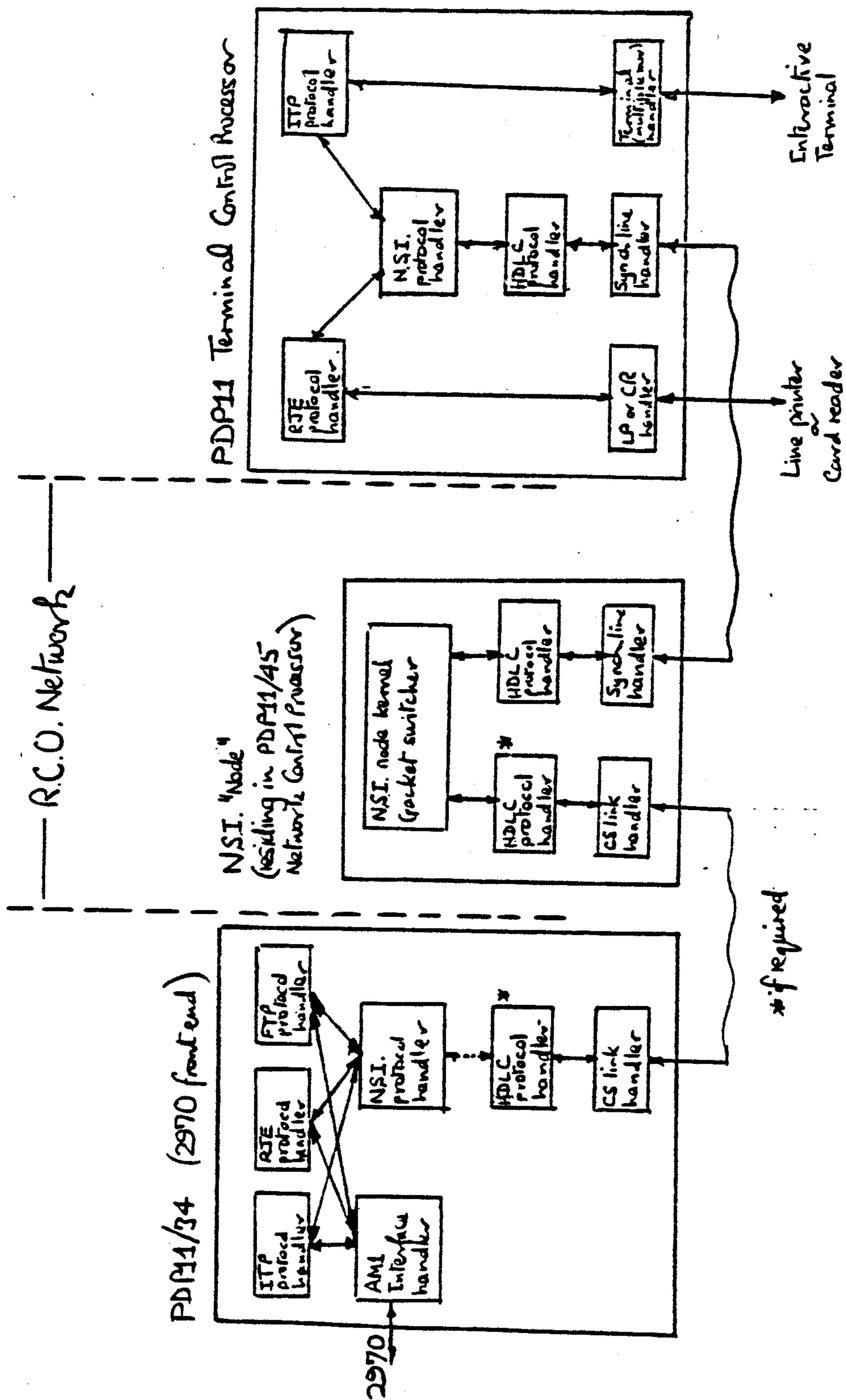
1. Introduction

This note describes the system software running in the EMAS 2900 Front End Processor, which is a DEC PDP11/34. It also describes operating procedures.

Section 2 summarises each software component in turn; subsection 2.1 then described the AMI handler in greater detail. It is intended that further subsections will be added in due course, each likewise dealing with a single component in detail.

Section 3 is taken largely from the EMAS 2900 Operator's Manual. It describes the operational aspects of the FEP.

Figure 1.1 below indicates how the software components of the FEP are interlinked, and, for the ERCC 2970 installation, how the FEP is connected to the EMAS 2900 mainframe on the one hand and the Regional Computing Organisation (RCO) ^{Network} on the other. The protocol hierarchy for interactive and non-interactive operation is shown diagrammatically in Figures 1.2 and 1.3, and for completeness the formats of the various protocol packets are also given (Figure 1.4).



Notes:

- ITP - Interactive Terminal Protocol
- NSI - Node Standard Interface
- CS link - Edinburgh University Dept. of Computer Science Link
- AMI - Application Module of ICL 2900 Series General Peripheral Controller
- Data flow -

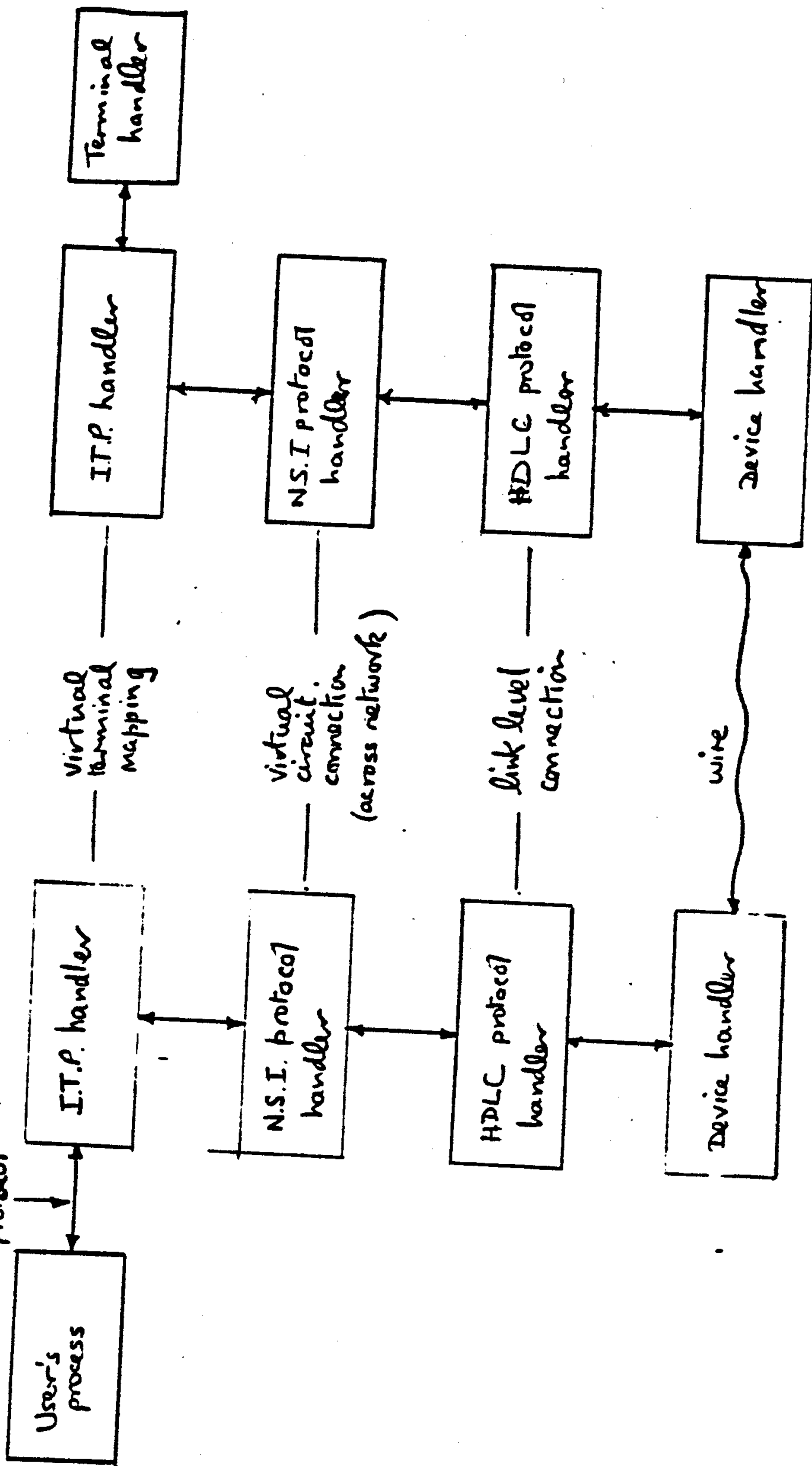
Figure 1.1: Hardware and Software Configuration
for ERCC 2970

In ICL 2970 and PDP11/34 front end

In 2970

In 11/34

2900 server/adaptor
protocol



Notes: I.T.P. - Interactive Terminal Protocol.
 N.S.I. - Node Standard Interface
 Data flow - →

Figure 1.2: RCO Network Protocol Hierarchy - Interactive

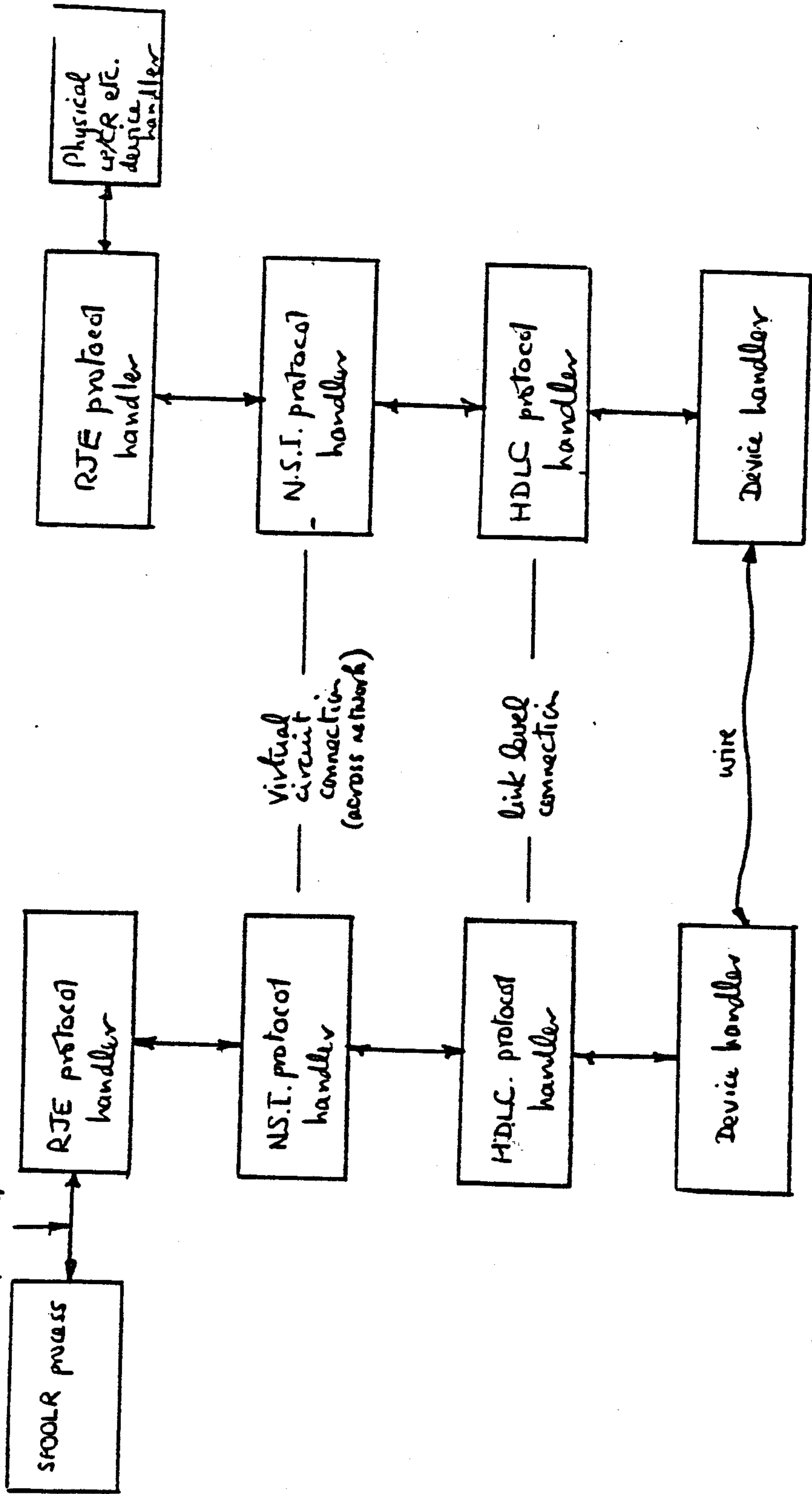
In ICL 2970 and PDP11/34 front end

In PDP11 Terminal Control Processor

In 2970

In 11/34

2900 Server/Adapter
protocol



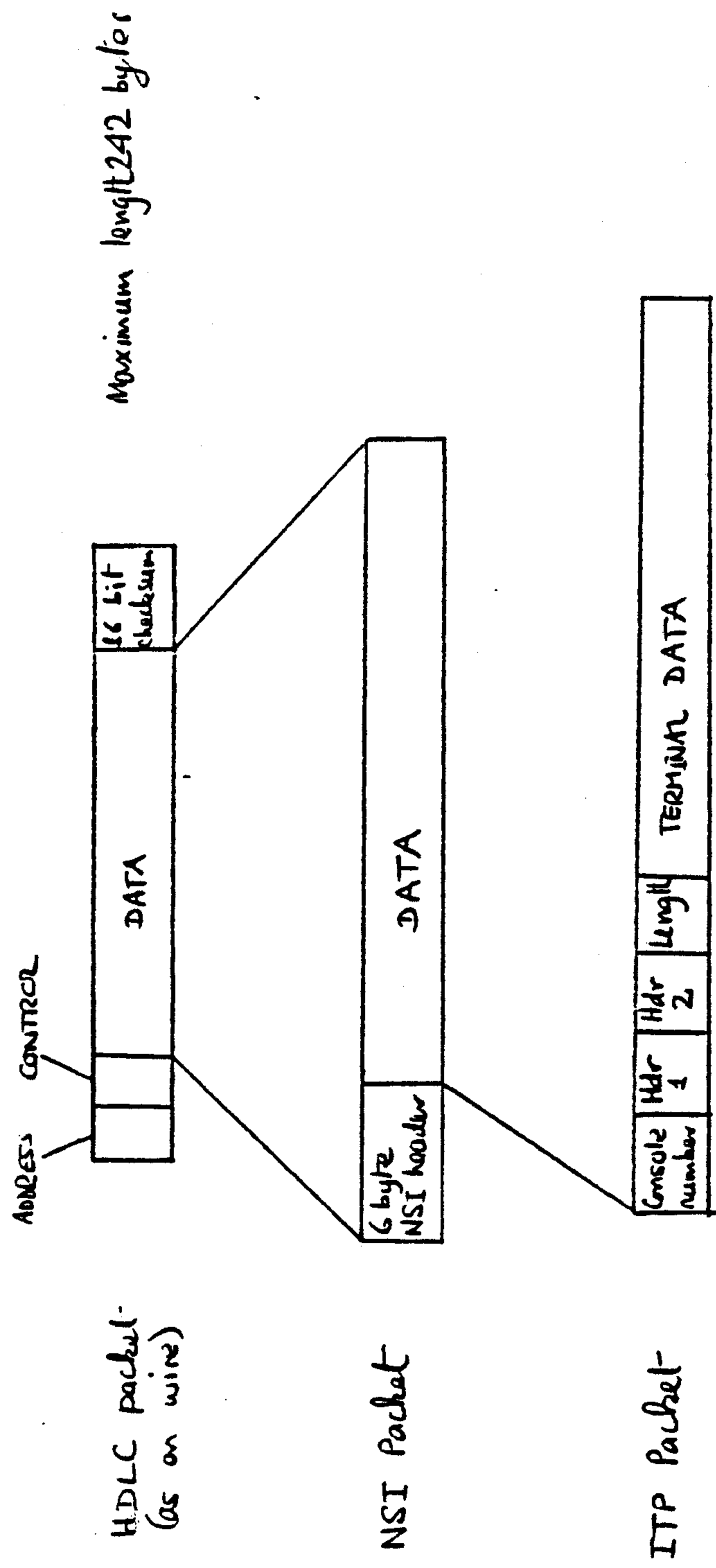
Notes:

N.S.I. - Node Standard Interface.

The structure for a File Transfer Protocol (FTP) is essentially the same, although it will normally be used for mainframe/mainframe communication.

Data flow -

Figure 1.3: RCO Network Protocol Hierarchy - Non-interactive



Notes: I.T.P. - Interactive Terminal Protocol
 NS.I. - Node Standard Interface

Figure 1.4: RCO Network - Packaging of Interactive Data

2. Software Components

The software consists of a number of co-operating programs, each running as a task under the operating system DEIMOS. (DEIMOS is a general purpose operating system designed for use in PDP11s with memory management.)

The left hand part of Figure 1.1 above shows the connections between the programs which drive the FEP. The arrows indicate the interfaces between the tasks. In addition all the tasks interface to the buffer manager and to the common TT output handler.

It should be obvious that nearly all the above tasks are critical; e.g. if the NSI handler crashes, then everything will stop. The only exception is the RJE handler, the failure of which will only bring down the RJE part of the service.

Each of the programs in the 2900 FEP is now summarised in turn.

1) AMI Handler [Task name: AM1H]

The AMI handler handles the "protocol" necessary on the AMI interface. For example, the 2970 will inform the handler that it is prepared to make a transfer on a stream 'n'. The handler determines what use the stream is put to, and instructs the appropriate protocol handler. The ITP handler, for example, will then read (or write) characters from the 2900, enclose the ITP packet headers round it and pass the "message" to the NSI interface module. This module adds on the NSI packet header, and passes the "message" to the CS link handler for transmission into the Network.

2) Interactive Terminal Protocol (ITP) Handler [Task name: ITPS]

This program handles the protocol-dependent features of the ITP. For example, after receiving user output from the 2900, it will add the protocol header to the output and, obeying the protocol flow control rules, will pass the output to the Network Standard Interface (NSI) protocol handler, for transmission into the RCO network.

In addition, the handler controls the virtual connections along which the data to and from each Terminal Control Processor (TCP) flows. Each TCP sets up one connection to the FEP along which all terminals connected to that FEP communicate.

3) RJE Protocol Handler [Task name: RJES]

This handler controls all remote file transfers to and from the EMAS 2900 mainframe.

Data formatting and flow control on each virtual connection is performed by this module. Messages from operator consoles, i.e. logons and queue enquiries, are passed through to the mainframe with the protocol headers removed. Messages to operator consoles are processed into the correct form and passed to the NSI handler.

4) Network Standard Interface (NSI) Protocol Handler [Task name: NSIH]

This handler processes the protocol-dependent features of the RCO NSI, passing the information contained within to the high-level protocol modules (ITPS and RJES), in a near network-protocol-independent manner. For example, when a high-level protocol accepts a data packet from the handler, it forms the network-dependent "SEND BLOCK RESPONSE" packet and transmits this through the network. In the case of a different protocol, e.g. X25 level 3, the handler would construct a "window change" or 'RR' packet from the same response from the higher level.

5) Computer Science Asynchronous Link Handler [Task name: CSLK]

The handler receives blocks from the NSI handler and transmits them, one character at a time, down the link. It blocks up the character record from the link and passes them in a block to the NSI handler. In addition, it contains watchdog timers to establish when the link goes down.

6) File Transfer Protocol (FTP) Handler

This module is still to be written; its function will be to transmit and receive files from the network in the EPSS Study Group 3 FTP protocol.

7) Buffer Manager [Task name: BUFF]

This program controls all the data buffers used in the FEP system. All other tasks send requests for buffers to the buffer manager and return buffers to it after they have been used.

8) HDLC Protocol Handler [Task name: PROT]

On systems which communicate to the network via a synchronous line an HDLC protocol handler is used instead of the CS link handler. This module performs the same functions as CSLK, but ensures an error-free path by implementing HDLC at the link-to-link level.

For details of how these tasks communicate with each other, and other features of the FEP system, consult the DEIMOS User Manual, by B.A.C. Gilmore (ERCC).

2.1 The AMI Handler

The AMI handler controls the PDP11 hardware interface with the 2900 mainframe. In the case of the 2900 P series the interface is via the GPC, with 2900 S series via the DCU. The handler also controls the low-level functions of the software "protocol" that is used to communicate between a 2900 and a PDP11.

The handler has three distinct functions:

- 1) To handle "internal" transactions with the 2900
- 2) To handle data transactions with the 2900
- 3) To interface to the higher-level protocol tasks

These are described in turn below (subsections 2.1.1 - 2.1.3).

2.1.1 "Internal" transactions with the 2900

There are a number of control signals identified by bit 0 set in the receive status register sent out by the 2900 which are completely handled by this task. The response to each is defined either by the 2900 interface hardware or software.

The functions, and their responses, are:

a) Send Property Codes

This is sent to the PDP11 on every IPL of the 2900, and is used by the 2900 hardware to establish how many front ends (if any) are currently available.

The response is 4 data bytes, followed by a normal primary status:

14,0,0,0, 16 + control bit.

The '14' defines a front end to EMAS 2900.

b) Identify

Identify is produced by the GPC microprogram (in a P series 2900) to determine which of the possible mechanisms on the interface has set an "attention bit" in primary status. This is not actually used in the FEP case, as there is only one mechanism, but nevertheless the identify must be replied to.

This response is two data bytes and a normal primary status:

0,X'80', 16+control bit.

c) Sense

This command is normally generated by EMAS 2900 software to obtain the secondary and tertiary status bytes after certain conditions, e.g. send property codes. Currently no "real" information is actually transferred to the mainframe.
Response: 2 data bytes and primary status:

0,0, 16+control bit.

d) Limit

Limit is sent by the 2900 when it has read the number of bytes specified in its current command. The action taken depends on the state of the handler. If it is handling one of the above control signals, e.g. identify only expects two data characters (so that a limit is seen before the primary status can be transmitted), then a primary status is sent and earlier, still pending, actions are ignored.

In the second case, the interface will be in a "user pending or writing to the 2900" state (see below). The action in this case is still to send a primary status.

There are other possible control signals, e.g. initialise, autoloading, which are not used by EMAS 2900. If any is actually received, the response is a primary status with "unsuccessful" set.

2.1.2 Data transactions with the 2900

All data transactions to or from a 2900 consists of two parts. First, a four-byte part describing the logical stream which the transfer is to be on (this also determines the direction), and the maximum length of that particular transfer. This four-byte "header" always goes from the 2900 to the FEP. The second part of the transaction is the actual data; this can be transferred in either direction.

From the point of view of the handler, the actual transaction is as follows:

- a) A write control command signal, followed by four data bytes - two for the stream number and two for the length of transfer, followed by a limit command - to indicate the end of that part.
- b) The handler must now send a primary status to the 2900, to signal the completion of the command.
- c) The 2900 will now send either a READ command or a WRITE command, depending on the direction of the transfer. A number of data bytes is thus transferred in the relevant direction and the command again terminates by a primary status.

There are two distinct types of data transfer: a control stream transfer, and a transfer on any other streams. The distinction is only made for the convenience of handling control stream transfers (see below) which are done by the AMI handler itself.

A control stream transfer (stream numbers -1 and -2) is done in units of 24 bytes. In the case of an outward transfer, the 2900 sends up to 8 units simultaneously. These are read by the handler, the stream number to which they refer is determined, and the entire message is then passed to the higher level protocol which "owns" that particular stream. In the case of inward transfers, the higher level protocol tasks send messages containing 24 bytes to the handler which queues them and, when permitted, sends up to 8 of them into the 2900 in any one transfer.

Control of transfers

If the link is idle, the 2900 may at any time start up a transfer to the FEP. If the FEP wishes to initiate a transfer to the 2900, it must send an ATTENTION to the 2900, which should then respond with a request to do an inward control transfer. If a transfer is in progress, the FEP may indicate its desire for a transfer by setting the ATTENTION bit in the next (and subsequent) primary statuses that it sends to the 2900.

Note: All transfers from the FEP to the 2900 are initiated with a control transfer on a stream that requests the 2900 to initiate an actual data transfer on the particular stream. Several of these may be outstanding at any time, and it is the prerogative of the 2900 to decide which it honours first.

Stream transfer

When a transfer is initiated on a stream other than the two control streams, the handler passes physical control of the link over to the higher level protocol task and lets it do the actual transfer. When it is finished, or when the other task senses a command on the link (e.g. a LIMIT), it passes control back to the handler, which performs the primary status.

2.1.3 Interface to the high-level protocol tasks

There are six messages from the handler to a higher task (parameters in brackets):

- | | | |
|---|---|----------------------------|
| 0 | Pass interface address
(physical address of interface) | |
| 1 | Do input (stream, max trf) | do data transfer from 2900 |
| 2 | Do output (stream, max trf) | do data transfer to 2900 |
| 3 | Message (address of message) | control message from 2900 |
| 4 | Mainframe up | state of mainframe |
| 5 | Mainframe down | state of mainframe |

There are five messages from high level protocol tasks to the AMI handler (parameters in brackets):

- | | | |
|---|---|---|
| 0 | Send high level control message
(address of message) | to be sent on inward
control stream |
| 1 | Send low level control message
(address of message) | to be sent on inward
control stream |
| 2 | Here I am (stream no) | claim of stream no |
| 3 | Return control (action) | pass control of link back
to the handler |
| 4 | Stop | tells the handler to close
down |

There are a number of actions that can be performed on "return control". The values and actions are:

- | | | |
|---|---|--|
| 0 | Send normal primary status | |
| 1 | Send normal primary status
+ short block | significant information
transferred, e.g. EOF |
| 2 | Send normal primary status
+ long block | transaction not satisfied -
2900 should put another one
on |
| 3 | Send unsuccessful primary status | |
| 4 | Send primary status + condition | |
| 5 | Same as 0 | |

In addition, if the top bit of the flag is set, then the handler should accept the test characters sent from the 2900; otherwise it should not.

2.1.4 AMI handler fault messages

The AMI handler produces error messages (on the FEP console) of the following form:

AMI: FAULT x n1,n2

where: x is a character indicating the class of fault
n1,n2 specify a subclass and/or give extra information

Note: except where stated otherwise, "input", "output",
"receiver" and "transmitter" are with respect to the PDP
11/34.

Classes

x= T - Timeout (character level) - a character from the 2900 was expected (i.e. in the middle of a sequence) but did not arrive within 20 ms (approx).

n1= state of receiver side of AMI
n2= state of transmitter side of AMI

x= I - Input fault

n1= 1 - after initial WRITE CONTROL (w.r.t. 2970), four characters should be received before a limit; n2 gives the number of characters actually received.

2 - a transfer request has been made by the 2970, on a stream that has not been CLAIMED by a higher level task; n2 is the stream number.

3 - a READ (w.r.t. 2970) has been issued on a stream >0 but a number of non-control characters have also been received; n2 is the number of characters in error.

4 - same as 3 except a WRITE (w.r.t. 2970) has been issued.

5 - an unexpected control character has been received from the 2970; n2 is the control character.

6 - non-control characters have been received before a control character; n2 is the number of characters.

x= S - A CONTROL message (high or low level) has been issued to a stream that has not been CLAIMED by a higher level task.

n1= number of offending stream
n2= offset into the entire block received from the 2970

x= 0 - Unexpected OUTPUT interrupt. An interrupt has been generated on the transmitter side of the AMI - this should not have occurred.

n1= 0
n2= output state of AMI

x= E - Transmitter state fault. The transmitter side of the AMI has got into an error state.

n1= output state of AMI
n2= previous output state of AMI

x= R - Return control fault. A RETURN CONTROL has been received from a higher level task at a time when the AMI handler has not released control.

n1= input state of AMI (w.r.t. 11/34)
n2= output state of AMI (w.r.t. 11/34)

x= C - Clock timeout (or major timeout). A request for transfer on the control stream, or an output interrupt, has been outstanding for 15 seconds. This will generate a message to the higher level tasks informing them that the 2970 has gone down.

n1= 2 - transfer outstanding
n2= 1 - interrupt outstanding

3. Operating the EMAS 2900 Front End

The following description relates specifically to the operation of the FEP on the ERCC 2970.

The EMAS 2900 Front End Processor, a DEC PDP 11/34, is connected to both GPCs on the 2970 mainframe by a locally produced NRPI link. Figure 3.1 illustrates this connection.

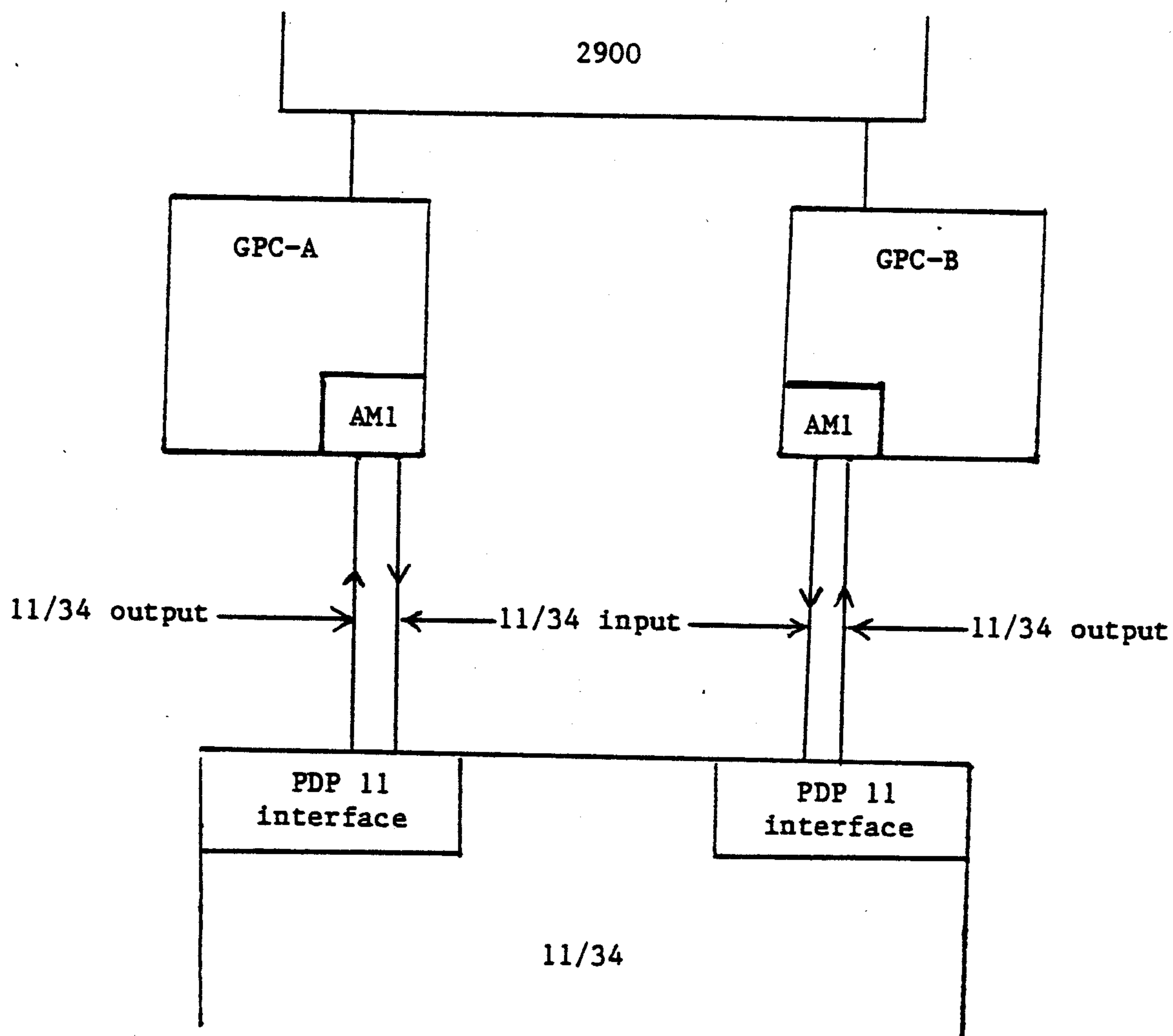


Fig. 3.1: Connection of FEP to Mainframe

As explained above, the FEP is linked to the 2970 via the AM1 interfaces. It uses only one of these - the other is for backup. The FEP is also linked to the PDP 11/45 Network Control Processor via a 1 Megabit Computer Science asynchronous link. All interactive and RJE traffic to the 2970 uses this link.

3.1 Loading Procedure

The procedure is as follows:

- * If the RUN light is on, hold the CTRL button down and press HALT, then release the CTRL button.
- * Boot the machine: hold the CTRL button down and press BOOT, then release the CTRL button.
- * The bootstrap should now identify itself with something like the following (the actual numbers do not matter):

010000 160200 170000 000002

To load the system from disc drive 0:

- * Ensure the LOAD/RUN switch is at RUN.
- * Type DK (cr).
- * After a pause the system identifier "DEIMOS VSN 6" appears, and after a further pause of a few seconds the system prompt "x" appears.

The programs that drive the link etc. must now be loaded, by typing:

LOGON 15 (cr)

When the prompt reappears, type either

GPCA (cr) or GPCB (cr)

depending on which connection to the mainframe is currently in use.

The Front End is now ready for use.

Notes:

- * In this section: (esc) = ESCAPE, (cr) = RETURN, _ = space.
- * Ensure that both GPCs are switched on.
- * "ATTACHED OK" indicates that the NCP (11/45) has recognised the 11/34.
- * Each TCP, as it "connects" to the 11/34, produces a message of the form:

TERMINAL_x_CONNECTED

- * After a reload of the NCP (11/45), connection of the 11/34 must be re-established by typing

(esc) INT_R_CSLK (cr)

3.2 Powering Off Procedure

- 1) The Front End should be closed down by typing on its console:

(esc) INT_D_GATE (cr)

The reply

GATE: REMOVED OK

should be received.

- 2) The Load/Run switch on the disc should be switched to LOAD.
- 3) The machine should be halted by holding down Ctrl and pressing Halt.
- 4) Finally, power the machine off.

3.3 Dumping the System

A dump should be taken if the 11/34 stops; that is, if the run light goes out. The sequence for taking a Front End dump is:

- * Holding CTRL down, press HLT.
- * Press CLR.
- * Press 1 then 4 then 0.
- * Press LAD.
- * Holding CTRL down, press START; the disc lights on unit 0 should blink and the run light will go on then off again.
- * Fill in the front end dump log and process the dump (See 3.4).

3.4 Processing a Front End Dump

A front-end dump always occupies the same site on the RK05 system disc and it is important to process this dump immediately before reloading the service system, as any future dump would overwrite the one already on the dump site. Three other sites are reserved on the system disc to hold processed dumps and these should be used cyclicly and their use recorded in the Front End dump log. The three sites are referred to as DMP001, DMP002 and DMP003.

The procedure is as follows:

- * Load the Deimos operating system, as in 3.1, until the first system prompt appears.

- * Log on to user number 16 rather than user 15:

 XLOGON_16

- * Process the dump using the program DPAL and the following replies to its prompts:

 X DPAL_/DMPnnn where nnn specifies the next dump site

 DPAL: T
 DPAL: M
 DPAL: Q_47
 DPAL: Q_51
 DPAL: Q_52
 DPAL: S

- * Continue with the normal load as described in 3.1 by logging on to user number 15.

3.5 Responding to crashes

The 2970 FEP will normally run without the need for operator intervention. However, under certain circumstances action is necessary.

1. GPC TIMEOUT + FEO DOWN on the 2970 Oper

a) If this message is accompanied by

AM1: Fault C 2, 0

EMAS 2900 DOWN

on the FEP console, then a dump of the FEP is not necessary, but

(esc)INT_F EAM5

should be typed on the FEP console, to obtain monitoring information. Once it finished, a reload of the 2970 will be necessary, but not a reload of the FEP.

b) If a message of the form:

xxxx: BAD SEGMENT - followed by two numbers	or
xxxx: ADDRESS ERROR	or
BUFF: NO SMALL BUFFERS ***	or
BUFF: NO BIG BUFFERS ***	

has been printed on the FEP then a dump (see 3.3) is necessary.

N.B. The last two messages above can appear without the system crashing; a dump should only be taken if the FEP is not responding to users.

2. If no users can log on to the 2970 and the 11/45 front end to the 4/75 link is up

In this case, the 2970 FEP should be dumped and then reloaded.

B.A.C. Gilmore
J.M. Murison

EMAS 2900 HARDWARE NOTE

No: 1
Date: 17/2/78

Information on 2900 Image Store

(Received verbally by J.K. Yarwood from Alan Darling and Bill Kell, 4th May, 1977; updated June 1977; updated with the help of Pat Ryan January 1978.)

The purpose of the discussion was to establish differences between the 2980 and the 2976, 2970 & 2960 with a view to trying EMAS 2900 on the former. The following table summarises the model-dependent addresses and uses for the four machines:

	2960.Bits	SML16 2970.Bits	2976.Bits	SML12 2980.Bits
<u>OCP Registers</u>				
LST Base	00006001	00006001	402A	402A
LST Limit	6000	6000	402C	402C
PST Base	6003	6003	4029	4029
PST Limit	6002	6002	402B	402B
Handkeys	6006	6006	4205	4205
Hooter (set bit to commence hoot; clear bit to terminate hoot)	6008 31 (Optional)	6008 31	4013 19	4013 19
System Interrupt Register (Bit set to indicate interrupting Port)	600A 28-31 for Ports 0 to 3	600A 28-31 for Ports 0 to 3	4014 22-27 for Ports 0 to 5	4014 22-27 for Ports 0 to 5
Real time clock (X)	600C	600C	44pt4000	44pt4000
Real time clock (Y)	600D	600D	44pt4100	44pt4100
Real time clock (Z)	600E	600E	44pt4200 (see footnote)	44pt4200 (see footnote)

2960.Bits SML16 2970.Bits 2976.Bits SML12 2980.Bits

OCF Control Registers

Control (inhibit when set)	6011	6011	4013
Clear slaves - write only (reads 0)	0		
Inhibit CPRs		24	
Inhibit main store		25	
Inhibit retry		26	13
Inhibit pipeline		27	12 (non overlap)
Inhibit stack slave		28	20
Inhibit instruction slave			12 (non overlap)
Inhibit operand slave	28	23	23
Inhibit full photo	31	28	28
Lock translation slave rejection		25	25

Software Gen. System Error	6013	6013	4013 17
Register Load ('Activate')	6014	6014	not implemented

SMAC Image Store

SMAC config. register (one per SMAC)	4C0s6A20	4C0s6A20	4Cs04A20
(s = SMAC ident no, in range 0-F)			
Available blocks	8-23 bit 23=block 0, bit 22=block 1, bit 21=block 2, etc	8-23	7-22 bit 7=block 0, bit 8=block 1, etc
SMAC blocksize	128K	128K	256K

2960.Bits SML16 2970.Bits 2976.Bits SML12 2980.Bits

SAC Trunk Register Image Store (2980 only)

DATA IN (B->A)

SPAD { 40pt4000 0-31
contents { 4100 32-63
(i.e. RAM) { 4200 64-95
 4300 96-127

SAC Registers

These are all model-independent.

[Footnotes:

- 1) 2980 Real-time Clock

pt is where you get the IPL interrupt from: p is normally 0 and t is "don't care" (zero).

You must also set the appropriate port bit in 4012 (bits 16 to 25 - Ports 0 to 5) to allow SAC external interrupts to reach the OCP.

The Real-time clock on the 2980 is out in a (the) SAC.

The 2976 is as the 2980 but p' = OCP port.

- 2) 2960 Real Addressing Mode

This is now thought to be available.]

J.K. Yarwood
P.D. Stephens

EMAS 2900 HARDWARE NOTE

No: 2

Date: 15/12/77

ICL 2970 Engineer's Panels

The attached description of the 2970 Engineer's panels has been taken from two ICL technical documents (diagrams: 65/94562, text: 65/94563); it describes the current state of the panels, i.e. following the SML19 hardware upgrade.

The question of providing this information was raised originally by my receipt (via John Maddock) of a similar document produced by the Post Office, relating to the Barbican machine. A number of people using the 2970 expressed an interest in it. However, although the P.O. document contains a slightly fuller description of the handkeys and indicators than the attached note, it is now out of date as a result of SML19. Nonetheless, much of the information in it is still relevant to our 2970; I have kept it and can give anyone interested a listing. It has been filed as Hardware Note 2a.

I should like to thank Les Nowbray (ICL 2900 Engineer) and John Maddock for their help.

John M. Murison

LETTER
MADDOCK

1 GENERAL LAYOUT

The OCP Engineer's facilities comprise 19 rows of monitor lamps with 72 lamps in each row & 1 row of monitor lamps with 36 lamps and 3 rows of handkeys with 36 in each row. The layout of handkeys and monitor lamps are shown on OD61 to 64 of document 65/94562 and their operational usage is described in the following text.

2 MONITOR LAMPS (Brackets at row number indicate MDP socket)

Row 1 (1R)	The group of seven lamps at the left-hand end of this row display ARU Fails; AK, LX, AL, AQ, XC, AN and MF. The remaining lights display the ARU K highway (bits 00-63).		
Row 2 (2R)	The ARU Parity Flags are indicated by the group of six lamps situated at the left hand end of this row; AK, LX, AL, AQ, ASE (Inhibit System Error), and FP (Force Parity error). The remaining lamps display the ARU L highway (bits 00-63).		
Row 3 (3R)	The ARU AA and AP parity stats. (E, A and L) are displayed at the left hand end of this row. The remaining lamps are devoted to displaying the ARU Q highway (bits 2-63).		
Row 4 (4R)	The ARU AS and RAM parity stats. (E, A and L) are displayed at the left hand end of this row, the remainder display the contents of the ARU Ram (00-63).		
Row 5 (11R)	RIF	00 - 06	} IFU
	RIK	00 - 06	
	RHF	00 - 06	
	RHK	00 - 06	
	IL	00,01	} PC
	ISL	27,28	
	RIF	27 - 30	
	RHF	27 - 30	
	FI		} Function Dependant Flags
	FSK		
	FTOS		
	FD		} AGU Microprogram Flags
	FX	General	
	FY	Purpose	
	FZ	Flags	
	PTS		
	JAM	Real Address Mode	
	D	PSR D bit see IM110 for fuller explanation	
	PRIV	Privilege	
	DGW	Diagnostic Write	
	MK	E,A,L Parity Stats	
	RMFP	ARU Mask and Filler Parity	
	RMF	00 - 15 ARU Mask and Filler.	

Row 6 (12R)	RGF	00 - 06	AGU Function Register
	RGK	00 - 05	
	ROP	00, 14 - 31	
	IBE		Instruction Buffer empty.
	SUCJ		Jump condition satisfied.
	EP		Event pending
	RAN	P, 00 - 07	ARU AN register
	RAP	00 - 03, 62, 63	ARU AP bits
	RMK	30, 31, 62, 63	ARU MK bits
	MILLSH	OM, OL, 01-05	
	APSH	00-02	AP shift
	MK	01, 02	
	AA	01	
	RAG	P, 00-03	
Row 7 (13R)	PC	00-30	AGU PC
	BV		AGU B Overflow.
	CC	00,01	AGU CC
	GAZ	00,01	AGU ACC size.
	GOZ	00,01	AGU Operand size.
	GTWIST		AGU TWIST stat.
	RRF	00-07	Buffer Function Register.
	MPSLD		Loading mps (HWV7 or overlay)
	FMP		Microprogram loaded flag
	FHMP		Hardwired microprogram flag
	SPX	00,01	ARU Scratchpad ext. bits
	AT	00,01	ARU Arithmetic type.
	DS		ARU Double/Single length shifting.
	JF		ARU Jump Flag.
	JFA		ARU Jump Flag A
	SA		ARU Address SPAD from AN.
	RE		ARU Register Extension
	LF		ARU Link register from microprogram bits.
	SPOS		ARU Sign Positive (Dec. arithmetic).
	CY		ARU mill carry flag.
	PM		Program Mask (PSR)
	P+C	00,01	ARU Multiplier.
	CS		ARU Carry Save.



Row 8 (14R)	WA	00-31	AGU WA register.
	EMFV		Emulation Floating point overflow
	V		Common overflow
	CC	00,01	Common CC
	RAZ	00,01	Buffer ACC size
	ROZ	00,01	Buffer Operand size
	ARUTWIST		ARU TWIST Stat.
	RAF	00-06	ARU Function Register.
	FBOOTS		Hardwired Bootstrap.
	LOAD DEC	00-03	Direction bits.
	GMPS AD DEC	00-02	AGU Microprogram Address selection.
	MPS DATA	00-13	AGU MPS Data Buffer (see Row 9).
Row 9 (15R)	WB	00-31	AGU WB register.
	AV		ARU Overflow
	CC	00,01	ARU CC
	NEW AZ	00,01	ARU New Instruction ACC size.
	OLD AZ	00,01	ARU Previous Instruction ACC size.
	MPS DATA	14-44	AGU MPS Data Buffer.
Row 10 (16R)	SB	00-31	AGU SPAD output highway.
	SB	P0-P3	AGU SPAD output highway parity
	MPS CURR ADD.	00-11, P	AGU Microprogram Current Add.
	MPS ACC ADD.	00-11, P	AGU Microprogram Access Add.
	NULL		Null Beat (AGU)
	FADD		Forced Address (AGU).
	FWD		
	STOLK3		AGU Lockout.
	LKS1		AGU Lockout.
	LKS2		AGU Lockout.
	ADD LKS		"
	RUN LKS		"
	STL1		"
	STL2		"
Row 11 (17R)	K	00-31	AGU K highway.
	DT	00,01	Descriptor Type.
	DZ	00-02	Descriptor Size.
	A		A bit.
	USC		Unscaled.
	BCI		Bound Check Inhibit.
	FRAM	00-06	Function RAM Addressing.

	KRAM	00-06	K Field RAM Addressing.
	IRAM	00-06	Descriptor RAM Addressing.
	VI Access	00-02	Volatile Index Address Selection.
	VI ADD	00-07	Volatile Index Addressing.
Row 12 (18R)	L	00-31	AGU L highway.
	FSP 2 & 3		AGU SPAD Flags.
	SPAD ADD	00-03	. " " Addressing
	OTB		Valid Stats. (Microprogram Address valid for replaced jumps if required).
	GOB		
	DTB		
	DZB		
	INTB		
	RLOTB		
	REALER		
	ARUSTP		
	AMP AD DEC	00-02	ARU Microprogram Address selection.
	REPLMT		
	FPNXT		Next ARU func. is floating point.
	RFB		AGU/ARU Interlocks.
	RBB		" "
	AAR		" "
	OPINT		Operand in Internal Format.
	ACINT		ACC in Internal Format.
	AOK		
	BUF 2F		Association Address
	SSNEQ		Stack Segment Number Equivalence
	ISL		Instruction Slave
	SSL		Stack slave
	IB		Slave hit
	GBM		
	GBL		
	RBM		
	RBL		
	LKS 1		ARU Lockout.
	LKS 2		"
	PLKS		Pipe Locks



Row 13 (19R)	Q	00-31	AGU Q highway.
	FSP0 & 1		AGU SPAD Flags
	MPS	00-35	ARU Microprogram word.
Row 14 (20R)	GBM	00-31, 0P-3P	AGU GB Most.
	MPS CURR ADD	00-11, P	ARU Microprogram current address.
	MPS ACC. ADD	00-11, P	ARU Microprogram Access address.
	WFL		ARU Wait for Literal.
	FADD		ARU Forced Address.
	VIADD	00-06	ARU Volatile Index Address.
Row 15 (21R)	GBL	32-63	AGU GB Least.
	GBL	04-07	AGU GB Least Parity
	TA	04-29, P	Store Address, (Real Address).
	BYTE FUNC	00-07, P	
Row 16 (22R)	PORT 0 SEINT		Direct wired interface interrupts.
	PORT 0 EXINT		
	PORT 0 OPINT		
	PORT 1 SEINT		
	PORT 1 EXINT		
	PORT 1 OPINT		
	PORT 2 SEINT		
	PORT 2 EXINT		
	PORT 2 OPINT		
	PORT 3 SEINT		
	PORT 3 EXINT		
	PORT 3 OPINT		
	MVSI		Masked Virtual Store Interrupt.
	MPE		Masked Program Error.
	MSCI		Masked System Call Interrupt.
	MOUT		Masked Out Interrupt.
	MXC		Masked Extracode.
	RGN	24-31	AGU GN
	RGNZR		AGU GN bits 8-23 = zero.

Row 16

Cont.

ACCESS E		Store Access Execute.
ACCESS W		Store Access Write.
ACCESS R		Store Access Read.
APF	03-11	Access Protection Field bits.
ACR	00-03	Access Control Register.
PERF		Permission Fail.
MEQ		Multiequivalence.
TAC		ATU Accept (Hit).
UMS		Update Miss (Use bits).
MS		Miss.
MISS		
PAGED		Paged segment address.
VAM		Virtual Address Mode.
SIP		Search In Progress (TS).
LOAD VIR		Load Virtual Address.
LOAD PAP		Load Page and Permission bits.
LOAD RRE		Load Real Even address.
LOAD REO		Load Real Odd address.
WD M, L		
CPR	P0-P2	ATU CPR Pointer.

Row 17 (23R)

SSE	}	Interrupt Register
GPE		
VSI		
EP		
OUT		
XC		
SE		
SGSE		
RTC		
XACT		
HRI		
APE		
IT		
IC		

Row 17
Cont.

FKI	Function K and Indirect RAMS	
ASE	ARU System Error	
IB	Instruction Buffer	
RC	ARU control	
GC	AGU control	
GK	K highway	
GL	L highway	
GQ	Q highway	
GKY	Carries	
GMISC	AGU Misc.	
SSD	Stack Slave Data	
SSA	Stack Slave Address	
ISD	Instruction Slave Data	
ISA	Instruction Slave Address	
SMAC		
SDIE	Software DIE	
DIE	Hardware DIE	
TRANS		
CHECK		
MODE		
PERMIT		
BUSY		
RIP		
REQ	Current stats. of ATU	
APHIP	ARU Photograph in Progress	
CASS	Cassette is Busy	
FEM	Emulation mode flag.	
EM 0-3	Emulation mode.	
H BUSY	Hash Busy	
STREQ	Store Request	
MPREQ	Microprogram request	
OFFLINE		
PERI	Perifail	
STORE	Store Fail	
UNSERVED		
CNT	Count Fail	
MPSINH	MPS Inhibit Fail	
RAM Add. X0-3	ATU RAM Address.	
Y0-3		
Y01, Y23	ACR use bit checks RAM address since only one ACR use bit is held for each Virtual Address Pair (even and odd).	

Parity Fail Stats.

Requests from input device
to main and Microprogram storeDiagnostic Unit
Input Device
Fails



Row 18 (24R)	WD	00-63, P0-P7	Write Buffer store.	
Row 19 (25R)	ARU COPY BUFFER	00 - 11		
	AGU COPY BUFFER	00 - 11		
	1900 EM LINK	00 - 11	1900 Emulation Link (AGU)	
	AGU SR	00 - 11	AGU SUBROUTINE	
	AGU LR	00 - 11	AGU Link	
	INHAM		Inhibit Hamming Interrupt	} CW STATS
	INSTO		Inhibit Main Store	
	IRTY		Inhibit Retry	
	ISRS		Inhibit Successful Retry Reporting	
	ISTSL		Inhibit Stack Slave	
	IINSL		Inhibit Instruction Slave	
	ICPRS		Inhibit Current page/segment Registers	
	IPHOT		Inhibit Photo	

External Monitor Strip

OPTINT	0,1	Peripheral Interrupt
IDLE		
OPERAND	14-31	
HMP		Hard Microprogram
GMPADD	8-11	AGU Microprogram Address
SEINT	0,1	System Error Interrupt
SMACFL		SMAC Fail

HANDKEYS

The handkeys are arranged in three rows A, B and C and are depicted on OD61 of document 65/94562.

Row A Handkeys 1 - 32 are used for setting up Image Store data.

Handkeys 33 - 36 vary the OCP Timing Margins as follows:

- a) Key 33 AGU FAST down - All AGU timing speeded up by 5%
- b) Key 34 AGU SLOW down - All AGU timing slowed down by 20-25%
- c) Key 35 ARU FAST down - All ARU timing speeded up by 5%
- d) Key 36 ARU SLOW down - All ARU timing slowed down by 20-25%

Note: The results of setting AGU FAST and AGU SLOW or ARU FAST and ARU SLOW is indeterminate.

- e) All four switches in centre position - timing at nominal speed.
- f) Key 33 up - Individual AGU timing pulses as determined by a decode of the remaining 3 switches speeded up by a constant 5 n Sec (see table 1).

Switches			AGU Timing Pulse Speeded up by 5n Sec.
KEY 34	KEY 35	KEY 36	
Centre	Centre	Centre	T08TM
Centre	Centre	TDEC2	T14TM
Centre	TDEC1	Centre	T17TM
Centre	TDEC1	TDEC2	T19TM
TDEC0	Centre	Centre	T12DF
TDEC0	Centre	TDEC2	T12TM
TDEC0	TDEC1	Centre	T17GMIL
TDEC0	TDEC1	TDEC2	Conditional Jump

Table 1. AGU timing Pulse Margin Speed Up

- g) Key 33 centre - Individual ARU timing pulses as determined by a decode of the remaining 3 switches speeded up by a constant 5n Sec (see table 2).



Switches			ARU Timing Pulse Speeded up by 5n Sec.
KEY 34	KEY 35	KEY 36	
Centre	Centre	Centre	All AGU and ARU Timing Nominal
Centre	Centre	TDEC3	T11DF
Centre	TDEC1	Centre	T10TM
Centre	TDEC1	TDEC3	T08RTIM
TDEC0	Centre	Centre	T12TIM
TDEC0	Centre	TDEC3	T12RTIM
TDEC0	TDEC1	Centre	T17RTIM
TDEC0	TDEC1	TDEC3	T17MILL

Table 2. ARU Timing Pulse Margin Speed Up

Row B This row of keys enables the OCP to stop on a specified address, stopping on either microprogram equivalence (handkeys 1 - 12) or, on PC and ZA equivalences (keys 1 - 31). The specific equivalence on which the OCP stops is determined by the four keys situated at the extreme right-hand end of this row (keys 33 - 36).

MISC STOP, Key 32 down, stops the OCP when the AGU Miscellaneous Function 'STOP ON MISC' is set.

Stop on Microprogram loaded/PC; Key 33 enables the OCP to be stopped when the microprogram has completed loading by detecting #F (Stop character) in the direction bits. The other position of the key "Stop on PC equivalence" stops the OCP immediately equivalence is detected between the bits set-up on the handkeys and PC providing that "Stop on AGU/ARU Microprogram address" is not set. The OCP will resume running when the key is returned to the centre off position.

Stop on - ARU/AGU microprogram address; Key 34 enables the OCP to be stopped when bits 00-11 of the ARU/AGU microprogram address are equal to the bits set up on the keys. The OCP will resume running when the handkey is returned to its centre-off position.

When both the "Stop on PC" Keys 13-31 and "Stop on AGU microprogram address" 1-12 are set, then the OCP will only stop when equivalence is detected between these conditions and that set up on the keys.

ZA Read/ZA All and ZA Write/ZA Execute; Keys 35 and 36 enable the OCP to be stopped when ZA (which normally contains the virtual address) is equal to the bits set on the keys. The OCP will resume running when the appropriate key is returned to its centre-off position.

Keys 1 - 12 can also be used to set up the Port, Trunk and Stream when doing a Bootstrap or dump (Row C handkey 34).

FEM, Key 13 down, stops on selected address (Keys 1 - 12) when in emulation mode and stops on selected address in normal mode when key in neutral position.

A further facility exists on this row in that the group of keys at the left-hand end can be set to force either the ARU K (AK 00-02) L (AL 00-02) and KX (AKX 00-02) highways and the ARU Scratchpad (ASP 00-03) and AC → AK, or alternatively the AGU Scratchpad Flags (0-3) and AGU Scratchpad, and J Highway. The condition to be displayed is dependant upon the setting of the Display ARU/ Display AGU handkey (key 31 Row C).

Row C

KEYS

1	{ - ENG	Enables Engineers panel and disables OPER
2	{ FORCE IT INHIBIT IT	Force Interval Timer interrupt Inhibits interrupts from the Interval Timer and RTC
3	{ RAM ACR0	Force Real Address Mode Force ACR0
4	{ STOP ON SMAC FL IGNORE SMAC FL	Allows OCP to stop on unrecovered SMAC Fails Allows OCP to ignore unrecovered SMAC Fails
5	{ STOP ON SE FL IGNORE SE FL	Stops OCP on System Error Fail Allows OCP to ignore System Error Fail
6	{ INHIBIT CPR EX INHIBIT CPR ALL	Prevents the CPR's being used during execute accesses Prevents the CPR's being used during all store accesses
7	{ - INHIBIT PORT ACT	Prevents the OCP from taking any action as a result of receiving an ACTIVATE
8	{ - INHIBIT PORT INT	Prevents the OCP from raising an interrupt as a result of receiving a port interrupt.
9	{ - STOP ON SACTOUT	Causes OCP to stop on SAC Time out
10	{ - INHIBIT IC	Inhibits the instruction counter interrupts
11	{ STOP ON TIMEOUT IGNORE TIME OUT	Causes OCP to stop on any time out Allows OCP to ignore any time out

KEYS

12	{ RESET FAILS INHIBIT PHOTO	Resets any outstanding fail Allows the system Error Interrupt routine to omit the Photo of the OCP hardware
13	{ — INHIBIT INSTSL	Inhibits the Instruction slave
14	{ RESET LOCKOUTS INHIBIT INTLKS	Reset any Lock outs (or any timing interlocks) Allows AGU and ARU to run irrespective of any interlocks that may be set.
15	{ — INHIBIT STK SL	Inhibits the Stack Slave causing all read accesses to come from main store.
16	{ — INHIBIT HAM INT	Inhibits any Hamming Interrupt
17	{ INHIBIT SRR INHIBIT RETRY	Inhibits successful Retry reporting Allows the system Error Interrupt to by pass the hardware retry.
18	{ — ENABLE SMAC FRZ	This key allows the OCP general reset (FREEZE) to be passed via a Port to the SMAC.
19	{ DUMP —	Results in a PAW Function of Dump being sent to the Controller selected on the Keys of Row B when the RUN Key (35 Row C) is set OCP then stops.
20	{ — STOP ALL	Allows Stop on nulls (AGU and ARU)
22	{ STOP AGU AGU LOOP	Stop AGU Clocks. Inhibits the strobing of the AGU microprogram data buffers, thus enabling a specific beat to be repeated indefinitely.
23	{ STOP ARU ARU LOOP	Stop ARU clocks. Inhibits the strobing of the ARU microprogram data buffers thus enabling a specific beat to be repeated indefinitely.

25 { —
EKS

This key is the M.S.B of the Engineers Facility Field and also the drop through Idle.

26-29 { EF00
01
02
03

These keys have two functions:-

(X) They are looked at, at Microprogram load time (IPL), and if the decode is #5, the VAR Register is loaded from the Image Store Keys Row A.

(Y) These four Keys and EKS (Key 25) provide a means of forcing several functions manually provided the Engineers facility (Key 32 Row C) is operated. The OCP should be at Real OTB. Key 35 Row C should be in OI position. The Engineers facility is performed by setting the decode, then pressing and releasing the Engineers Facility Key (Key 32 Row C), followed by the operation of the GO Key (Key 36 Row C).

The facilities provided are as follows:-

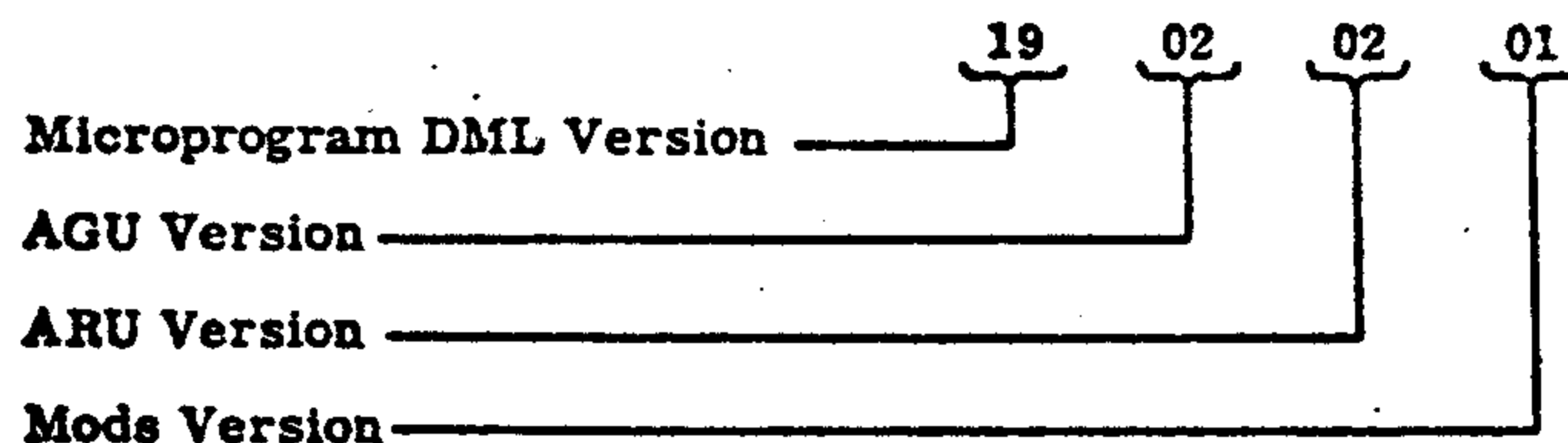
Engineers Facilities.

0
1 PC JUMP
2 MPVERS

No Action

Jump to Address on Image Store data keys Row A as per address mode, Virtual or Real.

Microprogram Version displays on the AGU Register WB display on Row 9. The decode is Hexadecimal. For example:



3-7
8 READ

No Action.

Read from virtual address in Store.

The address is set on Image Store Keys Row A.

The information read is displayed in Registers GBM and GBL Rows 14 and 15.

9	READ +	Read from virtual address is store. The address is set on Image Store Keys Row A. The information read is displayed in Registers GBM and GBL Rows 14 and 15. EKS key being pressed and released increments the virtual address by 2 words and the new information is displayed in GBM and GBL Registers. The microprogram can be returned to real OTB by returning EF 00-03 keys to neutral and pressing and releasing EKS Key.
A	Read R +	As 8 (READ) but the Real Address Mode Key (Key 3 up Row C) operated.
B	Read R +	As 9 (READ +) but the Real Address Mode Key (Key 3 up Row C) operated.
C	WRITE	Write to the virtual address set on the Image Store Keys Row A upon the operation of the GO Key (Key 36 Row C). The machine will now wait for data to be set on the Image Store Keys. When EKS Key is operated this data is written into the virtual address in store
D	WRITE +	Write to the virtual address set on the Image Store Keys Row A upon the operation of the GO Key (Key 36 Row C). The machine will now wait for data to be set on the Image Store Keys. When EKS key is operated this data is written into the virtual address in store. Further press and releases of the EKS key increment the address by 2 words and write the data on Image Store Keys. The microprogram can be returned to real OTB by returning EF00-03 keys to neutral and pressing and releasing the EKS key.
E	WRITE R	As C but Real Address Mode key operated (Key 3 up Row C).
F	WRITE R+	As D but Real Address Mode key operated. (Key 3 up Row C).
10	MOVEOVER	Move overlays from existing to new area of store set on Image Store keys (Row A).
11	FOTO	Force Photo, done automatically by this key.
12	READ INST SLAVE	Read the line of the Instruction Slave defined on Image Store Keys, and appears on monitor lights GBM and GBL Rows 14 and 15.
13	READ STACK SLAVE	Read the line of the Stack Slave defined on Image Store Keys, and appears on monitor lights GBM and GBL. Rows 14 and 15.
14,15		No Action.

16 READ AND WRITE SAC AND SMAC I.S.

To read set the Engineers Facility keys, set the image store address as defined below *, and perform the functions described in paragraph (Y), preceding the Engineers Facilities. The information in GBM is the Image Store address which has been set up and in GBL is the information Read.

To Write set up Image Store Keys Row A, perform the Engineers Facility operation described in paragraph (Y) but after the GO key has been pressed the OCP will wait for data which is to be set up on Image Store Keys. EKS key is pressed and released this causes the Image Store write action to be completed.

* Image Store Keys Row A.

Bit 0 :- 0 Read, 1 Write

4,5:- Both down :- SMAC; 5 only :- SAC; No keys operated:- SAC Trunk.

SMAC	{	Bits 8-15 SMAC number
		Bits 16-31 SMAC Image Store Address (see N. R. S. D 4, 2, 3 G)

SAC	{	Bits 8-11 Port number of SAC
		Bits 16-31 The Image Store Address

17 No Action

18 WRITE SPADS

On image Store keys 24-27 put the S. P. Flags, and on keys 28-31 put the S. P. line number. The Image Store Engineers Facility action described in paragraph (Y) is performed. The data to be written to the Scratch Pad is set on the Image Store Keys. EKS key is then pressed and released.

19 UPDATE SP

This updates the visible Registers to the current machine state automatically

e.g. ACS, CC, OV in PSR

SP SA (IT Copy)

SP PC

SP SW (A0 Copy)

SP SD (IC Copy)

SP SX (A1 Copy)

SP SY (A2 Copy)

SP SZ (A3 Copy)

30 { INHIBIT
STOP

Used in conjunction with stop keys to provide Engineers Scope trigger.

31 { DISPLAY
ARU
DISPLAY
AGU

This key causes the Register(s) selected by the keys on row B to be placed on the highways which in turn are displayed on monitor lamps.

32 { ENG
FAC
ORDER

Enables the Five Keys 25-29.

The ORDER position allows the primitive interface order set up on ROW A to be obeyed. If only one order is required the RUN key (35 Row C) is placed in the appropriate position and the GO Key is pressed once.

33 MPSLOAD ACTIVATE

When operating the MICROPROGRAM LOAD switch it is assumed that the microprogram is in the main store starting at #100 BYTES, the RUN switch (Key 35) should also be set. The microprogram is then loaded and entered and if it is a normal Microprogram it carries out a register load from #80 BYTES etc.

33 ACTIVATE

The ACT position of this switch causes an entry to the microprogram such that if a normal microprogram is in, a register load from #80 BYTES etc. is carried out when RUN is set.

34 { BOOTS

Operation of this key results in a PAW function of IPL being transmitted to the controller selected by the Keys of Row B. On receiving an interrupt from the controller the OCP loads its microprogram etc.

35 RUN

The RUN position allows the clock to run until it is stopped by a stop condition or RUN is switched off. The conditions for stopping are various and are as follows.

1. The stop conditions described under ROW B
2. System error with stop on System Error or system Error mask set.
3. Inhibit time out and the OCP is waiting for something.
4. AGU and ARU only with the interlock functioning.

35 OI

One instruction allows a single instruction for each operation of WALK/GO key.

35 SS

Single shot allows a single microprogram instruction for each operation of WALK/GO key.

36 { WALK
GO

A 16 μ sec clock used in conjunction with OI and SS.

A Single Shot used in conjunction with OI and SS.

ROW 1

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

ROW 2

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

ROW 3

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

ROW 4

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

ROW A

FUNCTION		PRIMARY		PRIMARY		SECONDARY		TERTIARY		E (2)		E (3)		E (4)		E (5)		E (6)		E (7)		E (8)		E (9)		E (10)		E (11)		E (12)		E (13)		E (14)		E (15)		E (16)		E (17)		E (18)		E (19)		E (20)		E (21)		E (22)		E (23)		E (24)		E (25)		E (26)		E (27)		E (28)		E (29)		E (30)		E (31)		E (32)		E (33)		E (34)		E (35)		E (36)		E (37)		E (38)		E (39)		E (40)		E (41)		E (42)		E (43)		E (44)		E (45)		E (46)		E (47)		E (48)		E (49)		E (50)		E (51)		E (52)		E (53)		E (54)		E (55)		E (56)		E (57)		E (58)		E (59)		E (60)		E (61)		E (62)		E (63)		E (64)		E (65)		E (66)		E (67)		E (68)		E (69)		E (70)		E (71)		E (72)		E (73)		E (74)		E (75)		E (76)		E (77)		E (78)		E (79)		E (80)		E (81)		E (82)		E (83)		E (84)		E (85)		E (86)		E (87)		E (88)		E (89)		E (90)		E (91)		E (92)		E (93)		E (94)		E (95)		E (96)		E (97)		E (98)		E (99)		E (100)		E (101)		E (102)		E (103)		E (104)		E (105)		E (106)		E (107)		E (108)		E (109)		E (110)		E (111)		E (112)		E (113)		E (114)		E (115)		E (116)		E (117)		E (118)		E (119)		E (120)		E (121)		E (122)		E (123)		E (124)		E (125)		E (126)		E (127)		E (128)		E (129)		E (130)		E (131)		E (132)		E (133)		E (134)		E (135)		E (136)		E (137)		E (138)		E (139)		E (140)		E (141)		E (142)		E (143)		E (144)		E (145)		E (146)		E (147)		E (148)		E (149)		E (150)		E (151)		E (152)		E (153)		E (154)		E (155)		E (156)		E (157)		E (158)		E (159)		E (160)		E (161)		E (162)		E (163)		E (164)		E (165)		E (166)		E (167)		E (168)		E (169)		E (170)		E (171)		E (172)		E (173)		E (174)		E (175)		E (176)		E (177)		E (178)		E (179)		E (180)		E (181)		E (182)		E (183)		E (184)		E (185)		E (186)		E (187)		E (188)		E (189)		E (190)		E (191)		E (192)		E (193)		E (194)		E (195)		E (196)		E (197)		E (198)		E (199)		E (200)		E (201)		E (202)		E (203)		E (204)		E (205)		E (206)		E (207)		E (208)		E (209)		E (210)		E (211)		E (212)		E (213)		E (214)		E (215)		E (216)		E (217)		E (218)		E (219)		E (220)		E (221)		E (222)		E (223)		E (224)		E (225)		E (226)		E (227)		E (228)		E (229)		E (230)		E (231)		E (232)		E (233)		E (234)		E (235)		E (236)		E (237)		E (238)		E (239)		E (240)		E (241)		E (242)		E (243)		E (244)		E (245)		E (246)		E (247)		E (248)		E (249)		E (250)		E (251)		E (252)		E (253)		E (254)		E (255)		E (256)		E (257)		E (258)		E (259)		E (260)		E (261)		E (262)		E (263)		E (264)		E (265)		E (266)		E (267)		E (268)		E (269)		E (270)		E (271)		E (272)		E (273)		E (274)		E (275)		E (276)		E (277)		E (278)		E (279)		E (280)		E (281)		E (282)		E (283)		E (284)		E (285)		E (286)		E (287)		E (288)		E (289)		E (290)		E (291)		E (292)		E (293)		E (294)		E (295)		E (296)		E (297)		E (298)		E (299)		E (300)		E (301)		E (302)		E (303)		E (304)		E (305)		E (306)		E (307)		E (308)		E (309)		E (310)		E (311)		E (312)		E (313)		E (314)		E (315)		E (316)		E (317)		E (318)		E (319)		E (320)		E (321)		E (322)		E (323)		E (324)		E (325)		E (326)		E (327)		E (328)		E (329)		E (330)		E (331)		E (332)		E (333)		E (334)		E (335)		E (336)		E (337)		E (338)		E (339)		E (340)		E (341)		E (342)		E (343)		E (344)		E (345)		E (346)		E (347)		E (348)		E (349)		E (350)		E (351)		E (352)		E (353)		E (354)		E (355)		E (356)		E (357)		E (358)		E (359)		E (360)		E (361)		E (362)		E (363)		E (364)		E (365)		E (366)		E (367)		E (368)		E (369)		E (370)		E (371)		E (372)		E (373)		E (374)		E (375)		E (376)		E (377)		E (378)		E (379)		E (380)		E (381)		E (382)		E (383)		E (384)		E (385)		E (386)		E (387)		E (388)		E (389)		E (390)		E (391)		E (392)		E (393)		E (394)		E (395)		E (396)		E (397)		E (398)		E (399)		E (400)		E (401)		E (402)		E (403)		E (404)		E (405)		E (406)		E (407)		E (408)		E (409)		E (410)		E (411)		E (412)		E (413)		E (414)		E (415)		E (416)		E (417)		E (418)		E (419)		E (420)		E (421)		E (422)		E (423)		E (424)		E (425)		E (426)		E (427)		E (428)		E (429)		E (430)		E (431)		E (432)		E (433)		E (434)		E (435)		E (436)		E (437)		E (438)		E (439)		E (440)		E (441)		E (442)		E (443)		E (444)		E (445)		E (446)		E (447)		E (448)		E (449)		E (450)		E (451)		E (452)		E (453)		E (454)		E (455)		E (456)		E (457)		E (458)		E (459)		E (460)		E (461)		E (462)		E (463)		E (464)		E (465)		E (466)		E (467)		E (468)		E (469)		E (470)		E (471)		E (472)		E (473)		E (474)		E (475)		E (476)		E (477)		E (478)		E (479)		E (480)		E (481)		E (482)		E (483)		E (484)		E (485)		E (486)		E (487)		E (488)		E (489)		E (490)		E (491)		E (492)		E (493)		E (494)		E (495)		E (496)		E (497)		E (498)		E (499)		E (500)		E (501)		E (502)		E (503)		E (504)		E (505)		E (506)		E (507)		E (508)		E (509)		E (510)		E (511)		E (512)		E (513)		E (514)		E (515)		E (516)		E (517)		E (518)		E (519)		E (520)		E (521)		E (522)		E (523)		E (524)		E (525)		E (526)		E (527)		E (528)		E (529)		E (530)		E (531)		E (532)		E (533)		E (534)		E (535)		E (536)		E (537)		E (538)		E (539)		E (540)		E (541)		E (542)		E (543)		E (544)		E (545)		E (546)		E (547)		E (548)		E (549)		E (550)		E (551)		E (552)		E (553)		E (554)		E (555)		E (556)		E (557)		E (558)		E (559)		E (560)		E (561)		E (562)		E (563)		E (564)		E (565)		E (566)		E (567)		E (568)		E (569)		E (570)		E (571)		E (572)		E (573)		E (574)		E (575)		E (576)		E (577)		E (578)		E (579)		E (580)		E (581)		E (582)		E (583)		E (584)		E (585)		E (586)		E (587)		E (588)		E (589)		E (590)		E (591)		E (592)		E (593)		E (594)		E (595)		E (596)		E (597)		E (598)		E (599)		E (600)		E (601)		E (602)		E (603)		E (604)		E (605)		E (606)		E (607)		E (608)		E (609)		E (610)		E (611)		E (612)		E (613)		E (614)		E (615)		E (616)		E (617)		E (618)		E (619)		E (620)		E (621)		E (622)		E (623)		E (624)		E (625)		E (626)		E (627)		E (628)		E (629)		E (630)		E (631)		E (632)		E (633)		E (634)		E (635)		E (636)		E (637)		E (638)		E (639)		E (640)		E (641)		E (642)		E (643)		E (644)		E (645)		E (646)		E (647)		E (648)		E (649)		E (650)		E (651)		E (652)		E (653)		E (654)		E (655)		E (656)		E (657)		E (658)		E (659)		E (660)		E (661)		E (662)		E (663)		E (664)		E (665)		E (666)		E (667)		E (668)		E (669)		E (670)		E (671)		E (672)		E (673)		E (674)		E (675)		E (676)		E (677)		E (678)		E (679)		E (680)		E (681)		E (682)		E (683)		E (684)		E (685)		E (686)		E (687)		E (688)		E (689)		E (690)		E (691)		E (692)		E (693)		E (694)		E (695)		E (696)		E (697)		E (698)		E (699)		E (700)		E (701)		E (702)		E (703)		E (704)		E (705)		E (706)		E (707)		E (708)		E (709)		E (710)		E (711)		E (712)		E (713)		E (714)		E (715)		E (716)		E (717)		E (718)		E (719)		E (720)		E (721)		E (722)		E (723)		E (724)		E (725)		E (726)		E (727)		E (728)		E (729)		E (730)		E (731)		E (732)		E (733)		E (734)		E (735)		E (736)		E (737)		E (738)		E (739)		E (740)		E (741)		E (742)		E (743)		E (744)		E (745)		E (746)		E (747)		E (748)		E (749)		E (750)		E (751)		E (752)		E (753)		E (754)		E (755)		E (756)		E (757)		E (758)		E (759)		E (760)		E (761)		E (762)		E (763)		E (764)		E (765)		E (766)		E (767)		E (768)		E (769)		E (770)		E (771)		E (772)		E (773)		E (774)		E (775)		E (776)		E (777)		E (778)		E (779)		E (780)		E (781)		E (782)		E (783)		E (784)		E (785)		E (786)		E (787)		E (788)		E (789)		E (790)		E (791)		E (792)		E (793)		E (794)		E (795)		E (796)		E (797)		E (798)		E (799)		E (800)		E (801)		E (802)		E (803)		E (804)		E (805)		E (806)		E (807)		E (808)		E (809)		E (810)		E (811)		E (812)		E (813)		E (814)		E (815)		E (816)		E (817)		E (818)		E (819)		E (820)		E (821)		E (822)		E (823)		E (824)		E (825)		E (826)		E (827)		E (828)		E (829)		E (830)		E (831)		E (832)		E (833)		E (834)		E (835)		E (836)		E (837)		E (838)		E (839)		E (840)		E (841)		E (842)		E (843)		E (844)		E (845)		E (846)		E (847)		E (848)		E (849)		E (850)		E (851)		E (852)		E (853)		E (854)		E (855)		E (856)		E (857)		E (858)		E (859)		E (860)		E (861)		E (862)		E (863)		E (864)		E (865)		E (866)		E (867)		E (868)		E (869)		E (870)		E (871)		E (872)		E (873)		E (874)		E (875)		E (876)		E (877)		E (878)		E (879)		E (880)		E (881)		E (882)		E (883)		E (884)		E (885)		E (886)		E (887)		E (888)		E (889)		E (890)		E (891)		E (892)		E (893)		E (894)		E (895)		E (896)		E (897)		E (898)		E (899)		E (900)		E (901)		E (902)		E (903)		E (904)		E (905)		E (906)		E (907)		E (908)		E (909)		E (910)		E (911)		E (912)		E (913)		E (914)		E (915)		E (916)		E (917)		E (918)		E (919)		E (920)		E (921)		E (922)		E (923)		E (924)		E (925)		E (926)		E (927)		E (928)		E (929)		E (930)		E (931)		E (932)		E (933)		E (934)		E (935)		E (936)		E (937)		E (938)		E (939)		E (940)		E (941)		E (942)		E (943)		E (944)		E (945)		E (946)		E (947)		E (948)		E (949)		E (950)		E (951)		E (952)		E (953)		E (954)		E (955)		E (956)		E (957)			
----------	--	---------	--	---------	--	-----------	--	----------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	--	--

ROW 5	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487	1488	1489	1
-------	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	---

ACU REGISTER COM																ARU MPS CURRENT ADDRESS										ARU MPS ACCESS ADDRESS										ARU VI ADDRESS																																										
01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	00	01	02	03	04	05	06	07	08	09	10	11	00	01	02	03	04	05	06	07	08	09	10	11	00	01	02	03	04	05	06	07	08	09	10	11	00	01	02	03	04	05	06	07	08	09	10	11
																PARITY																																																														
																										7 MS										DATA																																										

ROW 14

ACU REGISTER C.A.										ACU REGISTER V.A.										BYTE FUNCTION																																							
28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87
																				PARITY								SMAC No								REAL ADDRESS																							

POINT	PORT	PORT	PORT	AGN	ACCESS	APR	ACA	RESPONSE	LOAD	WD	CPR
SEINT	SEINT	SEINT	SEINT								
EXINT	EXINT	EXINT	EXINT								
OPINT	OPINT	OPINT	OPINT								
SEINT	SEINT	SEINT	SEINT								
EXINT	EXINT	EXINT	EXINT								
OPINT	OPINT	OPINT	OPINT								
SEINT	SEINT	SEINT	SEINT								
EXINT	EXINT	EXINT	EXINT								
OPINT	OPINT	OPINT	OPINT								
SEINT	SEINT	SEINT	SEINT								
EXINT	EXINT	EXINT	EXINT								
OPINT	OPINT	OPINT	OPINT								
SEINT	SEINT	SEINT	SEINT								
EXINT	EXINT	EXINT	EXINT								
OPINT	OPINT	OPINT	OPINT								
SEINT	SEINT	SEINT	SEINT								
EXINT	EXINT	EXINT	EXINT								
OPINT	OPINT	OPINT	OPINT								
SEINT	SEINT	SEINT	SEINT								
EXINT	EXINT	EXINT	EXINT								
OPINT	OPINT	OPINT	OPINT								
SEINT	SEINT	SEINT	SEINT								
EXINT	EXINT	EXINT	EXINT								
OPINT	OPINT	OPINT	OPINT								
SEINT	SEINT	SEINT	SEINT								
EXINT	EXINT	EXINT	EXINT								
OPINT	OPINT	OPINT	OPINT								
SEINT	SEINT	SEINT	SEINT								
EXINT	EXINT	EXINT	EXINT								
OPINT	OPINT	OPINT	OPINT								
SEINT	SEINT	SEINT	SEINT								
EXINT	EXINT	EXINT	EXINT								
OPINT	OPINT	OPINT	OPINT								
SEINT	SEINT	SEINT	SEINT								
EXINT	EXINT	EXINT	EXINT								
OPINT	OPINT	OPINT	OPINT								
SEINT	SEINT	SEINT	SEINT								
EXINT	EXINT	EXINT	EXINT								
OPINT	OPINT	OPINT	OPINT								
SEINT	SEINT	SEINT	SEINT								
EXINT	EXINT	EXINT	EXINT								
OPINT	OPINT	OPINT	OPINT								
SEINT	SEINT	SEINT	SEINT								
EXINT	EXINT	EXINT	EXINT								
OPINT	OPINT	OPINT	OPINT								
SEINT	SEINT	SEINT	SEINT								
EXINT	EXINT	EXINT	EXINT								
OPINT	OPINT	OPINT	OPINT								
SEINT	SEINT	SEINT	SEINT								
EXINT	EXINT	EXINT	EXINT								
OPINT	OPINT	OPINT	OPINT								

[illegible]

ACU	ALCISTERA	WD	ROW
00	01	02	03
04	05	06	07
08	09	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63
64	65	66	67
68	69	70	71
72	73	74	75
76	77	78	79
80	81	82	83
84	85	86	87
88	89	90	91
92	93	94	95
96	97	98	99
00	01	02	03
04	05	06	07
08	09	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63
64	65	66	67
68	69	70	71
72	73	74	75
76	77	78	79
80	81	82	83
84	85	86	87
88	89	90	91
92	93	94	95
96	97	98	99

	AMU COPY BUFFER	AGU COPY BUFFER	1900 EM LINK	AGU SA	AGU LA	CW-7
00 01 02 03 04 05 06 07 08 09 10 11	00 01 02 03 04 05 06 07 08 09 10 11	00 01 02 03 04 05 06 07 08 09 10 11	00 01 02 03 04 05 06 07 08 09 10 11	00 01 02 03 04 05 06 07 08 09 10 11	00 01 02 03 04 05 06 07 08 09 10 11	INCHI ISTO IRTY ISER ITSL IINS ICPS IPNO

ROW 19

OPINT	OPRAND	CMND	SRINT
0	1		
	2		
	3		
	4		
	5		
	6		
	7		
	8		
	9		
	10		
	11		
	12		
	13		
	14		
	15		
	16		
	17		
	18		
	19		
	20		
	21		
	22		
	23		
	24		
	25		
	26		
	27		
	28		
	29		
	30		
	31		
	32		
	33		
	34		
	35		
	36		
	37		
	38		
	39		
	40		
	41		
	42		
	43		
	44		
	45		
	46		
	47		
	48		
	49		
	50		
	51		
	52		
	53		
	54		
	55		
	56		
	57		
	58		
	59		
	60		
	61		
	62		
	63		
	64		
	65		
	66		
	67		
	68		
	69		
	70		
	71		
	72		
	73		
	74		
	75		
	76		
	77		
	78		
	79		
	80		
	81		
	82		
	83		
	84		
	85		
	86		
	87		
	88		
	89		
	90		
	91		
	92		
	93		
	94		
	95		
	96		
	97		
	98		
	99		
	100		

EXTERNAL MONITOR STRIP

EMAS 2900 HARDWARE NOTE

No: 3
Date: 13/12/79

How to Force a Program Error Interrupt from the Handkeys

Before a dump can be usefully taken the machine must be at IDLE "DEAD". This state is reached after most hardware and software errors. However, if the machine is not at IDLE "DEAD" an error should be forced, as follows:

For the 2960 (SM11, using TLOE/8):

On the ECP, type:

DW 9FFC
WMA 0 E
GO

For the 2970:

1. Off "RUN"
2. "ENG" down
3. Handkeys (top row) to ZERO
4. One tick on "GO" (usually, but more ticks may be necessary) until the RLOTB light comes on
5. "ORDER" down
6. Press "GO" until no lights on in RGF (2nd row of lights from top)
7. "ORDER" central
8. "RUN" - machine goes to IDLE "DEAD"

For the 2980:

1. Set switches as follows:

SMGO	UP	(Row 1 Key 1)
S.INST.MODE followed by S.SHOT MODE	DOWN/UP	(Row 6 Key 2)
I1 STOP	DOWN	(Row 3 Key 7)
STOP ON FAIL	UP	(Row 5 Key 4)
INH.ORD.STOP	DOWN	(Row 5 Key 9)

These are the only switches that should be set, apart from the deck TRUNK/STREAM switches. Unset any others.

2. Turn to Frame 01 (Critic and Modifier Dataflow) on monitor display.
3. Set Serial Monitoring address keys to Frame 1, Line 3.
4. Set bit 17 on DATA-KOD keys.

5. Press SET REG (Row 1, Key 4 - UP). Check that lamp FSEI on Line 3 (bit 17) comes on.
6. Turn to Frame 10 (Modification and B Arithmetic Stage Microprogram) on monitor display.
7. Set RUN (Row 6, Key 2 - CENTRAL).
8. Press GO (Row 6, Key 1 - DOWN). Check that machine stops with I1 (Line 7, bit 4) lamp lit.
9. Turn to Frame 21 (SAC trunk links E, F, G, H) on monitor display.
10. Rest I1 STOP (Row 3, Key 7 - CENTRAL).
11. Press GO (Row 6, Key 1 - DOWN) - machine goes to IDLE "DEAD".

R.D. Eager
J.F. Livingstone
P.D. Stephens
J.K. Yarwood

EMAS 2900 HARDWARE NOTE

No: 4
Date: 19/12/77

Hardware Testing

A process, ENGINR, exists to be used for running hardware tests.
Currently two tests exist:

- a) A test of the VMY instruction accessing the B register and data which goes across a page boundary. This used to give an erroneous bound check on the 2970 when a page fault occurred during its execution; it has now been corrected. The test is preserved for use on other 2900 machines. To run it, type:

COMMAND: RUN VMYBY

DO YOU WANT SECOND PAGE IN CORE?
Y/N: N or Y

VMY B WORKED OK.

Note that the program should work whether or not the second page is in core. If the fault exists the program will work when the reply is 'Y', but will fail with a band check if the reply is 'N'.

b) RAND READ

This is a program based on RAND READ on System 4 EMAS. It can be used to test up to 4 disk drives. The disks are referred to by their FSYS numbers - e.g. PETE01 is disk 1, EMAS10 is disk 10.

The program is run thus:

COMMAND: RANDREAD

NO OF DISKS: reply 1-4

FSYS NOS. OF DISKS: reply with FSYS numbers

Q SIZE PER DISK: reply 1-16, specifying number of transfers to
be PONned to each disk at once

READS/DISK: reply with number of READs to be done

When the specified number of reads have been completed, the number of faults, cumulative since the start of the command, are reported and the user is then invited to continue. The reply should be Y to continue or N to stop the tests. If the user replies with Y, the READS/DISK: prompt is given again; the other parameters are held at their previous values.

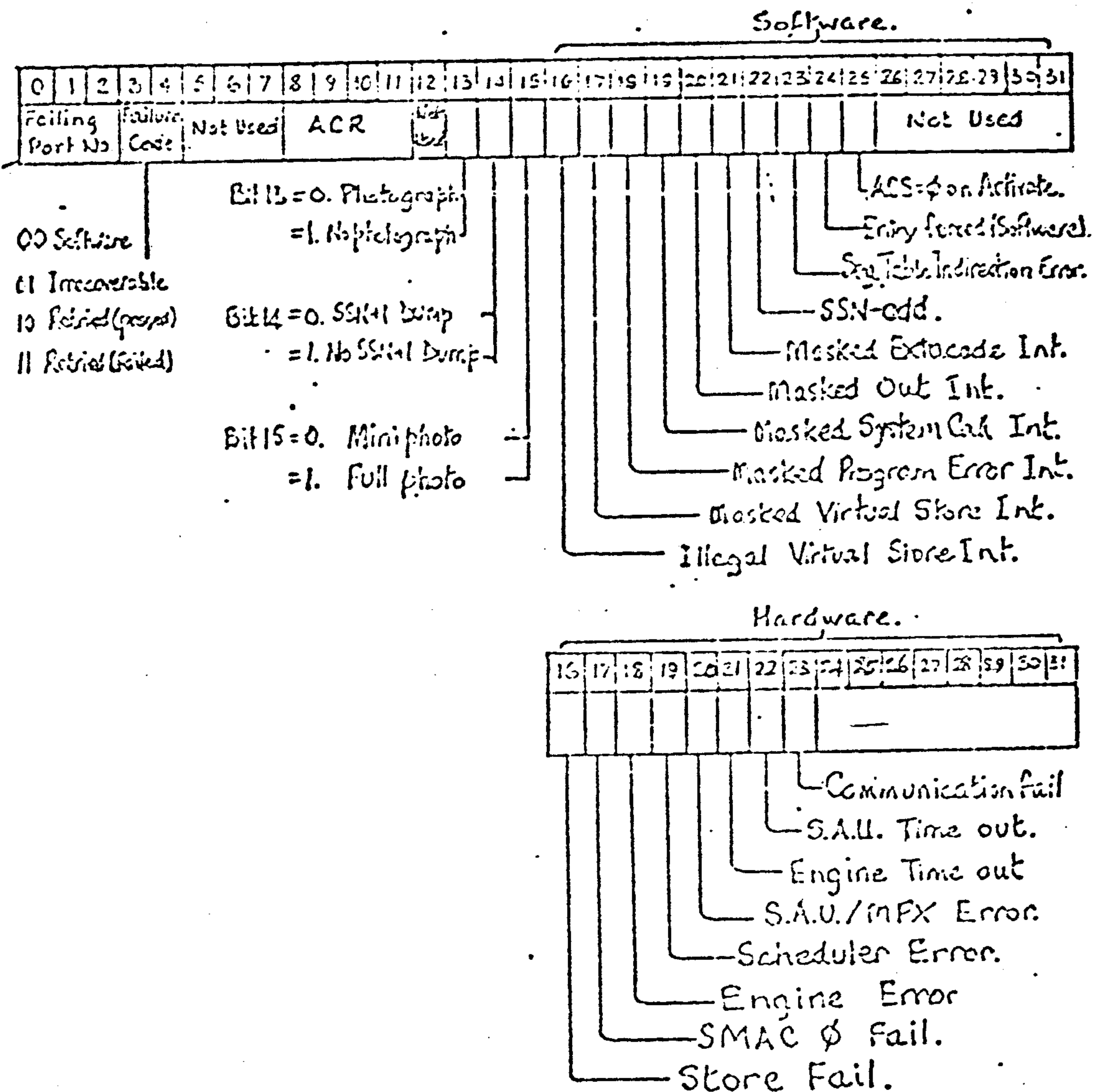
Roderick McLeod

System Error Interrupt Parameters for
2960, 2970, 2980

This note is taken from a badly-Xeroxed copy of ICL document
SYSB-22-0233.

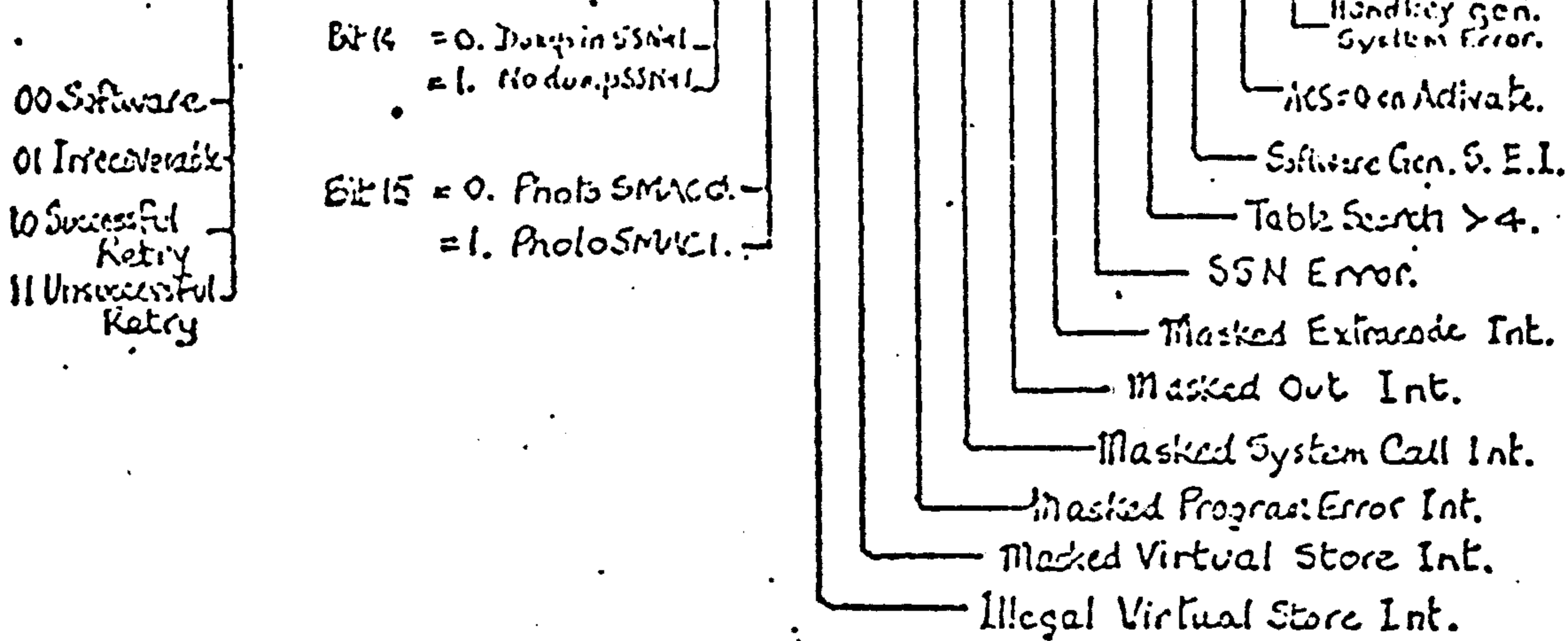
The System Error Interrupt Stack

The system error is a stack switching interrupt and the code of SEI
Analyser is entered with its LNB at the stack base. Word 0 contains the
stack segment number (in the current implementation this is public segment
8198). Words 1 and 2 are zero and words 3 and 4 point to the Procedure
Linkage Table (PLT) of the compilation module containing SEI Analyser.
Word 5 contains the SEI parameter as stacked by the hardware interrupt
routine and word 6 is the segment number of the interrupted stack.

SEI Parameter for 2960

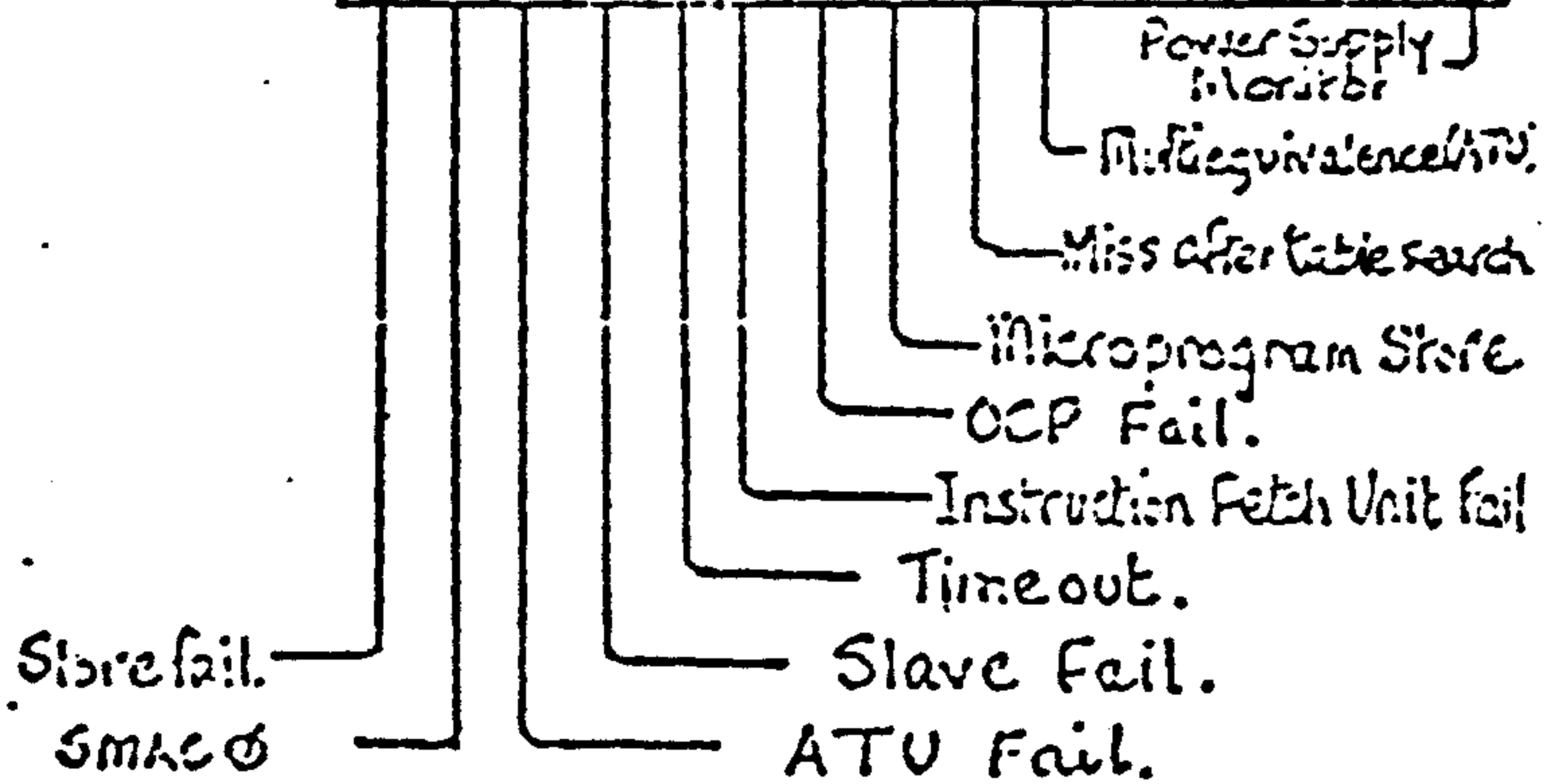
The parameter for interrupts other than on the current OCP port will look
like an empty software failure parameter (i.e. bits 16-31 not used).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Failing Part No.	Failure Code	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used



Hardware.

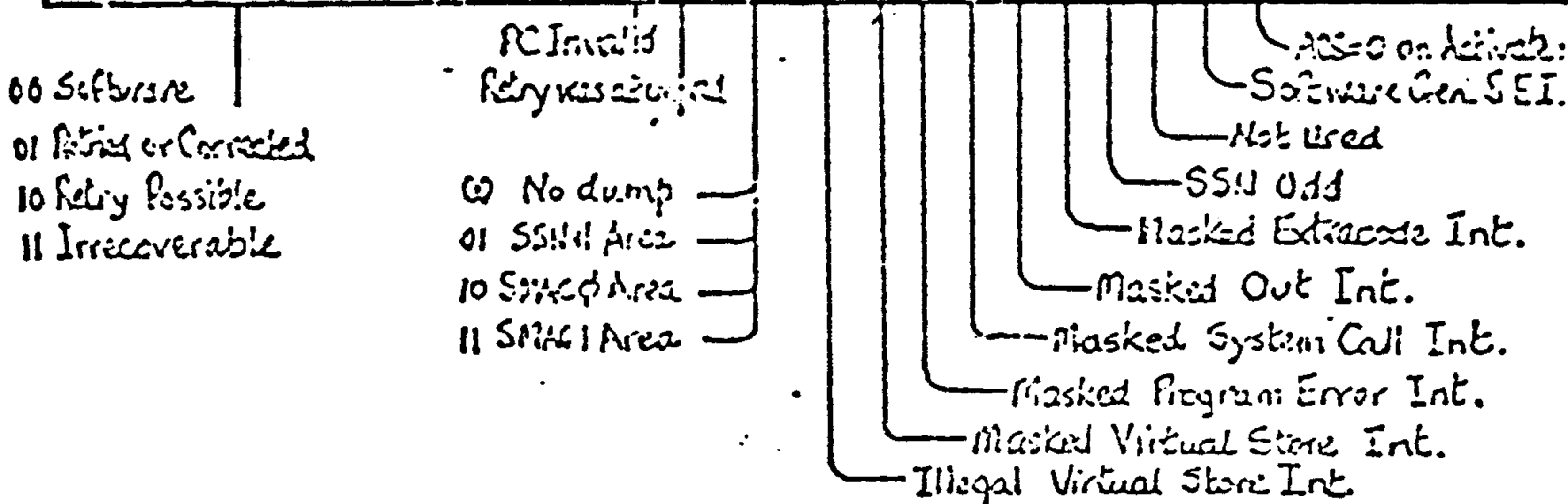
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31



SEI Parameter for 2980

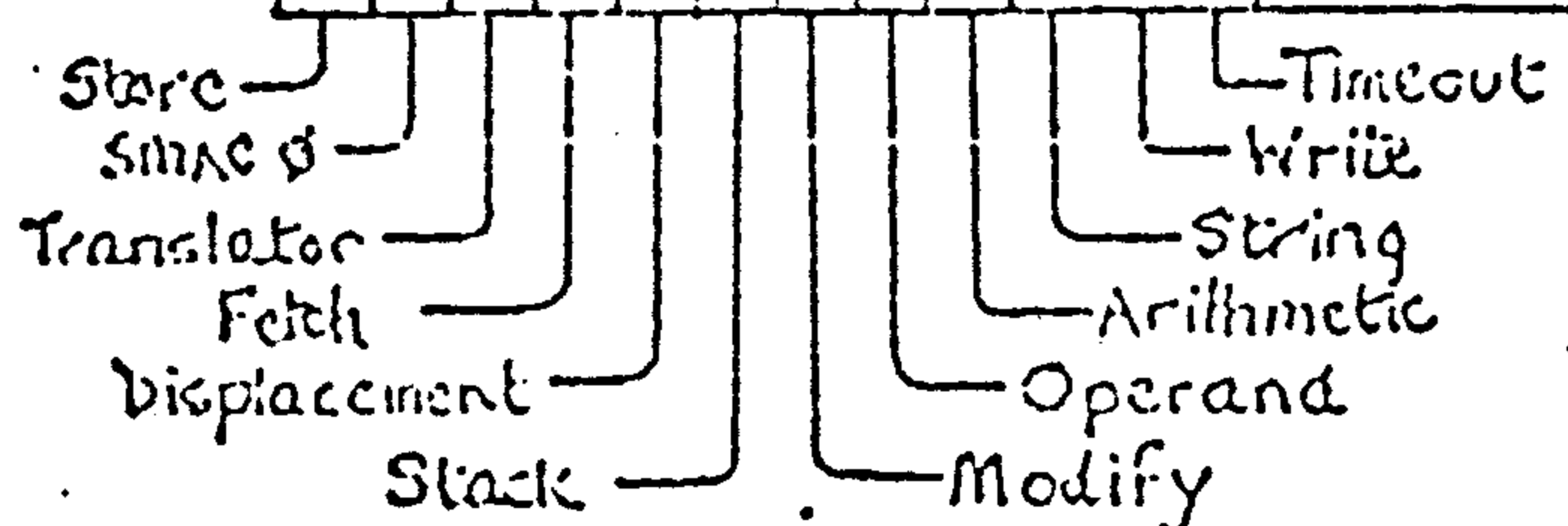
Software.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Failing Part No.			Failure Code		Not Used			ACR						Dmg Area																Not Used	



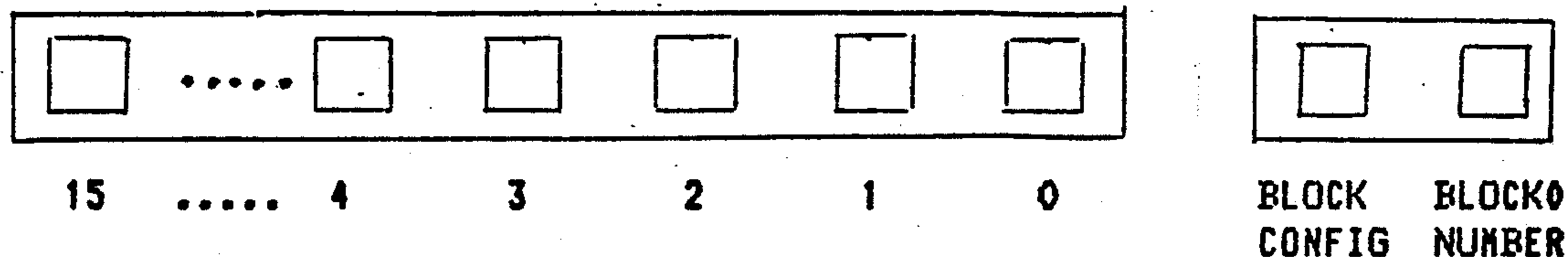
Hardware.

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31



2970: How to Reconfigure the Store on a SHAC

Use the bottom row of (mostly) blue keys in the store cabinet:



The BLOCK0 NUMBER key is grey, all other keys on this row are blue. Each 128 Kb block required to be on-line should have its corresponding key (0-15) pressed down. Effect the reconfiguration by pressing (and releasing) the adjacent BLOCK CONFIG key. The BLOCK0 NUMBER refers to the lowest numbered on-line block (generally real block 0). If the reconfiguration changes the lowest numbered block then the new BLOCK0 number should be set on the four grey keys on the row above the block configuration row and the BLOCK0 NUMBER key pressed and released. The fifth row of lights on the Store cabinet displays the current block configuration and BLOCK0 number.

N.B. The Store need not be contiguous except that the EMAS Supervisor loader requires blocks 0 and 1 of SHAC 0 to be configured on-line.

John Maddock

2950: Notes on Transfer Control Blocks (TCBs)
for People Who Are Used to the GPC

This Note is intended as a commentary on pages 30-40 of the ICL Product Specification document PSD 2.5.3, an extract of which is available on request. The diagram is taken from the document.

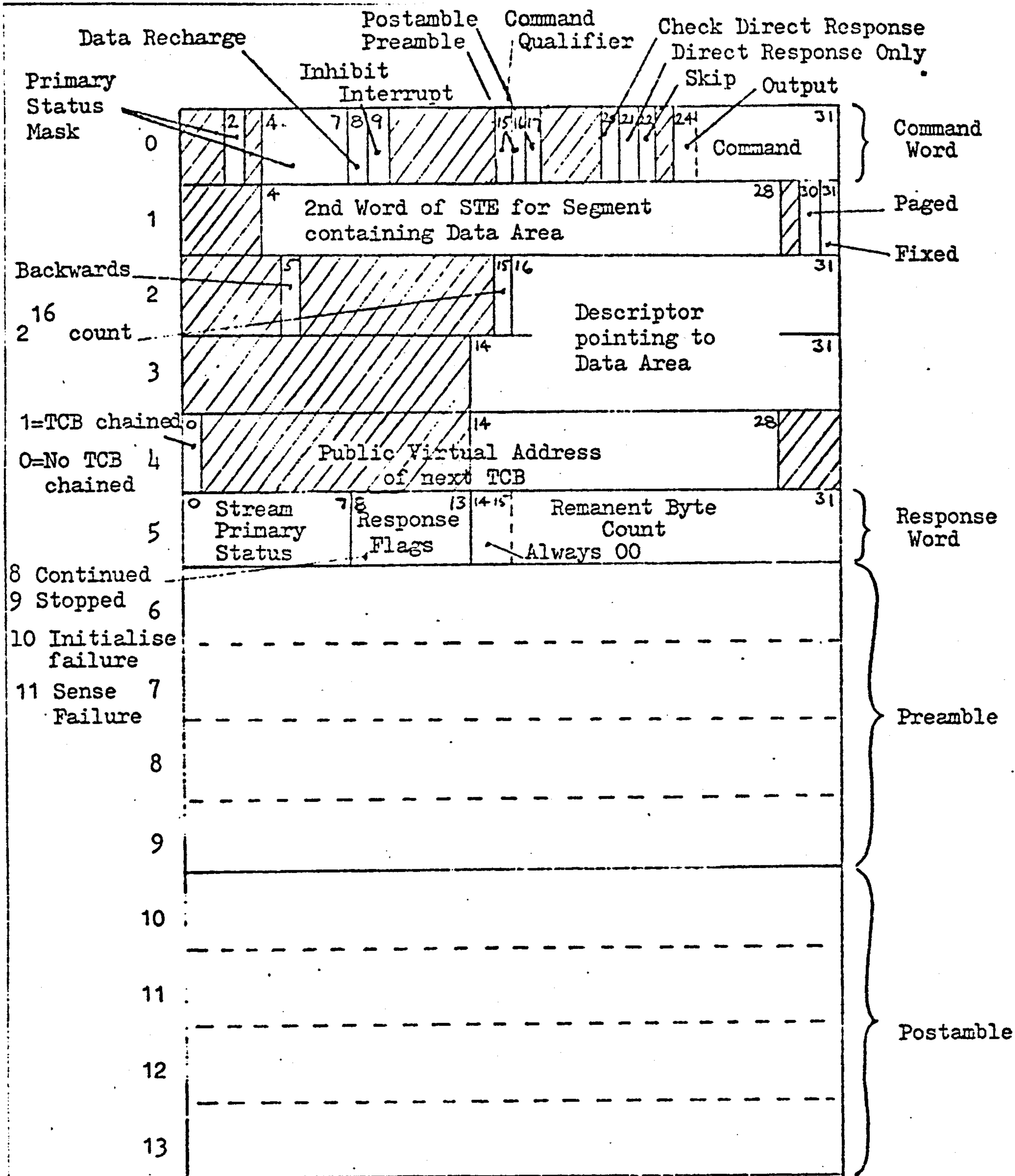
TCB Word 0

bits 0,1	not used
bit 2	if set, suppresses termination on "attention"
bit 3	not used
bits 4-7	suppress long block, short block, condition X, condition Y = bits 8 to 11 of logic block entry
bit 8	data recharge from next TCB: equivalent to bit 4 of LBE
bit 9	inhibit interrupt: converse of bit 3 (PCI) in LBE (could also serve for bit 2 of SAW flags in STREAM SAW in H/W comm. area)
bits 10-14	not used
bit 15	command qualifier: 0 for "device", 1 for "stream"; same as bit 15 in LBE (?)
bit 16	preamble valid: controls initialisation, like RCB word 0 bit 16
bit 17	postamble valid: controls collection of status after unsuccessful transfer
bits 18,19	not used
bits 20,21	only relevant with 1900 SI peripherals
bit 22	for read commands: don't write data to store (skip). ? <u>Not</u> same as LBE bit 2.
bit 23	not used
bit 24	output (this seems to be nonsense, except for stream commands)
bits 24-31	command (in actual fact bit 31 seems to signal 'output'). Whichever bit <u>actually</u> indicates 'output' should correspond to bit 0 of LBE

TCB Word 1

bits 0-3	not used
bits 4-28	copied from 2nd word of STE of data segment - real address of segment or page table
bit 29	not used
bit 30	paged (same as bit 1 in 1st word of STE)
bit 31	fixed - copied from 2nd word of STE; must be 1, unless you have a command to transfer no data - then it must be 0

This substitutes for the sort of information which is found in word 1 of RCB.



Notes

- (i) Shaded areas are ignored by IO System but those in word 0 are reserved (for hardware use).
- (ii) Word 0 must be on 2 word boundary in a Public Segment with the TCB in a contiguous area not crossing a page boundary.

The Transfer Control Block.

TCB Words 2 and 3

Data descriptor. Must be a byte or string descriptor. Only the bound and address-within-segment are significant, and only 16 bits of the bound are used - i.e., byte count in word 2 bits 16-31, and address in word 3 bits 14 to 31. A zero byte count is taken to mean 2^{**16} . S/w should have word 2 bit 15 =1 in that case, and =0 in all others. Word 2 bit 5 must be set for "backwards" transfers (? corresponds to bit 1 of LBE).

These two words correspond to the address list entry.

TCB Word 4

Public virtual address of next TCB - only bits 0 and 14-28 are significant. Bit 0 must be 1 (public address). If it is zero (local address) the chain is terminated. Therefore this corresponds to bit 4 V bit 5 in LBE (data chain V command chain). Only address-within-segment (bits 14-28) is significant because new TCB must be double word aligned and in same segment as previous TCB.

TCB Word 5

bits 0-7 stream primary status
 = bits 16-23 of word 0 of STREAM RESPONSE in Controller H/W Comm. Area

bits 8-13 response flags
 bit 8 stopped
 bit 9 continued
 bit 10 initialise failure
 bit 11 sense failure

bits 16-31 remanent byte count
 = bits 14-31 of word 1 of STREAM RESPONSE

TCB Words 6-9

Preamble area

word 6 initialise data (among other things)
 = word 6 of RCB

TCB Words 10-13

Postamble area. Gets sense bytes - secondary and tertiary status bytes.

J. Wexler

IMP Call Sequences

The pre-call sequence is first to odd-align the stack front and then to store the current LNB at SF (this SF to be the new LNB, i.e. in the called routine). The stack front is further raised by 4 words for the link descriptor (placed in old LNB+1,2 by the CALL) and for the called routine's PLT descriptor, to be placed in LNB+3,4 by the post-call sequence. The purpose of odd-aligning the stack front is so that the link and PLT descriptors are double word aligned. This is all done by the single instruction

PRCL 4

Next the parameters are stacked (if required) and LNB is raised by 5 plus the number of parameter words to point to the new display (starting at the stack front as it was after the odd-aligning but before the storing of LNB).

Programmers should note the effect of the pre-call alignment when designing routine headings. For example

routine ONE(integer A, integername B)

results in the descriptor to B being across a double-word boundary and requires 2 core cycles to fetch it.

routine TWO(integername B, integer A)

is more efficient.

XNB is set to the called routine's textually surrounding routine for internal calls.

Finally the CALL is executed, with operand either

- (i) for an internal routine: N where N is offset in halfwords from current PC
- (ii) for an external routine: @(XNB+N) where XNB+N has a descriptor which points to the code descriptor in the called routine's PLT.

The PLT is conceptually an array of code descriptors for the routines in the module. An external reference is a descriptor descriptor to one of these (i.e. a descriptor which points to one of these). The loader fills in the addresses appropriately.

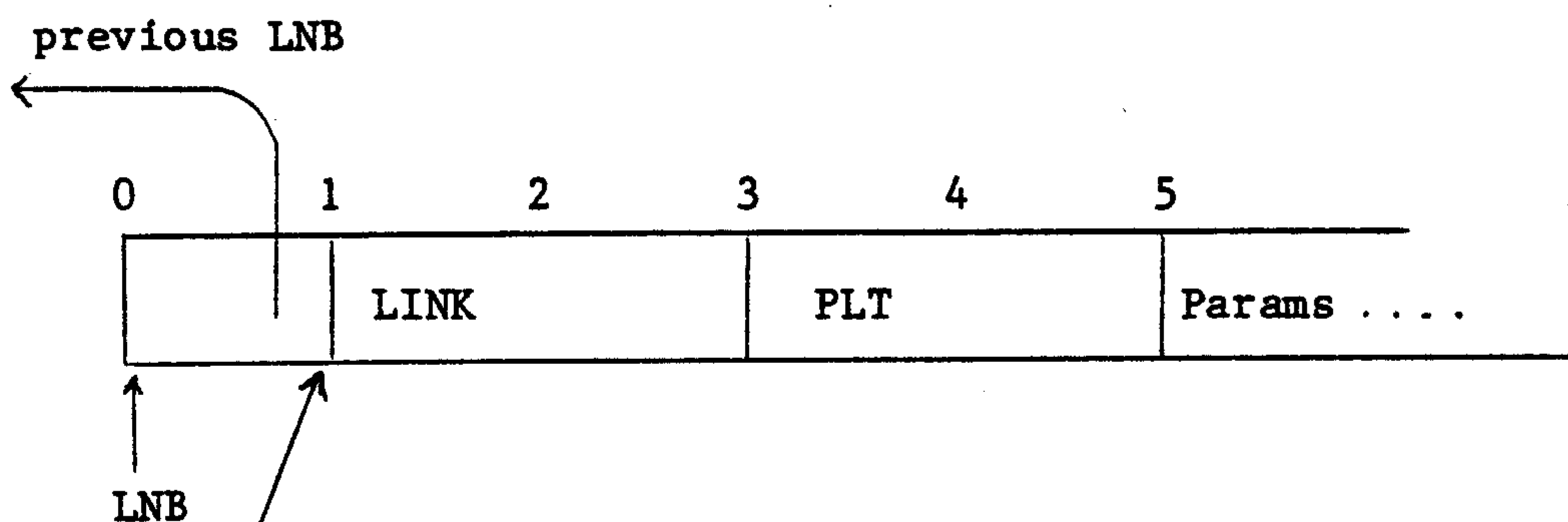
To summarise, the CALL instruction loads DR with the operand descriptor (if the operand is indirect). It also places a link descriptor at old LNB+1,2. LNB has been set up by the pre-call sequence to point to the new display.

For external routines, the post-call sequence (in the called routine) is to store DR, which has a descriptor to the descriptor in the called routine's PLT, at LNB+3,4, having used INCA to set the address in DR back to the start of PLT. That is, LNB+3,4 have a descriptor to this routine's PLT.

So far,

INCA	-N	where N is the offset of this routine's descriptor from the start of PLT (omitted if N=0).
LDB	0	set bound to zero now, but later set to offset from CST of ON and diagnostic block patched (ORed) in at end of routine.
STD	(LNB+3)	and 4.

LNB+4 has the address therefore of PLT (GLA), and this is loaded to XNB subsequently as required (e.g. for referencing own variables and for making further copies).



The last bit of the stored LNB is used to indicate that SF was increased by 1 in the pre-call sequence - it should be ignored when chaining back in dumps, etc.

P.D. Stephens

%NAME Parameters in IMP

The general parameter %NAME allows any entity to be passed to the called routine. This facility is used in the sequential and direct access routines but is not a user facility. It is possible to find out quite a lot about the actual parameter by examining the descriptor.

For variables as actual parameters the descriptor is a scaled vector descriptor with the address field pointing at the leftmost byte of the actual parameter. The size of the item can be obtained from the size code of the descriptor, as follows:

- 3 = byte
- 4 = half (care needed with this - not supported by hardware)
- 5 = integer or real (32-bit)
- 6 = long integer or long real (64-bit)
- 7 = long long real (128-bit)

The bound field gives the type as follows:

- 1 = integer
- 2 = real
- 3 = Boolean (ALGOL(E) only)

For structures (i.e. strings and records) an unscaled byte descriptor is passed with the bound field containing the size of the entity in bytes.

Having decoded the descriptor, the program can store items back into the variable using IMP mapping functions; i.e.

BYTEINTEGER(ADDR(name)) = ...

STRING(ADDR(name)) = ...

as appropriate.

P.D. Stephens

Extra Facilities in Release 8

Release 8 of IMP has some extra facilities on an experimental basis - they may be withdrawn at any time. To use the compiler, insert ERCC07.NRIMPO8Y in your directory and log out.

1) Profiling

If a file is compiled using PARM(PROFILE), extra code is inserted to maintain a count of how often each (executable) line is executed. If there are several statements on a line each will increment the count. A call on the special intrinsic routine PPROFILE will print out details of the most heavily used lines and reset all counts. Programs being run on other than the standard Subsystem will require to be linked with ERCC07.PPROF01Y to obtain special service routines for the profiling.

N.B. Profiling is very demanding of CPU time - expect programs to run at least 3 times slower.

2) Conditional compilation

If an IMP statement starts with ? it will be analysed but no code will be generated. However if PARM(PARMX) is specified then code for such a line will be generated normally. This provides a convenient method of including additional diagnostic code in software which can be quickly incorporated at moments of crisis.

3) Constant shuffling

Normally IMP pools all numeric and string constants by building a constant table which is added to the back of the code. This minimises program size. However when compiling files of routines where one may be used extensively and the others ignored it may be better to incorporate each routine's constants with its code. This will give a larger program but fewer page turns. If PARM(PARMY) is selected then the latter method of constant inclusion is used. It is likely that Subsystems, Directors and a few large libraries will benefit from this option.

Feedback on the effects (if any!) would be welcome.

P.D. Stephens

Interactive Debugging in IMP

DEBUG is an interactive debugging aid for IMP programs on EMAS 2900. It includes commands for examining and modifying data and hence changing the course of a computation, for setting breakpoints, and for the conditional execution of DEBUG commands. The breakpoint scheme makes it possible for the user to gain control during program execution and then specify any further action he wishes. The conditional execution facility allows DEBUG commands specified by the user to be stored, and executed only when certain conditions related to the running program are satisfied, e.g. when a given line number in the program has been reached.

In general, the intention has been to allow the user to intervene in the execution of the running program in a way consistent with the effect that would be produced by additional source statements in the program. Hence it is only possible to examine or change variables which are in scope when the DEBUG command is executed. However conditional commands may be issued at any time regardless of whether any variables they refer to are currently in scope.

Establish access to DEBUG by first typing:

Command:OPTION(SEARCHDIR=CONLIB.GENERAL)

Then compile the program with PARM(DEBUG) set. Additional parms may be set with the exception of NOTRACE, NODIAG and OPT.

When the program is run, DEBUG will print the current line number and then prompt for a command:

DEBUG n: (where n is the index number of the command - see Kill)

The user can then input one or more DEBUG commands. Commands take two forms, simple and conditional. Simple commands, which consist of a letter followed, if appropriate, by a parameter, are executed immediately and then discarded. Conditional commands are stored, and only executed when the specified condition is satisfied. The program will only resume execution when the command R (see below) is given.

Conditional commands take the form:

<simple command> .IF <conditional expr>

or, for typing convenience,

<simple command> @ <conditional expr>

The following simple commands are available:

A <variable>=<value>	- Assign the value to the variable; it is not possible to assign a value to an unequivalenced name type variable.
----------------------	---

- B - Set a breakpoint; when executed, accept a DEBUG command from the console. B is only used as part of a conditional command.
- D <addr>,<length>,<format> - Dump an area of the virtual memory of <length> bytes, starting at <addr> to the console. <format> may be C for a character dump or H for a hexadecimal dump. The user must have determined the relevant virtual memory addresses prior to the execution of the program.
- F - Print a map of the object file, indicating the starting line numbers of the routines and blocks it contains.
- H - Halt execution and return immediately to command level.
- I - Ignore all stored conditional commands and continue execution of the program in non-debug mode.
- K <number> - Kill the specified stored commands. The parameter can include several command index numbers or ranges of command index numbers separated by commas, e.g.

K 2,4,6-8

Note that the command index number n is given in the prompt "DEBUG n:" current when the command was input.

- M - Execute %MONITOR.
- P <variable> - Print the value of the named variable.
- R - Resume execution of the program. DEBUG will force a break at the next line unless at least one conditional command is currently stored.
- S <line no>,<count> - Print <count> lines of the source listing starting at line <line no>. A prompt is issued for the name of the compiler generated listing file.

Three types of conditional expression are available; they may be used singly or in combination:

- <simple command> @ <conditional expr>
- or <simple command> @ <conditional expr>&<conditional expr>
- or <simple command> @ <conditional expr>&<conditional expr>&<conditional
expr>

Only one of each type can be specified in a single command. The three types are as follows:

- C=<variable> - This is satisfied whenever the specified variable has its value changed. If the variable is reassigned its current value, this is not detected as a change.
- R=<routine/block> - This is satisfied when the current line being executed lies within the specified routine or block. For blocks, the starting line number should be given; for routines, functions or maps, the name should be given. Note that this condition is not satisfied when the current line lies within a routine or block which itself lies within the specified routine or block.
- L=<line spec> - This is satisfied when the current line being executed lies within the line spec. The line spec is defined below.
 - <line spec> : <line>[,<line spec>]
 - <line> : <line expr> ! <line range>
 - <line range> : <line no> - <line no>
 - <line expr> : +<line no> ! -<line no> ! *<line no> ! #<line no>

The use of these is explained below:

- L=n - Satisfied when the current line number is n.
- L=n-m - Satisfied whenever the current line number lies between n and m.
- L=+n - Satisfied when the line number equals the current line number plus n (i.e. the line number current when the command was given + n)
- L=-n - Satisfied when the line number equals the current line number minus n.

Commands involving these expressions are permanently stored. However two temporary forms are provided which cancel themselves once satisfied:

- L=*n - Satisfied when the current line is encountered for the n'th time. This is intended for skipping through an iterative sequence without stopping at every iteration.
- L=#n - Satisfied when the n'th executable line after the current line is encountered. This allows control to be returned to the user after a test and branch operation, the result of which is not known in advance.

Examples

Several examples of command lines follow:

- | | |
|-----------------------------------|--|
| DEBUG 3:P VAL @ C=VAL & R=COMBINE | - Print variable VAL, whenever its value changes within routine COMBINE. |
| DEBUG 1:A S="HEADING" | - Assign "HEADING" to string variable S. |
| DEBUG 4:B @ L=10-20 & C=JIM | - Break whenever the value of JIM changes and the current line lies between 10 and 20. |
| DEBUG 8:R | - Resume execution of the program. |

Notes

1. A newline terminates a command line unless it is part of a string (i.e. preceded by a quote). Spaces are not significant unless enclosed in quotes.
2. The index number n printed in the prompt DEBUG n: is incremented only if a storable (i.e. conditional) command is specified. When commands are Killed, the index number is unchanged except for the case in which all commands are Killed, when its value returns to 1.
3. If more than one C=<variable> commands are given for the same variable, then the same R=<routine> expression (if used at all) must be used in every case. This ensures that a single copy of the variable's value is held and hence avoids inconsistencies.
4. All the simple commands, with the exception of K (kill) and F (file map), may be used with a conditional expression to form a conditional command.
5. Commands B (break) and R (resume) operate by setting a single switch. Hence if a B conditional command succeeds but is followed by an R conditional command which also succeeds, the effect of the B is lost and no break occurs.
6. Except in line specification, a number may either be expressed in decimal or hexadecimal form, e.g. 32 or X20.
7. At present, users should only DEBUG one object file at a time. Thus a program and an external routine which it calls must not both be compiled with PARM DEBUG.
8. When a command such as B @ C=JIM is given, the condition is evaluated every time a line of the program is executed. In this case the evaluation involves a search for a declaration of the variable JIM; if it is not found, this is treated as a fault. In addition if there is more than one declaration of JIM in a program (or in the line range specified), the first accessed is used subsequently and the second, whenever encountered, is treated as a fault. Hence to avoid numerous error messages it is best to give a qualifying L= or R= expression with any C= expression.

9. Variables of all types except records may be referred to in Assign and Print commands or C= conditions. However, array elements must be specified with constant subscripts.
10. For every line of a program compiled with PARM DEBUG, a call is planted to the DEBUG routine which, consequently, is called many times in the execution of the program. To minimise overheads it is advisable to reach that part of the user program which is of interest before setting up several stored commands. In particular, repetitive operations like the initialisation of arrays become very slow if several stored commands have their conditional expressions evaluated on every line executed. However if only a single command of the form B @ L=n is stored, this will incur minimum overheads.

Sandy Shaw

The Scientific Jobber

The Scientific Jobber is a batch process which provides a fast, low-overhead environment for the development of ALGOL, FORTRAN and IMP programs. It incorporates a control language processor, a source program editor, a batch program loader and a set of standard mathematical procedures.

The Jobber was first developed for ICL's VME/B and VME/K operating systems, and its availability on EMAS 2900 provides a common environment for user programs across the three systems.

The speed and low overheads of the Jobber are achieved by making available a subset of the facilities provided by the full operating system, and by supporting this subset within a single program.

Input to the Jobber can be as a batch of cards assembled by the operators, or as a file of jobs sent from a user process.

Operator's view

To the operator, a Jobber process is controlled as if it were a spooled device: there is a stream called JOBBER, for which "documents" are queued until requested by a Jobber process.

The appropriate subset of Spooler commands is as follows:

S/Q JOBBER	Display the state of all Jobber processes and list the names and estimated execution times of any files of Jobber jobs waiting to be run.
S/JOBRnn?	(JOBRnn identifies a particular Jobber process). Display the state of this Jobber process.
S/ABORT JOBRnn	Stop this process; if a batch of jobs is being executed, it will be re-appended to the Jobber queue.
S/DELETEnnnnnn	Delete a file from the Jobber queue, as identified by its unique six-digit identifier.

The single Director command required is:

D/START JOBRnn	Starts a Jobber process.
----------------	--------------------------

User view

This is fully expounded in the ERCC 2980 User's Guide (Section F), and in the ICL 2900 series publication "Using the Scientific Jobber". In the latter, ignore references to system-dependent features such as libraries, and note that the EMAS 2900 Jobber runs in ISO mode.

There is an additional input route on EMAS 2900: the user can make up a file of jobs in a normal process and LIST or SEND it to the device ".JOBBER".

Installation manager's view

Several Jobber processes can be set up, to a maximum of one per file system. For example, the ERCC 2970 currently has three Jobbers (JOBR00, JOBR01 JOBR02), corresponding to the disc packs EMAS00, EMAS01, and EMAS02.

A Jobber process is a standard process with the following changes:

- a) The basefile is set to a fixed up Jobber subsystem. This is done by the command:

```
D/SET BASEFILE JOBRnn
D/user.filename'
```

- b) The ACR of the process must be less privileged than MANAGR and more privileged than its users, if installation options and level 2 facilities respectively are required (e.g. ACR 5 on the ERCC 2970).

- c) The maximum file size for the process must be raised, currently to 2 Megabytes; this figure may be reduced later.

When required, the process can be run with the Edinburgh Subsystem, by setting the basefile to a null string.

There exists a file in each Jobber process called JOBJOURNAL. This has READ permission granted. It contains a record of work. Specifically:

- a) For each batch of jobs accessed: the name of the file and the date and time of access.
- b) For each user job: the time it was begun, the CPU time used, the number of lines printed to primary output, and the number of sequential or direct access data record transfers.

When this file is full, it is listed and a new one created. It can be used as a basis for accounting outwith the standard EMAS 2900 accounting system. In the event of a major fault, this file can also be used as ultimate diagnostic output.

Installation options

A number of installation management options are available with the Jobber. There are two levels of control:

- a) Options applicable to a particular batch of jobs (set by including a control card with the batch).
- b) Options applicable to all jobs run by a Jobber process (set by command from the MANAGR process).

Note: Installation defaults may be set for type (a) options from MANAGR, but are overridden by any batch control card.

Options for a particular batch

The default values for the options applicable to a single batch of jobs can be modified by the insertion of a card (or line) beginning //JOBBER.

The options which may be modified are as follows:

1) Facility level

There are two facility levels for file access. These are specified by the FACLEVEL parameter, which can take the value 1 or 2 (the default). At Level 1 users are not allowed to access files and all input and output for such jobs must be via the Jobber control streams. At facility Level 2, in addition to the Level 1 facilities, creation of and access to source program and data files belonging to any process with a less privileged ACR is allowed (i.e. all normal users).

2) CPU limits

The default and maximum CPU limits for each user job can be modified, by the MDEFAULT and MMAX parameters respectively.

3) Printed output limits

The default and maximum printed output limits for each job can be modified, by the LDEFAULT and LMAX parameters, respectively.

4) File sizes

The maximum size permitted for a file created by a user can be modified (the FMAX parameter).

e.g. JOBBER(FACLEVEL=2, MMAX=20)

Options applicable to all jobs

- 1) The default compilation options may be modified by setting the parameter OPTIONS, which takes a value of one or more compilation options, separated by ampersands (see the ICL document "Using the Scientific Jobber").
- 2) The amount of printed output written to a Jobber output file can be controlled by the parameter BREAK. This parameter takes an integer value, which is a line count. Each output file will be closed and printed after the completion of the Jobber job during which this count was exceeded. A new file will be created and used for the output from subsequent jobs.

Defaults

If no action is taken to set options then Jobber will be controlled by a set of defaults, fixed in the Jobber program as follows:

FACLEVEL = 2
MDEFAULT = 30
HMAX = 300
LDEFAULT = 1000
LMAX = 5000
FMAX = X'200000' (bytes)
OPTIONS = ISO"ES

Setting options from the MANAGR process

Three commands are available to override the default option values with which the Jobber program was compiled. (The file ERSC04.JOPTY must first be INSERTed in MANAGR's directory.)

- 1) To create a file in one or more Jobber processes which will contain default values to be used in preference to those used within the Jobber program (taken from file JDEFAULTS), do the following:

CREATE J OPTIONS (jobbers)

jobbers = the names of one or more Jobber processes, separated by
commas

- 2) To change particular options, do the following:

SET J OPTIONS (Jobber)

The program will prompt for data, which should be given in the form:

DATA: KEYWORD = VALUE

The reply STOP causes a return to command level.

Keywords and permissible values are as follows:

<u>keyword</u>	<u>meaning</u>	<u>conditions on value</u>
FACLEVEL	Facility level	1 or 2
MDEFAULT	DEFAULT CPU time per job	10 <=MDEFAULT<=3000 sec
MMAX	Maximum CPU time per job	10 <=MMAX<= 3000 sec
LDEFAULT	Default lines output per job	50 <=LDEFAULT<=100,000
LMAX	Maximum lines output per job	50 <=LMAX<=100,000
FMAX	Maximum file size	10 <=FMAX<= 1024 Kbyte
BREAK	Output break limit	BREAK >= 100
OPTIONS	Compilation options	(See "Using the Scientific Jobber" or the ERCC 2980 User's Guide)

- 3) To list the current option defaults for a particular Jobber process, do the following:

PRINT JOPTIONS(jobber)

Future developments

- Dynamic loading of external library routines as an installation option.
- Direct-access data files.
- Shared compilers. Currently the Jobber has its own copy of the compilers, as does the Edinburgh Subsystem. It is possible for all compilations to share a single copy of the compilers, whether executing in a Jobber process or in a standard subsystem, and in background or foreground mode.
- Accounting. It may be possible to produce standard EMAS 2900 accounting records.

A. Anderson

EMAS 2900 OPERATIONS NOTE

No: 1

Date: 7/3/78

Messages from Oper to Executive and User Processes

Currently during an EMAS session, three executive processes are created, namely "DIRECT", "VOLUMS" and "SPOOLR".

DIRECT performs the following functions:

1. file system maintenance
2. checking of log-in messages from the network
3. creation of other processes
4. dispatch of operator messages and broadcast messages to other processes

VOLUMS performs the following functions:

1. back-up
2. archive
3. reload of files from back-up and archive, and from System 4 tapes

SPOOLR performs the following functions:

1. queueing and dispatch of files from user processes destined for local or remote output devices, or for batch execution locally or remotely
2. scheduling of batch (background) jobs for local execution
3. receipt of files for local users or for local execution from local or remote devices

These three executive processes require to handle arriving request messages "on demand", i.e. they request a message ("POFF"), go to sleep if there isn't one, and take the next one if there is. These processes are said to require their messages "synchronously", and the special message forms

D/<message>	(for DIRECT)
V/<message>	(for VOLUMS)
S/<message>	(for SPOOLR)

are designed to supply <message> to the process on a "synchronous" service number.

Messages from the Oper to other processes, however, need to operate in an essentially asynchronous manner, and are therefore sent on a separate "asynchronous" service number, which is used when an "n/<message>" is typed

(e.g. 5/ST). Unlike the System 4 implementation, however, text messages are not sent using n/<message>, but via facilities in the executive process DIRECT: SET MSG, SEND MSG, BROADCAST and MSG<username> (see below).

The special Director functions, e.g. ST, XST, are requested using n/<message>. These asynchronous messages may also be sent to the executive processes if necessary, but asynchronous messages other than the ones listed below will simply be echoed with a question-mark by any process, with no other effect.

Text messages from Oper to user processes

D/SET MSG

This causes further input to DIRECT to be concatenated into a stored string. Input is terminated by a ":" (colon) on a line by itself, or after the 255th input character. When the ":" is typed, the stored string is played back on the Oper.

D/BROADCAST

This causes the stored message (see SET MSG) to be broadcast to all processes started from interactive terminals.

D/SEND MSG <username>

This sends the stored message (see SET MSG) to the user specified (if the process was started from an interactive terminal).
[Not implemented at 7th March 78.]

D/MSG <username>

This causes the next line typed to be sent as a text message to <username>.

In the last two cases, the message will be sent to <username> even if he is not logged in, but a flag 61, "Process not available", result is given. Later versions of the Subsystem will be able to pick up stored messages at process start-up.

Special Director asynchronous messages

n/ST

This causes process n to be stopped when Director for process n is next uninhibited. This is a "safe" way of killing a process, viz files and i/o streams are properly disconnected. It should normally be "immediately" effected. The process "FAILS 14".

n/XST

This causes the process to be killed instantaneously. This is potentially dangerous, as Director procedures may currently be updating a file index, for example. This should only be used if n/ST is patently having no effect. The process "FAILS 13".

The remaining asynchronous messages are for diagnostic purposes:

n/CMP

This prints the "master page tables" on the main log.

n/DIRMON i j

This sets the Director monitor variables DIRMON, DEPTH to i, j respectively.

(0 0 means "off".)

n/SIGMON i j

This sets the Director monitor variables SIGMON, MONITORING to i, j respectively.

Bits in the SIGMON word have meanings associated with the following values:

Bit-values	Action
1	print monitor
2	print string describing contingency
4	print dump of stack
8	print dump of gla
16	print master page
32	dump code around PC
64	print regs at contingency

MONITORING=1 causes PONs, POFFs, OUTs etc from the process to be monitored on the main log.

(0 0 means "off".)

General

D/PROMPT

This causes the DIRECT process to prompt for further input. The prompt is "DIRECT:". The "D/" may then be omitted from further input, and the "COMMAND" key should not be pressed.

PROMPT 0

to DIRECT causes the "DIRECT:" prompt to be removed.

D/CLOSE

This stops DIRECT, VOLUMS and SPOOLR.

J.K. Yarwood

EMAS 2900 OPERATIONS NOTE

No: 2

Date: 5/9/79

Operation of a Card Punch Under EMAS 2900

It is normally necessary, when card punch output is being produced, for operator intervention between the punching of each file. This is because it is not always possible to tell where one punch file ends and the next begins. To alleviate the problem, EMAS 2900 uses the card punch in the following way.

Before a card output file can be punched, the operator must give the SPOOLR command

S/START CPO

This means that after each file output the card punch will stop - it will not go on to the next file. Also, at this point (i.e. after the file has been punched), SPOOLR will display a line of the following form on the OPER:

CPO username filename size delivery

e.g. CPO ERCC20 CARDOUTP 3K Room_2020_JCMB

The cards should then be removed from the card punch hopper, and the operator should either interpret the first card in the pack or write on the first card the information displayed on the OPER.

The cards are punched as follows:

Card 1	username filename delivery date time
Card 2	first line of file
.	
.	
.	
Card n-1	last line of file
Card n	as Card 1

Note that the date and time on the first card is the date and time at which the file was sent for punching, not the date and time it was punched.

W. Laing

Automatic Issue of OPER Commands

At System start-up, and on command D/AUTOFILE at the OPER console, the DIRECT process attempts to connect a file of operator commands of the form D/<text>, V/<text> or S/<text>, each preceded by a time-of-day in the form hh.mm representing the times at which the commands are to be actioned. This Note describes the detailed operation of the scheme.

1. Default filename

At start-up the System attempts to connect the file MANAGR.AUTOFILE as the default file of commands. There is no message if the file cannot be found or is not permitted to DIRECT. Use of this default filename is inhibited by:

- (a) previously destroying the file or withdrawing its permission to DIRECT, or
- (b) giving command D/AUTOFILE 0 before the file system consistency check is complete, or
- (c) giving command D/CLOSEUSERS before the file system consistency check is complete.

2. Assigning a file as the command file

To assign a file as the command file, or to re-assign the default or other command file, the command D/AUTOFILE is given at the OPER. The prompt "Give Auto-filename:" appears, and the next input line should be a "6.11-character" filename. For example:

MANAGR.AUTOFILE1

The effects are as follows:

- * the file currently being used as a command file (if any) is disconnected
- * the specified file is connected in read-mode if possible
- * the relevant start-point in the file is found, as described below ("Description of operation")
- * commands up to the current time are actioned

Possible responses are:

0	successful
32	file does not exist or no access permission
33	mode conflicts with a current connect mode
37	owner not found
89	invalid file

3. De-assigning the command file

The current command file is disabled by:

- (a) D/AUTOFILE 0
- or (b) D/AUTOFILE

which disconnects the current file and attempts to connect the file given in response to the "Give Auto-filename:" prompt.

4. File format

The file must be a character file conforming to the Edinburgh Subsystem conventions, otherwise the D/AUTOFILE command gives the result 89. Lines of the file which are to be recognised by the AUTOFILE scheme must have a time-of-day in the form hh.mm as the first five characters. All other lines are ignored. The times specified must be in the range 00.00 to 23.59; lines commencing with "times" outside this range are ignored. The times must be in ascending order, though several consecutive (non-ignored) lines may have the same time. If a line is incorrectly placed, result 89 is given by the D/AUTOFILE command. The file contents are ignored.

The rest of a line having a time as its first five characters may be:

- (a) a command to DIRECT, e.g. D/USERS 30
- or (b) a command to VOLUMS, e.g. V/REQLIST
- or (c) a command to SPOOLR, e.g. S/BATCHSTREAMS 2
- or (d) a marker line, IPL point (see "Description of operation", below)

Spaces following the time are ignored. Multiple spaces in the rest of the line are treated as single spaces. In marker lines only the characters "IPL" are significant. Commands should be in upper case. Lines not conforming to the descriptions (a) - (d) above are otherwise ignored.

Note that it is not currently feasible (for example) to initiate backup in VOLUMS, because there is no way of supplying data to the VOLUMS prompt in response to the V/BACKUP command. Methods could obviously be developed.

5. Description of operation

When a file becomes assigned as the auto-command file, either at System start-up or at a successful D/AUTOFILE command, the file is searched from the end to find the most recent "IPL point" marker line (if any) whose time is earlier than the current time of day. Commands from that IPL point (or beginning of file) up to lines timed not later than the current time are immediately actioned. Each command is reported in two lines output at the OPER. For example:

```
1/ Auto-file command:
1/ S/BATCHSTREAMS 1
```

Thereafter as the time-of-day reaches the time of the next command (or commands), each command is executed, with the above two-line report on the OPER. At midnight a new pass of the file commences.

6. Example of an auto-command file

```
00.00  IPL point (redundant)
00.00  D/SCARCITY 20
00.00  D/PREEMPTAT 30
00.00  S/BATCHSTREAMS 1
      :
      :
12.00  IPL point           System development
12.00  D/SCARCITY 20
12.00  D/PREEMPTAT 30
      :
14.00  S/BATCHSTREAMS 1
      :
17.00  IPL point           System development
17.00  D/SCARCITY 20
17.00  D/PREEMPTAT 30
18.00  S/BATCHSTREAMS 1
```

J.K. Yarwood

EMAS 2900 OPERATIONS NOTE

No: 4
Date: 14/04/80

Commands to OPER in SUP26C

A picture may be displayed on a screen with the command

P picture screen

where picture may be

L OPERLOG
P Process List
S SPOOLR's picture
V VOLUMS's picture

and "screen" is the number of the screen. The PGF and PGB keys on the keyboard apply to the interactive screen (0) only. A picture on another screen can be paged forwards and backwards with the commands

F screen
and
B screen

Notes

1. All OPER stations are treated equally; the concept of MAINOP disappears.
2. To display a SPOOLR picture it is necessary to do both

S/...
and
P S screen

Similarly for a VOLUMS picture.

A. Gibbons

EMAS 2900 OPERATIONS NOTE

No: 5
Date: 14/04/80

Commands to GPC in SUP26C

1. GPC QS dev

where dev is a device mnemonic, gives the status (i.e. NOT ALLOCATED or READY or REQUEST FIRED or SENSE FIRED or QUEUED or DISCONNECTED) of the device.

2. GPC ?

gives the status of all the GPC devices.

3. GPC CDS dev on/off

where dev is a device mnemonic and on/off is either ON or OFF, is used to connect or disconnect a device (strictly speaking a stream). A device may be disconnected from one GPC and connected on another. Only devices which are NOT ALLOCATED can be disconnected, and only devices which are DISCONNECTED can be connected.

Notes

1. When an LP is connected, the repertoire indicated by the thumbwheel settings is loaded. Thus

CDS OFF
CDS ON

can be used to change the repertoire.

2. In the case of magnetic tapes, the lowest mnemonic in a cluster (stream) should be specified. The whole cluster is disconnected and connected.
3. CDS may be used only on devices known to the system at IPL time, i.e. these given by

GPC ?

A. Gibbons

EMAS 2900 OPERATIONS NOTE

No: 6
Date: 24/4/81

CHOPSUPE and Supervisor OPER Commands

Note: Where a numeric parameter is required in the following commands, it may be specified in one of two ways: as a decimal number, e.g. 123, or as a hexadecimal number, in which case it is preceded by the letter 'X', e.g. X400.

CHOPSUPE Commands

(i) PON dsno dact p1 p2 p3 p4 p5 p6

PON a record to dsno<<16! dact with parameter 'p1' and optional parameters 'p2'.....'p6'. Unspecified parameters are not zero!

(ii) SRCE srce

Set the SRCE parameter in future PONs (see (i)) to 'srce'. (SRCE will otherwise be the relevant Oper.)

(iii) PLOT tsn file disc page npages (PLOT = Program Load Off Tape)

Load 'npages' of program 'file' from magnetic tape 'tsn' to 'disc' starting at 'page'.

E.g. PLOT MAC002 2 EMAS00 X380 256

(iv) PLOD disc1 p1 disc2 p2 npages (PLOD = Program Load Off Disc)

Copy 'npages' from 'disc1' starting at 'p1' to 'disc2' starting at 'p2'. There are no checks on this command; use with care!

E.g. PLOD EMAS00 X100 EMAS01 0 55

(v) LABEL disc vol

Write the six-character volume label 'vol' to 'disc' defined as its device mnemonic.

E.g. LABEL ED01 EMAS01

(vi) ILABEL disc vol

As LABEL (see (v)) but an IPL bootstrap program is written as part of the volume label.

(vii) FORMAT drive lc uc lt ut

Format the disc on drive 'drive' in standard EMAS format, where:

lc = lower cylinder
uc = upper cylinder
lt = lower track
ut = upper track

Each pair lc, uc and lt, ut may be replaced by -1, meaning format all cylinders or tracks respectively.

E.g. FORMAT ED21 -1 -1 formats the whole disc on ED21.

(viii) GPC text

PON 'text' to the GPC routine.

Text may be one of the following:

a) ?

Give status of all devices.

b) QS dev

Gives the status of the single device whose mnemonic is 'dev'.

c) CDS dev onoff

Where 'dev' is a mnemonic and 'onoff' is either ON or OFF. This configures the device ON or OFF. Only devices which are not allocated may be configured off, and only devices which are disconnected may be configured on.

Notes

- * If the device is a tape cluster the lowest deck should be given; the whole cluster is disconnected or connected.
- * LPs have their reps loaded when connected; thus CDS OFF followed by CDS ON may be used to reload the rep.
- * CDS may not be used on FEPs (except when they have gone down), as these cannot yet be deallocated.
- * The device may be reconfigured to another GPC between a CDS OFF and a CDS ON.

d) CDM ZXn dev

The command CDM is used to configure in to the system GPC attached devices that were not GROPED at IPL time. For each GPC having a spare stream there is a dummy device with a mnemonic of the form ZXn. To configure in say, CP0, CDM is used thus:

GPC CDM ZX0 CP0

If successful the response is: GPC: CP0 ptsm DISCNCTD

Otherwise the response is: GPC ??CDM ZX0 CP0

The device can now be made available to the system with the CDS command (see above).

Notes

- * Any ZX device displayed by the command GPC ? may be used for CDM.
- * CDM may also be used to configure out devices provided they are disconnected. (i.e. have been CDS'd off.)
- * CDM is not available for OPERs or MT devices.
- * If the required device is an FEP then the following action should be taken:

GPC CDM ZXn FEn
GPC CDS FEn ON
FEPUP n

(ix) SLOAD fsys page

Load into store and activate the supervisor which starts at 'page' on file system 'fsys'.

E.g. SLOAD 0 X40

(x) DUMP tsn disc npages

Copy 'npages' of 'disc' starting at page 0 to file 1 on magnetic tape 'tsn'.

E.g. DUMP JM0001 EMAS00 X400

(xi) PRIME tsn disc npages

Write 'npages' from file 1 of magnetic tape 'tsn' to 'disc', starting at page 0.

(xii) POFFMON n (0<=n<=1)

If 'n' is zero POFF monitoring is switched off, otherwise it is switched on.

(xiii) KMON serv n (0<=n<=1)

If 'n' is zero then switch off monitoring of kernel service 'serv', otherwise switch monitoring on.

Note that this will not take effect until the kernel is loaded.

(xiv) UNPLOT tsn file disc page npages

As PLOT (see (iii)), but from disc to magnetic tape.

(xv) INH serv

Inhibit service 'serv'.

(xvi) UNINH serv

Uninhibit service 'serv'.

(xvii) DIRVSN n

When director is to be loaded, use version 'n', where $1 \leq n \leq 3$ (default is version 0).

(xviii) DT date time

Set the date and time, where 'date' is of the form ddmmyy and 'time' is of the form hhmm.

(xix) XDUMP addr len

Print a hexadecimal dump of store from virtual address 'addr' of length 'len' bytes.

(xx) REP at with

Replace the integer at virtual address 'at' with the value 'with'.

(xxi) ISR n

Display the contents of image store location 'n'. Invalid 'n' will cause a System crash!

(xxii) ISW n x

Replace the contents of image store location 'n' with 'x'.

(xxiii) SHOW addr len

Display 'len' bytes of data from address 'addr' ($\text{len} \leq 64$).

Supervisor commands

(i) The following CHOPSUPE commands are also available in Supervisor:

PON	SRCE	PLOD	PLOT	GPC	DUMP	PRIME
KMON	UNPLOT	INH	UNINH	DIRUSN	XDUMP	REP
ISR	ISW	SHOW				

(ii) STARTD

Restart DIRECT process after it has failed or been accidentally stopped.

(iii) OPER <text>

Will pass text to the Oper handler; not yet implemented.

(iv) DDUMP label page

Read 'page' from the disc with label 'label' and dump it on the Mainlog in hex.

E.g. DDUMP BUSH20 X800

(v) SLAVES n

Turn all slaves off (n=0) or on (n=-1).

(vi) P ident n

Display picture 'ident' on screen 'n' of the Oper on which the command was entered.

Idents are currently as follows:

P = Process list
L = Oper Log
S = Spooler Picture

'n' (0<=n<=3) gives the screen no, 0 being the interactive screen.
If 'n' is omitted 0 is assumed.

(vii) B n (1<=n<=3)
F n

Page the picture currently on screen (displayed by the previous command) Backwards or Forwards by one screenful. Screen 0 is manipulated by the Page Forward and Page Backward keys.

(viii) TRACE

This command is provided to control tracing options in special tracing Supervisors used in research. It has no effect on standard Supervisors.

Dual Supervisors support also:

(ix) OCP n onoff

(where onoff is either ON or OFF) Configures OCP n ON or OFF.

(x) SAC n onoff

Configures SAC n ON or OFF.

(xi) SMAC n onoff

Configures SMAC n ON or OFF.

J. Maddock
P.D. Stephens

EDINBURGH REGIONAL COMPUTING CENTRE

EMAS 2900 SERVICE NOTE

No: 1 (version 5)
Date: April 1980

Conversion to EMAS 2900 for 4-75 Users

N.B. THIS NOTE REPLACES SERVICE NOTE 1 DATED AUGUST 1979.

Introduction

This note is intended for existing users of EMAS on the 4-75 who are planning to move to EMAS 2900. It primarily concerns differences in the appearance of the Subsystem, but other areas are mentioned where relevant. It reflects the state of the System as running on the ERCC 2970 at the time of writing. More up-to-date information can be found in the ALERT and HELP information.

Essential background information

It is assumed that the reader has access to

- * EMAS 2900 User's Guide, Second Edition, January 1980
- * EMAS 2900 Information Card, Third Edition, February 1980
- * For IMP users, the IMP Language Manual, and particularly its Update 1 (February 1978)

Design intention

The user interface has been based on that used for EMAS on the 4-75. The re-implementation for ICL 2900 Series computers has been used as an opportunity to tidy up some of the inconsistencies in the command language, in the light of experience of the problems which users have encountered on the 4-75. The intention is that it should be possible to carry out all the same operations - often using identical commands.

The rest of this note should be read in conjunction with the EMAS 2900 User's Guide.

Chapter 1

File names are constructed as on the 4-75 except that both file names and partitioned file member names can be up to 11 characters.

Chapter 2

The device code .TT is no longer valid. In its place the more general forms .IN and .OUT must be used. They refer respectively to the current primary input and output. For foreground sessions these are normally both the interactive terminal. Hence:

IMP(A,AY,.OUT)

Use HELP(DEVICES) to obtain an up-to-date list of output devices.

Chapter 3

When logging on, respond to the "HOST" prompt with "2970" or "2980", not with "EMAS".

Chapter 4

Note the distinction between SS# and T# files. Note also that the name of the default compiler listing file is T#LIST.

Chapter 6

The command FILEANAL is replaced by ANALYSE, which takes three parameters:

ANALYSE (file, option, out)

See page 6-1 for details.

COPYFILE becomes COPY.

Chapter 7

The distinction between Character files and Data files is more significant on the 2900. In particular the text editors only operate on Character files. The provision of a format "C" in DEFINE makes it possible to generate Character files as output from a FORTRAN program.

The first parameter for DEFINE is now an integer; hence:

DEFINE(3,FILEAB) not DEFINE(FT3,FILEAB)

Note that DDLIST is replaced by DEFINE(?).

Chapter 11

The program loading mechanism is significantly different. However, the majority of users will not be greatly affected.

INSERT replaces INSERTFILE. It must be called explicitly for object files, other than main programs, before they are first used.

The function of APPENDLIB is now replaced by a facility provided by the OPTION command. Briefly:

APPENDLIB (CONLIB.GRAPHICS) becomes

OPTION (SEARCHDIR=CONLIB.GRAPHICS)

There is a new command ALIAS which makes it easy to modify the names of commands.

Chapter 12

The first 31 characters of entry names are significant.

TESTINT is replaced by INTERRUPT.

SSINFO is replaced by two functions UINFI and UINFS.

Chapter 16

The implementation of BACKSPACE is incomplete. In the case of Character files there is no restriction. In the case of Data files consecutive calls of BACKSPACE are faulted.

Chapter 17

The OPTION command provides more facilities than its counterpart on the 4-75. In particular it replaces USERLIB, APPENDLIB and REMOVELIB. See page 17-5.

Command changes

The list on the next page gives the names of 4-75 EMAS commands, their EMAS 2900 equivalents (where different) and an indication of which commands are significantly different on EMAS 2900.

EMAS 4-75 command	EMAS 2900 equivalent (if different)	Significant differences	Notes
ACCEPT			
ALERT			
ALGOL			
APPENDLIB	OPTION	*	See Chapters 11 and 17
ARCHIVE			
CHERISH			
CLEAR			
CONCAT			
COPYFILE	COPY		
CPULIMIT			
DDLIST	DEFINE(?)		
DEFINE		*	
DEFINEMT			
DELETEJOB	DELETEDOC		
DELIVER			
DESTROY			
DETACH		*	See Chapter 16
DISCARD			
DISCONNECT			
EDIT			
FILEANAL	ANALYSE	*	
FILES			
FINDJOB	DOCUMENTS		
FORTE			
HAZARD			
HELP			
IMP			Changes in language
INSERTFILE	INSERT		
LIBANAL	ANALYSE	*	
LINK			
LIST			
LOOK			
METER			
NEWPDFILE			
NEWSMFILE			
OBEYFILE	OBEY		
OFFER			
OPTION		*	See Chapter 17
PARM			
PASSWORD			
PERMITFILE	PERMIT		
PERMITLIB	PERMIT	*	No direct equivalent Not available
PROJECT			
QUEUES	DOCUMENTS		
RECALL			
REMOVEFILE	REMOVE		
REMOVELIB	OPTION		See Chapters 11 and 17
RENAME			
RESTORE			
RUN			
SEND			
STOP			
SUGGESTION			
TIDYLIB	TIDYDIR		
USERLIB	OPTION	*	See Chapters 11 and 17, and command NEWDIRECTORY
USERS			

R.R. McLeod

Transferring Files between the 2970 and 2980: T080 and T070

The commands T080 and T070 are available for transferring a user file from one mainframe to another. If it is a character file it is dispatched immediately along communications lines to the target machine. Otherwise it is sent along communications lines as a background job, so that in due course a line printer report of the execution of the job (on the target machine) will be received. There is no restriction on the type of file which may be transferred, but the size of the file cannot be greater than 1023 Kbytes.

Establish access to the commands as follows:

Command:OPTION(SEARCHDIR=CONLIB.GENERAL) (2970)

Command:OPTION(SEARCHDIR=ERCLIB.GENERAL) (2980)

There are two forms of the command: T070 and T080, where the suffix indicates the target mainframe. The parameters for both commands are as follows:

Command:T0xx(file, newowner.newfile, back, overwrite)

- | | | |
|------------------|---|--|
| file | - | the file or partitioned file member to be transferred (may belong to another user so long as read access is permitted). |
| newowner.newfile | - | if newowner is omitted the owner of the process issuing the command is used. If newfile is also omitted the file is given the same name on the target machine. |
| back | - | the background password of the receiving owner on the target machine. |
| overwrite | - | OVERWRITE may optionally be specified: if file "newfile" already exists in the target machine process, it will then be overwritten. |

Please contact your local Advisory Service if any difficulties are encountered in using the above facilities.

A. Shaw

Accounting and Charging Routines

These routines are held in object file ACCNTS.PDCHARGE_CHARGEY on the 2970 and 2980; the corresponding source file is ACCNTS.PDCHARGE_CHARGE_nS (2970 only).

The routines are described below.

externalroutine CHARGES (test)

This routine takes an optional parameter, either a specific username or the word TEST; in either case a test run is implied. The effects are described below.

If no parameter is given, CHARGES reads and clears the file index accounting information for every accredited user in the System. The information read is put into a store map file called JUSE which is then sent to the JOURNAL process via the SPOOLR JOURNAL queue.

Before CHARGES does this, however, it makes use of the file created for JOURNAL to update a weekly accounting file and a "cumulative use" file.

The current accounting week is established and a file for the week's accounting is created unless it already exists. The form of the file name is A29nnmmmn, e.g., A2980JUN2.

The format of the file is as follows:

(1) "ddmmyyddmmyy"

the dates (inclusive) of the accounting week

(2) #n dd/mm/yy

n specifies the charge code 1, 2, 3, 4

dd/mm/yy gives the date on which the charges following were incurred

(3) hh.mm.ss AAAAnn ppp

hh.mm.ss is the time of day at which the charge was calculated

AAAAnn is the relevant user number

ppp is the charge (in pence) incurred

Line (1) appears once only, at the start of the file. Following each occurrence of a line with format (2) there are a number of lines with format (3). The charge code determines what sort of charge is being made in the lines following:

- 1 use of 2980 or 2972
- 2 file space on 2980 or 2972
- 3 use of 2970
- 4 file space on 2970

On each call of the CHARGES routine (with the exception of test runs), two sets of charges, use and file space, are calculated for the relevant machine and appended to the current accounting week file. This file is then normally accessed during the following accounting week (i.e. when it is complete).

The "cumulative use" file is called CUSE. It is created by CHARGES if necessary. It can be listed, in whole or in part, by use of the command LISTCUSE, described below.

When CHARGES is called with a parameter, a test run is carried out. In this case the file indexes are examined but not cleared, the weekly charge filename is not prefixed by A29nn, and the names of the files generated (JUSE, CUSE, MMMn) all have "TEST" appended. JUSETEST is not transferred to the JOURNAL process; it will be overwritten in any subsequent test run.

If the parameter is a user name then only the charges for that user are calculated. If the parameter is TEST then the charges for all accredited users are calculated.

Notes

- * If file JUSE is found by CHARGES to exist, CHARGES starts by sending it to the JOURNAL queue. This should not happen, since CHARGES should have done this the previous time it was called.
- * The structure of the JUSE file is described below.
- * The structure of the "cumulative use" file is described below.
- * At the time of writing, CHARGES is invoked nightly by a batch job which causes itself to be run the next day also. Thus the charging scheme is automatic and no operator intervention is required.
- * The processing of the completed weekly accounting files is carried out as a separate stage.

externalroutine LISTCUSE (string(255) S)

This routine either creates and initialises the "cumulative use" file added to by CHARGES (see above), or, if the file exists, lists the contents.

The form of a call of the routine is as follows:

Command:LISTCUSE(usermask,out,file)

All three parameters are optional.

usermask	can be null (the default), meaning that information on all users is to be output, <u>or</u> a six character username, meaning that information on that user only is to be output, <u>or</u> a name including one or more '?' signs, specifying a group of users about whom information is to be output; e.g. ERCCI?? would give information on all users whose names started ERCI.
----------	--

out can be null (the default), meaning that output is to be sent to the primary output device (normally the user's terminal), or an output device (e.g. .LP), or a filename.

file by default this is CUSE. If a test CUSE file is to be examined (see above), this parameter must be specified as CUSETEST.

If usermask is non-null and does not contain six characters, an error is flagged.

externalroutine LISTJUSE (string(255) S)

This routine lists the complete contents of the store map file JUSE (or JUSETEST). The form of a call of the routine is as follows:

Command:LISTJUSE(file,out)

file is JUSE by default

out is .OUT by default

Refer to the description of DSFI in the EMAS 2900 Subsystem Writer's Manual for explanations of the various items listed.

externalroutine GETUSE(string (6) USER, integer FSYS, RESET, ADR)

This routine returns accounting information in a record array starting at ADR, for all accredited EMAS users (if FSYS is -1), or for all users on the disc pack specified by FSYS, or for a single user (if USER is non-null). In the latter case FSYS can be used to specify where the user's file index resides, if known.

The record array starting at ADR is assumed to be (0:n) (STATSF):

recordformat STATSF(string(6) USER, c
integer NEXT, FSYS, KINST, PTURN, KBTSLDEV, KBFSLDEV, c
 MSOCP, CONNECT, AFILE, AKB, DFILE, DKB, CFILE, CKB)

The records are held in a list structure to give alphabetic order (_NEXT giving the next item in the list). This list is terminated by a dummy record held in array element 0, with (0)_USER="ZZZZZZ". The genuine last user record thus has _NEXT=0.

The 0th record, on return from GETUSE, holds two items of information:

(0)_NEXT gives the FIRST record in the list structure
 (0)_FSYS gives the TOTAL number of (genuine) user records in the structure.

If on entry (0) NEXT is non-zero, this indicates that a record array structure already exists. In this case each user's accounting information is added in to the corresponding record (unless no such record currently exists, when one is created). When (0) NEXT is non-zero on entry, it is assumed to point to the start of the structure, and (0) FSYS is assumed to give the total number of user records in the structure.

If all the file systems are searched and more than one file index is found for a particular user, then the accounting information for each is stored in a separate record of the record array.

If RESET is 0 on entry, the accounting information is merely read from the user's file indexes. If it is 1, the file indexes are also reset.

It is expected that GETUSE will be called on a regular basis with RESET=1, and that either all the discpacks will be searched in a single call, or that GETUSE will be called for each discpack in turn, the calling program having used GET AV FSYS beforehand.

The user of GETUSE must be running at an ACR level of 5 or less.

Structure of JOURNAL File JUSE[TEST]

This file has a standard Edinburgh Subsystem file header. It is a store map file.

Following the file header, the first two words contain the packed date-and-time and the pattern X'FFFFFF03' respectively.

The remainder of the file is used to hold a record array of the structure described in GETUSE (above).

Structure of Summary File CUSE[TEST]

This file has a standard Edinburgh Subsystem header. It is a store map file.

Following the header, there are two 8-byte strings (each allocated ten bytes) which give the start and finish dates corresponding to the information held.

Thereafter a record array is mapped onto the file, from byte 20 after the end of the Subsystem header. The record array is defined as follows:

recordformat CRECF(string(6) USER, integer NEXT, USEP, FILEP)

recordarray CREC(0:1024) (CRECF)

where USER is a user name NEXT points to the next user (in alphabetical order) USEP is the accumulated use charge in pence for that user
FILEP is the accumulated file charge in pence for that user.

Note that:

CREC(0)_USER = "ZZZZZZ"
CREC(0)_NEXT = array element containing first genuine user
 alphabetical order)
CREC(0)_USEP = total number of genuine users in array.

The 0th element of the array is used to terminate the list. Thus
CREC(i)_NEXT=0 if the user in element 1 is the last one in the alphabetical
list.

J.M. Murison

SPOOLER PROCESS 2900

No. 0
Date: 25/4/77

USING THE SPOOLER.

CIRUSN 2 NOMINATE DIRECTOR SITE X280 ON SPREADER

PON 3 1 4 START PROCESS 4. (THE SPOOLER) 2) //

SNOOC REPLY 84/SPOOLER USE STARTS.

AVAILABLE COMMANDS

4/PRIME NUMBER (COPY FILES ON SPECIFIED TAPE INTO ANY TAPETS AND OUTPUT TO. 30/3/77. All living

FILE SYSTEM
TAPE SNOOC GO WRITTEN BY MACH
PROGRAM UTAP ON EMAS.

4/RELOAD NUMBER N M (COPY chapters specified from tape to file system. Tape should be 4/LI (username, filename) Claiming the line printer, supervisor must wait for specified file to be printed when finished reform the printer to supervisor.

UP TO 4 PRINTER OR RECORDERS CAN BE DONE ~~SEPARATE~~ AT ONE TIME. AT PRESENT IF A FILE EXISTS IN THE FILE SYSTEM IT IS DESTROYED AND THE NEW ONE ON TAPE PRINTED.

All living
25/4/77

ERRORS

- 1) BAD PARAMS (more than 4 at once.)
- 2) READ PAGE FAILS (A page could not be completed from tape, the file is not reloaded.)
- 4) POSITION TAPE FAILS (step to tape mark failed.)

4/ALCOC CEE The card reader is allocated to the spooler and other cards are read or a status message is output.
4/CEALCOC CEE This deallocates the card reader and stops reading.

Design Aims and Philosophy

These notes are a description of the design aims and philosophy behind the Spooler process (SPOOLR) for EMAS 2900. They are intended as discussion notes for the EMAS 2900 team although they may be used to gain an overall impression of the facilities which will be provided by SPOOLR. Many points may change in detail but few should change in principle.

Contents

0. Introduction
1. Definitions
2. Structure
3. Input of Documents
4. Output of Documents
5. Controlling Spooler
6. Spooler Configuration
7. Document Scheduling Parameters
8. Document Control Commands
9. Operator Commands

Relevant Papers

Director: Starting Batch Jobs

Communications: EMAS 2900 Communications Design and Implementation Notes

0. Introduction

The Spooler executive (SPOOLR) is a privileged System paged process which is automatically started by the System after an IPL. SPOOLR maintains a File Index on each of the on-line file system devices. In these Indexes it keeps queues of batch jobs to be run, and queues of files to be output to local peripherals and remote job entry (RJE) stations.

The facilities provided by Spooler include the control of local slow peripherals and communication with the Front End system to handle remote traffic. It accepts card, paper tape and RJE input to the System and carries out the spooling of all output to line printers, card punches, paper tape punches and RJE stations. It also controls the queuing of batch jobs, either detached from interactive terminals or entered on cards or from RJE stations.

1. Definitions

Document

A unit of spooling comprising control information and a file. Each document has a unique identifier. All references to documents are via their identifiers.

Note that the Spooler is not interested in the contents of the file, only in the control information.

Document Scheduling Parameters

The items in the control information which is part of each document. This information describes the document, e.g. its size, where it is going, where it came from.

File

In the context of the Spooler's work, various types of file, and documents without control information, are identified; e.g.:

- * A job file, containing a complete batch job to be processed.
- * A listing file, for example to be sent to a line printer.
- * A file of binary data, for example containing graph plotter output.

Queue

A set of documents, ordered by priority, and size or time within priority levels.

Queues are served by streams (see next definition).

Streams

A communications path to a specific destination or from a specific source. Four types of stream are identified:

- * Streams to local and remote peripherals (i.e. output communications streams)
- * Streams from local and remote peripherals (i.e. input communications streams)
- * Batch job streams
- * Streams to processes which receive queued files, e.g. JOBBER

Documents are received by SPOOLR on input streams and are put into queues, as determined by their scheduling parameters. Each such queue (e.g. a line printer queue) can be serviced by several streams (e.g. a stream for line printer 0, a stream for line printer 1, etc.). Whenever a queue contains a document appropriate to a stream currently inactive, SPOOLR activates the stream to deal with the document. Streams are "queue servers".

Document Control Language

This language enables control to be exercised over documents after they have been queued.

Available to: Mainframe Operators
Remote Operators
Users

Each of these groups has different access.

Spooler Configuration

This is a file set up by the System Manager to define queues, streams, their attributes, the connections between them, etc. It is read by Spooler at each IPL to construct its tables and queues.

Operator Commands

These are commands available to local and remote operators for controlling streams.

2. Structure

A document and its descriptor, which is constructed from the document scheduling parameters, is held in the file system from which it originated.

Each queue is a linked list of document descriptors. After a System crash the queues can be completely rebuilt from the document descriptors.

When a document is placed in a queue all of the streams which serve this queue are searched, and if one is idle it is kicked to start the stream to serve the queue. If none is idle the document is just queued.

At Spooler start up, depending on the default status of the streams (either open or closed), the local streams are kicked or left idle. If the stream is to or from a remote device it is idle until the RJE terminal logs on, and is accepted or rejected depending on the status of the stream (either open or closed).

Batch streams are idle until process 1 (DIRECT) requests a number of batch streams to be started.

Streams which give documents to other processes, e.g. Jobber, are idle until such a process starts and requests a document.

Streams have default attachments to queues, but these may be changed dynamically. For example, if the local line printer is down it may be useful to serve the 'LP' queue with a remote line printer stream. When a document enters the Spooling system it acquires a unique identifier. This identifier consists of the file system number on which it is queued and a number in the range 1 to the number of descriptors held per file system. All subsequent references to the document are made using this unique identifier, which never changes.

All data transfers take place directly between files and a) devices on communications streams, b) batch jobs, or c) file processors (e.g. Jobber).

3. Input of Documents

Documents are input in two ways to the Spooler:

- 1) From other processes running in the EMAS 2900 System. These input documents are sent by users to the Spooler, either to be queued for batch execution, queued for printing or queued to be sent to an RJE.
- 2) From input communications streams. These documents come from local card readers, local paper tape readers, etc, or from RJE stations. They are dealt with according to their control parameters. They may be batch jobs for execution, files to be read into the file system, or files to be queued for printing or for output on another stream.

4. Output of Documents

Documents for output or execution are queued before being served by a stream. Queues are served by streams of three types:

- | | |
|-----------------------------------|---|
| 1) Send File (Device type) | Sends a file to a local or remote device. |
| 2) Start Process (Job type) | Initiates a batch job to process the document. |
| 3) Give to Process (Process type) | Sends a file to a process which requests documents. |

These queue servers only take files from queues when the server's parameters match those of the document to be handled. For example, the cpu time required by a job must be less than or equal to the current maximum on that stream; similarly with priority.

The 'Device Type' stream interface (type 1)

This uses the Communications Controller commands, i.e. CONNECT STREAM, ENABLE STREAM, etc. See the document "EMAS 2900 Communications Design and Implementation Notes".

The 'Job type' stream interface (type 2)

Batch input

The message to start a batch job is of the following structure:

integer DEST, SRCE, IDENT, FSYS, string (11) SPOOLR FILE

Contents of SPOOLR FILE:

VSN	<u>integer</u>
FSYS	<u>integer</u>
USER	<u>string</u> (6)
BACKPASS	<u>string</u> (6)
IDENT	<u>integer</u>
INST CNTR	<u>integer</u>
FILENAME	<u>string</u> (31)
OUTDEVICE	<u>string</u> (15)

The reply is of the form:

integer DEST, SRCE, IDENT, FLAG

FLAG: 1 - SYSTEM FULL
 2 - INVALID USERNAME
 3 - INVALID PASSWORD
 4 - ALREADY LOGGED ON
 5 - CANNOT START PROCESS
 6 - WORK FILE FAILURE
 7 - NO USER SERVICE
 8 - SPOOLR FILE NOT AVAILABLE

When a batch job stops, a message is sent to Spooler from process 1 (DIRECT):

recordformat DEST, SRCE, integer ID, CONCURR, MOREJOBS, REASON

CONCURR = Number of processes still running with username.
MOREJOBS = Number of batch process to be started.

Note that if ID=-1, only MOREJOBS is relevant.

The 'Process type' stream interface (type 3)

a) REQUEST Process asks Spooler for next file in specified queue.
 Spooler replies when it has a file to send:

REPLY name of file

b) FINISH Process replies that it is finished with file.

c) REQUEUE Process requests spooler to requeue file.

5. Controlling Spooler

There are four ways of controlling the Spooler process. These are more fully described in sections 6 to 9, but a brief explanation is given here.

1) Configuring Spooler

Each time the Spooler is started, a configuration is read in which defines the queues and streams to be used. This configuration is normally defined by each installation's System Manager.

2) Document Scheduling

Each time a document is sent to the Spooler, either from a process or on an input stream, parameters describing the document must be supplied. They are used to schedule the document, and are equivalent to the information found in the JOB card of other systems.

3) Document Control Commands

These commands are available to display the status of documents and allow their parameters to be changed once they are in the Spooling system. For example, a command is available to delete a queued document. There are different classes of access to documents for mainframe operators, remote operators and interactive users. It is also possible to enquire (for a limited period) about documents after they have left the Spooling system.

4) Operator Commands

Mainframe and remote operators require commands to control the streams in which their documents are being handled. Mainframe operators also require facilities to accept and refuse RJE log-ons.

6. Spooler Configuration

The Spooler configuration is an installation-dependent file of parameters which specifies to the Spooler which queues it has to maintain, the default parameters for documents put into these queues, and the streams which are to serve these queues. Streams are of various types: input and output communication streams to local peripherals, input and output streams to Remote Job Entry stations, streams to start batch jobs, and streams to send documents to other processes (e.g. JOBBER).

Queues are specified by giving them a name and a specification of default parameters for documents to be placed in that queue.

Examples

1) Defining a line printer queue:

```
QUEUE = LP           ! The queue is to be known as LP.

USER = SPOOLR        ! Documents which do not have USER specified are
                      ! to be known as SPOOLR documents.

DELIVERY = SPOOLED OUTPUT ! Documents with no delivery set are to
                      ! have this delivery.

START = 32           ! Byte in file at which output is to start (by
                      ! default).

PRIORITY = 1         ! The default priority of the document.

TIME = NA            ! This is not applicable to a line printer queue.

FORMS = 0            ! Default forms required.

MODE = ISO           ! Default output mode.

COPIES = 1           ! Default number of copies to be output.

RERUN = YES          ! Default is to reprint document if caught in a
                      ! System crash.
```


2) Defining a local line printer stream:

RMT = LOCAL ! Location of device.
STREAM = LPO ! Name of device.
STATUS = OPEN ! Default status of device at SPOOLR startup.
NAME = X'06000000' ! Adaptor no.: (dev type)<<24 ! (dev no.)<<16
etc.
QUEUE = LP ! Queue to be served.
LIMIT = 256 ! Maximum size of document to be accepted, in
kilobytes.
FORMS = 0 ! Default forms type in printer.
MODE = ISO ! Default output mode.
LOWEST PRIORITY = 1 ! Lowest document priority to be accepted.
TYPE = DEVICE OUTPUT ! Type of stream.

Note: Streams only accept documents when

STREAM LIMIT $\geq$ DOCUMENT LIMIT (SIZE or TIME)
and LOWEST PRIORITY $\leq$ DOCUMENT PRIORITY
and STREAM FORMS = DOCUMENT FORMS
and STREAM MODE = DOCUMENT MODE

Similar queues can be defined for batch jobs and documents queued for handling by other processes.

There are three types of output stream which can be defined:

- 1) DEVICE (document sent to a device)
- 2) JOB (document input to a batch process started)
- 3) PROCESS (document input to a running process, i.e. JOBBER)

Other parameters specified in the Spooler configuration include the number of documents which can be simultaneously queued and the number of document descriptors to be held per file system.

7. Document Scheduling Parameters

These are set by the user when he sends his document to be put into a Spooler queue or read into a file. Not all of these parameters are relevant for all queues, and defaults are supplied by the installation for each parameter where relevant.

<u>Name</u>	<u>Use</u>
USER	Name of the user owning the document.
PASS	Background password of user.
DEST	Name of queue or file the document is destined for.
SRCE	Name of file or device the document came from.
TIME	Cpu limit in seconds if document is a job.
DELIV	Delivery information of output from job or of file being output.
PRIO	Priority at which document is to be queued.
COPIES	Number of copies of document to be output.
FORMS	Form type required for output.
MODE	Mode of output: ISO, EBCDIC or BINARY.
ORDER	Order of job execution. Job run after all jobs for this user with value of ORDER < current run.
START	Byte in file at which output is to start.
LENGTH	Number of bytes to be output.
RERUN	Specifies whether document to be re-queued if caught in a System crash.
TAPES	Number of tapes required by the job.
DISCS	Number of discs required by the job.

Queue names are distinguished from filenames by their first character being a full stop.

Examples of document scheduling parameters

1) Sending a file to the line printer queue .LP:

DEST=.LP, USER=ERCC20, SRCE=LISTFILE, START=32, LENGTH=7194, DELIV=ERCC

2) Detaching a job to the batch job queue:

DEST=.BATCH, USER=ERCC20, SRCE=JOBFILE, TIME=75

When Documents are input from local or remote peripherals, the document scheduling parameters must be distinguished from the document. This is done by starting each line containing document scheduling parameters with //DOC and terminating the document with //DOCEND.

Example

Inputting a file to be listed on the line printer.

//DOC DEST=.LP, USER=ERCC20, DELIV=FRONT DOOR

<file to be printed on the line printer>

//DOCEND

8. Document Control Commands

Once documents are queued by the Spooler for execution or for output, it is necessary for local (i.e. mainframe) operators, RJE operators and interactive users to be able to control the documents in the queues and enquire about their status. Below are described a set of commands to facilitate this. There are three classes of access to these documents:

- 1) Mainframe operators: complete access.
- 2) RJE operators: read access to all queues, but can only change documents queued for their RJE or sent from their RJE.
- 3) Users: read access and change access to their own documents only.

<u>Commands</u>	<u>Params</u>
COUNT	<rje><queue><docs>
FIND	<rje><queue><docs>
DISPLAY	<rje><queue><docs>
RUSH	<rje><queue><docs>
HOLD	<rje><queue><docs>
RELEASE	<rje><queue><docs>
DELETE	<rje><queue><docs>
MOVE	<rje><queue><docs> TO <rje><queue>
PRIORITY	<rje><queue><docs><numeric>
COPY	<rje><queue><docs> TO <rje><queue>
ROUTE	<rje><queue><docs> TO <rje><queue>

If command comes from

- 1) Mainframe operator: nothing is done
- 2) RJE operator: <rje> is set
- 3) A user: <username> is set

where:

<rje> = <rmtname> / <>

<queue> = <queue> / <>

<docs> = <ident> / 'ALL' / <username>'.'<ident> / <>

<u>Commands</u>	<u>Use</u>
COUNT	Count all documents as specified by params. E.g.: 1) COUNT RMT25 counts all documents queued for RMT25 2) COUNT RMT25 BATCH counts all documents owned by RMT25 in batch queue 3) COUNT RMT25 BATCH ERCC20 counts all documents sent by RMT25 into the BATCH queue for user ERCC20.
FIND	Give summary of specified document.
DISPLAY	Display all information about documents as specified by parameters.
RUSH	Rush the document to the head of its queue.
HOLD	Hold the document specified in the queue.
RELEASE	Release the document specified from the held state.
DELETE	Delete the documents as specified.
MOVE	Move the documents specified to another queue.
PRIORITY	Set the priority of the documents specified.
COPY	Put copy of the documents specified in another queue but also leave them in the current queue.
ROUTE	Set the destination of output from a batch job.

9. Operator Commands

Some commands are available only to the mainframe operator, e.g. the accepting or rejecting of RJE log-ons, but other stream control commands are available to remote and local operators. Remote operators can only control streams connected to their RJE.

Mainframe only commands

<u>Command</u>	<u>Parameters</u>	<u>Use</u>
ACCEPT	<rje> or ALL	Allow RJE(s) to log on.
REFUSE	<rje> or ALL	Prevent RJE(s) from logging on.
WINDUP	<rje> or ALL	Allow current transaction to complete and log off RJE.
ABORT	<rje> or ALL	Log off RJE without finishing current transaction.
BROADCAST	<message>	Send message to all currently logged on RJE's.
NOTICEBOARD	<message>	Set message to be sent to all RJE's at log on.

Device commands for all operators

<device> = <RJE>'.'<devicename>/<devicename>

<device> FORMS = n

<device> LIMIT = n

<device> PRIORITY = n

} Set device/stream characteristics.

<device> TO <queue>

Start device serving specified queue.

<device> ?

Find status of device.

BACKSPACE <device>

FORWARDSPACE <device>

INTERRUPT <device>

RESTART <device>

CONTINUE <device>

} Device control commands not
relevant to all devices.

W. Laing

Finding out about Spooler's Configuration from a Subsystem

To find out which queues are available to a Subsystem a file belonging to Spooler called SPOOLR.CFILE must be connected in read shared mode, with the file system number specified as -1. File "CFILE" has a standard Edinburgh Subsystem header with the last two type-dependent words specifying the number of Spooler's queues and the number of Spooler's streams. The format of the file header is defined below.

Spooler has this file connected in write mode with read access permitted to all users. Therefore it should be borne in mind that some of the values in this file are constantly changing. Two IMP record arrays are maintained in this file, one describing the queues in the spooling system and the other the streams managed by the Spooling system.

Relevant record formats and sizes

1. File header format. Size 32 bytes.

```
recordformat FHF(integer END, START, SIZE, TYPE, SPARE, c
    DATETIME, QUEUES, STREAMS)
```

START specifies the number of bytes from the start of the file at which the record array describing the queues starts. The record array describing the STREAMS starts after the queues array, i.e. (START + QUEUES*QUEUE ENTRY SIZE) bytes from the start of the file.

QUEUES specifies the size of the record array describing the queues in the configuration.

STREAMS specifies the size of the record array describing the streams in the configuration.

2. Queue entry format. Size 128 bytes.

```
recordformat QUEUEF(string(15) NAME, c
    bytearrayarray STREAMS(0 : 15), c
    string(7) DEFAULT USER, c
    string(31) DEFAULT DELIVERY, c
    integer DEFAULT START, DEFAULT PRIORITY, DEFAULT TIME, c
    DEFAULT FORMS, DEFAULT MODE, DEFAULT COPIES, DEFAULT RERUN, c
    LENGTH, HEAD, MAX LENGTH, MAX ACR, SPARE2, SPARE3, AMOUNT)
```

NAME specifies the name by which the queue is known.

STREAMS specifies the streams which are serving this queue. If an element in the array is non zero it contains a stream number i.e. an index into the stream record array.

DEFAULTS various default values a document is given when added to the queue if they are not already specified.

LENGTH the current number of entries in the queue.

MAX LENGTH the maximum length to which the queue is allowed to grow.

MAX ACR the maximum ACR level at which a user process can be and still be able to access the queue.

AMOUNT the total amount, i.e. bytes or seconds, currently queued. If
DEFAULT TIME ≤ 0 then the amount is in bytes, otherwise the
amount is in seconds.

3. Stream entry format. Size 80 bytes.

recordformat STREAMF(string(15) RMT, DEVICE, c
integer STATUS, NAME, QUEUE, LIMIT, c
byteinteger FORMS, LOWEST PRIORITY, HEADER TYPE, c
HEADER NUMBER, integer SPARE1, COMMS STREAM, SECTION, c
BYTES SENT, BYTES TO GO, SERV ROUT, DOCUMENT)

RMT the location of the device or batch stream.

DEVICE the name of the stream.

STATUS the current status of the stream:

- 0 - CLOSED
- 1 - OPEN
- 2 - IDLE
- 3 - ACTIVE
- 4 - CONNECTING
- 5 - DISCONNECTING
- 6 - ABORTING
- 7 - SUSPENDING

NAME is only relevant for input or output streams. NAME is derived
from three numbers:

1. DEVICE TYPE $\ll 24$
2. DEVICE NUMBER $\ll 16$
3. EXTERNAL STREAM NUMBER if DEVICE TYPE is a Front End.

Device Types:

- 3 - Card Punch
- 4 - Card Reader
- 6 - Line Printer
- 14 - Front End Processor

QUEUE the queue number which this stream is attached to, unless it is
an input device, when the queue number specifies the output queue
for that input device.

LIMIT the current limit on that stream, i.e. in Kilobytes or Seconds.

FORMS the current forms type selected.

LOWEST PRIORITY the lowest priority of document which the stream will
currently handle.

HEADER TYPE the type of banner generated if the stream is an output device:

- 0 - NONE
- 1 - LINE PRINTER
- 2 - TAPE PUNCH
- 3 - CARD PUNCH

COMMS STREAM the communications stream currently allocated to the stream if the stream is an input or output device.

SERV ROUT specifies the type of stream:

- 0 - Output Device
- 1 - Input Device
- 2 - Job, i.e. batch job
- 3 - Process, i.e. JOBBER, JOURNAL

DOCUMENT if non-zero, specifies the current document being handled by the stream. It is interpreted in the form FSYS<<24!DOC NO, expressed as a six-character string "nndddd", with "nn" in the range 00 - 99 and "dddd" in the range 0001 - 9999.

W. Laing

The attached note describes the currently available facilities for file and job input to EMAS 2900. As the service develops, new facilities will become available, in particular new document scheduling parameters.

At the present time input to the 2970 is available from

1. EMAS 4-75 (The 2970 is known as .P49).

2. The card reader on the 2970.

3. The card reader and paper tape reader on the slow devices machine.

Input from JCMB Job Reception on Level 3 and from Buccleuch Place Lane is currently under test. You are urged to try these new facilities and send any comments or suggestions to Bill Laing, Room 2020, JCMB.

Input of Files and Jobs to EMAS 2900

The Input of Files and Jobs to EMAS 2900 is controlled by the paged System process SPOOLR. SPOOLR works with documents rather than with files. A document is made up of scheduling parameters, equivalent to a JOB statement on some other systems, plus the file to which they relate. SPOOLR is not interested in the contents of the file, only in the values of the scheduling parameters.

Using SPOOLR, it is possible to read files into disc storage and add files to System queues from cards or paper tape. This is usually achieved by submitting the file, with appropriate document scheduling parameters, to be read by a card reader or paper tape reader. In this case, an EMAS 2900 File Input Card should accompany the cards or paper tape(s). Alternatively, input can be sent from terminals connected to the medium speed network, and files can be transferred from other processors connected to the network. The details of these operations are subject to changes as the network is extended and modified; consult the HELP information for current facilities.

Card reader input

Cards can only be read in a mode involving translation from IBM 029 card code to ISO internal code. The file produced is an Edinburgh Subsystem character file. The following rules apply:

- * All 80 columns are read.
- * Trailing spaces are deleted; that is, all the spaces following the last non-space character on each card are excluded from the file.
- * A Newline character is inserted in the file to signify the end of each card.

Paper tape input

Paper tape can only be read in one mode: ISO-coded 8-hole even-parity. The following rules apply:

- * All characters with odd parity are converted to the SUB (ISO 26) character.
- * The parity bit is set to 0 on input, whatever its value on the paper tape.
- * Null (no holes punched) and Delete (all holes punched) are ignored.
- * Carriage return characters are ignored when they are adjacent to newline characters.

- * Trailing spaces (space characters immediately before a newline character) are ignored.
- * All paper tapes must have at least 12 inches of run-out at both ends.

Remote processor input

Files are normally constructed on the remote processor with the document scheduling parameters included; they are then sent via the network to the EMAS 2900 System. The following rules apply:

- * The final //DOC statement (see below) may be omitted from the file sent.
- * The maximum size of file which can be read is 8 Megabytes; however, it may be lower for individual users (usually 1 Megabyte at present).

Scheduling parameters for file and job input

A document is input in the following form:

```
//DOC parameter=value,parameter=value,...
```

```
contents of file
```

```
//DOC
```

The first DOC statement marks the start of a document and contains the scheduling information, specified as a series of parameter assignments; the relevant parameters are given in Table 1 below. Each assignment is terminated by a comma or newline (or end of card). If a line finishes with a comma, this implies that the DOC statement is continued on the next line. As many lines as desired may be used.

At least one space must follow the word DOC. Thereafter spaces are ignored unless they lie between the first non-space character and the last non-space character of the parameter value. Thus

```
//DOC PASS = 1234, U S E R = ERCC99,  
      DEST= FILE, NAME= NEW1,  
      DELIV = BLOGGS Chemistry
```

is equivalent to

```
//DOC PASS=1234,USER=ERCC99,DEST=FILE,NAME=NEW1,DELIV=BLOGGS Chemistry
```

Note that the DELIV parameter value can include spaces (as above), but not commas.

Parameter	Meaning
USER	The six-character username of the process receiving the file or executing the job.
PASS	The background password of the USER specified.
NAME	If the DEST specified is FILE it is the name to be given to the disc file to be created. It must be 11 characters or less, starting with a letter. A member of a partitioned file (an Edinburgh Subsystem file type) may not be specified. If the DEST specified is not FILE it is the name to be associated with the file as it is queued, executed or printed by the System.
DEST	The destination of the document. Table 2 gives the valid DESTinations which can be specified.
DELIV	Up to 31 characters of delivery information to be used on any output produced. If DELIV is not specified the default delivery for the user is used. Note that specifying DELIV does not change the user's default delivery.
TIME	The OCP time in seconds required by the job.
RERUN	Whether the job is to be rerun if caught in a machine crash. Should be specified as YES or NO.
COPIES	The number of copies of output required. Valid values 1-255.
FORMS	The forms type required for output. Valid values 1-255.
ORDER	The relative priority of execution of the job with respect to other jobs with the same DEST belonging to the same user. Thus, for example, if there are two BATCH jobs belonging to the same user in the system at the same time with ORDER values of 2 and 5, the job whose ORDER is 2 will be run first. Jobs must be submitted in the required running order. Valid values 1-255.
AFTER	The document is to be held in the queue until after the date and/or time specified. Valid values are of the form: date time time date date time where "date" is in the form dd/mm/yy, and "time" is in the form hh.mm.ss. If both "date" and "time" are given they must be separated by a space. See example (8) below.
FSYS	The file system on which the USER specified resides. This is not normally necessary: it is only required when the USER is accredited on more than one file system. Valid values 0-99.

Table 1: File and Job Input Scheduling Parameters

Notes

- * Not all of the parameters given above have to be specified - there are defaults. The defaults can differ for each document destination. If a file is to be created or a batch job executed, USER and PASS must be specified.
- * Only one file can be created per //DOC ... //DOC set of statements.
- * The //DOC statements must both start on a new line; this means that for paper tape input there have to be a line feed character before and after both //DOC statements.
- * The NAME chosen for an input file should not be the same as the name of an existing disc file belonging to the specified USER. If it is, the existing file will be retained and the one being read in will be ignored.
- * Files are CHERISHED automatically on input.
- * Files may be placed immediately in output queues for printing or routing to a remote processor.

All valid output queues,
i.e. LP, GP, LP15 etc.

JOBBER

BATCH

FILE

Table 2: Valid Document DESTinations

Examples

1) //DOC USER=AAAA99,PASS = ABAB,NAME=DATAFILE23,DEST=FILE

1 2 3 4 5 6
72 341 896 247 839

.
.
.
-1 -1
//DOC

2) //DOC USER=ABCD01,DEST=FILE ,NAME=PROGA,PASS=XZ09

%BEGIN
%INTEGER X,Y,Z

.
.
.
%ENDOFPROGRAM
//DOC

- 3) //DOC DEST= BATCH,USER=AAAA99,PASS=BACK, NAME=BATCHJOB,
TIME=20,DELIV=JCMB Room 2020
DEFINE(1,.LP)
RUN(PROGY)
STOP
//DOC

- 4) //DOC DEST=JOBBER
.
.
JOBBER commands, etc.
.
.
//DOC

- 5) //DOC DEST=BATCH,USER=AAAA99,PASS=BACK,ORDER=1,
TIME=300
.
.
.
//DOC

- 6) //DOC DEST=BATCH,USER=AAAA99,PASS=BACK,ORDER=2,
TIME=200
.
.
.
//DOC

- 7) //DOC DEST=BATCH, USER=AAAA99, PASS=BACK, ORDER=3,
TIME=100
.
.
.
//DOC

- 8) //DOC DEST=BATCH,USER=AAAA99,PASS=BACK,NAME=AFTERJOB,
AFTER=10/05/79 12.30.00
.
.
.
//DOC

Notes on the examples

- 1) Creation of disc file AAAA99.DATAFILE23; a file of the same name must not already exist. User AAAA99's background password must be ABAB.
- 2) Creation of disc file ABCD01.PROGA. Note that the space following FILE is ignored.

- 3) A batch job. Note that the delivery information (parameter DELIV) is not enclosed in quotes and that, because of the rules about spaces in DOC statements, the space characters in the delivery information are preserved.
- 4) Note all that is required (DEST=JOBBER) for JOBBER input.
- 5),6),7) If these three background jobs were submitted in that order, they would be run in the order specified via parameter ORDER - i.e. (5) then (6) then (7). If ORDER had not been specified, the order of running would have been (7) then (6) then (5), because of the TIME parameter settings; other things being equal, the shortest job will be run first.
- 8) This shows how a job can be held until after a specific date and time. Note that the use of AFTER does not guarantee that it will be run at the time specified, only that it will be run after this time.

W. Laing
J. Murison

Spooler Information Facilities for Subsystems

This is a brief description of how a subsystem writer can provide

(a) commands to find out about documents queued by SPOOLR, and

(b) commands to delete queued documents

There are currently three commands which can be issued by a subsystem:

QUEUE queueName, filename

DELETE document identifier

FIND document identifier, filename

where underlined parameters are obligatory.

These commands should be sent to SPOOLR using the DMESAGE procedure. In the message string they must be preceded by "COMMAND ". For example:

```
MESSAGE = "COMMAND QUEUE BATCH,".BFILE
LEN = LENGTH(MESSAGE)
ERR = DMESAGE("SPOOLR",LEN,1,-1,ADDR(MESSAGE)+1)
```

(See the EMAS 2900 Subsystem Writer's Manual, Appendix 1, for details of DMESAGE.)

Other examples of messages which could be sent to SPOOLR via DMESAGE:

COMMAND QUEUE ,FRED

COMMAND DELETE 010236

COMMAND FIND 020937,FRED

The general error flags which may be returned by these calls are:

201 Bad Params

214 Command Not Known

The parameters given should be of the form:

queueName The name of a SPOOLR queue, i.e. up to fifteen characters long, not preceded by a "."

filename A valid EMAS 2900 file name

document identifier A six character numerical identifier, of which the first two characters specify the caller's file system.

Some comments about these commands:

QUEUE

This command returns information described below about all of the files a user has in SPOOLR's queues or in a particular queue. The filename specified is created in the callers file index and should not already have existed. Its format is described below. If the queue specified does not exist the effect is as if the caller had no files in the queue. The error flags returned by this command are:

- 15 No Free File Descriptors
- 16 File Already Exists
- 17 Insufficient Free Cells
- 18 Bad Filename
- 236 SPOOLR File Error

DELETE

This command takes only one parameter: a document identifier. The document specified must be in a queue for this command to take effect. For example, a job in execution or a file currently being printed cannot be deleted. The result of calling this command is passed back in the flag from DMESSAGE. The error flags returned by the command are:

- 205 Document Not Queued
- 213 Invalid Descriptor
- 236 SPOOLR File Error

FIND

This command takes two parameters both of which are mandatory: a document identifier and a file name. The comments about the filename given under the command QUEUE also apply here. The document descriptor may refer to a document which is not currently in a queue; for example, the document may be being executed or may have been deleted. This command currently can trace the last 1000 documents on each file system. The error flags returned by the command are:

- 15 No Free File Descriptors
- 16 File Already Exists
- 17 Insufficient Free Cells
- 18 Bad Filename
- 213 Invalid Descriptor
- 236 SPOOLR File Error

File returned by SPOOLR

The file returned by SPOOLR has the standard Edinburgh Subsystem header shown below. Note that if FILE HEADER_START = FILE HEADER_END then the file is empty.

```
%RECORDFORMAT FHF(%INTEGER END, START, SIZE, TYPE, SPARE0, %C
                   DATETIME, SPARE1, SPARE2)
```

The file returned contains (FILE HEADER_END - FILE HEADER_START)//256 entries. Each entry is 256 bytes, in the following format:

```
%RECORDFORMAT INFOF(%INTEGER VSN, STATE, %C
  %STRING (7) IDENT, USER, %C
  %STRING (15) DEST, SRCE, OUTPUT, %C
  %STRING (31) NAME, DELIVERY, %STRING (63) MESSAGE, %C
  %INTEGER DATE AND TIME RECEIVED, DATE AND TIME STARTED, %C
  DATE AND TIME OUTPUT STARTED, DATE AND TIME DELETED, %C
  DATA START, DATA LENGTH, TIME, OUTPUT LIMIT, PHYSICAL SIZE, %C
  PRIORITY, START AFTER DATE AND TIME, AHEAD, %C
  %BYTEINTEGER FORMS, MODE, COPIES, ORDER, RERUN, TAPES, %C
  DISCS, FAILS)
```

where:

VSN The format version. Currently 1.

STATE The state of the document. The states are:

- 0 Deleted
- 1 Queued
- 2 Sending
- 3 Running
- 4 Receiving
- 5 Processing

IDENT The document identifier: two characters specifying the file system and four characters specifying a unique value.

USER The username of the caller.

DEST The name of the queue which the document is or was in.

SRCE The name of the file or remote which the document came from.

OUTPUT Not currently used.

NAME The name the user gave to the document.

DELIVERY The delivery information to be sent with the document when it is output.

MESSAGE Not currently used.

DATE AND TIME RECEIVED The date and time at which the document was placed in the queue.

DATE AND TIME STARTED The date and time at which the document started execution or printing.

DATE AND TIME OUTPUT STARTED Not currently used.

DATE AND TIME DELETED The date and time at which the document was deleted.

DATA START If TIME is zero, the offset from the start of the file of the data which is to be output.

DATA LENGTH If TIME is zero, the number of bytes to be output.

TIME The job cpulimit in seconds.

OUTPUTLIMIT Not currently used.

PHYSICAL SIZE Not currently used.

PRIORITY The current priority of the document. Note that negative values specify that the document is HELD at the modulus of the value given.

1 VLOW
2 LOW
3 STD
4 HIGH
5 VHIGH

START AFTER DATE AND TIME The date and time until which the document is HELD in the queue.

AHEAD The accumulated bytes or seconds queued in front of this document.

FORMS The forms type required by this document.

MODE The mode which the data in the document is in. The values are:

0 - ISO
1 - EBCDIC
2 - BINARY

COPIES The number of copies to be output. Note that 0 means 1.

ORDER The order in which the document is to be removed from the queue, relative to other documents. Only relevant if non-zero.

RERUN Whether the document is to be requeued or deleted if caught in a System crash. 0 - No; 1 - Yes.

TAPES Not currently used.

DISCS Not currently used.

FAILS The number of times the document has failed in execution. Only relevant if RERUN = Yes.

W. Laing