3.2.3 - RAU handler (H : RAU)

Any task can request an input output on the data bus through a CSV S = IØ with a control clock as a parameter.

The handler will support the S/S and EXP RAU H/W :

- S/S RAU :

  Basic 64 flexible inputs
  32 ON/OFF

  + 1 module + 64 flexible inputs
  + 32 ON/OFF

- EXP RAU :

  4 serial outputs

  4 serial inputs + request

  + time module

3.2.3.1 - Handler H1

The following fields of the IOCB will be meaningfull to the RAU handler :

EVE	IND
ORDER	LNAME
ADBUF	
COUNT	
RAU COMMAND	

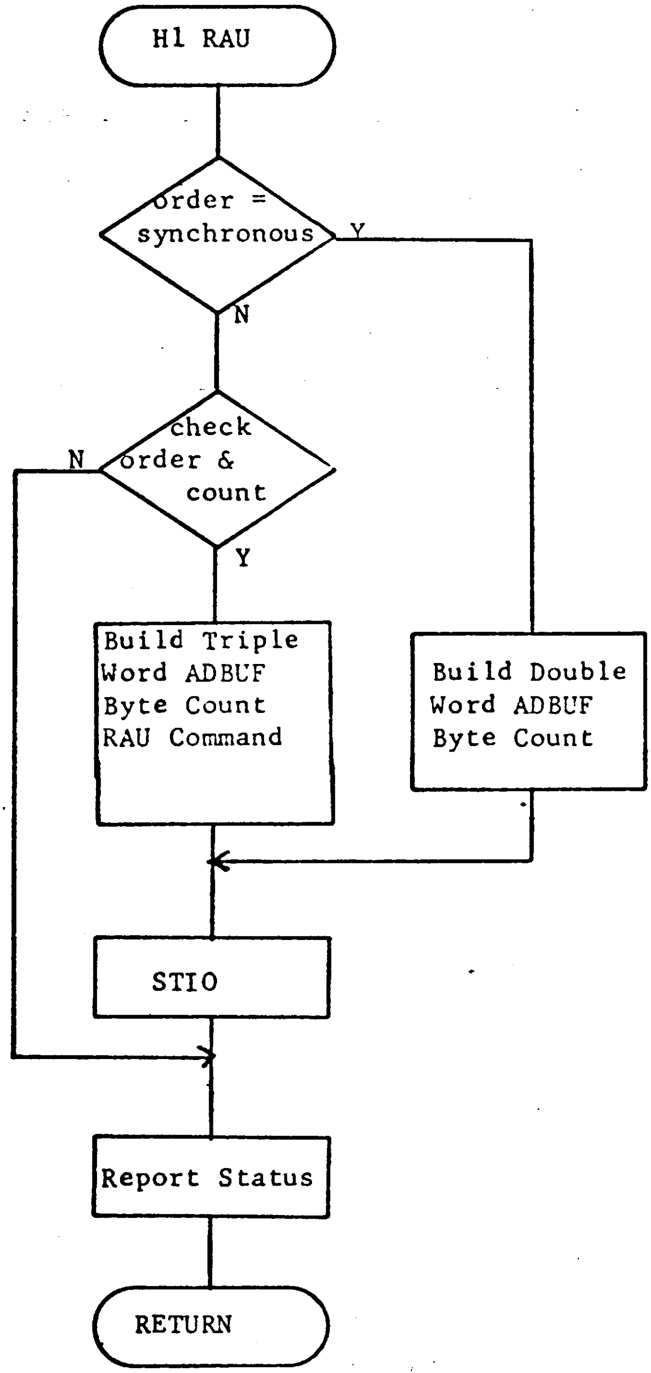
There are two kinds of orders :

- synchronous which are predefined commands included in predefined I/Ø lists,

Only limited checks are performed by the handler.

- asynchronous which can be randomly requested by an application task. Complete checks of the IØCB are performed by the handler.

Such a request will lead to only one command (described by the IOCB) to be sent to the addressed RAU.



RAU H1 FLOWCHART

- ADBUF : address of buffer
  synchronous : gives the first IO sublist address
  asynchronous : user buffer address

- COUNT : Buffer length
  synchronous : gives the number of commands in the 1st IO sublist
  asynchronous : user buffer length

- RAU COMMAND : used only for asynchronous requests :
  Bit pattern for 3rd word of command triplet (RAU address + line number)

- ORDER : SYNCHRONOUS
  The two words found in ADBUF and COUNT are used as double words for the STIO instruction to send the IO sublist to the RAU.

. SINGLE READ (ANALOG/DISCRETE)

The analog/discrete data will be read in a one word buffer.

. ON/OFF

ON/OFF are levels and remain in their state until the opposite command is sent in.

These outputs commands are used to set actuators in the ON/OFF states.

- For ECOS the following orders will be processed by the RAU handler.

. WRITE SERIAL

The calling task provides a buffer of a maximum of 32 words to be sent on one of the 4 serial outputs.

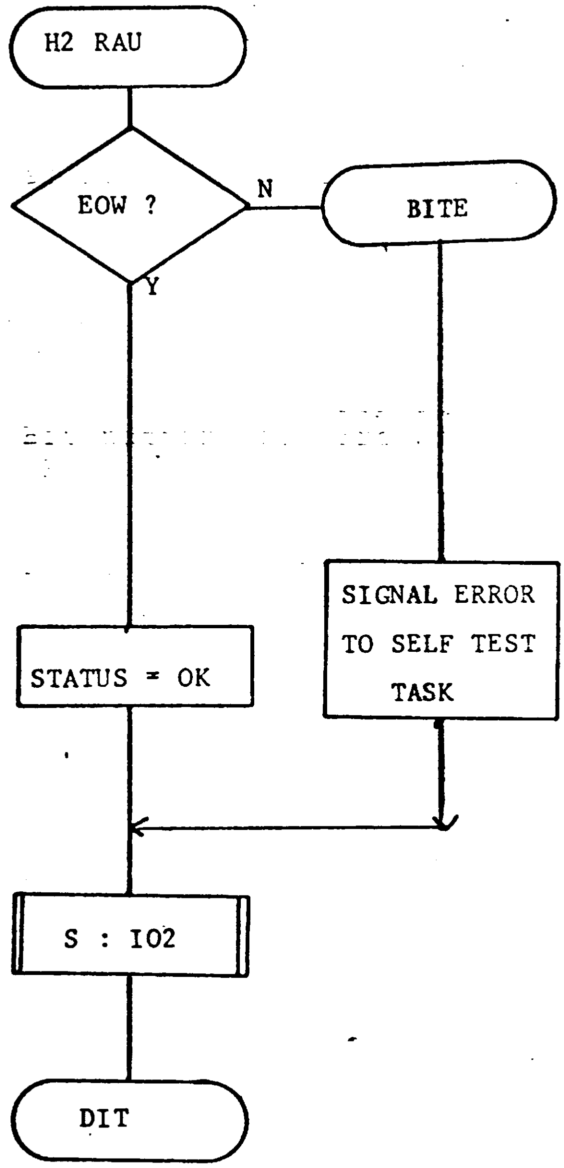
. READ SERIAL

The calling task provides a buffer of a maximum of 33 words to read up to 32 data words plus the number of effective words (from 0 to 32).

3.2.2.2 - Handler H2

H2 terminates the end of a transfer for the current request.

If an error occurs, the data will be passed to the self test task and the user will be warned.



RAU H2 FLOWCHART

3.2.2.3 - Asynchronous commands

For SCOS asynchronous commands are only needed to output ON/OFF commands and pulses.

The RAU handler supports ON/OFF commands.

The interpreter interface routine STATE OUTPUT supports both ON/OFF commands and pulsed commands.

For ECOS, the RAU handler supports input and outputs on the serial channels.

Refer to § 5.3.4.6 and 5.3.4.7 for more precise definition.

3.2.2.4 - Pulsed commands

It will be possible to generate pulsed commands on 64 RAU outputs by sending ON/OFF commands.

The pulse width will be :

n x 10 ms ± 10 ms (n defined in the request)

A maximum of 32 outputs can be in the ON state simultaneously, for one RAU. This will be accounted for, by the RAU handler for ON and OFF asynchronous commands. In case of overstepping the calling task is warned.

These commands will be used to activate relays. We make the assumption that the relay activations will be performed in sequence :

the user program will wait until the command is transmitted,

then checks that the command is correctly executed by sensing a discrete,

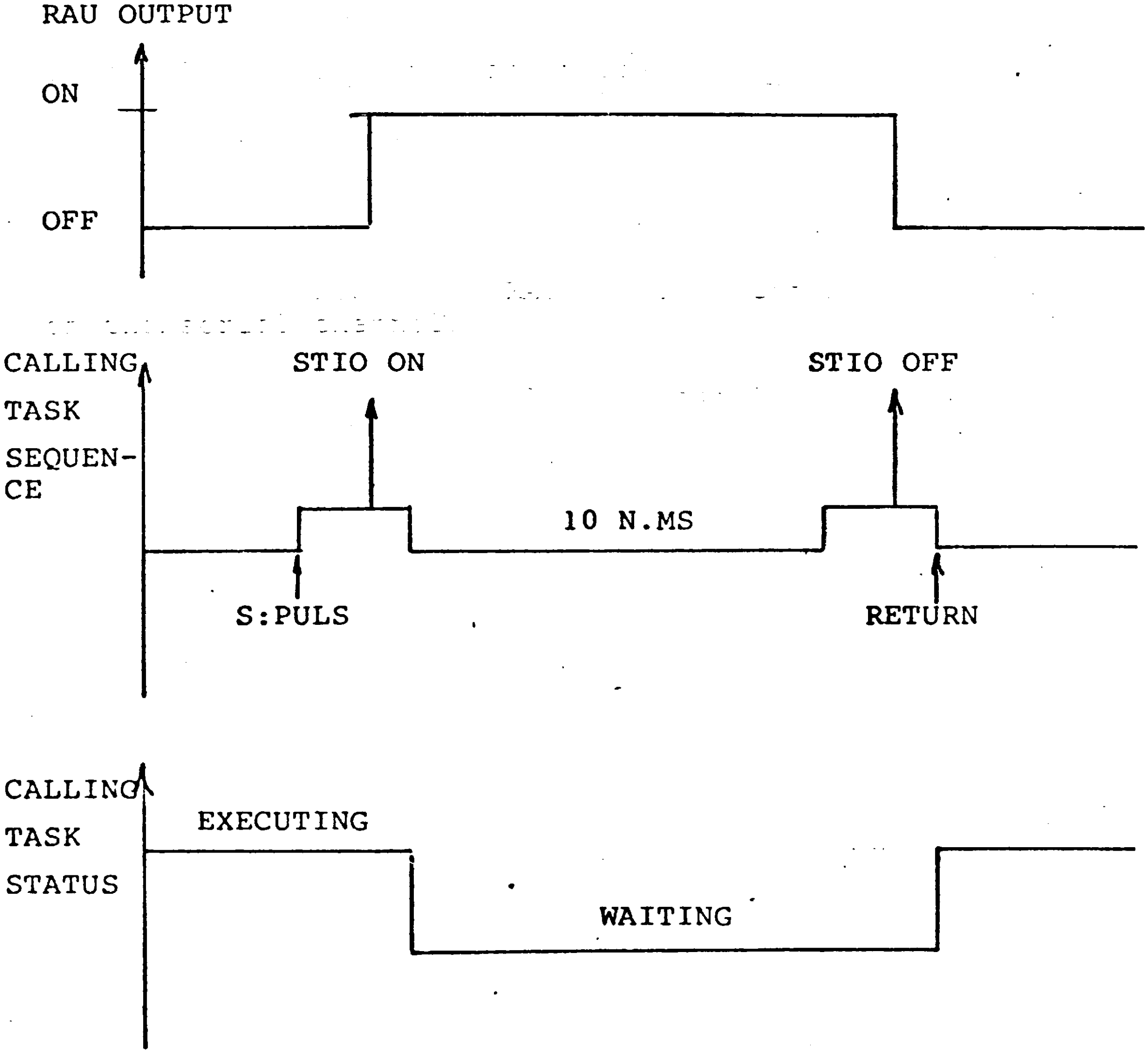
finally asks for the next command.

According to this assumption, the S:PULS routine will use the capability of issuing ON/OFF command asynchronously.

The ON/OFF commands are sent on the same RAU output with a delay of n x 10 ms between the two commands.

The requesting program will be put in a wait state during the complete pulse.

- Timing sequence



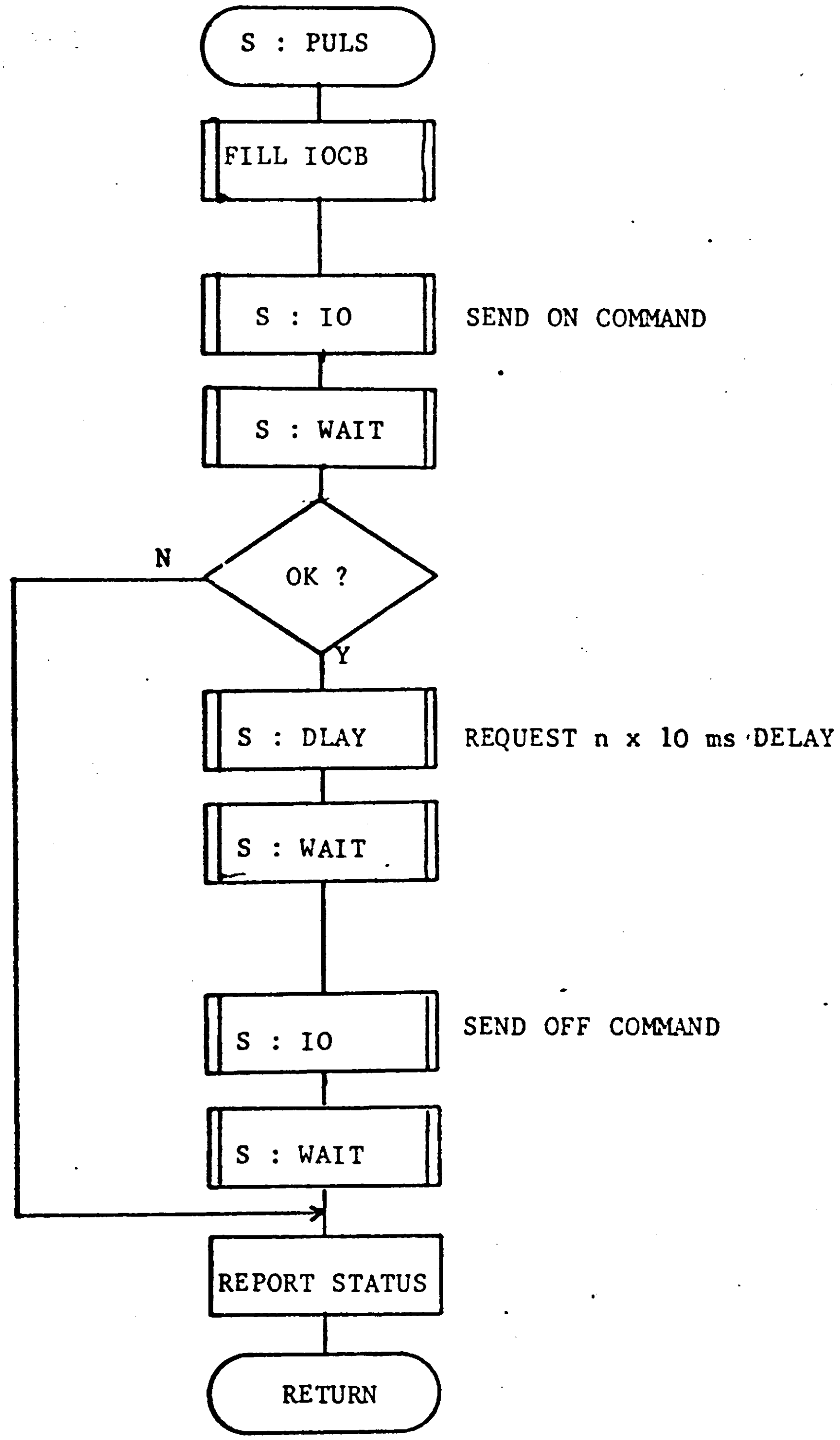
- Calling sequence

The S:PULS routine will be called with an input control block specifying :

. the RAU address

. the line address

. the pulse length.



<u>FLOWCHART FOR PULSED COMMANDS</u>

4.3.2 - Monitoring points

The BITE (Built In Test Equipment) of each equipment/ device/IOU/computer, provides an alternative mechanism for detection of failure.

The outputs of the equipment BITE's are connected to RAU inputs.

The information given by these BITE's can be :

- discrete (equipment is powered on/off, ...)
- analog (value of an internal current, ..).

These data are gathered by the GML into the GMLT where the Monitor tasks can fetch them. The Monitors treat these values for :

- limit check
- trend analysis.

In other words, detection by the Monitors is independent from the detection by the IOU : the value of a current can decrease without degradation of the performance of the device and of course without detection by the IOU BITE. All these data are telemetred and will be used as valuable material for analysis. by the EGSE and to cross correlated these variations with a possible IOU BITE interrupt, but there is no need to anticipate this event in the CDMS computers themselves.

4.3.3 - Computer trap and TBF

Another type of failure is due to a program error and is detected by the so-called "TRAP deroutement".

The origin of this failure is recorded in a TRAP Status Word for the following causes :

- end of EXEC instruction

- arithmetic operator overflow (FAO)

- program error :
  - size overflow
  - mode violation
  - memory protection
  - inexistant address
  - memory parity error
  - inexisting instruction.

The Trap Status Word is analysed by a S/W routine which must pin-point the real origin of this error.

If the error is recoverable, the S-TRAP routine will be transparent to the users (simulation of inexistant instructions).

If the error is irrecoverable but does not prevent the operating system working, then the task where the error occured will be aborted.

If the error is irrecoverable and prevent the operating system working, then the TBF suspension will be masked and the system will go to a halt after all necessary information has been stored in the self test tables. Finally the TBF light will be turned off to warn the operator.

The computer holds a micro-program called TBF which has to check that TBD% of the H/W circuits and TBD% of the core memory are working properly.

This TBF is exercised by either :

the computer Real Time Clock (RTC)

or a dedicated WD instruction.

In case a failure is detected, the TBF light is turned off and in most cases, the computer is stopped at the point where the micro-program has located the failure.

The RTC will send periodically (1024 Hz) a suspension to execute a part of the TBF micro-program. A program can start a particular sequence of the TBF with a WD instruction (e.g. for parity error isolation).

A failure detected by TBF will need a complete reloading of the computer, since the vital functions of the computer are failed.

4.3.4 - Self test task

The self test task will have additional test functions to bridge the gap left by the TBF/IOU BITE actual capabilities, such as :

- interrupts are working properly,
- GML is running,
- RTC is running,
- RAU's/MMU/PCM/GMT/MDM are working (test sequences).

When the selft test task has detected an error, it will be displayed on the DDU selftest lines.

SCOS will give the capability to the operator to display in a format acceptable to the operator the contents of the selftest tables.

The operator can also request KBIM to display values of the CDMS internal monitoring points.

Then the operator cooperation is needed to take any further action (switch to redundancies, ...).

5.2.5 - GML IO List modifications

5.2.5.1 - New measurements

There will be the capability on Ground Configurations, to add/delete MP's to/from GML IO lists for debugging purposes without the need of a complete generation of GML/CDT data.

As SCOS does not provide for dynamic memory allocation, any addition to existing IO lists will be possible only if room for those new items has been reserved at generation time.

The following assumptions are made :

- the room for triple words must have been reserved in the PCT table.

These triple words will be generated with the skipping bit set to 1 and will be executed by the RAU coupler as ineffective commands.

Some PCT triple words must be available for each sampling frequency.

- the room for the data to be read by the new commands must be reserved, in the same way, in GMLT.

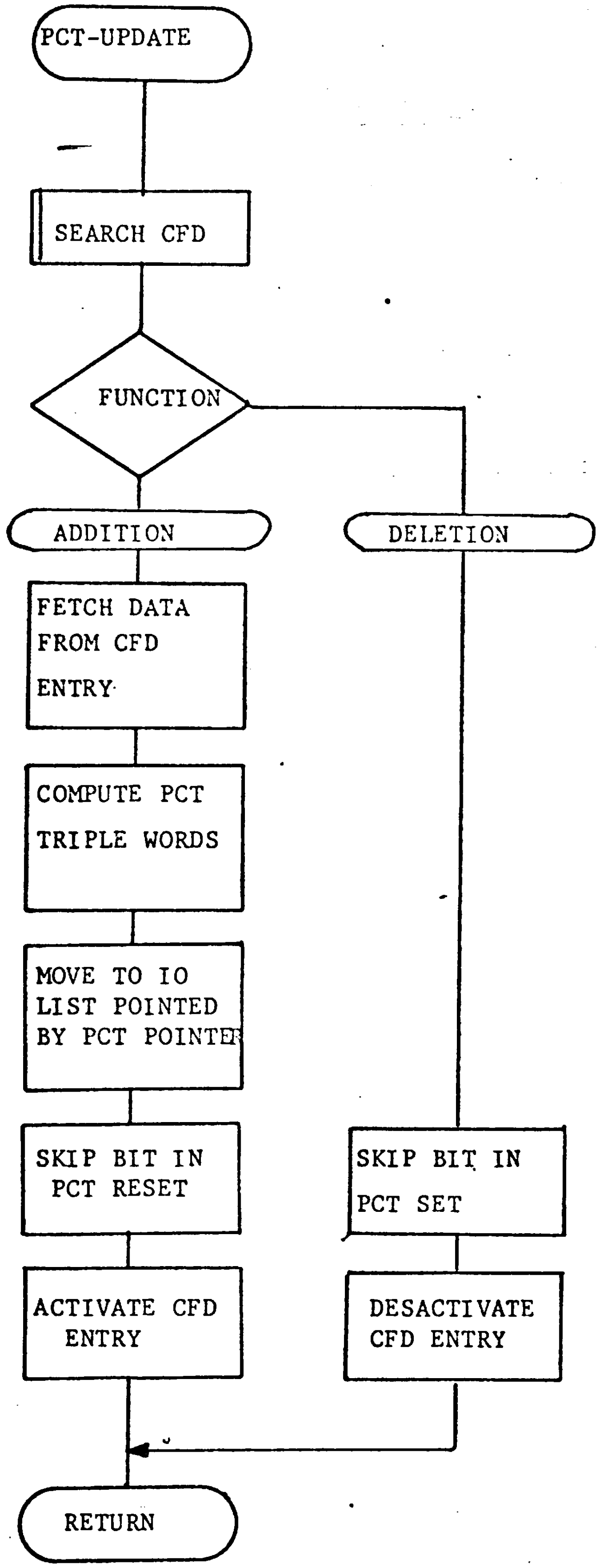
This room may be reserved at the end of the normal GMLT ; it must be coherent with the number of reserved PCT's and the acquisition mode (single mode assumed now).

According to ERNO' current design of the CDT, there is no relation between a CFD entry and the PCT triple words which actually performs the data acquisition from the bus.

In order not to increase the size of a CFD entry with a PCT pointer, this PCT pointer will be passed to the PCT-UPDATE routine by the calling program.

This solution implies that the I/O list generator will provide an output listing of all spare entries and their associated PCT pointers. These spare entries in the CDT must have their ACTIVE bit initialized in the INACTIVE state.

As the PCT-UPDATE routine cannot make a consistency check on the PCT POINTER, it is the responsability of the calling task to give a correct PCT POINTER.



IO LIST MODIFICATION FLOWCHART

If really needed, same procedure as :

Second case : sample a MP with a higher frequency. This case can be solved without additional S/W. The calling routine has :

- To delete the low frequency PCT with a CALL PCT-UPDATE (CFD, DELETE, PCT POINTER 1).

- To add a PCT entry to an IO list of higher frequency with a CALL PCT-UPDATE (CFD, ADD, PCT POINTER 2).

PCT POINTER 1 must be known for every MP (except if the change frequency procedure is known in advance). So, the I/O list generator has to print this list for every MP.

PCT POINTER 2 will be taken out from the set of spare PCT entries.

The RAU data will be read at the same GMLT location.

5.3 - INTERPRETER INTERFACE ROUTINES

The functions provided by the SCOS, to support the Interpreter, follow the requirements stated in 60-ICD-002.

5.3.1 - Interpreters interface routine list

S:SENM : To send a message over the MDM link to the GCOS tasks.

S:REDM : To request a message of a given type from the GCOS tasks via the MDM link.

S:ISER : To read data from a serial input channel.

S:OSER : To send data on a serial output channel.

S:OSTA : To send ON/OFF commands or pulsed commands.

S:IKBD : To provide the interpreter with the next input from any keyboard.

S:ØCRT : To send a line or part of a line of alphanumeric characters on a specified DDU.

5.3.2 - Interface description

The Interpreter is programmed in a subset of HAL/S. Therefore, the interfaces are described using HAL/S terminology. The HALS/S Language Specification explains the terms and provides the rules governing their use. The HAL/S-125 Manual (doc. n° TBD) will define their representation on the CII 125 MS computer. The OS is free to use HAL/S or any other language to effect the interface, so long as the machine representation which actually appears in the target computer is equivalent to the definition of the ICD according to the HAL/S-125 Manual.

The rules of HAL/S error recovery as specified in IR-61-5 Chapter 9, will be followed.