Herschel HIFI ICU OBS Software User Manual

Document Ref.: IFSI/OBS/MA/2005-001

Issue 4.5

Prepared by: Anna Maria Di Giorgio

Table of Contents

Document Status Sheet:

Document Title: Herschel HIFI ICU OBS Software User Manual			
Issue	**Revision**	**Date**	**Reason for Change**
1.0	0	14/06/2005	First Issue of the document
2.0	0	06/09/2005	Document updated to take intop account CGS, CAPTEC and SRON comments. 2 Annexes added
3.0	0	05/10/2005	Document updated to trace OBS V 3 new features. 2 Annexes eliminated
4.0	0	22/11/05	Document restructured to implement suggestions of OBS QM DRB.
4	1	11/05/2006	Document updated to implement suggestions of OBS preliminary DRB.
4	2	21/07/2006	TBW
4	3	18/12/2006	Document updated to align the manual with OBS version 4.3.
4	4	13/04/2007	Document updated to align the manual with OBS version 5.0
4	5	28/10/2009	Document updated to align the manual with OBS version 5.9

Document Change Record:

Document Title: Herschel HIFI ICU OBS Software User Manual	
Document Reference Number: IFSI/OBS/MA/2005-001	
Document Issue/Revision Number: 2.0	
Section	**Reason For Change**
All chapters	Several comments from CGS, CAPTEC and SRON have been taken into account in all sections.
ANNEX 1	Annex 1 added
ANNEX 2	Annex 2 added
Document Issue/Revision Number: 3.0	
Section	**Reason For Change**
All chapters	Changes to trace OBS V3.0
ANNEX 1	Annex 1 eliminated
ANNEX 2	Annex 2 eliminated
Document Issue/Revision Number: 4.0	
Section	**Reason For Change**
Sect. 6 + all subsections	Section reorganised to provide detailed info on the command execution onboard for each one of the implemented services. Failure detections info added.
Document Issue/Revision Number: 4.1	
Section	**Reason For Change**
Section 1.6	Applicable Doc. List updated
Section 3	Title and contents updated to trace new BSW ADs
Section 4	Section content updated to refer new Hk table
Section 6	All subsections updated to refer comments on preliminary DRB. Error info added.
Appendix A	Title changed to Appendix A2. Table with Nominal Hk ICU HK data added.
Appendix B	Title changed to Appendix A3. OBS VPF and Hifi ach content updated to trace OBS 3.6
Appendix A1	New section. OBS error codes andd execution failures.
Appendix A4	New section. Spectroscopy Measurements Flow charts.
Document Issue/Revision Number: 4.2	
Section	**Reason For Change**
Section 1.6	Applicable Doc. List updated
Section 3	Added the new section ICU POWER CYCLING
Section 5	List of ICU HK to monitor at startup
Section 7.3.2	OBS runtime errors. Table updated to trace param. field change.
Section 7.5.4	LCU configure commands reference updated
Section 7.5.10.1	Total power command consistency check updated
Appendix A1	Errors tables updated according to OBS 4.0.
Appendix A2	Table with Nominal Hk ICU HK data updated
Appendix A3.2	ICU HW parameters variability ranges updated.
Appendix A3.3	Onboard tables updated with reference to used input file.
Document Issue/Revision Number: 4.3	
Section	**Reason For Change**
Section 1.6	Applicable Doc. List updated
Section 2.5	Updated to trace OBS 4.3 OBS.MAP file
Section 4	Modified to include description of OBS runtime loading and patching.
Section 5.1	Added a section to describe the OBS short functional test.
Section 7.2.2	Non periodic HK packets section updated to include LCU non periodic hk
Section 7.2.3	IF Power calculation algorithms updated
Section 7.2.5	Diagnostic Hk section added
Section 7.4	Memory management section updated to include new DM addressable memory segments.
Section 7.5.6.3	Engineering scan section added
Section 7.5.7	Local oscillator functions section added, to trace new functions specifications.
Section 7.5.8	LCU IV curve section updated to trace new function specifications.

Section 7.5.11	Spectroscopy measurements sections updated to trace correct durations.
Section 7.7	Packet Transmission Control section written
Appendix A1	OBS error codes updated to trace errors in OBS version 4.3.
Appendix A2	OBS relevant files updated to trace errors in OBS version 4.3.
Appendix A3.3	Onboard tables updated with reference to used input file.
Document Issue/Revision Number: 4.4	
Section	**Reason For Change**
Section 1.6	Applicable Doc. List updated
Section 2.5	Updated to trace OBS 5.0 OBS.MAP file
Section 5.1	Updated to list PM checksums to use in short functional test
Section 7.1.1	Table 3 updated
Section 7.2	Table 4 updated
Section 7.2.1.1	Nominal HK rate updated. Table 5 updated. Added Table 6.
Section 7.5.3	Limit checking description updated according to new SCRs
Section 7.5.4	Updated to trace new Configure LCU commands
Section 7.5.5	Table 17 updated
Section 7.5.6.1	FCU parameter scan duration revised.
Section 7.5.6.2	Diplexers scans duration revised
Section 7.5.6.3	Engineering scan procedure description updated
Section 7.5.7	Local oscillator functions section updated
Section 7.5.8	LCU IV curve section updated to trace new function specifications.
Section 7.5.9.1/2	Tuning functions durations revised
Section 7.5.10	Calibration functions durations revised
Section 7.7	Packet Transmission Control section updated
Appendix A1	OBS error codes updated to trace errors in OBS version 5.0
Appendix A2	OBS relevant files updated to trace errors in OBS version 5.0
Appendix A3.3	Onboard tables updated with reference to used input file.
Document Issue/Revision Number: 4.5	
Section	**Reason For Change**
Section 1.6	Applicable Doc. List updated
Section 2.5	Updated to trace OBS 5.0 OBS.MAP file
Section 5.1	PM Checksums' list updated.
Section 7.5.12	Peakup section added
Appendix A5	Onboard used SIDs table added

1 Introduction

1.1 Intended readership

This document should be read by anybody who has to operate the HIFI ICU. At first, the ICU will be tested and operated in standalone configuration. After the integration with the other subsystems, the only way to operate the HIFI instrument will be sending telecommands to the ICU and receiving telemetry packets from it. Anybody involved in HIFI operations is thus a potential reader of this document.

1.2 Applicabiliy Statement

This document describes how to use the OBS run by the ICU inside the HIFI instrument.
This document doesn't apply to the HIFI ICU Boot Software (BSW), even if some references on how to interpret the messages provided by the BSW during the ICU start up will be made.

1.3 Purpose

1.3.1 Purpose of the document

This document describes in detail the procedures to start-up and run the HIFI OBS, the contents of the TC packets to be uplinked in order to perform the required function, and the contents of the TM packets that the OBS generates.

1.4 Definitions, acronyms and abbreviations

1.4.1 Acronyms

APID	APplication Identifier
BC	Bus Controller
BSW	Boot SoftWare
CDMS	Command and Data Management Subsystem
CNR	Consiglio Nazionale delle Ricerche
CPU	Control Processing Unit
CRC	Cyclic Redundancy Check
DM	Data Memory
DPU	Digital Processing Unit
EEPROM	Electrically Erasable Programmable Read Only Memory
FCU	FPU Control Unit
FIFO	First In First Out
HIFI	Heterodyne Instrument for FIRST
HK	HouseKeeping
HRS	High Resolution Spectrometer
HS	High Speed
ICU	Instrument Control Unit
IF	Intermediate Frequency
IFSI	Istituto di Fisica dello Spazio Interplanetario
ISR	Interrupt Service Routine
LCU	Local oscillator Control Unit
LS	Loe Speed
OBS	On Board Software

PM	Program Memory
PROM	Programmable Read Only Memory
RAM	Random Access Memory
ROM	Read Only Memory
RT	Remote Terminal
SRON	Space Research Organization Netherlands
SUM	Software User Manual
TBC	To Be Confirmed
TBD	To Be Defined
TBW	To Be Written
TC	TeleCommand
TM	TeleMetry
UR	User Requirement
URD	UR Document
VM	Virtual Machine
WBS	Wide Band Spectrometer

1.4.2 Abbreviations

ID	Identification
I/F	Interface
SW	SoftWare
HW	HardWare
S/C	SpaceCraft
S/S	Sub-System

1.5 Conventions

- `Text in this font` is a message reported on the screen, for instance a SCOS2000 error/event message;
- A number starting with "0x" is an hexadecimal number. To indicate a generic number letters from `a` to `f` are not used;
- All the TC and TM packets have a checksum at their end, computed on the whole packet content. This checksum is always indicated with CRC. In some cases, when the packet transport memory data, an additional checksum is computed on these data only, and is indicated with crc;
- *P# i* is the i-th parameter, for instance of a TC; this font is also used to identify the value of a parameter, e.g. the length of the packet is Length;

1.6 References

1.6.1 Applicable Documents

Doc. Ref.	Name	Number/version/date
AD1	HIFI ICU OBS User Requirements Document	IFSI/OBS/SP/2000-001 Issue 1.5 – 15/04/2005
AD2	HIFI ICU OBS Software Specifications Document	IFSI/OBS/SP/2002-001 Issue 2.2 – 10/05/2006
AD3	HIFI TC Packet ICD	SRON-U/HIFI/SP/2001-1 Issue 1.8 – 23/10/2006
AD4	HIFI TM Packet ICD	SRON-U/HIFI/SP/2001-2 Issue 1.7 19/04/2006
AD5	HIFI HK Packet ICD	SRON-U/HIFI/SP/2001-3

		Issue 1.11 26/10/2006
AD6	HIFI Command Specification	SRON-U/HIFI-SP-2001-4 Issue 1.10
AD7	HIFI Internal Databusses ICD	SRON-U/HIFI-SP-2001-10 Issue 1.5
AD8	Packet Structure Interface Control Document	SCI-PT-ICD-7527 Issue 5.0
AD9	Herschel/Planck Instrument Data Rates	H-P-1-ASPI-TN-0204 Issue 1.0 – 15/01/2002
AD10	HERSCHEL BOOT SOFTWARE SPECIFICATION DOCUMENT	HERS-SQ-CGS-002 Issue 1.0 – 10/07/2005
AD11	HERSCHEL DPUs/ICU BOOT SW TELEMETRY / TELECOMMANDS PACKETS USER MANUAL	HERS-GEN-MA-CGS-001 Issue 1.0 – 12/07/2005
AD12	IMPACT of LCU SW changes on EGSE and OBS	SRON-U/HIFI-TN-2006-5 Issue 1.4

1.6.2 Reference Documents

Doc. Ref.	Name	Number/version
RD1	HIFI-ICU Virtual Machine	IFSI/ICU/TN/2002-004
RD2	DPU-BSW Software Requirement Document	DPU-SQ-CGS-001
RD3	VIRTUOSO User Guide	
RD4	ADSP-21000 Family C Tools Manual	
RD5	ACTEL Alert Note- RT54SX32S High IccI Inrush Current	
RD6	Herschel CFM HIFI ICU Electrical Interface And Functional Performance Test Procedure	HERS-HIFI-PR-CGS-001 Issue 1.0 01/08/2005

2 OBS Compilation

This section describes the basic components that must be available to compile the OBS and the procedure to do it.

2.1 External Components

In order to be able to recompile the OBS two components must be installed on a Windows machine:

- ADSP- C Compiler and Tools, Release 3.3, Version 2.21 (see RD4)
- VIRTUOSO Real-Time Software Development Tool, v4.1, R2.04 (see RD3).

2.2 The VIRTUOSO Project File

The **OBS.vpf** file contains the settings of the VIRTUOSO services that are used in the OBS. It can either be edited under VIRTUOSO, or with any text editor. This is where objects like Tasks, Semaphores, FIFO services, Events, Timers are defined. Refer to RD3 for a detailed description of the various services used. The current content of the project file for OBS version 4.3 is part of the OBS distribution and is reported in Appendix A2.

2.3 The Architecture File

The **hifi.ach** file contains the definition of the various segments of the ICU PM and DM. In Appendix A2 the current content of the architecture file for the version 4.3 of the HIFI OBS, that is part of the OBS distribution, is reported. Refer to RD4 for a detailed description of the various segments and directives used in creating this file.

2.4 Compiling the OBS

If the Virtuoso OS has been installed in the directory c:\Virtuoso, the OBS source files shall be copied into the following directory:
1) C:\Virtuoso\ADI21020\Rev33\Sigma\MyProj\OBS.
The OBS distribution contains a **makefile** that manages the compilation and linking of the source code in the hypothesis that the relative path of the source code with respect to the Virtuoso home directory is the same as the one reported in point 1.
Typing make on the command line will compile all source files that have been updated with respect to previous compilation, or that depend on include files that have been modified; make rebuild will recompile all C and Assembler source code files.
Any compilation subsequent to a modification of the VIRTUOSO Project File (e.g. after adding another semaphore) will need a valid VIRTUOSO license.

2.5 The Compilation Products

The compilation will produce many intermediate files. The .o object files whose name starts with the suffix MIL should never be deleted since they contain the compiled MIL-1553B-STD drivers whose source codes are not included in the OBS delivery.
The most important compilation product is obviously the **OBS.EXE** that will contain the executable code.
Another useful ouput file is the memory map file that documents the actual ICU memory usage by the OBS. Here is an extract from the **OBS.MAP** file contained in the OBS distribution and valid for the OBS version 5.0.

```
Analog Devices ADSP-210x0 Linker            OBS.map                  Page 1
Release 3.3, Version 2.21                   Fri Apr 13 00:34:54 2007
Copyright (c) 1991-1996 Analog Devices, Inc.


Architecture Description: FirstDPU

Segment     Start     End       Length      Memory Type          Attribute    Width

seg_rth     000000    0000ff    256         Program Memory       RAM
seg_dmda    00000000  00027fff  163840      Data Memory          RAM
seg_init    004000    005fff    8192        Program Memory       RAM
seg_pmco    006000    06ffff    434176      Program Memory       RAM
seg_stak    00028000  000283ff  1024        Data Memory          RAM
heap1       00028400  0007ffff  359424      Data Memory          RAM
seg_pmda    070000    07ffff    65536       Program Memory       RAM
EEPROM      80000000  8003ffff  262144      Data Memory          RAM
Timer       81000000  81ffffff  16777216    Data Memory          PORT
watchdog    82000000  82ffffff  16777216    Data Memory          PORT
Int_mng     83000000  83ffffff  16777216    Data Memory          PORT
SMCS_reg    84000000  84ffffff  16777216    Data Memory          RAM
Bus_IF      88000000  8fffffff  134217728   Data Memory          RAM


Memory Usage (Actual):

Segment     Start     End       Length      Memory Type          Attribute

seg_rth     000000    0000ff    256         Program Memory       RAM
seg_init    004000    00400e    15          Program Memory       RAM
seg_pmco    006000    018aef    76528       Program Memory       RAM
seg_pmda    070000    070001    2           Program Memory       RAM
seg_dmda    00000000  00020de3  134628      Data Memory          RAM
seg_stak    00028000  0002800a  11          Data Memory          RAM
heap1       ********  ********  0           Data Memory          RAM
EEPROM      ********  ********  0           Data Memory          RAM
Timer       ********  ********  0           Data Memory          PORT
watchdog    ********  ********  0           Data Memory          PORT
Int_mng     ********  ********  0           Data Memory          PORT
SMCS_reg    ********  ********  0           Data Memory          RAM
Bus_IF      ********  ********  0           Data Memory          RAM


Memory Usage Summaries:

Memory Type     Attribute  Total

Program Memory  ROM        0
Program Memory  RAM        76801
Program Memory  PORT       0
Data Memory     ROM        0
Data Memory     RAM        134639
Data Memory     PORT       0
```

3 ICU POWER CYCLING

A problem related to a high Inrush Current in the FPGA chip mounted onboard ICU FM and CFM is described in the Alert Note from Actel referred as RD5 .
Due to the problem, the ACTEL suggested to wait a certain amount of seconds between the ICU power off and the following power on.
The suggested wait times are reported in table 2 page 3 of the ACTEL note (RD5).
The table is reported hereafter:

Temp(°C)	Time (min)
-30	371
0	45
25	4.3

40	1.2
65	0.183

The time can be very long at low temperatures.

The correct power on procedure is described in section 6.1 of the HERS_HIFI_PR_CGS_001 document (RD6) and implies to wait at least 5 minutes between the ICU power off and the following power on at room temperature (25 °C).

4 ICU Switch on and OBS Loading

When the ICU is switched on, the BSW is copied from PROM to PM and run.
The details of the boot procedure can be found elsewhere (see AD10, AD11); here we simply note that after all the tests are carried out, a (5,1) event is generated and the boot enters an infinite loop waiting for a TC.
The contents of the generated event are described in AD11.
In case of reception of a successful event, there are two possibilities for loading and executing the OBS:

1. using the image resident on the EEPROM on-board, see section 4.1
2. loading a new image via standard TCs, see section 4.2.

4.1 Running the EEPROM-resident OBS

The OBS is resident in EEPROM. Once the (5,1) event (with no errors reported) is received, the command **"HIFI_Force_bootpartition"** described in AD3, section 4.3.14.2, can be sent to the ICU.
Upon receipt of this TC, the BSW copies the OBS from the EEPROM partition specified in the TC to PM and jumps to the start location of the OBS in PM; at this point the OBS starts running.
If the ICU is connected to the CDMS simulator or SCOS2000, HK packets will be received (SID 0x404 for nominal HK packets and SID 0x01 for Essential HK packets). This can be considered as the confirmation that the startup procedure has been successfully completed. See section 5 for a complete descrition of the power on procedure.

4.2 Loading the OBS via telecommands with BSW

In the following the procedure for uploading the OBS in case of testing activities on ground is described. The actual procedure to be used during flight operations is still to be defined and will imply the use of the Software management Facility provided by ESA.

Once the BSW puts the ICU in a wait state, it is possible to uplink from SCOS2000 a new image of the OBS using standard TCs.
Three steps are necessary to perform the full operation:

1. It is necessary to prepare the Telecommands for uploading the new OBS image: the C program **TCGen** provided by CGS is available under Windows to translate the OBS image into a list of TC (6,2) ready to be sent to the ICU. The ADI21020 C Compiler must also be installed, since TCGen uses some C-tools (like cdump).
   The command to invoke the procedure is:

   >tcgen –i ***segfile.txt*** –p ***pagefile.txt*** –f ***path/OBS.EXE*** –a 0x400 –o ***path/suffix*** –m 0 > ***name output directory***

   where:
   - The *segfile.txt* file contains the list of memory segments (one per line) defined in the ICU program memory and reported in the architecture file hifi.ach; typically the segments are seg_rth, seg_init and seg_pmco. The segfile.txt file used for generating the OBS version 3.2 TC set is reported in Appendix A
   - The *pagefile.txt* file contains the list of memory pages to be avoided (it can be empty).

- *OBS.EXE* is the executable file as produced by the compilation of the OBS code.
- *Path* is the directory where the output TCs will be stored ;
- *Suffix* is a string that will be attached to the TC file names: the output files will be named *suffix*TCnnnnn.dm where nnnnn is a count number.

2. It is necessary to prepare the CDMS for the commands reception. On the PC hosting the CDMS simulator:
   a. Click on icon CDMS_SIM
   b. On the "Select Buslist" button, select the HIFI_Nominal buslist
   c. Click on Launch Router Command Interfac: set the IP address and the port number:
      IP address = <address of the router machine>
      Port: 9877
   d. Click on Connect
   e. On the "Select Command to send" option select NAME_CLIENT; write "hifi" and click Send Command
   f. On the "Select Command to send" option select ADD_CLIENT, write "400" and click Send Command
   g. Click on Close Without Sending
   h. Click on Start/Stop BC button

   At this point the CDMS is ready to accept the TCs.

3. The set of TCs containing the image of the can be uplinked using the "**ObswLoader**" script. The script loads TCs from a local directory on the Router machine and sends them to the CDMS that, in turn, sends them to the ICU. The following syntax should be used to invoke the script.

   >ObswLoader –dpu –apid 1024 –interval ***XXX path***/*Tc*.dm

   where ***path*** is the directory that hosts the telecommands prepared with the TCGen program, and *XXX* is the interval in milliseconds for the dispatch of subsequent TCs to the CDMS. Clearly, the dispatching interval should match the capabilities of the buslist currently running on the CDMS. For fast uploads a dedicated buslist has been prepared that allows the CDMS to send to the ICU a maximum of 20 TC/s; using this buslist allows to invoke the ObswLoader script with an interval parameter of 50 (milliseconds). If one uses the nominal buslist where only 2 TC/s can be uplinked, then the interval parameter should be set to 500.

Once all TCs have been sent, it will be necessary to send the **"Load TC and boot"** TC (see AD3, section 4.3.14.3) from SCOS2000 to command the BSW to copy the full image from DM to PM and start the application program.

To copy the loaded OBS into the onboard EEPROM an OBS dedicated TC shall be issued. This TC is described in and AD3, section 4.3.14.4. The overall EEPROM write procedure lasts for <=20secs.

4.3 Loading and patching the OBS via telecommands at runtime

A new version of the OBS can be uploaded and started also while an older version of the OBS is running. This can be made in two ways: by **loading** a new version of the OBS or by **patching** a previously existing OBS version. In the loading procedure the entire new OBS code is passed to the existing OBS version via a TC sequence. In the patching procedure only the difference between the currently running version and the new version of the OBS is loaded via a TC sequence.

In the following, after a general presentation of how the procedures work, we summarise the steps to follow to realise loading and patching. The way how the steps are implemented also depends on the auxiliary packages and on the operating environment. Therefore we also present indications on how the steps can be implemented in two different operating environments.

4.3.1 OBS loading

The OBS code is stored in the ICU PM, starting at address 0. It does not occupy the whole PM. More precisely the OBS up to version 4.3 occupies far less than 0x20000 PM words while the PM is 0x7ffff words long. Therefore a new OBS copy can be stored and modified directly in the high part of the PM without overwriting the running OBS, which is stored in the low part of the PM.

In order to load the new OBS the user has to prepare a sequence of *HIFI_load_PRAM* TCs to command the running OBS to build the new version of the OBS in the high PM, starting at a specified address (high_PM_address). The high_PM_address should be high enough that the running OBS is not corrupted by the copy of the new one: for this purpose, address 0x3ffff is compatible with any OBS version.

Once the OBS copy has been constructed in the high PM, the user should issue the command *HIFI_copy_mem_to_low*. This command has two parameters: OBS_destination and OBS_copy_len. When this command is received, the OBS takes the following actions:

1. it copies the procedure that handles the TC into the 100 PM words preceeding address OBS_destination (necessarily equal to the high_PM_address used in the previous step) and passes the control to this code;
2. the procedure copies OBS_copy_len PM words starting from address OBS_destination (high PM) into a PM area of the same length starting at address 0 (low PM);
3. when the copy is complete, the procedure invokes an *HIFI_reset* command.

As a result, if a copy of the OBS was prepared in the high PM starting at address high_PM_address, when this command is issued with OBS_destination = high_PM_address and provided that OBS_copy_len is higher than the number of words of the OBS version stored in high PM (a value of 0x20000 will always do) the copy of the new OBS is moved from the high PM to the low PM and re-started.

It is higly recommended to check that the OBS copy has been correctly constructed in the high PM before issuing the *HIFI_copy_mem_to_low* command. This can be done by issuing a proper sequence of *HIFI_check_memory* and *HIFI_check_PM_memory* commands.

Therefore we can summarise the steps for performing the OBS **loading** procedure as follows:

L1. Prepare a sequence of *HIFI_load_PRAM* TCs that constructs the new OBS version in the ICU high PM.

L2. While the OBS is running on the ICU, issue the sequence of TCs produced at step L1.

L3. Issue a sequence of *check_memory* commands in order to be sure that the copy has been correctly constructed. Abort the overall procedure in case the copy is damaged.

L4. Issue the *HIFI_copy_mem_to_low* TC.

4.3.2 OBS patching

The procedure for the OBS patching is slightly different. The difference is in how the new version of the OBS is constructed in the high PM. While in the loading procedure the user has to prepare a sequence of TCs that construct in high PM a whole copy of the OBS, in the patching procedure the user firstly makes a copy of the running OBS version in the high PM and next issues to the OBS a sequence of TCs that changes that copy into the newer version, i.e. the TCs have to carry only the differences between the running copy and the new copy.

The copy of the running version can be constructed by issuing the *HIFI_copy_mem_to_high* TC. This command is very similar to the HIFI_copy_mem_to_low command except that it works in the other direction, i.e. it copies from the low PM to the high PM: indeed they are both implemented by means of the *HIFI_copy_memory* command by specifying a different direction (1 for low to high and 2 for high to low).
In addition, this command does not perform a reset at completion.
The command has two parameters: OBS_destination and OBS_copy_len.

When the command is issued, the OBS will copy a block of OBS_copy_len PM words starting from address zero into a block of the same length starting at address OBS_destination. In this way, provided that OBS_copy_len is high enough (a value of 0x20000 will always do) the running OBS is copied into the high PM.

We can summarise the steps for performing the OBS **pacthing** procedure as follows:

P1. Prepare a sequence of *HIFI_load_PRAM* TCs that turn the current OBS into a new OBS version.

P2. While the OBS is running on the ICU, issue the TC *HIFI_copy_mem_to_high*.

P3. Issue the sequence of TCs produced at step P1.

P4. Issue sequence of *check memory* commands in order to be sure that the copy has been correctly constructed. Abort the procedure in case the copy is damaged.

P5. Issue the TC *HIFI_copy_mem_to_low*.

4.3.2.1 Example 1: Implementing a patching at the IFSI premises.

To clarify the procedure and illustrate how the procedures has to be customized for the existing operating environment we report here a detailed description on how the OBS patching is performed and tested in the operating environment used at the IFSI premises.
In the example, we will pacth OBS release X into OBS release Y.
We assume that the high_PM_address is 0x3ffff.

Implementation of step P1.
At IFSI the OBS Management facility is not available. Therefore the sequence of TCs is prepared by using a C program internally developed. The program is called *genera_TC_patch* and can be compiled and executed on any PC. It requires two input files containing the ascii dump of the two versions of the OBS executable. Both files can be produced by the ADI21020 compiler using the CDMP directive. The *genera_TC_patch* program by default assumes that the high_PM_address is 0x30000. In case the user wants to specify a different addres (like we want in this example) the program accepts an option in the form –o offset where offset is a HEX offset with respect to the default address: therefore, in order to provide the TC sequence for patching at address ox3ffff we need to invoke the program with a –0 oxffff option. Once the two input files are stored in the same directory where the program is located, the program can be executed by means of the following command

>genera_TC_patch p OBS_X.fil -n OBS_Y.fil –o 0xffff

The program output is a sequence of *HIFI_load_PRAM* commands that are stored in the sub-directory TC_for_patching. The number of provided TC packets depends on how much OBS.Y is different from OBS.X. If the two differ in one data only (e,g, the initial value of a global value or a table entry) a couple of TCs may be sufficient. If the two are very different the same number of packets as a normal full load could be required. The program outputs (on the screen) also the length, the starting address and the checksum of the three segments into which the code is divided, which are the seg_rth, the seg_init and the seg_pmco. These data are useful to contruct the check commands to be used in step P4.
In the following an example of the output of the program is provided. It has been obtainedpreparong the patches for the versions 4.3.1 ontop of 4.3.

```
>genera_TC_patch.exe -p OBS4_3.fil -n OBS4_3_1.fil -o 0xffff

Opening OBS4_3.fil
        Time of image compilation : Tue Nov 14 11:28:24 2006
        Found segment seg_rth
                Start address 0
                Size 0x100
        Found segment seg_init
                Start address 0x4000
                Size 0x11a7
        Found segment seg_pmco
```

```
                Start address 0x6000
                Size 0x11eae
        Writing temp file
                Processing section: 'seg_rth'
                Processing section: 'seg_init'
                Processing section: 'seg_pmco'
Opening OBS4_3_1.fil
        Time of image compilation : Tue Nov 14 16:17:54 2006
        Found segment seg_rth
                Start address 0
                Size 0x100
        Found segment seg_init
                Start address 0x4000
                Size 0x11a7
        Found segment seg_pmco
                Start address 0x6000
                Size 0x11eae
        Writing temp file
                Processing section: 'seg_rth'
                Processing section: 'seg_init'
                Processing section: 'seg_pmco'
Processing 'seg_rth'
                 crc on the segment: 0xc65e
Processing 'seg_init'
                 crc on the segment: 0x4987
Processing 'seg_pmco'
                 crc on the segment: 0xbf2e
```

Implementation of step P2.
Once OBS.X is running the TC *HIFI_copy_mem_to_high* is issued. The parameters of this command are OBS_destination = 0x3ffff, OBS_copy_len = 0x20000. The second parameter has to be at least equal to the actual length (in number of PM words) of the OBS.X code. The value 0x20000 is a safe value, since it is much higher than any allowed OBS version length (the maximum allowed length is defined by half the EEPROM size).
The execution of the *HIFI_copy_mem_to_high* takes a few seconds and is terminated when a command execution success TM packet (1,7) is received.

Implementation of step P3.
To send the patching TCs an automatic procedure is needed. At IFSI premises this can be achieved by running the *ObswLoader* procedure on the machine where the router is running (i.e. the SCOS2000 machine). The procedure is the same used to upload the TCs when the BSW is running:

>ObswLoader -apid 1024 -dpu -patch -interval 250 *path*/*.tc

The options have the following meaning:
-apid 1024: it instructs the OBS loader to generate packets with the HIFI apid.
-dpu: it tells the OBS loader that the commands are directed towards the ICU.
-patch: it is needed to specify to the OBS loader that the word dimension is 48 bits (that of the PRAM) and not 32 bits (that of the DRAM).
-interval 250: it sets the time interval between to successive command issues to 250 msec which is the maximum tolerable rate for the OBS.
The last parameter is the full path of the files to upload.

After issuing the command just described the SCOS machine will pass the sequence of commands to the CDMS which turns the commands to the OBS: on the CDMS simulator the sequence of commands being issued shall be seen. No TC acceptance reports will be generated because the ObswLoader procedure sets the corresponding bits in the TCs to 0. If there are problems in the ingestions of the TCs, TC execution failure reports (1,8) are generated. In this case the upload procedure shall be stopped and repeated from the beginning.
The time necessary to run the procedure depends on how many patching TC packets are needed and on selected time interval.

Implementation of step P4.

This step can be done by preparing two TCs. One is a *HIFI_check_memory* command that should be used to check the segment seg_init. The data to construct this command are produced by the program genera_TC_patch (see the example above). For this segment the program reports an init_offset, a init_len and a init_checksum. The user shall prepare a *HIFI_check_memory* command with the following parameters: OBS_mem_id = 0 (specifies the PM) OBS_mem_ start = 0x3ffff + init_offset (tells where the area to check starts) OBS_mem_length = init_len. After issuing this TC the user should check the relative report and verify that the reported crc is identical to the one produced by genera_TC_patch (i.e. to init_checksum). The second command is used to check the contents of the seg_pmco segment. Again the command is constructed on the data produced the the genera_TC_patch program which reports the segment offset (pmco_offset) len (pmco_len) and checksum (pmco_checksum). The user shall prepare an *HIFI_check_PM_memory* command with the following parameters: HIF_check_start = 0x3ffff + pmco_offset (tells where to start the check) HIF_check_end = 0x3ffff + pmco_offset + pmco_len –1 (tells the last word to check) and HIF_chck_crc = pmco_checksum (tells the expected checksum). After issuing the command the user should receive an execution succes TM packet (1,7) confirming that the expected checksum was indeed obtained. If an execution failure (1,8) TM packet is obtained the procedure should be aborted.

Implementation of step P5.
Issue the TC *HIFI_copy_mem_to_low* TC. The parameters of this command are OBS_destination = 0x3ffff, OBS_copy_len = 0x20000. The second parameter has to be at least equal to the actual length (in number of PM words) of the OBS.X code. The value 0x20000 is a safe value, since it is much higher than any OBS version length.
After this command has been issued the new copy of the OBS (i.e. OBS.Y) should be loaded and started. On the AVM1 at the IFSI premises a manual reset of the ICU is additionally needed due to a known problem in the ICU reset.

4.3.2.2 Example 2: Implementing patching at SRON.

We now report a description on how the patching could be realised with the OBSW Managment facility.
The two environment dependent steps are P1 and P3 and are discussed in the following. The other steps are identical.

Implementation of step P1.
When the OBSM is available it can be used to produce the TC patching sequence from a version of the OBS.X and of the OBS.Y executables in the ICD14 format.

Implementation of step P3.
The OBSM can be used to issue the sequence of TCs produced at step P1.

5 HIFI subsystems power on procedure

At switch on, the Boot Sofware executes a series of memory tests. See AD10 for a detailed description of the procedure and of the TM packets provided by BSW. In case of successful tests, a 95,1) packet is provided, indicating that the system is ready to boot.
A HIFI_Force_boot TC shall be issued. This command will cause a jump to the starting location of the OBS which will start running. Before any other operation onboard, the HIFI_Goto_Safe procedure is executed (according to what requested in SCR 818). The procedure is executed without checking if the subsystems are switched on.
The Low Speed commands issued by ICU to implement the procedure are:

Table 1 Goto_safe_commands

0xF0020202	HL_standby
0xCF010000	HF_band_0
0xFF900000	HR_stop
0xE4000011	HWH_laser_1_off
0xE4000012	HWH_laser_2_off
0xE8000011	HWV_laser_1_off
0xE8000012	HWV_laser_2_off
0xE400002A	WBS_H set zero switch

0xE400000C	WBS_H set COMB off
0xE41FFFEF	WBS_H set att to max value
0xE800002A	WBS_V set zero switch
0xE800000C	WBS_V set COMB off
0xE81FFFEF	WBS_V set att to max value
0xFC000005	WBS_stop
set HRS_H attenuation to max	
0xD590001F	block 8
0xD5B0001F	block 7
0xD5D0001F	block 6
0xD5F0001F	block 5
0xD5E0001F	block 4
0xD5C0001F	block 3
0xD5A0001F	block 2
0xD580001F	block 1
set HRS_V attenuation to max	
0xD990001F	block 8
0xD9B0001F	block 7
0xD9D0001F	block 6
0xD9F0001F	block 5
0xD9E0001F	block 4
0xD9C0001F	block 3
0xD9A0001F	block 2
0xD980001F	block 1
0xF00A0FF0	LOU heater to 6 V

The table containing the command is Cmd_safe_002 and is provided by SRON.

Once started, the OBS sends the nominal HK packets (refer to section 6.2.1 of this document for an overall description of the HK packet organisation) at a rate of 1 packet every 3 seconds.
The other S/S are switched off, so that the ICU functions are: communication with the spacecraft, via the 1553 interface, sampling of its HW HK and their check, shipping of the HK packets.

The OBS will be ready to accept any other TC only 1 sec after the reception of the HIFI_Force_Boot (or HIFI_Load_Boot) TC.

5.1 ICU OBS Short Functional Test

The purpose of the short functional test is to check the the ICU has started up successfully, if it is properly working and properly interfaced with the spacecraft.

1. Power on ICU and check that the correct TM packet(s) received from BSW.
2. Boot from EEPROM image (see section 4.1) and check the following:
   - HK packets received;
   - ICU HK values are nominal (see section 5.2 for the ICU/OBS HK reference values at startup);
3. Send HIFI_Check_PM_memory to verify memory content (the expected PM checksum is notified in the OBS release Note); see Table 2 for the TC parameters.
4. Send TC HIFI_Housekeeping_on 1_per_sec (see section and verify HK rate changes, and correct reporting of OBSID and BBID;
5. Send TC HIFI_Connection_Test and check reception of HIFI_Connection_report, see this document section 0;
6. Send TC HIFI_Enable_Time_Verify and check reception of a HIFI_Time_verification_report, see section 7.7;

7. Send TC HIFI_Simulate_Science: upon reception of this TC the OBS shall start generating Dummy science data for all the 4 spectrometers. See this document, section 7.6.14.6, to have more details about the xpected data.
8. Send TC HIFI_abort_spectroscopy to stop the dummy science data generation and check that the OBS provides a (1,8) TM packet, to signal that the current activity onboard has been aborted. The science data provision shall stop.

If the SFT has been carried out successfully, the OBS is supposed to be ready to start the other subsystems switch on procedure (see section 5.3).

Table 2 HIFI_Check_PM_memory TC parameters

OBS Version	HIF_check_start	Hif_check_end	Hif_check_CRC
4.3.2	6000	17eb6	2E60
4.3.3	6000	182a9	AAE5
4.4	6000	1827c	6B58
4.4.1	6000	18341	80A2
4.4.2	6000	18341	
4.4.3	6000	1838c	558E
5.0	6000	18aef	1490
5.1	6000	18a47	613C
5.1.1	6000	18a47	E8B3
5.1.2	6000	18a2b	595B
5.1.3	6000	189db	DBB0
5.2	6000	18a2a	A11
5.3	5500	17ff1	0d32
5.4	5500	17d0b	31a0
5.5	5500	17d0b	2fac

5.2 ICU HK reference values at startup

The following values are reported in the periodic HK packet in case of switched off subsystem:
LCU: 0x00AA
FCU: 0x00BB
HRSH/V: 0x00000000
WBSH/V: 0x0000

Other ICU HK values to monitor during startup are reported below, extracted from the complete ICU HK table contained in Appendix A3.1.

Table 3 List of ICU HK parameters

32 bits word idx	Start Byte	Start bit	Length	Monitor Parameter Description	Acceptance criterium
0	26	0	24	HI_SW_Version Version number of the OBS	= to the version of the OBS under test
0	29	0	8	HI_SW_Revision Revision number of the OBS	= to the revision of the OBS under test
1	30	0	32	HI_IDLE Number of loops in a second performed by the res_chk task (the lowest priority task).	(>=1200*HK acquisition rate(sec) at startup)
3	38	0	32	HI_CPU_Load_AV = Average percent CPU load.	<10 at startup
5	46	0	32	HI_EV_POOL Max # of blocks taken in Event Pool	<28 (= 0 at startup)
6	50	0	32	HI_HK_POOL Max # of blocks taken in HK Pool	<22 (= 4 at startup)
7	54	0	32	HI_SD_POOL Max # of blocks taken in Science Pool	<30 (= 8 at startup)
8	58	0	32	HI_TC_POOL Max # of blocks taken in TC Pool	<6 (= 0 at startup)
15	86	0	32	HI_VM_RUNNING_S	True if VM is running. False if it is stopped . (= 0 at startup).
16	90	0	32	HI_2P5_V 2.5 Volt actual value	N/A to AVM1 For FM see A3.2
17	94	0	32	HI_5P_V 5 Volt actual value	N/A to AVM1 For FM see A3.2

18	98	0	32	HI_15P_V 15 Volt actual value	N/A to AVM1 For FM see A3.2
19	102	0	32	HI_15M_V minus 15 Volt actual value	N/A to AVM1 For FM see A3.2
20	106	0	32	HI_CPU_T CPU Temperature	N/A to AVM1 For FM see A3.2
21	110	0	32	HI_SUBSYSTEM_S Current Subsystem Status	Equal to the commanded Subsystem Status Word = 0 at startup
21	110	2	1	HI_FCU_S FCU- subsystem status	OFF at startup
21	110	3	1	HI_LCU_S LCU- subsystem status	OFF
21	110	4	1	HI_WBSV_S WBS-H status	OFF
21	110	5	1	HI_WBSH_S WBS-V status	OFF
21	110	6	1	HI_HRSV_S HRS-H status	OFF
21	110	7	1	HI_HRSH_S HRS-V status	OFF
23	118	4	1	HI_WBSV_HK_S HK validity	Invalid at startup
23	118	5	1	HI_WBSH_HK_S HK validity	Invalid
23	118	6	1	HI_HRSV_HK_S HK validity	Invalid
23	118	7	1	HI_HRSH_HK_S HK validity	Invalid
24	122	0	32	AID_spectroscopy	AID of the presently running activity (= 0 at startup)
25				HRS trasfer counters	
26				WBS Transfer Counters	

5.3 HIFI subsystems switch on

The S/C switches on the subsystems one after the other:

- the S/C powers on a subsystem,
- the S/C notifies to the ICU the status of this subsystem, by sending to ICU a HIFI_notify_PDU_status TC packet (ref to AD3, section 4.3.2 for a decription of the packet).

From now on, the data from the switched on subsystem are included in the nominal HK packet with those of the ICU itself. Refer to section 6.2.1 of this document for a description of the HK packet organisation.
The other instrument subsystems are not polled with HK requests/data transfer requests until ICU has not been notified that they have been switched on.

6 HIFI subsystems power off procedure

The power off procedure of the subsystem is implemented as follows:

- the S/C notifies to the ICU that a subsystem is switched off, by sending to ICU a HIFI_notify_PDU_status TC packet (ref to AD3, section 4.3.2 for a decription of the packet).
- the ICU stops sending HK requests to the switched off subsystem,
- the S/C powers off this subsystem.

The procedure is repeated for each subsystem.

The power off procedure described above should be carried out when all subsystems are switched on and should not be used if only the ICU had been powered and no other subsystem is providing HK.

7 Runtime Instructions

In this section all services implemented onboard are presented, with an indication of the results expected when the execution of the service is commanded.
The services are commanded by using TC packets, whose overall structure is defined in AD1.

7.1 General Remarks

7.1.1 HIFI OBS Mutually Exclusive Commands (MEC)

In HIFI OBS the following policy has been adopted: each time a new TC is ingested, it shall be executed immediately. Depending on the nature of the two TCs (the ingested one and the previously running one)this assumption implies a possible abort of the running procedure.
The following table contains the list of commands that imply the abort of previously running commands.

All the TCs not included in the list don't interfere with the spectrometers data acquisition and are executed immediately.

Table 4 HIFI OBS Mutually Exclusive Commands (MEC)

TC	Type	Sub type	FID	AID
Load Vector Scan				
HIFI_Load_vector_nom_1a	8	4	8	1
HIFI_Load_vector_nomsafe				
Vector Scan				
HIFI_vector_scan	8	4	8	2
LO tuning				
HIFI_Tune_LO_Using_MXCH	8	4	8	3
HIFI_Tune_LO_Using_MXCV	8	4	8	4
LO set without retune				
HIFI_Conf_f_LCU_noretune	8	4	8	7
Verify LCU CRC				
HIFI_check_LCU_memory	8	4	3	5
IV curve				
HIFI_measure_LCU_IV	8	4	15	1
Non Periodic HK LCU				
HIFI_non_periodic_hk_LCU	8	4	3	3
Set HK rate				
HIFI_Housekeeping_on	8	4	3	1
Non Periodic HK FCU				
HIFI_non_periodic_hk_FCU	8	4	3	2
Non Periodic memory HK LCU				
HIFI_read_LCU_mem	8	4	3	4

HIFI_LCU_macro_tuning_hk	8	4	3	4
HIFI_LCU_macro_buffers	8	4	3	4
FCU parameter scan				
HIFI_FCU_parameter_scan	8	4	7	1
Diplexer scan without if				
HIFI_scan_diplexer	8	4	7	2
Diplexer scan with if				
HIFI_scan_diplexer_if	8	4	7	3
HRS tuning				
HIFI_Tune_HRS	8	4	9	1
WBS tuning				
HIFI_Tune_WBS	8	4	9	2
Spectroscopy measur.				
HIFI_Spectr_total_power	8	4	11	1
HIFI_Spectr_fast_chop	8	4	11	2
HIFI_Spectr_slow_chop	8	4	11	3
HIFI_Spectr_freq_switch	8	4	11	4
HIFI_abort_spectroscopy	8	2	11	
HRS mixer magnet tuning				
HIFI_Tune_mxmgc_useHRS	8	4	9	3
WBS mixer magnet tuning				
HIFI_Tune_mxmgc_useWBS	8	4	9	4
HRS functional test				
HIFI_HRS_functional_test	8	4	10	4
wbs comb (wbs calibrate)				
HIFI_WBS_Comb	8	4	10	3
wbs zero(wbs calibrate)				
HIFI_WBS_Zero	8	4	10	2
engineering scan				
HIFI_engineering scan	8	4	7	4
Peak up				
HIFI_configure_peakup	8	4	13	1
HIFI_correction_chopper	8	4	13	4
HIFI_acquire_peakup_hrs	8	4	13	2
HIFI_acquire_peakup_wbs	8	4	13	3
simulate science				
HIFI_simulate_science	8	4	16	127
goto safe				
HIFI_goto_safe	8	4	17	0
memory check				

HIFI_check_memory	6	9		
HIFI_check_PM_memory	8	4	10	5
copy OBS				
HIFI_copy_mem_to_high	8	4	10	6
HIFI_copy_mem_to_low	8	4	10	6
mem dump				
HIFI_dump_memory	6	5		
HIFI_abort_memory_dump	6	11		
eeprom write				
HIFI_eeprom_write	8	4	16	2

7.1.2 Buiding Block ID handling

Starting from OBS 5.9, the BBID handling onboard is implemented as follows:

1) We have two variables onboard: cur_bbid and meas_bbid.
- cur_bbid is changed every time a new TC with a BBid is received.
- meas_bbid is changed only when a MEC (see Table 4) command with BBid is received
2) all TM produced by a MEC (see Table 4) command reports meas_BBid
3) all other TM produced reports the cur_bbid

In the preceeding releases all TM packets have the last commanded BBId. As a consequence, if a new TC (with a new BBID) is ingested during a spectroscopic measurement, the science data belonging to that measurement show different BBIDs.

7.1.3 Spacecraft Interface Communication

7.1.3.1 DESCRIPTION OF THE CHECKS PERFORMED BY THE OBS ON THE 1553 INTERRUPTS

The goal of the checks is to detect if any of the two following requirements is violated:
Req.1: The OBS shall receive 64 interrupts per seconds from the 1153 bus.
Req.2: The OBS shall be able to serve an interrupt before the next one is received.

An additional implicit requirement is that:
Req.3: the minimum possible overhead shall be inflicted to the interrupt handling routines (which are time critical) for performing the checks.

OBS HANDLING OF THE INTERRUPTS

Let's number the interrupts received within one second from 0 to 63. These interrupts are conveniently divided into two types:
type 0: interrupt 0 (corresponding to the Mode Synch Without data word 1553 message)
type 1: all other interrupts

The OBS serves all the interrupts by means of a two steps procedure:
step 1: when an interrupt is received the assembler routine irq2.s is started. It rises an event to wake up the insterrupt service routine (in this case a C language routine).

step 2: when the event is rised the C routine isr1553 is triggered. It handles the two interrupts types differently (and in different pieces of code):
- if an interrupt of type 0 is received, no data transfer takes place. Only the OBS timing is corrected
- if an interrupt of type 1 is received a packet exchange (TC or TM) between the spacecraft (bus controller, BC) and the ICU OBS (rremote terminal, RT) can possibly be performed.

OBS variables involved in the processing:

1. com1553_isr_counter: this is an always increasing counter handled by the assembler routine. It is increased by one in the assembler routine every time the routine is executed.
2. Buf_0_SubFrame_Counter: this is a buffer handled (as described later) from the C routine when an interrupt of type 0 is received
3. Buf_1_SubFrame_Counter: this is a buffer handled (as described later)from the C routine when an interrupt of type 1 is received
4. Delta_0_SubFrame_Counter and Delta_1_SubFrame_Counter: these are variables to store the deviations from the nominal values in case of violations of the requirements
5. Counter_flag: this is a flag that is handled by the C routine and by the OBS. It is rised by the C routine as soon as a violation of the requirements is detected. When rised it stops further checks since we have to report the detected violation before restarting the checks. It is resetedd by the OBS as soon as the violation is reported in order to restart the checks.

PERFORMED CHECKS

1. When the C routine processes an interrupt of type 0 it performs the following check, aimed at detecting violations of Req.1:
   - Buf_0_SubFrame_Counter holds the value of the com1553_isr_counter when the last interrupt of type 0 was received
   - if com1553_isr_counter - Buf_0_SubFrame_Counter != 64 it means that at least one interrupt has not been received by the assembler routine -> rise error flag (i.e. Counter_flag) and compute Delta_0_SubFrame_Counter = com1553_isr_counter - Buf_0_SubFrame_Counter and Delta_1... = 1 (to store the violation type)

- Update Buf_0_SubFrame_Counter = Buf_1_SubFrame_Counter = com1553_isr_counter

2. When the C routine processes an interrupt of type 1 it performs the following check, aimed at detecting violations of Req.2:
   - Buf_1_SubFrame_Counter holds the value of the com1553_isr_counter when the last interrupt (of any type) was received
   - if Delta_1_SubFrame_Counter != 1 it means that at least one additional interrupt has been received before the current interrupt has been processed -> rise error flag (i.e. Counter_flag) and compute Delta_1_SubFrame_Counter = com1553_isr_counter - Buf_1_SubFrame_Counter and Delta_0... = 64 (to store the violation type)
   - Update Buf_1_SubFrame_Counter = com1553_isr_counter

Note that: the checks are not performed if the Counter_flag is rised: this is because we can only report one violation per HK period (see later) and there is already a violation to report.

VIOLATIONS REPORTING

The violations are reported in the HK packet (see A3.1).
For the sake of simplicity we report only one violation per HK packet
The HK procedure checks the Counter flag. If it is set there is a violation to report. It reads and report the Delta_X... buffers. If it is not set reprort the nominal values.
As soon as the HK packet is sent, reset the Counter_flag in order to restart the checks

The implemented procedure will always detect and report the first violation of one of the two requirements for every HK period. Further violations in the same HK period are lost.

7.2 Service Type 1: Telecommand verification

A telecommand sent to ICU by the CDMS shall be ingested as soon as its presence on the ICU-spacecraft interface is signalled by the satellite Bus Controller. This implies that the procedure for the acceptance of the TC is immediately executed, and, if the verification of its correctness is successfull, any presently running activity onboard due to the execution of previous TCs is aborted to allow the execution of the new TC.

The TC acceptance reports are delivered within 500msec from the TC ingestion.

7.2.1 TC acceptance

Each time the ICU receives a TC packet from the S/C, a check procedure is started to test the consistency of the packet with the ESA standard, as described in AD8.
A TM report (1,1) if the packet is accepted, or a TM report (1, 2) if an error is found, is generated.
In both cases, and in general for all the TM packets of this service, the first two words of the application data field are copied from the first two words of the TC packet header. Refer to AD4 for the definition of the structure of all the TM verification reports.

In case a TM report (1,2) is prepared, a failure code is reported in the packet data field. A list of all possible failure codes is given in Table 5 along with the other parameters included in the report.
The Successfull Acceptance TM report (1,1) is issued only if the received TC packet has the acknowledge acceptance bit set to 1 (see AD2).
The Acceptance failure TM report (1,2) is always generated in case of errors during the TC acceptance.
Both TM services can be disabled with Service 14 (see Section 4.7).

Table 5 - Failure codes provided in case of unsuccessfull acceptance

Report Type Subtype	OBS Mnemonic	Short Mnemonic	Code	Number of params.	Report parameters	Notes
1,2	NOK_CMDSEQ_ILLEGAL_APID	ILL_APID	0x0000	1	APID	List of HIFI APIDS in sect.
1,2	NOK_CMDSEQ_ INVALID_LENGHT	INV_LEN	0x0001	1	Packet length	Length=(number of bytes in Application Data field –1)
1,2	NOK_CMDSEQ_ INVALID_CRC	INV_CRC	0x0002	1	Received CRC	CRC algorithm described in
1,2	NOK_CMDSEQ_ ILLEGAL_PACKET_TYPE	ILL_TYPE	0x0003	1	Received Type,subtype	
1,2	NOK_CMDSEQ_ ILLEGAL_PACKET_TYPE	ILL_SUBTY	0x0004	1	Received Type,subtype	
1,2	NOK_CMDSEQ_ ILLEGAL_APPLICATION_DATA	ILL_APP_DATA	0x0005	1	Received FID,AID	Service (8,4): Perform activity of a function Wrong Function ID Wrong activity ID
1,2	NOK_CMDSEQ_LENGTH_SECOND_CK	INV_NUM_PAR	0x0010	2	Actual length, expected length	A TC with a unexpected length was received
1,2	NOK_CMDSEQ_OBSOLETE_AID	ILL_AID	0x0011	0		An Obsolete AID has been received

7.2.2 TC execution

Telecommand execution completed

This TM report (type,subtype 1,7) is generated only if the received TC packet has the acknowledge execution bit set to 1 (see AD2). The report is delivered after the ingested TC has been succesfully executed onboard.
The content of the report has the same structure of the TC acceptance report (refer to AD4) and its generation can be disabled (see Service 14).
Being related to the actual execution of the TC, this report is generated with a TC dependent delay time with respect to the TC ingestion.
The modality of the TC execution completed report generation will be described in detail in the various TC sections.

Telecommand execution failure

This TM report (type,subtype 1,8) is used to signal that the execution of a TC failed. It can be disabled (see Service 14). There are several sources of execution failures. For each one of the TC described in this document, the possible execution filure codes will be listed, along with the parameters included in the reports.

7.3 Service Type 3: Housekeeping Data Reporting

The OBS only generates HK packets of type 3 and subtype 25: TM (3,25).
The HK packet definition is stored in tables in the OBS contained in the file conf_tab.c.

APID N	APID R	HK Packet SID hex	HK Packet SID dec	Packet
1024	1025	0x0001	1	Essential HK Packet
1026	1027	0x0404	1028	Nominal HK Packet. A subset of the parameters contained in this packet will be used for monitoring.
1026	1027	0x0010	16	Non-periodic FCU_rev0
1026	1027	0x0011	17	Non-periodic FCU_rev1
1026	1027	0x0012	18	HIFI_LCU_safetyTables
1026	1027	0x0013	19	HIFI_LCU_RAM_HK
1026	1027	0x0014	20	HIFI_LCU_macro_buffers
1026	1027	0x0019	25	HIFI_HRS_H_IF_POWER_phase1
1026	1027	0x001A	26	HIFI_HRS_H_IF_POWER_phase2
1026	1027	0x001B	27	HIFI_HRS_V_IF_POWER_phase1
1026	1027	0x001C	28	HIFI_HRS_V_IF_POWER_phase2
1026	1027	0x001D	29	HIFI_WBS_H_IF_POWER_phase1
1026	1027	0x001E	30	HIFI_WBS_H_IF_POWER_phase2
1026	1027	0x001F	31	HIFI_WBS_V_IF_POWER_phase1
1026	1027	0x0020	32	HIFI_WBS_V_IF_POWER_phase2
1026	1027	0x010A	266	HIFI_FCU_parameter_scan_report
1026	1027	0x0102	258	HIFI_diplexer_scan_report_with_IF
1026	1027	0x010E	270	HIFI_diplexer_scan_report
1026	1027	0x0103	259	HIFI_LCU_IV_report
1026	1027	0x0104	260	HIFI_HRS_H_tune_report
1026	1027	0x0105	261	HIFI_HRS_V_tune_report
1026	1027	0x0106	262	HIFI_WBS_H_tune_report
1026	1027	0x0107	263	HIFI_WBS_V_tune_report
1026	1027	0x010B	267	HIFI_MX_MGC_useHRS_report
1026	1027	0x010C	268	HIFI_MX_MGC__useWBS_report
1026	1027	0x0110	272	HIFI_Vector_scan_report
1026	1027	0x010F	271	HIFI_engineering_scan_report
1026	1027	0x0C17	3095	HIFI_Vector_scan_report_nominal
1026	1027	0x0C18	3096	HIFI_Vector_scan_report_diagnostic

Table 6 List of allowed HK packets

The allowed HK packets can be generated contemporaneously:

- the Essential HK and the periodic HK are generated periodically with a commandable acquisition frequency.
- The non-periodic HK packets are generated upon command reception.
- The IF power HK packets are generated during the measurements, one packet per each spectrometer packetisation.

The OBS does not perform any check on the ICU workload implied by the HK parameters collection.
In particular, it should be remembered that the minimum time to issue a HK parameter request to the Low speed subsystems and receive the corresponding parameter is 3 milliseconds. This constraint, coupled with the fact that each HK request in case of LCU implies the sending of a preceeding prepare command, means that, given the total number of periodic HK to include in the packet, the minimum acquisition frequency is 1 sec.

Note: Due to a bug in the VIRTUOSO compiler, the first time a task under execution enters a Task Sleep after the task resume, the sleep time is shorter with respect to the commanded one. A statistical investigation of this

problems, made with the SRON subsystem simulator showed that the decreasing is, on average, of 500microsec, but can reach peaks of 800microsec. This is the reason why, instead of polling HK every 2 msec (which should be sufficient for the LCU HW), the HK requests are issued every 3msec.

7.3.1 Periodic HK acquisition

7.3.1.1 Set HK rate

The nominal/default periodic HK acquisition frequency is 1 packet/4 seconds.

The periodic HK acquisition frequency is commanded by using the *HIFI_Housekeeping_on* TC. The allowed rates are reported below, together with the corresponding value of the parameter HIF_HK_rate, to be set in the TC:

periodic acquisition rate	HIF_HK_rate value
1_pkt_per_s	2
1_pkt_per_5_s	0
1_pkt_per_10_s	1
1_pkt_per_3_s	3
1_pkt_per_4_s	4
1_pkt_per_2_s	2

Table 7 HK periodic acquisition rates

The change in the acquisition frequency is implemented by reprogramming the VIRTUOSO low resolution timer onboard.
As soon as the TC is ingested, an abort measurement procedure is started to stop any running measurement onboard. Then the timer used to trigger the HK_ask (see AD2, section 3) task is restarted, with a period equal to the new requested frequency. This implies that the time interval between two successive nominal HK packets can never be shorter than the acquisition interval.

The first HK packet will be issued after a time interval from the reception of the TC acceptance report corresponding to the new selected rate.
The TC execution completion report is issued after the LowResolution Timer restart.

In case of a HIF_HK_rate parameter >4, a TC execution failure report with code 0x404 is issued. See section 7.3.6 for a description of the error.

Staring from OBS version 5.0, the format of the *HIFI_Housekeeping_on* TC is changed according to what is reported in SCR 1163, in agreement with the following description.

Position (byte)	Length (bits)	Field	Remark
10	8	Function_ID	Fixed: 0x03
11	8	Activity_ID	Fixed: 0x01
12	16	Structure_ID	Fixed: 0x0001
14	16	HIF_HK_rate	See Table 5
16	16	HK_Subsysstatus	The word is coded as described in AD3, sect 4.3.2. for the notify PDU status TC. Default: 0x003f (all enabled)

Table 8 *HIFI_Housekeeping_on* TC structure

OBS polls for housekeeping data only those subsystems for which the corresponding bit in the HK_Subsysstatus parameter is set to 1: even if a subsystem is switched on and active, it is possible to disable the HK acquisition from it.

At startup the HK acquisition from all subsystems is enabled: as soon as the subsystem is switched on with the HIFI_notify_PDU_status TC, the OBS starts polling it for HK data. From now on, to disable the HK acquisition from a subsystem the HIFI_Housekeeping_on TC shall be used.
The HK acquisition from a subsystem is (re-)enabled by (re-)sending the HIFI_Housekeeping on TC with the updated HK_Subsysstatus word.

7.3.1.2 Stop periodic HK acquisition

The HK acquisition can be stopped by sending the *HIFI_Housekeeping_off* TC.
The command is executed by disabling the HK acquisition onboard, which means that the HK_ask task is still in execution, triggered at the current HK acquisition frequency by the VIRTUOSO low resolution timer, but no operations at all are done by the task.
It could happen that the last triggered HK request is still under processing onboard and that therefore one HK packet can be delivered even after the HK acquisition has been stopped. The reception of one only HK packet after the reception of the TC acceptance and TC execution reports shall not be cponsidered as an error.

The TC execution completion report is issued immediately after the TC acceptance report.
No TC execution failure reports are foreseen.

7.3.1.3 Periodic HK packets

Refer to AD5, section 3.1 for the packet content description.
Starting from OBS 4.3, the HIFI_periodic_HK_rev7 packets are implemented. See A2.1 to have the list of the actual HK requests issued by OBS.
Refer to Table 6 this document for the packets APID and SID.

7.3.2 Non-periodic HK Acquisition

7.3.2.1 FCU non periodic HK request

The non periodic HK acquisition from FCU is commanded by using the *HIFI_non_periodic_hk_FCU* TC: Function Management Service - Perform activity of a Function: Type 8, substype 4, FID 3, AID 2.
Refer to AD3 section 4.3.3.2
The TC is executed by the LS_HDL task (see AD2, section 3).
If only the nominal HK acquisition activity is running onboard, the command execution is started immediately, and the execution is completed within 200msec from the TC ingestion, with the generation of the Non periodic HK report.
If the FCU_non_periodic_HK request is ingested before the previous TC execution onboard has been completed, the previous TC execution is aborted. This implies an additional interval of 1100msec to be added to the TC duration, to account for the abort procedure duration.
The TC execution report is issued immediately after the generation onboard of the Non periodic HK report.

The following sources of failures are identified in the procedure execution:
1) If the HK acquisition from the FCU subsystem is disabled the TC execution is not started and a TC execution failure report is issued immediately.
2) A problem occurred in enqueuing the HK request in the LS high priority queue (see AD2, section 3): a TC execution failure is issued and the TC execution is stopped.

Refer to
Table 9 for the errors description.

7.3.2.2 FCU Non Periodic HK packets

Refer to AD5, section 3.2, 3.3 for the definition of the contents of the FCU non periodic HK reports.

Starting from OBS 4.3.3, the FCU_non_per_HK_003 table is implemented (see SCR 954).
See A2.1 to have the list of the actual HK requests issued by OBS.

Refer to Table 6 this document for the packets APID and SID.

7.3.2.3 LCU non periodic HK request

The non periodic HK acquisition from LCU is commanded by using the *HIFI_non_periodic_hk_LCU* TC: Function Management Service: Perform activity of a Function: Type 8, substype 4, FID 3, AID 3 (Refer to AD3 section 4.3.3.3) .
Refer to AD12, section 1.2.1 for a description of the TC and of the HK report.
Refer to AD12, section 1.2 for a description of the procedure.

The TC is executed by the LS_HDL task (see AD2, section 3).
The TC execution report is issued immediately after the generation onboard of the Non periodic HK report.
The LCU non periodic HK activity implies the sending out of 55 couples of LS commands and hk requests.
Each one of this LS transactions lasts for at least 3msec, which means that the duration of this part of the activity is at least 330msec.
Therefore, a conservative estimation of the duration of the overall procedure is 400msec (which is a little bit longer than the average time between the reception times of the TC acceptance report and the NP HK packet).
If a new TC is received before this time, the presently running acquisition is aborted and the new TC is processed.

A TC execution failure is issued in case of problems enqueuing the HK request in the LS high priority queue (see AD2, section 3). Refer to
Table 9 for the error description.

The following consistency checks are performed onboard on the TC parameters:

1) "Band nx" shall be >=1 and <= 14
2) "Freq Nx" shall be >=0 and <=31

A TC execution failure is issued in case of problems detected in these consistency checks.
An error report is generated in case of problems in enqueuing the HL_RD_GET_TAB HK requests to the LS task. In this case the procedure is not stopped, and the final report will contain an empty parameter corresponding to the missing Hk request.
A TC execution failure is issued in case of problems in enqueuing the HL_LD_INDEX_TAB command to the LS task: in this case the procedure is stopped and the final report will not be issued.

Refer to
Table 9 for the error description.

7.3.2.4 LCU Non Periodic HK packets

Refer to AD5, section 3.3.1 for the definition of the contents of the LCU non periodic HK reports (LCU safety Table dump).
Refer to Table 6 this document for the packets APID and SID.

7.3.2.5 LCU memory read

The purpose of this command is to read a certain part of the LCU-RAM and to get the result in the HK-stream.
The activity is commanded by using the *HIFI_LCU_memory_read* TC: Function Management Service: Perform activity of a Function: Type 8, substype 4, FID 3, AID 4 (Refer to AD3 section 4.3.3.4) .
Theactivity has been described in a dedicated SCR (SCR 721) .
The final agreement on the procedure is reported below:

At the reception of the TC the OBS shall:
1) issue a set page command to LCU (LScmd), built up using the 8 less significat bits of the start page field:

yy = start page & 0x00ff
LScmd=0xF30ACCyy

2) issue N HK request to LCU (Hkreq), built up using the 8 less significat bits of the start address field:
yy = (start address+2i) & 0x00ff
HKreq=0xB3yyFFFF
where i ranges from 0 up to N (the commanded HIF_LCU_memlen parameter)
A proper dump stays always at one page.

3) wait for the HK to be provided by LCU and prepare a non periodicHK report (type 3, subtype 2) with a SID equal to the commanded one (see TC structure in AD3 section 4.3.3.4).

The TC is rejected if *(start_add + 2*N+1)) > 0xFFFF*

In OBS 4.3 and following, the above specification has been implemented.
Refer to tables 7.8 and 7.9 for the failure reports generated during the execution of the TC.

The TC is executed by the LS_HDL task (see AD2, section 3).
The TC execution report is issued immediately after the generation onboard of the LCU_dump HK report.
The LCU non periodic HK activity implies the sending out of 55 couples of LS commands and hk requests.
Each one of this LS transactions lasts for at least 3msec, which means that the duration of this part of the activity is at least 330msec.
Therefore, a conservative estimation of the duration of the overall procedure is 400msec (which is a little bit longer than the average time between the reception times of the TC acceptance report and the NP HK packet).
If a new TC is received before this time, the presently running acquisition is aborted and the new TC is processed.

A TC execution failure is issued in case of problems enqueuing the HK request in the LS high priority queue (see AD2, section 3). Refer to
Table 9 for the error description.

The TC is rejected if (start_add + 2*(Num_words+1)) > 0xFFFF

7.3.3 IF-power HK packets

All spectroscopy measurements onboard provide in output an IF power HK packet per each packetisation per each polarisation per each accumulation buffer. Refer to AD5, section 3.5 for a description of the contents of the HRS and WBS IF-power HK reports.
Refer to Table 6 this document for the packets APID and SID.
The IF-power HK packet is always downloaded before the corresponding science data.

HRS:
In case of HRS data the IF power of each chip is calculated as follows

IF=(2*C0-duration)/(duration)

Duration is the second word of each chip, C0 is the fourth word of each chip. The duration is the total duration of all integrations.
In the Ifpower HK packet all 16 chips Ifpowers are provided.
For each chip the value reported in the packet is equal to **IF*2^16** (the scaling factor allows to fully exploit a 16 bits range).

WBS:
In case of WBS data, for each one of the four CCDs of each spectrometer the IF power is the avaraged value of the selected pixels, normalized by the scan counter and multiplied by 2^6, in order to fully exploit a 16 bits range.

WARNING: Per each WBS chip, to obtain the IfPower it is necessary first to add all data contained in the selected range, and then to divide by the number of data. No check onboard is done on the possibility that the result of the sum is greater than the maximum allowed for a 32bit number. In this case it will be no possible on ground to recover the actual Ifpower value. It is suggested, in case of expected large signals, to command a data right shift, which will be applied before the Ifpower calculation.

Both TM reports can be disabled with Service 14 (see Section 4.7).

7.3.4 Essential HK packets

The OBS generates the Essential HK packets at the same frequency of the periodic HK. The Essential HK can be disabled with Service 14 (see Section 4.7). The APID of Essential HK packets is described in AD5.

7.3.5 Diagnostic HK packets

Starting from OBS version 4.3, due to the request in SCR 929, the type/subtype of diagnostic science packets (i.e. scans and tune reports) has been changed to the type/subtype of HK packets (3,25). Refer to section 7.10.1 for a complete list of the diagnostic packets produced onboard.

7.3.6 HK acquisition errors

Table 9 Errors generated in the hk_ask task

Report Type Subt	EV type	OBS Mnemonic	Short Mnemonic	Code	N PAR	Notes
5,4	0xA000	ERR_LS_FIFOPUT_LP_QUEUE	LS_LP_QUEUE	0x502	0	FIFO put on LS low priority queue failed.
5,4	0xA000	ERR_LS_FIFOPUT_HP_QUEUE	LS_HP_QUEUE	0x503	0	FIFO put on LS high priority queue failed.
5,4	0xA000	ERR_HK_ASK_FIFOPUT_HK_TM_QUEUE	HK_TM_QUEUE_0	0x400	0	OBS runtime error: FIFO put on HK TM queue failed
5,4	0xA000	ERR_HK_ASK_GET_BLOCK_HKPOOL_FULL	HKASK_PUT_HKTMQ	0x402	0	No Blocks avilable on Hk Pool.
5,4	0xA000	ERR_HK_ASK_SWERR	HKASK_SWERR	0x403	1, 0	Error in the parameters to execute limit checking
5,4	0xA000	ERR_HK_ASK_PKT_NO	HKASK_PKTNO	0x405	1 (pkt numbe r)	Error in the number of generated nom./ess. HK packets
5,4	0xA000	ERR_HK_ASK_INTERNAL_ERROR	HKASK_ERROR	0x40A	0	Error in the table used to poll FCU/LCU HK
5,4	0xA000	ERR_HK_ASK_FIFOPUT_LS_HP_QUEUE	LS_HP_QUEUE_ERR	0x40E	0	OBS runtime error: FIFO put on LS high priority queue failed
5,4	0xA000	ERR_HK_ASK_LS_HRS_REQUEST	HKASK_ERROR	0x40C	0	FIFO put on LS HP queue failed for HRS data transfer cmd.
1,8		EXF_HK_ASK_FIFOPUT_HK_TM_QUEUE	HK_TM_QUEUE_1	0x401	0	Execution failure: FIFO put on HK TM queue failed
1,8		EXF_HK_ASK_HK_RATE_INDEX_OOL	HKPOOL_FULL	0x404	1	Exec. Failure: Requested rate not available. The parameter is the wrong index requested
1,8		EXF_HK_ASK_SINGLE_HK_LCU_INVALID_D	HKASK_LCUSHK	0x0406	0	Exec. Failure:

		ATA				Consistency check of parameters in LCU_single HK request failed
1,8		EXF_HK_ASK_ERROR_LS_HP_QUEUE	HKASK_LSHPQUEUE	0x407	0	Execution failure: FIFO put on LS HP queue failed
1,8		EXF_HK_ASK_LCU_OFF	HKASK_LCUOFF	0x408	0	Execution failure: LCU is Off
1,8		EXF_HK_ASK_FCU_OFF	HKASK_FCUOFF	0x409	0	Execution failure: LCU is Off
1,8		EXF_HK_ASK_MEM_HK_LCU_INVALID_DATA	HKASK_LCUMEM	0x40b	0	Execution failure: commanded mem dump parameters out of limits

7.4 Service Type 5: Event Reporting

Upon detection of errors/failures onboard the event reporting service guarantees the provision of an event to signal the occurrence of the detected anomaly.

7.4.1 Boot Software Errors

The following exception/alarm reports can be generated by the boot software during the ICU start-up procedure. They are described in RD3. The table will be updated with the ICU FM delivery.

Table 10 - Boot Software events

Event	APID	type	subtype	Event ID
HIFI_ready_event	1025	5	2	0x8008
HIFI_PM_test_event	1024	5	4	0x8001
HIFI_Data_memory_event	1024	5	4	0x8002
HIFI_EEPROM_memory_event	1024	5	4	0x8003
HIFI_TC_verification_event	1024	5	4	0x8004
HIFI_Load_EEPROM_PM_event	1024	5	4	0x8005
HIFI_Load_DM_PM_event	1024	5	4	0x8006
HIFI_boot_DM_PM_event	1024	5	4	0x8007

7.4.2 OBS Runtime Errors

Upon verification onboard of runtime errors, the OBS generates only event reports (type 5, subtype 4), with the structure defined reportd in the following table. For OBS RUNTIME ERRORS the Event ID is fixed 0xA000.

Table 11 Structure of Alarm Reports

Start Byte	Start bit		Length Monitor Parameter value
16	16	Event ID	0xA000
18	16	Structure ID	0
20	32	Observation ID	
24	32	Building block ID	

28	16	Alarm counter	
30	16	Error Code	See Table 29
32		Parameters	the parameter field is structured as follows: - 1st parameter (16 bits) = Number N of following parameters - N parameters (N*16 bits) = parameters relevant to the event interpretation

The list of all the possible error codes of the OBS runtime errors is provided in Appendix A1 ,Table 29.

Whenever one of the listed runtime errors can be generated only during the execution of one specific service, the details on the generation of the event are described in the section pertaining to the service.

In Table 29 the following error severity classification has been adopted:
1- NOTIFICATION (used to add info about a non nominal situation detected onboard, no action are taken onboard following this error detection)
2 - WARNING about the detection of an error that can have consequences in the espected results of the commanded activity.
3- ALARM: error for which an action is taken onboard to interrupt the presently running activity (e.g. abort measurement).

A particular attention shall be given to the case in which too many events (5,4) are generated with a rate greater than the TM download datarate (27 packets/sec). In this case there is a possibility that the Events queue onboard becomes saturated. The OBS generates a ERRHDL_EVQUEUE event, and purges the queue. The consequence of this behaviour is that neither (1,7) nor (1,8) TM packets, referred to the presently running measurements will be generated any more (see SPR 1412).

7.5 Service Type 6: Memory Management

This service allows to load, dump and check the onboard memory. These functionalities are commanded by the following TCs described in AD3 sect 4.1:

1. HIFI_load_PRAM
2. HIFI_load_DRAM
3. HIFI_load_LCU
4. HIFI_dump_memory
5. HIFI_check_memory

In the packet data field, after the 4 bytes of the data field header, a common structure containing the parameters necessary to identify the memory area that is the object of the memory operation is adopted.
Specifically the following fields are used to identify the area and are common to all memory management TCs:

Position (byte)	**Lenght (bit)**	**Name**	**Meaning**
10	8	Mem_id	ID of the memory segment
11	24	Mem_start	Initial address of memory area
14	16	Mem_len	Length of the memory area

The Mem_ID identifies the type of memory where the memory operation has to be carried out and can have the following values:

Block	**MemID**	**Physical starting address**	**Physical ending address**	**Max. Length (SAU)**
PM (PRAM)	0x00	0x0	0x7FFFF	0x80000
DM (RAM)	0x01	0x0	0x7FFFF	0x80000

DM(1553 DRAM)	0x02	0x8F000000	0x8F003FFF	0x4000
DM (EEPROM)	0x03	0x80000000	0x8003FFFF	0x40000
LCU	0x04	0x0	0xFFFF	0x1000

Table 12 - OBS Memory organisation

The Mem_start and the Mem_len parameters specify the start address and the length of the memory area involved in the memory operation. These parameters are always expressed in SAU (smallest addressable units).
This unit corresponds to 6 bytes (PRAM), 4 bytes (DRAM) or 2 bytes (LCU). Since the packet is organized in 16 bits words, the following table is applicable

memory type	Mem ID	TC words	bytes
PM	0	3	6
DM	1	2	4
1553 DM	2	2	4
EEPROM DM	3	2	4
LCU	4	1	2

Table 13 – SAU composition

The Mem_start and the Mem_len parameters have limits that depend on the actual size of the physical memory:
1. For the OBS Mem_dump and Mem_check any command that exceeds the limits reported in Table 12 would generate an execution failure and one or more runtime errors (seeTable 29 and Table 30 for a detailed description of the errors);
2. For the LCU Mem_dump the maximum dumpable area is limited by the constraints that the whole dump should fit into a single TM packet, the maximum dumpable length is teherfore 499 (see Table 29 for a detailed description of the errors); this constraint does not hold for the other dumps;
3. For the memory load the limit on the length is essentially imposed by the requirement that the words to be loaded fit in the same TC. Multiple Load commands are needed in order to load long parts of memory.

Not all the memory operations can be carried out on all the memory areas. The following table details what are the allowable operations for each ID:

Memory	ID	Load	Dump	Check
PM	0	partially	yes	yes
DM	1	yes	yes	yes
1553 DM	2	no	yes	yes
EEPROM	3	no	yes	yes
LCU	4	yes	yes	no

The PROM content is not actually visible to the OBS. However after the ICU is switched on, the PROM SW is copied in PM, where remains until ICU is switched off. The PM area reserved to the PROM mapping starts at the physical address 0x0000 and ends at the physical address 0x3fff. This is the reason why in the previous table the PM is marked as partially loadable: the area below 0x4000 cannot be loaded.

The execution time of the memory management TCs is variable. Assuming that the OBS is not carrying out any other activity onboard:
- The load operations in PM and DM (Ids from 0 to 3) are executed immediately and the Execution Completion report is issued within 500msec from the TC acceptance;
- The load operations on the LCU require 6 msec for each loaded word;
- The check operations are executed immediately and the check report is issued within 500msec from the TC acceptance;
- The dump operations on PM and DM depend on the number of dumped words: if several TM packets are needed to dump, a maximum rate of one TM packet every 300 msec is artificially enforced in order not to overflow the OBS pools.
- The dump operation on the LCU requires require 3 msec for each dumped word plus 10 msec (this is a safe value) for setting the LCU page.

All the possible sources of execution failures during the execution of the memory services are reported in Table 14.

All the possible sources of OBS runtime errors during the execution of the memory services are reported in Table 15.

Table 14 - Memory service Execution failure codes

Error ID	Onboard mnemonic	Short mnemonic	Detected runtime error	Parameters
0x1000	EXF_MEM_INVALID_MEMLENGTH	MEM_LEN	The commanded length in SAU is inconsistent with the number of 16 bit words of the parameter.	1: Length field
0x1001	EXF_MEM_INVALID_ADDRESS	MEM_ADD	This error is caused by a start address greater than the end address reported in Table 12.	1: Address as extracted from the TC
0x1002	EXF_MEM_INVALID_MEMID	MEM_ID	The commanded memory ID is referred to a not existing ID	1: Memory ID as extracted from the TC
0x1003	EXF_MEM_INVALID_CRC1	MEM_CRC1	Inconsistency between the OBS computed checksum and the checksum containe in the TC. Both Checksum are referred to the memory data in the TC.	1:The crc computed by the OBS
0x1004	EXF_MEM_INVALID_CRC2	MEM_CRC2	Inconsistency between the OBS computed checksum on the TC memory data after their copy on the destination memory and the checksum contained in the TC.	1: The crc computed by the OBS
0x1007	EXF_MEM_BOOT_OVERWRITE	MEM_BOOT_OVRW	A PM start address shorter that 0x4000 has been requested.	TBW
0x1008	EXF_MEM_UNKNOWN_RET_CODE	MEM_UKWN_RETC	Error in memory load. The procedure returned an unespected value.	TBW
0x100a	EXF_MEM_EEPROM_WRITE_OVERFLOW	MEM_EEWRITE_OVF	Error during the EEPROM write procedure (to copy in EEPROM the full OBS image).	TBW
0x100b	EXF_MEM_EEPROM_COPY_FAILED	MEM_EECOPY_FAIL	Error during the EEPROM write procedure (to copy in EEPROM the full OBS image).	TBW
0x100c	EXF_MEM_EEPROM_WRONG_PARTITION	MEM_EE_PARTIT	Error during the EEPROM write procedure: wrong commanded partition ID	TBW
0x100d	EXF_MEM_EEPROM_WRONG_NUM_BAD_PAGES	MEM_EE_BADNPAG	Error during the EEPROM write procedure: wrong commanded number of pages to avoid to copy on.	TBW
0x100e	EXF_MEM_HS_HDL_QUEUE		The command cannot be processed due to the overflow of the HS_hdl queue.	TBW
0x100f	EXF_MEM_COPY_OBS	MEM_COPY_OBS	Failure in the execution of the Copy OBS command.	TBW
0x1010	EXF_MEM_CHECK_PM_CRC	MEM_CK_PM_CRC	The check PM command resulted in a PM CRC different from expected, indicating a possible damage in the PM memory.	TBW
0x1011	EXF_MEM_EEPROM_WRONG_BAD_PAGE_ID	MEM_EE_BADPAGID	Error during the EEPROM write procedure: one of the IDs of pages to avoid to copy on is wrong	TBW
0x1012	EXF_MEM_LCU_INVALID_DATA	MEM_LCU_DATA	Error in LCU mem Load TC parameters: total number of loadable words exceeded.	TBW
0x1013	EXF_MEM_LCU_OFF	MEM_LCU_SSOFF	Error in LCU mem Load/dump TC: LCU subsystem is OFF.	TBW
0x1014	EXF_MEM_ERROR_LS_HP_QUEUE	MEM_LSHP_UEUE	Error in LCU mem Load/dump TC: failure in enqueuing a message to the LS HP queue	TBW

Table 15 - OBS runtime error in memory service

ERROR Name	Short Mnemonic	Error Code	Error condition	Severity
ERR_MEM_FIFOPUT_HK_TM_QUEUE	MEM_PUT_HKTMQ	0x1005	FIFO put on HK TM queue failed in Memory	2

			management task. Dump/check report lost.	
ERR_MEM_DATA_OVFL	DATA_OVFL	0x1006	Failure in memory manager initialisation. The HIFI settings for the memory management service have not been loaded.	2
ERR_MEM_FIFOPUT_SD_PKT_QUEUE	SD_PKT_Q	0x1009	FIFO put on SD TM queue failed in simulate science procedure. The simulated packet is lost.	2
ERR_MEM_INTERNAL	MEM_internal	0x1016	error in generating memory dump report: a wrong RAM type has been detected.	2

7.5.1 HIFI_Load_PRAM/DRAM

The Memory Load Telecommand has type 6, subtype 2.
Refer to AD3, chapter 4, section 4.1.1 for a general template of the TC. In sections 4.1.1.2 and 4.1.1.3 the TC is detailed for the two destination memories.

The ICU performs some checks on the application data; if one check is not passed, an unsuccsessful TC verification packet is issued, with a failure code described in 7.2.1. All listed failures are applicable to this service.
If the whole chain of checks is OK, the ICU reports a TM (1,1) report.
During the execution of this TC no OBS runtime errors are foreseen.

The ICU assumes that the words are little endian: as an example, to upload the PRAM word 0x123456789ABC at address 0x46789 in PRAM, the complete application data field, without the data field header and the packet checksum, is:

Memory ID	0x0004
Start address	0x6789
Length	0x0001
Datum	0x1234
Datum	0x5678
Datum	0x9ABC
crc	

To upload the word 0x12345678 in DRAM with start address 0x59876 the corresponding packet is:

Memory ID	0x0105
Start address	0x9876
Length	0x0001
Datum	0x1234
Datum	0x5678
crc	

The maximum dimension of the packet data field of a TC is 242 bytes, 4 of which are used for the data field header and 2 for the CRC; 8 bytes are used for the memory ID, start address, length and crc. Therefore the maximum length for a memory load in DRAM is 57 words and for PRAM is 38.
If the TC is inserted in the mission timeline, its data field is reduced by 20 bytes, so that **the maximum length is 52 or 34 words, for the two memory types.**

7.5.2 HIFI_Load_LCU

Refer to AD3, chapter 4, section 4.1.1 for a general template of the TC. In section 4.1.1.4 the TC is detailed for the LCU destination memory.
See AD12, section 1.1.2 for a specification of the LCU memory dump procedure.
The correct interpretation of the TC parameters has been integrated with a dedicated SCR (SCR 915). The content of the SCR is reported below:

> *The dump command specifies a start-page and a start-address. In the requirements of the dump-command the use of this command is restricted for words on one page only.*
> *The current implementation prevents the last word on a page to be dumped.*

The request is to drop the page-restriction, and only prevent the address to run above 0xFFFF:

10	*8*	*Memory ID= 4*
11	*8*	*Spare*
12	*16*	*startAddress*
14	*16*	*Length*

*Reject if startAddress*256 + 2*(Length-1) > 0xFFFF*

Pseudo code:

```
yyyy= startAddress
for i=0 to Length -1
{
  pp= (yyyy & 0xFF00) >>8
  LScmd(0xF30ACCpp)
  qq= (yyyy & 0x00FF)
  LShk(0xB3qq)
  yyyy=yyyy+2
}
```

In OBS 4.3, the specification of SCR 915 has been implemented.
Refer to tables 7.8 and 7.9 for the failure reports generated during the execution of the TC.

7.5.3 HIFI_Dump_Memory

The memory Dump service has type 6, subtype 5. Refer to AD3, section 4.1.2.
On reception of this command the OBS generates one or more memory dump packets (6,6). The structure of the TM dump report is specified in AD4, section 4.4.1.

The ICU performs some checks on the application data; if one check is not passed, an unsuccsessful TC verification packet is issued, with a failure code described in 7.2.1. If the whole chain of checks is OK, the ICU reports a TM (1,1) report.

The application data field of a TM packet can contain up to 1018 bytes. The data field header plus the packet checksum (CRC) need 12 bytes. Then we have the memory ID, the start address, the length and the checksum crc of the data, 8 bytes. In the end, the space left is 998 bytes: a single TM (6,6) packet can carry up to 166 or 249 words of PRAM or DRAM, respectively.
In both cases the used bytes are 996. This implies that a single TC can require more than one TM packets.

The memory words are splitted in 16 bits words, in little endian mode.
If, for instance, the OBS is required to dump 508 words of DRAM, with starting address `0x4FE14`, the following TM (6,6) packets are delivered:

Content of the 1st packet

P#1	0x0104
P#2	0xFE14
P#3	0x00F9 (the first 249 words)
P#4	16 MSb of word at address 0x4FE14
P#5	16 LSb of word at address 0x4FE14
P#500	16 MSb of word at address 0x4FFF0C
P#501	16 LSb of word at address 0x4FF0C
P#502	crc of the first 249 dumped words

Content of the 2nd packet

P#1	0x0104
P#2	0xFF0D
P#3	0x00F9 (the second set of 249 words)

P#4 16 MSb of word at address 0x4FF0D
P#5 16 LSb of word at address 0x4FF0D
P#500 16 MSb of word at address 0x50005
P#501 16 LSb of word at address 0x50005
P#502 crc of the second 249 dumped words

Content of the 3rd packet
P#1 0x0105
P#2 0x0006
P#3 0x000A (the last 10 words to be downloaded)
P#4 16 MSb of word at address 0x50006
P#5 16 LSb of word at address 0x50006
P#22 16 MSb of word at address 0x5000F
P#23 16 LSb of word at address 0x5000F
P#24 crc of the last 10 dumped words

The TC execution completion report is issued after enqueuing of the last dump packet towards the TMTC task. Therefore it is not possible to foresee the execution duration, which is a function of the size of the memory to dump.

Refer to Table 14 for the errors generated during the command execution.

7.5.4 HIFI_Check_Memory

The Memory Check TC has type 6, subtype 9. Refer to AD3, section 4.1.3.

The ICU performs some checks on the application data; if one check is not passed, an unsuccsessful TC verification packet is issued, with a failure code described in Table 30. The liste failures applicable to this service are those indicated with nums: 1,2,3,4.
If the whole chain of checks is OK, the ICU reports a TM (1,1) report.

On reception of this command the OBS calculates the checksum of the memory segment identified by the 3 TC fields: Memory ID, Start Address and Length, and generates one memory check packet (6,10) containing the value of the obtained checksum

The OBS performs some checks on the application data of the received memory dump TC: The check on the Length field is performed against the maximum allowed length calculated on the base of thecommanded start address and the maximum end address defined in the OBS architecture
If this maximum value is exceeded or if it is 0, an execution failure reporrt TM(1,8) with INVALID_MEMLENGTH error code is generated.
If the whole chain of checks is OK, the OBS reports a TM (1,1) report.

The TC execution completion report is issued after enqueuing of the memory check packet (6,10) towards the TMTC task.
Refer to Table 14 for the errors generated during the command execution.

7.6 Service Type 8: Function Management

Functions are handled by Service 8 of AD8.
HIFI uses only the services (8,4) to perform specific actions onboard or to send single commands to the subsystems and service (8,2) to stop an action ongoing onboard.
All the other services (i.e. (8,1) and (8,5)) are not executed.
All acceptance checks listed in section 7.2.1 are applicable to this service (see Table 5 for failure codes and conditions).

The following table contains the list of activities executed onboard upon reception of a (8,4) TC.
The Function ID and activity ID is indicated. In the sixth column the section of the AD3 document containing the detailed specification of the TC structure in the various cases is reported.

Table 16- List of all activities implemented onboard

Description	Type	Sub type	Func ID	Activ ID	AD3 section	Action executed onboard
HIFI_Set_OBS_ID	8	4	1		4.3.1	The Observation ID is set, ready to be reported in all the TM packets.
HIFI_notify_PDU_status	8	4	2		4.3.2	The subsystems status word is updated, see A3.1. It will be monitored in the HK packets.
HIFI_Housekeeping_on	8	4	3	1	4.3.3	The periodic HK acquisition frequency is modified according to the commanded one. The HK acquisition is stopped and restarted with the new frequency. If there is a running measurement onboard, it is aborted before the HK acquisition restarts.
HIFI_non_periodic_hk_FCU	8	4	3	2	4.3.3.2	A non periodic HK report is generated containing a subsmaple of FCU HK
HIFI_non_periodic_hk_LCU	8	4	3	3	4.3.3.3	A non periodic HK report is generated containing a subsmaple of LCU HK
HIFI_read_LCU_mem HIFI_LCU_macro_tuning_hk HIFI_LCU_macro_buffers	8	4	3	4	4.3.3.4	A dump of the commanded number of words from the LCU onboard memory is executed and reported as a NP HK packet.
HIFI_WH_LaserT_check_on HIFI_WV_LaserT_check_on	8	4	4	1,2	4.3.4.1,2	WBS Laser T monitoring function activation
HIFI_H_DHTR_C_check_on HIFI_V_DHTR_C_check_on	8	4	4	3,4	4.3.4.3,4	Mixer deflux heater current monitoring function activation
HIFI_FCU_nonresp_check_on	8	4	4	5	4.3.4.5	FCU HK response check function activation
HIFI_FCU_parameter_scan	8	4	7	1	4.3.6.1	The procedure described in AD6 section 3.8.7.1 is performed, to execute a mixer parameter scan. Before starting the procedure, if there is an onboard running measurement, it is aborted and the FIFOs are reset. At the end of the procedure, a parameter scan report is issued. The overall duration of the procedure depends on the number of commanded steps.
HIFI_scan_diplexer	8	4	7	2	4.3.6.2	The procedure described in AD6 section 3.8.7.2 is performed. Before starting the procedure, if there is an onboard running measurement, it is aborted and the FIFOs are reset. At the end of the procedure, a parameter scan report is issued. The overall duration of the procedure depends on the number of commanded steps.
HIFI_scan_diplexer_if	8	4	7	3	4.3.6.3	The procedure described in AD6 section 3.8.7.3 is performed, to execute a diplexer scan with the acquisition of the spectrometers IFpower at each step. Before starting the procedure, if there is an onboard running measurement, it is aborted and the FIFOs are reset. At the end of the procedure, a parameter scan report is issued. The overall duration of the procedure depends on the number of commanded steps.

HIFI_engineering_scan	8	4	7	4	4.3.6.4	The procedure described in SCR 413 is performed, to execute a measure of the response times of certain FPU components on timescales of < 1sec. Before starting the procedure, if there is an onboard running measurement, it is aborted and the FIFOs are reset. At the end of the procedure, a scan report (more than 1 packet) is issued. The overall duration of the procedure depends on the number of commanded steps.
HIFI_load_vector_scan (nominal and diagnostic)	8	4	8	1	4.3.7.1/2	The procedure described in AD6 section 3.8.8 and in AD12, sect. 1.4 is performed, to load the configuration for the LCU vector scan and LO tuning commands. The overall duration of the procedure is less than 500msec
HIFI_vector_scan	8	4	8	2	4.3.7.3	The procedure described in AD6 section 3.8.8.1 is performed, to execute a scan of the LO-power-amplifier and multiplier parameters simultaneously. Before starting the procedure, if there is an onboard running measurement, it is aborted and the FIFOs are reset. At the end of the procedure, a vector scan report and a Function Status reports are issued. The overall duration of the procedure depends on the number of commanded steps.
HIFI_Tune_LO_using_MXCH HIFI_Tune_LO_using_MXCV	8	4	8	3,4	4.3.7.4,5	The procedure described in AD6 section 3.8.8.2 is performed, to tune the LO level, by executing a vector scan and selecting the final LO configuration on the base of the obtained mixer currents. Before starting the procedure, if there is an onboard running measurement, it is aborted and the FIFOs are reset. At the end of the procedure, a vector scan report and a LO tuning reports are issued. The overall duration of the procedure depends on the number of commanded steps.
Tune HRS	8	4	9	1	4.3.8.1	The procedure described in AD6 section 3.8.9.1 is performed, to execute the HRS attenuators setting. Before starting the procedure, if there is an onboard running measurement, it isa aborted and the FIFOs are reset. The procedure is in two steps. For each step a full HRS science data packetisation is provided as well as a tune report. The overall duration of the procedure is <=4 sec starting from the TC acceptance..
Tune WBS	8	4	9	2	4.3.8.2	The procedure described in AD6 section 3.8.9.2 is performed, to execute the WBS attenuators setting. Before starting the procedure, if there is an onboard running measurement, it isa aborted and the FIFOs are reset. The procedure is in three steps. For each step a full WBS science data packetisation is provided. Two tune reports are provided at the end of the last two procedure steps. The overall duration of the procedure is <=10 sec starting from the TC acceptance.
Tune Mixer magnet current use HRS	8	4	9	3	4.3.8.3	The procedure described in AD6 section 3.8.9.3.1 is performed, to execute the mixer magnet tuning, using HRS data. Before starting the procedure, if there is an onboard running measurement, it isa aborted and the FIFOs are reset. At the end of the procedure a tune report is issued. The overall duration of the procedure depends on the number of commanded steps.
Tune Mixer magnet current use WBS	8	4	9	4	4.3.8.4	The procedure described in AD6 section 3.8.9.3.2 is performed, to execute the mixer magnet tuning, using HRS data. Before starting the procedure, if there is an onboard running measurement, it isa aborted and the FIFOs are reset. At the end of the procedure a tune report is issued.
Spectroscopy_total_power. The overall duration of the	8	4	11	1	4.3.11.2	The procedure described in AD6 section 3.8.11.3 is performed, to execute a total power spectroscopy measurement, using the measurement configuration

procedure depends on the number of commanded steps.						parameters already stored onboard. Before starting the procedure, if there is an onboard running measurement, it is aborted and the FIFOs are reset. The spectrometers commanding necessary to perform the measurement is made by the onboard command interpreter (i.e. VM). During the measurement, depending on the configuration parameters, several HRS and WBS packetisations can be issued.
HIFI_Configure_spectroscopy	8	4	11	16	4.3.11.1	The commanded Total power spectroscopy measurement configuration parameters are uploaded onboard.
HIFI_Configure_FCU	8	4	12	1	4.3.5.1	The commanded FCU configuration parameters are sent out to FCU through the LS I/F. The onboard table is reported in A3.3, Table 33. Before starting the commands sending, a Configure report is provided containing the echo of the commanded parameters, see Table 18.
HIFI_Configure_FCU_power	8	4	12	11	4.3.5.1	
HIFI_Config_HRS_H_att_lo	8	4	12	7	4.3.5.2	The commanded HRS-H attenuators settingsare sent out to HRS through the LS I/F. The onboard table is reported in A3.3, Table 33 Before starting the commands sending, a Configure report is provided containing the echo of the commanded parameters, see Table 18.
HIFI_Config_HRS_H_blocks	8	4	12	8	4.3.5.2	The commanded HRS-H configuration parameters are sent out to HRS-H through the LS I/F. The onboard table is reported in in A3.3, Table 33 Before starting the commands sending, a Configure report is provided containing the echo of the commanded parameters, see Table 18.
HIFI_Config_HRS_V_att_lo	8	4	12	9	4.3.5.2	The commanded HRS-V attenuators settings sent out toHRS through the LS I/F. The onboard table is reported in in A3.3, Table 33
HIFI_Config_HRS_V_blocks	8	4	12	10	4.3.5.2	The commanded HRS-V configuration parameters sent out to HRS-V through the LS I/F. The onboard table is reported in A3.3, Table 33 Before starting the commands sending, a Configure report is provided containing the echo of the commanded parameters, see Table 18.
HIFI_Configure_WBS_H	8	4	12	4	4.3.5.3	The commanded WBS-H configuration parameters are sent out to WBS-H through the LS I/F. The onboard table is reported in A3.3 Table 39 Before starting the commands sending, a Configure report is provided containing the echo of the commanded parameters, see Table 18.
HIFI_Configure_WBS_V	8	4	12	5	4.3.5.3	The commanded WBS-V configuration parameters are sent out to WBS-V through the LS I/F. The onboard table is reported in A3.3 Table 39. Before starting the commands sending, a Configure report is provided containing the echo of the commanded parameters, see Table 18.
HIFI_Configure_LCU (diagnostic and Nominal)	8	4	12	31 30	4.3.5.4 4.3.5.5	The commanded LCU configuration parameters sent out to FCU through the LS I/F. The onboard table is reported in A3.3 Table 39. Before starting the commands sending, a Configure report is provided containing the echo of the commanded parameters, see Table 18.
HIFI_Conf_safe_LCU	8	4	12	30	4.3.5.6	
HIFI_Conf_LCU_table	8	4	12	32	4.3.5.8	
HIFI_Conf_LCU_internal	8	4	12	13	4.3.5.7	

HIFI_Conf_LSU	8	4	12	14	4.3.5.9	
HIFI_HL_switch_off	8	4	12	12	4.3.5.10	
HIFI_Single_Command: HIFI_CPR_Chopper_Rot HIFI_reset_WBS_H HIFI_reset_WBS_V HIFI_HL_Standby HIFI_ HL_NORMAL HIFI_ HL_Reset HIFI_HF_CH1_DHTR_C HIFI_HF_CV1_DHTR_C HIFI_HL_switchon HIFI_HL_Diagnostic HIFI_HL_LSU_power HIFI_HL_LSU_subband HIFI_HL_store_tm HIFI_HL_set_FSW1 HIFI_HL_set_FSW2 HIFI_HL_heater HIFI_CPR_Chopper_Rot	8	4	12	1 26 26 26 29 29 29 27 28 29 29 29 29 29 29 29 29 29 26	4.3.5.11	The single command contained in the TC is forwarded to the subsystems using the LS I/F. The command in enqueued on the LS high priority queue, in order to be sent out with a higher priority with respect to the already enqueued periodic HK requests/prepare commands. The single command is transmitted within 100msec from the TC acceptance.

7.6.1 Perform activity – Set Observation ID (FID 1)

The activity is commanded with the TC described in AD3, sect 4.3.1.
The TC acceptance report and the TC execution Report are generated within 500msec from the TC ingestion.
All Acceptance error codes reported in Table 5 are applicable.

The OBS sets the Observation ID global variable to the commanded value. The Observation ID will remain the same until a new HIFI_Set_OBS_ID TC is sent with a different value.
The current Observation ID is reported in the periodic HK packet.

7.6.2 Perform activity – Notify PDU status (FID 2)

The activity is commanded with the TC described in AD3, sect 4.3.2.
The TC acceptance report and the TC execution Report are generated within 500msec from the TC ingestion.
All Acceptance error codes reported in Table 5 are applicable.

The command is executed by changing the global variable onboard that maps the status of the six subsystems.
Other variables necessary to map the spectrometers status are updated as well.
For each spectrometer that was off and has now been switched on, the corresponding HW FIFO is reset.
No error generation is foreseen.

7.6.3 Perform activity – LCU memory handling (FID 3)

The pupose of this set of command is to read a certain part of the LCU-RAM and to get the result in the HK-stream. Refer to AD3, section 4.3.3.4 for the TC detailed description.
Following SCR 1018 an additional procedure has been implemented onboard, to execute the new TC HIFI_check_LCU_memory (described in AD3, section 4.3.14.8).

The procedure allows to compute the LCU checksum and compare it with an expected one. The expected checksum is passed in the TC parameters. Also a wait_time (in msec), needed for the LCU to compute the checksum, is passed in the command's parameters.
If the two checksum do not match the following two actions are taken onboard:
1) a TM packet (5,4) is issued containing two parameters: the expected and the read checksum (in that order);
2) a flag (LCU_non_interaction_flag) is set.

When the LCU_non_ interaction_flag is set, all the requests to the LCU are aborted: this implies that the frequency switch command is rejected, since it makes use of the LCU.
Once set, the flag cannot be reset. During its execution the command suspends the HK requestes to the LCU: these are reactivated at command completion. However note that if the command is aborted the LCU HK needs to be explicitly reactivated using a proper set_HK_on command. The command duration is 1016 + wait_time msec (including a 10 msec margin for CPU load). Add one second if the HK requestes to the spectrometers are ON. The LCU_non_interaction flag is reported in the LSB of word 28 of the ICU HK.

7.6.4 Perform activity – Limit checking (FID 4)

The activity is commanded with the TCs described in AD3, sect 4.3.4.
The TC acceptance report is always generated within 500msec from the TC ingestion.
All Acceptance error codes reported in Table 5 are applicable.
Refer to Table 29 and Table 30 for the error code of the OBS runtime errors and execution failure reports that can be generated during the activity (sections related to task HK_ASK).

General remark: the limit checking and the non reponse checks can be switched ON or OFF. By default, at boot, all the monitoring activities are switched ON. The default limit at startup for DHTR current is 63168 (= 35 mA). Default NBreach=10. The default limit for the WBS temperature is 1638 decimal raw, Nbreach=10.

The ON/OFF status is reported in the ICU HK and specifically in word 27 of the ICU HK with the format described in SCR 882 and reported below:
bit 6 (msb) HK_check_wbsh
bit 5 HK_check_wbsv
bit 4 HK_check_htrv
bit 3 HK_check_htrh
bit 2 HK_error_countFCU_8C
bit 1 HK_error_countFCU_8D
bit 0 (lsb) HK_error_countFCU_8F

7.6.4.1 WBS laser temperature limit checking

The commanded activity is described in AD5 sect. 3.8.4.2.
For each spectrometer the Activity is commanded by two TCs:

- *HIFI_WH_Laser_T_check_on*: to start the WBS HK data monitoring activity and pass the activity parameters (upper limit and number of breaches that trigger the error).

- *HIFI_WH_Laser_T_check_off*: to stop the WBS HK data monitoring activity

The TC execution report is generated immediately after the global variables controlling the status of the limit checking service have been updated. In any case after the TC acceptance report and within 500msec from the Tc ingestion.

For the check to be really carried out all the following three conditions must be met:
3) the check is switched on
AND
2) the HK requests to the WBS are on
AND

3) the laser is switched on

When the check is carried out the *HWH_laser_T* parameter reported in the HK packet is checked against the commanded Upper limit. As soon as it exceeds this limit for a number of times greater than the N-breach commanded parameter the following actions are taken:

- The corresponding laser is switched OFF

- An event report is generated (see **Error! Reference source not found.** for the Event IDs. The parameter content of the generated events is described in AD4, sect. 4.3.3.6.7).

The *HWH_laser_T* parameter is monitored at the periodic HK acquisition rate.

7.6.4.2 Deflux Heater current limit checking

For each spectrometer the Activity is commanded by two TCs:

- *HIFI_H_DHTR_C_check_on*: to start the mixer deflux heater monitoring activity and pass the activity parameters (upper limit and number of breaches that trigger the error).

- *HIFI_H_DHTR_C_check_off*: to stop the mixer deflux heater monitoring activity

The TC execution report is generated immediately after the global variables controlling the status of the limit checking service have been updated. In any case after the TC acceptance report and within 500msec from the TC ingestion.

Additionally, when a *Configure_FCU_power TC(* that switches on a board) is received, the corresponding check is automatically switched on.

For the check to be really carried out the following two conditions must be met:

1) the check is switched on
AND
2) the HK requests to the FCU are on

When the check is carried out the *AH1_DHTR_C* parameter reported in the HK packet is checked against the commanded Upper limit. As soon as it exceeds this limit for a number of times greater than the N-breach commanded parameter the following actions are taken:

- The corresponding board is switched OFF

- An event report is generated (see **Error! Reference source not found.** for the Event IDs. The parameter content of the generated events is described in AD4, sect. 4.3.3.6.7).

- The limit checking activity is switched off

The *AH1_DHTR_C* parameter is monitored at the periodic HK acquisition rate.

7.6.4.3 FCU Heater check functions

The Activity is commanded by two TCs:

- *HIFI_FCU_nonresp_check_on*: to start the mixer deflux heater monitoring activity and pass the activity parameters (upper limit and number of breaches that trigger the error).

- *HIFI_FCU_nonresp_check_off*: to stop the heater monitoring activity

The TC execution completion report is generated immediately after the global variables controlling the status of the limit checking service have been updated. In any case after the TC acceptance report and within 500msec from the TC ingestion.

Additionally when a *Configure_FCU_power* TC is received that switches on a board, the corresponding check is automatically switched on.

For the check to be really carried out the following two conditions must be met:

1) the check is switched on
   AND
2) the HK requests to the FCU are on

The 8 MS bits of the FCU HK request for the H-chain are 0x8c, for the V-chain are 0x8d and for the chopper boards are 0x8f. When the check is on the HK data of the three types is monitored. The failure is detected in case of missing data in *Nbreach* (Nbreach is a commanded parameter included in the TC) successive cycles of HK collection. In case of failure in one of the three chains the following actions are taken:

- The corresponding board is switched OFF

- An event report is generated (see **Error! Reference source not found.** for the Event IDs. The parameter content of the generated events is described in AD4, sect. 4.3.3.6.9).

- The check is switched off

The parameters are monitored at the periodic HK acquisition rate.

Table 17 - OBS generated event reports for Autonomy errors (Event ID = SID listed in table)

ERROR Name	Short Mnemonic	SID Event ID	Notes
ERR_HK_ASK_ERR_8C	ERR_8C	0xb005	Mixer chain H was switched off due to too non-response
ERR_HK_ASK_ERR_8D	ERR_8D	0xb006	Mixer chain V was switched off due to too non-response
ERR_HK_ASK_ERR_8F	ERR_8F	0xb007	Chopper board was switched off due to too non-response
ERR_HK_ASK_HTRH_LIMIT	HTRH_LIM	0xb001	Mixer chain H was switched off due to too high Heater-current
ERR_HK_ASK_HTRV_LIMIT	HTRV_LIM	0xb002	Mixer chain H was switched off due to too high Heater-current
ERR_HK_ASK_WBSH_LIMIT	WBSH_LIM	0xb003	WBS-H laser was switched off due to too high HWH_Laser_T
ERR_HK_ASK_WBSV_LIMIT	WBSV_LIM	0xb004	WBS-V laser was switched off due to too high HWV_Laser_T

7.6.5 Perform activity – Configure Subsystems (FID 5/12)

All the HIFI subsytems can be configured via TC.
TheTC to configure FCU is described in AD3 sect. 4.3.5.1
The TC to configure LCU diagnostic is described in AD3 sect. 4.3.5.4 (see also AD12, section 1.3).
The TC to configure LCU nominal is described in AD3 sect. 4.3.5.5 (see also AD12, section 1.3).

The TC to configure LCU safe is described in AD3 sect. 4.3.5.6
The TC to configure LCU internal is described in AD3 sect. 4.3.5.7
The TC to configure LCU table is described in AD3 sect. 4.3.5.8
The TC to configure HRS-H/V is described in AD3 sect. 4.3.5.2
The TC to configure WBS-H/V is described in AD3 sect. 4.3.5.3
The TC acceptance report is generated within 500msec from the TC ingestion. All Acceptance error codes reported in Table 5 are applicable.

The command is executed by sending out to the subsystems through the Low Speed Interface all the 32 bit parameters passed by the TC.

Note on HIFI_Configure_LCU_nominal: In nominal mode the bias voltages can not be directly commanded. Instead, a series of LScommands must be sent to the LCU in a well-defined block (macro).
The final command of the macro contains a checksum-parameter (the sum of all thecommands masked with 0xFFFFF should equal 0x00000). This parameter is calculated onboard OBS and included in the less significant word of the LS command to LCU.

Note on HIFI_Configure_LCU_power: when a *Configure_FCU_power* TC is received that switches on a board, the corresponding limit check is automatically (re-)switched on.

In addition to the commands sent out through the LS interface, the output of this activity is the Function status report corresponding to the received Configure Subsystem TC.
The Function Status Reports contain a copy of all configuration parameters in the received TC (see Table 18 for a reference to the documents where the reports are defined).
The successful TC execution report is generated as soon as all configuration command are enqueued on the LS high priority queue (see AD2), in any case after the Function status report generation and within 500 msec from the TC ingestion.

There is only one possible execution failure:

ERROR Name	Short Mnemonic	Error Code	Notes
NOK_CMDEX_ERROR_LS_HP_QUEUE	CMDEX_LSHP_Q	0x0627	Failure in sending out to LS task one of the commanded configuration parameters. No execution failure parameters are provided.

Table 18 - Configure reports

Description	Type	Sub type	Funct ID	Activ ID	AD5 section	Report content
HIFI_Configure_FCU_report	8	6	12	1	3.6.3.1	The report is the echo of the received TC.
HIFI_Configure_FCU_power_report	8	6	12	11	3.6.3.2	The report is the echo of the received TC.
HIFI_Conf_HRS_H_att_lo_report	8	6	12	7	3.6.3	The report is the echo of the received TC.
HIFI_Conf_HRS_H_blocks_report	8	6	12	8	3.6.3	The report is the echo of the received TC.
HIFI_Conf_HRS_V_att_lo_report	8	6	12	9	3.6.3	The report is the echo of the received TC.
HIFI_Conf_HRS_V_blocks_report	8	6	12	10	3.6.3	The report is the echo of the received TC.
HIFI_Configure_WBS_H_report	8	6	12	4	3.6.3	The report is the echo of the received TC.
HIFI_Configure_WBS_V_report	8	6	12	5	3.6.3	The report is the echo of the received TC.
HIFI_Configure_LCU_nominal_report	8	6	12	30	3.6.3	The report is the echo of the received TC.

HIFI_Configure_LCU_diag_report	8	6	12	31	3.6.3	The report is the echo of the received TC.
HIFI_Configure_LCU1a-7b_report	8	6	12	12-25	3.6.3	The report is the echo of the received TC.
HIFI_Configure_LCU_table	8	6	12	20		The report is the echo of the received TC.
HIFI_Config_spectroscopy_report	8	6	11	16	3.6.2	The report is the echo of the received TC.

7.6.6 Perform activity – Send Single Command (FID 5/12)

The FID of this activity is equal to 12. All AIDs listed below (Table 19) are accepted.

The TC acceptance report is issued within 500ms from the TC ingestion.
All Acceptance error codes reported in Table 5 are applicable.

Table 19 List of accepted Single Commands

Single command	AID
HIFI_reset_WBS_H	26
HIFI_reset_WBS_V	26
HIFI_ HL_SWITCH_OFF	26
HIFI_ HL_STANDBY	29
HIFI_ HL_NOMINAL	29
HIFI_ HL_RESET	29
HIFI_set_ HF_CH1_DHTR_C	27
HIFI_set_ HF_CV1_DHTR_	28
HIFI_ HL_SWITCH_ON	29
HIFI_HL_Diagnostic	29
HIFI_HL_LSU_power	29
HIFI_HL_LSU_subband	29
HIFI_HL_store_tm	29
HIFI_HL_set_FSW1	29
HIFI_HL_set_FSW2	29
HIFI_HL_heater	29
HIFI_CPR_Chopper_Rot	26

In addition to the command sent out through the LS interface, the output of this activity is the Function status report corresponding to the received Single_Command TC.
The Function Status Reports contain a copy of the parameter received in the TC. Refer to AD5, sect. 3.6 for the report structure description.
The successful TC execution report is generated as soon as the single command is enqueued on the LS high priority queue (see AD2), in any case after the Function status report generation and within 500 msec from the TC ingestion.

There is only one possible execution failure:

ERROR Name	Short Mnemonic	Error Code	Notes
NOK_CMDEX_ERROR_LS_HP_QUEUE	CMDEX_LSHP_Q	0x0627	Failure in sending out to LS task one of the commanded configuration parameters. No execution failure parameters are provided.

7.6.7 Perform activity – Parameter Scan (FID 7)

7.6.7.1 FCU Parameter scan

The activity is commanded with the TC described in AD3, sect 4.3.7.1.
The TC acceptance report is generated within 500msec from the TC ingestion.

The FCU_parameter scan report is generated as soon as all FCU commands/HK requests belonging to the last step are enqueued on the LS high priority queue (see AD2). Therefore the report will be generated with a time delay with respect to the TC acceptance report approximately equal to

HIF_NVOLTAGE*HIF_NMAGNET*(HIF_STEP_TIME +25msec)
= Total number of commanded steps*(step time+ time to send all HK/cmd requests).

Refer to AD3, sect 4.3.7.1. for the TC parameters identification and to AD4, section 4.9.6.2 for the report structure.

The successful TC execution report is generated immediately after the scan report.
The following failure detections are implemented, all generating execution failure reports (1,8):

Table 20 List of possible execution failures for FCU parameter scan

ERROR Name	Short Mnemonic	Error Code	Notes
EXF_CMDEX_PAR_SCAN_WRONG_STEP_NUMBER	CMDEX_PSC_STEP	0x0605	The calculated total number of steps (HIF_NVOLTAGE*HIF_NMAGNET) is greater than the maximum allowed (80) 2 parameters: HIF_NVOLTAGE, HIF NMAGNET.
EXF_CMDEX_PAR_SCAN_WRONG_NVOLTAGE	CMDEX_PSC_VOLT	0x0606	Type (8,4) FCU parameter scan: execution failure due to wrong HIF_NVOLTAGE parameter (<1)
EXF_LS_HDL_FCU_OFF	LSHDL_FCU_OFF1	0x2001	Sweep Diplexer procedure in LS_HDL task: execution failure due to FCU off
EXF_LS_HDL_ERROR_LS_HP_QUEUE	LSHDL_LSHPQ	0x2006	FIFO Put on LS HP queue failure: no possibility to trnasmit to LS task one of the command of the procedure. The procedure is stopped. Applicable to all scan procedures and to LO tuning.
EXF_LS_HDL_GEN_REP_FAIL	LSHDL_GEN_REP	0x2005	Generate report error: no final (or step) report can be generated at the end of one of the scan functions.

Refer to Table 29 for the error code of the OBS runtime errors that can be generated during the activity.

7.6.7.2 Diplexer Scan

There are two possible diplexer scans: with or without the HRS If power acquisition. Both activities are described in AD6, section 3.8.6.2/3.

1. Diplexer Scan without IF Power

The activity is commanded with the TC described in AD3, sect 4.3.6.2.
The TC acceptance report is generated within 500msec from the TC ingestion.

The Scan Report is generated as soon as all FCU commands/HK requests belonging to the last step and enqueued on the LS high priority queue (see AD2) have been processed by the LS task. Therefore the report is issued with a time delay with respect to the TC acceptance report greater than

HIF_NDIPLEXER*(HIF_STEP_TIME +30msec) (+1sec)

= Total number of commanded steps* (step time+time necessary to send out all cmd/hk requests) (+ start_meas_overhead).

The start_meas_overhead of 1 sec shall be added when the commanded activity has aborted another activity already running onboard, to take into account the maximum delay in starting the TC execution (the delay is due to the WBS transfer time). This initial overhead can be minimised by disabling the periodic HK acquisition from spectrometers.
.
The successful TC execution report is generated immediately after the scan report.
The maximum number of steps is 61. In case a greater number of steps is commanded, a TC execution failure report is issued and the scan is not executed.
Refer to AD3, sect 4.3.7.2. for the TC parameters identification and to AD4, section 4.9.6.4 for the report structure.

2. Diplexer scan with HRS IF Power.

The activity is commanded with the TC described in AD3, sect 4.3.6.3.
The TC acceptance report is generated within 500msec from the TC ingestion.

The Scan Report is generated as soon as all FCU commands/HK requests belonging to the last step and enqueued on the LS high priority queue (see AD2) have been processed by the LS task.. Therefore the report will be issued with a time delay with respect to the TC acceptance report greater than

HIF_NDIPLEXER *(HIF_STEP_TIME + 132msec) + 42msec + 30msec (+ 1sec).

= Total number of commanded steps* (step time+time necessary to send out all cmd/hk requests + HRS integration time) + HRS transfer time + fixed overhead (+start_meas_overhead).

The start_meas_overhead of 1 sec shall be added when the periodic HK acquisition is running to take into account the maximum delay in starting the TC execution (the delay is due to the WBS transfer time).

The maximum number of steps is 20.
Refer to AD3, sect 4.3.7.3. for the TC parameters identification and to AD4, section 4.9.6.3 for the report structure.
The successful TC execution report is generated immediately after the scan report.

The following failure detections are implemented, all generating execution failure reports (1,8):

Table 21 List of possible execution failures for diplexer scan

ERROR Name	Short Mnemonic	Error Code	Notes
EXF_HS_HDL_ERROR_LS_HP_QUEUE	HSHDL_LSHPQ	0x0809	FIFO Put on LS HP queue failure: no possibility to trnasmit to LS task one of the command of the procedure. The procedure is stopped. Applicable to all tuning procedures.
EXF_HS_HDL_HRS_SUB_OFF	HSHDL_HRS_OFF	0x0810	A diplexer scan with IF power is requested for a switched off spectrometer. The procedure is not started
EXF_HS_HDL_FCU_SUB_OFF	HSHDL_FCU_OFF	0x0811	A diplexer scan with IF power is requested but FCU is off. The procedure is not started
EXF_HS_LIB_HRS_SUB_OFF	HSLIB_HRS_OFF	0x0906	A measurement involving HRS has been requested, but the spectrometers are OFF. No measurement is started.
EXF_HS_LIB_QUEUE	HSLIB_QUEUE	0x0912	FIFO Put on HS HDL Queue failure:no possibility to notify to HS HSLthat one of the tuning procedure is cstarted. The procedure is aborted. Applicable to all tuning procedures.
EXF_HS_LIB_FCU_SUB_OFF	HSLIB_FCU_OFF	0x0913	A measurement involving FCU has been requested, but the spectrometers are OFF. No measurement is started (applicable to scans and to mixer magnet tuning)
EXF_HS_LIB_DIPSCAN_NUM_STEP	HSLIB_DIPSCAN_NS	0x0914	The commanded number of steps for the diplexer scan is greater than the maximum allowed. The scan is not staretd.
EXF_HS_LIB_ERROR_LS_HP_QUEUE	HSLIB_LSHPQ	0x0905	FIFO Put on LS HP queue failure: no possibility to trnasmit to LS task one of the command of the procedure. The procedure is stopped. Applicable to all tuning procedures and to peakup procedures.
EXF_LS_HDL_ERROR_LS_HP_QUEUE	LSHDL_LSHPQ	0x2006	FIFO Put on LS HP queue failure: no possibility to trnasmit to LS task one of the command of the procedure. The procedure is stopped. Applicable to all

			scan procedures and to LO tuning.
EXF_LS_HDL_FCU_SUB_OFF	LSHDL_FCU_OFF_3	0x2007	Diplexer scan procedure in LS HDL: no execution started due to FCU off.
EXF_LS_HDL_DIPSCAN_NUM_STEP	LSHDL_DSCAN_NS	0x200c	Wrong number of steps in diplexer scan command.
EXF_LS_HDL_PUT_LS_HDL_QUEUE	LSHDL_LSHDL_Q	0x2014	FIFO put on LS HDL queue failure: it is not possible to execute the following steps of the procedure. Applicable to FCU parameter scan, vector and diplexer scans.
EXF_LS_HDL_GEN_REP_FAIL	LSHDL_GEN_REP	0x2005	Generate report error: no final (or step) report can be generated at the end of one of the scan functions.

Only one OBS runtime error (severity level 2) can be generated directly in the diplexer scan functions. It is reported in the table below. In case of Diplexer scan with If Power all the runtime errors related to the Spectrometers data handling are applicable too. In both cases the runtime errors generated by the LS task in case of failures with the LS link are applicable as well.

ERROR Name	Short Mnemonic	Error Code	Notes
ERR_HS_HDL_WRONG_SPECT_AID	WSPEC_AID	0x0800	Current value of AID spectroscopy different from the one necessary to the running function

7.6.7.3 Engineering Scan

The FP subsystem have identified a need for an engineering routine to measure the response times of certain FPU components on timescales of < 1sec. This is done with the Engineering scan TC. The required functionality has been specified in SCR 413, and the final agreed specification is reported below:

1) gather a sequence of N1 HK values from 1 up to 3 specified HK channels at a specified sampling interval , which must be >=3 msec (3 msec is the fixed time that task ls takes to process each hk request); TC packets with a "too high" sampling interval are rejected. The error on the sampling period is < 1 msec.
2) send a specified command: the single command is separated in time from the previous and subsequent hk requests by the sampling period
3) gather a sequence of 1 < N2 < 1000 HK values from up to 3 specified HK channels at a specified sampling interval of 1ms or more,
4) packetise the HK information and send.

During execution of this command normal HK and spectroscopy functions could be suspended.

The activity is commanded with the TC described in AD3, sect 4.3.6.4.
The TC acceptance report is generated within 500msec from the TC ingestion.

The command parameters include:
a) the HK channels to be gathered
b) the sampling interval, 3ms =< t < 1s,: HIF_interval is the time between two successive hk requests: if all the three HIF_HK_address_X parameters of the TC are valid, the sampling period of HK_address_1 is 3*HIF_interval, and the same for HK_address_2 and HK_address_3;
c) the number of preceding HK acquisitions, (HIF_N_samples_1), 0 < N1 < 150
d) the single command (unit & channel): HIF_command.
e) the number of subsequent HK acquisitions (HIF_N_samples_2) 1 < N2 < 1000

The acquisition phase preceeding the single command will last 3*HIF_interval*HIF_N_samples_1 ms (for example 3*3*150 = 1350 ms if HIF_N_samples_1 is 150 and HIF_interval is 3).
In other words, it is necessary to set to 0x0FFF two of the three HK_addresses in the TC in order to have the third sampled at the highest frequency.

The adopted HIFI_engineering_scan_report structure is:

Position	Length	Parameter	Value

16	16	Structure ID	270
18	32	Observation ID	
22	32	Building block ID	
26	16	NumberOfPacketsInScan	
28	16	PacketNumberInFrame	
30	16	HIF_interval[ms]	
32	16	HIF_HK_address_1	
34	16	HIF_HK_address_2	
36	16	HIF_HK_address_3	
38	32	HIF_command	
42	16	HIF_N_samples_1	
44	16	HK_value_1	
46	16	HK_value_2	
48	16	HK_value_3	
50	16	HIF_N_samples_2	
52	16	HK_value_1	
54	16	HK_value_2	
56	16	HK_value_3	

In case an HK-address in the TC is void (0x0FFF) the corresponding HK-will be left out

The procedure expected duration is:
HIF_interval*number_of_HK*(HIF_N_samples_1+ HIF_N_samples_2)+10 + initial overhead (<=1sec)

7.6.8 Perform activity – Local Oscillator Functions (FID 8)

7.6.8.1 HIFI_Load_vector_scan nominal/diagnostic

The vector scan function and the LO-tune functions make use of the same set of parameters. These parameters are uploaded with the HIFI_Load_vector_scan commands.
The activity is commanded with the TCs described in AD3, sect 4.3.7.1/2, and in in AD12, sect. 1.4..
The TC acceptance report is generated within 500msec from the TC ingestion.

The load vector scan command is used to specify the LS commands to be issued, their parameters initial values and incremental steps.
The configuration loaded with a HIFI_Load_vector_scan TC can be used only one time (either for a vector scan or for a LO tuning). To reperform a scan or a tuning it is necessary to re-send a HIFI_Load_vector_scan TC to reload the configuration. If the configuration is not loaded, when either for a vector scan or a LO tuning are received, the commanded TC is not executed and a report is generated, see Table 22.

7.6.8.2 HIFI_Vector_scan

The activity is described in AD6 section 3.8.8.1 and is commanded with the TCs described in AD3, sect 4.3.7.3.
The TC acceptance report is generated within 500msec from the TC ingestion.

The Scan Report is generated as soon as all LCU commands/HK requests belonging to the last step and enqueued on the LS high priority queue (see AD2) have been processed by the LS task. Therefore the report is issued with a time delay with respect to the TC acceptance report greater than

HIF_NSteps*(HIF_STEP_TIME + 51msec) (+1sec)
= Total number of commanded steps* (step time+time necessary to send out all hk requests)
(+start_meas_overhead).

The start_meas_overhead of 1 sec shall be added when the commanded activity has aborted another activity already running onboard, to take into account the maximum delay in starting the TC execution (the delay is due to the WBS transfer time).

The successful TC execution report is generated immediately after the vector scan report.
The maximum number of steps is 10: this limit is set to fit the vector scan report in one only TM packet.
In case a greater number of steps is commanded, a TC execution failure report is issued and the scan is not executed.
Refer to AD4, section 4.9.6.10 for the scan report structure.
The final setting of the vector scan is reported by means of a Function status report, described in AD4, section 4.9.6.10. The Report has Type, subtype = 8, 6 and the SID is obtained from the HIFI_load vector scan SID minus 16.

7.6.8.3 HIFI_Tune_LO_Using_MXCH/MXCV

The activity is described in AD6 section 3.8.8.2 and is commanded with the TCs described in AD3, sect 4.3.7.4/5.
The TC acceptance report is generated within 500msec from the TC ingestion.
The procedure executes a LO vector scan recording the mixer current as a function of the vector scan step. As soosn as the mixer current target value is reached, the scan is interrupted and a set of interpolated values for LCU parameters is comuputed. The algorithm implemented onboard is described in SCR 991.

The procedure has two outputs: a vector scan report and a LO tuning report (function status report). Both are generated after all LCU commands/HK requests belonging to the last step and enqueued on the LS high priority queue (see AD2) have been processed by the LS task.
Therefore the reports are issued with a time delay with respect to the TC acceptance report greater than

HIF_NSteps*(HIF_STEP_TIME + 60msec) (+1sec)
= Total number of commanded steps* (step time+time necessary to send out all hk requests)
(+start_meas_overhead).

The start_meas_overhead of 1 sec shall be added when the commanded activity has aborted another activity already running onboard, to take into account the maximum delay in starting the TC execution (the delay is due to the WBS transfer time).

The successful TC execution report is generated immediately after the tuning report.
The maximum number of steps is 10: this limit is set to fit the vector scan report in one only TM packet.
In case a greater number of steps is commanded, a TC execution failure report is issued and the scan is not executed.
Refer to AD4, section 4.9.6.10 for the scan report structure.
The final setting of the LO tuning is reported in a LO tuning Report report, described in AD4, section 4.9.6.10. The Report has Type, subtype = 8, 6 and the SID is obtained from the HIFI_load vector scan SID minus 16.

7.6.8.4 Local Oscillator functions error conditions

The following execution failure detections are implemented, all generating reports (1,8):

Table 22 LCU functions error codes

EXF_LS_HDL_VEC_SCAN_TABLE	LSHDL_BAD_VSTAB	0x2004	Invalid Length of the table onboard used to execute the vector scan
EXF_LS_HDL_GEN_REP_FAIL	LSHDL_GEN_REP	0x2005	Generate report error: no final (or step) report can be generated at the end of one of the scan functions.
EXF_LS_HDL_ERROR_LS_HP_QUEUE	LSHDL_LSHPQ	0x2006	FIFO Put on LS HP queue failure: no possibility to trnasmit to LS task one of the command of the procedure. The procedure is stopped. Applicable to all scan procedures and to LO tuning.
EXF_LS_HDL_VSCAN_REP_GET_HK_BLOCK	LSHDL_VSCAN_HKB	0x2008	Get block from HK Pool failure: no vector scan report can be produced. The procedure has been completed but the data cannot be downloaded.

EXF_LS_HDL_VSCAN_REP_PUT_HK_QUEUE	LSHDL_VSCAN_HKQ	0x2009	FIFO put on HK TM queue failure: : no vector scan report can be produced. The procedure has been completed but the data cannot be downloaded.
EXF_LS_HDL_VSCAN_OVFL	LSHDL_VSCAN_OVF	0x200A	Error in parameters of vector scan.
EXF_LS_HDL_VSCAN_SYSOFF	LSHDL_VSCAN_SOFF	0x200b	FCU or LCU are switched off. Vector scan cannot be xecuted.
EXF_LS_HDL_VEC_SCAN_CONF_EXPIRED		0x2016	Missing Load vector scan configuration
ERR_LS_HDL_VEC_SCAN_INTERNAL		0x2017	
EXF_LS_HDL_VEC_SCAN_INTERNAL		0x2018	
EXF_LS_HDL_VSCAN_REPORT_PROBL		0x2019	
ERR_LS_HDL_LO_TUNE_FOUND_NO_BRACKET		0x2020	Lo tune maximum number of steps reached without founding any bracketing pair.
EXF_LS_HDL_LCU_NOT_NORM		0x2021	
EXF_LS_HDL_LAST_INVALID		0x2022	
ERR_LS_HDL_LCU_CRC_MISMATCH		0x2023	

The following OBS Runtime errors can be detected:

ERR_LS_HDL_WRONG_SID	LSHDL_SIDERR	0x2015	Error in SID of Load Vector scan command

Refer to Table 29 for the error code of all the other the OBS runtime errors that can be generated during the activity.

7.6.9 Perform activity – Measure LCU_IV_Curve (FID 15)

The activity is commanded with the TC described in AD3, sect 4.3.13, and in AD12, section 1.6.
The TC acceptance report is generated within 500msec from the TC ingestion.

Currently an LO IV curve command can only be executed properly when the LO is first put into diagnostic mode.
The procedure implemented on board is based on specifications listed in SCR 979, and reported below:

1) set LO in diagnostic mode
2) measure IV requested curve
3) set LO back to normal mode
4) check LO HK to verify that LO is in normal mode
5) send HIFI_LCU_IV_Report

The function activities will be completed (and the report will be generated) after a time delay with respect to the TC acceptance report greater than

(HIF_LCU_IV_delay + 33msec+17msec) = commanded wait time + 10*(single hk request time)+2*(LCU commands send times)+time interval before LCU status verify..

Refer to AD3, sect 4.3.11. for the TC parameters identification and to AD4, section 4.9.6.5 for the report structure.
The successful TC execution report is generated immediately after the scan report.

The following failure detections are implemented, all generating execution failure reports (1,8):

ERROR Name	Short Mnemonic	Error Code	Notes
EXF_LS_HDL_LCU_OFF	LSHDL_LCU_OFF_2	0x2002	Commanded LCU parameter scan: execution failure due to LCU off

EXF_LS_HDL_ERROR_LS_HP_QUEUE	LSHDL_LSHPQ	0x2006	FIFO Put on LS HP queue failure: no possibility to trnasmit to LS task one of the command of the procedure. The procedure is stopped. Applicable to all scan procedures and to LO tuning.
EXF_LS_HDL_LCU_OFF	LSHDL_LCU_OFF_2	0x2002	Commanded LCU parameter scan: execution failure due to LCU off
EXF_LS_HDL_GEN_REP_FAIL	LSHDL_GEN_REP	0x2005	Generate report error: no final (or step) report can be generated at the end of one of the scan functions.

Refer to Table 29 for the error code of all the other the OBS runtime errors that can be generated during the activity.

7.6.10 Perform activity – Tune functions (FID 9)

7.6.10.1 Tune HRS

The activity is described in AD6, section 3.8.9.1 and is commanded with the TC described in AD3, sect 4.3.8.1.
The TC acceptance report is generated within 500msec from the TC ingestion.

The overall duration of the execution of the procedure onboard is
2.6sec + 1sec = duration of the procedure + initial overhead
even if to download all the science packets other 2 seconds are necessary.
The initial overhead (due to the WBS transfer time) can be eliminated by disabling the spectrometer data acquisition for the periodic HK polling. If the Tune HRS TC aborts a previously running procedure, the initial overhead shall be taken into account even if the periodic HK are not acquiring data from spectrometers..

If a new TC is received before 3.6 (2.6) secs from the reception of the Tune command, the tuning procedure is aborted.
The successful TC execution report is generated after 3.6 (2.6) secs, before the completion of the SD packets download.
Refer to AD4, section 4.9.6.6 for the report structure.

For each one of the two tuning steps, for each one of the polarizations a total of 19 packets are produced:

- 1 tune report
- 1 If Power Hk packet
- 1 start frame packet
- 16 Science Data Packets

The following execution failure detections are implemented, all generating reports (1,8):

ERROR Name	Short Mnemonic	Error Code	Notes
EXF_HS_LIB_ERROR_LS_HP_QUEUE	HSLIB_LSHPQ	0x0905	FIFO Put on LS HP queue failure: no possibility to trnasmit to LS task one of the command of the procedure. The procedure is stopped. Applicable to all tuning procedures and to peakup procedures.
EXF_HS_LIB_HRS_SUB_OFF	HSLIB_HRS_OFF	0x0906	A measurement involving HRS has been requested, but the spectrometers are OFF. No measurement is started.
EXF_HS_LIB_QUEUE	HSLIB_QUEUE	0x0912	FIFO Put on HS HDL Queue failure:no possibility to notify to HS HSLthat one of the tuning procedure is cstarted. The procedure is aborted. Applicable to all tuning procedures.

The following OBS Runtime errors can be detected:

ERROR Name	Short Mnemonic	Error Code	Notes	Severity

ERR_DATA_HDL_ZERO_DIV	ZERO_DIV_0	0x0701	Division by zero	2
ERR_DATA_HDL_FIFOPUT_HK_TM_QUEUE	DATAH_PUT_HKTMQ	0x0702	FIFO put on HK TM queue failed in Data Hdl task. IF Power HK packet lost.	2
ERR_DATA_HDL_FIFOPUT_SD_TM_QUEUE	PUT_SDTMQ	0x0703	FIFO put on SD TM queue failed in Data Hdl task	2
ERR_DATA_HDL_GET_STF_POOL_BLOCK	STF_GETB	0x0705	Get block from Start frames Pool failure (no start frame packet)	2
ERR_DATA_HDL_GET_HK_POOL_BLOCK	HKP_GETB	0x0706	Get block from HK Pool failure (no If power packet produced)	2
ERR_HS_LIB_FIFOPUT_HK_TM_QUEUE	HK_TM_QUEUE_2	0x0901	FIFO put on HK TM queue failed in HS_HDL task. Tune reports lost.	2
ERR_HS_LIB_WRONG_SPECT_AID	WSPECT_AID	0x0918	Inconsistency between current and espected AID spectroscopy	2
ERR_HS_LIB_ERROR_LS_HP_QUEUE	HSLIB_LSHP_Q	0x0927	LS high priority queue handling failure. The tuned parameters have not been commanded.	2
ERR_HS_LIB_GET_HK_POOL_BLOCK	HSLIB_HKPBLK	0x0928	Get block from HK Pool failure. Tuning has been executed but the tune report will not be genefated.	2

7.6.10.2 Tune WBS

The activity is described in AD6, section 3.8.7.9.2 and is commanded with the TC described in AD3, sect 4.3.9.2.
The TC acceptance report is generated within 500msec from the TC ingestion.

The overall duration of the execution of the procedure onboard is
8.7sec + 1sec = duration of the procedure + initial overhead
even if to download all the science packets other 2 seconds are necessary.
The initial overhead (due to the WBS transfer time) can be eliminated either by disabling the spectrometer data acquisition for the periodic HK polling, or by switching Off the periodic HK acquisition.
If the Tune WBS TC aborts a previously running procedure, the initial overhead shall be taken into account even if the periodic HK are not acquiring data from spectrometers..

If a new TC is received before 9.7 (8.7) secs from the reception of the Tune command, the tuning procedure is aborted.
The successful TC execution report is generated after 9.7 (8.7) secs, before the completion of the SD packets download.
Refer to AD4, section 4.9.6.7 for the report structure.

For each one of the three tuning steps, for each one of the polarizations a total of 31 packets are produced:

- 1 tune report
- 1 If Power Hk packet
- 1 start frame packet
- 28 Science Data Packets

The following execution failure detections are implemented, all generating reports (1,8):

ERROR Name	Short Mnemonic	Error Code	Notes
ERR_HS_LIB_WBS_SUB_OFF	HSLIB_WBS_OFF	0x0911	Requested measurement cannot be executed. The spectrometer is off
EXF_HS_LIB_QUEUE	HSLIB_QUEUE	0x0912	FIFO Put on HS HDL Queue failure:no possibility to notify to HS HSLthat one of the tuning procedure is cstarted. The procedure is aborted. Applicable to all tuning procedures.
NOK_HS_LIB_WRONG_INPUT_PARAM	HSLIB_INPUT_PAR	0x0916	Start tuning: wrong target in percent value

The following OBS Runtime errors can be detected:

ERROR Name	Short Mnemonic	Error Code	Notes	Severity

ERR_DATA_HDL_ZERO_DIV	ZERO_DIV_0	0x0701	Division by zero	2
ERR_DATA_HDL_FIFOPUT_HK_TM_QUEUE	DATAH_PUT_HKTMQ	0x0702	FIFO put on HK TM queue failed in Data Hdl task. IF Power HK packet lost.	2
ERR_DATA_HDL_FIFOPUT_SD_TM_QUEUE	PUT_SDTMQ	0x0703	FIFO put on SD TM queue failed in Data Hdl task	2
ERR_DATA_HDL_GET_STF_POOL_BLOCK	STF_GETB	0x0705	Get block from Start frames Pool failure (no start frame packet)	2
ERR_DATA_HDL_GET_HK_POOL_BLOCK	HKP_GETB	0x0706	Get block from HK Pool failure (no If power packet produced)	2
ERR_HS_LIB_FIFOPUT_HK_TM_QUEUE	HK_TM_QUEUE_2	0x0901	FIFO put on HK TM queue failed in HS_HDL task. Tune reports lost.	2
ERR_HS_LIB_ZERO_DIV	ZERO_DIV	0x0904	Division by zero in HS library function	2
ERR_HS_LIB_ERROR_LS_HP_QUEUE	HSLIB_LSHP_Q	0x0927	LS high priority queue handling failure. The tuned parameters have not been commanded.	2
ERR_HS_LIB_GET_HK_POOL_BLOCK	HSLIB_HKPBLK	0x0928	Get block from HK Pool failure. Tuning has been executed but the tune report will not be genefated.	2
ERR_HS_LIB_NUMERICAL_OVFL	HSLIB_NUM_OVF	0x0930	Error in WBS tuning procedure. Correction factor <0.	2
ERR_HS_LIB_ATTEN_ZERO	ATTEN_ZERO	0x0932	Error in WBS tuning procedure. Attenuation factor equal to 0. Correction factor not divided by atten	2
ERR_HS_LIB_DARK_OVFL	DARK_OVFLW	0x0947	Error in WBS tuning procedure. Difference between hot corr and dark = 0. Related to saturation problems.	2

7.6.10.3 Tune Mixer Magnet use HRS

The activity is described in AD6, section 3.8.9.3 and is commanded with the TC described in AD3, sect 4.3.8.3. The TC acceptance report, when requested, is generated within 500msec from the TC ingestion.

The following assumptions are made onboard on the input parameters:

HIF_step_time:	it shall be expressed in units of 100msec	(e.g. if a step time of 0.4sec is needed, HIF_step_time=4)
HIF_Nmagnet:	a maximum of 48 steps is foreseen.	this limit is to maintain the compatibility with the generation of one single tune report.*
HIF_ch1_mx_mg0_C:	no check/assumptions made onboard	
HIF_cv1_mx_mg0_C:	no check/assumptions made onboard	
HIF_mx_mg_step_C:	no check/assumptions made onboard	

*The maximum number of steps presently allowed onboard is 48. This because with 48 iterations we saturate the dimensions of the tune report TM packet: the max number of 16-bit parameters in a mixer magnet tune report, calculated from the max total length of a TM packet, is
498 parameters = 10 parameters*48 iter. + 14 fixed parameters.

The Tune Report is generated as soon as the last requested HRS frame has been acquired (see AD2) and the two tuned currents have been commanded.
Therefore the report will be generated with a time delay with respect to the TC acceptance report greater than

HIF_ NMAGNET *(HIF_STEP_TIME + 132msec) + 42msec + 30msec (+ 1sec).

= Total number of commanded steps* (step time+time necessary to send out all cmd/hk requests + HRS integration time) + HRS transfer time + fixed overhead (+start_meas_overhead).

The start_meas_overhead of 1 sec shall be added when the Hk acquisition is running to take into account the maximum delay in starting the TC execution (the delay is due to the WBS transfer time).

The tuning iterations are executed one after the other, without waiting for the complete transfer of the data of one iteration before starting the following one. After the commanding of the last iteration it is necessary to wait for the last data to be transferred, in order to calculate the Ifpower, to optimize the current and to send the tune report.

The successful TC execution report is generated after this time delay. Refer to AD4, section 4.9.6.8 for the report structure.

If a new TC is received before this time delay, the measurement is aborted.

Refer to Table 29 and Table 30 for the error code of the OBS runtime errors and execution failure reports that can be generated during the activity (sections related to task HK_LIB and HS_HDL).

7.6.10.4 Tune Mixer Magnet use WBS

The activity is described in AD6, section 3.8.7.9.4 and is commanded with the TC described in AD3, sect 4.3.9.4.
The TC acceptance report is generated within 500msec from the TC ingestion.

The Tune Report is generated as soon as the last requested WBS frame has been acquired (see AD2) and the two tuned currents have been commanded.
Therefore the report will be generated with a time delay with respect to the TC acceptance report greater than

HIF_NMAGNET*(HIF_STEP_TIME + 1050msec) + 900msec + 40msec (+1sec)

=Total number of commanded steps* (step time+time necessary to send out all cmd/hk requests + WBS integration time) + WBS transfer time + fixed overhead (+start_meas_overhead).

The start_meas_overhead of 1 sec shall be added when the HK acquisition is running to take into account the maximum delay in starting the TC execution (the delay is due to the WBS transfer time).

In case the TC interrupts a previously running TC, there is an unpredictable initial overhead, which can be at maximum equal to 3 secs.

The successful TC execution report is generated after this time delay. Refer to AD4, section 4.9.6.8 for the report structure.
If a new TC is received before this time delay, the measurement is aborted.

Refer toTable 29 and Table 30 for the error code of the OBS runtime errors and execution failure reports that can be generated during the activity (sections related to task HK_LIB and HS_HDL).

7.6.11 Perform activity – Calibrate functions (FID 10)

7.6.11.1 Calibrate WBS (WBS COMB)

The activity is described in AD6, section 3.8.7.10.2 and is commanded with the TC described in AD3, sect 4.3.10.2.
The TC acceptance report is generated within 500msec from the TC ingestion.

The procedure is implemented in two steps. The first step is the request of a ZERO spectrum (integration time of 1005msec) whose HK are used to know the current values of the attenuators setting. The second step is the measurement of a COMB spectrum with an integration time of (1005msec). The output of each step is a full WBS packetisation per each polarization:
1 IF Power Hk packet
1 Start science Data Frame
28 Science Data Packets.

The maximum overall duration of the execution of the procedure onboard is:
1 sec + 3400msec + 1sec = 5.4 sec
Maximum initial overhead + commanding procedure duration + (WBS transfer time+packetisation time)

If the periodic HK acquisition from spectrometers is disabled the overall duration is of the order of 4.4 sec, even if to download all the science packets other 2 seconds are necessary.
This implies that if a new TC is received within 5.4 (4.4) secs from the reception of the COMB command, the COMB procedure is aborted.

The successful TC execution report is generated within 5.4sec, before the completion of the SD packets download.

The following execution failure detections are implemented, all generating reports (1,8):

ERROR Name	Short Mnemonic	Error Code	Notes
EXF_HS_HDL_ERROR_LS_HP_QUEUE	HSHDL_LSHPQ	0x0809	FIFO Put on LS HP queue failure: no possibility to trnasmit to LS task one of the command of the procedure. The procedure is stopped. Applicable to all tuning procedures.
ERR_HS_LIB_WBS_SUB_OFF	HSLIB_WBS_OFF	0x0911	Requested measurement cannot be executed. The spectrometer is off
EXF_HS_LIB_QUEUE	HSLIB_QUEUE	0x0912	FIFO Put on HS HDL Queue failure:no possibility to notify to HS HSLthat one of the tuning procedure is cstarted. The procedure is aborted. Applicable to all tuning procedures.

Refer to Table 29 for the error code of the OBS runtime errors that can be generated during the activity.

7.6.11.2 Zero Measurement (WBS ZERO)

The activity is described in AD6, section 3.8.7.10.3 and is commanded with the TC described in AD3, sect 4.3.10.3.
The TC acceptance report is generated within 500msec from the TC ingestion.

The procedure is the measurement of a ZERO spectrum with an integration time of 1005 msec. The output is a full WBS packetisation per each polarization:
1 IF Power Hk packet
1 Start science Data Frame
28 Science Data Packets.

The overall duration of the execution of the procedure onboard is:
2.5sec + 1sec = procedure duration + initial overhead
even if to download all the science packets other 2 seconds are necessary.
This implies that if a new TC is received before 3.5 secs from the reception of the WBS ZERO command, it is possible that the WBS ZERO procedure is aborted.
If the procedure starts regularly (no abortion of a previously running procedure onboard) and if the periodic HK acquisition from spectrometers is disabled, the overall duration is 2.5 sec.
The successful TC execution report is generated after the measurement completion (2.5sec < delta time < 3.5 secs), before the completion of the SD packets download.

The following execution failure detections are implemented, all generating reports (1,8):

ERROR Name	Short Mnemonic	Error Code	Notes
EXF_HS_HDL_ERROR_LS_HP_QUEUE	HSHDL_LSHPQ	0x0809	FIFO Put on LS HP queue failure: no possibility to trnasmit to LS task one of the command of the procedure. The procedure is stopped. Applicable to all tuning procedures.
ERR_HS_LIB_WBS_SUB_OFF	HSLIB_WBS_OFF	0x0911	Requested measurement cannot be executed. The spectrometer is off
EXF_HS_LIB_QUEUE	HSLIB_QUEUE	0x0912	FIFO Put on HS HDL Queue failure:no possibility to notify to HS HSLthat one of the tuning procedure is cstarted. The procedure is aborted. Applicable to all tuning procedures.

Refer to Table 29 for the error code of the OBS runtime errors that can be generated during the activity.

7.6.12 Perform activity – Spectroscopy (FID 11)

7.6.12.1 Configure spectroscopy

The Configure spectroscopy TC structure is defined in AD3 sect. 4.3.10.1 and reported below with the default values stored onboard at the OBS initialization:

Table 23 – HIFI_config_spectr TC: packet data field structure

Field	value	Parameter description
Function ID	11	
Activity ID	17	
Spare	0	
HIF_N_WBS_START	1	Number of WBS starts during the whole measurement. If it is 0, no WBS starts are sent out.
HIF_R_HRS	1	Number of HRS-accumulations during one WBS accumulation If it is 0, no HRS starts are sent out.
HIF_N_WBS_INTEGR	1	Number of integrations in WBS-coaddition buffer per packetization.
HIF_N_HRS_INTEGR	1	Number of integrations in HRS-coaddition buffer per packetization<.
HIF_DEL_HRS	10	Delay time before start-HRS command. Time in milliseconds. *
HIF_DEL_WBS	10	Delay time before start-WBS command. Time in milliseconds.*
HIF_T_ACC_WBS	10	Accumulation time of WBS. Time in milliseconds.*
HIF_T_ACC_HRS	10	Accumulation time of HRS. Time in milliseconds.*
HIF_WBSH_OFFSET1	0	Specifies first channel of WBS_H output range 1
HIF_WBSH_WIDTH1	2048	Specifies number of channels in WBS_H output range 1
HIF_WBSH_OFFSET2	2048	Specifies first channel of WBS_H output range 2
HIF_WBSH_WIDTH2	2048	Specifies number of channels in WBS_H output range 2
HIF_WBSH_OFFSET3	4096	Specifies first channel of WBS_H output range 3
HIF_WBSH_WIDTH3	2048	Specifies number of channels in WBS_H output range 3
HIF_WBSH_OFFSET4	6144	Specifies first channel of WBS_H output range 4
HIF_WBSH_WIDTH4	2048	Specifies number of channels in WBS_H output range 4
HIF_WBSV_OFFSET1	0	Specifies first channel of WBS_V output range 1
HIF_WBSV_WIDTH1	2048	Specifies number of channels in WBS_V output range 1
HIF_WBSV_OFFSET2	2048	Specifies first channel of WBS_V output range 2
HIF_WBSV_WIDTH2	2048	Specifies number of channels in WBS_V output range 2
HIF_WBSV_OFFSET3	4096	Specifies first channel of WBS_V output range 3
HIF_WBSV_WIDTH3	2048	Specifies number of channels in WBS_V output range 3
HIF_WBSV_OFFSET4	6144	Specifies first channel of WBS_V output range 4
HIF_WBSV_WIDTH4	2048	Specifies number of channels in WBS_V output range 4
HIF_HRS_RSHIFT	0	Specifies number of bits for HRS data right-shift
HIF_WBS_RSHIFT	0	Specifies number of bits for WBS data right-shift
HIF_HRSH_SEL	0xFF	8 bit word selects banks to be packetized for HRS-H 0xFF = all banks selected. **
HIF_HRSV_SEL	0xFF	8 bit word selects banks to be packetized for HRS-V 0xFF = all banks selected.**
HIF_WBS_packing	16bit	Specifies to telemeter 24 or 16 bits WBS-data 1 – 16 bit data 2 – 24 bit data

*** Note on commanded times.**
Due to the usage of the onboard command interpreter (i.e. VM), the minimum delay between SS commands is 2 ms, so all the time parameters (i.e. integration, delay etc) must be >= 2ms. Due to the lock unlock mechanism of the Low Speed I/F (which is shared bewteen theHk acquisition task and the VM) the next possible delay time is 5 ms, higher delay have no constrain.
The HIFI total power time parameters are checked and corrected in the VM program:
1) T <2ms are changed to 2ms
2) 2ms < T < 5ms are changed to 5ms

Like all the other configure commands (see 7.6.5), the Configure spectroscopy TC will be echoed in a configuration report that will be provided within 500msec from the TC acceptance.

*** Note on HRS range selection.**
The following rule is applied onboard to decode the HRS range selection word:

```
Select = 0x1 (00000001) Block 8 is selected (both subblocks)
Select = 0x2 (00000010) Block 7 is selected (both subblocks)
Select = 0x4 (00000100) Block 6 is selected (both subblocks)
Select = 0x8 (00001000) Block 5 is selected (both subblocks)
Select = 0x10(00010000) Block 4 is selected (both subblocks)
Select = 0x20(00100000) Block 3 is selected (both subblocks)
Select = 0x40(01000000) Block 2 is selected (both subblocks)
Select = 0x80(10000000) Block 1 is selected (both subblocks)
```

In addition to the check on the consistency between the length of the packet and the number of parameters expected, the following checks on the WBS commanded integration time is implemented:

1. HIF_T_ACC_WBS = N*10msec + 5 msec. (where N is an integer number greater than 0).

2. In case of Total Power Spectroscopy :
   HIF_T_ACC_WBS >= (2+ HIF_DEL_HRS + HIF_T_ACC_HRS+10) *(HIF_R_HRS) +2
   Where:
   10 msec = time overhead due to the HRS data transfer commands issuing
   2msec = time overhead due to the HRS reset buffer commands issuing

In case of failure, this extra check will result in an execution failure report (1,8) with the following parameters:

ERROR Name	Short Mnemonic	Error Code	Notes
NOK_CMDEX_ILLEGAL_WBS_ACC_TIME	CMDEX_WBSTIMW	0x0626	Wrong Commanded WBS Time.

In addition to this, all the failure detections implemented for the other configure commands (see 7.6.5) are applicable.

7.6.12.2 Total Power Spectroscopy

The procedure described in AD6 section 3.8.11.3 is performed, to execute a total power spectroscopy measurement, using the measurement configuration parameters commanded with the Configure Spectroscopy Tc and stored onboard.
The TC to start a Total Power Spectroscopy measurement is described in AD3, sect 4.3.11.3.
The TC acceptance report is generated within 500msec from the TC ingestion.

Before starting the procedure, if there is an onboard running measurement, it is aborted and the FIFOs are reset.
The spectrometers commanding necessary to perform the measurement is made by the onboard command interpreter (i.e. VM) triggered by the HW timer interrupt. The flow chart of the VM procedure implemented onboard is reported in A4.2.

During the measurement, depending on the configuration parameters, several HRS and WBS packetisations can be issued.

Num. of WBS packetisations (per polarisation) = HIF_N_WBS_START/ HIF_N_WBS_INTEGR
Num. of HRS packetisations (per polarisation) = (HIF_N_WBS_START* HIF_R_HRS)/ HIF_N_HRS_INTEGR

For each Packetisation the following packets will be provided, see section 7.10.2 for a description of the packets content:

Table 24 Spectrometers packetisations content

HRS	WBS – Packing 16	WBS – Packing 24
1 If Power Hk packet	1 If Power Hk packet	1 If Power Hk packet
1 start frame packet	1 start frame packet	1 start frame packet
16 Science Data Packets (2 packets per selected block)	20 Science Data Packets (5 packets per each full CCD –2048 data)	28 Science Data Packets (7 packets per each full CCD –2048 data)

The first packetisation of WBS will be transmitted after a time approximately equal to:

T wbs >= (HIF_T_ACC_WBS * HIF_N_WBS _INTEGR) + WBS Transfer time

If HIF_N_HRS_INTEGR <= HIF_R_HRS the first packetisation of HRS will be transmitted after a time approximately equal to:

T hrs >= (HIF_T_ACC_HRS * HIF_N_HRS_INTEGR) + HRS Transfer time

If HIF_N_HRS_INTEGR > HIF_R_HRS the first packetisation of HRS will be transmitted after a time greater than:

T hrs > (HIF_N_HRS_INTEGR/HIF_R_HRS) * HIF_T_ACC_WBS + HRS Transfer time

WBS Transfer time= 900msec
HRS Transfer time= 42msec

In case of WBS switched on, the overall duration of the execution of the VM procedure onboard can be calculated according to :

Ttp (msec) = 115 + (HIF_T_ACC_WBS+14+ HIF_DEL_WBS)*(HIF_N_WBS _START)+2

115 msec = measurement initialization time
14 msec = time overhead due to the WBS data transfer commands issuing (12msec wait + 2 cmd trx)
2msec = time necessary to unlock the LS interface.

If only HRS in used, the duration becomes:

Ttp_hrs (msec) = 17 + (2 + HIF_T_ACC_HRS+10+HIF_DEL_HRS)*(HIF_R_HRS)

17 msec = measurement initialization time
10 msec = time overhead due to the HRS data transfer commands issuing
2msec = time overhead due to the HRS reset buffer commands issuing
This implies that if a new TC is received before Ttp from the reception of the Start Total Power Command, the VM procedure will be stopped and the measurement aborted.

The total duration of the measurement is equal to:

Initial overhead + Ttp + WBS transfer time = Ttp + 2 sec

The initial overhead is due to the fact that a science data frame could have been requested by the HK procedure just before the TC is ingested and it takes approx 900 msec to transfer that frame. Therefore the initial overhead is, prudently, specified to be 1 sec. However this overhead can be minimised by disabling the periodic HK acquisition from spectrometers.
.
The final overhead is (essentially) the time it takes to transfer one last science data frame requested by the command. Therefore assuming the worst case transfer time, i.e. the WBS one, it is approximately 900 msec and, for

prudence, it has been set to 1 sec. However if only the HRS is involved in the command this reduces to approximately 50 msec.

If a new TC is received at a time Ttp < t < Ttp+ 2sec (1sec), the measurement is not aborted, because all commanding with VM has been completed, but an Execution failure report is issued, because the last frames of the measurement are lost.

7.6.12.3 Slow/Chop Spectroscopy

The procedure described in AD6 section 3.8.11.4 is performed, to execute a slow chop spectroscopy measurement, using the measurement configuration parameters commanded with the Configure Spectroscopy TC and stored onboard.

The TC to start a Slow Chop Spectroscopy measurement is described in AD3, sect 4.3.11.5. and contains two extra parameters (the chopper positions for the modulation) necessary to implement the measurement.
The TC acceptance report is generated within 500msec from the TC ingestion.

The spectrometers commanding necessary to perform the measurement is made by the onboard command interpreter (i.e. VM) triggered by the HW timer interrupt. The flow chart of the VM procedure implemented onboard is reported in A4.2.

See section 7.6.12.2 for the number of packets/packetisations provided and for the calculation of the time expected for the transmission of the first WBS and HRS packetisations.

In case of WBS switched on, the overall duration of the execution of the procedure onboard can be calculated according to:

Tsc (msec) = 115 + (HIF_T_ACC_WBS+14+ HIF_DEL_WBS+2)* HIF_N_WBS_START

115 msec = measurement initialization time
14 msec = time overhead due to the WBS data transfer commands issuing (12msec wait + 2 cmd trx)
2msec = time necessary to command the new chopper position.

If only HRS in used, the duration is the same as in total power spectroscopy.

If a new TC is received before Tsc from the reception of the Start Slow Chop Command, the VM procedure will be stopped and the measurement aborted.
The total duration of the measurement is equal to:

Initial overhead + Tsc + WBS transfer time = Tsc + 2 sec

The initial overhead is due to the fact that a science data frame could have been requested by the HK procedure just before the TC is ingested and it takes approx 900 msec to transfer that frame. Therefore the initial overhead is, prudently, specified to be 1 sec. However this overhead can be minimised by disabling the periodic HK acquisition from spectrometers.
.
The final overhead is (essentially) the time it takes to transfer one last science data frame requested by the command. Therefore assuming the worst case transfer time, i.e. the WBS one, it is approximately 900 msec and, for prudence, it has been set to 1 sec. However if only the HRS is involved in the command this reduces to approximately 50 msec.

If a new TC is received at a time Tsc < t < Tsc+ 2sec (1sec), the measurement is NOT aborted, because all commanding with VM has been completed, but an Execution failure report is issued, because the last frames of the measurement are lost.

7.6.12.4 Fast Chop Spectroscopy

The procedure described in AD6 section 3.8.11.5 is performed, to execute a fast chop spectroscopy measurement, using the measurement configuration parameters commanded with the Configure Spectroscopy TC and stored onboard.

The TC to start a Fast Chop Spectroscopy measurement is described in AD3, sect 4.3.11.4. and contains 4 parameters (the 2 chopper positions for the modulation, the number HIF_N_HRS_TRANS of HRS integrations to be done during WBS transfer and the number HIF_N_WBS_1 of WBS modulations/integrations) necessary to implement the measurement.
The TC acceptance report is generated within 500msec from the TC ingestion.

The spectrometers commanding necessary to perform the measurement is made by the onboard command interpreter (i.e. VM) triggered by the HW timer interrupt. The flow chart of the VM procedure implemented onboard is reported in A4.2.

During the measurement, depending on the configuration parameters, several HRS and WBS packetisations can be issued.

Num. of WBS packetisations (per polarisation) =
(HIF_N_WBS_START*2) / HIF_N_WBS_INTEGR

Num. of HRS packetisations (per polarisation) =
(HIF_N_WBS_START*(HIF_N_WBS1 + 2 * HIF_N_HRS_TRANS))/ HIF_N_HRS_INTEGR

IT IS ASSUMED ONBOARD THAT HIF_R_HRS=1.

The overall duration of the execution of the VM procedure onboard can be calculated according to :

Tfc = 115 + ((HIF_T_ACC_WBS + HIF_DEL_WBS)*HIF_N_WBS1+
2*(HIF_N_WBS1-1)+12+(900+2)*2))*HIF_N_WBS_START

115msec = measurement initialization time
(900+2)msec = wbs transfer time + unlock I/F= duration of the transfer loops
12msec = delay between the end of t_acc_wbs and the start of the readout
2msec = time to command new chopper position. It is multiplied by
(N_WBS1-1) because the first time this setting is included in the initialization time.

If a new TC is received before Tfc from the reception of the Start Fast Chop Command, the VM procedure will be stopped and the measurement aborted.

The total duration of the measurement is equal to:

Initial overhead + Tfc = Tfc + 1 sec
The initial overhead can be minimised by disabling the periodic HK acquisition from spectrometers.

WARNINGS:
1) During the procedure, the data are acquired onboard onto two different integration buffers, swapping between the two buffers at each new frame received.
The Commanded coadditions (HIF_N_HRS_INTEGR and HIF_H_WBS_INTEGR) are applied to each buffer separately. This implies that if a coaddition of 5 is requested, at least 10 starts shall be commanded.

2) The above assumption implies that if the total number of commanded starts per each spectrometer is not an even number and if it is not an even multiple of the requested coaddition, the fast chop command is aborted.

2) In fast chop coadditions are particularly necessary to keep the rate of produced packets compatible with the downlink datarate. E.g. Use the HRS coaddition to keep the total number of HRS packetisation coming out from the

HIF_N_WBS1 loop compatible with the number of packets that can be downloaded during the loop duration and to avoid saturating the SD Pool before the transfer loop is started.

7.6.12.5 Spectroscopy measurement error conditions

The following execution failure detections are implemented, all generating reports (1,8):

ERROR Name	Short Mnemonic	Error Code	Notes
NOK_HS_LIB_WRONG_INPUT_PARAM	HSLIB_INPUT_PAR	0x0916	inconsistency between measurement parameters 6. - coaddition and number of integrations - total number of WBS starts = 0
EXF_HS_LIB_NO_INSTRU_INVOLVED	HSLIB_NO_INSTR	0x0917	inconsistency between measurement parameters (spectrometers involved and spectrometers switched on)

The following error Conditions are foreseen during execution:

ERROR Name	Short Mnemonic	Error Code	Notes	Severity
ERR_HS1_FIFOPUT_SD_PKT	PUT_SD_PKT	0x0302	Error in FIFO PUT. SD packet lost	2
ERR_HS1_FRAME_COMPLETE_FAIL	FPUT_FC	0x0303	Error in FIFO PUT. Completed Frames counter wrong.	2
ERR_HS1_SCAN_COUNT	SC_ZERO	0x0304	Scan Count equal to zero detected. SD packet lost.	2
ERR_HS1_SD_GET_BLOCK	SD_GET_FAIL	0x0305	Failure in getting a SD packet block from pool.	4
ERR_DATA_HDL_ZERO_DIV	ZERO_DIV_0	0x0701	Division by zero in ifpower calculation	2
ERR_DATA_HDL_FIFOPUT_HK_TM_QUEUE	DATAH_PUT_HKTMQ	0x0702	FIFO put on HK TM queue failed in Data Hdl task. IF Power HK packet lost.	2
ERR_DATA_HDL_FIFOPUT_SD_TM_QUEUE	PUT_SDTMQ	0x0703	FIFO put on SD TM queue failed in Data Hdl task	2
ERR_DATA_HDL_GET_STF_POOL_BLOCK	STF_GETB	0x0705	Get block from Start frames Pool failure (no start frame packet)	2
ERR_DATA_HDL_GET_HK_POOL_BLOCK	HKP_GETB	0x0706	Get block from HK Pool failure (no If power packet produced)	2
ERR_HS_LIB_QUEUE_FAIL	HSLIB_Q	0x0919	Failure in procedure to execute abort.	2
ERR_HS_LIB_BRING_SAFE_TIMEOUT	TIMEOUT	0x0902	Failure in procedure to execute abort.	2
ERR_RESCK_SD_POOL_OVERFLOW_WARN	SDP_OVF	0x1103	WARNING. SD Pool Overflow	4

Refer to Table 29 and Table 30 for the error codes of all OBS runtime errors related to task HS0 and TMTC obtained during the measurement.

7.6.12.6 Spectroscopy Measurements Activity AIDs

The status of the OBS during a measurement can be monitored by looking at the ICU HK.

For all the period in which the variable AID_Spectroscopy is different from zero, the measurement commanding is going on and no Data Transfer requests are issued to the spectroemters by the hk_ask task. In these case the spectrometers data are the ones of the last science data frame acquired.
Therefore, when AID Spectroscopy is not 0, the spectrometers data in the nominal HK packet can be missing, because the science data frame acquisition and the Nominal HK packet preparation are not synchrounous processes.

If the running measurement is one of the spectroscopy measurement, the VM Running variable shall be equal to 1.

Table 25 AID_spectroscopy codes

Mnemonic	Code	Meaning

REQUIRING_HK	0	No operation, only HK
STOPPING_HK	1000	No operaton – wait state
STOPPING_VM	2000	No operaton – wait state
STOPPING_OBS	3000	No operaton – wait state
TC_FUNMAN_SPCTRSM_AID	127	Simulated science
HRS_ATTSETT_ID	18	First step of hrs tune
HRS_ATTSETT2_ID	33	Second step of hrs tune
TUNE_HMIXMAG_ID	21	Mixer magnet hrs tuning
DIP_SCAN_IF_ID	23	Diplexer scan
WBS_ATTSETT_ID	19	First step of wbs tune
WBS_ATTSETT2_ID	34	Second step of wbs tune
WBS_ATTSETT2_ID	34	Third step of wbs tune
WBS_ATTSETT3_ID	35	Fourth step of wbs tune
TUNE_WMIXMAG_ID	22	Mixer magnet wbs tuning
ZSCAN_ID	20	Zero scan
TC_FUNMAN_SPCTRTP_AID	1	Total power
TC_FUNMAN_SPCTRSC_AID	3	Slow chop
TC_FUNMAN_SPCTRFC_AID	2	Freq. Switch
TC_FUNMAN_SPCTRFS_AID	4	Fast chop
WBS_COMB_STEP1_ID	38	Wbs comb step 1
WBS_COMB_STEP2_ID	39	Wbs comb step 2
SPECTROSCOPY_WBS_ZERO_ID	40	Wbs zero
PEAKUP_ACQUIRE_HRS	43	Peakup hrs
PEAKUP_ACQUIRE_WBS	44	Peakup wbs

7.6.13 Perform activity – Peakup (FID 13)

Peakup Overview

In the peakup procedure the OBS acquires 9 total power measurements (wbs or hrs, H or V) at appropriate telescope pointings. The telescope pointings are expressed in a coordinate system with two axis, one labeled **Y** axis and the other labeled **Z** axis. The 9 positions are organised into a normalised square grid defined in the telescope coordinate system. The normalised grid points have the following (Z,Y) coordinates

	-1	Telescope Z axis	+1
+1	-1,+1	0,+1	+1,+1
Telescope Y axis	-1,0	0, 0	+1,0
-1	-1, -1	0,-1	+1,-1

and are numbered from 1 to 9 according to the following rules:
- Cell rows [1-2-3], [4-5-6] and [7-8-9] are along the Z axis. Moreover, cells 1,4,7 have negative Z, cells 2,5,8 have Z=0 and cells 3,6,9 have positive Z.
- Cell rows [1-4-7], [2-5-8] and [3-6-9] are along the Y axis. Moreover, cells 1,2,3 have negative Y, cells 4,5,6 have Y=0 and cells 7,8,9 have positive Y.

Resulting in the following table

	-1	Telescope Z axis	+1
+1	7	8	9
Telescope Y axis	4	5	6
-1	1	2	3

After initialising the peakup procedure the CDMS points the telescope into the 9 positions and in each position instructs the OBS to take a measurement. When 9 measurements have been acquired the CDMS ask the OBS to compute the best pointing according to the measurements just taken. The OBS computes the best pointing using an algorithm developed by SRON (author: Do Kester) and returns to the CDMS the found best pointing. While the whole processing is done assuming the normalised grid, the OBS receives from the CDMS two scaling factors (HIFI_peakup_scl_Y and HIFI_peakup_scl_Z) which are used to scale the normalised best pointing coordinates. It also receives two offsets (HIFI_peakup _offset_Y, HIFI_peakup_offset_Z) that are added to the scaled best pointing coordinates to produce the data actually returned to the CDMS.

Peakup Steps

Configuration: The peakup procedure is started by the CDMS by sending to the OBS the **HIFI_Configure_Peakup** TC which carries the peakup scaling factors and the offsets. Upon reception of this TC the OBS resets the peakup static data and stores the peakup parameters. There is no TM generated by the OBS in response to this TC. The command is istantaneous and can be executed while other commands involving the spectrometers are running.

Acquisition: The peakup procedure continues with the CDMS sending to the OBS 9 **HIFI_Acquire_Peakup_using_HRS** TCs or **HIFI_Acquire_Peakup_using_WBS** TCs. With this TC the CDMS informs the OBS that the telescope is pointed in the right direction and asks the OBS to perform an instument read out. Upon reception of one of these TC the OBS executes a total power read out and stores the result for later processing. The OBS generates a **Peakup_step_report** TM packet in response of each of the 9 TCs. The intergration time used by

the OBS is 1 sec and therefore this is the approximate execution time of this TC. This TC is not compatible with other TCs involving the spectrometeres: if issued while another is running the running command would be aborted. If another TC involving the spectrometers is issued while this TC is running the TCV is aborted.

Interpolation: The peakup procedure is closed when the CDMS issues a **HIFI_ correction_AOCS** TC. With this TC the CDMS requires to the OBS to find the best telescope pointing. Upon reception of this TC the OBS computes the best pointing by interpolating the previously stored total power measurements. In response to this TC a **AOCS_correction_report** and a **HIFI_peakup_request_Event** TM packets are produced. The first summarises the data used in the interpolation while the second reports the best pointing found. This command is istantaneous and is compatible with other Tcs using the spectrometers.

Peakup specific errors:

ERROR Name	Short Mnemonic	Error Code	Notes	Severity
EXF_HS_LIB_HRS_SUB_OFF		0x0906	The HRS is OFF	
EXF_HS_LIB_PEAKUP_POLAR_TYPE_PAR		0x0922	A wrong polarization (H or V) parameter has been used	
EXF_HS_LIB_PEAKUP_SEQ_NUMBER_PAR		0x0923	The sequence number is wrong	
EXF_HS_LIB_ERROR_LS_HP_QUEUE		0x0927	The LS HP queue is full	
EXF_HS_LIB_WBS_SUB_OFF		0x0911	The wbs is OFF	
ERR_HS_LIB_HRS_DURATION_NULL		0x0941	The head_chan of HRS was zero	
ERR_HS_LIB_ZERO_DIV		0x0904	Divide by zero in wbs IF power computation	
EXF_HS_LIB_PEAKUP_INCOMPLETE		0x0925	Not all the 9 grid points were acquired from the same instrument	
EXF_HS_LIB_PEAKUP_NEGATIVE_POWER		0x0949	One of the IF power is negative	
EXF_HS_LIB_PEAKUP_FOUND_MINIMUM		0x0950	The interpolation resulted in a minimum and not a maximum	
EXF_HS_LIB_PEAKUP_OUT_OF_GRID		0x0951	The interpolated peak is out of the grid	
EXF_HS_LIB_PEAKUP_FOUND_NEGATIVE_PEAK		0x0952	The peak is negative	
EXF_HS_LIB_PEAKUP_Z_OVERFLOW		0x0953	The Z position exceeds 16 bits	
EXF_HS_LIB_PEAKUP_Y_OVERFLOW		0x0954	The Y position exceeds 16 bits	

Peakup specific TC packets

HIFI_Configure Peakup (type 8, subtype 4)

Position	Length	Field	value
10	8	Function ID	13
11	8	Activity ID	1
12	16	Spare	0
14	32	Building block ID	
18	16	HIFI_peakup_scl_Y	
20	16	HIFI_peakup_scl_Z	
22	16	HIFI_peakup_offset_Y	
24	16	HIFI_peakup_offset_Z	

HIFI_Acquire_Peakup_using_HRS (type 8, subtype 4)

Position	Length	Field	value
10	8	Function ID	13
11	8	Activity ID	2
12	16	Spare	0
14	32	Building block ID	
18	16	HRS polarization	0 = H 1 = V
20	16	HI_peakup_sqnr	1 - 9

HIFI_Acquire_Peakup_using_WBS (type 8, subtype 4)

Position	Length	Field	Value
10	8	Function ID	13
11	8	Activity ID	3
12	16	Spare	0
14	32	Building block ID	
18	16	WBS polarization	0 = H 1 = V
20	16	HI_peakup_sqnr	1 – 9

HIFI_ correction_AOCS (type 8, subtype 4)

Position	Length	Field	Value
10	8	Function ID	13
11	8	Activity ID	5
12	16	Spare	0
14	32	Building block ID	

Peakup specific TM packets

Peakup_step_report (type 3, subtype 25)

Position	Length	Field	value
16	16	Structure ID	272
18	32	Observation ID	
22	32	Building block ID	
26	16	Spectrometer ID	HRS_V_ID 3 HRS_H_ID 2 WBS_V_ID 1 WBS_H_ID 0
28	16	measurement seqn.	
30	16	calculated average Ifpower	

AOCS_correction_report (type 3, subtype 25)

Position	Length	Field	value
16	16	Structure ID	274
18	32	Observation ID	
22	32	Building block ID	
38	16	IF power position 1 (Y,Z=-1-1)	Y, Z
40	16	IF power position 2 (-1,0)	
42	16	IF power position 3 (-1,1)	
44	16	IF power position 4 (0,-1)	
46	16	IF power position 5 (0,0)	
48	16	IF power position 6 (0,1)	
50	16	IF power position 7 (1,-1)	
52	16	IF power position 8 (1,0)	
54	16	IF power position 9 (1,1)	
56	16	microrotation y	Scaled by the scaling factor and added with the offset
58	16	microrotation z	idem
60	16	HIFI_peakup_scl_Y	
62	16	HIFI_peakup_scl_Z	
64	16	HIFI_peakup_offset_Y	
66	16	HIFI_peakup_offset_Z	

HIFI_peakup_request_Event (type 5, subtype 1)

Position	Length	Field	value
16	16	Event ID	0xC000
18	16	Structure ID	0xC000
20	32	Observation ID	
24	32	Building block ID	
28	16	Event Counter	
30	16	Instrument ID	1 (HIFI)
32	16	microrotation y	Scaled by the scaling factor and added with the offset
34	16	microrotation z	idem

7.6.14 Perform activity – ICU Internal Activity (FID 16)

The following activities are implemented to execute ICU memory handling procedures:

7.6.14.1 EEPROM Write

Overview. The ICU is equipped with an EEPROM that can store two copies of the OBS. Either of these copies can be loaded into the Program Memory (PM) and started by the Boot Software (BSW). The contents of the EEPROM are managed by the OBS. Specifically a running version of the OBS can copy itself in the EEPROM by means of the EEPROM_write TC, described in the following.

EEPROM organisation. The EEPROM is constituted by 256 pages of 1024 words of 32 bits. This memory is logically divided into two partitions: partition 1 starts at page #1 and ends at page #128 while partition 2 starts at page #256 and ends at page #129. The two partitions are used to store two copies of the OBS.

The EEPROM memory can be adressed on the Data Bus: it starts at address 0x8000 0000. Each page has a length of 1024 = 0x400 words so that 0x8000 0000 is the physical address of the first page, 0x8000 0400 of the second page, 0x8000 0800 the third page and so on. Therefore page N (where N = 1..256) has a starting addres that can be computed as follows

start_address = 0x8000 0000 + (N-1) * 0x400.

Each EEPROM page has a 7 words header reported below

<table>
<tr><th>Byte #4</th><th>Byte #3</th><th>Byte #2</th><th>Byte #1</th></tr>
<tr><td colspan="2">PM Data start address pointer (20bits)</td><td colspan="2">PM Data length (12 bits)</td></tr>
<tr><td colspan="2">Index of Current Page (16bits)</td><td colspan="2">Total Number of Pages (16bits)</td></tr>
<tr><td colspan="4">next EEPROM/DM page pointer (32bits)</td></tr>
<tr><td colspan="2">Reserved</td><td>Boot options Byte (8bits)</td><td>Reserved</td></tr>
<tr><td colspan="4">ASW start address (32bits)</td></tr>
<tr><td colspan="2">Frame Check sequence of the PM Data (16 bits)</td><td colspan="2">Frame Check sequence of EEPROM/DM Page (16 bits)</td></tr>
<tr><td colspan="2">0x0000</td><td colspan="2">Frame Check Sequence Overall Program (CRC16)</td></tr>
</table>

Bad pages. When the ICU is switched on the BSW performs some tests to check the integrity of the PM, the DM and the EEPROM. The EEPROM test is accomplished by computing the *Frame Check sequence* (FCS) of that page and comparing it to the FCS stored in the page header. If the two do not match the loading process is interrupted and an error message TM(5,4) is issued. In the error packet (see [AD11] for its detailed structure) the number of bad pages and the pagenumbers are reported. The pagenumbers are in a range [0-255]

Each time the ICU is switched on all EEPROM pages are tested. An image can not be loaded from EEPROM to PM when it contains a corrupted page. At attempt to do so will fail as every page will tested again when it is loaded to the PM.

If a list of EEPROM bad pages is mantained, it is possible to avoid using these pages to store the OBS, as we see later. However note that, given the BSW functioning, a bad page can be detected only if it is used at least once to store the OBS.

The partition is organized as a linked list of memory-pages, the header of each page contains a pointer to the next page. This implies that the first page of the partitions is fixed and therefore it is not possible to avoid the first page. Losing page #1 corrupts partition #1, losing page #256 corrupts partition #2.

OBS organisation. The OBS code is divided into three segments: RTH, INIT and PMCO. The first segment hosts the interrupt vector and other Operating System information. The second segment hosts the initial values of the static data of the OBS. The third segment hosts the actual OBS code. The RTH segment starts at address 0x0. The INIT segment starts in 0x4000 in all the OBS versions delivered. The PMCO segment has a varying starting address, computed in order to minimise the wasted memory, close to the end of the INIT segment.

Functioning of the EEPROM_write TC. The EEPROM_write TC format is reported in [AD3]. Its main parameters will be discussed here and are the following:

- Start_address:	PM address of the starting word of the INIT segment - fixed to 0x4000
- OBS_mem_end	PM address of the last word of the PMCO segment
- HIF_partition_id	partition to use: can be 1 or 2
- HIF_Npages_to_skip	Number of bad pages
- HIF_page_nx	List of page-numbers to skip (omitted when HIF_Npages_to_skip = 0)

When the OBS receives this command it does the following. Firstly, it decides which partition to use based on the parameter *HIF_partition_id* (which has to be 1 or 2 according to the desired partition). Next, the RTH segment is copied into the first partition page since this segment always fits into one page. Finally, the PM memory comprised between the addresses *Start_address* and *OBS_mem_end* is copied in the following partition pages. Therefore, in order to copy the whole OBS (i.e. INIT + PMCO) into the EEPROM, these two parameters shall be set, respectively, equal to the PM address of the starting word of the INIT segment (which is fixed to 0x4000) and to the PM address of the last word of the PMCO segment. This address is obtained inspecting the compiler log files and is reported by IFSI at every OBS delivery.

Note that when the writing is done into partition 1, it follows the page ordering. That is, the first page to be written (RTH) is page 1. Then the writing proceeds to pages 2, 3, ... On the contrary when the writing is done in the second partition, it proceeds in revers ordering. That is the first page to be written (RTH) is page 256. Then the writing proceeds to pages 255, 254, ...

The TC can carry a list of bad EEPROM pages to skip in the writing process. The number of bad pages to skip is reported in the parameter *HIF_Npages_to_skip*. If this parameter is zero there are no (known) bad pages. If it is not zero the TC must be continued with a list of the page indexes to skip in the writing process. These indexes must lie in the range 1-256 and refer to the page ordering during the writing process. In other words, if we are writing into partition 1, page is is really page 1 but if the writing is into partition 2 then page 1 is actually the 256-th EEPROM pages.

Maximum size of the OBS and partition overflow. There is a limit to the size of the OBS that can fit into one partition. This limit is computed as follows, when we divide the EEPROM into two partitions of 128 pages and when we assume no bad pages. The first partition page is reserved for the RTH segment which leaves 127 pages free for the INIT and PMCO segments. Since each page has 1024 words of 32 bits but the first 7 words are reserved for the header, each page can store 1017 words of 32 bits and one partition can host

partition_capacity = 127 * 1017 (words of 32 bits)

Since the PM is organised into words of 48 bits, two PM words occupy 3 EEPROM words. Therefore the partition capacity can conveniently be expressed in terms of PM words as

partition_capacity = 127 * 1017 * 2 / 3 = 86106 = 0x1505A (words of 48 bits)

Pages that are not used by the image in one partition may be used by the image in the other partition.

In order to avoid partition overflow the PM range copied (given by *OBS_mem_end* - *Start_address* +1) should not exceed the capacity. That is the following shall hold

OBS_mem_end - *Start_address* +1 <= 0x1505A.

Therefore, if *Start_address*=0x4000 we get that the maximum value for *OBS_mem_end* is 0x19059. These values needs to be recalculated in the presence of bad pages.

Note that the OBS performs no check that the capacity limits are met. Therefore if the limits are not met, the writing takes place but a partition overflow occurs. This means that the OBS copied into one partiton will spill into the other partition, possibly corrupting the second OBS copy. However in this way we can renounce to the second copy and exploit the whole EEPROM for a single, longer OBS copy.

The EEPROM_write procedure extends the PM-memory area [Start_address, OBS_mem_end] to an integer number of pages. If OBS_mem_end is larger than the PM address of the last word of the PMCO undefined data (i.e. data that happened to be present in the PM) will be written into the EEPROM. If the difference between OBS_mem_end and the actual end of PMCO is large enough, full pages with undefined data may occur.
There are several reasons to select OBS_mem_end such that full pages of undefined data are prevented.

- The larger the size of an image, the larger to probability of an corrupted page
- A page that contains undefined data is not available for an image in the other partition.

Final notes. The TC acceptance report is generated within 500msec from the TC ingestion. The overall EEPROM copy procedure can last up to 20 seconds. During the procedure, the nominal HK acquisition rate can be altered, due to the execution of a higher priority task. For the error conditions and messages refer to section **7.6.14.7**.

7.6.14.2 ICU RESET

The TC to reset the OBS is described in AD3, sect 4.3.17.4.
The TC acceptance report is generated within 500msec from the TC ingestion.

Upon reception of this TC, the OBS disables all interrupts, executes a DSP reset and jumps to the initial address of the OBS. Everithing is reinitialised and the OBS restarts providing nominal HK at the default rate of 1 pkt/3sec.

7.6.14.3 Jump to Boot

The TC to reset the OBS is described in AD3, sect 4.3.17.5.
No TC acceptance report is generated by the OBS. The successful ingestion of the Jump to Boot TC is traced by the generation of the first (5,1) event provided by the Boot Software, within 10sec from the TC ingestion.

This function implements the warm reset of the ICU. Upon reception of this TC, the OBS disables all interrupts, restores the Interrupt table of the Boot Software, executes a DSP reset and jumps to the start address of the Boot Software. As soon as the initial (5,1) event is acquired the operations can be restarted with a Force Boot command.

7.6.14.4 Check PM

This command executes the check of the PM memory onboard. The TC to reset the OBS is described in AD3, sect 4.3.17.6. It contains the start address and the end address of the PM memory area to be checked and the value of the expected CRC. In case the CRC calculated onboard by the OBS is different from the one in the TC, an execuiton failure is issued.
For the error conditions and messages refer to section 7.6.14.7.

7.6.14.5 Copy OBS

The TC to reset the OBS is described in AD3, sect 4.3.17.7. Its structure is reported below:
Copy OBS :
Type 8, Subtype 4, FID 16, AID 6
Parameters:

16 bit "Direction"	1 copy from Low Memory to High memory 2 copy from High memory to Low memory and reset
32bit "Destination Address"	start Address in high memory of the copy area (Low memory always starts at 0)
32bit "Number of PM words to copy"	

The procedure to implement OBS patching onboard is based on the TC copy_OBS Command. It has three parameters: direction (16 bits), offset (32 bits) and num_of_words (32 bits) and it has two different behaviours based on direction.

- if direction is 1, it copies from low memory to high memory. Specifically it copies num_of_words words of the PM (starting from location 0) into the PM area starting at offset.

- if direction is 2, it copies from high memory to low memory and issues a hardware reset. Specifically it copies num_of_words words of the PM starting from location offset into the PM area starting at address zero.

The procedure therefore is the following:
1 - issue a Copy OBS with direction 1 to copy the whole OBS in high memory
2 - Patch the copy of OBS in High memory using Memory Load Service
3 - issue a Copy OBS with direction 2 to copy back the new OBS and restart the OBS.
See section 4.3.2 for a detailed description of the OBS patching procedure.

7.6.14.6 Simulated Spectroscopy

The TC to command this function is described in AD3, sect 4.3.15.
Upon reception of this TC the OBS starts generating Dummy science data for all the 4 spectrometers.
The dummy science packets are generated at the maximum data rate compatible with a CDMS buslist including 27 slots for TM packets download. A rate of 2 HK packets (1 nominal+1 essential) per second has been considered.
The simulate spectroscopy procedure executes a continuous series of simulated measurements. Each measurement is based on the configure spectroscopy default parameters stored onboard (see 7.6.12.1). Therefore a continuous series of HRS and WBS packetisations shall be expected, with WBS 16 bit packetisation format (see Table 27). No IF power HK packets are provided in this case.
The simulated data are increasing numbers ranging from 0 up to 258; this interval is repeated the number of times necessary to fill up a complete frame (4128 data for HRS and 8192 data for WBS).

The generation of simulated data can be stopped only by sending an HIFI_abort_spectroscopy TC.

7.6.14.7 ICU functions error conditions.

The following execution failure conditions are foreseen:

Err. Num	Error ID	Onboard mnemonic	Short mnemonic	Detected runtime error
0x100a	EXF_MEM_EEPROM_WRITE_OVER FLOW	EEWRITE_OVF	Error during the EEPROM write procedure (to copy in EEPROM the full OBS image).	
0x100b	EXF_MEM_EEPROM_COPY_FAILED	EECOPY_FAIL	Error during the EEPROM write procedure (to copy in EEPROM the full OBS image).	
0x100c	EXF_MEM_EEPROM_WRONG_PAR TITION	EE_PARTIT	Error during the EEPROM write procedure: wrong commanded partition	1: requested partition

			ID	
0x100d	EXF_MEM_EEPROM_WRONG_NUM_BAD_PAGES	EE_NUMPAGES	Error during the EEPROM write procedure: wrong commanded number of pages to avoid to copy on.	1: number of pages to avoid
0x100e	EXF_MEM_TUNING_QUEUE	TQUEUE	FIFO put on hs_hdl queue failure during memory dump or memory check.	
0x100f	EXF_MEM_COPY_OBS	COPY_OBS	Failure in the execution of the Copy OBS command.	1: return parameter of the procedure. Possible values: 1= wrong number of words 3= wrong requested direction
0x1010	EXF_MEM_CHECK_PM_CRC	CK_PM_CRC	The check PM command resulted in a PM CRC different from expected, indicating a possible damage in the PM memory.	1: obtained CRC.
0x1011	EXF_MEM_EEPROM_WRONG_BAD_PAGE_ID	EE_BADPAGE_ID	Error during the EEPROM write procedure: one of the IDs of pages to avoid to copy on is wrong	1: wrong page ID.

Refer to Table 29 and Table 30 for the error codes of all OBS runtime errors obtained during the ICU internal activities.

7.6.15 Stop activity

Description	Type	Subtype	Funct ID	Activ ID	AD3 section	Action executed onboard
Housekeeping_off	8	2	3		4.2.1	Stops periodic HK acquisition; see sect. 7.3.1.2
Switch off autonomy	8	2	11		4.2.3	Switch off limit checking activities; see sect 7.6.4
HIFI_abort spectroscopy	8	2	11		4.2.3	Aborts the presently running measurement

Upon reception of the HIFI_abort_spectroscopy TC, the presently running spectroscopy measurement is stopped and the acquired data buffers are reset. The overall procedure can take up to 1 sec starting from the TC acceptance. The reception of a TC execution failure report if foreseen within 1 sec, referred to the measurement that has been aborted.

7.7 Service Type 9: Time Management

The OBS receives the time from the bus controller which distributes the sync signal every second. This (absolute) time is stored in an internal variable. When the time information is necessary, the OBS computes the fraction of seconds elapsed from the last sync signal and adds this quantity to the absolute time. The expected accuracy is of the order of 30 µsec. The internal time is driven by the 20MHz clock and stored in a 32 bit register which then wraps around every about 214 seconds.

7.7.1 Time verification

The only command accepted for this service is the HIFI_Enable_Time_Verify, Type 9, subtype 7 (with no parameters), described in AD3, section 4.4.3.

When this command is received, the OBS reports its time in a HIFI_Time_verification_report (type 9, subtype 9), described in AD4 sect. 4.6.2, where the time is written in 48 bits: the first 32 bits are the seconds, while the LSB contains the fraction of seconds in 1/65536 seconds. Since this time refers to the next sync signal, what the OBS reports is just the absolute time at the last sync plus 1 second.

The following Execution Failures are foreseen:

ERROR Name	Short Mnemonic	Error Code	Notes
EXF_CMDEX_GET_BLOCK_HK_POOL	CMDEX_GET_HKP	0x0630	Get block from HK Pool failure: no report generation is possible. (applicable to time verification). No command activity is performed onboard.
NOK_CMDEX_PUT_HK_TM_QUEUE	CMDEX_PUT_HKTMQ	0x062b	FIFO Put on HK TM Queue failure:no possibility to notify to TMTC that there is a report ready to be downloaded. (applicable to enable time verification). The other activities of the command have been already executed.

7.8 Service Type 14: Packet Transmission Control

The service is used to enable/disable the transmission of TM packets, either specifying their type/subtype, or including in the specification also the SID (limiting the transmission control to only one kind of packet belonging to the same "cathegory", e.g., type 3, subtype 25, SID 271= only the engineering scan reports will be addressed).
The commands accepted for this service are:

1. HIFI_enable_TM: Type 14, subtype 1, see section AD3, section 4.5.1.
2. HIFI_disable_TM: Type 14, subtype2, see section AD3, section 4.5.2.
3. HIFI_report_enabled: Type 14, subtype 3, see section AD3, section 4.5.3.

When this command is received, the OBS disables the transmission of the requested Tm packets, but not their generation onboard.
In case of the TC 3 in the list above, a variable length HIFI_TM_generation_status_report is issued within 500 msec from the TC ingestion. The report is described in AD4, section 4.7.1.

The following TM packets can presently be controlled.

TYPE	SUBTYPE	SID
TC_VERIFICATION	TC_ACCREP_SUCCESS	0
TC_VERIFICATION	TC_ACCREP_FAILURE	0
TC_VERIFICATION	TC_EXEC_COMPLETED	0
TC_VERIFICATION	TC_EXEC_FAILURE	0
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	ESSENTIAL_HK_SID
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	PERIODIC_HK_SID
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	NON_PER_LCU_HK_SID
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	NON_PER_LCU_MEM_SID1
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	NON_PER_LCU_MEM_SID2
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	NON_PER_FCU_HK_SID
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	HRSH_IF1_SID
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	HRSH_IF2_SID
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	HRSV_IF1_SID
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	HRSV_IF2_SID
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	WBSH_IF1_SID
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	WBSH_IF2_SID
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	WBSV_IF1_SID
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	WBSV_IF2_SID
EVENT_REPORT	EVENT_EVENT_REPORT	PEAKUP_EVID
EVENT_REPORT	EVENT_EXCEP_REPORT	HIFI_READY_EVID
EVENT_REPORT	EVENT_ALARM_REPORT	PM_TEST_EVID

EVENT_REPORT	EVENT_ALARM_REPORT	DATA_MEMORY_EVID
EVENT_REPORT	EVENT_ALARM_REPORT	EEPROM_EVID
EVENT_REPORT	EVENT_ALARM_REPORT	TC_ERROR_EVID
EVENT_REPORT	EVENT_ALARM_REPORT	LOAD_EEPROM__EVID
EVENT_REPORT	EVENT_ALARM_REPORT	LOAD_DM_EVID
EVENT_REPORT	EVENT_ALARM_REPORT	BOOT_ERROR_EVID
EVENT_REPORT	EVENT_ALARM_REPORT	RUNTIME_ERROR_EVID
EVENT_REPORT	EVENT_ALARM_REPORT	ERR_HK_ASK_ERR_8C
EVENT_REPORT	EVENT_ALARM_REPORT	ERR_HK_ASK_ERR_8D
EVENT_REPORT	EVENT_ALARM_REPORT	ERR_HK_ASK_ERR_8F
EVENT_REPORT	EVENT_ALARM_REPORT	ERR_HK_ASK_WBSH_LIMIT
EVENT_REPORT	EVENT_ALARM_REPORT	ERR_HK_ASK_WBSV_LIMIT
EVENT_REPORT	EVENT_ALARM_REPORT	ERR_HK_ASK_HTRH_LIMIT
EVENT_REPORT	EVENT_ALARM_REPORT	ERR_HK_ASK_HTRV_LIMIT
MEMORY_MANAGEMENT	MEMORY_DUMP_REPORT	0
MEMORY_MANAGEMENT	MEMORY_CHECK_REPORT	0
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	CONF_LCU_REP
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	CONF_SPECTROSCOPY_REP
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	CONF_SPECTROSCOPY_NEW_REP
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	SPEC_FREQ_SWITCH_REP
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	SPEC_FAST_CHOP_REP
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	SPEC_SLOW_CHOP_REP
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	SINGLE_CMD_REP
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	CH1_DHTR_C_REP
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	CV1_DHTR_C_REP
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	HL_SWITCH_ON_REP
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	CONF_FCU_REPORT_NEW
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	CONF_FCU_POW_REPORT_NEW
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	CONF_HRSH_ATT_LO_REP_NEW
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	CONF_HRSH_BLOCKS_REP_NEW
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	CONF_HRSV_ATT_LO_REP_NEW
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	CONF_HRSV_BLOCKS_REP_NEW
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	CONF_WBSH_REP_NEW
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	CONF_WBSV_REP_NEW
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	CONF_LCU_OFF_REP
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	CONF_LCU_INT_REP
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	CONF_LSU_REP_REP
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	CONF_LCU_TABLES_REP
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	CONF_LCU_NOMIN_REP
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	CONF_LCU_DIAGN_REP
FUNC_STATUS_REPORT	FUNC_STATUS_SUBTYPE	CONF_LCU_TABLES_REP
TIME_MANAGEMENT	TIME_VERIFICATION	0
PACKET_TX_CONTROL	PACKET_GEN_REPORT	0
TEST_SERVICE	LINK_CONNECT_REPORT	0
SCIENCE_DATA	SCIENCE_DATA_SUBTYPE	START_HRSH1_SID
SCIENCE_DATA	SCIENCE_DATA_SUBTYPE	START_HRSH2_SID
SCIENCE_DATA	SCIENCE_DATA_SUBTYPE	START_HRSV1_SID
SCIENCE_DATA	SCIENCE_DATA_SUBTYPE	START_HRSV2_SID
SCIENCE_DATA	SCIENCE_DATA_SUBTYPE	START_WBSH1_SID
SCIENCE_DATA	SCIENCE_DATA_SUBTYPE	START_WBSH2_SID
SCIENCE_DATA	SCIENCE_DATA_SUBTYPE	START_WBSV1_SID
SCIENCE_DATA	SCIENCE_DATA_SUBTYPE	START_WBSV2_SID
SCIENCE_DATA	SCIENCE_DATA_SUBTYPE	WBSH1_16BIT_SID
SCIENCE_DATA	SCIENCE_DATA_SUBTYPE	WBSH2_16BIT_SID
SCIENCE_DATA	SCIENCE_DATA_SUBTYPE	WBSV1_16BIT_SID
SCIENCE_DATA	SCIENCE_DATA_SUBTYPE	WBSV2_16BIT_SID

SCIENCE_DATA	SCIENCE_DATA_SUBTYPE	HRSH1_24BIT_SID
SCIENCE_DATA	SCIENCE_DATA_SUBTYPE	HRSH2_24BIT_SID
SCIENCE_DATA	SCIENCE_DATA_SUBTYPE	HRSV1_24BIT_SID
SCIENCE_DATA	SCIENCE_DATA_SUBTYPE	HRSV2_24BIT_SID
SCIENCE_DATA	SCIENCE_DATA_SUBTYPE	WBSH1_24BIT_SID
SCIENCE_DATA	SCIENCE_DATA_SUBTYPE	WBSH2_24BIT_SID
SCIENCE_DATA	SCIENCE_DATA_SUBTYPE	WBSV1_24BIT_SID
SCIENCE_DATA	SCIENCE_DATA_SUBTYPE	WBSV2_24BIT_SID
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	FCU_PAR_SCAN_REP
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	DIP_SCAN_IF_REP
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	DIP_SCAN_NOIF_REP
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	LCU_IV_CURVE_REP
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	ENG_SCAN_REP
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	HRS_H_TUNE_REP
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	HRS_V_TUNE_REP
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	WBS_H_TUNE_REP
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	WBS_V_TUNE_REP
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	MIX_MAG_HRS_REP
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	MIX_MAG_WBS_REP
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	VEC_SCAN_NOMINAL_REP_SID
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	VEC_SCAN_DIAGN_REP_SID
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	VEC_SCAN_REP_SID (14 values)
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	VEC_SCAN_REP_SID
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	PEAKUP_ACQUIRE_REP_SID
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	PEAKUP_CHOP_COR_REP_SID
HK_DIAG_REPORT	HK_DIAG_SUBTYPE	PEAKUP_AOCS_COR_REP_SID

7.9 Service Type 17: Test command

The only command accepted for this service is the HFI_connection_test Type 17, subtype 1 (with no parameters), see section AD3, section 4.6.1.
The purpose of the command is to test the presence of the link connection between ground and the instrument.

When this command is received, the OBS issues a HIFI_Connection_Report ,Type 17, subtype 2 (with no parameters), described in AD4, section 4.8.1.

The following Execution Failures are foreseen:

ERROR Name	Short Mnemonic	Error Code	Notes
NOK_CMDEX_GET_EV_POOL	CMDEX_GET_EVP	0x0629	Get block from Ev Pool failure: no report generation is possible. (applicable to connection test e pkt enable). No command activity is performed onboard.
NOK_CMDEX_PUT_EV_TM_QUEUE	CMDEX_PUT_EVQ	0x062a	FIFO Put on EV TM Queue failure:no possibility to notify to TMTC that there is a report ready to be downloaded. (applicable to connection test e pkt enable). The other activities of the command have been already executed.

7.10 Service Type 21: Science Data Reporting

7.10.1 Diagnostic data reporting: scan and tune reports

Starting from OBS version 4.3, due to the request in SCR 929, the type/subtype of the diagnostic science reports has been changed to type=3, subtype =25, the same as the HK packets. The APID is 1026.
The following table give the references for the AD4 document section in which the diagnostic packets are described.

Description	Type	Subtype	SID	AD4 section	Report content
FCU parameter scan	3	25	266	4.10.1.2	It is a variable length report, depending on the commanded number of steps in the procedure. For each step some of the parameters are reported as they have been received from subsystems.
Diplexer scan report with IF power	3	25	258	4.10.1.3	TBW
Diplexer scan report without IF power	3	25	270	4.10.1.4	TBW
LCU IV curve	3	25	259	4.10.1.5	TBW
HRS_H tune report	3	25	260	4.10.1.6	TBW
HRS_V tune report	3	25	261	4.10.1.7	TBW
WBS_H tune report	3	25	262	4.10.1.8	TBW
WBS_V tune report	3	25	263	4.10.1.9	TBW
mixMagnetCurrent_useHRS report	3	25	267	4.10.1.10	It is a variable length report, depending on the commanded number of steps in the procedure. For each step aprt of the parameters are reported as they have been received from subsystems.
mixMagnetCurrent_useWBS report	3	25	268	4.10.1.11	It is a variable length report, depending on the commanded number of steps in the procedure. For each step aprt of the parameters are reported as they have been received from subsystems
Vector scan report	3	25	273 + channel number as extracted from the TC.	4.10.1.12 (see also SCR 955)	It is a variable length report containing the results of the vector scan activity.
Engineering scan report	3	25	271	4.10.1.14	It is a variable length report containing the results of the engineering scan activity.

Table 26 - Diagnostic packets

7.10.2 Science data reporting: HRS and WBS data packets

The data contained in the Integration Buffers are packed in *Frame*s. In OBS 2 Integration Buffers per each spectrometer, per each polarization, are foreseen, to be used in case of Slow Chop, Frequency Switch and Fast Chop measurements.
The Structure-ID (SID) defines from which Integration Buffer and from which Spectrometer the data originates (The integration buffer is an array inside the ICU).

Each frame consists of a *Start Science Dataframe* packet followed by a certain number of *Science Data Report* Packets. The following table give the references for the AD4 document section in which the Science Data packets are described.

Description	Type	Subtype	SID	APID	AD4 section
HIFI_HRS_H1_start	21	1	1	1028	4.9.3
HIFI_HRS_H2_start	21	1	2	1028	4.9.3
HIFI_HRS_V1_start	21	1	3	1029	4.9.3
HIFI_HRS_V2_start	21	1	4	1029	4.9.3
HIFI_HRS_H1_science24	21	1	17	1028	4.9.5.1
HIFI_HRS_H2_science24	21	1	18	1028	4.9.5.1
HIFI_HRS_V1_science24	21	1	19	1029	4.9.5.1
HIFI_HRS_V2_science24	21	1	20	1029	4.9.5.1
HIFI_WBS_H1_start	21	1	5	1030	4.9.4
HIFI_WBS_H2_start	21	1	5	1030	4.9.4
HIFI_WBS_V1_start	21	1	7	1031	4.9.4
HIFI_WBS_V2_start	21	1	7	1031	4.9.4
HIFI_WBS_H1_science16	21	1	13	1030	4.9.5.2
HIFI_WBS_H2_science16	21	1	14	1030	4.9.5.2
HIFI_WBS_V1_science16	21	1	15	1031	4.9.5.2
HIFI_WBS_V2_science16	21	1	16	1031	4.9.5.2
HIFI_WBS_H1_science24	21	1	21	1030	4.9.5.1
HIFI_WBS_H2_science24	21	1	22	1030	4.9.5.1
HIFI_WBS_V1_science24	21	1	23	1031	4.9.5.1
HIFI_WBS_V2_science24	21	1	24	1031	4.9.5.1

Table 27 - Science Data Reports

7.10.2.1 HRS science data packets.

The following notes are applicable:

4) Each sub-block (258 pixels) of the data frame provided by each spectrometer is packetised into one packet. This means that a full packetisation is made by 1 HRS start science dataframe + 16 HRS science data reports.

5) The range selection is a selection of the sub-blocks to be downloaded. The following rule is applied onboard to decode the HRS range selection word:

```
Select = 0x1 (00000001)    Block 8 is selected (both subblocks)
Select = 0x2 (00000010)    Block 7 is selected (both subblocks)
Select = 0x4 (00000100)    Block 6 is selected (both subblocks)
Select = 0x8 (00001000)    Block 5 is selected (both subblocks)
Select = 0x10(00010000)    Block 4 is selected (both subblocks)
Select = 0x20(00100000)    Block 3 is selected (both subblocks)
Select = 0x40(01000000)    Block 2 is selected (both subblocks)
Select = 0x80(10000000)    Block 1 is selected (both subblocks)
```

The number of packets to be downloaded depends on the number of selected subblocks.

6) HRS data are always packetised as 24 bit data.

7) The commanded coaddition (HIF_N_HRS_INTEGR parameter in the HIFI_Configure_Spectrosopy TC) is applied to all interested Integration Buffers onboard. The first pixel of each sub-block (Configuration pixel) is never coadded.

8) The commanded right shift (HIF_HRS_RSHIFT parameter in the HIFI_Configure_Spectrosopy TC) is never applied to the first pixel of each sub-block (Configuration pixel).

9) Unless explicitly requested with a dedicated TC (See section 7.8), an IF Power HK packet is provided per each packetisation, containing the sub-blocks IF Powers obtained with the packetised frame. The IF Power calculation is made on the coadded/rightshifted data. The HRS IF Power is calculated according to the algorithm described in section 7.3.3.

The following table lists the index of the first channel reported in a nominal HRS science data packet.

Packet #	Block #	First ch.	
		HEX	DEC
1	Block 8 GaAs1	0x0000	0
2	Block 8 GaAs2	0x0102	258
3	Block 7 GaAs1	0x0204	516
4	Block 7 GaAs2	0x0306	774
5	Block 6 GaAs1	0x0408	1032
6	Block 6 GaAs2	0x050A	1290
7	Block 5 GaAs1	0x060C	1548
8	Block 5 GaAs2	0x070E	1806
9	Block 4 GaAs1	0x0810	2064
10	Block 4 GaAs2	0x0912	2322
11	Block 3 GaAs1	0x0A14	2580
12	Block 3 GaAs2	0x0B16	2838
13	Block 2 GaAs1	0x0C18	3096
14	Block 2 GaAs2	0x0D1A	3354
15	Block 1 GaAs1	0x0E1C	3612
16	Block 1 GaAs2	0x0F1E	3870

Table 28 – HRS first channels reported in science data packets

7.10.2.2 WBS science data packets.

The following notes are applicable:

1) Each one of the specified ranges is packetised separately. No range concatenation is foreseen. A variable number of packets per range is allowed.

2) The total number of packets depends on the range selection and is given by the sum of the packets necessary to packetise the selected data of each CCD.

3) The total number of packets depends on the selected packetisation (16 or 24 bits packing). In case of 16 bit packing a maximum of 494 data per packet is allowed, in case of 24 bits packing the maximum number of data is 328.

4) This means that, in case of full range selected (2028 data/CCD, 4 CCDs), the following number of packets are expected:
   - WBS (1 polarization) 16 bit packing:
     1 IF Power HK packet
     1 start frame SD packet
     20 Science data packets, 5 packets per CCD (4 full length + 1 short packet);

- WBS (1 polarization) 24 bit packing:
  1 IF Power HK packet
  1 start frame SD packet
  28 Science data packets, 7 packets per CCD (6 full length + 1 short packet);

If all spectrometers are ON (included HRS), the total number of packets expected is:
a) (22+18)*2 = 80packets/packetisation (in case of WBS 16bit data)
b) (30+18)*2 = 96packets/packetisation (in case of WBS 24bit data);

5) The range selection allows to select 4 ranges of contiguous channels. For each range the first pixel absolute coordinate and the number of pixels in the range shall be specified (See section 7.6.12.1).
The WBS data frame is stored onboard as a contiguous array of 8192 data. The absolute coordinates of the pixels belonging to each CCD are reported below:
0 - 2047 CCD1
2048 – 4095 CCD2
4096 – 6143 CCD3
6144 – 8191 CCD 4
There is no check on the positioning of the 4 selected ranges in the array. This means that in principle more than one range belonging to the same CCD can be selected, as well as a range containing data in the overlapping region between two CCDs.

6) The commanded coaddition (HIF_N_WBS_INTEGR parameter in the HIFI_Configure_Spectroscopy TC) is applied to all interested Integration Buffers onboard.
7) The commanded right shift (HIF_WBS_RSHIFT parameter in the HIFI_Configure_Spectroscopy TC) is never applied to the first 4 pixels of each sub-block (dark current pixels).
8) Unless explicitly requested with a dedicated TC (See section 7.8), per each packetisation an IF Power HK packet is provided, containing the IF Powers obtained for the packetised frame. The IF Power calculation is made on the coadded/rightshifted data. The WBS IF Power is calculated according to the algorithm described in section 7.3.3.

A1. Appendix - OBS Error Codes

Table 29- OBS generated event reports for Runtime errors (Event ID = 0xa000)

ERROR Name	Short Mnemonic	Error Code	Notes	Severity
ERR_HS0_HSFLUSH_FOUND_NOBLOCK	NOBLOCK_1	0x0101	No Blocks found in HW FIFOs. It is an indication that there was a problem in receiving the data from the spectrometers at the end of a measurement.	4
ERR_HS0_FIFOPUT_FRAME_QUEUE	FRAME_QUEUE	0x0102	FIFO put on FRAME queue failed in HS0 task: problem is transmitting an already aqcuired frame from the task hs0 to the task hs1. The frame is lost.	2
ERR_HS0_FRAME_POOL_ID	POOL_ID	0x0103	Invalid FIFO Id in the procedure to deliver the acquired frame to HS1. No frame delivered. Frame is lost. 1 parameter: FIDO ID.	2
ERR_HS0_FIFO_HALF_FULL	FIFO_HF	0x0104	Warning in HS Flush task. It was triggered even in case of interrupt pending. The HW interrupt is served and teh task is rescheduled.	2
ERR_HS0_FIFO_NOT_EMPTY	FIFO_NE	0x0105	Warning in HS Flush task. Ther are still data present in the FIFO after all expected data have been read. Possible misalignments in the following frames data.	2
ERR_HS0_FIFO_OVFL	FIFO_OVFL	0x0106	FIFO FULL signal detected. Spectrometer data lost	4
ERR_HS0_TS_FIFO_VOID	TS_FIFO_EMPTY	0x0107	Time Stamps FIFO Empty	4
ERR_HS0_TS_FIFO_FULL	TS_FIFO_FULL	0x0108	Time Stamps FIFO Full	4
ERR_HS0_TS_POOL_FULL	TS_POOL_EMPTY	0x0109	Time Stamps Pool Empty	4
ERR_HS0_TS_POOL_ERR	TS_POOL_FULL	0x010A	Time Stamps Pool Full	4
ERR_HS0_LS_HP_QUEUE	LSQ_ERR	0x010B	Could not write to LS queue	
ERR_HIFI_POOL_FULL	HIFP_FULL	0x0200	HIFI pool get failed in HS0 task	2
ERR_HIFI_POOL_ID	HIFP_ID	0x0201	Wrong HIFI Pool ID in HS0/1 task 1 parameter: Pool ID	2
ERR_HIFI_POOL_BLOCK_NUMBER	HIFP_BLOCK	0x0202	Wrong block number in get/free block operations on HIF Pool	2
ERR_HS1_SPECT_AID	INV_AID	0x0300	Invalid AID spectroscopy in HS1 task..	2
ERR_HS1_FIFOPUT_SD_PKT	PUT_SD_PKT	0x0302	Error in FIFO PUT. SD packet lost	2
ERR_HS1_FRAME_COMPLETE_FAIL	FPUT_FC	0x0303	Error in FIFO PUT. Completed Frames counter wrong.	2
ERR_HS1_SCAN_COUNT	SC_ZERO	0x0304	Scan Count equal to zero detected. SD packet lost.	2
ERR_HS1_SD_GET_BLOCK	SD_GET_FAIL	0x0305	Failure in getting a SD packet block from pool.	4
ERR_HK_ASK_FIFOPUT_HK_TM_QUEUE	HKASK_PUT_HKTMQ	0x0400	FIFO put on HK TM queue failed: the Hk packet is lost.	2
ERR_HK_ASK_GET_BLOCK_HKPOOL_FULL	HKPOOL_FULL	0x0402	No Blocks available on Hk Pool.	2
ERR_HK_ASK_SWERR	HKASK_SWERR	0x0403	Error in the parameters to execute limit checking. 1- 1 parameter. Unespected FCU HK table length	2
ERR_HK_ASK_PKT_NO	HKASK_PKTNO	0x0405	Unespected number of nominal HK parameters	2
ERR_HK_ASK_INTERNAL_ERROR	HKASK_INTERR	0x040A	Unespected number of nominal/periodic FCU HK.	2
ERR_HK_ASK_LS_HRS_REQUEST	HKASK_HRSREQ	0x040C	FIFO put on LS HP queue failed for HRS data transfer cmd.	2
ERR_HK_ASK_FIFOPUT_LS_HP_QUEUE	HKASK_LSQUEUE	0x040E	FIFO put on LS HP queue failed for LCU HK request	2
ERR_LS_QUEUES_EMPTY	LSQ_EMPTY	0x0500	FIFO get failed on both LS high priority and low priority queues	2
ERR_LS_FIFOPUT_HP_QUEUE	LSLPQ	0x0502	FIFO put on LS high priority queue failed	1
ERR_LS_FIFOPUT_LP_QUEUE	LSHPQ	0x0503	FIFO put on LS low priority queue failed	1
ERR_LS_TRX_REG_BUSY	TRX_BUSY	0x0504	TX Busy bit asserted on LS I/F register	1
ERR_LS_INCORRECT_WORD	LS_BAD_WORD	0x0505	The cmd/hk request to be transmitted via the Low Speed I/F is not correct	1
ERR_LS_FIFOPUT_LS_HDL_QUEUE	LS_LSHDL_Q	0x0506	Error in transmitting end of transaction message (LCU IV Curve) Report not transmitted. Measurement not fully executed.	2

ERR_LS_FIFOPUT_HS_HDL_QUEUE	LS_HSHDL_Q	0x0507	Error in transmitting end of transaction message (Engineering Scan) Report not transmitted. Measurement not fully executed	2
ERR_CMDEX_TRX_CTRL_INVALID_TYPE	PKT_TYPE	0x0601	Packet transmission control command: invalid commanded type	2
ERR_CMDEX_TRX_CTRL_INVALID_SUBTYPE	PKT_SUBTYPE	0x0602	Packet transmission control command: invalid commanded subtype	2
ERR_CMDEX_TRX_CTRL_INVALID_PID	PKT_ID	0x0603	Packet transmission control command: invalid commanded packet ID	2
ERR_CMDSEQ_TC_ACCEP	TCACC_ERR	0x0610	Failure in the Telecommand acceptance procedure. Problems in enqueuing the TC acc. failure report. The TC is not executed.	2
ERR_CMDEX_PUT_HS_HDL_QUEUE	CMD_EX_PUT	0x062d	Failure in putting start Abort message.	4
ERR_DATA_HDL_ZERO_DIV	ZERO_DIV_0	0x0701	Division by zero	2
ERR_DATA_HDL_FIFOPUT_HK_TM_QUEUE	DATAH_PUT_HKTMQ	0x0702	FIFO put on HK TM queue failed in Data Hdl task. IF Power HK packet lost.	2
ERR_DATA_HDL_FIFOPUT_SD_TM_QUEUE	PUT_SDTMQ	0x0703	FIFO put on SD TM queue failed in Data Hdl task	2
ERR_DATA_HDL_FIFO_ID	FIFO_ID	0x0704	Invalid FIFO ID in DATA_HDL task	2
ERR_DATA_HDL_GET_STF_POOL_BLOCK	STF_GETB	0x0705	Get block from Start frames Pool failure (no start frame packet)	2
ERR_DATA_HDL_GET_HK_POOL_BLOCK	HKP_GETB	0x0706	Get block from HK Pool failure (no If power packet produced)	2
ERR_DATA_HDL_INVALID_MEAS_ID	WRONG_AID	0x0707	Not Valid AID spectroscopy in Data Handler	2
ERR_DATA_HDL_HRS_DURATION_NULL	DHDL_HRS_DIV0_1	0x0709	Division by zero error in HRS Ifpower calculation.	2
ERR_DATA_HDL_IFP_OVFL	DHDL_HRS_IFPOVFLW	0x070B	HRS Ifpower calculation: max range of 16 bit exceeded.	2
ERR_DATA_HDL_IDX_OVFL	DHDL_WBS_RANGEOOL	0x070C	WBS range selection incompatible with max number of WBS pixels	2
ERR_HS_HDL_WRONG_SPECT_AID	WSPEC_AID	0x0800	Current value of AID spectroscopy different from the one necessary to the running function	2
ERR_HS_HDL_MIXMAG_STEPS	MMAG_STEPS	0x0802	Mixer magnet tuning: mixer magnet total number of steps exceeded 1 parameter: total number of steps	2
ERR_HS_HDL_OPERATION_ID	HSHDL_OPID	0x0803	Invalid selected operation ID in HS_HDL task	4
ERR_HS_HDL_FIFO_0_OOL	FIFO_0_OOL	0x0805	Unespected number of Fifo 0 pending frames: 1 parameter: number of pending frames	2
ERR_HS_HDL_FIFO_1_OOL	FIFO_1_OOL	0x0806	Unespected number of Fifo 1 pending frames 1 parameter: number of pending frames	2
ERR_HS_HDL_FIFO_2_OOL	FIFO_2_OOL	0x0807	Unespected number of Fifo 2 pending frames 1 parameter: number of pending frames	2
ERR_HS_HDL_FIFO_3_OOL	FIFO_3_OOL	0x0808	Unespected number of Fifo 3 pending frames 1 parameter: number of pending frames	2
ERR_HS_LIB_INVALID_SPECTR_ID	INV_SPEC_ID	0x0900	Inavlid ID to initialise the spectroscopy table necessary to execute a tuning procedure. The procedure will be started with the deafult table	2
ERR_HS_LIB_FIFOPUT_HK_TM_QUEUE	HK_TM_QUEUE_2	0x0901	FIFO put on HK TM queue failed in HS_HDL task. Tune reports lost.	2
ERR_HS_LIB_BRING_SAFE_TIMEOUT	HSLIB_SAFETIMOUT	0x0902	The procedure to check if HS is in a safe status before starting a new activity failed in getting the status within 3secs. An error report is generated and the new activity is started.	2
ERR_HS_LIB_ZERO_DIV	ZERO_DIV	0x0904	Division by zero in HS library function	2
ERR_HS_LIB_WRONG_SPECT_AID	WSPECT_AID	0x0918	Inconsistency between current and espected AID spectroscopy	2
ERR_HS_LIB_QUEUE_FAIL	HSLIB_HSHDL_Q	0x0919	HS_HDL queue handling failure. A message to execute the next step of the measurements has not been enqueued.	2
ERR_HS_LIB_ERROR_LS_HP_QUEUE	HSLIB_LSHP_Q	0x0927	LS high priority queue handling failure. The tuned parameters have not been commanded.	2
ERR_HS_LIB_GET_HK_POOL_BLOCK	HSLIB_HKPBLK	0x0928	Get block from HK Pool failure. Tuning has been executed but the tune report will not be genefated.	2
ERR_HS_LIB_AID_SPECTR_CHANGED	HSLIB_WAIDSP	0x0929	Inconsistency between current and espected AID spectroscopy in measurement preparation. 1 parameter: AID spectroscopy	2
ERR_HS_LIB_NUMERICAL_OVFL	HSLIB_NUM_OVF	0x0930	Error in WBS tuning procedure. Correction factor <0.	2
ERR_HS_LIB_AID_STOPPING_VM	HSLIB_WAIDSP2	0x0931	Inconsistency between current and espected AID	2

			spectroscopy in LS measurement preparation.	
ERR_HS_LIB_HRS_DURATION_NULL	HSLIB_HRS_DIV0_1	0x0941	Division by zero error in HRS Ifpower calculation.	2
ERR_HS_LIB_SUBSYSID	HSLIB_HRS_WID	0x0944	Before starting the HRS tuning, a worng subsystem ID was passed to the procedure for saving the instrument status.	2
ERR_HS_LIB_FAILED_TO_RESTORE	HSLIB_HRSRESTORE	0x0945	At teh end of HRS tuning: error in restoring resulution mode.	2
ERR_HS_LIB_DARK_OVFL	HSLIB_DARK	0x0947	Overflow in dark current	
ERR_HS_LIB_NEW_CAL_STEP	HSLIB_STEP	0x0955	Unknown step in WBS calibrate procedure	
ERR_MEM_FIFOPUT_HK_TM_QUEUE	MEM_PUT_HKTMQ	0x1005	FIFO put on HK TM queue failed in Memory management task. Dump/check report lost.	2
ERR_MEM_DATA_OVFL	MEM_DATAOVFL	0x1006	Failure in memory manager initialisation.	2
ERR_MEM_FIFOPUT_SD_PKT_QUEUE	MEM_SD_PKT_Q	0x1009	FIFO put onSD TM queue failed in Memory management task.	2
ERR_MEM_INTERNAL	MEM_internal	0x1016	error in generating memory dump report: a wrong RAM type has been detected.	2
ERR_RESCK_DM_FAIL	DM_FAIL	0x1100	Failure in check DM routine: 1 32bit parameter: damaged memory cell address	4
ERR_RESCK_DM_WRAP	DM_WRAP	0x1101	Debug message to check correct wraparound of memory addresses in check DM routine	1
ERR_RESCK_TC_POOL_OVERFLOW_WARN	TCP_OVF	0x1102	WARNING. TC Pool Overflow	1
ERR_RESCK_SD_POOL_OVERFLOW_WARN	SDP_OVF	0x1103	WARNING. SD Pool Overflow	4
ERR_RESCK_HK_POOL_OVERFLOW_WARN	HKP_OVF	0x1104	WARNING. HK Pool Overflow	4
ERR_RESCK_DM_CK_STOPPED	RESCHK_DMCHK	0x1106	ERROR. DM check activity suspended because of too many memory faults.	4
ERR_ERR_HDL_GET_HK_POOL_BLOCK	ERRHDL_GET_HKB	0x1301	Get block from HK pool failed duriing a report generation. The report is lost.	2
ERR_ERR_HDL_FIFOPUT_HK_TM_QUEUE	ERRHDL_PUT_HKQ	0x1302	FIFO put error in HK TM queue. duriing a report generation. The report is lost.	2
ERR_ERR_HDL_SUPPRESSING_EVENTS	ERRHDL_EVQUEUE	0x1303	Event queue overflow. Event generation suspended until overflow is recovered (all messages have been spooled).	2
ERR_LS_HDL_LCU_CRC_MISMATCH	LSHDL_CRCM	0x2023	The LCU CRC is wrong	
ERR_TMTC_TRX_CTRL_PKT_NOT_FOUND	TXC_PKT_NF	0x4000	Packet transmission control command; combination (type, subtype, pack_id) not found. 3 parameters: received type, subtype, pack_id	2
ERR_TMTC_INVALID_PACKING_FIELD	TMTC_PACK	0x4001	Invalid SD packing Field. SD packet lost	2
ERR_TMTC_APID_IN_GET_COUNTER	TMTC_APID	0x4002	Failure in getting APID counter.	2
ERR_PF_GET_BLOCK_FAIL	PF_GB_ERR	0x5001	Error in getting a block from a pool	
ERR_PF_GET_BLOCK_UNKNOWN_POOLID	PF_GB_FAIL	0x5002	Error in getting block from memory pool.	2
ERR_PF_REL_BLOCK_UNKNOWN_POOLID	PF_RB_FAIL	0x5003	Error in releasing blocks from pool	2
ERR_PF_LCU_CRC	PF_CRC	0x5008	Error in LCU CRC	

Table 30 Execution Failures Error codes and description

ERROR Name	Short Mnemonic	Error Code	Notes
EXF_CMDSEQ_UNKNOWN_ERROR	CMDEX_UNK	0x0012	An unknown return code has been received from a procedure
EXF_HK_ASK_FIFOPUT_HK_TM_QUEUE	HKASK_FPUT	0x0401	Execution failure: FIFO put on HK TM queue failed
EXF_HK_ASK_HK_RATE_INDEX_OOL	HKASK_NORATE	0x0404	Exec. Failure: Requested rate not available.
EXF_HK_ASK_SINGLE_HK_LCU_INVALID_DATA	HKASK_LCUSHK	0x0406	Exec. Failure: Consistency check of parameters in LCU_single HK request failed.
EXF_HK_ASK_ERROR_LS_HP_QUEUE	HKASK_LSHP_FPUT	0x0407	Exec. Failure:FIFO put on LS HP queue failed.
EXF_HK_ASK_LCU_OFF	HKASK_LCU_OFF	0x0408	Exec. Failure: LCU single Hk not executable because subsystem is OFF
EXF_HK_ASK_FCU_OFF	HKASK_FCU_OFF	0x0409	Exec. Failure: LCU single Hk not executable because subsystem is OFF.
EXF_HK_ASK_MEM_HK_LCU_INVALID_DATA	HKASK_LCUDUMP	0x040B	Exec. Failure: commanded mem dump parameters out of limits
EXF_HK_ASK_SINGLE_HK_LCU_DBG_MSG	HKASK_LCUDBG	0x040D	Exec. Failure: internal inconsistency between table lenghts
EXF_HS1_DELIVER_SD_PKT	HS1_SD_LOST	0x0306	Failure in delivering the packet to DATA HDL. SD Packet lost
NOK_CMDEX_TRX_CTRL_LENGTH_CHECK	CMDEX_LEN_CHK	0x0600	Packet transmission control command: inconsistency between the packet length and the commanded number of enable/disable operations.
NOK_CMDEX_TRX_CTRL_TC_WRONG_DATA	CMDEX_PKT_DATA	0x0604	Packet transmission control command: execution failure due to wrong application data
EXF_CMDEX_PAR_SCAN_WRONG_STEP_NUMBER	CMDEX_PSC_STEP	0x0605	The calculated total number of steps (HIF_NVOLTAGE*HIF_NMAGNET) is greater than the maximum allowed (80) 2 parameters: HIF_NVOLTAGE, HIF NMAGNET.
EXF_CMDEX_PAR_SCAN_WRONG_NVOLTAGE	CMDEX_PSC_VOLT	0x0606	Type (8,4) FCU parameter scan: execution failure due to wrong N Voltage
EXF_CMDEX_PUT_LS_HDL_QUEUE	CMDEX_PUT _LSHQ	0x062c	Failure in passing data to LS_HDL to start the requested activity.
EXF_CMDEX_PUT_HS_HDL_QUEUE	CMDEX_PUT _HSHQ	0x062e	Failure in passing data to LS_HDL to start the requested activity.
EXF_CMDEX_GET_BLOCK_TC_POOL	CMDEX_GET_TCPL	0x062f	Failure in preparing data for LS_HDL to start the requested activity.
NOK_CMDEX_ILLEGAL_WBS_ACC_TIME	CMDEX_WBSTIMW	0x0626	Wrong Commanded WBS Time.
NOK_CMDEX_ERROR_LS_HP_QUEUE	CMDEX_LSHP_Q	0x0627	Failure in sending out one of the commanded configuration parameters.
NOK_CMDEX_NYI_TC_ERR	CMDEX_NYI	0x0628	A not yet implemented functionality has been requested.
NOK_CMDEX_GET_EV_POOL	CMDEX_GET_EVP	0x0629	Get block from Ev Pool failure: no report generation is possible. (applicable to connection test e pkt enable). No command activity is performed onboard.
NOK_CMDEX_PUT_EV_TM_QUEUE	CMDEX_PUT_EVQ	0x062a	FIFO Put on EV TM Queue failure:no possibility to notify to TMTC that there is a report ready to be downloaded. (applicable to connection test e pkt enable). The other activities of the command have been already executed.
NOK_CMDEX_PUT_HK_TM_QUEUE	CMDEX_PUT_HKTMQ	0x062b	FIFO Put on HK TM Queue failure:no possibility to notify to TMTC that there is a report ready to be downloaded. (applicable to enable time verification). The other activities of the command have been already executed.
EXF_CMDEX_GET_BLOCK_HK_POOL	CMDEX_GETHKBLOCK	0x0630	FIFO get on HK pool failure: :no possibility to get the block for storing the time verification report. The Enable Time verification Tc is not executed.
EXF_DATA_HDL_INVALID_SUBSYS_ID	DHDL_SUBSYSID	0x0700	Invalid measurement ID in data handler task
EXF_DATA_HDL_FIFOPUT_HSHDL_QUEUE	DHDL_HSHDLQ	0x0708	FIFO Put on HS HDL Queue failure:no possibility to notify to HS HSLthat one of the steps of the tuning procedure is completed. The procedure is stopped. Applicable to all tuning procedures.

EXF_HS_HDL_ERROR_LS_HP_QUEUE	HSHDL_LSHPQ	0x0809	FIFO Put on LS HP queue failure: no possibility to trnasmit to LS task one of the command of the procedure. The procedure is stopped. Applicable to all tuning procedures.
EXF_HS_HDL_HRS_SUB_OFF	HSHDL_HRS_OFF	0x0810	A diplexer scan with IF power is requested for a switched off spectrometer. The procedure is not started
EXF_HS_HDL_FCU_SUB_OFF	HSHDL_FCU_OFF	0x0811	A diplexer scan with IF power is requested but FCU is off. The procedure is not started
EXF_HS_HDL_WRONG_SPECT_AID	HSHDL_SPCAID	0x0812	Erroneous Spectroscopy AID
EXF_HS_LIB_ERROR_LS_HP_QUEUE	HSLIB_LSHPQ	0x0905	FIFO Put on LS HP queue failure: no possibility to trnasmit to LS task one of the command of the procedure. The procedure is stopped. Applicable to all tuning procedures and to peakup procedures.
EXF_HS_LIB_HRS_SUB_OFF	HSLIB_HRS_OFF	0x0906	A measurement involving HRS has been requested, but the spectrometers are OFF. No measurement is started.
EXF_HS_LIB_UPLOAD_VM_PARAM_TYPE	HSLIB_UPLVM_PAR	0x0907	Error in parameters of VM Upload TC.
EXF_HS_LIB_UPLOAD_VM_CODE_AREA	HSLIB_UPLVM_ARE	0x0908	Error in parameters of VM Upload TC
EXF_HS_LIB_UPLOAD_VM_INCONSISTENT_TC	HSLIB_UPLVM_TC	0x0909	Error in parameters of VM Upload TC
EXF_HS_LIB_START_VM_INVALID_INDEX	HSLIB_STVM_IDX	0x0910	Error in parameters of VM Upload TC
EXF_HS_LIB_WBS_SUB_OFF	HSLIB_WBS_OFF	0x0911	Requested measurement cannot be executed. The spectrometer is off
EXF_HS_LIB_QUEUE	HSLIB_QUEUE	0x0912	FIFO Put on HS HDL Queue failure:no possibility to notify to HS HSLthat one of the tuning procedure is cstarted. The procedure is aborted. Applicable to all tuning procedures.
EXF_HS_LIB_FCU_SUB_OFF	HSLIB_FCU_OFF	0x0913	A measurement involving FCU has been requested, but the spectrometers are OFF. No measurement is started (applicable to scans and to mixer magnet tuning)
EXF_HS_LIB_DIPSCAN_NUM_STEP	HSLIB_DIPSCAN_NS	0x0914	The commanded number of steps for the diplexer scan is greater than the maximum allowed. The scan is not staretd.
EXF_HS_LIB_MIXMAG_DATA	HSLIB_MIXMAG_D	0x0915	The commanded number of steps for the mixer magnet tuning is greater than the maximum allowed. The procedure is not started.
NOK_HS_LIB_WRONG_INPUT_PARAM	HSLIB_INPUT_PAR	0x0916	Type (8,4) Start spectroscopy measurement: execution failure due to inconsistency between measurement parameters (coaddition and number of integrations)
EXF_HS_LIB_NO_INSTRU_INVOLVED	HSLIB_NO_INSTR	0x0917	Type (8,4) Start spectroscopy measurement: execution failure due to inconsistency between measurement parameters (spectrometers involved and spectrometers switched on)
EXF_HS_LIB_PEAKUP_POLAR_TYPE_PAR	HSLIB_PKUP_PAR2	0x0922	Error in input parameter of peak up routine.
EXF_HS_LIB_PEAKUP_SEQ_NUMBER_PAR	HSLIB_PKUP_PAR3	0x0923	Error in input parameter of peak up routine.
EXF_HS_LIB_PEAKUP_INCOMPLETE	HSLIB_PKUP_INC	0x0925	Error in execution of peakup. Not all 9 points have been successfully acquired.
EXF_HS_LIB_ENGSCAN_INVALID_INTERVAL	HSLIB_ESCAN_INTV	0x0933	Error in input parameter of enginnering scan request.
EXF_HS_LIB_ENGSCAN_INVALID_NSAMP_1	HSLIB_ESCAN_NS1	0x0934	Error in input parameter of enginnering scan request.
EXF_HS_LIB_ENGSCAN_INVALID_NSAMP_2	HSLIB_ESCAN_NS2	0x0935	Error in input parameter of enginnering scan request.
EXF_HS_LIB_ENGSCAN_NO_VALID_ADDR	HSLIB_ESCAN_ADD1	0x0936	Error in input parameter of enginnering scan request.
EXF_HS_LIB_ERROR_LS_LP_QUEUE	HSLIB_LSLPQ	0x0937	
EXF_HS_LIB_ENGSCAN_WRONG_ADDR	HSLIB_ESCAN_ADD2	0x0938	Error in input parameter of enginnering scan request.
EXF_HS_LIB_ENGSCAN_WRONG_CMD	HSLIB_ESCAN_CMD	0x0939	Error in input parameter of enginnering scan request.
EXF_HS_LIB_WRONG_WBS_ACCTIME	HSLIB_WBS_ACCT	0x0940	Error in input parameter of totalpower spectroscopy. WBS accumulation time too short.
EXF_HS_LIB_INVALID_DATA	HSLIB_FUNCTEST	0x0943	Error in input parameter of HRS funct test. HRS func test ID wrong
EXF_HS_LIB_LCU_NON_INTERACTION	HSLIB_LCU_NI	0x0946	Attempt to use LCU while LCU non interaction is on
EXF_HS_LIB_PEAKUP_NEGATIVE_POWER	HSLIB_PK_NEG	0x0949	Peakup numerical problems: power is negative
EXF_HS_LIB_PEAKUP_FOUND_MINIMUM	HSLIB_PK_MIN	0x0950	Peakup numerical problems: min and not max found
EXF_HS_LIB_PEAKUP_OUT_OF_GRID	HSLIB_PK_GRI	0x0951	Peakup numerical problems: pointing is out of grid
EXF_HS_LIB_PEAKUP_FOUND_NEGATIVE_PEAK	HSLIB_PK_NEP	0x0952	Peakup numerical problems: peak is negative

EXF_HS_LIB_PEAKUP_Z_OVERFLOW	HSLIB_PK_ZOV	0x0953	Peakup numerical problems: Z overflow
EXF_HS_LIB_PEAKUP_Y_OVERFLOW	HSLIB_PK_YOV	0x0954	Peakup numerical problems: Y overflow
EXF_MEM_INVALID_MEMLENGTH	MEM_LEN	0x1000	The commanded length in SAU is inconsistent with the number of 16 bit words of the parameter.
EXF_MEM_INVALID_ADDRESS	MEM_ADD	0x1001	This error is caused by a start address greater than the end address reported in **Error! Reference source not found.**
EXF_MEM_INVALID_MEMID	MEM_ID	0x1002	The commanded memory ID is referred to a not existing ID
EXF_MEM_INVALID_CRC1	MEM_CRC1	0x1003	Inconsistency between the OBS computed checksum and the checksum contained in the TC. Both Checksum are referred to the memory data in the TC.
EXF_MEM_INVALID_CRC2	MEM_CRC2	0x1004	Inconsistency between the OBS computed checksum on the memory data after their copy on the destination memory and the checksum contained in the TC.
EXF_MEM_BOOT_OVERWRITE	MEM_BOOT_OVRW	0x1007	A PM start address shorter that 0x4000 has been requested.
EXF_MEM_UNKNOWN_RET_CODE	MEM_UKWN_RETC	0x1008	Error in memory load. The procedure returned an unespected value.
EXF_MEM_EEPROM_WRITE_OVERFLOW	MEM_EEWRITE_OVF	0x100a	Error during the EEPROM write procedure (to copy in EEPROM the full OBS image).
EXF_MEM_EEPROM_COPY_FAILED	MEM_EECOPY_FAIL	0x100b	Error during the EEPROM write procedure (to copy in EEPROM the full OBS image).
EXF_MEM_EEPROM_WRONG_PARTITION	MEM_EE_PARTIT	0x100c	Error during the EEPROM write procedure: wrong commanded partition ID
EXF_MEM_EEPROM_WRONG_NUM_BAD_PAGES	MEM_EE_BADNPAG	0x100d	Error during the EEPROM write procedure: wrong commanded number of pages to avoid to copy on.
EXF_MEM_HSHDL_QUEUE	MEM_TQUEUE	0x100e	FIFO put on hs_hdl queue failure during memory dump or memory check.
EXF_MEM_COPY_OBS	MEM_COPY_OBS	0x100f	Failure in the execution of the Copy OBS command.
EXF_MEM_CHECK_PM_CRC	MEM_CK_PM_CRC	0x1010	The check PM command resulted in a PM CRC different from expected, indicating a possible damage in the PM memory.
EXF_MEM_EEPROM_WRONG_BAD_PAGE_ID	MEM_EE_BADPAGID	0x1011	Error during the EEPROM write procedure: one of the IDs of pages to avoid to copy on is wrong
EXF_MEM_LCU_INVALID_DATA	MEM_LCU_DATA	0x1012	Error in LCU mem Load TC parameters: total number of loadable words exceeded.
EXF_MEM_LCU_OFF	MEM_LCU_SSOFF	0x1013	Error in LCU mem Load/dump TC: LCU subsystem is OFF.
EXF_MEM_ERROR_LS_HP_QUEUE	MEM_LSHP_UEUE	0x1014	Error in LCU mem Load/dump TC: failure in enqueuing a message to the LS HP queue
EXF_ERR_HDL_MEASUREMENT_ABORTED	ERRHDL_MEAS_ABO	0x1300	Measurement execution Aborted.
EXF_LS_HDL_FCU_OFF	LSHDL_FCU_OFF1	0x2001	Sweep Diplexer and FCU parameter scan procedures in LS_HDL task: execution failure due to FCU off
EXF_LS_HDL_LCU_OFF	LSHDL_LCU_OFF_2	0x2002	Commanded LCU parameter scan: execution failure due to LCU off
ERR_LS_HDL_WRONG_CMD	LSHDL_BADCMD	0x2003	Invalid selected operation ID in LS_HDL task
EXF_LS_HDL_VEC_SCAN_TABLE	LSHDL_BAD_VSTAB	0x2004	Invalid Length of the table onboard used to execute the vector scan
EXF_LS_HDL_GEN_REP_FAIL	LSHDL_GEN_REP	0x2005	Generate report error: no final (or step) report can be generated at the end of one of the scan functions.
EXF_LS_HDL_ERROR_LS_HP_QUEUE	LSHDL_LSHPQ	0x2006	FIFO Put on LS HP queue failure: no possibility to trnasmit to LS task one of the command of the procedure. The procedure is stopped. Applicable to all scan procedures and to LO tuning.
EXF_LS_HDL_FCU_SUB_OFF	LSHDL_FCU_OFF_3	0x2007	Diplexer scan procedure in LS HDL: no execution started due to FCU off.
EXF_LS_HDL_VSCAN_REP_GET_HK_BLOCK	LSHDL_VSCAN_HKB	0x2008	Get block from HK Pool failure: no vector scan report can be produced. The procedure has been completed but the data cannot be downloaded.
EXF_LS_HDL_VSCAN_REP_PUT_HK_QUEUE	LSHDL_VSCAN_HKQ	0x2009	FIFO put on HK TM queue failure: : no vector scan report can be produced. The procedure has been completed but the data cannot be downloaded.
EXF_LS_HDL_VSCAN_OVFL	LSHDL_VSCAN_OVF	0x200A	Error in parameters of vector scan.
EXF_LS_HDL_VSCAN_SYSOFF	LSHDL_VSCAN_SOFF	0x200b	FCU or LCU are switched off. Vector scan cannot be xecuted.
EXF_LS_HDL_DIPSCAN_NUM_STEP	LSHDL_DSCAN_NS	0x200c	Wrong number of steps in diplexer scan command.

EXF_LS_HDL_PUT_LS_HDL_QUEUE	LSHDL_LSHDL_Q	0x2014	FIFO put on LS HDL queue failure: it is not possible to execute the following steps of the procedure. Applicable to FCU parameter scan, vector and diplexer scans.
EXF_LS_HDL_VEC_SCAN_CONF_EXPIRED	LSHDL_VS_CONF	0x2016	Scan tryed without configuration refresh
EXF_LS_HDL_VEC_SCAN_INTERNAL	LSHDL_VS_INT	0x2018	Internal error
EXF_LS_HDL_VSCAN_REPORT_PROBLEMS	LSHDL_VS_REN	0x2019	Internal error
EXF_LS_HDL_LCU_NOT_NORM	LSHDL_LCUN	0x2021	LCU not back to normal status
EXF_LS_HDL_LAST_INVALID	LSHDL_LAST	0x2022	Could not retune because last LCU macro is invalid

A2. Appendix - Relevant files

OBS.vpf:

```
% Virtuoso Project File C:\Documents and Settings\Anna\Desktop\LP\OBS.vpf
% Generated by Sysgen Backend version 4.1 R2.04 on Thu Sep 28 15:53:31 2006


% GLOBALPAR NAME          PARVALUE
% ===================================
  GLOBALPAR DATALEN       16384
  GLOBALPAR TICKFREQ      1000
  GLOBALPAR CEILING_PRIO  5
  GLOBALPAR KERNEL_PRIO   0
  GLOBALPAR DRIVER_PRIO   0

% NLIFILE NLIFILEPATH
% ===================================
  NLIFILE 'OBS.nli'

% NODE NAME          NLINAME      NDPACKS NCPACKS NTIMERS KSTACK MONITSIZE MONITMASK
% ================================================================================
  NODE NODE1         NODE1              0      16      40    256      1024        15

% DRIVERTYPE      NODE          CALL
% ============================================================
  TIMERDRIVER     NODE1         'timer_driver(TMZLI)            '
  USERDRIVER      NODE1         'StackOverFlowChecker(NULL)     '

% NETLINK NODE1          CALL1                              DIR NODE2       CALL2
%
================================================================================================================
======

% TASKGROUP NAME          STARTUP
% ================================
  TASKGROUP EXE                 1
  TASKGROUP SYS                 1
  TASKGROUP FPU                 0
  TASKGROUP WAIT                0
  TASKGROUP EXE_NOBOOT          0

% TASK NAME          NODE          PRIO ENTRY            STACK GROUPS
% ============================================================================
  TASK TIME_TASK     NODE1            3 time_tsk          1024 [EXE]
  TASK ENTRY_POINT   NODE1            6 entry_point       1024 [EXE]
  TASK TMTC          NODE1            6 tmtc              2048 [EXE]
  TASK HS0           NODE1            7 hs0               1024 [EXE]
  TASK HK_ASK        NODE1           10 hk_ask            1024 [EXE]
  TASK LS            NODE1           10 ls                4096 [EXE]
  TASK CMD_SEQ       NODE1           10 cmd_seq           2048 [EXE]
  TASK LS_HDL        NODE1           10 ls_hdl            2048 [EXE]
  TASK DATA_HDL      NODE1           10 data_hdl          2048 [EXE]
  TASK RES_CHK       NODE1           12 res_chk           2048 [EXE]
  TASK HS_HDL        NODE1           10 hs_hdl            2048 [EXE]
  TASK HS_FLUSH      NODE1            8 hs_flush          1024 [EXE]
  TASK HS1           NODE1            9 hs1               1024 [EXE]
  TASK VM_MON        NODE1            5 vm_mon            1024 [EXE]

% FIFO NAME          NODE          DEPTH WIDTH
% ========================================
  FIFO SD_TM_QUEUE   NODE1           728    36
  FIFO HK_TM_QUEUE   NODE1            24    36
  FIFO LS_LP_QUEUE   NODE1           512    16
  FIFO EVENT_TM_QUEUE NODE1           24    36
  FIFO TC_QUEUE      NODE1             4    36
  FIFO LS_HP_QUEUE   NODE1           512    16
  FIFO LS_HDL_QUEUE  NODE1            40    40
  FIFO SD_PKT_QUEUE  NODE1            64    40
  FIFO HS_HDL_QUEUE  NODE1            20    40
  FIFO FRAME_QUEUE   NODE1            20     8
```

```
% EVENT NAME          NODE          CALL
% ============================================
  EVENT ISR_1553_EVENT NODE1       'NULL           '
  EVENT HS_EVENT      NODE1        'NULL           '
  EVENT TS_EVENT      NODE1        'NULL           '
  EVENT HK_EVENT      NODE1        'NULL           '
  EVENT MEAS_EVENT    NODE1        'NULL           '
  EVENT HS_FLUSH_EVENT NODE1       'NULL           '
  EVENT VM_REQ_EVENT NODE1         'NULL           '

% MAP NAME          NODE          BLOCKS BLOCKSIZE
% ============================================

% SEMA NAME          NODE
% ============================
  SEMA HK_REQ_SEMA  NODE1
  SEMA LS_SEMA      NODE1
  SEMA HS_HDL_WAIT_SEMA NODE1
  SEMA LS_HDL_WAIT_SEMA NODE1

% MAILBOX NAME          NODE
% ================================

% RESOURCE NAME          NODE
% ================================
  RESOURCE POOL_MONITOR NODE1

% POOL NAME          NODE          SIZE_SMALL SIZE_LARGE BLOCK_NUMBER
% ================================================================
  POOL SD_POOL       NODE1             32960      32960           32
  POOL HK_POOL       NODE1              2048       2048           24
  POOL EV_POOL       NODE1               256        256           24
  POOL TC_POOL       NODE1               512        512           32
  POOL STF_POOL      NODE1               512        512           32

***** end of file *****
```

HIFI.ach:

```
!=============================================================================
!=============================================================================
!Ric: Init parte da 4000, cstack=0x400
.system FirstDPU;
.processor = ADSP21020;
!=================== Program
.segment /pm /ram          /begin=0x004000          /end=0x005FFF   seg_init;
.segment /pm /ram          /begin=0x006000          /end=0x06FFFF   seg_pmco;
.segment /pm /ram          /begin=0x070000          /end=0x07FFFF   seg_pmda;
!.segment /pm /ram         /begin=0x010600          /end=0x07FFFF   seg_pmtest;
!
! reserved data memory area:
!                                 dm data  256 KWords 32 bits
!                                 dm stack 128 Kwords 32 bits was 64
!                                 dm heap  128 Kwords 32 bits was 192
!
.segment /pm /ram  /begin=0x00000000 /end=0x000000FF                        seg_rth;
.segment /dm /ram  /begin=0x00000000 /end=0x00027FFF                        seg_dmda;
.segment /dm /ram  /begin=0x00028000 /end=0x000283FF     /cstack            seg_stak;
.segment /dm /ram  /begin=0x00028400 /end=0x0007FFFF      /cheap            heap1;
!
!
!
.segment /dm /ram  /begin=0x80000000 /end=0x8003FFFF     EEPROM;
.segment /dm /port /begin=0x81000000 /end=0x81FFFFFF     Timer;
```

```
.segment /dm /port /begin=0x82000000 /end=0x82FFFFFF     watchdog;
.segment /dm /port /begin=0x83000000 /end=0x83FFFFFF     Int_mng;
.segment /dm /ram  /begin=0x84000000 /end=0x84FFFFFF     SMCS_reg;
.segment /dm /ram  /begin=0x88000000 /end=0x8FFFFFFF     Bus_IF;
!================================================================================
!
!Bank Description
!the PM bank1 is mot mounted
.bank /pm0 /wtstates=0 /wtmode=internal     /begin=0x000000;
!.bank /pm1 /wtstates=0 /wtmode=internal    /begin=0x800000;
!
! DM bank 0 is used for data storing
! DM bank 1 is reserved for Mezzanine IF and it is not used
! DM bank 2 is reserved for IEEE 1355
! DM bank 3 is reserved for the following register and Device
!                                     EEPROM, Interval Timer, Watchdog, Interrupt Manager
!                                     SMCS332 register, 32 bit bus interface
.bank /dm0 /wtstates=1 /wtmode=internal     /begin=0x00000000;
.bank /dm1 /wtstates=1 /wtmode=both         /begin=0x20000000;
.bank /dm2 /wtstates=0 /wtmode=internal     /begin=0x40000000;
.bank /dm3 /wtstates=1 /wtmode=both         /begin=0x80000000;
!================================================================================
.endsys;
!***** end of file *****
```

```
!=============================================================================
```

Segfile.txt:

```
seg_rth
seg_init
seg_pmco
```

```
!=============================================================================
```

Pagefile.txt:

```
1
2
3
4
5
6
7
8
9
10
```

A2.1 conf_tab.c

In this section the conf_tab.c implemented in the OBS reference version is reported.

```
// $RCSfile: conf_tab.c,v $Revision: 1.8 $Date: 2005/03/11 15:08:57
// This file contains the following arrays:
// HK_req_FCU_005 as per HK-ICD 1.11, 3rd modification
// FCU_non_per_HK_002 as per HK-ICD 1.9
// HK_req_LCU_009 as per HK ICD 1.11 for LCU-FM
// Essential_HK_001 as per HK-ICD 1.11
```

```
// IV curve FM
// LOscan as per TM-ICD 1.8
// commands to dump table
// Frequency max
// Cmd_safe_001
//-------------------------------------------------------------
#include "conf_tab.h"
#include "conf_tab.dox"

//--- variables ----------------------
//! length of the tables of this file
unsigned int Size_hk_FCU, Size_hk_LCU, Single_HK_len, LCU_vectorscan_commands_len;
unsigned int LCU_IV_param_len, single_HK_LCU_cmds_len, LCU_frequency_max_len;
unsigned int goto_safe_commands_len;

// HK_req_FCU_005 as per HK-ICD 1.11, 3rd modification
//! Sequence of LS Commands to be issued by hk_ask to gather HK FCU data
unsigned int HK_FCU_commands[] =
{
        0x8C01FFFF,    0x8C03FFFF,    0x8C04FFFF,    0x8C40FFFF,    0x8C41FFFF,
        0x8C42FFFF,    0x8C43FFFF,    0x8C44FFFF,    0x8C45FFFF,    0x8C48FFFF,
        0x8C49FFFF,    0x8C4AFFFF,    0x8C4BFFFF,    0x8C4CFFFF,    0x8C4DFFFF,
        0x8C4EFFFF,    0x8C4FFFFF,    0x8C50FFFF,    0x8C51FFFF,    0x8C60FFFF,
        0x8C61FFFF,    0x8C62FFFF,    0x8C63FFFF,    0x8C64FFFF,    0x8C65FFFF,
        0x8C66FFFF,    0x8C78FFFF,    0x8C79FFFF,    0x8C81FFFF,    0x8C83FFFF,
        0x8C84FFFF,    0x8CC0FFFF,    0x8CC1FFFF,    0x8CC2FFFF,    0x8CC3FFFF,
        0x8CC4FFFF,    0x8CC5FFFF,    0x8CC8FFFF,    0x8CC9FFFF,    0x8CCAFFFF,
        0x8CCBFFFF,    0x8CCCFFFF,    0x8CCDFFFF,    0x8CCEFFFF,    0x8CCFFFFF,
        0x8CE8FFFF,    0x8CE9FFFF,    0x8CEAFFFF,    0x8CEBFFFF,    0x8CECFFFF,
        0x8CEDFFFF,    0x8CEEFFFF,    0x8CEFFFFF,    0x8CF0FFFF,    0x8CF1FFFF,
        0x8CF2FFFF,    0x8CF3FFFF,    0x8CF4FFFF,    0x8CF5FFFF,    0x8CF6FFFF,
        0x8CF8FFFF,    0x8CF9FFFF,    0x8CFAFFFF,    0x8CFBFFFF,    0x8D01FFFF,
        0x8D03FFFF,    0x8D04FFFF,    0x8D40FFFF,    0x8D41FFFF,    0x8D42FFFF,
        0x8D43FFFF,    0x8D44FFFF,    0x8D45FFFF,    0x8D48FFFF,    0x8D49FFFF,
        0x8D4AFFFF,    0x8D4BFFFF,    0x8D4CFFFF,    0x8D4DFFFF,    0x8D4EFFFF,
        0x8D4FFFFF,    0x8D50FFFF,    0x8D51FFFF,    0x8D60FFFF,    0x8D61FFFF,
        0x8D62FFFF,    0x8D63FFFF,    0x8D64FFFF,    0x8D65FFFF,    0x8D66FFFF,
        0x8D78FFFF,    0x8D79FFFF,    0x8D81FFFF,    0x8D83FFFF,    0x8D84FFFF,
        0x8DC0FFFF,    0x8DC1FFFF,    0x8DC2FFFF,    0x8DC3FFFF,    0x8DC4FFFF,
        0x8DC5FFFF,    0x8DC8FFFF,    0x8DC9FFFF,    0x8DCAFFFF,    0x8DCBFFFF,
        0x8DCCFFFF,    0x8DCDFFFF,    0x8DCEFFFF,    0x8DCFFFFF,    0x8DE8FFFF,
        0x8DE9FFFF,    0x8DEAFFFF,    0x8DEBFFFF,    0x8DECFFFF,    0x8DEDFFFF,
        0x8DEEFFFF,    0x8DEFFFFF,    0x8DF0FFFF,    0x8DF1FFFF,    0x8DF2FFFF,
        0x8DF3FFFF,    0x8DF4FFFF,    0x8DF5FFFF,    0x8DF6FFFF,    0x8DF8FFFF,
        0x8DF9FFFF,    0x8DFAFFFF,    0x8DFBFFFF,    0x8F01FFFF,    0x8F03FFFF,
        0x8F04FFFF,    0x8F08FFFF,    0x8F09FFFF,    0x8F0AFFFF,    0x8F0BFFFF,
        0x8F0CFFFF,    0x8F40FFFF,    0x8F41FFFF,    0x8F42FFFF,    0x8F43FFFF,
        0x8F44FFFF,    0x8F45FFFF,    0x8F46FFFF,    0x8F47FFFF,    0x8F48FFFF,
        0x8F49FFFF,    0x8F4AFFFF,    0x8F4BFFFF,    0x8F4CFFFF,    0x8F4DFFFF,
        0x8F4EFFFF,    0x8F4FFFFF,    0x8F50FFFF,    0x8F51FFFF,    0x8F52FFFF,
        0x8F53FFFF,    0x8F54FFFF,    0x8F55FFFF,    0x8F56FFFF,    0x8F58FFFF,
        0x8F59FFFF,    0x8F5AFFFF,    0x8F5BFFFF,    0x8F5CFFFF,    0x8F5DFFFF,
        0x8F60FFFF,    0x8F61FFFF,    0x8F62FFFF,    0x8F63FFFF,    0x8F64FFFF,
        0x8F65FFFF,    0x8F66FFFF,    0x8F67FFFF,    0xCF050078,    0x8F78FFFF,
        0xCF050079,    0x8F79FFFF,    0xCF05007A,    0x8F7AFFFF,    0xCF05007B,
        0x8F7BFFFF,    0xCF05007C,    0x8F7CFFFF,    0xCF05007D,    0x8F7DFFFF,
        0x8F7EFFFF,    0x8F7FFFFF,    0x8F10FFFF,    0x8F11FFFF
};
```

```
// FCU_non_per_HK_003 as per HK-ICD 1.12
//! Sequence of LS Commands to be issued by single hk to gather HK data
unsigned int Single_HK_FCU_commands[] =
{
        0x8C13FFFF,    0x8CA8FFFF,    0x8CA9FFFF,    0x8CABFFFF,    0x8CACFFFF,
        0x8CAEFFFF,    0x8CAFFFFF,    0x8CB1FFFF,    0x8CB2FFFF,    0x8CB4FFFF,
        0x8CB5FFFF,    0x8D13FFFF,    0x8DA8FFFF,    0x8DA9FFFF,    0x8DABFFFF,
        0x8DACFFFF,    0x8DAEFFFF,    0x8DAFFFFF,    0x8DB1FFFF,    0x8DB2FFFF,
        0x8DB4FFFF,    0x8DB5FFFF,    0x8F11FFFF,    0x8F13FFFF,    0x8F14FFFF,
        0x8F15FFFF,    0x8F16FFFF,    0x8F17FFFF,    0x8F18FFFF,    0x8F19FFFF,
        0x8F26FFFF,    0x8C20FFFF,    0x8C22FFFF,    0x8D20FFFF,    0x8D22FFFF,
        0x8F10FFFF,    0x8C10FFFF,    0x8D10FFFF,    0x8C24FFFF,    0x8D24FFFF,
        0x8CFCFFFF,    0x8CFDFFFF,    0x8DFCFFFF,    0x8DFDFFFF
};



// HK_req_LCU_009 as per HK ICD 1.11 for LCU-FM
//! Sequence of LS Commands to be issued by hk_ask to gather HK LCU data
unsigned int HK_LCU_commands[] =
{
        0xB200FFFF,    0xB031FFFF,    0xB032FFFF,    0xB033FFFF,    0xB034FFFF,
        0xB035FFFF,    0xB036FFFF,    0xB037FFFF,    0xB038FFFF,    0xB039FFFF,
        0xB03AFFFF,    0xB03BFFFF,    0xB03CFFFF,    0xB03DFFFF,    0xB03EFFFF,
        0xB041FFFF,    0xB042FFFF,    0xB043FFFF,    0xB044FFFF,    0xB045FFFF,
        0xB046FFFF,    0xB047FFFF,    0xB051FFFF,    0xB052FFFF,    0xB053FFFF,
        0xB054FFFF,    0xB055FFFF,    0xB056FFFF,    0xB057FFFF,    0xB080FFFF,
        0xB081FFFF,    0xB082FFFF,    0xB083FFFF,    0xB084FFFF,    0xB085FFFF,
        0xB086FFFF,    0xB089FFFF,    0xB08AFFFF,    0xB08BFFFF,    0xB08CFFFF,
        0xB08DFFFF,    0xB08EFFFF,    0xB090FFFF,    0xB091FFFF,    0xB092FFFF,
        0xB093FFFF,    0xB0A1FFFF,    0xB0A2FFFF,    0xB0A3FFFF,    0xB0A4FFFF,
        0xB0B0FFFF,    0xB0B1FFFF,    0xB0B2FFFF,    0xB0B3FFFF,    0xB0B4FFFF,
        0xB0B5FFFF,    0xB20AFFFF,    0xB208FFFF,    0xB209FFFF,    0xB201FFFF,
        0xB20BFFFF,    0xB20CFFFF,    0xB204FFFF,    0xB20EFFFF,    0xF30ACC7A,
        0xB33AFFFF,    0xF30ACC7A,    0xB33CFFFF,    0xF30ACC7A,    0xB33EFFFF
};

// Essential_HK_001 as per HK-ICD 1.11
//! Vector of the positions (byte) of the HK to be copied on the essential HK. Always two bytes are copied.
int essential_hk_positions[HK_ESS_PARAMS_NR] =
{
        64,             68,             72,             76,             80,             84,
        108,    112,    120,    156,
        218,    220,    254,    256,    278,    280,    284,    346,    348,    382,
        384,    406,    408,    412,    414,    416,    418,    420,    448,    450,
        452,    454,    456,    482,    536,    538,    540,    552,    576,    654,
        656,    782,    784,    872,    964,    966,    968,    970,    978
};

// IV curve FM
// Sequence of HK request to collect IV-data
unsigned int LCU_IV_param[] =
{
        0xF30ACC7A,    0xB330FFFF,    0xB328FFFF,    0xB32AFFFF,    0xB32CFFFF,
        0xB32EFFFF,    0xB320FFFF,    0xB322FFFF,    0xB324FFFF,    0xB326FFFF
};
```

```
// LOscan as per TM-ICD 1.8
//! Sequence of LS Commands to be issued by LOscan to gather LCU data
unsigned int LCU_vectorscan_commands[] =
{
        0xB041FFFF,     0xB042FFFF,     0xB043FFFF,     0xB044FFFF,     0xB045FFFF,
        0xB046FFFF,     0xB047FFFF,     0xB051FFFF,     0xB052FFFF,     0xB053FFFF,
        0xB054FFFF,     0xB055FFFF,     0xB056FFFF,     0xB057FFFF,     0xB080FFFF,
        0x8C61FFFF,     0x8D61FFFF
};

// commands to dump table
//! Sequence of LS Commands to be issued by single hk to gather HK data
unsigned int Single_HK_LCU_commands[] =
{
        0xF10F2081,     0xF10F0001,     0xF10F2001,     0xF10F2091,     0xF10F4091,     0xF10F0011,
        0xF10F4011,     0xF10F2011,     0xF10F6011,     0xF10F6091,     0xF10F20A1,
        0xF10F40A1,     0xF10F0021,     0xF10F4021,     0xF10F2021,     0xF10F6021,     0xF10F60A1,
        0x0FFFFFFF,     0x0FFFFFFF,     0x0FFFFFFF,     0x0FFFFFFF,     0x0FFFFFFF,
        0x0FFFFFFF,     0x0FFFFFFF,     0xF10F20C1,     0xF10F40C1,     0xF10F0041,     0xF10F4041,
        0xF10F2041,     0xF10F60C1,     0xF10F6041,     0xF10F20D1,     0xF10F40D1,
        0xF10F0051,     0xF10F4051,     0xF10F2051,     0xF10F60D1,     0xF10F6051,     0xF10F20E1,
        0xF10F40E1,     0xF10F0061,     0xF10F4061,     0xF10F2061,     0xF10F60E1,
        0xF10F6061,     0xF10F20F1,     0xF10F40F1,     0xF10F0071,     0xF10F4071,     0xF10F2071,
        0xF10F60F1,     0xF10F6071,     0xF10F8081,     0xF10F8001,     0xF10FA001,
        0xF10F2082,     0xF10F0002,     0xF10F2002,     0xF10F2092,     0xF10F4092,     0xF10F0012,
        0xF10F4012,     0xF10F2012,     0xF10F6012,     0xF10F6092,     0xF10F20A2,
        0xF10F40A2,     0xF10F0022,     0xF10F4022,     0xF10F2022,     0xF10F6022,     0xF10F60A2,
        0x0FFFFFFF,     0x0FFFFFFF,     0x0FFFFFFF,     0x0FFFFFFF,     0x0FFFFFFF,
        0x0FFFFFFF,     0x0FFFFFFF,     0xF10F20C2,     0xF10F40C2,     0xF10F0042,     0xF10F4042,
        0xF10F2042,     0xF10F60C2,     0xF10F6042,     0xF10F20D2,     0xF10F40D2,
        0xF10F0052,     0xF10F4052,     0xF10F2052,     0xF10F60D2,     0xF10F6052,     0xF10F20E2,
        0xF10F40E2,     0xF10F0062,     0xF10F4062,     0xF10F2062,     0xF10F60E2,
        0xF10F6062,     0xF10F20F2,     0xF10F40F2,     0xF10F0072,     0xF10F4072,     0xF10F2072,
        0xF10F60F2,     0xF10F6072,     0xF10F8082,     0xF10F8002,     0xF10FA002,
        0xF10F2083,     0xF10F0003,     0xF10F2003,     0xF10F2093,     0xF10F4093,     0xF10F0013,
        0xF10F4013,     0xF10F2013,     0xF10F6013,     0xF10F6093,     0xF10F20A3,
        0xF10F40A3,     0xF10F0023,     0xF10F4023,     0xF10F2023,     0xF10F6023,     0xF10F60A3,
        0xF10F20B3,     0xF10F40B3,     0xF10F0033,     0xF10F4033,     0xF10F2033,
        0xF10F6033,     0xF10F60B3,     0xF10F20C3,     0xF10F40C3,     0xF10F0043,     0xF10F4043,
        0xF10F2043,     0xF10F60C3,     0xF10F6043,     0xF10F20D3,     0xF10F40D3,
        0xF10F0053,     0xF10F4053,     0xF10F2053,     0xF10F60D3,     0xF10F6053,     0xF10F20E3,
        0xF10F40E3,     0xF10F0063,     0xF10F4063,     0xF10F2063,     0xF10F60E3,
        0xF10F6063,     0xF10F20F3,     0xF10F40F3,     0xF10F0073,     0xF10F4073,     0xF10F2073,
        0xF10F60F3,     0xF10F6073,     0xF10F8083,     0xF10F8003,     0xF10FA003,
        0xF10F2084,     0xF10F0004,     0xF10F2004,     0xF10F2094,     0xF10F4094,     0xF10F0014,
        0xF10F4014,     0xF10F2014,     0xF10F6014,     0xF10F6094,     0xF10F20A4,
        0xF10F40A4,     0xF10F0024,     0xF10F4024,     0xF10F2024,     0xF10F6024,     0xF10F60A4,
        0xF10F20B4,     0xF10F40B4,     0xF10F0034,     0xF10F4034,     0xF10F2034,
        0xF10F6034,     0xF10F60B4,     0xF10F20C4,     0xF10F40C4,     0xF10F0044,     0xF10F4044,
        0xF10F2044,     0xF10F60C4,     0xF10F6044,     0xF10F20D4,     0xF10F40D4,
        0xF10F0054,     0xF10F4054,     0xF10F2054,     0xF10F60D4,     0xF10F6054,     0xF10F20E4,
        0xF10F40E4,     0xF10F0064,     0xF10F4064,     0xF10F2064,     0xF10F60E4,
        0xF10F6064,     0xF10F20F4,     0xF10F40F4,     0xF10F0074,     0xF10F4074,     0xF10F2074,
        0xF10F60F4,     0xF10F6074,     0xF10F8084,     0xF10F8004,     0xF10FA004,
        0xF10F2085,     0xF10F0005,     0xF10F2005,     0xF10F2095,     0xF10F4095,     0xF10F0015,
        0xF10F4015,     0xF10F2015,     0xF10F6015,     0xF10F6095,     0xF10F20A5,
        0xF10F40A5,     0xF10F0025,     0xF10F4025,     0xF10F2025,     0xF10F6025,     0xF10F60A5,
        0xF10F20B5,     0xF10F40B5,     0xF10F0035,     0xF10F4035,     0xF10F2035,
```

0xF10F6035, 0xF10F60B5, 0xF10F20C5, 0xF10F40C5, 0xF10F0045, 0xF10F4045,
0xF10F2045, 0xF10F60C5, 0xF10F6045, 0xF10F20D5, 0xF10F40D5,
0xF10F0055, 0xF10F4055, 0xF10F2055, 0xF10F60D5, 0xF10F6055, 0xF10F20E5,
0xF10F40E5, 0xF10F0065, 0xF10F4065, 0xF10F2065, 0xF10F60E5,
0xF10F6065, 0xF10F20F5, 0xF10F40F5, 0xF10F0075, 0xF10F4075, 0xF10F2075,
0xF10F60F5, 0xF10F6075, 0xF10F8085, 0xF10F8005, 0xF10FA005,
0xF10F2086, 0xF10F0006, 0xF10F2006, 0xF10F2096, 0xF10F4096, 0xF10F0016,
0xF10F4016, 0xF10F2016, 0xF10F6016, 0xF10F6096, 0xF10F20A6,
0xF10F40A6, 0xF10F0026, 0xF10F4026, 0xF10F2026, 0xF10F6026, 0xF10F60A6,
0xF10F20B6, 0xF10F40B6, 0xF10F0036, 0xF10F4036, 0xF10F2036,
0xF10F6036, 0xF10F60B6, 0xF10F20C6, 0xF10F40C6, 0xF10F0046, 0xF10F4046,
0xF10F2046, 0xF10F60C6, 0xF10F6046, 0xF10F20D6, 0xF10F40D6,
0xF10F0056, 0xF10F4056, 0xF10F2056, 0xF10F60D6, 0xF10F6056, 0xF10F20E6,
0xF10F40E6, 0xF10F0066, 0xF10F4066, 0xF10F2066, 0xF10F60E6,
0xF10F6066, 0xF10F20F6, 0xF10F40F6, 0xF10F0076, 0xF10F4076, 0xF10F2076,
0xF10F60F6, 0xF10F6076, 0xF10F8086, 0xF10F8006, 0xF10FA006,
0xF10F2087, 0xF10F0007, 0xF10F2007, 0xF10F2097, 0xF10F4097, 0xF10F0017,
0xF10F4017, 0xF10F2017, 0xF10F6017, 0xF10F6097, 0xF10F20A7,
0xF10F40A7, 0xF10F0027, 0xF10F4027, 0xF10F2027, 0xF10F6027, 0xF10F60A7,
0x0FFFFFFF, 0x0FFFFFFF, 0x0FFFFFFF, 0x0FFFFFFF, 0x0FFFFFFF,
0x0FFFFFFF, 0x0FFFFFFF, 0xF10F20C7, 0xF10F40C7, 0xF10F0047, 0xF10F4047,
0xF10F2047, 0xF10F60C7, 0xF10F6047, 0xF10F20D7, 0xF10F40D7,
0xF10F0057, 0xF10F4057, 0xF10F2057, 0xF10F60D7, 0xF10F6057, 0xF10F20E7,
0xF10F40E7, 0xF10F0067, 0xF10F4067, 0xF10F2067, 0xF10F60E7,
0xF10F6067, 0xF10F20F7, 0xF10F40F7, 0xF10F0077, 0xF10F4077, 0xF10F2077,
0xF10F60F7, 0xF10F6077, 0xF10F8087, 0xF10F8007, 0xF10FA007,
0xF10F2088, 0xF10F0008, 0xF10F2008, 0xF10F2098, 0xF10F4098, 0xF10F0018,
0xF10F4018, 0xF10F2018, 0xF10F6018, 0xF10F6098, 0xF10F20A8,
0xF10F40A8, 0xF10F0028, 0xF10F4028, 0xF10F2028, 0xF10F6028, 0xF10F60A8,
0x0FFFFFFF, 0x0FFFFFFF, 0x0FFFFFFF, 0x0FFFFFFF, 0x0FFFFFFF,
0x0FFFFFFF, 0x0FFFFFFF, 0xF10F20C8, 0xF10F40C8, 0xF10F0048, 0xF10F4048,
0xF10F2048, 0xF10F60C8, 0xF10F6048, 0xF10F20D8, 0xF10F40D8,
0xF10F0058, 0xF10F4058, 0xF10F2058, 0xF10F60D8, 0xF10F6058, 0xF10F20E8,
0xF10F40E8, 0xF10F0068, 0xF10F4068, 0xF10F2068, 0xF10F60E8,
0xF10F6068, 0xF10F20F8, 0xF10F40F8, 0xF10F0078, 0xF10F4078, 0xF10F2078,
0xF10F60F8, 0xF10F6078, 0xF10F8088, 0xF10F8008, 0xF10FA008,
0xF10F2089, 0xF10F0009, 0xF10F2009, 0xF10F2099, 0xF10F4099, 0xF10F0019,
0xF10F4019, 0xF10F2019, 0xF10F6019, 0xF10F6099, 0xF10F20A9,
0xF10F40A9, 0xF10F0029, 0xF10F4029, 0xF10F2029, 0xF10F6029, 0xF10F60A9,
0x0FFFFFFF, 0x0FFFFFFF, 0x0FFFFFFF, 0x0FFFFFFF, 0x0FFFFFFF,
0x0FFFFFFF, 0x0FFFFFFF, 0xF10F20C9, 0xF10F40C9, 0xF10F0049, 0xF10F4049,
0xF10F2049, 0xF10F60C9, 0xF10F6049, 0xF10F20D9, 0xF10F40D9,
0xF10F0059, 0xF10F4059, 0xF10F2059, 0xF10F60D9, 0xF10F6059, 0xF10F20E9,
0xF10F40E9, 0xF10F0069, 0xF10F4069, 0xF10F2069, 0xF10F60E9,
0xF10F6069, 0xF10F20F9, 0xF10F40F9, 0xF10F0079, 0xF10F4079, 0xF10F2079,
0xF10F60F9, 0xF10F6079, 0xF10F8089, 0xF10F8009, 0xF10FA009,
0xF10F208A, 0xF10F000A, 0xF10F200A, 0xF10F209A, 0xF10F409A, 0xF10F001A,
0xF10F401A, 0xF10F201A, 0xF10F601A, 0xF10F609A, 0xF10F20AA,
0xF10F40AA, 0xF10F002A, 0xF10F402A, 0xF10F202A, 0xF10F602A, 0xF10F60AA,
0x0FFFFFFF, 0x0FFFFFFF, 0x0FFFFFFF, 0x0FFFFFFF, 0x0FFFFFFF,
0x0FFFFFFF, 0x0FFFFFFF, 0xF10F20CA, 0xF10F40CA, 0xF10F004A, 0xF10F404A,
0xF10F204A, 0xF10F60CA, 0xF10F604A, 0xF10F20DA, 0xF10F40DA,
0xF10F005A, 0xF10F405A, 0xF10F205A, 0xF10F60DA, 0xF10F605A, 0xF10F20EA,
0xF10F40EA, 0xF10F006A, 0xF10F406A, 0xF10F206A, 0xF10F60EA,
0xF10F606A, 0xF10F20FA, 0xF10F40FA, 0xF10F007A, 0xF10F407A, 0xF10F207A,
0xF10F60FA, 0xF10F607A, 0xF10F808A, 0xF10F800A, 0xF10FA00A,
0xF10F208B, 0xF10F000B, 0xF10F200B, 0xF10F209B, 0xF10F409B, 0xF10F001B,
0xF10F401B, 0xF10F201B, 0xF10F601B, 0xF10F609B, 0xF10F20AB,

```
        0xF10F40AB,  0xF10F002B,  0xF10F402B,  0xF10F202B,  0xF10F602B,  0xF10F60AB,
        0xF10F20BB,  0xF10F40BB,  0xF10F003B,  0xF10F403B,  0xF10F203B,
        0xF10F603B,  0xF10F60BB,  0xF10F20CB,  0xF10F40CB,  0xF10F004B,  0xF10F404B,
        0xF10F204B,  0xF10F60CB,  0xF10F604B,  0xF10F20DB,  0xF10F40DB,
        0xF10F005B,  0xF10F405B,  0xF10F205B,  0xF10F60DB,  0xF10F605B,  0xF10F20EB,
        0xF10F40EB,  0xF10F006B,  0xF10F406B,  0xF10F206B,  0xF10F60EB,
        0xF10F606B,  0xF10F20FB,  0xF10F40FB,  0xF10F007B,  0xF10F407B,  0xF10F207B,
        0xF10F60FB,  0xF10F607B,  0xF10F808B,  0xF10F800B,  0xF10FA00B,
        0xF10F208C,  0xF10F000C,  0xF10F200C,  0xF10F209C,  0xF10F409C,  0xF10F001C,
        0xF10F401C,  0xF10F201C,  0xF10F601C,  0xF10F609C,  0xF10F20AC,
        0xF10F40AC,  0xF10F002C,  0xF10F402C,  0xF10F202C,  0xF10F602C,  0xF10F60AC,
        0x0FFFFFFF,  0x0FFFFFFF,  0x0FFFFFFF,  0x0FFFFFFF,  0x0FFFFFFF,
        0x0FFFFFFF,  0x0FFFFFFF,  0xF10F20CC,  0xF10F40CC,  0xF10F004C,  0xF10F404C,
        0xF10F204C,  0xF10F60CC,  0xF10F604C,  0xF10F20DC,  0xF10F40DC,
        0xF10F005C,  0xF10F405C,  0xF10F205C,  0xF10F60DC,  0xF10F605C,  0xF10F20EC,
        0xF10F40EC,  0xF10F006C,  0xF10F406C,  0xF10F206C,  0xF10F60EC,
        0xF10F606C,  0xF10F20FC,  0xF10F40FC,  0xF10F007C,  0xF10F407C,  0xF10F207C,
        0xF10F60FC,  0xF10F607C,  0xF10F808C,  0xF10F800C,  0xF10FA00C,
        0xF10F208D,  0xF10F000D,  0xF10F200D,  0xF10F209D,  0xF10F409D,  0xF10F001D,
        0xF10F401D,  0xF10F201D,  0xF10F601D,  0xF10F609D,  0xF10F20AD,
        0xF10F40AD,  0xF10F002D,  0xF10F402D,  0xF10F202D,  0xF10F602D,  0xF10F60AD,
        0x0FFFFFFF,  0x0FFFFFFF,  0x0FFFFFFF,  0x0FFFFFFF,  0x0FFFFFFF,
        0x0FFFFFFF,  0x0FFFFFFF,  0xF10F20CD,  0xF10F40CD,  0xF10F004D,  0xF10F404D,
        0xF10F204D,  0xF10F60CD,  0xF10F604D,  0xF10F20DD,  0xF10F40DD,
        0xF10F005D,  0xF10F405D,  0xF10F205D,  0xF10F60DD,  0xF10F605D,  0xF10F20ED,
        0xF10F40ED,  0xF10F006D,  0xF10F406D,  0xF10F206D,  0xF10F60ED,
        0xF10F606D,  0xF10F20FD,  0xF10F40FD,  0xF10F007D,  0xF10F407D,  0xF10F207D,
        0xF10F60FD,  0xF10F607D,  0xF10F808D,  0xF10F800D,  0xF10FA00D,
        0xF10F208E,  0xF10F000E,  0xF10F200E,  0xF10F209E,  0xF10F409E,  0xF10F001E,
        0xF10F401E,  0xF10F201E,  0xF10F601E,  0xF10F609E,  0xF10F20AE,
        0xF10F40AE,  0xF10F002E,  0xF10F402E,  0xF10F202E,  0xF10F602E,  0xF10F60AE,
        0x0FFFFFFF,  0x0FFFFFFF,  0x0FFFFFFF,  0x0FFFFFFF,  0x0FFFFFFF,
        0x0FFFFFFF,  0x0FFFFFFF,  0xF10F20CE,  0xF10F40CE,  0xF10F004E,  0xF10F404E,
        0xF10F204E,  0xF10F60CE,  0xF10F604E,  0xF10F20DE,  0xF10F40DE,
        0xF10F005E,  0xF10F405E,  0xF10F205E,  0xF10F60DE,  0xF10F605E,  0xF10F20EE,
        0xF10F40EE,  0xF10F006E,  0xF10F406E,  0xF10F206E,  0xF10F60EE,
        0xF10F606E,  0xF10F20FE,  0xF10F40FE,  0xF10F007E,  0xF10F407E,  0xF10F207E,
        0xF10F60FE,  0xF10F607E,  0xF10F808E,  0xF10F800E,  0xF10FA00E
};

//Frequency max
//! Indicates the maximum applicable frequency index
unsigned int LCU_frequency_max[] =
{
        0,      0,      0,      0,      0,      0,      0,      0,      31,     31,
        0,      0,      0,      0,      0,      31,     31,     0,      0,      0,
        0,      0,      31,     31,     0,      0,      0,      0,      0,      31,
        31,     0,      0,      0,      0,      0,      31,     31,     0,      0,
        0,      0,      0,      31,     31,     0,      30,     0,      0,      0,
        31,     31,     30,     30,     30
};

// Cmd_safe_001
//! Sequence of LS command to issue in a goto_safe TC
unsigned int goto_safe_commands[] =
{
        0xF0020202, /* HL_standby */
        0xCF010000, /* HF_band_0 */
```

```
        0xFF900000, /* HR_stop */
        0xE4000011, /* HWH_laser_1_off */
        0xE4000012, /* HWH_laser_2_off */
        0xE8000011, /* HWV_laser_1_off */
        0xE8000012, /* HWV_laser_2_off */
        0xFC000005  /* WBS_stop */
};


//---------- init_conf_tables_len ---------------------------------------------
//! stores the length of the tables of this file into appropriate global variables
void init_conf_tables_len (void)
{
        Size_hk_FCU = sizeof(HK_FCU_commands);
        Size_hk_LCU = sizeof(HK_LCU_commands);
        Single_HK_len = sizeof(Single_HK_FCU_commands);
        LCU_vectorscan_commands_len = sizeof(LCU_vectorscan_commands);
        LCU_IV_param_len = sizeof(LCU_IV_param);
        single_HK_LCU_cmds_len = sizeof(Single_HK_LCU_commands);
        LCU_frequency_max_len = sizeof(LCU_frequency_max);
        goto_safe_commands_len = sizeof(goto_safe_commands);
}
```

A3. Appendix - Applicable Tables

A3.1 ICU Housekeeping

The detailed structure of the HK packet is described in AD5, section 3.1. and reported below:

Start Byte	Start bit	Length	Monitor Parameter Description	Acceptance criterium
26	0	24	HI_SW_Version Version number of the OBS	= to the version of the OBS under test
29	0	8	HI_SW_Revision Revision number of the OBS	= to the revision of the OBS under test
30	0	32	HI_IDLE Number of loops in a second performed by the res_chk task (the lowest priority task).	(>=1200*HK acquisition rate(sec) at startup)
34	0	32	HI_CPU_Load_Min Minimum delay (in one sec) in msec from one loop and the next one in res_chk task. Expressed in units of 20nsec.	
38	0	32	HI_CPU_Load_AV = Average percent CPU load.	<10 at startup
42	0	32	HI_CPU_Load_Max Maximum delay (in one sec) in msec from one loop and the next one in res_chk task. Expressed in units of 20nsec.	
46	0	32	HI_EV_POOL Max # of blocks taken in Event Pool	<28 (= 0 at startup)
50	0	32	HI_HK_POOL Max # of blocks taken in HK Pool	<22 (= 4 at startup)
54	0	32	HI_SD_POOL Max # of blocks taken in Science Pool	<30 (= 8 at startup)
58	0	32	HI_TC_POOL Max # of blocks taken in TC Pool	<6 (= 0 at startup)
62	0	32	HI_LS_QUEUE_MAX Max depth reached in LS Queue (Virtuoso FIFO)	<512 (0x200)
66	0	32	HI_HK_QUEUE_MAX Max depth reached in HK Queue (Virtuoso FIFO)	<24 (0x18)
70	0	32	HI_SD_QUEUE_MAX Max depth reached in Science Queue (Virtuoso FIFO)	<728 (0x2d8)
74	0	32	HI_EV_QUEUE_MAX Max depth reached in Event Queue (Virtuoso FIFO)	<24 (0x18)
78	0	32	HI_TC_QUEUE_MAX Max depth reached in TC Queue (Virtuoso FIFO)	<4 (0x04)
82	0	32	HI_SDPKT_QUEUE_MAX Max depth reached in Error Queue (Virtuoso FIFO)	<64 (0x40)
86	0	32	HI_VM_RUNNING_S	True if VM is running. False if it is stopped . (= 0 at startup).
90	0	32	HI_2P5_V 2.5 Volt actual value	N/A to AVM1 For FM see A1.2
94	0	32	HI_5P_V 5 Volt actual value	N/A to AVM1 For FM see A1.2
98	0	32	HI_15P_V 15 Volt actual value	N/A to AVM1 For FM see A1.2
102	0	32	HI_15M_V minus 15 Volt actual value	N/A to AVM1 For FM see A1.2
106	0	32	HI_CPU_T CPU Temperature	N/A to AVM1 For FM see A1.2
110	0	32	HI_SUBSYSTEM_S Current Subsystem Status	Equal to the commanded Subsystem Status Word = 0 at startup
110	2	1	HI_FCU_S FCU- subsystem status	
110	3	1	HI_LCU_S LCU- subsystem status	
110	4	1	HI_WBSV_S WBS-H status	
110	5	1	HI_WBSH_S WBS-V status	
110	6	1	HI_HRSV_S HRS-H status	
110	7	1	HI_HRSH_S HRS-V status	
114	0	32	HI_HP_QUEUE_MAX Max depth reached in LS cmd queue (Virtuoso FIFO)	<512 (0x200)
118	0	32	HI_Spectr_HK_valid Spectrometer Housekeeping validity flags	1 = spectr. data in HK 0 = spectr. data NOT in HK (= 0 at startup)
118	4	1	HI_WBSV_HK_S HK validity	
118	5	1	HI_WBSH_HK_S HK validity	
118	6	1	HI_HRSV_HK_S HK validity	
118	7	1	HI_HRSH_HK_S HK validity	
122	0	32	AID_spectroscopy	AID of the presently running activity

				(= 0 at startup)
126	0	32	HRS transfer counters	
126	0	16	HRS V transfer counter	
128	0	16	HRS H transfer counter	
130	0	32	WBS transfer counters	
130	0	16	WBS V transfer counter	
132	0	16	WBS H transfer counter	
134	0	32	Limit checking flags	See section 7.5.3
138	0	16	HK acquisition enabling flag	
140	0	16	HI_LCU_commands	
142	0	16	HICU_HK_29 MSW – dummy array checksum	this is the checksum of the memory area covered by a dummy array that was inserted in OBS 4.1.1 to prevent the Time stamps fifos from memory corruptions. The checksum shall not change with time. its reference value is 0xfbbc
144	0	16	HICU_HK_29 LSW – S/C I/F interrupts number	the total number of 1553 interrupts received in one second: a default value=64 in case no errors have been detected, and the first detected wrong number in case of error detection (total number of received different from 64) during the HK acquisition period
146	0	16	HICU_HK_30 MSW	a flag proportional to the pending interrupts: its default value shall be 1, if it is >=2, one or more interrupts have been lost
148	0	16	HICU_HK_30 – LSW Spare	spare
150	0	32	HICU_HK_31	Spare

A3.2 ICU Hardware parameters details

VOL_25P 2.5 V reference voltage. The allowed variability is 2.375V-2.625V

VOL_5P the output of the 5 V analogical channel. The allowed variability is 4.75V – 5.25V

VOL_15P the output of the +15 V analogical channel. The allowed variability is 14.25 – 15.75

VOL_15N the output of the -15 V analogical channel. The allowed variability is –15.75V - -14.25V

ICU TEMPERATURE: -40°C - +70°C

Table 31 Data memory map – see req.

Range	Bank	Peripheral
0x00000000-0x0007FFFF	0	Data Memory (512KWord 32-bit wide)
0x20000000-0x3FFFFFFF	1	IF Mezzanine (not used)
0x40000000-0x400003FF	2	IEEE1355 interface (1K X32-bit wide)
0x80000000-0x8003FFFF	3	EEPROM (256Kx32-bit wide)
0x81000000-0x81FFFFFF	3	Interval timer
0x82000000-0x82FFFFFF	3	Watchdog

0x83000000-0x83FFFFFF	3	Interrupt manager
0x84000000-0x84FFFFFF	3	SMCS332 configuration registers
0x88000000-0x8FFFFFFF	3	32-bit Bus Interface (1553)

Table 32 Payloas Interface Board Status register

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
ACQ3	FF3	HF3	EF3	ACQ2	FF2	HF2	EF2	ACQ1	FF1	HF1	EF1	ACQ0	FF0	HF0	EF0

D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	1553 \INC MD	Ferr3	Ferr2	Ferr1	Ferr0

A3.3 Onboard Configuration Tables

In addition to the tables provided by SRON in the file OBS tables008.xls, hereafter are reported the other tables used onboard for the subsystems related activities.

Table 33 FCU_config table

CMD Address
0xCF010000
0xCC130000
0xCCA80000
0xCCA90000
0xCCAB0000
0xCCAC0000
0xCCAE0000
0xCCAF0000
0xCCB10000
0xCCB20000
0xCCB40000
0xCCB50000
0xCD130000
0xCDA80000
0xCDA90000
0xCDAB0000
0xCDAC0000
0xCDAE0000
0xCDAF0000
0xCDB10000
0xCDB20000
0xCDB40000
0xCDB50000
0xCF110000
0xCF130000
0xCF140000
0xCF150000
0xCF160000
0xCF170000
0xCF180000

0xCF190000
0xCF260000
0xCC200000
0xCC220000
0xCD200000
0xCD220000
0xCF100000
0xCC100000
0xCD100000

Table 34 **HRS_H_att table**

CMD Address
0xD5000000
0xD5800000
0xD5900000
0xD5A00000
0xD5B00000
0xD5C00000
0xD5D00000
0xD5E00000
0xD5F00000
0xD5100030
0xD5200030
0xD5300030
0xD5400030
0xD5500030
0xD5600030
0xD6F00030

Table 35 **HRS_V_att table**

CMD Address
0xD9000000
0xD9800000
0xD9900000
0xD9A00000
0xD9B00000
0xD9C00000
0xD9D00000
0xD9E00000
0xD9F00000
0xD9100030
0xD9200030
0xD9300030
0xD9400030
0xD9500030
0xD9600030
0xD6F00030

Table 36 **HRS_V_Block_config table**

CMD Address
0xD6000000
0xD6100000

0xD6200000
0xD6300000
0xD6400000
0xD6500000
0xD6600000
0xD6700000

Table 37 **HRS_H_Block_config table**

CMD Address
0xDA000000
0xDA100000
0xDA200000
0xDA300000
0xDA400000
0xDA500000
0xDA600000
0xDA700000

Table 38 **WBS_V_config table**

CMD Address
0xE8000000
0xE8000000
0xE8000000
0xE8000000
0xE8000000

Table 39 **WBS_H_config table**

CMD Address
0xE4000000
0xE4000000
<0xE4000000
0xE4000000
0xE4000000

A4. Spectroscopy Measurements Flow Diagrams.

A4.1 Total Power

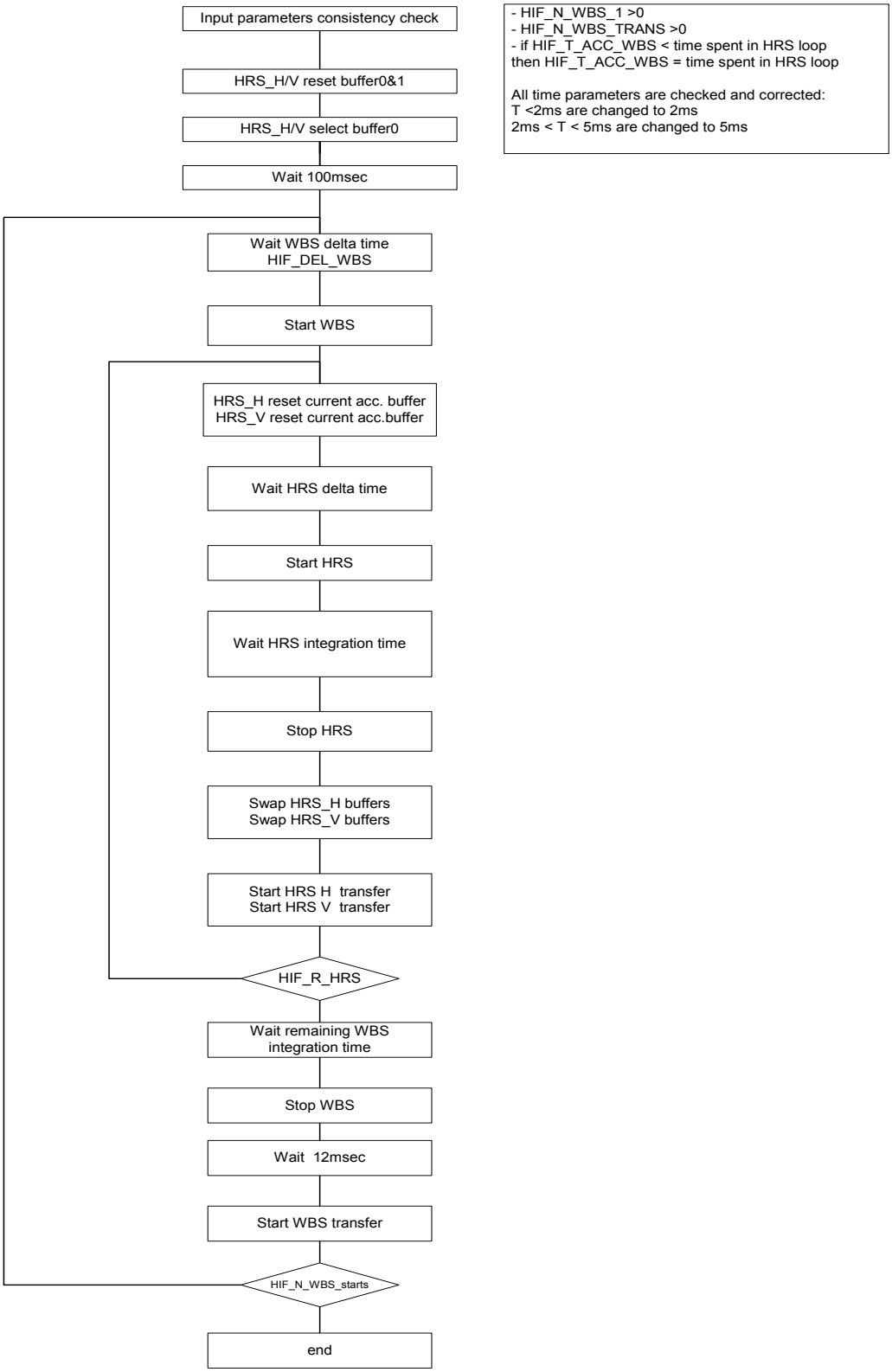

Figure 1 Total Power spectroscopy commanding flow diagram

A4.2 Slow Chop

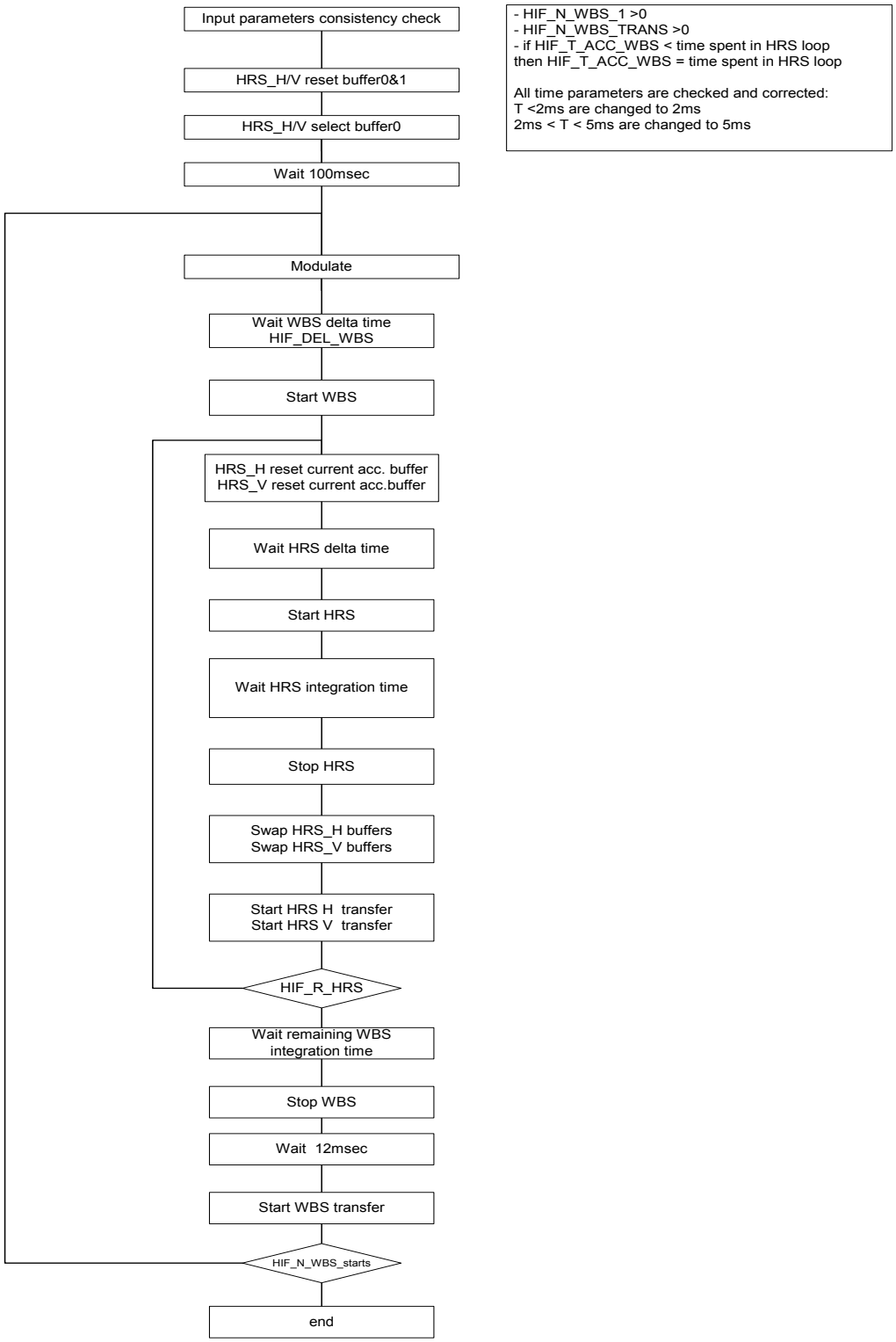

Figure 2 Slow Chop spectroscopy commanding flow diagram

A4.3 Fast Chop

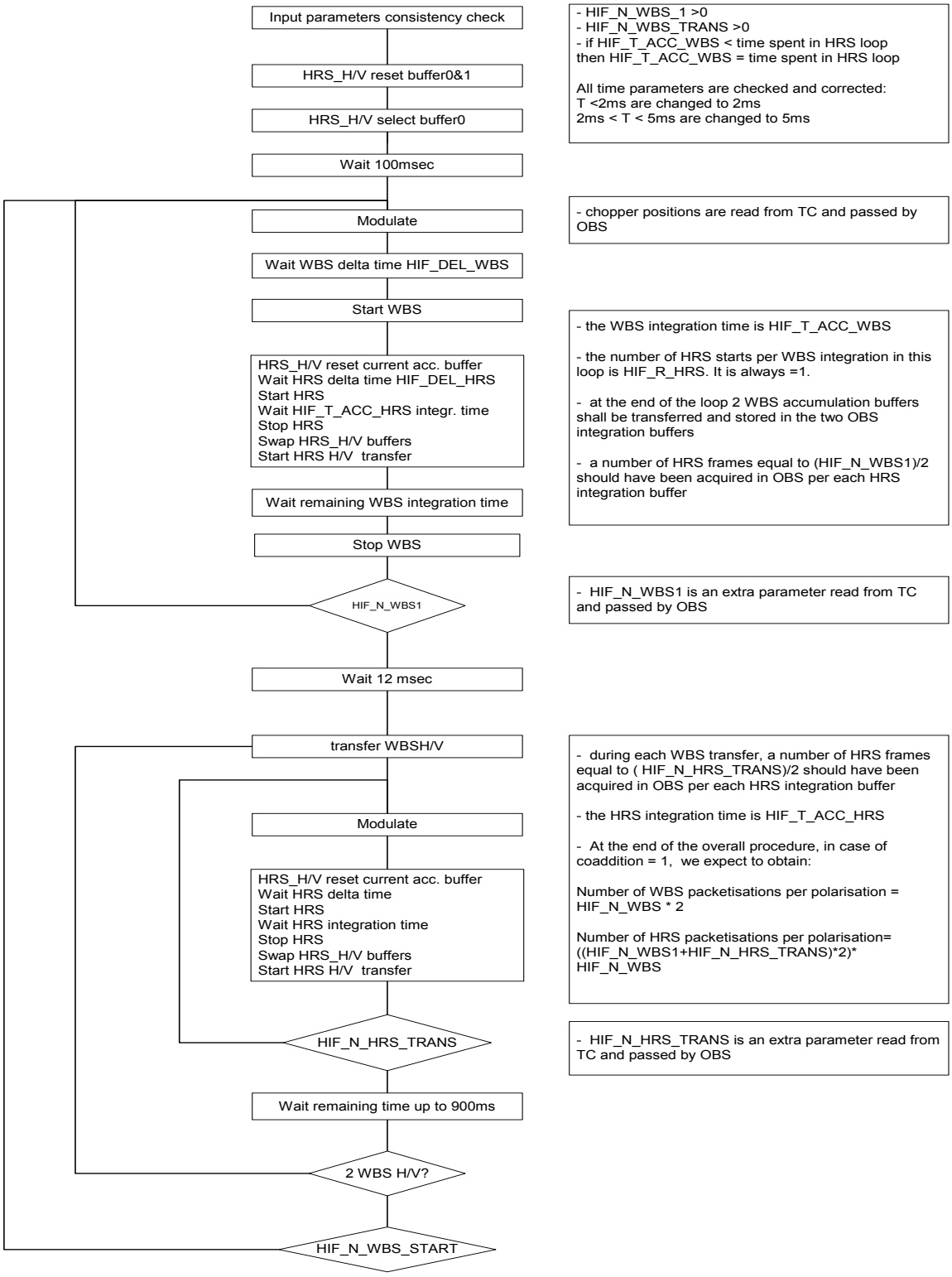

Figure 3 Fast Chop spectroscopy commanding flow diagram

A5. Appendix – OnBoard used SIDs

	Description	APID	Type	Subtype	FID	AID	SID	EventID	hex SID
	Service type 1: TC verification								
TM	TC acceptance succes	1024	1	1					
TM	TC acceptance failure	1024	1	2					
TM	TC execution completed	1024	1	7					
TM	TC execution completed - failure	1024	1	8					
	Service type 3: Housekeeping								
TM	HIFI_essential_HK	1024	3	25			1		1
TM	HIFI_Periodic_HK	1026	3	25			1027		403
TM	Non-periodic FCU	1026	3	25			17		11
TM	Non-periodic LCU	1026	3	25			18		12
TM	HIFI_LCU_macro_buffers_hk	1026	3	25			20		14
TM	HIFI_LCU_macro_tuning_hk	1026	3	25			19		13
TM	HIFI_HRS_H_IF_POWER_phase1	1026	3	25			25		19
TM	HIFI_HRS_H_IF_POWER_phase2	1026	3	25			26		1A
TM	HIFI_HRS_V_IF_POWER_phase1	1026	3	25			27		1B
TM	HIFI_HRS_V_IF_POWER_phase2	1026	3	25			28		1C
TM	HIFI_WBS_H_IF_POWER_phase1	1026	3	25			29		1D
TM	HIFI_WBS_H_IF_POWER_phase2	1026	3	25			30		1E
TM	HIFI_WBS_V_IF_POWER_phase1	1026	3	25			31		1F
TM	HIFI_WBS_V_IF_POWER_phase2	1026	3	25			32		20
TM	FCU parameter scan	1026	3	25			266		10A
TM	FCU diplexer scan with IF power	1026	3	25			258		102
TM	FCU diplexer scan without IFpower	1026	3	25			270		10E
TM	LCU IV curve	1026	3	25			259		103
TM	HRS_H tune report	1026	3	25			260		104
TM	HRS_V tune report	1026	3	25			261		105
TM	WBS_H tune report	1026	3	25			262		106
TM	WBS_V tune report	1026	3	25			263		107
TM	mixMagnetCurrent_useHRS report	1026	3	25			267		10B
TM	mixMagnetCurrent_useWBS report	1026	3	25			268		10C
TM	Vector scan report	1026	3	25			272		110
TM	Peakup step report	1026	3	25			272		110
TM	Peakup tune Chopper report	1026	3	25			273		111
TM	Peakup AOCS correction report	1026	3	25			274		112
TM	Engineering scan report	1026	3	25			172		AC
	Service type 5: Event reports								
TM	Peakup events	1024	5	1			0xC000	0xC000	
TM	HIFI_ready_event	1025	5	2				0x8008	
TM	HIFI_PM_test_event	1024	5	4				0x8001	

	Description	APID	Type	Subtype	FID	AID	SID	EventID	hex SID
TM	HIFI_Data_memory_event	1024	5	4				0x8002	
TM	HIFI_EEPROM_memory_event	1024	5	4				0x8003	
TM	HIFI_TC_verification_event	1024	5	4				0x8004	
TM	HIFI_Load_EEPROM_PM_event	1024	5	4				0x8005	
TM	HIFI_Load_DM_PM_event	1024	5	4				0x8006	
TM	HIFI_boot_DM_PM_event	1024	5	4				0x8007	
TM	HIFI_OBS_runtime_error	1024	5	4				0xA000	
TM	AV1_DHTR_C_OOL	1024	5	4			0xB001	0xB000	
TM	AV1_DHTR_C_OOL	1024	5	4			0xB002	0xB000	
TM	HWH_Laser_T_OOL	1024	5	4			0xB003	0xB000	
TM	HWV_Laser_T_OOL	1024	5	4			0xB004	0xB000	
TM	MX_H_nonresponse	1024	5	4			0xB005	0xB000	
TM	MX_V_nonresponse	1024	5	4			0xB006	0xB000	
TM	Chop_nonresponse	1024	5	4			0xB007	0xB000	
	Service type 6: Memory management								
TC	HIFI_Load_memory	1024	6	2					
TC	HIFI_Dump_Memory	1024	6	5					
TC	HIFI_Check_memory	1024	6	9					
TC	HIFI_Abort_Memorydump	1024	6	11					
TM	HIFI_memory_dump	1024	6	6					
TM	HIFI_memory_check	1024	6	10					
	Service type 8: Function management								
TC	HIFI_Housekeeping_off	1024	8	2	3	1			
TC	HIFI_Limit_checking_off	1024	8	2	4				
TC	HIFI_WH_Laser_T_check_off	1024	8	2	4	1			
TC	HIFI_WV_Laser_T_check_off	1024	8	2	4	2			
TC	HIFI_H_DHTR_C_check_off	1024	8	2	4	3			
TC	HIFI_V_DHTR_C_check_off	1024	8	2	4	4			
TC	HIFI_FCUnonresp_check_off	1024	8	2	4	5			
TC	HIFI_abort spectroscopy	1024	8	2	11				
TC	HIFI_Goto_safe	1024	8	4	17	0			
TC	HIFI_Set_OBS_ID	1024	8	4	1	0			
TC	HIFI_notify_PDU_status	1024	8	4	2	0			
TC	HIFI_Housekeeping_on	1024	8	4	3	1			
TC	HIFI_non_periodic_hk_FCU	1024	8	4	3	2			
TC	HIFI_non_periodic_hk_LCU	1024	8	4	3	3			
TC	HIFI_read_LCU_mem	1024	8	4	3	4			
TC	HIFI_Limit_checking_on	1024	8	4	4				
TC	HIFI_WH_Laser_T_check_on	1024	8	4	4	1			
TC	HIFI_WV_Laser_T_check_on	1024	8	4	4	2			
TC	HIFI_H_DHTR_C_check_on	1024	8	4	4	3			
TC	HIFI_V_DHTR_C_check_on	1024	8	4	4	4			
TC	HIFI_FCUnonresp_check_on	1024	8	4	4	5			

	Description	APID	Type	Subtype	FID	AID	SID	EventID	hex SID
TC	HIFI_Configure_FCU	1024	8	4	12	1	512		200
TC	HIFI_Configure_FCU_Power	1024	8	4	12	11	531		213
TC	HIFI_Config_HRS_H_att_lo	1024	8	4	12	7	513		201
TC	HIFI_Config_HRS_H_blocks	1024	8	4	12	8	514		202
TC	HIFI_Config_HRS_V_att_lo	1024	8	4	12	9	515		203
TC	HIFI_Config_HRS_V_blocks	1024	8	4	12	10	516		204
TC	HIFI_Configure_WBS_H	1024	8	4	12	4	517		205
TC	HIFI_Configure_WBS_V	1024	8	4	12	5	518		206
TC	HIFI_Configure_LCU1a	1024	8	4	12	12	519		207
TC	HIFI_Configure_LCU1b	1024	8	4	12	13	520		208
TC	HIFI_Configure_LCU2a	1024	8	4	12	14	521		209
TC	HIFI_Configure_LCU2b	1024	8	4	12	15	522		20A
TC	HIFI_Configure_LCU3a	1024	8	4	12	16	523		20B
TC	HIFI_Configure_LCU3b	1024	8	4	12	17	524		20C
TC	HIFI_Configure_LCU4a	1024	8	4	12	18	525		20D
TC	HIFI_Configure_LCU4b	1024	8	4	12	19	526		20E
TC	HIFI_Configure_LCU5a	1024	8	4	12	20	527		20F
TC	HIFI_Configure_LCU5b	1024	8	4	12	21	528		210
TC	HIFI_Configure_LCU6a	1024	8	4	12	22	529		211
TC	HIFI_Configure_LCU6b	1024	8	4	12	23	530		212
TC	HIFI_Configure_LCU7a	1024	8	4	12	24	531		213
TC	HIFI_Configure_LCU7b	1024	8	4	12	25	532		214
TC	HIFI_Configure_LCU_nominal	1024	8	4	12	30	533		215
TC	HIFI_Configure_LCU_diagnostic	1024	8	4	12	31	534		216
TC	HIFI_Configure_LCU_tables	1024	8	4	12	32	0		
TC	HIFI_FCU_parameter_scan	1024	8	4	7	1			
TC	HIFI_Sweep_Diplexer_without_IF	1024	8	4	7	2			
TC	HIFI_Sweep_Diplexer_with_IF	1024	8	4	7	3			
TC	HIFI_Engineering_Scan	1024	8	4	7	4			
TC	HIFI_Load_vector_scan_nominal	1024	8	4	8	5	550		226
TC	HIFI_Load_vector_scan_diagnostic	1024	8	4	8	6	551		227
TC	HIFI_vector_scan	1024	8	4	8	2			
TC	HIFI_tune_LO_Using_MXCH	1024	8	4	8	3			
TC	HIFI_tune_LO_Using_MXCV	1024	8	4	8	4			
TC	Tune HRS	1024	8	4	9	1			
TC	Tune WBS	1024	8	4	9	2			
TC	HIFI_Tune_mxmgc_useHRS	1024	8	4	9	3			
TC	HIFI_Tune_mxmgc_useWBS	1024	8	4	9	4			
TC	HIFI_WBS_zero	1024	8	4	10	3			
TC	HIFI_WBS_comb	1024	8	4	10	2			
TC	HIFI_HRS_functional_Test	1024	8	4	10	4			
TC	HIFI_Spectr_total_power	1024	8	4	11	1			
TC	HIFI_Spectr_fast_chop	1024	8	4	11	2			
TC	HIFI_Spectr_slow_chop	1024	8	4	11	3			
TC	Spectroscopy_freq_switch	1024	8	4	11	4			

	Description	APID	Type	Subtype	FID	AID	SID	EventID	hex SID
TC	HIFI_Configure_spectroscopy	1024	8	4	11	17			
TC	HIFI_simulate_Science	1024	8	4	16	127			
TC	Send_Single_Command	1024	8	4	12	26			
TC	HIFI_Reset_WBS_H	1024	8	4	12	26			
TC	HIFI_Reset_WBS_V	1024	8	4	12	26			
TC	HIFI_HL_Switch_off	1024	8	4	12	26			
TC	HIFI_HL_Standby	1024	8	4	12	26			
TC	HIFI_HL_Nominal	1024	8	4	12	26			
TC	HIFI_HL_Reset	1024	8	4	12	26			
TC	HIFI_set_HF_CH1_DHTR_C	1024	8	4	12	27			
TC	HIFI_set_HF_CV1_DHTR_C	1024	8	4	12	28			
TC	HIFI_HL_switchon	1024	8	4	12	29			
TC	HIFI_Configure_Peakup	1024	8	4	13	1	0		0
TC	HIFI_Acquire_Peakup_HRS	1024	8	4	13	2	0		0
TC	HIFI_Acquire_Peakup_WBS	1024	8	4	13	3	0		0
TC	HIFI_Peakup_Correction_Chopper	1024	8	4	13	4	0		0
TC	HIFI_Peakup_Correction_AOCS	1024	8	4	13	5	0		0
TC	HIFI_LCU_IV_curve	1024	8	4	15	1			
TC	HIFI_force_boot	1024	8	4	112	3			
TC	HIFI_load_boot	1024	8	4	112	2			0
TC	HIFI_EEPROM_Write	1024	8	4	16	2			0
TC	HIFI_Reset	1024	8	4	16	3			0
TC	HIFI_Jump_to_Boot	1024	8	4	16	4			0
TC	HIFI_check_PM	1024	8	4	16	5			0
TC	HIFI_copy_OBS	1024	8	4	16	6			0
TC	HIFI_Simulate_Peakup_	1024	8	4	16	126	1		1
TC	HIFI_Simulated_Spectroscopy	1024	8	4	16	127			0
TM	HIFI_Configure_FCU_report	1024	8	6	3073				0
TM	HIFI_Configure_FCU_power_report	1024	8	6	3083				0
TM	HIFI_Conf_HRS_H_att_lo_report	1024	8	6	3079				0
TM	HIFI_Conf_HRS_H_blocks_report	1024	8	6	3080				0
TM	HIFI_Conf_HRS_V_att_lo_report	1024	8	6	3081				0
TM	HIFI_Conf_HRS_V_blocks_report	1024	8	6	3082				0
TM	HIFI_Configure_WBS_H_report	1024	8	6	3076				0
TM	HIFI_Configure_WBS_V_report	1024	8	6	3077				0
TM	HIFI_Configure_LCU1a_report	1024	8	6	3084				0
TM	HIFI_Configure_LCU1b_report	1024	8	6	3085				0
TM	HIFI_Configure_LCU2a_report	1024	8	6	3086				0
TM	HIFI_Configure_LCU2b_report	1024	8	6	3087				0
TM	HIFI_Configure_LCU3a_report	1024	8	6	3088				0
TM	HIFI_Configure_LCU3b_report	1024	8	6	3089				0
TM	HIFI_Configure_LCU4a_report	1024	8	6	3090				0
TM	HIFI_Configure_LCU4b_report	1024	8	6	3091				0
TM	HIFI_Configure_LCU5a_report	1024	8	6	3092				0
TM	HIFI_Configure_LCU5b_report	1024	8	6	3093				0

	Description	APID	Type	Subtype	FID	AID	SID	EventID	hex SID
TM	HIFI_Configure_LCU6a_report	1024	8	6	3094				0
TM	HIFI_Configure_LCU6b_report	1024	8	6	3095				0
TM	HIFI_Configure_LCU7a_report	1024	8	6	3096				0
TM	HIFI_Configure_LCU7b_report	1024	8	6	3097				0
TM	HIFI_Configure_LCUnom_report	1024	8	6	3102				0
TM	HIFI_Configure_LCUdiag_report	1024	8	6	3103				0
TM	HIFI_Configure_LCUtables_report	1024	8	6	3104				0
TM	HIFI_Config_spectroscopy_report	1024	8	6	2833				0
TM	HIFI_Spectr_slow_chop_report	1024	8	6	2820				0
TM	HIFI_CH1_DHTR_C_report	1024	8	6	3099				0
TM	HIFI_CV1_DHTR_C_report	1024	8	6	3100				0
TM	HIFI_HL_switchon_report	1024	8	6	3101				0
TM	Local oscillator tune report nominal	1024	8	6	3095		534		216
TM	Local oscillator tune report diagnostic	1024	8	6	3096		535		217
	Service type 9: Time management								
TC	Enable time synchronisation	1024	9	4					
TC	Time code	1024	9	5					
TC	Enable time verification	1024	9	7					
TM	Time reference	1024	9	8					
TM	Time verification	1024	9	9					
	Service type 14: Packet transmission Control								
TC	HIFI_enable_TM	1024	14	1					
TC	HIFI_disable_TM	1024	14	2					
TC	HIFI_report_enabled_TM	1024	14	3					
TM	HIFI_TM_generation_status_report	1024	14	4					
	Service type 17: Test								
TC	HIFI_connection_test	1024	17	1					
TM	HIFI_connection_report	1024	17	2					
	Service type 21: Science								
TM	HIFI_HRS_H1_start	1028	21	1			1		1
TM	HIFI_HRS_H1_science24	1028	21	1			17		11
TM	HIFI_HRS_H2_start	1028	21	1			2		2
TM	HIFI_HRS_H2_science24	1028	21	1			18		12
TM	HIFI_HRS_V1_start	1029	21	1			3		3
TM	HIFI_HRS_V1_science24	1029	21	1			19		13
TM	HIFI_HRS_V2_start	1029	21	1			4		4
TM	HIFI_HRS_V2_science24	1029	21	1			20		14
TM	HIFI_WBS_H1_start	1030	21	1			5		5
TM	HIFI_WBS_H1_science16	1030	21	1			13		D
TM	HIFI_WBS_H1_science24	1030	21	1			21		15
TM	HIFI_WBS_H2_start	1030	21	1			6		6

	Description	APID	Type	Subtype	FID	AID	SID	EventID	hex SID
TM	HIFI_WBS_H2_science16	1030	21	1			14		E
TM	HIFI_WBS_H2_science24	1030	21	1			22		16
TM	HIFI_WBS_V1_start	1031	21	1			7		7
TM	HIFI_WBS_V1_science16	1031	21	1			15		F
TM	HIFI_WBS_V1_science24	1031	21	1			23		17
TM	HIFI_WBS_V2_start	1031	21	1			8		8
TM	HIFI_WBS_V2_science16	1031	21	1			16		10
TM	HIFI_WBS_V2_science24	1031	21	1			24		18