CARLO GAVAZZI SPACE S.p.A.

HSO/FIRST DPU

DOCUMENT TYPE : REQUIREMENTS **DRD :**

TITLE: DPU-BSW SOFTWARE REQUIREMENTS DOCUMENT

DOCUMENT No: DPU-SQ-CGS-001 **PAGE:** 1 **OF** 81

ISSUE No: 2 **DATE:** July 12, 2001

PREPARED BY: L.TAMPELLINI, A.BERTOLI

APPROVED BY:

THIS DOCUMENT IS SUBJECT TO THE APPROVAL OF:

	YES	NO	Approval Signature	Date
Product Assurance	X			
Configuration Control	X			
Project Control		X		
Engineering	X			
Managing Director		X		
CGS Program Manager	X			

	YES	NO	Approval Signature	Date
Higher Level Contractor	X			
Customer		X		

Data Management		
	Signature	Date

DOCUMENT CHANGE RECORD

ISSUE N°	DATE	CHANGE AUTHORITY	PAGES AFFECTED	REMARKS
2	July 12, **2001**			

LIST OF VALID PAGES									
PAGE	**ISSUE N°**	**PAGE**	**ISSUE N°**	**PAGE**	**ISSUE N°**	**PAGE**	**ISSUE N°**	**PAGE**	**ISSUE N°**
1-81	2								

TABLE OF CONTENTS

INDEX OF TABLES

INDEX OF FIGURES

1 INTRODUCTION

1.1 PURPOSE OF THE DOCUMENT

The purpose of this document is to state the applicable requirements for the “DPU Basic S/W” (DPU BSW) , which will be developed in the frame of the FIRST/PLANCK DPU project.

The FIRST DPU Basic S/W will be tested and delivered as integral part of the FIRST/PLANCK DPU H/W.

The content of this document follows the guidelines of ESA PSS-05-0 [RD1].

This document shall be the basis for the production of the DPU Basic S/W Architectural Design Document.

The DPU Basic S/W software requirements are mainly derived from [AD1] [AD2] [AD3] .

The DPU Basic S/W is prepared and maintained by the CGSPACE FIRST/PLANCK DPU software development team.

1.2 SCOPE OF THE SOFTWARE

The DPU Basic S/W constitutes the Board Support Package of the FIRST/PLANCK DPU Board allowing usage of provided HW services from the Virtuoso RTOS.

This S/W implements the following basic services :

- Boot Procedure
- IEEE 1355 I/F row data link drivers (for the SMCS332 device)
- MIL-STD-1553B I/F drivers: this document defines the S/W requirements to drive the ACE/Mini-ACE Integrated 1553 Terminal (DDC BU-61580, DDC BU-61582) .
- Driver for EEPROM writing
- Watchdog management

1.3 DEFINITIONS ACRONYMS AND ABBREVIATIONS

- **Acronyms**

AD#	Applicable Document number #
APF	Application Program File
ASI	Agenzia Spaziale Italiana
ATP	Authorization to proceed
BSP	Board Support Package
CGS	Carlo Gavazzi Space SpA
CNR	Consiglio Nazionale delle Ricerche
CPP	Coordinated Part Procurement
DPU	Data Processing Unit
DPU BSW	DPU Basic S/W
DDC	Data Device Corporation
DM	Data Memory
DPR	Dual Port Ram
DSP	The 21020 DSP microprocessor
EDAC	Error Detector And Corrector
EEPROM	Electrically Erasable Programmable Read Only Memory
EF	Empty FIFO
EM	Engineering Model
EPL	Emergency Program Loader
EPROM	Erasable Programmable Read Only Memory
EQM	Engineering Qualification Model
FIRST	Far Infra-Red and Sub-millimeter Telescope
FF	Full FIFO
FM	Flight Model
FPGA	Field Programmable Gate Array
FS	Flight Spare
FSDL	Fast Science Data Link
HF	Half Full FIFO
HIFI	Heterodyne Instrument for First

IFSI	Istituto per la Fisica dello Spazio Interplanetario
I/F	Interface
LSL	Low Speed Link
OBDH	On Board Data Handling
PA	Product Assurance
PACS	Photoconductor Array Camera and Spectrometer
PCB	Printed Circuit Board
PM	Program Memory
PROM	Programmable Read Only Memory
PL	Payload
RAM	Random Access Memory
RD#	Reference Document number #
SEU	Single Event Upset
S/C	Spacecraft
SPIRE	Spectral and Photometric Imaging Receiver

- **Definitions**

BOOT PROCEDURE The program that the FPGA must load into the program memory beginning by address 0x0

APPLICATION PROGRAM FILE The Application Program file located in the EEPROM (it includes the RTOS kernel, libraries and the application layer sw)

FRAME Set of MIL-STD-1553B standard messages which are transmitted/received in contiguous time slots

1.4 APPLICABLE DOCUMENTS

AD #	Doc Number	Issue/Date		Rev /Date		Title
1	CNR.IFSI.2000TR01	1	Sep 15, 2000			Documento di Specifiche Tecniche per il Contratto delle Data Processing Units del Satellite First dell'ESA" IFSI
2	DPU-SP-CGS-001	2	Oct 30, 2000			FIRST CPU Board Specification
3	DPU-SP-CGS-002	1	Oct 12, 2000			Payload & Spacecraft Interface Board Specification
4			Nov 30, 1999			Technical proposal CGS (Ref. S9-030 November 99)
5						Allegato Tecnico al Contratti ASI
6	CNR.IFSI.2001TR02	1	April 5, 2001			Herschel Space Observatory DPU Applicable Documents Guidelines
7	SCI-PT-ICD-7527	1	Sep 1, 2000	0		Packet Structure - Interface Control Document
8	IFSI-OBS-PL-2000	1	Oct 13, 2000			DPU/ICU OBS PA Plan
9	IFSI-OBS-SP-2000-001	1		3	April 6, 2000	DPU/ICU OBS URD, On Board S/W User Requirement Document
10	CNR.IFSI.2001TR01	Draft3	March 21, 2001			DPU/ICU Switch-ON Procedure

1.5 REFERENCE DOCUMENTS

RD #	Doc Number	Issue/Date		Rev /Date		Title
1	ESA PSS-05-0	2	Feb, 1991			ESA Software Engineering standards
2						Space Level MIL-STD-1553 BC/RT/MT Advanced Communication Engine Terminal BU-61582 Data Book ILC Data Device Corporation 1995
3						MIL-STD-1553 BC/RT/MT Advance Communication Engine (ACE) Data Book ILC Data Device Corporation
4				J-2	June, 1999	ACE/Mini-ACE Series BC/RT/MT Advanced Communication Engine Integrated 1553 Terminal… User's Guide ILC Data Device Corporation, june 1999, REV J-2
5	MIL-STD-1553B Change Notice 4		Jan 15,1996			Interface Standard for Digital Time Division Command/Response Multiplex Data Bus

1.6 REQUIREMENT NUMBERING

In this document has been applied the following requirements identification code

Reqts absolute ID: TBD#A-h1.h2.h3.h4-NNN *(see bottom the meaning of the symbols**)***

The Verification Method of the requirements is defined in the Verification matrix at the end of the document. The verification method follows the acronyms rules as specified in this paragraph.

Reqts absolute ID	Verification
TBD#A-h1.h2.h3.h4-NNN	***MMM–YY***

Where

Identification Code

TBD#A= An acronym identifying the current document

h1.h2.h3.h4= Section number of the document containing the requirement up to the fourth level (dot separated)

NNN = Requirement sequential number starting from 000.

Verification Method

MMM –YY

Where MMM can be :

SL = System Level

SEL = Subsystem Level/Equipment Level

YY can be a combination of :

D : Definition (not to be verified)

R : Review of Design

A: Analysis

T: Test

I : Inspection

2 SW DEVELOPMENT ENVIRONMENT

2.1 SOFTWARE DEVELOPMENT SYSTEM

The DPU Board is based on the ADSP 21020 Digital Signal Processing. DPU BSW programming and debugging is based on the ADSP 21020 software development tools, running on a PC - Windows 98 environment. The Figure 2-1:shows the process of developing an application using the ADSP 21020 development tools.

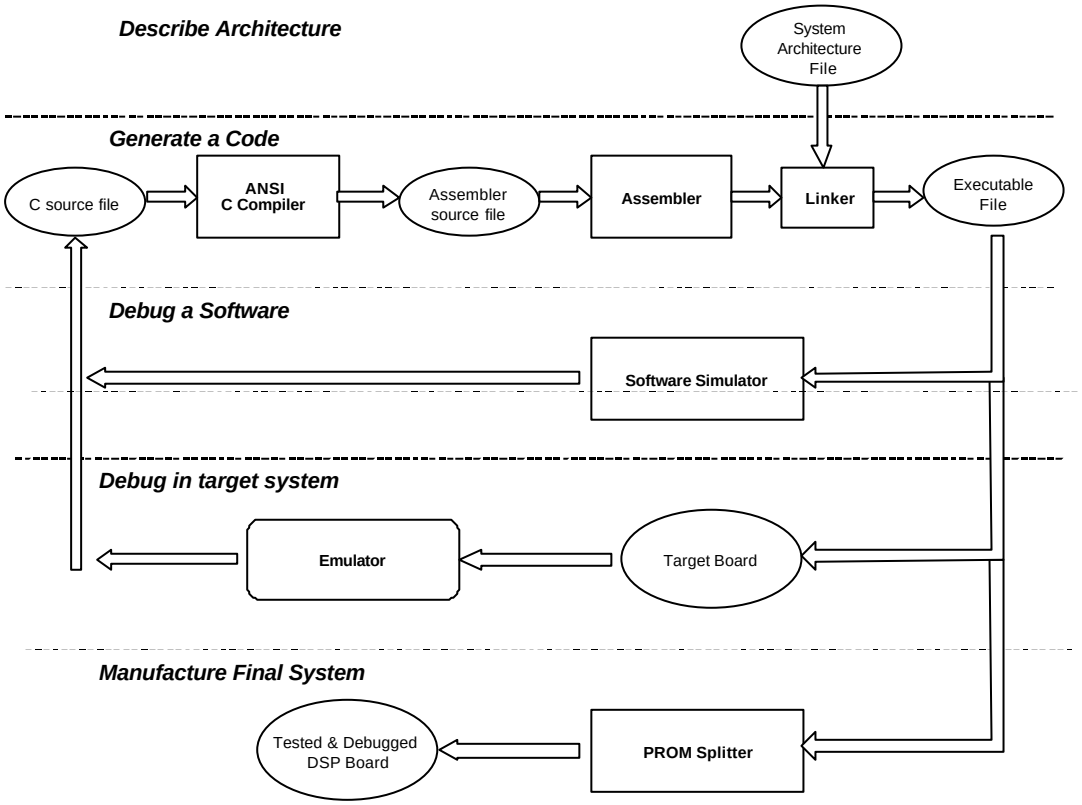

Figure 2-1: DSP system development

The ADSP 21000 family development software includes the G21K, an ANSI C compiler based on the industry standard GNU C Compiler (the Free Software Foundation). Other Components of the ADSP Family Development Software are: a C Runtime Library with custom DSP functions, CBUG C source level Debugger, Assembler, Assembly Library, Linker and Simulator.

- C Compiler & Runtime C Library. The G21K C Compiler reads source files written in ANSI-standard C and outputs ADSP-21020 assembly language files

- C source-Level Debugger. The CBUG debugger is integrated with the Software simulator and the emulator to provide symbolic debugging of C source code. debugger displays variable and expressions, with automatic updating. And evaluate ANSI-standard C expressions.

- Assembler. The assembler inputs a file of ADSP 21020 source code and outputs a relocatable object file.

- Linker. The linker process separately, assembled object and library files to create a single executable program. It assign memory locations to code and ddata in accordance with a user-defined architecture file, a text file that describes the target system hardware.

- Assembly Library. The assembly library contains standard arithmetic and DSP routines that can be called from program, Routines can be added to this library using library function.

- Simulator. The simulator executes an ADSP-21020 program in software in the same way that the processor would in hardware. The simulator also simulates the memory I/O devices specified in the architecture file.

- PROM splitter the PROM splitter translate an ADSP 21020 executable program into one of several formats (Motorola S3 (TBC)) that can be used to configure a PROM or downloaded to a target .

- The EZ-ICE Emulator. The EZ-ICE emulator provides hardware debugging capabilities for ADSP-21020 systems with standalone in circuit-emulation, running the target board processor in self-emulation mode. The EZ-ICE provides a fully speed emulation allowing inspection and modification of memory, registers, and processor stack. The EZ-ICE uses the IEEE 1149.1 JTAG test access port of the ADSP-21020 to monitor and control the target board processor during emulation.

2.2 COTS BOARDS TO SUPPORT THE SW DEVELOPMENT

The Hardware and Software Driver development and test is supported by usage of COTS Board to simulate the DPU external interface:

- IEEE 1355: a SPICEWIRE 4Links board (TBC) is used to test the IEEE1355 I/F and related S/W Driver. The SPICEWIRE board is controlled by Software tools provided with the board.

- MIL-STD-1553: a MIL-STD-1553 CONDOR BUS TOOL (TBC) is used to support the test of the FIRSt/PLANCK DPU MIL-STD-1553 I/F and S/W Driver. The CONDOR BUS TOOL provides all the functionality to simulate the First/Planck protocol and it can be configured as BC, BM, and RT.

3 FUNCTIONAL REQUIREMENTS

3.1 BOOT Procedure

3.1.1 INTRODUCTION

The boot procedure of an HSO instrument can only take place when there is direct link between ground and satellite. If there is one of the failures described in the following, the DPU/ICU is unable to go in the instrument INIT mode, that is, the DPU/ICU is in Rescue Program Loader mode. The S/C CDMU is then expected only to upload a new program image; under these conditions no HK, but only Events can get out of the DPU/ICU.

When the DPU/ICU model is switched on, that is the relevant 28 V lines dedicated to DPU/ICU are powered, the FPGA device loads the PROM content in the Program Memory while the DSP is kept in a RESET state and thus inactive (Reset is Asserted). At the end of the operation the DSP reset is released (Reset Deasserted) and the DSP starts program execution from the address 0x08: the Power ON Procedure starts. The first operation is the test of the Program Memory (PM)

If the test fails, the procedure generates a Telemetry Event Packet (Error/Alarm report) containing mainly the indication of the type (PM, DM, EEPROM) of the on-board memory where the check failed, the total number of the damaged and the ID of the damaged segments. The Program Memory, and this applies also for the Data Memory, can be divided in segments (pages), as an indication 256 segments/pages, in order to avoid a large generation of events if the memory damage is serious.

This damage information could allow the on ground generation of a new software image to be loaded only on valid memory segments.

After the PM test, even if the test failed, the Data Memory test is started.

If the test of the DM fails, then the procedure generates a Telemetry Event Packet (Error/Alarm report) with the same type of information indicated before.

After the DM test , the EEPROM content is checked, but in this case only the check sum(s) will be checked. This test can be made by comparing the checksum of the EEPROM pages with the corresponding values reported in the header of each page.

If the test of the EEPROM fails, then the procedure generates a Telemetry Event Packet (Error/Alarm report) with the same type of information indicated before. The Boot procedure will check the EEPROM segments including the Program to be uploaded in the Program Memory.

At the end of the three tests, if just one of them failed, then the procedure starts waiting for a tele-command (TC) from ground. The TC can either force the power on procedure to continue with the next step (load the application program from EEPROM to PM) or start a “Rescue Program Loader” procedure, capable to load the new code image from ground by using the Memory Management service (memory up-load TC). In this case the new executable is first loaded in DATA RAM and then copied from DM to PM in not corrupted/defective memory segments. This will be possible because the new executable will have the same structure as an EEPROM software image, which shall be organised in segments, each of which with its own header containing an indication of the PM address where the segment shall be mapped.

If the procedure of copying the new image from DM to PM fails, as well as in case of error in the reception of the memory load telecommand, the procedure generates and event and starts waiting for a new telecommand from ground.

The new loaded executable shall not be copied in EEPROM at this stage.

If all tests were OK, the next step in the procedure is to copy the OBS (application software) from EEPROM to PM. A new and final check is performed on the program copied in PM, again check sum(s). In case of successful check the program jumps directly to the OBS first location. If this final checksum test fails a new TM Event packet is generated and the procedure waits for a new telecommand from ground.

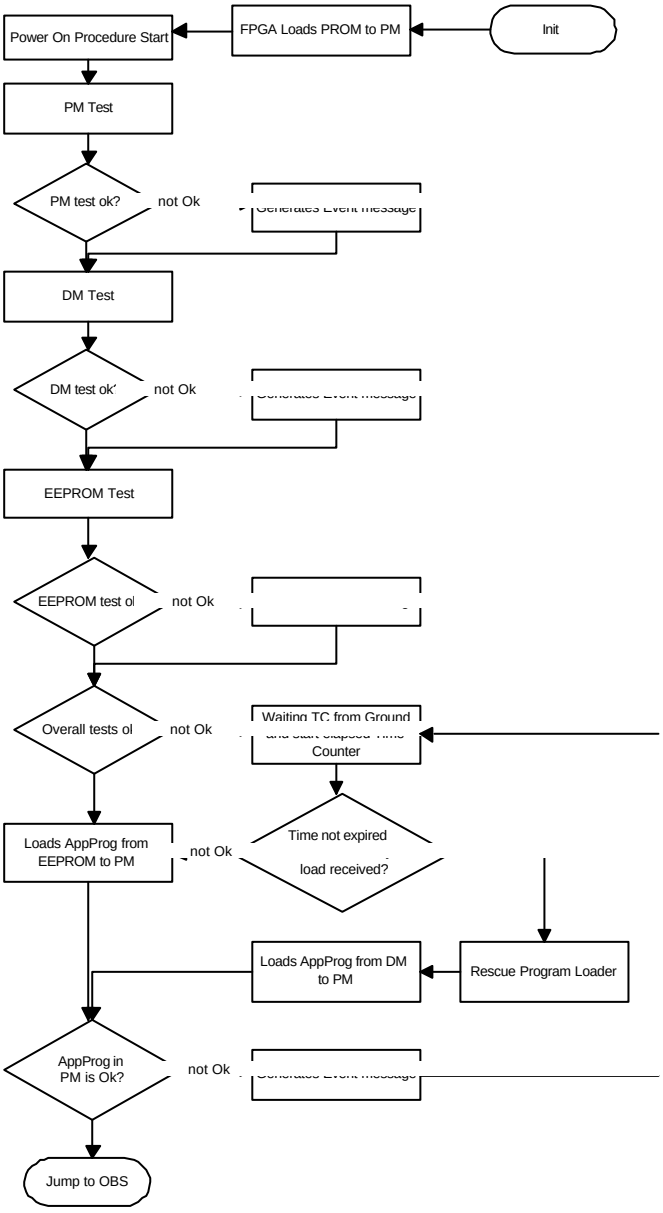

Figura 3-1: Boot Procedure flow chart

SRD-3.1.1.0-000

The Boot Procedure shall be as defined in [AD-10] document

Note: The DM, PM, EEPROM Test philosophy, the EEPROM to PM upload and the DM to PM upload are detailed in the next sections.

In order to trace the requirement in [AD-10] document an Equivalence table is inserted in this requirements Document. The function of this table is to redefine the numbering of the requirements in [AD-10] for the current SRD, maintaining the same requirements of the original document [AD-10] and avoiding to report them.

SRD requirements	[AD-10] requirements
SRD-3.1.1.0-040	PROM-ON-01
SRD-3.1.1.0-050	PROM-ON-02
SRD-3.1.1.0-060	PROM-ON-03
SRD-3.1.1.0-070	PROM-ON-04
SRD-3.1.1.0-080	PROM-ON-04.1
SRD-3.1.1.0-090	PROM-ON-04.2
SRD-3.1.1.0-100	PROM-ON-04.3
SRD-3.1.1.0-110	PROM-ON-04.4
SRD-3.1.1.0-120	PROM-ON-04.5
SRD-3.1.1.0-130	PROM-ON-04.6
SRD-3.1.1.0-140	PROM-ON-04.7
SRD-3.1.1.0-150	PROM-ON-05
SRD-3.1.1.0-160	PROM-ON-06
SRD-3.1.1.0-170	PROM-ON-06.1
SRD-3.1.1.0-180	PROM-ON-06.2
SRD-3.1.1.0-190	PROM-ON-06.3
SRD-3.1.1.0-200	PROM-ON-06.4
SRD-3.1.1.0-210	PROM-ON-06.5
SRD-3.1.1.0-220	PROM-ON-06.6
SRD-3.1.1.0-230	PROM-ON-07
SRD-3.1.1.0-240	PROM-ON-08
SRD-3.1.1.0-250	PROM-ON-09
SRD-3.1.1.0-260	PROM-ON-09.1
SRD-3.1.1.0-270	PROM-ON-09.2
SRD-3.1.1.0-280	PROM-ON-09.3
SRD-3.1.1.0-290	PROM-ON-09.4
SRD-3.1.1.0-300	PROM-ON-09.5
SRD-3.1.1.0-310	PROM-ON-09.6
SRD-3.1.1.0-320	PROM-ON-010
SRD-3.1.1.0-330	PROM-ON-011
SRD-3.1.1.0-340	PROM-ON-012
SRD-3.1.1.0-350	PROM-ON-013
SRD-3.1.1.0-360	PROM-ON-013.1
SRD-3.1.1.0-370	PROM-ON-014

Table 3-1: Equivalence Table between [AD-10] and SRD. The [AD-10] are all applicable to the SRD

SRD-3.1.1.0-010

The Boot Procedure shall be able to acquire the TC's via MIL-STD-1553B when the EEPROM test has been completed and an EEPROM/DM/PM failure has been detected.

SRD-3.1.1.0-020

After the EEPROM test completion, If an EEPROM/DM/PM failure has been detected, the Boot Procedure shall wait for TBD seconds a "Memory Management TC" in order to upload the Application Program via MIL-STD-1553B or "Function Management TC" in order to force the Application Program uploading by EEPROM.

SRD-3.1.1.0-030

When the defined Time is expired and a TC has not been received, the Boot Procedure shall force the Application Program uploading by EEPROM.

3.1.2 EEPROM, DM SEGMENTATION AND FORMAT OF THE SEGMENT

The section details the format of a generic segment loaded in EEPROM and in Data Memory. A generic segment size is set to 1024x32bits. Considering that the EEPROM capacity is 256K x 32bits wide, the EEPROM can store maximum 256 pages of 1024 x 32bits size. The Data Memory capacity is 512K x 32bits wide and it can store maximum 512 page of 1024x32bits size. This value has the following reason:

- Optimizes the Cyclic Redundancy Check Code (Frame Check Sequence) limiting the encoded block of less 32768bits. In this way the FCS has the capabilities to maximize the error detection. It is the maximum power of 2 value minor the 32768/ 32 = 1024 x 32bits.

Notes: Concerning the MIL-STD-1553B Application Program uploading, the Data field of the Loading Data in memory Tele-Command packet is only 242 bytes length. This means that for uploading a page/segment of Application program are necessary about 32 TC packets.

The following pictures shows the EEPROM/DM segmentation. For each page 1024x32bit is shown the header and the data. The header is located in the first 6x32bits words of the page

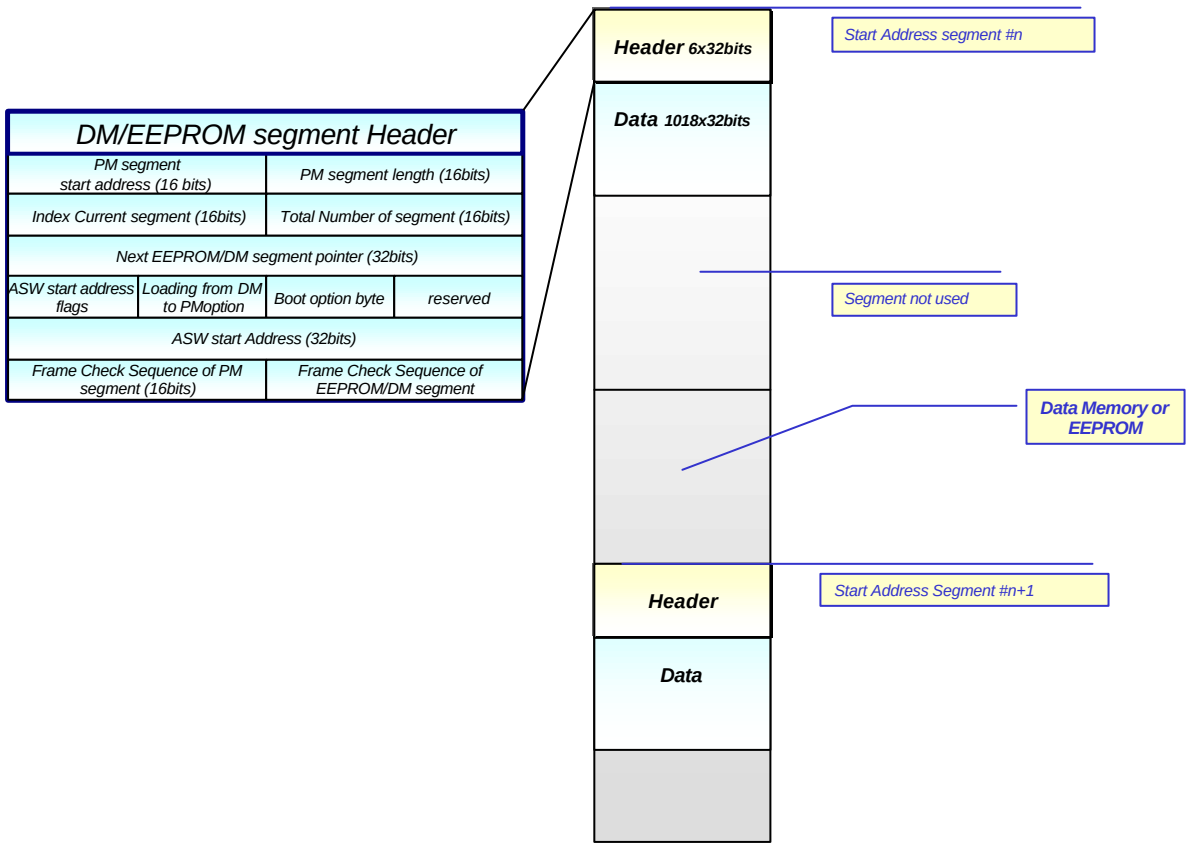

Figure 3-2: EEPROM / Data Memory Structure

In the following table is detailed the header of a generic EEPROM/DM page. It is 6x32bits words and contains the following field:

- PM segment start address pointer (32bits): It is the starting address for coping the data from EEPROM/DM to PM.
- Index of the current segment : the application program is shared in n segments. Each segment shall be numbered in progressive order. The field represents the number of the current segment.
- Total Number of the segments: it is the total number of segments required for storing the Application Program.
- PM length: it is the length of the segment to be copied in the PM.
- Next EEPROM/DM segment pointer: it is the address of the successive segment in the EEPROM or DM. This pointer allows to select the next segment in EEPROM/DM to be loaded in Program Memory(32bits).
- ASW start address flag: if it is set the boot program reads the ASW entry address (start address) written in the start address field. The current ASW start address is the ASW start address in the last segment uploaded in Program Memory with set the ASW start Address flag.
- Loading from DM to PM: it assumes three values; 1) Loading the current segment into Program Memory from Data Memory; 2) Loading of all the uploaded DM segments into program memory (PM). 3) waiting for uploading in Program Memory.
- Boot Option byte: it allows to select two options; Boot from primary partition and Boot from secondary partition. This field is considered for the first and the last segments. In all other segments the field is ignored.
- ASW start address: Application Software program entry point.
- Frame Check Sequence of the PM segment;
- Frame Check Sequence of the EEPROM/DM segment.

Header of a generic segment in EEPROM /Data Memory

Byte #4	Byte #3	Byte #2	Byte #1
PM segment start address pointer (32bits)		PM segment length (16 bits) (one page is 2048 Bytes)	
Index Current Segment		Total Number of Segments	
next EEPROM/DM segment pointer (32bits)			
ASW start address flag	Loading from DM seg. to PM Seg. options	Boot options Byte	reserved
ASW start address (32bits)			
Frame Check sequence of the PM segment (16 bits)		Frame Check sequence of EEPROM/DM segment (16 bits)	

Figure 3-3: Header of a generic segment in EEPROM / DATA memory

SRD-3.1.2.0-000

The EEPROM shall be shared in 256 segments of 1024x32bits.

SRD-3.1.2.0-005

The Data Memory shall be shared in 512 segments of 1024x32bits.

SRD-3.1.2.0-010

The EEPROM and Data Memory segments shall have the same segment structure. The single segment/page structure shall include a 6x32bits Header and 1018x32bits data.

SRD-3.1.2.0-020

The Header shall be composed of the following field as define in the table:

- *PM segment start address (16bits): contains the PM start address for coping the segment (see Figure 3-4).*
- *PM segment Length (16bits): contains the Lenght of the segment to be copied in Program memory (see Figure 3-4).*
- *Index current segment: indicates the number of the current segment (Max 512 segments)*
- *Total Number of the segments (Max 512 segments)*
- *Next EEPROM Segment Pointer (32bits) (see Figure 3-5).*
- *ASW start address flag: if set the Boot program reads the ASW Start Address field.*

- *Loading from Data Memory segment to Program Memory segment. The possible values are the following:*
  1. *Writes directly the current segment into Program memory*
  2. *Writes all the segments already uploaded in Data memory to Program memory.*
  3. *Waits before writing the segment in data memory to program memory*
- *Boot option byte: if set to 1 Boot from primary partition, if set to 2 Boot from secondary partition.*
- *ASW start address: entry point of the ASW program (32bits)(see Figure 3-6).*
- *Frame Check sequence of PM segment (16bits) (see Figure 3-7).*
- *Frame Check sequence of EEPROM/DM segment (16bits) (see Figure 3-7).*

PM start address and PM length of segment fields utilization

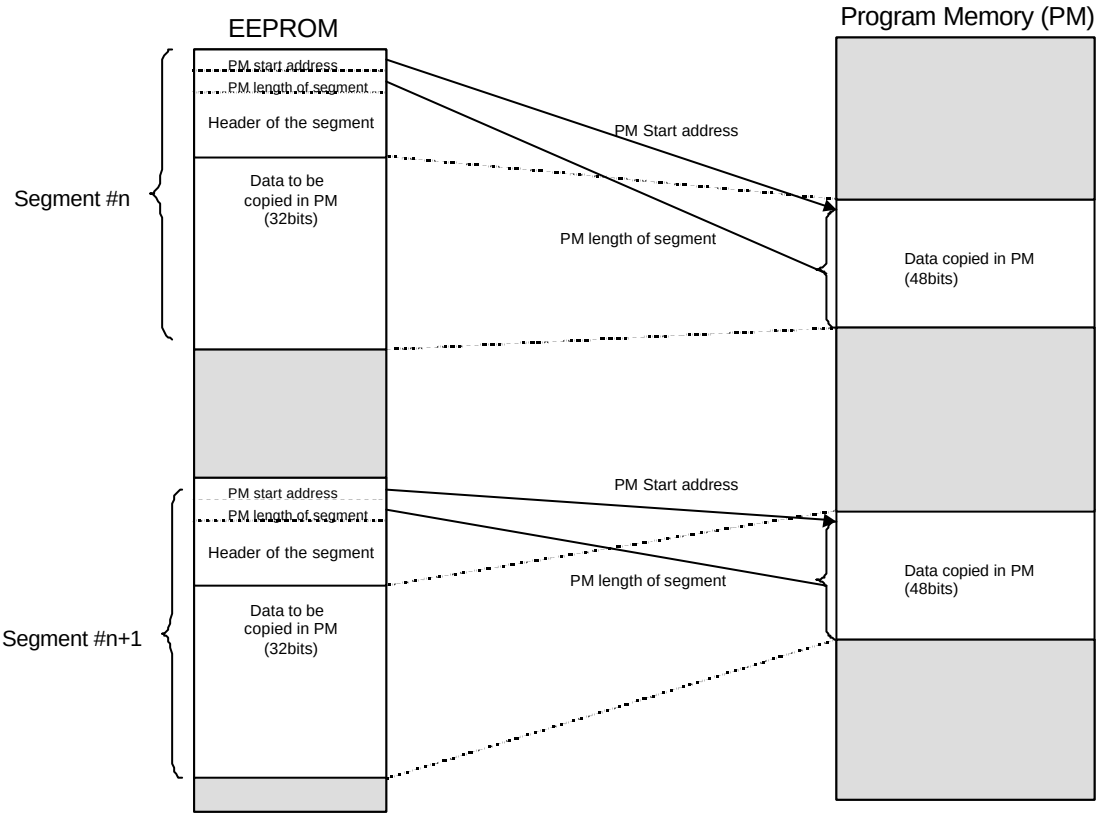

Figure 3-4: "PM start address" and "PM Length of segment" fields utilization

EEPROM pointer to the next segment field utilization

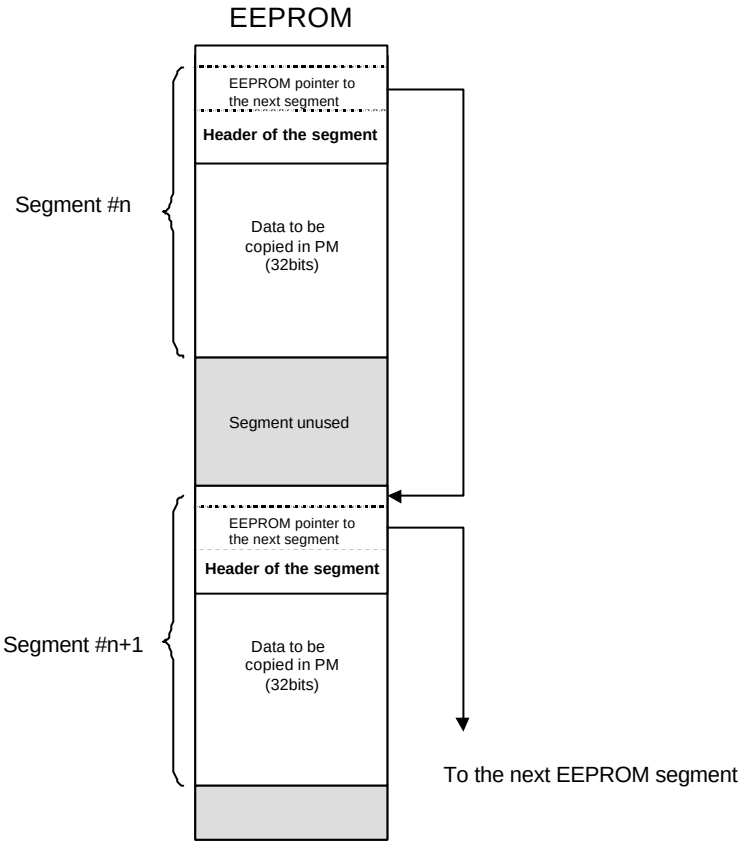

Figure 3-5: "EEPROM pointer to the next segment" field utilization

ASW start address field utilization

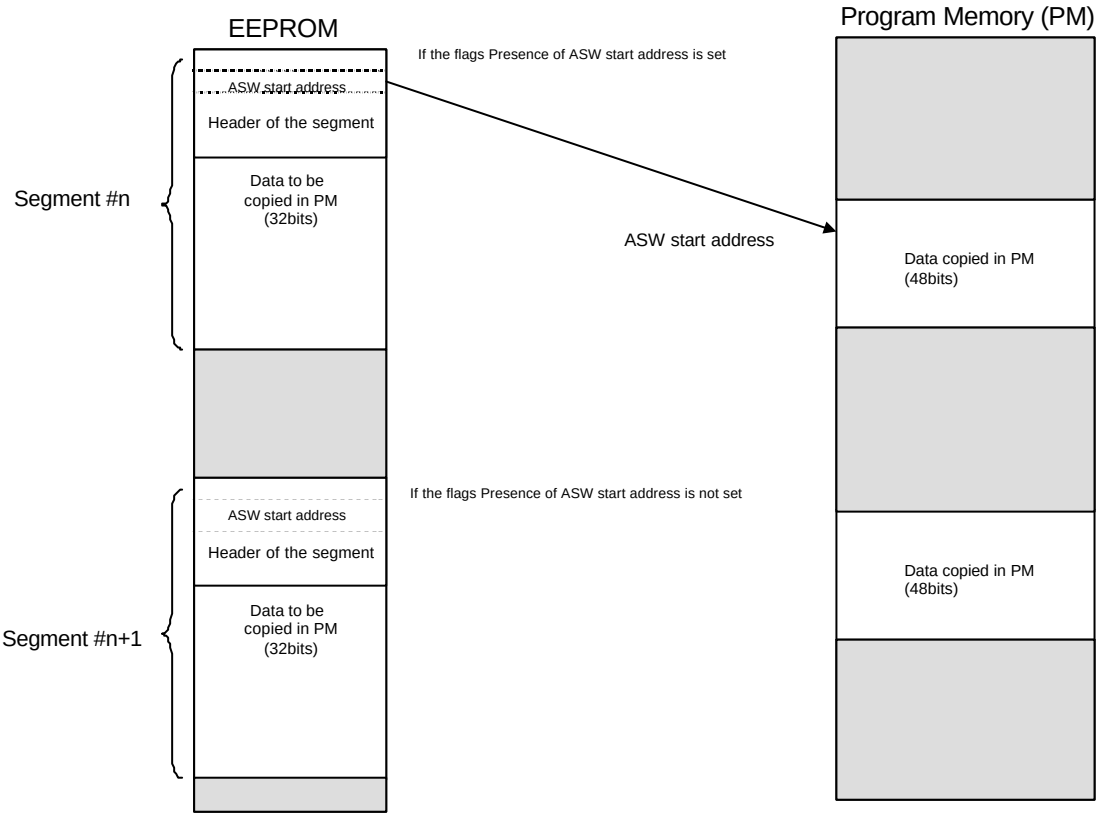

Figure 3-6: "ASW start address" field utilization

PM FCS e EEPROM FCS fieldsutilization

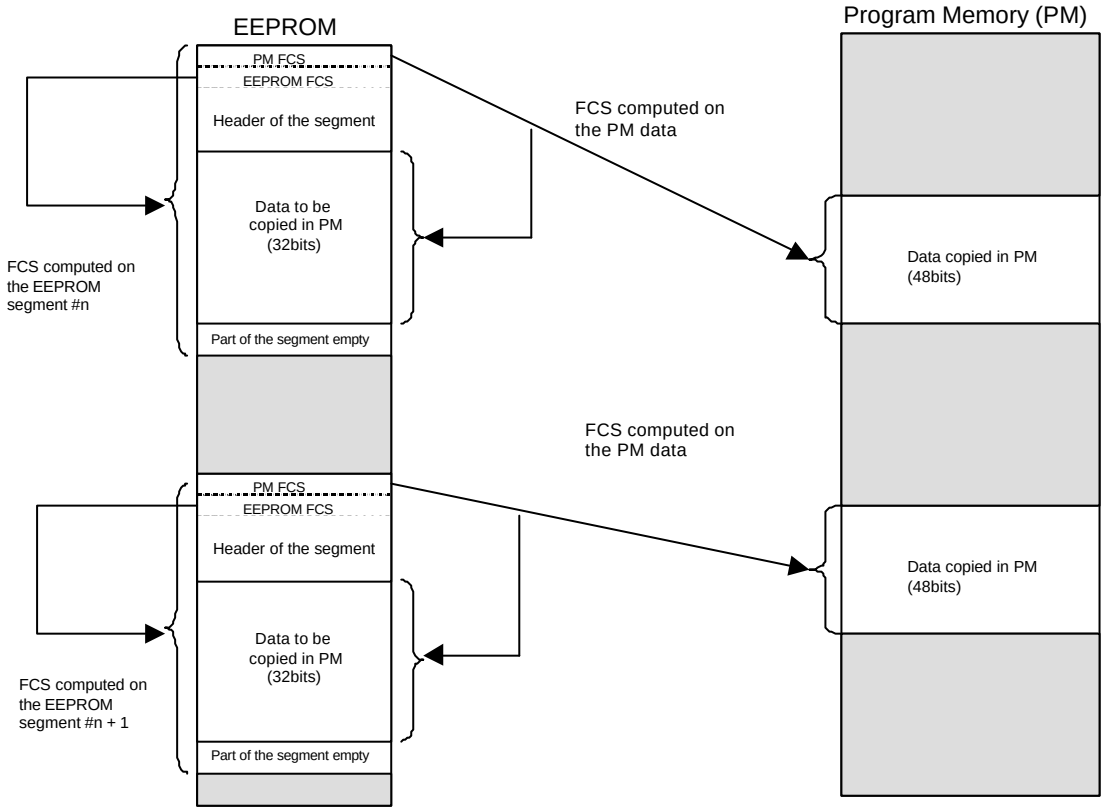

Figure 3-7: "PMC FCS" and "EEPROM FCS" fields utilization

3.1.3 PROGRAM MEMORY SEGMENTATION

The Program Memory is segmented in 1024 segments/pages of 1024x48bits in analogue way to the EEPROM and DM. The only difference concerns the words length 48bits instead 32bits. The segmentation allows a more simple test philosophy only. When a single or more memory cells are corrupted the related segment will be declared corrupted and not used. This method avoids to transmit information for all the corrupted cells and the information concerning the segment in failure is only transmitted.

The segmentation don't limit the users management of the Program Memory in term of unused cells. Two or more DM/EEPROM segment could be copied in the same PM segment if the space is available and all the PM space can be utilized;

SRD-3.1.3.0-000

The Program Memory shall be segmented in 1024 segment of 1024x48bits.

3.1.4 DATA MEMORY TEST REQUIREMENTS

SRD-3.1.4.0-000

The Data Memory Test shall be as shown in the flow chart in Figure 3-8

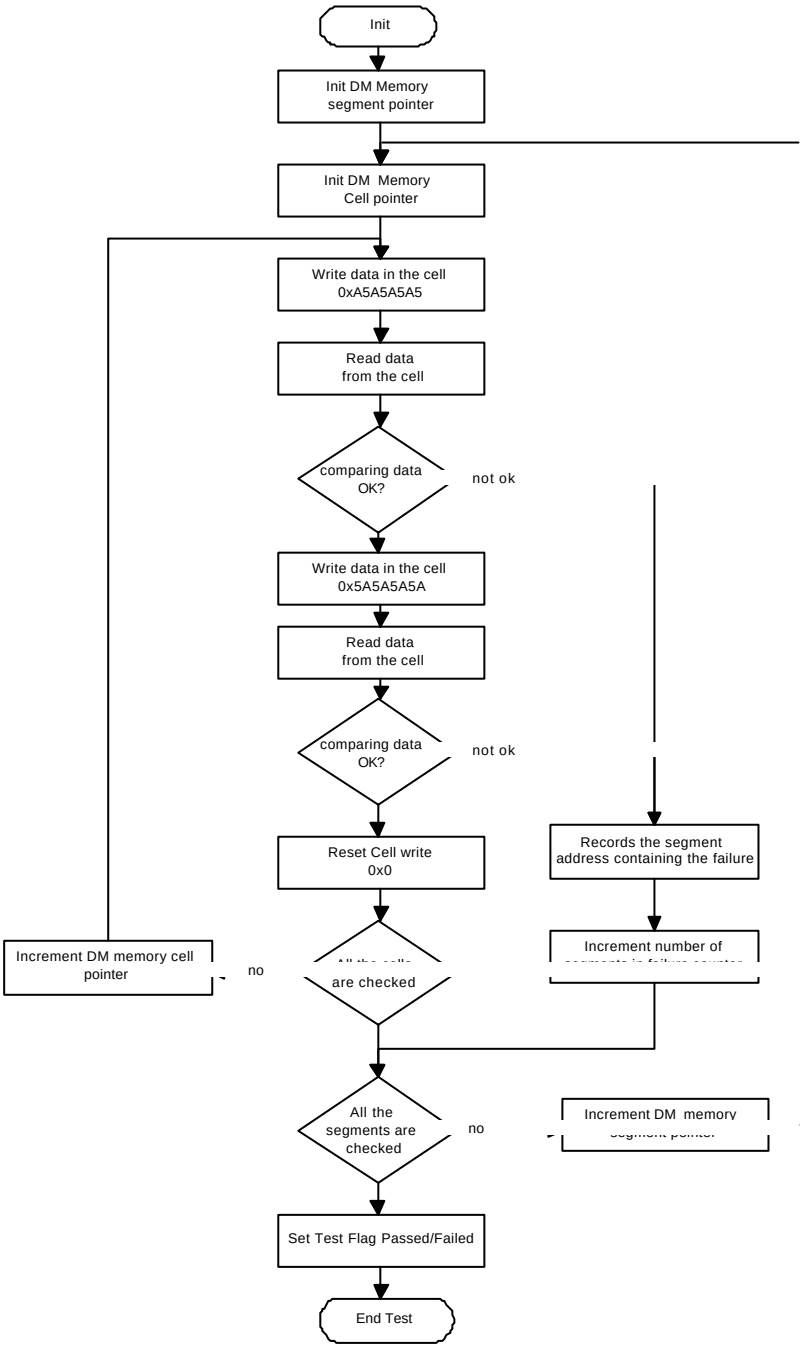

Figure 3-8: Data Memory Test Procedure

SRD-3.1.4.0-010

The single Data Memory cell shall be checked in two steps; in the first step the values (Hex) 0xA5A5A5A5 is written in the memory and than it is read. In the second step the 0x5A5A5A5A is written and than is read and checked. The read Data is compared with the value and a failure is generated in case of not equal data The two values above reported are choosen in order to stress all the bits of the 32bits memory cells in independent way.

SRD-3.1.4.0-020

The Boot Software shall record the Address of the segment in failure when the failure of one memory cell inside the segment is detected at least. After the recoding of the Address of the segment in failure, the Boot Software shall test the next segment.

SRD-3.1.4.0-030

The Boot Software shall send the Event Report message containing the number of segments in failure and the address of the segments in failure, when the test of all the DM segments has been complited. The Event Report shall be generated when a segment is declared corrupted at least.

SRD-3.1.4.0-040

The Boot Software shall send in an Event Report message up to 20 Addresses of corrupted segments. In case of number of corrupted segments greater than 20, more than one Event Report message shall be consecutively generated in order to report all the segments in failure.

3.1.5 PROGRAM MEMORY TEST REQUIREMENTS

SRD-3.1.5.0-000

The Program Memory Test shall be as shown in the Flow chart in Figure 3-9

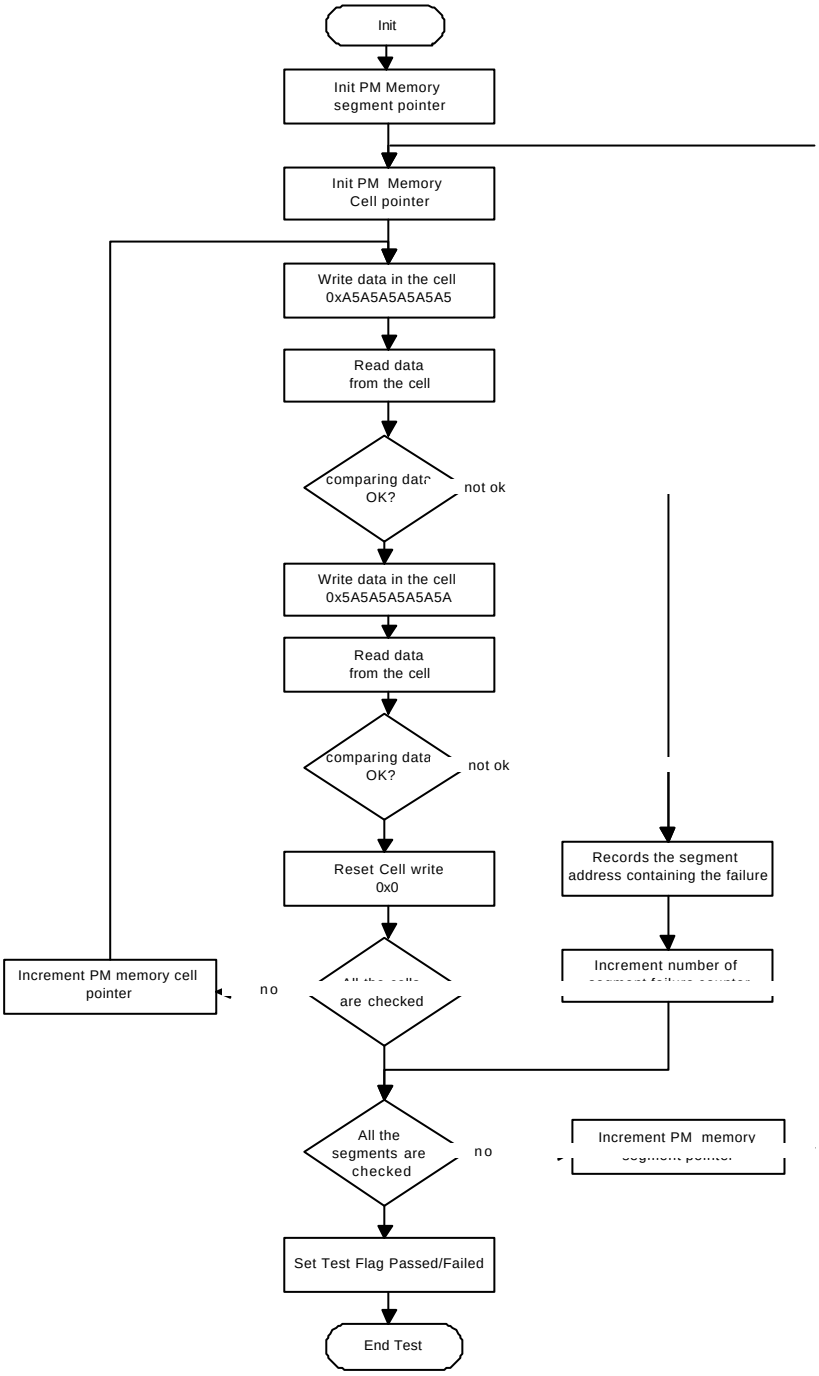

Figure 3-9: Program Memory Test Procedure

SRD-3.1.5.0-010

The single Data Memory cell shall be checked in two steps; in the first step the values (Hex) 0xA5A5A5A5A5A5 is written in the memory and than it is read. In the second step the 0x5A5A5A5A5A5A is written and than is read and checked. The read Data is compared with the value and a failure is generated in case of not equal data The two values above reported are choosen in order to stress all the bits of the 32bits memory cells in independent way.

SRD-3.1.5.0-020

The Boot Software shall record the Address of the segment in failure when the failure of one memory cell inside the segment is detected at least. After the recoding of the Address of the segment in failure, the Boot Software shall test the next segment.

SRD-3.1.5.0-030

The Boot Software shall send the Event Report message containing the number of segments in failure and the address of the segments in failure, when the test of all the DM segments has been complited. The Event Report shall be generated when a segment is declared corrupted at least.

SRD-3.1.5.0-040

The Boot Software shall send in an Event Report message up to 20 Addresses of corrupted segments. In case of number of corrupted segments greater than 20, more than one Event Report message shall be consecutively generated in order to report all the segments in failure.

3.1.6 EEPROM MEMORY TEST REQUIREMENTS

SRD-3.1.6.0-000

The EEPROM Memory Test procedure shall be as shown in the flow chart in Figure 3-10, Figure 3-11

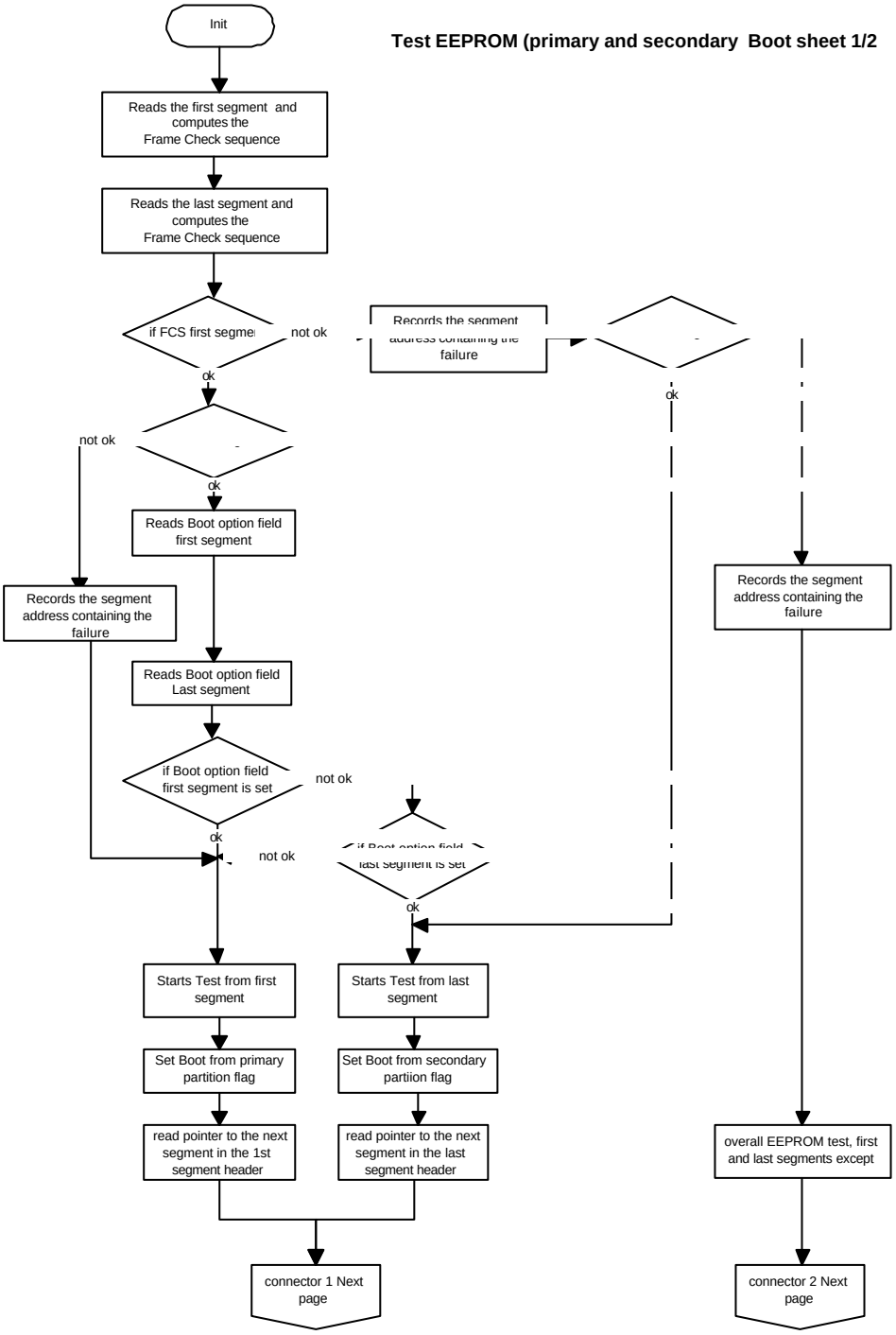

Figure 3-10: EEPROM Test Procedure with Primary and secondary Boot

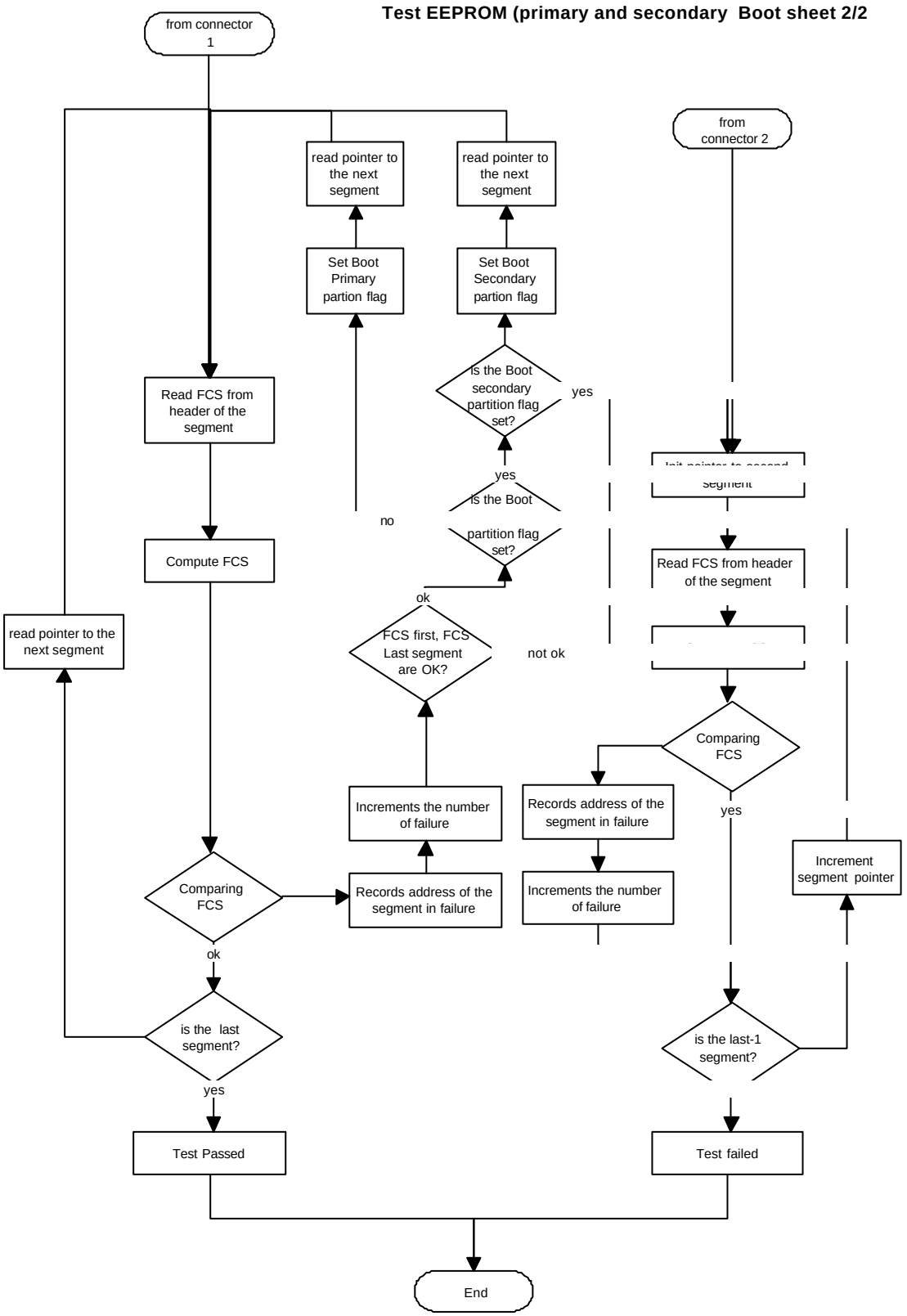

Figure 3-11: EEPROM Test Procedure with Primary and Secondary Boot

SRD-3.1.6.0-010

The Boot Software shall manage a primary and secondary boot option by reading the flag "Boot Option" in the EEPROM header.

SRD-3.1.6.0-020

The Boot Software shall start the EEPROM test computing the FCS of the first EEPROM segment and of the last EEPROM segment and comparing them with the "FCS EEPROM" field of the EEPROM Header .

SRD-3.1.6.0-030

The Boot Software shall perform the EEPROM test of the segments by checking the FCS of the segment in the "FCS EEPROM" field of the EEPROM header.

SRD-3.1.6.0-040

The Boot Software shall send an Event Report Message containing the EEPROM segment in failure on the EEPROM test completion.

SRD-3.1.6.0-050

The Boot Software shall report an Event Report Message if an EEPROM segment at least is corrupted.

SRD-3.1.6.0-060

The Boot Software shall perform an EEPROM test overall (first and last segment exception) if the primary and secondary EEPROM test have failed.

SRD-3.1.6.0-070

The Boot Software shall skip the test of unused EEPROM segment. The EEPROM segment shall be declared unused if the 6 Words x32bits header are filled with 0.

3.1.7 EEPROM TO PM UPLOADING REQUIREMENTS

SRD-3.1.7.0-000

The Boot Software shall perform the EEPROM to PM uploading as shown in the flow chart in Figure 3-12

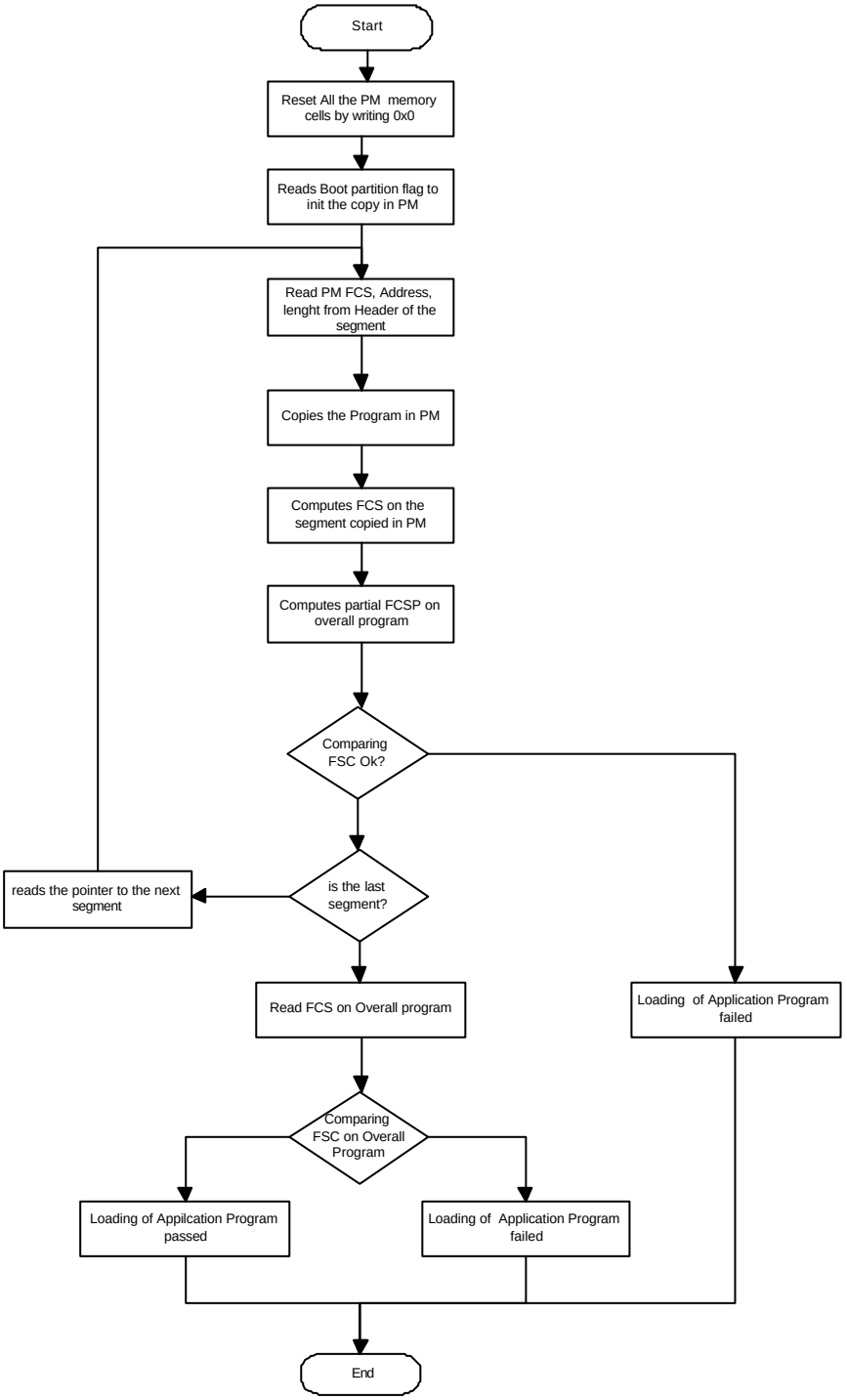

Figure 3-12: EEPROM To PM uploading procedure

SRD-3.1.7.0-010

The Boot Software shall send an Event Report Message on the "EEPROM to PM uploading" completion, if an uploaded EEPROM segment to PM at least is failed.

SRD-3.1.7.0-020

The Boot Software shall check the correctness of the uploaded PM segment by checking the "FCS Program Memory" related to the uploaded segment.

SRD-3.1.7.0-030

The Boot Software shall check the correctness of the uploaded Application Program in PM by checking the FCS on overall the Application Program.

SRD-3.1.7.0-040

The Boot Software shall upload the EEPROM segments into PM on the base of the selected Boot partition flag during the EEPROM test.

3.1.8 DM TO PM UPLOADING REQUIREMENTS

SRD-3.1.8.0-000

The Boot Software shall perform the DM to PM uploading procedure as shown in the flow chart in Figure 3-13

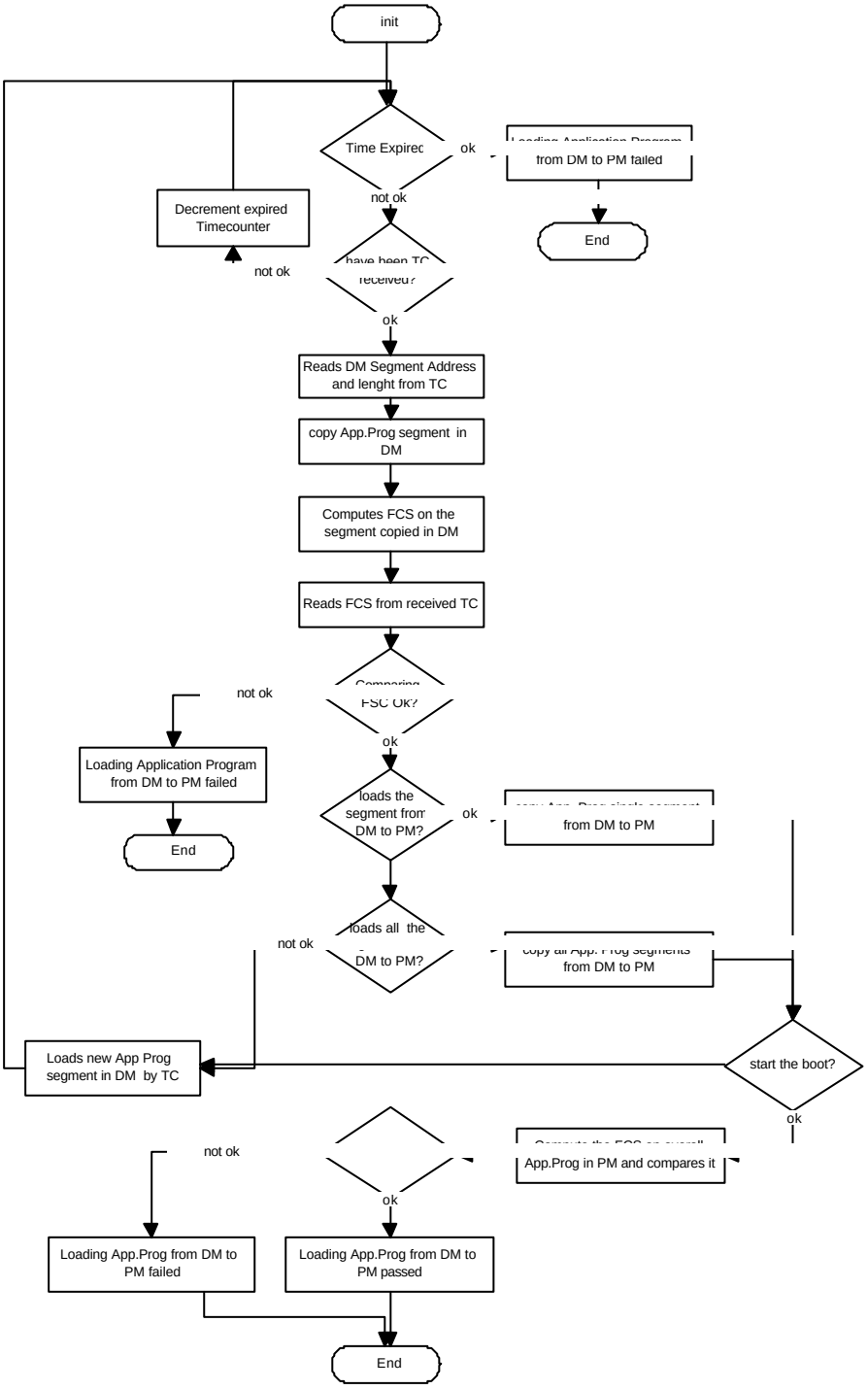

Figure 3-13: DM to PM uploading procedure

SRD-3.1.8.0-010

The Boot Software shall send an Event Report Message if the uploading of the Application software from DM to PM uploaded is failed.

SRD-3.1.8.0-020

The Boot Software shall check the correctness of the uploaded Application S/W program in PM by checking the FCS on overall the Application S/W Program.

SRD-3.1.8.0-030

The Boot Software shall be able to receive one Telecommand every 15,625 millisecond (TBC) for uploading a single segment of the Application S/W program.

SRD-3.1.8.0-040

The Boot Software shall be able to receive consecutive DM segments up to 1Hz (TBC) for uploading the Application S/W program, when the uploading option from DM to PM of the memorized DM segments are not selected. (In this case the frequency will be lower than 1Hz).

3.2 IEEE 1355 DRIVERS

SRD-3.2.0.0-000

The 1355 Drivers Sw shall set the SMCS332 and allow data transfer in the "Transparent" SMCS332 operation mode only.

SRD-3.2.0.0-010

The 1355 Drivers Sw shall allow to perform a complete reset of the SMCS332 links

SRD-3.2.0.0-020

The 1355 Drivers Sw shall allow to individually set the transmit speed of each of the three link to the following values :

Link Speed	
Link_speed_10000_Kbit	10 Mbit/sec
Link_speed_2500_Kbit	2,5 Mbit/sec
Link_speed_1250_Kbit	1,25 Mbit /sec

SRD-3.2.0.0-030

The 1355 Drivers Sw shall allow to set the time-out value for the SMCS332 (Time out will be only used if a read or write data is set to wait until completion)

SRD-3.2.0.0-030

The 1355 Drivers Sw shall allow to open each one of the three link of the SMCS

SRD-3.2.0.0-040

The 1355 Drivers Sw shall allow to close each one of the three link of the SMCS

SRD-3.2.0.0-050

The 1355 Drivers Sw shall allow to starts a SMCS332 Link as a Master at a specified Link speed.

SRD-3.2.0.0-060

The 1355 Drivers Sw shall allow to starts a SMCS332 Link as a Slave at a specified Link speed.

SRD-3.2.0.0-070

The 1355 Drivers Sw shall allow to stops a currently running SMCS332 Link.

SRD-3.2.0.0-080

The 1355 Drivers Sw shall allow to gets the status register for a SMCS332 Link.

SRD-3.2.0.0-090

The 1355 Drivers Sw shall allow to transmit <T_ByteSize> fixed number of Bytes starting at address <pBuffer> over the link <nLink> of the <smcs_ID> device .

SRD-3.2.0.0-100

The 1355 Drivers Sw transmission function shall allow the user to set a <bWait> parameter. If bWait is true, the transmission function returns immediately, other way it returns when the transfer is compleited or an error is detected.

SRD-3.2.0.0-120

The 1355 Drivers Sw shall allow to receive , over the link <nLink> of the <smcs_ID> device, <R_ByteSize> fixed number of Bytes an writes them in a user buffer starting at address <pBuffer> .

SRD-3.2.0.0-130

The 1355 Drivers Sw receiving function shall allow the user to set a <bWait> parameter. If bWait is true the functions returns immediately, other way it returns when the transfer is completed or an error is detected.

SRD-3.2.0.0-140

The 1355 Drivers Sw shall allow to return the current transmit status for the specified link <nLink>

Possible values of such status are are:

Transfer_not-started, Transfer_started, Transfer_Done, Transefr_error_diconnect, Transfer_error Parity, Transfer_error_timeout, Transfer_error_link_not _started, etcc.)

SRD-3.2.0.0-150

The 1355 Drivers Sw shall allow to return the current receive status for the specified link <nLink>

Possible values of such status are:

Transfer_not-started, Transfer_started, Transfer_Done, Transfer_error_diconnect, Transfer_error Parity,, Transfer_error_timeout, Transfer_error_link_not _started, etcc.)

SRD-3.2.0.0-160

The 1355 Drivers Sw shall allow to receive a defined number of packets <NumPackets> with variable length over the link <nLink> of the <smcs_ID>.

SRD-3.2.0.0-170

The 1355 Drivers Sw shall allow to return the number of received data over the link <nLink> of the <smcs_ID>.

SRD-3.2.0.0-180

The 1355 Drivers Sw shall allow to return the number of received packets over the link <nLink> of the <smcs_ID>

SRD-3.2.0.0-190

The 1355 Drivers Sw shall allow to return the number of transmitted data over the link <nLink> of the <smcs_ID>.

SRD-3.2.0.0-200

The 1355 Drivers Sw shall allow to write/read a generic <smcs_ID> 1355 register.

SRD-3.2.0.0-210

The 1355 Drivers Sw shall allow to write/read <nbytesize> to/from DPRAM of the <smcs_ID>.

3.3 MIL-STD-1553B DRIVERS

3.3.1 INTRODUCTION

The MIL-STD-1553B Drivers S/W shall provide functions to manage the Data Link Layer and part of the Transfer Layer as defined in the following sections.

3.3.2 TRANSFER LAYER

SRD-3.3.2.0-000

The MIL-STD-1553B Drivers S/W shall allow to create frames of MIL-1553B standard messages

SRD-3.3.2.0-010

The MIL-STD-1553B Drivers S/W shall be able to send and receive single MIL-1553B standard messages

3.3.3 DATA LINK LAYER

3.3.3.1 Remote Terminal Requirements

SRD-3.3.3.1-000

MIL-BUS DRV's S/W shall be able to configure the ACE/MiniACE chip as Remote Terminal only.

SRD-3.3.3.1-010

MIL-BUS DRV's S/W shall be able to support the Mode Code as defined in [AD-7] document.

SRD-3.3.3.1-020

MIL-BUS DRV's S/W shall be able to receive the BC->RT standard messages

SRD-3.3.3.1-030

The MIL-BUS DRV's S/W shall be able to transmit the RT->BC standard messages

NOTE: the MIL-BUS DRV's S/W will not be able the transmit or receive of RT->RT standard messages

SRD-3.3.3.1-040

The MIL-BUS DRV's S/W shall be able to receive Broadcast messages

SRD-3.3.3.1-050

The MIL-BUS DRV's S/W shall allow to read the Remote Terminal address.

SRD-3.3.3.1-060

The MIL-BUS DRV's S/W shall receive at address 31 the Broadcast messages.

3.3.3.2 ACE/MiniACE programming requirements

SRD-3.3.3.2-000

The MIL-BUS DRV's S/W shall initialize the ACE/MiniACE chip to operate as 1553B remote terminal setting the Configuration Register #1.

SRD-3.3.3.2-010

The MIL-BUS DRV's S/W shall manage the ACE/MiniACE Messages in the shared memory in Buffer Circular mode

SRD-3.3.3.2-020

The MIL-BUS DRV's S/W shall be able to detect reception/transmission of illegalized messages.

SRD-3.3.3.2-030

The MIL-BUS DRV's S/W shall support the ACE/MiniACE selftest.

SRD-3.3.3.2-040

The MIL-BUS DRV's S/W shall configure the ACE/MiniACE in Enhanced mode.

SRD-3.3.3.2-050

The MIL-BUS DRV's S/W shall be able to perform the ACE/MiniACE chip software reset

SRD-3.3.3.2-060

The MIL-BUS DRV's S/W shall be able to perform the 1553B Hardware reset.

3.3.3.3 ACE/MiniACE Initialization requirements

SRD-3.3.3.3-000

The MIL-BUS DRV's S/W shall initialize the ACE/MiniACE chip in compliance to the following initialization procedure in the following table.

Step#	Operation Description
1	The MIL-BUS DRV's S/W shall command a ACE/MiniACE software reset by writing 0x0001 to the Start/Stop Register.
2	The MIL-BUS DRV's S/W shall configure ACE/MiniACE RT's in Enhanced mode by writing 0x8000 to the Configuration Register #3
3	The MIL-BUS DRV's S/W shall set the Interrupt Mask Register enabling the End Of Message interrupt (TBC)
4	The MIL-BUS DRV's S/W shall load the starting location of the Stack area A into the Active area stack pointer location in RAM
5	Initialize the Active Stack Area writing 0x0000 in the word 1 of the Message Block Description of each messages
6	Initialize the Active Area Lookup Table for each transmit, receive and broadcast subaddress and initializa the Subaddress Control Word for the active area
7	The MIL-BUS DRV's S/W shall set the ACE/MiniACE Enhanced Mode Code Handling writing the bit 4 = 1 of the Configuration Register #4.
8	The MIL-BUS DRV's S/W shall initialize the Configuration Register #2 selecting the Circular Buffer mode memory management for the BC->RT messages and Double Buffering mode for RT->BC messages. The MIL-BUS DRV's S/W shall enable separate lookup table pointers for non-boradcast and broadcast receive messages
9	The MIL-BUS DRV's S/W shall set the TIME TAG RESOLUTION bits to 64microsec (TBC). The MIL-BUS DRV's S/W shall set the CLEAR TIME TAG ON SYNCHRONIZE bit = 1 for cleaning the Time Tag registrer following receipt of a synchronize without data
10	The MIL-BUS DRV's S/W shall configure the Configuration Register #3. It shall program the RT stack size field = 256Words , 64 messages. It shall enable the feature to manage the ILLEGALIZATION COMMAND It shall enable the feature to manage the ENHANCED MODE CODE HANDLING
12	The MIL-BUS DRV's S/W shall initialize the Illegalization table
13	The MIL-BUS DRV's S/W shall write in the appropriate Data block the data to be transmitted to Bus Controller
14	The MIL-BUS DRV's S/W shall configure the ACE/MiniACE as Remote Terminal writing the

	Configuration register #1 setting the bit 15 = 1 and 14=0.

Table 3-2: Initialization Procedure

3.3.3.3.1 ACE/MINIACE MEMORY MANAGEMENT REQUIREMENTS

SRD-3.3.3.3-010

The MIL-STD-1553B DRV's S/W shall insert the blocks descriptors messages (4 words for messages) in the stack Area A in DPRAM.

SRD-3.3.3.3-020

In order to allow the single buffered mode messages, the MIL-STD-1553B DRV S/W shall map in the shared memory frames of blocks descriptors messages pointing to different transmit Subaddress.

SRD-3.3.3.3-030

The MIL-STD-1553B DRV's S/W shall organize the shared memory for all transmitted Subaddress (SA) messages in single buffered mode as follow.

- *It shall set the transmit subaddress Lookup Pointer in the Lookup Pointer Table to point to the next free Data block in Data Block area.*
- *It shall configure the subaddress Control Word for the management of transmit subaddress messages in Single Messages.*

SRD-3.3.3.3-040

The MIL-STD-1553B DRV's S/W shall organize the shared memory for all received Subaddress (SA) messages in Circular Buffer Mode as follow.

- *It shall set the transmit subaddress Lookup Pointer in the Lookup Pointer Table to point to the next free Data block in Data Block area.*
- *It shall configure the subaddress Control Word for the management of receive subaddress messages in Circular Buffer Mode.*

SRD-3.3.3.3-050

The MIL-STD-1553B DRV's S/W shall size the circular buffer of each receive sub-address message to 128words

3.3.3.3.2 RT BLOCKS DESCRIPTOR MANAGEMENT

SRD-3.3.3.3-060

After a messages is received, the MIL-STD-1553B DRV's S/W shall read the received Block Status Word, Time Tag, Data Block starting Address, and Command word.

SRD-3.3.3.3-070

The MIL-STD-1553B DRV's S/W shall check the Block Status Word to validate the received messages. The MIL-STD-1553B DRV S/W shall validate the message if the Block Status Word shall contains 0x4000 after receiving the messages.

SRD-3.3.3.3-080

The MIL-STD-1553B DRV's S/W shall read the Time Tag Register indicating the message reception time.

SRD-3.3.3.3-090

The MIL-STD-1553B DRV's S/W shall reset the Time Tag register when a Synchronize message without data word is received.

SRD-3.3.3.3-100

The MIL-STD-1553B DRV's S/W shall read, by Data Block Pointer, the starting location of the Data Word block for the respective messages, in case the RT receives a non-mode code message.

SRD-3.3.3.3-110

The MIL-STD-1553B DRV's S/W shall read by Data Block Pointer the received Mode Code with data word messages.

3.3.3.3.3 RT COMMAND ILLEGALIZATION

SRD-3.3.3.3-120

The MIL-STD-1553B DRV's S/W shall allow to enable the Remote Terminal Illegalization command for all Receive, Transmit, Broadcast Subaddresses up to the word level.

3.3.3.3.4 RT INTERRUPT MANAGEMENT

SRD-3.3.3.3-130

The MIL-STD-1553B DRV's S/W shall allow to set in the Remote Terminal subaddress Control word the Interrupt management type associated to a specific subaddress.

SRD-3.3.3.3-140

The MIL-STD-1553B DRV's S/W shall allow to enable the End of Messages Interrupt management.

SRD-3.3.3.3-150

The MIL-STD-1553B DRV's S/W shall allow to enable the Circular Buffer Rollover interrupt management.

SRD-3.3.3.3-160

The MIL-STD-1553B DRV's S/W shall allow to enable Interrupts management following the reception of a specific mode code.

3.4 EEPROM DRIVERS S/W

SRD-3.5.0.0-000

The EEPROM DRIVERS S/W shall be able to write a single EEPROM memory cell. After writing the data in the EEPROM location, the EEPROM DRV's S/W shall read the same data to verify its correctness.

SRD-3.5.0.0-010

The EEPROM DRV's S/W shall allow to "delete" a single EEPROM memory cell by writing 0x0(hex) value.

SRD-3.5.0.0-020

The EEPROM DRIVERS S/W shall be able to read a single EEPROM memory cell.

SRD-3.5.0.0-030

The EEPROM DRIVERS S/W shall provide services for writing an EEPROM segment (header + data)

SRD-3.5.0.0-040

The EEPROM DRIVERS S/W shall allow to delete an EEPROM segment.

3.5 WATCHDOG DRIVERS

SRD-3.6.0.0-000

The WATCHDOG DRV S/W shall allow to program the delay time between two watchdog triggers. The allowed values of such delay time are defined in [AD 3].

SRD-3.6.0.0-010

The WATCHDOG DRV S/W shall allow to program the delay time one time only after the Power On. The WATCHDOG DRV S/W shall return a warning message in case of second access to the delay register.

SRD-3.6.0.0-020

The WATCHDOG DRV S/W shall allow to refresh the watchdog writing the value 0x5A6C (hex) in watchdog refresh register. If the programmed time expires without refreshing of the Watchdog, the CPU is reset.

4 PERFORMANCE REQUIREMENTS

4.1.1 IEEE 1355 DRIVERS

SRD-4.1.1.0-000

The max length of the transmitted packet <T_ByteSize> shall be: 16Kbytes / 6

4.1.2 MIL-STD-1553B DRIVERS

SRD-4.1.2.0-000

The MIL-STD-1553B DRV's S/W shall manage up to 10(TBC) Transmit Sub-address Frames .

SRD-4.1.2.0-010

The MIL-STD-1553B DRV's S/W shall manage up to 10(TBC) Received Sub-address Frames

SRD-4.1.2.0-020

The MIL-STD-1553B DRV's S/W shall manage Transmit Sub-address Frame composed of 16 (max) single 1553B messages.

SRD-4.1.2.0-030

The MIL-STD-1553B DRV's S/W shall manage Received Sub-address Frame composed of 4(TBC) (max) single 1553B messages.

5 HARDWARE INTERFACE REQUIREMENTS

5.1 MIL-STD-1553B DRIVERS

SRD-5.1.0.0-000

The software Interface of the MIL-STD-1553B DRIVERS to the ACE/MiniACE shall consist of 17 Internal registers for normal operation an additional 8 test register , plus 4Kx16bit of shared memory address space.

SRD-5.1.0.0-010

The address space registers and the 4Kx16bit shared memory shall be mapped starting from the CHIP SELECT 7 of the Bus Interface (BASE_ADDRESS = 0x8F000000)

SRD-5.1.0.0-020

The MIL-STD-1553B DRIVER S/W shall read/write the address space register summing the BASE_ADDRESS and the OFFSET indicated in the table following. The registers are described in detail in [RD 4] document

Address	Register	Size [bit]	Direction
0x4000	Interrupt mask register	16	read/write
0x4001	Configuration register #1	16	read/write
0x4002	Configuration register #2	16	read/write
0x4003	Start/Reset register	16	Write
0x4003	BC/RT Command Stack Pointer register	16	Read
0x4004	BC Control Word/RT Subaddress Control Word register	16	read/write
0x4005	Time Tag register	16	read/write
0x4006	Interrupt Status register	16	Read
0x4007	Configuration register #3	16	read/write
0x4008	Configuration register #4	16	read/write
0x4009	Configuration register #5	16	read/write
0x400A	Data Stack Address register	16	read/write
0x400B	BC Frame Time Remaining register	16	read/write
0x400C	BC Time Remaining to Next Message register	16	read/write
0x400D	BC Frame Time/RT Last Command/MT Trigger Word register	16	read/write
0x400E	RT Status Word register	16	read
0x400F	RT BIT Word register	16	read
0x4010	Test Mode register 0	16	
0x4011	Test Mode register 0	16	
0x4012	Test Mode register 0	16	

Address	Register	Size [bit]	Direction
0x4013	Test Mode register 0	16	
0x4014	Test Mode register 0	16	
0x4015	Test Mode register 0	16	
0x4016	Test Mode register 0	16	
0x4017	Test Mode register 0	16	
0x4018	Reserved	16	--
0x4019	Reserved	16	--
0x401A	Reserved	16	--
0x401B	Reserved	16	--
0x401C	Reserved	16	--
0x401D	Reserved	16	--
0x401E	Reserved	16	--
0x401F	Reserved	16	--

Table 5-1: Address space registers

SRD-5.1.0.0-030

The MIL-STD-1553B DRIVER S/W shall read/write the Data Shared Memory summing the BASE_ADDRESS and the OFFSET inside the range indicated in the table following.

Address	Peripheral	Size [bit]	Direction
0x0000 ÷ 0x3FFF	Buffer RAM	16	Read write

Table 5-3: Address Shared Memory

SRD-5.1.0.0-040

In the Remote Terminal configuration, the MIL-STD-1553B DRIVERS S/W shall map the shared memory as shown in the following table:

Offset Address (HEX)	Description
0000-00FF	StackA
0100	RT Command Stack Pointer A (fixed location)
0101-0103	RESERVED
0104	(not used) RT Command Stack Pointer B (fixed location)
0105-0107	RESERVED
0108-010F	Mode Code Selective Interrupt Table (fixed area)
0110-013F	Mode Code Data (fixed area)
0140-01BF	Lookup Table A (fixed area)
01C0-023F	(not used) Lookup Table B (fixed area)
0240-0247	(not used) Busy bit Lookup Table (fixed area)
0248-025F	(not used)
0260-027F	(not used) Data Block 0
0280-029F	(not used) Data Block 1
….	….
02E0-02FF	(not used) Data Block 4
0300-03FF	Command Illegalizing table (fixed area)

0400-041F	Data Block 5
0420-043F	Data Block 6
…..	…..
0FE0-0FFF	Data Block 100

Table 5-5: Memory Map in Remote Terminal configuration

SRD-5.1.0.0-050

The MIL-STD-1553B DRV S/W shall map the Lookup Table A fixed area as detailed in the following table

Area A	Description	Comment
0140 …… …… 015F	Rx Subadress 0 ….. ….. Rx Subaddress 31	Receive Lookup table pointers
0160 …… …… 17F	Tx Subaddress 0 …… …… Tx Subaddress 31	Transmit Lookup table pointers
0180 …… …… 019F	Bcst Subaddress 0 ….. ….. Bcst Subaddress 31	Broadcast Lookup table pointers
01A0 ….. ….. 01BF	Sub Address Control Word 0 ….. ….. Subaddress Control Word 31	Subaddress Control Word Table

Table 5-7: Remote Terminal Lookup Table

SRD-5.1.0.0-060

The MIL-STD-1553B DRVS S/W shall insert in the Stack Area A 64(max) MIL-1553B messages descriptor blocks. The messages descriptor block consist of four words as shown in the following table:

Number	Word	Description
1	Block Status Word	This word is initialize by MIL-STD-1553B DRV S/W and it is written by ACE/MiniACE after the message receiving. The word includes message status bit, it is read by DRV S/W to check the status of the messages.
2	Time Tag Word	This word is update by ACE/MiniACE chip during both the Start-of-Message and End-of-Message sequences. The DRV S/W read this word at the End-of-Mesage
3	Data Block Pointer or Mode Data Word	This word is used for managing the mode code or to pointer the Data Messages. The DRV S/W reads this word to access to the Data
4	Command Word	This is the Command Word received from the Bus Controller

Table 5-9: Descriptor Block words

Offset Address (HEX)	Description
0000-0003	Descriptor Block Words Location for Message #0
0004-0007	Descriptor Block Words Location for Message #1
0008-000B	Descriptor Block Words Location for Message #3
……..	……
……..	……
00FC-00FF	Descriptor Block Words Location for Message #64

Table 5-11: Stack Area A

SRD-5.1.0.0-070

The MIL-STD-1553B DRV S/W shall manage the COMMAND ILLEGALIZATION Area at the following address as shown in the table.

Offset Address	Description
0300	Broadcast/Rx, Subaddress 0 Mode Code 15-0
0301	Broadcast/Rx, Subaddress 0 Mode Code 31-16
0302	Broadcast/Rx, Subaddress 1 Word Count Code 15-0
0303	Broadcast/Rx, Subaddress 2 Word Count 31-16
……..	……..
033E	Broadcast/Rx, Subaddress 31 Mode Code 15-0
033F	Broadcast/Rx, Subaddress 31 Mode Code 31-16
0340	Broadcast/Tx, Subaddress 0 Mode Code 15-0
0341	Broadcast/Tx, Subaddress 0 Mode Code 31-16
…….	(not used) Non mode code Broadcast Transmit commands SA 1-30
037E	Broadcast/Tx, Subaddress 31 Mode Code 15-0
037F	Broadcast/Tx, Subaddress 31 Mode Code 31-16
0380	Own Address Rx Subaddress 0 Mode Code 15-0
0381	Own Address Rx Subaddress 0 Mode Code 31-16
0382	Own Address Rx Subaddress 1 Word Count 15-0
0383	Own Address Rx Subaddress 1 Word Count 31-16
……	
03BC	Own Address Rx Subaddress 30 Word Count 15-0
03BD	Own Address Rx Subaddress 30 Word Count 31-16
03BE	Own Address Rx Subaddress 31 Mode Code 15-0
03BF	Own Address Rx Subaddress 31 Mode Code 31-16
03C0	Own Address Tx Subaddress 0 Mode Code 15-0
03C1	Own Address Tx Subaddress 0 Mode Code 31-16
03C2	Own Address Tx Subaddress 1 Word Count 15-0
03C4	Own Address Tx Subaddress 1 Word Count 31-16
……	

03FC	Own Address Tx Subaddress 30 Word Count 15-0
03FD	Own Address Tx Subaddress 30 Word Count 31-16
03FE	Own Address Tx Subaddress 31 Mode Code 15-0
03FF	Own Address Tx Subaddress 31 Mode Code 31-16

Table 5-13: illegal commands ram map

SRD-5.1.0.0-080

The MIL-STD DRV S/W shall manage the mode code with data locations in shared memory area and the mode code interrupt lookup table ram map at the address defined in the tables:

Offset Address	Mode Code with Data
0110	UNDEFINED
0111	SYNCHRONIZE WITH DATA
0112	UNDEFINED
0113	UNDEFINED
0114	SELECTED TRANSMITTER SHUTDOWN
0115	OVERRIDE SELECTED TRANSMITTER SHUTDOWN
0116-011F	RESERVED (RECEIVE MODE CODES)
0120	TRANSMIT VECTOR WORD
0121	UNDEFINED
0122	UNDEFINED
0123	TRANSMIT BIT WORD
0124	UNDEFINED
0125	UNDEFINED
0126-012F	RESERVED (TRANSMIT MODE CODE)
0130	UNDEFINED BROADCAST
0131	BROADCAST SYNCHRONIZE WITH DATA
0132	UNDEFINED BROADCAST
0133	UNDEFINED BROADCAST
0134	BROADCAST SELECTED TRANSMITTER SHUTDOWN
0135	BROADCAST OVERRIDE SELECTED TRANSMITTER SHUTDOWN
0136-013F	RESERVED BROADCAST

Table 5-15: Mode Code data location

Offset Address	***Description***
0108	Receive Mode Commands 0-15 (Undefined)
0109	Receive Mode Commands 31-16 (With Data)
010A	Transmit Mode Commands 0-15 (Without data)
010B	Transmit Mode Commands 16-31 (With Data)
010C	Broadcast Receive Mode Commands 0-15 (undefined)
010D	Broadcast Receive Mode Commands 16-31 (with data)
010E	Broadcast Transmit Mode Commands 0-15 (Without data)
010F	Broadcast Transmit Mode Commands 16-31 (undefined/reserved)

Table 5-17: Mode Code Interrupt Look up table

SRD-5.1.0.0-090

The MIL-STD-1553B DRV S/W shall set to "not used" the following mapped fields in shared memory:

- *Stack Pointer B*
- *Lookup Table B*
- *Busy Bit (TBC)*

6 USER INTERFACE REQUIREMENTS

6.1 IEEE 1355 DRIVERS

SRD-6.1.0.0-000

The IEEE 1355 DRV's shall implement the routines (prototype) defined in the following table.

FUNCTION	DESCRIPTION
Smcs_ResetLink(smcs_ID,nLink)	Performs a complete reset of the link <nLink> of the <smcs_id> SMCS device.
Smcs_SetTimeOut (smcs_ID, timeout)	Set the time-out value for the SMCS; *Time out is only used if a read or write data is set to wait until completion.*
Smcs_OpenLink(smcs_ID,nLink)	Opens one of the three link of the SMCS
Smcs_CloseLink(smcs_ID,nLink)	Closes one of the three link of the SMCS
Smcs_StartLinkMaster(smcs_ID,nLink,speed)	Starts a Link as a Master at a specified Link <speed> : Links sends out NULLs tokens. A link cannot be started unless it is currently open,possible speed values are: Link_Speed_10000_Kbit 10 Mbit/sec Link_Speed_2500_Kbit 2,5 Mbit/sec Link_Speed_1250_Kbit 1,25 Mbit/sec
Smcs_StartLinkSlave(smcs_ID,nLink,speed)	Starts a Link as a Slave at a specified Link speed: This function first wait until it receives a NULL token from the corresponding Master Link. If the NULL is not received within the timeout period then it returns FALSE. Possible Link Speed values are: Link_Speed_10000_Kbit 10 Mbit/sec Link_Speed_2500_Kbit 2,5 Mbit/sec Link_Speed_1250_Kbit 1,25 Mbit/sec If the NULL is received within the time-out period, then the link is started.
Smcs_StopLink(smcs_ID,nLink)	Stops a currently running Link, This command stops a link transmitting which also causes a disconnect error in the other

	end of the link,
GetLinkStatus(smcs_ID, nLink)	Gets the status register for that Link (CHx_DSM_STAR)
Smcs_WriteLink(smcs_ID,nLink, pBuffer,ByteSize,bWait)	Transmits <ByteSize> fixed number of Bytes starting at address <pBuffer> over the link <nLink> of the <smcs_ID> device . If bWait is true the functions returns immediately, other way it returns when the transfer is completed or an error is detected. <ByteSize> determines the length of the packet transmitted ; it must be <= 16Kbytes / 6
Smcs_Read Link (smcs_ID,nLink, pBuffer,ByteSize,bWait)	Receives , over the link <nLink> of the <smcs_ID> device , <ByteSize> fixed number of Bytes and writes them in a user buffer starting at address <pBuffer>. If bWait is true the functions returns immediately, other way it returns when the transfer is completed or an error is detected. <ByteSize> determines the length of the packet received ; it must be <= 16Kbytes / 6
Smcs_GetReadStatus(smcs_ID,nLink,)	Returns the current transmit status for the specified link <nLink> Possible values: Transfer_not-started, Transfer_started, Transfer_Done, Transefr_error_diconnect, Transfer_error Parity, Transfer_error_timeout, Transfer_error_link_not _started, etc.)
Smcs_GetWriteStatus(smcs_ID,nLink,)	Returns the current receive status for the specified link <nLink> Possible values: Transfer_not-started, Transfer_started, Transfer_Done, Transfer_error_diconnect, Transfer_error Parity,, Transfer_error_timeout, Transfer_error_link_not _started, etcc.)
Smcs_ReadPackets(smcs_ID, nLink, pBuffer, ByteSize, NumPackets, bWait)	Reads a defined <NumPackets> over the link <nLink> of the <smcs_ID> device, and assigns the number of byte <ByteSize>. The data are stored in a buffer pointed by pBuffer. If <bwait> is true the functions return immediately, other way it returns when the transfer is completed or an error is detected.
Smcs_GetLastReadSize(smcs_ID, nLink)	Returns the number of received data over the link <nLink> of the <smcs_ID> device.
Smcs_GetLastWriteSize(smcs_ID,nLink)	Returns the number of transmitted data over the Link <nLink> of the <smcs_ID> device.
Smcs_GetLastPacketsNum(smcs_ID, nLink)	Returns the number of received packet over the link <nLink> of the <smcs_ID> device.
Low Level Functions	
Smcs_WriteToBoardMemory(smcs_ID, startaddress, bytesize, pbuffer)	Writes <ByteSize> data, from the buffer pointed by <pbuffer>, in the <smcs_ID> DPRAM starting from <startaddress> of the <smcs_ID> device
Smcs_ReadFromBoardMemory(smcs_ID, startaddress, bytesize, pbuffer)	Reads <bytesize> data from <smcs_ID> DPRAM starting from <startaddress> and writes them in the buffer pointed by <pbuffer>

Smcs_WriteRegister(smcs_ID, offset, value)	Writes <value> in the <smcs_ID> with displacement <offset>, in a generic register
Smcs_ReadRegister(smcs_ID, offset)	Reads and returns the value contained in a register of the <smcs_ID> device with displacement <offset>

6.2 MIL-STD-1553B DRIVERS

SRD-6.2.0.0-000

The MIL-STD-1553B DRV's shall implement the routines (prototype) defined in the following table.

FUNCTION	DESCRIPTION
SET UP ROUTINES	
RTOpen(Conf)	This sets up all global structures (Conf) used by the software driver and presets them to default values. The designated configuration structure is then set active and the routine returns a pointer to Conf structure.
RTClose(Conf)	Called at end of ACE operation. This routine clears up all global structures used by the Drivers. It deactivates the Conf "context".
RTAddressRead(Conf, Rtaddr)	It reads the RT address for the ACE Chip and returns it into Rtaddr.
RTsacw2word(Conf, Sacw)	It converts the information contained into subaddress control word (Sacw) data structure fields to a single 16bit word.
RTword2sacw (Conf, word)	It converts the information contained into a single 16bit word to a subaddress control word data structure fields.
RTDefSA(Conf, SubAddr, Sacw)	This routine is used to configure the memory management and interrupt schemes for the respective subaddress for transmit, receive and broadcast commands. Valid memory management schemes are SINGLE_MESSAGE and CIRCULAR_128. End of message interrupts may be generated by enabling the selected message type (ex: Sacw->RxEoMInt=YES). Circular buffer interrupts may be generated by enabling the selected message type (ex: Sacw->TxCircBuffInt=YES) provided.
RTConfigureMemory(Conf)	This functions will configure the DPRAM memory interfacing the ACE chip as defined by the user.
RTRun(Conf)	This function will clear the descriptor stack, it will reset the stack pointer and it will put the ACE device in running.
RTDefMsgLegal(Conf, MessType, subaddr, wc)	Sets the message legality to legal, based on the message subaddress, word count, and transmit/receive bit
RTDefMsgIllegal(Conf, MessType, subaddr, wc)	Sets the message legality to illegal, based on the message subaddress, word count and transmit/receive bit
RTIrqMsgSaEnable(Conf, sa, t_r, selection)	This routine will enable selected interrupts (EndOfMessage, Circular Buffer RollOver, SubAddress)

RTIrqMsgSaDisable(Conf, sa, t_r, selection)	This routine will disable selected Interrupt (EndOfMessage, Circular Buffer RollOver, SubAddress)
RTModeIrqEnable(Conf, broadcast, t_r, data, map);	This routine will enable selected mode code interrupts.
RTStop(Conf)	It stops the Remote Terminal
RTSelfTest(Conf)	This routine will perform a self Test of the RT internal register Stack Area and Lookup tables.
RTModeIrqDisable(Conf, broadcast, t_r, data, map)	This routine will disable selected mode code interrupts
Reset(Conf)	It performs a Software Reset of the ACE chip. It resets all the ACE register and the DPRAM area.
DATA LINK LAYER ROUTINES	
RTReadDescMsg(Conf, MessageNum, Message)	This routine will return a structure which contains the four word descriptor stack entry (block status word, time tag, data block pointer, and command word).
BuRTWriteEnhMCData(Conf, addr, data)	The ACE has separate data locations for mode codes. These data locations (from 110h-13fh) can be written by this routine.
RTReadEnhMCData(Conf, addr)	The ACE has separate data locations for mode codes. These data locations (from 110h-13fh) can be written by this routine
TRANSFER LAYER ROUTINES	
RTCreateFrame(Conf, FrameID)	Create a Frame pointer to manage the Frame of standard message packets.
RTAddMessageToFrame(Conf, FrameID, sa, t_r, word_count)	It adds Standard Message to the selected Frame. This function allows to create homogeneous frame (for Instance TM frame, or TC frame). NOTE: The functions will allow to add to the same frame, standard messages with different subaddress only. The functions will not allow to insert in the frame MODE CODE messages.
RTFrameRead(Conf, FrameID, buffer)	It reads the received messages of the frame and it writes them into the buffer
RTFrameWrite(Conf, FrameID, buffer)	It write the buffer data into the messages of the Frame.
RTReadSingleMessage(Conf, buffer, sa, WordCount)	It reads the single standard message and it puts the read data in the buffer.
RTWriteSingleMessage(Conf, buffer, sa, WordCount)	It write the data in the buffer into the single standard message.

6.3 EEPROM DRIVERS

SRD-6.3.0.0-000

The EEPROM DRV's shall implement the routines (prototype) defined in the following table.

FUNCTION	DESCRIPTION
EEPROMWriteCell(Address, Data)	It writes a single EEPROM memory cell. It reads the same data to verify if the written data matches with the read data. it returns an error code if a mismatch is detected or 0 if the data is correct.
EEPROMClearCell(Address)	It clears a single EEPROM memory cell by writing 0x00000000. It returns an error code if the memory cell is not cleared.
EEPROMReadCell(Address)	It reads a single EEPROM memory cell and it returns the read value
EEPROMWriteSegment(NumberOfSegment, Header(structure), Data(buffer))	It writes an EEPROM segment including header and Data. The Users has to specify the Number of Segment, the Header parameters and the Data. The function writes the header and Data in the EEPROM and it computes and verifies the specified EEPROM FCS in the Header parameter with the computed FCS. In case of failure the function returns an error message.
EEPROMDeleteSegment(NumberOfSegment)	It deletes the EEPROM segment at NumberOfSegment address.

6.4 WATCHDOG DRIVERS

SRD-6.4.0.0-000

The WATCHDOG DRV's shall implement the routines (prototype) defined in the following table.

FUNCTION	DESCRIPTION
WDSet(value)	It sets the delay time between two Watchdog triggers. The possible value are defined in [AD2]
WDRefresh(void)	It refresh the watchdog

7 RESOURCE REQUIREMENTS

SRD-7.0.0.0-000

The following functions of the DPU BSW shall be stored in the 32Kbyte DPU PROM:

- *POWER ON Procedure*
- *MIL-STD-1553B Drivers*
- *WATCHDOG Drivers*

SRD-7.0.0.0-010

The following functions of the DPU BSW shall be part of the APF and stored in the 1Megabyte DPU EEPROM:

- *IEEE 1355 Drivers*
- *MIL-STD-1553B Drivers*
- *EEPROM Drivers*
- *WATCHDOG Drivers*

8 ACCEPTANCE TESTING REQUIREMENTS

SRD-8.0.0.0-000

The DHPU BSW will be accepted integrated with the DPU HW by a sequence of "black box" functional test procedures (covering all functional requirements stated in this document) to be approved by the Customer.

9 DOCUMENTATION REQUIREMENTS

SRD-9.0.0.0-000

The following documents will be delivered:

- *SW Req Doc*
- *Architectural Design Doc*
- *Test Procedures*
- *Test Reports*

10 TRACEABILITY MATRIX

SRD/Requirements	Documents/Requirements
SRD-3.1.1.0-000	*[AD-10]-PROM-ON-01*
SRD-3.1.1.0-010	*[AD-10]-PROM-ON-02*
SRD-3.1.1.0-020	*[AD-9]-OBS-CUR-ON-04*
SRD-3.1.1.0-030	*[AD-9]-OBS-CUR-ON-04*
SRD-3.1.1.0-040	*[AD-10]-PROM-ON-01*
SRD-3.1.1.0-050	*[AD-10]-PROM-ON-02*
SRD-3.1.1.0-060	*[AD-10]-PROM-ON-03*
SRD-3.1.1.0-070	*[AD-10]-PROM-ON-04*
SRD-3.1.1.0-080	*[AD-10]-PROM-ON-04.1*
SRD-3.1.1.0-090	*[AD-10]-PROM-ON-04.2*
SRD-3.1.1.0-100	*[AD-10]-PROM-ON-04.3*
SRD-3.1.1.0-110	*[AD-10]-PROM-ON-04.4*
SRD-3.1.1.0-120	*[AD-10]-PROM-ON-04.5*
SRD-3.1.1.0-130	*[AD-10]-PROM-ON-04.6*
SRD-3.1.1.0-140	*[AD-10]-PROM-ON-04.7*
SRD-3.1.1.0-150	*[AD-10]-PROM-ON-05*
SRD-3.1.1.0-160	*[AD-10]-PROM-ON-06*
SRD-3.1.1.0-170	*[AD-10]-PROM-ON-06.1*
SRD-3.1.1.0-180	*[AD-10]-PROM-ON-06.2*
SRD-3.1.1.0-190	*[AD-10]-PROM-ON-06.3*
SRD-3.1.1.0-200	*[AD-10]-PROM-ON-06.4*
SRD-3.1.1.0-210	*[AD-10]-PROM-ON-06.5*
SRD-3.1.1.0-220	*[AD-10]-PROM-ON-06.6*
SRD-3.1.1.0-230	*[AD-10]-PROM-ON-07*
SRD-3.1.1.0-240	*[AD-10]-PROM-ON-08*
SRD-3.1.1.0-250	*[AD-10]-PROM-ON-09*
SRD-3.1.1.0-260	*[AD-10]-PROM-ON-09.1*
SRD-3.1.1.0-270	*[AD-10]-PROM-ON-09.2*
SRD-3.1.1.0-280	*[AD-10]-PROM-ON-09.3*
SRD-3.1.1.0-290	*[AD-10]-PROM-ON-09.4*
SRD-3.1.1.0-300	*[AD-10]-PROM-ON-09.5*
SRD-3.1.1.0-310	*[AD-10]-PROM-ON-09.6*
SRD-3.1.1.0-320	*[AD-10]-PROM-ON-10*
SRD-3.1.1.0-330	*[AD-10]-PROM-ON-11*
SRD-3.1.1.0-340	*[AD-10]-PROM-ON-12*
SRD-3.1.1.0-350	*[AD-10]-PROM-ON-13*
SRD-3.1.1.0-360	*[AD-10]-PROM-ON-13.1*
SRD-3.1.1.0-370	*[AD-10]-PROM-ON-14*
SRD-3.1.2.0-000	*[AD-10]-PROM-ON-12*
SRD-3.1.2.0-005	*[AD-10]-PROM-ON-13*
SRD-3.1.2.0-010	*[AD-10]-PROM-ON-13.1*
SRD-3.1.2.0-020	*[AD-10]-PROM-ON-13.1*
SRD-3.1.3.0-000	*[AD-10]-PROM-ON-13.1*
SRD-3.1.4.0-000	*[AD-10]-PROM-ON-02*
SRD-3.1.4.0-010	*[AD-10]-PROM-ON-04.6*
SRD-3.1.4.0-020	*[AD-10]-PROM-ON-04.6*
SRD-3.1.4.0-030	*[AD-10]-PROM-ON-04.6*

SRD-3.1.4.0-040	***[AD-10]-PROM-ON-04.6***
SRD-3.1.5.0-000	***[AD-10]-PROM-ON-02***
SRD-3.1.5.0-010	***[AD-10]-PROM-ON-04.6***
SRD-3.1.5.0-020	***[AD-10]-PROM_ON-04.6***
SRD-3.1.5.0-030	***[AD-10]-PROM-ON-04.6***
SRD-3.1.5.0-040	***[AD-10]-PROM-ON-04.6***
SRD-3.1.6.0-000	***[AD-10]-PROM-ON-02***
SRD-3.1.6.0-010	***[AD-10]-PROM-ON-04.6***
SRD-3.1.6.0-020	***[AD-10]-PROM-ON-04.6***
SRD-3.1.6.0-030	***[AD-10]-PROM-ON-04.6***
SRD-3.1.6.0-040	***[AD-10]-PROM_ON-04.6***
SRD-3.1.6.0-050	***[AD-10]-PROM-ON-04.6***
SRD-3.1.6.0-060	***[AD-10]-PROM-ON-02***
SRD-3.1.6.0-070	***[AD-10]-PROM-ON-02***
SRD-3.1.7.0-000	***[AD-10]-PROM-ON-13.1***
SRD-3.1.7.0-010	***[AD-10]-PROM-ON-09.5***
SRD-3.1.7.0-020	***[AD-10]-PROM-ON-13.1***
SRD-3.1.7.0-030	***[AD-10]-PROM-ON-13.1***
SRD-3.1.7.0-040	***[AD-10]-PROM-ON-13.1***
SRD-3.1.8.0-000	***[AD-10]-PROM-ON-13***
SRD-3.1.8.0-010	***[AD-10]-PROM-ON-09.5***
SRD-3.1.8.0-020	***[AD-10]-PROM-ON-13.1***
SRD-3.1.8.0-030	***[AD-10]-PROM-ON-13.1***
SRD-3.1.8.0-040	***[AD-10]-PROM-ON-13.1***
SRD-3.2.0.0-000	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.2.0.0-010	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.2.0.0-020	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.2.0.0-030	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.2.0.0-040	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.2.0.0-050	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.2.0.0-060	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.2.0.0-070	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.2.0.0-080	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.2.0.0-090	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.2.0.0-100	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.2.0.0-110	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.2.0.0-120	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.2.0.0-130	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.2.0.0-140	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.2.0.0-150	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.2.0.0-160	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.2.0.0-170	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.2.0.0-180	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.2.0.0-190	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.2.0.0-200	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.2.0.0-210	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-3.3.2.0-000	***[AD-7]-Appendix9 section 4.1.1***
SRD-3.3.2.0-010	***[AD-7]-Appendix9 section 4.1.1***
SRD-3.3.3.1-000	***[AD3]-section 7.1***
SRD-3.3.3.1-010	***[AD-7]-Appendix9 section 3.2.4***
SRD-3.3.3.1-020	***[AD-7]-Appendix9 section 3.2.4***
SRD-3.3.3.1-030	***[AD-7]-Appendix9 section 3.1.2***

SRD-3.3.3.1-040	***[AD-7]-Appendix9 section 3.1***
SRD-3.3.3.1-050	***[AD-7]-Appendix9 section 3.1***
SRD-3.3.3.1-060	***[AD-7]-Appendix9 section 3.1***
SRD-3.3.3.2-000	***[AD3]-section 7.1***
SRD-3.3.3.2-010	***[AD3]-section 7.1***
SRD-3.3.3.2-020	***[AD3]-section 7.1***
SRD-3.3.3.2-030	***[AD3]-section 7.1***
SRD-3.3.3.2-040	***[AD3]-section 7.1***
SRD-3.3.3.2-050	***[AD3]-section 7.1***
SRD-3.3.3.2-060	***[AD3]-section 7.1***
SRD-3.3.3.3-000	***[AD3]-section 7.1***
SRD-3.3.3.3-010	***[AD3]-section 7.1***
SRD-3.3.3.3-020	***[AD3]-section 7.1***
SRD-3.3.3.3-030	***[AD3]-section 7.1***
SRD-3.3.3.3-040	***[AD3]-section 7.1***
SRD-3.3.3.3-050	***[AD3]-section 7.1***
SRD-3.3.3.3-060	***[AD3]-section 7.1***
SRD-3.3.3.3-070	***[AD3]-section 7.1***
SRD-3.3.3.3-080	***[AD3]-section 7.1***
SRD-3.3.3.3-090	***[AD3]-section 7.1***
SRD-3.3.3.3-100	***[AD3]-section 7.1***
SRD-3.3.3.3-110	***[AD3]-section 7.1***
SRD-3.3.3.3-120	***[AD3]-section 7.1***
SRD-3.3.3.3-130	***[AD3]-section 7.1***
SRD-3.3.3.3-140	***[AD3]-section 7.1***
SRD-3.3.3.3-150	***[AD3]-section 7.1***
SRD-3.3.3.3-160	***[AD3]-section 7.1***
SRD-3.5.0.0-000	***[AD2]-section 7.5***
SRD-3.5.0.0-010	***[AD2]-section 7.5***
SRD-3.5.0.0-020	***[AD2]-section 7.5***
SRD-3.5.0.0-030	***[AD2]-section 7.5***
SRD-3.5.0.0-040	***[AD2]-section 7.4***
SRD-3.6.0.0-000	***[AD2]-section 7.4***
SRD-3.6.0.0-010	***[AD2]-section 7.4***
SRD-3.6.0.0-020	***[AD2]-section 7.4***
SRD-4.1.1.0-000	***[AD-2]- section 7.1, 7.3, 7.8***
SRD-4.1.2.0-000	***[AD-7]-Appendix9 section 4.1.1***
SRD-4.1.2.0-010	***[AD-7]-Appendix9 section 4.1.1***
SRD-4.1.2.0-020	***[AD-7]-Appendix9 section 4.1.1***
SRD-4.1.2.0-030	***[AD-7]-Appendix9 section 4.1.1***
SRD-5.1.0.0-000	***[AD3]-section 7.1***
SRD-5.1.0.0-010	***[AD3]-section 7.1***
SRD-5.1.0.0-020	***[AD3]-section 7.1***
SRD-5.1.0.0-030	***[AD3]-section 7.1***
SRD-5.1.0.0-040	***[AD3]-section 7.1***
SRD-5.1.0.0-050	***[AD3]-section 7.1***
SRD-5.1.0.0-060	***[AD3]-section 7.1***
SRD-5.1.0.0-070	***[AD3]-section 7.1***
SRD-5.1.0.0-080	***[AD3]-section 7.1***
SRD-5.1.0.0-090	***[AD3]-section 7.1***
SRD-6.1.0.0-000	***NTBT***
SRD-6.2.0.0-000	***NTBT***

SRD-6.3.0.0-000	***NTBT***
SRD-6.4.0.0-000	***NTBT***
SRD-7.0.0.0-000	***[AD2]-section 4***
SRD-7.0.0.0-010	***[AD2]-section 4***
SRD-8.0.0.0-000	***[AD5]***
SRD-9.0.0.0-000	***[AD5]***

11 VERIFICATION MATRIX

Requirements	Verification Method
SRD-3.1.1.0-000	***SEL-T***
SRD-3.1.1.0-010	***SEL-T***
SRD-3.1.1.0-020	***SEL-T***
SRD-3.1.1.0-030	***SEL-T***
SRD-3.1.1.0-040	***SEL-R***
SRD-3.1.1.0-050	***SEL-T***
SRD-3.1.1.0-060	***SEL-T***
SRD-3.1.1.0-070	***SEL-T***
SRD-3.1.1.0-080	***SEL-T***
SRD-3.1.1.0-090	***SEL-T***
SRD-3.1.1.0-100	***SEL-T***
SRD-3.1.1.0-110	***SEL-T***
SRD-3.1.1.0-120	***SEL-T***
SRD-3.1.1.0-130	***SEL-T***
SRD-3.1.1.0-140	***SEL-T***
SRD-3.1.1.0-150	***SEL-T***
SRD-3.1.1.0-160	***SEL-T***
SRD-3.1.1.0-170	***SEL-T***
SRD-3.1.1.0-180	***SEL-T***
SRD-3.1.1.0-190	***SEL-T***
SRD-3.1.1.0-200	***SEL-T***
SRD-3.1.1.0-210	***SEL-T***
SRD-3.1.1.0-220	***SEL-T***
SRD-3.1.1.0-230	***SEL-T***
SRD-3.1.1.0-240	***SEL-R***
SRD-3.1.1.0-250	***SEL-T***
SRD-3.1.1.0-260	***SEL-T***
SRD-3.1.1.0-270	***SEL-T***
SRD-3.1.1.0-280	***SEL-T***
SRD-3.1.1.0-290	***SEL-T***
SRD-3.1.1.0-300	***SEL-T***
SRD-3.1.1.0-310	***SEL-T***
SRD-3.1.1.0-320	***SEL-T***
SRD-3.1.1.0-330	***SEL-R***
SRD-3.1.1.0-340	***SEL-R***
SRD-3.1.1.0-350	***SEL-T***
SRD-3.1.1.0-360	***SEL-T***
SRD-3.1.1.0-370	***SEL-T***
SRD-3.1.2.0-000	***SEL-R***
SRD-3.1.2.0-005	***SEL-R***
SRD-3.1.2.0-010	***SEL-R***
SRD-3.1.2.0-020	***SEL-R***
SRD-3.1.3.0-000	***SEL-R***
SRD-3.1.4.0-000	***SEL-T***
SRD-3.1.4.0-010	***SEL-T***
SRD-3.1.4.0-020	***SEL-T***
SRD-3.1.4.0-030	***SEL-T***

SRD-3.1.4.0-040	***SEL-T***
SRD-3.1.5.0-000	***SEL-T***
SRD-3.1.5.0-010	***SEL-T***
SRD-3.1.5.0-020	***SEL-T***
SRD-3.1.5.0-030	***SEL-T***
SRD-3.1.5.0-040	***SEL-T***
SRD-3.1.6.0-000	***SEL-T***
SRD-3.1.6.0-010	***SEL-T***
SRD-3.1.6.0-020	***SEL-T***
SRD-3.1.6.0-030	***SEL-T***
SRD-3.1.6.0-040	***SEL-T***
SRD-3.1.6.0-050	***SEL-T***
SRD-3.1.6.0-060	***SEL-T***
SRD-3.1.6.0-070	***SEL-T***
SRD-3.1.7.0-000	***SEL-T***
SRD-3.1.7.0-010	***SEL-T***
SRD-3.1.7.0-020	***SEL-T***
SRD-3.1.7.0-030	***SEL-T***
SRD-3.1.7.0-040	***SEL-T***
SRD-3.1.8.0-000	***SEL-T***
SRD-3.1.8.0-010	***SEL-T***
SRD-3.1.8.0-020	***SEL-T***
SRD-3.1.8.0-030	***SEL-T***
SRD-3.1.8.0-040	***SEL-T***
SRD-3.2.0.0-000	***SEL-T***
SRD-3.2.0.0-010	***SEL-T***
SRD-3.2.0.0-020	***SEL-T***
SRD-3.2.0.0-030	***SEL-T***
SRD-3.2.0.0-040	***SEL-T***
SRD-3.2.0.0-050	***SEL-T***
SRD-3.2.0.0-060	***SEL-T***
SRD-3.2.0.0-070	***SEL-T***
SRD-3.2.0.0-080	***SEL-T***
SRD-3.2.0.0-090	***SEL-T***
SRD-3.2.0.0-100	***SEL-T***
SRD-3.2.0.0-110	***SEL-T***
SRD-3.2.0.0-120	***SEL-T***
SRD-3.2.0.0-130	***SEL-T***
SRD-3.2.0.0-140	***SEL-T***
SRD-3.2.0.0-150	***SEL-T***
SRD-3.2.0.0-160	***SEL-T***
SRD-3.2.0.0-170	***SEL-T***
SRD-3.2.0.0-180	***SEL-T***
SRD-3.2.0.0-190	***SEL-T***
SRD-3.2.0.0-200	***SEL-T***
SRD-3.2.0.0-210	***SEL-T***
SRD-3.3.2.0-000	***SEL-T***
SRD-3.3.2.0-010	***SEL-T***
SRD-3.3.3.1-000	***SEL-T***
SRD-3.3.3.1-010	***SEL-T***
SRD-3.3.3.1-020	***SEL-T***
SRD-3.3.3.1-030	***SEL-T***

SRD-3.3.3.1-040	***SEL-T***
SRD-3.3.3.1-050	***SEL-T***
SRD-3.3.3.1-060	***SEL-T***
SRD-3.3.3.2-000	***SEL-T***
SRD-3.3.3.2-010	***SEL-T***
SRD-3.3.3.2-020	***SEL-T***
SRD-3.3.3.2-030	***SEL-T***
SRD-3.3.3.2-040	***SEL-T***
SRD-3.3.3.2-050	***SEL-T***
SRD-3.3.3.2-060	***SEL-T***
SRD-3.3.3.3-000	***SEL-T***
SRD-3.3.3.3-010	***SEL-T***
SRD-3.3.3.3-020	***SEL-T***
SRD-3.3.3.3-030	***SEL-T***
SRD-3.3.3.3-040	***SEL-T***
SRD-3.3.3.3-050	***SEL-T***
SRD-3.3.3.3-060	***SEL-T***
SRD-3.3.3.3-070	***SEL-T***
SRD-3.3.3.3-080	***SEL-T***
SRD-3.3.3.3-090	***SEL-T***
SRD-3.3.3.3-100	***SEL-T***
SRD-3.3.3.3-110	***SEL-T***
SRD-3.3.3.3-120	***SEL-T***
SRD-3.3.3.3-130	***SEL-T***
SRD-3.3.3.3-140	***SEL-T***
SRD-3.3.3.3-150	***SEL-T***
SRD-3.3.3.3-160	***SEL-T***
SRD-3.4.0.0-000	***SEL-T***
SRD-3.4.0.0-010	***SEL-T***
SRD-3.4.0.0-020	***SEL-T***
SRD-3.4.0.0-030	***SEL-T***
SRD-3.5.0.0-000	***SEL-T***
SRD-3.5.0.0-010	***SEL-T***
SRD-3.5.0.0-020	***SEL-T***
SRD-3.5.0.0-030	***SEL-T***
SRD-3.5.0.0-040	***SEL-T***
SRD-3.6.0.0-000	***SEL-T***
SRD-3.6.0.0-010	***SEL-T***
SRD-3.6.0.0-020	***SEL-T***
SRD-4.1.1.0-000	***SEL-R***
SRD-4.1.2.0-000	***SEL-R***
SRD-4.1.2.0-010	***SEL-R***
SRD-4.1.2.0-020	***SEL-R***
SRD-4.1.2.0-030	***SEL-R***
SRD-5.1.0.0-000	***SEL-R***
SRD-5.1.0.0-010	***SEL-R***
SRD-5.1.0.0-020	***SEL-R***
SRD-5.1.0.0-030	***SEL-R***
SRD-5.1.0.0-040	***SEL-R***
SRD-5.1.0.0-050	***SEL-R***
SRD-5.1.0.0-060	***SEL-R***
SRD-5.1.0.0-070	***SEL-R***

SRD-5.1.0.0-080	***SEL-R***
SRD-5.1.0.0-090	***SEL-R***
SRD-5.2.0.0-000	***SEL-R***
SRD-5.2.0.0-010	***SEL-R***
SRD-5.3.0.0-000	***SEL-R***
SRD-5.3.0.0-010	***SEL-R***
SRD-5.3.0.0-020	***SEL-R***
SRD-6.1.0.0-000	***SEL-R***
SRD-6.2.0.0-000	***SEL-R***
SRD-6.3.0.0-000	***SEL-R***
SRD-6.4.0.0-000	***SEL-R***
SRD-7.0.0.0-000	***SEL-R***
SRD-7.0.0.0-010	***SEL-R***
SRD-8.0.0.0-000	***SEL-R***
SRD-9.0.0.0-000	***SEL-R***