SPIRE-ESA-COM-001495

From: Stefan.Thuerey@esa.int
Sent: 14 January 2003 16:04
To: ohb@mpe.mpg.de; Serge.Valera@esa.int; Bryan.Melton@esa.int; K.J.King@rl.ac.uk; L.Dubbeldam@sron.nl; A.R.W.de.Jonge@sron.rug.nl
Cc: David.Verrier@esa.int; Frederick.Wechsler@esa.int; John.Dodsworth@esa.int; Stefan.Thuerey@esa.int; Johannes.Riedinger@esa.int; Kevin.Galloway@esa.int; jhl@iac.es; molinari@ifsi.rm.cnr.it; beney@lal.in2p3.fr; fgr@ll.iac.es; D.J.Parker@rl.ac.uk; J.A.Long@rl.ac.uk; J.Payne@rl.ac.uk; bart@ster.kuleuven.ac.be; butler@tesre.bo.cnr.it; erw@mpe.mpg.de; fgb@mpe.mpg.de; ohb@mpe.mpg.de; Thomas.Passvogel@esa.int; Gerald.Crone@esa.int; Astrid.Heske@esa.int; Andrei.Oganessian@esa.int; Yves.Bordes@esa.int; Farid.Guettache@esa.int; Goeran.Pilbratt@esa.int
Subject: Re: Herschel: 1553 I/F test plan for instruments

Mac Word 3.0

Dear colleagues,
The note below contains a very short, however correct, assessment of the suitability of two DPU/ICU documents to support adequate testing of instrument-to-spacecraft data bus interfaces. Especially almost all requirements according to Appendix 9 of the PS-ICD (the SDB-Protocol) are seemingly not covered in these "test procedures".

Moreover, already in October 2001 we have commented on a draft 0.1 of CNR.IFSI.2001TR04 that it is no acceptable that, according to para. 1.2 of that document, many basic electrical and functional tests are seemingly not carried out. Therefore, as far as currently visible to me, the data I/F of Herschel instruments may be only poorly/inadequately tested on instrument level. Nevertheless, with corrective actions early enough, this point should be resolvable.

What concerns ESA's SDBP Testbed it can play a supporting role for protocol verification and functional testing of the data I/F, however it is not foreseen to use it for the verification of electrical parameters, or for any nominal testing of Herschel/Plank interfaces. (This needs to be done together with our industrial Prime)

In order to provide you with an updated view of the architecture and capabilities of the SDBP Testbed find attached the first draft of the description. A second version is promised to follow within a few days with more supplementary information.
(See attached file: TOS-ESD SDBP_TestBesd_Overview.doc)

Best regards,
Stefan Thürey

Herschel / Planck Project, European Space Agency
Avionics and RF Systems Engineer
tel +31-71-565 5330, fax +31-71-565 5244
----- Forwarded by Stefan Thuerey/estec/ESA on 14.01.03 16:13 -----
|--------+----------------------->
	Andrei
	Oganessian
	13.01.03 16:46
--------+----------------------->

>---|
| |
| To: Stefan Thuerey/estec/ESA@ESA |
| cc: Yves Bordes/estec/ESA@ESA |
| Subject: Re: Herschel: 1553 I/F test plan for instruments |
| (Document link: Stefan Thuerey) |
>---|

Dear Stefan,

Bellow are my comments on the two documents submitted earlier by you.

1. Both documents:
     - DPU/ICU Spacecraft Interface Test Plan (CNR.IFSI.2001TR04)
     - DPU onboard software validation and verification Plan/Acceptance Test Plan (PACS-CR-PL-012)
do not cover Herschel/Planck Transfer layer Satellite Data Bus Protocol (SDPB) verification aspects.

2. Avionics Lab objective is to evaluate and verify the SDBP for Hershel/Planck onboard communication system. The input test data for such verificaton starts at Transfer layer. Although some test cases with such input data may cover parts of Instrument level Interface testing, specified in the first document ("DPU/ICU Spacecraft Interface Test Plan"), the standard Instrument level interface testing should be performed by Instrument provider. Avionics Lab Testbed is not needed for that purpose as requested in the same document.

Kind regards,

Andrei Oganessian

==========
Software Engineer

Serco FM BV c/o
Data Handling Computers Section (TOS-ESD)
ESTEC, European Space Agency
Keplerlaan 1, P.O. Box 299, Noordwijk
2200 AG, The Netherlands

Tel: (31) 71 565 3438
Fax: (31) 71 565 4295
E-mail: Andrei.Oganessian@esa.int
==========

OVERVIEW

Of The TESTBED For The SDBP PROTOCOL

HERSCHEL-PLANCK PROJECT

DRAFT

14-JANUARY 2003 ESTEC/ESA TOS-ESD

Introduction

The Mil-1553 test-bed in TOS-ESD Avionics Lab/Space data Systems provides an infrastructure for prototyping, simulation and verification of on-board bus protocols.

The complete system includes hardware and software components; various processor boards, I/O boards, harness and software tools.

For the Herschel-Planck project the testbed can be configured and set up to run the SDBP bus protocol where the CDMS is seen as bus controller and instruments / on board units seen as simulated remote terminals. The data traffic on the bus is monitored and checked for different scenarios.

Any Instrument or a representative unit, engineering or flight model, can swap any simulated RT.

I – System Description

I - a - *Objective*

The Mil-1553 test-bed for the Herschel-Planck project is a tool with the aim to provide as much as possible a realistic environment of the data traffic on the bus MIL-STD1553 with the possibility, using prototyping and simulation, to analyze and provide information about the protocol, assess performance and protocol correctness, verify and validate the protocol. The data load will be generated from both statistical and deterministic generators(the latest TBC).

I - b - *Bus Controller*

For the purpose of verification the simulated Bus Controller can be implemented in two ways, each implementation one will use a specific hardware and software:

- Cross platform implementation, in this case a commercial Pentium based board and mounted on distinct PC will be used. The candidate for Mil-Std-1553 BC simulation is a commercial board which is able to emulate the BC (see figure 1).

- Target implementation using an ERC32-based board. The candidate will be either the SAAB board or the Tharsys board; both are based on the cPCI bus (see figure 2).

The latest configuration is much closer to the real model.

I - b - *Remote terminals*

The implementation of the simulated RT at SDBP level can also be done with two possibilities:

- Cross platform implementation: a commercial Pentium based board and mounted on distinct PC. The board dedicated to simulate the RT is also a commercial board that is able to simulate all the RTs simultaneously (see figure 1).
- Plug of a model of the instrument in the Mil-1553 bus and to simulate the remaining RT (if needed) using COTS (commercial) boards (see figure 1/figure2).

I – c – Bus analysis tools

One of the available channels of the I/O boards implementing the Mil-1553 standard, will be used to run the Bus analyzer tools.

The figure below presents the two versions of the testbed configuration:

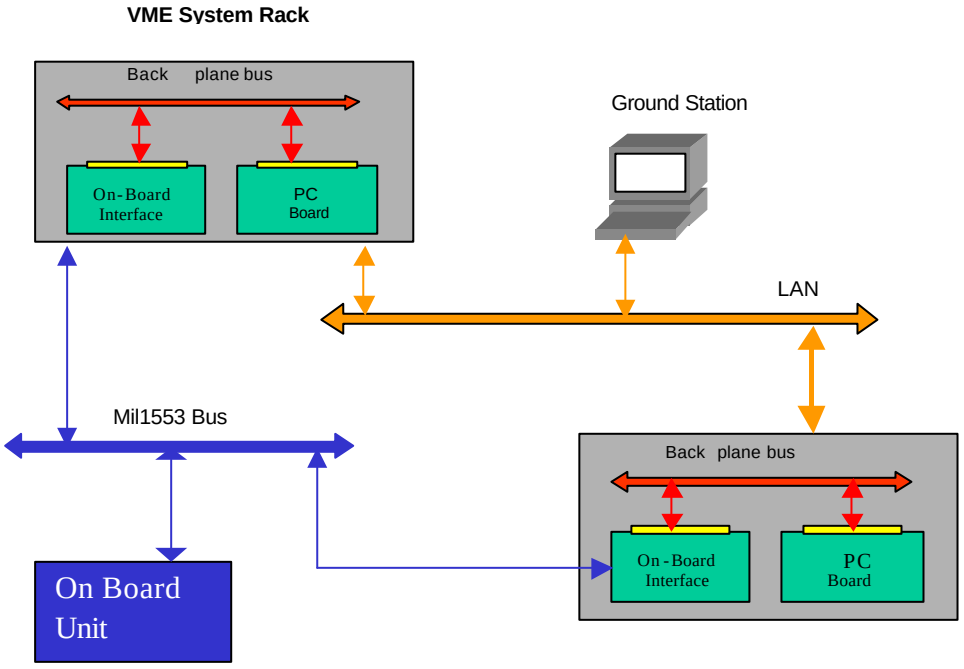


Figure1: CrossPlatform architecture

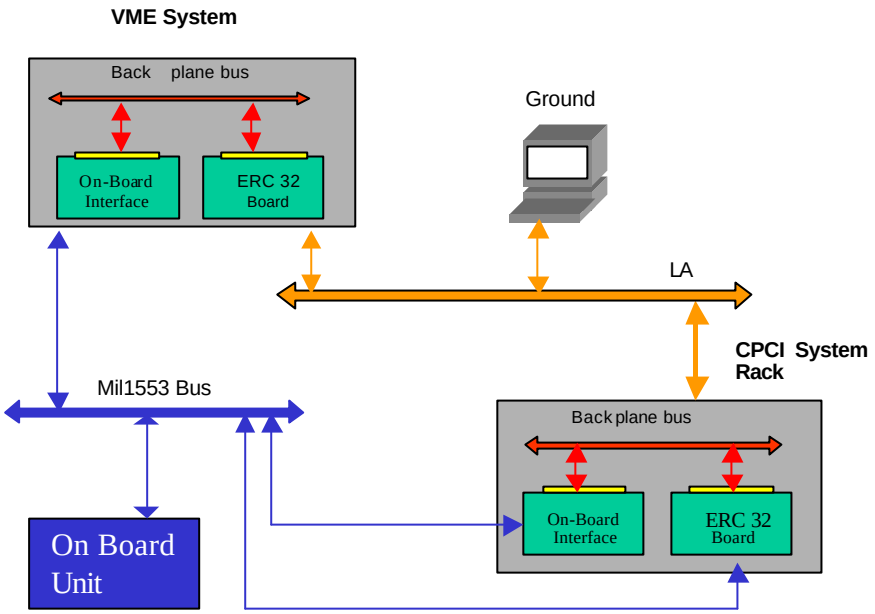


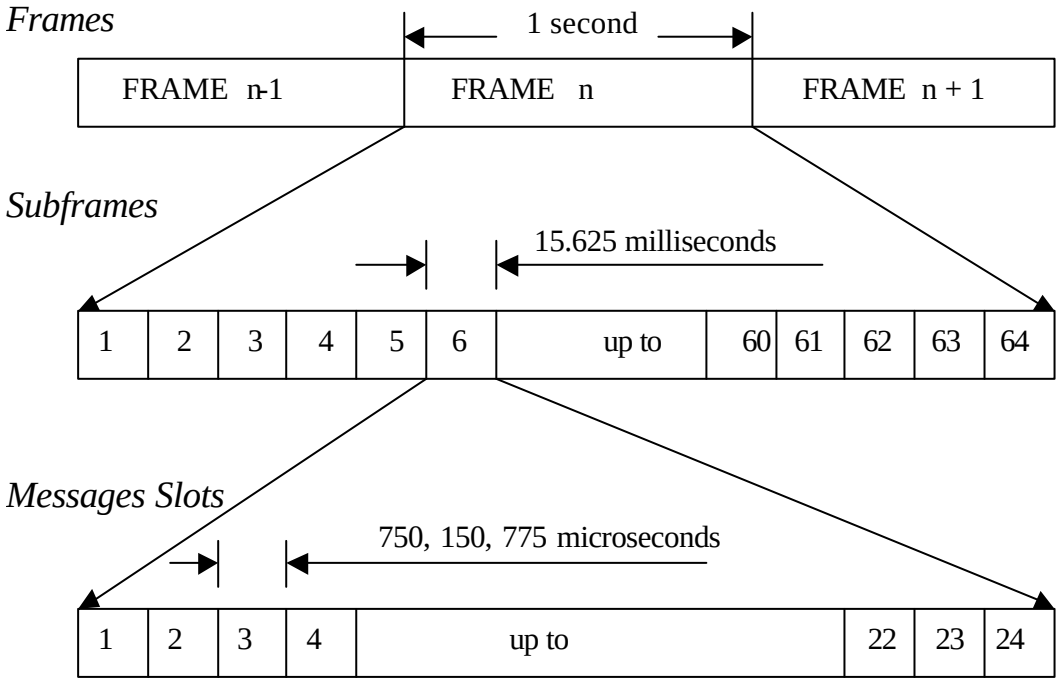
Figure 2: *Target architecture*

II - Overview of the SDBP protocol

II – a - *Framing*

The SDBP protocol is a synchronous protocol and introduces the notion of frames, sub-frame and messages slots. The frequency of frames is one second, each frame is subdivided in 64 subframes whose the duration is 15.625 milliseconds. Each sub-frame contains 24 Mil-1553 message slots. The slot timing is not equal for the all the messages; three types of slot are available:

- The 750 microseconds slot, enables to send a Mil-1553 message with the maximum data words (32 words) it corresponds in Herschel-Planck terminology to the Packet transfer
- The 150 microseconds slot which enables to send mode code, or short Mil-1553 message with one or two data words. Example: sync message or handshaking packet control.
- The 775 microseconds slot is the 24th of the sub frame its main role is the regulation and used to send the asynchronous messages.



II - b - *Protocol Layers*

APPLICATION LAYER **-PUS level** **-FDIR, retry function** **-Bus traffic monitoring** **-Bus profile manager**	
TRANSFER LAYER	**SDBP**
DATA LINK LAYER	**Mil-Std-1553**
PHYSICAL LAYER	**Mil-Std-1553**

In each layer a number of services or primitives are defined by the implementation of the protocol, all the transaction resulting from these primitives are atomic; once a transaction is initiated on the bus, it must be completed before next transaction can start:

- The Data Link Layer

  - ✓ Receive message
  - ✓ Send message
  - ✓ Broadcast message

- The Transfer layer

- ✓ Segmented Exchange
  - ➢ Transfer of telemetry packets.
  - ➢ Transfer of telecommand packet.
- ✓ Non segmented Exchange
  - ➢ Handshaking messages
    - o TC Packet descriptor
    - o TC Packet confirmation
    - o TM Packet request
    - o TM Packet Confirmation
  - ➢ Short and high priority telemetry and telecommand packets
    - o -TM Event
    - o -Asynchronous TM
- ✓ Broadcast Exchange
  - ➢ Time distribution
  - ➢ Synchronization

II – c - *Messages slots and Subaddress allocation*

Subframe	Type	Subframe	Type	Subframe	Type	Subframe	Type
1	TC	17	TC	33	TC/Time	49	TC
2	TM (TC)	18	TM (TC)	34	TM (TC)	50	TM (TC)
3	TM (TC)	19	TM (TC)	35	TM (TC)	51	TM (TC)
4	TM (TC)	20	TM (TC)	36	TM (TC)	52	TM (TC)
5	TM (TC)	21	TM (TC)	37	TM (TC)	53	TM (TC)
6	TM (TC)	22	TM (TC)	38	TM (TC)	54	TM (TC)
7	TM (TC)	23	TM (TC)	39	TM (TC)	55	TM (TC)
8	TM (TC)	24	TM (TC)	40	TM (TC)	56	TM (TC)
9	TM (TC)	25	TM (TC)	41	TM (TC)	57	TM (TC)
10	TM (TC)	26	TM (TC)	42	TM (TC)	58	TM (TC)
11	TM (TC)	27	TM (TC)	43	TM (TC)	59	TM (TC)
12	TM (TC)	28	TM (TC)	44	TM (TC)	60	TM (TC)
13	TM (TC)	29	TM (TC)	45	TM (TC)	61	TM (TC)
14	TM (TC)	30	TM (TC)	46	TM (TC)	62	TM (TC)
15	TM (TC)	31	TM (TC)	47	TM (TC)	63	TM (TC)
16	TM (TC)	32	TM (TC)	48	TM (TC)	64	TM (TC)

Slot	Content/Purpose	Message Description	Sub Address	Duration (uS)
1	Subframe synchronization	Sync Broadcast	31	150
2	Command/Acquisition slot	Status Polling or Low level command	1T 1R	750
3	Command/Acquisition slot	Broadcast Time (subfrm33) Or Event TM	8R 4T	750
4	Command/Acquisition slot	Event TM	5T	750
5	Packet transfer	Packet TM/Packet TC	11T/11R	750
6	Packet transfer	Packet TM/Packet TC	12T/12R	750
7	Packet transfer	Packet TM/Packet TC	13T/13R	750
8	Packet transfer	Packet TM/Packet TC	14T/14R	750
9	Packet transfer	Packet TM/Packet TC	15T	750
10	Packet transfer	Packet TM/Packet TC	16T	750
11	Packet transfer	Packet TM/Packet TC	17T	750
12	Packet transfer	Packet TM/Packet TC	18T	750
13	Packet transfer	Packet TM/Packet TC	19T	750
14	Packet transfer	Packet TM/Packet TC	20T	750
15	Packet transfer	Packet TM/Packet TC	21T	750
16	Packet transfer	Packet TM/Packet TC	22T	750
17	Packet transfer	Packet TM	23T	750
18	Packet transfer	Packet TM	24T	750
19	Packet transfer	Packet TM	25T	750
20	Packet transfer	Packet TM	26T	750
21	Packet control	TC Descriptor Or TC Confirmation	27R 27T	150
22	Packet control	TC Descriptor Or TM Confirmation	27R 10R	150
23	Packet control	TC Descriptor Or TM Request	27R 10T	150
24	Regulation slot	Asynchronous TC	3R/4R	<=775

II - d - *Type of Data*

The telemetry packet will carry one of the following data:

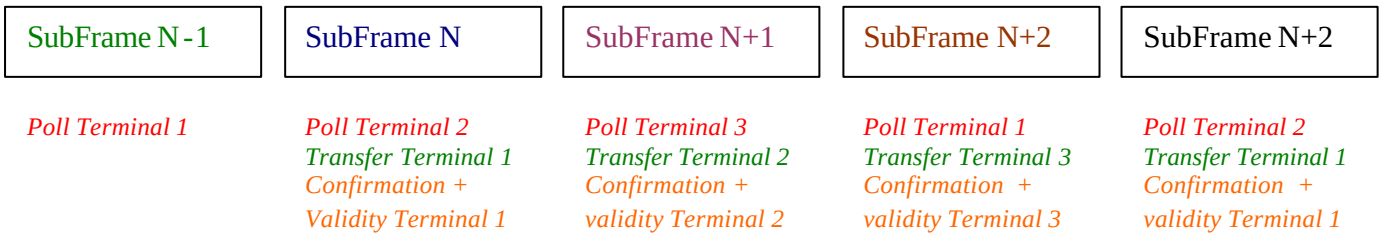
- Science data produced by the instruments.
- Housekeeping.
- AOCS and Control Sensor data.
- PUS Messages

The telecommand packet can be

- On board Commands
- Ground Commands

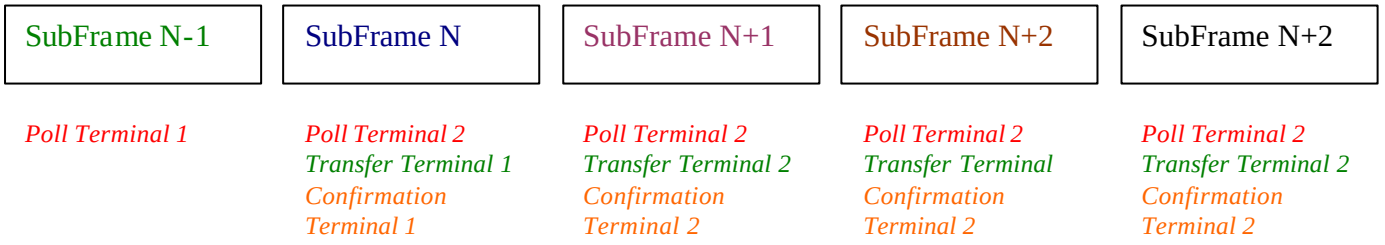
II – e - *Polling*

- Transfer of TM packet
  - ✓ Nominal Mode

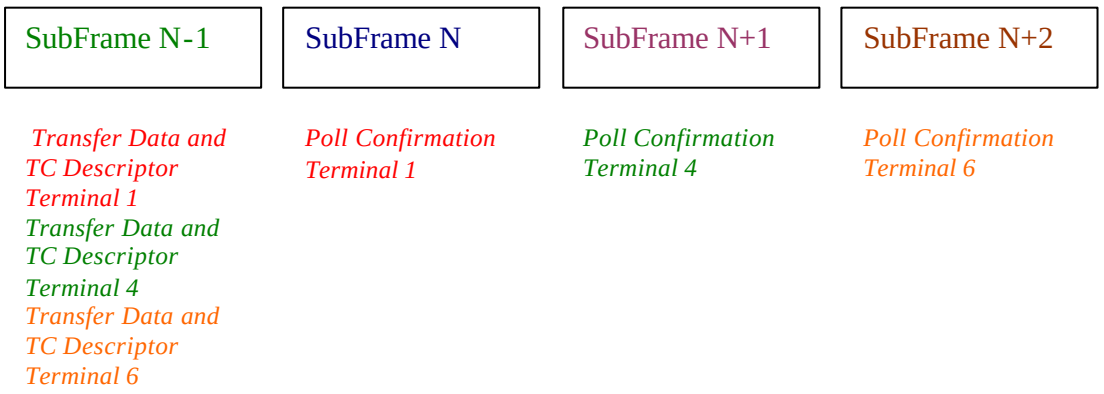


✓ Burst Mode

Below is an example of burst mode granted to user 2



- Transfer of TC packet



III Implementation of the protocol

III – a - The CDMS simulator

The AIM board will be used for implementation of the BC simulator for SDBP protocol.

III – b - The Remote Terminals simulator

The SBS board has been dedicated to implement the simulated Remote terminal for at SDBP level. The system is able to simulate up to 31.

The remote terminals are implemented in Linux environment using the RTAI Real Time extension.

The RTAI is a patch to Linux to make it meeting hard real time requirements. The Real time requirements for the RT protocol are determined by:

- The small number of task needed.
- The small number of interruption are needed to the host processor.
- A simple scheduling policy.
    - In reality only one interruption is needed for the nominal traffic.

What are the real time requirements for the remote terminals?

- The RT shall do after receiving the next sync message:
    - Update the subframe count value.
    - Check in the sync data word if it is allowed to send its TM in this subframe.
    - Check the status of the TM packet transfer that took place in the previous subframe.
    - In case the transfer took place without error; it shall update the TM packet data buffer and the TM Packet Transfer Request

Words within 2 ms if there is any TM Packet pending. It should furthermore update the packet count value.

- Evaluate the TC Packet Transfer Descriptor, carry out the transfer if any and update the TC Packet Transfer Confirmation.

  - The main time constraint that the system has to meet is the minor frame frequency.

II – b- 1- *Implementation of the different primitives*

A – Data Link Layer

At this stage we use the API supplied with the board. These primitive consist essentially on:

- Send or transmit Message: The RT sends to the BC a Mil-1553 message; it is a transaction consisting of command word, a status word and with/without a number of data words.

- Receive Message, a basic transaction on the mil-1553 bus. The message will consist on one command word, a status word and with/without a number of data words.

- Broadcast message this function is not implemented in the RT, we have to define the RT31, and the messages received by this RT are dispatched to all the RTs

B –Transfer Layer

- Nominal Transfer

Nominal TM packet transfer using handshaking protocol:

1- The RT fills the output buffer with new data (SA11-SA26).
2- The RT initializes a TM request packet control.
3- The RT sets flow control bit to 01 in the TM request data word
4- TM request packet control (one Mil-1553 message) is transferred over the bus to Bus Controller

5- BC reads, process and decodes the TM request data words and creates transfer messages.
6- The transfer takes place.
7- BC writes TM confirmation with flow control =11.
8- The RT reads and processes the TM confirmation message.
9- Initialize a new TM packet transfer.

Burst mode TM packet transfer using handshaking protocol

1 - The RT fill the output buffer with new data (SA11-SA26)
2 - The RT sets flow control bit to 01 in the TM request word
3 - TM request packet control is transferred over the bus
4 - BC reads, process and decodes the TM request message and creates transfer messages
5 - The transfer takes place
6 - BC writes TM confirmation with flow control =11
7 - Initialize a new TM packet transfer

a. TM Asynchronous transfer

1 - The RT fill the output buffer with new data, SA5 or SA6 depending on whether it is Event A or Event B.
2 - The RT sets the Event A or Event B flags in the TM request word to 1.
3 - The RT sets flow control bit to 01 in the TM request word.
3 - TM request packet control is transferred over the bus
4 - The bus controller reads, process and decodes the TM request message and creates transfer message.
8 - The transfer takes place
9 - The BC writes exactly the received data message in the SA5 or SA6 data buffer.
10 - The BC put the flow control flags to the value 11.
11 - Initialize a new TM packet transfer if any is pending.

b. TC Asynchronous transfer

VI - Detailed System Description

The overall system is based mainly on subsystems; on is based on VME system bus and the second one on compact PCI system bus, the Processor board can a normal PC board or ERC32 based board.

❖ A VME-based system which includes the following components:

(See Annex 1)

- A 600 MHz PIII PC board.
- A Dual channel and Full Function (BC RT BM) simulation board from SBS manufacturer. This VME board implements both the Mil-Std-1553A and B standards, this board is able to simulate up to 32 RT (See Annex 4).
- A VME crate.
- The Operating System is Linux-RTLinux

❖ A compactPCI-baseds system which includes the following components:

(See Annex 2)

- An 800 MHz PIII PC Board.
- A Dual channels Full Function (BC, RT and BM) simulation board from AIM Company. This cPCI board implements the Mil-Std-1553 A and B standard, this board is able to simulate up to 32 RT (See Annex 5).
- A cPCI crate.
- The Operating System is Windows NT.

❖ Harness

(See Annex 3)

- Stubs
    - 1 input stub

- 2 input stub
- 3 input stub

- Termination

  - Bus termination 78 ohms.
  - Remote terminal termination: 1000/3000 ohms

- Cables

  ✓ Cables with different lengths are available.

❖ ERC32 Based boards.

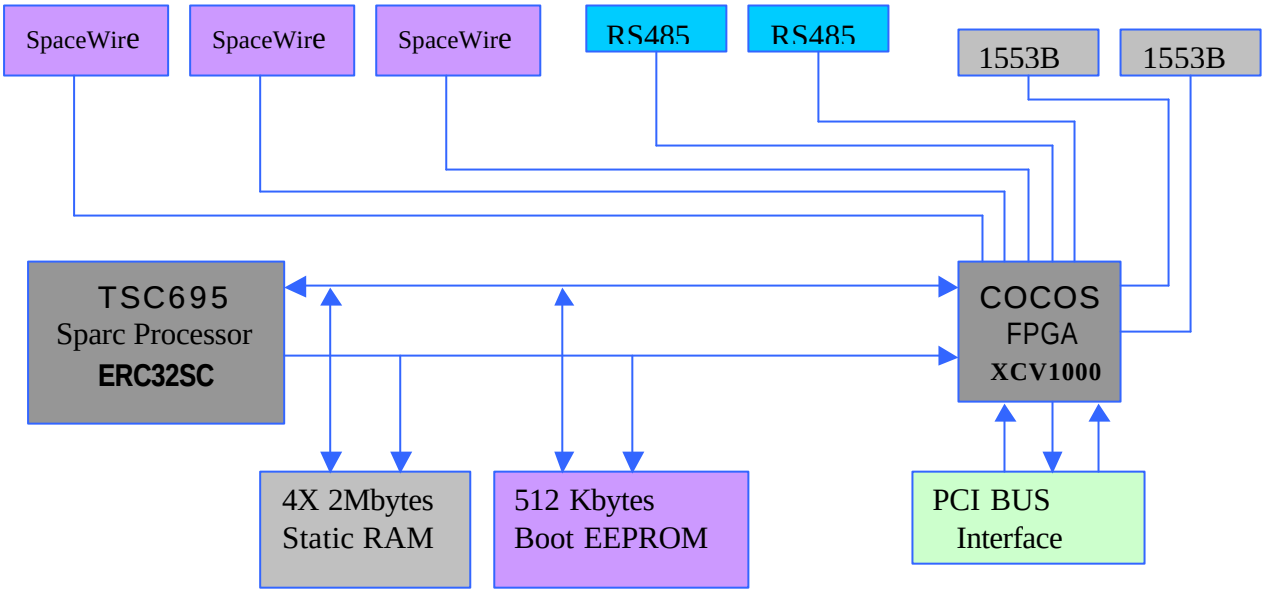
- SAAB board:

(See Annex 6)

Features:

- TSC695 SPARC processor running at 20Mhz
- 8 Mbytes Static RAM with EDAC protection
- 512 Kbytes boot EEPROM
- Compact PCI backplane
- Dual Mil-Std-1553 Buses With BC, RT and BM possibilities
- Dual high –speed RS-485 serial links with up to 476800 baud data rate
- Three Packet Wire Receiver links, with u to 20Mbps data rate
- Three Packet Wire Transmitter links, with u to 20Mbps data rate
- Additional CPU support: Watchdog, Interrupt controller Hardware memory copy
- 6U base Board

System Overview



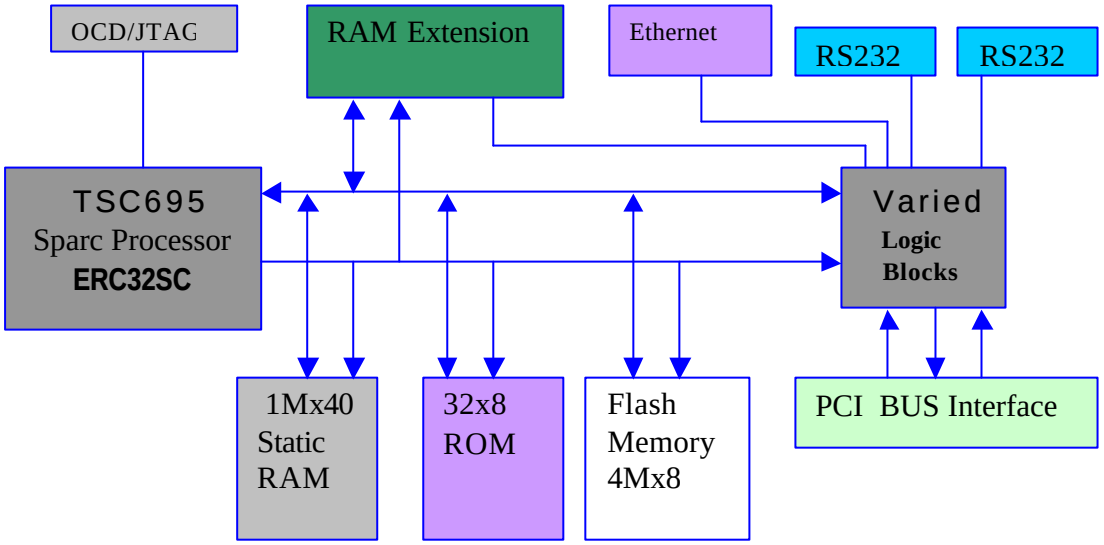
- Tharsys Board

(See Annex 7)

Features:

- ERC32 single chip processor (TSC695)
- Static Memory RAM + EDAC 4Mbyte
- Flash Memory 4Mbytes
- ROM 32 Mbytes
- 2 Serial ports RS232
- 10Baset Ethernet interface
- 2.0 rev 3.0 cPCI-Bus interface
- JTAG Connector
- 6U base Board

System Overview



❖ Software Tools

- Windows NT and Linux drivers for the boards.
- API and library provided by the manufacturer (SBS and AIM).
- Linux Redhat 8.0, Windows NT 4.0.
- Mil-Std-1553 Bus analyzer **(See Annex 8)**.
- Mil-Std-1553 Parameters Viewer **(See Annex 9)**.
- Visual C++, Tcl-tk.
- GCC compiler.