ThalesAlenia Space

ESA COTS TRP

System Definition of COTS-based computer for on-board systems

B. Dellandrea, G. Estaves, B. Alison

TEC-SW & TEC-ED FINAL PRESENTATION DAYS 9TH DECEMBER

Agenda

- Objectives, perimeter and schedule of the CoCs study
- What is a COTS for space area ?
- Requirements applicable to the 3 computers
- Architectures and Building Blocks for the 3 computers
- Adaptation of the space methodologies to the use of COTS
  - Technology selection process
  - Reliability/availability forecast method
  - Performance evaluation methods
- Evaluation of a COTS board (Hi-R) and the COTS software

Agenda

- **Objectives, perimeter and schedule of the CoCs study**
- **What is a COTS for space area ?**
- **Requirements applicable to the 3 computers**
- **Architectures and Building Blocks for the 3 computers**
- **Adaptation of the space methodologies to the use of COTS**
  - Technology selection process
  - Reliability/availability forecast method
  - Performance evaluation methods
- **Evaluation of a COTS board (Hi-R) and the COTS software**

What are the objectives of the COTS study ?

- **To specify and define 3 computer units based on COTS**
  - Hi-R: Highly Reliable computer – OBC for platform
  - Hi-P: High Performance computer – PDHU for payload
  - Hi-V: Highly Available computer – central computer for fail-operational mission
- **To define the methods to apply to the development of these computers**
  - Performance validation
  - Technology selection method
  - Reliability/availability forecast method
  - Qualification method
- **To develop breadboards of the computers**
- **To develop a software able to detect/correct the transient fault induced by the radiation environment into the COTS parts (in relation with specific HW features)**
  - Whatever the selected implementation for the computer

5 ESA contracts (started in July 2008):

- **System contract**

  - Coordination of the 4 contracts, management of the WG, main requirements and architecture definition
  - Prime: TAS-F (Cannes + Toulouse)
  - Subcontractors: TAS-I (Milano) and RSA (A)

- **High-Reliable contract:**

  - Hi-R: TAS-I prime + Italian academics

- **High-Performance contract:**

  - Hi-P: AST-F prime + CGS (I)

- **High -Available contract:**

  -Hi-V: AST-ST (D) prime

- **SW contract:**

  - Delta Technologies(F) prime + TAS-F

Agenda

- Objectives, perimeter and schedule of the CoCs study
- **What is a COTS for space area ?**
- Requirements applicable to the 3 computers
- Architectures and Building Blocks for the 3 computers
- Adaptation of the space methodologies to the use of COTS
  - Technology selection process
  - Reliability/availability forecast method
  - Performance evaluation methods
- Evaluation of a COTS board (Hi-R) and the COTS software

What is a COTS part ? Commercial Off The Shelf

ECSS-Q-20-10A draft: Off-the-shelf items utilization in space systems

COTS items are commercial items that can be procured from the market which are not developed for space application and that can be procured and utilized in a space program

Proposed definition

- Part is not designed and manufactured for space application
- Don't confuse
  - commercial part and commercial temperature range
  - commercial part and military application
- A COTS could be sold for different temperature ranges:
  - Commercial: 0 – 70 °C
  - Industrial: - 40 – 85 °C
  - Extended: - 40 – 125 °C
  - Military: - 55 – 125 °C
- COTS part has not been initially designed to sustain space environment
  - It is not rad-hard or rad tolerant guaranteed
  - Its life duration could be shorter than spacecraft life cycle
- COTS part is not available in hi-rel quality level (QML-Q or QML-V)
  - An adequate screening process associated to the procurement increases the reliability figure (more precisely the confidence level associated to the failure rate value)

Main characteristics of a COTS

Higher performance

- **The COTS allows to fulfil performances which are achievable with difficulties or not achievable with hi-rel parts**
  - The most obvious example is the processing power of the COTS microprocessors

Simpler procurement

- **Shorter procurement delay**
  - Partially counterbalanced by the needed screening tests and radiation characterisation
- **Less exportation restriction**

Economical consideration

- **Initial selling price of COTS parts is always lower but the overall procurement cost is generally higher than for hi-rel parts**
- **Procurement shall include evaluation, radiation characterization, screening operations**
- **Obsolescence (shorter manufacturing cycle) → strategic based on stock to buy & screening of a stock**
- **Additional design effort to**
  - Implement HW and SW features to detect and correct transient faults
  - Design validation using simulation
  - Test equipment including fault injection capability

Which part to procure as COTS ?

Due to the high non-recurring cost implied by the use of a COTS part, the use of COTS will not be generalized to all the parts of a computer

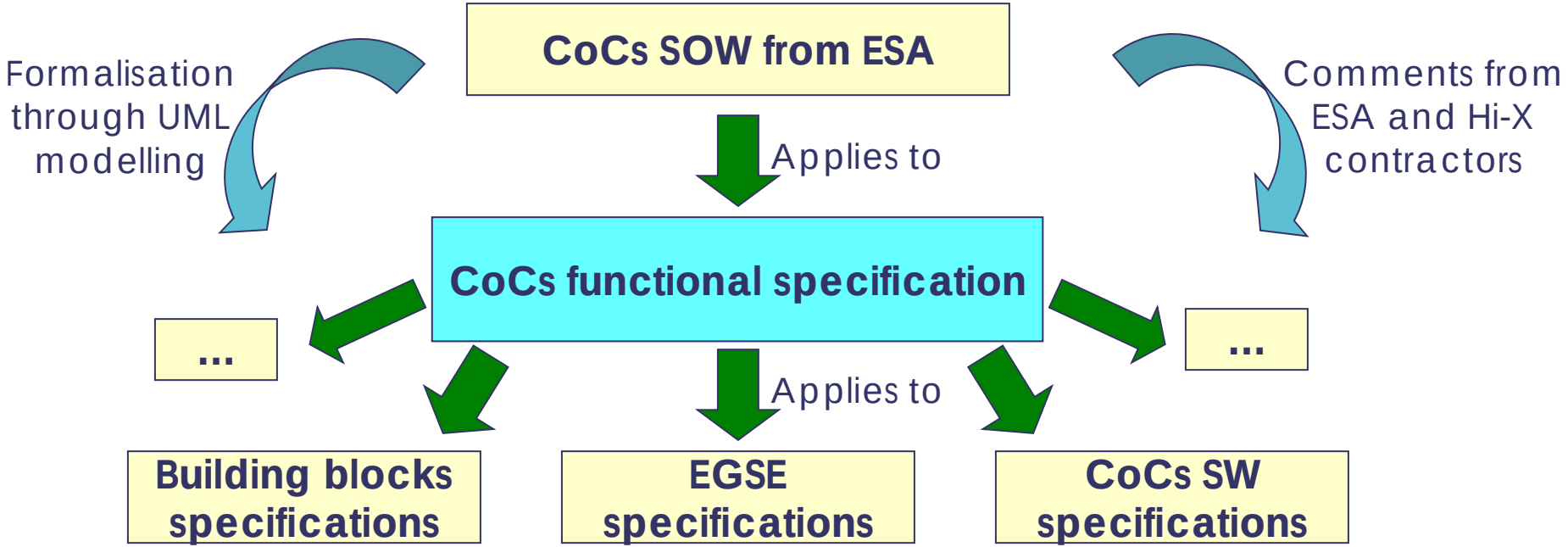
- **Procurement of COTS instead of hi-rel parts should be limited to complex and strategic parts, because:**
  - COTS part provide performance not filled by hi-rel parts
  - Individual cost of the hi-rel part is high enough to justify the required effort to replace it
  - There are exportation restrictions on hi-rel parts
- **Consequently, the main targets for COTS introduction are in a first step:**
  - Processor
  - Companion chip of the processor (FPGA) / DSP
  - Memory
- **Other COTS parts could be added on a case by case basis if the procurement of the hi-rel parts is limited due to ITAR restrictions.**

Agenda

- Objectives, perimeter and schedule of the CoCs study
- What is a COTS for space area ?
- **Requirements applicable to the 3 computers**
- Architectures and Building Blocks for the 3 computers
- Adaptation of the space methodologies to the use of COTS
  - Technology selection process
  - Reliability/availability forecast method
  - Performance evaluation methods
- Evaluation of a COTS board (Hi-R) and the COTS software

Requirements applicable to the 3 computers: CoCs functional specification

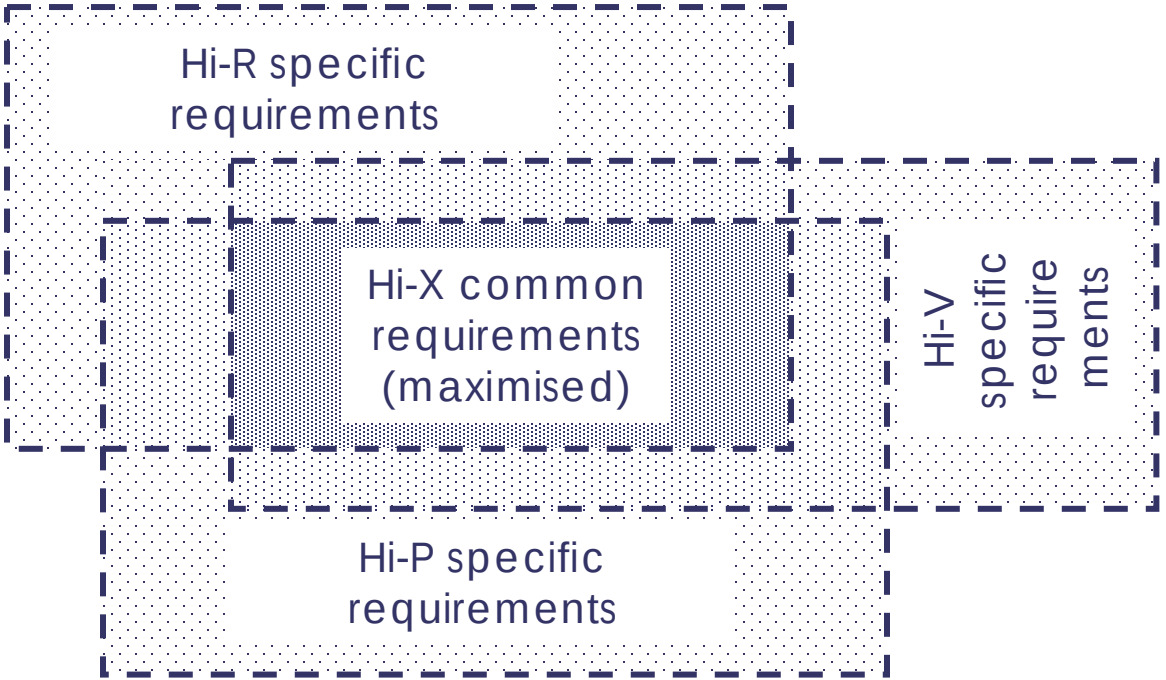
All high-level requirements applicable to the 3 computers are gathered into one single document: “CoCs functional specification”



THALES

Requirements applicable to the 3 computers: Which areas are covered? (1/3)

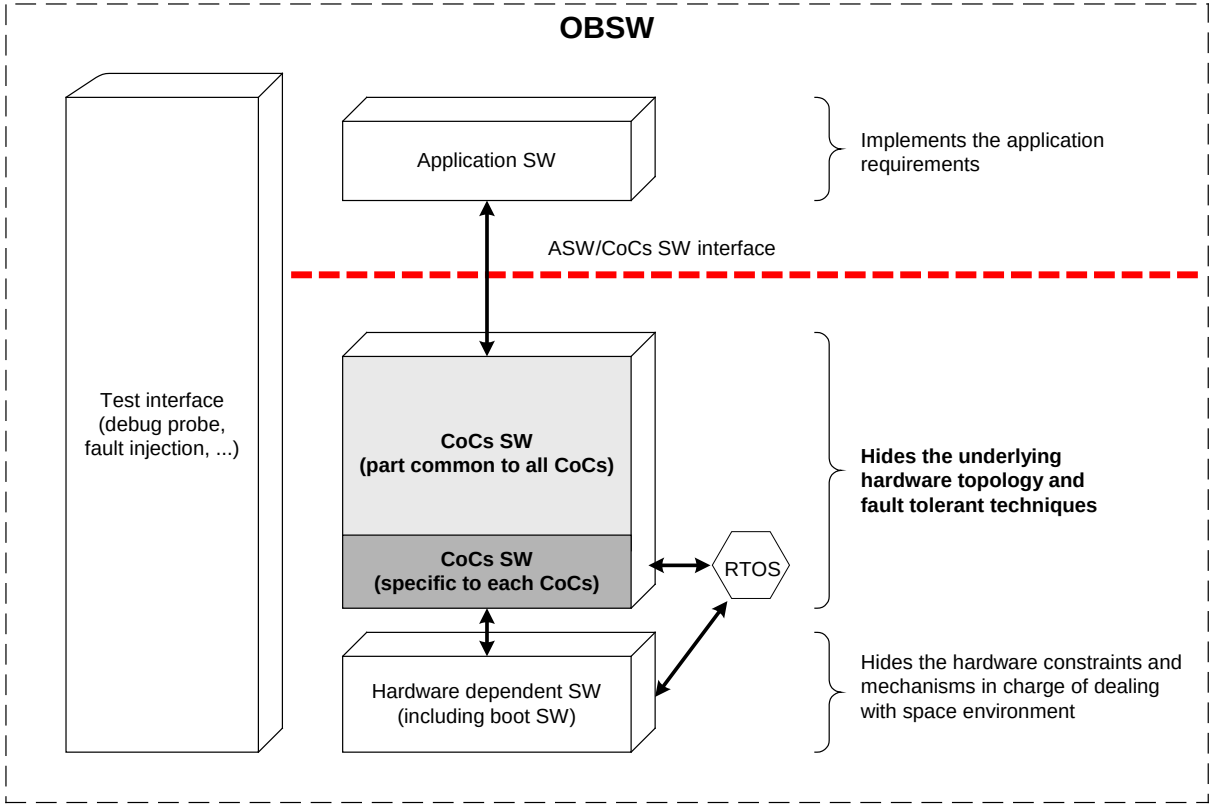
Functional requirements for Hi-X CoCs: commonalities & differences:



THALES

Requirements applicable to the 3 computers: Which areas are covered? (2/3)

Functional requirements for CoCs SW:

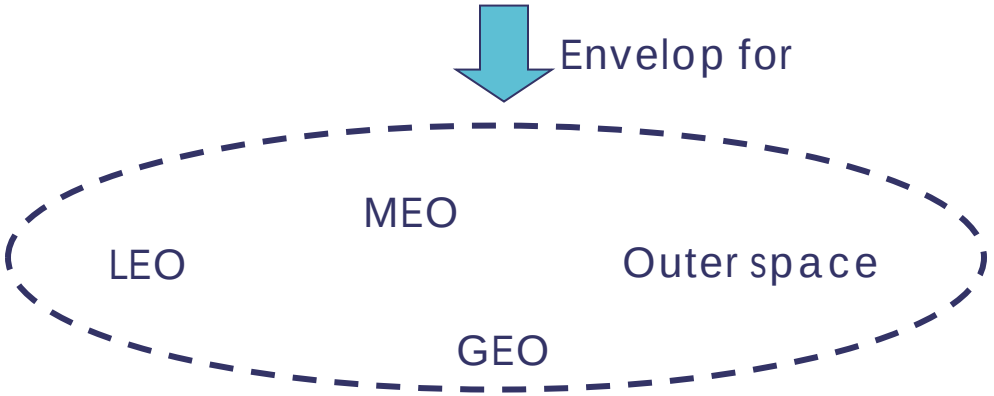


Requirements applicable to the 3 computers: Which areas are covered? (3/3)

Programmatic & quality requirements:

- RAMS.
- Technology selection.
- Building blocks design commonality.
- Evaluation of performances.
- Development, testing and validation requirements (including EGSE).

Physical requirements (mechanical, radiation):



OPEN

THALES

Requirements applicable to the 3 computers: Main features of Hi-X CoCs

Initial requirements

	Hi-R *(TAS-I)*	Hi-P *(ADS-F)*	Hi-V *(ADS-D)*
CPU power	**200 MIPS** (400 MIPS desirable)	**500 MIPS or MFLOPS**	**200 MIPS** (400 MIPS desirable)
RAM	**32 MB**	**128 MB**	**32 MB**
Non volatile memory	**32 MB**	**16 MB**	**32 MB**
SGM	**2 areas 50 Mbps**	**Not required**	**Not specified**
High speed I/O	**3 x 100 Mbps** (target: 3 x 200 Mbps)	**3 x 100 Mbps** (target: 3 links higher than 200 Mbps)	**3 x 100 Mbps** (target: 3 x 200 Mbps)
Low speed I/O	**3 x 1 Mbps**	**3 x 1 Mbps**	**3 x 1 Mbps**
Discrete I/Os	**100 (few kbps)**	**No specific requirement**	**20 (few kbps)**

Requirements applicable to the 3 computers: FDIR terminology

Distinction is done between:

- Permanent failures, which correspond to hardware failures inside the CoCs, preventing it to correctly function. The only possibility to recover this kind of failures is to switch to the redundant units.
- Transient failures, which correspond to failures occurring on CoCs and do not permanently damage it. CoCs HW features and functions developed within the CoCs SW layer must protect as far as possible the on-board SW processing against transient failure occurrences, in particular those due to possible COTS weaknesses.

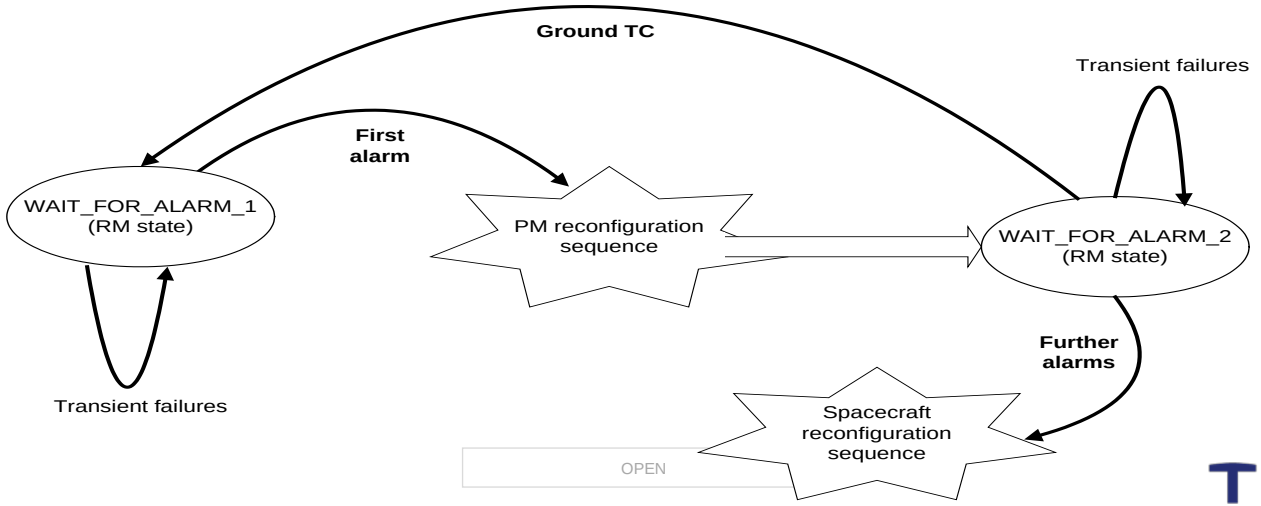
An outage is a temporary loss of the mission due to a failure (permanent or transient).

Different FDIR strategies have been defined with respect to each CoCs mission in order to cope with failures and outages.

In case of failure:

- If possible it is corrected by either the CoCs HW or SW and then the mission carries on after a limited outage.
- If not possible (permanent or not recoverable transient failure), Hi-R CoCs tries a PM reset with an OBSW saved context.

In case of second not recoverable failure, Hi-R CoCs enters in safe mode with use of redundant units. Exit from safe mode is then under Ground responsibility.



Requirements applicable to the 3 computers: Hi-V FDIR strategies

The notion of availability period is introduced, during which Hi-V CoCs must avoid any outage.

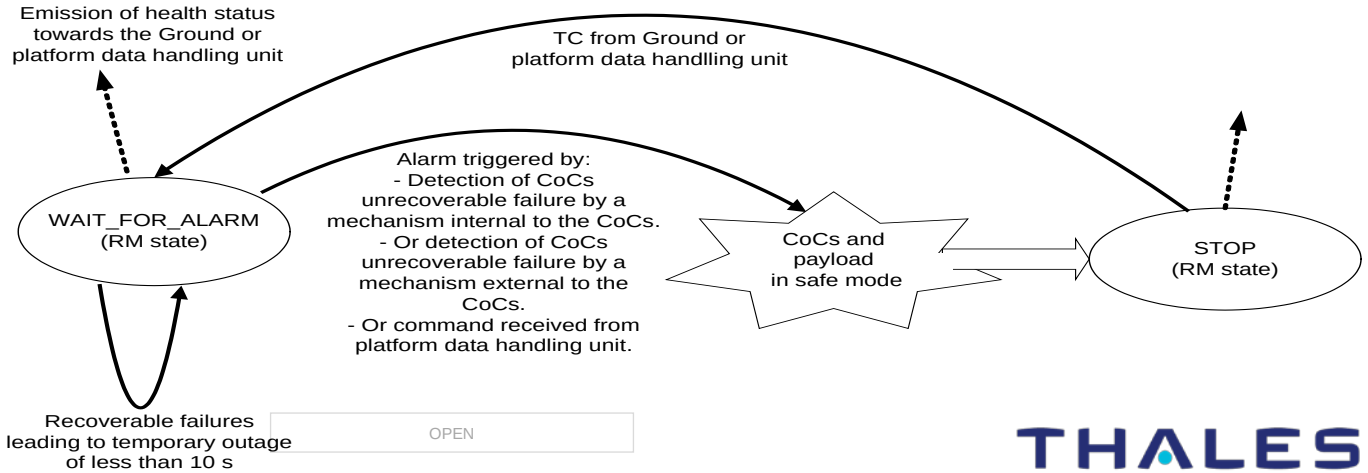
- **Whatever the kind of failure that occurs, Hi-V CoCs must try to ensure mission continuity as far as possible, even if this may lead to the spacecraft loss.**

Outside availability periods, Hi-V CoCs must be able to ensure the mission needs, but without neither mission continuity constraints nor stringent reaction times.

The notion of high processing period is introduced, during which Hi-P CoCs must be able to provide higher CPU capabilities.

In case of failure:

- If possible it is corrected by either the CoCs HW or SW and then the mission carries on after a limited outage.
- If not possible (permanent or not recoverable transient failure), Hi-P CoCs enters in safe mode. Exit from safe mode is then under Ground responsibility.



Requirements applicable to the 3 computers: Availability features of Hi-X CoCs

	Hi-R	Hi-P	Hi-V
Outage in case of transient failure	< 10 s	None (during availability periods) Not specified (outside availability periods)	< 10 s (during high processing periods) < 20 s (outside high processing periods)
Transition duration towards SAFE mode in case of permanent failure	< 10 s	Not applicable	< 10 s
Mean time between outages linked to transient failure	> 30 days	Not applicable (during availability periods) > 2 days (outside availability periods)	> 2 days
Mean time between outages linked to permanent failure	> 30 days	Not applicable (during availability periods) > 2 days (outside availability periods)	Not applicable
Reliability at the end of the mission	> 0.95	> 0.95	> 0.8

Common Building Blocks

In the frame of System contract we focused on:

- Identification of Commonalities among the three architectures
- Definition of potentially common building blocks
- Specific Building Blocks features to match with different architectures

Building Blocks analyzed:

- TMTC Module
- Processor Module
- Reconfiguration Module
- I/O Module
- Power Supply

Block	Hi-P	Hi-V	Hi-R
TMTC	NOT APPLICABLE (except in the hypothesis of merging PF and PL computers)	YES (Connection with the rest of the CoCs need to be further assessed)	YES
PM	YES (with cPCI Interface TBC)	YES (with cPCI Interface)	YES
RM	TBD (reduced configuration ?)	NOT APPLICABLE	YES
I/O	YES (with cPCI Interface TBC)	YES (with cPCI Interface)	YES

Agenda

- Objectives, perimeter and schedule of the CoCs study
- What is a COTS for space area ?
- Requirements applicable to the 3 computers
- **Architectures and Building Blocks for the 3 computers**
- Adaptation of the space methodologies to the use of COTS
  - Technology selection process
  - Reliability/availability forecast method
  - Performance evaluation methods
- Evaluation of a COTS board (Hi-R) and the COTS software

Architectures

Three different architectures have been defined in the frame of the three HW Contracts:

- **Hi-R:**
  - Classical redundant Platform Computer: architecture.
  - Cross-strapping among Modules
  - Internal interfaces based on Serial Links (mainly SpaceWire)
- **Hi-V:**
  - Computer pool made of self-standing computer systems.
  - Specific Fault Management Layer.
  - Each Computer System based on parallel cPCI Backplane
- **Hi-P:**
  - modular architecture based on cPCI Backplane
  - no cross-strapping among modules.
  - Specific I/O Module, based on Space technology, acting as supervisor of COTS Based High Performance Processing Module

Agenda

- Objectives, perimeter and schedule of the CoCs study
- What is a COTS for space area ?
- Requirements applicable to the 3 computers
- Architectures and Building Blocks for the 3 computers
- **Adaptation of the space methodologies to the use of COTS**
  - Technology selection process
  - Reliability/availability forecast method
  - Performance evaluation methods
- Evaluation of a COTS board (Hi-R) and the COTS software

Technology Selection Process (1/2)

Selection method composed of three steps :

- Preliminary COTS selection (COTS lists)
- COTS candidates characterization
- COTS selected candidates procurement and qualification

Selection and Procurement Philosophy

- Die procurement approach
- Plastic COTS approach for processors
- Plastic COTS approach for memories

COTS LISTS

- Hardware designers have selected a list of COTS for the three computers : Hi-P, Hi-V and Hi-R.

Selection Criteria and Ranking

- A list of criteria and a ranking have been set to help to the selection of COTS.

Technology Selection Process (2/2)

Technology Selection Sheet

- **A technology selection sheet has been elaborated thanks to the contribution of all the Technology group members remarks and comments.**
  This selection sheet includes 8 main chapters :
  - **1. General information**
  - **2. Procurement**
  - **3. Information from manufacturer**
  - **4. Other informations/ Experience**
  - **5. Radiation status**
  - **6. Technical Status for Processors**
  - **7. Technical Status for memories**
  - **8. Technical Status for FPGA**

Reliability standard selection

MILHDBK 217 F + N2 is the only worldwide known reliability standard for commercial projects.

- **But**
  - Obsolete for today complex ICs and packages (last issue dated 1995)
  - Not suited for COTS

Using MIL 217 IC model extension (processors)

- **Moore law extended**
- **Moore law shifted from reference of MIL 217 to take into account the gap between commercial and Hi-rel technologies (8 years)**

Using MIL 217 memories model extension

- **capacity up to 16 Gb**
- **MIL 217 package model extension**
- **BGA assessed as a CQFP**
- **TSOP assessed as a CQFP**

COTS Failure rate assessment guidelines

Quality factor defined with respect to the additional screening procedure. Typically between 0.5 and 1 (0.25 for class S)

→ important to tune the formulae & get appropriate figures

All other parameters defined by MIL HDBK 217 F + N2

Warning:

- Failure rate is not the only concern for COTS complex ICs.
- Deep sub micron technologies (< 90 nm) feature a shorter life duration than conventional ones. Life duration may be below 15 years.
- Most applications for COTS have a short life duration. Hence Silicon manufacturers don't need to improve this parameter.

Performance evaluation methods (1/9)

Main goal :

- **To define a common way to evaluate the performance of the 3 CoCs computers embedding fault-tolerance mechanisms**

Performance Evaluation meanings :

- **In terms of pure processing power**
- **In the Space context, the occurrence of failure induced by the environment (SEU/SEFI)**
- **In terms of fault mitigation techniques efficiency**
  - Overhead due to fault Detection/protection mechanisms
  - Overhead due to recovery procedures execution

Processing power performance

- **Benchmark set is proposed to :**
  - Evaluate the global performance of each CoCs
  - Get a common reference for comparison (between CoCs)
- **For each CoC, must be checked at least :**
  - the required processing capability,
  - the I/O management capability,
  - the overhead induced by fault tolerance mechanisms
  - → several test config. with and without FT mechanisms and/or FT injections

NB : In the case of Hi-P computers, high & low processing power mode to be evaluated by checking also the power consumption for each mode.

- **Benchmarks based on the "NGSDSP software benchmark" ESA document**

Processing power performance (benchmark list 1/2)

- **B0 - Dhrystone benchmark: allow a “simple” comparison in DMIPS between CoCs device.**
- **B1 – I/O performance: measure and quantify the performance of the processor interfaces using simultaneous data transmission in both direction.**
  - Test A: test each link individually.
  - Test B: test by incrementing the number of current used links.
- **B2 - Analogue Data Acquisition, Processing and Output: represents a typical DSP application. The aim is to :**
  - simulate the data acquisition from an ADC (by reading data from memory),
  - process signal filtering or/and transformation (FIR LP filter or/and FFT),
  - write the output to memory through bus interface (simulated DAC).

Performance evaluation methods (4/9)

Processing power performance (benchmark list 2/2)

- **B3 - Image Data Compression and Packaging: simulate an application based on image compression.**
  - compress two defined images using the CCSDS Standard Image Compression Algorithm (lossless and lossy mode).
- **B4 - Onboard Data Processing Case 1: simulate an application based on optical remote sensing data acquisition and processing.**
  - read simulated integer data from bus interface,
  - filter data by a complex 128 tap FIR filter,
  - decimate to achieve a data rate reduction of 80%,
  - compress data using CCSDS Lossless Data Compression.
- **B5 - Onboard Data Processing Case 2: simulate an application based on the processing of microwave (radar) remote sensing data.**
  - Read simulated data from bus interface,
  - perform a digital I/Q demodulation,
  - perform a decimal filtering for each part (I and Q).
  - write the output in memory (through bus interface).

Most algorithms are provided in C language to facilitate the porting to specific architecture.

General method of performance evaluation presented

- **Hi-P, Hi-V and Hi-R implementing a different fault-tolerant method**

→ **a common way to evaluate reliability & performance of each computer was necessary.**

3 methods were proposed

- **HWIFI (Hardware Implemented Fault Injection)**
  - Heavy ions Fault injection (ions, protons)…
  - Laser Fault injection (interest: accuracy in FT injection area)
- **SCIFI (Scan Chain Implemented Fault Injection):**
  - Faults injection via scan-chains (JTAG) & HW built-in logic.
- **SWIFI (Software Implemented Fault Injection) :**
  - Requiring additional specific software functions :
  - at CoCs SW level,
  - at CoCs HW level (direct access to mem., registers (via Jtag), fault injection at companion chip level)

➔ **The SWIFI method was the method selected to evaluate the CoC SW on Hi-R board with a bit of SCIFI (JTAG-driven)**

Fault-Tolerance mechanisms techniques performance

Evaluation Method proposed

- Based on Fault Injection using SWIFI (Runtime Injection technique with interrupt-based triggers) and SCIFI (JTAG interface).
  - Faults are injected in the system to simulate SEU thanks to the use of the JTAG probe,
  - Triggers = breakpoints,
  - Scripts = Faults shall be generated automatically and randomly => “Fault Injection Scenarios”.

- Check/monitor the behaviour of CoCs SW/HW by using
  - CoCs SW Real-time observation tool,
  - A recording or history function to log the CoCs SW behavior (code profiling, traces…) and perform post analyses.
  - Performance analysis tools : CPU Load, code profiling, memory usage analysis, execution timing…

Fault injection example : In simulation environment

Python script using Simics API and Simics interface

Faults Injection

SIMICS (under Windows XP/Linux)
+ running virtual platform (CoCs HW)
+ running CoCs SW
+ debugger capabilities

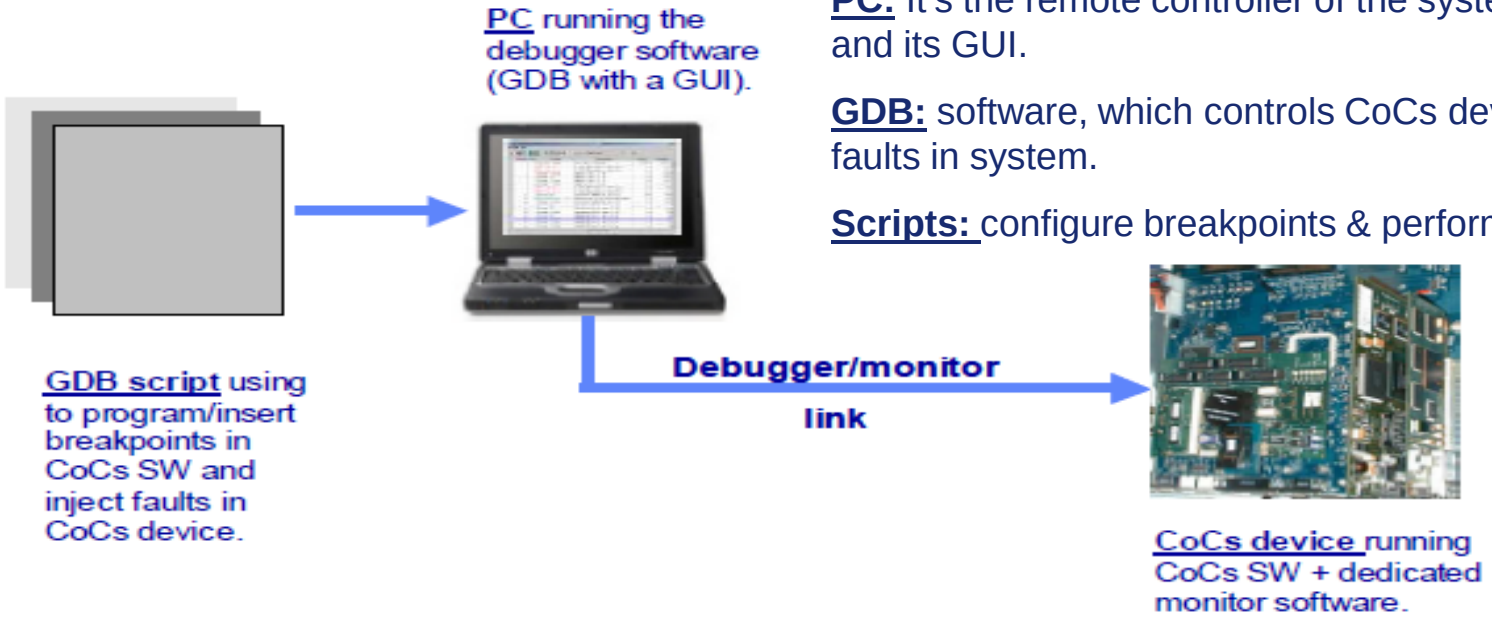
Traces: create tracer to watch changes in:
+ I/O access
+ Control Registers
+ Exceptions raising
+ Simics events generation

Code Profiling: create branch recorder to watch:
+ address profile
+ data profile

Haps (Simics Events): attach customised call-back functions to specific event:
+ Control register R/W
+ Given exception/trap raising
+ Simulation start/stop
+ etc…

THALES

Fault injection example : In Real target environment



CoCs device: runs CoCs SW & monitor software. Offer an interface to communicate with GDB.

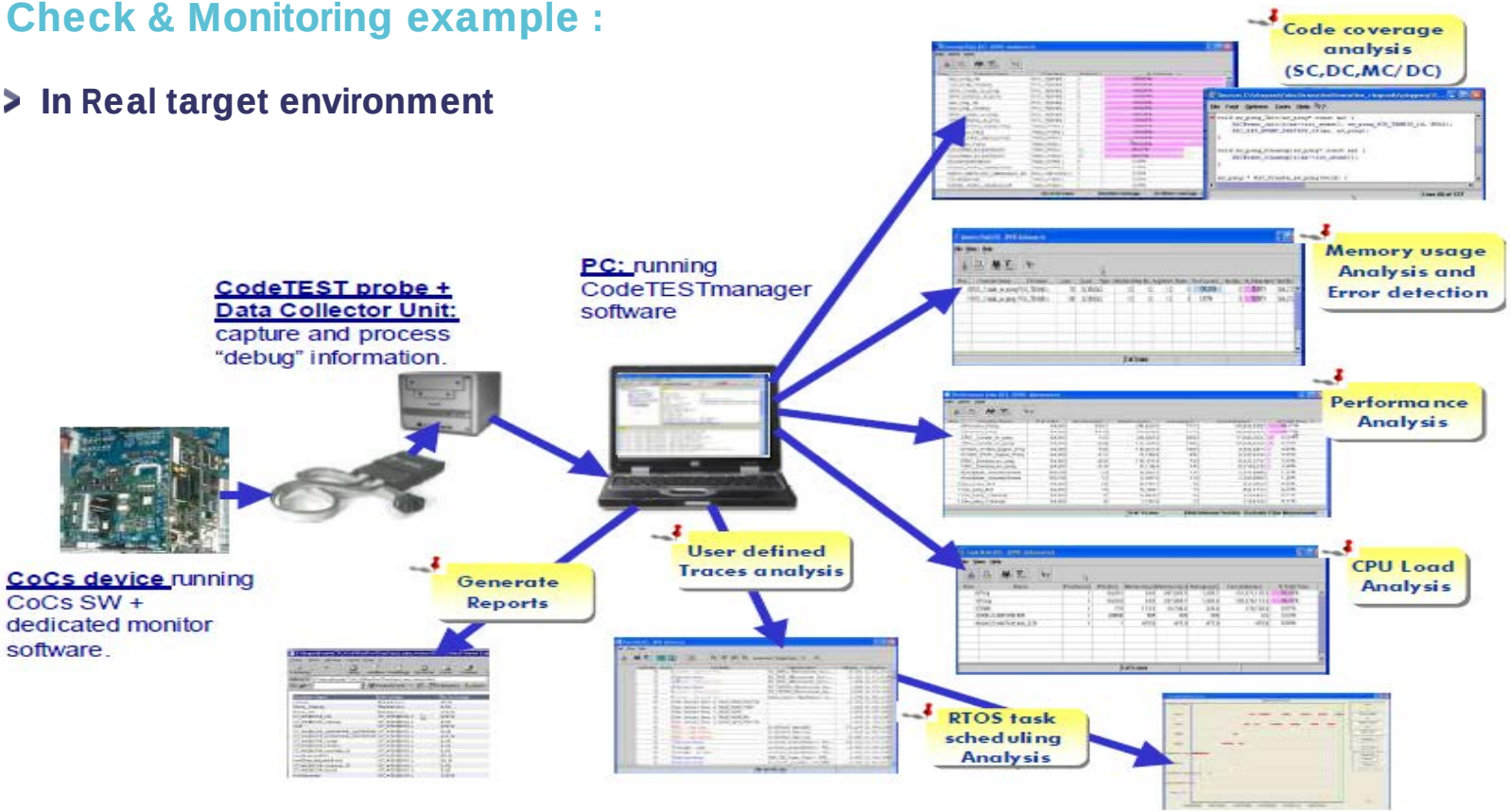
PC: It's the remote controller of the system. It runs GDB and its GUI.

GDB: software, which controls CoCs device & injects faults in system.

Scripts: configure breakpoints & perform fault injection.

Check & Monitoring example :

- **In Real target environment**



Agenda

- Objectives, perimeter and schedule of the CoCs study
- What is a COTS for space area ?
- Requirements applicable to the 3 computers
- Architectures and Building Blocks for the 3 computers
- Adaptation of the space methodologies to the use of COTS
  - Technology selection process
  - Reliability/availability forecast method
  - Performance evaluation methods
- **Evaluation of a COTS board (Hi-R) and the COTS software**

Evaluation performed in several steps

Step 1: Evaluated the global performances with a synthetic benchmark

- **Goal: Become familiar with the board and dev. environment (installation & run “simple tests”)**
- **Execute benchmark B0 – Dhrystone Benchmark (as Ref : ESA doc. Next Gen DSP SW Benchmark)**

Step 2: Run a real DSP application to assess performances of HI-R without CoCs SW

- **Goals: To evaluate the difficulties related to the port of a “Payload SW” on the Hi-R board :**
  - Performing FFT computation , Ready to be coupled with CoCs SW
  - Measure the execution timings on a typical scenario

→ **DTSO integrated the COCS SW on the Hi-R Board**

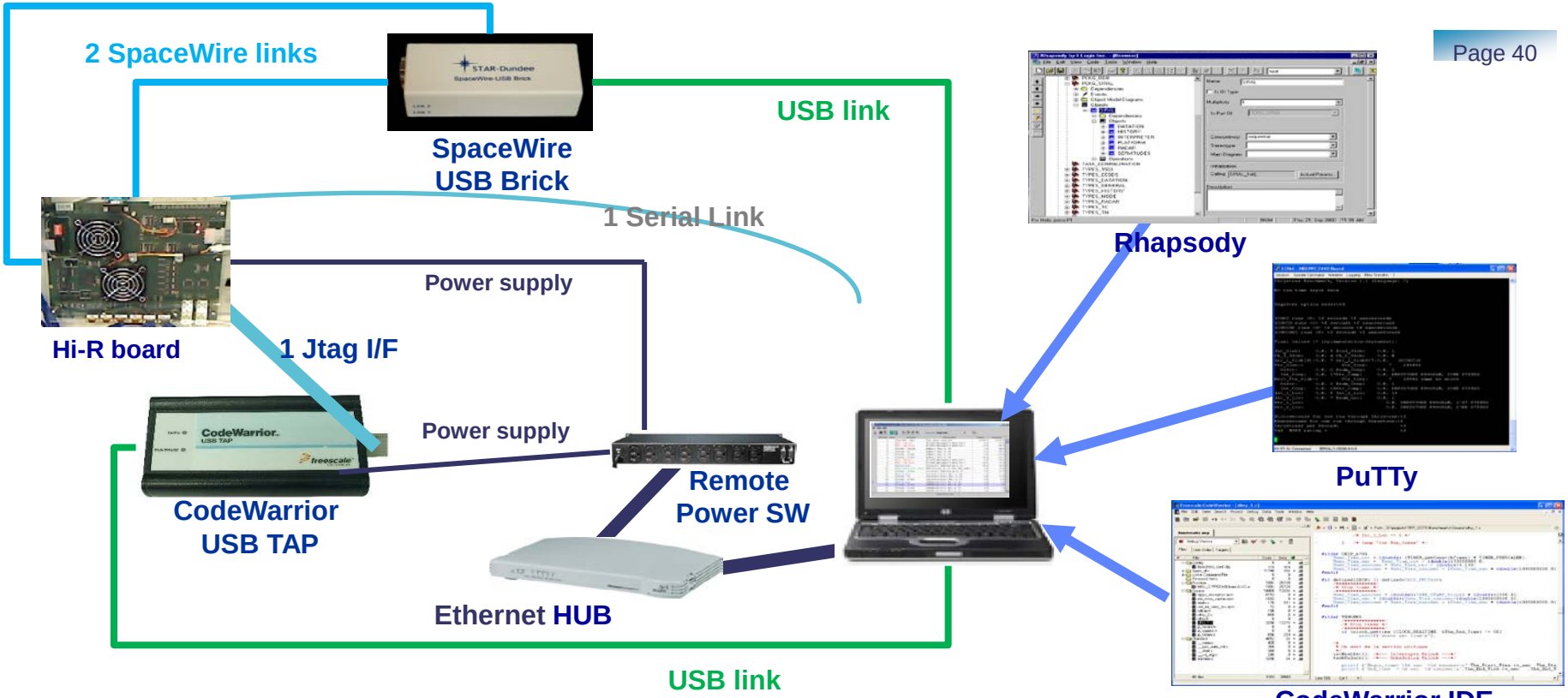
Step 3: Run the CoCs SW from DTSO on the Hi-R board

- **Goals: Become familiar with CoCs SW implementation, replaying test suite as per the “CoCs Software Final Report” on the HI-R board**

Step 4: Execute Test Application with CoCs SW on the Hi-R Board

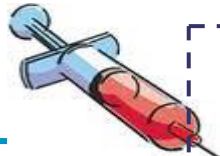
- **Goals : To evaluate**
  - APIs to control the Tasks execution, to manage fault and voting mechanisms , to inject faults via SWIFI
  - The potential technical difficulties, the performance impact of the test application, the efficiency of Fault injection and tolerance mechanisms

Lab. Configuration for the Evaluation



SW configuration for the CoCs SW evaluation

Fault injection



Beacon Signal Simulated Acquisition

Test ASW
no RTOS

CoCs SW

HI-R Board

FPGA | PPC7448 | IO

TM : Demodulated Beacon messages

CoCs SW offers a large set of services to manage faults

- Quite complete list of API
- No major lack in the API provided

Some improvement were proposed:

- Documentation need more detailed description of underneath concepts
- API description can be improved sometimes
- Abstraction of CoCs SW mechanisms not enough hidden to the User Application developer (too close to the HW)

➔ The successful evaluation confirmed the performances of the HI-R Board and validated the usage of CoCs SW supporting layers to mitigate faults.

Hi-R TAS-I board performances confirmed

- **Dhrystone Benchmark result :** 981.29 DMIPS
  - **Config :** PPC7448@1GHz – Bus@100Mhz - 133Mhz for DDRII - optimisation level 2
- **Same benchmark gave :** 834 DMIPS for Maxwell Board
  - **Config :** SCS750 @800Mhz – Bus 40Mhz
- **Same benchmark gave :** 739 DMIPS for Freescale HPCII evaluation Board
  - **Config :** PPC7448@1Ghz – Bus 100Mhz- 133Mhz for DDRII 133Mhz

Hi-R Board is even faster than Freescale evaluation board with PPC744@1GHz

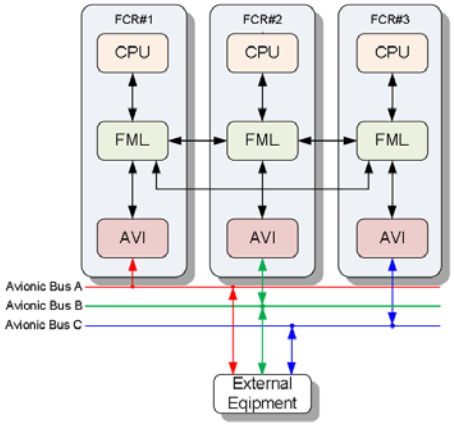
Hi-R Board was quickly up and running in TAS-F Toulouse labs once probes & EGSEs received

Hi-R Board is really simple to use, no major difficulties encountered

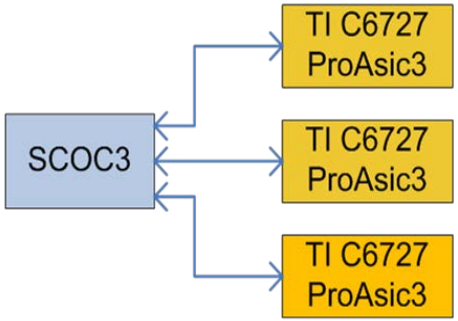
Results: architectures & innovative solutions

The requirements breakdown led to very different architectures:

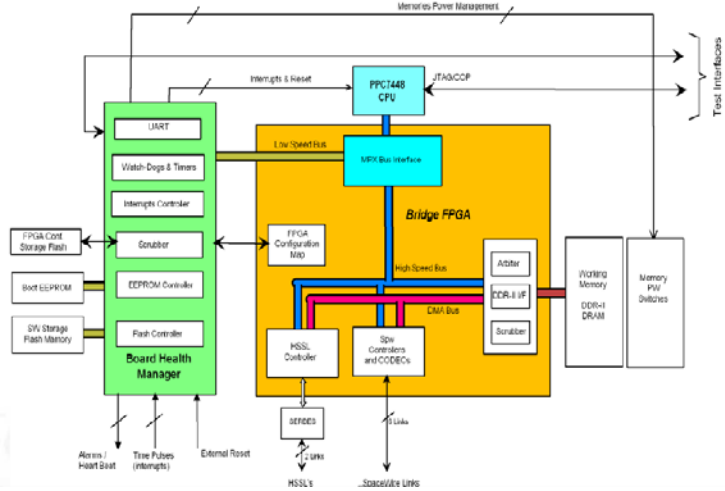
- Hi-V: triplicated architecture with voters
- Hi-P: “standard” non-COTS LEON3-based PM using companion COTS DSPs
- Hi-R: redundant COTS processors with HW&SW protections



Led to ORION OBC (ADS-D)



Evolutions within ADS-F roadmap



Led to MTG-I LI ICU (TAS-I)

THALES

Thanks for your attention