# Modular Architecture for Robust Computation

**Dr. W.Gasti:** *wahida.gasti@esa.int*
**A.Senior:** *alan.senior@sea.co.uk*
**Dr. O. Emam:** *omar.emam@astrium.eads.net*
**S. Fowell:** *stuart.fowell@scisys.co.uk*

# MARC Development
# Background & Objectives

Background:

- ✓ Compilation and Analysis of future On-Board Computing Requirement from ESA Application Programs and Science Missions

General Objectives :

- ✓ Improving on-board intelligence to maximize mission return (i.e. information per mission & per € invested)
- ✓ Securing the feasibility of complex systems
- ✓ Bounding the development times
- ✓ Designing to cost …. While reducing risks
- ✓ Automating repetitive tasks

# MARC Development Strategy

Since the mid 90s, an integrated approach to built On-board Computers has been used, based on three pillars

- ✓ Specific functions allocation to modules and units
  - Able to evolve according to consistent roadmaps
  - Made attractive for non institutional and export markets
- ✓ Technology rationalisation
  - Well selected devices (ASSPs)
  - Proper design tools
  - User support
- ✓ Standardisation: in particular at interfaces level
  - ECSS
  - CCSDS

MARC System:

✓Distributed HW System Architecture

✓for both payload data processing & spacecraft data handling

✓based on a SpaceWire network.

## Agenda

- ✓ MARC Requirements
- ✓ MARC FT-Design Principles
- ✓ MARC FT-Design
  - MARC FT-Design: Hardware Building Blocks
    - MARC Highly Reliable HW Components
    - MARC Cluster & HW Architecture
    - RMAP for SpW-Module memory access
  - MARC FT-Design: Software Building Blocks
    - MARC GenFas SW & SOIS Architecture
    - MARC FDIR Manager & Tool
- ✓ MARC Demonstrator
- ✓ Conclusion

# MARC Requirements

MARC Robust and Fault Tolerant Computing System Principal Requirements are:

- A Modular and Scalable Network System Architecture
- A SpW network capable of handling the future demands of both spacecraft data handling and payload data processing.
- Hierarchical Fault Detection Isolation and Recovery (FDIR) principles
- CCSDS SOIS layered Software Architecture
- Flight Qualifiable technologies such that the design can be considered as “One Step from a Flight Model".
- The key SpW components and standards developed by ESA: SpW links, SpW-10X router, RMAP, Leon2FT based processor.

# MARC FT-Design Principle

MARC Hierarchal FDIR is based on 5 Reliability levels

(from the highest to lowest level)

- Ground
- TMTC system
- Hardware Reconfiguration Controller (HRC)
- Master Core Computing Module (CCM)
- General Computing Module (GCM) based on COTS processor and/or Module(s) level FDIR.

The MARC Fault Tolerant (FT) design principles:

- No single point of failure:
- No single point of repair
- Fault isolation to the failing component, module;
- Fault containment to prevent propagation of the failure;
- Availability of autonomous safe computing system mode and reversion to operational modes on external command;

# MARC FT-Design

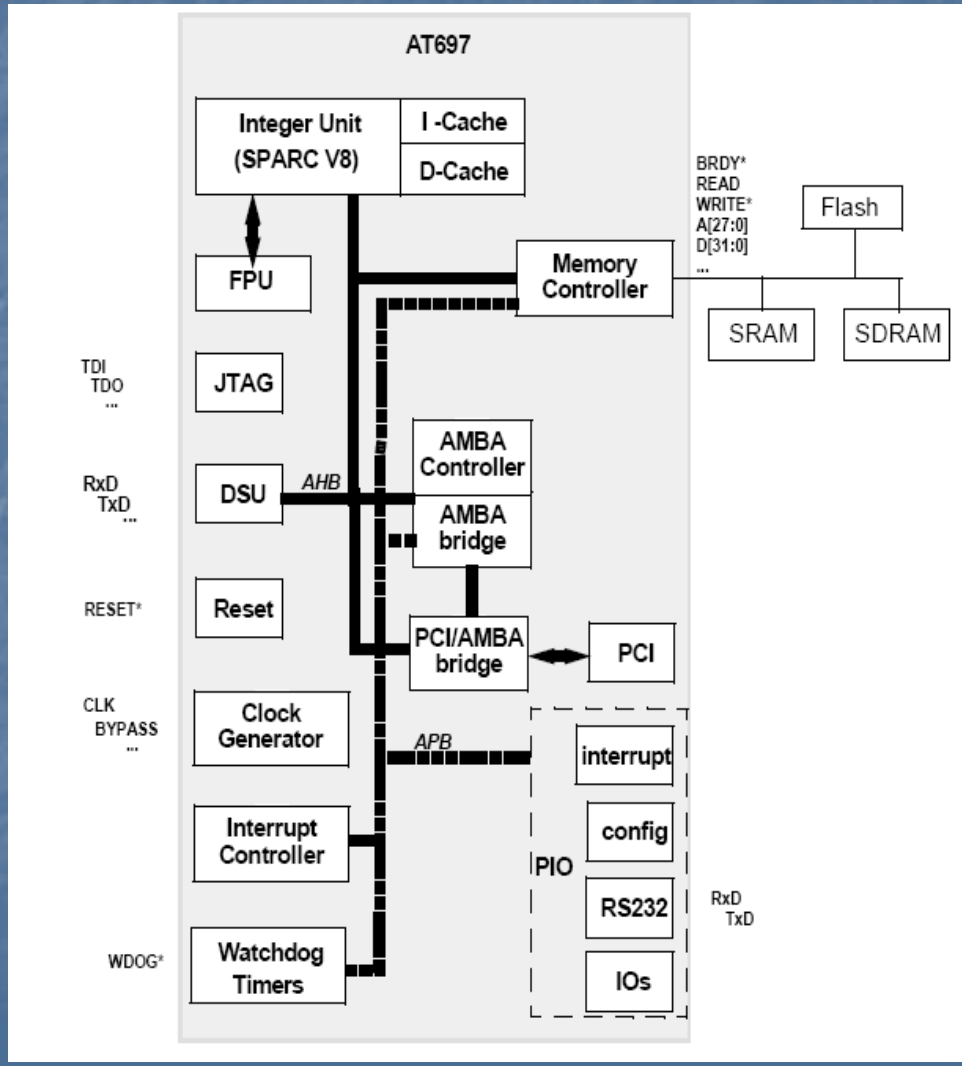
- To fulfil the MARC requirements and the FT-design principles:
- MARC has been built starting from a set of HW and SW Building Blocks (BBs) designed to insure its robustness
- MARC Hardware Building Blocks
  - MARC Highly Reliable HW Components
  - MARC Cluster & HW Architecture
  - RMAP for MARC memory access
- MARC Software Building Blocks
  - MARC GenFas SW & SOIS Architecture
  - MARC FDIR Manager & Tool

## MARC Highly Reliable HW Components: AT697

htpp://www.atmel.com

SPARC V8 High Performance Low-power 32-bit Architecture

- LEON2-FT 1.0.13 compliant
- 8 Register Windows
- Advanced Architecture: On-chip Amba Bus, 5 Stage Pipeline, 16 kbyte Multi-sets Data Cache, 32 kbyte Multi-sets Instruction Cache
- On-chip Peripherals: Memory Interface, PROM Controller, SRAM Controller, SDRAM Controller
- Timers: Two 24-bit Timers, Watchdog Timer, Two 8-bit UARTs, Interrupt Controller with 4 External Programmable Inputs, 32 Parallel I/O Interface, 33MHz PCI Interface Compliant with 2.2 PCI Specification, Integrated 32/64-bit IEEE 754 Floating-point Unit
- Fault Tolerance by Design: Full Triple Modular Redundancy (TMR), EDAC Protection, Parity Protection, Debug and Test Facilities, Debug Support Unit (DSU) for Trace and Debug, IEEE 1149.1 JTAG Interface, Four Hardware Watchpoints, •Speed Optimized Code RAM Interface8, 16 and 40-bit boot-PROM (Flash) Interface Possibilities

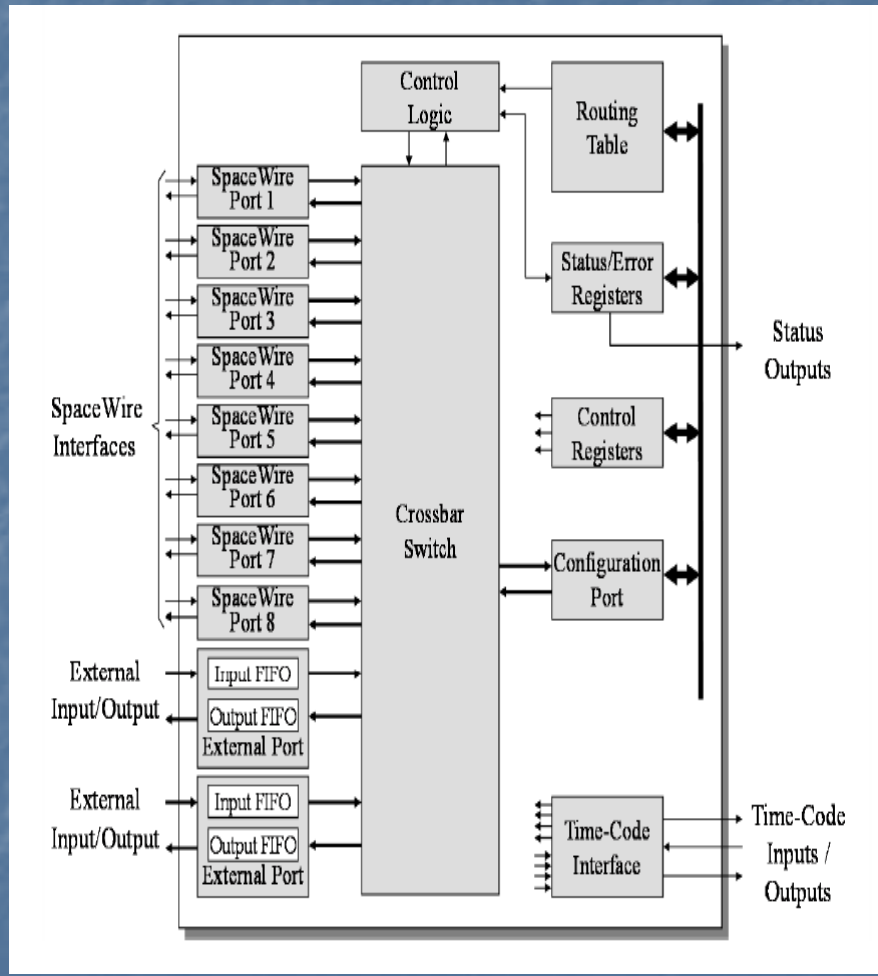


## *MARC Highly Reliable HW Components: AT7913*

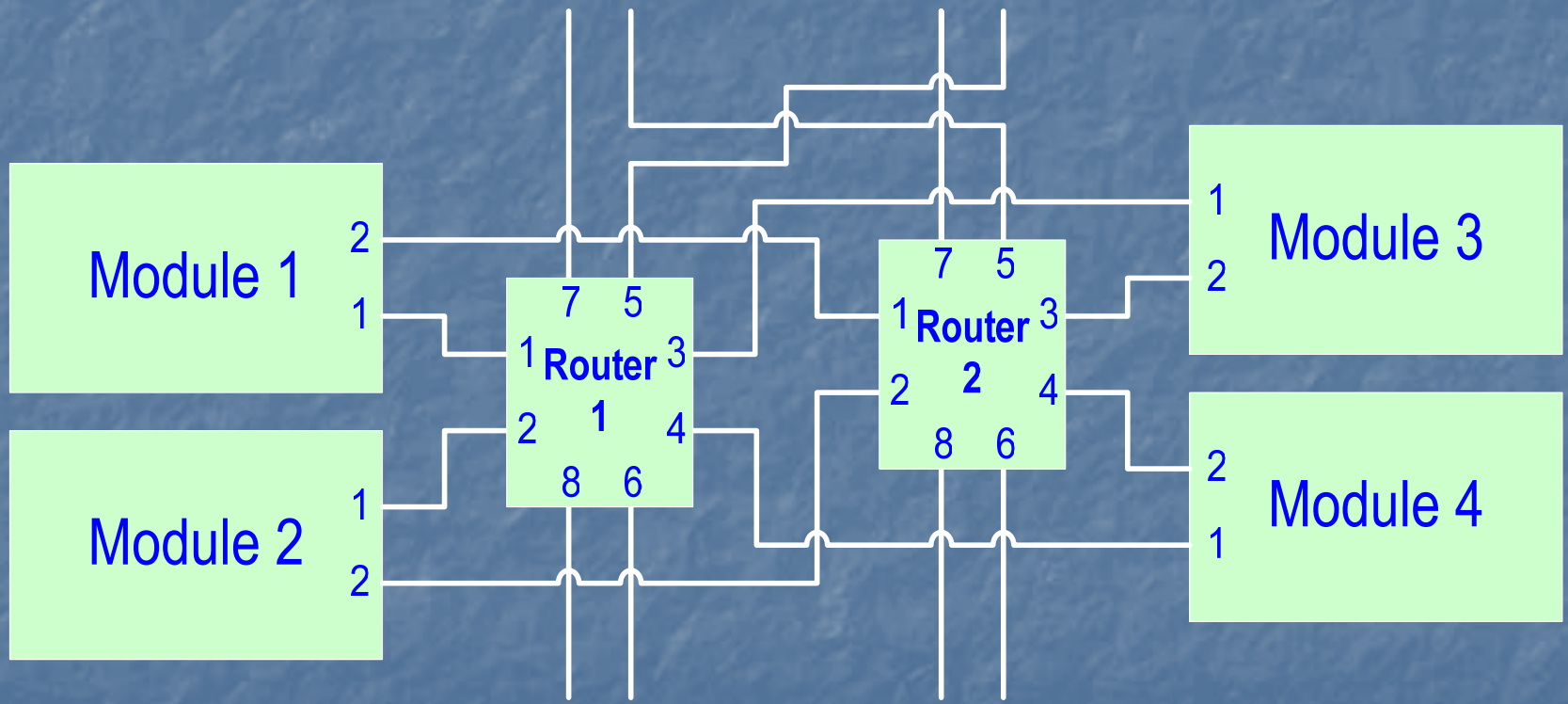
htpp://www.atmel.com

SpW-10X SpW router implements the following block functional:

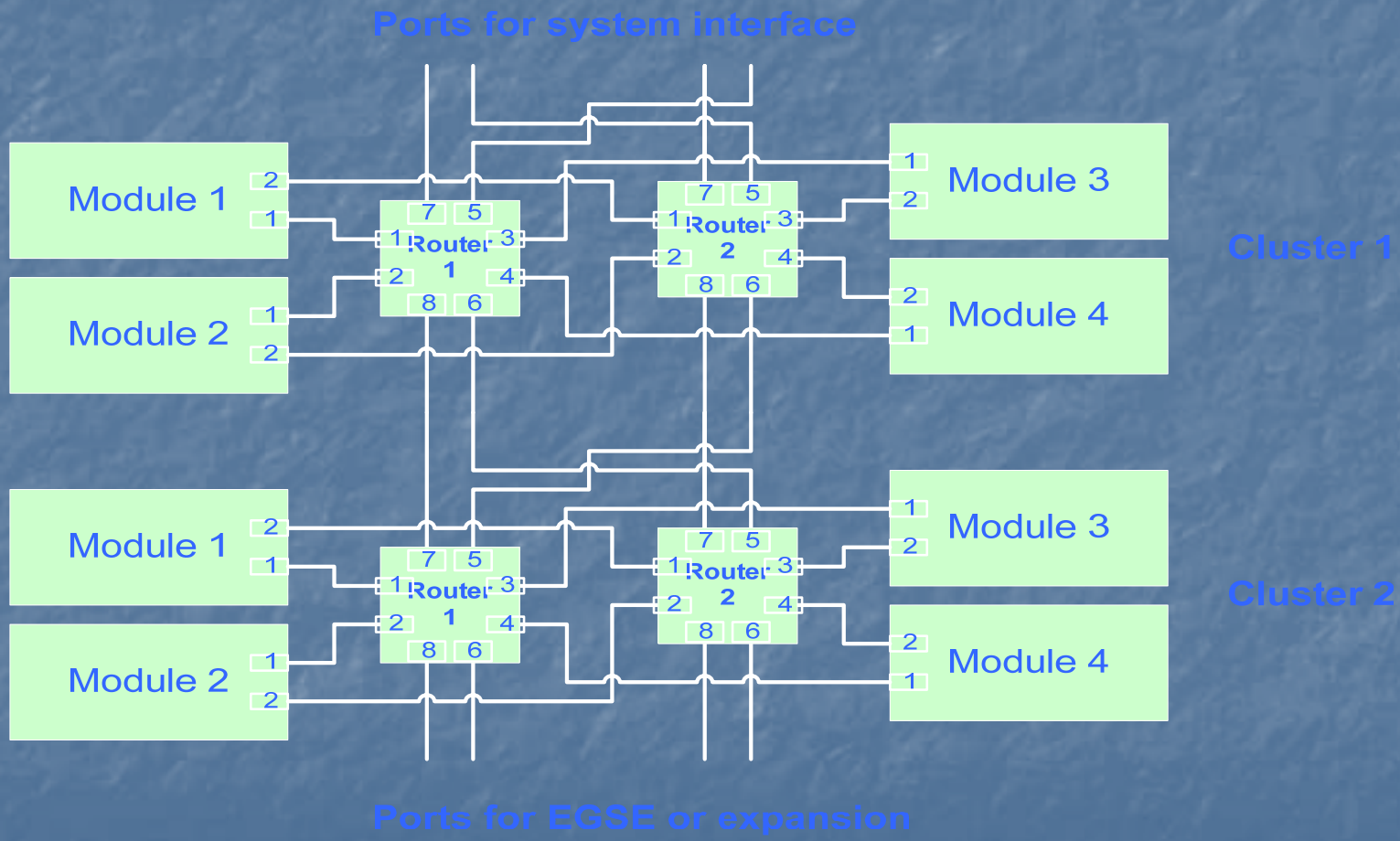
- 8 SpW bi-directional serial ports.
- 1 external parallel input/output ports ( input FIFO & output FIFO).
- One crossbar switch
- An internal configuration port accessible via the crossbar switch from the external parallel input/output port or the SpW input/output ports.
- A routing table accessible via the configuration port.
- Control logic to control the operation of the switch
- Control registers
- An external time-code interface comprising tick_in, tick_out and current tick count value.
- Internal status/error registers.
- Watchdog timers on all ports.
- Internal status/error registers
- External status/error signals.



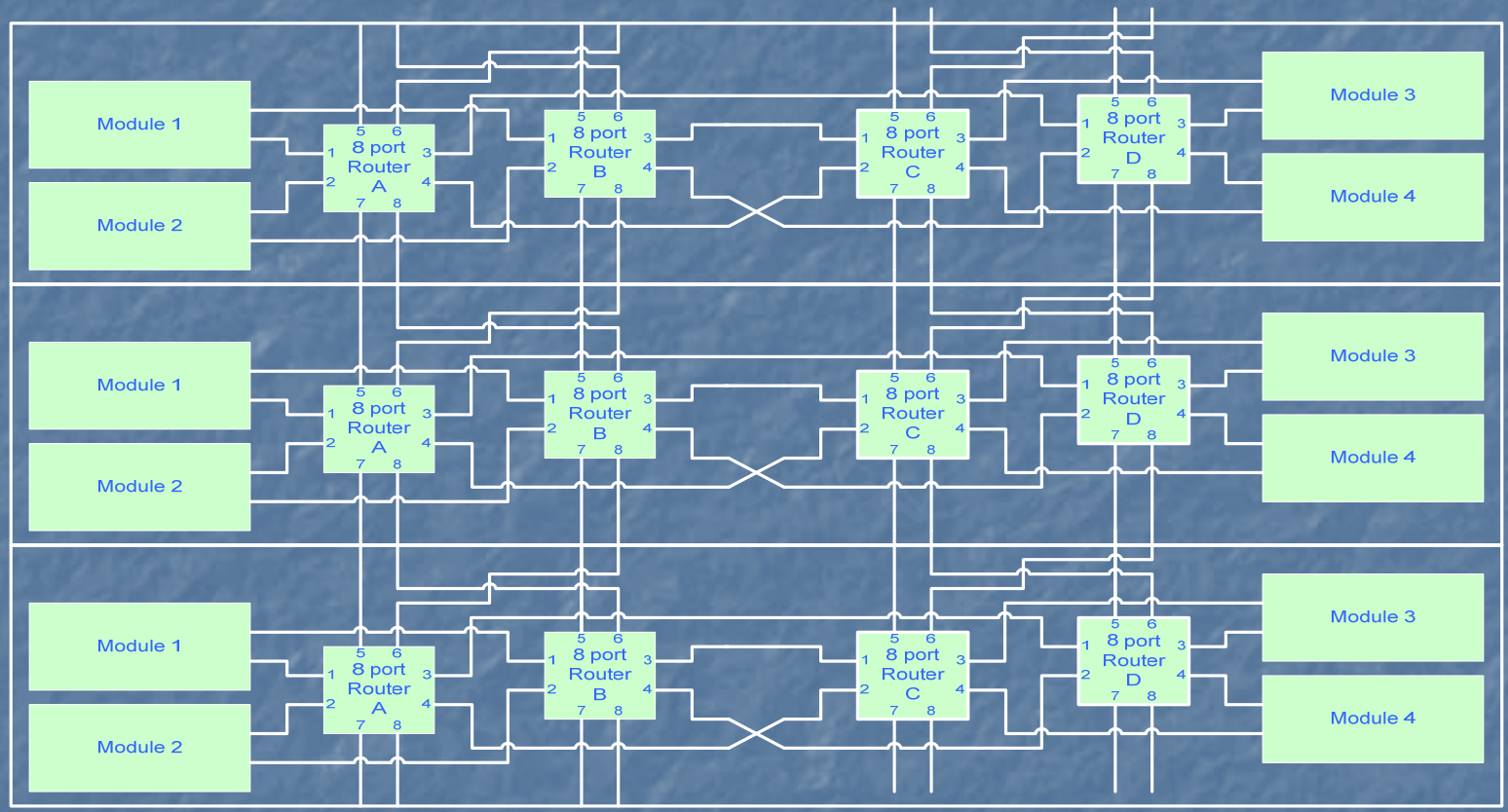
## MARC SpW High Flexibility Cluster (SpW-HFC)



# SpW-HFCs Vertical Assembly



# SpW-HFCs Lateral Assembly



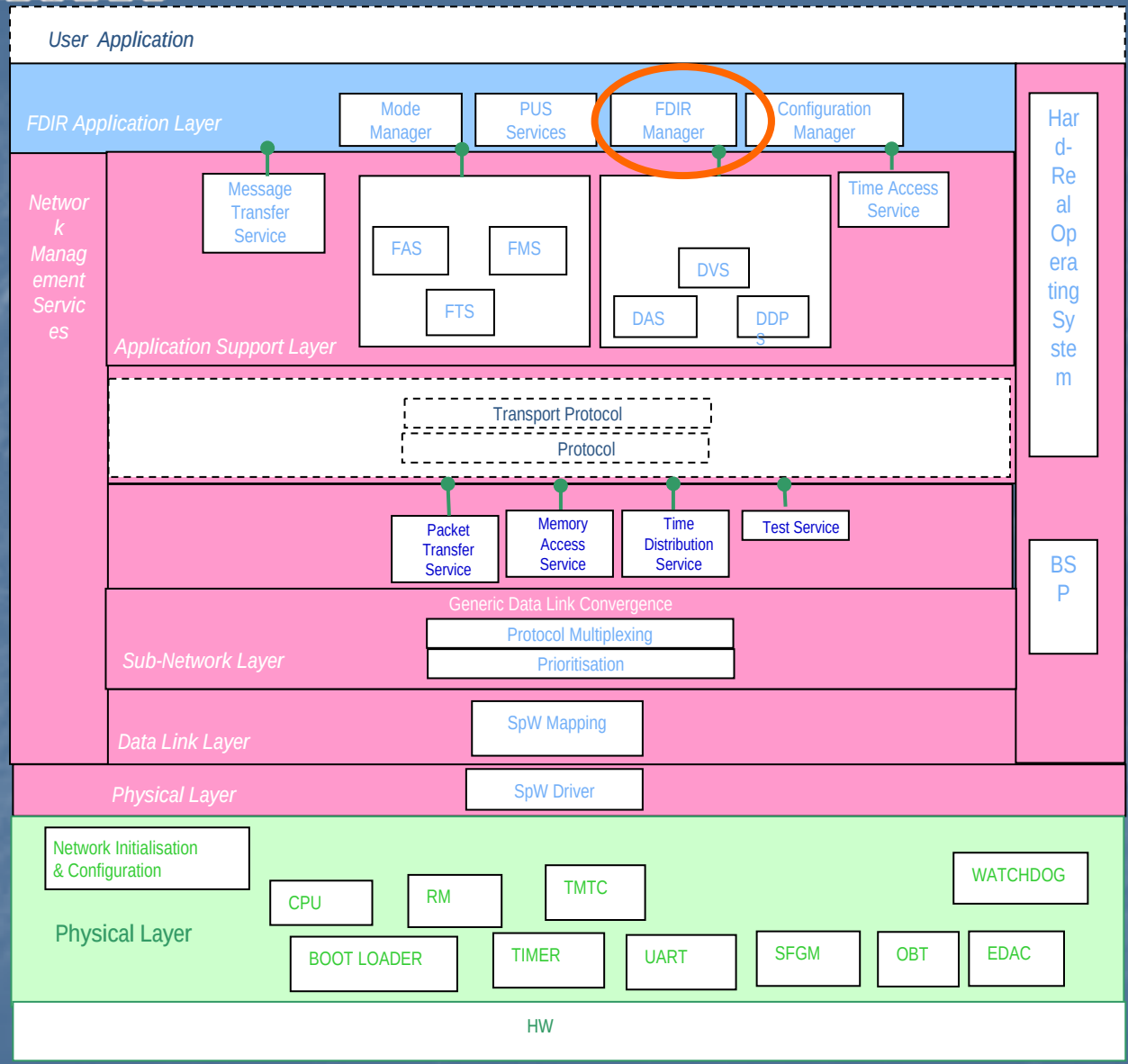
RMAP (ECSS-E-50-11): Read, Write and Read-Modify-Write Operation for Remote Memory Access.

- All MARC modules have at least two SpW interfaces (Nom & Red) with ESA-RMAP-IP-Core (Initiator and/or Target)
- CCM access memory banks in:
  - Local Memory,
  - Another CCM,
  - Mass Memory.
- RMAP simple memory access capability is very efficient to handle Prime CCM failure and permits context saving memory access by the redundant CCM.
- Memory access is possible without any system reconfiguration.
- At Configuration and Initialisation phase: Possibility to distribute boot SW, context saving and data mission storage to the available memory banks of MARC.

# MARC FT-Design: SW Building Blocks

**GenFas SW:**

- ✓ based on CCSDS-SOIS Architecture
- ✓ using AT697 processor of CCM
- ✓ SpW-RT Protocol




**CCSDS-SOIS layered SW architecture**

MARC FDIR Manager is GenFas embedded SW (SOIS standard).

Hierarchical FDIR management architecture:

- ✓ Failover strategies or switching to a safe computing mode
- ✓ Fault Containment Regions (FCRs) possibly confined to components, modules and/or clusters,
- ✓ Faults autonomously handled or escalated at each level in the hierarchy (with module and system as the minimum defined levels).

FDIR Manager uses tables (FDIR-Tabs):

- ✓ Define the initial system configuration
- ✓ Define alternative configurations (i.e. to be used in response to identified failure(s) of a SpW link, router or system module).

FDIR-Tabs are generated by an analysis tool:

Analysis Tool is conceived to run off-line during the architectural design phase of MARC system.

MARC FDIR Manager Strategy performs 4 principal functions:

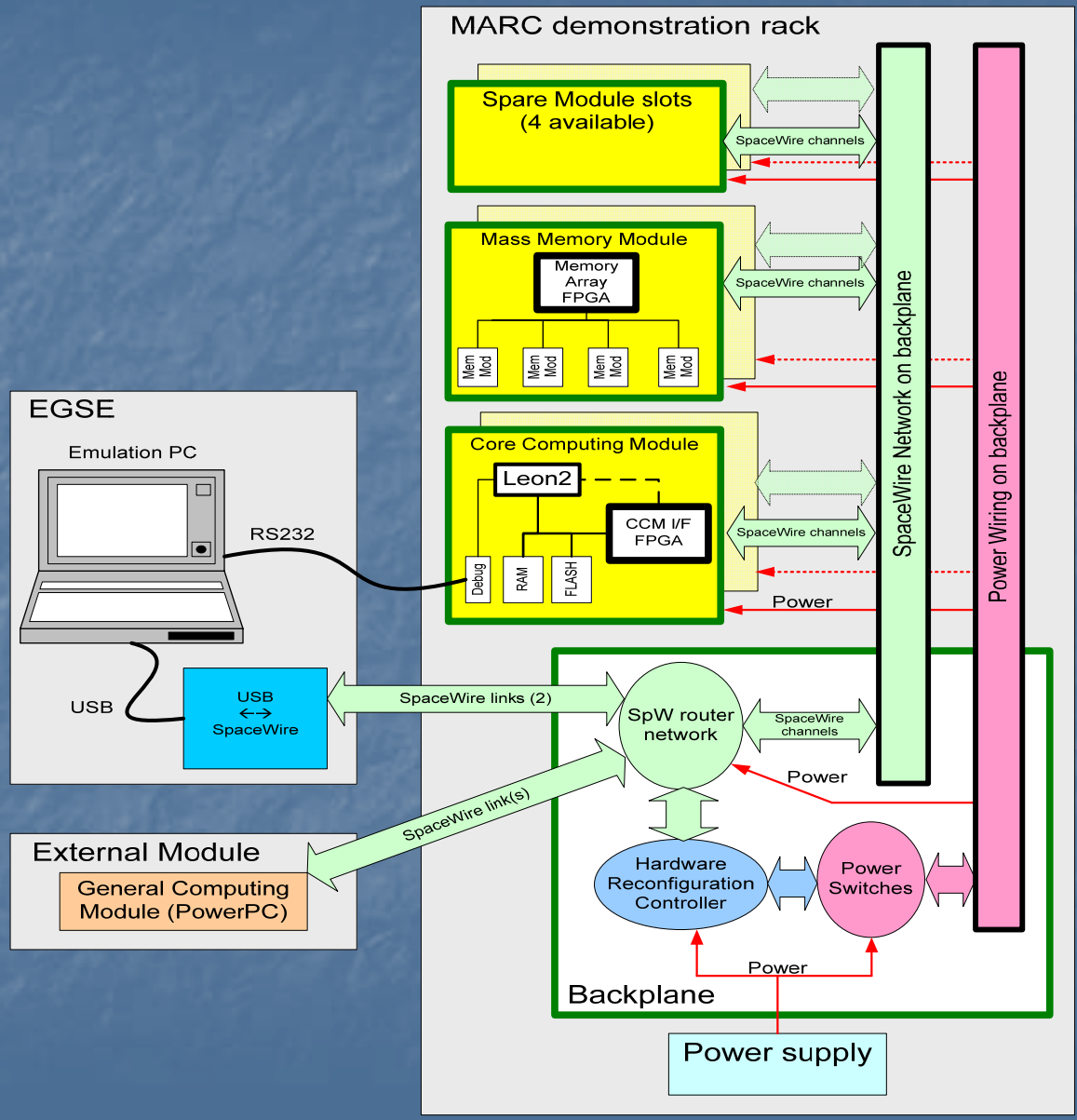
- *Health Check Monitoring*:
  - Regular message MHS solicited (pulled) or unsolicited (pushed).
  - Each component has a programmable time period (T*hs*) associated with it.
- *Fault Detection*:
  - Content of the messages themselves (e.g. 'I am OK')
  - Arrival time T*hs* of expected HSM: T0, T1 and T2 (where T0<T1<T2), HSM can be classified as **Valid**, **Missing, Late** or **Lost**.
- *Fault Diagnosis and Identification:*
  - FDIR Manager will log the event in its Registered Fault Log.
  - The Fault Diagnosis Procedure is then initiated and during this process, statuses are assigned by the FDIR Manager to the fault events **Transient, Intermittent** and **Permanent.**
- *Fault Recovery:*
  - a) Initial Recovery Procedure
  - b) *Router Recovery Procedure*
  - c) *Node Recovery Procedure*

# MARC Demonstrator

MARC First Prototype:

MARC Demonstrator based on 2 clusters with the following features:

- ✓ Backplane (i.e. SpW_10X routers and HRC)
- ✓ Core Computing Module: CCM based on AT697
- ✓ Solid Mass Memory Module: SMMM including file system
- ✓General Computing Module: GCM based on COTS Power-PC I/Fs)
- ✓Emulation of I/O: IOM & TMTC module via PCs and specific SpW EGSEs.
- ✓GenFas software including FDIR Manager



# MARC Present Status Activity

| | |
|---|---|
| Phase 1 Objectives | ◆ MARC Specification<br>◆ MARC Preliminary Design (at architectural level)<br>◆ MARC feasibility Plan |
| Phase 2 Objectives | ◆ MARC Detailed Design<br>◆ MARC Manufacturing<br>◆ Test and characterization |

Phase 1 has been successfully finished
Phase 2 is on going

# Conclusion

- MARC design is relying on BBs ensuring its robustness and its high flexibility.
- MARC preliminary design is consistent with previous requirements.
- Preliminary design of MARC is fulfilling many future ESA mission OBCS needs.
- The time and the cost needed to adapt MARC Demonstrator system to new mission appears to be significantly shorter than a development of a new system.
- General requirement related to the assessment of MARC overall performances compared to existing solutions: MARC Demonstrator will be the first MARC System prototype allowing performance assessment.
- MARC Demonstrator will be available at the level of Elegant-Bread Board (as defined by ECSS-E-10-02A).

# MARC Mini-Project Partners

ARIGATO