CENTRE NATIONAL D'ÉTUDES SPATIALES

SERESSA 2015

Puebla – Mexico
Nov. 30 – Dec. 04, 2015

System Hardening *and Real Space Applications*

Michel PIGNOL

CNES

DCT/TV/IN
18 avenue Edouard Belin
31401 Toulouse Cedex 9 - FRANCE

michel.pignol@cnes.fr ***http://www.cnes.fr***

Motivation
Rationale for fault-tolerant architectures in the space domain

- **Up to now, space computers are mainly developed with rad-hard ICs**

- **Mainly for performance reasons (not for cost reasons), commercial electronic integrated components (COTS ICs) will probably be more and more used**
  - **For microprocessors (µP), the performance gap is around 50 (average value)**
    - **LEON2 = 100 MIPS peak PowerPC7448 = 5100 MIPS peak**
  - **This gap is growing**
    - **PowerPC is superscalar, not LEON2**

- **Due to the SEE sensitivity of COTS, they must be protected by fault-tolerant mechanisms or architectures**

- **SEE protections = high cost / planning overheads**
  => it is important to assess carefully the safety/availability requirements of the project to select the optimal fault-tolerant solution
  - **Such solutions could range from very simple mechanisms having limited error detection/recovery capabilities to complete protection with FT archi.**

OUTLINE

1 – INTRODUCTIVE PART

- Avionics architecture of a satellite
- Good practices to face SEE

2 – ARCHITECTURE AND SYSTEM PROTECTIONS

- 2-A – FDIR overview
- 2-B – Processing units
  - Time replication
  - Structural duplex
  - Triplex / Quadruplex
  - Micro-synchronized triplex
  - Fault-tolerant trade-off with analysis of theoretical case studies

3 – REAL CASE STUDIES

- ATV, the ESA Automated Transfer Vehicule
- MYRIADE, the CNES micro-satellite family
- CALIPSO, a Franco-American mini-satellite
- REIMEI (INDEX), a Japanese small satellite

4 – CONCLUSION

Acronyms

CNES Centre National d'Etudes Spatiales, the French Space Agency
ESA European Space Agency

ADC	Analog to Digital Converter
Acq	Acquisition
ALU	Arithmeric and Logic Unit
ATV	Automated Transfer Vehicule
Cmd	Command (actuation)
Cntl	Control
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CTXT	Context (software variable)
DRAM	Dynamic RAM
DSP	Digital Signal Processor
EDAC	Error Detection And Correction
FDIR	Fault Detection, Isolation and Recovery
FT	Faut-Tolerant
FTC	Fault-Tolerant Computer
GIPS	Giga Instructions Per Second
IC	Integrated Circuit
I/O	Input/Output
ISS	International Space Station
NG	Next Generation
OBC	On-Board Computer
PARAM	Parameter (software variable)
PF	PlatForm (of a satellite)
PL	PayLoad (of a satellite)
R/W	Read/Write
TC	TeleCommand
TM	TeleMetry
Tx/Rx	Transmitter/Receiver
µP	MicroProcesseur
µSL	Micro-Satellite
VLIW	Very Long Instruction Word (superscalar DSP having several execution units working in parallel)
WD	WatchDog
wrt	With Regard To
DMT	Duplex Multiplexed in Time (time replication at task level, CNES architecture)
DT2	Double Duplex Tolerant to Transient (mini structural duplex at task level, CNES architecture)
N-MR	N-Modular Redundancy
DMR	Double-MR = Duplex
TMR	Triple-MR = Triplex
QMR	Quad-MR = Quadruplex
MBU	Multiple Bit Upsets
SEE	Single Event Effect
SEFI	Single Event Functional Interrupt
SEL	Single Event Latch-up
µSEL	Micro latch-up
SET	Single Event Transient
SEU	Single Event Upset
TID	Total Ionizing Dose

1 – INTRODUCTIVE PART

© CNES / ISRO

© cnes - labo Photon/M Regy

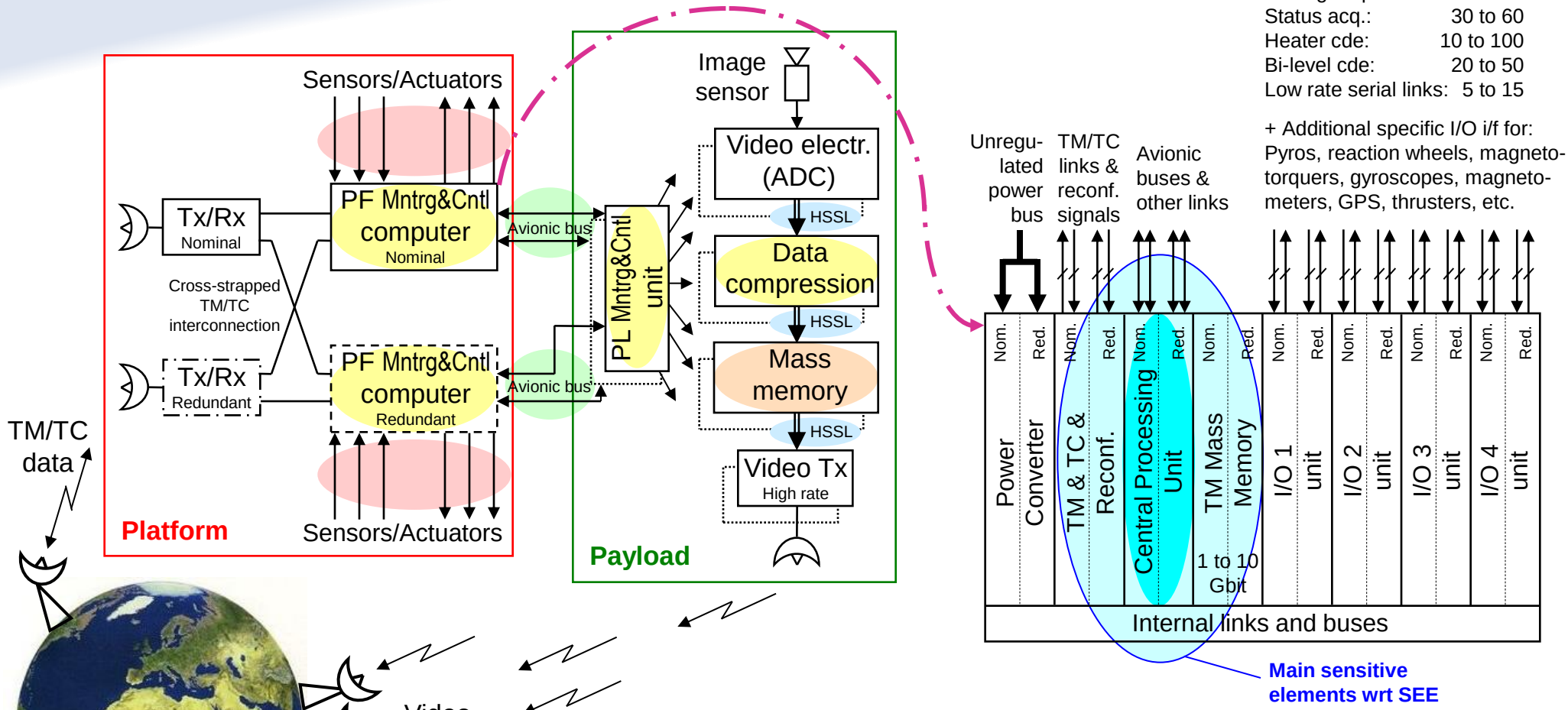
MEGHA-TROPIQUES: a French / Indian mission to improve our knowledge on the tropical climate system; launched in 2011

Avionics architecture of a satellite

Nb of input/output interfaces for small/large satellites:

Thermistors acq.:	30 to 200
Analog acq.:	30 to 100
Status acq.:	30 to 60
Heater cde:	10 to 100
Bi-level cde:	20 to 50
Low rate serial links:	5 to 15

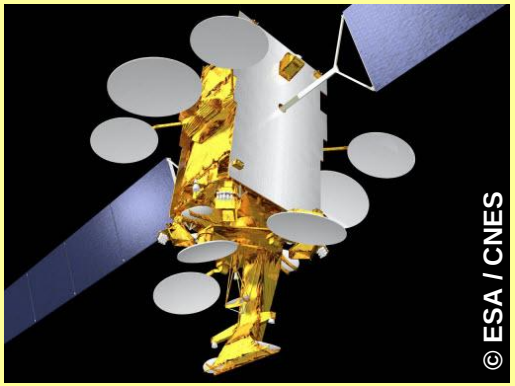
+ Additional specific I/O i/f for: Pyros, reaction wheels, magneto-torquers, gyroscopes, magneto-meters, GPS, thrusters, etc.



Ex. for Mntrg&Cntl computers

Usual budgets for small satellites
No redundancy
Volume = 3,3 litres / 5 boards
Mass = 3 kg
Power dissipation = 6 W

Usual budgets for large satellites
Nominal + Redundant units
Volume = 30 litres / 15 boards
Mass = 19 kg
Power dissipation = 55 W



ALPHABUS, a family of European Telecom satellites with a common platform from EADS ASTRIUM and THALES ALENIA SPACE

Max 8800 kg
Max 18000 W

The 1st launch is ALPHASAT in 2013

Launched in 2002
3000 kg
2400 W
5.7x3.1x3.1 m
2x60 km swath
2.5 m resolution

SPOT 5 for Earth observation

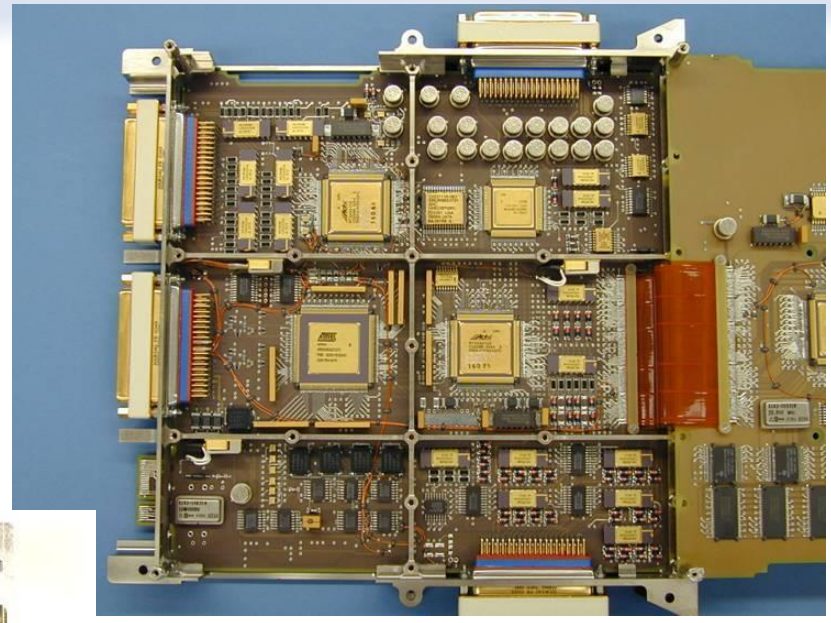
Launched in 2005
6000 kg
14000 W
7x2.9x2.3 m
45 m solar arrays
10 m diam. antenna

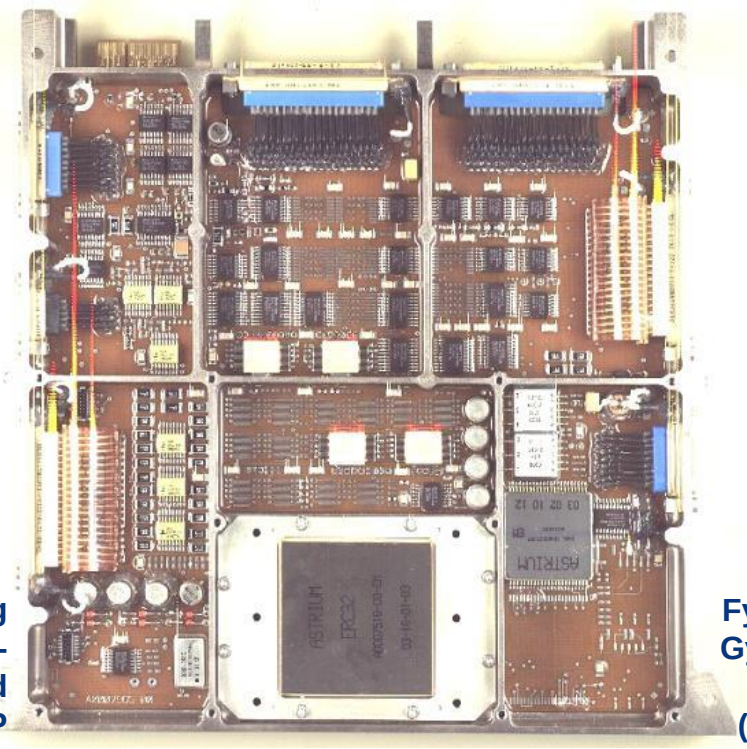
INMARSAT 4-F1 for mobiles-to-mobiles telecommunications

AIRBUS DEFENCE & SPACE computers

GSTB-V2 computer (Galileo System Test Bench; proposal for Galileo satellites)

TM & TC & Reconf. board, including 3 ASTRIUM ASICs:
- TC processing and reconf.
- TM formatting and routing
- Storage control for reconf.





CPU board using ASTRIUM Multi-Chip-Module (2003) based on ERC32SC space µP

Fyber Optic Gyro Electr. Module (I/O board)

© Courtesy of AIRBUS DEFENCE & SPACE

SEE – Effects of radiation on digital parts

SEE : Single Event Effect

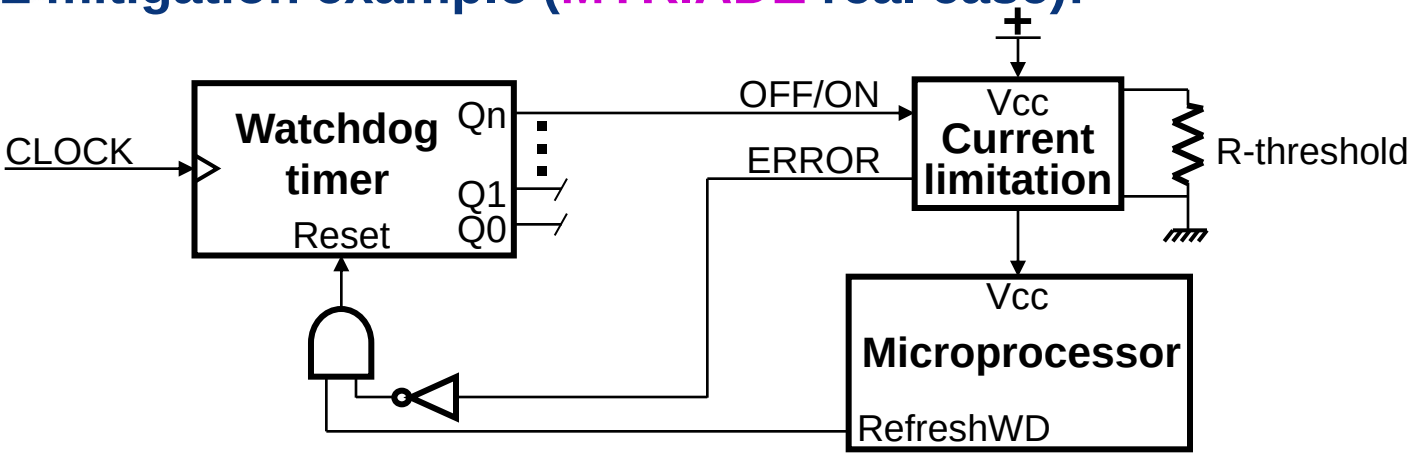
SEE in digital ICs = SEL + µSEL + SEU + MBU + SET + SEFI

- SEE concerns all effects due to a single particle
- SEL/µSEL – (micro) Single Event Latch-up
  - Local short-circuit
  - Detection: loss of functionality or over-consumption / Protection: power-cycling
  - It is a good practice to avoid components which are sensitive to SEL
  - And if not possible, to limit their usage and to protect them with adequate solutions
- SEU/MBU – Single Event (Multiple Bit) Upset / SET – Single Event Transient
- SEFI – Single Effect Functional Interrupt
  - The component is put in a blocking state and a reset is not always capable to bring it back into an operational state
  - Detection: loss of functionality (as for SEL)
  - Protection: reset (optional but recommended) and/or power cycling (mandatory)
- *Goal of faul-tolerant architecture protections*
  - *Thanks to DSM technos, more and more COTS parts are compliant with TID (Total Ionizing Dose) and SEL space constraints*
  - *But all digital COTS components are sensitive to transients and upsets*

=> The presentation targets SEU / MBU / SET / SEFI mitigation, mainly on µP

- **An ingenious SEL mitigation example (MYRIADE real case):**

minimizing the number of parts and, nevertheless, implementing both detection and both mitigation methods



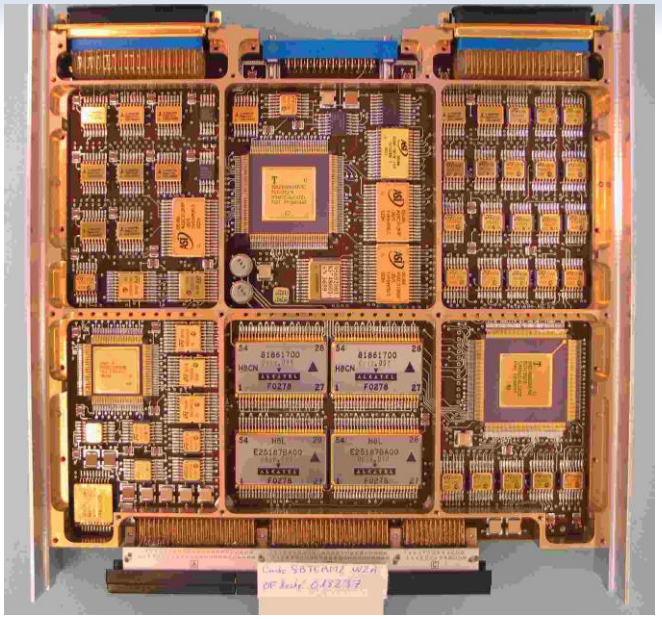
- **Whatsoever the 'detection' method is, it is a good practice to have a gradual 'recovery' process based on several levels, for instance:**
  - **First attempt following a detection: a quick 'standard' recovery (i.e. without reset) is tried (in case of simple effect of an SET/SEU/MBU)**
  - **Second attempt: if the first attempt is not successful, a reset of the computer is done (in case of more complex effect of an SET/SEU/MBU or in case of SEFI)**
  - **Third attempt: if the computer still does not become operational, then a power supply cycling is done (in case of SEFI or SEL)**
  - **Such a multi-level recovery process is implemented on CNES MYRIADE micro-satellite: See Section "3 – Real Case Studies"**

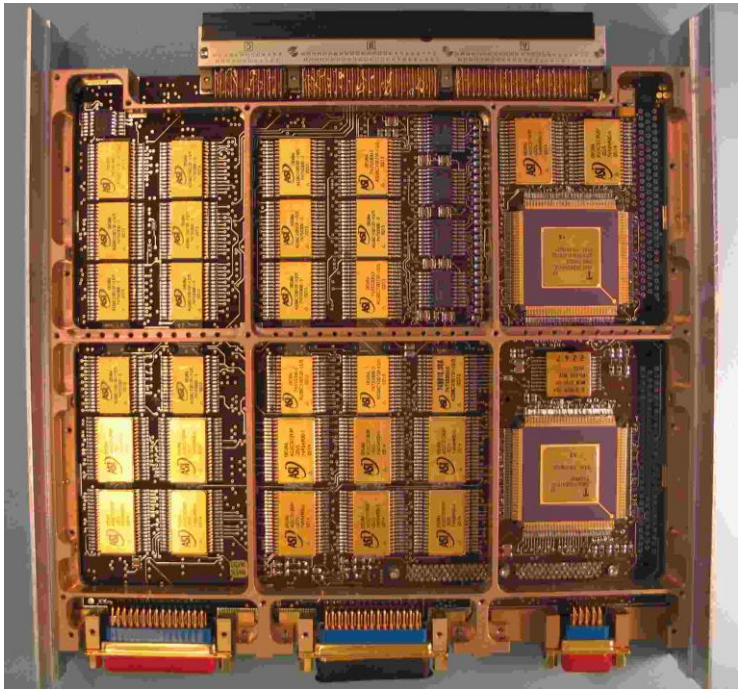
THALES ALENIA SPACE computers

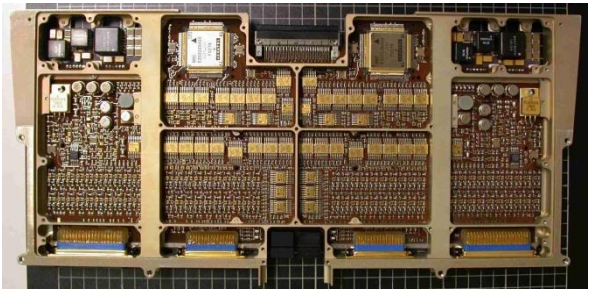
SMU-V1 computer (Satellite Management Unit; platform computer for SpaceBus4000 Telecom family satellites and Globalstar2 satellite)

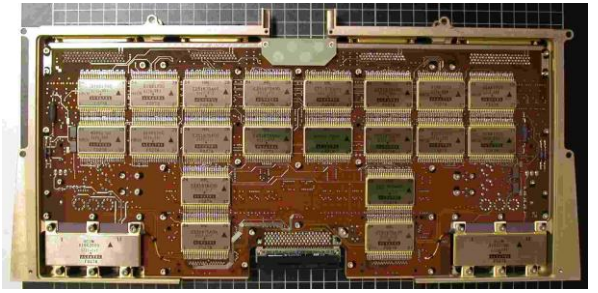
TM & TC & Reconf. board, including 2 THALES ASICs:
- TC processing and reconf.
- TM formatting and routing

and including 4 THALES hybrids for generating command signals







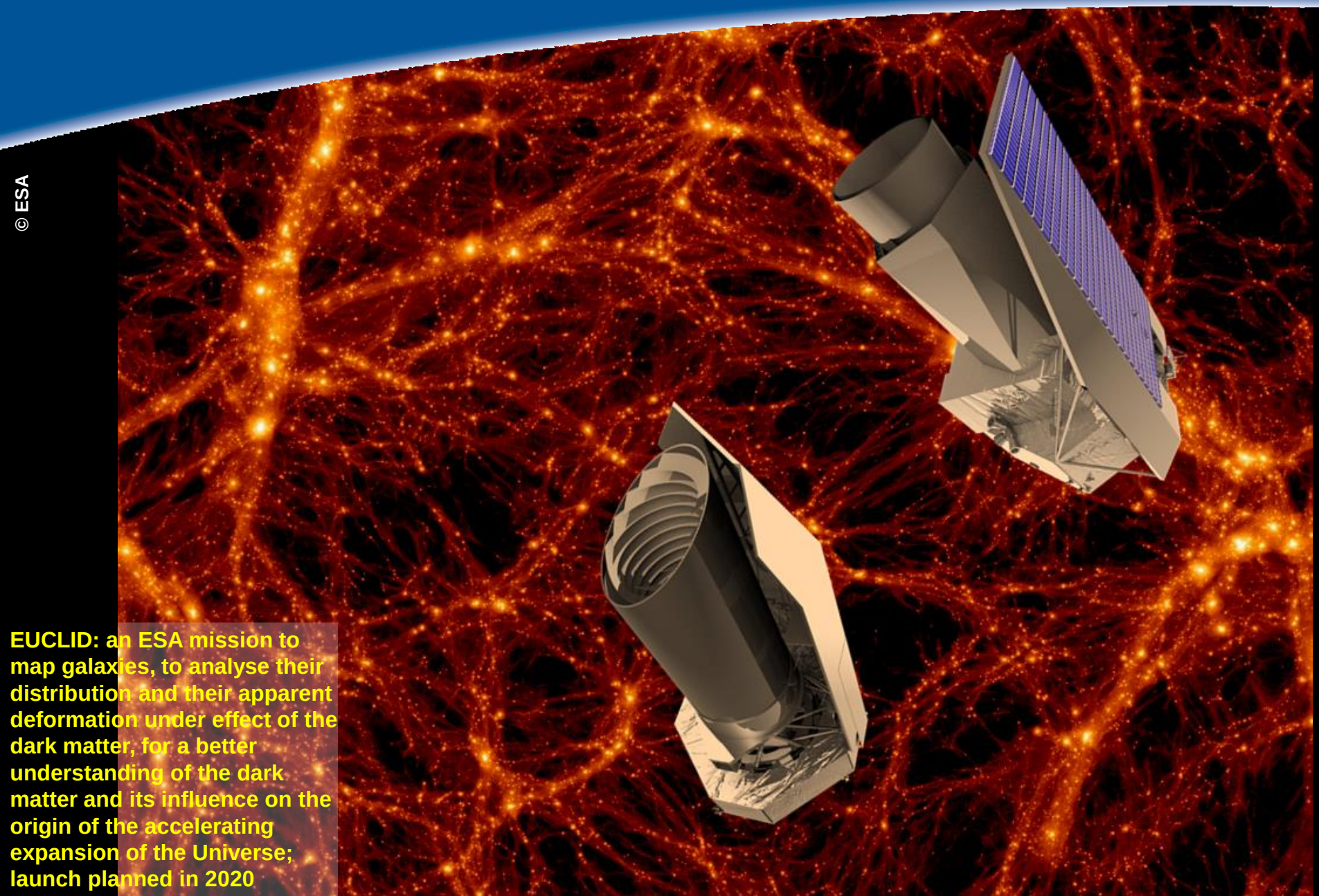


© Courtesy of THALES ALENIA SPACE

CPU board using ATMEL ERC32SC space µP and COPRES THALES ASIC

Satellite Distribution and Interface Unit for Telecom satellites (I/O board)

2 – ARCHITECTURE AND SYSTEM PROTECTIONS



© ESA

EUCLID: an ESA mission to map galaxies, to analyse their distribution and their apparent deformation under effect of the dark matter, for a better understanding of the dark matter and its influence on the origin of the accelerating expansion of the Universe; launch planned in 2020

2-A – FDIR overview

The FDIR strategy – Fault Detection, Isolation and Recovery

- **Main objective of the FDIR strategy**
  - To keep the integrity of the satellite (i.e. its operational capability) in presence of anomalies
    - There is not an universal strategy, it is a case-by-case basis definition depending on the mission and on the considered faulty unit

- **Usual FDIR strategies when an anomaly is detected**
  - ✓ *"Satellite survival mode" = minimal mode allowing to keep at an acceptable level the electric pw, the internal temperature and the TM/TC link with the ground cntrl station*
  - Earth observation satellites: To pass in the survival mode and to leave to the ground control station the detection of the source of the anomaly then the selection of the best recovery strategy
  - Telecom satellites: To reconfigure the avionics architecture to try to passivate the anomaly in order to remain in operational mode as long as possible to comply with the availability requirements; to limit the survival mode usage to exceptional cases

  => Telecom satellites have an higher autonomy than Earth observation satellites

The FDIR strategy (cont.)

- **Recovery action when an anomaly is detected**
  - **Only few alarms are highly critical and directly start a recovery action**

    => examples of such critical alarms
    - **power falling down**
    - **software watchdog**
    - **Earth sensor alarm** for some missions

    => and associated recovery action in case of cold redundancy
    - switch-off nominal computer and nominal peripheral units
    - switch-on redundant computer and a mini. of redund. periph.
    - then start from scratch and put the spacecraft in "attitude acquisition & safe hold" mode

    sensors & actuators
  - **For all the other alarms, the general rule is "to try to confirm the alarm before starting a recovery action", thanks to the "anomaly filtering process"**
- **Some examples of the "anomaly filtering process"**
  - **Time redundancy at the system level**
    - when a task (thermal control, attitude and orbit control system, etc.) trigger an alarm during a given iteration, it is checked if the same alarm is still triggered during the next iteration(s) of this task
  - **Comparison between sensors to confirm an incoherent data**
    - coupling with dedicated algorithms of linked data issued from gyro sensors and from the star sensor
  - **Start a BIST (Built-In Self Test) into the intelligent sensor which have issued the incoherent data**

2-B – Processing units / Fault-tolerant architectures

General remarks

- Comparators and voters are usually implemented in **FPGA / ASIC** either **not sensitive to SEE** by design (D-FF triplication, etc. => thus COTS are usable) or implemented in radiation-tolerant technologies

- **Definitive failures** are mitigated through a **redundant unit**; this is the general way to process in the space domain where repair is not possible even with HiRel ICs

2-B – Processing units / Fault-tolerant (FT) architectures

- **Time replication**
  - **Time replication at instruction level**
    - Example of Time-TMR from SPACE MICRO Inc.
  - **Granularity for CNES FT architectures**
  - **Time replication at task level**
    - Example of DMT from CNES
- **Structural duplex**
  - Example of DT2 from CNES
- **TMR-Triplex & QMR-Quadruplex**
  - Examples issued from the SHUTTLE, GUARDS and ATV
- **Micro-synchronized triplex**
  - Example of SCS750 from MAXWELL Tech.
- **FT architectures trade-off**
- **Other methods and elementary protection mechanisms**

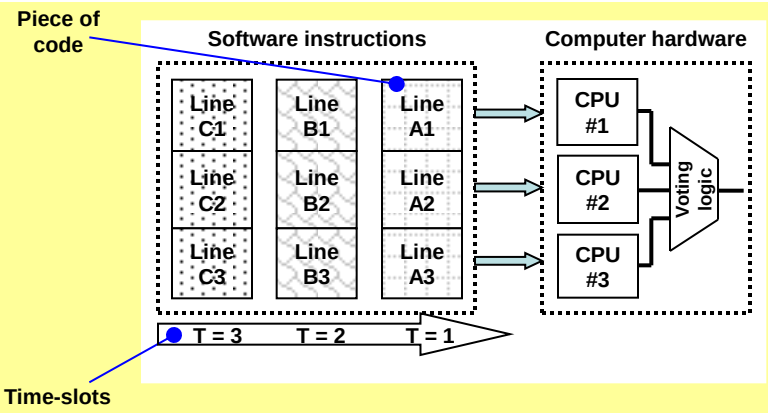
Time replication

- **Principle**
  - **No hardware replication => No extra recurring cost**
  - **The same software is processed N-times successively on the same CPU**
  - **Detection capability: the results of the different replicas are compared**

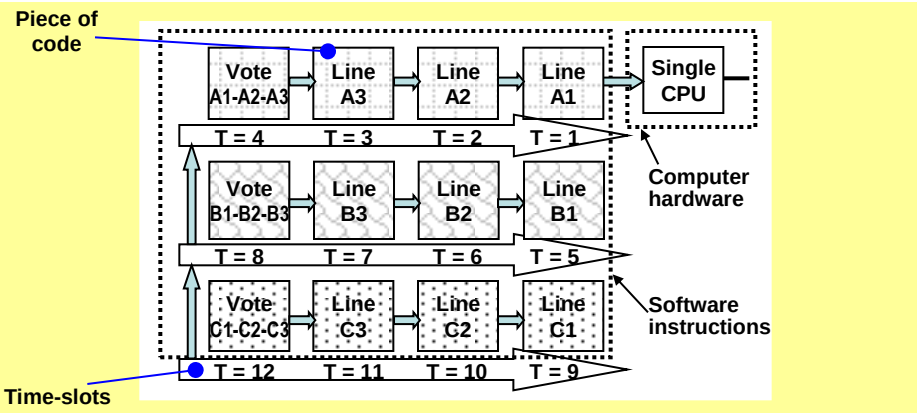
- **Time replication at instruction level**
  - **See the talk "Hardening at Software level" by Politecnico di Torino**

- Time replication at instruction level: real case example of an industrial development

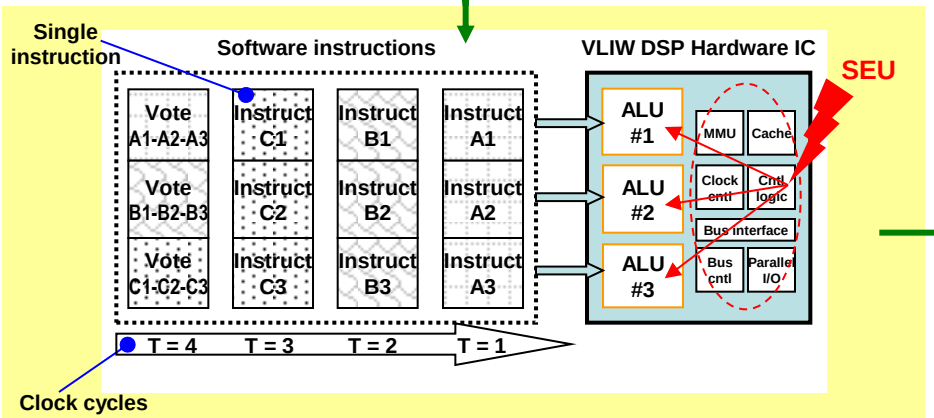
TTMR – Time-TMR (Space Micro Inc. – USA)



TMR architecture



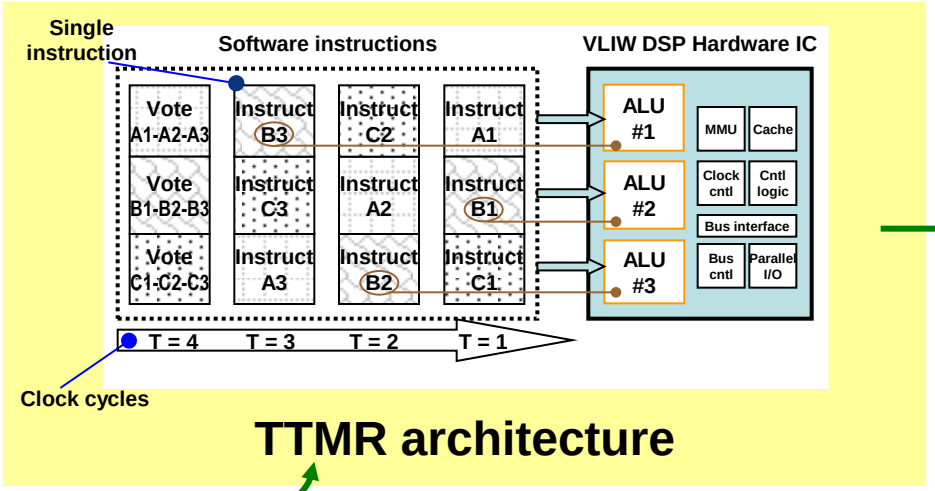
Time redundancy architecture



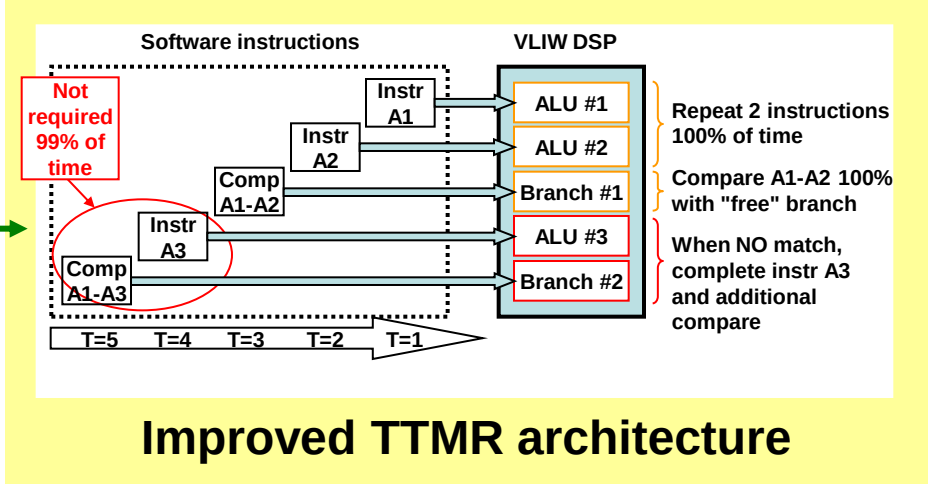
One TTMR possibility… with weakness

…/…

TTMR – Time-TMR (Space Micro Inc. – USA) (cont.)



TTMR architecture

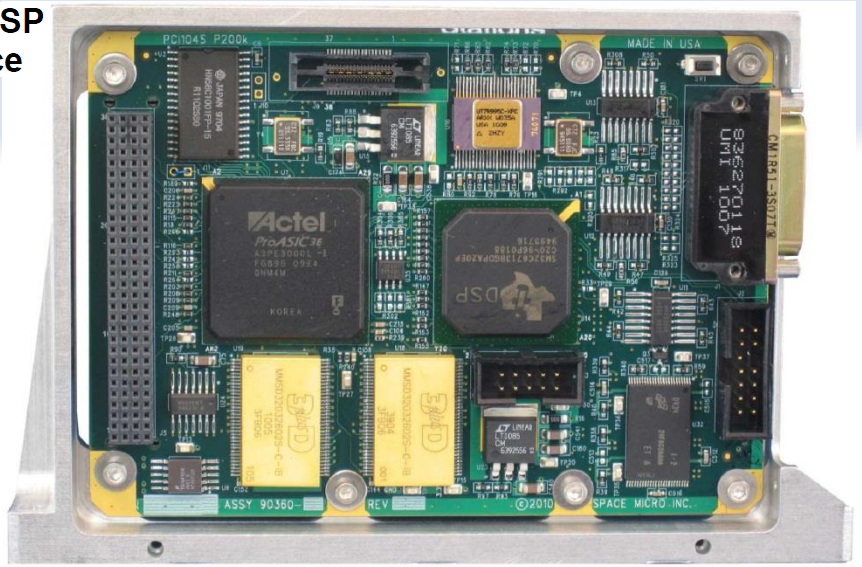


Improved TTMR architecture

.../...

© IEEE – Space Micro Inc.
(adapted from)

Proton200k™ DSP
Processor Slice



TTMR pros/cons (cont.)

- Proprietary architecture
  - Space Micro Inc. patent
- Dedicated to VLIW DSP (Very Long Instruction Word - Digital Signal Processor)
  - Given that the ALUs are generally speaking not all fully used, not too much time is lost due to the time replication
- The TTMR algorithm is coded into a "post-compiler"
  - All the know-how lies in the "post-compiler": instruction replication + vote insertion + instr.->ALU assignment + instr. reordering to avoid empty slots
  - The "post-compiler" must be developed for each targetted DSP
- The SEFIs are processed by a patented rad-hard watchdog circuit

Selected on USAF micro-satellites and ISS

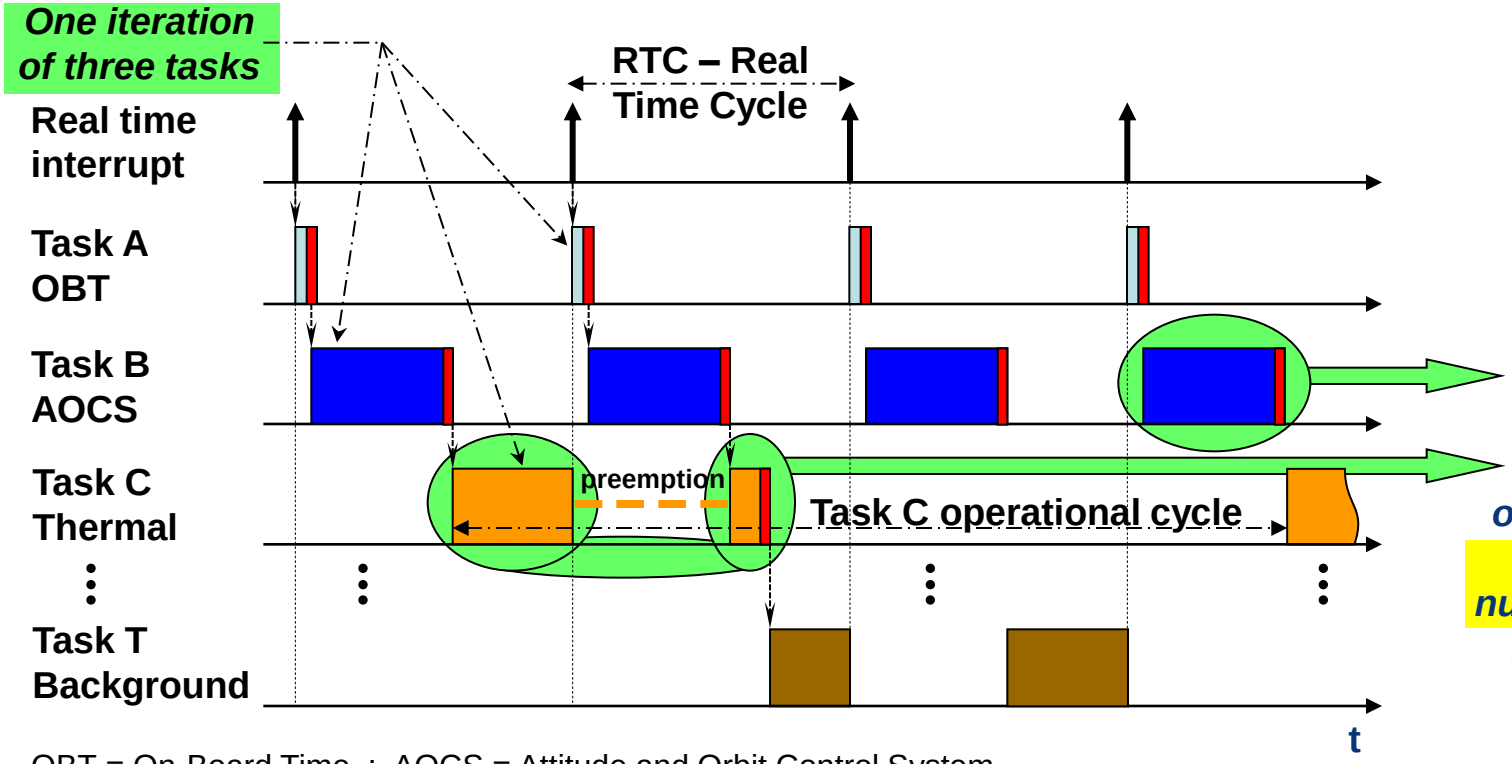
2-B – Processing units / Fault-tolerant (FT) architectures

- **Time replication**
  - **Time replication at instruction level**
    - Example of Time-TMR from SPACE MICRO Inc.
  -  **Granularity for CNES FT architectures**
  - **Time replication at task level**
    - Example of DMT from CNES
- **Structural duplex**
  - Example of DT2 from CNES
- **TMR-Triplex & QMR-Quadruplex**
  - Examples issued from the SHUTTLE, GUARDS and ATV
- **Micro-synchronized triplex**
  - Example of SCS750 from MAXWELL Tech.
- **FT architectures trade-off**
- **Other methods and elementary protection mechanisms**

Granularity for CNES DMT and DT2 fault-tolerant architectures

- Granularity impact deeply the definition and latency/overhead of FT mechanisms
- Coarse-grained granularity (macro-granularity) => task operational cycle
  - the checking procedure runs at the end of each iteration of each task
  - a low number of data to check => minimisation of overheads
  - the main fault-containment region

Simple example of a static scheduling of a flight software in a platform computer



Macro-granularity

only all output data (but not the huge number of local data) must be checked

OBT = On-Board Time ; AOCS = Attitude and Orbit Control System

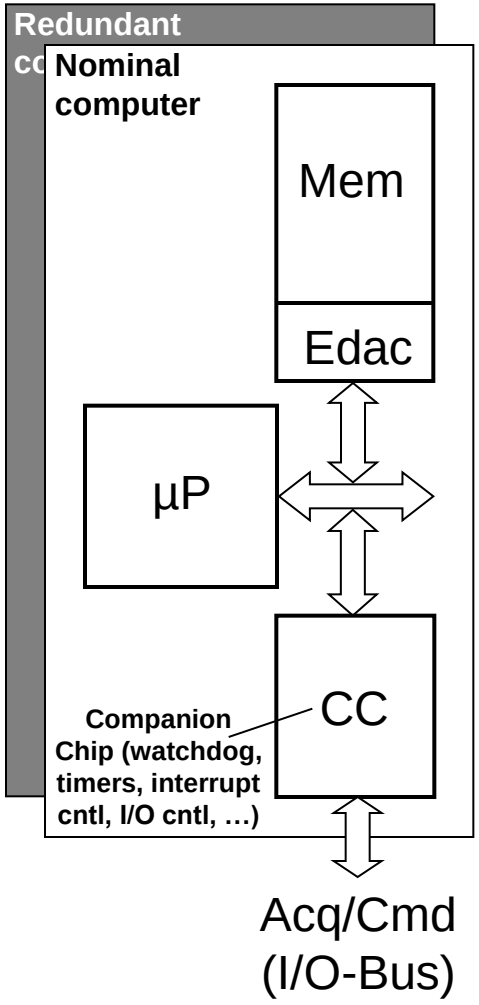
2-B – Processing units / Fault-tolerant (FT) architectures

- **Time replication**
  - **Time replication at instruction level**
    - Example of Time-TMR from SPACE MICRO Inc.
  - **Granularity for CNES FT architectures**
  - **Time replication at task level**
    - Example of DMT from CNES
- **Structural duplex**
  - Example of DT2 from CNES
- **TMR-Triplex & QMR-Quadruplex**
  - Examples issued from the SHUTTLE, GUARDS and ATV
- **Micro-synchronized triplex**
  - Example of SCS750 from MAXWELL Tech.
- **FT architectures trade-off**
- **Other methods and elementary protection mechanisms**

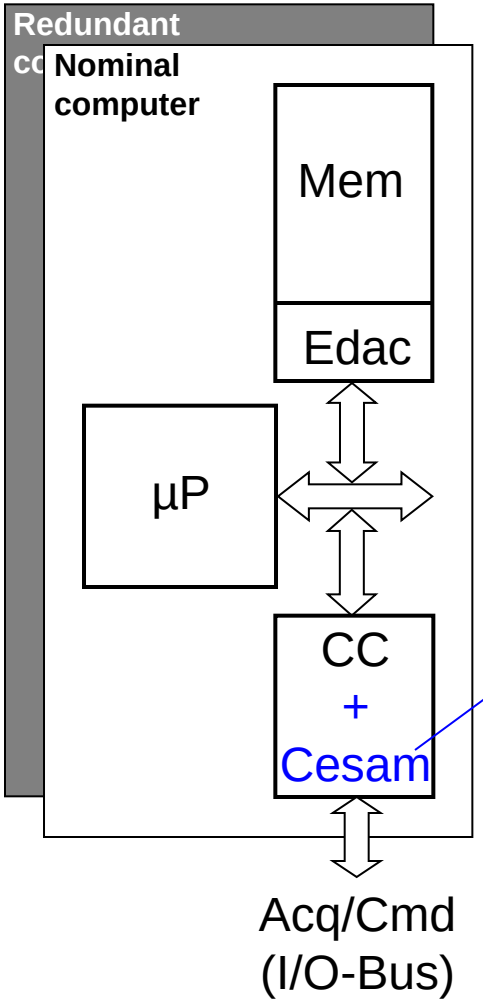
DMT - Duplex Multiplexed in Time (CNES – Fr)

Time replication duplex with recovery capability

PUC without DMT



Processing Unit Core with DMT

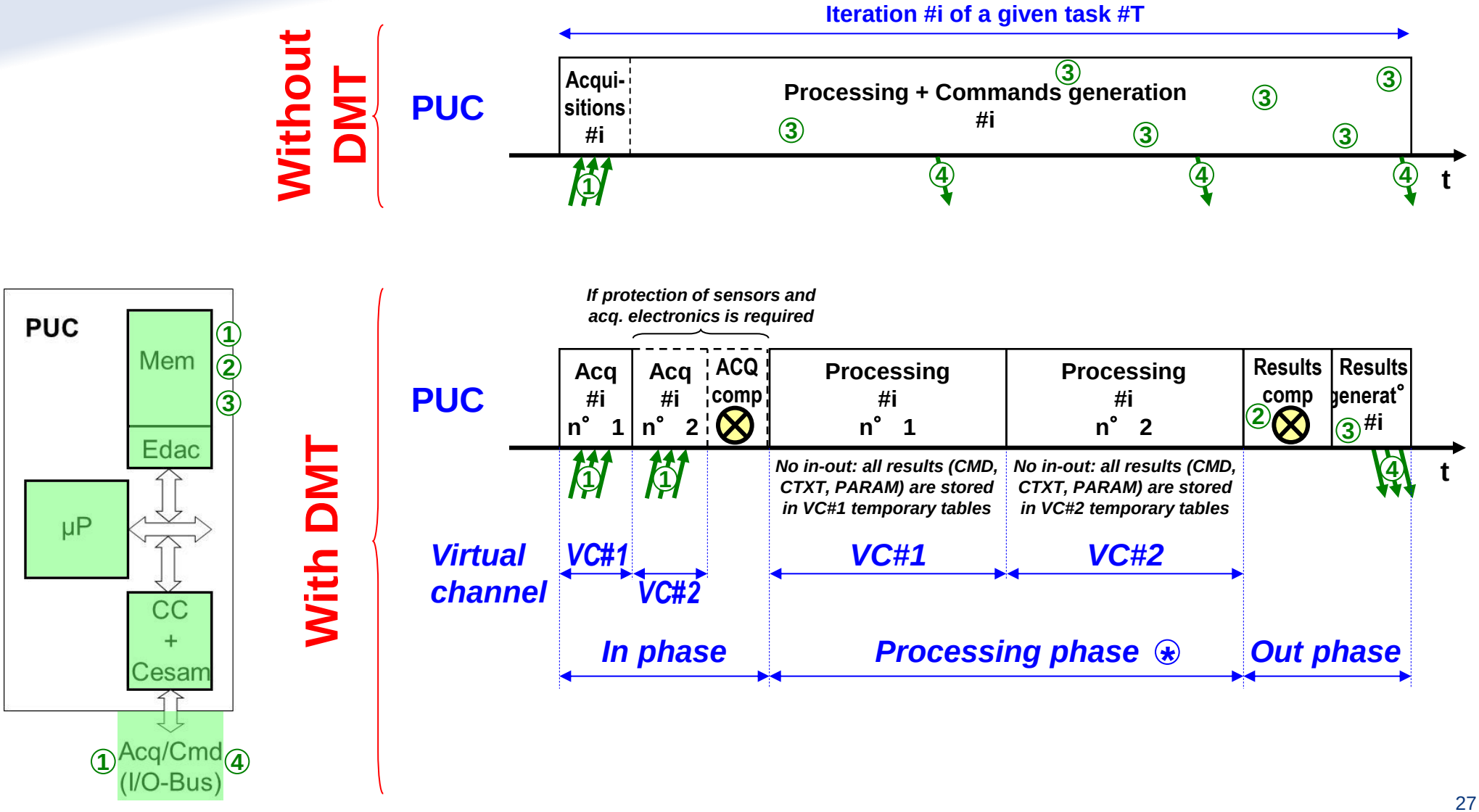


Redundant computer
Switched-off in cold-redundancy strategy

CESAM allows to segment the memory for monitoring of access rights:
1/ Avoid fault propagation between virtual channels
2/ Secure context data even if the µP is faulty

CESAM works as a Block Protection Unit (of a Memory Management Unit) with specific mechanisms

DMT – Scheduling and fault detection principle



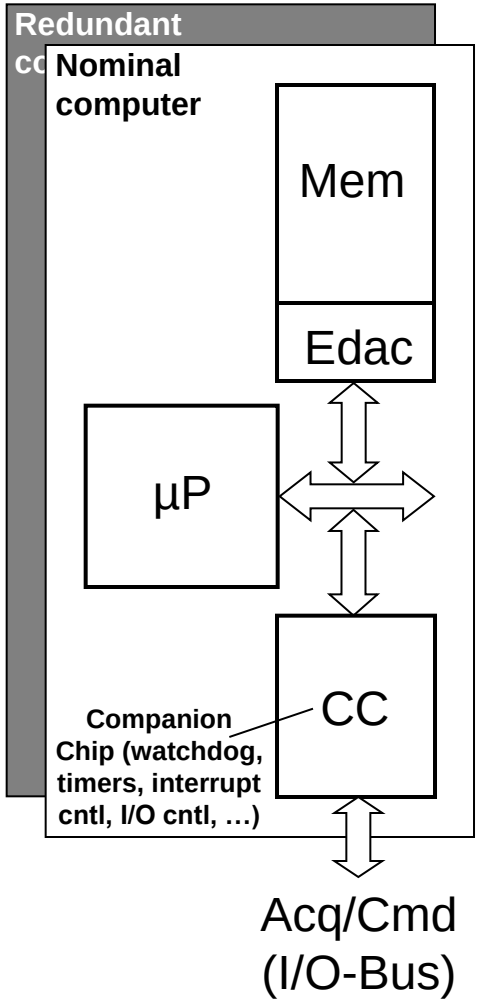
2-B – Processing units / Fault-tolerant (FT) architectures

- **Time replication**
  - **Time replication at instruction level**
    - Example of Time-TMR from SPACE MICRO Inc.
  - **Granularity for CNES FT architectures**
  - **Time replication at task level**
    - Example of DMT from CNES
- **Structural duplex**
  - Example of DT2 from CNES
- **TMR-Triplex & QMR-Quadruplex**
  - Examples issued from the SHUTTLE, GUARDS and ATV
- **Micro-synchronized triplex**
  - Example of SCS750 from MAXWELL Tech.
- **FT architectures trade-off**
- **Other methods and elementary protection mechanisms**

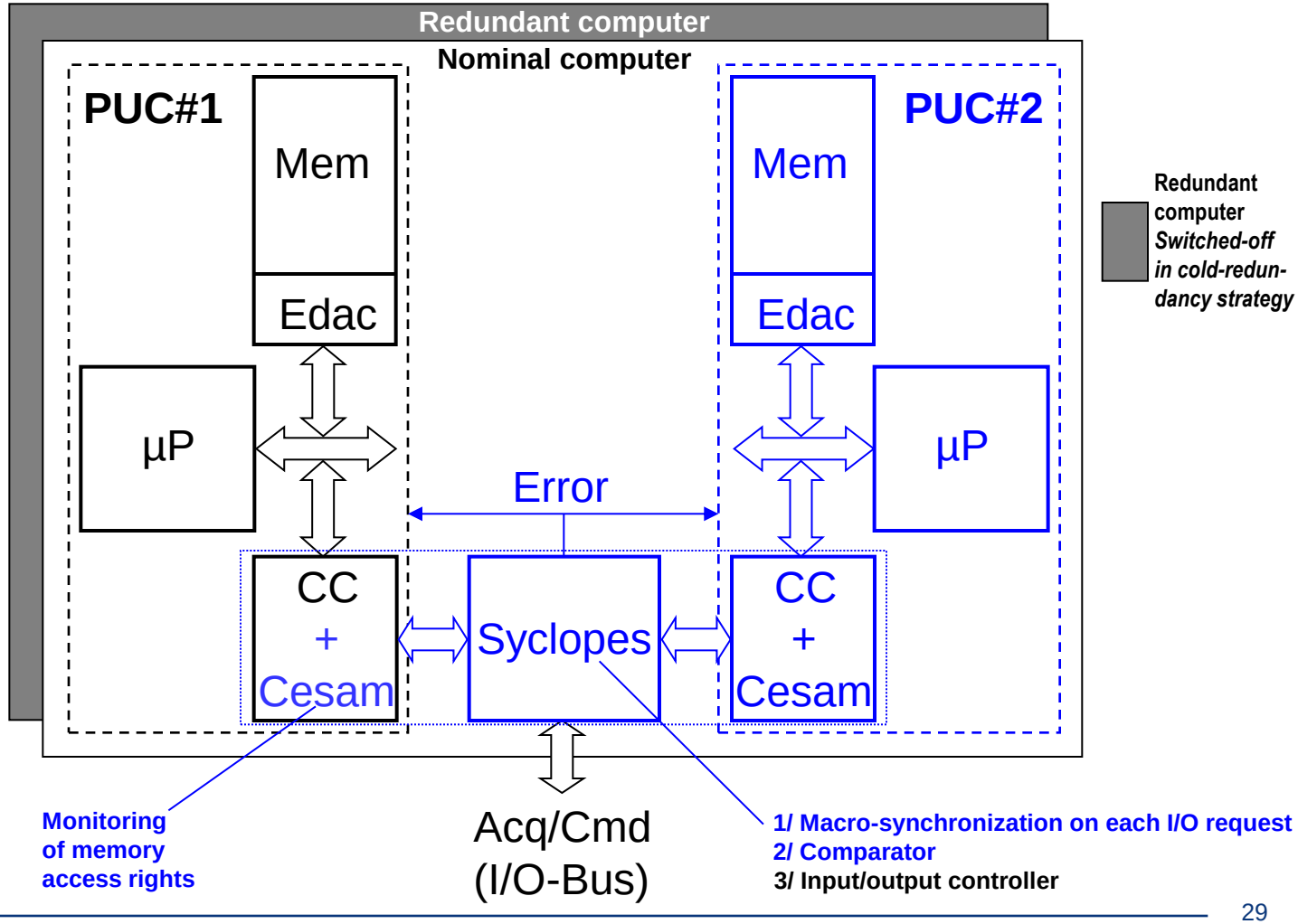
Mini-structural duplex with recovery capability

DT2 - Double Duplex Tolerant to Transients (CNES – Fr)

PUC without DT2



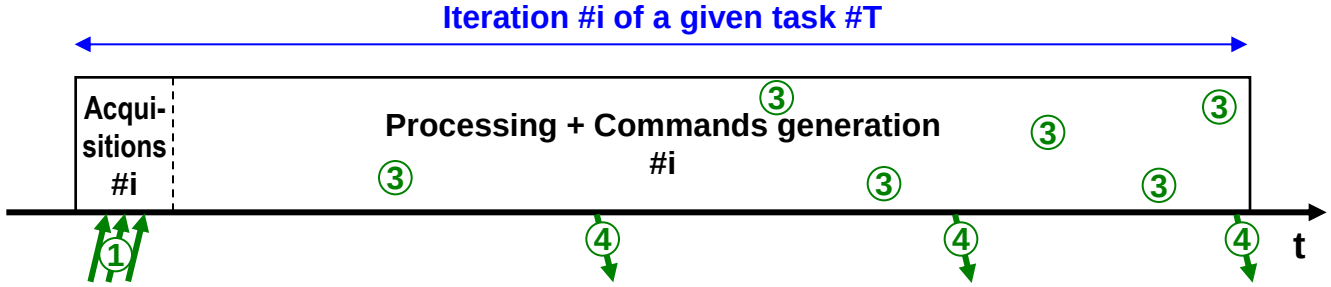
Processing Unit Core with DT2



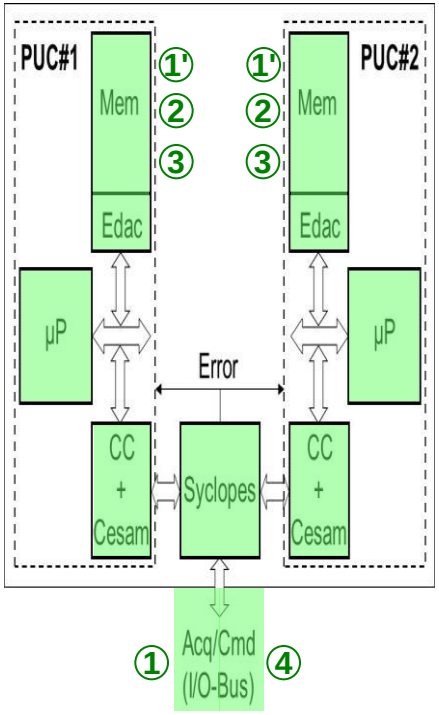
DT2 – Scheduling and fault detection principle

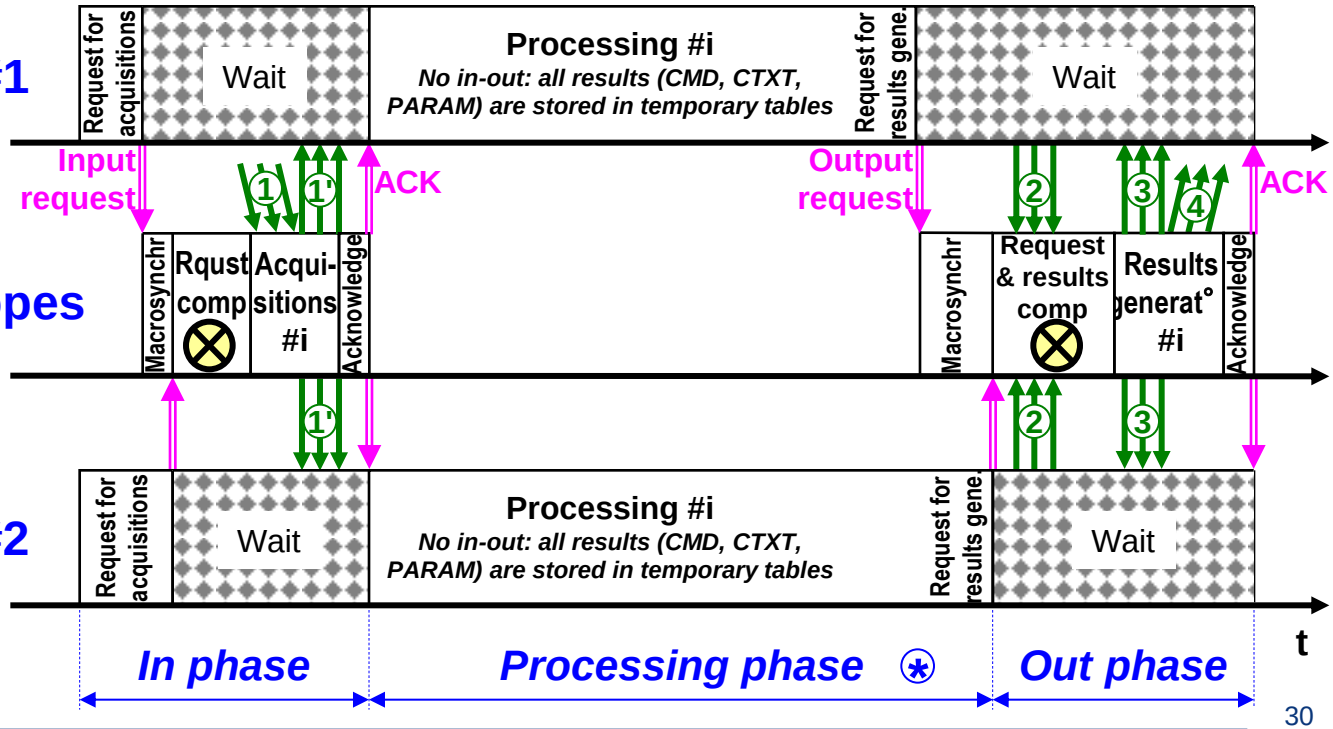
Without DT2

PUC



With DT2





Problem of recovery with a duplex

- A duplex is able to <u>detect</u>
  - comparison

=> *A duplex is intrinsically a "fail-stop" architecture*

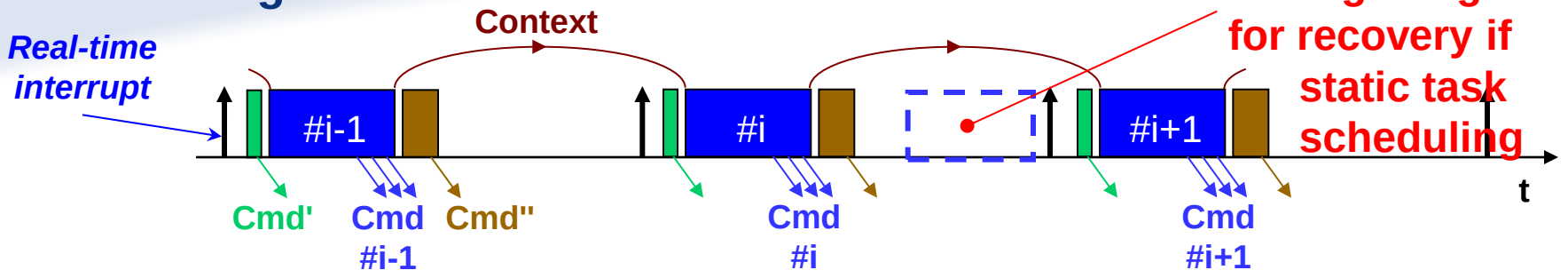
- A duplex is not able to <u>recover</u>
  - no information is available for determining which is the healthy/faulty channel (unlike a triplex architecture)

=> Specific mechanisms are required for implementing a <u>recovery</u> with a <u>duplex</u> architecture

Example of a DT2 backward recovery

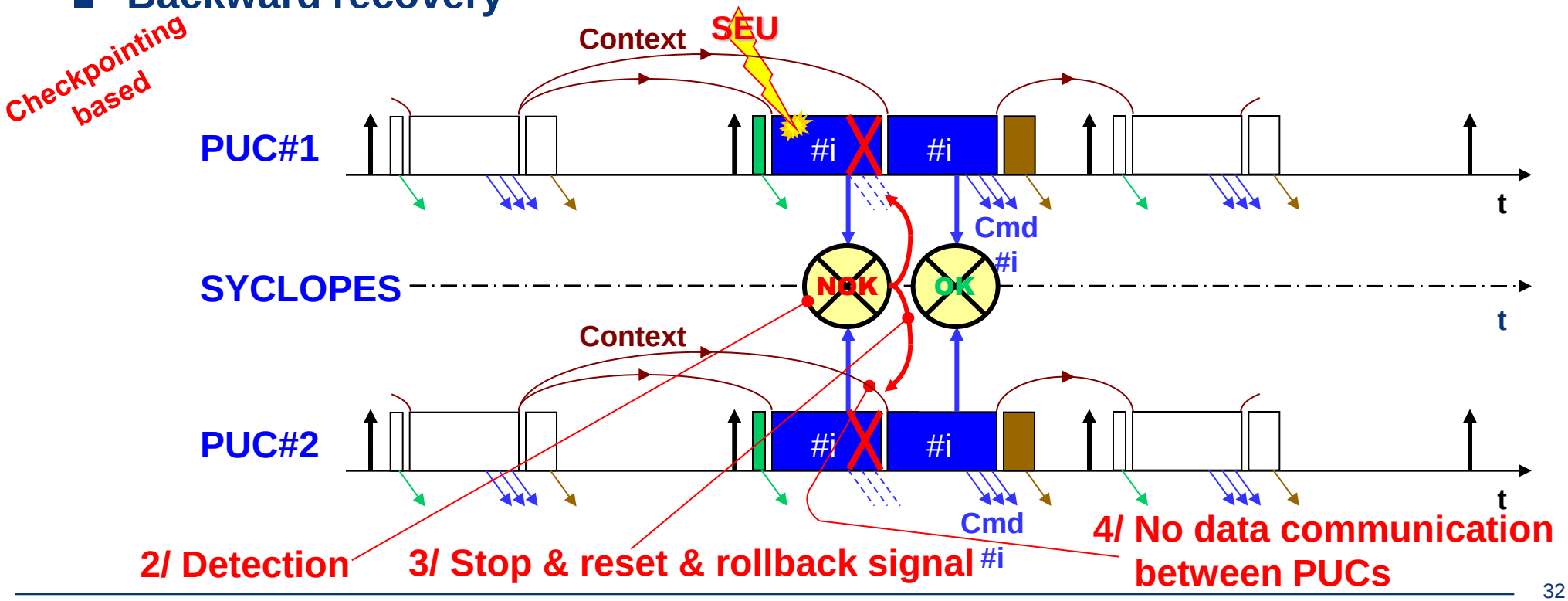
Recovery in the DMT is based on the same principle

- Nominal timing for PUC#1 and PUC#2



- Backward recovery

Checkpointing based



Two main conditions to recover successfully

- **The context data – basis of the recovery – must be healthy**
  - **The memory is considered SEE-free, thanks to an EDAC**
  - **A completely crashed µP must not be able to errouneously write in the memory zone where is stored all the context data**
    - Thanks to CESAM which checks the memory access rights
    - The final location of context data is updated only after the comparison of all results, and only if 100 % of results (CMD + CTXT + PARAM) are healthy
- **A completely crashed / hanged µP must be detected, and a warm-restart must be done on the software**
  - **A µP crash or hang will be detected**
    - By several mechanisms, e.g. memory access right monitoring
    - In the DT2: by the very short timeout monitoring each macro-synchro request
    - In the DMT: at least by the usual watchdog-timer
  - **A µP reset allows to passivate SEFI**
  - **The software warm-restart is possible thanks to the healthy context**

1/ CTXT data in memory must not be crashed

2/ A crashed µP must be detected and warm-restarted

DMT / DT2 pros/cons

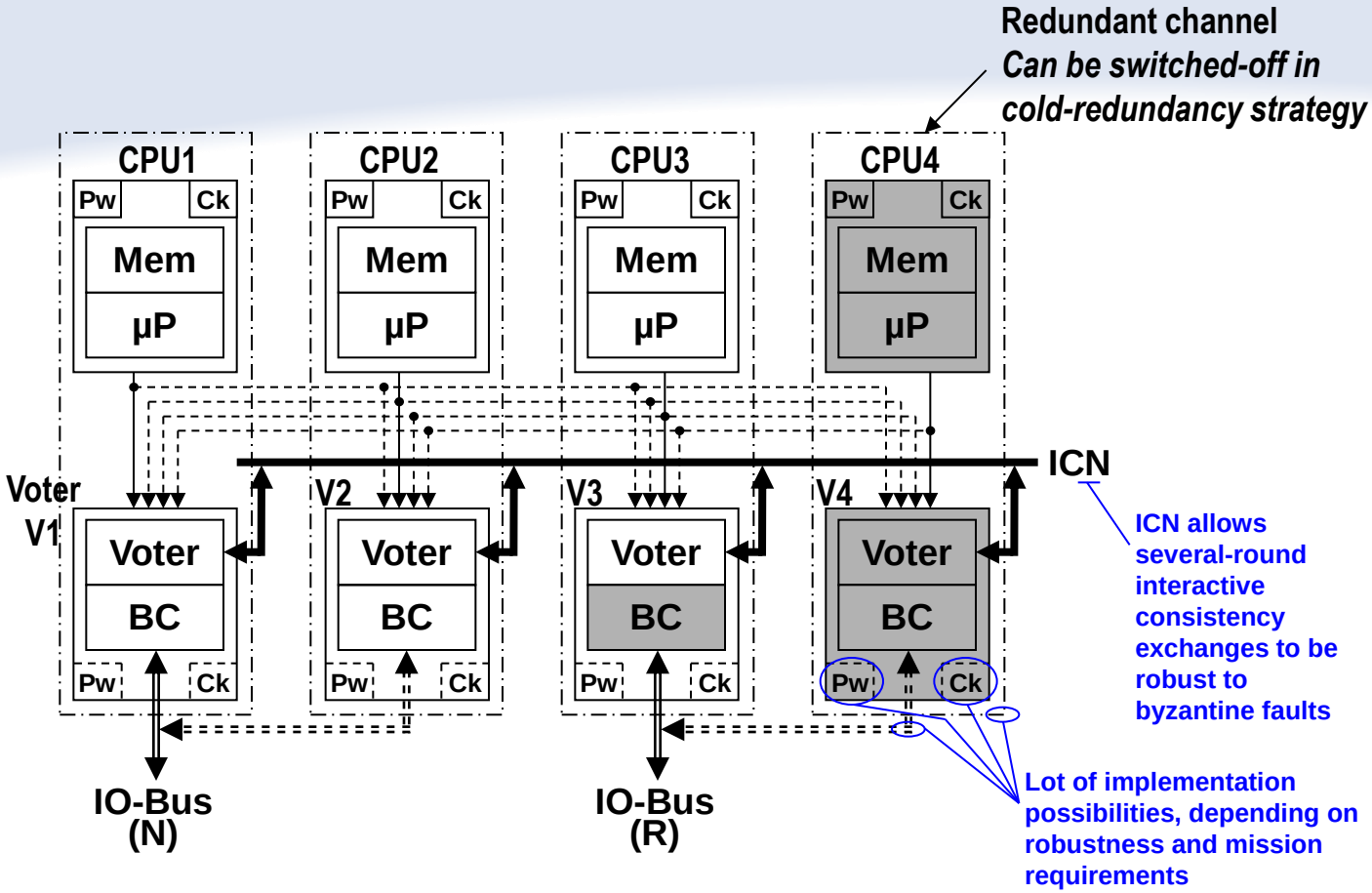
- ☺ **CNES proprietary architectures**
  - Available for every company
  - Open and scalable architectures
    - Possibility to implement evolutions
    - Possibility to select a subset of the validated mechanisms
- ☺ **Generic architectures independent from the microprocessor choice**
  - DSP or general purpose µP, single or multi-cores, superscalar or not, VLIW or not
  - No new development required when used on a new microprocessor
- ☺ **A single know-how for a two-fold architecture**
  - Same general principles for DMT and DT2 => one development for two different implementations, compatible with a larger part of potential applications
- ☺ **Low cost architectures**
- ☹ **Error coverage rate less than the one of a triplex architecture …**
- ☺ **… nevertheless suffisant for payloads**

2-B – Processing units / Fault-tolerant (FT) architectures

- **Time replication**
  - **Time replication at instruction level**
    - Example of Time-TMR from SPACE MICRO Inc.
  - **Granularity for CNES FT architectures**
  - **Time replication at task level**
    - Example of DMT from CNES
- **Structural duplex**
  - Example of DT2 from CNES
- **TMR-Triplex & QMR-Quadruplex**
  - Examples issued from the SHUTTLE, GUARDS and ATV
- **Micro-synchronized triplex**
  - Example of SCS750 from MAXWELL Tech.
- **FT architectures trade-off**
- **Other methods and elementary protection mechanisms**

TMR-Triplex & QMR-Quadruplex architecture

ICN = Inter-Channel Network
BC = I/O Bus Controller
IO-Bus = e.g. MIL-STD-1553
(N) = Nominal
(R) = Redundant
Pw = Power supply
Ck = Clock generator
Can be switched-off
Options



- Detection done by the majority vote
- Recovery in two steps
  - Fault-masking: The channels continue the processing for a short period of time; results of the faulty channel are continuously masked thanks to the healthy data issued by healthy channels => all commands and actuations will be correct
  - Channel alignment: The faulty channel is reinserted later because it takes a long time

TMR & QMR pros/cons (cont.)

- **Specificities / constraints**
  - Architecture pertaining to the distributed computing domain
  - Architecture requiring the highest level of theoretical analysis
  - Architecture generating an incredible number of theoretical studies (PhD, R&D, …), and a lot of different implementations depending on the user needs and system requirements

- **Pros/cons**
  - ☺ The best level of error coverage + masking capability (delayed recovery) well suited to some kind of applications
  - ☹ Overheads:
    - Mass
    - Recurring cost (extra ICs)
    - Power consumption
    - Complexity

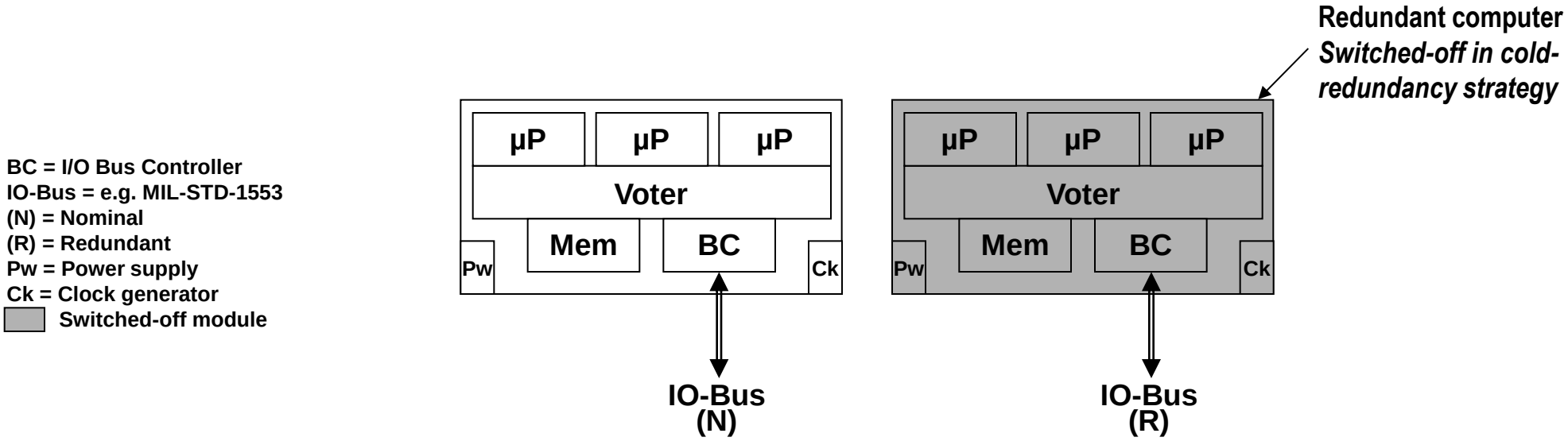
Selected on Shuttle & ATV

2-B – Processing units / Fault-tolerant (FT) architectures

- **Time replication**
  - **Time replication at instruction level**
    - Example of Time-TMR from SPACE MICRO Inc.
  - **Granularity for CNES FT architectures**
  - **Time replication at task level**
    - Example of DMT from CNES
- **Structural duplex**
  - Example of DT2 from CNES
- **TMR-Triplex & QMR-Quadruplex**
  - Examples issued from the SHUTTLE, GUARDS and ATV
- **Micro-synchronized triplex**
  - Example of SCS750 from MAXWELL Tech.
- **FT architectures trade-off**
- **Other methods and elementary protection mechanisms**

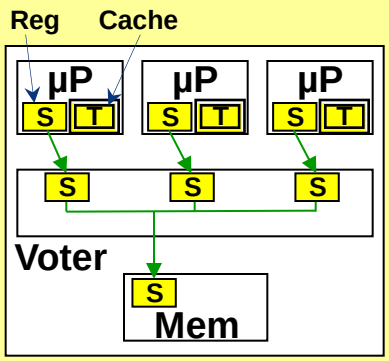
All previously presented architectures were MACRO-SYNCHRONIZED

Micro-synchronized triplex architecture ("lock-stepping")

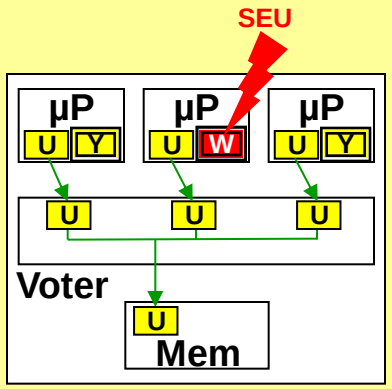


- All the µPs execute the same instruction at exactly the same clock cycle
- It requires to have a µP having a lock-stepping capability (e.g. synchro-nization of internal clock generators, bus comparators, …)
  - Very old µP: Intel Pentium & i960, IBM RH6000, Atmel three-chip ERC-32
  - Old µP: IBM PowerPC740/750
  - Recent µP: ARM Cortex-R family (dual-core)

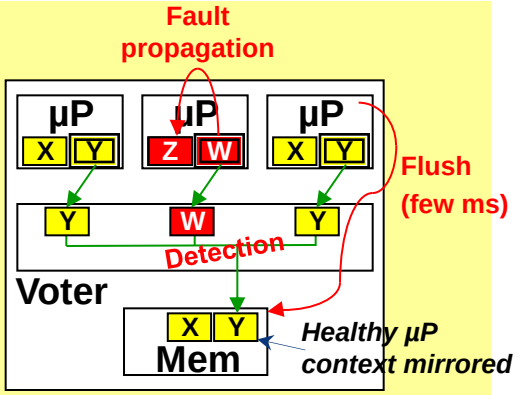
Micro-synchronized triplex (cont.)



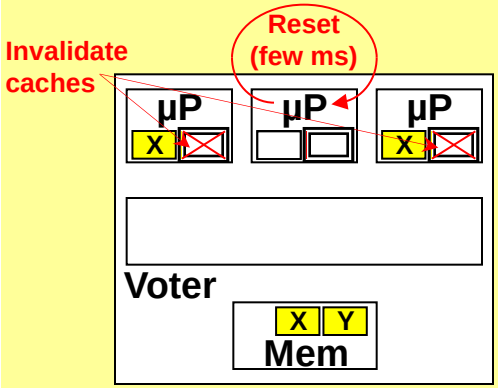
Nominal processing phase



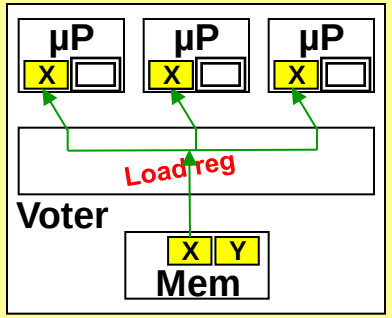
SEU, then processing before detection



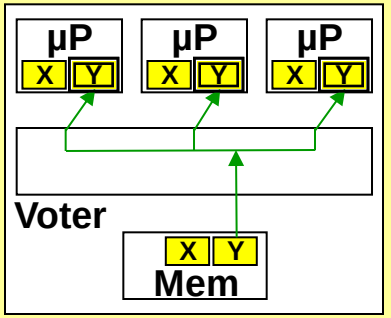
Error detect° => Start recov: Flush reg/caches…



…+ invalidate caches + reset faulty µP…



…+ load reg/conf reg…



…+ resume

Micro-synchronized triplex (cont.)

- When an error is detected on 1 of the 3 µPs, a recovery phase is started
  - Flush all the registers / caches of the µPs to the single main memory
    - Thanks to the masking capability of the voter, the data set written back in main memory is 100 % healthy => a full and healthy µP context is saved into memory
  - Then invalidate the caches (i.e. reset the caches)
    - To force the µP to read back the main memory for all data without exception
  - Then reset the faulty µP
  - Then load the faulty µP registers (including configuration registers)
  - Then start again the processing phase
    - The three µPs must read all their data in the main mem. (due to cache invalidat°)
    - So the faulty µP will be "aligned" on the two healthy ones thanks to the healthy context mirrored into the external memory

=> µP alignment in 3 steps:

- Flush = Ctxt mirrored
- Cache invalidation
- Resume: alignment is inherent

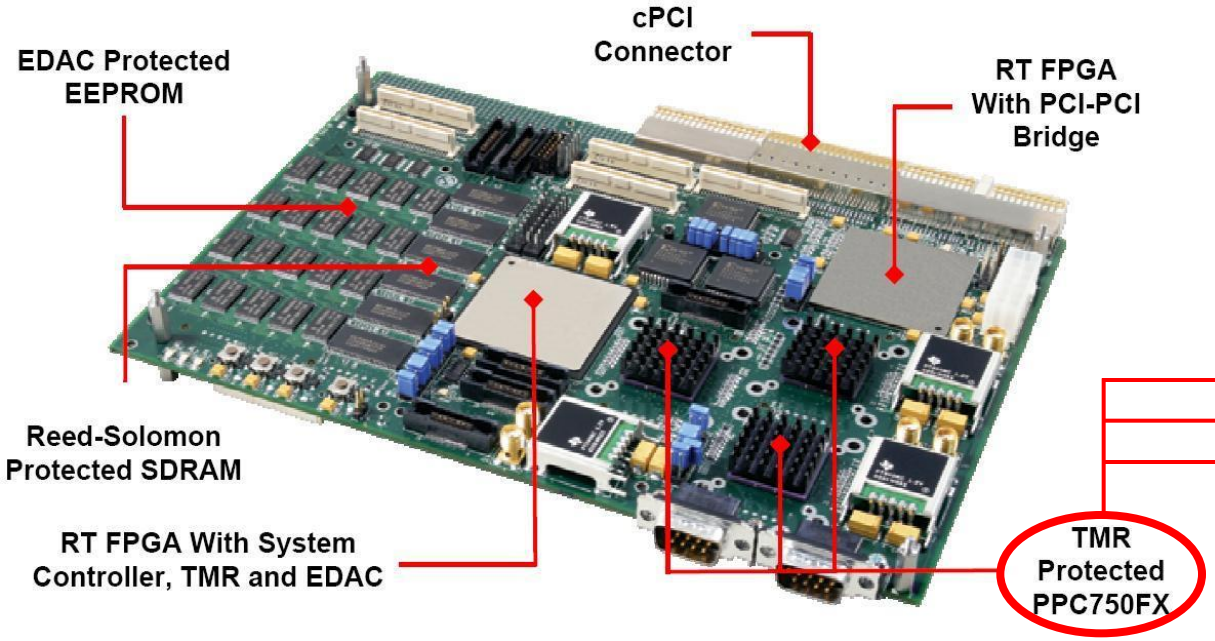
Alignment performance:

- Flush = few ms
- µP reset = few ms
- Resume: processing is slowing down (cache empty)

Micro-synchronized triplex (cont.)

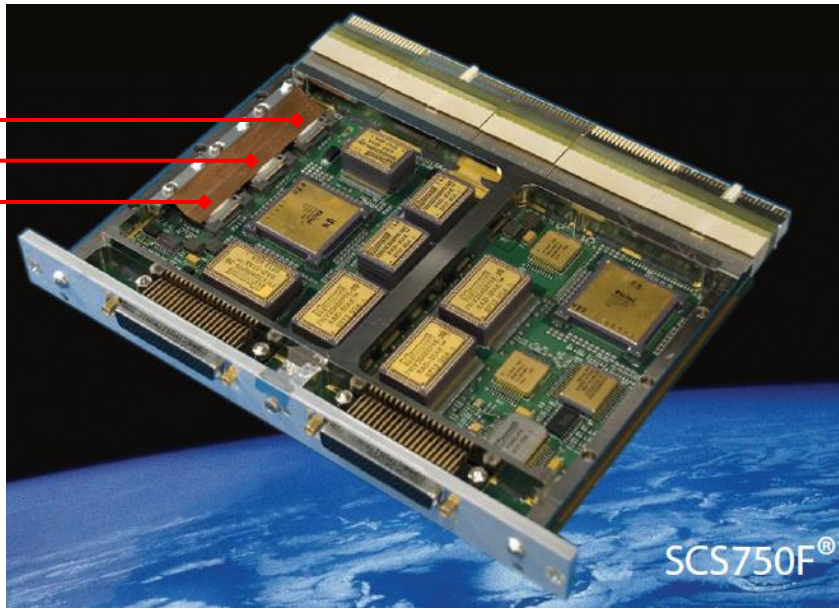
- Real case example of an industrial development

SCS750 - Super Computer for Space (Maxwell Tech. – USA)



SCS750P Prototype Model

Flying on GAIA (GLORY launch failed)



SCS750F Flight Model

7 – 25 W (typ) depending on clock rate

Micro-synchro. triplex pros/cons

- µ-synchro. archi. is dedicated to µP having a lock-stepping capability
- ☹ This capability is becoming obsolescent due to deep submicron techno
  - TID effects => asymmetric modif. of internal propagation delays between µPs
  - Fully deterministic timing is less and less feasible
  - Low-level fix-up routines to tolerate timing violations and soft-errors
  - Multiple and complex clock trees
  - etc.

Nevertheless …

Micro-synchro. duplex

- Recent automotive safety norm: ISO26262 (adaptation of IEC61508)
  - "Functional safety standard" which stipulates regulations for HW and SW in electronics control systems to manage the risk of hazardous events
  - ARM Cortex-R is oriented "Real-Time" for deeply embedded systems
    - with a focus on fast/deterministic response to interrupts, determinism (tightly-coupled memories) and safety/dependability (memory protection unit, ECC/parity, **lock-step**)
    - dual-core µP allowing implementation of a lock-step configuration to ease the compliance with ISO26262

2-B – Processing units / Fault-tolerant (FT) architectures

- **Time replication**
  - **Time replication at instruction level**
    - Example of Time-TMR from SPACE MICRO Inc.
  - **Granularity for CNES FT architectures**
  - **Time replication at task level**
    - Example of DMT from CNES
- **Structural duplex**
  - Example of DT2 from CNES
- **TMR-Triplex & QMR-Quadruplex**
  - Examples issued from the SHUTTLE, GUARDS and ATV
- **Micro-synchronized triplex**
  - Example of SCS750 from MAXWELL Tech.
- **FT architectures trade-off**
- **Other methods and elementary protection mechanisms**

Fault-tolerant architectures trade-off

- **There is not an universal solution …**
  - **Optimization is predominant over standardization**

Real cases of COTS-based computers:

- UCTM-C/D (ARIANE 5, first launch 1996) = Double **structural duplex**, recovery without context
- ARGOS (launched in 1999) = EDDI / **Time replication** at instruction level
- BIRD (2001) = Double **structural duplex**, specific recovery mechanisms
- MYRIADE (2004) = Mix of **elementary protection mechanisms**
- REIMEI (2005) = **Macro-synchronized triplex** with a single voter
- ROADRUNNER (2006) = TTMR / **Time-TMR** at instruction level
- CALIPSO (2006) = **Lock-stepping quadruplex** with a redundant voter
- GLORY (2011) & GAIA (2013) = **Lock-stepping triplex** with a single voter

HiRel-based:

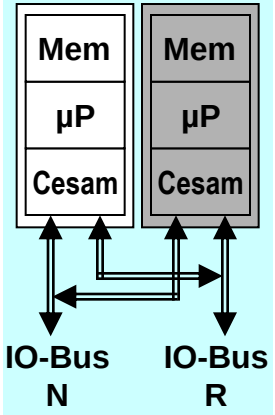
- Shuttle (first launch 1981) = **"4+1"-MR** (QMR + 1 backup)
- ATV (2009) = **Triplex** + **Duplex**
- DMS-R (on ISS) = **Triplex**

- **… the final choice of the best suited architecture for a given project is application dependent**
  - **Only 'detection', or 'detection and recovery'**
  - **Hardware and software cost overhead**
  - **Development and recurring cost overhead**
  - **Power consumption overhead**
  - **The time required for the recovery process**

Redundant computer
Switched-off in cold-redundancy strategy

Fault-tolerant architectures trade-off (cont.)

DMT

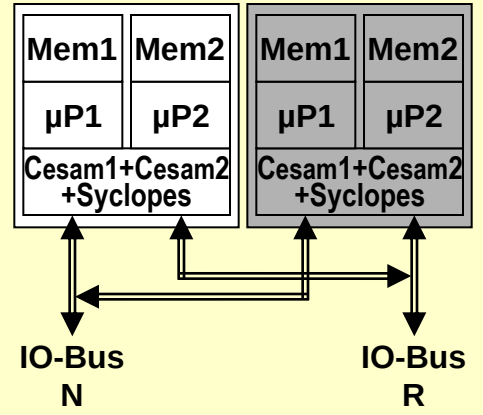


a = Nb of main items µP + Mem + Asic => Mass
b = Nb of main items ON => Power consumption
c1 = Computing pwr available (detection only)
c2 = Computing pwr available (detect° +recov.)
d1 = Availability
d2 = Correct actuations

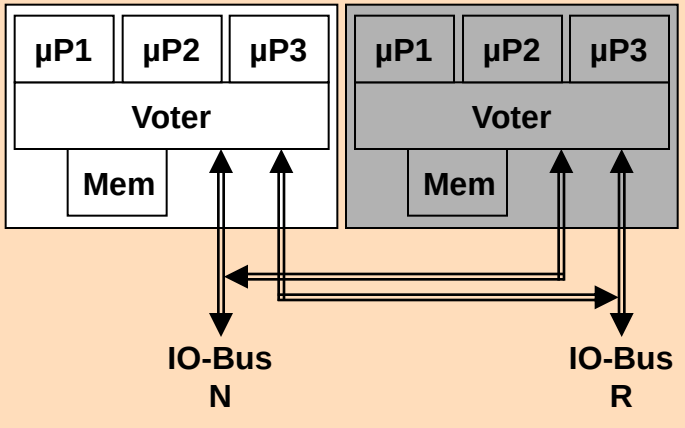
a =	6	10
b =	3	5
c1 / c2 =	0.5 / 0.3	1 / 0.7
d1 / d2 =	0.95 / 0.99	0.99
a =	10	12
b =	5	9
c1 / c2 =	1 *	1 *
d1 / d2 =	0.995	0.999

* = Requires time for computers alignment

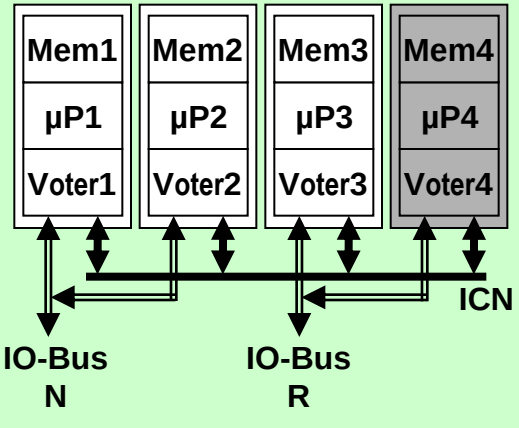
DT2



Micro-synchronized triplex



Triplex/quadruplex



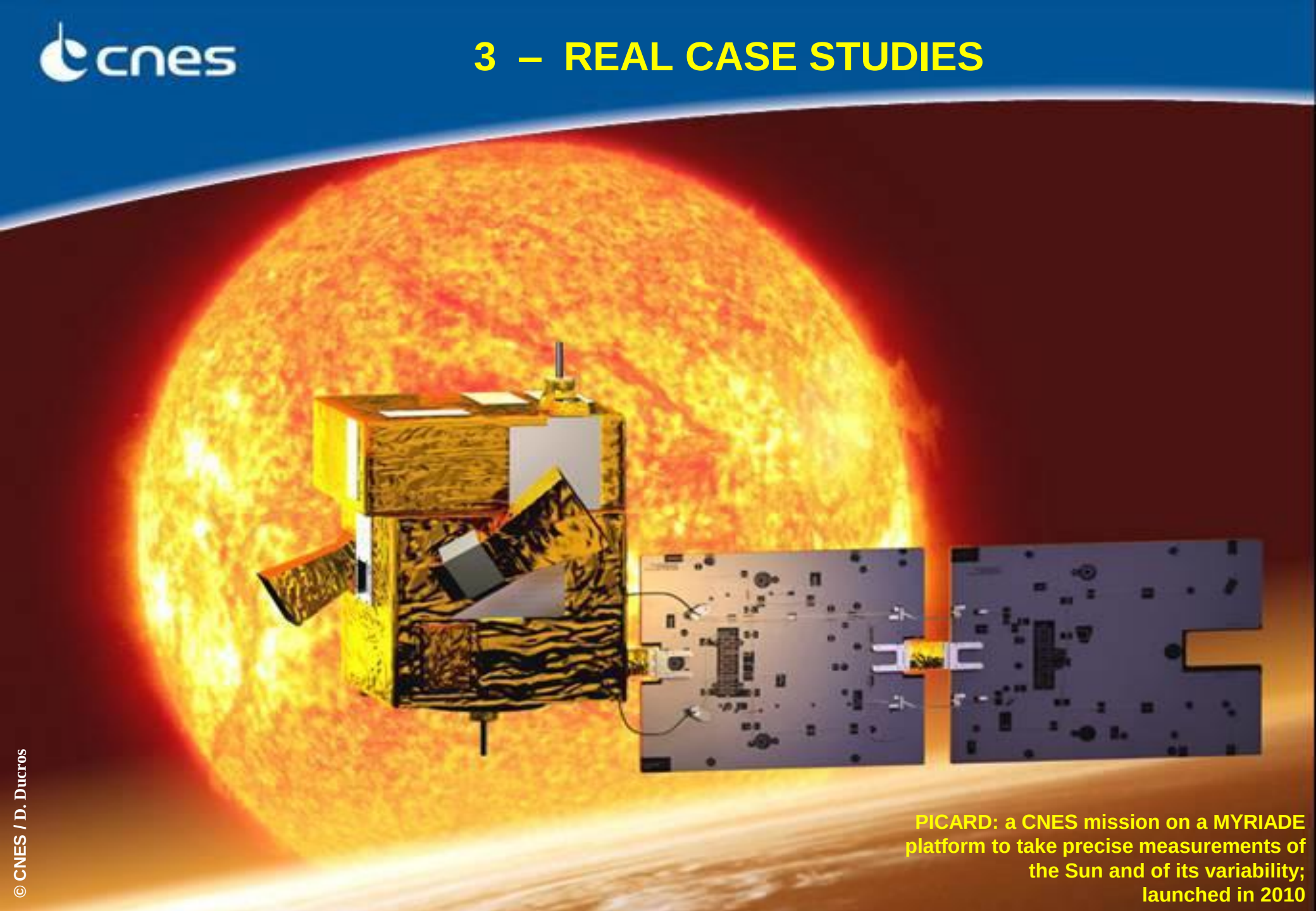
2-B – Processing units / Fault-tolerant (FT) architectures

- **Time replication**
  - **Time replication at instruction level**
    - Example of Time-TMR from SPACE MICRO Inc.
  - **Granularity for CNES FT architectures**
  - **Time replication at task level**
    - Example of DMT from CNES
- **Structural duplex**
  - Example of DT2 from CNES
- **TMR-Triplex & QMR-Quadruplex**
  - Examples issued from the SHUTTLE, GUARDS and ATV
- **Micro-synchronized triplex**
  - Example of SCS750 from MAXWELL Tech.
- **FT architectures trade-off**
- **Other methods and elementary protection mechanisms**

Other methods and elementary protection mechanisms

- ABFT – Algorithm-Based Fault Tolerance
- BIST – Built-In Selft Test
- WDP – WatchDog Processor (signature analysis)
- Wrappers
- etc.
- Mix of different elementary protection mechanisms
  - For protection at component level: ASIC
    - e.g. ERC32 and LEON European space microprocessors
  - For protection at the system level
    - e.g. The CNES MYRIADE micro-satellite => See Part III

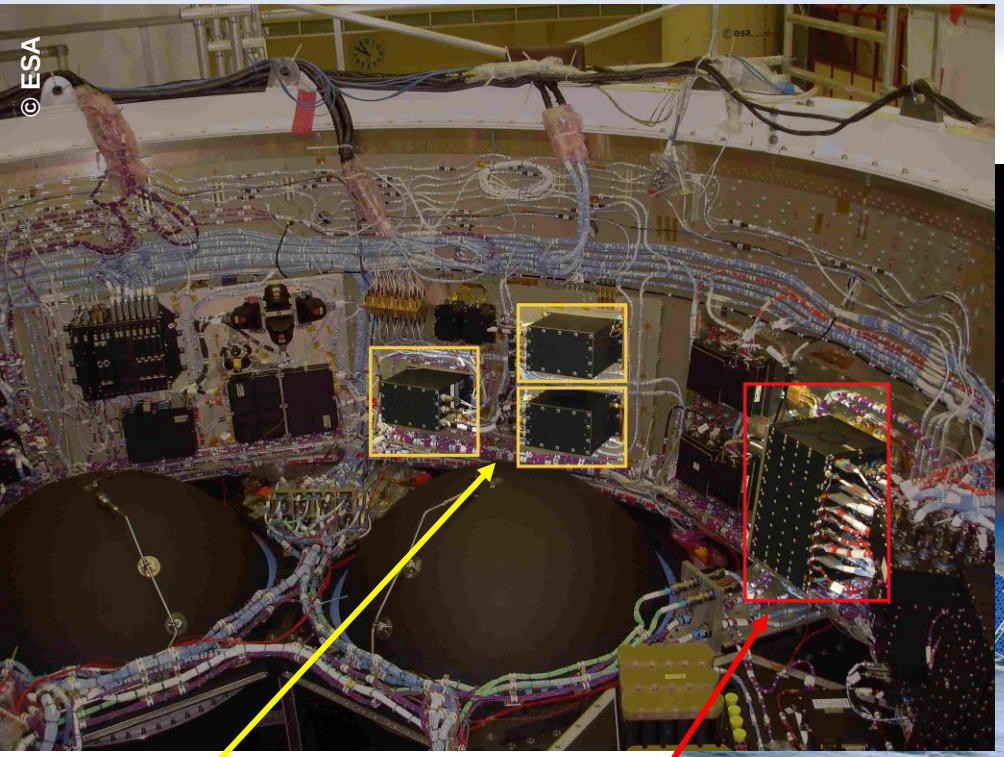
3 – REAL CASE STUDIES



PICARD: a CNES mission on a MYRIADE platform to take precise measurements of the Sun and of its variability; launched in 2010

The ATV example (with rad-hard ICs)

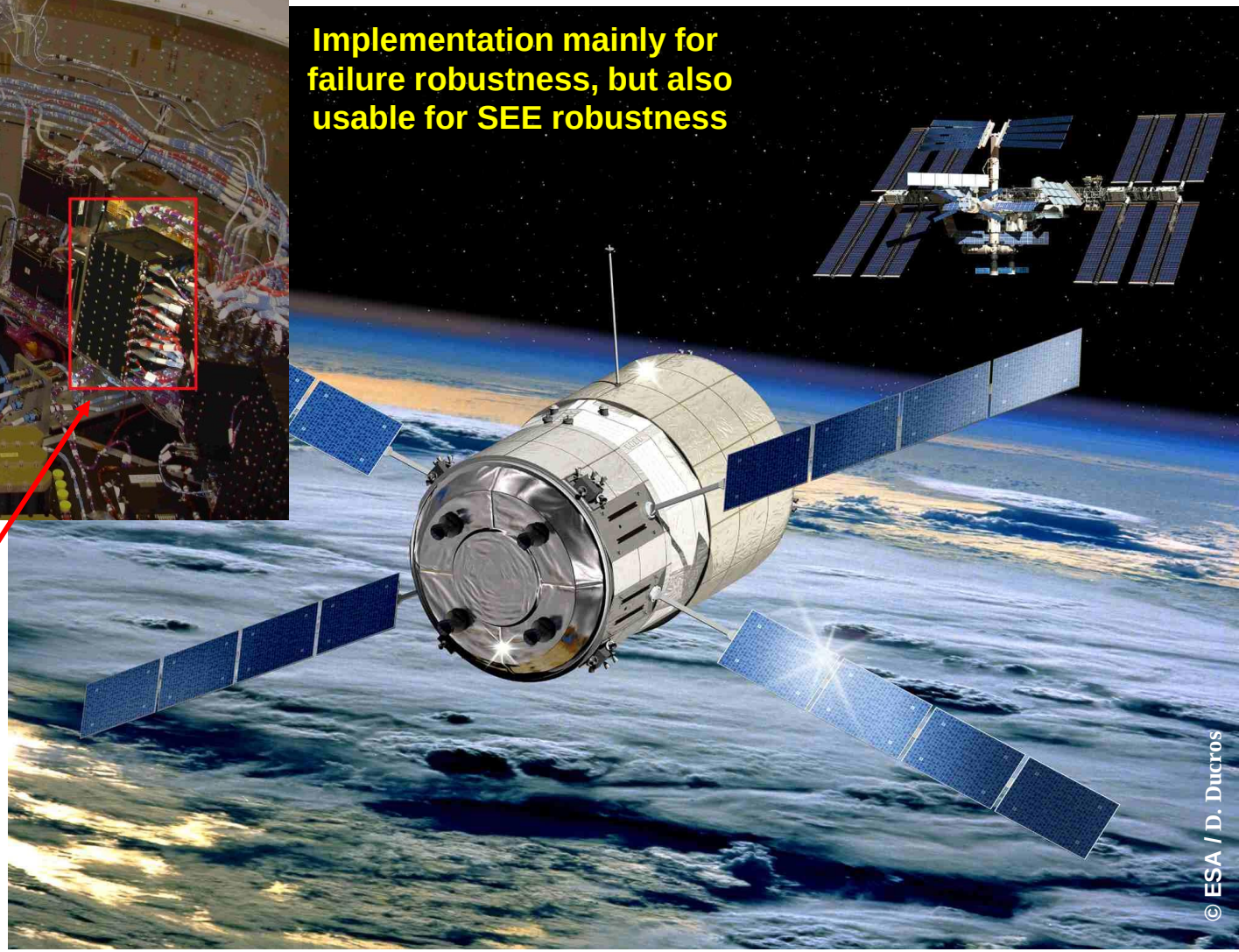
ESA Automated Transfer Vehicle servicing the ISS (1st launch in 2008)



Triplex + Duplex

Duplex goal: tolerance to software bugs

Implementation mainly for failure robustness, but also usable for SEE robustness



The main monitoring and control computer FTC (triplex)

The checker computer MSU (duplex) monitoring the critical docking phase (collision avoidance)

MYRIADE: a CNES µSL family developed mainly with COTS

Contribution from: J-L. Carayon (CNES - DCT/TV/AV)

- TID: Switch-off sensitive ICs when not used
- Protons: The Transputer µP is protected with a 2 mm tungsten shield
- SEL: Serial resistors on power supply tracks or current limiter
- SET: Filtering of analog acquisitions
  - Time redundancy + average value computation
- SEU: Protection of link/bus data exchanges
  - Checksum/CRC and recovery protocols
- SEU-SET: Flash and FRAM are protected
  - Redunded data, checksum or CRC
  - Flash and FRAMS are switched-off after the boot of the flight software
- SEU: FPGA with critical registers implemented in with a TMR structure
- SEU-MBU: TMR for critical data stored in the Transputer memory
  - For flight software memory (4 Mbytes), not for TM memory (120 Mbytes)
- SEU: Monitoring of some µP internal critical registers (timers, …)

MYRIADE family:
44 years of total cumulated in-orbit life, with any failure and only two system reboots!; all the other events were hidden to the system, and filtered at a low level

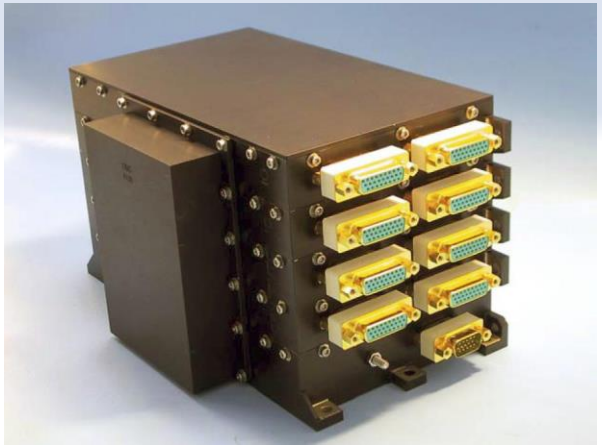
MYRIADE (cont.)

- **SEU-SEL: Watchdog (WD) implemented with several levels**
  - *Note: Each I/O block* **is** *constituted by a PIC nanocontroller and i/f ICs*
  - **Internal PIC WD set to 100 ms: protection of PIC itself against SEL/µSEL or software hang due to a SEE (SEFI)**
  - **Global WD for each I/O block set to 250 ms**
  - **Local WD for Transputer CPU set to 500 ms**
  - **Global WD for computer set to 1 sec with four levels of actions having deeper and deeper effect on the computer**
    - **Transputer reset**
    - **Transputer Off/On (in case of SEL)**
    - **CPU board Off/On (at this level, the Transputer memory content is lost)**
    - **Computer Off/On (in order to passivate any residual SEL)**

=> MYRIADE is a typical example of a computer developed with commercial components and protected by a mix of elementary mechanisms for a mission without high availability requirements

MYRIADE (cont.)

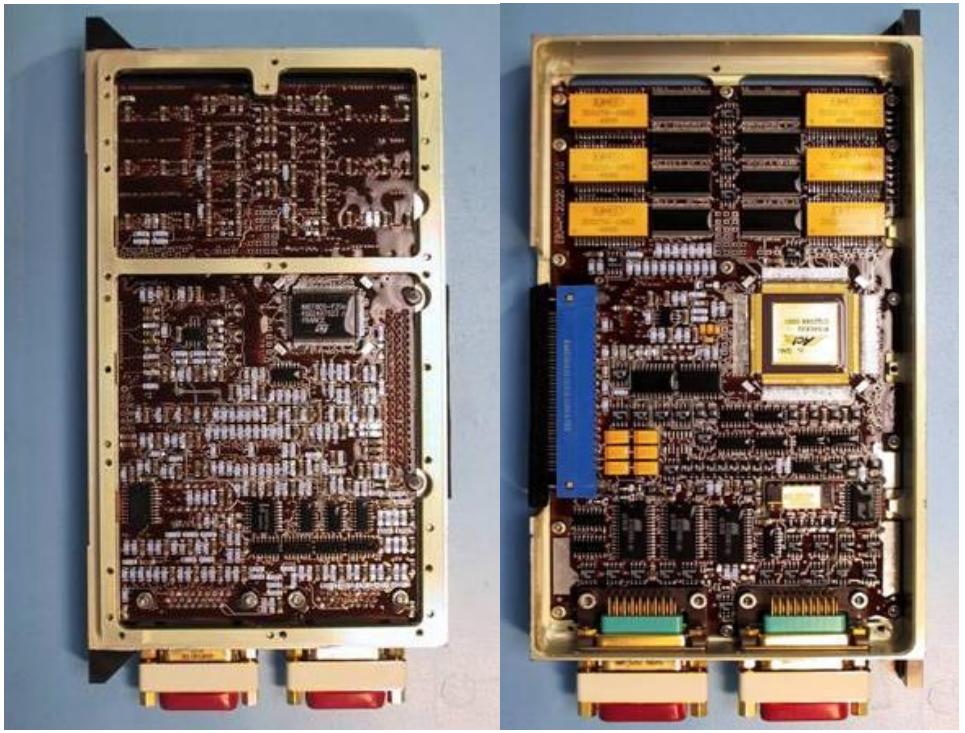
MYRIADE platform during integration



MYRIADE computer (CNES and Steel Electronique development)

DEMETER: 1st mission based on a MYRIADE platform (launched in 2004)

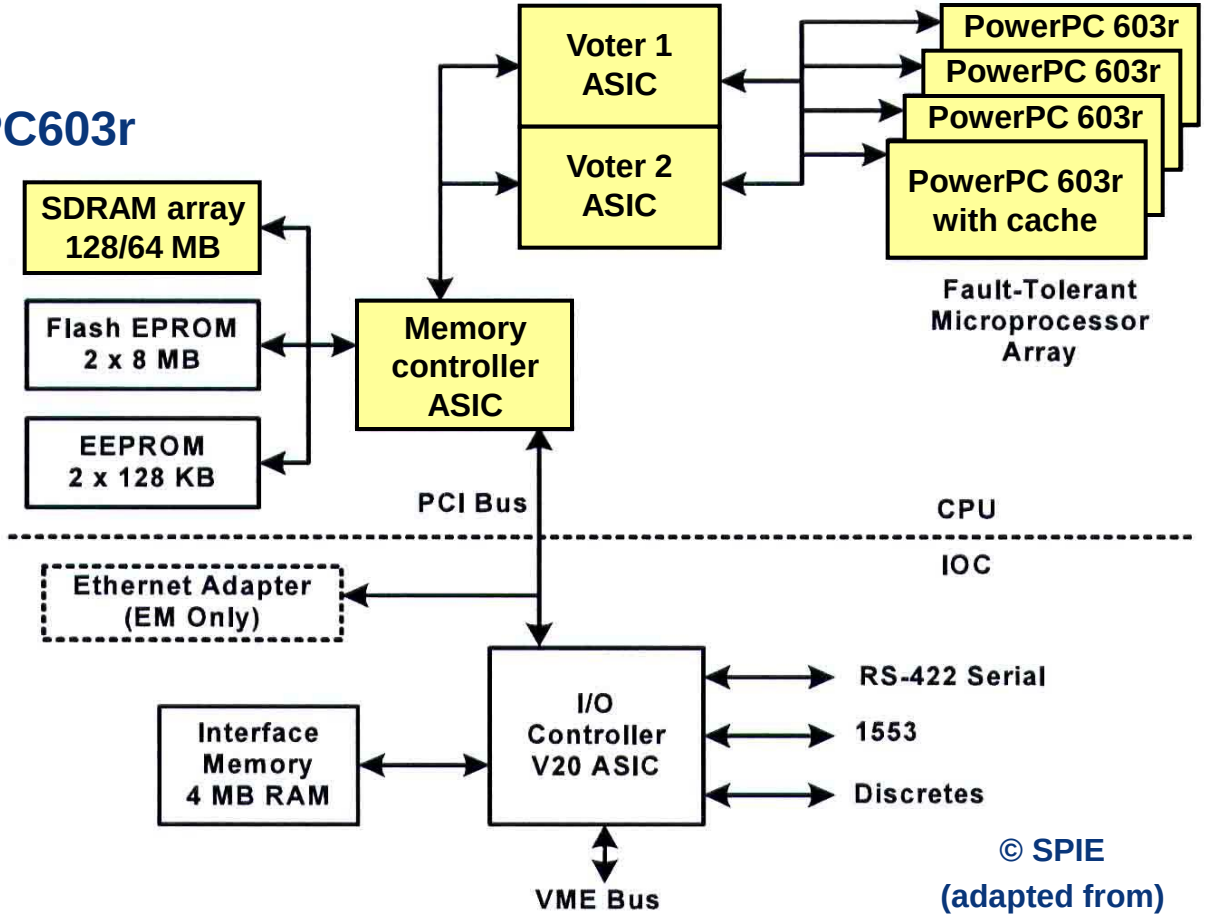
MYRIADE CPU board



CALIPSO: a US fault-tolerant COTS-based space computer developed by GDAIS (General Dynamics Advanced Information Systems)

- CALIPSO is a Franco-American payload on a CNES PROTEUS mini-satellite platform for cloud-aerosol and infrared observations, launched in 2006

- COTS µP = Freescale PowerPC603r
- 4-MR architecture
- Voter is not a SPF
- Micro-synchro. / lock-stepping

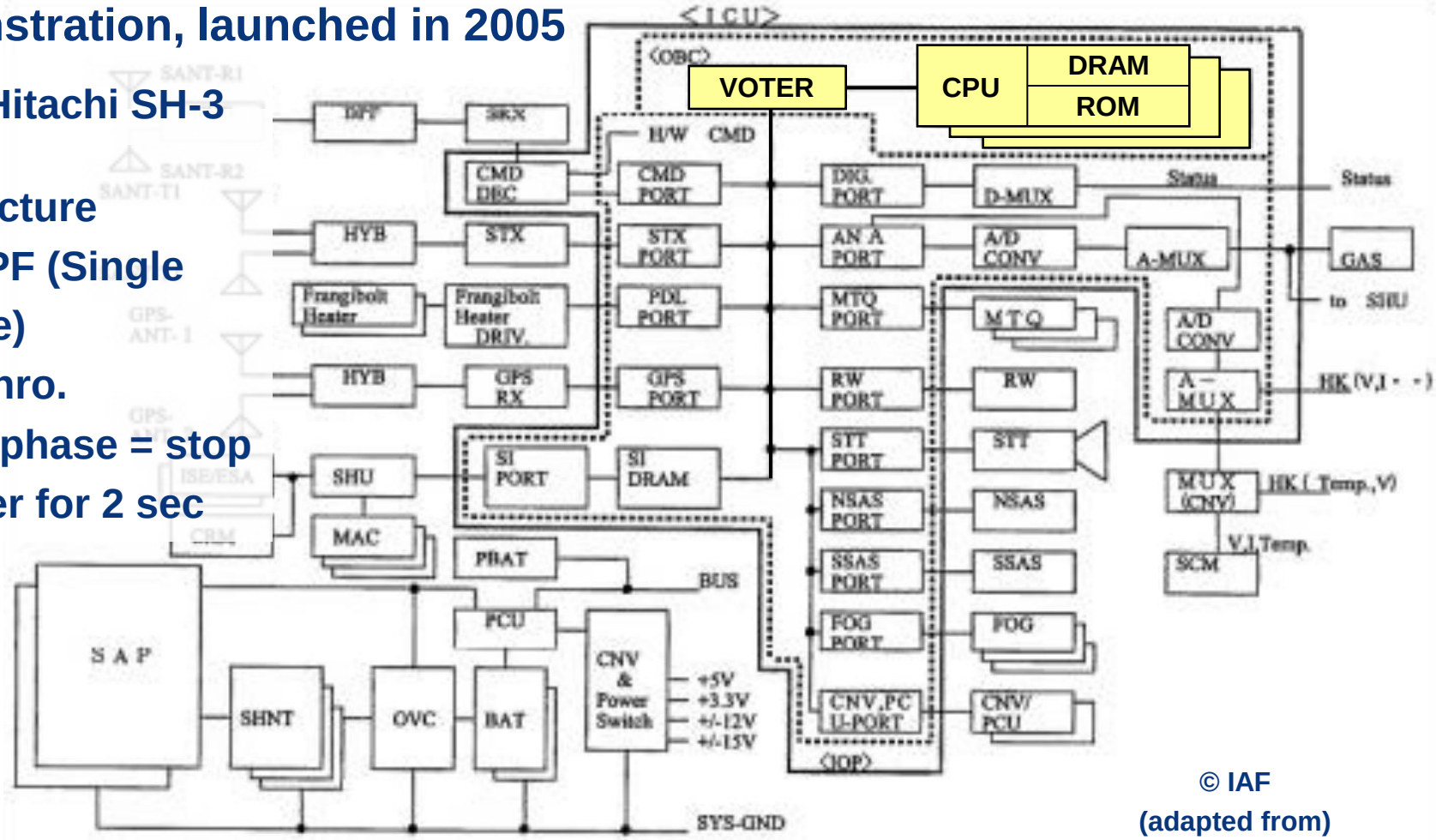


© SPIE (adapted from)

REIMEI (INDEX): a Japanese fault-tolerant COTS-based space computer developed by ISAS/JAXA + University of Tokyo

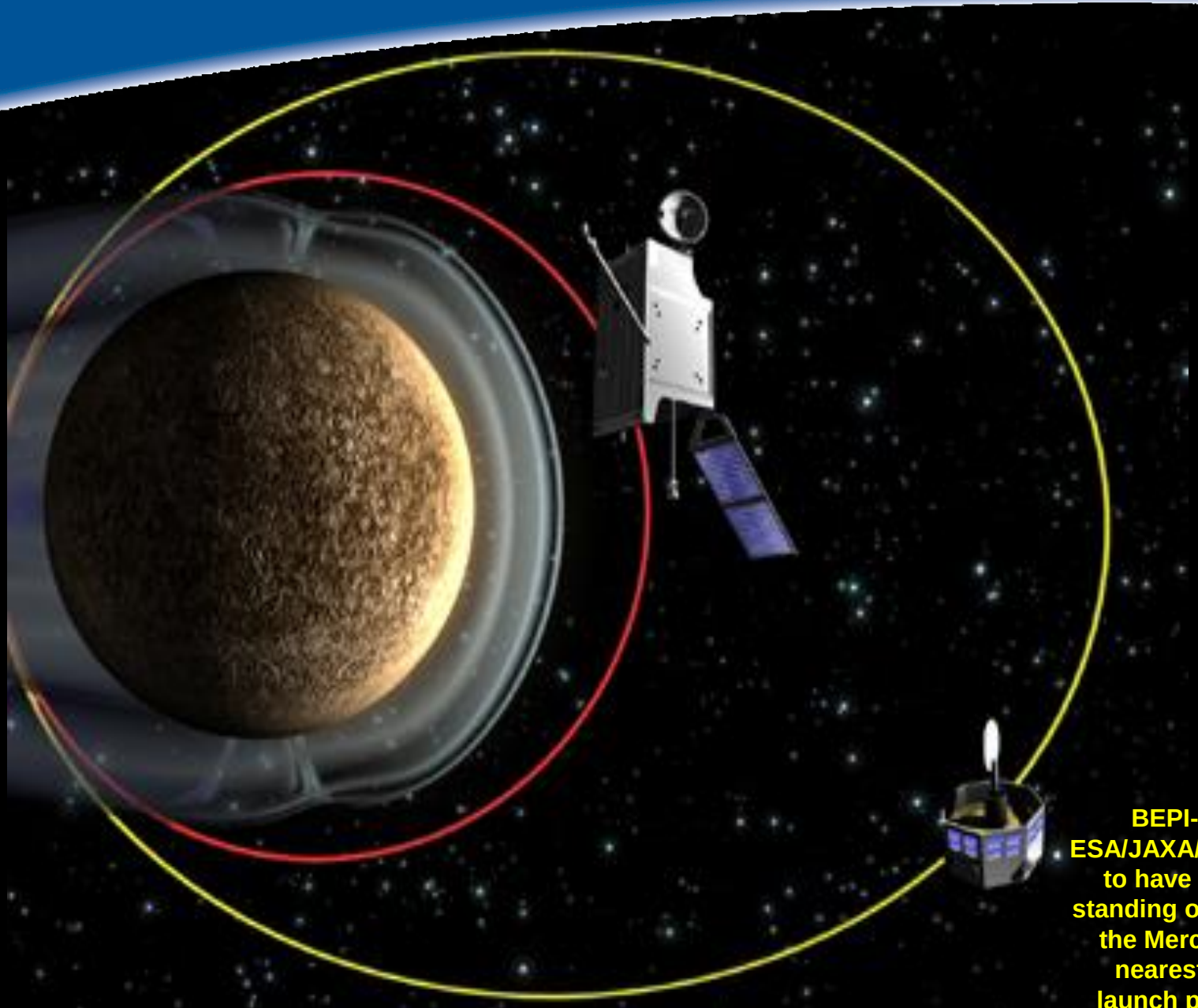
- REIMEI is a small satellite for aurora observation and technology demonstration, launched in 2005

- COTS µP = Hitachi SH-3
- TMR architecture
- Voter is a SPF (Single Point Failure)
- Macro-synchro.
- Reinsertion phase = stop the computer for 2 sec



© IAF
(adapted from)

4 – CONCLUSION



BEPI-COLOMBO: an ESA/JAXA/CNES mission to have a better understanding of the history of the Mercure planet, the nearest from the Sun; launch planned in 2015

COTS-based supercomputers have been selected on two major space programs

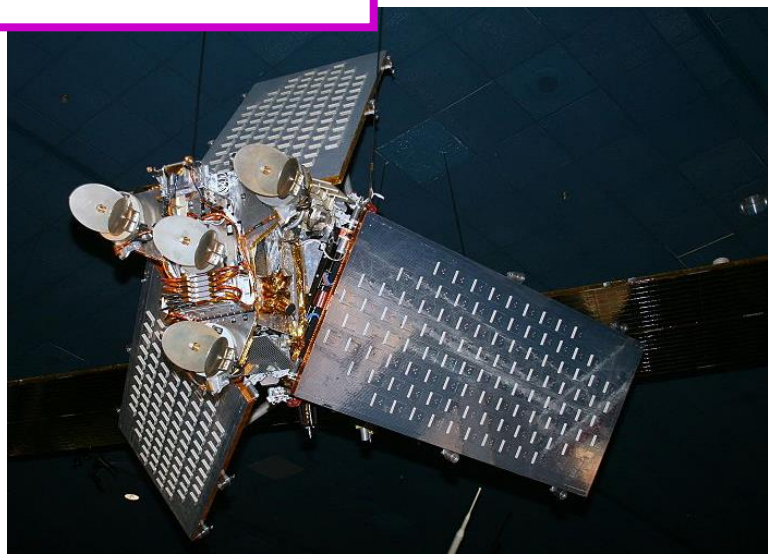
The ESA GAIA mission (Hipparcos NG)

7 COTS-based computers

=> # 12 GIPS peak

Generation of the largest and most precise three-dimensional map of our Galaxy

(2030 kg, launched in 2013)



1st gene. : 7 COTS-based computers per SL

=> # 1.5 GIPS peak per SL in 1998

2nd gene.: k x COTS-based computers

=> k x 2.4 GIPS peak

Satellite phone constellation

(2nd generation: 81 satellites including spares, 800 kg each, 1st launch planned in 2015)

... and today on other space programs

REFERENCES

References

Processing unit – Fault tolerant architectures

General references

- Siewiorek D. P. and Swarz R. S., "Reliable Computer Systems - Design and Evaluation", 908 p., Digital Press, Bedford, MA, USA, 1992.
- Geffroy J.-C. and Motet G., "Design of Dependable Computing Systems", 700 p., Kluwer Academic Publishers, 2002.
- Laprie J.-C., Arlat J., Blanquart J.-P., Costes A., Crouzet Y., Deswarte Y., Fabre J.-C., Guillermain H., Kaâniche M., Kanoun K., Mazet C., Powell D., Rabéjac C. and Thévenod-Fosse P., "Guide de la sûreté de fonctionnement", 369 p., Cépaduès-Éditions, Toulouse, 1995-96.
- Pignol M., "How to Cope with SEU/SET at System Level?", Proc. 11th IEEE Int. On-Line Testing Symp. (IOLTS), pp. 315-318, 2005.
- Pignol M., Carayon J.-L., Chapuis T., Peus A. and Saba B., "Radiation Effects on Digital Systems", in Chapter III-03 of *Space Radiation Environment and its Effects on Spacecraft Components and Systems (SREC'04)*, Space Technology Course of CNES/ONERA/RADECS Association, Cépaduès Editions, Toulouse (France), ISBN 2-85428-654-5, pp. 411-459, 2004.
- Mitra S., Seifert N., Zhang M., Shi Q. and Kim K.S., with participation of Nicolaidis M. and Chardonnereau D., "Robust System Design with Built-In Soft-Error Resilience", IEEE Computer, Vol. 38, Iss. 2, pp. 43-52, Feb. 2005.
- Nicolaidis M., "A Low-Cost Single-Event Latchup Mitigation Scheme", Proc. 12th IEEE Int. On-Line Testing Symp. (IOLTS), pp. 111-115, 2006.

- ***Presentation of some architectures***

– **BIRD computer architecture:** Behr P., Bärwald W., Briess K. and Montenegro S., "Fault Tolerance and COTS: Next Generation of High Performance Satellite Computers", Eurospace/ESA/CNES DAta Systems In Aerospace Conf. (DASIA), June 2003.

– **CNES DMT:** Pignol M., "Processing Procedure for an Electronic System Subject to Transient Error Constraints and a Memory Access Monitoring Device", patent US 6 839 868 B1.

– **CNES DT2:** Pignol M., "Software System Tolerating Transient Errors and Control Process in such a System", patent US 7 024 594 B2.

– **CNES DMT & DT2 architectures:** Pignol M., "DMT and DT2: Two CNES Fault-Tolerant Architectures Developed by CNES for COTS-based Spacecraft Supercomputers", Proc. 12th IEEE Int. On-Line Testing Symp. (IOLTS), 2006, pp. 203-212.

– **CNES DMT & DT2 architectures:** Pignol M., Parrain T., Claverie V., Boléat C., Estaves G., "Development of a Testbench for Validation of DMT and DT2 Fault-Tolerant Architectures on SOI PowerPC7448", Proc. 14th IEEE Int. On-Line Testing Symp. (IOLTS), 2008, pp. 182-184.

– **CRAFT architecture:** Hihara H., Yamada K., Adachi M., Mitani K., Akiyama M. and Hama K., "CRAFT: An Experimental Fault Tolerant Computer System for SERVIS-2 Satellite", Proc. 21th AIAA Int. Communications Satellite Systems Conf. (ICSSC), Paper n° 2003-2291, 2003.

– **EDDI:** Lovelette M.N., Wood K.S., Wood D.L., Beall J.H., Shirvani P.P., Oh N. and McCluskey E.J., "Strategies for Fault-Tolerant, Space-Based Computing: Lessons Learned from the ARGOS Testbed", IEEE Proc. of Aerospace Conf., vol. 5, pp. 2109-2119, 2002.

– **GUARDS architecture:** Bondavalli A., Di Giandomenico F., Grandoni F., Powell D. and Rabejac C, "State Restoration in a COTS-based N-Modular Architecture", 1st Int. Symp. on Object-Oriented Real-Time Distributed Computing (ISORC), pp. 174-183, 1998.

– **GUARDS architecture:** Powell D., Arlat J., Beus-Dukic L., Bondavalli A., Coppola P., Fantechi A., Jenn E., Rabéjac C. and Welling A., "GUARDS: A Generic Upgradable Architecture for Real-Time Dependable Systems", IEEE Trans. on Parallel and Distributed Systems, Vol. 10, N° 6, pp. 580-599, June 1999.

– **GUARDS architecture:** Powell D. (Ed.), "A Generic Fault-Tolerant Architecture for Real-Time Dependable Systems", Edited by David Powell, Kluwer Academic Publishers, Boston, 2001.

– **HERMES computer architecture:** Guidal C. and David P., "Development of a Fault Tolerant Computer System for the Hermes Space Shuttle", Proc. IEEE Fault-Tolerant Computing Symp. (FTCS-23), 1993.

– **REE computer architecture:** Whisnant K., Iyer R.K., Jones P., Some R. and Rennels D.A., "An Experimental Evaluation of the REE SIFT Environment for Spaceborne Applications", Proc. IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN), pp. 585-594, 2002.

– **SPACE SHUTTLE computer architecture:** Sklaroff J. R., "Redundancy Management Technique for Space Shuttle Computers", IBM Journal of Research and Developments, Vol. 20, pp. 20-28, Jan. 1976.

– **TMR – Byzantine faults:** Lamport L., Shostak R. and Pease M. (SRI International), "The Byzantine Generals Problem", ACM Transactions on Programming Languages and Systems, vol. 4, n° 3, July 1982, pp. 382-401.

- ***Other kind of elementary protections***

– **ABFT:** Turmon M. and Granat R., "Algorithm-Based Fault Tolerance for Spaceborne Computing: Basis and Implementations", Proc. IEEE Aerospace Conf., March 2000.

– **BIST:** Nicolaidis M., "Efficient UBIST Implementation for Microprocessor Sequencing Parts", Int. Test Conference (ITC), pp. 316-326, 1990.

– **BIST:** Nicolaidis M. and Boudjit M., "New Implementations, Tools, and Experiments for Decreasing Self-Checking PLAs Area Overhead", IEEE Int. Conf. on Computer Design (ICCD), pp. 275-281, Oct. 1991.

- **WDP:** Leveugle R., "Analyse de signature et test en ligne intégré sur silicium", Thèse de Doctorat en microélectronique (PhD), INPG Grenoble, France, Jan. 1990.
- **WDP:** Mahmood A. and McCluskey E. J., "Concurrent Error Detection Using Watchdog – A Survey", IEEE Trans. on Computers, Vol. 37, N° 2, Feb. 1998.
- **WDP:** Valentin T., "Méthode de conception d'architectures tolérantes aux fautes transitoires à base de composants commerciaux : application aux communications en milieu spatial", chapitre II.3, Thèse de Doctorat en électronique et informatique industrielle (PhD), Bretagne Sud University, France, May 2000.
- **Wrappers:** Arlat J., Fabre J-C., Rodriguez M. and Salles F., "Dependability of COTS Microkernel-Based Systems"; IEEE Trans. on Computers, Vol. 51, N° 2, pp. 138-163, Feb. 2002.
- **Wrappers:** Salles F., Rodriguez M, Fabre J.-C. and Arlat J., "MetaKernels and Faults Containment Wrappers", IEEE Int. Symp. on Fault-Tolerant Computing (FTCS-29), pp. 22-29, June 1999.
- **µP protections:** Gaisler J., "Concurrent Error-Detection and Modular Fault-Tolerance in an 32-bit Processing Core for Embedded Space Flight Applications", Proc. IEEE Fault-Tolerant Computing Symp. (FTCS-24), 1994.
- **µP protections:** Gaisler J., "A Portable and Fault-Tolerant Microprocessor Based on the SPARC V8 Architecture", Proc. IEEE Dependable System and Networks (DSN), 2002.

Real case studies

- **MYRIADE computer architecture:** Carayon J.-L., Dubourg V., Danto P., Galéa G., "An Innovative Onboard Computer for CNES Microsatellites", Proc. 21th IEEE Digital Avionics Systems Conf. (DASC), 2002.
- **CALIPSO computer architecture:** R. DeCoursey, R. Melton, and R. Estes, "Non Radiation Hardened Microprocessors in Space-Based Remote Sensing Systems", Proc. SPIE Sensors, Systems, and Next-Generation Satellites X, vol. 6361, 2006.
- **REIMEI/INDEX computer architecture:** Saito H., Masumoto Y., Mizuno T., Miura A., Hashimoto M., Ogawa H., Tachikawa S., Oshima T., Hirahara M., Okano S., *et al.*, "INDEX: Piggy-Back Satellite for Aurora Observation and Technology Demonstration", 51th IAF Int. Astronautical Congress, Oct. 2000.

Industrial products

Maxwell Technologies

- Hillman R., Swift G., Layton P., Conrad M., Thibodeau C. and Irom F., "Space Processor Radiation Mitigation and Validation Techniques for an 1,800 MIPS Processor Board", Proc. 12th RADECS Association/ESA/IEEE European Conf. on Radiations and its Effects on Components and Systems (RADECS), Sept. 2003.
- MAXWELL Technologies SCS750 web site: http://www.maxwell.com/go/scs750a.html

Space Micro Inc.

- Czajkowski D. and McCartha M., "Ultra Low-Power Space Computer Leveraging Embedded SEU Mitigation", Proc. IEEE Aerospace Conf., March 2003.
- Space Micro Inc. web site: http://www.spacemicro.com/

ARM Ltd

- Cortex-R family web site: http://www.arm.com/products/processors/cortex-r/index.php


cnes
Gracias!
Thank you!
Any questions?