De-RISC

Dependable Real-time Infrastructure for Safety-critical Computer

www.derisc-project.eu

De-RISC

Dependable Real-time Infrastructure for Safety-critical Computer

Official project presentation

fentISS, Barcelona Supercomputing Center, Thales Research and Technology, and Cobham Gaisler

September 2022

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement EIC-FTI 869945

TABLE OF CONTENTS

Table of contents


CONSORTIUM

CONSORTIUM PARTNERS

PARTNERS

- **Hypervisor provider (XtratuM)**
  - Space qualified hypervisor
  - Efficiently supporting multicores to deliver high performance and isolation features

- **Extended Statistics Unit**
  - Monitors and controls multicore interference
  - Eases multicore adoption in safety critical systems in general, and space in particular

PARTNERS

- **End-user experience**
  - Sets requirements and assesses technology through their use cases

- **MPSoC provider (based on NOEL-V cores)**
  - High performance fault tolerant multicore (includes general and space specific interfaces)

ABOUT De-RISC: GENERAL INFORMATION

ABOUT De-RISC

- The open-source Instruction Set Architecture (ISA) **RISC-V** has become extremely popular:
  - Supported by a plethora of companies and research institutions.
  - Unique opportunity to develop **EU-based products** with no dependence on external technology or licenses,
  - Tailored to fulfil the requirements of critical embedded systems by means of a hypervisor.

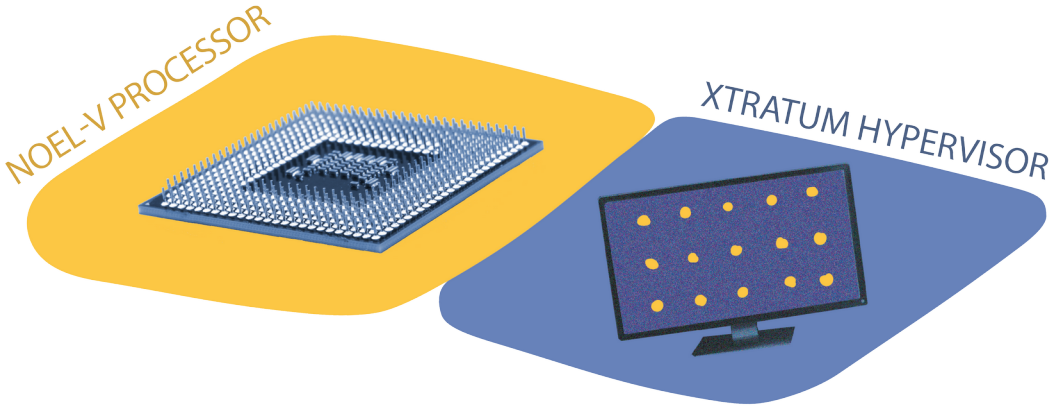
De-RISC

Dependable Real-time Infrastructure for Safety-critical Computer

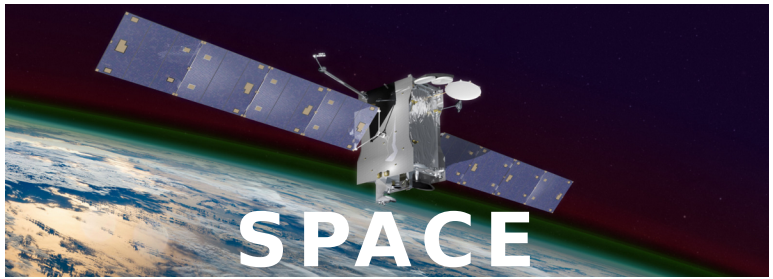
ABOUT De-RISC

Hardware and software platform based around the **RISC-V standard:**

Multi-core RISC-V system-on-chip + **XtratuM hypervisor** = full platform consisting of hardware and software



EUROPEAN DIMENSION AND MARKET UPTAKE



Considering both the new and traditional space systems, it is of outmost importance for Europe to be **nondependent** both within space technology and in access to space. De-RISC addresses these trends, and the shift towards **new computer architectures** with a European solution. Moreover, De-RISC leverages the momentum of **RISC-V** in the commercial domain and deeply embedded markets.

Having a unique **Integrated Modular Avionics (IMA) multicore platform** with hardware and software technology fully developed in Europe as is the case of the De-RISC platform, will be a **unique selling point** for its adoption in future aviation projects of European aircrafts manufacturers.

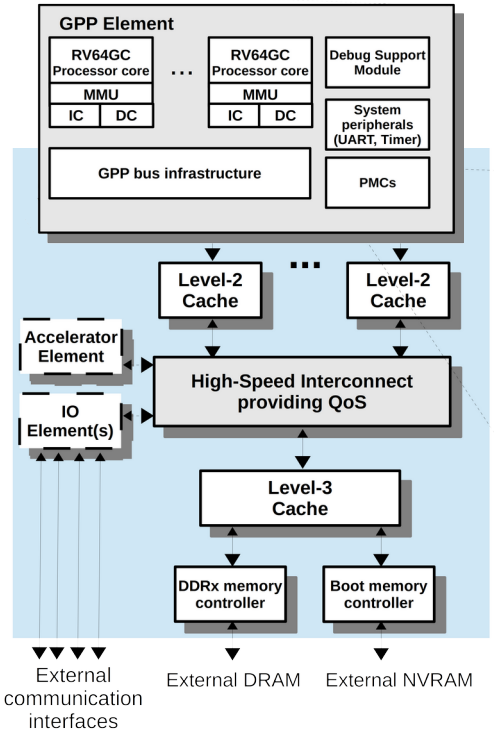
OBJECTIVES

- **No US export restrictions**: De-RISC's IP core platform and software will not be subject to any US regulatory influence.
- **Multi-core interference mitigation concepts integrated in the RISC-V SoC**: this becomes a unique feature and will provide a key advantage by limiting drastically interference while preserving high-performance operation.
- **Portability**: The proposed development can be implemented in FPGAs and application specific standard products (ASSPs). This provides an edge for integrators that can adapt their choice of implementation technology based on mission requirements.
- **Fault-tolerance concepts**: The platform will be provided by companies with experience in the space domain and with heritage in design of fault-tolerant systems.
- **Future-proof selection for new platforms**: With an established vendor providing a RISC-V platform there are guarantees of continued support for the hardware platform while developments from the commercial domain for the RISC-V architecture can be leveraged over time.


ABOUT De-RISC:
TECHNICAL INFORMATION

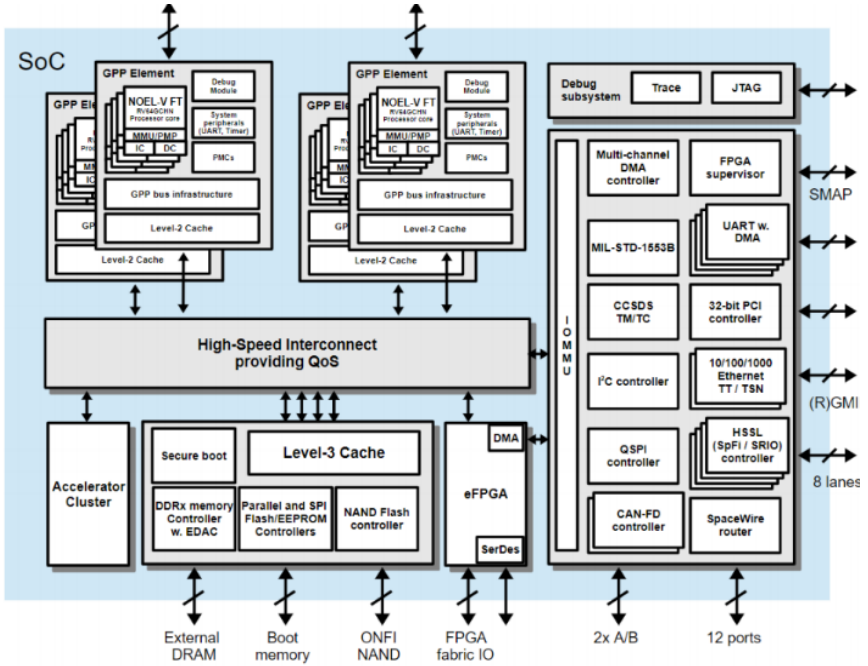
HARDWARE ARCHITECTURE: De-RISC MPSoC

- **General Purpose Processing (GPP)**
  - Cluster with multiple cores and a shared L2 cache memory, connected through AMBA AHB2 High Speed Interconnect
- **IO Subsystem**
- **Shared L3 cache and DDR memory controller**



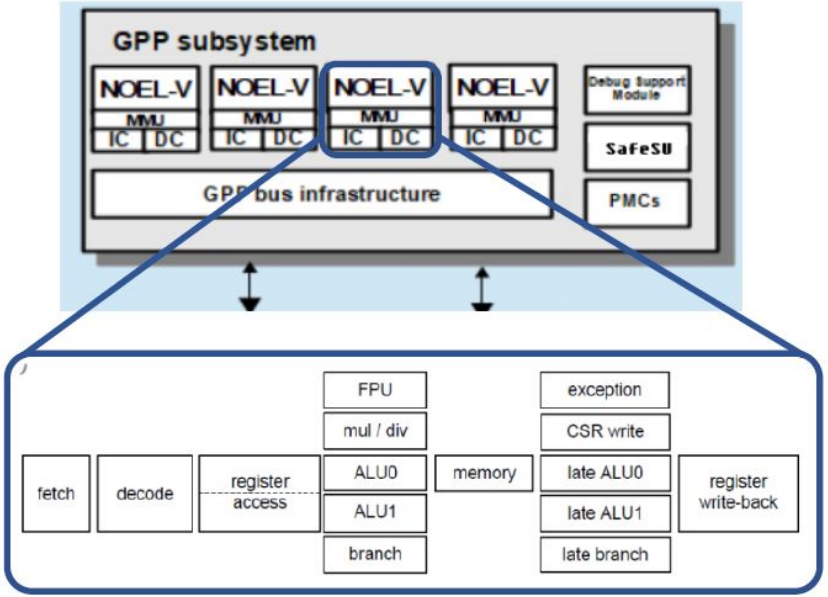
HARDWARE ARCHITECTURE: De-RISC MPSoC

- **Flexible and extensible multicore architecture**
  - Multiple clusters can be set and connected, sharing an L3 cache memory
  - Accelerators can also be deployed
- **De RISC delivering the MPSoC on an FPGA (Xilinx Kintex Ultrascale)**
  - ASIC deployments are part of future work (radiation hardened technology is mandatory for space)
- **Setup fitting FPGA capacity is being integrated and validated**
  - A single GPP with 4 cores
  - Neither L3 cache nor accelerators



HARDWARE ARCHITECTURE: NOEL-V

- **64 bit processor:**
  - RISC V cores w/ dual issue in order pipeline
  - Supporting RV64GCH extensions
  - 4 fully pipelined integer units (+ late ALUs)
  - Integer multiplications and divisions
  - FPU (floats and doubles)
  - MMU, PMP, branch predictors
  - Data and Instruction L1 caches
- **Fault tolerance**
  - Core local and cache SRAMs ECC protected
  - DDR2/3 SDRAM Memory controller with strong error correction code to achieve double device correction capability: can deliver correct data despite one full device failure and random SEU induced errors on the other devices

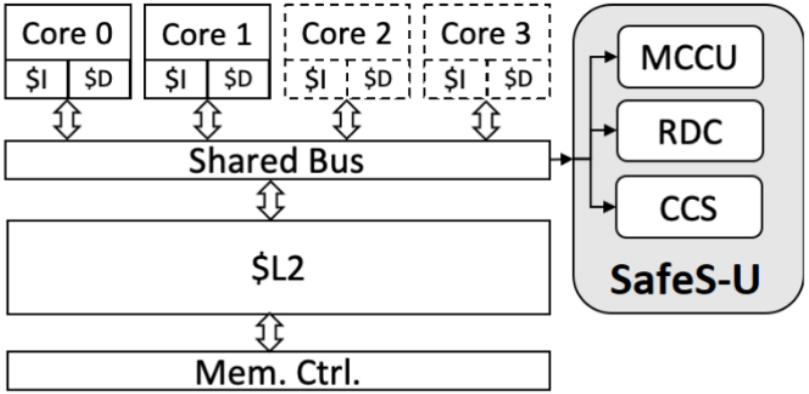


HARDWARE ARCHITECTURE: Communication interfaces and peripherals

- Memory interface
  - The SoC also has a **NAND Flash interface** to provide a large non volatile memory storage
  - Boot memory is provided through a parallel memory interface to support existing MRAM and NOR Flash devices
  - Boot via a SPI memory interface is also supported
- The communication interfaces include:
  - High Speed Serial Link support through **SpaceFibre controllers**
  - **SpaceWire communication** links, connected to an on chip router
  - 10/100/1000 Mbit **Ethernet interfaces**
  - SPI, I2C, UARTs, GPIO

HARDWARE ARCHITECTURE: SafeSU Statistics Unit

- Connected to the cluster local AMBA bus
- **Timing verification → RDC**
  - Collects maximum latency values for WCET estimation
- **Timing validation/diagnosis → CCS**
  - Statistics on how much contention (interference) each core causes on every other core
- **Implementation of safety measures related to timing → MCCU**
  - Allows setting interference quotas and raise interrupts when user-defined interference quotas are exceeded

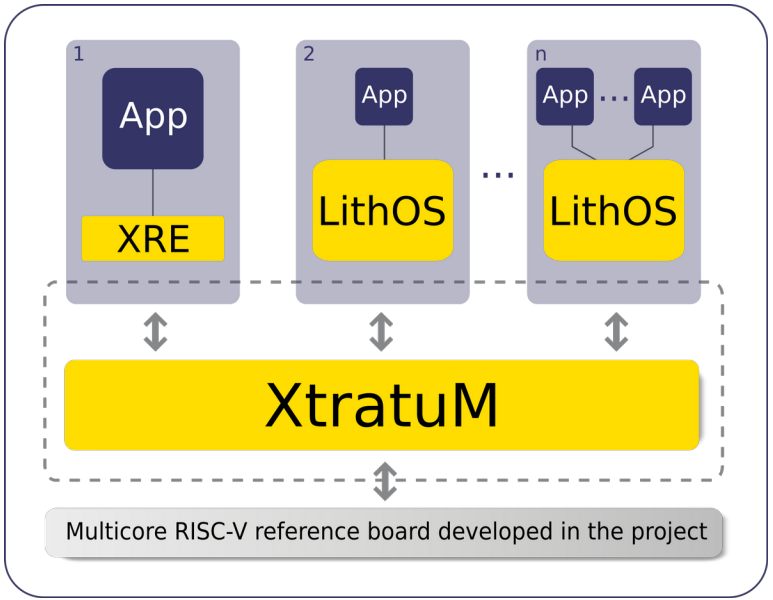


HARDWARE ARCHITECTURE: SafeSU Statistics Unit

- **Request Duration Counter (RDC)**
  - Counts cycles since a request is issued until its response arrives releasing the bus
  - Keeps maximum latency observed per request type (e.g., read, write, locked, burst length)
  - Used for WCET estimation : max_interference = num_requests × max_latency
- **Cycle Contention Stack (CCS)**
  - Measured stall cycles in the bus per core: implemented as a table of N × N-1 counters, where N is the core count
  - Broken down per contender (e.g. , M cycles stalled by core Ci , K cycles stalled by core Cj)
  - During testing provides info on how much each core interferes in the others
    - Needed for overrun diagnosis, and for optimization
- **Maximum Contention Control Unit (MCCU)**
  - Allows setting total interference quotas per contender core and offended core
    - e.g. core Ci can cause up to N stall cycles to core Cj
  - Monitors actual or maximum interference generated, subtracting it from the actual quota
  - Interrupt raised when a quota is exhausted
  - Software released from actively monitoring interference

SOFTWARE ARCHITECTURE: XtratuM Hypervisor

- **fentISS XtratuM Next Generation (XNG) hypervisor**
  - Different execution environments:
    - XtratuM Runtime Environment (XRE) for execution of bare metal applications
    - LithOS guest operating system by fentISS
    - Other guest operating systems
- **XNG fits RISC-V ISA and exploit Cobham Gaisler's MPSoC hardware features**
  - Mitigation of multicore interference through shared cache partitioning
  - Interfacing with BSC's SafeSU
  - Fault tolerance support
- **XNG hypervisor features for safety critical systems**
  - Partition management through the invocation of hypercalls
  - Support for system and non system partitions
  - Resource virtualization : making virtual resources available to a partition as if it were the only one using a given resource in the system
  - Temporal partitioning based on a cyclic scheduling policy
  - Spatial partitioning based on the support provided by the hardware
  - Inter Partition Communication (IPC) IPC): through sampling and queuing ports
  - Health Monitor (HM) service detects faults in hardware and in XNG itself
  - XtratuM Configuration File (XCF): a set of XML files to allow the system integrator to configure the system

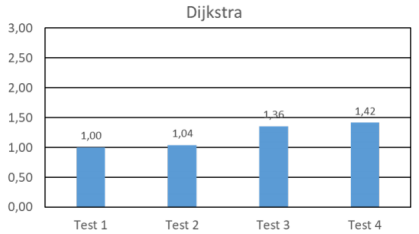


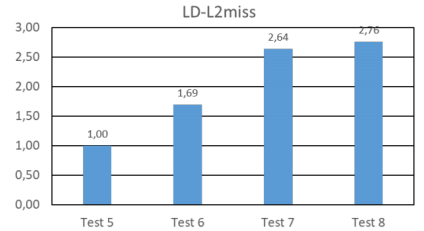
PERFORMANCE VALIDATION

- Current prototype results
- Basic validation:
  - Basic RISC V ISA and the implemented extensions (e.g. M, A, and FD, etc…) work properly
- Stress tests:
  - Dijkstra or LD L2miss→ Task under analysis
  - LD DL1hit → contender
  - LD L2miss → contender
- Normalized execution time for Dijkstra:
  - More memory contenders → higher execution time
  - The SoC allows scaling performance for applications taking advantage of local caches, despite having aggressive contenders
- Normalized execution time for LD L2miss:
  - Expose much higher multicore interference
  - LD L2miss still achieves some performance gains w.r.t. the single core case
- We expect scaling to be improved:
  - Addressing MC interference through a move to a multi layer bus structure

TEST	Core 0	Core 1	Core 2	Core 3
Test1	Dijkstra	LD-DL1hit	LD-DL1hit	LD-DL1hit
Test2	Dijkstra	LD-DL1hit	LD-DL1hit	LD-L2miss
Test3	Dijkstra	LD-DL1hit	LD-L2miss	LD-L2miss
Test4	Dijkstra	LD-L2miss	LD-L2miss	LD-L2miss

TEST	Core 0	Core 1	Core 2	Core 3
Test5	LD-L2miss	LD-DL1hit	LD-DL1hit	LD-DL1hit
Test6	LD-L2miss	LD-DL1hit	LD-DL1hit	LD-L2miss
Test7	LD-L2miss	LD-DL1hit	LD-L2miss	LD-L2miss
Test8	LD-L2miss	LD-L2miss	LD-L2miss	LD-L2miss



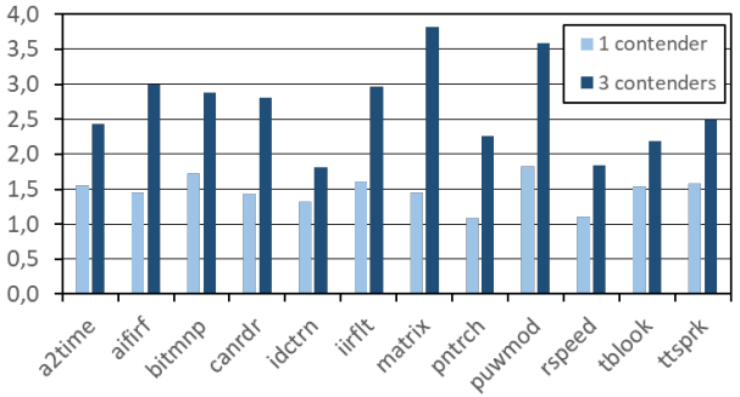


PERFORMANCE VALIDATION

- Current prototype results:
  - NOEL-V core @ 80 MHz (High performance config):
  - EEMBC CoreMark benchmark: 4.41 for CoreMark/MHz
  - Whetstone : 14.3 Millions of Whetstones/second

PERFORMANCE VALIDATION

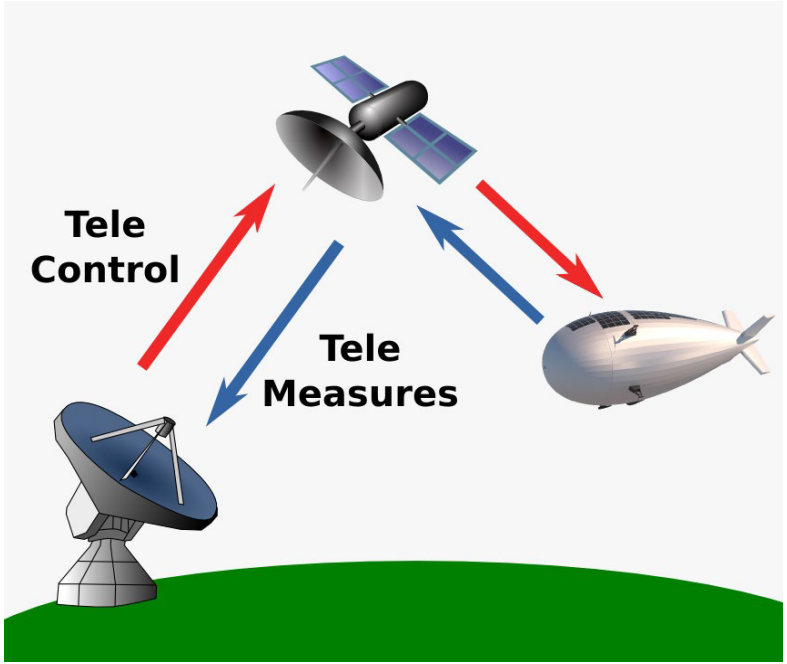
- Current prototype results:
  - TACLeBench benchmark suite
  - EEMBC Autobench (in figure)
    - All cores attempt to access shared resources at the same time
    - Execution time results normalized wrt the isolation case
    - The MPSoC allows sharing resources with slowdowns below 2 x and 4 x with 1 and 3 contenders respectively



USE CASES

FIRST USE CASE: COMMAND & DATA HANDLING SUBSYSTEM

- This representative application was originally developed as part of the EMC² Artemis project by Thales Alenia Space.
- Purpose: to evaluate the usage of multi-core processors in the context of mini- and micro-satellite constellations.
- Both the LEON4FT-based GR740 SoC and a previous version of fentISS XtratuM hypervisor have been evaluated with regards to their ability to handle space and time partitioning in a multicore context, and to perform inter-partition communications.
- Porting the C&DH application to the De-RISC platform will allow us to directly compare the evolution of the time and space isolation support that are key requirements for safety and security, and to assess the mechanisms we propose with regards to these properties.



USE CASES

SECOND USE CASE: LOW-LEVEL BENCHMARK EXECUTION

- Execution of two low-level benchmarks: bare metal execution and XNG XRE (XtratuM Runtime Environment, a minimal system to run applications with no guest operating system).
- This use case will validate the basic functionality of the architecture and will provide simple performance estimation to compare the architecture performance to other solutions available on the market. The several levels of such benchmark applications will include the checking for the compliance with RISC-V standards, some providing single-value performance scores, and the evaluation of the impact of the memory hierarchy and shared hardware resources on safety and security requirements.

THIRD USE CASE: ON-BOARD SATELLITE SOFTWARE STACK

- Use of LVCUGEN framework, a generic on-board framework developed by the French National Center for Space Studies (CNES) using XtratuM and LithOS.
- The purpose of this framework is to provide common on-board functions which are re-usable for different missions to minimize the satellite integration effort and ease the testing activities.
- Through LVCUGEN, critical partitions performing typical operations of a satellite (such as ground-to-satellite communications, positioning, mission control, etc.) and non-critical partitions performing payload computation, such as image compression, run in the same computer without interfering one another thanks to XtratuM and innovative hardware features minimizing interference channels.


GOALS AND OBJECTIVES

GOALS

G1: to become European leaders in the supply of an Integrated Modular Avionics HW and SW platform.

G2: to reduce the gap between Europe and USA in real-time embedded processors technology and in the SW support of mixed-criticality systems in general and of critical aerospace systems in particular.

G3: to setup a new and widely accepted Open Source ISA for the embedded market in Europe after the slowing down of innovations taking place in the SPARC open architecture, thus capitalizing on the economy of scale.

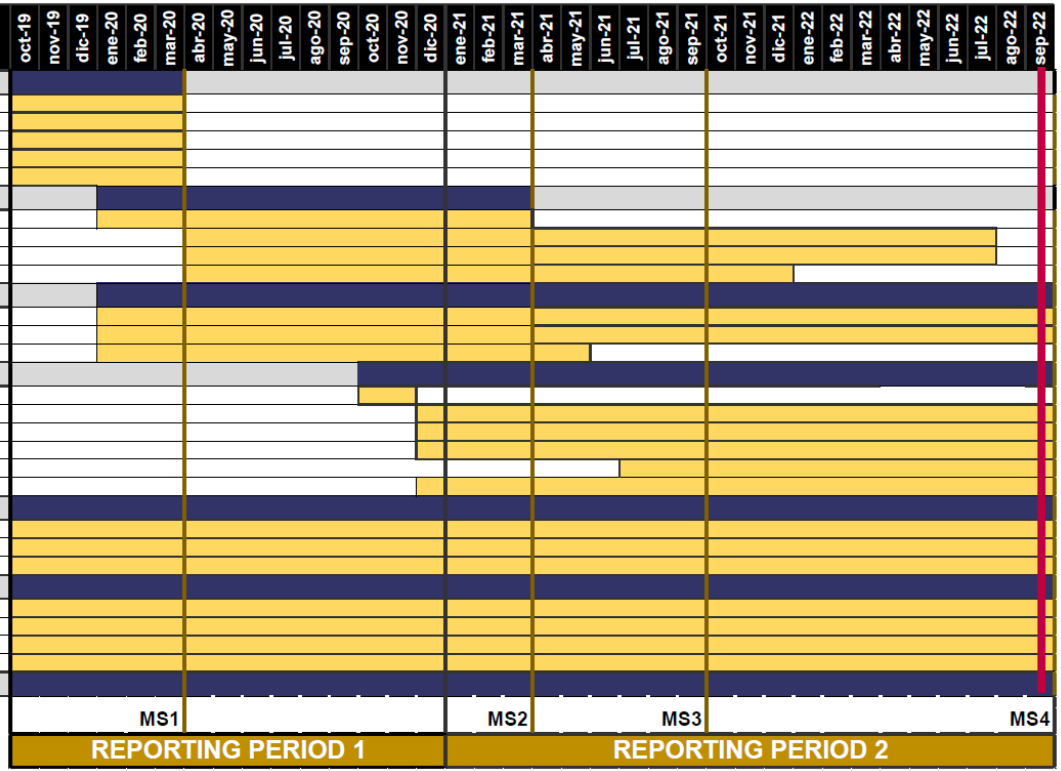
OBJECTIVES

- **O1**: Establish baseline system-on-chip multicore platform. It contributes to goal G2.
- **O2**: Introduce support for incremental software verification. It contributes to goal G2.
- **O3**: Develop prototype board with a short path to flight board. It contributes to goals G1, G2 and G3.
- **O4**: Port state-of-the-art hypervisor software to the new hardware platform and perform the activities required to certify it. It contributes to goals G1 and G3.
- **O5**: Perform radiation testing of the HW/SW platform developed. It contributes to goals G1, G2, G3.
- **O6**: Establish and market the hardware and the software platforms and introduce an integrated platform for the space and for the aeronautical industries with a TRL 8. It contributes to goal G1.


PROGRESS

General progress

WORK PACKAGES AND TASKS	Lead
WP1: Platform Requirements Definition	**TRT**
Task 1.1: Hardware Requirements	FEN
Task 1.2: Sofrware Requirements	TRT
Task 1.3: Initial HW platform setup	CG
Task 1.4: Board requirements definition	CG
Task 1.5: Real-time and security requirements	TRT
WP2: Hardware Platform Integration	**CG**
Task 2.1: System-on-chip setup	CG
Task 2.2: Functional System-on-chip Verification	CG
Task 2.3: Non-functional System-on-chip Features Verification	BSC
Task 2.4: Board design, manufacture and SoC integration	CG
WP3: Software Platform Integration	**FEN**
Task 3.1: XtratuM Hypervisor setup	FEN
Task 3.2: Software toolchain integration	FEN
Task 3.3: Use-case definition and implementation	TRT
WP4: Validation	**CG**
Task 4.1: Definition of the validation strategy	BSC
Task 4.2: HW validation	CG
Task 4.3: SW validation	FEN
Task 4.4: End-user Platform Validation	TRT
Task 4.5: Platform radiation Validation	CG
Task 4.6: Multidomain assessment	BSC
WP5: Dissemination, Communication and Commercialization	**FEN**
Task 5.1: Communication strategy, planning and tools	BSC
Task 5.2: Technology and market watch	CG
Task 5.3: Platform commercialization plan	FEN
WP6: Project Management	**FEN**
Task 6.1: Legal, financial and administrative issues	FEN
Task 6.2: Scientific and technical coordination	FEN
Task 6.3: Internal Communication, Quality and Risk Management	FEN
Task 6.4: Data management plan	BSC
WP7: Ethics Requirements	**FEN**



De-RISC

Dependable Real-time Infrastructure for Safety-critical Computer

www.derisc-project.eu

@DeRISC_H2020_EU

De-RISC

info@derisc-project.eu

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement EIC-FTI 869945