



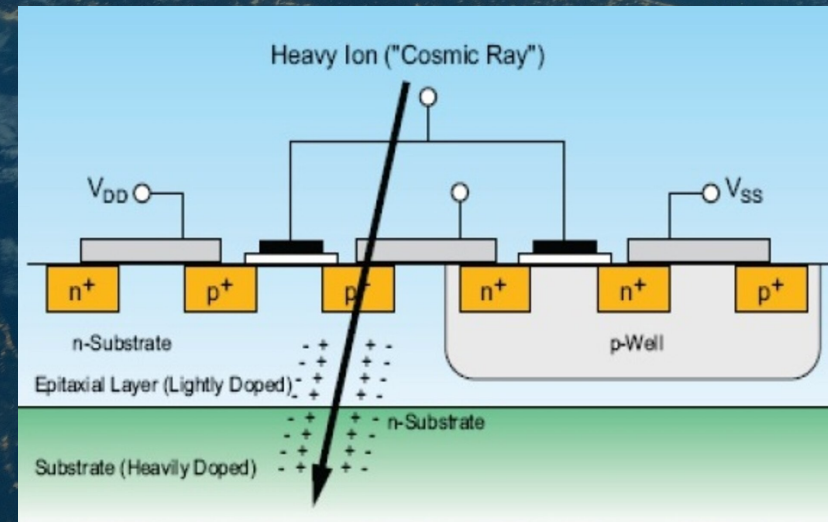
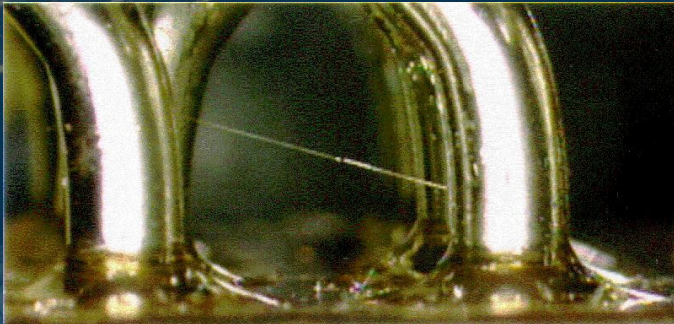
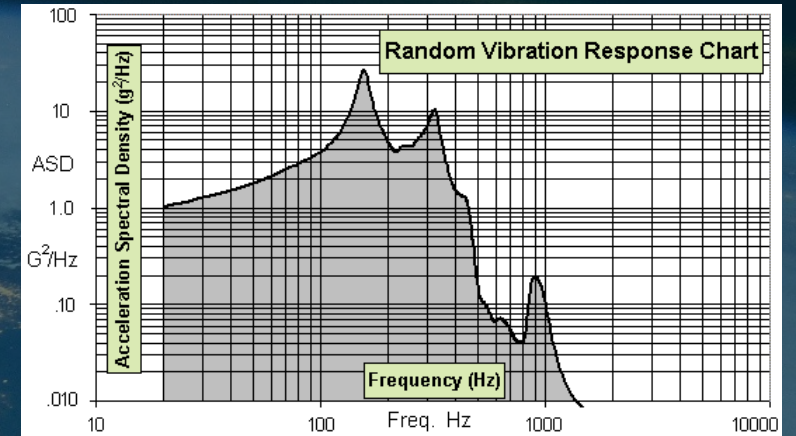
# RISC-V: a Rising Star in Space

Roland Weigand, European Space Agency – Microelectronics Section  
RISC-V Summit Europe – 8<sup>th</sup> June 2023

- **Constraints in Space**
  - Reliability : Radiation, Vacuum, Vibration, Temperature, Whiskers
  - Functional : Determinism, specific interfaces
  - Economics : Custom design with low part counts
  - Organisational: Long product life cycles, supply chain, public funding, geographical return
- **The use of open ISA for space – general motivation**
  - SPARC – a success model in Europe and beyond
  - Why changing ISA?
  - RISC-V – opportunities and challenges
- **RISC-V developments relevant for space**
  - GR765 GPP: Scalar General Purpose Processor
  - HPC: High Performance Computing : from accelerators to RISC-V Vector extension (RVV)
  - NASA HPSC – transition to RISC-V
  - Space Microcontrollers – transition to RISC-V
  - RISC-V already flying in space
- **Conclusion**

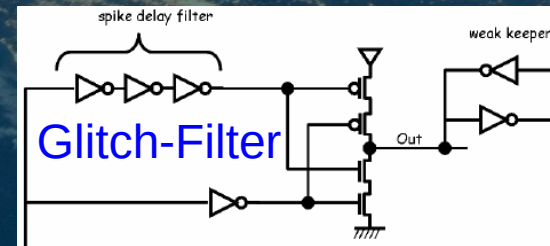
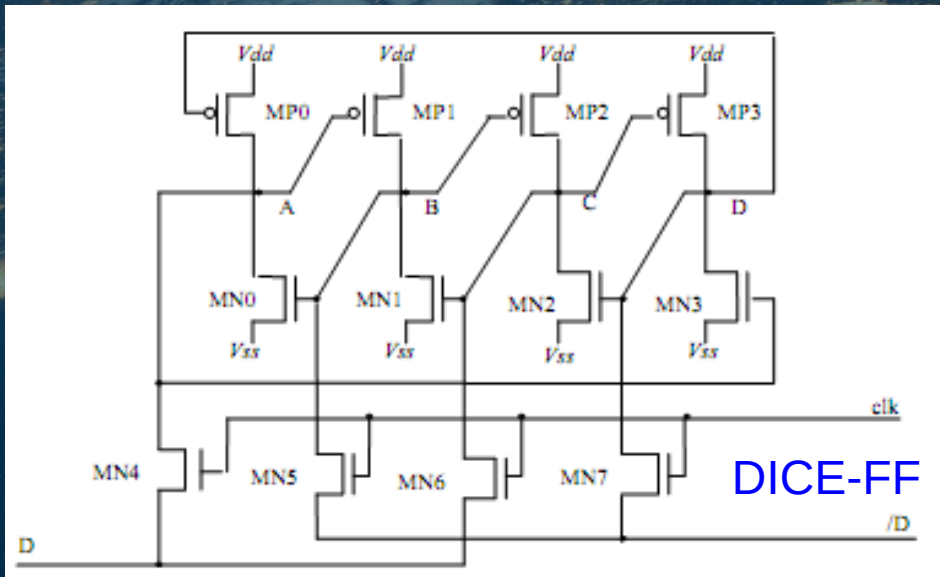
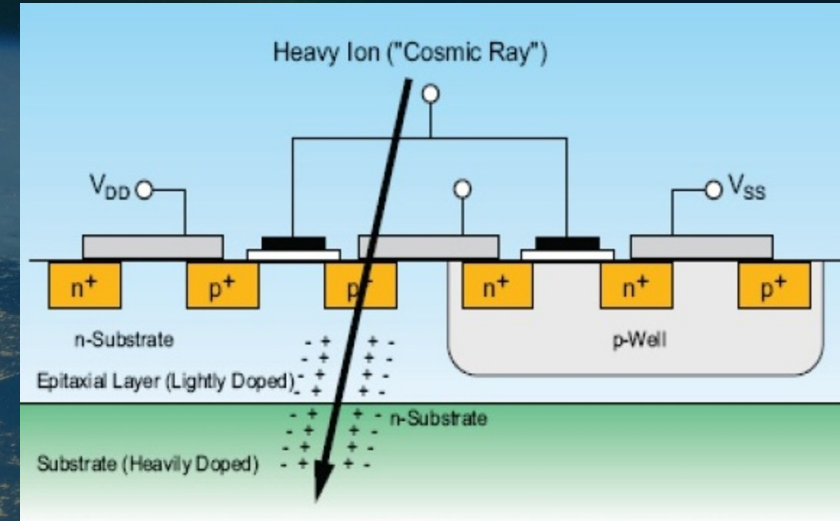
# Reliability Constraints in Space

- Space radiation: Total Ionising Dose (TID) and Single Event Effects (SEU)
- Commercial Off The Shelf (COTS) devices: thorough testing and system level mitigation
- Custom design with mitigation (RHBD = Radiation Hardening By Design)
- A satellite can be very hot or very cold : wide temperature range (often -55 to 125 deg C)
- Thermal dilatation leading to defects interconnect (bumps / balls / columns)
- Challenge for timing closure
- High temperature degrades silicon life time
- Strong vibration (low frequency) at launch: stress on interconnect, bondwires
- Vacuum: Outgassing → contamination of sensors
- Vacuum arcing “tin whiskers” → leaded soldering still used in space
- Life time of satellites up to 20 years >> than commercial electronics
- **No repair!**



# Radiation Hardening by (library) Design

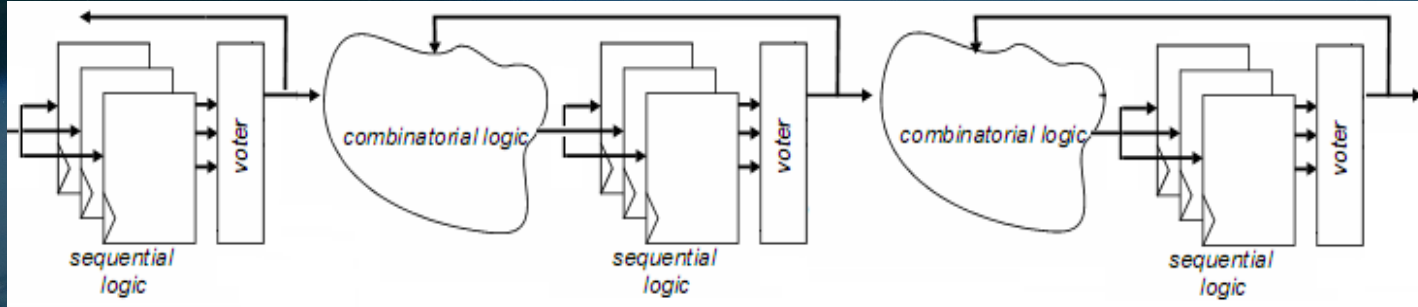
- Neutrons, Protons, Heavy Ions cause carrier generation in semiconductor
- Single Event Latchup (SEL): mitigation by process and standard cell library design
- Single Event Transient (SET): Glitches – a priori harmless in synchronous design, but ...
  - can be latched into a storage cell (flip-flop, memory) and become SEU
  - increasing problem in advanced technology due to high clock frequency
  - mitigation by temporal (glitch) filtering buffers, in particular in clock trees
- Single Event Upset (SEU): Bit-flip in storage cells → loss of data or state → **Redundancy**
- RH flip-flop in a standard cell library, e.g. DICE (Dual Interlocked Storage Cell)



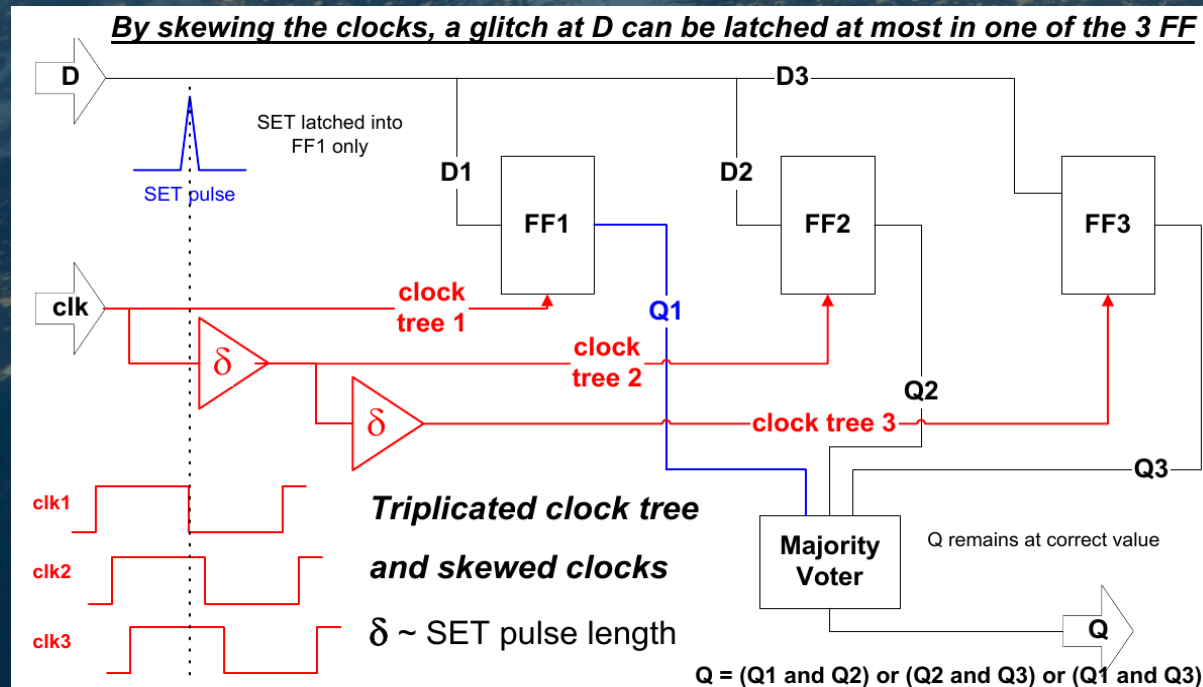
- Inserted during synthesis and place&route, **no RTL modification**
- Area overhead (2x-3x), also timing and power overhead
- **Penalty too high for bulk memories** (cache, register files)

# Triple Modular Redundancy (TMR)

- TMR using non-hardened flip-flops from any commercial standard cell library



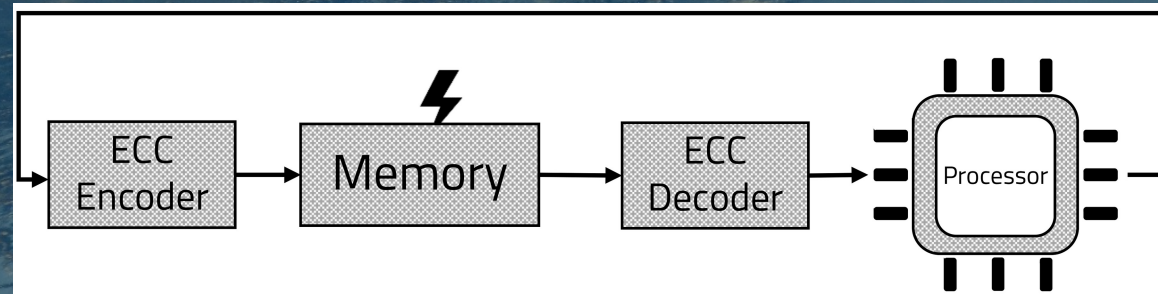
- STMR : TMR with triple skewed clock : covers SEU and SET



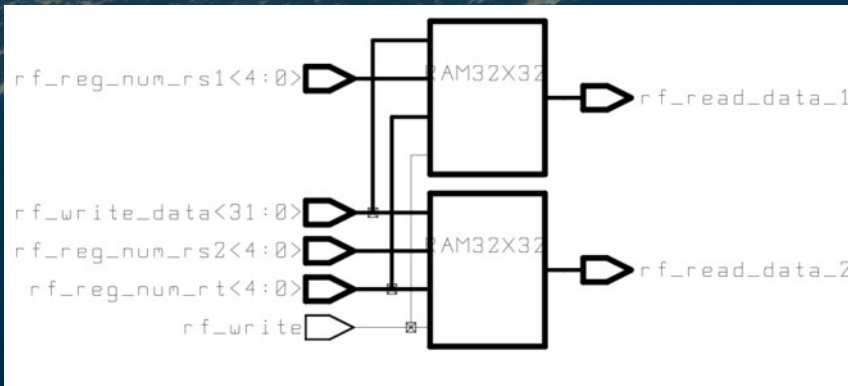
- Can be done with any commercial cell library
- Preferably inserted during or after synthesis
- $\delta$  delay  $\sim$  glitch duration (few 100 ps)  $\rightarrow$  timing penalty
- Tricky design flow : tools may try to remove redundancy
- STMR requires three coherent clock trees
- SEUs are corrected at next clock pulse, but...
- ... accumulation of SEU with clock gating
- AT697 (SPARC single core LEON2 @100MHz in 180 nm)
- Workhorse – flown to the back side of the moon
- TMR penalty too high for bulk memories** (caches...)

# Error Correcting Codes (ECC)

- For RAM blocks ECC are applied, e.g. Hamming / Bose-Chaudhuri-Hocquenghem (BCH) codes
- SEC-DEC : Single Error Correction and Double Error Detection in a memory word : 32 data + 7 parity = 39-bit codeword
- Parity bits generated at write, and errors corrected during read “on the fly” for the read-out data, but we need to correct the memory as well



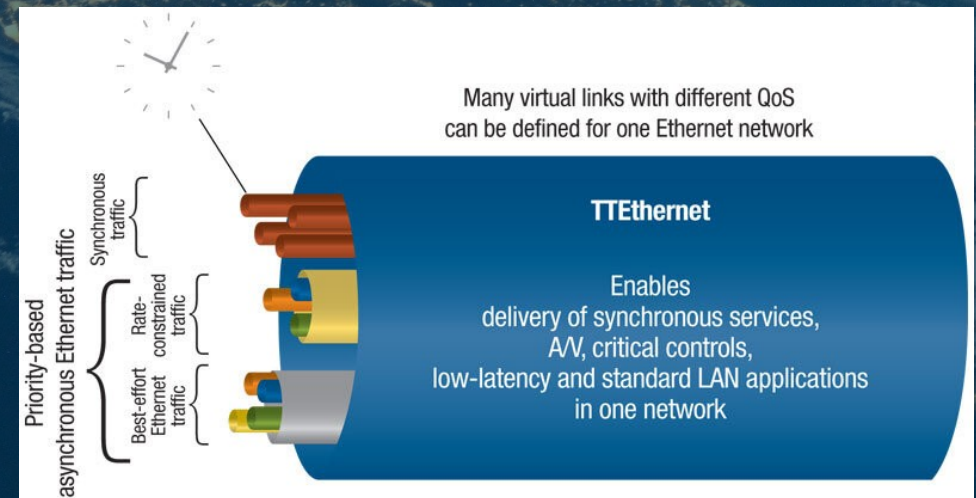
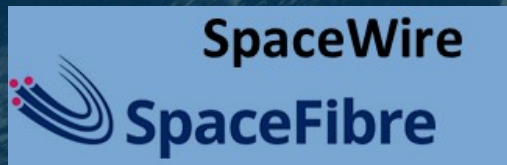
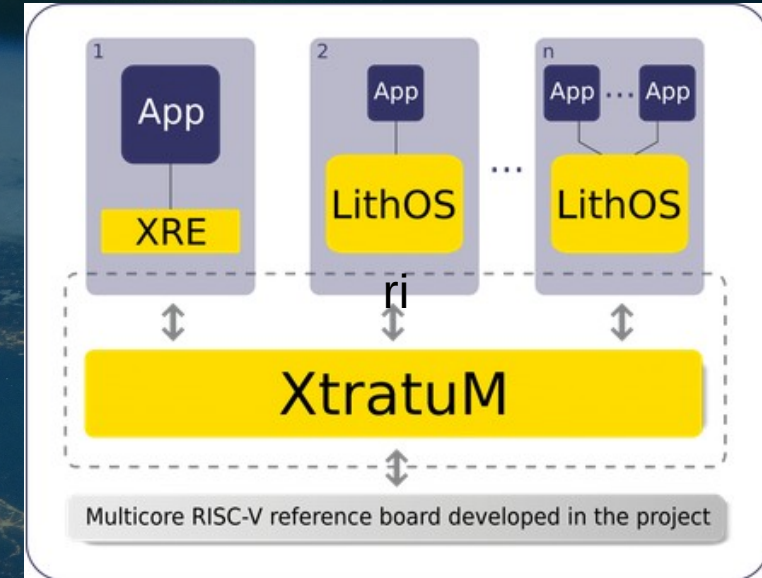
- For write-through cache or duplicate register files
- Simple parity bits (per byte): lower timing penalty
- Data can be restored from redundant copy of data



- Some SEU effects are also seen on earth, and affect high-rel applications
- Commercial processor IP often comes with built-in parity and ECC schemes  
→ sometimes not sufficient for space, stronger codes are required
- Smart recovery mechanism, not just a reboot, e.g. cache miss on parity error
- Minimize execution time impact of SEU handling
- Transparent to application SW : SEU handling in HW or by IRQ
- **We want to modify the control of cache and processor pipeline**  
→ often impossible with proprietary ISA and commercial IP cores

# Functional Constraints for Computers in Space

- Timing critical control functions (e.g. launch, landers) require deterministic processing
  - Careful containment of multi-core interference and caches
  - Use single core only? Switch off caches?
- Mixed criticality software : a spacecraft computer may execute SW from multiple origins
  - Different level of trust (high-rel platform vs instruments)
  - Fault containment by Time and Space Partitioning, use of hypervisor
- Specific space interface standards often inherited from commercial, automotive or military standards with additional constraints (determinism, electrical environment)
  - MIL-1553B (galvanic isolation, inherited from aeronautics)
  - SpaceWire (derived from IEEE1355 D/S encoding)
  - SpaceFibre (extension of SpaceWire to high speed SerDes)
  - Time Sensitive Networking (TSN) / TimeTriggered-Ethernet (TTE) synchronous, rate constrained and best effort traffic on the same link



# Economic and Institutional Constraints in Space

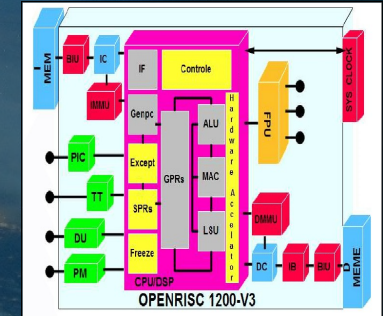
- **Recurrent part counts are low** → Design, CAD tool and mask cost dominate the economic equation, not silicon area
  - In the past, we made custom project specific ASIC for just a handful or few 10s of parts in a one-off satellite
  - Successful space standard products can reach 1000s or 10000s of parts – only
- **Long product life cycles**: “10+10+10” (thumb rule) = 10 years development, 10 years part sales, 10 years in orbit
  - Obsolescence issues : technology nodes, engineers, companies change owner or disappear
- **Expensive supply chain**: ceramic hermetic packaging, leaded interconnect, solder columns: expensive niche technologies
  - Space qualification batch requires dozens of parts for destructive testing
  - Recurrent prices dominated by packaging, assembly, testing, screening, not by silicon area
  - Price per space-qualified part can be > 50k€ (compare with commercial parts ~ 1 – 100\$)
- **Getting attention of commercial players** – ruled by investors and accountants – PR and prestige may be an incentive
  - Foundry backend services are there to fill the fab – they only support high volume products  
TSMC is producing 12 million wafers a year (2020) – our projects may order only 12 wafers
  - IP providers want short term revenue – royalties from space use may come years later → they ask for important licence fees
- **Public funding** : dependency on political decisions – a change of government may disrupt space projects
  - Multiple public funding organizations and schemes : ESA is only one of them, there are also national bodies, EC, ...
  - Spending rules : geographical return, consortium composition (LSI, SME, academia)



- **Constraints in Space**
  - Reliability : Radiation, Vacuum, Vibration, Temperature, Whiskers
  - Functional : Determinism, specific interfaces
  - Economics : Custom design with low part counts
  - Organisational: Long product life cycles, supply chain, public funding, geographical return
- **The use of open ISA for space – general motivation**
  - SPARC – a success model in Europe and beyond
  - Why changing ISA?
  - RISC-V – opportunities and challenges
- **RISC-V developments relevant for space**
  - GR765 GPP: Scalar General Purpose Processor
  - HPC: High Performance Computing : from accelerators to RISC-V Vector extension (RVV)
  - NASA HPSC – transition to RISC-V
  - Space Microcontrollers – transition to RISC-V
  - RISC-V already flying in space
- **Conclusion**

# Why using an open ISA...

- Keynote Jiri Gaisler, ADCSS 2017 <https://indico.esa.int/event/182/timetable>
- Why did ESA choose SPARC (30 years ago...)?
  - Non-dependence : multiple IP sources, or own development
  - Free and open Standard, allowing for customization
  - Need to modify for radiation hardening : difficult with proprietary ISA
  - Industry and academia backing: commercial and open source (Sun, Fujitsu)
  - Stable specification, low complexity, GNU tool chain
- Using open ISA does not come “for free”, and it does not always offer superior performance
  - Open ISA is not necessarily open source, and open source is not free source
  - Our first SPARC implementation was using a commercial IP
  - High quality IP always comes at a cost : “use free IP = spend fortune on validation”
- **Open ISA allows for competition of multiple IP sources → warrants for flexibility and non-dependence**



# SPARC in space > 25 years – 5 generations

- ERC32 – Based on a commercial SPARC-V7 IP-core, TSC69x series
- LEON2 SPARC V-8 IP-core owned by ESA → AT697F in 180nm
- LEON3 IP-core owned by CAES (aka Cobham Gaisler) → GR712RC
- LEON4 IP-core, introducing L2 cache → GR740 in 65nm, QML-V in 2022
- LEON5, introducing dual-issue → GR765 in 28nm, under development  
moreover, LEON IP is on many other ASIC and FPGA



SPARC is in all our missions

Europe, US, China, Japan

> 10000 flight parts

Standard processors

SPARC IP in FPGA

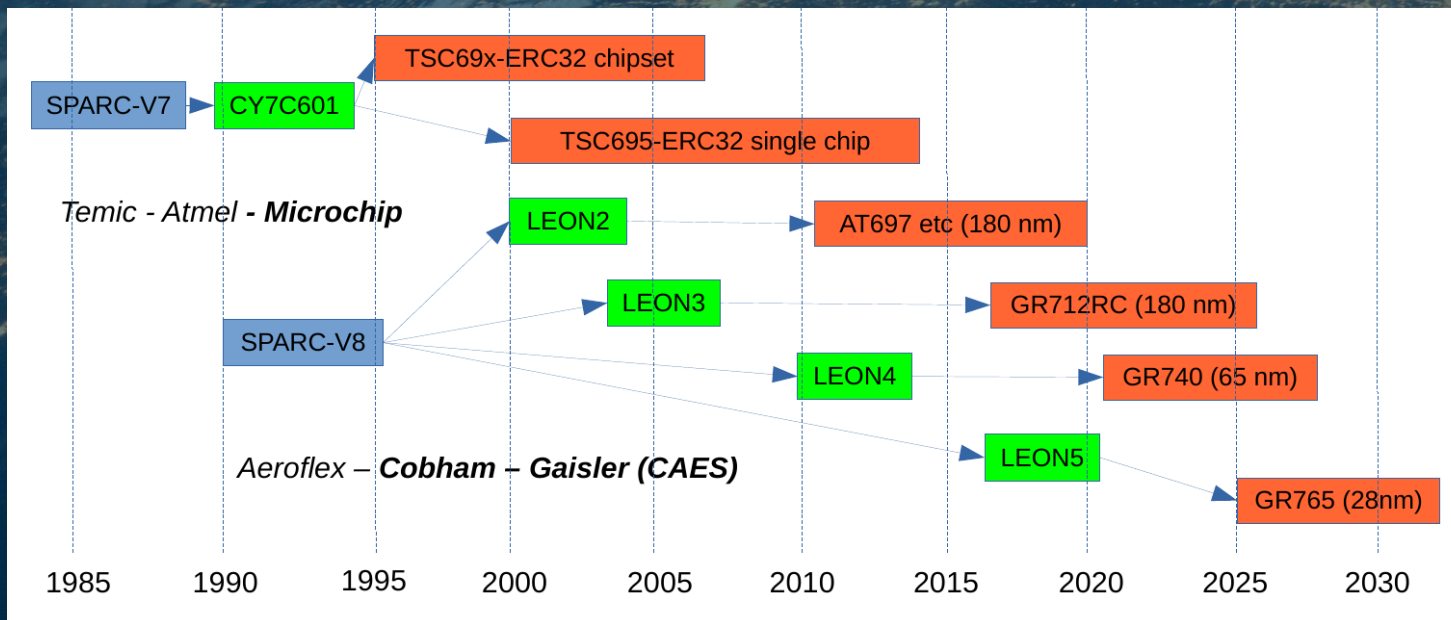
**A SUCCESS STORY**

mature SW ecosystem

demand in space continues

new design-ins to projects

still demand for LEON IP



**WHY SHOULD WE CHANGE?**



# Why RISC-V?

## Why not SPARC forever?

obsolete in the commercial world

lack of community backing

lack of developers

## Why not a proprietary ISA?

Yes, we do! - ARM is flying

Nanosplore space FPGA

No IP modifications

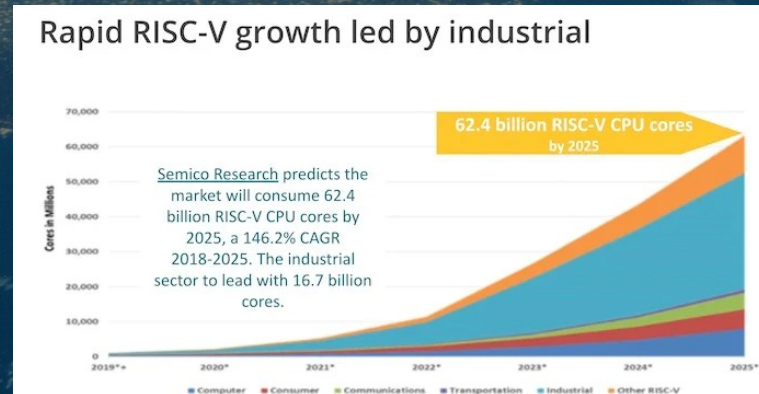
Dependency

# Why RISC-V?

Name	Description	Version	Status <sup>[A]</sup>
<b>Base</b>			
RVWMO	Weak Memory Ordering	2.0	Ratified
I	Base Integer Instruction Set, 32-bit	2.1	Ratified
	Base Integer Instruction Set (embedded), 32-bit, 16 registers	2.0	Ratified
RV64I	Base Integer Instruction Set, 64-bit	2.1	Ratified
RV64E	Base Integer Instruction Set (embedded), 64-bit	2.0	Ratified
RV128I	Base Integer Instruction Set, 128-bit	1.7	Open
<b>Extension</b>			
M	Standard Extension for Integer Multiplication and Division	2.0	Ratified
A	Standard Extension for Atomic Instructions	2.1	Ratified
F	Standard Extension for Single-Precision Floating-Point	2.2	Ratified
D	Standard Extension for Double-Precision Floating-Point	2.2	Ratified
Zicsr	Control and Status Register (CSR) Instructions	2.0	Ratified
Zifencei	Instruction-Fetch Fence	2.0	Ratified
G	Shorthand for the IMAFDZicsr_Zifencei base and extensions <sup>[1]:129</sup>	—	—
Q	Standard Extension for Quad-Precision Floating-Point	2.2	Ratified
L	Standard Extension for Decimal Floating-Point	0.0	Open
C	Standard Extension for Compressed Instructions	2.0	Ratified
B	Standard Extension for Bit Manipulation	1.0	Ratified
J	Standard Extension for Dynamically Translated Languages	0.0	Open
T	Standard Extension for Transactional Memory	0.0	Open
P	Standard Extension for Packed-SIMD Instructions	0.9.10	Open
V	Standard Extension for Vector Operations	1.0	Ratified
Zk	Standard Extension for Scalar Cryptography	1.0.1	Ratified
H	Standard Extension for Hypervisor	1.0	Ratified
S	Standard Extension for Supervisor-level Instructions	1.12	Ratified
Zam	Misaligned Atomics	0.1	Open
Zihintpause	Pause Hint	2.0	Ratified
Zihintntl	Non-Temporal Locality Hints	0.2	Open
Zfa	Additional Floating-Point Instructions	0.1	Open
Zfh	Half-Precision Floating-Point	1.0	Ratified
Zfhmin	Minimal Half-Precision Floating-Point	1.0	Ratified
Zfinx	Single-Precision Floating-Point in Integer Register	1.0	Ratified
Zdinx	Double-Precision Floating-Point in Integer Register	1.0	Ratified
Zhinx	Half-Precision Floating-Point in Integer Register	1.0	Ratified
Zhinxmin	Minimal Half-Precision Floating-Point in Integer Register	1.0	Ratified
Zmmul	Multiplication Subset of the M Extension	1.0	Ratified
Ztso	Total Store Ordering	1.0	Ratified

## Opportunities:

- Open ISA – just like SPARC – but RISC-V has active standardization
- Rapidly growing, **multiple IP sources**: commercial, open-source, free
- **Highly extensible**: ISA covers low-end to high-performance applications
- **Hypervisor** : suited for mixed criticality systems in space
- **RVV** – Vector Extension as key to HPC and AI applications
- Overcomes annoying “features” of SPARC (delay slots, register windows)
- Rapidly growing developer community at all levels (industry, academia)



## Challenges:

- RISC-V recent: initiated in 2010, high popularity after 2020, but space is conservative
- At ESA: no talks about RISC-V until 2016 – we are late, but going fast now
- Maturity of available IP to be confirmed – compliance tests, certification?
- Not clear if all extensions will “survive”, even the ratified ones
- Custom extensions are risky: SW support! - Probably not affordable for us

- **Constraints in Space**
  - Reliability : Radiation, Vacuum, Vibration, Temperature, Whiskers
  - Functional : Determinism, specific interfaces
  - Economics : Custom design with low part counts
  - Organisational: Long product life cycles, supply chain, public funding, geographical return
- **The use of open ISA for space – general motivation**
  - SPARC – a success model in Europe and beyond
  - Why changing ISA?
  - RISC-V – opportunities and challenges
- **RISC-V developments relevant for space**
  - GR765 GPP: Scalar General Purpose Processor
  - HPC: High Performance Computing : from accelerators to RISC-V Vector extension (RVV)
  - NASA HPSC – transition to RISC-V
  - Space Microcontrollers – transition to RISC-V
  - RISC-V already flying in space
- **Conclusion**

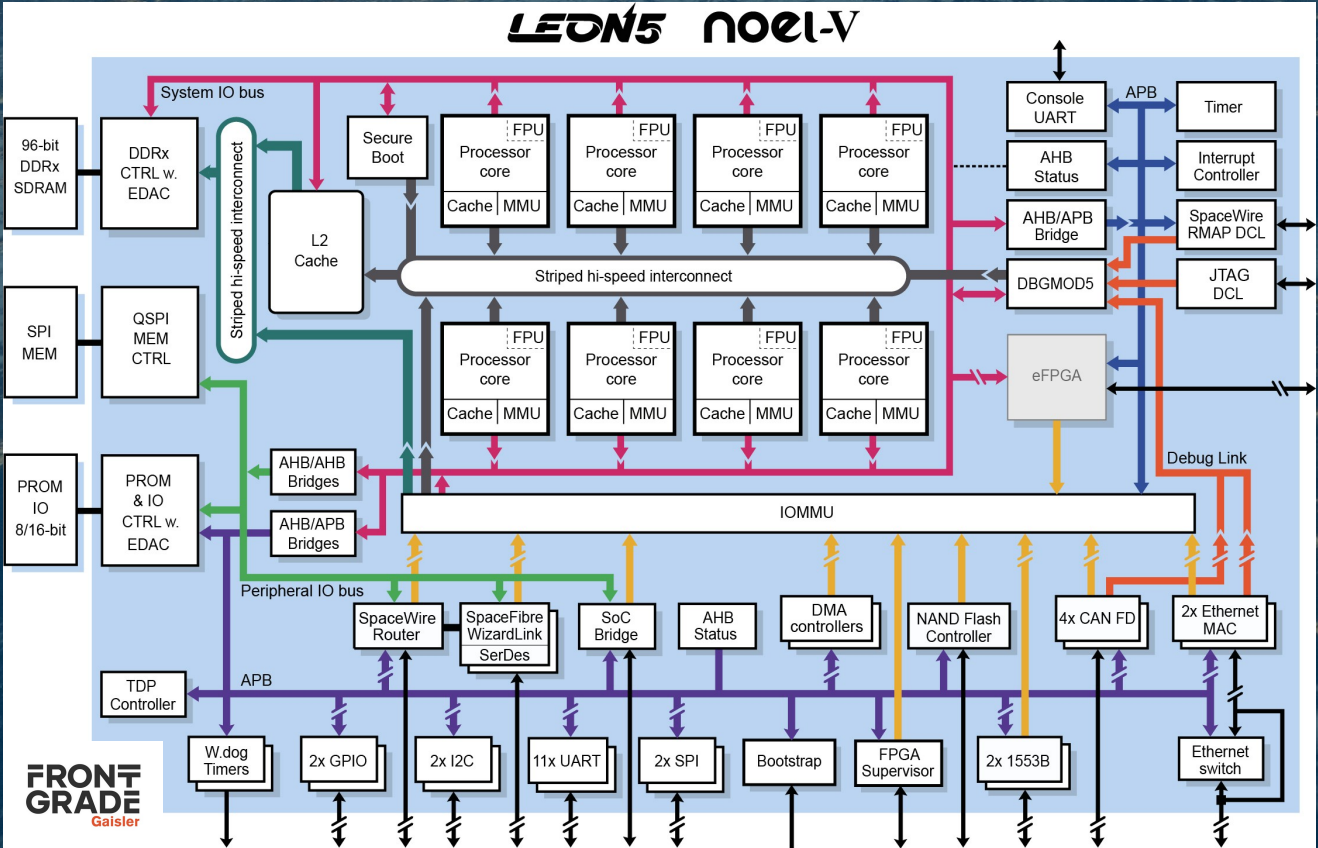
# GR765 – transition from SPARC ... to RISC-V



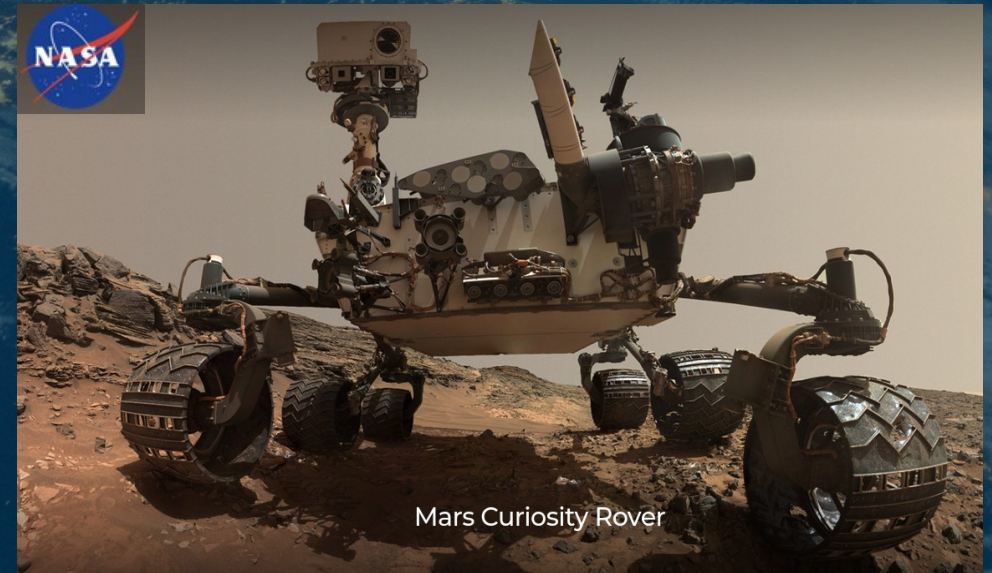
**One chip – two ISA: octacore chip in 28 nm – boot either in SPARC or in RISC-V mode**

Synergy for development and mask cost

Facilitate transition for users



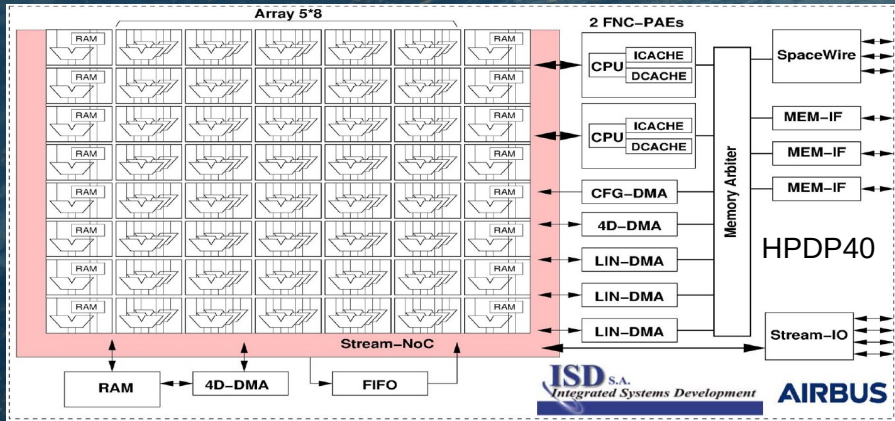
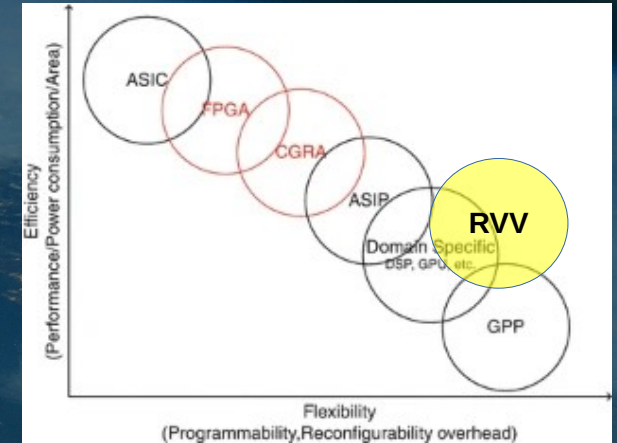
- Why HPC in space – Scientists usually want all data on ground for post-processing?
  - Bit-exact downlink required until a few years ago – only lossless compression was allowed
  - Computing in space used mainly for Attitude and Orbit Control System (AOCS) and data handling
- Recent requirements for HPC and AI
  - Instruments producing more and more data – downlink bandwidth remains limited and costly
  - Data (pre-) processing / reduction of massive amount of Earth Observation data
  - Autonomous planetary exploration: communication delay (3-22 min to Mars) → earth-controlled operation is very slow
    - HPC to improve precision and reliability of landers and increase mobility of rovers or helicopters
  - Telecom: Software Defined Radio and Digital Beamforming, 5GNTN
  - Cryptography and Quantum Key Distribution
- Many different algorithms are required...
  - Machine Learning CNN/RNN for feature / data selection, fault detection
  - e.g. cloud detection: eliminate useless EO images
  - Filtering, FFT, compression
- Common denominator: massive parallel multiplication / MAC?
  - → **Vector and matrix operations**



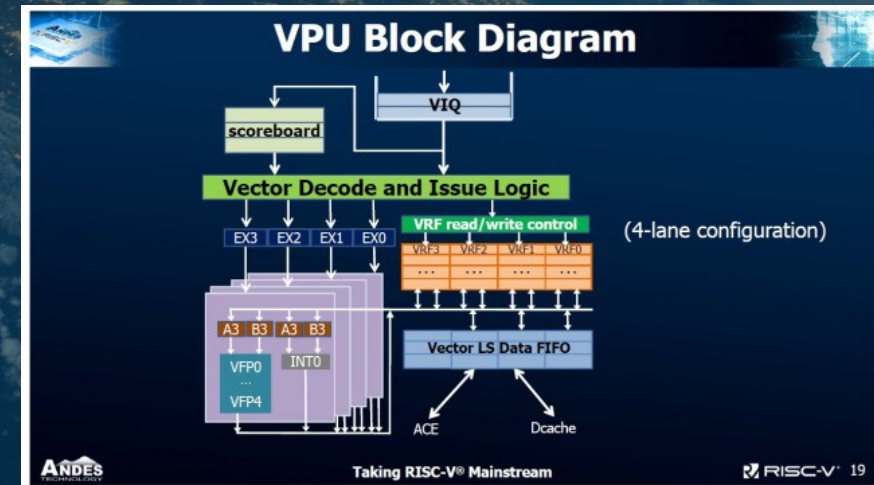


# HPC architectures – flexibility vs energy efficiency

- We need HPC, high flexibility, low power and affordable cost (chip / HW development and application/SW development)
  - ASIC – inflexible, unaffordable for low volume space applications
  - European RH FPGA : Nanoxplore (65 nm, 28 nm, 7 nm planned)
  - European RH CGRA: Airbus/ISD HPDP40, 40-core PACT-XPP (65 nm)
  - FPGA / CGRA : efficient only at high utilization, HW design tools critical (synthesis, P&R)

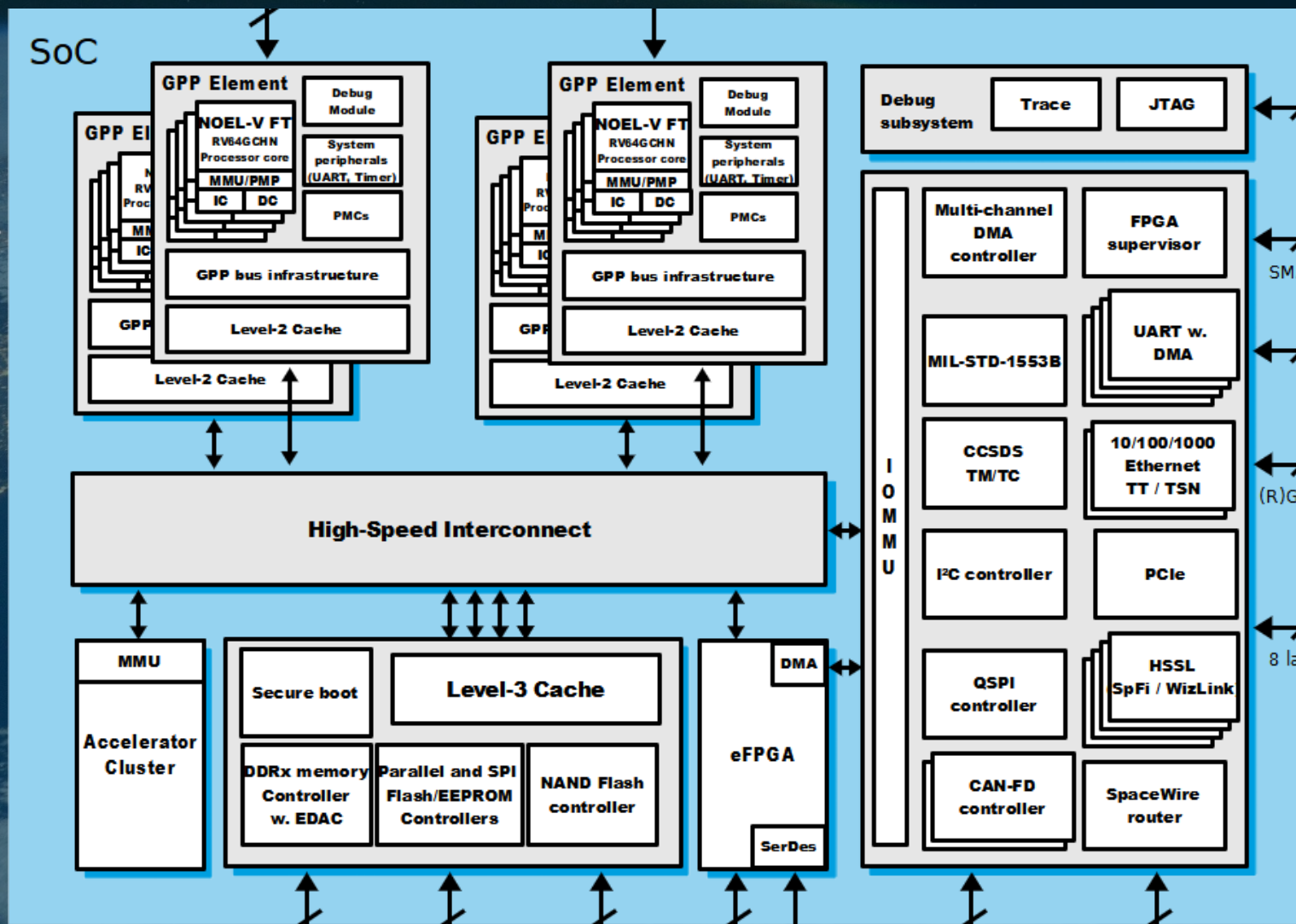


- Scalar GPP : low power efficiency – fetch 1 instruction per operation
- **GPP with mainstream parallel instruction extensions**
  - Case for RISC-V Vector Extension (RVV)
  - Will vector compilers, libraries, CNN models become available?
  - Major RISC-V IP vendors have adopted RVV

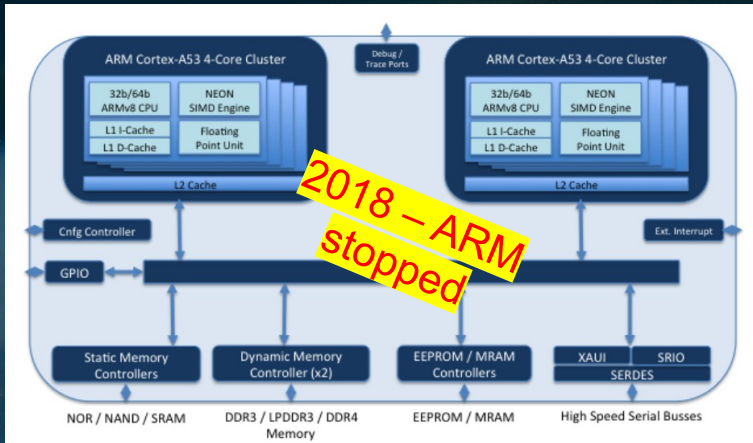
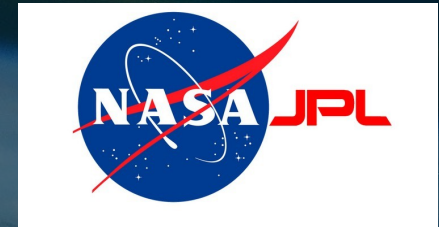


# GR7xV – HPC for space

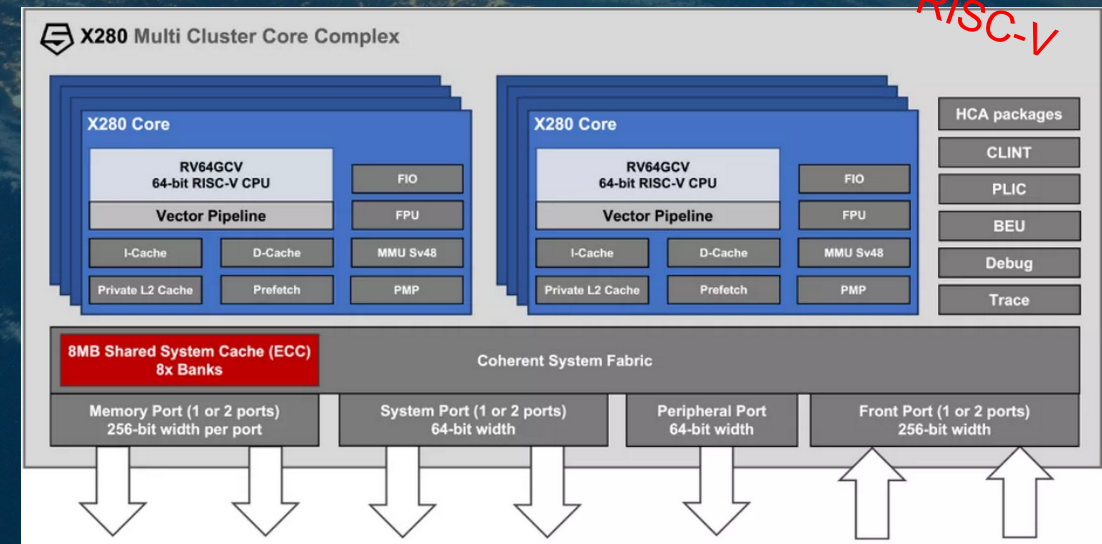
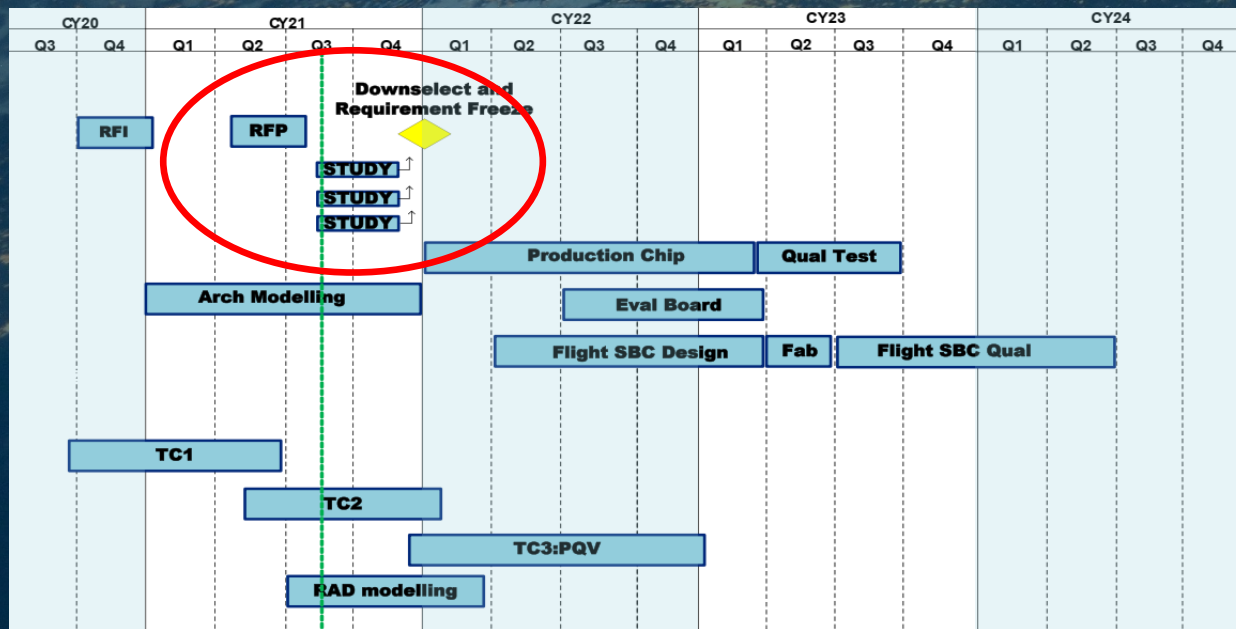
- Preliminary design, initially based on scalar NOEL-V
- Symmetric clustered (4x4 core) architecture
- Trade-offs
- **Vector extensions? - Yes! → RVV1.0**
  - VLEN? Data types (INT16, 8, 4, BFLOAT16, Posit)?
- Symmetric or asymmetric architecture?
  - See PULP-HERO concept
  - Microchip PolarFire-SoC : 4 + 1 RISC-V
  - NASA / Microchip HPSC  
4x RISC-V GPP + 8x VP (SiFive X280)
- Cache and memory architecture
  - **Challenge: shuffle data in/out for sustained vector processing**
- Accelerator? Which IP?
- Monolithic or chiplets:
  - FPGA / accelerator / IO on separate chips?
- **Work in progress**
- **Pending availability of 7nm for space**



# NASA High Performance Space Computing (HPSC)



- 2018 plans: 32 nm SOI, ARM Cortex A53 [W. Powell, RadHard Electr Techn 2018]
- 2021 refurbishment: 22 nm (or below), ISA open [J. Butler, Space Comp Conf 2021]
- 2021: 3 parallel architecture studies: **trade-off ARM and RISC-V?**
- 2022/08: contract to Microchip, using **RISC-V cores with Vector Extension** (SiFive X280), 12 nm <https://www.techspot.com/news/95911-sifive-risc-v-cores-microchip-processors-power-nasa.html>



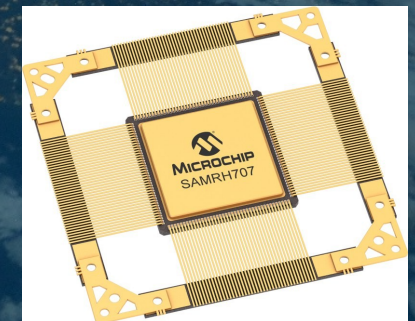
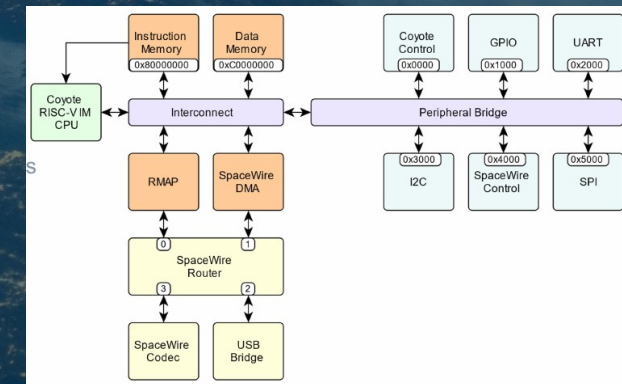
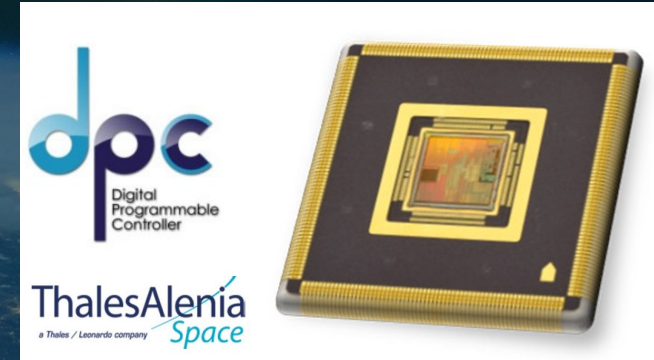
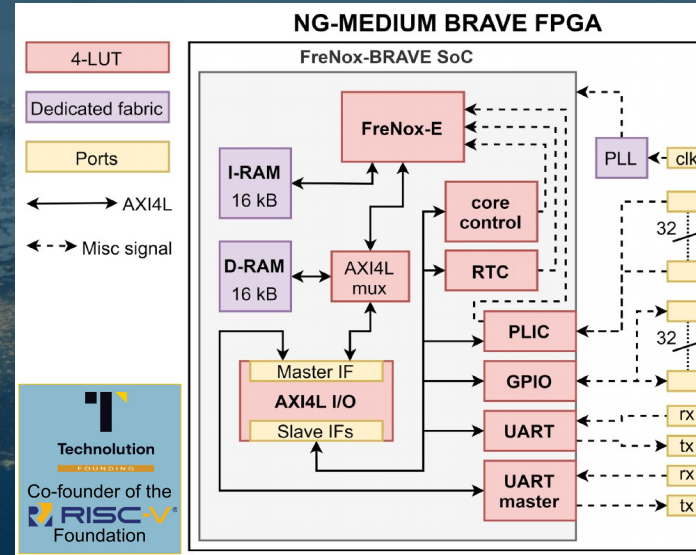
# Microcontrollers for Space

- Focus for microcontrollers is low overall (system) cost and determinism, rather than pure processing performance

- Mixed signal chip integration → reduce component count
- Low component price → use 'mature' technologies
- Easy board level assembly → simple package
- Complete and smooth SW development environment (SDE)
- Various ISA, often re-use from designs for commercial market
- ARM Cortex-M (e.g. Microchip SAMRH707)
- OpenMSP430 (Thales Alenia Space DPC)
- GR716 SPARC LEON3

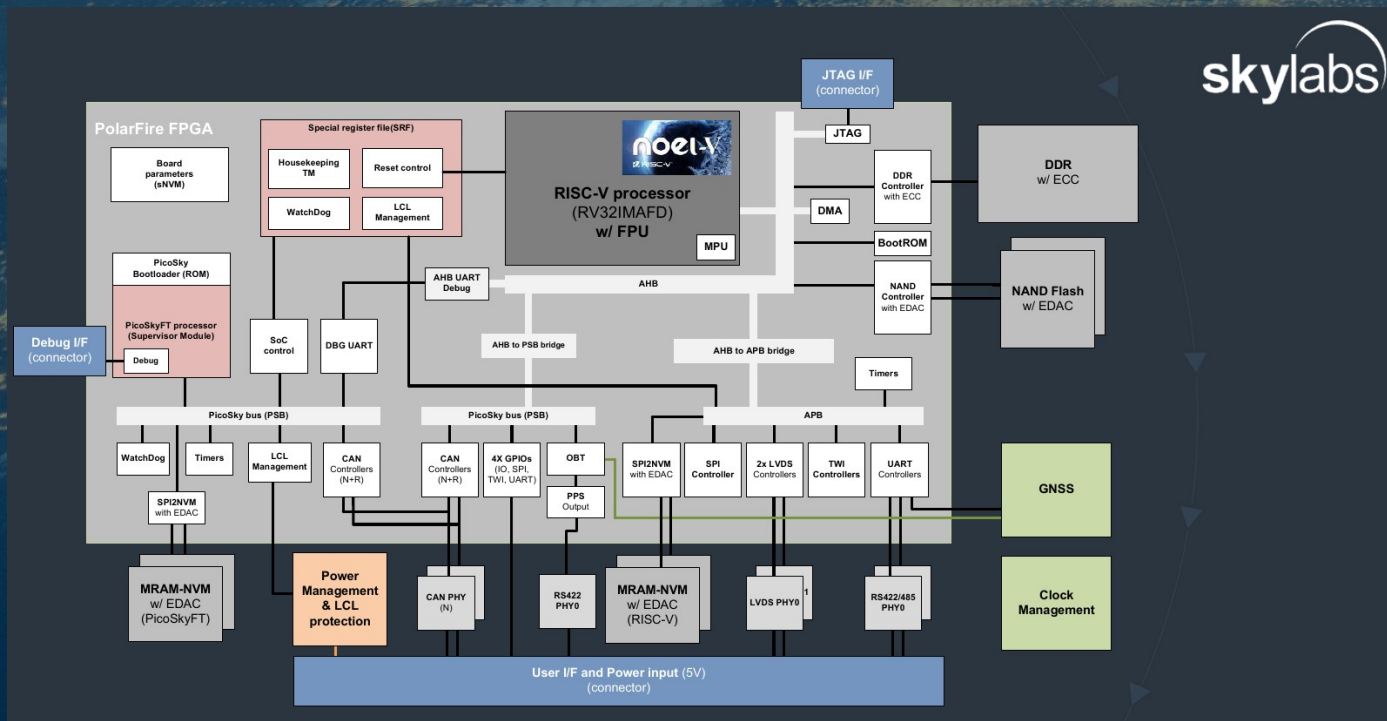
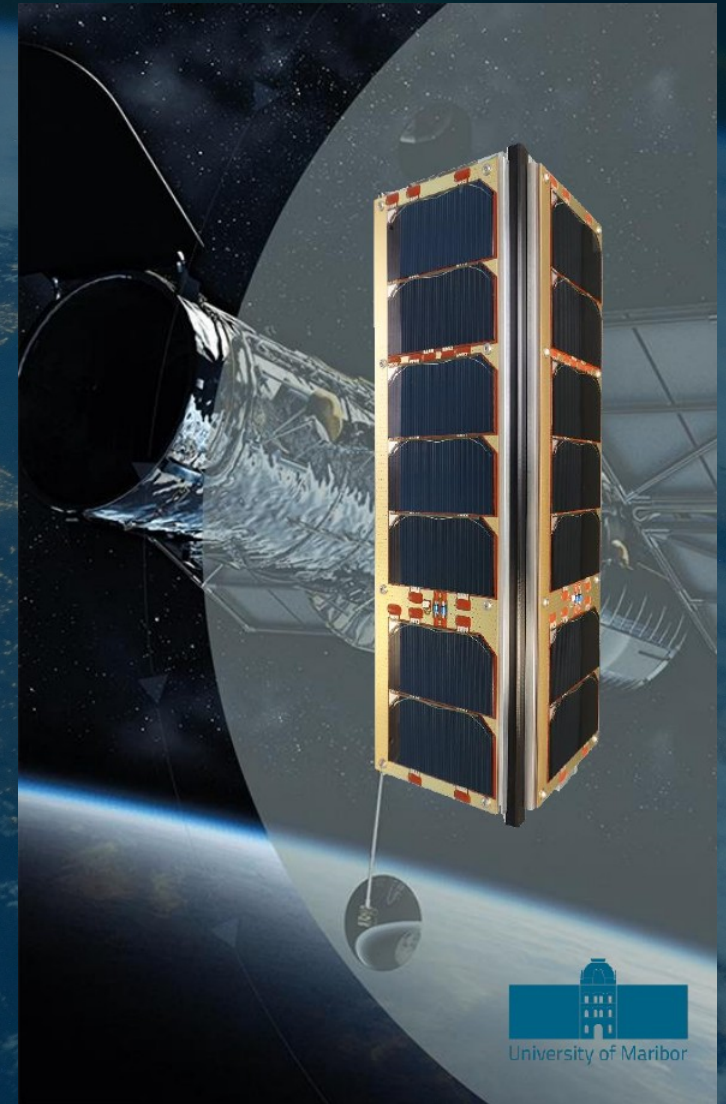
## Transition to RISC-V

- Technolution FreeNox : port from non-space into RH FPGA
- Coyote RV32IM SystemVerilog IP by Felix Siegle (ESA)
- Planned: Development of a RISC-V Microcontroller IP**
- ESA Technology Development Funding, Frontgrade Gaisler
- Smaller than NOEL-V, fully deterministic operation
- To become available on ESA IP portfolio for space applications
- Target soft-IP on (limited complexity) space FPGA
- Target standard microcontroller



# RISC-V is flying in space

- TRISAT-R
  - Nanosatellite by University of Maribor (SI), launched July 2022
  - Ionizing Radiation Measurement in Medium Earth Orbit (~6000km)
- NanOhpm-obc by Skylabs (SI)
  - NOEL-V soft-IP @ 50MHz on PolarFire FPGA
- Other non-disclosed use of RISC-V in space may exist...



- Chip design for space is different from commercial world
  - Reliability and environmental constraints, radiation hardening, different cost equation: low part count, public funding
- ESA has been successfully using SPARC open ISA for 3 decades
  - Open ISA allows for competition, ensuring non-dependence
  - Open ISA makes it easier to do modifications for rad hardening
  - Open ISA is not for free – mature IP has its price
- Transition to RISC-V continues the SPARC-in-space success model
  - Additional value by standardized RISC-V extensions
  - Careful with custom extensions: high effort to develop SDE
  - Certify IP, carefully watch perennity of extensions
- First ESA RISC-V chip: GR765 SPARC/RISC-V octa-core chip in 28 nm technology
  - Multi-standard: one chip – two ISA : smooth transition for users
- HPC in space: matrix / vector operations = parallel multiplication / MAC
  - Challenge: energy efficiency vs flexibility – limits for scalar GPP
  - Way forward: RISC-V with RVV Vector extension
- GR7XV: NOEL-V based architecture to be extended to RVV (7 nm TBC)
- NASA HPSC : switched from ARM (2018) to RISC-V (2022)
- RISC-V is already flying in a European space mission



**RISC-V is an opportunity – a rising star in space  
ESA promotes transition to RISC-V**