A three-perspective analysis of RISC-V design tools for safety & security architectures

Javier Mora (Javier.MoradeSambricio@collins.com), Alejandro García Gener, Gonzalo Salinas
Connected & Real-time Systems – Collins Aerospace Applied Research & Technology, Ireland

RISC-V Summit
Collins Aerospace | APPLIED RESEARCH & TECHNOLOGY

RISC-V Summit Europe
5-9 June 2023, Barcelona, Spain

Vision & Value

MOTIVATION

Embedded SoC design in **Aerospace** industry →

- Stringent **safety & security** requirements
    - High reliability
    - Fault tolerance
- Increasing demand of **computing power**
- Failure → **catastrophic** → need S&S to avoid them
- **COTS** processors:
    - **Unreliable** (not designed for S&S)
    - Not **versatile** enough

RISC-V can provide a solution!

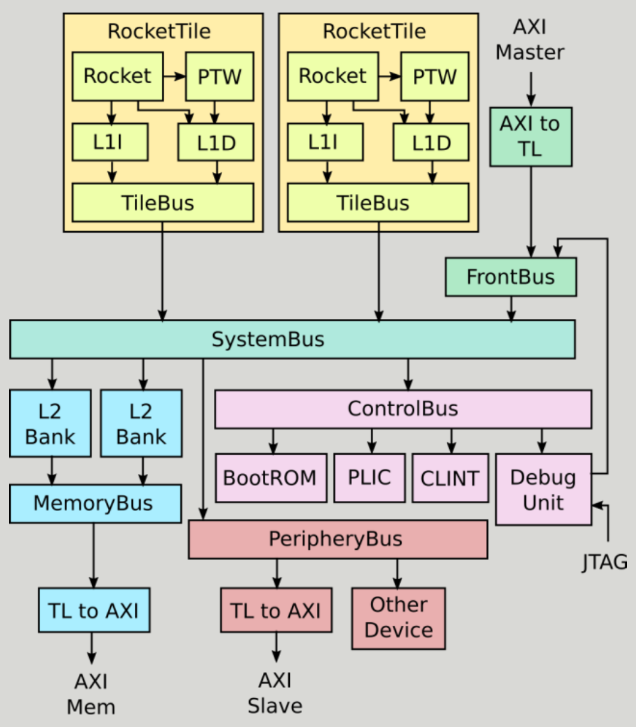
- **Open** ISA & tools
- **Customizable** processor architecture

GOAL

To demonstrate the feasibility of designing **safe and secure** architectures using **open-source** tools based on **RISC-V**

- From architecture design (Chisel)
- To verification on an FPGA

This work is based on **Rocket-chip** and **Chipyard.**



APPROACH

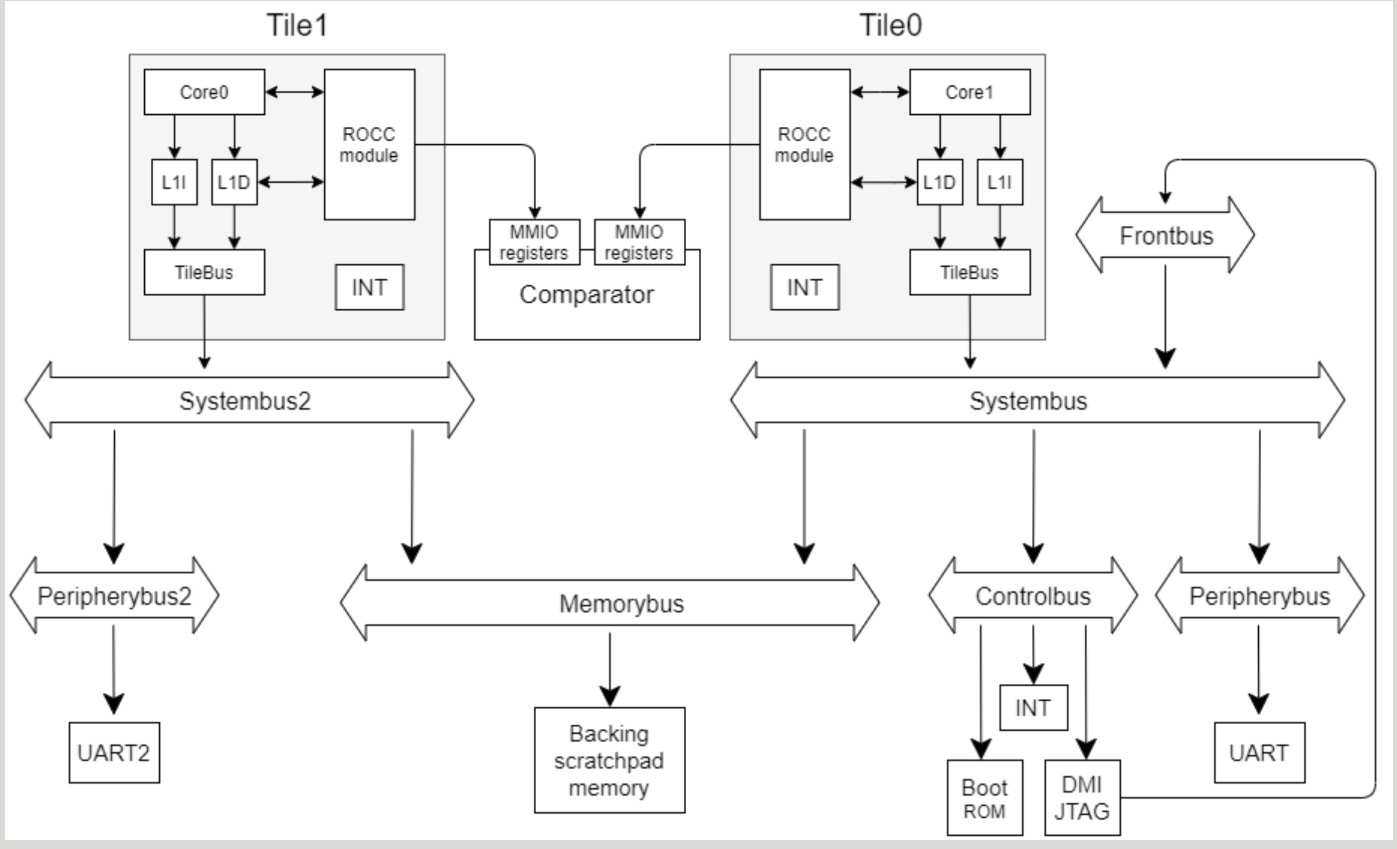
Analyze **S&S** constraints from **three different perspectives:**

- **Architecture design** → HW isolation, checksum
- **External IPs** → DMR/TMR
- **Fault tolerance** → ECC on internal memory

Methodology

ARCHITECTURE DESIGN

- **Rocket-chip:** flexible, but not quite **safety**-oriented.
- We built an SoC architecture based on Rocket-chip but fully oriented to **S&S aerospace applications.**
- Use case: redundant module in **lockstep** on two subsystems + comparator (secure checksum).
- **Isolate** subsystems → **physical separation.**
    - Avoids **contention** → ensures **time determinism** → **certification.**
    - Provides a layer of **HW security:** attacks on one subsystem won't affect the other.
- If one subsystem malfunctions, the whole SoC can be reset.



SAFETY INTEGRATION OF EXTERNAL IPs

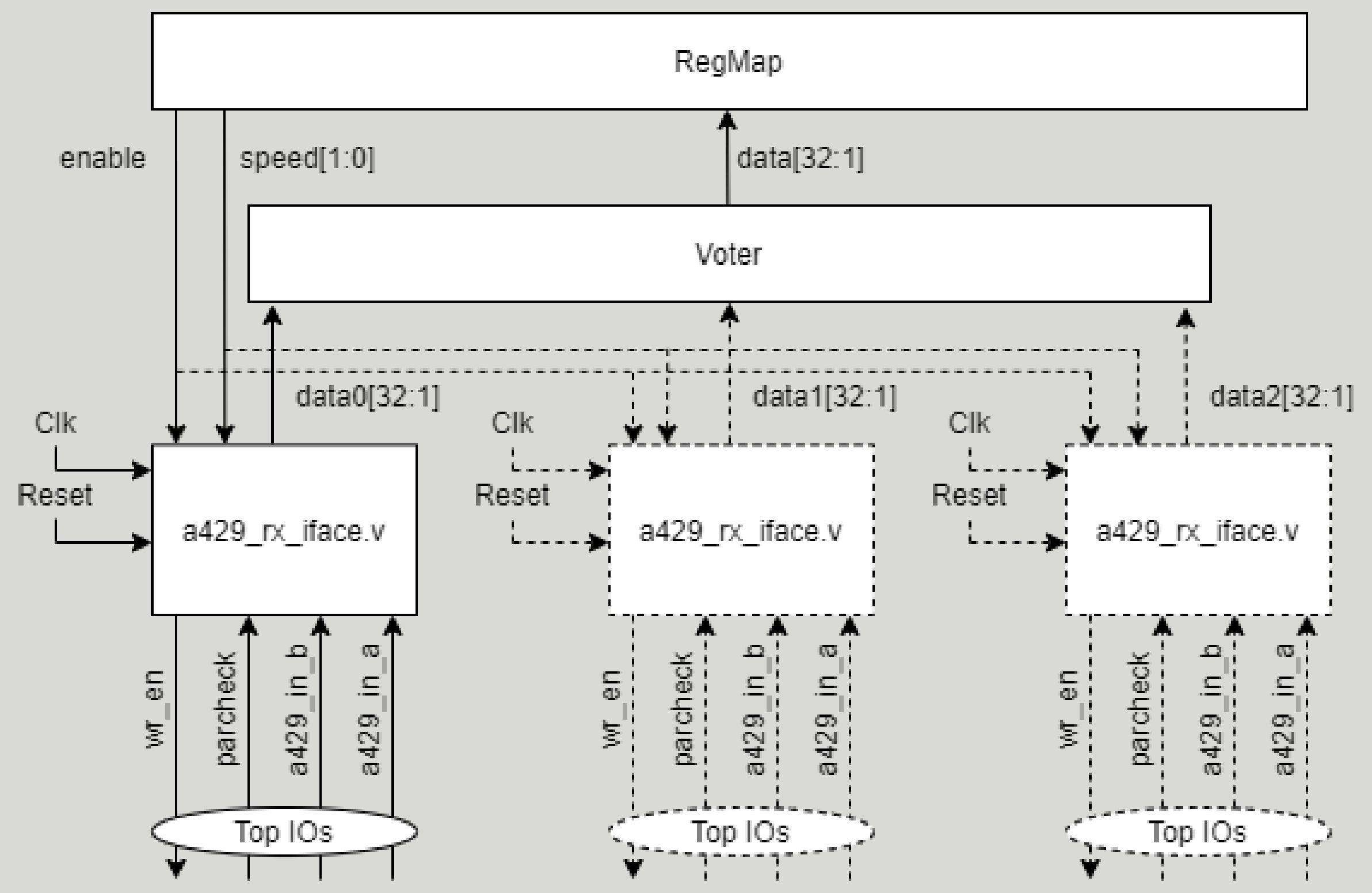
Legacy IPs:

- Preexisting blocks, already developed and tested.
- Integrating into custom SoCs would save time and costs.
- Often defined in **HDL** (Verilog/VHDL); sometimes **IP-XACT.**
- **IP-XACT** is a very extended standard for IP description.
- **Chisel** supports **HDL** "black boxes", but **not IP-XACT.**

We have developed a **Chisel class** that:

- **Parses IP-XACT** XML → Rocket-chip **diplomatic nodes**
- Optionally adds **DMR** or **TMR** to the black box

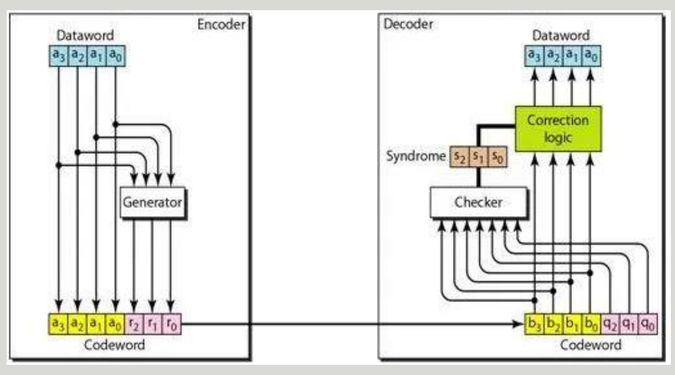
Validation use case: an **ARINC-429 interface** with **TMR.**



FAULT TOLERANCE AND ERROR DETECTION

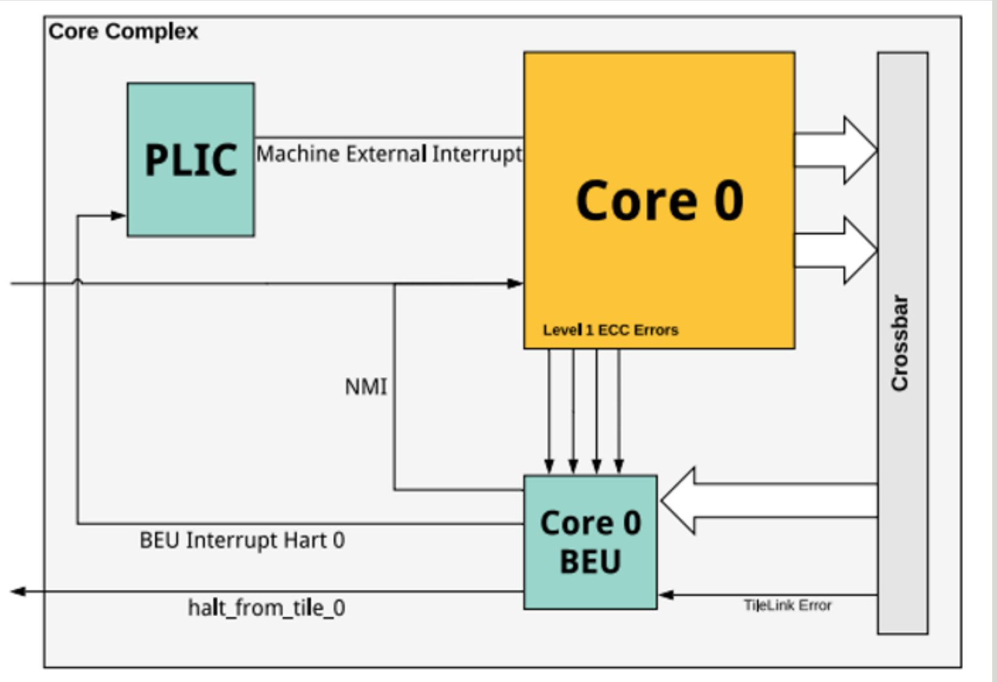
Single event upset → randomly flip a bit in memory

- On a flip-flop → can be mitigated with TMR
- On a **block RAM** → use **error correction codes**
    - Can **detect** and **correct** single-bit errors in memory



Chipyard includes ECC support:

- Add **ECC** to **cache/scratchpad** BRAM
- **Bus Error Unit** (BEU)
    - Trigger an **interrupt** on error detection/correction
    - If many errors are detected, the system can **take action**



Assessment: *How do we verify that ECC works?*

- We want to **test** ECC feature → artificially **inject faults**
- Chipyard does not provide fault injection mechanisms
- Alternative: use dual-port BRAMs → too complicated
- **Solution:** use **ICAP** → rewrite FPGA **config memory**
    - Read config memory corresponding to BRAM
    - Flip one or several bits
    - Write result back to config memory
- Done by a system independent from RISC-V processor

After applying this, we verified that **ECC works as expected.**

Collins Aerospace | APPLIED RESEARCH & TECHNOLOGY

© 2023 Collins Aerospace. | Collins Aerospace Proprietary. | IE Export Classification: NSR - CLS25370182.