

Space Applications: Leveraging FPGAs and RISC-V

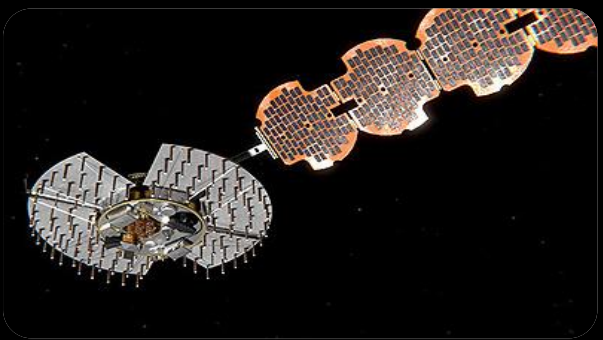
**Göran Bilski, AMD Fellow
Gothenburg, Sweden**

Agenda

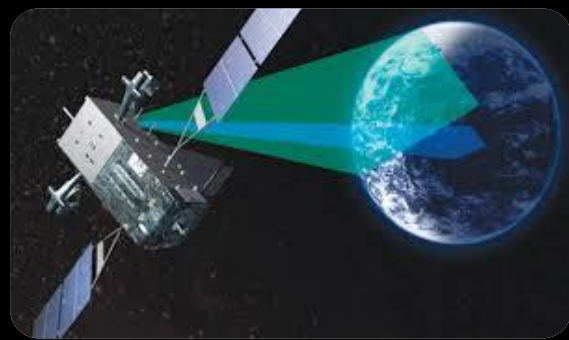
- This presentation aims to answer the following questions
 - Why use FPGAs for space applications?
 - Why use soft CPU cores?
 - Why use RISC-V?

Why use FPGA?

Missions Baselined with Kintex™ UltraScale™



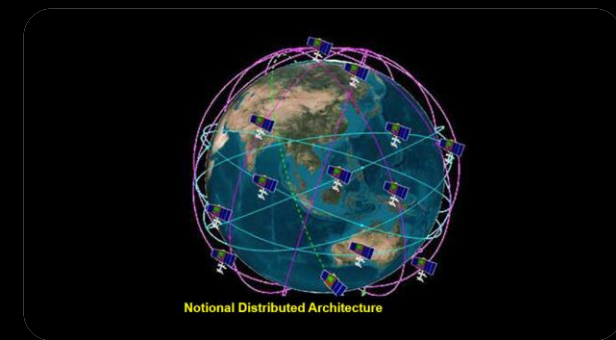
Navigation Program



Missile Defense System



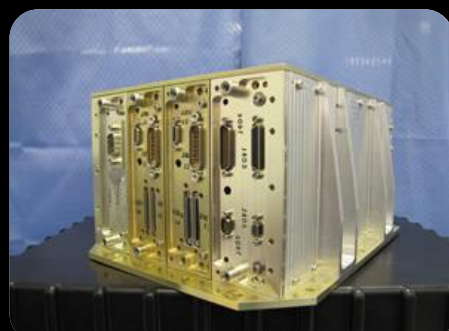
RF Processor



Electro-Optical / Infrared
Weather System



SDA Tranche 0



NASA SpaceCube



GEO Internet



Lunar Program

Exciting Recent Activities!

Strong Adoption & Heritage

JUICE
European
Space Agency

June 2023



Virtex4 Series
Science Instrument
Chemical Composition of
Jupiter Icy Moons

Europa Clipper
NASA/JPL

Oct 2024



Virtex5 SIRF
Spacecraft Avionics and
Science Instruments

Chandrayaan-3
ISRO Agency

July 2023



**Virtex5 SIRF &
Virtex4 Series**
Science
Instruments

SDA Tranche 0

Feb 2024



Kintex Ultrascale
Imaging

LEO Constellations

1H2023
Onwards



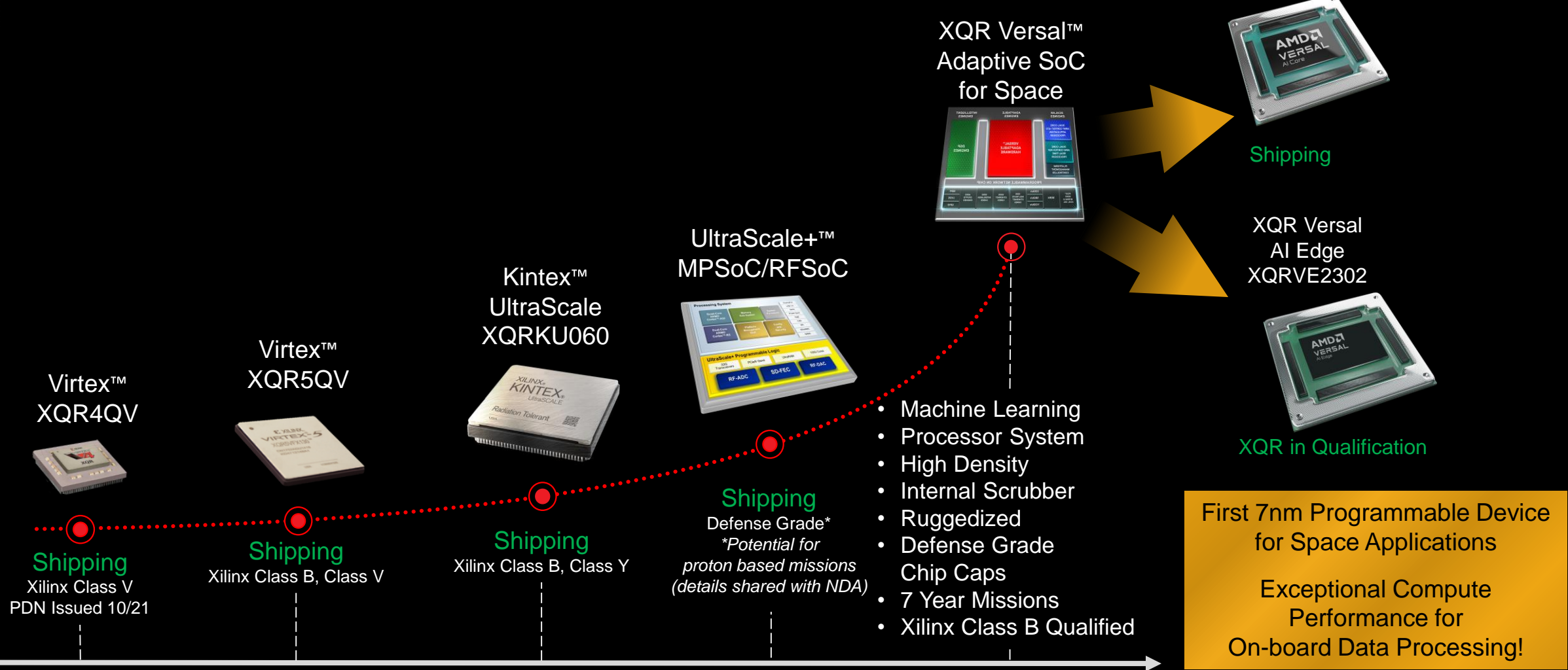
Versal
Four Constellations:
One USG LEO
Three Commercial LEO
(Two USA, one Asia)



Why use FPGA in Space?

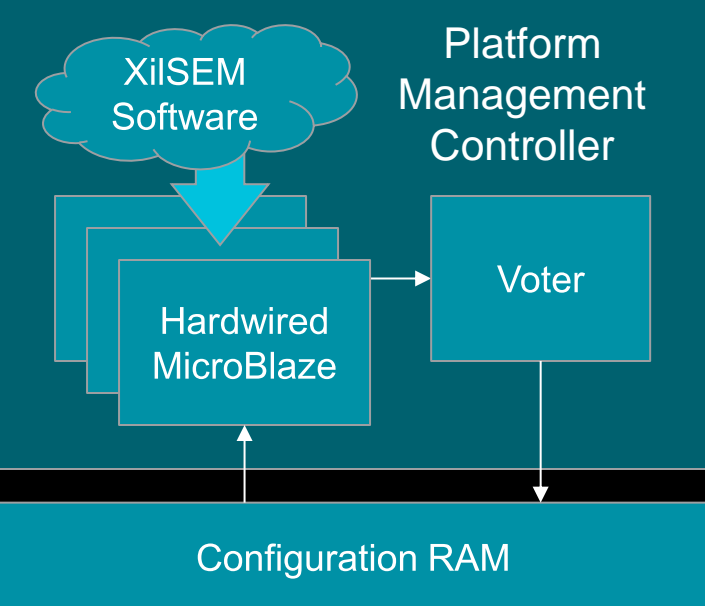
- Off-the-shelf vs ASIC vs FPGA
- Each has its benefits and drawbacks
- FPGA benefits
 - Cheaper than ASIC is unless volumes extremely high
 - Reconfigurable after deployment (new features or bug fixes)
 - Available in Space Quality

AMD Space Grade Products



CRAM Soft Error Mitigation in XQR Versal™ Adaptive SoC

- XQR Versal uses a novel approach to mitigate SEUs in configuration RAM (CRAM)
 - Previous generations of Xilinx FPGAs use SEM (Soft Error Manager) IP residing in the programmable logic fabric
 - XiISEM uses the hardwired TMR MicroBlaze™ processors in the Platform Management Controller (PMC) as a fault-tolerant platform to mitigate upsets in the configuration RAM
 - Approx 30 times greater protection than previous techniques



	Kintex™ UltraScale™ XQRKU060	Versal XQRVC1902	Comments
Configuration Memory (Mb)	193 Mb	363 Mb	VC1902 has ~ 80% more CRAM than KU060
SEFI Rate per Device* (with mitigation, GEO solar min)	1 in 6 years Using SEM	1 in 200 years Using XiISEM	30X Improvement

* AMD internal radiation test data

Why use soft CPU?

Why Soft CPU?

- All larger devices today have small control processors distributed on the silicon
 - CPUs, FPGAs, GPUs, Disc Drives, ...
 - Versal FPGAs have more hard MicroBlaze cores than ARM application processors
 - Configuration, Power management, High-Speed Serial Interfaces, Memory controllers
- Large systems implemented in FPGA logic, have the same need for control processor
 - The soft CPU used needs to be very resource efficient to minimize the area cost
 - Extreme configurability for each control processor requirement is beneficial
 - MicroBlaze has more than 70 parameters

Hardware control processor

MicroBlaze is used in the Mars 2020 Perseverance rover's instrument flight software (iFSW). The iFSW runs on the MicroBlaze and coordinates the functions of the rover's DEA hardware.

How does the iFSW work?

Receives commands from Earth

Executes commands

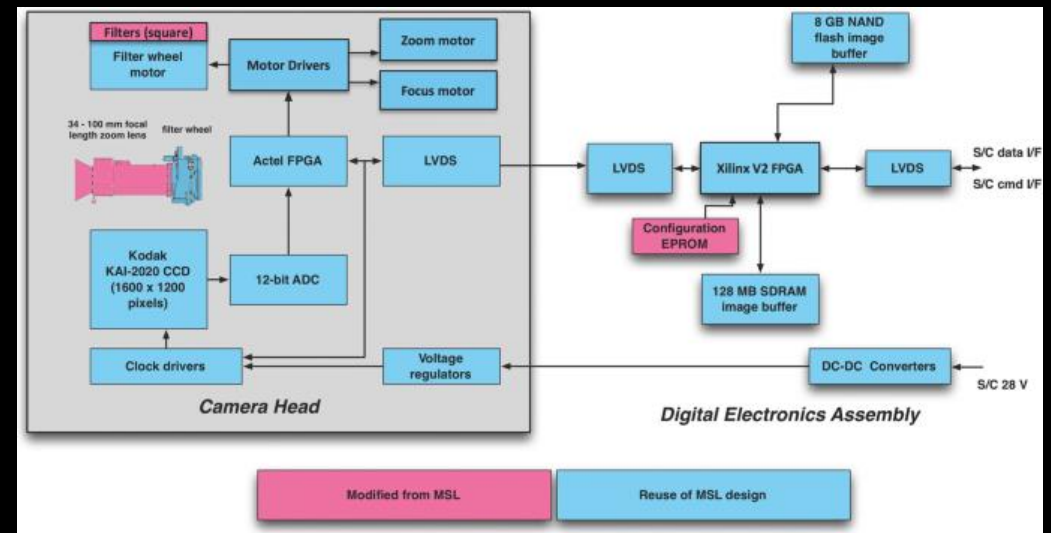
Transmits data back to Earth

Performs autofocus, "Z-stack", and auto-exposure

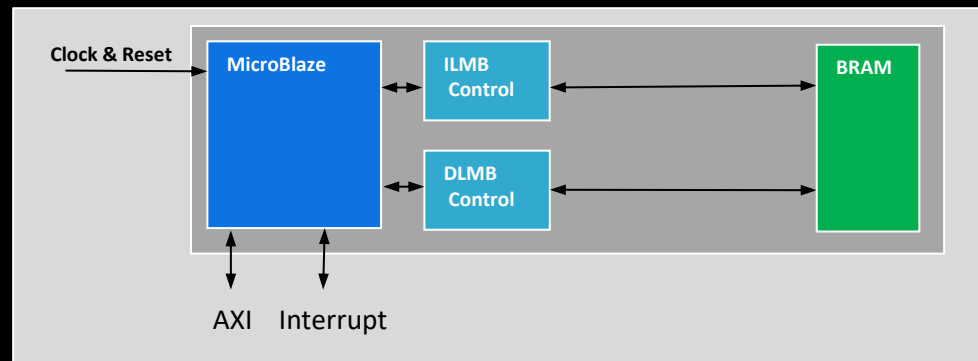
Corrects errors in flash memory

Controls mechanisms and protects against faults

The iFSW consists of about 10,000 lines of ANSI C code.

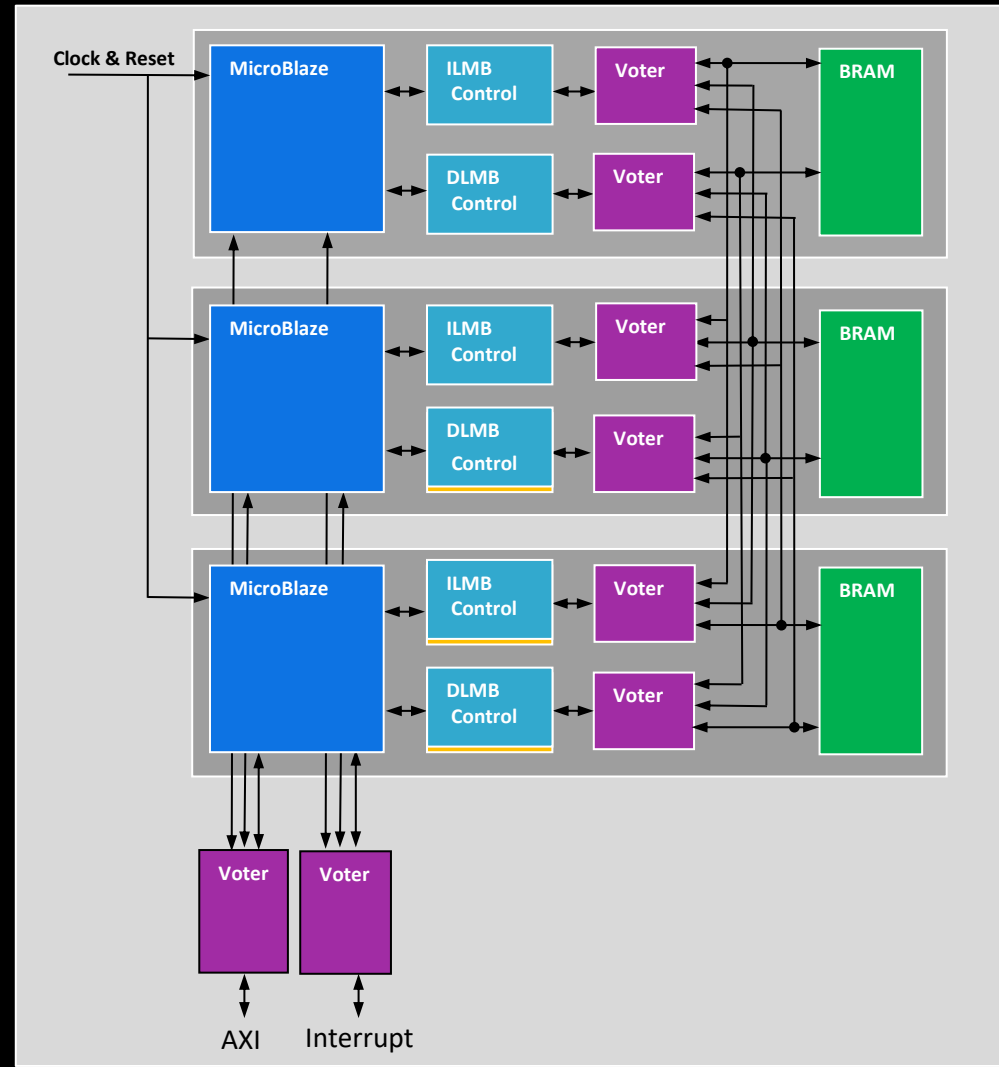


MicroBlaze TMR Voting to be Fail Operational (FO)



MicroBlaze TMR Voting to be Fail Operational (FO)

- Triplicate MicroBlaze, LMB Controllers and BRAM
- Voters on
 - AXI DP – AXI Interconnect
 - Interrupt – Interrupt Controller
 - Data from BRAM
- Voters mask output from failing MicroBlaze



MicroBlaze TMR Comparators to be Fail Operational (FO)

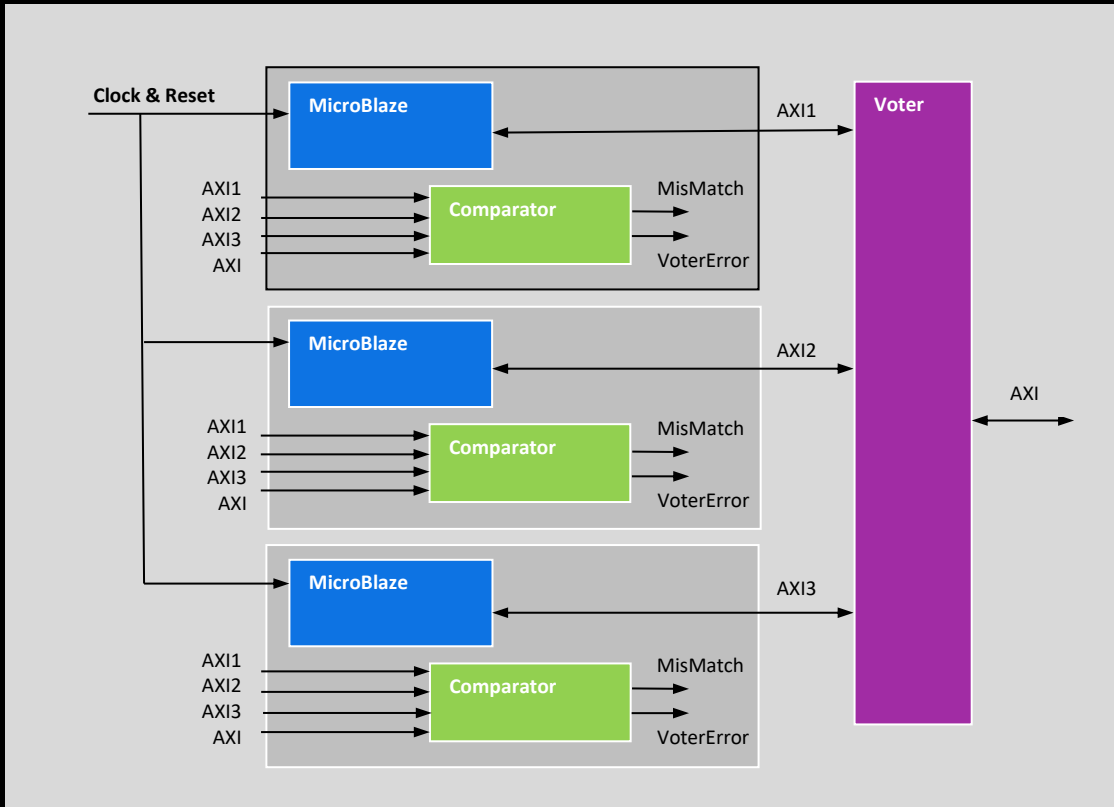
➤ Comparators have multiple functionality

- Decide which CPU is faulty going from FO to FS
- Detect compare errors in FS-mode and halt execution
- Voter self-check

➤ Comparators triplicated

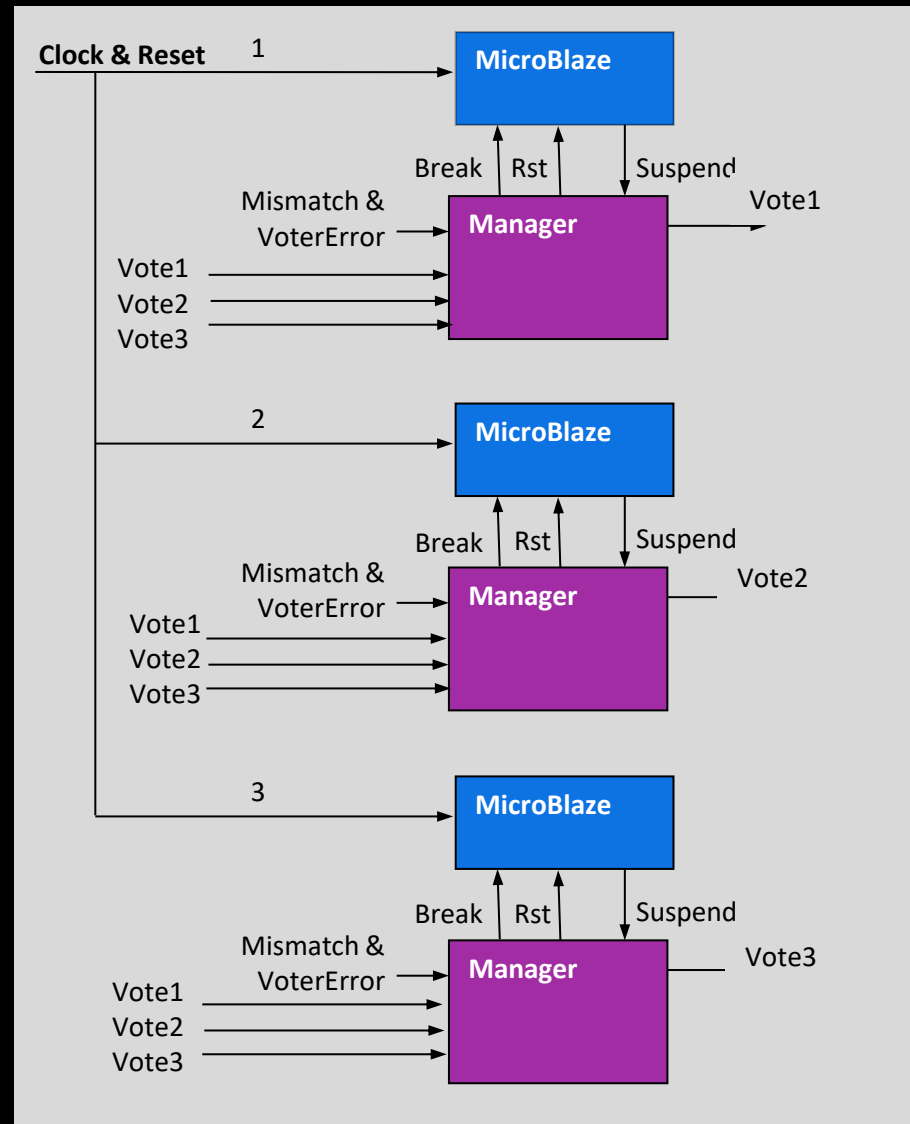
- Remove single point failure
- Placed together with triplicated MicroBlaze instances

➤ Figure only shows AXI



MicroBlaze TMR Management and Control

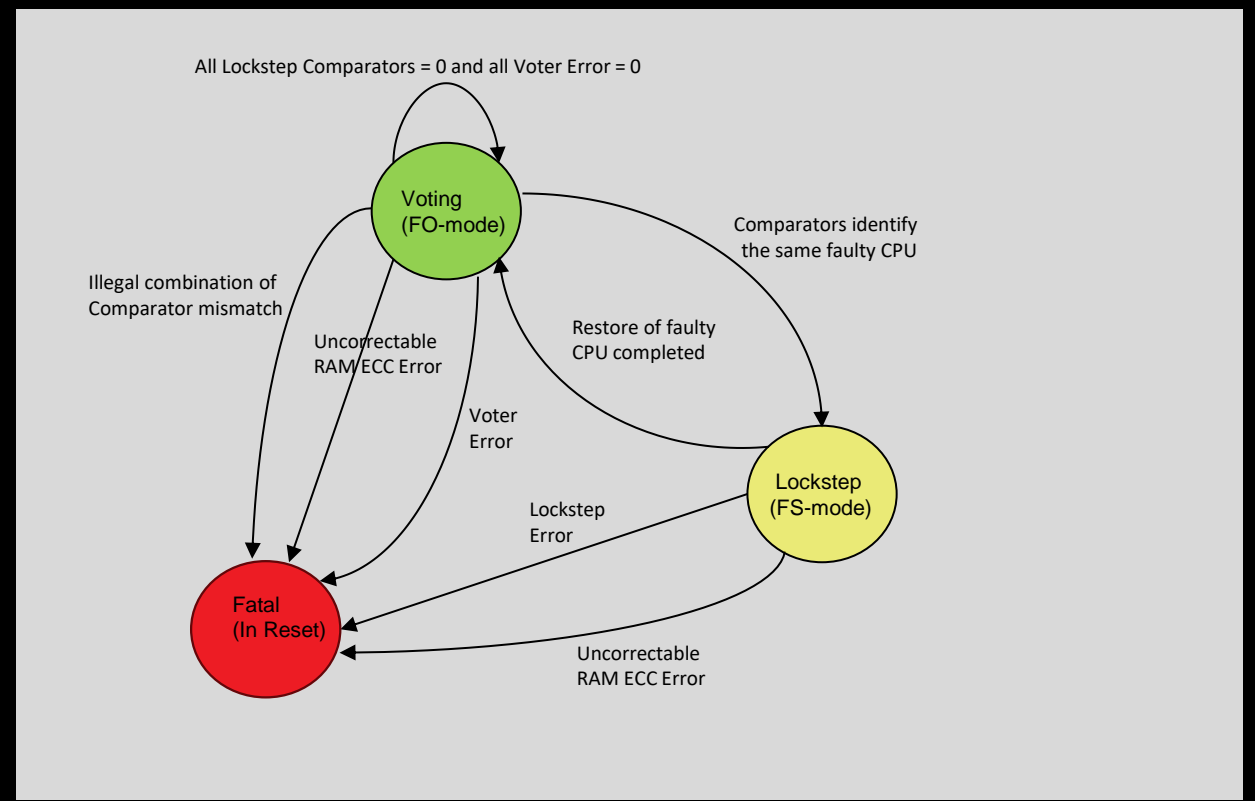
- Placed together with triplicated MicroBlaze sub-system
- TMR on all DFFs
- LMB to MicroBlaze for register access



MicroBlaze TMR Management and Control

- Voters works without any control or maintenance
- Fail Safe comparison needs to keep track of faulty MicroBlaze
- Recovery of faulty MicroBlaze
- Control of fatal condition
- Fault tolerance states

- Voting (FO-Mode)
 - All three MicroBlaze sub-systems are healthy
- Lockstep (FS-Mode)
 - Two MicroBlaze sub-systems are healthy
- Fatal (Reset)
 - System has detected unrecoverable error and is put in reset



Why use RISC-V?

My ISA Journey

- 6809E (Dragon 32k)
- 6502A (BBC Micro)
- ARM2 (Acorn Archimedes A310)
- Transputer (Inmos T400)
- Mil-Std-1750A (MDC281, MAS281, MA31750)
- Motorola 68k (Ariane5)
- Thor (Thor1, Thor2)
- Sparc (ERC32)
- PowerPC (Xilinx Virtex II Pro)
- MicroBlaze (my first real own ISA, architecture and design)
- AI-Engine (Initial architectures and design)
- RISC-V (MicroBlaze V)

My ISA conclusion

- The actual ISA is less important today
 - Implementation criteria are more important (same ISA -> very different products)
 - An ISA that allows very different products is needed
 - Development/debug tools, library supports is a must, the more the merrier
 - A living community and research, otherwise no new features and products
 - An ISA that doesn't evolve is stagnant (like human languages)
 - Latin might be a cool language, but it is not easy to speak to many today or find newly released books
- RISC-V fulfils all these criteria making it a successful ISA

RISC-V Soft-CPU Special Interest Group (SIG)

- **Goals and Scope**

- 1. Provide a forum for FPGA vendors, developers, and users to pursue common interests, benchmarks, and develop community-based recommendations.
- 2. Represent FPGA implementation considerations within RISC-V TGs, acting as a resource for consultations and to monitor progress of RISC-V standards from the perspective of Soft CPUs.
- 3. Develop strategy, perform gap analysis, and propose RISC-V extensions, RISC-V platforms, RISC-V profiles, and other technical product for FPGA implementations and applications.
- 4. Promote FPGA Soft-CPU IP development and inclusion in RISC-V International activities, publications and directories.

- **Current work**

- **CX (Composable Custom Extension) Task group**
 - Effort to standard custom instructions (hardware and software) and create a plug&play infrastructure.
- **CIO extension (Cache Index Operations)**
 - Cache operation without use of address and use cache index instead

BACKUP

MicroBlaze TMR Recovery Flow

- From SW application view seen as serving an interrupt
- Save, reset and restore of MicroBlaze takes ~100 clock cycles
- Flow:

1. TMR Management and Control

- I. Identify faulty CPU, latch failure information and go to lockstep state
- II. HW break signaled to MicroBlaze

2. MicroBlaze service HW break

- I. Save Register File and Special Registers to BRAM
- II. Issue SUSPEND instruction, which empties pipeline and settles all external bus transactions

3. TMR Management and Control

- I. Detect MicroBlaze HW Suspend signal asserted
- II. Assert reset of MicroBlaze system

4. MicroBlaze comes out of reset

- I. Read failure status from TMR Management and Control
- II. Restore MicroBlaze state by reading saved Register File and Special Registers from BRAM
- III. Execute Return from Break instruction (RTBD)

5. MicroBlaze is now back executing user application with faulty MicroBlaze sub-system brought back executing inline with the two healthy sub-systems

AMD 