Implementing Functional Safety in a RISC-V Interleaved-Multi-Threading Processor Core

Marcello Barbirotta, Abdallah Cheikh, Antonio Mastrandrea, Francesco Menichelli, Mauro Olivieri

Sapienza University of Rome, Italy

marcello.barbirotta@uniroma1.it, mauro.olivieri@uniroma1.it

Spring 2022 RISC-V Week Paris

RISC-V

Tuesday 3rd to Thursday 5th May 2022

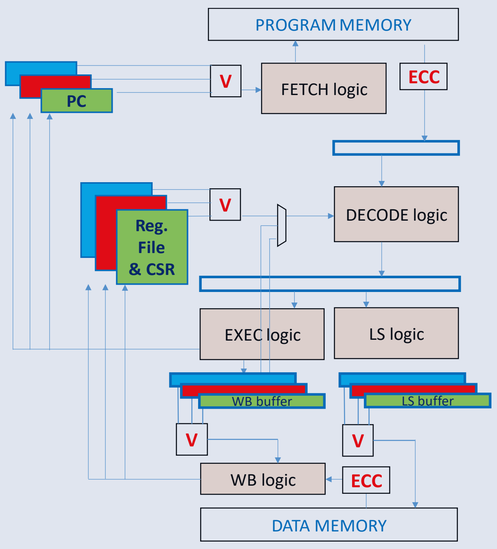
Challenge

Safety critical errors in digital microprocessors increase with the reduction of minimum feature size and voltage margins. For that reason, Functional Safety in microprocessor cores through **Fault Tolerance (FT)** architecture techniques, is a key requirement in several embedded application domains, especially when based **on commercial-off-the-shelf (COTS)** hardware components. Yet, classical **Triple Modular Redundancy (TMR)** protection against faults occurring in any bit of the core may be oversized for many applications, especially for low-cost implementations.

While being a revolution in embedded system processor cores**, RISC-V** is now receiving increased interest for space and safety applications. In this work we illustrate a novel approach for developing robust Fault Tolerant RISC-V processor cores, using an **Interleaved-Multi-Threading (IMT)** architecture as a basis for Radiation Hardening technique. IMT cores are interesting for low-cost embedded systems, because they reach relatively high performance by hiding data dependency stalls and providing a fence between the read access and the write access to the register file. Contrary to FT approaches based on Simultaneous Multi-Threading (SMT) or Multi-Cores (MC) schemes [4], the proposed approach exploits the IMT scheme to merge spatial redundancy with temporal redundancy, leveraging specific logic structures to deal with the fact that the instruction results are not simultaneously available.

Features

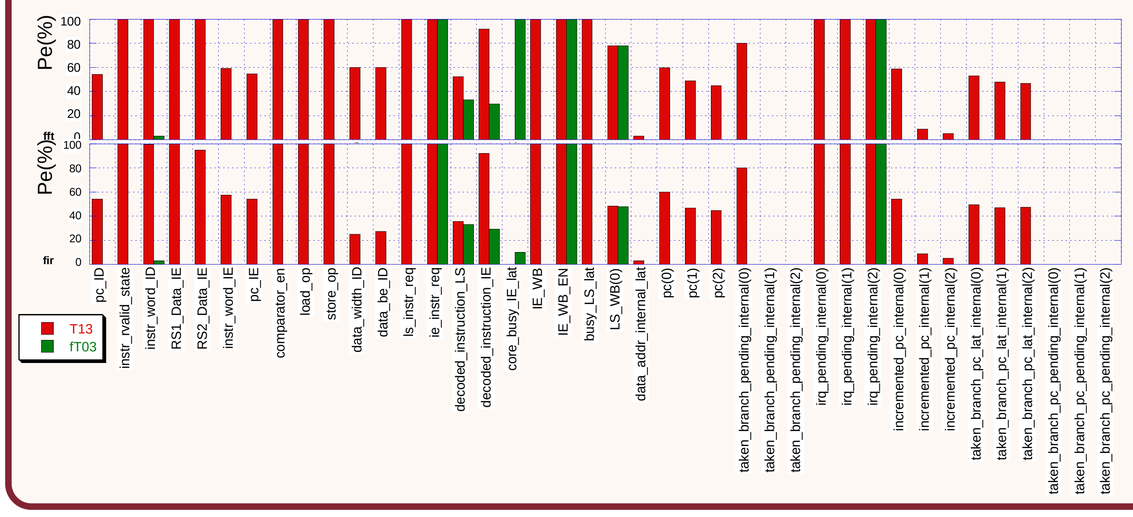
The original IMT architecture natively offers spatial redundancy, by means of replicated register file, PC and Control/Status Registers (CSRs) to maintain the states of the three threads being executed. Voting among the logic signals produced by three identical threads could be introduced in several points of the pipeline microarchitecture. We call the proposed paradigm ***Buffered TMR***, defining precise architecture modifications with general validity. The values produced by three harts in selected architectural units (PC, Register File, Write Back unit and Load Store Unit) are *buffered* in dedicated registers and voted at the end of each thread instruction cycle, by performing an intrinsic TMR protection and correct data retention.



Fault Injection Results

The obtained RISC-V processor core, named Klessydra-fT03 [1], possesses FT features briefly quoted in *Table below* and validated by an extensive Fault Injection simulation campaign with Single-Event-Upset (SEU) faults targeting all the most used register bits in the architecture (see *figure below*).

benchmarks	Klessydra-T03 (Pe%)	Klessydra-fT03 (Pe%)
FFT	62 %	13 %
FIR	60 %	10 %



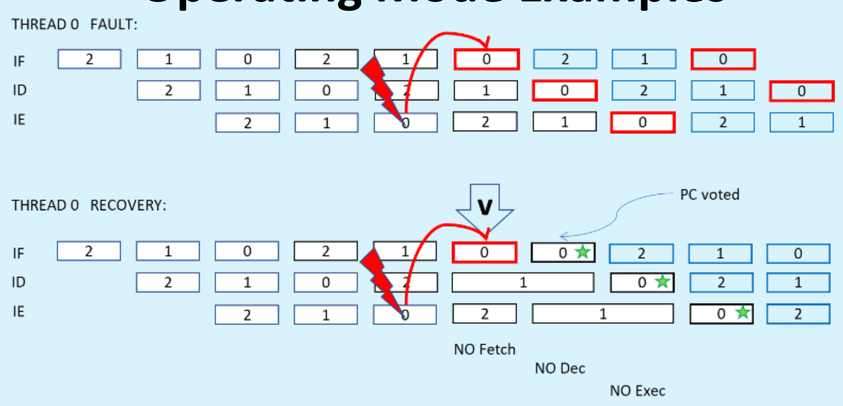
Operating Principle

The basic concept of our work is the intrinsic FT capability of an IMT core running three threads, each having its own Register File (RF), Program Counter (PC) and Control/Status Registers (CSRs) incorporating ***spatial redundancy;*** yet sharing the pipeline logic and registers to execute the same instructions in different clock cycles, thus providing ***temporal redundancy***.

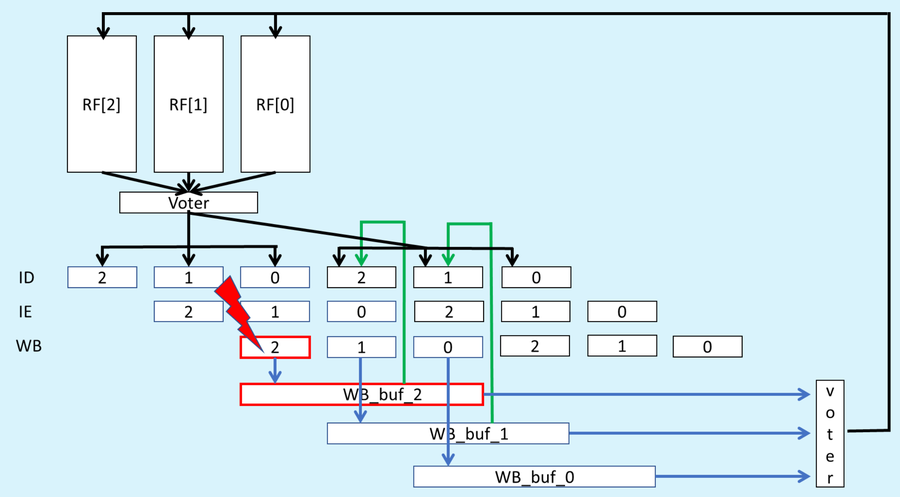
We started from an open-source RISC-V soft core family named **Klessydra-T**[2], developed at the Digital System's Lab group at Sapienza University of Rome, with three or more hardware interleaved threads in a round robin fashion on a four-stage in-order pipeline, fully compatible with the **PULPino** open-source microcontroller platform [3].

Executing the instructions of identical threads on the same hardware in different clock cycles, protects the architecture from **Single Event Upset (SEU)** in sequential logic and from **Single Event Transient (SET)** faults that may occur in combinational logic.

Operating Mode Examples



- **The case of PC fault:** In case of a fault, fore example on thread zero, it can potentially jump to another wrong instruction (red color in picture), changing the instruction flow among the subsequent cycles. The recovery mechanism implies that during thread zero fetch state, a voting among the PC signals is done, and the correct result is re-fetched in the next cycle, restoring the correct instruction flow with only one cycle penalty.



- **The case of RF&WB fault:** The three RF works in parallel as single TMR one, and each thread reads from it with voting mechanism. The results from the threads are saved in the Write Back buffers and taken from there in case of dependencies using Bypass logic. In case of a fault, for example on thread two, the voting mechanism eliminates it completely
- **The case of LS fault:** Since the threads are exactly replicated, the Load/Store unit should avoids performing replicated load-store pairs to the same location or propagating any fault to the memory sub-system. Thread 2 and Thread 1 memory operations are converted into two *buffered* operations, by saving all the related signals like address, data and control signals into dedicated registers, giving to Thread 0 the task of doing the real load-store operation with voted signals

Conclusions

In this work, we presented a way to use IMT processor core to achieve Fault Tolerant features. By following some specific steps precautions it is possible to add features that are fully comparable with other fault protection mechanisms. The core behaviour appears like a single thread core, but it has performance of TMR core in his sphere of replication (SOR) with less overhead because not all the signals in the pipeline needs to be protected with the buffered TMR solution.

References

[1] M. Barbirotta, A. Cheikh, A. Mastrandrea, F. Menichelli, F. Vigli and M. Olivieri, "A Fault Tolerant soft-core obtained from an Interleaved-Multi- Threading RISC- V microprocessor design," 2021 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2021, pp. 1-4.

[2] Cheikh, A., Cerutti, G., Mastrandrea, A., Menichelli, F., Olivieri, M., The microarchitecture of a multi-threaded RISC-V compliant processing core family for IoT end-nodes. In: Applepies 2017. Lecture Notes in Electrical engineering, vol 512. 2019. Springer, Berlin.

[3] D. Rossi, F. Conti, A. Marongiu, A. Pullini, I. Loi, M. Gautschi, G. Tagliavini, A. Capotondi, P. Flatresse, and L. Benini, "PULP: A parallel ultra low power platform for next generation IoT applications," 2015 IEEE Hot Chips 27 Symposium, HCS 2015, 2016.

[4] Oz. Isil et al., "A survey on multithreading alternatives for soft error fault tolerance", ACM Computing Surveys (CSUR) 52.2, (2019), pp. 1-38.