Adding Dynamic Triple Modular Redundancy on a RISC-V Microarchitecture

Marcello Barbirotta, Abdallah Cheikh, Antonio Mastrandrea, Francesco Menichelli, Marco Ottavi and Mauro Olivieri

Sapienza University of Rome, Tor Vergata University of Rome, Italy

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

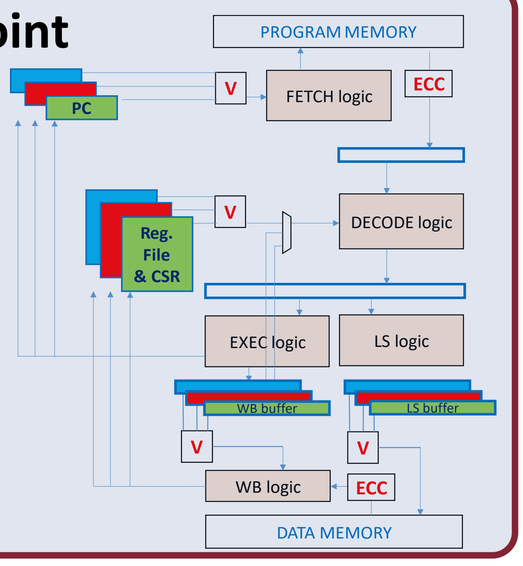
Challenge

Fault Tolerance represents an important area for digital applications, as how to protect electronic circuits from the different types of **SEE (Single Event Error)** faults, is a very complex process [1]. Among the various FT techniques developed, we find software methodologies, **Double Modular Redundancy (DMR)** and **Triple Modular Redundancy (TMR)** hardware methodologies, temporal redundancy techniques, and hybrid techniques based on previous ones. All techniques in the literature protect circuits by acting at different levels [2], making them more expensive and less performing than the respective non-redundant architectures.

Safety and **Reliability** are, therefore, two key concepts in the technological world, and **RISC-V** plays an interesting role in this context for its inherent extensibility and the availability of open-source microarchitecture designs.

Starting Point

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 are *buffered* in dedicated registers and voted at the end of each thread instruction cycle.



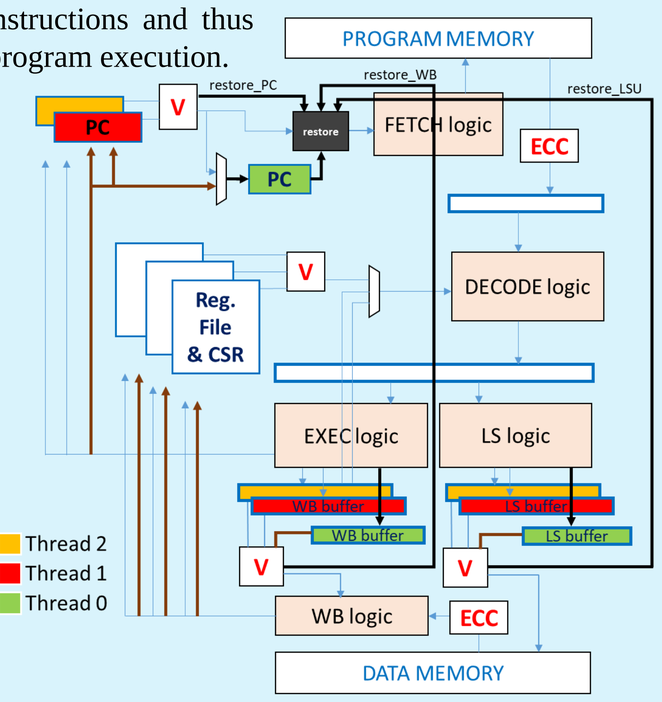
Operating Principle

This work summarizes the idea presented in [3], centered on the application of the DMR paradigm within an **Interleaved Multi-Threading (IMT) RISC-V** architecture, gaining the low overhead advantages of the DMR technique, and yet overcoming the cost of saving checkpoints and restoring the software state using **Dynamic TMR (DTMR) protection**, which actually implies the behavior of a **TMR only in the case of error detection**.

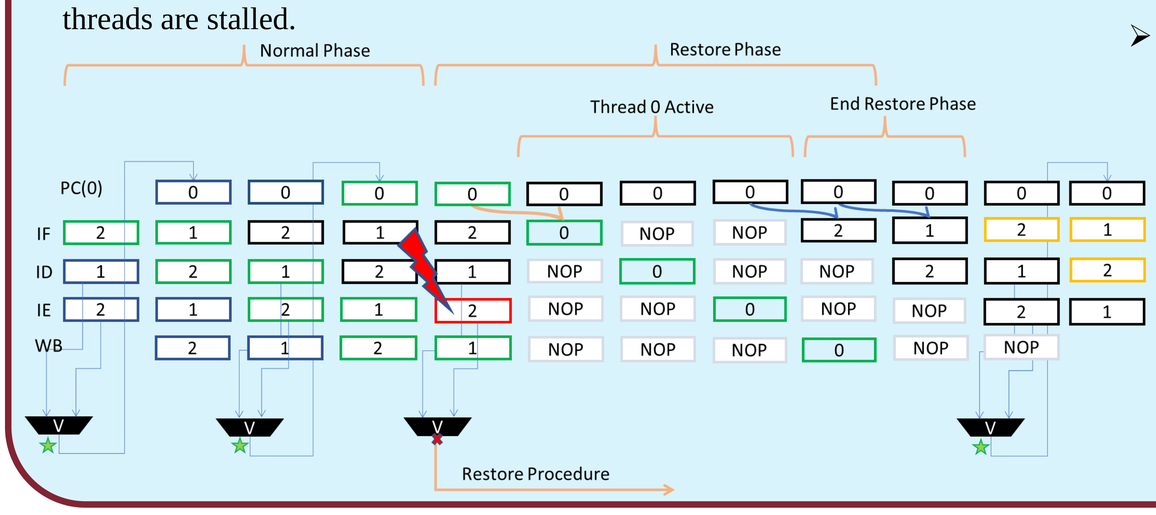
Operating Mode Examples

The principle of the **Dynamic TMR** is the implementation of a Double Modular Redundancy instead of a triple one **without adding overhead for recovery and checkpointing mechanisms** typically visible in these environments, turning the *Buffered TMR* into a DMR technique to reduce power consumption and increase speed.
From that idea, we built a new core named **Klessydra-dfT03**, where "d" stands for dynamic, and we introduced a single register that saves the address of the last correct instruction and restores it in the PC with a latency overhead of four clock cycles, without any changes in the data memory or the Register File [3].We use three hardware threads, numbered Thread 2, Thread 1, and Thread 0 (blue, red, and green colors in Pipeline), and we leave only threads 2 and 1 active while turning off Thread 0 (called the auxiliary thread) activated only in case of fault detection, having no dynamic consumption for its dedicated hardware units. The operation are organized into three modes [3]:

- **Normal or "Buffered DMR" mode:** Threads 2 and 1 work in interleaved mode (blue arrows in Pipeline), executing the same instructions and thus implementing spatial and temporal redundancy, with a *buffered* voting mechanism in the critical units to check for the correctness of the program execution.

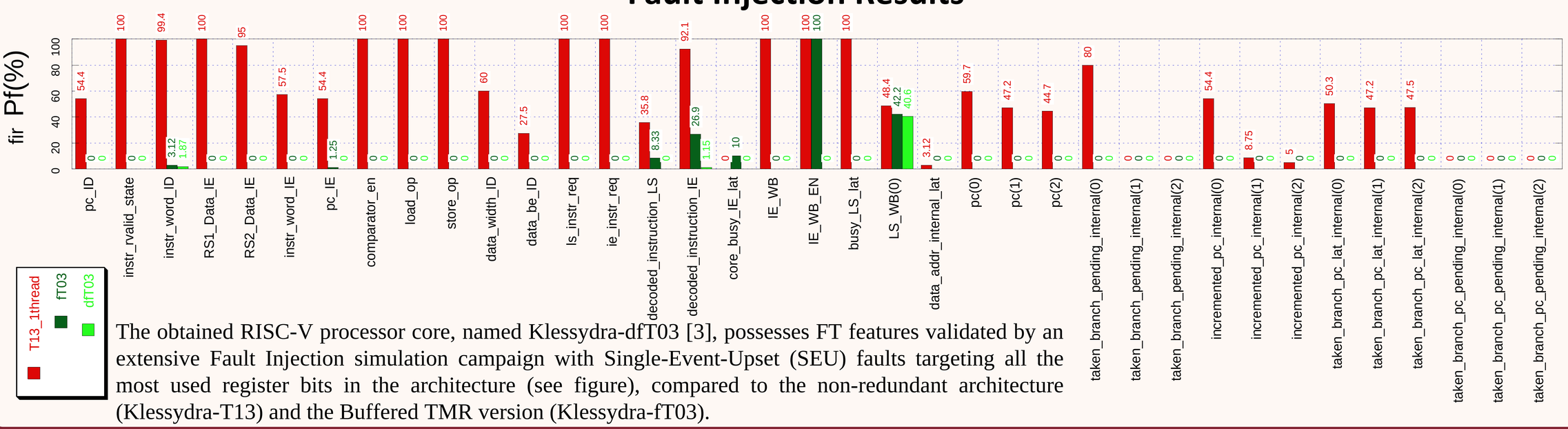


- **Restore or Recovery mode:** If the voting logic gives a negative result due to a fault, specific control signals named *restore* (Pipeline) are asserted, and the core enters the recovery mode. Notably, a fault is always detected before the Register File wrong WB. Following the black arrows in Figure, the *restore* activate the Restore block (black unit in Pipeline), which wakes up the auxiliary sleeping thread. As it enters the IMT pipeline, it fetches the last successfully executed instruction indicated by the dummy PC register, while the other threads are stalled.



- **End of Restore Phase mode:** Once the recovered instruction is completed, the produced result is compared with the results previously produced (brown arrows in Pipeline), thus obtaining a majority voting similar to a TMR system, writing back the correct value into the Register File. The recovery procedure ends with the suspension of the auxiliary Thread 0, and the loading of the address of the next instruction in the PCs of Threads 2 and 1, so that they restart from the instruction following the faulted one.

Fault Injection Results



The obtained RISC-V processor core, named Klessydra-dfT03 [3], possesses FT features 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), compared to the non-redundant architecture (Klessydra-T13) and the Buffered TMR version (Klessydra-fT03).

Conclusions

In this work, we discussed the creation of a new **hardening-by-design Dynamic TMR concept** technique starting from an open-source **IMT RISC-V microprocessor core**, exploiting the advantages of both DMR and TMR techniques obtaining a system that can dynamically switch from one to another in case of faults [3].

References

[1] Marcello Barbirotta et al. "Design and Evaluation of Buffered Triple Modular Redundancy in Interleaved-Multi-Threading Processors". In: IEEE Access 10 (2022), pp. 126074–126088. [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.
[2] Alessandro Bernardini, Wolfgang Ecker, and Ulf Schlicht-mann. "Where formal verification can help in functional safety analysis". In: 2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). ACM. 2016,pp. 1–8.
[3] Marcello Barbirotta et al. "Evaluation of Dynamic Triple Modular Redundancy in an Interleaved-Multi-Threading RISC-V Core". In: Journal of Low Power Electronics and Applications 13.1 (2023). issn: 2079-9268. doi: 10.3390/jlpea13010002. url: https://www.mdpi.com/2079-9268/13/1/2.

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

Find us on Github:
- https://github.com/klessydra