SEU and SET-tolerant ARM Cortex-R4 CPU for Space and Avionics Applications

Conference Paper · May 2013

CITATIONS
13

READS
2,009

3 authors, including:

Emre Özer
ARM
64 PUBLICATIONS **903** CITATIONS
SEE PROFILE

David M. Bull
ARM
32 PUBLICATIONS **1,143** CITATIONS
SEE PROFILE

SEU and SET-tolerant ARM Cortex-R4 CPU for Space and Avionics Applications

Massoud M. Ghahroodi[*], Emre Ozer[†] and David Bull[†]

[*]*Southampton University and* [†]*ARM*

[*]mmg08r@ecs.soton.ac.uk, [†]emre.ozer@arm.com and david.bull@arm.com

Abstract: *Soft errors induced by radiation, causing anomalies in satellite equipment and spacecrafts, have become one of the most challenging issues that impact the reliability of modern processors even in ground-level applications. In this work, we investigate the feasibility of using ARM Cortex-R4 CPU for space and avionics applications. We create a single-event upset and transient tolerant variant Cortex-R4 CPU without modifying the original CPU microarchitecture. We do this by applying the Triple Modular Redundancy (TMR) techniques to all sequential elements at gate-level. We present the implementation methodology, power, area and performance overheads of Cortex-R4-TMR normalized to a baseline Cortex-R4, without TMR.*

Keywords: Avionics, CPU, SET, SEU, Soft Errors, Space, TMR, Radiation Tolerance

I. Introduction

Soft errors induced by radiation have become one of the most challenging issues that impact the reliability of modern electronic systems. They affect the reliability of electronic components in high altitudes such as deep space, satellites and aircraft. NASA [1] report that 80% of radiation-induced anomalies in space are due to single-event effects such as SEUs and SETs. Single-Event-Upset (SEU) occurs when a particle strike causes a bit-flip (upset) in a memory cell or a latch. Single-Event-Transients (SET) occurs in combinational logic when a particle hits the combinational circuit causing a glitch that shows up as a narrow-width voltage pulse. With shrinking semiconductor technology, electronic systems at ground level are also affected by soft errors caused by ground-level radiation in automotive, railway, medical and industrial systems [2]. Addressing the soft error issue is critical for ground-level applications not only for reliability but also safety.

SEU and SET tolerance can be achieved at different levels of abstraction such as layout, transistor, gate, RTL, CPU microarchitecture and even software. In this preliminary study, we choose to implement gate-level solution to tolerate SEUs and SETs. Our approach avoids the need to modify the original CPU microarchitecture. We choose to do so because The **Cortex-R4** CPU microarchitecture is mature and has undergone extensive verification and validation processes as an industrial product. Any change in the CPU microarchitecture will have repercussions in verification, validation. For these reasons, we opted to triplicate all the sequential cells at gate-level in a product-quality CPU, also known as "Triple Modular Redundancy (TMR)". All the flip-flops and latches are replaced by their TMR versions in the gate-level net-list. A majority voter is used to compare the output of the triplicated cell. Only the sequential cells are triplicated, so no redundancy is required for the combinational logic in the design.

TMR is a well-established technique for radiation-hardened electronics, and it has advantage of providing forward progress in the case of errors but it comes with area, power and delay overhead. Clearly, there are other low-cost techniques used in SEU

tolerance such as DICE [3]. However, the rationale behind applying TMR to an ARM CPU is to establish an upper bound design and measure its power, area and performance overheads. Although TMR techniques are often used in space and avionics environments, they can also be used in other mission-critical applications such as automotive, railway, medical and industrial systems.

The rest of the paper is organised as follows: *Section 2* introduces the **Cortex-R4** CPU, its applications and the implementation of its TMR version. *Section 3* presents the experimental results showing the power, performance and area overheads. Finally, *Section 4* concludes the paper.

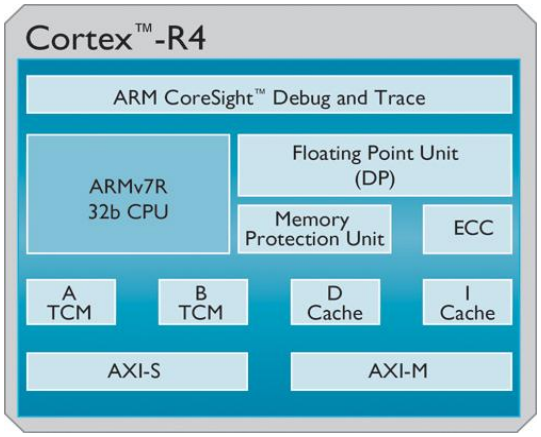

Fig 1. ARM Cortex-R4 CPU diagram

II. SEU and SET Tolerant Cortex-R4 Design

The ARM Cortex-R Series embedded processors are fast, real-time and cost effective. They offer high performance, highly deterministic behavior, and built in safety features. **Cortex-R4** [4] is an implementation of the ARMv7-R architecture specifically designed for deeply embedded real-time applications such as HDD/SSD storage controllers, communications modems, and electronic control units (ECUs) for automotive and industrial systems. It offers significant energy efficiency, real-time response and predictable performance for real-time systems. The block diagram of **Cortex-R4** is shown in **Fig 1**. It has an 8-stage in-order dual-issue CPU pipeline, optional Memory Protection Unit (MPU) and *CoreSight* debug and trace units. It has optional tightly-coupled memories (TCMs), instruction/data caches and double-precision floating-point unit. Also, two 64-bit master and slave AXI ports are used to communicate with the external world. All caches and TCMs can be protected by ECC or parity against SEUs. **Cortex-R4** can also be used in a dual-core lock step (DCLS) configuration where a second redundant CPU can run in lock step with the first one. Both cores will share inputs and caches, and the outputs of the CPU are compared at every cycle to detect errors. This feature is used mainly in automotive systems but may also be useful for some space and avionics applications. The key difference between these two methods is that the majority voting of the TMR system enables the system to continue after a SEU or SET, providing tolerance and maintaining availability. With DCLS an SEU or SET can be detected but the correct result is unknown, as there is no majority voting, so the system must take an appropriate corrective action.

All the sequential cells in **Cortex-R4** CPU are replaced by TMR cells that we create. **Fig 2** shows a TMR cell where each D-type flip-flop is triplicated and the majority voter selects the correct output in the case of an upset in one of the D-flops. To protect against SETs, three flip-flops sample the input signal at three different points in time (using delay

elements in the redundant flip-flops) so that any glitch in the combinational logic due to a SET will affect at most only one D flip-flop.

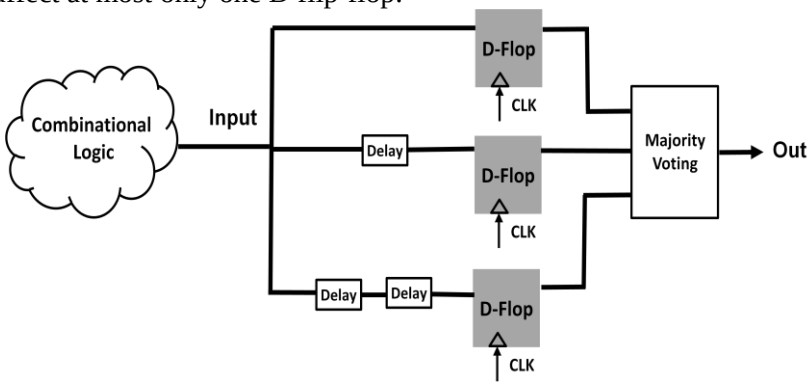

Fig.2 Triplicated flip-flops

We use a baseline **Cortex-R4** that does not have the optional floating-point unit and TCMs. However, it has both instruction and data caches of 16KB each. Both caches are protected by ECC. So we take the TMR cell and apply it to every single flip-flop in the design excluding the caches as they are protected by ECCs.

Ideally, we need to triplicate every single sequential cell in the cell library, characterize them and eventually add them to the cell library. During implementation phase, the tools must be limited to choose only the TMR version of each flip-flop cell from the cell library. In this preliminary study, we pursue a non-optimal alternative where a single flip-flop cell that has all the functionalities (Set, Reset, etc) is selected from the standard cell library, and then this cell is triplicated to replace all the cells in the design.

III. Experimental Results

We implement the TMR version of **Cortex-R4** in 65nm GP technology node. *Synopsys* IC compiler implementation flow is used to compile and P&R the entire design with the new TMR cell. Spice-level simulation is performed to validate the correct functionality and the radiation immunity of the TMR cell. SET pulses are applied to the input of the TMR cell with various widths to find the point-of-failure in SET immunity. Different SET immunity levels can be achieved depending on the amount of input delay in the delay cells. For the case of our TMR cell, the immunity is for SET pulses with the width of 105ps. Any SET pulse wider than this can be potentially captured (depending on its time of arrival and its relation to the clock rising-edge) and cause an error.

We have measured the target clock frequency, silicon area, and dynamic and power. The results are presented normalized to the baseline **Cortex-R4** CPU in **Fig 3**. The target clock frequency in the TMR version is 31% slower than **Cortex-R4**. This is mainly because the additional delays in the delay elements and majority voters in the TMR cells.

The chip area is doubled because all flip-flops are triplicated. The area of an individual TMR cell is about 6 times larger than a regular flip-flop. In the original **Cortex-R4**, sequential cells occupy 20% of the CPU area, and the area occupied by TMR cells becomes 60% of the overall **Cortex-R4 TMR** chip area. So, the area occupied by TMR cells dominates the area of combinational and macro cells consisting of 16 KB instruction and data caches each. The overhead of the TMR cells also depends on the configuration options of the **Cortex-R4** CPU. We have chosen 16KB instruction and data caches by default. If we had chosen 32KB instruction and data caches, the area overhead of the **Cortex-R4 TMR** would have been only 70% of the overall chip area rather than 100%.

We have also measured the total dynamic power in the post-layout of the design. Total dynamic power increases by 104% in comparison to the baseline **Cortex-R4** when **Cortex-R4** runs at the same clock frequency as **Cortex-R4 TMR**.

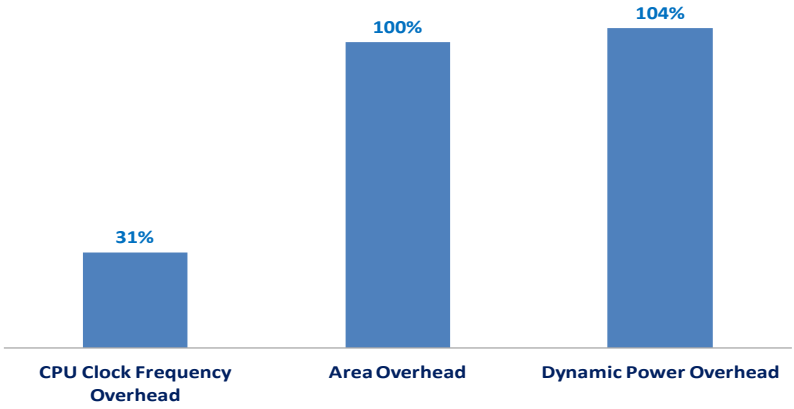

Fig 3. CPU clock, area and dynamic power overheads in **Cortex-R4-TMR** normalized to the baseline **Cortex-R4**

IV. Conclusion

This preliminary study investigates designing a SEU and SET tolerant ARM **Cortex-R4** CPU targeting space and avionics applications. We design it by triplicating all flip-flops at gate-level in **Cortex-R4** CPU. In this way, the microarchitecture of the **Cortex-R4** CPU is not modified as this has the advantage of using proven CPU product and no software/tools change is required. We have measured the overheads of the SEU and SET tolerant **Cortex-R4** with respect to the original **Cortex-R4**. **Cortex-R4 TMR** runs 31% slower than **Cortex-R4**. It also occupies twice the chip area as the **Cortex-R4** with 104% total dynamic power overhead when both CPUs run at the clock frequency of **Cortex-R4 TMR**.

The results of this study demonstrate that by triplicating all of the flip-flops at gate-level can deliver radiation protection for mission-critical systems. The overhead of the TMR has been outlined for the **Cortex-R4**. Combined with other features, such as the deterministic behavior, ECC/parity protection, dual-core redundant lock-step configurations, the Cortex-R Series can offer advanced protection in harsh environments.

References

[1] K. L. Bedingfield, R.D. Leach, and M.B. Alexander, "Spacecraft System Failures and Anomalies Attributed to the Natural Space Environment", NASA Reference Publication 1390, August 1996.

[2] R. Mariani and P. Fuhrmann, "Comparing Fail-safe Microcontroller Architectures in Light of IEC 61508", 22nd IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 2007.

[3] T. Calin, M. Nicolaidis, R. Velazco, "Upset Hardened Memory Design for Submicron CMOS Technology", IEEE Transactions on Nuclear Science, New York, v.43, No.6. Dec 1996.

[4] ARM, "Cortex-R4 and Cortex-R4F Technical Reference Manual", http://infocenter.arm.com/help/topic/com.arm.doc.ddi0363e/DDI0363E_cortexr4_r1p3_trm.pdf, 2009.