Evaluation of critical flip-flops in RISC-V cores using fault injection for improved single-event-upset resilience

Darek Palubiak[1], Vitali Karasenko[1], and Connie O'Shea[1]

[1]Cadence Design Systems, Cork, Ireland

Abstract

In this paper, we present the application of fault injection simulations to identify and selectively harden the most ciritical flip-flops of RISC-V microprocessors. The critical flip-flops are obtained by calculating the Architectural Vulnerability Factor (AVF) using several different processor workloads upon injection of Single-Event-Upset (SEU) faults. The impact of selective hardening on the RISC-V cores is assessed by comparing power, performance and area (PPA) metrics.

Introduction

Over the years, CMOS scaling through the application of Moore's Law has led to a significant improvement in the speed and functionality of microprocessors (μP) and integrated circuits (IC). This continual increase in transistor density has led to larger silicon die areas as well as an increase in circuit complexity which, in turn, introduces negative side-effects such as potential security issues and greater susceptibility to adverse environmental conditions.

One of the most challenging operating conditions for a μP occurs when it is deployed in radiation environments where heavy ions or high-energy protons are present. These highly energetic radiation particles can induce transient faults in μPs by their interaction with the silicon lattice whereby the logic state of a circuit can become inverted. A single-event upset (SEU) occurs when such a transient fault appears in a sensitive node of a memory cell provoking a bit-flip. [1]. For μPs used in safety-critical applications such as automotive, the functional safety standard ISO 26262 requires that failures due to SEU must be thoroughly analyzed during the development process such that the risk of unpredictable data corruption or system malfunction is lower than the required threshold [2].

Another increasingly important and challenging aspect of μP deployment is due to the presence of adversaries that can physically tamper with the devices' operation with the aim of extracting confidential information or corrupting the normal operating function. For example, through the injection of deliberate faults to induce SEU into a cryptographic device and the observation of the corresponding erroneous outputs, attackers can drastically reduce the number of experiments needed to obtain the bits of a secret key [3]. As a result, the secure implementation of cryptographic primitives that are immune to fault injection attacks remains a significant challenge.

Many different approaches relying on spatial and/or temporal redundancy can be used to reduce μP sensitivity to faults. Dual and Triple Core LockStep (DCLS/TCLS) techniques replicate the whole processor and execute the same application in each core simultaneously, comparing the code execution in every clock-cycle to detect any mismatchs and recover the system state in the presence of faults [4]. While this approach is practical for fault mitigation, it introduces additional design effort and significant cost in terms of power, performance and area (PPA), which are very important constraints in many applications.

Although the techniques that replicate the entire μP core certainly can offer very high levels of reliability, these approaches can result in over-hardening of the core, resulting in the hardness of a μP exceeding its requirements with an unecessary penalty in PPA. Another approach is to selectively replicate only the most critical elements of a μP whose errors will significantly affect the correct system functionality, resulting in a more reasonable PPA overhead. The main challenge of this approach is to select the critical flip-flops in the μP for selective hardening.

In this work, we employ a SEU-aware verification flow to enable selective hardening of RISC-V μP cores based on triplicating a subset of sequential cells identified as critical through extensive fault injection (FI) campaigns. As the impact of soft errors mainly manifests as bit-flips of core memories and registers comprised of flip-flops, we consider register faults as the only source of the soft errors, since the core memories are commonly protected by fault-tolerant Error Correction Coding (ECC) techniques. We focus on SEU only, as they are the most common type of faults. To provide verifiable hardening as early as possible in the design flow, we primarily focus on Register Transfer Level (RTL) simulations. The proposed methodology will be shown to result in a sizable reduction of failures due to SEU without resorting to the use of wholesale redundancy or modification of the original RTL, thereby achieving the lowest-possible system costs while minimizing additional design effort and fulfilling the reliability requirements.

Methodology

Fault injection (FI) simulation is one of the most frequently used approaches to make quantitative decicions on the criticality of individual components in μPs. Applications running on μPs may exhibit a variety of different failure

modes and the severity of these failures as they relate to individual sequential elements can be quantified by the Architectural Vulnerability Factor (AVF). The AVF for a flip-flop *j* was defined as [5] :

$$AVF_j = \frac{1}{N}\sum_{i=0}^{N-1} \omega_i^j$$

where *N* is the number of injected faults, and the weights ω are assigned a value of 1 or 0 based on whether or not the injected fault had caused a failure to occur in the simulation run *i* when the fault was injected on flip-flop *j*. A simulation was classified a failure under the following scenarios:

- An exception was raised, for example, due to an ilegal instruction or out-of-bound memory access and the program crashes.
- Simulation timed out before the final instruction was retired.
- Software detected corrupted data in the data-path.

We investigated three different 32-bit in-order RISC-V cores implementing the RV32ICM instruction set: a 4-stage pipelined CV32E40P core from OpenHW Group [6], a 4 stage pipelined VeeR EL2 core from CHIPS Alliance [7], and a single-stage PicoRV32 core from YosysHQ [8]. We selected these small in-order cores because they provided reasonable run-times for comprehensive FI campaigns and because they illustrate the main features of microarchitectural design. More complex processors are left for study in future work.

The AVF of each flip-flop of the μP cores was computed by running FI campaigns with Cadence Xcelium™ Safety App under several different processor workloads. Coremark and Dhrystone benchmarks were selected as baseline general purpose compute workloads. Additional custom C and assembly programs that were present in the repositories' verification environemnt were also included in the FI campaigns to increase the code coverage. For each core under investigation, *M* workloads were applied for the FI campaign. For each workload, SEU faults were injected at eight randomly chosen times. As a result, for each flip-flop in the design, $N=M\times 8$ fault simulations were executed to evaluate its AVF.

At the end of each fault simulation, the simulation log files were examined to determine whether or not the test was successful. Each flip-flop was assigned an AVF score reflecting the number of simulation runs where a fault caused an error in the execution state. The flip-flops were classified as vulnerable if their AVF was found to be non-zero at the end of the FI campaign. Table I presents the results of the FI campaigns as the ratio of vulnerable flip-flops per pipeline stage in each RISC-V core that was investigated. Although the table indicates that the AVF of the VeeR EL2 core was the lowest, it also had the lowest coverage score due to the limited workloads avaialble in its repository. More workloads are required to increase code coverage of this core to bring it in line with the other two.

Table 1: AVF calculations.

Core	Pipeline stage			
	IF	DEC	EX	LSU
CV32E40P	186/329 (56%)	842/1701 (49%)	7/113 (6%)	6/42 (14%)
VeeR EL2	133/1279 (10%)	60/2096 (3%)	18/632 (3%)	32/1454 (2%)
PicoRV32	843/2115 (40%)			

Discussion

The FI jobs were dispatched on a load sharing facility (LSF) cluster where a maximum of 500 batch jobs were excecuted in parallel. The run-time of the longest FI campaign was 376 minutes, while the shortest was 9 minutes. Further work will be focused on improving run times and implementing triple mode redundancy (TMR) on the identified critical flip-flops. A Unified Safety Format (USF) file can be used to describe the implementation of TMR. With a USF file defined, TMR can be automatically inserted by Genus™ Synthesis Solution into the gate-level netlist. Table II shows the preliminary post-synthesis results comparing the TMR implementation of the cores in a generic 45 nm PDK. Innovus™ Implementation System will be used to drive the physical implementation accordingly to assess the total PPA overhead of the selectively hardened the RISC-V μP cores.

Table 2: Preliminary post-synthesis gate counts.

Core	No TMR	Full TMR	Partial TMR
CV32E40P	13,337	26,919	15,695
VeeR EL2	57,850	99,375	59,123
PicoRV32	9,519	24,322	11,983

References

[1] Baumann R.C. "Radiation-Induced Soft Errors in Advanced Semiconductor Technologies". In: *IEEE Trans. Device Mater. Reliab.* 5.3 (Sep. 2005), pp. 258-266. doi: 10.1109/JPROC.2012.2188769.

[2] Sherer A. et. al. "Ensuring functional safety compliance for ISO 26262". In: *Proc. 52nd Annual Design Automation Conference*. (Jun. 2015), pp. 1-3. doi: 10.1145/2744769.2747924

[3] Barenghi A. et. al. "Fault Injection Attacks on Cryptographic Devices: Theory, Practice and Countermeasures". *Proc. IEEE*. (Nov. 2012), pp. 3056-3076, doi: 10.1109/JPROC.2012.2188769.

[4] Kasap S., et. al, "Novel Lockstep-based Approach with Roll-back and Roll-forward Recovery to Mitigate Radiation-Induced Soft Errors," *IEEE Nordic Circuits and Systems Conference (NorCAS)*, (2020), pp. 1-7, doi: 10.1109/NorCAS51424.2020.9265137.

[5] Bottoni C., et. al, "Partial triplication of a SPARC-V8 microprocessor using fault injection". *IEEE 6th Latin American Symposium on Circuits & Systems (LASCAS)*, (2015), pp. 1-4, doi: 10.1109/LASCAS.2015.7250415

[6] https://github.com/openhwgroup/cv32e40p

[7] https://github.com/chipsalliance/Cores-VeeR-EL2

[8] https://github.com/YosysHQ/picorv32