Fault Recovery for Coarse-Grained TMR Soft-Core Processor Using Partial Reconfiguration and State Synchronization

Karel Szurman, Zdeněk Kotásek
Brno University of Technology, Faculty of Information Technology, IT4Innovations Centre of Excellence
Bozetechova 1/2, 61266 Brno, Czech Republic

{iszurman, kotasek}@fit.vutbr.cz

Keywords. TMR, fault recovery, state synchronization, processor, FPGA reconfiguration

Abstract

SRAM FPGAs are being more commonly integrated into safety-critical systems nowadays. These digital circuits can provide suitable platform for a fault tolerant system implementation meeting the trade-offs between performance, reliability, cost and hardware resources. However, SRAM technology is vulnerable to radiation-induced faults and mainly to Single Event Upset (SEU) effect. The SEU can cause "bit-flip" faults in SRAM memory cells which may affect internal FPGA routing (clock and reset signals), user memory (flip-flops, block RAM) and the functionality of implemented circuits. SEU mitigation must be implemented into the safety-critical design to achieve required system reliability in the harsh environment. SEU mitigation strategy may combine hardware redundancy and Partial Dynamic Reconfiguration (PDR) in order to implement error detection, self-repair ability and fault recovery mechanism into the system. With respect to the compromise between the system reliability and the resource overhead, various hardware redundancy schemes can be used. The most used form is Triple Modular Redundancy (TMR) which can be applied on different granularity levels in the system design. Coarse-grained TMR and PDR are often combined in one reconfigurable architecture. The time between SEU occurrence and the completion of fault recovery become a crucial parameter because the reliability of the TMR with one failed replica is worse than the reliability of an unprotected system. The fault recovery process can be generally divided into three phases: 1) fault detection, 2) fault removal by reconfiguration of a region containing replica identified as faulty, and 3) state synchronization bringing the reconfigured replica into the operating state consistent with other correctly operating replicas.

Combination of TMR and PDR is the approach also often addressed by fault mitigation methods designed for soft-core processors. The processor state is stored in internal memories and various architectural registers. After a faulty processor replica is reconfigured, its internal registers holding the processor state need to be synchronized with their up-to-dated copies from other processors replicas which were correctly operating. Various approaches had been proposed by researchers in the past. Four different synchronization methods which balance differently the trade-off between the synchronization speed and hardware overhead are evaluated in [2]. Synchronization of processors in known-blocking state by dumping and reading all processors data through shared Wishbone interconnection memory is presented in [3]. The largest amount of data which needs to be synchronized is the content of internal memories. With respect to a huge resource overhead, the use of a shared memory accessible from all three processor replicas is only practice solution. The critical part of the processor state synchronization is the maintenance of all internal registers. This requires implementation of a synchronization mechanism directly in the hardware to enable access to all registers and to minimize the synchronization time.

We propose a fault recovery mechanism for soft-core processor NEO430. In our PDR design, the NEO430 CPU core is protected by reconfigurable TMR architecture. In the TMR, the same input signals are shared between all CPU replicas and their output signals are brought into the majority voters. Each TMR voter is enhanced by additional error detection logic for identification of a failed CPU. The FPGA design floorplan is divided into two static and dynamic areas. Replicated CPU instances are placed into dedicated Partial Reconfiguration Modules (PRMs) in the dynamic area. Other design components are static; including reconfiguration controller GPDRC, synchronization controller and TMR voters. In the experiments, the reconfiguration of specific PRM corresponding to the faulty CPU replica is started based on the PRM error vector generated by TMR voters. A test application executed by triplicated CPUs periodically checks the digital input for activation of the synchronization enable request. This signal is generated by the GPDRC after the reconfiguration is finished. Afterwards, repaired CPU is restarted. During its startup, the test application reads the digital inputs and checks if request for synchronization is active. Since the request was activated by GPDRC, the CPU switches into the SLEEP mode. When the application executed by other operating CPUs is in a state suitable for synchronization, it will indicate readiness for the hardware synchronization through processor digital output to a synchronization controller. This is special circuit responsible for parallel addressing of all synchronized registers and their copying from the correctly working CPUs to the recovered one. Then, operating CPUs go into the SLEEP mode as well. In this state, CPUs are waiting for an external IRQ generated by the synchronization controller which will activate normal operating mode. In parallel, the synchronization controller performs synchronization of all internal registers while CPUs are idle in the SLEEP mode. After the hardware synchronization phase is finished, the external IRQ signal is triggered to bring CPUs again into the operating mode. Since that moment, all CPUs continue in synchronized program execution and with consistent data stored in the internal registers. By this FT design, we demonstrated possibility to implement a fault recovery mechanism for soft-core processor with the state synchronization logic embedded into the processor architecture and with the non-blocking CPU execution aware of fault recovery phases.

Paper origin

This paper has been accepted and presented at the 22nd International Symposium on Design and Diagnostics of Electronic Circuits and Systems in Cluj-Napoca [1].

Acknowledgment

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project IT4Innovations excellence in science – LQ1602 and the BUT project FIT-S-17-3994.

References

[1] Szurman, K.; Kotasek, Z.: Run-Time Reconfigurable Fault Tolerant Architecture for Soft-Core Processor neo430. 22nd International Symposium on Design and Diagnostics of Electronic Circuits and Systems. Cluj-Napoca: IEEE Computer Society, 2019, pp. 136-140. ISBN 978-1-72810-072-2.

[2] Kretzschmar, U; Gomez-Cornejo, J.; Astarloa, A.; Bidarte, U.; Del Ser, J.: Synchronization Of Faulty Processors In Coarse-Grained TMR Protected Partially Reconfigurable FPGA Designs. Reliability Engineering & System Safety, 2016.

[3] Morillo, A.; Astarloa, A.; Lazaro, J.; Bidarte, U.; Jimenez, J.: Knownblocking synchronization method for reliable processor using tmr & dpr in sram fpgas. VII Southern Conference on Programmable Logic (SPL), April 2011, pp. 57-62.