Design of A Soft Error Detection Technique for FPGA Based Softcore Processors

Ishan M Safarulla
PG student, VLSI & Embedded Systems, ECE Department
TKM Institute of Technology
Karuvelil P.O, Kollam, Kerala-691505, India

Karthika Manilal
Assistant professor, ECE Department
TKM Institute of Technology
Karuvelil P.O, Kollam, Kerala-691505, India

Abstract

SRAM-based FPGAs are susceptible to radiation-induced temporary faults called as single-event upsets (SEUs) or Soft errors. Soft errors affects or changes only some logic states of memory elements, but the device itself is not permanently damaged. SEUs may directly alter the logic states of any static memory element or induce changes to configuration memory. A new fault detection system architecture can be incorporated on any SRAM based FPGA with integrated soft core processors. It allows for detection of error in the system and also detects the processor with the error, so that the system can continue execution with the fault free processor. The fault detection system consists of a Lockstep scheme which is based on DWC technique. Lockstep scheme is built using a pair of Picoblaze cores, Comparator, and a MUX module. Lockstep Scheme detects the presence of error in the system but fails to point in which core, error is present. The Faulty core is detected using RESO Method which is based on DWC-CED technique. It uses the principle of time redundancy where the computations are performed repeatedly and is done differently. The coding is done in VHDL language, synthesized using Xilinx ISE 13.2 and simulated using ISim.

Keywords—*Soft Error, SRAM, FPGA, DWC, RESO, DWC-CED, Fault detection, Soft core processor***.**

IJERT

1. Introduction

Field Programmable Gate Arrays (FPGA) are well known devices concerning reconfigurable hardware. FPGAs consist of an array of programmable logic blocks of potentially different types, including general logic, memory and multiplier blocks, surrounded by a programmable routing fabric that allows blocks to be programmably interconnected. The array is surrounded by programmable input/output blocks, labelled I/O, that connect the chip to the outside world. The “programmable” term in FPGA indicates an ability to program a function into the chip after silicon fabrication is complete Every FPGA relies on an underlying programming technology that is used to control the programmable switches that give FPGAs their programmability. There are a number of programming technologies present and their differences have a significant effect on programmable logic architecture which include EPROM, EEPROM, flash, static memory, and anti-fuses.

SRAM-based FPGA devices are steadily becoming the most suitable platform for implementing modern embedded applications due to their high re-configurability, low cost and availability. Static memory cells are the basis for SRAM programming technology which are distributed throughout the FPGA to provide configurability. SRAM programming technology has become the dominant approach for FPGAs because of its two primary advantages:

re-programmability and the use of standard CMOS process technology[1]. The re-programmability leads to high logic density in terms of SRAM memory cells. Due to high logic density in terms of SRAM memory cells, SRAM based FPGA's are sensitive to radiation and require protection to work in harsh environments. Due to the increasing integration density FPGA chips are getting more and more prone to faulty behaviour caused by cosmic or artificial radiation .Such faults are modelled as Single Event Upsets (SEUs).

Transient faults, also called Single Event Upset (SEU), are the major concern in space applications**,** with potentially serious consequences for the Spacecraft, including loss of information, functional failure, or loss of control. SEU occurs when a charged particle hits the silicon transferring enough energy in order to provoke a bit flip in a memory cell or a transient logic pulse in the combinational logic[4]. It is imperative that FPGA based applications, where high reliability is required, include mechanisms that can easily and quickly detect and correct SEUs. Many techniques have been developed to protect critical systems on SRAM FPGAs against SEU[6]. At the design level of the FPGA these techniques are classified as SEU mitigation techniques which prevent SEU to disturb the normal operation of the target design, and SEU recovery techniques that recover the original programmed information in the FPGA configuration memory after an upset. The most common SEU mitigation techniques employ hardware redundancy like Triple Modular Redundancy (TMR), Duplication with Comparison (DWC), Duplication with Comparison with Concurrent Error Detection (DWC-CED) and Error Correcting Codes (ECC).

This paper is organized as follows. Section 2 focuses on Softcore processors, single event upsets and some SEU tolerant techniques applied to FPGAs and to ASICs. In Section 3, proposed method; fault detection in Softcore processors incorporated in SRAM based FPGA's, ie, Lockstep scheme using DWC technique and RESO method using DWC-CED are discussed. In section 4 the simulation results of the proposed techniques were discussed. Conclusions and ongoing works are discussed in section 5.

IJERT

2. Literature Survey

2.1 Softcore processors

Soft-core processors are pretested and predesigned intellectual property microprocessors whose architecture and behaviour are fully described using a synthesizable subset of a hardware description language (HDL). They can be synthesized for any Application-Specific Integrated Circuit (ASIC) or Field Programmable Gate Array (FPGA) technology; therefore they provide designers with a substantial amount of flexibility. It can be designed for a reprogrammable fabric such as an FPGA[3].

The two key attributes of soft processors are: 1) the ease with which they can be customized and subsequently implemented in hardware and 2) that they are designed to target the fixed resources available on a reprogrammable fabric. Soft processors are compelling because of the flexibility of the underlying reconfigurable hardware in which they are implemented and so it is possible to customize the soft processor architecture to match the specific needs of a given application by allowing designers to tune the processor with additional hardware instructions and also by sub setting the ISA of the processor to better match their application requirements.

2.2 Single Event Upsets

Single event upset (SEU) is defined as "radiation-induced errors in microelectronic circuits caused when charged particles (usually from the radiation belts or from cosmic rays) lose energy by ionizing the medium through which they pass, leaving behind a wake of electron hole pairs. A single charged particle can hit either the combinational or sequential logic in the silicon. When a charged particle strikes a memory cells sensitive nodes, such as drain in an off state transistor, it generates a transient current pulse that can mistakenly turn on the opposite transistors gate[4].

The effect can invert the stored value ie, produce a bit flip in the memory cell. This effect is called SEU or soft error. SEUs are soft errors, and are non-destructive. An SEU may occur in analogue, digital, optical components, or may have effects in surrounding interface circuitry. SEUs are also called as

soft errors, since it can be corrected by resetting or rewriting of the device and also only some logic state(s) of memory element(s) are changed but the circuit/device itself is not permanently damaged.

In FPGAs, SEUs may directly corrupt computation results or induce changes to configuration memory. Upsets in configuration memory can be detected by comparing its contents with a known, good state and can then be corrected by refreshing the state of memory. Static upsets in configuration memory do not affect functionality. Upsets need to be corrected only to ensure that errors do not accumulate. And transient fault changes the mapped circuit permanently, when it hits the memory. In addition to affecting memory, charged particles also change the logic function of the mapped circuit when they hit the on-chip configuration[4]. In this process, partial re-configuration is used to correct the upsets once the errors are detected without interfering with the operation of the loaded design. Transient faults occur because of radiations, electromagnetic interference, and power glitches. SEUs can affect both combinational and sequential circuits. It cause transient pulses in combinational logic paths. SEUs in configuration memory may result in modifications of the functionalities of the application design the FPGA implements. For a given design, all configuration memory bits can be classified as being sensitive (whose upset induces errors) and non-sensitive[1]. This is because among numerous configuration memory bits only some are actually utilized by the user's design, hence SEUs affecting the configuration bits that are not utilized by a specific design will not affect the behaviour of that design. Although of temporary nature, SEUs may have permanent effects until the device is reconfigured, e.g., by read back or scrubbing. The sensitive bits can be further categorized into two following categories:

Non persistent bits are those configuration bits which, when upset, may induce non persistent functional errors which disappear once the device is reconfigured, so that the design can return to normal operation. The non persistent bits generally involve purely combinational circuitry of the design.

Persistent bits are those configuration bits which, when upset, induce persistent functional errors, which do not disappear even after the device is reconfigured. The persistent bits generally involve any part of the design that contains the sequential circuitry or BRAM. An internal reset of the registers and flip-flops is a feasible solution to avoid such types of functional errors.

2.3 SEU Mitigation Techniques

Mitigation Technique is a process of applying design techniques to strengthen the functional integrity of the circuit, and protect it from the effect of any Single Event Upset. Fault-tolerant methods used to mitigate logic errors in FPGA based on redundancy technique are as follows. **Duplication with Comparison (DWC)** for detecting the presence of faults in the system, **Duplication with comparison with Concurrent Error Detection(DWC-CED)** for detecting the faulty module in the system and **Triple Modular Redundancy (TMR)** with majority voter for masking faults.

A SEU immune circuit may be accomplished through a variety of mitigation techniques based on redundancy. Redundancy is provided by extra components (hardware redundancy), by extra execution time or by different moment of storage (time redundancy), or by a combination of both. DWC is a simple hardware redundancy to detect errors in the circuit. Duplication with Comparison (DWC) is a detecting technique, in which the circuit to be protected is replicated twice and the results produced by the original circuit and the outputs of replicated circuits are compared to detect faults. The comparator circuit detects differences in the operation of the two circuits and signals the system with an error flag[5]. Figure 1 shows the duplication with comparison method. To begin, the circuit is duplicated to create two identical designs. Circuit duplication is accomplished by duplicating each primitive instance in the original design.

Two identical circuits module1 and module2 receive the same inputs and simultaneously execute the same instructions, their results are compared step by-step at each clock cycle. Circuit module2 generates the reference results to be compared against those of module1 that provides the system output.

IJERT

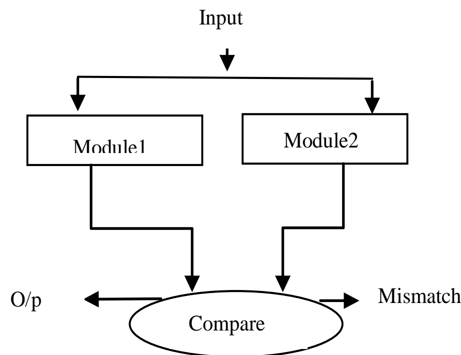

Figure 1: Duplication with Comparison

Basically, DWC is able to detect but not to correct errors and also fails to indicate the fault location, since it cannot point out the faulty circuit. DWC offers so many advantages such as easy to apply to any circuit, can be used to detect a variety of errors, can detect errors immediately and allow the system to quickly respond to circuit problems and requires limited external hardware support.

DWC-CED combines DWC method with a CED (Concurrent Error Detection) machine based on time redundancy that work as a self checking block. Figure 2 shows the DWC-CED technique based on time redundancy for fault detection.

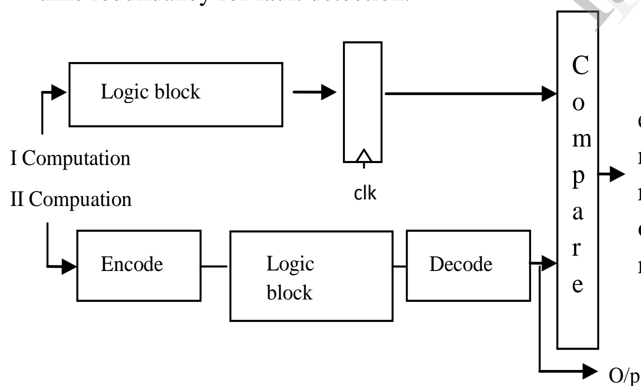

Figure 2: Time redundancy for fault detection

DWC detects the faults in the system and CED detects which block is fault free. The important characteristic of this approach is that there is always a correct value in the output of the scheme in the presence of a single fault, because the mechanism is able to detect the faulty module and to select the correct output of the two.

CED which is based on time redundancy is the way to detect a fault without area overhead. The basic concept of time redundancy is the repetition of computation in a way that allows the errors to be detected. To allow redundancy to detect faults, the repeated computations are performed differently. During the first computation, the operands are used directly in the combinational block and the result is stored for further comparison. During the second computation , the operands are modified, prior to use, in such a way that errors resulting from permanent faults in the combinational logic are different in the first calculation than in the second and can be detected when results are compared. These modifications are seen as encode and decode processes.

3. Proposed Method `

This section presents the Lockstep scheme, a method for detecting the presence of fault in the system and RESO (REcomputing with Shifted Operands) method to detect the faulty module in the system. The lockstep scheme is based on DWC technique at processor level and RESO method is based on DWC-CED technique.

3.1 Lockstep Scheme

The Lockstep scheme based on DWC technique (Duplication with Comparison), detects the presence of fault in the system without interrupting normal functioning.

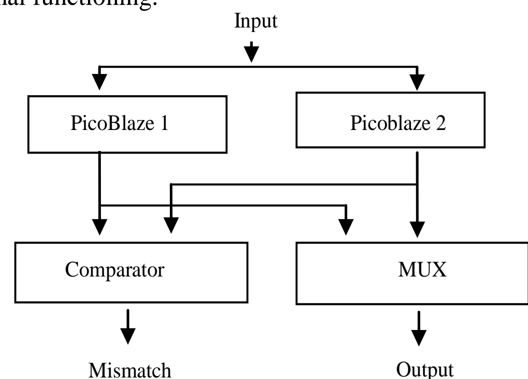

Figure 3: Lockstep Scheme

Lockstep scheme consists of a Pair of Picoblaze cores with one faulty, comparator and a multiplexer. Figure 3 shows the lockstep scheme for detecting the presence of error defined based on a hardware redundancy technique known as Duplication with comparison.

IJERT

The PicoBlaze is a compact, capable, and cost-effective fully embedded 8-bit RISC virtual soft core optimized for the Xilinx FPGA families. The PicoBlaze core is totally embedded within the target FPGA and requires no external resources[8]. It is extremely flexible. The PicoBlaze provides abundant, flexible I/O at much lower cost than off-the-shelf controllers.

The PicoBlaze is delivered as synthesizable VHDL source code, so the core is future-proof and can be migrated to future FPGA architectures. Being integrated within the FPGA, the Pico Blaze reduces board space, design cost, and inventory[8]. The Constant (k) Coded Programmable State Machine (KCPSM) solution is a fully embedded 8-bit microcontroller macro for different FPGA devices. The PicoBlaze soft core is available in different versions of KCPSM module[9]. KCPSM3 version is used here.

Two identical picoblaze cores are the essential parts of the Lockstep scheme. These two cores are provided with the same input. Their outputs are identical during fault-free functioning, any mismatching indicating error(s). In order to distinguish the faulty circuit, two blocks used are the Comparator (COMP) and Multiplier (MUX). COMP indicates any mismatch between the outputs of Picoblaze core1 and picoblaze core2. The mismatch signal from the comparator output can be used as the trigger signal to the system for the error recovery. MUX connects one of the cores to the system output, so that if one of them is reported to be faulty, the other is switched on. Lockstep scheme indicates the presence of errors in the core but fails to indicate in which core the error is present.

3.2 DWC-CED technique using REcomputing with Shifted Operand (RESO) method

The RESO method based on DWC-CED technique detects the faulty core in the system and switches the correct core to output. Recomputing with shifted operands (RESO) is one of methods of Concurrent error detection (CED). The basic idea is to modify the operands before performing the recomputation so that an error affects different parts of the circuits[6]. Here the computations are carried out twice, once on the basic input and once on the shifted input. Result from these two operations is compared to detect an error.

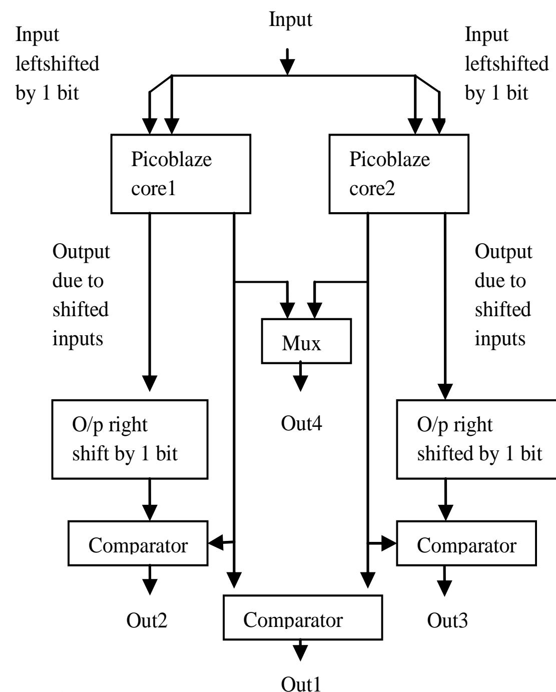

Figure 4: DWC-CED technique (RESO Method)

Figure 4 shows the DWC-CED technique using RESO method. The concurrent error detection method RESO uses the principle of time redundancy where the coding function is the left shift operation and the decoding function is the right shift operation. Thus, in the first computation, the operands are computed and stored in a register. At the second computation, the operands are shifted k bits to the left, computed and the result is shifted k bits to the right .The result of the second step is compared to the previous result stored in the register. A mismatch indicates the presence of a fault in the module.

Using DWC combined with CED for detecting faults, it is possible to take advantage of the simple comparison at the output of the duplication scheme to inform whether it is necessary to re-compute the data for permanent fault detection[6]. The re-computation is needed only when a mismatch of the outputs occurs. If an output mismatch occurs, the output register will hold its original value for one extra clock cycle, while the CED block detects the permanent fault. After this, the output will receive the data from the fault free module until the next reconfiguration (fault correction). The important characteristic of this method is that it does not incur performance penalty when the system is operating free of faults or with a single fault. The

IJERT

method just needs one clock cycle in hold operation to detect the faulty module, and after that it will operate normally again without performance penalties. The final clock period is the original clock period plus the propagation delay of the output comparator.

The two identical picoblaze cores are provided with same inputs. The input is then left shifted by 1 bit and provided to the picoblaze cores along with the actual input. So each picoblaze core produces 2 outputs. One due to actual inputs and other by the left shifted inputs. The output due to actual inputs from both cores is then compared. If it shows a mismatch it indicates that one of the cores is faulty. Now to determine which core is faulty, the output due to left shifted inputs from each core is right shifted by 1 bit and then compared with the actual output of the corresponding core. If the comparator shows a mismatch, it indicates the corresponding core is the faulty one. Now the correct core is switched to output by using MUX module.

4. Simulation Results

The design entry is modelled using VHDL in Xilinx ISE Design Suite 13.2 and the simulation of the design is performed using Isim from Xilinx ISE to validate the functionality of the design.

Lockstep scheme which is based on Duplication with Comparison technique defined at the processor level is designed using a pair of picoblaze cores, a comparator and a mux which detects the presence of error in the system. The outputs from picoblaze core 1 and core 2 are provided as the inputs to the comparator which compares the output of both the cores. The out1 port indicates the output of the comparator. If it is low, it indicates the mismatch between the outputs of picoblaze cores. Now once the core with the error is detected, the core without error is switched to the output and the core with error is switched to fault tolerant configuration engine for error recovery. The out4 port indicates the output of the mux module.

Figure 5 shows the simulation result of the lockstep scheme based on DWC technique to detect the presence of error in the system.

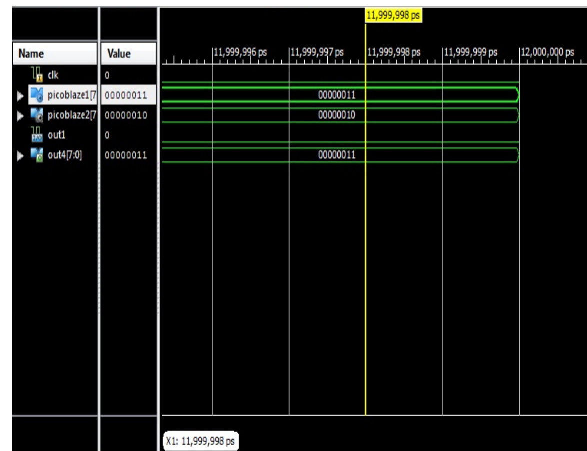

Figure 5: Simulation result of lockstep scheme

Re-computing with shifted operands based on DWC-CED technique identifies the core with error in the system. Figure 6 shows the simulation results of the RESO method to identify the core with error in the system and to switch the correct core to the output.

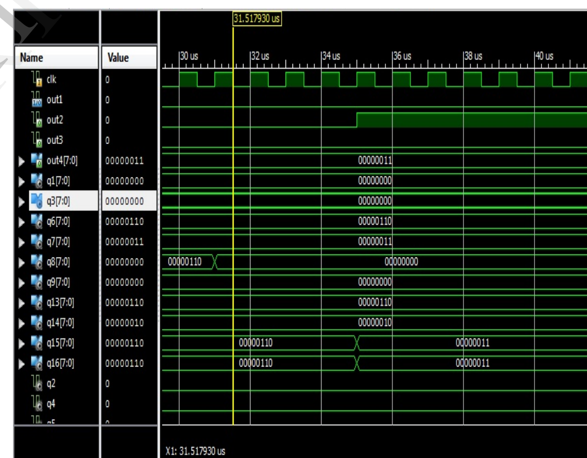

Figure 6: Simulation result of RESO method

Each core produces two outputs. One due to actual input and other due to shifted inputs. The output from picoblaze core1 due to actual input is signal q7 and output due to shifted input is shown by signal q6.The output from picoblaze core2 due to actual input is signal q14 and output due to shifted input is shown by signal q13.Now the actual output from both cores is compared using comparator indicated by signal out1. The out1 is zero, which indicates the mismatch between the actual outputs. The output due to shifted inputs is right shifted by 1 bit, ie, q6 and q13 is right shifted by

IJERT

one bit to obtain the signal q15 and q16 respectively. Now q15 is compared with q7 and the output of comparator shown by signal out2. Out2 is high which indicate the picoblaze core1 has no error. Now the signal q16 is compared with q14 and the output of comparator shown by signal out3. Out3 is low which indicate the picoblaze core2 has error. Now core without error, core1 is switched to the output by using MUX indicated by out4.

5. Conclusions

SRAM based FPGA's are sensitive to radiation induced faults and require protection to work in harsh environments due to it's to high logic density in terms of SRAM memory cells. SRAM based FPGAs are affected by radiation induced temporary faults called as single event upsets (SEUs) or soft errors. That may alter the logic states of any static memory elements. The paper is intended to design a new architecture for soft error detection technique which can be incorporated on any SRAM based FPGA with integrated Softcore processors. PicoBlaze is used as the Softcore processor, which is a compact, capable, and cost-effective fully embedded8-bit RISC virtual soft core optimized for the Xilinx FPGA families. Lockstep scheme based on DWC technique at the processor level which is used to detect the presence of error in the system and RE-computing with Shifted Operand (RESO) method based on DWC-CED to detect the core with error are designed and simulated.

IJERT

6. References

[1] Hung-Manh Pham, SebastienPillement, Stanisław J. Piestrak "Low-Overhead Fault Tolerance Technique for a Dynamically Reconfigurable Softcore Processor", *IEEE transactions on computers*, vol. 62, no. 6, june 2013

[2] Bertrand Le Gal and Christophe Jego "Soft core Processor Optimization According to Real-Application Requirements" *IEEE embedded systems letters* vol. 5, no.1,pp 4-7. march 2013

[3] P. Yiannacouras, J. G. Steffan, and J. Rose, "Exploration and customization of FPGA-based soft processors," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.,*vol. 26, no. 2, pp. 266–277, Feb.2007.

[4] DirettoredellaScuolaand Andrea Manuzzato*"*SINGLE EVENT EFFECTS ON FPGAs*"*, Vol 3, december 2006

[5] Jonathan Johnson, William Howes, and Michael Wirthlin "Using Duplication with Compare for On-line ErrorDetection in FPGA-based Designs"December 2007

[6] VatsyaTiwari , Pratap Singh Patwal " *Design and analysis of software fault-tolerant techniques for softcore processors in reliable sram-based fpga*"pratap Singh Patwal , Int. J. Comp. Tech. Appl., Vol 2 (6), 1812-1819 June 2006

[7] Xilinx "Error detection and correction Techniques for Virtex FPGAs" February 2005**.**

[8] PicoBlaze 8-bit Embedded Microcontroller User Guide for Extended Spartan®-3 and Virtex®-5 FPGAs Introducing PicoBlaze for Spartan-6, Virtex-6 and 7 Series FPGAs, UG129 June 22, 2011

[9] Ken Chapman PicoBlaze 8-Bit Microcontroller for Virtex-E and Spartan-II/IIE Devices (v2.1) February 4, 2003.