International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

A Novel Adaptive Technique to Mitigate Radiation Effects on FPGAS

T. Mary Daphne[1], Dr. T. Latha[2]

[1]Department of CSE, Udaya School of Engineering, Kanyakumari, Tamilnadu, India

[2]Department of ECE, St.Xaviers Catholic College of Engg, Chunkankadai, Kanyakumari, Tamilnadu, India

Abstract: ***The radiation effects on electronic components are a major concern for devices used in space environment. The change in configuration bits due to radiation effects may cause permanent or short term failures to the electronic devices such as flipflops, memories etc. Various techniques are being used for mitigating soft errors. This paper surveys some of the techniques used for providing fault tolerance in fpga based systems. One technique namely temporal data sampling is discussed here. A multilevel (ML) Flash memory on-chip error correction system design based on the concept of trellis coded modulation (TCM) is conferred to provide fault tolerance . Flash memory's combination of nonvolatility and easy in-system updateability are key attributes driving its adoption into today's system designs. A integrated memory structure is used for non volatile storage of bitstreams. This integrated memory structure integrates the advantageous features of volatile and non volatile memory.***

Keywords: Scrubbing Partial Reconfiguration, TMR, TCM, Reconfigurable Architecture, Weighted Voting, Critical Blocks, Critical Variable

1. Introduction

FPGA (Field Programmable Gate Array) is an integrated circuit containing gate matrix which can be programmed by the user “in the field” without using expensive equipment. An FPGA contains a set of programmable logic gates and rich interconnect resources, making it possible to implement complex digital circuits. The FPGA matrix also has dedicated memory blocks called Block SelectRAMs (BRAMs), clock delay-locked loops (DLLs) for clock distribution delay compensation and clock domain control and other components that vary according to the FPGA family. The characteristic of the CLB logic and slice may change consistent with the FPGA family.FPGA devices are produced by a number of semiconductor companies: Xilinx, Altera, Actel, Lattice, QuickLogic and Atmel. Configuration bitstream can be stored in FPGA using various technologies.

2. FPGA Implementation Techniques

Memory technology, therefore, is central to the concept of an FPGA, and the performance characteristics of FPGAs directly reflect the memory technology used to construct them. Thus, FPGAs can be generally classified according to the type of memory structure used to store configuration data.

A. SRAM-based FPGAs

The majority of FPGAs is based on SRAM (Static RAM).SRAM-based FPGA[1][2] stores logic cells configuration data in the static memory (organized as an array of latches). Since SRAM is volatile and can't keep data without power source, such FPGAs must be programmed (configured) upon start.

There are two basic modes of programming:

- *Master mode*: When FPGA reads configuration data from an external source, such as an external Flash memory chip.
- *Slave mode* : When FPGA is configured by an external master device, such as a processor. This can be usually done via a dedicated configuration interface or via a boundary-scan (JTAG) interface.

The newest SRAM-based FPGA devices provide a runtime tool for detecting and eliminating the configuration faults: partial reconfiguration [3]. Partial reconfiguration allows part of the FPGA to be reprogrammed while the rest of the FPGA continues to run uninterrupted. Partial reconfiguration detects errors within the configuration memory and reprograms the faulty configuration to its original operation through a technique known as scrubbing. SRAM-based FPGAs are generally quite fast, but the volatile nature of SRAM requires that the configuration data be supplied externally at power up, often from an electrically erasable programmable read-only memory (EEPROM) chip. SRAM-based FPGAs have highest densities, but consume a lot of power and need an external non-volatile memory to store configuration bitstream. SRAM-based FPGAs with an internal flash module doesn't need an external configuration memory.

B. SRAM-based FPGAs with an internal flash memory

This type of FPGA is generally like the previous, except that these chips contain internal flash memory blocks, thus eliminating the need to have an external non-volatile memory.Internal non-volatile memory can be also useful to prevent unauthorized bitstream copying.

C. Flash-based FPGAs

The true flash-based FPGAs shouldn't be confused with the previous type. The SRAM-based FPGAs with internal flash memory use flash only during startup to load data to the SRAM configuration cells. On the contrary, true flash-based FPGA uses flash as a primary resource for configuration storage, and doesn't require SRAM This technology has an advantage of being less power consumptive. Flash-based

Licensed Under Creative Commons Attribution CC BY

FPGAs are also more tolerant to radiation effects. Flash-based FPGAs are generally slower than anti-fuse and SRAM-based FPGAs, but have the advantage that configuration data can be retained even when power is off, as well as a smaller cell size than SRAM. Flash-based and Antifuse based FPGAs consume much less power than their SRAM-based counterparts.

D. Antifuse-Based Fpgas

Antifuse-based FPGAs are different from the previous ones in that they can be programmed only once. Anti-fuse devices have the lowest cell size and can generally deliver fast performance, but can be programmed only once, restricting them to applications that do not require in-system reconfiguration.

3. Radiation Issues in FPGAS

A. Short-Term Failures

In the short term, they can cause transient upsets in circuits known as Single Event Upsets (SEUs). SEUs can cause state bits to change and logic outputs to evaluate incorrectly. SEUs are soft errors, and are non-destructive. An SEU may occur in analog, digital, optical components, or may have effects in surrounding interface circuitry. The FPGA‟ s configuration bits (Bit-stream) are used to configure both the logic elements and the routing switches. Upset of a programming bit in a FPGA is much more serious than a conventional data bit upset. If a logic element control bit changes state, then the logic functionality of the FPGA is altered. If a control-bit of a routing switch experiences an upset, then the FPGA essentially becomes rewired. In either case, the programmed circuit function is no longer what was intended. For these reasons, the configuration storage of the FPGA must be totally immune to SEU.

B. Permanent device failures

Radiation can cause permanent damage to silicon devices over time, rendering all or part of the device unusable. Radiation which results into Soft Error (SEU and SET) can cause permanent faults in reconfigurable architectures, for example when it hits configuration memory of the architecture. The configuration memory contents control the overall functionality of architecture through routing.

This bit-flip due to SEU or SET remains effective till the time bit-stream (configuration data) is not reloaded (scrubbing technique) or not corrected by dedicated hardware. The results from bit-stream fault injection [6] and radiation ground testing [7] have confirmed the efficacy of Triple Modular Redundancy (TMR)[8] structure combined with scrubbing, to recover upsets in reconfigurable architectures. However, the TMR with scrubbing technique has some limitations, such as an area overhead, does not cover SET faults and voting circuit faults.

The SRAM that is used to configure the different generations of Xilinx FPGAs is sensitive to Single-Event Upset (SEU). The effect of these upsets on the configurations depends on the application in which the FPGA device is implemented, including the number of configurations.

4. Soft Error Mitigation Technique for Reconfigurable Architectures

Reconfigurable architectures are becoming increasingly popular with space related design engineers as they are inherently flexible to meet multiple requirements and offer significant performance and cost savings for critical applications. Single event effects are most difficult to avoid in space-borne reconfigurable architectures.

We discuss a design technique based upon unique temporal data sampling with weighted voting. to cope with both SEU and SET faults. This design technique not only gives 100% fault recovery from SEU but also gives 100% fault recovery from SET, dual event faults and 50% recovery from triple event faults. The technique has an auto correction mechanism and does not need scrubbing; hence, it improves the overall performance and speed of the system. In the case of reconfigurable architectures, the problem of finding an efficient technique in terms of area, performance and power is very challenging due to high complexity of the architectures. An SEU is classified as a soft error but has permanent effects on a reconfigurable architecture. The consequences of SEU cannot be handled through standard ASIC fault tolerant schemes such as Error Correction and Detection Codes (EDAC) or standard TMR with single voter circuits because a fault in encoder/decoder or in voter circuit will invalidate the technique. TMR is an attractive scheme for reconfigurable architectures because it provides full hardware redundancy including user's combinational, sequential circuits, the routing and IO pads. However, TMR has limitations like area overhead, IO pad limitations, power dissipation and faults in voter circuitry itself.

In order to overcome the problems, described above and limitations of the schemes introduced so far, we present a mitigation technique which is based on Temporal Data Sampling with weighted voting.

A. Temporal Sampling

Temporal Data Sampling sampling technique eliminates all SETs in the clock and data signal. The circuit consists of six level sensitive latches (3 at the input stage and 3 at the output stage)Each level sensitive latch is in sampling mode at the high level of its clock signal and is in blocking mode when its clock signal is low. In blocking mode the latch holds the data and the data changes at the input are blocked. In sampling mode the latch is transparent to the data. The set of three latches (L1, L2), (L3, L4), (L5, L6) operate in parallel and form the temporal sampling stage of the technique. Each set of level triggered latches constitute a negative edge triggered Flip Flop. The temporal sampling stage stores data samples at different time intervals. These samples are used in weighted voting logic to eliminate single event upsets.

B. Weighted Voting Circuitry

Majority voting is commonly used in TMR systems. The proposed mitigation technique is based upon unique "weighted majority voting". Each node has been assigned a voting weight based on a simple rule which is "voting weight of a node is inversely proportional to the probability of that node being disturbed by the radiation". The total

Licensed Under Creative Commons Attribution CC BY

possible voting weight for any logic value can be 9. The output of each 2/3 simple majority voter is fed into the correction unit. The correction unit checks the output and the input to correct voter circuit faults and to override incase of fault. These corrected outputs are fed into the last majority voter circuit to resolve the final output. Final output is decided on the basis of a 2/3 simple majority. The majority voter circuit gives one additional output as recovery. Recovery output is only asserted when more than one SEU/SET fault is detected and corrected. The recovery output may request the main microprocessor/DSP of the reconfigurable architecture for scrubbing. As the scrubbing request is only sent when more than one SEUs are detected and corrected, this improves the system performance as scrubbing frequency is halved than previously introduced SEU mitigation schemes [9][10][11].The weighted voter circuitry has the capability to recover from internal faults. This unique feature enhances the reliability. In case of an SEU, the down load process of bit-stream is no longer required due to auto correction feature of the proposed technique. The auto correction mechanism is activated through recovery command. This feature makes the design very flexible and enhances the overall system performance.

5. Integrated Memory with Adaptive Error Recovery for FPGAS

Due the existing disadvantages in the existing system a novel technique named integrated memory with adaptive error recovery is adapted to mitigate the soft errors. It involves the integration of a integrated memory structure that combines the advantageous features of volatile and non volatile memories. Along with these a soft error recovery technique is used to eradicate and recover the soft errors.

Algorithm:

a) A new memory structure is formed by using new programming elements for routing switches and configuration elements
b) Considering the functioning program in use, the critical variables and critical blocks are identified.
c) On identification of the critical blocks computations are performed on the critical data.
d) Such computations are made twice and the resulting output is stored individually .
e) Recomputed results are compared to verify if they are consistent or not.
f) If the recomputed values are not consistent the particular program is identified from the back up known as golden memory.
g) If the recomputed values show consistency the program continues execution from the current block.

The detailed procedure can be explained using the following diagram

Figure 1: Block diagram for Integrated Memory with Adaptive

6. Error Recovery

A. New Programmable Element for Routing Switches

A new PE for the design of high speed routing switches is introduced.. The non-volatile device coupled with a word line transistor and a small pull down transistor acts as the PE for a thin gate oxide pass transistor that in turn acts as the switch, Data in this PE is non-volatile. The source node of the non-volatile device is tied to the gate input of the pass transistor, and supplies the non-volatile data. If the nonvolatile device is in the erased state, then applying read voltages to the non-volatile device causes a current to flow through the device. This will raise the gate voltage of the pass transistor to high, turning the switch on. If the non-volatile device is in the programmed state, no current will flow through the device. Any charge build up on the gate of the pass transistor due to leakage current from the device is pulled down through the bottom transistor (or alternatively, a diode) which is used to pull down the voltage. Programming the non-volatile device is performed by supplying 0 or 3 V to the input.

B. New Programmable Element for the Configuration of Logical Elements

An analogous approach can be applied to the non-volatile storage of configuration data for logical elements. Two of the new non-volatile memory devices are integrated with an nMOS latch .The inputs to the non-volatile memory devices consist of the control gate signal, the select gate signal, and the signal. The integrated non-volatile memory PE ("iNVM"), is used for LUT entries and configuration of multiplexers within a logical element. In order to store data onto the integrated non-volatile memory, data is placed onto the input port and the word-line is turned high. The nMOS latch is allowed to change its state to low high or high-low. Then, programming voltages are applied to the control gate, the select gate, and the drain of the two nonvolatile devices. The PE will retain its configuration even when the power is turned off.

C. New Non-Volatile Flip-Flop

These flip-flops temporarily store clock-driven data and are an important part of the logic block functionality. It contains the traditional structure for a flip-flop latch in the form of an SRAM as well as two additional non-volatile devices attached to the m and mb nodes of the SRAM cell. The WR signal is the write control signal for the flip-flop and is typically controlled by the clock. Operation of the flip-flop during normal operation is the same as that of a conventional flip-flop, utilizing the data-in and WR signals. One non-volatile device is to be programmed while the other device is to remain erased, reflecting the data stored in the latch. This completes a direct transfer of data from the flip-flop to the non-volatile parts. When the power is restored, only the erased device conducts current, raising the corresponding node voltage on only one side of the SRAM latch high and thereby restoring the contents of the flip-flop to its previous state. The possibility of storing the flip-flop states into non-

Licensed Under Creative Commons Attribution CC BY

volatile memory creates a number of new algorithm and application opportunities. By "remembering" the state of the state machine of the FPGA, a system can be quickly restored to its last state without having to reset the state machine. The non-volatile devices can be viewed as a high speed, low current "backup" memory for the flip-flop. By combining SRAM with the non-volatile devices, these block RAMs will operate with the speed of conventional SRAM while maintaining the option to store the data into non-volatile memory as needed. The presence of the non-volatile flip-flop and the non-volatile embedded block RAM create opportunities for power-down strategies, as some or all of the FPGA can be put into sleep mode or be powered down completely to reduce or eliminate leakage power. When power is restored, data from the non-volatile memory is used to restore the appropriate configuration information. Thus, the FPGA can be either locally or globally powered down when it will be known that the FPGA, or some portion of it, will not be used for a given period of time. In short, utilizing non-volatile flip-flops to store template information significantly decreases the time and energy costs of data transfers, which decreases the minimum power-down required time to recover the cost and provides for potentially shorter power-down cycles. This in turn leads directly to greater energy savings because, among other things, the use of integrated non-volatile memory enables the FPGA to be powered down when the traditional FPGA would need to be in standby mode.

D. Creation and storage of the backup of operational program

A back up of the progam is created and it is stored in the memory for further functioning.

E. Critical variable and critical block identification

The critical blocks and critical variables are identified using certain factors such as recursion, dependencies etc.

F. Soft error detection recovery

Using the knowledge of critical blocks, critical values are computed twice to produce outcomes which are stored as separate outcomes. If the values are identical then program execution is allowed to continue. Else specific error correction techniques are utilized by classifying the type of error involved.

F. Adaptive error recovery

Using this technique suitable error recovery procedure is selected based on the characteristics of the error. An FPGA-based space system can rapidly adapt to changing mission conditions and requirements. However, the more likely scenario is for the system to adaptively increase and decrease the fault tolerance as needed, since fault tolerance schemes incur significant penalties in terms of logic utilization, memory utilization, and power consumption/heat dissipation.

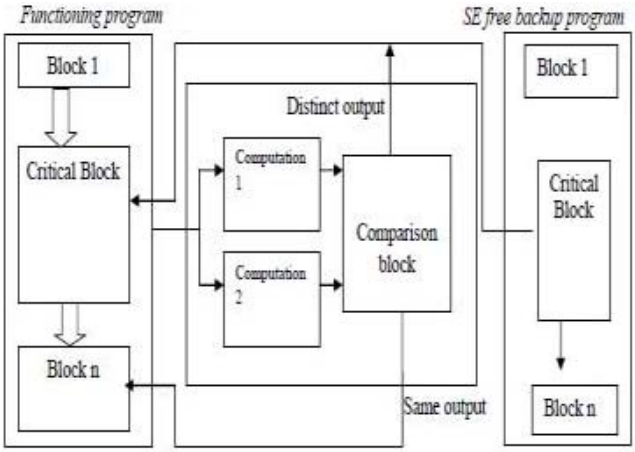

Figure 2: Soft Error Detection and Recovery

Adaptable Fault Tolerance

Adaptable fault tolerance [12] is useful for attenuating to varying radiation conditions. Different applications have different reliability requirements: The actual vulnerability of an application to faults can vary, not only between applications but also between different parts of the program. Output data for critical control systems would likely need full protection; however, temporary variables, such as those on the stack, may not need as much protection. Similarly, program data or read-only variables might not need distributed, multi-FPGA protection, as they can be reloaded from nonvolatile storage if found to be corrupt. The effect that a fault has on program execution can be classified into the following categories

Benign Fault: A transient fault which does not propagate to affect the correctness of an application is considered a benign fault. A benign fault can occur for a number of reasons. Examples include a fault to unused data or a fault to dead (unreachable) code.

Silent Data Corruption (SDC): A transient fault which goes undetected and propagates to corrupt program output is considered an SDC. Note that these errors are not always serious; a single bit flip in a digital image, for example, would appear merely as a tiny amount of noise.

Detected Unrecoverable Error (DUE): A transient fault which is detected without possibility of recovery is considered a DUE. Such an error can either cause obviously incorrect execution or force a processor or FPGA reset.

When error correction is desired, different error-correcting codes (ECC) such as Hamming or Reed-Solomon codes can correct for one or more bit errors. This work is interested in on-chip error correction system design for code storage ML Flash memory. The TCM-based design[13] solution can provide better coding redundancy vs. error-correcting performance trade-offs.

The motivation is two-fold:

1) The *more-than-two-levels-per-cell* storage capacity of ML memory makes the modulation process non-trivial and an integral part of the on-chip error correction system.
2) TCM can effectively integrate ECC with modulation and

Licensed Under Creative Commons Attribution CC BY

achieve significant gain over the conventional design practice that considers ECC and modulation separately.

7. Conclusion

Various implementation techniques for FPGA are reviewed. A design technique to cope with SEU and SET faults based on temporal data sampling and weighted circuitry is being analyzed. A set of new designs for core FPGA logic, routing, and flip-flop circuits that offer the speed advantages of SRAM and the area savings and non-volatility advantages of Flash is introduced . These designs are enabled by a non-volatile device that has the advantage of a low programming current that will not damage low voltage switching devices. FPGA architectures using the core FGPA elements described here offer the potential to dramatically improve power efficiency, area, and cost, and enable a whole new class of adaptive, power-aware application mapping methods.

Hybrid software fault tolerance approaches (combination of H/W and S/W and combination of various S/W fault tolerance schemes) produces better results.

References

[1] C. Bolchini and A. Miele, "Design space exploration for the design of reliable SRAM-based FPGA systems," in Proc. IEEE Int. Symp. Defect Fault-Tolerance in VLSI Syst., Cambridge, MA, 2008, pp. 332–340.
[2] Lima,Carro,L.,Reis ," Designing fault tolerant systems into SRAM based FPGAs" DAC03
[3] Carmichael C., Caffrey M., and Salazar A., XAPP216, "Correcting Single Event Upsets through Virtex Partial Configuration," June 2000, http://www.xilinx.com/support/documentation/application_notes/xapp216.pdf.
[4] Carmichael, C., Fuller, E., Fabula, J., Lima, F., "Proton Testing of SEU Mitigation Methods for the Virtex FPGA",Proc. of MAPLD, 2001.
[5] Massengill, L., "SEU Modeling and Prediction Techniques", IEEE Nuclear and Space Radiation Effects Conference Short Course Text, 1993.
[6] Lima, F., Carmichael, C., Fabula, J., Padovani, R., Reis, R., "A Fault Injection Analysis of Virtex® FPGA TMR Design Methodology", Proc. Of (RADECS), Sept. 2001.
[7] E. Normand, "Single Event Upset at Ground Level," IEEE Transactions on Nuclear Science, vol. 43, pp. 2742- 2750, 1996.
[8] Pratt, B., M. Caffrey, P. Graham, K. Morgan, and M. Wirthlin, "Improving FPGA Design Robustness with Partial TMR," 44th Annual IEEE International Reliability Physics Symposium Proceedings, 2006, pp. 226–232.
[9] Design of a Single Event Upset (SEU) Mitigation Technique for Programmable Devices
[10] Sexton,F.W."Measurement of Single Event Phenomena in devices and ICs",IEEE Nuclear and space Radiation effects Conference Short course Text,1992
[11] Mavis,eaton ,"SEU and SET mitigation techniques for FPGA cicuit and configuration bit storage Design" MAPLD International Conference
[12] D. Fay, A. Shye, S. Bhattacharya, D. A. Connors, and S. Wichmann, "An adaptive fault-tolerant memory system for FPGA-based architectures in the space environment," in Proc. NASA/ESA Conf. Adapt. Hardware Syst., Edinburgh, U.K., 2007, pp. 250–257.
[13] E. Biglieri, D. Divsalar, P. J. McLane, and M. K. Simon, Introduction to Trellis-Coded Modulation with Applications. Prentice Hall, 1992.
[14] RAIDwww4.comp.polyu.edu.hk/~csajaykr/my home/teaching/.../raid.pdf
[15] S. Gregori, A. Cabrini, O. Khouri, and G. Torelli, "On-chip error correcting techniques for new-generation flash memories," Proceedings of The IEEE, vol. 91, no. 4, pp. 602–616, April 2003.
[16] L. Grupp, A. Caulfield, J. Coburn, S. Swanson, E.Yaakobi,P.H.Siegel,andJ.K.Wolf, "Characterizing flash memory : anomalies, observations, and applications," MICRO-42, pp. 24– 33, December 2009
[17] K. Andrews, et. al., "The Development of Turbo and LDPC Codes forDeep-Space Applications," Proceedings of the IEEE, November 2007.

Licensed Under Creative Commons Attribution CC BY