University of South Florida

DIGITAL COMMONS
@ UNIVERSITY OF SOUTH FLORIDA

Digital Commons @ University of South Florida

USF Tampa Graduate Theses and Dissertations

USF Graduate Theses and Dissertations

3-24-2005

Partial Evaluation Based Triple Modular Redundancy For Single Event Upset Mitigation

Sujana Kakarla
University of South Florida

Follow this and additional works at: https://digitalcommons.usf.edu/etd

Part of the American Studies Commons

Scholar Commons Citation

Kakarla, Sujana, "Partial Evaluation Based Triple Modular Redundancy For Single Event Upset Mitigation" (2005). *USF Tampa Graduate Theses and Dissertations.*
https://digitalcommons.usf.edu/etd/714

This Thesis is brought to you for free and open access by the USF Graduate Theses and Dissertations at Digital Commons @ University of South Florida. It has been accepted for inclusion in USF Tampa Graduate Theses and Dissertations by an authorized administrator of Digital Commons @ University of South Florida. For more information, please contact digitalcommons@usf.edu.

Partial Evaluation Based Triple Modular Redundancy For Single Event Upset Mitigation

by

Sujana Kakarla

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Engineering
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Srinivas Katkoori, Ph.D.
Nagarajan Ranganathan, Ph.D.
Soontae Kim, Ph.D.

Date of Approval:
March 24, 2005

Keywords: radiation, hardening, temporal, redundant gates, simulator

© Copyright 2005, Sujana Kakarla

DEDICATION

In memory of my loving grand parents.

ACKNOWLEDGEMENTS

My sincere thanks to Dr. Srinivas Katkoori, without whom, this work would not have been possible. I am grateful to him for assigning me this interesting task and helping me through out the project with his valuable suggestions.

I would acknowledge the real debt of gratitude that I owe to my parents and brother for always being there for me.

I also thank Dr. Ranganathan and Dr. Kim for being on the committee and giving me detailed review and helpful suggestions.

My special thanks to my friends in VCAPP group, for providing a wonderful working environment in the lab. I would also like to thank all my friends for their support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES iii

LIST OF FIGURES iv

ABSTRACT vi

CHAPTER 1 INTRODUCTION 1

1.1 Radiation Effects 2
1.1.1 Types of SEUs 5
1.2 CMOS Technology 9
1.3 FPGAs 9
1.4 Radiation Hardening by Design 11
1.5 Partial Evaluation 13
1.6 Temporal TMR 14
1.7 Partial Evaluation Based Triple Modular Redundancy 15
1.8 Results 17
1.9 Organization of the thesis 17

CHAPTER 2 BACKGROUND AND RELATED WORK 18

2.1 Radiation Effects 18
2.2 Radiation Hardening by Shielding 22
2.3 Radiation Hardening by Fabrication Techniques 23
2.4 Radiation Hardening by Design 25
2.4.1 System Level Design Hardening Techniques 25
2.4.1.1 Coding Techniques 25
2.4.1.2 Current Monitoring Techniques 28
2.4.2 Resistive or Capacitive Hardening 29
2.4.3 Circuit and Logic Design Techniques 30
2.4.3.1 Ratioing 31
2.4.3.2 Rockett Cell 31
2.4.4 Redundancy 32
2.4.4.1 Lockstep System 33
2.4.4.2 Triple Modular Redundancy 33

2.4.4.3 Dual Voting Double Redundancy 34
2.4.4.4 Selective Triple Modular Redundancy 34
2.5 Miscellaneous Techniques 35
2.6 Partial Evaluation 37
2.7 Temporal Latch 39
2.8 Voter Circuits 43
2.9 Summary 47

CHAPTER 3 PARTIAL EVALUATION REDUNDANCY 48
3.1 Definitions 48
3.2 Characterizing Input Environment 49
3.3 Implementation of the Partial Evaluation Based TMR Technique 50
3.4 Algorithms used in Partial Evaluation Based TMR 54
3.5 Advantages and Disadvantages 55
3.5.1 Advantages 55
3.5.2 Disadvantages 57
3.6 Illustrative Example 57
3.7 Summary 60

CHAPTER 4 EXPERIMENTAL RESULTS 62
4.1 Experimental Setup 62
4.2 Results 67

CHAPTER 5 CONCLUSIONS 77

BIBLIOGRAPHY 78

LIST OF TABLES

Table 1.1 Summary of Single Event Effects on Space Electronics 6

Table 2.1 Error Detection and Correction Codes [2] 26

Table 3.1 Computation of Output Probability 50

Table 4.1 Results for circuits with positive gains and rounding range as $0.0 < p < 0.2$ => logic ‘0’ and $0.9 < p < 1.0$ => logic ‘1’ 67

Table 4.2 Results for circuit with negative gains and rounding range as $0.0 < p < 0.2$ => logic ‘0’ and $0.9 < p < 1.0$ => logic ‘1’ 69

Table 4.3 Results for circuits with positive gains and rounding range as $0.0 < p < 0.3$ => logic ‘0’ and $0.8 < p < 1.0$ => logic ‘1’ 70

Table 4.4 Results for circuit with negative gains and rounding range as $0.0 < p < 0.3$ => logic ‘0’ and $0.8 < p < 1.0$ => logic ‘1’ 72

Table 4.5 Results for circuit with positive gains and rounding range as $0.0 < p < 0.4$ => logic ‘0’ and $0.7 < p < 1.0$ => logic ‘1’ 73

Table 4.6 Results for circuit with negative gains and rounding range as $0.0 < p < 0.4$ => logic ‘0’ and $0.7 < p < 1.0$ => logic ‘1’ 75

LIST OF FIGURES

Figure 1.1	Plot of Variation of Critical Transient Width With Feature Size [14]	3
Figure 1.2	Occurrence of SET Errors in Sequential Circuits	7
Figure 1.3	Temporal Relationship for Latching a Data SET as an Error	8
Figure 1.4	Temporal Relationship for Latching a Clock SET as an Error [14]	8
Figure 1.5	Power vs Throughput Graph for SET-Mitigation Techniques [35]	14
Figure 1.6	Flow of PTMR Technique	16
Figure 2.1	Configuration Memory [15]	21
Figure 2.2	Resistive Hardened CMOS SRAM Cell Design [36]	29
Figure 2.3	Hardening by Use of Ratioing	31
Figure 2.4	Rockett Cell [1]	32
Figure 2.5	Triple Redundant FPGA Inputs [18]	34
Figure 2.6	Dual Voting Double Redundancy [15]	35
Figure 2.7	Effect of Transients With Low Pulse Widths	40
Figure 2.8	Effect of Transients With High Pulse Widths	41
Figure 2.9	Spatially Redundant Latch	42
Figure 2.10	Temporally Redundant Latch	43
Figure 2.11	Majority Vote Circuit [15]	44
Figure 2.12	Majority Vote Circuit Using LUTs [15]	44

Figure 2.13 Majority Vote Circuit Using BUFTs [15] 45

Figure 2.14 Majority Vote Circuit Using LUT and FF [15] 46

Figure 2.15 Implementation of Module Redundancy and Mitigation On Single Chip [15] 46

Figure 3.1 Implementation of Redundancy Using Partially Evaluated Circuit 51

Figure 3.2 Evaluation of Temporal TMR 52

Figure 3.3 Overall Implementation of the Technique 52

Figure 3.4 Algorithm to Generate the Reduced Circuit 53

Figure 3.5 Algorithm to Identify if a Gate is Redundant 54

Figure 3.6 Algorithm That Describes the Overall Implementation of the PTMR Techinque 55

Figure 3.7 Comparison of Area/Delay Overhead For Different Redundancy Techniques 56

Figure 3.8 Original Circuit 57

Figure 3.9 The Rounded Values Are Propagated Over the Circuit 58

Figure 3.10 Reduced Circuit 59

Figure 3.11 Circuit That Computes Output From Partially Evaluated Circuit 59

Figure 3.12 Circuit That Computes Output From Temporal TMR Circuit 60

Figure 3.13 Circuit Showing The Overall Implementation 61

Figure 4.1 Experimental Flow 63

Figure 4.2 Resolution Function 64

Figure 4.3 Fault Insertion On Line “A” by SEU Simulator 65

Figure 4.4 Fault Insertion On Line “A” by SEU Simulator 65

Figure 4.5 Validation of the Technique 66

PARTIAL EVALUATION BASED TRIPLE MODULAR REDUNDANCY FOR SINGLE EVENT UPSET MITIGATION

Sujana Kakarla

ABSTRACT

We present a design technique, called partial evaluation triple modular redundancy for hardening combinational circuits against Single Event Upsets (SEU). The input environment is given in terms of signal probabilities of the lines. This is useful information to determine the redundant gates of the given circuit. The basic ideas of partial redundancy and temporal triple modular redundancy are used together to harden the circuit against SEUs. The concept of partial redundancy is used to eliminate the gates whose outputs can be determined in advance. This technique fails in cases when the actual inputs to the circuit are not in accordance to the rounded logic values. In such cases the technique of temporal TMR is used. However, there is some overhead in this process because of the voter circuits and the need to choose the outputs computed by partially evaluated circuit and circuit using temporal TMR. For testing the circuit exhaustively against SEUs, a fault insertion simulator is used. This simulator introduces errors in the circuits during simulation which represent SEUs. This technique of partial evaluation redundancy is thoroughly tested on MCNC'91 benchmarks using Cadence NCLaunch simulator. By employing this technique, in most of the cases we can reduce the area overhead of the hardened circuit when compared with the traditional Triple Modular Redundancy (TMR). The improvement in area is based on the total number of gates and the actual number of outputs. For circuits with large number of gates and less number of outputs, there is greater savings in area. In some cases, the area overhead because of the proposed technique is greater than the traditional TMR. This usually occurs in smaller circuits or in circuits with more number of outputs.

CHAPTER 1

INTRODUCTION

High performance, low power consumption, increased speed, and cost are the key factors in circuit design. One of the major obstacles for reliable space based computational systems is the occurrence of SEUs in electronic circuitry. Upset of the control has more serious system level consequences [1].

Radiation in space occurs because of the fusion process occurring in the sun which creates a constant stream of particles flowing through space. The earth's atmosphere helps in filtering the ionization radiation. Protons and heavy ions emitted by sun, galactic cosmic rays, and particles trapped in the earth's magnetic field are the main contributors to space radiation. There are different types and levels of radiation around the earth. At regions 500 km above the surface of earth, i.e., at Low Earth Orbit, radiation doses are the lowest. Only a few heavy ions penetrate the magnetic fields at this level. Van Allen belts increase the dosage levels at Polar Regions. More heavy ions are able to penetrate in these regions. At geosynchronous orbit, doses are still higher. Geomagnetic shielding causes interplanetary space to have the highest dosage levels. Continuous exposure to radiation causes degradation of the device. The level of degradation is based on the total dose and dose rate of irradiation. Radiation has long-term effects such as total ionizing dose and single particle effects such as single-event latchup and single-event upset [38].

1.1 Radiation Effects

The behavior of an electronic circuitry is different in outer space as compared to the normal environment because it is exposed to a flux of ionized particles [32]. The various effects of ionization are summarized below:

1. Penetration of ions through the space craft generates X-rays. These X-rays could cause the ionization of silicon and silicon dioxide layers. This results in temporary effects such as corruption of memory cell contents or permanent effects when the ionization triggers latchup in the device. Electron-hole pairs created by the electrons and x-rays are usually accumulated at power supply nodes [38].
2. Data in the device changes because of charge collection at a circuit node [38].
3. Ionization changes the characteristics of a device by shifting transistor thresholds [38].
4. The major effect is seen as a change in the contents of a MOS memory cell which occurs when the energy level of charged particle is high and it passes through the diffusion region of a susceptible node [32].
5. The other effects ionization could have are change in leakage current, charge trapping, and generation of interface states [32].

When a highly energized particle passes through a sensitive device, it loses energy and ionizes the material. This forms a dense track of electron-hole pairs. Stopping power, usually termed as Linear Energy Transfer (LET), is the rate at which the ion loses energy. Under the influence of electric field, the electron-hole pairs drift in opposite direction and are collected at the respective voltage sources. This produces a current transient [14]. Feature size plays a role here, in that the critical charge collected at a sensitive node which is able to produce an upset decreases as the square of the feature size [14]. Thus, decrease in feature size increases radiation tolerance. Oxide purity and thickness also determine the amount of degradation. The system function of a memory cell determines the effects a SEU would have on it.

Figure 1.1 shows the relation between feature size and critical transient pulse width. If the transient pulse width is smaller than the critical width, the transient is attenuated because of the inherent inertial delay of the gate. This causes the pulse to die out after it passes through some gates. However, if the pulse width is equal to or greater than the critical width, the transient propagates through the gate [14].

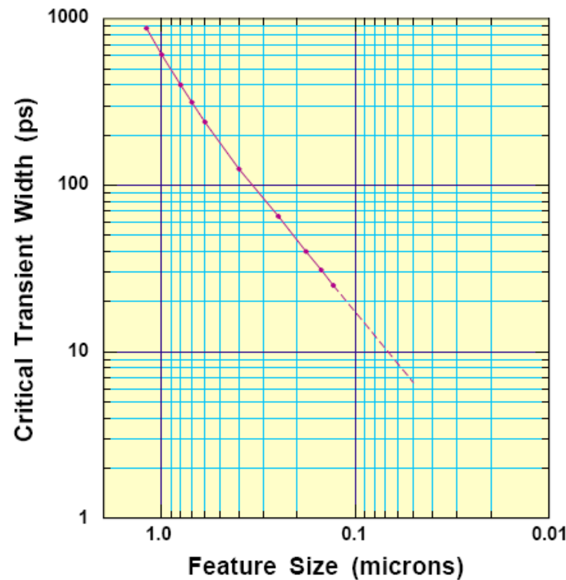

Figure 1.1 Plot of Variation of Critical Transient Width With Feature Size [14]

The main radiation effects in Microelectronics are:

1. Long Term Ionizing Radiation Effects (total dose)
2. Transient Ionizing Radiation Effects (dose rate)
3. Single Event Effects
    - (i) Single Event Upset (SEU)
    - (ii) Single Event Transient (SET)
    - (iii) Single Event Latchup (SEL)
    - (iv) Single Event Functional Interrupt (SEFI)
    - (v) Single Event Gate Rupture (SEGR)
    - (vi) Single Event Burnout (SEB)
4. Displacement Damage

Total Ionizing Dose (TID): High energy protons and electrons cause TID. High-energy ionization radiation generates electron-hole pairs within the oxide of a MOS device which cause charge buildup. Charge buildup has many effects such as change of threshold voltage, increase in leakage current and change in timing of the MOS transistors. This could lead to functional failure of the device [27]. Leakage currents are generated at the edge of MOS transistors and neighboring N-type diffusions.

Single Event Effects result from a single energetic particle.

Single-Event Upset (SEU): Change in the state of a device or transient induced by a heavy ionizing particle such as a cosmic ray or proton. SEUs alter the logic state of a static memory element and cause transient pulses in combinational logic paths. These errors can be corrected by resetting or rewriting of the device. Hence they are termed as soft errors. These soft errors occur due to the change in state of a digital memory element because of the ionizing particle. SEUs occur when a high energy particle hits a storage node and generates a strong ionization tract [3]. Charge transfer from one node to another occurs as the ionizing particle passes through the device. This lowers the voltage of a memory cell and changes its internal state. The stored logic state is reversed if the collected charge on node is larger than the critical charge [3]. Digital, analog, and optical components are prone to SEUs.

Single event latchup (SEL) is a condition that causes loss of device functionality due to a single-event induced current state. Latchup within a CMOS device occurs because of a single charged particle. With sufficient energy the charged particle triggers the parasitic npn-pnp circuit found within CMOS circuits. Because of latchup, high currents flow through the parasitic bi-polar transistors and destroy the device. Device design characteristics such as material resistivity, device geometry, and layout and contact characteristics affect the SEL resistance of a device [27]. SELs are hard errors, and are potentially destructive (*i.e.*, may cause permanent damage). An SEL is cleared by a power off-on reset or by completely removing power to the device.

Single event burnout (SEB) is a condition that can cause device destruction due to a high current state in a power transistor. SEB causes the device to fail permanently because of failures of power MOSFET transistors in high power applications.

Single-event gate rupture (SEGR), which is the formation of a conducting path (*i.e.*, localized dielectric breakdown) in the gate oxide resulting in a destructive burnout.

Single Event Transient: Effects (e.g., current spikes in operational amplifiers) of short time duration that may lead to other effects downstream of the affected site that are longer in duration. SETs cause soft-error in the user data when it is registered at the flip flop inputs [39].

Displacement damage is due to nuclear interactions and causes lattice defects. It is mainly a long-term non-ionizing damage caused by protons, electrons and neutrons. Collision between the incoming particle and a lattice atom subsequently displaces the atom from its original lattice position.

1.1.1 Types of SEUs

1. Single bit errors: A single ion passage causes a single bit to change its state.
2. Multiple bit errors: If the single ion causes multiple bits to flip in adjacent cells, then a multiple bit upset occurs. Multiple bit upsets require that the adjacent cells be mapped to the same logical address.
3. Control errors: SEUs causing operational difficulties in the device such as reading or writing to the incorrect address or the occurrence of a functional halt [13].

System level effects of SEUs can be categorized as those that affect data responses and those that effect control of the device [2]. The effect of SEU transients on analog devices is slightly different from their effect on digital systems. Transients in analog devices propagate to the digital electronics of the surrounding circuitry. Specific circuit designs determine the effect these transients would have on the system. The definition of an analog SEU phenomenon is specific to the interface circuitry surrounding

the radiation-sensitive device. SEUs for the conventional analog-to-digital converters can be categorized as noise, offset, and control errors. Although noise and offset errors do not disturb the system performance, control errors affect device operation, and hinder system performance [2]. Table 1.1 summarizes the list of errors occurring in space electronics [32].

Table 1.1 Summary of Single Event Effects on Space Electronics

Type of error	Description
SEU in the configuration memory	A bit flip in the configuration memory caused by a single particle strike, Neutron or alpha.
SEFI in the target circuit	A permanent mismatch of the output of the target circuit. It is created by a SEU in the configuration memory that alters the Look-up Tables (LUT) or the routing of signals in the target circuit.
Configuration circuitry failure	A failure in the controlling circuitry of the FPGA. Configuration and read back operations fail.
Latchup	The activation of a parasitic structure in the silicon by a single neutron strike. The main consequences of latchup effect are an increase of the current consumption and failures in the target circuit, the configuration memory or the controlling circuitry of the FPGA.
Hard error	A permanent failure in the FPGA that cannot be recovered after switching the beam off, switching the power off/on, and reconfiguration.

Definition: SEU Mitigation: The process of applying design techniques to strengthen the functional integrity of the user design and protect it from the effect of any Single Event Upset [29].

Various techniques are implemented to evaluate the rate of functional bit failure. Systems using ASIC technology use static upset rate while those using Virtex series FPGAs define a dynamic upset rate [29].

SEEs affect sequential circuits as well. Figure 1.2 represents the circuit topology of sequential circuits. Data from latch U1 is released to the combinatorial logic on a clock edge. Before the next clock edge, the output of the combinatorial logic reaches the latch U2. The latch stores the data present at its input, at this clock edge. Consider a heavy ion striking within the combinatorial logic. For fast logic, the SET appears at the latch U2. The SET is propagated through the circuit based on the arrival time of the SET and latching edge of the clock. Figure 1.3 illustrates this. Suppose the actual data to the circuit is low and a positive SET appears at the input of the latch. If the transient is high for the period between setup time before the clock edge to a hold time after the clock edge, then the transient is interpreted as data and stored in the latch. Figure 1.3 shows four instances of time at which the SET can arrive. Conditions (a) and (d) satisfy a non-latching condition. Condition (b) represents the earliest arrival time for latching condition and condition (c) represents the latest time. Similar errors occur from transients appearing on the clock line as shown in Figure 1.4.

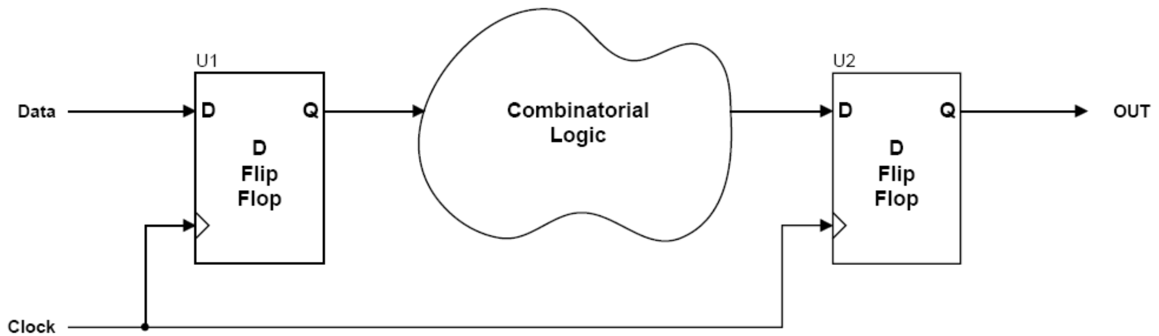

Figure 1.2 Occurrence of SET Errors in Sequential Circuits

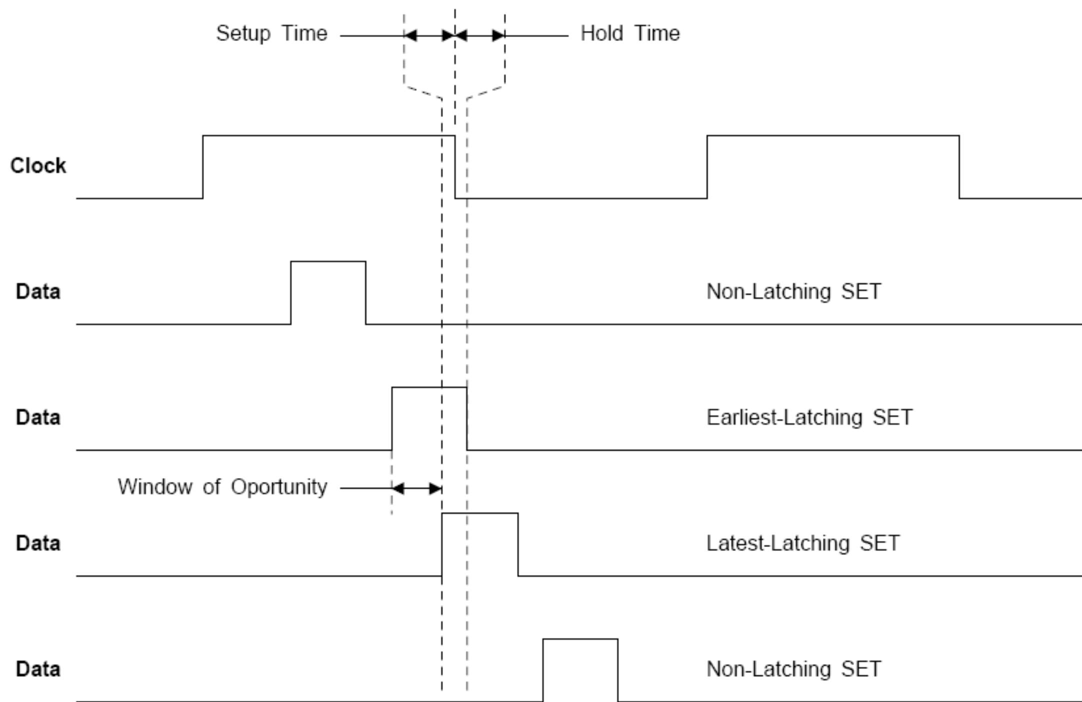

Figure 1.3 Temporal Relationship for Latching a Data SET as an Error

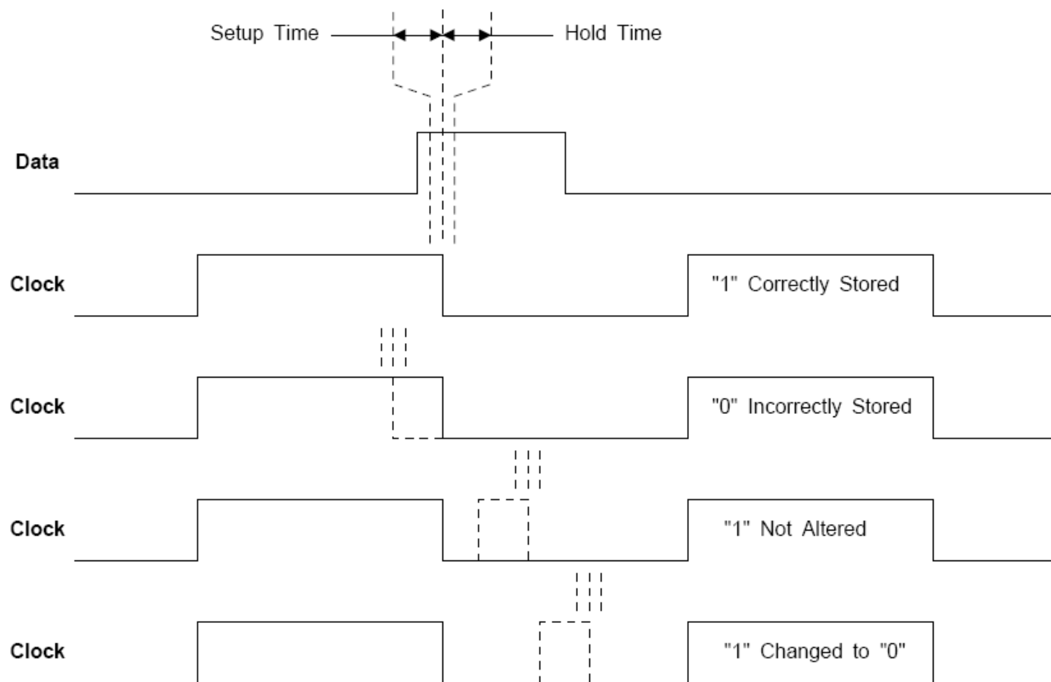

Figure 1.4 Temporal Relationship for Latching a Clock SET as an Error [14]

Upsets occur in sequential circuits when the clock is low and the latch is in hold state. Thus, latch SEU rates do not depend on clock frequency. SETs in combinatorial logic are stored if they occur at clock edge. Thus the effect of SETs depends linearly on the frequency.

1.2 CMOS Technology

Most of the circuits in space applications use CMOS processes because of their desirable features such as high integration density, low power dissipation, and high noise immunity [7]. However, CMOS circuits are prone to the following failures:

1. Increase in Standby supply current.
2. Degradation of input levels and internal noise immunity. This introduces functional failures.
3. Increase of rise time and decrease of fall time. This causes change in switching and dynamic parameters [33].

CMOS static RAMs have very high dynamic power consumption. There is negligible static leakage current [3].

CMOS devices are susceptible to single event effects (SEE). This includes cell error events such as single event upset (SEU), and destructive conditions such as single event latchup (SEL). SEE results may vary based on temperatures [13].

1.3 FPGAs

FPGAs consist of devices comprised of an array of cells that can implement a variety of logic functions plus some interconnection network. Configuration information is downloaded onto the FPGA to set up the cells and interconnection network to realize a specific circuit [17]. FPGAs have resulted in higher speed, lower core voltages, improved integration, and lower power consumption [20]. They are used in applications where

speed, efficiency, and the ability to program hardware to perform any user-specified operation are important and cannot be achieved by the traditional programmable processors. This is possible because of the ability to customize the data path within an FPGA to an application-specific computation [16]. FPGAs find their importance when production volumes are too low to develop an ASIC [24]. The development of SRAM cells enables FPGAs to be reprogrammed during design or in space [38]. FPGAs differ from ASICs in that they can be configured after the space-craft launch. This enables reusing of FPGA resources for multiple instruments [16]. The main advantages of using FPGAs in space circuits are:

1. High flexibility in achieving multiple requirements such as cost, performance, and turnaround time.
2. In-flight reconfiguration SRAM-based FPGAs have the ability to change on-site the implemented function and hence are very convenient for space-based applications.
3. Decrease in number of devices required reduces weight.
4. The decrease in number of solder connections improves reliability.
5. Increased flexibility to make design changes after board layout is complete [40]. Hence changing or updating of hardware is easy.

However, there are some problems associated with using FPGAs in space environment:

1. FPGA-based applications are sensitive to heavy ion and proton induced SEUs (internal flip-flops). User design flip-flops, FPGA configuration bit stream, FPGA registers, latches, and internal state are affected by SEUs.
2. SRAM-based FPGA might have their configuration altered by radiation. The function implemented by a device can be permanently affected by affecting the configuration memory of SRAM-based FPGAs by SEUs. The general solutions to this problem are Partial or total reconfiguration, and TMR.

3. FPGA Logic may be sensitive to transient errors. Smart clocking strategies can reduce this problem [40].

The basic concepts underlying the radiation tolerant Virtex FPGAs are as follows:

1. Re-configurable Logic Devices: Logic devices that can be customized more than one time are called Re-configurable Logic devices. These devices use remote hardware changes and functional evolution for fast SEU Detection and correction [15].

2. SEU Protection Design Techniques: These techniques employ SEU detection and correction strategies which do full design verification in a very short time and SEU correction without any functional interrupt. SEU mitigation is achieved by:
    1. SEU Resistant Mitigation circuit
    2. Module and logic node redundancy and mitigation
    3. Logic partitioning for mitigation
    4. Dual and triple device redundancy and mitigation [15].

Errors in an FPGA can be corrected by fixing the incorrect design and reconfiguring the FPGA with an updated configuration bit stream. Also, custom circuit designs can be created to avoid FPGA resources that have failed during the course of the spacecraft mission [16].

1.4 Radiation Hardening by Design

In recent years, high performance ICs for radiation environments are being designed because of the rapid pace of design innovation for commercial IC applications. Decrease in feature size of the device helps in the increase in speed and density of IC designs [38]. Creation of autonomous spacecraft which rely on information processing on-board the vehicle made the radiation hardening of circuits a more crucial aspect [38]. The main goal of design hardening techniques is to manufacture SEU-immune circuits using standard CMOS processing, with no additional masks. The basic idea is to provide

memory elements with feedback [9]. Hardening commercial CMOS technologies has many advantages such as low power, low cost, higher speed, and higher density. This also facilitates the use of more advanced, deep sub-micron technologies [38].

Radiation hardening means the extra protective package that is provided to the chips to make them more resistant to the ionizing radiation [38]. Hardness of a circuit can be achieved in three ways:

1. Shielding the circuit.
2. Process hardness: Altering the methods of manufacturing the chips.
3. Design or layout techniques: Changing the designs of the chip. The principle of radiation hardening by design is the minimization of radiation impact by the use of layout techniques.

Design of radiation-resistant chip is a complicated process and involves deep understanding of the various concepts such as:

1. Device physics of the transistors and other circuit elements.
2. Effect of various kinds of radiation on the circuit elements at the atomic level.
3. Effect of fabrication processes (such as patterning, oxide deposition, ion implantation, etc.,) on device sensitivity, when exposed to different radiation sources [41].

The conventional design and manufacturing techniques employed to speed-up the circuit or lower the price could lead to adverse effects when exposed to ionizing radiation. Component radiation hardness depends on the orbit and time frame of the mission. Generally, two complementary approaches are used in the design of radiation-hardening circuits. They are:

1. Design radiation-hard characteristics into the chip.
2. Use special techniques that mitigate radiation effects at the processing phase [38].

Radiation hardening makes satellite design flexible. The development of radiation hardened SRAM cells enables the use of field programmable gate arrays which can be reprogrammed during design or in space. This simplifies circuitry, reduces cost, and eases space-based usage [38]. To design circuits which are radiation hardened a set of hardened cells suitable for gate array or full custom designs are developed. Elements which are not radiation tolerant are replaced with the hardened elements. A general way of hardening the circuits is to make changes in technology parameters such as varying the Linear Energy Transfer (LET) threshold, power supply, and on-chip detection mechanisms such as use of parity bit checks to provide error notices (in which case SEE are treated at system level by interruption handling) [33].

Radiation hardened versions have improved tolerance against long term Total Dose degradation, and Heavy ion induced effects such as latch-up. These versions for deep sub-micron technologies offer solutions to applications above 100 Krad (Si) range total dose requirements [33].

1.5 Partial Evaluation

Partial Evaluation is an optimization technique commonly used in software applications to increase the efficiency of the code. Optimization of a design by having the knowledge of its properties and structure is generally termed as partial evaluation. The process of specialization is done systematically, we start from a general circuit and some data known at run-time and then using this data, transform the general circuit into specialized circuit [17]. The basic principle involved in this technique is that the known arguments to function calls are propagated throughout the definition of a function, yielding a new specialized function [24]. Thus partial evaluation can be thought of as specializing a program to its static inputs [17]. The specialized program is specific to a particular application and hence works on fewer cases. This program is more efficient because some computation has already been done during specialization. The program which performs partial evaluation is termed as partial evaluator. The resultant program obtained is called residual program [26].

Logical inferences by unfolding predicate calls, propagation of instantiated values through the program, and evaluation of built-in predicates are the main techniques implemented by the partial evaluator [26]. Operations involving constant operands are eligible for reduction based on partial evaluation [28]. A very simple form of partial evaluation corresponds to run-time constant propagation. Since propagated values are known at run-time only, we call this, dynamic synthesis of circuits [17]. The atomically synthesized circuit is determined at run-time, and the dynamic input data determines the circuit. Partial evaluation can be implemented 'in place [24].'

1.6 Temporal TMR

Temporal TMR has the main advantage of low power requirements. There are two ways of implementing Temporal TMR. In the first kind, three identical threads of computation followed by a voting circuit is used. This means that same data value is applied on each of the thread at three successive clock cycles. The next kind of implementation uses the register filtering technique. This uses multiple registers for each combinational logic output. Each register is clocked by a separate delayed clock. Voting of the resulting sampled data values is done to determine the correct output [35].
The graph in Figure 1.5 shows the throughput versus power graph for the different techniques. Singlet is the unhardened case [35].

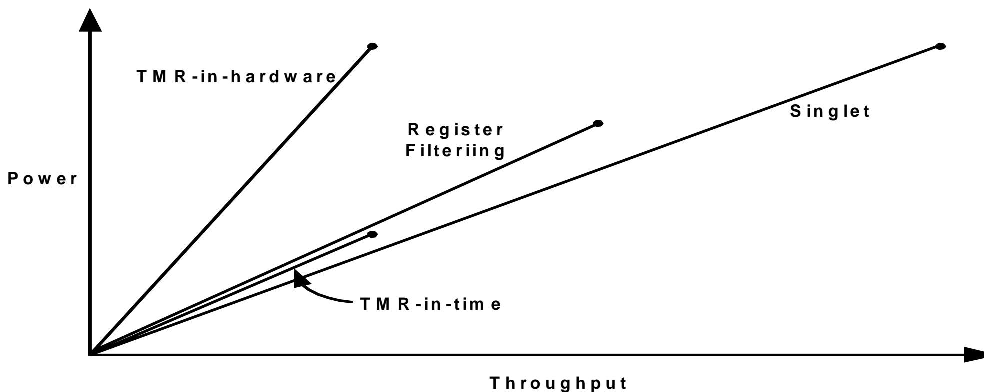

Figure 1.5 Power vs Throughput Graph for SET-Mitigation Techniques [35]

The main disadvantage of the TMR-in-time technique is that the throughput is reduced to 1/3rd of the original value. However, the power requirements of this technique are low when compared to the standard TMR technique. It is because, in the absence of radiation-induced transients, data switching occurs only at the beginning of the first clock cycle and none during the two successive clock cycles [35].

Based on the delay between clocks, register filtering exhibits a variable throughput. If the delay is zero, the throughput, and power requirements are equal to that of an unhardened design. Advantage of this technique is that the separation of clocks can be controlled by programming to determine the maximum transient duration that can be filtered. This determines the energy of radiation that can be tolerated. Hence the chip can be hardened to the desired degree by programming [35].

1.7 **Partial Evaluation Based Triple Modular Redundancy**

In the proposed approach, the basic ideas of partial evaluation and triple modular redundancy are used together to device a scheme for designing radiation tolerant circuits. However, the behavior of input environment should be known in advance to implement this technique. The input environment is given in terms of signal probabilities of the lines. Knowing the signal probability, these values are propagated to the output of the circuit. Logic value of signals with probabilities within the range of 0.0 to 0.2 is set to ‘0’ and for those in the range of 0.9 to 1.0, it is set to ‘1’. These integer values are then propagated instead of the original probabilities. If any input to a logic gate is its controlling value, then the gate can be eliminated. For instance, the controlling value of “and” and “nand” gates is a logic ‘0’ and for “or” and “nor” gates it is a logic ‘1’. By eliminating all the redundant gates, a reduced circuit which is functionally equivalent to the original circuit is obtained. This reduced circuit is then duplicated. To get the final correct output, the outputs generated by the original circuit and the two duplicated circuits are voted. Voting is done by a majority voter. The technique of partial evaluation fails in cases where the actual inputs to the circuit are against the rounded logic values. In such cases the output is evaluated by the technique of Temporal TMR. A multiplexer is

used to choose the outputs determined from partially evaluated circuit and from Temporal TMR. The overall flow of the Partial Evaluation based TMR technique is shown in Figure 1.6. In the whole process, we assume that the majority voter circuit and the multiplexer are radiation hardened. Efficiency of the circuit depends on the actual inputs to the circuit and signal probabilities of the lines.

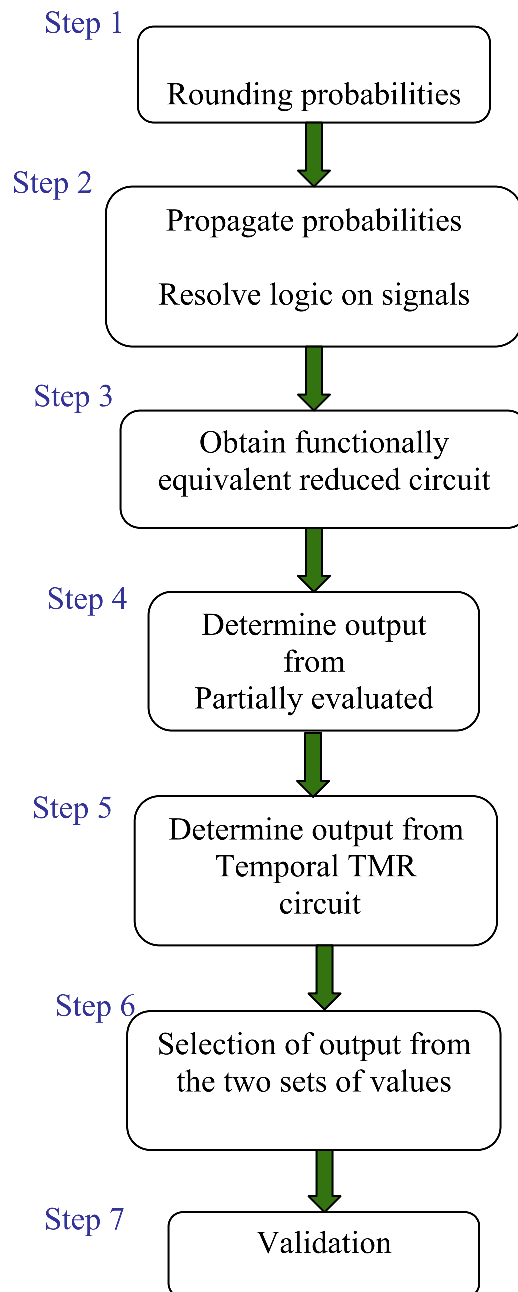

Figure 1.6 Flow of PTMR Technique

1.8 Results

By implementing the partial evaluation based redundancy technique, it was found that a good amount of area savings is achieved. The area savings are high for circuits with large number of gates and less number of outputs. In cases where there are less number of gates and relatively more number of outputs, the area savings are very less. In the worst case, it could so happen that overhead involved in the implementation of technique is greater than area savings. This occurs in circuits where the number of gates that can be eliminated is more than the overhead involved in the process.

1.9 Organization of the Thesis

The rest of thesis is organized as follows:

Chapter 2 deals with the background, and related work. It describes in detail the various effects of radiation, techniques devised to provide radiation hardness, work done in partial evaluation, and the various kinds of majority voters developed.

Chapter 3 describes in detail the partial evaluation based triple modular redundancy.

Chapter 4 discusses about the experimental setup, the results obtained, and analysis of the results.

Chapter 5 gives the conclusions of the thesis.

CHAPTER 2

BACKGROUND AND RELATED WORK

One of the major concerns for reliable space based computation is the occurrence of SEUs. This chapter deals with the various radiation effects on space electronics and the techniques employed to mitigate these errors. The mitigation techniques can be classified into three categories, namely radiation hardening by shielding, radiation hardening by fabrication, and radiation hardening by design. Each of these techniques is explained in detail in the following sections. The concept of partial evaluation, which is an efficient optimization technique in software, is explained and its application in hardware is presented. The selection of proper majority voter is a key point in implementing techniques using redundancy. The voter circuit should be highly radiation hardened to ensure that no errors are introduced because of the voter circuit. Thus different kinds of voter circuits are explained.

2.1 Radiation Effects

The earth is surrounded by particle charged belts called Van Allen Belts. These belts mainly arise from the earth's magnetosphere-field. They are:

1. Proton trapped belt which extends from 400 to 900 km above the earth's surface. This belt consists of electrons and protons.
2. Electron trapped belt which extends up to 56,000 km. This belt is almost entirely composed of electrons [33].

However the height of these belts is not fixed. They could vary depending on the concentration of electrons and protons. For instance, when solar flares occur, there is intense burst in the number of high energy protons and heavy ions. This increases the Van Allen belt by a factor of 1000. Similarly, Galactic cosmic rays are composed of heavy ions in varied abundance. The heavy radiation is due to the influence of increase in the number of high-energy protons from heavy solar flares formed during the excursion of satellite [33].

The generic circuits work without any faults in their intended environments. However, when they experience harsh conditions in outer space such as the ionizing radiation, they do not work properly. Electrical parameters of the chip change when the ionizing radiation strikes the transistors in the circuit. This leads to the generation or flow of extra electrical currents which alter the operation of circuit. Also, poor design techniques cause chemical weaknesses in the atomic structure of the transistors when they are exposed to ionizing radiation [41]. Cosmic rays are high energy ions, protons, and neutrons. Interaction of ICs with cosmic rays leads to SEE [33].

Conversion of primary radiation causes the generation of electromagnetic rays when electrons and protons interact with any kind of material that they encounter. Heavy particles are generated by nuclear reaction when protons directly interact with the material. These heavy particles can induce volume effects [33]. Electromagnetic rays and electrons result in electron-hole generation which creates ionization in SiO_2. Basic device characteristics such as threshold voltage and mobility change due to combination effects and mobility differences. Increase in dose causes sub-threshold currents to increase and affect NMOS structures by changing parameters. These currents are induced by parasitic structures resulting from ionization [41].

Electron-hole pairs are created in device nodes and diffusions because of ionization. Latch-up effect causes the activation of parasitic SCR structures when the electron-hole pairs reach the P or N well. Thermal destruction of the component occurs because of high currents. Soft errors are created when the current pulse appears in a depleted zone (drain of the transistor in the off-state mode) because of collection

phenomenon. When CMOS technology is being used, NMOS is more sensitive than PMOS. Geometry of the device plays a role in that the electrical charge required for bit flip decreases with scaling down [33].

Bipolar technologies are mainly affected by the degradation mechanisms of neutrons which involve atomic dislocation and nuclear displacement and reduction of minority carrier lifetime and mobility. Electromagnetic pulses (EMP) is another factor against which the circuits must be protected. High currents and voltages are generated in conductors and electronics because of EMP. Box and hardware shielding are the common precautionary measures for this kind of radiation [33].

Configuration memory defines the function of logic resources and interconnections of these resources [4]. Configuration bit stream determines the function of device. The design function can be changed by making changes to the bit stream. This has the advantage of adaptability. However it is this property that makes the device susceptible to SEUs [19]. 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 [4]. Static upsets in configuration memory do not affect functionality. Upsets need to be corrected only to ensure that errors do not accumulate [29]. Virtex Configuration Memory is composed of static latch memory cells as shown in Figure 2.1. It is divided into frames and each frame is uniquely addressable. Design functionality for Static RAM based Programmable Logic Devices is defined by configuration SRAM contents [15]. The original behavior of a SRAM-based FPGA could be changed by the mapped design, when a flux of highly energized particles hits its surface. Thus any 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 SRAM [18]. Single Event Upsets alter design functionality which leads to transient upsets that induce undesired logical conditions [15]. Upsets in configuration memory cause a local high current because of driver contention when two inverter outputs of different states get connected. Errors in the configuration memory can be corrected with the process of non-intrusive scrubbing. 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 [20]. Memory cells are anticipated to upset at a rate of 0.13 upsets/hour (3.2 upsets/day) in a normal sun environment and upset at a rate of 4.2 upsets/hour during the peak upset rate [16].

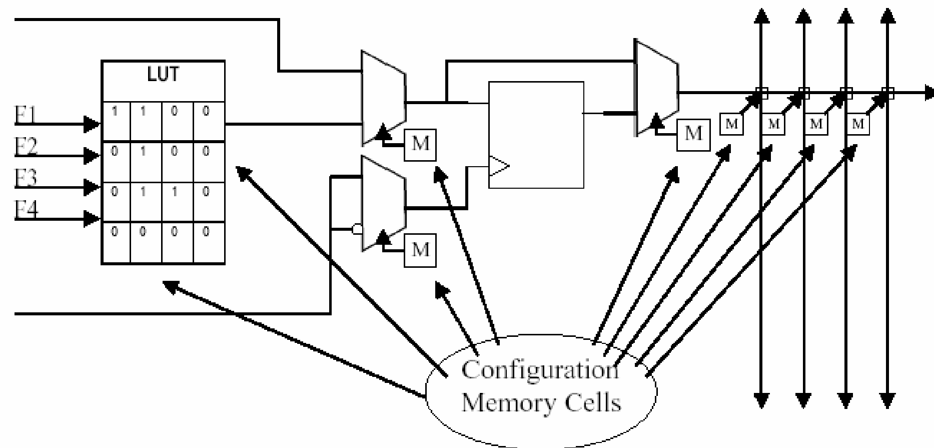

Figure 2. 1 Configuration Memory [15]

Constant ‘0’ and ‘1’ logic values in Virtex FPGA designs are usually generated by half-latches. This avoids the use of expensive logic resources, such as look up tables. These half-latches are neither initialized nor controlled with programming data. Thus reading back of the device’s programming data does not help in detecting half-latch inversions. Other techniques such as updating the FPGAs configuration memory or partial configuration are also ineffective. Thus SEU effects in half-latches are not easily detectable and correctable [4]. The SEU mitigation techniques implemented for half-latches are discussed in [4].

SEUs affect software systems as well. Data and code of the application are affected by faults. SEUs may cause information corruption, leading to a change in program flow or causing a program to execute an infinite loop. Transient faults occur because of radiations, electromagnetic interference, and power glitches [21]. Pure software methods are required for control flow error detection in cases where the hardware design is fixed and cannot be changed [22]. A correct control flow is a fundamental requirement for correct execution of computer programs. Control flow

errors pose threat to the dependability of computer systems [21]. The use of software techniques to detect and tolerate faults in the hardware is termed as Software Implemented Hardware Fault Tolerance. This has the advantage of improving the availability of the system without introducing any hardware overhead [22].

Every circuit has some amount of inherent tolerance to mild radiation. A radiation tolerant IC exhibits some degree of radiation survivability. However a radiation hardened circuit is specifically designed to withstand certain radiation levels. Radiation hardness is ensured by using Radiation Hardness Assured (RHA) devices for electronic circuitry. These devices are process monitored, designed, and layout controlled [42]. Hardness of a circuit is a measure of the total dose of radiation to which an IC can be subjected before critical parameters cross a predefined threshold [31]. Various hardening techniques have been proposed.

The main concepts of SEU correction techniques are:

1. Re-configuration: This is a widely used technique for traditional FPGAs. It is capable of repairing all static upsets. However, there is a momentary loss of service.
2. Partial Configuration: This is used mainly for Virtex FPGAs only. The main advantage of this technique is that it can repair single upsets in individual frames with no loss of service and functional disruptions.
3. Partial Configuration Cycles: These act on single frames [15].

2.2 Radiation Hardening by Shielding

A simple technique of providing radiation hardness is shielding the circuits made from standard components by metals such as lead which attenuate radiation and electromagnetic pulses. It may seem obvious that increasing the thickness of the package increases the radiation tolerance. This however is not practically true, as there is a

limitation on the attenuation of the signals because of shielding [42]. Also, the increased packaging slows down the high-energy particles and gets charged resulting in TID [36].

2.3 Radiation Hardening by Fabrication Techniques

A few general chip fabrication techniques employed to make radiation hardened chips:

1. Change the method of manufacturing transistors in the circuit: The unwanted charge which changes the operating characteristics of the devices is accumulated in the thin layers of oxide that are used to form the working and insulating parts of the devices and in the regions between the transistors. Thus reducing the thickness of these layers without compromising reliability decreases the sensitivity of transistors to ionizing radiation because of the limitation of unwanted charge. However the design of very thin oxide layers calls for high perfection, which would otherwise cause an electrical short. Additional processing steps after growing gate oxide layer onto the substrate must be carried out at lower temperatures, because higher temperatures would alter the gate oxide's atomic structure. This leads to inefficient radiation resistance [41].

2. Change the way that the transistors are combined to form working circuits: Parameters affecting the radiation resistance of the circuit could be varied and set such that immunity of the circuit increases. For instance, increasing the width of interconnections helps in easy handling of radiation-induced currents which makes the circuit more radiation tolerant [41].

3. Another technique is implementation of certain rules such as stacking fewer transistors in logic gates, circuit redundancy, and use of radiation tolerant parts [42].

4. Isolating a device form surrounding components ensures that charged ions cannot travel far in the components. This eliminates the possibility of latchup and SEU. Four

schemes are implemented based on this principle to make radiation-hardened devices. They are:

1. Junction Isolation (JI): This method is used for CMOS, and other unhardened bipolar designs. It is an electrical method which isolates on-chip components by reverse biasing the junction. However, this technique is not efficient for circuits exposed to very high radiation levels as they would be susceptible to latchup due to their parasitic PNPN SCR structure.

2. Dielectric Isolation (DI): Component isolation is achieved by thermally growing thick layer of silicon dioxide between adjacent devices. An oxidation mask is used to grow oxide on wafer only in chosen places. Dielectric isolation is a better choice for stringent radiation hardness applications.

3. Silicon-on-Sapphire (SOS): SOS is a more complex form of dielectric isolation. A single-crystalline silicon film is grown over a sapphire substrate which is a dielectric that has tolerance to radiation and protects the device from transient, neutron, and single event effects. A bipolar or FET transistor is made by doping silicon. In SOS, active devices can be packaged closer together, as the transistors are built on an insulating substrate because of which leakage current cannot flow between the devices. Latchup cannot occur as there are no parasitic transistors and capacitances.

4. Silicon-on-Insulator (SOI): SOI technology is similar to the process used for SOS devices, except for the substrate used. Silicon-on-Insulator devices can take several forms; one common technique is the SIMOX which is Separation by Implanted Oxygen. Heavy concentration of oxygen is deposited below the wafer's surface by high current ion implantation system. The wafer is then heated to form SiO_2 from Oxygen and also to anneal the damage caused by implant. This leaves a thin, high-quality layer of silicon on top of an insulating layer of SiO_2 which is used for device fabrication. Complete isolation of device is achieved by replacing the silicon between active transistor areas with oxide.

This dielectric-isolation plane enables increased circuit speeds, and radiation hardness [42].

To make the integrated circuits inherently radiation hardened, significant design changes are a better option than chip fabrication techniques [41].

2.4 Radiation Hardening by Design

Many techniques use hardware redundancy to reduce the probability of failure. The design hardening techniques can be categorized as:

1. System level design hardening techniques
2. Resistive or capacitive hardening techniques.
3. Circuit design / Logic design techniques [8].

2.4.1 System Level Design Hardening Techniques

System level design hardening solutions include coding techniques and current monitoring techniques. Coding techniques for error detection and correction can be adopted in high capacity memory arrays [7].

2.4.1.1 Coding Techniques

Use data word coding techniques and additional error detecting and correcting (EDAC) circuitry. Table 2.1 gives the different EDAC codes that are commonly used. The EDAC processor corrects all single bit errors by periodically scrubbing the entire memory [3]. The technique of SEU scrubbing requires less overhead as all the data frames are reloaded at a chosen interval. This avoids the use of processes such as readback detection, and data verification operations. Scrub rate is fixed such that any

SEU on the configuration memory is fixed before the next occurs. Time between the SEU occurrence and its subsequent correction should be reduced [29]. Maximum error latency is defined by the time interval between two successive accesses of the same memory word. Applications involving large memory systems have long error latencies which increases the probability of having multiple upsets on a single word. This makes the EDAC codes ineffective [3]. EDAC technique is implemented on memory systems and uses checksum-based technique to detect and correct errors on the same processing cycle. System availability is improved by reducing transient error recovery latency. This technique has the advantages of lower size, weight, power, cost, and improved reliability. However, it can be applied only to matrix multiply functions [23].

Table 2.1 Error Detection and Correction Codes [2]

Types of Error Detection and Correction codes (EDAC codes)	Capability of the EDAC code
Parity	Single bit error detect
CRC code	Detects if any error occurred in a given data structure
Hamming code	Single bit correct, double bit detect
RS code	Corrects consecutive and multiple bytes in error
Convolution encoding	Corrects isolated burst noise in communication stream
Overlying protocol	Specific to each system implementation

In the self-checking scheme, computations are performed on data belonging to two independent memories. This is to ensure that the errors are statically independent as a

common error cannot be detected by the self-checking scheme. This technique has the advantage of 100% coverage of virtually any source of error. However, it can only detect an error, but cannot correct it. It also has the disadvantages of higher cost, weight, power, size, and low reliability [23].

1. Use of parity checks: A single bit used to determine if the number of logic '1's in the data structure are even or odd is termed as parity. Thus, if odd number of errors occurs in the structure then they can be detected by parity. This technique however cannot be used for mitigation.

2. Cyclic-redundancy check (CRC) coding: This scheme performs modulo-two arithmetic operations on a given data stream by considering N data bits as N-1 order polynomial. The result is again interpreted as a polynomial and is the CRC character that could be appended to the data structure. For decoding, the generating polynomial divides a bit structure consisting of data and CRC bits.

3. Hamming code: This is a block error encoding scheme that encodes entire block of data with a check code. This scheme can be used for single bit correction but double bit detection. The position of single error is determined by the syndrome represented by Q-digit word when the parity-check matrix generates Q check bits. This is a commonly used scheme and is suitable for systems with low probabilities of multiple errors in a single data structure.

4. Reed-Solomon coding: This is a block error-correcting coding scheme which groups the check bits into separate words and adds them at the end of the data structure. Multiple consecutive errors in a data structure can be detected and corrected. A single chip can implement this R-S scheme.

5. Convolutional encoding: Convolutional encoding is a process of adding redundancy to a signal stream. This scheme is used to detect and correct multiple bit errors. It interleaves the check bits into the actual data stream and provides very good

immunity for mitigating isolated burst noise. The scheme is highly suitable for communication systems.

6. System-level protocol: Errors in this method are detected using parity checks and detection of a non-valid Manchester encoding of data. In case an error is detected, error correction is done via retransmission. The system-level protocol retransmits the transaction for a maximum of three times [2].

2.4.1.2 Current Monitoring Techniques

SEU detection and correction can be achieved by using static RAM architectures which employ transient current sensing circuits. In time-critical applications where reduction in error latency is a key requirement, current monitoring techniques are used [7]. A new technique is devised which uses CMOS static RAM and implements error detection based on current monitoring. Small fluctuations in power supply current are noticed when a heavy ion strikes and accumulates upsets. Abnormal currents produced by upsets in RAM columns are detected by the built-in-current-sensors (BICS). The supply line of each memory column is provided with a BISC. An internal latch is set when an SEU error occurs in a memory column. Error correction sequence is started immediately by a logic signal output, thus ensuring zero fault latency. The affected word in the column is detected by a parity bit per word. This parity bit also helps in error correction. Correction is performed only if after reading the memory words an erroneous parity is discovered. The main advantage of this scheme is that upset detection is asynchronous, fully independent and concurrent to normal memory operation. A proper determination of BICS detection threshold is very important. The BICS must be insensitive to the transient currents of active read or write cycles. They should not detect small transient currents induced by radiation which do not generate upsets. Estimation of storage node sensitivity to upsets helps in avoiding false alarms. Slow rate of false alarms pose no problem [8].

2.4.2 Resistive or Capacitive Hardening

Resistive hardening involves the use of passive, polysilicon intracell decoupling resistors in the cross-coupling segments of each SRAM cell. The cell identifies an upset-causing voltage transient by using decoupling resistors that slow the regenerative feedback response of the cell. SEU hardness is provided by gated resistors which are actively clocked, high resistance polysilicon resistors. They are built by two layers of polysilicon separated by a thin layer of thermal oxide. The high resistance of these gated resistors protects the stored cell data from SEUs. Figure 2.2 illustrates a resistively hardened CMOS SRAM cell schematic. Resistors R1 and R2 are the intra-cell decoupling resistors that improve the SEU hardness of the CMOS SRAM cell design.

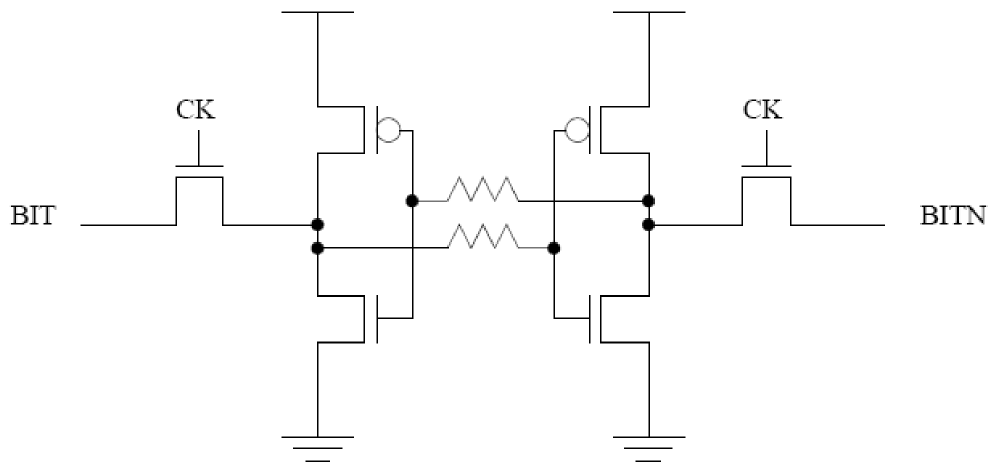

Figure 2.2 Resistive Hardened CMOS SRAM Cell Design [36]

SEU hardening by use of gated resistors is achieved in two ways:

1. Providing adequate off-state channel resistance
2. Maximizing transconductance to achieve maximum on-state channel conductance.

This technique has the advantage of less area overhead. However, the circuit response is slowed down because of the increase in switching time constants. The added intra-cell resistance increase the minimum write-cell time. Dense designs require larger

intra-cell resistances. Higher polysilicon resistivity increases temperature coefficients of resistors, thus the technique becomes inapplicable at higher temperatures. The technique cannot be used for designs with very less feature sizes. The use of resistive hardening for elements in critical signal paths is avoided, because the delays add up and affect the overall circuit response [10].

2.4.3 Circuit and Logic Design Techniques

Circuit design and processing techniques provide radiation hardness. This is achieved by making the stored information insensitive for the usual energy range of incident particles. However there are certain disadvantages such as high cost, low performance, and high power dissipation [3].

Hardening techniques at circuit level ensure immunity against single node upsets and are fully compatible with standard CMOS technologies. They are based on storage latch duplication and use state-restoring feedback circuits which are more compact and add lower delays than the TMR circuits. However because of high area overhead and power dissipation, they are inapplicable to high density circuit architectures [7].

Various SEU tolerant SRAM cells such as Whitaker design, Dice design, HIT cells, Rockett cells, and Barry-Dooley design have been developed [36]. These cells have better electrical performance and consume less silicon area [9]. All these designs are based on three main principles:

1. Information can be stored at two places thus providing a source of uncorrupted data.
2. When an upset is detected, the information from the uncorrupted part can be fed-back to mitigate the error.
3. A p-transistor storing a logic '1' cannot be upset to '0' and a n-transistor storing a '0' cannot be upset to a '1' [36].

2.4.3.1 Ratioing

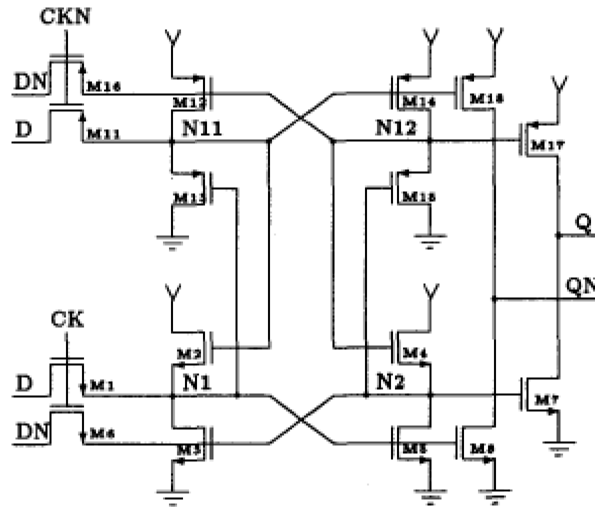

Figure 2.3 Hardening by Use of Ratioing

New logic configurations along with ratioing the strengths of transistors within the cell is a technique generally used. The device sizes are determined from the desired write time, read time and recovery time. The SEU immunity is independent of processing parameters. The main advantage of this method is that there is reduction of loading on the clock signal and transient faults will not propagate from the new flip flop. However, because of the body effects on the device threshold voltage, there is degradation in internal voltage levels which reduces the available noise margin and thus affects the static power. Also, the use of ratioed devices progressively loses the immunity to upset propagation due to the accumulated total dose effects [1]. Figure 2.3 illustrates this.

4.3.2 Rockett Cell

Another cell which uses circuit design techniques is the Rockett cell shown in Figure 2.4 in which no ratioed situation exists between devices. The Rockett cell includes redundant storage nodes that are driven by and connected to p+ diffusion regions. Hence upset due to an ionizing particle occurs only due to upsetting of a low voltage to high.

Since there is no n+ diffusion, logic '1' cannot be upset to '0'. Thus the redundant storage is a source of uncorrupted '1's. Radiation hardening is achieved by transistor sizing [12]. This causes the cell to consume no static power. Also, only a single phase clock is required. The SEU immunity is independent of processing tolerances, voltage supply tolerances, and temperature. However, there are several disadvantages of this technique such as higher capacitance on clock inputs. The outputs of the Rockett cell will not swing rail to rail and, when configured as a flip flop, upsets can propagate from the cell into downstream logic and memory cells [1].

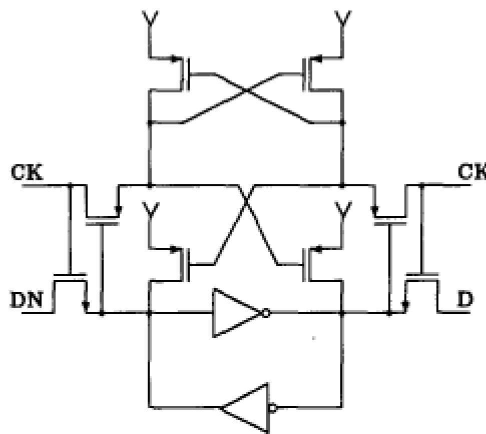

Figure 2.4 Rockett Cell [1]

2.4.4 Redundancy

The basic idea of redundancy is to implement multiple copies of the same circuit, and compare the outputs of each of these circuits. Disparity in these outputs indicates the occurrence of an error. Redundancy technique can be implemented at various levels such as circuits, systems etc. Autonomous or ground-controlled switching occurs from a prime system to redundant spare system. This process of switching is a simple process when both the designs meet the system restrictions identically [2]. Logic paths in between the flipflops are composed of hard-wired, non-reconfigurable gates. Hence they are immune to SEUs. However they are vulnerable to SETs. Thus full module redundancy is essential to protect the device [30].

2.4.4.1 Lockstep System

This technique involves operation of two identical circuits with synchronized clocking. A difference in the output values of the processor indicates the occurrence of SEU. Recovery actions such as reinitializing and switching to safe mode are implemented. False triggers could occur if the devices respond even slightly [2].

2.4.4.2 Triple Modular Redundancy

TMR is the most reliable safeguard for total device failure as it rapidly detects and corrects SEUs. In the reconfigurable logic devices, user logic and logic paths are susceptible to SEUs. This makes the triple modular redundancy an effective technique [20]. The basic concept of triple redundancy is that a sensitive circuit can be hardened to SEUs and SETs by implementing three copies of the same circuit and performing a bit-wise "majority vote" on the output of the triplicate circuit. Implementation of TMR has the advantages of complete data retention and autonomous recovery [30]. It operates with the main aim of removing all single points of failure from the design. Each set of the triplicated circuit has its own set of inputs as shown in Figure 2.5, to avoid errors occurring due to propagation of wrong inputs. However it requires external mitigation device. The sensitivity of the TMR design technique mainly depends on the characteristics of the adopted TMR architecture in terms of placing and routing. Xilinx builds its majority voters from the Output Buffer Three-state cell (OBUFT) provided by Xilinx library primitives. Fault-tolerant memory elements in sequential digital logic are usually implemented by TMR technique [18]. Combination of TMR and scrubbing techniques offer an effective solution for SEU mitigation of SRAM-based FPGA design. By this technique single errors in the user or path logic and static errors in the memory can be corrected before the next error occurs. The circuit is made immune to functional errors [20]. However, this method adds system level overhead and increases the power dissipation [7]. TMR can be implemented in various forms such as Simple TMR, TMR circuit with three voters, TMR circuit with three voters and clock, and feedback TMR [37].

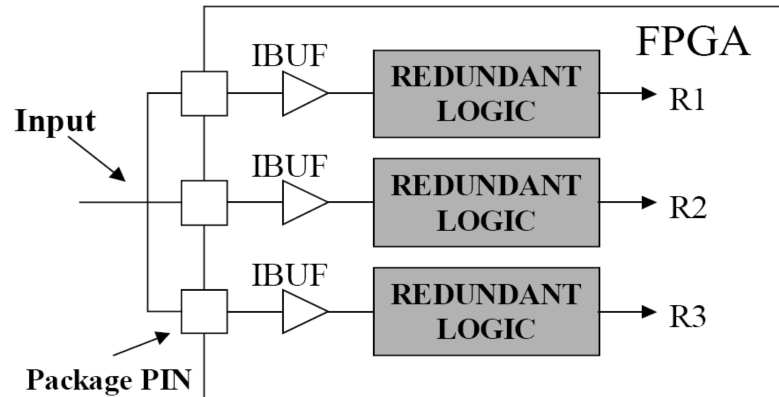

Figure 2.5 Triple Redundant FPGA Inputs [18]

2.4.4.3 Dual Voting Double Redundancy

This technique provides reliable safeguard against total device failure and is used in designs that are less than half of the total device. Logic of the circuit is duplicated and the outputs are compared. An SEU or SEFI is said to occur if there is a difference in the output. This technique operates without external mitigation device and has independent input-output mitigation. This has the advantage that in case of total device failure, the redundant device continues processing. There is also no single point susceptibility [15]. The disadvantage of this technique is that skew in the output transitions times increases noise. Figure 2.6 shows the implementation.

2.4.4.4 Selective Triple Modular Redundancy

An effective technique employed for SEU mitigation is the Selective Triple Modular Redundancy. This technique selectively applies TMR on sensitive sub-circuits of a given combinational circuit. The sensitivity of a gate to an SEU is determined by the signal probability of its input lines. A gate is said to be sensitive if an SEU on any one of the inputs is likely to be propagated to the output of the gate. The advantage of this technique is less area overhead. [11].

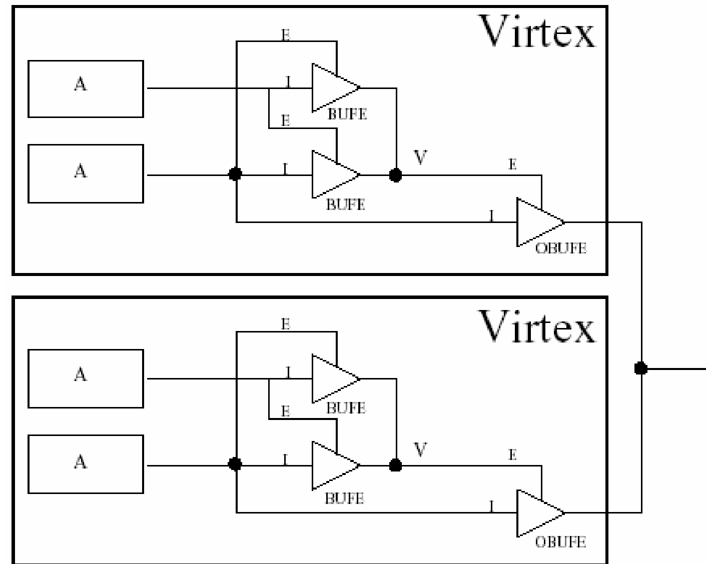

Figure 2.6 Dual Voting Double Redundancy [15]

2.5 Miscellaneous Techniques

1. A new technique for designing radiation tolerant circuits is re-programmable FPGA technologies. There are two kinds; one of it uses a FLASH/EEPROM configuration switch while the other uses an SRAM switch. Each design is implemented as a configuration of many switches. Latch type storage elements can be hardened by increasing the threshold. This can be done by increasing the critical charge by circuit design techniques or by reducing charge collection by technology changes such as SOI or by wafer fabricating process changes such as thin epitaxial silicon or double well. SRAM based Re-programmable FPGA for space applications are explained in [5].

2. Watch dog timers are operated by internal microprocessor timers and pass Health and Safety messages between spacecraft systems. They can be implemented in hardware or software and at many levels. If safety message is not received in certain amount of time, the system takes action on the device. There are many recovery actions such as resetting pulse, removing power, sending a telemetry message to the ground, or placing the spacecraft in safe mode. Watch dog timers could be active or passive. A passive watchdog does not send health messages but monitors normal operating conditions [2].

3. Separation of p-type and n-type diffusion nodes within a memory circuit also helps in SEU mitigation [2].

SEU failure rate is a function of the logic density. Size of the design and resources used by the design influence the failure rate of a design. The circuit functionality of larger designs which use more logic and routing resources is defined by a larger portion of the configuration bit stream. Smaller designs that use fewer resources contain more "don't care" configuration bits within the bit stream and can tolerate more configuration upsets [16]. Before the final launching of the object into space, radiation risk assessment must be done.

Definition: Radiation Risk Assessment: A radiation risk assessment for any electronic device includes the determination of total dose damage and SEE susceptibility of the device caused by the projected radiation environment of the spacecraft [38].

High energy protons, electrons and secondary radiation cause total dose damage, while heavy ions contribute to SEE. System hardness for space devices is assured by analyzing all electronic parts susceptible to SEE. The radiation environment experienced by a device depends on orbital parameters, solar activity level, shielding provided by the spacecraft walls and materials inserted between the device and outside environment [38]. Preliminary check on radiation hardened devices includes evaluating the radiation environment that the device will be exposed to. Thus preliminary check should be made on the device before assessing the total dose and SEE sensitivity of the devices selected for system design. Exposure of the device to gamma rays from a Cobalt-60 source helps in the evaluation of total dose test. Low dose rates are recommended for space applications as most device types and technologies exhibit higher total dose tolerance at low dose rates. The dose rates used depend on the predicted device radiation sensitivity and the projected mission total dose [38].

There are two kinds of testing. One is the static type and the other is of dynamic type. Static testing: This involves quantifying upsets in the configuration memory elements without toggling clock, inputs and outputs of a fully-configured device during

irradiation. Dynamic testing: This requires observation of a functional design under irradiation to determine the sensitivity of the combinatorial logic as well as upsets during transient signal propagation [20]. The parameters to be considered for any fault injection experiment are selection of fault location, injection time, fault duration and input stimuli for the application [22].

Space radiation is simulated within a natural space environment by using ground-based radiation sources to study the behavior of upsets within the FPGA configuration memory. Total dose response and proton-induced single-event effects are tested with the help of electron linear accelerators and proton cyclotrons. This method, however, has several drawbacks:

1. Radiation testing is relatively expensive.
2. The number of radiation tests is limited by the availability of the testing facility and the need to travel.
3. Ground-based radiation tests insert high-energy particles into the device in a random, undirected manner. This does not allow the ability to create targeted tests that evaluate the behavior of specific FPGA resources [16].

2.6 Partial Evaluation

Partial evaluation is an optimization technique commonly found in functional programming languages [24]. Unnecessary recalculation can be avoided in functional languages by making specialized versions of functions by partial evaluation. Partial Evaluation simplifies code structure by eliminating statically determinable computations in the source code. This consists of symbolically executing the program, unrolling loops, unfolding subroutine calls, and performing static data manipulations based on the program input [25]. These processes remove the redundant elements and result in smaller, faster circuits which are more suited for implementation on an FPGA. Calling a function with too few parameters is called partial application. The IO and routing resources are limited in FPGA. The requirement of IO and routing resources can be reduced by partial application of inputs. The difference between partial evaluation and other existing

simplification techniques is that partial evaluation performs simplification at run-time while others are implemented at compile time [24].

The timing upper bound should be lower for a specialized circuit when compared to the general circuit. Partial evaluation results in faster execution and better space utilization. The need to hold “constant” values is eliminated and this reduces the complexity of the circuit. Although partial evaluation has proven to be quiet effective for software problems, its use in hardware is highly restricted. The simplest form of hardware partial evaluation is Boolean optimization. Not much research has been done to use partial evaluation in hardware optimizations. The static nature of hardware makes partial evaluation an uninteresting technique. The circuit cannot be specialized dynamically at run time because of the static nature. However, dynamic reconfiguration of circuits is gaining importance with the advent of SRAM based FPGAs [24].

The main techniques implemented by partial evaluation are:

1. Symbolic computation which means to compute with expressions involving variables along with values.
2. Unfolding which replaces call with instantiated body of function.
3. Program point specialization which creates new version of function specialized to some arguments. This technique can be thought of as a combination of two aspects, namely definition and folding. Definition means to introduce a new definition or extend an existing definition, while folding means to replace an expression with a function call.
4. Memoization which stores the result of some computation and reproduces it again when needed [34].

Advantages of implementing partial evaluation:

1. Speed up: Decreases the time to compute and time to run the program.

2. Efficient and modular solution: Partial Evaluation specializes generic program to specific instance [34].

Development in partial evaluation has allowed the following improvements with respect to FPGA design:

1. Partially evaluated hardware description language design descriptions to generate run-time specialized software for Xilinx FPGA devices.
2. Similar technique is applied for data encryption circuits as well.
3. A method was proposed for expressing dynamic reconfiguration for FPGAs by means of partial evaluation.
4. Partially evaluatable circuit generators are developed for FPGA based circuits.
5. Transistor level partial evaluation has been developed for symbolic simulation.
6. A component reduction algorithm has been developed that partially evaluates a transistor level library element to generate a reduced version of the library component when some operands are tied to constants [28].

By using partial evaluation, partial evaluators have been designed which generate efficient specialized programs for ray tracing, FFT, circuit and planetary simulations. Partial evaluators also compile using interpreters for programming languages and generate compilers from interpreters [34].

2.7 Temporal Latch

Consider 3 delay paths, one with no delay, the other with a delay dT added to it and the third with a delay 2dT added. Under the normal operation of a temporal latch it rejects transients by setting a pre-determined delay between these 3 delay paths [6].

Any transient occurring travels down the first path with no delay, the second path with a dT offset and the third with a 2dT offset. As long as the transient is shorter than dT only one path can have an error and the majority voter determines the correct output by

rejecting the transient. However, if the transient pulse width is greater than dT, then two paths of the temporal latch are corrupted and the majority voter outputs wrong value as the error is passed through it. Thus the variation in the value of dT makes the temporal latch fully SET/SEU immune. The value of dT should be larger than any transient possible. This makes the temporal latch immune to all possible soft errors. Figure 2.7 shows the relation between pulse width of transient and the delay time dT [6].

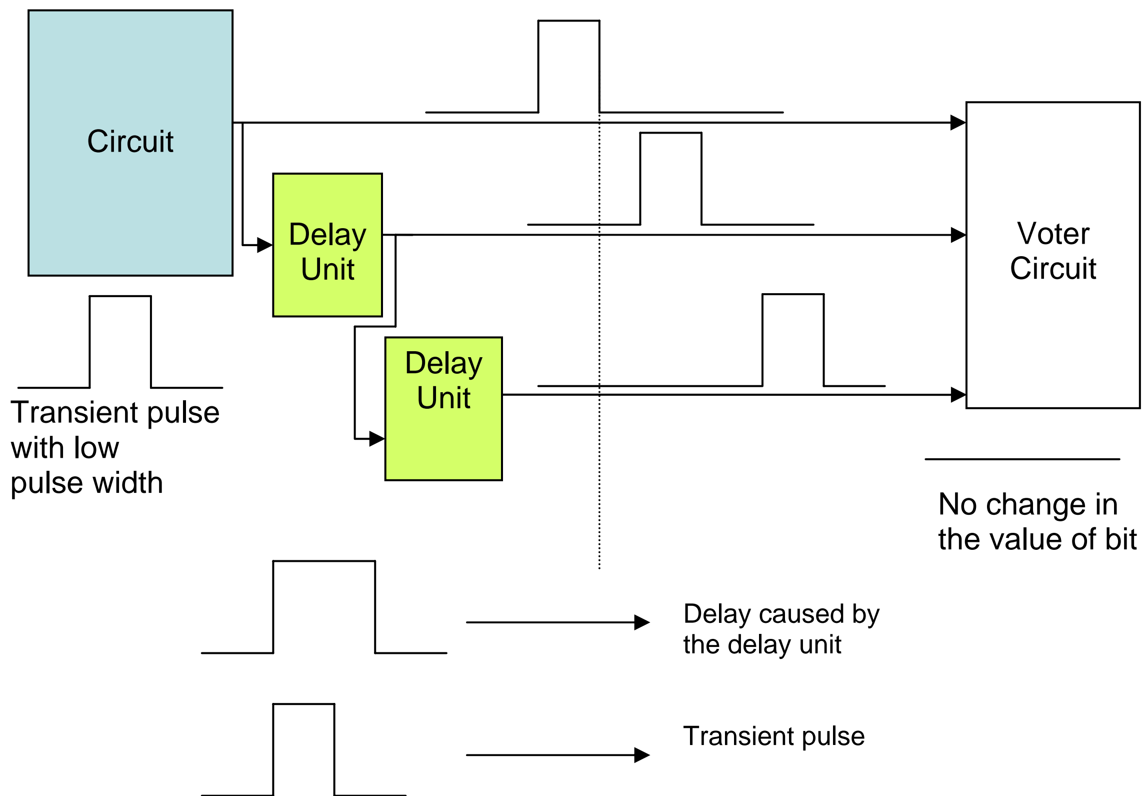

Figure 2.7 Effect of Transients With Low Pulse Widths

Since the pulse width of the transient is smaller than the delay time dT, it is captured by the first path but rejected by the other two paths. Since two out of the three values give a correct output, the majority voter gives a correct output.

When the pulse width of the transient is greater than the delay time, then it is highly likely that two out of the three paths allow the transient to pass through. Since two out of the three output values are wrong, the majority voter circuit finally outputs a wrong value. This is illustrated in Figure 2.8.

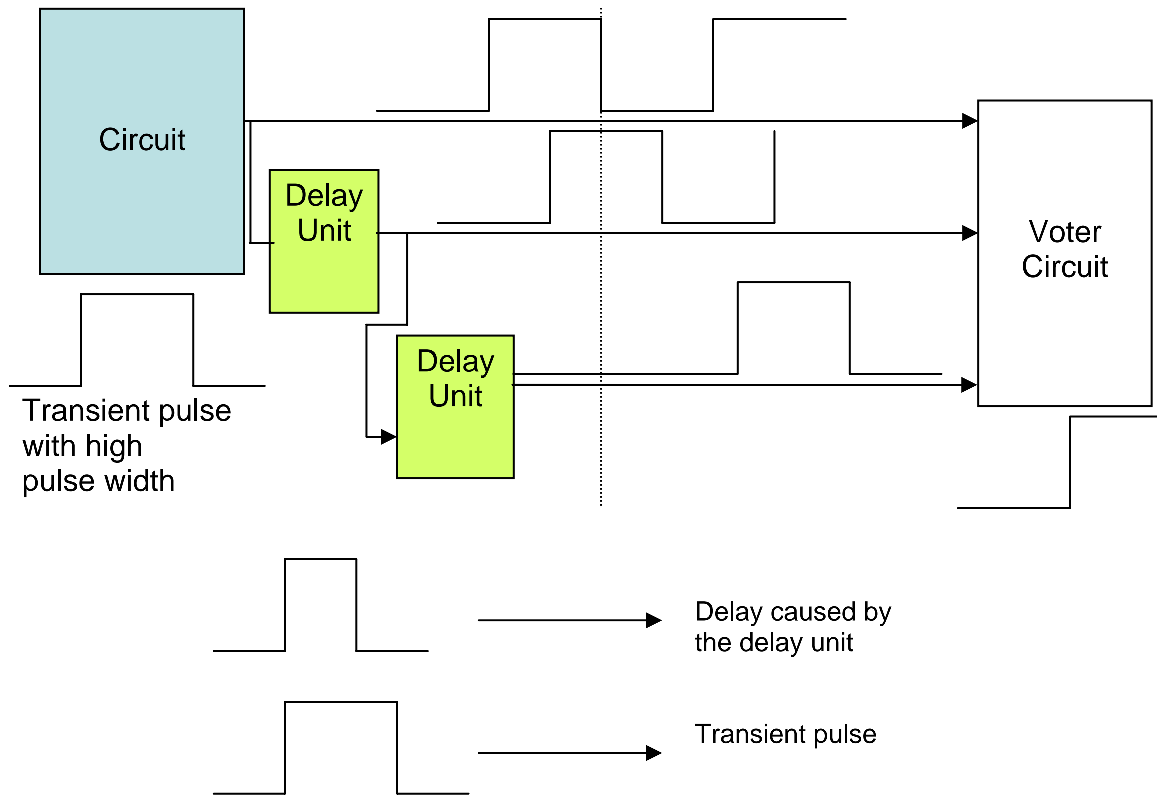

Figure 2. 8 Effect of Transients With High Pulse Widths

The following changes are observed in a conventional static latch when it experiences SEU:

1. Junction collection in static latches and SRAMs
2. Voltage transient effects magnified by circuit feedback.
3. Data state of latch flips if voltage crosses switch point.
4. Critical charge decreases as square of feature size
5. Error rates are independent of clock frequency [14].

The following changes are observed in a conventional static latch when it experiences SET:

1. Junction collection in combinational logic circuitry.
2. Voltage transients are no longer attenuated in submicron devices.

3. Incorrect data gets latched if transient arrives at a clock edge.
4. Critical width for unattenuated propagation decreases as square of feature size.
5. Error rates increase in proportion to clock frequency [14].

To overcome the above changes, temporal sampling latches are designed: This can be done in two ways. One is by spatially reducing the circuit and the other technique is to temporally reduce the clocking [14].

1. Spatially redundant circuit (as shown in Figure 2.9):

    1. This circuit replaces the conventional latches.
    2. It uses three parallel sampling latches.
    3. The resultant circuit is immune to latch SEUs, clock SETs and set/reset SETs.
    4. However there is an error latency of one clock cycle.

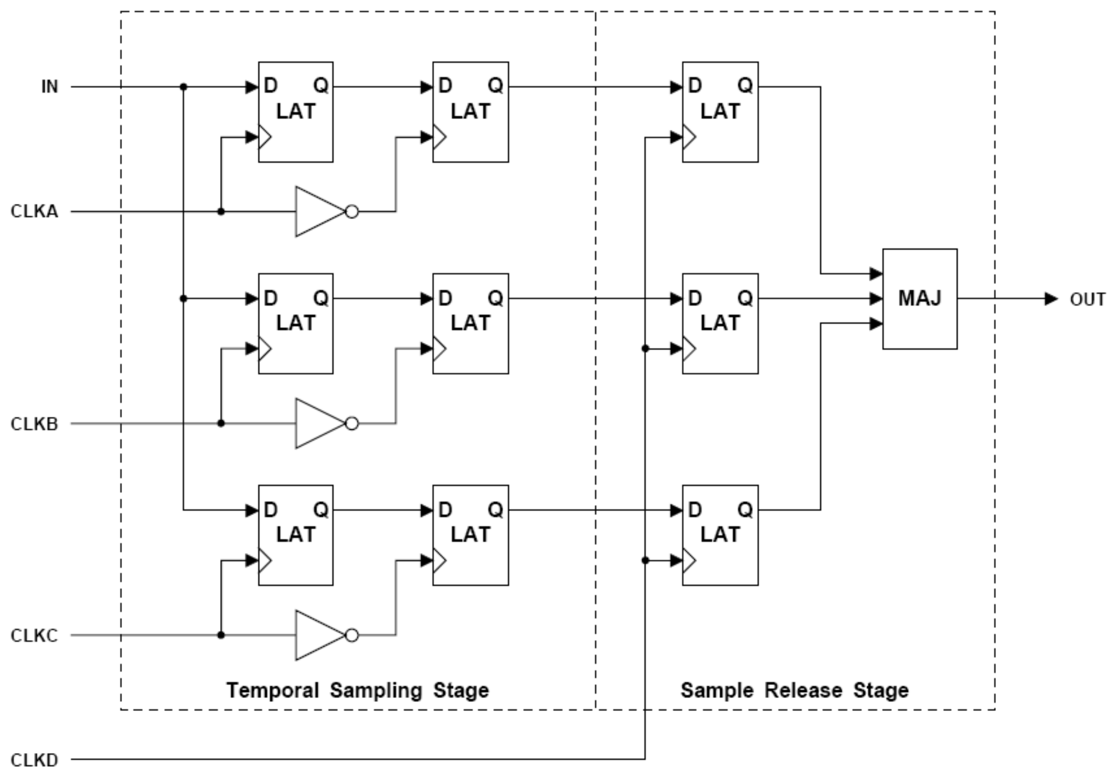

Figure 2.9 Spatially Redundant Latch

3. Temporally redundant clocking (as shown in Figure 2.10):

    1. This replaces the conventional clock.
    2. It has three sampling clock phases.
    3. One data voting occurs for each clock phase.

4. DICE latch with programmable separation delay (0.8 micron).
5. Temporally redundant sampling latch for deep submicron [14].

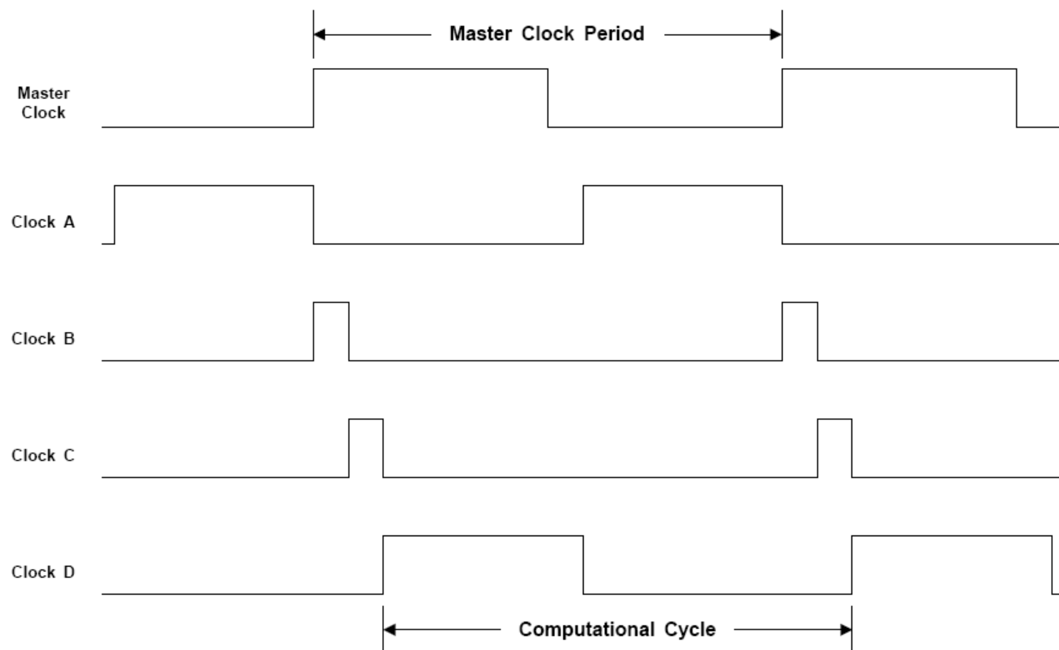

Figure 2.10 Temporally Redundant Latch

2.8 Voter Circuits

Single event errors are mitigated through a majority vote circuit. Assuming the voter circuit itself will not upset, a majority vote circuit requires multiple upsets for a logical error to occur [15]. A simple implementation of voter circuit using the basic logic gates is shown in Figure 2.11. Voter circuits used in traditional SRAM FPGAs are

implemented using LUTs as shown in Figure 2.12. Faster circuit implementation can be achieved by building majority voters using LUTs. Majority voters of this kind are used in designs where logic resources are not a constraint but fastest possible timing performance is required. Combinatorial logic is usually implemented using LUTs. This helps in decreasing the propagation delay. However, immunity to SEUs is less [30].

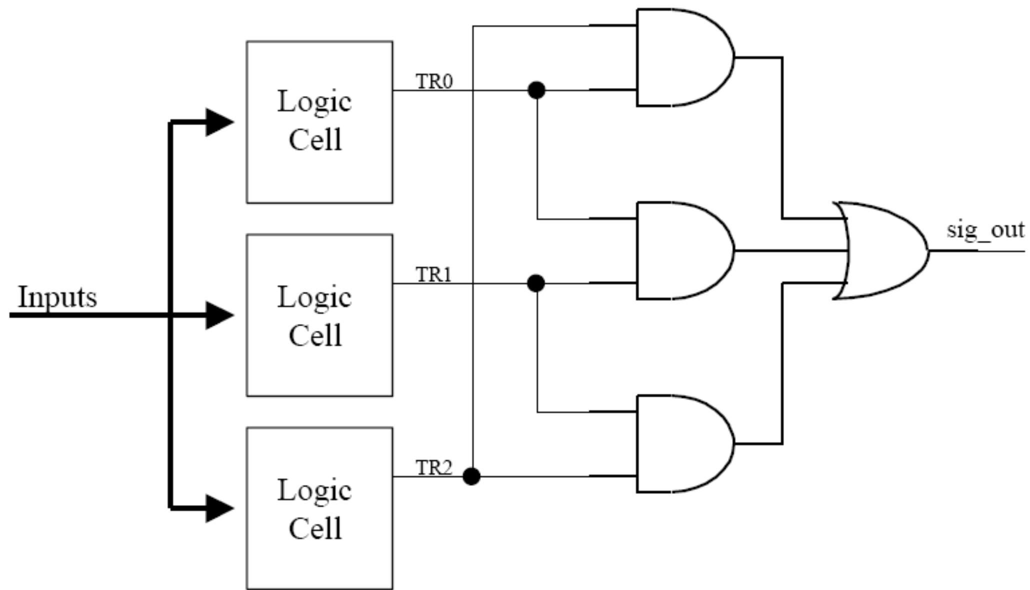

Figure 2.11 Majority Vote Circuit [15]

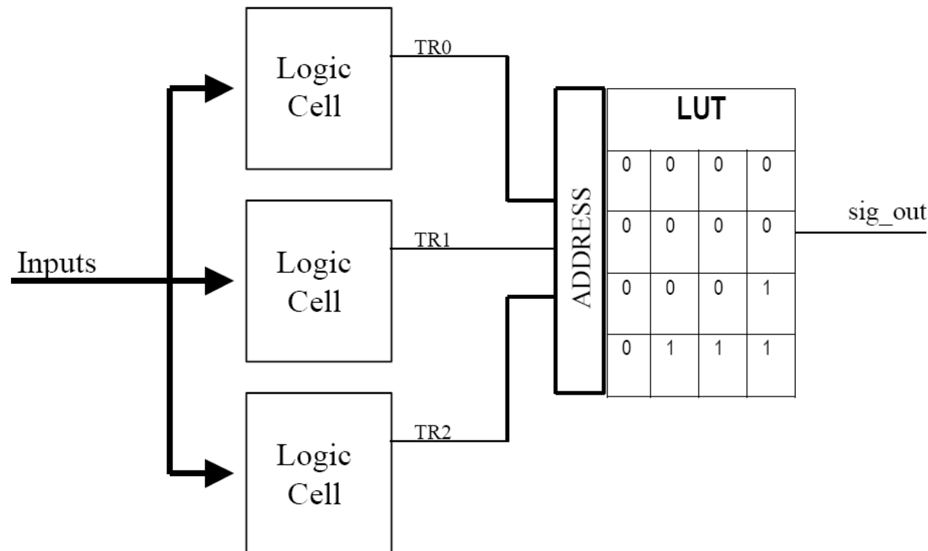

Figure 2.12 Majority Vote Circuit Using LUTs [15]

Virtex makes use of the abundant and free BUFTs for voter to make the circuit immune. Virtex internal 3-state buffers are hard-wired OR-AND gates and are used to implement all Boolean functions in the user's design. They do not use any CLB (logic) resources [15]. In designs where the available logic resources are less in number, majority voters can be implemented using these active low enabled 3-state buffers. However, the full functionality of the majority voter cannot be simulated. All inputs to the voter should be same to ensure that the user's design is free from SEUs [30]. The BUFTs are instantiated as shown in Figure 2.13.

A different kind of voter implementation is by use of simple logic module in which combinations of 4-input Boolean functions implement a logic equation as shown in the Figure 2.14. The register stages are pipelined and this enhances the performance. However the disadvantage of this implementation is that LUTs and FFs are susceptible to SEUs and interconnects are susceptible to transient upsets [15].

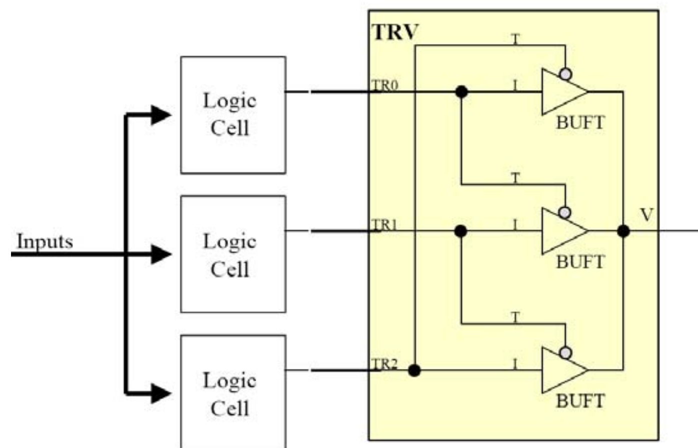

Figure 2.13 Majority Vote Circuit Using BUFTs [15]

Figure 2.15 shows the implementation of module redundancy and mitigation on a single chip. Redundant instances of the entire module are replicated and the final outputs of the modules are mitigated. This is a simple method for implementing SEU mitigation on FPGA designs.

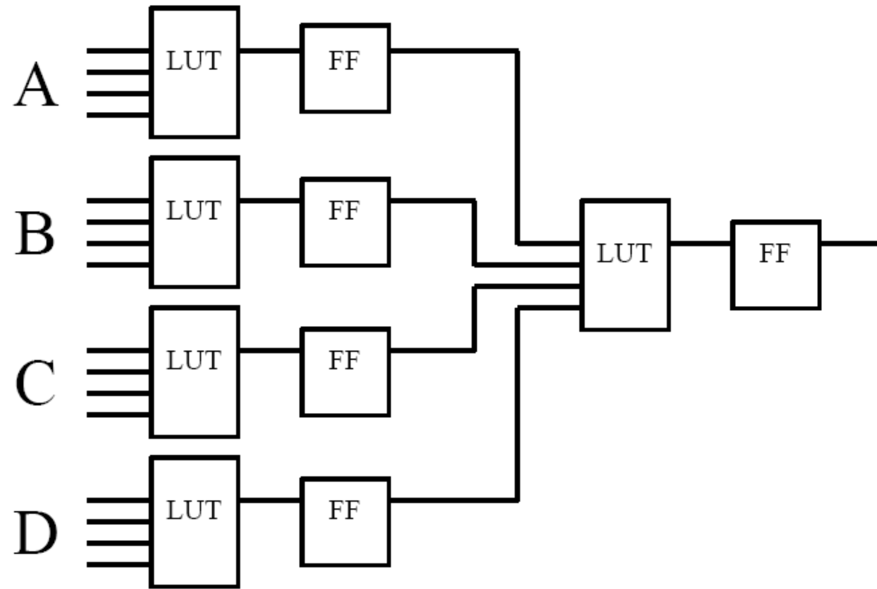

Figure 2.14 Majority Vote Circuit Using LUT and FF [15]

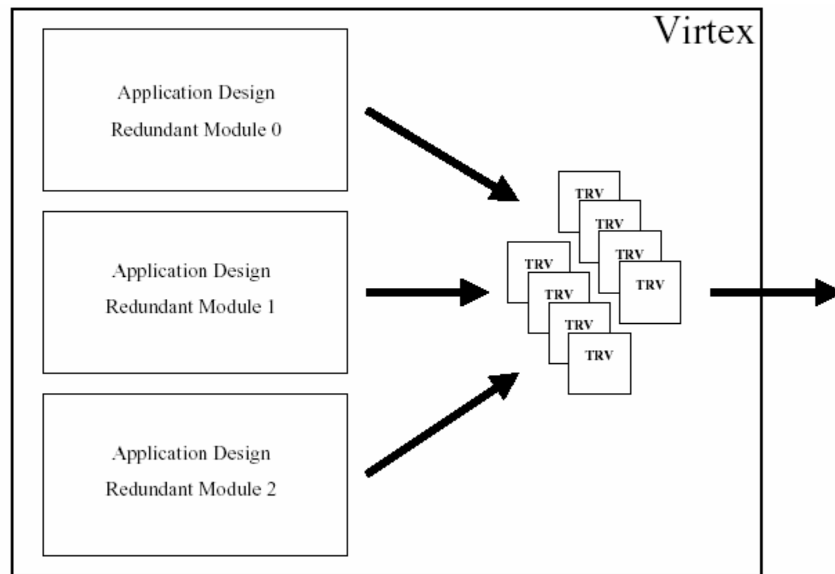

Figure 2.15 Implementation of Module Redundancy and Mitigation On Single Chip [15]

2.9 Summary

We find that among the various radiation effects, single event upsets are of major concern. Protecting the space electronics from SEUs improves efficiency. There are various techniques to achieve this. Radiation hardening by fabrication is an effective technique but increases the cost of circuit. Thus radiation hardening by design is a better option. One of the techniques implemented in hardening by design is redundancy. The development of SRAM based FPGAs is encouraging the use of partial evaluation in hardware. By using partial evaluation an area, and performance efficient technique can be devised for hardening by design.

CHAPTER 3

PARTIAL EVALUATION REDUNDANCY

We present in detail the proposed Partial evaluation Triple Modular Redundancy (PTMR) method. The overall idea is as follows: Given the primary input signal probabilities, the gates that can be eliminated are determined. Circuit that is obtained after eliminating the redundant gates is duplicated and outputs obtained from the original circuit and the two duplicated circuits are voted. This technique can be implemented if the actual inputs to the circuit are in accordance with the assumptions made. In case the actual inputs are violating the assumptions made, temporal triple modular redundancy is implemented. Thus, efficiency of the technique depends on the nature of input probabilities and the actual inputs to the circuit.

This chapter is organized as follows: Since the technique is based on the input environment information, we first discuss how to obtain such information. We then talk about the basics involved in identifying the gates that can be eliminated. Based on this we proceed to describe the technique in detail. Finally, we shall illustrate the idea using a small example.

3.1 Definitions

Redundant Gate: A gate is said to be redundant if its output can be determined in advance based on the knowledge of its inputs.

Rounding the Input Probabilities: Suppose the input probability p is such that $0 \leq p \leq 0.2$, we round the probability value to ‘0’ and in case the probability is such that $0.9 \leq p \leq 1.0$, then we round the value to ‘1’.

Input Value in Accordance With / Against the Assumed Probability: If the input probability is in the range $0 \leq p \leq 0.2$, it is rounded to 0.0. In case the actual value on this input line is a logic ‘0’ then it is said that input value is in accordance with the assumed probability, while a logic ‘1’ means that the input value is against the assumed probability. Analogous definitions hold for probabilities p such that $0.9 \leq p \leq 1.0$.

Controlling Value of a Boolean Gate: If the presence of a value on at least one of the inputs of a gate forces the output to a known value.

For instance, the controlling value for “AND” and “NAND” gates is ‘0’ while for “OR” and “NOR” gates it is a ‘1’.

Sensitizing Value: It is the Non-controlling value, i.e. the complement of controlling value.

For instance, the controlling value for “AND” and “NAND” gates is ‘1’ while for “OR” and “NOR” gates it is a ‘0’.

3.2 Characterizing Input Environment

Input environment is characterized by a popular method called profiling. Software profiling techniques are widely used in the software development to identify the often executed portions of the code. Profile data is gathered from representative benchmarks. Low power systems are designed by profiling hardware design.

The profiled data can be summarized either in the form of input signal probabilities or in terms of “representative” input sequence. In the latter case, vector-compaction based scheme has been proposed to reduce the length of such sequences. Representative sequence to input probabilities can be reduced by simulating the circuit with the sequence. This justifies the assumption that the input environment information is available in the form of input signal probabilities.

A method of calculating the probability of an upset due to a SET on a given combinational circuit was proposed in the context of SEU-hardening synthesis technique.

The probabilities are determined based on the radiation environment it will be subjected to and the nature of the circuit.

In the context of ASIC testing, the detection probabilities of the lines can be determined from the signal probabilities. The detection probabilities of the lines are calculated by propagating the signal probabilities of the primary inputs through the circuit [36]. The signal probability at the output of a gate is determined as shown in table 3.1.

Table 3.1 Computation of Output Probability

Gate Type	Output probability
AND	$\prod_i P_i$
NAND	$1-\prod_i P_i$
OR	$\sum_i P_i - \prod_i P_i$
NOR	$1 - (\sum_i P_i - \prod_i P_i)$
XOR	$\sum_{i,j} P(i)\left(1-P(j)\right)$
XNOR	$1 - (\sum_{i,j} P(i)\left(1-P(j)\right))$

3.3 Implementation of the Partial Evaluation Based TMR Technique

This technique is based on input environment as well as the truth table of basic logic gates. If one of the inputs to a logic gate is a controlling value, then the gate is

redundant since its output can be determined in advance. Thus the gate is eliminated and its output value is set accordingly.

The input probabilities are rounded and then propagated to the output. Now based on the above concepts, the redundant gates are identified and the output values are set accordingly. For instance, if one of the input probabilities of an AND gate is in the range 0.0 to 0.2, it is rounded to 0.0 and is identified as redundant. The output net value is set to 0. The resultant circuit obtained after eliminating all the redundant gates is functionally equivalent to the original circuit but has less number of gates. This reduced circuit is then duplicated. The output value is determined by three circuits, the original circuit and the two sets of reduced circuits. The correct output is obtained by taking the majority vote of these three outputs. The implementation of redundancy using partially evaluated circuit is shown in Figure 3.1.

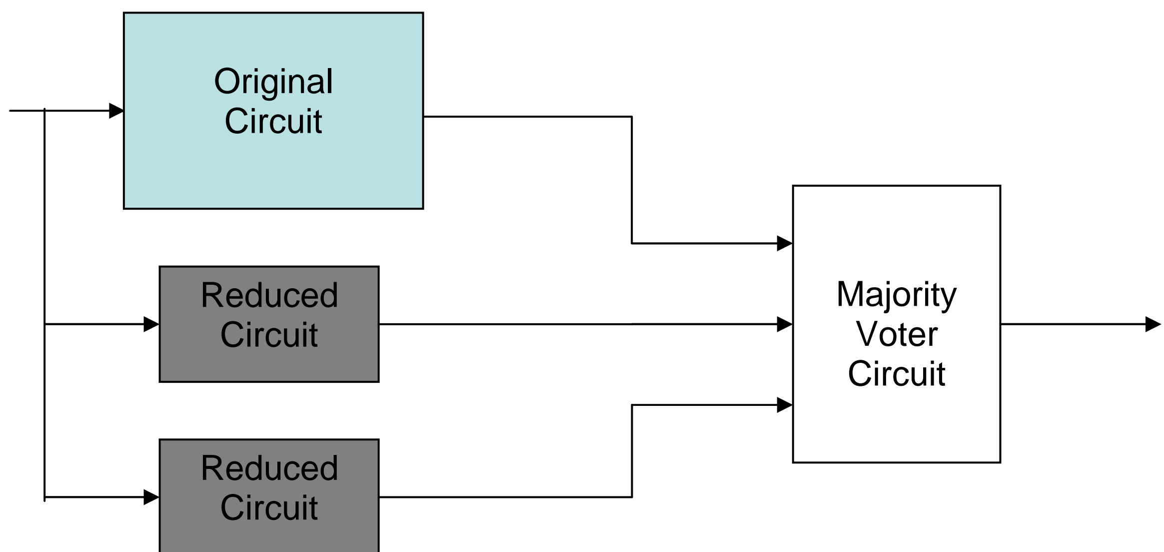

Figure 3.1 Implementation of Redundancy Using Partially Evaluated Circuit

This technique works fine if the actual inputs to the circuit are in accordance with the rounded values in all cases. Incase any one of the inputs is against the rounded value then Temporal TMR technique is used instead of the partial evaluation. To determine the outputs using temporal TMR, the idea of delaying is used. First the outputs of the circuit are determined. These outputs are then passed through a series of two delay elements.

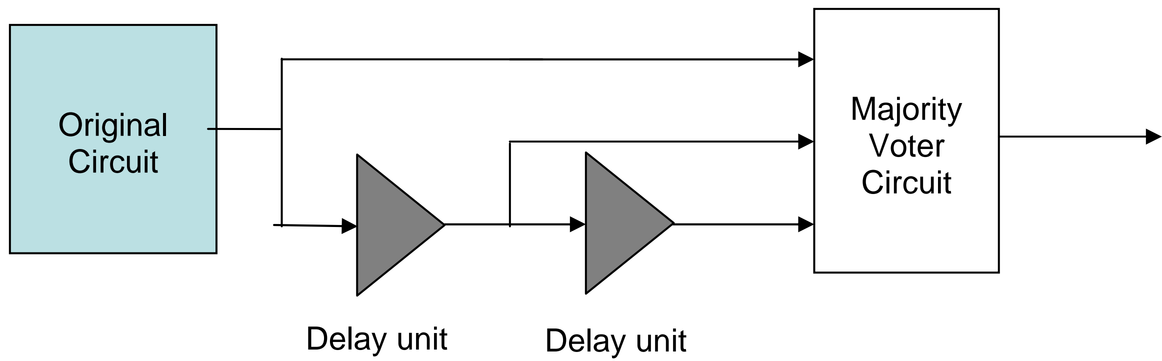

Figure 3.2 Evaluation of Temporal TMR

Thus, we have the outputs of the circuit determined at three different time instances. A majority vote of these three outputs is taken in order to determine the correct output of circuit. The implementation of Temporal TMR is shown in Figure 3.2.

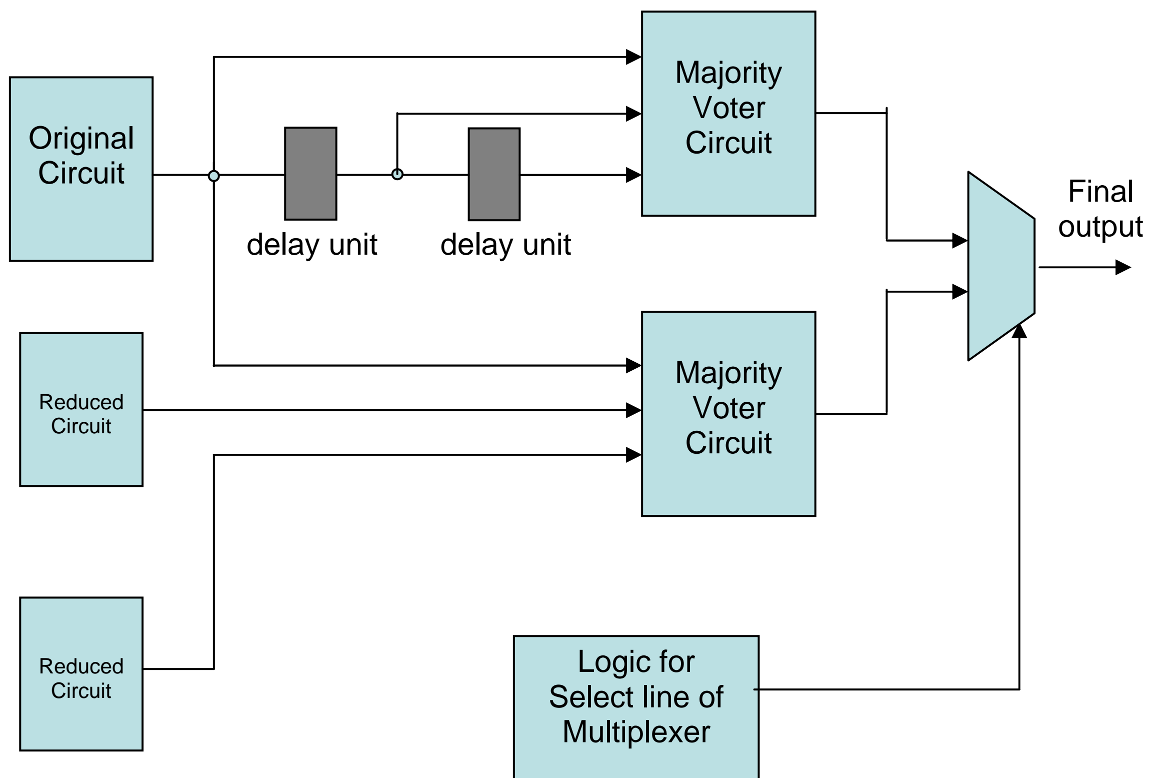

Figure 3.3 Overall Implementation of the Technique

```
1.  Algorithm Reduced_Circuit (gate G, input probability p)
2.  Inputs: A Boolean Gate G, and 0 < p <1.
3.  Output: copies of reduced circuit
4.  begin
5.  /* Levelize the circuit */
6.     Maxlevel ← Levelize (C)
7.  /* Resolve logic on signals at each level */
8.     for each level l from 0 to Maxlevel do
9.        for each gate g at level l do
10.          Compute the signal probability of g's output
11.       end for
12.    end for
13. /* At each level find if a gate can be eliminated */
14. for each level from 0 to Maxlevel do
15.       for each gate g at level l do
16.          switch (gate type of G):
17.             case: AND
18.                call Algorithm AND
19.             case: OR
20.                call Algorithm OR
21.             case: XOR
22.                call Algorithm XOR
23.             case: XNOR
24.                call Algorithm XNOR
25.          end switch
26.       end for
27.    end for
28. for each gate g that can be eliminated
29.    C' ← C –gate g
30.    C" ← C'
31. end for
32. end Algorithm Reduced_Circuit
```

Figure 3.4 Algorithm to Generate the Reduced Circuit

Let there be n inputs with probabilities in the range $0 \leq p \leq 0.2$ and m inputs in the range $0.9 \leq p \leq 1.0$, then the select line for the multiplexer is $\sum_m \overline{inp} + \sum_n inp$.

When the temporal TMR circuit is used, there are more issues to be considered. There is delay in determining the output because of considering the output at three different time instances. The outputs are computed sequentially, hence the Temporal TMR technique takes more than thrice the time taken by sequential TMR technique. This

means that the next set of inputs have to wait till the output is computed. This requires the use of latches to store the inputs. A signal from the output circuit acts as a clock to this latch. The overall implementation of the technique is shown in Figure 3.3.

3.4 Algorithms used in Partial Evaluation Based TMR

Figure 3.4 gives the algorithm to generate a reduced circuit.
The circuit is first levelized. Gates having primary inputs are level 1 gates. Gates having their inputs as the outputs of level n are marked as level n+1. At each level from 1 to maximum level, each gate at level l is considered. In lines from 8 to 12, the logic value of signals is resolved, i.e., for each gate, the probability of output net is computed. Lines 14 -27 call the necessary algorithm. The called algorithm determines if the gate can be eliminated or not. Lines 29 and 30 generate a circuit that is functionally equivalent to the original circuit but with less number of gates.

1. **Algorithm Identify-eliminatable-gate (gate G, input probability p)**
2. Inputs: A Boolean Gate G, and 0 < p <1.
3. Output: sets the output probability q and returns true if the gate can be eliminated else false
4. **begin**
5. **/*** First check if the gate is already identified as redundant or not */
6.   **if** retun_value ← FALSE
7. /* If the gate is not identified as redundant */
8.   **for** each input i of gate G do
9. /* check if any one of the inputs is a control value */
10.   **if** (p_i = = control value) then
11. /* identify the gate as redundant and set the output name appropriately */
12.   return_value ← TRUE
13.   output net name ← appropriate input net name
14.   **endif**
15.   **end for**
16.   **endif**
17. **end Algorithm Identify-eliminatable-gate**

Figure 3.5 Algorithm to Identify if a Gate is Redundant

Figure 3.5 give an algorithm to identify if the gate is redundant or not.

Line 6 checks to see if the gate is already identified as redundant. If the gate is not identified as redundant the algorithm is executed. Line 10 checks to see if any one of the inputs is a control value. If so, then lines 12 and 13 set the variables.

Figure 3.6 gives the algorithm for overall implementation of the technique. Lines 4-11 determine the select line for the multiplexer. Line 15 determines the output from the technique of partial evaluation and line 16 determines the output of temporal circuit. Line 17 selects the correct output from the outputs determined from partially evaluated circuit and temporal circuit using the select line from line 12.

1. **Algorithm Partial_TMR (Circuit C)**
2. **Begin**
3. **/*** determination of select line for multiplexer */
4. **for** each input i do
5. **if** 0 < p(i) < 0.2 then
6. temp = temp + i
7. **end if**
8. **if** 0.9 < p(i) < 1.0 then
9. temp = temp + $\overline{i}$
10. **end if**
11. **end for**
12. select line of multiplexer = temp
13. /* selection of correct output from the two sets of output */
14. **for** each output o of the circuit do
15. partial (o) = majority voting from circuit C, C' and C"
16. temporal (o) = majority voting of the outputs from temporal circuit
17. final output (o) = mux (partial (o), temporal(o), select line, output(o))
18. **end for**
19. **end Algorithm Partial_TMR**

Figure 3.6 Algorithm That Describes the Overall Implementation of the PTMR Technique

3.5 Advantages and Disadvantages

3.5.1 Advantages

1. Less area overhead: In the full TMR, the entire circuit is tirplicated. This increases the area overhead. In case of partial evaluation, two copies of reduced

circuit and one copy of original circuit are used. This decreases the area overhead.

2. High tolerance to SEUs: The resultant circuit exhibits 100% SEU immunity.

The efficiency of the technique is illustrated in Figure 3.7. Spatial TMR technique has high area overhead, since three copies of the circuit are implemented. However, there is no delay overhead as the implementation is done in parallel. In case of temporal TMR, outputs at three different instances of time are considered, hence there is high delay overhead. Area overhead is less. The proposed technique has area overhead and delay overhead values lying in between spatial and temporal TMR.

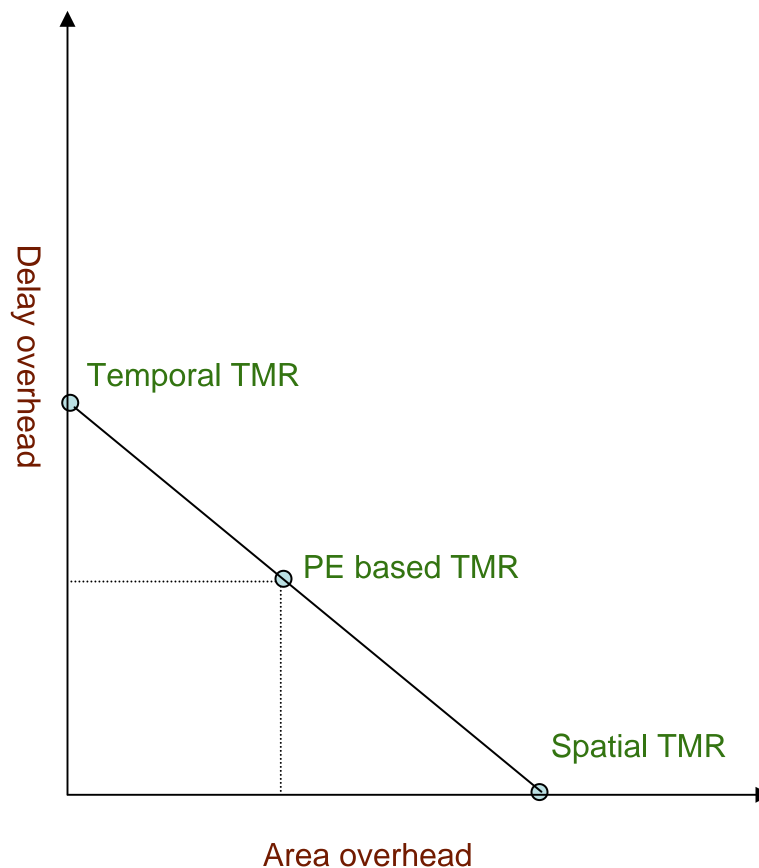

Figure 3.7 Comparison of Area/Delay Overhead For Different Redundancy Techniques

3.5.2 Disadvantages

1. In the worst case, no gates of the circuit could be eliminated. In this case, the area overhead is greater than the full TMR.

2. In cases when all the inputs to the circuits are against the assumed probabilities, Temporal TMR has to be used. This gives rise to certain delay complications.

3.6 Illustrative Example

Consider the circuit given in Figure 3.8.

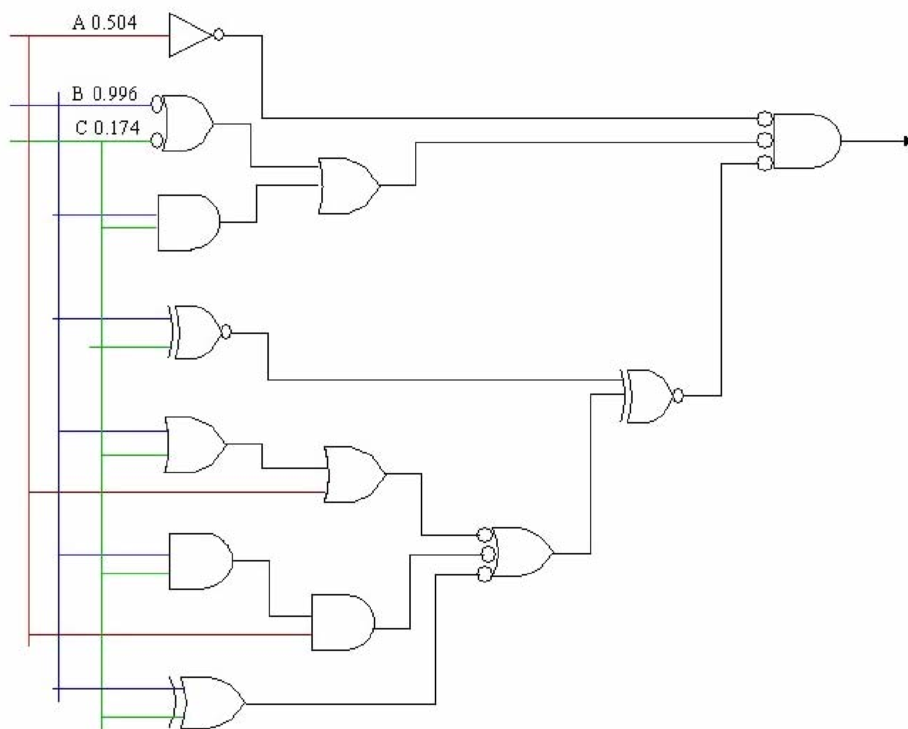

Figure 3.8 Original Circuit

The circuit has 3 inputs, 1 output and 13 gates.

The input probabilities of the circuit are P(A) = 0.504, P(B) = 0.996 and P(C) = 0.174

Since $0.9 \leq P(B) \leq 1.0$, we round the logic value of B to '0' and

$0 \leq P(A) \leq 0.2$, we round the logic value of A to '1'.

We now propagate the rounded values through the circuit as shown in Figure 3.9. The shaded gates indicate that they are redundant.

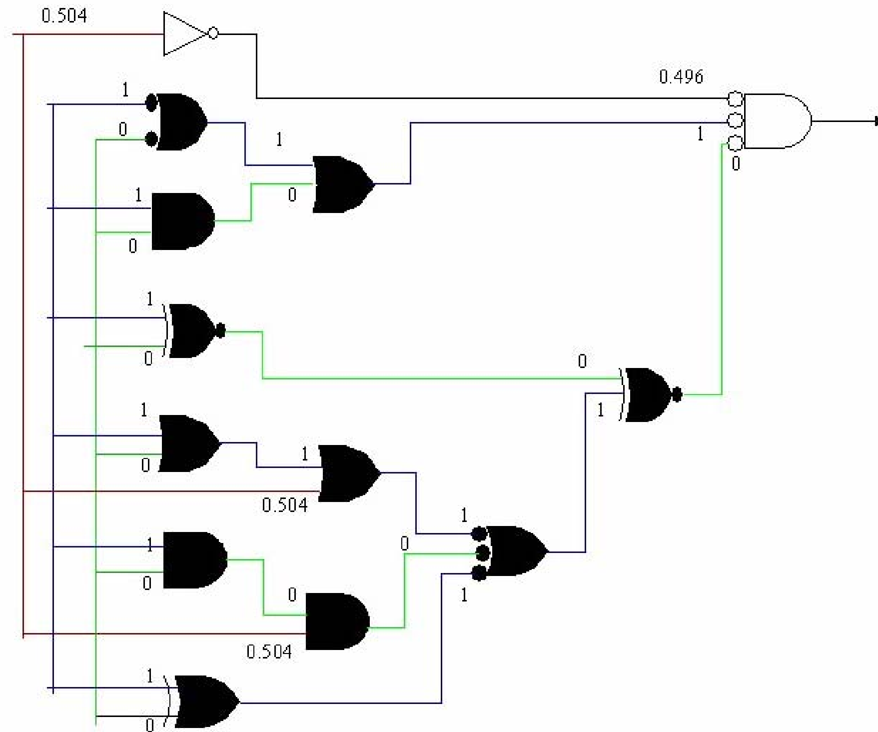

Figure 3.9 The Rounded Values Are Propagated Over the Circuit

Based on the inputs to the gate and type of gate we identify some gates to be eliminatable. On eliminating all the redundant gates we get a reduced circuit as shown in Figure 3.10.

We now have the output determined by the original circuit and the reduced circuit. The lightly shaded gates represent gates of reduced 1, while the darkly shaded gates represent gates of reduced circuit 2. The correct output is obtained by taking the majority vote of these outputs as shown in Figure 3.11.

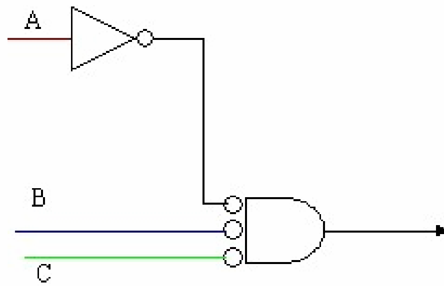

Figure 3.10 Reduced Circuit

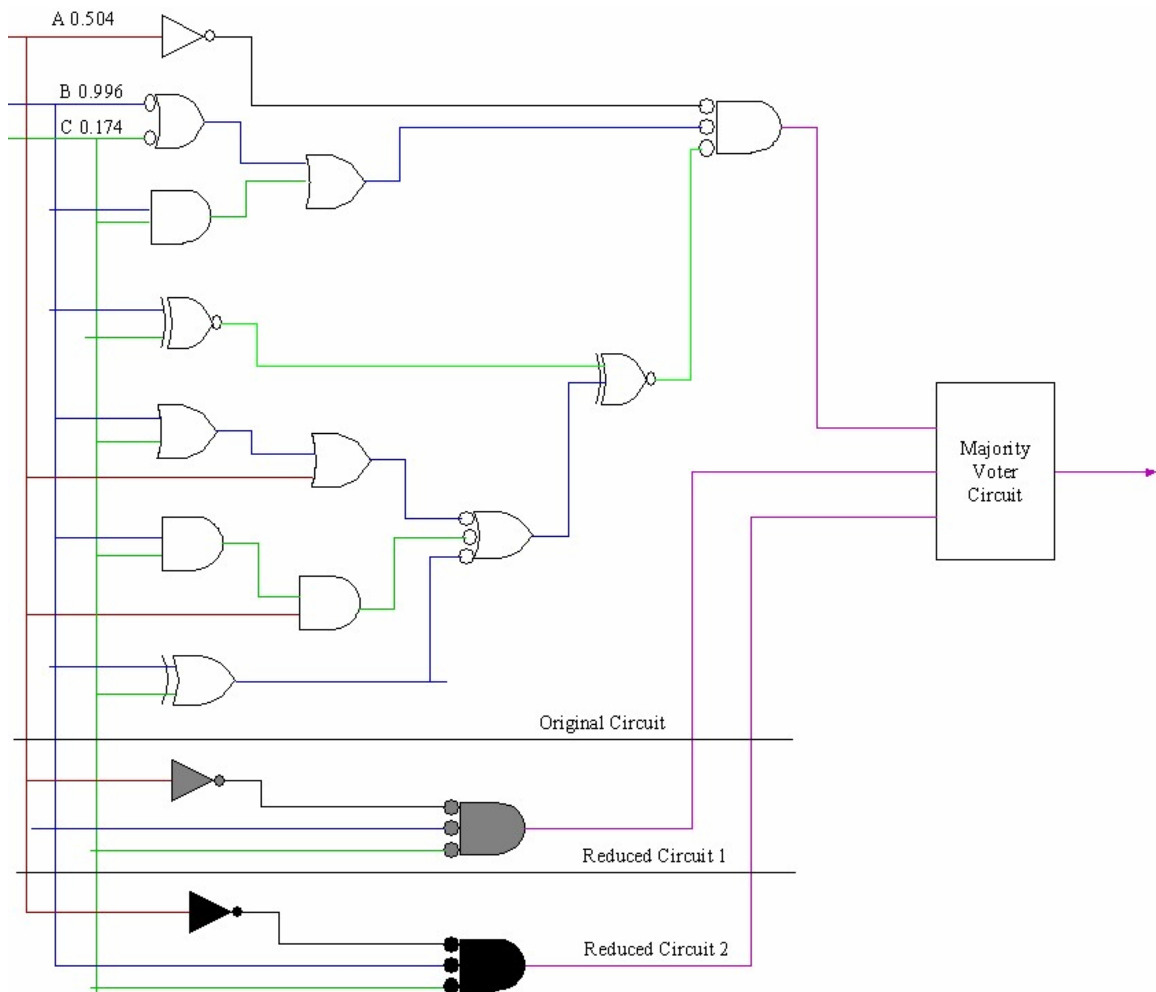

Figure 3.11 Circuit That Computes Output From Partially Evaluated Circuit

The temporal part of the circuit uses two delay units. Thus we have the output of the circuit determined at three instances of time. The correct output is obtained by taking the majority of these outputs. The technique of temporal TMR is illustrated in Figure 3.12.

To choose the correct output from the two sets of outputs, one computed using partially evaluated circuit and the other using temporal circuit a multiplexer is used. If the output of the logic circuit is a ‘0’, then the output of partially evaluated circuit is chosen and if it is a ‘1’, then the output of temporal circuit is chosen. This is shown in Figure 3.13.

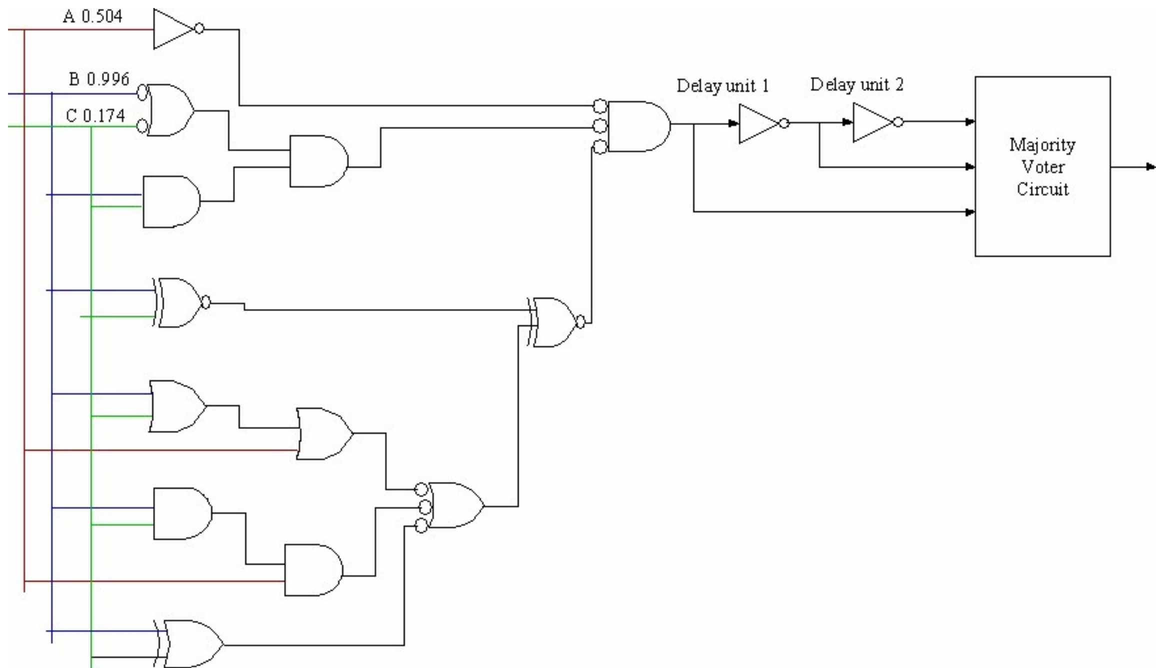

Figure 3.12 Circuit That Computes Output From Temporal TMR Circuit

3.7 Summary

We have presented a design technique for SEU mitigation. The partial evaluation based triple modular redundancy is an efficient technique that provides 100% SEU mitigation in addition to savings in area. Inputs to this technique are given in terms of probabilities. A brief description of characterizing the inputs is presented. The various

algorithms implemented for the technique are described in detail. The advantages and disadvantages of the technique are discussed. The different steps involved are described in detail using an illustrative example.

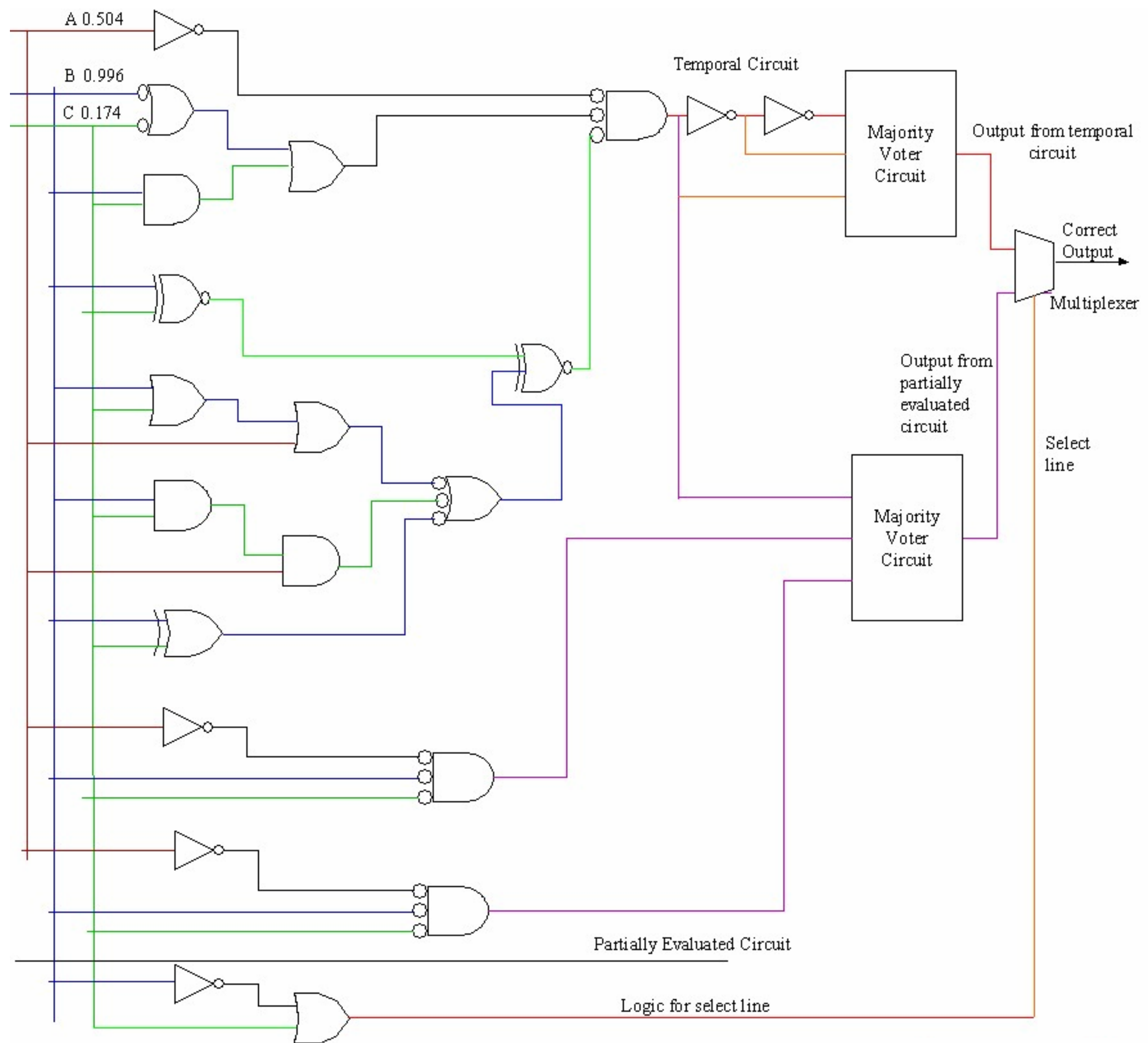

Figure 3.13 Circuit Showing The Overall Implementation

CHAPTER 4

EXPERIMENTAL RESULTS

The experimental flow used to validate the proposed Partial evaluation based redundancy is presented. A SEU simulator is used to insert faults representing SEUs. A functional testing procedure to assess the Immunity of the circuit against SEU is presented. Finally the results of applying the technique on standard benchmarks are analyzed.

4.1 Experimental Setup

The experimental flow is as shown in Figure 4.1.

1. Inputs to the circuit are in the form of BLIF (Berkeley Logic Interchange Format) format. These netlists in blif format are converted to VHDL format. This is to allow the use of Xilinix Foundation tools 4.1i to map the designs onto Virtex FPGAs.

2. Insertion of Partial Evaluation Technique: Random numbers are generated using a a random function. These random numbers are given as inputs to the circuit. These input probabilities are propagated over the circuit by using the Table 3.1. Based on the algorithm described which is coded in C the gates that can be eliminated are identified. This gives a circuit which is functionally equivalent to the original circuit but has less number of gates. The VHDL netlist generated from step 1 is given as the input to the C program. A new netlist is generated which represents the overall technique. This netlist is passed through the SEU

simulator. The results so obtained represent the behavior of the circuit before mapping.

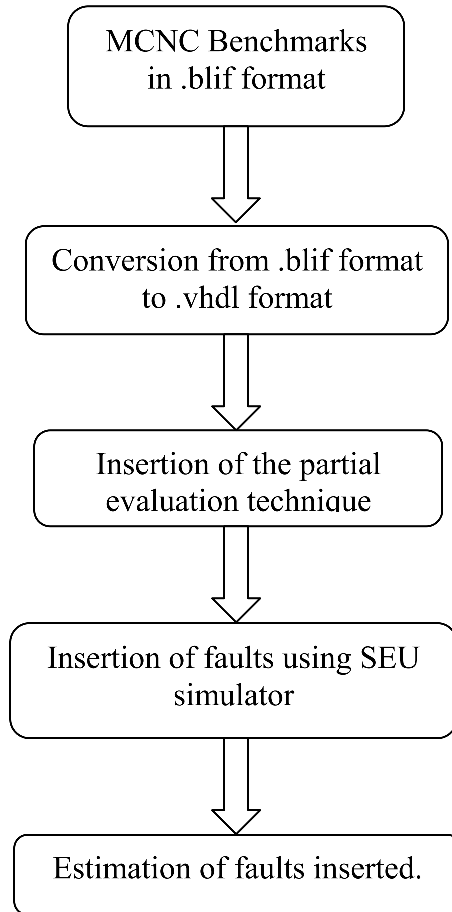

Figure 4.1 Experimental Flow

3. SEU Simulation: Faults due to SEUs are represented by a SEU simulator. It works on the following principles:

   i. The simulator randomly injects a fault on any one signal. This represents the situation that a fault due to SEU can occur on any line of the circuit.
   ii. An SEU at a node temporarily inverts the value on that line. Duration of the SEU is controlled by the user.

iii. SEU introduction is a random process and can be done at any instant of time.

The SEU simulator generates outputs equal to the number of signals in the circuit and each output has value of logic ‘Z’ or logic ‘1’. Thus the nets of the circuit are driven by two sources, one is the original value and the other is simulator output. The final value imposed on the net is determined by the resolution function given in Figure 4.2. The resolution function is illustrated in Figures 4.3 and 4.4.

```
function resolve_std_logic (values: in std_logic_vector)
    return std_logic is
    variable  result: std_logic;
    begin
            if values(0) = values(1)
            result := not values(0);
            end if;

            if (values(0) =  0 && values(1) = 1)
            result = not values(0);
            end if;

            if (values(0) =  1 && values(1) = 0)
            result = values(0);
            end if;

            for index in values’range loop
                    if values(0) = ‘Z’ then
                            result := values(1);
                    elsif values(1) = ‘Z’ then
                            result := values(0);
                    end if;
            end loop;
return result;
end resolve_std_logic;
```

Figure 4.2 Resolution Function

Thus it is clear from the Figure above that when the simulator output is a ‘Z’, the original value of the net is imposed and when the simulator output is a ‘1’, the original value of the net is inverted.

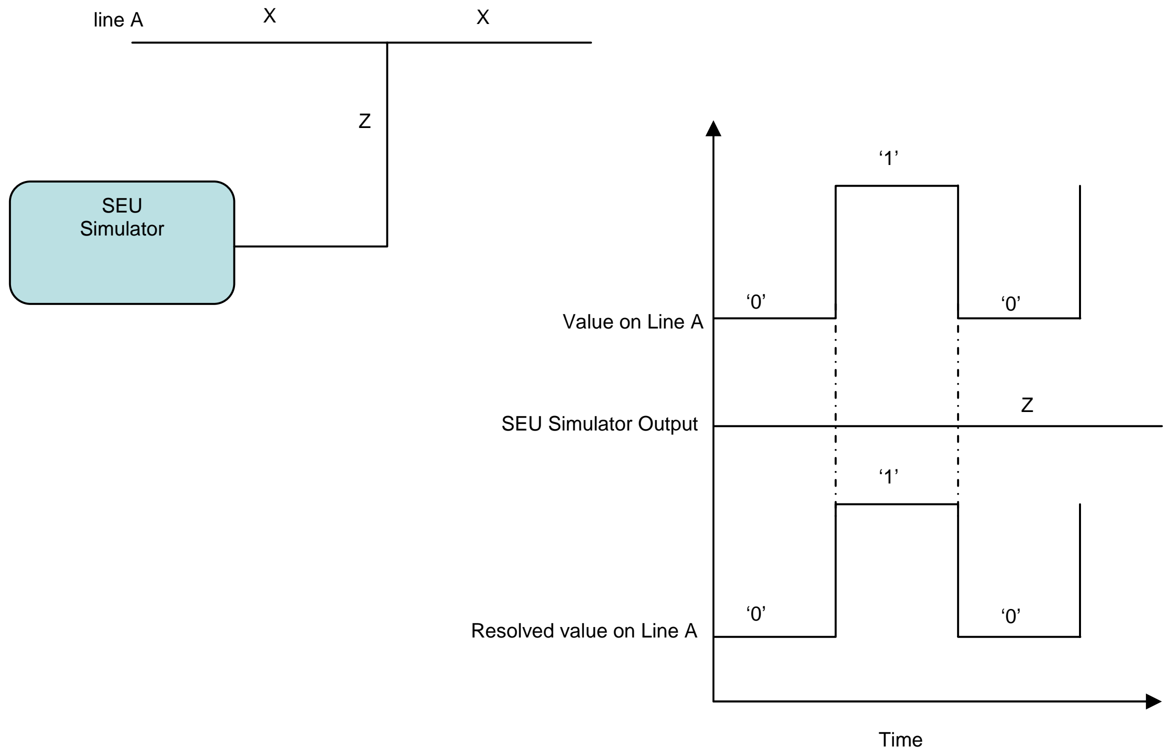

Figure 4.3 Fault Insertion On Line "A" by SEU Simulator

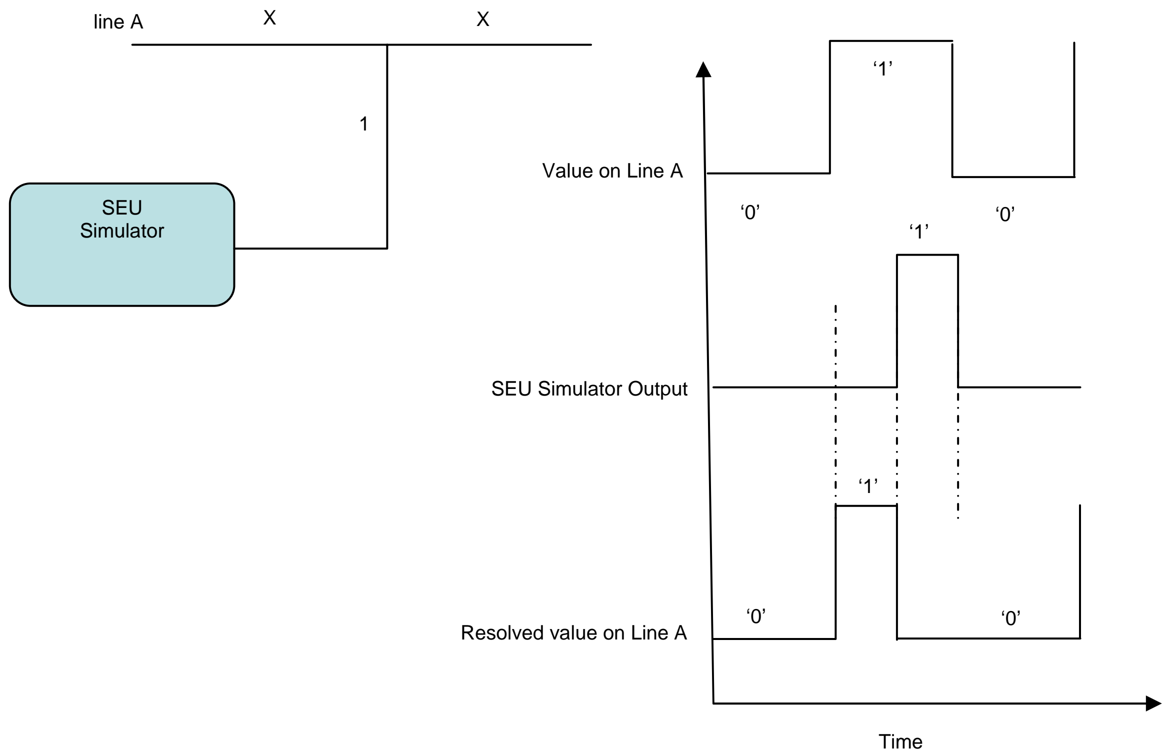

Figure 4.4 Fault Insertion On Line "A" by SEU Simulator

4. Error Calculation: To check the SEU immunity of the reduced circuit, the functional operation of this circuit after introducing errors is compared against the original unfaulted circuit. As discussed above, the reduced circuit is faulted by introducing SEUs using the simulator. If the comparison process leads to a disparity, it indicates that the reduced circuit is unable to compute the correct output. Figure 4.5 shows the steps involved in error calculation.

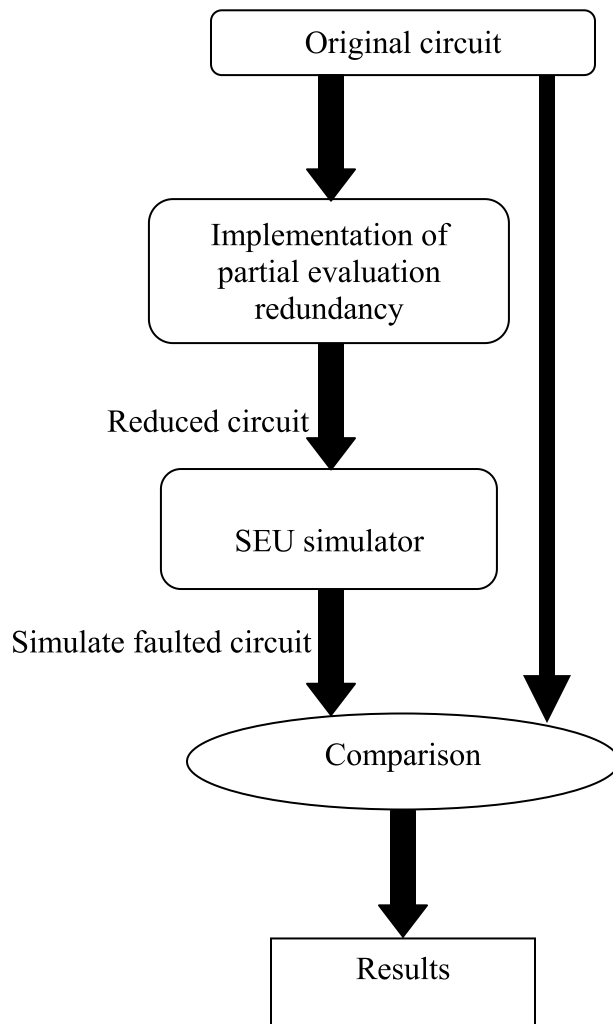

Figure 4.5 Validation of the Technique

4.1 Results

The partial evaluation redundancy technique is tested on different benchmark circuits.

Table 4.1 Results for circuits with positive gains and rounding range as

$0.0 < p < 0.2$ => logic '0' and $0.9 < p < 1.0$ => logic '1'

Name of the circuit	Total number of gates	Number of redundant gates	Total number of outputs	% of redundant gates	$\frac{A_{TMR} - A_{PE}}{A_{TMR}}$ (as percentage)
Count	105	62	1	59.04	33.54
Cm150a	58	32	1	55.17	26.40
9symml	166	73	1	43.97	25.69
Alu2	335	157	5	46.86	24.68
Alu4	674	270	6	40.05	22.87
Too_large	674	241	3	35.75	21.77
I2	161	60	1	37.26	21.14
Cm151a	29	18	1	62.06	20.87
Frg1	100	48	3	48	18.26
Mux	79	29	1	36.70	17.01
C432	209	77	4	36.84	16.17
term1	364	132	9	36.26	14.09
Cordic	64	28	2	43.75	14
F51m	123	65	7	52.84	11.83
Vda	805	362	39	44.96	11.24

Table 4.1 Continued

Results for circuits with positive gains and rounding range as

1.0 < p < 0.2 => logic '0' and 0.9 < p < 1.0 => logic '1'

Name of the circuit	Total number of gates	Number of redundant gates	Total number of outputs	% of redundant gates	$\frac{A_{TMR} - A_{PE}}{A_{TMR}}$ (as percentage)
Cm152a	23	12	1	52.17	9.58
I4	154	59	6	38.31	9.46
C1908	398	196	25	49.24	8.57
I3	106	50	6	47.16	8.18
C2670	677	272	34	40.17	7.56
Z4ml	52	30	4	57.69	5.81
C880	317	159	24	50.15	4.58
Ttt2	204	133	21	65.19	4.16
Cmb	43	28	4	65.11	4.13
X1	307	177	28	57.65	3.87
I9	757	396	63	52.31	3.68
Cm85a	32	20	3	62.5	0.92

Table 4.2 Results for circuits with negative gains and rounding range as $0.0 < p < 0.2$ => logic '0' and $0.9 < p < 1.0$ => logic '1'

Name of the circuit	Total number of gates	Number of redundant gates	Total number of outputs	% of redundant gates	$\frac{A_{TMR} - A_{PE}}{A_{TMR}}$ (as percentage)
c8	162	89	17	54.93	-2.70
My_adder	146	84	17	57.53	-4.94
cm163a	43	26	5	60.46	-6.04
Parity	15	7	1	46.66	-6.12
cm162a	44	24	5	54.54	-8.55
X3	706	296	99	41.92	-20.00
i7	552	153	67	27.71	-22.71
x4	439	158	65	35.99	-25.68

Table 4.3 Results for circuits with positive gains and rounding range as

0.0 < p < 0.3 => logic '0' and 0.8 < p < 1.0 => logic '1'

Name of the circuit	Total number of gates	Number of redundant gates	Total number of outputs	% of redundant gates	$\frac{A_{TMR} - A_{PE}}{A_{TMR}}$ (as percentage)
Count	105	68	1	64.76	36.36
9symml	166	90	1	54.21	32.47
alu4	674	344	6	51.03	29.96
Alu2	335	180	5	53.73	28.87
Too_large	674	311	3	46.14	28.51
Cm150a	58	35	1	60.34	28.08
I2	161	72	1	44.72	25.46
Frg1	100	56	3	56	23.39
C432	209	95	4	45.45	21.30
Cordic	64	33	2	51.56	17.5
term1	364	168	9	46.15	20.47
Vda	805	478	39	59.37	20.26
Mux	79	34	1	43.03	19.91
Cm151a	29	19	1	65.51	19.78
Cordic	64	33	2	51.56	17.5
F51m	123	76	7	61.78	17.38
C2670	677	354	34	52.28	14.99

Table 4.3 Continued

Results for circuits with positive gains and rounding range as

$0.0 < p < 0.3$ => logic '0' and $0.8 < p < 1.0$ => logic '1'

Name of the circuit	Total number of gates	Number of redundant gates	Total number of outputs	% of redundant gates	$\frac{A_{TMR} - A_{PE}}{A_{TMR}}$ (as percentage)
C1908	398	229	25	57.53	13.67
I3	106	59	6	55.66	12.57
I4	154	68	6	44.15	12.55
Cm152a	23	14	1	60.86	10.95
I9	757	486	63	64.20	10.70
C880	317	188	24	59.30	9.83
X1	307	198	28	64.49	7.64
Ttt2	204	144	21	70.58	6.89
Z4ml	52	32	4	61.53	6.39
Cmb	43	30	4	69.76	4.82
Cm85a	32	22	3	68.75	1.85

Table 4.4 Results for circuits with negative gains and rounding range as $0.0 < p < 0.3$ => logic '0' and $0.8 < p < 1.0$ => logic '1'

Name of the circuit	Total number of gates	Number of redundant gates	Total number of outputs	% of redundant gates	$\frac{A_{TMR} - A_{PE}}{A_{TMR}}$ (as percentage)
c8	162	96	17	59.25	-0.72
My_adder	146	88	17	60.27	-3.95
cm163a	43	27	5	62.79	-6.71
cm162a	44	27	5	61.36	-6.57
X3	706	364	99	51.55	-14.71
i7	552	213	67	38.58	-16.63
x4	439	208	65	47.38	-19.53

Table 4.5 Results for circuits with positive gains and rounding range as $0.0 < p < 0.4 \Rightarrow$ logic '0' and $0.7 < p < 1.0 \Rightarrow$ logic '1'

Name of the circuit	Total number of gates	Number of redundant gates	Total number of outputs	% of redundant gates	$\frac{A_{TMR} - A_{PE}}{A_{TMR}}$ (as percentage)
Count	105	82	1	78.09	39.18
9symml	166	108	1	65.06	38.44
Too_large	674	398	3	59.05	37.06
alu4	674	403	6	59.79	35.72
Alu2	335	206	5	61.49	33.95
I2	161	85	1	52.79	30.80
Frg1	100	69	3	69	29.80
Vda	805	600	39	74.53	29.52
Cm150a	58	36	1	62.06	29.21
term1	364	215	9	59.06	28.28
C432	209	117	4	55.98	28.14
Mux	79	43	1	54.43	27.38
Cordic	64	41	2	64.06	25.5
F51m	123	94	7	76.42	24.93
C2670	677	450	34	66.46	23.85
C1908	398	288	25	72.36	22.33
Cm151a	29	20	1	68.96	21.97
I4	154	80	6	51.94	17.48

Table 4.5 Continued

Results for circuits with positive gains and rounding range as

$0.0 < p < 0.4$ => logic '0' and $0.7 < p < 1.0$ => logic '1'

Name of the circuit	Total number of gates	Number of redundant gates	Total number of outputs	% of redundant gates	$\frac{A_{TMR} - A_{PE}}{A_{TMR}}$ (as percentage)
I3	106	66	6	62.26	16.66
I9	757	561	63	74.10	16.64
C880	317	220	24	69.40	15.95
Ttt2	204	169	21	82.84	14.08
Cm152a	23	15	1	65.21	13.69
X1	307	226	28	73.61	13.06
Z4ml	52	36	4	69.23	11.04
Cmb	43	33	4	76.74	8.96
Cm85a	32	24	3	75	5.55
c8	162	110	17	67.90	4.33
My_adder	146	99	17	67.80	0.397

Table 4.6 Results for circuits with negative gains and rounding range as $0.0 < p < 0.4 \Rightarrow$ logic '0' and $0.7 < p < 1.0 \Rightarrow$ logic '1'

Name of the circuit	Total number of gates	Number of redundant gates	Total number of outputs	% of redundant gates	$\frac{A_{TMR} - A_{PE}}{A_{TMR}}$ (as percentage)
cm163a	43	29	5	67.44	-4.02
Parity	15	9	1	60	-4.08
cm162a	44	28	5	63.63	-5.26
i7	552	318	67	57.60	-5.71
X3	706	441	99	62.46	-8.59
x4	439	259	65	58.99	-13.06

Tables 4.1, 4.3 and 4.5 shows circuits that have a positive savings in area while Tables 4.2, 4.4 and 4.6 shows circuits with negative gains. This means that the overhead involved in the technique is greater than the number of gates than can be eliminated. It is observed that, in general, circuits with large number of gates or less number of outputs have positive gains. The number of delay units, majority voters, and multiplexers required increases with the number of outputs. Thus, the area overhead involved in the technique increases with the number of outputs. Hence efficiency of the technique in terms of area decreases with increase in the number of outputs.

By increasing the rounding range, more number of inputs can be given integer values. This increases the number of gates having integer inputs and thus the number of redundant gates increases. Hence, higher savings in area are obtained. However, the disadvantage of increasing the rounding range is that the need to use Temporal TMR

technique increases. This increases the delay overhead. Based on the area and delay requirements, the rounding range can be set to different values. For circuits with less number of gates or less number of inputs, the variation in rounding range has less impact.

CHAPTER 5

CONCLUSIONS

Based on the above results, it can be concluded that the proposed PTMR technique is effective in hardening the FPGA circuits against SEUs. Along with achieving good area savings, the technique is capable of providing 100% SEU immunity. Thus high tolerance can be achieved with less area overhead.

One limitation to this technique is that it is addressed for combinational circuits only. Assuming that a sequential circuit can be modeled as a synchronous state machine with a feedback path consisting of state registers, this technique can be extended to sequential circuits as well.

Further work can be done in this area to improve the efficiency of the technique. For instance, a study can be made to determine the subset of inputs, rounding the values of which, causes more number of gates to be eliminated.

The delay involved in the circuit and the power consumed by it can be evaluated to improve the efficiency of the technique in terms of power and delay.

BIBLIOGRAPHY

[1] S.Whitaker, J. Canaris, and K. Liu, “ SEU Hardened Memory Cells for a CCSDS Reed Solomon Encoder,” *IEEE Transactions on Nuclear Science*, vol. 38, no 6, pp. 1471-1477, December 1991.

[2] K.A. LaBel and M.M. Gates, “Single-Event-Effect Mitigation from a System Perspective,” *IEEE Transactions on Nuclear Science*, vol. 43, No. 2, pp. 654-660, April 1996.

[3] T. Calin, F. Vargas, M. Nicolaidis, and R. Velazco, “A Low-Cost, Highly Reliable SEU-Tolerant SRAM: Prototype and Test Results,” *IEEE Transactions on Nuclear Science*, vol. 42, no 6, pp. 1592-1598, December 1995.

[4] P. Grham, M. Caffrey, D.E. Johnson, N. Rollins, and M. Wirthlin, “SEU Mitigation for Half-Latches in Xilinx Virtex FPGAs,” *IEEE Transactions on Nuclear Science*, vol. 50, No. 6, pp. 2139-2146, December 2003.

[5] J.J. Wang, R.B. Katz, J.S. Sun, B.E Cronquist, J.L McCollum, T.M Speers, and W.C Plants, “SRAM Based Re-programmable FPGA for Space Applications,” *IEEE Transactions on Nuclear Science,* vol. 46, no. 6, pp. 1728-1735, December 1999.

[6] P. Eaton, J. Benedetto, D. Mavis, K. Avery, M. Sibley, M. Gadlage, and T. Turflinger, “Single Event Transient Pulsewidth Measurements Using a Variable Temporal Latch Technique,” *IEEE Transactions on Nuclear Science*, Vol. 51, No. 6, pp. 3365-3368, December 2004.

[7] T. Calin, M. Nicolaidis, and R. Velazco, “Upset Hardened Memory Design for Submicron CMOS Technology,” *IEEE Transactions on Nuclear Science*, vol. 43, no 6, pp. 2874-2878, December 1996.

[8] T. Calin, F. Vargas, M. Nicolaidis, and R. Velazco, “A Low-Cost, Highly Reliable SEU-Tolerant SRAM: Prototype and Test Results,” *IEEE Transactions on Nuclear Science*, vol. 42, no 6, December 1995.

[9] R. Velazco, D. Bessot, S. Duzellier, R. Ecoffet, and R. Koga, “Two CMOS Memory Cells Suitable for the Design of SEU-Tolerant VLSI Circuits,” *IEEE Transactions on Nuclear Science*, vol. 41, no 6, pp. 2229-2234, December 1994.

[10] L.R Rockett, “Simulated SEU Hardened Scaled CMOS SRAM Cell Design Using Gated Resistors,” *IEEE Transactions on Nuclear Science*, vol. 39, no 5, p.p 1532-1541, October 1992.

[11] P.K. Samudrala, J. Ramos, and S. Katkoori, “Selective Triple Modular Redundancy (STMR) Based Single-Event Upset (SEU) Tolerant Synthesis for FPGAs,” *IEEE Transactions on Nuclear Science*, vol. 51, no 5, pp. 2957-2969, October 2004.

[12] J.W. Gambles, K.J. Hass, and S.R. Whitaker, “Radiation Hardness of Ultra Low Power CMOS VLSI.” *11th NASA Symposium on VLSI Design*, May 2003.

[13] K.A. Label, D.K. Hawkins, J.A. Kinnison, W.P. Stapor, and P.W. Marshall, “Single Event Effect Characteristics of CMOS Devices Employing Various Epi-layer Thickness,” *IEEE Conference on Radiation and its Effects on Components and Systems*, pp. 258-262, 1995.

[14] D.G. Mavis and P.H. Eaton, “SEU and SET Mitigation Techniques for FPGA Circuit and Configuration Bit Storage Design,” *Proceedings of the 2000 Military and Aerospace Applications of Programmable Devices and Technologies Conference*, September 2000.

[15] C. Carmichael, E. Fuller, P. Blain, M. Caffrey, and H. Bogrow, “SEU Mitigation Techniques for Virtex FPGAs in Space Applications,” *Proceedings of the 1999 Military and Aerospace Applications of Programmable Devices and Technologies Conference*, 1999.

[16] M. Wirthlin, E. Johnson, and N. Rollins, “The Reliability of FPGA Circuit Designs in the Presence of Radiation Induced Configuration Upsets,” *Proceeding of the IEEE Symposium on Field Programmable Custom Computing Machines*, pp. 133-142, April 2003.

[17] N. McKay and S. Singh, “Dynamic Specialization of XC6200 FPGAs by Partial Evaluation,” *Proceedings of IEEE Symposium on FPGAs for Custom Computing Machines*, pp. 308-309, April 1998.

[18] P. Bernardi, M. Sonza Reorda, L. Sterpone, and M. Violante, “On the evaluation of SEU sensitiveness in SRAM-based FPGAs,” IEEE on-line Testing Symposium (IOLTS), pp. 115-120, July 2004.

[19] F. Sturesson, S. Mattsson, and C. Carmichael, “Heavy Ion Characterization of SEU Mitigation Methods for the Virtex FPGA,” *IEEE Conference on Radiation and its Effects on Components and System*, pp. 285-291, September 2001.

[20] C.C Yui, G.M. Swift, C. Carmichael, R. Koga, and J.S. George, “SEU Mitigation Testing of Xilinx Virtex II FPGAs,” *IEEE Data Workshop on Radiation Effects*, pp. 92-97, July 2003.

[21] E.K. Heironimus and J.G. Tront, “Insertion of Fault Detection Mechanisms in Distributed ADA Software Systems,” *IEEE Conference on System Sciences*, vol. 2, pp. 260-266, January 1989.

[22] A. Benso, S. Di Carlo, G. Di Natale, and P. Prinetto, “Static Analysis of SEU Effects on Software Application,” *IEEE International Test Conference*, paper 18.2, pp. 500-508, 2002.

[23] J.R. Samson, L.D. Torre, P. Wiley, T. Stottlar, and J. Ring, “A Comparison of Algorithm-Based Fault Tolerance and Traditional Redundant Self-Checking for SEU Mitigation,” *IEEE Conference on Digital Avionics Systems (DASC),* vol. 2*,* pp. 8E4/1-8E4/11, October 2001.

[24] S. Singh, J. Hogg, and D. McAuley, “Expressing Dynamic Reconfiguration by Partial Evaluation,” *IEEE Symposium on FPGAs for Custom Computing Machines (FCCM),* pp. 188-194, 1996.

[25] Q. Wang and D.M. Lewis, “Automated Field-Programmable Compute Accelerator Design Using Partial Evaluation,” *IEEE Symposium on FPGAs for Custom Computing Machines (FCCM),* pp. 145-154, April 1997.

[26] J. Hao and J. Chabrier, “Combining Partial Evaluation and Constraint Solving: A New Approach to Constraint Logic Programming,” *IEEE Conference on Tools for Artificial Intelligence,* pp. 494-500, November 1990.

[27] M. Writhlin, E. Johnson, and N. Rollins, “The Reliability of FPGA Circuit Designs in the Presence of Radiation Induced Configuration Upsets,” *Proceedings of the 11th Annual IEEE Symposium on Filed-Programmable Custom computing Machines (FCCM)*, pp. 133-142, April 2003.

[28] M. Mukherjee and R. Vemuri, “A Novel Synthesis Strategy Driven by Partial Evaluation Based Circuit Reduction for Application Specific DSP Circuits,” *Proceedings of the 21st International Conference on Computer Design*, 2003.

[29] C. Carmichael, “Correcting Single Event Upsets through Virtex Partial Configuration,” Xilinx, xapp216 (v1.0) edition, 2000.

[30] C. Carmichael, “Triple Module Redundancy Design Techniques for Virtex FPGAs,” Xilinx, xapp197 (v1.0) edition, November 2001.

[31] P. Brinkley, Avnet, and C. Carmichael, “SEU Mitigation Design Techniques for the XQR4000XL,” Xilinx, xapp181 (v1.0) edition, March 2000.

[32] E. Dupont, O. Lauzeral, R. Gaillard, and M. Olmos, “Radiation Results of the SER Test of Actel, Xilinx and Altera FPGA instances,” *NASA conference*, March 2004.

[33] "Aerospace Products Radiation Policy,"
http://www.atmel.com/dyn/resources/prod_documents/doc4170.pdf

[34] E. Visser, "Partial Evaluation,"
http://catamaran.labs.cs.uu.nl/twiki/st/pub/Ist/ProgramTransformation/IST-PT2.pdf

[35] V. Tang, J.L.D Lam, L. McMurchie, and C. Sechen, "High-Performance SEE-Hardened Programmable DSP Array," *International Conference on Military and Aerospace Applications of Programmable Devices*, 2004.

[36] Praveen K. Samudrala, "Selective Triple Modular Redundancy (STMR) Based Single Event Upset (SEU) Tolerant Synthesis for FPGAs".

[37] M. Writhlin, N. Rollins, M. Caffery, and P. Graham, "Hardness by Design Techniques for Field Programmable Gate Arrays".

[38] "Radiation Hardening by Space",
http://www.mse.vt.edu/faculty/hendricks/mse4206/projects97/group02/space.htm

[39] "Single Event Effects", http://www.eas.asu.edu/~holbert/eee460/see.html

[40] "A comprehensive method for the evaluation of the sensitivity to SEUs of FPGA-based applications,"
http://klabs.org/DEI/References/Radiation/velazco_fpga_seesymp02.ppt#256,1

[41] http://www.sandia.gov/media/rhp.htm

[42] "Hardening Devices Against Radiation,"
http://www.mse.vt.edu/faculty/hendricks/mse4206/projects97/group02/hardening.htm#soi