Variation Tolerant Hierarchical Voltage Monitoring Circuit for Soft Error Detection

by

Ashay Narsale

Submitted in Partial Fulfillment of the Requirements for the Degree

Master of Science

Supervised by

Professor Michael C. Huang

Department of Electrical and Computer Engineering
Arts, Sciences, and Engineering

University of Rochester
Rochester, New York

2008

Curriculum Vitae

Ashay Narsale was born in Mumbai, India. He received his Bachelors Degree in Electronics Engineering from Mumbai University, India in 2006. He is currently pursuing a Master's Degree in Electrical Engineering from University of Rochester. He is part of the Advanced Computer Architecture Laboratory in the university. His research interests include digital circuit design, computer architecture and circuit reliability.

Acknowledgement

It is my pleasure to thank the people who made this thesis possible. First of all, I would like to thank my advisor, Prof. Michael Huang who showed a lot of enthusiasm as well as patience with my work. Throughout my thesis work, he provided me encouragement, sound advise and lot of good ideas for which I am extremely grateful. I would also like to express my thanks to Srinivas Vaidynathan and Dr. John Liobe for their timely help and discussions. Lastly and most importantly. I would like to thank my parents for their unconditional support.

Abstract

As device feature size continues to scale down to the nanometer regime, the decreasing critical charge fundamentally reduces noise margins of devices and in turn increases the susceptibility of the ICs to external noise sources such as particle strikes. While protection techniques for memory such as ECC are mature and effective, protections for logic errors remain imperfect. Full-blown redundancy solutions for microprocessors such as mirrored cores and triple-modular redundancy incur significant overhead and are clearly limited to the niche market of mission-critical servers. The fundamental inefficiency of such redundancy lies in the repetition of all operations to detect the discrepancy caused by events much rarer than cycle-to-cycle activities. Clearly, for the vast majority of general-purpose systems, a detection mechanism that has low standby energy consumption is called for. In this thesis, we propose a circuit-level solution to detect errors by monitoring the supply rail disturbance caused by a particle strike. Combined with checkpointing and rollback support, such a circuit can provide a high level of protection against particle-strike induced soft errors. At 17%, the power overhead of the design is reasonable and much lower than prior art. The design is also tolerant to process, voltage, and temperature (PVT) variations and to power supply noises.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In computer systems, an error is a wrong electrical value at the output due to an erroneous design or a breakage in one of the system components. However, a soft error or single event upset (SEU) is also an erroneous output but not due to the above mentioned reasons. It is a random occurrence that can be caused due to a cosmic particle strike on the system resulting in an erroneous value at the output. However, this does not mean that the system is less reliable than before. It is a naturally occurring rare phenomenon which causes a temporary bit flip which, if latched, generates an error. Soft errors are caused by cosmic particles like alpha particles, neutron and protons. Whenever a charged particle strikes a chip, it generates an ion track in the substrate. These free electron-hole pairs can diffuse near the channel to flip the state of the bit. The study and mitigation of soft errors can be broadly divided into two parts, applications at sea level and at higher altitudes. At higher altitudes, systems come in contact with more error causing cosmic particles resulting in both soft and hard errors. Also, reliability of space applications is crucial and hence requires extensive error protection. The flux of cosmic particles at the sea level is much less resulting in much fewer errors. However, with reducing device size and supply voltage, the soft error susceptibility of ICs has increased.

Memory elements can be effectively and efficiently protected via information redundancy. Modern systems routinely employ error correcting codes (ECC) to protect their memory elements [13]. At a modest cost in performance and area overhead, ECC protects memory against a large majority of memory soft errors. However, studies have projected that as technology continues to scale, soft error rate (SER) of combinational logic will continue to rise and become comparable to, and eventually more severe than, that of memories [14, 15]. Clearly, error detection and correction in combinational logic will become increasingly crucial in providing high overall system integrity.

Unfortunately, dealing with errors in logic elements proves much less convenient. While error coding can be applied to protect arithmetic operations [16, 17], ALUs occupy only a small percent-

age of the transistor budget in modern microprocessors, whose logic is heavily devoted to execution control and orchestration. To date, the most practical approach to protecting general logic against soft errors remains brute-force replication. This can be done at the transistor level [18, 19] or at the architecture level [20]. Both are less than ideal solutions: transistor-level solutions have a number of fundamental limitations [21]; duplicating an entire core [20] (or worse, triplicating [22]) is obviously a very expensive proposition that would most probably remain a niche-market solution for mission-critical systems. Ironically, such heavy-handed redundancy is only necessary for error *detection* – in contrast, in the case of memory, soft error detection is almost trivial – once an error is detected, we can easily roll back the system to an early checkpoint, and retry the computation. Checkpointing and rollback are among the earliest and the most studied topics in fault tolerance [23–25].

To tackle the issue of efficient soft error detection in logic, in this thesis, we propose a novel hierarchical soft error detection circuitry which monitors the ground voltage to detect the pulses as a result of particle strike-induced switching. To avoid the impact of natural noises on the ground line, we decouple the ground terminal of a functional block from the ground bus and monitor this ground terminal of the functional block for errors. This approach increases detection sensitivity and also gives the designer the flexibility to choose vulnerable areas of the chip to monitor. Compared to schemes that monitor the bulk current [9], our design does not require additional routing nor a substantial area and power overhead. Our design has a low overhead of about 17-18% in area and power consumption and can also be used to monitor memory arrays, simplifying or improving the overall protection mechanism. The proposed scheme is also tolerant to PVT variations and power supply noises. The rest of the thesis is organized as follows. Chapter 2 gives a brief overview of passage of cosmic particles through silicon resulting in an error. Chapter 3 discusses the various mitigation techniques proposed earlier. In Chapter 4, we discuss the proposed scheme. Chapter 5 concludes the thesis.

Chapter 2

Causes of Soft Errors

The main cause of soft error is due to cosmic particles hitting the surface of the chip. These cosmic particles are usually in the form of alpha particles and neutrons.

2.1 Alpha Particles

2.1.1 Source

Table 2.1: Natural emission rates of processed and packaged materials [1]

Material	Emission Rate [$\alpha/(cm^2$ hr]
Cu metal	0.0019
Al metal	0.0014
Fully processes wafer	<0.001
Mold compound	0.024 - <0.002
Flip-chip underfill	0.002 - 0.009
Pb-based solder	7.2 - <0.002
Package	0.01 - 0.001

The alpha particles are doubly charged helium nucleus with 2 protons and 2 neutrons. Alpha particles are released from elements of high atomic number during radioactive decay and they immediately interact with other atoms. Alpha particles cannot travel large distance in a material. So the main source of alpha particles is not the atmosphere but an IC itself. Packaging and solder bumps contain traces of radioactive isotopes, which emit alpha particles in addition to other particles like gamma and beta particles, when they decay to a lower state. Table 2.1 gives the emission rate of alpha particles from various constituents of a chip. The kinetic energy of alpha particles emitted

is between 1 to 9 MeV. The most common radioactive isotope found in packaging is Th-232 and U-238 isotopes. In solder, the alpha particles are emitted due to Po-210 and Pb-210 impurities [3].

2.1.2 LET

Linear Energy Transfer (LET) is the measure of energy that is transferred in the material when an ionizing particle passes through it. When alpha particles with a specific kinetic energy travel through silicon, they transfer their energy to the silicon along their path and induce electron hole pairs through electric interaction. To create an electron hole pair, an alpha particle loses about 3.6eV per pair. Thus a 4 MeV alpha particle traveling through silicon will generate more than a million electron hole pairs. As the alpha particle travels through the silicon, it loses kinetic energy to silicon leading to a decrease in its velocity in silicon. As the alpha particle loses its velocity in silicon, it deposits more energy to the silicon since it takes more time to travel through the same distance than before. Hence the deposited energy is maximum at the end of the alpha particle trajectory. The amount of charge generated when a charged particle travels through a material can be denoted by dQ/dx which is the differential charge. Fig. 4.1 shows the differential charge generated by an alpha particle as a function of its kinetic energy. As the kinetic energy of the particle increases, it can travel more distance in the material and hence the dQ/dx decreases as charge generated gets distributed over a greater distance.

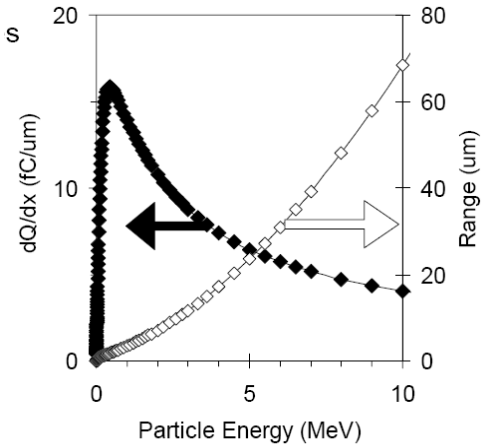


Figure 2.1: LET curve [4]

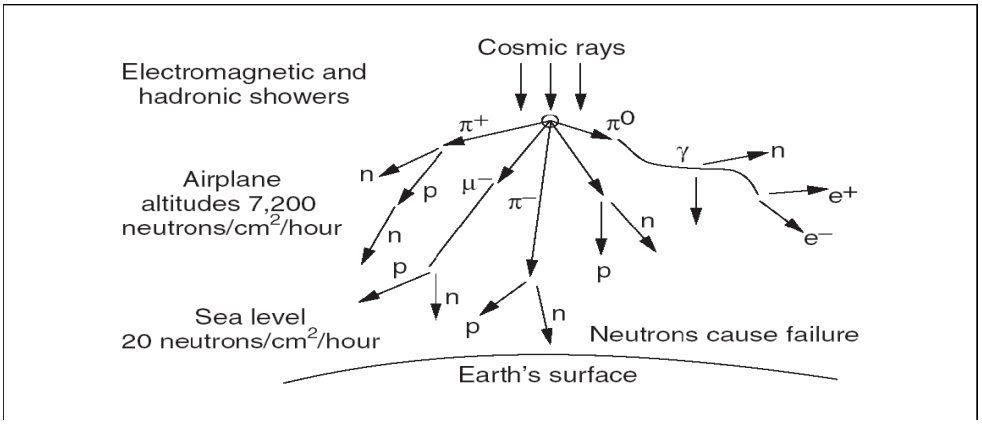


Figure 2.2: Cosmic ray disintegration, causing a cascade of nuclear reactions [3]

2.2 High Energy Neutrons

2.2.1 Source

Primary cosmic radiation consists of galactic winds (<1 GeV) and solar winds (> 1 GeV). These two types of cosmic radiation include protons (92%), alpha particles (6%) and other heavy ions (2%). These primary particles react with earth's atmosphere easily resulting in a secondary and higher generation particles.The second generation particles consist of protons, electrons, neutrons, muons and pions. Protons, electrons, muons and pions do not get past the earth's atmosphere since protons and electrons are affected by coulomb interaction with earth's magnetic field while muons and pions are short-lived particles. Thus at sea-level the proton flux is 5% of original while the electron and neutron flux is 1% of the original. All three particles can cause soft error, however charge generation property of neutron for the same energy is more than proton or electron. Fig. 2.2 denotes the cosmic neutron flux per unit time at sea-level as a function of neutron kinetic energy. Neutron flux varies as much as 30% as a result of solar cycles. The flux of neutrons with kinetic energy above 1 MeV above sea-level is about 25 neutrons/cm^2-hr. Neutrons with energy above 5 MeV usually cause soft errors in silicon devices [3].

2.2.2 Interaction

Since a neutron does not carry any charge, it does not induce ionization in a material by itself. A neutron interacts with the material (silicon) atoms causing a nuclear reaction which gives out charged particles as its products. These charged particles cause ionization, hence soft error in the device. A neutron can have elastic and inelastic interaction with nucleus of a material. If the particles collide and separate without changing its original structure, the process is called an elastic

Table 2.2: Interaction of fast neutrons with silicon [2]

Reaction	Kinetic energy [Mev]
n + Si $\rightarrow$ ^{25}Mg + α	2.75
n + Si $\rightarrow$ ^{28}Al + p	4.00
n + Si $\rightarrow$ ^{27}Al + d	9.70
n + Si $\rightarrow$ ^{24}Mg + n + α	10.34
n + Si $\rightarrow$ ^{27}Al + n + p	12.00
n + Si $\rightarrow$ ^{26}Mg + ^{3}He	12.58
n + Si $\rightarrow$ ^{21}Ne + 2e	12.99

collision (transfer of energy). Scattering processes that involve nuclear excitation by the incident neutron are known as inelastic reactions. Usually the reaction of neutron with semiconductor results in a spallation reaction where one or more charged particles are created. The LET of a spallation fragment with a greater charge is larger than that of a fragment with lesser charge with the same energy. This is the case since, the larger fragment moves slowly within the semiconductor and hence transfers more energy to the semiconductor. Table 2.2 shows the interaction of a neutron with silicon which results in the emission of an α, β, proton and other ionized particles.

2.3 Thermal Neutrons

Table 2.3: Interaction of neutron neutrons with silicon [3]

Reaction	α-energy [Mev]
n + ^{10}B $\rightarrow$ α + ^{7}Li	1.47;0.84
n + ^{28}Si $\rightarrow$ α + ^{7}Li	1.78;1.02

Thermal neutrons are low energy neutrons with a kinetic energy of about 0.025 eV (approx. 2.4 MJ/kg, hence a speed of 2.2 km/s) which is the most probable energy at a temperature of 290 K (17 °C). After a number of collisions with nuclei (scattering) in a medium (neutron moderator) at this temperature, neutrons arrive at about this energy level, provided that they are not absorbed. Thermal neutrons have a different and often much larger effective neutron absorption cross-section for a given nuclide than fast neutrons, and can therefore often be absorbed more easily by an atomic nucleus, creating a heavier - and often unstable - isotope of the chemical element as a result. In semiconductors, boron is used as a dopant and also used in the manufacturing process. The probability of thermal neutrons interacting with boron is the highest and it goes on decreasing with increasing

kinetic energy. Natural boron contains two isotopes, 80% ^{10}B and 20% ^{11}B. ^{10}B is highly unstable when exposed to neutrons. By absorbing a neutron, the ^{10}B atom breaks (nuclear fission) into a lithium atom and an alpha particle. The first one is the dominant reaction with a probability of 94%. Both the lithium and alpha particle can cause error. The lithium particle and alpha particle are emitted in the opposite direction according to conservation of momentum principle and hence there is little chance of them traveling in parallel in the device. Hence, the probability of these particles hitting a sensitive area in the device is high. Lithium recoil has a maximum charge generation rate of 25fC/μm. Thus compared to alpha particles lithium recoil has a greater chance of causing a soft error. Table 2.3 shows the thermal neutron reactions with boron. Boron is used as a p-type dopant and also in BPSG (boron phospho-silicate glass) layers. The concentration of boron used in BPSG is much greater than that used in diffusion and implant layers as a result of which BPSG was a dominant source of soft errors (in 0.25μm and above technologies). The solution to this problem is to exclude BPSG from the design flow. However, BPSG is used to improve the step coverage and contact reflow at lower processing temperature. If the use of BPSG is a must, then enriched boron ^{11}B is used instead of ^{10}B. BPSG if used near the substrate will most probably cause errors since a lithium recoil or alpha particle can travel utmost 3um in the device. Hence it is a good idea just to replace/remove BPSG layer near the substrate so that even if a particle strike occurs, it does not produce an error. Fig. 2.3 shows the placement of the BPSG compound in a chip. Table 2.4 shows that the soft error rate (SER) due to thermal neutron reduces to zero in the absence of BPSG [1].

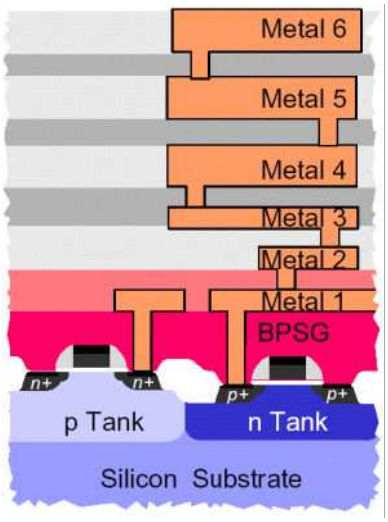


Figure 2.3: BPSG model [5]

Table 2.4: Comparison of the contributions to the SER of SRAMs from two production technologies [1]

Contribution	0.25 μm (with BPSG)	0.18 μm (no BPSG)
Alpha particles	4%	18%
High-energy neutrons	15%	82%
10B fission	81%	0%
Total SER (a.u)	7.5	1.0

2.4 Physical Mechanism

When a ionizing particle strikes a semiconductor device, the following three mechanisms take place in the device.

1.E-h pairs created in the depletion region and have not recombined will separate due to the external field applied.

2.Funneling effect takes place wherein the charge generated near the depletion region is collected by the depletion region due to the funneling phenomenon.

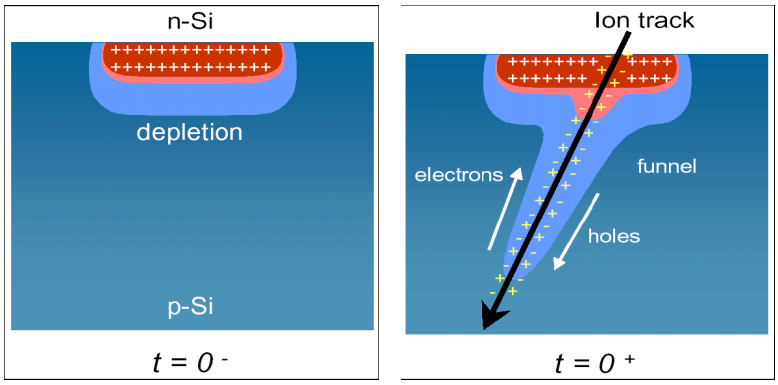
3.The remaining charge generated in the device will drift to the depletion region due to diffusion.

A charged particle (alpha particle or a recoil) striking a device travels in a straight line. The energy of the particle is transferred to the device by coulomb forces. This leads to the creation of huge number of electron-hole pairs as the energy given to the device generates an electron-hole pair for every 3.6eV of energy transferred to the device. Thus an ionization particle striking the device generates a column of highly conductive, neutral and localized electron-hole pairs in a matter of a few pico-seconds. Initially the radius of the column is around 1μm while charge density is 10^{19} to 10^{21} cm^{-3}. There are 2 charge collecting mechanisms: 1. Charge Funneling, 2. Diffusion.

2.4.1 Charge Funneling

The charge funneling as a method of charge collection in reverse-bias junctions was first introduced by Hsieh [6].When an ionization particle strikes a reverse biased p-n junction, the charge generated by the passage of the particle disturbs the electric fields of depletion region. Two separate processes follow after this. In the first stage, the depletion region collapses and in the second stage, it recovers. The reverse biased depletion region pulls up the electrons and pushes the holes down as shown in Fig. 2.4, causing a charge separation. Before this, the column of the charge generated is neutral because equal number of electrons and holes compensate each other. The shifting of these charge carriers causes a net shift in the charge of the depletion region and hence the potential difference

across it reduces. The amount of potential drop created depends on the number of e-h pairs generated by the ionizing particle. The equi-potential lines of the depletion region spread down the substrate and envelope the whole length of the track. Thus this extending electric field helps to collect charge that was deposited below the original depletion region. This phenomenon is called funneling. The second stage, which is the recovery stage, takes a longer time than the first stage (collapse) since a complete recovery only occurs after all the holes are pushed out of the depletion region. Funneling strongly depends on the substrate doping. Substrate with a lower doping concentration shows a slower field distortion, but greater charge collection. Lower doping means that the substrate has higher resistivity due to which the time required for the holes to be pushed out of the depletion region is more. Hence the depletion region can collect more charge in the process. If the particle strike happens between the two n+ regions, then resulting funneling action can collect enough charge to turn the transistor on, resulting in the change of state, thus leading to a soft error.



(a) Before the particle strike (b) Funnel generation

Figure 2.4: Charge generation and collection after a hit [6]

2.4.2 Diffusion

Diffusion is a process where in the charge generated by the ionization particle moves towards the depletion region. This process takes a longer time than funneling in the orders of several nanoseconds. Initially, the electrons and holes diffuse together keeping the device electrically neutral. Most of the e-h pairs deeper in the substrate diffuse away, some may be captured by the sensitive node leading to an error or glitch.

Chapter 3

Process of Error Mitigation

3.1 Device (Process) Level

Device level techniques modify the original fabrication process of the chip to make it resilient to soft errors. The following are a few process level techniques used to mitigate soft errors :

3.1.1 Selection of material

Soft errors are caused by alpha particles or neutron striking the substrate. Alpha particles are usually emitted by the materials/compounds used in packaging. Table 2.1 shows the probability of an alpha particle being emitted by various components used in packaging. Thus probability of an alpha particle emission is material dependent and can be minimized by selecting the appropriate material. For example, as previously discussed the removal of BPSG layer near the substrate completely eliminated the SER due to thermal neutrons in SRAM.

3.1.2 Boosting node capacitance

As technology scales, the device size is reducing, thus leading to smaller feature size per generation. This leads to a reduction in the critical charge Q_{crit} of the device leading to more number of SEUs. This mitigation technique focuses on increasing the node capacitance and hence the Q_{crit} of the device to reduce SEUs. This technique is mostly used in memories. Most common implementation is the SDRAM metal in metal (MIM) capacitance, DRAM capacitance on top of memory cell and trench DRAM capacitance. The shortcomings of this technique are the extra delay and power burned due to the additional capacitance. Fig. 3.1 shows that FIT (Failure in Time) reduces as the junction capacitance of the sensitive nodes Q_{crit} increases [7].

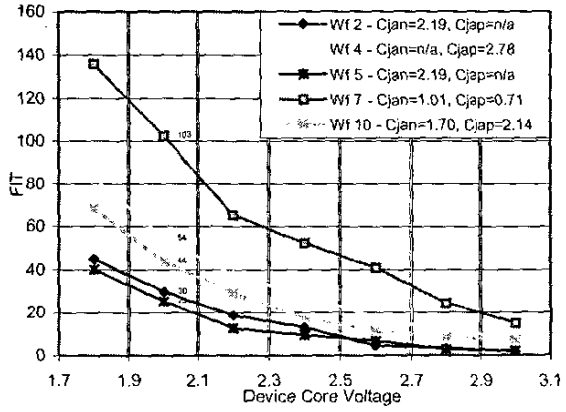


Figure 3.1: The measured results of alpha particle caused failure rate per 412KB [7]

3.1.3 Silicon on Insulator (SOI)

The common structure of SOI technologies is full dielectric isolation between individual transistors. This structure leads to the following advantages over the bulk devices:

a. It avoids latchup by suppressing the parasitic npnp thyristor structure.

b. By reducing the silicon volume in which radiation generates charge carriers, it reduces the sensitivity to radiation.

c. No funneling can occur since depletion region is isolated from the substrate by a dielectric [8].

Fig. 3.2 shows a SOI CMOS device structure which isolates adjacent transistors using a dielectric. The SOI device dramatically reduces the area for collected charge.

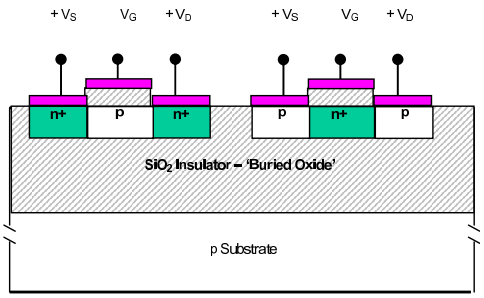


Figure 3.2: SOI CMOS device structure [8]

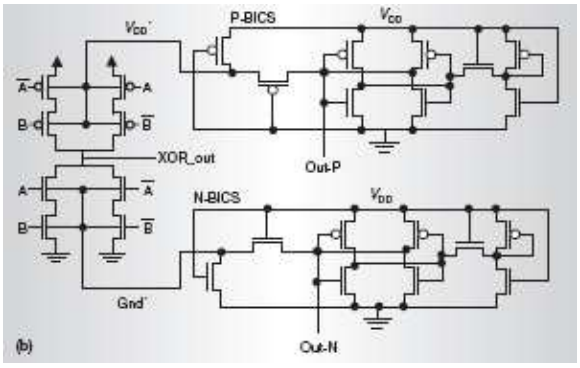


Figure 3.3: SEU detection scheme based on monitoring the substrate current [9]

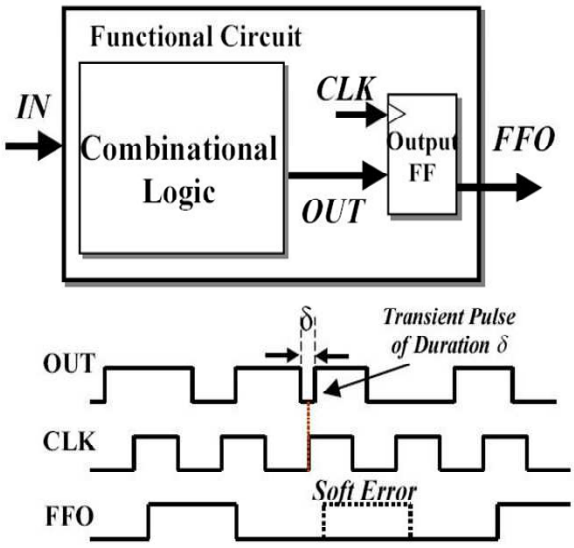


Figure 3.4: Hierarchical Scheme monitoring latch input and output [10]

3.2 Circuit/Block Level Techniques

3.2.1 Current Signature

Most of the concurrent SEU detection schemes using built-in current sensor (BICS) have been proposed for memories. However, as SEUs in combinational circuits become more critical, the need for a detection circuit that can detect SEUs in both combinational and memory circuits is gaining importance. [9] have proposed such a scheme which monitors the bulk current to detect a soft error. During normal operation, the bulk current has a nominal value corresponding to the leakage current in the substrate. When a transient strike occurs, the bulk current increases due to the charge generated by the particle and hence shows a change in its value, which can be detected by the detection circuitry. Fig. 3.3 shows the bulk-BICS scheme implemented for an ex-or gate.

However, this scheme detects all the particle strike even though the particle strike may not lead to error, causing a false positive. A hierarchical architecture and technique for soft error detection based on current sensing is proposed in [10]. This scheme monitors the inputs and outputs of the flip flops in the sequential logic to detect soft errors. Fig. 3.4 shows the hierarchical architecture of this scheme. [26] and [27] propose two different detection circuits to detect a transient upset in memories by monitoring the supply current.Using different detection circuits, they monitor supply and ground buses of the memory to detect a glitch caused by the particle strike.

3.2.2 Radiation Hardened Logic Families

Circuits can be made SEU resistant by adding extra hardware redundancy. Basic circuit blocks like NAND, NOR gates, flip flops etc. can be made SEU resistant by adding more transistors than required. By using this redundant hardware, the gate is made to work in spite of a soft error occurring. A library of such circuits with redundant logic to protect them against soft errors is known as Rad-Had libraries. This technique is widely used in rad-had designs since the designer can use blocks(gates, flip flops) from the already designed library to make the circuit SEU-resistant. Thus the designer has the advantage of controlled cost and efficiency. However this technique does not scale with technology and hence is technology dependent. Whitaker design [28–30], Dice design [31], HIT cells [32], and Barry-Dooley design [33, 34] are some of the SEU tolerant SRAM cells that exist in the literature.

3.3 Block Level Techniques

3.3.1 Triple Modular Redundancy

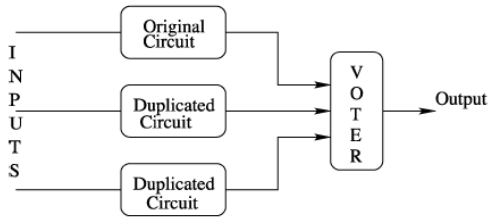


Figure 3.5: TMR block diagram [11]

Triple Modular Redundancy (TMR) is a fault tolerant form of N-modular redundancy, in which three systems perform a process and that result is processed by a voting system to produce a single output as shown in Fig. 3.5. If any one of the three systems fails, the voter accepts the majority decision and masks the fault. The main disadvantages of TMR are its area and power overhead and

reliability of the voter. The hardened design has 200% more design than the origins design. The reliability of the voter is also an important issue since if the voter fails then the complete system will fail. Also TMR cannot detect more than one error due to SEU since the voter will select the error prone data as its output. However, in a TMR system the voter is much more reliable than the other TMR components. TMR is widely used since it is one of the most reliable methods against SEU and is easy to implement [11].

3.3.2 Error Detection and Correction Codes

Error detection and correction (EDAC) algorithms are widely used to protect memories. Typically every bit of memory is refreshed periodically (at least 15 times in 1 sec). When the refreshing operation takes place, EDAC codes are used continuously to check for errors in the memory [13]. Reliability is proportional to refreshing frequency since higher refreshing frequency means that memory will be checked more number of time for error. Usually, hamming code is used to correct 1 bit and detect 2 bit errors in the system. Even though a single cosmic ray can upset many physically neighboring bits in a DRAM, such memory systems are designed so that neighboring bits belong to different words, so such single event upsets (SEUs) cause only a single error in any particular word, and hence can be corrected by a single-bit error correcting code. As long as no more than a single bit in any particular word is hit by an error between refreshes, such a memory system presents the illusion of an error-free memory. When using EDAC, the memory becomes slower, as EDAC is implemented in the critical path.

3.4 Architectural Level

3.4.1 Watchdog Processor

A watchdog processor is a simple processor used to perform concurrent system level error detection by monitoring the behavior of the main processor. Fig. 3.6 shows the organization of a system using a watchdog processor. Error detection using watchdog consists of two stages. The first stage is known as the setup stage in which the watchdog processor is given information about the processor and the processes to be checked. In the second stage, it concurrently monitors the main processor and collects the necessary information needed to check for an error in the main processor [12].

3.4.2 Multi-threading Mitigation

Multiple copies of the same process can run on a multi-threaded processor and the processes can be checked with each other at well defined checkpoints. Fig. 3.7 shows the flowchart of multi-thread

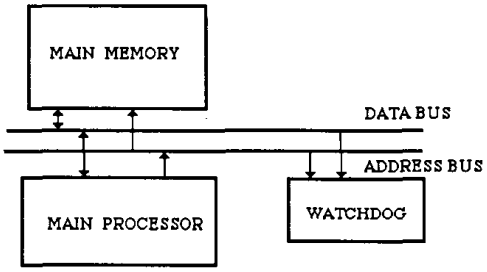


Figure 3.6: Watchdog block diagram [12]

operation to detect SEU. The threads work on the same data and execute the same operation and compare their results at some defined checkpoints. A SEU will cause result mismatch, so that the system can be rolled back to the previous checkpoint to correct the error [35].

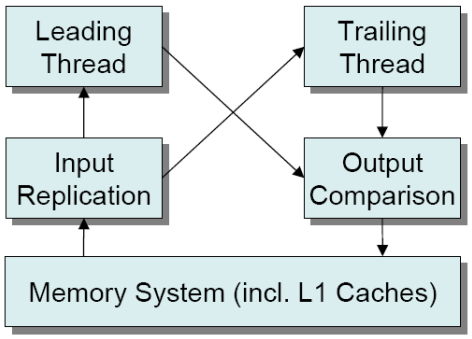


Figure 3.7: Multi-threaded block diagram [5]

Chapter 4

Hierarchical Error Detection Scheme

4.1 Basic Principle of Current Monitoring

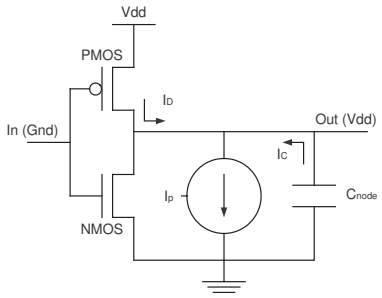


Figure 4.1: Modeling a particle strike on an inverter [9].

A charged particle striking a sensitive node in a CMOS circuit can be modeled as current flowing between the reverse biased p-n junction of the NMOS transistor [9]. Fig. 4.1 shows a particle striking the NMOS transistor of the inverter and the current source I_p acts as the supplicant of the current generated by the particle strike. At the instance of the strike, the 'OFF' NMOS transistor is turned 'ON' for the short duration of the transient pulse, creating a temporary path from V_{dd} to ground. Therefore the ground current at the time of the particle strike, consists of I_d and the node current I_c since the output capacitance discharges through the NMOS to change the state of the output from logic 1 to 0. Similarly, a particle striking the PMOS, turns it 'ON' for the duration of the particle strike, creating a short circuit path between V_{dd} and ground. However in this case, the drain current I_d is used to charge the output capacitance to change the output of the circuit from logic 0 to 1. Therefore, it is observed that whenever a particle strike occurs, there is a conduction path created between V_{dd} and ground giving rise to short circuit current. However, a particle strike is not the only cause for short circuit current. When a change in inputs results in a change in the

output of a functional block, at that instant there exists a conducting path between V_{dd} and ground, leading to short circuit current. However, the short circuit current caused by a particle strike tends to be greater than that caused by the changing inputs mainly due to two reasons. First, the striking particles generate more charge carriers in the device due to the ionization process which causes more current to flow through the device. Second, the switching speed of the CMOS devices is very high, causing less amount of switching current. Therefore, we can detect a soft error in a functional block by detecting the current spike caused by the soft error.

Finally, we note that like any other protection mechanism, this is not a fail-safe guarantee to detect all errors. We see this as a relatively inexpensive mechanism to increase general-purpose systems' tolerance to SEUs. As we will see later, this design can be tuned to balance detection sensitivity and performance consequences.

4.2 Hierarchical SEU Detection Circuit

In combinational logic, a large number of gates switch concurrently, creating a huge transient current. Hence, a detection circuit which is connected to the supply rails of a block of combinational logic results in a high voltage drop. Further, noise in the power supply rails make soft error hard to detect [26]. To overcome this problem, we propose a hierarchical structure. Instead of monitoring the supply rails of a whole block of gates, we monitor supply voltage at each smaller functional block as shown in Fig. 4.2.

The detection circuitry has two levels of voltage comparators. The first level compares the ground voltages of the functional blocks, while the second comparator amplifies the error signal. This approach has the following advantages: (a) The transient voltage produced at the supply rail due to simultaneous switching action of gates is reduced to a very low value since we are monitoring individual functional blocks. Hence, any distortion in the supply voltage due to an SEU can be detected. (b) The circuit designer has greater flexibility in choosing the functional blocks to be monitored for soft errors. This is useful since not all the errors will affect the architectural state. (c) The hierarchical approach allows the designer greater control over the sensitivity of the detection circuit.

4.2.1 Error Detection in Combinational Logic

In our design, only the ground voltage is monitored to detect error. For that purpose, a single NMOS is connected between the ground bus line and ground terminal of the functional block. The addition of this transistor helps to separate the ground bus from the functional block ground terminal, thus creating a 'virtual ground' (GND') at the ground terminal of the functional block. The voltage

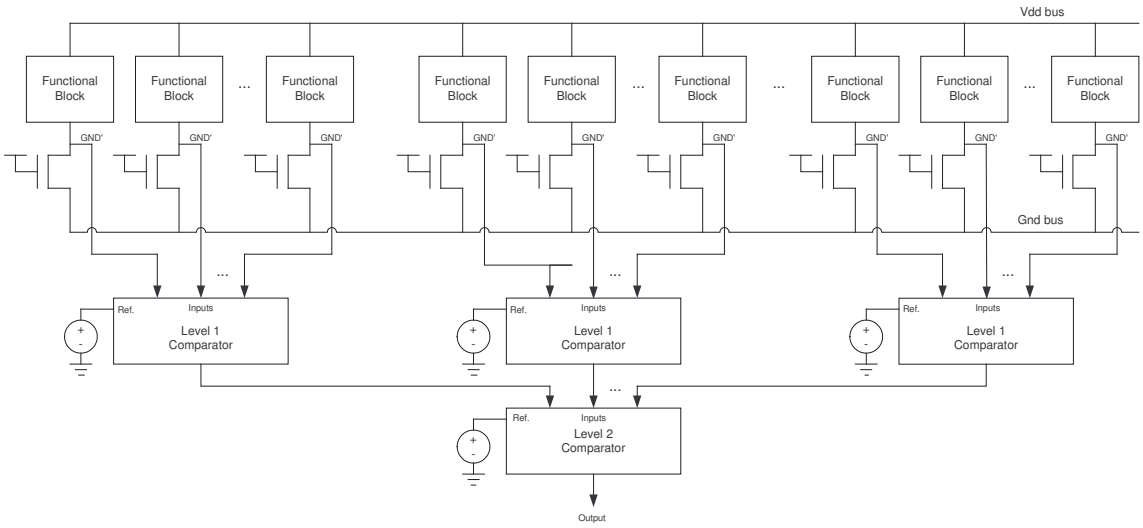


Figure 4.2: Proposed Hierarchical Error Detection Scheme

fluctuations at this GND', which reflects the switching voltage generated by the functional block, can be monitored to detect an error. The GND' terminals of the individual functional blocks are supplied as inputs to the voltage comparator. This voltage will have transient switching noise as well as the spikes generated due to an SEU. The comparator rejects switching noises and amplifies spikes generated by SEU. To achieve this objective with high success rate, the threshold voltage has to be set just above the switching noise level, but below the level of the minimum SEU-induced spikes that we want to detect. As a result, the number of inputs that can be fed into the first-level comparator depends on the switching activity of the functional block.

Whenever the voltage of any input GND' terminal rises above the set threshold, the comparator gives a positive output as the input to the second-level comparator. This comparator serves two purposes. First, it amplifies the weak error signal of the first-level comparator to give a full swing output which denotes an error being generated in the system. Second, all the outputs from the first level comparator can be simultaneously compared to detect an error so that the whole combinational logic block can be monitored. Therefore, a hierarchical approach can be used to detect soft error in any combinational logic irrespective of its size or functionality.

Detection mechanism

An important question in monitoring the combinational logic is which functional blocks to protect against soft error. A particle strike in a combinational logic creates a glitch which if latched results in a soft error. However, not every particle strike which creates a glitch gets latched. We use

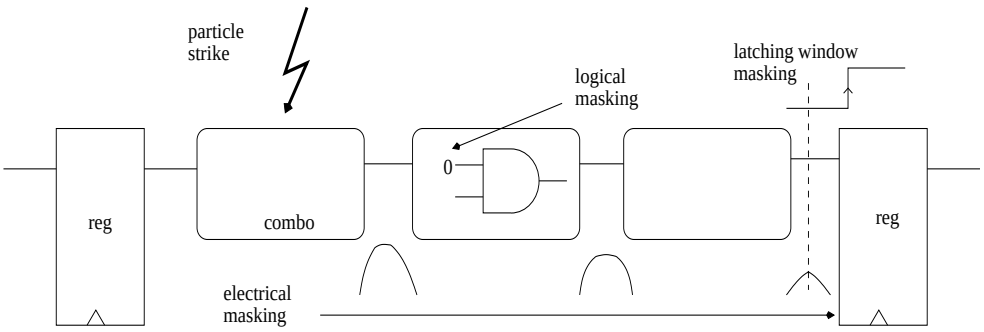


Figure 4.3: Masking in combinational logic

the logical and timing masking properties of the combinational circuit, as shown in Fig. 4.9, to determine which gates/functional blocks to monitor. Thus instead of monitoring every gate for a soft error which would increase the cost of the system tremendously, we monitor the gates which have the most probability of causing a particle generated glitch to be latched.

Implementation

Fig. 4.4 shows the detection circuit used to detect soft error in two ex-or gates.

The detection circuit consists of a comparator which compares the GND' signals with a reference voltage. The main component of the comparator is the single ended differential amplifier formed by transistors M1-M4. M5-M8 form the voltage reference which is connected to the inverting terminal of the differential amplifier while transistors M9-M11 form the current mirror. The differential amplifier used in this scheme is modified so that it can compare multiple inputs simultaneously with a single reference voltage applied to its inverting terminal. This way, we can compare GND' from multiple functional blocks at the same time. The simulation waveforms for the same are shown in Fig. 4.5. Fig. 4.5(a) and Fig. 4.5(b) show the inputs to the ex-or gate. Fig. 4.5(c) shows the output of the ex-or gate with a 0-1 and 1-0 soft errors at 21ns and 33ns respectively when injected with a particle strike. These errors are consequently detected by the detection circuit. Fig. 4.5(d) and Fig. 4.5(e) show the input and output of the comparator circuit. Fig. 4.5(f) shows the output of the comparator circuit after passing the output signal through a buffer.
Since we are adding an NMOS in series with the pull-down network, there will be an effect on the performance, area and power.

a. Performance, Area and Power : Adding a minimum NMOS transistor between the GND' and the gnd terminal increases the resistance of the pull down path of the gate. Hence the V_{HL} will be more than the original implementation. We can retain the original performance of the circuit by

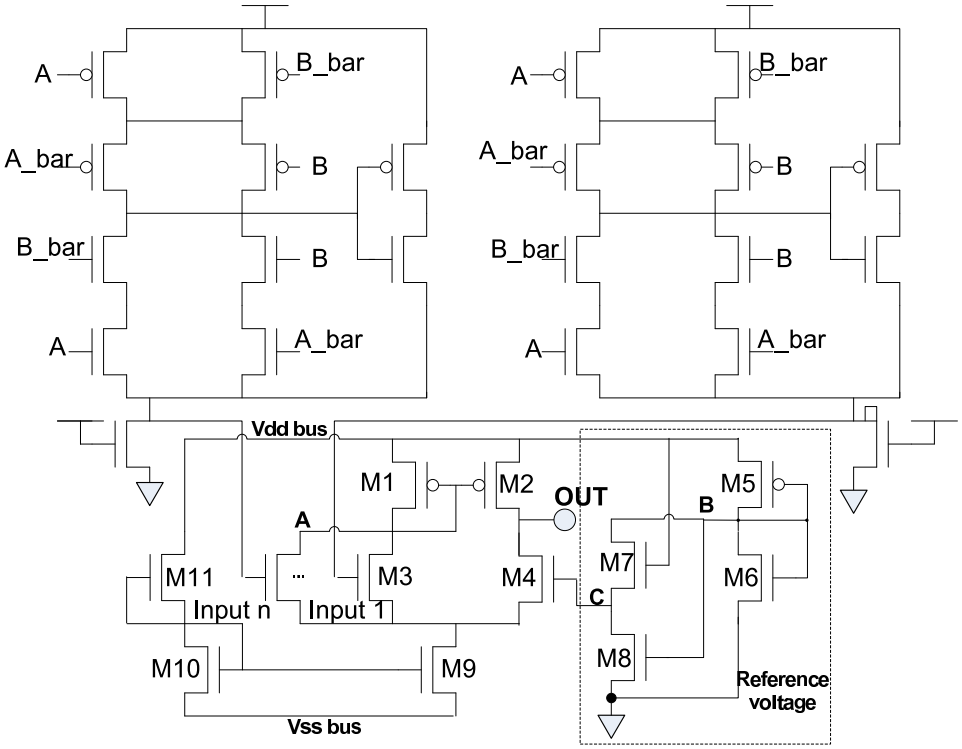
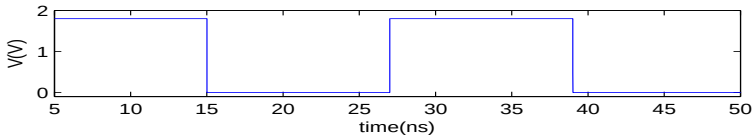


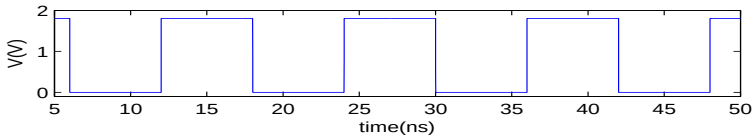
Figure 4.4: Comparator circuit used to detect particle strike in ex-or gates

sizing the PDN appropriately to compensate for the extra NMOS or increase the size of the NMOS itself. However increasing the size of the NMOS affects the voltage signature at GND' making it harder to analyze. Thus we can adjust both the NMOS and the PDN sizing to get optimum performance for minimum area overhead without affecting the GND' signal. Our simulations show that adding a minimum sized NMOS in series with the original gate degrades the performance by 57% while having 5% area and 7% power overhead per gate respectively. If we size both NMOS and PDN we can get the original performance (<1% degradation) at the cost of 10% area overhead per gate respectively. Also, increasing the size of the transistors will mean more current will flow in the PDN and hence we will get a better signature at the GND' which will make the detection process easier. Lastly, according to [36], appropriate sizing also helps in reducing the soft error susceptibility of a gate/combinational block.

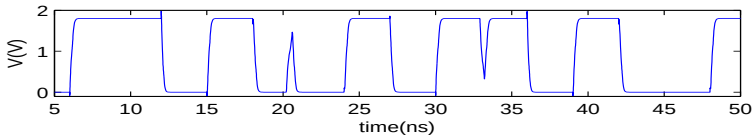
Note that a sophisticated reference voltage is part of the future work and Fig. 4.5 only includes a simple demo circuit for concept proving. Another effect to consider is the degradation of the noise immunity of the circuit due to the addition of NMOS between GND' and gnd bus. This degradation effect is less pronounced since we are monitoring a limited number of transistors in each functional block causing limited fluctuation in the GND'. However, this phenomenon needs to be studied more



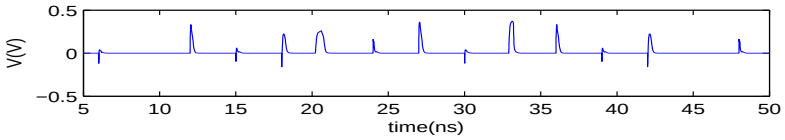
(a) Input A of ex-or gate



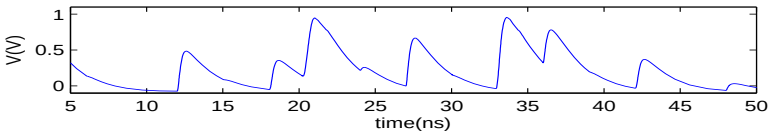
(b) Input B of ex-or gate



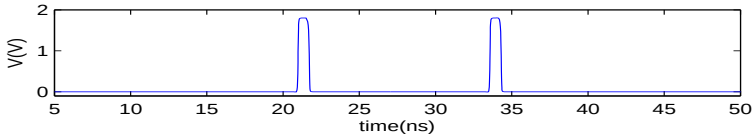
(c) Output of the ex-or gate with 0-1 and 1-0 soft errors



(d) Voltage at GND'



(e) Output of the of the detection scheme



(f) Buffered output

Figure 4.5: Simulation waveforms of particle strike on an ex-or gate

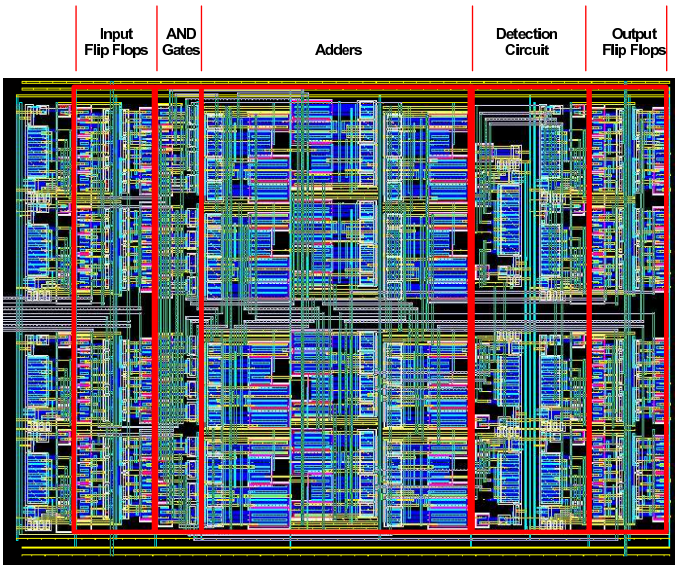


Figure 4.6: Layout of 4x4 pipelined multiplier with detection scheme

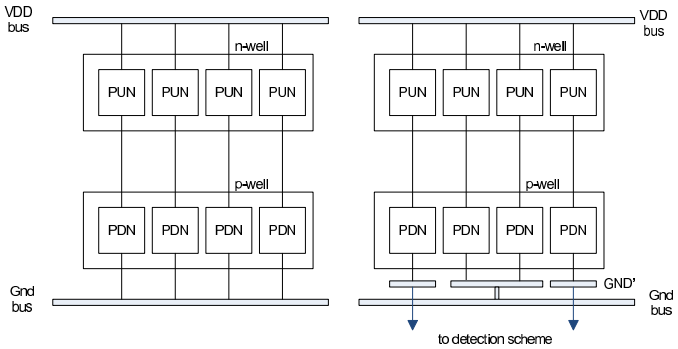


Figure 4.7: Partitioning the combinational logic without changing the layout

and also would be part of the future work.

b. Routing : Since we are monitoring small functional blocks at a time, it is necessary to consider the effect of partitioning the circuit. However, partitioning the circuit into logical blocks does not mean that we have to physically partition the transistors. It just means that the gnd line is partitioned into equivalent GND' lines as shown in Fig. 4.7. The length of the GND' line can be fixed or varied depending on the physical layout of transistors. If that block is not to be connected to the detection circuit, that GND' line can be just shorted to gnd bus. Also, routing of the GND' lines is kept to a minimum by placing the detection circuit between the flip-flops and their drivers so that their respective GND' signals are not routed over long distances as shown in Fig. 4.6. We need not modify the original functional block that is to be connected to the detection mechanism as NMOS

connecting GND' and gnd terminal is placed with the detection circuit. This makes the design flexible since any functional block can be easily connected to the detection mechanism.

c. Determining the Threshold Voltage : The main design parameter to tune is the threshold voltage of the comparators. When setting the threshold, we need to balance the risk of false positives (detecting switching transients as particle strikes) and false negatives (missing particle strikes). In general, false negatives can be reduced by lowering the threshold used for comparison at the first-level comparators. All else being equal, this in theory will increase the possibility of false positives. However, in today's high-speed microprocessors, the transient pulse widths are typically much smaller than that due to SEU. Therefore, in practice, the increase in false positives is small. Note that a false positive only causes overhead by forcing the processor to rollback and restart from a previous checkpoint. Finally, using smaller functional block will have reduced probability of peak transient pulses. The value of the threshold voltage depends on the current passing through the gate. The threshold voltage of the detection circuit can be set by changing the size of M8 transistor as shown in Fig. 4.4. The size of this transistor depends on the relative size of the PMOS and NMOS used in the functional block/gate. Thus a larger gate (larger PMOS,NMOS) will have a higher threshold voltage which can be set appropriately by choosing (lowering) the size of the M8 transistor.

d. Susceptibility to soft errors : Though our main aim is to monitor soft errors in combinational logic, we should also consider the effect of a particle strike on the detection circuit. Whenever a particle strikes the detection circuit without affecting any other logic/gate, the output of the detection mechanism depends on which node it strikes. If it strikes node A, then the output of the detection circuit will go high symbolizing a particle strike. However, this is a false positive since the combinational logic is not affected. If it strikes the output node of the detection circuit, then the output remains low. However, our concern is about false negatives. If a particle strikes both, combinational logic and the detection mechanism at the same, the detection circuit should not mask the error because of the particle strike affecting it. However, the detection circuit offers protection against this error masking. If the output node of the detection circuit is hit by a particle, it pulls down the output node, thus pulling down the reference voltage. As a result, the detection circuit used for some other gates/flipflops using the same reference voltage will give a high output detecting an error. Thus, the detection circuit only has to pay the penalty of a false positive if it gets hit by a particle.

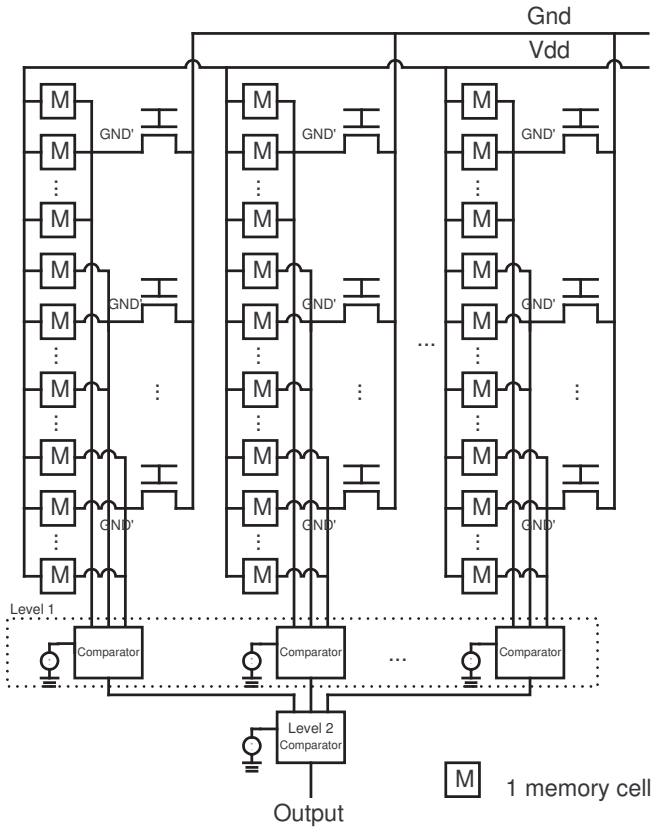


Figure 4.8: Detection Scheme for a memory array

4.2.2 Memory Array

Memory elements in ICs are generally well protected using ECC. However, ECC detects and corrects data during each memory cycle which adds to the access time of the memory. On the other hand, circuit level techniques used to detect errors in memory can detect an error immediately after a bit flip and can be corrected asynchronously without waiting for the read cycle. Therefore it is a possible alternative or complement for coding-based error detection and correction mechanism. This is especially attractive for memory arrays on timing critical path as error-checking is no longer part of the access time. The asynchronous correction of memory elements can be done by using circuit level techniques and ECC as proposed in [37] or using circuit level techniques only. When working in conjunction with ECC, comparator-based circuit provides almost instantaneous detection of an SEU and identifies the memory blocks being affected. ECC can be then used to correct the content. Without this immediate detection, conventional system relies on scrubbing – periodical scanning of the entire memory region to detect and correct latent errors and prevent them from accumulating into more severe forms (*e.g.*, multi-bit errors) that can not be corrected.

In addition to providing the asynchronous detection capability, our design also fills in the gap of protecting the combinational logic in the support elements such as decoders that ECC-type mechanisms can not provide. Furthermore, it makes multi-bit error correction easier to implement as the circuit can easily monitor spikes for rows as well as columns (Fig. 4.8) at the same time. In contrast to Built-in Current Sensors (BICS) proposed in [26, 38, 39] for detecting SEU in memory arrays, which monitors both the V_{dd} and ground current, our proposed scheme detects SEUs by monitoring the ground voltage only. Hence, instead of monitoring the V_{dd} and ground bus of one memory column, we use the hierarchical structure to detect SEUs in memory. The structure used for detecting SEUs in memory is the same as described previously for combinational logic. Memory supply rails show disturbance for read or write signals and a particle strike. The hierarchical scheme has to differentiate between the read/write signals and the particle strike to detect an error. Conveniently, in one column of a memory array, only one cell can be read or written to at any given instance of time. Thus level-one comparator compares GND' inputs from the same memory column simultaneously with the threshold voltage. Hence, each column in the memory array needs one comparator to detect an error in that column. Level-two comparator compares all the level-one comparator outputs to give the final error signal. Fig. 4.8 shows the hierarchical detection scheme applied to a memory array. A memory cell in each column stores one bit of a word, then an error generated in any cell can be detected by the column or level-one comparator.

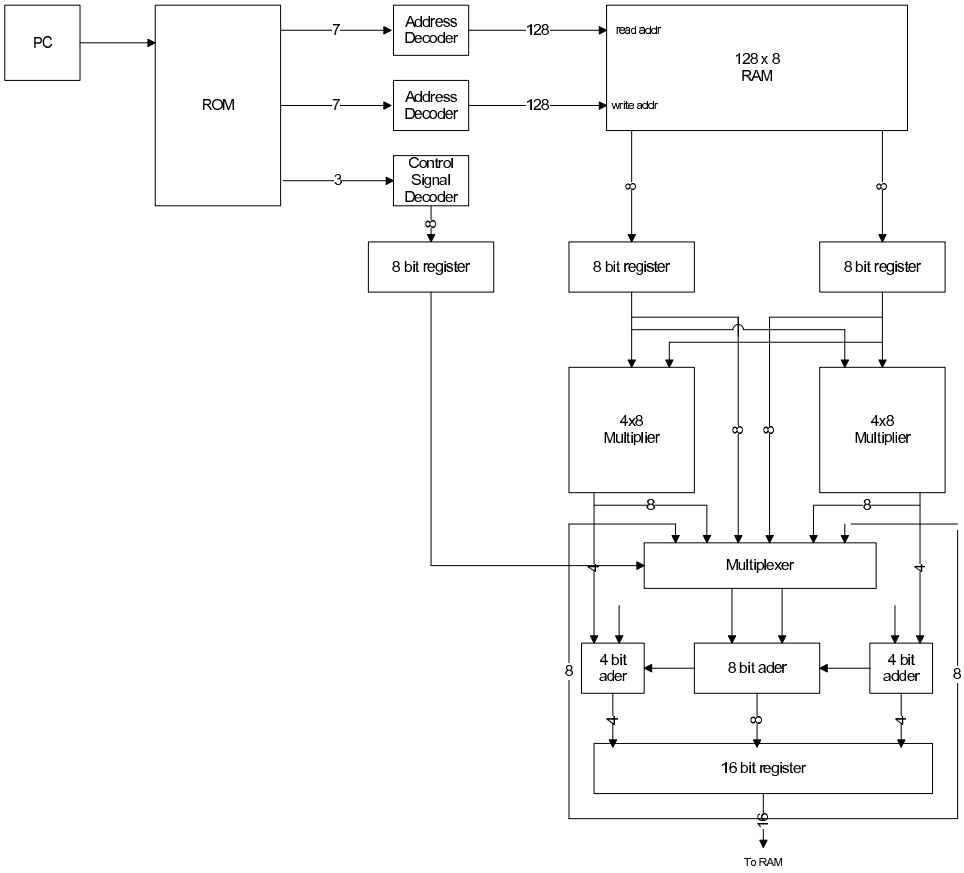


Figure 4.9: Simulation setup

4.3 Simulations and Results

The comparators, combinational logic, and memory arrays were designed in TSMC $0.18\mu m$ process technology. The power supply voltage used for this technology was 1.8V and the spice model is from Spectre.

We model a particle strike by injecting a current pulse at the sensitive node. The shape of the current pulse injected in the circuit is made identical to the actual current pulse observed during a particle strike [9]. Thus the current pulse has exponential nature, with its magnitude and pulse width as the main parameters which can be changed to induce a bit flip. At the circuit level, the particle strike created charge deposition can be modeled as an exponential current pulse at the particle strike site.

$$I(t) = I_0[e^{(-t/t_f)} - e^{(-t/t_r)}] \tag{4.1}$$

where $I(t)$ is the transient current pulse, I_0 is approximately the maximum charge collection current (current peak), t_r is the rise time and t_f is the fall (decay) time of current pulse corresponding to time constant for initially establishing the ion track and collection time constant of the junction, respectively [9]. An error is said to occur in the circuit if the output changes its current state by $V_{dd}/2$.

We applied the detection scheme to a 3-stage pipelined (Fetch/Decode, Execute and Writeback) architecture which consists of a RAM, a control unit and an ALU. The control unit consists of a ROM which stores the instructions and the data address. In the Fetch/Decode stage, the Program Counter provides the ROM address which puts out the address of the data and control signals. In the same cycle, we get the data from the RAM and the control signals are decoded to be given to the ALU. In the Execute stage, the ALU operates on the data obtained from the RAM. The ALU can perform 8x8 addition, subtraction, multiplication, add-accumulate and subtract-accumulate. In the Writeback stage, the output from the ALU is written back to the memory.

In combinational logic, the probability of a soft error propagating through logic and getting latched is very low since it can get masked easily. Therefore, all the functional blocks whose outputs are connected to the next stage of flip-flops are monitored by the detection circuit. Particle strikes modeled a transient current pulses were injected in the sensitive nodes to generate error. The magnitude and width of the current pulses were varied so as to model particle strikes ranging from the lowest to the highest energy levels. The architectural implementation is injected with current pulses with I_0 ranging from 0.8mA to 1.2mA while t_f was varied from 100ps to 500ps.

Table 4.1 shows the simulation results for 1-0 and 0-1 bit flip. Each column in the table shows two symbols. The first symbol represents if an error has occurred or not while the second symbol represents whether the error was detected or not. A cross(x) symbolizes no error or no detection and

Table 4.1: Combinational Logic Simulation Results

0 - 1 flip					
	I_0 **(peak current)**				
t_f **(decay time)**	0.8mA	0.9mA	1.0mA	1.1mA	1.2mA
100ps	xx	xx	xx	xx	xx
150ps	xx	xx	xx	xx	✓✓
200ps	xx	xx	xx	x✓	✓✓
250ps	xx	xx	xx	✓✓	✓✓
300ps	xx	xx	x✓	✓✓	✓✓
350ps	x✓	✓✓	✓✓	✓✓	✓✓
400ps	✓✓	✓✓	✓✓	✓✓	✓✓
450ps	✓✓	✓✓	✓✓	✓✓	✓✓
500ps	✓✓	✓✓	✓✓	✓✓	✓✓

1 - 0 flip					
	I_0 **(peak current)**				
t_f **(decay time)**	0.8mA	0.9mA	1.0mA	1.1mA	1.2mA
100ps	xx	xx	xx	xx	xx
150ps	xx	xx	xx	xx	✓✓
200ps	xx	xx	x✓	x✓	✓✓
250ps	xx	x✓	x✓	✓✓	✓✓
300ps	x✓	x✓	x✓	✓✓	✓✓
350ps	x✓	x✓	✓✓	✓✓	✓✓
400ps	x✓	x✓	✓✓	✓✓	✓✓
450ps	x✓	✓✓	✓✓	✓✓	✓✓
500ps	✓✓	✓✓	✓✓	✓✓	✓✓

a ✓ symbolizes error or detection. For example a x✓ represents that an error has not occurred but has been detected (false positive). The simulation results show that the detection circuit can detect all the particle strikes that result in an error. But sometimes an error is detected by the detection circuit even though the particle strike does not result in error. However, the probability of this situation occurring is very low since the particle strike has to be of very low magnitude. Again, a false positive has no correctness impact. The only cost is the slowdown due to an unnecessary rollback.

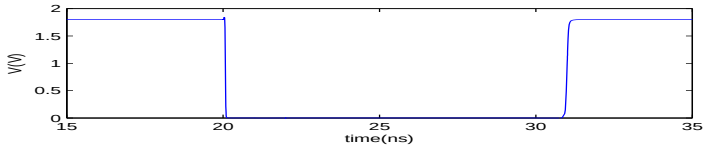
Performance

To determine the speed and power overhead of the system due to the detection circuit, the system was simulated exhaustively. The performance degradation in the combinational logic in the 3 stage architecture implementation is 8%. The performance degradation of the flip-flop measured as the increase in its clock to output (Q_{c-q}) value is 10%. However, this is the worst case performance when a minimum sized NMOS is placed between GND' and gnd terminal (lowest area overhead). The performance degradation can be reduced to a minimal value (less than 1%) by sizing the NMOS and PDN. The power and area overhead are 17% and 18% respectively for less than 1% performance degradation. The power overhead is measured without taking the reference voltage into consideration. The reference voltage always has a short circuit path from V_{dd} to ground terminal and hence burns a lot of power. However, a single reference voltage can be used for multiple comparator circuits reducing its overhead.

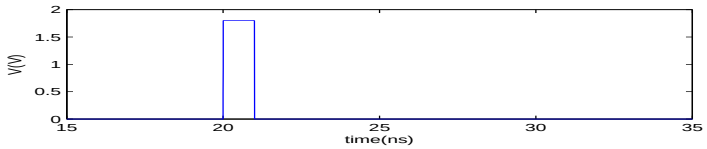
4.3.1 Memory Array

We also simulated the effect of the design on 128 memory cells. Fig. 4.10 shows the simulation waveforms of a particle strike on one memory cell. Fig. 4.10(a) shows the bitline bus signals for the memory column. Fig. 4.10(b) and 4.10(c) show the read and write signals for one cell in the memory column. Fig. 4.10(d) shows the voltage at the virtual ground of the memory cell which is one of the inputs to the level-one comparator. Fig. 4.10(e) shows the final output of the detection scheme. A "0" is written into the RAM cell at time 20ns and a read signal is given to at time 30ns to read it. A particle strike is simulated at time 25ns which causes the written data to be changed to a '1'. Hence the read signal reads the erroneous '1' data instead of the correct '0' data which was written at 20ns. Fig. 4.10(d) shows three spikes corresponding to the write, particle strike and read signal respectively observed at the GND' terminal of the memory cell. The particle strike has the highest amplitude and is detected by the detection circuit in the form of a pulse as shown in Fig. 4.10(e).

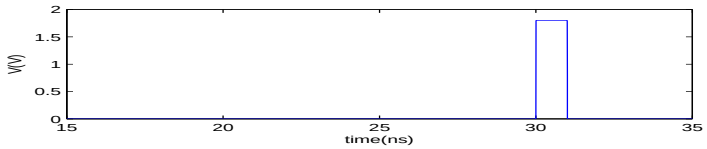
Table 4.2 summarizes the error detection capability of the proposed scheme. Since, the 1-0 and



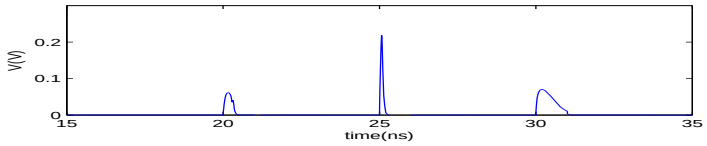
(a) Bitline bus of the memory cell shows a '1' after the read operation at 30ns even though a '0' was written to it at 20ns



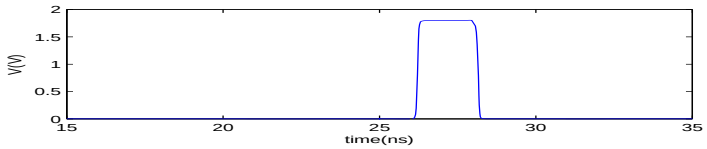
(b) Memory Write writes a '0' at 20ns



(c) Memory read reads a '1' from the memory cell at 30ns



(d) GND' shows the perturbations due to write, particle strike and read



(e) Detection scheme outputs the signal due to a soft error

Figure 4.10: A memory cell is flipped by a particle strike and is detected by the detection scheme.

Table 4.2: Memory array simulation results

	I_0 **(peak current)**			
t_f **(decay time)**	0.3mA	0.4mA	0.5mA	0.6mA
100ps	xx	xx	xx	✓✓
150ps	xx	xx	✓✓	✓✓
200ps	xx	x✓	✓✓	✓✓
250ps	xx	✓✓	✓✓	✓✓
300ps	x✓	✓✓	✓✓	✓✓
350ps	x✓	✓✓	✓✓	✓✓
400ps	x✓	✓✓	✓✓	✓✓
450ps	✓✓	✓✓	✓✓	✓✓
500ps	✓✓	✓✓	✓✓	✓✓

Table 4.3: Impact of process variation for combinational logic

Process Corner	Power Overhead	Detection Time
TT	17%	220 ps
FF	81%	120 ps
SS	6.6%	350 ps

0-1 detection results are the same, they are shown in a single table. It can be observed that the detection circuit is able to detect the all the errors caused by the particle strike. The memory array shows no performance degradation due to the extra detection circuit.

4.4 Reliability Analysis

4.4.1 Process, Voltage, and Temperature (PVT) Variation Analysis

Due to process variations, transconductance and threshold voltage (V_t) of the transistor can vary, affecting the drain current and may give incorrect circuit operation. To evaluate the effect of process variation on the detection circuit, we simulate the process corners as shown in Table 4.3 for combinational logic. The 'FF' corresponds to fast PMOS and fast NMOS (increased current), 'SS' corresponds to slow PMOS and slow NMOS (decreased current) and 'TT' corresponds to the typical case. The 'FF' case has the highest detection speed and power dissipation while 'SS' has the lowest. It can be seen that for just 350ps response time, the power overhead can be as low as 6.6%.

For supply voltage variation, V_{dd} was varied from 1.6V to 2V, 1.8V being the nominal voltage. It

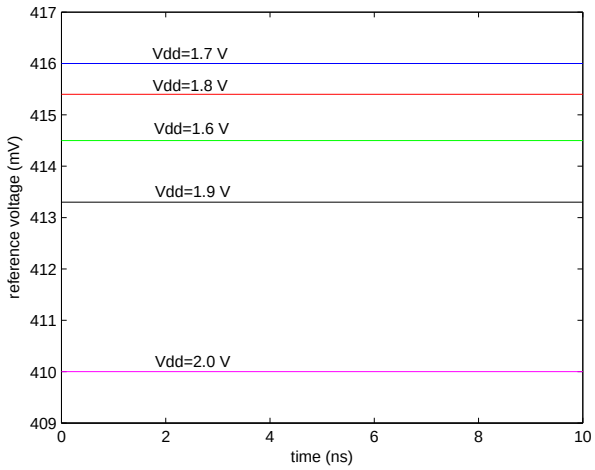


Figure 4.11: Effect of supply voltage variation on the reference voltage

Table 4.4: Comparison between the proposed scheme and other soft error detection schemes

	Proposed Scheme	Bulk-BICS [9]	TMR [11]	[39]	ECC [13]
Area Overhead (Combinational Logic)	18%	29%	100%	–	–
Power Overhead (Combinational Logic)	17%	100%	–	–	–
Area Overhead (Memory Array)	7%	15%	–	7%	11%

is essential that the threshold voltage remains unchanged during supply voltage variation. As shown in Fig. 4.4, node B voltage changes proportionally to the supply voltage change which influences the pulldown transistor M8. So a decrease in supply voltage, decreases the voltage at node B which causes a reduction in M8 gate voltage, resulting in a weaker pull down effect at node C. As a result, the reference voltage (node C) shows only a minor fluctuation due to supply voltage variation as shown in Fig. 4.11. It should be noted that the variation in the reference voltage would vary for different reference voltage magnitudes and this is just one of the many implementations of generating the reference voltage.

The temperature was varied from -25°Cto 70°C [26, 40]. In all cases, we found that the detection circuit was able to detect even the weakest particle strike capable of causing an error.

4.4.2 Power Supply Noise Analysis

Since the detection scheme is dependent on the supply rails for detecting soft error, it is necessary to guarantee correctness of the detection scheme under the influence of power supply noise. To simulate supply noise, 64 large inverters were connected in the circuit having the same supply lines as those used in the detection circuit. The inverters were switched at the rate of 2 GHz to simulate the noise [26]. We found that the circuit continue to work properly under this condition.

4.5 Comparison

Table 4.4 shows the area and power overhead comparison for various soft error detection schemes. For combinational logic, the proposed scheme is compared to bulk-current BICS proposed in [9] and TMR techniques proposed in [11]. The proposed scheme shows a reduction of 11% in area overhead and 83% in power overhead compared to bulk-current BICS [9]. The area improvement is due to the hierarchical structure while improvement in power overhead is due to negligible static power consumption in the proposed scheme. Compared to TMR, the proposed scheme uses 82% less area. Although TMR allows error correction, we note that in real-world environment, the influx of cosmic particle is exceedingly rare in contrast to cycle-to-cycle activities. Relying on rollback for correction would be a far more economic approach. For memory circuits, the proposed scheme is compared with [39] and ECC [13]. The proposed scheme shows a slight area overhead improvement compared to previous schemes. However, for multi-bit protection, it can be used with ECC to provide increased error detection and correction capability at a lower cost.

Chapter 5

Conclusion

The thesis provides a brief overview of the problem of particle strike-induced soft errors affecting ICs at sea-level. The main causes of soft errors being alpha particles and neutrons, we analyzed their sources and interaction with silicon in generating the SEUs. Various error mitigation techniques proposed in literature, from the device level to the architecture level have been discussed. Finally, a novel hierarchical approach to detecting soft errors in combinational logic and memory arrays has been presented in this thesis. The proposed scheme uses just 18% more area and 17% more power to protect combinational logic against soft errors, providing significant power and/or area overhead improvement over prior art. The design is also tolerant to noise and PVT variations, providing robust error detection capability.

Bibliography

[1] R. Baumann and E. Smith, "Neutron-induced Soft Boron Fission as a Major Source of Soft Errors in Deep Submicron SRAM Devices," *IEEE Transactions on Device and Material Reliability,*, vol. 1, no. 1, pp. 152–157, 2000.

[2] F. Wrobel, J. Palau, M. Calvet, O. Bersillon, and H. Duarte, "Incidence of Multi-Particle Events on Soft Error Rates Caused by nSi Nuclear Reactions," *IEEE Transactions on Nuclear Science*, vol. 47, no. 6, pp. 2580–2585, 2000.

[3] T. Heijmen, "Radiation-Induced Soft Errors in Digital Circuits - A Literature Survey," *Unclassified Report*, 2002.

[4] R. Baumann, "Soft Errors in Advanced Semiconductor Devices - Part I:The Three Radiation Sources," *IEEE Transactions on Device and Material Reliability*, vol. 1, no. 1, pp. 17–22, 2001.

[5] R. Baumann, "The Impact of Single Event Effects on Advanced Digital Technologies," in *IEEE EDS Distinguished Lecturer Series*, 2006.

[6] C. Hsieh, Ph. Murley, and R. O'Brien, "Dynamics of Charge Collection from Alpha-Particle Tracks in Silicon Devices," in *IEEE International Reliability Physics Symposium (IRPS'81)*, 1981, pp. 38–39.

[7] Y. Xu, O. Pohland, and H. Puchner, "Influence of Junction Capacitance on SRAM SEU," in *33rd Conference on European Solid-State Device Research (ESSDERC)*, 2003, pp. 537–540.

[8] O. Musseau, "Single-Event Effect in SOI Technologies and Devices," *IEEE International Reliability Physics Symposium*, vol. 43, no. 2, pp. 603–613, 1996.

[9] E. Neto, I. Ribeiro, M. Vieira, G. Wirth, and F. Kastensmidt, "Using Bulk Built-in Current Sensors to Detect Soft Errors," *IEEE Micro*, vol. 26, no. 5, pp. 10–18, Sept/Oct, 2006.

[10] Y. Tsiatouhas, A. Arapoyanni, D. Nikolos, and T. Haniotakis, "A Hierarchical Architecture for Concurrent Soft Error Detection Based on Current Sensing," in *8th IEEE International On-Line Testing Workshop*, 2002, pp. 56–60.

[11] R. Oliveira, A. Jagirdar, and T. Chakraborty, "A TMR Scheme for SEU Mitigation in Scan Flip-Flops," in *8th International Symposium on Quality Electronic Design (ISQED'07)*, 2007, pp. 905–910.

[12] A. Mahmood and E. McCluskey, "Concurrent Error Detection Using Watchdog Processors - A Survey," *IEEE Transactions on Computers*, vol. 37, no. 2, pp. 160–174, 1988.

[13] J. A. Fifield and C. H. Stapper, "High-speed On-Chip ECC for Synergistic Fault-Tolerant Memory Chips," *IEEE Journal of Solid State Circuits*, vol. 26, no. 10, pp. 1449–1452, 1991.

[14] P. Shivakumar, M. Kistler, S. Keckler, D. Burger, and L. Alvisi, "Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic," in *Int'l Conf. Dependable Systems and Networks (DSN 02)*, 2002, pp. 389–398.

[15] N. Seifert, D. Moyer, N. Leland, and R. Hokinson, "Historical Trend in Alpha-Particle induced Soft Error Rates of the Alpha™Microprocessor," in *Proc. Int'l Reliability Physics Symp.*, Apr. 2001, pp. 259–265.

[16] W. Peterson, "On Checking an Adder," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 166–168, Apr. 1958.

[17] J. Watterson and J. Hallenbeck, "Modulo 3 Residue Checker: New Results on Performance and Cost," *IEEE Transactions on Computers*, vol. 37, no. 5, pp. 608–612, 1988.

[18] P. Hazucha, T. Karnik, S. Walstra, B. Bloechel, J. Tschanz, J. Maiz, K. Soumyanath, G. Dermer, S. Narendra, V. De, and S Borkar, "Measurements and Analysis of SER-Tolerant Latch in a 90-nm Dual-Vt CMOS Process," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 9, pp. 1536–1543, Sept. 2004.

[19] M. Nicolaidis, "Time Redundancy Based Soft-Error Tolerance to Rescue Nanometer Technologies," in *Proc. IEEE VLSI Test Symp.*, Apr. 1999, pp. 86–94.

[20] T. Slegel, R. Averill III, M. Check, B. Giamei, B. Krumm, C. Krygowski, W. Li, J. Liptay, J. MacDougall, T. McPherson, J. Navarro, E. Schwarz, K. Shum, and C. Webb, "IBM's S/390 G5 Microprocessor Design," *IEEE Micro*, vol. 19, no. 2, pp. 12–23, Mar./Apr. 1999.

[21] M. Rashid and M. Huang, "Supporting Highly-Decoupled Thread-Level Redundancy for Parallel Programs," in *Proc. Int'l Symp. on High-Perf. Comp. Arch.*, Feb. 2008.

[22] L. Longden, C. Thibodeau, R. Hillman, P. Layton, and M. Dowd, "Designing A Single Board Computer For Space Using The Most Advanced Processor and Mitigation Technologies," in *European Space Components Conference, ESCCON 2002*, Sept. 2002, pp. 313–316.

[23] J. Rohr, "STAREX Self-Repair Routines: Software Recovery in the JPL-STAR Computer," in *International Symposium on Fault-Tolerant Computing*, 1973, pp. 11–16.

[24] Y. Tamir, M. Tremblay, and D. Rennels, "The Implementation and Application of Micro Rollback in Fault-Tolerant VLSI Systems," in *International Symposium on Fault-Tolerant Computing*, June 1988, pp. 234–239.

[25] K. Wu, W. Fuchs, and J. Patel, "Error Recovery in Shared Memory Multiprocessors Using Private Caches," *IEEE Transactions on Parallel and Distributed Systems*, vol. 1, no. 2, pp. 231–240, Apr. 1990.

[26] B. Gill, M. Nicolaidis, F. Wolff, C. Papachristou, and S. Garverick, "An Efficient BICS Design for SEUs Detection and Correction in Semiconductor Memories," in *Design, Automation and Test in Europe (DATE 05)*, 2005, pp. 592–597.

[27] F. Vargas and M. Nicolaidis, "SEU-Tolerant SRAM Design Based on Current Monitoring," in *Proc. 24th Intl Symp. Fault-Tolerant Computing (FTCS 94), IEEE CS Press*, 1994, pp. 106–115.

[28] S. Whitaker, "Single Event Upset Hardening CMOS Memory Circuit," U.S. Patent no. 5, 2003.

[29] J. Canaris, S. Whitaker, and M. Liu, "SEU Hardened Memory Cells for a CCSDS Reed Solomon encoder," *IEEE Transactions on Nuclear Science*, vol. 38, no. 6, pp. 1471–1477, 1991.

[30] M. Liu and S. Whitaker, "Low Power SEU Immune CMOS Memory Circuits," *IEEE Transactions on Nuclear Science*, vol. 39, no. 6, pp. 1679–1684, 1992.

[31] 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, 1996.

[32] 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, 1994.

[33] M. Barry, "Radiation Resistant SRAM Memory Cell," U.S. Patent no. 5 157 625, 1994.

[34] J. Dooley, "SEU-immune For Gate Array, Standard Cell, and Other ASIC Applications," U.S. Patent no. 5 311 070, 1994.

[35] R. Saxena and J. McCluskey, "Dependable Adaptive Computing Systems - The ROAR Project," in *IEEE International Conference on Systems, Man and Cybernetics*, 1998, pp. 2172–2177.

[36] R. Rao, D. Blaauw, and D. Sylvester, "Soft Error Reduction in Combinational Logic Using Gate Resizing and Flipflop Selection," in *11th IEEE/ACM International Conference on Computer-aided Design (ICCAD'06)*, 2006, pp. 502–509.

[37] B. Gill, M. Nicolaidis, and C. Papachristou, "Radiation Induced Single-Word Multi-bit Upsets Correction in SRAM," in *11th IEEE International On-Line Testing Symposium (IOLTS'05)*, 2005, pp. 266–271.

[38] F. Vargas and M. Nicolaidis, "SEU-Tolerant SRAM Design Based on Current Monitoring," in *24th Int'l. Symp. on Fault-Tolerant Computing (FTCS 94)*, 1994, pp. 106–115.

[39] P. Ndai, A. Agrawal, Q. Chen, and K. Roy, "A Soft Error Monitor Using Switching Current Detection," in *International Conference on Computer Design (ICCD'05)*, 2005, pp. 185 – 190.

[40] E. Neto, F. Kastensmidt, and G. Wirth, "A Built-in Current Sensor for High Speed Soft Errors Detection Robust to Process and Temperature Variations ," in *20th Annual Conference on Integrated Circuits and Systems Design (SBCCI'07)*, 2007, pp. 190–195.