Variation-Tolerant Hierarchical Voltage Monitoring Circuit for Soft Error Detection*

Ashay Narsale and Michael C. Huang
Department of Electrical & Computer Engineering
University of Rochester
{narsale, huang}@ece.rochester.edu

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 paper, 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.

1 Introduction

Cosmic and environmental particles (directly or indirectly) ionize silicon, generating charges that can accumulate, change device state, and cause errors known as single event upsets (SEU) or soft errors [1]. Evidence of particle-induced errors in integrated circuits dates back several decades [2]. However, it is technology advancement with the resulting massive integration and feature size and supply voltage reduction that has made the issue of particle-induced errors a concern in the general-purpose domain. Highly publicized incidents underscore the genuineness of the issue and the importance of effective and reliable countermeasures [3].

SEU can occur in memory as well as combinational or sequential logic elements. In today's circuit, combinational logic is much less susceptible to soft errors than memory elements as they offer a structural resistance in the form of electrical, logical, and latching-window masking [4]. Fortunately, memory elements can be effectively and efficiently protected via information redundancy. Modern systems routinely employ error correcting codes (ECC) to protect their memory elements [5]. 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 [4, 6]. 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 [7, 8] , ALUs occupy only a small percentage 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 [9,10] or at the architecture level [11]. Both are less than ideal solutions: transistor-level solutions have a number of fundamental limitations [12]; duplicating an entire core [11] (or worse, triplicating [13]) 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 [14–16].

To tackle the issue of efficient soft error detection in logic, in this paper, 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 [17], our design does not require additional routing nor a substantial area and power overhead. Our

*This work is supported in part by the National Science Fundation under the grants 0747324, 0719790, 0509270, 0829915, and 0824075.

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.

The rest of the paper is organized as follows: Section 2 describes the basic working principle. The design is explained in detail in Section 3. Simulation results are provided in Section 4. Reliability analysis is presented in Section 5. Section 6 compares the proposed scheme with other soft error detection schemes. Finally, Section 7 concludes.

2 Basic Principle of Current Monitoring

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 [17]. Fig. 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. 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. Note that 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} to 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, detecting the current spike caused by the is a viable approach to detecting particle-strike-induced soft errors.

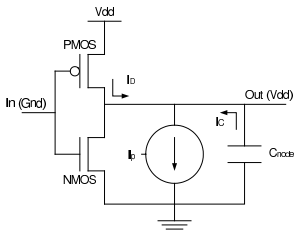


Figure 1. Modeling a particle strike on an inverter [17].

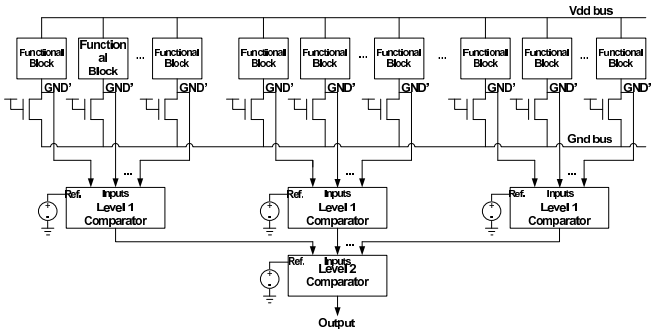


Figure 2. Proposed hierarchical error detection scheme.

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.

3 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 [18]. 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. 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.

3.1 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 fluctuations at this GND', which reflects the switching voltage generated by the functional block, can be monitored to detect an error. The GND'

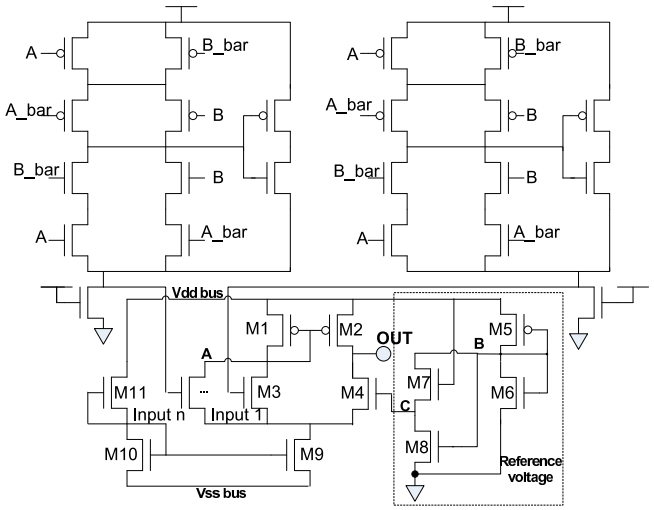


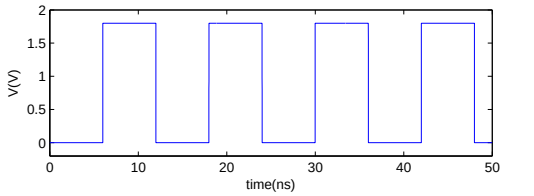
Figure 3. Comparator circuit used to detect particle strike in ex-or gates.

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.

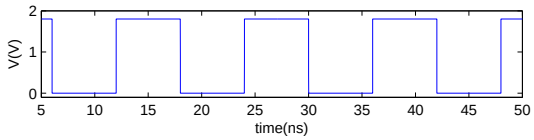
Whenever the voltage of any input GND' terminal rises above the set threshold, the comparator gives a positive output as the input to the second-level comparator. This comparator serves two purposes. First, it amplifies the weak error signal of the first-level comparator to give a full swing output which denotes an error being generated in the system. Second, all the outputs from the first level comparators 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.

3.1.1 Detection Scheme

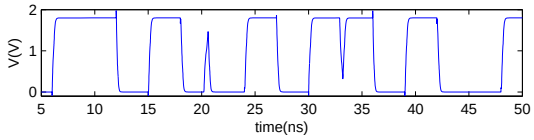
Fig. 3 shows the detection circuit used to detect soft error in two ex-or gates. The detection circuit consists of a comparator which compares the GND' signals with a reference voltage. The main component of the comparator is the single ended differential amplifier formed by transistors M1-M4. M5-M8 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 are shown in Fig. 4. Specifically, Fig. 4(a) and Fig. 4(b) show the inputs to the ex-or gate.



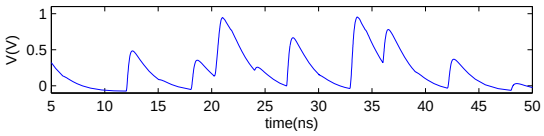
(a) Input A of ex-or gate



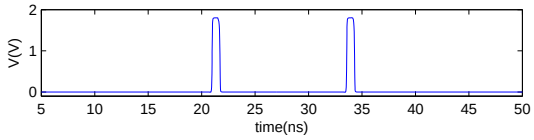
(b) Input B of ex-or gate



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



(d) Output of the of the detection scheme



(e) Buffered output

Figure 4. Simulation waveforms of particle strike on an ex-or gate.

Fig. 4(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 subsequently detected by the detection circuit. Fig. 4(e) shows the buffered output of the comparator circuit. Since we are adding an NMOS transistor in series with the pull-down network (PDN), there will be an effect on the speed, area, and power of the circuit.

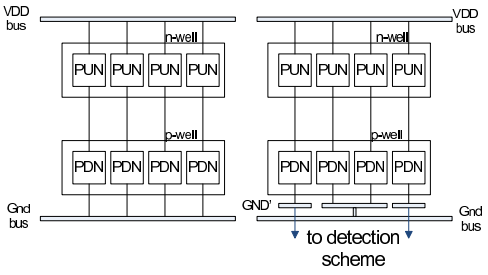


Figure 5. Partitioning the combinational logic without changing the layout.

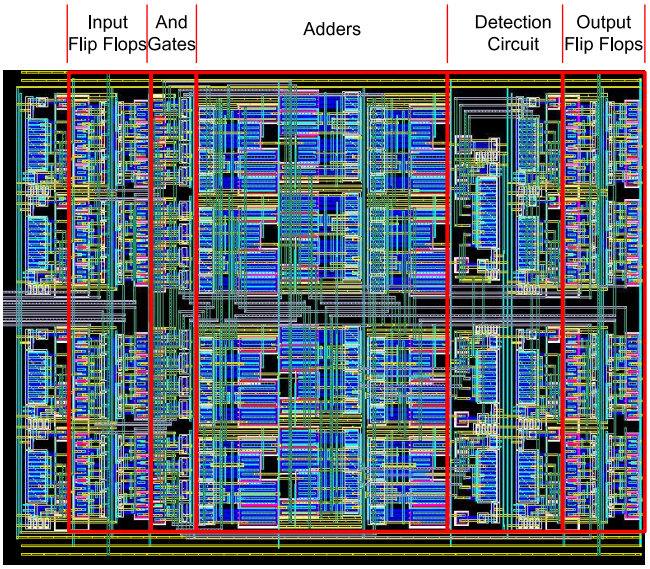


Figure 6. Layout of 4x4 pipelined multiplier with detection scheme.

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} of the gate will increase. We can retain the original performance of the circuit by 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 hard to analyze. Thus we can adjust both NMOS and PDN sizing to get optimum performance with 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 mean more current will flow in the PDN and hence we can get a better signature at the GND' which will make the detection process easier. Lastly, according to [19], appropriate sizing also helps in reducing the soft error susceptibility of a gate/combinational block.

Routing Since we are monitoring small functional blocks at a time, it is necessary to consider the effect of partitioning the circuit. Partitioning the circuit into logical blocks does not mean that we have to physically partition the transistors. It just means that the ground line is partitioned into equivalent GND' lines as shown in Fig. 5. 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 ground 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. 6. We need not modify the original functional block that is to be connected to the detection mechanism as NMOS connecting GND' and ground terminal is placed with the detection circuit. This makes the design flexible since any functional block can be easily connected to the detection mechanism.

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 roll back 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. 3. 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.

3.2 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. Instead, 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. Hence 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 [20] 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

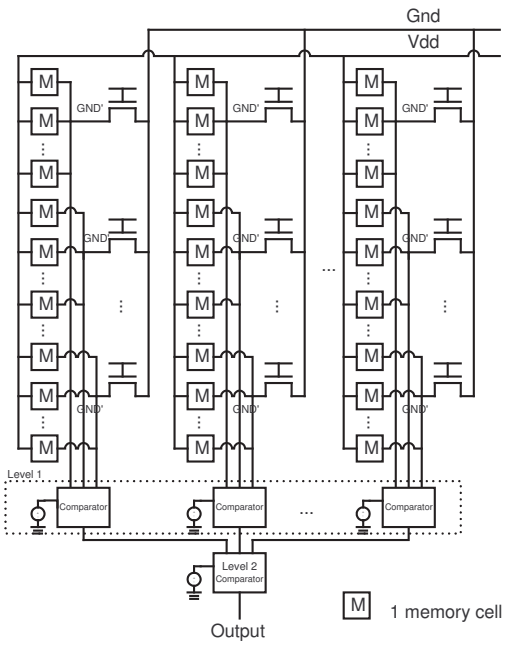


Figure 7. Detection scheme for a memory array.

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 for columns (Fig. 7) at the same time.

In contrast to Built-in Current Sensors (BICS) proposed in [18, 21, 22] 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/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. 7 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/level-one comparator.

4 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 [17]. 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 doubly exponential current pulse at the particle strike site.

$$I(t) = I_0[e^{(-t/t_f)} - e^{(-t/t_r)}] \quad (1)$$

where $I(t)$ is the transient current pulse, I_0 is 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 [17]. An error is said to occur in the circuit if the output changes its current state by $V_{dd}/2$ or more.

We applied the detection scheme to a 3 stage pipelined (Fetch/Decode, Execute, and Writeback) architecture which consists of a RAM array, a control unit, and an ALU. The control unit consists of a ROM array which stores the instructions and the data address. In the Fetch/Decode cycle, 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 cycle, the ALU operates on the data obtained from the RAM. The ALU can perform 8-bit addition, subtraction, multiplication, add-accumulate, and subtract-accumulate. In the Writeback cycle, 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 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 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 an error. However,

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	✓✓	✓✓	✓✓	✓✓	✓✓

Table 1. Combinational Logic Simulation Results.

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 slow-down due to an unnecessary rollback.

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 ground 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.

5 Process, Voltage, and Temperature 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 2 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%.

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

Table 2. Impact of process variation for combinational logic.

For supply voltage variation, V_{dd} was varied from 1.6V to 2V, 1.8V being the nominal voltage. It is essential that the threshold voltage remains unchanged during supply voltage variation. As shown in Fig. 3, node B voltage changes proportionally to the supply voltage change which influences the pull-down 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.
The temperature was varied from -25°C to 70°C [18]. In all cases, we found that the detection circuit was able to detect even the weakest particle strike capable of causing an error.

6 Comparison

Table 3 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 [17] and TMR techniques (implemented for flip flops) proposed in [23]. The proposed scheme shows a reduction of 11% in area overhead and 83% in power overhead compared to bulk-current BICS [17]. 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 forward error correction, we note that in a general-purpose environment, the influx of cosmic particle is exceedingly rare in contrast to cycle-to-cycle activities. Relying on rollbacks for correction would be a far more economic approach.

	Proposed Scheme	Bulk-BICS [17]	TMR* [23]	[22]	ECC [5]
Area Overhead (Combinational Logic)	18%	29%	100%	–	–
Power Overhead (Combinational Logic)	17%	100%	–	–	–
Area Overhead (Memory Array)	7%	15%	–	7%	11%

Table 3. Comparison between the proposed scheme and other soft error detection schemes.*TMR scheme is implemeted for flip flops only.

For memory circuits, the proposed scheme is compared

with [22] and ECC [5]. 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.

7 Conclusion

With continued scaling and ever larger-scale integration, soft errors are projected to become more and more frequent. While error detection and correction are straightforward and efficient with memory arrays, we still do not have an efficient mechanism to even detect errors in the logic elements. Relying on full-blown redundancy will probably continue to be a niche-market solution. In this paper, we have presented a novel hierarchical approach of detecting particle-induced soft errors in combinational logic as well as in memory arrays. The proposed scheme relies on detecting the current spikes on the power rail due to particle strikes. In our simulations, the particular design investigated is capable of detecting all spikes that led to circuit errors, resulting in no false negatives. The detection circuit incurs an 18% area overhead and 17% power overhead. This represents a significant improvement over prior art. The design is also tolerant to PVT variations, providing a robust error detection capability. Overall, we believe such design is a promising approach for general-purpose architectures and further exploration in this direction should be performed.

References

[1] R. C. Baumann, "Radiation-Induced Soft Errors in Advanced Semiconductor Technologies," *IEEE Trans. on Device and Material Reliability,*, vol. 5, no. 3, pp. 305–316, 2005.

[2] T. C. May and M. H. Woods, "Alpha-Particle-Induced Soft Errors in Dynamic Memories," *IEEE Trans. Electron Device*, vol. 26, no. 1, pp. 2–9, 1979.

[3] D. Lyons, "Sun Screen," *Forbes*, Nov. 2000, http://www.forbes.com/global/2000/1113/0323026a.html.

[4] 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 *Proc. Int'l Conf. Dependable Systems and Networks*, 2002, pp. 389–398.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

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

[11] T. Slegel, R. A. 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.

[12] 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, pp. 393–404.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

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

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

[20] B. Gill, M. Nicolaidis, and C. Papachristou, "Radiation Induced Single-Word Multi-bit Upsets Correction in SRAM," in *Proc. 11th IEEE Int'l On-Line Testing Symposium*, 2005, pp. 266–271.

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

[22] P. Ndai, A. Agrawal, Q. Chen, and K. Roy, "A Soft Error Monitor Using Switching Current Detection," in *Proc. Int'l Conference on Computer Design*, 2005, pp. 185 – 190.

[23] R. Oliveira, A. Jagirdar, and T. Chakraborty, "A TMR Scheme for SEU Mitigation in Scan Flip-Flops," in *Proc. 8th Int'l Symposium on Quality Electronic Design*, 2007, pp. 905–910.