# False Error Study of On-line Soft Error Detection Mechanisms

Kiran Kumar Reddy M [1]
E-mail:mkiran@cedt.iisc.ernet.in

Bharadwaj S. Amrutur [1]
E-mail:amrutur@ece.iisc.ernet.in

Rubin A. Parekhji [2]
E-mail:parekhji@ti.com

[1] Department of ECE, Indian Institute of Science, Bangalore, India
[2] Texas Instruments (India) Pvt. Ltd., Bangalore, India

## Abstract

*With technology scaling, vulnerability to soft errors in random logic is increasing. There is a need for on-line error detection and protection for logic gates even at sea level. The error checker is the key element for an on-line detection mechanism. We compare three different checkers for error detection from the point of view of area, power and false error detection rates. We find that the Double Sampling checker (used in Razor), is the simplest and most area and power efficient, but suffers from very high false detection rates of 1.15 times the actual error rates. We also find that the alternate approaches of Triple Sampling and Integrate and Sample method (I&S) can be designed to have zero false detection rates, but at an increased area, power and implementation complexity. The Triple Sampling method has about 1.74 times the area and twice the power as compared to the Double Sampling method and also needs a complex clock generation scheme. The I&S method needs about 16% more power with 0.58 times the area as Double Sampling, but comes with more stringent implementation constraints as it requires detection of small voltage swings.*

## 1. Introduction

Radiation-induced soft errors on large-scale integrated circuits are becoming increasingly problematic as device sizes are scaled down, operating voltages are reduced, and node capacitances shrink [1]. Memories are more vulnerable to soft errors due to small transistor sizes and storage capacitances. But recently, combinational logic is also becoming susceptible to soft errors due to technology scaling. Soft errors are caused by the charge deposition induced by collision of high energy radiation particles like neutrons and alpha particles with substrate in semiconductor materials. Neutrons are produced by collision of cosmic rays with atmospheric particles [2], while alpha rays are emitted by the impurities in the packaging materials and interconnect [3]. While both neutrons and alpha particles deliver parasitic charge, the charge deposited by neutrons (25-150 fC/$\mu$m) is much greater in magnitude than alpha-particles (4-16 fC/$\mu$m) [4].

The key ingredient of an on-line error detection mechanism is the error checker. The important parameters for a error checking circuit are its true error detection capability and false error rejection capability. False error is one which doesn't affect the system output but the detection circuit flags an error due to its inherent nature of comparison. Ideally it should detect all true errors and reject all false errors. Too high a false detection rate will lead to very significant degradation in system performance. In the next section we will introduce the problem of false errors. In section 3 we give an overview of on-line detection mechanisms. We will then describe the simulation setup we have used to characterize their performance in Section 4. We will provide some analysis and discuss simulation results in Section 5 followed by our conclusions in Section 6.

## 2. Problem of false errors

There is a problem of false errors involved in the error checkers used for soft error detection. There is an error in the system only if there is a soft error glitch around the positive edge of the flip-flop. For example, Razor works by comparing the samples at the positive and negative edge of the clock. It flags an error if there is a mismatch between the two samples. It detects the actual error when there is a glitch at the positive edge of the clock (Fig. 1). But it also flags an error when there is glitch at the negative edge of clock. The false errors occur because of the comparison used in error detection (Fig. 1).

The SER due to neutrons increases by orders of magnitude from the ground level to higher altitudes. Let us examine the SER at the ground levels and at an altitude of 60,000 ft. where the neutron flux is maximum. The authors in [5] propose a simple model for the estimation neutron induced SER.

$$SER(Q_c) = BGR(d, Q_c).N.V_s.C \quad (1)$$

where BGR is the burst generation rate, N is the neutron flux, $V_s$ is the sensitive volume (sensitive area x sensitive depth), C is the collection efficiency, and $Q_c$ is the critical

978-0-7695-3264-6/08 $25.00 © 2008 IEEE
DOI 10.1109/IOLTS.2008.29

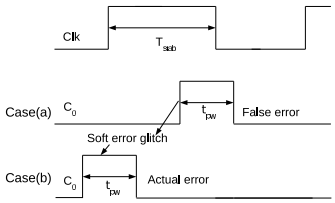



**Figure 1. False errors (a) and Actual error (b) waveforms**

charge of the soft error. Using this formula, they calculate the $SER_{neutron}$ for 350nm process @3V for a drain area of around 36 $\lambda^2$ to be $6X10^{-12}errors/hr$, where $\lambda$ is half the minimum channel length. We will use this number to extrapolate and estimate the SER for a unit inverter of NMOS/PMOS size of 6$\lambda$/12$\lambda$ in a 65nm process. With a typical diffusion extension of 5$\lambda$, the diffusion area at the output of this unit inverter is 90 $\lambda^2$. The critical charge, $Q_c$, the BGR and hence the $SER_{neutron}$ for this unit inverter in the 350nm process at a reduced voltage of 1V, is also $1.5X10^{-12}errors/hr$ since $Q_c \propto Areajunction \times Voltage$. The authors in [6] show that the SER at 65nm is approximately 0.25 times that of SER at 350nm at 1V. They also show that at 65nm the rates due to the alpha and neutron induced SER are same at ground levels and hence,

$$SER_{\alpha} = SER_{neutron} = 1.5X10^{-12}\frac{errors}{hr} \quad (2)$$

and the total SER is the sum of the two SERs as they are independent.

Consider a system with cycle time of $T_{cycle}$ and let $Tp_{min}$ be the minimum sustainable pulse. Lets assume that the soft error strike can occur at any point within the clock cycle and it creates a pulse of width no more than $Tp_{min}$ which gets propagated to the sampling flip-flop. The probability of this pulse causing a false error (see Fig. 1) in the Razor scheme is

$$ErrorRate_{false} = Tp_{min}/T_{cycle} * SER * N_{flops} \quad (3)$$

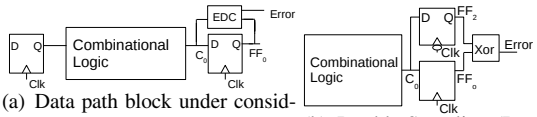
where $N_{flops}$ is the number of flops in the system. Consider a large ASIC with a cycle time of around $T_{cycle} = 40FO4$ and $N_{flops} = 4million$. Typically $Tp_{min}$ is about 3 FO4 (fanout-4) inverter delays. For such an ASIC in a 65nm process, Equation 3 gives a $FIT_{false} = 12000$ and a Mean Time to False Error (MTFE) = 126 years at ground level. At an altitude of 60,000 ft., the Neutron flux increases 230 times that at ground level to 4580 $cm^{-2}hr^{-1}$ [7]. However the alpha flux will remain unchanged and hence

$$SER_{60,000ft.} = (SER_{\alpha} + 230SER_{neutron})_{ground} \quad (4)$$

This leads to an MTFE at 60,000 ft. of only 524 days. Note that from Equation 3, we can see that the false error rate increases and hence MTFE decreases for high performance chips like microprocessors which have much smaller cycle times.

## 3. Overview of On-line Error Detecting Mechanisms

Consider a single pipeline stage shown in the Fig. 2(a) where Error Detection Circuit (EDC) appends the flip-flop to be protected. Different methods have been proposed for on-line testing of soft errors, delay and crosstalk faults in combinational logic. The basic idea behind all these methods is that they try to distinguish between a fault free and a faulty system. This is done by comparing combinational output, $C_o$, with the latched value, $FF_o$, over a period where $C_o$ is expected to be stable. Typically, $C_o$ is stable in the interval $T_{clk}$ to $T_{clk} + T_{cmin}$, where $T_{cmin}$ is the contamination delay of the logic. Usually $T_{cmin}$ is very small and any comparison over that period is difficult. Hence if one wants to implement error detection, $T_{cmin}$ is increased to a larger $T_{stab}$, called stability window, such that the comparison of $C_o$ and $FF_o$ can be done. Hence this error detection comes with the overhead of having to buffer all the fast paths in the combinational logic to have a delay of at least $T_{stab}$.




(a) Data path block under consideration

(b) Double Sampling (Razor)method

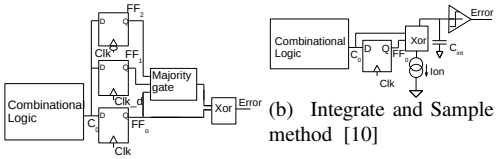
**Figure 2. Data path and Razor blocks.**

This class of detection methods can be realized in more than one form depending on the way comparison is done between $C_o$ and $FF_o$ in the stability window. Based on the type of comparison, which can be continuous or discontinuous, there are three methods:

1. **Double Sampling:** Compares the latched value $FF_o$ with a sample taken at the end of the stability period ($FF_2$) [11] as shown in Fig. 2(b). The error checker outputs from each Flop are combined in a big OR tree.

2. **Triple Sampling:** In addition to samples taken in Double Sampling method ($FF_o$, $FF_2$), another sample is taken at the middle of the stability window ($FF_1$). Taking the majority of these samples and compare it with actual latched value ($FF_o$) as shown in Fig. 3(a) gives the error [8]. This needs an extra clock ($Clk\_d$) at the center of stability window. The error checker outputs from each flip-flop are combined in a big OR tree.

3. **Integrate and Sample (I&S):** Here comparison is done by using a current integrator that integrates difference between $C_o$ and $FF_o$ over the stability period ($Tstab_{eff}$)( referred to as Current Sensing Comparator in [9] and Modified Stability Checker in [10]) as shown in Fig. 3(b). The error checker output from Flops are combined using a dynamic OR structure which also provides for the capacitor for current integration as shown in Fig. 3(b).




(a) Triple Sampling method

(b) Integrate and Sample method [10]

**Figure 3. I&S and Triple Sampling methods**

The Double Sampling method is much like the Razor [11] technique that has been proposed for dynamic voltage scaling in order to operate the circuit at lowest possible supply voltage that meets the speed specifications. It compares two samples of the combinational output $C_o$, one sampled at the positive clock edge ($FF_o$) and the other at the negative clock edge ($FF_2$). A mismatch between the two is used to detect a delay error due to inadequate supply (Fig. 5(b)). The same structure can also be used to detect soft errors. Fig. 5(a) schematically shows that a soft error some where in the logic block causes a glitch at $C_o$ during the sampling phase thus leading to an error which gets detected by the checker. While this technique can detect any delay errors with minimal extra hardware, when used for detection of soft errors, it can result in a considerable number of false positives due to inadequate sampling. For example, if a soft error related glitch occurs around the negative edge of clock, then it will be falsely reported as an error, even though there is no error in the operation of the system. In general, this method can detect all delay errors of the size of $T_{stab}$.

Since soft error in combinational logic is a transient disturbance, it can be detected by averaging the difference of

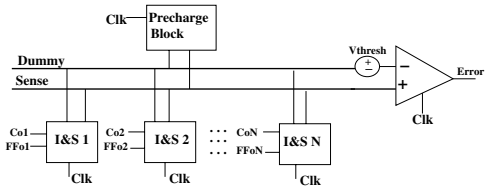
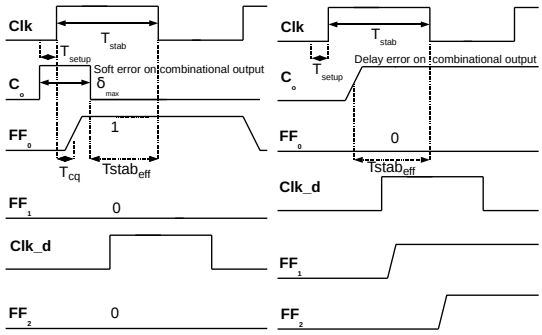



**Figure 4. Dynamic ORing structure of I&S method [10]**




(a) fig.a

(b) fig.b

**Figure 5. Soft error (fig.a) and Delay error (fig.b) cases**

the combinational output and the flip-flop output over the stability window. An approximation to this averaging can be achieved via the Triple Sampling method [8]. Here, three samples of $C_o$ are taken in the stability window ($FF_o$, $FF_1$, $FF_2$) and the majority value of these samples is compared to the latched value ($FF_o$) to detect an error. This method can detect soft error and delay errors up to to a size $\frac{T_{stab}}{2}$.

The I&S scheme uses a stability window of $T_{stab}$ to integrate the difference between $C_o$ and $FF_o$ onto the sense line [10] as shown in Fig. 4. The sense and dummy are dynamic nodes that are precharged during the negative half cycle. During the positive half cycle, upon the occurrence of fault, one or more of the I&S discharge the sense node which is detected by a comparator at the end of positive half cycle. At the end of stability window the difference between sense and dummy lines has to be at least $V_{thresh}+\delta V$ ($\delta V$ is off-set of comparator) for it to trigger an error.

## 4. Simulation setup

When a particle strikes a sensitive node in the circuit, it produces a current pulse with a rapid rise time, but a gradual fall time. The shape of the pulse is modeled by a one-parameter function [12] as shown in Equation 5

$$I_{ser}(t) = \frac{2Q}{\sqrt{\pi}T}\sqrt{\frac{t}{T}}e^{\frac{-t}{T}} \quad (5)$$

Here Q refers to the amount of charge collected due to the particle strike, T is the time constant for the charge collection process and is a property of the CMOS process used for the device. The probability distribution of the collected charge Q is extracted from the data given in [13]. The current pulse produced by a particle strike results in a voltage pulse at output node of the device.

We study the false error rates of the different checker schemes by comparing their operation with that of an ideal error checker which detects all true errors and doesnt flag

| Error (ideal checker) | Error (actual checker) | Type of error |
|---|---|---|
| 0 | 0 | No error |
| 0 | 1 | False error |
| 1 | 0 | Undetected error |
| 1 | 1 | Actual error |

**Table 1. Definition of different types of injected errors**

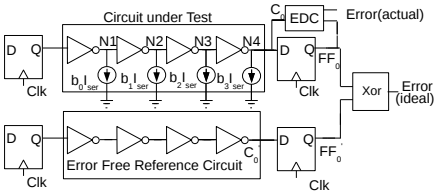



**Figure 6. Datapath under test, $b_i = 1$ if node i has SEU**

any false error. We use two copies of the circuit, where one (circuit under test) is modified by incorporating current sources at each node, which can inject the desired current pulse as in Equation 5 to mimic a soft error event. The other is the ideal error free circuit which gives the correct result (see Fig. 6). The circuit under test block also has the error checker attached to its output flip-flop. The ideal checker just compares the output of the circuit under test ($FF_o$) with that of the ideal error-free circuit ($FF_o{}'$), thus flagging an error only when there is an actual error and does not flag any false errors. Comparing the error results of the ideal checker and actual checker, four conditions arise as shown in Table 1. The circuits are simulated for the worst case charge injection. A set of 4000 errors are injected into the circuit at 1000 different injection points each clock for the four nodes ($N_{1-4}$) in the circuit under test. The errors are injected randomly at one of the four injection points, at a random time within each clock cycle. The same simulation framework is used for all the three types of error checkers.

The design of the checkers for certain error detection depends on the maximum soft error size ( $\delta_{max}$) it can tolerate. The voltage pulse width developed due to the current pulse depends on:

1. Drive strengths of the pull-up and pull down paths of the inverter.
2. The charge induced, Q due to the particle strike.
3. The load capacitance, $C_{node}$.

The width of the output pulse developed due to the soft error strike is maximum when the drive strength and the load capacitance are minimum and the charge induced, Q is the maximum tolerable value used by the designer. Since our circuit under test is a FO4 chain we simulated the weakest gate in the chain to find $\delta_{max}$. In 130nm technology, we find that the maximum pulse width of soft error ($\delta_{max}$) is

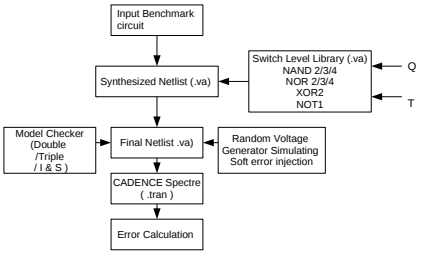



**Figure 7. Simulation flow for soft error estimation**

approximately 200ps for a maximum charge of 120fC, obtained through circuit simulation using the current injection model of Equation 5. Note that $\delta_{max}$ is under the control of the designer by the appropriate choice of sizing for the library elements.

While accurate transistor level simulation can be done for the simple chain of four inverters shown in Fig. 6, this becomes computationally infeasible for larger circuits. Hence we have developed a switch level library of gates in terms of which bigger benchmark circuits are synthesized. Our switch level library consists of NAND2/3/4, NOR2/3/4, XOR2, NOT1 gates. In general, any logic gate with function f (in1, in2....) is implemented as a ideal logic function followed by pull-up and pull-down resistors connecting the output of the cell to power and ground respectively. A current source (Q,T) is appended at the output of the gate to simulate the induced charge of the soft error strike. The simulation flow is explained in the Fig. 7.

The I&S [10] is modeled by current source, which is enabled by difference between $C_o$ and $FF_o$ (Fig.3(b))and an enable signal (Clk), that discharges the sense node. The comparator has an offset of $\delta V$ and triggers an error if the $V_{sense} < V_{thresh} - \frac{\delta V}{2}$. The voltage $V_{sense}$, must be at least $V_{thresh} + \frac{\delta V}{2}$ for it not to trigger error($V_{sense} = vdd - \frac{I_{on}}{C_{int}}t$, sense node is initially precharged to vdd). We have used a comparator with a threshold voltage ($V_{thresh}$) of 150mV and $\delta V$=100mV which are nominal and conservative values. The change in voltage of the node $V_{sense}$ is given by

$$\Delta V_{sense} = \frac{I_{on}}{C_{int}}t \tag{6}$$

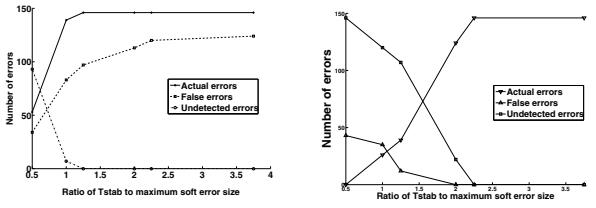
## 5. False Error Analysis and Simulation Results

Every error detection mechanism has a tendency to flag false errors due to inherent nature of comparison involved. In Double Sampling method as long as $T_{stab} > \delta_{max}$ all the errors are detected. But it also gives false errors when there is a transition on the combinational output ($C_o$) around the second latching edge as shown in Fig. 1.

Assuming uniform distribution of time of soft error strike and uniform distribution of delays from the node of soft error strike to the output node, there will always be soft error pulses around the negative clock. The values of $FF_o$

and $FF_2$ are different and hence Double Sampling method gives an error even though there is no error.

Fig. 8(a) shows the error detection statistics for worst case injected charge of Q=120fC for C17 benchmark circuit simulated at switch level. All the actual errors are detected for $T_{stab} > \delta_{max}$, but the false errors are around 67% of actual errors.




(a) Error statistics for Double Sampling method (C17, switch level, $Q_c = 120fC$)

(b) Error statistics for Triple Sampling method (C17, switch level, $Q_c = 120fC$)

**Figure 8. Error statistics.**

In case of Triple Sampling method, a majority operation of three samples separated by a minimum separation of $\delta_{max}$ always gives correct value and avoids any false errors i.e. $T_{stab} > 2 * \delta_{max}$. But this comes with overhead of generating and distributing an extra clock ($Clk\_d$) to do an additional sampling in the middle of the stability window. Fig. 8(b) shows that under worst case charge injection, the undetected errors decrease while both the actual and false errors increase till 2.25*$\delta_{max}$ for the Triple Sampling method.

In I&S method the false errors can be eliminated by proper choice of the $I_{on}$. The first condition on $I_{on}$ is put by the offset voltage of the comparator.

$$\frac{I_{on}(T_{stabeff} - \delta_{max})}{C_{int}} > \delta V \tag{7}$$

where $T_{stabeff} = T_{stab} - (\delta_{max} - T_{setup})$, which is the actual period available for integration shown in Fig. 5(a).

In order to prevent the false soft error created glitches (of duration $\delta_{max}$) from triggering an error, the condition that must be satisfied is:

$$\frac{I_{on}\delta_{max}}{C_{int}} < V_{thresh} - \frac{\delta V}{2} \tag{8}$$

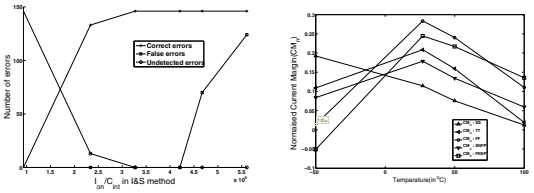
Similarly, for the actual errors to trigger the error the condition is given by:

$$\frac{I_{on}T_{stabeff}}{C_{int}} > V_{thresh} + \frac{\delta V}{2} \tag{9}$$

This puts a condition on the charging current $I_{on}$

$$\frac{(V_{thresh} + \frac{\delta V}{2})}{T_{stabeff}} < \frac{I_{on}}{C_{int}} < \frac{(V_{thresh} - \frac{\delta V}{2})}{\delta_{max}} \tag{10}$$

Fig. 9(a) clearly shows that there is a range of currents for complete detection and false error rejection as predicted by Equation 10.




(a) Different errors as function of the ratio $\frac{I_{on}}{C_{int}}$ for C17

(b) Normalized Current Margin ($CM_n$) as function of temperature and process

**Figure 9. Characteristics of I&S method**

In order to prove whether I&S method can be practically implemented we have measured the $I_h$ (lower end of current in Equation 10) and $I_l$ (higher end of current shown in Equation 10) and the charging current ($I_{on}$) set in the circuit across different process corners and ambient temperatures for a 130nm industrial process. Fig. 9(b) plots the normalized current margin ($CM_n = min(\frac{(I_h - I_{on})}{I_{on}}, \frac{(I_{on} - I_l)}{I_{on}})$ as function of process and temperatures. It shows that the margin are not met only at the extreme temperature of $-50^oC$, for FNSP corner.

The Double and Triple Sampling methods have been simulated for $T_{stab} = \frac{T_{clk}}{2}$ for comparison with the I&S method. The value of $I_{on}$ satisfying Equation 10 is chosen. All the comparison are made for worst case induced charge of Q=120fC(Table 2). The simple inverter chain circuit has been simulated at the transistor level whereas the remaining benchmarks (C17, C432, C499) are simulated at the switch level.

The Triple Sampling method has higher area ( 1.7 times) and power overhead ( 1.83 times) than Double Sampling method because of the one extra flip-flop and the majority voter. The false error to correct error ratio is around 1.15 for the Double Sampling in comparison to 0 for the Triple Sampling method and I&S method. The I&S method consumes around 6% more power and occupies 0.58X the area of the Double Sampling method. The areas were estimated using Cadence Vituoso Layout Editor. Note that the overheads for generation and distribution of extra clock for Triple Sampling have not been included in this analysis. The variability in the extra clock that is to be generated is critical as it increases the $T_{stab}$ and hence the area of the combinational block, in order to satisfy condition ($T_{stab} > 2 * \delta_{max}$).

Arguments for the existence of false error detection cases of the Double Sampling method are independent of the circuit under test as seen from the results. As discussed earlier, the Double Sampling method flags a false error if the soft error event causes a glitch at the input of the extra latch

| Item | Double Sampling ($T_{stab} = \frac{T_{clk}}{2}$) | | | Triple Sampling ($T_{stab} = \frac{T_{clk}}{2}$ | | | I&S ($I_{on} = 35\mu A$) | | |
|---|---|---|---|---|---|---|---|---|---|
| | A | U | F | A | U | F | A | U | F |
| FO4 | 153 | 0 | 177 | 153 | 0 | 0 | 153 | 0 | 0 |
| C17(6 gates) | 146 | 0 | 124 | 146 | 0 | 0 | 146 | 0 | 0 |
| C432(160 gates) | 148 | 0 | 125 | 148 | 0 | 0 | 148 | 0 | 0 |
| C499(202 gates) | 278 | 0 | 212 | 278 | 0 | 0 | 278 | 0 | 0 |
| Area | 1 | | | 1.74 | | | 0.583 | | |
| Power | 1 | | | 2 | | | 1.06 | | |

**Table 2. Comparison of three methods (Double, Triple, Integrate and Sample method) for $T_{clk} = 1.5ns$ and $T_{stab} = 750ps$. A-Actual errors, U-Undetected errors, F-False errors**

around the end of the $T_{stab}$ window. I&S and the Triple Sampling checkers can be designed such that their false error detection can be made to be zero.

## 6. Conclusions

In this paper we have presented a comparative study of three different error checkers for soft error detection for online testing. Our study was based on SPICE and switch-level simulations of the circuit under test, with models of current sources mimicking soft error strikes at potentially every node of the circuit. This framework also includes the ideal circuit under test without any errors as well as the ideal error checker against which the three error checkers can be compared.

All three error checkers can be designed to detect all true errors. The Double Sampling method has a large false error detection rate, while the I&S and Triple Sampling checker can be designed to have zero false errors.

In terms of the implementation complexity, the Double Sampling technique is the simplest and can be easily integrated into the existing digital design flows. But it suffers from a high false error detection rate of 1.15x the actual errors for our simple test circuit (FO4 chain).

The Triple Sampling method required about 1.74 times the area and twice the power of the Double Sampling method, but can be designed to have zero errors. However the problem of generating an additional sampling edge in the center of the stability window is quite difficult to solve.

The I&S method can also be engineered to have zero false error rates. It occupies about 0.58X the area and consumes 6% more power than the Double Sampling method. It also doesn't need an extra edge unlike the Triple Sampling method. However, it is a semi analog technique and its integration with existing digital flows needs to be investigated further.

Our estimates show that the false error rates at ground level are sufficiently low for small chips and hence Double Sampling technique will be an adequate technique for detecting soft errors in most chips. However for large high performance chips, operating at high altitudes, the false error rates for the Double Sampling technique will become unacceptably high and one will need to use either the Triple Sampling checker or the I&S checker, both of which can be designed to yield zero false errors.

## References


[1] P.Hazucha et al. Neutron Soft Error Rate Measurements in a 90-nm CMOS Process and Scaling Trends in SRAM from 0.25-m to 90-nm Generation. IEDM, pp. 523-526, 2003.

[2] P. Shivakumar et al. Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic. International Conference on Dependable Systems and Networks, pp. 389-398, 2002.

[3] T. C. May and M. H. Woods. Alpha-particle-induced soft error in dynamic memories. IEEE Trans. Elec. Dev., Vol. 26, No. 1, pp. 29, 1979.

[4] R. Baumann. Soft Errors in Advanced Semiconductor DevicesPart I: The Three Radiation Sources. IEEE Transactions on Device and Materials Reliability, pp. 17-22, 2001.

[5] Y. Tosaka, H.Kanata, S.Satoh, T.Itakura. Simple Method for Estimating Neutron-Induced Soft Error Rates Based on Modied BGR Model. IEEE Electron Device Letters, Vol. 20, No. 2, pp. 89-91, 1999.

[6] N. Seifert et al. Radiation induced soft error rates of advanced CMOS bulk devices. IRPS, pp. 217-225, 2006.

[7] A. Taber and E. Normand. Single Event Upset in Avionics. IEEE Transactions on Nuclear Science, Vol. 40, No. 2, pp. 120-125, 1993.

[8] M.Nicolidis. Time redundancy Based Soft error Tolerance to Rescue Nanometer Technologies. VTS, pp. 86-94, 1999.

[9] Y.Tsiatouhas, A. Arapoyanni, D. Nikolos, Th. Haniotakis. A hierarchical Architecture for Concurrent Soft error Error Detection Based on Current Sensing. IOLTW, pp. 56, 2002.

[10] Y. Satish, B. Amrutur, R. Parekhji. Modified Stability Checking for On-line Error Detection. VLSID,pp. 787-792, 2007.

[11] D. Ernst et al. Razor: Circuit-level correction of timing errors for low-power operation. IEEE Micro, Vol. 24, No. 6, pp. 10-20, 2003.

[12] L. B. Freeman. Critical charge calculations for a bipolar SRAM array. IBM Journal of Research and Development, Vol 40, No. 1, pp. 119-129, 1996.

[13] D. Harminder, D. Sylvester, D. Blaauw. Gate-Level Mitigation Techniques for Neutron-Induced Soft Error Rate. Proceedings of the Sixth International Symposium on Quality Electronic Design, pp.175-180, 2005.