SETTOFF : A Fault Tolerant Flip-Flop for Building Cost-efficient Reliable Systems

Yang Lin and Mark Zwolinski
Electronics and Computer Science, University of Southampton
Southampton SO17 1BJ, UK
Email:{yl5g09,mz}@ecs.soton.ac.uk

***Abstract*—Conventional fault tolerance techniques either require big overheads or have limited reliability. We propose a novel fault tolerant flip-flop (SETTOFF) that addresses timing errors and soft errors in one cost-efficient architecture. In SETTOFF, most SEUs are detected by monitoring the illegal transitions at the output of a flip-flop and recovered by inverting the cell state. SETs, timing errors and the other SEUs are detected by a time redundancy-based architecture. For a 10% activity rate, SETTOFF consumes 35.8% and 39.7% more power than a library flip-flop in 120nm and 65nm technologies, respectively. It only consumes about 5.7% more power than the detection based RazorII flip-flop [1]. The result indicates that SETTOFF is suitable for building high reliable systems at lower cost than the traditional techniques.**

***Index Terms*—Fault tolerance, single-event upset, single-event transient, timing error.**

I. Introduction

Shrinking feature sizes, increasing operating frequency and supply voltage scaling have significantly reduced the reliability of electronic systems. Soft errors induced by high-energy particle strikes (such as cosmic neutron [2] and alpha particle strikes [3]) have become a concern for system reliability even at ground level. When struck, a sensitive node may produce a transient current pulse. If a struck node belongs to a storage cell, the pulse may reverse the cell state, resulting in a typical soft error: a single-event upset (SEU). When the node belongs to a logic gate, the transient current pulse may create a single-event transient (SET). An SET can turn into a soft error if it propagates and is sampled by a sequential element.

Error Correction Codes (ECC) are a traditional soft error tolerance technique [4]. Calculating and reading the ECC bits during each operation induces performance and power overheads. This overhead can be big in blocks that are frequently accessed, such as the register file in a processor. Triple Modular Redundancy (TMR) is another conventional technique [5], but the overhead (more than 300% area and power) is too great for consumer electronics. Other techniques such as parity coding and duplication only detect and therefore need to be combined with recovery schemes [6].

Another effective way to combat soft errors is to use hardened storage circuits, [7] [8] [9] [10] [11]. Although these approaches are more cost-efficient than the traditional TMR-latch [12], they have various drawbacks [7]. A time redundancy-based, cost-efficient, hardened architecture, Razor, was proposed for timing faults in a DVS system [13]. Razor cannot address soft errors. RazorII introduced a new latch that can detect both soft errors and timing errors [1]. However it is a latch-based storage cell and does not have any correction functionality. The combined recovery scheme in RazorII is again only suitable for recovering timing errors.

In this paper, we introduce a novel Soft Error and Timing error TOlerant Flip-Flop (SETTOFF) for the three types of errors (SETs, SEUs and timing errors). SETTOFF takes advantage of a Time Redundancy-based Detection technique (TRD) to detect SEUs occurring during an interval, and the SETs and timing errors occurring in the preceding logic blocks. SEUs occurring outside the TRD detection interval are interpreted as illegal transitions and are detected by a transition monitor, which then generates a signal to recover these SEUs on the fly by inverting the state of the flip-flop again. Since the errors detected by TRD are those occurring in the write operation of the flip-flop, they can be easily recovered by using a low-cost replay recovery mechanism. The simulation results show that for a 10% activity rate, SETTOFF consumes 35.8% and 39.7% more power than a conventional flip-flop in 120nm and 65nm technologies, respectively. It also requires 48 extra transistors. Although the overhead is bigger than that of RazorII flip-flop (28.5% power overhead and 31 extra transistors) [1], SETTOFF gives a better fault tolerance capability and is suitable for using in both timing- and safety-critical flip-flops and systems. Moreover, it also overcomes most of the drawbacks in the previous hardened storage architectures.

This paper is organized as follows, the next section describes the background to this work. In section III, we describe the design of SETTOFF. The verification results and analysis are discussed in section IV and we conclude in section V.

II. Background and Previous Work

A. Previous Published Hardened Storage Cells

Some fault-tolerant storage circuits have been published previously [7] [8] [9] [10] [11]. The main drawback of these circuits is that they can only tolerate the SEUs inside the latch other than the sampled errors which originate in the preceding logic blocks. Moreover, some architectures, [8] [9], are susceptible to particles with high energy [7], while others, [10] [11], cannot protect all the internal nodes of a latch [7]. Another approach called FERST, [7], can mitigate both SETs and SEUs, but can still be corrupted by timing errors.

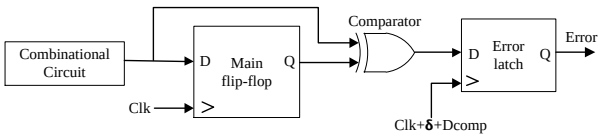

Fig. 1. Time redundancy based fault tolerant flip-flop [14].

B. Time Redundancy-Based Error Detection Technique (TRD)

Because SETs occurring in the logic blocks only manifest themselves for a limited period of time, and will be recovered automatically, Time Redundancy-based error Detection (TRD) moves hardware duplication into the time-domain [14] [15]. The technique is conceptually shown in Fig. 1. $Dcomp$ is the delay of the comparator. With no hardware duplication, TRD can detect SETs that are manifest on the input of the flip-flop with the maximum pulse width of $Dtr \leq \delta - Dsetup$ ($Dsetup$ is the setup time of the latch). Such SETs, if captured by the main flip-flop at $t0$, will recover at $t0 + \delta - Dsetup$ while the comparator will assert the error signal due to inconsistent input values. This architecture can also detect SEUs in the main flip-flop during $t0$ to $t0 + \delta$. Although TRD is cost-efficient, it has no correction ability. Moreover, the SEUs occurring in the main flip-flop after $t0 + \delta$ will escape detection. Unlike SETs, SEUs cannot be recovered until the flip-flop is overwritten by the next input value.

Razor uses an enhanced version of the TRD architecture for combating timing errors with a delay no greater than $\delta - Dsetup$ [13]. It adds a shadow latch to sample the input again at $t0 + \delta$ for the combined recovery mechanism. A sufficient δ is used to prevent the shadow latch from re-sampling the timing error. However, Razor does not consider the presence of soft errors which may corrupt the shadow latch. Thus the combined recovery mechanism may actually corrupt the system if the shadow latch is affected.

C. RazorII

Fig. 2 illustrates the architecture of the RazorII flip-flop and its operating principle [1]. The positive-edge sensitive latch is augmented with a transition detector (TD) which is controlled by a detection clock (DC) generator. The error detection is realized by detecting illegal transitions on the internal latch node N. To prevent valid transitions from being flagged as errors, DC disables TD within the period of the CLK-to-Q delay of the latch to allow the latch to capture its correct input state. An architectural replay recovery signal is generated by the asserted error signal.

While this architecture addresses both timing errors and certain soft errors with a small overhead, its SEU recovery ability limits its reliability. In RazorII, the architectural replay scheme re-fetches the instruction in the write back (WB) stage for correcting SEUs in the pipeline registers. Such a replay operation will re-write all the current data in the pipeline registers in which the SEUs will be overwritten before they contaminate other blocks in the system. The same recovery

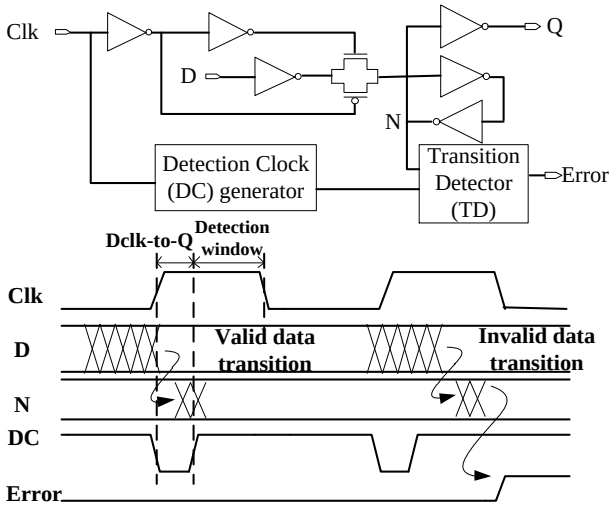

Fig. 2. RazorII Flip-Flop and the operation diagram [1].

scheme cannot apply to the registers that store the architectural state of the processor, such as the register file (RF). These registers may not be written in every cycle, but may be contaminated by SEUs at any time (e.g. in a store cycle much after a write operation cycle). Hence it is hard to target the replay point for re-writing the corrupted register before it contaminates the following logic. The RF is vulnerable to soft errors, [16], and hence needs to be protected with high priority. This makes RazorII difficult to use for fault tolerant systems.

Actually, [1], the RazorII flip-flops are only used for protecting timing-critical pipeline registers; safety-critical storage cells like RF and caches are protected against soft errors using ECC, which may induce more overhead. Another limitation is that the power consumed by the DC generator to provide a sufficient negative pulse may increase as the clock frequency increases. Therefore the RazorII flip-flop may have a much bigger power overhead in processors with higher speed.

III. Key Concepts of SETTOFF

In order to develop a low-cost fault-tolerant scheme with high efficiency, we propose a fault tolerant flip-flop, named SETTOFF. Three novel components of the SETTOFF are:

1) SETTOFF can detect both soft and timing errors. With a small overhead to the detection architecture, SETTOFF provides on-the-fly recovery for those SEUs (those occurring from the write operation of the flip-flop) that are hard to recover by the low-cost architectural replay technique. Therefore SETTOFF minimizes the recovery overhead by separating the recovery process into two.
2) SETTOFF can be used in both safety-critical and timing-critical systems (such as the DVS system). Soft errors and timing errors may occur simultaneously. Moreover, SETTOFF completely eliminates the need for using high-cost TMR or ECC techniques.
3) SETTOFF is a flip-flop based storage cell. It is easier to use in mainstream flip-flop based systems than latch-based cells.

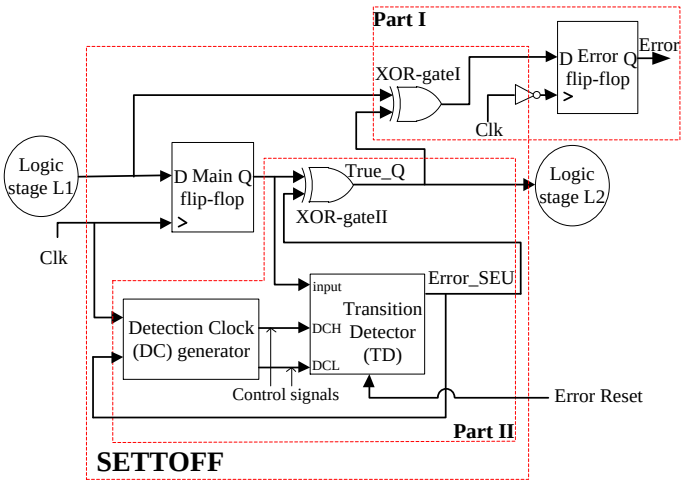

Fig. 3. The architecture of SETTOFF.

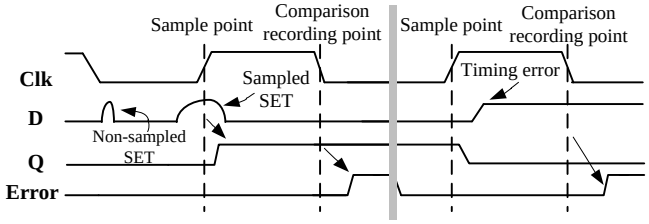

Fig. 4. The operation principle of Part I in SETTOFF.

A. Principle of Operation

Fig. 3 illustrates the architecture of SETTOFF. Part 1 is the TRD architecture. It contains XOR-gate I that compares the input and output (*True_Q*) of SETTOFF, and the error flip-flop which is driven by the falling clock edge. The delay element $\delta = Dpclk - Dxor\_gateI$, where $Dpclk$ is the period of the positive clock phase. Part 1 is responsible for detecting three types of error occurring during the write operation of the main flip-flop. As shown in Fig. 4, SETs on the output of Logic Stage L1 with a pulse width no greater than $Dpclk - Dxor\_gateI - Dsetup$ will be detected if captured by the main flip-flop. Timing errors with a delay no more than $Dpclk - Dxor\_gateI - Dsetup$ are also detected (Fig. 4). Moreover, SEUs occurring in the flip-flop during the positive clock phase will also be detected in Part 1. The error flip-flop generates a signal for architectural replay discussed in III-C.

Part 2 is responsible for detecting and recovering the SEUs that occur during the negative clock phase (i.e. out of the write operation). It comprises a transition detector (TD) at the output (Q) of the main flip-flop, a detection clock (DC) generator for TD, and XOR-gate II which propagates or inverts Q to *True_Q* according to the *Error_SEU* signal. Only those SEUs that corrupt the output of the main flip-flop need to be considered; others are masked. Two operation states are defined in Table I. In normal operation, the DC generator disables TD during the positive clock phase to avoid legal transitions being flagged as errors. Q propagates to *True_Q* since *Error_SEU* is zero. During the negative clock phase, TD detects the SEUs that reverse the state of the flip-flop and asserts the *Error_SEU* signal, which feeds into XOR-gate II and the DC generator. Therefore if Q is inverted by an SEU, XOR-gate II and the asserted *Error_SEU* signal will then invert Q back to the correct state (*True_Q*) at nearly the same time. Meanwhile, with the asserted *Error_SEU* signal, the DC generator generates control signals to enable TD during all clock phases.

TABLE I
THE OPERATION STATE OF PART II IN SETTOFF

State	Activities
Normal Operation (Error_SEU=0)	Propagate Q Enable TD during negative clock phase
Fault Operation (Error_SEU=1)	Invert Q Enable TD during all clock phase

The SETTOFF will not return to the normal operation state until TD is reset by the *Error Reset* signal, or TD detects the next transition in Q, which will assign *Error_SEU* back to zero. To explain this, we assume two circumstances as shown in Fig. 5(a) and Fig. 5(b):

(a) If the next transition is another SEU occurring during the fault operation, it will reverse the state of the main flip-flop back to the correct value. Hence TD switches the SETTOFF to the normal operation state to propagate Q to *True_Q* (Fig. 5(a)). In other words, an even number of SEUs during one cycle can correct the state of the flip-flop.

(b) If the next transition is caused by the SETTOFF sampling the next input, the former detected SEU will be overwritten and the flip-flop will operate normally (Fig. 5(b)).

A third circumstance can happen when the next input does not reverse the corrupted state of the main flip-flop. In this case, no transition occurs in Q therefore SETTOFF stays in the fault operation state which inverts the correct Q to generate a faulty output *True_Q*. However, the faulty output will not be read since it will be detected by Part 1 at the following falling clock edge. A replay signal will then be generated to recover the faulty output and reset the SETTOFF back to the normal operation state (Fig. 5(c)). Part 2 guarantees the output of the SETTOFF will never be corrupted by SEUs occurring in the negative clock phase; other faults are detected by Part 1.

Finally, Part 2 of SETTOFF is complementarily protected by Part 1. When a soft error corrupts the output of Part 2 (*Error_SEU*), Part 1 will detect the erroneous output of SETTOFF (*True_Q*) at the following falling edge of the clock and invoke the replay process, which prevents the error from contaminating the following logic. The only susceptible element in the fault-tolerant circuitry is the error flip-flop in Part 1. However, in most cases, the erroneous error indication signal that is generated by the error flip-flop only results in an unnecessary replay operation. Furthermore, since the error flip-flop can be easily shared within a system (see III-C), we can protect it by a conventional mechanism (such as ECC, TMR) which has acceptable overheads with respect to the system.

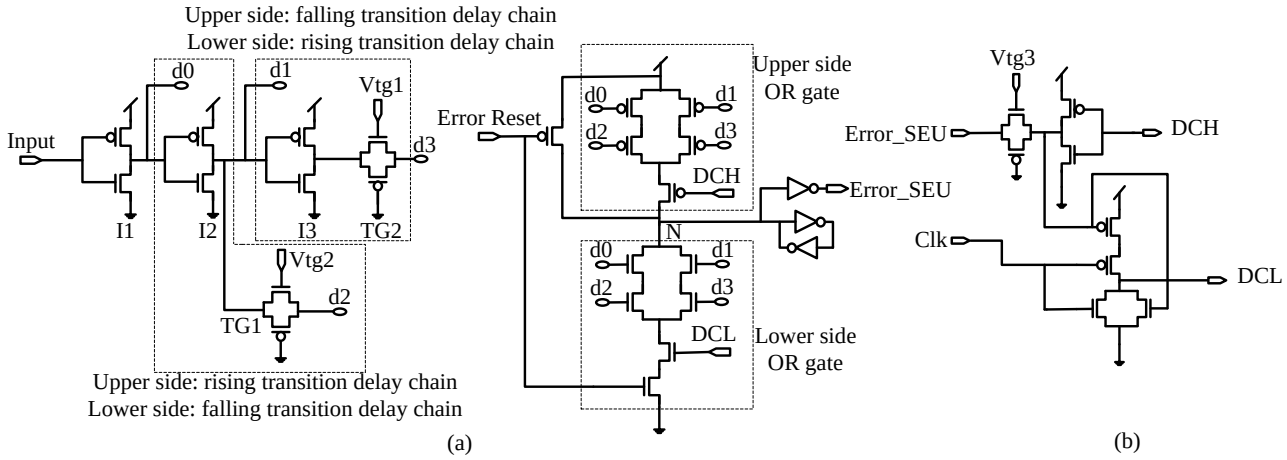

Fig. 6. The circuit schematic of SETTOFF. (a) The transition detector. (b) The detection clock generator

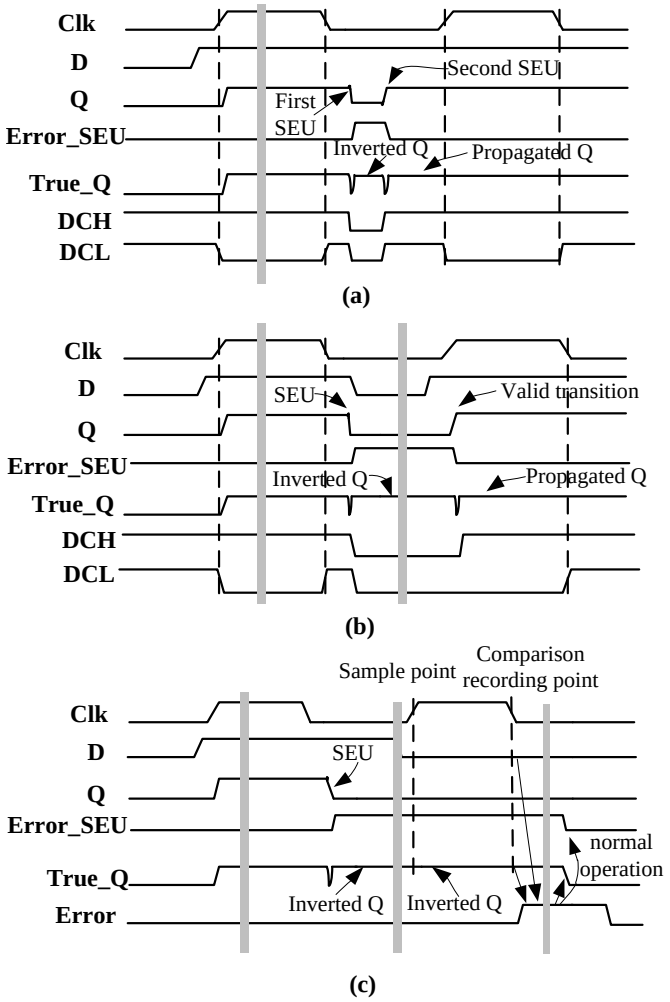

Fig. 5. The operating principle of Part II in SETTOFF.

B. Transistor Level Design of SETTOFF

The circuit schematic of the transition detector and the detection clock generator are shown in Fig. 6. The transition detector (Fig. 6(a)), is developed from the transition detector proposed for RazorII [1]. It requires two pulse generators to generate an 'implicit' pulse out of a rising and a falling transition at its input node, respectively. Each pulse generator requires a delay-chain which is formed by an inverter and a transmission gate, with an AND gate to generate the pulse. The pulse is then captured by the dynamic OR gates to switch the *Error_SEU* signal. Based on the transition detector in RazorII, another dynamic OR gate that contains two AND gates is added to the upper side of the circuit. The two control signals, DCH and DCL are the on/off switches for the circuit on upper and lower sides, respectively. The two sides work in turn in the two states shown in Table I. In the normal operation state, the upper side is switched off; DCL turns on the lower side during the negative clock phase. In this case, I3 and TG2, which with the AND gate, d1 and d3, act as the pulse generator for the rising transition. I2, TG1, d0 and d2 act as the pulse generator for the falling transition. In the fault operation state, the lower side is switched off and the upper side is switched on by DCH during all clock phases. This time I3 and TG2 are the delay chain for the falling transitions, while I2 and TG1 form the delay chain for the rising transitions. The operating principle of DCH and DCL is given in Fig. 5(a) and Fig. 5(b). The *Error_SEU* pin will switch between '0' and '1' in the presence of multiple transitions at the input of the transition detector.

The *Error Reset* signal pre-charges node N to set the *Error_SEU* pin to zero and the circuit to the normal operation state. It can be generated in a architectural replay process which is invoked by the *error* signal in the error latch. The cross-coupled inverter pairs are used to protect the dynamic node N from discharging or charging by the leakage current.

In the circuit of the detection clock generator (Fig. 6(b)), a delay is created by the transmission gate for generating DCL and DCH signal. This allows sufficient time for the

dynamic node N to be charged or discharged when a transition is captured by the delay chains. The voltage supply for all the transmission gates in SETTOFF is tunable for controlling the delay they generate. However, the normal supply voltage (1.2V) is used when validating the design.

C. Incorporating SETTOFF in a Processor

As mentioned above, a longer positive clock phase in SETTOFF (δ) in the TRD architecture can detect SETs with wider pulse widths and timing errors with longer delays. Hence, the detection capability of SETTOFF can be tuned by altering the duty cycle of the clock without affecting the operating speed. However, this constrains the minimum propagation delay of the combinational logic, and buffers may need to be inserted, which induce area and power overheads.

The errors detected by the TRD architecture are easy to recover with a low-cost architectural replay mechanism. This is because these errors are detected in the write operation of the flip-flop. The instruction executing the faulty write operation can be easily targeted by check-pointing the program counter. The faulty instruction can be then re-fetched and re-executed to overwrite the detected errors. Specifically, the scheme can be realized by simply amending the replay mechanism which often already exists in high-performance processors [1]. Moreover, if a detected error is not recovered after a certain number of replays, we can consider it as an timing error and adjust the voltage or frequency during the next replay process. The replay mechanism will be investigated in future work.

The error flip-flop can be shared by multiple SETTOFFs. Only the upper side OR gate in the TD and two XOR gates are added compared to the RazorII architecture (the DC generators of SETTOFF and Razor have different architectures but the same number of transistors). The area overhead increase is small. Additionally, the XOR gate added on the output of the main flip-flop can induce certain performance penalties as it increases the propagation delay of the main flip-flop. An alternative way to minimize the performance overhead is to replace the output inverter inside the main flip-flop by the correction XOR gate, then use TD to monitor the internal node rather than the output of the main flip-flop.

IV. Verification and Comparative Analysis

A. Simulation Results

The SETTOFF was implemented in two technology libraries, 65nm and 120nm, for verification and evaluation. A conventional flip-flop is picked from each technology library for the main flip-flop in SETTOFF. We used Spice to simulate the circuits. Two independent current sources are added on the output and input node of the main flop-flop for injecting SEUs, SETs and timing errors, respectively. SETs with different pulse widths, timing errors with different delay values, and SEUs with different appearance time were injected. Several example simulation waveforms (Fig. 7, Fig. 8) are shown. In Fig. 7, a SET is injected and is captured by the main flip-flop. The error signal is asserted on the falling edge of the clock due to the inconsistent values on the input and output of SETTOFF. Fig.

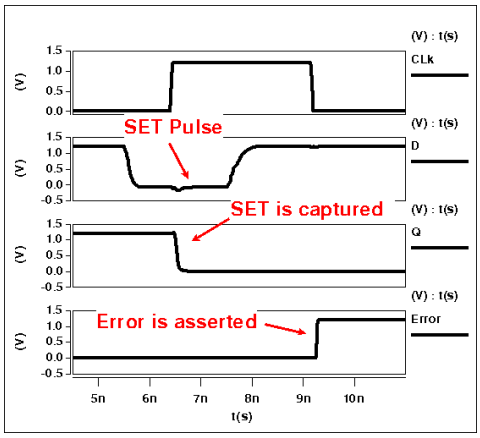

Fig. 7. The operation diagram when a SET is injected.

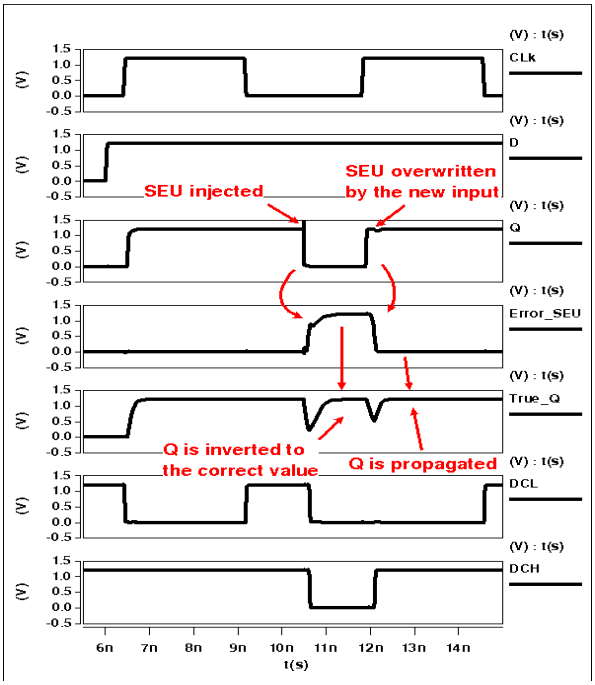

Fig. 8. The operation diagram when a SEU is injected.

8 shows an SEU injected during the negative clock phase. TD detects the illegal transition and asserts the *Error_SEU* to invert the output of SETTOFF to the correct value. The circuit is switched back to the normal operation state after the SEU is overwritten by the next input. The SETTOFF (*True_Q*) generates two glitches instead of being corrupted by the SEU.

B. Comparative Analysis

We compare SETTOFF with Razor flip-flops in terms of reliability and fault-tolerance overheads – Table II.

The power consumption overhead of SETTOFF is measured at an operating frequency of 185MHz and a supply voltage of 1.2V. With the 120nm technology library, SETTOFF consumes 28.4% extra power when data does not switch on the main flip-flop, and 102.0% extra power when data

TABLE II
COMPARATIVE RESULTS OF THE FAULT TOLERANT OVERHEAD

	SETTOFF	Razor	RazorII
Power Overhead[1] (For a 10% Activity Rate)	35.8%	30.0%	28.5%
Area Overhead (Extra Transistors Required)	48	54	31

switches. Hence for a 10% activity rate, SETTOFF consumes 35.8% more power than a conventional flip-flop. According to [1], a RazorII flip-flop consumes 28.5% more power than a conventional flip-flop for the same voltage, frequency and activity rate in a 130nm technology[1], while Razor consumes 30.0% more power. Therefore SETTOFF consumes about 5.7% more power than RazorII, and about 4.5% more power than Razor. In addition, with the same activity rate, SETTOFF consumes 39.7% more power (136.0% for non-switching data and 29.0% for switching data) in the 65nm technology. Moreover, without the need to provide a sufficient negative phase to cover the CLK-to-Q delay, we can use smaller transistors for the transmission gate in the DC generator compared to RazorII. This eliminates the big power overhead that might be induced by the large transmission gate at a much higher clock frequency, e.g. 1GHz. 48 extra transistors are added to the main flip-flop for SETTOFF. The Razor and RazorII flip-flops require 54 and 31 extra transistors, respectively.

In terms of the reliability, SETTOFF can detect SETs with pulse widths no bigger than $Dpclk - Dsetup - Dcomp$, timing errors with delay no greater than $Dpclk - Dsetup - Dcomp$, and all the SEUs during all clock phases. It also provides an on-the-fly recovery capability for the SEUs occurring during the negative clock phase. RazorII flip-flop only provides detection and has a similar detection capability as SETTOFF for all three types of error. Razor possesses the same timing error tolerant capability but cannot address soft errors.

In respect to an electronic system, SETTOFF induces a desirable extra overhead comparing to the Razor flip-flops. However, it can potentially tolerate the soft errors which may corrupt Razor flip-flop, and the SEUs occurring out of the write operation of the storage cell (such as the SEUs occurring in the store cycle) which may corrupt RazorII flip-flop and the TRD technique.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we present a new flip-flop architecture named SETTOFF, for tolerating both soft errors and timing errors efficiently. Beside the detection ability for SETs, SEUs and timing errors, we add a moderate overhead to the detection architecture to provide an on-the-fly circuit-level recovery capability for SEUs occurring in the write operation of the flip-flop. A single SETTOFF requires 48 extra transistors and with a 10% activity rate, it consumes 35.8% and 39.7% more power than a conventional flip-flop in 120nm and 65nm technology, respectively. If combined with a low-cost architectural replay mechanism, SETTOFF can reduce the soft error rate of a system significantly by tolerating all the detected errors. It is applicable for both timing-critical and safety-critical parts in the system, and completely eliminates the need for traditional high-cost fault tolerant techniques such as ECC and TMR. SETTOFF is suitable for mainstream flip-flop based designs that have either DVS or fault-tolerant requirements. Future research will focus on implementing a high-performance processor with SETTOFF to assess system-level trade-offs.

[1]We consider the difference between 120nm and 130nm to be minor.

REFERENCES

[1] S. Das, C. Tokunaga, S. Pant, W.-H. Ma, S. Kalaiselvan, K. Lai, D. Bull, and D. Blaauw, "Razorii: In situ error detection and correction for pvt and ser tolerance," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 1, pp. 32 –48, jan. 2009.

[2] J. Ziegler and W. Lanford, "The effect of sea level cosmic rays on electronic devices," in *Solid-State Circuits Conference. Digest of Technical Papers. 1980 IEEE International*, vol. XXIII, feb 1980, pp. 70 – 71.

[3] T. May and M. Woods, "Alpha-particle-induced soft errors in dynamic memories," *Electron Devices, IEEE Transactions on*, vol. 26, no. 1, pp. 2 – 9, jan 1979.

[4] C. L. Chen and M. Y. Hsiao, "Error-correcting codes for semiconductor memory applications: A state-of-the-art review," *IBM Journal of Research and Development*, vol. 28, no. 2, pp. 124 –134, march 1984.

[5] W. Van Gils, "A triple modular redundancy technique providing multiple-bit error protection without using extra redundancy," *Computers, IEEE Transactions on*, vol. C-35, no. 7, pp. 623 –631, july 1986.

[6] G. Memik, M. Kandemir, and O. Ozturk, "Increasing register file immunity to transient errors," in *Design, Automation and Test in Europe, 2005. Proceedings*, march 2005, pp. 586 – 591 Vol. 1.

[7] M. Fazeli, S. Miremadi, A. Ejlali, and A. Patooghy, "Low energy single event upset/single event transient-tolerant latch for deep submicron technologies," *Computers Digital Techniques, IET*, vol. 3, no. 3, pp. 289 –303, may 2009.

[8] R. Naseer and J. Draper, "The df-dice storage element for immunity to soft errors," in *Circuits and Systems, 2005. 48th Midwest Symposium on*, aug. 2005, pp. 303 – 306 Vol. 1.

[9] M. Baze, S. Buchner, and D. McMorrow, "A digital cmos design technique for seu hardening," *Nuclear Science, IEEE Transactions on*, vol. 47, no. 6, pp. 2603 –2608, dec 2000.

[10] L. Wang, S. Yue, and Y. Zhao, "Low-overhead seu-tolerant latches," in *Microwave and Millimeter Wave Technology, 2007. ICMMT '07. International Conference on*, april 2007, pp. 1 –4.

[11] M. Omana, D. Rossi, and C. Metra, "Latch susceptibility to transient faults and new hardening approach," *Computers, IEEE Transactions on*, vol. 56, no. 9, pp. 1255 –1268, sept. 2007.

[12] M. Favalli and C. Metra, "Tmr voting in the presence of crosstalk faults at the voter inputs," *Reliability, IEEE Transactions on*, vol. 53, no. 3, pp. 342 – 348, sept. 2004.

[13] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: a low-power pipeline based on circuit-level timing speculation," in *Microarchitecture, 2003. MICRO-36. Proceedings. 36th Annual IEEE/ACM International Symposium on*, dec. 2003, pp. 7 – 18.

[14] L. Anghel and M. Nicolaidis, "Cost reduction and evaluation of a temporary faults detecting technique," in *Design, Automation and Test in Europe Conference and Exhibition 2000. Proceedings*, 2000, pp. 591 –598.

[15] M. Nicolaidis, "Time redundancy based soft-error tolerance to rescue nanometer technologies," in *VLSI Test Symposium, 1999. Proceedings. 17th IEEE*, 1999, pp. 86 –94.

[16] N. Mehdizadeh, M. Shokrolah-Shirazi, and S. Miremadi, "Analyzing fault effects in the 32-bit openrisc 1200 microprocessor," in *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*, march 2008, pp. 648 –652.

[17] B. Narasimham, M. Gadlage, B. Bhuva, R. Schrimpf, L. Massengill, W. Holman, A. Witulski, and K. Galloway, "Test circuit for measuring pulse widths of single-event transients causing soft errors," in *Microelectronic Test Structures, 2008. ICMTS 2008. IEEE International Conference on*, march 2008, pp. 142 –146.