A New Bulk Built-in Current Sensor-Based Strategy for Dealing with Long-Duration Transient Faults in Deep-Submicron Technologies

Rodrigo Possamai Bastos, Giorgio Di Natale, Marie-Lise Flottes, Bruno Rouzeyre

A New Bulk Built-in Current Sensor-Based Strategy for Dealing with Long-Duration Transient Faults in Deep-Submicron Technologies

Rodrigo P. Bastos, Giorgio Di Natale, Marie-Lise Flottes, Bruno Rouzeyre
LIRMM (Université Montpellier II / CNRS UMR 5506)
Montpellier, France
{possamaiba, dinatale, flottes, rouzeyre}@lirmm.fr

***Abstract*—Today's deeply-scaled technology-based integrated circuits are highly sensitive to soft error, tending to be even more in the future. In fact, emerging critical issues are related to transient faults that now can be as long as circuits' clock periods. This work presents a novel improved strategy based on bulk built-in current sensors that is able to cope with long-duration transient faults. Our cost-effective approach is a concurrent error detection scheme with recovery procedure, and rather than existing similar strategy, it has faster correction latency and uses less recovery resources.**

Keywords-concurrent error detection schemes, fault attacks, long-duration transient faults, soft errors

I. INTRODUCTION

In the past, reliability was an issue only for safety-critical systems. Nowadays, device size reduction, power supply restriction, increased operating frequency, and high circuit density have made it a design challenge for any integrated circuit. Hence, modern systems demand higher resilience against many new issues like radiation-induced particles, aging problems, and environmental and device parameter variations. Alpha particles and cosmic neutrons, for instance, are able to generate transient voltage variations even at ground level – the so-called transient faults (TFs) that can provoke soft errors (SEs) by inverting stored results of circuit operations. Another today's related topic is the growing need for secure systems – like smartcards – where TFs can be intentionally induced as a form of fault-based attack to bypass security mechanisms and retrieve confidential information such as stored secret keys.

Related researches until early 2000's were focused essentially on protecting systems against TFs affecting memory elements, which were considered the system's most vulnerable circuits. Hence, many concurrent error detection and/or correction mechanisms were proposed to mitigate SEs induced by TFs in memory cells – the direct SEs. Nevertheless, in the last decade, more sensitive deeper-submicron technologies as well as new manners of performing fault injection-based attacks – e.g. differential fault analysis [1][2] – have also pushed for countermeasures against indirect SEs originated from TFs in system's combinational logic circuits.

Today's deeper-submicron technology-based ICs can even suffer with other severe TF effects related to their clock periods. In fact, it is possible that fault's durations are comparable or even longer than circuit's clock cycles. As illustrated in Fig. 1, faults, which were short in 180-nm technology-based 10-inverter chains, become long by reaching several clock cycles in 32-nm technology-based counterparts [3]. Such long-duration transient (LDT) faults have clearly a much higher probability of not being masked, and so they also stand a greater chance of producing system failures. This emerging problem therefore introduces supplementary difficulties to protect the circuits. Most of the existing protection techniques are indeed not optimized to deal with such a phenomenon, and so they either require very high overheads or they are not sufficiently efficient. Hence, novel mitigation solutions need to be devised.

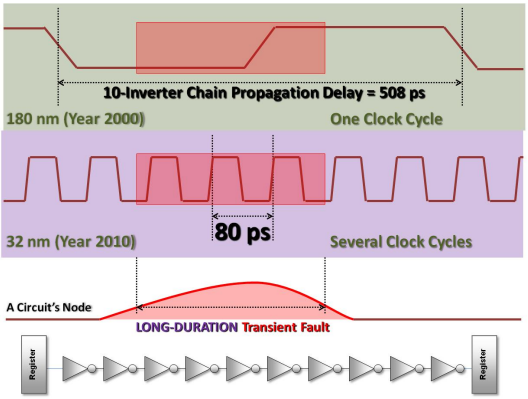

Figure 1. A long-duration transient fault in a 32-nm technology used in 2010's commercial microprocessors

The current trend in low-cost solutions to efficiently cope with new challenges of deep-submicron technologies is applying protection techniques at different abstraction levels of the design [3][4][5][6]. Concurrent error detection (CED) mechanisms are designed at lower abstraction levels while recovery-based error correction procedures (ECPs) at higher levels. This strategy takes advantage of recovery circuits that are already present in modern microprocessors to recompute instructions in case of branch misprediction [5][6]. Then, only CED mechanisms need to be implemented, and as they are optimized at lower levels, they can monitor circuit's fault-prone nodes much closer than higher-level schemes could. Hence the detection can be done early, as soon as the fault happens, preventing more critical failure scenarios such as the induction and propagation of multiple errors to other clock cycles, stages, or parts of the system. This approach, therefore, allows adding higher detection capability to the system at the expense essentially of the CED devices. The penalties of extra clock cycles to perform ECPs occur only in fault scenarios.

The effects of LDT faults in state-of-the-art CED schemes are investigated in section II of this paper. We show that classic and other very recent approaches either are not cost-effective or they are not so efficient to deal with LDT faults. On the contrary, a low-cost and efficient strategy for mitigating LDT faults is proposed in [6] by using a CED scheme based on bulk built-in current sensors (BBICS). We demonstrate in section III.A that such a strategy requires a recovery circuit with at least two files to save two previous cycles' input status, which would be later necessary in case of ECP. Then, we propose in section III.B a novel improved BBICS-based strategy for dealing with LDT faults that requires only one file and penalizes the system with less extra cycles than [6]'s approach.

II. Long-Duration Transient-Fault Effects on State-of-the-Art CED Schemes

Every CED technique employed to cope with TFs makes use of some form of redundancy such as:

1) Spatial redundancy by which the whole circuit is, for instance, duplicated and an additional module is able to identify the presence of an error;

2) Redundancy of information in which the application of error detection codes replaces the duplication of the entire circuit, thus reducing surface and power consumption overheads at the cost of smaller error detection capability; and

3) Time redundancy at which circuit's operations are evaluated twice with the same data and a comparison of the results can allow identifying non-permanent faults. It attempts to reduce the amount of extra hardware at the expense of additional time.

Therefore, CED schemes typically compare two redundant results of which at least one must be safe to permit the detection of errors. If for instance one result fails, the comparison provides an error flag such as an error signal at logical level "1". It would indicate the occurrence of a TF in the circuit during a certain Clock Cycle that we define as CC-Start-TF. On the other hand, such an error flag needs to remain at a steady state enough time to be correctly sampled by a recovery circuit, which would be responsible either to configure an ECP, or else to save input status of the cycle in case of no sampled error flag. Thereby, after TF vanishing, saved input status from CC-Start-TF could be recomputed in the Clock Cycle that we name as CC-ECP.

In subsection II.A a classical spatial redundancy-based CED example is analyzed in terms of inner characteristics and capability to detect LDT faults. Similar behaviors with respect to such faults can be extrapolated to techniques based on redundancy of information since they are also a derived form of spatial redundancy. On the other hand, subsection II.B discusses some particular issues related to the capability of time redundancy-based architectures in detecting LDT faults. Eventually, subsection II.C highlights LDT-induced problems of very recent CED techniques based on transition detectors, which are schemes devised by using concepts of time redundancy.

A. Long-duration transient fault effects on Duplication with Comparison (DWC)

In DWC schemes the original logic block is duplicated and the output of each block is captured into registers, one for each block as shown in Fig. 2 (a). A comparison is performed in order to detect possible differences between the outputs of the two blocks. In a fault-free circuit, the clock cycle duration is longer than the longest path within the logic blocks, thus assuring the correctness of sampling operation. The comparison is done after the registers, therefore input values of the comparator reaches a steady state until the end of a clock cycle.

This type of technique guarantees high level of error detection, also in case of a LDT fault. Fig. 2 (b) shows an example in which a TF starting in CC-Start-TF achieves Logic Block's output for duration comparable to two clock cycles. TF1, which indeed affects one of the two logic blocks, succeeds in reaching the register without being masked electrically, logically or by latching window, then it generate an indirect SE. During the following Clock Cycle named as CC-Post-TF, the comparator notifies the presence of such an error that will be processed by an ECP in the subsequent cycle "CC-ECP". In case of an even longer-duration TF2 illustrated in Fig. 2 (c), the circuit will switch between CC-ECP and CC-Post-TF states till TF disappearing. Anyway, it will be able to detect the indirect SE, and eventually to correct it.

The main drawback of this architecture is related to its cost in terms of additional area. In fact, its overhead is more than 100%.

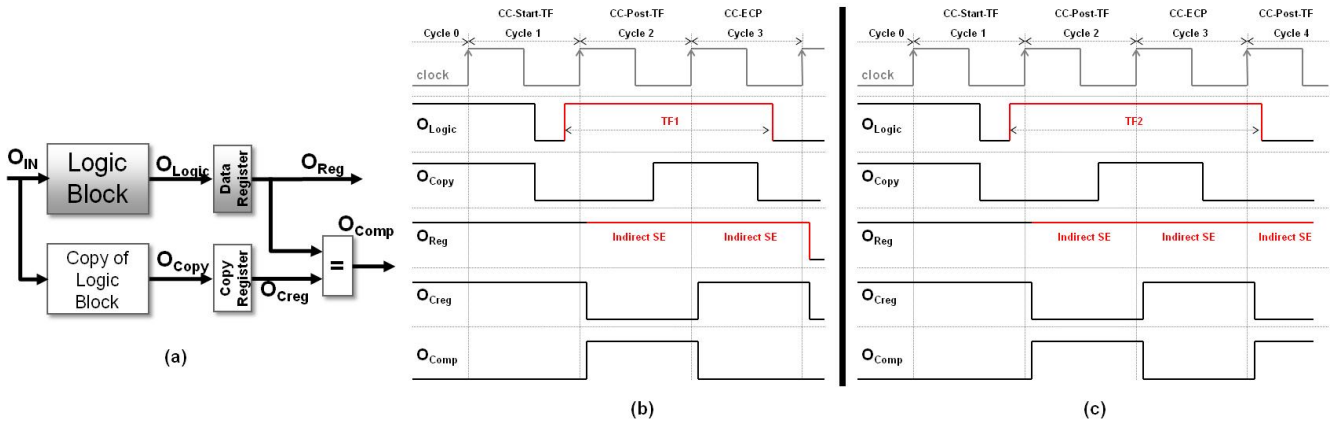

Figure 2. DWC General Scheme (a) and its behavior under a LDT fault "TF1" (b), and an even longer-duration transient fault "TF2" (c)

B. Long-duration transient fault effects on Time Redundancy (TR) schemes

In TR schemes the basic idea is to repeat twice the same computation with the same hardware in two different instants, and to compare the two results [7].

TR architectures can be implemented at different abstraction levels: system, algorithmic, micro-architectural, logical, or electrical. At system and algorithmic levels, the basic idea is to start the computation with the input data, to store the result in a temporary variable, and then to repeat the same computation with the same data, and to check the results. The system delivers the results only if they match. In these cases, the delay δ between two computations is usually in the order of several clock cycles. Therefore, TR at such abstraction levels can be very effective also for LDT faults, with low additional area costs but huge and permanent performance penalties.

However, in case of micro-architectural, logical, and electrical levels, TR architectures can exhibit some issues. In this case, the circuit where TR takes effect is composed of a single combinational logic block, and the register at its outputs is duplicated as illustrated in Fig. 3 (a). The additional Copy Register receives the output data on O_{Logic} after a delay δ. The outputs of the two registers are compared, and in case of discrepancy the circuit raises an error flag. The delay δ between the two computations less flip-flop's $T_{Set\text{-}up}$ and T_{Hold} corresponds to the maximum TF duration that can be detected by this solution.

The first limitation of this scheme is that the delay δ must be short in order to not much penalize the circuit's operating frequency. Thus, it allows covering only small range of short-duration TFs. If longer-duration TFs are required to be detected, the delay δ has to be increased, and then the performance is even more penalized.

Fig. 3 (b) details a case of a short-duration TF that is longer than the delay δ, and so resultant indirect SE may be not detected because $T_{Set\text{-}up}$ and T_{Hold} of both registers are not respected. Note that the TF transition occurs at the end of the Cycle 1. It then violates the $T_{Set\text{-}up}$ of both registers. In fact, the registers' behaviors become non-deterministic, and in the worst case their values may result in indirect SEs. As both registers contain an indirect SEs at logical level "1", the comparator is not able to detect such an error.

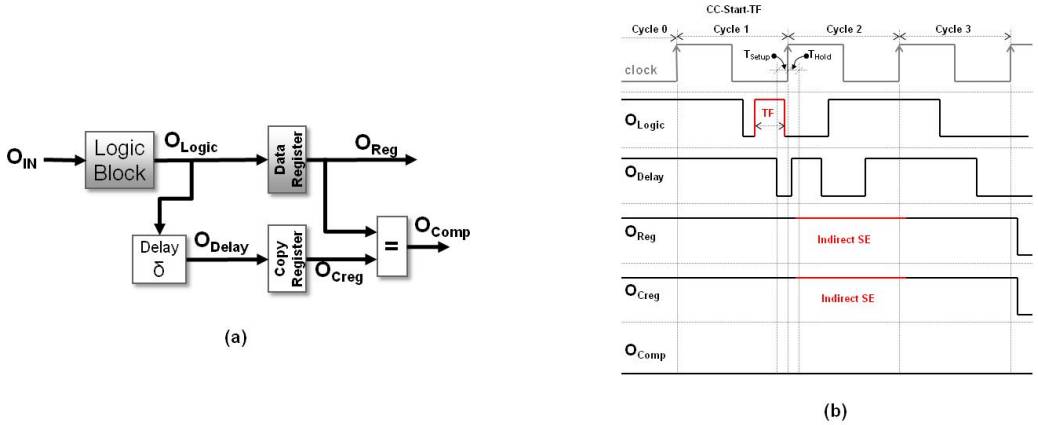

Figure 3. TR general scheme (a) and its behavior under a particular TF case (b)

C. *Long-duration transient fault effects on Transition Detector (TD)-based schemes*

The last cost-effective CED-based solutions [8][9][10][11] have been devised by using transition detectors. They aim at coping with environmental and manufacturing variability-induced delay faults that cause timing errors, but they are also claimed to mitigate TFs. The basic principle of these techniques is to detect illegal transitions of some specific nodes of the circuit within a detection time window "DW" on which the data should be stable. The task of identifying a transition is done through TR concepts by which the target node is analyzed at two different instants [7], such as section II.B further explains. The major problem of TD-based schemes is when LDT faults or even shorter-duration TFs cover completely such a time window "DW" reserved to identify anomalous transitions, and so TD is not able to distinguish a fault from a legal event.

Let us take a TD-based CED example proposed in [10] to further explain such a problem. This approach indeed detects and counts TFs affecting flip-flop inputs. As shown in Fig. 4 (a), the scheme monitors inputs "D_i" (i = 0 to N) of all target flip-flops. This solution becomes low cost only because a large number of signals "D_i" are combined into a single one "P", such as the parity, by using a Xor tree. Thereby just one block "TF Detector" is required.

TF Detector includes a Transition Detector Block and a Sticky Block. The first one generates a signal "TD" that presents logical level "0" if P is already stable. However it renders a pulse at logical level "1" if P switches. Sticky Block works to count only once a TF affecting D_i since it is possible that the delay difference between P and O provokes an additional glitch at signal "TD". Moreover, in function of detection clock "DC", Sticky Block is responsible to enable the counter only during a monitoring time interval defined as the detection window "DW". DW is mandatory to distinguish legal logic transitions from anomalous event like a TF.

Fig. 4 (b) shows D_0 being affected by a fault "TF1" during a time interval on which it should be stable. TF1 reflects at P after propagation through Xor Tree. Then, there is a transition at P that is able to be detected since it appears at TD during DW.

DW is indeed defined as $T_{Margin} + T_{Set\text{-}up} + T_{Hold}$ assuming that during a fault-free operation signals Di should reach steady values before a flip-flop's set-up time and maintain it after the clock edge for a flip-flop's hold time. A time margin "T_{Margin}" is typically added to take into account inter-die variations, clock's jitter and skew. Furthermore, DW must be delayed by a time interval equals to $D_P + D_{TD}$ (i.e. Xor Tree's delay + Transition Detector Block's delay) since transitions of D_i are propagated to Transition Detector Block's output "TD".

It is important to note that whether the fault does not lead TD switching during DW, it cannot be detected. This is illustrated in Fig. 4 (c). The fault "TF2" affects data Q_0 causing an indirect SE as in the first example; however it arises before the timing interval on which signal D_0 should be stable. Consequently, P's response at TD is out of DW, and so the indirect SE is not detected. The only solution to mitigate LDT faults like TF2 would be harshly penalizing circuit's nominal operating frequency in order to increase DW.

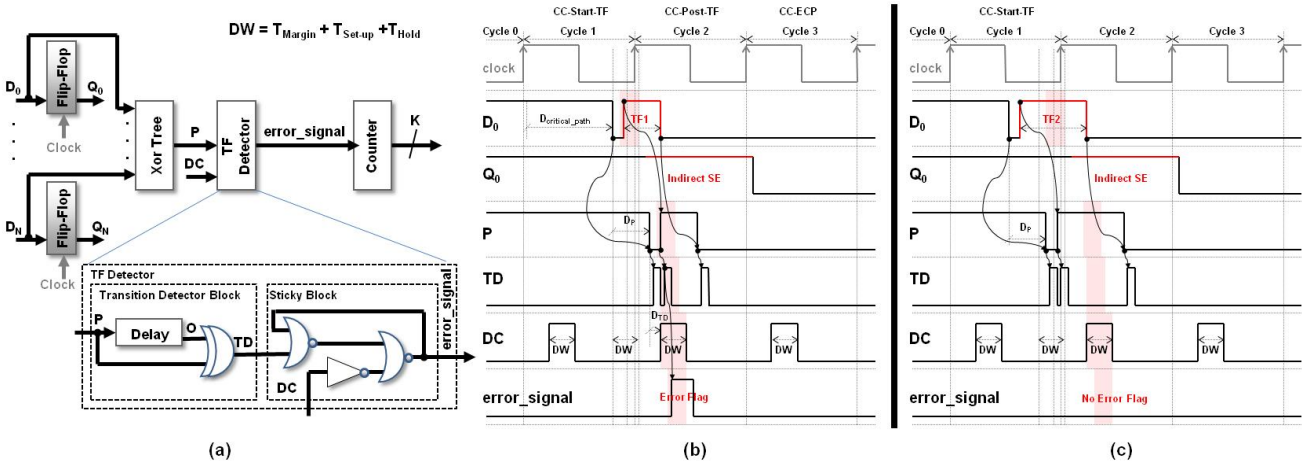

Figure 4. TD-based TFMS (a) generating an error flag for a transient fault "TF1" (b) and not detecting a longer-duration TF2 (c)

III. CURRENT-SENSOR-BASED CED SCHEMES FOR MITIGATING LONG-DURATION TRANSIENT FAULTS

A very low-cost and efficient CED solution for coping with LDT faults is proposed in [6] by using Built-In Current Sensors connected to the Bulk of transistors (BBICS). Results of implementations shows that area overheads imposed by BBICS mechanisms to protect adder circuits can be up to 13.4 % without impact on the system's operating frequency. The costs therefore are considerably smaller than the ones due to classic CED schemes such as discussed in section II. Moreover, BBICS approach is much more efficient to deal with multiple faults and LDT faults.

BBICS are implemented as Fig. 5 (a) shows. These mechanisms detect anomalous transient currents flowing between any reverse biased drain junction and the bulk of circuits perturbed by events like radiation-induced particles. BBICS indeed takes advantage of the fact by which such currents are negligible in fault-free scenarios, and they are much higher than leakage

currents flowing through biased junctions during fault scenarios [12]. BBICS's transistors are sized to latch an error flag for abnormal currents that generate voltage pulses (TFs) representing a SE risk. BBICS's outputs in Fig. 5 (a) result in Fig. 5 (b)'s error_signal. These outputs, which keep an error flag in TF scenarios, need to be reset in order to detect other TFs. This is done through inputs "RST" as soon as ECPs are started by recovery circuits.

Fig. 5 (b) illustrates the BBICS-based CED scheme interacting with a recovery circuit that acts to repeat a faulty cycle (i.e. CC-Start-TF) and then correcting the error (i.e. performing CC-ECP defined in section I). Area overheads related with recovery circuits are minimal whether circuit's architecture has already a machine to repeat operations in branch-misprediction situations. Furthermore, performance costs through extra clock cycle executions occur only when a fault is detected, and so the circuit operates without penalty in fault-free scenarios.

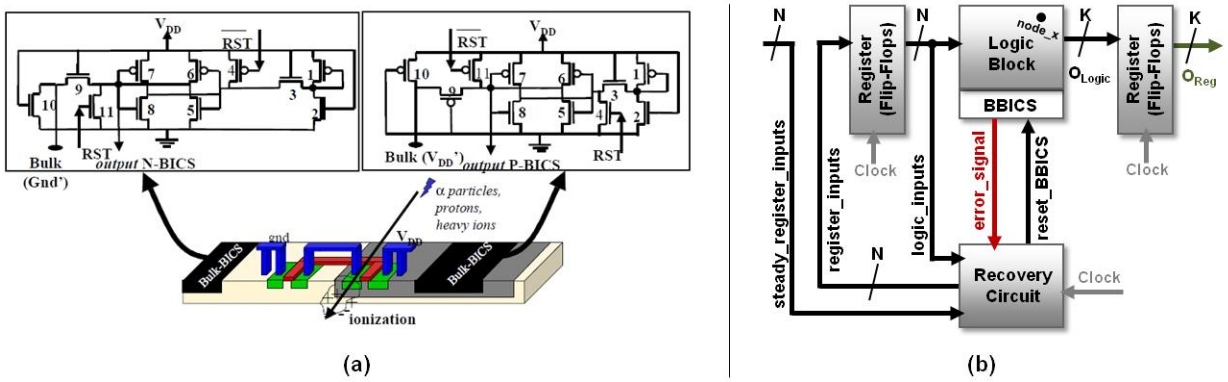

Figure 5. Bulk Built-In Current Sensors (BBICS) with reset (figure (a) from [6]) and their CED scheme with recovery in (b)

Fig. 5 (b)'s error_signal is a recovery circuit's asynchronous primary input due to the natural features of TFs, which can happen at any instant. Then, a strategy for dealing with this type of error signal is mandatory to prevent metastability problems as well as to accurately configure ECPs. Next subsection III.A analyzes the strategy proposed in [6] for LDT scenarios, and subsection III.B presents our novel improved strategy.

A. Strategy for Dealing with Error Signals of BBICS-based CED Schemes

The detection and correction scheme suggested in [6] for mitigating LDT faults requires at least a recovery circuit similar to Fig. 6 (a)'s illustration to efficiently perform ECPs. The highlighted blocks are resources that may be already present in certain architectures to recompute operations. This machine saves logic_inputs of two previous clock cycles by using two files with N latches each one. Then, if BBICS indicates an error flag at error_signal, the recovery circuit is able to restore saved input status (saved_logic_inputs) of state executed two clock cycles ago the instant at which the error flag is identified and registered at recomputing_signal.

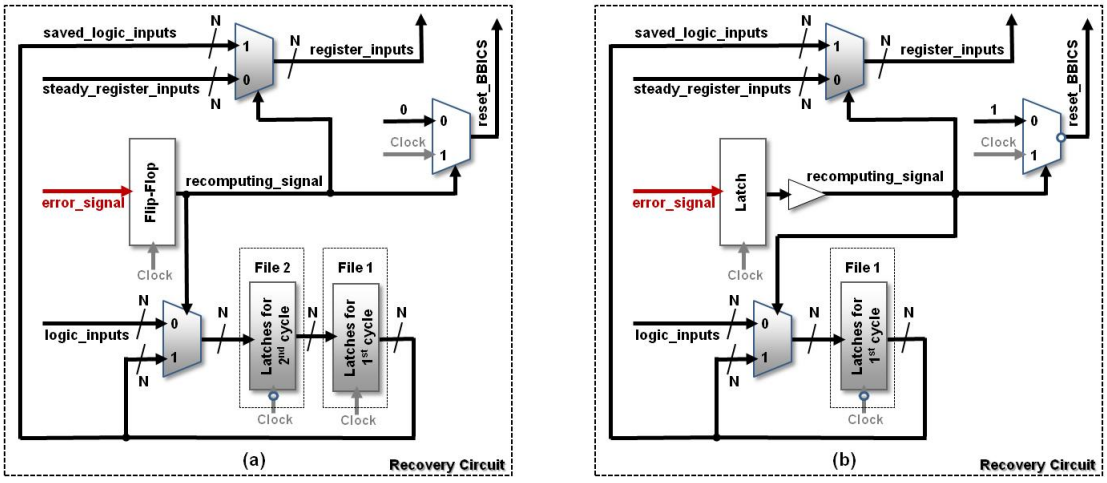

Figure 6. (a) is a recovery circuit to perform [6]'s strategy, and (b) is another one to use our novel improved strategy (section III.B)

Signals' behaviors of [6]'s strategy are shown in Fig. 7's diagram. It illustrates a LDT fault "TF1" starting on a node_x of Fig. 5 (b)'s Logic Block during Cycle 1. BBCIS thus raises in Cycle 2 an error flag at error_signal after a maximum response time "RT" equals to 50 % of the circuit's clock period "T". However, this error flag is only registered at recomputing_signal in Cycle 3. Then, as such a recomputing_signal achieves steady logical level "1" during Cycle 3, input status of Cycle 1 (i.e. CC-

Start-TF's logic_inputs kept at saved_logic_inputs) are restored on logic_inputs at the beginning of Cycle 4. Nevertheless, as TF1 spans up to Cycle 3, BBICS raises again an error flag that keeps recomputing_signal at "1" in Cycle 4, and so input status of Cycle 1 is restored once more in order to recompute them in Cycle 5 as an ECP. TF1 therefore penalizes the system with a latency of four extra clock cycles by using [6]'s strategy.

Note that even though calibrating BBICS with faster RT, there are chances of TFs starting in Cycle 1, for instance, not to raise recomputing_signal in Cycle 2. Hence, [6]'s strategy necessarily needs at least two files with N latches to save input status of two previous cycles. Moreover, if RT is greater than T, more than two file are required. Therefore, the slower the RT the greater is the number of required files. On the other hand, Lisboa et al. in [6] notice that the slower the RT the lower is the area overhead with BBICS mechanisms. In fact, RT depends on the bulk resistance and the number of transistors monitored by each sensor [12]. RT must be thus calibrated in function of the number of recovery circuit's files. Consequently, the lower can be the area overhead with BBICS mechanisms if the greater is the number of available files. The major problem is only when the recovery circuit has no a minimum number of files to properly accomplish ECPs.

Furthermore, Fig. 7 also shows a fault "TF2" that makes recomputing_signal's flip-flop metastable. Then, such a recomputing_signal results in an unknown value that may be, for instance, "0", and so TF2 would penalize the system with a latency of three extra cycles instead of two whether the resultant value was "1".

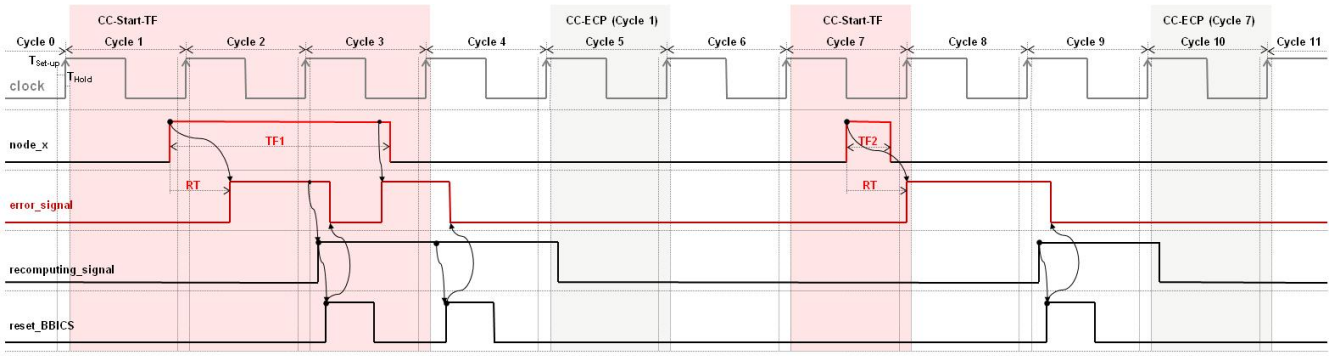

Figure 7. Functional behavior of [6]'s strategy to cope with faults TF1 and TF2

B. Novel Improved Strategy for Dealing with Error Signals of BBICS-based CED schemes

We propose in this subsection a considerable improvement of [6]'s strategy discussed in IV.A. Our improved strategy indeed requires a recovery circuit such as Fig. 6 (b) illustrates. This machine has a smaller number of resources than that one required by [6]'s strategy. In fact, only a file is necessary since our approach need to save logic_inputs of just one clock cycle ago the instant at which an error flag is identified and registered at recomputing_signal. This optimization is possible only by using a latch to sample error_signal at clock's falling edge.

Fig. 8 gives further details about our proposed strategy. It illustrates the mitigation of faults "TF1", "TF3", "TF2", and "TF4". Note that the same faults "TF1" and "TF2" tested in Fig. 7 for [6]'s strategy are also analyzed for our approach. Unlike the reactions of Fig. 7's recomputing_signal due to TF1 and TF2, this signal in Fig. 8 raises earlier during Cycle 2 and Cycle 8 instead of respectively Cycle 3 and Cycle 9 in Fig. 7. In fact, Fig. 8's recomputing_signal gets steady logical level "1" after clock's falling edge in Cycle 2 and Cycle 8, then logic_inputs of CC-Start-TF are restored earlier at the beginning of Cycle 3 and Cycle 9 on which ECPs are performed. As TF1 lasts until Cycle 3, logic_inputs of Cycle 1 are restored again at the beginning of Cycle 4, and so an ECP is now properly re-executed without the fault presence. Therefore, TF1 and TF2 penalizes the system respectively with three and two extra cycles instead of four and three by using [6]'s strategy. Our improved strategy shows thus requiring smaller latencies to complete ECPs due to short or long-duration transient faults. In fact, our approach advances ECPs by anticipating the identification of error flags at clock's falling edge instead of rising edge as [6]'s strategy does.

Note that there are two important design constraints which ensure the anticipation of ECPs as well as the use of only one file to save previous input status. In order to explain the first constraint, let us initially take Fig. 8's limit fault scenario in Cycle 5. TF3 starts on the border on which Cycle 4 leaves of being perturbed, and so from such an instant, which is defined as T_{Hold} after clock's rising edge, Cycle 4 is not necessary to be recomputed later by an ECP. Then, if clock's high Pulse Width "hPW" is ensured sufficiently longer than RT for clock's falling edge sampling correctly TF3-induced error flag; any TF starting before TF3 has certainly its error flag sampled in Cycle 5, and so only previous Cycle 4 has to be saved. Equation (1) below defines this first constraint by using a $T_{MarginFall}$ as additional time margin for variations in clock's falling edge operations (jitter and skew), and manufacturing and environmental variabilities:

$$hPW > RT + T_{Hold} + T_{Set-up} + T_{M\,\mathrm{arg}\,inFall} \qquad (1)$$

On the other hand, the second design constraint is related to low Pulse Width "lPW" that complements hPW to make a clock period "T". In fact, by taking similar TF3 scenario but with TF starting a little after, lPW must last enough time in Cycle 5 to ensure the worst case when an error flag at error_signal causes metastability in recomputing_signal's latch after clock's edge falling, but recomputing_signal stabilizes at logical level "1". In this situation, the condition below in (2) must be respected in order to the circuit configures properly an ECP that is executed in Cycle 6. $T_{MarginRise}$ is similar to $T_{MarginFall}$ but for clock's rising edge, D_{Mux2x1}, D_{Latch}, and D_{Buffer} are respectively delays of register_inputs's mux, and recomputing_signal's latch and buffer.

$$lPW > T_{Hold} + T_{M\arg inFall} + D_{Latch} + D_{Buffer} + D_{Mux2x1} + T_{M\arg inRise} + T_{Set-up} \qquad (2)$$

By using the fact that T is equal to hPW plus lPW, (1) and (2) result in the design condition expressed in (3). Note that RT and T are adjustable whether derived equations (4) and (5) are respected. D_{Logic} is the Logic Block's longest delay.

$$RT + T_{Hold} + T_{Set-up} + T_{M\arg inFall} < hPW < T - \left(T_{Hold} + T_{M\arg inFall} + D_{Latch} + D_{Buffer} + D_{Mux2x1} + T_{M\arg inRise} + T_{Set-up}\right) \qquad (3)$$

$$RT < T - \left(2 \cdot T_{Hold} + 2 \cdot T_{M\arg inFall} + D_{Latch} + D_{Buffer} + D_{Mux2x1} + T_{M\arg inRise} + 2 \cdot T_{Set-up}\right) \qquad (4)$$

$$T > T_{Hold} + D_{Logic} + T_{M\arg inRise} + T_{Set-up} \qquad (5)$$

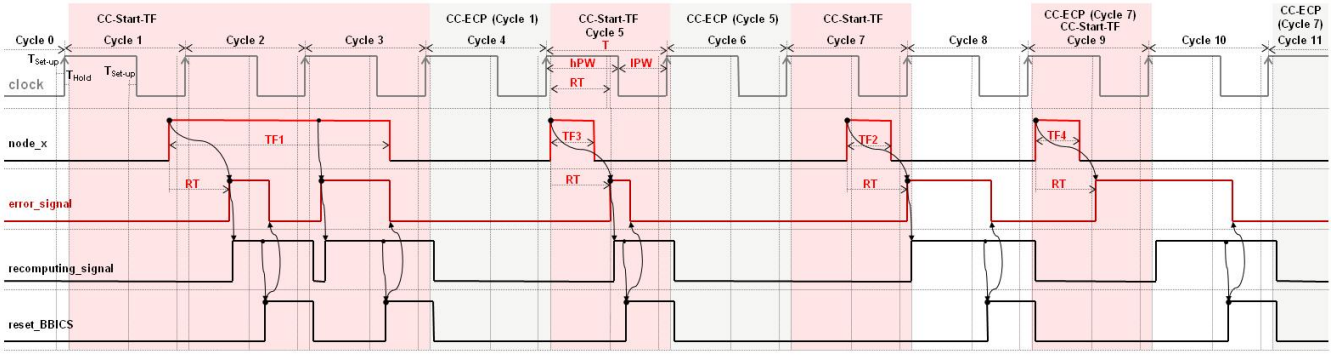

Figure 8. Functional behavior of our novel improved strategy to cope with faults "TF1", "TF3", "TF2", and "TF4"

IV. CONCLUSIONS

An analysis of LDT fault effects on CED schemes shows that classic approaches impose high overheads to cope with such a phenomenon. On the other hand, very recent CED techniques that can result in low overheads are not so efficient for dealing LDT faults. Then, emerging CED solutions based on BBICS appear as a very attractive alternative in terms of efficiency and overhead. In this paper we have proposed a novel improved strategy for dealing with error signals of BBICS-based CED schemes. We have shown that our approach, which uses the clock's falling edges to sample error flags from short or long-duration TFs, allows reducing scheme's correction latency by one cycle. Moreover, it also permits to use only a file to save previous status. Our solution therefore requires much smaller recovery circuits than the state-of-the-art BBICS-based strategy for dealing with LDT faults in deep-submicron technology-based integrated circuits.

REFERENCES

[1] C.N. Chen, and S.M. Yen, "Differential Fault Analysis on AES Key Schedule and Some Countermeasures," in Proc. ACISP, v. 2727 of LNCS, 2003, pp. 118-129.
[2] P. Dusart, G. Letourneux, and O. Vivolo, "Differential Fault Analysis on A.E.S," in Proc. ACNS, v. 2846 of LNCS, 2003, pp. 293-306.
[3] C. Lisboa, M. Erigson, and L. Carro, "System level approaches for mitigation of long duration transient faults in future technologies," in Proc. ETS, IEEE, 2007, pp. 165-170.
[4] C. Albrecht et al., "Towards a Flexible Fault-Tolerant System-on-Chip," in Proc. ARC, VDE Verlag GMBH, 2009, pp. 83-90.
[5] S. Z. Shazli, and M. B. Tahoori, "Transient Error Detection and Recovery in Processor Pipelines," in Proc. DFT, IEEE, 2009, pp. 304-312.
[6] C. Lisboa et al., "Using Built-in Sensors to Cope with Long Duration Transient Faults in Future Technologies," in Proc. ITC, IEEE, 2007, pp. 1-10.
[7] M. Nicolaidis, "Time redundancy based soft-error tolerance to rescue nanometer technologies," in Proc. VTS, IEEE, 1999, pp. 86-94.
[8] K. Bowman et al., "Energy-efficient and metastability-immune resilient circuits for dynamic variation tolerance," IEEE JSSC, v. 44, n. 1, pp. 49–63, Jan. 2009.
[9] S. Das et al., "RazorII: In situ error detection and correction for PVT and SER Tolerance," IEEE JSSC, v. 44, n. 1, pp. 32–48, Jan. 2009.
[10] D. Rossi, M. Omanã, and C. Metra, "Transient fault and soft error on-die monitoring scheme," in Proc. DFT, IEEE, 2010, pp. 391–398.
[11] D. J. Palframan, N. S. Kim, M. H. Lipasti, "Time Redundant Parity for Low-Cost Transient Error Detection," in Proc. DATE, IEEE, 2011.
[12] E. H. Neto et al., "Using Bulk Built-in Current Sensors to Detect Soft Errors," IEEE Micro, v. 26, n. 5, pp. 10–18, Sep. 2006.