Timing Error Tolerance in Nanometer ICs

S. Valadimas, Y. Tsiatouhas[*] and A. Arapoyanni

University of Athens, Dept. of Informatics and Telecommunications, 15784 Athens, Greece
*University of Ioannina, Dept. of Computer Science, 45110 Ioannina, Greece

***Abstract*—Timing error tolerance turns to be an important design parameter in nanometer technology, high speed and high complexity integrated circuits. In this work, we present a low cost, multiple timing error detection and correction technique, which is based on a new Flip-Flop design. The proposed design approach provides timing error tolerance at the small penalty of one clock cycle delay in the circuit operation for each error correction. In addition, it is characterized by very low silicon area requirements compared to previous design schemes in the open literature. The proposed technique has been applied in a 90nm pipeline design of a digital FIR filter and the simulation results validated its efficiency.**

Keywords: Timing failures, Timing errors, Error detection and correction, Timing error tolerance.

I. Introduction

CMOS technology evolution and integrated circuits and systems complexity explosion, in the nanometer era, result in an ongoing difficulty to achieve adequate reliability levels and keep the cost of testing within acceptable bounds [1-2]. The device size scaling, the power supply reduction and the operating frequency increase affect circuits' noise margins and reliability. In this context, the probability of transient faults generation increases making hard to limit error rate levels within specifications.

Various mechanisms like crosstalk, power supply disturbance or ground bounce have been accused for timing error generation. The increased path delay deviations, due to process variations, and the manufacturing defects that affect circuit speed may also result in timing errors that are not easily detectable (in terms of test cost) in high frequency and/or high device count integrated circuits (ICs). Although very complex testing procedures are followed, these are not sufficient to exercise the huge number of paths in modern circuit designs, and consequently they are not capable to effectively screen out all timing related defective ICs. Thus, a considerable part of defective ICs may escape the fabrication tests. Additionally, and for the same reasons, timing verification turns to be a hard task escalating the probability of timing failures in a design. Furthermore, modern systems running at multiple frequency and voltage levels may suffer from an increased timing error rate due to numerous environmental and process related as well as data dependent variabilities that may affect circuit performance. In addition, transistor aging problems significantly impact the performance of nanometer circuits resulting in the appearance of timing errors early in the circuit lifetime [3-4]. Such examples are the *Negative Bias Temperature Instability* (*NBTI*) induced aging of PMOS transistors and the *hot-carrier injection* (*HCI*) induced aging of NMOS transistors, which increase the threshold voltage over time increasing so the path delays [5]. From the above, it is evident that concurrent on-line testing techniques for timing error detection and correction are becoming mandatory in order to achieve acceptable levels of error robustness and meet reliability requirements. Besides, dynamic voltage scaling (DVS) techniques, for low power operation, may perform more efficiently tolerating timing errors by exploiting error detection and correction mechanisms to overcome increased error rates [6], [7].

Timing failures in a combinational logic block result in delayed responses at its outputs. Such a delayed response arrival, after the triggering edge of the clock signal that drive the memory elements at the outputs of the combinational block, will produce an erroneous value and the generation of a timing error on the data stored in the pertinent memory element. A number of error detection techniques have been proposed in the open literature [8]–[14]. These sense the delayed circuit response and provide error tolerance using time redundancy approaches. A well known error detection scheme is based on the use of a comparator that is realized by a simple XOR gate [10]–[12]. The monitoring circuitry consists of an additional memory element plus a XOR gate for every memory element (main latch or Flip-Flop) in the design. The secondary memory element is clocked by a delayed version of the system clock that feeds the main memory element. This delay is equal to the maximum signal delay (d_{max}) that must be tolerated in order to achieve an acceptable level of timing error rate, plus the setup time of the used memory elements (t_{su}). Thus, the secondary memory element captures a delayed version of the data stored in the main memory element. In the presence of a timing error the stored data in the two memory elements differ, while the secondary memory element holds the correct delayed response of the combinational logic. The XOR gate "compares" the contents of the two memory elements and in case of discrepancy it raises its output to high indicating the error detection. The local error indication signals are collected by an OR gate (realized as an OR tree) to generate a global error indication signal. This signal is exploited to achieve error tolerance by performing a retry procedure after error detection. During the retry operation the period of the system clock must be increased to provide the necessary time for correct response evaluation.

In this work we present a low cost timing error detection and correction scheme, that is based on a new Flip-Flop topology. Moreover, we introduce a pipeline architecture to exploit this Flip-Flop and provide timing error tolerance in a design. The paper is organized as follows. In Section II, relevant timing error detection and correction techniques, presented in the open literature, are discussed. Next, in Section III, the new design technique is introduced and its operation is analyzed. Section IV provides simulation results

978-1-4244-7723-4/$26.00 ©2010 IEEE

from the application of the proposed approach in a pipeline design of a FIR digital filter. Finally, Section V concludes this work.

II. Earlier Design Solutions

A pipeline architecture (named *Razor*) with timing error detection and correction capabilities, targeting the substantial energy reduction of integrated circuits exploiting dynamic voltage scaling, has been presented in [6]-[7] and modified in [15]. According to this architecture, the stage registers are constructed using the Razor Flip-Flops (see Fig. 1). A Razor Flip-Flop consists of the main system Flip-Flop plus an assistant shadow latch, a multiplexer (MUX) and a XOR gate. The shadow latch captures, with a proper delay with respect to the main Flip-Flop, the responses of the combinational logic. The XOR gate acts as a comparator and compares the outputs of the main Flip-Flop and the shadow latch after this time interval. In case of a difference (error detection), the error correction mechanism is activated, which redirects the input of the Main Flip-Flop to receive the correct data of the shadow latch and provide them to the subsequent logic stage during the next, recovery, clock cycle.

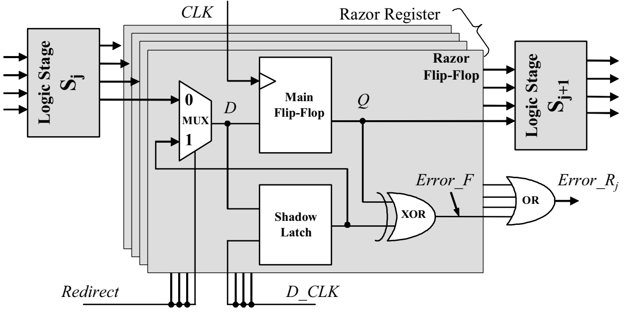

Fig. 1: The Razor Flip-Flop

Recently, in [16] and [17], the Time Dilation pipeline architecture has been introduced for timing error detection and correction. The stage registers are constructed using the Time Dilation Flip-Flop, shown in Fig. 2, which consists of a Scan Flip-Flop, a multiplexer (MUX-B) and a XOR gate. This design approach reduces the silicon area cost by eliminating the shadow memory element of previous solutions. Instead, the two multiplexers are dynamically modified to work as a latch (mux-latch) during the normal mode of operation. After an error detection by the XOR gate, the Main Flip-Flop is fed by the correct data stored in the mux-latch for error correction.

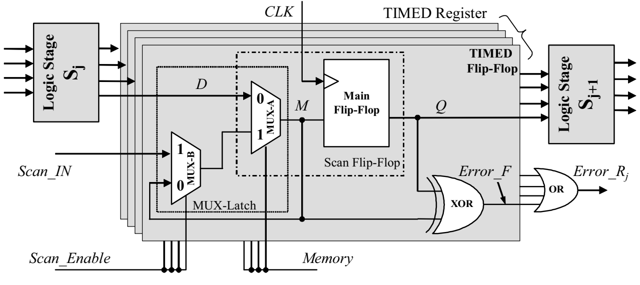

Fig. 2: The Time Dilation Flip-Flop

III. The proposed timing error detection and correction mechanism

A. The error detection and correction Flip-Flop

Fig. 3(a) illustrates the new Error-Detection-Correction Flip-Flop (EDC Flip-Flop) that is suitable for detection and correction of timing errors. Apart from the original Flip-Flop (Main Flip-Flop), it consists of a XOR gate, a Latch and a multiplexer (MUX). The XOR gate compares the *D* input and the *Q* output of the Main Flip-Flop and provides the result to the Latch. The Latch drives the select signal of the MUX at the outputs of the Main Flip-Flop. Depending on the comparison result within a specified time interval, either the *Q* or the Q_{bar} signal of the Main Flip-Flop is passed at the MUX output. By default the *Q* signal feeds the subsequent logic. Briefly, the proposed timing error detection and correction technique operates as follows. Suppose that a timing error is detected at one or more inputs of the combinational logic stage S_{j+1}, due to a delayed response of the previous stage S_j. Thus, the response of S_{j+1} will be erroneous and must be corrected. To achieve error correction, the evaluation time of the circuit is extended by one clock cycle and S_{j+1} is fed with the valid complement values of the input signals where a timing error has been detected.

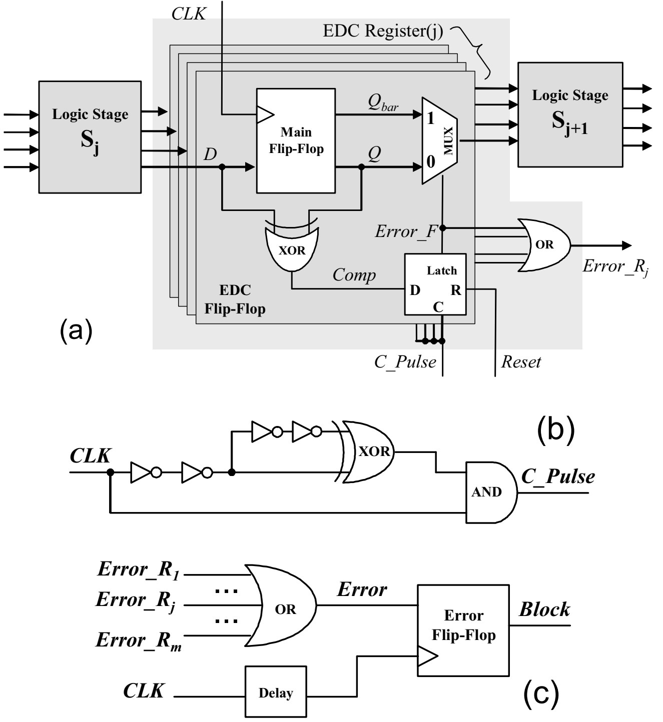

Fig. 3: a) The proposed EDC Flip-Flop, b) The pulse generator and c) Clock blocking signal generation

Initially, the Latch's output *Error_F* is reset to zero so that by default the *Q* output signal of the Main Flip-Flop feeds the subsequent logic stage. In the error free case the comparison result is a low value at the output of the XOR gate after the triggering edge of the clock signal *CLK*. This value is captured by the Latch which retains the selection of the *Q* output signal of the Main Flip-Flop, which carries the correct value, to pass the MUX and feed the subsequent logic stage S_{j+1}. However, in the presence of a timing fault in logic stage S_j, a delayed signal arrives at the *D* input of the Main Flip-Flop after the triggering edge of the clock signal *CLK*. In that case, a timing error is captured at the *Q* output of the Main Flip-Flop and erroneous data are provided to the subsequent logic stage S_{j+1}. In addition, the *Q* value differs from the *D* value. The XOR gate detects this difference and raises its output *Comp* to high. The Latch captures and holds this response selecting the Q_{bar} output signal of the Main Flip-Flop, which now carries the correct value, to pass the MUX and feed the subsequent logic stage S_{j+1}. This way the error is corrected.

A clock pulse *C_Pulse* is used to capture the comparison result of the XOR gate in the Latch (memory state when *C_Pulse* is low). This pulse can be generated locally in every register from the *CLK* signal using a Pulse Generator like the one illustrated in Fig. 3(b). Thus, the routing overhead of an extra clock signal is relaxed. The AND gate in Fig. 3(b) ensures that a single pulse will be generated only at the first phase of every clock cycle. The time interval between the *CLK* triggering edge and the pulse deactivation is the maximum detectable signal delay. Moreover, the pulse width is equal to the time required by the Latch to capture the comparison result. Every signal transition at the *D* input of an EDC Flip-Flop between the triggering edge of *CLK* and the falling edge of *C_Pulse*, is considered as a delayed response. So the circuit design must guarantee that there are not signal transitions at the inputs of EDC Flip-Flops, within this time interval, in order to be able to provide timing error tolerance.

However, in order to provide the error correction capability, extra time is required by the S_{j+1} logic stage to perform its evaluation with the corrected input values. For that reason the error indication signal *Error_F* is used to block the clock signal from feeding the logic (exploiting a global clock gating technique) during the clock cycle next to the one where the error has been detected. This way, a single clock cycle is provided for state recovery. To achieve this, the *Error_F* signals of the EDC Flip-Flops in a register (j) generate the register's error indication signal $Error\_R_j$ through a local OR gate. Next, all registers' $Error\_R_j$ signals are collected by a second OR gate which generates the global error indication signal *Error*, as it is shown in Fig. 3(c). The *Error* signal is captured by a Flip-Flop (the Error Flip-Flop) and its output signal *Block* is used for clock gating. The Error Flip-Flop is clocked by a delayed copy of the clock signal CLK. This delay is equal to the time required for the generation of the *Error_F* signal and its propagation through the pair of OR gates to the Error Flip-Flop.

B. Circuit operation

Fig. 4 illustrates the timing diagrams for the operation of the EDC Flip-Flop. In clock cycle (i) the response of logic stage S_j is within the timing specifications of the circuit (fault free case). This means that after the triggering edge of the clock *CLK*, input *D* and output *Q* of the Main Flip-Flop have the same values. So the signal *Comp* of the XOR gate is low and the same stands for both signals *Error_F* and $Error\_R_j$ after the clock pulse *C_Pulse*. Consequently, the MUX retains the predefined selection of the *Q* signal to feed the subsequent logic stage S_{j+1} with correct data. In that case the circuit operation remains unaltered.

In the next cycle (i+1) a timing fault occurs due to a timing failure in stage S_j. The data captured by the Main Flip-Flop are erroneous and a timing error appears at its *Q* output. So the response of stage S_{j+1} at the next cycle (i+2) will be also erroneous. Moreover, due to the fault, a transition occurs at the *D* input of the Main Flip-Flop, inside (i+2) cycle, just after the triggering edge of the clock *CLK* and before the activation of the clock pulse signal *C_Pulse*. The XOR gate detects the difference between the signal values on *D* and *Q* and raises its output *Comp* to high. Then, after the clock pulse *C_Pulse*, the Latch captures this high value setting the *Error_F* signal to high. As a result, the Q_{bar} signal is selected to pass the MUX and feed with correct data the subsequent logic stage S_{j+1}. Thus, the error at this specific input of the subsequent stage is corrected. The same stands for every other input where an error has been detected. The rest already correct inputs remain unaltered. Note that the error correction is achieved without the need to recalculate the response of the failing stage S_j.

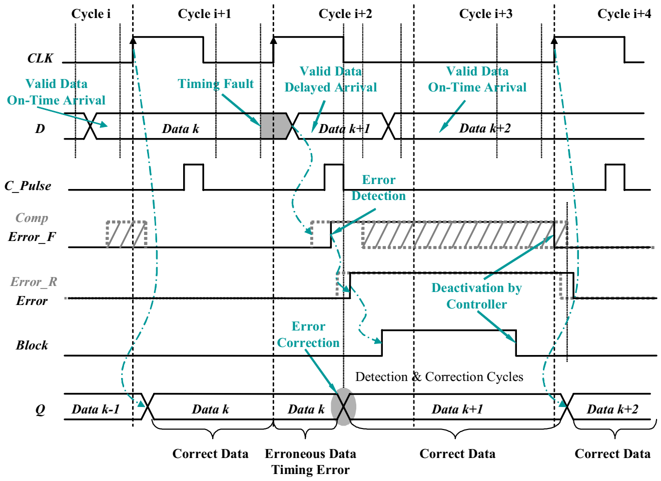

Fig. 4: Timing diagrams

In parallel the *Error_F* signal activates the register error indication signal $Error\_R_j$, through the OR gate, which collects all Flip-Flops' error indication signals. Finally the $Error\_R_j$ signal activates the global error indication signal *Error* (see Fig. 3c) which is captured by the Error Flip-Flop raising the *Block* signal to high. Thus, the clock signal *CLK* is blocked for one cycle (i+3) in order to provide the required time for the correct evaluation of stage S_{j+1}. This is an one cycle penalty for correction. At the end of the correction cycle (i+3) a control logic resets the Error Flip-Flop setting the *Block* signal to low and releasing the clock signal *CLK*. In addition, the Latch in the EDC Flip-Flop is also reset and the system returns to the standard operation for the next clock cycle (i+4) and until the next error detection.

C. Pipeline recovery

Every error detection is succeeded by a pipeline state recovery action. As it is illustrated in Fig. 5(a), a clock gating technique is used for pipeline recovery. In case of one or more timing errors, the clock is blocked for the next clock cycle exploiting the *Block* signal of the Error Flip-Flop. Then, those stages (e.g. LS3 in Fig. 5(b)) that initially received erroneous input data due to a timing fault in a previous stage (LS2), they recalculate their responses with corrected input data during the time interval of the extra clock cycle (correction cycle). The rest stages remain inactive retaining at their outputs the correct responses. Note that there is no need for the failing stage LS2 (the stage where the timing fault occurs) to recalculate its response since the correct response is automatically retrieved by the EDC Flip-Flop. A simple control logic counts one clock cycle and then releases the *CLK* signal by the activation of the *Release* signal which resets the Error Flip-Flop. The *Release* signal is also used to generate the *Reset* signal which resets the Latches in the EDC Flip-Flops at the end of the correction cycle.

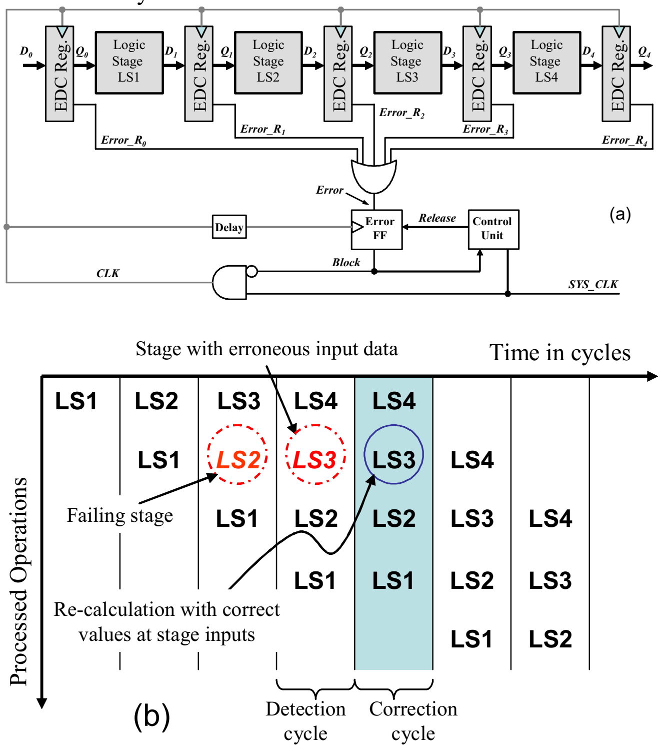

Fig. 5: Pipeline recovery, a) architecture and b) operation

The proposed pipeline error detection-correction architecture can tolerate any number of errors in any number of stages within a clock cycle, since all stages are capable to recalculate their responses with correct data at their inputs during the extra clock cycle. In case that one or more stages fail in each clock cycle, the pipeline will continue to run at half of the normal speed.

D. Discussion

A main characteristic and an advantage of the proposed technique, with respect to the Razor and the Time Dilation techniques, is that a standard Flip-Flop is replaced by the new EDC Flip-Flop, only if it is at the end (output) of a critical path (slow path) in a logic stage, that is a path susceptible to be affected by a timing fault. In case of Razor and Time Dilation, the pertinent error detection and correction Flip-Flops are used to replace every standard Flip-Flop in a stage register where timing error protection is required for at least one Flip-Flop. Thus, the proposed solution reduces drastically the silicon area cost providing the same timing error detection-correction efficiency as in the earlier approaches. In addition, the area overhead of the OR gate at the output of an EDC register is less than this of the corresponding OR gates for the Razor and Time Dilation topologies due to its smaller fan-in. For the same reason, its speed performance is improved. The area and the performance are further improved when a Domino design style is used for this OR gate. Finally, the rest circuitry (the Error Flip-Flop and the Control logic) is shared on the whole pipeline and thus its cost is insignificant. The area overhead related to this kind of circuitry is also present in the Razor and the Time Dilation topologies.

Another advantage of the proposed EDC Flip-Flop scheme is that the extra multiplexer is not placed at the input side of the Main Flip-Flop, so that it does not involuntarily introduce an additional delay to the critical path under monitoring, as it is the case in the Razor Flip-Flop. On the other side, when a critical path starts from an EDC Flip-Flop, then the performance is affected by the extra delay of this MUX. However, it is not true for every circuit design that critical paths will start from EDC Flip-Flops, while this situation is realization dependent.

During the time interval between the triggering edge of the clock *CLK* and the deactivation of the clock pulse *C_Pulse*, no signal transition is permitted at the input of an EDC Flip-Flop. Although EDC Flip-Flops are placed at the outputs of critical (slow) paths, where no signal transitions are expected within this time interval, possible lateral fast paths ending also to these EDC Flip-Flops may provide such prohibitive signal transitions. To avoid this, a minimum path delay constraint must be considered in the design, only for these fast paths that appear at the side of critical paths. A trade-off arises. A large value for the minimum path delay constraint may increase the silicon area penalty on these paths. On the other side, a small value reduces the error tolerance due to the reduction of the maximum detectable signal delay. The same problems, but at a greater extent, exist for the Razor and the Time Dilation techniques, since all fast paths, anywhere in the circuit, must fulfill a relevant minimum path delay constraint.

IV. Experimental Results

In order to evaluate the proposed timing error detection and correction technique, it has been applied in the pipeline design of a 4th order finite duration impulse response (FIR) digital filter (see Fig. 6), in a 90nm CMOS technology (V_{DD}=1V). The pipeline consists of two stages. The circuit response depends on the weighted sum of its four most recent input samples, as expressed by the following feed forward difference equation:

$$y[n] = \sum_{k=0}^{4} b_k x[n-k]$$

The basic building block of the filter is the multiply and accumulate (MAC) unit. The length of the stage registers ranges from 14-bits to 62-bits (with a total of 116 Flip-Flops in the design). The standard cells based design provided a 220MHz clocking frequency, which is the same as in the original design without the application of the proposed technique. This is due to the fact that no critical paths start from the used EDC Flip-Flops. Only 12% of the registers'

Flip-Flops are related to (fed from) circuit critical paths, that are paths with delay greater than the 75% of the clock period. Consequently, only these Flip-Flops are replaced by the proposed EDC Flip-Flops. The maximum detectable delay is equal to 25% of the clock period, without the need to add extra delays in any lateral fast paths at the side of critical paths in the circuit, as discussed earlier.

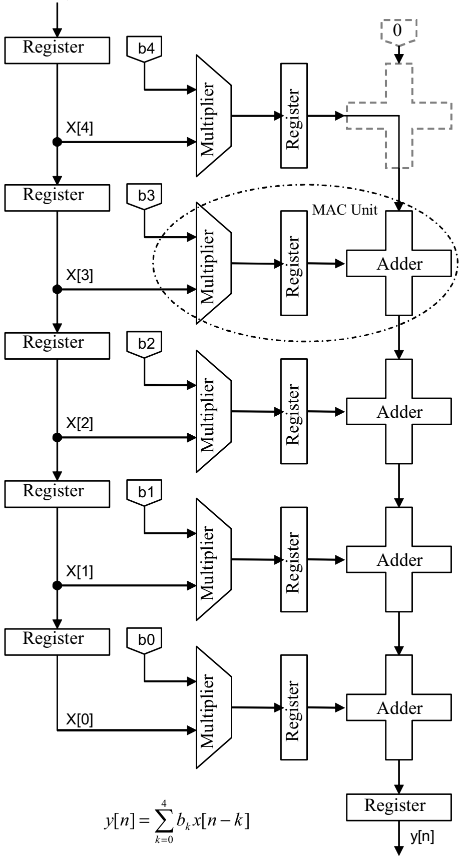

Fig. 6: Fourth-order FIR digital filter topology

Fig. 7 presents simulation results on the final layout. Multiple errors are injected in the circuit by resetting the EDC Flip-Flops just after the triggering edge of the clock signal *CLK* and while their input data are still stable. These errors are detected and the pertinent *Error_F* signals are raised to high, activating the *Block* signal that is used for clock gating during the next clock cycle. In Fig. 7 the *Comp* and *Error_F* signals of a single EDC Flip-Flop are presented for convenience. Then, in the correction cycle, the errors at the EDC Flip-Flop outputs are corrected and the subsequent stages recalculate the correct responses. At the end of the correction cycle the clock signal *CLK* is released and the circuit continues its operation in the normal mode until the next error detection. The total silicon area cost of the filter according to the proposed technique, compared to the Razor and Time Dilation techniques, is reduced by 14.8% and 8.6%, respectively. Moreover, the estimated power consumption is reduced by 39.4% and 17.5% with respect to Razor and Time Dilation.

V. CONCLUSIONS

A timing error tolerance technique is presented in this work. It is based on a new Flip-Flop design that provides the ability to detect and correct multiple timing errors in a circuit at the cost of a single clock cycle for each detected error. The proposed approach is characterized by low cost and reduced design complexity, that also result in reduced power consumption with respect to earlier design schemes in the literature. Although we illustrated for convenience the application of the proposed technique in pipeline architectures, it can be also applied in any sequential circuit design.

REFERENCES

[1] S. Mitra, N. Seifert, M. Zhang, Q. Shi and K. S. Kim, “Robust System Design with Built-In Soft-Error Resilience,” *IEEE Computer,* Volume 38, Number 2, pp. 43–52, 2005.

[2] S. Mitra, M. Zhang, S. Waqas, N. Seifert, B. Gill and K-S. Kim, “Combinational Logic Soft Error Correction,” *IEEE Internatonal Test Conference*, 2006.

[3] M. Agarwal, B.C. Paul, M. Zhang and S. Mitra, “Circuit Failure Prediction and its Application to Transistor Aging,” *IEEE VLSI Test Symposium,* pp. 277-284, 2007.

[4] M. Agarwal, V. Balakrishnan, A. Bhuyan, K. Kim, B.C. Paul, W. Wang, B. Yang, Y. Cao and S. Mitra, Optimized Circuit Failure Prediction for Aging: Practicality and Promise, *IEEE International Test Conference*, 2008.

[5] S.P. Park, K. Roy and K. Kang, “Reliability Implications of Bias-Temperature Instability in Digital ICs,” *IEEE Design and Test of Computers*, vol. 23, no. 6, pp. 8-17, 2009.

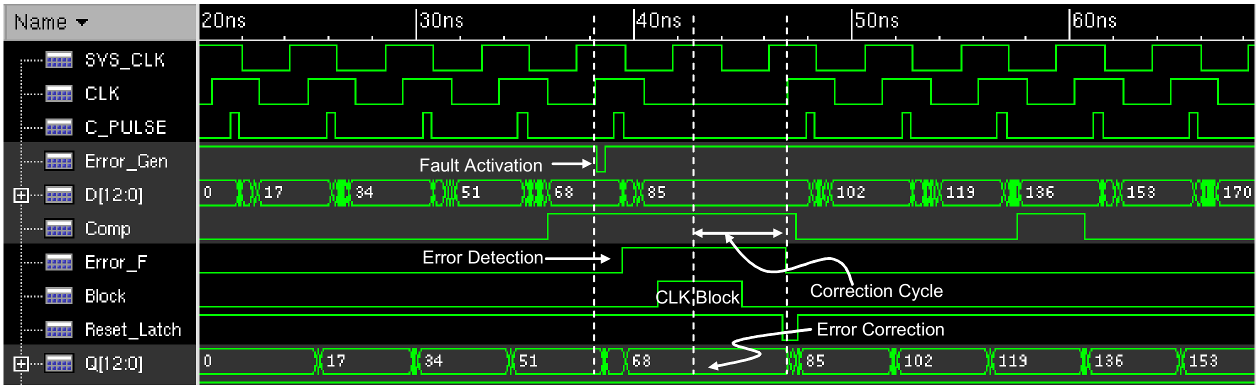

Fig. 7: Simulation results

[6] T. Austin, D. Blaauw, T. Mudge and K. Flautner, “Making Typical Silicon Matter with Razor,” *IEEE Computer,* vol. 37, no. 3, pp. 57–65, 2004.

[7] 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,” International Symposium on Microarchitecture, pp. 7-18, 2003.

[8] M. Nicolaidis and Y. Zorian, “On-Line Testing for VLSI – A Compendium of Approaches,” *Journal of Electronic Testing: Theory and Applications*, vol. 12, no. 1-2, pp. 7-20, 1998.

[9] C. Metra, R. Degiampietro, M. Favalli and B. Ricco, “Concurrent Detection and Diagnosis Scheme for Transient, Delay and Crosstalk Faults,” *IEEE International On-Line Testing Workshop*, pp. 66-70, 1999.

[10] M. Nicolaidis, “Time Redundancy Based Soft-Error Tolerance to Rescue Nanometer Technologies,” *IEEE VLSI Test Symposium,* pp. 86-94, 1999.

[11] Y. Tsiatouhas and Th. Haniotakis, “A Zero Aliasing Built-In Self Test Technique for Delay Fault Testing,” *IEEE Symposium on Design for Testability of VLSI Systems*, pp. 95-100, 1999.

[12] L. Anghel and M. Nicolaidis, “Cost Reduction and Evaluation of Temporary Faults Detecting Technique,” *Design Automation and Test in Europe Conference*, pp. 591-598, 2000.

[13] S. Matakias, Y. Tsiatouhas, A. Arapoyanni, and Th. Haniotakis, “A Circuit for Concurrent Detection of Soft and Timing Errors in Digital CMOS ICs,” *Journal of Electronic Testing: Theory and Applications*, vol. 20, no. 5, pp. 523-531, 2004.

[14] K. Kang, S.P. Park, K. Kim and K. Roy, “On-Chip Variability Sensor Using Phase-Locked Loop for detecting and Correcting Parametric Timing Failures,” *IEEE Tran. on VLSI Systems*, vol. 18, no. 2, pp. 270-280, 2010.

[15] S. Das, C. Tokunaga, S. Pant, W-H. Ma, S. Kalaiselvan, K. Lai, D.M. Bull and D.T. Blaauw, “Razor II: In Situ Error Detection and Correction for PVT and SER Tolerance,” *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 32-48, 2009.

[16] A. Floros, Y. Tsiatouhas, A. Arapoyanni and Th. Haniotakis, “A Pipeline Architecture Incorporating a Low-Cost Error Detection and Correction Mechanism,” *IEEE International Conference on Electronics, Circuits and Systems*, pp. 692-695, 2006.

[17] A. Floros, Y. Tsiatouhas and X. Kavousianos, “The Time Dilation Scan Architecture for Timing Error Detection and Correction,” *IFIP/IEEE International Conference on Very Large Scale Integration*, pp. 569-574, 2008.