

Timing Error Tolerance in Pulsed Latch Based Pipelines

Nikolaos Georgouloupoulos and Yiorgos Tsiatouhas

Dept. of Computer Science and Engineering, University of Ioannina, P.O. Box 1186, 45110 Ioannina, Greece

Abstract—Timing error tolerance turns to be a major design concern in modern nanometer technology integrated circuits. In this work, we present a technique for multiple timing errors detection and correction, which is suitable for pulsed latch based pipelines. The proposed design provides timing error tolerance at a small penalty of extra time for each error correction. Besides that, it is characterized by low silicon area overhead and reduced power consumption, with respect to previous design schemes in the open literature. The proposed technique was applied in six designs that are combinations of benchmark circuits from the ISCAS'85 family, and the simulation results verified its efficiency.

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

I. INTRODUCTION

The increased complexity of modern nanometer technology integrated circuits, demands the development of urgent solutions in order to achieve appropriate reliability levels and keep the cost of testing within acceptable bounds. Reliability levels are mostly affected by the reduced power supply, the continuous transistor scaling, as well as the increased operating frequencies. As a result, transient faults are generated in a more frequent basis, making it difficult to bound error rate levels within specifications [1].

There are various causes for timing error generation, such as power supply disturbances, crosstalk and ground bounce phenomena, increased path delay deviations, manufacturing defects. Moreover, despite the fact that complex testing procedures are followed, the massive increase of the number of the paths in modern integrated circuits (ICs) does not facilitate their timing verification in order to reduce the probability of timing failures. In addition, the timing of modern systems is easily affected due to their operation at multiple frequency and voltage levels, which also results to increased timing error rates. Moreover, transistor aging phenomena must be considered, since they lead to the early occurrence of timing errors in a circuit's lifecycle [2].

Taking into account the above situation, and aiming to achieve acceptable reliability levels in modern ICs, concurrent online testing techniques for timing error detection and correction are becoming obligatory. Furthermore, dynamic voltage scaling (DVS) techniques, for low power operation, can accomplish more effectively timing error tolerance by exploiting error detection and correction mechanisms.

When a timing failure occurs in a combinational logic block, the outcome is a delayed response at its outputs. Thus, after the triggering edge of the clock signal the memory elements at the outputs of this combinational block captures an erroneous value and so a timing error is generated. Numerous error detection techniques have been proposed in the open literature [3], [4], [7], [8]. These techniques can detect the delayed circuit response and provide timing error tolerance by using time redundancy approaches.

In this work, we present a multiple timing error detection and correction scheme, which is oriented to pulsed latch based designs. A new pulsed latch topology is proposed. Additionally, we introduce a pipeline architecture to exploit the new pulsed latch 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. In Section III, the new design technique is introduced and its operation is analyzed. Section IV provides simulation results from the application of the proposed scheme on various benchmark circuits. Finally, Section V concludes this work.

II. EARLIER TIMING ERROR TOLERANCE SOLUTIONS

The Razor pipeline architecture, which supports timing error detection and correction and exploits dynamic voltage scaling for dynamic power reduction, has been introduced in [3]. Regarding this architecture, the stage registers are constructed using the Razor Flip-Flops, as shown in Fig. 1. Apart from the main Flip-Flop, a Razor Flip-Flop consists of an additional assistant shadow latch, a multiplexer (MUX) and a XOR gate. The shadow latch captures, with a proper delay regarding the main Flip-Flop, the responses of the combinational logic. The XOR gate compares the outputs of the main Flip-Flop and the shadow latch after this time duration. When the XOR gate reports a difference (error detection), the error correction mechanism is activated, which redirects the input of the Main Flip-Flop (through the multiplexer) to receive the correct data of the shadow latch and provide them to the logic stage that follows, during the next, recovery, clock cycle.

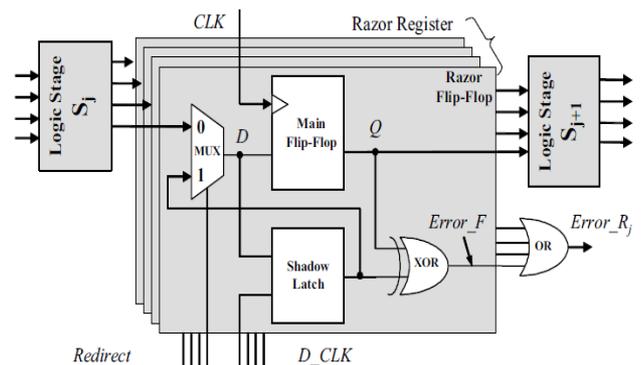


Fig. 1: The Razor Flip-Flop

Recently, another pipeline architecture has been presented in [4] for timing error detection and correction. The stage registers are constructed using the Error-Detection-Correction Flip-Flop (EDC Flip-Flop), as illustrated in Fig. 2. An EDC Flip-Flop consists of the original Flip-Flop (Main Flip-Flop), a Latch, a XOR gate 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 Qbar signal of the Main Flip-Flop is passed at the MUX output. When the XOR gate detects a difference (error detection), the error correction mechanism is activated, and the Qbar signal is passed at the EDC Flip-Flop's output. Thus, the error is corrected. When no error occurs, the Q signal is passed at the output of the EDC Flip-Flop, as it is expected.

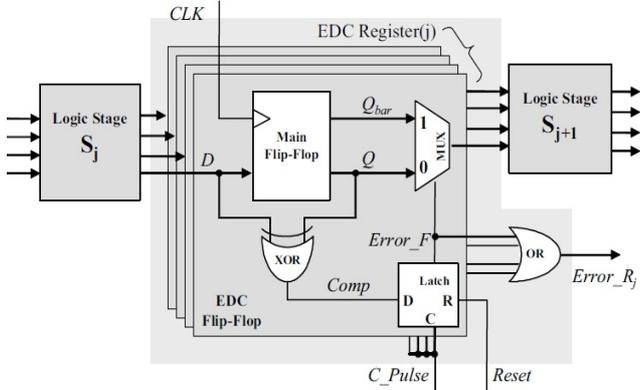


Fig. 2: The EDC Flip-Flop

III. THE PROPOSED TIMING ERROR DETECTION AND CORRECTION MECHANISM

A. The Error Detection and Correction Pulsed Latch

The proposed technique targets pipeline structures that are based on pulsed latches for the stage registers construction. A pulsed latch is a latch that is clocked by a brief pulse (which is generated by a pulse generator). Pulsed-latch circuits may retain the advantages of both latch or flip-flop based designs, by offering higher performance and/or lower power consumption [5]. The proposed timing error detection and correction mechanism is based on a new pulsed latch with error detection and correction capabilities (Error detection and Correction Pulsed Latch - ECPL), which is illustrated in Fig. 3. It consists of an additional pulsed latch (correction latch - PLC) for each protected functional latch (main latch - PLM) and two XOR gates, one for error detection and a second for error correction. A pulsed clock signal CLK drives the PLM, while a delayed pulsed clock signal DCLK drives the PLC. It is important to mention that PLC has the ability of asynchronous reset. Also we have to mention that both CLK and DCLK clock signals are generated by a system clock signal SCLK.

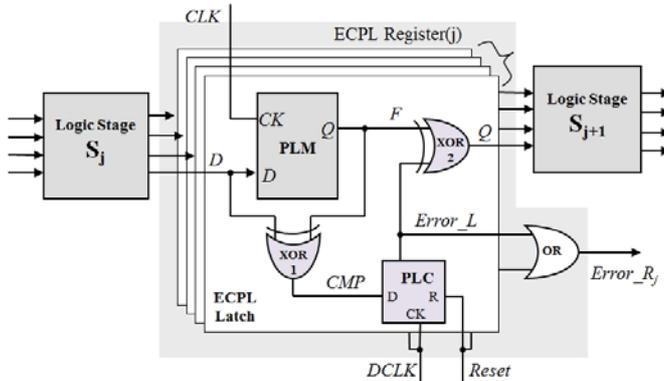


Fig. 3: The proposed ECPL latch

The first XOR gate compares the D input with the F output of the main latch PLM and passes the result to the CMP signal (logic '1' in case of difference). PLC holds the value of this

comparison and drives the local error indication signal Error_L at the input of the second XOR. Depending on the result of this comparison, the F signal passes either inverted (error correction) or as it is (error free case) at the Q output of the second XOR gate.

Initially, PLC is reset to zero. When no timing error occurs, the comparison result (between the PLM's input and output) by the first XOR gate, after the CLK pulse, is '0'. The PLC captures this result, using a delayed clock signal DCLK, and feeds the second XOR. Consequently, the correct value at the output of the PLM still feeds the output of the ECPL latch, which further feeds the next logic stage S_{j+1} . In the presence of a timing fault at logic stage S_j , a delayed signal arrives, after the pulse of CLK, at the input of PLM. In that case, a timing error is generated at PLM. Thus, erroneous data feed the next logic stage S_{j+1} . In addition, the value at the PLM output differs from the correct value at the PLM input. The first XOR detects this difference and sets its output to '1'. PLC captures this high value which is used to drive the second XOR gate. Thus, the erroneous value at the output of PLM is inverted (corrected) by the second XOR gate and is used to feed the next logic stage S_{j+1} . In this way, the timing error is corrected at the ECPL's output.

In general we can distinguish two cases in the proposed timing error tolerance scheme. A timing error is detected at a register of the pipeline where, a) the logic stage that follows is a deep stage or b) the logic stage that follows is a shallow stage. A stage is characterized as shallow when subtracting the error detection and correction time of the ECPL latch from the clock period, the remaining time is enough for its computation in the worst case. Otherwise, the stage is a deep stage. In the first case, the technique provides to the system an extra clock cycle for the recovery of the pipeline from the error. For this purpose, a clock gating scheme is adopted for the suspension of the clock signal (see Fig. 4). Clock gating is activated by the register error indication signals ($Error_{Rj}$), in order to provide the required time to the subsequent logic stage S_{j+1} to complete its computation. On the contrary, in the second case, the technique corrects the error in the same clock cycle at which it is detected, without suspending the clock signal CLK.

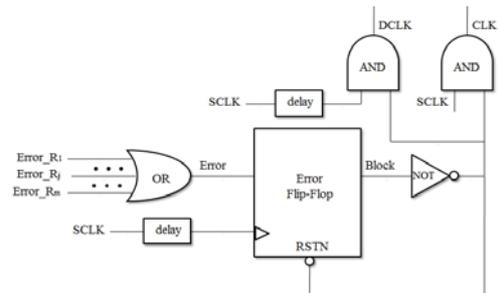


Fig. 4: Clock gating mechanism - Control unit

After the local error detection and correction inside ECPL, the Error_L signal must be '1', until the next CLK pulse so that the next logic stage will have the corrected data at its input, during the current clock period. At the end of this clock period, the Error_L signal must be reset to '0'. This can be achieved as follows. Initially, all the error indication signals Error_L of a register j generate a single register error indication signal $Error_{Rj}$, with the use of an OR gate. In case of a shallow logic stage that follows, the $Error_{Rj}$ feeds a NAND gate along with the clock signal CLK. NAND's output drives the asynchronous reset inputs of all the PLCs of the

register. If an error is detected ($\text{Error_L}=1$), then in the next pulse of CLK, the asynchronous reset input of all PLCs is activated (at low) and resets Error_L outputs to '0'.

In case of a deep logic stage that follows, when a timing fault occurs at the logic stage S_j , false data feed the next deep logic stage (S_{j+1}). The timing error is detected and corrected locally inside the ECPL. Provided that the logic stage S_{j+1} is deep, it is not guaranteed that during the remaining time, after the error correction, the stage will complete its computation successfully. Therefore, additional time is required, in order to ensure the stage's computation with the correct input data. A solution to the problem is the extension of the computation time for an extra clock period, while the rest logic stages will remain stalled, enabling the circuit to recover from the error.

This is achieved by exploiting the clock gating technique. For that purpose, we introduce the control unit that is shown in Fig. 4. The mechanism works as follows: The error indication signals of all the registers (mentioned above) are combined in a global error indication signal Error. The Error signal is captured by a Flip-Flop (Error Flip-Flop - EFF), that is clocked by the system clock signal SCLK with a proper delay. This delay equals the time required for the generation of the Error_L signal and its transmission from the OR tree to the EFF. It is essential to mention that the EFF has synchronous reset capability.

The global error indication signal passes at the EFF's output, generating the clock blocking signal Block. The Block signal prevents the propagation of CLK and DCLK signals to the pipeline, with the use of AND gates. This way, pulse clock signals CLK and DCLK are gated after the detection of an error, for as long as the Block signal is '1'. The gating duration is selected so that the next clock pulse, this after the error detection, will not pass to the pipeline. Thus, the inverted Block signal is used to drive the synchronous reset input of the EFF. Therefore, at the next rising edge of the delayed SCLK, EFF's output will be reset to '0'. As a result, CLK and DCLK are released to feed the circuit.

B. Circuit operation

In Fig. 5 we provide the timing diagrams for the operation of the ECPL latch in Fig. 3. We assume that the logic stage S_{j+1} has long signal paths (it is a deep logic stage). During the clock cycle (i) the response of the logic stage S_j meets the time requirements of the circuit (fault-free case). This means that immediately after the pulse of the clock CLK the input D and the output F of the main pulsed latch hold the same values. Thus, the signal CMP of the first XOR gate is '0' and the same stands for the two signals Error_L and Error_Rj, after the DCLK. Therefore, the second XOR gate passes the default correct signal F to the Q output which feeds the next logic stage S_{j+1} . In that case the operation of the circuit remains unchanged.

During the next cycle (i+1) a timing fault occurs due to a timing failure at the logic stage S_j . The data captured by the main pulsed latch are erroneous and a timing error appears at F. Hence the response of stage S_{j+1} during the next cycle (i+2) will be also incorrect. In addition, as a result of the fault, a transition takes place at the input D of the main pulsed latch, during the (i+2) cycle, just after the pulse clock CLK and before the pulse of clock DCLK. The first XOR gate detects the difference between the values of the signals D and F and sets the output CMP to '1'. Then, after the clock pulse DCLK, the PLC captures the value of CMP and sets the Error_L

signal to '1'. The Error_L signal feeds the input of the second XOR, so that signal F of PLM appears inverted at the output Q. Thus, the next logic stage S_{j+1} is fed with correct data. This way, the error is corrected at this particular memory element. The same situation occurs at any other memory element where an error is detected. The rest of the memory elements that have correct data remain unchanged. It should be noted that the error correction is accomplished without the need to recalculate the response of the failing stage S_j .

Furthermore, the Error_L signal initiates the register error indication signal Error_Rj, via an OR gate, which collects all the error indication signals of the ECPLs. Then all register error indication signals feed the control unit of the Fig. 4, as discussed earlier, to activate Block signal. Consequently, the signals CLK and DCLK are blocked for one cycle (i+3), in order to provide the required time by the logic stage S_{j+1} for its calculation. Thus, an additional clock cycle is provided for the error correction and the recovery of the pipeline. Finally, at the end of the correction cycle (i+3), the Block signal is deactivated and the PLCs, where an error was detected, are reset to '0', so that at the next clock cycle (i+4) the pipeline returns to normal operation.

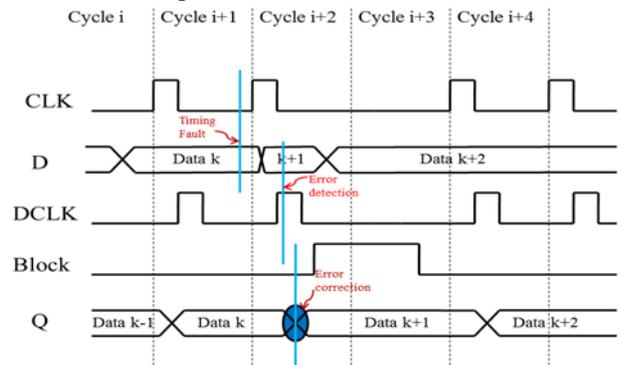


Fig. 5: Timing diagrams

C. Pipeline recovery

An error detection by the proposed mechanism is followed by a pipeline recovery. According to Fig. 6(a), the control unit of the Fig. 4 is used in order to exploit the clock gating technique for the pipeline recovery. In this figure, each logic stage is supported by error correction (EC) registers, where the proposed timing error detection and correction mechanism is embedded. In case of a timing error occurrence, the system clock signal is blocked for the next clock cycle, with the assistance of the Block signal of the EFF. During this time interval (recovery cycle), the stages that initially are fed by erroneous data, like stage LS3 in Fig. 6(b) due to a timing fault in the previous stage LS2, they recalculate their responses with corrected data at their inputs.

The rest of the logic stages remain inactive, while holding the correct responses at their outputs. It is worth to mention that it is not required by the failing stage LS2 to recalculate its response, since the correct response is automatically recovered from the ECPL.

The proposed pipeline timing error detection and correction architecture can tolerate any number of timing errors in any logic stage within a clock cycle, due to the fact that all stages are capable to recalculate their responses with correct data at their inputs, during the additional clock cycle. In the worst case scenario, where one or more stages fail in each clock cycle, the pipeline will carry on its operation at half of the normal speed.

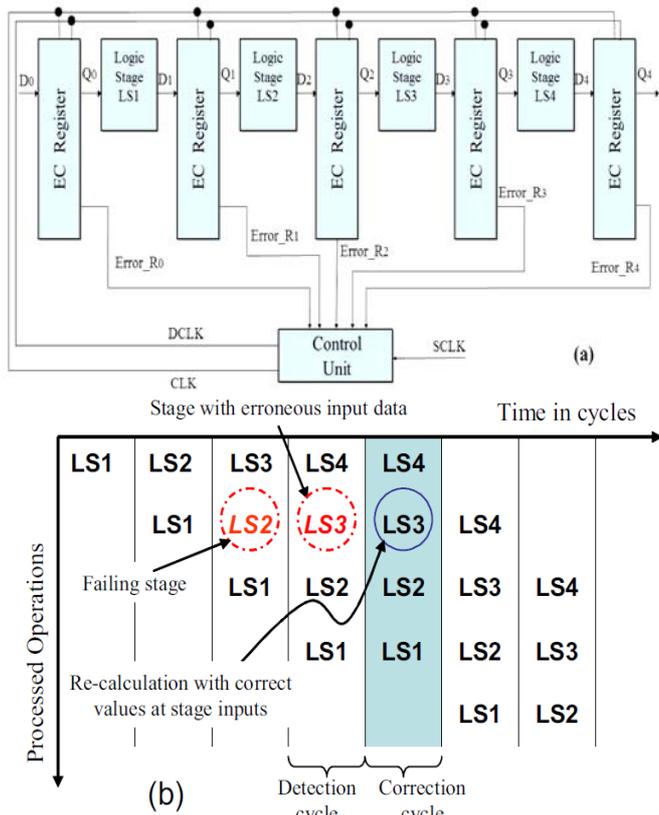


Fig. 6: Pipeline recovery, (a) architecture and (b) operation

IV. EXPERIMENTAL RESULTS

The proposed and the well-known Razor techniques were separately applied in six benchmark circuits that are constructed by combining the c17, c432 and c499 benchmark circuits of the ISCAS'85 family [6]. Altera Quartus II was used to design and simulate the specific circuits and also to extract estimations on silicon area (in logic elements (LE) that are used), power consumption and operating frequency. The simulation results verified the ability of the proposed technique to detect and correct multiple timing errors. In Table I comparison results are presented between the two techniques, where reduced power consumption and lower silicon area requirements are reported for the proposed technique with respect to the Razor topology, at the same operating frequencies.

In Fig. 7, simulation waveforms from the proposed circuit operation are illustrated. Multiple errors are injected as follows; a XOR gate is inserted before the input of the PLM block in each ECPL latch. The second input of the XOR gate is control by an external to the circuit Error_Ctrl signal, which is exploited for error generation. In order to generate an error, a transition is performed on the Error_Ctrl inside the monitoring window after the pulse of CLK and before the pulse of DCLK. Thus, the input D of PLM is inverted (this represents a delayed response) and a difference is generated between the input D and the output F of PLM. Consequently, the error tolerance mechanism is activated. The pertinent errors are detected and the corresponding error indication signals are set to '1'. The combined error indication signals produce the global error indication signal, which is captured by the EFF of the control unit, which is clocked by the ECLK signal (this is the system clock signal SCLK with a proper

delay). As a result, the Block signal is activated for clock gating during the next clock cycle. In the correction cycle that follows, the errors at ECPL outputs are corrected and the subsequent stages recalculate the correct responses. In the last part of the correction cycle, the clock pulses CLK and DCLK are released, permitting the circuit to continue its normal operation, until the next error detection.

Table I: Comparison results

ICs	Area (LE)		Power (mW)		Frequency (MHz)	
	Razor	Proposed	Razor	Proposed	Razor	Proposed
#1	282	270	22.69	19.21	50	50
#2	856	204	52.87	14.85	50	50
#3	24	24	4.19	4.51	50	50
#4	195	56	17.30	5.75	50	50
#5	331	162	11.41	6.16	25	25
#6	473	271	16.10	10.31	25	25

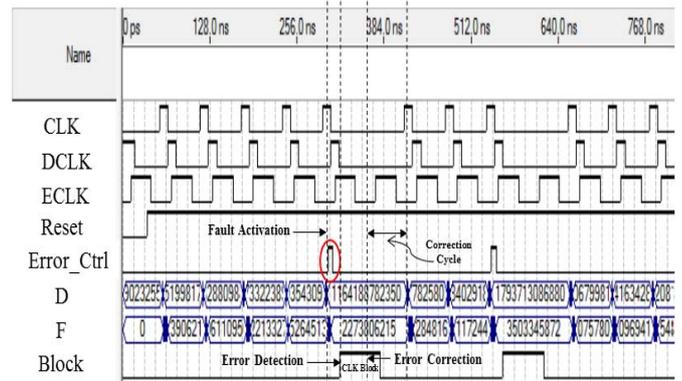


Fig. 7: Indicative simulation waveforms

V. CONCLUSIONS

A timing error tolerance technique for pulsed-latch based pipelines is presented in this work. It exploits a new pulsed-latch design, which provides the ability of multiple timing error detection and correction. The proposed scheme is a low silicon area solution that is characterized by reduced power consumption, with respect to the Razor design approach.

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*, vol. 38, no 2, pp. 43–52, 2005.
- [2] 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.
- [3] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, T. Mudge, "Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation," *36th Annual International Symposium on Microarchitecture*, pp. 7–18, 2003.
- [4] S. Valadimas, Y. Tsiatouhas and A. Arapoyanni, "Timing Error Tolerance in Nanometer ICs," *IEEE 16th International On-Line Testing Symposium*, pp. 283–288, 2010.
- [5] Y. Shin and S. Paik, "Pulsed-Latch Circuits: a New Dimension in ASIC Design," *IEEE Design & Test of Computers*, vol. 28, no. 6, pp. 50–57, 2011.
- [6] F. Brglez, D. Bryan and K. Kozminski, "Combinational Profiles of Sequential Circuits," *IEEE International Symposium on Circuits and Systems*, pp. 1929–1934, 1989.
- [7] M. Nicolaidis, "GRAAL: A New Fault Tolerant Design Paradigm for Mitigating the Flaws of Deep Nanometric Technologies", *IEEE International Test Conference*, p. 4.3, 2007.
- [8] R. Samanta, G. Venkataraman, N. Shah and J. Hu, "Elastic Timing Scheme for Energy-Efficient and Robust Performance", *9th International Symposium on Quality Electronic Design*, pp. 537–542, 2008.