

Timing Error Detection and Correction for Reliable Integrated Circuits in Nanometer Technologies

Stefanos Valadimas *

Department of Informatics and Telecommunications
National and Kapodistrian University of Athens
s.valadimas@di.uoa.gr

Abstract. Timing error tolerance turns to be an important design parameter in nanometer technology, high speed and high complexity integrated circuits. This thesis presents three concurrent on-line timing error tolerance techniques which enhance circuit's reliability. The proposed techniques detect and correct timing errors efficiently, in flip-flop based designs, with low power consumption and low silicon area overhead. To validate the three novel techniques, they have been applied in the design of a 32-bit MIPS R2000 pipeline microprocessor.

Keywords: concurrent on-line testing, timing errors, error detection and correction, timing error tolerance, reliability-aware design.

1 Introduction

As technology scales down timing errors are a real concern in high complexity and high frequency integrated circuits. Process, Voltage and Temperature (PVT) variations [1] lead to large spreads in delay, at the system level, which undermine circuit's reliability. Moreover, crosstalk [2], power supply disturbances and resistive IR-drop [3] affect circuit performance increasing the overall impact of timing errors.

In addition, aging mechanisms [4] cause gradual speed degradation of the designs over their service life, mainly due to Bias Temperature Instability (BTI) [5], which is one of the most important phenomena that degrade the performance of nano-scale circuits. BTI primarily accelerates the aging process of MOS transistors by increasing their threshold voltage. BTI-induced delay shifts in logic paths, are related to timing violations during the circuit lifetime.

The increased path delay deviations, due to the above factors, result in timing errors that are not easily detectable in terms of test cost. To mitigate the impact of nanometer scaling, conservative approaches, with wider safety margins, are adopted to guarantee the reliability during system lifetime. In this context, it is evident that timing error tolerance techniques are becoming necessary to provide robustness against timing violations and meet system reliability requirements.

* Dissertation Advisor: Angela Arapoyanni, Professor.

2 Previous Solutions In Error Tolerance

A number of error tolerance techniques have been proposed for flip-flop and latch based designs. Aiming the detection of errors the techniques proposed in [6] and [7] sense the delayed circuit response and provide error tolerance using time redundancy approaches. A well-known and commonly used scheme for flip-flop based designs is the Razor pipeline architecture [8]. The Razor flip-flop consists of the main system flip-flop plus an assistant shadow latch, a multiplexer and a XOR gate (Fig. 1). The shadow latch is clocked by a delayed version of the system clock in order to capture delayed responses of the combinational logic. The XOR gate compares the outputs of the main flip-flop and the shadow latch for error detection. Whenever a timing error occurs the correct data, which are stored in the shadow latch, are injected into the pipeline during the next clock cycle. For every main flip-flop an extra latch, a multiplexer and a XOR gate are required. Hence, this approach suffers from high power consumption and high silicon area cost. Moreover, a metastability detector is required to guarantee high levels of reliability.

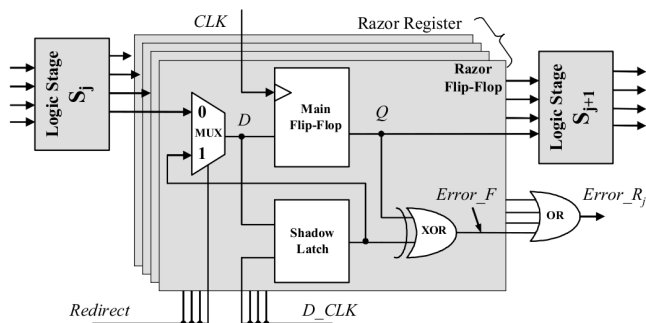


Fig. 1: The Razor flip-flop.

For latch based designs a modified version of the Razor topology (Razor II) is presented in [9]. Its application to a 32-bit ARM microprocessor is discussed in [10]. Also in that case a transition detector is used, at the output of the latch, for error detection while error correction is performed through architectural replay. Another solution to enhance tolerance for latch based designs is the GRAAL architecture [11]. It is based on the XOR comparator for error detection and an additional flip-flop per latch for error correction.

An alternative approach, which masks timing errors by borrowing time from successive pipeline stages, is presented in [12]. An additional latch per main system flip-flop is used to re-sample the input data with a proper delay. Various double-sampling architectures are discussed in [13].

3 Time Dilation Technique

In this section, the first proposed error detection and correction technique is presented. The Time Dilation [14] technique exploits a new scan flip-flop which supports both the standard off-line scan testing capability as well as the on-line (concurrent) error detection and correction capability. According to the proposed technique, after error detection the evaluation time for the logic is automatically extended by a single clock cycle for error correction using correct and valid data stored in each flip-flop. Unlike earlier solutions, no extra memory elements are required in the Time Dilation approach.

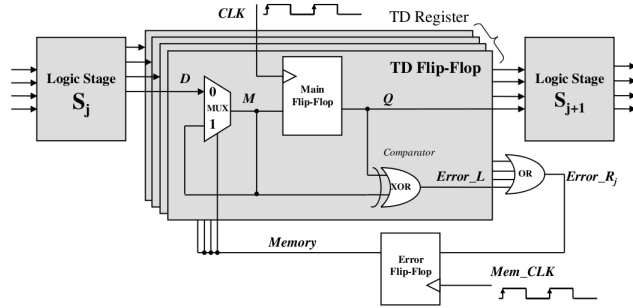


Fig. 2: The Time Dilation flip-flop.

The Time Dilation flip-flop (TD flip-flop) [14] is presented in Fig. 2. This topology utilizes a multiplexer (MUX) and a XOR gate per system flip-flop (Main flip-flop) to provide timing error detection and correction capabilities. The XOR gate compares the input and the output of the Main flip-flop for error detection, while the multiplexer with the feedback configuration forms an extra memory element (a MUX-latch) that captures delayed valid data for error correction. After error detection the logic evaluation time is extended by a clock cycle for error correction, by re-feeding the Main flip-flop with the correct and valid data of the MUX-latch.

The operation of the new flip-flop is quite simple. Initially, the Error flip-flop is reset to low, so that the TD flip-flop is in the normal mode of operation and the D input feeds the Main flip-flop. In the fault free case, the data arrive in time at the D input of the TD flip-flop, they propagate to the M input of the Main flip-flop and they are captured at the Q output by the triggering edge of the clock signal CLK . After the triggering edge the inputs of the XOR gate (signals M and Q) hold the same logic value and the output signal $Error_L$ of the XOR gate is low (no error detection). Consequently, the Error flip-flop retains the low state at its $Memory$ output, after the triggering edge of the MEM_CLK clock signal, and the TD flip-flop remains in the normal mode of operation.

The MEM_CLK clock signal is a delayed version of the CLK clock signal. However, in the presence of a timing failure, which results in a delayed arrival of the data at the signal lines D and M , the logic values on M and Q differ after the triggering edge of the clock signal CLK . Thus, the signal Error_L is high indicating an error detection. Consequently, the register error indication signal Error_R will be also high and the same stands for the *Memory* signal after the triggering edge of the MEM_CLK clock signal. As a result, the MUX-latch enters the memory state of operation capturing the delayed but correct data at the M input of the Main flip-flop. These correct data feed the Main flip-flop at the next triggering edge of CLK for error correction and circuit operation recovery.

The hardware overhead and the power consumption of the TD flip-flop is much lower than this of the Razor topology, since in the latter topology except of the multiplexer and the XOR gate an additional shadow latch is required.

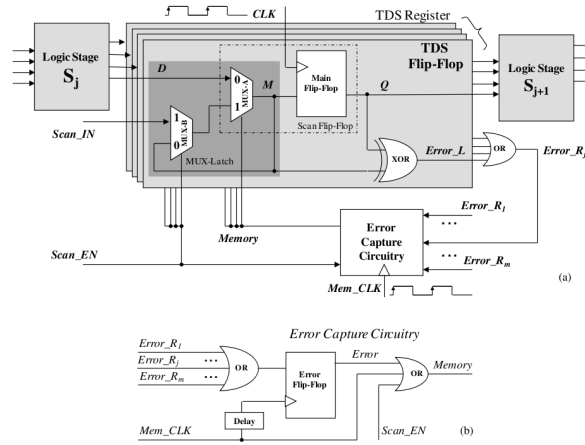


Fig. 3: (a) TDS flip-flop and support circuitry and (b) error capture circuitry.

The scan version of the Time Dilation architecture is presented in Fig. 3. The Time Dilation Scan flip-flop (TDS flip-flop) provides error detection and correction capabilities by appending only a multiplexer (MUX-B) which is utilized for the scan operation, as in a standard scan design. When the scan enable signal ($Scan_EN$) is “high” the TDS flip-flop operates like a Scan flip-flop to support off-line scan testing procedures. At the same time the *Memory* signal must be also “high”. Consequently, the test data are propagated from the $Scan_IN$ port to the input line M of the main flip-flop where they are captured. Then, they are provided through the Q line to the $Scan_IN$ port of the next flip-flop in the chain and so on. In the normal mode of operation (where $Scan_EN$ is “low”) the TDS flip-flop behaves like an ordinary flip-flop enhanced with the ability to detect and correct timing errors as it has been analyzed above.

As in the Razor technique, a crucial issue in the proposed technique is the possible existence of short (fast) paths in the combinational logic which may corrupt the data in the MUX-latches. This is the well known hold time problem. As fast paths we define paths with response times inside the monitoring window. To avoid the hold time problem, a minimum path delay constraint is proposed in Razor. This constraint is fulfilled adding delay buffers during logic synthesis to slow down short paths (paths padding).

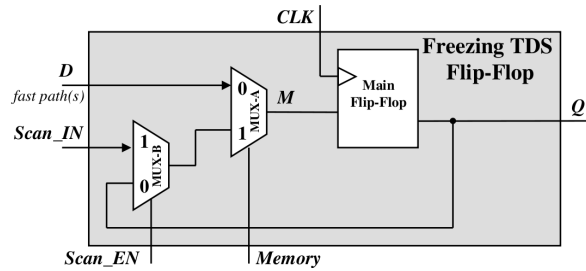


Fig. 4: Freezing TDS flip-flop.

Path padding techniques can be also applied in the proposed TDS technique. However, aiming to reduce the pertinent cost, an alternative design approach can be used instead. The Freezing TDS flip-flop in Fig. 4 is exploited at the end of fast paths that do not intersect with critical paths. Delay buffers are inserted only in the rest fast paths that intersect with time critical paths in order to avoid data corruption in the MUX-latches of the standard TDS flip-flops. In those cases, the minimum path delay constraint is equal to the delay of the *Memory* signal with respect to the system clock *CLK*, plus the hold time of the MUX-latch.

The operation of the Freezing TDS flip-flop in Fig. 4 is based on the fact that the data captured by a flip-flop at the end of a fast path are always correct since they are not affected by timing failures. Consequently, the comparator (XOR gate) is eliminated. The main difference in this new topology is that the *Q* output of the Main flip-flop drives MUX-B instead of the *M* line. Thus, in the memory phase of MUX-latch (*Memory*="high") the output data of the Main flip-flop re-feed its input *M* and latched by the MUX-latch. After a timing error detection at a TDS flip-flop anywhere in the circuit, the correct data of the MUX-latch in a Freezing TDS flip-flop are re-captured at the output *Q* of the Main flip-flop (data freezing) by the triggering edge of *CLK* in the next correction cycle.

In order to evaluate the proposed timing error detection and correction technique, it has been applied in the design of a 32-bit pipelined MIPS R2000 microprocessor, with scan testing support, in the 90nm CMOS technology of UMC using the standard cells of Faraday Technologies. In parallel, the same microprocessor was designed, in the same technology, using the corresponding

flip-flop oriented Razor technique, with scan support. Comparisons between the two MIPS core designs proved that Time Dilation outperforms over Razor with respect to power consumption and silicon area cost. The Time Dilation based design presents a 12.6% reduction in the power consumption and 1.6% reduction in the silicon area with respect to the Razor based design.

4 Error Detection and Correction Technique

The second proposed technique, the Error Detection and Correction (EDC) technique [15], is based on the bit-flipping flip-flop concept. This is synopsised as follows: in case of error detection at the output of a flip-flop the corresponding logic value is asynchronously complemented for error correction.

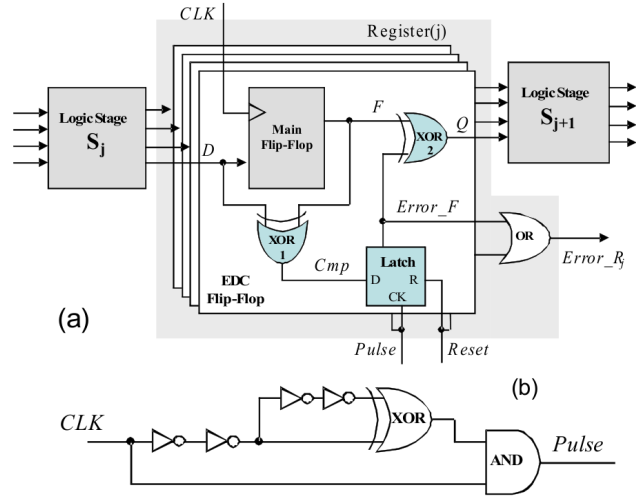


Fig. 5: (a) The EDC flip-flop and (b) The pulse generator.

Fig. 5(a) illustrates the new Error Detection/Correction flip-flop (EDC flip-flop) that is suitable to confront with timing errors. Apart from the original flip-flop (Main flip-flop), it consists of two XOR gates and a latch. The first XOR gate compares the D input and the F output of the Main flip-flop and provides the result to the latch. The latch feeds the second XOR gate at the output of the Main flip-flop. Depending on the comparison result within a specified time interval, either the F signal of the Main flip-flop or its complement is propagated to the output Q of the EDC flip-flop. 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 output of each flip-flop, at the register between the two stages, where a timing error has been detected is complemented so that valid values feed the S_{j+1} logic stage. Moreover, in case that this stage is not fast enough (not a shallow stage), the evaluation time of the circuit is extended by one clock cycle to guarantee its correct computation.

Initially, the output *Error_F* of the latch is reset to zero so that by default the *F* signal of the Main flip-flop propagates to the output *Q* of the XOR gate and feeds the subsequent logic stage. In the error free case the comparison result is a low value at the *Cmp* output of the first XOR gate after the triggering edge of the clock signal *CLK*. This value is captured by the latch. Thus, the *Q* output signal is identical to the *F* signal of the Main flip-flop, which carries the correct value. This signal feeds 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 present at the *F* output of the Main flip-flop and erroneous data are provided to the subsequent logic stage S_{j+1} through the *Q* output. In addition, the *F* signal value differs from the *D* signal value. The first XOR gate detects this difference and raises its output *Cmp* to high. The latch captures and holds this response. Thus, the second XOR gate provides at its output *Q* the complement of the *F* signal. Now the *Q* output of the EDC flip-flop carries the correct value, which feeds the subsequent logic stage S_{j+1} for its computation. Consequently, the error is locally corrected.

A clock pulse (*Pulse* signal) is used to capture the comparison result of the first XOR gate in the latch (memory state when the *Pulse* is low). This clock pulse can be generated locally from the *CLK* signal using a single Pulse Generator per register like the one illustrated in Fig. 5 (b). Thus, the routing overhead of an extra clock signal is relaxed. The AND gate in Fig. 5 (b) ensures that a single pulse will be generated only during the first phase of every clock cycle. The pulse width is at least equal to the time required by the latch to capture the comparison result. The time interval between the triggering edge of *CLK* and the falling edge of *Pulse* (minus the latch set up time and the XOR propagation delay time) determines the maximum detectable signal delay. Every signal transition at the *D* input of an EDC flip-flop within this time interval is considered as a delayed response. So the circuit design must guarantee that in the fault free case there are no signal transitions at the inputs of EDC flip-flops within this time interval, in order to avoid false alarms.

However, in the general case and in order to ensure the correct operation, extra time is required by the S_{j+1} logic stage to perform its computation after the correction of its input values. For that reason the error indication signal *Error_F* is used to block the clock signal from feeding the flip-flops during the subsequent clock cycle of the cycle where the error has been detected. Thus, a single clock cycle is dedicated for state recovery.

A core level clock gating technique can be exploited. Note that core level clock gating techniques are in common use for low power operation. To achieve this, the *Error_F* signals of all EDC flip-flops in a register (*j*) generate the register's

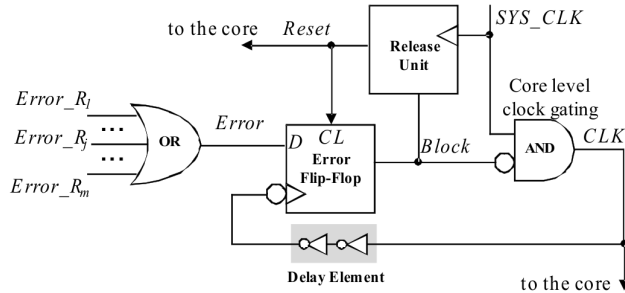


Fig. 6: Clock gating signal generation.

error indication signal $Error_R_j$ through a local OR gate (see Fig. 5(a)). Next, all registers' $Error_R_j$ signals are collected by a second OR gate which generates the core level error indication signal $Error$, as it is shown in Fig. 6. The $Error$ signal is captured by a single flip-flop, the $Error$ flip-flop. Its output signal $Block$ is used for core level clock gating and to activate the $Release$ unit. The latter releases the clock signal, after the expiration of the next system clock cycle, by the activation of the $Reset$ signal which clears the $Error$ flip-flop.

Moreover, the $Reset$ signal clears the latches in the EDC flip-flops. Actually, the $Release$ unit is a counter that counts one system clock cycle after its activation. 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. Considering small processing cores, the propagation of the $Error_F$ signal will be fast enough to properly block the clock signal.

Comparisons on the MIPS pipelined microprocessor design proved that the EDC technique outperforms over Razor and Time Dilation with respect to power consumption and silicon area cost. The EDC supported design presents 20.8% and 9.2% reduction in the estimated power consumption with respect to the Razor and the Time Dilation supported designs respectively. Considering the silicon area, the EDC supported design presents 11.5% and 10.3% less silicon area with respect to Razor and Time Dilation supported designs respectively.

5 Timing Error Tolerance Technique

The Timing Error Tolerance (TET) technique [16], the third proposed error detection and correction technique, exploits the fact that after the triggering edge of the clock signal in a flip-flop, the data at its output must retain their value until the next triggering edge of the clock. Thus, any signal transition detected at the input of the flip-flop, during this time interval, is related to a timing error that can be corrected by bit-flipping the data stored in the flip-flop. Moreover, according to the adopted scheme, only the flip-flops at the end of

critical paths are replaced by the proposed flip-flop. The timing error tolerant oriented flip-flop structure is presented in Fig. 7(a). It consists of a Transition Detection (TD) unit for error detection and a flip-flop with preset and clear options, which is exploited for error correction.

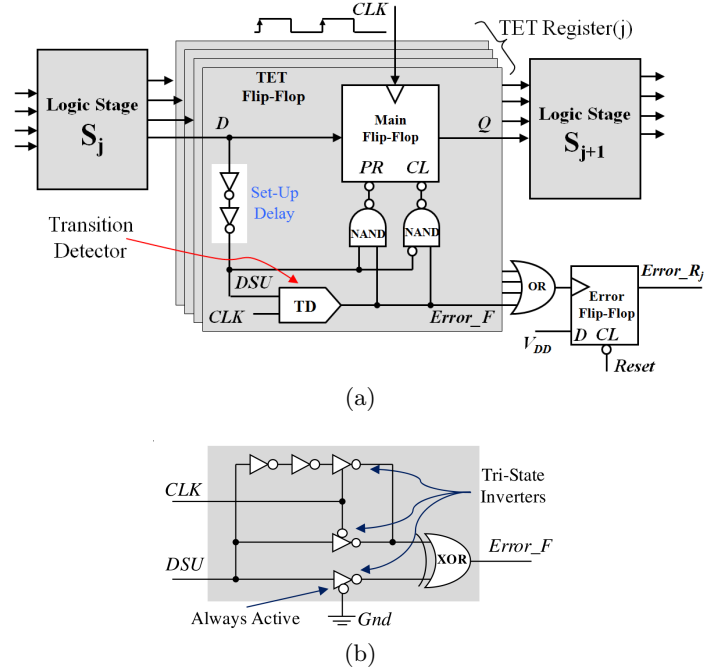


Fig. 7: (a) The proposed Timing Error Tolerant flip-flop and (b) the Transition Detector scheme.

The TD unit monitors the input D of the flip-flop within a time period (monitoring window) after the triggering edge of the clock CLK . During this time interval, no signal transitions are expected at the input of the flip-flop. In case of a signal transition within the monitoring window, the TD unit indicates an error detection by raising its output $Error_F$ to high. A signal transition within the flip-flop's setup time is also considered as a timing violation. In order to be detected as timing error, it must arrive after the triggering edge of the clock. Thus, the TD unit is driven by a delayed version of the flip-flop input signal. This delay is equal to the setup time of the flip-flop. With the signal $Error_F$ at logic "high", the correction operation follows. Two NAND gates are used, which are driven by $Error_F$ and the delayed input signal DSU . If the final input data are at logic "high" then the $Error_F$ signal activates the first NAND gate which presets the flip-flop output to high. If the final input data are at logic "low" then the $Error_F$ signal activates the second NAND gate which clears the

flip-flop. In both cases the output Q of the Main flip-flop turns to the value of the correct but delayed data.

The TD unit design is illustrated in Fig. 7(b). It consists of a two input XOR gate, three tri-state inverters and delay elements. One input of the XOR gate is always driven by the \overline{DSU} signal, because the bottom tri-state inverter is always active. The other input of the XOR gate is driven either by the \overline{DSU} signal or by a delayed version of that signal, depending on the value of CLK . When the CLK signal is at logic “low” the two bottom signal paths are activated. Thus, any transition at the input of the TD unit arrives concurrently at both of the XOR gate inputs and no pulse is generated at its output. When the CLK signal is at logic “high” the top and the bottom signal paths are active. In this case, due to the delay elements inserted in the top path, there is a delay between the arrivals of the signals at the two inputs of the XOR gate. Thus, a pulse is generated at the XOR’s output. The pulse width is equal to the delay inserted in the top path and adequate to activate the preset or clear operation at the Main flip-flop.

From the above analysis, it is clear that the monitoring window of the TD unit is determined by the “high” pulse width of the CLK signal. Any transition at input D , in this time interval, is detected as timing error, so that in the normal operation of the circuit, no transitions are permitted at this input. To avoid false alarms, either the duty cycle of CLK signal is adjusted or the fast paths are delayed, according to a minimum path delay constraint, or both techniques are applied.

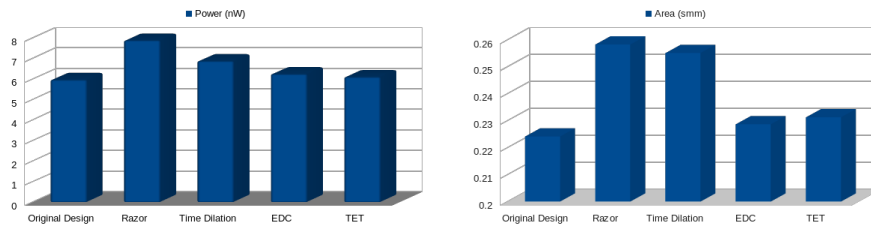


Fig. 8: Comparison graphs for power consumption and silicon area

In the comparisons that follow the standard cells of the 90nm Faraday library are used for the design of all four techniques at the same operating frequency. The TET based design presents 25.59%, 11.21% and 2.24% reduction in power consumption with respect to the Razor [8], the Time Dilation [14] and the EDC [15] based designs respectively. Considering the silicon area, the TET design presents 10.46% and 9.33% less silicon area with respect to Razor and Time Dilation designs respectively and 1.1% increase with respect to the EDC technique. Comparison graphs are presented in Fig. 8.

This scheme was also applied on a 32-bit pipelined MIPS microprocessor, which was fabricated in the 65nm Low Leakage technology of UMC, through the

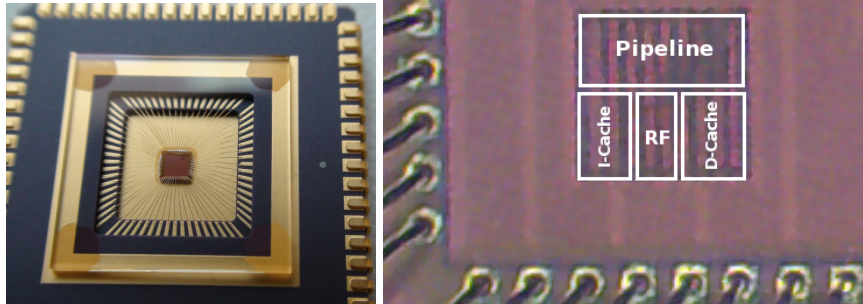


Fig. 9: Fabricated chip and die photo

EUROPRACTICE IC Service, offered by IMEC and Fraunhofer. Fig. 9 shows the fabricated chip and the die photo with the MIPS core.

For the evaluation of the proposed technique on the fabricated chip, timing errors are created by operating the microprocessor at a lower voltage level than the nominal. The design has two outputs: the global error indication signal and the signature output of a Multiple-Input Signature Register (MISR), which is used to compress the response of the design. The error indication signal shows whether timing errors are detected, while the value of the signature shows whether these errors are corrected or not, by comparing this value (i.e. the compacted response of the design) with the expected one. Experimental results show that the proposed technique detects and corrects the generated timing errors efficiently with low power consumption and low silicon area overhead.

6 Conclusions

Timing errors in the memory elements of a design are of increasing importance in nanometer technology microprocessor cores. This thesis presents three low cost timing error detection and correction technique. The first technique provides concurrent error detection and correction in the field of application and also supports off-line manufacturing scan testing. By utilizing a new scan flip-flop, this technique is capable to detect and correct multiple errors at the minimum penalty of one clock cycle delay. The second technique is based on a new bit flipping flip-flop. Whenever a timing error is detected, it is corrected by complementing the output of the corresponding flip-flop. The last technique exploits a transition detector for timing error detection along with asynchronous local error correction schemes to provide timing error tolerance. The proposed approaches are characterized by low cost and reduced design complexity, that also result in reduced power consumption area with respect to earlier design schemes in the open literature.

References

1. J. Semião, J.F. Freijedo, J.J. Rodriguez-Andina, F. Vargas, M.B. Santos, I.C. Teixeira and P.J. Teixeira, "Time Management for Low-Power Design of Digital Systems", ASP Journal of Low Power Electronics (JOLPE), vol. 4, no. 3, pp. 410-419, 2008.
2. M. Cuvillo, S. Dey, X. Bai, Y. Zhao., "Fault Modeling and Simulation for Crosstalk in System-on-Chip Interconnects," Int. Conf. on Computer Aided Design, pp. 297-303, 1999.
3. H. Chen, L. Wang., "Design for Signal Integrity: The New Paradigm for Deep-Submicron VLSI Design," Proc. Int. Symp. on VLSI Technology, pp. 329-333, 1997.
4. S.V. Kumar, C.H. Kim, S. Sapatnekar, "Adaptive Techniques for Overcoming Performance Degradation due to Aging in Digital Circuits," Proc. IEEE ASP-DAC, pp. 284-289, 2009.
5. S. Khan, S. Hamdioui, H. Kukner, P. Raghavan and F. Catthoor, "BTI impact on logical gates in nano-scale CMOS technology," Proc. IEEE Int. DDECS, pp. 348-353, 2012.
6. 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.
7. 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 Transactions on VLSI Systems, vol. 18, no. 2, pp. 270-280, 2010.
8. 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.
9. S. Das, C. Tokunaga, S. Pant, W-H. Ma, S. Kalaiselvan, K. Lai, D.M. Bull and D.T. Blaauw, "RazorII: 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.
10. D. Bull, S. Das, K. Shivashankar, G.S. Dasika, K. Flautner and D.T. Blaauw, "A Power-Efficient 32 bit ARM Processor Using Timing-Error Detection and Correction for Transient-Error Tolerance and Adaptation to PVT Variation," IEEE Journal of Solid-State Circuits, vol. 46, no. 1, pp. 18-31, 2011.
11. M. Nicolaidis, "GRAAL: a new fault tolerant design paradigm for mitigating the flaws of deep nanometric technologies," IEEE International Test Conference, 2007.
12. M. Choudhury, V. Chandra, R. Aitken, and K. Mohanram, "Time-borrowing circuit designs and hardware prototyping for timing error resilience," IEEE Transactions on Computers, vol. 63, no. 2, pp. 497-509, 2014.
13. M. Nicolaidis, "Double-Sampling Design Paradigm—A Compendium of Architectures," IEEE Transactions on Device and Materials Reliability, vol. 15, no. 1, pp. 10-23, 2015.
14. S. Valadimas, A. Floros, Y. Tsiatouhas, A. Arapoyanni, X. Kavousianos, "The Time Dilation Technique for Timing Error Tolerance," IEEE Transactions on Computers, vol. 63, no. 5, pp. 1277-1286, 2014.
15. S. Valadimas, Y. Tsiatouhas, A. Arapoyanni, "Timing Error Tolerance in Small Core Designs for SoC Applications," IEEE Transactions on Computers, vol. 65, no. 2, pp. 654-663, 2016.
16. S. Valadimas, Y. Tsiatouhas, A. Arapoyanni, P. Xarchakos, "Effective Timing Error Tolerance in flip-flop Based Core Designs," Springer Journal of Electronic Testing: Theory and Applications, vol. 29, no. 6, pp. 795-804, 2013.