Exploiting Parametric Power Supply and/or Temperature Variations to Improve Fault Tolerance in Digital Circuits

U. Algarve, Portugal  U. Vigo, Spain  PUCRS, Brazil  IST / INESC-ID, Portugal
jsemiao@ualg.pt  jjrdguez@uvigo.es  vargas@computer.org  paulo.teixeira@ist.utl.pt

Abstract

The implementation of complex functionality in low-power nano-CMOS technologies leads to enhance susceptibility to parametric disturbances (environmental, and operation-dependent). The purpose of this paper is to present recent improvements on a methodology to exploit powersupply voltage and temperature variations in order to produce fault-tolerant structural solutions. First, the proposed methodology is reviewed, highlighting its characteristics and limitations. The underlying principle is to introduce on-line additional tolerance, by dynamically controlling the time of the clock edge trigger driving specific memory cells. Second, it is shown that the proposed methodology is still useful in the presence of process variations. Third, discussion and preliminary results on the automatic selection (at gate level) of critical FF for which DDB insertion should take place are presented. Finally, it is shown that parametric delay tolerance insertion does not necessarily reduce delay fault detection, as multi-vdd or multi-frequency self-test can be used to recover detection capability.

1. Introduction

Signal integrity is a key issue for high-performance digital System-on-Chip (SoC) products [1] and may have many causes. However, existing techniques do not prevent operation dependent errors from happening without loosing performance. VDD and/or temperature (T) variations can make signals to be captured and transmitted in the incorrect time window. Such operation-dependent timing faults are just enough to make the system fail its performance specs.

From the yield (and profitability) point of view, making the product more tolerant to $V_{DD}$ and/or T variations is also very rewarding. In fact, test power usually exceeds the power consumption in normal operation mode, thus leading to wider $V_{DD}$ and/or T variations. This may induce a test result which is a false negative, i.e., the product is defect-free, it may even comply with timing specs in normal mode, but the abnormal delays in test mode may lead to a test failure [2].

Pipeline circuits exhibit a particular architecture in synchronous circuits, which allows the use of time borrowing techniques [3][4]. The methodology proposed in this paper makes use of time borrowing techniques to increase the robustness of pipeline-based digital circuits against power noise and temperature disturbances. However, enhancing the tolerance to delay-faults should not mask existing physical defects.

This paper is organized as follows. In Section 2, the previous work is provided. Section 3 describes the proposed methodology, highlighting the methodology limitations and methodology implementation. In Section 4, experimental results based on SPICE simulations and methodology improvements are presented and discussed. Finally, Section 5 outlines the main conclusions.

2. Previous Work

With different objectives, the Razor technique has been proposed [5] for pipeline based circuits, to correct (not to prevent) errors (de-synchronization errors) caused by aggressive DVS (Dynamic Voltage Scaling) techniques for power management of advanced microprocessors.

In order to enhance performance, design techniques, such as skew-tolerant domino design and timing analysis algorithm (including different amount of clock skew between different elements) have been introduced [6]. The time borrowing concept has been used in [3] to make skew-tolerant circuits operate at a higher speed. All these techniques consider static time definition, in the sense that the clock signals are pre-defined, and the amounts of time borrowed also pre-computed. None takes into account $V_{DD}$ and/or T variations. Incidentally, structural delay testing of such high-performance circuits is a difficult problem, as addressed in [7].
In order to guarantee correct timing performance in the presence of unwanted \( V_{DD} - V_{SS} \) transients, the ultimate solution is to reduce the clock signal frequency. However, performance degradation cannot be tolerated. Recently, a methodology has been proposed [8] that uses clock duty cycle (CDC) modulation to adjust the instant of data capture, in the presence of \( V_{DD} \) or \( T \) variations. In [8] a CSL (Clock Stretching Logic) block is used to modulate CDC and to enhance circuit tolerance without performance degradation. However, many practical synchronous circuits have simultaneous rising and falling edge triggered memory elements. This makes mandatory the use of complementary CSL blocks, which limits its usefulness.

To overcome this problem and still enhance circuit tolerance without degrade performance, the authors recently proposed a methodology ([9][10]) to control and adjust the clock phase (instead of the clock duty-cycle) of pre-determined memory cells’ clock signal (the critical memory cells), and thus enhancing tolerance in the presence of \( V_{DD} \) and \( T \) variations. The work presented in [9] is focused on the design of a mixed-signal cell, the Dynamic Delay Buffer (DDB) cell, that senses \( VDD \) and \( T \) variations and changes clock phase accordingly. In reference [10], the work was updated focusing now the application of the methodology to pipeline circuits and explaining how the improvements in tolerance can be obtained. There is also the application of the methodology to some circuit examples and SPICE simulation.

The current work is a continuation of the work presented in [9] and [10], and the main new contributions are: (1) the analysis of the methodology behavior with process variation; (2) the resolution of two methodology limitations previously identified (and now implemented and analyzed with process variations); (3) preliminary results on a proprietary software tool (Dynamic Delay Analyzer (DyDA)), to automatically identify the critical memory cells and calculate the DDB insertion points; and (4) the analysis of delay fault detection in a circuit with enhanced tolerance to \( VDD \) and \( T \) variations.

### 3. Proposed Methodology

In the proposed methodology, a dynamic time borrowing strategy is used to dynamically delay the clock signal of specific memory cells, providing additional time to accommodate the increased signal propagation delay through the logic whose \( V_{DD} \) or temperature is being disturbed. This allows controlling the instant of the active clock edge trigger for the memory cells with the smallest time slack margin.

The methodology is illustrated in Figure 1. When a \( V_{DD}-V_{SS} \) reduction (or \( T \) increase) increases propagation delays in signal paths, synchronization errors may occur in memory cells driven by the circuit critical paths. We refer these key memory cells as **critical**. The solution is to dynamically adapt the clock skew of these critical memory cells, in order to allow them to capture the correct data, with the addition of a DDB cell, a minimum overhead cell placed in the clock signal path of the critical memory cells. A DDB cell acts as a sensor (sensitive to temperature and/or \( V_{DD} \) fluctuations) and as an actuator, introducing the necessary delay (skew) in the clock signal and borrowing time from the subsequent pipeline stage to accommodate the delayed signal, while maintaining at-speed clock rate.

An architecture that implements the DDB functionality, powerful for its simplicity of usage and reduced area overhead [9], can be two simple unbalanced inverters (referred here as architecture \( a1 \)). This is used when the input stage of the memory element have a buffer in the clock lines. However, a second architecture is proposed in [10], using two additional balanced inverters (architecture \( a2 \)), to restore fast transitions at DDB output. Sizing appropriately the transistors allows changing the DDB’s delay dependence on \( V_{DD} \) and \( T \) variations [9].

The innovative idea in the proposed methodology is to adapt the clock delay (skew) locally and dynamically, as needed. Clock delay may be temporarily modified for some functional parts of a SoC, but maintained unchanged for the other parts. This does not result in a circuit functional error if all the combinational parts of the circuit keep on working synchronously.

![Figure 1. Dynamic Delay Buffer (DDB) application and operation.](image)

### 3.1. Limitations and Proposed Solutions

The proposed methodology has two limitations. One is associated with the presence of short delay paths. In fact, the use of a clock skew in a memory element raises the possibility that a short delay path in its input
combinational logic cone will corrupt the data in the memory element. To prevent this corruption, a minimum-path length constraint is added at the input of each clock skew controlled flip-flop in the design. As done in other design solutions ([5]), these minimum-path constraints result in the addition of buffers during logic synthesis to slow down fast paths. In our approach, we simply use an additional DDB cell in the data short path (see Figure 2(a)) to dynamically change the small path propagation delay accordingly with the controlled clock delay. The minimum-path constraint is equal to the propagation delay of the DDB cell at nominal $V_{DD}$-$V_{SS}$ and temperature values.

The second limitation is associated with the application of the method in pipeline circuits. In this case, if a critical path is followed by another critical path in the following pipeline stage, the methodology still applies and we propose two possible solutions, that can be used separately or together: (1) the addition of more than one DDB cell in the clock path of the second pipeline stage (or a DDB cell with enhanced skew capability) (see Figure 2(b)); and (2) the addition of a new pipeline stage to accommodate the borrowed time.

In case of applying the methodology in generic sequential circuits (or FSM), the limitations are generally the same, however the second limitation presented can be found prohibitive when circuit loops and complexity grows (some particular loops make the circuit insensitive to clock skew variations).

$$\text{Figure 2. Solutions to overcome methodology limitations: (a) the short path problem; (b) the consecutive critical paths problem.}$$

### 3.2 Methodology Application

Time borrowing techniques have been widely used in pipeline based circuits, because of their particular architecture [3][4]. Hence, the methodology presented in this paper is easily applicable to pipeline based circuits. However, as it can be seen on Section IV, it is possible to achieve improvements in time slack margins on FSM (Finite State Machines) circuits, depending on their structure.

To analyze the applicability of the proposed methodology in a circuit, the first step is to perform a static timing analysis (STA), to evaluate the critical paths and the short paths. It is not the purpose of this work to define a STA algorithm or tool. Instead, we use commercial EDA tools (like Primetime from Synopsys) to perform the STA analysis, focusing our work on integrating commercial tools with proprietary tools to interpret the STA results, decide the best implementation procedure and perform a complete design flow.

Another key issue in the methodology implementation is the critical memory cells selection. A proprietary software tool (DyDA, Dynamic Delay Analyzer) was developed to analyze the dynamic delay insertion, in order to automatically perform the critical memory cells selection and DDB insertion analysis. This analysis is based on the calculation of the expected time slack margins' improvements achieved with each DDB insertion. The purpose of each critical memory cell selection is to enhance the time slack margin of the critical path. The algorithm tests the DDB insertion in the most critical memory cell, updates the time slack margins, and repeats the calculus until no more improvements can be achieved or when a violation occurs (like feedback loops in critical memory cells). In some circuit topologies, it can be determined more than one level of tolerance improvements, depending on the number of DDBs to insert in the clock path of each memory cell.

This paper shows preliminary results on the critical memory cells automatic selection and DDB insertion. Research work is ongoing, in order to automatically integrate the methodology in a complete design flow.

### 4. Results

In order to demonstrate the effectiveness of the proposed methodology, extensive SPICE electrical-level simulations under the Cadence framework have been performed, using the CMOS AMS 350nm technology. All the application circuits presented in this paper consider only $V_{DD}$ variations. However, previous papers presented temperature variation examples, as well as both temperature and $V_{DD}$ variations, like [8], [9] and [10].

The first example is a Pipeline Multiplier with width 4 and 2 stages. This pipeline multiplier has 52 logic gates, 24 flip-flops, 9 inputs and 8 outputs. In this case, the combinational critical paths are the combinational multiplier paths and they all end at one register (width 8). To implement the proposed
methodology in this multiplier, we have to delay the data capture in this register, borrowing time from the next pipeline stage. Adding one DDB cell to connect all flip-flops in the register allows to dynamically delay the capture of multiplier’s combinational critical data.

Figure 3 shows the SPICE simulation results for the pipeline multiplier circuit, stimulated with random vectors. As we can see in Figure 3(a), the value of one flip-flop in the critical path changes when \( V_{DD} \) is decreased. Although, when we introduce the modifications described above (insertion of one DDB cell), the clock skew at the critical register changes to a delay of +12% from the original clock period, at a depleted \( V_{DD} \) of 2.7V. This change allows the register to capture the correct data in the presence of the error, as we can see in Figure 3(b).

Consider now a pipeline example with 3 stages, where a combinational critical path (implemented with 23 inverters in chain) located between two stages is followed in the next pipeline stage by a non-critical combinational path (implemented with 9 inverters’ chain). Considering the operation with \( V_{DD} \) down to 90% of its nominal value, the maximum clock frequency is \( f_{clk} = 0.455 \) GHz. However, this maximum frequency is reduced to \( f_{clk} = 0.4 \) GHz (12% degradation), if we want all the samples to pass the performance test at a nominal \( V_{DD} \) and allow worst-case conditions for process variations. This value is again reduced if we want to enable all the samples to pass the test with a \( V_{DD} \) of 90% of its nominal value (\( f_{clk} = 0.357 \) GHz, an overall 21.5% degradation). Additional performance degradation results from assuming correct behavior within a temperature range (commercial or military specs). In fact, the operation with worst-case conditions reduces performance to accommodate the uncertainty of all variables, taking a share for temperature, another for \( V_{DD} \) and another for process variations.

In this example, attention is focused on process variations. Hence, the frequency was set to \( f_{clk} = 0.4 \) GHz. Monte Carlo simulations (100 iterations for each \( V_{DD} \) value, with process and mismatch analysis variation) were performed to determine how the performance is being degraded when \( V_{DD} \) is reduced. To measure this degradation, a Performance Yield (PY) metric was introduced. In this context, it is defined here as the quotient between the number of samples which pass the performance test and the total number of samples:

\[
PY = \frac{n_{good\ samples}}{n_{total\ samples}} \tag{1}
\]

In Figure 4 this Performance Yield degradation with \( V_{DD} \) reduction is shown for the 3-stage pipeline example. The line with squares represents the normal design, where no (deliberate) skews were introduced in pipeline stages. Using the DDB basic cell (\( a1 \)) in the critical path stage flip-flop, the performance yield was increased for all different \( V_{DD} \), as shown in the line with triangles. Basically, the introduction of the DDB cell allows us to accommodate both \( V_{DD} \) and process variations with the time penalty associated with only \( V_{DD} \) variations (normal design).

It is interesting to notice that, if two common balanced inverters were used in the DDB (instead of two unbalanced inverters), the result would be worse, with less improvement in performance yield. This can be referred as “static skews” and is represented in Figure 4 by the line with circles. Reinforcing what it was said in Section III, dynamic skews are less...
intrusive to the circuit and can be more effective, when compared to static skews.

Nevertheless, unbalancing the inverters in the DDB cell (a1 architecture) causes also the weakness of its driving capability. To overcome this problem, restore fast transitions at DDB output and/or add an extra clock skew, the DDB basic cell can be buffered, creating static and dynamic clock skews (a2 architecture). The results show improvements in the performance yield, represented by the line with diamonds in Figure 4. In particular, for a reduction of 21% in the supply voltage, 29% yield gain is achieved when comparing the proposed methodology with the static skew, and this performance yield increases to 48% when compared to the normal design (with no deliberated skews introduced). The results can also be analyzed in the performance point of view. With skews and at the experimented frequency of operation, it is guaranteed that all the samples pass the test with a VDD reduction of 15%. When compared to the normal design, as mentioned before, only at a reduced clock frequency of fck=0.357 GHz we could have PY=100% (performance yield) with a VDD reduction of 10%.

As referred in Section III.A, more than one critical path in consecutive pipeline stages can limit the usefulness of the methodology or limit delay tolerance improvements. To evaluate one possible solution to the problem, the original 3-stage pipeline was augmented with an extra critical path stage (implemented with 23 inverters in chain, the same as the existing critical path), located before the existing critical path pipeline stage. In this case, the application of the methodology requires the solutions described in Figure 2(b). The critical memory cell terminating the first critical path captures now a signal delayed by one DDB cell, while the memory cell terminating the second critical path captures now a signal delayed by two DDB cells, borrowing time from the following stage. Monte Carlo simulations for different VDD values for this 4-stage pipeline shows similar results compared with the 3-stage pipeline (and one critical path), as it can be seen still in Figure 4, represented by the line with crosses.

Table 1 presents results from the proprietary software tool DyDA, when analyzing the applicability of the proposed methodology to several benchmark circuits from ITC’99 and to free open cores from [11]. From the table we can see that the methodology can not be applied in two circuits: “B02” (a FSM that recognizes BCD numbers) and “B10” (a voting system). The limitation in these circuits is a feedback loop in the critical memory cell that drives the critical path. Other FSMs with feedback loops are present in the table (in fact, only three are pipeline based circuits), but as they are not in a critical memory cell, the methodology can still improve tolerance.

Notice that circuits “B09” and “Bal. PM 4-2” have two different results. These are two different methodology applications for the same circuits, with different tolerance improvements.

One important implementation issue is that the number of DDBs in the clock path referred in the table can be substantially reduced, because each DDB cell can drive several Flip-flops (instead of one for each FF, as it was calculated). Another important implementation issue is that the methodology can be applied with standard DFT (Design For Testability) techniques, like scan-path or BIST. In the table, the circuit “B06” has an implementation with scan-path and only additional DDBs placed in the data scan path are needed.

However, delay-fault defects must not be masked with the enhanced tolerance. The next simulations show that parametric delay tolerance insertion does not necessarily reduces delay fault detection, as multi-VDD or multi-frequency self-test can be used to recover detection capability.

A typical dependence cone of XTRAN is used as test vehicle. The dependence cone of XTRAN is driven with a 7-bit LFSR (primitive polynomial) and a 7-bit MISR is used at the single output of the cone. The critical path corresponds to the X4-G5-G6-Q path. Fault injection is carried out assuming resistive bridging defects, using a discrete set of resistance values, Rtest. Figure 5 shows the dependence cone of XTRAN with LFSR and MISR and the DDB cell inserts in the end of the critical path. Another DDB has been inserted between the EXOR gates to give an extra time to the short path.

In order to increase the visibility of the differences between fault-free and defective signatures, decimal values of the signatures were obtained and plotted (e.g., in the following Figures, “127” corresponds to the “1111111” signature of the MISR). Figure 6 shows the set of faulty signatures in presence of a resistive bridging and Figure 7 shows the same but in the circuit with DDB cells.

In the circuit without DDB cells, the first detection of the fault appears at VDD=3V (90.91%) when the circuit recognizes BCD numbers) and “B10” (a voting system). The limitation in these circuits is a feedback loop in the critical memory cell that drives the critical path. Other FSMs with feedback loops are present in the table (in fact, only three are pipeline based circuits), but as they are not in a critical memory cell, the methodology can still improve tolerance.

![Table 1](https://example.com/table1.png)

Table 1. Automatic selection of critical memory cells and tolerance improvement analysis in benchmark circuits.
value of signature is “59”. In Figure 7, as expected, the faulty signature appears later, \( V_{DD}=2.7V \) (81.82%), the defect is observable at lowered \( V_{DD} \) due to the time increased for the DDB cells. These results prove that the multi-vdd or multi-frequency self-test, with parametric delay tolerance insertion, can still be used to recover detection capability.

5. Conclusions

A methodology has been proposed in this paper that aims to increase the delay-fault tolerance of pipeline based digital circuits to power-supply voltage and/or temperature variations in a SoC. The goal is to keep circuit synchronism and performance (i.e., to keep at-speed operation) even in the presence of \( V_{DD} \) and/or T disturbances that, due to increased signal path delays, may cause synchronism disturbances.

SPICE simulations demonstrate the effectiveness of the proposed methodology. Monte Carlo simulation also demonstrates that the proposed methodology is still effective in the presence of process variations. A proprietary tool was introduced to automatically select the critical memory cells and evaluate the insertion of DDBs and the improvement margins. It was also shown that multi-vdd and multi-frequency tests can be used to allow the detection of resistive delay-fault defects, when tolerance is enhanced.

References