A DFT Technique for Low Frequency Delay Fault Testing in High Performance Digital Circuits

Bhaskar Chatterjee, Manoj Sachdev
Dept. of Electrical & Computer Engineering
Waterloo, Ontario,
Canada
{bhaskar, msachdev}@vlsi.uwaterloo.ca

Ali Keshavarzi
Circuit Research, Intel Labs
Hilsboro, OR,
USA
ali.keshavarzi@intel.com

Abstract

This paper presents a DFT technique for delay fault testing of high performance, dynamic CMOS circuits. A high performance, delay fault testable, 16-bit adder is designed in 0.18μm CMOS technology. Simulations for the adder demonstrate that this technique can detect delay faults greater than 35ps and improves delay fault detection capability. It also allows at least 10X reduction in test mode clock frequency. Furthermore, the proposed method is capable of providing delay fault diagnostics. However, the proposed DFT technique increases delay by 8.6% with minimal power penalty.

1. Introduction

The clock frequency of high performance μP has exceeded 2GHz. Over the past 20 years, aggressive scaling of CMOS process technology has resulted in a 30% annual speed improvement for semiconductor circuits. However, tester speed has improved by only about 12% every year. Figure 1 shows data from the International Technology Roadmap for Semiconductors (ITRS'01) [1], for device under test period (DUT), automatic test equipment (ATE), and overall device accuracy requirements.

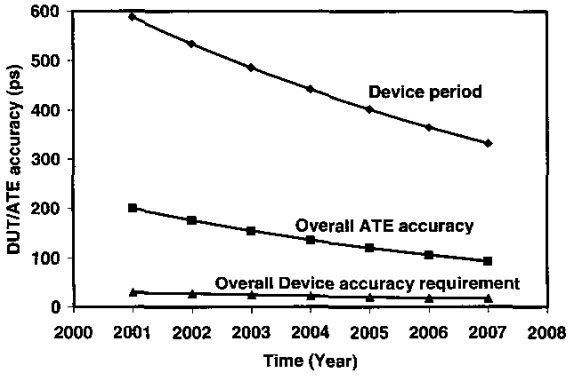

Figure 1: ITRS'01 projections for DUT and ATE performances

In the 1980s, ATEs typically offered a headroom of 5X or more over the DUTs. However, this advantage has now almost disappeared, and if the current trends continue, tester-timing errors are expected to approach the cycle time of the fastest devices [1]. As a result, it is believed that at-speed testing will become increasingly more difficult. Thus, tester inaccuracy along with scaled geometry, and higher device speed is expected to compromise IC yield and quality. Moreover, the increased number of DUT pins, demand for higher ATE accuracy, and larger vector memories are expected to increase the cost of the state-of-the-art ATEs. The above discussion motivates us to investigate low frequency design-for-testability (DFT) techniques for high performance dynamic circuits such as adders and arithmetic logic units (ALUs), with minimal performance/power overhead.

2. High-Performance Circuit Testing: Background

VLSI defects are physical deformations caused by missing or extra material and manifest in the form of shorts or opens. Depending on their impact, defects are typically classified as 1) global and 2) local defects. Global defects generally affect large areas or even entire wafers. On the other hand, local defects impact a smaller area on a die. Local defects are more difficult to detect/control, and require rigorous test practices for their detection. As reported in [2,3], local defects causing timing-only failures are becoming increasingly important in scaled deep-sub-micron (DSM) technologies. These defects often result in parametric failures and can cause delay faults that are difficult to detect [4]. Such defects are known to pose a serious threat to the long-term reliability of complex ICs.

Techniques used to detect these defects can be broadly categorized as 1) indirect (correlation based) methods and, 2) direct test methods. Keshavarzi et. al. presented an example of an indirect test technique in [5], where I_{DDQ} test results are correlated with the maximum operating frequency of a 32-bit μP. The fact that shorter channel lengths lead to higher operating frequency and quiescent leakage current forms the basis of this technique. Another methodology proposed by Hao and McClusky [4], is based on the Very Low Voltage Test (VLV) technique where ICs are performance tested at reduced V_{dd}. It was observed that delay faults were more noticeable at a lower V_{dd} and hence easier to detect. However, the VLV technique affects only the transistor delay, while leaving the interconnect delay largely unchanged. In modern μPs, an increasingly larger segment of the total delay is in the interconnects. Hence, this method's suitability in DSM is being eroded. Furthermore, aggressive scaling of CMOS technology with the reductions in V_{dd}, V_t, and L_{eff} is making both VLV and I_{DDQ} techniques less effective.

0-7803-7542-4/02 $17.00 © 2002 IEEE

There has been an increased focus on direct test techniques which rely on 1) ATEs with improved capabilities/higher frequencies and, 2) DFT and BIST (Built-In Self Test) for DUTs for improved testability. It should be noted that of the two approaches the latter is the less expensive. Direct test methods have been presented in [6,7]. These methods are based on the incorporation of additional DFT structures and the creation of a low frequency TEST mode. The basic idea is to include an externally controlled, quantifiable delay to enable slow-speed testing. Such techniques are applicable to combinational circuits bounded by flip-flips (F/Fs) as shown in Figure 2. The F/Fs are designed to have large D-Q delay (T_{prop}) in the TEST mode. Therefore, in the TEST mode, the test clock frequency can be reduced while the combinational logic is tested for specified delay. However, [6,7] fail to detect delay faults with resolution less than $T_{nominal}$ (Figure 2). In addition, there exists the need for the routing of an externally available timing critical signal (Test_Clk) in the TEST mode.

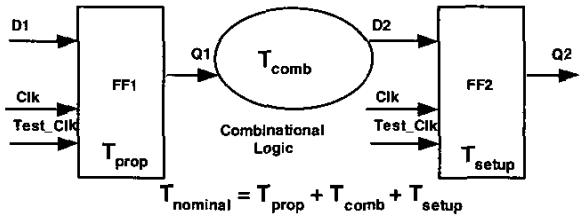

Figure 2: Controlled Delay F/F (CDFF) block diagram

DFT techniques enabling detection of delay faults in CMOS circuits are presented in [8,9]. The outputs of combinational circuits are sampled by clocked sensing circuits. However, these techniques suffer from the following drawbacks: 1) need for externally available timing critical signal(s) in the TEST mode, 2) inability to detect faults with resolution less than the critical path delay, and 3) operation of the delay fault "sensing" circuit based on charge sharing between parasitic node capacitances. In addition, [8,9] do not explore the possibility of low frequency, cheaper testing techniques. Dynamic CMOS circuit implementations offer higher performance over their static counterparts. Therefore, building blocks of high performance ICs, such as adders, ALUs and multipliers are often implemented using dynamic circuits. The work presented in this paper describes a DFT strategy for the testing of high performance, dynamic circuits at a reduced clock frequency. High performance logic blocks are normally organized in several stages of cascaded dynamic gates. Thus in this paper, we describe a mechanism of delay fault testing/diagnosis in multi-stage dynamic circuits. Finally, our objective is to test these high performance circuits at reduced clock frequencies. Our simulations indicate that by using the proposed DFT technique the TEST mode clock frequency may be reduced by 10X or more. The effectiveness of this technique is demonstrated using a high performance 0.18μm 16-bit adder.

3.1. DFT Technique: Basic Concept

Figure 3 shows the circuit and timing diagrams of a conventional 2-input dynamic AND gate [10].

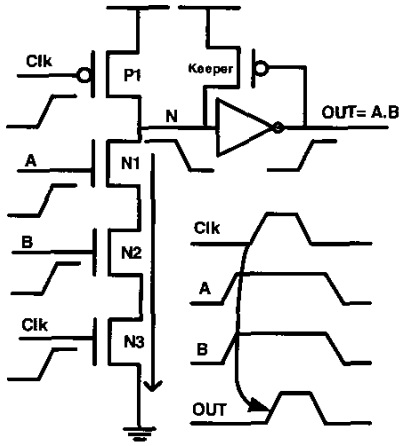

Figure 3: Conventional dynamic gate without DFT

Dynamic circuits are faster than their static counterparts and are extensively used to implement the performance critical blocks of high-end μP [11,13,15]. Therefore, in order to test high performance circuits at reduced clock frequencies owing to the possible non-availability of high-end ATEs, we need to devise a DFT strategy for dynamic circuits. The implementation of the proposed DFT technique is illustrated in Figure 4. This dynamic circuit has two modes of operation: 1) NORMAL, and 2) TEST. In the NORMAL mode of operation, input signal Test_Clk is a dc signal connected to V_{dd}. As a result, transistor N4 is always ON allowing operation of the circuit as in Figure 3. However, in the TEST mode, the Test_Clk input signal is inverted and phase-shifted with respect to the Clk signal, as shown in the timing diagram (Figure 4). The signals Clk and Test_Clk create an "evaluation window" equal to the phase shift between the two signals. It is only during this time that both N3 and N4 are ON allowing logic evaluation.

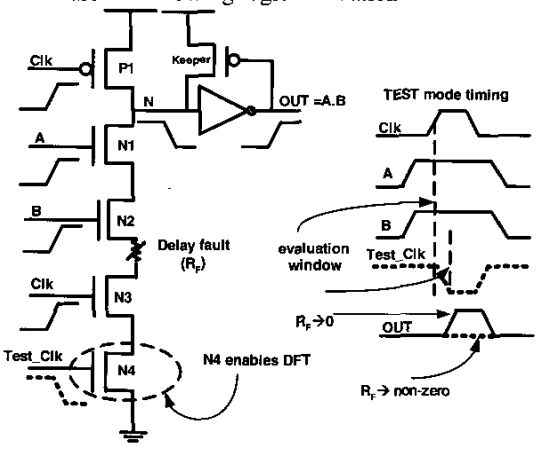

Figure 4: Dynamic gate with DFT (TEST mode operation)

Let us now consider the effectiveness of the proposed circuit modification in the detection of delay faults and its usefulness as a DFT technique. This is done by introducing a resistive defect, R_F, in the evaluation path causing a delay fault (Figure 4). We can now analyze the circuit operation for the following cases: 1) No delay fault (R_F = 0), and 2) Delay fault present (R_F = non-zero). In scenario (1), node N is successfully discharged in the available "evaluation window" and OUT correctly transitions as shown in the timing waveform ($R_F \rightarrow 0$ case). In scenario (2), (R_F = non-zero), the pull-down path RC time constant is larger, resulting in delayed evaluation of OUT. If the delay fault is larger than the "evaluation window", OUT remains at logic 0. Hence, the delay fault is converted to a more readily detectable stuck at fault at the output of the gate. Note should be taken of the following factors in the context of the above-mentioned technique:

1) Delay fault detection depends on the overlap duration between the signals Clk and Test_Clk and not on their absolute frequencies. This allows reduction of the clock frequency in the TEST mode by 10X and enables slow speed testing of high performance circuits.

2) Test_Clk is a delayed and inverted version of Clk signal. Thus, it can be generated on-chip from Clk thereby eliminating the need for any extra timing critical inputs to the DUT.

3) In the NORMAL mode, Test_Clk is a dc signal equal to V_{dd}. Thus, N4 is not switched every clock cycle resulting in minimal additional power overhead.

4) The addition of the extra footer transistor, N4, increases the nominal gate delay. Thus, the "overlap time" of Clk and Test_Clk must be appropriately adjusted.

5) In this study, delay faults have been modeled as resistors as has also been reported in literature [4,9,12]. In this paper, parametric simulations in HSPICE with different values of defect resistance (R_F) have been performed to model faults with variable delays.

6) In this technique, we focus on the impact of delay faults on the evaluation time. This technique does not address the impact of defects on the pre-charge time. This is because the evaluation time is critical and limits circuit performance as opposed to the pre-charge time, which happens in parallel and is non-critical from timing considerations.

7) In subsequent discussions, our proposed DFT methodology is demonstrated for gates with a maximum of 4 transistors in series between V_{dd} and ground. This reduces the impact of body effect and allows design of high performance circuits/logic blocks. However, it should be noted that the DFT technique in itself does not impose such a restriction on the stack height of the DUT gates.

Figures 5(a) and 5(b) illustrate delay fault simulation results without and with the above-mentioned DFT technique. All simulations are for 0.18µm technology at a V_{dd} of 1.8V for the typical-typical process corner. Figure 5(a) depicts the output response of the dynamic gate in the absence of DFT for parametric simulations as R_F varies from 0Ohms to 7kOhms in steps of 1kOhms (labels 1 to 7). Clk is a 1GHz, 50% duty cycle signal. As a result, the effective "evaluation window" is equal to 500ps (50% of the Clk period). It is apparent from the simulations, that higher the value of R_F, larger is the output rise time. It is observed, that for the range of R_F from 0Ohms to 6kOhms, the circuit evaluates correctly (OUT > switching threshold of $V_{dd}/2$), with appreciably larger delay. However, when R_F = 7kOhms or more, the circuit fails to evaluate. It is apparent from Figure 5(a), that the circuit's delay behavior goes undetected until it fails completely.

Now, let us consider parametric simulation results for the same defect in the presence of DFT as shown in Figure 5(b). Simulations indicate that the defect free dynamic gate in Figure 4 requires 42ps for evaluation under nominal conditions ($T_{nominal}$). In order to account for process corner variations and not reject "good" parts while testing for ICs with delay faults we allow for an extra "safety margin" (δ), while estimating the "ovelap window" (T_{window}), between the Clk and Test_Clk signals. In this paper we use a "safety margin" equal to 25% of the nominal delay and estimate the "evaluation window" as shown in equation (1) below:

$$T_{window} = T_{nominal} + \delta = 1.25T_{nominal} = 53ps \quad \ldots\ldots\ldots\ldots(1)$$

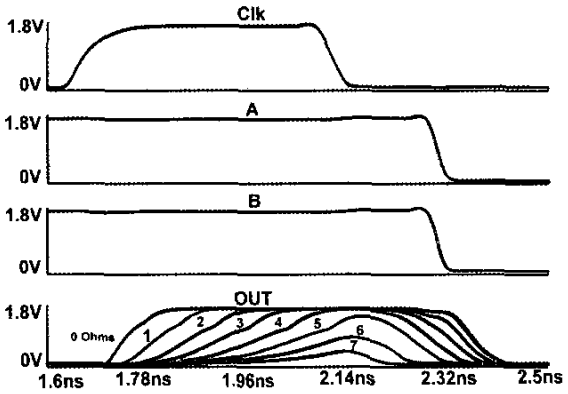

Figure 5(a): Delay fault simulation: without DFT

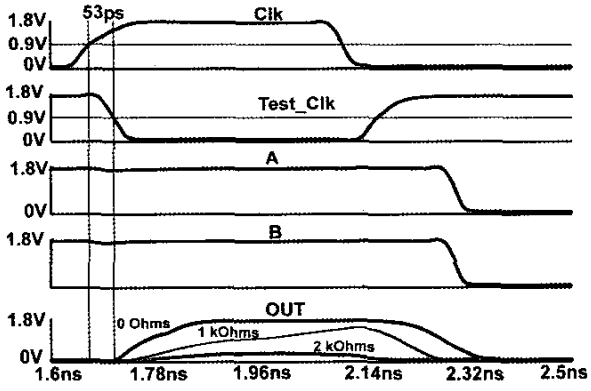

Figure 5(b): Delay fault simulation: with DFT

The parametric simulations shown in Figure 5(b) indicate that OUT reaches a voltage level equal or more than the switching threshold ($V_{dd}/2$) when R_F is in the range 0Ohm to 1kOhms. However, when the defect resistance R_F = 2kOhms or more, the delay impact is large enough and is detected as a logic failure at the output. These simulations demonstrate the ability of this DFT technique to:

1) Convert delay faults to more easily detectable stuck at fault at the gate output, and

2) Detect a larger range of R_F/delay faults.

In Section 3.2, the DFT technique is extended to compound domino logic. This enables us to implement a delay-fault-testable 16-bit high-performance adder (Section 4.1). The extension of the scope of this DFT technique to enable: 1) low frequency testing, 2) stage-to-stage delay fault testing and diagnosis are addressed in Sections 3.3 and 3.4.

3.2 Compound Domino Logic: Concept and DFT

Compound Domino Logic (CDL) is a circuit technique used to obtain enhanced circuit performance compared to n-MOS domino logic [11]. As shown in Figure 6, a CDL gate differs from a conventional n-MOS (p-MOS) domino gate (Figure 3) in that it implements complementary CMOS logic gates at the output of the dynamic stages instead of explicit inverters. As a result, it can implement the same logic function in fewer stages and is thus faster than its n-MOS (p-MOS) domino logic counterpart. In addition, OUT is actively driven and is logic low during pre-charge. Thus, it can be readily interfaced with the dynamic gate of the next CDL stage.

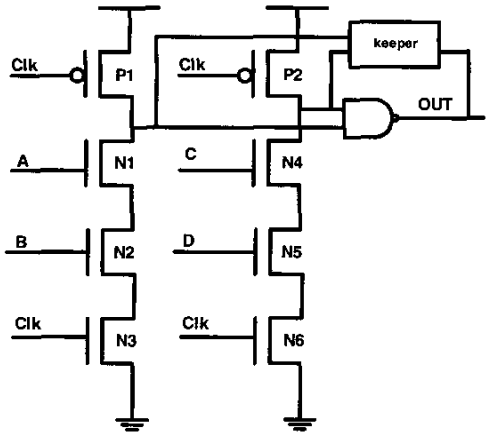

Figure 6: Conventional CDL gate: OUT = AB + CD

It is possible to extend the DFT technique presented in Section 3.1 to the CDL family to enable delay fault testing. A complex logic block (Adder, ALU, multiplier) can be designed by cascading and interconnecting several stages of CDL gates. This would result in the formation of a chain of alternate dynamic and static gates. Such an arrangement enhances circuit performance without compromising robustness and finds application in high performance logic blocks. Figure 7 shows six cascaded logic stages (keepers omitted in the figure for clarity) and represents the carry-generate-section of the 16-bit adder (discussed later in Section 4.1). All the dynamic gates have a clocked n-MOS footer transistor shown explicitly (N1, N3, N5). It is clear, that the footer transistor N1 in stage 1 is essential since input signals ($A_1 \ldots A_M$) to the logic block may not be domino compatible and hence may cause pull-down contention during pre-charge. However, for all subsequent stages, internal signals available at the output of the static logic blocks (P, Q) are domino compatible and can be directly interfaced with the n-MOS domino stages making transistors N3 and N5 optional.

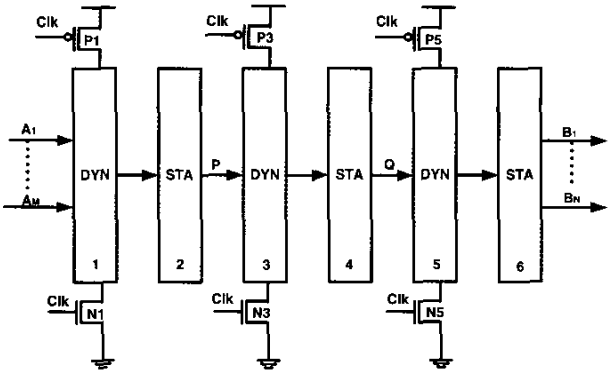

Figure 7: Cascaded CDL gates with clocked footer transistors

Thus, in our research, we utilized these transistors for DFT. In the NORMAL mode, N3 and N5 are connected to V_{dd} and are ON. However, in the TEST mode a signal that is phase shifted and appropriately delayed with respect to the Clk signal is used to control transistors N3 and N5 allowing delay fault diagnosis. The modified arrangement is illustrated in Figure 8.

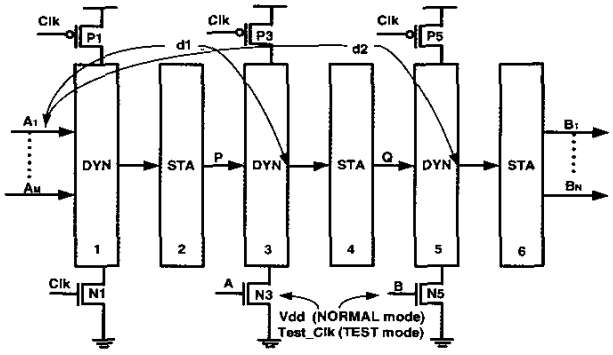

Figure 8: Improved CDL with DFT capability

This modification results in the development of a delay fault testable CDL family with the following features:

1) Equal stack heights for dynamic gates with and without DFT,

2) Reduced Clk load and buffer sizing, and hence lower overall power consumption compared to conventional CDL (Figure 7) in the NORMAL mode of operation,

3) Incorporation of DFT technique in high performance CDL gates,
4) Since the footer transistors are not switched in the NORMAL mode, it can be made wider to reduce performance impact without significant power penalty.

Figure 9 shows the power-delay product (PDP) plots in the NORMAL mode of operation for the different CDL families with and without DFT as a function of their fan-out. Graph 1 corresponds to the improved CDL family with DFT capability as described in Figure 8. Graph 2 corresponds to the conventional CDL family without DFT (described in Figure 6) while graph 3 pertains to the improved CDL family without DFT where the n-MOS footer transistors (N3, N5) are entirely removed for enhanced performance. Simulations indicate that for a fan out of 3, graphs 1 and 3, show up to 61-65% reduction in PDP compared to graph 2 (PDP_1 and PDP_3 vs. PDP_2). This can be attributed to (i) fewer clocked transistors, (ii) smaller clock buffers, and (iii) improved performance. In addition, graph 3 shows up to 9% PDP improvement compared to graph 1 (PDP_3 vs. PDP_1). This can be attributed to performance improvement arising from complete removal of the n-MOS transistors from the pull-down stack. Both graphs 1 and 3 have similar power consumption because the footer transistors are not switched in the NORMAL mode.

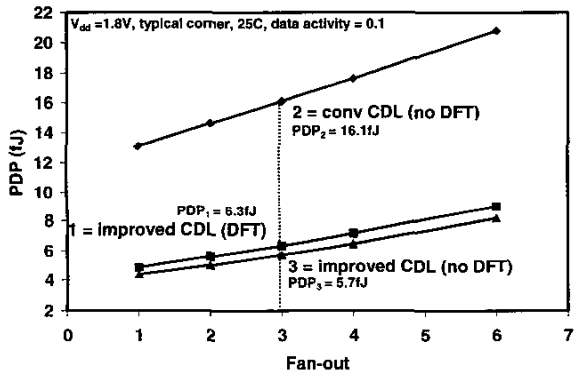

Figure 9: PDP vs. fan-out plots for CDL family gates

3.3 DFT: Enabling Low Frequency Testing

In Section 3.1, we observed that the effectiveness of the proposed DFT technique in diagnosing delay faults lies in the creation of an "evaluation window" in the TEST mode. This "window" is created by an overlap between Clk and Test_Clk signals as shown in Figures 4 and 5(b). Figure 10 shows parametric simulation results with Clk and Test_Clk signals operating at 100MHz while maintaining an "overlap window" of 53ps ($1.25T_{nominal}$). R_F increases from 0Ohms to 2kOhms in steps of 1kOhms. It is clear from Figure 10, that when R_F is less than or equal to 1kOhms, the extra delay due to R_F is not large enough to cause a logic failure and the output evaluates correctly. However, when R_F is equal to or more than 2kOhms, the output cannot evaluate in the "evaluation window" and thus the fault is detected. The defect resistance coverage obtained in Figure 10 is same as that in Figure 5(b) the only difference being the reduction of Clk and Test_Clk signal frequencies by 10X. This demonstrates the fact, that the proposed DFT methodology can be used as a low frequency-low cost solution for delay-fault detection in high performance DUTs.

Dynamic circuits are designed to operate above a certain minimum clock frequency (F_{min}). If the circuit is operated below F_{min}, the information on the dynamic node (N in Figure 4) may be corrupted due to charge sharing and leakage through the evaluation stack. Hence, for the DUT, we sized the keeper to ensure robust operation at the worst-case process corner and for 10X clock frequency reduction.

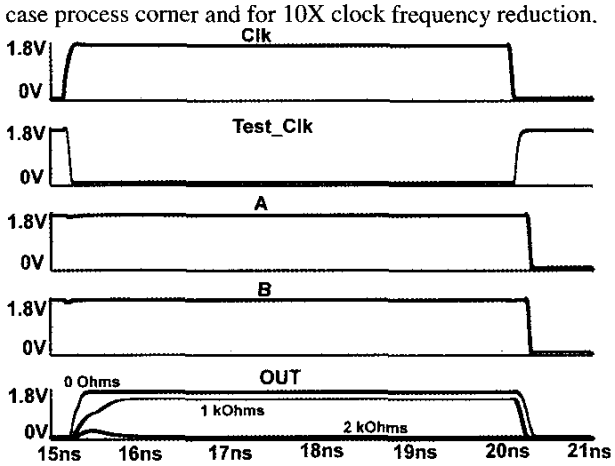

Figure 10: Delay fault detection with 10X reduction in clock frequency

3.4 DFT: Stage-to-Stage Delay Testing/Diagnostics

The concept of stage-to-stage delay fault testing can be understood with respect to the block diagram (Figure 8) and timing scheme shown in Figure 11. Stage-to-stage delay testing allows testing of logic between two successive stages of dynamic logic. Let us assume that, we need to perform delay testing for the logic stages 1-2-3 of Figure 8. This can be achieved in the TEST mode by doing the following:

1) Apply Clk to stage 1 (N1) and stage 5 (N5),
2) Apply inverted-delayed Clk (Test_13) to stage 3 (N3),
3) The overlap window between Clk and Test_13 is 1.25d1 ($T_{nominal}$ + safety margin). Note that d1 is the nominal delay ($T_{nominal}$) for correct logic evaluation from stage 1 to stage 3 under defect free conditions.

This scheme ensures the creation of a "tight" evaluation window for stages 1-2-3 while the timing between stages 3-4-5 is "relaxed" as shown in Figure 11. Test_13 signal in Figure 11 allows us to test for delay faults present in stages 1-2-3. Thus, any delay fault related logic failure detected at the primary outputs ($B_1 \ldots B_N$) would necessarily be located in stages 1-2-3. The above concept can be extended to detect delay faults present in stages 3-4-5 as well. This is achieved when the Clk signal ("relaxed" timing) is applied to N1 and N3 while Test_15 delayed by 1.25d2 is applied to N5 ("tight" timing).

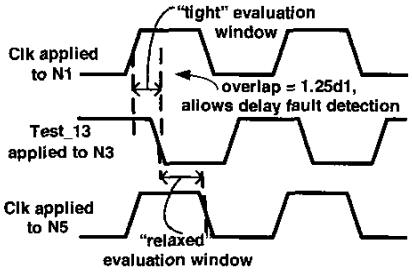

Figure 11: Timing diagrams for testing stages 1-2-3

The delay fault testing strategy for a DUT can thus be formulated as a two-step process. In step 1, stages 1-2-3 are tested for delay faults. If the DUT fails, we know that the delay fault exists in stages 1-2-3. However, if the DUT passes in step 1, we carry out step 2 of the testing process in which stages 1-5 are tested. If the DUT fails in this phase of testing, it is evident that the delay fault is located in stages 3-4-5. Thus, stage-to-stage testing allows us to carry out delay-fault diagnostics and debugging. However, such an arrangement requires additional DFT logic over the basic technique proposed in Section 3.1. These hardware overheads are enumerated as follows:

1) Extra dc input signals to select the particular stage(s) to be tested;

2) Extra logic for the local generation of the delayed-inverted clock signal(s),

3) Extra testing time for diagnostics and debugging.

3.5 DFT logic: Test_Clk Generation and MUX-ing

In this section, we deal with some of the implementation issues involved with the practical realization of the proposed DFT technique. Table 1 shows the truth table for mode and stage selection, while Figure 12 shows a MUX based implementation of the DFT scheme. The basic idea here is to be able to control the timing of the footer transistors N3 and N5 (Figure 8) using appropriate signals in the NORMAL/TEST modes. As seen in Figure 12, the two input signals Ctrl1 and Ctrl2 are used for the purpose of MUX control. The figure shows two MUX-es namely MUX1 and MUX2. MUX1 is the input MUX and is used to prevent the inverter chain from switching in the NORMAL mode of operation while MUX2 is used for mode/stage selections. Both MUXes were realized using static transmission gate logic. In addition, is shown a chain of inverters with two tap points that serve the purpose of generating inverted-delayed versions of the input Clk signal (Test_13 and Test_15). The transistors of the inverter chain and MUX-es are optimized so that signals Test_13 and Test_15 are delayed by 1.25d1 and 1.25d2 respectively with respect to Clk. It should be borne in mind, that in the TEST mode, the signal Test_13 is used to control N3 when stages 1-2-3 are tested for delay faults. Similarly, Test_15 is used to control N5 when stages 3-4-5 are tested.

Now we discuss the various modes of operation of the DFT logic with reference to Table 1 and Figure 12. When both Ctrl1 and Ctrl2 signals are logic 0, the DUT operates in the NORMAL mode with nodes A and B connected to V_{dd}. As a result, N3 and N5 are turned ON allowing the dynamic gates to evaluate. On the other hand, when Ctrl1 = 0 and Ctrl2 = 1 the DUT is in the NORMAL mode of operation with all the n-MOS footer transistors connected to Clk via nodes A and B. In this mode, the dynamic circuits have 50% of Clk cycle time for evaluation and operate like conventional CDL.

It is to be noted that when the control signal Ctrl1 = 0, the DUT is in NORMAL mode of operation and hence any switching of the inverter chain is unwarranted. Thus in this mode, MUX1 is used to connect node C to V_{dd} and bypass the Clk signal. This helps prevent unnecessary increase in Clk load and switching power loss in the NORMAL mode.

Table1: Truth table for DFT logic: enabling mode/stage selection

Ctrl1	Ctrl2	C	A	B	Comments
0	0	V_{dd}	V_{dd}	V_{dd}	NORMAL mode
0	1	V_{dd}	Clk	Clk	NORMAL mode; with clocked footers, no diagnostics
1	0	Clk	Test_13	Clk	TEST mode, diagnostic for stages 1-2-3
1	1	Clk	Clk	Test_15	TEST mode, diagnostics for stages 3-4-5

On the other hand, when Ctrl1 = 1, node C is connected to Clk allowing generation of Test_13/Test_15 signals for delay fault testing as shown in Table 1.

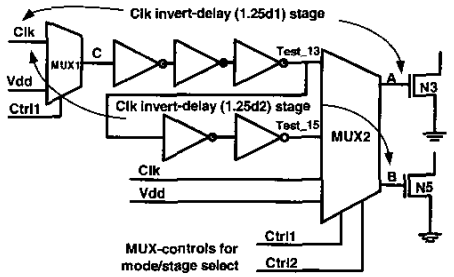

Figure 12: Scheme for local generation of Test_13/Test_15 signals

Suppose in the TEST mode it is desired to diagnose the presence of delay faults in stages 1-2-3. Also, let us assume that the $T_{nominal}$ for stages 1-2-3 is equal to d1. Now, in order to diagnose delay faults in stages 1-2-3, we need to apply a "tight" evaluation window for stages 1-2-3 while maintaining a "relaxed" evaluation window for stages 3-4-5. The above can be achieved by setting control signals Ctrl1 = 1 and Ctrl2 = 0 (Table 1). This allows signal Test_13 to be connected to N3 transistor while N5 is connected to Clk. This results in the creation of an "evaluation window" of duration 1.25d1($T_{nominal}$ + safety margin) for stages 1-2-3 while maintaining "relaxed" timing for stages 3-4-5. Consequently, any delay faults detected at the primary

outputs can be attributed to logic stages 1-2-3. On the other hand, the setting when both signals Ctrl1 and Ctrl2 are logic 1 allows delay fault diagnostics for stages 3-4-5. This is achieved by connecting N5 to signal Test_15 ("tight" timing) and N3 to Clk ("relaxed" timing). Test_15 is delayed by an amount 1.25d2 with respect to Clk. For the particular case of the 16-bit adder, simulation results in 0.18μm show that d1 = 138ps, while d2 = 267ps.

Thus, we now have a DFT technique that allows detection and diagnosis of delay faults in high performance CDL circuits at reduced clock frequency. In addition, we have described a scheme for the local generation of Test_13 and Test_15 signals thereby eliminating the need for any external timing critical clock signals in the TEST mode. The above DFT implementation to generate delayed local clock signals is similar to the concept of Clock Delayed Domino logic (CDD) discussed by Yee and Sechen in [13] where a "safety margin" of at least 20% of $T_{nominal}$ was used to account for process variations. In addition, it was reported in [13], that a careful layout with local buffers and matched interconnect lengths were used to equalize the delays of the logic gates and the inverter delay chains on the clock path. It is believed that by following the same design principle it will be possible to maintain appropriate phase relationship between Test_13/Test_15 and Clk signals in a practical realization.

4.1 Delay Fault Testable Adder

In this section, our DFT technique is implemented for a 16-bit adder. The adder is used as a vehicle to demonstrate the effectiveness of the scheme for high performance circuits. In particular, the current section discusses the following issues: 1) 16-bit adder architecture and design, 2) Power and delay penalty incurred due to incorporation of the DFT scheme, 3) Impact of process corner variation on safety margin and, 4) Low frequency delay fault testing. The basic architecture of the 16-bit propagate/generate based adder is shown in Figure 13. The PG (propagate/generate) block and carry-merge-tree form the carry generate section; while the Carry Select Adders (CSA) and the output MUX-es form the sum generate section of the adder [14]. The PG block and carry-merge-tree sections are the performance critical blocks and are implemented using CDL gates [15,16]. The CSA adders operate in parallel with the carry generate section and are implemented using static CMOS gates while dynamic logic is used for the output MUX-es. The PG block forms the propagate (P) and generate (G) terms based on the primary inputs A[15:0] and B[15:0] (Eq. 2 and 3). The carry-merge-tree employs a binary merge algorithm that implements the recursive logic equation [11] as shown below (Eq. 4), to produce every 1-in-4 carry signals, namely C_3, C_7, C_{11} and C_{15}:

$\mathbf{P_i = A_i + B_i}$ (propagate function) ..... (2),
$\mathbf{G_i = A_i . B_i}$ (generate function) ..... (3),
$\mathbf{C_i = G_i + P_i . C_{i-1}}$ (carry-merge function) ..... (4),

where, A_i and B_i represent i^{th} bit-position of input vectors A[15:0] and B[15:0], P_i and G_i are the corresponding propagate and generate signals while C_{i-1} and C_i are the input and output carry signals for the i^{th} bit position. For each of the 4-bit adder blocks (A, B, C, D) shown in Figure 13, there are two parallel 4-bit CSA adders. One of these adders generates sum output assuming input carry to be logic 0 while the other generates the sum output assuming input carry to be logic 1. For example, the carry select adders pertaining to Block C, in Figure 13, generate two sets of sum signals $S11:8^{(1)}$ and $S11:8^{(0)}$ using the corresponding input block carry C_{in}^{c} to be logic 1 and logic 0 respectively. This happens in parallel with the carry-merge-tree and the CSA output signals become available at the inputs of the 2:1-MUX-es. When the C_7 signal becomes valid, the appropriate sum signals are selected and are available at the S11:8 outputs of the adder.

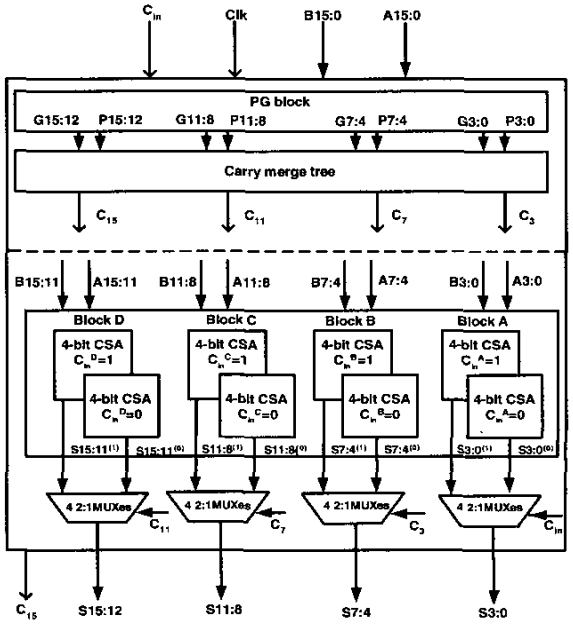

Figure 13: 16-bit adder block diagram

This adder is targeted for high performance application. Thus, all the static and dynamic gates were implemented with two inputs. This restricts the stack height to a maximum of 4 transistors. The sparse carry-merge-tree in parallel with the CSA adders further enhances performance. The 1-in-4 carry tree has fewer logic gates and is expected to have fewer interconnects, and hence lesser power consumption. The output stage is a 2:1 dynamic MUX instead of an explicit XOR further improving performance.

4.2 DFT Adder: Power/Performance Impact

The 16-bit adder architecture described above is implemented using the DFT circuit technique described earlier. This allows us to develop a high performance, delay

fault testable adder. Three different adder implementations based on the architecture of Figure 13 are listed below:

Case (1): Conventional CDL adder without DFT (based on Figure 6),

Case (2): Improved-CDL adder with DFT (based on Figure 8),

Case (3): Improved-CDL adder (no DFT): This adder has the n-MOS footer transistors (N3, N5) completely removed to enhance performance.

Table 2 shows the simulation results for the worst-case power and delay for the above-mentioned adders in the NORMAL mode of operation. These results enable us to quantify the overhead impact of the DFT technique on high performance circuits and logic building blocks.

Table 2: Worst-case power and delay for adders

	Case (1) Conventional CDL (without DFT)	**Case (2)** Improved CDL (with DFT)	**Case (3)** Improved CDL (no DFT)
Worst case power(mW) (A=FFFF; B=FFFF)	48.5 **(reference)**	43.11 **(-11.1%)**	43.08 **(-11.2%)**
Critical path delay(ps) (A=FFFF; B=0001)	309 **(reference)**	303 **(-1.9%)**	278.9 **(-9.7%)**

Simulations for the three adders were carried out at an input clock frequency of 1GHz and under identical simulation conditions (V_{dd} =1.8V, 25^{0}C, typical process corner, load =20fF@ each primary output, data activity = F_{data}/F_{clk} = 0.1). Our results indicate that the Improved-CDL adder without DFT shows 11.2% lower power and 9.7% lower delay than the Conventional CDL without DFT. In addition, the Improved-CDL adder with DFT has 11.1% lower power and 1.9% lower delay than the Conventional CDL without DFT. The reduction in the worst-case power for Cases (2) and (3) over Case (1) can be attributed to fewer clock transistors, smaller clock buffers and reduced power consumption in the clock network. Case (2) shows higher critical path delay compared to Case (3). This can be attributed to the complete removal of the footer transistors and subsequent stack height reduction for Case (3). It should be noted that for Case (2), the n-MOS footer transistors are connected to V_{dd} and are always ON in the NORMAL mode of operation. Hence, both Cases (2) and (3) have almost the same worst-case power consumption. The above results are in line with the simulation results already presented in Figure 9. The above simulation results enable us to conclude that the proposed DFT technique has minimal power penalty. However, for a circuit like a 16-bit adder, it results in an increase in critical path delay of 8.6% (Case (2) vs. Case (3)).

4.3 Impact of Process Corner on Safety Margin

As explained earlier, we budget a safety margin (δ) equal to 25% of the nominal stage delay ($T_{nominal}$) being tested. The delayed signals Test_13 and Test_15 are generated locally using the DFT logic discussed in Section 3.5. However, we need to consider the impact of process variations on the safety margin in a practical implementation. For the fast-fast process corner, the stage 1-2-3, stage 3-4-5 and critical path signals pull in compared to the typical-typical corner. Similarly, for the slow-slow corner, the above signals push out. Under these circumstances, if the Test_13 and/or Test_15 signals do not pull in or push out proportionately, it is possible for the safety margin to be altered. If the logic and DFT signal delays do not track with process corner variations, it might result in the rejection of defect free parts, which is clearly unacceptable. Table 3 shows simulation data for the variations in stage 1-2-3, stage 3-4-5, critical path, and the safety margin timings with process corner.

Table 3: Stage delay, critical path and safety margin variation

Process Corner	$T_{1\text{-}2\text{-}3}$ (ps)	$T_{3\text{-}4\text{-}5}$ (ps)	$T_{critical}$ (ps)	$\delta_{1\text{-}2\text{-}3}$ (ps)	$\delta_{3\text{-}4\text{-}5}$ (ps)
Typical-Typical	**138**	**267**	**303**	**35**	**67**
Fast-Fast	102.7 **-26%**	203.3 **-24%**	235.3 **-24%**	26.8 **-23%**	53.3 **-21%**
Slow-Slow	181.6 **+31.6%**	343.5 **+28.6%**	393.3 **+30%**	44.8 **+28%**	85 **+27%**

It is clear that at the fast-fast corner, all the signals and the safety margins pull in by approximately the same amount (21% to 26%) compared to the typical-typical process corner. Similarly, they also slow down proportionately by the same amount (27% to 31.6%) at the slow-slow process corner. This indicates that the Test_13/Test_15 and the safety margins track the logic delay of the DUT satisfactorily with process variation.

This DFT technique involves the introduction of extra clock transistors (MUX-es/delay line) to enable low frequency delay fault testing. This extra clock load is expected to adversely impact the clock skew of high performance circuits. However, the above data indicates that the safety margin tracks the circuit delay across process corners in the TEST mode of operation. In addition, with the use of local delay lines and careful layout it is believed that the impact on clock skew can be minimized in the TEST mode. In the NORMAL mode, the delay lines are disconnected from the clock network using MUX1 (Figure 12) and the footer transistors are connected to V_{dd}. This results in a minimal increase in the overall clock load in the NORMAL mode of operation. Hence, in both the NORMAL and TEST modes the additional clock skew due to the proposed DFT technique should be within acceptable limits. However, the actual quantification remains the topic of future research.

4.4 Delay Fault Types and Detection Algorithm

During our research, we introduced different types defects causing delay faults in the adder. These defects can broadly be classified as either series or parallel types. Figure 14 shows the position of some of the typical defects in a simple

CDL gate (keepers omitted in figure for clarity). Series defects, as the name implies are in series with a transistor.

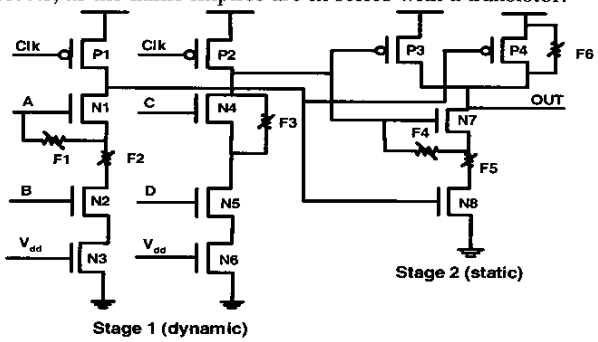

Figure 14: Delay fault locations in CDL gates

In Figure 14, the defects F2 and F5 are examples of series defects. The delay impact of such defects is proportional to the value of their fault resistance (R_F). On the other hand, defects that are in parallel with a transistor result in parallel faults as for example F1, F3, F4 and F6. For such faults, the delay impact is inversely proportional to the fault resistance. It should be noted that during this study, series and parallel faults were introduced in both the static and the dynamic gates. Defects were also introduced in critical and non-critical logic paths of the adder. During the course of our research, we introduced only one defect at a time in the adder. Now, we will discuss the algorithm adopted to detect the presence of a delay fault in the adder.

Step 1: Defect introduction step: Introduce a defect in the fault free adder.

Step 2: Assign a value to the defect resistance (R_F) to cause a catastrophic logic failure at primary output(s) (R_F = 0 for parallel, R_F = 1GOhms for series faults).

Step 3: Vector determination step: With R_F as in Step 2, apply vectors (A[15:0] and B[15:0]) at primary inputs of the adder so as to excite the defect path. Compare the adder primary output response with and without defect for the same input vectors.

Step 4: TEST mode simulations: The adder is next put into the TEST mode using the Ctrl1 and Ctrl2 signals (Table 1) depending on the stage in which the defect has been introduced.

Step 5: Delay fault simulation and detection: This is carried out using the vector from Step 3 and mode selections from Step 4. The input clock frequency is kept at 100MHz (10X reduction compared to nominal)

Step 6: Parametric simulations are carried out for varying R_F values resulting in varying delay impact. Input vectors from Step 3 and the "evaluation window" selected in Step 4 are used to determine the extent of the delay fault detectable as a stuck at fault at the primary output(s).

Step 7: The above process steps are repeated for each of the defects one at a time.

4.5 Delay Fault Detection: Simulation Results

As explained in Section 3.5, delay defects introduced in the 16-bit adder are detected using the stage-to-stage testing technique. While detecting these defects, an "evaluation window" is created with a "safety margin". This allows us to budget for fluctuations in the stage delay ($T_{nominal}$) due to process variations. We decided to incorporate the "safety margin" as a fraction of $T_{nominal}$ rather than a fixed value. This is because a longer delay path with larger number of transistors will require larger safety margin. In our study, we used a "safety margin" equal to 25% of the $T_{nominal}$. Such an approach enables us to distinguish between good, slow parts and parts that have delay faults. Table 4 shows the $T_{nominal}$, "safety margin", and "evaluation window" for stage-to-stage testing of the adder. It should be noted that in the presence of DFT, the minimum detectable delay fault is equal to the "safety margin".

Table 4: $T_{nominal}$, "safety margin", and "evaluation window" for stage-to-stage testing of adder

Stages under TEST	Stage delay	Safety Margin ($0.25T_{nominal}$)	Evaluation Window
Stages 1-2-3	$T_{nominal(1\text{-}2\text{-}3)}$ = 138ps	35ps	173ps
Stages 3-4-5	$T_{nominal(1\text{-}5)}$ = 267ps	67ps	334ps

During this research, 22 delay defects were introduced in the adder. Several of these defects had the same delay impact and resulted in multiple defects mapping to the same delay fault. Thus, the 22 delay defects were mapped to 10 delay faults as shown in Table 5. The table depicts the nature (series=S/parallel=P), location and extent of the delay fault detectable with/without the proposed DFT technique. It is clear from this table, that the proposed DFT technique is able to detect substantially smaller delay faults compared to the non-DFT case. For example, in the case of F3, delay faults greater than 35ps can be detected with the proposed DFT technique while only those delays larger than 362ps can be detected for the non-DFT case.

Table 5: Delay fault simulation results @ 100MHz

Defect No.	Fault No.	Defect Location (Stage No.)	Defect Type	Min. detected Delay Fault (DFT)	Min. detected Delay Fault (no DFT)
1,2,3	F1	1	P	35ps	362ps
4,5,6,7	F2	2	P	35ps	362ps
8	F3	3	P	35ps	362ps
9,10	F4	4	P	67ps	233ps
11	F5	4	S	67ps	233ps
12, 13	F6	5	P	67ps	233ps
14	F7	5	P	67ps	233ps
15,16, 17,18	F8	2	P	35ps	362ps
19,20	F9	5	P	67ps	233ps
21,22	F10	2	S	35ps	362ps

As shown in Table 5, we are able to detect delays greater than 35ps for stages 1-2-3 (Faults: F1, F2, F3, F8, F10) and 67ps (Faults: F4, F5, F6, F7, F9) for stages 3-4-5. In order

to have a fair comparison of the effectiveness of the proposed DFT technique in detecting/diagnosing delay faults, the same defects as those discussed above (F1-F10) were also introduced in the non-DFT adder. The non-DFT adder was tested using a 50% duty cycle, 1GHz clock signal. This results in the creation of an "evaluation window" of 500ps. This means that the minimum delay fault that can be detected for stages 1-2-3 and stages 3-4-5 are given by equations (5) and (6) respectively:

$$T_{non\_DFT(1\text{-}2\text{-}3)} = T_{(50\%\ cycle\ time)} - T_{nominal(1\text{-}2\text{-}3)} = 362ps \quad .....(5)$$
$$T_{non\_DFT(3\text{-}4\text{-}5)} = T_{(50\%\ cycle\ time)} - T_{nominal(1\text{-}5)} = 233ps \quad .....(6)$$

In addition, it should be noted that low-frequency delay fault testing is not trivial for a non-DFT adder. This is because, at a reduced frequency of testing, the "evaluation window" is itself "stretched". If for example, in the TEST mode, a 50% duty cycle 100MHz clock is used, the "evaluation window" is equal to 5ns. Using Eq. (5) and (6) we know that, only those delay faults that are greater than 4.86ns for stages 1-2-3 and 4.73ns for stages 3-4-5 will cause logic failures while all others will go undetected.

Our DFT technique is based on the addition of extra footer transistor(s) and logic to control them in TEST mode. This creates a boundary around the DUT inside which the transistors are delay fault testable while those outside are not. The higher the overall fault coverage, more effective is the DFT technique. The proposed DFT technique was found to achieve delay fault coverage for 96.3% of the transistors in the 16-bit adder (testable transistors = 1625, total transistor count = 1688). In other words, approximately 3.7% of the transistors cannot be delay fault tested using this DFT technique. Most of these transistors are in the test logic itself. Delay fault coverage of these transistors remains the subject of future research.

5. Conclusions

In this paper, a DFT technique for delay fault detection/diagnosis in high performance digital circuits has been proposed. The effectiveness of the DFT technique in reducing test frequency by at least 10X has been demonstrated with the help of a 0.18µm 16 bit CDL adder. Stage-to-stage delay testing concept, resulting in improved delay fault diagnosis has been implemented (35ps vs. 362ps). The delay impact (+8.6%) and fault coverage (96.3%) of the proposed DFT technique have also been quantified for the adder. It is expected that the proposed DFT technique will help reduce testing cost and ensure improved long-term reliability of high performance digital circuits.

Acknowledgment

The authors would like to thank R. Krishnamurthy, Intel Corp. for his discussions and suggestions on high performance adder architecture/design and M. Nummer, University of Waterloo for his insights on low frequency testing.

References

[1] Semiconductor Industry Associations, "International Technology Roadmap for Semiconductors, 2001Edition", 2001.

[2] P. Nigh, W. Needham, K. Butler, P. Maxwell, R. Aitken, and W. Maly "So What is an Optimal Test Mix? A Discussion of the Sematech Methods Experiment", *Proc. of International Test Conference*, pp. 1037-1038, 1997.

[3] W. Needham, C. Prunty, and E.H. Yeoh, "High Volume Microprocessor Test Escapes, An Analysis of Defects Our Tests are Missing", *Proc. of International Test Conference*, pp. 25-34, 1998.

[4] H. Hao, and E.J. Mc Cluskey, "Very-Low Voltage Testing for Weak CMOS Logic ICs", *Proc. of International Test Conference*, pp. 275-284, 1993.

[5] A. Keshavarzi, K. Roy, and C.F. Hawkins, "Intrinsic Leakage in Low Power Deep Submicron CMOS ICs", *Proc. of International Test Conference*, pp. 146-155, 1997.

[6] V. D. Agrawal, T. J. Chakraborty, "High-Performance Circuit Testing with Slow-Speed Tester", *Proc. of International Test Conference*, pp. 302-310, 1995.

[7] M. Shashaani, and M. Sachdev, "A DFT Technique for High Performance Circuit Testing", *Proc. of International Test Conference*, pp. 267-285, 1999.

[8] M. Favalli, P. Olivo, M. Damiani, and B. Ricco, "Novel Design for Testability", *IEEE Journal of State-State Circuits*, pp. 1239-1246, 1990.

[9] K. Raahemifar, and M. Ahamadi, "Design-for-Testability for Detecting Delay Faults in CMOS/ BiCMOS Logic Families", *IEEE Trans. on Circuits and Systems – II: Analog and Digital Signal Processing*, vol. 47, no.-11, pp. 1279-1290, Nov. 2000.

[10] A. Bellaouar and M. I. Elmasry. *Low-Power Digital VLSI Design, Circuits and Systems.* Boston, MA: Kluwer Academic Publishers, 1995.

[11] K. Bernstein, K. Carrig, C. Durham, P. Hansen, D. Hogenmiller, E. Nowak, and N. Rohrer. *High Speed CMOS Design Styles*. Boston, MA: Kluwer Academic Publishers, 1999.

[12] Z. Cheng, L. Wei, and K. Roy, "On Effective I_{DDQ} Testing of Low Voltage CMOS Circuits Using Leakage Control Techniques", *Proc. of the IEEE International Symp. on Quality Electronic Design*, pp. 181-188, 2000.

[13] G. Yee and C. Sechen, "Clock-Delayed Domino for Dynamic Circuit Design", *IEEE Trans. on VLSI Systems*, vol. 8, no.-4, pp. 425-430, Aug. 2000.

[14] C. Nagendra, M. J. Irwin, and R. W. Owens, "Area-Time-Power Tradeoffs in Parallel Adders", *IEEE Trans. on Circuits and Systems – II: Analog and Digital Signal Processing*, vol. 43, no. -10, pp. 689-702, Oct. 1996.

[15] S. Matthew, R. Krishnamurthy, M. Anders, R. Rios, K. Mistry, and K. Soumyanath, "Sub-500ps 64-b ALUs in 0.18µm SOI/Bulk CMOS: Design and Scaling Trends", *IEEE Journal of Solid-State Circuits*, vol. 36, no.-11, pp. 1636-1646, Nov. 2001.

[16] J. Park, H. Ngo, J. Silberman, and S. Dhong, "470ps 64-bit Parallel Adder", *Proc. of the IEEE Symp. on VLSI Circuits*, pp. 192-193, 2000.