JOURNAL OF ELECTRONIC TESTING: Theory and Applications 19, 299–314, 2003

A DFT Technique for Testing High-Speed Circuits with Arbitrarily Slow Testers

MUHAMMAD NUMMER AND MANOJ SACHDEV
Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1
mnummer@vlsi.uwaterloo.ca
msachdev@ece.uwaterloo.ca

Received September 6, 2001; Revised July 31, 2002

Editor: V.D. Agrawal

Abstract. This paper presents a design for testability (DFT) technique for testing high-speed circuits with a low-speed test mode clock. With this technique, the test mode clock frequency can be reduced with virtually no lower limit. Even with the reduced speed requirement on the automatic test equipment (ATE), our method facilitates the test of the rated-speed timing and allows performance binning. A CMOS implementation of the DFT hardware with 50 ps timing accuracy is presented. To demonstrate the effectiveness of the technique we designed a 16-bit, 1.4 GHz pipelined multiplier as a test vehicle. Simulations using a test clock frequency much lower than the rated clock frequency show that delay faults of sizes as small as 50 ps are detected and that the new test technique provides correct performance binning.

Keywords: delay-fault testing, high-performance testing, design for delay testability, built-in self test, controlled-delay flip-flop

1. Introduction

The on-chip clock frequency of high-performance state-of-the-art VLSI CMOS circuits has surpassed 1.5 GHz. It is expected that the speed of such circuits will continue to increase for future technology generations. The 1999 edition of the International Technology Roadmap for Semiconductors (ITRS) expects that the on-chip clock frequency will exceed 3 GHz by year 2005 and 13 GHz by year 2014 [1]. With smaller geometries, higher speeds, and increased interconnects, it is more likely for small imperfections in the fabrication process to cause device failure.

According to the ITRS, most of the technology problems causing yield losses and cost increases are related to the slower growth in ATE's capabilities versus the ever increasing device clock frequency [1]. In the past, accuracy of ATEs used to be 4–5 times higher than the state-of-the-art ICs. This is why it was easy to perform at-speed functional testing. In the last two decades, however, while the clock frequencies of VLSI circuits have improved at an average rate of 30% per year, the tester accuracy has improved only at a rate of 12%. If this trend continues, tester timing accuracy will soon approach the cycle time of high-performance devices making at-speed test almost impossible. Table 1 shows the ITRS expected trends for yield, off-chip device speed, and tester accuracy. It is clear from this data that even long before the tester timing accuracy reaches the cycle time of the devices, yield loss due to insufficient accuracy of the tester will become unacceptably high. As yield for future technology generations becomes a major issue, the importance of performing a test capable of ensuring acceptable quality

Table 1. ITRS Trends in yield, off-chip device speed, and tester accuracy [1].

Year	1999	2001	2003	2005	2008	2011	2014
Yield (%)	87	84	79	73	64	56	50
Off-chip device period (ps)	830	700	580	500	400	340	260
Overall ATE accuracy (ps)	200	160	130	100	100	100	100

levels becomes crucial. In the same context, if future ATEs are not able to keep up with device speed, not only the yield but the out-going quality of these devices will also be greatly affected.

The cost of ATE per pin for high-performance circuits has remained approximately constant for the past 20 years at around $10–12 K. Recently, this value has begun to fall below $8K/pin and is expected to continue to decrease in years to come. Nevertheless, it is expected that the demand for higher speed, greater accuracy, more time sets, and increased vector memory will offset most of the gains seen for reducing ATE cost [1]. According to the ITRS, it may cost more to test a transistor than it costs to manufacture it by 2014.

Due to the slow advances and the high cost of ATE, we might not be able to test future high-performance VLSI circuits. Therefore, it will be essential to design these circuits with DFT/BIST techniques to reduce the reliance on traditional, high-cost, full-feature testers. The requirements of ATEs designed to work with DFT/BIST techniques are much simpler than the traditional testers. In this paper, we propose a DFT technique for testing high-speed circuits with arbitrarily slow testers. This work is an extension for our work published earlier [2]. Testing high-speed circuits with slow testers has several advantages. It provides the capability of detecting the subtle timing failures with relative ease resulting in improved quality. Furthermore, with these techniques the life time of an ATE can span multiple life cycles of a product. As a result, using these techniques to test high-speed circuit is expected to reduce the cost of testing and manufacturing.

The rest of this paper is organized as follows. Section 2 gives a concise review of the techniques used for high-performance circuit testing. Section 3 focuses on the use of controlled-delay flip-flops (CDFFs) to test high-performance circuits with slow testers. In Section 4, we illustrate how to generate clocks used for CDFFs in order to reduce the test mode clock frequency arbitrarily. Section 5 provides design details of the clock generation circuit. An overview of the 16-bit pipelined multiplier used as a test vehicle is given in Section 6. Simulation results for the clock generation circuit, performance binning for different process corners, and delay fault detection are given in Section 7. Section 8 discusses some of the implementation issues associated with the proposed technique.

2. High-Performance Circuit Testing: A Review

Defects in semiconductor devices can cause them to fail either functionally or parametrically. A functional failure occurs when a device performs an incorrect logical operation under nominal operating conditions. On the other hand, a parametric failure may occur if any of the electrical parameters such as voltage, current, capacitance, speed, or gain are out of specifications. Defects resulting in parametric failures are considered to be major quality and reliability threats. The impact of these defects is subtle and often testing techniques such as burn-in, I_{DDQ}, and performance testing are used to uncover them. Although, burn-in and I_{DDQ} testing are effective, their limitations are becoming prominent as we marsh into the deep-submicron regime [3–5, 15]. The cost and the limited accuracy of high-performance testers make it difficult and uneconomic to continue to do performance testing on such very high-speed devices.

The size of a defect determines whether the defect affects the logic functionality of a circuit or not. Normally, smaller defects, which are likely to cause partial shorts or opens, have a higher probability of occurrence. Such defects often cause timing failures without altering the logic functionality of the circuit. A number of recent studies show concerns about new failure mechanisms in scaled geometries that are harder to detect with conventional means. Nigh et al. [6] reported a significantly large number of timing only failures that did not affect the steady-state logic functionality. Similarly, for Intel's manufacturing processes, Needham et al. [7] reported an increasing shift towards soft defects as technology moved from 0.35 to 0.25 μm. These defects do not always cause failures at all temperature and voltage conditions and are considered to be major long term reliability threats.

Agrawal and Chakraborty [8] classified high-speed test methods into indirect and direct techniques. Indirect test methods assume that the delays of all gates on a chip are correlated. This implies that the performance characteristics of a chip can be predicted by testing only a small part whose delays are within the capability of the tester. In direct test methods, the test is applied directly to the functional logic of the device under test (DUT). These techniques are generally preferred over the indirect techniques [9].

There are two approaches to perform direct tests on VLSI circuits. The first approach employs correlation-based testing methods not requiring high-speed test equipment. Keshavarzi et al. [3] reported a strong correlation between I_{DDQ} and the maximum operating frequency of a 32-bit microprocessor. They argued that the two parameters are fundamentally related, as both are functions of the channel length. This information can be used as a means for high performance binning. Another technique uses the correlation between supply voltage and device speed. CMOS digital circuits exhibit an increasingly large switching delay as supply voltage is reduced. Hao and McCluskey [10] suggested the use of very-low-voltage testing as a means for testing weak ICs. Supply voltage reduction causes the delay faults to be more noticeable. Hence, these faults can be detected easily at frequencies much lower than the operating frequency.

In the second approach, special techniques to enhance the capabilities of testers and/or the testability of the DUT are used. One example of these techniques is the multiplexing of tester clock pins in order to extend the clock frequency range of the tester. This is a standard feature offered in most modern digital testers. Other techniques include the introduction of special structures and test signals resulting in lower timing requirements for testers. DFT and BIST techniques fall under this category.

The creation of a low frequency test mode in digital circuits was first introduced by Agrawal and Chakraborty [8]. In their proposal, a quantifiable, externally controlled delay is added such that high-performance testing can be carried out with relatively slow-speed testers. They used a pulse-triggered flip-flop in which a dynamic latch is introduced inside a traditional master-slave flip-flop. The resulting three-latch structure has two modes of operation; normal mode and test mode. In normal mode, the intermediate latch must hold data for most of the clock period while the other two latches remain transparent. In test mode, flip-flop delay can be modulated by changing clock's pulse width. This allows for testing combinational logic and interconnects for delay faults with a lower clock frequency. Although the concept of adding delay in test mode is elegant, this implementation has some important shortcomings as the dynamic latch makes the flip-flop operation sensitive and timing critical. Shashaani and Sachdev proposed the controlled delay flip-flop [9] as an alternative to the pulse-triggered flip-flop. In this technique an additional test mode clock is used to control the delay of the flip-flop. The main advantages of the CDFF over the pulse-triggered flip-flop are the stable operation and improved performance in normal mode. Details of the operation of the CDFF in normal and test modes are given in the following section.

3. CDFF for Testing High-Performance Circuits at Low Speed

Fig. 1 illustrates a gate level implementation of the CDFF. The transfer of data from the master latch to the slave latch is controlled through a control logic and depends on the relative timing of the clock (CLK) and the test clock (TCLK). To illustrate the operation of the CDFF, a simple model of digital VLSI circuits is depicted in Fig. 2(a). In this model, a combinational block is sandwiched between two sequential blocks (registers, flip-flops, ... etc.). In normal mode, TCLK is kept high ensuring normal flip-flop operation (Fig. 2(b)). Under this condition, the normal mode clock period (T_{NM}) is given by:

$$T_{\mathrm{NM}} = t_{\mathrm{prop}} + t_{\mathrm{comb}} + t_{\mathrm{setup}} \quad (1)$$

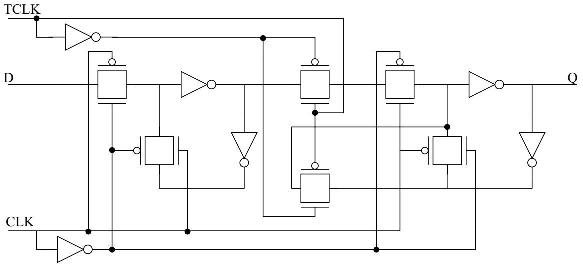


Fig. 1. Controlled delay flip-flop [9].

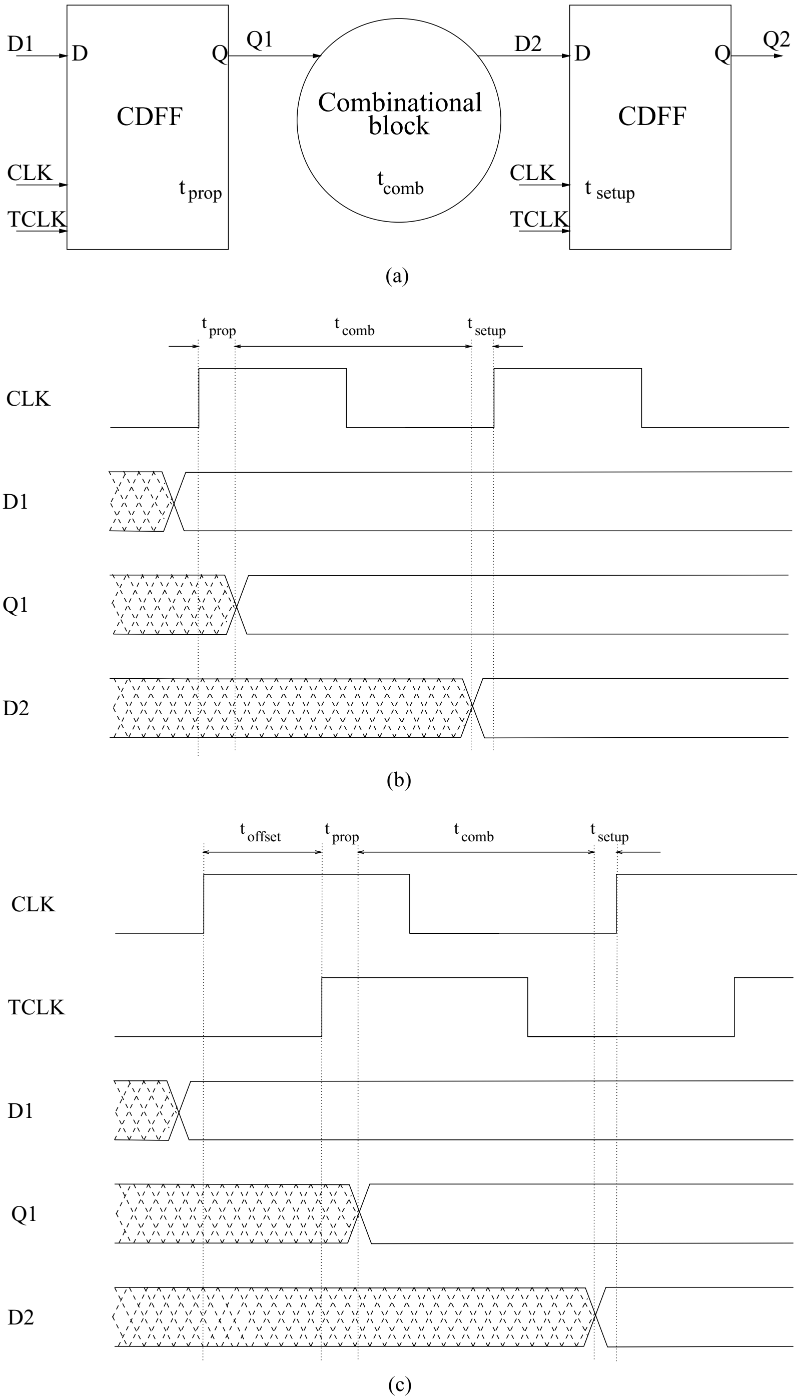


Fig. 2. CDFF operation. (a) Circuit model. (b) Normal mode. (c) Test mode.

where t_{prop} is the propagation delay of the flip-flop, t_{comb} is the time window allowed for the combinational block to evaluate its input, Q_1, and produce the input of the next sequential block, D_2, and t_{setup} is the setup time of the flip-flop.

In test mode, a tester programmed time offset of the clock is used to generate TCLK. Consequently, flip-flop output, Q_1, appears after an additional delay equal to the time offset between the two clocks. This scenario is illustrated in Fig. 2(c). Under this condition, the test mode clock period is given by:

$$T_{\text{TM}} = t_{\text{prop}} + t_{\text{comb}} + t_{\text{setup}} + t_{\text{offset}} \tag{2}$$

where t_{offset} is the time offset between the clock and the test clock. The test mode clock period should be large enough to accommodate all delay terms in Eq. (2). It is clear from this equation that increasing t_{offset} allows the circuit to be tested at a frequency lower than the normal mode frequency. By keeping the time delay between the rising edges of TCLK and CLK equal to T_{NM}, the test mode clock frequency can be reduced while the combinational circuit delays are tested with the same high performance delay margins. Using this technique, the test vectors can also be applied at the same speed as the test mode clock frequency.

4. Using CDFF to Arbitrarily Reduce Test Mode Clock Frequency

In this section, we present a methodology for generating the clock and the test clock for a device using CDFFs in a way that allows the test mode clock frequency to be reduced arbitrarily. This is done through an on-chip clock generation circuit. When generating the clock and the test clock for a circuit using CDFF to improve testability, one has to take into consideration the timing requirements for correct operation. For the CDFF to function properly, the timing of the clock and the test clock must be carefully adjusted to accommodate both the setup time (t_{setup}) and the propagation delay (t_{prop}) of the flip-flop. For the combinational block, it is necessary to have the flexibility to change the value of t_{comb} so as to determine, with reasonable accuracy, the delay through this block and test the circuit for delay faults. This is also important to enable us to do performance binning to know how well does the circuit meet its timing specifications.

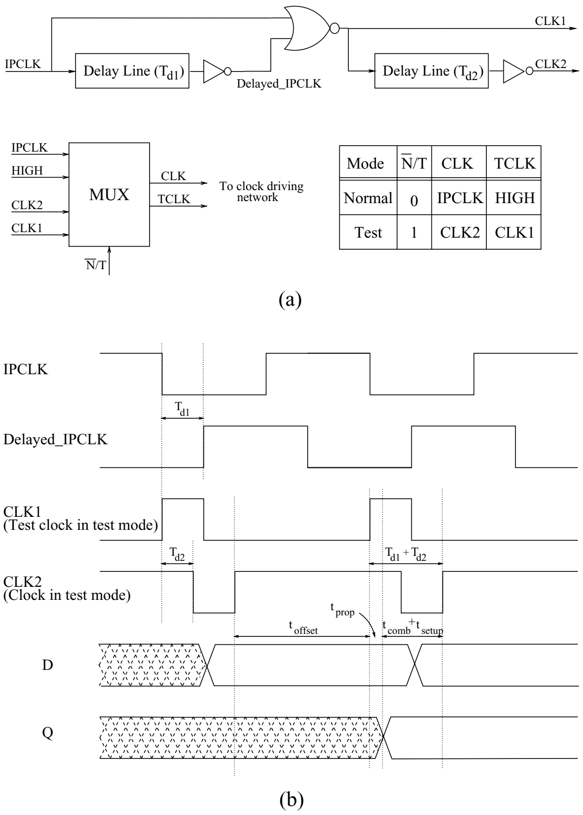
4.1. Reducing Test Mode Clock Frequency

In test mode, reducing clock frequency while maintaining correct timing operation for all parts of the circuit means that, if the clock frequency becomes very low, t_{offset} has to be extremely large. As suggested in [9], the test clock can be generated as a delayed version of the clock with a delay of t_{offset}. The problem with this approach is that a slow tester is a low specification device. It is normally difficult for such a device to provide very large time offset with state-of-the-art timing accuracy. As an alternative, H. Speek et al. [11, 16] suggested the use of two programmable duty-cycle controllers and a programmable delay line to generate the clock and the test clock in test mode. Using their design, reducing the test mode clock frequency to a very small value requires a large delay line to generate the required delay with appropriate timing resolution.

Careful examination of the timing diagram in Fig. 2(c) shows that, instead of generating the test clock by delaying the clock in test mode, the clock can be generated by delaying and inverting the test clock. Generating the clock this way makes t_{offset} (which is the key factor in reducing the test mode clock frequency) independent on the relative timing of the two clocks and allows its value to be increased arbitrarily. Increasing t_{offset} while keeping all the other terms in Eq. (2) unchanged implies a reduction in test mode clock frequency without affecting the time window allowed for the evaluation of the combinational block. It is clear that by doing this, the test mode clock frequency can be reduced with no lower limit.

4.2. Clock and Test Clock Generation

Fig. 3(a) depicts a block diagram of a system for generating the clock and the test clock. The input clock, IPCLK, is a rated frequency signal in normal mode and a low frequency, 50% duty cycle signal in test mode. A multiplexer (MUX) is used to select the mode of operation through the mode select input ($\bar{N}/T$). For normal mode operation ($\bar{N}/T$ = LOW), IPCLK passes through the MUX to the CLK driving network while TCLK is kept high. In test mode ($\bar{N}/T$ = HIGH), two delay lines are used to generate both CLK and TCLK. This is illustrated by the timing diagram in Fig. 3(b). A delay line is used to generate a clock with pulse width T_{d1} (CLK1). This clock is selected by the MUX to be the test clock, TCLK. CLK1 passes through the second delay line (with delay T_{d2}), resulting in CLK2. The



Mode	$\overline{N}$/T	CLK	TCLK
Normal	0	IPCLK	HIGH
Test	1	CLK2	CLK1

Fig. 3. Generating clock and test clock. (a) Block diagram. (b) Timing diagram.

MUX selects CLK2 to be CLK in test mode. Fig. 3(b) also shows the D and Q signals of a CDFF to illustrate the relationships amongst the various timing parameters of the system in Fig. 2(a) on one side and the delays T_{d1} and T_{d2} and the frequency of IPCLK, f, on the other side . These relationships can be expressed by the following two equations.

$$T_{d1} + T_{d2} = t_{\text{prop}} + t_{\text{comb}} + t_{\text{setup}} \quad (3)$$

$$1/f = T_{d1} + T_{d2} + t_{\text{offset}} \quad (4)$$

Assuming fixed f, t_{prop}, and t_{setup}, these equations suggest that a change in either T_{d1}, T_{d2}, or both, leads to an equal change in t_{comb}. This allows the combinational block to be tested for delay faults by changing the time slot allowed for the evaluation of its inputs. For constant T_{d1} and T_{d2}, changing f causes only t_{offset} to change without affecting the operation of neither the flip-flop nor the combinational block. In order to ensure correct flip-flop operation with variable $T_{d1} + T_{d2}$, we characterize the CDFF to find the limiting values of T_{d1} and T_{d2}. For the flip-flop used in our study, simulations show that when T_{d2} falls below 122 ps, the flip-flop ceases to function properly. This is attributed to the fact that T_{d2} has to be large enough to allow the propagation of data from the master to the slave. This value of T_{d2} is equal to the worst case propagation delay of the flip-flop. The limiting value of T_{d1} is 53 ps which is equal to the setup time of the flip-flop.

5. Design of Clock Generation Circuit

The main objective of our design is to have the capability of testing high speed combinational blocks having delays as low as 400 ps with a 50 ps timing accuracy. Moreover, as explained before, we need to provide the ability to do performance binning in order to know how well does the DUT meet its timing specifications. To achieve these objectives, we designed the clock generation circuit such that it allows t_{comb} to be varied from 400 ps to 1150 ps. Referring to Eq. (3) and considering the limiting values of T_{d1} and T_{d2} (t_{setup} and t_{prop} of the CDFF, respectively), the minimum and maximum values of $T_{d1} + T_{d2}$ are found to be 575 ps and 1325 ps, respectively. As stated before, T_{d1}, T_{d2}, or both can be varied to achieve these requirement. It is clear that keeping one of them constant while varying the other should save hardware required for programmable delay lines.

Two factors should be taken into consideration when choosing the values of T_{d1} and T_{d2}. Firstly, due to interconnect delays, the propagation of extremely small pulses might be difficult to achieve. Secondly, it might be difficult to maintain a very small time delay between the two clocks due to clock skew across the chip. Our implementation is designed such that T_{d1} can be varied from 275 ps to 1025 ps, while T_{d2} is held constant at 300 ps. Although a 275 ps pulse width might seem very small, propagating such a small pulse is within the capability of state-of-the-art circuits. Building the TCLK driving network as a replica of the CLK driving network should help minimize the skew between the two clocks.

Two delay lines are used to generate CLK1 and CLK2 (TCLK and CLK in test mode). This is shown in Fig. 4(a). This circuit is designed in 0.18 μm CMOS technology. Fig. 4(b) shows the signals at different points in the circuit when T_{d1} is equal to 275 ps.

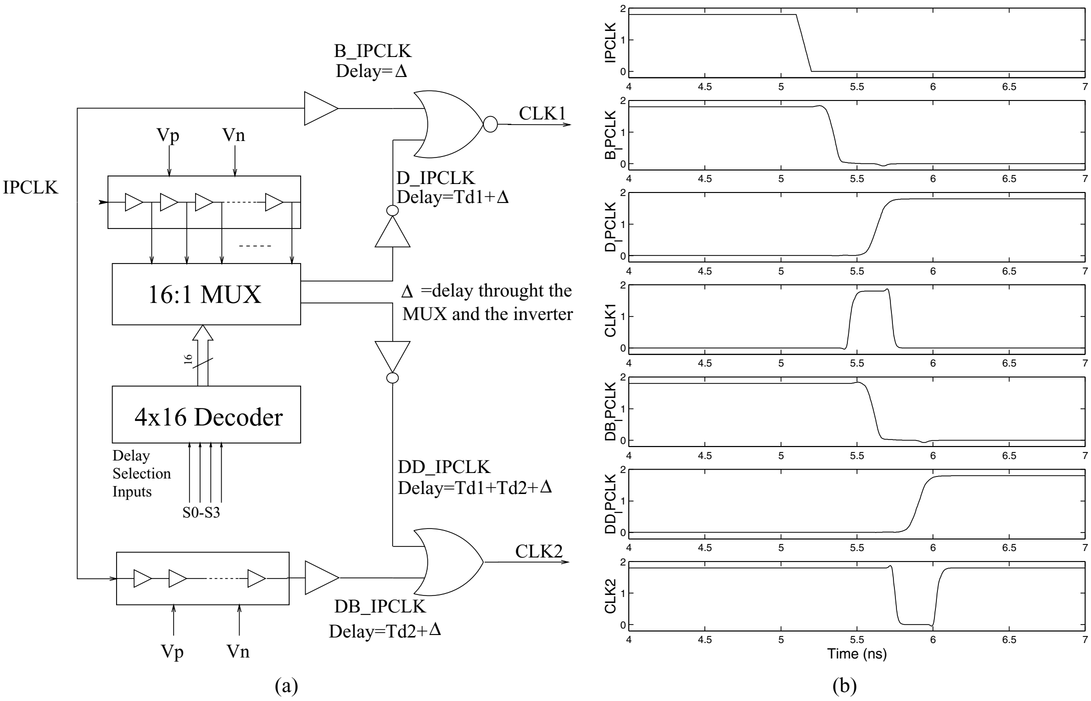


Fig. 4. (a) Circuit used to generate CLK1 and CLK2. (b) Signals at different points for $T_{d1} = 275$ ps.

5.1. Delay Element

Each delay line consists of a chain of delay elements each having a delay of 50 ps. The design of the delay element is very crucial to ensure accurate delays regardless of process, temperature, and supply voltage variations. The delay element used in our design is shown in Fig. 5. It consists of two inverters with current control transistors M1 and M6. Referring to Fig. 4(b), it can be shown that the delay for only the negative edge of IPCLK is critical for correct timing of CLK1 and CLK2. Therefore, the delay element is designed such that the delay is 50 ps for negative going input only. This makes the sizing of transistors M3 and M4 not critical and these two transistors have close to minimum sizes. This is important to minimize the loading of the previous stage and consequently help reduce the delay for the negative going edge of the input.

The delay of the delay element is controlled by controlling the currents through transistors M1 and M6. This is done by two control voltages, V_p and V_n. If V_p and V_n are set to V_{ss} and V_{dd} respectively, currents through M1 and M6 will be maximum resulting in minimum delay for the delay element. Alternatively, if V_p and V_n are set to $V_{dd} - V_{thp}$ and $V_{ss} + V_{thn}$

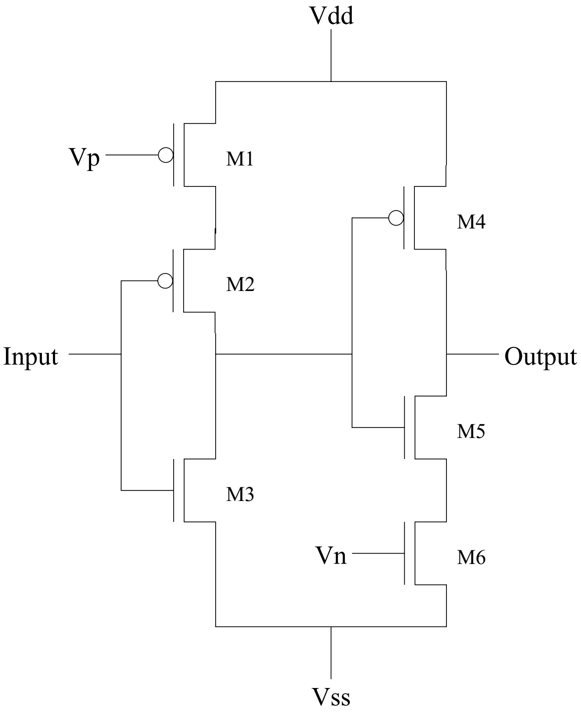


Fig. 5. Schematic diagram of the delay element used for the delay lines.

respectively, where $V_{thp/n}$ is the threshold voltage of the PMOS/NMOS transistor, currents through M1 and M6 will be small resulting in a large delay. The sizes of M1 and M6 should be large enough to provide currents sufficient to achieve the required delay. Area overhead due to large control transistors can be reduced considerably by sharing the current control transistors among multiple delay elements. In our design, only three PMOS (for M1) and three NMOS (for M6) transistors are used for all 32 delay elements used in the delay lines.

Optimum sizing of transistors M2 and M5 is important to provide sufficient currents without excessively loading the previous stage. For $V_n = V_{dd}$ and $V_p = V_{ss}$, transistor sizes are selected such that the delay element has a delay of 50 ps under worst case conditions (slow-NMOS and slow-PMOS transistor models, $T = 100°\text{C}$, and V_{dd} is 10% less than its nominal value).

5.2. Programmable Delay Line

A programmable delay line is used to generate two delayed signals, D_IPCLK and DD_IPCLK, as shown in Fig. 4(a). The delay line consists of a chain of delay elements whose outputs are tapped and fed to a dual 16-1 multiplexer. Two of these signals are selected to be the outputs of the multiplexer. The delay of D_IPCLK (used to generate CLK1) is set to the required value of T_{d1}, while DD_IPCLK (used to generate CLK2) always has a fixed delay of 300 ps (T_{d2}) with respect to D_IPCLK. This means that the delay chain has to be designed to have a minimum delay of 275 ps ($T_{d1_{\min}}$) and a maximum delay of 1325 ps ($T_{d1_{\max}} + T_{d2}$).

Due to the propagation delay of the multiplexer, the delays of its outputs are larger than those generated from the delay chain. In order to minimize the impact it has on the timing of the delay line's outputs, the multiplexer is designed to have minimum delays. On the other hand, the design of the multiplexer's decoder is not timing critical. Hence, it is built with minimum size transistors.

5.3. Buffers, Gates, and Fixed-Delay Delay Line

As shown in Fig. 4(a), IPCLK is buffered to generate B_IPCLK which is NORed with D_IPCLK to generate CLK1. A fixed-delay line is used to generate DB_IPCLK to have a delay of 300 ps (T_{d2}) with respect to IPCLK. CLK2 is generated by ORing DB_IPCLK and DD_IPCLK. In order to balance and minimize their effect, the OR and NOR gates are designed to have equal delays as well as small rise and fall times. Proper sizing of the buffers is important to adjust the minimum delay of the programmable delay line as $T_{d1_{\min}}$ is not a multiple of the delay element's delay ($T_{d1_{\min}}$ is 275 ps while the delay element's delay is 50 ps). Furthermore, these buffers are crucial to compensate for the delays through the multiplexer and the inverters ($\triangle$ in Fig. 4(a)).

6. Test Vehicle: A 16-bit Pipelined Multiplier

In order to verify the benefits of the proposed technique, a 16-bit pipelined multiplier is designed and utilized as a test vehicle. Parallel multiplication is done in three steps. In the first step, the two operands are used to generate partial products whose number depends on the type of encoding algorithm used. The second step is to add the partial products together in a summation network which reduces the partial products to two operands. The product is generated in the final step by adding the resulting two operands using a carry propagate adder.

In our design, no encoding is used to generate the partial products. This results in a number of partial products equal to the size of the multiplier (16). These partial products are added in the summation network using 4-2 compressors as the main component. A 4-2 compressor accepts 4 partial sums and reduces them to two [12]. In order to reduce the 16 partial products to 2 operands, this has to be done in three levels of 4-2 compression. A carry-lookahead adder with conditional sum select [13] is used to generate the product from these two operands.

The multiplier is implemented with five pipelined stages, as shown in Fig. 6. The first stage is used to generate the 16 partial products and reduce them to 8 partial sums after the first level of the summation network. The second and third levels of the summation network are implemented in the second and third stages of the pipeline. The final addition is done in the last two stages.

Pipeline stages are separated by registers to control the timing of data flow through the multiplier. Timing analysis of the first and last stage (SN_L1 and CLA_L2 in Fig. 6) shows that these stages can be tested for delay faults even if the input and output registers (R0 and R5) are implemented using regular flip-flops rather than CDFFs. With this arrangement, R0 is clocked

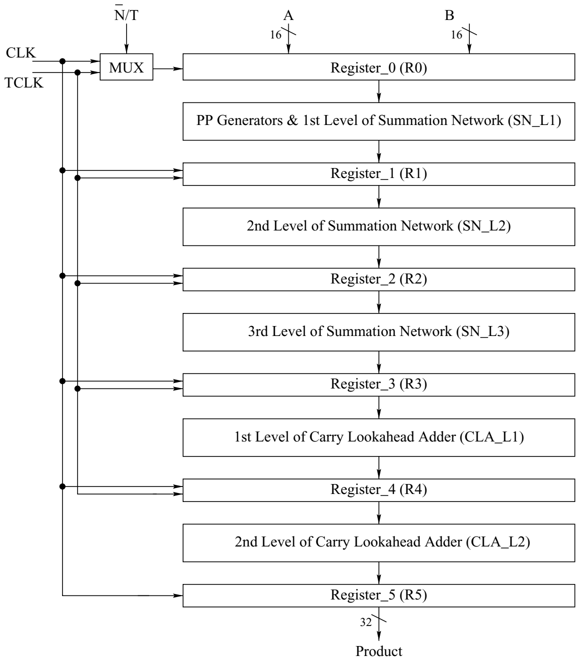

$\bar{N}$/T
A
B
16
16
CLK
TCLK
MUX
Register_0 (R0)
PP Generators & 1st Level of Summation Network (SN_L1)
Register_1 (R1)
2nd Level of Summation Network (SN_L2)
Register_2 (R2)
3rd Level of Summation Network (SN_L3)
Register_3 (R3)
1st Level of Carry Lookahead Adder (CLA_L1)
Register_4 (R4)
2nd Level of Carry Lookahead Adder (CLA_L2)
Register_5 (R5)
32
Product

Fig. 6. Block diagram of the pipelined multiplier used as a test vehicle.

using CLK in normal mode and TCLK in test mode while R5 is clocked using CLK in both modes. All other four registers are built using CDFFs. The timing diagram in Fig. 7 shows the test mode data flow through the pipeline for two consecutive test vectors under these conditions (for simplicity, t_{setup} and t_{prop} of the flip-flops are not shown). On positive edges of TCLK, data propagates from the multiplier input through R0 to the input of SN_L1. Moreover, these edges cause data stored in registers R1, R2, R3, and R4 to propagate to the input of the next stage. Positive edges of CLK cause R1, R2, R3, and R4 to capture and hold their inputs. Same edges cause input of register R5 to propagate directly to multiplier output. Building R0 and R5 using regular flip-flops reduces the hardware associated with CDFFs without degrading the benefits of using CDFFs to enhance the testability of all pipeline stages.

Table 2. Critical path delays through multiplier stages.

Pipeline stage	Critical path delay (ps)
SN_L1	715
SN_L2	690
SN_L3	690
CLA_L1	708
CLA_L2	645

Performance characterization of the multiplier is carried out in order to find its maximum operating frequency and the critical path through each stage of the pipeline. These results are shown in Table 2. The delays shown include the propagation delay of the register feeding the stage and the setup time of the register accepting the output of the stage. As shown in Table 2, the first stage (SN_L1) has the largest delay and the operating frequency of the multiplier is determined by this stage. This delay is equal to 715 ps which translates to a maximum operating frequency of 1.4 GHz.

7. Simulation Results

Several simulations are carried out to test our design and to verify the possibility of reducing test mode clock

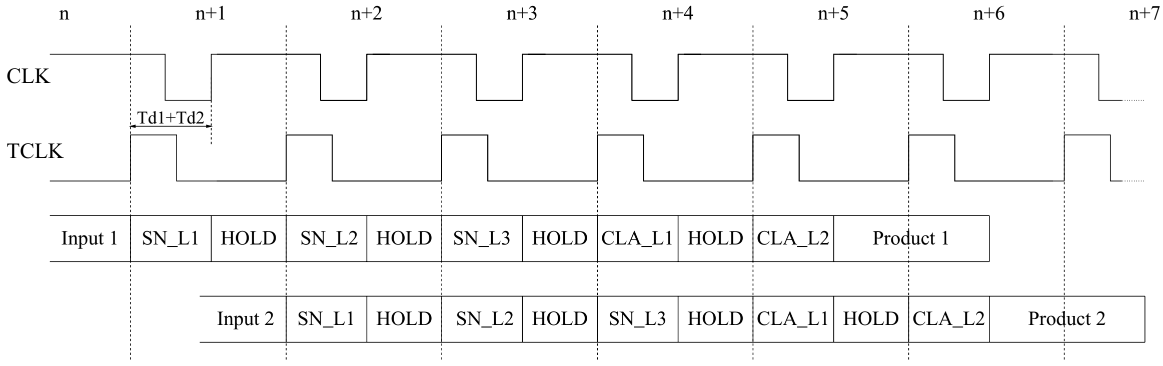


Fig. 7. Data flow through all pipeline stages of the multiplier in test mode.

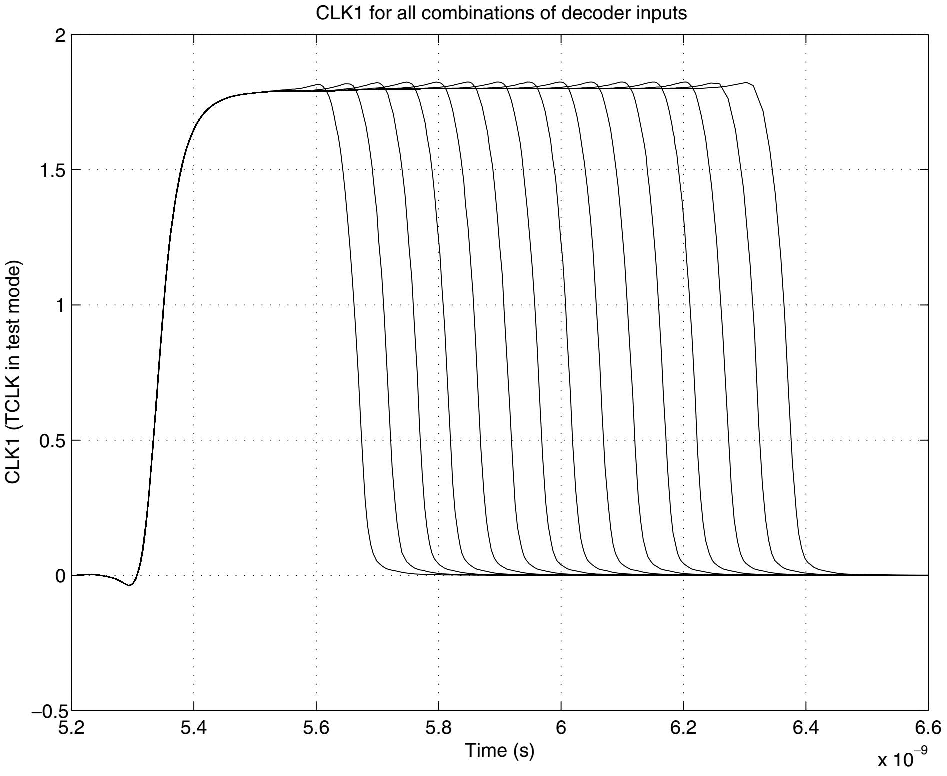


Fig. 8. CLK1 for all possible values of S0–S3.

frequency. Performance binning and delay fault simulations are done on the test vehicle to verify the capability of the technique to predict the maximum operating frequency of the DUT and to detect delay faults at frequencies much lower than the operating frequency.

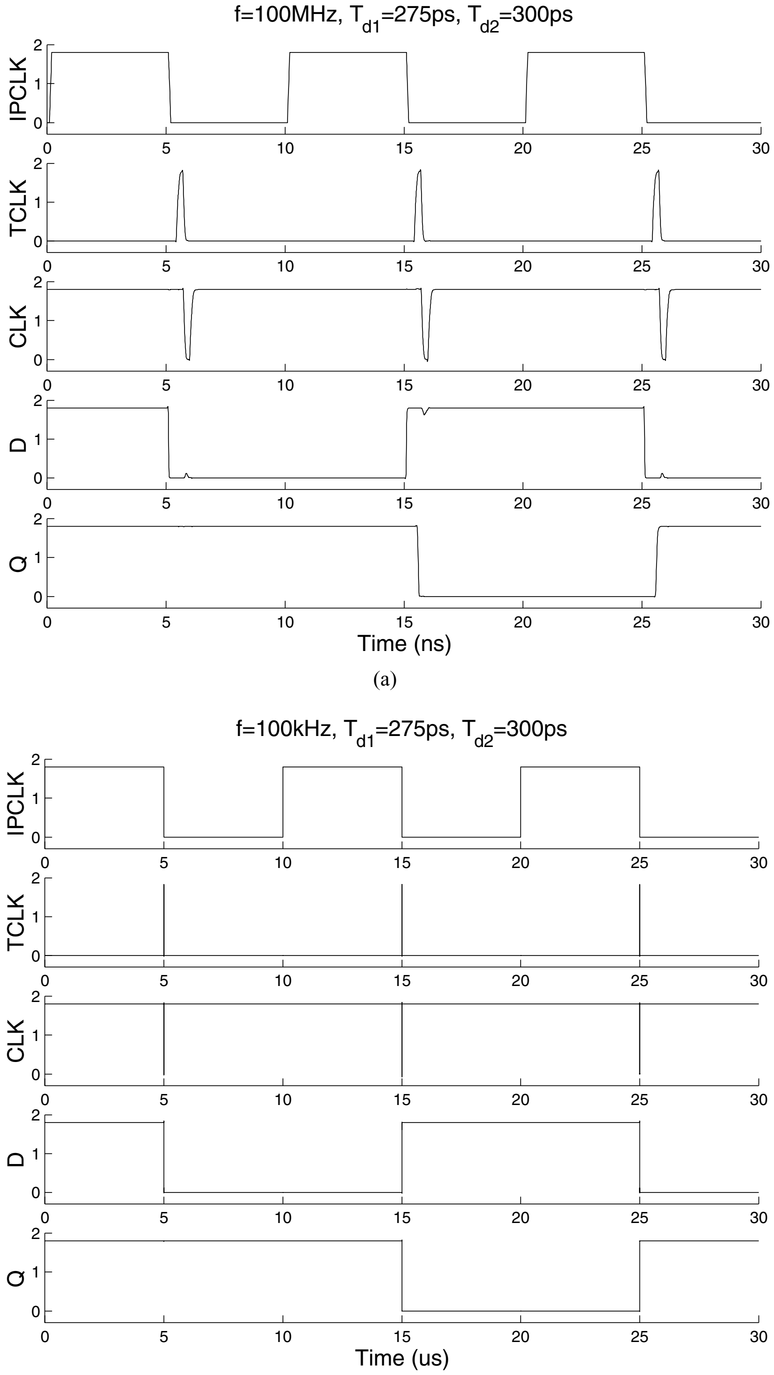
7.1. Clock Generation Circuit

Fig. 8 shows CLK1 for all possible combinations of multiplexer inputs. Simulations show that our design is capable of accurately controlling the value of T_{d1} to achieve the required design goals. The maximum deviation in the delays of the programmable delay line is ±15.8% which is acceptable given the small delay values. For all values of T_{d1}, simulations show that T_{d2} is always constant at 298 ps.

Signals from the clock generation circuit are used to test the CDFF at a wide range of frequencies. These results are shown in Fig. 9 for $T_{d1} = 275$ ps and at frequencies of 100 MHz and 100 kHz. These results show that any increase in the period of IPCLK is reflected as an equal increase in the CLK-Q delay of the flip-flop. This means that reducing the test mode clock frequency affects only the CLK-Q delay of the CDFF and has no effect on the time window allowed for the evaluation of the combinational block. These results show that generating CLK and TCLK using the proposed technique allows the test mode clock frequency to be reduced arbitrarily without degrading the testability of the DUT.

7.2. Performance Binning

The proposed technique can be used to do performance binning of high-speed circuits. A simple algorithm for this task is shown in Fig. 10. As discussed before, binning can be done by changing the value of $T_{d1} + T_{d2}$. In this algorithm $T_{d1} + T_{d2}$ is varied between a minimum and a maximum value with step $(T_{d1} + T_{d2})_{\text{step}}$. DUTs are segregated in a number of bins with numbers ranging from $i_{\min}$ to $i_{\max} + 1$, where $i_{\max}$ is the number of steps between $(T_{d1} + T_{d2})_{\min}$ and $(T_{d1} + T_{d2})_{\max}$. For a given DUT, the test is first applied using the maximum value of $T_{d1} + T_{d2}$. If the device fails the test under this condition, it is placed in bin number $i_{\max} + 1$, which



(a)

(b)

Fig. 9. CDFF simulation results for $T_{d1} = 275$ ps. (a) $f = 100$ MHz. (b) $f = 100$ kHz.

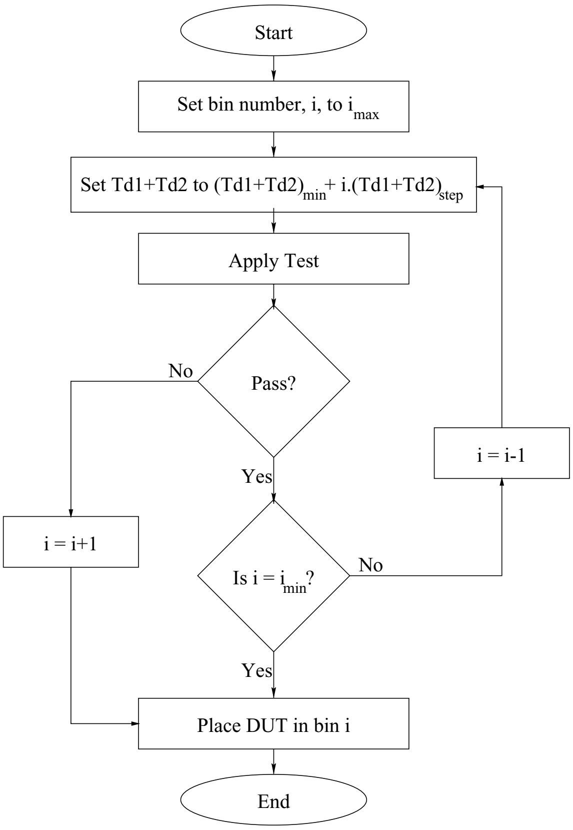


Fig. 10. Algorithm for performance binning using the proposed technique.

indicates that the circuit doesn't work at the low end of frequency range covered by the test. Otherwise, the test is repeated with a smaller value of $T_{d1} + T_{d2}$. This process continues until the DUT fails the test or the minimum value of $T_{d1} + T_{d2}$ is reached. A device passing the test with minimum value of $T_{d1} + T_{d2}$ is placed in bin $i_{\min}$, which indicates that the DUT works at the high end of frequency range. The circuit might work at still higher frequencies. The maximum operating frequency of devices in bin $i_{\min}$ can be found through a clock generation circuit capable of generating even smaller values of $T_{d1} + T_{d2}$. This discussion shows that the choice of the minimum and maximum values of $T_{d1} + T_{d2}$ is critical. These values determine the range of operating frequencies which can be covered when doing performance binning using this technique.

As an example of applying the algorithm in Fig. 10, performance binning is performed on the pipelined multiplier using CLK and TCLK from the clock generation network. Hence, the values of $i_{\min}$ and $i_{\max}$ are 0 and 15 respectively. Simulations are carried out under fault-free conditions. Performance binning simulations are first done using typical process models. These simulations are then repeated for all process corners. As stated in Section 6, for typical process models, SN_L1 has the maximum critical path delay among all stages of the pipeline. For simplicity, we assume that this remains the case for all process corners. Hence, binning is carried out using test vectors that activate the critical path through SN_L1.

Table 3. Performance binning results for various process models.

Process models	T_{d1} (ps)	$T_{d1} + T_{d2}$ (ps)	Bin #
Typical	425	725	3
Fast NMOS, fast PMOS	275	575	0
Fast NMOS, slow PMOS	425	725	3
Slow NMOS, fast PMOS	425	725	3
Slow NMOS, slow PMOS	625	925	7

Performance binning results are given in Table 3. The value of T_{d1} given in the table is the minimum value allowing the multiplier to produce correct output. For typical process models, these results show that the minimum value of $T_{d1} + T_{d2}$ necessary for correct operation is 725 ps. This value is slightly higher than the 715 ps given in Table 2. This is attributed to the finite resolution of the clock generation circuit as it can only predict the performance to the closest 50 ps. Table 3 also shows that for fast NMOS and fast PMOS models, the multiplier functions properly even for the minimum value of T_{d1}.

7.3. Delay Fault Simulation

Delay fault simulations are carried out using typical process models. Delay faults are inserted in the test vehicle one at a time using buffers. Using active elements to generate extra delays ensures the existence of a predetermined delay fault without degrading the quality (rise and fall times) of the delayed signals. Using performance binning results for typical process models given in Table 3, T_{d1} is always set to 425 ps. Fig. 11 shows the algorithm used for fault simulation. Delay faults are inserted in 50 ps increments until a value causing malfunction of the DUT is reached. Delay faults are inserted in the critical path as well as one quasi-critical path in all stages of the pipeline except SN_L3. This is because of the similarity between this stage

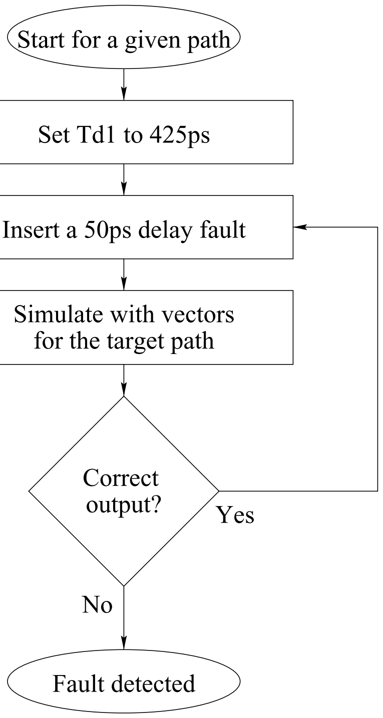


Fig. 11. Algorithm for delay fault simulation.

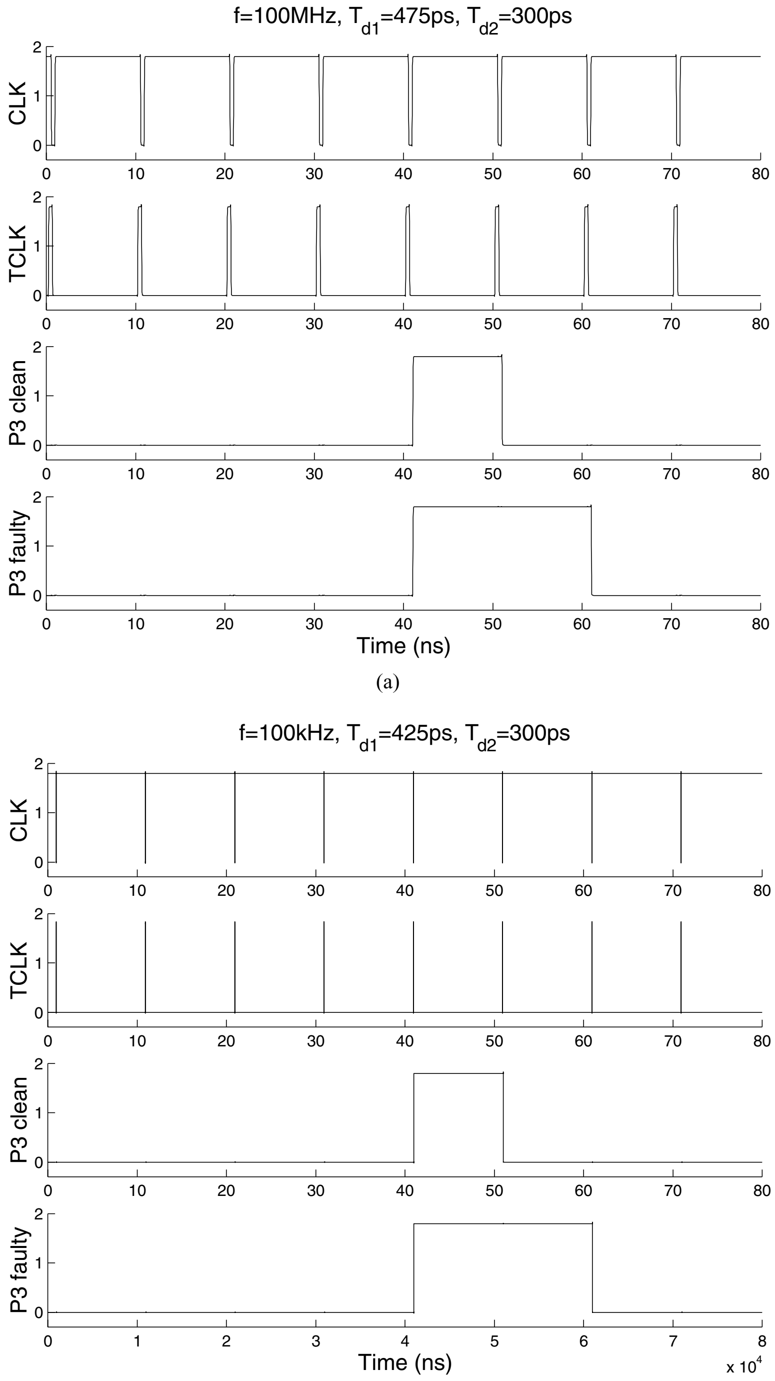
and SN_L2 in terms of their structure and critical path characteristics.

For every target path, two vectors are used to test the circuit. The first vector initializes the DUT while the second vector activates the faulty path of the stage to be tested. Initialization vectors are chosen such that they always result in correct output even under faulty conditions. For correct multiplier operation, activation vectors should result in correct output with the sixth rising edge of CLK (refer to Fig. 7). For a faulty circuit, failure to get the correct output at this edge of CLK means that we are able to detect the fault. The test is done at two frequencies; 100 MHz and 100 kHz. Fig. 12(a) and (b) show examples of test results at these two frequencies for a 50 ps delay fault in the critical path of SN_L1. These graphs show CLK, TCLK and P3 for fault-free and faulty SN_L1 operation (P3 is the only bit of the product affected by the delay fault in the critical path of SN_L1). As shown, we are able to detect the delay fault at both frequencies. Similar results are obtained for all other stages.

Table 4 shows delay fault simulation results for all paths tested in our study. The left half of the table gives the delays of the different paths and the amount of delay fault that has to be inserted in the target path in order to cause the DUT to malfunction. The right half of the table gives the test vectors used for each path as well as the fault-free and faulty product of the multiplier. As is apparent from column 3 and 4, the extent of delay fault that goes undetected is a function of the delay (slack) in a given path. A path with smaller delay will have a larger undetectable delay fault. For example, in the case of path # 6, the path delay is 460 ps. It will take a delay fault of 300 ps to cause a timing failure. On the other hand, for path # 1, it will take a delay fault

Table 4. Delay fault simulation results.

					Input		Product	
Path #	Pipeline stage	Path delay (ps)	Delay fault (ps)	Vector type	A	B	Fault-free	Faulty
1	SN_L1	715	50	Init.	0002	FFFF	0001 FFFE	0001 FFFE
				Activ.	0000	FFFF	0000 0000	0000 0008
2	SN_L1	670	100	Init.	000C	FFFF	000B FFF4	000B FFF4
				Activ.	0008	FFFF	0007 FFF8	0007 FFF0
3	SN_L2	690	50	Init.	0001	FFFF	0000 FFFF	0000 FFFF
				Activ.	0000	FFFF	0000 0000	0000 0040
4	SN_L2	665	100	Init.	0070	FFFF	006F FF90	006F FF90
				Activ.	0040	FFFF	003F FFC0	006F FF80
5	CLA_L1	708	50	Init.	0000	FFFF	0000 0000	0000 0000
				Activ.	FFFF	FFFF	FFFE 0001	FFFD 0001
6	CLA_L1	460	300	Init.	FFFF	FFFF	FFFE 0001	FFFE 0001
				Activ.	0000	FFFF	0000 0000	0001 0000
7	CLA_L2	645	100	Init.	0000	FFFF	0000 0000	0000 0000
				Activ.	FFF0	FFFF	FFD0 0200	FFC0 0200
8	CLA_L2	450	300	Init.	FFF0	FFFF	FFD0 0200	FFD0 0200
				Activ.	0000	FFFF	0000 0000	0010 0000



(a)

(b)

Fig. 12. Fault simulation for the critical path of SN_L1 (a) $f = 100\,\text{MHz}$. (b) $f = 100\,\text{kHz}$.

of only 50 ps to cause the timing failure. In general, delay fault detection is dependent on the target path delay. Most of the delay fault testing techniques have similar limitations. Balancing path delays is the most commonly used method to alleviate this problem.

8. Discussion and Implementation Issues

Simulations results presented in Section 7 demonstrate the effectiveness of the proposed technique in detecting delay faults in high performance digital circuits using a slow speed tester. Delay faults constitute one of the main sources of problems in high-performance circuits. Nevertheless, other defect mechanisms/ phenomena also have an impact on circuit operation in high-frequency. Among these phenomena are crosstalk and signal integrity loss. In general, crosstalk and signal integrity loss have a cause and effect relationship. The critical issue in detecting faults due to crosstalk is the identification of the aggressor and victim nets. Once this information is available, the test can be applied to detect whether there is a fault due to crosstalk. The extent of crosstalk depends on the transition speed and the nature of coupling between the nets. Both of these factors are frequency independent. Since the proposed technique doesn't change the transition speed, it is expected that its crosstalk fault coverage should be similar to that of at-speed tests.

One of the important implementation issues associated with our design is the number of extra pins to be added to the chip. Although, the design requires 7 extra pins for $\bar{N}/T$, V_n, V_p, and S0–S3, there is a potential for reducing these pins. The number of pins associated with the select inputs of the multiplexer can be reduced by storing the data in a shift register. The need for external control signals (V_n and V_p) can be eliminated using techniques like that presented in [14], in which a delay-locked loop (DLL) is used to generate the control signals.

Another important issue is the area and cost overhead due to the routing of two clocks and the use of the CDFF. The area and cost associated with the routing of the two clocks is governed by several factors, such as chip size, number of flip flops, clock frequency, and metal line characteristics. An accurate estimation for this cost is not feasible in a simulation environment. It is also worth noting that the test clock is active in the test mode only. As a result, the test clock has minimal impact on the normal mode power consumption. The reduced test mode clock frequency guarantees low power consumption in test mode. Replacing normal flip-flops with CDFFs has some impact on the chip area, performance, and cost. This impact was studied in detail and presented in [9]. Although the proposed technique adds to the cost of the chip in terms of area and power consumption, this cost can be justified as the design is expected to reduce the cost of testing and manufacturing.

9. Conclusion

We have presented a DFT technique for testing high-speed circuits with little requirements on ATEs. The technique uses CDFF to control the delay of the DUT in order to facilitate its testing. This technique allows such circuits to be tested with arbitrarily low-frequency, 50% duty cycle input clock.

Circuit used to generate CLK and TCLK in test mode has been presented along with various design and implementation issues. Simulation results show that using this design, pulse widths as small as 275 ps for TCLK can be efficiently achieved. The accuracy of the design is 50 ps which is high enough given the the technology generation used in our study.

Simulations of the CDFF with clocks from our design show the validity of this methodology. These simulations prove that the test mode clock frequency can be reduced with no lower limit while keeping the time window allowed for DUT evaluation constant.

We tested the proposed technique with a 16-bit pipelined multiplier as a test vehicle. Testing results show that performance binning of the DUT can be carried out using a simple algorithm. Fault simulation results verify the ability of the technique to detect delay faults using a clock frequency much lower than the rated frequency.

Acknowledgments

The authors gratefully acknowledge support of Communications and Information Technology Ontario (CITO) operating grant for this work.

References

1. Semiconductor Industry Association, "International Technology Roadmap for Semiconductor, 1999 Edition," 1999.
2. M. Nummer and M. Sachdev, "A Methodology for Testing High-Performance Circuits at Arbitrarily Low Test Frequency," in *19th IEEE VLSI Test Symposium*, April 2001, pp. 68–74.

3. A. Keshavarzi, K. Roy, and C.F. Hawkins, "Intrinsic Leakage in Low Power Deep Submicron CMOS ICs," in *Proc. of International Test Conference*, 1997, pp. 146–155.
4. M. Sachdev, "Deep Sub-Micron I_{DDQ} Testing: Issues and Solutions," in *Proc. of European Design and Test Conference*, 1997, pp. 271–278.
5. T.W. Williams et al., "I_{DDQ} Test: Sensitivity Analysis of Scaling," in *Proc. of International Test Conference*, 1996, pp. 786–792.
6. P. Nigh et al., "So What is an Optimal Test Mix? A Discussion of The SEMATECH Methods Experiment," in *Proc. of International Test Conference*, 1997, pp. 1037–1038.
7. W. Needham, C. Prunty, and E.H. Yeoh, "High Volume Microprocessor Test Escapes, An Analysis of Defects Our Tests are Missing," in *Proc. of International Test Conference*, 1998, pp. 25–34.
8. V.D. Agrawal and T.J. Chakraborty, "High-Performance Circuit Testing with Slow-Speed Testers," in *Proc. of International Test Conference*, 1995, pp. 302–310.
9. M. Shashaani and M. Sachdev, "A DFT Technique for High-Performance Circuit Testing," in *Proc. of International Test Conference*, 1999, pp. 267–285.
10. H. Hao and E.J. McCluskey, "Very-Low-Voltage Testing for Weak CMOS Logic ICs," in *Proc. of International Test Conference*, 1993, pp. 275–284.
11. H. Speek et al., "Bridging the Test Speed Gap: Design for Delay Testability," in *Proc. of the IEEE European Test Workshop*, 2000, pp. 3–8.
12. M. Mehta, V. Parmar, and E. Swartzlander, Jr., "High-Speed Multiplier Design Using Multi-Input Counter and Compressor Circuits," in *Proc. of the 10th IEEE Symposium on Computer Arithmetic*, 1991, pp. 43–50.
13. N. Ohkubo et al., "A 4.4 ns CMOS 54 × 54-b Multiplier Using Pass-Transistor Multiplexer," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 3, pp. 251–257, March 1995.
14. G. Moyer et al., "The Delay Vernier Pattern Generation Technique," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 4, pp. 551–562, April 1997.
15. P. Nigh and A. Gattiker, "Test Method Evaluation Experiments and Data," in *Proc. of International Test Conference*, 2000, pp. 454–463.
16. H. Speek et al., "A Low-Speed BIST Framework for High-Performance Circuit Testing," in *Proc. of the 18th IEEE VLSI Test Symposium*, 2000, pp. 349–355.

Muhammad Nummer received the B.Sc. degree in Electrical Engineering from Zagazig University, Egypt in 1996, and the M.A.Sc degree in Electrical and Computer Engineering from University of Waterloo, Canada in 2001. Currently, he is a Ph.D. candidate at the Department of Electrical and Computer Engineering, University of Waterloo. His research interests include digital VLSI system design and test, design-for-testability, and high-performance circuit testing with slow testers.

Manoj Sachdev is a professor in electrical and computer engineering department at University of Waterloo, Canada. His research interests include low power and high performance digital circuit design, mixed-signal circuit design, test and manufacturing issues of integrated circuits. He has written a book, two book chapters on testing and has published significantly in conferences and journals. He received the best paper award for his paper in European Design and Test Conference, 1997 and an honorable mention award for his paper in International Test Conference, 1998. He holds more than 10 granted and several pending US patents in the area of VLSI design and test. He is a senior member of IEEE.

He received his B.E. degree (with Honors) in electronics and communication engineering from University of Roorkee (India), and Ph.D. from Brunel University (UK). He was with Semiconductor Complex Limited, Chandigarh (India) from 1984 till 1989 where he designed CMOS Integrated Circuits. From 1989 till 1992, he worked in the ASIC division of SGS-Thomson at Agrate (Milan). In 1992, he joined Philips Research Laboratories, Eindhoven, where he researched on various aspects of VLSI testing and manufacturing.