A Design-for-Testability Technique for Detecting Delay faults in Logic Circuits

K. Raahemifar
Electrical Engineering Department
University of Windsor
Windsor, Ontario, Canada
N9B 3P4

M. Ahmadi
Electrical Engineering Department
University of Windsor
Windsor, Ontario, Canada
N9B 3P4

Abstract

This paper provides a simulation-based study of the delay fault testing in logic circuits. It is shown that delay testing is necessary in order to achieve a high defect coverage. By detecting delayed time response in a transistor circuit, three types of faults are detected: 1) faults which cause delayed transitions at the output node due to some open defects, 2) faults which cause an intermediate voltage level at the output node, and 3) most stuck-at faults which halt the circuit at '1' or '0'. An on-line checker is presented which enables the concurrent detection of delay faults. Since one checker is used for each output signal, the area overhead is minimal. This technique does not degrade the speed of the circuit under test (CUT). We show that the test circuit is independent of the size of the CUT. Simulation results show that this technique can be adjusted to fit to any design style.

1 INTRODUCTION

Defects in both CMOS and BiCMOS technologies include shorts between the connections[1], open connections[2] , and transistor stuck-on[3]. CMOS circuits can also fail due to circuit degradation (e.g., threshold voltage variations).

It was reported in [12] that in a conventional BiCMOS NAND gate (shown in Figure 1), each injected defect may result in one or more logical fault(s), as explained below. Defects may cause a parametric fault by altering either of the delay time, quiescent supply current, or output voltage magnitude. For example, if a defect turns q_2 off, the output must discharge solely through the NMOS transistors m_3 and m_4. This results in *delay fault*. Some defects create a low-impedance path between V_{DD} and GND, leading to an excessive leakage current (I_{DDQ} fault). Some defects degrade the output voltage, causing an intermediate voltage level or a *soft* logic level (an output which lies in the correct logic level, but has a degraded voltage). The result is a reduced noise margin. The other class is that of functional faults, which includes *stuck-open* (the output becomes floating and the circuit shows a sequential behavior), *truth-table* (the output takes a wrong logic level for certain input combinations), and *stuck-at* faults.

A short may be detectable as a logical fault if one of the nodes overpowers the other logic value and a path of sensitized gates leads from the overpowered node to a primary output, so that the error can be observed. Such defects can manifest themselves as stuck-at "0" or "1" at some nodes including the input and the output nodes. A short can also be detectable as a delay fault if the logic value of one node cannot overpower the value on the other node, but it increases the second node's transition time. As the results of such faults, the output node may be slow to rise or slow to fall. Intermediate voltage on nodes which present a high impedance can be seen as a transition with infinite rising time or falling time. A short may be detectable as an I_{DDQ} fault if the two shorted nodes make a low-impedance path between V_{dd} and GND (or V_{SS}), leading to an excessive leakage current, consequently, an abnormal increase in quiescent power supply current (I_{DDQ}). Although many short defects may create a stuck-at fault, they most frequently result in an I_{DDQ} fault and/or intermediate output. Short defects and their effects on BiCMOS circuits were studied in [4, 11, 12], and a DFT I_{DDQ} testing technique was proposed in [12] for facilitating the detection of these defects.

The fault characterization reported in [7, 8, 10, 12,

[1] Shorts are modeled as a small resistor between the two nodes.

[2] Open-circuits are modeled as a large resistor inserted between the affected node and the node to which it would normally have been connected.

[3] A transistor stuck-on is a transistor that is always in the conducting mode.

0-8186-8409-7/98 $10.00 © 1998 IEEE

14, 15, 17] has shown that most open defects in BiCMOS logic families do not create a stuck-at fault. These defects, however, result in either a delay or a stuck-open fault.

The fault models as well as the fault distributions corresponding to solid short/open defects for a conventional BiCMOS, a BiNMOS, a full-swing BiCMOS, and a CMOS NAND gate are provided in [12]. The notation used in this paper for identifying the short and open defects follows that of [12]: If a fault is caused by a short between two nodes, we refer to it by the first letters of those nodes, followed by the name of the transistor; e.g., a fault caused by a short between the base and emitter nodes of q_2 will be referred to as beq_2, while a short between the drain and source of m_3 will be called dsm_3. A fault caused by an open in a node, will be referred to by the first letter of that node, followed by the name of the transistor; e.g., an open in the collector of q_1 is called cq_1, while one in the gate of m_2 is gm_2.

An important result, concluded from [12], is that most of the short defects in CMOS/BiCMOS circuits cause a stuck-at and/or I_{DDQ} fault, while most of the open defects result in delay or stuck-open faults. The results further reveal that as the redundancy in BiCMOS structure increases, fewer defects can cause a stuck-open fault. For example, in the full-swing structure— which has 100% redundancy— none of the open defects create a stuck-open fault. This is due to the fact that alternative path to the output exists due to the redundancy (full-swing structure is actually a CMOS gate connected in parallel with a conventional BiCMOS gate. A single open defect in this structure is, therefore, masked by the parallel redundancy). In order to detect these defects, delay testing must be performed. Therefore, delay fault testing is quite important in CMOS/BiCMOS logic families, because it is the only means of detecting certain manufacturing defects. This paper deals with delay faults, and presents a design-for-testability (DFT) technique for detecting them. The proposed DFT approach for delay faults allows on-line testing and can easily be compatible with any design style. It is also capable of detecting faults which cause intermediate voltage level at the output of a given circuit. There is no need to modify the given design, and the faults in the added testing circuitry are either detectable or have no effect on the normal operation of the circuit. Moreover, this method lead to smaller hardware overhead. This circuit is applicable to any given block, and is efficient in terms of area and speed.

2 TYPES OF DELAY FAULTS

Some defects do not change the functional behavior of the circuit, and only increase the propagation delay of the circuit beyond the nominal limit. The result is a delay fault, which may manifest as a slow-to-rise or a slow-to-fall transition. Some delay faults may cause cause other faults in transistor circuits and create functional problems for the circuit, see [9]. Thus, delay fault testing is of great importance, specially for high-speed applications.

Delay faults appear as one of the two alternatives shown in Figure 2. Graph "a" shows a fault-free output signal, in a high-to-low transition. The waveform in graph "b" is slow-to-fall. Although it may start its transition as early as the fault-free signal, it requires much more time to reach its final value:i.e., slowed rising or falling transition. The waveform depicted in graph "c" accomplishes its high-to-low transition as fast as the fault-free signal. However, there is a time delay (t_3 - t_1) before it starts the transition. Similar malfunctionings can occur in a low-to-high transition. While the first is due to larger capacitive loading on some nodes, the latter is due to parametric changes of active elements, or the next-stage effect of the first.

In any of the two cases depicted in Figure 2, the faulty output finally reaches the correct value. Therefore, the fault is not detectable by functional testing. However, if the output is observed at an *appropriate* time (at the maximum acceptable normal delay), the fault can be detected. To detect a delay fault, a two-pattern test is applied to generate and propagate signal transitions along the path to be tested. The first input vector initializes the output to a known state. The circuit is allowed to stabilize under this input vector. At time t_1, the second input vector is applied such that the output changes to the opposite logical level in the fault-free circuit. The output is sampled at time t_2, where $(t_2 - t_1)$ corresponds to the maximum acceptable delay for the fault-free operation. In the presence of a delay fault (of each of the two mentioned types), the output has not reached the opposite logical level and the fault can be detected.

3 PREVIOUS WORK

The simplest form of delay testing is to apply pseudo-random patterns to the circuit under test (CUT) at desired speed. Consider the simple combinational circuit in Figure 3 implementing the function $F = a\bar{b} + \bar{b}c + b\bar{c}$. The delay (for a rising transition) of the path from a to F can be tested by the two vectors $V_1 = (a, b, c) = (0, 0, 0)$, and $V_2 = (1, 0, 0)$.

Two-pattern tests have been widely used for testing stuck-open and delay faults in logic circuits. There

are a number of drawbacks, however. This method requires very long test lengths to achieve an acceptable fault coverage [16]. The reason is that faults need to be sensitized through long paths. The other drawback of this method is that tests can be invalidated by delays in other parts of the circuit, hazards, and timing skews [6,13]. A test that detects a delay fault of a certain size does not necessarily detect faults that have longer or shorter delays [5]. As a result, there have been efforts to eliminate the need for them.

One such effort is the design-for-testability technique proposed in [3]. There are two controlling signals, ϕ_1 and $\phi_2 = \bar{\phi}_1$, see Figure 4. The transition edges of these controlling signals announce the end of the valid reading time, t_{sample}, for the circuit signal under the test, S. Signals ϕ_1, ϕ_2, and S are used to sense the coming transition on S, and T_{out} is generated through an intermediate logic circuit. If no delayed transition of the signal, S, occurs at $t > t_{sample}$, T_{out} is "1". If S changes after the transition on ϕ_1 and ϕ_2 due to a delay fault, the extra logic circuit will set T_{out} to zero.

The method has several drawbacks.

- As the authors have pointed out, the synchronization between the circuit signals, and the controlling signals is quite difficult. The circuit signal can be within ϕ_1 and ϕ_2, or it can be outside of transition of controlling signals when the circuit is fault-free.
- As described by the authors, the presented method needs ratioed capacitors.
- The controlling signals (ϕ_1 and ϕ_2) are difficult to generate. Depending on the physical place of delayed signal under testing, there has to be additional hardware to generate the controlling signals.
- The method can detect one type of faults which causes a delayed transition (see Figure 2 (c)).

Another effort is made in [5] to propose a DFT technique to detect delay faults. The authors discuss the importance of observing the output waveform between the samples, instead of only latching the output at the sampling time. Figure 5 shows how a delay test, based on latching the output at specific sampling times, can be invalidated due to dynamic hazard.

Since there is significant information in the output waveform between samples, they propose to perform delay testing by applying test patterns and sampling the output in the conventional manner, together with a circuit that monitors the output waveform between samples. In order to observe any changes after the sampling time, they have proposed the DFT technique shown in Figure 6. This circuit uses the transient switching current in CMOS inverters to detect signal changes. While Φ is low, the bus is kept at V_{DD} by the precharging transistor, so the inverters operate normally. Φ changes to logic high when all the nodes of interest in the CUT have been given enough time to stabilize in the fault-free circuit. Once Φ is high, the bus is left floating. If D_i does not change, the corresponding inverter will draw negligible static current and the bus will remain high. If any D_i changes, the transient current while the inverter changes state will partially discharge the bus, which can be detected with a sense amplifier.

This technique suffers from the following shortcomings:

- As described by the authors, this method is capable of detecting delay faults only in presence of hazards in the CUT response. If all the outputs have single transitions, the delay fault remains undetected. We simulated their proposed circuit and observed that with careful selection of transistors width-to-length ratios, it is possible to detect slow-to-fall signals even for single transition outputs. However, slow-to-rise signals cannot be detected because output M in Figure 6 can be discharged to 0 only if out_i makes a transition from 1 to 0.
- Node M does not go all the way down to 0 V, because PMOS transistors are imperfect switches when passing a 0 (see Chapter 2 in [18]). This is due to the fact that a PMOS transistor with gate node at voltage 0, begins to turn *off* when the output voltage reaches V_{tp} (threshold voltage). Therefore, the output does not go lower than V_{tp}. Our simulations show that M in Figure 6 does not discharge below 1.5 V. This, reduces the noise margin and may cause difficulties in fault detection.

4 PROPOSED DFT TECHNIQUE FOR DETECTING DELAY FAULTS

We propose a DFT technique for detecting delay faults. Our delay fault detector circuit is based on three facts.

- Any transition, rising or falling transitions, at any node of the circuit, intermediate node or output node, is caused by a transition on the primary input. We assume that the input transitions are

introduced without delays for the sake of simplicity.

- The voltage level of the output signal is used to sense the end of the transition. If it is a falling transition, a voltage less than V_t, which is nominally one volt, marks the end of transition. If it is a rising transition, a voltage greater than $V_{dd} - V_t$ is used to mark the end of transition.

- An RC Circuit is used to measure the time response between the output (or node) transition and the input transition.

In the proposed technique, a test circuit is designed to sense the transitions on the input and the output nodes, and the maximum time response is checked by an RC circuit. The technique proposed is schematically illustrated in Figure 7 and Figure 8 shows a diagram of the resulting waveform at the non-grounded node of the capacitor. Here, we check the falling transition on both input and output nodes. The test circuit runs concurrently with the circuit under the test; there is no test-mode operation.

At the beginning, the input and output signals are high, the capacitor is discharged, N_1 and N_2 are on, and P_1 and N_3 are off. As the input signal falls to zero, N_1 turns off. The capacitor starts charging toward V_{dd} through the resistor. At this time, the output signal is still '1', which keeps N_2 on, and N_3 off. When the output signal falls to zero, this transition disables N_2 and turns P_1 on, connecting the gate of N_3 to V_{dd}. This activates N_3 which discharges the capacitor to zero. The capacitor voltage level is compared with a threshold voltage level using a latch. The latch's output shows whether the capacitor is charged over V_t or not. Since the amount of charge on capacitor depends to the time it is charging, capacitor voltage level is an indication of time as well. Therefore, we can measure the delay between the output and input transitions.

If the CUT is fault-free, the time between output and input transitions is less than or equal to the acceptable delay. The maximum voltage level the capacitor reaches, V_{cap}, is less than a threshold voltage, V_t, because the capacitor is discharged while on its way of rising to V_{dd}. Therefore, the latch output, T_{out}, remains zero. If the response time is later than a desired delay, the capacitor would charge to a voltage enough to trigger the latch resulting $T_{out} = 1$. This shows that the time interval between input and output transitions is beyond the desired limit.

The values of the capacitor and the resistor should be chosen based on the desired delay, and the desired threshold voltage for the latch.

In order to evaluate the capability of the circuit in detecting delay faults, we used a conventional BiCMOS NAND gate (Figure 1) as the CUT. HSPICE simulations were done for all defects which cause delay fault.

Simulations show that **all** the defects which result in delay faults can be detected with this technique. We repeated the simulations for other logic families (BiNMOS, full-swing BiCMOS, and CMOS). Simulations show that the proposed circuit is applicable to all mentioned families, and is capable of detecting all delay faults.

5 PERFORMANCE EVALUATION

In order to test the CUT, suitable test patterns are applied to its inputs, and the outputs are monitored by the checker for possible delay faults. If the CUT is recognized as fault-free, there is no need for frequent off-line delay testing, and the CUT will be tested for delay faults concurrently with the normal operation. This is due to the fact that the error signal is created if any of the observed outputs makes a transition after the maximum specified delay. It is not important whether the transition is from low to high, or vice versa. The checker does not need to know the fault-free output voltage level for detecting the delay fault. Therefore, **special test-pattern generation is not required for this method.**

The test circuit proposed has the following characteristics:

- There is no need for generating the sampling signal, ϕ, as proposed in [3], which is dependent to the physical location of the node under the test, and its generation is quite complex.

- The proposed technique can be used concurrently with the normal operation of the circuit to detect delay faults as they occur; i.e., the test circuit works on-line.

- The test circuit does not cause any loading on the circuit under test: i.e., The output voltage levels of the circuit remain unaffected.

- Since only one small-size MOS transistor is added to each desired node, the speed performance of the circuit is not degraded.

- The power dissipation increases slightly. This increase is insignificant for a complex circuit in normal-mode operation. To further minimize the power dissipation due to test circuit, a PMOS transistor is added in series with the resistor. This

transistor is controlled by a test signal. Whenever needed, this transistor is activated, otherwise, it is off. This reduces the static power dissipation.

The important consequence of these points is that the proposed technique can be used *concurrently* with the normal operation of the circuit to detect delay faults as they occur. This eliminates the difficult task of generating robust test patterns for detection of delay faults, and the increase in testability is quite significant, as described previously.

There is an area penalty due to the added devices and due to the circuitry needed for each output node. If we are interested to check the circuit for delay $D = min$maximum delays allowed, the area overhead is about 5 transistors for an output which has the slowest transition. However, if we would like to check different parts of the circuits for delay fault, the complexity of the circuit proposed increases as the number of primary inputs and the nodes under the test increase since each output need to be checked for a specific delay response. However, it is independent of the size of the circuit under the test. The NMOS connected to the output would have to be PMOS to detect the rising signals.

Replacing the resistor in the RC section of the test circuit by an adjustable current source would increase the accuracy of the delay expected.

A general purpose chip can be designed for testing delay faults of any given logic circuit. In general, the test circuit has the same primary inputs as the CUT has. Two extra lines can be used to detect the transition type. For circuits which are more complex in terms of input/output nodes, we recommend the use of extra controlling signals to choose appropriate resistors or capacitors. (A number of resistors and capacitors are connected in parallel or series using these control signals to match with the desired delay and threshold voltage.) This method is applicable for circuits as small as NAND gates, and as large as microprocessors.

6 CONCLUSION

Delay testing is important in high-speed applications. Two-pattern tests are normally used for delay fault detection. However, they require very long test lengths to achieve an acceptable fault coverage, and can also be invalidated by delays in other parts of the circuit.

In this paper, we have proposed a DFT technique for detecting delay faults in transistors circuits (combinational and/or sequential). The operation of the delay fault checker is based on observing the time delay between transitions happening inside the circuit, instead of the normally practiced observation of the outputs *at* a specific sampling time. The proposed checker is applicable to all logic families, and is capable of on-line detection of delay faults.

The checker does not need any extra control signal, except the test signal. Applying the proposed DFT technique has the following effects on the normal behavior of the circuit under test: since only one small-size MOS transistor is added to each output, the speed performance of the circuit is not degraded. The output voltage level remains unaffected. There is an area overhead, because 5 transistors are required to implement the checker for one output. The area overhead is negligible compared to that of proposed by [3]. These overheads are well justified by the great increase in the testability of the circuit. The static power dissipation can be reduced efficiently using an additional transistor. The test circuit functions independently of the size of the circuit under test, from small gates to large logic blocks such as microprocessors.

References

[1] A. R. Alvarez, *BiCMOS Technology and Applications, second edition.* Kluwer Academic Publishers, Norwell, MA, 1993.

[2] S. H. Embabi, A. Bellaouar, and M. I. Elmasry, *Digital BiCMOS Integrated Circuit Design.* Kluwer Academic Publishers, Norwell, MA, 1993.

[3] M. Favalli, P. Olivo, M. Damiani, and B. Ricco, "Novel Design for testability Schemes for CMOS IC's," *IEEE Journal of Solid-State Circuits*, vol. 25, pp. 1239–1245, October 1990.

[4] M. Favalli *et al.*, "Analysis of Steady State Detection of Resistive Bridging Faults in BiCMOS Digital ICs," *Proc. IEEE International Test Conference*, pp. 466–475, 1992.

[5] P. Franco and E. J. McCluskey, " Delay Testing of Digital Circuits by Output Waveform Analysis," *Proc. IEEE International Test Conference*, pp. 798–807, 1991.

[6] A. P. Jayasumana, Y. K. Malaiya, and R. Rajsuman, "Design of CMOS Circuits for Stuck-open Fault Testability," *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 58–61, Jan. 1991.

[7] M. E. Levitt, K. Roy, and J. A. Abraham, " BiCMOS Fault Models: Is Stuck-at Adequate?, " *Proc. IEEE International Conference on Computer Design*, pp. 294 - 297, 1990.

[8] S. C. Ma and E. J. McCluskey, " Non-Conventional Faults in BiCMOS Digital Circuits," *Proc. IEEE International Test Conference*, pp. 882 - 891, Oct. 1992.

[9] S. C. Ma and E. J. McCluskey, " Open Faults in BiCMOS Gates," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, pp. 567–575, May 1995.

[10] S. M. Menon, Y. K. Malaiya, and A. P. Jayasumana, "Behavior of Faulty Single BJT BiCMOS Logic Gates," *Proc. IEEE VLSI Test Symposium*, pp. 315–320, Apr. 1992.

[11] S. M. Menon, Y. K. Malaiya, and A. P. Jayasumana, "Bridging Faults in BiCMOS Circuits," *Proc. of the 5th NASA Symposium on VLSI Design*, pp. 7.1.1–7.1.10, Nov. 1993.

[12] K. Raahemifar, S. Hessabi, and M. I. Elmasry, "A Design-for-Testability Technique for Shorts and Bridging Faults in BiCMOS Logic Families," in *1995 Canadian Conference on Electrical and Computer Engineering*, vol. 1, pp. 221–224, Montreal, Canada, Sep. 5–8, 95.

[13] S. M. Reddy and M. K. Reddy, "Testable Realizations for FET Stuck-open Faults in CMOS Combinational Logic Circuits," *IEEE transactions on Computers*, vol. c-35, pp. 742–754, Aug. 1986.

[14] K. Roy, M. E. Levitt, and J. A. Abraham, " Test Considerations for BiCMOS Logic Families," *Proc. IEEE Custom Integrated Circuit Conference*, pp. 17.2.1 - 17.2.4, 1991.

[15] A. E. Salama and M. I. Elmasry, " Fault Characterization, Testing Considerations, and Design for Testability of BiCMOS Logic Circuits," *IEEE Journal of Solid-State Circuits*, vol. 27, pp. 944–947, June 1992.

[16] J. Savir and W. H. McAnney, " Random Pattern Testability of Delay Faults," *IEEE Transactions on Computers*, vol. 37, pp. 291–300, Mar. 1988.

[17] B. E. Stewart, D. Al-Khalili, and C. Rozon, " Defect Modelling and Testability Analysis of BiCMOS Circuits," *Canadian J. Elect. & Comp. Eng.*, vol. 16, no. 4, pp. 148 - 153, 1991.

[18] N. H. E. Weste and K. Eshraghian, *Principles of CMOS Design: A Systems Perspective, second edition*. Addison-Wesley Publishing Company, 1994.

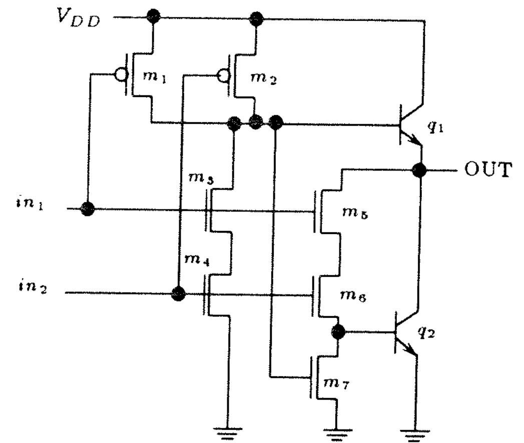

Figure 1: Conventional BiCMOS NAND gate

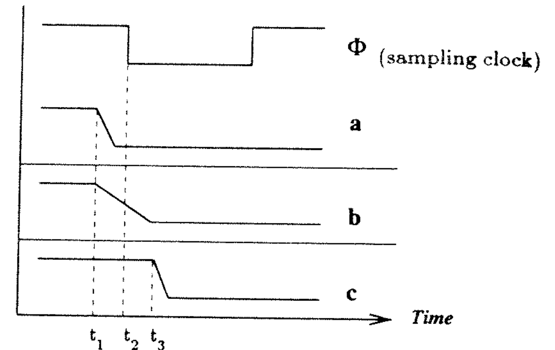

Figure 2: a) Fault-free output, b,c) two different appearances of delay faults. Φ is the sampling clock for detecting delay faults.

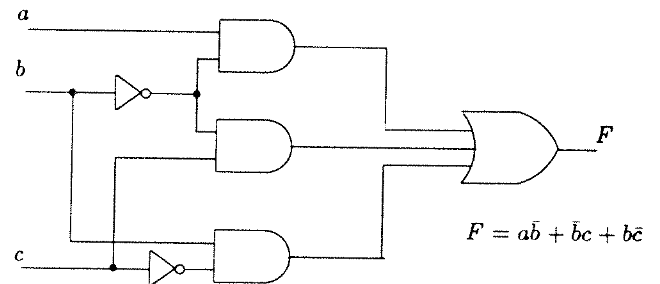

Figure 3: Two-pattern test for delay faults in logic circuits.

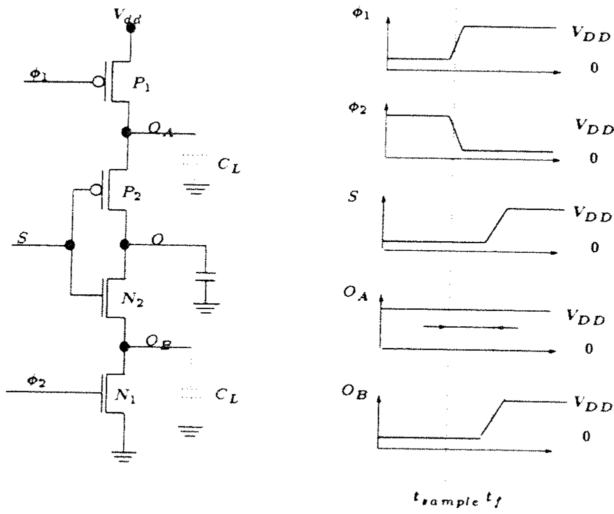

Figure 4: (a) Delay testing circuit, (b) Schematic timing diagram as presented in [3].

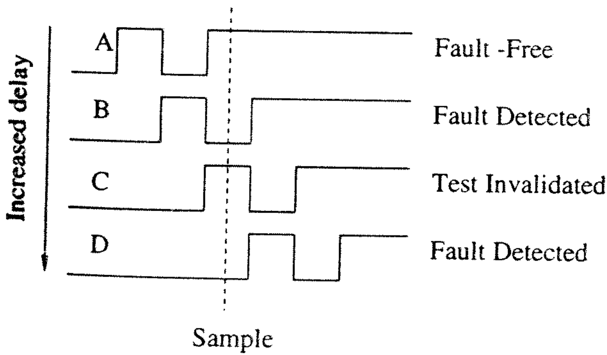

Figure 5: Delay test invalidation by dynamic hazard [5]

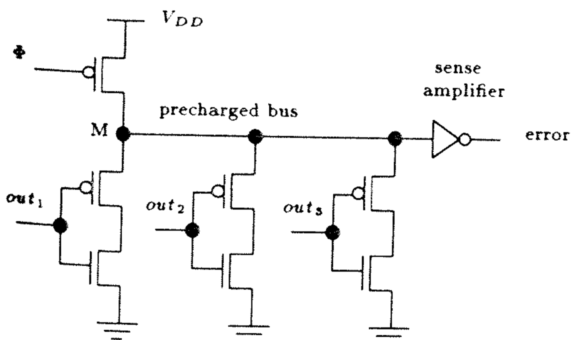

Figure 6: DFT technique proposed in [5] for post-sampling analysis

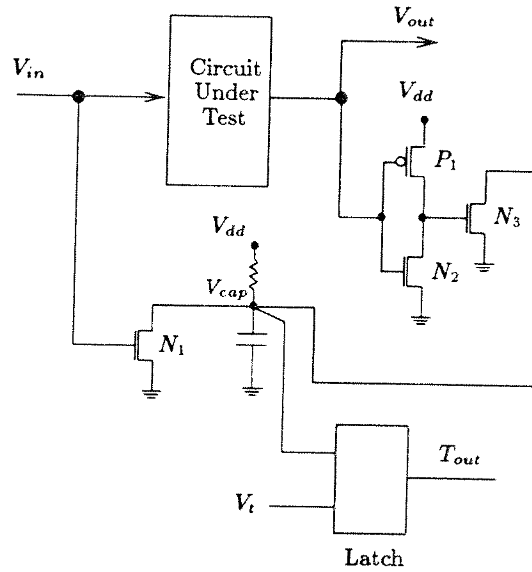

Figure 7: Proposed Delay fault DFT technique: Delay testing circuit.

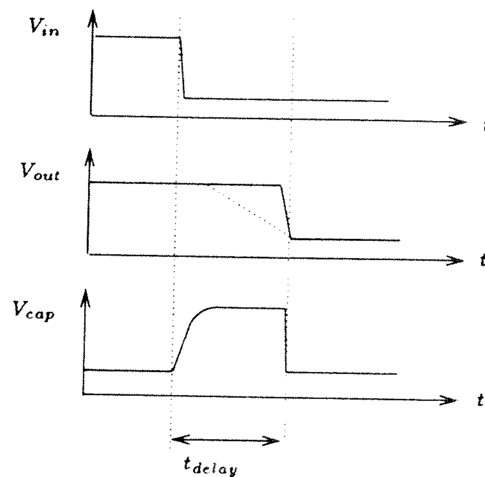

Figure 8: Proposed Delay fault DFT technique: Schematic timing diagram.