# Department of Computer Science

## Detection of Feed-Through Faults in CMOS Storage Elements

Waleed K. Al-Assadi, Yashwant K. Malaiya, and Anura P. Jayasumana

Technical Report CS-93-119

September 8, 1993

# Colorado State University

# Detection of Feed-Through Faults in CMOS Storage Elements *

Waleed K. Al-Assadi, Yashwant K. Malaiya†, and Anura P. Jayasumana
Department of Electrical Engineering
† Computer Science Department
Colorado State University
Fort Collins, CO 80523

## Abstract

In testing sequential circuits, internal faults in the storage elements (SEs) are sometimes modeled as stuck-at faults in the combinational circuits surrounding the SE. The detection of some transistor-level faults that cannot be modeled as stuck-at are considered. These *feed-through* faults, cause the cell to become either *data-feed-through*, which makes the cell combinational, or *clock-feed-through*, causing the clock signal or its complement to appear at the output. Under such faults, the cell does not function as a memory element. Here it is shown that such faults may or may not be detected depending on delays involved. Conditions under which *race-ahead* occurs are identified.

*This work was supported by a SDIO/IST funded project monitored by ONR.

# 1 Introduction

Testing of sequential circuits has long been known to be a very difficult problem. Unlike the combinational logic, a test sequence is required to detect a fault in a sequential circuit. The test sequence has to include an initialization sequence and a propagation sequence. A common approach is to convert the problem of testing synchronous sequential circuits into the simpler problem of testing combinational circuit. This is accomplished by using testable design approaches like LSSD which provide direct access to inputs and outputs of combinational blocks [1, 2, 3]. If one can assumes that most faults within a SE can be modeled as stuck-at-0/1 faults on the inputs or outputs, then these faults do not need to be explicitly considered. This is because such faults are equivalent to the stuck-at faults in the combinational logic surrounding the SEs.

Considering SEs as primitives for the purpose of fault simulation and test generation for sequential circuits can significantly reduce computational complexity. This paper considers the problem of detecting faults in the CMOS synchronous SEs that cannot be modeled as stuck-at-0/1. Such faults, termed *feed-through* faults, cause a SE to become either *data-feed-through* or *clock- feed-through* and cause the cell to lose the sequential behavior [4]. These faults generally occur due to some internal bridging faults and are independent of transistor sizing.. As an example consider the D-latch in Figure 1. Bridging faults between nodes $D$ and $D1$ causes the cell to become *data-feed-through*, i.e. $Q = D$. Bridging faults between nodes $CLK$ or $\overline{CLK}$ and D1 cause the cell to be *clock-feed-through*, i.e. $Q = CLK$ or $\overline{CLK}$. These faults can lead to timing problems because of coupling between combinational blocks normally separated by SEs. The formal definition for these two behaviors is given below.

**Definition 1:** Let T=$\{t_1,....,t_n\}$ be the set of all possible input combinations for an elementary synchronous SE with input $D$ and a control signal $CLK$. Here $t_i$ is a 2-tuple corresponding to $(D, CLK)$ and n=4. Let $R(s, t_i)$ be the response of the cell to the input

vector $t_i$ applied to the cell when the cell is at state $s$. A faulty SE cell is said to have a *feed-through* fault if it becomes either *data-feed-through* or *clock-feed-through.*

(i) A faulty SE cell is said to be *data-feed-through* when its behavior becomes combinational such that $R(s, t_i) = f(y)$ for each $t_i \in T$, where y is the data part of $t_i$.

(ii) A faulty synchronous SE cell is said to be *clock-feed-through* if $R(s, t_i) = CLK$ or $\overline{CLK}$ where CLK is the control signal.

In a master-slave or a two-phase clocking circuit, the *clock-feed-through* fault may cause the succeeding SEs to always latch a 1 or a 0. This will cause the *clock-feed-through* faults to appear as stuck-at faults. Here we will show that in some cases *data-feed-through* faults cannot be detected because they can be masked by combinational propagation delays. If the timing allows, the *data-feed-through* faults can result in *race-ahead*, causing SE to reach the next state one clock period too early. The following section examines the problem of detecting *data-feed-through* faults in detail.

# 2 Feed-through Faults in Sequential Circuits

A latch is in the *transparent phase* when the clock is high. The falling edge of the clock serves as the sampling edge when the latch locks in the input value. and the latch enters the *latch phase.* It is common to use a pair of latches such that they are triggered by non-overlapping clock phases $\phi 1$ and $\phi 2$. This avoids the problem of race-ahead when feedback is present, because at any time only one of the two latches can be in the transparent phase.

When one of the two latches in a pair is *data-feed-through*, it is possible for a transition to race-ahead in one clock period through two combinational blocks (figure 2a). If the circuits involves feedback (figure 2b), then the presence of a *data-feed-through* fault may cause transitions normally corresponding to two successive clock-periods to occur within a single clock-period. This can give rise to race-ahead as defined below.

**Definition 2:** A *race-ahead* occurs when a SE goes from state $s_i$ to $s_{i+2}$ in one clock period, whereas normally a transition from $s_i$ to $s_{i+1}$ should occur, followed by a $s_{i+1}$ to $s_{i+2}$ transition in the next clock period.

It can be shown that a race-head may not occur in some cases and thus a *data-feed-through* fault may not be detected. Let us consider the diagram in figure 2a consisting of two pipelined combinational blocks. For the normal circuit the propagation of a transition (new logical values) can be described by the following sequence:

1. $\phi 1 \downarrow$ : a transition latched in $L11$.
2. $\phi 2 \downarrow$ : corresponding transition (CT) latched in $L12$.
3. $\phi 1 \downarrow$ : CT latched in $L21$ (after passing through $C1$).
4. $\phi 2 \downarrow$ : CT latched in $L22$.
5. $\phi 1 \downarrow$ : CT latched in $L31$ (after passing through $C2$).
6. $\phi 2 \downarrow$ : CT latched in $L32$.

Where $\downarrow$ indicates the falling edge.

The two phases constitute one clock period. To specify the requirements for correct operation, let us adapt the following notation:

$d_{c1}$, $d_{c2}$ = propagation delays through $C1$ and $C2$, respectively.

$d_{LCQ}$ = clock-to-output delay through a single latch.

$d_{LDQ}$ = data-to-output delay through a single latch (in transparent phase or when data-feed-through fault is present).

$t_s$ = latch set-up time with respect to the falling edge.

$t_{\phi 1 \phi 2}$, $t_{\phi 2 \phi 1}$ = duration between falling edge of $\phi 1$ and $\phi 2$, and $\phi 2$ and $\phi 1$ respectively.

$g_{\phi 1 \phi 2}$, $g_{\phi 2 \phi 1}$ = the gap between $\phi 1 \downarrow$ and $\phi 2 \uparrow$, and between $\phi 2 \downarrow$ and $\phi 1 \uparrow$ respectively.

The major requirements for correct operation are,

$$t_{\phi 1\phi 2} \geq max(g_{\phi 1\phi 2}, d_{LCQ}) + t_s \quad (1)$$

$$t_{\phi 2\phi 1} \geq max(g_{\phi 1\phi 2}, d_{LCQ}) + d_{c1} + t_s \quad (2)$$

and similarly

$$t_{\phi 2\phi 1} \geq max(g_{\phi 1\phi 2}, d_{LCQ}) + d_{c1}max + t_s \quad (3)$$

Now let us consider the case when latch $L21$ has data-feed-through fault, when $\phi 2$ signal is active, there exists a combinational path $L12 - C1 - L21 - L22$. The following sequence is possible:

1. $\phi 1 \downarrow$ = a transition latched in $L11$.
2. $\phi 2 \downarrow$ = CT latched in $L22$, provided the inequality (4) below is satisfied:
3. $\phi 1 \downarrow$: CT latched in $L31$.
4. $\phi 2 \downarrow$ : CT latched in $L32$.

Thus between two successive falling edges of $\phi 2$ (i.e. within a single clock period) both $C1$ and $C2$ are traversed. This can occur only if:

$$t_{\phi 1\phi 2} \geq max(g_{\phi 1\phi 2}, d_{LCQ}) + d_{LDQ} + d_{c1} + d_{LDQ} + t_s \quad (4)$$

where $d_{c1}$ is the delay through a sensitized path.

In some situation, the condition in inequality (4) may not be satisfied and the following sequence may occur.

1. $\phi 1 \downarrow$ : a transition latched in $L11$.
2. $\phi 2 \downarrow$ : CT does not arrive at $L22$ in time but is latched in $L12$.
3. $\phi 1 \downarrow$ : has no effect on $L21$.
4. $\phi 2 \downarrow$ : CT arrives at $L22$ and is latched.

5. $\phi 1 \downarrow$ : CT latched in $L31$.

6. $\phi 2 \downarrow$ : CT latched in $L32$.

The above sequence will occur if:

$$T \geq d_{LCQ} + d_{c1} + d_{LDQ} + t_s \tag{5}$$

Where $T$ is the clock period ($t_{\phi 1 \phi 2} + t_{\phi 2 \phi 1}$), and if (4) does not hold. In this case the operation is normal and race-ahead is not observed. This suggest that in some cases higher propagation delays can mask *data-feed-through* faults.

Similar conditions can be obtained if the second latch, for example $L22$ has a *data-feed-through* fault. Race-ahead can also occur in circuits with feedback [4].

# 3 Conclusion

Some defects in storage elements can cause *feed-through* faults which need to be considered when high fault coverage is required. The *clock-feed-through* faults will generally appear as stuck-at-0/1 faults. We have shown here that *data-feed-through* faults can cause race-ahead in synchronous sequential circuits. In some cases, the *data-feed-through* faults may be masked. The conditions for these have been presented.

# References


[1] M. K. Reddy and S. M. Reddy, "Detecting FET Stuck-Open Faults in CMOS Latches and Flipflops," *IEEE Design and Test,* pp. 17-26, October 1986.

[2] D. L. Liu and E. J. McCluskey, "A CMOS Cell Library Design for Testability," *VLSI Systems Design,* pp. 58-65, May 4, 1987.

[3] R. Anglada and R. Rubio, "Functional Fault Models for Sequential Circuits," *Research Report DEE-3*, Electronic Engineering Department, Polytechnical University of Catalunya, Barcelona 1987.

[4] W. K. Al-Assadi, Y. K. Malaiya, and A. P. Jayasumana, " Use of Storage Elements as Primitives for Modeling Faults in Sequential Circuits," To appear in Proc. Int. Conference on VLSI Design, January 1993.

[5] Y. K. Malaiya and R, Narayanaswamy, "Modeling and Testing for Timing Faults in Synchronous Sequential Circuits," *IEEE Design & Test of Computers,* vol. 1, pp. 62-74, November 1984.

[6] M. R. Dagenais and N. C. Rumin, "On the Calculation of Optimal Clocking parameters in Synchronous Circuits with Level-Sensitive Latches," *IEEE Transaction on Computer-Aided Design,* vol.8, no. 3, pp. 268-278, March 1989.

[7] S. H. Unger and C. Tan, "Clocking Schemes for High-Speed Digital Systems," *IEEE Transaction on Computers,* vol. C-35, no. 10, pp. 880-895, October 1986.

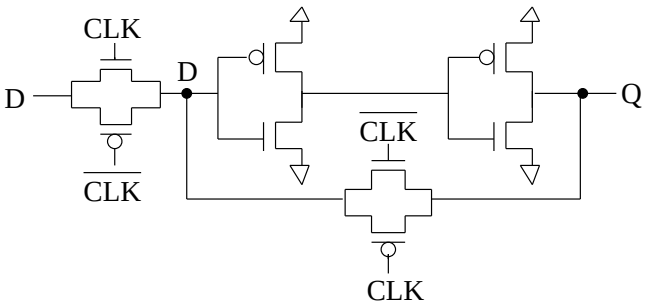
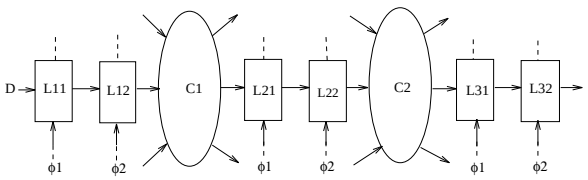
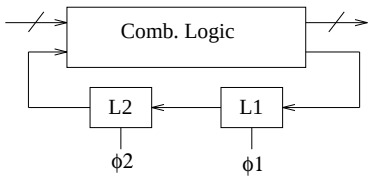



Figure 1: The transmission gate latch




(a) Two-pipeline stage sequential circuit




(b) Synchronous sequential circuit with feedback

Figure 2: Use of lathes with 2-phased clock

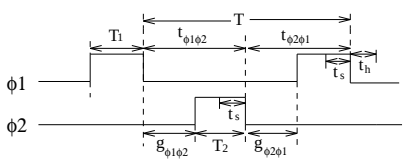



Figure 3: The two clock phases