Detailed Analysis of Bridging Faults in CMOS Scan Registers *

by

Kuen-Jong Lee and Melvin A. Breuer

Technical Report No. CENG 89-05

Detailed Analysis of Bridging Faults in CMOS Scan Registers *

by

Kuen-Jong Lee and Melvin A. Breuer

Technical Report No. CENG 89-05

Detailed Analysis of Bridging Faults in CMOS Scan Registers*

Kuen-Jong Lee Melvin A. Breuer

Department of Electrical Engineering-Systems
University of Southern California
Los Angeles, CA 90089-0781

January 14, 1989

Abstract

In this paper all possible single bridging faults(BFs) within the scan path of a scan-based CMOS circuit are analyzed. Several new observations on the effects of BFs are given. It is shown that some BFs can only be detected by monitoring the current supply, and some only by observing scanned test data. However, all but one BF can be detected when both methods are used. The one which cannot be detected is a redundant fault and has no effect on the logic function of the scan path. A test sequence which can detect all irredundant faults is derived. The length of this test

*This work was supported by Defense Advanced Research Projects Agency and monitored by the Office of Naval Research under contract no. N00014-87-K-0861.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

sequence is $2n + 3$ bits, where n is the number of storage elements in the scan path. The results of SPICE simulation for those faults requiring current supply monitor are given.

1 Introduction

As the complexity of CMOS circuits increases, there is an increasing demand for more efficient and effective ways of generating tests as well as testing an IC. The behavior of a faulty circuit can be described using several classical fault models, such as *stuck-at 0/1 faults, stuck on/open faults* and *bridging faults.* Recent studies have shown that about half of the faults in CMOS circuits are bridging faults(BFs)[1]. This suggests the importance of testing for such faults.

Detailed examinations of BFs in combinational circuits can be found in [2,3]. However, little analysis of BFs in sequential circuits exists. Test generation in sequential circuits is an extremely difficult problem. The complexity of this problem can be reduced by using *design-for-test* techniques[4]. For example, by employing a scan-based design, such as LSSD[5], the sequential circuit test generation problem reduces to one of generating tests for a combinational circuit. However, a BF may occur within a scan register. Thus it is essential to detect faults in scan registers in order to guarantee the proper function of a scan-based sequential circuit.

Classical methods for detecting BFs have adopted a *wired-OR* or *wired-AND* model which assumes that when two nodes with complemented values are shorted, the resulting voltage on both nodes is either logic high (*wired-OR*) or logic low (*wired-AND*). By this assumption a bridging fault can be detected using a method similar to that used for detecting stuck-at faults[6,7]. However, this model is often not applicable to CMOS circuits where a bridging fault may force both nodes to take on an intermediate voltage value v between VDD and GND, where v cannot be interpreted as a logic high or logic low. Furthermore, as will be described, most BFs in a scan path may change the state of storage elements, and a *wired-OR* or *wired-AND* model does not help in detecting such BFs.

An alternative way to detect BFs is by monitoring the current supplied by the power lines[8,9,10]. This method is based on one of the design criteria for CMOS circuits, namely during steady state a fault-free circuit should never have a conducting path between *VDD* and *GND*, and thus only a very small *leakage current*(typically of the order of 10^{-9} amperes) is consumed. When two nodes are shorted together, where one is connected to *VDD* and the other to *GND* through paths of conducting transistors, the supply current becomes much larger than normal. In our SPICE simulations we have obtained values of the order of 10^{-4} amperes. Even if the bridging fault results in only a partially conducting path between *VDD* and *GND*, SPICE simulations indicate that the steady state current is still of the same order as, or at most one order less than the current where a fully conducting path exists. An example of this type of situation would be a BF between the input and output of an inverter. In some cases a BF results in circuit oscillation. However, an oscillation implies that the circuit is in a transient state and thus a large supply current probably exists. SPICE simulations show that for this situation the supply current is of the same order as that which exists for a fully conducting path. Thus by monitoring the steady state current it is possible to detect the existence of a bridging fault. This method will be referred to as the *current supply monitoring method* (CSM).

Current methods for detecting faults in scan registers consist of scanning in a test sequence consisting of 1's and 0's, and observing the scan-out data[5]. This approach has two problems. First, as will be shown, this may not detect some bridging faults in a scan register. Secondly, it is not clear exactly what test sequence should be used. To overcome these problems, a systematic analysis for all possible bridging faults which can occur in scan registers is necessary. The results of this analysis should indicate which BFs are detectable and which are redundant. If a fault is detectable, then a sequence for detecting this fault can be derived. From these results a test sequence which detects all detectable faults can be constructed.

In this paper all possible BFs in the scan path of a scan-based system are considered. The paper is organized as follows. Section 2 describes scan registers and related notation. Section 3 discusses the effects of bridging fault in scan registers. Some BFs are shown to be detectable only by monitoring the current supply, and some only by observing scan test data. Several effects of bridging fault in scan registers which have not been considered previously in the literature are discussed. It is shown that to ensure high fault coverage, such a careful examination is required. A systematic analysis for all possible BFs in scan registers is given in Section 4. It is shown that all but one BF is detectable if both current monitoring and normal monitoring of output data are employed. The one fault which cannot be detected is actually a redundant fault and cannot affect the normal operation of the circuit. A test for each irredundant fault is derived. Section 5 summarizes the results of Section 4 and gives a simple test sequence which can detect all irredundant BFs in a scan path. The length of this sequence is $2n + 3$, where n is the number of storage elements in the scan path. Section 6 gives the results of SPICE simulations for faults requiring CSM.

2 CMOS Scan Registers & Test Sequences

Scan registers can be implemented in several ways[11]. The one shown in Figure 1 will be used in this paper. We will only consider bridging faults between nodes within the scan path. Detecting bridging faults which involve data input nodes to the scan register (DIs in Figure 1) or nodes in the combinational part of circuit will not be discussed here. Two-phase clocking (ϕ_1 and ϕ_2) on the scan path is assumed. Each system clock cycle is partitioned into 4 subcycles: $p_1 = (\phi_1 = 0, \phi_2 = 0)$, $p_2 = (\phi_1 = 1, \phi_2 = 0)$, $p_3 = (\phi_1 = 0, \phi_2 = 0)$ and $p_4 = (\phi_1 = 0, \phi_2 = 1)$ as shown in Figure 2(a). Figure 2(b) shows the circuit configuration of the scan path during each subcycle.

Without loss of generality assume a bridging fault occurs between nodes x and y, where x is closer to the scan input (SI in Figure 1) than y is. A *scan cell* consists of two inverters and two pass transistors, and a *storage element* consists of two scan cells. In Figure 1, C_x and C_y are two scan cells and S_x and S_y are two storage elements. In general C_a (S_a) denotes a scan cell (storage element) which contains node a. C_{a-1} and C_{a+1} represent the scan cells which immediately precede and follow C_a, respectively. A scan cell always forms a feedback loop during p_1 and p_3. It also forms a loop either during p_2 or p_4, but not both. For example in Figure 2, C_x forms a loop during p_1, p_3 and p_4. During p_2, x is actually in the feedback loop of cell C_{x-1}.

The switch-level representation of a scan cell is indicated in Figure 3(a). A bridging fault which involves *VDD* or *GND* or any clock signal is assumed to be easily detected and is not considered in this paper. Thus in each scan cell three nodes (nodes 1, 2 and 3) may be shorted (bridged) to other nodes. These three nodes correspond to the three nodes in the gate-level description as shown in Figure 3(b). For convenience, a scan cell C_a is said to have a value v if the value of its node 1 is v. This is denoted by $C_a = v$. Thus $C_x = 0$ means the first node in scan cell C_x has a value of 0. This also implies that the second and third nodes of C_x have values of 1 and 0, respectively.

For each node a, two functions, $n(a)$, $l(a)$, are defined. $n(a)$ is the number of scan cells between the scan input and the scan cell containing node a, and $l(a)$ is the location of a (1, 2 or 3) in the scan cell. A *distance function* D between two nodes x and y is defined as $D(x,y) = n(y) - n(x)$. For example in Figure 1, $l(x) = 2$, $l(y) = 1$ and $D(x,y) = 3$.

A BF on the scan path can be detected by scanning a test sequence through the scan path and observing the scanned-out data or monitoring the current supply. For CSM, the fault is detected when the appropriate part of the test sequence is scanned through the fault site. It is not necessary to scan out the test data. As will be shown later, for some BF which cannot be detected by CSM, it may not be necessary to fully scan out the test sequence to detect the fault. We will use the notation $ud(k)v$ to denote a sequence of scan data, where v and u are the first and last bit to be scanned in, and $d(k)$ is a sequence of k *don't care* bits, $k \geq 0$. When a bit must be scanned out and observed, it will be underlined. For example, $1d(3)\underline{0}$ denotes a sequence of bits $1\times\times\times0$ and only the first bit (0) need be scanned out and observed. The complement value of v is denoted by $\bar{v}$. Thus a sequence $\underline{v}d(k)\underline{\bar{v}}$ means the first bit and the last bit must be complement values, and both must be scanned out and observed.

3 Effects of BFs

This section analyzes the effects of BFs. The necessity of both monitoring currents and observing scan data in order to detect all irredundant BFs is illustrated. The reason why a test sequence of arbitrary 1's and 0's may fail to detect some BFs is also given.

The necessity of CSM is obvious. Consider a BF between the input and output of an inverter. Since for many CMOS circuits the logic value for this situation cannot be determined, the only way to detect such a fault is by using CSM. The necessity of observing scan data needs some justification. The basic requirement for CSM is that the BF to be detected must cause a large current consumption in the faulty circuit when an appropriate test vector is applied. This is generally true for BFs in combinational circuits, even when oscillation occurs. However, in a sequential circuit, the state of a storage element may change due to a BF and/or reach an erroneous steady state where no conducting path exists between *VDD* and *GND*. Consider a BF between node $1(x)$ of C_x and node $1(y)$ of C_y in Figure 2. Assume during p_1, $C_{x-1} = 1$ and $C_{y-1} = 0$. When entering p_2, the BF results in a *clash* between the feedback loops in C_{x-1} and C_{y-1}. Since there is no external control for these two loops, eventually both will reach a steady state where either both are logic high or logic low. There will be no large current consumption during this steady state. Thus to detect such a fault the test data must be scanned out and observed.

The above clash situation occurs whenever two loops which are shorted become *isolated* from their input lines due to the clock setting. The final value on both loops depends on circuit implementation parameters such as the physical size of transistors. Four possible results may occur when two cells C_x and C_y clash: 1) cell C_x dominates, 2) cell C_y dominates, 3) logic high dominates, and 4) logic low dominates. We assume that the dominance relation between two cells for a particular BF is time-invariant. For example, if C_x dominates C_y when there is a BF between them at a certain time, then C_x always dominates C_y for that BF. However, for a different BF between C_x and C_y (there are 9 different BFs between C_x and C_y), the dominance relation may be different, e.g., C_y may dominate C_x. For two different pairs of clashes their dominance results are assumed to be independent.

Now consider the detection of BFs which may result in clashes. The BF between x and y in Figure 2 is easy to detect. If a test sequence contains a transition from 0 to 1 or 1 to 0, then the fault is detected. However, not all BFs are so east to detect. Consider a fault between x and y such that $D(x, y) = 4$ and $l(x) = l(y) = 1$. Unless a test sequence containing 1×0 or 0×1 is used, the fault cannot be detected. The test sequence 010101... obviously does not detect such a fault. One may think the problem is still easy and the only requirement is to scan two different values to two shorted nodes such that the clash results in a value change at one node. By scanning out the changed value, the fault can be detected. Unfortunately, this is not always true due to two reasons. First, two complement values must be scanned to the two shorted nodes in order to have a clash. However before arriving at the shorted nodes, the test data may have been changed. Secondly, even if this data successfully arrives at the shorted nodes and a clash occurs, the test data still must be scanned out and observed. During the scan out process, the test data may again be modified to its original value and thus the fault effect is masked. Consider BF F14 in Figure 4. Assume 1 and 0 have been successfully scanned to the second node of C_y and the third node of C_{x-1} respectively. There is a clash between these two cells during next p_2. Assume logic high dominates this clash and thus the value at x is changed to 1. To observe this effect, this value must be scanned out. But during the scan out process, this 1 must pass through C_y and node y should take the value 0, since $l(x) = 1, l(y) = 2$. If at this time the value of x happens to be 1, then the clash between C_y and C_{x-1} occurs again. Since 1 dominates this clash, the value of y will be changed to 1 and the fault effect is masked.

An even more complex case is F13, where two clashes between two different pairs of cells (C_{x-1} and C_y during p_2, and C_x and C_{y-1} during p_4) can occur. Since there are 4 possible results for each clash, sixteen subcases have to be considered for this fault.

The above discussion suggests that unless a systematic analysis for each fault is given, it is difficult to determine the coverage of BFs in a scan path for a particular test sequence.

4 Detailed Examination of BFs

This section gives a systematic analysis on all possible BFs in a scan path. The BFs are classified into 4 major categories according to the value of $D(x, y)$. Under each category, each fault is further classified using the values of $l(x)$ and $l(y)$. It is shown that only one type of fault may be redundant ($D(x, y) = 0$, $(l(x), l(y)) = (1, 3)$). Without loss of generality, in the following discussion the pass transistors in C_x are controlled by ϕ_1 and $\overline{\phi_1}$. The results can be applied to the case where the pass transistors in C_x are controlled by ϕ_2 and $\overline{\phi_2}$ by interchanging p_1 with p_3, and p_2 with p_4. For each fault, a test sequence (or sequences) is given which detects the fault when applied to the scan path. All discussions refer to Figure 4.

4.1 $D(x, y) = 0$ — BFs in the same scan cell

F1: $(l(x), l(y)) = (1, 2)$ or $(2, 3)$. This fault always causes a short between the input and output of an inverter and thus is always detected by CSM.

F2: $l(x) = 1$, $l(y) = 3$. This fault forces C_y to have a permanent loop. Thus if C_{x-1} and C_y had different values at p_1, then a clash between them occurs during p_2. If C_y always dominates this clash, a sequence $b_2 b_1 = \underline{v\bar{v}}$ ($\bar{v}$ is the first bit to be scanned in) can detect this fault as described next. At some p_1, b_1 is in C_y and C_{y+1}, and b_2 is in C_{x-1}. If b_1 has been changed to v at this time due to the BF, then C_{y+1} must have the same value v and this value cannot be further changed during the scan out process. Thus the fault can be detected by observing b_1. If b_1 is not changed, then since b_2 cannot be changed before arriving at C_{x-1}, there must be a clash between C_{x-1} and C_y during p_2. Since C_y always dominates the clash, the value in C_y cannot be changed to v as in the fault-free circuit.

Thus during p_3, C_y still has a value of $\bar{v}$ and hence the value of b_2 is not propagated through C_y. This fault effect cannot be masked during further scan operation and the fault will be detected by observing b_2. Thus $b_2b_1 = \underline{v\bar{v}}$ can detect this fault if C_y always dominates the clash. If logic high (1) always dominates the clash, then the sequence $b_2b_1 = \underline{0}1$ will detect this fault as described next. Since 1 always dominates the clash, b_1 can never be changed. b_2 cannot be changed before arriving C_{x-1}. Thus a clash occurs during some p_2 and the value of b_2 will be changed to a 1 at that time. This 1 cannot be further changed and thus the fault can be detected by observing b_2. Similarly if 0 always dominates the clash, then $b_2b_1 = \underline{1}0$ can detect the fault. Finally if C_{x-1} always dominates the clash, then during p_2 the value of C_{x-1} is propagated to C_y in both the faulty and fault-free circuits. Thus this fault is redundant. Obviously this fault cannot affect the logic function of the scan path. As will become clear later, this is the only redundant fault among all possible BFs. In summary this fault is either redundant or is detected by a test sequence which contains $\underline{10}$ and $\underline{01}$.

4.2 $D(x, y) = 1$ — BFs between two adjacent scan cells

F3: $(l(x), l(y)) = (1,1)$ or $(1,3)$. A clash between C_{x-1} and C_y occurs during p_2 if such a fault exists. If C_y always dominates the clash, a sequence $b_2b_1 = \underline{v\bar{v}}$ can detect this fault as described next. If the value of b_1 becomes v when b_1 is in C_y and b_2 is in C_x and C_{x-1}, b_1 cannot be further changed during the scan out operation. If b_1 is not changed, b_2 will be changed to $\bar{v}$ by the clash and cannot be further changed thereafter. Thus $\underline{v\bar{v}}$ detects the fault. If C_{x-1} dominates, then $b_2b_1 = v\underline{\bar{v}}$ can detect the fault since the value of b_1 will be changed to v during the clash and this fault effect cannot be masked. If 1 always dominates, then $b_2b_1 = 1\underline{0}$ or $b_2b_1 = \underline{0}1$ detects this fault since the 0 bit in both cases will be changed to 1 and can be observed. If 0 always dominates, then similarly $b_2b_1 = \underline{1}0$ or $b_2b_1 = 0\underline{1}$ detects this fault. In summary, a sequence $b_2b_1 = \underline{v\bar{v}}$ detects this fault.

F4: $(l(x), l(y)) = (1,2)$ or $(3,2)$ or $(2,1)$ or $(2,3)$. During p_4 this fault causes a short between the input and output of either an inverter or three serial inverters. In both cases a large current is drawn on the power lines, even if oscillation occurs, and thus this fault can be detected by CSM.

F5: $(l(x), l(y)) = (2,2)$ or $(3,1)$ or $(3,3)$: This fault can be detected by $b_2b_1 = v\underline{\bar{v}}$ since the value of b_1 will be changed to v during some p_2 and this change cannot be masked thereafter.

4.3 $D(x, y) = 2k,\ k \geq 1$

F6: $l(x) = l(y) = 1$. A clash occurs between C_{x-1} and C_{y-1} during p_2 if these cells had different values during p_1. If C_{y-1} dominates the clash, this fault can be detected by the sequence $b_{k+1}d(k-1)b_1 = \underline{v}d(k-1)\underline{\bar{v}}$ since either b_1 is changed to v when it passes through C_{x-1}, or b_{k+1} is changed to $\bar{v}$ if b_1 is not changed. If C_{x-1} dominates, then $b_{k+1}d(k-1)b_1 = vd(k-1)\underline{\bar{v}}$ can detect the fault since b_1 will be changed to v when passing through C_{y-1}. If 1 always dominates, then $\underline{0}d(k-1)1$ or $1d(k-1)\underline{0}$ detects this fault. Similarly if 0 always dominates, then $\underline{1}d(k-1)0$ or $0d(k-1)\underline{1}$ detects this fault. In summary a sequence of $b_{k+1}d(k-1)b_1 = \underline{v}d(k-1)\underline{\bar{v}}$ detects this fault.

F7: $l(x) = 1,\ l(y) = 2$. If this fault is present, the value of x is dominated by the value of y during p_2, i.e., node x takes the value of node y, since y is controlled by an inverter which is not in a feedback loop. Thus the sequence $b_{k+1}d(k-1)b_1 = \underline{v}d(k-1)\underline{v}$ detects this fault since either b_1 is changed to $\bar{v}$ before arriving at C_y, or b_1 is not changed but b_{k+1} is changed to $\bar{v}$. Note that b_{k+1} and b_1 have the same value instead of complement values.

F8: $l(x) = 1,\ l(y) = 3$. The discussion for this fault is the same as for **F7**, except that the test sequence must contain $\underline{v}d(k-1)\underline{\bar{v}}$.

F9: $l(x) = 2$, $l(y) = 1$. During p_2 the value of x always dominates the value of y and thus the sequence $b_{k+1}d(k-1)b_1 = vd(k-1)\underline{v}$ detects this fault.

F10: $(l(x), l(y)) = (2,2)$ or $(3,3)$. During p_2, x and y are both driven by an inverter which is not in a feedback loop. Thus by scanning in $b_{k+1}d(k-1)b_1 = vd(k-1)\underline{\bar{v}}$ the fault can be detected by CSM if b_1 is not changed when it arrives at C_{y-1}. If b_1 is changed, then it cannot be further changed and thus the fault effect can be observed at the scan output.

F11: $(l(x), l(y)) = (2,3)$ or $(3,2)$. This case is similar to **F10**, except that the test sequence must contain $b_{k+1}d(k-1)b_1 = vd(k-1)\underline{v}$.

F12: $l(x) = 3$, $l(y) = 1$. During p_2 the value of y is always dominated by the value of x. Thus the sequence $b_{k+1}d(k-1)b_1 = vd(k-1)\underline{\bar{v}}$ detects this fault.

4.4 $D(x,y) = 2k+1,\ k \geq 1$

F13: $l(x) = l(y) = 1$. Two possible clashes can occur when this fault is present: the clash between C_{x-1} and C_y during p_2, and the clash between C_x and C_{y-1} during p_4. The former is referred to as *Clash 1*, and the latter as *Clash 2*. If C_{x-1} dominates *Clash 1*, then a sequence $b_{k+2}d(k)b_1 = vd(k)\underline{\bar{v}}$ can always detect the fault no matter what the result of *Clash 2* is, since the value of b_1 will become v after passing through C_y. If 1 always dominates *Clash 1*, then $1d(k)\underline{0}$ detects the fault as the first bit (0) will be changed to a 1. Similarly if 0 always dominates *Clash 1*, then $0d(k)\underline{1}$ detects this fault. If C_y dominates *Clash 1*, then depending on the result of *Clash 2*, four subcases exist. If C_x dominates *Clash 2*, then the sequence $b_2b_1 = \underline{v\bar{v}}$ can detect this fault as explained next. During some p_1, b_1 is in C_y and C_{y-1}, and b_2 is in C_{y-2}. If the value of b_1 has been changed to v at this time, then this value cannot be further changed (because C_y dominates *Clash 1*) and the fault can be detected by observing b_1. Thus assume b_1 is not changed. When entering p_2, *Clash 1* occurs and the value of C_{x-1} will be changed to $\bar{v}$ no matter what its previous value is since C_y dominates this clash. Also the value of C_x will become $\bar{v}$ since it is controlled by C_{x-1} during p_2. The value of b_2 will arrive C_{y-1} at the same time. When entering the next p_4, *Clash 2* occurs between C_x and C_{y-1}. Since C_x dominates this clash, the value of b_2 will become $\bar{v}$. This value cannot be further changed since C_y dominates *Clash 1*, which is the only way that b_2 can be changed again. Thus if C_x dominates *Clash 2*, a sequence $\underline{v\bar{v}}$ can detect this fault. Now consider the case where C_{y-1} dominates *Clash 2*. This fault can be detected by a sequence $b_{k+1}d(k-1)b_1 = \underline{v}d(k-1)\underline{\bar{v}}$. During some p_3, b_1 is in C_{y-1} and C_{y-2}, and b_{k+1} is in C_x. If b_1 has been changed to v, then b_1 cannot be further changed. Thus the fault can be detected by observing b_1. If b_1 has not been changed, then during the next p_4, *Clash 2* occurs and b_{k+1} will have a value of $\bar{v}$. This value cannot be further changed and thus the fault is detected. If 1(0) always dominates *Clash 2*, then $b_2b_1 = \underline{01}$ ($\underline{10}$) can detect this fault since either b_1 or b_2 will change its value and the fault effect can propagate to scan output. In summary this fault can be detected by a test sequence containing $1d(k)\underline{0}$, $0d(k)\underline{1}$, $\underline{v}d(k-1)\underline{\bar{v}}$, $\underline{01}$ and $\underline{10}$.

F14: $l(x) = 1$, $l(y) = 2$. A clash between C_{x-1} and C_y occurs during p_2. If C_y dominates, then the fault is detected by a sequence $b_{k+1}d(k-1)b_1 = \underline{v}d(k-1)\underline{v}$ as explained next. During some p_3, b_1 is in C_{y-1} and C_{y-2}, and b_{k+1} is in C_x and C_{x-1}. If b_1 has been changed, then it cannot be further changed and the fault is detectable by scanning out and observing b_1. If b_1 is not changed, then during p_4, C_{x+1} and C_x will be dominated by C_y and b_{k+2} will have a value of $\bar{v}$. This value cannot be further changed and thus the fault is detectable. If C_{x-1} dominates the clash, then a sequence $b_{k+2}d(k)b_1 = vd(k)\underline{v}$ can detect this fault since during some p_2, C_{x-1} will dominate C_y and C_{y+1}. Thus the value of b_{k+2} will dominate the value of b_1 such that C_{y+1} becomes $\bar{v}$ and the fault can be detected. If 1(0) dominates the clash, then it can be shown that a sequence $1d(k)\underline{1}$ ($0d(k)\underline{0}$) detects this fault using similar arguments. In summary a test sequence containing $\underline{v}d(k-1)\underline{v}$, $1d(k)\underline{1}$ and $0d(k)\underline{0}$ detects this fault.

F15: $l(x) = 1$, $l(y) = 3$. The analysis for this fault is similar to the case where $l(x) = 1$, $l(y) = 2$. The fault can be detected by a test sequence containing $\underline{v}d(k-1)\underline{\bar{v}}$, $1d(k)\underline{0}$ and $0d(k)\underline{1}$.

F16: $(l(x), l(y)) = (2,1)$, $(2,3)$ or $(3,2)$. This fault is detected by a sequence $b_{k+2}d(k)b_1 = vd(k)\underline{v}$ since during some p_2, C_y is controlled by an inverter of C_x which in turn is controlled by C_{x-1}. Thus the value of b_1 will be changed to $\bar{v}$ and is observable at the scan output.

F17: $(l(x), l(y)) = (2,2)$, $(3,1)$ or $(3,3)$. This fault is similar to the previous one and is detected by a test sequence containing $vd(k)\underline{\bar{v}}$.

5 A Test Sequence for All BFs

An analysis of all possible BFs on the scan path was presented in Section 4. The results are summarized in Table 1. If a test sequence contains all sequences shown, then it must detect all irredundant BFs. Using the notation $1^n(0^n)$ to represent a sequence of n 1's(0's), we have found the following result.

Theorem 1 *A test sequence $TS = 0^n\underline{01^{n+1}0}$ can detect all irredundant BFs in a scan path consisting of n storage elements.*

Proof: The proof follows by examining all sequences in Table 1 and showing that each of them is a subsequence of TS. □

Note that it is not necessary to scan out all test data in TS in order to detect all irredundant faults. Only the first $n + 3$ bits need to be scanned out and observed. The last n bits can serve as reset data (0) for the scan path. Furthermore, TS is very easy to generate. The external test controller need only store one piece of data, i.e., n, in order to generate TS. Since n must be available because it is the length of the scan path, there is no storage overhead for generating TS.

Fault	$D(x,y)$	$(l(x),l(y))$	Test method/sequence
F1	0	(1,2) or (2,3)	CSM
F2	0	(1,3)	Redun. or detected by $\underline{10}$ and $\underline{01}$
F3	1	(1,1) or (1,3)	$\underline{v\bar{v}}$
F4	1	(1,2),(3,2),(2,1) or (2,3)	CSM
F5	1	(2,2),(3,1) or (3,3)	$v\underline{\bar{v}}$
F6	$2k$	(1,1)	$\underline{v}d(k-1)\underline{\bar{v}}$
F7	$2k$	(1,2)	$\underline{v}d(k-1)\underline{v}$
F8	$2k$	(1,3)	$\underline{v}d(k-1)\underline{\bar{v}}$
F9	$2k$	(2,1)	$vd(k-1)\underline{v}$
F10	$2k$	(2,2) or (3,3)	$vd(k-1)\underline{\bar{v}}$ and CSM
F11	$2k$	(2,3) or (3,2)	$vd(k-1)\underline{v}$ and CSM
F12	$2k$	(3,1)	$vd(k-1)\underline{\bar{v}}$
F13	$2k+1$	(1,1)	$1d(k)\underline{0}$, $0d(k)\underline{1}$, $\underline{v}d(k-1)\underline{\bar{v}}$, $\underline{01}$, $\underline{10}$
F14	$2k+1$	(1,2)	$\underline{v}d(k-1)\underline{v}$, $1d(k)\underline{1}$ and $0d(k)\underline{0}$
F15	$2k+1$	(1,3)	$\underline{v}d(k-1)\underline{\bar{v}}$, $1d(k)\underline{0}$ and $0d(k)\underline{1}$
F16	$2k+1$	(2,1),(2,3) or (3,2)	$vd(k)\underline{v}$
F17	$2k+1$	(2,2),(3,1) or (3,3)	$vd(k)\underline{\bar{v}}$

Table 1: Summary of test for each BF

Fault	$(l(x), l(y))$	Subcycle	Oscillation	Current value
F1	(1,2)	p_2	no	8.1×10^{-5}
F1	(2,3)	p_2	no	8.1×10^{-5}
F4	(1,2)	p_4	no	1.3×10^{-4}
F4	(3,2)	p_4	no	9.1×10^{-5}
F4	(2,1)	p_4	no	8.1×10^{-5}
F4	(2,3)	p_4	yes	$8.0 \times 10^{-5} \sim 1.65 \times 10^{-4}$
F10	(2,2)	p_2	no	1.8×10^{-4}
F10	(3,3)	p_2	no	1.5×10^{-4}
F11	(2,3)	p_2	no	1.6×10^{-4}
F11	(3,2)	p_2	no	1.6×10^{-4}

Table 2: Current values when CSM is used

6 SPICE simulation

Table 1 indicates that four types of BFs require CSM. The current values (in amperes) obtained by SPICE simulation for these faults are given in Table 2. The third column shows when the current is measured. The fourth column shows whether or not an oscillation occurs. Among all BFs, only one fault (F4, $(l(x), l(y)) = (2,3)$) results in oscillation. The average current value for this fault is of the same order as for the other faults.

7 Conclusion

In this paper a systematic analysis of all bridging faults on the scan path of a scan-based system is given. It is shown that some BFs can be detected only by monitoring current supply and some only by observing scanned test data. It is also shown that all BFs except one can be detected if both methods are used. The one which cannot be detected is a redundant fault. A test sequence of length $2n + 3$ which can detect all irredundant faults is derived. The advantages of this test sequence are also discussed. SPICE simulation shows that CSM is an effective method for detecting some BFs.

Future work includes the analysis of *TS* for detecting other faults such as stuck-at faults, stuck on/open faults and breaking faults. It is also interesting to investigate the effects of multiple faults within the scan path.

Acknowledgement

The authors would like to thank Dr. Bing Sheu and Dr. Rajiv Gupta for their valuable help and suggestions.

References

[1] F. J. Ferguson and J. P. Shen. Extraction and Simulation of Realistic CMOS Faults using Inductive Fault Analysis. *IEEE Intl. Test Conf.*, 475–484, 1988.

[2] Y. K. Malaiya, A. P. Jayasumana, and R. Rajsuman. A Detailed Examination of Bridging Faults. *IEEE Intl. Test Conf.*, 78–81, 1986.

[3] R. Rajsuman, Y.K. Malaiya, and A.P. Jaylasumana. On Accuracy of Switch-Level Modeling of Bridging Faults in Complex Gates. *IEEE Design Automation Conf.*, 244–250, 1987.

[4] T. W. Williams and K. P. Parker. Design for Testability—A Survey. *The Proceedings of IEEE*, 98–112, Jan. 1983.

[5] E. B. Eichelberger and T. W. Williams. A Logic Design Structure for LSI Testability. *IEEE Design Automation Conf.*, 462–468, 1977.

[6] K. C. Mei. Bridging and Stuck-At-Faults. *IEEE Trans. on Computers*, 720–727, July 1974.

[7] M. Abramovici and P. R. Menon. A Practical Approach to Fault Simulation and Test Generation for Bridging Faults. *IEEE Intl. Test Conf.*, 138–142, 1983.

[8] M. W. Levi. CMOS Is Most Testable. *IEEE Intl. Test Conf.*, 217–220, 1981.

[9] Y. K. Malaiya and S. Y. Su. A New Fault Model and Testing Technique for CMOS Devices. *IEEE Intl. Test Conf.*, 25–34, 1982.

[10] J. M. Acken. Testing for Bridging Faults (Shorts) in CMOS Circuits. *IEEE Design Automation Conf.*, 717–718, 1983.

[11] N. Weste and K. Eshraghian. *Principles of CMOS VLSI Design*, chapter 6, pages 259–269. Addison-Wesley, Mass., 1985.

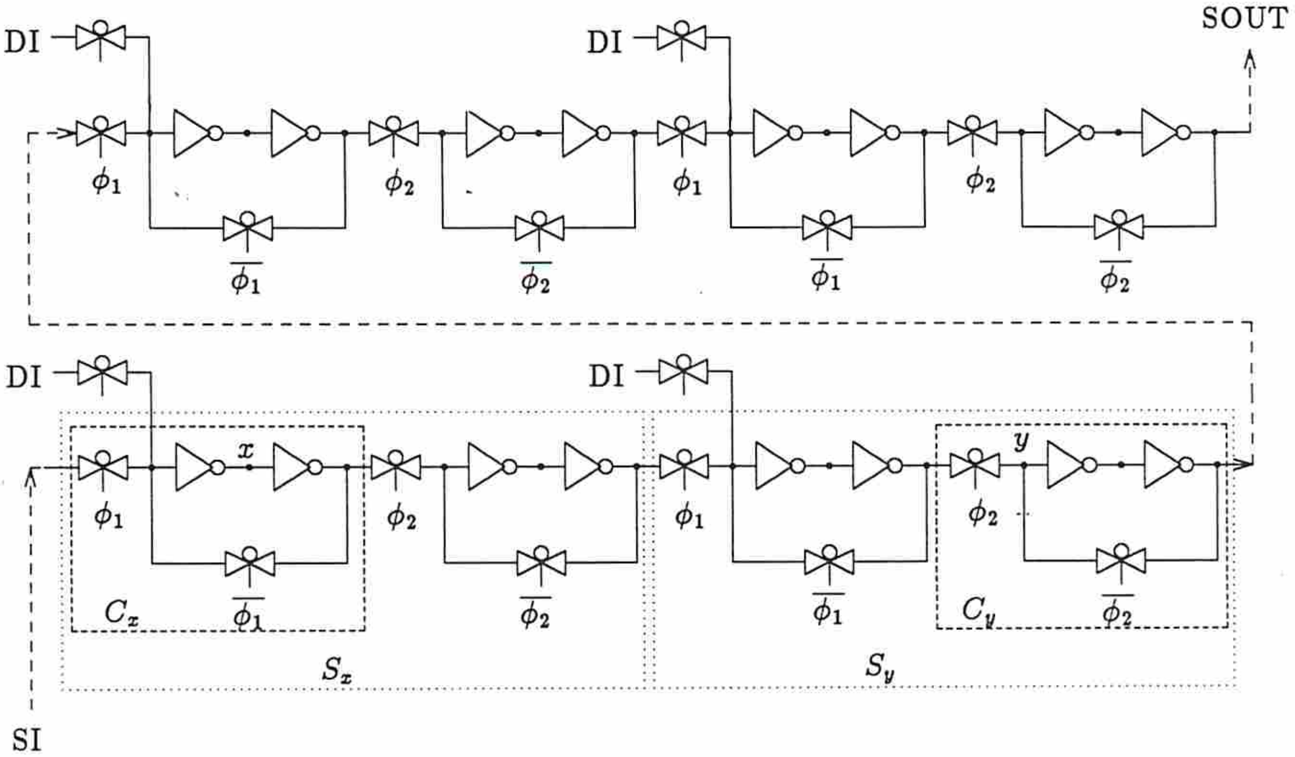
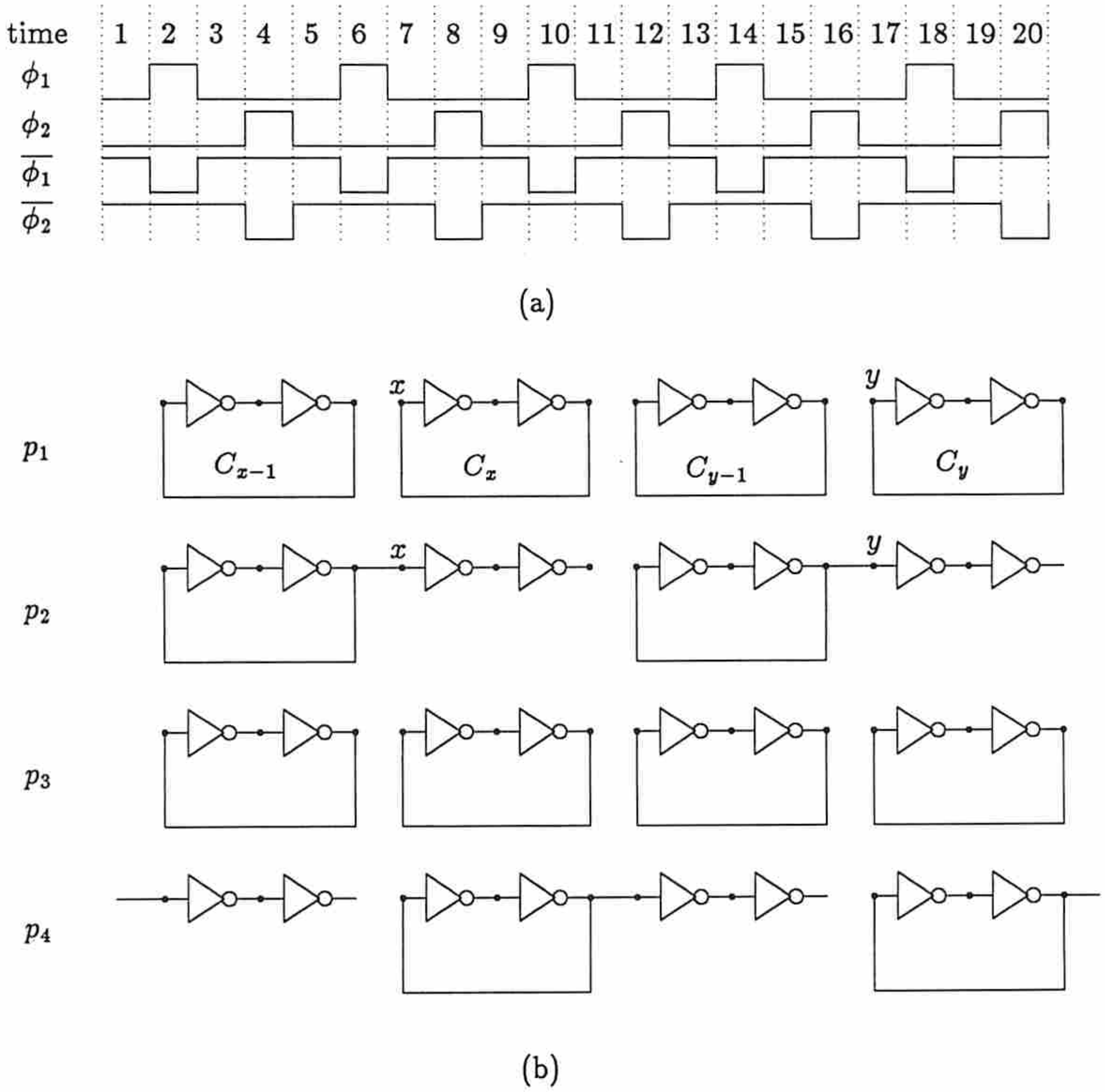


Figure 1: A scan path, storage elements and scan cells



(a)

(b)

Figure 2: Timing of the scan path

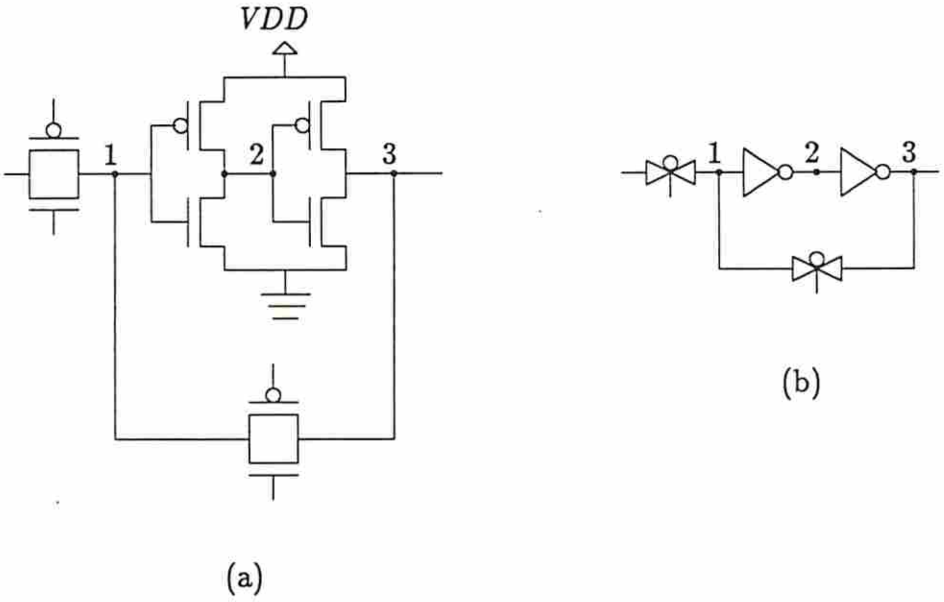


Figure 3: Switch-level(a) and gate-level(b) representations of a scan cell

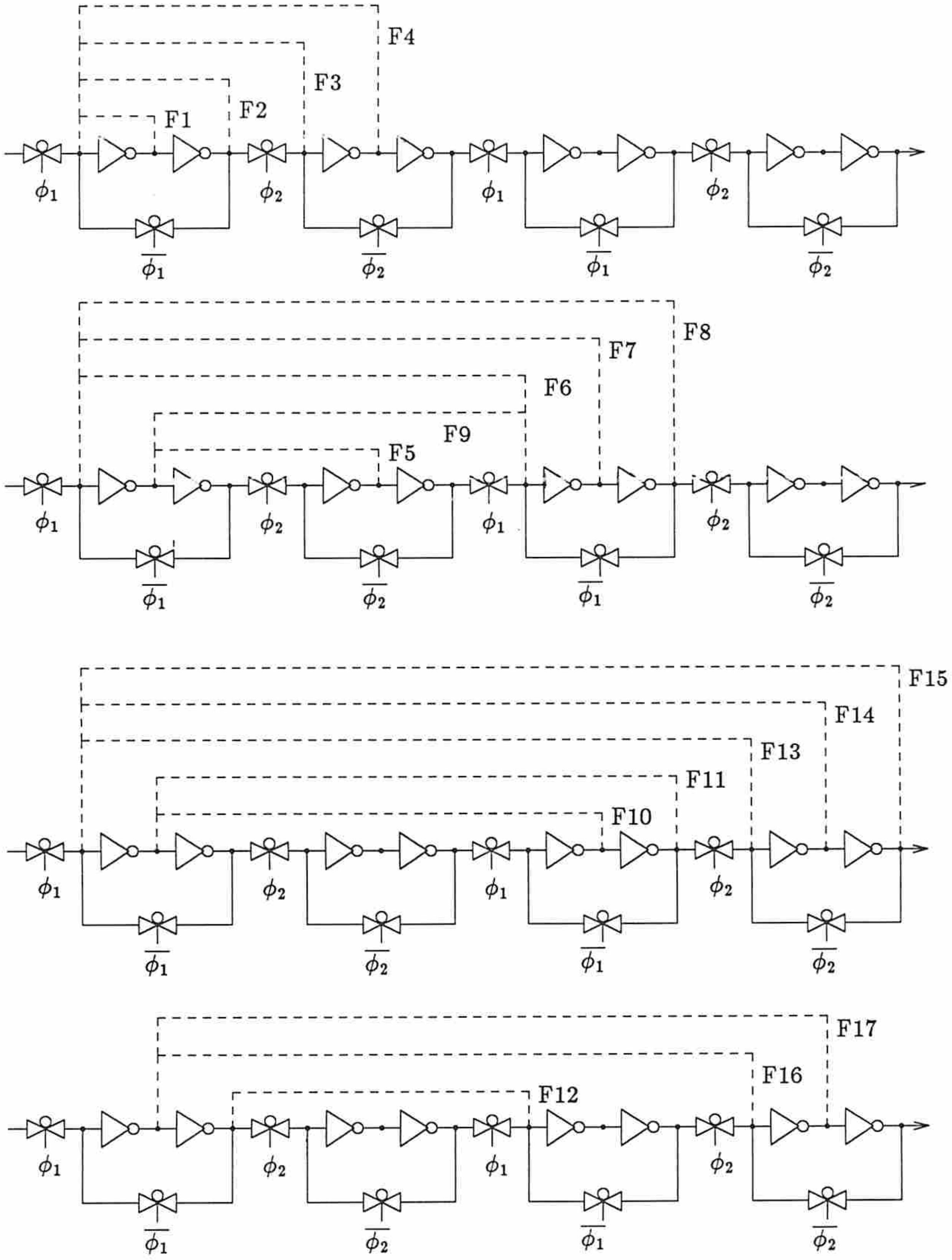


Figure 4: Bridging faults on a scan path