MORGAN & CLAYPOOL PUBLISHERS

An Introduction to Logic Circuit Testing

Parag K. Lala

SYNTHESIS LECTURES ON DIGITAL CIRCUITS AND SYSTEMS

Mitchell A. Thornton, *Series Editor*

An Introduction to Logic Circuit Testing

An Introduction to Logic Circuit Testing
Parag K. Lala
www.morganclaypool.com

ISBN: 9781598293500 paperback

ISBN: 9781598293517 ebook

DOI: 10.2200/S00149ED1V01Y200808DCS017

A Publication in the Morgan & Claypool Publishers series

SYNTHESIS LECTURES ON DIGITAL CIRCUITS AND SYSTEMS # 17
Lecture #17

Series Editor: Mitchell Thornton, Southern Methodist University

Series ISSN
ISSN 1932-3166 print
ISSN 1932-3174 electronic

An Introduction to Logic Circuit Testing

Parag K. Lala
Texas A&M University–Texarkana

SYNTHESIS LECTURES ON DIGITAL CIRCUITS AND SYSTEMS # 17

MORGAN & CLAYPOOL PUBLISHERS

ABSTRACT

An Introduction to Logic Circuit Testing provides a detailed coverage of techniques for test generation and testable design of digital electronic circuits/systems. The material covered in the book should be sufficient for a course, or part of a course, in digital circuit testing for senior-level undergraduate and first-year graduate students in Electrical Engineering and Computer Science. The book will also be a valuable resource for engineers working in the industry. This book has four chapters. Chapter 1 deals with various types of faults that may occur in very large scale integration (VLSI)-based digital circuits. Chapter 2 introduces the major concepts of all test generation techniques such as redundancy, fault coverage, sensitization, and backtracking. Chapter 3 introduces the key concepts of testability, followed by some ad hoc design-for-testability rules that can be used to enhance testability of combinational circuits. Chapter 4 deals with test generation and response evaluation techniques used in BIST (built-in self-test) schemes for VLSI chips.

KEYWORDS

digital circuits, logic circuit testing, VLSI, fault detection, design-for-testability, response evaluation techniques, BIST, D-Algorithm, PODEM, FAN, LFSR

Dedication

To my wife Meena

What lies behind us and lies before us are tiny matters compared to what lies within us.
Ralph Waldo Emerson

Preface

This book provides a detailed coverage of techniques for test generation and testable design of digital electronic circuits/systems. The material covered in the book should be sufficient for a course or part of a course in digital circuit testing for senior-level undergraduate and first-year graduate students in Electrical Engineering and Computer Science.

This book has four chapters. Chapter 1 deals with various types of faults that may occur in very large scale integration (VLSI)-based digital circuits. The modeling of faults at the gate level and at the transistor level is considered. It discusses the fundamental concepts of fault detection and also introduces the concepts of controllability, observability, and fault equivalency. The chapter finishes with a brief discussion of temporary faults.

Chapter 2 introduces the major concepts of all test generation techniques such as redundancy, fault coverage, sensitization, and backtracking. It examines in detail various techniques available for fault detection in combinational logic circuits such as D-algorithm, PODEM, and FAN. This chapter also discusses test generation for sequential circuits. It covers the state table verification approach for fault detection in sequential circuits as well as a technique that utilizes both structure and function (state table) of sequential circuits.

Chapter 3 introduces the key concepts of testability, followed by some ad hoc design-for-testability rules that can be used to enhance testability of combinational circuits. This chapter also covers in detail major design methods for enhancing testability of sequential circuits implemented on VLSI-based digital systems.

Chapter 4 deals with test generation and response evaluation techniques used in built-in self-test (BIST) schemes for VLSI chips. Because linear feedback shift register (LFSR)-based techniques are used in practice to generate test patterns and evaluate output responses in BIST, such techniques are thoroughly discussed. In addition, some popular BIST architectures are examined.

Contents

CHAPTER 1

Introduction

1.1 FAULTS IN LOGIC CIRCUITS

A *failure* is said to have occurred in a logic circuit or system if it deviates from its specified behavior [1]. A *fault*, on the other hand, refers to a physical defect in a circuit. For example, a short between two signal lines in the circuit or a break in a signal line is a physical defect. An *error* is usually the manifestation of a fault in the circuit; thus a fault may change the value of a signal in a circuit from 0 (correct) to 1 (erroneous) or vice versa. However, a fault does not always cause an error; in that case, the fault is considered to be *latent*.

A fault is characterized by its *nature*, *value*, *extent*, and *duration* [2]. The nature of a fault can be classified as *logical* or *nonlogical*. A logical fault causes the logic value at a point in a circuit to become opposite to the specified value. Nonlogical faults include the rest of the faults such as the malfunction of the clock signal, power failure, etc. The value of a logical fault at a point in the circuit indicates whether the fault creates fixed or varying erroneous logical values. The extent of a fault specifies whether the effect of the fault is localized or distributed. A local fault affects only a single variable, whereas a distributed fault affects more than one. A logical fault, for example, is a local fault, whereas the malfunction of the clock is a distributed fault. The duration of a fault refers to whether the fault is *permanent* or *temporary*.

1.1.1 Stuck-At Fault

The most common model used for logical faults is the *single stuck-at fault*. It assumes that a fault in a logic gate results in one of its inputs or the output is fixed at either a logic 0 (*stuck-at-0*) or at logic 1 (*stuck-at-1*). Stuck-at-0 and stuck-at-l faults are often abbreviated to *s-a-0* and *s-a-1*, respectively.

Let us assume that in Figure 1.1 the A input of the NAND gate is s-a-1. The NAND gate perceives the A input as a logic 1 irrespective of the logic value placed on the input. For example, the output of the NAND gate is 0 for the input pattern A=0 and B=1, when input A is s-a-1 in. In the absence of the fault, the output will be 1. Thus, AB=01 can be considered as the *test* for the A input s-a-l, since there is a difference between the output of the fault-free and faulty gate.

The single stuck-at fault model is often referred to as the *classical fault model* and offers a good representation for the most common types of defects [e.g., shorts and opens in complementary

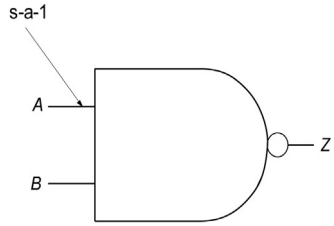

FIGURE 1.1: Two-input NAND gate.

metal oxide semiconductor (CMOS) technology]. Figure 1.2 illustrates the CMOS realization of the two-input NAND:

The number 1 in the figure indicates an open, whereas the numbers 2 and 3 identify the short between the output node and the ground and the short between the output node and the V_{DD}, respectively. A short in a CMOS results if not enough metal is removed by the photolithography, whereas over-removal of metal results in an open circuit [3]. Fault 1 in Figure 1.2 will disconnect input *A* from the gate of transistors *T1* and *T3*. It has been shown that in such a situation one transistor may conduct and the other remain nonconducting [4]. Thus, the fault can be represented by a stuck at value of *A*; if *A* is s-a-0, *T1* will be ON and *T3* OFF, and if *A* is s-a-l, *T1* will be OFF and *T3* ON. Fault 2 forces the output node to be shorted to V_{DD}, that is, the fault can be considered as an s-a-l fault. Similarly, fault 3 forces the output node to be s-a-0.

The stuck-at model is also used to represent multiple faults in circuits. In a multiple stuck-at fault, it is assumed that more than one signal line in the circuit are stuck at logic 1 or logic 0; in other

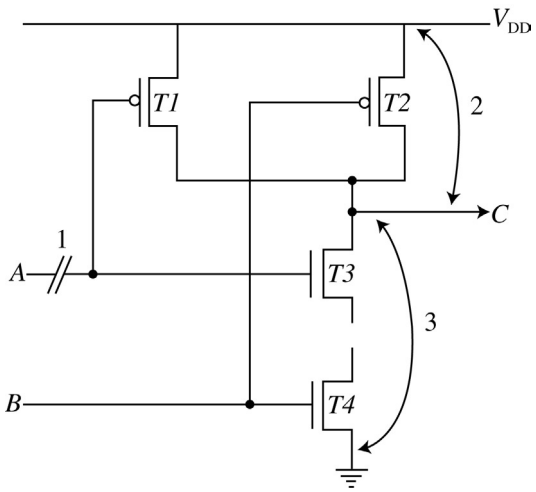

FIGURE 1.2: Two-input NAND gate in CMOS gate.

words, a group of stuck-at faults exist in the circuit at the same time. A variation of the multiple fault is the *unidirectional fault*. A multiple fault is unidirectional if all of its constituent faults are either s-a-0 or s-a-l but not both simultaneously. The stuck-at model has gained wide acceptance in the past mainly because of its relative success with small scale integration. However, it is not very effective in accounting for all faults in present day very large scale integrated (VLSI), circuits which mainly uses CMOS technology. Faults in CMOS circuits do not necessarily produce logical faults that can be described as stuck-at faults [5, 6, 7]. For example, in Figure 1.2, faults 3 and 4 create stuck-on transistors faults. As a further example, we consider Figure 1.3, which represents CMOS implementation of the Boolean function:

$$Z = \overline{(A+B)(C+D)\cdot EF}.$$

Two possible shorts numbered 1 and 2 and two possible opens numbered 3 and 4 are indicated in the diagram. Short number 1 can be modeled by s-a-1 of input E; open number 3 can be modeled by s-a-0 of input E, input F, or both. On the other hand, short number 2 and open number

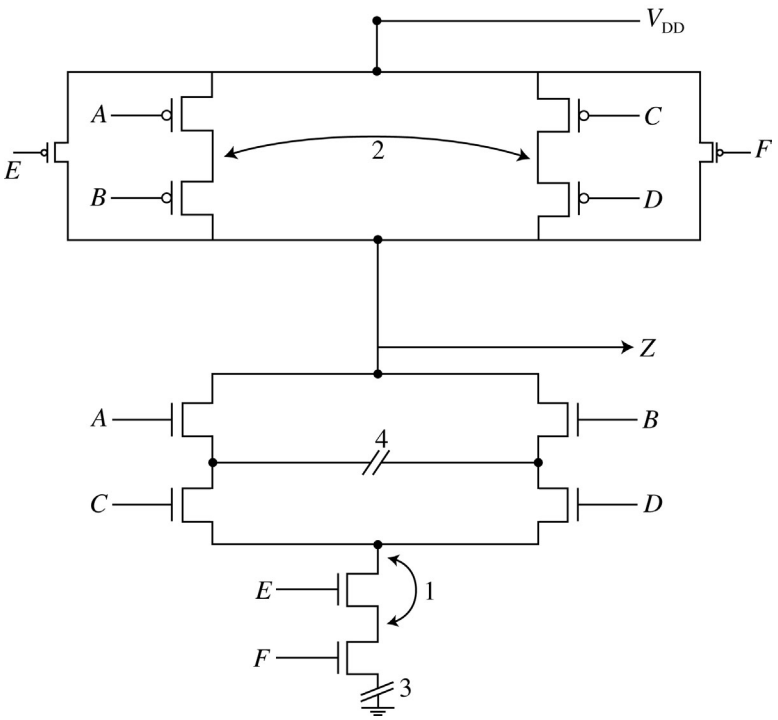

FIGURE 1.3: CMOS implementation of $Z = \overline{(A+B)(C+D)\cdot EF}$.

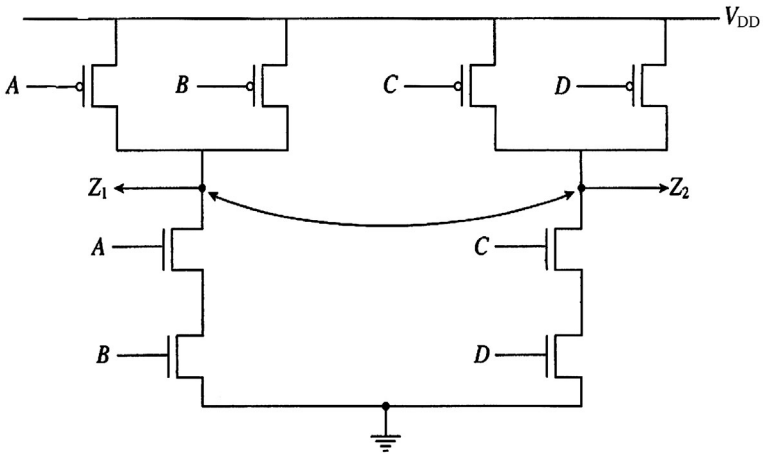

FIGURE 1.4: CMOS implementation of $Z_1 = \overline{AB}$ and $Z_2 = \overline{CD}$.

4 cannot be modeled by any stuck-at fault because they involve a modification of the network function. For example, in the presence of short number 2, the network function will change to:

$$Z = \overline{(A + C)(B + D) \cdot EF},$$

and open number 4 will change the function to:

$$Z = \overline{(AC) + (BD) \cdot EF}.$$

For this reason, a perfect short between the output of the two gates (Figure 1.4) cannot be modeled by a stuck-at fault. Without a short, the outputs of gates Z_1 and Z_2 are:

$$Z_1 = \overline{AB} \quad \text{and} \quad Z_2 = \overline{CD},$$

whereas with the short,

$$Z_1 = Z_2 = \overline{AB} + \overline{CD}.$$

1.1.2 Bridging Faults

Bridging faults form an important class of permanent faults that cannot be modeled as stuck-at faults. A bridging fault is said to have occurred when two or more signal lines in a circuit are ac-

cidentally connected together. Earlier study of bridging faults concentrated only on the shorting of signal lines in gate-level circuits. It was shown that the shorting of lines resulted in *wired logic* at the connection.

Bridging faults at the gate level has been classified into two types: *input bridging* and *feedback bridging*. An input bridging fault corresponds to the shorting of a certain number of primary input lines. A feedback bridging fault results if there is a short between an output and input line. A feedback bridging fault may cause a circuit to oscillate, or it may convert it into a sequential circuit.

Bridging faults in a transistor-level circuit may occur between the terminals of a transistor or between two or more signal lines. Figure 1.5 shows the CMOS logic realization of the Boolean function:

$$Z_1 = Z_2 = \overline{AB} + \overline{CD}$$

A short between two lines, as indicated by the dotted line in the diagram will change the function of the circuit.

The effect of bridging among the terminals of transistors is technology-dependent. For example, in CMOS circuits, such faults manifest as either stuck-at or stuck-open faults, depending on the physical location and the value of the bridging resistance.

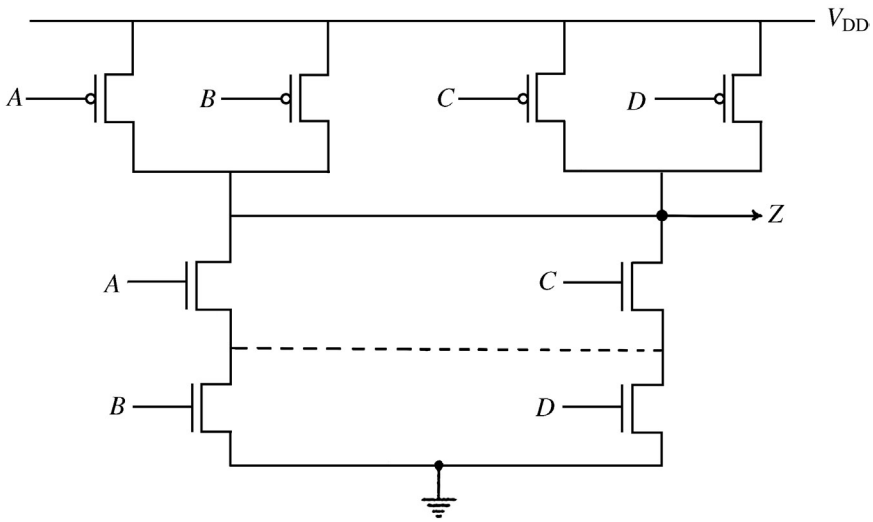

FIGURE 1.5: CMOS implementation of $Z\,(A, B, C, D) = \overline{AB} + \overline{CD}$

1.1.3 Delay Faults

As mentioned previously, not all manufacturing defects in VLSI circuits can be represented by the stuck-at fault model. The size of a defect determines whether the defect will affect the logic function of a circuit. Smaller defects, which are likely to cause partial open or short in a circuit, have a higher probability of occurrence due to the statistical variations in the manufacturing process [8]. These defects result in the failure of a circuit to meet its timing specifications without any alteration of the logic function of the circuit. A small defect may delay the transition of a signal on a line either from 0 to 1, or vice versa. This type of malfunction is modeled by a *delay fault*.

Two types of delay faults have been proposed in literature: *gate delay fault* and *path delay fault*. Gate delay faults have been used to model defects that cause the actual propagation delay of a faulty gate to exceed its specified worst case value. For example, if the specified worst case propagation delay of a gate is x units and the actual delay is $x+\Delta x$ units, then the gate is said to have a delay fault of size Δx. The main deficiency of the gate delay fault model is that it can only be used to model isolated defects, not distributed defects, for example, several small delay defects. The path delay fault model can be used to model isolated as well as distributed defects. In this model, a fault is assumed to have occurred if the propagation delay along a path in the circuit under test exceeds the specified limit.

1.2 BREAKS AND TRANSISTORS STUCK-OPEN AND STUCK-ON OR STUCK-OPEN FAULTS IN CMOS

As discussed previously, not all defects in CMOS VLSI can be represented by using the stuck-at fault model. It has been shown that breaks and transistor stuck-ons are two other types of defects that, like bridging, may remain undetected if testing is performed based on the stuck-at fault assumption. These defects have been found to constitute a significant percentage of defects occurring in CMOS circuits [2]. In the following two subsections, we discuss the effects of these defects on CMOS circuits.

1.2.1 Breaks

Breaks or opens in CMOS circuits are caused either by missing conducting material or extra insulating material. Breaks can be either of the following two types [3]:

1. Intragate breaks;
2. Signal line breaks.

An intragate break occurs internal to a gate. Such a break can disconnect the source, the drain, or the gate from a transistor, identified by b_1, b_2, and b_3, respectively, in Figure 1.6. The presence of b_3, will have no logical effect on the operation of a circuit, but it will increase the propagation delay; that is, the break will result in a delay fault. Similarly, the break at b_1 will also produce a delay fault without changing the function of the circuit. However, the break at b_2 will make the p-transistor nonconducting; that is, the transistor can be assumed to be *stuck-open*.

An intragate break can also disconnect the p-network, the n-network, or both networks (b_4, b_5, and b_6 in Figure 1.6) from the circuit. The presence of b_4 or b_5 will have the same effect as the output node getting stuck-at-0 or stuck-at-1, respectively. In the presence of b_6, the output voltage may have an intermittent stuck-at-1 or stuck-at-0 value; thus, if the output node simultaneously drives a p-transistor and an n-transistor, then one of the transistors will be ON for some unpredictable period of time. Signal line breaks can force the gates of transistors in static CMOS circuits to float.

As shown in Figure 1.6, such a break can make the gate of only a p-transistor and an n-transistor to float. It is also possible, depending on the position of a break, that the gates of both transistors may float, in which case one transistor may conduct and the other remain in a

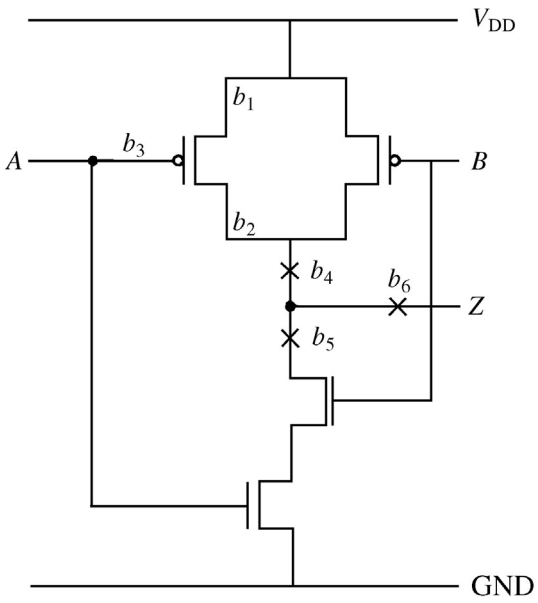

FIGURE 1.6: Two-input CMOS NAND gate showing occurrence of breaks.

nonconducting state [9]. In general, this type of break can be modeled as a stuck-at fault. On the other hand, if two transistors with floating gates are permanently conducting, one of them can be considered as stuck-on. If a transistor with a floating gate remains in a nonconducting state due to a signal line break, the circuit will behave in a similar fashion as it does in the presence of the intragate break b_2.

1.2.2 Stuck-On and Stuck-Open Faults

A *stuck-on* transistor fault implies the permanent closing of the path between the source and the drain of the transistor. Although the stuck-on transistor, in practice, behaves in a similar way as a *stuck-closed* transistor, there is a subtle difference. A stuck-on transistor has the same drain-source resistance as the on resistance of a fault-free transistor, whereas a stuck-closed transistor exhibits a drain-source resistance that is significantly lower than the normal on-resistance. In other words, in the case of stuck-closed transistor, the short between the drain and the source is almost perfect, and this is not true for a stuck-on transistor. A transistor stuck-on (stuck-closed) fault may be modeled as a bridging fault from the source to the drain of a transistor.

A *stuck-open* transistor implies the permanent opening of the connection between the source and the drain of a transistor. The drain-source resistance of a stuck-open transistor is significantly higher than the off-resistance of a nonfaulty transistor. If the drain-source resistance of a faulty transistor is approximately equal to that of a fault-free transistor, then the transistor is considered to be *stuck-off*. For all practical purposes, transistor stuck-off and stuck-open faults are functionally equivalent.

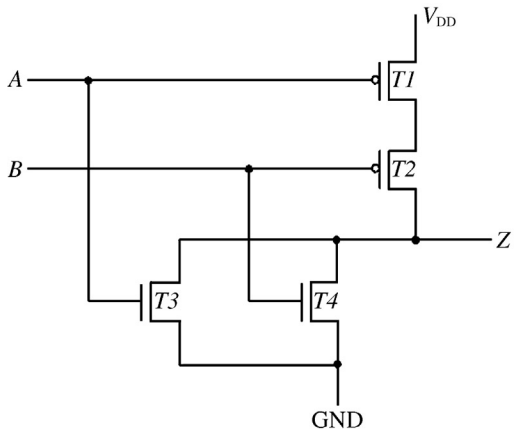

FIGURE 1.7: A two-input CMOS NOR gate.

TABLE 1.1: Truth table of two-input CMOS NOR gate with and without stuck-open fault

A	*B*	*Z*	*Z* (*A*s-op)	*Z* (*B*s-op)	*Z* (V_{DD}s-op)
0	0	1	1	1	Z_t
0	1	0	0	Z_t	0
1	0	0	Z_t	0	0
1	1	0	0	0	0

A stuck-open transistor fault like a feedback bridging fault can turn a combinational circuit into a sequential circuit [10]. Figure 1.7 shows a two-input CMOS NOR gate. A stuck-open fault causes the output to be connected neither to GND nor to V_{DD}. If, for example, transistor *T2* is open-circuited, then for input *AB*=00, the pull-up circuit will not be active and there will be no change in the output voltage. In fact, the output retains its previous logic state; however, the length of time the state is retained is determined by the leakage current at the output node.

Table 1.1 shows the truth table for the two-input CMOS NOR gate. The fault-free output is shown in column *Z*; the three columns to the right represent the outputs in presence of the three stuck-open (s-op) faults. The first, *As-op*, is caused by any input, drain, or source missing connection to the pull-down FET *T3*. The second, Bs-op, is caused by any input, drain, or source missing connection to the pull-down FET *T4*. The third, V_{DD}*s-op*, is caused by an open anywhere in the series, p-channel pull-up connection to *VDD*. The symbol Z_t is used to indicate that the output state retains the previous logic value.

1.3 BASIC CONCEPTS OF FAULT DETECTION

Fault detection in a logic circuit is carried out by applying a sequence of tests and observing the resulting outputs. A *test* is an input combination that specifies the expected response that a fault-free circuit should produce. If the observed response is different from the expected response, a fault is

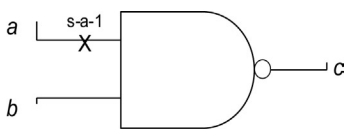

FIGURE 1.8: A NAND gate with a stuck-at-1 fault.