URC97020

Off-Line Testing for Bridge Faults in CMOS Domino Logic Circuits

K. Bennett ‘, P.K. Lala and F, Busaba
Department of Electrical Engineering
North Carolina A&T State University
Greensboro, NC 27411

Abstract

Bridge faults, especially in CMOS circuits, have unique characteristics which make them difficult to detect during testing. This paper presents a technique for detecting bridge faults which have an effect on the output of CMOS Domino logic circuits. The faults are modeled at the transistor level and this technique is based on analyzing the off-set of the function during off-line testing.

1. Introduction

Bridge faults are created during the design layout or manufacturing process of an integrated circuit (IC). It can result from two or more conducting paths placed too close together, the addition of extra conducting material, or insufficient insulating material. In order to minimize the number of bridge faults present in the IC, testing methods must be able to accurately detect the faults which have an effect on the normal operation of the circuit. Previous work in the area of bridge fault detection focused on static CMOS circuits.

Chess and Larrabee [1] have presented a method for generating test patterns for gate level bridge faults in static CMOS ICs. It focuses on the connection of two gate outputs which is modeled using a Fault Block and Primitive Bridge Function (PBF). The PBF represents the logic function or the behavior of the bridged components and is generated by determining whether the stimulation of the bridge fault occurs from the wire closest to the inputs of the bridged path or closest to the output.

Di and Jess [2] have presented a technique for modeling transistor level bridge faults in static CMOS circuits by evaluating the electrical behavior of the circuit and converting this behavior into logic boolean expressions called Faulty Boolean Expressions (FBE). This allows for the use of existing techniques for logic problems to determine test patterns. Gate-to-Drain type bridges are not able to be modeled.

Ferguson [3] discusses approaches for designing the physical layout of the static CMOS circuit in such a way to improve its testability of bridge faults. The three approaches are to design the circuit which reduces the number of faults, make the difficult-to-detect faults easier to detect by adding control and observation points, and make the difficult-to-detect faults unlikely to occur by considering the gate placement, circuit routing, and logic selection.

Chess, Roth, and Larrabee [4] have evaluated and compared various models used to represent bridge faults existing only between gate inputs and outputs. The different models assume that either the bridges cause wired “AND” or “OR’ behavior, that the circuit value at the fault node is

[1] Currently at Motorola SPS Division -911 Battle Bend Blvd , Austin Texas

represented by a boolean function, that the analog behavior created by the fault extends beyond the fault node, or that full analog simulation is performed.

In this paper, we propose a method for detecting transistor level bridge faults in CMOS Domino Logic circuits. Only the bridge faults which are realistic at the layout level are considered. Domino logic is a type of Dynamic logic in which the on-set is used in realizing the n-logic block. Figure 1 shows a genera! diagram of a domino logic circuit. An invertor is connected at the output to make it low during the precharge phase. The output node *f* is precharged to "O" when the clock is low. During, the evaluation, phase i .e. when clock is high, if the input pattern closes the path between ground and output node *f*, the output is pulled to " 1" otherwise it remains at "O".

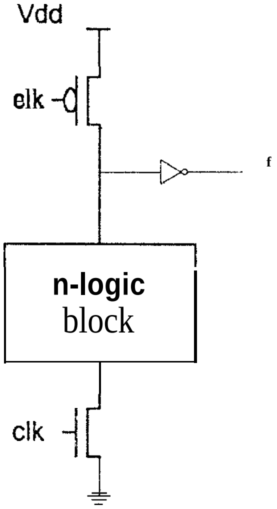

Figure 1. Domino logic circuit

Figure 2 shows a cascaded Domino logic circuit in which the output of one stage feeds the n-logic network of a subsequent stage, Only one clock is necessary for the precharge and the evaluati on phase. The number of stages in the cascade depends on whether the sequence can evaluate within the evaluation cycle.

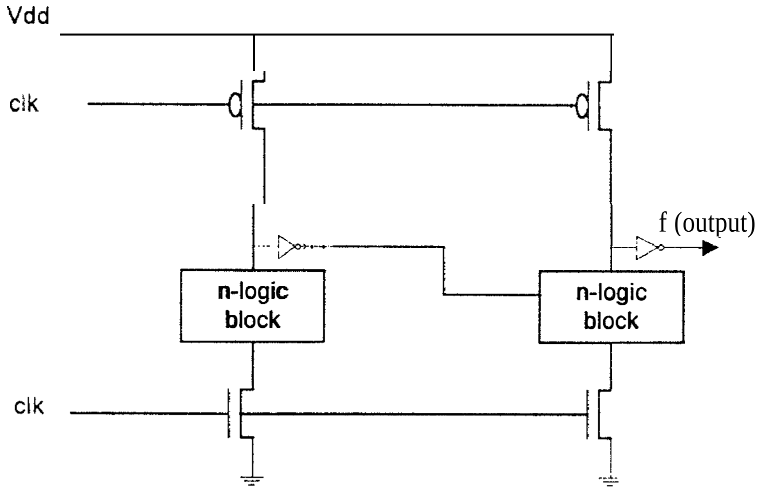

Figure 2. Cascaded domino logic circuit

The majority of the bridge faults in domino CMOS circuits occurs between the Drain/Source of one transistor and-the Drain/Source of another transistor, the Drain/Source of one transistor and the Output of the circuit, and the Drain of one transistor and Ground.

We do not consider the bridging between the Gate and Drain of the same transistor and between the gates of two parallel transistors .

2. Test Pattern Generation

As mentioned previously, we consider only bridge faults which are realistic at the layout level. The detection of these faults is based on the following lemmas.

Lemma 1 :

The output of a circuit containing bridge faults will generate more “1 ‘s” than the fault-free circuit.

Lemma 2:

The bridge faults in a circuit can be detected by applying only the original off-set of the circuit.

To illustrate the validity of the above lemmas, we consider two examples. Example 1 considers a single stage/single output domino logic circuit. Example 2 considers a cascaded domino logic circuit.

Example 1: Figure 3 shows the implementation of the function f = (B+ $\bar{A}$)C + A B. The dashed lines indicate bridge faults obtained by connecting the signal lines with metal.

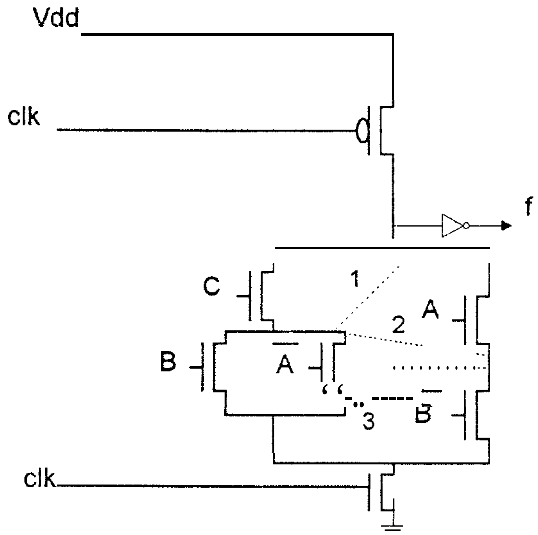

Figure 3. f= (B+ $\bar{A}$)C + A B

Table 1 shows the truth table of the fault-free and faulty functions. “A”, “3”, and “C” are the input variables and “f” represents the fault-free output. The outputs “fl_1”,“fl_2”, and “fl_3” represent the outputs corresponding to bridge faults #1,#2, and #3 respectively.

A	B	C	f	f,	f_2	f_3
0	0	0	0	1	0	0
0	0	1	1	1	1	1
0	1	0	0	1	0	0
0	1	1	1	1	1	1
1	0	0	1	1	1	1
1	0	1	1	1	1	1
1	1	0	0	1	1	1
1	1	1	1	1	1	1

Table 1

As indicated by Lemma 1, the input combinations for the faulty circuits produce more “1s” than the fault-free circuit. In order to detect a bridge fault, only the input combinations which produce a “0” for the fault-free circuit need to be applied as test patterns as discussed in Lemma 2.

Table 2 shows the input combinations that detect each bridge fault.

Bridge Fault Number	**Input** Combination

#1	A=0 B=0 C=0
	A=0 B=1 C=0
	A=1 B=1 C=0

#2	A=1 B=1 C=0
#3	A=1 B=1 C=0

Table 2

Example 2: Figure 4 shows the implementation of the function f = (A+ $\bar{B}$C)D using cascaded domino logic. A bridge fault in each stage of a cascaded circuit must be tested separately. In order to propagate the effect of the bridge fault to the final output, the inputs of the subsequent n-logic networks are set such that a “O” is produced at the outputs only if the output from the previous stage is a “O”, otherwise the output is a “1”. Hence, if the output of the stage under test produces a “1”, this value will be propagated through the subsequent stages to the final output.

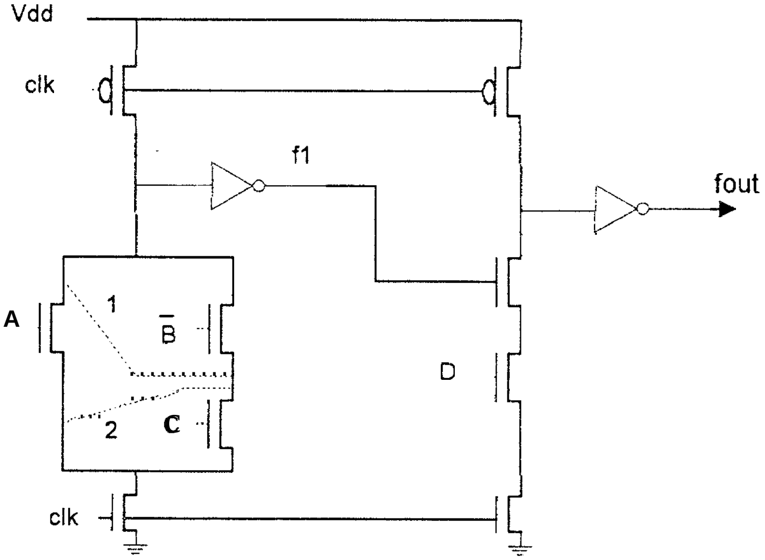

Figure 4. Cascaded domino circuit

In this example, we chose to test the first stage for the presence of a bridge fault. The value of “ 1” is assigned to the input variable “D” in order to allow for the propagation of the effects produced by bridge faults # 1 and #2. Since the first stage is being tested, only its function expression needs to be evaluated (f= A+ $\overline{B}$C). Table 3 shows the input combinations that detect each bridge fault in the first stage.

Bridge Fault Number	**Input** Combination
#1	A=0 B=1 C=1
#2	A=0 B=0 C=0

Table 3

In order to determine the test patterns for the overall circuit, input variable “D” must be added to the input combinations shown in Table 8. Hence, the input combinations which detect bridge faults # 1 and #2 arc.

A=0 B=1 C=1 D=1 and **A=0 B=0 C=0 D=1**

respectively.

5. Conclusion

We have presented-a method for detecting bridge faults in CMOS Domino logic circuits. This method is based on two lemmas discussed in section 3. This method is efficient in detecting all bridge faults except Gate-to-Drain and parallel Polysilicon bridge faults.

This technique also. applies to each stage of a cascaded circuit. The number of test patterns will be reduced corresponding to the reduction in the number of input variables used to represent the function of each stage. In Example 3, the number of test patterns was reduced from 11 to 3 because the overall function was separated into two stages. The function expression of the stage under test consisted of only 3 input variables instead of 4.

References

[1] B. Chess and T. Larrabee. Generating Test Patterns for Bridge Faults in CMOS ICs. *Proc. The European Design & Test Conference. EDAC, The European Conference on Design A utomation. ETC, Eropean Test Conference. EUROASIC, The European Event in ASIC Design,* pages 165-170. IEEE, 1994.

[2] C. Di and J.A.G. Jess. On CMOS Bridge Fault Modeling and Test Pattern Evaluation. *Digest of Papers. Eleventh Annual 1993 IEEE VLSI Test Symposium,* pages 116-118. IEEE, 1993.

[3] F, Ferguson. Physical Design for Testability for Bridges in CMOS Circuits. *Digest of Papers, Eleventh Annual 1993 IEEE VLSI Test* ***symposium,*** pages *290-295.* IEEE, 1993.

[4] B. Chess, C. Roth, and T. Larrabee. On Evaluating Competing Bridge Fault Models for CMOS ICs. *Proc. 12th IEEE VLSI Test Symposium,* pages 446-451. IEEE, 1994.

Acknowledgment: This work was supported partially by National Aeronautics and Space Administration and the Office of Naval Research under contracts ACE-48146 and ONR-N000 14-95-1-0947 respectively.