Switch Level Random Pattern Testability Analysis

Mehmet A. Cirit
Silicon Compiler Systems
Martinsville Rd, P.O. Box 16
Liberty Corner, New Jersey 07938

ABSTRACT

We present a new statistical, probabilistic algorithm for calculating controllability and observability for signal nets assuming the circuit can be described as a directed graph of unidirectional MOS switches. Application of the new algorithms to the testability analysis of CMOS circuits is described.

1 INTRODUCTION

In addition to aiding the designers in identifying the areas of poor testability in their circuits[1], testability analysis has been used to generate test vectors[2, 3, 4], and for the estimation of fault coverage of a vector set. In this analysis statistical and probabilistic algorithms play an important role. Although less precise, the applications of these methods in testability analysis is gaining wide spread acceptance because of the speed of the algorithms and the complexity of the circuits they can handle. These methods rely on the concepts of one and zero controllability which is the probability that a signal net will have a value of one or zero at a given time. One and zero observability of a line is the probability that a signal net having a value of one or zero will have a sensitized path to a primary output. In STAFAN[5], a statistical fault analyzer, controllabilities are calculated using a fault-free simulator. Observabilities are estimated assuming the underlying circuit is made up of AND, OR, and inverter primitives, taking special note of the feedback loops.

In this paper we are concerned with the so called random pattern testability problem along the lines of PREDICT[6], a testability analyzer in which the controllabilities are deduced from the controllabilities of the primary inputs to the circuit rather than calculated from signal statistics obtained by good circuit simulation as in the case of STAFAN. In order to correlate internal controllabilities to the primary inputs of the circuit, PREDICT decomposes the gate level circuit into circuit subgraphs called "supergates", which are collections of minimal subcircuits whose inputs are mutually independent. This approach has been reported to yield good fault coverage results[7], and is used for test vector generation[8] as well. The possibility that the number of vectors necessary to test some circuits can be significantly reduced by using patterns with non-uniform distributions, makes random pattern testability analysis a worthwhile effort[9, 10, 11].

In this paper we describe algorithms for calculating controllabilities and observabilities at the switch level, primitives being the MOS switches which conduct in a definite direction. The signal flow direction of each transistor can be found using some heuristic rules developed by Jouppi[12]. The calculation of controllabilities is then a matter of propagating the probabilities, modulated by the probability that each transistor is conducting, into internal nets of the circuit, starting from the primary inputs of the circuit. As the signals fan in or fan out, the usual probability combination rules are used to estimate the new controllability and observability.

In the next section the procedures used for assigning directions to MOS switches are discussed briefly. In Section III, we derive the controllability and in Section IV the observability relationships between the terminals of a directed MOS transistor. In Section V some implementation details are discussed.

The algorithms described here are implemented in LTIME, a CMOS timing analyzer. We have also applied these techniques to dynamic power dissipation analysis of CMOS circuits[13], as well as used in predicting chip level failure rates due to hot-electron effects.

2 DIRECTION ASSIGNMENT

The basic assumption of our approach is that each MOS transistor conducts signals in one definite direction which does not change as different stimuli are applied to the circuit. For most circuits this is a reasonable approximation. The signal flow direction is essential for finding paths in the circuit. The assignment of directions to transistors is done a la TV[14], using a set of heuristic rules[12]. The drain of a set transistor, which is assigned a direction, is a sink of signal flow, and its source is a source of signal flow. The most basic rule is that power and ground lines

must be a source of signal flow, which accounts for setting the directions of a majority of the transistors in a circuit. Another important rule is the analog of Kirchoff's current law, which applies when there is a single unset transistor at an internal net and all the others are sourcing or sinking the net. In this case the unset transistor should provide a path in or a path out. These rules are illustrated in Fig. 1 for a NAND gate. After applying power and ground rule, the Kirchoff rule is applied successively to nets 4 and 5.

Unfortunately these two rules are not enough to resolve direction ambiguities in general. We have developed some special rules which look for certain patterns like inverters, pseudo-NMOS structures, transmission gate adders, and Manchester carry chains. A direction is assigned to the transistor based on its particular usage. Kirchoff's rule is used repeatedly to propagate the direction information into the circuit. For most circuits these rules are sufficient for the identification of paths without much input from the user.

As a result of this direction finding analysis, each transistor in the circuit is assigned a definite direction. By following the direction of the transistors, all possible signal paths in the circuit can be traced.

The algorithms described here are implemented in LTIME which is a CMOS critical path analyzer like TV[14] and CRYSTAL[15]. LTIME assigns a signal flow direction to each transistor in the circuit to identify signal paths. Incorporation of a testability analyzer within its framework was a natural choice and straightforward, as all the ground work was already done.

3 CONTROLLABILITIES

Once the signal flow directions are determined, the controllabilities are found by propagating them in the same way signals are propagated. A transistor conducts the controllability at its source to its drain subject to the controllability of its gate. The probability of finding a signal at its drain is the intersection of the probability of finding a signal at its source and the probability of finding a signal at its gate which will turn it on. With this assumption, for an n-type transistor which conducts only when controlled by a high signal, the drain controllabilities are given by:

$$C_{1d} = C_{1g}\, C_{1s} \tag{3.1}$$

$$C_{0d} = C_{1g}\, C_{0s} \tag{3.2}$$

while for a p-type transistor which conducts only when controlled by a low signal, the following expressions can be written:

$$C_{1d} = C_{0g}\, C_{1s} \tag{3.3}$$

$$C_{0d} = C_{0g}\, C_{0s} \tag{3.4}$$

where C_{1d}, C_{1g}, C_{1s}, and C_{0d}, C_{0g}, C_{0s} are the drain, gate, and source controllabilities for high and low signal values respectively.

The controllability at a given signal net is the union of all the incoming controllabilities from various transistors conducting into the signal net. Assuming that all the incoming signals are independent of each other:

$$C = C_{n0} \cup C_{n1} \cup \cdots \tag{3.5}$$

where unions should be performed only on one or zero controllabilities.

If the input controllabilities are given, using the formulae listed above all the controllabilities can be propagated into the circuit. The algorithm which makes a breadth first search of all target nets is described elsewhere[13].

TABLE I

NODE	C	B
1	1/2, 1/2	1/4, 1
2	1/2, 1/2	1/4, 1
3	1/2, 1/2	1/4, 1
4	0, 1/2	1/4, 1/4
5	0, 1/4	1/2, 1/2
6	7/8, 1/8	1, 1

Figure 1 illustrates the propagation of controllabilities for a 3-input NAND gate. For simplicity we assume inputs have equal one and zero controllabilities of (0.5, 0.5). Starting from the ground net which has a controllability of (0,1), controllabilities are propagated upward along the pulldown chain using Eq. (3.2). At the output of the NAND gate a union operation is performed on all incoming controllabilities using Eq. (3.5). The resulting conrollabilities are shown in Table 1.

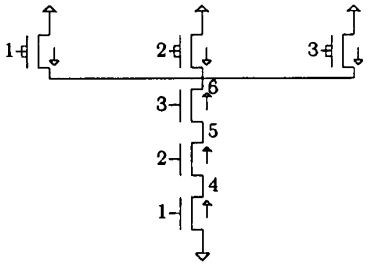

Figure 1. Signal Flow Directions for a CMOS NAND gate. Arrows indicate the signal flow direction.

4 OBSERVABILITIES

Propagation of observabilities proceeds in the opposite direction to the propagation of controllabilities. They propagate from the drain of the transistor to its source. Again the observability is modified by the controllability of the gate of the transistor. For an n-type transistor which conducts only when controlled by a high signal :

$$B_{1s} = C_{1g} B_{1d} \tag{4.1}$$

$$B_{0s} = C_{1g} B_{0d} \tag{4.2}$$

while for a p-type transistor which conducts only when controlled by a low signal:

$$B_{1s} = C_{0g} B_{1d} \tag{4.3}$$

$$B_{0s} = C_{0g} B_{0d}\,, \tag{4.4}$$

where B_{1d}, B_{1s}, and B_{0d}, B_{0s} are the drain and source controllabilities for high and low signal values respectively.

The observability of the gate can be expressed in terms of source controllability and drain observability. The gate observability is the sum of the probabilities of a signal at the source of the transistor being conducted to its drain and being observable there. For an n-device only a high at the gate could be observable:

$$B_{1g} = C_{0s} B_{0d} + C_{1s} B_{1d}\,, \tag{4.5}$$

for a p-device only a low signal at the gate is observable:

$$B_{0g} = C_{0s} B_{0d} + C_{1s} B_{1d}. \tag{4.6}$$

The observability at a given signal net is the union of all the incoming observability from various transistors conducting out of the signal net. Assuming they are all independent, observability is:

$$B = B_{n\,0} \cup B_{n\,1} \cup \cdots. \tag{4.7}$$

Fig. 1 illustrates the calculation of observabilities for the NAND gate of previous example. The controllabilities calculated at previous stage are used. Starting with an observability of (1,1) at net 6, the observabilities of the gates are calculated using Eq. (4.5) and (4.6), followed by a union operation as there are two paths to output from each gate. The observabilities of the internal nets 4,5 are found using Eq. (4.1), (4.2), (4.3), and (4.4).

5 IMPLEMENTATION

The algorithms and methods described above have been implemented in C, and incorporated into LTIME. LTIME provided the natural setting for developing controllability and observability calculations, as it has to find the signal flow directions in order to identify critical paths. It is capable of identifying the feedback loops, and for the clocked circuits, precharged nets that are very common for CMOS designs.

While propagating the controllabilities care should be taken to avoid controllabilities that are not independent of each other. The most common situation where this can happen in a CMOS circuit is the transmission gate. In this case the same signal which fanned out into n- and p- transistors of the transmission gate fans into the same net. Another important case is the feedback loop where the original signal should be distinguished from the feed back signal as they are not independent of each other. We first identify all the possible feedback loops in the circuit. The transistors which could contribute to a loop are classified into primary, which actually inject a signal, and secondary classes which inherit their signals from others. While calculating the controllabilities only the contributions of the primary inputs are considered. We do not iterate on the feedback loops because the regenerated signal is not independent of the original signal.

TABLE II

NODE	C	B
1	1/2, 1/2	1/2, 1
2	1/2, 1/2	1/2, 1
3	0, 1	1/4, 1/4
4	0, 1/2	1/2, 1/2
5	3/4, 1/4	1, 1

In the calculation of the controllabilities, the precharged nets should be handled carefully. LTIME, being a timing analyzer, is able to propagate the primary clock signals into the circuit as well as to identify the precharged nets. Figure 2 illustrates a simple precharged output. Output is always high unless discharged. In addition the data inputs should be synchronized with the evaluate signal ϕ, that is, whenever the data inputs are ready they should be enabled. Controllabilities and observabilities are given in table II. We assume that net 3 has controllability of (0,1). Controllability of net 5 is calculated assuming that it is high unless discharged. Observabilities listed in Table II are calculated assuming that the precharged output node has the observability (1,1). Low observabilities of the gate inputs are the same as the high observability of the output. Output controllability is calculated assuming the drain of the evaluate transistor has a 0-controllability of one. This essentially reduces the circuit to a NAND gate which is what it is functionally.

In the calculation of observabilities the same precautions are taken. For feed back loops the contributions of only the primary outputs are considered. We do not iterate on the observability as it is done in STAFAN. In the case of precharged nets, the high observability of the precharged net is assigned directly to the low observabilities of the data inputs.

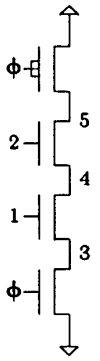

Figure 2. Precharged circuit example. Controllabilities and observabilities are given in Table II.

6 CONCLUSIONS

We described in this paper a method for calculating pattern independent controllabilities and observabilities of transistor level circuits. The method requires that a definite direction be assigned to each transistor in the circuit. The assigned directions can then be used for propagating the controllabilities from the inputs to the internal nets and for propagating the observabilities from outputs back to the internal nets. We derive a set of relations correlating the gate, source, and drain observabilities and controllabilities of n and p-type transistors. Some implementation issues are discussed.

The advantage of our method is in its simplicity and the generality which comes with it. It is based only on one primitive element, the MOS transistor. However, MOS transistors are assumed to be strictly uni-directional which is not a good assumption in certain cases. Probably another advantage is the speed of the algorithms, because of the reduced complexity. However, we do not have any comparative data to justify this asserttion.

The controllability and observability analysis at switch level has its own special applications where gate level analysis may not be adequate. We have used some of the methods described here for power dissipation analysis, in analyzing various reliability problems related to transistor switching rates etc..

I would like to thank C. Christensen, R. Gordon and D. Adler who read and criticized the initial manuscript of this paper.

References

1. L. H. Goldstein, "Controllability/Observability Analysis of Digital Circuits," *IEEE Trans. Circuits and Systems* **CAS-26**(9) pp. 685-693 (Sept 1979).
2. V. D. Agrawal, S. C. Seth, and C. C. Chuang, "Probabilistically Guided Test Generation," pp. 687-690 in *Proc. Int. Symp. on Circuits and Systems*, (June 5-7, 1985).
3. F. C. Wang and A. C. Hung, "A Method for Test Generation Directly from Testability Analysis," in *International Test Conference*, , Philadelphia, PA (November 19-21, 1985).
4. E. Trischler and M. Schulz, "Application of Testability Analysis to ATPG: Methods and Experimental Results," pp. 691-694 in *Proc. Int. Symp. on Circuits and Systems*, , Kyoto, Japan (June 5-7, 1985).
5. S. K. Jain and V. D. Agrawal, "STAFAN: An Alternative to Fault Simulation," pp. 18-23 in *Proc. 21th DAC*, (June 1984).
6. S. C. Seth, L. Pan, and V. D. Agrawal, "PREDICT - Probabilistic Estimation of Digital Circuit Testability," pp. 220-225 in *Proc. Int. Symp. on Fault Tolerant Computing*, (June 19-21, 1985).
7. S. H. Seth, "Predicting Fault Coverage From Probabilistic Testability," pp. 803-805 in *Proc. International Test Conference*, (1985).
8. R. Lisanke, F. Brglez, A. de Geus, and D. Gregory, "Testability-Driven Random Pattern Generation," in *Proc. ICCAD*, (1986).
9. K. P. Parker, "Adaptive Random Test Generation," *J. Design Automation and Fault-Tolerant Computing*, pp. 62-83 (Oct. 1976).
10. P. Agrawal and V. D. Agrawal, "On Monte Carlo Testing of Logic Tree Networks," *IEEE Trans. on Computers* **Vol. C-25** pp. 664-667 (June 1976).
11. H. Wunderlich, "PROTEST: A Tool for Probabilistic Testability Analysis," pp. 204-211 in *Proc. 22th DAC*, (1985).
12. N. P. Jouppi, "Derivation of Signal Flow Direction in MOS VLSI," *IEEE Trans. CAD* **CAD-6** pp. 480-489 (May 1987).
13. M. A. Cirit, "Estimating Dynamic Power Consumption of CMOS Circuits," pp. 534-537 in *Proc. ICCAD*, (1987).
14. N. P. Jouppi, "Timing Analysis for NMOS VLSI," pp. 411-418 in *Proc. 20th DAC*, (1983).
15. J. K. Ousterhout, "Crystal: A Timing Analyzer for NMOS VLSI Circuits," pp. 57-59 in *Proc. Third Caltech VLSI Conference*, (1983).