A Correlation Matrix Method of Clock Partitioning for Sequential Circuit Testability

Yong Chang Kim
Dept. of Elec. and Comp. Engr
University of Wisconsin-Madison
1415 Engineering Dr.
Madison, WI 53706
kimy@ece.wisc.edu

Vishwani D. Agrawal
Bell Labs
Lucent Technologies
700 Mountain Ave.
Murray Hill, NJ 07974
va@research.bell-labs.com

Kewal K. Saluja
Dept. of Elec. and Comp. Engr
University of Wisconsin-Madison
1415 Engineering Dr.
Madison, WI 53706
saluja@engr.wisc.edu

Abstract

We propose a method of partitioning the set of all flip-flops in a circuit for multiple clock testing. In the multiple clock testing, flip-flops are partitioned into different groups and each group of flip-flops has an independent clock control. In our method, we use a test generator assuming an independent clock control for each flip-flop. We than determine correlation between clock activity for all pairs of flip-flops. This information is than used to an optimal or near optimal partition of flip-flops in. Through experiments, we demonstrate that our partitioning method increases fault coverage and reduces test length with almost no hardware overhead or performance penalty.

1 Introduction

As complexity of VLSI circuits is growing, test generation for sequential circuits is becoming increasingly difficult and time consuming. This problem can be alleviated by using design for testability (DFT) techniques. The most popular DFT techniques are full-scan and partial-scan methods. In these methods often a large portion of flip-flops is needed to be scanned to provide adequate or high fault coverage. As a result, there are some major drawbacks in scan approaches. These are, expensive hardware overhead, long test application time, and for partial scan long test generation time of sequential ATPG. Partial scan also requires an additional scan clock control besides the normal scan hardware.

A somewhat different DFT technique proposed recently, which makes use of individual clock control of sets of flip-flops (FFs) and is known as multiple clock (MC) method, as oppose to the conventional single clock (SC) method. Agrawal et al. [1] proposed a method to partition the FFs into two groups that are controlled by independent clocks during the test. Cycle and loop breaking was used as FF partitioning method. They showed that, in general, two clocks were not sufficient for breaking all feedbacks. Einspahr et al. [4, 5] extended the idea further to the multiple groups and FFs in a loop are partitioned into two or more groups. Similarly, Baeg and Rogers [2] proposed partitioning of FFs to different groups, each with its own clock control, but all FFs within a loop are kept in the same group. Test generation is somewhat simplified but if the circuit has many self-loops, the number of partitions increases. Also the partitioning of FFs is difficult, if not impossible, when there is a loop covering all FFs in the circuit. Rajan et al. [8] introduced a method of converting hard to reach states to easier to reach states using a clock transformation, which uses line probabilities to determine if certain states (FFs to be exact) are hard to reach. Next, if it is determined that hard to reach states are really needed for test generation, then the corresponding FFs are placed in different groups. Only a pair of FFs is considered at a time for making hard to reach states easier to reach by individually clocking the FFs. However, this method is only applicable to hard states given by a pair of FFs and does not consider three or more FFs which may cause hard states. Another notable work [7] is on synthesis of state machines for dual clock control. However, such methods will not scale to large and existing sequential circuits. In [3], clock control is used to reduce the minimum number of transitions between pairs of states. However, this method requires a complete state description and hence only applicable to circuits with a small number of FFs.

In this paper, we propose a new partitioning scheme for the multiple clock DFT technique. Unlike other grouping schemes, which rely on a static structural analysis, our approach determines the grouping dy-

0-7695-0104-4/99 $10.00

namically using a test generator and no structural analysis is required.

2 Multiple Clock DFT Approach

To illustrate how an MC DFT technique can be applied to a sequential circuit, consider a sequential circuit shown in Figure 1. This circuit cannot be initialized using either three values or symbolic simulation by the SC control. However, the MC control can initialize this circuit by first setting FF2 to 0 using $PI : a = 0$ and then setting FF1 to 1 by independently clocking FF1 using $CLK1$ and setting $PI : a = 1$.

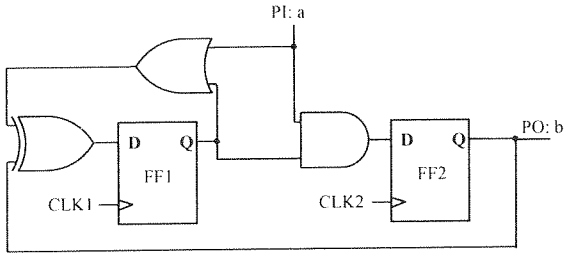

Figure 1: An Example Circuit

As shown in the above example, MC requires that we clock FFs individually. In order to use an existing test generator to apply MC control as DFT method, we develop a *mux-multi-clock* model as shown in Figure 2. In this model, a 2:1 multiplexer is inserted in front of each FF. When $PI{:}Select_1$ is set to 1, each CLK pulse will update the FF1, whereas FF1 will hold its content when $PI{:}Select_1$ is set to 0. Similarly, $PI{:}Select_2$ can be used to clock FF2 or to hold its value.

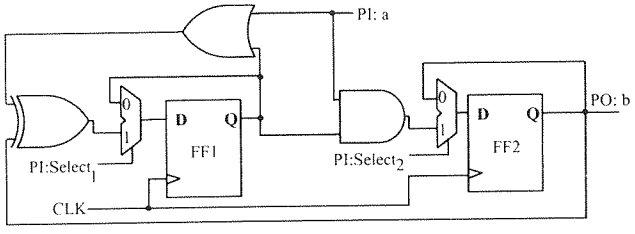

Figure 2: An MC Control Circuit using Mux-Multi-clock Model

In order to use a DFT technique to its fullest potential, effects and benefits of the DFT technique must be fully investigated. In this section, we present theorems and comment on MC control DFT method for understanding the effects and benefits of MC.

Theorem 1 *If the MC control cannot initialize a circuit, then the SC control also cannot initialize the circuit.*

Proof 1 *We can prove Theorem 1 by proving its contrapositive, which is* if the SC control can initialize the circuit, then the MC control can also initialize the circuit. *Since k-clock control circuit (where $k > 1$) can be converted to a single-clock control circuit by tieing all k clocks together, it can mimic the behavior of the corresponding SC control circuit.*

Theorem 2 *The fault coverage of the MC control circuit need never be less than that of the SC control circuit.*

Proof 2 *Suppose we have an n-clock control circuit, where $n > 1$. If we tie n individually controlled clocks together, then it will form a single-clock control circuit. Thus, any vector sequence that is a test for some fault in the SC circuit is also possible in the MC circuit. Consequently the MC control circuit can achieve same fault coverage as the SC circuit. This establishes MC control circuits can have no less fault coverage than that of SC control circuit.*

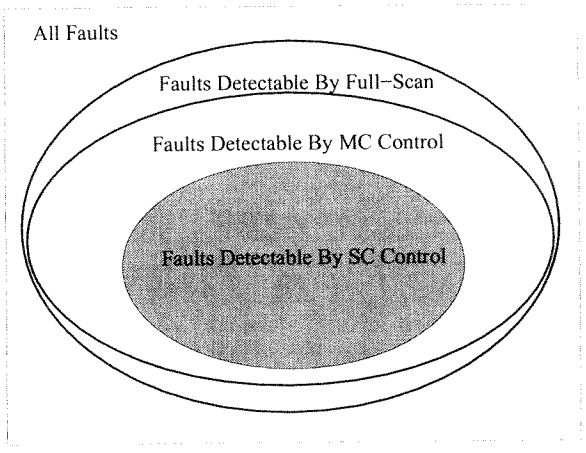

Figure 3: Fault Coverage Relationship

It is important to note that the converse of Theorem 1 is not true. As shown in Figure 1 some circuits that are not initializable by SC control can be initialized under MC control. However, there are some circuits which cannot be initialized by either MC and SC control. Figure 3 shows the fault coverage relationship between MC control, SC control and full-scan method. In our *Results* section, we will show such cases using

ISCAS 89 benchmark circuits. In general, MC control can reset more circuits than SC control, and MC control has a better controllability and/or observability than SC control. This is likely to yield a higher fault coverage for MC control circuits than SC control circuits as shown in Figure 3.

3 Correlation Matrix Method

Our grouping method is based on deriving a correlation matrix using the the results of test generation on a *mux-multi-clock* modified circuit. As shown in Figure 2, circuit is modified to be k individually controlled clock, where k is total number of FFs in the circuit. We use a test generator on a k mux-multi-clock circuit to generate the test. From the generated test sequence, let us consider a pair of FFs. We notice that for FFs i and j, $PI{:}Select_i$ and $PI{:}Select_j$ can either equal or not equal. We define the *Correlation* between FF_i and FF_j as $\frac{number\ of\ times\ PI:Select_i\ equal\ PI:Select_j}{length\ of\ the\ test\ sequence}$. A high correlation (*correlation* close to 1) indicates two FFs were clocked/not-clocked together, and low correlation (*correlation* close to 0) indicates two FFs were not simultaneously clocked/not-clocked. There are three phases in the proposed approach.

During the first phase, the circuit is modified using *mux-multi-clock* model and the test generator finds the test on the modified circuit. The modification allows the test generator to use an individual clock for each FF. In the second phase, the correlation matrix is generated and processed to determine the grouping of FFs. The correlation matrix describes which FFs were clocked together in the first phase. The idea is that if FFs were clocked together then they can be placed in the same group. During the final phase, the original circuit is modified to allow individual clock controls to the FF groups obtained from phase 2. Then, the test generator is used on the newly modified circuit to generate the test. Since our goal is to keep the hardware overhead minimum but achieve a higher fault coverage, we limit our focus to the use of only two clock groupings in this paper. However, the correlation matrix method described here can also be used for generating more than two clock groups.

4 Algorithm

To understand our method, consider an ISCAS benchmark circuit s1488. It has six FFs. In the first phase, s1488 is modified using the *mux-multi-clock* model. The *PI:Select* of each multiplexer become primary inputs and are added to the primary input list. Since there are six FFs in s1488, a total of six 2:1 multiplexers are added to the circuit, one for each FF. Then the ATPG generates the test vectors on the modified circuit. For our experiment, we used FASTEST [6] to generate the test on modified circuit. It is important to note that the modification during the first phase is not the modification required for the actual DFT implementation. Rather these modifications are required only in an ATPG model used to generate the correlation matrix.

In the second phase, using the test vector generated from the phase one, we examine the six *PI:Select* signals and generate the matrix which indicates matching of each *PI:Select* signal to other *PI:Select* signals. If the *PI:Select* signals for two FFs are a perfect match (i.e., two FFs were clocked/not-clocked at the same time), then the corresponding matrix entry should be written as 1.0 to signify 100% matching. On the other hand, if the *PI:Select* signal is completely different from the other *PI:Select* signal,then the corresponding entry in the correlation matrix is 0 to signify 0% matching. Using the correlation matrix, one can determine which FFs were more frequently clocked/not-clocked during the first phase. A simple algorithm based on greedy method, described below is used to find most desirable grouping.

Step 1 *Sort the correlation matrix, so that the top of list contains highest correlation (i.e., most matched pair), and the bottom of the list contains the lowerest correlation (i.e., least matched pair).*

$$W_{Local} = \frac{Sum\ of\ correlations\ in\ the\ group}{Total\ number\ of\ correlations\ on\ the\ group}$$
$$W_{Global} = \frac{Sum\ of\ all\ W_{Local}}{Total\ number\ of\ used\ groups}$$

Step 2 *Take the pair of FFs from the top of the list, enumerate all possible partitions and compute the weights, W_{Local} and W_{Global}, of all possible partitions.*

Step 3 *Select the partition which results in largest W_{Global} and repeat the Step 2 until all FFs are partitioned.* Obviously, FFs with a high correlation entry should go into the same group, where as FFs with a low correlation entries should be separated and placed in two different groups. We used several heuristics to keep the number of FFs in each group evenly distributed.

Finally in phase three, the original circuit is modified to have individual clock controls according to the grouping obtained from the second phase. To accomplish this, we simply tie all *PI:Select* inputs within the group to one PI. Then the test generator is used to find the test for the newly modified circuit. FASTEST is used to generate the tests for Step 1 and Step 3.

5 Results

Table 1: Fault Coverages of Benchmark Circuits

Circuit	s208	s298	s344	s349	s382	s386
# of FFs	8	14	15	15	21	6
SC(Cov)	8.29	82.5	96.2	95.7	20.0	77.6
MC(Cov)	100	90.3	96.5	96.0	85.2	99.0
G/M/N	G	G	N	M	G	G

Circuit	s400	s420	s444	s510	s526	s641
# of FFs	21	16	21	6	21	19
SC(Cov)	68.4	6.15	58.0	0.00	11.9	86.3
MC(Cov)	91.3	6.15	85.9	0.00	80.2	86.3
G/M/N	G	N	G	N	G	N

Circuit	s713	s820	s832	s838	s953
# of FFs	19	5	5	32	29
SC(Cov)	81.8	52.0	53.4	5.16	7.97
MC(Cov)	81.8	93.4	92.1	5.16	7.97
G/M/N	N	G	G	N	N

Circuit	s1196	s1238	s1423	s1488	s1494
# of FFs	18	18	74	6	6
SC(Cov)	99.8	94.7	82.9	70.3	70.2
MC(Cov)	100	94.7	87.1	99.7	99.1
G/M/N	M	N	M	G	G

Table 1 shows the results of using our method for ISCAS 89 benchmark circuits. The number of FFs are shown for each circuit, *SC(Cov)* represent the fault coverage of single clock circuit (unmodified), *MC(Cov)* denotes fault coverage of k-mux-multi-clock modified circuits, where k is the number of the FFs in the circuit. When comparing *SC(Cov)* and *MC(Cov)* of the Table 1, we can classify circuits to three classes: one that benefited from MC, one that marginally benefited from MC and one that did not benefit from it at all. We denote each class as G, M and N, representing a Good candidate, Marginal and Not a good candidate, respectively. There are 11 good candidates on Table 1. Table 2 shows the fault coverages of 2-clocks on 11 good candidate circuits. As we expected, all circuits show increase in fault coverages, and for some circuit, the gain can be substantially high. For certain circuits, such as s400 and s526, gain is noticeable using only 2-clocks, but use of three or more clocks can increase the fault coverages even further. Some circuits such as s510 showed no improvement and it is a case where MC control as well as SC control cannot initialize the circuit.

6 Conclusion

Several ISCAS 89 benchmark circuits were analyzed and it was confirmed that our method of clock grouping provides a significant increase in fault coverage and large reduction in ATPG time over single clock designs. We believe our grouping is near optimum in the sense that using a given ATPG, no other grouping would yield a better fault coverage. One of the most important thing to notice is that we have only shown improvements of MC DFT method using the FASTEST. Since the FASTEST is a general purpose ATPG and not a specialized MC ATPG, the improvement of MC we have shown may not be the maximum achievable improvements. If we were to design a specialized MC ATPG, we believe further improvements in the already promising results are possible.

Table 2: Fault Coverages Using 2-clocks

Circuits	s208	s298	s382	s386	s400	s444
2-Clocks(Cov)	89.4	89.9	76.7	92.2	76.2	78.1
Increase(%)	81.1	7.4	56.7	14.6	7.8	20.1

Circuit	s526	s820	s832	s1488	s1494
2-Clocks(Cov)	35.8	86.7	81.4	95.8	95.4
Increase(%)	23.9	34.7	28.0	25.5	25.2

References

[1] V. D. Agrawal, S. C. Seth, and J. S. Deogun, "Design for Testability and Test Generation with two clocks," in *Proc. 4th International Symposium on VLSI Design*, January 1991, pp. 44–51.

[2] S. H. Baeg and W. A. Rogers, "Hybrid Design for Testability Combining Scan and Clock Line Control for Test Generation," in *Proc. Int. Test Conf.*, October 1994, pp. 340–349.

[3] K. L. Einspahr, S. K. Mehta, and S. C. Seth, "Synthesis of Sequential Circuits with Clock Control to Improve Testability," in *Proc. 7th IEEE Asian Test Symposium*, Dec. 1998, pp. 472–477.

[4] K. L. Einspahr, S. C. Seth, and V. D. Agrawal, "Clock Partitioning for Testability," in *Proc. 3rd IEEE Great Lakes Symposium on VLSI*, March 1993, pp. 42–46.

[5] K. L. Einspahr, S. C. Seth, and V. D. Agrawal, "Improving Circuit Testability by Clock Control," in *Proc. 6th IEEE Great Lakes Symposium on VLSI*, March 1996, pp. 288–293.

[6] T. P. Kelsey, K. K. Saluja, and S. Y. Lee, "An Effiecient Algorithm for Sequential Circuit Test Generation," *IEEE Trans. on Computers*, vol. 42, pp. 1361–1371, Nov. 1993.

[7] S. K. Mehta, S. C. Seth, and K. L. Einspahr, "Synthesis for Testability by Two-Clock Controls," in *Proc. 10th International Conference on VLSI Design*, January 1997, pp. 279–283.

[8] K. B. Rajan, D. E. Long, and M. Abromovici, "Increasing Testability by Clock Transformation," in *Proc. 14th IEEE VLSI Test Symposium*, April 1996, pp. 224–231.