

Partial Reset and Scan for Flip-Flops Based on States Requirement for Test Generation

Hsing-Chung Liang
Dep. of Electronics Engineering
Van Nung Inst. of Tech.
Chung-Li, Taiwan, R.O.C.

Chung Len Lee
Dep. of Electronics Engineering
National Chiao Tung University
Hsin-Chu, Taiwan, R.O.C.

Jwu E Chen
Dep. of Electrical Engineering
Chung Hua University
Hsin-Chu, Taiwan, R.O.C.

Abstract

This paper proposes a method to select flip-flops for partial reset and/or partial scan for sequential circuits to increase their testability. The method gives weights for flip-flops for consideration for partial reset and/or scan based on information on required states for activating faults and the number of faults which propagate to flip-flops, which are obtained during test generation. Since the above information offers the reasons causing the untestable and/or hard-to-detect faults, the method is very efficient in locating flip-flops for partial reset and/or scan to ease test generation task. Experiments showed that this method selected less number of flip-flops for partial reset and scan while produced more testable circuits for benchmark circuits.

1. Introduction

Test generation for sequential circuits becomes more and more difficult when the circuits grow larger for design necessity. This makes design for testability(DFT) techniques[1,2] become greatly important because they can reduce much effort for test generation. In DFT methods, *full scan* increases the testability massively by transforming the sequential circuit into a combinational one during test mode. The trade-off is that it adds much area and delay penalty to the circuit and needs lengthy time for applying test patterns. *Partial scan*[3-17] is a good alternative for full scan because it links only a partial set of flip-flops into a scan chain. It can improve the circuit's testability at the cost of less change in hardware and delay and shorter test application time. However, both scan methods have the disadvantages of three additional I/O pins, extra routing lines, and more complex flip-flops for scan. Simultaneously, in timing affair, they reduce the clock speed of circuit and need additional time to apply test patterns.

Hardware reset is another considerable DFT technique since it increases the circuit's controllability. It is superior to scan design in that it needs only one additional input pin, less routing lines, and less change to flip-flops. In addition, it affects much less on clock speed and makes the circuit be able to be tested at speed, which allows testing for delay faults. But the circuits with reset may be still harder for test generation as compared to those with scan design.

In hardware reset design, *full reset*[18,19] was initially proposed to ease the initialization in sequential circuits by setting all flop-flops to some definite state, e.g. all be zero, when the reset signal is applied. But, for sequential circuits, not all the flip-flops are always difficult to be initialized. Cheng[20] suggested that a selected subset of flip-flops should be reset if they are difficult for initialization, which is referred to as *partial reset*. Pomeranz[21] used state stable to select flip-flops for partial reset and showed the potential advantages of partial reset to increase the fault coverage as well as decrease the test length. Obviously, a circuit with partial reset will not return to a unique state when its reset signal is applied, which is different from that with full reset. In addition, full reset can only be applied at the start of a test sequence for not destroying the effect of previous vectors, but partial reset can be applied in the middle of a test sequence if needed. Therefore partial reset is likely to achieve higher testability than full reset.[22,24]

In the methods of partial reset, Mathew[22] selected flip-flops by breaking cycles and chose the reset values with those having lower controllability calculated by SCOAP[23]. Abramovici[24] improved the results by calculating the cost function for a sensitivity analysis to rank the flip-flops. The authors also suggested a combination of partial reset and partial scan to obtain better results. Recently, Parikh[25] combined partial scan, partial reset, and partial observation[26,27] into one unified technique. They hoped to make use of the benefits and compensate the disadvantages of each method to achieve higher testability with less cost in hardware and

test application time.

In this paper, we propose a new method to select flip-flops for partial reset and partial scan designs. Test generation results and invalid states information[28] are analyzed. For each selected fault of a circuit, the initial test generation provides both the required states and the flip-flops to which the fault can be propagated in the activation time frame. The obtained required states and invalid states are used to calculate the selection weights of reset 1 and 0 for each flip-flop. Adding these weights to the number of faults that can be propagated to this flip-flop gives its selection weight for partial scan. According to the selection weights, benchmark circuits are redesigned with reset and scan and run on test generation. Comparing the results to those of [25] shows that our method can select less number of flip-flops to generate more testable circuits.

2. Flip-flop selection

Both partial reset and partial scan need a proper selection of flip-flops to improve circuit's testability as much as possible with little cost. To select the flip-flops, we analyze the information obtained during an initial test generation. Simultaneously, because the occupied percentage of valid states, i.e., *density of encoding*, affects the testability of a sequential circuit[29,30], the invalid states obtained in [28] are also considered to help generate the selecting sequence of flip-flops.

2.1 Initial test generation

At first, some faults of a circuit are targeted. They can be the total faults or the hard-to-detect faults obtained after running an initial test generation with small time or backtrack limit. During the test generation, we collect the information about activating and propagating each fault in its *activation time frame*, which is the clock time when the fault is excited. The information includes the required state to activate and propagate the fault and the destination flip-flops to which the fault can be propagated in that time frame. After processing all the target faults, we group the required states into a set RS and, in these states, count the number of times of 1 and 0 occurrence for each flip-flop, which are defined respectively as $NO_1(n)$ and $NO_0(n)$ for flip-flop n . In addition, the number of faults that can be propagated to flip-flop n in the activation time frame is also calculated and denoted as $NF(n)$. These parameters will be used to determine the selection priority of flip-flops for reset and scan.

Example 1: Assume some hard-to-detect faults of a circuit with four flip-flops need the following required states in the activation time frames, which are represented in

the tri-nary form.

$$RS = \{xx10, x100, 1000, xx11, 11x0, 1111, x111, x010\}$$

For each required state, the MSB represents the value on flip-flop 4 and the LSB represents the value on flip-flop 1. The value 'x' means that the corresponding flip-flop don't need a definite value in the activation time frame. It is easy to count the number of times of 1 and 0 occurrence for each flip-flop, i.e. $NO_1(4)=3$, $NO_0(4)=0$, $NO_1(3)=4$, $NO_0(3)=2$, etc. ■

2.2 Invalid states covering

In a sequential circuit, the *invalid states* are the states that the circuit cannot reach under whatever the input sequences. They usually obstruct the test generation using backward justification because they cannot be justified. In other words, they are responsible for the untestable and hard-to-detect faults for sequential circuit test generation. The flip-flops associated with these states are therefore good candidates for partial reset and/or partial scan selection when the testability of the circuit is sought to be improved.

In Ref. [28], a method to identify the complete or partial set of invalid states needed for test generation was proposed. The invalid states found by that method are used to help select flip-flops for partial reset and scan design.

Let a circuit have M invalid states which are expressed in cubes and let IV_m be the m th invalid state cube. Let the number of required states, which are the states needed to be justified in order to obtain tests for faults, that cover IV_m be denoted to be NS_m . A required state RS_i is said to cover an invalid state cube IV_m if it has a 0(or 1) in all coordinate positions in which IV_m has a 0(or 1), e.g. (11x0) covers (1xx0). Obviously, the more the required states cover an invalid state, the more difficult a test is to be generated. The flip-flops associated with this invalid state are good candidates for partial reset and scan selection.

Example 2: Assume the circuit given in example 1 has the following three invalid state cubes: $\{(xx1x), (x1xx), (1xx0)\}$, i.e., $IV_1=(xx1x)$, $IV_2=(x1xx)$, and $IV_3=(1xx0)$. From the required states RS given in example 1, it is easy to count the numbers of required states covering each invalid state cube: $NS_1=5$, $NS_2=4$, and $NS_3=2$. ■

2.3 Selection weight

The importance of a flip-flop for the reset or scan design is to be quantified. Though NS_m accounts for the number of required states covering IV_m , it still does not

consider the number of flip-flops having definite value in IV_m . In addition, a flip-flop with one definite value may appear in more than one invalid state cube. To fairly consider the numbers obtained so far for the selection of flip-flops, we desire to calculate them in the following way.

Let K_m be the number of flip-flops with definite value in IV_m and let $Q_m^1(n)$ and $Q_m^0(n)$ be the weight of flip-flop n to be 1 and 0 in IV_m , respectively. $Q_m^1(n)$ and $Q_m^0(n)$ denote the importance of flip-flop n contributed to IV_m and are calculated by NS_m and K_m as follows.

$$Q_m^1(n) = \begin{cases} \frac{NS_m}{K_m} & \text{if flip-flop } n \text{ has value 1 in } IV_m; \\ 0 & \text{otherwise.} \end{cases}$$

$$Q_m^0(n) = \begin{cases} \frac{NS_m}{K_m} & \text{if flip-flop } n \text{ has value 0 in } IV_m; \\ 0 & \text{otherwise.} \end{cases}$$

Example 3: In example 1 and 2, the invalid states set is $\{(xx1x), (x1xx), (1xx0)\}$. Therefore $K_1=1=K_2$, $K_3=2$. Since $NS_1=5$, $NS_2=4$, and $NS_3=2$, it implies that $Q_1^1(2)=5/1=5$, $Q_2^1(3)=4/1=4$, $Q_3^1(4)=2/2=1=Q_3^0(1)$. ■

For partial reset and partial scan, the values $NO_1(n)$ and $NO_0(n)$ obtained in the initial test generation are added to the summation of $Q_m^1(n)$ and $Q_m^0(n)$, respectively to derive the selection weights for each flip-flop. Let the circuit have M invalid state cubes. The selection weights of flip-flop n to be reset to 1 and 0 are represented by

$$PR_1(n) = \sum_{m=1}^M Q_m^1(n) + NO_1(n) \text{ and}$$

$$PR_0(n) = \sum_{m=1}^M Q_m^0(n) + NO_0(n), \text{ respectively.}$$

It is obvious that we can add coefficients to the two items in the right side for the purpose of insisting the affection of required states or invalid states. But they are considered equally important here for simplicity.

Example 4: For the circuit in the previous examples, the PR values of its four flip-flops are calculated as

$$\begin{aligned} PR_1(4) &= Q_3^1(4) + NO_1(4) = 1 + 3 = 4 \\ PR_0(4) &= 0 \\ PR_1(3) &= Q_2^1(3) + NO_1(3) = 4 + 4 = 8 \\ PR_0(3) &= NO_0(3) = 2 \\ PR_1(2) &= Q_1^1(2) + NO_1(2) = 5 + 5 = 10 \\ PR_0(2) &= NO_0(2) = 2 \\ PR_1(1) &= NO_1(1) = 3 \\ PR_0(1) &= Q_3^0(1) + NO_0(1) = 1 + 5 = 6 \end{aligned} \quad \blacksquare$$

According to the PR values, we can arrange the

selection order of flip-flops for partial reset. For the above example, the first selection is flip-flop 2 with reset 1 value, then the next is flip-flop 3 with reset 1 value, etc. It is mentioned that the testability of the circuit does not linearly increase with numbers of flip-flops selected for partial reset[24,25]. Though not knowing the optimum number of flip-flops selected for partial reset, it would be better to select a few of them since the circuit will then be able to reset to more states.

The selection weight of partial scan for each flip-flop can be directly calculated by adding the selection weights of reset 1 and 0 for that flip-flop. However, in addition to increasing controllability, partial scan also enhance the observability of the circuit. The value $NF(n)$, i.e. the number of faults that can be propagated to flip-flop n , being obtained in the initial test generation is therefore added to give the selection weight of flip-flop n for partial scan.

$$PS(n) = PR_1(n) + PR_0(n) + NF(n)$$

Similarly, each item in the right side of the above equation can be put a coefficient if needed. The PS values give a selection order of flip-flops for partial scan design, like those PR values for partial reset. But the circuit with partial scan differs from that with partial reset in that it is inclined to have higher testability as more flip-flops are scanned.

3. Experimental results

To verify the effectiveness of the proposed strategy for partial reset and partial scan, we generate selection orders of flip-flops and run test generation for benchmark circuits[31]. For partial reset, the circuits are redesigned in a simple way in order to include the reset part of circuit for test generation. One common input line is added for the reset signal and one inverter is added for the reversion of this signal. At the output of each flip-flop selected for reset 0(1), one two-input AND(OR) gate is added. For this gate, one of its inputs is the output line of the selected flip-flop. Another one comes from the reset input signal directly or by way of the added inverter, depending on that the gate is AND or OR type. The output line of this gate connects to those gates which were connected from the corresponding flip-flop. Figure 1 shows the connection. The reset input signal is 1 in normal mode, i.e., without applying reset signal, and 0 in reset mode, respectively. In Figure 1, flip-flop $FF1$ will be reset to 0 and flip-flop $FF2$ will be reset to 1 in reset mode. The faults on the added lines are also considered for test generation. Comparing to partial reset design, the circuits with partial scan are simply altered by considering the output lines of selected flip-flops as primary inputs and the input lines as primary outputs in test mode.

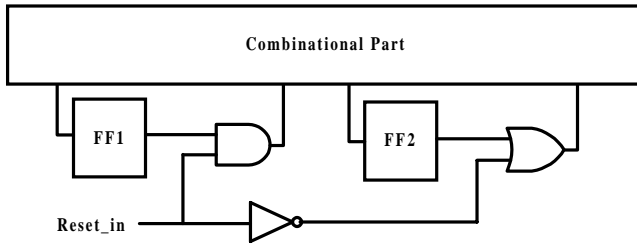


Figure 1: An example of partial reset connection for test generation

The numbers of selected flip-flops and test generation results are compared to those of [25], which implemented the circuits with partial reset the first, partial observation the second, and partial scan at the final stage. We omitted the partial observation step but still obtained better results as shown later. Since the detectable fault coverage could be a misleading indicator for sequential circuits[32], as mentioned in [25], they reported the results only by fault coverage to evaluate the testability of the circuit. To be comparable with their results which were run by GENTEST[33], we also implement a BACK[34]-like

algorithm for doing test generation. Table 1 and 2 give the results for comparison.

Table 1 lists the