Fast Test Application Technique Without Fast Scan Clocks

Seonki Kim and Bapiraju Vinnakota
ECE Department, University of Minnesota, Minneapolis, MN 55455
{skkim, bapi}@ece.umn.edu

Abstract

Built-in self-test (BIST) schemes need to set the state of the circuit under test (CUT) for each test vector applied. The two primary techniques by which the state is set are test-per-scan and test-per-clock. In a test-per-scan scheme, circuit states are set using one or more scan chains. Several scan cycles are required to apply a single test vector. In very large circuits, the time to apply each test vector may be quite high. The direct option of reducing test time with a fast scan clock is difficult to realize in practice. In a test-per-clock scheme, all circuit flip-flops are loaded in parallel. A new test vector can be applied in each cycle. The area overhead incurred in accessing each storage element directly is quite significant. We propose a new Broadcast BIST ($B^2$IST) scheme as a compromise between the two approaches. $B^2$IST uses time-division multiplexing (TDM) to load multiple storage elements in a broadcast group in a single clock cycle, but through only a single scan data input. Based on our $B^2$IST simulation, we compare the layout overhead and performance of $B^2$IST with that of traditional BIST schemes on ISCAS benchmark circuits. Thus, $B^2$IST can achieve the performance of a test-per-clock scheme, but only incur the overhead of a test-per-scan scheme.

1 Introduction

A built-in self-test (BIST) system consists of three primary functional parts: a test pattern generator (TPG), scan chains for test application and an output response analyzer (ORA). In general, the TPG is a linear feedback shift register (LFSR) that generates pseudo-random test patterns [1]. The scan chains enable the vectors from the TPG to be applied to internal circuit nodes. The area and performance impact of the scan chains should be minimized. The ORA is used to analyze the output response and flag defective circuits. Data compression techniques are generally used to reduce the storage space of the result of the ORA [2]. A multiple input signature register (MISR) is the most well known ORA.

Our focus in this paper is on the test application and response capture processes in a BIST scheme. Test application methods can be characterized into two broad groups: test-per-scan and test-per-clock BIST. In a test-per-scan scheme, flip-flop states are set using one or more scan chains. Before a new test vector can be applied, several cycles are required to set the state of the circuit. Using a very fast scan clock can reduce test application time. The generation of high-speed scan clocks is a challenge in the submicron chip design. Increasing the speed requirements on the scan clock requires a greater design effort and also consumes more circuit area. In a test-per-clock scheme, all the circuit flip-flops are accessed directly from the TPG and loaded in parallel in each cycle. To enable a parallel load, all the circuit memory elements must be directly accessible from the TPG. In very large circuits, both schemes have a significant impact either on test application time or in area overhead. Essentially, the two schemes offer a test time-area overhead tradeoff.

Contribution We present a new test application and response capture process, Broadcast BIST ($B^2$IST). In $B^2$IST, as with a test-per-clock scheme, a broadcast group of memory elements is loaded in one clock cycle. However, as with a test-per-scan scheme, this group is loaded with a single scan data input. Instead of being used as the first element in a scan chain, the scan input is broadcast to all the elements in the broadcast group over a single scan line. Each element has a non-overlapped narrow time window. The time-division multiplexing (TDM) process can be realized with the scan clock, that is phase-shifted using simple delay elements. Thus the actual rate of data loads is a multiple of the clock rate. $B^2$IST is applicable both to test application and to response capture processes. We present a detailed implementation for a $B^2$IST scheme. Our implementation discusses several details including the design of $B^2$IST scan elements, scan clock generation, TDM data generation. We simulated $B^2$IST with a 0.3µm technology. We compare $B^2$IST with the other two alternatives on ISCAS benchmark circuits. $B^2$IST is able to achieve test application times comparable to test-per-clock schemes with overhead comparable to test-per-scan schemes.

The rest of this paper is organized as follows. In Section 2, we review related previous work. In Section 3, we discuss the $B^2$IST architecture. Experimental results on ISCAS benchmark circuits are presented in Section 4. We conclude this paper in Section 5 with a summary of our work and the future work.

2 Previous work

Space limitations constrain us to a brief review of previous work. The purpose of test application is to control the logic values at specific nodes in the circuit. Here on, we will refer to the memory elements and the other nodes controlled by the test application process as control points. A full test-per-clock BIST and a test-per-scan STUMPS are shown in Figure 1(a) and (b) respectively.

In a test-per-clock scheme, new test patterns can be loaded every clock cycle. To control $n$ points, $n$ nets will have to be routed from the TPG to the appropriate nodes in the circuit. The out-
put response is available at the immediately next clock cycle. This scheme results in large area overheads on the data paths from the TPG to the control points and from the observation points to the ORA. BILBO BIST [4], circular BIST [5] are examples of test-per-clock schemes. The PSBIST was proposed in [8] as a combination of the test-per-clock and test-per-scan BIST. Test patterns were provided at every clock by the primary inputs and the scan chains. The responses were captured at the MISR at every clock cycle. The scan chain captured the response only when the new test pattern was completely scanned in. The PSBIST were modified into the BISTSCAN [9]. The BISTSCAN identified non-scan latches. They were excluded from the scan chain to reduce test application time and to break correlation in test patterns.

As mentioned earlier, the primary limitation of a test-per-scan scheme is on the test application time. The test time for \( n \) control points requires \((n+1)\) clock cycles. Accelerating the scan clock directly reduces test application time. Increasing the speed requirements on the scan clock requires a greater design effort, since one would have to route two high-speed clocks in the circuit under test (CUT). Faster clocks also consume more circuit area. Increasing the number of the scan chains, as in the STUMPS [3, 6] can also reduce test time. The longest scan path determines the test time of the STUMPS. Adding individual pattern paths and the level sensitive latches significantly reduced the long scan time in the STUMPS. The area overhead, however, was very significant, but expected to drop less than 5% in the future [7].

3 Broadcast BIST

Ideally, a test application and response capture process should have the test time of the test-per-clock approach and the area cost of the test-per-scan approach. A parallel load is not the only technique by which multiple memory elements are set in a single cycle. In practice, scan clocks are typically much slower than the speed at which the system operates.

Assume we treat the data input line in a scan scheme as a resource. A single resource can be shared among several consumers by using multiplexing. We propose the use of TDM, a common approach used extensively in signal transmission, to set the states of memory elements in a circuit. Our scheme, \( B^2\text{IST} \) works as follows. \( B^2\text{IST} \) uses the principle of data broadcasting and TDM. The memory elements are partitioned into groups called broadcast groups. The entire elements in a broadcast group can be set in one clock cycle. One clock cycle interval is divided into several non-overlapping windows. Each window is exclusively assigned to a single cell in the group. In each window, only one element in the broadcast group receives a latching clock edge. We refer to the memory element as being “triggered” in that window. When it is triggered, the memory element simply latches the value present at that time on the data line.

The data to be loaded into the memory elements in the broadcast group is input serially on a single data line. The values on this data line are simultaneously broadcast to all the elements in the broadcast group. A different value, corresponding to the value to be loaded into the memory element that is triggered, is input to the data line on each window in a clock cycle. Note that in all other windows, when it is not triggered, a memory element ignores the values on the data line. Thus, in each window though the data line is broadcast to all elements, one and only one memory element is loaded. However, in one clock period all the memory elements in the broadcast group are loaded.

3.1 \( B^2\text{IST} \) Scan Cell

A \( B^2\text{IST} \) scan cell is an improved test-per-scan BIST cell. Like other conventional test-per-scan BIST cells, it consists of two D-type flip-flops (DFF) triggered by two different clocks and a MUX. \( B^2\text{IST} \) scan cells use edge-triggered flip-flops. The difference, between the proposed scan cell and the conventional scan cell lies in the clocking strategy. The conventional scan cells require fast scan clocks to improve the test application time. Further, the BIST control circuitry needs to include the scan clock generation module and phase locked loops [10].

On the other hand, the proposed scan cell does not require a fast scan clock. The scan clocks are generated by phase-shifting of the system clock using delay elements. We denote the scan clocks as \( D_i\text{CLK} \), where \( D_i \) represents the degree of the skew delay.

Among the conventional test-per-clock BIST, a BILBO cell is implemented with two DFFs and a MUX. The BILBO cell works as a test-per-clock BIST when it uses the built-in LFSR. Basically it works as a shift register for transferring the test data between the memory elements and the TPG or the ORA. The BILBO gives no flexibility to designers on the TPG and the ORA, but the \( B^2\text{IST} \) does.

In the test-per-scan BIST, incoming inputs are buffered by the scan cells and propagated to the next scan cell by clocking. The test application time depends on the number of the scan cells. However, there is no buffering of the incoming inputs in the \( B^2\text{IST} \) scan cell. The data is broadcast to all the scan cells in the broadcast group. Only one scan cell is triggered to capture the incoming broadcast data with the same skew delay scan clock as used in the broadcast data generation.

Figure 2 shows an implementation of the \( B^2\text{IST} \). Each scan cell consists of two storage elements A and B. The skew delay clock triggers storage cell A in order to load the test pattern from the broadcast line. The system clock triggers storage cell B to apply the test pattern in storage cell A to the CUT. Hence, the \( B^2\text{IST} \) is also applicable at-speed tests. The control input, \text{TEST} determines the function of the \( B^2\text{IST} \) scan cell, as given in Table 1.
### Table 1: The modes of $B^2$IST

<table>
<thead>
<tr>
<th>Control Inputs</th>
<th>Functional Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>$TEST$</td>
<td>CLK</td>
</tr>
<tr>
<td>0</td>
<td>CLK</td>
</tr>
<tr>
<td>1</td>
<td>CLK</td>
</tr>
</tbody>
</table>

Storage cell $A$ captures the broadcast test data, regardless of $TEST$. Storage cell $B$ applies the test pattern at $TEST = 1$ and captures the response of the CUT at $TEST = 0$. The control strategy is shown in Figure 3.

![Figure 3: The control strategy](image)

Each $B^2$IST scan cell receives a pair of control clocks: the system clock (CLK) and a skew delay clock ($D_1$CLK). $D_1$CLK is used to generate the broadcast data and to restore the test data to the storage elements. Since the TDM windows and all the control clocks are periodical, each cell has access only to its exclusive TDM window.

So far, we studied on the fast scan-in process. Due to a space limitation, we briefly review the scan-out process. The broadcast data contains the test response in the scan-out process by the same scan BIST in order to achieve reduced test application time and the number of parallel scan paths. In the STUMPS, the number of the scan cells in the longest scan chain determines the test application time. Note that the STUMPS uses the first scan element in the chain to load the test pattern. The STUMPS has flexibility on the number of scan cells in a scan chain. In $B^2$IST, the physical factors such as the setup and hold time and the delay unit constrain the number of the parallel scan chains. However, $B^2$IST completes the test application in one clock cycle.

### 4 Results

$B^2$IST is characterized by two parameters: the number of the TDM windows and the number of the parallel scan chains. The overall framework of $B^2$IST is similar to that of the STUMPS. The difference lies in the test application time and the number of parallel scan paths. In the STUMPS, the number of the scan cells in the longest scan chain determines the test application time. Note that the STUMPS uses the first scan element in the chain to load the test pattern. The STUMPS has flexibility on the number of scan cells in a scan chain. In $B^2$IST, the physical factors such as the setup and hold time and the delay unit constrain the number of the parallel scan chains. However, $B^2$IST completes the test application in one clock cycle.

### 3.2 Broadcast Data Generation

Shrinking in the time axis is a good way to improving the speed performance. One clock period is divided into small TDM windows. Each TDM window must satisfy the setup and hold time constraints. The feasible unit delay time directly determines the number of the TDM windows. Having short delay units in the $B^2$IST is equivalent to generating fast scan clocks in the conventional test-per-scan BIST. For example, placing two TDM windows in one system clock cycle is equivalent to generating a scan clock that is twice faster than the system clock. A typical circuit of generating a broadcast data is shown in Figure 5. This circuit is applicable both to the scan-in and to the scan-out process in order to generate the broadcast data.

![Figure 5: A broadcast data generation](image)

$B^2$IST is characterized by two parameters: the number of the TDM windows and the number of the parallel scan chains. The overall framework of $B^2$IST is similar to that of the STUMPS. The difference lies in the test application time and the number of parallel scan paths. In the STUMPS, the number of the scan cells in the longest scan chain determines the test application time. Note that the STUMPS uses the first scan element in the chain to load the test pattern. The STUMPS has flexibility on the number of scan cells in a scan chain. In $B^2$IST, the physical factors such as the setup and hold time and the delay unit constrain the number of the parallel scan chains. However, $B^2$IST completes the test application in one clock cycle.

### 4.1 TDM Window Sizing

We performed the timing simulations to determine the width of the TDM window in the off-the-shelf technology. First we found the minimum width of the TDM window. The number of the TDM windows increased at the minimum width. The 0.4um technology was used on the Cadence timing simulation engine. Delay units were built with a group of inverters. With each window size of 0.3nsec, we would successfully place six TDM windows at the 500MHz clock rate. We could achieve the benefit that a 500MHz system clock in $B^2$IST is equivalent to a 3.0GHz scan clock in the test-per-scan BIST.

In practice, the period of the system clock is much longer than the unit delay time. We can estimate the number of the TDM windows by the two factors: the setup and hold time and the feasible unit delay element. We denote the setup and hold time requirement by $T_s$, the unit delay by $T_d$ and the clock period by $T_c$. The number of the TDM windows is determined by the lower bound of $\{\frac{T_s}{T_d}, \frac{T_s}{T_c}\}$. 

![Figure 4: The $B^2$IST block diagram](image)
4.2 Benchmark Comparison

The area overhead for the three reference BIST models is given in Table 2. The logic level of each circuit was computed from the SIS. We computed the number of the 0.3pm TDM windows in one clock cycle. We numerically estimated the benefit of the $B^2$IST on area and performance overhead on some ISCAS benchmark circuits. Columns 2, 5 and 8 show the number of scan data lines to be routed between the scan cells and the TPG. Column 3, 6 and 9 indicate the overhead to the critical path. We considered the impact of the propagation delay of a 2-input MUX. Column 4, 7 and 10 give the test application time. $B^2$IST achieves the test-per-clock speed performance with the incurred cost of the test-per-scan STUMPS. Columns 2 to 4 characterize the test-per-clock BIST. The area overhead of the test-per-clock BIST increases rapidly as the number of the scan data paths does. Column 5 through 7 characterize the test-per-scan BIST. It incurs the least area overhead, but through the significant test application time. Columns 8 through 10 characterize the $B^2$IST. It has a significant area reduction as compared with that of the test-per-clock BIST. It also significantly reduces the test application time as compared with that of the test-per-scan STUMPS. The $B^2$IST is a compromise between the two BIST alternatives to achieve the better test application time with the less area overhead. Table 2 visualizes that the $B^2$IST can play an important role in deep submicron technology in the future.

Table 2: The impact of the $B^2$IST at the 0.4µm technology

<table>
<thead>
<tr>
<th>Benchmarks</th>
<th>Test-per-clock BIST</th>
<th>Test-per-scan BIST</th>
<th>$B^2$IST</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># of nets</td>
<td>crit. path (%)</td>
<td>R/W time (cycle)</td>
</tr>
<tr>
<td>s113</td>
<td>19</td>
<td>11.0</td>
<td>1</td>
</tr>
<tr>
<td>s832</td>
<td>5</td>
<td>20.0</td>
<td>1</td>
</tr>
<tr>
<td>s1488</td>
<td>6</td>
<td>18.1</td>
<td>1</td>
</tr>
<tr>
<td>s3375</td>
<td>179</td>
<td>15.1</td>
<td>1</td>
</tr>
<tr>
<td>s13207</td>
<td>699</td>
<td>4.4</td>
<td>1</td>
</tr>
<tr>
<td>s38584</td>
<td>591</td>
<td>5.2</td>
<td>1</td>
</tr>
</tbody>
</table>

5 Conclusion

Even when tests are generated on-chip, a BIST scheme needs access to internal circuit nodes for test application. Fast scan clocks are needed to accelerate a test-per-scan scheme. A test-per-clock scheme incurs significant costs because of the number of internal nodes that need to be directly controlled from the inputs. We developed a new test application scheme, Broadcast BIST. The $B^2$IST transfers test patterns to multiple memory elements in one cycle through a single broadcast line. We presented a design for the $B^2$IST scan cell and for the broadcast data generation. We showed that $B^2$IST could be realized with a system clock that was skewed using simple delay elements. The operation of the circuits was verified at the circuit level on a 0.4µm technology. To enable an effective comparison, we verified the estimated overhead of one benchmark circuits, with $B^2$IST and two alternatives. Our experimental result shows that the real test application time in $B^2$IST is almost as low as that of a test-per-clock scheme. However, the area and performance overhead is comparable to that of a test-per-scan STUMPS scheme. The $B^2$IST is a cost-effective alternative to these two orthogonal options. The $B^2$IST scheme presented in this paper can be improved through the development of better delay elements and the deep submicron technology.

REFERENCES