Abstract-Built-In Self-Test (BIST) techniques constitute an attractive and practical solution to the problem of testing VLSI circuits and systems. Input vector monitoring concurrent BIST schemes perform testing concurrently with the operation of the circuit. In this paper a novel input vector monitoring concurrent BIST scheme is presented that compares favorably to previously proposed schemes with respect to the required hardware overhead.

1. Introduction
Built-In Self Test (BIST) schemes utilize a Test Pattern Generator to generate the test patterns that are applied to the inputs of the Circuit Under Test (CUT) [1]. In off-line BIST, the normal operation of the CUT is stalled in order to perform the test. Thus, if the CUT is critical for the function of the circuit, the performance is degraded. Input vector monitoring concurrent BIST schemes [2] - [9] exploit vectors appearing to the inputs of the CUT during normal operation to perform on line testing. The measures utilized to evaluate this class of schemes are (a) the Concurrent Test Latency (CTL), i.e. the number of cycles that the CUT must operate in normal mode in order to expect that the concurrent test is complete and (b) the hardware overhead.

Sharma and Saluja [7] proposed the Built-In Concurrent Self Test (BICST) scheme illustrated in Figure 1. BICST is based on a pre-computed test set and monitors the vectors V arriving at the combinational CUT inputs and the respective outputs B. The Concurrent BIST Unit (CBU) compares the incoming vector against the pre-computed test set and, if the input vector belongs to the test set, compares the response of the CUT with the known good response; if the two vectors differ, an error is assumed to have occurred in the CUT and the error signal is immediately activated.

In [9] a scheme was proposed that, based on the idea of selecting a subset of the input signals and utilizing them as inputs to a decoder-based structure, achieves a reduction in the hardware overhead comparatively to the scheme proposed in [7], while at the same time provides for off-line test capabilities. Almukhaizim et al investigated the problem of concurrent testing of sequential machines in [16]-[20]. Other concurrent online BIST schemes [11] - [14] utilize test patterns containing don’t care values. In [21] an Input Vector Monitoring On line Concurrent BIST based on multilevel decoding logic was presented.

The paper is organized as follows. In Section 2 the scheme proposed by Kunzmann is presented. In Section 3 we present the proposed scheme. In Section 4 we present implementation and comparison data; finally, in Section 5 we conclude the paper.

2. Generation of test patterns based on minimal test sets
Kunzmann [22] proposed a procedure based on the observation that, very often, pairs of patterns can be identified that differ in one bit position. This observation led to the approach to derive from each deterministic test pattern distinct test patterns that differ in only one bit from the original i.e. any pattern with Hamming distance exactly 1. For example, for the combinational benchmark c6288 from the ISCAS85 suite, 35 patterns are enough to detect all faults. Alternatively, using 7 basic test patterns and the 7*32 patterns that differ from these 7 patterns, i.e. a total of 224 patterns, all faults are detected. Kunzmann then went on to discover that, in fact, not all bit positions should be inverted for a complete coverage. In fact, he gave us the following table that presents his findings with respect to the number of patterns in the basic test set and the number of bits to be inverted [22].
<table>
<thead>
<tr>
<th>Benchmark</th>
<th>n</th>
<th>T</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>c432</td>
<td>36</td>
<td>16</td>
<td>27</td>
</tr>
<tr>
<td>c499</td>
<td>41</td>
<td>10</td>
<td>28</td>
</tr>
<tr>
<td>c880</td>
<td>60</td>
<td>14</td>
<td>34</td>
</tr>
<tr>
<td>c1355</td>
<td>41</td>
<td>16</td>
<td>36</td>
</tr>
<tr>
<td>c1908</td>
<td>33</td>
<td>20</td>
<td>33</td>
</tr>
<tr>
<td>c2670</td>
<td>233</td>
<td>20</td>
<td>96</td>
</tr>
<tr>
<td>c3540</td>
<td>50</td>
<td>35</td>
<td>38</td>
</tr>
<tr>
<td>c5315</td>
<td>178</td>
<td>25</td>
<td>96</td>
</tr>
<tr>
<td>c6288</td>
<td>32</td>
<td>7</td>
<td>22</td>
</tr>
<tr>
<td>c7552</td>
<td>207</td>
<td>38</td>
<td>111</td>
</tr>
</tbody>
</table>

**Table 1**: Results from [22] presenting the findings for the ISCAS 85 benchmarks

In Table 1, in the first column we present the benchmark name; in the following columns the number of inputs of the benchmark (n), the number of patterns in the basic test set (T) and the number of inputs to be inverted are presented. The hardware proposed by [22] in order to apply the test patterns comprises a RAM to store the test patterns, a counter and a pair of shift registers to enable / disable the toggling of the bits. In the next Section we will present the proposed idea for concurrently testing a module, based on this concept.

At this point, a note should be done with respect to the \( t \geq \log_2 T \) bits that are utilized to address the ROM and RAM inputs. Following [9] these bits must be row-distinct, in the sense that no two \( t \)-tuples should have the same values. In [9] it was shown that, for the test sets extracted from the Compactest tool [26] \( t \) is extremely close to \( \log_2 T \). In fact, \( t \) is closer to \( \log_2 T \) for smaller values of \( T \). For the proposed scheme, since the basic test set is very small, \( t \) will be close to \( \log_2 T \).

In order to further clarify the proposed scheme let us consider the example of a 7-input CUT and the basic test set comprising the following patterns:

\[
\begin{align*}
0010110 \\
1001001 \\
0111011
\end{align*}
\]

Let us further assume that \( c=3 \), i.e. three bits must be inverted in order to completely test the CUT. Finally, without loss of generality, let us assume that the bits to be inverted are the rightmost three bits. Therefore, the following 12 patterns will be generated to completely test the CUT.

\[
\begin{align*}
0010110 & \quad 0010010 & \quad 0010100 & \quad 0010111 \\
0111011 & \quad 0111111 & \quad 0111001 & \quad 0111010 \\
1001001 & \quad 1001101 & \quad 1001011 & \quad 1001000
\end{align*}
\]

The application of the proposed scheme for the testing of the specific module is presented in Figure 2.

---

In Figure 1, the Circuit Under Test (CUT) has \( n \) inputs and \( m \) outputs. The Concurrent BIST Unit (CBU) comprises a \( T \times c \) word ROM, a \( T \times \log_2 c \) RAM, an \( \log_2 c \) incremeremt, an \( \log_2 c \) to-\( c \) decoder, \( c \) two-input XOR gates and a \( n \)-stage comparator. The inputs arriving to the inputs of the CUT are driven to the inputs of the CBU as follows: \( t \) bits are driven to drive the inputs of the ROM and the RAM. The \( \log_2 c \) bit output of the RAM is driven to the decoder and to the inputs of the incremeremt; the \( c \) outputs of the decoder are driven to a series of \( c \) XOR gates. The outputs of the \( c \) XOR gates, along with the \( t \) inputs of the CUT are driven to the inputs of the \( n \)-stage comparator. If the inputs of the comparator match, then the RV is enabled to capture the outputs of the CUT and the increased value of the RAM is written to the inputs.

3. Proposed scheme

The basic idea of the proposed scheme is presented in Figure 1.
Initially, all the words of the RAM are set to zero. When one of the words that belong to the basic test set reaches the CUT inputs, e.g. 0010110, then the two left most bits drive the RAM and ROM inputs; the pattern 10110 appears at the ROM outputs; the pattern 000 appears at the RAM outputs. The topmost output of the decoder (the one left unconnected) is enabled; hence the input of the CUT is compared to the pattern 0010110, and since the output of the high order bit of the RAM is 0, hit is enabled and the RV is enabled to capture the response of the CUT, the increased value 001 is written to the first word of the RAM and the concurrent test goes on.

When the value 0010111 appears at the CUT inputs (and the contents of the first RAM word are 11) the value written at the RAM word is 100. From this time, all the patterns that differ in one bit from the first patterns of the basic set have appeared at the CUT inputs, hence no hit will be performed again for this group of patterns (i.e. \{0010110, 0010011\}).

It should be noted that, when the number of patterns in a group is not a power of two (as is the case in the above example) then the number of bits on the RAM is not \(\lceil \log_2 c \rceil + 1\), but exactly \(\lceil \log_2 c \rceil\). In this case, the AND gate that drives the enable input of the RV and the write enable signal of the RAM is not driven by the high-order bit of the output of the RAM, but from the output of a module that is driven by the outputs of the RAM. This output is enabled when the output of the RAM is greater than the number of the patterns in the group which, in turn, means that all the bits required to toggle for this word have been toggled. For example, for the case that \(c=2\) (i.e. two bits must be toggled) the module is a 2-input gate that detects the value 11 at the outputs of the RAM.

4. Evaluation and Comparisons

In order to evaluate the proposed scheme, in the sequel we present the calculation of the concurrent test latency and the hardware overhead. In order to implement the proposed scheme, we use Table 2.

<table>
<thead>
<tr>
<th>Module</th>
<th>Size</th>
<th>Hardware (gates)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM</td>
<td>(T) words (\times \log_2 c) bits each</td>
<td>(1.5 \times T \times \log_2 c)</td>
</tr>
<tr>
<td>ROM</td>
<td>(T) words (\times (n-t)) bits each</td>
<td>(0.25 \times T \times (n-t))</td>
</tr>
<tr>
<td>Decoder</td>
<td>(\log_2 c) to (c)</td>
<td>(&lt;3 \times c)</td>
</tr>
<tr>
<td>Inc</td>
<td>(\log_2 c) stages</td>
<td>(2 \times \log_2 c)</td>
</tr>
<tr>
<td>Comparator</td>
<td>(n) stages</td>
<td>(2 \times n)</td>
</tr>
<tr>
<td>XOR gates</td>
<td>(c) XOR gates</td>
<td>(c)</td>
</tr>
</tbody>
</table>

Table 2: Calculation of the hardware overhead of the proposed scheme

For the calculations we have considered a 6-transistor SRAM cell, the fact that a ROM cell accounts for 1/4 gates, that a XOR gate can be implemented with 4 transistors [23], and a decoder can be implemented (see e.g. [24]) using a two-level implementation and is at most 3 gates times the number of outputs of the decoder.

In order to evaluate the proposed scheme we compare it to BICST [7] and the MICSET scheme proposed in [9] with respect to the application to some of the ISCAS 85 benchmarks [27]. The hardware comparisons are presented in Table 3 in terms of gate equivalents. In the comparisons we have not taken into account the hardware required for the offline test generation, i.e. the test pattern generation, response verifier and the multiplexers required to the inputs of the CUT. In Table 3, in the first column we present the benchmark name. In the second, third and fourth column we present the hardware overhead of the schemes proposed in [7], [9] and the newly proposed scheme. In the last column of the Table, we present the Decrease in hardware compared to the MICSET scheme.

From Table 3, the proposed scheme presents lower hardware overhead than the previously proposed schemes (88% on average).
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>c432</td>
<td>1507</td>
<td>2231</td>
<td>374</td>
<td>83%</td>
</tr>
<tr>
<td>c499</td>
<td>4112</td>
<td>2972</td>
<td>287</td>
<td>90%</td>
</tr>
<tr>
<td>c880</td>
<td>4847</td>
<td>4181</td>
<td>493</td>
<td>88%</td>
</tr>
<tr>
<td>c1355</td>
<td>6544</td>
<td>4588</td>
<td>430</td>
<td>91%</td>
</tr>
<tr>
<td>c1908</td>
<td>7201</td>
<td>4926</td>
<td>453</td>
<td>91%</td>
</tr>
<tr>
<td>c6288</td>
<td>964</td>
<td>1584</td>
<td>209</td>
<td>87%</td>
</tr>
</tbody>
</table>

Table 3: Comparison with previously proposed schemes w.r.t. hardware overhead (gate equivalents)

6. Conclusions
Concurrent on-line Input vector monitoring concurrent BIST schemes exploit vectors appearing to the inputs of the CUT during normal operation to perform concurrent on-line testing. In this work we have presented an input vector monitoring concurrent BIST scheme that, utilizing the concept of basic test sets, presents lower hardware overhead than previously proposed schemes.

Acknowledgments
This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Archimedes III. Investing in knowledge society through the European Social Fund.

References


