An Embedded Autonomous Scan-Based Results Analyzer (EARA) for SoC Cores

Mohsen Nahvi and André Ivanov

Department of Electrical and Computer Engineering
University of British Columbia
2356 Main Mall, Vancouver, B.C. V6T 1Z4, Canada
Telephone: \{mohsenn, ivanov\}@ece.ubc.ca

Abstract
Relying solely upon external ATE resources for scan test in complex SoC designs is increasingly difficult. In this work, we develop the concept and implementation of an Embedded Autonomous Results Analyzer (EARA) to be used in our modified Dedicated Autonomous Scan-based Testing (DAST) methodology. DAST introduces hierarchy and separates the functionality of ATE resources into two distinctive classes: a) test data communication, and b) test data control and observation. Consequently, test data control/observation functions are transferred to embedded blocks. In this work, we extend DAST to include the sending of expected test results along with the test stimulus to enable on-chip comparison. We present implementation results of EARA when applied to a number of SoC benchmarks.

1. Introduction

With every scaling node in the semiconductor industry, more transistors can be integrated on a chip thereby increasing the productivity gap between manufacturing capability and designers’ ability to map high-level functions onto the large number of transistors. To overcome this gap, previously designed/verified functional blocks, or Intellectual Properties (IPs), are used as embedded cores in the integrated solutions of System-on-a-Chip (SoC).

An SoC design integrates pre-designed cores on a single piece of silicon. Testing core-based SoCs is a major challenge in terms of ultra deep sub-micron (UDSM) issues, limited accessibility of the embedded cores, the integration/scheduling of different core test programs, and more [1].

In addition, testing core-based SoCs has reached a level of complexity that relying on external Automatic Test Equipment (ATE) is insufficient [2]. The cost of ATE rises as the complexity of chips increase. Moreover, ATEs can quickly become outdated compared to the chips that they test, and for example often lack the necessary resolution and accuracy to test new devices effectively. A classical approach to the test problem has been to make testers bigger, faster, more flexible, and equipped with more memory such that they be able to handle any chip. In effect, today’s testers are very specialized, complex, and expensive “super-computers”.

The increasing complexity, prohibitive cost, and typically lagging technology of external testers call for specialized embedded support infrastructure blocks suited for embedded testing. Embedded testing takes over some functionality of external ATE resources [3][4]. It is imperative that such embedded solutions work in harmony with their specialized off-chip counterparts.

Built-In-Test (BIST) techniques have been used previously to mitigate the rising cost of testing [5]. In BIST techniques, on-chip hardware, such as Linear Feedback Shift Registers (LFSR), typically generate pseudo-random test vectors. These test vectors are then applied to the circuit-under-test (CUT) and the test results are compared to the expected results on-chip. A go/no-go signal can then be generated upon completion of the test. A potential problem with such pseudo-random pattern BIST is the potentially lower fault coverage compared to that resulting from deterministic patterns obtained via Automatic Test Pattern Generation (ATPG). High fault coverage well over 97% is now the norm in industry. The relatively lower fault coverage potentially associated with a BIST methodology can be addressed in different ways. One method consists of inserting additional test points in the design [5]. A judged better strategy is known as mixed-mode BIST, which uses embedded deterministic test vectors as well as pseudo-random vectors. A number of techniques can be used to realize mixed-mode BIST [6]: e.g., weighted random pattern generation; store-and-generate; bit-fixing or bit-flipping.

To reduce the memory requirements of external ATEs and, hence, reduce test cost, test data compression/compaction techniques have been suggested [7][8]. In addition, since most SoC designs have a microprocessor block, the on-chip microprocessor can be used as an embedded tester to test other cores [9]. Potential shortcomings of these techniques lie in complicated scheduling, on-chip routing, long test time, and the difficulty in testing cores in parallel.

The concept and the first implementation of a Dedicated Autonomous Scan-based Testing (DAST) methodology for embedded digital cores were presented in [10]. DAST amounts to a specialized embedded support infrastructure. In DAST, core test vectors, generated by ATPG, are pre-processed into a new test-data protocol such that all the control and observation sequence of testing is transferred from the external ATE to a dedicated block, i.e., the EAS, associated with single or multiple cores on an SoC.

In this work, we develop the concept and implementation of an Embedded Autonomous Results Analyzer (EARA) to be used with DAST. One shortcoming with DAST as presented in [10]
springs from the fact that the test results are sent to a signature analyzer for comparison and generation of a go/no-go signal. Using a signature analyzer at the output of a core for test results analysis can mask some faults due to inherent aliasing in signature analyzer compaction schemes. Another problem with signature analyzers is the zero-ing out, where the history of the circuit response is lost [5]. In the present work, we replace the signature analyzer with EARA that deterministically analyzes the test results.

The introduction of an EARA in DAST has minimal overhead in terms of test time and area in our new scheme. In addition, the generation and application of test vectors as well as test results comparison does not deviate from the conventional scan/ATPG approach. Hence, our scheme does not imply any compromise in terms of test-quality (coverage and fault models), ease of use, and broad applicability. Moreover, as the EARA block is external to the core, the design impact is also minimal.

2. Dedicated Autonomous Scan-based Testing (DAST)

By nature, scan-based testing is a repetitive procedure. The steps in scan testing a core can be summarised by the generic waveforms in Fig. 1. In Fig. 1, the signal labels PI, SI, CLK, ST_PO, and ST_SO denote primary inputs values, scan inputs values, test clock, strobe primary outputs, and strobe scan outputs, respectively. Also, TS represents the value of the Test Select signal.

In the traditional methodology, the ATE is responsible for transforming raw test vector data into relatively complex scan sequence waveforms of Fig. 1. In DAST, however, simple hierarchy is introduced whereby the complex ATE scan functionality is replaced by a simple off-chip test-data compiler and the waveforms of Fig. 1 are faithfully generated on-chip.

3. Embedded Autonomous Results Analyzer (EARA)

3.1. Concept

In our proposed modification to DAST, we replace the signature analyzer at the output of a core with an EARA block. To this end, both test stimulus and expected test results are sent to a core via a Test Access Mechanism (TAM). Upon receiving test data, an Embedded Autonomous Sequencer (EAS) applies the test data to a core and generates the test clock. Simultaneously, and in parallel to the EAS operation, the EARA collects the test results from the PO and SO pins of the core and compares them with the incoming expected test results. On the first occurrence of an inconsistency between the expected and actual results, a “sticky” no-go signal is generated by the EARA block marking the detection of a fault. This idea is illustrated in Fig. 2.

The design flow incorporating EARA in DAST is shown in Fig. 3. This flow is only partially different from a flow using conventional ATPG/Scan. Similar to the case of a conventional ATPG/Scan test flow, scan chains are assumed to be inserted automatically and/or manually before the test vector and expected test result files are generated by the ATPG. The function of EARA must be fully deterministic. Hence no unknown values, i.e., “X” values, can be allowed in the test stimulus and expected test results, as otherwise two lines are needed to encode the 3-valued data bits. In our specific case, without loss of generality, all “X” values in the test vector file are replaced with the zero logic value. This modified test vector file is used to apply the test vectors to a gate-level model of the circuit and real expected results are collected and replaced with “X” values in the expected test results file. Using our EAS- and EARA- compilers, the test vector and expected results files are subsequently compiled into the test data and the expected test result protocols of EAS and EARA. (Note that the EAS design is modified from that in [10] to allow its working in synchrony with EARA).

Both the EAS and EARA logic is generated in RTL VHDL or Verilog and placed at the periphery of a core under test, and are thereby generic and non-intrusive to a core under test. The EAS and EARA logic only depends on a core’s primary...
inputs/outputs numbers, scan chains number/depth, and specific type of scan cells used to implement the chains. They do not depend on a core’s functionality. Both the EAS and EARA blocks also include bypass circuitry to allow the SoC’s normal functional mode.

For the test flow, we only require a simple data transmitter to send the test and the expected result data onto the chip as shown in Fig. 2. The test control, waveform generation, and comparison to the real results are all done automatically by the EAS and EARA blocks.

3.2. Implementation

In this work, we implemented EARA and the modified EAS assuming that each is connected to the external test resource block via a serial connection. We did not assume any specific TAM architecture. However, in our experiments, we used a direct connection as well as the indirect TAM (NIMA) introduced in [11] as example TAMs. Assuming NIMA as a TAM, only three test pins are required in DAST: one for each of test- and expected results-data delivery, and one for the Go/No-Go signal. This is because, in NIMA, control signals of the TAM are embedded in the packets.

The modification of the ATPG-generated file, as described in Section 3.1, is core dependant and can take any length of time ranging from a few minutes for simple cores to several hours for more complex ones. We developed two C programs that take as input the modified ATPG test vectors and expected test results and insert three-bit op-codes to the beginning of each section of the test program to convert the test vectors and expected results into EAS and EARA protocols, respectively. In general, the time taken for this protocol conversion is less than one second on Sun Blade100 machines. The following op-codes are used as simple instructions in the EAS and EARA:

1. Shift-PI-BSR: shift into primary input boundary scan register;
2. Shift-SC: shift into scan chains;
3. Shift-SC-BSR: shift into scan inputs boundary scan register;
4. Assert-CIk: assert the test clock;
5. Shift-PO-BSR: shift into primary output boundary scan register;

Fig. 4 gives the modified Algorithmic State Machine (ASM) of EAS as a pseudo-code where the modification to the design of EAS in [10] is highlighted.

Fig. 5 illustrates the Algorithmic State Machine (ASM) of EARA as a pseudo-code while Fig. 6 illustrates its hardware block diagram. Clearly, the hardware implementation of EARA is extremely simple and compact, and requires minimal design effort for any given core. Moreover, the modularity of the EARA design allows for its automation with minimal effort.

As shown in Fig. 6, in the EARA block, the incoming data is captured in a shift register until its FSM block can decode the op-codes. Expected test results for the primary outputs are then sent to the PO-BSR and appropriate shift/capture signals are asserted by the FSM. The expected test results of the scan outputs, however, are always captured in the SO-BSR. Using XOR gates, the outputs of both the PO- and SO-BSR are always compared to a core’s PO and SO values, respectively. However, using AND gates, the results of the comparison is gated by the ST_P0 and ST_SO signals, respectively. The output of the AND gates then act as chip enable signals to two flip fops such that if a mismatch is detected the Go/No-Go signal is asserted high.

IEEE P1500 [12] requires boundary scan registers at both primary inputs and outputs of an SoC core. For this reason, the effective area of the EARA block is taken to be the amalgam of the FSM, the shift registers, and the user logic, and is denoted by Adj_EARA in this paper. Similarly, Adj_EAS denotes the effective area for EAS blocks.
both the cores and their EAS and EARA blocks. Eqns. 1, 2, and 3 we developed DAST test time models as given by the following several states as illustrated in Fig. 1 and described in Section 2, each test pattern, both EAS and EARA sequence through. Given that op-codes consist of three bits and that, typically, for these test-benches allowed us to simulate the application of test SoC3 benchmark. We assumed the same clock frequency for EAR codes were then combined into a top VHDL gate-level representation of each core and its dedicated EAS and constituent cores, we developed a corresponding EAS and its description of SoC3 as presented in [10]. For each of its 4. Experimental Procedure For the first phase of our experiment, we used a gate-level description of SoC3 as presented in [10]. For each of its constituent cores, we developed a corresponding EARA and its modified EAS blocks in VHDL, as discussed in Section 3. The gate-level representation of each core and its dedicated EAS and EARA RTL codes were then combined into a top VHDL module and these modules were used as core instances in the SoC3 benchmark. We assumed the same clock frequency for both the cores and their EAS and EARA blocks. Separate test-benches were also developed in VHDL. Using these test-benches allowed us to simulate the application of test vectors to each core in the SoC and predict expected test time. Given that op-codes consist of three bits and that, typically, for each test pattern, both EAS and EARA sequence through several states as illustrated in Fig. 1 and described in Section 2, we developed DAST test time models as given by the following Eqs. 1, 2, and 3 developed DAST test time models as given by the following Eqs. 1, 2, and 3.

\[ T_{TS1_{-}DAST} = (9 + 2PI + SI + SI + SE) + TP(4 + 2PO + SI + SE) \]  
iff \( PO \geq PI + 6 \)  

where \( PI, SI, \) and \( PO \) are as defined in Section 2, \( TP \) is the total number of test patterns, \( SE \) is the maximum number of scan cells in the scan chain(s), and \( TM1_{-}DAST, TM2_{-}DAST, \) and \( TM3_{-}DAST \) are the test time models for DAST in terms of clock cycles. In the second phase of our experiment, we used these former models to predict the test time performance of DAST on the ITC’02 SoC Benchmarks [13], for which the test vector files are unavailable. In addition, to compare DAST test time to that of a serial connection in a conventional external ATE-based methodology, we developed theoretical lower bound test-time models for the latter. Underlying assumptions of the latter models are that test data is applied to the core serially and that the results are also observed serially. These models are given in Eqs. 4, 5, and 6 where \( TS1, TS2, TS3 \) are in clock cycles.

\[ T_{TS3} = (2PI + SI + SI + SE) + TP(2PI + SI + SI + SE) \]  
iff \( PO < PI \)  

\[ T_{TS2} = (2PI + SI + SI + SE) + TP(PI + PO + SI + SI + SE) \]  
iff \( PI + SI > PO \geq PI \)  

\[ T_{TS1} = (2PI + SI + SI + SE) + TP(2PI + PO + SI + SI + SE) \]  
iff \( PO \geq PI + SI \)  

For each constituent core of the SoC3 benchmark, we synthesized a corresponding EAS and EARA block in TSMC’s 0.18 µm technology and collected area and power data. These data were collected using the Area and Power Report tools of Synopsys’ Design Compiler. Moreover, using the parameters given for the cores (modules) in the ITC’02 SoC Benchmarks, we designed corresponding EAS and EARA blocks. These EAS and EARA blocks were synthesized as described earlier and their area requirements and power consumption collected with Design Compiler. Finally, using our DAST test time models, we compiled DAST test times for ITC’02 Benchmark modules.

<p>| Table 1: EAS Area and Power for ITC’02 SoC Benchmark Modules |
|---------------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|</p>
<table>
<thead>
<tr>
<th>ITC’02 cores</th>
<th>PI BSR</th>
<th>SE BSR</th>
<th>Buffer in</th>
<th>FSM</th>
<th>Ads EAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SoC3 m6</td>
<td>40335</td>
<td>925</td>
<td>1125</td>
<td>26</td>
<td>229</td>
</tr>
<tr>
<td>SoC3 m7</td>
<td>7155</td>
<td>164</td>
<td>6033</td>
<td>138</td>
<td>229</td>
</tr>
<tr>
<td>SoC3 m8</td>
<td>11678</td>
<td>260</td>
<td>3221</td>
<td>89</td>
<td>229</td>
</tr>
<tr>
<td>SoC3 m9</td>
<td>6684</td>
<td>195</td>
<td>6033</td>
<td>138</td>
<td>229</td>
</tr>
<tr>
<td>SoC3 m10</td>
<td>6684</td>
<td>195</td>
<td>3221</td>
<td>89</td>
<td>229</td>
</tr>
<tr>
<td>SoC3 m11</td>
<td>6684</td>
<td>195</td>
<td>3221</td>
<td>89</td>
<td>229</td>
</tr>
<tr>
<td>SoC3 m12</td>
<td>6684</td>
<td>195</td>
<td>3221</td>
<td>89</td>
<td>229</td>
</tr>
<tr>
<td>SoC3 m13</td>
<td>6684</td>
<td>195</td>
<td>3221</td>
<td>89</td>
<td>229</td>
</tr>
<tr>
<td>SoC3 m14</td>
<td>6684</td>
<td>195</td>
<td>3221</td>
<td>89</td>
<td>229</td>
</tr>
<tr>
<td>SoC3 m15</td>
<td>6684</td>
<td>195</td>
<td>3221</td>
<td>89</td>
<td>229</td>
</tr>
<tr>
<td>SoC3 m16</td>
<td>6684</td>
<td>195</td>
<td>3221</td>
<td>89</td>
<td>229</td>
</tr>
<tr>
<td>SoC3 m17</td>
<td>6684</td>
<td>195</td>
<td>3221</td>
<td>89</td>
<td>229</td>
</tr>
<tr>
<td>SoC3 m18</td>
<td>6684</td>
<td>195</td>
<td>3221</td>
<td>89</td>
<td>229</td>
</tr>
<tr>
<td>SoC3 m19</td>
<td>6684</td>
<td>195</td>
<td>3221</td>
<td>89</td>
<td>229</td>
</tr>
<tr>
<td>SoC3 m20</td>
<td>6684</td>
<td>195</td>
<td>3221</td>
<td>89</td>
<td>229</td>
</tr>
<tr>
<td>SoC3 m21</td>
<td>6684</td>
<td>195</td>
<td>3221</td>
<td>89</td>
<td>229</td>
</tr>
<tr>
<td>SoC3 m22</td>
<td>6684</td>
<td>195</td>
<td>3221</td>
<td>89</td>
<td>229</td>
</tr>
<tr>
<td>SoC3 m23</td>
<td>6684</td>
<td>195</td>
<td>3221</td>
<td>89</td>
<td>229</td>
</tr>
<tr>
<td>SoC3 m24</td>
<td>6684</td>
<td>195</td>
<td>3221</td>
<td>89</td>
<td>229</td>
</tr>
<tr>
<td>SoC3 m25</td>
<td>6684</td>
<td>195</td>
<td>3221</td>
<td>89</td>
<td>229</td>
</tr>
<tr>
<td>SoC3 m26</td>
<td>6684</td>
<td>195</td>
<td>3221</td>
<td>89</td>
<td>229</td>
</tr>
</tbody>
</table>

\[ T_{TS1_{-}DAST} = (9 + 2PI + SI + SI + SE) + TP(4 + 2PO + SI + SE) \]  
iff \( PO \geq PI + 6 \)  

\[ T_{TS2_{-}DAST} = (9 + 2PI + SI + SI + SE) + TP(2PI + PO + SI + SI + SE) \]  
iff \( PI + SI + 6 \geq PO \geq PI + 3 \)  

Fig. 6: EARA Hardware Block Diagram.
Table 2: EARA Area and Power for ITC’02 SoC Benchmark Modules

<table>
<thead>
<tr>
<th>Core Type</th>
<th>Adjusted EARA Area (u2)</th>
<th>Adj EARA Power (uW)</th>
<th>P1500 BSR Area (u2)</th>
<th>P1500 BSR Power (uW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>d281_m5</td>
<td>30016</td>
<td>654</td>
<td>785</td>
<td>20</td>
</tr>
<tr>
<td>d695_m5</td>
<td>40335</td>
<td>883</td>
<td>3744</td>
<td>95</td>
</tr>
<tr>
<td>d695_m6</td>
<td>20218</td>
<td>443</td>
<td>1915</td>
<td>49</td>
</tr>
<tr>
<td>d695_m9</td>
<td>42559</td>
<td>933</td>
<td>3744</td>
<td>96</td>
</tr>
<tr>
<td>f2126_m1</td>
<td>69550</td>
<td>1529</td>
<td>1012</td>
<td>26</td>
</tr>
<tr>
<td>f2126_m2</td>
<td>18409</td>
<td>404</td>
<td>1915</td>
<td>49</td>
</tr>
<tr>
<td>f2126_m3</td>
<td>2769</td>
<td>61</td>
<td>207</td>
<td>5</td>
</tr>
<tr>
<td>g1023_m1</td>
<td>36062</td>
<td>792</td>
<td>1695</td>
<td>43</td>
</tr>
<tr>
<td>g1023_m2</td>
<td>28358</td>
<td>616</td>
<td>329</td>
<td>9</td>
</tr>
<tr>
<td>g1023_m4</td>
<td>20633</td>
<td>452</td>
<td>549</td>
<td>14</td>
</tr>
<tr>
<td>g1023_m10</td>
<td>50023</td>
<td>1088</td>
<td>207</td>
<td>5</td>
</tr>
<tr>
<td>h953_m1</td>
<td>20218</td>
<td>443</td>
<td>549</td>
<td>14</td>
</tr>
<tr>
<td>h953_m2</td>
<td>11900</td>
<td>259</td>
<td>329</td>
<td>9</td>
</tr>
<tr>
<td>h953_m8</td>
<td>9176</td>
<td>202</td>
<td>1012</td>
<td>26</td>
</tr>
<tr>
<td>p34392_m1</td>
<td>12563</td>
<td>273</td>
<td>207</td>
<td>5</td>
</tr>
<tr>
<td>p93791_m12</td>
<td>10717</td>
<td>236</td>
<td>5395</td>
<td>136</td>
</tr>
<tr>
<td>t512505_m7</td>
<td>4858</td>
<td>107</td>
<td>207</td>
<td>5</td>
</tr>
<tr>
<td>t512505_m8</td>
<td>19527</td>
<td>427</td>
<td>443</td>
<td>14</td>
</tr>
<tr>
<td>t512505_m9</td>
<td>16108</td>
<td>354</td>
<td>207</td>
<td>5</td>
</tr>
<tr>
<td>t512505_m14</td>
<td>50544</td>
<td>1100</td>
<td>207</td>
<td>5</td>
</tr>
<tr>
<td>t512505_m18</td>
<td>11771</td>
<td>354</td>
<td>207</td>
<td>5</td>
</tr>
<tr>
<td>t512505_m19</td>
<td>9176</td>
<td>202</td>
<td>1012</td>
<td>26</td>
</tr>
<tr>
<td>t512505_m31</td>
<td>42006</td>
<td>921</td>
<td>3281</td>
<td>83</td>
</tr>
<tr>
<td>u226_m7</td>
<td>8489</td>
<td>187</td>
<td>2403</td>
<td>62</td>
</tr>
</tbody>
</table>

Table 3: Simulated DAST Test Time (clock cycles) and its Test Time Models Prediction Values for Cores of SoC3

<table>
<thead>
<tr>
<th>Circuit Type</th>
<th>% Error of TMD to TD</th>
</tr>
</thead>
<tbody>
<tr>
<td>b10_1SC</td>
<td>1.3</td>
</tr>
<tr>
<td>b10_3SC</td>
<td>0.6</td>
</tr>
<tr>
<td>b15_1SC</td>
<td>0.0</td>
</tr>
<tr>
<td>b15_2SC</td>
<td>0.0</td>
</tr>
<tr>
<td>b15_16</td>
<td>0.0</td>
</tr>
</tbody>
</table>

5. Results

5.1. Area and Power Overhead

Table 1 reports area and power for components of EAS blocks for cores (modules) of the ITC’02 SoC Benchmarks. In Table 1, PI_BSR, SI_BSR, Buffer_in, and FSM refer to primary inputs BSR, scan input BSR, buffer input, and the FSM blocks of the EAS blocks. The data for the EAS block excluding the mandatory components of IEEE P1500 wrapper appears in the Adj_EAS column. The values reported for EAS blocks in Table 1 are only marginally greater than those reported in [10] owing to the modifications in their ASM illustrated in Fig. 4. Here, for the different benchmarks under consideration, the EAS blocks amount to an area equal to approximately 350 to 450 2-Input NAND gates.

Table 2 reports area and power for components of EARA blocks for cores (modules) of the ITC’02 SoC Benchmarks. In Table 1, PO_BSR, SO_BSR, Buffer_in, and FSM refer to primary outputs BSR, scan output BSR, buffer input, and the FSM blocks of the EARA blocks. Again, similar to the EAS blocks, the data for the EARA blocks excluding the mandatory components of IEEE P1500 wrapper is given in the Adj_EARA column. Here, for different benchmarks, the EARA blocks amount to an area equalling that of 250 to 300 2-Input NAND gates. Thus, the total additional area for our improved DAST methodology is minimal as it amounts to the equivalent of about 400-600 2-Input NAND gates. Adj_EAS, Adj_EARA, and IEEE P1500 BSR areas for ITC’02 Benchmarks’ modules are plotted in Fig. 7.

5.2. Test Time Overhead

The test time, using our new DAST, for the SoC3 cores is tabulated in Table 3. In this table, TP, SE, PI, PO, SI, and TD denote the number of test patterns, maximum number of flip-flops in the scan chain(s), functional input numbers, functional output numbers, scan input numbers and the simulated DAST test time (clock cycles), respectively. In the same table, TS denotes the theoretical lower bound test time for a serial connection in a conventional external ATE-based approach, as given by Eqns. 4, 5, and 6.
ATPG/scan-based testing can be performed with minimal area requirements. In addition, we developed test time models from the same high test-quality, ease of use, and broad applicability characteristics associated with the conventional ATE-based scan testing methodology.

7. References


Table 4: Predicted Test Time (clock cycles) for ITC’02 SoC Benchmarks Modules in DAST Methodology with EARA

<table>
<thead>
<tr>
<th>ITC’02 cores</th>
<th>SI</th>
<th>PI</th>
<th>PO</th>
<th>SE</th>
<th>TP</th>
<th>TS</th>
<th>TMD</th>
<th>% Overhead of TMD to TS</th>
</tr>
</thead>
<tbody>
<tr>
<td>5281 m9</td>
<td>6</td>
<td>194</td>
<td>223</td>
<td>34</td>
<td>118</td>
<td>71084</td>
<td>7845</td>
<td>2.2</td>
</tr>
<tr>
<td>595 m5</td>
<td>32</td>
<td>39</td>
<td>303</td>
<td>45</td>
<td>110</td>
<td>228196</td>
<td>228345</td>
<td>0.7</td>
</tr>
<tr>
<td>595 m6</td>
<td>19</td>
<td>62</td>
<td>155</td>
<td>41</td>
<td>234</td>
<td>225420</td>
<td>228105</td>
<td>1.5</td>
</tr>
<tr>
<td>595 m21</td>
<td>32</td>
<td>35</td>
<td>320</td>
<td>54</td>
<td>126</td>
<td>302144</td>
<td>303911</td>
<td>0.6</td>
</tr>
<tr>
<td>7126 m1</td>
<td>8</td>
<td>355</td>
<td>520</td>
<td>100</td>
<td>334</td>
<td>6034084</td>
<td>6039769</td>
<td>0.9</td>
</tr>
<tr>
<td>7126 m2</td>
<td>16</td>
<td>85</td>
<td>139</td>
<td>319</td>
<td>422</td>
<td>2276479</td>
<td>2282395</td>
<td>0.3</td>
</tr>
<tr>
<td>7126 m3</td>
<td>14</td>
<td>139</td>
<td>273</td>
<td>43</td>
<td>134</td>
<td>533511</td>
<td>557290</td>
<td>4.4</td>
</tr>
<tr>
<td>1023 m1</td>
<td>14</td>
<td>139</td>
<td>273</td>
<td>43</td>
<td>134</td>
<td>154712</td>
<td>156907</td>
<td>1.5</td>
</tr>
<tr>
<td>1023 m2</td>
<td>2</td>
<td>221</td>
<td>215</td>
<td>84</td>
<td>74</td>
<td>45868</td>
<td>47609</td>
<td>3.7</td>
</tr>
<tr>
<td>1023 m4</td>
<td>2</td>
<td>182</td>
<td>129</td>
<td>166</td>
<td>92</td>
<td>141474</td>
<td>145235</td>
<td>2.7</td>
</tr>
<tr>
<td>1023 m10</td>
<td>1</td>
<td>301</td>
<td>377</td>
<td>13</td>
<td>29</td>
<td>22858</td>
<td>23273</td>
<td>1.8</td>
</tr>
<tr>
<td>953 m1</td>
<td>4</td>
<td>112</td>
<td>152</td>
<td>348</td>
<td>341</td>
<td>579562</td>
<td>584735</td>
<td>0.8</td>
</tr>
<tr>
<td>953 m2</td>
<td>4</td>
<td>112</td>
<td>152</td>
<td>348</td>
<td>341</td>
<td>579562</td>
<td>584735</td>
<td>0.8</td>
</tr>
<tr>
<td>953 m8</td>
<td>32</td>
<td>35</td>
<td>320</td>
<td>54</td>
<td>126</td>
<td>302144</td>
<td>303911</td>
<td>0.6</td>
</tr>
<tr>
<td>334352 m1</td>
<td>1</td>
<td>15</td>
<td>94</td>
<td>800</td>
<td>210</td>
<td>209576</td>
<td>212525</td>
<td>1.4</td>
</tr>
<tr>
<td>303751 m12</td>
<td>46</td>
<td>301</td>
<td>93</td>
<td>93</td>
<td>391</td>
<td>1977896</td>
<td>1998855</td>
<td>1.5</td>
</tr>
<tr>
<td>303751 m19</td>
<td>46</td>
<td>301</td>
<td>93</td>
<td>93</td>
<td>391</td>
<td>1977896</td>
<td>1998855</td>
<td>1.5</td>
</tr>
<tr>
<td>512505 m7</td>
<td>1</td>
<td>122</td>
<td>36</td>
<td>514</td>
<td>608</td>
<td>462230</td>
<td>476223</td>
<td>3.0</td>
</tr>
<tr>
<td>512505 m8</td>
<td>3</td>
<td>108</td>
<td>147</td>
<td>172</td>
<td>1023</td>
<td>4634560</td>
<td>4845815</td>
<td>0.3</td>
</tr>
<tr>
<td>512505 m9</td>
<td>1</td>
<td>151</td>
<td>132</td>
<td>530</td>
<td>199</td>
<td>151624</td>
<td>154363</td>
<td>1.8</td>
</tr>
<tr>
<td>512505 m14</td>
<td>1</td>
<td>751</td>
<td>381</td>
<td>1225</td>
<td>278</td>
<td>761111</td>
<td>767514</td>
<td>0.5</td>
</tr>
<tr>
<td>512505 m15</td>
<td>1</td>
<td>406</td>
<td>132</td>
<td>386</td>
<td>151</td>
<td>182247</td>
<td>185729</td>
<td>1.9</td>
</tr>
<tr>
<td>512505 m16</td>
<td>2</td>
<td>221</td>
<td>215</td>
<td>84</td>
<td>74</td>
<td>45868</td>
<td>47609</td>
<td>3.7</td>
</tr>
<tr>
<td>512505 m23</td>
<td>2</td>
<td>99</td>
<td>124</td>
<td>1372</td>
<td>532</td>
<td>1546868</td>
<td>1602414</td>
<td>3.5</td>
</tr>
<tr>
<td>512505 m24</td>
<td>2</td>
<td>163</td>
<td>129</td>
<td>1669</td>
<td>428</td>
<td>874619</td>
<td>884495</td>
<td>1.1</td>
</tr>
<tr>
<td>512505 m31</td>
<td>20</td>
<td>151</td>
<td>310</td>
<td>1550</td>
<td>3370</td>
<td>1465430602</td>
<td>14847851</td>
<td>0.7</td>
</tr>
</tbody>
</table>

In Table 3, we also included TMD obtained from the DAST time models given by Eqns. 1-3. From Table 3, our test-time models (Eqns 1-3) closely follow actual simulated test times. Table 4 reports the estimated DAST test time based on Eqns. 1-3 for the ITC’02 SoC Benchmark modules and the percentage overhead compared to a conventional external ATE-based approach. Note that in Table 4, we have not included the bi-directional pins of a module. Instead, we counted these pins both in the primary inputs and outputs. From Table 4, the increase in test time with DAST using EARA is always less than 5% when compared to a lower bound serial connection in a conventional external ATE-based approach.

6. Conclusions

We developed the concept and implementation of an Embedded Autonomous Results Analyzer (EARA) to be used with a Dedicated Autonomous Scan-based Testing (DAST) Methodology. With an area overhead equal to less than 300 2-input NAND gates, EARA enables on-chip comparison of test results for an SoC designed with an ATPG/scan-based DFT methodology. Embedded results evaluation is enabled by sending the expected test results in addition to the test stimulus while EARA operates in synchrony with an Embedded Autonomous Sequencer (EAS).

We implemented EARA components in TSMC 0.18 µm for different cores of the ITC’02 Benchmarks and reported their area requirements. In addition, we developed test time models for our scheme and, for comparison purposes, developed test time models corresponding to a serial connection that one would expect to find in a conventional external ATE-based methodology. The test time penalty in using DAST and EARA is typically less than 2%.

Overall, the advantage of DAST with the EARA is that ATPG/scan-based testing can be performed with minimal external components providing a basis for SoC testing that is cost-effective, scalable, and portable. Since our scheme is based on the conventional scan/ATPG approach, our scheme benefits from the same high test-quality, ease of use, and broad applicability.