A 2-D compaction Method using Macro block for Post-Silicon Validation

Won Jung, Hyunggoy Oh, Dongho Kang and Sungho Kang
Dept. of Electrical and Electronic Engineering
Yonsei University
Seoul, Korea

1{dino12h, kyob508, fourier2}@soc.yonsei.ac.kr, 2shkang@yonsei.ac.kr

Abstract— The post-silicon validation has been an important step as the complexity of system on chip (SoC) increases. Conventional trace buffer based debug methods offer consecutive observability and real time debug, but the size constraint of the trace buffer still is a challenge. The proposed method uses 2-D compaction for expanding the depth of observation window in a trace buffer. Moreover, the macro block, which is used with 2-D compaction, offers tolerance to various error patterns as a virtual window. The errors identified by the 2-D compaction using the macro block are selectively captured by using the new tag map. The experimental results show that the proposed method enables the reduction of error misidentification.

Keywords-- 2-D compaction; macro block; post-silicon debug.

I. INTRODUCTION

To develop semiconductor chips with high performance and low power, chip design technologies have been complex and time to market and the size of transistors has been shrinking. However, errors which have not been found in the design phase are gradually increased. For this reason, post-silicon validation has become increasingly important to develop SoCs [1], [2]. Trace buffer based debug methods have been studied to enhance observability for post-silicon validation [3]. In this paper, the proposed method enables error identification using a trace buffer.

II. PROPOSED ALGORITHMS

A. Overview Of Proposed Architecture

The proposed method uses the 2-D compaction technique used in [4]. The 2-D compactor consists of a multi-input signature register (MISR) and cycling register. The 2-D compactor generates MISR signatures and cycling register signatures. The cycling register is a technique that generates a signature by XORing the data words which are signals tapped from the circuit under debug (CUD). By intersections of those signatures, suspect data words in signatures can be identified. Additionally, the 2-D compaction technique significantly extends the capacity of a trace buffer. [4] shows error identification capability during three debug session but there are some drawbacks. The error identification capability in [4] varies considerably with error patterns and the tag bit generation for selective capturing suspect data words is inefficient due to the size constraint of the trace buffer. Our proposed method not only performs three debug sessions, but also maintains the ability of 2-D compaction regardless of error patterns and reduces the amount of tag bits significantly. Fig. 1 illustrates the proposed architecture. In the first debug session, the error rate is roughly estimated to the size of observation window through parities by using the parity generator, which is the same as in [4]. In the second debug session, the 2-D compactor compacts observed signals based on the macro block with the comparator. In the third debug session, the suspect data words are captured based on the tag map, which consists of parent bits and child bits after the second debug session.

B. 2D-Compaction and Macro Block

The proposed method uses golden signatures to compare with the signatures generated from the MISR in real-time. The golden signatures must be large enough to cover the clock cycles in which the system failure occurred. The golden signatures can be gotten by the pre-silicon verification or design simulation. After the comparator compares both
signatures, a parent tag bit, which is 1bit (match: 0, mismatch: 1), is generated by the comparator and stored into the trace buffer. If the current signature is different from its golden signature, the parent bit becomes 1 and the counter of the macro block generator is updated. The macro block is a virtual window for collecting the suspect data words in suspect signatures, which are un-identical with its golden signature. Whenever the counter value is the same as the height of macro block, which has been defined after the first debug session, the cycling register generates cycling register signatures in the vertical direction based on the macro block, consisting of the suspect data words, which have maintained in the reserved area in the trace buffer. The cycling register signatures then are stored in the cycling register signature area in the trace buffer. When the last golden signature is consumed, the second debug session is finished and the parent tag bits and cycling register signatures are unloaded from the trace buffer for error identification.

C. Tag Map and Capture Trigger

The tag map, which consists of parent tag bits and child tag bits, is generated by the error identification using the parent bits and cycling register signatures, which have been unloaded from the trace buffer after the second debug session. The child tag bits denote data words in suspect signatures, and they are generated in case that the parent bits are 1. Additionally, if one of the data words is identified as the suspect data word, its child tag bit becomes 1. The number of child tag bits depends on the expansion ratio, which is calculated by the equation in [4]. Fig. 3 shows an example of a tag map with parent and child tag bits. The proposed tag map enables the significant reduction of the number of tag bits without any compression. The tag map is uploaded to the trace buffer before the third debug session. The capture trigger selectively captures data words based on the tag map.

III. EXPERIMENTAL RESULT

The experiment results include a comparison of the proposed method with [4]. For the comparison, the proposed debug architecture on the Amber core [5] is implemented as a behavioral level. Fig. 4 (a) shows the number of error misidentifications by comparing the proposed method with [4]. As can be seen, in the random error, the error misidentification in the proposed method was less than that in [4] because the suspect data words, which were determined by parent tag bits, zoomed in with the low-rate compaction based on the macro block by the cycling register. Fig. 4 (b) indicates the number of tag bits in the two methods. As shown in Fig. 4 (b), the number of tag bits, which have been generated by the tag map in the proposed method, was significant reduced compared with the tag bits with lossy compression (x1, x2, x4) in [4]. Table I shows the experimental results using a 2-KB trace buffer with different error rates and error patterns (random and clustered error). In Table I, the number of the error misidentifications in the proposed method was small compared with [4]. Moreover, the proposed method maintained the error identification capability in case of clustered errors, but in [4], the ability was deteriorated significantly. The proposed method can maintain the error identification capability, because the macro block generator collects suspect data words, which is similar to clustered error pattern, and the data words are compacted with the lower-rate by the cycling register.

<table>
<thead>
<tr>
<th>Error Rate (%)</th>
<th>Expansion Ratio</th>
<th># of Error Misidentifications [4]</th>
<th># of Error Misidentifications [Proposed]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Random Error</td>
<td>Clustered Error</td>
</tr>
<tr>
<td>0.016</td>
<td>112</td>
<td>10</td>
<td>1091</td>
</tr>
<tr>
<td>0.051</td>
<td>78</td>
<td>21</td>
<td>1327</td>
</tr>
<tr>
<td>0.097</td>
<td>67</td>
<td>38</td>
<td>1672</td>
</tr>
<tr>
<td>0.513</td>
<td>21</td>
<td>47</td>
<td>946</td>
</tr>
<tr>
<td>1.387</td>
<td>12</td>
<td>70</td>
<td>859</td>
</tr>
</tbody>
</table>

IV. CONCLUSION

In this paper, the proposed debug method enables high error identification capability and maintains the capability regardless of the error patterns. Moreover, the tag map in the proposed method can reduce the number of tag bits, which affect debug time. As a result, the proposed debug method allows precise debug for post-silicon validation.

ACKNOWLEDGMENT

"This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No. 2015R1A2A1A13001751)."

REFERENCES