UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Master Thesis Proposal

Convolutional Compaction for Faster-than-At-Speed-Test (FAST)

Yuan Zhang
6766441
Electrical Engineering

Submitted to:
Prof. Sybille Hellebrand

Supervisor: Alexander Sprenger

1. Introduction

The test with nominal speed is an effective method for stuck-at faults [1]. But it has no way to deal with small delay faults which cannot be propagated over long paths. Those called hidden delay faults (HDFs) are considered as a potential threat, although they do not affect the function of the chips.

To thoroughly detect this threat, the method of Faster-than-At-Speed Test (FAST) has been proposed. The principle of FAST is to increase the test frequency, hence an early observation time is used, such achieving the detection of HDFs. However, it follows that some outputs have not yet reach a stable state and those resulting unknown values (X-values) are challenging for test response compaction. Because the X-values cannot be uniquely predicted, it may lead to error masking during the compaction. On the other hand, for example consider the use of classical signature analyzers, such as multiple-input signature registers (MISRs), the signature bites whose expected values are Xs will be ignored, because the comparison of this excepted signature with the actual signature cannot be made.

Especially for Built-In Self-Test (BIST) and embedded test, the test response compaction is an important part. For example, the X-canceling MISR scheme can dramatically reduce the test response data volume and the number of pins required to collect test response from a chip, but it might not be an efficient scheme in presence of high ratio X-values. Therefore, the convolutional compaction [2] scheme will be investigated on its application to FAST.

2. Introduction of Convolutional Compaction

Figure 1 shows a typical example for the structure of a convolutional compactor with 16 inputs (observing 16 scan chain outputs), two outputs, and 6 registers – three per output. This compactor uses the 3 out of 6 code (for brevity denoted as 3/6). It means that each input connects three of the total six registers with XOR gates. To guarantee that the inputs are not connected to the same register, every input must have its own unique polynomial. For example, input 1 connects the register 1,2,3, so it employs the polynomial $P_1 = X^1+X^2+X^3$. Finally, all registers are sequentially connected by XOR gates to the output and the number of outputs depends on the compaction ratio. When a test response is shifted into the compactor using scan chain 1-16, after three shift clocks, the compacted test response is generated. During this process, a special case may appear. For example, the polynomial $X^2+X^3+X^5$ and $X^1+X^2+X^4$ will cancel each other after one shift clock. Hence polynomials with the above characteristics will be treated as a group and only one polynomial of each group can be selected for an input. At last according to the used code k/M, employed polynomial, and the number of the output b, the structure of compactor is determined, and it can have upto S inputs.

$$S=\sum_{i=1}^{b}\binom{M-i}{k-1}$$

In the absence of X values, a convolutional compactor can detect faults from one, two, or any odd number of scan chains and the faults can be injected at the same time

or at different shift cycles. Because any two polynomials of inputs are different, and any odd number of faults cannot cancel each other [2].

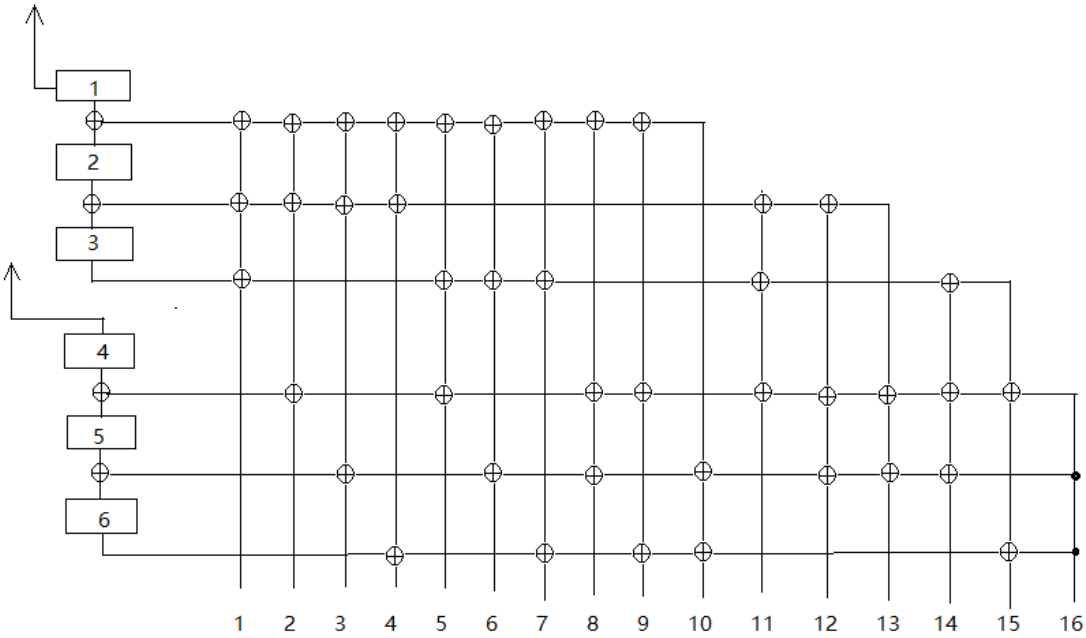

Fig.1 Example of a convolutional compactor [2]

The compactor can detect a single fault in the presence of a single X-value produced by another scan cell. Because the unique polynomial of every input, the X-value cannot mask the single fault. But in the presence of multiple X values the observability of scan cell is influenced by the compaction ratio. In general, as the compaction ratio increases, the observability decreases [2].

3. The goal of this thesis

1. According to the given conditions, such as the amount of input, the convolutional compactor is implemented with C++. Using this compaction simulation, the compacted test responses can be generated. At last, with the analysis of the simulation results I can prove whether convolutional compactor is suitable for FAST.
2. If this scheme is not suitable for FAST, I will optimize the compactor scheme to adapt to FAST.

4. Time schedule

Task	Duration	Description
1	Week 1 – Week 8	Implement the convolutional compactor
2	Weed 6 – Week 14	Compaction simulation
3	Week 15 – Week 16	Result analysis
4	Week 16 – Week 22	The compaction scheme optimization
5	Week 1 – Week 24	Paper work

References:

[1] Laung-Terng Wang, Cheng- Wen Wu, Xiaoqing Wen: VLSI Test Principles and Architectures: Design for Testability (Morgan Kaufmann Series in Systems on Silicon) ISBN: 0123705975

[2] J. Rajski, J. Tyszer, C. Wang und S. M. Reddy, “Convolutional compaction of test responses”, in Proc. ITC 2003. Int. Test Conf, Bd. 1, Sep. 2003, S. 745–754. doi: 10.1109/TEST.2003.1270904.