Submitted to Third International Test Synthesis Workshop 1996:

Altering a Pseudo-Random Bit Sequence for Mixed-Mode Scan BIST

Nur A. Touba and Edward J. McCluskey

Center for Reliable Computing
Computer Systems Laboratory, ERL-460
Departments of Electrical Engineering and Computer Science
Stanford University
Stanford, CA 94305-4055

Designated presenter: Nur A. Touba

Telephone: (415) 723-1258
FAX: (415) 725-7398
E-mail: touba@shasta.stanford.edu

EXTENDED ABSTRACT

1. Introduction

One approach for built-in self-test (BIST) of circuits with scan is to use a linear feedback shift register (LFSR) to shift a pseudo-random sequence of bits into the scan chain. When a pattern has been shifted into the scan chain, it is applied to the circuit-under-test (CUT) and the response is loaded back into the scan chain and shifted out into a signature register for compaction as the next pattern is shifted into the scan chain. Figure 1 shows a block diagram for this "test-per-scan" approach. Unfortunately, many circuits contain random-pattern-resistant (r.p.r.) faults which limit the fault coverage that can be achieved with this approach. One solution is to insert test points in the circuit to increase the detection probabilities of the r.p.r. faults. However, test points require that the function logic be modified, and more importantly, they add delay to the circuit which can degrade system performance. Another solution is to use "mixed-mode" testing in which deterministic patterns are used to detect the r.p.r. faults that the pseudo-random patterns miss. This paper presents a new approach for mixed-mode testing in which deterministic test cubes (i.e., deterministic patterns in which the unspecified inputs are left as *X's*) are embedded in the pseudo-random sequence of bits. Logic is added at the serial output of the LFSR to alter the pseudo-random bit sequence so that it is guaranteed to contain patterns that detect the r.p.r. faults. The new approach can be used for either a "test-per-scan" or "test-per-clock" scheme and is capable of providing complete fault coverage with a reasonable test length. For circuits with scan, no function logic modification is required, and there is no system performance degradation. A procedure is described for embedding the deterministic test cubes in the pseudo-random bit sequence in a way that minimizes hardware area.

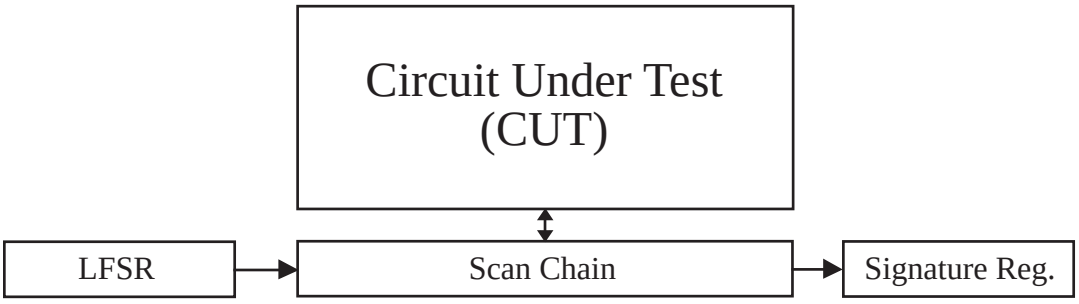

Figure 1. Block Diagram

2. Hardware for Altering the Pseudo-Random Bit Sequence

The pseudo-random sequence of bits that is generated at the serial output of the LFSR is altered by adding logic that fixes certain bits in the sequence to be a '1' or a '0'. As illustrated in Figure 2, some decoding logic is used to detect certain states of the LFSR and fix the value of the serial output to either a '1' or a '0'. If the LFSR has a primitive feedback polynomial and the test length is less than 2^{n-1}, where n is the number of stages in the LFSR, then each bit in the sequence can be uniquely decoded by the state of the LFSR. The hardware required to decode only one state of the LFSR and no others would be an n-input AND gate, where n is the number of stages in the LFSR, however, the hardware required to decode all the states of the LFSR in which the first 3 bits are all '1' for example, would be only a 3-input AND gate. The key to minimizing the hardware area is to embed the deterministic test cubes in the bit sequence in a way that minimizes the amount of decoding logic that is required. This paper presents a procedure for accomplishing this.

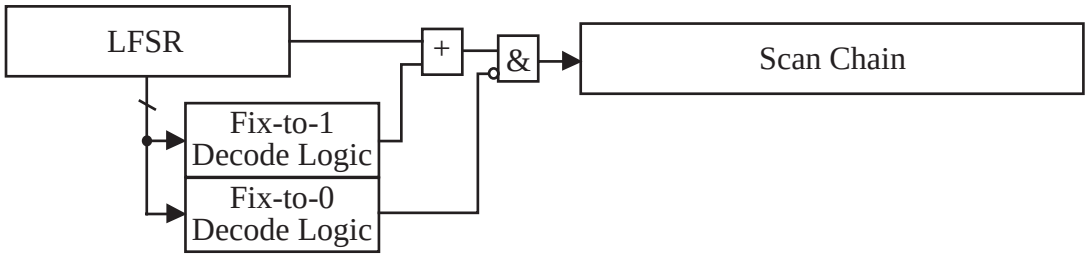

Figure 2. Logic for Altering the Pseudo-Random Bit Sequence

3. Synthesis Procedure

The goal of the synthesis procedure is to minimize the amount of logic that is required to alter the pseudo-random sequence of bits generated by an LFSR so that it detects all the faults in the circuit-under-test. The first step of the procedure is to simulate the LFSR for the given test length to obtain the pseudo-random sequence of bits that is shifted into the scan chain. Fault simulation is performed for each pattern that is generated in the scan chain (this can be on either a "test-per-scan" or "test-per-clock" basis), and the first pattern that detects each fault is recorded. The undetected faults are the faults for which deterministic test cubes must be embedded into the bit sequence.

The next step of the procedure is to determine which bits in the sequence can be altered without reducing the fault coverage, i.e., causing a fault that is currently detected to longer be detected. A

set of bits that guarantees detection of all of the currently detected faults is given by the set of bits in each of the first patterns that detected each fault. However, many of these bits are actually don't cares ($X's$)with respect to detecting each fault. To determine which bits are don't cares, a bit-flipping procedure is used. Each of the bits in the first pattern that detected each fault is successively flipped (complemented) and three-valued simulation is performed to see the fault is still detected: if so, then the bit can be replaced by an X. This procedure inserts $X's$ into each of the first patterns that detected each fault. If the set of specified bits in each of the first patterns that detected each fault is not altered, then the bit sequence is guaranteed to detect all of the currently detected faults. This set of specified bits will be referred to as the "essential bits." The other bits in the sequence are effectively don't cares and can be altered to embed deterministic test cubes.

Automated test pattern generation (ATPG) is performed for the undetected faults to generate test cubes that detect them. A procedure is described for searching the bit sequence to find an optimal location in the sequence to embed each test cube. For each bit in the sequence that needs to be fixed to a 1 (0) in order to embed a test cube, the minterm corresponding to the state of the LFSR when the bit is generated is placed in the on-set for the fix-to-1 (fix-to-0) decoding logic. For each "essential bit" that is a 0 (1), the minterm corresponding to the state of the LFSR when the bit is generated is placed in the off-set for the fix-to-1 (fix-to-0) decoding logic. The on-set and off-set for the fix-to-1 and fix-to-0 decoding logic form an incompletely specified function which is passed to a logic synthesis tool to generate the decoding logic. Because the vast majority of the bits in the pseudo-random sequence are not essential for detecting any faults, there are a large number of don't cares which results in a small amount of decoding logic.

By finding a location in the pseudo-random bit sequence where only a small number of bits need to be altered to embed each deterministic test cube, the hardware required for this mixed-mode approach is significantly minimized. Results will be presented comparing the BIST hardware required by the proposed approach with other approaches.

A hardware tradeoff that is made possible by the proposed approach is that a smaller LFSR can be used for generating the pseudo-random bit sequence. This may cause some faults to not be detected because of linear dependencies in the patterns that are generated, but deterministic patterns for those faults can be embedded at the expense of additional decoding logic. Data will be presented showing the how much decoding logic is required for different size LFSR's.