

Built-In Self-Test Design of Motion Estimation Computing Array

Donglin Li, Mingzeng Hu¹ and Otmame Ait Mohamed
 Department of Electrical and Computer Engineering
 Concordia University, Montreal, Canada, H3G 1M8
 li_don@ece.concordia.ca

Abstract—This paper presents an implementation of Built-In Self-Test (BIST) design of Motion Estimation (ME) Computing Array, which is the main part of MPEG-2 Video Encoder. The goal of the design is to offer high reliability for the coding system. Our design is carried out on gate level in both VHDL and Verilog HDL and then synthesized into FPGA. Design verification is also performed using logic simulation and fault simulation.

I. INTRODUCTION

MPEG-2 [1] is an important standard for the compression of moving pictures established by Moving Picture Experts Group of International Standard Organization. As a commercial chip, it is absolutely necessary for MPEG-2 Video Encoder to introduce design for testability (DFT), whose goal is to increase the ease with which a device can be tested, in order to guarantee high reliability of the system. Our goal in this paper is to present a design which offers high reliability. In order to reduce the cost and complexity of the design, we will focus on Motion Estimation (ME), which is the main part of the encoder.

After analyzing and comparing many DFT technologies, we choose Built-In Self-Test (BIST) [2] [3] as the method adopted by the design. By putting test pattern generation and response analysis logic into the circuit, the circuit with BIST function can generate test stimulation and analyze test response without outside support, and then can make the test and diagnosis of digital system quick and effective.

First, this paper studies the BIST technology. Then, based on the analysis of structural features of ME Computing Array and its basic process element (PE), BIST design is divided into two parts: the test of PEs and the test of data channels. After the discussion of how this design is implemented and verified, a conclusion is given in the end.

II. BIST TECHNOLOGIES

There are many approaches to DFT, which can be divided into three categories: Ad hoc (problem oriented), structured and BIST. Among them, BIST has obvious advantages that make itself the number one choice of this project. For example, with BIST, there is no need to buy expensive test equipment, like Automatic Test Equipment, and it can efficiently reduce the test cost.

In general, BIST technique refers to the capability of a chip board, or system to test itself. The goal of BIST is to add components to a design that will allow it to test itself. Generally, three parts are added to the circuit under test (CUT) to achieve BIST design: test pattern generator (TPG), output response analyzer (ORA) and test controller. Fig.1 shows the general module of BIST.

More than one technique can be applied to generate test pattern, which can be classified in the following groups: pseudoexhaustive testing [4], pseudorandom testing [5], weighted random testing [4], deterministic tests [6] and mixed mode pattern generation [7], [8]. In this paper, we use the pseudorandom test technique, which can be implemented with a Linear Feedback Shift Register (LFSR) and whose structure will be discussed later. ORA, the signature analyzer, can also

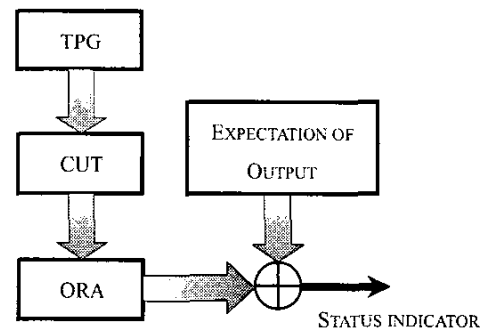


Fig. 1. General module of BIST

be implemented with a LFSR.

Two general structures of an n-bit LFSR, which are functionally equal [9], are shown in Fig.2. From the graphs, we can see the typical parts of LFSR are D-flip-flops and XOR gates, which form the shift register and feedback logic respectively. The value of 'h' here is either '1' or '0', which means the path exists or not. LFSRs have different applications for BIST. First, Maximal Length LFSRs can be used as pseudorandom pattern generators. They can also be used as signature analyzers, because different input sequences correspond to different output sequences. We can simply tell if the input sequences are that we have expected or not through the output ones. To form a signature analyzer, an

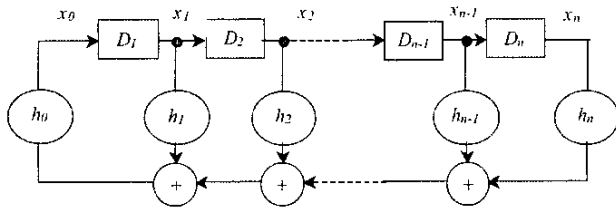


Fig. 2 (a). External-XOR structure of LFSR

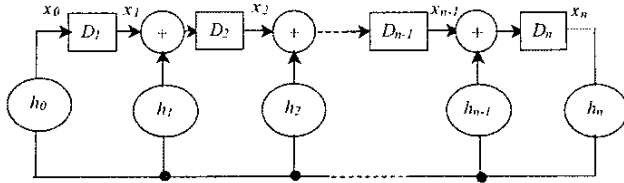


Fig. 2 (b). Internal-XOR structure of LFSR

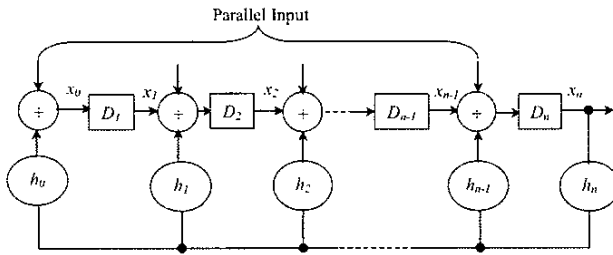


Fig. 3. Internal-XOR structure of MISR

extra XOR gate and an input are needed to add to a Maximal length LFSR. This type of signature analyzer is called Serial Input Signature Analyzer (SSA). If multiple inputs are added parallel to a Maximal length LFSR, then forms a Parallel Input Signature Analyzer (PSA). An LFSR of this structure is called a Multi-Input LFSR (MISR) as shown in Fig.3.

III. FAULT MODELS

Actual physical defects in a circuit / chip are too many to be considered individually. So some logical fault models are introduced in practice, which can drastically reduce the number of faults to be handled and also can cover most of the possible physical failures.

Here are some common fault models [10]: the stuck-at fault model, the stuck-open fault model, the bridging fault model and the delay fault model. After careful study, we choose the single stuck-at fault model and the transitional delay fault model. The single stuck-at fault model is the most widely used fault model in the industry. It is simple to use during test generation and fault simulation; research has shown that it covers a large percentage of physical defects; and other fault models, can be mapped into sequences of single stuck-at faults. There are however, some defects that cannot be covered by the single stuck-at model, that's why we introduce transitional delay fault model as a supplementary.

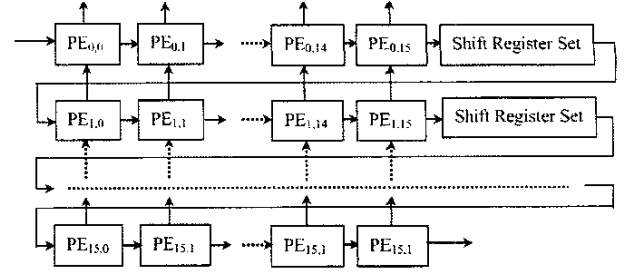


Fig. 4. Systolic array architecture of ME Computing Array

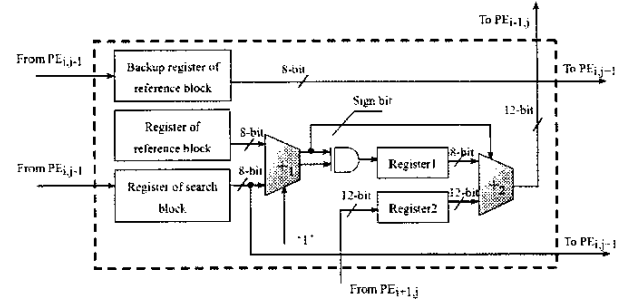


Fig. 5. Architecture of PE

IV. BIST DESIGN OF ME COMPUTING ARRAY

A. Architecture of ME Computing Array

The algorithm for motion estimation studied in this paper is Full search Block Matching Algorithm (FBMA) [11] [12]. This algorithm is implemented using a systolic array as shown in Fig.4. ME Computing Array is made up of Process Elements (PEs) with the size of (16, 16). Besides, some shift register sets are added into the array to meet the needs of Motion Estimation algorithm.

PE is made up of two adders, five registers and one XOR gate. Also there are three array channels connecting PEs. Fig.5 shows the architecture of PE.

The BIST design of ME Computing Array can be divided into two parts: BIST design of PE and BIST design of array channels.

B. BIST design of PE

In order to simplify the test design, our BIST design of PE focuses on the two adders, using single stuck-at fault model. Adder1, the left adder in Fig.5, is an 8-bit ripple carry adder (RCA); and adder2, the right one, is a 12-bit adder cascaded by three 4-bit carry lookahead adders (CLAs). We will discuss the test pattern generation, the output response analysis, the collecting & comparing and the test strategy.

1) Test Pattern Generation of Adder1

It can be proved [13] that single stuck-at faults of an 8-bit RCA can be covered by the test patterns shown in TABLE I. These 17-bit test patterns can be simplified as 3-bit patterns by some means and generated by a 3-bit LFSR, as shown in

Fig. 6, after adding some extra logics.

2) Test Pattern Generation of Adder2

It can be proved [13] that single stuck-at faults of a 4-bit CLA can be covered by the test patterns shown in TABLE II. If these 9-bit test patterns are generated by a 9-bit LFSR, it will need 512 test clocks, but after some improvement these test patterns can be generated by a 5-bit shift register and a 4-bit length-changing LFSR, which will only cost 32 test clocks and therefore can dramatically shorten the test period. What is more, by some means, single stuck-at faults of both the three CLAs and the carry chain connecting them can be covered by these test patterns. Fig.7 shows the structure of the 4-bit length-changing LFSR.

3) Output Response Analysis

Output response analysis of the two adders can be implemented by a PSA formed by a 12-bit MISR, the structure of which has been discussed above. The outputs of the adders with the input test patterns can be calculated, and if the PSA is initialized properly, the output sequence of PSA after the test should be all '0', which is easy to check.

4) Collecting & Comparing Construct

After testing, the output sequence of the PSA in each PE will be sequentially pumped out of the array into one Collecting & Comparing Construct, at which the test results will be collected and compared with the expectations calculated in advance, and a final test conclusion will be given according to the results.

5) Test Strategy of PE

As for a BIST design, test construct must be implemented inside the same chip of the CUT. The principle of the design is to add extra hardware spending as little as possible. One of the ways is to multiplex some of the existing parts of CUT. Details of how the multiplexing is implemented in PE are as following:

- Register of Reference Block and Register of Search Block can be reconstructed as LFSRs to generate the two addends of adder1.
- Register1 and Register2 can be reconstructed as LFSRs to generate the two addends of adder2
- Backup Register of Reference Block can be reconstructed as Signature Analyzer to perform the output response analysis of adder1 and adder2.

The two adders are tested sequentially.

C. BIST Design of Array Channels

From Fig.5, we can see that there are three channels to be tested. Transitional delay fault model is adopted here. The faults of this model can be covered by the following test patterns: 000...000 \rightarrow 000...000 \rightarrow 111...111 \rightarrow 111...111 \rightarrow 000...000.

These test patterns are generated by the test controller directly and the test results are verified by the Collecting & Comparing Construct mentioned above.

D. Architecture of BIST of ME Computing Array

As a whole, there are three modules: the reconstructed ME

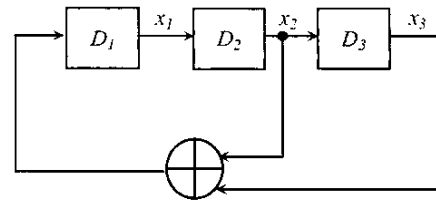


Fig. 6. Structure of a 3-bit LFSR

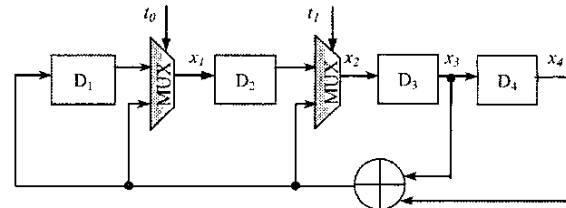


Fig. 7. Structure of the 4-bit length-changing LFSR

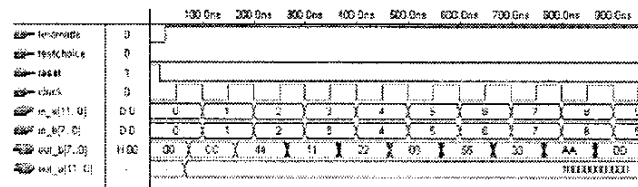


Fig. 8. A waveform for logic simulation

computing array with TPGs and ORAs, Collecting & Comparing Construct and Test Controller. The functions of Test Controller are to control the chip to be in and out of the test mode and to control the test process.

E. Test Process

First, all the PEs are tested in parallel and inside each PE adders are tested sequentially. Finally, the three array channels are tested in parallel.

When the first test result is sent out of the array (at the 9th test clock), the Collecting & Comparing Construct will begin working and it will end 8 test clocks later than the time that all the results have been pumped out. The whole test period is 1286 test clocks.

V. DESIGN IMPLEMENTATION AND VERIFICATION

The design is carried out top-down at the gate level in the system of Max+plus II in VHDL and Verilog HDL and synthesized into the FPGA FLEX10K100 (100,000 logic gates) of ALTERA.

The design verification is performed in two steps: logic simulation and fault simulation.

Logic simulation is executed in Max+plus II by means of waveform and the design finally passes both the unit test and the integrated test. Fig.8 shows an example of the waveforms for logic simulation.

Fault simulation, executed in Verifault of Cadence, has two purposes: one is to verify the validity of the test patterns used

TABLE I
TEST CODE SET FOR A 8-BIT RCA WITH A SINGLE FAULT

| Test pattern | 1 | 2 | 3 | 4 |
|----------------|---------|---------|---------|---------|
| $a_7...a_1a_0$ | 0...000 | 0...101 | 1...010 | 1...111 |
| $b_7...b_1b_0$ | 0...000 | 0...101 | 1...010 | 1...111 |
| c_i | 0 | 0 | 1 | 1 |

| Test pattern | 5 | 6 | 7 | 8 |
|----------------|---------|---------|---------|---------|
| $a_7...a_1a_0$ | 1...111 | 1...111 | 0...000 | 0...000 |
| $b_7...b_1b_0$ | 0...000 | 0...000 | 1...111 | 1...111 |
| c_i | 1 | 0 | 1 | 0 |

TABLE II
TEST CODE SET FOR A 4-BIT CLA WITH A SINGLE FAULT

| $a_1a_2a_1a_0$ | $b_3b_2b_1$ | $a_1a_2a_1a_0$ | $b_3b_2b_1$ | $a_1a_2a_1a_0$ | $b_3b_2b_1$ |
|----------------|-------------|----------------|-------------|----------------|-------------|
| c_i | b_0 | c_i | b_0 | c_i | b_0 |
| | 0000 | | | | 1110 |
| | 0001 | | | | 1010 |
| | 0011 | | 0001 | 00100 | 0110 |
| | 0111 | | 0011 | | 0010 |
| 00001 | 1111 | 00010 | 0111 | | |
| | 1110 | | 1111 | | 1100 |
| | 1101 | | 1101 | 01000 | 0100 |
| | 1011 | | 1011 | | |
| 11111 | 0000 | | | 10000 | 1000 |

in the test of adders and the other is to verify if the simplification in test of PE is reasonable.

A. Fault Simulation of adders

Single stuck-at faults of the two adders are simulated using the test patterns mentioned above. From the fault coverage report shown in Fig.9, we can see the coverage ratio is 100%. This perfectly proves the validity of the test patterns.

B. Fault Simulation of PE

There are two reports shown in Fig.10. The first report tells us the sum of faults of the two adders plus the three registers in PE, the second tells us the sum of faults of the whole PE. Note that the faults of the three registers are covered by the test of array channels. The fault coverage of PE is 96%. That means the simplification in the BIST design of PE is reasonable.

VI. CONCLUSION

The simplification in BIST design of PE is proved to be reasonable, through which the extra hardware spending introduced by BIST and the time cost of test are reduced.

Original resources of the CUT are adequately multiplexed, effectively decreasing the cost for the added hardware.

The LFSR used to generate the test patterns of Adder2 is improved from a 9-bit LFSR to a 4-bit changing LFSR adding a 5-bit shift register, dramatically shortening the test period.

Finally, our design is synthesized into an FPGA, and fully verified through logic simulation and fault simulation.

| | Total # | Total % | Prime # | Prime % |
|---------------|---------|---------|---------|---------|
| Untestable | 2 | | 2 | |
| Drop_detected | 322 | 100.0 | 296 | 100.0 |
| Detected | 0 | 0.0 | 0 | 0.0 |
| Potential | 0 | 0.0 | 0 | 0.0 |
| Undetected | 0 | 0.0 | 0 | 0.0 |
| All | 322 | | 296 | |

Fig. 9. Fault coverage report for simulation of adders

| | Total # | Total % | Prime # | Prime % |
|---------------|---------|---------|---------|---------|
| Untestable | 92 | | 44 | |
| Drop_detected | 0 | 0.0 | 0 | 0.0 |
| Detected | 0 | 0.0 | 0 | 0.0 |
| Potential | 0 | 0.0 | 0 | 0.0 |
| Undetected | 446 | 100.0 | 362 | 100.0 |
| All | 446 | | 362 | |

| | Total # | Total % | Prime # | Prime % |
|---------------|---------|---------|---------|---------|
| Untestable | 92 | | 44 | |
| Drop_detected | 0 | 0.0 | 0 | 0.0 |
| Detected | 0 | 0.0 | 0 | 0.0 |
| Potential | 0 | 0.0 | 0 | 0.0 |
| Undetected | 490 | 100.0 | 378 | 100.0 |
| All | 490 | | 378 | |

Fig. 10. Fault coverage reports for simulation of PE

ACKNOWLEDGMENT

The authors wish to thank Yuzhuo Fu, Yizheng Ye and Jinxiang Wang from Harbin Institute of Technology (China) for their generous help and comments on this work.

REFERENCE

- [1] ISO/IEC 13818-2: 1995 (E)
- [2] E. J. McCluskey. Built-In Self-Test Technique. IEEE Design & Test of Computers, June 1993, pp. 9-77
- [3] Vishwani D. Agrawal, Charles R. Kime and Kewal K. Saluja. A Tutorial on Built-In Self-Test, Part1: Principles. IEEE Design & Test of Computers, March 1993
- [4] WANG, L.T. - McCLUSKEY, E.J.: Condensed linear feedback shift register (LFSR) testing - A pseudoexhaustive test technique. IEEE Trans. on Comp. Vol. C-35, Apr. 1986, pp. 367-370
- [5] BARDELL, P. - MCANNEY, W. H.- SAVIR, J.: Built-In Test for VLSI. New York: Wiley-Interscience, 1987
- [6] DAEHN, W. - MUCHA, J.: Hardware test pattern generators for built-in test. Proc. of IEEE ITC.1981, pp. 110-113
- [7] KOENEMANN, B.: LFSR - coded test patterns for scan designs. Proc. Europ. Test Conf., Munich, Germany, 1991, pp. 237-242
- [8] HELLEBRAND, S. - RAJSKI, J.- TARNICK, S.- VENKATARAMAN, S. -COURTOIS, B.: Built-In Test for Circuits with Scan Based on Reseeding of Multiple-Polynomial Linear Feedback Shift Registers. IEEE Trans. on Comp., vol. 44, No. 2, February 1995, pp. 223-233
- [9] E.J.McCluskey. Built-in Self-test Structures. IEEE Design & Test of Computers, Vol. 2, No. 2, Apr. 1985, pp. 29-36
- [10] Shiyuan Yang, Fault Diagnosis and Reliable Design of Digital Systems. Tsinghua University Press, 2000, pp. 6-8, 174, 60-65
- [11] H.T.Kung. Why Systolic Architecture? IEEE Computer, Vol 1.15, No.1, Jan 1982
- [12] T.Komarek and P.Pirsch. Array Architectures for Block Matching Algorithm. IEEE Trans.Circuits and Systems, Vol. 36, No. 10 October 1989
- [13] Pingying Zeng, Research on Built-In Self-Test Techniques for Arithmetic Circuits, PhD Thesis of Harbin Institute of Technology, July 1998, pp. 3-4, 69