Testable Design and BIST Techniques for Systolic Motion Estimators in Transform Domain

Shyue-Kung Lu and Wei-Yuan Liu

Abstract—Testable design techniques for systolic motion estimators based on M-testability conditions are proposed in this paper. The whole motion estimator can be viewed as a two-dimensional iterative logic array (ILA) of processing elements (PEs) and multiplying elements (MULs). The functions of each PE and MUL are modified to be bijective to meet the M-testable conditions. The number of test patterns is $2^w$, where $w$ denotes the word length of a PE. The proposed testable design techniques are also suitable for built-in self-test implementation. According to experimental results, our approaches can achieve 99.27% fault coverage. The area overhead is about 9%. To verify our approaches, an experimental chip is also implemented.

Index Terms—Built-in self-test, design for testability, fault coverage, motion estimator.

1. Introduction

Motion estimation is the major component that demands the most computation time in video coding standards, such as H.26X and MPEG X. These standards consist of discrete cosine transform (DCT), inverse DCT (IDCT), motion estimation (ME) and motion compensation, quantization, inverse quantization, run length coding (RLC), and variable-length coding (VLC) to remove spatial, temporal, and statistical redundancies presented in the video data. About 60% to 80% computation time of video coding is consumed in motion estimation\[1\]. Therefore, reducing computation time of motion estimation for video encoding is inevitable.

Several motion estimation algorithms and VLSI (very large scale integration) architectures have been proposed for practical video coding applications. These algorithms are almost based on the spatial domain. Therefore, some characteristics of video information in the transform domain can not be used to further reduce the computation complexity. In [2], [3] and [4], frequency domain motion estimation algorithms are proposed. Their corresponding systolic architectures are also proposed in [4].

Since a large number of processing elements are integrated into a single chip, it will result in the increase of the logic-per-pin ratio. The controllability and observability of the chip are drastically reduced. Consequently, testing such highly complex and dense circuits becomes very difficult and expensive.

In this paper we consider testable design techniques for transform domain motion estimators by modifying the functions of PE and MUL to meet the M-testable conditions, where the motion estimator is viewed as a two-dimensional iterative logic array\[5][15\] of processing elements (PEs) and multiplying elements (MULs).

The organization of this paper is described as follows. In Section 2, M-testability conditions are reviewed and some definitions are given. In Section 3, the architecture of the systolic motion estimator in the transform domain is described. The design-for-testability techniques for this architecture are described in Section 4. According to the proposed testable design, the corresponding built-in self-test designs are described in Section 5. Experimental results are shown in Section 6. Finally, some conclusions are given in Section 7.

2. Review of M-Testability Conditions

In order to ease our discussion, some definitions used in our previous works\[5][8\] are reviewed first.

Definition 1. A cell is a combinational machine $(\Sigma, \Delta, f)$, where $f: \Sigma \rightarrow \Delta$ is the cell function, $\Sigma = \{0, 1\}^j$ and $\Delta = \{0, 1\}$, for $I, O \in N$. A cell can be a bit-level cell such as the adder cell. Moreover, it can also represent a word-level cell such as a 2-point butterfly module. An ILA is an array of cells. We use the terms array and ILA interchangeably.

Definition 2. A complete or exhaustive input sequence $\sigma$ for a cell is an input sequence consisting of all possible input combinations for the cell, i.e., $\sigma = \sigma_1 \sigma_2 \cdots \sigma_k$, where $\sigma_i \in \Sigma$, $i \in \{1, 2, \cdots, k\}$.

Definition 3. A complete output sequence, $\delta = \delta_1 \delta_2 \cdots \delta_k$, is defined analogously. A minimal complete sequence is the shortest, which has a length of $2^w$, $w$ denotes the word length of a cell.
Definition 4. A C-testable array is an array testable with a constant number of test patterns independent of the size of the array. An M-testable array is also an array testable with a constant number of test patterns.

An ILA is homogeneous when it consists of functionally identical cells, otherwise it is heterogeneous. We assume that the behavior of cell is invariant over time, even if it is faulty. A faulty cell is a combinational machine \( (\Sigma, \Delta, f') \), where \( f': \Sigma \rightarrow \Delta \) and \( f' \neq f \). This model is called the cell fault model, which has been used as the cell-level fault model by most researchers working on ILA testing. We also assume that there is at most one faulty cell in the array, i.e., the single cell fault model (CFM) \([6][8]\) is adopted. The function of a faulty cell may deviate from the correct one in any manner, as long as it remains combinational. That is, we are testing permanent combinational faults only.

3. Systolic Motion Estimator in Transform Domain

The architecture of the systolic motion estimator \([4]\) is shown in Fig. 1. It consists of four modules—the controller, ROM (read only memory), output buffer, and the systolic array. According to the adopted motion estimation algorithm, the controller is used to determine the ROM access pattern. It can also apply power-on/off operations to the zig-zag ordered PEs. The purpose is to select the required values for calculating the sum of absolute differences (SAD). For this architecture, full search algorithm of motion estimation is used. The ROM stores the DCT-transformed shift matrices. There are four kinds of shift matrices—up, down, right, and left shifting and each with eight shifting distance \([4]\).

The detail structure of the systolic array is shown in Fig. 2. It mainly consists of PEs and MULs. Each PE contains a multiplier-accumulator module, one multiplexer, and three pipelined flip-flops as shown in Fig. 3. The results of multiplication operation are added to the previous results through the feedback loop. The Mux sel signal is used to switch between multiplier-accumulator operation and data transfer operation. The final results are forwarded to the output buffer through the PEs.

Fig. 1. Schematic diagram of the systolic motion estimator.

![Systolic array](image)

The functions of each PE can be expressed as follows:
\[
A_{out} = A_{in};
B_{out} = B_{in};
C_{out} = \text{Mux}_{sel}(C_{in}) + \text{Mux}_{sel}(A_{in} \times B_{in}).
\]

When Mux_{sel}=0, the values of C_{out} will not be shifted out. The PE simply acts as an accumulator. Therefore, the function of each PE is bijective when Mux_{sel}=0 since the inputs are directly forwarded to the outputs. This property is helpful that the following theorem can be applied directly to make the PE array M-testable.

The MUL module consists of one multiplier and two output flip-flops as shown in Fig. 4. The input signal and the multiplication result are both delayed one clock cycle at the output flip-flops. The original function of a MUL module is described as follows:
\[
A_{out} = A_{in};
B_{out} = A_{in} \times B_{in}.
\]

4. Design-for-Testability Techniques

In this section, we propose a DFT scheme based on M-testability conditions. We first review a theorem \([5]\) that will be applied to the systolic motion estimator.
Fig. 4. Schematic diagram of the multiplier element (MUL).

Fig. 5. Multiply/PE Module (MPE).

*Theorem:* If a two-dimensional (rectangular) homogeneous ILA has bijective cell function, then it is $2^n$-testable\(^5\).

According to the theorem described above, the function of the MUL module is not bijective. Moreover, the systolic array shown in Fig. 2 is not a homogeneous array (the array contains two different modules — PEs and MULs). Therefore, we should modify the function of the MUL module into a multiply/PE (MPE) module as shown in Fig. 5. In this figure, three multiplexers and one adder are added. The multiplexers are used to switch between normal mode and test mode. In test mode (Test_mode = 1), the MPE module acts as a processing element. Therefore, the systolic array shown in Fig. 2 can be viewed as a 2-D array of PEs in test mode. In test mode (Test_mode = 1), the MPE module acts as a processing element. Therefore, the systolic array shown in Fig. 2 can be viewed as a 2-D array of PEs in test mode. When Test_mode = 0, the MPE acts simply as a multiplier as used in normal mode. Although an adder is added into the MPE module, however, this adder is helpful for checksum operation in test mode. Since the function of each PE (including MPEs in test mode) is bijective, the systolic array shown in Fig. 2 is M-testable according to the theorem described above.

5. Built-In Self-Test

The BIST architecture of our technique is shown in Fig. 6. In Fig. 6, instead of the original controller coordinating the computation of motion vectors, a BIST controller and an output response analyzer (ORA) are added. The BIST_EN signal is used to activate the BIST session. When it is asserted, the BIST session starts. The Test_Mode signal is used to switch the MPE modules in the systolic array between normal mode and test mode.

The BIST controller includes a test pattern generator (TPG) which generates a minimal complete input sequence of a PE. Since the input word length of a PE is 24, therefore, a 24-bit binary counter can be used as the TPG. The generated test patterns are sent to the left-upmost PE through the 24-bit Test_pattern signal bus. The controllability and observability of each PE can be derived directly from the above theorem. Therefore, $2^{24}$ test patterns are required to test the whole array. Although 16 M patterns are required, the test time is only 0.16 s if 100 MHz operating frequency is used. According to the theorem, the fault coverage of each PE (MPE) can achieve 100%.

After applying of these patterns, the accumulated sums stored in the accumulators of all PEs are the same (the outputs of each PE are a minimal complete output sequence). We can then switch the PEs into shift mode to shift out the accumulated sums to the output buffer and then forward to the ORA for further analysis. The ORA can be simply a comparator used to compare all the shifted out accumulated sums. If a discrepancy occurs, it means that the array is faulty. The checksum technique may still incur aliasing. However, according to previous researches, the aliasing probability is almost negligible. The BIST controller is basically a simple finite state machine (FSM) which supervises the BIST operations. The hardware overhead of this BIST scheme is very low.

A more detailed diagram of the proposed BIST scheme is shown in Fig. 7. The TPG within the BIST controller generates a minimal complete input sequence for the left-upmost MPE. In order to apply $+45^\circ$ test tessellation, pipeline latches are added along the boundaries of the top row of MPEs and the leftmost column of MPE and PEs. In Fig. 7, $m$ and $n$ denote time frames. As indicated by the $+45^\circ$ dashed lines, minimal complete input sequence will be...
propagated to all MPEs and PEs. If all processing elements are fault free, the accumulated sums stores in all MPEs and PEs are the same. This property is helpful for the ORA to compare the accumulated sums when they are shifted out.

The systolic motion estimator is controlled by the BIST controller in test mode. The block diagram of the BIST controller is shown in Fig. 8 which consists of a 25-bit binary counter, a finite state machine, and a 3-bit counter. The finite state machine is the main part of the controller which contains seven states as shown in Fig. 9. The operations of this FSM are described as follows.

1) Normal mode. The circuit calculates matrix shifting for DCT coefficients. The finite state machine stays in normal mode waiting for test claims issued by the host processor.

2) Start State (Clr=0, Test_mode=1) When the Test_signal goes to high, the Clr signal resets all signal values in the systolic array to prepare self testing. The entire system then switches to test mode by asserting the Test_mode signal to 1. The state machine then enters to the start state.

3) TPG State (Cont_enable=1, TPG_enable=1, Clr=0). In this state, the controller starts to activate the test pattern generator which is composed of a 25-bit counter and a multiplexer. The inputs of the multiplexer are the outputs from the binary counter and the ground signal as shown in Fig. 8. The TPG generates all the 24-bit binary patterns until Count_out=10000000. The Count_out signal is basically the combination of the 25th bit and the first to the 7th bits of the counter outputs. Thereafter, the FSM goes to the next state.

4) Wait State (TPG_enable=0). In this state, the FSM waits 15 clock cycles for sending the test patterns to the motion estimator array.

5) Scan_Out State (Test_sel[3]=1, Scan_sel=0). In this state, the highest bit of Test_sel signal activates the scan operation. The remained bits are used to select a specific column to scan out. The scanned out data are received by the ORA. The function of the ORA is to verify if the received data is faulty or not and locate the faulty processing elements in the systolic array. The ORA contains a register, a 4-bit address counter, and a comparator.

6) Scan_out_1 State (Scan_sel=1). Our FSM checks Count_out with the values in Group-A (shown in Fig. 9). The data in Group-A increases 9 from 10010110 to 11010111. It controls the 3-bit counter to add one to the Group-A=(10010110,10100000,10110010,10111011,11000100,11001101,11010111)
Test_sel[2:0] signal for each nine clock cycles period. The controller receives the Test_sel signal for scanning out. The ORA locates fault address by Test_sel. The FSM will return to the Scan_out_1 without any exception.

7) Finish State (Test_mode=0, Test_sel=0, Count_enable=0). When all the test operations are completed, the FSM enters the Finish State by deactivating the Test_mode signal. Thereafter, the controller shown in Fig. 6 controls the normal operations of the systolic motion estimator.

6. Experimental Results

Our systolic motion estimator with DFT and BIST techniques are designed and synthesized with TSMC (Taiwan Semiconductor Manufacturing Company) 0.18 μm CMOS cell library. Table 1 and Table 2 show the characteristics of our chip. The chip layout is shown in Fig. 10. The multiplier and rotate-carry adder used in each PE determine the critical path. According to simulated results, it can operate at 125 MHz. The area overhead of our design is about 9%. Fault coverage of individual PE and MPE are shown in Table 3. It will not achieve 100% due to the added multiplexers in PEs and MPEs. The alternate inputs of MUXs are not tested in test mode. Therefore, the fault coverage of the estimator array is 99.28%. If we want to increase the fault coverage, we can return to normal mode to test the alternate inputs of multiplexers.

![Fig. 10. Chip layout.](image)

Table 1: Core characteristics

<table>
<thead>
<tr>
<th>Items</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inputs</td>
<td>19 bits</td>
</tr>
<tr>
<td>Outputs</td>
<td>20 bits</td>
</tr>
<tr>
<td>Technology</td>
<td>TSMC 0.18 μm</td>
</tr>
<tr>
<td>Gate count</td>
<td>233904</td>
</tr>
<tr>
<td>Core area</td>
<td>4 mm²</td>
</tr>
<tr>
<td>Clock rate</td>
<td>125 MHz</td>
</tr>
<tr>
<td>Mode selection</td>
<td>Test/Normal mode</td>
</tr>
<tr>
<td>Block size</td>
<td>8×8</td>
</tr>
</tbody>
</table>

Table 2: Processor I/O specification

<table>
<thead>
<tr>
<th>Function</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCT_coefficient</td>
<td>12</td>
</tr>
<tr>
<td>Data_out</td>
<td>12</td>
</tr>
<tr>
<td>RLC</td>
<td>6</td>
</tr>
<tr>
<td>Fault_out</td>
<td>1</td>
</tr>
<tr>
<td>Fault_address</td>
<td>7</td>
</tr>
<tr>
<td>Clock</td>
<td>1</td>
</tr>
<tr>
<td>Clear</td>
<td>1</td>
</tr>
<tr>
<td>Ground</td>
<td>5</td>
</tr>
<tr>
<td>VDD</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 3: Fault coverage

<table>
<thead>
<tr>
<th>Structure</th>
<th>Fault coverage (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPE</td>
<td>97.55</td>
</tr>
<tr>
<td>PE</td>
<td>99.49</td>
</tr>
<tr>
<td>Array</td>
<td>99.28</td>
</tr>
</tbody>
</table>

7. Conclusions

We propose efficient design-for-testability techniques and their corresponding BIST scheme for motion estimators in the transform domain. The main concept is based on M-testability conditions. In order to meet these conditions, the function of processing elements of the computing array is modified to be bijective. The result number of test patterns is $2^{24}$ and can be generated by a simple binary counter. The output response analyzer is basically a comparator. In order to verify our techniques, an experimental chip is designed and simulated. According to experimental results, it can operate at 125 MHz and the hardware overhead is about 9% for implementing our DFT and BIST techniques.

References


Shyue-Kung Lu was born in Taoyuan, Taiwan, China, in 1963. He received the M.S. degree in 1991 from National Tsinghua University, Hsinchu, Taiwan, and Ph.D. degree in 1996 from National Taiwan University, Taipei, Taiwan, China both in electrical engineering. From 1996 to 1998, he was an associate professor with Electrical Engineering, Lunghua Junior College of Technology and Commerce, Taoyuan, Taiwan, China. From 1998 to 2009, he was with Department of Electronic Engineering, Fu-Jen Catholic University. From 2006 to 2009, he also served as the department chairman. Since 2009, he has been with Department of Electrical Engineering, National Taiwan University of Science and Technology, where he is a professor. His research interests are in the areas of VLSI testing and fault-tolerant computing, video coding techniques and architectures design.

Wei-Yuan Liu was born in Taipei, Taiwan, China, in 1982. He received the B.S. degree in 2005 and the M.S. degree in 2007 from the Department of Electronic Engineering, Fu Jen Catholic University, Taipei, Taiwan, China. His research interests include VLSI design and testing, as well as the testing and self-repair of digital signal processors. He is currently serving his military service.