ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

An Efficient Architecture for Motion Estimation in Video Coding using EDCA

Pattan Shareef
Dept of VLSI & ESD,
SKR College of Engineering & Technology,
Konduru Satram [V], Manubolu [M], S.P.S.R.
Nellore [Dt], Andhra Pradesh, India.

Mr. Goni Mahendra
Associate Professor
Dept of VLSI Systen Design,
SKR College of Engineering & Technology,
Konduru Satram [V], Manubolu [M], S.P.S.R.
Nellore [Dt], Andhra Pradesh, India.

Abstract:

Given the critical role of motion estimation (ME) in a video coder, testing such a module is of priority concern. While focusing on the testing of ME in a video coding system, this work presents an error detection and correction (EDCA) design, based on the residue-and-quotient (RQ) code, to embed into ME for video coding testing applications. An error in processing elements (PEs), i.e. key components of a ME, can be detected and recovered effectively by using the proposed EDCA design. Experimental results indicate that the proposed EDCA design for ME testing can detect errors and recover data with an acceptable area overhead and timing penalty. Importantly, the proposed EDCA design performs satisfactorily in terms of throughput and reliability for ME testing applications.

Index Terms:

Area overhead, data recovery, error detection, motion estimation, reliability, residue-and-quotient (RQ) code.

1. INTRODUCTION:

Advances in semiconductors, digital signal processing, and communication technologies have made Multimedia applications are becoming more flexible and reliable. Some of the Video Compression standards include MPEG-1, MPEG-2 and MPEG-4. The advanced Video Coding standard is MPEG-4. Video compression is essential in various applications to reduce the total amount of data required for transmitting or storing the video data.

Motion Estimate is of priority concern in removing the temporal redundancy between the successive frames in a video coding system and also it consumes more time. Motion Estimate is considered as the intensive unit in terms of computation. Regular arrangement of PEs with size 4x4 constitutes a Motion Estimate. Advancements in VLSI technologies facilitate the integration of large number of PEs into a single chip. Large number of PEs arranged as an array helps in accelerating the computation speed. Testing of PEs (processing elements) is essential as an error in PE affects the video quality and signal-to-noise ratio. Numerous PEs in a ME can be tested concurrently using Concurrent Error Detection (CED) methods. In this method, different operations are performed on the same operand. An error is detected by the conflicting results produced by the operations performed.Concurrent fault simulation is essentially an event-driven simulation with the fault-free circuit and faulty circuits simulated altogether. Design for Testability (DFT) techniques are required in order to improve the quality and reduce the test cost of the digital circuit, while at the same time simplifying the test, debug and diagnose tasks. Logic built-in self-test (BIST) is a design for testability technique in which a portion of a circuit on a chip, board, or system is used to test the digital logic circuit itself. BIST technique for testing logic circuits can be online or offline. Online BIST is performed when the functional circuitry is in normal operational mode. In concurrent online BIST, testing is conducted simultaneously during normal functional operation. The functional circuitry is usually implemented with coding techniques. Any input pattern or sequence of input patterns that produces a different output response in a faulty circuit from that of the fault-free circuit is a test vector, or sequence of test vectors, that will detect the faults.The goal of test generation is to find an efficient set of test vectors that detects all faults considered for that circuit. Because a given set of test vectors is usually capable of detecting many faults in a circuit, fault simulation is typically used to evaluate the fault coverage obtained by that set of test vectors. Because of the diversity of VLSI defects, it is difficult to generate tests for real defects. Fault models are necessary for generating and evaluating a set of test vectors. Generally, a good fault model should satisfy two criteria: (1) It should accurately reflect the behaviour of defects, and (2) it should be computationally efficient in terms of fault simulation and test pattern generation.

1.1Manufacturing Test of Integrated Circuits:

Fabrication anomalies in the IC manufacturing process may cause some circuits to behave erroneously. Manufacturing test helps to detect physical defects (e.g., shorts or opens) prior to delivering the packaged circuits to end-users. Once a defective chip has been detected, comprehensive defect screening through fault diagnosis is required to adjust the manufacturing process and accelerate the yield learning curve. The physical defects lead to faulty behaviours that can be detected by parametric tests for chip pins and tests for functional blocks. Parametric tests include DC tests (such as voltage, leakage test and output drive current test) and AC tests (setup and hold time tests and propagation test). These tests are usually technology-dependent and can be done without any understanding of the chip functionality. The test for functional blocks involves modelling manufacturing defects at a certain level of design abstraction, such as behavioural level, register-transfer level (RTL), gate level or transistor level. Fault models based on gate level net lists are technology-independent and over time have been proven to be very efficient for testing digital circuits. Basic fault models for gate level testing are stuck-at, bridging and delay fault models. The single stuck-at fault model is the most popular fault model in digital system testing and is based on the assumption that a single node (line) in the structural net list of logic gates can be stuck to a logic value 0 (SA0) or 1 (SA1). The test for functional blocks determines whether the manufactured chip behaves as designed and because the gate count keeps on growing, the testing time for functional blocks is also increasing. Since the time a chip spends on an expensive tester directly influences the production cost, reducing the testing time of functional blocks is an essential task, which needs to be accomplished in order to lower the cost associated with manufacturing test. The test of functional blocks can further be divided into structural test and functional test. If the test depends on the net list structure of the design then it is called structural test. Based on the targeted fault models (e.g., stuck-at), automatic test pattern generation (ATPG) tools generate test sets which sensitize the fault and propagate its effects to observation points (e.g., primary outputs). Functional test programs, on the other hand, generate a set of test patterns to verify the functionality of each component in the circuit. Because functional test is an exhaustive test method, testing time is prohibitively large for combinational logic blocks, which makes it infeasible for complex digital systems. One exception is the test of semiconductor memories due to their regularity. Since the cells in a memory block have identical structure and they are not related one to each other, and because memory operations are simple (read and write), exhaustive functional test becomes tractable. Chapter 2 gives detailed information on functional memory fault models and test algorithms.

1.2Digital Test Methodologies: ATE vs. BIST:

The basic principle of manufacturing testing is illustrated in Figure. Circuit under test (CUT) can be the entire chip or only a part of the chip (e.g., a memory core or a logic block). Input test vectors are binary patterns applied to the inputs of the CUT and the associated output responses are the values observed on the outputs of the CUT. Using a comparator output responses are checked against the expected correct response data, which is obtained through simulation prior to design tape-out. If all the output responses match the correct response data, the CUT has passed the test and it is labelled as fault-free. Based on the techniques how the test vectors are applied to the CUT and how the output responses are

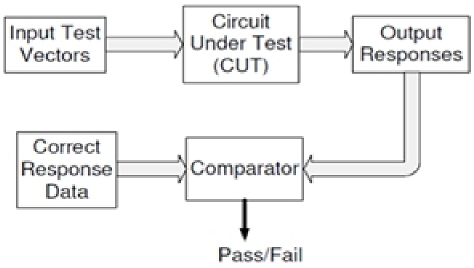


Fig 1.1: Basic Principle of Testing

compared, there are two main directions to test electronic circuits: external testing using automatic test equipment (ATE) and internal testing using built-in self-test (BIST). When external testing is employed, the input test vectors and correct response data are stored in the ATE memory. Input test vectors are generated using ATPG tools, while correct response data is obtained through circuit simulation. For external testing, the comparison is carried out on the tester. Although the ATE-based test methodology has been dominant in the past, as transistor to pin ratio and circuit operating frequencies continue to increase, there is a growing gap between the ATE capabilities and circuit test requirements (especially in terms of speed and volume of test data). ATE limitations make BIST technology an attractive alternative to external test for complex chips. BIST is a design-for-test (DFT) method where part of the circuit is used to test the circuit itself (i.e., test vectors are generated and test responses are analyzed on-chip). BIST needs only an inexpensive tester to initialize BIST circuitry and inspect the final results (pass/fail and status bits).However, BIST introduces extra logic, which may induce excessive power in the test mode (see next section for details), in addition to potential performance penalty and area overhead. BIST circuitry can further be divided into logic BIST for random logic blocks (e.g., control circuitry or data path components) and memory BIST for on-chip memory cores. The cost and quality of logic BIST has been subject to extensive research over the last two decades. It is important to note that the main problem with logic BIST lies in the computational overhead required to synthesize compact and scalable test pattern generators and response analyzers such that high fault coverage is achieved in low testing time and with limited interaction to external equipment. In contrast, due to the regular memory block structure and simple operations of memory cores, memory BIST (MBIST) can be implemented using compact and scalable test pattern generators and response analyzers and it can rapidly achieve high fault coverage for certain functional fault models.

1.3Systems-on-a-Chip Test Challenges:

As process technologies continue to shrink, designers are able to integrate all or most of the functional components found in a traditional system-on-a-board (SOB) onto a single silicon die, called system-on-a-chip (SOC). This is achieved by incorporating pre-designed components, known as intellectual property (IP) cores (e.g., processors, memories), into a single chip.

While SOCs benefit designers in many aspects, their heterogeneous nature presents unique technical challenges to achieve high quality test, i.e., acceptable fault coverage's for the targeted fault models. In the following, several SOC test challenges are enumerated along with the motivation for a shift from ATE-based SOC testing to BIST.

• Controllability and observability:

An SOC contains several embedded IP cores. Although the IP cores are predesigned and pre-verified by the core providers, SOC composition is the system integrators' duty, which is also in charge of verification and manufacturing testing of the entire SOC, including the IP-protected internal cores. Since most of the input/output (I/O) ports of these embedded cores are not directly connected to the SOC's pins, the testability, i.e., both the controllability and the observability, is reduced and, unless some special DFT techniques are employed, the fault coverage will be lowered. To increase the testability, test access mechanisms (TAMs) and core wrappers are two new and important DFT techniques in SOC testing. TAM delivers test vectors (propagates test responses) to (from) embedded cores from (to) primary inputs (outputs), while core wrappers connect the embedded cores to the TAM. The wrapper/TAM co-design can be solved for different optimization objectives (e.g., testing time or TAM width) and constraints (e.g., layout or power dissipation) . However, when ATE-based testing is employed (i.e., patterns and responses are stored on the tester), since the number of tester channels is limited in practice, test concurrency is bounded by the number of these channels, which can adversely influence the cost of test. This problem can be addressed by moving the generation and analysis functions on-chip and use an inexpensive tester to initialize, control and observe the final results of the testing process.

• Volume of test data, tester channel capacity and testing time:

The volume of test data is determined by the chip complexity and it grows rapidly as more IP cores are integrated into a single SOC. The easiest way to deal with increased volume of test data is to upgrade the tester memory and use more tester channels to increase test concurrency, however this is infeasible since it will prohibitively increase the ATE cost. A more cost effective approach is to use test data compaction and/or compression.

Test data compaction reduces the number of test patterns in the test set (by discarding test patterns that target faults detected by other patterns in the test set) and test data compression decreases the number of bits (that need to be stored for each pattern) and uses dedicated decompression hardware (either off or on-chip) for real-time decompression and application. Test data compaction reduces the volume of test data; however it is trading-off the tester channel capacity against the testing time. If the decompression hardware is placed on-chip, then test data compression eliminates this trade-off. Deterministic BIST is a particular case of test data compression where the compressed bits are used for BIST initialization (i.e., seeds) and BIST observation (i.e., signatures). The benefits of memory BIST technology are justified mainly by its deterministic nature.

• Heterogeneous IP cores:

Many SOC designs incorporate cores that use different technologies, such as random logic, memory blocks, and analog circuits. For systems assembled on printed circuit boards (PCBs) each of these components was tested using different types of dedicated ATEs (e.g., digital, memory or analog testers). For SOC testing one can use generic high-performance mixed-signal ATEs, however their high production cost brings limited benefits to complex designs, since cores using heterogeneous technologies still need to be tested sequentially, thus lengthening the testing time and ultimately raising the manufacturing test cost. In addition, embedded core controllability and observability issues cannot be addressed without dedicated on-chip DFT hardware, whose necessity justifies a shift toward BIST. The use of different BIST circuitry for the appropriate technologies (logic, memory or analog BIST), increases both testability and test concurrency of SOCs comprising heterogeneous IP cores.

• At-speed test:

As VLSI technology moves below 100 nm, traditional stuck-at fault testing is not sufficient. This is because unanticipated process variations, weak bridging defects, and crosstalk violations (only to mention a few) may cause only timing malfunctions, which cannot be detected by the stuck-at fault test vectors delivered by ATEs whose frequency is lower than the maximum CUT frequency. These logical faults caused by timing-related defects are known as delay faults and they can only be detected when the chip is tested at the functional (rated) speed. This type of test is called at-speed test. For microprocessor-based circuits, at-speed test can be accomplished by running a set of functional test programs (stored in an off-chip or on-chip memory). Since design automation for functional test program development is still an emerging research area, this approach is very time consuming (even for decent delay fault coverage). An alternative for logic blocks is to use structural scan patterns and specialized scan chain clocking schemes coupled with two-pattern test application strategies through scan. In any of the above cases at-speed test can be performed using high-speed ATEs (note, however, even the highest performance/cost ATEs will be slower than the fastest new chips), or more cost effectively, by BIST interacting with a low-speed testers required only to activate the self-test circuitry and to acquire the BIST signatures.

• Power dissipation:

Power dissipation is becoming a key challenge for the deep sub-micron CMOS digital integrated circuits. Placing more and more functions on a silicon die has resulted in higher power/heat densities, which imposes stringent constraints on packaging and thermal management in order to preserve performance and reliability. There are two major sources of power consumption in CMOS VLSI circuits: dynamic power dissipation, due to capacitive switching, and static power dissipation, due to leakage and sub threshold currents. The 2001 International Technology Roadmap for Semiconductors (ITRS) anticipates that power will be limited more by system level cooling and test constraints than packaging. This is because, if packaging and thermal management parameters (e.g., heat sinks) are determined only based on the functional operating conditions, the higher test switching activity and test concurrency will affect both manufacturing yield and reliability.

On the one hand, dynamic power dissipation dominates the chip power consumption for digital CMOS technology in 180 nm range or higher. Dynamic power dissipation can be analyzed from two different perspectives. Average power dissipation which stands for the average power utilized over a long period of operation, and peak power dissipation which is the power required in a very short time period such as the power consumed immediately after the rising or falling edge of the system clock. When considering SOC test, to achieve high fault coverage with less test data, the test patterns are usually uncorrelated.

This means the switching activity during test can differ from that during functional operation. In most cases, the testing power consumption is the higher one. A practical measurement is reported in which indicate the switching activity is 35-40% more during scan-based transition test than that in normal functional mode. For traditional stuck-at fault test, one straightforward solution to meet the power constraints is to reduce the system clock frequency during test which implies longer testing time. However, as described in the previous challenge, to test time related faults, at-speed testing is necessary. Consequently, the power dissipation during at-speed test can exceed the maximum power limit which may lead to chip malfunctions or to burn the overheated chip. There are two research directions to address dynamic power problem during at-speed test: the first direction aims to limit the number of concurrent test blocks using test scheduling under power constraints. The second research direction is to reduce the switching activity during test. On the other hand, static power dissipation is becoming an important component for low power design and test in 130nm or lower CMOS technologies with low gate sub threshold.

Power gating is an efficient method to reduce static power dissipation and it based on disconnecting the idle module(s) from the power and ground network to reduce the leakage currents. This technique is particularly useful for SOCs with a high number of embedded memories. Note, due to the experimental setup (based on digital 180 nm process technology), the power dissipation problem addressed in this thesis is focused only on dynamic power dissipation during test. However, by using the power-gating method, it is anticipated that the proposed methods can be adapted to solve the static power dissipation problem by turning off the idle memories during test to increase test concurrency. All the above mentioned SOC test challenges need to be overcome in order to reduce the ever-growing cost of manufacturing test while enabling high manufacturing yield and reliability through satisfactory test quality. Although the cost of test is dominated by many factors, such as the cost of production ATEs, testing time, performance of test automation tools (e.g., ATPG), area and performance overhead caused by additional DFT or BIST circuitry, it is essential to balance this cost against the benefits of enabling high product reliability and a fast yield learning curve. As the SOC complexity increases and more physical defects manifest themselves only in the timing domain, at-speed BIST is emerging as an essential and necessary technology, which can enable short time-to-volume and low cost of manufacturing test. This is also correlated to the fact that, as total chip area continues to increase, the overhead associated with consciously-designed BIST architectures is decreasing.

3. BIST (Built-in Self Test):

The basic idea of BIST, in its most simple form, is to design a circuit so that the circuit can test itself and determine whether it fault-free or faulty. This typically requires that additional circuitry and functionality be incorporated into the design of the circuit to facilitate the self-testing feature. This additional functionality must be capable of generating test patterns as well as providing a mechanism to determine if the output responses of the circuit under test (CUT) to the test patterns correspond to that of a fault-free circuit.

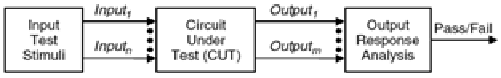


Fig 3.1: Basic approach of Testing

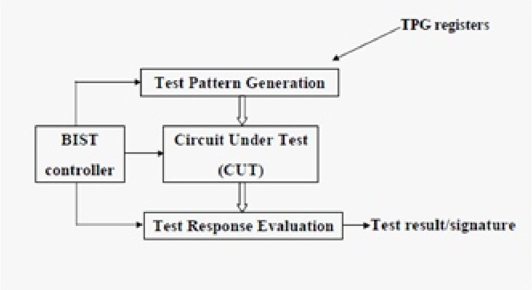


Fig 3.2: General BIST Architecture

Built-in Self Test, or BIST, is the technique of designing additional hardware and software features into integrated circuits to allow them to perform self-testing, i.e., testing of their own operation (functionally, parametrically, or both) using their own circuits, thereby reducing dependence on an external automated test equipment (ATE). BIST is a Design-for-Testability (DFT) technique, because it makes the electrical testing of a chip easier, faster, more efficient, and less costly. The concept of BIST is applicable to just about any kind of circuit, so its implementation can vary as widely as the product diversity that it caters to.

Why BIST?

The main drivers for the widespread development of BIST techniques are the fast-rising costs of ATE testing and the growing complexity of integrated circuits. It is now common to see complex devices that have functionally diverse blocks built on different technologies inside them.

input patterns that are applied to the device's internal scan chain, and a multiple input signature register (MISR) for obtaining the response of the device to these test input patterns. An incorrect MISR output indicates a defect in the device.

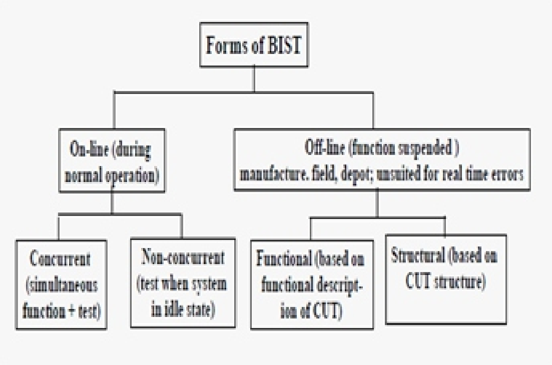

Forms of BIST
On-line (during normal operation)
Off-line (function suspended) manufacture. field, depot; unsuited for real time errors
Concurrent (simultaneous function + test)
Non-concurrent (test when system in idle state)
Functional (based on functional descript-ion of CUT)
Structural (based on CUT structure)

Fig 3.3: BIST Classification

Online BIST is performed when the functional circuitry is in normal operational mode. It can be done either concurrently or non-concurrently.In concurrent online BIST, testing is conducted simultaneously during normal functional operation. When an intermittent or transient error is detected, the system will correct the error on the spot, rollback to its previously stored system states, and repeat the operation, or generate an interrupt signal for repeated failures.In non-concurrent online BIST, testing is performed when the functional circuitry is in idle mode. This is often accomplished by executing diagnosis software routines (macrocode) or diagnosis firmware routines (microcode). The test process can be interrupted at any time so that normal operation can resume.Offline BIST is performed when the functional circuitry is not in normal mode. This technique does not detect any real-time errors but is widely used in the industry for testing the functional circuitry at the system, board, or chip level to ensure product quality.Functional offline BIST performs a test based on the functional specification of the functional circuitry and often employs a functional or high-level fault model. Normally such a test is implemented as diagnostic software or firmware.Structural offline BIST performs a test based on the structure of the functional circuitry.

Engineers design BISTs to meet requirements such as:

-High reliability
-Lower repair cycle times
- Cost of testing during manufacture
- Constraints such as limited technician accessibility

3.2Applications:

» BIST offers reliability of electronic circuits aboard satellites and space shuttles. Once launched in space these systems maintain their functional integrity. The detection and diagnostics of any malfunctions is done from the earth stations.
» BIST is commonplace in integrated circuits in avionics, medical devices, automotive electronics, complex machinery of all types, and unattended machinery of all types.
» Almost all avionics now incorporate BIST. In avionics, the purpose is to isolate failing line-replaceable units, which are then removed and repaired elsewhere, usually in depots or at the manufacturer.
» Safety-critical devices.
» Medical devices test themselves to assure their continued safety. Normally there are two tests. A power-on self-test (POST) will perform a comprehensive test. Then, a periodic test will assure that the device has not become unsafe since the power-on self test.
» Automotive tests itself to enhance safety and reliability. For example, most vehicles with antilock brakes test them once per safety interval. If the antilock brake system has a broken wire or other fault, the brake system reverts to operating as a normal brake system.
» Unattended machinery performs self-tests to discover whether it needs maintenance or repair. Typical tests are for temperature, humidity, bad communications, or a bad power supply.

3.3 Advantages of BIST:

•Vertical Testability: The same testing approach could be used to cover wafer and device level testing, manufacturing testing as well as system level testing in the field where the system operates.

•Reduction in Testing Costs: The inclusion of BIST in a system design minimizes the amount of external hardware required for carrying out testing significantly. A 400 pin system on chip design not implementing BIST would require a huge (and costly) 400 pin tester, when compared with a 4 pin (vdd, gnd.,clock and reset) tester required for its counterpart having BIST implemented.

•In-Field Testing capability: Once the design is functional and operating in the field, it is possible to remotely test the design for functional integrity using BIST, without requiring direct test access.

•Robust/Repeatable Test Procedures: The use of automatic test equipment (ATE) generally involves the use of very expensive handlers, which move the CUTs onto a testing framework. Due to its mechanical nature, this process is prone to failure and cannot guarantee consistent contact between the CUT and the test probes from one loading to the next. In BIST, this problem is minimized due to the significantly reduced number of contacts necessary.

•At-speed testing: All testing by the BIST circuitry is performed at the system clock frequency. This is typically referred to as at-speed testing and is important since it facilitates the detection of faults that lead to excessive delay in working CUT.

•Fast, efficient and hierarchical - same hardware is capable of testing chips, boards and systems.

•No need of expensive ATE (cost >= $10 million).

•Testing during operation and maintenance.

•Uniform technique for production, system and maintenance tests.

•Dynamic properties of the circuit can be tested at speed.

•Support concurrent testing.

•Can be used for delay testing as it can be used in real time.

•Lower cost of test, since the need for external electrical testing using an ATE will be reduced, if not eliminated.

•Better fault coverage, since special test structures can be incorporated onto the chips.

•Shorter test times if the BIST can be designed to test more structures in parallel.

• Easier customer support.

•Capability to perform tests outside the production electrical testing environment.

•Built-in self-test (BIST) has been proven to be one of the most cost-effective and the last advantage mentioned can actually allow the consumers themselves to test the chips prior to mounting or even after these are in the application boards.

3.4 Disadvantages of

•Area Overhead: The inclusion of BIST in a particular system design results in greater consumption of die area when compared to the original system design. This may seriously impact the cost of the chip as the yield per wafer reduces with the inclusion of BIST.

•Performance penalties: The inclusion of BIST circuitry adds to the combinational delay between registers in the design. Hence, with the inclusion of BIST the maximum clock frequency at which the original design could operate will reduce, resulting in reduced performance.

•Additional Design time and Effort: During the design cycle of the product, resources in the form of additional time and man power will be devoted for the implementation of BIST in the designed system.

3.5 BIST Key Elements:

-Circuit under test (CUT)
-Test pattern generators (TPG)
-Output response analyzer (ORA)
-BIST controller

The test pattern generator (TPG) automatically generates test patterns for application to the inputs of the circuit under test (CUT). The output response analyzer (ORA) automatically compacts the output responses of the CUT into a signature. Specific BIST timing control signals, including scan enable signals and clocks, are generated by the BIST controller for coordinating the BIST operation among the TPG, CUT, and ORA. The BIST controller provides a pass/fail indication once the BIST operation is complete.

BIST Controller:

BIST Controller is responsible for the control of test pattern generator (TPG) and output response analyzer (ORA) and provides a pass/fail indication once the BIST operation is complete.

Test Pattern Generation for BIST:

• Exhaustive test patterns : They produce every possible combination of input test patterns. This is easy to see in the case of an N-input combinational logic circuit where an N-bit counter produces all possible test patterns and will detect all detectable gate-level stuck-at faults as well as all detectable wired-AND/OR and dominant bridging faults in the combinational logic circuit without the need for fault simulation

• Pseudo-random test patterns: They are the most commonly produced patterns by TPG hardware found in BIST applications. The primary hardware for producing these test patterns are LFSRs and CA. Pseudo-random test patterns have properties similar to those of random pattern sequences but the sequences are repeatable

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

• Pseudo-exhaustive test patterns:

They are an alternative to exhaustive test patterns. In this case, each K-input sub circuit receives all possible patterns, where K < N, while the outputs of the sub circuit are observed during the testing sequence. Hardware for this type of test pattern generation for BIST includes counters, Linear Feedback Shift Registers (LFSRs), and Cellular Automata (CA).

Output Response Analyser:

The ORA compacts the output responses of the CUT to the many test patterns produced by the TPG into a single Pass/Fail indication (usually a multiple-bit "signature"). The ORA is sometimes referred to as an output data compaction (ODC) circuit. The significance of the ORA is that there is no need to compare every output response from the CUT with the expected output response external to the device. Only the final Pass/Fail indication needs to be checked at the end of the BIST sequence in order to determine the fault-free/faulty status of the CUT.Concentration typically performs output data compaction of multiple outputs from a CUT into a single output data stream. It is often used with other ORA techniques such as counting techniques to reduce the area overhead of the overall ORA implementation.Comparison-based ORAs use a comparator to detect mismatches in the fault-free and faulty circuits. An example of a comparator-based approach is the simple BIST where the expected responses were stored in a ROM and compared on a vector-by-vector basis to the output responses of the CUT.Counting techniques count the number of 0s, 1s, or transitions in the output response of the CUT with the resultant count value of the specific attribute at the end of the test sequence providing the "signature" for the Pass/Fail indication.Signature analysis is the most commonly used technique for ORAs in BIST implementations. Signature analysis uses an LFSR as the primary component of the ORA implementation. The basic idea behind signature analysis is to divide the polynomial representing the output response of the CUT by the characteristic polynomial of the LFSR used to implement the ORA. The resultant "signature" is the remainder of the polynomial division and is compared to the signature for the fault-free circuit at the end of the BIST sequence. Accumulators sum the output responses of the CUT by treating each output response as a binary magnitude. These are the same circuits that are used for checksum-based concurrent fault detection circuits. At the end of the BIST sequence, the accumulated value (or checksum) is compared to that obtained for the fault-free circuit to determine the pass/fail status of the CUT.

4. EDCA ARCHITECTURE:

The proposed EDCA scheme consists of two major blocks, i.e. error detection circuit (EDC) and data recovery circuit (DRC), to detect the errors and recover the corresponding data in a specific CUT. The test code generator (TCG) in the architecture utilizes the concepts of RQ code to generate the corresponding test codes for error detection and data recovery.

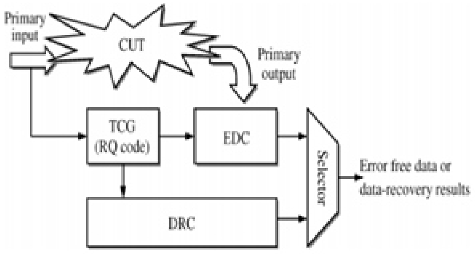


Fig.1.Conceptual View of the proposed EDCA design

The output from the circuit under test is compared with the test code values in the EDC. The output of EDC indicates the occurrence of error. DRC is in charge of recovering data from TCG. Additionally, a selector is enabled to select the error free data or data-recovery results. A ME consists of many PE's incorporated in a 1-D or 2-D array for video encoding applications. A PE generally consists of two ADDs (i.e. an 8-b ADD and a 12-b ADD) and an accumulator (ACC). Next, the 8-b ADD (a pixel has 8-b data) is used to estimate the addition of the current pixel (Cur_pixel) and reference pixel (Ref_pixel). Additionally, a 12-b ADD and an ACC are required to accumulate the results from the 8-b ADD in order to determine the sum of absolute difference (SAD) value for video encoding application. Fig. 2 shows the proposed EDCA circuit design for a specific Pei of a ME. This architecture consists of blocks that generate the residue and quotient values that are used to detect the errors.

4.1Fault Model:

A more practical approach is to select specific test patterns based on circuit structural information and a set of fault models. This approach is called structural testing. Structural testing saves time and improves test efficiency. A stuck-at fault affects the state of logic signals on lines in a logic circuit, including primary inputs (PIs), primary outputs (POs), internal gate inputs and outputs, fan-out stems (sources), and fan-out branches.

A stuck-at fault transforms the correct value on the faulty signal line to appear to be stuck at a constant logic value, either logic 0 or logic 1, referred to as stuck-at-0 (SA0) or stuck-at-1 (SA1), respectively. The stuck at fault model can also be applied to sequential circuits; however, high fault coverage test generation for sequential circuits is much more difficult than for combinational circuits. The stuck-at (SA) model, must be adopted to cover actual failures in the interconnect data bus between PEs. The SA fault in a ME architecture can incur errors in computing SAD values. A distorted computational error and the magnitude of e are assumed here to be equal to SAD′- SAD where SAD′ denotes the computed SAD value with SA faults.

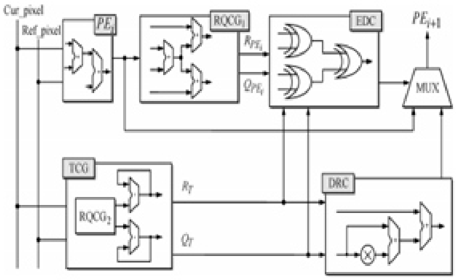


Fig.2. Proposed EDCA design

4.2Residue-And-Quotient Code Generation:

Earlier codes like Parity Codes, Berger Codes were used for detecting a single bit error. Next evolved the residue code which is generally a separable arithmetic code that estimates the residue of the given data and appends it to the data. This code is capable of detecting a single bit error. Error detection logic using residue codes are simple and it can be easily implemented. For instance, assume that N denotes an integer, N1and N2 represent data words, and m refers to the modulus.

A separate residue code is one in which N is coded as a pair (N, |N|m).Notably; |N|m is the residue of N modulo m. However, only a bit error can be detected based on the residue code. Additionally, error correction is not possible by using the residue codes. Therefore, this work presents a quotient code to assist the residue code in detecting multiple bit errors and recovering errors. The mathematical model of RQ code is simply described as follows. Assume that binary data X is expressed as

$$X = \{b_{n-1}b_{n-2}\ldots\ldots b_2b_1b_0\} = \sum_{j=0}^{n-1} b_j 2^j$$

$$X = \{b_{n-1}b_{n-2}\ldots\ldots b_2b_1b_0\} = \sum_{j=0}^{n-1} b_j 2^j$$

The RQ code for X modulo m is expressed as R= |X|_M and Q=⌊X/m⌋ respectively. Notably, ⌊i⌋ denotes the largest integer not exceeding i. According to the above RQ code expression, the corresponding circuit design of the RQCG can be realized. In order to simplify the complexity of circuit design, the implementation of the module is generally dependent on the addition operation. Additionally, based on the concept of residue code, the following definitions shown can be applied to generate the RQ code for circuit design.

Definition 1:

$$|N_1 + N_2|_m = \big||N_1|_m + |N_2|_m\big|_m \qquad (2)$$

Definition 2: Let, $N_j = n_1 + n_2 + \ldots.. + n_j$, then

$$|N_j|_m = \big||n_1|_m + |n_2|_m + \ldots. + |n_j|_m\big|_m \qquad (3)$$

To accelerate the circuit design of RQCG, the binary data shown in (1) can generally be divided into two parts:

$$\begin{aligned} X &= \sum_{j=0}^{n-1} b_j 2^j \\ &= \left(\sum_{j=0}^{k-1} b_j 2^j\right) + \left(\sum_{j=k}^{n-1} b_j 2^{j-k}\right)2^k \\ &= Y_0 + Y_1 2^k \qquad (4) \end{aligned}$$

Significantly, the value of k is equal to ⌊n/2⌋ and the data formation of Y0 and Y1 are a decimal system. If the modulus m= (2 power k)-1, then the residue code of X modulo m is given by,

$$\begin{aligned} R &= |X|_m \\ &= |Y_0 + Y_1|_m = |Z_0 + Z_1|_m = (Z_0 + Z_1)\alpha \qquad (5) \end{aligned}$$

$$\begin{aligned} Q &= \left\lfloor \frac{X}{m} \right\rfloor \\ &= \left\lfloor \frac{Y_0 + Y_1}{m} \right\rfloor + Y_1 = \left\lfloor \frac{Z_0 + Z_1}{m} \right\rfloor + Z_1 + Y_1 \\ &= Z_1 + Y_1 + \beta \qquad (6) \end{aligned}$$

where,

$$\alpha(\beta) = \begin{cases} 0(1), & \text{if } Z_0 + Z_1 = m \\ 1(0), & \text{if } Z_0 + Z_1 < m \end{cases}$$

Since the value of Y0and Y1is generally greater than that of modulus m, the equations in (5) and (6) must be simplified further to replace the complex module operation with a simple addition operation by using the parameters Z0, Z1, α and β. Based on (5) and (6), the corresponding circuit design of the RQCG is easily realized by using the simple adders (ADDs). Namely, the RQ code can be generated with a low complexity and little hardware cost.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

4.3Test Code Generation (TCG):

TCG is the main block of the proposed EDCA design. Test Code Generation (TCG) design is based on the ability of the RQCG circuit to generate corresponding test codes in order to detect errors and recovers data. The specific PEi estimates the absolute difference between the Cur_pixel of the search area and the Ref_pixel of the current macro block. The SAD value for the macro block with size of NxN can be evaluated:

$$SAD = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|X_{ij} - Y_{ij}\right|$$

$$= \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|(q_{xij}.m + r_{ij}) - (q_{yij}.m + r_{ij})\right| \qquad (7)$$

Where rxij, qxij and ryij, qyij denote the corresponding RQ code of Xij and Yij modulo m. Importantly, Xij and Yij represents the luminance pixel value of Cur_pixel and Ref_pixel, respectively. Based on the definitions shown in (2) and (3) the generation of the RQ code (RT and QT) from the TCG block is obtained. The circuit design of TCG is achieved by equations (8) and (9),

$$\begin{aligned}
\mathrm{R_T} &= \left|\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(X_{ij} - Y_{ij})\right|_m \\
&= \left||X_{00}-Y_{00})|_m + |(X_{01}-Y_{01})|_m + \ldots + |(X_{(N-1)(N-1)} - Y_{(N-1)(N-1)})|_m\right|_m \\
&= \left||(q_{x00}m + r_{x00}) - (q_{y00}m + r_{y00})|_m + \ldots + |(q_{x(N-1)(N-1)}m + r_{x(N-1)(N-1)}) \right. \\
&\qquad \left. - (q_{y(N-1)(N-1)}m + r_{y(N-1)(N-1)})|_m\right|_m \\
&= \left||(r_{x00} - r_{y00})|_m + |(r_{x01} - r_{y01})|_m + \ldots + |r_{x(N-1)(N-1)} - r_{y(N-1)(N-1)})|_m\right|_m \\
&= \left||r_{00}|_m + |r_{01}|_m + \ldots. + |r_{(N-1)(N-1)}|_m\right|_m \qquad (8)
\end{aligned}$$

$$\begin{aligned}
\mathrm{Q_T} &= \left\lfloor \frac{\sum_{j=0}^{N-1}\sum_{j=0}^{N-1}(X_{ij} - Y_{ij})}{m} \right\rfloor \\
&= \left\lfloor \frac{(X_{00}-Y_{00}) + (X_{01}-Y_{01}) + \ldots. + (X_{(N-1)(N-1)} - Y_{(N-1)(N-1)})}{m} \right\rfloor \\
&= \left\lfloor \frac{(q_{x00}m - q_{y00}m)}{m} + \frac{(r_{x00}-r_{y00})}{m} + \frac{(q_{x01}m - q_{y01}m)}{m} + \frac{(r_{x01}-r_{y01})}{m} + \ldots \right\rfloor \\
&= \left\lfloor (q_{x00}-q_{y00}) + (q_{x01}-q_{y01}) + \ldots + \frac{(r_{x00}-r_{y00}) + (r_{x01}-r_{y01}) + \ldots.}{m} \right\rfloor \\
&= q_{00} + q_{01} + \ldots + q_{(N-1)(N-1)} + \left\lfloor \frac{(r_{00} + r_{01} + \ldots + r_{(N-1)(N-1)})}{m} \right\rfloor \qquad (9)
\end{aligned}$$

4.4Error Detection Process:

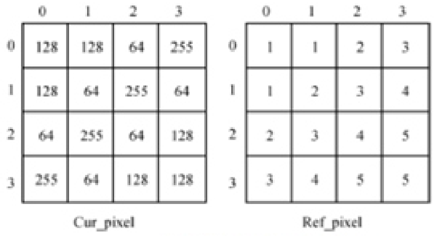
Error Detection Circuit (EDC) is used to perform the operation of error detection in the specific Pei as shown in Fig.2. This block is used to compare the output from the TCG i.e., (RT and QT) with output from RQCG1 i.e., (RPEi and QPEi), in order to detect the occurrence of an error. If the value of RPEi ≠ RT and QPEi ≠ QT, then the error in the PEi can be detected. The EDC output is then generated as a 0/1 signal to indicate that the tested PEi is error free/ with error.

4.5Error Correction Process:

The original data is recovered in the Data Recovery Circuit (DRC) during error correction process, by separating the RQ code from the TCG. The data recovery is possible by implementing the mathematical model as,

$$\begin{aligned}
SAD &= m \times Q_T + R_T \\
&= (2^j - 1) \times Q_T + R_T \\
&= 2^j \times Q_T - Q_T + R_T \qquad (10)
\end{aligned}$$

The proposed EDCA design executes the error detection and data recovery operations simultaneously. Additionally, error-free data from the tested PEi or the data recovery that results from DRC is selected by a multiplexer (MUX) to pass to the next specific PEi+1for subsequent testing.



	0	1	2	3
0	128	128	64	255
1	128	64	255	64
2	64	255	64	128
3	255	64	128	128

Cur_pixel

	0	1	2	3
0	1	1	2	3
1	1	2	3	4
2	2	3	4	5
3	3	4	5	5

Ref_pixel

Fig.3. Example of pixel values

Sample calculation:

A sample of the 16 pixels for a 4x4 macro block is taken. Fig. 3 presents an example of pixel values of the Cur_pixel and Ref_pixel. Based on (7), the SAD value of the 4x4 macro block is

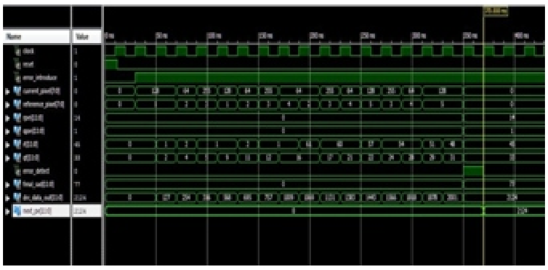
$$\begin{aligned}
SAD &= \sum_{i=0}^{3}\sum_{j=0}^{3}\left|X_{ij} - Y_{ij}\right| \\
&= |X_{00} - Y_{00}| + |X_{01} - Y_{01}| + \ldots. + |X_{33} - Y_{33}| \\
&= (128-1) + (128-1) + \ldots. + (128-5) \\
&= 2124 \qquad (11)
\end{aligned}$$

The modulo is assumed here to be m=26-1=63. Thus, based on (8) and (9), the RQ code of the SAD value shown in (11) are RPEi= RT= |2124|63 =45 and QPEi= QT= └2124/63┘ =33. Since the value of RPEi (QPEi) is equal to RT (QT), EDC is enabled and a signal "0" is generated to describe a situation in which the specific is error-free. Conversely, if SA1 and SA0 errors occur in bits 1 and 12 of a specific PEi, i.e., the pixel values of PEi, 2124=(100001001100)2is turned into 77= (000001001101)2, resulting in the transformation of the RQ code of RPEi and QPEi into |77|=14and └77/63┘ =1. Thus, an error signal "1" is generated from EDC and sent to the MUX in order to select the recovery results from DRC.

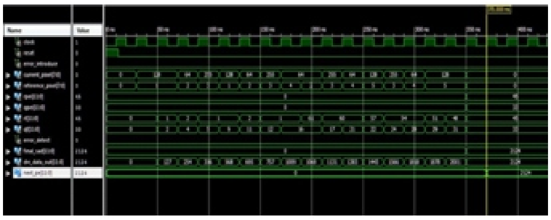
RESULTS

Simulation Results:

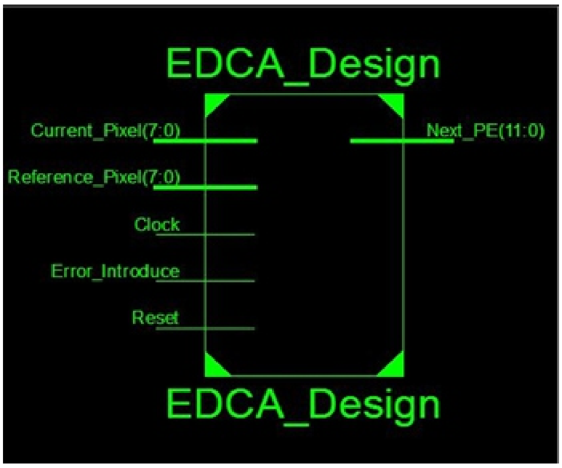
Top module simulation with error:



Top module simulation with out error:



7.1.RTL Schematic:



CONCLUSION AND FUTURE SCOPE:

The motion estimation process is done by the video coding system to find the motion vector pointing to the best prediction macro-block in a reference frame or field. An error occurring in ME can cause degradation in the video quality. Order to make the coding system efficient an Error Detection and Correction Architecture (EDCA) is designed for detecting the errors and recovering the data of the PEs in a ME. Based on the RQ code, a RQCG-based TCG design is developed to generate the corresponding test codes that can detect multiple bit errors and recover data. The proposed EDCA design is implemented using VHDL and synthesized by using Xilinx with the target device as XC3S500E from the family Spartan 3E. Experimental results indicate that the proposed EDCA design can effectively detect the errors and recover data in PEs of a ME. In present project we take the 2 frames for motion estimation one is current frame and reference frame, and we compare the value only SAD (Sum of Absolute Difference) , in future we take 3 frames one is current frame and 2 frames are reference frames, One is forward frame and one is backward frame in video framing systems, and we compare the values of Mean Square Error (MSE) along with SAD value.

REFERENCES:

[1]Advanced Video Coding for Generic Audiovisual Services, ISO/IEC 14496-10:2005 (E), Mar. 2005, ITU-T Rec. H.264 (E).

[2]Information Technology-Coding of Audio-Visual Objects—Part 2: Vi-sual, ISO/IEC 14 496-2, 1999.

[3]Y. W. Huang, B. Y. Hsieh, S. Y. Chien, S. Y. Ma, and L. G. Chen, "Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 4, pp. 507–522, Apr. 2006.

[4]C. Y. Chen, S. Y. Chien, Y. W. Huang, T. C. Chen, T. C. Wang, and L. G. Chen, "Analysis and architecture design of variable block-size motion estimation for H.264/AVC," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 53, no. 3, pp. 578–593, Mar. 2006.

[5]T. H. Wu, Y. L. Tsai, and S. J. Chang, "An efficient design-for-testa-bility scheme for motion estimation in

H.264/AVC," in Proc. Int. Symp. VLSI Design, Autom. Test, Apr. 2007, pp. 1–4.

[6]M. Y. Dong, S. H. Yang, and S. K. Lu, "Design-for-testability tech-niques for motion estimation computing arrays," in Proc. Int. Conf. Commun., Circuits Syst., May 2008, pp. 1188–1191.

[7]Y. S. Huang, C. J. Yang, and C. L. Hsu, "C-testable motion estimation design for video coding systems," J. Electron. Sci. Technol., vol. 7, no. 4, pp. 370–374, Dec. 2009.

[8]D. Li, M. Hu, and O. A. Mohamed, "Built-in self-test design of motion estimation computing array," in Proc. IEEE Northeast Workshop Cir-cuits Syst., Jun. 2004, pp. 349–352.

[9]Y. S. Huang, C. K. Chen, and C. L. Hsu, "Efficient built-in self-test for video coding cores: A case study on motion estimation computing array," in Proc. IEEE Asia Pacific Conf. Circuit Syst., Dec. 2008, pp. 1751–1754.

[10]W. Y Liu, J. Y. Huang, J. H. Hong, and S. K. Lu, "Testable design and BIST techniques for systolic motion estimators in the transform domain," in Proc. IEEE Int. Conf. Circuits Syst., Apr. 2009, pp. 1–4.

[11]J. M. Portal, H. Aziza, and D. Nee, "EEPROM memory: Threshold voltage built in self diagnosis," in Proc. Int. Test Conf., Sep. 2003, pp. 23–28.

[12]J. F. Lin, J. C. Yeh, R. F. Hung, and C. W. Wu, "A built-in self-repair design for RAMs with 2-D redundancy," IEEE Trans. Vary Large Scale Integr. (VLSI) Syst., vol. 13, no. 6, pp. 742–745, Jun. 2005.

[13]C. L. Hsu, C. H. Cheng, and Y. Liu, "Built-in self-detection/correc-tion architecture for motion estimation computing arrays," IEEE Trans. Vary Large Scale Integr. (VLSI) Systs., vol. 18, no. 2, pp. 319–324, Feb. 2010.

[14]C. H. Cheng, Y. Liu, and C. L. Hsu, "Low-cost BISDC design for motion estimation computing array," in Proc. IEEE Circuits Syst. Int. Conf., 2009, pp. 1–4.

About Authors:

1.Pattan Shareef Was Born In A.S.Peta [P] [M], S.P.S.R.Nellore [Dt], Andhra Pradesh, India. He Received The B.Tech Degree In Electronics And Communication Engineering From Jnt University, Anantapur In 2013 And Pursuing M.Tech Degree In Vlsi & Esd From Jnt University, Anantapur. He Completed His B.Tech Degree In Priyadarshini Institute Of Technology, Kanupartipadu [V], Nellore [M], S.P.S.R.Nellore [Dt], Andhra Pradesh, India, And M.Tech Degree In Skr College Of Engineering & Technology, Konduru Satram [V], Manubolu [M], S.P.S.R.Nellore [Dt], Andhra Pradesh, India. He Can Be Reachable At Sharu.Pattan@Gmail.Com

2.Mr. Goni Mahendra Was Born In Andhra Pradesh, India. He Received The B.Tech Degree In Elecronics And Communication Engineering And M.Tech Degree In Vlsi System Design From P.B.R.Visvodaya Institute Of Technology & Science, Kavali - 524201, S.P.S.R.Nellore [Dt], Andhrapradesh From Jnt University, Anantapur. He Had Working As An Associate Professor In Dept. Of Electronics And Communication Engineering And Vlsi System Design From Skr College Of Engineering & Technology, Konduru Satram [V], Manubolu [M], S.P.S.R Nellore [Dt], Andhra Pradesh, India. He Can Be Reachable At Mahendra.Goni@Gmail.Com