

Architecture to Detect and Correct Error in Motion Estimation of Video System Based on RQ Code

¹Purva Choudhary, ²K.Mallikarjuna Lingam

¹PG Scholar, Dept of ECE, MRCET, Hyderabad

²AMIETE, M.Tech, (Ph.D), Assoc.Prof, Dept. of ECE, MRCET, Hyderabad

ABSTRACT

Motion Estimation (ME) is the process of determining the motion vectors that describe the transformation from one 2D image to other, usually from adjacent frames in a video sequence. The process of ME is the critical part of any video coding system as the video quality will be affected if an error has occurred in the ME. In order to test processing element of Motion Estimation in a video coding system, design Architecture to Detection and Correction of errors of PE. This Design based on the Residue-and-Quotient (RQ) code. This PE process the 8x8 block of pixel for video systems, mainly this PE is SAD calculator for motion estimation in video systems. Proposed architecture is detect the errors and corrected by recovery circuit.

Processing Element, Error detection, Error correction, Motion estimation, Residue-and-quotient (RQ) code

INTRODUCTION

Multimedia applications are becoming more flexible and reliable with the advancements in semiconductors, Digital Signal Processing and Communication technologies. 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. ME is of priority concern in removing the temporal redundancy between the successive frames in a video coding system and also it consumes more time. ME is considered as the intensive unit in terms of computation [1].

Regular arrangement of PEs with size 8x8 constitutes a ME. 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 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 [8]. 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 [6].

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 [3]. 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 [4]. 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 [5].

**Address for correspondence:*

jtinureddy@gmail.com

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 behavior of defects, and (2) it should be computationally efficient in terms of fault simulation and test pattern generation.

The rest of this paper is organized as follows. Section 2 introduces the proposed EDCA, fault model definition, blocks in the architecture and test method. Next, Section 3 evaluates the performance in terms of area, delay to demonstrate the feasibility of the proposed EDCA for ME testing applications. Conclusions are finally drawn in Section 4.

PROPOSED EDCA DESIGN

The proposed EDCA scheme shown in Fig. 1 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 [9]. 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.

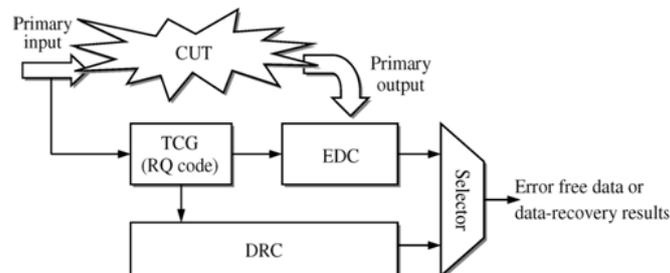


Figure1. Conceptual View of the proposed EDCA design

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) [2]. 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 PE_i of a ME [9]. This architecture consists of blocks that generate the residue and quotient values that are used to detect the errors.

Fault 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 [7]. 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.

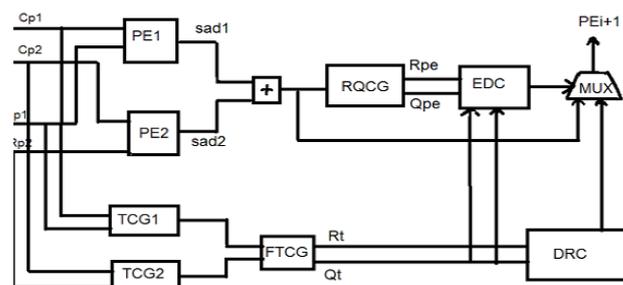


Figure2. Proposed EDCA design

Residue-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 [10]. 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, N_1 and N_2 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

$$X = \{b_{n-1}b_{n-2} \dots b_2b_1b_0\} = \sum_{j=0}^{n-1} b_j 2^j \quad (1)$$

The RQ code for X modulo m is expressed as $R = |X|_m$ and $Q = \lfloor X/m \rfloor$, respectively [9]. Notably, $\lfloor i \rfloor$ 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:

$$|N_1 + N_2|_m = \left| |N_1|_m + |N_2|_m \right|_m \quad (2)$$

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

$$|N_j|_m = \left| |n_1|_m + |n_2|_m + \dots + |n_j|_m \right|_m \quad (3)$$

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

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

Significantly, the value of k is equal to $\frac{n}{2}$ and the data formation of Y_0 and Y_1 are a decimal system. If the modulus $m = 2^k + 1$, then the residue code of X modulo m is given by,

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

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

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 Y_0 and Y_1 is 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 Z_0, Z_1, α and β [10].

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.

Test Code Generation

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 PE_i 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 $N \times N$ can be evaluated

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

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

$$\begin{aligned}
 Q_T &= \left[\frac{\sum_{j=0}^{N-1} \sum_{y=0}^{N-1} (X_y - Y_y)}{m} \right] \\
 &= \left[\frac{(X_{00} - Y_{00}) + (X_{01} - Y_{01}) + \dots + (X_{(N-1)(N-1)} - Y_{(N-1)(N-1)})}{m} \right] \\
 &= \left[\frac{(q_{x00}m - q_{y00}m) + (r_{x00} - r_{y00}) + (q_{x01}m - q_{y01}m) + (r_{x01} - r_{y01}) + \dots}{m} \right] \\
 &= \left[\frac{(q_{x00} - q_{y00}) + (q_{x01} - q_{y01}) + \dots + \frac{(r_{x00} - r_{y00}) + (r_{x01} - r_{y01}) + \dots}{m}}{m} \right] \\
 &= q_{00} + q_{01} + \dots + q_{(N-1)(N-1)} + \left[\frac{(r_{00} + r_{01} + \dots + r_{(N-1)(N-1)})}{m} \right] \quad (9)
 \end{aligned}$$

Error Detection Process

Error Detection Circuit (EDC) is used to perform the operation of error detection in the specific PE_i as shown in Fig.2. This block is used to compare the output from the TCG i.e., (R_T and Q_T) with output from RQCG1 i.e., (RPE_i and QPE_i), in order to detect the occurrence of an error. If the value of $RPE_i \neq R_T$ and $QPE_i \neq Q_T$, then the error in the PE_i can be detected. The EDC output is then generated as a 0/1 signal to indicate that the tested PE_i is error free/ with error.

Error 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

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

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

5	176	193	168	168	170	167	165	0	172	198	190	149	159	149	179
6	176	158	172	162	177	168	151	17	166	131	175	168	195	192	140
5	167	172	232	158	61	145	214	12	167	177	255	140	39	123	203
33	179	169	174	5	5	135	178	22	178	170	145	1	28	146	187
8	104	180	178	172	197	188	169	11	103	183	190	160	207	189	155
63	5	102	101	160	142	133	139	60	9	80	103	157	144	115	156
51	47	63	5	180	191	165	5	60	48	67	11	176	196	139	19
49	53	43	5	184	170	168	74	49	58	47	0	190	168	176	60

RESULTS AND DISCUSSIONS

The EDCA design is developed in a top down design methodology using Xilinx 13.2 with Verilog coding. The top down design methodology is code mixed version of both behavioral and structural. The architecture consists of basic modules like Process element, TCG, RQ code generator, Multiplexer, EDC and DRC modules. Residue-and-quotient code is used in the existing error detection architecture that has 1 Processing Elements (PEs) and 1 Test Code Generation (TCG) blocks for computing the test codes for each pixel value in the macro block. Though this architecture can detect multiple bit errors, by process the 4x4 block pixels, we get the SAD value in 16 clock cycles, the proposed architecture is used which has a two PE and a two TCG for computing the test codes for high throughput. We get the SAD value in 8 clocks cycles by using 2 processing elements. And we process the 8X8 block of pixels in this architecture. This one gives the sad value in 32 clock cycles.

Verification of the proposed design is performed by using verilog. The performance of the proposed architecture design is estimated in terms of its area. In this architecture, the pixel values in the 8x8 macro block are taken out for the code generation based on the clocking signal. At each triggering edge of the clock the 8-bit pixel value from the ref_pixel and cur_pixel is taken and given to the TCG and PE blocks for code generation. The generated code from test code generation block and the RQCG output is compared in the EDC block, output of which indicates the error.

In Figure3 simulation results shows that it process the 8X8 block of pixels (64 pixels) value with in 32 clock cycles by parallel computing SAD processing elements. In simulation results it shows the with error results, we get the wrong SAD value from SAD output and we get correct SAD value from the DRC (data recovery circuit). It shows in above diagram. So we pass the sad value to next PE.

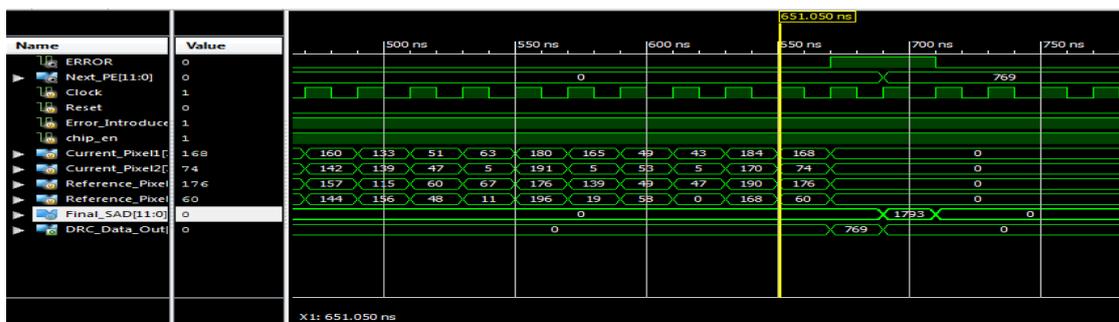


Figure3. With error simulation result

In Figure4 results shows the without error sad value simulation results, we get the correct value from both units, that is DRC and SAD PE (processing element). In Below simulation results there is no error, we get correct SAD value from SAD output and DRC output so we send the SAD output to next pe.

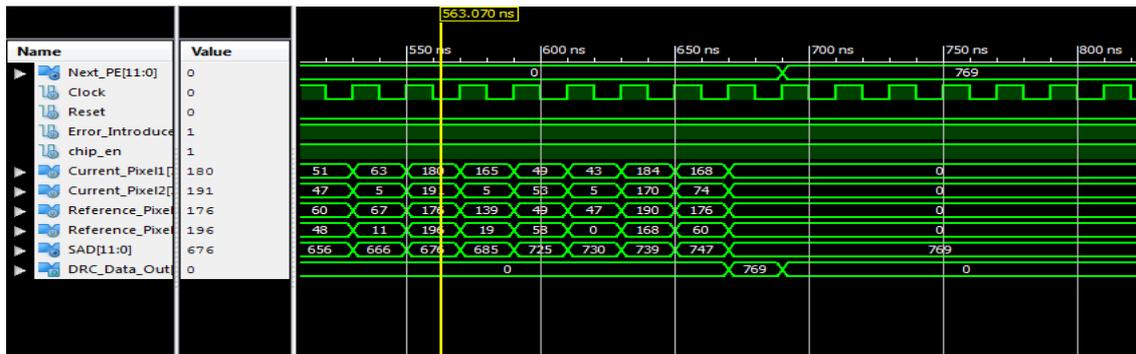


Figure4. Without error simulation result

CONCLUSION

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. In 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.

REFERENCES

- [1] 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.
- [2] S.Surin and Y.H. Hu,”Frame-level pipeline motion estimation array processor,” IEEE Trans. Circuits Syst. Video Technol., vol. 11, no.2, pp.248-251, Feb.2001.
- [3] M.Y.Dong, S.H.Yang, and S.K. Lu,” Design-for-testability techniques for motion estimation computing arrays,” in Proc. Int. Conf. Commun., Circuits Syst., May 2008, pp.1188-1191.
- [4] 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. Very Large Scale Integr. (VLSI) Syst., vol.13, no. 6, pp.742-745, Jun. 2005.
- [5] C.L. Hsu, C.H. Cheng, and Y. Liu,” Built-in self detection/correction architecture for motion estimation computing arrays,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no.2, pp. 319-324, Feb.2010.
- [6] S.Bayat-Sarmadi and M.A. Hasan,”On concurrent detection of errors in polynomial basis multiplication,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 15, no.4, pp. 413-426, Apr.2007.
- [7] D. K. Park, H.M.Cho, S.B. Cho, and J.H. Lee, “A fast motion estimation algorithm for SAD optimization in sub-pixel,” Proc. Int. Symp. Integr. Circuits, Sep. 2007, pp. 528-531
- [8] C.W. Chiou, C.C. Chang, C.Y. Lee, T.W. Hou, and J.M. Lin,” Concurrent error detection and correction in Gaussian normal basis multiplier over GF (2^m), IEEE Trans. Comput., vol.58, no.6, pp.851-857, Jun.2009.
- [9] Chang- Hsin Cheng, Yu Liu, and Chun-Lung Hsu, Member, IEEE “ Design of an Error Detection and Data recovery Architecture for Motion Estimation Testing Applications,” IEEE transactions on Very Large scale integration (VLSI) systems, vol.20, no.4, April.2012.
- [10] L.Breveglieri, P. Maistri, and I. Koren, “ Anote on error detection in an RSA architecture by means of residue codes,” in Proc. IEEE Int. Symp. On-Line Testing, Jul. 2006, 176 177

AUTHORS' BIOGRAPHY



PURVA CHOUDHARY received B.Tech degree in Electronics and Communication Engineering from JIET SETG Jodhpur, Rajasthan, India, affiliated to Rajasthan Technical University, Kota, India in 2012. She is pursuing M.Tech in VLSI Design& Embedded Systems at Malla Reddy College of Engineering and Technology, Hyderabad, India.



Mr K.Mallikarjuna Lingam is an Associate professor at Malla Reddy College of Engineering and Technology, Hyderabad. He is a Research Scholar of J.N.T.U.K, Kakinada. He received his Bachelor degree in Electronics and Telecommunication Engineering from IETE, New Delhi in 2004. M.Tech, VLSI Design from Satyabhama University, Chennai in 2007. He is Active Member of IETE. His current research interest includes Information Retrieval and Content based Image/Video Retrieval.