

Design of an Built In Self Detection & Correction (BISDC) Architecture for Motion Estimation Computing Arrays

KASTHALA CHARAN TEJA¹, YELITHOTI SRAVAN KUMAR²

¹Research Scholar, Dept of EEE, Sri Ramulu College of Engineering, Tanuku, AP-INDIA,
E-mail: Charan419@gmail.com

²Asst Prof, Dept of EEE, Sri Ramulu College of Engineering, Tanuku, AP-INDIA,
E-mail: sravan.yelithoti@gmail.com.

Abstract: Given the critical role of motion estimation computing array (MECA) in a video coder, testing such a module is of priority concern. While focusing on the testing of MECA in a video coding system, this work presents an Built in self detection and correction (BISDC) 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 BISDC design. Experimental results indicate that the proposed BISDC design for ME testing can detect errors and recover data. The functional verification and synthesis can be done by Xilinx ISE 12.3 Version.

Keywords: Processing Element (PE), Built In Self-Detection & Correction (BISDC), Motion Estimation (ME), Residue-And-Quotient Code Generation (RQCG), Test Code Generator (TCG).

I. INTRODUCTION:

The new Joint Video Team (JVT) video coding standard has garnered increased attention recently. Generally, motion estimation computing array (MECA) performs up to 50% of computations in the entire video coding system, and is typically considered the computationally most important part of video coding systems. Thus, integrating the MECA into a system-on-chip (SOC) design has become increasingly important for video coding applications. Additionally, the visual quality and peak signal-to-noise ratio (PSNR) at a given bit rate are influenced if an error occurred in ME process. Although advances in VLSI technology allow integration of a large number of processing elements (PEs) in an MECA into an SOC, this increases the logic-per-pin ratio, thereby significantly decreasing the efficiency of chip logic testing. For a commercial chip, a video coding system must introduce design for testability (DFT), especially in an MECA. The

objective of DFT is to increase the ease with which a device can be tested to guarantee high system reliability. Many DFT approaches have been developed. These approaches can be divided into three categories: ad hoc (problem oriented), structured, and built-in self-test (BIST). Among these techniques, BIST has an obvious advantage in that expensive test equipment is not needed and tests are low cost. BIST schemes not only detect faults but also specify their location for error correcting. BIST can generate test simulation and test responses without outside support. The extended BIST schemes generally focus on memory circuit; testing-related issues of video coding have been addressed. Thus, exploring the feasibility of an embedded testing approach to detect errors and recover data of a ME is of worthwhile interest. Additionally, the reliability issue of numerous processing elements (PEs) in a ME can be improved by enhancing the capabilities of concurrent error detection (CED). The CED approach can detect errors through conflicting and undesired results generated from operations on the same operands. CED can also test the circuit at full operating speed without interrupting a system. Thus, based on the CED concept, this work develops an BISDC architecture based on the RQ code to detect errors and recovery data in PEs of a ME.

II. MOTION ESTIMATION

A. Existing system

The Motion Estimation Computing Array is used in Video Encoding applications to calculate the best motion between the current frame and reference frames. The MECA in decoding application occupies large amount of area (LUT) and timing penalty. By introducing the concept of Built-in Self test technique the area overhead is increased in less amount of area.

B. Proposed system

In this Project the Built-in Self test Technique (BIST) is included in the MECA and in each of

Processing Element in MECA. Thus by introducing the BIST Concept the testing is done internally without Connecting outside testing Requirements. So the area (LUT) required is also reduces. And in this Project the Errors in MECA are Calculated and the Concept of Diagnoses i.e. Self Detect and Self Repair Concepts are introduced.

C. Block Matching Motion Estimation

Several different algorithms derived from various theories, including object-oriented tracking, exist to perform motion estimation. Among them, one of the most popular algorithms is the Block Matching Motion Estimation (BME) algorithm. BME treats a frame as being composed of many individual sub-frame blocks, known as macro Blocks. Motion vectors are then used to encode the motion of the macro Blocks through frames of video via a frame by frame matching process. When a frame is brought into the encoder for compression, it is referred to as the current frame. It is the goal of the BME unit to describe the motion of the macro Blocks within the current frame relative to a set of reference frames. The reference frames may be previous or future frames relative to the current frame. Each reference frame is also divided into a set of sub frame blocks, which are equal to the size of the macro Blocks. These blocks are referred to as reference Blocks. The BME algorithm will scan several candidate reference Blocks within a reference frame to find the best match to a macro Block. Once the best reference Block is found a motion vector is then calculated to record the spatial displacement of the macro Block relative to the matching reference Block, as shown in Figure 1.

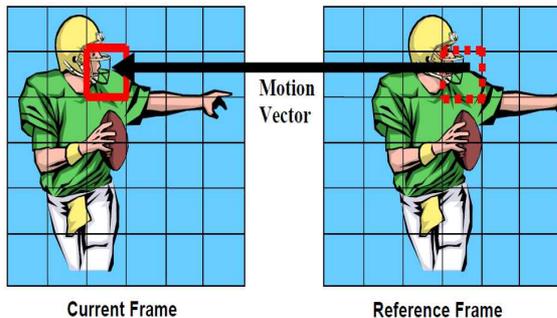


Fig 1: Block Matching between Current & reference frames

D. Search Windows

When searching a reference frame for possible macro Block matches, the entire reference frame is not searched. Instead the search is restricted within a search window. Search windows in most H.264 implementations have a size of 48-pixel (rows) x 63-pixel (columns). In this Paper, we use the same 48x63 search window size. This window consists of a vertical search range of [-16, +16] and a horizontal search range

of [-24, +23] pixels as illustrated in Figure 2. In the figure 2, the dashed large rectangle in the reference frame represents the 48x63 search window area. The dashed square in the top left corner of the search window represents the first of the 1584 possible candidate 16x16 reference Blocks. Each subsequent

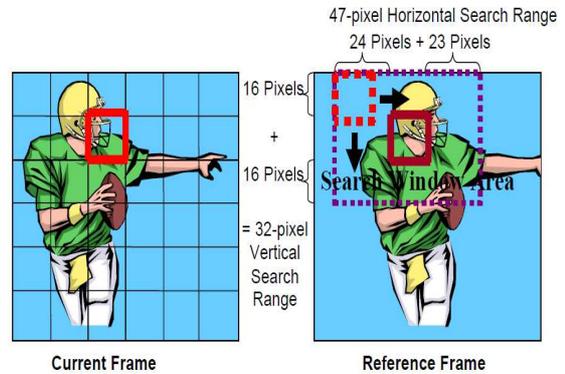


Fig 2: Search Window size Definition

reference Block is offset by either one pixel row or one pixel column from its predecessor while the entire search window area is covered by the overlapping candidate reference Blocks. Note that the original 16x16 macro Block is positioned at the centre of the search window. In order to compare it to every candidate reference Block within the search window, the macro Block has a maximum displacement of 24 pixels to the left, 23 pixels to the right, 16 pixels up, and 16 pixels down from its original position – resulting in a horizontal search range of [-24, +23] and a vertical search range of [-16, +16].

III. RQ CODE GENERATION

Coding approaches such as parity code, Berger code, and residue code have been considered for design applications to detect circuit errors. Residue code is generally separable arithmetic codes by estimating a residue for data and appending it to data. Error detection logic for operations is typically derived by a separate residue code, making the detection logic is simple and easily implemented. Error detection logic for operations is typically derived using a separate residue code such that detection logic is simply and easily implemented. However, only a bit error can be detected based on the residue code. Additionally, an error can't be recovered effectively by using the residue codes. Therefore, this work presents a quotient code, which is derived from the residue code, to assist the residue code in detecting multiple errors and recovering errors. 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. The mathematical model of RQ code is simply described as follows. Assume that binary data X is expressed as

Design of an Built In Self Detection & Correction (BISDC) Architecture for Motion Estimation Computing Arrays

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

The RQ code of X modulo m expressed as $R=[X]_m$, $Q=[X/m]$, respectively. Notably [i] denote 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:

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

Definition 2:

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

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

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 $\lceil n/2 \rceil$ and the data formation of Y0 and Y1 are a decimal system. If the modulus $m = 2k - 1$, then the residue code of modulo 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

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

Notably, since the value of $Y_0 + Y_1$ 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 . 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.

IV. PROPOSED BISDC ARCHITECTURE DESIGN

Figure 3 shows the conceptual view of the proposed BISDC scheme, which comprises two major circuit designs, i.e. error detection circuit (EDC) and data recovery circuit (DRC), to detect errors and recover the corresponding data in a specific CUT. The test code generator (TCG) in Figure 3 utilizes the concepts of RQ code to generate the corresponding test codes for error detection and data recovery. In other words, the test codes from TCG and the primary output from CUT are delivered to EDC to determine whether the CUT has errors. DRC is in charge of recovering data from TCG. Additionally, a selector is enabled to export error-free data or data-recovery results. Importantly, an array based computing structure, such as ME, discrete cosine transform (DCT), iterative logic array (ILA), and finite impulse filter (FIR), is feasible for the proposed BISDC scheme to detect errors and recover the corresponding data.

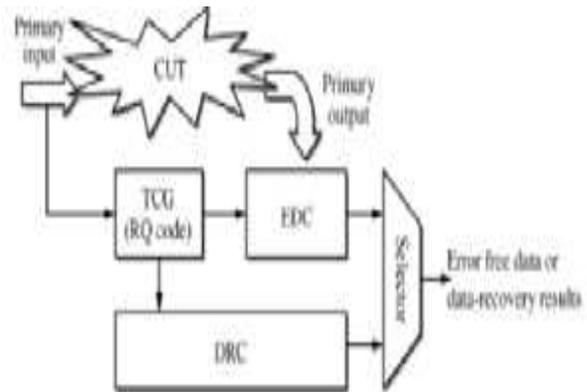


Fig 3: Conceptual view of the proposed BISDC architecture

This work adopts the systolic ME [19] as a CUT to demonstrate the feasibility of the proposed BISDC architecture. A ME consists of many PEs 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 applications. Notably, some registers and latches may exist in ME to complete the data shift and storage. Figure 5 shows an example of the proposed BISDC circuit design for a specific PE_i of a ME. The fault model definition, RQCG-based TCG design, operations

of error detection and data recovery, and the overall test strategy are described carefully as follows.

A. SAD Tree

PEs utilizes the concept of the proposed SAD Tree architecture. The proposed SAD Tree is a 2-D intra-level architecture and consists of a 2-D PE array and one 2-D adder tree with propagation registers. Current pixels are stored in each PE, and reference pixels are stored in propagation registers for data reuse. In each cycle, current and reference pixels are inputted to PEs. Simultaneously, continuous reference pixels in a row are inputted into propagation registers to update

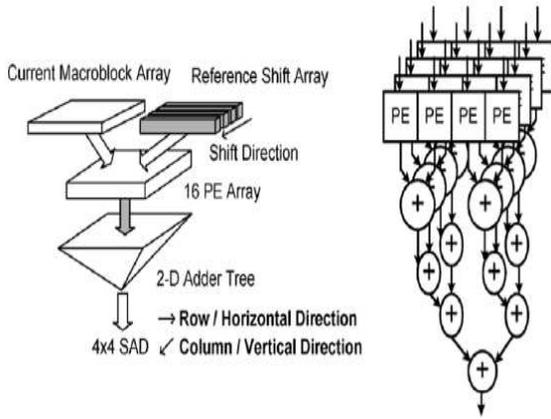


Fig 4: SAD Tree Architecture

reference pixels. In propagation registers, reference pixels are propagated in the vertical direction row by row. In SAD Tree architecture, all distortions of a searching candidate are generated in the same cycle, and by an adder tree, distortions are accumulated to derive the SAD in one cycle.

B. Fault Model

The PEs is essential building blocks and are connected regularly to construct a ME. Generally, PEs are surrounded by sets of ADDs and accumulators that determine how data flows through them. PEs can thus be considered the class of circuits called ILAs, whose testing assignment can be easily achieved by using the fault model, cell fault model (CFM). Using CFM has received considerable interest due to accelerated growth in the use of high-level synthesis, as well as the parallel increase in complexity and density of integration circuits (ICs). Using CFM makes the tests independent of the adopted synthesis tool and vendor library. Arithmetic modules, like ADDs (the primary element in a PE), due to their regularity, are designed in an extremely dense configuration. Moreover, a more comprehensive fault model, i.e. the stuck-at (SA) model, must be adopted to cover actual failures in the interconnect data bus between PEs. The SA fault is a well known structural fault model, which assumes that faults cause a line in the circuit to behave as if it were

permanently at logic “0” (stuck-at 0 (SA0)) or logic “1” [stuck-at 1 (SA1)]. The SA fault in a ME architecture can incur errors in computing SAD values. A distorted computational error(e) and the magnitude of (e) are assumed here to be equal to SAD|- SAD, where SADdenotes the computed SAD value with SA faults.

C. TCG Design

According to Figure 5, TCG is an important component of the proposed BISDC architecture. Notably, TCG design is based on the ability of the RQCG circuit to generate corresponding test codes in order to detect errors and recover data. The specific in Figure 5 estimates the absolute difference between the Cur_pixel of the search area and the Ref_pixel of the current macro block. Thus, by utilizing PEs, SAD shown in as follows, in a macro block with size of N X N can be evaluated:

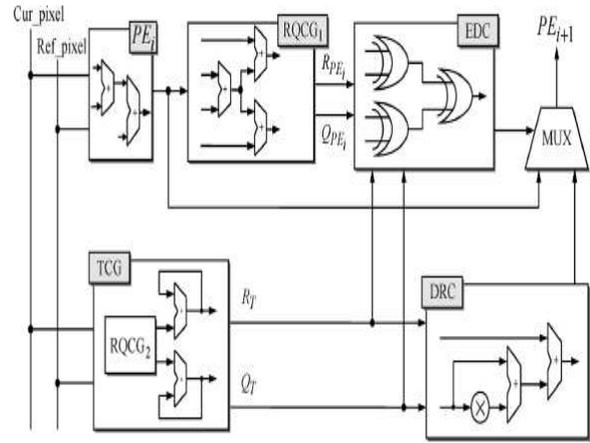


Fig 5: A specific PEi testing processes of the proposed BISDC architecture

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

$$= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |(q_{xij} \cdot m + r_{xij}) - (q_{yij} \cdot m + r_{yij})| \quad (7)$$

$$R_T = \left| \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (X_{ij} - Y_{ij}) \right|_m$$

$$= |(X_{00} - Y_{00})|_m + |(X_{01} - Y_{01})|_m + \dots$$

$$+ |(X_{(N-1)(N-1)} - Y_{(N-1)(N-1)})|_m |_m$$

$$= |(q_{x00} \cdot m + r_{x00}) - (q_{y00} \cdot m + r_{y00})|_m$$

$$+ \dots |(q_{x(N-1)(N-1)} \cdot m + r_{x(N-1)(N-1)})$$

$$- (q_{y(N-1)(N-1)} \cdot m + r_{y(N-1)(N-1)})|_m |_m$$

$$= |(r_{x00} - r_{y00})|_m + |(r_{x01} - r_{y01})|_m + \dots$$

$$+ |(r_{x(N-1)(N-1)} - r_{y(N-1)(N-1)})|_m |_m$$

$$= |r_{00}|_m + |r_{01}|_m + \dots + |r_{(N-1)(N-1)}|_m |_m \quad (8)$$

Design of an Built In Self Detection & Correction (BISDC) Architecture for Motion Estimation Computing Arrays

where in (7) r_{xij} , q_{xij} and r_{yij} , q_{yij} denote the corresponding RQ code of X_{ij} and Y_{ij} modulo m . Importantly, X_{ij} and Y_{ij} represent the luminance pixel value of Cur_pixel and Ref_pixel, respectively. Based on the residue code, the definitions shown in (2) and (3) can be applied to facilitate generation of the RQ code (RT and QT) form TCG. Namely, the circuit design of TCG can be easily achieved (see Fig. 6) by using (8) & (9) we use to derive the corresponding RQ code.

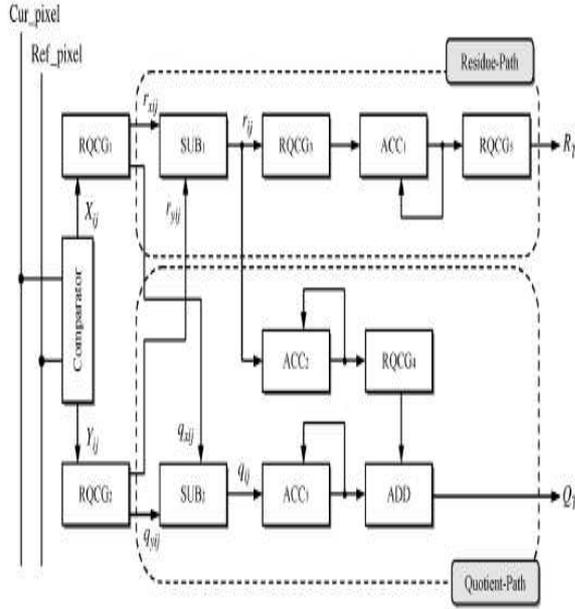


Fig 6: Circuit design of the TCG

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

D. BISDC Processes

Fig. 5 clearly indicates that the operations of error detection in a specific PE_i is achieved by using EDC, which is utilized to compare the outputs between TCG and $RQCG_1$ in order to determine whether errors have occurred. If the values of $R_{PE_i} \neq R_T$ and/or $Q_{PE_i} \neq Q_T$, then the errors in a specific PE_i can be detected. The EDC output is then used to generate a 0/1 signal to indicate that the tested PE_i is error-free/errancy. This

work presents a mathematical statement to verify the operations of error detection. Based on the definition of the fault model, the SAD value is influenced if either SA1 and/or SA0 errors have occurred in a specific PE_i . In other words, the SAD value is transformed to $SAD' = SAD + e$ if an error occurred. Notably, the error signal e is expressed as

$$e = q_e \cdot m + r_e \quad (10)$$

$$\begin{aligned}
 R_{PE_i} &= |SAD'|_m \\
 &= \left\lfloor \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (X_{ij} - Y_{ij}) + e \right\rfloor_m \\
 &= \left\lfloor |r_{00}|_m + |r_{01}|_m + \dots + |r_{(N-1)(N-1)}|_m + |r_e|_m \right\rfloor_m \quad (11) \\
 Q_{PE_i} &= \left\lfloor \frac{SAD'}{m} \right\rfloor \\
 &= \left\lfloor \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (X_{ij} - Y_{ij}) + e}{m} \right\rfloor \\
 &= q_{00} + q_{01} + \dots + q_{(N-1)(N-1)} + q_e \\
 &\quad + \left\lfloor \frac{(r_{00} + r_{01} + \dots + r_{(N-1)(N-1)}) + r_e}{m} \right\rfloor. \quad (12)
 \end{aligned}$$

to comply with the definition of RQ code. Under the faulty case, the RQ code from $RQCG_2$ of the TCG is still equal to (8) and (9). However, R_{PE_i} and Q_{PE_i} are changed to (12) and (13) because an error e has occurred. Thus, the error in a specific PE_i can be detected if and only if (8) \neq (11) and/or (9) \neq (12): During data recovery, the circuit DRC plays a significant role in recovering RQ code from TCG. The data can be recovered by implementing the mathematical model as

$$\begin{aligned}
 SAD &= m \times Q_T + R_T \\
 &= (2^j - 1) \times Q_T + R_T \\
 &= 2^j \times Q_T - Q_T + R_T. \quad (13)
 \end{aligned}$$

To realize the operation of data recovery in (13), a Barrel shift [3] and a corrector circuits are necessary to achieve the functions of $(2^j \times Q_T)$ and $(-Q_T + R_T)$, respectively. Notably, the proposed BISDC design 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.

E. Overall Test Strategy

By extending the testing processes of a specific PE_i in Fig. 5, Fig. 7 illustrates the overall BISDC architecture design of a ME. First, the input data of Cur_pixel and Ref_pixel are sent simultaneously to PEs and TCGs in order to estimate the SAD values and Design of An Error generate the test RQ code R_T and

Q_T . Second, the SAD value from the tested object PE_i , which is selected by MUX₁, is then sent to the RQCG circuit in order to generate R_{PEi} and Q_{PEi} codes.

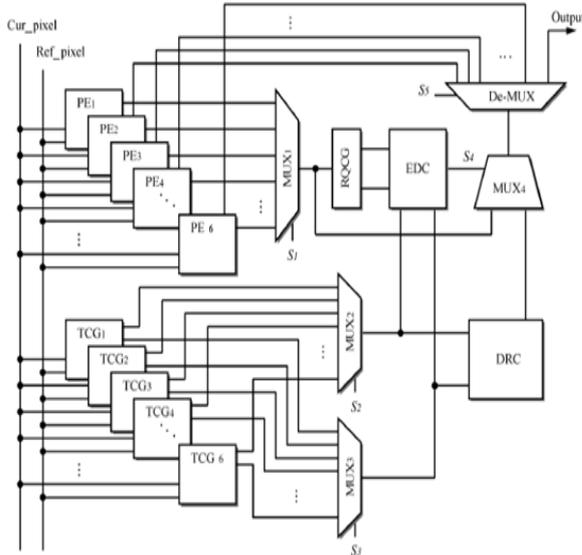


Fig 7: Proposed BISDC architecture design for a ME.

Meanwhile, the corresponding test codes R_{Ti} and Q_{Ti} from a specific TCG_i are selected simultaneously by MUXs 2 and 3, respectively. Third, the RQ code from TCG_i and RQCG circuits are compared in EDC to determine whether the tested object PE_i have errors. The tested object PE_i is error-free if and only if $R_{PEi} = R_{Ti}$ and $Q_{PEi} = Q_{Ti}$. Additionally, DRC is used to recover data encoded by TCG_i , i.e. the appropriate R_{Ti} and Q_{Ti} codes from TCG_i are selected by MUXs 2 and 3, respectively, to recover data. Fourth, the error-free data or data recovery results are selected by MUX₄. Notably, control signal S_4 is generated from EDC, indicating that the comparison result is error-free ($S_4 = 0$) or errancy ($S_4 = 1$). Finally, the error-free data or the data-recovery result from the tested object PE_i is passed to a De-MUX, which is used to test the next specific PE_{i+1} ; otherwise, the final result is exported.

V. RESULTS

Curr pixels : 10 15 25 18 10 14
 Ref pixels : 12 29 34 32 11 15
 PE output: sum of absolute difference (SAD) value = 41
 RQCG output: $R_{PEi} = \text{mod}11 = 6$, $Q_{PEi} = \text{q}11 = 5$
 TCG output: $R_T = \text{mod}10 = 6$, $Q_T = \text{sum}11 = 5$

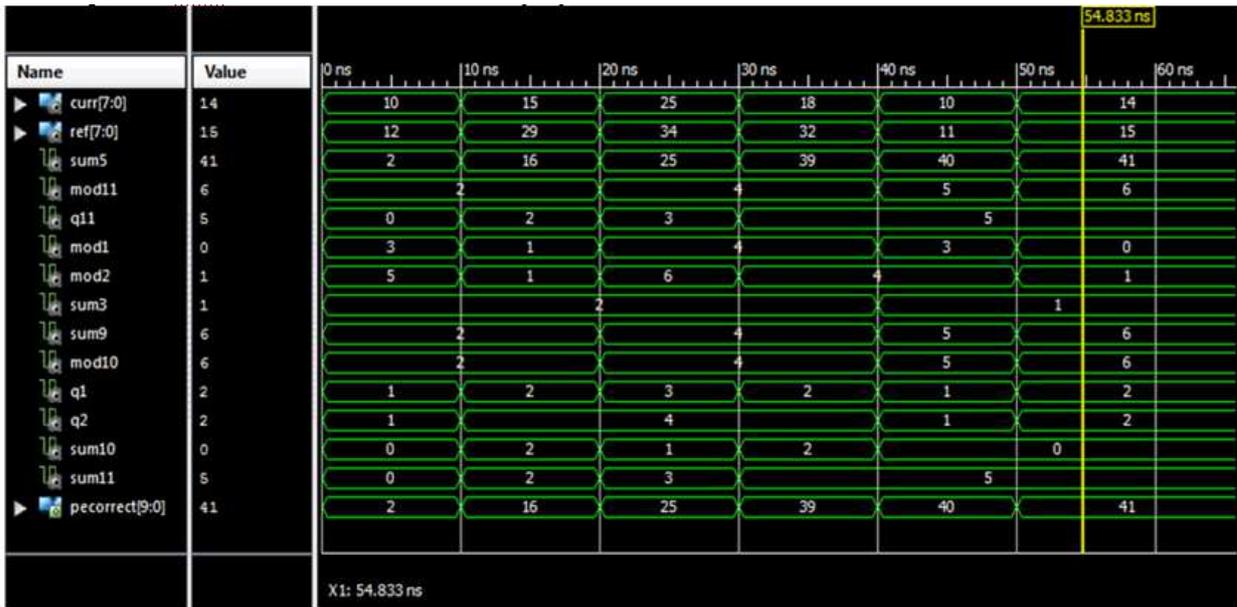


Fig 8: Wave form

If the values of $R_{PEi} \neq R_T$ and/or $Q_{PEi} \neq Q_T$, then the errors in a specific PE_i can be detected. The EDC output is then used to generate a 0/1 signal to indicate that the tested PE_i is error-free/errancy. Notably, the proposed BISDC design 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.

A. Timing Summary:

- Speed Grade: -4
- Minimum period: No path found
- Minimum input arrival time before clock: 27.607ns
- Maximum output required time after clock: 43.070ns
- Maximum combinational path delay: 50.621ns

Design of an Built In Self Detection & Correction (BISDC) Architecture for Motion Estimation Computing Arrays

B. Device Utilization Summary:

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Latches	7	1,920	1%	
Number of 4 input LUTs	342	1,920	17%	
Number of occupied Slices	201	960	20%	
Number of Slices containing only related logic	201	201	100%	
Number of Slices containing unrelated logic	0	201	0%	
Total Number of 4 input LUTs	373	1,920	19%	
Number used as logic	342			
Number used as a route-thru	31			
Number of bonded IOBs	24	66	36%	
Average Fanout of Non-Clock Nets	3.05			

VI. CONCLUSION

This project proposes BISDC architecture for self-detection and self-correction of errors of PEs in an ME. Based on the RQ code, a RQCG-based TCG design is developed to generate the corresponding test codes to detect errors and recover data. Performance evaluation reveals that the proposed BISDC architecture effectively achieves self-detection and self-correction capabilities with minimal area (LUT). The Functional-simulation has been successfully carried out with the results matching with expected ones. The design functional verification and Synthesis is done by using Xilinx-ISE 12.3 Version.

VII. 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: Visual, 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-testability 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 techniques 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 Circuits Syst., Jun. 2004, pp. 349–352.