Implementation of BISDC Architecture in MECA for Video Coding Applications

D. Rajitha Asst.Prof[1], K. Suresh Asso.Prof[2]
1, 2 (Department of Electronic Communication & Engineering, Chaitanya Engineering College, JNTUK, VSP-48

Abstract

This paper develops a built-in self-detection/correction (BISDC) architecture for motion estimation computing arrays (MECAs). Based on the error detection/correction concepts of biresidue codes, any single error in each processing element in an MECA can be effectively detected and corrected online using the proposed BISD and built-in self-correction circuits. Performance analysis and evaluation demonstrate that the proposed BISDC architecture performs well in error detection and correction with minor area overhead and timing penalty.

***Index Terms* -** Area overhead, built-in self-correction (BISC), built-in self-detection (BISD), motion estimation computing array (MECA).

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 [1], [2].

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 de-vice can be tested to guarantee high system reliability. Many DFT ap-proaches have been developed. These approaches can be divided into three categories: ad hoc (problem oriented), structured, and built-in self-test (BIST) [3], [4]. Among these techniques, BIST has an obvious advantage in that expensive test equipment is not needed and tests are low cost. Moreover, BIST can generate test simulations and analyze test responses without outside support, making tests and diagnoses of digital systems quick and effective. However, as the circuit complexity and density increases, the BIST approach must detect the presence of faults and specify their locations for subsequent repair. The extended techniques of BIST are built-in self-diagnosis [5] and built-in self-re-pair (BISR) [6]. Although BIST and BISR are utilized in many studies, most studies focused on memory testing. Nowadays, the computational complexity of modern video coding systems has increased; thus, effi-cient self-detection and self-correction techniques are needed to im-prove reliability.

Based on the concepts of BIST and biresidue codes, this paper presents a built-in self-detection/correction (BISDC) architecture that effectively self-detects and self-corrects PE errors in an MECA. Notably, any array-based computing structure, such as the discrete cosine transform (DCT), iterative logic array (ILA), and finite-impulse filter (FIR), is suitable for the proposed method to detect and correct errors based on biresidue codes.

II. ERROR DETECTION/CORRECTION CODES

The use of residue codes to detect error is a useful approach in com-puter arithmetic [7]. Residue codes are separable arithmetic codes that calculate a residue for data, and then apply this residue to data. For instance, we assume N denotes an integer, N1 and N2 represent data words, and A is the modulus. A separate residue code of interest is one in which N is coded as a pair (N, $|N|_A$) . Notably, $|N|_A$ is the residue of N modulo A. Error detection logic for operations is typi-cally derived using a separate residue code such that detection logic is simply and easily implemented. However, error correction cannot be performed effectively using residue codes. The arithmetic code, namely biresidue codes, can be supported to realize error detection and error correction.

The biresidue codes separate residue coding using two residue detectors with respect to two suitable moduli. Consider integer N coded as a triple (N, $|N|_A$, $|N|_B$), where A and B are two rela-tively prime integers. Let moduli $A = 2^a - 1$ and $2^b - 1$ such that GCD(a,b) = 1. The set of all single errors denoted by

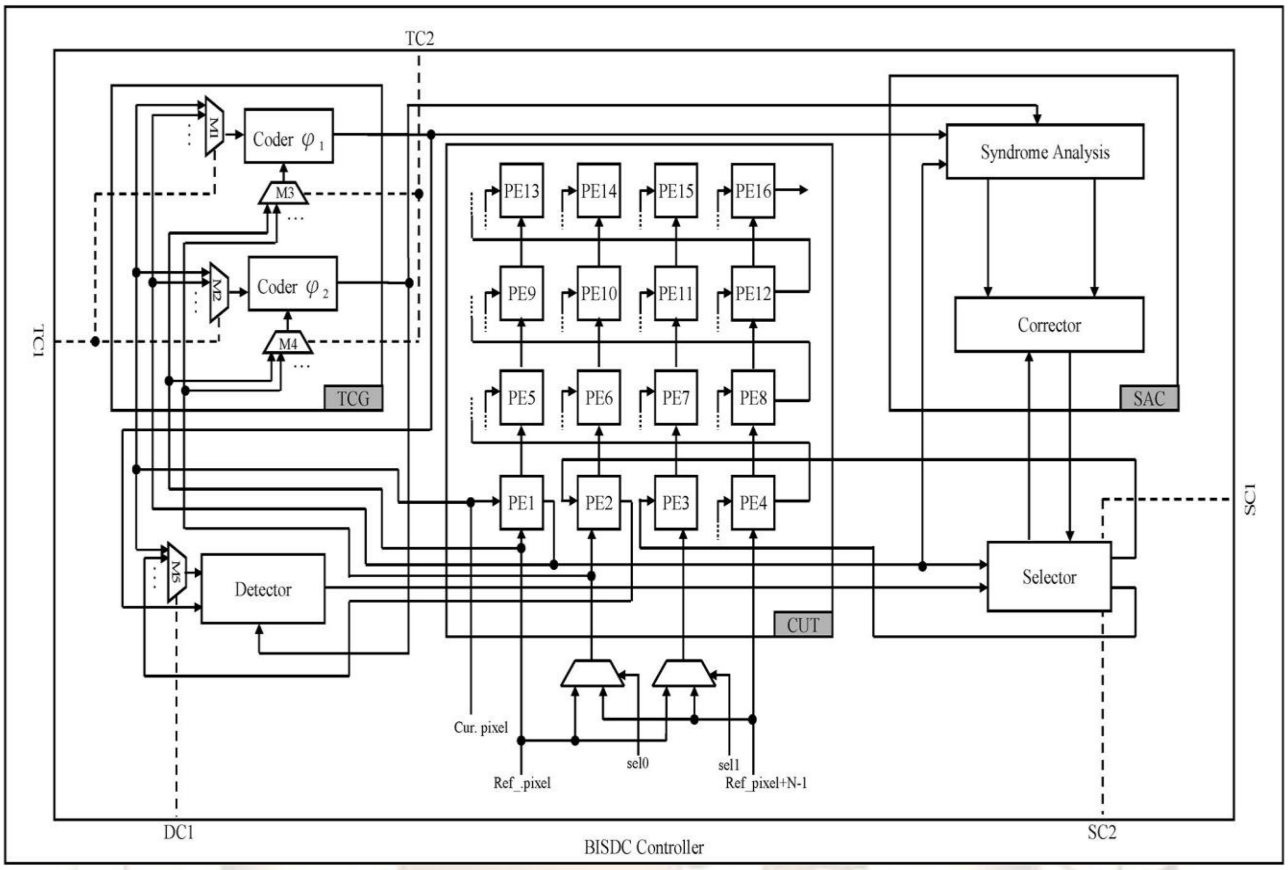


Fig. 1. Proposed MECA BISDC implementation.

$\{e=\pm 2^i$, i= 0, 1, 2, 3, ..., n-1$\}$ will have distinct syndromes with respect to A and B provided n is not greater than ab. Additionally, the triple (X, Y, Z) is considered a biresidue code word with respect to moduli A and B if and only if $Y=|X|_A$ and $Z=|X|_B$. Moreover, the syndrome for the triple (X, Y, Z) with respect to moduli A and B, denoted as S(X, Y, Z), is a pair (s_a, s_b) where $s_a= |X-Y|_A$ and $s_b= ||X-Z|_B$. Thus, a triple (X, Y, Z) of integers is a biresidue code word with respect to A moduli B and if and only if its syndrome S(X, Y, Z) with respect to moduli A and B equals zero ($s_a = 0$, $s_b = 0$). In other words, the error in any component is detected and lo-cated based on the form of its corresponding syndrome. In accordance with the error detection and error correction concepts in biresidue codes, this paper proposes a BISDC architecture to self-detect and self-correct PE errors in an MECA.

III. PROPOSED METHODOLOGY

To demonstrate the feasibility of the proposed BISDC architecture, this paper adopts the MECA as a CUT [8]. The MECA consists of many PEs connected into a 1-D or 2-D array for video encoding applications. Generally, a PE is made up of two adders (an 8-bit adder and 12-bit adder) and an accumulator. The PE in an MECA computes the absolute difference between one pixel of the search area and one pixel of the current macroblock. Thus, by utilizing PEs, the sum of absolute differences (SADs) shown in (1) between the current macroblock and each search position can be evaluated

$$\mathrm{SAD} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |c(i, j) - r(i, j)| \quad (1)$$

where c(i, j) and r(i, j) are the luminance pixel value of current pixel (Cur.pixel) and reference pixel (Ref.pixel), respectively. The macroblock size is N x N. The 2-D H.264/Advanced Video Coding (AVC) motion estimation architecture published in [9] is a clear example of MECA operations. The best motion position of a 4 x 4 block from the previous frame to the current frame can be captured easily using MECA operations in the video encoding system.

According to MECA characteristics, Fig.1 shows the corresponding BISDC implementation. Signals TC1 and TC2 are utilized to select datapaths from Cur.pixel and Ref.pixel, respectively. The output of a specific PE_i can be delivered to a detector for detecting errors using the DC1 signal. Moreover, the selector circuit is controlled by signals SC1 and SC2 that receive data from a specific PE_i, and then export these data to the next specific PE_{i+1} or syndrome analysis and corrector (SAC) for error correction. The self-detection and self-correction operations (Fig. 1) are simply described as follows.

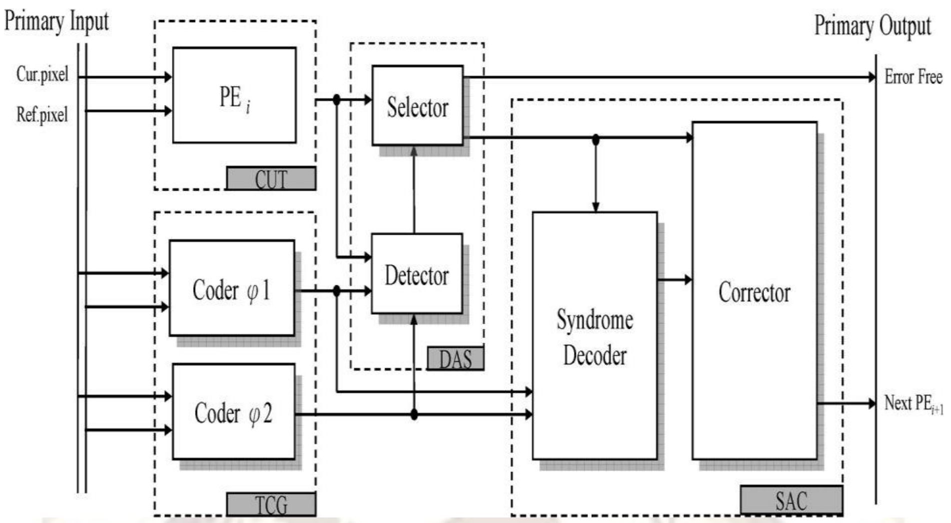


Fig. 2. Example of the self-detection/correction operations

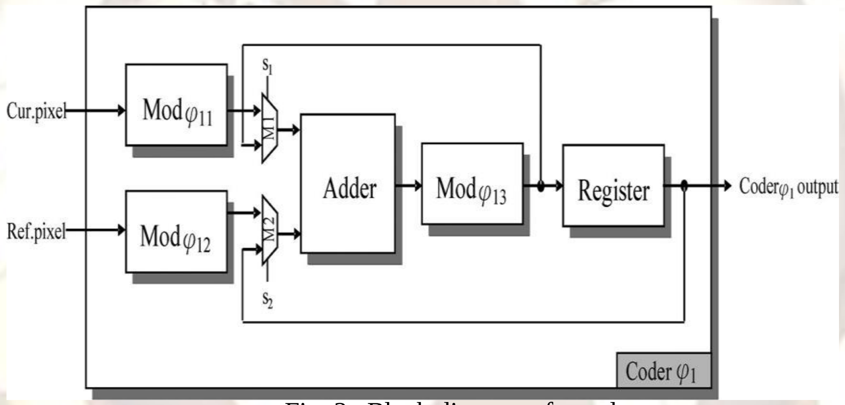


Fig. 3. Block diagram of a coder.

Selector circuit in DAS delivers the error signal to SAC for error correction. Finally, the error correction data from SAC, or error-free data from the selector circuit in DAS, are passed to the next specific PE_{i+1} for subsquent testing.

B. Fault Model

The PEs are important builiding blocks and are connected in a regular manner to construct an MECA. Generally, Pes are surrounded by set of adders and accumulators that determine how data flows through them. Thus, Pes can be considered the class of circuits called ILAs, whose testing assignment can be easily achieved using the fault model called as cell fault model (CFM) [10]. The use of the CFM is currently of considerable interest due to the rapid growth in the use of high-level synthesis and the parallel increasing complexity and density of Ics. Using the CFM allows tests to be independent of the adopted synthesis tool and vendor library. Arithmetic modules, like adders (the primary element in a PE), due to their regularity, are designed in a very dense configuration.

Moreover, the use of a relatively more comprehensive fault model, the single stuck –at (SSA) model, is required to cover actual failures in the interconnected databus between PEs. The SSA fault is a well-known structural fault model that assumes faults cause a line in the circuit to bahave as it work permenantly at logic “0” [stuck-at 0 (SA0)] or logic “1” [stuck-at 1 (SA1)]. The SSA fault in an MECA architecture can result in errors in computed SAD values. This paper refers to this as a distorted computational error; its magnitude is e = SAD’ – SAD , where SAD1 is the computed SAD value with an SSA fault.

C. BISDC Processes

Fig.2 shows an example of a specific PE, to describe explicitly the self-detection and self-correction of errors in an MECA using the proposed BISDC architecture. The TCG circuit uses two coders (coders φ1 and φ2) to generate test codes. The following definitions, based on the biresidue codes,

Definition 1:

$|N1 + N2|_{\varphi} = ||N1|_{\varphi} + |N2|_{\varphi}|_{\varphi}$. (2)

Definition 2: Let $N_j = n_1 + n_2 + n_3 + \ldots + n_j$, then

$|N_j|_{\varphi} = ||n_1|_{\varphi} + |n_2|_{\varphi} + |n_3|_{\varphi} + \ldots + |n_j|_{\varphi}|_{\varphi}$. (3)

Based on the definitions 1 and 2, the design of the coder φ1 (or coder φ2) circuit can be realized and shown in fig.3.

Fig.4 shows the timing chart for a specific PE_i in an MECA to describe the operation of coder φ1 circuit in the TCG. If data n_1 and n_2, obtained from Cur.pixel and Ref.pixel, are sent to the $Mod\varphi_{11}$ and $Mod\varphi_{12}$ circuits at the first clock, then values $|n_1|_{\varphi1}$ and $|n_2|_{\varphi1}$ can be treated at the second clock. When the third clock is triggered, the summation of modulus values, $\alpha = |n_1|_{\varphi1} + |n_2|_{\varphi1}$, is generated by the adder. Additionally, the following two data, n_3 and n_4, are simultaneously passed into the Modφ11 and $Mod\varphi_{12}$ circuits for modulo operations. The $Mod\varphi_{13}$ are applied to verify the feasibility of the two coders in the TCG.,

circuit executes the operation of $|\alpha|_{\varphi1} = ||n_1|_{\varphi1} + |n_2|_{\varphi1}|_{\varphi1}$, and then stores the computational results in the register at the fourth clock. Moreover, the summation of the modulus value $\beta = |n_3|_{\varphi1} + |n_3|_{\varphi1}$ can be captured by an adder $_{at}$ this time. Notably, the two selected signals S_1 and S_2 in multiplexers M_1 and M_2 are set to 0 to sum the modulus values at clocks 1 – 4. At the fifth clock, the next two data, n_5 and n_6, are transmitted to the coder φ1 circuit, and signals S_1 and S_2 are changed from 0 to 1 to deliver the values of $|\alpha|_{\varphi1}$ and $|\beta|_{\varphi1}$ to the adder for addition. Fig.4 clearly demonstrates that the regularity processes will continue after the fifth clock. Since a 4 x 4 macroblock in a specific PE_i of the MECA contains 16 pixels, the accumulated input data from coder φ_1 and coder φ_2 circuits in the TCG are exported to the DAC and SAC circuits for error detection and error correction, respectively, after 50clocks.

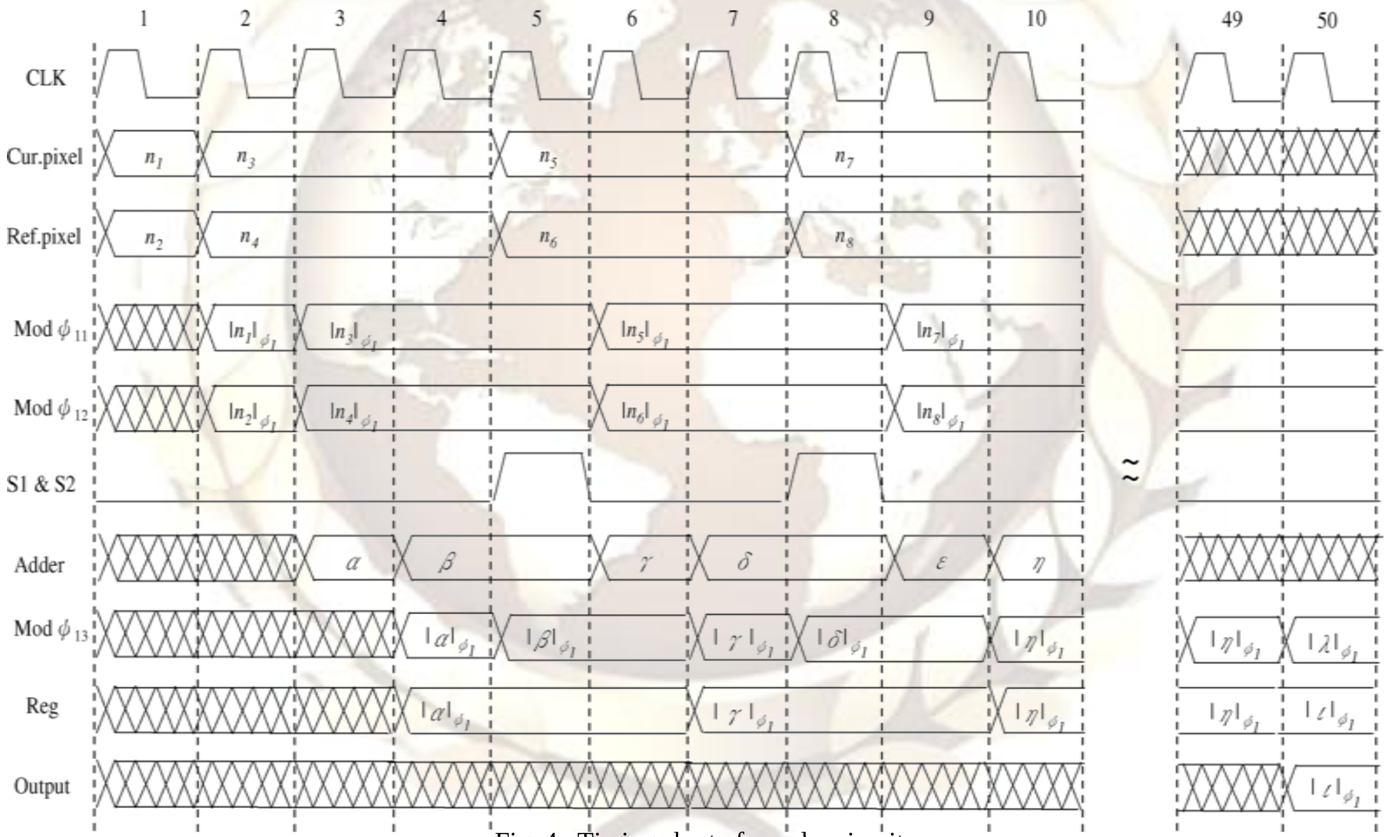


Fig. 4. Timing chart of a coder circuit

TABLE 1
SYNDROME CORRESPONDING TO ALL SINGLE BOT ERRORS

bit i	0	1	2	3	4	5	6	7	8	9	10	11
2^i syndrome $(s_{\varphi1}, s_{\varphi2})$	1,1	2,2	4,4	1,8	2,1	4,2	1,4	2,8	4,1	1,2	2,4	4,8
-2^i syndrome $(s_{\varphi1}, s_{\varphi2})$	6,14	5,13	3,11	6,7	5,14	3,13	6,11	5,7	3,14	6,13	5,11	3,7

1) Self-Detection Operation: The self-detection operation can be achieved using the DAS circuit. The detector circuit is utilized to com-pare the outputs between a specific PE_i and the TCG for determining whether an error has occurred (Fig. 2). A selector circuit in DAS is then enabled to place the error in the SAC circuit for error correction or to export the error-free results to the output directly.

A mathematical statement is presented herein to verify the self-de-tection operation. According to Definition 2, the residue of the N_j modulo φ is $|N_j|_\varphi = ||n_1|_\varphi + |n_2|_\varphi + |n_3|_\varphi + \ldots + |n_j|_\varphi|_\varphi$ when the specific PE_i has j pixels. Moreover, based on the biresidue codes theorem, a triple (N_j, X, Y) with respect to moduli φ1 and φ2 is given by

$$X = |N_j|_{\varphi 1} = ||n_1|_{\varphi 1} + |n_2|_{\varphi 1} + |n_3|_{\varphi 1} + \ldots + |n_j|_{\varphi 1}|_{\varphi 1} \quad (4)$$
$$Y = |N_j|_{\varphi 2} = ||n_1|_{\varphi 2} + |n_2|_{\varphi 2} + |n_3|_{\varphi 2} + \ldots + |n_j|_{\varphi 2}|_{\varphi 2} \quad (5)$$

Thus, the syndrome can be represented by the pair ($s_{\varphi 1} = |N_j - X|_{\varphi 1}, s_{\varphi 2} = |N_j - X|_{\varphi 2}$). Additionally, we assume the pixel value is adjusted to $|N'_j| = N_j + e$ when an error bit is present in the specific PE_i. According to Definition 2, the residue of N'_j modulo φ1 and φ2 is given by

$$|N'_j|_{\varphi 1} = |N_j + e|_{\varphi 1} = ||N_j|_{\varphi 1} + |e|_{\varphi 1}|_{\varphi 1} \quad (6)$$
$$|N'_j|_{\varphi 2} = |N_j + e|_{\varphi 2} = ||N_j|_{\varphi 2} + |e|_{\varphi 2}|_{\varphi 2} \quad (7)$$

Thus, the single error bit in the specified PE_i can be detected if and only if (4) ≠ (6) and/or (5) ≠ (7).

2) Self-Correction Operation: In the self-correction operation, the SAC circuit plays an important role in correcting errors in a specific PE_i. The SAC circuit (Fig. 2) receives data from the TCG and DAS circuits to start error correction. The syndrome decoder and corrector circuits in the SAC are employed to diagnose single error and further correct error signal, respectively. In other words, the syndrome decoder in the SAC generates syndromes $s_{\varphi 1}$and $s_{\varphi 2}$ by adopting the error correction concepts of biresidue codes. Table I and Fig. 5 show the syndromes corresponding to all cases of single bit error and the corrector circuit, respectively, i.e., any single bit error of a specific PE_i of the MECA can be obtained by comparing the syndrome ($s_{\varphi 1}, s_{\varphi 2}$) with Table I, and then the bit error is corrected using the circuit in Fig. 5.

For instance, based on (4)–(7), syndromes $s_{\varphi 1}$and $s_{\varphi 2}$ can be ex-pressed as

$$(s_{\varphi 1}, s_{\varphi 2}) = (|N'_j - X|_{\varphi 1}, |N'_j - Y|_{\varphi 2}) = (|e|_{\varphi 1}, |e|_{\varphi 2}). \quad (8)$$

Here, the specific PE_i is error-free when the syndrome ($s_{\varphi 1}, s_{\varphi 2}$) = (0, 0). However, a single error bit can be detected when the syndrome ($s_{\varphi 1}, s_{\varphi 2}$) ≠ (0, 0), and the error bit can be located and corrected using the syndrome listed in Table I and the circuit shown in Fig. 5.

IV. PERFORMANCE EVALUATION

The proposed BISDC architecture was generated using VHDL and synthesized using the Synopsys Design Compiler with TSMC 0.18 m 1P6M CMOS technology. The MECA was selected to act as the CUT to demonstrate the effectiveness of the proposed BISD and built-in self-correction (BISC) procedures. The area overhead, timing penalty, and throughput are also utilized to demonstrate the good performance of the proposed BISDC architecture.

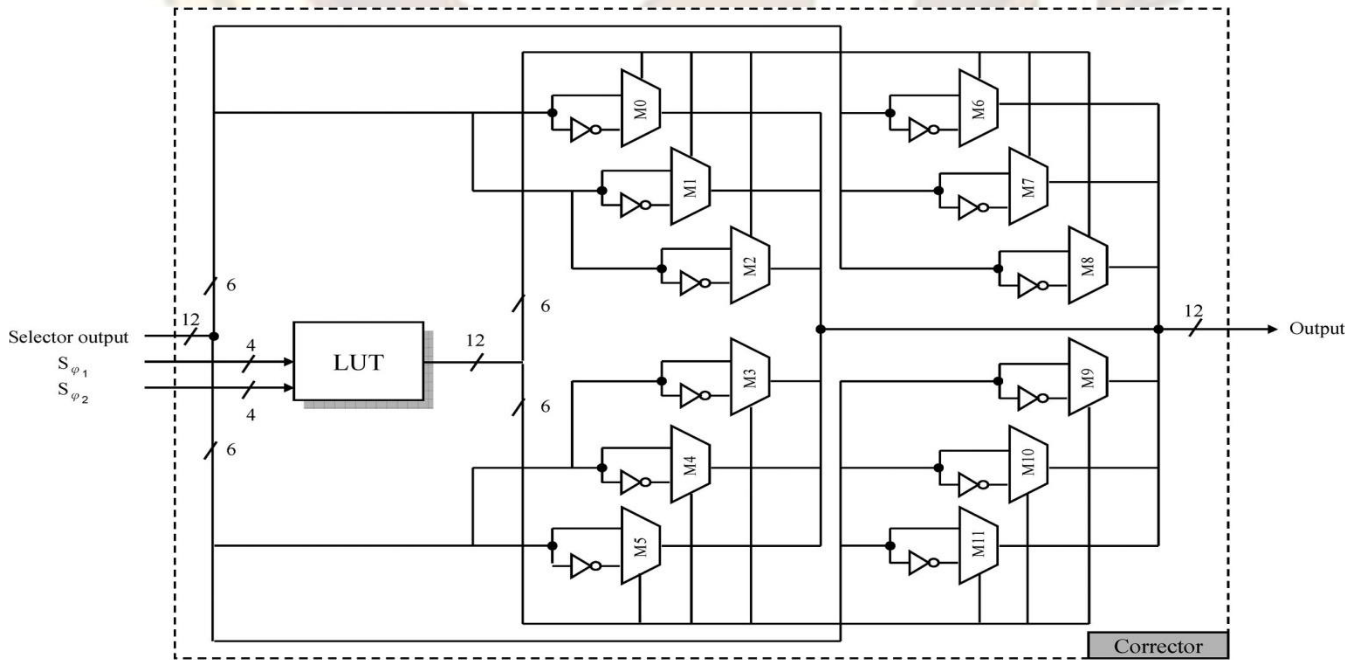


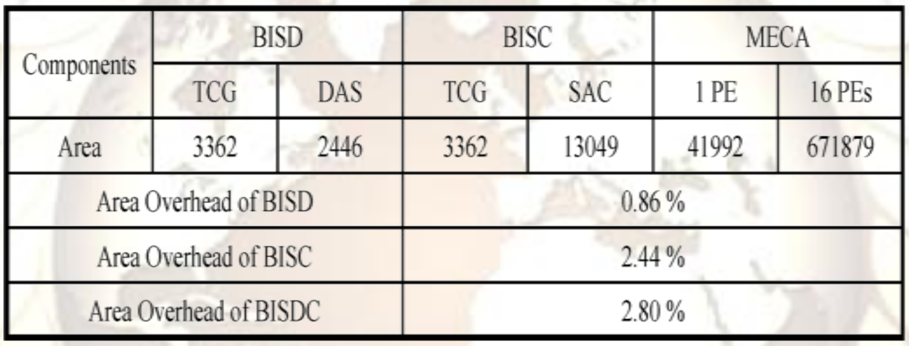
Fig. 5. Circuit of corrector in SAC.

Table II summarizes synthesis results of area estimation of the circuit design. The area is estimated based on the number of cells. Considering 16 PEs in an MECA, the area overhead introduced is given by

$$\text{AREAOVERHEAD}=\frac{AREA\,\text{BISD}+AREABISC-AREATCG}{AREAMECA} \quad (9)$$

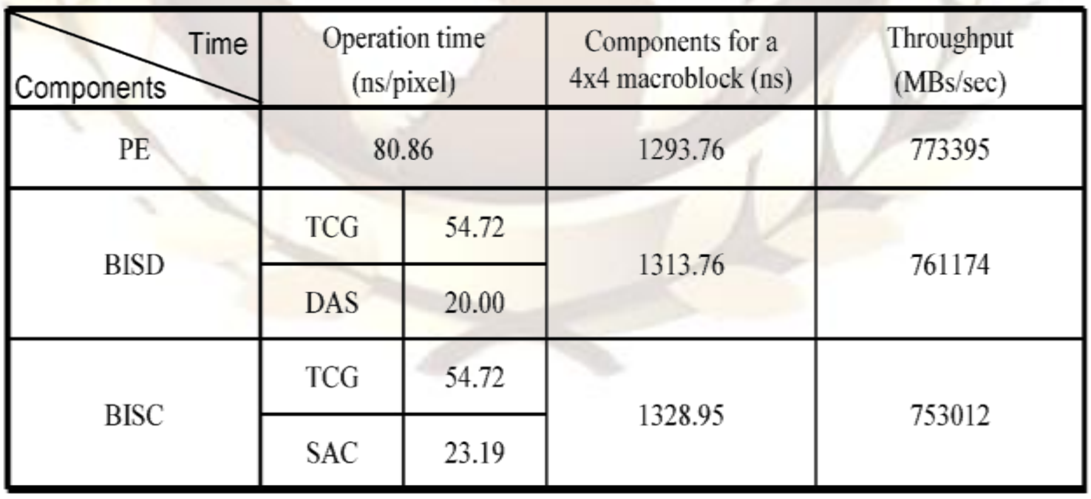
The self-detection and self-correction functionalities can be achieved using the proposed BISD and BISC with about 0.86% and 2.44% area overhead, respectively (Table II). Based on (9), Table II also shows the total area overhead of the proposed BISDC architecture, only 2.80%, which is reasonable for designing a circuit for testing. The timing penalty and throughput must also be demonstrated to verify the flexibility of the proposed BISDC architecture. Table III shows the operating time estimation of a specific and each component in the proposed BISDC architecture. Each in an MECA is tested in succession; thus, a 4 2 4 macroblock (with 16 pixels) is used to estimate timing penalty. A specific requires 1313.76 and 1328.95 ns for self-detection and self-correction operations, respectively (Table III). Notably, the proposed BISDC architecture is a noncurrent online testing scheme for error detection/correction of faulty PEs. In other words, an error detection/correction operation for each faulty PE in an MECA is executed in sequence. Thus, if the pro-posed BISDC architecture is embedded into the MECA for testing, the entire timing penalty is equal to that for testing a single PE, i.e., about 1.55% and 2.72% (Table III) for BISD and BISC, respectively.

TABLE II
AREA OVERHEAD ESTIMATION



Components	BISD		BISC		MECA	
	TCG	DAS	TCG	SAC	1 PE	16 PEs
Area	3362	2446	3362	13049	41992	671879
Area Overhead of BISD			0.86 %			
Area Overhead of BISC			2.44 %			
Area Overhead of BISDC			2.80 %			

TABLE III
TIMING PENALTY AND THROUGHPUT ESTIMATION



Time / Components	Operation time (ns/pixel)		Components for a 4x4 macroblock (ns)	Throughput (MBs/sec)
PE	80.86		1293.76	773395
BISD	TCG	54.72	1313.76	761174
	DAS	20.00		
BISC	TCG	54.72	1328.95	753012
	SAC	23.19		

The test processes for the TCG circuit and tested are operated in parallel (Fig. 2). Preparation time of the TCG circuit is faster than that of the one-pixel operation of the tested (Table III). Therefore, the operation time of the TCG circuit can be neglected because the timing penalty of the TCG is covered by the pixel operation. Table III also shows the total number of accessing macroblocks per second. Obviously, when the MECA is combined with the proposed BISDC architecture, complexity and throughput will be increased and decreased, respectively, i.e., complexity increases to 2.6% when an error occurs. Although the throughput of the MECA with the proposed BISDC architecture is lower than that of the MECA without the BISDC architecture, fewer data access is indicative of reduced demand on input/output bandwidth. Generally, a low memory bandwidth is desirable in the progressive high-definition television (HDTV) video format.

V. RESULTS

Simulation results of Absolute Difference

Absolute Difference as a two inputs a, b i.e. current and reference pixels each of 8-bit length and one output result also 8-bit length. The behavioral simulation waveform for the Absolute Difference. The two inputs with 8-bit length are ‘a’ (current Pixels) and ‘b’ (reference pixels) and 8-bit output result.

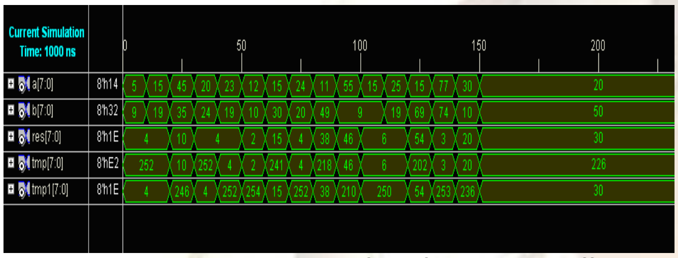


Fig 6: Simulation Waveform for Absolute difference

Simulation Results of Compressor Module

Compressor Module as a two inputs a, b each of 8-bit length and one output c also 8-bit length. The behavioral simulation waveform for the Compressor. The two inputs with 8-bit length are ‘a’ (current Pixels) and ‘b’ (reference pixels) and 8-bit output.

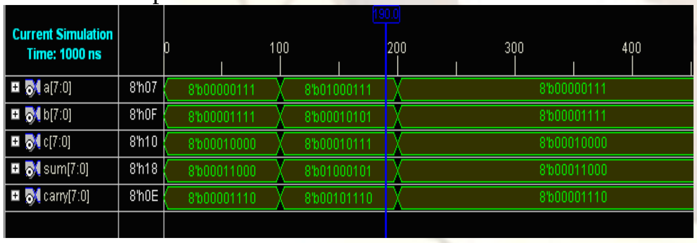


Fig 7:Simulation Waveform of compressor

Simulation waveforms of Processing Element module

Processing Element as a three inputs create_error, current pixel, reference pixel each of 8-bit and output is a sad_dash as a 12-bit data. The input of PE is a current pixel and reference pixels. The behavioral simulation waveform for the Processing Element. The two inputs are 8-bit length are ‘a’ (current Pixels) and ‘b’ (reference pixels) input and 12-bit output.

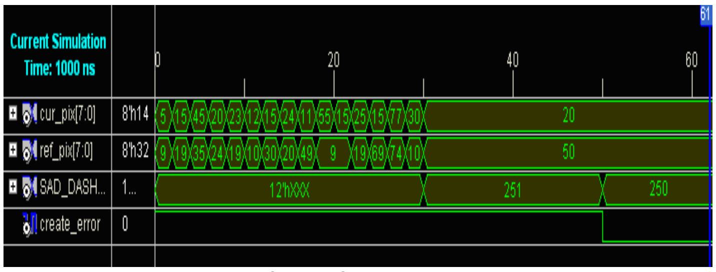


Fig 8:Simulation Waveform of Processing Element

Simulation Results of Modulus code

Modulus code as a two inputs i.e. dividend, divider each of 12-bit length and it has one output it as a modulus 4 –bit of length. The behavioral simulation waveform for the Modulus Division code as a two inputs i.e. dividend, divider each of 12-bit length and it has one output it as a modulus 4 –bit of length.

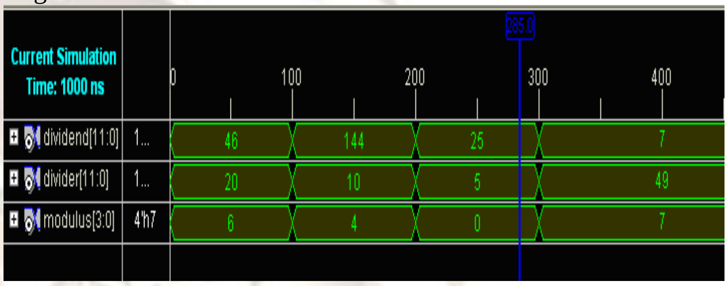


Fig 9:Simulation Waveform of Modulus

Simulation Results of Coder module :

Coder as a three inputs clk, cur_pix, ref_pix and each of 8-bit length and output consists two coders i.e. out_a, out_b it consists of 4-bit length. The input of a coder is clk, current and reference pixels. The behavioral simulation waveform for the Coder as a three inputs clk, cur_pix, ref_pix and each of 8-bit length and output consists two coders i.e. out_a, out_b it consists of 4-bit length. The input of a coder is clk, current and reference pixels.

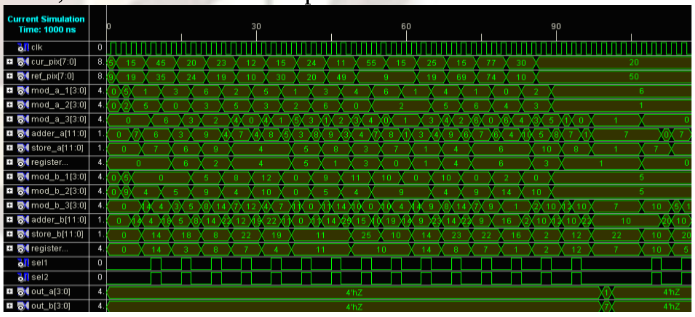


Fig10: Simulation Waveform of Coder

Simulation Results of Selector Module:

Selector takes the output of the PE as an input. Another input to the selector is the output of the detector. It has three inputs clk, select, PE_out. And the output is select_out, error_free each of 12-bit length. The behavioral simulation waveform for the Selector takes the output of the PE as an input.

Another input to the selector is the output of the detector. It has three inputs clk, select, PE_out. And the output is select_out, error_free each of 12-bit length.

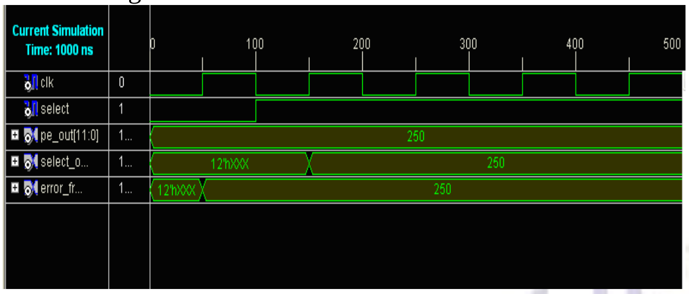


Fig 11:Simulation Waveform of Selector

Simulation results of Corrector Module:

Input to the Corrector module is the output of the selector module which is SAD that needs to be corrected. It as three inputs select_out, sphi_1, sphi_2 and output as a corr_out as a12-bit length. The behavioral simulation waveform for the Input to the Corrector module is the output of the selector module which is SAD that needs to be corrected. It as three inputs select_out, sphi_1, sphi_2 and output as a corr_out as a12-bit length.

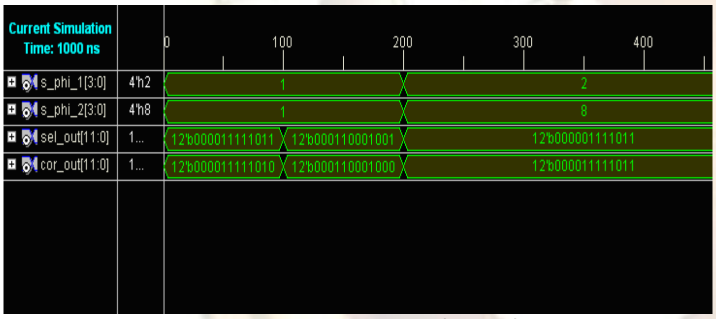


Fig12:simulation waveform of Corrector

Simulation results of Top Module

The proposed design is developed in a top down design methodology that the code is a mixed version of both behavioral and structural. The proposed Architecture consists of basic modules like Absolute Difference, Compressor, Processing Element, Modulus Division, Coder, Selector and Corrector modules. The behavioral simulation results for Top Module i.e., BISDC Architecture for MECA with inputs of clk, cur_pixel[7:0], ref_pixel[7:0], Create_error and outputs with error, with_out_error are given in Fig 7.16. This waveform contains signals like N (sum of total number of current pixels and reference pixels without error), N_dash_error (sum of total number of current pixels and reference pixels with error), syndrome_7 [3:0], syndrome_15 [3:0].

Current pixels: 5,15,45,20,23,12,15,24,11,55,15,25,15,77,30,20
Reference pixels: 9,19,35,24,19,10,30,20,49,9,9,19,69,74,10,50

N=n1+n2+n3+……………………………………… ………..n32

N = 5+9+15+19+45+35+20+24+23+19+12+10+15+30+24+20+11+49+55+9+15+9+25+19+15+69+77+74+30+10+20+30
N = 862
Sum of Absolute Difference (SAD) = 4+4+10+4+4+2+15+4+38+46+6+6+54+3+20+30
Sum of Absolute Difference = 250
e=SAD’-SAD=1 (error should create)
Where SAD’=Sum of absolute difference of with error
SAD’=251
N_dash = N+1 = 862+1=863.

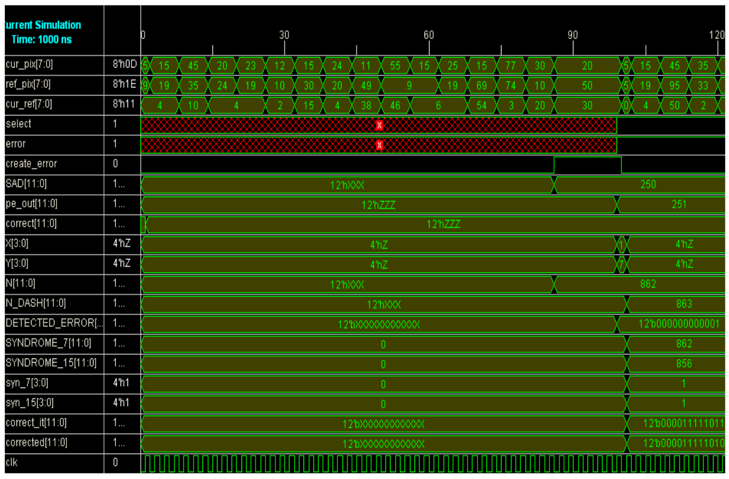


Fig 13 : Simulation waveform of Top module

VI. CONCLUSION

This paper proposes a BISDC architecture for self-detection and self-correction of errors of PEs in an MECA. Based on the error de-tection/correction concepts of biresidue codes, this paper presents the corresponding definitions used in designing the BISD and BISC cir-cuits to achieve self-detection and self-correction operations.

Perfor-mance evaluation reveals that the proposed BISDC architecture effec-tively achieves self-detection and self-correction capabilities with min-imal area overhead and a small timing penalty.

REFERENCES

[1] Z. L. He, C. Y. Tsui, K. K. Chan, and M. L. Liou, “Low-power VLSI design for motion estimation using adaptive pixel truncation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 5, pp. 669–678, Aug. 2000.

[2] C. G. Peng, D. S. Yu, X. X. Cao, and S. M. Sheng, “Efficient VLSI de-sign and implementation of integer motion estimation for H.264 SDTV encoder,” in *Proc. IEEE Int. Conf. Solid-State Integr. Circuits*, 2006,

pp. 2019–2021.

[3] P. Gallagher, V. Chickermane, S. Gregor, and T. S. Pierre, "A building block BIST methodology for SOC designs: A case study," in *Proc. Int. Test Conf.*, Oct. 2001, pp. 111–120.

[4] 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 Des. Autom. Test*, Apr. 2007, pp. 25–27.

[5] J. C. Yeh, K. L. Cheng, Y. F. Chou, and C. W. Wu, "Flash memory testing and built-in self-diagnosis with march-like test algorithms," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 6, pp. 1101–1113, Jun. 2007.

[6] X. Xiong, Y. L. Wu, and W. B. Jone, "Reliability analysis of self-re-pairable MEMS accelerometer," in *Proc. IEEE Int. Symp., Defect Fault Tolerance VLSI Syst.*, Oct. 2006, pp. 236–244.

[7] R. J. Higgs and J. F. Humphreys, "Two-error-location for quadratic residue codes," *Proc. Inst. Electr. Eng. Commun.*, vol. 149, no. 3, pp. 129–131, Jun. 2002.

[8] J. C. Tuan, T. S. Chang, and C. W. Jen, "On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 1, pp. 61–72, Jan. 2002.

[9] C. Wei and M. Z. Gang, "A novel VLSI architecture for VBSME in MPEG-4 AVC/H.264," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2005, pp. 1794–1797.

[10] J. F. Li and C. C. Hsu, "Efficient testing methodologies for conditional sum adders," in *Proc. Asian Test Symp.*, 2004, pp. 319–324.