A Novel Fault Recovery Scheme for Motion Analysis Testing Applications

S.Vasanth Vigneshwaran[1], R.S.Venkatesan[2]

[1]PG Part Time Student, ECE Department, Anna University, Chennai, Tamilnadu, India
[2] Assistant Professor, ECE Department, Kamaraj College Of Engineering and Technology, Virudhunagar, Tamilnadu, India

ABSTRACT

This paper develops a novel Fault Recovery(FR) architecture for Motion Analysis Computing Arrays (MACA). Any single fault in each Processing Element (PE) in an MACA can be effectively detected and corrected using the concept of Dual-Remnant codes i.e., Remnant and Proportionate (RP) code. A Good Example is the H.264 video compression standard, also known as MPEG-4 Advanced Video Coding application. It uses a context-based adaptive method to speed up the multiple reference frames Motion Analysis by avoiding unnecessary reference frames computation. A large PE array accelerates the computation speed especially in High Resolution devices such as HDTV(High Definition Television).The Visual Quality and Peak Signal-to-Noise Ratio (PSNR) at a given bit rate are influenced if a fault occurred in MA process.

KEYWORDS :*Motion Analysis, Fault Recovery, Remnant and Proportionate Code, Processing lement.*

I. INTRODUCTION

Improved advancements in semiconductors, digital signal processing, and communication technologies have made multimedia applications more flexible and reliable. A good example is the H.264 or MPEG-4 Part 10 Advanced Video Coding, which is the next generation video compression [1],[2]. that is necessary for a wide range of applications to reduce the total data amount required for transmitting or storing video data. Among the coding trends, a MA is of high importance in exploiting the temporal redundancy between subsequent frames that provides the less computation time for coding. Moreover, while performing up to enormous computations encountered in the entire coding system, a MA is widely regarded as the most computationally intensive of a video coding system [3]. A MA generally consists of PEs with a size of NxN. Thus, increasing the speed of manipulation towards a high dimension of PE array, particularly in devices having more resolution factor with a large as N=4 for HDTV(High Definition Television) [4] search range. Also, the video quality and peak signal-to-noise ratio (PSNR) are influenced for a given bit rate if a fault obtained in MA process. A testable design is thus increasingly important to ensure the reliability of numerous PEs in a MA. Moreover, although the advance of VLSI technologies facilitate the integration of a large number of PEs of a MA into every chip, the logic-per-pin ratio is consistently increased, thereby slightly decreasing the logic testing efficiency on the chip. For the commercial purpose, it is mandatory for the MA to enhance Design For Testability (DFT) [5]-[7]. DFT concentrates on improving the usage of testing the devices, thus makes the system highly reliable. DFT techniques depend on circuit re-configuration during testing to enhance the features of testable nature. Hence Design For Testability methods improves the testability design of circuits [8]-[10]. Due to recent trends in micron technologies and increasing complexity of electronic systems and circuits, the Built-In Self-Test (BIST) schemes have supremely become necessary in this modern digital universe[11]-[14]. The rest of this paper is organized as follows. Section 2 describes the overall Fault Recovery system.Section 3 & 5 then describes the various modules and numerical example consideration of Fault Recovery process . Next, Section 4 generalizes the methodology in designing of Fault Recovery process. Section 6 & 7 formulates the Simulation set up and its results Section 8 evaluates the results and its discussion to demonstrate the feasibility of the proposed Fault Recovery architecture for MA testing applications. Conclusions are finally drawn in Section 9.

II. FAULT RECOVERY DESIGN

The conceptual view of the proposed Fault Recovery scheme, which comprises two major circuit designs, i.e. Fault Detection Circuit (FDC) and data recovery circuit (DRC), to identify faults and recover the corresponding data in a specific CUT. The test code generator (TCG) in Fig. utilizes the concepts of RP code to generate the corresponding test codes for fault identification and data recovery. In other words, the test codes from

TCG and the primary output from CUT are delivered to FDC to determine whether the circuit under test has faults. DRC is in charge of recovering data from TCG. Additionally, a selector is enabled to export fault-free data or data recovery results[15],[16]. Importantly, an array-based computing structure, such as MA, discrete cosine transform (DCT), iterative logic array (ILA), and finite impulse filter (FIR), is feasible for the proposed Fault Recovery scheme to identify faults and recover the corresponding data. In our proposed circuit the output will be gating in second clock cycle not a 22th clock cycle, because we change the RP block structure. Also the proposed Fault Recovery design for MA testing can identify faults and recover data with an acceptable area and time limit. Importantly, the proposed Fault Recovery design performs satisfactorily in terms of throughput and reliability for MA testing applications.

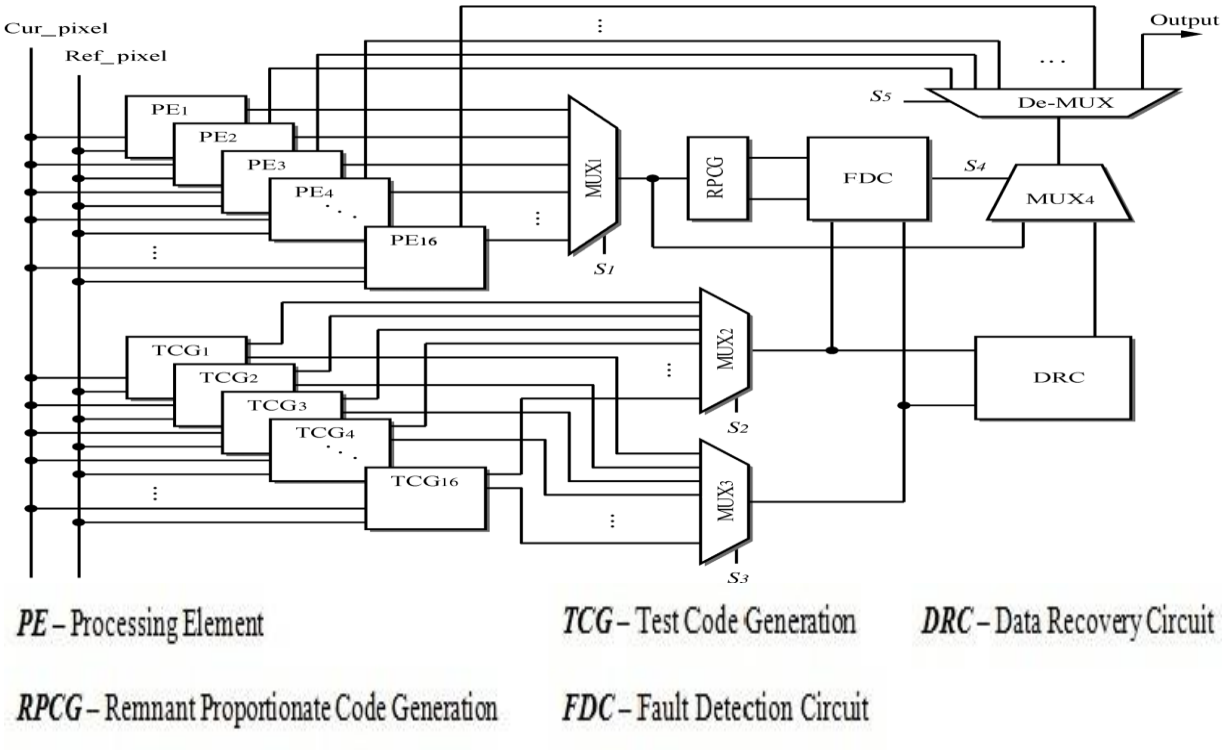

Figure 1. Design Layout of Fault Recovery Scheme

III. MODULE DESCRIPTION

3.1. PROCESSING ELEMENT

A MA (Motion Analysis) 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

3.2. SUM OF ABSOLUTE DIFFERENCE TREE

We propose a 2-D intra-level architecture called the Propagate Partial SAD. This Architecture is composed of PE arrays with a 1-D adder tree in the vertical direction. Current pixels are stored in each PE, and two sets of continuous reference pixels in a row are broadcasted to PE arrays at the same time. In each PE array of a Adder tree, harmonics are identified and added by a adder tree to generate single row SAD. The row SADs are accumulated and propagated with propagation registers in the vertical direction The reference data of searching candidates in the even and odd columns are inputted by Ref. Pixel 0 and Ref Pixel 1. Then the SAD of the initial search candidate in the zeroth column is generated, and the SADs of the other searching candidates are sequentially generated in the following cycles. When computing the last searching candidates in each column, the reference data of searching candidates in the next columns begin to be inputted by another input reference. while navigating in partial SAD, by sending reference pixel rows and also partial row SADs in the vertical scale direction, it gives the usage of lesser reference registers and a minimum critical path.

3.2.1. SUM OF ABSOLUTE DIFFERENCE VALUE CALCULATION

By utilizing PEs, SAD shown in as follows, in a macro block with size N X N of can be evaluated

$$\mathrm{SAD} = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} |X_{ij} - Y_{ij}|$$
$$= \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} |(q_{xij}\cdot m + r_{xij}) - (q_{yij}\cdot m + r_{yij})|$$

Where $r_{xi,j}$,$q_{xi,j}$ and $r_{yi,j}$,$q_{yi,j}$ denote the corresponding RP code of $X_{i,j}$, $Y_{i,j}$ and modulo M. Importantly, and represent the luminance pixel value of Cur_pixel and Ref_pixel subsequently.

3.3. REMNANT PROPORTIONATE CODE GENERATION ALGORITHM

In this RPCG Algorithm Remnant code is generally separable arithmetic codes by estimating a remnant for data and appending it to data [17],[18]. Fault detection logic for operations is typically derived by a separate remnant code, making the detection logic is simple and easily implemented. Fault detection logic for operations is typically derived using a separate remnant code such that detection logic is simply and easily implemented. However, only a bit fault can be detected based on the remnant code. Additionally, a fault can't be recovered effectively by using the remnant codes. Therefore, this work presents a proportionate code, which is derived from the remnant code; to assist the remnant code in rectifying multiple faults[19].The corresponding circuit design of the RPCG is easily realized by using the simple adders (ADDs).Namely, the RP code can be generated with a low complexity and little hardware cost[20].

3.4. TEST CODE GENERATION

TCG is an important component of the proposed Fault Recovery Design. Notably, TCG design is based on the ability of the RPCG Circuit to generate corresponding test codes in order to identify faults and recover data.

3.5. FAULT DETECTION CIRCUIT

In this module indicates that the operations of fault detection in a specific PE_i is achieved by using FDC, which is utilized to compare the outputs between TCG and in order to determine whether faults have occurred. The FDC output is then used to generate a 0/1 signal to indicate that the tested PE_i is fault-free/faulty. Using XOR operation can be identify the fault if any variation in terms of remnant and proportionate value. Because a fault only affects the logic in the fan-out cone from the fault region. Concurrent fault simulation exploits this fact and simulates only the differential parts of the whole circuit Concurrent fault simulation is essentially an event-driven simulation with the fault-free circuit and faulty circuits simulated altogether.

3.6. DATA RECOVERY CIRCUIT

In this module will be generate fault free output by proportionate multiply with constant value and add with proportionate code. During data recovery, the circuit DRC plays a significant role in recovering RP code from TCG.

IV. METHODOLOGY

Coding approaches such as Parity code, Berger code, and Remnant code is considered only to identify circuit faults. Remnant code R is a separable arithmetic codes by estimating a remnant for data and appending it to data. i.e., $R = |X|_m$ .Binary Data X is coded as a pair (X,R) and modulus $m = 2^w - 1$, w is word length. Proportionate code P =X/m is derived from the Remnant code to identify and recover multiple faults. To simplify the complexity of circuit design, the implementation is carried out using the simple Adders (ADDs).

V. NUMERICAL EXAMPLE

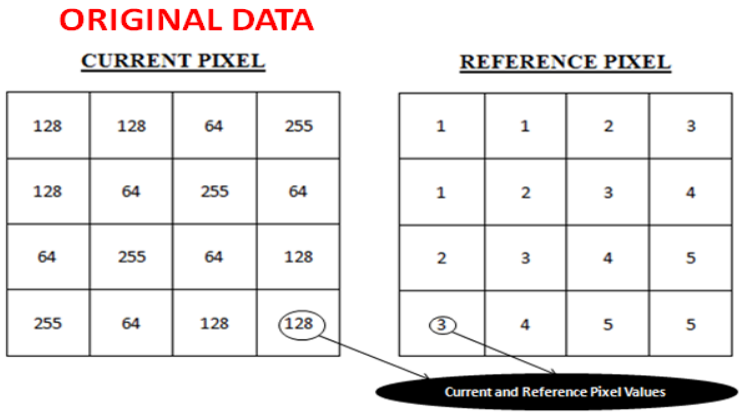

Table 1. Accumulation of Original data having pixel value of 128 in the (1,1) position of a 4x4 Current Macro Block

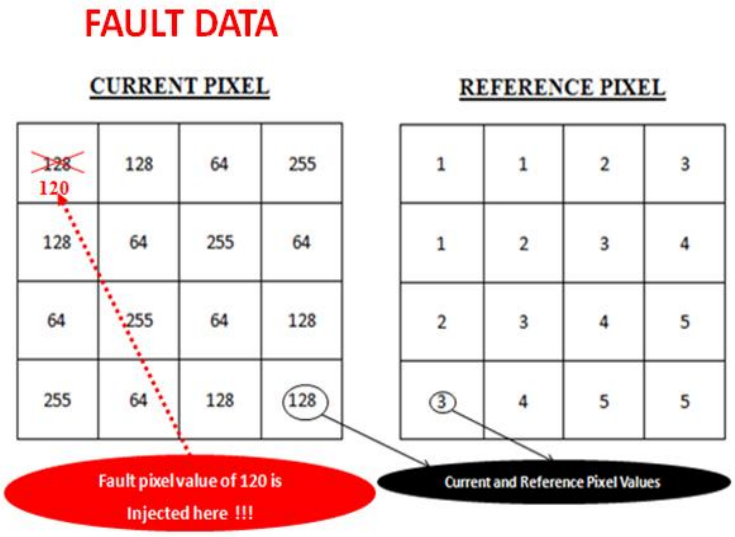

Table 2. Injection of Fault data having pixel value of 120 in the (1,1) position of a 4x4 Current Macro Block

VI. SIMULATION SET UP

6.1. CREATING THE WORKING LIBRARY

ModelSim is a verification and simulation tool for VHDL, Verilog, System Verilog, and mixed language designs.In ModelSim, all designs are compiled into a library. Typically start a new simulation in ModelSim by creating a working library called "work". "Work" is the library name used by the compiler as the default destination for compiled design units.

6.2. COMPILING YOUR DESIGN

After creating the working library, compile the design units into it. The Model Simlibrary format is compatible across all supported platforms. Simulate the design on any platform without having to recompile the design. Loading the Simulator with the Design and Running the Simulation. With the design compiled, load the simulator with respective design by invoking the simulator on a top-level module (Verilog) or a configuration or entity/architecture pair(VHDL).Assuming the design loads successfully, the simulation time is set to zero, and enter a run command to begin simulation.

6.3. DEBUGGING RESULTS

If we don't get the results as we expect, we can use Modelsim robust debugging environment to track down the cause of the problem.

VII. SIMULATION RESULTS

Figure 2. Simulation Result of Processing Element Circuit obtaining Absolute Differences from the Current and Reference pixels.

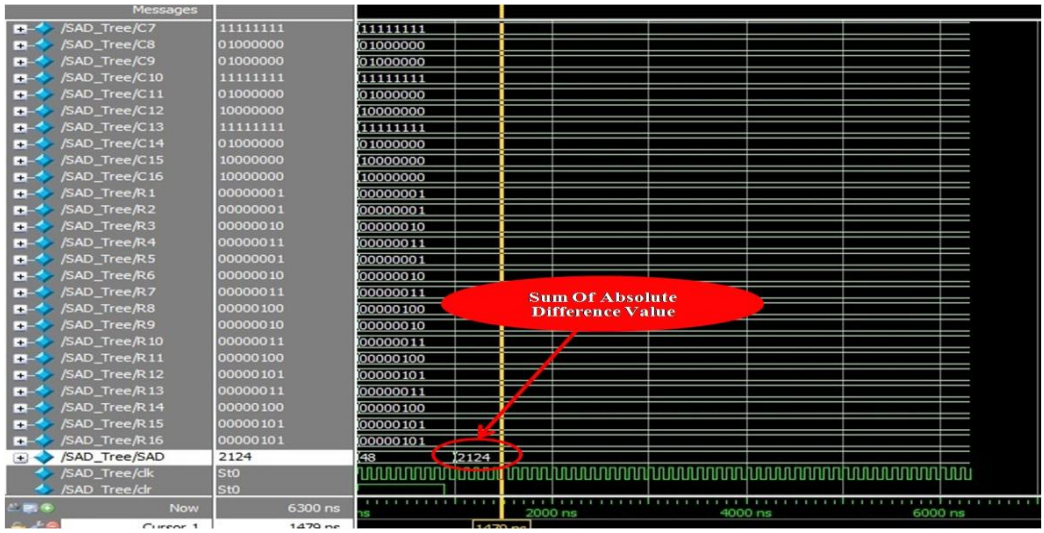

Figure 3. Simulation Result of SAD Generation Circuit obtaining its value from a suitable Candidate Block for a specific PE

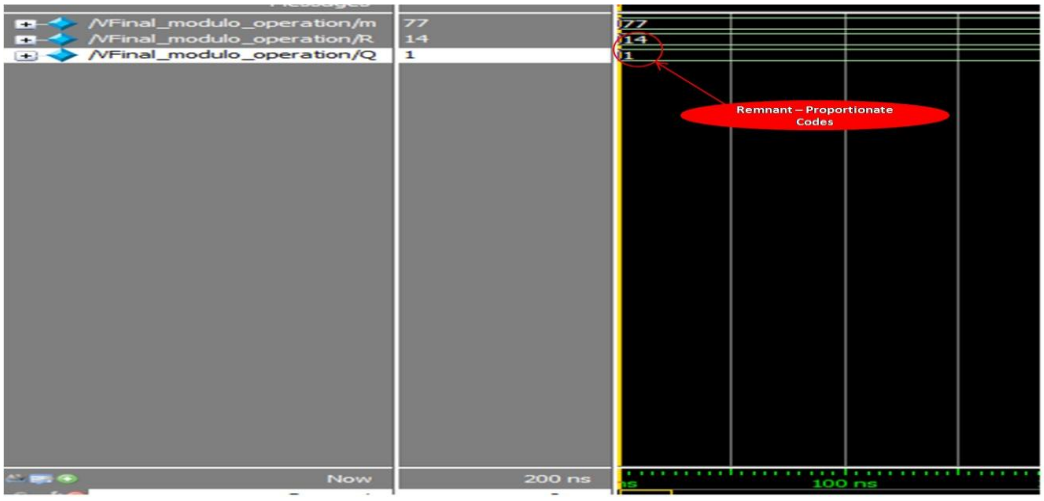

Figure 4. Simulation Result of RPCG Circuit obtaining its value from lowest distortion SAD value

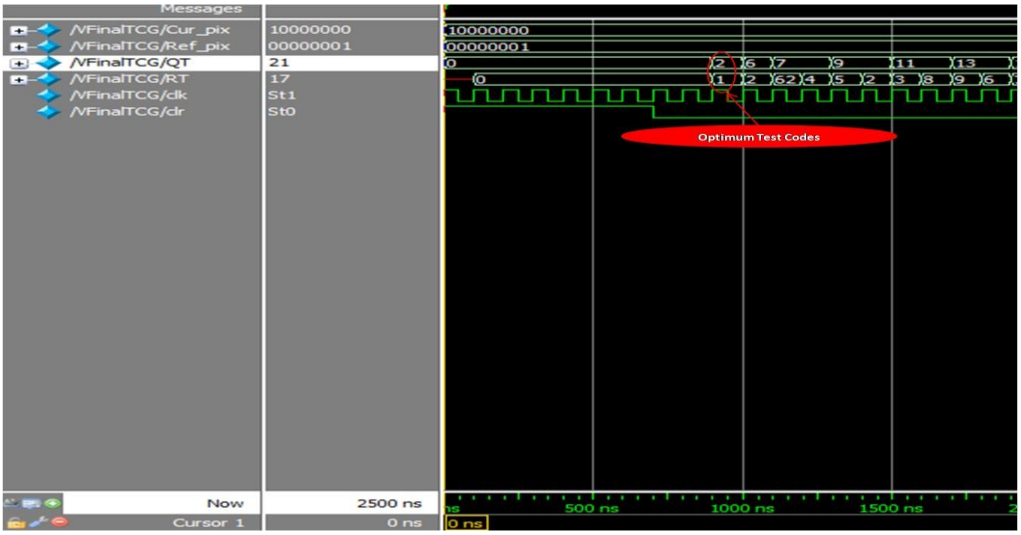

Figure 5. Simulation Result of TCG Circuit obtaining its value from Current and Reference pixel values

Figure 6. Simulation Result of Fault Detection Circuit identifying the fault using RP Codes and Test Codes for a Specific PE in a Motion Analysis Process

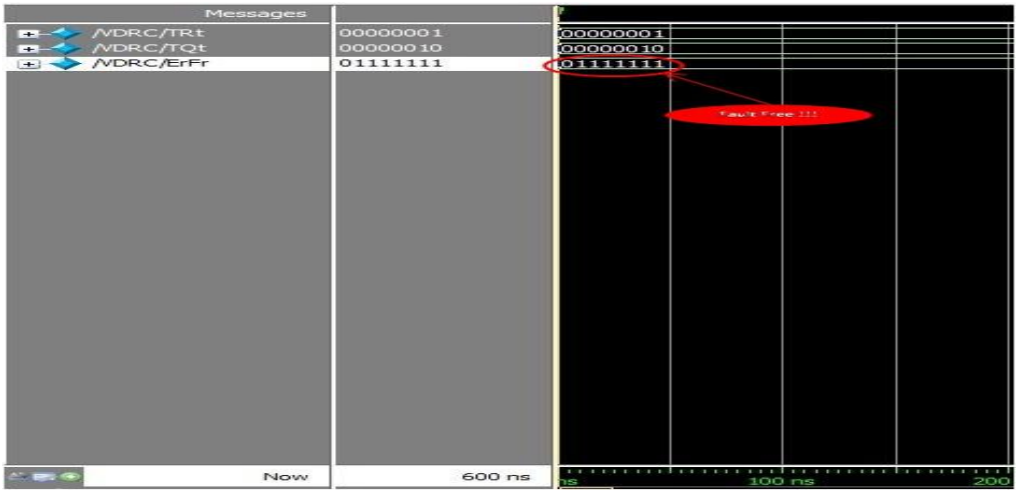

Figure 7. Simulation Result of Data Recovery Circuit identifying the fault using Test Codes for a Specific PE in a Motion Analysis Process

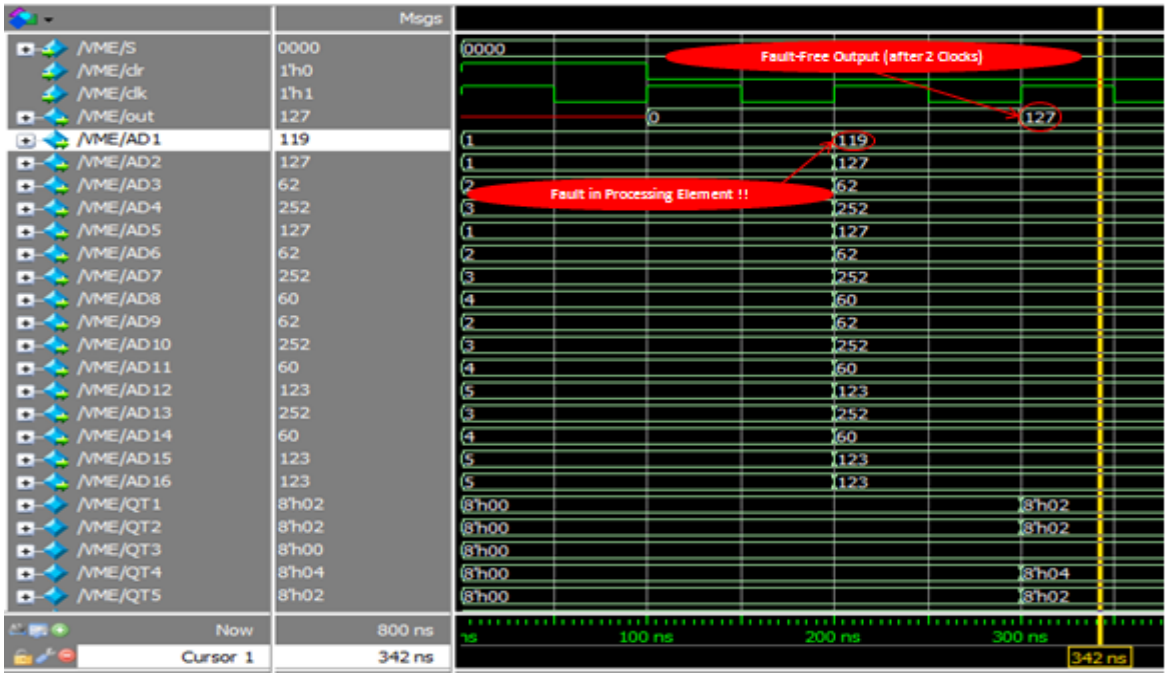

Figure 8.Simulation Result of Fault Recovery design in identifying the fault and recovering the original data for a Specific PE in a Motion Analysis Process

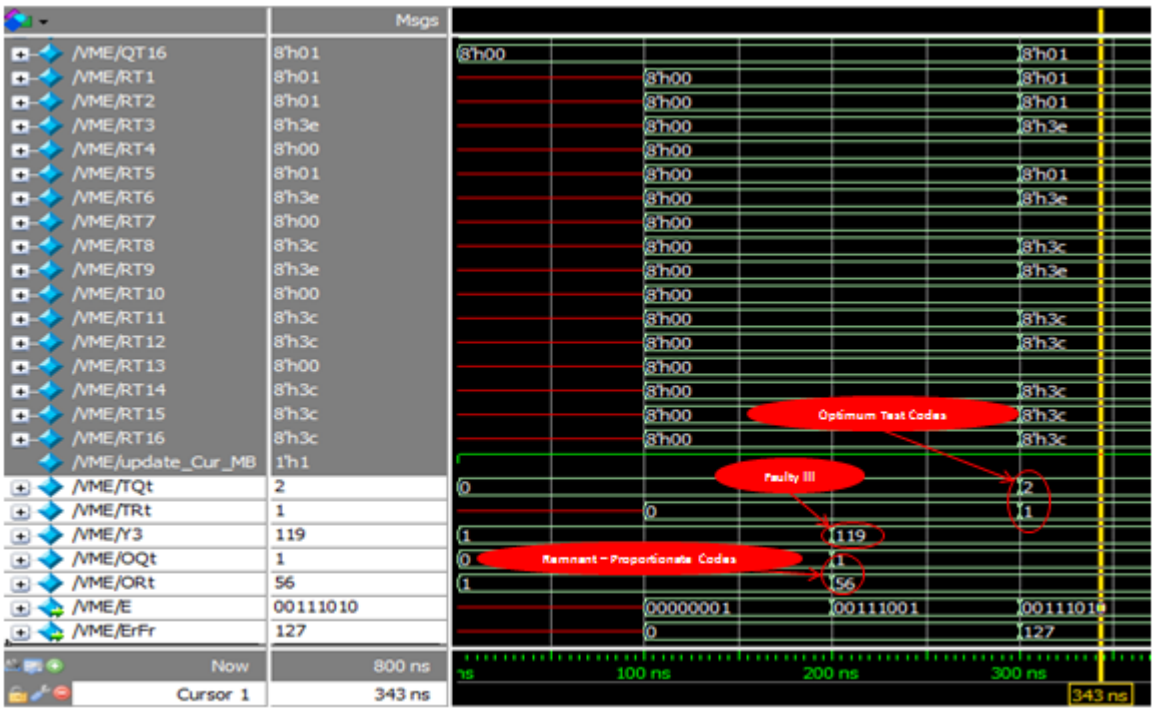

Figure 9. Simulation Result of Fault Recovery design in identifying the fault using RP Codes and Test Codes for a Specific PE in a Motion Analysis Process

VIII. RESULTS AND ITS DISCUSSION

Table 3. Design Specifications of Fault Recovery Design with 16 PEs and 16 TCGs in a Motion Analysis Process for a specific Processing Element

S.NO	DESIGN SPECIFICATIONS	
1.	Algorithm	Remnant Proportionate Code Generation Algorithm
2.	No. of Processing Elements	16 (4x4 Array)
3.	Supported Block Size	All Available Sizes
4.	Process Technology	TSMC 0.18-µm 1P6M CMOS technology
5.	Maximum Frequency	538.92 MHz

Table 4. Performance Analysis of Fault Recovery Design with 16 PEs and 16 TCGs in a Motion Analysis Process for a specific Processing Element

S.NO	PERFORMANCE ANALYSIS	
1.	No. of Test Patterns	16
2.	Operating Speed	10.973 ns (91.13 MHz)
3.	Area(Gate Counts)	11726
4.	Power Consumption	289.68 mW
5.	Operating Conditions	29°C
5.	Fault Coverage	100 %

IX. CONCLUSION

This work presents a novel Fault Recovery scheme for detecting the faults and recovering the data of PEs in a MA. Based on the RP code, a RPCG-based TCG design is developed to generate the corresponding test codes to identify faults and recover data. The proposed Fault Recovery scheme is also implemented by using Verilog Hardware Description Language and synthesized by the synopsys Design Compiler with TSMC 0.18-m1P6MCMOS technology. Experimental results indicate that that the proposed Fault Recovery design can effectively identify faults and will recover data in PEs of a MA with an operating time of 10.973ns with acceptable area cost.

REFERENCES

[1] Chang-Hsin Cheng; Yu Liu; Chun-Lung Hsu , "Design of an Error Detection and Data Recovery Architecture for Motion Estimation Testing Applications," *IEEE Trans. Vary Large Scale Integr. (VLSI) Systs.*, vol. 20, no. 4, pp. 665–672, Apr. 2007.

[2] L. Breveglieri, P. Maistri, and I. Koren, "A Note on Error Detection in RSA architecture by means of Residue Codes," in *Proc. IEEE Int. Symp. On-Line Testing*, pp. 176-177, Jul. 2006.

[3] 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.

[4] 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.*, pp.1–4, 2009.

[5] 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.*, pp. 1188–1191, May 2008.

[6] C. L. Hsu, C. H. Cheng, and Y. Liu, "Built In Self Detection/Correction architecture for Motion Estimation Computing Arrays," *IEEE Trans. Vary Large Scale Integr.(VLSI) Systs.*, vol. 18, no. 2, pp. 319–324, Feb. 2010.

[7] 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.*, pp. 1751–1754, Dec. 2008.

[8] 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.

[9] 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.

[10] C. P. Kung, C. J. Huang, and C. S. Lin, "Fast fault simulation for BIST applications," *in Proceedings of the Fourth Asian Test Symposium*, pp. 93–99, 1995.

[11] 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.*, pp. 349–352, Jun. 2004.

[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] 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.*, pp. 1–4, Apr. 2009.

[14] E. J. McCluskey, "Built In Self Test Technique," *IEEE Design and Test of Computers*, vol. 2, no. 2, pp. 29–36, Apr. 1985.

[15] D. K. Park, H. M. Cho, S. B. Cho, and J. H. Lee, "A fast motion estimation algorithm for SAD optimization in sub pixel," in *Proc. Int. Symp. Integr. Circuits*, pp. 528–531, Sep. 2007.

[16] S. J. Piestrak, D. Bakalis, and X. Kavousianos, "On the design of self testing checkers for modified Berger codes," in *Proc. IEEE Int. Workshop On Line Testing*, pp. 153–157, Jul. 2001.

[17] J. M. Portal, H. Aziza, and D. Nee, "EEPROM memory: Threshold voltage Built In Self Diagnosis," in *Proc. Int. Test Conf.*, pp. 23–28, Sep. 2003.

[18] 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.

[19] 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*, pp. 1–4, Apr. 2007.

[20] N. A. Touba and E. J. McCluskey, "Pseudo Random pattern testing of bridging faults," *in International Conference on Computer Design*, pp. 54–60, 1997.