[International Journal of Electronics and Electrical Engineering](https://www.interscience.in/ijeee)

Volume 2 | Issue 4 Article 1

April 2014

DESIGN OF AN ERROR DETECTION AND DATA RECOVERY ARCHITECTURE FOR MOTION ESTIMATION TESTING APPLICATIONS

V. SWARNALATHA
VLSI System Design A.I.T.S, Rajampet Kadapa (Dt), A.P., India, swarna.vankam@gmail.com

K. SRINIVASA RAO
Dept of E.C.E, A.I.T.S, Rajampet Kadapa (Dt), A.P., India, ksraoaits@gmail.com

Follow this and additional works at: https://www.interscience.in/ijeee

Part of the Power and Energy Commons

Recommended Citation

SWARNALATHA, V. and RAO, K. SRINIVASA (2014) "DESIGN OF AN ERROR DETECTION AND DATA RECOVERY ARCHITECTURE FOR MOTION ESTIMATION TESTING APPLICATIONS," *International Journal of Electronics and Electrical Engineering*: Vol. 2 : Iss. 4 , Article 1.
Available at: https://www.interscience.in/ijeee/vol2/iss4/1

This Article is brought to you for free and open access by Interscience Research Network. It has been accepted for inclusion in International Journal of Electronics and Electrical Engineering by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

DESIGN OF AN ERROR DETECTION AND DATA RECOVERY ARCHITECTURE FOR MOTION ESTIMATION TESTING APPLICATIONS

V. SWARNA LATHA[1] & K. SRINIVASA RAO[2]

[1]VLSI System Design A.I.T.S, Rajampet Kadapa (Dt), A.P., India &
[2]Dept of E.C.E, A.I.T.S, Rajampet Kadapa (Dt), A.P., India
E-mail : swarna.vankam@gmail.com & ksraoaits@gmail.com

Abstract - Motion estimation (ME) in a video coding system is the critical role, so testing such a module is of priority concern. While focusing on the testing of ME in a video coding system, this work presents an error detection and data recovery (EDDR) design based on residue -and- quotient (RQ) code. An error in processing elements (PEs) can be detected and recovered effectively by using the proposed EDDR design. Importantly, the proposed EDDR design performs satisfactorily in terms of throughput and reliability for motion estimation (ME) testing applications.

***Keywords**: Motion Estimation, Processing elements, TCG, RQ code, EDC*

I. INTRODUCTION

Advances in semiconductors, digital signal processing, and communication technologies have made multimedia applications more flexible and reliable.

A good example is the H.264 video standard, also known as MPEG-4 Part 10 Advanced Video Coding, which is widely regarded as the next generation video compression standard Video compression is necessary in a wide range of applications to reduce the total data amount required for transmitting or storing video data.

Among the coding systems, a ME in a video coder is the critical role so testing such a module is of priority concern. 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.

In the advance of VLSI technologies facilitate the integration of a large number of PEs of a ME into a chip; the logic-per-pin ratio is subsequently increased, thus decreasing significantly the efficiency of logic testing on the chip.

As a commercial chip, it is absolutely necessary for the ME to introduce design for testability (DFT) .DFT focuses on increasing the ease of device testing, thus guaranteeing high reliability of a system. DFT methods rely on reconfiguration of a circuit under test (CUT) to improve testability. While DFT approaches enhance the testability of circuits, advances in sub-micron technology and resulting increases in the complexity of electronic circuits and systems have meant that built-in self-test (BIST) schemes have rapidly become necessary in the digital world.

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 EDDR architecture based on the RQ code to detect errors and recovery data in PEs of a ME.

This paper is organized as follows. Section 2 gives the circuit design of RQ code generator. Section 3 introduces the EDDR architecture and test method. Conclusions are offered in section 4.

II. 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

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

The RQ code of X modulo m expressed as $R=|X|_m$ Q=[X/m], respectively. Notably [i] denotes the largest integer not exceeding i.

According to the above RQ code expression, the corresponding circuit design of the RQCG can be realized. In order to simplify the complexity of circuit design, the implementation of the module is generally dependent on the addition operation. Additionally, based on the concept of residue code, the following definitions shown can be applied to generate the RQ code for circuit design.

Definition 1:

$|N_1+N_2|_m = ||N_1|_m + |N_2|_m|_m$. (2)

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

$|Nj|m = ||n_1|_m + |n_2|_m\ldots+|n_j|_m|_m$. (3)

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

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

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

R = |X|m

=|Y0+Y1|m = |Z0 + Z1|m = (Z0+Z1)α (5)

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

where

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

Notably, since the value of Y0 + Y1 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 Z0,Z1, α and β .

Based on (5) and (6), the corresponding circuit design of the RQCG is easily realized by using the simple adders (ADDs). Namely, the RQ code can be generated with a low complexity and little hardware cost.

III. EDDR ARCHITECTURE

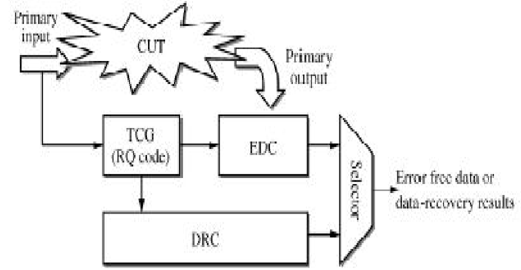


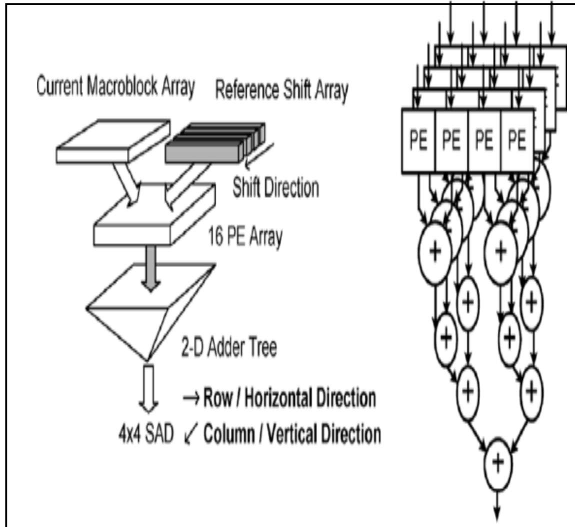
Fig. 1. Conceptual view of the EDDR Architecture

Fig. 1 shows the conceptual view of the proposed EDDR 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 Fig. 1 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 EDDR scheme to detect errors and recover the corresponding data.

This work adopts the systolic ME [19] as a CUT to demonstrate the feasibility of the proposed EDDR 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. Fig. 2 shows an example of the proposed EDDR 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 utilizing 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 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 are 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 SAD' denotes the computed SAD value with SA faults.

C. TCG Design

According to Fig. 2, TCG is an important component of the proposed EDDR 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 Fig. 2 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:

$$\begin{aligned}\mathrm{SAD} &= \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}|X_{ij}-Y_{ij}| \\ &= \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}|(q_{xij}\cdot m+r_{xij})-(q_{yij}\cdot m+r_{yij})| \quad (7)\end{aligned}$$

where r_{xij}, q_{xij} and r_{yij}, q_{yij} denote the corresponding RQ code of X_{ij} and Y_{ij} modulo m . Importantly, Xij and Yij 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 (R_T and Q_T) form TCG. Namely, the circuit design of TCG can be easily achieved (see Fig. 3) by using

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

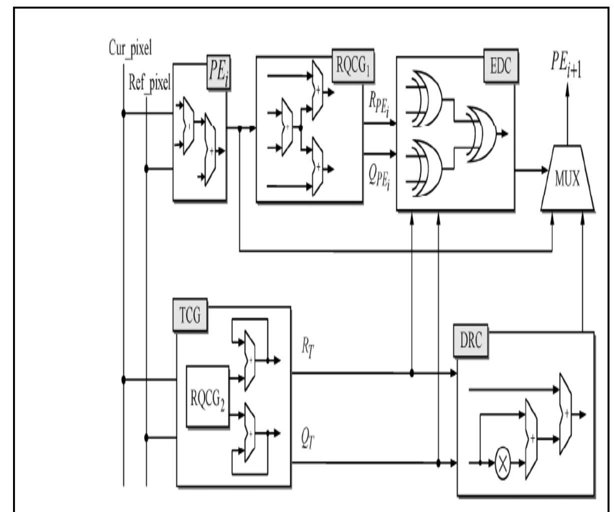


Fig. 2. A specific Pei testing processes of the proposed EDDR architecture

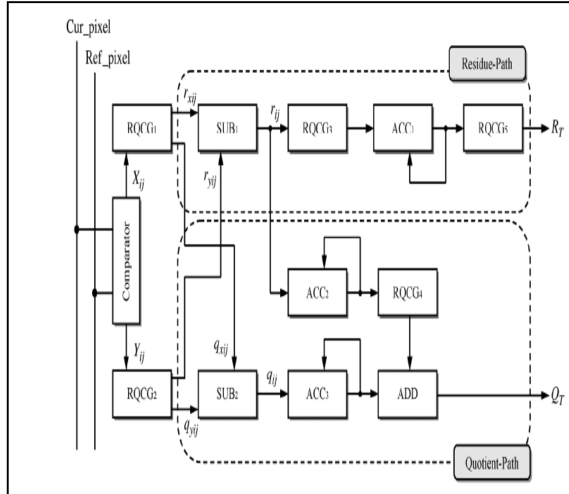


Fig. 3. Circuit design of the TCG

D. EDDR Process

Error detection in a specific PE is achieved by using EDC, which is utilized to compare the outputs between TCG and RQCG in order to determine whether errors have occurred. The EDC output is then used to generate a 0/1 signal to indicate that the tested 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. In other words, the SAD value is transformed to $\mathrm{SAD}' = \mathrm{SAD} + e$ if an error occurred. Notably, the error signal is expressed as

$e = q_e.m + r_e$

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

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

To realize the operation of data recovery in, a Barrel shift and a corrector circuits are necessary to achieve the functions of $(2^j \times Q_T)$ and $(-Q_T + R_T)$,

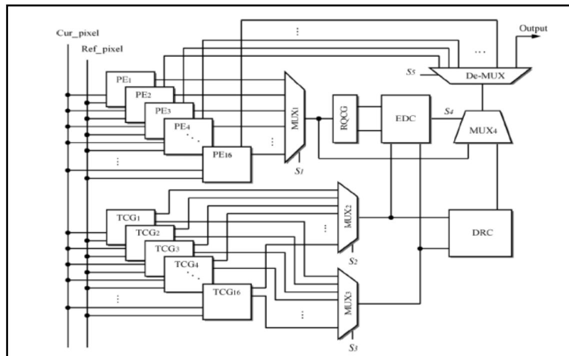


Fig.4. EDDR Architecture design for a ME

respectively. Notably, the proposed EDDR 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 PEi+1 for sub sequent testing.

E. Overall Test Strategy

By extending the testing processes of a specific PE_i in Fig. 2, Fig. 4 illustrates the overall EDDR 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 generate the test RQ code R_T and Q_T. Second, the SAD value from the tested object PEi , which is selected by MUX1, is then sent to the RQCG circuit in order to generate R_{PEi} and Q_{PEi} codes. Meanwhile, the corresponding test codes R_{Ti} and Q_{Ti} from a specific TCGi are selected simultaneously by MUXs 2 and 3, respectively. Third, the RQ code from TCGi and RQCG circuits are compared in EDC to determine whether the tested object PEi have errors. The tested object PEi 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 TCGi , i.e. the appropriate R_{Ti} and Q_{Ti} codes from TCGi are selected by MUXs 2 and 3, respectively, to recover data. Fourth, the error-free data or data recovery results are selected by MUX . Notably, control signal S4 is generated from EDC, indicating that the comparison result is error-free (S4 = 0) or errancy (S4 = 1) . Finally, the error-free data or the data-recovery result from the tested object PEi is passed to a De-MUX, which is used to test the next specific PE_{i+1}; otherwise, the final result is exported.

CONCLUSION

This work presents EDDR architecture for detecting the errors and recovering the data of PEs in a 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. The RQ code generation, test code generation was also discussed

REFERENCE

[1]. 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, Sep. 2007, pp. 528–531.

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

[3]. D. P. Vasudevan, P. K. Lala, and J. P. Parkerson, “Self-checking carryselect adder design based on two-rail encoding,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 54, no. 12, pp. 2696–2705, Dec. 2007.

[4]. X. Yu, T. Meng, Z. Dai, and X. Yang, "Design and implementation of reconfigurable shift unit using FPGAs," in Proc. IEEE Int. Symp. Pervasive Comput. Applic., Aug. 2006, pp. 543–545.

[5]. K. Neubeck, Practical Reliability Analysis. Englewood Cliffs, NJ: Pearson Prentice-Hall, 2004.

[6]. X. Li, J. Qin, B. Huang, X. Zhang, and J. B. Bernstein, "A new SPICE reliability simulation method for deep submicrometerCMOSVLSI circuits," IEEE Trans. Device Mater. Reliabil., vol. 6, no. 2, pp. 247–257,Jun. 2006.

[7]. 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 , *IEEE Trans. Comput.*, vol. 58, no. 6, pp. 851–857, Jun. 2009.

[8]. L. Breveglieri, P. Maistri, and I. Koren, "A note on error detection in an RSA architecture by means of residue codes," in *Proc. IEEE Int. Symp. On-Line Testing*, Jul. 2006, 176 177.

[9]. S. J. Piestrak, D. Bakalis, and X. Kavousianos, "On the design of selftestingcheckers for modified Berger codes," in *Proc. IEEE Int. WorkshopOn- Line Testing*, Jul. 2001, pp. 153–157.

[10]. 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.