IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Efficient EDDR Architecture for Motion Estimation in Advanced Video Coding Systems

M.Supraja[*1], M.Pavithra Jyothi[2]

[*1,2]Assistant Professor Department of ECE , JNTUA, India

m.supraja517@gmail.com

Abstract

Motion estimation algorithms are used in various video coding systems. With the advent of VLSI technology, a large collection of processing elements can be assembled to achieve high-speed computation economically, 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 the residue-and-quotient (RQ) code, to embed into ME for video coding testing applications. this paper describes a novel testing scheme of motion estimation. The key part of this scheme is to offer high reliability for motion estimation architecture. The experimental result shows the design achieve 100% fault coverage. And, the main advantages of this scheme are minimal performance degradation, small cost of hardware overhead and the benefit of throughput at-speed testing.

Keywords: Area overhead, data recovery, error detection, motion estimation, reliability, residue-and-quotient (RQ) code.

Introduction

Multimedia applications are more flexible and reliable when we used Advances in semiconductors, digital signal processing, and communication technologies. 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., Motion estimation explores the temporal redundancy, which is inherent in video sequences, and it represents a basis for lossy video compression. Other than video compression, motion estimation can also be used as the basis for powerful video analysis and video processing. A ME generally consists of PEs with a size of 4 x 4. 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. 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. Systems have meant that built-in self-test(BIST) schemes have rapidly become necessary in the digital world. BIST for the ME does not expensive test equipment, ultimately lowering test costs . The built-in testing approach not only detect faults but also specify their locations for error correcting. Thus, extended schemes of BIST referred to as built-in self-diagnosis and built-in self-correction have been developed recently. While the extended BIST schemes generally focus on memory circuit, testing-related issues of video coding have seldom been addressed.

RQ Code Generation

To detect circuit errors Coding approaches such as parity code, Berger code, and residue code have been considered for design applications. Residue code is generally separable arithmetic codes by estimating a residue for data and appending it to data. For instance, assume that N denotes an integer, and N_1 and N_2 represent data words, and m refers to the modulus. A separate residue code of interest is one in which N is coded as a pair $(N,N|_m$. Notably, $|N|_m$ is the residue of N modulo m 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 cannot 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 mathematical model of RQ code is simply described as follows. Assume that binary data is expressed as

$$X = \{b_{n-1}b_{n-2} \dots \dots \dots b_2 b_1 b_0\} = \sum_{j=0}^{n-1} b_j 2^j \quad \dots \dots \dots \dots (1)$$

The RQ code of X modulo expressed as R= R=|Xm|, Q=[X/m],respectively. Notably, [i] denotes the largest integer not exceeding i. 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 $N_j=n_1+n_2+$-------n_j , then

$||N_j|_m=||n_1|_m+ |n_2|_{m+}$--------+ $|n_j|_m|_m$.-------(3)

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

$$X = \sum_{j=0}^{n-1} b_j 2^j$$

$$= \left(\sum_{j=0}^{k-1} b_j 2^j\right) + \left(\sum_{j=k}^{n-1} b_j 2^{j-k}\right) 2^k$$

$$= Y_0 + Y_1 2^k \dots \dots \dots (4)$$

Significantly, the value of k is equal to[n/2] and the data formation of Y_0 and Y_1 are a decimal system. If the modulus $m=2^k-1$, then the residue code of X modulo m is given by

$$R = |X|_m$$

$$= |Y_0 + Y_1|_m = |Z_0 + Z_1|_m = (Z_0 + Z_1)\alpha \dots \dots .(5)$$

$$Q = \left[\frac{X}{m}\right]$$

$$= \left[\frac{Y_0 + Y_1}{m}\right] + Y_1$$

$$= \left[\frac{Z_0 + Z_1}{m}\right] + Z_1 + Y_1$$

$$= Z_1 + Y_1 + \beta \dots \dots \dots \dots \dots (6)$$

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 $Y_0 + Y_1$ 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 parametersZ_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 little hardware cost.

Proposed EDDR Architecture Design

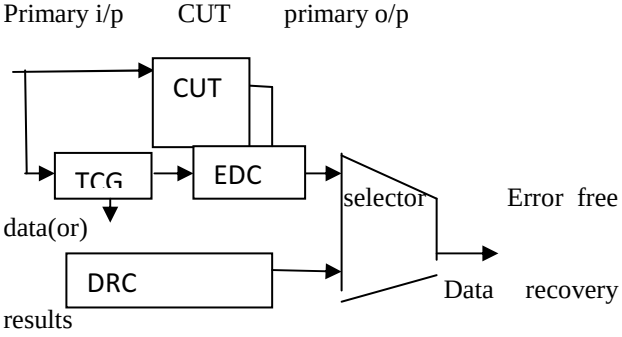


Fig. 1. Conceptual view of the proposed EDDR architecture.

In the above Fig. Our 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 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 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.

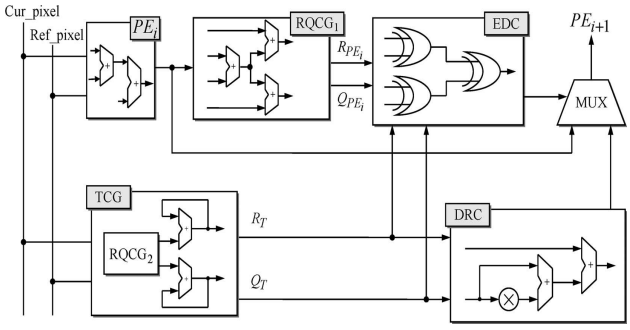


Fig (2):A specific PE_i testing processes of the proposed EDDR architecture

Fig. 2 shows an example of the proposed EDDR circuit design for a specific PE_i of a ME.

A. Fault Model

To construct a ME The PEs are essential building blocks and are connected regularly. Generally, PEs are surrounded by sets of ADDs and accumulators that determine how data flows through them.testing assignment can be easily achieved by using the fault model, cell fault model (CFM) .Using CFM makes the tests independent of the adopted synthesis tool and vendor library. Moreover, a more comprehensive fault model, i.e. the stuck-at (SA) model, The stuck-at fault is a logical fault model that has been used successfully for decades. A stuck-at fault affects the state of logic signals on lines in a logic circuit, including primary inputs (PIs), primary outputs (POs), internal gate inputs and outputs, fanout stems (sources), and fanout branches. A stuck-at fault transforms the correct value on the faulty signal line to appear to be stuck at a constant logic value, either a logic 0 or a logic 1, referred to as stuck-at-0 (SA0) or stuck-at-1 (SA1), respectively.

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.

B. TCG Design

TCG design is based on the ability of the RQCG circuit to generate corresponding test codes in order to detect errors and recover data.

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

Proposed EDDR architecture. By utilizing PEs, SAD shown in as follows, in a macroblock with size of N x N can be evaluated:

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

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

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

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

$$= ||(X_{00} - Y_{00})|_m + |(X_{01} + Y_{01})|_m \ldots\ldots + |(X_{(N-1)(N-1)} + Y_{(N-1)(N-1)})|_m|_m$$

$$= \left|\left|(q_{xDD}.m + r_{xDD}) - (q_{yDD}.m + r_{yDD})\right|_m + \cdots\right.$$

$$\left.|(q_{x(N-1)(N-1)}.m - r_{x(N-1)(N-1)}) - (q_{y(N-1)(N-1)}.m \mp r_{y(N-1)(N-1)}|_m|_m\right.$$

$$= ||(r_{xDD} - r_{yDD})|_m + |(r_{xD1} - r_{yD1}|_m \cdots\cdots |(r_{x(N-1)(N-1)} - r_{y(N-1)(N-1)}|_m|_m$$

$$= ||r_{DD}|_m + |r_{D1}|_m + \cdots\cdots + |r_{(N-1)(N-1)}|_m|_m \cdots\cdots\cdots\cdots (8)$$

$$Q_T = \left\lfloor \frac{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(X_{ij} - Y_{ij})}{m} \right\rfloor$$

$$= \left\lfloor \frac{(X_{DD} - Y_{DD}) + (X_{D1} - Y_{D1}) + ---}{+(X_{(N-1)(N-1)} - Y_{(N-1)(N-1)})} \right\rfloor$$

$$= \left\lfloor \frac{(q_{xDD}.m - q_{yDD}.m)}{m} + \frac{(r_{xDD} - r_{yDD})}{m} + \frac{(q_{xD1}.m - q_{yD1}.m)}{m} + \frac{(r_{xD1} - r_{yD1})}{m} + \cdots \right\rfloor$$

$$= \left\lfloor \frac{(q_{xDD}.m - q_{yDD}.m)}{m} + \frac{(r_{xDD} - r_{yDD})}{m} + \frac{(q_{xD1}.m - q_{yD1}.m)}{m} + \frac{(r_{xD1} - r_{yD1})}{m} + \cdots \right\rfloor$$

$$= \left\lfloor \frac{(q_{xDD} - q_{yDD}) + (q_{xD1} - q_{yD1}) + \cdots + (r_{xDD} - r_{yDD}) + (r_{xD1} - r_{yD1}) + \cdots}{m} \right\rfloor$$

$$= q_{DD} + q_{D1} + q_{(N-1)(N-1)} + \left\lfloor \frac{(r_{DD} - r_{D1} + \cdots (r_{(N-1)(N-1)})}{m} \right\rfloor \cdots\cdots\cdots (9)$$

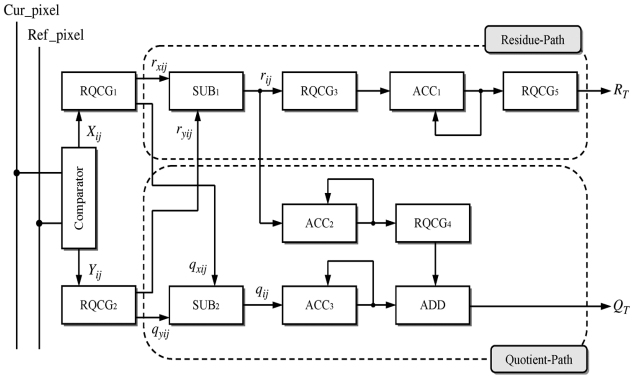


Fig. 3. Circuit design of the TCG.

In fig(3) the quasi block will generate quotient value according to given input. Here we applied 3 bit input then generate 8 bit signal as output.

In this module consists flipflop act as a accumulator. We can store a bit of data. Flip-flop" is the common name given to two-state devices which offer basic memory for sequential logic operations.

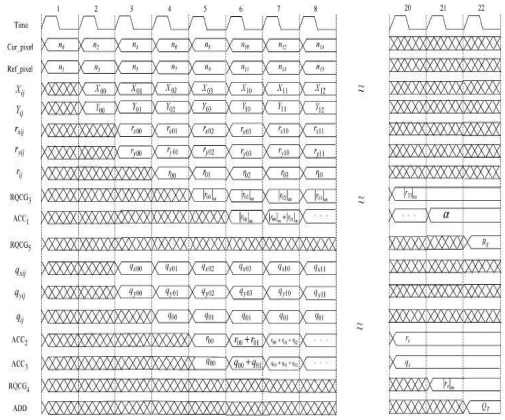


Fig. 4. Timing chart of the TCG.

Fig. 4 shows the timing chart for a macro block with a size Of 4 X 4 in a specific PE_i to demonstrate the operations of the TCG circuit. The data n_0 and n_1 from Cur_pixel and Ref_pixel must be sent to a comparator in order to determine the luminance pixel value X_{ij} and Y_{ij} at the 1st clock. Notably, if $X_{ij} >= Y_{ij,}$ then X_{ij} and Y_{ij} and are the luminance pixel value of Cur_pixel and Ref_pixel, respectively. Conversely X_{ij} represents the luminance pixel value of Ref_pixel, and Y_{ij} denotes the luminance pixel value of Cur_pixel when $X_{ij} <Y_{ij}$. At the 2nd clock, the values of X_{00} and Y_{00} and are generated and the corresponding RQ code $rx_{00,}$ $qx_{00,}$ $ry_{00,}$ qy_{00} can be captured by the $RQCG_1$ and $RQCG_2$ circuits if the 3rd clock is triggered. Equations (8) and (9) clearly indicate that the codes of r_{00} and q_{00} can be obtained by using the circuit of a subtracter (SUB).The 4th clock displays the operating results. The modulus value of r_{00} is then obtained at the 5th clock. Next, the summation of quotient values and residue values of modulo m are proceeded with from clocks 5–21 through the circuits of ACCs. Since a 4x4 macroblock in a specific of a ME contains 16 pixels, the corresponding RQ code (R_T and Q_T) is exported to the EDC and DRC circuits in order to detect errors and recover data after 22 clocks. Based on the TCG circuit design shown in Fig. 4, the error detection and data recovery operations of a specific PEi in a ME can be achieved.

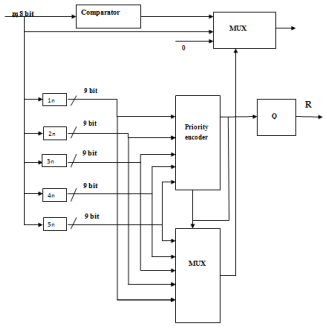


Fig.5. M mod N Operation

C. EDDR Processes

Fig. 2 clearly indicates that by using EDC the operations of error detection in a specific PEi is achieved, which is utilized to compare the outputs between TCG and $RQCG_1$ in order to determine whether errors have occurred. If the values of $R_{PEi} \neq R_T$ 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.

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 e occurred. Where the error signal e is expressed as

$$E = q_e . m + r_e \quad \ldots\ldots\ldots\ldots\ldots\ldots (10)$$

Under the faulty case, the RQ code from $RQCG_2$ of the TCG is still equal to (8) and(9). However, R_{PEi} and Q_{PEi} are changed to (13) and (14) because an error e has occurred. Thus, the error in a specific PEi can be detected if and only if (8)≠ (11) and/or (9)≠ (12):

$$R_{PEi} = |SAD'|_m$$

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

$$= ||r_{00}|_m + |r_{01}|_m + \cdots + |r_{(N-1)(N-1)}|_m + r_e|_m|_m \quad \ldots\ldots\ldots\ldots (11)$$

$$Q_{PEi} = \left\lceil \frac{SAD'}{m} \right\rceil$$

$$= \left\lceil \frac{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(X_{ij} - Y_{ij}) + e}{m} \right\rceil$$

$$= q_{00} + q_{01} + \cdots + q_{(N-1)(N-1)} + q_s + \left[\frac{(r_{00} + r_{01} + \cdots + r_{(N-1)(N-1)} + r_s)}{m}\right] \ldots\ldots\ldots\ldots\ldots\ldots (12)$$

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

$$SAD = m * Q_T + R_T$$
$$= (2^j - 1) * Q_T + R_T$$
$$= 2^j * Q_T - Q_T + R_T \ldots\ldots\ldots (13)$$

To realize the operation of data recovery in (13), a Barrel shift and a corrector circuits are necessary to achieve the functions of ($2^j * Q_T$) and ($- Q_T + R_T$) , 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 tothe next specific PE_{i+1} for subsequent testing.

D.Numerical Example

A numerical example of the 16 pixels for a 4 x 4 macro blockin a specificPE_i of a ME is described as follows.

	0	1	2	3
0	128	128	64	255
1	128	64	255	64
2	64	255	64	128
3	255	64	128	128

Cur_pixel

	0	1	2	3
0	1	1	2	3
1	1	2	3	4
2	2	3	4	5
3	3	4	5	5

Ref_pixel

Fig. 6 . Example of pixel values.

Fig6. presents an example of pixel values of the Cur_pixel and Ref_pixel.

Based on (7), the SAD value of the 4 x 4 macro block is

$$SAD = \sum_{i=0}^{3}\sum_{j=0}^{3} |X_{ij} - Y_{ij}|$$
$$= |X_{00} - Y_{00}| + |X_{01} - Y_{01}| + \ldots\ldots + |X_{33} - Y_{33}|$$
$$= (128 - 1) + (128 - 1) + \ldots\ldots + (128 - 1)$$
$$= 2124 \ldots\ldots\ldots\ldots\ldots (14)$$

According to the description of RQ code, the modulo is assumed here to be $m=2^6 -1 = 63$. Thus, based on (8) and (9), the RQ code of the SAD value shown in (14) are $R_T = R_{PEi} = |2124|_{63} = 45$ and $Q_T = Q_{PEi} = [2124/63] = 33$.Since the value of R_T (Q_T) is equal to R_{PEi} (Q_{PEi}), EDC is enabled and a signal "0" is generated to describe a situation in which the specific PEi is error-free.

Conversely, if SA1 and SA0 errors occur in bits 1 and 12 of a specific PEi, i.e. the pixel values of PEi, $2124=100001001100_2$ is turned into $77=000001001101_2$, resulting in a transformation of the RQ code of R_{PEi} and Q_{PEi} into $77|_{63} = 14$ and [77/63]=1. Thus, an error signal "1" is generated from EDC and sent to the MUX in order to select the recovery results from DRC.In the same way we can get the recovery data for all PE_i values.

E. Overall Test Strateg ,

Fig. 7 illustrates the overall EDDR architecture design of a ME.

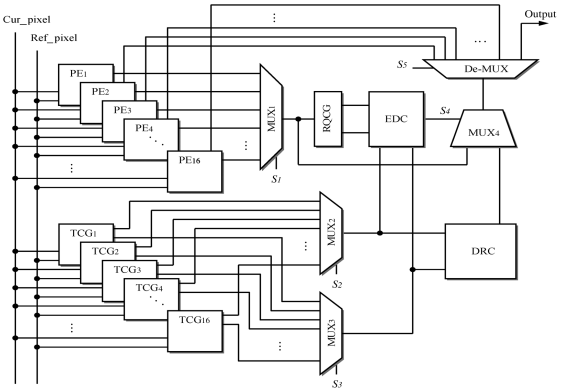


Fig. 7. Proposed EDDR architecture design for a ME.

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 PE_i, which is selected by MUX_1 , 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 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 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 are selected by MUXs 2 and 3, respectively, to recover data. Fourth, the error-free data or data recovery results are selected by MUX_4.

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.

Results and Discussion

The verification of the circuit design is performed using the VHDL and then synthesized by the Synopsys Design Compiler with TSMC 0.18- µm 1P6M CMOS technology to demonstrate the feasibility of the proposed EDDR architecture design for ME testing applications

A.Experimental Results

Table 1 Estimation of area overhead and time penalty

Components	PE	RQCG	EDC	TCG	DRC
Area (Gate counts)	69482	1779	667	3265	2376
Operation time (ns)	973.76	10.17	6.02	1016.56	17.99
Area overhead(%)	5.13				
Time Penalty (%)	6.24				

Table I summarizes the synthesis results of area overhead and time penalty of the proposed EDDR architecture.The area is estimated based on the number of gate counts. By considering 16 PEs in a ME and 16 TCGs of the proposed EDDR architecture, the area overhead of error detection, data recovery, and overall EDDR architecture (AO_{ED} , AO_{DR}, and AO_{EDDR}) are

$$AO_{ED} = \frac{1779 + 3265 * 16 + 667}{69482 * 16} = 4.92\% \quad \ldots\ldots\ldots\ldots (15)$$

$$AO_{DR} = \frac{3265 * 16 + 2376}{69482 * 16} = 4.91\% \quad \ldots\ldots\ldots\ldots (16)$$

$$AO_{EDDR} = \frac{1779 + 667 + 3265 * 16 + 2376}{69482 * 16} = 5.13\% \quad \ldots\ldots\ldots\ldots (17)$$

The time penalty is another criterion to verify the feasibility of the proposed EDDR architecture. Table I also summarizes the operating time evaluation of a specific PE_i and each component in the proposed EDDR architecture. The following equations show the time penalty of error detection and data recovery.
(TP_{ED} and TP_{DR}) operations for a 4 x 4 macroblock (a PE with 16 pixels):

$$TP_{ED} = \frac{(1016.56 + 6.02) - 973.76}{973.76} = 5.01\% \quad \ldots\ldots\ldots (18)$$

$$TP_{DR} = \frac{(1016.56 + 17.99) - 973.76}{973.76} = 6.24\% \quad \ldots\ldots\ldots (19)$$

If the proposed EDDR architecture is embedded into a ME for testing, in which the entire timing penalty is equivalent to that for testing a single PE., i.e. approximately about 5.01% and 6.24% time penalty of the operations of error detection and data recovery, respectively.

The operating time of the RQCG circuit can be neglected to evaluate TP_{ED} because TCG covers the operating
time of RQCG. Additionally, the error-free/errancy signal from EDC is generated after 1022.58 ns (1016.56+6.02). Thus, the error-free data is selected directly from the tested object PE_i because the operating time of the tested object is faster
than the results of data recovery from DRC.

A. Performance Discussion

The number of TCGs significantly influences the circuit performance in terms of area overhead and throughput.

Figs.8and 9 illustrate the relations between the number of
TCGs, area overhead and throughput.

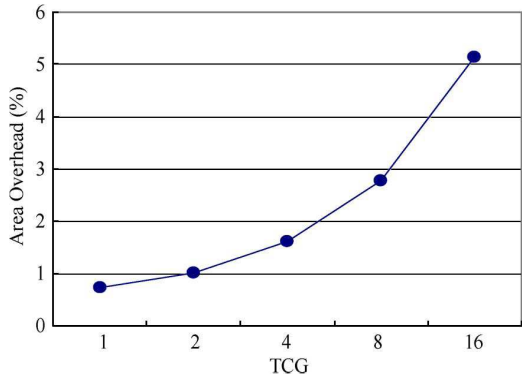


Fig.8. Relation between TCG and area overhead.

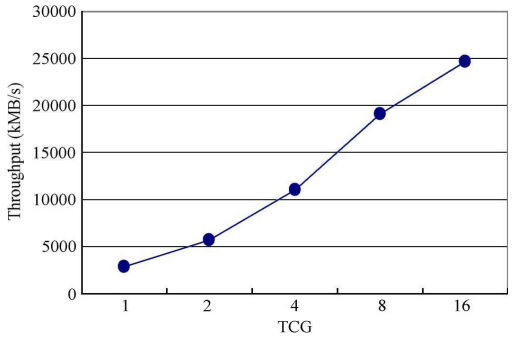


Fig. 9. Relation between TCG and throughput.

The area overhead is less than 2% if only one TCG is used to execute; however,at this time, the throughput is extremely small.

Notably, the throughput of a ME without embedding the proposed EDDR architecture is about 25 800 kMB/s.

Fig. 9 clearly indicates that the throughput is around 25000 kMB/s, if the proposed EDDR architecture with 16 TCGs is embedded into a ME for testing. Thus, to maintain the same throughput as much as possible, 16 TCGs must be adopted in the proposed EDDR architecture for a ME testing applications. Although the area overhead is increased if 16 TCGs used (see Fig.8), the area overhead is only about 5.13%, i.e. an acceptable design for circuit testing. This work also addresses reliability-related issues to demonstrate the feasibility of the proposed EDDR architecture. Reliability is the probability that a component or a system performs its required function under different operating conditions encountered for a certain time period. The constant failure rate reliability model is used to estimate the reliability of the proposed EDDR architecture for ME testing applications,

$$R(t) = e^{-\lambda t} = e^{-\lambda_b \pi_T \pi_Q \pi_E t}$$

$$= e^{-\left(\frac{1.5 D_G}{T^{1.5}}\right) G^{\frac{5}{T^{1.5}}} \pi_T \pi_Q \pi_E c} \quad ---(20)$$

where λ denotes the failure- rate; represents the base failure-rate of MOS digital logic; G refers to Gate count; Π_t=1.0(25^0c); (hermetic package); and (ground benign environment).The failure-rate in (20) can be expressed as the ratio of the total number of failures to the total operating time, i.e. failure rate in time (FIT), which represents the number of failures per device hours of accelerated stress tests.

The proposed EDDR architecture is synthesized by using TSMC 0.18 m 1P6M CMOS technology,1998 is given as the year of manufacturing for a wide variety of components. Thus, is defined as 12 years, because the year of manufacturing is 1998.

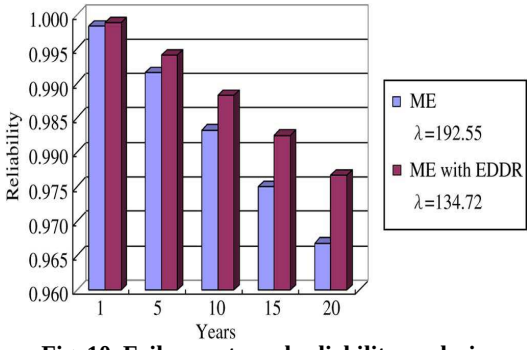


Fig. 10. Failure-rate and reliability analysis.

Fig. 10 clearly indicates that the low failure-rate and high reliability levels can be obtained if the proposed EDDR architecture is embedded into a ME for testing applications.

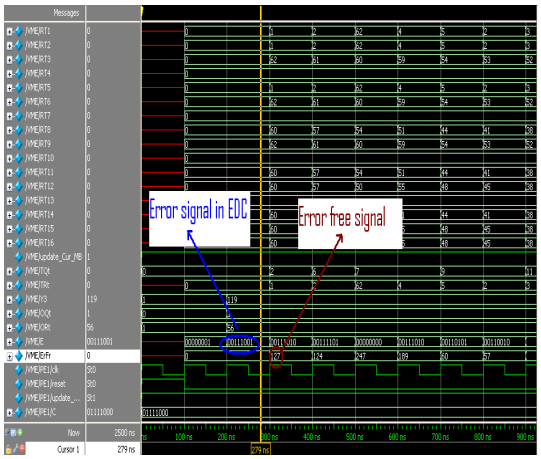
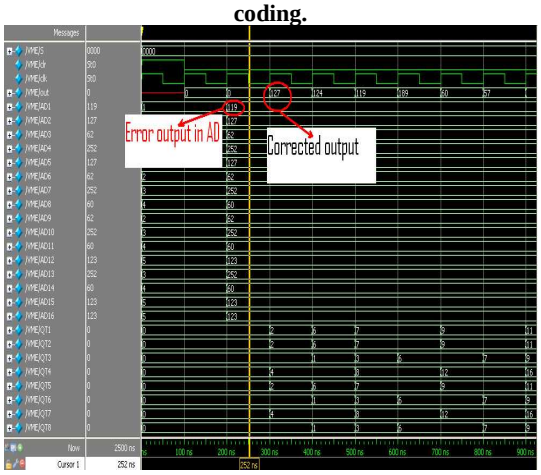


Fig (11)In the above fig we can observe that Error signal and Error free signal by using the residue and quotient coding.



Fig(12) In the above fig we can observe the Error output signal and corrected output signal.

Conclusion

The proposed EDDR architecture is implemented by using VHDL and synthesized by the Synopsys Design Compiler with TSMC 0.18-µm1P6 MCMOS technology. Experimental results indicate that that the proposed EDDR architecture can effectively detect errors and recover data in PEs of a ME with reasonable area overhead and only a slight time penalty. Throughput and reliability issues are also discussed.

References

[1] *Advanced Video Coding for Generic Audiovisual Services*, ISO/IEC 14496-10:2005 (E), Mar. 2005, ITU-T Rec. H.264 (E).

[2] 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.

[3] 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.

[4] S. Bayat-Sarmadi and M. A. Hasan, "On concurrent detection of errors in polynomial basis multiplication," *IEEE Trans. Vary Large Scale Integr. (VLSI) Systs.*, vol. 15, no. 4, pp. 413–426, Apr. 2007.

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

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