International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.6, No.9, September 2017 DOI:10.15693/ijaist/2017.v6i11.186-194

Data Processing for Video Applications

Dr.C.Arunabala
Professor &Head,Electronics and communications engineering,, Samskruti
College of engineering and technology-Ghatakesar-Hyderabad Email:
hodece@samskruti.ac.in,arunabala4700@gmail.com,Mob:9441782170

Abstract*:* Given the critical role of motion estimation (ME in video coder, 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 datarecovery(EDDR)design,basedontheresidueandquotient(RQ)code,to\embedintoMEforvideocodingtestinapplications.Anerrorinprocessingelements (PEs), i.e. key components of a ME, can be detected and recovered effectively by using the proposed EDDR design. Experimental results indicate that the proposed EDDR design for ME testing can detect errors and recover data with an acceptable area overhead and timing penalty. Importantly, the proposed EDDR design performs satisfactorily in terms of throughput and reliability for ME testing applications

Index Terms—**Area overhead, data recovery, error detection, motion estimation, reliability, residue**

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 is of priority concern in exploiting the temporal redundancy between successive frames, yet also the most time consuming aspect of coding. Additionally, while performing up to maximum computations encountered in the entire coding system, a ME is widely regarded as the most computationally intensive of a video coding system.

A ME generally consists of PEs with a size of 4x4. However, accelerating the computation speed depends on a large PE array, especially in high-resolution devices with a large search range such as HDTV. 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. A testable design is thus increasingly important to ensure the reliability of numerous PEs in a ME. Moreover, although 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).

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. For instance, assume that N denotes an integer N1 and N2, and 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, ǀNǀm). Notably, Nm 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 X is expressed as

$$X=\{b_{n-1}\ b_{n-2}\ldots\ldots b_2b_1b_0\}=\sum_{j=0}^{n-1} bj2j \qquad (1)$$

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:

$$|N_1+N_2|m=||N_1|m+|N_2|m|m. \qquad (2)$$

Definition 2: let $Nj=n_1+n_2+\ldots\ldots+n_j$, then

$$|N_j|m=||n_1|_m+|n_2|m+\ldots\ldots+|nj|m|m. \qquad (3)$$

To accelerate the circuit design of RQCG, the binary data
Shown in can generally be divided into two parts:

$$X=\sum_{j=0}^{n-1} bj2j$$

$$=\Big(\sum_{j=0}^{k-1} bj2j\Big)+\Big(\sum_{j=k}^{n-1} bj2j-k\Big)2^k$$

$$=Y_0+Y_12^k. \qquad (4)$$

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

$$R=|X|m$$

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

$$Q=\Big\lfloor \frac{X}{M} \Big\rfloor$$

$$=\Big\lfloor \frac{Y0+Y1}{M} \qquad (6)$$

$$=Z1+Y1+\beta \qquad (7)$$

Notably, since the value of Y_0+Y_1 is generally greater than that of modulus *m* , the equations in and must be simplified further to replace the complex module operation with a simple addition operation by using the parameters Z_0,Z_1,α and β Based on and , 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. PROPOSED EDDR ARCHITECTURE DESIGN

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 as a CUT to demonstrate the feasibility of the proposed EDDR architecture. A ME consists of many Reincorporated 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.1 Shows an example of the proposed EDDR circuit design for a specific 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.

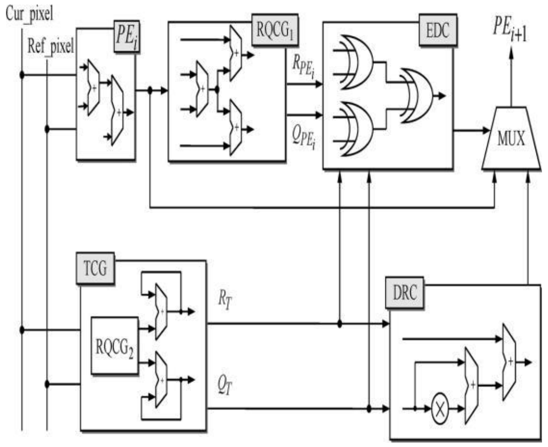


Fig.1 A specific _ testing processes of the proposed EDDR architecture.

A. 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.

B.TCG Design

In fig 1. TCG is an important component of the proposed EDDR architecture. This 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 PEi in fig 3.1 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 NXN can be evaluated:

$$SAD \quad {}_{i\ 0}^{N\ 1} \quad {}_{j\ 0}^{N\ 1} \left| X_{ij} \quad Y_{ij} \right|$$

$$ {}_{i\ 0 j\ 0}^{N\ 1\ \ N\ 1} \left| q_{xij} \quad m \quad r_{xij} q_{yij} \quad m \quad r_{yij} \right| \qquad (8)$$

Wherer_{xij} , q_{xij} and r_{yij} , q_{yij} denote the corresponding RQ code of X_{ij} and Y_{ij} modulo m. Importantly, X_{ij} and Y_{ij} represent the luminance pixel value of Cur_pixel and Ref_pixel. Respectively. Based on the residue code, the definitions shown in (2) and (3) can be applied to facilitate generation of the RQ code R_T *and* Q_T from TCG. Normally, the circuit design of TCG can be easily achieved (see Fig.3.2) by using

$$R_T \quad \left| {}_{i\ 0}^{N\ 1} \quad {}_{j\ 0}^{N\ 1} \quad X_{ij} \quad Y_{ij} \right|_m$$

$$\left| \left| X_{00} \quad Y_{00} \right|_m \quad \left| X_{01} \quad Y_{01} \right|_m \quad ..... \quad \left| X_{N\ 1\ \ N\ 1} \quad Y_{N\ 1\ \ N\ 1} \right|_m \right|_m$$

$$\left| q_{x00} \ m \ r_{x00} \quad q_{y00} \ m \ r_{y00} \right|_m \quad .....$$

$$\left| \left| q_{xN1N1} \quad m \ r_{xN1N1} \quad q_{y\ N\ 1\ N\ 1} \quad m \ r_{y\ N\ 1\ N\ 1} \right|_m \right|_m$$

$$\left| \left| r_{x00} \quad r_{y00} \right|_m \quad \left| r_{x01} \quad r_{y01} \right|_m \quad .... \quad \left| r_{x\ N\ 1\ \ N\ 1} \quad r_{y\ N\ 1\ \ N\ 1} \right|_m \right|_m$$

$$\left| r_{00} \right|_m \quad \left| r_{01} \right|_m \quad .... \quad \left| r_{N\ \ 1} \right|_{N\ \ 1} \ {}_{m\ m}$$

(9)

And Q_T calculated as follows

$$Q_T \quad \frac{{}^{N\ 1\quad N\ 1}_{i\ 0\quad j\ 0}\quad X_{ij}\ Y_{ij}}{m}$$

$$\frac{X_{00}\quad Y_{00}\ X_{01}\quad Y_{01}\ \ldots..\quad X_{N\,1\,N\,1}\quad Y_{N\,1\,N\,1}}{m}$$

$$\frac{q_{x00}\ m\ q_{y00}\ m}{m}\ \frac{r_{x00}\ r_{y00}}{m}\ \frac{q_{x01}\ m\ q_{y01}\ m}{m}\ \frac{r_{x01}\ r_{y01}}{m}\ \ldots$$

$$q_{x00}\ q_{y00}\ q_{x01}\ q_{y01}\ \ldots.\ \frac{r_{x00}\ r_{y00}\ r_{x01}\ r_{y01}\ \ldots.}{m}$$

$$q_{00}\ q_{01}\ \ldots..\ q_{N\,1\,N\,1}\quad \frac{r_{00}\ r_{01}\ \ldots.\ r_{N\,1\,N\,1}}{m} \qquad (10)$$

1. Circuit Design of the TCG

Fig 2 shows the circuit design of TCG. 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 first clock. Notably, if $X_{ij} \geq Y_{ij}$, then X_{ij} and Y_{ij} 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 $X_{ij} < Y_{ij}$.

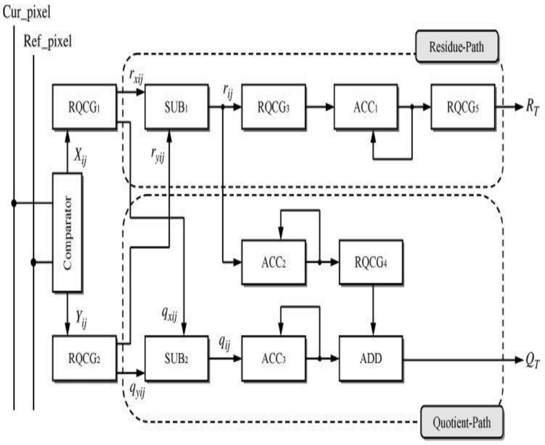


Figure 2: Circuit design of the TCG

C.EDDR PROCESS

The figure 3.1 shows clearly indicates that the operations of error detection in a specific PE_i is achieved by using EDC, which is utilized to compare the outputs between TCG and $RQCG_1$ in order to determine whether errors have occurred. If the values of $R_{Pei} \neq R_T$ and/or $Q_{PE}i \neq Q_T$, then the errors in a specific PE_i can be detected. The EDC output is then used to generate a 0/1 signal to indicate that the tested PE_i is error-free/errancy. This work presents a mathematical statement to verify the operations of error detection. Based on the definition of the fault model, the SAD value is influenced if either SA1 and / or SA0 errors have occurred in a specific PE_i. In other words, the SAD value is transformed to SAD' = SAD+e if an error e occurred. Notably, the error signal e is expressed as

$$e\quad q_e\quad m\quad r_e \qquad (11)$$

To comply with the definition of RQ code. Under the faulty case, the RQ code form $RQCG_2$ of the TCG is still equal to (8) and (9). However,

R_{PE_i} and Q_{PE_i} are changed to (13) and (14) because an error e has occurred. Thus, the error in a specific PE_i can be detected if and only if

(8) (11) and/or (9) (12)

$$R_{PE_i}\quad \left|SAD'\right|_m$$

$$\left|{}^{N\ 1\quad N\ 1}_{i\ 0\quad j\ 0}\quad X_{ij}\quad Y_{ij}\quad e\right|_m$$

$$\left|r_{00}\right|_m\quad \left|r_{01}\right|_m\quad \ldots.\quad \left|r_{N\ 1\ N\ 1}\right|_m\quad \left|\ \right|\quad \left|r_e\right|_m{}_m \qquad (12)$$

$$Q_{PE_i}\quad \frac{SAD'}{m}$$

$$\frac{{}^{N\ 1\quad N\ 1}_{i\ 0\quad j\ 0}\quad X_{ij}\quad Y_{ij}\quad e}{m}$$

$$q_{00}\quad q_{01}\quad \ldots..\quad q_{N\,1\ N\,1}\quad q_e\quad \frac{r_{00}\quad r_{01}\quad \ldots.\ r_{N\,1\ N\,1}\quad r_e}{m}$$

(13)

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

$$SAD\quad m\quad Q_T\quad R_T$$

$$2^j\quad 1\quad Q_T\quad R_T$$

$$2^{j} \quad QQR_{TTT} \qquad (14)$$

To realize the operation of data recovery in , a Barrel shift and a corrector circuits are necessary to achieve the functions of $2^{j} \quad Q_T$ and $Q_T \quad R_T$, respectively. Notably, the proposed EDDR design executes the errors detection and data recovery operations simultaneously. Additionally, error-free data from the tests PE_i or the data recovery that results from DRC is selected by a multiplexer (MUX0 to pass to the next specific PE_{i+1} for subsequent testing.

D.NUMERICAL EXAMPLE OF PIXEL VALUES

	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

Figure 4.1: Current Pixel and Reference Pixel values

Numeric Calculation

A numerical example of the 16 pixels for a 4X 4 macro block in a specific PEi of a ME is described .We presents an example of pixel values of the Cur_pixel and Ref_pixel. Based on, the SAD value of the 4X 4 macro block is

$$SAD \quad \sum_{i\,0}^{3} \sum_{j\,0}^{3} \left| X_{ij} \; Y_{ij} \right|$$

$$X\,|_{00} \quad Y_{00} \;|\; |X_{01} \quad Y_{01}| \quad .... \;|X_{33} \quad Y|_{33}$$

$$128 \;\; 1 \quad 128 \;\; 1 \quad .... \quad 128 \;\; 5$$

$$= 2124$$

According to describe about RQ Code for modulo operation is assumed M=2^6-1=63 RQ Code for SAD value

$$R_T \;\; R_{PE_i} \quad \left| \frac{2124}{63} \right| \quad 45$$

and $QQ_{T \;\; PE_i} \left| \frac{2124}{63} \right| 33.$ Since the value of $R_T \; Q_T$ is equal to $R_{PE_u} \;\; Q_{PE_i}$, EDC is enabled and a signal “0” is generated to describe a situation in which the specific PE_i is error-free. Conversely, if SA1 and SA0 errors occur in bits 1 and 12 of a specific PE_i, i.e., the pixel values of PE_i $2124=100001001100_2$ is turned into $77=$**0**0000100110**1**$_2$, resulting in a transformation of theRQ code R_{PE_u} *and* Q_{PE_i} into $|77|_{63} \;\; 14$ and $\left|\frac{77}{63}\right| \; 1$. Thus, an error signal “1” is generated from EDC and sent to the MUX in order to select the recovery results from DRC.

E. OVERALL TEST STARTEGY

Figure 3.3 shows that the proposed EDDR architecture design for a motion estimation. 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. Mean while, the corresponding test codes R_{Ti} and Q_{Ti} from a specific TCG_i are selected simultaneously by MUXs 2 and 3 respectively. Third, the RQ code from TCG_i and RQCG circuits are compared in EDC to determine whether the tested object PE_i have errors. The tested object PE_i is error-free if and only if $R_{PEi} = R_{Ti}$ and $Q_{PEi} = Q_{Ti\,.}$ Additionally, DRC is used to recover data encoded by TCG_i, i.e. the appropriate R_{Ti} and Q_{Ti} codes from TCG_i, are selected by MUXs 2 and 3, respectively to recover data. Fourth, the error-free data or data recovery results are selected by MUX_4. Notably the 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.

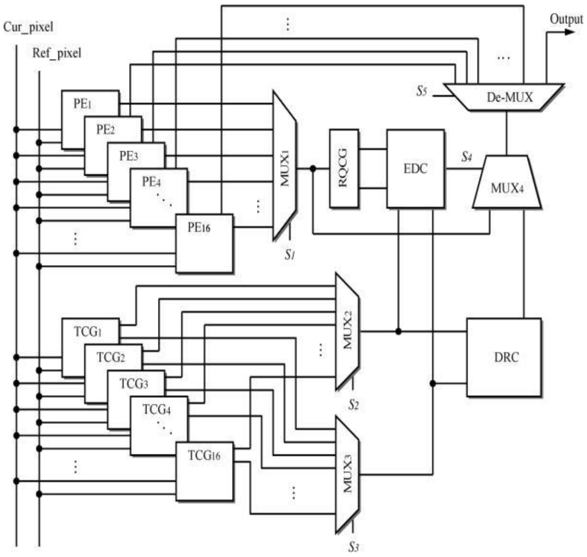


Figure 3: Proposed EDDR architecture design for a ME

IV RESULTS AND DISCUSSION

A.RESULTS ANALYSIS

Extensive 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.

Table1: Estimation of Area Overhead and Time Penalty

Component	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 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 as shown in the equations (15), (16) and (17). The time penalty is another criterion to verify the feasibility of the proposed EDDR architecture. Table 7.1 also summarizes the operating time evaluation of a specific PEi and each component in the proposed EDDR architecture. The equations (7.4) and (7.5) shows the time penalty of error detection and data recovery (TP$_{ED}$ and TP$_{DR}$) operations for a 4x4 macro block (a PEi with 16 pixels).

$$AO_{ED} \quad \frac{1779\ \ 3265\ \ 16\ \ 667}{69482\ \ 16} \quad 4.92\% \qquad (15)$$

$$AO_{DR} \quad \frac{3265\ \ 16\ \ 2376}{69482\ \ 16} \quad 4.91\% \qquad (16)$$

$$AO_{EDDR} \quad \frac{1779\ \ 667\ \ 3265\ \ 16\ \ 2376}{69482\ \ 16} \quad 5.13\% \qquad (17)$$

$$TP_{ED} \quad \frac{1016.56\ \ 6.02\ \ 973.76}{973.76} \quad 5.01\% \qquad (18)$$

$$TP_{DR} \quad \frac{1016.56\ \ 17.99\ \ 973.76}{973.76} \quad 6.24\% \qquad (19)$$

Also each PE of a ME is tested sequentially in the proposed EDDR architecture. Thus if it 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.

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

B. PERFORMANCE DISCUSSION

The TCG component plays a major role in the EDDR architecture to detect errors and recover data. Additionally, the number of TCGs significantly influences the circuit performance in terms of area overhead and throughput. The figures 7.1 and 7.2. Illustrate the relations between the number of TCGs, area overhead and throughput. The area overhead is less than 2% only if only one TCG is used to execute; However at this time the throughput is extremely small. Notably, the throughput of a ME with out embedding the EDDR architecture is about 25800 k MB/s. Fig: 7.2. Clearly indicates that the throughput is around 25000k MB/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 ME testing applications. Although the area overhead is increased if 16 TCGs used (see fig 7.1) the area overhead is only about 5.13%, i.e. an acceptable design for circuit testing.

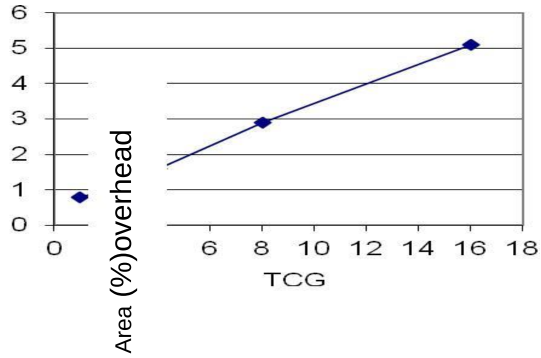


Figure7.1: Relation between TCG and area overhead

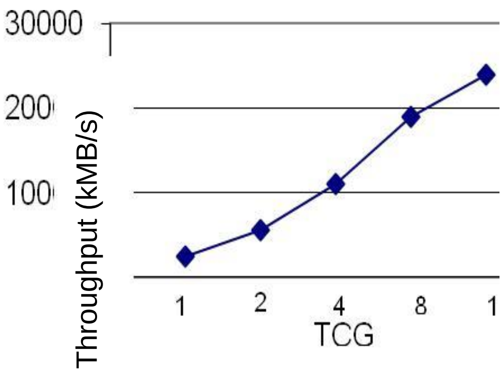


Figure7.2: Relation between TCG and throughput

This work also addresses reliability-related issues to demonstrate the feasibility of the 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.The failure-rate 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 10^9 device hours of accelerated stress tests.

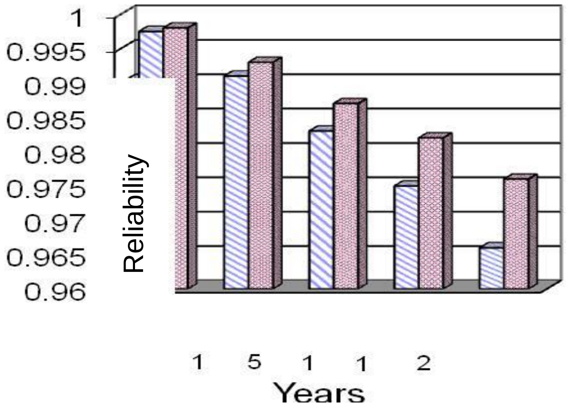


Figure 7.3: Failure-rate and reliability analysis.

The figure 7.3 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.

C.SIMULATION RESULTS

RESULTS:

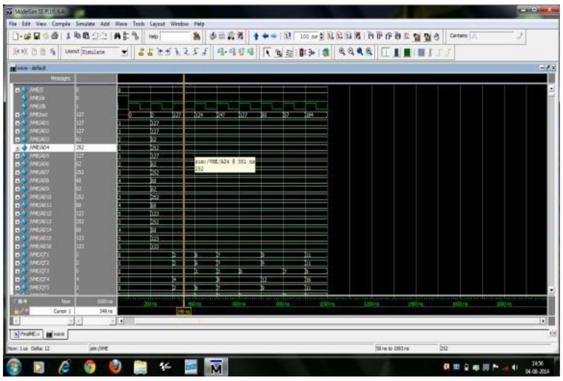


Figure 8.1: Simulation Result of Motion Estimation

SYNTHESIS RESULTS

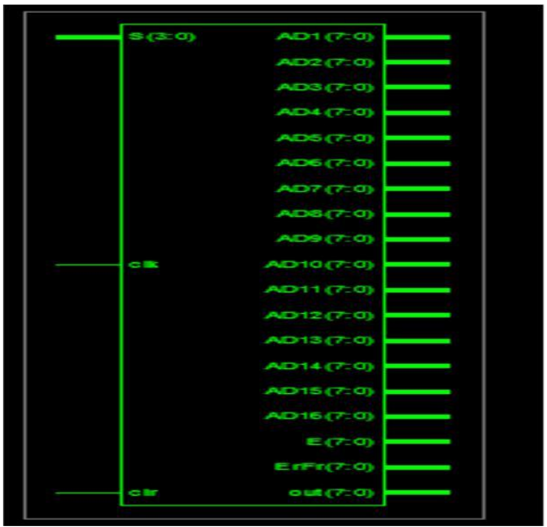


Figure8.2: Schematic Diagram of Motion Estimation

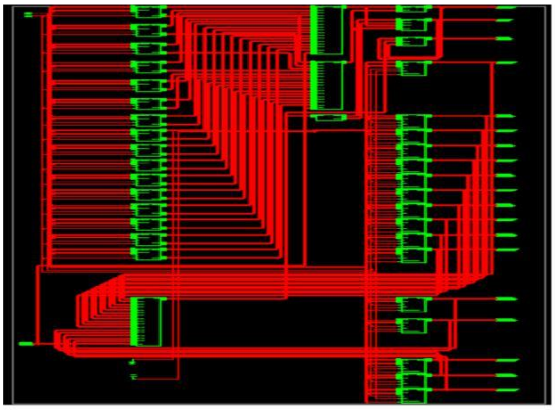


Figure8.3: Internal Diagram of Motion Estimation

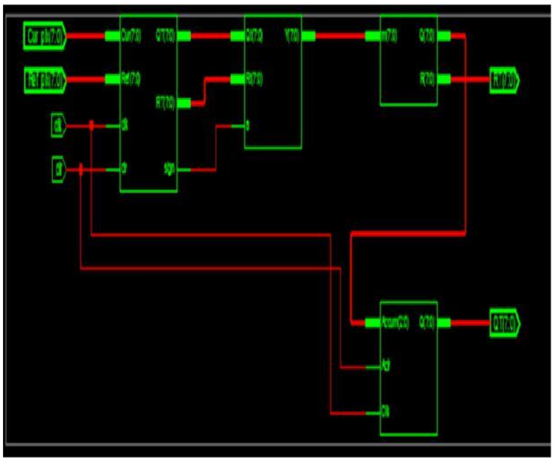


Figure 8.4: Diagram of Test Code Generator

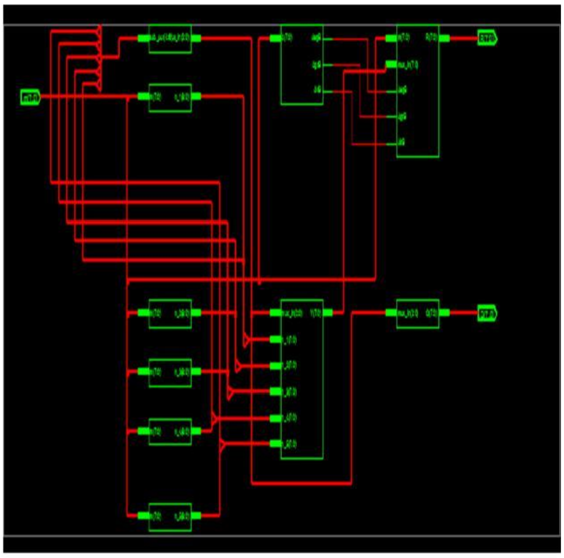


Figure 8.5: Diagram of Residue-Quotient (RQ) Code Generator

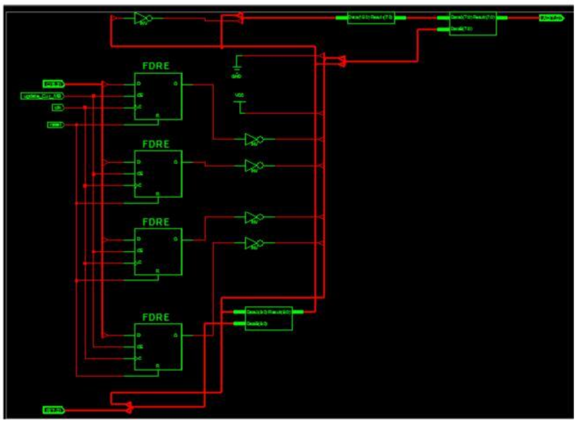


Figure 8.6: Diagram of Processing Element

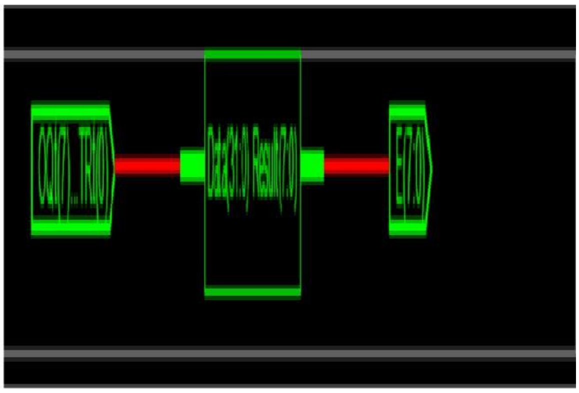


Figure 8.7: Diagram of Error Detection Circuit

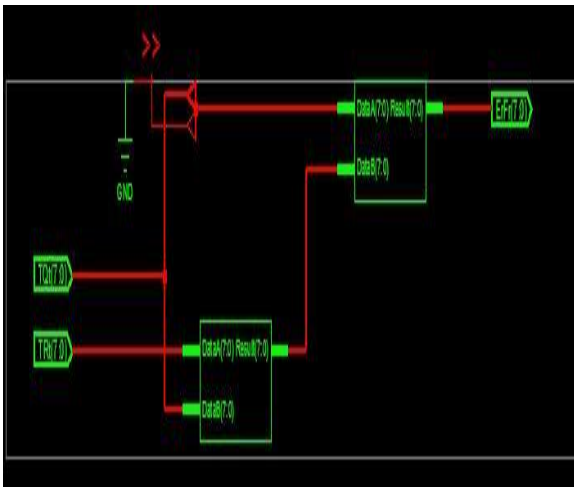
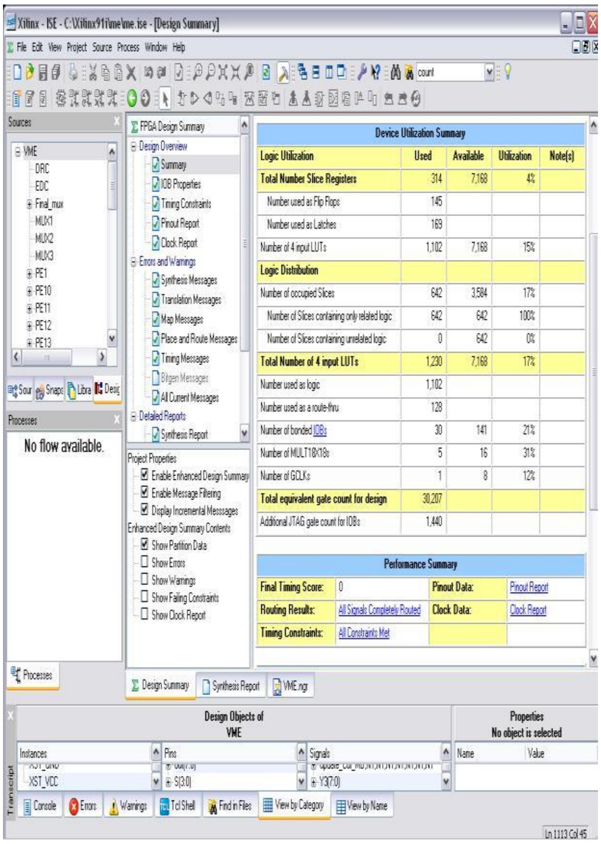


Figure 8.8 : Diagram of Data Recovery Circuit

DESIGN SUMMARY


Xilinx - ISE - C:\Xilinx91i\me\me.ise - [Design Summary]
File Edit View Project Source Process Window Help

No flow available.

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Total Number Slice Registers	314	7,168	4%	
Number used as Flip Flops	145			
Number used as Latches	169			
Number of 4 input LUTs	1,102	7,168	15%	
Logic Distribution				
Number of occupied Slices	642	3,584	17%	
Number of Slices containing only related logic	642	642	100%	
Number of Slices containing unrelated logic	0	642	0%	
Total Number of 4 input LUTs	1,230	7,168	17%	
Number used as logic	1,102			
Number used as a route-thru	128			
Number of bonded IOBs	30	141	21%	
Number of MULT18X18s	5	16	31%	
Number of GCLKs	1	8	12%	
Total equivalent gate count for design	30,207			
Additional JTAG gate count for IOBs	1,440			

Performance Summary			
Final Timing Score:	0	**Pinout Data:**	Pinout Report
Routing Results:	All Signals Completely Routed	**Clock Data:**	Clock Report
Timing Constraints:	All Constraints Met		

Design Objects of VME

Properties
No object is selected

V CONCLUSION

This work presents Error Detection and Data Recovery (EDDR) architecture for detecting the errors and recovering the data of Processing Elements (PEs) in a ME. Based on the Residue-and-Quotient (RQ) code, a RQCG-based TCG design is developed to generate the corresponding test codes to detect errors and recover data. The proposed EDDR architecture is also implemented by using VERILOG and synthesized by Xilinx ISE. Experimental results indicate 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 to demonstrate the satisfactory performance of the proposed EDDR architecture design for ME testing applications

REFERENCES

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

2) *Information Technology-Coding of Audio-Visual Objects—Part 2: Visual*, ISO/IEC 14 496-2, 1999.

3) Y. W. Huang, B. Y. Hsieh, S. Y. Chien, S. Y. Ma, and L. G. Chen, "Analysis and complexity reduction of multiple reference framesmotion estimation in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 4, pp. 507–522, Apr. 2006.

4) Y. Chen, S. Y. Chien, Y. W. Huang, T. C. Chen, T. C. Wang, and L. G. Chen, "Analysis and architecture design of variable block-size motion estimation for H.264/AVC," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 3, pp. 578–593, Mar. 2006.

5) T. H. Wu, Y. L. Tsai, and S. J. Chang, "An efficient design-for-testability scheme for motion estimation in H.264/AVC," in *Proc. Int. Symp. VLSI Design, Autom. Test*, Apr. 2007, pp. 1–4.

6) M. Y. Dong, S. H. Yang, and S. K. Lu, "Design-for-testability techniques for motion estimation computing arrays," in *Proc. Int. Conf.Commun. Circuits Syst.*, May 2008, pp. 1188–1191.

7) Y. S. Huang, C. J. Yang, and C. L. Hsu, "C-testable motion estimation design for video coding systems," *J. Electron. Sci. Technol.*, vol. 7, no. 4, pp. 370–374, Dec. 2009.

8) Li, M. Hu, and O. A. Mohamed, "Built-in self-test design of motion estimation computing array," in *Proc. IEEE Northeast Workshop CircuitsSyst.*, Jun. 2004, pp. 349–352.

9) 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.*, Dec. 2008, pp. 1751–1754.

10) W. Y Liu, J. Y. Huang, J. H. Hong, and S. K. Lu, "Testable design and BIST techniques for systolic motion estimators in the transformdomain," in *Proc. IEEE Int. Conf. Circuits Syst.*, Apr. 2009, pp. 1–4.

11) J. M. Portal, H. Aziza, and D. Nee, "EEPROM memory: Threshold voltage built in self diagnosis," in *Proc. Int. Test Conf.*, Sep. 2003, pp. 23–28.

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