High Speed Error Detection and Data Recovery Architecture for Video Applications

D. Kranthi Kumar, T. Mahaboob Doula
P. G. Student scholar M. Tech (VLSI) Department of ECE SKDEC, GOOTY
Assistant professor, M. E. Department of ECE SKDEC, GOOTY

Abstract
Given the critical role of motion estimation (ME) in a 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 data recovery (EDDR) design, based on the residue-and-quotient (RQ) code, to embed into ME for video coding testing applications. An error in processing elements (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-and-quotient (RQ) code.

1. 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 [1], [2]. 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 60%-90% of the computations encountered in the entire coding system, a ME is widely regarded as the most computationally intensive of a video coding system [3].

A ME generally consists of PEs with a size of 4 x 4. However, accelerating the computation speed depends on a large PE array, especially in high-resolution devices with a large search range such as HDTV [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. 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) [5]–[7]. 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 for the ME does not expensive test equipment, ul- timately lowering test costs [8]–[10]. Moreover, BIST can generate test simulations and analyze test responses without outside support, subsequently streamlining the testing and diagnosis of digital systems. However, increasingly complex density of circuitry requires that 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 [11] and built-in self-correction [12]–[14] have been developed recently.

While the extended BIST schemes generally focus on memory circuit, testing-related issues of video coding have been seldom 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 PEs in a ME can be improved by enhancing the capabilities of concurrent error detection (CED) [15], [16]. 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 a novel EDDR architecture based on the RQ code to detect errors and recovery data in PEs of a ME and, in doing so, further guarantee the excellent reliability for video coding testing applications. The rest of this paper is organized as follows. Section II describes the mathematical model of RQ code and the corresponding circuit design of the RQ code generator (RQCG). Section III then introduces the proposed EDDR architecture, fault model definition, and test method. Next, Section IV evaluates the performance in area overhead, timing penalty, through put and reliability analysis to demonstrate the feasibility of the proposed EDDR architecture for ME testing applications. Conclusions are finally drawn in Section V.

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 [17], [18]. 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 \( N \) is an integer, \( N_1 \) 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 \( [N_1 | N_2]_m \). Notably, \( |N|_m \) is the residue of 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 can not 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 data. The mathematical model of RQ code is simply described as follows. Assume that binary data \( X \) is expressed as

\[
X = \{b_{n-1}b_{n-2} \ldots b_2b_1b_0\} = \sum_{j=0}^{n-1} b_j2^j. \tag{1}
\]

The RQ code of \( X \) modulo \( m \) expressed as \( R = |X|_m \) \( Q = \left| X/m \right| \), 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:**

\[
|N_1 + N_2|_m = |N_1|_m + |N_2|_m. \tag{2}
\]

**Definition 2:** Let \( N_j = n_j + n_{j+1} + \ldots + n_k \) then

\[
|N_j|_m = |n_j|_m + |n_{j+1}|_m + \ldots + |n_k|_m. \tag{3}
\]

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

\[
X = \sum_{j=0}^{n-1} b_j2^j = \left( \sum_{j=0}^{n-1} b_j2^j \right) + \left( \sum_{j=n}^{n-1} b_j2^j \right) = Y + Y'. \tag{4}
\]

Significantly, the value of \( k \) is equal to \( \lfloor n/2 \rfloor \) and the data formation of \( Y \) and \( Y' \) are a decimal system. If the modulus \( m = 2^k \), then the residue code of \( X \) modulo \( m \) is given by

\[
R = \left| X \right|_m = \left| Y + Y' \right|_m = \left| Z_1 + Z_2 \right|_m = \left( Z_1 + Z_2 \right) c. \tag{5}
\]

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

\[
Q = \left[ \frac{X}{m} \right] = \frac{Z_1 + Z_2}{m} + \frac{Z_1 + Z_2}{m} = \frac{Z_1 + Z_2}{m} + \frac{Z_1 + Z_2}{m}. \tag{6}
\]

where

\[
(\alpha, \beta) = \begin{cases} 0(1) & \text{if } Z_1 + Z_2 < m. \\ 0(0) & \text{otherwise.} \end{cases}
\]

Notably, since the value of \( Z_1 + Z_2 + \alpha \) 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 \( Z_0, Z_1, c \) and \( f \). 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. 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 RQC 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 TCG 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.

![Diagram](image)

**Fig. 2. A specific PEi testing process of the proposed EDDR architecture.**

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 [20]. 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 of a ME. The fault model definition, RQC-based TCG design, operations of error detection and data recovery, and the overall test strategy are described carefully as follows.

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) [21]. Using CFM has received considerable interest due to acceleration 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 [22]. 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.

B. 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 ROQC circuit to generate corresponding test codes in order to detect errors and recover data. The specific FEi in Fig. 2 estimates the absolute difference between the Cur_pixel of the search area and the Ref_pixel of the current macroblock. Thus, by utilizing PEs, SAD shown in as follows, in a macroblock with size of $N \times N$ can be evaluated:
where $r_{ij}(x_{ij})$ and $r_{ij}(y_{ij})$ 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 the generation of the RQ code ($R_{T}$ and $Q_{T}$) from TCG. Namely, the circuit design of TCG can be easily achieved (see Fig. 3) by using

\[ R_{T} = \left| \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (X_{ij} - Y_{ij}) \right| \]

\[ = \left| (X_{00} - Y_{00}) + (X_{01} - Y_{01}) + \ldots \right| \]

\[ + \left| (X_{(N-1)(N-1)} - Y_{(N-1)(N-1)}) \right| \]

\[ = \left| r_{x_{00}} \cdot m + r_{x_{01}} \right| + \left| r_{y_{00}} \cdot m + r_{y_{01}} \right| \]

\[ + \ldots \left| r_{x_{(N-1)(N-1)}} \cdot m + r_{y_{(N-1)(N-1)}} \right| \]

\[ = \left| r_{x_{00}} \cdot m + r_{y_{00}} \right| + \left| r_{x_{01}} \cdot m + r_{y_{01}} \right| + \ldots \]

\[ + \left| r_{x_{(N-1)(N-1)}} \cdot m + r_{y_{(N-1)(N-1)}} \right| \]

\[ = \left| r_{x_{00}} \cdot m + r_{y_{00}} \right| + \left| r_{x_{01}} \cdot m + r_{y_{01}} \right| + \ldots \]

and (9), shown at the bottom of the following page, to derive the corresponding RQ code.

Fig. 4 shows the timing chart for a macroblock with a size of $4 \times 4$ in a specific $FE$, to demonstrate the operations of TCG circuits. The data $m_{0}$ and $m_{1}$ 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. Namely, if $X_{ij} \geq Y_{ij}$, then $Y_{ij}$ is the luminance pixel value of Cur_pixel.

![Fig. 3. Circuit design of the TCG.](image)

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 $Y_{00}$ are generated and the corresponding RQ code $r_{x_{00}}$ and $r_{y_{00}}$ can be captured by the RQC G1 and RQC G2 circuits if the 3rd clock is triggered. Equations (8) and (9) clearly indicate that the codes of $r_{x_{00}}$ and $r_{y_{00}}$ can be obtained by using the circuit of a subtractor (SUB). The 4th clock displays the operating results. The modulus value of $F_{E}$ is then obtained at the 5th clock. Next, the summation of quotient values and residue values of modulo are proceeded with from clocks 5-21 through the circuits of ACCs. Since a $4 \times 4$ macroblock in a specific $FE$ 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 $FE$ in a ME can be achieved.

![Fig. 3. Circuit design of the TCG.](image)

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 $Y_{00}$ are generated and the corresponding RQ code $r_{x_{00}}$ and $r_{y_{00}}$ can be captured by the RQC G1 and RQC G2 circuits if the 3rd clock is triggered. Equations (8) and (9) clearly indicate that the codes of $r_{x_{00}}$ and $r_{y_{00}}$ can be obtained by using the circuit of a subtractor (SUB). The 4th clock displays the operating results. The modulus value of $F_{E}$ is then obtained at the 5th clock. Next, the summation of quotient values and residue values of modulo are proceeded with from clocks 5-21 through the circuits of ACCs. Since a $4 \times 4$ macroblock in a specific $FE$ 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 $FE$ in a ME can be achieved.

![Fig. 3. Circuit design of the TCG.](image)

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 $Y_{00}$ are generated and the corresponding RQ code $r_{x_{00}}$ and $r_{y_{00}}$ can be captured by the RQC G1 and RQC G2 circuits if the 3rd clock is triggered. Equations (8) and (9) clearly indicate that the codes of $r_{x_{00}}$ and $r_{y_{00}}$ can be obtained by using the circuit of a subtractor (SUB). The 4th clock displays the operating results. The modulus value of $F_{E}$ is then obtained at the 5th clock. Next, the summation of quotient values and residue values of modulo are proceeded with from clocks 5-21 through the circuits of ACCs. Since a $4 \times 4$ macroblock in a specific $FE$ 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 $FE$ in a ME can be achieved.

![Fig. 3. Circuit design of the TCG.](image)

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 $Y_{00}$ are generated and the corresponding RQ code $r_{x_{00}}$ and $r_{y_{00}}$ can be captured by the RQC G1 and RQC G2 circuits if the 3rd clock is triggered. Equations (8) and (9) clearly indicate that the codes of $r_{x_{00}}$ and $r_{y_{00}}$ can be obtained by using the circuit of a subtractor (SUB). The 4th clock displays the operating results. The modulus value of $F_{E}$ is then obtained at the 5th clock. Next, the summation of quotient values and residue values of modulo are proceeded with from clocks 5-21 through the circuits of ACCs. Since a $4 \times 4$ macroblock in a specific $FE$ 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 $FE$ in a ME can be achieved.

![Fig. 3. Circuit design of the TCG.](image)

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 $Y_{00}$ are generated and the corresponding RQ code $r_{x_{00}}$ and $r_{y_{00}}$ can be captured by the RQC G1 and RQC G2 circuits if the 3rd clock is triggered. Equations (8) and (9) clearly indicate that the codes of $r_{x_{00}}$ and $r_{y_{00}}$ can be obtained by using the circuit of a subtractor (SUB). The 4th clock displays the operating results. The modulus value of $F_{E}$ is then obtained at the 5th clock. Next, the summation of quotient values and residue values of modulo are proceeded with from clocks 5-21 through the circuits of ACCs. Since a $4 \times 4$ macroblock in a specific $FE$ 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 $FE$ in a ME can be achieved.

![Fig. 3. Circuit design of the TCG.](image)
\[ Q_T = \left[ \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (X_{ij} - Y_{ij}) \right] / m \]

\[ = \left( \frac{q_{00}m + q_{01}m + \ldots + q_{N-1,0}m + q_{N-1,N-1}m}{m} \right) \]

\[ = \left( \frac{r_{00}m + r_{01}m + \ldots + r_{N-1,N-2}m + r_{N-1,N-1}m}{m} \right) \]

\[ = q_{00} + q_{01} + \ldots + q_{N-1,N-1} + \frac{r_{00} + r_{01} + \ldots + r_{N-1,N-1}}{m} \]  \hspace{1cm} (9)

\[ Q_{FE} = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |X_{ij} - Y_{ij}| + e}{m} \]

\[ = \frac{|r_{00}| + |r_{01}| + \ldots + |r_{N-1,N-1}| + e}{m} \]

\[ = q_{00} + q_{01} + \ldots + q_{N-1,N-1} + q_{0} \]

\[ + \frac{r_{00} + r_{01} + \ldots + r_{N-1,N-1}}{m} \]  \hspace{1cm} (11)

During data recovery, the circuit DRC plays a significant role in recovering RQ code from ICG. The data can be recovered by implementing the mathematical model as:

\[ SAD = m \times Q_{T} + R_{T} \]

\[ = (2^l - 1) \times Q_{T} + R_{T} \]

\[ = 2^l \times Q_{T} - Q_{T} + R_{T} \]  \hspace{1cm} (12)

To make the operation of data recovery in (13), a Barrel shift [23] and a corrector circuits are necessary to achieve the functions of \[ (2^l \times Q_{T}) \] 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 FE, or the data recovery that results from DRC is selected by a multiplexer (MUX) to pass to the next specific FE, for subsequent testing.
D. Numerical Example

A numerical example of the 16 pixels for a 4X4 macroblock in a specific $FE_i$ of a ME is described as follows. Fig. 5 presents an example of pixel values of the Cur_pixel and Ref_pixel. Based on (7), the SAD value of the $4 \times 4$ macroblock is

$$SAD = \sum_{i=0}^{3} \sum_{j=0}^{3} |X_{ij} - Y_{ij}|$$
$$= |X_{00} - Y_{00}| + |X_{01} - Y_{01}| + \ldots + |X_{33} - Y_{33}|$$
$$= (128 - 1) + (128 - 1) + \ldots + (128 - 5)$$
$$= 2124.$$

(14)

According to the description of RQ code in Section II, the modulo is assumed here to be $m = 2^5 - 1 = 63$. Thus, based on (8) and (9), the RQ code of the SAD value shown in (14) are $R_T = R_{PE} = [2124/63] = 45$ and $Q_T = Q_{PE} = [2124/63] - 3$. Since the value of $R_T/Q_T$ is equal to $R_{PE}/Q_{PE}$, EDC is enabled and a signal “0” is generated to describe a situation in which the specific $FE$ is error-free. Conversely, if SA1 and SA0 errors occur in $FE$, $FE$

bits 1 and 12 of a specific , i.e. the pixel values of 

2124 = 1000010110102 is turned into 77 = 0000010011012, resulting in a transformation of the RQ code of $R_{PE}$ and $Q_{PE}$ into $[77/63] - 1$ 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.
E. Overall Test Strategy

By extending the testing process of a specific \( I_E \) in Fig. 2, Fig. 5 illustrates the overall EDDR architecture design of a ME. First, the input data of \( O_E \) and \( E_E \) are sent simultaneously to PEs and TCGs in order to estimate the SAD values and generate the test EQ code \( R_T \) and \( Q_T \). Second, the EQ value from the tested object \( I_E \), which is selected by MUX1, is then sent to the EQC circuit in order to generate \( R_{PE} \) and \( Q_{PE} \) codes. Meanwhile, the corresponding test codes \( R_T \) and \( Q_T \) from a specific EQC are selected simultaneously by MUX2 and 3, respectively. Third, the EQ code from TCG, and EQC circuits are compared in EDC to determine whether the tested object \( I_E \) have errors. The tested object \( I_E \) is error-free if and only if \( R_{PE} = R_T \) and \( Q_{PE} = Q_T \). Additionally, DRC is used to recover data encoded by TCG, i.e., the appropriate \( R_T \) and \( Q_T \) codes from TCG are selected by MUX2 and 3, respectively, to recover data. Finally, the error-free data or data recovery results are selected by MUX4. Notably, control signal \( S_E \) is generated from EDC, indicating that the comparison result is error-free \( (S_E = 0) \) or \( S_E = 1 \) otherwise. The error-free data or the data recovery result from the tested object \( I_E \) is passed to a De-MUX, which is used to test the next specific \( I_E \), otherwise, the final result is exported.

IV. RESULTS AND DISCUSSION

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.

A. Experimental Results

Table 1 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 \( AO_{EDRD} \) are

\[
AO_{PE} = \frac{1779 + 3265 \times 16 + 667}{69482 \times 16} = 4.92\% \quad (15)
\]

\[
AO_{DR} = \frac{3265 \times 16 + 237\ell}{69482 \times 16} = 4.9% \quad (16)
\]

\[
AO_{EDRD} = \frac{1779 + 3265 \times 16 + 667 + 237\ell}{69482 \times 16} = 5.13% \quad (17)
\]

The time penalty is another criterion to verify the feasibility of the proposed EDDR architecture. Table 1 also summarizes the operating time evaluation of a specific \( I_E \) and each component in the proposed EDDR architecture. The following equations show the time penalty of error detection and data recovery \( T_{PE} \) and \( T_{DR} \) operations for a \( 4 \times 4 \) macroblock (a \( I_E \) with 16 pixels).

<table>
<thead>
<tr>
<th>Components</th>
<th>PE</th>
<th>EQC</th>
<th>EDC</th>
<th>TCG</th>
<th>DRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area (Gate counts)</td>
<td>69482</td>
<td>1779</td>
<td>667</td>
<td>3265</td>
<td>2376</td>
</tr>
<tr>
<td>Operation Time (ns)</td>
<td>973.76</td>
<td>10.17</td>
<td>6.02</td>
<td>1016.56</td>
<td>17.99</td>
</tr>
<tr>
<td>Area Overhead (%)</td>
<td>5.13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time Penalty (%)</td>
<td>6.24</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: ESTIMATION OF AREA OVERHEAD AND PENALTY

Notably, each PE of a ME is tested sequentially in the proposed EDDR architecture. Thus, 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., approximatel y about 5.01% and 6.24% time penalty of the operations of error detection and data recovery, respectively.

Notably, the operating time of the EQC circuit can be neglected to evaluate because TCG covers the operating time of EQC. 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 \( I_E \), because the operating time of the tested object \( I_E \) is faster than the results of data recovery from DRC.

B. Performance Discussion

The TCG component plays a major role in the proposed 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. Figs. 7 and 8 illustrate the relations between the number of TCGs, area overhead 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. 8 clearly indicates that the throughput is around 25 000 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. 7), 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 [24]. The constant failure-rate reliability model

\[
R(t) = e^{-\lambda t}
\]

is used to estimate the reliability of the proposed EDDR architecture for ME testing applications, where \( \lambda \) denotes the failure-rate; \( \lambda_b \) represents the base failure-rate of MOS digital logic; \( G \) refers to Gate count; \( \Pi_F = 1.0 \); \( \Pi_C = 1.0 \); \( \Pi_E = 1.0 \) (hermetic package); and \( \Pi_{ME} \) (ground benign environment).

The failure-rate \( \lambda \) 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 (FTT), which represents the number of failures per \( 10^9 \) device hours of accelerated stress tests [25]. Notably, the total operating time, in \( \lambda_b \) can be expressed as the year of manufacturing. Since the proposed EDDR architecture is synthesized by using TSMC 0.18 m IP6M 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. 9 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.

V. CONCLUSION

This work presents an 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 proposed EDDR architecture is also implemented by using VHDL and synthesized by the Synopsys Design Compiler with TSMC 0.18 m IP6M CMOS technology. Experimental results indicate that the proposed EDDR architecture can effectively detect errors and recover data in PEs of a ME with reason- able 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


