CHAPTER 4

PROPOSED ERROR CORRECTION CODES

As the technology scaling continues, on-chip interconnects are placed closely. As the separation between the on-chip interconnects is reduced, transition of a signal on a single wire causes the signal to transit on neighboring wires due to crosstalk. Crosstalk may result in logical errors in the on-chip interconnect owing to timing violations. In addition to the crosstalk errors, on-chip interconnects are exposed to other noise sources like supply voltage fluctuation, EMI and temperature variation. These errors extremely affect the reliability of on-chip interconnects. The probability of adjacent multi-wire errors (burst error) is higher than the probability of multiple random multi-wire errors. Hence, detection and correction of random as well as burst errors is very important to make the NoC interconnects robust against errors. In this chapter, three different error correction codes are proposed to correct random and burst errors. The proposed error correction codes enhance the reliability of data through on-chip interconnects greatly.

4.1 CROSSTALK AVOIDING ENHANCED DOUBLE ERROR CORRECTION CODE (CAEDEC)

To increase the reliability of the NoC interconnects, error correcting codes are employed in NoC. Error correcting codes with crosstalk avoiding
property avoids crosstalk and also corrects errors. The proposed CAEDEC code includes Triplicate Add Parity (TAP) code to avoid crosstalk and to correct random and burst errors up to three bits. The HARQ scheme is used to correct burst error of three bits. The proposed CAEDEC code is an improved version of Crosstalk Avoiding Random, Burst Error Detection and Correction Code (CARBEDC) proposed by (Maheswari & Seetharaman 2014). The CARBEDC code corrects any single bit error, two bit errors that occur in different groups and burst error of three bit. If two bit errors occur in same group, the CARBEDC code does not correct. The proposed CAEDEC code overcomes this drawback and corrects all the one bit, two bit error patterns as well as all the three bit error patterns except for three error patterns as shown in Table 4.1.

4.1.1 Triplicate Add Parity (TAP) Code

Similar to triplication error correction scheme used in (Huang & Hwang 2012), TAP scheme triplicates original data bits and an extra parity bit calculated from the original data bits is added to enhance error correction capability. As shown in Figure 4.1, the replica of the data bits are placed on each side of the original data bit. As same data bits are placed on both the side of the original data bit, the voltages on these wires are the same. Hence, the coupling capacitance between these wires is reduced reducing crosstalk. The addition of parity bit improves the error correction capability compared to the triplication error correction scheme proposed in (Huang & Hwang 2012). Thus, the TAP scheme avoids crosstalk and increases the reliability of the on-chip interconnects.
4.1.2 Encoder and Decoder Design

(i) Design of the proposed CAEDEC encoder

The proposed CAEDEC encoder uses the triplication error correction scheme (Huang & Hwang 2012) added with overall parity bit. An overall parity bit is calculated from the original data bits before triplication. This overall parity bit is appended to the triplicated bits and then transmitted. The added parity bit increases the error correction capability of the proposed CAEDEC code compared to the triplication error correction scheme proposed in (Huang & Hwang 2012). The novelty of the proposed scheme is the addition of a single parity bit to the triplication bit to increase the reliability of on-chip interconnects wires. In addition, a new low complex decoding
algorithm is proposed for additional error correction and detection compared to the error correction scheme proposed in (Huang & Hwang 2012). Though the CAEDEC error control code increases the link wires, on-chip wires are cheap and plentiful with the increasing metal layers in advanced technologies (Huang & Hwang 2012). Triplcation of the message bit increases the minimum Hamming distance to 3 and the addition of overall parity increases the minimum Hamming distance to 4. Based on information coding theory, minimum Hamming distance of \( k \) can detect \((k-1)\) errors and/or correct \( \lfloor (k - 1)/2 \rfloor \) errors. Thus, the proposed CAEDEC code is capable of detecting burst error of three bits and corrects one bit error, two bit errors and three bit errors except for three error patterns, as shown in Table 4.1. The proposed 32 bit CAEDEC encoder diagram is shown in Figure 4.2.

(ii) Design of the proposed CAEDEC decoder

The proposed CAEDEC decoder diagram is shown in Figure 4.3(a). The data bits are received and separated into three groups: group A, group B, group C and sent parity \((p0)\) using a group separator shown in Figure 4.3(b). The parities for the three groups are computed and denoted \( p1, p2 \) and \( p3 \). The sent parity is denoted as \( p0 \). Table 4.1 shows the possible distribution of one bit, two bit and three bit errors among the three groups. When a burst error of three occurs in the interconnection link, they are separated as single error in three groups as shown in column 16 in Table 4.1. When all the two bit errors occur in any one of the three groups as shown in columns 7, 8 and 9 in Table 4.1, the parities of these groups do not reveal the occurrence of two bit errors (even number of errors cannot be located using parity computation). Hence in this case, the received data (32 bit data) from three groups are compared to select the error free group.
Figure 4.2  32 bit CAEDEC encoder

Table 4.1  Possible distribution for one bit, two bit and three bit errors among three groups

| Column No | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| One bit Error | Two bit Errors | Three bit Errors |
| Error Correction | Error Correction | Error Correction | Uncorrectable |
| Group A | 1 | 0 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 3 | 1 | 2 | 2 | 1 |
| Group B | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 2 | 0 | 1 | 2 | 0 | 3 | 0 | 0 | 1 | 1 | 0 | 2 |
| Group C | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 2 | 2 | 1 | 2 | 0 | 3 | 0 | 1 | 0 | 1 | 0 |
Figure 4.3. (a) 32bit CAEDEC Decoder  (b) Group separator

Decoding Algorithm

The decoding procedure of the proposed CAEDEC code is explained with the help of Table 4.1 and flow diagram in Figure 4.4. The decoding procedure is explained as follows:

1. The received bits are grouped into three groups as group A, group B, group C and the sent parity p0

2. The parities of the three groups are computed separately and indicated as p1, p2, p3.

3. Change in the calculated parity bit reveals occurrence of single error in that group.
4. For the occurrence of two bit errors in any group, the computed parity will not reveal the errors (even number of errors cannot be located using parity computation). Hence, two bit errors are identified by comparing the received data as it will not be equal when errors are present in it.

5. As shown in Table 4.1, for the columns 1, 4, 12 and 15 p1 \neq p2 \& p2= p3. For this condition, to select the error free group, p1 is compared with the sent parity p0 (Is p1=p0?). If p1= p0, then group A is selected as an error free output. Otherwise, group B is selected as an error free output.

6. For the columns 3, 6, 11 and 14 in Table 4.1, p1= p2 \& p2 \neq p3. To select the error free group, p1 is compared with the sent parity p0 (Is p1=p0?). If p1= p0, then group A is selected as error free output. Otherwise group C is selected as an error free output.

7. In Table 4.1 for the column 2, 5, 10 and 13 p1 \neq p2 \& p2 \neq p3. For this condition, to choose the error free group, p1 is compared with the sent parity p0 (Is p1=p0?). If p1= p0, then group A is selected as an error free output. Otherwise group B is selected as an error free output.

8. For the columns 7, 8, 9 and 16, p1= p2 \& p2= p3. For this condition, there are two possibilities: (i) All the three groups may have single bit errors as shown in column 16. (ii) Two bit errors may have occurred in any one of the three groups as shown in column 7, 8 and 9. To detect the burst error of three, p1 is compared with p0. If p1 \neq p0, it reveals the occurrence of burst error of three. Otherwise, two bit errors might have occurred in any one of the three groups. For this condition,
error free output is selected by comparing the received data, as shown in Figure 4.3(a) and Figure 4.4.

![Flow Diagram for CAEDEC Decoder](image)

**Figure 4.4 Flow diagram for CAEDEC decoder**

4.1.3 Hybrid ARQ using CAEDEC Codes
The HARQ scheme is used in the proposed CAEDEC code to correct the burst error of three. Figure 4.5 shows the implementation of the HARQ scheme in the proposed CAEDEC code. The proposed CAEDEC encoder is placed in the output port of the NoC router (transmitter). The input flit is first encoded by the proposed CAEDEC encoder and then transmitted and simultaneously it is stored in the buffer for retransmission, in case of burst error of three is detected. In the input port of the NoC router (receiver), the data is first decoded by the proposed CAEDEC decoder. If the burst error of three is detected, a control signal is sent back to the control logic in the transmitter. The control logic triggers the MUX to retransmit the flit stored in the buffer.

![Diagram of the HARQ Scheme using CAEDEC code](image)

**Figure 4.5** HARQ Scheme using CAEDEC code

### 4.1.4 Performance Analysis

In this section, the performance of the proposed error correction scheme is evaluated in terms of codec area, power, delay, average flit latency, reliability and link energy and power consumption. The evaluation is performed for a 32 bit flit. The proposed CAEDEC code for (97, 32) is compared with random error correcting codes and the burst error correcting code. For random error correcting codes, the proposed CAEDEC code is compared with Hamming SEC code (38, 32) proposed in (Yu et al 2010,
Rossi et al 2007), DAP code (65, 32) proposed in (Pande et al 2006), CADEC (77, 32) code proposed in (Ganguly et al 2008), and SCG code (120, 32) proposed in (Huang & Hwang 2012). For the burst error correction, the proposed CAEDEC code is compared with CADEC (77, 32) code proposed in (Pande et al 2006) and multiple Hamming SEC with interleaving (48, 32) proposed in (Yu & Ampadu 2008).

Codec Area and Power

The proposed CAEDEC code is evaluated by performing simulation on 1 x 2 NoC. The proposed CAEDEC code and other error correction schemes are designed and synthesized in TSMC 0.18μm technology using the Cadence RTL encounter tool. Table 2 summarizes the area and power consumption of the 1 x 2 NoC with no coder and decoder placed (uncoded) and with different error correction codes placed in the router (coded). For coded schemes, different error correction encoders are placed in output port of the router and the decoders are placed in the input port of the NoC router to evaluate the area and power of the NoC router, including the encoder and decoder. The silicon area and the power consumption of the 1 x 2 NoC with the proposed CAEDEC encoder and decoder placed in the router are 25630 μm² and 476.63μW respectively. The NoC router placed with the proposed CAEDEC code has 3.2% & 4.2% less area and 6 % & 3% less power consumption compared to Hamming (38, 32) code and Hamming (48, 32) code respectively. The NoC router placed with the proposed CAEDEC code has 0.1% less area and 1.7% less power consumption compared to CADEC code (Ganguly et al 2008), which is the nearest competitor to the proposed CAEDEC code. The proposed CAEDEC code achieves only 1.4% and 1.2% increase in the area compared to SCG (120, 32) coding proposed in (Huang & Hwang 2012) and DAP (65, 32) code proposed in (Pande et al 2006) respectively. This area increase
is very little and is negligible as the proposed CAEDEC code has high reliability compared to the reliability of the DAP (65, 32) and SCG (120, 32) codes. This is because, the proposed CAEDEC code corrects all the error patterns of one bit error, two bit errors, three bit errors except 3 error patterns as shown in Table 4.1. But DAP (65, 32) and SCG (120, 32) codes correct only a single error.

Table 4.2  Silicon area and power over head for 1 x 2 NoC embedded with CAEDEC code and different error correction codes

<table>
<thead>
<tr>
<th>Codec Scheme</th>
<th>Area in μm²</th>
<th>Power in μW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Un coded (router without error correction code)</td>
<td>24981 (100%)</td>
<td>458.4</td>
</tr>
<tr>
<td>DAP (65, 32) code (single random error)</td>
<td>25322 (101%)</td>
<td>474.62</td>
</tr>
<tr>
<td>Hamming (38, 32) code (single random error)</td>
<td>26496 (106%)</td>
<td>507.7</td>
</tr>
<tr>
<td>Hamming + Interleaving (48, 32) (Single random error &amp; burst Error of two)</td>
<td>26791 (107%)</td>
<td>492.06</td>
</tr>
<tr>
<td>CADEC (77, 32) (All the 1 bit, 2 bit random and burst errors)</td>
<td>25666 (102%)</td>
<td>485.17</td>
</tr>
<tr>
<td>SCG coding (120, 32) (Single random error)</td>
<td>25269 (101%)</td>
<td>471.26</td>
</tr>
<tr>
<td>CARBEDC code (97, 32) (All the 1 bit error, 2 bit errors in different group, 3 bit burst errors)</td>
<td>25333 (101%)</td>
<td>475.36</td>
</tr>
<tr>
<td>Proposed CAEDEC code (97, 32) (All the 1 bit, 2 bit and majority of 3 bit error patterns including burst errors)</td>
<td>25630 (102%)</td>
<td>476.63</td>
</tr>
</tbody>
</table>
Table 4.3 compares the area and power consumption of the coder + decoder alone for the proposed CAEDEC code and different error correction codes for the flit width of 32 bits. In Table 4.3, Phit refers physical transfer unit. Table 4.3 also compares error correction capability for different error correction codes, number of interconnection wires needed and the crosstalk avoidance property. The proposed CAEDEC code has less area and power consumption compared to other error correction codes except DAP and SCG codes.

<table>
<thead>
<tr>
<th>Codec Scheme</th>
<th>Error correction capability</th>
<th>Crosstalk Avoidance</th>
<th>Phit Sizes</th>
<th>Power in µW</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAP (65, 32)</td>
<td>single error</td>
<td>Duplication</td>
<td>Router: 32, Wire: 65</td>
<td>341</td>
</tr>
<tr>
<td>Hamming (38,32)</td>
<td>single error</td>
<td>No</td>
<td>Router: 32, Wire: 38</td>
<td>1515</td>
</tr>
<tr>
<td>Hamming + Interleaving (48, 32)</td>
<td>single error &amp; burst error of two</td>
<td>No</td>
<td>Router: 32, Wire: 48</td>
<td>1810</td>
</tr>
<tr>
<td>CAEDEC (77, 32)</td>
<td>Corrects all the one bit &amp; two bit error patterns</td>
<td>Duplication</td>
<td>Router: 32, Wire: 77</td>
<td>685</td>
</tr>
<tr>
<td>SCG coding (120, 32)</td>
<td>Single error</td>
<td>Triplication</td>
<td>Router: 32, Wire: 120</td>
<td>288</td>
</tr>
<tr>
<td>CARBEDC code (97, 32)</td>
<td>Corrects random and burst error of one bit, two bit error that occur in different group</td>
<td>Triplication</td>
<td>Router: 32, Wire: 97</td>
<td>352</td>
</tr>
<tr>
<td>Proposed CAEDEC code (97, 32)</td>
<td>Corrects all the one bit, two bit errors and three bit errors except for 3 error patterns</td>
<td>Triplication</td>
<td>Router: 32, Wire: 97</td>
<td>649</td>
</tr>
</tbody>
</table>
Codec delay and average packet latency

The delay of the 1 x 2 NoC for the proposed CAEDEC code and different error correction schemes for different packet injection rate is shown in Figure 4.6. The SCG code achieves least delay among the single error correction codes and the proposed CAEDEC code achieves minimum delay compared to the burst error correction codes. Figure 4.7 shows average packet latency against flit error rate for the two noise conditions: (i) 1 bit error and (ii) 3 bit errors. All the error correcting codes used for comparison can detect and correct 1 bit error.

Hence, for 1 bit error, all the error correcting codes achieve the same average packet latency. But for the 3 bit errors, in high noise environment (i.e $10^{-10}$ and $10^{-1}$) except the proposed CAEDEC code, all other error correcting codes achieve the same average packet latency. This is because more errors occur in high noise environment. Only the proposed CAEDEC code corrects three bit errors and retransmission is needed to correct 3 bit burst errors increasing packet latency. But all the other error correcting codes do not detect 3 bit errors and hence, no retransmission is needed. But these codes fail to correct three bit errors. Hence, the received flits are not guaranteed to be error free in the remaining error correcting codes, except in the proposed CAEDEC code.
Figure 4.6  Delay of the NoC Switch against packet injection rate for different error correction schemes

Figure 4.7 (Continued)
Reliability

Residual flit error probability of the proposed scheme: The proposed CAEDEC coding scheme is capable of correcting all one bit errors, two bit random and burst errors. The proposed CAEDEC code also corrects the majority of the three bit random errors and detects three bit burst errors. The HARQ system is used for retransmission to correct three bit burst errors. The residual flit error rate of the HARQ system can be expressed as in Equation (4.1) given below

\[ P_{\text{residual}} = P_e + P_{d-nc} \cdot P_e + P_{d-nc}^2 \cdot P_e + \ldots \]  \hspace{1cm} (4.1)

Where, \( P_e \) is the probability of undetectable error (residual error), \( P_{d-nc} \) is the probability that an error is detectable but not correctable. For the proposed scheme, the residual flit error rate is computed for the two
cases where \( p_n = 0 \) (only random occurs) and \( p_n = 10^{-2} \) (both random and burst error occur).

**a) For the case \( p_n = 0 \):** For \( p_n = 0 \), only random errors occur. To compute the residual flit error rate for this case, probability of correct decoding for random error is computed first. The proposed CAEDEC code encodes a flit of \( n \) information bits as \( (3n+1, n) \). Correct decoding for the proposed CAEDEC scheme is possible when the number of random errors in the flit is two or less (some three bit errors are also corrected. But only two bit errors are considered). Hence \( P_{\text{correct, random}} \) is given in Equation (4.2)

\[
P_{\text{correct, random}} \geq P(3n+1, 0) + P(3n+1, 1) + P(3n+1, 2) \quad (4.2)
\]

where \( P(3n+1, 0), P(3n+1, 1) \) and \( P(3n+1, 2) \) are the probabilities of zero errors, one bit error and two bit errors in a \( (3n+1) \) number of bits. The probability of \( m \) errors in \( l \) number of bits for BER \( \varepsilon \) is given by

\[
P(l, m) = \binom{l}{m} \varepsilon^m (1 - \varepsilon)^{l-m} \quad (4.3)
\]

Therefore, \( P_{\text{correct, random}} \) for the proposed scheme is given by,

\[
P_{\text{correct, random}} = \sum_{m=0}^{2} \binom{3n+1}{m} \varepsilon^m (1 - \varepsilon)^{(3n+1)-m} \quad (4.4)
\]

where, \( P_{\text{correct, random}} \) is the probability of correct decoding in the case of random errors. The probability of undetectable error (residual error) \( P_e \), is given as in Equation (4.5)

\[
P_e(p_n=0) = 1 - P_{\text{correct, random}} \quad (4.5)
\]
The probability of undetectable error (residual error) $P_e$, for the proposed scheme is obtained by substituting Equation 4.4 in Equation 4.5

$$P_e(p_n=0) = 1 - \sum_{m=0}^{2} \binom{3n+1}{m} \varepsilon^m (1 - \varepsilon)^{(3n+1)-m}$$

(4.6)

The proposed scheme can detect the burst errors of three but cannot correct it. Hence, the probability that an error is detectable but not correctable $P_{d-nc}$ for the proposed scheme is given by

$$P_{d-nc}(p_n=0) = \binom{3n+1}{3} \varepsilon^3 (1 - \varepsilon)^{(3n+1)-m}$$

(4.7)

b) **For the case $p_n=10^{-2}$**: For $p_n=10^{-2}$, there is a possibility of occurring random errors and burst error. A combination of single random error and two burst error is considered. The probability of correct decoding for burst error of two is given below (Fu & Ampadu 2009b) in Equation (4.8).

$$P_{\text{correct, burst}} = 2 \cdot p_n \cdot (1 - p_n) \cdot \varepsilon$$

(4.8)

where, $P_{\text{correct, burst}}$ is probability of correct decoding for the case of burst error. The probability of undetectable error $P_e$, is given in Equation (4.9) and (4.9a)

$$P_e( p_n=10^{-2} ) = 1 - ( P_{\text{correct, random}} + P_{\text{correct, burst}} )$$

(4.9)

$$P_e( p_n=10^{-2} ) = 1 - \left( \sum_{m=0}^{1} \binom{3n+1}{m} \varepsilon^m (1 - \varepsilon)^{(3n+1)-m} + 2 \cdot p_n(1 - p_n) \cdot \varepsilon \right)$$

(4.9a)

The proposed code can detect the burst error of three, but cannot correct it. Hence, the probability that an error is detectable but not correctable, $P_{d-nc}$ for the proposed scheme is given in Equation (4.10).
\[ P_{d-n}(p_n=10^{-2}) = p_n^2 \cdot \varepsilon \]  \hspace{1cm} (4.10)

Substituting Equation (4.9a) and Equation (4.10) in Equation (4.1), we get

\[
P_{\text{residual}} = 1 - \left( \sum_{m=0}^{3n+1} \binom{3n+1}{m} \varepsilon^m (1 - \varepsilon)^{(3n+1)-m} + 2 \cdot p_n \cdot (1 - p_n) \cdot \varepsilon \right) + p_n^2 \cdot \varepsilon \cdot 1 - \left( \sum_{m=0}^{3n+1} \binom{3n+1}{m} \varepsilon^m (1 - \varepsilon)^{(3n+1)-m} + 2 \cdot p_n \cdot (1 - p_n) \cdot \varepsilon \right) + \ldots \]  \hspace{1cm} (4.11)

Figure 4.8 shows the probability of residual flit error rate for different error correction schemes. Two noise situations: (i) random errors only (ii) combination of random and burst errors are considered with noise voltage deviation \( \sigma_n \) from 0.06 to 0.16. Figure 4.8 (a) shows the residual flit error rate for random errors only. The residual flit error rate increases as the noise voltage deviation \( \sigma_n \) increases. As, both Hamming (38, 32) code and DAP(65, 32) code are single error correcting codes, only DAP (65, 32) code is considered for comparison of residual flit error rate. The residual flit error rate of the proposed scheme is four magnitude order less compared to CADEC (77, 32) error correction scheme when \( p_n = 0 \). This is because, the proposed code corrects more errors compared to CADEC (77, 32) error correction code. Figure 4.8 (b) shows the residual flit error rate for a combination of random and burst errors. When \( p_n \) value increases, occurrence of burst error increases. Hence, the residual flit error rate of DAP (65, 32) increases. This is because DAP (65, 32) code fails to correct burst error. However, Hamming + interleaving (48, 32) code, CADEC (77, 32) code and the proposed code achieve better residual flit error rate.
Figure 4.8  Comparison of residual flit error rate for CAEDEC code and different error correction codes (a) for random errors only (b) for combination of random and burst errors
Link Swing Voltage

The reliability of NOC interconnects is enhanced by the application of error correction coding as it is robust against errors. This robustness allows reduction in the supply voltage as it can tolerate noise margins. Hence, this result in the savings in power consumption as it depends quadratically on voltage swing (Sridhara & Shanbhag 2005, Fu & Ampadu 2009b). In (Rossi et al 2007, Yu et al 2010), lower swing voltages are used for more powerful error correction coding. For the given reliability requirement, the link swing voltage (Ejlali et al 2007) can be decided by

\[ V_{dd} = 2. \sigma n \cdot Q^{-1}(\hat{\epsilon}) \]  (4.12)

Where, \( Q^{-1}(\hat{\epsilon}) \) is the inverse Q function and \( \hat{\epsilon} \) is the value at which \( P_{\text{residual}}(\hat{\epsilon}) = P_{\text{req}} \). Where \( P_{\text{req}} \) is the required flit error rate.

![Graph](image)

**Figure 4.9** Link swing voltage for CAEDEC code and different error correction codes
Figure 4.9 compares the link swing voltage for different error correction schemes. The proposed method achieves lower swing voltage for the same required flit error rate, compared to other error correction schemes. For the required flit error rate of $10^{-3}$, the proposed method achieves up to 26% reduction in the swing voltage compared to compared to DAP (65, 32) code, 20% compared to SCG coding and 14% compared to CADEC (77, 32) code.

**Link power consumption**

The link power consumption $P_{WL}$ for on-chip interconnects is related to interconnect capacitance $C_L$, the wire switching factor $\alpha$, the link width $W_L$, link swing voltage $V_{\text{swing}}$ and clock frequency $f_{\text{clk}}$. The link power $P_{WL}$ can be expressed (Adler & Friedman 2000) by the Equation (4.13)

$$P_{WL} = C_L \cdot \alpha \cdot W_L \cdot V_{\text{swing}}^2 \cdot f_{\text{clk}}$$  \hspace{1cm} (4.13)

where $\alpha$ is assumed to be 0.5, $W_L$ depends on the error control schemes. The link swing voltage $V_{\text{swing}}$ depends on the reliability requirement of different error control schemes according to Equation (4.12). The link swing voltage $V_{\text{swing}}$ can be reduced for the code with high error correction capability (Rossi et al 2007, Yu et al 2010). For the given reliability requirement, the error control codes with low error correction capability need a higher link swing voltage than the error control schemes with high error correction capability (Rossi et al 2007, Yu et al 2010).
Figure 4.10  Link power consumption for CAEDEC code and different error correction codes

Link power consumption of CAEDEC code and different error correction codes for the residual flit error rate of $10^{-3}$ is shown in Figure 4.10. In NoC environment, the length of on-chip interconnects is the distance between two routers. The link length is decided by the core size and is assumed to be of few millimeters (Fu & Ampadu 2008b). The link Power consumption is evaluated for TSMC 0.18 μm technology. The capacitance value for TSMC 0.18 μm technology is assumed to be 208 fF for 1mm link (Fu & Ampadu 2008b). The switching factor $\alpha$ is assumed to be 0.5. Clock frequency $f$ is 1 GHz. For the residual flit error rate of $10^{-3}$, the proposed CAEDEC code achieves 20% reduction in the link power consumption, compared to DAP (65, 32) code, 53% reduction compared to SCG (120, 32) code and 9% less link power consumption compared to CADEC (77, 32) code.
**Link energy consumption**

Link energy consumption is one of the metrics for analyzing the performance of on-chip interconnects. Link energy consumption $E_{\text{link}}$ is given by (Fu & Ampadu 2009a) Equation (4.14)

$$E_{\text{link}} = \alpha \cdot W_L \cdot C_L \cdot V_{\text{swing}}^2$$  \hspace{1cm} (4.14)

where $\alpha$ is the switching factor, $W_L$ is the interconnect width, $C_L$ is the capacitance of on-chip interconnects and $V_{\text{swing}}$ is the link swing voltage. For on-chip interconnects, link energy consumption must be reduced to lower the total energy consumption. By reducing the link swing voltage $V_{\text{swing}}$ the link energy consumption is reduced, as energy consumption is quadratically related to link swing voltage. For the code with high reliability (more error correction capability) low link swing voltage can be applied, compared to the code with low reliability (less error correction capability) (Rossi et al 2007, Yu et al 2010). Link energy consumption is analyzed for different error control schemes for different link lengths. Figure 4.11 compares the link energy consumption for the reliability requirement of $10^{-3}$ and for different link lengths. For the reliability requirement of $10^{-3}$, the proposed CAEDEC code achieves link energy minimization of 53% compared to SCG (120, 32) code and 20% compared to DAP (65, 32) code.
4.2 MULTI BIT RANDOM AND BURST ERROR CORRECTION CODE WITH CROSSTALK AVOIDANCE (MBRBEC)

The error correction code CAEDEC proposed in section 4.1 corrects one bit error, two bit errors, and a majority of the three bit errors. Due to continued technology scaling, probability of error occurrence also increases. Hence it is necessary to have an error correction code with higher error correcting capability. In this section, an error correction code is proposed to correct errors up to five bits (i.e. 1,2,3,4 and 5 errors). The proposed code is based on single error correction-double error detection (SEC-DED) extended Hamming code and standard triplication error correction scheme. Using single error correction-double error detection (SEC-DED) extended Hamming code and standard triplication error correction scheme a new decoding algorithm is proposed, to correct multiple errors of upto five bits in on-chip interconnection link. Triplication error correction scheme provides crosstalk
avoidance by reducing the coupling capacitance of the interconnection wire. The proposed code is called Multi Bit Random and Burst Error Correction code with crosstalk avoidance (MBRBECC) code.

4.2.1 Encoder and Decoder Design

(i) Design of the proposed MBRBECC Encoder

The proposed MBRBECC encoder uses SEC-DED extended Hamming code (39, 32) to encode initial message bits. Triplication error correction scheme is one of the standard error correction schemes used in communication system to correct errors. Triplication error correction scheme is proposed to correct errors in on-chip interconnection link as in (Huang & Hwang 2012). Using triplication error correction scheme, each of the encoded message bit is triplicated (Huang & Hwang 2012). Thus, if the initial SEC-DED extended Hamming code is \((n, l)\), where \(n\) is the encoded message and \(l\) is the original message, then the final number of bits in the triplication message is \(3n\). The triplication of the message bit is used to correct the errors and simultaneously avoid crosstalk.

The minimum Hamming distance of the SEC-DED extended Hamming code is 4. The triplication of the encoded message increases the minimum Hamming distance to 12. Based on information coding theory, the minimum Hamming distance of \(k\) can correct \(\left\lfloor (k-1)/2 \right\rfloor\) errors. Therefore, MBRBEC code can correct up to five errors. The block diagram of the proposed MBRBEC encoder is shown in Figure 4.12 (a) and the flow diagram is shown in Figure 4.12 (b).
(ii) Design of the proposed MBRBEC Decoder

The proposed MBRBEC decoder exploits the fact that SEC-DED decoder corrects single error and detects double errors. SEC-DED decoder sometimes detects four bit errors if the syndrome value is not zero. If the
occurrence of four bit errors produces the syndrome value as zero, then SEC-DED decoder does not detect four bit errors (Ganguly et al 2009). The complete block diagram of the proposed decoder is shown in Figure 4.13 (a). The received bits are grouped as group A, group B and group C in a group separator. The group separator is a simple wired connection that separates received bits into three groups (Received_A, Received_B and Received_C) as shown in Figure 4.13 (c). The three received groups are given to three SEC-DED decoders that compute the syndrome values Syn_A, Syn_B, Syn_C and occurrence of double errors as Double_error_A, Double_error_B and Double_error_C for the three groups. Based on the syndrome value, each SEC-DED decoder corrects the occurrence of single error and detects the occurrence of the double errors in each group. The decoded values (output from three decoders) Decoded_A, Decoded_B and Decoded_C are given to the multiplexer. The three received groups Received_A, Received_B, Received_C and Decoded_A, Decoded_B are compared in the comparator as shown in Figure 4.13 (b) to generate signals Received_Not_eql and DecodeA_eql_DecodeB. Received_Not_eql signal will be ‘1’ if all the three groups are different. DecodeA_eql_DecodeB signal will be ‘1’ if decoded output from decoder A is equal to decoded output from decoder B (if both decoders have zero error or single error). Tables 4.4 – 4.7 show all possible distributions of 1, 2, 3, 4, and 5 bit errors in the three groups. As shown in Tables 4.4 – 4.7, for the occurrence of errors upto 5 bits, the outputs of decoder A and decoder B will be the same: (i) if both the decoders have error free outputs (either 0 error or single error to be corrected by SEC-DED decoder). (ii) both the decoders have 2 bit errors (for the burst errors of four or five, the place of occurrence of errors in both the decoders is same and so, the decoded value is also same).
Figure 4.13 (a) Block diagram of the proposed MBRBEC decoder

Figure 4.13 (b) Comparator
Table 4.4 Possible distribution of 1 bit and 2 bit errors among three decoders

<table>
<thead>
<tr>
<th>Column No</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoders</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decoder A</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Decoder B</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Decoder C</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.5 Possible distribution of 3 bit errors among three decoders

<table>
<thead>
<tr>
<th>Column No</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoders</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Three bit errors</td>
</tr>
<tr>
<td>Decoder A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Decoder B</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Decoder C</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 4.6  Possible distribution of four bit errors among three decoders

<table>
<thead>
<tr>
<th>Column No</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoders</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decoder A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Decoder B</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Decoder C</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.7  Possible distribution of five bit errors among three decoders.

| Column No | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
|-----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Decoders  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Decoder A | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2  | 2  | 2  | 3  | 3  | 4  | 4  | 4  | 4  | 5  |    |    |    |
| Decoder B | 5 | 0 | 4 | 1 | 2 | 4 | 0 | 3 | 1 | 2  | 3  | 0  | 2  | 1  | 2  | 0  | 1  | 1  | 0  | 0  |    |    |    |
| Decoder C | 0 | 5 | 1 | 4 | 2 | 3 | 0 | 4 | 1 | 3  | 2  | 0  | 3  | 1  | 2  | 0  | 2  | 1  | 0  | 1  | 0  |    |    |

Figure 4.14 shows the detailed diagram of the SEC-DED decoder. The syndrome values C1, C2, C3, C4, C5 and C6 are computed from the received bits in the syndrome computation block. The syndrome values are given to the syndrome decoder unit which detects the error location for single error. The error is corrected in the xor block. The message decoder separates the 32 message bits from the 39 bits which includes 32 message bits and 7 parity bits. Double error is detected using syndrome values and overall parity bit.
Figure 4.14 Detailed diagram of the SEC-DED decoder

As shown in Figure 4.15, based on the double error detection Double error_A, Double error_B and Double error_C, syndrome values Syn_A, Syn_B and Syn_C, Received_Not_eql and DecodeA_eql_DecodeB signals, the multiplexer selects the decoded value which is error free as the single bit error will be corrected by the SEC-DED decoder. The decoding algorithm to detect 1, 2, 3, 4, and 5 bit errors is explained through a flow diagram shown in Figure 4.15. The decoding algorithm consists of the following steps:

Step 1: Receive the bits and group the bits into three groups A, B and C using group separator.
Step 2: Compute the signals Syn_A, Syn_B, Syn_C, Double error_A, Double error_B, Doubleerror_C, Received_Not_eql and DecodeA_eql_DecodeB. Received_Not_eql signal is used to handle certain situations like (4,1,0) (decoder A has 4 errors, decoder B has 1 error and decoder C has 0 error), (4,0,1), (0,4,1), (0,1,4) and (5,0,0). These error patterns are shown in Table 4.7 in columns 19, 20, 3, 4, 21. Similarly DecodeA_eql_DecodeB signal is used to handle situations like (4,0,0), (0,4,0), (0,0,4), (1,1,3), (1,3,1), (2,2,1) and (3,1,1). These error patterns are shown in Table 4.6 in columns 15, 4, 5 and in Table 4.7 in columns 10, 9, 14.

Step 3: The decoding process starts from Double error_A is equal to 1 or not.

- As shown in Table 4.4 for 1 bit and 2 bit errors, for all the columns except the last one, Double error_A is not equal to 1. Hence, it follows the ‘No’ path in the flow diagram and finally selects one of the three copies which is error free using the signals Syn_A, Syn_B, Syn_C, Double error_A, Double error_B, Double error_C, Received_Not_eql and DecodeA_eql_DecodeB. For the last column Double error_A is equal to 1. For this case, Double error_B, Syn_B, Syn_C and Received_Not_eql signals are used to select the decoded B which is error free.

- Similarly, as shown in Tables 4.5 - 4.7 for 3 bit errors, 4 bit errors and 5 bit errors, the flow diagram follows ‘Yes’ path or ‘No’ path based on the occurrence of double errors in group A. Syn_A, Syn_B, Syn_C, Double error_A, Double error_B, Double error_C, Received_Not_eql and DecodeA_eql_DecodeB signals are used to select the copy which is error free.
Figure 4.15 Flow chart of the proposed MBRBEC decoder
4.2.2 Performance Analysis

The performance of the proposed error control scheme is evaluated in terms of codec area, codec power, codec delay, link power, link energy consumption, reliability and link swing voltage. The evaluation is performed for a 32 bit flit. The proposed MBRBEC code is compared with Hamming SEC (38, 32) code, Hamming + Interleaver (48, 32) code, SCG coding (120, 32), DAP code (65, 32), CADEC (77, 32), and JTEC (77, 32).

Codec area, power and delay

The proposed MBRBEC code and the other error control codes are designed and synthesized in TSMC 0.18μm technology using cadence tool. Table 4.8 summarizes the area and power consumption of the 1 x 2 NoC with no coder and decoder placed (uncoded) and with different error control schemes placed in the router (coded). For coded schemes, different error control encoder and decoder are placed in a NoC router to evaluate the area and power consumption of the NoC router including the encoder and decoder. The different encoders are placed in the output port of the router and the decoders are placed in the input port of the router. The silicon area and the power consumption of the 1 x 2 NoC with the proposed encoder and decoder placed in the router are 27270 μm² and 517.28 μW respectively. This is only 3.3% increase in area and 3.8% increase in power consumption compared to the NoC router with JTEC error control scheme proposed in (Ganguly et al 2009), which is the nearest competitor to the proposed code. The small increase in the area and power consumption is a little price to be offered for high reliability of the proposed MBRBEC code.
Table 4.8  Silicon area and power over head for 1 x 2 NoC embedded with MBRBEC code and different error correction codes

<table>
<thead>
<tr>
<th>Codec Scheme</th>
<th>Area in μm²</th>
<th>Power in μW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Un coded</td>
<td>24981</td>
<td>458.4</td>
</tr>
<tr>
<td>(router without error control code)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DAP (65, 32) code (single random error)</td>
<td>25322</td>
<td>474.62</td>
</tr>
<tr>
<td>Hamming (38, 32) code (single random error)</td>
<td>26496</td>
<td>507.7</td>
</tr>
<tr>
<td>Hamming + Interleaving (48, 32) (single random error &amp; burst Error of two)</td>
<td>26791</td>
<td>492.06</td>
</tr>
<tr>
<td>CADEC (77, 32) (random and burst error of two)</td>
<td>25666</td>
<td>485.17</td>
</tr>
<tr>
<td>JTEC (77, 32) (random and burst error of three)</td>
<td>26377</td>
<td>497.89</td>
</tr>
<tr>
<td>SCG coding (120, 32) (single random error)</td>
<td>25269</td>
<td>471.26</td>
</tr>
<tr>
<td>Proposed MBRBEC code (117, 32) (random &amp; burst error of five)</td>
<td>27270</td>
<td>517.28</td>
</tr>
</tbody>
</table>

Table 4.9 summarizes the area, power and delay for the proposed codec (encoder + decoder) and different error control codes for a flit width of 32 bits. Table 4.9 also lists the link swing voltage required for residual flit error rate of $10^{-20}$, error correction capability and the required physical transfer unit size in routers and channels. The proposed MBRBEC code occupies 2289 μm² area, which is a slightly more compared to other error control schemes and consumes 58.88 μW power which is fairly more compared to other error control schemes. This is because, the proposed decoder architecture has three SEC-DED decoders to compute syndrome
values for the three groups and has more decoding steps. But the delay of the proposed code is the same as that of the JTEC scheme (Ganguly et al 2009). As seen in Table 4.9, the proposed MBRBEC code has the highest error correction capability and the lowest link swing voltage compared to other error control schemes which lead to low link power consumption.

<table>
<thead>
<tr>
<th>Codec Scheme</th>
<th>Error correction capability</th>
<th>Crosstalk Avoidance</th>
<th>Phit Sizes</th>
<th>Link Swing Voltage (V)</th>
<th>Codec</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAP (65, 32)</td>
<td>single error</td>
<td>Duplication</td>
<td>32 65</td>
<td>1.02</td>
<td>341</td>
</tr>
<tr>
<td>Hamming (38, 32)</td>
<td>single error</td>
<td>No</td>
<td>32 38</td>
<td>1.02</td>
<td>1515</td>
</tr>
<tr>
<td>Hamming + Interleaving (48, 32)</td>
<td>single error &amp; burst error of two</td>
<td>No</td>
<td>32 48</td>
<td>1.00</td>
<td>1810</td>
</tr>
<tr>
<td>CADEC (77, 32)</td>
<td>Random &amp; burst error of two</td>
<td>Duplication</td>
<td>32 77</td>
<td>0.89</td>
<td>685</td>
</tr>
<tr>
<td>JTEC (77, 32)</td>
<td>Random &amp; burst error of three</td>
<td>Duplication</td>
<td>32 77</td>
<td>0.81</td>
<td>1396</td>
</tr>
<tr>
<td>SCG coding (120, 32)</td>
<td>Single error</td>
<td>Triplcation</td>
<td>32 120</td>
<td>0.96</td>
<td>288</td>
</tr>
<tr>
<td>Proposed MBRBEC code (117, 32)</td>
<td>Random &amp; burst error of five</td>
<td>Triplcation</td>
<td>32 117</td>
<td>0.61</td>
<td>2289</td>
</tr>
</tbody>
</table>

Table 4.9  Summaries of MBRBEC code and different error correction codes for the flit size of 32 bit
Reliability

**Residual flit error probability of the proposed scheme:** The proposed MBRBEC coding scheme is capable of correcting up to five random errors or burst error and the combination of burst error and random errors that counts up to five. For the proposed MBRBEC coding scheme, the residual flit error rate is computed for the two noise scenarios: (i) only random errors occur, where $p_n=0$ (ii) a combination of random and burst errors occur where, $p_n=10^{-2}$.

**a)** **Only random errors occur, $p_n=0$:**

For $p_n=0$, only random errors occur. As the coupling probability $p_n$ is zero, the noise sources affecting the adjacent wire is also zero. Hence only random errors occur. To compute the residual flit error rate for this case, probability of correct decoding for random error is computed first. The proposed MBRBEC coding scheme encodes any flit of ‘n’ information bits as $(3n, n)$. Correct decoding for the proposed MBRBEC coding scheme is possible when the number of errors in the flit is five or less. The proposed scheme might also correct some higher number of errors. The residual flit error probability of the proposed scheme is computed by considering all the cases where correct decoding is possible (Ganguly et al 2009). Hence, the lower bound on the probability of correct decoding for random error, $Pr_{\text{correct, random}}$ is given as follows

$$Pr_{\text{correct, random}} \geq Pr(3n, 0) + Pr(3n, 1) + Pr(3n, 2) + Pr(3n, 3) + Pr(3n, 4) + Pr(3n, 5) \quad (4.15)$$

where $Pr(3n, 0)$, $Pr(3n, 1)$, $Pr(3n, 2)$, $Pr(3n, 3)$, $Pr(3n, 4)$, $Pr(3n, 5)$ are the probabilities of zero errors, one error, two errors, three errors, four errors and
five errors in a \((3n)\) number of bits. The probability of \(\text{‘m’}\) errors in \(\text{‘n’}\) number of bits for BER \(\varepsilon\) is given by

\[
\Pr(n, m) = \binom{n}{m} \varepsilon^m (1 - \varepsilon)^{n-m}
\]  \hspace{1cm} (4.16)

Therefore, \(\Pr_{\text{correct, random}}\) for the proposed scheme is given by,

\[
\Pr_{\text{correct, random}} = \sum_{m=0}^{5} \binom{3n}{m} \varepsilon^m (1 - \varepsilon)^{(3n-m)}
\]  \hspace{1cm} (4.17)

The probability of residual flit error for random errors is given by,

\[
\Pr_{\text{residual}} = 1 - \Pr_{\text{correct, random}}
\]  \hspace{1cm} (4.18)

Substituting Equation (4.17) in Equation (4.18) and for small value of \(\varepsilon\) \(\Pr_{\text{residual}}\) is given by,

\[
\Pr_{\text{residual}} = \binom{3n}{6} \varepsilon^6
\]  \hspace{1cm} (4.18a)

b) **Combination of multiple random errors and burst error:**

For the combination of multiple random errors and burst error, we assume \(p_n=10^{-2}\). As the coupling probability \(p_n\) is not zero, the noise sources affecting the adjacent wire is more. Hence, there is also a possibility of burst error occurring. For the purpose of analysis, a burst error of three and random error of two are considered. Residual flit error probability for this case is given as

\[
\Pr_{\text{residual}} = 1 - (\Pr_{\text{correct, random}} + \Pr_{\text{correct, burst}})
\]  \hspace{1cm} (4.19)

As given in chapter 1, the probability of three burst error \((p(b=3))\), is modeled as (Fu & Ampadu 2009b)

\[
P_{\text{burst}}(b=3) = p_n^2 \cdot \varepsilon.
\]  \hspace{1cm} (4.20)
Pr\textsubscript{residual} is obtained by substituting Equation (4.20) and Equation (4.17) in Equation (4.19) and is given as

$$\text{Pr}_{\text{residual}} = 1 - \left( \sum_{m=0}^{2} \binom{3n}{m} \varepsilon^m (1 - \varepsilon)^{(3n-m)} + p^2 n \cdot \varepsilon \right)$$  \hspace{1cm} (4.21)

For the residual flit error rate computation, two noise situations are considered for different error control schemes: (i) five random errors only (ii) mixture of two random errors and three burst errors. Figure 4.16(a) shows residual flit error rate as a function of noise voltage deviation for multiple random errors only. DAP, SCG, CADEC, JTEC codes are considered for comparison. Reliability of Hamming code (38,32) and DAP code are same as both codes are single error correction codes. Hence, only DAP code is considered for comparison. The simulation results show that the proposed MBRBEC code achieves several order of lower residual flit error rate compared to other error control schemes. This is due to the higher error correction capability (up to 5 bit errors) of the proposed MBRBEC code. SCG codes perform better than DAP code, because SCG coding corrects more than two bit random errors if they occur in two different sets. Figure 4.16 (b) shows the residual flit error rate as a function of noise voltage deviation for the combination of multiple random errors and burst error. The proposed MBRBEC code achieves lowest residual flit error rate compared to other error control codes. For the combination of random errors and burst error, the residual flit error rate of the DAP (65,32) increases as the code fails to correct burst error. Compared to DAP code, SCG coding has lower residual flit error rate as SCG coding corrects burst error of two, if the burst error of two occurs in two different sets.
Figure 4.16 Comparison of residual flit error rate for MBRBEC code and different error correction codes. (a) Multiple random errors only. (b). Combination of multiple random errors and burst error.
Link swing voltage of the proposed MBRBEC code

The link swing voltage of the proposed scheme is computed as mentioned in Equation (4.12). The proposed MBRBEC code uses low swing voltage as it has high error correction capability. Figure 4.17 compares the link swing voltage for different error correction schemes. The proposed MBRBEC code achieves lowest swing voltage compared to other error correction schemes. For the required flit error rate of $10^{-20}$, the proposed MBRBEC code achieves 40% reduction in the swing voltage compared to DAP (65, 32) code, 36% reduction in the swing voltage compared to SCG coding scheme, 31% and 24% reduction in the swing voltage compared to CADEC and JTEC codes respectively.

![Graph showing link swing voltage comparison](image)

**Figure 4.17** Link swing voltage of MBRBEC code and different error correction codes
**Link Power consumption of the proposed MBRBEC code**

The link power consumption of the proposed scheme is computed as given in Equation (4.13). Figure 4.18 shows the link power consumption for the different error correction schemes and for different link lengths for the given reliability requirement of $10^{-20}$ and $10^{-5}$. The power consumption is estimated for 0.18µm technology. The wire capacitance $C_L$ for 0.18µm is obtained as 208fF/mm from [59] and the clock frequency is 1GHz. Because of the increased reliability of the proposed MBRBEC code, it uses low swing voltage that results in low link power consumption compared to other error correction codes.

For the given reliability requirement of $10^{-20}$, the proposed MBRBEC code achieves 12% reduction in the link power compared to JTEC scheme, 27% reduction compared to CADEC scheme, 33% reduction compared to DAP coding, 58% reduction compared to SCG coding and 7% reduction for Hamming + Interleaving (48, 32) code. The proposed code achieves only 9% increase in the link power compared to Hamming (38, 32) code. This small increase is the price offered to achieve high reliability for the proposed code. For the reliability requirement of $10^{-5}$, the proposed MBRBEC code achieves 19% less link power compared to JTEC, 42% less link power compared to CADEC, 41% compared to DAP, 22% reduction compared to Hamming + Interleaving (48, 32) code, 8% reduction compared to Hamming (38, 32) code and 68% compared to SCG coding respectively. This power reduction is because of low swing voltage which comes from higher error correction capability of the proposed MBRBEC code.
Figure 4.18 Link power consumption of MBRBEC code and different error correction codes

Link energy consumption for the proposed MBRBEC code

Link energy consumption is evaluated for different link lengths and for different error control schemes using the Equation (4.14). Figure 4.19 shows the link energy consumption for the different link lengths and for the reliability requirement of $10^{-5}$. The proposed MBRBEC code achieves 19% reduction in the link energy consumption compared to JTEC code (77, 32), 42% reduction compared to CADEC code (77, 32), 22% reduction compared to Hamming + Interleaving (48, 32) code, 8% reduction compared to Hamming (38, 32) code and 68% reduction compared to SCG code (120, 32). This energy minimization is due to low link swing voltage which is because of the high reliability of the proposed MBRBEC code.
Figure 4.19  Link energy consumption of MBRBEC code and different error correction codes

4.3  MULTIPLE BIT ERROR CORRECTION CODE WITH EXTENDED HAMMING PRODUCT CODES (MBECEHPC)

The error correction codes proposed in the previous section use triplication error correction scheme to correct multiple random and burst errors. In this section, an error correction code using extended Hamming product code is proposed to correct multiple random and burst errors. Product codes provide protection against random as well as burst errors with low complex hardware. However, the direct implementation of the product codes requires more number of interconnection links. Hence, Type II HARQ is combined with the product code (in Type II HARQ parity bits alone are incrementally transmitted instead of transmitting the whole data word as in Type I HARQ). This reduces the number of interconnection links. The proposed error correction code uses extended Hamming product code...
combined with Type II HARQ as in (Fu & Ampadu 2008a, Fu & Ampadu 2008b, Fu & Ampadu 2009a). The error correction code proposed in (Fu & Ampadu 2008a, Fu & Ampadu 2008b, Fu & Ampadu 2009a) uses a three stage decoding process (row decoding - column decoding - row decoding). But the code proposed in this section uses only a two stage decoding process (row decoding - column decoding). In the proposed code, keyboard scan based error flipping circuit is used to replace the third stage row decoding. The keyboard scan based decoding greatly reduces the hardware complexity. The proposed error correction code corrects burst errors up to six bits. The proposed code also corrects up to eight bit random errors if the occurrence of error is not more than two bit in each row and in each column. The proposed code is named Multiple Bit Error Correction code with Extended Hamming Product Code (MBECEHPC).

4.3.1 Encoder and Decoder Design

(i) Design of the proposed MBECEHPC encoder

The encoder for the proposed MBECEHPC code is shown in Figure 4.20. The M bit input message is organized as m_2 rows and m_1 columns. Here 32 bit input message is considered. The 32 bit message is arranged as 4 row and 8 columns. The four rows are encoded using four extended Hamming row encoder (13, 8). The row encoded data is given to the row-column interleaver and is transmitted over the interconnection links. The row-column interleaved data is encoded using extended Hamming column encoder (8, 4). From the column encoder, only column parity check bits are retrieved. The retrieved column parity check bits are again encoded, row wise for checks on checks. The column parity check bits and checks on checks are stored in a buffer till NACK signal is received from the decoder. When the NACK signal is received from the decoder, the encoder sends the column parity check bits along with checks on checks.
(ii) Design of the proposed MBECEHPC decoder

The block diagram of the proposed decoder is shown in Figure 4.21. The encoded data is applied to the de-interleaver. The de-interleaved data is applied to the extended Hamming row decoder for decoding. The extended Hamming row decoder corrects any single error that occurs in each row and detects the occurrence of double errors, as shown in Figure 4.14. If double errors are detected, it cannot be corrected. Hence, NACK signal is sent back to the encoder after storing the row decoded data and row parity bits in a buffer. The row condition vector is formed as shown below:

The $k^{th}$ place in the row condition vector is set to ‘1’ if errors in the $k^{th}$ row are detectable but not correctable. Or else it is set to ‘0’.
When the column parity check bits and checks on checks are received they are first passed through the row decoder to correct any error. The previous row decoded data and parity bits stored in the buffer is row-column interleaved. The column parity check bits and row column interleaved data are passed to the second stage column decoding. The column decoder corrects any single error that occurs in each column and detects double errors in the column. The column condition vector is formed as shown below:

The L\textsuperscript{th} place in the column condition vector is set to ‘1’ if the errors are detectable but not correctable. Or else it is set to ‘0’.

![Block diagram of the proposed MBECEHPC decoder](image)

**Figure 4.21 Block diagram of the proposed MBECEHPC decoder**

The row condition vector and column condition vector generation for a rectangular error pattern is shown in Figure 4.22. Row condition vector is set to ‘1’ if double errors are detected in a row and column condition vector is set to ‘1’ when double errors are detected in a column.
Keyboard Scan Based Error Flipping

In the proposed MBECEHPC code, keyboard scan based error flipping replaces the third stage row decoders in the existing work (Fu & Ampadu 2008a, Fu & Ampadu 2008b, Fu & Ampadu 2009a). The simple design of the keyboard scan based error flipping circuit reduces hardware complexity and delay of the decoder compared to the number of row decoders employed in the third stage. The column condition vector, row condition vector, and column decoded data from column decoder are passed to the keyboard scan based error flipping circuit. The keyboard scan based error flipping method locates and corrects the rectangular error pattern as follows:

Step 1: The row condition vector is considered as reference for error flipping. If any bit of the row condition vector is set to ‘1’, then the column condition vector is checked.
Step 2: If any bit of the column condition vector is set to ‘1’ and if the row condition vector is already checked for ‘1’ in step 1, then the matching point leads to a single position as shown in Figure 4.23.

Step 3: After a single position is found out, the particular bit is flipped to correct the error.

Step 4: The same step is repeated for the entire row condition vector bits.

![Figure 4.23 Keyboard scan based error flipping](image)

The overall decoding process of the proposed pipelined two stage multiple error correction code is summarized as follows:

Step 1: The row condition vector is formed based on the error pattern in the row decoders. If any bit is set in the row condition vector, NACK signal is sent to the encoder. The encoder sends column parity check bits.

Step 2: After receiving the column parity check bits, column decoding of the row decoded data is performed. The column condition vector is formed based on the error pattern in the column decoding.
Step 3: Row condition vector, column condition vector, row parity check bits and column decoded data are passed to the keyboard scan based error flipping circuit. The error flipping circuit corrects the error using keyboard scan based decoding as shown in Figure 4.23.

4.3.2 Performance Analysis

The performance of the proposed MBCEHPC code is evaluated for codec area, power consumption, delay, reliability, link swing voltage, link power consumption and link energy consumption. The performance evaluation is done for a 32 bit flit. The 32 bit message is arranged into 4 X 8 matrices. Each row message is encoded using extended Hamming (13, 8) code. Each column is encoded using extended Hamming (8, 4) code. The link width is set equal to 52 which is the number of bits of the row encoders output. The proposed MBCEHPC code is compared with similar error correction codes which have high error correcting capability.

Codec area, power and delay

The area, power consumption and delay of the proposed MBCEHPC error correction scheme and other error correction schemes are shown in Table 4.10. The proposed MBCEHPC code and other error control codes are designed in verilog HDL and synthesized in TSMC 0.18μm technology using the cadence RTL encounter tool. The proposed MBCEHPC error correction scheme has a small increase in area, power and delay compared to Hamming + Interleaving (48, 32) code, CADEC (77, 32) code and JTEC (77, 32) error correction schemes. This is due to the proposed code has little bit more complex hardware compared to other error correction codes. This increase in area, power consumption and delay is a small price offered for the high error correction capability of the proposed code. However, the proposed MBCEHPC error correction scheme has 86% less
codec area and 23% less codec delay compared to the error correction code proposed in (Fu & Ampadu 2009a) which uses Type II HARQ and three stage decoding (row-column-row decoding). This is because, the proposed MBECEHPC error correction scheme has lower decoder complexity as it uses only two stage decoding (row-column decoding) and keyboard scan based error flipping instead of the third stage row decoders used in (Fu & Ampadu 2009a). Table 4.10 also summaries the error correction capability and the link swing voltage required for the residual flit error rate of $10^{-20}$ for different error correction codes. As seen in the Table 4.10, the proposed error correction code has the highest error correction capability and lowest link swing voltage compared to other error correction codes.

**Table 4.10  Summaries of the proposed MBECEHPC codes and other error correction codes**

<table>
<thead>
<tr>
<th>Codec Scheme</th>
<th>Error correction capability</th>
<th>Crosstalk Avoidance</th>
<th>Phit Sizes</th>
<th>Codec</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Router</td>
<td>Wire</td>
</tr>
<tr>
<td>Hamming + Interleaving (48, 32)</td>
<td>single error &amp; burst error of two</td>
<td>No</td>
<td>32</td>
<td>48</td>
</tr>
<tr>
<td>CADEC (77, 32)</td>
<td>Random &amp; burst error of two</td>
<td>Duplication</td>
<td>32</td>
<td>77</td>
</tr>
<tr>
<td>JTEC (77, 32)</td>
<td>Random &amp; burst error of three</td>
<td>Duplication</td>
<td>32</td>
<td>77</td>
</tr>
<tr>
<td>Proposed MBECEHPC (52, 32) code</td>
<td>Multi bit error</td>
<td>No</td>
<td>32</td>
<td>52</td>
</tr>
<tr>
<td>Proposed MBECEHPC (88, 64) code</td>
<td>Multi bit error</td>
<td>No</td>
<td>64</td>
<td>88</td>
</tr>
<tr>
<td>Hamming product code with type II hybrid ARQ (88, 64) [4]</td>
<td>Multi bit error random error and burst error of six</td>
<td>No</td>
<td>64</td>
<td>88</td>
</tr>
</tbody>
</table>
Reliability

The reliability of the proposed MBCEHPC code is dependent on two conditions: (i) error correction and detection capability in the first transmission (ii) the error correction capability when the flit is retransmitted [49]. The residual flit error rate is estimated as given in (Fu 2010)

\[ P_{\text{residual}} = P_{ud} + P(e_{\text{decoding, detect}}) \]  \hspace{1cm} (4.22)

where \( P_{ud} \) is undetectable error probability in the first transmission and \( P(e_{\text{decoding, detect}}) \) is the probability of error after the first retransmission and decoding process is over.

\( P_{ud} \) can be given by

\[ P_{ud} = 1 - (P_{ne} + P_{\text{detect, correct}} + P_{\text{detect, uncertain}}) \]  \hspace{1cm} (4.23)

where \( P_{ne} \) is the probability of no error.

\( P_{\text{detect, correct}} \) is the probability of detectable and correctable error in first transmission.

\( P_{\text{detect, uncertain}} \) is the probability of detectable but uncorrectable error in the first transmission.

\( P(e_{\text{decoding, detect}}) \) can be expressed by

\[ P(e_{\text{decoding, detect}}) = P_{\text{detect, uncertain}} \cdot (1 - P_{ne} - P_{\text{detect, correct}}) \]  \hspace{1cm} (4.24)

By substituting the Equations (4.23) and (4.24) in Equation (4.22), the residual flit error rate for the proposed code is obtained as

\[ P_{\text{residual}} = 1 - (1 + P_{\text{detect, uncertain}}) \cdot (P_{ne} + P_{\text{detect, correct}}) \]  \hspace{1cm} (4.25)
The 32 bit flit is arranged as $m_2$ rows and $m_1$ columns (given in chapter 2, section 2.4.5). Each row is encoded using extended Hamming encoder $EH(n_1, m_1)$. The probability of no error in the first transmission can be given as in (Fu B 2010)

$$P_{ne} = (1 - \varepsilon)^{k_2.n_1} \quad (4.26)$$

In the first transmission, the proposed code corrects all the error patterns with single error that occur in each row. Hence, $P_{detect\_correct}$ can be expressed as

$$P_{detect\_correct} = \sum_{t=1}^{k_1} (p(b=1))^t (1-\varepsilon)^{k_1.n_1-t} + \sum_{t=2}^{k_1} P(b=t)(1-\varepsilon)^{k_1.n_1-1} \quad (4.27)$$

where $P(b=t)$ is probability of $t$-bit burst errors and $P(b=1)$ is probability of single random error. The first term represents the probability that a single error occurs in each row. The second terms represents the probability of occurrence of burst error of length $k_2$.

$P_{detect\_uncorrect}$ can be expressed as

$$P_{detect\_uncorrect} = P_{detect\_uncorrect\_random} + P_{detect\_uncorrect\_burst} \quad (4.28)$$

where $P_{detect\_uncorrect\_random}$ is the probability of detectable but uncorrectable random error in the first transmission and $P_{detect\_uncorrect\_burst}$ is the probability of detectable but uncorrectable burst errors in first transmission.

The extended Hamming code detects two bit errors. Therefore, the proposed error correction code corrects up to 8 bit random errors if the occurrence of errors in the first transmission is not more than two bit in each row and in each column. Hence, $P_{detect\_uncorrect\_random}$ can be expressed as
\[ P_{\text{detect\_uncorrect\_random}} = \sum_{t=1}^{8} (p(b=1))^t (1-\varepsilon)^{k,n-t} \] (4.29)

Equation (4.29) represents the probability of error detection for random errors up to 8 bit when the occurrence of errors in the first transmission is not more than two bit in each row and in each column.

The proposed error correction code corrects any combinations of burst errors up to 6 bit with one retransmission. The combinations are: (i) one two bit burst error with 1 one bit random error (ii) two 2 bit burst errors (iii) one three bit burst error and one random error (iv) one three bit burst error and one two bit burst error (v) two three bit burst errors (vi) one 6 bit burst error.

\[ P_{\text{detect\_uncorrect\_burst}} \text{ can be expressed as given in (Fu 2010)} \]

\[ P_{\text{detect\_uncorrect\_burst}} = P(b=2)P(b=1)(1-\varepsilon)^{k,n-2} + \\
(P(b=2))^{2} (1-\varepsilon)^{k,n-2} + \\
P(b=3)P(b=1)(1-\varepsilon)^{k,n-2} + \\
P(b=3)P(b=2)(1-\varepsilon)^{k,n-2} + \\
(P(b=3))^{3} (1-\varepsilon)^{k,n-2} + \\
P(b=6)(1-\varepsilon)^{k,n-1} \] (4.30)

Each term in Equation (4.30) represents the probability of burst error detection in the first transmission for the above mentioned combinations provided the occurrence of errors is not more than two bit in each row and in column. At the receiver, burst errors are distributed to each row in the row-
column interleaver. Hence, burst error of six will also be distributed such that each row and column does not have errors more than two bit.

The residual flit error rate for the proposed code as a function of noise voltage deviation for different error correction codes is shown in Figure 4.24. The residual flit error rate is examined for the combination of random and burst errors.

![Graph showing comparison of residual flit error rate for MBECEHPC code and different error correction codes for the combination of random and burst errors.](image)

**Figure 4.24** Comparison of residual flit error rate for MBECEHPC code and different error correction codes for the combination of random and burst errors

Compared to Hamming +Interleaving (48, 32), CADEC and JTEC codes the proposed MBECEHPC error correction code achieves the lowest residual flit rate. This is because, the proposed MBECEHPC error correction code corrects maximum of up to 6 bit burst errors. While the other error correction codes like Hamming +Interleaving (48, 32) and CADEC codes correct only two bit burst error. The JTEC code corrects burst error of three bit. Compared to these codes the proposed MBECEHPC code corrects maximum number of burst errors. Hence, the residual flit error rate (uncorrected flits) is reduced.
Link swing voltage of the proposed MBECEHPC code

The error correction codes with higher error correction capability use low link swing voltage which minimizes the link energy. The link swing voltage of the proposed MBECEHPC error correction code is computed using Equation (4.12). Figure 4.25 shows link swing voltage for MBECEHPC code and different error correction codes. The proposed MBECEHPC code uses low link swing voltage compared to other error correction codes as it has higher error correction capability. For the residual flit error rate of $10^{-15}$, the proposed MBECEHPC code achieves 19%, 30% and 39% of less link swing voltage compared to JTEC, CADEC and Hamming + Interleaving (48, 32) codes respectively.

![Link Swing Voltage for MBECEHPC code and different error correction codes](image)

**Figure 4.25** Link Swing Voltage for MBECEHPC code and different error correction codes

Link power consumption for the proposed MBECEHPC code

The link power consumption of the proposed MBECEHPC scheme is estimated as given in Equation (4.13). The link power consumption for
different error control schemes and for different link lengths for the given reliability requirement of $10^{-20}$ and $10^{-5}$ is shown in Figure 4.26. Power consumption is estimated for 0.18\textmu m technology. The wire capacitance $C_l$ for 0.18\textmu m is obtained as 208fF/mm from (Kim et al 2006) and the clock frequency is 1GHz. As the Proposed MBECEHPC code has higher reliability compared to other error correction codes, it uses low link swing voltage. The low swing voltage results in low link power consumption.

![Graph showing link power consumption for different codes](image)

**Figure 4.26 Link power consumption for MBECEHPC code and different error correction codes**

For the reliability requirement of $10^{-20}$, the proposed MBECEHPC code achieves 60%, 52% and 50% reduction in link power consumption compared to CAEDEC, JTEC and Hamming + Interleaving codes respectively. For the reliability requirement of $10^{-5}$, the proposed MBECEHPC code achieves 69%, 56% and 58% less link power consumption.
compared to CAEDEC, JTEC and Hamming + Interleaving codes respectively.

**Link energy consumption for the proposed MBECEHPC code**

Link energy consumption is estimated for different link lengths and for different error control schemes using the Equation (4.14). Figure 4.27 shows the link energy consumption for the different link lengths and for the reliability requirement of $10^{-5}$. The proposed MBECEHPC code achieves 56% reduction in the link energy consumption compared to JTEC code, 69% reduction compared to CADEC code, 58% reduction compared to Hamming + Interleaving (48, 32) code, This energy minimization is due to a lesser amount of link swing voltage due to the higher reliability of the proposed MBECEHPC code and reduced number of interconnection links by the use of Type II HARQ.

![Figure 4.27 Comparison of link energy consumption of MBECEHPC code and different error correction codes](image-url)
4.4 COMPARISON

The proposed three error correction codes are compared with the existing error correction codes. Table 4.11 shows comparison of the proposed three error correction codes and the existing error correction codes regarding error correction capability, codec area, power consumption, delay, residual flit error rate and link power consumption. The residual flit error rate is given for the noise voltage deviation of $\sigma_n=0.06V$ and link power consumption is given for the link length of 1mm and residual flit error rate of $10^{-5}$. Compared to other error correction codes, the proposed MBBEC and MBECEHPC error correction codes have higher error correction capability and less link power consumption with a slight increase in area, power consumption and delay. This minor increase in area and power consumption is negligible due to advanced process technology. Compared to the existing Hamming product based Type II HARQ proposed by Fu (2009), the proposed MBECEHPC error correction code occupies 86% lesser codec area and has 23% delay.

The proposed CAEDEC error correction code has high error correction capability than other existing error correction codes except the JTEC error correction code. This is because, the JTEC code corrects all the three bit error patterns, but the proposed CAEDEC code corrects all the three bit error patterns except for 3 error patterns. The proposed CAEDEC code also has very less codec area, power consumption and link power consumption compared to other error correction codes.
### Table 4.11 Comparison of the existing and proposed error correction codes

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Existing error correction codes</th>
<th>Proposed error correction codes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DAP (65,32)</td>
<td>Proposed CAEDEC</td>
</tr>
<tr>
<td>Error correction capability</td>
<td>Hamming (38,32)</td>
<td>Proposed MBRBEC</td>
</tr>
<tr>
<td></td>
<td>Hamming + Interleaving (48,32)</td>
<td>Proposed MBECEHPC</td>
</tr>
<tr>
<td></td>
<td>CADEC (77,32)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>JTEC (77, 32)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hamming product code with type II hybrid ARQ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCG coding (120,32)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Single random error</td>
<td>Random &amp; burst error of two</td>
</tr>
<tr>
<td></td>
<td>Single random error</td>
<td>Random &amp; burst error of three</td>
</tr>
<tr>
<td></td>
<td>Single random error &amp; burst error of two</td>
<td>Multiple random error &amp; burst error of six</td>
</tr>
<tr>
<td></td>
<td>Random &amp; burst error of three</td>
<td>Single random error</td>
</tr>
<tr>
<td></td>
<td>Multiple random error &amp; burst error of six</td>
<td>Random &amp; burst error of two and some three bits</td>
</tr>
<tr>
<td></td>
<td>Single random error</td>
<td>Random &amp; burst error of five</td>
</tr>
<tr>
<td>Codec Area in μm²</td>
<td>341</td>
<td>4256</td>
</tr>
<tr>
<td>Codec Power in μW</td>
<td>16.22</td>
<td>62.27</td>
</tr>
<tr>
<td>Codec Delay in ns</td>
<td>0.82</td>
<td>1.7</td>
</tr>
<tr>
<td>Residual Flit error rate</td>
<td>10^{-34}</td>
<td>10^{-45}</td>
</tr>
<tr>
<td>Link Power in mW</td>
<td>4.1</td>
<td>1.2</td>
</tr>
</tbody>
</table>


4.5 SUMMARY

Due to various noise sources, the reliability of the on-chip interconnection link is greatly affected. To enhance the reliability of the on-chip interconnect links, three different error correction codes are proposed in this chapter.

The proposed CAEDEC error correction code utilizes TAP scheme, which triplicates the data bits to avoid crosstalk and a parity bit is added to enhance the reliability of the code. The CAEDEC error correction code corrects all the error patterns up to two bits and also corrects three bit error patterns except for three error patterns. The CAEDEC error correction code consumes little area, power with a higher error correction capability and small link power consumption than the existing error correction codes except JTEC error correction code.

The proposed MBRBEC error correction code uses extended Hamming code and standard triplication error correction code. Extended Hamming code corrects single error and detects double errors. Triplication error correction scheme avoids crosstalk. A decoding algorithm is proposed to correct all the error patterns up to five bits. The MBRBEC error correction code has higher error correction capability and small link power consumption with little increase in codec area and power consumption compared to the existing error correction codes.

The proposed MBECEHPC error correction code makes use of extended Hamming product code with Type II HARQ. The proposed MBECEHPC code uses only two stage (Row-Column) decoding with keyboard scan based error flipping. The keyboard scan based error flipping circuit reduces the codec area and delay significantly compared to the existing error correction code proposed by Fu (2009) which uses Type II HARQ and three stage decoding (Row-Column-Row).