Comparative Reliability Analysis between Horizontal-Vertical-Diagonal Code and Code with Crosstalk Avoidance and Error Correction for NoC Interconnects

Hamdan Kadhim Abed 1,*, Wameedh Nazar Flayyih2
Department of Computer Engineering, College of Engineering, University of Baghdad, Baghdad, Iraq
hamdan.abed2005m@coeng.uobaghdad.edu.iq, wam.nazar@coeng.uobaghdad.edu.iq2

ABSTRACT

Ensuring reliable data transmission in Network on Chip (NoC) is one of the most challenging tasks, especially in noisy environments. As crosstalk, interference, and radiation were increased with manufacturers’ increasing tendency to reduce the area, increase the frequencies, and reduce the voltages. So many Error Control Codes (ECC) were proposed with different error detection and correction capacities and various degrees of complexity. Code with Crosstalk Avoidance and Error Correction (CCAEC) for network-on-chip interconnects uses simple parity check bits as the main technique to get high error correction capacity. Per this work, this coding scheme corrects up to 12 random errors, representing a high correction capacity compared with many other code schemes. This candidate has high correction capability but with a high codeword size. In this work, the CCAEC code is compared to another well-known code scheme called Horizontal-Vertical-Diagonal (HVD) error detecting and correcting code through reliability analysis by deriving a new accurate mathematical model for the probability of residual error \( P_{\text{res}} \) for both code schemes and confirming it by simulation results for both schemes. The results showed that the HVD code could correct all single, double, and triple errors and failed to correct only 3.3% of states of quadric errors. In comparison, the CCAEC code can correct a single error and fails in 1.5%, 7.2%, and 16.4% cases of double, triple, and quadric errors, respectively. As a result, the HVD has better reliability than CCAEC and has lower overhead; making it a promising coding scheme to handle the reliability issues for NoC.

Keywords: Error Detection and Correction Codes, Network on Chip Noc, Reliability Analysis, Residual Error Probability.
تحليل موثوقية مقارن بين ترميز افقي-عمودي-محوري وترميز تجانب الحث المتتالي

وتتصحيح الخطأ لروابط شبكة الرقاقة

حمدان كاظم عبد، ومبكر زائر فليل2

قسم هندسة الحاسبات، كلية الهندسة، جامعة بغداد، بغداد، العراق

الخلاصة

أحد أكثر المهام تحداً، خاصة عندما زادت تأثيرات Network on Chip (NoC) البيئات الصاخبة مثل الحب المتبادل والتداخل والكشف مع زيادة انبعاث الشركات المصممة لتقليل المساحة وزيادة الترددات وتقليل الفولتاية. نتيجة لذلك، تم اقتراح العديد من رموز التحكم في الأخطاء (ECC) بفترات مختلفة للكشف عن الأخطاء وتصحيحها ودرجات مختلفة من التعقيد. على سبيل المثال، يستخدم الرمز مع تنبؤ الحث المتبادل وتتصحيح الخطأ (CAEC) للشبكة على التوصيات البيئية للرقاقة بنات التحقق من التكافؤ البسيط كأسلوب رئيسي للحصول على قدرة تصحيح خطأ عالية حيث يمكن لمخطط الترميز هذا وفقًا للمؤلفين تصحيح ما يصل إلى 12 خطأ عشوائي، وتعتبر ذو كفاءة تصحيح كبرى مقارنة بالعديد من مخططات الترميز الأخرى. هذا الترميز لديه قدرة تصحيح عالية ولكن مع حجم كلمة مفتوحة عالية. في هذا البحث، تم التكفل عن HVD مقارنة ترميز آخر معروف يسمى CCAEC بتقنياته HVD مع مخطط ترميز آخر موضح يسمى اللوغ دف يامت الخطا المتبقي الأخطاء وتصحيحها من خلال تحليل الموثوقية والذي يتم من خلال اشتقاق نموذج رئيسي جديد لاحتال الخطا المتبقي بكمية المحاكاة لكل من مخططي الترميز وتأكيدها من خلال نتائج المحاكاة لكل المخططي. أظهرت النتائج أن ترميز HVD يمكنه تحسين جميع الأخطاء المحاكاة والمزدوجة والثلاثية وفشل في تصحيح 3.3٪ فقط من حالات الأخطاء الرباعية، بينما يمكن للترميز الآخر تحسين خطأ واحد وفشل في 1.5٪ و 7.2٪ و 16.4٪ من حالات الأخطاء المزدوجة والثلاثية والرباعية على التوالي. نتيجة لذلك، تمت تعميم HVD بمواصفات أفضل من CCAEC ولديها حمل أقل، مما يجعله مخططاً متميزًا واعدًا للتعامل مع مسائل الرقاقة NoC الموثوقية لـ.

الكلمات الرئيسية: رموز اكتشاف الأخطاء وتصحيحها، الشبكة على الرقاقة، تحليل الموثوقية، احتمالية الخطأ المتتالي.

1. INTRODUCTION

Traditional links, such as the standard bus, lack IP scalability and reusability. Therefore, NoC has been adopted to improve modularity, reliability, and scalability in on-chip communications for multi-core architectures. The requirements for reliable on-chip communication in the current network-on-chip (NoC) have increased concerning increasing the number of node blocks (Giovanni et al., 2002). The NoC comprises the Network Interface (NI), routers, and interconnect links (Mohammed and Flayyih, 2019). These interconnect links are the most affected by the noisy environment, such as crosstalk, radiation, and interference effects (Rahimipour et al., 2012; Rahimipour et al., 2020). In addition to the constraints, Such as shrunk area, increased frequencies, and decreased supply voltage (Murali et al., 2005; Flayyih et al., 2015). Many types of Error Detection And Correction Code (EDAC) techniques have been proposed to face these challenges, representing the most effective approach for supporting reliable on-chip communication. Single Error Detection (SED) is the simplest code using a simple parity check bit technique (Fu and Ampadu, 2012). Single Error Correction and Single Error Detection (SEC-SED) were achieved with hamming code (Singh, 2016). Single Error Correction and Double Error Detection (SEC-DED) were proposed with Extended Hamming code (Murali et al., 2005). Hamming product codes (HPC) (Fu and Ampadu, 2009) can correct up to 5 errors by using
two dimensions of extended hamming in a row and column with type II Hybrid Automatic Repeat Request (HARQ) to reduce transmitted codeword size. Horizontal-Vertical-Diagonal error detecting and correcting code (HVD) uses four direction simple parity bits to detect up to 7 errors or detect up to 4 error bits and correct up to three errors (Kishani et al., 2011). In Shamshiri et al., 2011, an end-to-end error location-aware correction code is produced for 64-bit data which can correct up to 16 bursts and 2 random errors. For 32-bit data, the 14-bit burst error can be corrected by using Hamming code, forbidden pattern code (FPC), and overlapping, which was proposed in multiple continuous errors correct coding (MCECC) (Wang, 2011).

In Multibit Error Correcting Coding with Reduced Link Bandwidth (MECCRLB) (Vinodhini and Murty, 2018), 25 parity bits are added to the 32-bit input data. MECCRLB can correct an 11-bit random error, a 4-bit burst error, or the 4-bit random error and the built-in burst error. But, it is proved by Asaad et al., 2020 that it can correct only one error.

After proposing some techniques for crosstalk avoidance, such as shielding (Kose et al., 2010), adding buffers or repeaters (Zangeneh and Masoumi, 2010), duplication (Ganguly et al., 2009), and Crosstalk Avoidance Code (CAC) (Shirmohammadi and Miremadi, 2017). Many EDAC techniques were combined with crosstalk avoidance techniques to enhance the reliability of NoC, where in Joint crosstalk avoidance and Triple Error Correction (JTEC) (Ganguly et al., 2009), a coding technique (77, 32), Hamming code and duplication are combined to correct 3 bits of error. This code is further enhanced for Triple Error Correction and Quadruple Error Detection (JTEC-SQED). The Hsiao code is used with triplication, giving a 117-bit codeword for 32-bit input data, 5-bit errors can be corrected or 6-bit errors detected upon decoding (Vinodhini et al., 2015). In Duplicated Two-Dimensional Parities (DTDP) (Flayyih et al., 2014), two-dimensional parity bits with codeword duplicated are used for 7-bit error detection. When the decoder detects the error, the receiver sends a request signal to the sender for retransmission. Later, this code was enhanced to correct a single error and detect 6 errors (Flayyih et al., 2020). In Joint Crosstalk Aware Multiple Error Correction (JMEC) (Gul, 2017), 32-bit data duplication and interleaving results in a 104-bit codeword that can correct up to 10 random errors or 9 burst errors. Finally, Code with Crosstalk Avoidance and Error Correction (CCAEC) code (Lakshmi et al., 2020) with an interesting error correction capacity of up to 12 errors was proposed by using two dimensions one of them is a horizontal simple parity vector where each parity bit is produced from one row of the input data block and another vertical parity check bits that are produced in a two-step. In the first step, the mask check bits for each row are calculated. In the second step, the vertical bits from the generated masked check bits are calculated. After duplication, the 104-bit codeword length is generated. According to the above techniques, error correction capacity is lower in techniques not joined with crosstalk avoidance. Although combining crosstalk avoidance with EDAC codes increased the error-correcting capacity, there was an increase in the bit overhead which led to an increase in the power consumption of the link.

This paper analyzes the reliability of the HVD code and CCAEC code by deriving a new accurate mathematical model for the probability of error residual for both coding schemes. This analysis compares the coding mentioned above schemes to evaluate the best scheme used in the NoC. Also, the simulation results are calculated using the Verilog code by Modelsim program to confirm the derived model.
2. METHODOLOGY

To analyze the HVD and CCAEC codes, we need first to clarify the mechanism of the encoder and decoder for each technique which can be introduced as follows:

2.1 Horizontal-Vertical-Diagonal Error Detecting and Correcting Code HVD.

The HVD code (Kishani et al., 2011) is a Hybrid Automatic Repeat Request HARQ that corrects specific errors and detects others without correcting according to the capability of the algorithm code scheme.

On the encoder side, the input data (M) is arranged in a matrix of (m x n) where m and n represent several rows and columns of the data block, respectively. From this matrix, four sets of parity check bits are derived, namely horizontal (H), vertical (V), slash diagonal (D), and backslash diagonal (D'), as shown in Fig. 1 for 64-bit input data. In addition, a parity check bit is added for each of the four parity check vectors. The encoder algorithm produces a 114-bit codeword.

The decoder algorithm depends on the syndrome of parity check bits which are commonly used in many coding schemes such as BKLC, BCH, Golay, and Hamming codes (Ahmed and Al-Hindawi, 2023) to detect or correct errors that may affect the transmitted data. If all syndrome of check bits is equal to zeros, this means either no error or an undetectable state. In contrast, if not equal to zeros, the algorithm detects up to 7 errors, or corrects up to 3 errors and detects up to 4 errors depending on the intersection among parity check bits, as shown in the algorithm in Fig. 2.
Figure 2. Decoder algorithm of HVD code.

2.2 Code with Crosstalk Avoidance and Error Correction for NoC (CCAEC).

In the CCAEC code (Lakshmi et al., 2020), the input data for M bits are arranged in an m×n matrix where n = 4 and m = M/n. Here m and n are the numbers of rows and columns, respectively. The number of columns stays constant for any size of input data. Horizontal and vertical parity check bits are coded for each row. The number of horizontal parity bits (H) equals m, and the number of vertical parity bits (V) is 3×m/2. For example, the 32-bit input data is arranged as 8×4 with 8 rows and 4 columns. The number of horizontal and
vertical parity bits is 8 and 12, respectively. Horizontal parity check bits are obtained directly for each row. Also, the masked parity bit is encoded from each row using adaptive hamming code. These parity bits are known as masked parity because they are not added to the codeword nor transmitted. Vertical parity bits are derived from these masked parity bits. Finally, the codeword consists of only data bits, horizontal parity bits, and vertical parity bits, which the sender transmits. Masked parity J bits are obtained as shown in Fig. 3 from the following equations (Lakshmi et al., 2020):

\[
J_i = M_i \oplus M_{i+24} \tag{1}
\]

\[
J_{i+1} = M_{i+8} \oplus M_{i+24} \tag{2}
\]

\[
J_{i+2} = M_{i+16} \oplus M_{i+24} \tag{3}
\]

Where \(\oplus\) is XOR logic operation, \(i=3n\) and \(n=0,1,2,...,7\).

The vertical parity bits are obtained by:

\[
V_Q = J_i \oplus J_{i+12} \tag{4}
\]

Where \(j = 0, 1, ..., 11\).

Hence, a codeword consisting of 52 bits consisting of 20 parity bits was added to the 32-bit input data, as shown in Fig. 3. Finally, to enhance crosstalk avoidance, the codeword was duplicated to become 104 bits before transmitting it. The decoder algorithm for this code is shown in Fig. 4. First, the received data is separated into two blocks. Calculating the syndrome for each type of parity check bit, comparing the syndrome of horizontal check bits to choose the copy with fewer errors, and then determining whether these errors are correct depends on the syndrome for both horizontal and vertical check bits.
Figure 4. The decoder of the CCAEC code (redrawn) (Lakshmi et al., 2020).

3. RELIABILITY ANALYSIS

One of the metrics to determine reliability is calculating the probability of residual error ($P_{res}$), which measures the reliability of the NoC of any EDAC technique (Flayyih et al., 2020).
$P_{\text{res}}$ is the probability of finding an error(s) in the received flit after completing the decoding process. That means the flit has an undetectable state which is out of the capability of the code scheme decoding algorithm, and this can occur in the first transmission or after retransmitting one or more times (Yu, 2009). In the following subsection, both HVD and CCAEC codes will be evaluated according to the calculation of the $P_{\text{res}}$ for random errors only because the burst errors are acceptable for both codes:

A. HVD Code.

Because it is a HARQ technique, this code type depends on correcting some errors and detecting others without correcting them according to the capability of scheme code; therefore, the $P_{\text{res}}$ is given as in (Flayyih et al., 2014):

$$P_{\text{res}} = P_{\text{und}} + P_{\text{und}} \times P_{\text{ret}} + P_{\text{und}} \times P_{\text{ret}}^2 + \cdots + P_{\text{und}} \times P_{\text{ret}}^n$$  \hspace{1cm} (5)

The $P_{\text{ret}}$ is the probability of retransmission, and $P_{\text{und}}$ is the probability of undetectable errors in the decoder. Eq. (5) can be simplified using geometric series reduction as:

$$P_{\text{res}} = \frac{P_{\text{und}}}{1 - P_{\text{ret}}}$$  \hspace{1cm} (6)

where $P_{\text{und}}$ for HVD is derived using Eq. (7) by summatting undetectable cases, which are very few cases the HVD decoder cannot detect. Where in the first case, when two of eight errors have happened in any one direction of HVD message bits matrix, row, column, slash, or back slash direction, which is represented in the first, second, and third term of Eq. (7), respectively, and other six errors are in parity check bits related to former two bits that make syndromes of them equal to zero as shown Fig. 5 (a), and (b). The second undetectable case, as shown in the fourth term of Eq. (7), is when four errors, each two of them are located in a different row in data bits. The other four bits have happened in parity check bits related to the former four errors to make all syndromes equal to zeros, as shown in Fig. 5 (c). The other cases, such as shown in Fig. 5 (d), where all eight errors occurring in message bits are neglected since there is little probability and to avoid complexity.

$$P_{\text{und}} = \{ \binom{n}{2}(\binom{m}{1})^2 + \binom{m}{2} \binom{n}{1} + \sum_{c=2}^{n-2} \binom{c}{2} \times 4 + \binom{n}{4} \binom{m}{1}^2 \} \times t_d^1$$  \hspace{1cm} (7)

In general,

$$\binom{y}{x} = \frac{y^x}{x!(y-x)!}$$  \hspace{1cm} (8)

Eq. (8) is a general mathematical form to calculate the possible combinations of $x$ elements from a set of $y$, where $x$ and $y$ are any two integer numbers and $t_d$ (for HVD $t_d = 7$) is the maximum error detection capacity. Such the HARQ technique can correct some errors and detect others according to the scope of code scheme capacity. Assuming that code can correct up to $t_c$ and can detect up to $t_d$ errors, then $P_{\text{ret}}$ can be written as:

$$P_{\text{ret}} = \sum_{t_c+1}^{t_d} P_{i\text{-error}}$$  \hspace{1cm} (9)

$P_{i\text{-error}}$ is the probability that an $L$-bit codeword (where $L$ for HVD equal to $m \times n + H + V + D + D' + 4 = 69$ bits for 32-bit input message) has $i$ errors, and $t_c$ (for HVD $t_c = 3$) is the maximum error correction capacity. Thus, $P_{i\text{-error}}$ is given by (Flayyih et al., 2018):

127
\[ P_{-\text{error}} = \left( \frac{L}{i} \right) \epsilon^i (1 - \epsilon)^{L-i} \tag{10} \]

For small \( \epsilon \), the probability of \((t_c + 1)\) errors dominates, and the Eq. (9) can be rounded to:

\[ P_{\text{ret}} = \left( \frac{L}{t_c + 1} \right) \epsilon^{t_c+1} \tag{11} \]

Finally, by substituting Eq. (7) and Eq. (11) in Eq. (6), \( P_{\text{res}} \) can be found for the HVD code easily.

---

**Figure 5.** Samples of undetectable cases of 8 errors for HVD 32 message bits decoder

(a) Double errors in one row (b) Double errors in one Diagonal (c) Quadruple of errors in data bits (d) Eight errors in data bits.
B. CCAEC Code.

The probability of residual error depends on the probability of retransmission of detected error \( P_{\text{ret}} \) and the probability of correction error concerning the technique type and capability of scheme code; the general form is given by (Fu and Ampadu, 2009) as:

\[
P_{\text{res}} = P_{\text{und}} + P(e_{\text{decoding}}, e_{\text{detecting}})
\]

\( P(e_{\text{decoding}}, e_{\text{detecting}}) \) is the Probability of error after the retransmission and decoding process is completed. Since CCAEC is a Forward Error Correction (FEC) technique capable of correcting all errors detected, resulting in \( P_{\text{ret}} = 0 \) and Probability of uncorrectable error \( (P_{\text{unc}}) = P_{\text{und}} \). So Eq. (5) and Eq. (12) become:

\[
P_{\text{res}} = P_{\text{unc}}
\]

Based on (Lakshmi et al., 2020), the CCAEC code can correct up to 12 random errors for 32 bits of input data, and \( P_{\text{res}} \) was given as:

\[
P_{\text{res}} = \left(2^{(k+m+\left(\frac{m+3}{2}\right))}\right)^{13}
\]

But this equation is inaccurate because the CCAEC decoding algorithm cannot correct all 12 random errors. We can prove that based on the minimum hamming distance \( d_{\text{min}} \), which is the minimum number of bits that can be changed to jump from a valid codeword to another valid one. The \( d_{\text{min}} \) is used to determine the maximum detection and correction capacity for any linear coding scheme by using equation \((d_{\text{min}}-1)/2\). For the CCAEC code, the \( d_{\text{min}} = (d_{\text{min}} \text{ for horizontal simple parity check vector} \times d_{\text{min}} \text{ for vertical simple parity check vector}) \) (Asaad et al., 2020) produce as \( d_{\text{min}} = 2 \times 2 = 4 \). So, the maximum correcting capability for the CCAEC code is \((4-1)/2 = 1\). As well as for this theoretical limitation and based on the CCAEC algorithm (Lakshmi et al., 2020), Fig. 6 shows some cases of two, three, and four errors in which the CCAEC code fails to correct them.

So the new accurate estimated model was derived based on the decoding algorithm, as given in Fig. 4. After duplicated codeword is received, the decoder separates it into two copies. And then, select the copy with the least number of 1’s in its syndrome horizontal check bits and when all syndrome of any of two copies equals zero. On the other side, if both copies are equal in the number of 1’s, the decoder will select any of them as a default copy. To simplify, the two copies will be denoted as copies A and B; when both are equal in the number of 1’s, copy A will be considered a default copy. As a result, Eq.(16) through Eq.(18) represent the uncorrectable error probability of CCAEC code where Eq.(16) expresses undetectable double error probability \( (P_{\text{unc2}}) \) when two errors are located in one row in copy A as shown in Fig. 6 (a). The following equation represents three undetectable errors probability \( (P_{\text{unc3}}) \). In the first term, two errors are located in the same row in the default copy, and the third error is anywhere in the duplicated codeword except the message bits and horizontal check bits of copy A. The second term is when two errors are in one row in copy B, and the third error is in message bits and horizontal check bits of copy A. In the third term, when one error occurs in copy A, specifically in the last bit of any row in the message bits or in the horizontal check bit, the second error is in the vertical check bit which is related to the first error in
copy A, and the third error happens anywhere in message bits or horizontal check bits of copy B.

**Figure 6.** Samples of fail cases for (a) CCAEC decoder, (b) Double errors, (c) Triple errors, and (d) Quadruple errors.
In the last term, one error locates any row’s first three bits. The second error is in the vertical check bit, related to the former first error in copy A. The third error happens anywhere in the message bits and horizontal check bits of copy B, examples of three undetectable errors, as shown in Fig. 6 (b) and (c).

The last equation is related to four uncorrectable errors probability ($P_{unc4}$). The first case is when two errors are in the same row in the message bits of the default copy, and the other two errors are located anywhere in the codeword except the message bits of the default copy. The second term is when two errors are in the same row in the message bits of the default copy, and the other two errors, one of them is located anywhere in the message bits or horizontal check bits of copy A, and the other error is anywhere in the message bits or horizontal check bits of copy B. Finally, the last term is if two errors are in the same row in the message bits and horizontal check bits of copy B, and the other two errors, one of them is anywhere in the message bits. Horizontal check bits of copy A and the other error are anywhere in the codeword except the message bits and horizontal check bits of copy B, as shown in Fig. 6 (d). However, some cases are ignored, especially in the four error case, because it has a very small error bit rate and also to avoid the complexity of the equation.

Finally, The $P_{res}$ for the upper bound of CCAEC can be written as:

$$P_{res} = \sum_{i=2}^{4} P_{unci}$$  \hspace{1cm} (15)

Where $i$ is given from $i=2$ up to 4 errors.

Where $P_{unc}$ can express by:

$$P_{unc2} = \binom{(m+1)}{2} \binom{n}{1} \epsilon^2$$  \hspace{1cm} (16)

$$P_{unc3} = \left[ \binom{(m+1)}{2} \binom{n}{1} \left( \frac{n \times (m+1) + 2 \times \nu_r \times \nu_c}{1} \right) + \binom{(m+1)}{2} \binom{n}{1} \binom{n \times (m+1)}{1} + \frac{(\nu_c)}{1} \binom{n \times (m+1)}{1} \right] \epsilon^3$$  \hspace{1cm} (17)

$$P_{unc4} = \left[ \binom{(m+1)}{2} \binom{n \times (m+1) + 2 \times \nu_r \times \nu_c}{1} \binom{n \times (m+1) + 2 \times \nu_r \times \nu_c}{1} \binom{n}{1} + \binom{(m+1)}{2} \binom{n \times (m+1)}{1} \binom{n \times (m+1) + 2 \times \nu_r \times \nu_c}{1} \binom{n}{1} \right] \epsilon^4$$  \hspace{1cm} (18)

Where $\nu_r$ and $\nu_c$ are the number of rows and columns for the matrix of vertical parity check bits, respectively.

By substituting Eq. (16), Eq. (17), and Eq. (18) in Eq. (15), the $P_{res}$ of CCAEC can be found.

### 4. RESULTS AND DISCUSSION

For more analysis, simulation results were done by Verilog code under the Modelsim program for both code schemes, as shown in Table 1. Where $10^6$ random samples of 32-bit input data are injected into both code schemes with different numbers of errors, and the failure percentage is given (Asaad et al., 2020):
Failure percentage = \frac{\text{the number of uncorrected samples}}{\text{the total number of samples}} \times 100\% \quad (19)

Injected samples are fed with one, two, three, four, and five errors randomly located in the transmitted codeword. As shown in Table 1, it is clear that for a single error, both schemes can correct all samples with a single error. Then, for samples with double and triple random errors, the HVD code can correct all of them, while the CCAEC code fails to correct 1.5% and 7.2% for double and triple errors, respectively. Similarly, the HVD code could not correct 3.3% of quadruple errors, and the other code could not correct 16.4% of the total samples. Since the HVD code is HARQ technique, the undetectable error probability \( P_{\text{und}} \) must be found because it's important in \( P_{\text{res}} \) calculation where it appears in case of 8 errors, and its value was \( 0.4 \times 10^{-7} \) in simulation results.

**Table 1.** Simulation results.

<table>
<thead>
<tr>
<th>Numbers of errors</th>
<th>HVD Failure Percentage (%)</th>
<th>CCAEC Failure Percentage (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1.5</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>7.2</td>
</tr>
<tr>
<td>4</td>
<td>3.3</td>
<td>16.4</td>
</tr>
</tbody>
</table>

**Fig. 7.** represents the simulation and estimation of the probability of residual error with respect to different values of bit error rate where the simulation is done by Verilog language under Modelsim program for both scheme codes and injected \( 10^7 \) samples of 32 data bits for each number of errors located in random position for each scheme algorithm and found failure rate for each of them and then multiply each of these rates by error bit rate with the power of the number of errors then make summation for them to find simulated \( P_{\text{res}} \). And we can observe very little difference between estimation and simulation results, which confirms the derived model’s validity.

**Figure 7.** \( P_{\text{res}} \) with respect to the bit error rate.
As well as it is clear to notice as given in Table 2. The bit overhead for CCAEC is more than the HVD code due to duplicating the codeword to reduce the crosstalk effect. Also, as shown in the same table, the code rate of HVD is higher than CCAEC. According to the reliability analysis results and both of the last two previous features and correct on capacity, the HVD coding is considered a better choice than the CCAEC code.

Finally, when taking the same value of $P_{\text{res}}$ for both scheme codes with respect to the error bit rate, the HVD refers to an error bit rate more than CCAEC, which is proportional to voltage swing that leads to a reduction in the power consumption in HVD as according to Gaussian noise model which is discussed in more details in many research papers (Rahimipour et al., 2020; Asaad et al., 2020). In Table 2, bit overhead and code rate was calculated, and it is clear that HVD has values better than CCAEC code. These rates affect the area and power consumption of the Network on the chip.

**Table 2.** Bit overhead and code rate results.

<table>
<thead>
<tr>
<th>Method</th>
<th>Input data (bits)</th>
<th>Redundant (bits)</th>
<th>Codeword length (bits)</th>
<th>Bit Overhead (%)</th>
<th>Code Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HVD</td>
<td>32</td>
<td>37</td>
<td>69</td>
<td>115.63</td>
<td>45.71</td>
</tr>
<tr>
<td>CCAEC</td>
<td>32</td>
<td>20</td>
<td>104 (duplicated)</td>
<td>125</td>
<td>30.77</td>
</tr>
</tbody>
</table>

### 5. CONCLUSIONS

In this paper, a new accurate mathematical model for the probability of residual error of HVD and CCAED codes was derived and used to analyze the reliability of these two codes. After comparing the reliability analysis results of the HVD and CCAEC codes, it was found that the CCAEC results of the new estimation model are very close to the simulation results, which confirms the newly derived model. This confirms the inaccuracy of the old estimation results and invalidates the claim that it can correct 12 errors as it fails to correct some patterns of two errors. The HVD method was found to have higher reliability than CCAEC due to correction and detection capacity, where HVD can correct all messages with two and three errors and 96.7 % of total messages with four errors. In contrast, the CCAEC code can correct 98.5%, 92.8%, and 83.6% of all messages with two, three, and four errors, respectively. As a result of the analysis of the results mentioned above, the CCAEC code has high power consumption due to its high voltage swing compared with the HVD code. Finally, the HVD code remains a more reliable code and still an efficient code to handle the reliability issues for NoC. It can improve its performance by using a simple crosstalk avoidance method, such as increasing link spacing to equal other codes in this important feature.

### NOMENCLATURE

- $d_{\text{min}}$: minimum hamming distance.
- $P_{\text{res}}$: probability of residual error.
- $P_{\text{und}}$: probability of undetectable error.
REFERENCES


