DESIGN AND VERIFICATION OF A 4-BIT ERROR CORRECTING DECODER USING BCH CODING TECHNIQUE

R.Vasanth* and V.Balamurugan

Department of Electronics and Communication Engineering, Sathyabama University, Chennai
E-mail:*vasvasanth.ece@gmail.com; vbbala@gmail.com

ABSTRACT
The scope of this paper deals with the design of a high-speed BCH decoder that corrects single-bit errors in parallel manner and double, double adjacent, triple and four bit errors in serial manner. The proposed decoder is capable of detecting, locating and correcting single bit to four bit errors that occur in the high speed memory operations and data transmission. If we consider the case of the Nano-scale memory system the probability of occurrence of the multi bit errors is high, in such case the proposed decoder is capable enough to correct the error at the faster rate. In the existing system only the single and double errors are detected and corrected and locating and correcting single bit to four bit errors that occur in the high speed memory operations and data transmission. If we consider the case of the Nano-scale memory system the probability of occurrence of the multi bit errors is high, in such case the proposed decoder is capable enough to correct the error at the faster rate. In the existing system only the single and double errors are detected and corrected, whereas in the proposed scheme single to four bit errors are detected, located and corrected using BCH coding technique. In the evolution of the high speed digital communication there is high chance for occurrence of the data message to get corrupted during transmission and reception via noisy channels, hence for the error free communication error correcting codes are required, one such efficient coding is the BCH code which has well defined mathematical properties. The encoder and decoder is designed to detect and correct single to four bit errors using VHDL and simulation is performed using Xilinx ISE 14.3.

Index Terms—BCH (Bose Chaudhuri Hocquenghem) coding, ECC – Error Correcting codes, LDPC

I. INTRODUCTION
There is a rapid evolution and advancement in the usage of the internet using the Laptop’s, smart phones and other portable devices hitting billions and billions of web pages that are hosted in the internet based on the respective request. The data traffic in the internet is consumed by various services like email, ftp request, IIS request, http request, https request, VPN, public and various other private network traffics. The in-turn overloads the data provider to be dumped with the data usage of several terra bytes per day. The data transmission protocol i.e. the request and response between network hub and user occurs via various communication medium such as Optical Fiber cables, Copper wires, wireless links (micro wave), FSOC and other communication, in such huge data usage in this heavy traffic that is increasing day by day it is inevitable to ensure reliable communication in each communication medium. In such high data transfer rate and huge traffic, occurrence of a small bit error may cause a catastrophic data loss, hence in order to ensure reliability of the data communication we need to have Error correcting codes designed for the noisy channels that has high Bit error rate and less SNR.

Each and every communication medium has its own advantage and disadvantages, the designed error correcting code should be capable enough to handle the error incurred by the noisy channel. The Error correcting codes are used to correct the error in the received data caused by the noisy channel. ECC has its own application areas in high speed memory system, digital data communication and other efficient fault tolerant digital system design. Based on the error expected, medium of communication and various other aspects different coding schemes can be used, few of them are, Hamming code, Low Density Parity Check code (LDPC), Bose-Chaudhuri Hocquenghem code (BCH), Reed Solomon code, Turbo code. This paper deals with the use of BCH coding to detect, locate and correct single to four bit errors.

II. LITERATURE SURVEY
Kazuteru Namba et al., (1) proposed a BCH decoder that corrects double-adjacent and single-bit errors in parallel which resembled a Wilker’s parallel BCH decoder which is used to correct only single-bit errors. The decoder section is operated serially for a double and double adjacent error where in the memory system the probability of occurrence of single and multi-bit error occurs in the memory system.

Amit Kumar et al., (2) presented a design of a (15,k) BCH encoder on FPGA with multi bit error correction control, in which the primitive polynomial is implemented in the Linear Shift Register logic. By the use of the cyclic codes, the reminder parameter b(x) is calculated in a LFSR (15,k) with feedback connections which corresponds to the coefficient of the polynomial generated. In this technique three encoders are designed to encode the single, double and triple error correcting BCH codes.

III. REVIEW OF BCH CODES
The BCH codes is an efficient Error correcting code which is a polynomial code capable of correcting combination of ‘e’ or lesser errors in the block of n=2^m-1 digits.

BCH Encoder
A (n, k) binary BCH code is capable to encode k-bits of message into n-bits of code word. The message vector is expressed in polynomial form as below,

\[ m(x) = m_0 + m_1 x + m_2 x^2 + \cdots + m_{k-1} x^{k-1} \]  \hspace{1cm} (1)

The code word comprises of the message digits. The encoding scheme is implemented by the below expression:

\[ c(x) = p(x) + x^{n-k} m(x) \]

Here the reminder is p(x) and the same can be denoted as,

\[ p(X) = x^{n-k} m(x) \mod g(x) \]  \hspace{1cm} (2)

Based on the definition of the “e” error correcting BCH code for length n=2^m-1, each code has a polynomial and
has the roots as $\alpha, \alpha^2, \ldots, \alpha^{2t}$ for $c(\alpha^i) = 0$, for $(1 \leq i \leq 2t)$.

It follows from the definition of a t-error-correcting BCH code of length $n=2^m-1$, that each code polynomial $c(x)$ is normally syndrome that satisfies the two equations $(3)$ and $(5)$ appear as a correct code word, or as a double-adjacent error; the syndrome of any correctable and detectable error except double-adjacent errors does not satisfy these two equations.

**BCH Decoder**

Existence of the efficient decoding methods based on the specialized algebraic structure hosted in the BCH code provides a biggest advantage. If for a code word $c(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1}$ is transmitted and if the error pattern results in the below revised vector $r(x) = c(x) + e(x)$, where $e(x)$ is the error pattern. If the error pattern $e(x)$ has 'v' errors at locations $X^{j_1}, X^{j_2}, \ldots, X^{j_v}$ that is, $e(x) = X^{j_1} + X^{j_2} + \cdots + X^{j_v}$, Where $0 \leq j_1 < j_2 < \cdots < j_v < n$. Since $v(\alpha^i) = 0$, then $r(\alpha^i) = e(\alpha^i)$. Then algebraic decoding of the BCH codes has the following steps:

1. **Step 1: Syndrome Calculation**
2. **Step 2: Calculation of error location polynomial $\sigma(x)$, the roots of which indicates where the errors are located.**
3. **Step 3: The roots of the error locating polynomial has to be derived, which is normally done using the Chien search algorithm, which performs an exhaustive search above all the components in the field.**

**IV. RELATED WORK**

The Kazuteru Namba et al., (1) have proposed a single and double adjacent error correcting BCH code decoder. The double adjacent error is a kind of a double-bit error, in the occurrence of which the values at two adjacent bits erroneously change.

The generic design of the high speed decoder is shown in figure Fig 1.

![Diagram of a BCH decoder](image)

**Fig 1. Diagram of a BCH decoder**

In this implementation for a double-adjacent error that occurs at $i^\text{th}$ and $(i+1)^\text{th}$ bits, the syndrome appears as,

$$s_{\text{parity}} = 0$$

$$s_1 = \alpha^j + \alpha^{j+1}$$

$$s_2 = \alpha^{(j+1)}$$

$$s_3 = (s_j + \alpha^{(j+1)}) (\alpha + 1)$$

The double-adjacent bit error occurrence is detected by verifying that the equations (3) and (5) are valid. As well, the location of the error is established from the equation (4). The error type is determined by the syndrome $s_{\text{parity}}$. If the $s_{\text{parity}} = 0$ then the error is double adjacent error, else if the $s_{\text{parity}} = 1$ then the errors occurrence is single-bit error. If there is no error, then $s_{\text{parity}} = 0$ and $s_1 = s_3 = 0$.

Each and every syndrome that satisfies the two equations (3 & 5) appear as a correct code word, or as a double-adjacent error; the syndrome of any correctable and detectable error except double-adjacent errors does not satisfy these two equations.

![Parallel Decoder for double error correction](image)

**Fig. 3. Parallel Decoder for double error correction**

Fig. 3 illustrates an example of the construction of the parallel decoder for double-error correcting BCH codes. It consists of a syndrome generator, an error pattern generator and an error detector.

- The syndrome generator generates the syndrome $s = \text{from a received word } v$.
- The error pattern generator generates the error pattern $e$, and the decoder then outputs $(v + e)$ as a decoded word.
- The error detector detects un-correctable errors, i.e., errors that are neither single-bit nor double-adjacent.

**V. PROPOSE METHODOLOGY**

In the proposed work, the designed decoder is capable of detecting, locating and correcting single bit error, double & double adjacent error, triple error and four bit error based on the message overriding random error correction technique. The existing work deals with the usage of the systematic matrix generation to correct single and double adjacent error whereas the current work deals with error detection, location and correction using message override random error correction methodology up to four bit error. In this method single bit error is corrected in parallel manner in one clock cycle, double bit error and double adjacent error is corrected serially in two clock cycles, triple bit error is corrected in three clock cycle, four bit error is corrected in four clock cycle.
Figure Fig 4 shows the conceptual block diagram of the typical Error correcting code used by the BCH coding technique.

In the proposed technique the Syndrome generator is used to generate the Syndrome code at the encoder section and the Syndrome checker is used to detect and locate the error at the decoder side. The syndrome is calculated based on the BCH coding technique.

The encoded data is received at the decoder end and the error is corrected based on the message override random matrix error correction technique. Here the decoded bit is random matrix Exor with the code word to find the type of error it has. If the error is single bit, then the received data is iterated with the code word to form a random matrix to locate the erroneous bit and invert the same. In case of the double and double adjacent, the iteration for a single bit is done once and the output of the same is iterated again to locate the second erroneous bit and invert the same.

In the case of the triple error the output from the double is iterated again with code word to correct triple error and for four bit error the output from triple block is iterated again to correct the four bit error.

VI. RESULTS AND DISCUSSION

A Pseudo code to explain the above correction methodology is below. BCH coding is used to calculate the syndrome at the encoder side. Based on the encoded data received the error is found based on the message overwrite error detection method shown in the below pseudo code,

RX_message= Received message from encoder;
Error_flag = Syndrome checker();
if (Error_flag < 5)
    {for loop ( Iterate the ith bit value from 15th bit to 0th bit as ‘1’)
        {decoded_data=RX_message XOR (0000000000000000i);
         Invert the erroneous bit;}}
else
    {Set the output error correction flag as null for no error;}

Hence based on the above approach the single to four bit errors can be detected, located and corrected.
Fig 7. Double adjacent Bit Error correction

Fig 8. Triple Bit Error correction

Fig 9. Four Bit Error correction

Design summary of the proposed design is tabulated below, which shows the usage of the IO’s is less compared to the existing design.

Table 1 Device utilization summary

<table>
<thead>
<tr>
<th>Logic Utilization</th>
<th>Used</th>
<th>Available</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Slices</td>
<td>325</td>
<td>4656</td>
<td>6%</td>
</tr>
<tr>
<td>Number of slice flip flops</td>
<td>88</td>
<td>9312</td>
<td>0%</td>
</tr>
<tr>
<td>Number of 4 input LUT’s</td>
<td>588</td>
<td>9312</td>
<td>6%</td>
</tr>
<tr>
<td>Number of bonded IO’s</td>
<td>110</td>
<td>158</td>
<td>69%</td>
</tr>
<tr>
<td>Number of GCLK’s</td>
<td>1</td>
<td>24</td>
<td>4%</td>
</tr>
</tbody>
</table>

VII. CONCLUSION
This paper presented the design of a BCH decoder for detecting, locating and correcting errors up to 4 bits compared to the existing scheme which is capable to correct only two bit errors and incapable of locating the error. The hardware overhead of the proposed scheme is comparable to the existing work as the number of IO’s consumed is less in the proposed scheme compared to the existing. BCH codes have exhibited an excellent error correcting capability compared to other error correction coding which ensures a reliable communication in the data transmission and high speed memory operation. In the proposed decoder only single bit error is corrected in parallel manner, rest of the bits are corrected serially in-order to reduce the high hardware utilization compromising on the speed, as fully parallelized decoder has increased hardware overhead. The future road map would be to design a fully parallel mode decoder with less hardware utilization using various other coding techniques.

VIII. REFERENCES