A Method of Error Detection and Correction Using Euclidean Geometry Low Density Parity Check Codes

Vrinda S Anand[1], Sukanya Sundaresh[2]

[1]P G Scholar, VLSI and Embedded Systems, Department of ECE, T K M Institute of Technology, Kollam, India

[2]Assistant Professor, Department of ECE, T K M Institute of Technology, Kollam, India

Abstract*: **Data transmission is the physical transfer of data over the communication channel. Error correcting coding has become one of the essential part in the modern data transmission and storage systems. During transmission, there is a chance of error affecting on channel. So the error correction codes are used to reduces the number of errors in the channel. Various error correction codes can be used to detect and correct the errors. Low Density Parity Check (LDPC) codes are a class of linear block codes, has the superior performance in error detection and correction. This method presents an error detection and correction method using Euclidean Geometry Low Density Parity Check codes (EG-LDPC) with majority logic decoding. EG-LDPC codes are the class of Low Density Parity Check codes for detecting and correcting the errors. It is based on the bit flipping algorithm. The bit flipping algorithm utilizes the check sum equations in order to detect and correct the errors. The VHDL language is used for coding. Synthesis is done in Xilinx ISE design suite 13.2 and ModelSim 6.3f is used for simulation.***

Keywords: Error correction code, Majority logic decoding, LDPC, Euclidean Geometry Low Density Parity Check Codes (EG-LDPC), Bit flipping.

1. Introduction

In digital communication [4], data transmission is the active process of transporting a data from sender to receiver. Environmental interference and physical defects in the communication medium can cause random bit error during data transmission. Error coding is a method of detecting and correcting these errors to ensure information transferred from its source to destination. In order to detect these errors Error Correction Code (ECC) [6] can be used.

Error detection and correction is also known as channel coding [7]. Channel coding is the process of coding data to transmit it over a communication channel. So that if error occurs during transmission it is possible to detect and correct those errors once the data has been received. In order to achieve this error detection or correction some bit pattern will be identified as error free at the receiver, whereas other bit pattern will be identified as erroneous. To increase the number of identifiable bit patterns at the receiver to represent the data, additional bit known as redundant bits are added to the data or information bit. Various types of error correction codes [10] are available in channel coding. Error correction code has two types: block code and convolution code. Block code consist of various codes such as Hamming code [2], Reed Solomon code [3], Low Density Parity Check (LDPC) code [9] etc.

LDPC codes are a class of linear block codes, in which the name comes from the characteristics of their parity check matrix which contains only a few 1s in comparison to the amount of 0s. Generally block codes are decoded using Majority Logic (ML) decoding. In LDPC majority logic decoding is used. For a large code word [8] , LDPC codes are usually decoded using bit flipping algorithm. This method mainly focus on the error detection and correction using Euclidean Geometry Low Density Parity Check (EG-LDPC) codes [1].

2. Theory

A general communication system transmits information data from a source to a destination, through a specific channel or medium. The received data at the destination could be different with respect to the original data sent at the source, and this because of the channel distortion or the external noise. To avoid data distortion a more complex structure is used, which includes also an encoder and a decoder. The channel encoder introduces redundancy to the source information such that the transmission is more reliable. The channel decoder recovers the original information data from the received data. In other words the encoder transforms a sequence of information symbols into a codeword, and the decoder retrieves the original data sequence from the received codeword. There are two main kinds of channel coding techniques. The first kind is called Automatic Repeat Request (ARQ) and in this case the receiver requests for a retransmission of the data if the received codeword is unreliable. The second kind is called Forward Error Correction (FEC), it is a technique used for controlling errors in data transmission over a noisy communication channel.FEC gives the receiver the ability to correct errors without needing a reverse channel to request retransmission of data. Types of FEC codes are

1. Block codes
2. Convolution codes

2.1 Low Density Parity Check Codes

A low-density parity-check (LDPC) code is a code defined by a parity-check matrix with low density ,ie., the parity check matrix H has a low number of 1's. LDPC codes are constructed using the parity-check matrix not generator matrix. Codes can be constructed using either the matrix itself or its graphical representation, the Tanner graph. Consider the parity check matrix [m*n]. m is the number of rows and n is the number of columns. It should have wc ones in every columns and wr ones in every rows. The low density means wc ones in every columns are less than or equal to number of rows and wr ones in every rows are less than or equal to number of columns.

For example, Consider [4*8] parity check matrix is given below. It consists

- number of rows m = 4
- number of columns n = 8
- ones in every column wc = 2
- ones in every rows wr = 4

ie,wc <= m(2 <= 4), wr <= n(4 <= 8)
So the given matrix is low density parity check matrix.

$$H = \begin{array}{cccccccc|c} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 & V_8 & \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & C_1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & C_2 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & C_3 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & C_4 \end{array}$$

Figure 1: Parity check matrix

2.2 Graphical representation of LDPC codes

The parity check matrix can also be represented in a graphical form known as a Tanner graph. A Tanner graph consists of two sets of vertices: n vertices for the codeword bits (bit nodes), and m vertices for the parity-check equations (check nodes). An edge joins a bit node to a check node. In tanner graph, first, create a check node (squares denoted by Cs) for every row in H. Next, create a variable node (circles denoted by Vs) for every column in H. Where there is a 1 in H, draw a line between a check node and variable node that corresponds to its location.

For example if consider a check node C1, there is a 1 in the first row and second column of H, so a line is drawn connecting C1 to V2.Then consider 1 in the first row and 4thcolumn of H, line drawn from C1 to V4 and so on. The graphical representation of the above parity check matrix is shown in figure 2.

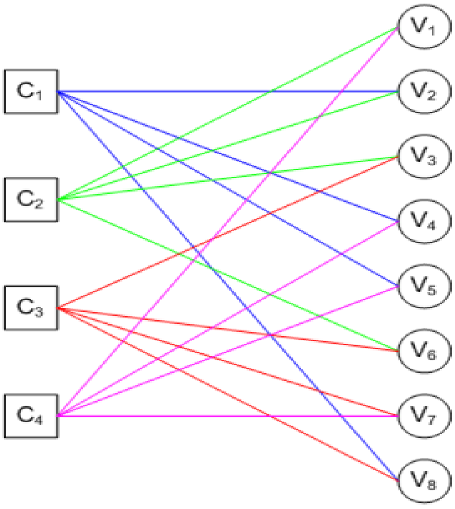

Figure 2: Tanner graph

3. Methodology

Single bit error, double bit error and multiple bit error detection codes are present in digital circuits. The encoding and decoding is used in this type of codes for detecting and correcting the errors. The majority logic (ML) decoder is used for detection and correction of errors. The ML decoder has error control capability that is to detect and correct the error occur in the codeword. The block diagram of ML decoder is shown below:

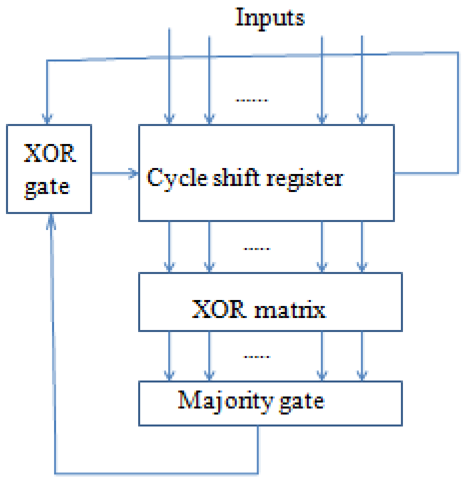

Figure 3: Block diagram of ML decoder

Based on the number of parity checksum equation the ML decoder has the ability of correcting multiple bit flips. A sub group of Low Density Parity Check codes are used which belongs the family of ML decoder. It consists of four parts:

- A cyclic shift register
- An XOR matrix
- A majority gate
- A XOR gate for correcting the codeword bit under decoding

An XOR matrix gives the parity checksum equation. Parity checksum equation is the XOR operation of the information bit and parity bit. Schematic of majority logic decoder is shown below:

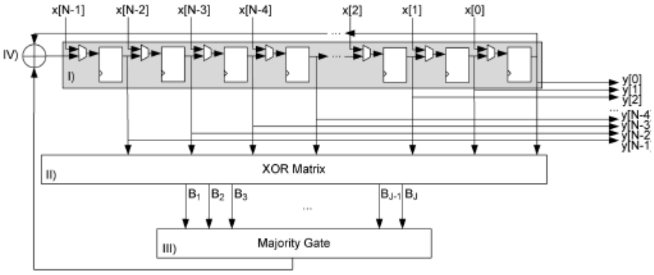

Figure 4: Schematic of majority logic decoder

Initially the codeword bits are stored in the cyclic shift registers and shifted to all the registers. It circulates all codeword bits of the register around both MSB and LSB ends with no loss of information. The bits in between each register blocks are used to calculate the output results of the checksum equation from the XOR matrix. The result has reached the final block, producing final output. After the initial step, in that the codeword is loaded into the cyclic shift registers, the decoder starts to calculate the parity check equation from the XOR matrix. Syndrome fault detector block is used here. It is an XOR matrix that calculates the syndrome based on parity check matrix. If any error occur in the data, the syndrome calculate the error. ie, when atleast one of the syndrome bit is one the faulty codeword is detected. This triggers the majority logic decoder to start the decoding. The resulting value is forward into the majority gate for calculating the correctness. If the number of 0's is less than number of 1's,that indicating that the current bit under decoding is wrong. Otherwise no extra operation is needed which means the current bit as error free. Finally the parity checksum should be zero if the codeword has been correctly decoded.

3.1 Cyclic shift register

Cyclic shift register is used to shift the data cyclically. For an N- bit input data, decoder requires N tap shift register. The cyclic shift register consists of two modules as follows:

- D flip- flop
- Multiplexer

The input codeword bits are stored in the cyclic shift registers and shifted through all the taps. The cyclic shift register is designed in parallel in serial out fashion. Initially the selection pin of each mux is kept low and the data is loaded into the flip flops and then selection pin is kept high, during this time the input signal is shifted. And input of the first flip flop is XORed result of the LSB and output of majority gate.

3.2 XOR matrix

XOR matrix is based on the parity check sum equations. An example of calculating parity check equation is as follows:
Consider a code of length 6:
x = (x1; x2; x3; x4; x5; x6)

Suppose that :
x1 XOR x2 XOR x3 XOR x4=0
x2 XOR x3 XOR x5=0
x1 XOR x3 XOR x6 =0
Assign any values to x1,x2,x3 solve for x4,x5,x6

$$H=\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Figure 5: Parity check matrix

3.3 Majority Gate

Majority gate evaluate the result of parity check sum such that if the number of1's is greater than number of 0's means the decoding current bit is wrong, and it would be triggered. The output of the majority gate is the majority result of the parity checksum equations.

3.4 LDPC Decoding

- Bit Flipping algorithm

For the bit-flipping algorithm the messages passed along the Tanner graph edges are also binary: a bit node sends a message declaring if it is a one or a zero, and each check node sends a message to each connected bit node, declaring what value the bit is based on the information available to the check node. The check node determines that its parity check equation is satisfied if the modulo-2 sum of the incoming bit values is zero. If the majority of the messages received by a bit node are different from its received value the bit node changes (flips) its current value. This process is repeated until all of the parity check equations are satisfied, or until some maximum number of decoder iterations has passed and the decoder gives up. The bit-flipping decoder can be immediately terminated whenever a valid codeword has been found by checking if all of the parity-check equations are satisfied.

The Bit-flipping algorithm does is to compute the syndrome s and if s is a zero vector then the algorithm stops, otherwise it looks for the bit to flip in the codeword. In this the syndrome is calculated by verifying that all the parity check equations are satisfied: for each matrix row (that is for each check node) a XOR operation is made between all the codeword bits in the same positions of the ones in the row (or the codeword bits connected to the same check node). A 0 as final result means that a valid codeword has been found and the algorithm stops, a 1 means that not all parity check equations are satisfied (there is at least a wrong one) and so the algorithm goes on.

4. Results and Discussion

The modules are modeled using VHDL in Xilinx ISE design suite 13.2 and the simulation of the design is performed using ModelSim SE 6.3f to verify the functionality of the design.

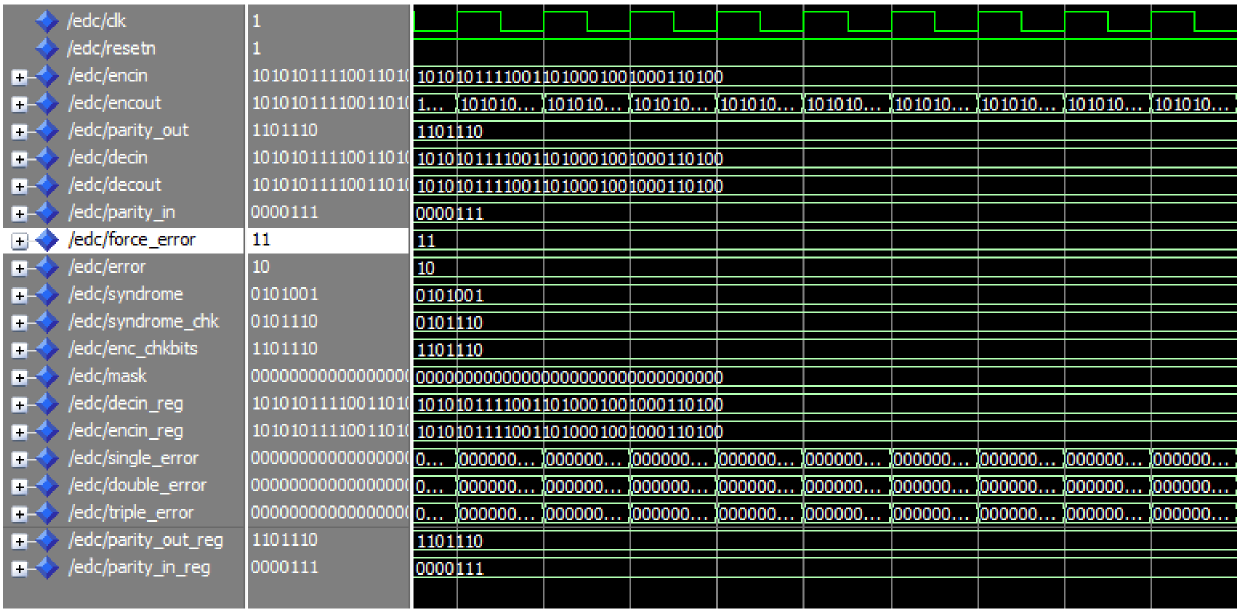
/edc/clk 1
/edc/resetn 1
/edc/encin 101010111100110101000100100011010
/edc/encout 101010111100110101000100100011010
/edc/parity_out 1101110
/edc/decin 101010111100110101000100100011010
/edc/decout 101010111100110101000100100011010
/edc/parity_in 0000111
/edc/force_error 11
/edc/error 10
/edc/syndrome 0101001
/edc/syndrome_chk 0101110
/edc/enc_chkbits 1101110
/edc/mask 0000000000000000000000000000000
/edc/decin_reg 101010111100110101000100100011010
/edc/encin_reg 101010111100110101000100100011010
/edc/single_error 000000000000000
/edc/double_error 000000000000000
/edc/triple_error 000000000000000
/edc/parity_out_reg 1101110
/edc/parity_in_reg 0000111

Figure 6: Error detection and correction using ML decoder in binary

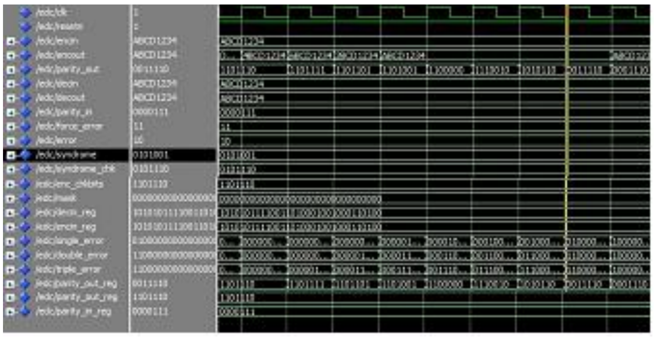

Figure 7: Error Detection and Correction Using Majority Logic decoder in hexadecimal

The codeword (input along with the parity bits) is given as input. Syndrome fault detector is used to detect the errors by multiplying the incoming codewords by the transposed parity matrix. All zero syndrome means no error has occurred. If any error occurs, the syndrome calculate the error and it will correct. In this an error signal is forced in the codeword that is called force error signal. When the value of force error signal is ’00’ means no error is added in the codeword and there will be no change in the codeword. But when the value of force signal is ’01’ means single error added ,’10’ means double error added, ’11’ means tripple error added in the codeword and it will be changed. These are corrected by using syndrome.ie, it will invert the corrupted data in the codewords.

5. Conclusion

The error detection and correction are important in data communication. Various error correction codes such as Hamming codes, Reed Solomon codes, Low Density Parity Check codes (LDPC) etc are used for detecting and correcting the errors. Hamming code is used to detect double bit errors and it can correct single bit error. But it cannot correct the multiple errors. Reed Solomon codes are used for correcting multiple errors, but the complexity and computational cost is very high. So in this method, LDPC codes are used to detect and correct the multiple errors. LDPC means low density of ones in the parity check matrix. These codes are based on the check sum equation of parity check matrix by using XOR operation. This work is synthesized in Xilinx ISE design suite 13.2 and simulated in ModelSim 6.3f.

References

[1] Pedro Reviriego, Juan A. Maestro, and Mark F. Flanagan, “Error Detection in Majority Logic Decoding of Euclidean Geometry Low Density Parity Check (EG-LDPC) Codes,” IEEE Trans. Very Large Scale Integration.(VLSI) Syst.. vol. 21, no. 1,Jan 2013..

[2] Himanshu Sharma, Amit Kumar, “Hamming Code for Error Detection and Correction using VHDL”,International Journal Of Engineering Research & Management Technology (IJEMRT), vol.1,march 2014.

[3] Priyanka Shrivastava, Uday Pratap Singh, “ Error Detection and Correction Using Reed Solomon Codes” International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, August 2013.

[4] Simon Haykin, “Digital Communications” Wiley Global Education, 15-Feb-2013.

[5] H. Naeimi,A. DeHon, ”Fault secure encoder and decoder for Nano Memory applications,” IEEE Trans. Very Large Scale Integration.(VLSI) Syst.. vol. 17, no. 4,April 2009.

[6] S. Lin and D. J. Costello,” Error Control Coding,” 2nd ed. Englewood Cliffs, NJ: Prentice- Hall, 2004.

[7] Michael Duck, Richard Read, Data communication and computer networks , second edition., Pearson/Prentice Hall, 2003.

[8] Y. Kou, S. Lin, and M. P. C. Fossorier, ”Low-density parity-check codes based on finite geometries,” IEEE Trans. Inf. Theory, vol. 47, no. 7, Jul. 2001.

[9] R.G. Gallager, “Low-Density Parity-Check Codes,” Cambridge, MA:MIT Press, 1963.

[10] I.S Reed,” A class of multiple error correcting codes and the decoding scheme,” IRE Trans. Inf. Theory, vol. 49, 1954.