International Journal of Electrical and Electronics Engineers
Vol. No. 9, Issue No. 01, January-June 2017
ISSN (O) 2321-2055
ISSN (P) 2321-2045

A NOVEL APPROACH FOR OPTIMIZING THE DESIGN OF MODERN ERROR CONTROL CODE

Prof. Dhanisha Khatri[1] , Mr. Hemanshu Khatri[2]

[1]Electronics and TeleCommunication Department, Army Institute of Technology, Pune (INDIA)

[2] R&D Department, Power Electronics, Pune (INDIA)

ABSTRACT

Power optimization in data communication is an open challenging issue for VLSI designer. The demand of high data storage and transmission along with good lifetime and compact size, forces to develop coding scheme with long lasting power performance. In this work an attempt is done to develop minimum distance calculator which is being used in modern error control codes like LDPC, Polar and Turbo codes for error detection and correction during their decoding process with various algorithms. These codes find applications in Wi-Fi, DVBs, Compact Disks and etc. Low power design for minimum distance calculator is proposed by using 3 transistors and compared with existing standard design.

Keywords: CMOS transistors, Data transmission,* E*rror correction, Minimum distance calculator.

I. INTRODUCTION

Modern error correcting codes consist of LDPC, Polar, Turbo code which are used at various level of Data processing, Data storage and data communication for the purpose of error correction and detection. Since its inception, coding theory has drawn from a rich and interacting variety of mathematical areas, including detection theory, information theory, linear algebra, finite geometries, combinatory, optimization, system theory, probability, algebraic geometry, graph theory, statistical designs, Boolean functions, number theory, and modern algebra. It is important, therefore, to motivate the mathematics carefully and implement that logic on power efficient hardware which will be cost effective and also optimized in terms of area and efficiency.

Error control coding in the context of digital communication has a history dating back to the middle of the twentieth century. In recent years, the field has been revolutionized by codes which are capable of approaching the theoretical limits of performance, the channel capacity. This has been impelled by a trend away from purely combinatory and discrete approaches to coding theory toward codes which are more loosely tied to a physical channel and soft decoding techniques. The purpose of paper is to present error correction and detection coding in a modern setting, covering both traditional concepts thoroughly as well as modern developments in soft-decision and iteratively decoded codes and recent decoding algorithms for algebraic codes with low power design. Initially code construction for LDPC, polar, and turbo is discussed. Secondly significant of error detector along with minimum distance calculator is proposed and results are compared.

II. ERROR CONTROL CODES

2.1 LDPC code

An LDPC code is a linear block code defined by a very sparse parity check matrix, which is populated primarily with zeros and sparsely with ones. The LDPC code also showed improved performance when extended to non-binary code as well as binary code to define code words. The LDPC code yields a signal to noise ratio approaching a Shannon channel capacity limit, which is the theoretical maximum amount of digital data that can be transmitted in a given bandwidth in presence of certain noise interference.

2.1.1 LDPC Codes: Construction and Notation

To denote the length of the code we use *N* and *K* to denote its dimension and information bits *M = N – K.* Low density parity check codes are linear codes defined by a parity check matrix. We will consider binary codes, where all operations are carried out in the binary field. Since the parity check matrices we consider are generally not in systemic form, the symbol A is use to represent parity check matrices, reserving the symbol H for parity check matrices in systematic form. Following the general convention in the literature for LDPC codes, assume that vectors are column vectors. A message vector **m** is a *K*1* vector; a codeword is a *N*1* vector. The generator matrix *G* is *N*K* and parity check matrix *A* is *(N-K)*N*, such that *H.G* = 0. The row of a parity check matrix as

$$A=\begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_M^T \end{bmatrix}$$

The equation $a_i^T c$ = 0 is said to be a linear parity-check constraint on the codeword **c**. The notation $z_m = a_m^T c$ where z_m is parity check or, a check. For a code specified by a parity check matrix *A*, it is necessary for encoding purposes to determine the corresponding generator matrix *G*. A systematic generator matrix may be found as follows.

$$H = A_p^{-1} A = [I \quad A_2]$$

Having found *H*, form

$$G = \begin{bmatrix} A_2 \\ I \end{bmatrix}$$

Then *HG* = 0, so A_p *HG* = *AG* = 0, so *G* is a generator matrix for *A*. while *A* may be sparse, neither the systematic generator *G* nor *H* is necessarily sparse. A matrix is said to be sparse if fewer than half of the elements are nonzero. Parity check matrix should be such that no two columns have more than one row in which elements in both columns are nonzero.

2.1.2 Belief Propagation Algorithm

The graph *G* representing the parity check matrix *H* consists of two sets of vertices *V* and *C*. The Tanner graph for the parity check matrix is as shown in Fig.1. The set *V* consists of *n* vertices that represent the *n* codeword bits and are called variable nodes, denoted by $v_0, v_1, \ldots, v_{n-1}$. Variable node index correspond to the column number of the parity check matrix. An edge is contained in the graph *G* if and only if the variable node v_n is contained in a parity check sum c_j .

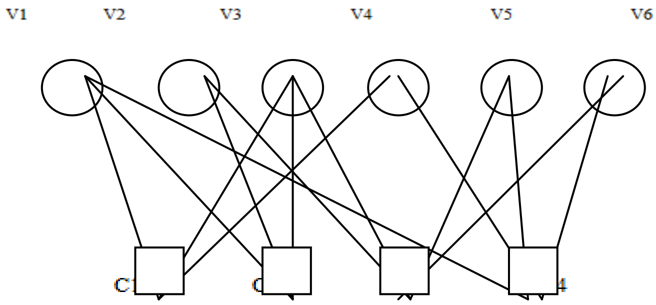


Fig. 1. Tanner graph for LDPC code [2]

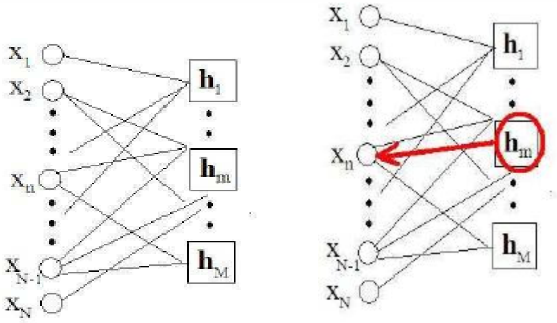


Fig.2. Massage passing through Belief propagation [2]

It can be seen from above diagram that in right bound iteration massages are send from variable nodes to check nodes and in left bound iteration massages are sent from check node to variable nodes.

BP decoding is an iterative process in which neighboring variables “talk” to each other, passing messages such as:“I (variable *x*) think that you (check h) belong in these states with various likelihoods”. After enough iteration, this series of conversations is likely to converge to a consensus that determines the marginal probabilities of all the variables. Estimated marginal probabilities are called beliefs. So BP algorithm is the process to update messages until convergence, and then calculate beliefs. Using only the sign bit of LLR (λ), one can estimate the most probable value of C_n.

$$\lambda\ (X_n|\mathrm{y})=\log\frac{p(Xn=1|y)}{\mathrm{p(Xn=0|y)}}=\log\frac{p(Xn=1|yn,\{Yi:i\neq n\})}{\mathrm{p(Xn=0|Yn,\{yi:i\neq n\})}}$$

2.2 Polar and turbo codes

Polar codes, invented by Arıkan, are the first “practical” codes that are known to achieve the capacity for a large class of channels. Their code construction is based on a phenomenon called “channel polarization”. This suggests a simple scheme where we fix the inputs to the channels that are bad and transmit reliably over the clean channels without any coding. The rate of such a scheme approaches the capacity of the channel. The channel polarization phenomenon suggests using the noiseless channels for transmitting information while fixing the symbols transmitted through the noisy ones to a value known both to sender as well as receiver. For symmetric channels we can assume without loss of generality that the fixed positions are set to 0. The encoding as well as the decoding operation of polar codes can be implemented with O(NlogN) complexity, where N is the block length of the code[5]. Channel Polarization synthesizes N channel B-DMC then it split into noiseless channel approaching the capacity of *I* (*W*) or into a pure noise channel approaching 1- *I* (*W*)[5] .So we can use this sort of polarization to construct Polar codes by sending data through those channel with high capacity and fix the inputs through those channels with low capacity. Now when the information bits are encoded in the channel, in encoding structure of polar codes, there is the XOR operation is performed at channel combining stage of channel polarization. This construction yields Polar codes whose lengths are powers of two. For example, the generator matrix of an **n=8** for Polar Codes Shown in figure 3. Polar codes can also be illustrated using a graph representation. In this case, the information vector u is presented on the left-hand side of the graph and the resulting decoded codeword x is obtained on the right-hand side. The $\oplus$ symbols represent XOR operations [5].

$\bar{x}$= [u_0 u_1 u_2 u_3 u_4 u_5 u_6 u_7]*F_8

$$F^{\otimes 1} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = F$$

$$F^{\otimes 2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$F^{\otimes 3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$x_i = \mathbf{u}\mathbf{G}_i$$

Fig.3.Generator matrix of n=8 [5]

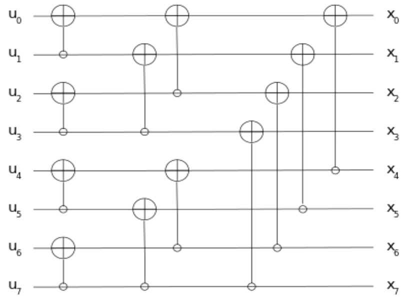


Fig.4. Graph representation of polar code[5]

It is also referred that Polar codes are based on the observation that specific bits in the input data going to be better protected from noise than others. As described in [5] that as the code length N grow larger, individual bits in the input word tend to become either very well or very poorly protected. Polar codes are constructed by identifying those well-protected bit indices in the information vector and using them to transmit information. They are called *information set* while the remaining positions form the *frozen set*, which is usually set to a predetermined value known by both the encoder and the decoder. Now when bit are encoded in the channel they are XORed with each other .At channel combining stage of channel polarization which is shown in fig.5 which shows the first recursion manner of n=1 for channel W_n where N=2^n,N>=0.

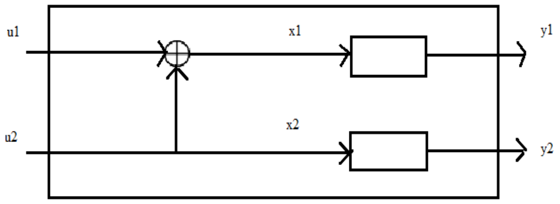


Fig.5. Combining of channels W_2[5]

2.2.1 SC Decoding Algorithm

The above procedure can be seen as transmitting a codeword and decoding at the receiver with a successive cancellation (SC) decoding strategy. Decoding of polar code can be done using Successive Cancellation decoding scheme. Frozen set are kept fix, Information set is passed during decoding. For each i=1,2,…,N. If u_i is frozen, set $\hat{u}$ =0, otherwise generate decision,

$$\hat{u}_i = \begin{cases} 0, & \text{if } L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) \geq 0, \\ 1, & \text{otherwise,} \end{cases}$$

where

$$L_N^{(i)}(y_1^N, \hat{u}_i^{i-1}) = \log \frac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|0)}{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|1)}.$$

By using above formula, channel which tends rate of one is to polarize in one noise free and the other channel which tends to zero is noisy channel.

Turbo codes were introduced in 1993 at the International conference on communication (ICC) by Berrou, Glavieux and Thitimajshima in their paper "Near shennon's limit error correction coding and decoding-Turbo codes Turbo codes are actually a quasi-mix between Block and Convolutional codes [7]. Convolutional codes are commonly specified by three parameters; (n, k,m) n = number of output bits, k = number of input bits, m = number of memory registers. A block code is any member of the large and important family of error-correcting codes that encode data in blocks. The term block code may also refer to any error correcting code that acts on a block of k bits input data to produce n bits of output data (n, k). The rate of the code is k/n. Each message word is associated with one and only one codeword. The total number of codewords in a code equals that of message words, 2k. It follows from the property of subspace that linear block codes have the following two important properties:

1). The sum of any two codewords in C is another codeword in C: $C_i + C_j = C_k$ Where C_i, C_j and C_k €C.

2). There exists a set of k codewords in C which are linearly independent such that every codeword in C is a linear combination of the k codewords:

$C = m_0g_0 + m_1g_1 + \ldots\ldots + m_{k-1}g_{k-1}$

Where g_0, g_1, g_2,……g_{k-1} are the k linearly independent code words, and m0 ,m1,……,mk-1 are some scalars. Linearly independent it means that $m_0g_0+m_1g_1+\ldots\ldots+m_{k-1}g_{k-1} \neq 0$ unless m_0 ,m_1,……,mk-1 = 0 . Figure: 6 show turbo encoder structure. Those discrete symbols are not suitable for transmission over physical channel. So, that the modulator transforms each output symbols of channel encoder to wave form that is enters in the channel and corrupted by noise. The demodulator processes each wave form and may be produced discrete output. Channel encoder transforms this received sequence to estimated sequence. And finally source encoder transforms estimated sequence to estimated output.

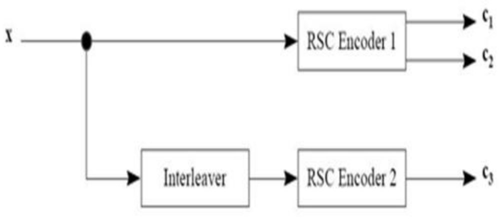


Fig.6. Recursive Systematic Convolution encoder[7].

2.2.2 Channel

Channel as AWGN (additive white Gaussian noise) 'Additive' because it is added to any noise that might be intrinsic to the information system. 'White' refers to idea that it has uniform power across the frequency band for the information system. It is an analogy to the color white which has uniform radiations at all occurrences in the observable spectrum. 'Gaussian' because it has a usual distribution in the time domain with an average time domain value of zero. The channel capacity for the AWGN channel is given by:C=1/2 log(1+P/N) Where, P represents the maximum channel power & N represents Noise [2].

2.2.3 Turbo Decoder

There are different decoding algorithms for turbo code such as Viterbi decoding, MAP (Maximum a posterior), SOVA (soft output Viterbi algorithm), etc. From the literature survey, Maximum A Posteriori algorithm, sometimes also called as BCJR(bahl, cocke, jelinek and raviv) algorithm, offerings an optimal decoding method for linear codes which minimizes the symbol error possibility. This is different from usually used Viterbi algorithm, In essence the Viterbi algorithm minimizes the probability of sequence (or word) error, which does not translate to minimizing the probability of individual bit (symbol) errors[7].

III. SIGNIFICANCE OF XOR GATE

It can be conclude from above all modern error control code's construction that all of them require XOR gate at various stage of Encoder and Decoder. Further more for any particular code error correction and detection capacity is given by hamming distance between massage bits and decoded bits which can be implemented by simple XOR gate.Minimum distance is given by $d$min = n - k + 1 and the code is capable of correcting any combination of t or fewer errors, where t can be expressed as $t = \frac{d_{min}-1}{2}$. So, as the number of bits increased the requirements for number of XOR gates will be linearly increases which inspire to generate less transistor XOR gate design.

3.1 Design of XOR Gate

A new design of XOR Gate is proposed in which only 3 transistors are used as compared to conventional 8 transistors. This decrease in the number of transistors will lead to decrease in the chip size and hence the cost but won't affect the overall performance.

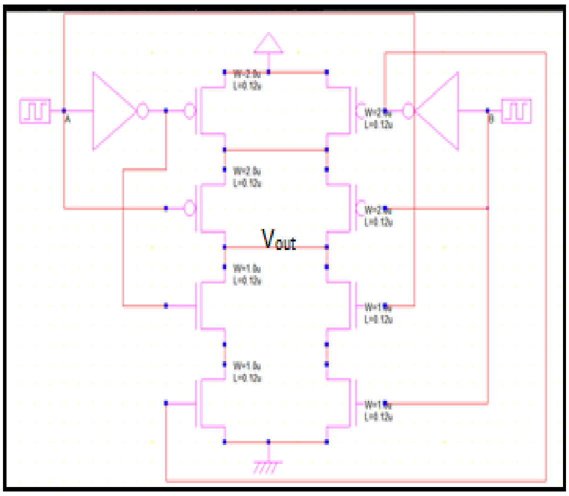


Fig.7 Conventional XOR gate design[1]

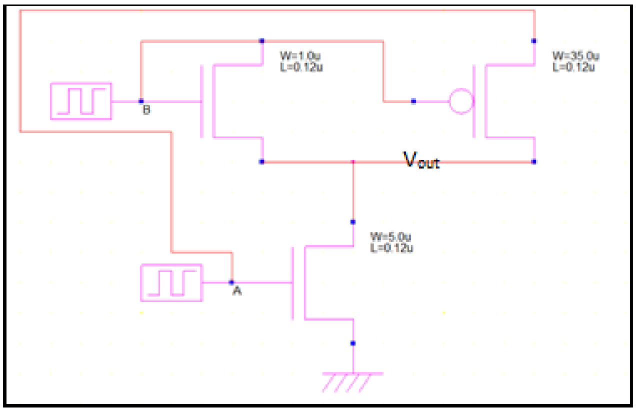


Fig.8 Proposed XOR gate design

In Proposed XOR gate, when both the inputs are at logic '0', pMOS transistor will turn on and hence V_{out} will be logic '0'. When input A is at logic '0' and B is at logic '1', upper nMOS transistor will be on ,Vout will be at logic '1'. When A is at logic '1' and B is at logic '0', lower nMOS is on and pMOS is on but still Vout will be at logic '1'. When both inputs are at logic '1', Vout will be at logic '0'.

3.1.1 Simulated Waveforms of XOR Gate

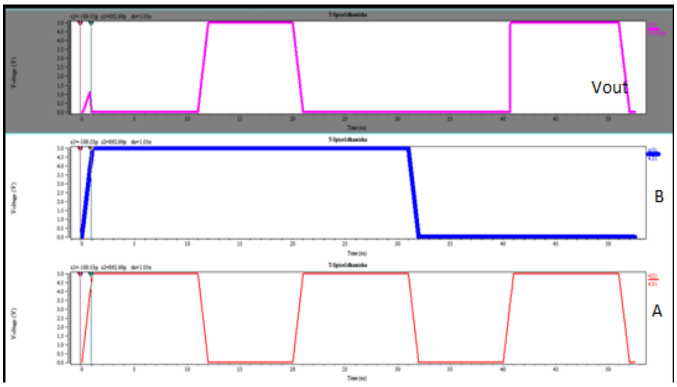


Fig. 9. Waveforms of Conventional XOR gate

\

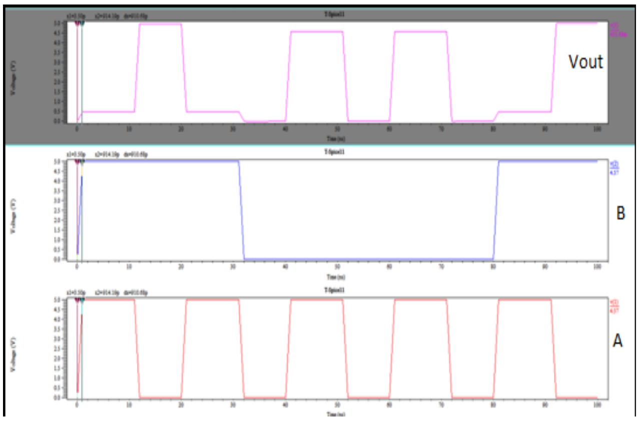
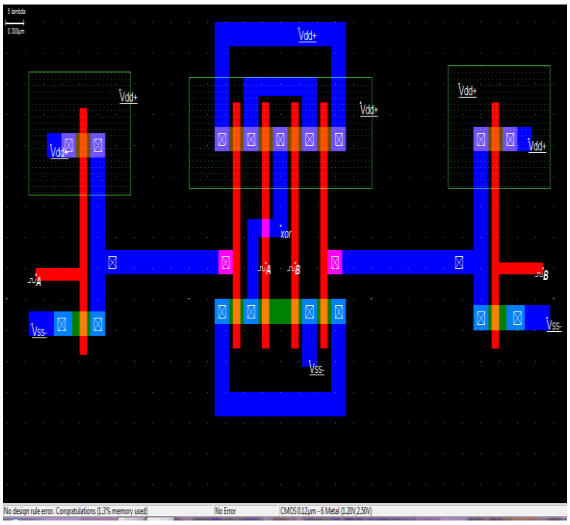


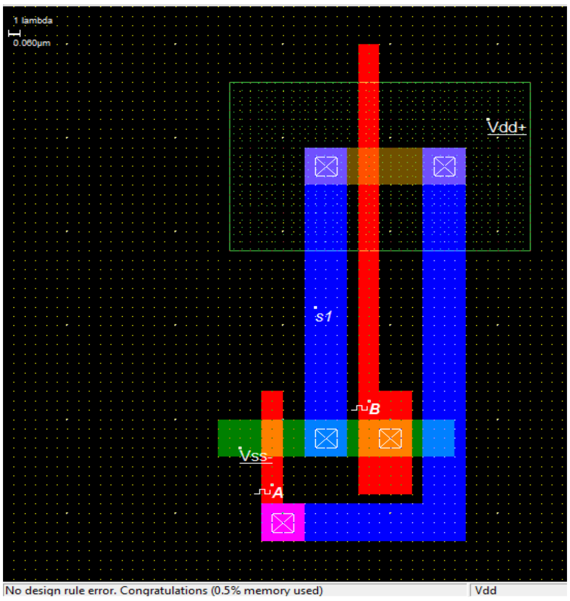
Fig. 10. Waveforms of Proposed XOR gate

3.1.2. .Layout of XOR Gate



No design rule error. Congratulations (1.3% memory used) | No Error | CMOS 0.12µm - 6 Metal (1.20V,2.50V)

Fig.11. Layout of conventional XOR gate



No design rule error. Congratulations (0.5% memory used) | Vdd

Fig.12. Layout of Proposed XOR gate

VI. Result and Conclusion

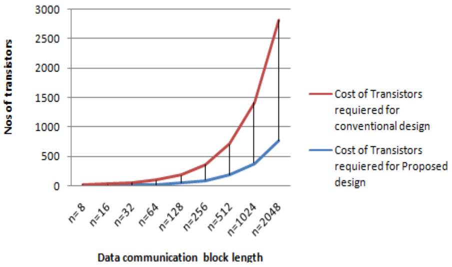


Fig.13. Cost Comparison

TABLE1. Results of Various Design Parameters

Parameters	Conventional Design	Proposed Design
Nos. of Transistors	8	3
Memory Used	1.3%	0.5%
Propagation Delay	1.05ns	910ps

Layout, Simulated waveforms and results shows that as compared to conventional design our proposed design has less numbers of transistors and accordingly cost, memory and delay can be optimized which can be used at different stages of development of modern error control code.

REFERENCES

[1] Manisha and A P archana kumara ,"A novel design of ultra low voltage energy efficient full adder", The IUP Journal of Telecommunications, Aug 2014.

[2] R. G. Gallager, "Low density parity check codes," *IEEE Trans. Inform. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.

[3] Saeedi, and Hamid," Performance of Belief Propagation for Decoding LDPC Codes in the Presence of Channel Estimation Error ". *Communication IEEE transaction* on Dept. of Syst. & Comput. Eng., Carleton Univ., Ottawa, Ont. vol. IT-55, pp. 83–87, Jan. 2007

[4] Susmitha Remmanapudi, Balaji Bandaru, Department of Mechanical Engineering, Indian Institute of Technology Madras, Chennai – 600036, "*AN FPGA IMPLEMENTATION OF LOW DENSITY PARITY-CHECK CODES CONSTRUCTION & DECODING*" in Devices, Circuits and Systems (ICDCS), 2012 International Conference on ,m ISBN 978-1-4577-1545-7, March 2012.

[5] E. Arikan, "Channel polarization: A method for constructing capacityachievingcodes for symmetric binary-input memoryless channels," *IEEETrans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.

[6] A New Design of XOR-XNOR gates for low power application;Nabihah Ahmad, RezaulHasan/International Conference on Electronic Devices, Systems & Applications (ICEDSA)2011

[7] Nabeel Arshad, Abdul Basit," Implementation and Analysis of Turbo Codes Using MATLAB", Journal of Expert Systems (JES) 2013