

Implementation of Power Efficient Decoding Algorithm for LDPC Code

Namrata H. Patel

Computer engineering department,
Hasmukh Goswami College of Engineering, Vahelal

Risha A Tiwari

Professor, Computer engineering department,
Hasmukh Goswami College of Engineering, Vahelal

ABSTRACT

Mobile computing devices like mobile, laptops & tablets that covered large market of end user facing battery issues is an open challenge for research scholars. In this project work I have attempt to optimize power [batteries relate issues] at decoding. State of most efficient error control LDPC code which is being used. In this work bit flipping decoder hard decision decoding is used which can be implemented by using 8 transistor by using XOR gate. This paper proposed 6 transistor XOR gate design and implemented which can be used in this decoder.

Keywords

Decoding, Encoding, Low density parity check (LDPC), Bit flipping, tanner graph, XOR-XNOR.

1. INTRODUCTION

LDPC code was first presented to the world by Robert G. Gallager at MIT in 1960. There are numerous types of channel codes in coding theory known as block codes and convolution codes [16]. Early in the 90's people were designing different kind of codes like convolution codes, Reed Solomon codes, BCH code etc who detects and correct single bit errors or randomly multiple bit errors and they turned out that the performance in terms of channel capacity. The main problem is to satisfy the industry terms like power consumption and as well as to achieve the best capacity of channel which is possible with LDPC Codes. High Speed Communication Systems require very high speed, less hardware complexity, Low power consumption, high coding gain and good error performance decoders to decode the error-correction codes [1]. In 1960s where this code proved the extraordinary performance with iterative decoding. LDPC codes which are really advanced to achieve spectrum efficiency in recent applications like whatsapp, line, viber as well as in DVB-S2/ DVB-T2/DVB-C2. These codes are also included in many future standards as well as to replace many existing channel coding techniques. Low-Density Parity-Check (LDPC) codes have several advantages over turbo codes, including reduced implementation complexity, better bit error rate (BER) performance at low signal to noise ratio (SNR) and the inherent code structure that supports high degree of parallelism. LDPC codes have become increasingly popular [4]. LDPC have been adopted in latest generation high data rate applications such as WLAN, WiMax and Digital Video Broadcasting - Second Generation (DVB-S2) [4]. LDPC decoding algorithms can be classified into soft-decision decoding algorithm such as the Sum-Product Algorithm (SPA) and hard-decision decoding algorithm such as Bit-Flipping (BF) algorithm. A decoder which has all these properties is a challenging work in communication field.

Another challenging work is to use 6 number of MOS Transistors with this to achieve the power consumption.

1.1 LDPC Code: Construction

The construction of LDPC codes are special linear block code with parity check matrix H , Which assigning a small number of the values in an all-zero matrix to be 1 so that the rows and columns have the required degree distribution. LDPC codes belong to a class of block codes. As name suggests, its parity-check matrix (H) consists of very small number of non-zero elements. A Low-Density Parity-Check (LDPC) codes matrix is described by various parameters.

The matrix H is said to be regular if the degree distribution of rows and columns are uniform, otherwise it is Irregular. Tanner in 1981 introduced an effective graphical representation for LDPC tanner code. Parity check matrix H can be represented as a graph called Tanner graph. A cycle in the graph is a sequence of connected nodes, which start and end at the same node. A regular parity-check matrix with 10-bit code length is shown in Fig. 1 (a) and the corresponding Tanner graph representation of the matrix is shown in Fig. 1(b)

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

Fig: 1 (a) LDPC matrix

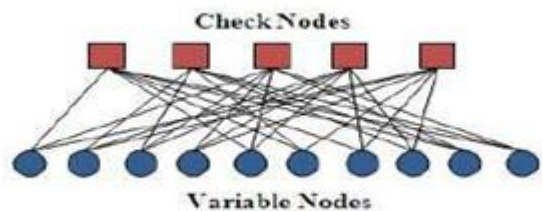


Fig: 1(b) Tanner graph representation of LDPC

The decoding of LDPC code involves passing of messages between the check nodes and variable node in the Tanner graph. This decoding algorithm is often called as message passing algorithm [2]. Each of the nodes in the Tanner graph works in isolation with information available along the connected edges only.

These decoding algorithms require passing of the messages between the nodes for a fixed number of times. Hence such algorithms are also known as iterative algorithms.

2. THEORETICAL BACKGROUND

2.1 Encoding of LDPC code

Encoding of codes, especially of who has higher block lengths can be quite difficult to implement in hardware but there are several methods of generating H such that encoding can be done via shift registers. If the generator matrix G of a linear block code is known then encoding can be done using Parity check matrix [9]. The cost of the method depends on the Hamming weights i.e. the no of 1's of the basis vectors of G . If the vectors are dense, then cost of encoding using this method is proportional to n^2 [9]. If G is sparse then this cost becomes linear with n . However, the blank space of matrix H is given by LDPC code. It is unlikely that the generator matrix G will also be sparse. Therefore the straightforward method of encoding LDPC would require number of operations proportional to n^2 . It is very slow for most practical applications. Therefore it is desirable to have encoding that run in linear time.

2.2 Decoding of LDPC codes

There are different iterative decoding algorithms having two derivations. They are hard decision decoding and soft decision decoding respectively. There are various algorithms to deal with decoding of LDPC codes. They are classified according to their complexities to get decoded.

Hard Decision Decoding

In this hard decoding scheme the check nodes finds the bit error by checking the parity which may be even or odd and the messages from variable nodes are transmitted to check nodes, check node checks the parity of the data stream received *from variable nodes connected to it. If number of 1's received at check nodes satisfies the required parity, then it sends the same data back to message node, else it adjusts each bit in the received data stream to satisfy the required parity and then transmits the new message back to message nodes. So here in case each node f_j looks at the message received from the variable nodes and calculates the bit that the fourth variable node should have n order to fulfil the parity check equation [5].

2.2.1 Message passing algorithm (MPA)

Message passing on the BEC can be as per the following. Here the bit is received correctly or it will be completely erased and be null. The messages passed by the Tanner graph edges are also binary: a bit node declaring a message 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 and if the modulo-2 sum of the incoming bit values is zero. If the messages received by a bit node are different from its received value the bit node changes (flips) its current value [7]. 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.

Message-passing decoding of the received string $y = [0\ 0\ 1\ x\ x\ x]$. Here each subfigure indicates the decision to make at each step of the decoding algorithm which is based on the messages from the previous step. For the messages, a dotted arrow corresponds to the messages represent bit = 0 while a

solid arrow corresponds to bit Represent bit = 1, and a light dashed arrow correspondsto "bit x".

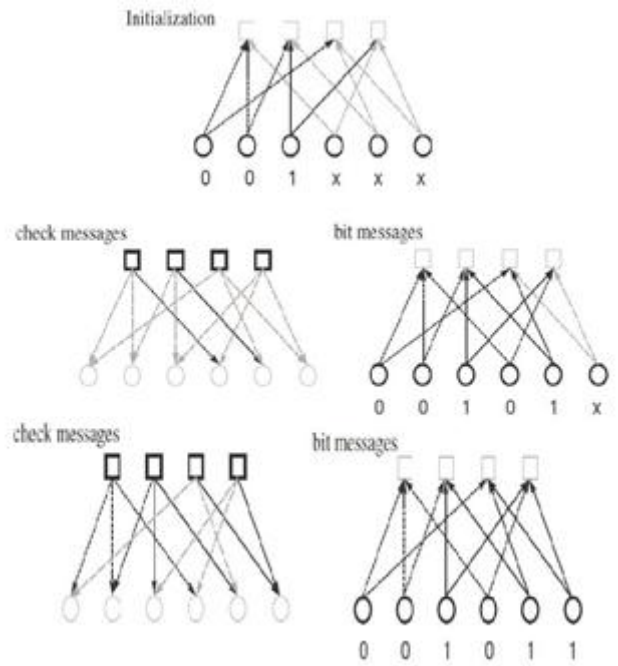


Fig 2: Messages Passing

3. BIT FLIPPING ALGORITHM

The bit-flipping algorithm is a hard-decision message-passing algorithm, which is used in LDPC codes. A binary (hard) decision for each received bit is made by the detector and passed to the decoder. For the bit-flipping algorithm the messages passed along with the Tanner graph edges are also binary: a bit node declaring a message 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 and if the modulo-2 sum of the incoming bit values is zero. If the messages received by a bit node are different from its received value the bit node changes (flips) its current value [7]. 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.

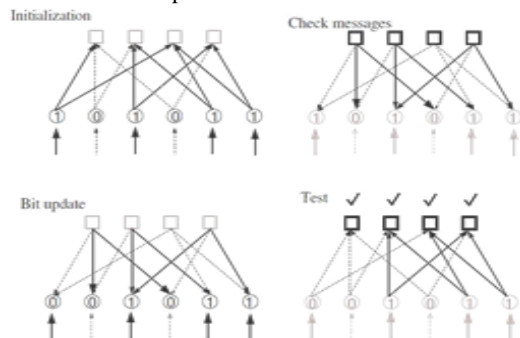


Figure 3.Bit flipping Representation [7]

Bit-flipping decoding of the received word $y = [1\ 0\ 1\ 0\ 1\ 1]$. The sub-figure indicates the decision made at each step of the decoding algorithm. Decoding algorithm are based on the messages from the previous step. A cross(\times) represents that the parity check is not satisfied while a tick (\times). For the messages, a dashed arrow represents to the messages "bit message bit = 0" while a solid arrow represents to "bit = 1".

Bit Flipping Algorithm

A. Statement

Bit Flipping Algorithm decoder computes all the parity checks and changes any digit that is contained in more than some fixed number of unsatisfied parity checks equations. Using these new values, the parity checks are recomputed and this process is repeated until all the parity checks are satisfied or maximum iteration reached.

B. Steps of the Algorithm

Gallager proposed the following BF decoding algorithm:

1. Compute each parity check for the received word r .
2. For each received bit, count the number of failed parity checks.
3. Flip the bit(s) with the largest number of failed check(s).
4. Repeat steps 1, 2, and 3 until all parity checks are satisfied or until the predetermined number of iterations is reached.

The Tanner graph of the code is drawn in next to it. Suppose that the codeword is an all-zero vector $c = 001011$, and the received word contains two errors in the second and fourth bits such that $r = 101011$.

Numerous designs were reported to realize the XOR XNOR functions using different number of circuit techniques and approaches. They vary in methodologies and transistor count to improve the circuit performance in term of speed and density. Among these, the Conventional design of XOR-XNOR circuit using static CMOS network can be found in Each input is connected to both an NMOS transistor and a PMOS transistor. It provide a full output voltage swing but with a large number of transistors. It uses eight transistors and complementary inputs and has a drawback of loss of driving capability. Wanget al. also designed XOR-XNOR circuits based on inverter gates. It does not require a complementary inputs but it has no driving capability because there is no direct connection to Vdd and Gnd. The improved version of this circuit has been designed by adding a standard inverter to the output. This modified circuit provides a good driving capability but uses twelve transistors for XOR-XNOR circuits. Shiv et al. proposed two of XOR-XNOR circuits (Figure 4) and less power dissipation and faster compared to design in with a low supply voltage. However both of the circuits give a poor signal output voltage in certain input combination. In the XOR and XNOR circuit based on Pass Transistor Logic (PTL) using 6 transistors is reported as shown in Figure 5. It has a full output voltage swing and better driving capability by using Vdd and Gnd connection. In the XOR-XNOR circuit adds a forward and backward feedback between the XOR and XNOR gate and additional transistors to rectify the degraded logic level problem.

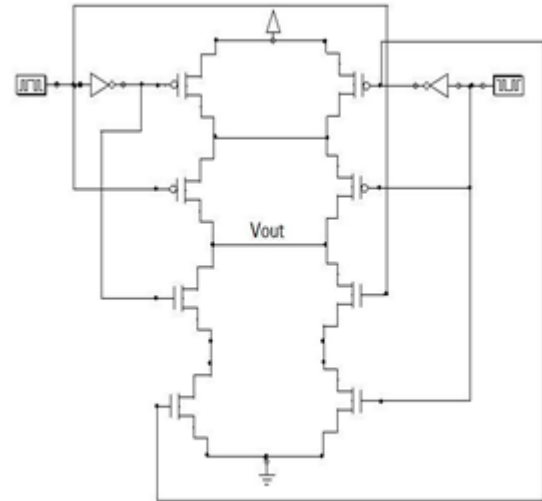


Fig: 4 XOR –XNOR gate using 8 transistors in [3]

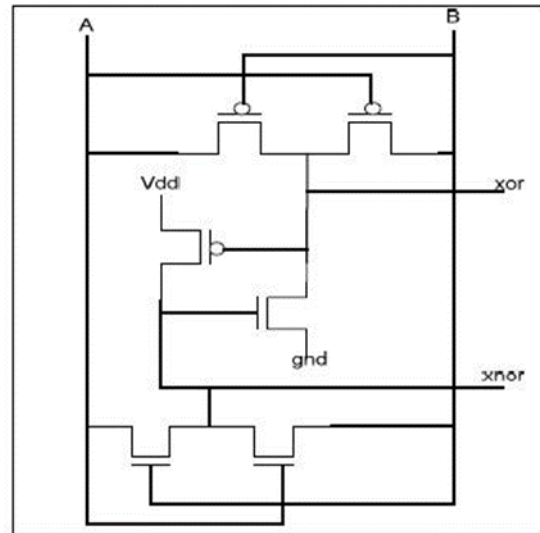


Fig: 5 XOR-XNOR gate using 6 transistors in [3]

3.1 Proposed Xor-Xnor Gate Circuit:

The XOR and XNOR gate functions are shown in Table 4.1 and denoted by \oplus and respectively. The logic expression for XOR and XNOR are
 $A \oplus B = A'B + AB'$
 $A \odot B = A'B' + AB$

Table 1. XOR AND XNOR Gate function

A	B	XOR	XNOR
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

4. SIMULATION RESULTS

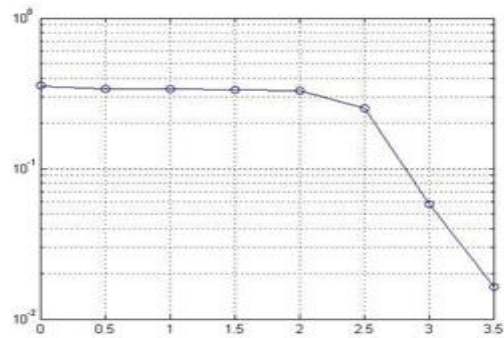


Fig: 6 Simulation Result of XOR-XNOR

The proposed adder cell use XOR-XNOR circuits shown in figures and provides the advantage of good driving capability, delay using minimum supply voltage and minimum number of transistors.

4.1 Microwind Design Xor Gate Using 8 Transistor

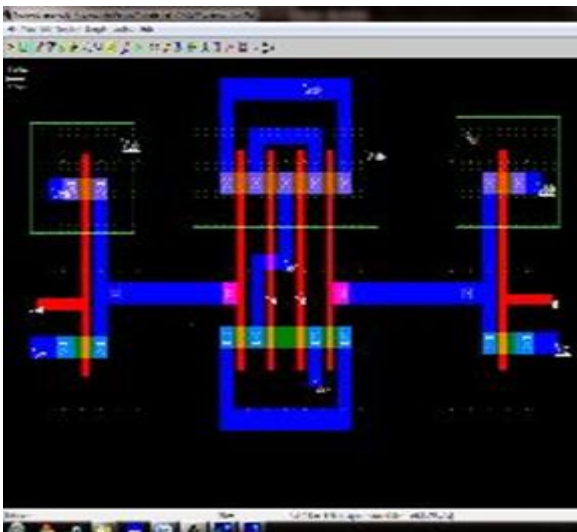


Fig 7: Proposed Xor Gate Using 8 Transistor

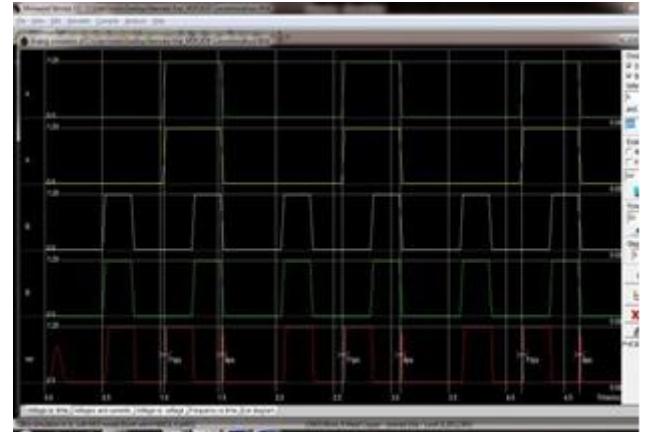


Fig 8: Waveforms of conventional XOR gate

Figure 7 Screenshot of the result of the XOR gate with 8 transistor generated design which are later on converted to waveform. They vary in methodologies and transistor count to improve the circuit performance in term of speed, density and power consumption. Among these, the conventional design of XOR-XNOR circuit using static CMOS network can be found in [4]. Each input is connected to both an NMOS transistor and a PMOS transistor. It provide a full output voltage swing but with a large number of transistors.

4.2 Microwind Design Xor Gate Using 6 Transistor

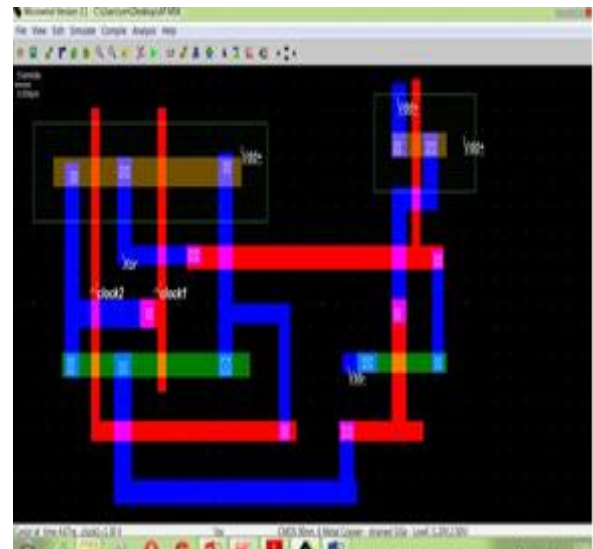


Fig 9: Proposed Xor Gate Using 6 Transistor

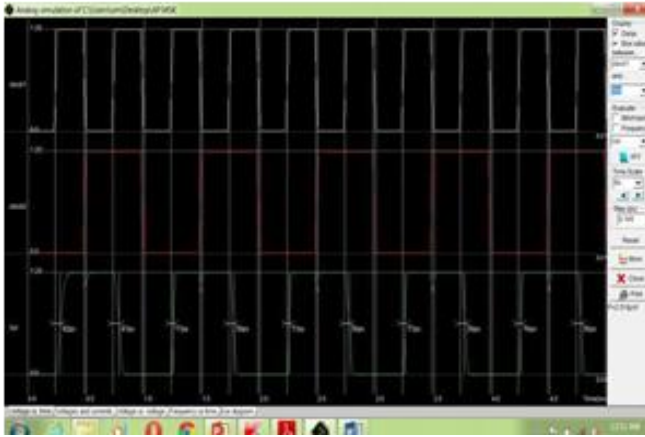


Fig 10: Waveforms of conventional XOR gate

Figure 9 Screenshot of the result of the XOR gate with 8 transistor generated design which are later on converted to waveform. They vary in methodologies and transistor count to improve the circuit performance in term of speed, density and power consumption which reduce the power compare to the 8 transistor.

5. RESULT OF 8 & 6 TRANSISTOR OUTPUT OF POWER SUPPLY

Table 1: Result of various design

Parameters	Conventional Design	Proposed Design
No.of transistor	8	6
Memory used	1.3%	0.5%
Power consumption	6.9 mw	2.5mw

6. CONCLUSION

According to Literature Hard decision Decoder for LDPC is implemented and simulated on MATLAB and the Hardware design for particular is proposed. The new design of combination XOR-XNOR circuit configuration. According to the Simulation results, the proposed circuit offers a better, more competitive than other design and low supply voltage.

The XOR design of transistors is selected to optimize power in proposed decoder. Hard-decision based solutions such as those based on the Bit Flip Algorithm (BFA).The proposed algorithm also reduces the average number of decoding iterations compared to BFA more numerical on Decoder. We compare the result of 8 transistor circuit and 6 transistor circuit. In 8 transistor power consumption is 6.9 mw and in 6 transistor power consumption is 2.5 mw. The proposed of full adder cell used 8 transistor Micro wind implementation of XOR with 6 transistors will be done which can be sent to Fab-Lab to actual chip level manufacturing.

7. ACKNOWLEDGMENTS

I would like to thanks ass prof. Risha A Tiwari also for precious suggestions and motivation throughout and to thanks to anonymous reviewers for their valuable suggestion.

8. REFERENCES

- [1] R. G. Gallager, 'Low-Density Parity-Check Codes'.Cambridge,m.i.t. Press, 1962.
- [2] A novel design of ultra low voltage energy efficient full adder,2014.
- [3] A New Design of XOR-XNOR gates for low power application; Nabihah Ahmad, International Conference on Electronic Devices, (ICEDSA)2011
- [4] FPGA Implementation of a LDPC Decoder Using a Reduced Complexity Message Passing Algorithm journal of networks,vol 6,no. 1,january 2011.
- [5] Implementation for Two-Stage Hybrid Decoding for LDPC Codes;International Journal of Computer Applications Volume 80, October 2013.
- [6] Efficient Encoding of Low-Density Parity-Check code /Thomas j.Richardson L.,Urbanke;IEEE,Vol47, No.2,FEBRUARY 2001.
- [7] LDPC Codes – a brief Tutorial,Bernhard M.J.Leiner, Stud.ID.:53418Lbleiner@gmail.com,April 8, 2005.
- [8] Performance of QPSK-OFDM with LDPC & Concatenated Reed Solomon/Convolutional Coding in Outdoor Environments, ISSC, 2011.
- [9] Introducing Low-Density Parity-Check, Codes,Sarah J. Johnson,School of Electrical Engineering and Computer Science,The University of Newcastle, Australia.
- [10] Systematic construction, verification and implementation methodology for LDPC codes.