Impact Factor (SJIF): 3.632

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

Volume 3, Issue 4, April-2016

DESIGN AND IMPLEMENTATION OF ERROR CONTROL CODE USING LOW POWER VLSI TECHNIQUE

Ms. Pallavi Timmappa Shetty[1], Mrs. Jayashree C. N[2],

[1] *P.G. Student, Department of ECE, SDM College of Engineering and Technology , Dharwad*
[2]*Professor, Department of ECE, SDM College of Engineering and Technology , Dharwad*

Abstract – In this paper for Error control code Low Density Parity Check (LDPC) Codes are chosen as it offer remarkable error correction performance and therefore increase the design space for communication systems. This performance made them suitable for many modern applications such as Digital Satellite Broadcasting system (DVB-S2), Wireless Local Area Network (IEEE 802. 1 1n) and Metropolitan Area Network (802.16e). The decoding process of these codes is based on an iterative algorithm that requires many computational cycles. This paper mainly presents the modified normal sum product algorithm which gives accurate result compared to other existing systems. Clock gating technique is mainly used in the proposed paper to reduce the dynamic power consumption. The proposed LDPC Decoder architecture is simulated on Xilinx 14.5 ISE Simulator and implemented on Spartan 6 using Verilog code.

Keywords--- Low Density Parity Check (LDPC) codes, iterative decoding, Sum Product algorithm, Low Power, FEC(Forward Error Correction), Check Node Unit(CNU), Variable Node Unit(VNU).

I. INTRODUCTION

Low density parity check codes are first introduced in the 1962 PhD thesis of Gallager [1] and after 35 years, they have been rediscovered by MacKay and Neal [2]. Their remarkable error correcting performance has made them a basis of many modern standards, such as satellite transmission standard DVB-S2, WIMAX standard IEEE 802.16e, and WIFI standard IEEE 802. 11n[3], [4],[5] LDPC codes can be represented by a sparse parity check matrix or using a bipartite graph called Tanner graph [2]. A Tanner graph has two sets of nodes: the first set of nodes is called "variable nodes" and they represent the N bits of a codeword, the second set of nodes is called "check nodes' and they represent the parity check equations. An edge is drawn between ith variable node and jth check node if the ith code bit is included in the jth parity check equation. LDPC are most commonly decoded using iterative algorithm that is message passing algorithm also known as Belief Propagation or Sum Product Algorithm where the exchange of messages back and forth takes place. Many implementation methods have been proposed to design highly efficient and flexible decoders . Fully parallel architecture where every variable and check nodes are routed leading to complexity in hardware implementations and computations. Serial Decoders are too slow . So we go for partially parallel decoders where multiple variable node and check node processors and memory are tradeoff between performance and cost. Here we present efficient implementations of the SPA and describe new reduced-complexity derivatives thereof. In our approach, log-likelihood ratios (LLR) are used as messages between symbol and parity-check nodes. It is known that in practical systems, using LLRs offers implementation advantages over using probabilities or likelihood ratios. In the proposed architecture it is modified normal sum product algorithm . It is a combination of unicasting at check nodes as it performs complex multiplications and multicasting at variable nodes as it performs simple addition . It won't check for each iteration. The iterations are fixed . After fixed say 10 iterations it check for valid code word in order to reduce computational cycles.

II. LDPC DECODING ALGORITHM

LDPC codes are block codes with parity-check matrices that contain only a very small number of non-zero entries. The main difference between LDPC and ordinary block codes is how they are decoded. LDPC codes are decoded iteratively using a graphical representation of their parity-check matrix, called a Tanner graph. There are two main types of LDPC Decoding namely Hard Decision Decoding and Soft Decision Decoding. In this hard decoding scheme the check nodes finds the bit in 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. And then if number of 1's received at check nodes satisfies the required parity, then it sends the same data back to message nodes, 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.

The bit flipping algorithm is an example of hard decision decoding . The Bit flipping decoder is going to be immediately terminated, whenever a valid code word has been found. By checking if all the parity check equations are satisfied then process get terminated. Soft-decision decoding gives enhanced performance in decoding procedure of LDPC and QC-LDPC codes which is based on the idea of belief propagate. In soft scheme, the messages are the conditional probability that in the given received vector received. bit is a 0 or a 1. The sum product algorithm is a soft decision decoding message passing algorithm. Priori probabilities for the received bits is the input probabilities as here they were known in advance before running the LDPC and QC-LDPC decoder. The bit probabilities returned by the decoder are called the posterior probabilities. A LDPC code parity-check matrix is called (w_c, w_r)-regular if each code bit is contained in a fixed number , w_c of parity check equation and each parity-check equation contains a fixed number, WI' , of code bits. Equation (I) shows a regular parity-check matrix with w_c 3 and w_V 4.

$$H = \begin{pmatrix}
1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1\\
1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0\\
0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0\\
0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0\\
0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1\\
1&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0\\
0&1&0&0&0&0&1&0&0&0&1&0&0&0&0&1&0&0&0&0\\
0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&1&0\\
0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&1&0&0&0\\
0&0&0&1&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1
\end{pmatrix} \quad (1)$$

In general, LDPC decoding algorithms are called Message Passing Algorithm. Algorithms are named based on the type of information being passed through edges that is represented through Tanner Graphs. The name is called Sum Product Algorithm because the operations performed in Check Nodes are complex multiplications and the operations performed in Variable Node processors are Simple additions. The Check node computations are done parallely followed by Variable Node computations . The information are passed through Interconnect block. In the proposed architecture instead of RAM for variable node and check node we are using registers for each and every node.

The actions performed are similar but localized. Here we represent the probabilities as Log Likelihood Ration(LLR). In that case Belief Propagation is called Sum Product Algorithm. Following the power efficiency , we have adopted a partially-parallel architecture where a separate VNU or CNU is designated for each variable node or check node in the code Tanner graph. Another advantage of partially-parallel decoder architecture is that unlike most -parallel decoders that are based on a particular code construction (such as the (3, k)-regular construction in or the Architecture- Aware code construction in [13]), the partially-parallel architecture can be applied to irregular codes with no constraint on the code structure. This is done simply by instantiating VNUs and CNUs of the desired degree and connecting them based on the

code graph. The only consideration is that the timing performance of the decoder for irregular codes will be typically limited by a critical path through the nodes with highest degree.

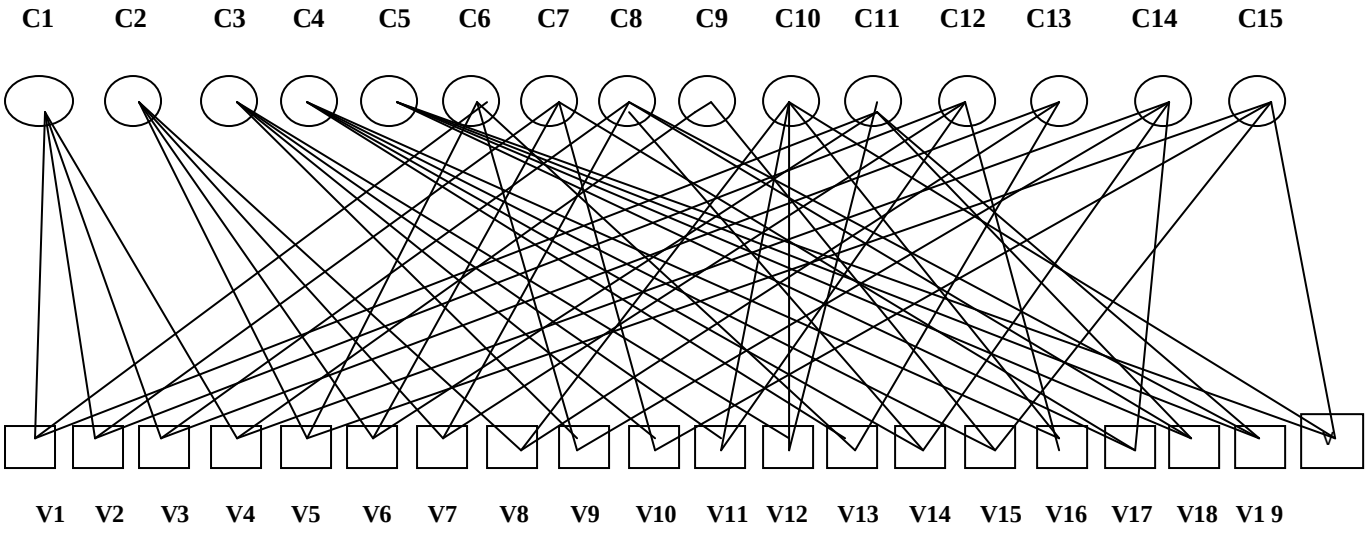

Figure 1. Tanner Graph Representation of the Parity Check Matrix

In sum product algorithm, the message from check node j to variable node i, $E_{i,j}$ is the LLR of the probability that bit i causes parity check equation j to be satisfied.

$$E_{j,i} = 2\tanh^{-1}\left[\pi_{\propto\in V[j],\propto\neq i}\right]\tanh\frac{M_{j,\propto}}{2} \quad (2)$$

Here V[j] represents the set variable nodes that are connected to check node j it is clearly represented in the above parity check matrix. $M_{j,\propto}$ represents their LLR values. Each variable node will be having initially LLR values and receives LLR values from each connected check node. The total LLR of each bit is the sum of these LLR and is given by:

$$L_{i=}r_i+\sum_{\propto\epsilon[j]}(E_{\alpha,j}) \quad (3)$$

Here C[j] represents the set of Check nodes connected to Variable node i. The messages are sent from variable nodes to check nodes they are in the form of LLR values and does not contain any $E_{j,i}$. So the messages sent from each variable node to all check node is given by the following equation

$$M_{j,i} = r_i + \sum_{\propto\epsilon C[j]}(E_{\alpha,j})$$

III. PROPOSED ARCHITECTURE

The implementation mainly depends on the combination of unicasting and multicasting strategies. To give high throughput and increased bit error rate . Check node computations are given by equation (2). As it requires complex multiplication it performs unicasting and cannot afford extra processing cycles. Variable node computations are multicasting as they perform simple additions. This implementation has greatly reduced hardware complexity and due to partially parallel architecture and using basic gates for variable and check node processor implementation leads to less power consumption thus leads to low power decoder. The proposed architecture supports unicasting at check nodes and multicasting at the variable nodes. It mainly composed of Registers, variable node processor, Check node processor, buffers and subtractor.

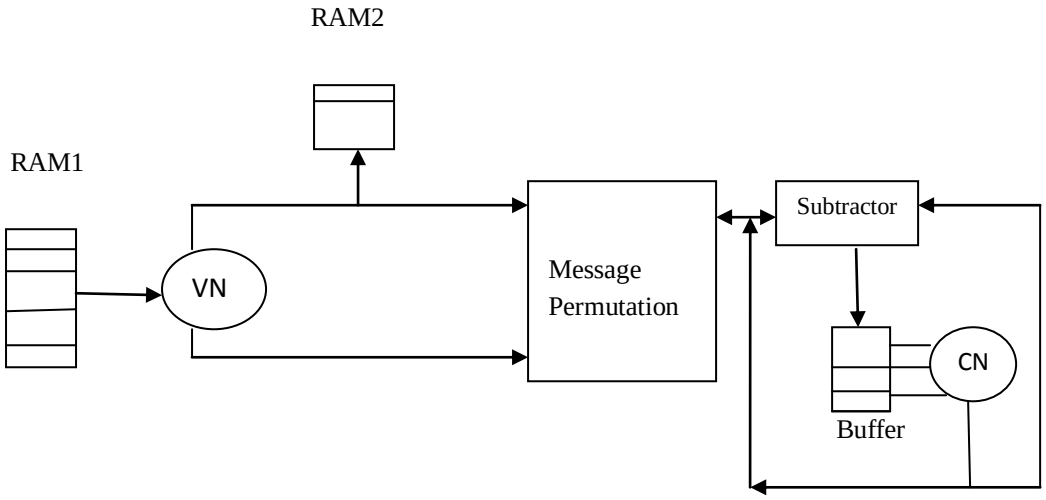

Figure 2. Architecture of proposed LDPC Decoder

Every variable node processor is associated with an arrangement of RAM units. One unit holds the underlying LLRs to be prepared by this processor (RAMI). Different units hold the estimations of messages imparted over the associated edges to this processor; RAM2 units for associated edges. Then again, every check node processor additionally has an arrangement of buffer units. One unit holds the past emphasis estimations of the edges associated with this processor . Another unit acts as a cradle that holds the contrast between the multicasted message from the VN processor and the past emphasis computed esteem. The proposed decoder does not check for a substantial codeword, on the grounds that an ideal opportunity to perform a check would be the length of the check node half iteration. Thus it checks for fixed iteration.

A. Variable Node Processor

The variable node processor is composed of an adder with number of inputs equivalent to the quantity of edges associated with the VN processor in addition to one info for the underlying LLR estimation of the right now handling bit (3 edges for this situation).

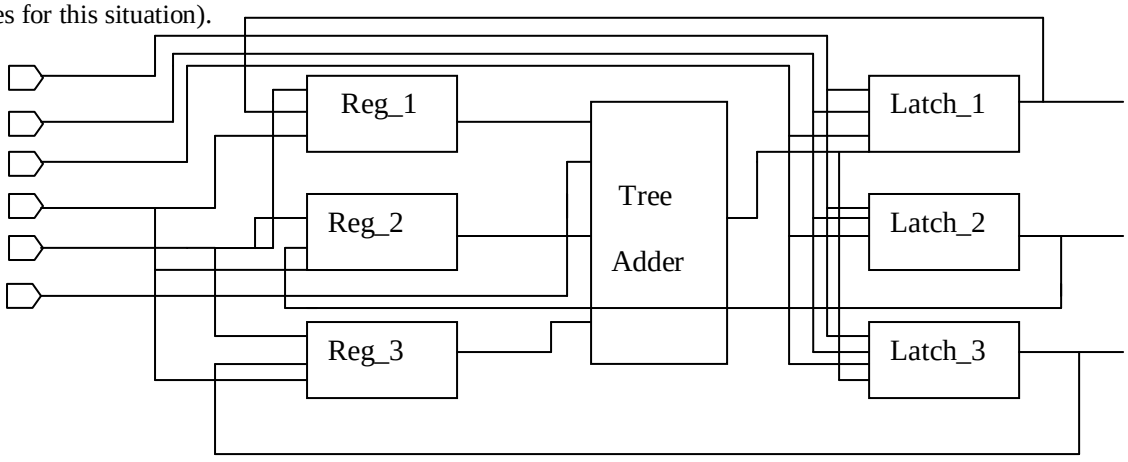

Figure 3. Block diagram of variable node processor

Numerous adder executions are accessible, however it was found that the carry look-ahead adder is the best bargain amongst execution and unpredictability , so a tree adder was worked with carry look-ahead adders. The processor loads three qualities from three associated RAM units through bidirectional ports. In the meantime, the underlying LLR quality is stacked from the associated codeword RAM unit. The result of addition is latched with the goal that it can be composed back to the edges RAM units through the bidirectional ports.

B. Check Node Processor

Check node processor is supposed to calculate the complex multiplications involving the hyperbolic functions. the heart of the check node processor is composed of an adder with a lookup table on each input and another lookup table on the output. The sign of the output LUT is corrected according to the signs of the inputs. By XOR'ing the MSB of the inputs and if the result is '1', then the result of the output LUT is negated.
The estimations of the edges to be prepared by the CN processor are stacked from the RAM units through the stage system to the minuend input of the subtractor. In the meantime, the past cycle input are stacked from CN's nearby RAM to the subtrahend input of the subtractor. The result of the subtractor is put away in the CN's buffer. The CN processor takes a shot at the right edges' qualities put away in its support and figures . The outcome is latched with the goal that it can be composed to the nearby RAM unit, and to the edges RAM units through the permutation network

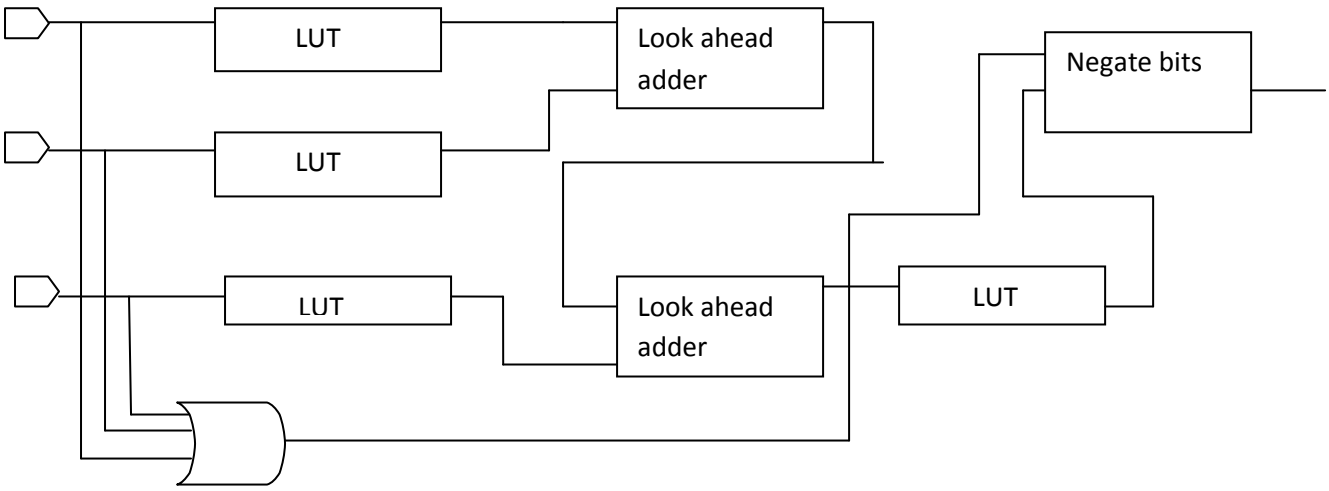

Figure 4. Block diagram of Check node processor

C. Message Permutation Network

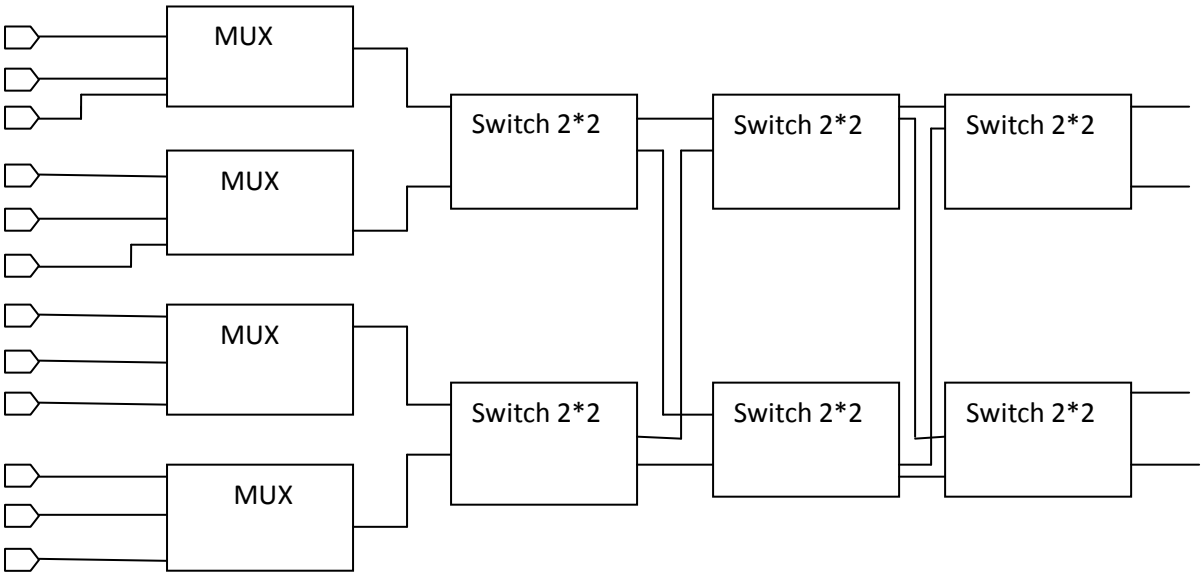

Figure 5. Block diagram of Permutation Network

To exchange values from edges RAM units to and from CN processors, a permutation network is required. Numerous sorts of change systems are accessible Benes system is observed to be a decent trade off amongst versatility and adaptability. It is a non blocking system and it is utilized to interface each cN processor to any RAM unit at VN hubs.

The Benes organize customarily comprises of (2Log2N-1) number of stages where every stage comprises of (N/2) 2x2 rudimentary switches. For this plan, each VN processor has numerous RAM units for edges values, and the structure is proposed for standard codes. In this way, multiplexers are utilized between the VN RAM units and the stage system.

D. Clock Gating Technique

Clock gating is one of the most frequently used techniques in RTL to reduce dynamic power consumption without affecting the functionality of the design. One method involves inserting gating conditions in the RTL, which the synthesis tool translates to clock gating cells in the clock-path of a register bank. This helps to reduce the switching activity on the clock network, thereby reducing dynamic power consumption in the design. Since the translation done by the synthesis tool is purely combinational, it is referred to as combinational clock gating. This transformation does not alter the behavior of the register being gated.

IV. RESULTS

In this paper, a Low-Density Parity Check decoder is implemented using VERILOG coding technique. The decoder designed here can detect and correct the error and the decoded data signal of 20 bits length. For simulation, synthesis and timing analysis Xilinx 14.5 is used. The experiment results are divided into following parts which are simulation results as waveform, RTL schematic, utilization of resources , area in terms of LUTs & delay. Steps for simulation in any coding is development of the algorithm to be coded..Once the algorithm is ready, coding is started and different modules are developed. These modules are independently checked for errors and later they are assembled to form the exact code. The whole code is also checked for errors and logic is checked by simulating the code.

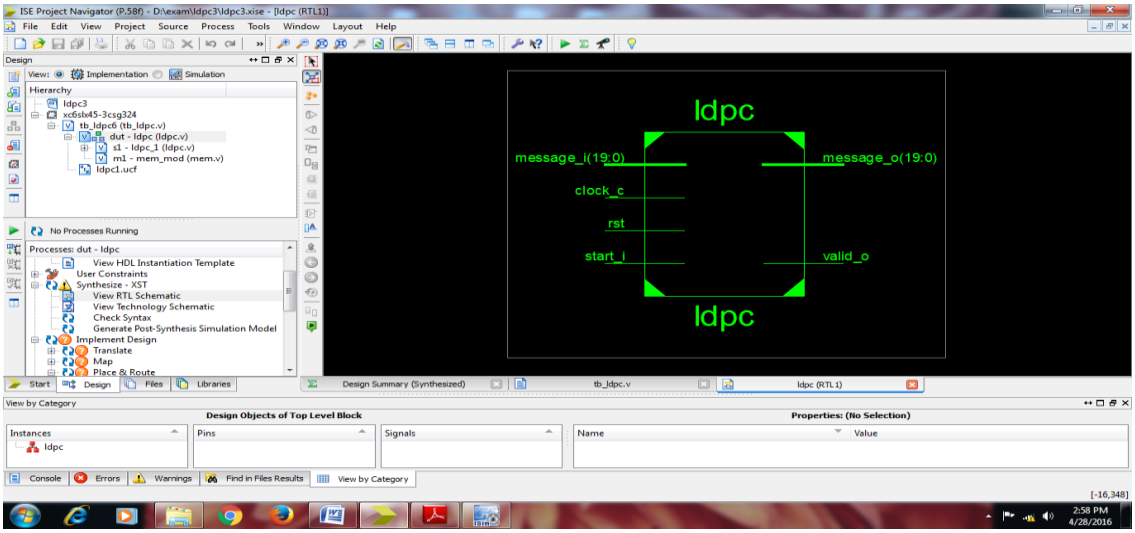

Figure 6. Symbol of LDPC Decoder

A. Simulation Results

Simulation is done on Xilinx ISE Simulator, where 20 bit data is decoded with the help verilog code, By entering any value in input array we can obtain decoded output that can correct upto single bit error.

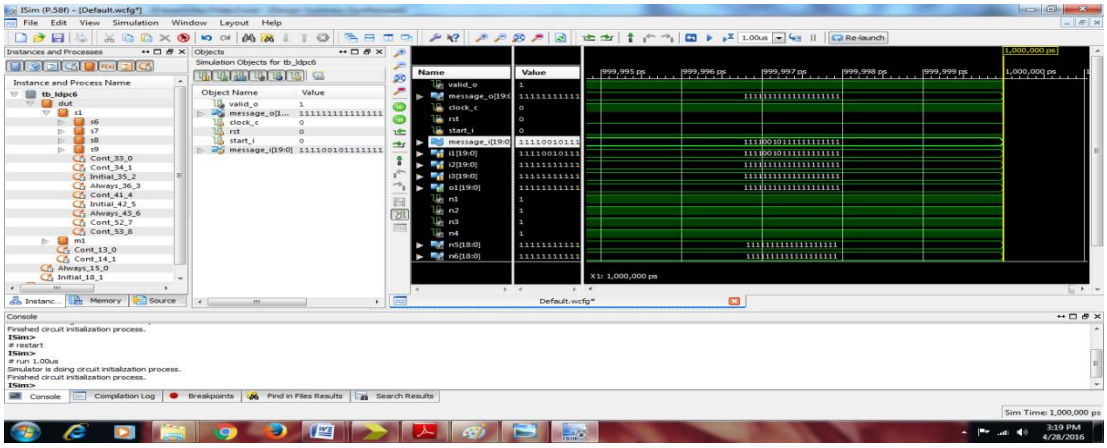

Figure 7. Waveform of LDPC Decoder

B. RTL Schematic

After the phase of the synthesis process, we can display a schematic representation of our synthesized source file. This schematic shows a representation of the pre-optimized design in terms of generic symbols, such as adders, multipliers, counters, AND gates, and OR gates, that are independent of the targeted Xilinx device.

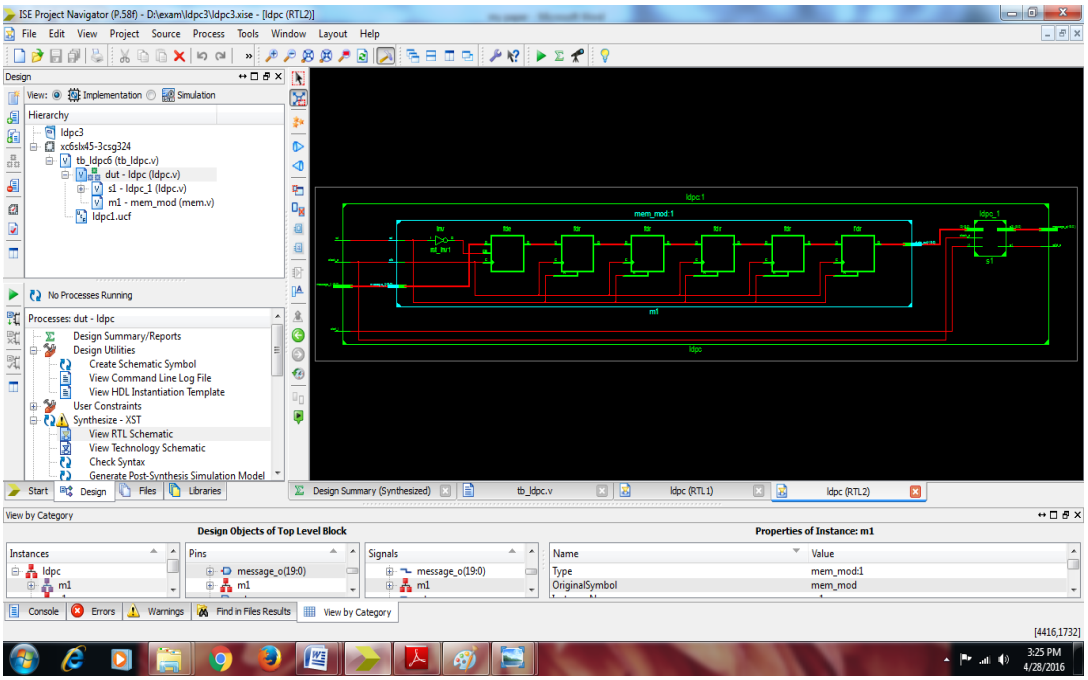

Figure 8. RTL Schematic of LDPC Decoder

C. Utilization of Resources

In any design the utilization of resources is most important, it defines the utility of device. For the above LDPC design utilization is as follows:

Table 1. Utilization of resources

Logic Utilization	Used	Available
Number of Slice Registers	102	54578

Number of Slice LUT's	145	27286
Number of fully used LUT- FF pairs	62	184

D. Power Summary

The below table gives the information regarding Total power , Quiescent Power, and Dynamic power for the clock frequency of 20MHz.

Table 2. Power Summary

	Total	Dynamic	Quiescent
Supply Power(W)	0.040	0.003	0.036

V. CONCLUSION

The decoder for LDPC codes is implemented using the bipartite graph of the parity check matrix. The partially parallel architecture is a good trade-off between throughput and hardware cost. The fully parallel LDPC decoding architecture can achieve high decoding throughout, but it suffers from large hardware complexity caused by a large set of processing units and complex interconnections. This paper introduces a modified implementation of the usual sum-product decoding algorithm with the goal of minimizing the communication overhead between the processing elements of the decoder, and at the same time, keeping the processing tasks as simple as possible. The simulation and implementation results showed that the suggested algorithm was able to reduce the number of messages transmitted between the processing elements. It also showed that the decoder was able to deliver high throughput with linear scalability to support even higher bitrates with more processing elements. Using Clock gating technique and partially parallel architecture leads to low power LDPC decoders.

ACKNOWLEDGEMENT

I am highly obliged to Department of Electronics and Communication Engineering, SDM College of Engineering and Technology.

REFERENCES

[1] R. Gallager, "Low-density parity-check codes, " Information Theory, IRE Transactions on, vol. 8, no. 1, pp. 21 -28, 1962-

[2] David J.C. MacKay (2003) Information theory, Inference and Learning Algorithms, CUP, ISBN 0-521-64298-1

[3] Ahmed M Sadekl, Aziza I. Hussein, "Flexible LDPC Decoder architecture for (3,6) Regular codes", IEEE , 2015.

[4] John C. Porcello, " Designing and Implementing Low Density Parity Check (LDPC) Decoders using FPGAs", IEEE, 2014.

[5]. Todd K. Moon (2005) Error Correction Coding, Mathematical Methods and Algorithms. Wiley, ISBN 0-471-64800-0

[6]. Larry Hardesty (January 21, 2010). "Explained: Gallager codes".*MIT News*. Retrieved August 7, 2013.

[7]. Ashima sood, shubham singh, nagendra sah,"comparative study of LDPC decoders and Viterbi decoders",vol2 Issue 3 IJSRD may 2014.

[8] .Hao Zhong and Tong Zhang, "Joint code-encoder decoder design for ldpc coding system Vlsi implementation" , ISCAS 2, page 389-392. (2004).

[9]. Li-Hsieh Lin and Kuei-Ann Wen, "A Novel Application of LDPC-Based Decoder for WiMAX Dual-mode Inner Encoder" , 2- 9600551-5-2© 2006 EuMA

[10]. Ashima sood, shubham singh, nagendra sah,"comparative study of LDPC decoders and Viterbi decoders", vol2 issue 3 IJSRD may 20.

[11] Sklar, B., Digital Communications Fundamentals and Applications, 2nd Edition, Prentice Hall, Upper Saddle River, NJ, 2001.

[12] Proakis, J. G., Salehi, M., Digital Communications, 5th Edition, McGraw Hill, Inc. New York, 2007.

[13] A. J. Blanksby and C. J. Howland, "A 690-mW I-Gb/s 1024-b, rate 'h low-density parity-check code decoder " IEEE J. Solid-State Circuits, vol. 37, no. 3, pp. 404-412, Mar. 2002

[14] V. A. Chandrasetty, S. M. Aziz, "An area efficient LDPC decoder using a reduced complexity min-sum algorithm Integration" , the VLSI Journal, Volume 45 Issue 2, pp. 141-148, March, 2012.

[15] D. Hayes, "FPGA implementation of a Flexible LDPC decoder", PSc thesis, University of Newcastle, Australia 2008.