International Journal of Engineering Science Invention
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org ||Volume 4 Issue 5 || May 2015 || PP.18-21

Belief Propagation Decoder for LDPC Codes Based on VLSI Implementation

Priyanka Kumawat, Himanshu Purohit
[1,2]*Department of ECE, Sir Padampat Singhania University, India)*

ABSTRACT: *Low density parity check (LDPC) codes are most widely used error correcting codes (ECC). Because of their popularity they are used in several applications such as the digital satellite broadcasting system (DVB-S2), Wireless Local Area Network (IEEE 802.11n) and Metropolitan Area Network (802.16e). These codes are used to transmit a message over noisy transmission channel. LDPC codes are constructed using sparse parity check matrices. LDPC codes provide very fast encoding and decoding algorithms. In this paper, low density parity check decoder is implemented using Verilog technique. A partially parallel decoder is design using belief propagation (BP) algorithm. For simulation, Modelsim is used and for synthesis, Mentor Graphics Leonardo Spectrum with vertex IV technology is used.*

KEYWORDS: *Low density parity check (LDPC) code; Error correcting code (ECC); Belief propagation (BP).*

I. INTRODUCTION

Low density parity check (LDPC) code is an error correcting code which is used in noisy communication channel to reduce probability of loss of information. LDPC codes are capacity-approaching codes, which means that practical constructions exist that allow the noise threshold to be set very close the theoretical maximum (the Shannon limit) for a symmetric memoryless channel. The noise threshold defines an upper bound for the channel noise, up to which the probability of lost information can be reduce.

Low density parity check (LDPC) codes are also known as Gallager codes because these codes proposed by R.G. Gallager in 1962[1]. With increased capacity of computers and the development of relevant theories such as belief propagation algorithm LDPC codes were rediscovered by Mackay and Neal in 1996[2].

LDPC codes are linear block codes which defined by sparse M × N parity check matric H. Where N is LDPC code length [3]. The tanner graph has been introduced to represent LDPC codes [4]. Tanner graphs are bipartite graph. Tanner graph have two types of nodes are variable node (v- node) and check node(c-node). The n coordinates of the codewords are associated with the n message nodes. The codewords are those vectors such that for all check nodes the sum of the neighbouring positions among the message nodes is zero.

$$H = \begin{array}{c|cccccccccc} & v1 & v2 & v3 & v4 & v5 & v6 & v7 & v8 & v9 & v10 \\ \hline C1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ C2 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ C3 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ C4 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ C5 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{array}$$

Figure 1. H- Matrix

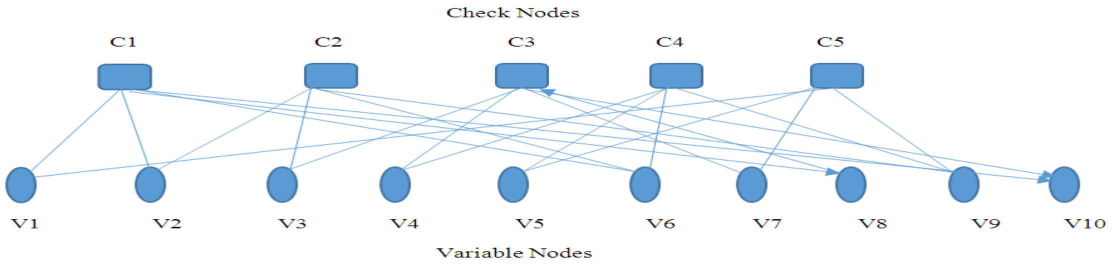

Figure 2. Tanner graph corresponding to the parity check matrix

Here we consider H be the parity check matrix of irregular (10, 5) LDPC code and its tanner graph is also shown in fig. 1. For this LDPC code the path (c1 → v8 → c3 → v10 → p1) with the black bold lines.
In recent year studies the decoding is done by various algorithms and different types of decoders are designed such as partially parallel decoder, memory efficient decoders. In all decoding scheme the belief propagation decoding is lead to good approximate decoder. A belief propagation decoding of LDPC codes on memoryless channel is best practical decoding algorithm.
In this paper the belief propagation decoding algorithm is define then modified sum product algorithm is defined and in next part of this paper decoder is implemented in Xilinx using Verilog.
The rest of the paper is arranged as follows: belief propagation algorithm is defined in section 2.1; Modified sum product algorithm is described in section 2.2; Experimental results are presented in section 3; Finally, the conclusion is given in section 4.

II. LDPC DECODER USING BELIEF PROPAGATION

Belief propagation, also known as sum-product algorithm is a message passing algorithm for performing inference on graphical models, such as Bayesian networks and Markov random fields. It calculates the marginal distribution for each unobserved node, conditional on any observed nodes. Belief propagation is used in artificial intelligence and information theory and has demonstrated empirical success in numerous applications including low-density parity-check codes, turbo codes.

(a).Belief Propagation Algorithm

Update Check Messages
For each check node j, and for every bit node associated with it j compute: we assume BPSK modulation, which maps a codeword c= (c1,c2,………., cN) into a transmitted sequence s= (s1,s2,…..sN). Then S is transmitted over a channel corrupted by additive white Gaussian noise (AWGN) [5].
Sum product algorithm consist the following steps. First LLRs used for priori message probabilities, then parity check matrix H and maximum number of allowed iterations Imax.
Steps for sum product algorithm
Initialise
Set Qij = λj, this initialise the check nodes with priori message probabilities.

$$Rij = 2 \tanh\text{-}1 \prod_{\alpha \varepsilon V\,(j,\alpha \neq i)} \tanh(\frac{Q\alpha j}{2}) \quad (1)$$

Test for valid codeword
Make a tentative decision on codeword

$$Li = \lambda j + \sum_{j\varepsilon c\,(j)} Q\alpha j \quad (2)$$

If number of iterations is Imax or valid codeword has been found then finish
Update Bit Messages
For each bit node j, and for every check node associated with it j compute:

$$Qij = = \lambda j + \sum_{j\varepsilon c\,(j)} R\alpha j[k-1] \quad (3)$$

(b). Modified sum product algorithm

Decoder which implement using modified sum product algorithm which is an approximated algorithm in context with normal SPA algorithm. Modified SPA is easy for implementation of decoder [6].
When encoder output is transmitted through the AWGN channel the output is given to the decoder’s variable node. Let M (n) denote the set of check nodes connected to the symbol node n and N (m) the set of symbol nodes participating in the m-th parity-check equation.

Step1
Initialization
Assuming the AWGN channel with noise variance $\sigma 2$, the reliability value is Lc = 2/σ2. The initialization is done in every position *(m, n)* of the parity check matrix *H*, where *Hm, n* =1 as

$$\lambda n \rightarrow m\ (u_n) = L\ (u_n) \quad (4)$$

$$\Lambda m \rightarrow n\ (u_n) = 0 \quad (5)$$

Step 2

Iterative Process
Update the check-node LLR, for each m and for each n∈ N (m), as

$$\Lambda_{m\to n}(u_n) = 2\, tanh{-1}\ \{\textstyle\prod_{n'\epsilon N(m)/n} \tanh\,[\frac{\lambda n' \to m(un')}{2}]\} \qquad (6)$$

Note that both the tanh and $\tanh^{-1}$. functions are increasing and have odd symmetry. Thus, the sign and the magnitude of the incoming messages can be used in a simplified version, as

$$\Lambda_{m\to n}(u_n) = 2\ \{\textstyle\prod_{n'\in N(m)/n} \text{sign}\ [\lambda n' \to m(un')]\}\ tanh{-1}\ \{\textstyle\prod_{n'\in N(m)/n} \tanh[\frac{\lambda n' \to m(un')}{2}]\} \qquad (7)$$

Step 3
Variable node update
Update the variable node LLR, for each n and for each *m* ∈ *M (n)*, as

$$\lambda_{n\to m}(u_n) = L(u_n) + \textstyle\sum_{m'\epsilon M(n)/m} \Lambda m' \to n(un) \qquad (8)$$

Step 4
Decision Process
Decide if $\lambda_n\ (u_n) \geq 0$, then u_n=0 and if $\lambda_n\ (u_n) \leq 0$ then u_n=1. Then compute the syndrome uH^T=0, then the codeword (*u*) is the final codeword, otherwise the iteration takes place till valid code word is obtained.

III. EXPERIMENTS AND RESULTS

(a).*Simulation result*
Simulation is done to check the correctness of the code. In this paper Verilog coding technique is used. Verilog code is designed to check errors like syntax errors, logical errors, etc. simulation is done by using modelsim. Decoding is done using Verilog in Xilinx and simulation results are given below.

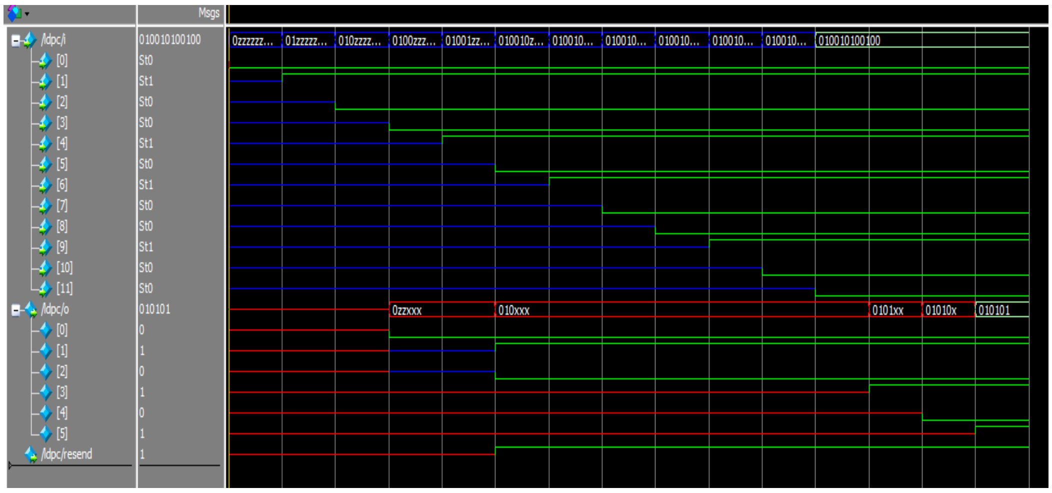

Figure 3. Simulation result

(b).*Synthesis*
Synthesis is done using the software mentor graphics Leonardo spectrum. And the technology used is Virtex IV. In synthesis the result is observed like chip area, delay, etc. chip area is find out by the number of look up tables (LUTs). The delay gives propagation delay.
The table show the results obtained during the synthesis of the code to implement the LDPC decoder.

TABLE 1. RESULT FOR DECODER IN VERTEX IV BOARD

Pass	Area (LUTs)	Delay (in ns)	DFF	PIs	POs
1	15	1	0	12	7

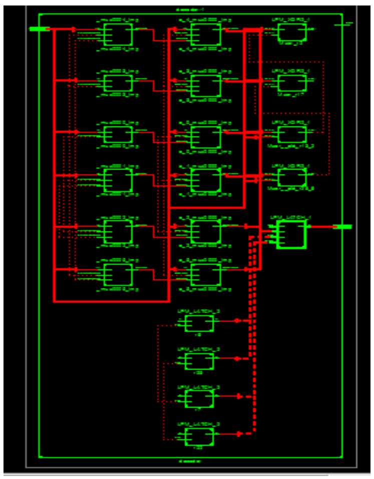
Figure 4. Decoder RTL Schematic

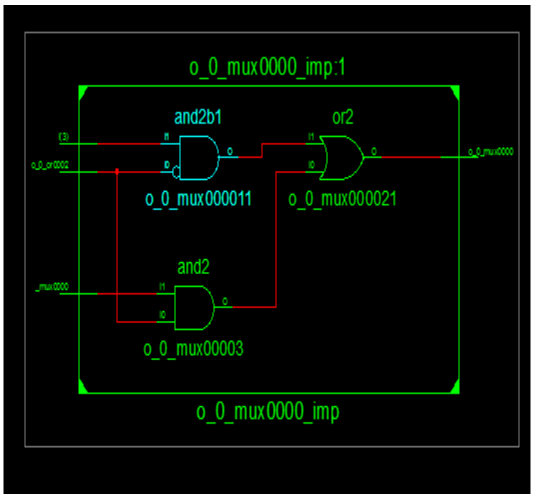

Figure 5. Enlarged part of Decoder RTL Schematic

IV. CONCLUSIONS

The decoder for the LDPC codes is implemented with use of bipartite graph. Code is implemented by using Verilog and in Xilinx and simulation is done by using modelsim. The decoder modified sum product algorithm was found to be effective for decoding. We observed that high throughput LDPC decoding architectures should exploit the benefit of parallel decoding algorithms.

V. ACKNOWLEDGMENT

This research paper is supported by Department of ECE at Sir Padampat Singhania University, Udaipur.

REFERENCES

Journal Papers

[1]. Robert G. Gallager, "*Low Density Parity Check Codes", IRE Trans. Inf. Theory, Vol. IT-8, No.1, pp. 21–28*, Jan 1962.
[2]. MacKay, D.J.C., "*Good error-correcting codes based on very sparse matrices", IEEE Trans. Inform. Theory, Vol. 45, No. 3, pp. 399–431*, March 1999.
[3]. Lei Yang, Hui Liu, and C.-J. Richard Shi," *Code Construction and FPGA Implementation of a Low-Error-Floor Multi-Rate Low-Density Parity-Check Code Decoder", Department of Electrical Engineering University of Washington, Seattle, WA 98195*
[4]. D. J. C. Mackay, S. T. Wilson and M. C. Davey, "*Comparison of construction of irregular Gallager codes", IEEE Transactions on Communications, Vol. 47, pp. 1449-1454*, Oct. 1999.
[5]. Jinghu Chen, and Marc P. C. Fossorier, Senior Member, "*Near Optimum Universal Belief Propagation Based Decoding of Low-Density Parity Check Codes", IEEE Transactions on Communications, Vol. 50, No. 3*, March 2002.
[6]. S. Papaharalabos, P. Sweeney, B.G. Evans, P.T. Mathiopoulos, G. Albertazzi, A. Vanelli-Coralli and G.E. Corazza, "*Modified sum-product algorithms for decoding low-density parity-check codes", IET Communication., 2007, Vol. 1, No. 3, pp. 294–300*, 2007
[7]. Guido Masera, Federico Quaglio, and Fabrizio Vacca, "*Implementation of a Flexible LDPC Decoder", IEEE Transactions on circuits and systems, Vol. 54, No. 6,* June 2007.
[8]. T. Richardson, "*Error floors of LDPC codes", in Proc. Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, pp. 1426-1435*, 2003
[9]. Tuan Ta, *"a Tutorial on Low Density Parity-Check Codes", The University of Texas at Austin.*
[10]. Edward Liao1, Engling Yeo2, Borivoje Nikolic, "Low-Density Parity-Check Code Constructions for Hardware Implementation, IEEE Communications Society" 2004.
[11]. Jin Sha, Minglun Gao, Zhongjin Zhang,Li Li, Zhongfeng Wang, "*High-Throughput and Memory Efficient LDPC Decoder Architecture", Proceedings of the 5th WSEAS Int. Conf. on Instrumentation, Measurement, Circuits and Systems, Hangzhou, China, pp. 224-229*, April 2006

Books:

[12]. Samir Palnitkar, *Verilog HDL- A Guide to digital design and synthesis, IEEE* 1364-2001.
[13]. *Verilog golden reference guide*, Doulos.