International Journal of Electronics and Communication Engineering.
ISSN 0974-2166 Volume 5, Number 2 (2012), pp. 133-141
© International Research Publication House
http://www.irphouse.com

A Novel Decoding Approach for Non-binary LDPC Codes in Finite Fields

[1]A. Anand and [2]Dr. P. Senthilkumar

[1]Research Scholar/Assistant Professor/ECE
Saveetha Engineering College, Chennai, India
E-mail: anandtce@gmail.com
[2]Professor/ECE, Saveetha Engineering College, Chennai, India
E-mail: muneeskumar05@yahoo.in

Abstract

In this paper we propose an integrated version of EMS (Extended Min-Sum) and layered turbo decoding in Low density Parity Check Codes. Algorithmic complexity and memory problems are the major problem faced in NB-LDPC. This can be reduced by EMS algorithm under logarithm domain in the order of (n_{m} log$_2$ n_{m})**.**Speed is increased by using layered turbo scheduled decoding algorithm. Efficient implementation of non-binary LDPC decoders is a progressing field which is updated currently. This paper is based on (1) the hardware implementation costs for NB-LDPC decoders with Galois field(8,16,128,256) on FPGA and (2) To set the noise threshold very close to the theoretical maximum (Shannon Limit).

Keywords: NB-LDPC, EMS, layered turbo scheduled decoding algorithm, BP

Introduction

Coding is the conversion of information to another form for some purpose. Source Coding: The purpose is lowering the redundancy in the information. (e.g. ZIP, JPEG, MPEG2)Channel Coding: The purpose is to defeat channel noise.

A low-density parity-check (LDPC) code is a linear error correcting code. It is a method of transmitting a message over a noisy transmission channel, and is constructed using a sparse bipartite graph.

LDPC codes are capacity-approaching codes which means that practical constructions exist that allow the noise threshold to be set very close to the theoretical maximum (the Shannon limit) for a symmetric memory-less channel. LDPC codes are finding increased use in applications where reliable and highly efficient information transfer over bandwidth or return-channel constrained links in the presence of data-corrupting noise is desired.

These Low-density parity-check (LDPC) codes have received tremendous attention in the coding community because of their excellent error correction capability and near-capacity performance. Some randomly constructed LDPC codes, measured in Bit Error Rate (BER), come very close to the Shannon limit for the AWGN channel (within 0.05 dB) with iterative decoding and very long block sizes (on the order of 10^6 to 10^7).

NB-LDPC codes can be decoded efficiently with message passing algorithms such as the belief propagation (BP) decoder, but the size of the messages varies in the order q of the Finite field. Here n_m << q. we propose to store only n_m reliabilities instead of q in each message. The truncation of messages from q to n_m values has to be done in an efficient way in order to reduce its impact on the performance of the code. It has been shown how GF (q) LDPC codes can outperform precisely engineered binary codes of dimension up to $log_2 q$ times bigger.

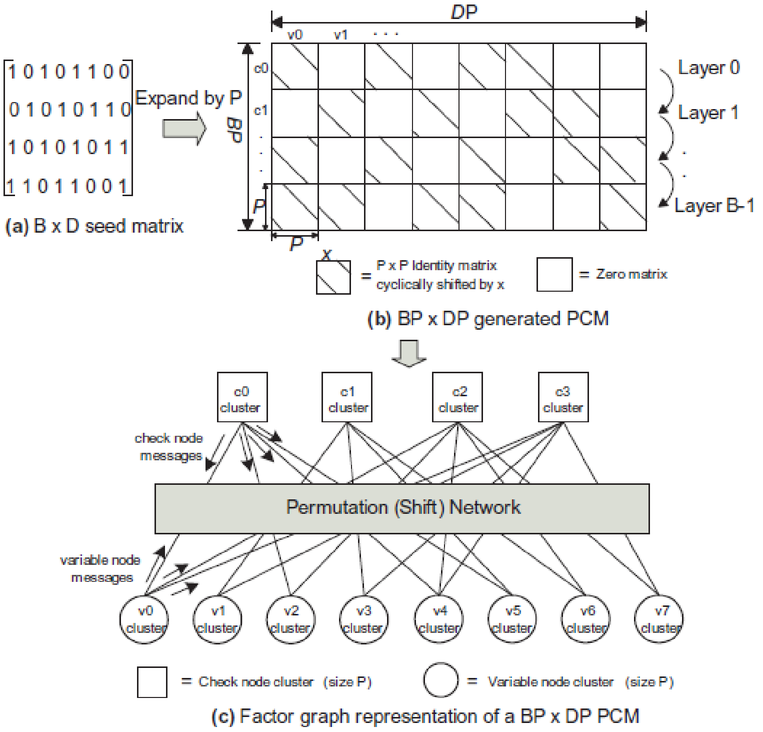
An efficient hardware implementation of binary LDPC decoders is very well investigated. However, efficient hardware implementation of non-binary LDPC decoders is still an open issue, only a few publications exist so far. The authors of [2] present an FPGA implementation of a non flexible LDPC decoder for Galois field 8 only, in logarithm domain. In [3] hardware architecture for the suboptimal extended-min-sum (EMS) algorithm from [4] is presented, but implementation data are missing. In this paper Section 1 discusses about the algorithm in which the layered algorithm and min-sum algorithm are integrated. Section 2 is about the implementation of the architecture in FPGA kit and Section 3 deals with implementation in FPGA kits and Section 4 is about the simulation results obtained.

Algorithm

Layered decoding algorithm

A good trade off between design complexity and decoding throughput is partially parallel decoding by grouping a certain number of variable and check nodes into a cluster for parallel processing. Furthermore, the layered decoding algorithm [6] can be applied to improve the decoding convergence time by a factor of two and hence increases the throughput by 2x. The structured QC-LDPC code makes it effectively suitable for efficient VLSI implementation by significantly simplifying the memory access and message passing. As shown in Fig. 1(b), the PCM can be viewed as a group of concatenated horizontal layers, where the column weight is at most 1 in each layer due to the cyclic shift structure. The belief propagation algorithm is repeated for each horizontal layer and the updated APP (a posteriori probability) messages are passed between layers. Let M_{ij} denote the check node LLR (Log-likelihood ratios) messages sent from the check node i to the variable node j. Let L (p_{ij}) denote the variable node LLR messages sent from the variable node j to the check node i. Let L(p_j) (j = 1, . . . ,N) represent the APP messages for all the variable nodes (coded bits) which are initialized with the channel messages (assuming BPSK on AWGN channel) for each code bit j by $2r_j/\sigma^2$, where σ^2 is the noise variance and r_j is the received value. For each variable node j inside the current horizontal layer, messages L (p_{ij}) that correspond to a particular check equation i are computed according to:

$$L\,(p_{ij}) = L\,(p_j) - M_{ij}. \qquad (1)$$



(a) B x D seed matrix

(b) BP x DP generated PCM

(c) Factor graph representation of a BP x DP PCM

Figure 1: Parity check matrix and its factor Graph representation.

For each check node i, messages Mij, corresponding to all variable nodes j that participate in a particular parity-check equation, are computed according to:

$$M_{ij} = \prod_{j' \epsilon N(j)\backslash\{i\}} sign(L(p_{ij'}))\, \Psi\Big[\sum_{j' \epsilon N(i)\backslash\{j\}} \Psi(L(p_{ij'}))\Big] \qquad (2)$$

where N (i) is the set of all variable nodes from parity check equation i,

$$\Psi(x) = -log\left[\tanh\left(\frac{|x|}{2}\right)\right]$$

The APP messages in the current horizontal layer are updated by:

$$L\,(p_j) = L\,(p_{ij}) + M_{ij.} \qquad (3)$$

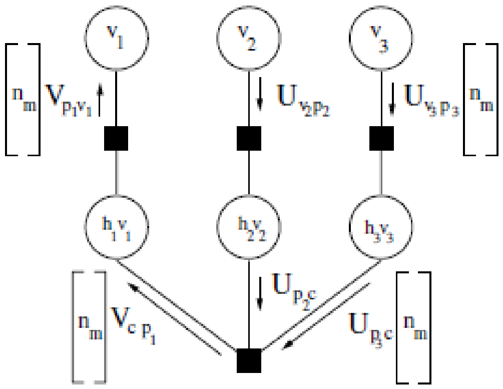


Figure 2: Factor graph representation

V	-	Variable node
n_m	-	largest values of messages at the Input of check node
$V_{p(i)v}$	-	set of messages entering into a variable node v
$U_{vp(i)}$	-	output messages of variable node
$V_{cp(i)}$	-	input messages of check node
$U_{p(i)c}$	-	output messages of check node
d_v	-	degree of variable node
d_c	-	degree of check node

Min-sum algorithm and fixed-point implementation

The belief propagation algorithm [7] is the most powerful iterative soft decoding algorithm for LDPC codes. But due to its high design complexity in (5), many implementations for decoding LDPC codes are based on the modified (normalized or offset) min-sum algorithm because of its satisfactory performance and simple implementation [8]. By applying the offset min-sum algorithm, equation (2) is reduced to:

$$M_{ij} \approx \prod_{j' \epsilon N(i)\backslash\{j\}} sign\left(L(p_{ij'})\right) \times max\left(\min_{j' N(i)\backslash\{j\}} |L(p_{ij'})| - \beta, 0\right) \quad (4)$$

Implementation

As shown in Fig.3, the PE inputs are *wr* elements comprising of L (pj) and Mij, where *wr* is the number of nonzero values in each row of the PCM. L (p_{ij}) is calculated based on (1). The sign and magnitude of L (pij) are processed based on (4) to generate new Mij. Then the L (pij) is added to the Rij to generate new L (pj) (*wr* of them) based on (3). The outputs (L (pj) and Mij) of all the P_x PEs are concatenated and stored in one address of the APP and Check memories. For each layer's sub-iteration, it takes about 2*wr* clock cycles to process, so the decoding throughput is:

$$\text{Throughput} \approx \frac{D \times P_x \times R \times fclk_{max}}{2 \times E \times iterations}$$

where R is the code rate and E is the total number of edges between all variable nodes and check nodes in the seed matrix. Clearly, the throughput would be linearly proportional to the expansion factor Px for a given seed matrix.

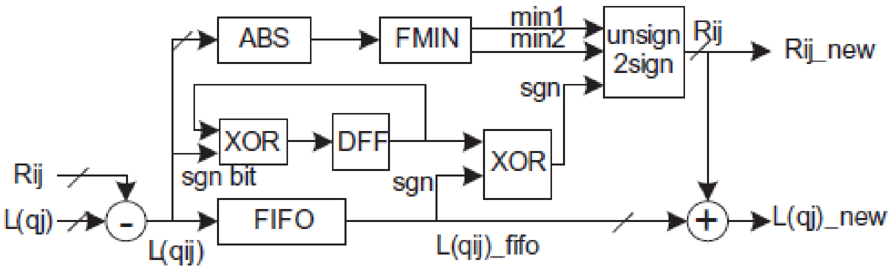


Figure 3: Processing Element

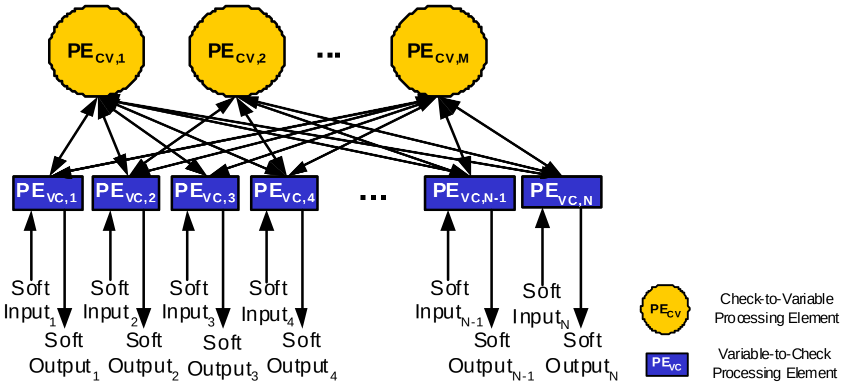


Figure 4: Parallel Architecture for LDPC decoders

Fig.4.shows the parallel architecture for LDPC decoders, where passing of messages and updating of messages can be clearly understood. The characteristics of parallel architecture are

- High Throughput Efficiency
- Improved Power Efficiency
- Complex Interconnect
- Improved error rate performance

Table 1: Code Rate versus seed matrix

Rate	H_{seed}	Rate	H_{seed}	Rate	H_{seed}
¼	18×24	3/5	10×25	5/6	4×24
1/3	16×24	2/3	8×24	7/8	3×24
2/5	15×25	3/4	6×24	8/9	3×27
½	12×24	4/5	5×25	9/10	3×30

The decoding throughput can be further improved by overlapping the decoding of two layers using a pipelined method. The decoding of each layer of the parity check matrix is performed in two stages: 1) Memory read and min-sum calculation and 2) Memory write back. However, due to the possible data dependence between two consecutive layers (there is no data dependency inside each layer because the column weight is at most 1 in each layer), a pipelining data hazard might occur.

Layer i	Read/Min-Sum	Write back	
	Layer i+1	Read/Min-Sum	Write Back

Figure 5(a): Two layer pipeline decoding

The simulation result of the implementation part is done with Xilinx software. The above processing element is simulated and synthesized. Spartan 3 kit is used for implementation of non-binary LDPC decoders.

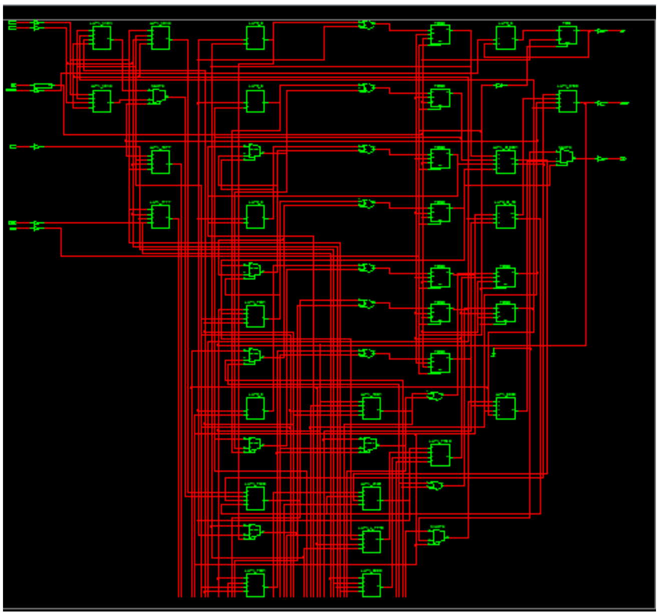


Figure 5(b): Layout of pipelined decoder

Simulation Results

The simulation results are obtained using Matlab. The sparsity patterns of the parity check matrix H and generator matrix G are obtained. A graph with a number of iterations in Belief Propagation algorithm is plot with n=15. The dimensions of the parity check matrix is m=10 and n=15.

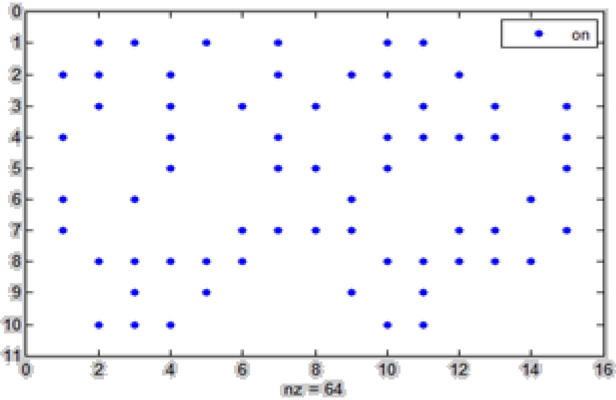


Figure 6: Parity Check Matrix H

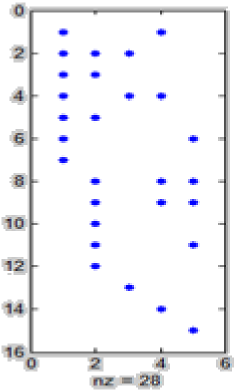


Figure 7: Generator matrix G

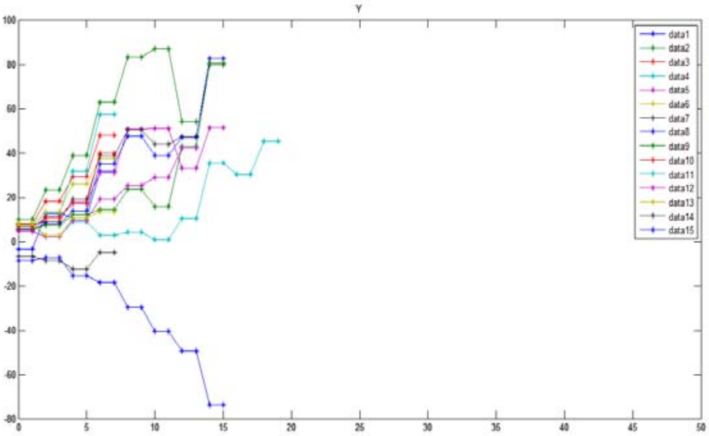


Figure 8: Iterate BP

Table 1: Device utilization

Logic Utilization	Used	Available	Utilization
Number of slice flip flops	10	1,536	1%
Number of 4 input LUTs	31	1,536	2%

Table 2: Hardware synthesis result

Logic distribution	Used	Available	Utilization
Number of occupied slices	17	768	2%
Number of slices containing only related logic	17	17	100%
Number of slices containing unrelated logic	12	17	8%
Number of 4 input LUTs	31	1,536	2%
Number of bonded IOBs	10	63	15%
Number of BUFGMUXs	1	8	12%

Conclusion

From the Simulation result it is so obvious that the NB LDPC in GF(16) with EMS and turbo layered approach under log-like hood domain is giving best performance and in simulation and synthesis of LDPC. Fig (3) gives the idea of processing element of LDPC encoder For 2 bit of operation. Fig (6) and (7) plots the Generator and Parity check matrices for 10X 15. Decoding done by EMS algorithm and the result is shown in Fig (8) Iterative BP. Decoding error in reduced when the iteration EMS (modified BP) algorithm. The synthesis results from Xilinx software is shown in Table 1 and 2 and the number of LUT is utilized only 2% for this hardware implementation. Hence the Hardware complexity is also reduced

References

[1] Timo Lehnigk-Emden, Norbert Wehn *"Complexity Evaluation of Non-binary Galois Field LDPC Code Decoders"* in 2010 6th International Symposium on Turbo Codes& Iterative Information Processing,

[2] C. Spagnol, W. Marnane, and E. Popovici, "*FPGA Implementations of LDPC over GF(2m) Decoders*," in Proc. IEEE Workshop on Signal Processing Systems, Oct. 2007, pp. 273-278.

[3] A. Voicila, F. Verdier, D. Declercq, M. Fossorier, and P. Urard, "*Architecture of a low-complexity non-binary LDPC decoder for high order fields*," in Proc. International Symposium on Communications and Information Technologies ISCIT '07, Oct. 2007, pp. 1201-1206.

[4] A. Voicila, D. Declereq, F. Verdier, M. Fossorier, and P. Urard, "*LowComplexity, Low-Memory EMS Algorithm for Non-Binary LDPC Codes"*

in Proc. IEEE International Conference on Communications ICC '07, Jun. 2007, pp. 6371-676.

[5] M. Karkooti, P. Radosavljevic, and J. R. Cavallaro, *"Configurable, High Throughput, Irregular LDPC Decoder Architecture:Tradeoff Analysis and Implementation,"* IEEE 17th International Conference on Application specific Systems, Architectures and Processors, pp. 360–367, Sep. 2006

[6] M. M. Mansour and N. R. Shanbhag, *"High-throughput LDPC decoders,"* IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 11, pp. 976–996, Dec. 2003.

[7] R. Gallager, *"Low-density parity-check codes,"* IEEE Transactions on Information Theory, vol. 8, pp. 21–28, Jan. 1962.

[8] J. Chen, A. Dholakia, E. Eleftheriou, M. Fossorier and X. Hu, *"Reduced-Complexity Decoding of LDPC Codes,"* IEEE Transactions on Communications, vol. 53, pp. 1232–1232, 2005.