THE INTERNATIONAL JOURNAL OF SCIENCE & TECHNOLEDGE

FPGA Implementation of LDPC Codes

Hema Tewatia
M. Tech Student, Department of Electronics and Communication Engineering
AFSET, Faridabad, Haryana, India
Javed Ashraf
Assistant Professor, Department of ECE, AFSET, Faridabad, Haryana, India
O. P. Malk
Head, Department of ECE, AFSET, Faridabad, Haryana, India

Abstract:
This thesis is about FPGA implementation of LDPC codes and their performance evaluation. Low-density parity-check (LDPC) codes are forward error-correction codes, first proposed in the 1962 PhD thesis of Gallager at MIT. At the time, their incredible potential remained undiscovered due to the computational demands of simulation in an era when vacumm tubes were only just being replaced by the first transistors.
Our codes are found by optimizing the degree structure of the underlying graphs. We develop several strategies to perform this optimization. We also present some simulation results for the codes found which show that the performance of the codes is very close to the asymptotic theoretical bounds.

1. Introduction

In recent years, the revival of LDPC codes has breathed new life into the ability of Forward Error Control (FEC) codes to enable communication systems to reach the Shannon performance bound. Having been dismissed after discovery by Gallagher in 1963 due to their practical unfeasibility, LDPC codes have recently proven invaluable, enabling communication systems to perform within 0.04 dB of the Shannon limit. Up to the rediscovery of LDPC codes by McKay and Neal turbo codes, discovered in 1993 was the class of codes that offered the best performance. LDCP codes are linear block codes constructed by designing a sparse parity check matrix. Contrary to conventional decoding algorithms, the iterative decoding algorithm of LDPC codes is low in complexity, given that the parity check matrix is sparse. The complexity of this decoding algorithm is linear in the codeword length and exponentially related to the sparseness of the parity check matrix. When discovered, the decoding of this class of codes was not practically possible, but with the processing power available today, these codes are practical and very promising.

1.1. Error Correction Using Parity-Checks

When we consider binary messages and so the transmitted messages consist of strings of 0's and 1's. The essential idea of forward error control coding is to augment these message bits with deliberately introduced redundancy in the form of extra check bits to produce a codeword for the message. These check bits are added in such a way that codewords are sufficiently distinct from one another that the transmitted message can be correctly inferred at the receiver, even when some bits in the codeword are corrupted during transmission over the channel. The simplest possible coding scheme is the single parity check code (SPC). The SPC involves the addition of a single extra bit to the binary message, the value of which depends on the bits in the message. In an even parity code, the additional bit added to each message ensures an even number of 1s in every codeword.

1.2. Low Density Parity check Codes (LDPC)

As their name suggests, LDPC codes are block codes with parity-check matrices that contain only a very small number of non-zero entries. It is the sparseness of H which guarantees both a decoding complexity which increases only linearly with the code length and a minimum distance which also increases linearly with the code length.

LDPC codes are constructed by defining the parity check matrix H. If the parity check matrix A has N columns and M rows, any codeword generated for this LDPC code consists of N bits which satisfy M parity checks, where the location of a 1 in the parity check matrix indicates that a bit is involved in a parity check. The total length of the codeword is N bits, the number of message bits is K= N - M, and the rate of the code is R = K / N, assuming that the matrix is full rank.

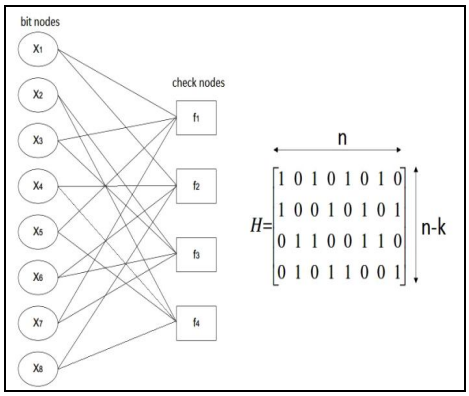

Figure 1: Tanner graph made for a simple parity check matrix H

1.3. Encoding

As stated, LDPC codes are linear block codes constructed with a parity check matrix with special roperties. Known from the theory of linear block codes, given a systematic block code $C_b(n,k)$, the generator matrix is given by

$$\mathbf{G} = \begin{bmatrix} p_{00} & p_{01} & \cdots & p_{0,n-k-1} & 1 & 0 & 0 & \cdots & 0 \\ p_{10} & p_{11} & \cdots & p_{1,n-k-1} & 0 & 1 & 0 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k,n-k-1} & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

Also written as

$$\mathbf{G} = [\mathbf{P}\mathbf{I}_k]$$

where **P** is the k×(n-k) parity sub-matrix and $\mathbf{I}_k$ identity matrix. The systematic form of the parity check matrix **H** is

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & \cdots & 0 & p_{00} & p_{10} & \cdots & p_{k-1,0} \\ 0 & 1 & \cdots & 0 & p_{01} & p_{11} & \cdots & p_{k-1,1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & p_{0,n-k-1} & p_{1,n-k-1} & \cdots & p_{k,n-k-1} \end{bmatrix}$$

Also written as,

$$\mathbf{H} = [\mathbf{I}_{n-k}\mathbf{P}^{\mathbf{T}}]$$

The generated codeword with message vector m of length k,

C = m.H

1.4. Decoding

The decoding algorithm is called the sum-product algorithm, or belief propagation algorithm. This algorithm determines the a posteriori probability of each message symbol, based on the received signal and the connectivity properties of the sparse parity check matrix **H**. This algorithm attempts to satisfy the parity check equations with maximum likelihood do produce the optimal estimate for the original message m. The functioning of this algorithm will be described with an example.

By using a linear block code C (12,4) of code rate 1/3, with

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and sparse parity check matrix

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

Here we receives the corrupt codeword vector.

2. Bounded Distance and Maximum Likelihood Decoding

For any linear code, A= 1, meaning that there is precisely one codeword of weight zero. For good codes, A = 0 for all j less than some value d, called the minimum distance. A code with minimum distance d can always correct
[(d-1)/2] errors using a bounded distance decoder (BDD). Imagine the codeword vectors as points in space. No two words are closer together than the minimum distance, d. If we draw spheres around each codeword of radius [(d-1)/2], no two spheres will overlap. If no more than errors are made by the channel,
the received word will lie within the sphere of the transmitted word, and thus be correctly decoded.
Figure below illustrates this decoding. The smaller circles represent codewords, and the large circles a radius of . The codeword in centre was transmitted.
If less than [(d-1)/2] errors are made, the received word resembles the small circle labelled A and is clearly within the sphere for the desired codeword. If slightly more errors are made, the result could be something like small circles B or C. A bounded distance decoder would make an error in both of these cases. However, the circle labelled C, though outside the sphere of radius [(d-1)/2] closer to the transmitted codeword than any other codeword.
A maximum likelihood decoder always finds the closest codeword to the received word. Because of this, it can decode more than [(d-1)/2] errors some of the time.

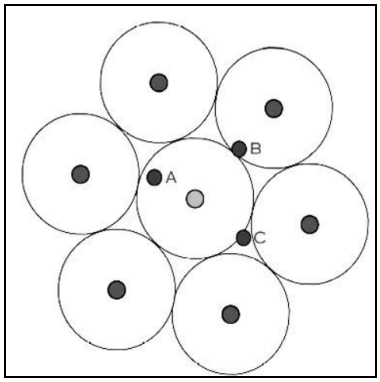

Figure 2: message received on bounded region map

For a code of rate R and length n, there are 2^Rn codewords. A maximum likelihood decoder has to find the distance between the received word and each of the codewords in order to choose the
smallest one. So, while the maximum likelihood decoder can correct the most errors, its complexity grows exponentially with the length of the codewords. For this reason, iterative decoding methods, which have complexity that still grows linearly with the length of the codewords, are much preferred.

3. Message Passing Decoding

Message passing is easiest to understand on the binary erasure channel (BEC). This channel introduces no errors, but erases some message bits. In the Tanner graph representation, then, each variable node either knows with certainty what its value is, or it does not. Decoding starts when the variable nodes send messages to their adjacent check nodes that indicate whether or not the variable node knows its value. The check nodes examine the messages they received from their adjacent variable nodes. If all the adjacent variables but one knew their value, the check node can determine the value of the remaining node because even parity is required. A round is completed when all the check nodes that can make this calculation send a message to the last variable node, letting it know its value.
Check nodes connected to variables that all know their value can be removed from the decoding process. The cycle then repeats. With every round, more variable nodes learn their true values, until all is known or no more progress can be made. When no more progress can be made, the set of erasures remaining is known as a stopping set. Since the deep space channel is very close

to BEC characteristics, and is suitable for large size codes, the message passing decoding (sum product algorithm) was taken for implementation.

4. References

1. Consultative Committee for Space Data Systems (CCSDS). Low density Parity Check Codes for use in NearEarth and Deep Space Applications, Sept. 2007. Orange book, available at http://public.ccsds.org/publications/archive/131x1o2e1.pdf
2. Z. Li. "Efficient Encoding of Quasi-Cyclic Low-Density Parity-Check Codes." IEEE Transactions on
3. Communications 54, no. 1 (January 2006): 71-81.
4. S. Lin and D. Costello, Jr. Error Control Coding. 2nd ed. New Jersey: Pearson Prentice Hall, 2004.
5. D. Divsalar, S. Dolinar, and C. Jones. "Construction of Protograph LDPC Codes with Linear Minimum
6. Distance." In Proceedings of the IEEE International Symposium on Information Theory (Seattle, Washington).
7. Piscataway, NJ: IEEE, July 2006.
8. M. Mansour and N. Shanbhag, \High-throughput LDPC decoders," Very Large Scale Integration (VLSI)
9. Systems, IEEE Transactions on, vol. 11, pp. 976{996, Dec. 2003
10. S. Johnson, "Introduction to ldpc codes," in ACoRN Spring School on Coding, Multiple User Communications
11. and Random Matrix Theory, 2006
12. K. Zhang, X. Huang and Z. Wang, "High-throughput layered decoder implementation for quasi-cyclic LDPC
13. codes," IEEE J. Selected Areas Commun., vol. 27, no. 6, Aug. 2009
14. Z. Wang, Z. Cui, "Low-complexity high-speed decoder design for quasicyclic LDPC Codes," IEEE Trans. VLSI
15. Systems, vol. 15, no. 1, Jan. 2007
16. Abhishek Kumar,Madhusmita Mishra,Sarat Kumar Patra" Performance Evaluation and Complexity analysis of Re-jagged AR4JA code over AWGN channel" International Journal of Scientific & Engineering Research, Volume 4, Issue 6, June 2013 ISSN 2229-5518
17. Hu Y. and E.Eleftheriu, "Efficient Implementations of the Sum-Product Algorithm for Decoding LDPC
18. codes", Global Telecommunications Conference 2001,Volume 2, pp. 1036, 2001
19. Karkooti M. and J.R. Cavallaro, "Semi-parallel reconfigurable architectures for real- time LDPC decoding",
20. IEEE International Conference on Information Technology 2004, Volume 1, pp. 579 – 585, April 2004.
21. Lee W.L. and A.Wu, "VLSI implementation for low density parity check decoder", IEEE International Conference on Electronics, Circuits and Systems 2001, Volume 3, pp. 1223-1226, 2001.
22. Theocharides T., G.Link and E.Swankoski, "Evaluating alternative implementations for LDPC decoder check node function", IEEE Computer society Annual Symposium on VLSI 2004, pp.77-82, 2004.
23. Wymeersch H., H. Steendam and M. Moeneclaey, "Log-domain decoding of LDPC codes over GF(q)", IEEE International Conference on Communications 2004, Volume 2, pp. 772-776, June 2004.
24. Karkooti M. and P. Radosavljevic, "Configurable, High Throughput, Irregular LDPC Decoder Architecture: Tradeoff Analysis and Implementation", International Conference on Application-specific Systems, Architectures and Processors 2006, pp. 360-367, September 2006.
25. Liao E.,Y. Engling, B. Nikolic,"Low-density parity-check code constructions for hardware Implementation", IEEE International Conference on Communications, Volume 5, pp. 2573-2577, June 2004