Low Complexity Implementation of LDPC Decoder using MIN-Sum Algorithm

Geo Niju Shanth
P.G Student
Velammal Engineering College
Chennai

Saru Priya
Assistant Professor
Velammal Engineering College
Chennai

ABSTRACT

This paper presents a resource efficient LDPC decoder architecture. The algorithm used for decoding LDPC is the min-sum algorithm. The decoder reduces the inter-connect complexity by restricting the extrinsic message length to 2 bits and also simplifies the check node operation. The algorithm is simulated and the results show that the performance is better than that of other algorithms. This algorithm can be incorporated into partially parallel hardware architecture to get significant savings in hardware resources when implemented in FPGA.

General Terms

Digital communication, Error Correction, VLSI

Keywords

Error correction coding, LDPC code, Iterative decoding, VLSI design

1. INTRODUCTION

Low-density parity-check codes are linear block codes. They were first proposed by Gallager in 1960 [1]. As the computational complexity of implementing LDPC codes is very high compared to other codes they were ignored until recently. LDPC codes were resurrected by Mackay[2] and others who found the advantages of linear block codes which possess sparse parity check matrix. LDPC codes are the most attractive error correcting codes because of its high performance and also it is suitable for high data rate applications like wimax and dvbs2 [3]. The performance of LDPC codes is very close to the capacity for lot of different channels. The structure of LDPC code provides high degree of parallelism in practical implementation [4].

2. LDPC REPRESENTATION

The parity check matrix of an LDPC code can be represented either by a matrix or by using tanner graph. The two sets of nodes in a tanner graph are called check nodes and variable nodes. Check nodes represent the rows of the matrix and the variable nodes represent the columns of the matrix. A parity check matrix and its corresponding tanner graph are shown in Fig.1.

The a^{th} check node is connected to the b^{th} variable node if and only if $H_{a,b}=1$. Check nodes f_0...f_5 represent the six rows of the matrix, whereas v_0...v_{11} are the columns. The number of edges in each check node is equal to the column weight. The row and column weights are four and two respectively in this example. A cycle in a parity check matrix is formed by a complete path through ‘1’ entries with alternating moves between rows and columns. In a tanner graph a cycle is formed by a path starting from a node and ending at the same node. The length of the cycle is given by the number of edges in the path. The smallest cycle on a tanner graph or parity check matrix is called its girth. The smallest possible girth is four. A bipartite graph has a minimum cycle of length four and has even cycle lengths.

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

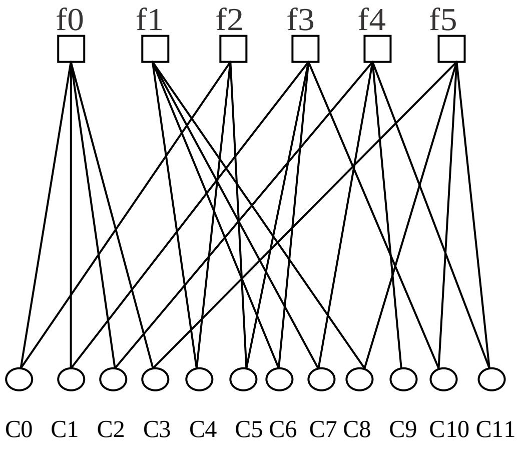

Fig.1. Parity check matrix and its tanner graph representation

3. REGULAR AND IRREGULAR LDPC CODES

An LDPC code is called regular if each row and each column has constant number of 1's. If each row and each column has variable number of 1's then it is called an irregular LDPC code.

4. CONSTRUCTING LDPC CODES

There are two methods for constructing LDPC codes namely Gallagar and Mackay.

4.1 Gallagar Codes

Gallagar codes are regular LDPC codes with an H matrix of the form

$$\begin{bmatrix} H_1 \\ H_2 \\ . \\ . \\ . \\ H_n \end{bmatrix}$$

Each sub matrix H_d is $\mu \times \mu w_r$ with row weight w_r and column weight 1. w_r and μ are integers. For i=1,2… μ, the ith row contains all of its w_r 1's in columns (i-1) w_r +1 to w_r for H$_1$.

4.2 Mackay Codes

Mackay provided algorithms to semi randomly generate sparse H matrices. H is created by randomly generating weight w_c columns and uniform row weight. H is created by randomly generating weight w_c columns and uniform row weight while ensuring weight w_r rows and no two columns having overlap greater than 1. The drawback with Mackay code is that the encoding complexity is very high.

5. PERFORMANCE AND COMPLEXITY

The performance of LDPC codes is better only for large block lengths. Large block lengths results in large parity check matrix and generator matrix. The sparse parity check matrix can be put in the form [PTI] through Gaussian elimination. From this the generator matrix can be calculated as G=[I P]. The P matrix is not sparse and hence the encoding complexity will be high. So the complexity rows with increase in block length. To reduce the complexity iterative decoding algorithms are used.

6. DECODING LDPC CODES

The most common method for decoding LDPC codes is the belief propagation algorithm. It is also called the message passing algorithm or the sum product algorithm.

6.1 Sum Product Algorithm

Given the transmitted codeword c={c$_0$,c$_1$,…c$_{n-1}$} and the received word v={v$_0$,v$_1$,…v$_{n-1}$}, we need to calculate a Log Likelihood Ratio(LLR) of the received symbol.

LLR is given by (1)

$$L(v_i) = \log\left(\frac{\mathrm{P_r}(v_i = 0 \mid y_i)}{\mathrm{P_r}(v_i = 1 \mid y_i)}\right) \tag{1}$$

If the signal considered is antipodal and the channel is AWGN, the LLR for the received channel symbol is given by (2).

$$L(v_i) = 2y_i / \sigma^2 \tag{2}$$

Here y_i is the received symbol and σ^2 is th e noise power.

These messages are passed to the check nodes becoming $L(q_{ij})$ from variable node I to check node j. The sign and magnitude of $L(q_{ij})$ are α_{ij} and β_{ij} respectively.

At check nodes these messages gets processed and the messages from check node to variable node is given by (3).

$$L(r_{ji}) = \left[\prod_{i' \in V_j \backslash i} \alpha_{i'j}\right] \cdot \phi\left(\sum_{i' \in V_j \backslash i} \phi(\beta_{i'j})\right) \tag{3}$$

$$\phi(x) = \log\left(\frac{e^x + 1}{e^x - 1}\right) \tag{4}$$

The message $L(r_{ji})$ does not depend upon the message that came from ith variable node

The variable node processes the message received according to the equation given by

$$L(q_{ij}) = L(V_i) + \sum_{j' \in C_i \backslash j} L(r_{j'i}) \tag{5}$$

The above equation shows that the message from variable node to check node is a simple addition of messages coming from all check nodes exceopt the jth node. Processing of messaes by both check node and variable node processor is equivalent to one iteration of the SPA.

After some specified number of iterations the symbols are decoded by comparing $L(Q_i)$ with the threshold.

$$L(Q_i) = L(v_i) + \sum\nolimits_{j \in C_i} L(r_{ji}) \tag{6}$$

The decoded symbol is given by

$$C_i = \begin{cases} 1 & if\,(L(Q_i < 0) \\ 0 & else \end{cases} \tag{7}$$

6.2 Min Sum Algorithm

The Min-Sum algorithm (MSA) [5] is the modified version of the Sum-Product algorithm (SPA) [5]. Here the check node operation is simplified to reduce the complexity of the algorithm. In MSA, the quantized intrinsic message, also known as log-likelihood ratio (LLR) and the extrinsic messages (between variable and check nodes) are equal in length. Hence, the hardware implementation complexity increases as the quantized message length increases. The check node update equation is given by

$$L(r_{ji}) = \left[\prod_{i' \in V_j \backslash i} \alpha_{i'j}\right] \cdot \min_{i' \in V_j \backslash i} \beta_{i'j} \tag{8}$$

7. QUANTIZATION OF THE CHANNEL DATA

Since a decoder cannot operate on real values we need to quantize it and the quantized values are imported to the decoder.

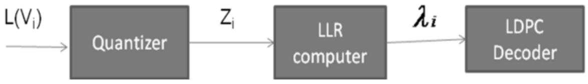

Fig.2. Block diagram of the system

The performance of the algorithm is impacted by the quantization of the soft input messages [6]. Higher the level of quantization higher will be the hardware resource requirement. When we reduce the quantization level it leads to reduction in BER performance. The quantization of the soft input messages affects the performance of the min-sum algorithm. The quantization of the channel data in LDPC decoding is represented by the block diagram shown in fig.2.

8. MODIFIED MIN SUM ALGORITHM

The MMS Algorithm [7] uses higher precision LLR messages and lower precision (2-bit) extrinsic messages. The check node and variable node operations of the MMS algorithm are as follows:

8.1 Variable Node operation

This algorithm is almost similar to the min-sum algorithm. The only difference is that the variable node performs higher precision quantized LLR operations and it maps the result to 2-bit message. The 2-bit message is passed to the check node for processing. The variable node operation is given by (9).

$$Vi = g\left(LLR_n + \sum_{j \neq i} f(C_j)\right) \tag{9}$$

where n=1, 2,…,N (variable nodes) and

i=j=1, 2,…,dv (degree of variable node 'n')

$$g(y) = \begin{cases} 01 & y > T_m \\ 00 & 0 \leq y \leq T_m \\ 10 & 0 > x \geq -T_m \\ 11 & x < -T_m \end{cases} \tag{10}$$

$$f(x) = \begin{cases} +W & x = 01 \\ +w & x = 00 \\ -w & x = 10 \\ -W & x = 11 \end{cases} \tag{11}$$

where T_m is the threshold for mapping, W is the higher integer constant and w is the lower integer constant.

8.2 Check Node Operation

In this step XOR operation (M_k) is used to find the product of the sign of incoming messages. AND operation (S_k) is used to determine the minimums. The output message (C_k) is obtained by concatenating the sign bit and the magnitude bit.

$$S_k = V_1^{(s)} \oplus V_2^{(s)} \oplus \cdots V_l^{(s)} \quad \forall l \neq k \tag{12}$$

$$M_k = V_1^{(m)} \oplus V_2^{(m)} \oplus \cdots V_l^{(m)} \quad \forall l \neq k \tag{13}$$

$$C_k = \{S_k M_k\} \tag{14}$$

9. PERFORMANCE COMPARISON OF THE ALORITHMS

The software model of the Sum Product Algorithm (SPA), Min Sum Algorithm (MSA) and the Modified Min Sum Algorithm (MMS) has been created and the BER performance of each algorithm is plotted using MATLAB. Progressive Edge Growth (PEG) algorithm is used for generating LDPC codes [8]. The codes were modulated using Binary Phase Shift Keying (BPSK). The channel used for transmission is Additive White Gaussian Noise (AWGN) [9]. The simulations were performed for 1200 bit code length at a maximum iteration of 10. Fig.3 shows the BER performance comparison of all the three algorithms.

It can be seen from the plot that the performance of the MMS algorithm is better than the MSA but less than SPA. The implementation complexity of the MMS algorithm is very less. So it gives better performance than MSA and lower hardware complexity than SPA.

10. SIMULATION RESULTS

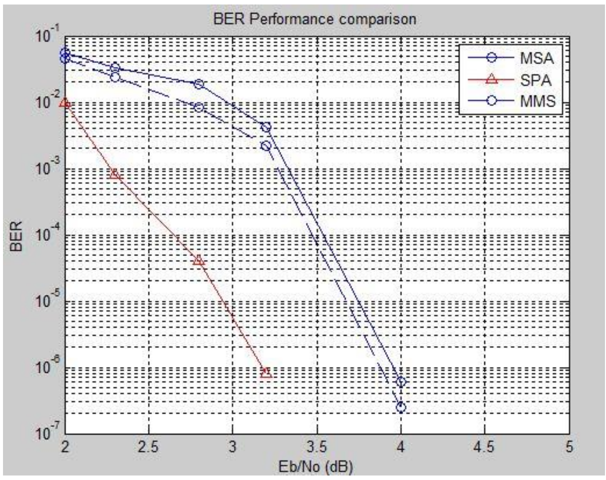

Fig.3. BER performance comparison of Min Sum Algorithm (MSA), Sum Product Algorithm (SPA) and the Modified Min Sum Algorithm (MMS).

11. LDPC DECODER ARCHITECTURES

In most applications LDPC decoding is implemented in hardware to improve speed of processing. The decoding architectures are divided into two main categories: fully-parallel and partially-parallel.

11.1 Fully parallel decoders

Fully parallel architectures resemble a tanner graph of a parity check matrix. Each node of the tanner graph is mapped on to a processing node (check node and variable node) along with all the connections required for passing messages between them. The check nodes and variable nodes are processed in parallel and thus each iteration of decoding can be done in just one clock cycle. The advantage of fully parallel architecture is high throughput and it does not need memory to store intermediate results. The fully parallel architecture is also power efficient. The problem with the fully parallel design is the complexity caused by large number of long global wires between check nodes and variable. It also requires large circuit area. The fully parallel decoder architecture is shown in Fig.4.

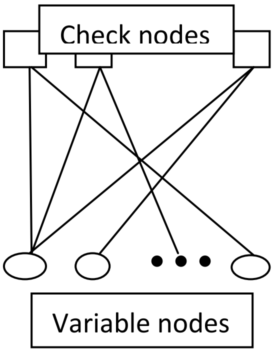

Fig.4. Fully parallel LDPC decoder architecture

11.2 Partially parallel decoders

In this architecture, a subset of variable nodes and check nodes are implemented in hardware, and by changing the routing network between implemented nodes, different partitions of parity check matrix are processed. Since intermediate messages need to be stored, memory resources are essential for this architecture. One iteration of decoding takes multiple cycles. Thus the throughput is lower compared to Fully parallel decoders. However, the decoding circuit is much smaller.

Most of LDPC codes adopted in recent standards belong to Quasi-cyclic class of codes [10]. The parity check matrices of these codes are block structured, which makes them well suited for partial-parallel implementations. The parity check matrix for this class of codes is constructed by smaller sub matrices; each is either an all-zero sub matrix or a permutation of an identity matrix. The partially parallel decoder architecture is shown in fig.5.

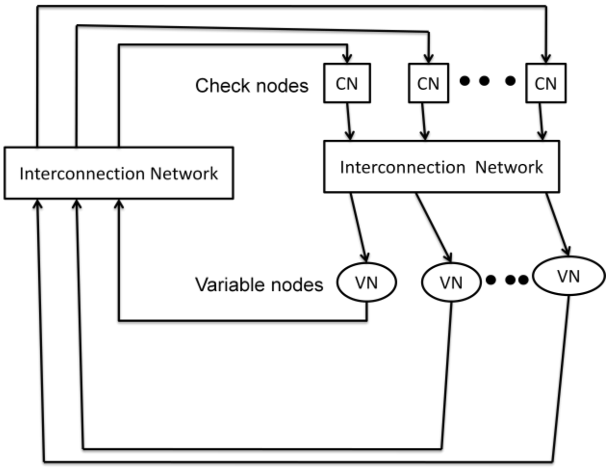

Fig.5. Partially parallel LDPC decoder architecture

12. HARDWARE IMPLEMENTATION

There are number of issues to be considered while designing hardware architecture for LDPC decoder. In partially parallel hardware architecture large memory is required to store intermediate results. Several techniques have been suggested to reduce decoder memory. Most techniques use MPA algorithm or its approximations. The min-sum algorithm reduces complexity by simplifying the check node update. Issues related to its implementation are explored in [11].

Decoding Latency is another critical factor for most applications. The overall decoding time could be reduced by faster convergence of the decoding algorithm and simpler computations. Fully parallel architecture requires large amount of hardware resources compared to partially parallel hardware architecture. The use of MMS algorithm in partially parallel hardware architecture provides more savings in hardware resources compared to previous reported works. It also provides acceptable level of BER performance. The algorithm when incorporated into partially parallel hardware design given in [12] provides further reduction in hardware resource requirement.

13. CONCLUSION

This paper has presented the different algorithms for decoding LDPC codes. The algorithms were simulated using MATLAB and the BER performances of different algorithms were compared. The MMS algorithm provides better performance and also can provide significant reduction in hardware resources when implemented. Various types of hardware architecture were studied. The MMS algorithm when implemented in partially parallel hardware architecture requires very less hardware for implementation.

14. REFERENCES

[1] R. Gallager, Low-density parity-check codes, IRE Transactions on Information Theory 8 (1) (1962) 21–28..

[2] D.J.C. MacKay, Good error-correcting codes based on very sparse matrices, IEEE Transactions on Information Theory 45 (2) (1999) 399–431.

[3] Tetsuo Nozawa, LDPC Adopted for Use in Comms, Broadcasting, HDDs, Tech On, 2005. Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[4] G.L.L. Nicolas Fau, LDPC (Low Density Parity Check)— A Wireless PHY Layers EE Times Network, 2008

[5] A. Anastasopoulos, A comparison between the sum-product and the min-sum iterative detection algorithms based on density evolution, in: Proceedings of the IEEE Global Telecommunications Conference, San Antonio, TX, pp. 1021– 1025, 25–29 November 2001.

[6] R. Zarubica, R. Hinton, S.G. Wilson, E.K. Hall, Efficient quantization schemes for LDPC decoders, in: Proceedings of the 'IEEE Military Communications Conference, San Diego, CA, pp. 1–5, 16–19 November 2008.

[7] Vikram Arkalgud Chandrasetty, Syed Mahfuzul Aziz, An area efficient LDPC decoder using a reduced complexity min-sum algorithm. Integration, the VLSI journal, Volume 45, Issue 2, March 2012.

[8] X.Y.Hu, Software to Construct PEG LDPC code, [cited on May 2009]. Available from: <http://www.inference.phy.cam.ac.uk/mackay/PEG_ECC.html>.

[9] J.G. Proakis, Digital Communications, Fifth ed, McGraw-Hill, New York, 2008.

[10] M. Fossorier. Quasi-cyclic low-density parity-check codes from circulant permutation matrices. IEEE Transaction Information Theory, 50(8):1788–1793, Aug. 2004.

[11] J. Zhao, F. Zarkeshvari, and A.H. BanihaShemi. On the implementation of Min-Sum Algorithm and its modifications for Decoding Low-Density Parity-Check Codes. IEEE Transactions on Communications, 53(4):549–554, April 2005.

[12] K. Shimizu, T. Ishikawa, N. Togawa, T. Ikenaga, S. Goto, Partially-parallel LDPC decoder based on high-efficiency message-passing algorithm, in: Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors, pp. 503–510, 2–5 October 2005.