ISSN (ONLINE) : 2395-695X
ISSN (PRINT) : 2395-695X
Available online at www.ijarbest.com
International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST)
Vol. 1, Issue 8, November 2015

Modified min sum algorithm for low density parity check decoder for FPGA implementation

P.Kavitha

HOD, Mechatronics Engineering, M.A.M.School of Engineering, Trichy, India

***Abstract*—This paper presents a high-throughput decoder architecture for low-density parity-check (LDPC) codes. Various optimizations are employed to increase the clock speed. A reduced complexity Low-Density Parity-Check (LDPC) decoder is designed and implemented on FPGA using a modified Min-Sum algorithm. Simulation Results reveal that the proposed decoder has improvement and requires fewer Decoding iterations compared to original 2-bit min-sum Algorithm. With a comparable bit error rate performance to that of 3- Bit min-sum algorithm, the decoder implemented using Modified 2-bit min-sum algorithm saves about 18% Low-Density Parity-Check (LDPC) codes have become one of the most attractive error correction codes due to its excellent performance and suitability in high data rate applications**

Keywords – Low Density Parity Check, min-sum algorithm, error detecting code, variable node, Check node..

I. INTRODUCTION

Recently Low-Density Parity-Check (LDPC) codes have gained significant attention due to their near Shannon limit performance 2]. They have been adopted in several wireless standards, such as DVB-S2 [3], IEEE 802.16e [4] and IEEE 802.11n [5], because of their\excellent error correcting performance. A LDPC code is a linear block code defined by a sparse parity check matrix It can be represented by a bipartite graph, called Tanner Graph [6] which contains two sets of nodes: variable nodes that represent the bits of a codeword and check nodes that implement the parity check constraints. The standard decoding procedure is the message passing algorithm, also known as "sum-product" or "belief propagation" (BP) algorithm [2], which iteratively exchanges the messages between the check nodes and the variable nodes along the edges of the graph. In the original message passing algorithm, the messages first are broadcasted to all the check nodes from the variable nodes and then along the edges of the graph the updated messages are fed back from the check nodes to the variable nodes to finish one iteration of decoding.

Low density parity check codes are error correcting codes, first discovered by Robert Gallager in the 1960′s, but were largely forgotten because of the encoding and decoding complexity at the time [1]. Although some work was done in the 1980′s, it was not until they were independently rediscovered, starting with work by David MacKay, that their use began to proliferate. Basically, an LDPC code consists of a sparse matrix with very few ones in each row and column, called the "Parity Check Matrix". Usually, each row of the matrix is designed to be linearly independent from the other rows, which is an assumption we have made for our project. Regular codes have an equal number of non-zero elements in each column and an equal number in each row, while irregular codes can have a varying number.

An example of an LDPC parity check matrix with three 1′s in each column and four 1′s in each row, created by Gallager. When multiplied by a column vector "codeword" in modulo 2 arithmetic, the result should be a column vector of 0′s. The code word consists of a block of "parity" check bits, which has the same length as the number of rows, and a block of data bits. The rate of the code is the number of data bits divided by the number of columns in the matrix. The process of encoding is determining the set of parity bits that, when concatenated with the data bits, will multiply the parity check matrix and create a column of zeros. The decoding process determines the set of bits that were transmitted with this property based on the received packet.

LDPC codes are powerful because, with a sufficiently long block length, one can approach Shannon's capacity limit for a noisy channel with arbitrary precision. In some cases, LDPC codes can be constructed to have better performance than Turbo codes, another widely used error correction code. Additionally, LDPC codes are advantageous in that they use a lower complexity iterative decoding "belief propagation" algorithm which can be implemented in parallel in hardware. The decoder is also very good at detecting errors in the received codeword while also determining when it is unable to correctly decode the packet. Although the encoding complexity is somewhat high, the powerful properties of LDPC codes have warranted their inclusion in many standards such as IEEE 802.16, 802.20, 802.3 and DVB-RS2 .

1.1 Representations for LDPC codes

Basically there are two different possibilities to represent LDPC codes. Like all linear block codes they can be described via matrices. The second possibility is a graphical representation.

1.1.1 Matrix Representation

Let's look at an example for a low-density parity-check matrix first. The matrix defined in fig 1.1 is a parity check matrix with dimension n ×m for a (8, 4) code. We can now define two numbers describing these matrix. w_r for

ISSN (ONLINE) : 2395-695X
ISSN (PRINT) : 2395-695X
Available online at www.ijarbest.com
International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST)
Vol. 1, Issue 8, November 2015

the number of 1's in each row and w_c for the columns. For a matrix to be called low-density the two conditions w_c<<n and w_r<< m must be satisfied.

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

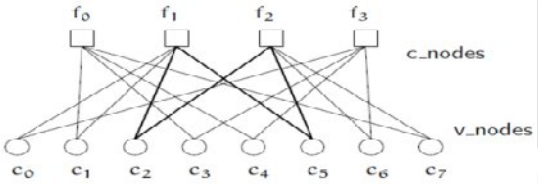


FIGURE 1.1 TANNER GRAPH

In order to do this, the parity check matrix should usually be very large, so the example matrix can't be really called low-density.

1.1.2. *Graphical Representation*

Tanner introduced an effective graphical representation for LDPC codes. Not only provide these graphs a complete representation of the code, they also help to describe the decoding algorithm as explained. Tanner graphs are bipartite graphs. That means that the nodes of the graph are separated into two distinctive sets and edges are only connecting nodes of two different types. The two types of nodes in a Tanner graph are called variable nodes (v-nodes) and check nodes (c-nodes).

A Tanner graph represents the same code as the matrix in 1. The creation of such a graph is rather straight forward. It consists of m check nodes (the number of parity bits) and n variable nodes (the number of bits in a codeword). Check node is connected to variable node if the element h_{ij} of H is a 1.

1.2 Regular and irregular LDPC codes

A LDPC code is called regular if w_c is constant for every column and $w_r = w_c \cdot (n/m)$ is also constant for every row. The example matrix from equation (1) is regular with $w_c = 2$ and $w_r = 4$. It's also possible to see the regularity of this code while looking at the graphical representation. There is the same number of incoming edges for every v-node and also for all the c-nodes. If H is low density but the numbers of 1's in each row or column aren't constant the code is called a irregular LDPC code.

There are different ways to implement the LDPC decoder, based on the number of processing units available. In general, the LDPC decoder architecture can be classified into three types: fully parallel architecture, serial architecture and partial parallel architecture. In fully parallel architecture [7], a check node processor is needed for every check node, which usually results in large hardware cost and complicated routing, and hence is less flexible. The serial architecture uses just one check node processor to share the computation of all the check nodes and is too slow for most applications. For partial parallel architectures, multiple processing units are used allowing proper tradeoff between the hardware cost and the throughput and are commonly adopted in the actual implementation [9]–[23]. In order to achieve higher convergence speed, i.e., to minimize the number of decoding iteration, serial message passing algorithm, also known as layered decoding algorithm, has been proposed [8], [9] together with the corresponding partial parallel architecture.

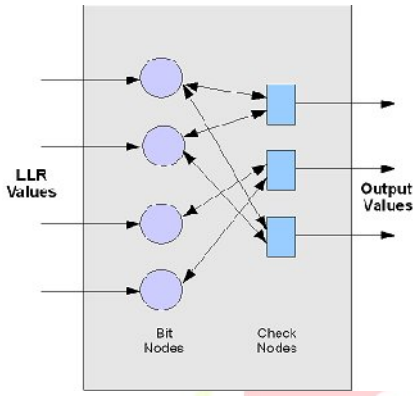
There are two types of layered decoding schemes: vertical layered decoding and horizontal layered decoding [8]. In the horizontal layered decoding, a single or a certain number of check nodes (called layer) are first updated. Then the whole set of neighboring variable nodes are updated, and the decoding process proceeds layer after layer. Dually, in the vertical layered decoding, a single or a certain number of variable nodes (layer of variable nodes) may be updated first. Then the whole set of neighboring check nodes are updated [11]. Because the serial check node processor is easier to be implemented in VLSI and therefore the horizontal layered decoding is preferable for practical implementations [11]. Because of the faster convergence and regular architecture, layered decoders are commonly found in the LDPC decoder implementation [9]–[18].

In this work, we focus on the LDPC decoding implementation based on the layered decoding algorithm. Low-density parity-check (LDPC) codes are a class of linear block codes. The name comes from the characteristic of their parity-check matrix which contains only a few 1's in comparison to the amount of 0's. Their main advantage is that they provide a performance which is very close to the capacity for a lot of different channels and linear time complex algorithms for decoding. Furthermore are they suited for implementations that make heavy use of parallelism. The LDPC Decoder core provides designers an LDPC Decoder block used in DVB-S2 systems. LDPC codes have easily parallelizable decoding algorithms. The parallelizability is 'adjustable' providing an option to choose between throughput and complexity. LDPC decoder along with BCH decoder forms a Forward Error Correction (FEC) block, which is used in DVB-S2. During encoding, the information data is added with redundant data called parity data. This parity data is useful in detecting the errors that are introduced during the transmission of information data through channel. DVB-S2 is the second-generation for satellite broad-band applications, developed by Digital Video Broadcasting (DVB) Project in 2003. This is the first standard that uses LDPC mechanism for error detection and correction. It's a single and very flexible standard that covers variety of satellite broadcasting applications. This system also have it's applications in interactive services, professional applications such as digital TV contribution, news gathering, data content distribution and Internet trunking.

Fig 1.2. Functionality of LDPC decoder.

Figure shows the functionality of the LDPC Decoder unit. The encoded data after modulation is transmitted through a channel. At the receiver end, the data from the channel is demodulated to generate Log Likelihood Ratio (LLR) values. This LLR values gives a probability value indicating the amount of certainty that a transmitted bit is either 0 or 1.

ISSN (ONLINE) : 2395-695X
ISSN (PRINT) : 2395-695X
Available online at www.ijarbest.com
International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST)
Vol. 1, Issue 8, November 2015



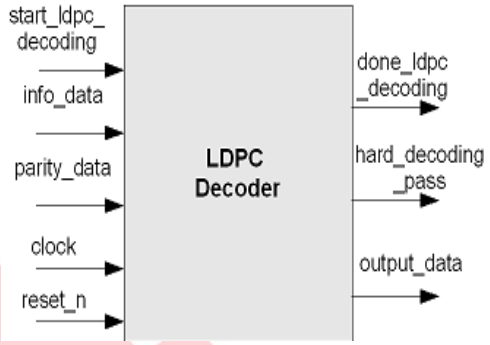


Fig.1.3.Block Diagram Of LDPC Decoder

For example, a very high negative LLR value indicates a very good 1 value. A low negative LLR value indicates that the transmitted bit can be 1. Similarly, a very high positive LLR value indicates a very good 0 value and a low positive LLR value indicates that the transmitted bit can be zero. The core contains Bit Node units and Check Node Units. The transmitted LLR channel values are fed as input to the LDPC Decoder. Bit node and Check node units communicate with each other to detect and correct the errors present in the transmitted data.

The communication between bit node and check nodes is iterative. This process ends when the decoder converges to a code word or maximum iteration length is reached. The decoding algorithm has the following four stages.

Initialization:

The channel LLR Values are assigned to the edges that goes out from Bit Node Unit.

Check Node Update:

Check Node Units receives the data from Bit Node edges and process the data according to Min Sum algorithm. The processed data is transmitted to Bit Node Units.

Bit Node Update:

All the edges that goes out from Check node to Bit node are added to compute a sum value. This Sum value is used for Hard Decoding. Bit Node value is updated by subtracting its value from sum value.

Hard Decoding:

Depending on the sign of Sum value, the transmitted data is determined. Using these values, the parity check equations are computed. If all parity equations are satisfied then the decoder stops, otherwise another Check Node and Bit Node Update is performed.

Figure 1.3 shows the schematic block diagram of LDPC Decoder.

With the assertion of start_ldpc_decoding signal, the decoder core starts receiving data from the pins info_data and parity_data at a rate, 360 LLR values per clock cycle. Once a complete frame of data is received, the decoding process starts. When the decoding process finishes, the LDPC coreasserts done_ldpc_decoding signal indicating the end of decoding. hard_decoding_pass signal indicates the success of ldpc decoding. The decoded data is transmitted out at a rate 360 bits per clock.

1.3.Constructing LDPC codes

Several different algorithms exists to construct suitable LDPC codes. Gallager himself introduced one. Furthermore MacKay proposed one to semi-randomly generate sparse parity check matrices. This is quite interesting since it indicates that constructing good performing LDPC codes is not a hard problem. In fact, completely randomly chosen codes are good with a high probability. The problem that will arise, is that the encoding complexity of such codes is usually rather high.

.

III. VARIABLE NODE OPERATION

The variable node operation is similar to that of the original Min-Sum algorithm. The difference in the proposed algorithm is that the variable node (Vi) performs higher precision quantized LLR operations (LLRn), but maps the computed result to 2-bit message to be passed to the check nodes. The 2-bit message consists of a sign bit and a magnitude bit representing the computed LLR sum. The mapping is based on a threshold (Tm) obtained from simulations [6]. Depending on the message received from the check nodes (Cj), the 2-bit information is again mapped to constant values (±W or ±w) to perform the LLR sum operation in the variable node. These constant values for mapping are also obtained from

ISSN (ONLINE) : 2395-695X
ISSN (PRINT) : 2395-695X
Available online at www.ijarbest.com
International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST)
Vol. 1, Issue 8, November 2015

simulations.

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

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

$$g(y) = \begin{cases} 01 & if, \quad y > T_m \\ 00 & if, \quad 0 \le y \le T_m \\ 10 & if, \quad 0 > x \ge -T_m \\ 11 & if \quad x < -T_m \end{cases}$$

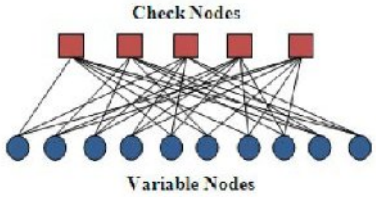
$$f(x) = \begin{cases} +W & if, \quad x = 01 \\ +w & if, \quad x = 00 \\ -w & if, \quad x = 10 \\ -W & if, \quad x = 11 \end{cases}$$

Where, Tm is the optimized threshold for mapping obtained from simulations; W is the optimized higher integer constant obtained from simulations; w is the optimized lower integer constant obtained from simulations. Monte Carlo simulations are carried out to obtain Tm, W and w values that provide best decoding performance.

IV. CHECK NODE OPERATION

In MSA, the check node is expected to determine the product of the sign of incoming messages and also find the minimum of the magnitude of the input messages. In the proposed MMS2, the product of the sign of incoming messages are computed by using XOR operation (Sk) and the minimums are determined using AND operation (Mk). The check node output message (Ck) is obtained simply by concatenating the sign bit and the magnitude bit.

The message passing between the nodes continues till the parity check is satisfied or maximum iteration is reached. The message mapping in the variable node described above is shown in fig 3.3similar to that presented in. However, the proposed MMS2 algorithm eliminates the overhead of using scaling factor used, uses higher precision LLR for variable node operation and incorporates simple logic for check node operation [5]. These modifications lead to further improvement in performance and yet retain the reduced complexity of routing only 2-bit messages between the variable and check nodes in the LDPC decode



V. LOG-LIKELIHOOD RATIO

Researchers have used reliability information from soft-decision channel decoders for a variety of adaptation and control techniques. As nearly all wireless systems have some form of channel coding (often in the form of turbo, convolution, or low-density parity check codes), and that soft-decision decoders are almost always used to decode channel codes, the use of soft-output reliability information for the measurement and adjustment of radio resources is a subject of great significance in physical layer and cross-layer design. In fact [4], abundant literature exists on application of soft-output reliability information for the design or enhancement of Hybrid Automatic-Repeat-request (HARQ) procedures, Adaptive Modulation and Coding (AMC) algorithms, and Channel Quality Indication (CQI) methods.

The use of error rate estimates for AMC in GSM was proposed. It was found that soft reliability information lead to much more effective adaptation than SNR-based methods in the presence of SNR inaccuracy. In an unbiased estimator for PER and BER from log-likelihood ratio was analytically derived which removed dependency on the estimation of SNR and its uncertainty. In a HARQ scheme was described based on codeword reliability metrics derived from soft-output Viterbi decoders. These approaches pave the way for cross-layer design of efficient mobile ad hoc networks with substantially lower packet delay than distance-based routing algorithms. Most of these methods, however, focus on first mapping the Log-Likelihood Ratio (LLR) to a probability of bit error (BEP) (or estimating error rates from LLR), and then using simple time-averages of BEP for adaptation and control. In this paper, we take a different approach.

We treat the LLR and BEP as random variables, or stochastic time-series, and we focus on determining the probability distribution function (PDF) of each metric. We believe this has several advantages. First, sample averages tend to obscure important statistics of the time-series. The sample realizations of two different time-series may have the same mean, yet highly different characteristics. Second, and perhaps more importantly, under certain channel conditions, both LLR and BEP may turn out to be non-stationary and consequently not have constant mean values. The fact remains that while the mapping of LLR to BEP is well-known in the literature, to the best of our knowledge, no attempt has been made to derive the PDF of BEP by analysis. This is the principal contribution of this paper. We provide closed-form expressions for the PDF of both LLR and BEP, and we show simulation results to demonstrate the accuracy of our method. We also clarify the conditions under which LLR can be used

ISSN (ONLINE) : 2395-695X
ISSN (PRINT) : 2395-695X
Available online at www.ijarbest.com
International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST)
Vol. 1, Issue 8, November 2015

instead of BEP for possible adaptation and control of radio resources.

5.1 The LLR and its Probability Density Function

The Log-Likelihood ratio is defined as;

$$\lambda_i = \log \frac{P(b_i = 1 | y_i)}{P(b_i = 0 | y_i)}$$

Where b_i is the ith transmitted bit and y_i is the corresponding received signal sample. For binary phase-shift keying (BPSK) signaling in additive white Gaussian noise (AWGN) $y_i = u_i + n_i$, where $u_i = \pm\sqrt{E_b}$ is the transmitted signal, and ni is Gaussian noise of average power σn. We first derive the PDF of λ_i

VI.COMPARISON OF SUM- PRODUCT ALGORITHM AND MIN-SUM ALGORITHM

From the numerical results in **Table I** we observed that the thresholds corresponding to the sum-product algorithm are slightly smaller than those reported [7]. This is due to the particular discretization considered herein. Comparing the thresholds for the sum-product and the min-sum algorithm a difference as small as 0.27dB and as large as 0.98dB is observed. The inferior performance of the min-sum algorithm can be attributed to both, the sub optimality of the min-sum algorithm, as well as the specific quantization scheme used (*i.e.*, R = 6 bits and _ = 40/64). In order to quantify the performance loss due to discretization, several values for R were examined. Specifically, the results for the LDPC code with (dv, dc) = (3, 6) are shown in Table II. It can be observed that low resolution affects equally

THRESHOLDS FOR THE $(d_v, d_c) = (3, 6)$ LDPC CODE WITH SUM PRODUCT AND MIN SUM ALGORITHMS, AND DIFFERENT VALUES FOR R ($R = 6$ BITS).

d_v	d_c	rate	$\sigma_{\text{sum-prod}}$	$\sigma_{\text{min-sum}}$	diff. (dB)
3	6	0.5	0.8689	0.8177	0.53
4	8	0.5	0.8187	0.7455	0.81
5	10	0.5	0.7685	0.6957	0.86
3	5	0.4	0.9916	0.9154	0.69
4	6	1/3	0.9757	0.8716	0.98
3	4	0.25	1.2310	1.1020	0.96
4	10	0.6	0.7346	0.6776	0.70
3	9	2/3	0.7017	0.6751	0.34
3	12	0.75	0.6271	0.6079	0.27

Table 6.1 Difference in (db) of min sum and sum product

THRESHOLDS FOR THE $(d_v, d_c) = (3, 6)$ LDPC CODE WITH SUM PRODUCT AND MIN SUM ALGORITHMS, AND DIFFERENT VALUES FOR R.

R	$\sigma_{\text{sum-prod}}$	$\sigma_{\text{min-sum}}$	diff. (dB)
3	0.6492	0.6492	0.00
4	0.7646	0.7646	0.00
5	0.8488	0.8051	0.46
6	0.8689	0.8177	0.53
7	0.8777	0.8211	0.58
8	0.8802	0.8219	0.60
9	0.8806	0.8222	0.60

Table 6.2 Thresholds values differ for min sum and sum product algorithm.

The significant advantage of the sum-product over the minsum algorithm, which is evident from the results in Tables I, and II motivates the use of modifications to the min-sum algorithm such that the sum-product algorithm is approximated. Since the min-sum algorithm can be thought of as an approximation to the sum-product algorithm for high SNR, we seek more accurate approximations that are valid even for low SNR values.

II. TOOLS USED

- Target Device : Xilinx virtex 5,
- Target Platform : Xilinx ISE, Modelsim, Matlab

VIII.APPLICATIONS

- 10GBase-T Ethernet (802.3an) [8]
- G.hn/G.9960 (ITU-T Standard for networking over power lines, phone lines and coaxial cable)
- DVB-S2 (Digital video broadcasting)
- WiMAX (IEEE 802.16e standard for microwave communications)
- IEEE 802.11n (Wireless Local Area Network)
- WiGig Standard (Wireless Gigabit Communication over the unlicensed 60GHz Band)

IX.CONCLUSION

In this paper, a modified 2-bit Min-Sum algorithm is proposed to reduce the implementation complexity of LDPC decoders. It is shown that with a slight degradation in performance of about 0.3 dB at a BER of 10-5 compared to 3-bit Min-Sum, the proposed decoder leads to significant saving in hardware resource utilization and tremendous increase in average throughput. The performance of the proposed algorithm and its feasibility for practical systems are also verified by implementing the decoder suitable for WLAN. Therefore, the proposed LDPC decoder is a highly

ISSN (ONLINE) : 2395-695X
ISSN (PRINT) : 2395-695X
Available online at www.ijarbest.com
International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST)
Vol. 1, Issue 8, November 2015

attractive solution for applications requiring high performance.

Mat Lab was used to communicate with the FPGA using the serial port. LLRs were generated and sent to FPGA with appropriate control signals for decoding. The decoded data received via the same serial port was used to analyze the performance of the decoder. The BER performance and average iterations required by the decoder implemented on FPGA

X.ACKNOWLEDGMENT

I wish to acknowledge Mr. L.M.I. LEO JOSEPH, ASSOCIATE PROFESSOR, M.I.E.T.ENGINEERING COLLEGE for his advice on carrying out the performance simulations

XI.REFERENCES

1. R. Gallager, *Low-density parity-check codes.* IRE Transactions on Information Theory, 1962. 8(1): p. 21-28.
2. D.J.C. MacKay and R.M. Neal, *Near Shannon limit performance of low density parity check codes.* Electronics Letters, 1997. 33(6): p. 457-458.
3. Tetsuo Nozawa (2005) *LDPC Adopted for Use in Comms, Broadcasting, HDDs*. Nikkei Electronics Asia.
4. G.L.L. Nicolas Fau (2008) *LDPC (Low Density Parity Check) - A Better Coding Scheme for Wireless PHY Layers* Design and Reuse Industry Article.
5. S. Papaharalabos and P.T. Mathiopoulos, *Simplified sum product algorithm for decoding LDPC codes with optimal performance.* Electronics Letters, 2009. 45(2): p. 116-117.
6. N. Miladinovic and M.P.C. Fossorier, *Improved bit-flipping decoding of low-density parity-check codes.* IEEE Transactions on Information Theory, 2005. 51(4): p. 1594-1606.
7. Anastasopoulos. *A comparison between the sum-product and the min-sum iterative detection algorithms based on density evolution*. in *IEEE Global Telecommunications Conference*. 2001.
8. R. Zarubica, et al. *Efficient quantization schemes for LDPC decoders*. in *IEEE Military Communications Conference*. 2008.
9. Z. Cui and Z. Wang, *Improved low-complexity low-density parity-check decoding.* IET Communications, 2008. 2(8): p. 1061-1068.
10. R. Zarubica, S.G. Wilson, and E. Hall. *Multi-Gbps FPGA-Based Low Density Parity Check (LDPC) Decoder Design*. in *IEEE Global Telecommunications Conference*. 2007.
11. *IEEE 802.11n Wireless LAN Medium Access Control MAC and Physical Layer PHY specifications*. 2006, IEEE 802.11n-D1.0.