Designing LDPC system using prob domain decoder in Xilinx system generator

F S Hasan¹, R M Azeez² and M F Mosleh³
¹Department of Electrical Engineering, Al Mustansiriyah University, Baghdad, Iraq.
²,³Department of Computer Engineering Techniques, Electrical Engineering Technical College, Middle Technical University, Baghdad, Iraq.
E-mail: fadel_sahib@yahoo.com (Fadhil S Hasan).

Abstract. Low-Density Parity-Check (LDPC) code represents a type of Forward Error Correction (FEC) codes, it is a very perfect code with the ability to correct the errors and it has a performance in term of Bit Error Rate (BER) close to Shannon limits. As this code has many features like reliability, flexibility in implementation and a facility of adaptation. So it takes place in the current area of 4G and even 5G network communications. In this paper, an efficient and optimum design for LDPC system using prob domain decoder is implemented using a Xilinx system generator to evaluate its performance in terms of BER, its complexity and the time needed. FPGA device Kintex7 (XC7K325T-2FFG900C) is used for the implementation of the proposed model. The results show that by increasing the Signal to Noise Ratio (SNR), the values of the BER will be improved significantly.

1. Introduction

During the Ph.D. thesis, Gallager [1] suggested LDPC codes for the first time but at that time almost for thirty years, this work done by Gallager was ignored until MacKay [2] reintroduced these codes in 1996. These codes represent a type of linear block codes so they use a generator matrix denoted as G in the encoder and in the decoder they use a parity check matrix (PCM) H [3] with m rows and n columns. The rows represent the Check nodes (CNs) and the columns represent the Variable Nodes (VNs) [4], this PCM represents a method to represent these codes. Another method that can be used to represent them is by using the Tanner graph [5, 6].

There are two main types of decoding algorithms for LDPC codes [7]. The first type is the hard decision decoding algorithms and the second type is the soft decision decoding algorithms [8]. The soft decision algorithms, can correct more errors and for that reason, they are more robust and they have higher performance. This makes these algorithms more complex than hard decision algorithms [9].

Many researchers go to experiment with the hardware applications for LDPC, in order to adopt its facilities for modern communications. Like [10] implemented a half-parallel decoder using extended Min-Sum Algorithm (MSA) based FPGA design for Quasi-Cyclic (QC) LDPC codes over Galois Field (GF). The results showed that the maximum frequency was 131.411MHz and the throughput was 50Mb/s for this decoder. Also, [11] used the Log domain algorithm for decoding the code words, a regular LDPC code having a code rate of ½ was employed on FPGA using the Xilinx Virtex device, the throughput was 82 Mbps for this decoder, and the complete design was done in VHDL using the Xilinx ISE program. In addition, [12] designed and implemented a low complexity LDPC decoder using an improved 2-bit MSA. The results of the simulations showed that the suggested decoding algorithm had an enhancement in BER and needed fewer iterations for decoding than the original 2-bit MSA. Finally, [13] designed and implemented the parallel construction of a reduced complication for decoding LDPC
codes for high data rate applications, the chosen code was a regular code (3, 4). The results illustrated that the data rate was high and the latency was low with much-reduced complication. The BER against SNR may be more enhanced by increasing the size of the code by saving the same principle of parallelism. In this paper, an efficient design of the LDPC system using prob domain algorithm will be implemented using Xilinx System Generator (XSG) which is a new and an efficient technique to design many systems such as in [14-15]. The LDPC system will be implemented using XSG with the software tools Xilinx Vivado 2017.4 and Xilinx Kintex7 (XC7K325T-2FFG900C). The BER results between XSG and Matlab program are compared. The rest of this paper is organized as the following; the second section includes the theory of prob domain algorithm used in the decoder, the third section includes the XSG design of the proposed system. In the fourth and fifth sections, the XSG waveforms and synthesis reports are presented respectively. Finally, the conclusion is presented in the sixth section.

2. LDPC System Using Prob Domain Decoder

Figure 1 [16] illustrates the block diagram of the LDPC system using the prob-domain decoder. The matrix method is used to implement the encoder with the number of rows \( m \) is 10 and the number of columns \( n \) is 20.

![Figure 1. LDPC System model based on prob decoder.](image)

2.1. Prob domain

This algorithm is considered a soft decision message-passing algorithm. It is similar to the bit flipping algorithm but the messages representing each decision with its probabilities [17]. The steps of the prob algorithm are illustrated below:

1. Initialization:
   
   \[
   P_n(1) = \frac{1}{1 + e^{-2r_iN_0}} \tag{1}
   
   P_n(0) = 1 - P_n(1) \tag{2}
   
   \]

   Where \( P_n(1) \) and \( P_n(0) \) are the posterior probabilities of ones and zeros, they can be computed depending on \( r_i \) which is the received signal, and \( N_0 \) is the noise variance. \( q_{ij_mn}(1) = P_n(1) \) and \( q_{ij_mn}(0) = P_n(0) \). \( q_{ij} \) is a message sent by the variable node \( n \) to the check node \( m \). Every message always contains the pair \( q_{ij_mn}(0) \) and \( q_{ij_mn}(1) \) which stands for the amount of belief that \( r_i \) is zero or one.

2. Horizontal Step (CNs Processing):

   \[
   dr_{ji} = \prod_{n \in B_m, n' \neq n} (q_{ij_mn'}(0) - q_{ij_mn}(1)) \tag{3}
   
   r_{ji_mn}(0) = \frac{1}{2}(1 + dr_{ji}) \tag{4}
   
   r_{ji_mn}(1) = \frac{1}{2}(1 - dr_{ji}) \tag{5}
   
   \]

   Where \( dr_{ji} \) can be computed by taking the product of subtraction \( q_{ij_mn}(1) \) and \( q_{ij_mn}(0) \) excluding the bit \( n \) according to [18]. \( r_{ji_mn}(0) \) and \( r_{ji_mn}(1) \) represent the probability that check node \( m \) is satisfied with either one or zero value from the VN.
3. Vertical Step (VNs processing): Computing the messages $Prod_{mn}(0)$ and $Prod_{mn}(1)$ sent from CNs to VNs except the message from CN $m$.

$$Prod_{mn}(0) = \prod_{i \in A_n, m \neq m} rj_{i'n}(0)$$  \hspace{1cm} (6)

$$Prod_{mn}(1) = \prod_{i \in A_n, m \neq m} rj_{i'n}(1)$$  \hspace{1cm} (7)

$k_{mn}(0)$ and $k_{mn}(1)$ are scales factors applied to each element. Updating these two scales are done by:

$$k_{mn}(0) = P_n(0) \times Prod_{mn}(0)$$  \hspace{1cm} (8)

And

$$k_{mn}(1) = P_n(1) \times Prod_{mn}(1)$$  \hspace{1cm} (9)

Updating the values of $qi_{mn}(0)$ and $qi_{mn}(1)$:

$$qi_{mn}(0) = \frac{k_{mn}(0)}{k_{mn}(0) + k_{mn}(1)}$$  \hspace{1cm} (10)

And

$$qi_{mn}(1) = \frac{k_{mn}(1)}{k_{mn}(0) + k_{mn}(1)}$$  \hspace{1cm} (11)

4. Computing the A Posteriori Probabilities (APP) likelihood ratios $Qi_{n}(0)$ and $Qi_{n}(1)$ for each bit and checking the stopping condition: $ki_{n}(0)$ and $ki_{n}(1)$ are scales factors, and they can be updated as shown below

$$ki_{n}(0) = P_n(0) \times \prod_{m=1}^{m} rj_{i'm}(0)$$  \hspace{1cm} (12)

$$ki_{n}(1) = P_n(1) \times \prod_{m=1}^{m} rj_{i'm}(1)$$  \hspace{1cm} (13)

$$Qi_{n}(0) = \frac{ki_{n}(0)}{ki_{n}(0) + ki_{n}(1)}$$  \hspace{1cm} (14)

$$Qi_{n}(1) = \frac{ki_{n}(1)}{ki_{n}(0) + ki_{n}(1)}$$  \hspace{1cm} (15)

The stopping condition is computed according to [18]:

$$v_{hat} = \begin{cases} 1, & Qi_{n}(1) > Qi_{n}(0) \\ 0, & Qi_{n}(1) \leq Qi_{n}(0) \end{cases}$$  \hspace{1cm} (16)

3. Implementation of Prob Decoder based XSG

The LDPC system in Figure 1 is implemented using the XSG tools as shown in Figure 2.
Figure 2. XSG block diagram of LDPC system using prob decoder.

Details for all the blocks in the system will be illustrated in the following subsection with focusing on the prob decoder.

3.1. Transmitter Side of LDPC system

This side contains the blocks of Bernoulli binary generator, serial to parallel, LDPC encoder and finally, the modulation block as shown in Figure 3. The first block in the transmitter side is the Bernoulli binary generator. This block will produce a binary signal that will be either 0 or 1 value. This block parameter is set to 1 for the sample time and the format of the output data is Boolean. The gateway-in block is XSG block used to convert the data to the unsigned format with Word Length (WL) is 1 and Fraction Length (FL) is 0.

The output from the gateway-in block will then enter the Serial to Parallel (S2P) block, the number of bits is 1 and latency is 1 in this block. The output will be symbols of 10 bits that will be converted to parallel bits by using ten slice blocks to extract ten bits by selecting a particular sort from every sample in the input. Bits number in each one of these slices is 1 and the output data is unsigned with WL is 2 and FL is 0. The output data from the (S2P) block will then enter the encoder block to encode each message using the equations of the $H$ matrix, the encoder consists of Xor gates where each one of these gates corresponds to the check equations of the rows in the PCM and concat block as illustrated in Figure 4.

In the modulation block, there are three blocks that are parallel to serial block, Rom block that will be used for the mapping process and delay block to mask the bits as shown in Figure 5.

Figure 3. Transmitter side of the LDPC system in XSG.
3.2. Channel block
This block consists of the AWGN generator which is a system generator block set that can be used to produce random signals with Gaussian distribution of zero mean and unity variance with WL is 18 bits and FL is 16 bits. The output from the AWGN generator will be multiplied by the square root of $N_0$ then the result from this multiplication will be added to each bit of the transmitted data to represent the noise for different SNR as shown in Figure 6.
3.3. Receiver Side of LDPC System
This section contains initialization block, S2P block, prob decoder, down sample and P2S block. The first block will calculate the prior probabilities for $P_n(1)$ and $P_n(0)$. In order to reduce the complexity of implementing the exponential function on XSG. Approximate values are used that can give very rounded values for each SNR. The value of $P_n(1)$ will be 0.9998 if $\tau_i$ is greater than zero otherwise it will be 0.0035 and the values of $P_n(0)$ will be computed according to Eq.2. This process will be applied to 20 bits serially by taking the average of probabilities that represent values of zeros and ones. Two blocks of serial to parallel are used to convert the values of $P_n(1)$ and $P_n(0)$ from serial samples to parallel samples, where each sample is with format WL is 18 bits and FL is 16 bits. Figure 7 illustrates the receiver side of the LDPC system.

The decoder block consists of two steps horizontal and vertical step as shown in Figures 8 and 9 respectively, the output will be symbols of 10 bits which will represent the information message.
In the horizontal step, the computations are made depending on the number of ones in every column of the ten rows in the parity check matrix according to Eq.3, 4 and 5 respectively in step 2.

In the vertical step, the computations are made depending on the number of ones in every row of the twenty columns in the PCM according to the equations in step 3. Also, a decision for each bit is made based on the equations in step 4. In the final block, the down sample block is used to decrease the rate of the received signals with sample rate is 1 and latency is 20. After this stage, these bits will be converted from parallel to serial by using concat block and Parallel to Serial (P2S) block as shown in Figure 10.
4. XSG Results and Discussion
The XSG simulation waveforms for each block of the system model is plotted using Matlab program. Figure 11 illustrates the XSG waveform of the S2P block. The waveforms of LDPC encoder and the modulation block are illustrated in Figures 12 and 13 respectively. Figures 14 shows the waveform of the AWGN channel. The waveforms of the initialization process are illustrated in Figure 15. Figure 16 illustrates the XSG waveform of decoding and down sample for the second bit. Figure 17 shows the XSG waveform of the concat block output. The comparison between XSG waveforms of the transmitter and receiver is shown in Figure 18.

Figure 10. Down sample, concat and P2S blocks.

Figure 11. S2P waveform in XSG.
Figure 12. The waveform of LDPC encoder in XSG.

Figure 13. Modulation block waveforms in XSG.
Figure 14. The waveform of AWGN channel in XSG when the SNR is 6.

Figure 15. Initialization process waveforms in XSG at 6 SNR.
Figure 16. Decoding and down sample waveforms for the second bit.

Figure 17. Concat block waveform when the SNR is 6.
5. Synthesis Results
Table 1 and Figure 19 illustrates the BER in Matlab/Simulation and in FPGA.

Table 1. Matlab/Simulation BER and FPGA BER.

| SNR(dB) | Matlab BER | FPGA BER |
|---------|------------|----------|
| 0       | 0.1571     | 0.1485   |
| 1       | 0.1286     | 0.1287   |
| 2       | 0.1143     | 0.09901  |
| 3       | 0.0714     | 0.08911  |
| 4       | 0.0429     | 0.05941  |
| 5       | 0.0429     | 0.0297   |
| 6       | 0          | 0.009901 |
| 7       | 0          | 0.009901 |
| 8       | 0          | 0        |

Figure 18. The transmitted signal and the received signal when the SNR is 6.

Figure 19. Comparing the BER in Matlab/Simulation and in FPGA.
FPGA device Kintex7 (XC7K325T-2FFG900C) was chosen to build the system using Vivado 2017.4 program as shown in table below. The minimum period is 52.571ns and the maximum frequency is 19.022MHz. In addition, the proposed LDPC system based on prob domain decoder has been tested and downloaded successfully on the FPGA device kintex7 (XC7K325T-2FFG900C) as shown in Figure 20.

Table 2. Utilization summary of LDPC system using Kintex7 (XC7K325T-2FFG900C).

| Resource       | Utilization | Available | Utilization Percentage |
|----------------|-------------|-----------|------------------------|
| LUT            | 4380        | 203800    | 2.15                   |
| LUTRAM         | 252         | 64000     | 0.40                   |
| FF             | 2157        | 407600    | 0.53                   |
| BRAM           | 4.50        | 445       | 1.01                   |
| DSP            | 527         | 840       | 62.74                  |
| Number of bonded IOBs | 4  | 500       | 0.80                   |
| BUFG           | 1           | 32        | 3.13                   |

Figure 20. Test the system using kintex7 (XC7K325T-2FFG900C).

6. Conclusion
In this paper, a communication system based LDPC code with soft decision decoding is designed and implemented successfully using Xilinx system generator. The software tools are Vivado 2017.4 and FPGA device Kintex7 (XC7K325T-2FFG900C) are used because it has a large amount of DSP slices which are suitable to build this design. The obtained results from the system proved that the system works correctly with results very close to results obtained in Matlab/Simulation with reduced design for
prob decoder and with minimum values of BER. For SNR values 0, 1, 2 …8, the corresponding results of BER based FPGA were 0.1485, 0.1287, 0.09901, 0.0891, 0.05941, 0.0297, 0.009901, 0.009901 and 0 respectively, so it can be seen that by increasing the SNR values the BER will be improved. These results confirm that the theoretical results based on pure simulation are very close to those results of FPGA implementation as indicated in Table 1. That means the proposed system can be implemented successfully in modern communications.

References
[1] Gallager R G 1963 Low Density Parity Check Codes Number 21 in Research monograph series. MI Press, Cambridge.
[2] Mackay D J C 1999 Good Error-Correcting Codes Based on Very Sparse Matrices *IEEE Trans. Info. Theory* 45(2) 399-431.
[3] Sonia Er, Gupta Er S 2015 Hard Decision and Soft Decision Decoding Algorithms for LDPC And QC-LDPC Codes *International Journal of Computer Science and Mobile Computing* 4(9)182 – 191.
[4] Lee M H, Han J H and Sunwoo M H 2008 New Simplified Sum-Product Algorithm for Low Complexity LDPC Decoding 2008 *IEEE Workshop on Signal Processing Systems Conference* 61-66.
[5] Mosleh M F, 2011, Evaluation of Low Density Parity Check Codes Over Various Channel Types, *Engineering and Technology Journal*, 29(5), 961-971.
[6] Khalifa O O, et al. 2008 Performance Evaluation of Low Density Parity Check Codes *International Journal of Electrical and Electronics Engineering* 2(6).
[7] Bhavsar N P and Vala B 2014 Design of Hard and Soft Decision Decoding Algorithms of LDPC, *International Journal of Computer Applications*, 90(16).
[8] Mosleh M F, Hasan F S, Azeez R M 2019 Comparison between Different Decoding Algorithms for Low Density Parity Check 4th *SICN-2019* Conference.
[9] Babu C, Prathyusha S and Sree V U 2014 Hard Decision and Soft Decision Decoding Algorithms of LDPC and comparison of LDPC with Turbo Codes, RS Codes and BCH Codes *9th IRF International Conference*, India.
[10] Sun Y 2009 FPGA Implementation of Non binary Quasi-Cyclic LDPC Decoder Based on EMS Algorithm *International Conference on Communications, Circuits and Systems* (ICCCAS) *IEEE*.
[11] Vijayakumar S 2009 FPGA Implementation of Low Density Parity Check Codes Decoder MSc. Thesis University of North Texas.
[12] Chandrasetty V A and Aziz S M 2010 FPGA Implementation of High Performance LDPC Decoder using Modified 2-bit Min-Sum Algorithm *Second International Conference on Computer Research and Development*.
[13] Boudaoud A, El Haroussi M and Abdelmoumin E 2017 VHDL Design and FPGA Implementation of LDPC Decoder for High Data Rate *International Journal of Advanced Computer Science and Applications* 8(4).
[14] Mosleh M F, Abid M F and Al-Sadoon M 2018 Implementation of Turbo Code Based Xilinx System Generator 9th *International EAI Conference* Broadnets Faro Portugal.
[15] Lee D 2004 Hardware Designs for Function Evaluation and LDPC Coding Ph.D. Thesis Imperial College London.
[16] Reddy B S and Rao V S R 2017 FPGA Implementation of LDPC Encoder and Decoder using Bit Flipping Algorithm 6(9) 1683–1690.
[17] Porcello J C 2014 Designing and Implementing Low Density Parity Check (LDPC) Decoders using FPGAs *Aerospace Conference* *IEEE*.
[18] Marotkar D, Kapur V, Bramhe S 2015 Decoding Techniques of LDPC Codes *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*, 3(3).