Reduce the Resources Utilization of LDPC Decoder Based on Min-Sum Decoder for Spread Spectrum Applications

Mahmood F. Mosleh¹, Fadhil S. Hasan², Aya H. Abdulhameed³

¹drmahmoodfarhan@gmail.com, fadel_sahib@uomustansiriyah.edu.iq, aya.hazim@mtu.edu.iq
²Middle Technical University, Department of Computer Engineering Techniques, Baghdad, Iraq
³Department of Electrical Engineering, Al Mustansiriyah University, Baghdad, Iraq

Abstract. Communications with spread spectrum (SS) have attracted attention because of their channel attenuation immunity and low potential for interception. Aside from some extra features such as basic transceiver structures, the analogy alternative to digital SS systems would be chaotic communication. In this brief, Differential Chaos Shift Keying (DCSK) systems, non-periodic and random characteristics among chaos carriers and their interaction with soft data are designed based on low-density parity-check (LDPC) codes. Because of simple structure and magnificent error correction capability. Using the development kit Xilinx kintex7 FPGA, we investigate the DCSK communication system's hardware efficiency and resource requirement tendencies based on the LDPC decoding algorithm Min-Sum over AWGN channel. The results demonstrate that the proposed system model has major improvements in Bit Error Rate (BER) performance and the real-time process. The Min-Sum decoder has comparatively fewer FPGA resources than other decoders for soft decisions. The implemented system achieves a coding gain of 10-4 BER efficiency with 5.6 dB associate Eb/No.

1. Introduction
Chaotic signals, being wideband, are well suited for spread-spectrum communications [1]. Chaos shift keying (CSK) and differential CSK (DCSK) were the most commonly studied among the digital schemes proposed [2]. For a coherent implementation, where synchronized replicas of the chaotic basis functions are required at the receiver, the former was originally suggested [3]. This requirement, however, has not been proved realistic yet. On the other hand, the DCSK scheme represents a more stable non-coherent scheme [4] in which precise knowledge of the receiver’s chaotic base functions is not needed. Instead, the two signal samples are correlated and a threshold is compared with the correlation product. The efficiency of digital communication systems based on chaos has been extensively studied in an additive white Gaussian noise (AWGN) environment [5,6,7]. Experts have seen rapid developments in the field of wireless communications in recent years. The aim of the next generation's wireless system is to send a higher data rate and provide a wider area of coverage [8]. However, due to regulatory and system power consumption factors, radio transmissions need to be reduced to a minimum. To solve this problem, it is feasible to use the Forward-error-correction (FEC) system [9]. The coding gain provided by FEC Low-Density Parity Check (LDPC) codes allows the system, at the cost of coding bandwidth overhead, to transmit data at the same error performance with lower signal power [10]. LDPC codes can be classified into two categories, hard decision and soft decision. Those algorithms are bit flipping algorithms, a sum-product algorithm (Prob. Domain, Log Domain and Min-Sum) decoding algorithms [11], etc. Many studies are concerned with a digital communication system that includes LDPC code and DCSK, such as [12],...
where LDPC codes are improved with the FM-DCSK system proposed. The researchers used chaotic Walsh transform codes in [13] and a convolutional code and LDPC code. The code category is therefore sponsored by the wireless MIMO communication system. A specification of alternative receivers based on the M- Array DCSK is also provided in [14].

Our purpose here is to design a flexible architecture and evaluate its performance for the DCSK system combine with the soft-decision Min-Sum (MS) LDPC code algorithm in the AWGN channel by using FPGA. The Hardware Co-Simulation results will be compared with the simulation results obtained earlier for the system.

2. DCSK with LDPC encoded communication system

The diagram of the proposed system shown in figure1 included a binary data source encoded by LDPC block by a code rate of 0.5. The codeword is fed to the DCSK modulator which uses a chaotic carrier to spread the digital signal over a large frequency band to achieve a modulated signal with a spreading factor value equal to 16. The modulated signal will be suggested to AWGN noise, which is often used for simplicity as experimental applications. At the receiver side, non-coherent demodulation will be applied to recover the received codeword which will be decoded to regenerate the original data.

![Figure 1. Block diagram of the system model.](image)

3. Hardware communication system design

As mentioned in section 2 the system consists of three main parts: transmitter, channel and receiver. A transmitter section is established by the subsequent, Logistic-map as input chaotic data generator, LDPC encoder by using the systematic form of H matrix and computing the parity check equation for each row of this matrix and DCSK modulator. The receiver is established by the subsequent AWGN noise, DCSK de-modulator and LDPC decoder with Min-Sum decoder. The system model will be implemented using Xilinx System Generator (SG) as shown in figure 2, each block is designed with specific parameters to match the overall system implementation as will be explained in detail individually.
3.1. LDPC encoder

The message word generated by Bernoulli random binary number with a length of k=10 is fed to the LDPC block via gateway block to achieve a code word at the output of such block with a length of n=20. Inside the LDPC encoder block shown in figure 3, there is a serial to parallel block which is used to convert a group of samples presented serially at the input into single samples presented at the output see figure 4.

\[ S_i = [c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{16}, c_{17}, c_{18}, c_{19}, c_{20}]^T \]  

(1)

The parity bits are introduced using eqs.2 & 3 which represent the first and second parity bits corresponding to the first and second row of the matrix of eq.4 respectively. The rest of the parity is produced in the same methods represented by the rest of such matrix rows [16,17].
\[ c_{11} = c_3 \oplus c_6 \oplus c_9 \]  
\[ c_{12} = c_4 \oplus c_6 \oplus c_7 \oplus c_8 \]  

\[
 H = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 
\end{bmatrix}
\]  

The hardware implementation of such equations is shown in figure 5. These equations are applied using 10 XOR gates to produce 10 parity check bits depending on the input message.

\[ \text{Figure 5. The hardware of Encoder.} \]

Now, the 20-bits code-word is fed to concat block which concatenates two or more inputs bits to get a symbol of these bits in the output. Finally, a parallel to serial block drives the code word as an output block, so each sample presented in the input of this block becomes several samples presented serially in the output.
3.2. DCSK modulator

The code-word produces by LDPC is fed to the DCSK modulator which is designed by Xilinx SG. In the DCSK modulator, every bit $S_i$ is represented by two sets of chaotic signal samples with a period of $\beta$, where $2\beta$ refers to the spreading factor with an integer number. The first and second sets represented the reference segment and data segment, respectively. To transmit +1, data segment samples are equal to the reference segment, and an inverted version of a reference segment is used to transmit bit -1. During the $i^{th}$ bit period, the output of the transmitter $e_k$ is [18,19].

$$e_k = \begin{cases} X_k & \text{For } 1 < K \leq \beta \\ S_iX_{K-\beta} & \text{For } \beta < K \leq 2\beta \end{cases} \quad (5)$$

Figure 6. Shows the Xilinx SG block of DCSK modulation system with $\beta$ equal to 8.

All blocks of the chaos generator and mapping are shown in Figures 7 and 8 respectively. Chebyshev Polynomial Function (CPF) of order two as in eq.6 is chosen as a chaotic generator with the initial condition value of 0.1 for simply implementation by using multiplexers, adder and multiplier blocks:

$$X_{K+1} = 1 - 2X_K^2 \quad (6)$$

Figure 7. The Chaos-Generator block.

Mapping block consists of ROM to indicate either \{0\} input or \{1\} input corresponding to the initial value vector of ROM [-1 1]. Delay has been linked to enabling the pin of the ROM block to mask all the serial bits till they are ready for the map process.
3.3. Channel
When the modulated signal is transmitted through a channel, it will be corrupted by noise in the channel represented by Additive White Gaussian Noise (AWGN), $n_k$. Such noise is statistically random radio noise characterized by a wide frequency range $[20]$. The hardware implemented of such a channel is illustrated in figure 9 with seed value 512.

![AWGN Channel block.](image)

3.4. DCSK demodulation
The received signal $r_k$ passes to an S2P block to convert the 16 serial to parallel samples in order to de-spread into one bit. The details of the Xilinx SG S2P block which consists of 16 latches and its corresponding delay blocks are illustrated in figure 10.

![16-samples S2P blocks at the demodulator.](image)

The 16 output samples of S2P are connected with a correlator, wherever the reference samples and also the corresponding information samples are correlated by using multipliers between them. The output of the $i^{th}$ correlator is the $i^{th}$ decision variable, $D_i$ [21].

$$D_i = \sum_{k=1}^{\beta} r_k r_{k+\beta}$$ (7)

The $i^{th}$ bit is demodulated by computing the sign of the final correlator output using a zero threshold. The Xilinx SG DCSK demodulator is shown in figure 11.
Figure 11. Xilinx SG DCSK Demodulator block.

3.5. Min-Sum decoder algorithm

The MS algorithm is an iterative message-passing algorithm at $ith$ iteration, first, the variable-to-check messages that represent an estimate of the posteriori log-likelihood ratio (LLR) are computed and sent to the corresponding neighbor check nodes. Second, the check-to-variable messages are computed and sent back to the neighbor variable nodes to obtain 10-bit symbols representing the original signal after the decoding process. The demodulated message is firstly fed to the S2P converter block to match each other within SG blocks. The S2P will be used to convert the 20-stream of bits to parallel bits to be initialized. In general, the mathematical model of the Min-sum decoding algorithm is as follows:

Step1: **Initialization**: In MS decoder, the Log-Likelihood Ratio (LLR) of prior (which represents the receiving messages from the channel) and posterior (which represents the medial messages moved between CNs and VNs) probability is used. So the prior messages sent from BN n to the CN m represent the LLR [22].

$$L_{ci} = -r_x$$  \hspace{1cm} (8)

$$\alpha_{m,n} = \text{sign} (L_{ci})$$  \hspace{1cm} (9)

$$\beta_{m,n} = |L_{ci}|$$  \hspace{1cm} (10)

This process is applied to 20 received bits $r_x$ serially. Then, the absolute value and sign value (will calculate the sign value whether it is positive or negative or equal to zero) will be calculated for each bit. The implementation of the initialization block is done by using Mult,
Mux and Relational Xilinx SG block. The process of initialization block and the sign block will be declared in figures 12 and 13, respectively.

![Figure 12. Initialization process (Lic).](image1)

![Figure 13. Details of Sign block.](image2)

Step 2: **Check node update (Horizontal Step):** The computations of the horizontal step are created depend upon the amount of 1s per column of ten rows throughout the H matrix. Computing the extrinsic messages for each set of bits connected to CN m by excluding the bit n’ the equation below will be used [23, 24].

\[ A_{lm,n} = \min_{n \in \mathcal{B}_m, n \neq n'} (beta_{ij,m,n'}) \]  

Get products of \( \alpha_{ij,n,m} \) excluding the bit n:

\[ Pr_{m,n} = \prod_{n \in \mathcal{B}_m, n \neq n'} \alpha_{ij,m,n'} \]  

Finally, compute \( F_{ji,m,n} \)

\[ Pr_{m,n} = \prod_{n \in \mathcal{B}_m, n \neq n'} \alpha_{ij,m,n'} \]  

For example, the Xilinx SG design for the first row will be implemented by using (Multi, Mux, and Relational Xilinx blocks) block to manage the value of the variable node according to eqs. 11 & 12 & 13 as illustrated in figure 14. The details for each one of the blocks in figure 15 can be declared in figures 16, 17 and 18.
Figure 14. Horizontal step process of the first row.

Figure 15. Details of Mi block.

Figure 16. Details of Mi 1,13,16 and 19 blocks.
Step 3: **Variable node update (Vertical Step):** this step’s calculations are based on the number of 1s in all rows of the twenty columns in H matrix throughout the equation:

\[
L_{Qi} = L_{ci} + \sum_{m \in A_n} r_{j_{m,n}} \tag{14}
\]

\[
v_{\text{hat}} = \begin{cases} 
1, & L_{Qi} < 0 \\
0, & L_{Qi} \geq 0 
\end{cases} \tag{15}
\]

Where \( L_{Qi} \) represent the collective log-likelihood ratio for n’th digit. Also, a \( L_{Qi} \) decision is made according to eq.15 for each bit. The design model is implemented using Xilinx SG blocks (Add-Sub, Mux, and Relational Xilinx blocks) to manage the value of the variable node. The details for the Xilinx SG block of the twelve columns are described in figure 19.

**Figure 17.** Details of Pr block.

**Figure 18.** Details of Pr 1,13,16 and 19 blocks.

**Figure 19.** Vertical step blocks of column 12.

3.6. **Downsample**
Finally, the fully flexible LDPC decoder design output bits were connected to down-sample block from Xilinx SG blocks with a sampling rate equal to 320 which is used to control the output rate signal at the receiver. These bits from down-sample blocks' output will be converted from parallel bit to stream bits after passing through concat and P2S blocks respectively, to get the information data, as seen in figure 20. Once the Simulink models are developed for the proposed system, the VHDL code can be generated automatically using the SG block.

4. SIMULATION RESULTS
The performances of codes obtained from the proposed system were measured by BER as a function of SNR. To complete the simulation with an optimum parameter was the No. of one in each column \(w_c = 3\), No. of iteration \(itr\) = 6, frame length \(F\) = 100, code rate \(1/2\) and spreading factor \(2\beta\) value used are equal to 8, 16, and 32. The results illustrated in figures 21 are clear-out the comparisons between these communication systems with and without LDPC code in terms of SNR form with standard accepted BER about \(10^{-3}\).

5. FPGA SYNTHESIS RESULTS
Table 1 shows the hardware resources devices utilization required for Min-Sum decoder algorithms.

![Figure 20. Down Samples Xilinx SG block.](image)

![Figure 21. Performance comparisons, with spearing factor = 8, 16 and 32.](image)

Table 1. Min-Sum resources device summary using Kintex 7.
Device Summary

| Slice Logic Utilization | Used   | Available |
|-------------------------|--------|-----------|
| Number of Slice Registers | 3,322  | 407,600   |
| Number of slice register sites lost to control set restrictions | 1,984  | 50,950    |
| Number of Slice LUTs   | 4,927  | 203,800   |
| Number used as logic   | 3,984  | 203,800   |
| Number used as Memory  | 943    | 64,000    |
| Number of LUT Flip Flop pairs used | 1,259  | 203,800   |
| Number of bonded IOBs  | 3      | 500       |
| Number of RAMB8BWERs   | 4.5    | 445       |
| Number of BUFG/BUFGMUXs | 1      | 32        |
| Number of DSP48A1s     | 189    | 840       |

To compare the amount of complexity for the overall system in the four types of soft decision decoder, Table 2 summarized the resource device utilization for (Prob. Domain, Log Domain and Min-Sum). From all the comparisons it is easy to conclude that the optimum decoding algorithms in the term of complexity and performance are the Min-Sum algorithms due to their BER performance and suitable resource utilization.

Table 2. Resource device comparison.

| Devices utilization | Algorithms |
|---------------------|------------|
|                     | Prob. Domain | Log Domain | Min-Sum |
| Slice regester      | 3,856       | 3,025      | 3,322   |
| LUT slice           | 9.157       | 83,417     | 4,927   |
| Memory              | 1,196       | 808        | 943     |
| DSP                 | 539         | 227        | 189     |
| Max-Delay (ns)      | 9,558       | 6,612      | 7,561   |

Figure 22 shows the Xilinx SG simulation test result among the Bernoulli binary transmitted signal and the received signal where the Min-Sum decoder record the original signal with 960ns delay.

Figure 22. The simulated result of data recovery signal processing of Min-Sum decoder algorithm.
The hardware Co-Simulation results of the user data information and recovered information for the proposed algorithm are shown in figures 23 where there is a delay due to the operation of the extraction process and there are some Xilinx blocks causes delay.

![Figure 23. Min-Sum hardware Co-Simulation implementation result.](image)

6. CONCLUSION

In this work, the DCSK communication system has been developed with LDPC codes by using Min-Sum decoding algorithm, and implementation has been done on the Kintex 7 FPGA development kit using Xilinx SG tools. The simulation results show that Min-Sum achieved 5.6, 5, and 4.5 dB improvement gain for the corresponding Spreading factor of 8, 16, and 32, respectively. The system performance proves that increasing the value of the spreading factor leads to a decrease in the performance. Subsequently, better results could be obtained with the spreading factor ($2^\beta$) equal to 8. With FPGA, the results show that the consumption of the Min-Sum algorithm was 1% of the slice registers, 23% of DSP and 2% of LUT. In contrast, other soft-decision decoders algorithm (Prob. Domain, Log Domain and Min-Sum) has high complexity in terms of the resource utilization which means that the optimum decoding algorithms are the Min-Sum algorithm due to its BER performance and suitable resource utilization. The hardware implementation confirms that the proposed system is suitable for future communications, especially in real-time applications.

7. References

[1] Anantharaman, R., Kwadiki, K., & Kerehalli Shankar Rao, V. P. (2019). Hardware implementation analysis of min-sum decoders. Advances in Electrical and Electronic Engineering, 17(2), 179–186. https://doi.org/10.15598/aeee.v17i2.3042

[2] Anbuselvi, M., & Salivahanan, S. (2013). A complexity reduction method for extended min-sum based nonbinary LDPC decoder. International Journal of Engineering and Technology, 5(6), 4605–4612.

[3] Angarita, F., Valls, J., Almenar, V., & Torres, V. (2014). Reduced-complexity min-sum algorithm for decoding ldpc codes with low error-floor. IEEE Transactions on Circuits and Systems I: Regular Papers, 61(7), 2150–2158. https://doi.org/10.1109/TCSI.2014.2304660

[4] Bassam, N. Al, & Zayegh, A. (2018). Chaos-Based Communication Systems. Chaos Theory. https://doi.org/10.5772/intechopen.71447

[5] Chandrasetty, V. A., & Aziz, S. M. (2011). Fpga implementation of a LDPC decoder using a reduced complexity message passing algorithm. Journal of Networks, 6(1), 36–45. https://doi.org/10.4304/jnw.6.1.36-45

[6] Chandrasetty, V. A., & Aziz, S. M. (2012). An area efficient LDPC decoder using a reduced complexity min-sum algorithm. Integration, the VLSI Journal, 45(2), 141–148. https://doi.org/10.1016/j.vlsi.2011.08.002

[7] Choi, G., Park, K. Bin, & Chung, K. S. (2018). Optimization of FPGA-based LDPC decoder
using high-level synthesis. ACM International Conference Proceeding Series, 256–259. https://doi.org/10.1145/3290420.3290441

[8] Gallager, R. G. (2001). Claude E. Shannon: A retrospective on his life, work, and impact. IEEE Transactions on Information Theory, 47(7), 2681–2695. https://doi.org/10.1109/18.959253

[9] Hasan, F. S. (2017). FPGA based 2× 2 MMSE MIMO-OFDM system using Xilinx system generator. Journal of Engineering and Sustainable Development, 21(1), 152–176.

[10] Kaddoum, G., Soujieri, E., Arcila, C., & Eshteiwi, K. (2015). I-DCSK: An Improved Noncoherent Communication System Architecture. IEEE Transactions on Circuits and Systems II: Express Briefs, 62(9), 901–905. https://doi.org/10.1109/TCSII.2015.2435831

[11] Lau, F. C. M., Tse, C. K., Ye, M., & Hau, S. F. (2004). Coexistence of chaos-based and conventional digital communication systems of equal bit rate. IEEE Transactions on Circuits and Systems I: Regular Papers, 51(2), 391–408. https://doi.org/10.1109/TCSI.2003.822398

[12] Chen, Q., Wang, L., Lyu, Y., & Chen, G. (2018). Designing Protograph-Based LDPC Codes for Iterative Receivers on M-ary DCSK Systems. IEEE Transactions on Circuits and Systems II: Express Briefs, 65(4), 466–470.

[13] Faruk, O., & Ullah, S. E. (2018). Performance Evaluation of Encrypted Text Message Transmission in 5G Compatible Orthogonal Multi-level Chaos Shift Keying Modulation Scheme Aided MIMO Wireless Communication System.

[14] Wang, L., & Chen, G. (2004). Using LDPC codes to enhance the performance of FM-DCSK. Midwest Symposium on Circuits and Systems, 1, 401–404.

[15] Li, N., Martinez-Ortega, J. F., Diaz, V. H., & Chaus, J. M. M. (2018). A New High-Efficiency Multilevel Frequency-Modulation Different Chaos Shift Keying Communication System. IEEE Systems Journal, 12(4), 3334–3345. https://doi.org/10.1109/JSYST.2017.2715661

[16] Mackay, D. J. C., & Neal, R. M. (1997). Near Shannon Limit Performance of Low Density Parity Check Codes-To be published in Electronics Letters. Electronics Letters, 33(6), 457–458.

[17] Mahmood F. Mosleh, Fadhil S. Hasan, A. H. A. (2020). FPGA Hardware Co-simulation of DCSK Based on Bit-Flipping LDPC decoding using Xilinx System Generator. International Journal of Advanced Science and Technology, 29(4), 2285–2302.

[18] Mosleh, M. F., Hasan, F. S., & Azeez, R. M. (2020). Design and implementation of log domain decoder. International Journal of Electrical and Computer Engineering, 10(2), 1454–1568. https://doi.org/10.11591/ijece.v10i2.pp1454-1568

[19] Unal, B., Akoglu, A., Ghaffari, F., & Vasić, B. (2018). Hardware Implementation and Performance Analysis of Resource Efficient Probabilistic Hard Decision LDPC Decoders. IEEE Transactions on Circuits and Systems I: Regular Papers, 65(9), 3074–3084. https://doi.org/10.1109/TCSI.2018.2815008

[20] Xia, Y., Tse, C. K., & Lau, F. C.-M. (2004). Performance of differential chaos-shift-keying digital communication systems over a multipath fading channel with delay spread. IEEE Transactions on Circuits and Systems II: Express Briefs, 51(12), 680–684.

[21] Xu, W., Tan, Y., Lau, F. C. M., & Kolumban, G. (2018). Design and optimization of differential chaos shift keying scheme with code index modulation. IEEE Transactions on Communications, 66(5), 1970–1980. https://doi.org/10.1109/TCOMM.2018.2805342

[22] Yang, H., Jiang, G. P., Tang, W. K. S., Chen, G., & Lai, Y. C. (2018). Multi-Carrier Differential Chaos Shift Keying System with Subcarriers Allocation for Noise Reduction. IEEE Transactions on Circuits and Systems II: Express Briefs, 65(11), 1733–1737. https://doi.org/10.1109/TCSII.2017.2752754

[23] Zhang, C., Wang, L., & Chen, G. (2008). Promising performance of PA-coded SIMO FM-DCSK communication systems. Circuits, Systems, and Signal Processing, 27(6), 915–926. https://doi.org/10.1007/s00034-008-9067-2

[24] Mosleh, M. F. (2011). Evaluation of low density parity check codes over various channel types. Engineering and Technology Journal, 29(5), 961–971.