

FPGA implementation of LDPC soft-decision decoders based DCSK for spread spectrum applications

Fadhil S. Hasan¹, Mahmood F. Mosleh², Aya H. Abdulhameed³

^{1,3}Department of Computer Engineering Techniques, Middle Technical University, Baghdad, Iraq

²Department of Electrical Engineering, Al-Mustansiriyah University, Baghdad, Iraq

Article Info

Article history:

Received Aug 23, 2020

Revised Apr 23, 2021

Accepted May 19, 2021

Keywords:

DCSK

Hardware Co-simulation

LDPC

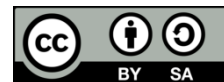
Soft decision decoder

Xilinx SG

ABSTRACT

Spread spectrum (SS) communications have attracted interest because of their channel attenuation immunity and low intercept potential. Apart from some extra features such as basic transceiver structures, chaotic communication would be the analog alternative to digital SS systems. Differential chaos shift keying (DCSK) systems, non-periodic and random characteristics among chaos carriers as well as their interaction with soft data are designed based on low-density parity-check (LDPC) codes in this brief. Because of simple structure, and glorious ability to correct errors. Using the Xilinx kintex7 FPGA development kit, we investigate the hardware performance and resource requirement tendencies of the DCSK communication system based on LDPC decoding algorithms (Prob. Domain, Log Domain and Min-Sum) over AWGN channel. The results indicate that the proposed system model has substantial improvements in the performance of the bit error rate (BER) and the real-time process. The Min-Sum decoder has relatively fewer FPGA resources than the other decoders. The implemented system will achieve 10-4 BER efficiency with 5 dB associate E_b/N_0 as a coding gain.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Fadhil S. Hasan

Department of Electrical Engineering

Al Mustansiriyah University

Baghdad, Iraq

Email: fadel_sahib@uomustansiriyah.edu.iq

1. INTRODUCTION

Being wide-band, chaotic signals are well suited for communication with the spread-spectrum [1]. Among the proposed digital schemes, the most comprehensive research was on chaos shift keying (CSK) and differential CSK (DCSK) [2]. The former was originally introduced for such a coherent implementation [3], requiring synchronized replicas of the chaotic base functions at the receiver. This requirement has still not been proved practical, though. On the other hand, the DCSK scheme represents a rather more robust non-coherent scheme [4] in which the receiver does not need the exact information of chaotic basis functions. Rather the two signal samples were correlated as well as the result of the correlation is compared with a threshold. The performance of digital communication systems based on chaos under an additive white gaussian noise (AWGN) environment has been studied thoroughly [5]-[8].

Low-density parity-check (LDPC) is one of the most effective techniques amongst the error correction codes. LDPC codes have gained a lot of attention recently since they can achieve exceptional performance close to the Shannon limit over the binary symmetric channel (BSC) and also the additive white gaussian noise (AWGN) channel [9]. Decoding an LDPC code allows for a high degree of parallelism, making it ideally suited for high data rate applications including wide-band wireless multimedia

communications and magnetic storage systems. The parity check matrix's low-density nature thus contributes to both great distance properties and the relatively low complexity of its decoding algorithm [10], [11].

Moreover, the excessive noise derives from a wide scale of mobile communication, the bit error rate (BER) is acceptable in modern communication with an even more high-speed data rate. Hence, support for the DCSK communication system through LDPC codes becomes essential to mitigate the high error [12]. The field programmable gate array (FPGA) is used to evaluate a system that included each DCSK and LDPC codes in a real-time environment. Gallager 's suggested basic decoding algorithm for LDPC codes in 1962 was soft decisions such as the sum-product algorithm (SPA) and hard decision such as the bit flipping (BF) [13]. SPA requires more steps to addition and multiplication which enables implementation in the case of real-time implementation with FPGA due to the simplicity of this operation. A crucial trade-off between 'complexity' and 'efficiency' is needed in iterative decoding. In this paper, we propose a model of a communication system including a DCSK as a modulation technique supported by three SPA (Prob. Domain, Log Domain and Min-Sum which is designed to reduce hardware complexity) to improve the system performance. The proposed system will be implemented using an FPGA Kintex 7 development kit integrated with Vivado 2017.4 software.

2. DCSK WITH LDPC ENCODED COMMUNICATION SYSTEM

The proposed system diagram is shown in Figure 1 that included a binary data source that will be encoded through the LDPC block with a code rate of 0.5. The codeword is sent to a DCSK modulator that uses a chaotic carrier for spread the digital signal across a wide frequency band to achieve a modulated signal with such a spreading factor value equal to 16. AWGN noise is indicated with the modulated signal which is mostly used for the experimental applications for simplicity. Non-coherent demodulation will then be adapted on the receiver side to recover the received code-word, that will be decoded to regenerate the original information.

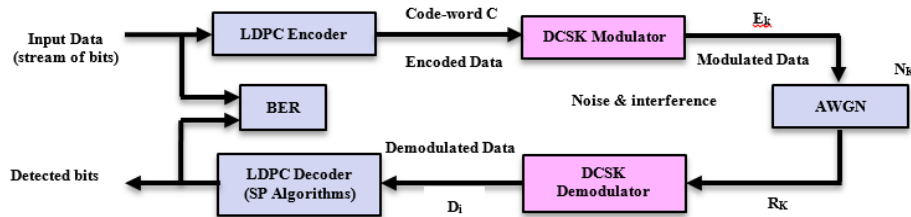


Figure 1. Block diagram of the system model

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 demodulator and LDPC decoder with three types of decoder algorithms such as; Prob. Domain, Log Domain and Min-Sum algorithms. The system model will be implemented using Xilinx 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. It is worth to mention that all the three decoder algorithms have the same design architecture except the part deal with decoder block.

3.1. LDPC encoder

The Bernoulli random binary number generated message word with a length of $k=10$ was fed to the LDPC block through gateway block to obtain a code word with a length of $n=20$ at the output of a certain block. There is a serial to parallel block within the LDPC encoder block shown in Figure 3 which will be used to convert a group of samples serially presented at the input to single samples present at the output is shown in Figure 4. In this paper, we have $k = 10$ as mentioned previously represented by $C_1 - C_{10}$ in (1) while $C_{11} - C_{20}$ represented the redundancy bits ($r = n - k$) [14].

$$S_i = [C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8 C_9 C_{10} C_{11} C_{12} C_{13} C_{14} C_{15} C_{16} C_{17} C_{18} C_{19} C_{20}]^T \tag{1}$$

The parity bits are incorporated using (2) and (3) which reflect the first and second parity bits corresponds respectively to first and second rows of the (4) matrix. The remaining parity is generated in a

same method that the rest of such matrix rows represent [15]. Figure 5 illustrates the hardware implementation of these equations that are implemented using 10 XOR gates to generate 10 parity check bits, based on the input message.

$$c_{11} = c_3 \oplus c_6 \oplus c_9 \tag{2}$$

$$c_{12} = c_5 \oplus c_6 \oplus c_7 \oplus c_8 \tag{3}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{4}$$

Now, the 20-bit code-word is supplied to concat block, concatenating two or more bits of inputs to get a symbol of these bits in the output. Eventually, the parallel to serial block induces the code word as output block. So that each sample provided in this block's input became multiple samples displayed serially in the output.

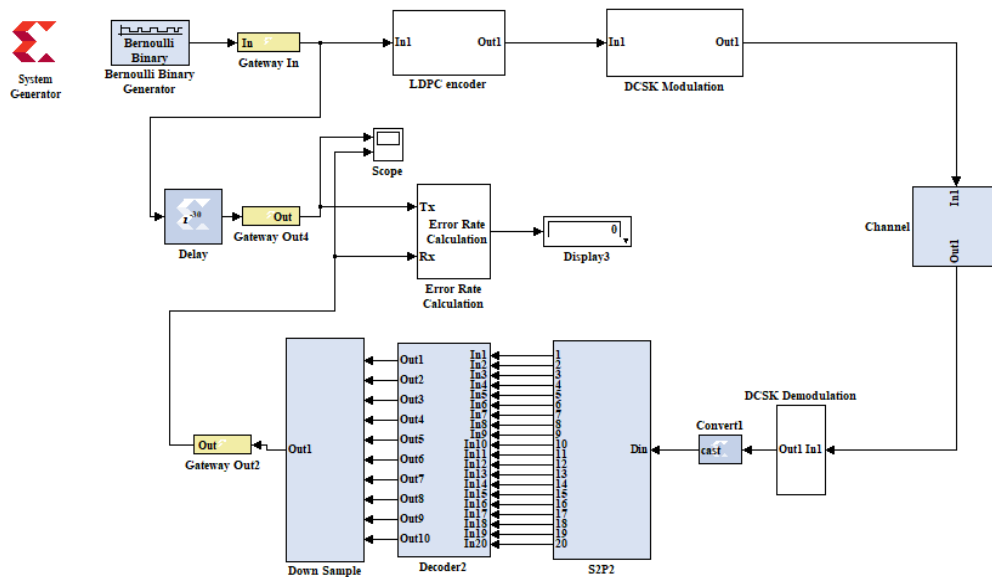


Figure 2. XSG of the system model

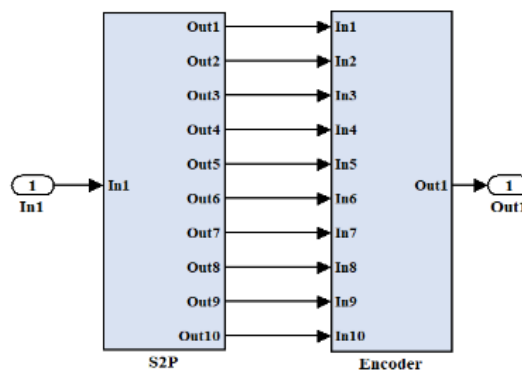


Figure 3. LDPC encoder

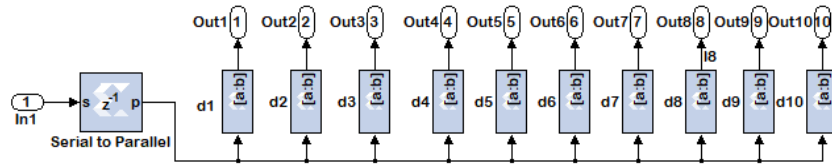


Figure 4. S2P block

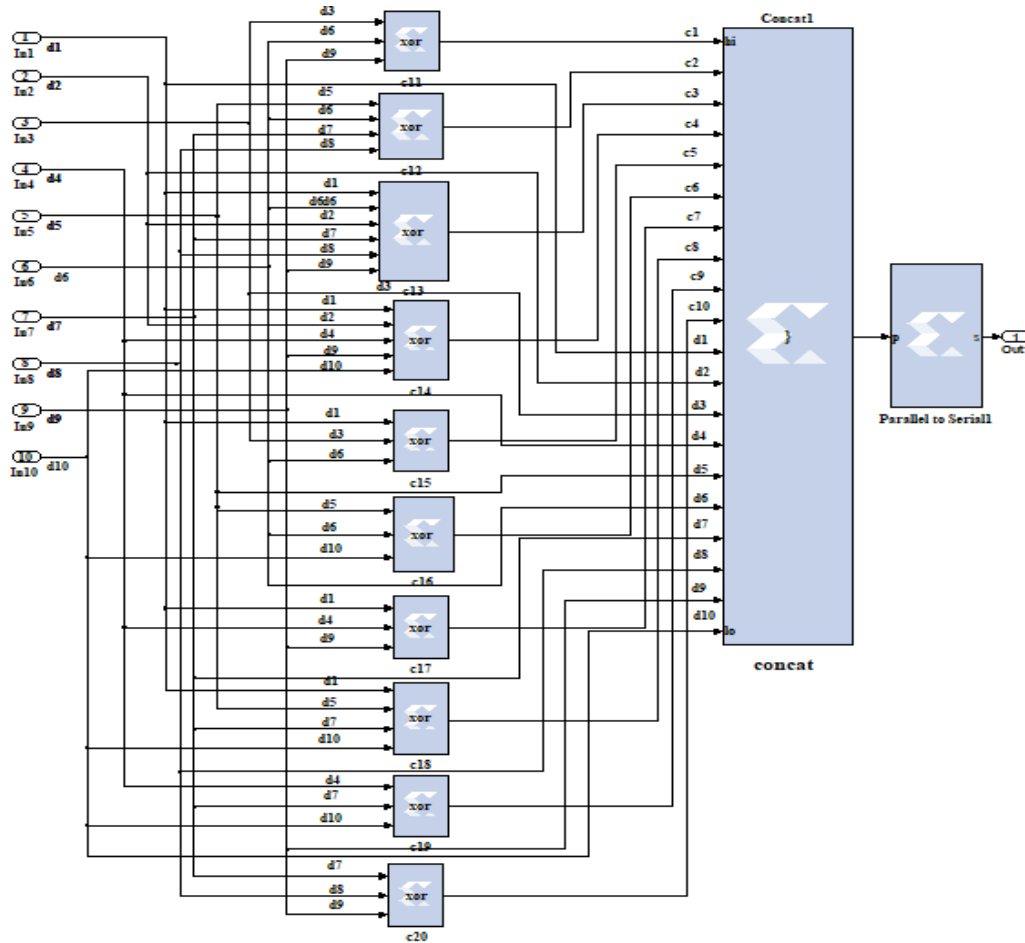


Figure 5. The hardware of encoder

3.2. DCSK modulator

The code-word generated by LDPC was fed to a Xilinx SG DCSK modulator designed. Throughout the DCSK modulator, each bit S_i is expressed by 2 sets of β -period chaotic signal samples where 2β refers to the spreading factor with such an integer number. The first and the second sets respectively represented the segment of reference and the segment of data. Samples from the data segment are equal to the reference segment for transmission +1, and an inverted variant of a reference segment is being used for transmission of bit -1. The transmitter e_k 's output during the i^{th} bit period is [16]-[18]. Figure 6 Shows the Xilinx SG block of DCSK modulation system with β equal to 8.

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

Both Chaos generator and mapping blocks are shown respectively in Figures 7 and 8. Chebyshev polynomial function (CPF) of order two as in (6) is selected as a chaotic generator with such an initial condition value of 0.1 to be implemented simply using a multiplexer, multiplier and adder blocks:

$$X_{K+1} = 1 - 2X_K^2 \tag{6}$$

The mapping block consists of a ROM to indicate either input {0} or input {1} according to the initial ROM vector value [-1, 1]. Delay was connected so the ROM block enabling pin can mask all serial bits before they become ready for the map operation.

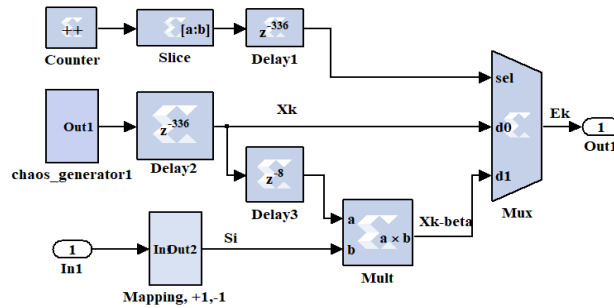


Figure 6. Xilinx SG design of DCSK modulator

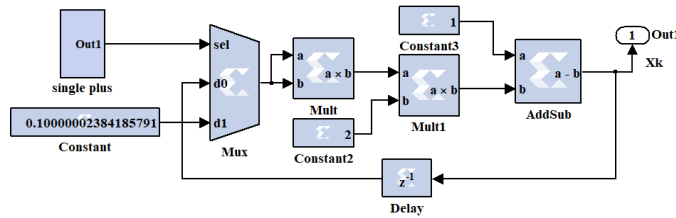


Figure 7. The chaos-generator block

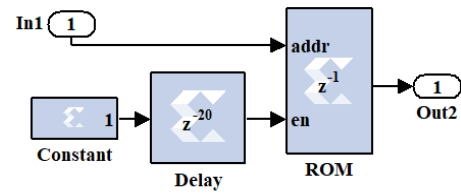


Figure 8. Mapping {+1,-1} block

3.3. Channel

If the modulated signal is transmitted over a channel, noise in the channel represented by additive white gaussian noise (AWGN), n_k , can corrupt it. Such noise characterized by a wide frequency spectrum, which is statistically random radio noise. The hardware-implemented for these channels is illustrated with seed value 512 in Figure 9.

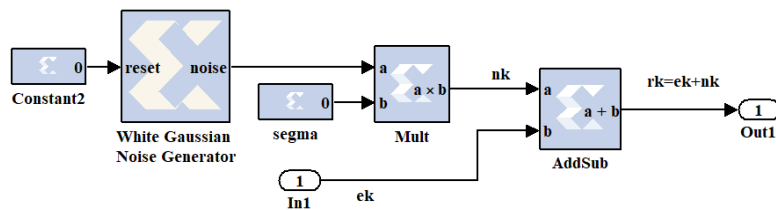


Figure 9. AWGN channel block

3.4. DCSK demodulation

To convert the 16 serial to parallel samples, the obtained signal r_k passes to an S2P block to de-spread within a single bit. Figure 10 illustrates the details of the Xilinx SG S2P block consisting of 16 latches and its corresponding delay blocks. S2P's 16 output samples are linked to a correlator, in which the reference samples as well as the corresponding information samples were correlated using multipliers. i^{th} correlator output is the variable i^{th} decision, D_i [19].

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

The i^{th} bit is demodulated using a zero threshold, measuring the sign of the final correlator output. The DCSK demodulator Xilinx SG is shown in Figure 11.

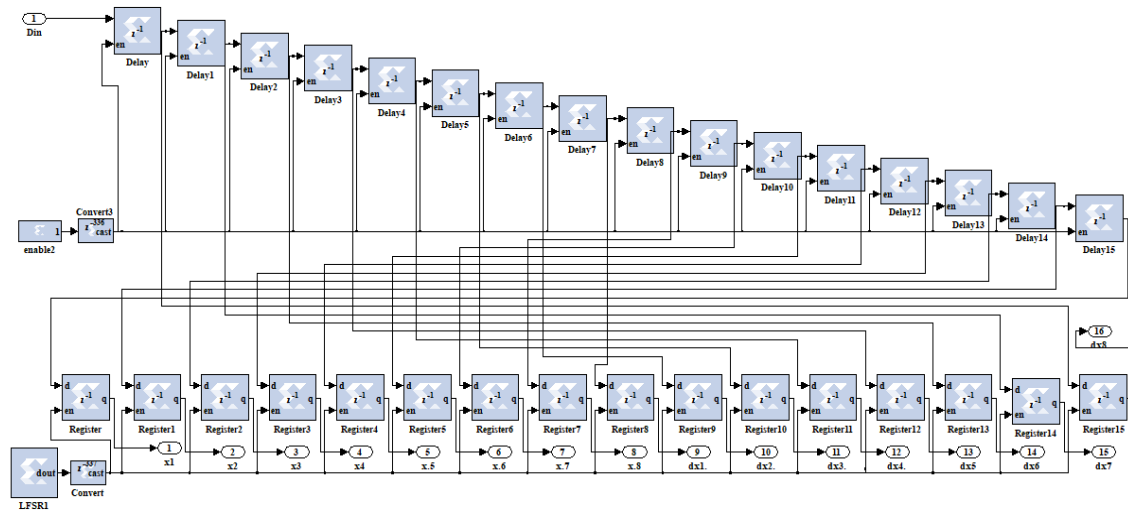


Figure 10. 16-samples S2P blocks at the demodulator

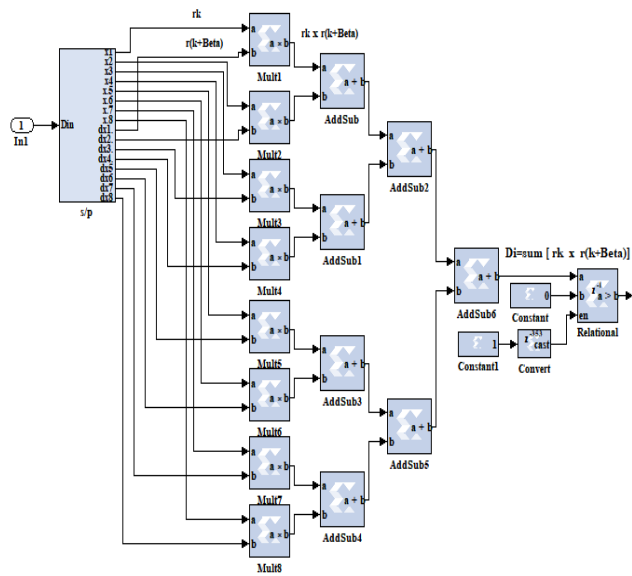


Figure 11. Xilinx SG DCSK demodulator block

3.5. Decoders

Prob. Domain, Log Domain and Min-Sum algorithms represent a type of soft decision algorithms and it works by passing messages between the CNs and the variable nodes (VNs). It is used to correct the received bits to obtain 10-bit symbols which will represent the original signal after the decoding process. The demodulated bits are 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, these algorithms perform in main three steps.

- Initialization
- Horizontal Step
- Vertical Step and Decoding/Estimation

The step details for each type of these algorithms will be described with a zoom-in view below.

3.5.1. Prob. domain decoder

Initialization: In the initialization step the value of $P(1)=Mij_{m,n}(1)$ and $P(0)=Mij_{m,n}(0)$ is obtained from (9) & (10) which represents the message that will be sent from the VN n to CN m . Each one of the messages has the pair $Mij_{m,n}(0)$ as well as $Mij_{m,n}(1)$ referring to the indication that rx has been one or zero [20].

$$P_n(1) = \frac{1}{1 + e^{\frac{-2rx}{N_0}}} \tag{8}$$

$$P_n(0) = 1 - P_n(1) \tag{9}$$

Where $P_n(1)$ and $P_n(0)$ are the posterior probabilities and their values can be found based on the received signal, also N_0 representing noise variance.

Horizontal step: The horizontal step calculation is created depending on the number of 1s over all columns of ten rows throughout the H matrix (11)-(13).

$$dFji = \prod_{n \in B_m, n' \neq n} (Mij_{m,n'}(0) - Mij_{m,n'}(1)) \tag{10}$$

$$Fji_{m,n}(0) = 1/2(1 + dFji) \tag{11}$$

$$Fji_{m,n}(1) = 1/2(1 - dFji) \tag{12}$$

These variable node has been designed by using the basic add-sub and mult Xilinx SG blocks to evaluate the $dFji$ by taking the product of subtraction $Mij_{m,n'}(1)$ also $Mij_{m,n'}(0)$ excluding bit n' , $Fji_{m,n}(0)$ as well as $Fji_{m,n}(1)$ which representing the probability which check m can be designed easily by using add-sub to add or sub one from $dFji$ value then mux the results by 0.5 using Xilinx SG block. Figures 12 and 13 illustrate the first row computations according to the (10)-(12) of the horizontal step.

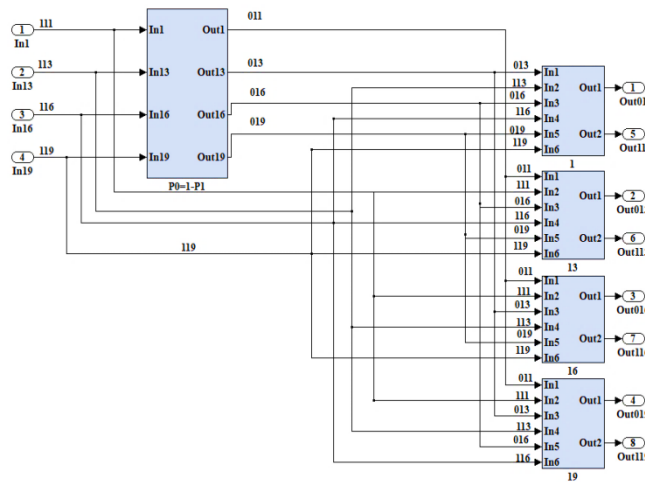


Figure 12. Horizontal step for the first row of prob. domain decoder

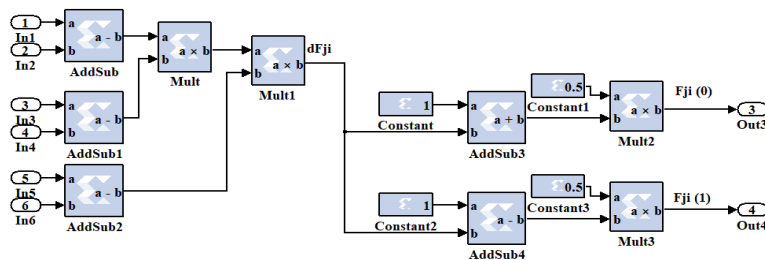


Figure 13. Details of the blocks (1) of the first row

Vertical step: This step's calculations are measured by the number of 1s for each row of such Twenty columns as in H matrix consistent with the (13) and (14).

$$ki_n(0) = P_n(0) \cdot \prod_{m=1}^m Fij_{m,n}(0) \tag{13}$$

$$ki_n(1) = P_n(1) \cdot \prod_{m=1}^m Fij_{m,n}(1) \tag{14}$$

Where $ki_n(0)$ and $ki_n(1)$ are scales factors which are implemented using Mult Xilinx SG blocks. Also, a decision for each bit is made according to (15).

$$vhat = \begin{cases} 1, & Q_{i_n}(1) > Q_{i_n}(0) \\ 0, & Q_{i_n}(1) \leq Q_{i_n}(0). \end{cases} \tag{15}$$

The Xilinx SG block of column twelve is described in Figure 14.

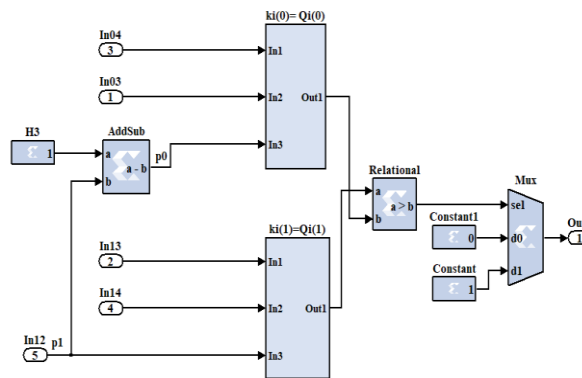


Figure 14. Vertical step (column 12)

3.5.2. Log domain algorithm

Initialization: In Log Domain 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 [21].

$$Lci_n = \frac{-4r_x}{N_0} \tag{16}$$

$$alpha_{m,n} = sign(Lci_n) \tag{17}$$

$$beta_{m,n} = |Lci_n| \tag{18}$$

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 15 and 16 respectively.

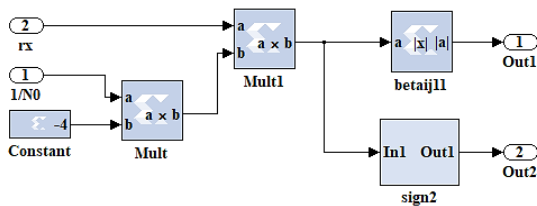


Figure 15. Initialization process

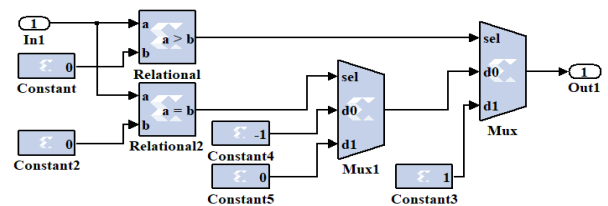


Figure 16. Details of Sign block

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. To Computing the extrinsic messages for each set of bits connected to CN m by excluding the bit n' the equation below will be used.

$$P_{i,m,n} = \log\left(\frac{e^{\beta \alpha_{i,j,m,n} + 1}}{e^{\beta \alpha_{i,j,m,n} - 1}}\right) \tag{19}$$

Where $P_{i,n,m}$ represents the probability that parity-check m has been achieved in the case when bit n is supposed to be a 1 for the LLR. Compute the summation of $P_{i,m,n}$ excluding the bit n .

$$SP_{m,n} = \sum_{n \in B_m, n' \neq n} P_{i,m,n'} \tag{20}$$

Then compute $PiS_{m,n}$:

$$PiS_{m,n} = \log\left(\frac{e^{SP_{m,n} + 1}}{e^{SP_{m,n} - 1}}\right) \tag{21}$$

Get products of $\alpha_{i,j,n,m}$ excluding the bit n :

$$Pr_{m,n} = \prod_{n \in B_m, n' \neq n} \alpha_{i,j,m,n'} \tag{22}$$

Finally, compute $Fji_{m,n}$

$$Fji_{m,n} = PiS_{m,n} \times Pr_{m,n} \tag{23}$$

To reduce the complexity of Xilinx SG design related to the logarithm (log) process in (19), a look-up table is proposed by taking the values of r_x and applying (19) as shown in Table 1. Figure 17 shows the details of the Xilinx SG design for the P_i calculation according to (19) for each bit equal to one.

Table 1. Look-up table for the values of P_i

Value of $\beta \alpha_{i,j}$	Value of P_i
$0 \leq r_x < 0.5$	0
$0.5 \leq r_x < 1$	1.4068
$1 \leq r_x < 1.5$	0.77194
$1.5 \leq r_x < 2$	0.45389
$2.5 \leq r_x < 3$	0.27234
$3 \leq r_x < 3.5 \dots \dots$ Continue	0.164540
$r_x > 31$	0

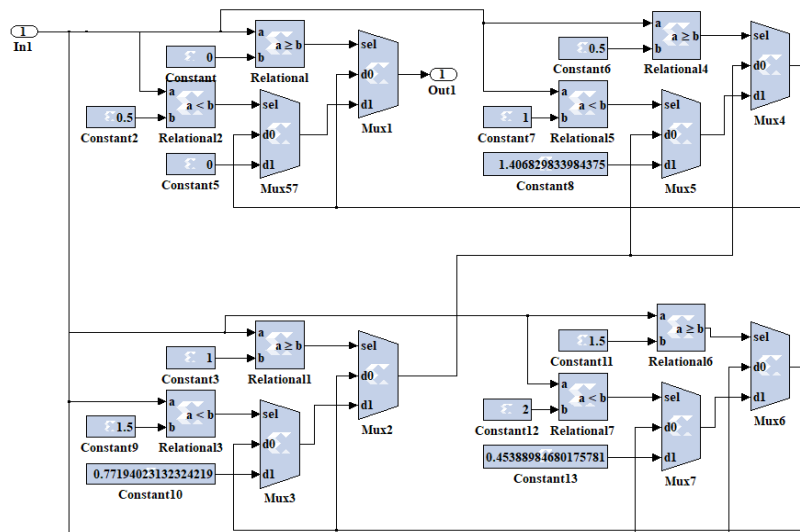


Figure 17. The P_i process block of Xilinx SG

For example, the Xilinx SG design for the first row will be implemented by using Mult and Add-Sub block according to (20)-(23) as illustrated in Figure 18. The details for each one of the blocks in Figure 18 can be declared in Figures 19-21. Furthermore, to reduce the complexity of Xilinx SG design related to the logarithm (log) process according to (21), a look-up table is proposed as illustrated in Figure 20 and Table 2.

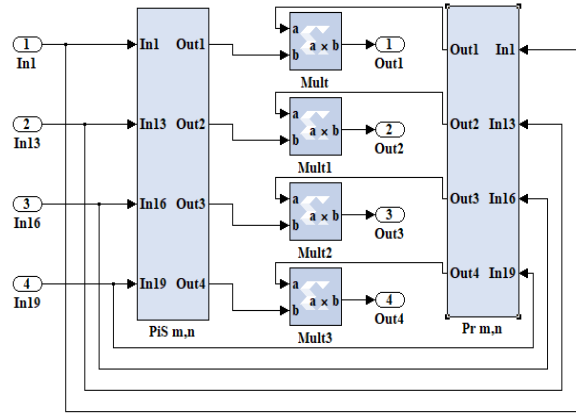


Figure 18. Horizontal step for the first row

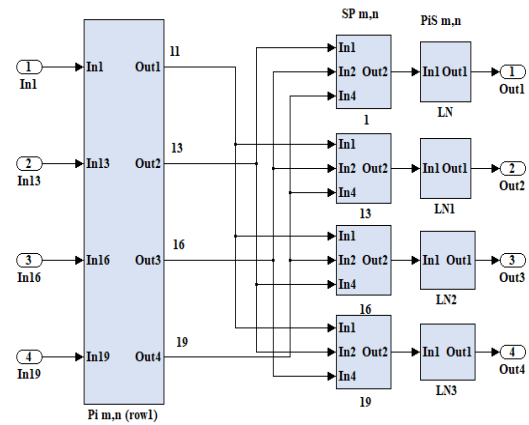


Figure 19. The PiS block of the first row

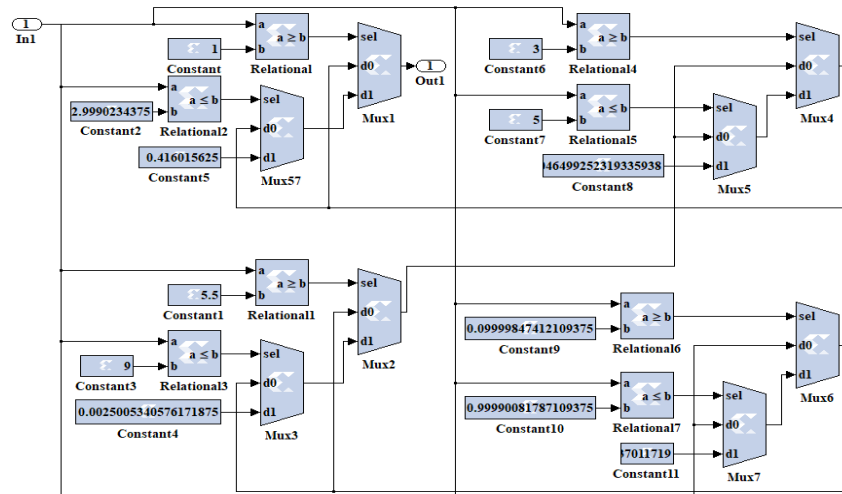


Figure 20. The PiS process block of Xilinx SG

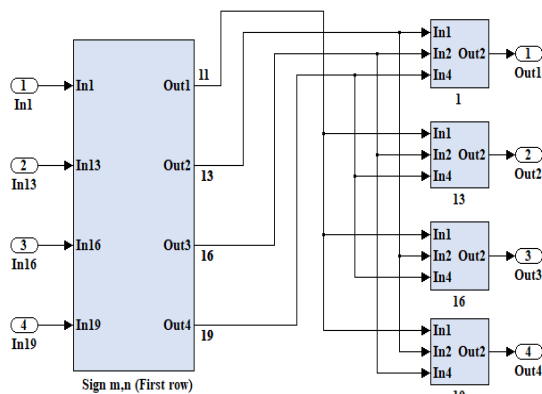


Figure 21. Pr block details

Table 2. The value of PiS

Values of SP	Values of PiS
$1 \leq SP \leq 2.999$	0.4195
$3 \leq SP \leq 5$	0.0465
$5.5 \leq SP \leq 9$	0.0025
$0.1 \leq SP \leq 0.9999$	1.5731
$0.01 \leq SP \leq 0.0999$	3.3603
$0.001 \leq SP \leq 0.009999$	6.1532
$0.0001 \leq SP \leq 0.0009999$	7.9917
$0.000001 \leq SP \leq 0.000009999$	10.7583
$0.0000001 \leq SP \leq 0.0000009999$	12.2428
$0.00000001 \leq SP \leq 0.00000009999$	15.3635
$0.000000001 \leq SP \leq 0.000000009999$	16.6158
$0.0000000001 \leq SP \leq 0.0000000009999$	19.9684
$0.00000000001 \leq SP \leq 0.00000000009999$	21.4741
$0.000000000001 \leq SP \leq 0.000000000009999$	23.1070
$0.0000000000001 \leq SP \leq 0.0000000000009999$	26.8764
Else	28.8261

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 (24), (25).

$$L_{Qi} = Lci_n + \sum_{m \in A_n} Fji_{m,n} \tag{24}$$

$$vhat = \begin{cases} 1, & L_{Qi} < 0 \\ 0, & L_{Qi} \geq 0. \end{cases} \tag{25}$$

Where L_{Qi} represent the collective log-likelihood ratio for n^{th} digit. Also, a L_{Qi} decision is made according to (25) 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 22.

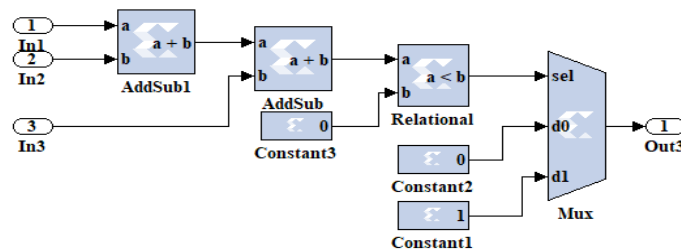


Figure 22. Vertical step blocks of column 12

3.5.3. Min-sum algorithm

The receiver side of the Min-Sum algorithm is very similar to the Log Domain algorithm except for the initialization process and horizontal step. It's implemented by replacing the implementation of (19)-(23) that belongs to the Log Domain initialization and horizontal step block by (26)-(29) respectively [22-24].

$$Lci_n = -r_x \tag{26}$$

$$Ai_{m,n} = \min_{n \in B_m, n' \neq n} (betai_{j,m,n'}) \tag{27}$$

$$Pr_{m,n} = \prod_{n \in B_m, n' \neq n} alphai_{j,m,n'} \tag{28}$$

$$Fji_{m,n} = A_{m,n} \times Pr_{m,n} \tag{29}$$

This process is implemented using Xilinx SG blocks (Multi, Mux, and Relational Xilinx blocks) to manage the value of the variable node as illustrated. The details are illustrated in Figures 23-28.

3.6. Down sample

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 rate of the output signal at the receiver. These bits from the output of the down-sample blocks 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 29. Once the Simulink models are developed for the proposed system, the VHDL code can be generated automatically using the SG block.

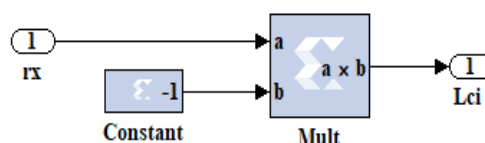


Figure 23. Initialization process (Lic)

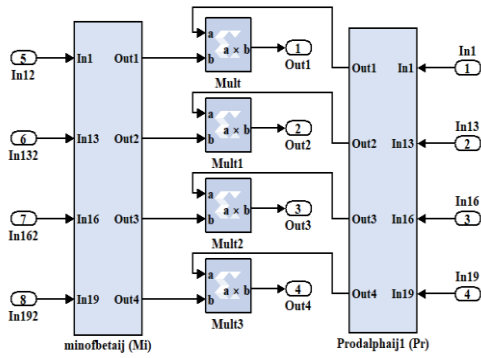


Figure 24. Horizontal step process of the first row

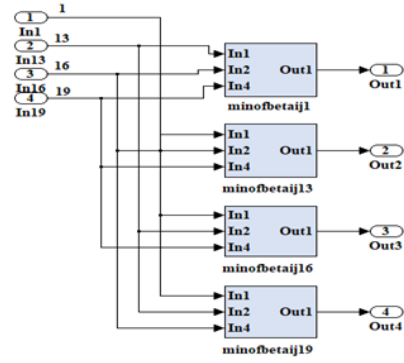


Figure 25. Details of Mi block

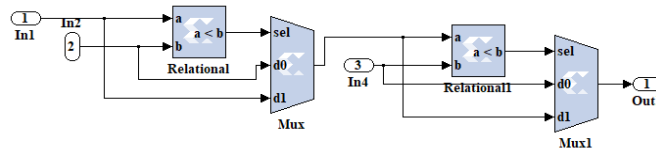


Figure 26. Details of Mi 1,13,16 and 19 blocks

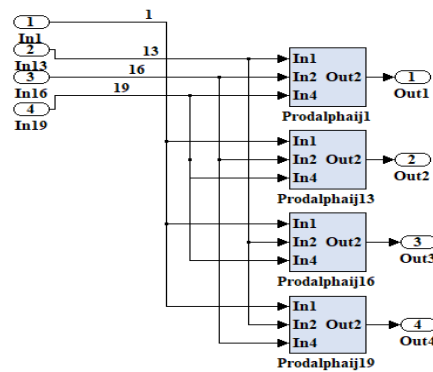


Figure 27. Details of pr block

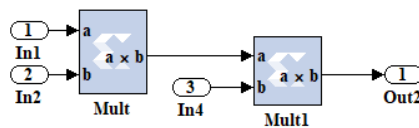


Figure 28. Details of Pr 1, 13, 16 and 19 blocks

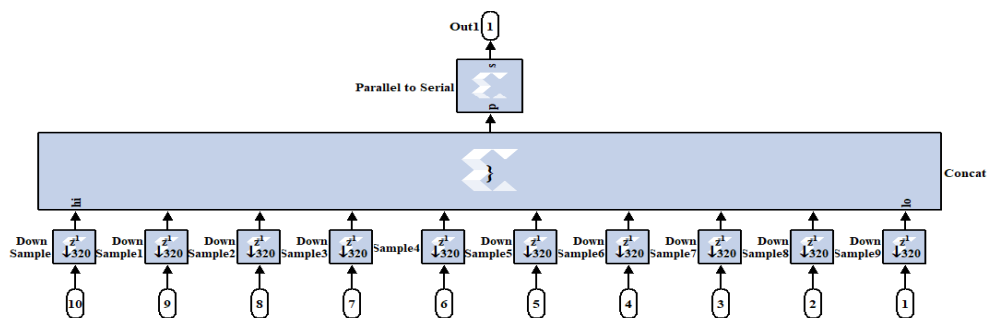


Figure 29. Down samples Xilinx SG block

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 (iter)=6, frame length (F)=100, code rate=1/2 and spreading factor (2β) value used are equal to 16. The results illustrated in Figure 30 are clear-out the comparisons between these communication systems with and without LDPC code.

From previous results, it is clear that the performance of the hard decision BF decoders will be worse than the soft decision decoder [25]. Also, the results show that close performances are illustrated when using Prob. Domain, Log Domain and Min-Sum, but all give approximately the same amount of gain. The system needs more than 13 dB SNR to approach 10^{-3} BER without code. However, using soft-decision decoder algorithms (Prob. Domain, Log Domain and Min-Sum) will improve the performance of the communication system at low and high SNR. All results are summarized in Table 3 for the spreading factor 16 at BER of 10^{-3} .

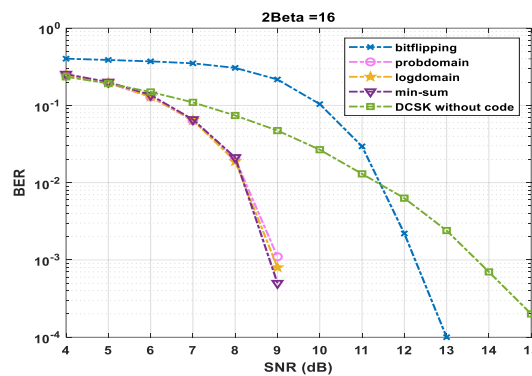


Figure 30. Performance comparisons, with spreading factor=16

Table 3. The summarized results of the simulation at the BER of 10^{-3}

Algorithms	SNR (dB)
2β	16
DCSK without LDPC	3.8
DCSK with BF decoder	2.3
DCSK with Prob. Domain decoder	9
DCSK with Log Domain decoder	9
DCSK with Min-Sum decoder	8.8

5. FPGA SYNTHESIS RESULTS

Tables 4 show the hardware resources devices utilization required for (Prob. Domain, Log Domain and Min-Sum) algorithms respectively. To measure the amount of complexity for the overall system in each type of soft decision decoder, Table 5 summarized the resource device utilization for each one. 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 4. Prob. Domain, Log Domain and Min-Sum resources device summary using Kintex 7

Device Summary	Prob. Domain	Log Domain	Min-Sum	Available
Slice Logic Utilization	Used	Used	Used	
Number of Slice Registers	3,856	3,025	3,322	407,600
Number of slice register sites lost to control set restrictions	3,071	23,997	1,984	50,950
Number of Slice LUTs	9,157	83,417	4,927	203,800
Number used as logic	7,961	82,609	3,984	203,800
Number used as Memory	1,196	808	943	64,000
Number of LUT Flip Flop pairs used	1,515	1,173	1,259	203,800
Number of bonded IOBs	3	3	3	500
Number of RAMB8BWERs	4.5	4.5	4.5	445
Number of BUFG/BUFGMUXs	1	1	1	32
Number of DSP48A1s	539	227	189	840

Table 5. Resource device comparison

Devices Utilization	Algorithms		
	Prob. Domain	Log Domain	Min-Sum
Slice register	3,856	3,025	3,322
LUT slice	9,157	83,417	4,927
DSP	539	227	189
Max-Delay (ns)	9,558	6,612	7,561

Figure 31 shows the Xilinx SG simulation test result among the Bernoulli binary transmitted signal and the received signal where all type of decoders is to record the original signal with 960 ns delay. The hardware Co-Simulation results of the user data information and recovered information for the proposed algorithms are shown in Figure 32 respectively where there is a delay due to the operation of the extraction process and there are some Xilinx blocks causes delay.

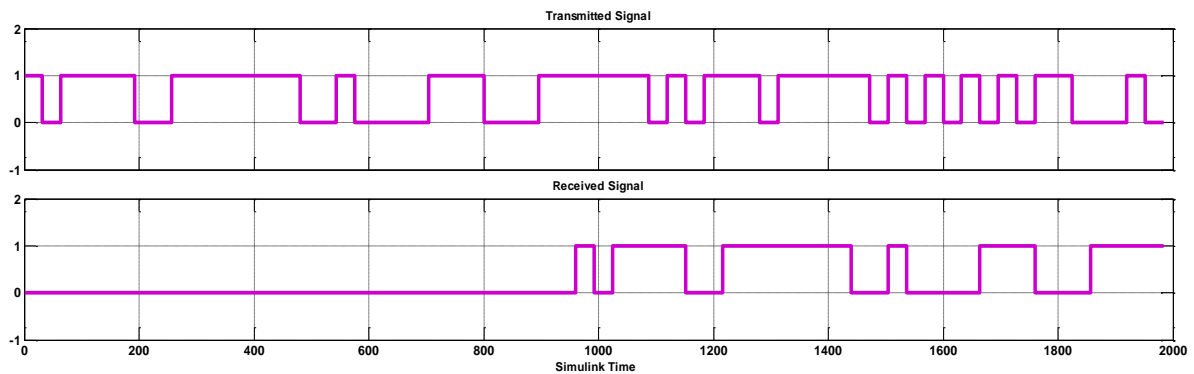


Figure 31. The simulated result of data recovery signal processing of Log Domain, Prob. Domain and Min-Sum decoder algorithms



Figure 32. Prob Domain, Log Domain and Min-Sum hardware Co-simulation implementation result

6. CONCLUSION

In this work, the DCSK communication system has been developed with LDPC codes by using Prob. Domain, Log Domain and Min-Sum decoding algorithms and implementation have been done on the Kintex 7 FPGA development kit using Xilinx SG tools. The simulation results show that Min-Sum is outperformed from other decoders were achieved 8.8 dB improvement gain while when using Prob. Domain and Log Domain the improvement gain is approximately the same amount as 9 dB for the 16 Spreading factors. With FPGA the results show that the Prob. Domain algorithm has the highest complexity in term of the resources utilization where its consumed 1% of the slice registers, 64% of DSP and 4% of LUT while Log Domain has consumed 3% of the slice registers, 27% of DSP and 41% of LUT. The consumption of the Min-Sum algorithm was 1% of the slice registers, 23% of DSP and 2% of LUT which means that the optimum decoding algorithms are the Min-Sum algorithm due to its BER performance and suitable resource utilization. The implementation of hardware confirms that the proposed system is suitable for future communications, especially in real-time applications.

REFERENCES

- [1] Y. Xia, C. K. Tse, and F. C.-M. Lau, "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*, vol. 51, no. 12, pp. 680-684, Dec. 2004, doi: 10.1109/TCSII.2004.838329.
- [2] F. C. M. Lau, C. K. Tse, M. Ye, and S. F. Hau, "Coexistence of chaos-based and conventional digital communication systems of equal bit rate," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 2, pp. 391-408, Feb. 2004, doi: 10.1109/TCSI.2003.822398.
- [3] H. Dedieu, M. P. Kennedy and M. Hasler, "Chaos shift keying: modulation and demodulation of a chaotic carrier using self-synchronizing Chua's circuits," in *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 40, no. 10, pp. 634-642, Oct. 1993, doi: 10.1109/82.246164.
- [4] A. Dukhan, D. Jayalath, P. van Heijster, B. Senadji, and J. Banks, "A generalized multilevel-hybrid chaotic oscillator for low-cost and power-efficient short-range chaotic communication systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, 2020, Art. no. 23, doi: 10.1186/s13638-020-1644-5.
- [5] W. M. Tam, F. C. M. Lau, and C. K. Tse, "2-Chaos-based digital communication systems," *Digital Communications with Chaos*, pp. 11-31, 2006.
- [6] G. Kolumban, "Theoretical noise performance of correlator-based chaotic communications schemes," in *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 12, pp. 1692-1701, Dec. 2000, doi: 10.1109/81.899921.
- [7] G. Kolumban, B. Vizvki, W. Schwarz, and A. Abel, "Differential chaos shift keying: A robust coding for chaotic communication," *Proc. 4th Int. Work. Nonlinear Dyn. Electron. Syst.*, pp. 87-92, 1996.
- [8] W. Xu, Y. Tan, F. C. M. Lau, and G. Kolumban, "Design and optimization of differential chaos shift keying scheme with code index modulation," *IEEE Transactions on Communications*, vol. 66, no. 5, pp. 1970-1980, May 2018, doi: 10.1109/TCOMM.2018.2805342.
- [9] V. A. Chandrasetty and S. M. Aziz, "FPGA implementation of a LDPC decoder using a reduced complexity message passing algorithm," *Journal of Networks*, vol. 6, no. 1, pp. 36-45, 2011, doi: 10.4304/jnw.6.1.36-45.
- [10] I. A. Altin, "New decoding methods for LDPC codes on error and error-erasure channels," M.S. Thesis, Department Electrical Engineering, Delft University of Technology, Delft, The Netherlands, 2010.
- [11] M. F. Mosleh, "Evaluation of low density parity check codes over various channel types," *Engineering and Technology Journal*, vol. 29, no. 5, pp. 961-971, 2011.
- [12] C.-L. Wang, X. Chen, Z. Li, and S. Yang, "A simplified min-sum decoding algorithm for non-binary LDPC codes," *IEEE Transactions on Communications*, vol. 61, no. 1, pp. 24-32, Jan. 2013, doi: 10.1109/TCOMM.2012.101712.110709.
- [13] R. G. Gallager, "Claude E. Shannon: A retrospective on his life, work, and impact," *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2681-2695, Nov. 2001, doi: 10.1109/18.959253.
- [14] P. Hailes, L. Xu, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "A Flexible FPGA-Based Quasi-Cyclic LDPC Decoder," *IEEE Access*, vol. 5, pp. 20965-20984, 2017, doi: 10.1109/ACCESS.2017.2678103.
- [15] M. F. Mosleh, F. S. Hasan and R. M. Azeez, "Comparison between different decoding algorithms for low density parity check," *2019 4th Scientific International Conference Najaf (SICN)*, 2019, pp. 5-10, doi: 10.1109/SICN47020.2019.9019379.
- [16] R. A. Mohammed, "Design of haar wavelet packet modulation based differential chaos shift keying and implemented by Xilinx system generator," M.S. Thesis, Middle Technical University, Baghdad, Iraq, 2018.
- [17] F. S. Hasan, "FPGA based 2x2 MMSE MIMO-OFDM system using Xilinx system generator," *Journal of Engineering and Sustainable Development*, vol. 21, no. 1, pp. 152-177, 2017.
- [18] M. F. Mosleh, F. S. Hasan, and A. H. Abdulhameed, "Enhancement of differential chaos shift keying communication system based on LDPC codes European Alliance for Innovation," *Proceedings of the 1st International Multi-Disciplinary Conference Theme: Sustainable Development and Smart Planning, IMDC-SDSP 2020*, Cyberspace, Jun. 2020, doi: 10.4108/eai.28-6-2020.2297915.
- [19] F. S. Hasan, M. J. Zaiter, and R. A. Mohammed, "Design and Analysis of a wavelet packet modulation based differential chaos shift keying communication system," *Wireless Personal Communications*, vol. 109, no. 4, pp. 2439-2450, 2019, doi: 10.1007/s11277-019-06691-6.
- [20] F. S. Hasan and M. F. Mosleh, "Designing LDPC system using prob domain decoder in Xilinx system generator," *IOP Conference Series: Materials Science and Engineering*, vol. 745, 2019, Art. no. 012033.
- [21] M. F. Mosleh, F. S. Hasan, and R. M. Azeez, "Design and implementation of log domain decoder," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 2, pp. 1454-1568, Apr. 2020, doi: 10.11591/ijece.v10i2.pp1454-1468.
- [22] F. Angarita, J. Valls, V. Almenar, and V. Torres, "Reduced-complexity min-sum algorithm for decoding ldpc codes with low error-floor," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 7, pp. 2150-2158, Jul. 2014, doi: 10.1109/TCSI.2014.2304660.
- [23] R. Anantharaman, K. Kwadiki, and V. P. Kerehalli Shankar Rao, "Hardware implementation analysis of min-sum decoders," *Advances in Electrical and Electronic Engineering*, vol. 17, no. 2, pp. 179-186, 2019, doi: 10.15598/aece.v17i2.3042.
- [24] W. Han, J. Huang and Fangfei Wu, "A modified Min-Sum algorithm for low-density parity-check codes," *2010 IEEE International Conference on Wireless Communications, Networking and Information Security*, 2010, pp. 449-451, doi: 10.1109/WCINS.2010.5541818.

- [25] M. F. Mosleh, F. S. Hasan, and A. H. Abdulhameed, "FPGA hardware co-simulation of DCSK based on bit-flipping LDPC decoding using Xilinx system generator," *International Journal of Advanced Science and Technology*, vol. 29, no. 4, pp. 2285-2302, 2020.

BIOGRAPHIES OF AUTHORS



Mahmood F. Mosleh, He received his B.Sc, M.Sc and Ph.D degrees in 1995, 2000 and 2008 respectively from University of Technology, Baghdad. He has been a Prof. of Communication Eng. at the Middle Technical University, Iraq. He has more than 60 publications in National and International Journals. Also, participate more than 20 International Conferences in the field of Communication Systems. Prof. Mahmood is the Head of Iraqi Electro-technical Committee which joint the International Electro-technical Committees, Member of Editorial Committee Technical Journal, and Member of various Committees for many International Conferences Organization.



Fadhil S. Hasan was born in Baghdad, Iraq in 1978. He received his B.Sc. degree in Electrical Engineering in 2000 and his M.Sc. degree in Electronics and Communication Engineering in 2003, both from the Mustansiriyah University, Iraq. He received Ph.D. degree in 2013 in Electronics and Communication Engineering from the Basrah University, Iraq. In 2005, he joined the faculty of Engineering at the Mustansiriyah University in Baghdad. His recent research activities are Wireless Communication Systems, Multicarrier System, Wavelet based OFDM, MIMO System, Speech Signal Processing, Chaotic Modulation, FPGA and Xilinx System Generator based Communication System. Now he has been an Assist. Prof. at the Mustansiriyah University, Iraq.

Email: fadel_sahib@uomustansiriyah.edu.iq



Aya H. Abdulhameed was born in Iraq on 1993. Recived the B.Sc. in Computer Engineering Techniques, Middle Technical University, Baghdad Iraq, in 2016. Currently pursuing the MSC. Program in Computer Technical Engineering with an emphasis in communication systems.