TELKOMNIKA Telecommunication Computing Electronics and Control
Vol. 20, No. 3, June 2022, pp. 510~518
ISSN: 1693-6930, DOI: 10.12928/TELKOMNIKA.v20i3.23300

A hybrid soft bit flipping decoder algorithm for effective signal transmission and reception

Shaik Asif Hussain[1], Jyothi Chinna Babu[2], Raza Hasan[3], Salman Mahmood[3]
[1]Centre for Research and Consultancy, Middle East College, Muscat, P.B. No. 79, Al Rusayl 124, Oman
[2]Department of Electronics and Communication Engineering, AITS, Rajampet, AP, India
[3] Department of Information Technology, School of Science and Engineering, Malaysia University of Science and Technology, Petaling Jaya 47810, Selangor, Malaysia

Article Info

Article history:

Received Jun 09, 2021
Revised Mar 06, 2022
Accepted Mar 14, 2022

Keywords:

EG-LDPC codes
Hybrid SBF algorithm
LDPC decoder
Medical applications
Signal transmission

ABSTRACT

The Euclidean geometry (EG) based low-density parity check (LDPC) codes are enciphered and deciphered in various modes. These algorithms have the back-and-forth between decoding delay, and power usage, device unpredictability resources, and error rate efficacy are all available with these methods. As a result, the goal of this paper is to develop a comprehensive method to describe both soft and burst error bits for optimal data transfer. As a result, for EG-LDPC codes, a hybrid soft bit flipping (HSBF) decoder is suggested, which decreases decoding complications while improving message data transfer. A simulation model is formed using Xilinx synthesis report to study decoding latency, hardware usage, and power usage. A HSBF decoder is used in this paper, which accepts a 64-bit coding sequence and assigns 64 Adjustable nodes to it. It checks all customizable cluster connections and quantifies adjustable node values and actions. As a consequence of the data collected, our simulation model demonstrates that the HSBF technique outperforms soft bit flipping (SBF) algorithms. As a result, the techniques are ideal for usage in intermediate applications and as well as in cyber security processing technologies, medical applications.

Corresponding Author:

Shaik Asif Hussain
Centre for Research and Consultancy, Middle East College, Muscat
P.B. No. 79, Al Rusayl 124, Oman
Email: shussain@mec.edu.om

1. INTRODUCTION

Euclidean geometry LDPC codes

The Euclidean geometry low-density parity check (LDPC) codes are a better-quality type of fault detection and pattern development codes, operating speed, error rate efficiency, which is assumed to be difficult to decode the codes. In this subject of cryptography, fast development is demonstrated in fields such as digital television streaming video, wireless local area network, health data storage, and third-generation mobile telephony using Euclidean geometry LDPC codes. As per the coding rates of LDPC codes, 5G LDPC codes are a data coding method that is meant to handle high throughput, variable code rate and length, and hybrid automated repeat requests, as well as strong error correction capabilities. LDPC codes are non-systematic codes by nature and are created at random. Using systematic rather than non-systematic codes help the decoder to acquire the decoded data without having to go through a time-consuming mapping process.

Illustration of LDPC codes

Euclidean geometry LDPC codes are better versions of linear block codes having independent sizes of equality-controlled matrices, which can have fewer ones, as indicated by the categorization. This patterned matrix is usually constructed using an unconstrained approach that is constrained by such strict constraints.

Such Euclidean geometry LDPC codes deviate from the parity patterned matrix (H) in case of functional equivalence verification [1]. Every row has 'μ' number of ones. Each column can be designed with 'γ' quantity of ones. The frequent amount of the ones in any preceding columns is 'σ' and it should not more than unity. Here 'ρ' and 'γ' are not connected to distance of the code and the rows available in the patterned parity matrix H [2]. When 'μ' and 'σ' are small and the patterned parity matrix H has a less significant deliberation. The recommended Euclidean geometry LDPC codes are consequently deciphered by an iterative technique and imagery of their patterned parity check sparse matrix H [3]. Frequently, these codes are depicted in numerous ways; they are matrix and graphical representation [4]. The dimension of the provided matrix is eight by four, with n number of adjustable and patterned nodes. Every row is having the strength of 6 and column has the strength of 3 [5]. While the whole building block distance end to end is 'n' and the number of communication bits 'k' and check equivalency or parity information bits $n-k$ are helpful to size, the information word (n,k). The patterned parity H-matrix is simply one when there is a connection between the changeable nodes and reproduction nodes inevitable for all the column prerequisites and rows of the graph [6]. Such illustration contains n nodes $(i.e.n=8)$ and k nodes $(i.e.k=4)$ as revealed in Figure 1.

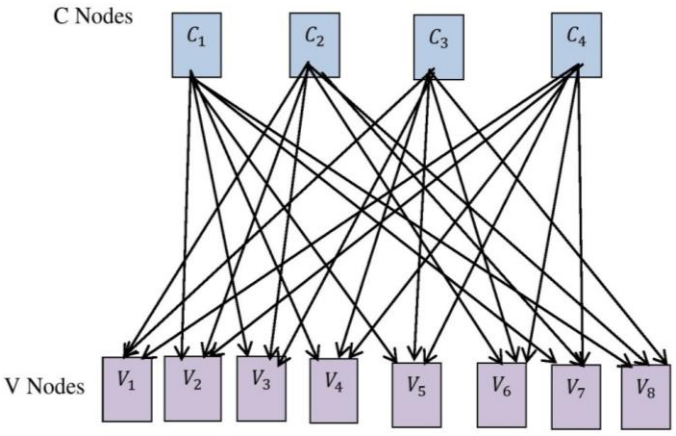

Figure 1. Graphical representation of H-matrix

Creation of patterned parity check matrix

The H_{mxn} equivalence patterned check matrix is indicated as (N,K), where N is length of general code vector, represents the amount of information bits in addition to it consists of $N-K$ amount of equivalent information bits [7], [8]. A parity check matrix of a linear block code in coding theory. C is a matrix that outlines the linear relationships that codeword components must follow. It is used in decoding algorithms and may be used to determine whether a given vector is a codeword. The coefficients of the parity check equations are the rows of a parity check matrix [9]. That is, they demonstrate how particular digits (components) of each code word may be linearly combined to equal zero.

Conception of sparse parity check matrix

- Step 1: determination of primitive polynomial
- Step 2: determination of degree terms
- Step 3: determination of generator polynomial
- Step 4: construction of parity patterned matrix

Decoding of LDPC codes

To decrypt Euclidean geometry based LDPC codes on very large-scaleintegration (VLSI), the soft bit flipping (SBF) decrypting algorithm, majority based decoder/detector method and belief propagation decoding method have been proposed in the work [10], [11]. These methods are being used to estimate performance of a system by using the two parameters: resource usage and decoding latency. In this paper, a novel hybrid soft-bit-flipping (HSBF) deciphering approach for LDPC codes was suggested [12]. The suggested method evaluates every bit is true error term, node information verification, and fundamental data information using two variety of information transfer mechanisms. The SBF decoder [13], [14], the weighted bit flipping (WBF) decoder, belief propagation decoding (BPD) decoder, sequential peeling decoder (SPD), and parallel peeling decoder (PPD) techniques were used to investigate the potential causes of problems when transferring a message among a transmitter and the receiver. The Figure 2 depicts the general building blocks of the decryption process. Considering the aforementioned occurrence, this work suggested a novel architecture for Euclidean geometric (EG)-LDPC codes termed as HSBF method.

The objectives of the paper are:

– A 64-bit coding sequence is received and allocated to 64 adjustable nodes in order to construct an algorithm which uses HSBF method. It checks the node connections to all changeable nodes and computes the node values of pattern, and conducts the majority based patterned activities. This procedure will be followed till the operations are completed.
– Created a simulation model to demonstrate that the HSBF method outperforms other deciphering techniques such as SBF, BPD algorithm, SPD, WBF algorithm, majority logic decoder/detector (MLDD) algorithm and PPD decoding algorithms. The LDPC decrypting methods common block structure is shown in diagram and briefly detailed in section 2.1. The following diagram has input is message and output is error free output. Where the matrix mentioned in diagram is a systematic based parity check matrix.

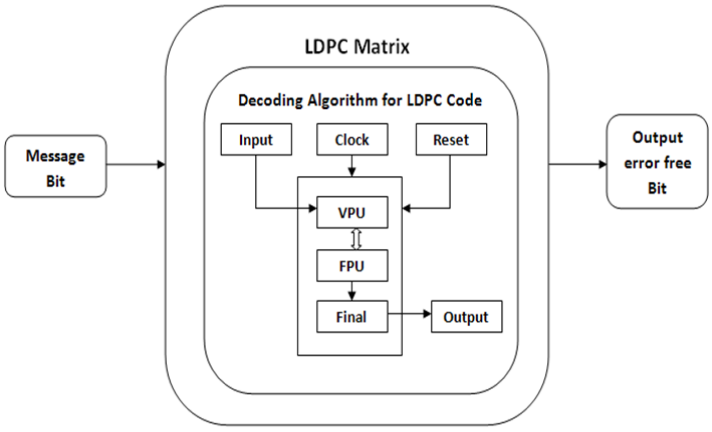

Figure 2. Schematic diagram of LDPC decoder

2. COMMON PROCESS FOR ENCIPHERING AND DECIPHERING OF EG-LDPC CODES

The code pattern and patterned parity matrix medium is different as (N, K), everywhere N is entire data bits size and K represents number of data bits and $N - K$ indicates quantity of the parity data bits. Where the matrix 'G' is known as designer matrix, and it can be constructed from the parity matrix by considering the following conditions and the systematic matrix $Hsys = [Im|\, Pm \times k]$ is determined by changing the patterned parity matrix and by means of row and column computations. Reorganize the patterned parity matrix $(Hsys)\ asG = [P^T k \times m|I_k]$. Authenticate the result and it is shown as $G.H^T = 0$. It might produce the code vector by considering the product with information$m$ and with designer matrix$G$. The enciphered information vector is supposed to be legally recognized if it satisfies the condition $c.H^T = 0$.

2.1. Concise description of the conventional decrypting methods

2.1.1. SBF decoding algorithm

The SBF decrypting algorithm is considered as a hard decision decrypting-based data transitory algorithm to decipher the Euclidean geometry based LDPC codes. The received hard decision data bit is binary and it can be identified then given to the decoder to consider the processing of further steps [15]. Then the data is moved to SBF decoder in the course of edges of the tanner graph for representing with ones and zeros. If the valid codeword is obtained, then the steps of SBF algorithm can be stopped [16], [17].

2.1.2. MLDD algorithm

The MLDD decrypting algorithm works as an error detection and correction method with fewer amounts of iterations than the majority logic decoder (MLD) method. The MLDD decoder is only dedicated for binary codes [18]. The majority based deciphering technique can be chosen to detect the original decoding data. In MLDD, the performance can be improved by reducing the decoding latency [19], [20].

2.1.3. SPD decoding algorithm

SPD decoding algorithm used to reduce the irresolvable variable nodes and used to determine the output bits during the decoding process. This SPD algorithm is generally used binary channel to transfer the information from transmitter to receiver. The SPD algorithm is an iterative based method and having less complexity to decode the output bits [21]-[23].

2.1.4. PPD decoding algorithm

PPD decoding algorithm is an Iterative based signal computation method. In every phase, the generic code vector involves in transferring the data from test nodes to changeable nodes. PPD algorithm can be used to resolve the deg-1 check nodes with the less time cycles compared to SPD algorithm [24], [25].

2.1.5. BPD decoding algorithm

BPD algorithm is an iterative process. In BPD, through the variable nodes the data is transmitted, and these variable nodes are connected together. BPD algorithm can be used to decoding the data based the belief propagation method. This decoding algorithm can be stopped, when it reaches maximum no. of iterations or parity check equations are equal to zero [26].

3. HYBRID SBF ALGORITHM

Hybrid SBF algorithm is proposed in this paper to decode the data bits. This algorithm is used in data decoding systems. In this paper, HSBF decoder is used to decode 64-bit data by using hard decision decoding (HDD) sequence method. The numerical method of hybrid SBF decoder is given below.

3.1. Numerical method

Let us consider the expected actual sequence is $y = y_1, y_2 \dots \dots . . y_n$ analogous to the twofold output assessment progression can be $Z = Z_1, Z_2 \dots \dots . . Z_n$. The LDPC code has a linear graph with quantity j for all variable nodes and degree k for all check nodes. Euclidean geometry based LDPC code pattern is indicated by using N, K, dv, dc. Everywhere N, K indicates the code length of vector and thus the bits of the data correspondingly. Where dv, dc represents the weights of rows & columns respectively. Let us assume n-bit code pattern calculates $n - k$ syndrome vector bits. Expected word $R = C + E$
Where C is an expected code vector, and E is an arbitrary variable.

Calculate the syndrome pattern $S = R.\ H^T$ (1)

$l\ x\ k$ $k\ x\ n$ $\rightarrow$ $l\ x\ n$
Data Size Originator Matrix Code Vector Size
The Originator matrix is $G = (I_{(k*k)}, P_{(k*(n-k))}$
By considering d from patterned parity matrix (2)

$$H = \left(P^T_{((n-k)*k)}, I_{(n-k)*(n-k)}\right) \quad (3)$$

3.2. Syndrome pattern steps

a) Syndrome pattern is computed for a particular code vector error pattern, and the resultant syndrome vector pattern is updated using:

$$S_i = E_i . H^T \quad (4)$$

b) At the receiver end, estimate the code sequence of every code vector which is received. In the hybrid SBF decoding method the step for decoding the data is used in the following manner. Consider the received input data $y = y_1, y_2 \dots \dots . . y_n$ and acquire the decoding data sequence $Z = Z_1, Z_2 \dots \dots . . Z_n$. Depends on the data input, calculate the syndrome pattern, and conclude the maximum likelihood based random code vector considering in every test.

$$S_m = \sum_{n=1}^{N} Z_m . H_{mn}(1) y_m^{min} = {}_{n \in N(m)}^{min} |y_n| \quad (5)$$

Where $|y_n|$ represents the complete assessment of the n^{th} transmitted data node assessment, whereas y_m^{min} is the smallest quantity of total message nodes considering inside the m^{th}check. Where, $N(m)$ indicates total amount of variable nodes which are available in the m^{th} node check.

c) Determine error term E_n by choosing the equation.

$$E_n = \sum_{m \in M(n)} (2\ S_m - 1) \times t_n \quad (6)$$

Locate one of most frequently associated data node for every test node.

$$r_m^{min} = {}_{n \in N(m)}^{min} |yn| \quad (7)$$

For every communication node, determine the error data term.

$$E_n = \min_{m \in M(n)} (2S_m - 1) \times t_n \quad (8)$$

d) Complement the data bit with the maximum error term value E_n by considering the majority voting process and modernize the syndrome pattern $S = R.H^T$ where majority voting process is given by:

$$S_n \leftarrow S_n \cup (E_n xor S_n). \quad (9)$$

e) Iterations are performed based on parity check satisfaction until then the decoder is frequent from the steps a to c. In authenticity, exclusive of transmission of data communication, every changeable node's data considered from the preceding nodes is strictly inhibited. The term $2Sm - 1$ is suitable for transferring the required data coming from the test nodes. The innovative error phrase is given by:

$$E_n = \sum_{m \in M(n)} (2\, S_m - 1) \times t_n \quad (10)$$

The lack of knowledge of the reliability of environs incompatible nodes can be appreciably decreasing the process of decoding complexity.

3.3. Computational analysis

In comparison to conventional bit flipping deciphering techniques, the suggested hybrid SBF algorithm is more hardware friendly. In hybrid SBF decoder, required number of wires can be verified first.

$$E_n = \frac{\sum_{m \in M(n)} (2S_m - 1)}{|y_n|} \quad (11)$$

From the (11), to calculate the new error word E_n, every changeable node needs two data inputs: S_m and $|y_n|$. Where $|y_n|$ the amount of the acknowledged value, that value is is stored in the changeable nodes. The syndrome data bit S_m can be established by the preceding test of the bit and it can decode one data bit at an instance. Likewise, every test node needs the indication of every changeable node to calculate S_m, and it can be tested one data bit at an instance. Therefore, all edges in the pictorial representation allocates into only one couple of wires. The novel HSBF decrypting method takes $\frac{q-1}{q}$, where q is quantization bits of wires only, at any time by distinguishing with the other conventional bit Flipping decoders. By means of considering the changes in (8) and it can be modified as:

$$t_n = \frac{1}{y_n} \quad (12)$$

$$E_n = \sum_{m \in M(n)} (2S_m - 1)\, X t_n \quad (13)$$

It's worth noting that in (8), the division operation is depending on the testing operation of each and every bit. In addition, the obtained sequence y is fed into the decoding process one bit at a time. As a result, just one basic multiplier is required when getting y for the first time. Instead of Yn, tn is kept in each variable node. This multiplication may be performed using a look-up table (LUT) operation, which is extremely fast and space-saving for input words with short lengths.

$$E_n = \sum_{m \in M(n)} (2S_m - 1)\, X t_n \quad (14)$$

In the (10) specifies a product term with few options. When analyzing the proposed methodology to other available methods, it is clear that conventional decrypting algorithms require more transactions in order to calculate E_n. Existing decoding techniques are hardware intensive and time expensive due of these complicated procedures.

In the bit flipping and peeling based decoding methods, Table 1 demonstrates the logical functions utilization per cycle in each changing node and test node. Weighted bit flipping decoder and improved weighted bit flipping algorithms took 38 cycles per iteration, MLDD took 9 cycles per iteration, BPD took 9 cycles per iteration, reliability ratio based weighted bit flipping (RRWBF) decoder took 39 cycles per iteration, and HSBF algorithm took 7 cycles per iteration. As a result, the hybrid SBF decoder uses the fewest possible cycles to complete one loop. As a result, this study suggests that HSBF has lower latency than alternative decoding methods. Hardware reduction is achieved if complexity of the decoding process and delay are reduced.

Table 1. Decoding latency (per cycle) comparison of conventional and proposed decrypting methods

No.	Decoder name	Decoding latency in cycles
1	WBF decoder	38 Cycles/iteration
2	MLDD decoder	09 Cycles/iteration
3	MWBF decoder	38 Cycles/iteration
4	BPD decoder	09 Cycles/iteration
5	RRWBF decoder	39 Cycles/iteration
6	SRWSBF decoder	07 Cycles/iteration

4. RESULTS AND DISCUSSIONS

The modeling technique and results are provided in this section. The Xilinx integrated synthesis environment (ISE) modelsim simulator is used to carry out the simulations. Experiments are carried out using test bench, which verifies all information, reducing testing durations and information footprints for each verification. The following are the findings of the modeling and synthesis for the hybrid SBF method. Verilog hardware description language (HDL) code was used to develop the HSBF method for Euclidean geometric-LDPC codes. The findings are seen using the Xilinx ISE modelsim simulator. Figure 3 shows the deciphering process for 64-bit data. The (12) is used to determine the assessment parameter decoding delay.

$$Latency = CC \times CP \tag{15}$$

Where:
CC = Total required no. of clock cycles to obtain the output
CP = Minimum required clock period
The required amount of clock cycles known as CC is determined from the (16).

$$CC = \frac{Time\ period\ needed\ for\ decoding\ the\ output}{Time\ period\ Interval} \tag{16}$$

4.1. Decoding procedure of 64-bit data

Initially, the given 64-bit input is divided into four16-bit frames and they are $d\,[15:0]$, $d\,[31:16]$, $d\,[47:32]$, and $d\,[63:48]$. The individual frame is given to each encoder as an input, hence it can use four modules of encoder. Since it can have four encoders and need to use four decoders. For encoder, 16-bit frame is given as input and is denoted as $d\,[15:0]$. The resultant procedure of 64-bit data input is shown in Figure 3. The experimental results of the hybrid SBF decoder was compared with SBF decoder, MLDD decoder, BPD decoder, SPD decoder, and PPD decrypting techniques.

Table 1 represents the decoding latency (per cycle) comparison of conventional and proposed decrypting methods. It should be mentioned that the suggested research work in this paper reduces the usage of power by a value 41.68% at an average by compared to the available methods considered in this paper. Table 2 shows that, how the performance metrics of proposed algorithm achieves better computation than existing algorithms. Where, hybrid SBF decoder is also known as self reliability based weighted soft bit flipping (SRWSBF) decoder. Table 2 compares the deciphering procedure' delay to that of many alternative deciphering methods. The typical delay of 16.65% is observed to be lowered by employing the suggested deciphering method, as shown in Table 2. The resource consumption is seen to be reduced to an aggregate of 4.25% in the examined synthesis conditions. The suggested HSBF deciphering technique uses significantly more slices flip-flops resources (1.85%) than previous vailable methods. Figures 4 and Figure 5 illustrate the rate of consumption that clearly demonstrates that when trying to compare HSBF; suggested study with deciphering techniques are known to be SBF decoder, MLDD decoder, BPD decoder, PPD decoder and SPD decoder the usage of hardware resources is decreased. The suggested identified research HSBF deciphering technique was also discovered to employ a significantly maximum amount of slices flip-flops.

Table 2. Comparison of performance metrics

No	Algorithm	Decoding latency (ns)	Hardware complexity (%)	Power consumption (W)
1	HSBF	1556.00	23.68	0.442
2	SBF	2131.92	25.90	0.764
3	MLDD	1801.32	26.85	1.608
4	BPD	1749.36	27.54	0.818
5	SPD	1898.52	27.74	1.124
6	PPD	1753.16	24.60	0.818

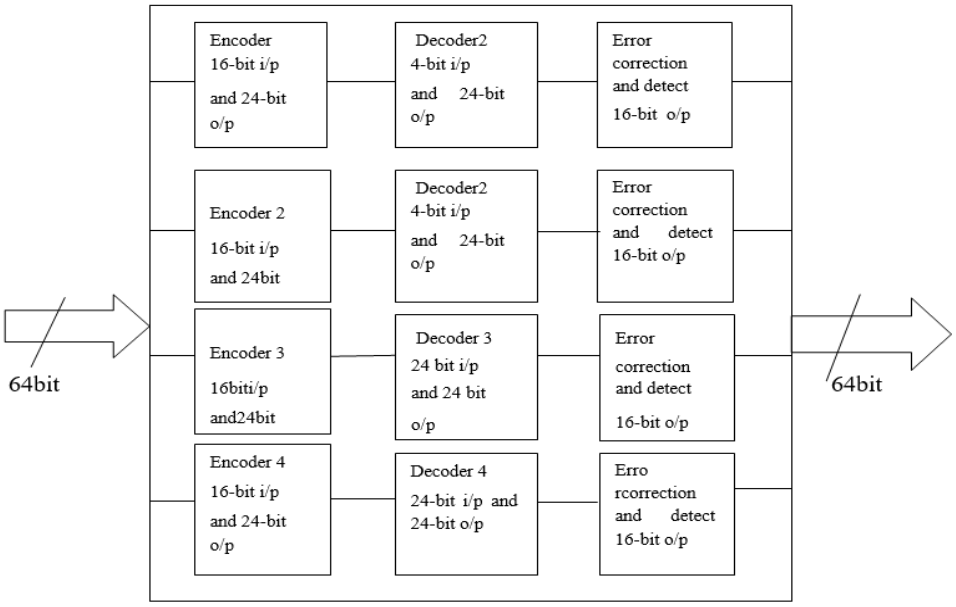

Figure 3. Schematic block diagram of the 64-bit decoding process

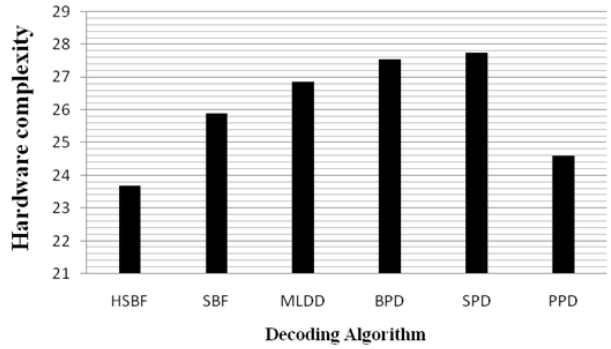

Figure 4. Hardware complexity in percentage

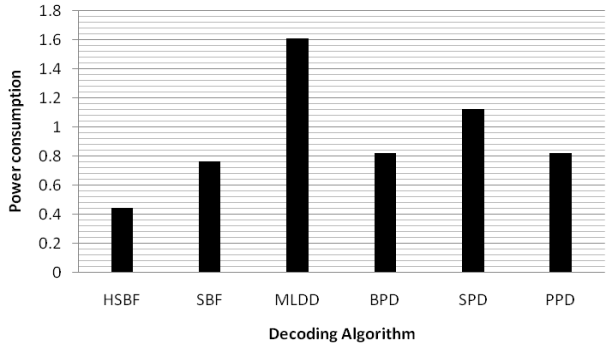

Figure 5. Power consumption of HSBF algorithm

5. CONCLUSION

The HSBF deciphering technique is introduced inside this research for EG-LDPC deciphering, which is utilized in communications, cyber security, and communication processing technologies to reduce complexity of the decoding and increase in data transmission and reception. Using the modelsim simulator and by using simulation model depicted in this paper and power usage and logical resource utilization parameters, a simulation method was provided for the estimation of throughput of hybrid SBF decoding

technique as well as SBF decoder, MLDD decoder, BPD decoder, SPD decoder, and PPD decoding techniques. The suggested approach improves performance by reducing the number of resources used when signals are transferred between transmitter and receiver. It also decreases hardware overhead, decryption delay, and power usage. Furthermore, the simulation demonstrates that perhaps the suggested hybrid SBF deciphering technique performed best over conventional bit flipping (BF) decoding algorithms to decode 64-bit data. The typical delay of 16.65% is determined to be lowered by employing the suggested decoding method, based on these simulation results. It is noticed that the resource use is decreased to an optimum of 4.25% in the considered synthesis conditions. The suggested HSBF deciphering technique uses 1.85% of the slices flip-flops resource, which is somewhat more than conventional analysis are presented. It should be mentioned that the mentioned experimental research results reduce the usage of power by a value 41.68% at an average in comparison to the different conventional methods considered in this paper. Declaration statements: no conflicts of interest.

REFERENCES

[1] A. Khalid, M. Ahmad, N. Khan, and A. Ali, "Enhancement of Visible Light Communication Transceiver using LDPC Codes," *2018 International Conference on Frontiers of Information Technology (FIT)*, 2018, pp. 94-98, doi: 10.1109/FIT.2018.00024.

[2] J. C. Babu, C. C. Reddy, and M. N. G. Prasad, "Comparison of technologies for the implementation of SBF decoder for geometric LDPC codes," *Indian Journal of Science and Technology*, vol. 9, no. 30, pp. 1-9 2016, doi: 10.17485/ijst/2016/v9i30/99033.

[3] M. N. G. Prasad, C. C. Reddy, and J. C. Babu, "VLSI implementation of decoding algorithms for EG-LDPC codes," *Procedia Computer Science*, vol. 115, pp. 143-150, 2017, doi: 10.1016/j.procs.2017.09.119.

[4] Z. B. K. Egilmez, L. Xiang, R. G. Maunder, and L. Hanzo, "The Development, Operation and Performance of the 5G Polar Codes," in *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 96-122, Firstquarter 2020, doi: 10.1109/COMST.2019.2960746.

[5] M. B. Abdessalem, A. Zribi, T. Matsumoto, E. Dupraz, and A. Bouallègue, "LDPC-based joint source channel coding and decoding strategies for single relay cooperative communications," *Physical Communication*, vol. 38, Feb. 2020, doi: 10.1016/j.phycom.2019.100947.

[6] S. Wang, Q. Huang, and Z. Wang, "Symbol Flipping Decoding Algorithms Based on Prediction for Non-Binary LDPC Codes," in *IEEE Transactions on Communications*, vol. 65, no. 5, pp. 1913-1924, May 2017, doi: 10.1109/TCOMM.2017.2677438.

[7] M. K. Roberts, S. S. Mohanram, and N. Shanmugasundaram, "An improved low complex offset min-sum based decoding algorithm for LDPC codes," *Mobile Networking and Applications*, vol. 24, no. 1, pp. 1848-1852, 2019, doi: 10.1007/s11036-019-01392-7.

[8] Q. Xie and Q. Sun, "Computation and application of robust data-driven bandwidth selection for gradient function estimation," *Applied Mathematics and Computation*, vol. 361, pp. 274-293, 2019, doi: 10.1016/j.amc.2019.05.044.

[9] J. Dai, L. Zhou, C. Chang, and W. Xu, "Robust Bayesian learning approach for massive MIMO channel estimation," *Signal Processing*, vol. 168, Mar. 2020, doi: 10.1016/j.sigpro.2019.107345.

[10] J. C. Babu, C. C. Reddy, and M. N. G. Prasad, "Comparison of various decoding algorithms for EG-low density parity check codes," *Advances in Systems, Control and Automation*, vol. 442, pp. 605-613, 2017, doi: 10.1007/978-981-10-4762-6_58.

[11] Y. Chen, H. Cui, J. Lin, and Z. Wang, "Fine-Grained Bit-Flipping Decoding for LDPC Codes," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 5, pp. 896-900, May 2020, doi: 10.1109/TCSII.2020.2980846.

[12] Z. Zhang, K. Zhang, Z. Peng, D. Zeng, and M. Dai, "Velocity Analysis of BP Decoding Waves for SC-LDPC Ensembles on BMS Channels: An Interpolation-Based Approach," in *IEEE Transactions on Communications*, vol. 68, no. 6, pp. 3286-3301, June 2020, doi: 10.1109/TCOMM.2020.2976982.

[13] Y. Delomier, B. Le Gal, J. Crenne, and C. Jego, "Model-based design of flexible and efficient LDPC decoders on FPGA devices," *Journal of Signal Processing Systems*, vol. 92, pp. 727-745, 2020, doi: 10.1007/s11265-020-01519-0.

[14] Q. Li *et al*., "Process Variation Aware Read Performance Improvement for LDPC-Based nand Flash Memory," in *IEEE Transactions on Reliability*, vol. 69, no. 1, pp. 310-321, March 2020, doi: 10.1109/TR.2019.2893287.

[15] H. Lopez, H. -W. Chan, K. -L. Chiu, P. -Y. Tsai, and S. -J. J. Jou, "A 75-Gb/s/mm2 and Energy-Efficient LDPC Decoder Based on a Reduced Complexity Second Minimum Approximation Min-Sum Algorithm," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 4, pp. 926-939, April 2020, doi: 10.1109/TVLSI.2019.2955925.

[16] S. Zhao and X. Ma, "Construction of High-Performance Array-Based Non-Binary LDPC Codes With Moderate Rates," in *IEEE Communications Letters*, vol. 20, no. 1, pp. 13-16, Jan. 2016, doi: 10.1109/LCOMM.2015.2494581.

[17] J. C. Babu, N. M. Rao, K. Ramana, and V. Bhaskar, "A dynamic hybrid decoder approach using EG-LDPC codes for signal processing applications," *Wireless Personal Communications*, vol. 122, pp. 1435-1454, 2022, doi: 10.1007/s11277-021-08956-5.

[18] Z. Xing, K. Liu, and Y. Liu, "Low-complexity companding function design for PAPR reduction in OFDM systems," *IET Communications*, vol. 14, no. 10, pp. 1581-1587, 2020, doi: 10.1049/iet-com.2019.0812.

[19] S. C. Arum, D. Grace, and P. D. Mitchell, "A review of wireless communication using high-altitude platforms for extended coverage and capacity," *Computer Communications*, vol. 157, pp. 232-256, 2020, doi: 10.1016/j.comcom.2020.04.020.

[20] F. Ye, E. Dupraz, Z. Mheich, and K. Amis, "Optimized Rate-Adaptive Protograph-Based LDPC Codes for Source Coding with Side Information," in *IEEE Transactions on Communications*, vol. 67, no. 6, pp. 3879-3889, June 2019, doi: 10.1109/TCOMM.2019.2903498.

[21] Y. Fang, G. Han, G. Cai, F. C. M. Lau, P. Chen, and Y. L. Guan, "Design Guidelines of Low-Density Parity-Check Codes for Magnetic Recording Systems," in *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1574-1606, Secondquarter 2018, doi: 10.1109/COMST.2018.2797875.

[22] K. Arora, J. Singh, and Y. S. Randhawa, "A survey on channel coding techniques for 5G wireless networks," *Telecommunication Systems*, vol. 73, pp. 637-663, 2020, doi: 10.1007/s11235-019-00630-3.

[23] S. -K. Ahn, K. -J. Kim, S. Myung, S. -I. Park, and K. Yang, "Comparison of Low-Density Parity-Check Codes in ATSC 3.0 and 5G Standards," in *IEEE Transactions on Broadcasting*, vol. 65, no. 3, pp. 489-495, Sept. 2019, doi: 10.1109/TBC.2018.2874541.

[24] M. Awais, A. Razzaq, A. Ahmed, and G. Maesera, "LDPC check node implementation using reversible logic," *IET Circuits Devices and Systems*, vol. 13, no. 4, pp. 443-455, 2019, doi: 10.1049/iet-cds.2018.5222.

[25] M. Gholami and M. Samadieh, "Design of Binary and Nonbinary Codes from Lifting of Girth-8 Cycle Codes with Minimum Lengths," in *IEEE Communications Letters*, vol. 17, no. 4, pp. 777-780, April 2013, doi: 10.1109/LCOMM.2013.020513.122820.

[26] C. Sun, H. Xu, D. Feng, and B. Bai, "(3, L) quasi-cyclic LDPC codes: Simplified exhaustive search and designs," *2016 9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, 2016, pp. 271-275, doi: 10.1109/ISTC.2016.7593119.

BIOGRAPHIES OF AUTHORS

Shaik Asif Hussain is an Assistant Professor in the Department of Electronics and communication Engineering at Middle East College. He is associated with MEC since 2015 and serves as the officer for the Technology Transfer office in center of Research and consultancy in the college. His expertise includes Artificial Intelligence, Digital signal and Image Processing, Biomedical Engineering and Embedded systems. Over the last 10 years, he has taught several topics and programmes related to Electronics and Telecommunication Engineering students in highly reputed higher education institutions in Oman, UK, and India. His research focusses on Digital Image processing, Internet of things (IoT), Design and automation, Networking, and cloud architecture. With an H-index of 6, Dr. Shaik has published 30 papers on all major research journals and has presented papers in 26 conferences across the globe. Dr. Shaik has received funding for four research projects as Principal investigator from Department of Science and Technology (DST), All India council of Technical Education (A.I.C.T. E) and the research council of Oman. Before entering academia, Dr. Shaik worked as a student advisor and technical consultant to Texas instruments in India and the UK on Analog maker designs. He is member of Indian society for Technical Education (ISTE), The Institution of Engineers (IEI), the society Association of scientists, Developers and Faculties (ASDF), Computer science Teachers Association (CSTA) and an active member of Institute of Electrical and Electronics Engineers (IEEE). Dr. Shaik has received his PhD in Electronics and Communication Engineering in 2015 from the Jawaharlal Nehru Technological University. Dr. Shaik was a recipient of the All-India council for Technical Education (A.I.C.T.E.) Young talented Researcher Award for Excellence in 2011, and presently he is acting as a member in IEEE student branch of Oman. He can be contacted at email: shussain@mec.edu.om.

Dr. Jyothi Chinna Babu has received his B. Tech. degree from JNTU, Hyderabad. M. Tech. degree in VLSI System Design and Ph.D. in VLSI signal processing degrees from JNTUA, Ananthapuramu. Currently, he is working as Associate Professor in the Department of Electronics & Communication Engineering, AITS, Rajampet, Kadapa, A.P, and India. He has published number of research papers in various International Journals and Various National and International Conferences. He has joined at AITS, Rajampet as an Assistant Professor in the Department of Electronics and Communication Engineering in the year 2008. He is the Board of Studies member of the ECE department from 2014.He is a life Member in ISTE and Member in IACSIT, MIRED and CSTA. His current research interests are VLSI Signal processing, Micro processor, Embedded Systems and Communication and Signal Processing. He can be contacted at email: jchinnababu@gmail.com.

Raza Hasan received his B.S. (Computer Science) from Khadim Ali Shah Bukhari Institute of Technology (KASBIT), Karachi, Pakistan in 2004. He received Mc. in Forensic Computing from Staffordshire University UK in 2007. Currently, he is pursuing Doctor of Philosophy in Informatics from Malaysia University of Science and Technology (MUST), Malaysia. He is working as Senior Lecturer in Department of Computing at Middle East College (MEC), Muscat, Oman. His research interests include Learning Analytics, Education Data Mining, BigData and Artificial Intelligence. He can be contacted at email: Raza@mec.edu.om.

Salman Mahmood is currently a PhD fellow in informatics at Malaysia University of Science & Technology (MUST), Malaysia. He prior to that graduated in BS computer engineering from Sir Syed University of Engineering and Technology (SSUET), Pakistan in 2002 and then went to the university of Northampton, United Kingdom (UK) to earn his MSc. degree in IT service Management in 2007. He also has done his MBA Marketing from Newports Institute of Communication and Economics (NICE) in 2012.He has been teaching as a lecturer in department of computer engineering at SSUET, Pakistan. He has been actively participating in IEEE conference and has presented on IOT,BigData,Artificial Intelligence and machine learning. He has keen interest in study about IOT, machine learning and big data. His PhD research is specifically intended to study impact of big data in higher education institute (HEI). He can be contacted at email: salman.mahmood@pg.must.edu.my.