Indonesian Journal of Electrical Engineering and Computer Science
Vol. 23, No. 2, August 2021, pp. 890~901
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v23.i2.pp890-901

Improving bit error-rate based on adaptive Bose-Chaudhuri Hocquenghem concatenated with convolutional codes

Ahmed Samy, Ashraf Y. Hassan, Hatem M. Zakaria
Electrical Engineering Department, Benha Faculty of Engineering, Benha University, Egypt

Article Info

Article history:

Received Mar 5, 2021
Revised May 23, 2021
Accepted Jun 1, 2021

Keywords:

Bose-Chaudhuri Hocquenghem
Concatenation
Convolutional
Decoder
Encoder
Viterbi

ABSTRACT

Several algorithms have been proposed to avoid the error floor region, such as the concatenation codes that requires high computational demands in addition to high complexity. This paper proposes a technique based on using cascaded BCH and convolutional codes that leads to better error correction performance. Moreover, an adaptive method based on sensing the channel's noise to determine the number of the parity bits that will be added to the used BCH that reduces the consumed bandwidth and the transmitted parity bits is presented. A further enhancement is fulfilled by using parallel processing branches, resulting in reducing the consumed time and speed up the performance. The results show that the proposed code presents a better performance. A high reduction in the number of cycles that will be used in the encoding and decoding compared with the classical method and finally a flexible parity bits method based on the signal-to-noise ratio of the channel that reduced the parity bits which leads to reduce the consumed bandwidth. The MATLAB simulation and the field programmable gate array (FPGA) implementation will be provided in this paper to validate the proposed concept.

Corresponding Author:

Ahmed Samy
Department of Electrical Engineering
Benha Faculty of Engineering, Benha University, Egypt
Email: eng.ahmed.samy46@gmail.com

1. INTRODUCTION

The communications field is witnessing a struggle to achieve better performance. The efforts to get a high throughput, low bit-error-rate (BER) and also low consumption of power are the main target of all the communication systems. In this paper, two techniques will be discussed: Bose-Chaudhuri Hocquenghem (BCH) and the convolutional codes [1]. The error floor is the phenomenon that faces the error correcting algorithms such as BCH, the BER decreases as the signal-to-noise ratio (SNR) becomes better [2]. There is a point after which the curve does not fall as quickly as before, the region where the performance flattens is called the error floor region and the region before the huge drop is called the waterfall [3], [4].

This paper will use the BCH and the convolutional codes to approve the idea of this article. The BCH features are the possibility to design code that can correct multiple errors and the easy decoding process using low-power electronic sources. The main features of using convolutional codes are the easy implementation, the better performance in the cases of having higher noisy channels and error probability rates, these codes have memory and Information bits are spread along the sequence.

Concatenated codes concatenation of an inner convolutional code with an outer block code is very common and effective coding structure for solving the error floor problem. The convolutional code works better in low SNR range, but its BER curve roll-off slowly and can have irreducible error floor in fading

channel [5], [6]. The BCH code has fast BER roll-off and well suited to correct the burst output errors common with a Viterbi decoder. An interleaver can be used to spread the Viterbi output error bursts across multiple BCH code-words for burst error correction [7], [8].

The design of concatenated code of BCH and convolutional had a problem of degradation of the throughput due to serial concatenation. Accordingly, a motivation is raised to improve the degradation of throughput of serial concatenation code, to be suited for real time application. This is done via designing parallel processing of BCH code with convolutional code [9], [10].

Another challenge that will face the designer is that how to design a transitional transmission protocol between these two codes to read and write between the encoders and the decoders correctly. Furthermore, earlier work [11], [12], the designers concatenated the codes without taking into account channel's SNR that made the system could add more parity bits than the required. In which this will increase the consumed bandwidth.

The proposed scheme provides three main goals. First proposing a concatenated method to solve the error floor problem, second, providing parallel processing of BCH code with convolutional for improving throughput. Finally, improving error correction capability. The design of parallel BCH code with convolutional is based on the time delay of a single branch of BCH code. Moreover, in this study a presentation of an adaptive method that let the system select the suitable BCH from the channel's SNR point of view.

This paper is organized as follows. Section 2 describes BCH and the convolutional coding. Section 3 presents the proposed algorithm. Section 3 discusses the experimental hardware implementation. Simulation and results are presented in section 5. Finally, section 6 concludes the proposed work.

2. BCH AND CONVOLUTION CODING SYSTEMS

2.1. BCH codes

The BCH codes, discovered in 1960 [13]–[15], is a class of cyclic codes that has well-known implementation algorithms and powerful error-correcting properties [16]. BCH encoder is based on linear feedback shift register (LFSR) as shown in Figure 1 [17]. A three types of BCH codes are used in this study. BCH(15,5,3), BCH(15,7,2) and BCH(15,11,1). Based on the SNR of the channel, the transmitter will choose one of these BCHs to use [18].

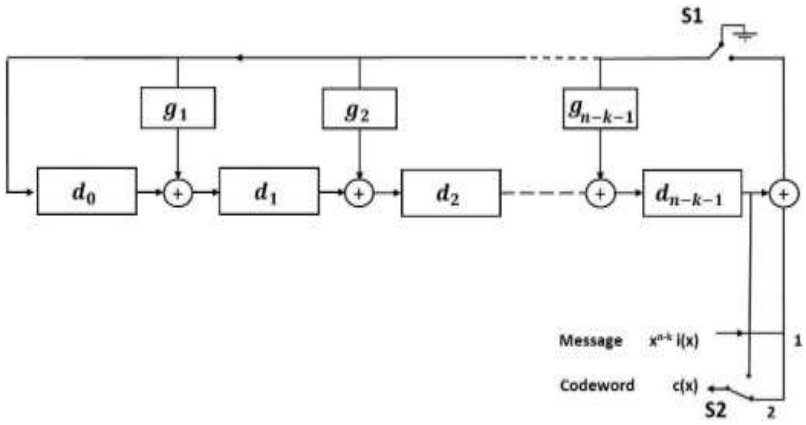

Figure 1. The encoder LFSR design [19]

The BCH decoding process is based on four main operations as shown in Figure 2. The first is determining the syndrome [20]. Second, it calculates the error location polynomial [21]. Then it finds the roots of error locator polynomial. The final one is the error correction.

2.2. The convolutional codes

An (n, k, m) convolutional code can be implemented with a k-input, n-output linear sequential circuit with input memory m [22]. Typically n and k are small integers with k<n. However. A convolutional code is generated by passing the information sequence to be transmitted through a linear finite-state shift register [23].

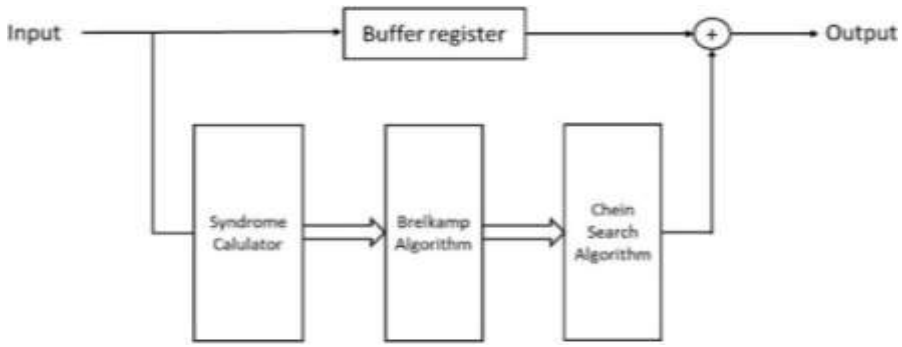

Figure 2. The BCH decoding process [24]

2.3. The Viterbi algorithm

It is based on the maximum likelihood criteria, as it computes the received sequence with every possible code sequence. The criterion for deciding between two paths is to select the one having the smallest metric. This method maximizes the probability of a correct decision [25]. The Viterbi decoder consists of four main processes: branch metric computation, state metric update, survivor path recording and output decision generation [26].

3. RESEARCH METHOD

This section is dedicated to present the proposed work to fulfill the motivated objectives. The proposed concatenated codes use parallel branches of BCH, which enhances the performance compared to the standard concatenated codes. It is efficient to use convolutional code concatenated with parallel branches of BCH code to improve the performance in the error floor region, Figure 3 shows the serial concatenated code based on hard decision [27].

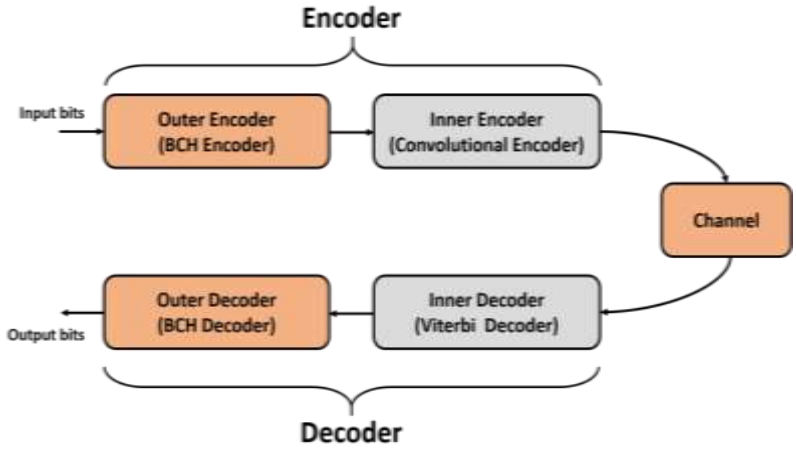

Figure 3. Single branch BCH concatenated with convolutional encoder

There are three different BCH are used in this study to reduce the parity bits and reduce the consumed bandwidth. As shown in Figure 4, the channel noise-sensing unit is added in the transceiver. In which, the transceiver will estimate the value of the SNR by transmitting and receiving specific code between the source and destination before the encoding process. Based on this process, the SNR will be estimated. This noise-sensing unit will measure the power of the received signal as the source power is known for both the source and the destination, the noise-sensing unit can estimate the SNR of the channel. This estimation process will be repeated in a predetermined user defined period; the BCH selector will choose the suitable BCH based on the estimated SNR. Furthermore, the user will determine the values' range of comparison to detect the suitable BCH.

Assume these values are A, B and C (the SNR value in dB). If the estimated SNR is less than A then the BCH detector will choose BCH (15,11,1) to make the encoding process that will add 4 parity bits. However, If the estimated SNR is more than A and less than C then the BCH detector will choose BCH (15,7,2) to make the encoding process that will add 8 parity bits. Finally, If the estimated SNR is more than C then the BCH detector will choose BCH (15,5,3) to make the encoding process that will add 10 parity bits. This SNR estimating process and choosing BCH depending on the SNR is a proposed adaptive method, which will improve the performance and reduce the bit error rate.

There is an interleaver between the outer encoder and the inner encoder to correct burst errors (Errors are depended on each other). The de-interleaver will be found between the outer decoder and the inner decoder. As shown in Figure 4, the BCH detector will determine the used BCH in the encoding process at the transmitter. At the same time there is a BCH detector in the receiver that determine the same BCH decoder to perform the decoding process. The BCH detector will send to the interleaver to receive form the used BCH and neglect any random bits from the other BCHs. The BCH detector at the receiver will send to the de-interleaver for sending the first stage-decoded bits to the suitable BCH decoder after the de-interleaver process.

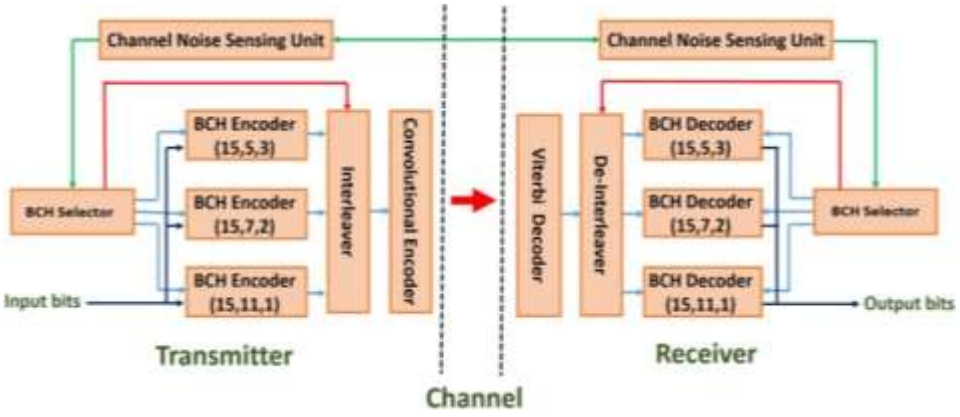

Figure 4. The proposed methodology

The proposed setup will have made the bits transfer management is critical issue. Once the BCH receive the input bits, it will start the encoding process. However, during this encoding process, there are other bits are being inputted so these bits must be stored and after the BCH finish the required process, it will read the next bits, this process need big memory and has high time delay. Therefore, the proposed parallel branches method is a suitable solution for this issue. The proposed method based on using multiple branches for each BCHs, As, some of the BCH reads, while the others perform the encoding process. The proposed steps will be described in the next paragraphs.

As shown in Figure 5, the BCH (15,11,1) encoder concatenated with convolutional encoder, the system will use two parallel branches because the encoder consumes 11 cycles to read the bits and 4 cycles to perform the encoding process. So, at the same time of the encoding process for the first BCH, there will be another BCH that reads the input data. After 15 cycles, the first BCH will send the bits to the convolutional encoder which will consume 7 cycles for the processing. These 7 cycles in addition to the 4 cycles of the processing of the first BCH will be 11 cycles that will be used in the second BCH to read the input bits. Therefore, once the convolutional encoder finishes its work, the second BCH will start the processing and the first BCH starts to read the input.

As shown in Figure 6, the BCH (15,7,2) encoder concatenated with convolutional encoder, the system will use three parallel branches because the encoder consumes 7 cycles to read the bits and 8 cycles to perform the encoding process. So, at the same time of the encoding process for the first BCH, there will be another BCH that reads the input data. So, after 15 cycles, the first BCH will send the bits to the convolutional encoder which will consume 7 cycles for the processing. These 7 cycles in addition to the 8 cycles of the processing of the first BCH will be 15 cycles that will be used in the second BCH to read the input bits and perform the encoding. As noticed, the second BCH start the encoding process before the first BCH finish the encoding process by one cycle so the third BCH branch will start reading the input bits while the first BCH and the second BCH are performing the encoding process. Once the third BCH start the processing, the first BCH will start the reading.

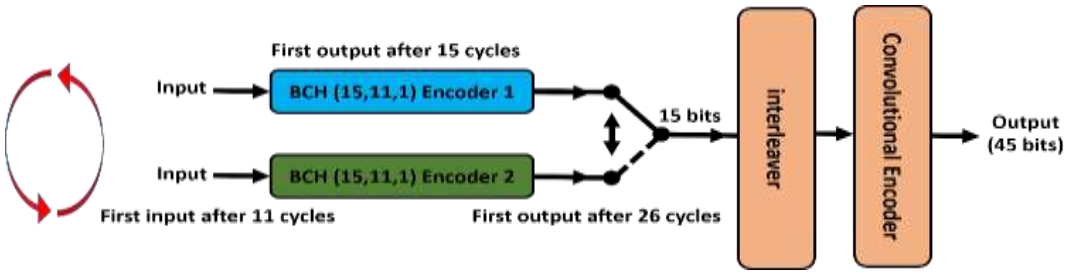

Figure 5. Parallel BCH (15,11,1) branches concatenated with convolutional encoder

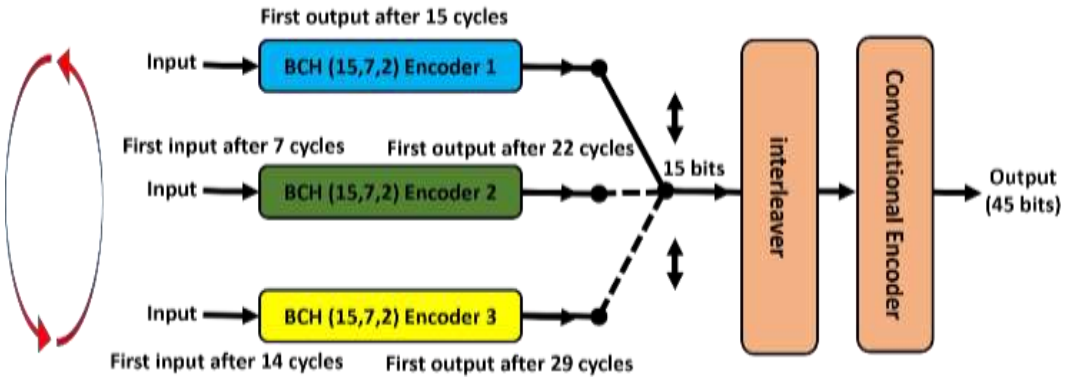

Figure 6. Parallel BCH (15,7,2) branches concatenated with convolutional encoder

As shown in Figure 7. The BCH (15,5,3) encoder concatenated with convolutional encoder, the system will use three parallel branches because the encoder consumes 5 cycles to read the bits and 10 cycles to perform the encoding process. So, at the same time of the encoding process for the first BCH, the second BCH is reading the input data. Therefore, after 10 cycles, the first BCH is performing the processing, the second BCH start the processing, and the third BCH starts reading the input bits. After 15 cycles, the first bch will send the bits to the convolutional encoder, which will consume 7 cycles for the processing.

As noticed, the second BCH start the encoding process before the first BCH finishes the encoding process by 5 cycles so the third BCH starts the reading of the input bits. After 5 cycles the third BCH starts the encoding process before the second BCH finishes the process by 5 cycles but at the same time, the first BCH will finish the processing so it starts the reading of the input bits so the first BCH branch will start reading the input bits while the second BCH and the third BCH are performing the encoding process. The overall code rate of the proposed algorithm will be (45,5,3) or (45,7,2) or (45,11,1). As seen, the proposed algorithm will use high channel bandwidth and consume more hardware resources to achieve the required results and this is a logic results as a payment corresponded to the improved results.

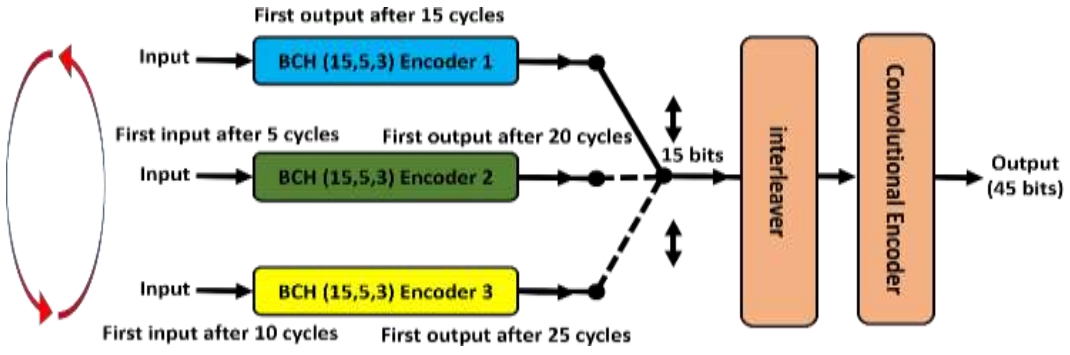

Figure 7. Parallel BCH (15,5,3) branches concatenated with convolutional encoder

It is clear that the proposed parallel branch schemes save 11 cycles in each generated codeword for the BCH (15,11,1), save 7 cycles in each generated codeword for the BCH (15,7,2) and save 5 cycles in each generated codeword for the BCH (15,5,3) than the standard single branch scheme, which achieves higher throughput than the traditional one. The same technique will be used in the decoding process but the Viterbi decoder will perform the first decoding stage and then the BCH decoder will be perform the second decoding stage. The input bits for the Viterbi decoder are 45 bits because we use convolutional encoder with rate 1/3 so each bit form the output BCH encoder will leads to 3 bits from the convolutional encoder. The input bits for the Viterbi decoder will be received in parallel form as each package consists of 9 bits so the Viterbi decoder will consume 5 cycles for reading in addition to 7 cycles for the processing so the total number of cycles are 12 cycles . The three types of BHC decoder will consume 17, 18 and 19 cycles (15 cycles for reading bits, 2 cycles for (15,11,1) decoding, 3 cycles for (15,7,2) decoding and 4 cycles for (15,5,3) decoding) so the system will use two BCH decoders from each other so as there is one BCH is performing the decoding process, there is also another BCH is reading from the Viterbi decoder. Figure 8 shows a Parallel BCH decoder branches concatenated with Viterbi decoder.

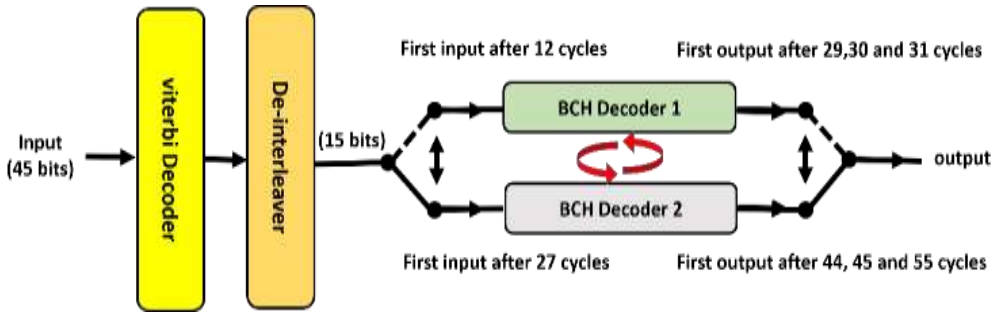

Figure 8. Parallel BCH decoder branches concatenated with Viterbi decoder

4. HARDWARE IMPLEMENTATION

The FPGA implementation of the proposed method is presented in this section. In the present work, the Xilinx Spartan-3A/3AN FPGA Starter Kit 700K-gate XC3S700A is shown in Figure 9. This kit contains 4-Mbit flash PROM, 512 Mbit DDR2 SDRAM, 50 MHz clock oscillator, 6-data channels, SPI based digital to analog converter and SPI based analog to digital converter, 8 LEDs, and 4 slide switches and 4 push buttons.

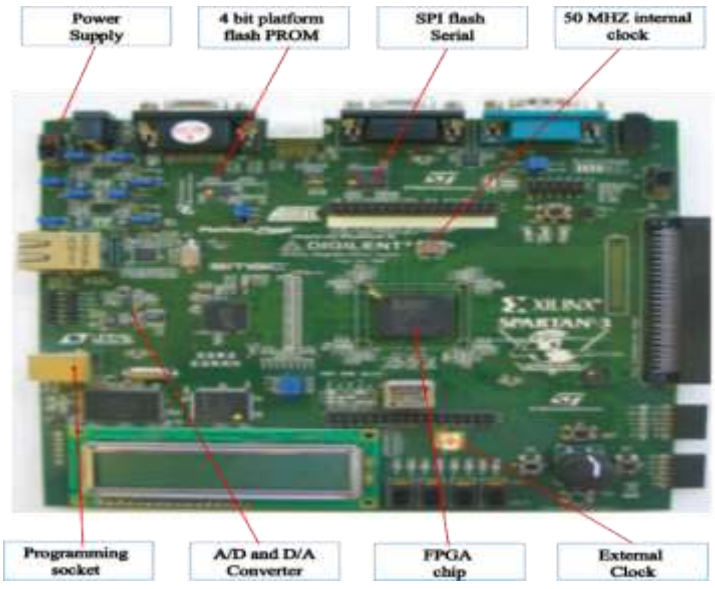

Figure 9. Xilinx Spartan-3A/3AN FPGA Starter Kit 700K-gate XC3S700A

The method is implemented by using VHSIC Hardware Description Language (VHDL) code and the Xilinx package ISE14.7, the simulation was performed using ISim program. As shown in Figure 10, the proposed encoding system consists of SNR sensing unit, two parallel branches of BCH(15,11,1) encoder, three parallel branches of BCH (15,7,2), three parallel branches of BCH (15,5,3) encoder, the interleaver and the Convolutional encoder. There is a frequency down conversion that perform the synchronization between the BCHs and the convolutional encoder. Each schematic symbol represents VHDL code.

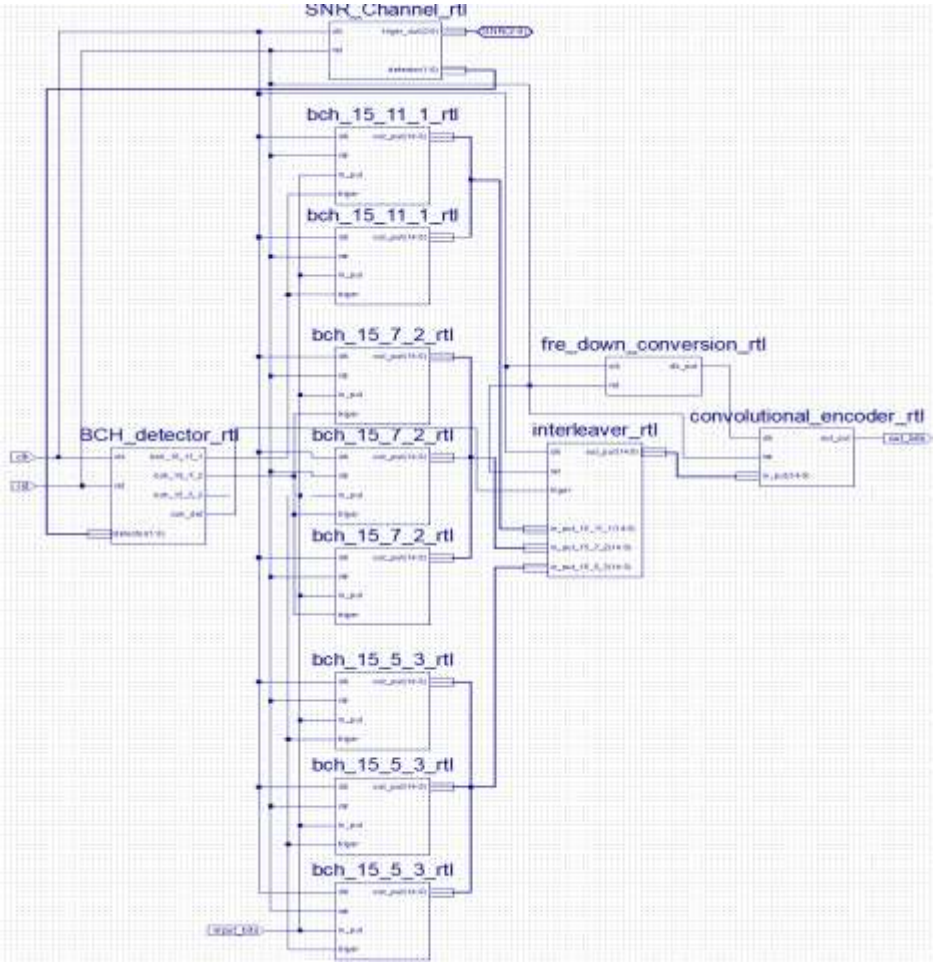

Figure 10. The proposed encoder implementation

As discussed before, the SNR_Channel block will estimate the SNR of the channel, based on this value the BCH_Detector block will determine the used BCH, BCH_15_11_1 block, BCH_15_7_2 or BCH_15_5_3 and also it will send to the interleaver to receive from the suitable BCH and neglect any random values. The interleaver will receive 15 bits from the BCH and then send these bits to convolutional encoder that perform the sensing encoding stage.

As shown in Figure 11, the proposed decoding system consists of SNR sensing unit, two parallel branches of BCH(15,11,1) decoder, two parallel branches of BCH (15,7,2) decoder, two parallel branches of BCH (15,5,3) decoder, the de-interleaver and the Viterbi decoder. There is a frequency down conversion that perform the synchronization between the BCHs and the Viterbi encoder. As discussed before, the SNR_Channel block will estimate the SNR of the channel, based on this value the BCH_Detector block will determine the used BCH, BCH_15_11_1 block, BCH_15_7_2 or BCH_15_5_3 and also it will send to the de-interleaver to send to the suitable BCH. The de-interleaver will receive from the Viterbi decoder and then send these bits to BCH decoder that perform the sensing decoding stage. As shown in Figure 11, the Viterbi decoder will perform the first stage of decoding as it is the inner stage but the BCH will perform the second stage of decoding as it is the outer coding system.

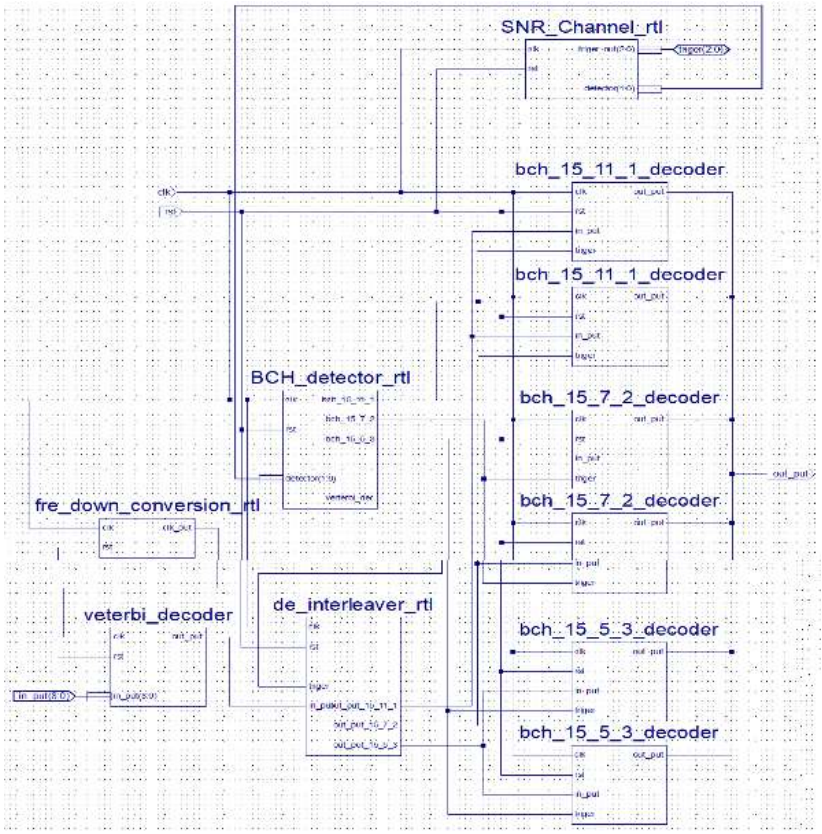

Figure 11. The proposed decoder implementation

5. RESULTS AND DISCUSSION

BER is a main parameter that can be used to evaluate the performance. The BER of the proposed method has been simulated by using a the Matlab package. The comparison between the existing channel code and the proposed code are shown in Figures 12 and 13. It can be observed that the proposed code shows a better performance compared by the BCH code. At SNR=10 dB, the BER steadily decreases in the form of a curve as the SNR condition becomes better and the curve does not fall as quickly as before, in other words, there is a region in which performance flattens. This region is called the error floor region. This issue was solved by using the proposed system.

By comparing Figures 12 and 13, It can be observed that the proposed code shows a better performance compared to those of BCH and the BER decreases in the form of a curve as the SNR condition becomes better till it reaches the minimum value at SNR=6 dB.

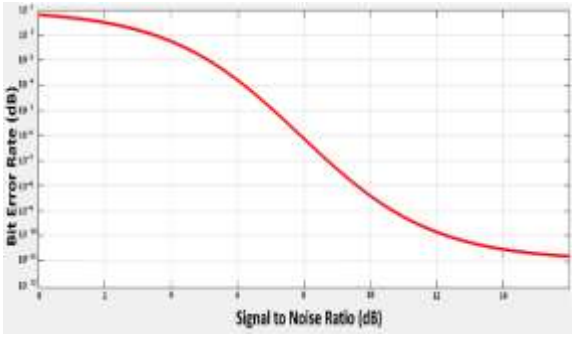

Figure 12. SNR Vs BER of the BCH

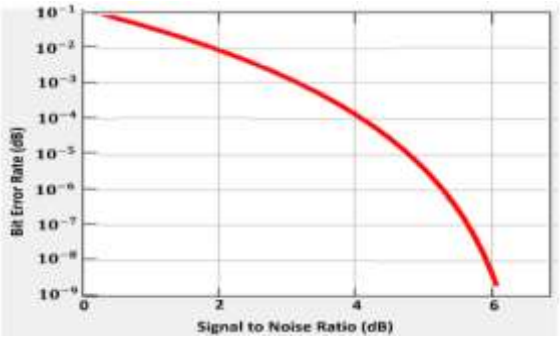

Figure 13. SNR Vs BER of the proposed concatenated system

Figures 14 to 16 show the ISim simulation of the encoding stage. Based on the SNR of the channel that will be estimated by using the proposed adaptive technique. The BCH detector will choose the BCH (15,11,1) encoder, the BCH (15,7,2) encoder or the BCH (15,5,3) encoder to perform the first encoding stage. The CLK signal is used to provide the synchronization process and the R-S-T signal is used to return the system parameters to its initial values. Figures 17 to 19 show the ISim simulation of the decoding stage in the case of BCH (15,11,1), BCH (15,7,2) and BCH (15,5,3).

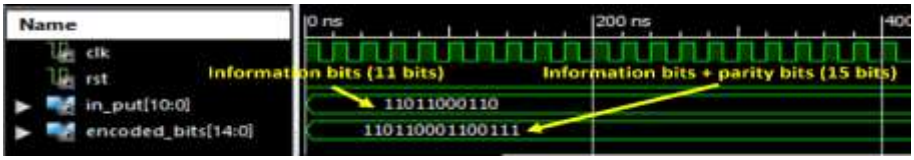

Figure 14. ISim simulation of the BCH (15,11,1) encoding

Figure 15. ISim simulation of the BCH (15,7,2) encoding

Figure 16. ISim simulation of the BCH (15,5,3) encoding

Figure 17. ISim simulation of the BCH (15,11,1) decoding

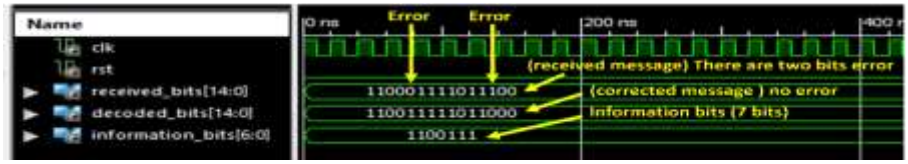

Figure 18. ISim simulation of the BCH (15,7,2) decoding

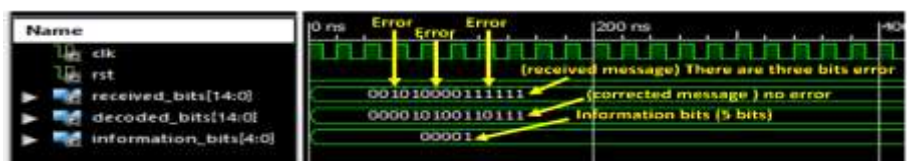

Figure 19. ISim simulation of the BCH (15,5,3) decoding

Figure 20 shows the ISim simulation of the encoding for the overall proposed system, the simulated case when the used BCH encoder is BCH(15,,5,3), the encoded bits are 45 as the used convolutional encoder is 1/3 that generate three bits for each one input bits so the input bits for the BCH is 5 bits that will leads to 15 bits at the BCH output which will be the input of the convolutional encoder that leads to 45 encoded bits.

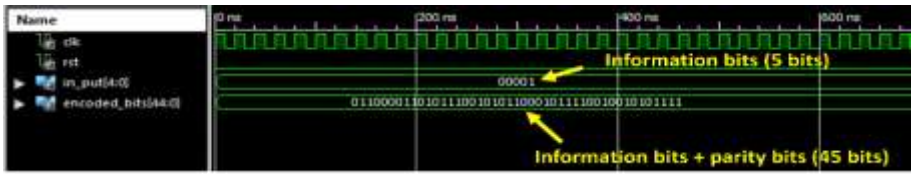

Figure 20. ISim simulation of the proposed encoding

Figure 21 shows the ISim simulation of the decoding for the previous case. The received message consists of 45 bits, there are three bits error that were detected and corrected. The information bits after the parity bits were removed are 5 bits.

Figure 21. ISim simulation of the proposed decoding

As shown in Table 1, for the serial concatenation, all BCHs will consume the same numbers of cycles as the encoder consume 15 cycles for reading bits and encoding. In the case of the concatenation, these numbers of cycles were reduced as in the case of BCH(15,11,1), the proposed method will saves 11 cycles in the encoding for the one codeword and saves 15 cycles in the decoding for the one codeword that will increase the system throughput. The proposed method also will saves 7 cycles in the encoding and saves 15 cycles in the decoding for the one codeword in the case of BCH(15,7,2) and finally, it saves 5 cycles in the encoding and saves 15 cycles in the decoding for the one codeword in the case of BCH(15,5,3). As shown in Table 2, the serial design consumed 13% of the used kit resources, while the parallel branches design consumed 47%. The performance improved at the expense of the hardware resources.

Table 1. The cycles consumption of the serial and parallel concatenation

	Serial concatenation time (Cycle)		Parallel concatenation time (Cycle)	
	Encoder	Decoder	Encoder	Decoder
BCH(15,11,1)	22	27	11	12
BCH(15,7,2)	22	27	15	12
BCH(15,5,3)	22	27	17	12

Table 2. The device utilization summary

Logic utilization	Serial concatenation	Parallel concatenation
Slice Flip Flops	952	5869
Occupied Slices	1586	5120
Total Number of 4 input LUTs	1235	4210

6. CONCLUSION

The proposed scheme provides three main goals. It proposes a concatenated method to solve the error floor problem, the paper succeed to improve the performance of the system at the error floor region. It also provides parallel processing of BCH code with convolutional to improve throughput besides improving error correction capability. The design of parallel BCH code with convolutional is based on the time delay of

a single branch of BCH code. The simulation results prove that the proposed methods reduced the number of cycles that was used in the encoding and decoding in the classical processing. The paper also presents an adaptive method that makes the system choose the suitable BCH depending on the SNR of the channel that reduced the parity bits which leads to reduce the consumed bandwidth. The implementation and the simulations are provided to verify the goal of this paper and finally a real case study was presented to show the difference between the classical processing and the proposed one.

REFERENCES

[1] F. Li and X. Sun, "The Hermitian Dual Containing Non-Primitive BCH Codes," *IEEE Communications Letters*, vol. 25, no. 2, pp. 379-382, Feb. 2021, doi: 10.1109/LCOMM.2020.3032731.

[2] Z. Xie and X. Zhang, "Miscorrection Mitigation for Generalized Integrated Interleaved BCH Codes," *IEEE Communications Letters*, vol. 25, no. 7, pp. 2118-2122, July 2021, doi: 10.1109/LCOMM.2021.3074461.

[3] C. Gan, C. Li, and H. Qian, "Parameters of Hulls of Primitive BCH Codes of Length $q^3 - 1$," *IEEE Communications Letters*, vol. 25, no. 4, pp. 1070-1073, April 2021, doi: 10.1109/LCOMM.2020.3043448.

[4] X. Li and Q. Yue, "A New Family of Quantum Synchronizable Codes," *IEEE Communications Letters*, vol. 25, no. 2, pp. 342-345, Feb. 2021, doi: 10.1109/LCOMM.2020.3027803.

[5] C. Tang and C. Ding, "An Infinite Family of Linear Codes Supporting 4-Designs," *IEEE Transactions on Information Theory*, vol. 67, no. 1, pp. 244-254, Jan. 2021, doi: 10.1109/TIT.2020.3032600.

[6] Z. Xie and X. Zhang, "Fast Nested Key Equation Solvers for Generalized Integrated Interleaved Decoder," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 1, pp. 483-495, Jan. 2021, doi: 10.1109/TCSI.2020.3025847.

[7] H. -C. Lee, "An Efficient BCH Decoder for WBAN Applications," *IEEE Communications Letters*, vol. 25, no. 6, pp. 1766-1770, June 2021, doi: 10.1109/LCOMM.2021.3066246.

[8] A. Mondal and S. S. Garani, "Efficient Hardware Architectures for 2-D BCH Codes in the Frequency Domain for Two-Dimensional Data Storage Applications," *IEEE Transactions on Magnetics*, vol. 57, no. 5, pp. 1-14, May 2021, Art no. 3101214, doi: 10.1109/TMAG.2021.3060807.

[9] M. V. Burnashev, "The "Nontriviality" of the Union Bound for Decoding of Convolutional Codes," *IEEE Communications Letters*, vol. 25, no. 2, pp. 446-449, Feb. 2021, doi: 10.1109/LCOMM.2020.3030626.

[10] F. Cheng, "Asymptotically Good Codes Obtained by an Extended Primitive BCH Code Concatenated with Interleaved Codes," in *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, 2019, pp. 417-421, doi: 10.1109/ITAIC.2019.8785548.

[11] S. D. Potey and P. M. Dhande, "Error Detection and Correction Capability for BCH Encoder using VHDL," in *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, 2019, pp. 1-4, doi: 10.1109/I2CT45611.2019.9033847.

[12] A. M. Mahdy, M. Abdelaziz and M. H. A. El-Azeem, "Design and Simulation of Parallel BCH Code with LDPC Code for Flash Memories," in *2020 12th International Conference on Electrical Engineering (ICEENG)*, 2020, pp. 196-199, doi: 10.1109/ICEENG45378.2020.9171743.

[13] C. Li, P. Wu, and F. Liu, "On Two Classes of Primitive BCH Codes and Some Related Codes," *IEEE Transactions on Information Theory*, vol. 65, no. 6, pp. 3830-3840, June 2019, doi: 10.1109/TIT.2018.2883615.

[14] Z. Xie and X. Zhang, "Reduced-Complexity Key Equation Solvers for Generalized Integrated Interleaved BCH Decoders," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 12, pp. 5520-5529, Dec. 2020, doi: 10.1109/TCSI.2020.2999823.

[15] Z. Du, C. Li and S. Mesnager, "Constructions of Self-Orthogonal Codes from Hulls of BCH Codes and Their Parameters," *IEEE Transactions on Information Theory*, vol. 66, no. 11, pp. 6774-6785, Nov. 2020, doi: 10.1109/TIT.2020.2991635.

[16] A. Mondal and S. S. Garani, "Efficient Hardware Design Architectures for BCH Product Codes in the Frequency Domain," in *2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2020, pp. 703-706, doi: 10.1109/MWSCAS48704.2020.9184633.

[17] M. Rezaei and M. A. Attari, "Blind Detection of BCH Length from Noisy Received Vectors," in *2020 28th Iranian Conference on Electrical Engineering (ICEE)*, 2020, pp. 1-5, doi: 10.1109/ICEE50131.2020.9260791

[18] S. Jihwan and H. Lee, "Burst Error Correction for Convolutional Code Concatenated with Hamming code with a block interleaver," in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2020, pp. 531-533, doi: 10.1109/ICAIIC48513.2020.9065198.

[19] L. Mu, "Ensemble of high performance structured binary convolutional LDPC codes with moderate rates," *China Communications*, vol. 17, no. 10, pp. 195-205, Oct. 2020, doi: 10.23919/JCC.2020.10.014.

[20] P. J. Almeida and J. Lieb, "Complete j-MDP Convolutional Codes," *IEEE Transactions on Information Theory*, vol. 66, no. 12, pp. 7348-7359, Dec. 2020, doi: 10.1109/TIT.2020.3015698.

[21] S. Li, J. Yuan, B. Bai, and N. Benvenuto, "Code-Based Channel Shortening for Faster-Than-Nyquist Signaling: Reduced-Complexity Detection and Code Design," *IEEE Transactions on Communications*, vol. 68, no. 7, pp. 3996-4011, July 2020, doi: 10.1109/TCOMM.2020.2988922.

[22] J. Ye, T. -Y. Wu, J. Xing, and L. Chen, "Iterative Soft Decoding of Reed-Solomon Tail-Biting Convolutional Concatenated Codes," in *2020 International Conference on Wireless Communications and Signal Processing (WCSP)*, 2020, pp. 957-961, doi: 10.1109/WCSP49889.2020.9299744.

[23] Y. Ding, Z. Huang, and J. Zhou, "An Improved Blind Recognition Method for Synchronization Position and Coding Parameters of k/n Rate Convolutional Codes in a Noisy Environment," *IEEE Access*, vol. 8, pp. 171305-171315, 2020, doi: 10.1109/ACCESS.2020.3025177.
[24] D. Spasov, "Decoding of LTE Turbo Codes Initialized with the Two Recursive Convolutional Codes," in *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, 2020, pp. 393-396, doi: 10.23919/MIPRO48935.2020.924528.
[25] C. Ma, D. Liu, X. Peng, L. Li and F. Wu, "Convolutional Neural Network-Based Arithmetic Coding for HEVC Intra-Predicted Residues," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 1901-1916, July 2020, doi: 10.1109/TCSVT.2019.2927027.
[26] U. Martínez-Peñas and D. Napp, "Locally Repairable Convolutional Codes with Sliding Window Repair," *IEEE Transactions on Information Theory*, vol. 66, no. 8, pp. 4935-4947, Aug. 2020, doi: 10.1109/TIT.2020.2977638.
[27] T. Venugopal and S. Radhika, "A Survey on Channel Coding in Wireless Networks," in *2020 International Conference on Communication and Signal Processing (ICCSP)*, 2020, pp. 0784-0789, doi: 10.1109/ICCSP48568.2020.9182213.

BIOGRAPHIES OF AUTHORS

Ahmed Samy received the B.Sc. degree of Electronics and Communications in 2009. He received master's degree in Electronics and Communications in 2015 from Benha Faculty of Engineering, Benha University, Egypt. He is working towards Ph.D. degree at Benha Faculty of Engineering at Electronics and Communications Department.
Email: eng.ahmed.samy46@gmail.com.

Ashraf Y. Hassan received the B.Sc. degree (with honour) and the M.S. degree in Electrical Engineering from Benha University, Benha, Egypt, in 2000 and 2004, respectively, and the PhD degree from Cairo University, Cairo, in 2010. From 2000 to 2010, he served as a research and teaching assistant. In 2010, he employed as an assistant professor. He works nine years as a researcher in the research and development centre in Egyptian Telephone Company from 2000 to 2009. From 2012 to 2015, he works as a visiting assistant professor at Faculty of Engineering, Northern Border University, Saudi Arabia. In 2017, he was promoted to associated professor degree. Now he was the head of the Electrical Engineering Department. Email: Ashraf.fahmy@bhit.bu.edu.eg.

Hatem M. Radwan received the B.Sc. degree (with honour) and the M.S. degree in Electrical Engineering from Benha University, Benha, Egypt, in 2002 and 2006, respectively, and the Ph.D. degree in Electrical Engineering from Université de Grenoble in 2011. He served as a research and teaching assistant at the Electrical Technology Department in Benha Faculty of Engineering. In 2011, he employed as an assistant professor in electronics and communication engineering in the Electrical Engineering Department. Email: Hatem.Radwan@bhit.bu.edu.eg.