International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 6, December 2021, pp. 4774~4784
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i6.pp4774-4784

An improvement and a fast DSP implementation of the bit flipping algorithms for low density parity check decoder

Mouhcine Razi[1], Mhammed Benhayoun[2], Anass Mansouri[3], Ali Ahaitouf[4]
[1,2,4]SIGER Laboratory, Faculty of Sciences and technology, Sidi Mohammed Ben Abdellah University, Fez, Morocco
[3]SIGER Laboratory, National School of Applied Sciences, Sidi Mohammed Ben Abdellah University, Fez, Morocco

Article Info

Article history:

Received Oct 12, 2020
Revised Feb 23, 2021
Accepted Apr 26, 2021

Keywords:

Bit Flipping
DSP implementation
LDPC decoders
WiMAX

ABSTRACT

For low density parity check (LDPC) decoding, hard-decision algorithms are sometimes more suitable than the soft-decision ones. Particularly in the high throughput and high-speed applications. However, there exists a considerable gap in performances between these two classes of algorithms in favor of soft-decision algorithms. In order to reduce this gap, in this work we introduce two new improved versions of the hard-decision algorithms, the adaptative gradient descent bit-flipping (AGDBF) and adaptative reliability ratio weighted GDBF (ARRWGDBF). An adaptative weighting and correction factor is introduced in each case to improve the performances of the two algorithms allowing an important gain of bit error rate. As a second contribution of this work a real time implementation of the proposed solutions on a digital signal processor (DSP) is performed in order to optimize and improve the performance of these new approchs. The results of numerical simulations and DSP implementation reveal a faster convergence with a low processing time and a reduction in consumed memory resources when compared to soft-decision algorithms. For the irregular LDPC code, our approachs achieves gains of 0.25 and 0.15 dB respectively for the AGDBF and ARRWGDBF algorithms.

Corresponding Author:

Mouhcine Razi
SIGER Laboratory, Faculty of Sciences and Technology
Sidi Mohammed Ben Abdellah University
P.O. Box 2202 Fez, Morocco
Email: mouhcine.razi@usmba.ac.ma

1. INTRODUCTION

In digital transmissions, there is an extraordinary rise in throughput demand in order to respond the various multimedia uses increasingly favoered by users who want universal connections. More for used mobile radio systems, the information is frequently disturbed by noise in the transmission channel. Thus, a high-performance error-correcting code is essential and obviously vital for digital transmissions making a development of high-performance decoders with low latency, high working frequency and high throughput a challenged research problem [1], [2].

Low density parity check (LDPC) codes, introduced in 1962 by Gallager [3], have an important correction power that makes them very attractive for use on highly disturbed channels. Due to their capacity to error correction performances, the LDPC codes are widely used in many communication systems and standards, such digital video broadcasting–satellite–second generation (DVB-S2), IEEE 802.16e (WiMAX), IEEE 802.11n (Wi-Fi), and 5G [4]-[6]. LDPC codes use a binary sparse parity check matrix H and for their, decoding procedure, the hard-decision and soft-decision algorithms are the two main usually used algorithms.

The soft-decision algorithms calculate the extrinsic log likelihood ratio (LLRs) to evaluate the reliability of received messages (y_n), these methods achieve the best bit error rate (BER) performances [7]-[9], but these iterative decoding algorithms require large number of arithmetic operations; and can introduce prohibitive delays for very high-speed transmissions where latency plays an important role. Alternatively, for hard-decision algorithms (or bit flipping algorithms) the decoding time can be strongly reduced with some relative loss in performances [10]. Even that, they still helpful for some applications where the speed and high throughput are needed, more particularly if one finds a way to reduce the performances gap between the two classes of algorithms [11]. These hard-decision algorithms have been introduced to address three important problems: BER performance, latency issues, and computational complexity. These issues can be seen as a trade-off problem where the challenge is to optimize all of them under specific requirements. This type of algorithm propsed by Gallager [3] simplifies the decoding method by taking a hard-decision (x_n) on the message received from the transmission channel (y_n) at the beginning of the decoding process. It first calculates the sum of the syndromes $\sum_{i\in M(n)}\prod_{j\in N(i)} x_j$, if this sum is equal to the number of lines in the matrix H, it stops the decoding, otherwise it calculates the inversion function $\Delta k(x) \triangleq \sum_{i\in M(n)}\prod_{j\in N(i)} x_j$ that estimates the reliability of received channel messages; and the bit which corresponds to the minimum of this function will be switched. This bit flipping (BF) algorithm has very low complexity since it requires, in each iteration, only a simple summation over binary parity-check values for each bit. However, this method provides poor decoding performance, for instance three or more orders when compared of the soft-decision algorithm for an SNR of 3.5 dB. To overcome these problems, the hard-decision algorithms have been largely investigated and numerous variants of BF algorithms has been proposed. Among which the weighted BF (WBF) algorithm [12], the modified weighted BF (MWBF) algorithm [13], gradient descent bit-flipping GDBF [11] and reliability ratio weighted GDBF (RRWGDBF) [14]. These works use an additive or multiplicative weighting factors in $\Delta k(x)$ to evaluate the reliability of syndromes [15], [16].

Kou *et al*. [12] proposed the WBF algorithm in which a weighting factor based on the minimum value of y_n is considered in the syndromes calculation to make $\Delta k(x)$ more reliable. This process increases the complexity and the number of iterations even if some improvements in performances, in term of BER, have been achieved. A modified version (MWBF) has been introduced by Zhang *et al*. [13] who added an offset value, based on the absolute value y_n, in $\Delta k(x)$ of the WBF algorithm. This algorithm, even it leads to some signal quality enhancement it involves a slight increase in complexity of $\Delta k(x)$. Another improved MWBF (IMWBF) version was introduced by Jiang *et al*. [15] which offered further improvement by using a weighting factor aiming to avoid the SNR dependency. This new weighting factor can be determinated via Monte Carlo simulations. Always in order to enhance the error rate performances, Wadayama *et al*. [11] suggested the concurrent GDBF algorithm as a gradient-descent optimization model for the maximum likelihood decoding problem. This algorithm adds in $\Delta k(x)$ a relation between the message after the hard decision and received channel message allowing a maximum of correlation, then searches for the minimum value of $\Delta k(x)$, and finally flips the corresponding bits. To improve the decoding performance of this GDBF algorithm another version called reliability-ratio weighted GDBF (RRWGDBF) algorithm has been proposed by Phromsa-ard *et al*. [14] that uses a weighted summation over syndrome components with an adaptive threshold to obtain reduced latency. The GDBF and RRWGDBF algorithms are methods that gives better trade-off between performance and complexity among all hard-decision algorithms [11], [14], [17], but when compared, for instance, to the min sum (MS) algorithm, which is a soft-decision algorithm, these two algorithms show relatively limited performances in term of BER [18]. Their main advantage is the simplicity of their hardware implementation compared to the MS algorithm for instance, which needs high material resources and increases the decoding latency. Several researchers proposed alternative GDBF algorithms to improve quality, but these algorithms require more than hundred iterations to converge toward best performances. Even that, the GDBF algorithm outperformed the WBF and MWBF algorithms in error correcting ability and more significantly in the average number of iterations. Nevertheless, during the decoding with GDBF algorithm, there is a risk of flipping some correct bits and again flipping them at another times in the next iterations, which causes a performance degradation with additional delays.

Thus, in this work we propose the adaptative GDBF (AGDBF) algorithm where a solution to solve this problem is developped. By following the bits flipping procedure, when a twice flipped bit is detected, we stop the flipping of this bit by adding a multiplicative weighting coefficient α. In the same framework of the decoding improvement, we also propose an adaptative RRWGDBF (ARRWGDBF) algorithm, this time by first using a pre-processing step to check the columns of the short cycles in the H matrix and finally using a weighting correction factor to eliminate the impact of these short cycles. By theses ways, these algorithms allow better performances as hard-decision algorithms making them useful for high-speed applications. After being validated by simulations, the proposed algorithms are hardware implementation on a digital signal processors (DSP) platform in order to improve their performances and to reduce the processing time of these new approchs, as a second contribution of this work. The rest of the paper is structured as follows, an

overview of the approaches of the GDBF and RRWGDBF algorithms is presented in the section 2 that permit a hardware implementation of the simplified LDPC decoder. In the next section we announce our new approach for these two algorithms. Finally, the DSP implementation results will be presented.

2. RESEARCH METHOD

2.1. Decoding algorithms

The LDPC codes use a binary sparse $m \times n$ parity check matrix H, where $m=n-k$, k being the information length and n the code length. The H matrix can be represented by a conventional Tanner graph as illustrated in Figure 1, where m represents the check nodes (CNs), and n represent the variable nodes (VNs). Each variable node v_i is connected to a set of check nodes and each check node c_i is connected to a set of variable nodes. M(n) denotes the set of check nodes connected to an involved n^{th} variable node and N(m) the set of variable nodes that participate in m^{th} check node.

For the communication systems and the standards, it is known that an optimized irregular LDPC code has better performance than a regular LDPC code [19], besides, the quasi cyclic LDPC (QC-LDPC) codes showed good performances for large codeword length [20]-[22]. The decoding complexity is proportional to E/Rn, where E is the number of links between the check nodes and the variable nodes, and R=k/n is the code rate [23]. For the present work, we assume an additive white Gaussian noise (AWGN) channel with a variance $\sigma^2=N_0/2$, where N_0 is the spectral power density, and binary phase-shift keying (BPSK) modulation [24].

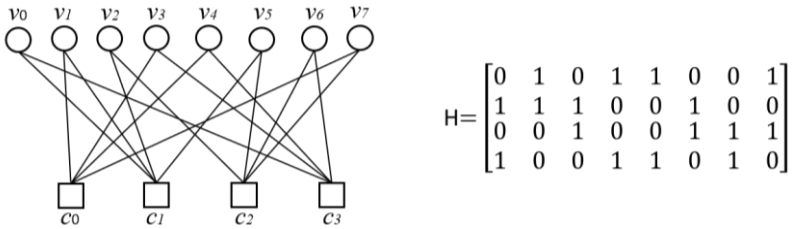


Figure 1. Example of Tanner graph and its parity check matrix

2.1.1. Soft-decision decoding

The MS algorithm is the simplest way to implement the Soft-decision algorithms [18], [25]. It is mainly based on the calculation of extrinsic LLR messages exchanged between the check nodes and the variable nodes of the Tanner graph. This algorithm achieves very high performances in terms of BER [26], but his major disadvantage is the implementation which needs more material resources and consumes more time resulting in a decoding latency increase.

2.1.2. Hard-decision decoding

Soft-decision algorithms calculate the LLRs to evaluate the reliability of received messages, this calculation is more complexe. To overcome this constraint, Gallager [3] has proposed BF algorithm that works in hard-decisions. This type of algorithm simplifies the decoding method by a hard-decision of the message received from the transmission channel at the beginning of the decoding process, and the algorithm calculates the sum of the syndromes per line. If this syndrome is equal to the number of lines in the matrix H, it stops the decoding, otherwise it calculates the inversion function which allows to define the false bits to be inverted. The basic version of BF algorithm [3] is defined in Figure 2.

Compared to soft-decision algorithms, this algorithm searches the minimum value of the Δk function to flip the corresponding bits, so it inverts several bits in the same iteration, which makes the BF algorithm the simplest to implement among all the inversion methods. But inverting several bits at the same time can lead to generation of new errors and finally the decoder cannot detect and correct all the errors. As a consequence, the performances of this algorithm still very far from those obtained by the soft-decision algorithms [15], [16].

To overcome these problems, some previous works use an additive or multiplicative weighting in $\Delta k(x)$ to evaluate the reliability of syndromes. By this way, it can be easy to detect and correct almost all the errors as confirmed for instance by Jinag *et al.* [21] and Gua *et al.* [16]. Modified expressions of $\Delta k(x)$ to improve the BER performances are then proposed, they are all based on (1). In this equation $\emptyset(x_n, y_n)$

represents the metric (reliability) of the received messages y_n, α and ω are weighting factors to balance the values between $\emptyset(x_n, y_n)$ and $\sum_{i\in M(n)} \prod_{j\in N(i)} x_j$.

$$\Delta k^{global}(x) \triangleq \alpha \times \emptyset(x_n, y_n) + \sum_{i\in M(n)} \omega_i \times \prod_{j\in N(i)} x_j \quad (1)$$

Decision:
for $j=1$ ***to*** N ***do*** $x_j = sign(y_j)$ (x_j : hard decision of y_j)
for $l=1$ ***to*** L_{max} ***do*** (L_{max}: maximum iterations)
  for $i=1$ ***to*** M ***do*** $S_i = \prod_{j\in N(i)} x_j$ (S_i : the syndromes by line)
  if $\sum S_i = M$ ***then*** *break*
  for $k=1$ ***to*** N ***do*** $\Delta k(x_k) \triangleq \sum_{i\in M(k)} \prod_{j\in N(i)} x_j$ (Δk : inversion function)
  for $j=1$ ***to*** N ***do***
    if $\Delta k(x_j) = min(\Delta k)$ ***then*** $x_j = -x_j$
until Finished
end Function

Figure 2. Main steps of the BF decoding algorithm

2.1.3. GDBF and RRWGDBF algorithms

The GDBF algorithm is a method that gives a better trade-off between performance and complexity among all hard-decision algorithms [11]. It becomes a viable alternative to the belief propagation (BP) algorithm. In the GDBF algorithm, one must find the code-word that gives the maximum correlation value. The function to be optimized is defined by (2).

$$f(x) \triangleq \sum_{i=1}^{n} x_k y_k + \sum_{i=1}^{m} \prod_{j\in N(i)} x_j \quad (2)$$

For a correct code word, the $f(x)$ function achieves its maximum value. One then has to check to maximize this function by changing the values of x_k. The inversion function, defined by (3), of this algorithm gives the metric for each individual bit that lead to take a decision to flip or not the corresponding bit.

$$\Delta k(x) \triangleq x_k y_k + \sum_{i\in M(k)} \prod_{j\in N(i)} x_j \quad (3)$$

Another algorithm named RRWGDBF has been proposed to improve the GDBF algorithms [14]. This algorithm further increases the convergence speed of the BF algorithm by adding the multiplicative weighting factor, β, in the syndromes, the new metric is then given by (4).

$$\Delta k(x) \triangleq x_k y_k + \sum_{i\in M(k)} \beta_{ik} \prod_{j\in N(i)} x_j \quad (4)$$

Where: $\beta_{ik} = \frac{1}{|y_k|} \sum_{j\in N(i)} y_j$ (5)

Before going further, we undertook to evaluate the BER performances of the above cited algorithms. For that we performed a simple comparison in the case of n=576. Results are shown on Figure 3.

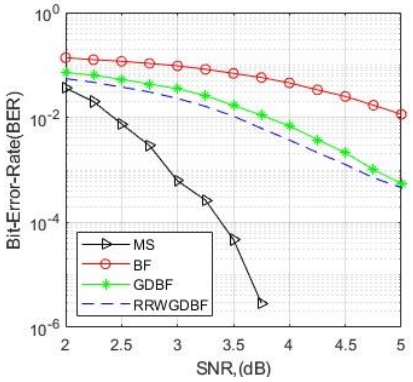


Figure 3. BER performance of MS, BF, GDBF and RRWGDBF algorithms for LDPC code of length-576

It can be observed that the performances of these two last inversion algorithms are better than the BF algorithm but still far from the BER of the MS procedure. Thus, to reduce the gap between these two classes of algorithms, we introduce in the following, two new approaches to improve the BER performances of both the GDBF and the RRWGDBF decoding algorithms by introducing weighting factors to make more reliable their inversion functions.

2.2. Proposed decoding method

As a test experimentation we will focus on the decoding process for the WiMAX standard. Therefore, for the GDBF and RRWGDBF algorithms and for an easier implementation system, we will consider two matrix H based on an irregular QC-LDPC code of codeword length 576 and 1056 [27].

2.2.1. Proposed AGDBF approach

The GDBF algorithm searches for the minimum value of the inversion function, then flips the corresponding bits. During decoding, there is a risk of flipping some correct bits and flipping them again at other times in the next iterations, which induces some performances degradation with an additional delay. Thus, to solve this problem, we introduce a new weighting coefficient, which adjust the values between $x_k y_k$ and the syndrome $\sum_{i\in M(k)} \prod_{j\in N(i)} x_j$. The key of this proposal is to follow the bit flipping, and if we detect that a bit is flipped twice ($N_k = 2$), we stop its flipping procedure by multiplying the first term in the inversion function by a weighting factor α in order to increase its value, therefore, it will not be affected by the flipping next times. And the inversion function becomes:

$$\Delta k(x) = \begin{cases} \alpha \times x_k y_k + \sum_{i\in M(k)} \prod_{j\in N(i)} x_j & if \quad N_k = 2 \\ x_k y_k + \sum_{i\in M(k)} \prod_{j\in N(i)} x_j & otherwise \end{cases} \tag{6}$$

2.2.2. Proposed ARRWGDBF approach

The codes (576,384) and (1056,792) give the best performance for the RRWGDBF algorithm, but the H matrix will not be very sparse, so there will be the presence of short cycles in the H matrix. A cycle starts from a given variable node and shows all the parity and variable nodes to which it will be connected falling back on the starting variable node. Figure 4 illustrate some examples of short cycles of order 4, order 6 and order 8 as shown in Figures 4(a)-(c) usually encountered in H matrix.

To improve the performance of the RRWGDBF algorithm, it is necessary to avoid the generation of short cycles in the Tanner graph, In fact short cycles are very penalizing when calculating the inversion function. The "girth" is the minimum cycle length that can be encountered in a Tanner graph. With the appearance of cycles, the result of the sum of the syndromes $\sum_{i\in M(k)} \prod_{j\in N(i)} x_j$ will not be reliable, which decreases the performance of the decoder. Therefore, it is essential to eliminate short cycles to obtain good decoding performance.

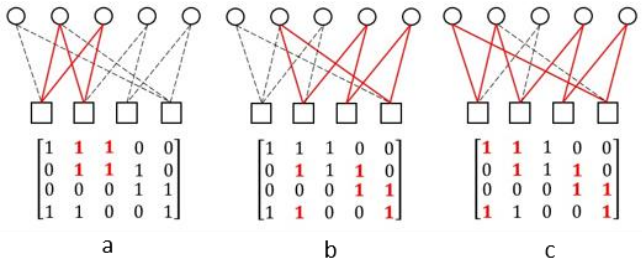


Figure 4. Example of short cycles; (a) order 4, (b) order 6 and (c) order 8

To overcome the problem of the speed convergence for the RRWGDBF algorithm, we suggest in this work to introduce a pre-processing step to search the columns of the short cycles in the H matrix in order to identify them and then to multiply them by a reweighting factor to obtain a gain of bit error rate performances. To identify the columns of short cycles we followed the same method of Yang *et al.* [28] and we found that columns between 265 and 312 for H(192, 576) matrix and columns between 1 and 727 for H(264, 1056) matrix are the columns that present short cycles of order 4. The multiplying factor is then introduced in the first term of the inversion function which is the calaculated for n=576 by the following (7) and (8):

$$\Delta k(x) = \begin{cases} \alpha \times x_k y_k + \sum_{i \in M(k)} \beta_{ik} \prod_{j \in N(i)} x_j & if \quad (264 < k < 313) \quad (7) \\ x_k y_k + \sum_{i \in M(k)} \beta_{ik} \prod_{j \in N(i)} x_j & otherwise \quad (8) \end{cases}$$

3. RESULTATS AND DISCUSSION

3.1. Software validation

The different hard-decision algorithms with the proposed algorithms were coded using the C/C++ programing language. And for simulation, we used the host computer of Intel Core i7 7500U, 2.7 GHz.

3.1.1. Weighting factors determination

For the AGDBF, as the aim is to increase the value of the inversion function when a flipped bit is detected, we permformed a series of simulation using a set of arbitrary positives values of α. Results for the representative ones (2, 5, 10, 20, 50) are presented in Figure 5 and Figure 6 for the two codes (576,384) and (1056,792) respectively. The value of 5 is the minimal value where an appreciable gain can measure and the value of 20 is the higher value from which the observed changes still negligible and even not measurable.

For the ARRWGDBF, we follow the same procedure and representative results for α in the list (2, 5, 10, 20) illustrated in Figures 7 and 8, respectively for n=576 and n=1056. Values of α=5 and α=10 can be assigned to the check nodes that are crossed by the short cycles of order 4, for these two codeword lenghts. In fact, for higher than these values the changes in the BER becomes very negligeables.

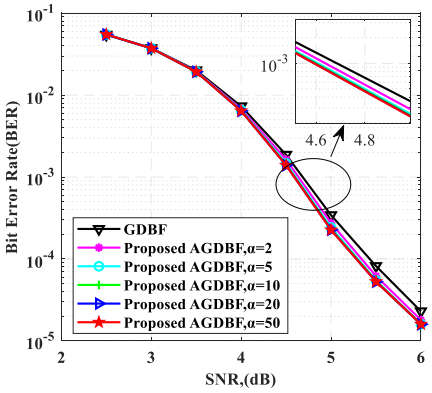


Figure 5. Comparison of performance of AGDBF algorithm for different α values, for an irregular LDPC code (576,384)

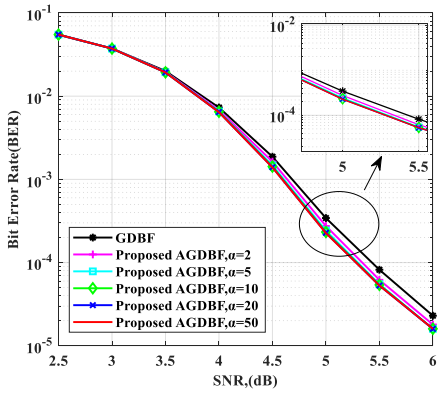


Figure 6. Comparison of performance of AGDBF algorithm for different α values, for an irregular LDPC code (1056,792)

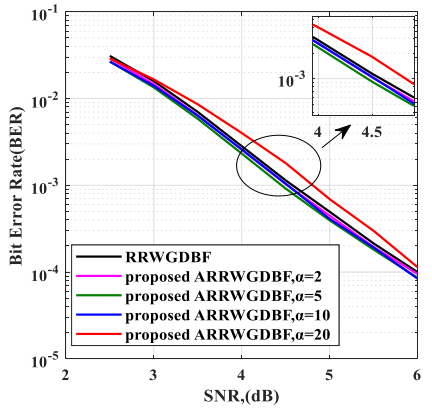


Figure 7. BER performance of ARRWGDBF algorithm for different values of α for an irregular LDPC code (576,384)

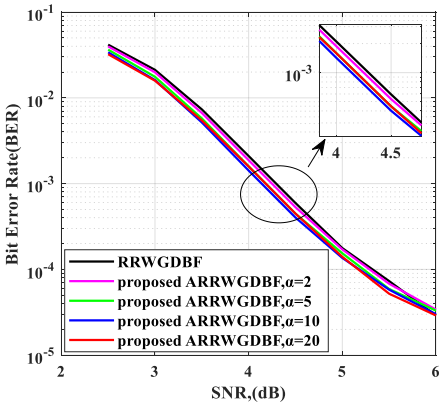


Figure 8. BER performance of ARRWGDBF algorithm for different values of α for an irregular LDPC code (1056,792)

3.1.2. Simulation results

Figures 9 and 10 show respevctiveley the achieved decoding performances by the two proposed algorithms in the case of targeted codeword length 576 and 1056. The AGDBF algorithm gives approximately a gain of 0.2 dB at a BER=10^{-4} for n=576 and approximately 0.17 dB gain at 4×10^{-5} for n=1056 show in Figure 9, thus, the use of the weighting factor improves the decoding performance compared to the GDBF algorithm. For the ARRWGDBF algorithm a gain of approximately 0.14 dB is obtained at a BER=10^{-3} for n=576 and a gain of approximately 0.17 dB at BER=4×10^{-4} for n=1056 show in Figure 10, again, the pre-processing step in the H matrix and the use of the reweighting factor in the inversion function lead to non negligeable gain in the BER performances.

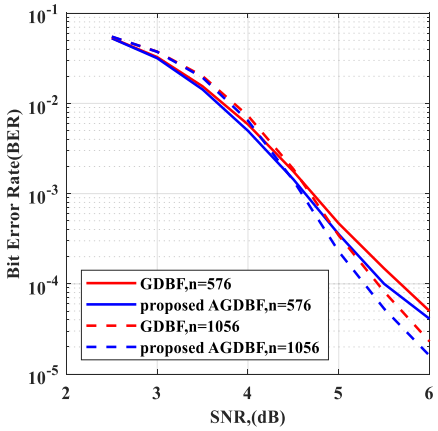


Figure 9. BER performance of the GDBF and the proposed AGDBF algorithms

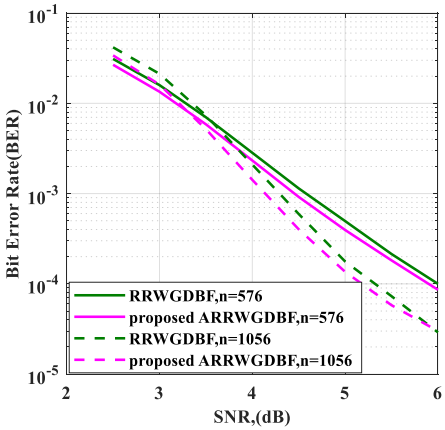


Figure 10. BER performance of the RRWGDBF and the proposed ARRWGDBF algorithms

In Table 1, a comparison between the new and basic algorithms in term of BER is presented. It can be seen that the BER performance, between the GDBF [29] and the proposed AGDBF algorithms is improved and also the number of iterations is highly reduced, 30 iterations in our case instead of 60 in the previous version [29]. Idem, in the proposed ARRWGDBF algorithm the BER is improved.

Table 1. Comparison of performance of GDBF [29], RRWGDBF, AGDBF, and ARRWGDBF algorithms

Algorithms	Code length	Rate	SNR [dB]	L*max*	BER
GDBF [29]	1000	0.9	4	60	10^{-2}
Proposed AGDBF	1056	0.75	4	30	6×10^{-3}
RRWGDBF	1056	0.75	4	30	2×10^{-3}
Proposed ARRWGDBF	1056	0.75	4	30	1.5×10^{-3}

3.2. Hardware implementation

DSP implementation of the proposed solutions is helpful to to evaluate the performances and the delays of algorithms. The DSP processor has a fast kernel that allows high speed memory accesses and it also can suggest some improvement ways. The platform used in this work is the Texas Instrument's TMS320C6713 floating point DSP processor [30]. The LDPC decoder has been developed in software on the Code Composer Studio Simulator using C/C++ programming language.

For applications that require a large codes length, to speed up the decoding process and optimize memory, we adopt the same method as we reported previously [31], we have stored just the positions of the 1s in the H matrix. This method simplifies the check for the 1s during decoding and decreases the time processing by reducing the number of memory access. Figure 11 illustrates the steps we followed to implement our algorithms:

In the case of the code (576,384), Figure 12 shows that, for AGDBF algorithm, a gain of 0.25dB is obtained for a BER of 5×10^{-5} and for ARRWGDBF algorithm, a gain of 0.15dB is obtained for a BER of 3×10^{-4}. On Figure 13, (case of the (264,1056) code), for the AGDBF algorithm, a gain of 0.13dB is obtained for a BER of 5×10^{-5} and for ARRWGDBF algorithm, a gain of 0.15dB is obtained for a BER of 3×10^{-4}, illustrating the performance improvement of the decoding process in each case.

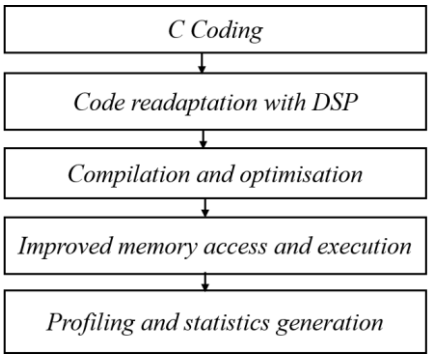


Figure 11. Flowchart of the DSP implementation

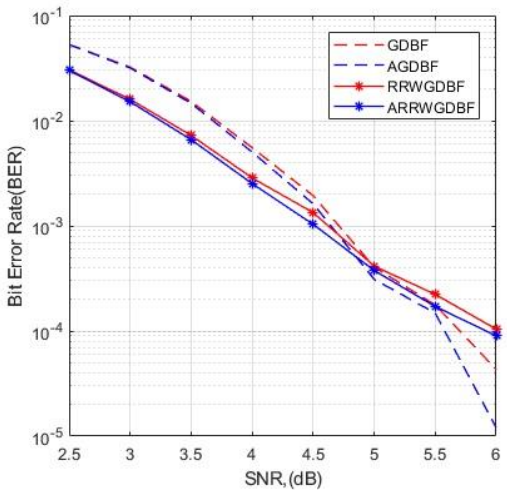


Figure 12. BER of GDBF, AGDBF, RRWGDBF and ARRWGDBF algorithms for an irregular LDPC code (576,384)

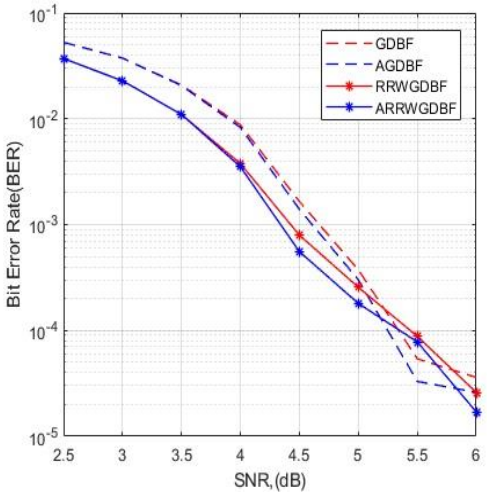


Figure 13. BER of GDBF, AGDBF, RRWGDBF and ARRWGDBF algorithms for an irregular LDPC code (1056,792)

Table 2 shows a comparison of the processing time per iteration and the memory resources consumed for the MS, AGDBF and ARRWGDBF algorithms. As far as memory requirements are concerned, the sparse matrix implementation, storing just the positions of the 1's in the H matrix, saves a considerable amount of memory by allowing the entire code to reside only on the on-chip memory. The use of on-chip memory IRAM avoids accesses to slower off-chip memory, and the AGDBF and ARRWGDBF algorithms exploit less memory. On the other hand, the MS algorithm uses the on-chip and external memory SDRAM. Thus, the number of cycles per iteration of the proposed AGDBF and ARRWGDBF algorithms is much lowered compared to the MS algorithm. Therefore, the system can implement a large codes length based on the proposed algorithms. In the same table, it can also be seen that the number of cycles and memory allocation, between the GDBF and AGDBF algorithms from one part and between the RRWGDBF and ARRWGDBF algorithms from the other part, are almost unchanged.

Table 2. Comparison of number of cycles per iteration and memory allocation for MS, GDBF, AGDBF, RRWGDBF and ARRWGDBF algorithms for n=576

Algorithms	MS	GDBF	Proposed AGDBF	RRWGDBF	Proposed ARRWGDBF
Number of cycles per iteration	8.7×10^6	411×10^3	425×10^3	1.39×10^6	1.4×10^6
Memory allocation (IRAM) (KBytes)	41	29	29.3	32.4	32.5
Memory allocation (SDRAM) (KBytes)	12700	0	0	0	0

4. CONCLUSION

In this paper, an improvement of the decoder performances is presented, based on new proposed adaptative versions of the GDBF and RRWGDBF algorithms. Our approach modifies the GDBF decoding algorithm with an added weighting coefficient to avoid multiple flipping of some right bits leading to the named adaptive GDBF (AGDBF) version. In the case of RRWGDBF decoding algorithm, a new named adaptive RRWGDBF (ARRWGDBF) is suggested in which a pre-processing step to check the columns of the short cycles in the H matrix and using a weighting correction factor.

Software simulation and real time implementation using DSP platform have been carried out to evaluate and to optimize the performance of the proposed LDPC decoder. Implementation results show the improvement of the proposed approaches in terms of convergence and also conserved the same processing time. Moreover, for length-576, at BER of 5×10^{-5}, the AGDBF achieves approximately 0.25 dB gain over the GDBF algorithm, and for BER of 3×10^{-4}, the ARRWGDBF achieves approximately 0.15 dB gain; For the code length 1056, at BER of 5×10^{-5}, the AGDBF gives gain of 0.13 dB over the GDBF algorithm, and for the BER of 3×10^{-4}, the ARRWGDBF reaches a gain of 0.15 dB. The proposed algorithms are useful for the communication system such as wireless systems, such WiMAX, 4G, 5G. The term adaptative is introduced here because the weighting coefficients can be changed from one situation to another, making it adaptable for each case.

REFERENCES

[1] A. M. A. Hamdoon, Z. G. Mohammed and E. A. Mohammed, "Design and implementation of single bit error correction linear block code system based on FPGA," *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, vol. 17, no. 4, pp. 1785-1795, Aug. 2019, doi: 10.12928/telkomnika.v17i4.12033.

[2] G. A. Hussain and L. Audah, "UFMC system performance improvement using RS codes for 5G communication system," *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, vol. 18, no. 4, pp. 1843-1848, Aug. 2020, doi: 10.12928/telkomnika.v18i4.15703.

[3] R. G. Gallager, "Low density parity check codes," *IRE Transaction on Information Theory*, vol. 8, no. 1, pp. 21-28, Jan. 1962, doi: 10.1109/TIT.1962.1057683.

[4] P. Urard *et al*., "A 135Mb/s DVB-S2 compliant codec based on 64800b LDPC and BCH codes," *ISSCC. 2005 IEEE International Digest of Technical Papers. Solid-State Circuits Conference*, vol. 1, 2005, pp. 446-609, doi: 10.1109/ISSCC.2005.1494061.

[5] K. V. Kumar, R. Shrestha and R. Paily, "Design and implementation of multi-rate LDPC decoder for IEEE 802.16e wireless standard," *2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE)*, 2014, pp. 1-5, doi: 10.1109/ICGCCEE.2014.6922226.

[6] A. H. Nalband, M. Sarvagy and M. R. Ahmed, "Power saving and optimal hybrid precoding in millimeter wave massive MIMO systems for 5G," *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, vol. 18, no. 6, pp. 2842-2851, Dec. 2020, doi: 10.12928/telkomnika.v18i6.15952.

[7] M. Razi, A. Mansouri, M. Benhayoun, A. A. Madi and A. Ahaitouf, "Fast DSP Implementation of a Low Complexity LDPC Decoder," *2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, 2019, pp. 1-5, doi: 10.1109/WITS.2019.8723838.

[8] N. El Maammar, S. Bri and J. Foshi, "A comparative simulation study of different decoding schemes in LDPC coded OFDM systems for NB-PLC channel," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 15, no. 1, pp. 306-313, Jul. 2019, doi: 10.11591/ijeecs.v15.i1.pp306-313.

[9] 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.

[10] B. Unal, A. Akoglu, F. Ghaffari and B. Vasić, "Hardware Implementation and Performance Analysis of Resource Efficient Probabilistic Hard Decision LDPC Decoders," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 9, pp. 3074-3084, Sep. 2018, doi: 10.1109/TCSI.2018.2815008.

[11] T. Wadayama, K. Nakamura, M. Yagita, Y. Funahashi, S. Usami and I. Takumi, "Gradient descent bit flipping algorithms for decoding LDPC codes," *2008 International Symposium on Information Theory and Its Applications*, 2008, pp. 1-6, doi: 10.1109/ISITA.2008.4895387.

[12] Y. Kou, S. Lin and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," in *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2711-2736, Nov. 2001, doi: 10.1109/18.959255.

[13] J. Zhang and M. P. C. Fossorier, "A modified weighted bit-flipping decoding of low-density Parity-check codes," in *IEEE Communications Letters*, vol. 8, no. 3, pp. 165-167, Mar. 2004, doi: 10.1109/LCOMM.2004.825737.

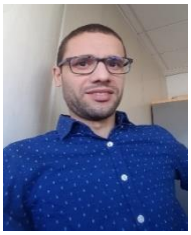
[14] T. Phromsa-ard, J. Arpornsiripat, J. Wetcharungsri, P. Sangwongngam, K. Sripimanwat and P. Vanichchanunt, "Improved Gradient Descent Bit Flipping algorithms for LDPC decoding," *2012 Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP)*, 2012, pp. 324-328, doi: 10.1109/DICTAP.2012.6215420.

[15] M. Jiang, C. Zhao, Z. Shi and Yu Chen, "An improvement on the modified weighted bit flipping decoding algorithm for LDPC codes," in *IEEE Communications Letters*, vol. 9, no. 9, pp. 814-816, Sep. 2005, doi: 10.1109/LCOMM.2005.1506712.

[16] F. Guo and L. Hanzo, "Reliability ratio based weighted bit-flipping decoding for low-density parity-check codes," *IEEE Electronics Letters*, vol. 40, no. 21, pp. 1356-1358, 2004, doi: 10.1049/el:20046400.
[17] B. Ünal, F. Ghaffari, A. Akoglu, D. Declercq and B. Vasić, "Analysis and implementation of resource efficient probabilistic Gallager B LDPC decoder," *2017 15th IEEE International New Circuits and Systems Conference (NEWCAS)*, 2017, pp. 333-336, doi: 10.1109/NEWCAS.2017.8010173.
[18] M. P. C. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," in *IEEE Transactions on Communications*, vol. 47, no. 5, pp. 673-680, May. 1999, doi: 10.1109/26.768759.
[19] T. J. Richardson, M. A. Shokrollahi and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," in *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619-637, Feb. 2001, doi: 10.1109/18.910578.
[20] S. Myung, K. Yang, and J. Kim, "Quasi-cyclic LDPC codes for fast encoding," in *IEEE Transactions on Information Theory*, vol. 51, no. 8, pp. 2894-2901, Aug. 2005, doi: 10.1109/TIT.2005.851753.
[21] M. P. C. Fossorier, "Quasicyclic low-density parity-check codes from circulant permutation matrices," in *IEEE Transactions on Information Theory*, vol. 50, no. 8, pp. 1788-1793, Aug. 2004, doi: 10.1109/TIT.2004.831841.
[22] X. Y. Shih and H. R. Chou, "Flexible design and implementation of QC-Based LDPC decoder architecture for on-line user-defined matrix downloading and efficient decoding," *Integration*, vol. 64, pp. 40-49, 2019, doi: 10.1016/j.vlsi.2018.07.008.
[23] M. Ardakani, B. Smith, W. Yu, and F. R. Kschischang, "Complexity-optimized low-density parity-check codes," *In Proceedings of the Forty-Third Annual Allerton Conference on Communication, Control and Computing*, 2005, pp. 45-54.
[24] N. El Maammar, S. Bri, and J. Foshi, "Performances Concatenated LDPC Based STBC-OFDM System and MRC Receivers," *International Journal of Electrical and Computer Engineering (IJECE)*, vol.8, no. 1, pp. 622-630, Feb. 2018, doi: 10.11591/ijece.v8i1.pp622-630.
[25] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 33, no. 18, pp. 1645–1646, 1996, doi: 10.1049/el:19961141.
[26] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier and X-Yu Hu, "Reduced-complexity decoding of LDPC codes," in *IEEE Transactions on Communications*, vol. 53, no. 8, pp. 1288-1299, Aug. 2005, doi: 10.1109/TCOMM.2005.852852.
[27] B. Classon *et al*., "LDPC coding for OFDMA PHY," document IEEE C802.16e-05/0066r3, 2005.
[28] L. Yang, Hui Liu and C. R. Shi, "VLSI implementation of a low-error-floor and capacity-approaching low-density parity-check code decoder with multi-rate capacity," *GLOBECOM '05. IEEE Global Telecommunications Conference*, 2005, pp. 6, doi: 10.1109/GLOCOM.2005.1577854.
[29] A. Li, V. Meghdadi, J. Cances and C. Aupetit-Berthelemot, "High Rate LDPC Based Decoder Architectures with High Speed ADC for C-RAN Optical Fronthaul," *2016 International Conference on Computer and Communication Engineering (ICCCE)*, 2016, pp. 386-391, doi: 10.1109/ICCCE.2016.88.
[30] R. Gummattira, P. Baltz, N. Seshan, "TMS320C6713 Digital Signal Processor Optimized for High Performance Multichannel Audio Systems," Texas Instruments, Texas, USA, Aplication Report SPRA921-June 2003. [Online]. Available: http://www.ti.com/lit/an/spra921/spra921.pdf.
[31] M. Benhayoun, M. Razi, A. Mansouri and A. Ahaitouf, "New Memory Load Optimization Approach for Software Implementation of Irregular LDPC Encoder/Decoder," *2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, 2019, pp. 1-6, doi: 10.1109/WITS.2019.8723841.

BIOGRAPHIES OF AUTHORS

Mouhcine Razi, in 2007, he obtained the engineering degree in electronics systems engineering from National school of applied sciences, Tangier, Morocco. He is currently an engineer in the Faculty of Sciences and technology of Fez, Morocco. His research activities focus on VLSI and embedded systems.



Mhammed Benhayoun, in 2006, he obtained the engineering degree in Networks and Telecommunications from National school of applied sciences, Tangier, Morocco. He is currently an engineer in telecommunications campany in Morocco. His research activities focus on telecommunication, signal processing and embedded systems.

Anass Mansouri received Ph.D degrees in Microelectronics and Embedded System from Faculty of sciences & technology, Fes, Morocco, in 2009. He is an Associate Professor in National School of Applied Sciences, Fes. His major research interests include Real Time Embedded System, VLSI and embedded architectures design, video and image Processing.

Ali Ahaitouf, teacher and researcher in the University of Sidi Mohammed Ben Abdellah University of Fes, Morocco, since 1992. He obtained his PhD in electronics since 1998. Currently, he is director of the laboratory of intelligent systems, georesources and renewable energies. His research field covers microelectronics and solar compounds, digital and analog design of the integrated circuits, Image and data compression. He managed many research multilateral projects, related to analog design optimization, electronics component characterization and optimization as well as solar energy under concentration (CPV). He works in partnership with several international teams, including the joint International Unit Georgia-Tech-CNRS, where he spent a full-time year (2012) and where he hold a chair of excellence (2014) on the CPV. He authored and co-authored more than sixty papers in the international journals. He supervised a dozen of PhD thesis and many post-graduate students.