IMPLEMENTATION OF DECODING ARCHITECTURE FOR LDPC CODE USING A LAYERED MIN-SUM ALGORITHM

SANDEEP KAKDE[1*], ATISH KHOBRAGADE[1], SHRIKANT AMBATKAR[1] AND PRANAY NANDANWAR[1]

[1]Department of Electronics, Faculty of Engineering, YeshwantraoChavan College of Engineering, Nagpur, India.

**Corresponding author: sandip.kakde@gmail.com*

(Received: 25th April 2016; Accepted: 3rd April 2017; Published on-line: 1st Dec. 2017)

ABSTRACT: LDPC codes provide remarkable error correction performance and therefore enlarge the design space for communication systems. This paper shows comparison of different digital modulation techniques and found that BPSK modulation technique is better than other modulation techniques in terms of Bit Error Rate. It also gives error performance of LDPC decoder over AWGN channel using Min-Sum algorithm. VLSI Architecture is proposed which uses the value re-use property of min-sum algorithm and gives a high throughput. The proposed work has been implemented and tested on Xilinx Virtex 5 FPGA.The MATLAB result of LDPC decoder for low bit error rate gives a bit error rate in the range of 10^{-1} to $10^{-3.5}$ at SNR=1 to 2 for 20 iterations. Thus, it gives good bit error rate performance. The latency of the parallel design of the LDPC decoder has also decreased. It has attained a maximum frequency of 141.22 MHz and throughput of 2.02 Gbps while consuming less area.

ABSTRAK: Kod LDPC adalah sistem prestasi pembetulan ralat yang luar biasa dan dengan ini meluaskan peluang reka sistem komunikasi. Kertas ini menyediakan perbandingan perbezaan pelbagai teknik digital modulasi dan didapati teknik modulasi BPSK adalah lebih baik daripada teknik modulasi lain dalam bentuk Kadar Ralat Bit (Bit Error Rate). Ia juga memberi prestasi ralat LDPC penyahkod yang bagus berbanding saluran AWGN menggunakan algoritma Min-Sum. Struktur VLSI telah dicadangkan dengan mengguna semula nilai pakai dan menghasilkan nilai penghantaran yang tinggi. Kerja cadangan ini telah dibina dan diuji mengguna Xilinx Virtex 5 FPGA. Keputusan MATLAB penyahkod LDPC bagi kadar ralat bit telah berhasil pada kadar 10^{-1} to $10^{-3.5}$ dan SNR=1 kepada 2 dengan 20 kali iterasi. Jadi ia berhasil memberikan prestasi kadar ralat bit yang bagus. Latensi pada reka serentak LDPC penyahkod juga telah menurun. Ia berhasil mencapai 141.22 MHz frekunsi maksimum dan penghantaran sebanyak 2.02 Gbps pada luas kawasan yang sedikit.

KEYWORDS: *low-density parity-check (LDPC) codes; min-sum algorithm (MSA); TDMP;offset min-sum algorithm (OMS); bit error rate (BER); SNR; AWGN*

1. INTRODUCTION

Currently, Low Density Parity Check codes (LDPC) have gained major attention because of their performance near to the Shannon limit and remarkable error correction capability. LDPC codes were invented by Gallager in 1962 [1] and got more attention when they were rediscovered by McKay and Neal in 1996 [2]. LDPC codes can achieve very low bit error rate (BER) for low signal to noise ratio (SNR) applications [1]. LDPC codes has got several advantages such as low hardware complexity, inherit parallelism, no

error floor at high SNRs and high throughput potential. They have been used in several wireless standards such as IEEE 802.16e, IEEE 802.11n and DVB S-2 because of their excellent error correcting performance.

A LDPC code is a linear block code which is defined by a sparse parity check matrix (H-matrix), as shown in Fig 1. In matrix representation, consider a parity check matrix with n x m dimension.

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

Fig. 1: Example of parity check matrix.

Consider *Wr* as weight of rows (i.e. number of 1's in each row) and *Wc* as weight of columns (i.e. the number of 1's in each column). Now for the matrix to be called a Low-Density matrix two conditions must be satisfied: $Wc<<n$ and $Wr<<m$. They can be represented by a bipartite graph also called a Tanner Graph, shown in Fig 2. A Tanner graph contains two sets of nodes, bit nodes and check nodes. Bit nodes represent the bits of a codeword and check nodes apply the parity-check constraints. Between each bit node '*i*' and check node '*j*', there exists an edge only if $H(i,j)=1$.

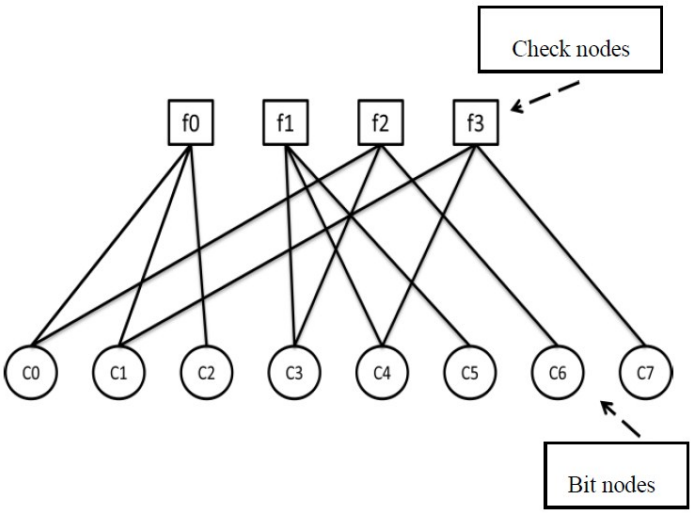

Fig. 2: Tanner graph.

Message passing algorithms, known as "sum-product" (SP) and "belief propagation" (BP) provide the standard decoding procedure, that iteratively exchange the message bits between the check nodes and the bit nodes along the edges of the tanner graph. Based on the number of the processing units, there are various ways to implement the LDPC decoder. In general, they can be classified into three types: fully parallel architecture [3], serial architecture [4], and partial parallel architecture [5-7]. In serial architecture, one check node processor shares the computation of all the check nodes and thus it is very slow. In fully parallel architecture, every check node requires a check node processor for its operations which results in very large hardware, complex routing and is therefore less flexible. The LDPC decoder's implementation requires a balance between factors such as

error correction performance, decoding throughput, and the hardware complexity. Hence, partial parallel architecture is the best suited because multiple processing units are provided that allows proper trade-off between the hardware complexity and the decoding throughput. Lambda min-sum algorithm based architectural issues, described and based on a serial-parallel-serial schedule based architecture is proposed in the paper [8]. Chen and Fossorier had given a performance comparison for different check updates [9,10]. In their research they showed that the offset min-sum decoding algorithm with 5-bit quantization could achieve the same BER performance as that of floating point sum-product and Jacobian-based BCJR algorithm (named after its discoverers Bahl, Cocke, Jelinik, and Raviv) with less than 0.1 dB penalty in SNR). Layered decoding can be done in two ways i.e. vertical layered decoding [11] and horizontal layered decoding [12, 13]. Theoretically, both the methods are equivalent and prove that both can converge twice as fast as the belief propagation algorithm [14]. For min-sum algorithm, horizontal layered decoding is preferable and hence used in this paper. A different and multi rate block length high throughput architecture is proposed in the paper [15, 16].

2. MIN-SUM DECODING ALGORITHM

Assuming the codeword (*a1,a2,a3,...,an*) was sent over AWGN channel and received as (*b1,b2,b3,....,bn*).

A minimum sum algorithm can be summarized as follows:

- Variable nodes are first initialized with soft information bits, also called log-likelihood ratio (LLRs) from channel. After the initialization, each bit node updates its corresponding check nodes with bits to check the message.

$$Qij = Li = log\frac{P(a_i = 0|bi)}{P(a_i = 1|bi)} \tag{1}$$

  where Q_{ij} is message from bit node V(i) to check node C(i).

- Check node update: check node messages are calculated using the Q message received from the bit node as follows:

$$R_{ij} = \prod_{j\prime \in V(i)\backslash j} sign(Q_{ij}) \times \min|Q_{ij\prime}| \tag{2}$$

  where *Rij* is the message from the check node C(i) to the bit node V(j).

  While calculating the $min|Q_{ij'}|$for R_{ij}, check node processing needs to exclude *V(j)* therefore it is necessary to find two minima i.e. 1st minimum (Min1) and 2nd minimum (Min2). More precisely $\min|Q_{ij'}|$ can be defined as

$$min_{j\prime \in V(i)\backslash j}|Q_{ij'}| = \begin{cases} Min1, & if\ j \neq argmin(Min1i) \\ Min2, & if\ j = argmin(Min2i) \end{cases} \tag{3}$$

- Bit node update: Bit nodes calculate variables to check messages, Q_{ij}, using its old value and received R_{ij} message from all neighboring check nodes:

$$Q_{ijnew} = Q_{ijold} + \sum_{i\prime \in C(j)\backslash i} R_{i\prime j} \tag{4}$$

Same as in case of check to bit messages, message from bit node *V(i)* to check node *C(j)* is calculated using all *R* message from neighboring check nodes except *R* message from check node *C(j)*.

- After bit node update is finished, for each bit the reliability value is calculated as in Eq.(4) but with all *R* values received from neighboring check nodes as given in Eq.(5)

$$y_i = Q_{ijold} + \sum_{i \in C(j)} R_{ij} \tag{5}$$

From above equation ,the *Y* vector is

$Y = (y1, y2, y3, \ldots, y_n)$

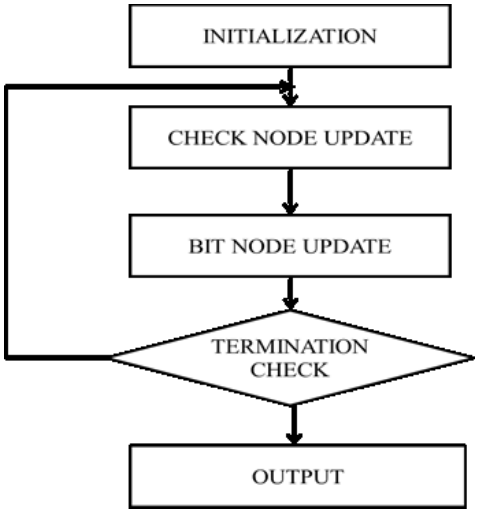

Fig. 3: Flowchart of min-sum algorithm.

Bit values are calculated by

$$z_i = \begin{cases} 1 \; if \; y_i \leq 0 \\ 0 \; if \; y_i > 0 \end{cases} \tag{6}$$

Now if H x $Z^T = 0$, then Z is a valid code and the decoding process stops here, or else the decoding process continues until a valid code is obtained or until it reaches the maximum number of iterations count, resulting in the output of an error flag and proceeding for decoding of the next data frame.

3. DECODER ARCHITECTURE AND OPERATION

The architecture for the LDPC decoder is given in Fig. 4. Basically, the working of the decoder can be defined as two-state processing i.e. Bit node processing and Check node processing. The check node and the bit node processing are done in a time division multiplexed way. It takes *m* clock cycles to process a block row i.e. a layer. The check node processing unit (CNU) used here is a serial CNU.

The working of the CNU can be explained from Fig 5. Incoming variable messages (Q messages) are compared with the two up-to-date least minimum numbers stored in a partial state to generate the new partial state i.e. first minimum (M1), second minimum (M2), and

the index of M1. The function of the min finder block is to find the first and the second minimum value. The offset block applies the offset correction and stores the values in the final state block. The final block contains three values M1, -M1, and +/-M2. The R-select block then assigns one of these 3 values to output R messages based on the R-sign value, which is obtained from XORing of the sign of R, and the index of M1. Normally, bit node processing is done in 2's. Complement arithmetic and the check node processing is done using signed magnitude arithmetic and therefore, here, 2's compliment is required, at the input of CNU, to perform sign conversion and at the output signed to 2's compliment.

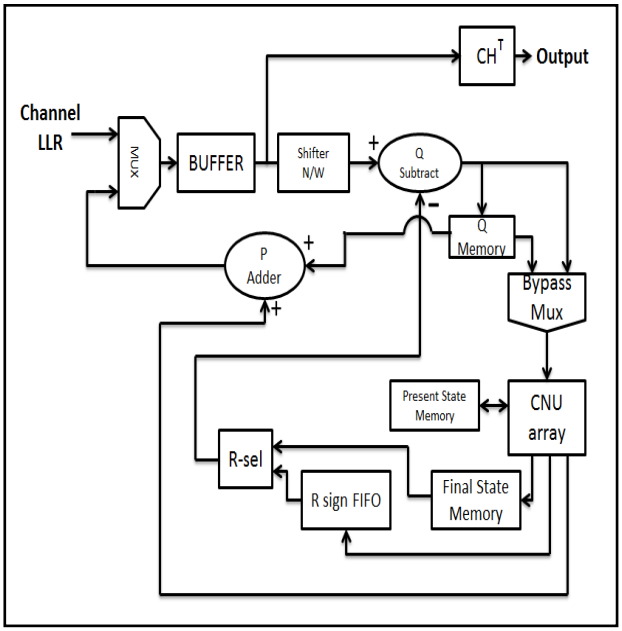

Fig. 4: LDPC decoder architecture.

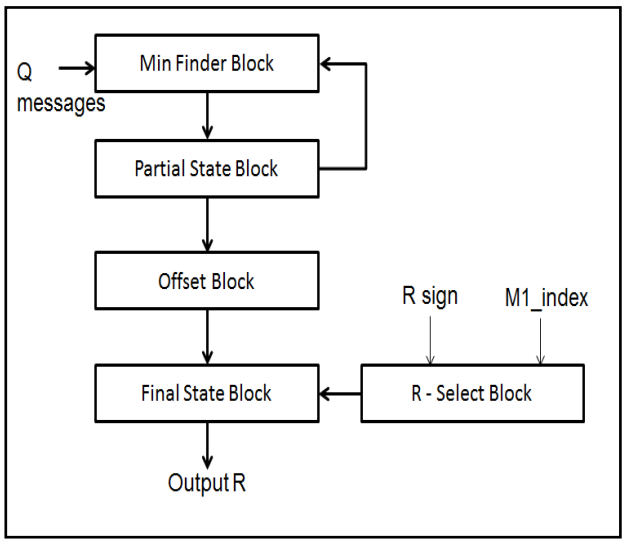

Fig. 5: Flowchart of CNU block.

The CNU array operates in pipelined mode on the partial states and the R messages of the adjacent block rows. The final state depends on the partial state, P and Q messages are reliant on the final state. The final state of the previous block row is needed for passing the LDPC decoding message and this compact information for CNU is stored in the final state

memory. Partial state memory is required to store the partial states. The CNU array will initialize the partial state processing for succeeding block row as soon as the computation of the previous block row is over.

The Q messages that have been feed in the present block row depend on R messages of the current layer as well as the R messages belonging to different blocks of different layers, requiring the R selection process which can be obtained by the R-sel block. The R-sel block generates the R message for edges of a check node out of the three values stored in final state memory which is associated with the specific check node in a serial way. Its working is the same as that of the R-select block in CNU. The flow chart of the CNU block is as shown in Fig. 5.

4. MATLAB IMPLEMENTATION RESULTS

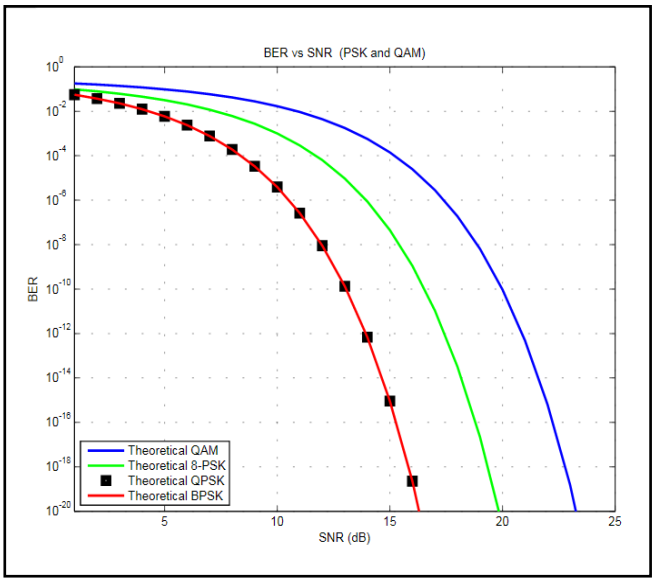

Fig. 6: Simulation result of BER versus SNR (dB) for different modulation techniques over AWGN channel.

Firstly, simulation of bit error rate (BER) performance of various digital modulation techniques viz. QAM, BPSK, QPSK and 8-PSK over an AWGN channel was shown, and demonstrated that the BER performance of BPSK is superior when compared to other digital modulation techniques, as shown in Fig. 6. This is because in the BPSK modulation technique, only 1 bit per symbol is used for modulation and in case of QPSK it's 2 bits per symbol and for 8-PSK it's 3 bits per symbol. As the number of bits per symbol is increased, the chances of getting an error in code also increases, hence the error rate increases.

BER against SNR performance of un-coded BPSK (theoretical and practical) and Min-Sum algorithm over an AWGN channel, as shown in Fig. 7, is simulated in MATLAB software and found that min-sum algorithm shows exceptional BER against SNR performance when compared with the un-coded BPSK. Here, code length is 1000 and the maximum iterations is limited to 20.

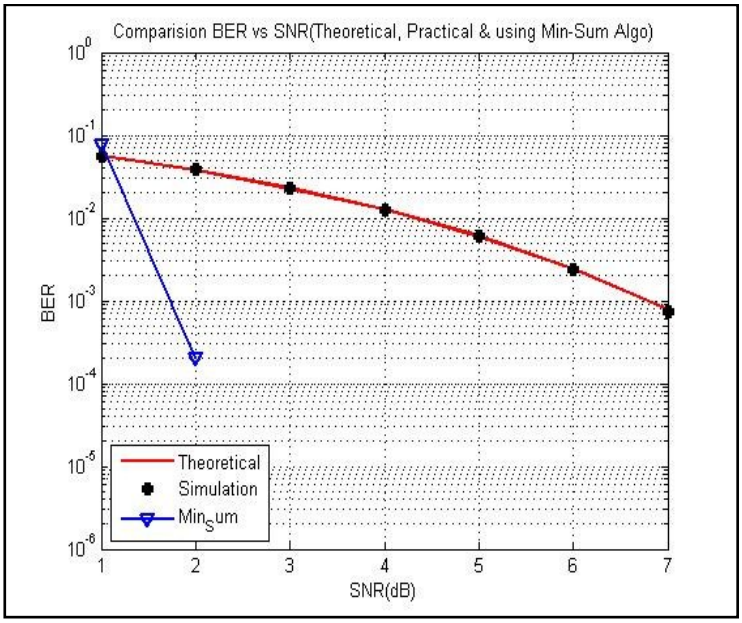

Fig. 7: Theoretical and Practical BER versus SNR performance of BPSK (un-coded) and using min-sum algorithm over AWGN channel. (code length = 1000, code rate = 1/2, max iteration = 20)

5. VERILOG HDL IMPLEMENTATION RESULTS

Table 1: Device utilization summary for LDPC Decoder (576-bits)

LDPC Decoder			
Family:Virtex5 **Target Device**: xc5vfx130t-2ff1738			
Features	**Used**	**Available**	**Utilization**
No.of Slice Registers	1794	81920	2%
No.of Slice LUTs	1630	81920	1%
No.of fully used LUT-FF Pairs	981	2443	40%
No. of BUFG/BUFGCTRLs	1	32	3%

Table 2: Comparison of the proposed LDPCdecoder with the state of the art implementations

Parameters	**[14]**	**[15]**	**[16]**	**This Work**
Platform	ASIC	FPGA	FPGA	FPGA
Tech./Family	130 nm	Xilinx	Virtex-4	Virtex-5
Codelength	576~2304	648	1296	576
Iterations	10, 15	5	15	10
LUTs	--	19332	36452	1630
Throughput	133-928 Mbps	127.36 Mbps	1.1 Gbps	2.02Gbps

Throughput Calculations

Throughput of the LDPC Decoder is given by the formula

$$T = \frac{codelength \times f_{max}}{N_{it} \times \theta}$$

Where,

f_{max} is the maximum operating frequency, N_{it} is the number of decoding iterations, and $\theta$is the number of clock cycle required to complete one iteration.

Here the codelength is of 576 bits, from the synthesis result, the maximum operating frequency of decoder is 141.22 MHz, the number of decoding iterations required is 10, 4 clock cycles are required to complete one iteration, and throughput is 2.02 Gbps.

6. CONCLUSION

This work provides the concept of implementing the LDPC Decoder with a new Check-Node Unit using a Min-Sum algorithm. The results show that the area of the proposed design is reduced from previous designs. The design is simulated in Verilog HDL with Xilinx 14.7 and in MATLAB for bit error rate performance. It can be concluded that the overall area needed for the design is less as compared to other design results of LDPC decoders. Simulation has been carried out in MATLAB for different codelength and iterations assuming a binary phase shift-keying (BPSK) modulated signal transmitted over the AWGN channel. The MATLAB result of the LDPC decoder for bit error rate gives a BER in the range of 10^{-1} to $10^{-3.5}$ at SNR=1 to 2 for 20 iterations. Therefore, it gives a good bit error rate performance. The latency of the parallel design of the LDPC decoder has also decreased. It has accomplished a 141.22 MHz maximum frequency anda throughput of 2.02 Gbps while consuming less area (in terms of Slice LUTs required) for the design.

REFERENCES

[1] Gallager R.(1962) Low-Density Parity-Check Codes," IRE Tans. Inf. Theory, 7:21-28.
[2] MacKay D, Neal R. (1996) Near Shannon limit performance of low density parity check codes," Electron. Lett., 32(18):1645-1646.
[3] Blanksby A, Howland C. (2002) A 690-mW 1-Gb/s 1024-b, rate-1/2 low-density parity-check code decoder. IEEE J. Solid-State Circuits, 37(3):404-412.
[4] Yeo E, Pakzad P, Nikolic B, Anantharam V. (2001) VLSI architectures for iterative decoders in magnetic recording channels. IEEE Trans. Magn., 37(2):748-755.
[5] Zhang T, Parhi K. (2001) VLSI implementation-oriented (3,k)-regular low-density parity-check codes. In Proc. IEEE Workshop on Signal Process. Syst. (SiPS), pp 25-36.
[6] Mansour M, Shanbhag N. (2003) High-throughput LDPC decoders. IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 11(6):976-996.
[7] Wang Z, Cui Z. (2007) Low-Complexity High-Speed Decoder Design for Quasi-Cyclic LDPC Codes. IEEE Trans. VLSI Syst., 15(1):104-114.
[8] Guilloud EB, Danger J. (2003) λ-Min Decoding Algorithm of Regular and Irregular LDPC codes. In Proc. 3rd Int. Symp. Turbo Codes and Related Topics, pp 451-454.
[9] Chen J, Dholakia A, Eleftheriou E, Fossorier M, Hu X-Y. (2005) Reduced-Complexity Decoding of LDPC Codes," IEEE Trans. Commun., 53(8):1288-1299.
[10] Fossorier M, Mihaljevic M, Imai H. (1999) Reduced complexity iterative decoding of low-density parity-check codes based on belief propagation. IEEE Trans. Commun., 47(5):673-680.

[11] Zhang J, Fossorier M. (2005) Shuffled iterative decoding. IEEE Trans. Commun., 53(2):209-213.
[12] Gunnam KK, Choi GS, Yeary MB, Atiquzzaman M. (2007) VLSI Architectures for Layered Decoding for Irregular LDPC Codes of WiMax, IEEE International Conference on Communications, pp 4542-4547.
[13] Wu X, Song Y, Cui L, Jiang M, Zhao C.(2010) Adaptive-normalized min-sum algorithm. Future Computer and Communication (ICFCC), 2nd International Conference on, 2:661-663.
[14] Brack T, Alles M, Kienle F, When N.(2006) A synthesizable IP core for WiMAX 802.16e LDPC code decoding.In Proc. IEEE 17th Int. Symp.Personal, Indoor and Mobile Radio Communications, pp 1–5.
[15] Itsara T, Choomchuay S. (2012) A hardware design of MS/MMS-based LDPC decoder. In Electron Devices and Solid State Circuit (EDSSC), 2012 IEEE International Conference on, pp 1-4.
[16] Predrag R, De Baynast A, Karkooti M, Cavallaro JR. (2006) Multi-rate high-throughput LDPC decoder: tradeoff analysis between decoding throughput and area. In Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on, pp 1-5.
[17] Zhang J, Fossorier M, Gu D, Zhang J. (2005) Improved min-sum decoding of LDPC codes using 2-dimensional normalization," Global Telecommunications Conference, GLOBECOM '05. IEEE, pp1187-1192.
[18] Wei H, Jianguo H, Wu F. (2010) A modified Min-Sum algorithm for low-density parity-check codes," Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on, pp 449-451.
[19] Mahankal N, Kakde S, Khobragade A. (2016) BER Evaluation of LDPC Decoder with BPSK Scheme in AWGN Fading Channel, 9(40):397-404.
[20] Sun Y, Cavallaro JR. (2012) VLSI Architecture for Layered Decoding of QC-LDPC Codes With High Circulant Weight. IEEE transactions on very large scale integration (VLSI) systems, pp 1063–8210.
[21] Kakde S, Khobragade A. (2016) VLSI Implementation of a Rate Decoder for Structural LDPC Channel Codes. Procedia Computer Science, 79:765-771.
[22] Kakde S, Khobragade A. (2016) HDL Implementation of an Efficient Partial Parallel LDPC Decoder Using Soft Bit Flip Algorithm. International Journal of Control Theory and Applications, 9(20):75-80.
[23] Kakde S, Khobragade A, Ekbal Husain MD. (2016). FPGA Implementation of Decoder Architectures for High Throughput Irregular LDPC Codes. Indian Journal of Science and Technology, 9(48):1-6.
[24] Scholl S, Schläfer P, When N. (2016) Saturated min-sum decoding: An “afterburner” for LDPC decoder hardware. In Design, Automation & Test in Europe Conference & Exhibition (DATE), pp 1219-1224.