International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 9 (2015) pp. 24201-24214

http://www.ripublication.com

Hybrid LDPC Decoder For High Error Detection and Correction Applications

M.Revathy[1], R.Saravanan[2]
[1]Assistant Professor, Department of ECE, PSNA College of Engineering and Technology,
Tamil Nadu 624622, India
Mail id: revathim@psnacet.edu.in
[2]Director, RVS Educational Trust Group of Institutions, Tamil Nadu 624005, India

Abstract

In this paper, a low-power high-efficient Hybrid LDPC Decoder Architecture for high error detection and correction applications is proposed. The majority logic circuit is integrated with LDPC decoder in order to reduce the maximum number of errors. The existing architecture requires a large decoding time. For memory applications, this increases the memory access time. The proposed hybrid method detects for errors in a word in the first iteration of majority logic decoding. If no errors are detected, then the decoding stops and remaining iterations are not performed. The scope of this paper is to reduce the decoding latency and hardware complexity by ending the decoding process if errors are not detected. The architecture is synthesized on Xilinx 9.2i and targeted to 90nm device. Synthesis report shows that the proposed architecture reduces the decoding latency and power consumption when compared to the conventional architecture design.

Keywords: Error correction codes, Iterative decoding, Linear code, Parity check codes, Turbo codes.

Introduction

Low density parity-check (LDPC) codes are a class of linear block codes which provide near-capacity performance on a large collection of data transmission and storage channels. LDPC codes have excellent performance for high speed data transmission and low complexity. However, moderate-length or short length binary LDPC codes have been shown to have an early error floor and degraded decoding performance. These codes have been implemented in various standards such as WiMax (IEEE 802.16) and other high speed applications, where parallel

implementations of iterative message-passing algorithms are ideally used in LDPC decoding.

The iterative decoding process of LDPC codes consists of two main steps: 1) To compute independent messages proportional to the aposteriori probability distributions of the code bits; and 2) communicating the messages. The complexity present in both steps depends on the message communication with respect to the process of computing the messages. The communication mechanism for LDPC codes is defined by a pseudorandom bipartite graph and is internal with respect to message computation or internal interleaver, whereas the computational complexity is very low. The complex interconnects and stringent memory requirements in a serial decoder implementation, restricts the LDPC codes to be used in low latency and power-dependent applications.

LDPC codes are suitable for iterative decoding. LDPC implements parallelism in the decoding process thereby achieving high decoding throughput. Several methods on LDPC decoding mainly focuses on floating point arithmetic or infinite precision. However, hardware implementations of the decoding algorithms for LDPC codes must deal with quantization effects in a fixed-point realization.

This paper presents a novel design of LDPC decoder to detect and correct errors in low power applications. The remaining sections are organized as follows: Section 2 exhibits the related works. In section 3, the design of proposed hybrid LDPC decoder is introduced. Section 4 discusses the results of the proposed decoder and section 5 concludes the work.

Related Works

Reference [1] investigated the limits of communication systems with noisy decoders. The author studied the motivation to determine the working efficiency of error control codes when the decoders are faulty. The second reason was to determine the fundamental limits for processing unreliable signals with unreliable computational devices. The results provided an achievability result for reliable memory systems constructed from unreliable components.

An efficient log-likelihood-ratio-based belief-propagation (LLR-BP) decoding of LDPC codes and its reduced-complexity versions have been presented in [2]. Two main approaches for simplified check-node updates by incorporating either a normalization term or an additive offset term in the BP-based approximation were presented. Their results showed that both offset and normalized BP-based decoding algorithms achieved a performance very close to that of BP decoding simultaneously providing advantages for hardware implementation.

Reference [3] proposed high-throughput memory-efficient decoder architecture for LDPC codes based on turbo decoding algorithm. A new merged-schedule merge-passing algorithm was also proposed to reduce the memory overhead of their algorithm for low to moderate-throughput decoders. Moreover, a parallel soft-input–soft-output (SISO) message update mechanism was proposed. Simulation results showed that their algorithm attained a throughput of 1.92Gbps for a frame length of

2304 bits, power saving of 89.13% and 69.83% reduction in silicon area with a reduced interconnect length of 60.5%.

The various methods for constructing Block-Circulant (BC) Reed-Solomon-Based Low-Density Parity-Check (RS-LDPC) codes were investigated in [4]. Their system resulted in a BC form of a parity-check matrix from a random parity-check matrix for RS-LDPC codes. A decoder architecture and switch network for BC-RS-LDPC code was developed based on the new BC parity-check matrix with high performance to demonstrate the efficiency of the technique. Their results showed that the designed decoder consumed 1.3M gates at an operating frequency of 450MHz and achieved a data throughput of 41Gbps with 8 iterations.

The VLSI architectures for LDPC decoders for low power and low voltage applications were studied in [5]. They implemented the algorithm on a bit-serial fully-parallel LDPC decoder fabricated in a 0.13µm CMOS process and evaluated the power consumption. The prototype decoded with 10.4pJ/bit/iteration, whilst performing within 3dB of the Shannon limit at a BER of 10-5 and total throughput of 3.3Gbps.

Proposed Decoder Design

The decoder designed using our method consists of three processing elements-Check Node Unit (CNU), Variable Node Unit (VNU) and Majority Logic Circuit (MLC) as shown in Fig. 1. The CNUs and VNUs are connected through the routing network. The input and output of the check node unit are denoted by the connectivity given by parity-check matrix. The outputs taken from the VNU are given to the MLC after the required number of iterations has been completed. One step Majority Logic Decoding (MLD) corresponds to the decoder for the Euclidean geometry low density parity check (EG-LDPC) code with N=15. First, the data block is loaded into the registers and then the check equations are calculated. If a majority of them has a value of one, the last bit is inverted and all other bits are cyclically shifted. All these operations constitute a single iteration. After N iterations are completed, the bits are in the same position in which they were loaded. Every bit could be corrected only once during the process. A long decoding time is required if N is large. The following properties of check equations have to be followed:

1) All the equations must include variables whose values are stored in the last register [P15 in Fig. 1].
2) The remaining registers are included in at most one of the check equations.

In MLD, if errors are detected in the first few iterations, the decoding process can be stopped without carrying out the remaining iterations. In the first iteration, if at least one of the check equations is affected by an odd number of bits in error, then errors will be detected. In the second iteration, as bits are cyclically shifted by one position, errors will affect other equations such that some undetected errors in the first iteration will be detected. As the iterations progress, all the errors will be ultimately detected.

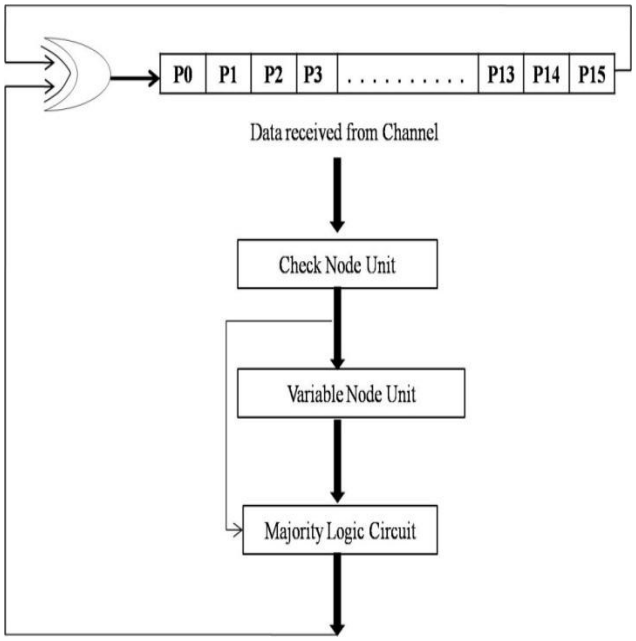


Figure 1: Proposed LDPC Decoder Architecture

Conventional LEDCC decoder

The basic Logic error detection and correction circuit (LEDCC) decoder is a simple but efficient decoder which corrects multiple random bit-flips based on the parity check equations. The LEDCC circuit in Fig. 2 consists of a linear feedback shift registers, majority gate, XOR matrix and XOR for codeword bit correction. The input signal from the linear feedback shift register is shifted from end to end and the intermediate values at each tap are used to determine the results {Bj} of the check sum equations from the XOR matrix.

The output signal is obtained at the final tap as the decoded value of input. The register contents are rotated and the same processes are continued until all codeword bits are processed. It is checked that if the codeword has been correctly decoded, the parity check sum should be zero. Other conventional designs require the number of cycles to be equal to the number of input bits, i.e. number of intermediate taps, which is a major drawback.

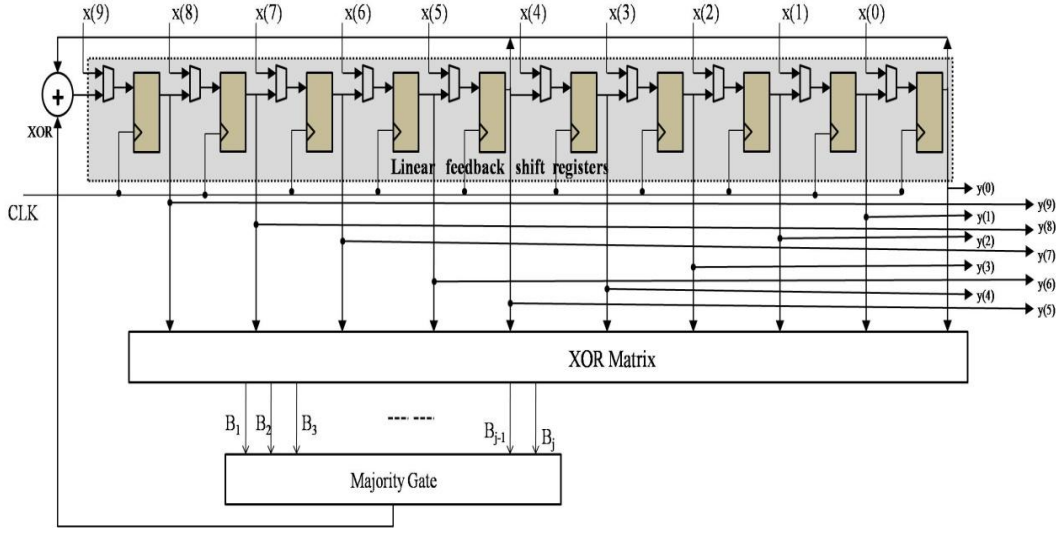


Figure 2: A Basic LEDCC Decoder

Proposed LEDCC Design

The proposed decoder design is depicted in Fig. 3. The designed LEDCC decoder has a linear feedback shift register, a majority gate for bit inversion, an XOR matrix and additionally two blocks—control unit and tristate buffers. The control unit sets a finish flag after the third cycle for no errors detected. The output tristate buffers have high impedance always unless it receives the finish signal from the control unit so that the current values of the shift register are pushed to the output y. The control unit is shown in Fig. 4 and it manages the error detection process.

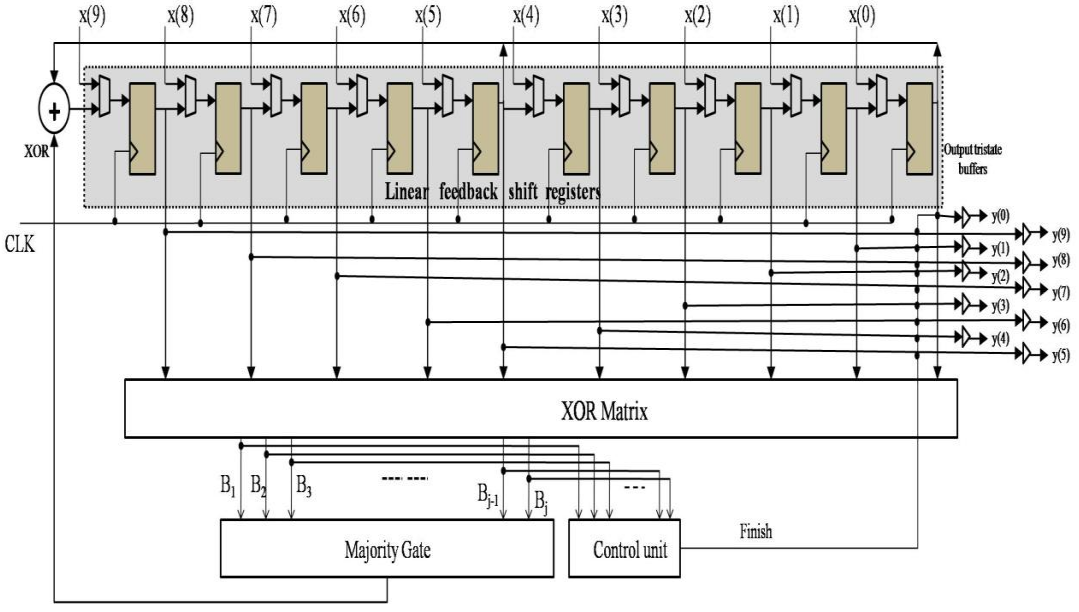


Figure 3: LEDCC decoder with N-tap shifter

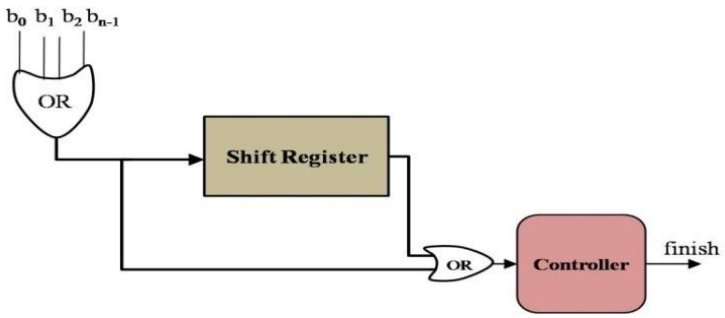


Figure 4: Control Unit

Design of proposed Check Node

The architecture of the check node module of LDPC decoder is shown in Fig. 5. It is used in determining the strength of the received signal against noises in the channel. Four directional difference vectors are calculated with twelve |SUB|, four ADD, and four sorter & concatenator modules. Then, the value of smallest difference is decided by using the sorter & concatenator units. In the proposed design, ADD and subtractor units are required for the design of check node.

Since, the gate count or hardware complexity of one multiplier or division is much more than one adder or subtractor unit, all multipliers or divisions are replaced by adder or subtractor unit, thereby reducing the hardware cost.

After addition and subtractions are determined on received sequences, the sorting and concatenation (Fig. 6) is performed over these sequences in order to compute the response of the check node unit in LDPC decoder. Each check node block produces 6 bit length of sequence and thus it produces 24 bit length sequence.

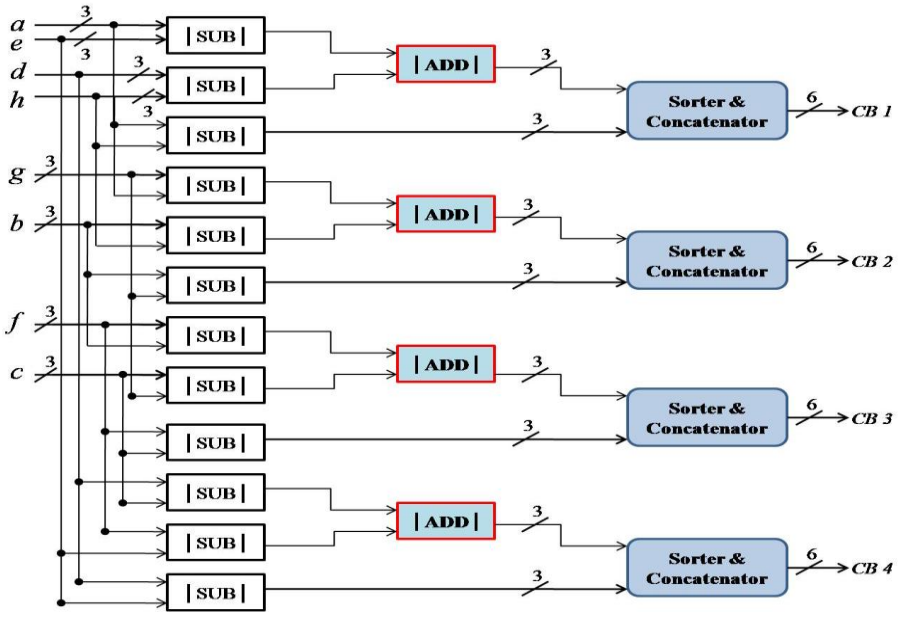


Figure 5: Check Node Architecture

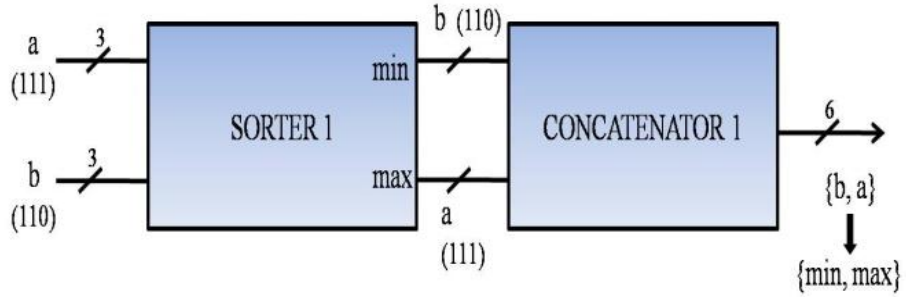


Figure 6: Internal of Sorter & Concatenator Unit

Design of Proposed Variable Node

The variable node unit architecture shown in Fig. 7 computes the hard decision vector x. This vector is routed through the routing network to the check node block (CNB). The routing between two nodes has single-bit value and the routing network size is smaller compared to that used in sum product algorithm. The variable node processor unit is comprised of a flip-detection circuit, line buffer, multiplexed adder and concatenator. Initially the received signal y from check node unit is fed to the Variable node block (VNB) through a register-feedback assembly. The received values are 6-bit signed-magnitude (SM) values. Let $[sn : mn]$ be n^{th} 4-bit value provided to the n^{th} VNB, where sn denotes hard decision value and mn denotes the magnitude of the same.

The SM makes the correlation calculation simple to be implemented. The correlator circuit consists of an inverter followed by a 1-bit multiplexer. The proposed VNA unit consists basic multiplexer, line buffer, summer and concatenator modules. The block datas received from CNU unit are divided in to four subblocks. The first three submodule blocks are processed by adder module and last submodule block is processed directly by multiplexer unit. The control signal for multiplexer module acts as a select line in this unit, which is generated and illustrated in Fig. 8.

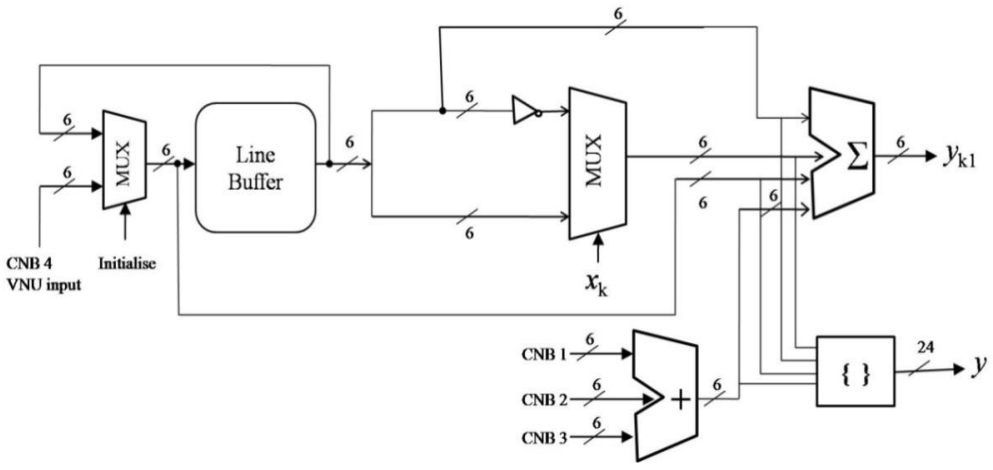


Figure 7: Variable Node Architecture

Fig. 8 explains the generation of control signal (x_k). The signed bit input is applied to the shift operator. The multiplexers and line buffers help in producing the control signal, which is fed to the VNU. The use of simple computational methodologies along with less row and column weights reduces the operations thereby, resulting in significantly less power consumption.

The subtraction unit receives six bits from variable node architecture and line buffer unit and thus produces single sign bit which satisfies the following constraint:

x_k =1 if (y_{k1}-output of line buffer) ≥0 (1)

x_k =0 if (y_{k1}-output of line buffer) <0 (2)

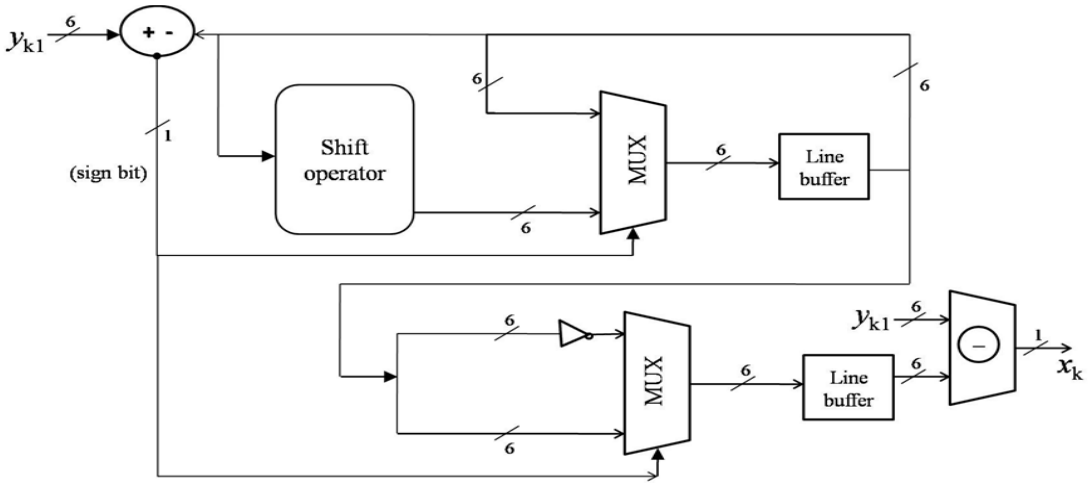
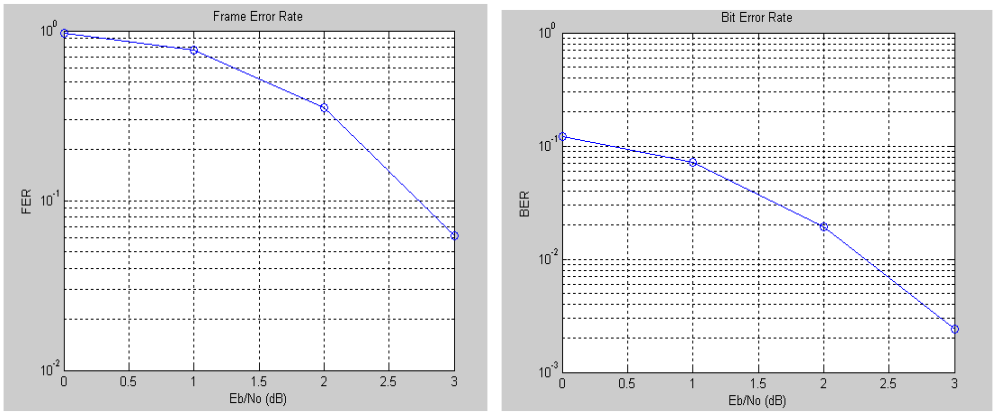


Figure 8: Generation of control signal x_k

Results and Discussions

A number of performance evaluation and resource utilization parameters are determined for the proposed LDPC decoder. The present research is focused on the design and development of pipelined LDPC decoder for low power applications. The simulation results shown in Fig. 9 illustrates the relationship between SNR (Signal to noise ratio)–FER (Frame error rate) and SNR–BER (Bit error rate).



(a) (b)

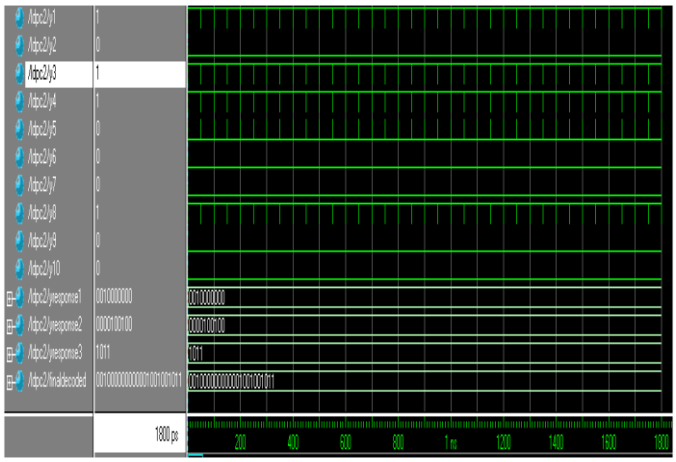
Figure 9: Result showing plot between (a) SNR and FER, (b) SNR and BER.

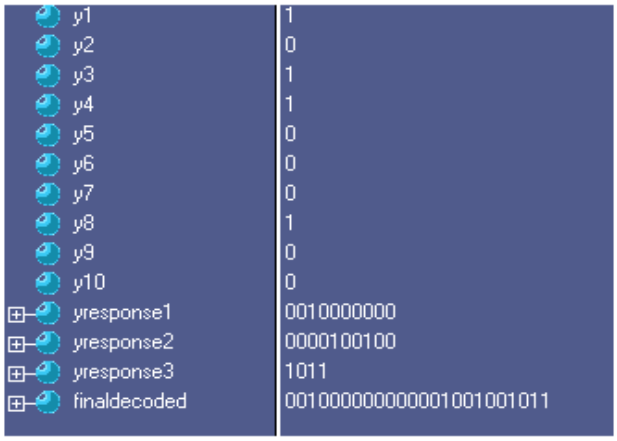
In Fig. 10, the variable count1 indicates the maximum number of errors detected in the received code words. In this work, the maximum number of detectable errors in the received code word is 4. The RTL schematic view and Technology schematic view and the Chip layout of proposed LDPC decoder IC are shown in Fig. 11.

The parameters considered for investigation include number of slices (S), number of LUT's (Look up table), Slice latches (SL), gate counts (GA), Power consumption (PC), and Area Utilization. All these parameters are tabulated in Table 1 and graphically plotted in Fig. 12.

Table 1: Performance Analysis of Hardware Utilization

Performance parameters	**Proposed Architecture**
Slices	56
LUTs	100
Gate Count	2304
IOBs	48





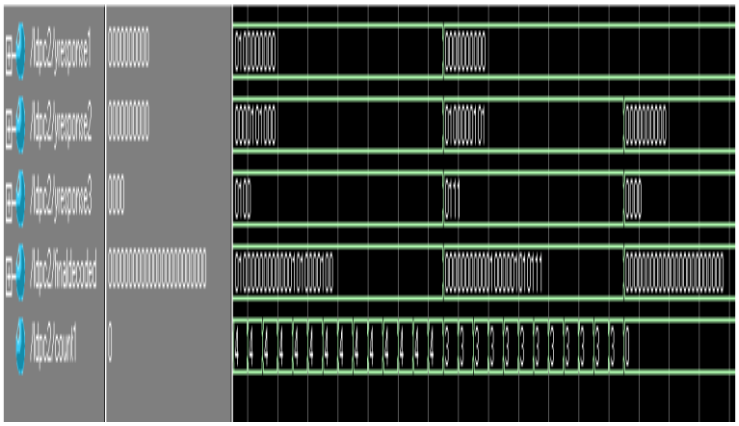
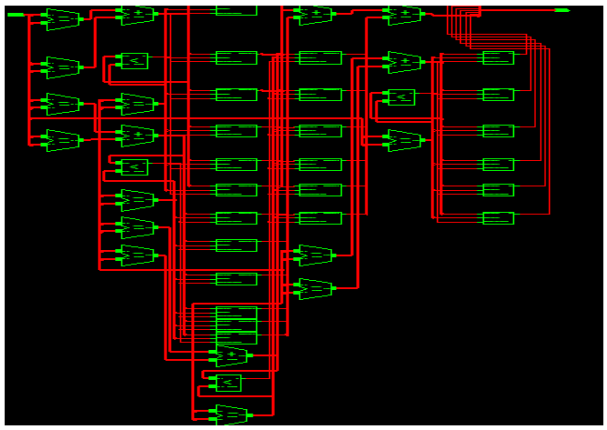
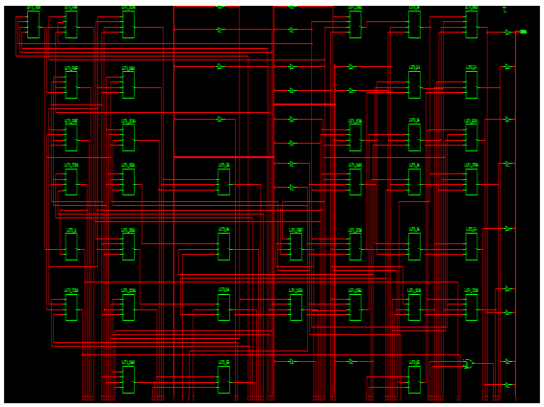


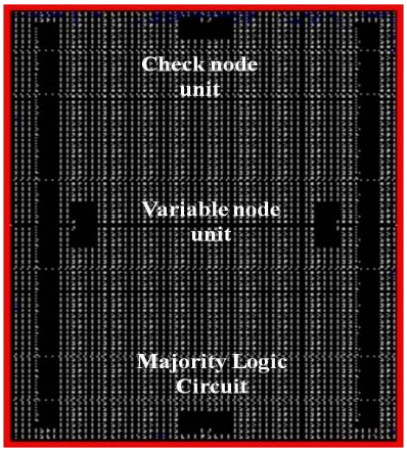
Figure 10: Error Detection Process In Proposed Method



(a)



(b)



(c)

Figure 11: Simulation results of proposed technique. (a) RTL schematic view, (b) Technology schematic view and (c) Chip layout of proposed decoder IC.

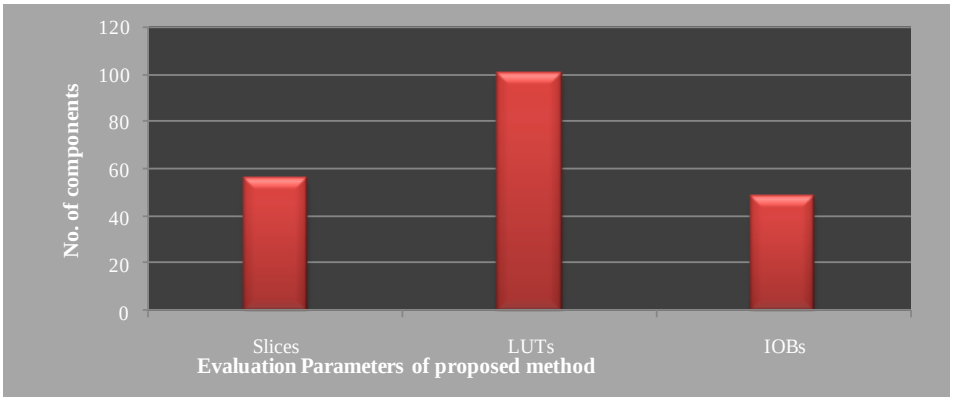


Figure 12: Graphical Plot of Hardware Utilizations

The proposed LDPC decoding architecture is synthesized using Xilinx project navigator version 9.2i tool. The proposed architecture provided decoding latency of 28ns for both check node and variable node implementation.

Sangmin *et al*. (2013) [8] achieved the decoding latency about 16.32ns whereas Spagnol *et al*. (2009) [9] achieved decoding latency about 50.05ns.The proposed LDPC architecture is performed as superior in this way. Also, the proposed architecture consumed 4-input LUTs about 100. Sangmin *et al*. (2013) [8] consumed 6422 LUTs and Spagnol *et al*. (2009) [9] consumed 12924 LUTs with respect to their various designing methodologies. The various comparisons of proposed technique in terms of several parameters are illustrated in Table 2, Table 3 and Table 4.

Table 2: Performance Comparison in terms of Latency and LUTs

Methodology	**Decoding Latency (ns)**	**LUTs**
Proposed	16.32	100
Sangmin kim *et al*. (2013) [8]	56.36	6422
Spagnol *et al*. (2009) [9]	50.05	12924

Table 3: Performance Comparison of Spartan and Virtex Families

Family	**Power Consumption (mW)**	**Latency (ns)**
Spartan	81	16.32
Virtex	151	17.72

Table 4: Performance Analysis based on Error Detection and Correction

Methodology	**Error detecting capabilities**	**Error correcting capabilities**
Proposed work	4	4
MLDD (Liu *et al*.) [6]	3	3

Conclusion

The hybrid LDPC architecture integrated with majority logic algorithm is proposed in this paper. The proposed Hybrid Architecture detects and reduces the maximum number of errors in the received code words. The simplified architecture for variable and check node is designed for low power applications. The simulation and synthesis result shows that the proposed hybrid LDPC Architecture consumed 16.32ns of decoding latency which reduces the decoding latency rate about 28.9% whereas compared with conventional methods.

References

[1] Varshney, L. R., 2011, "Performance of LDPC codes under faulty iterative decoding," IEEE Transactions on Information Theory, 57(7), pp. 4427–4444.

[2] Chen, J., Dholakia, A., Eleftheriou, E., Fossorier, M. P. C., and Hu, X. Y., 2005, "Reduced-Complexity decoding of LDPC codes," IEEE Transactions on Communications, 53(8), pp. 1288–1299.

[3] Mansour, M. M., and Shanbhag, N. R., 2003, "High-Throughput LDPC decoders," IEEE Transactions on Very Large Scale Integration Systems, 11(6), pp. 976–996.

[4] Seong-In, H., and Lee, H., 2013, "Block-circulant RS-LDPC code: Code construction and efficient decoder design," IEEE Trans. Very Large Scale Integration Syst., 21, pp. 1337–1341.

[5] Darabiha, A. Carusone, A. C., and Kschischang, F. R., 2008, "Power reduction techniques for LDPC decoders," IEEE Journal of Solid-State Circuits, 43(8), pp. 1835–1845.

[6] Liu, S. F., Reviriego, P., and Maestro, J. A., 2012, "Efficient majority logic fault detection with difference-set codes for memory applications," IEEE Transactions on Very Large Scale Integration Systems, 20(1), pp. 148–156.

[7] Swapna, S., and Anbuselvi, M., 2012, "Design and analysis of iterative decoder using wave pipelining," Proc. International Conference on Computing and Control Engineering, pp. 1–4.

[8] Sangmin, K., and Sobelman, G. E., 2013, "Scaling, offset and balancing techniques in FFT-based BP nonbinary LDPC decoders," IEEE Trans. Circuits Syst. II: Express Briefs, 60, pp. 277–281.

[9] Spagnol, C., Popovici, E., and Marnane, W., 2009, "Hardware implementation of GF(2^m) LDPC decoders," IEEE Trans. Circuits Syst. I, 56, pp. 2609-2620.

[10] Reviriego, P. J., Maestro, A., and Flanagan, M. F., 2013, "Error detection in majority logic decoding of Euclidean geometry low density parity check (EG-LDPC) Codes," IEEE Trans. Very Large Scale Integration Syst., 21, pp. 156–159.

[11] Fossorier, M. P. C., Mihaljevic, M., and Imai, H., 1999, "Reduced complexity iterative decoding of low density parity check codes based on belief propagation," IEEE Trans. Commun., 47(5), pp. 673–680.

[12] Zhao, J. Zarkeshvari, F., and Banihashemi, A. H., 2005, "On implementation of min-sum algorithm and its modifications for decoding low-density parity-check (LDPC) codes," IEEE Trans. Commun., 53(4), pp. 549–554.

[13] Rejimon, T., Lingasubramanian, K., and Bhanja, S., 2009, "Probabilistic error modeling for nano-domain logic circuits," IEEE Trans. VLSI Syst., 17(1), pp. 55–65.

[14] Berron, C., Glavieux, A., and Thitimajshima, P., 1993, "Near Shannon limit error correcting coding and decoding: Turbo codes," Proc. IEEE Int. Conf. on Communications, pp. 1064–1070.

[15] Richardson, T. Shokrollahi, M., and Urbanke, R., 2001, "Design of capacity approaching irregular low-density parity-check codes," IEEE Trans. In- form. Theory, 47(2), pp. 619–637.

[16] Chung, S. Y., Forney, G. D., Richardson, T. J., Urbanke, R., 2001, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limits," IEEE Commun. Lett., 5(2), pp. 58–60.

[17] Mansour M. M., and Shanbhag, N. R., 2002, "Turbo decoder architectures for low-density parity-check codes," Proc. IEEE GLOBECOM, Taipei, Taiwan, R.O.C., pp. 1383–1388.

[18] Hu, X. Y., Eleftheriou, E., Arnold, D. M., Dholakia, A., 2001, "Efficient implementations of the sum-product algorithm for decoding LDPC codes," Proc. IEEE GLOBECOM, 2, pp. 1036–1036E.

[19] Howland, C., and Blanksby, A., 2001, "Parallel decoding architectures for low density parity check codes," Proc. of IEEE Int. Symp. on Circuits and Systems, Sydney, Australia, pp. 742–745.

[20] Mansour, M. M., and Shanbhag, N. R., 2002, "Construction of LDPC codes from Ramanujan graphs," Proc. of the 36th Annu. Conf. on Information Sciences and Systems 2002 (CISS'02), Princeton, NJ.

[21] Boutros, J., Pothier, O., and Zemor, G., 1999, "Generalized low density (Tanner) codes," Proc. ICC'99, Vancouver, Canada, pp. 441–445.

[22] Bahl, L. R., Cocke, J., Jelinek, F., and Raviv, J., 1974, "Optimal decoding of linear codes for minimizing symbol error rate," IEEE Trans. Inform. Theory, IT-20, pp. 284–287.

[23] Eleftheriou, E., Mittelholzer, T., and Dholakia, A., 2001, "Reduced-complexity decoding for low-density parity-check codes," Electron. Lett., 37, pp. 102–104.

[24] Gallager, R. G., 1962, "Low-density parity-check codes," IRE Transactions on Information Theory, 8(1), pp. 21-28.

[25] Naeimi, H., and DeHon, A., 2007, "Fault secure encoder and decoder for memory applications," Proc. IEEE Int. Symp. Defect Fault Toler. VLSI Syst., pp. 409–417.