ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, *Vol 9(30), DOI: 10.17485/ijst/2016/v9i30/99033, August 2016*

Comparison of Technologies for the Implementation of SBF Decoder for Geometric LDPC Codes

J. Chinna Babu[1]* , C. Chinnapu Reddy[2] , M. N. Giri Prasad[3]

[1]A. I. T. S Rajampet, Kadapa - 516126, Andhra Pradesh, India;
jchinnababu@gmail.com
[2]TEQIP-II, SPFU AP, Hyderabad - 500 063, Andhra Pradesh, India;
ccreddyece@gmail.com
[3]JNTU, Anantapuram - 515002, Andhra Pradesh, India;
mahendragiri1960@gmail.com

Abstract

Background/Objectives: The main aim of the proposed design is to optimize the consumption in chip area by improving the error performance by detection and correction. Generally, it is difficult to implement the VLSI based decoding of Geometric LDPC codes because of high complexity and large memory requirements. **Methods/Statistical Analysis:** In this proposed design architecture we have considered the Soft-Bit Flipping (SBF) algorithm employed here utilizes reliability estimation to improve error performance and it has advantages of Bit Flipping (BF) algorithms. **Findings**: This proposed design architecture is compared for different technologies using Leonardo spectrum software in Mentor Graphics Tools. We can also obtain the area and delay reports using this tool and optimization of the design is being proposed. **Application/Improvement:** In future works, this algorithm can be improved with still more security level by having a trade off between performance and data transmission. It can also enhanced by implementing it in real time applications for data decoding and correction, for smaller size datum.

Keywords: IOB, Leonardo Spectrum, MG (Mentor Graphics), SBF (Soft Bit Flipping)

1. Introduction

Low-Density Parity-Check (LDPC) codes have been developed by Robert Gallager is of great interest since late 1990s because of the improved error performance[1]. These are error correcting codes[2]. The Bit-Flipping (BF) algorithms that are developed in the initial stage of the LDPC history are based on hard decision scheme. Though SPA is gives the best error performance but due to its high complexity it is difficult to implement it in hardware. In contrast the BF algorithm which has low complexity presents even poor error performance than Sum Product Algorithm (SPA)[3]. Combining both BF and SBF algorithms a hybrid decoding scheme has been proposed to reduce decoding duration. In this work, comparisons are made for the SBF decoder for different technologies, and the hybrid decoding procedure is explained clearly.

2. Soft Bit Flipping (SBF) Algorithm

The underlying structure of SBF algorithm is that of the MWBF algorithm[4,5] using pseudo marginalization but employs improved flipping criteria to attain better error performance.

*Author for correspondence

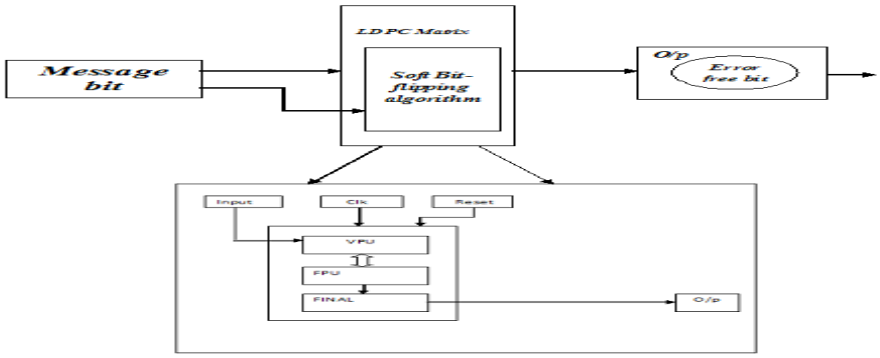

Figure 1. Block diagram of SBF decoder for geometric-LDPC codes.

3. Decoding Algorithms

The following steps explain the procedure for the SBF decoding algorithm:

Step 1: Syndrome bits (Parity check sums) are computed. If the entire syndrome bits are zero indicates that all the parity check equations are satisfied, then decoding is stopped.

Step 2: Check for the number of parity check equations that are not satisfied for each code bit position, denoted $f_{i,}$ i=0,1,…. n-1.

Step 3: The set Ω of bits with largest $f_{i\ are}$ identified.

Step 4: The bits in set Ω should be flipped.

Step 5: Steps from 1 to 4 should be repeated until entire parity equations meet the condition in 1st step (in this case, we stop the iteration in step 1) or a predefined maximum number of iterations is reached[6–8].

4. Architectures for Soft-Bit Flipping decoder

The other possibilities of SBF decoder architectures are described in this section. The means to minimize the hardware area and to maximize the throughput of baseline parallel architecture are presented clearly. The block diagram of SBF decoder for Geometry based-LDPC codes[9] is shown in Figure 1.

4.1 SBF Decoder Architecture

SBF decoder architecture consists of VPU (Variable Processing Unit), FPU (Floating Point Processing Unit) and AND Matrix. It is shown in Figure 2.

4.2 Serial Architecture

The serial SBF decoder comprises of shift registers of two bit for the storage of variable and check node values. Two processing units one for Variable Node (VNU) and one for Check Node (CNU) are used. As decoding starts[10,11], all the received signals will be stored in the variable nodes. Then the VNU evaluates a parity-check from the variable nodes, and store it to the respective check node. VNU continues its operation till all the check nodes are revised, meanwhile all the registers are moved by one stage for every cycle. Soon after the updating of check node is completed, the CNU sums up the connected three check nodes and compare the result with the flipping thresholds; as a result the flipping strength for the output variable node is generated. Each variable node is updated by adding the flipping strength if the current variable node is negative or by subtracting it otherwise.

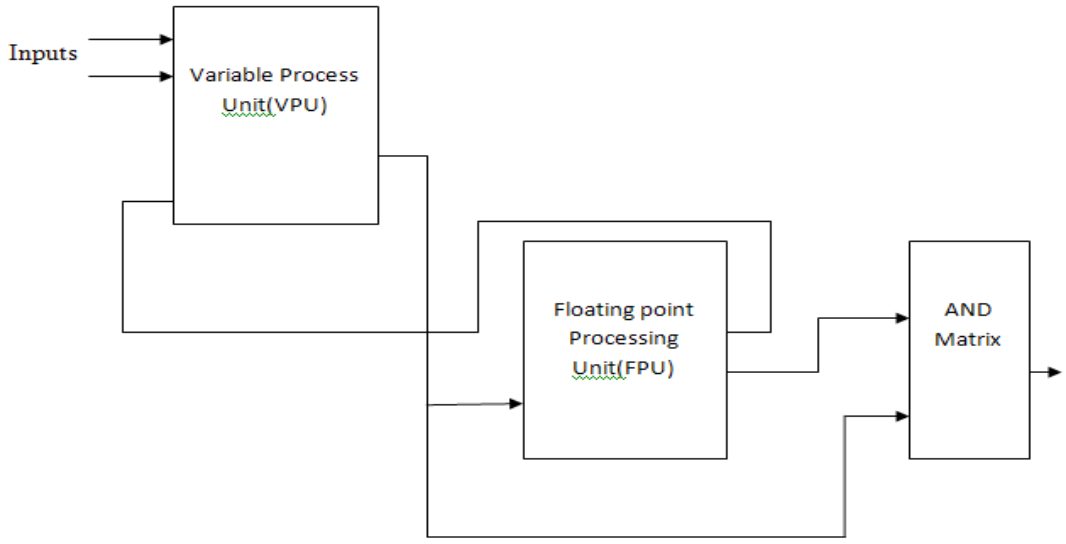

Figure 2. SBF decoder with VPU and FPU.

Figure 3 shows the serial hybrid SBF decoder that is generated in this paper. The decoder consists of an buffer input (I buff), a rollback buffer (R buff), an buffer output (O buff), check nodes, variable nodes, a flip unit, a Variable Node processing unit (VNU), and a Check Node[12] processing unit (CNU), where shift registers are used to implement the buffers and nodes.

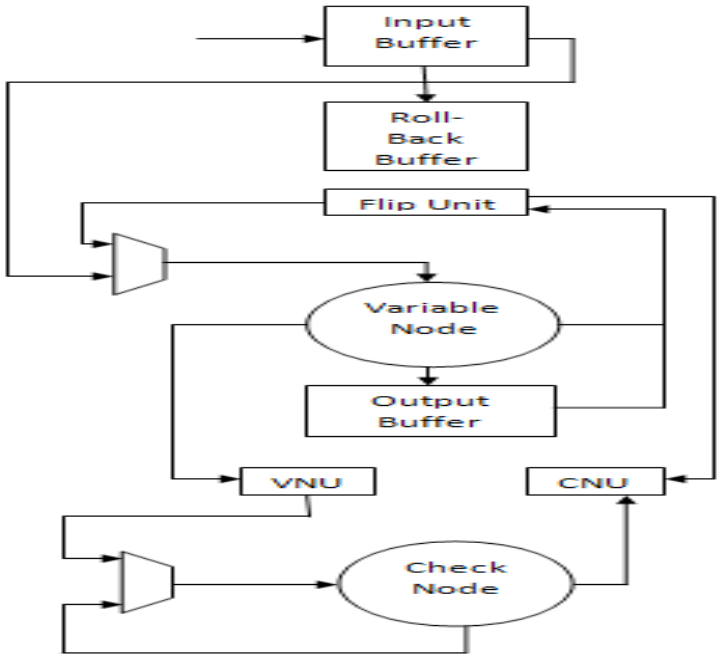

Figure 3. Block diagram of hybrid SBF decoder using serial architecture.

5. VLSI Realization Results and its Comparison

An efficient decoder was synthesized and compared and the results were tabulated shown in Table 1.

(a) Technology: AMI 0.5um

The schematics for various technologies are obtained using Mentor Graphics Tools and the results for different technologies are shown in the figures numbered from Figure 4 to Figure 12. The technology schematic and the critical path schematics vary from one technology to

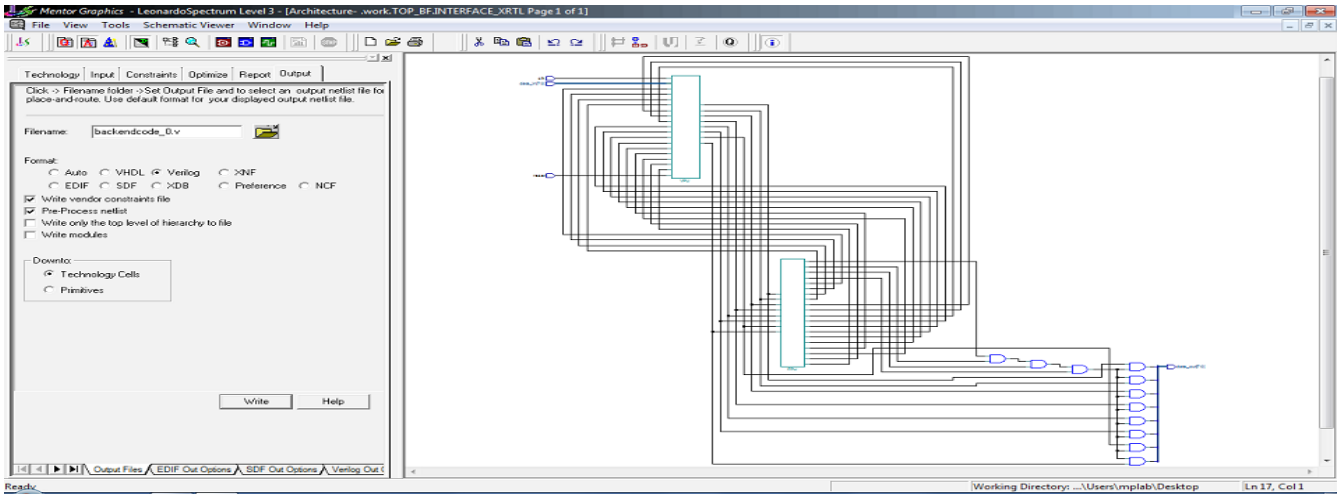

Figure 4. RTL schematic.

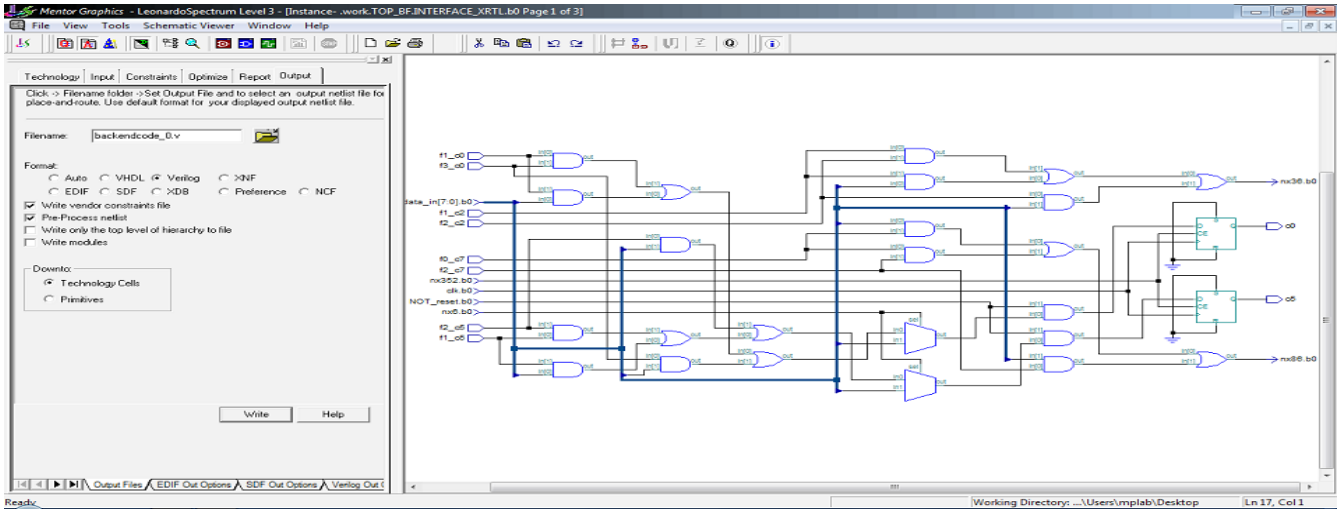

Figure 5. Internal schematic of VNU.

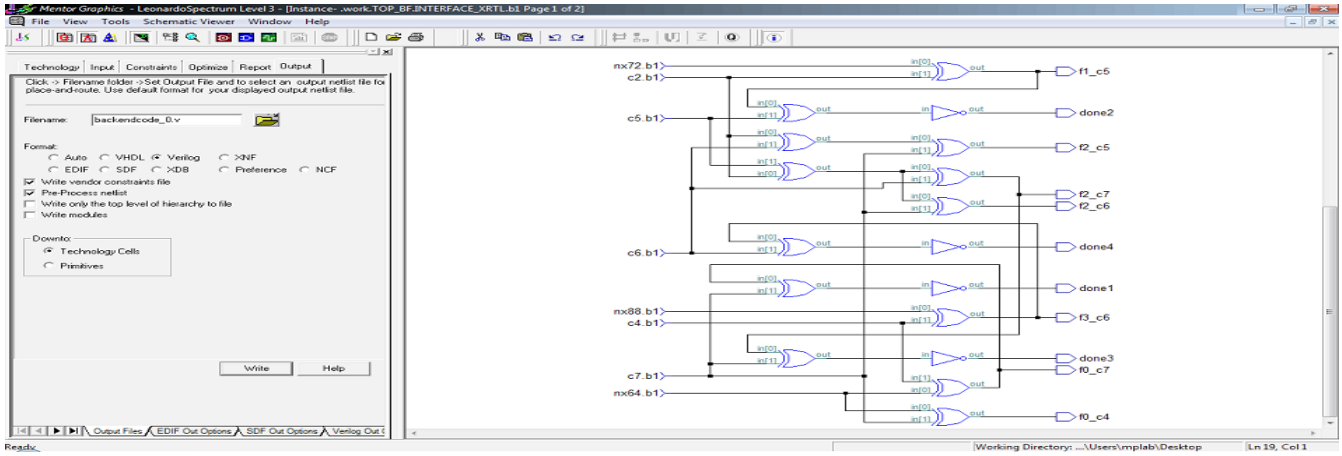

Figure 6. Internal schematic of FPU.

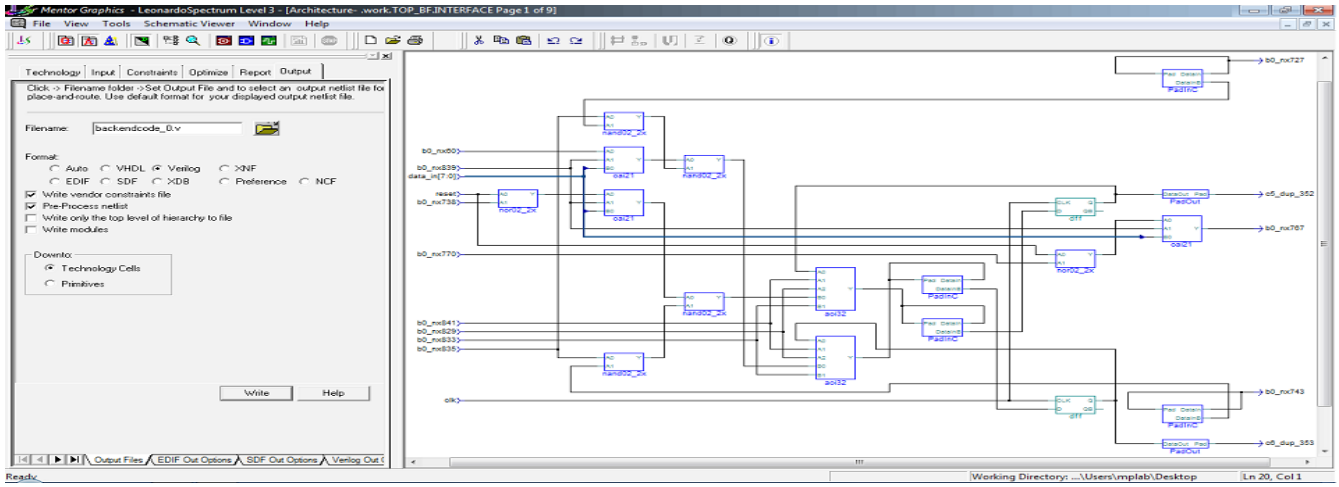

Figure 7. Technology schematic.

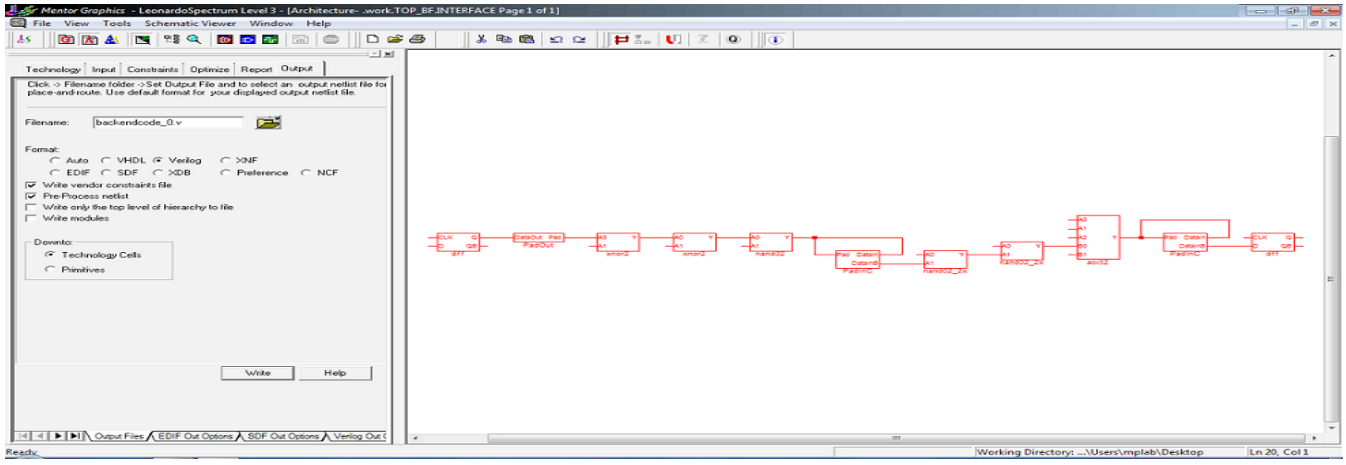

Figure 8. Critical path schematic.

(b) Technology: AMI 1.2um.

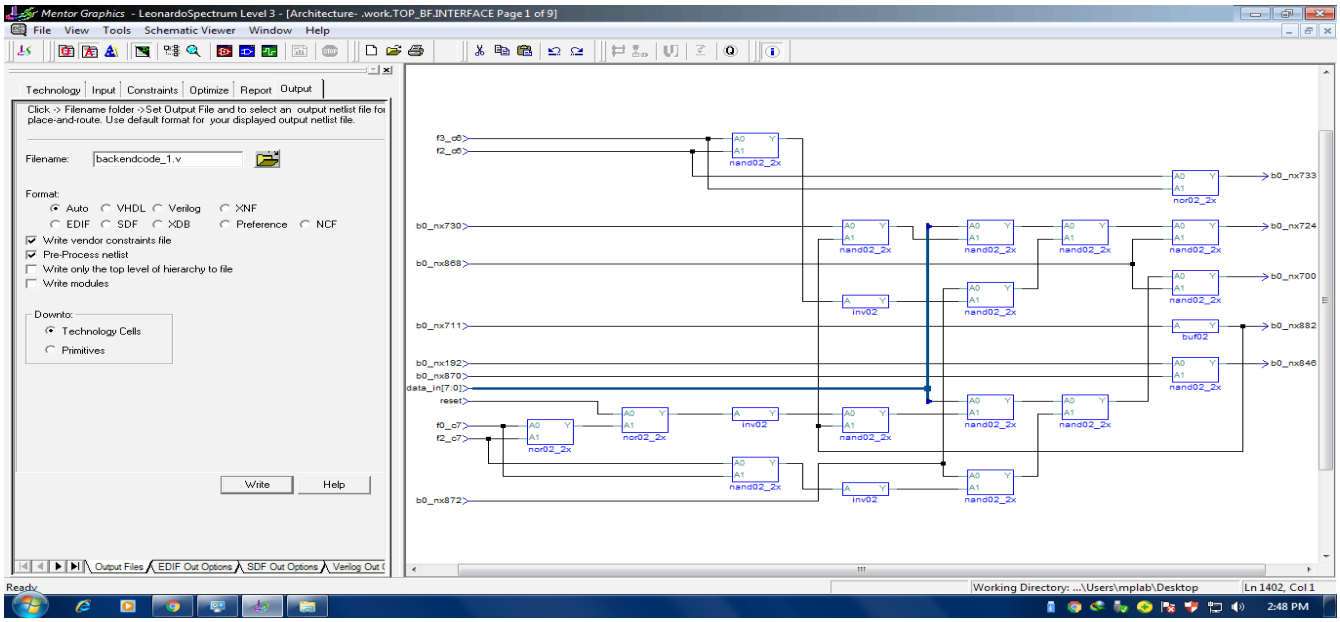

Figure 9. Technology schematic.

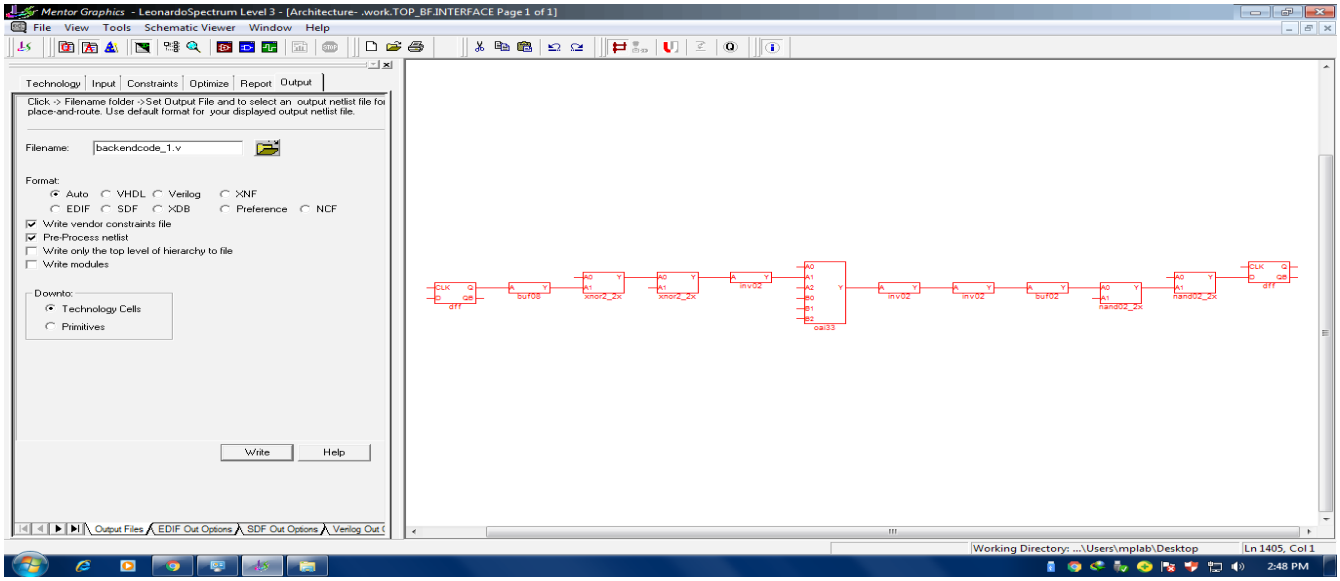

Figure 10. Critical path schematic.

(c) Technology: TSMC 0.35um

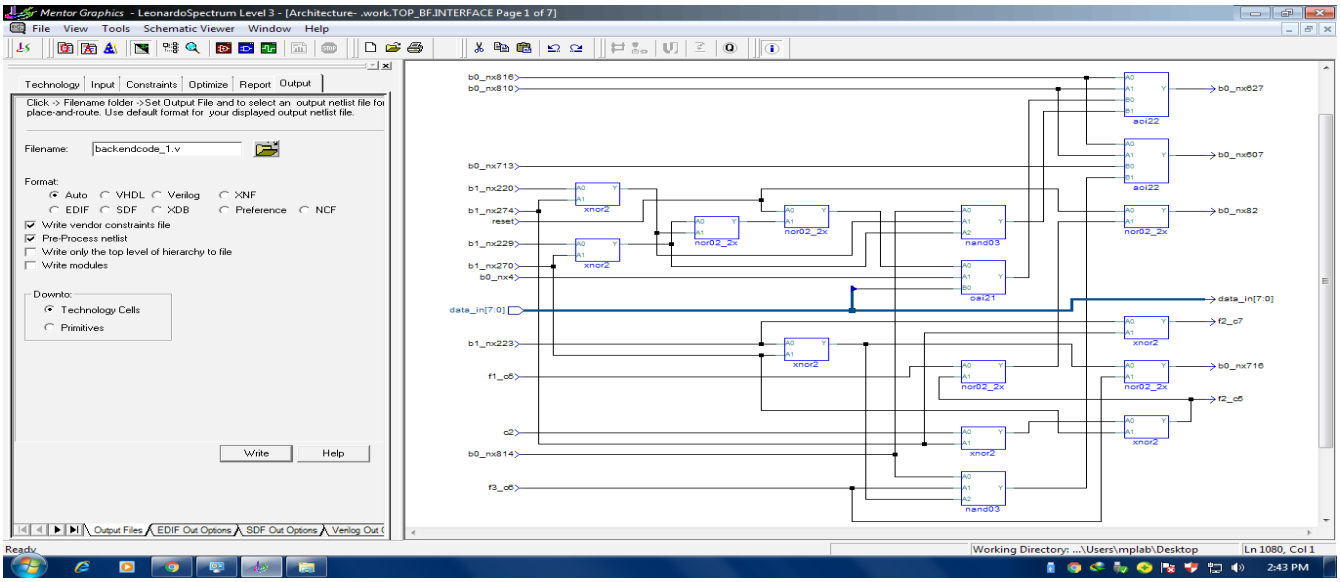

Figure 11. Technology schematic.

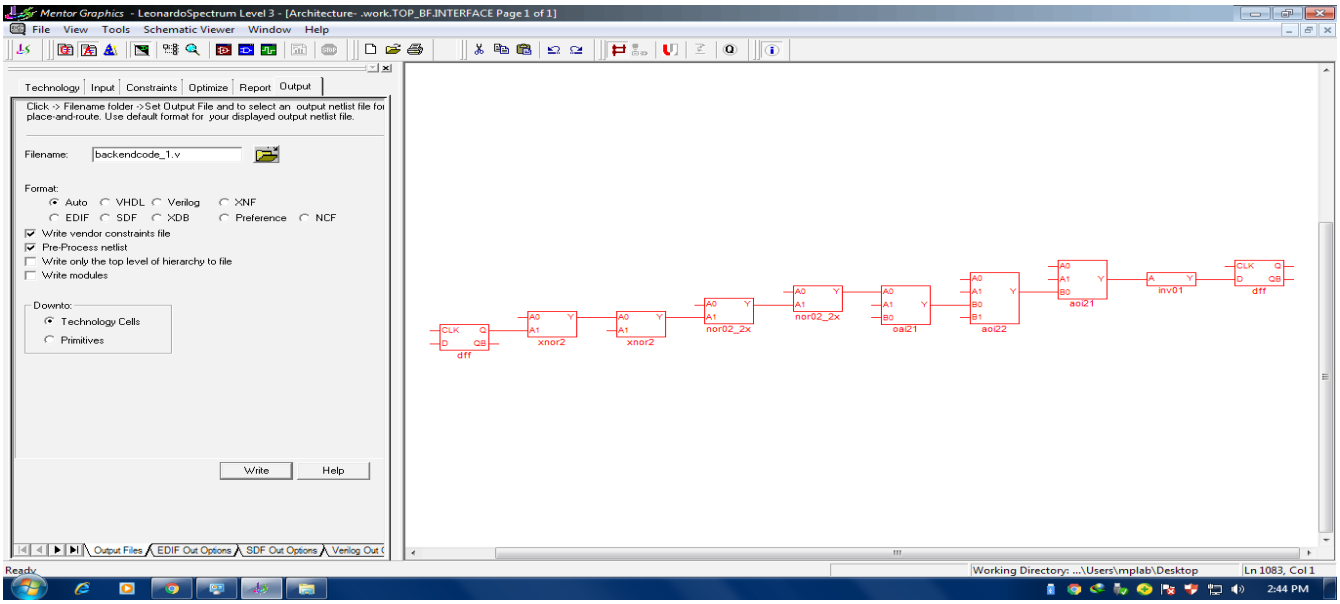

Figure 12. Critical Path Schematic.

Table 1. Implementation and comparison Results for various technologies

Technology	Clock Frequency	Area Report	Delay Report
AMI 0.5um	25MHz	Number of gates= 194	Slack time= -6.98
	20MHz	Number of gates= 194	Slack time= 3.02
AMI 1.2um	25MHz	Number of gates= 261	Slack time=31.98
	50MHz	Number of gates= 261	Slack time=11.98
TSMC 0.35um	25MHz	Number of gates= 202	Slack time=31.98
	200MHz	Number of gates= 202	Slack time=-3.62

other. The upgraded practical codes with area and timing optimization can be developed for large weighted LDPC codes because of the realization viability.

6. Conclusion

The proposed soft bit flipping provides marginalization scheme for reduction in hardware complexity but utilizes BF techniques to attain better error performance. For geometric LDPC codes, the reduction of hardware is of more important. By comparing with other hardware decoding algorithms for large-weight LDPC codes this decoder yields better results. The comparison results of various technologies are presented in this paper. The area and delay reports for different technologies are also provided.

7. References

1. Tanner RM. A recursive approach to low complexity codes. IEEE Trans Inf Theory. 1981; 21(5):533–47.
2. Gallager RG. Low density parity-check codes. IRE Trans Information Theory. 1962; 8(1):21–8.
3. MacKay DJ. Good error-correcting codes based on very sparse matrices. IEEE Trans Inf Theory. 1999; 45(2):399–432.
4. Kschischang FR, Frey BJ, Loeliger HA. Factor graphs and the sum–product algorithm. IEEE Trans Inf Theory. 2011; 45(2):498–519.
5. Zhang J, Fossorier. A modified weighted bit-flipping decoding of low-density parity-check codes. IEEE Commun Lett. 2004 Mar; 8(3):165–7.
6. Jiang M, Zhao C, Shi, Chen. An improvement on the modified weighted bit flipping decoding algorithm for LDPC codes. IEEE Commun Lett. 2005; 9(1):814–16.
7. Fossorier MPC, Mihaljevic M, Imai. Reduced complexity iterative decoding of low-density parity check codes based on belief propagation. IEEE Trans Commun. 1999; 47(5):673–80.
8. Palanki R, Fossorier F, Yedidia Y. Iterative decoding of multiple-step majority logic decodable codes. IEEE Trans Commun. 2007; 55(6):1099–102.

9. Cho J, Sung S. High-performance and low-complexity decoding of high-weight LDPC codes (in Korean). J Korea Inf Commun. 2009; 34(5):498–504.
10. Chen J, Dholakia D, Eleftheriou E, Fossorier MPC. Reduced-complexity decoding of LDPC codes. IEEE Trans Commun. 2005; 53(8):1288–99.
11. Liu L, Shi CJR. Sliced message passing: High throughput overlapped decoding of high-rate low-density parity-check codes. IEEE Trans Circuits Syst I. 2008; 58(11):3697–710.
12. Ablodun S, Jafar AA, Omar A, Tim OF. Near capacity irregular turbo code. Indian Journal of Science and Technology. 2015; 8(23):110–11.