

An Efficient VLSI Implementation of First Two Minimum Value Generator for Bit Serial LDPC Decoder

S. Shailaja¹, K. Vijayakanth²

¹ME VLSI Design, Srinivasan Engineering College, Pperambalur
Shailaja889[at]gmail.com

²Assistant professor, Srinivasan Engineering College, Perambalur
k.vijayakanth85[at]gmail.com

Abstract: Low Density Parity Check code has excellent error correcting capability and parallel structure, A low-complexity generator is crucially important, Because the hardware complexity of generator utilizes a significant portion of the min sum Low-Density Parity Check Decoder. To reduce hardware complexity an existing bit-serial generator that finds only one minimum value instead of two has been proposed; however they use minimum value generator to design a decoder, proposed low complexity bit-serial generator can find the exact first two minimum values, and thus can improve the Bit Error Rate performance. Moreover, the proposed generator does not suffer from any throughput loss since its latency is almost the same as that of the existing generator. The Proposed System designed using Verilog HDL and Simulated by Modelsim 6.4a and Synthesized by Xilinx tool. The proposed system implemented in Spartan 3E 3S250E PQ 208 FPGA.

Keywords: Minimum value generator, Low Density parity check decoder (LDPC), Low complexity design, Check node unit (CNU)

1. Introduction

Low density parity check codes has error-correcting capability and parallel structure. LDPC codes are used to transmit messages from noisy transmission channel and LDPC is constructed by sparse bipartite code. LDPC codes have been considered for a wide range of applications such as error correction code for flash memory [2], [3], 10GBase-T[4], DVB S2[5], data transmitter and receiver, error correction, error reduction. In min-sum algorithm, the check-node unit is used for row operations to produce the outputs. Instead of producing the actual output values, the Check Node Unit can significantly reduce the memory space by producing only the two minimum values, and signs of all output values. Low density parity check codes allow a fine-level parallel message-passing decoding. This parallelism can potentially be used to build serial Low Density Parity Check decoders [7]-[9] with Multi-Gbit/s throughput. The crucial part of the calculation n-module in LDPC decoders is a generator that finds the two minimum values over the magnitude of all variable node messages [10]. Thus, for high-speed and low-complexity LDPC decoder designs, the first-two-minimum-values generator is the key building block of Check node unit [11]. The bit-parallel generator for finding the first W ($W > 2$) minimum values relies on the bitwise analysis, where W is the number of values for finding. The hardware complexities exponentially increase with an increase in input messages [12]. Low density parity check code is a linear error correcting code, a method of transmitting a message over a noisy transmission channel. Using iterative belief propagation techniques, Low Density Parity codes can be decoded in time linear to their block length. In digital electronics, a decoder can take the form of a multiple-input, multiple-output logic circuit that converts coded inputs into coded outputs, where the input and output codes are different e.g. n-to-2n, binary-coded decimal decoders. Decoding is necessary in applications such as

data multiplexing, 7 segment display and memory address decoding. In digital electronics a decoder is a combinational circuit that converts a binary integer value to an associated pattern of output bits. They are used in a wide variety of applications, including data demultiplexing seven segment displays, and memory address decoding. There are several types of binary decoders, but in all cases a decoder is an electronic circuit with multiple input and multiple output signals, which converts every unique combination of input states to a specific combination of output states. In addition to integer data inputs, some decoders also have one or more "enable" inputs. When the enable input is negated (disabled), all decoder outputs are forced to their inactive states. Depending on its function, a binary decoder will convert binary information from n input signals to as many as 2^n unique output signals. Some decoders have less than 2^n output lines; in such cases, at least one output pattern will be repeated for different input values.

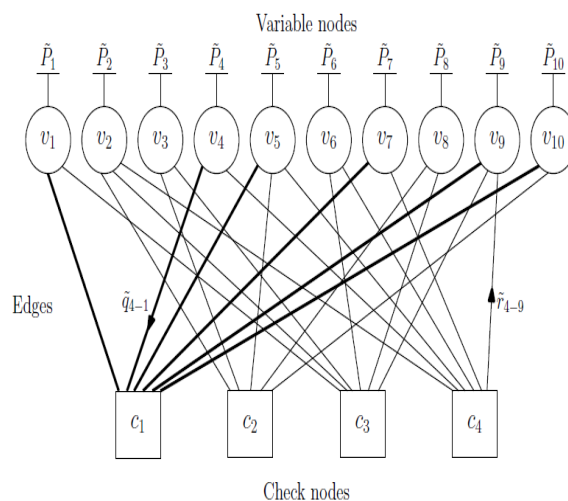


Figure 1: LDPC codes

2. BIT Serial Scheme

Incoming bit $X(n-t)$ arrives in one bit per cycle from the Most Significant Bit to the Least Significant Bit of the corresponding message x_i where t denotes the time index. Each message x_i has a status flag $f_i(t)$ which represents whether x_i is a candidate having min 1. In the beginning, the status flags are all 0 to indicate that all of the messages can be candidates for containing the minimum value. The incoming bit $x_i(n-t)$ message x_i enters the OR gate OR_i . It is fed to the OR_i gate, the OR_i output is 1. Let $N_0(t)$ denote the number of 0's among all of the unmasked incoming bits of all messages at time t . If $N_0(t) \geq 1$, i.e., at least one OR_i output is equal to 0, the 4-input NAND ($NAND_1$) output is 1.

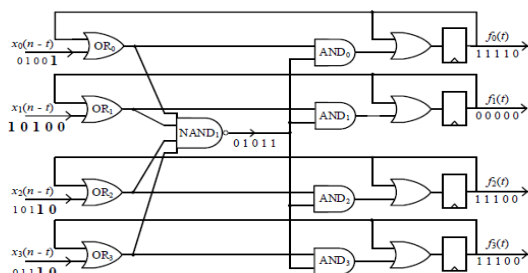


Figure 2: Minimum value generator for Bit serial scheme

3. Decoder Using Proposed Minimum Value Generator

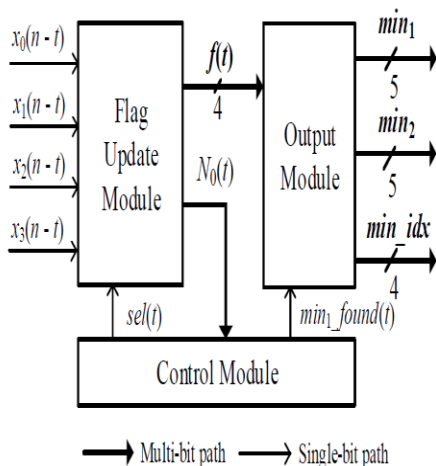


Figure 3: Proposed system

3.1 Control module

The inputs of the proposed generator are bit-serially received from the MSB first. The additional signal $sel(t)$ for which can decide to update the status flags. In order to find min_2 , the control module handles the two signals min_1_found and sel . When the signal min_1_found is initialized to 0, the generator tries to find min_1 . If $t = 1$ and $min_1_found(t) = 0$, the generator can determine min_1 . The signal min_1_found will be 1 at the next cycle $min_1_found(t + 1) = 1$, and will remain this way until the end.

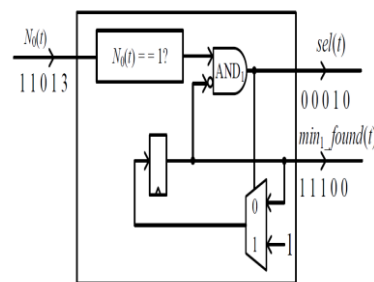


Figure 4: Control module

3.2 Flag update module

The bits on the wires show the bit transitions in time using the four messages of the example. The grayed boxes are the additional circuits compared with the existing generator. To find min_2 , the proposed architecture additionally requires one NOT gate and one 2x1 multiplexer (MUX) for updating each status flag $f_i(t)$. $sel(t)$ selects one of the MUX input. If $N_0(t) = 1$ and $min_1_found(t) = 0$, then $sel(t)$ becomes 1. The generator can determine min_1 and min_1_idx as X_i and i , respectively, where $x_i(n-t) = 0$ and $f_i(t) = 0$. Since the message having min_1 ($X_{min_1_idx}$) is no longer a candidate, the status flag having min_1 ($f_{min_1_idx}$) is changed to 1 by the NOT gate in the shaded box $f_{min_1_idx} + 1$, and the other status flags remain at their previous values.

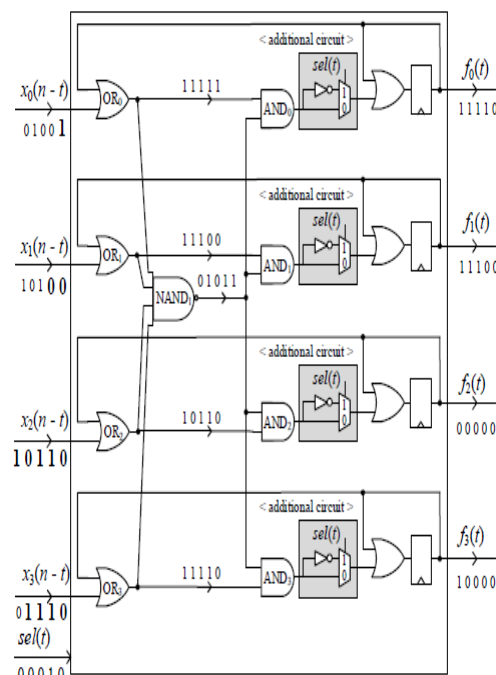


Figure 5: Flag update module

3.3 Output module

The output module produces min_1 and min_2 when t reaches n . When $min_1_found(n) = 1$, the generator has already found min_1 and min_1_idx in the previous cycles. Thus, the generator can determine min_2 as X_i where $f_i(n) = 0$. If $min_1_found(n)$ then the generator fails to find the only minimum value. In other words, there are multiple equal minimum values among the messages. Thus, min_1 and min_2 are the same as X_i where $f_i(n) = 0$. The output

unit is designed for finding a minimum Value of Four Input Values, constructed an output module with Comparator Block. Totally Five comparator block needed for designed an output module. Comparator Minimum is consider as First Minimum and Last Comparator is find the Second Minimum.

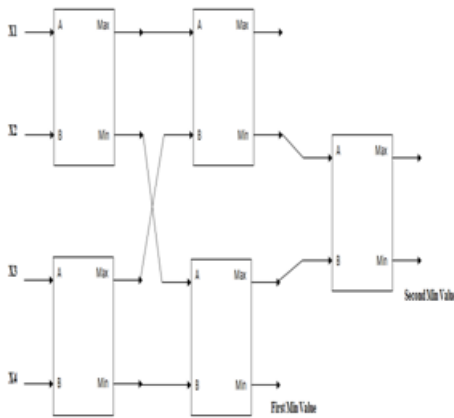


Figure 6: Output module

3.4 Decoder

The encoding of a message is the production of the message. It is a system of coded meanings, and in order to create that, the sender needs to understand how the world is comprehensible to the members of the audience. In the process of encoding, the sender (i.e. encoder) uses verbal (e.g. words, signs, images, video) and non-verbal (e.g. body language, hand gestures, face expressions) symbols for which he or she believes the receiver (that is, the decoder) will understand. The symbols can be words and numbers, images, face expressions, signals and/or actions. It is very important how a message will be encoded; it partially depends on the purpose of the message. The decoding of a message is how an audience member is able to understand, and interpret the message.

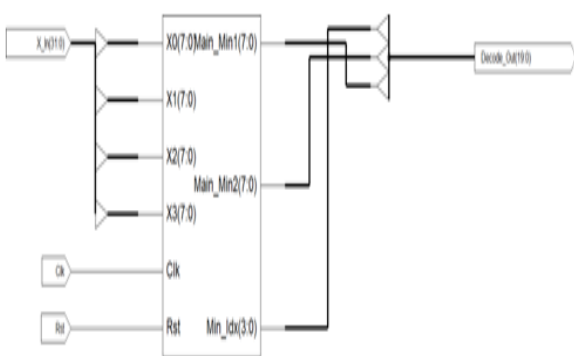


Figure 7: Decoder

4. Implementation Results

The CNU block need to eliminate the error during transmission, the error has been detected and reduce the error with the help of CNU and find the exact two minimum values.

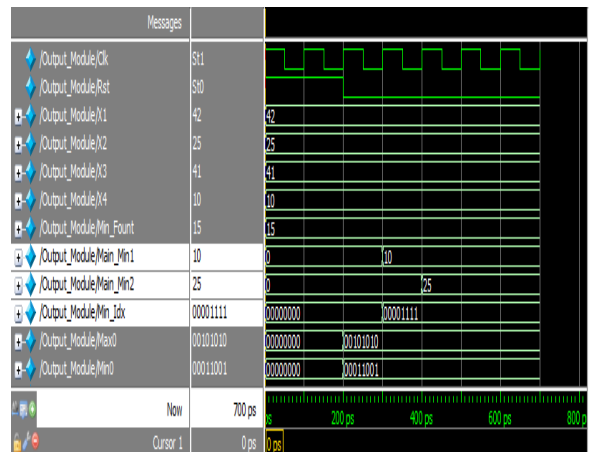


Figure 8: Output module

The flag update module, control module and output module to find the two minimum values and adding a additional decoder unit to design a bit serial LDPC decoder, they use minimum value generator to design a decoder.

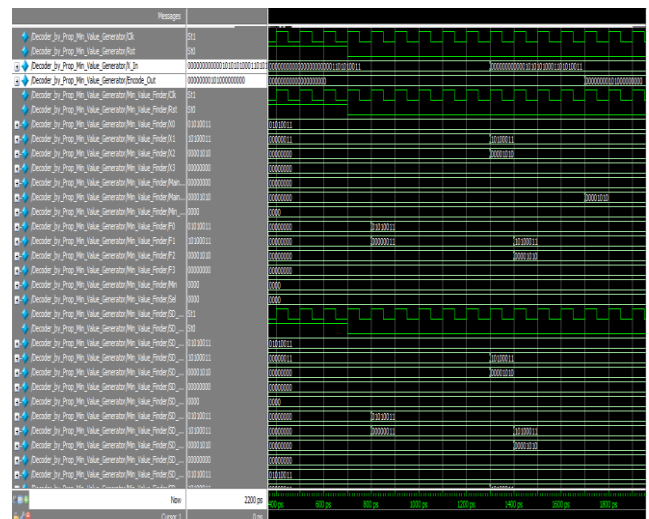


Figure 9: Decoder using proposed minimum value generator

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	32	3,840	1%	
Number of 4 input LUTs	48	3,840	1%	
Logic Distribution				
Number of occupied Slices	32	1,920	1%	
Number of Slices containing only related logic	32	32	100%	
Number of Slices containing unrelated logic	0	32	0%	
Total Number of 4 input LUTs	48	3,840	1%	
Number of bonded IOBs	67	97	69%	
Number of GCLKs	1	8	12%	
Total equivalent gate count for design	547			
Additional JTAG gate count for IOBs	3,216			

Table 1: Summary report

5. Comparison Table II

Method Name	Area			Delay		
	LUT	Slices	Gate	Total	Gate Delay	Path Delay
Existing Design	48	32	547	7.281ns	6.364ns	0.917ns
Proposed Design	48	32	547	7.271ns	6.364ns	0.907ns

Table 2: Comparison for area and delay

6. Conclusion

The first two minimum values has been find to reduce error in Check node unit of Low Density Parity check code. The input can be a 32 bit to design a decoder with the help of minimum value generator, hence its reduce the power and path delay. This system does not suffer from any throughput loss. To improve the Bit Error Rate performance, the proposed generator use two exact minimum values to design a decoder.

References

- [1] R. G. Gallager, Low-Density Parity-Check Codes. Cambridge, MA: MIT Press, 1963
- [2] J. Kim and W. Sung, "Rate-0.96 LDPC decoding VLSI for soft-decision error correction of NAND flash memory, " IEEE Trans. Very Large Scale Integration (VLSI) Systems, vol. 22, no. 5, pp. 1004–1015, May. 2014
- [3] G. Dong, N. Xie, and T. Zhang, "On the use of soft-decision error-correction codes in NAND flash memory," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 58, no. 2, pp. 429–439, Feb. 2011
- [4] IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements Part 3: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, , Sep. 2006, IEEE Std. 802.3an
- [5] A. Morello and V. Mignone, "DVB-S2: The Second Generation Standard for Satellite Broad-band Services," Proc. IEEE, vol. 94, pp. 210–227, 2006.
- [6] S. Lin and D. J. Costello, Error Control Coding: Fundamentals and Applications, 2nd ed. Prentice-Hall, 2004
- [7] A. Darabiha, A. C. Carusone, and F. R. Kschischang, "Power reduction techniques for LDPC decoders, " IEEE J. Solid-State Circuits, vol. 43, no. 8, pp. 1835–1845, Aug. 2008
- [8] F. Cai, X. Zhang, D. Declercq, S. Planjery, and B. Vasic "Finite alphabet iterative decoders for LDPC codes: Optimization, architecture and analysis," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 5, pp. 1366–1375, May 2014
- [9] P. A. Marshall, V. C. Gaudet, and D. G. Elliott, "Deeply pipelined digit-serial LDPC decoding, " IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 59, no. 12, pp. 2934–2944, Dec. 2012
- [10] J. Li, J. Ma, and G. He, "A memory efficient parallel layered QC-LDPC decoder for CMMB systems, "

Compare to the existing design, the proposed bit serial LDPC decoder has reduced the path delay.

Elsevier, Integration, the VLSI journal, vol. 46, no. 4, pp. 359–368, Sep. 2013

- [11] F. Gutierrez, G. Corral-Briones, D. Morero, T. Goette and F. Ramos, "FPGA implementation of the parity check node for min-sum LDPC decoders, " in Proc. Conf. Programmable Logic, Mar. 2012
- [12] C. Wey, M. Shieh, and S. Lin, "Algorithms of finding the first two minimum values and their hardware implementation," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, no. 11, pp. 3430–3437, Dec. 2008
- [13] L. Amaru, M. Martina, and G. Masera, "High speed architectures for finding the first two maximum/minimum values, " IEEE Trans. Very Large Scale Integration (VLSI) Systems, vol. 20, no. 12, pp. 2342–2346, Dec. 2012
- [14] Y. Lee, B. Kim, J. Jung, and I.-C. Park, "Low-Complexity Tree Architecture for Finding the First Two Minima, " IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 62, no. 1, pp. 61-64, Jan. 2015