Design of Area Efficient Low Latency 5G Compliant LDPC Decoder Architecture

by

Abdullah A

2020VLSI-01

A thesis submitted in partial fulfillment of the requirements for the

award of the degree of

Master of Technology

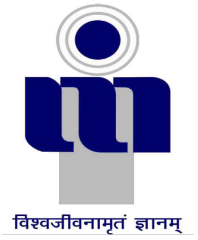
in

VLSI and Embedded Systems

2020-22

Under Supervision of

Prof. G. K. Sharma



ATAL BIHARI VAJPAYEE-

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT

GWALIOR - 474015, MADHYA PRADESH, INDIA

Thesis Certificate

I hereby certify that the work, which is being presented in the report/thesis, entitled **"Design of Area Efficient Low Latency 5G Compliant LDPC Decoder Architecture"**, in the partial fulfillment of the requirement for the award of the degree of **Master of Technology** in **VLSI and Embedded Systems** and submitted to the institution is an authentic record of my own work carried out during the period *July-2021* to *May-2022* under the supervision of **Prof. G. K. Sharma**. I also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.

Signature:

Name: Abdullah A

Roll. No: 2020VLSI-01

Date:

To the best of my/our knowledge, the statements made by Mr. Abdullah A (Roll No. 2020VLSI-01) in the above thesis certificate is correct.

Signature and Name of the Supervisor: Prof. G. K. Sharma

Date:

Candidate's Declaration

I hereby certify that I have properly checked and verified all the items as prescribed in the check-list and ensure that my thesis is in the proper format as specified in the guideline for thesis preparation. I declare that the work containing in this report is my own work.

I understand that plagiarism is defined as any one or combination of the following:

(1) To steal and pass off (the ideas or words of another) as one's own

(2) To use (another's production) without crediting the source

(3) To commit literary theft

(4) To present as new and original idea or product derived from an existing source.

I understand that plagiarism involves an intentional act by the plagiarist of using someone else's work/ideas completely/partially and claiming authorship/originality of the work/ideas. Verbatim copy as well as close resemblance to some else's work constitute plagiarism.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programmes, experiments, results, websites, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report/dissertation/thesis are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable. My faculty supervisor(s) will not be responsible for the same.

Signature:

Name: Abdullah A

Roll. No: 2020VLSI-01

Date:

Acknowledgments

I am highly obliged to **Prof. G. K. Sharma**, and thankful for providing me the sovereignty of functioning and experimenting with ideas. I would like to express my immense gratitude to him not only for his academic guidance but also for his personal interest in our project and constant support coupled with confidence boosting and motivating sessions which proved very fruitful and were instrumental in infusing self-assurance and trust within me.

The nurturing and blossoming of the present work is mainly due to his valuable guidance, suggestions, astute judgment, constructive criticism and an eye for perfection. My mentor always answered myriad of my doubts with smiling graciousness and prodigious patience, never letting me feel that I am novices by always lending an ear to my views, appreciating and improving them and by giving me a free hand in our project. It's only because of his overwhelming interest and helpful attitude, the present work has attained the stage it has. I am also thankful to Dr. Bharat Garg, Assistant Professor in ECE Department at Thapar Institute of Engineering and Technology for helping me whenever I ran into trouble.

Finally, I am grateful to our Institution and colleagues whose constant encouragement served to renew our spirit, refocus our attention and energy and helped me in carrying out this work.

(Abdullah A)

Abstract

Area Efficient and Low Latency Low Density Parity Check (LDPC) decoder is the imperative requirement in the modern wireline and wireless communication systems. Due to the migration of most communication systems on batteries or limited power sources it became important to make the LDPC decoder units embedded on this devices, more energy efficient. The Second Minimum Approximation (SMA) based LDPC decoder prove to best suit the requirement of such applicaions. Further, many applications require low latency Check Node Unit (CNU) architectures to support high throughput applications. Therefore, a novel Area Efficient Low Latency CNU architecture for 5G compliant LDPC decoder is presented. The proposed as well as existing architectures are synthesized on Artix VII series of FPGA Board using Xilinx Vivado 2016.2, to compare the hardware complexities. The synthesis results at FPGA level shows that the proposed architecture exhibits 45.5 % reduction in number of LUTs along with 24.8 % reduction in the data path delay than state of the art architectures. Further, the proposed architecture performance is evaluated using the Bit Error Rate performance analysis graph. It shows that this SMA decoding method gives a reasonable error while comparing it with conventional Min Sum and Offset Min Sum decoding methods.

Keywords: Low-density Parity-Check (LDPC) codes, Check Node Unit (CNU) architecture, 5G new radio (5G-NR), Communication systems.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction

The information bits are encoded & modulated at the sending side, transmitted through the wireline or wireless channel, and demodulated & decoded at the receiving side as shown in Fig. 1.1. The noise may get added to the encoded information while transmitting and it will cause an error in the received signal.

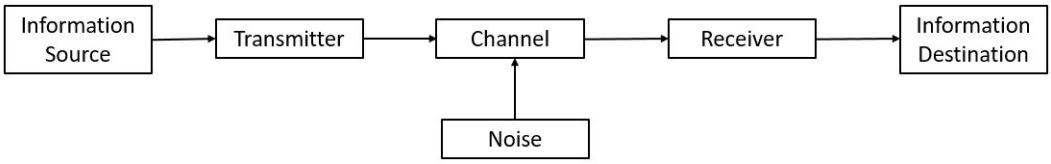


Figure 1.1: Communication System

So, the system has to get rid of the error. For that, a better error correcting code is required to make the system resilient from misinformation. Low density parity check (LDPC) codes are considered to be a better error correcting code for the past several decades. They are introduced by R. Gallager [1] in 1962 which is not so prevalent at that time. Later, they are rediscovered by Mackay [2] in 1999 and reported that they have performance capacity near Shannon limit. The parity check matrix of LDPC codes is extremely sparse and it possess the property of high degree of parallelism. Consequently, they have become one of the most efficient techniques for error correction. They have applications in wireless communication systems, 5G new radio (5G-NR), digital video broadcast (DVB), and space systems.

1.1 Low Density Parity Check Codes

They are represented with the help of $M \times N$ sparse parity check matrix (PCM) H as well as Tanner Graph. The example PCM with 4 rows and 6 columns; consists of 1's and 0's is shown in the Equation (1.1).

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \tag{1.1}$$

The corresponding Tanner Graph representation is shown in Fig. 1.2. The columns which are also known as variable nodes (VN); are denoted by using the variables $C1, C2, \cdots C6$. The rows which are also known as check nodes (CN); are denoted by using the variables $R1, R2, \cdots R4$. The line connecting the VN and CN are termed as edges and these are based on the position of non-zero elements of parity check matrix (H). The number of lines (edges) connected to each node is known as node's degree. The decoding of the LDPC codes are performed by iteratively generating and exchanging messages between the variable nodes and check nodes, via the lines of the Tanner graph. There are different decoding algorithms and schedules to determine the specific operation of each node, the format of the messages and the order in which they are exchanged.

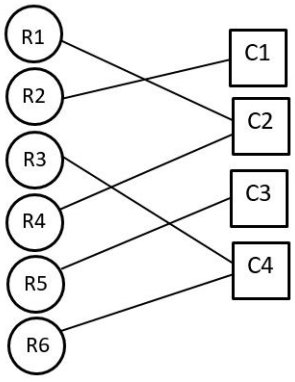


Figure 1.2: Tanner Graph

1.2 Low Density Parity Check decoder

The LDPC decoder block diagram is shown in Fig. 1.3. The four major components are Check Node Unit (CNU), Variable Node Unit (VNU), Finite State Machine (FSM) and Early Termination Module.

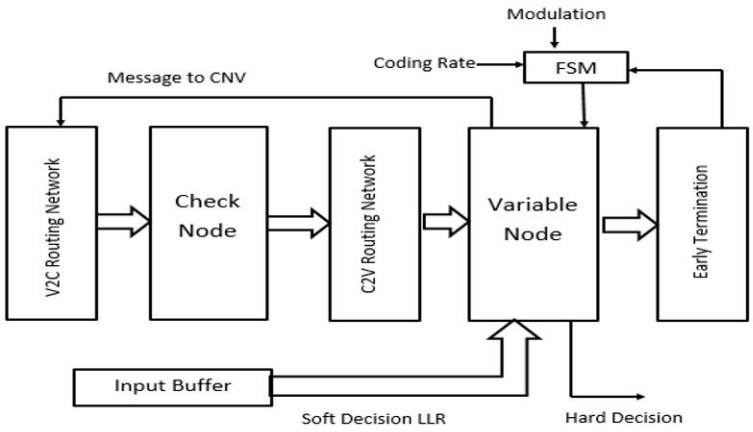


Figure 1.3: LDPC Decoder

The Basic operation of Check Node unit is to find the first and second minimum values from the given inputs and pass it to the Variable Node Unit. The encoded messages are passed back and forth iteratively between CNU and VNU until the code word is found or maximum number of iterations are attained. The early termination submodule is used as an additional power management technique to stop teh decoding process as soon as the code word is correctly decoded. The Finite State machine (FSM) takes code rate, modulation and outputs from early termination blocks as inputs and performs necessary operation. The Decoder working is based on the belief propagation algorithms like Sum Product, Min Sum, Offset Min Sum, Normalized Min Sum, and Reweighted Offset Min Sum algorithm.

1.2.1 Key research areas in LDPC Decoder

The research domains in LDPC decoder can be categorized in following three improvement areas

(1) **Computation Complexity**-The computation complexity of LDPC decoder depends on the minimum value unit used to compute the first and second minimum values. The second minimum computation unit requires more hardware than the first minimum computation unit. Thus, design attempts are made to reduce the hardware resources required for the computation of second minimum value.

(2) **Memory Requirement**- The number of memory locations are required to store the processed data from the check node unit until the new message is arrived. Design attempts are focused to reduce the memory requirement in LDPC decoder implementation using various memory optimization algorithms.

(3) **Flexible Fully Parallel Architecture**- The Standards of the communication systems are improving rapidly, so the implementation of LDPC decoder in flexible physical layers are required. The throughput requirement also increases as the generation of the communication system evolves, so the fully parallel architectures are required. Thus, the communication systems require high throughput and flexible architectures.

1.3 Motivation

LDPC Decoder are widely used in growing communication systems. Performance, Implementation area and latency of the LDPC decoder are mainly considered in its designing. The applications like 5G, the physical layer are likely to be changing rapidly because of the rapid evolution of the communication standards. So. the reprogrammable FPGA implementation is required in that field. The implementation in FPGA requires less area because to reduce the cost of the hardware. It also requires low latency to address the high throughput applications. The lower Bit Error Rate (BER) is required at lower signal to noise ratio (SNR) as well as at lesser number of iterations. Therefore, the LDPC decoder with reduced area, reduced latency with reasonable BER performance is required. Thus, the LDPC decoder with Second Minimum Approximation decoding algorithm is emphasized.

1.4 Thesis Organization

The rest of the chapters of this report are organized as follows: Chapter 2 introduces the existing literature on Minimum Value Computation Unit and Check Node Unit architectures followed by pointing out the gaps in it. Then the set objectives for research is discussed in next subsection based on which research methodology is presented. The proposed design and architecture of Check Node Unit is presented in Chapter 3. Chapter 4 presents the implementation of proposed architecture followed by synthesis results analysis and BER performance analysis graph. Finally, the work is concluded in chapter 5 after which future scope is given.

Chapter 2

Literature review

The sum-product decoding algorithm has high performance metrics and in turn it requires high hardware resources also than any other decoding algorithms. In order to achieve high throughput even with high hardware resource usage a multicore architecture is proposed [4]. But the cost of the system will be high, if a multicore architecture is used. Therefore, to avoid it, the conventional min-sum decoding algorithm with reduced hardware complexity is introduced. These traditional decoding algorithms for LDPC codes generate and exchange messages between check node and variable node only based on loglikelihood ratio (LLR) of bits, and correlations introduced by modulation are not utilized. In [5], symbol based approach is used *i.e.* the correlation factor introduced while modulating the signals are taken into account which results in an alternative low-cost way to better utilize available information. These symbol based sum product and min sum (S-SPA and S-MSA) achieve better error correcting performance than conventional decoding algorithm.

In order to reduce the cost and optimize the throughput, the generated messages are stored using a lower precision by truncating some bits [6]. The truncation happens only after the generated messages are updated by the processing units. The pipeline architectures are faster than the conventional architectures provided the data and control hazards are not there. But, all the pipeline architectures have these two hazards as bottlenecks. The conflicts occurred due to these bottlenecks are addressed in [7] by providing a flooding schedule.

Though the min-sum decoding algorithm has reduced hardware complexity, it also has reduced performance metrics. To overcome this, the offset min-sum [8] - [11] and normalized min-sum

[12] decoding algorithms are introduced. The physical layer of the telecommunication systems are evolving rapidly, so the reprogrammable FPGA based implementation [9] of the 5G LDPC decoders are required. It will reduce the cost and time to implement such physical layers. Further, the length of the cycles is measured to change the number of iterations dynamically is proposed in [13] as reweighted offset min-sum algorithm (ROMS).

2.1 Existing Minimum Value Unit architectures

The LDPC decoder design requires the efficient way to find the first and second minimum values from the given set of inputs. The computation of the first minimum value is easier; but the computation of the second minimum requires a better and efficient hardware architecture. There are two basic methods to find the first two minimums from the given inputs like sorting-based approach and tree-based approach which are discussed in following sections. Basically, the sorting based approach requires less number of comparisons and tree based approach achieves higher speed at lower hardware cost.

2.1.1 Sorting-Based Approach

Consider the following set of numbers in ascending order: $(x_2, x_1, x_7, x_3, x_4, x_6, x_5, x_0)$. From this sequence it is clear that the first minimum value is x_2 and the second minimum value is x_1. The sorting based method works based on the flow shown in Fig. 2.1.

In this figure, the x_{10}, x_{11}, x_{12}, and x_{13} are the minimum of x_0 & x_1, x_2 & x_3, x_4 & x_5, and x_6 & x_7 respectively. The x_{00} and x_{01} are the minimum of x_{10} & x_{11} and x_{12} & x_{13}. The min 1st value is the minimum of x_{00} and x_{01}. The Table 2.1 shows the values of all the variables based on the above set of numbers.

Table 2.1: First Minimum Computation by Sorting method

x_{10}	x_{11}	x_{12}	x_{13}	x_{00}	x_{01}	min 1st
x_1	x_2	x_4	x_7	x_2	x_7	x_2

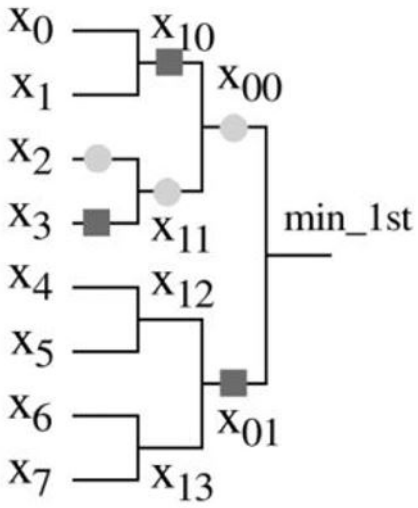


Figure 2.1: Sorting Based First Minimum [3]

The first minimum x_2 occurs at the input, x_{11}, and x_{00} as mentioned in the figure with circles. The numbers at the input x_3, x_{01}, and x_{10} which are denoted in squares in the above figure; are higher than x_2 and they are the possible numbers for the comparison of second minimum. Thus, the second minimum value is computed based on the comparisons as shown in Fig. 2.2. The updated table with first and second minimum values is shown in Table 2.2.

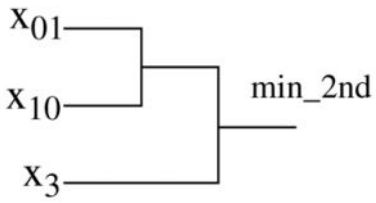


Figure 2.2: Sorting Based Second Minimum [3]

Table 2.2: Second Minimum Computation by Sorting method

x_{10}	x_{11}	x_{12}	x_{13}	x_{00}	x_{01}	min 1st	min 2nd
x_1	x_2	x_4	x_7	x_2	x_7	x_2	x_1

The Hardware implementation of the sorting based method requires a minimum value unit (MVU) as shown in Fig. 2.3. The Comparison operation is based on the following Equations 2.1

and 2.2.

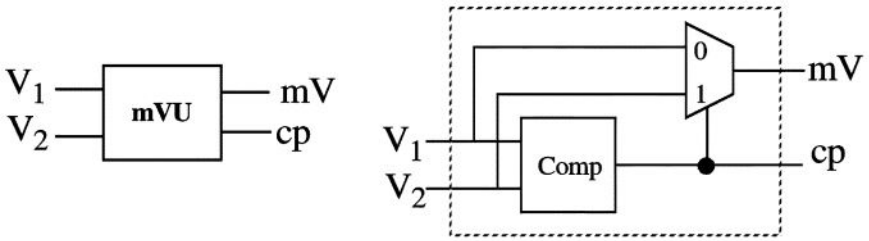


Figure 2.3: Minimum value Unit [3]

$$cp = \begin{cases} 0, & \text{if } x_0 < x_1. \\ 1, & \text{otherwise.} \end{cases} \tag{2.1}$$

$$mV = \begin{cases} V_1, & \text{if } Index = 0. \\ V_2, & \text{otherwise.} \end{cases} \tag{2.2}$$

The first minimum computation architecture of the 8-inputs sorting based approach is shown in Fig. 2.4. The architecture of the 2-inputs minimum value unit is expanded to obtain the 8-inputs minimum value unit. It takes 8 inputs and computes the minimum value and index value at each stage as shown in the figure. The index value also found for all stages because that value is used in finding the second minimum value.

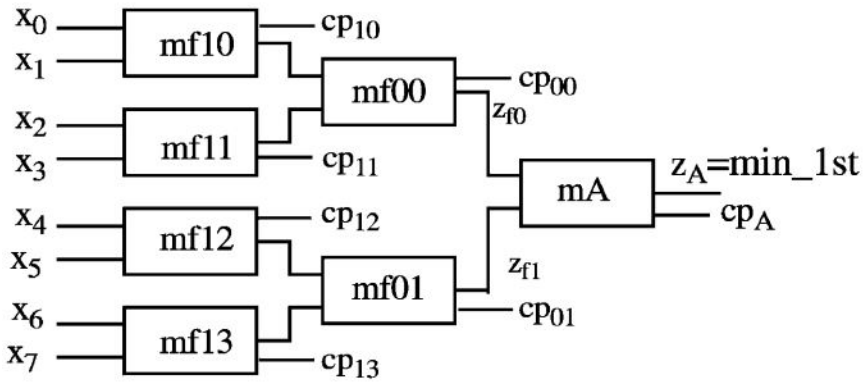


Figure 2.4: Architecture of First Minimum Computation in Sorting method [3]

The computation of the second minimum value requires an extra architecture as shown in Fig. 2.5 to find the index of second minimum. It computes the output value based on the following Equations 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 2.10, and 2.11.

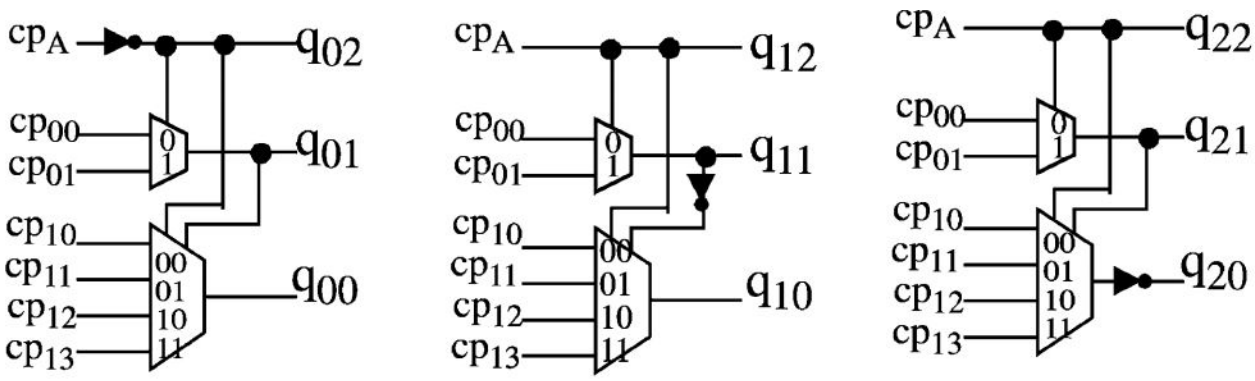


Figure 2.5: Architecture of Second Minimum Index Computation in Sorting method [3]

$$q_{02} = cp_A \tag{2.3}$$

$$q_{01} = \begin{cases} cp_{00}, & \text{if } q_{02} = 0. \\ cp_{01}, & \text{if } q_{02} = 1. \end{cases} \tag{2.4}$$

$$q_{00} = \begin{cases} cp_{10}, & \text{if } q_{02} = 0 \ \&\& \ q_{01} = 0 \ . \\ cp_{11}, & \text{if } q_{02} = 0 \ \&\& \ q_{01} = 1. \\ cp_{12}, & \text{if } q_{02} = 1 \ \&\& \ q_{01} = 0. \\ cp_{13}, & \text{if } q_{02} = 1 \ \&\& \ q_{01} = 1. \end{cases} \tag{2.5}$$

$$q_{12} = cp_A \tag{2.6}$$

$$q_{11} = \begin{cases} cp_{00}, & \text{if } q_{12} = 0. \\ cp_{01}, & \text{if } q_{12} = 1. \end{cases} \tag{2.7}$$

$$q_{10} = \begin{cases} cp_{10}, & \text{if } q_{12} = 0 \ \&\& \ q_{11} = 0 \ . \\ cp_{11}, & \text{if } q_{12} = 0 \ \&\& \ q_{11} = 1. \\ cp_{12}, & \text{if } q_{12} = 1 \ \&\& \ q_{11} = 0. \\ cp_{13}, & \text{if } q_{12} = 1 \ \&\& \ q_{11} = 1. \end{cases} \tag{2.8}$$

$$q_{22} = cp_A \tag{2.9}$$

$$q_{21} = \begin{cases} cp_{00}, & \text{if } q_{22} = 0. \\ cp_{01}, & \text{if } q_{22} = 1. \end{cases} \tag{2.10}$$

$$q_{10} = \begin{cases} cp_{10}, & \text{if } q_{22} = 0 \ \&\& \ q_{21} = 0 \ . \\ cp_{11}, & \text{if } q_{22} = 0 \ \&\& \ q_{21} = 1. \\ cp_{12}, & \text{if } q_{22} = 1 \ \&\& \ q_{21} = 0. \\ cp_{13}, & \text{if } q_{22} = 1 \ \&\& \ q_{21} = 1. \end{cases} \tag{2.11}$$

Based on these index value, the second minimum value is computed using the architecture shown in Fig. 2.6.

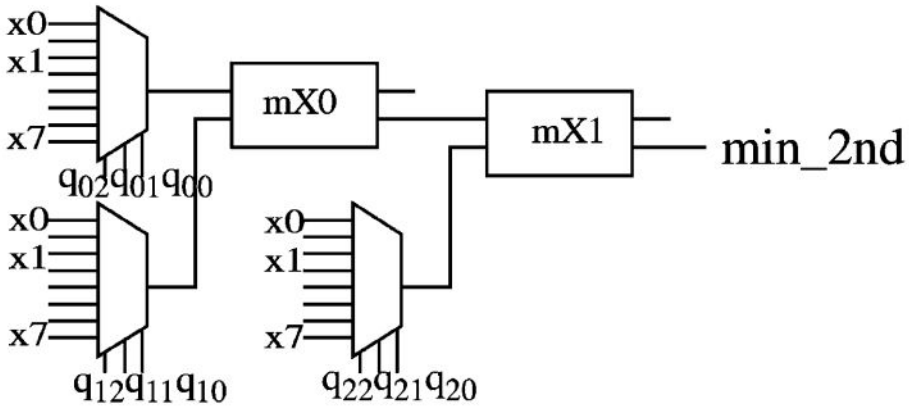


Figure 2.6: Architecture of Second Minimum Computation in Sorting method [3]

2.1.2 Tree-Based Approach

The Hardware implementation of the Tree Based method requires the 2 input basic building block as shown in Fig. 2.7. It consists of 2 input minimum value unit as explained in the above method and an extra 2:1 mux for computing the second minimum. The block gives the first minimum, second minimum and index as the outputs. The second minimum value is computed based on the Equation

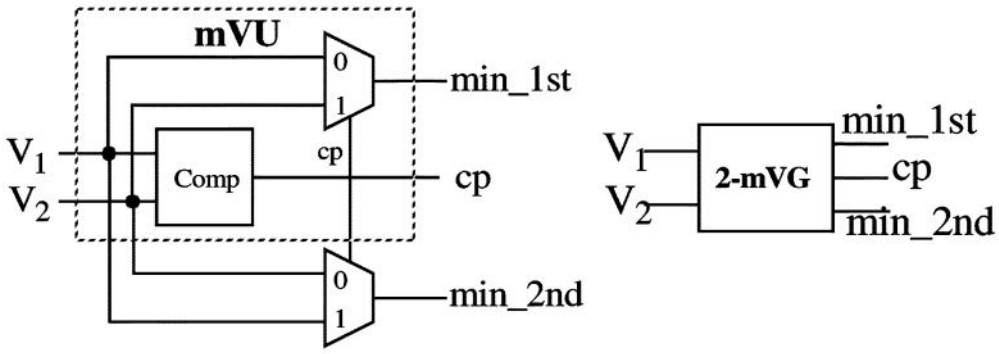


Figure 2.7: Architecture of 2 inputs tree based approach [3]

$$min\ \ 2 = \begin{cases} V_2, & \text{if } Index = 0. \\ V_1, & \text{otherwise.} \end{cases} \tag{2.12}$$

The Hardware implementation of the 4 input tree based approach is shown in Fig. 2.8. It consists of two 2 input tree based architectures and a connection unit. The first 2 input unit computes the minimum value from the first two inputs and the second one computes the minimum value from the next two inputs. The first minimum of both the previous 2 input units are compared again using 2 input sorting based minimum value unit and the first minimum value is computed finally. The second minimum also computed with the help of two 2 input sorting based minimum value unit and 2:1 mux as shown in the figure. The index of the first minimum also computed from the indices of previous units; with the help of 2:1 mux.

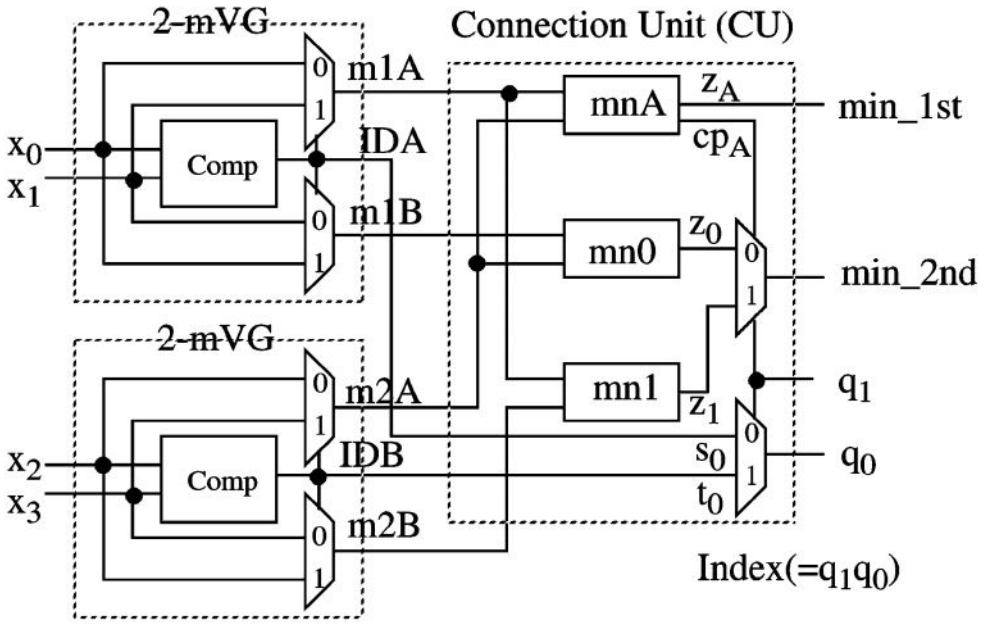


Figure 2.8: Architecture of 4 input tree based approach [3]

2.2 Existing Check Node Unit architectures

2.2.1 Minimum Value Generator Approach

Significant design efforts have been given to reduce the area and to achieve better performance by combining both the check node unit and variable node unit into a single unit as Combined Check-Node & Variable-Node processing-units (CVCPU) as shown in Fig. 2.9. Here, the Check Node Unit (CNU) architecture is denoted as Min Sum Approximation Unit (MSAU). The basic building block of MSAU is minimum Value Generator (mVG-19) which computes the first minimum m1 and second minimum m2 from the given 19 inputs. First, the inputs are converted from two's complement to sign magnitude form with the help of TCSM submodule. The m1 and m2 are subtracted by the binary offset value '0001' and checked whether the subtracted value is negative or not. If the subtracted value is negative, then the output is zero otherwise the output is actual subtracted value. Now, the processed m1 is compared with all the inputs to check whether it is equal or not. The m1 value is the output of all the ports of MSAU except the port where the m1

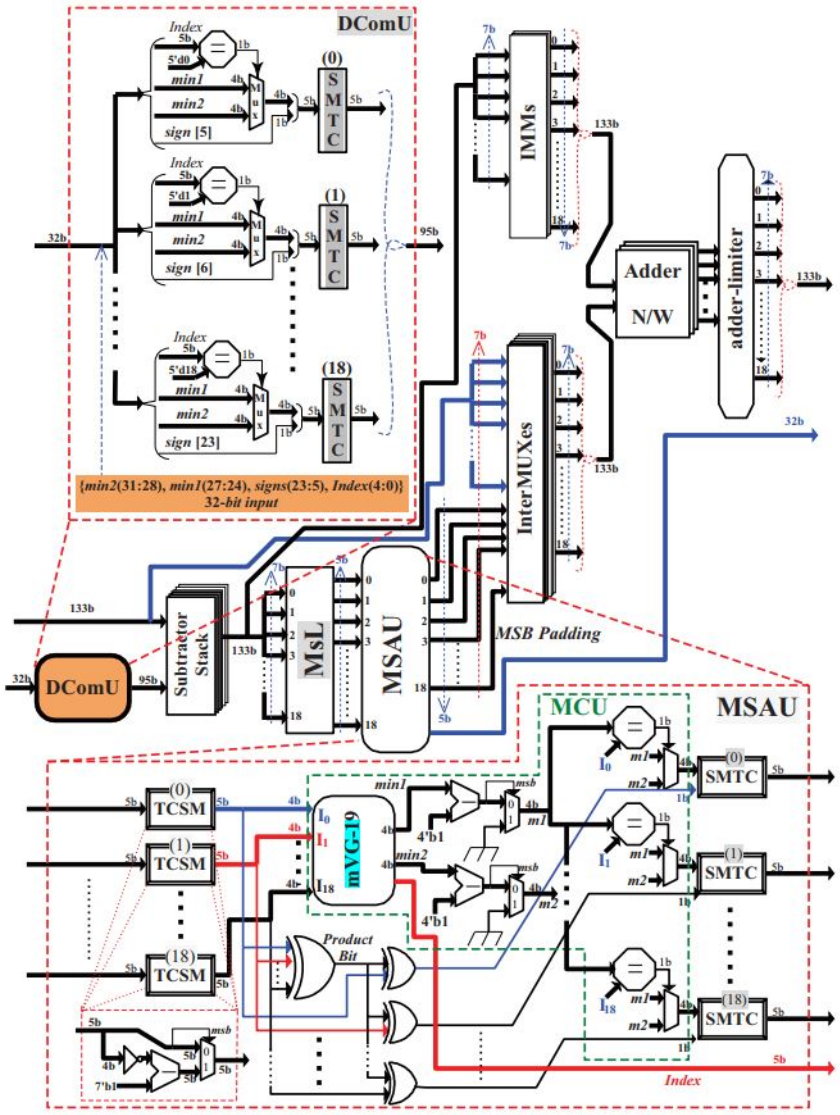


Figure 2.9: Architecture of combined check-node & variable-node processing-units [8]

value and the input are same; the output of that port is m2. This computation is achieved by using equalizers and 2:1 mux. In parallel, the product bit is generated by EXORing the msb of all the inputs and each msb is EXORed with the product bit to get the sign processed outputs. Finally, these 19 updated magnitudes and updated signs are processed through SMTC submodule that converts sign magnitude form to two's complement form.

The similar MSAU architecture with slight changes is mentioned in Fig. 2.10. The number of bits for the input is 8 bits and the offset value subtracted is '7'b2' while in previous architecture

the number of bits for inputs is 5 bits and the offset value subtracted is '4'b1'.

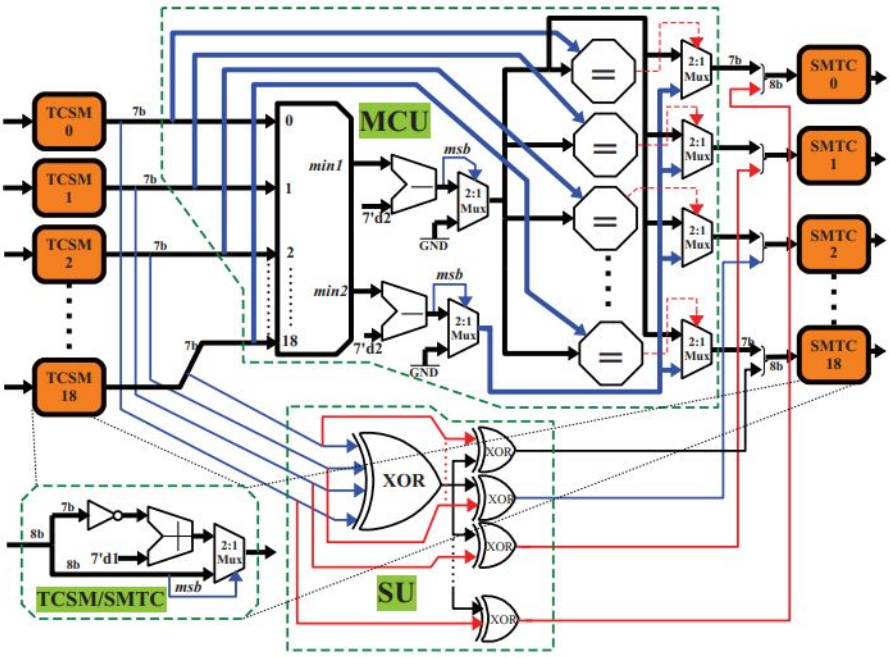


Figure 2.10: Min-Sum Approximation Unit [11]

2.2.2 Conversion Unit Approach

The t_b number of inputs are equally splitted into z sets of input and processed using z number of Min Sum Units (MSU) as shown in Fig. 2.11.

The conventional way of computing the two's complement from sign magnitude is by complementing the magnitude of the given number and adding one to it. It requires one adder and an inverter which uses 21 gate equivalents. Here, the computation of two's complement to sign magnitude unit uses a logically optimized conversion unit and 2:1 mux. The conversion of the magnitude is based on the boolean expression given in Eq. 3.3, Eq. 3.4, Eq. 3.5 and Eq. 3.6 respectively. The expression based approach uses only 16 gate equivalents. Thus, 5 gate equivalents are reduced.

$$f_3 = \bar{g_3}|(\bar{g_2}.\bar{g_1}.\bar{g_0}) \tag{2.13}$$

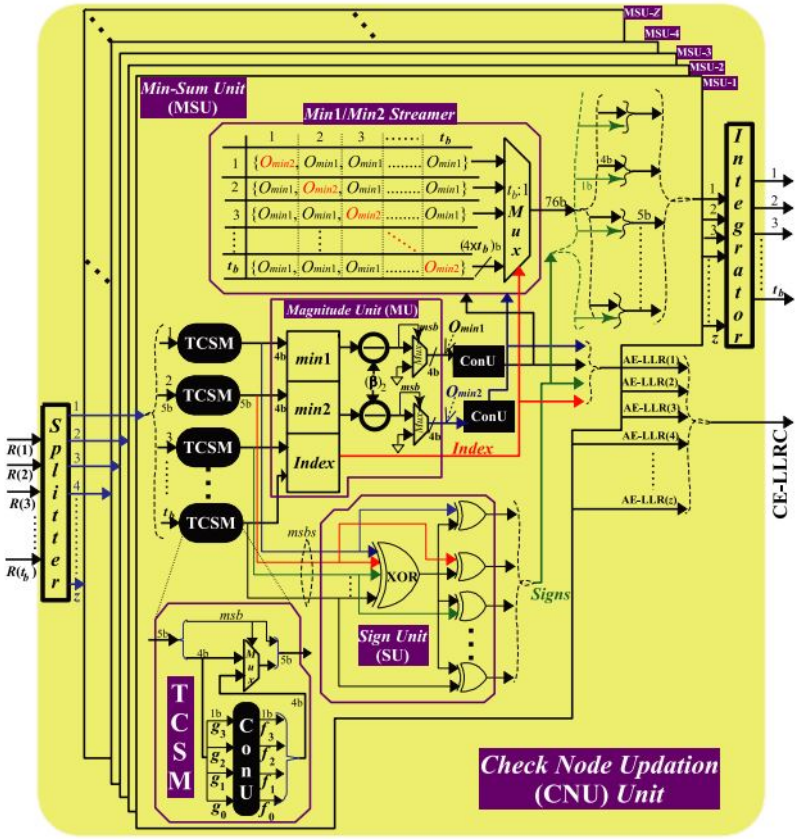


Figure 2.11: Architecture of Check Node Unit with Conversion unit [10]

$$f_2 = (\bar{g_3}.\bar{g_2})|(\bar{g_2}.g_1)|\bar{g_2}.\bar{g_0})|(g_2.\bar{g_1}.\bar{g_0}) \tag{2.14}$$

$$f_1 = g_1 \oplus g_0 \tag{2.15}$$

$$f_0 = g_0 \tag{2.16}$$

In the minimum value generator approach, the output ports are all first minimum except the port where the first minimum and the input are same; the output of that port is second minimum. It is done with the help of equalizers and 2:1 mux. Here, the same is achieved efficiently with the help of Min1/Min2 streamer.

2.3 Research gaps

According to literature review done following research gaps can be concluded

- The existing Check Node Unit architectures use an extra hardware for the computation of the second minimum value.
- The overall hardware complexity and data path delay have increased because of this extra hardware resources.
- The LDPC decoder architecture with conventional Min-Sum decoding algorithm has poor Bit Error Rate performance.
- Most of the existing designs become inefficient as the number of iterations are decreased.

2.4 Research Objectives

- To design a Check Node Unit (CNU) with Conventional Min-Sum Decoding Algorithm.
- To design other existing CNU Architectures in the literature in order to compare.
- To design a New CNU Architecture based on the comparison. This Architecture must have both Algorithmic and Architectural improvements than the existing Architectures.
- To study the behaviour of the Complete LDPC Decoder architecture with the help of BER performance analysis graph.

2.5 Research Methodology

- Thorough review of all existing design methods and Check Node Unit architectures of LDPC decoders.
- Formulation of research problem based of the research gaps.
- Proposal of design satisfying the set constraints.
- Comparative analysis of hardware complexities and performance parameters of proposed design with existing.
- Necessary modification of design based on the analysis and meet the final research objectives.

Chapter 3

Proposed Design

The architecture of the proposed Check Node Unit consists of three major building blocks such as minimum value computation unit (MVCU), sign computation unit (SU), and two's complement to sign magnitude/sign magnitude to two's complement unit (TCSM/SMTC) which are detailed in the following sections.

3.1 Minimum Value Computation Unit

The basic building block of MVCU is 20-input minimum value unit (MVU-20). It is designed using 2-input minimum value unit (MVU) as shown in Fig. 3.1. It compares two 5-bit inputs and gives the minimum value and index as outputs.

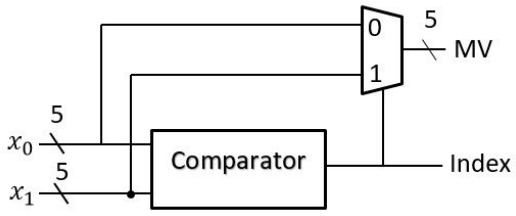


Figure 3.1: Architecture of 2-input minimum value unit

The index value is computed by comparing two inputs using comparator and MV is computed based on the index value with the help of 2:1 multiplexer as shown in 2.10 and 2.11 respectively.

$$Index = \begin{cases} 0, & \text{if } x_0 < x_1. \\ 1, & \text{otherwise.} \end{cases} \tag{3.1}$$

$$mv = \begin{cases} x_0, & \text{if } Index = 0. \\ x_1, & \text{otherwise.} \end{cases} \tag{3.2}$$

The 4-input minimum value unit is shown in Fig. 3.2. It consists of three 2-input MVUs and an adder. The two 2-input MVUs which are named as A and B are used to compute the minimum values from the given inputs and the third MVU is used to compute the minimum of mv_A and mv_B. The output of the third MVU is considered as first minimum min_1. The second minimum value min_2 is computed by adding a small value alpha (α) to the first minimum.

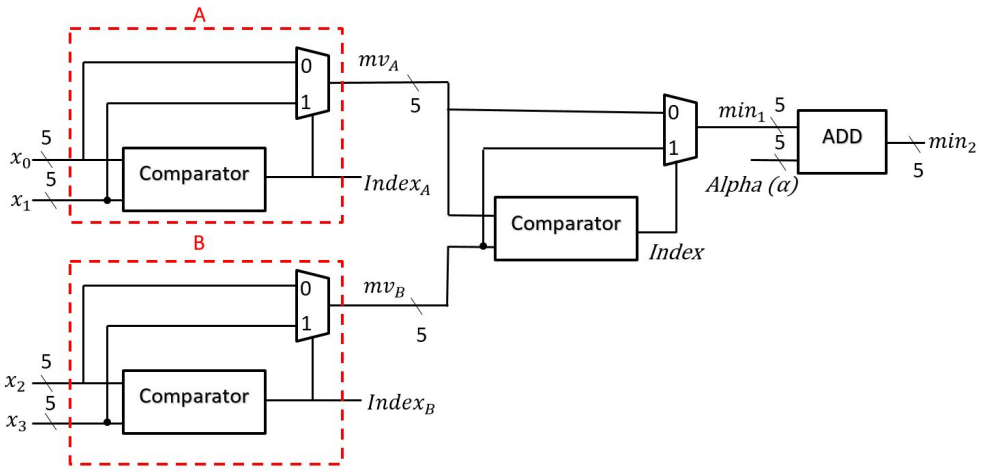


Figure 3.2: Architecture of 4-input minimum value unit

Similarly, the 8-input MVU is constructed by using seven 2-input MVUs and an adder. The 16-input MVU is constructed by using fifteen 2-input MVUs and an adder. In general, the K-input MVU requires (K-1) number of 2-input MVUs and an adder.

Since our design requires 20-input MVU (MVU-20), it can be constructed by nineteen 2-input MVUs and an adder as shown in Fig. 3.3. The ten 2-input MVUs are considered as stage A where it takes primary inputs and computes the minimum value. The remaining nine 2-input MVUs

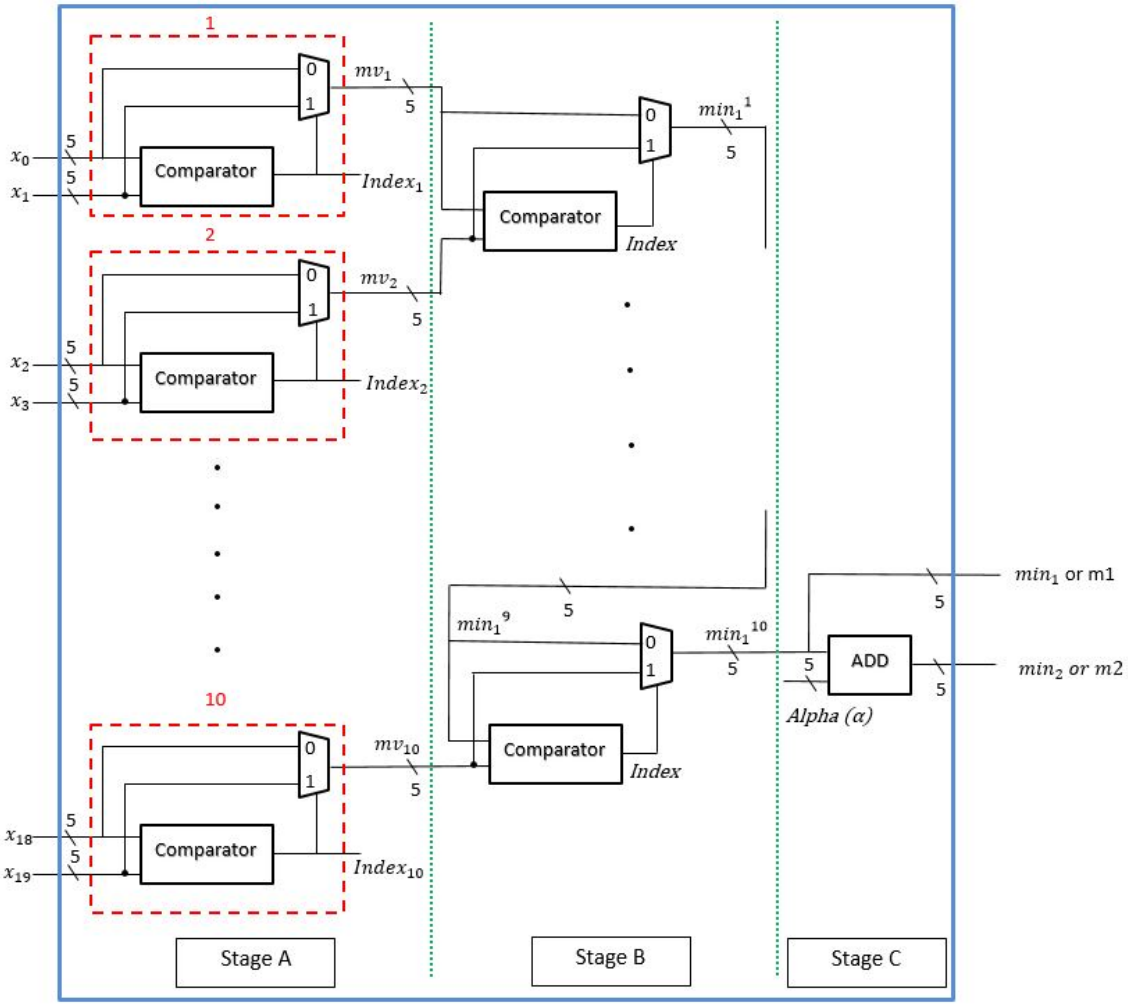


Figure 3.3: Architecture of 20-input minimum value unit

are considered as Stage B where it takes the intermediate minimum values and computes the first minimum value min_1. The 5-bit adder is considered as stage C where it adds Alpha (α) to first minimum value to compute second minimum value min_2.

Now, the MVU-20 is connected in a manner to construct the MVCU as shown in Fig. 3.4. It computes first minimum min_1 and second minimum min_2 using the MVU-20 module. It subtracts the offset value $4'b1$ from min_1 and min_2. Then the most significant bit (MSB) is checked to find whether the subtracted value is positive or negative. If the subtracted value is negative, then the output of 2:1 Mux is zero otherwise it is the actual subtracted value. Now, all the output of the MVCU is replaced with min_1 expect the index of min_1; this magnitude of min_1 is replaced by

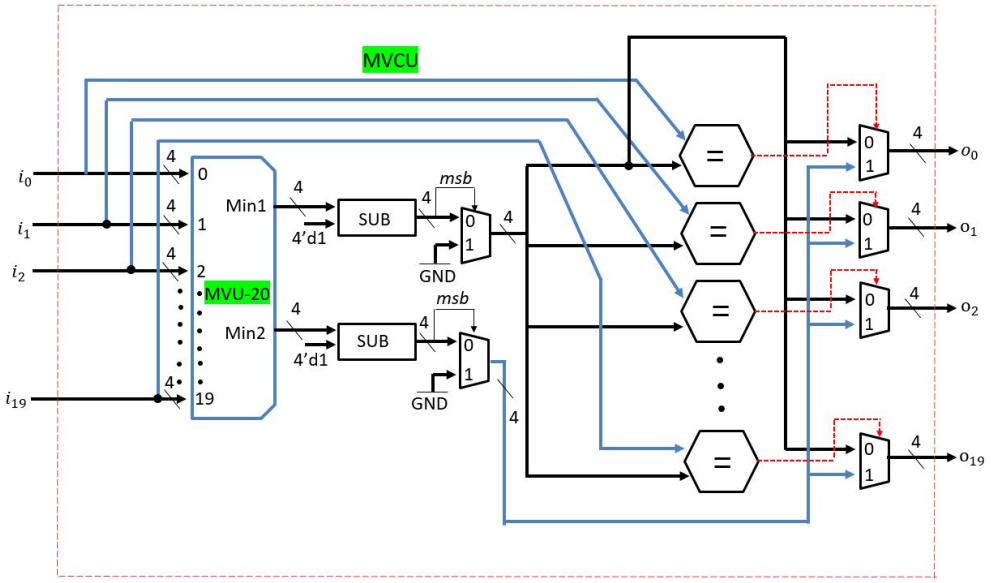


Figure 3.4: Minimum value computation unit architecture

min_2. This is done by using 20 equalizers and 2:1 multiplexers.

3.2 Sign Unit

The sign unit takes all the MSB of the given inputs and EX-OR those bits to compute a product bit. Then, the product bit is EX-ORed with each MSB to obtain the sign processed outputs as shown in Fig. 3.5. These sign processed outputs are concatenated with the outputs of MVCU in the MSB position.

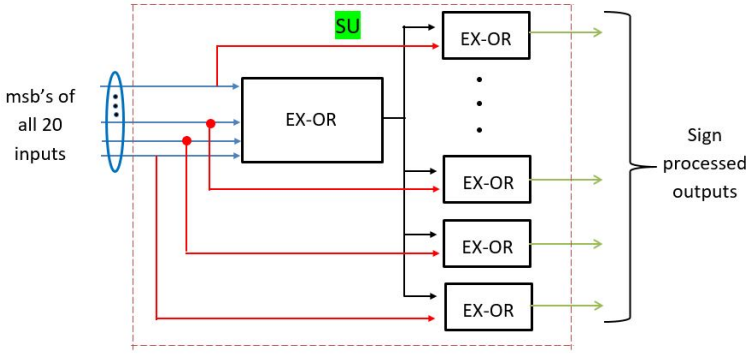


Figure 3.5: Sign computation unit architecture

3.3 TCSM/SMTC

TCSM/SMTC module is used at both the sides of CNU architecture. It takes 5 bits as input and processes the lower 4 bits with the help of a conversion unit (ConU). The simplified Boolean expressions for ConU are given in Eq. 3.3, Eq. 3.4, Eq. 3.5 and Eq. 3.6 respectively.

$$f_3 = \bar{g_3} | (\bar{g_2}.\bar{g_1}.\bar{g_0}) \tag{3.3}$$

$$f_2 = (\bar{g_3}.\bar{g_2}) | (\bar{g_2}.g_1) | \bar{g_2}.\bar{g_0}) | (g_2.\bar{g_1}.\bar{g_0}) \tag{3.4}$$

$$f_1 = g_1 \oplus g_0 \tag{3.5}$$

$$f_0 = g_0 \tag{3.6}$$

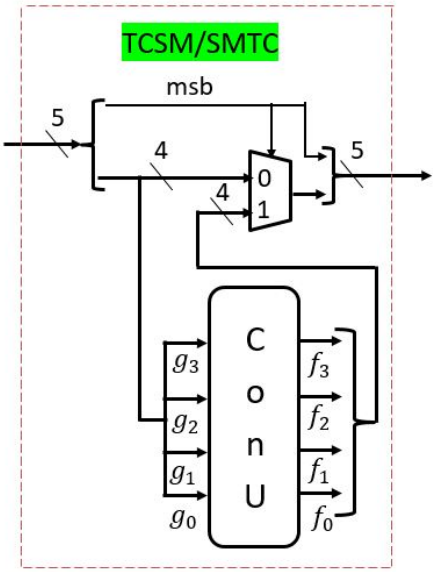


Figure 3.6: Architecture of magnitude conversion unit

The output of this module is either actual value or converted value based on the value of MSB as shown in Fig. 3.6.

3.4 Complete CNU Architecture

All the major modules are interconnected as shown in Fig. 3.7 to achieve a novel CNU architecture.

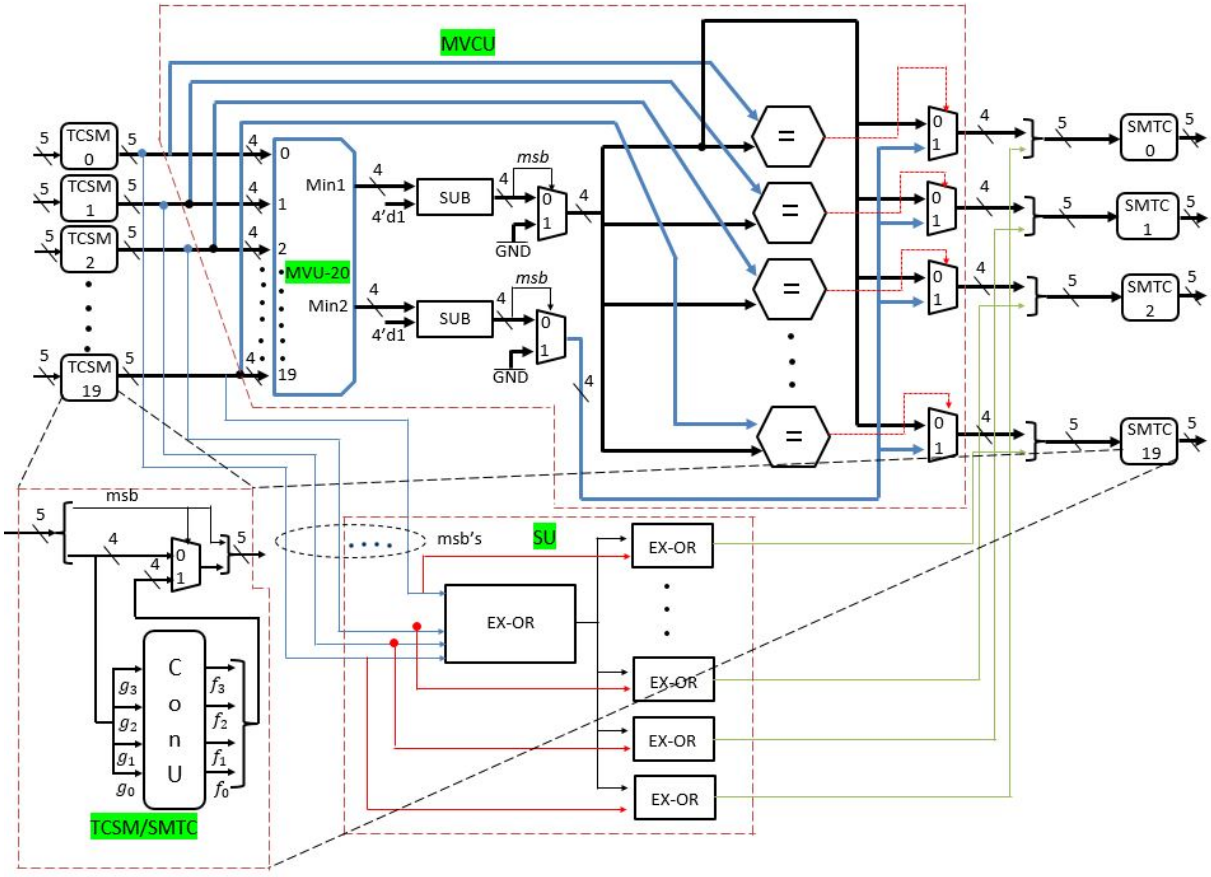


Figure 3.7: Proposed novel check node unit architecture

Chapter 4

Implementation and Results Analysis

The efficacy of the proposed Check Node Unit architecture is evaluated and compared with the existing architectures. The BER Performance analysis of the complete LDPC Decoder is also obtained. They are discussed in the following sections.

4.1 Hardware Synthesis Results

The existing architectures are in different number of word length, different offset value, and synthesized in different FPGA board. For a fair comparison, all the existing CNU architectures are converted to the common evaluation method i.e. the number of bits are 5, offset value is binary '0001', and FPGA board used is Artix VII series.

Initially, the exisitng CNU architectures with conventional Min Sum are designed and synthesized. Now, the proposed CNU architecture based on second minimum approximation (SMA) decoding algorithm is written in Verilog Hardware-Description-Language and synthesized. The comparison synthesis results of the proposed design with the existing architectures are shown in Table 4.1.

Table 4.1: Hardware Utilization Comparison Report

Metrics	[8] & [11]	[10]	Proposed Work
No. of word length (bits)	5	5	5
Core Area (No. of LUTs used)	493	345	269
Data Path Delay (ns)	20.12	18.17	15.13

The [8] & [11] are two different existing works with different CNU architectures. But the difference between those two are only in the number of bits and offset value used i.e. [8] uses 5 number of bits and '4'b1' offset value and [11] uses 8 number of bits and '7'b2' offset value. The difference in the Core Area and Data Path Delay between [8] & [11] and [10] are because the former uses the conventional way to compute the two's complement to sign magnitude and the latter uses the conversion unit approach.

From the above table, it is clear that our proposed design has 45.5% and 22.02% reduction in number of LUTs than [8] & [11] and [10] respectively. It also has 24.8% and 16.7% reduction in data path delay than [8] & [11] and [10] respectively.

4.2 BER Performance Analysis

The BER performance analysis graph is used to evaluate the Bit Error Rate (BER) versus Signal to Noise (SNR) of the LDPC decoder. Basically, it gives the error incurred while approximating the computation of second minimum value. The lesser the value, the better the system. This study is carried out in MATLAB 2022a version. The complete decoder operation is simulated using the m-file. The corresponding MATLAB commands are used to achieve this graph.

The LDPC decoder behaviour is studied for two conventional decoding algorithms *i.e.* min-sum (MS) & offset min-sum (OMS) and one proposed decoding algorithm *i.e.* second minimum approximation (SMA). First, the LDPC codes corresponding to 5G New Radio (5G-NR) standards are encoded. The binary phase-shift keying (BPSK) modulation is performed in the encoded information bits i.e. The negative encoded signal bits are converted to 0 and the positive encoded signal bits are converted to 1 at the sending end.

The basic noise model used to mimic the disturbances occuring in the nature is known as Additive White Gaussian Noise (AWGN). It has uniform power density over all frequency range and it follows normal distribution in time domain. The modulated signals are transmitted through AWGN channel.

Then, it is BPSK demodulated i.e. the noisy received binary signals are converted to random

positive and negative demodulated signal bits.Then, it is decoded at the receiving end.

The BER performance analysis graph is shown in Fig. 4.1. The following graph is plotted for all the above mentioned decoding algorithms for 10 iterations and 20 iterations for the SNR (E_b/N_0) value ranging from 0 to 3 dB.

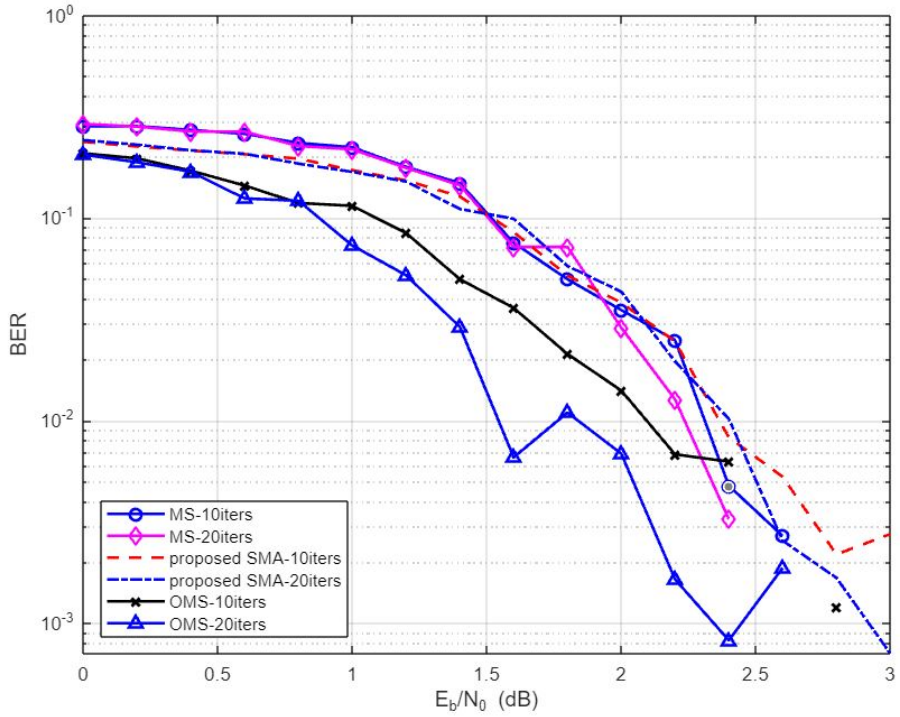


Figure 4.1: BER Performance Analysis of LDPC Decoder

There are 6 graphs in the above figure i.e. Conventional Min Sum decoding Algorithm with 10 iterations (MS-10iters) & 20 iterations (MS-20iters), Offset Min Sum decoding Algorithm with 10 iterations (OMS-10iters) & 20 iterations (OMS-20iters), and Proposed Second Minimum Approximation decoding Algorithm with 10 iterations (Proposed SMA - 10iters) & 20 iterations (Proposed SMA - 20iters).

The above graph is interpreted from the left side at its starting point. It is clear that the starting point of conventional Min Sum is at the higher position followed by Proposed Second Minimum Approximation and Offset Min Sum decoding Algorithm. It implies that the BER is higher for conventional Min Sum than Proposed Second Minimum Approximation; than Offset Min

Sum decoding Algorithm. Our Proposed Second Minimum Approximation decoding Algorithm has reasonable BER performance. Finally, it is clear from the graph that; as the number of iterations increase, the BER decreases.

Chapter 5

Discussions and Conclusion

An Area Efficient Low Latency 5G Compliant LDPC Decoder architecture is proposed. The Area Efficiency and Low Latency are achieved by using the proposed Second Minimum Approximation decoding algorithm.

The designs are synthesized at Artix VII FPGA Board using Xilinx Vivado 2016.2. From the FPGA results, it is clearly observable that the proposed design has 45.5% and 22.02% reduction in number of LUTs than [8] & [11] and [10] respectively. It also has 24.8% and 16.7% reduction in data path delay than [8] & [11] and [10] respectively.

5.1 Contributions

- For a fair comparison, the synthesis of the existing Check Node Unit architectures is performed in the same FPGA board i.e. Artix VII.
- Based on the comparison results, the Area Efficient Low Latency Check Node Unit architecture for the 5G Compliant LDPC decoder is proposed.
- The proposed architecture is synthesized on FPGA to evaluate hardware effectiveness.
- The synthesis results at FPGA level show that the proposed architecture exhibits 45.5% and 22.02% reduction in number of LUTs than [8] & [11] and [10] respectively. It also has 24.8% and 16.7% reduction in data path delay than [8] & [11] and [10] respectively.

5.2 Limitations

The Area efficiency in the proposed design is obtained by eliminating the conventional hardware for computing the second minimum value and computing it by using the approximation. The approximation method utilizes less implementation area and data path delay but the BER performance analysis graph shows that the BER value is slightly higher for the proposed algorithm than the offset min sum decoding algorithm. The increase in the BER value is because of using an adder for computing second minimum value. If the actual second minimum value computed by conventional hardware approach is higher than the approximate approach, then the BER value also increases. So, the area and data path delay are reduced due to this approach but slight increment in BER value shows that the architecture may become inefficient when the actual second minimum value is higher than approximate value.

5.3 Future scope

The proposed design exhibits that the second minimum approximate Check Node Unit architecture is more area efficient and lower latency than the conventional Check Node Unit architectures. It is also shown that the proposed SMA decoding algorithm performs better than conventional min sum decoding algorithm. But, the generation of communication systems are evolving rapidly. So, all the implementation of these architectures should be done in reconfigurable processors. Thus, the LDPC decoder with SMA decoding algorithm implemented in the reconfigurable FPGA can be seen as the future scope of this proposed work.

Bibliography

[1] R. Gallager, "Low-density parity-check codes," in *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[2] D. .J. C. Mackay and R. W. Neal, "Good Codes based on very sparse matrices," in *Proc. 5th IMA Conf. Cryptogr. Coding*, vol. 1025, pp. 100–111, Dec. 1995.

[3] Chin-Long Wey, Ming-Der Shieh, and Shin-yo Lin, "Algorithms of Finding the First Two Minimum Values and Their Hardware Implementation," in *IEEE Transactions on Circuits and Systems—I: Regular Papers*, vol. 55, no. 11, pp. 3430–3437, Dec. 2008.

[4] M. Li, V. Derudder, K. Bertrand, C. Desset, and A. Bourdoux, "High-Speed LDPC Decoders Towards 1 Tb/s," in *Transactions on Circuits and Systems I*, vol. 68, no. 5, pp. 2224–2233, Mar. 2021.

[5] Jingbo Liu, Jingyang Yuan, and Jin Sha, "Symbol-Based Algorithm for Decoding Binary LDPC Codes with Higher Order Modulations," in *Int. Symp. on circuits and Systems*, May 2020.

[6] T. Truong Nguyen-Ly, V. Savin, K. Le, D. declercq, F. Ghaffari, and O. Bonacalo, "Analysis and Design of Cost-Effective, High-Throughput LDPC Decoders," in *Transactions on VLSI Systems*, Vol. 26, No. 3, pp. 508–521, Mar. 2018.

[7] L. Petrovic, M. Markovic, M. El Mezeni, V. Saranovac, and A. radosevic, "Flexible High Throughput QC-LDPC Decoder with Perfect Pipeline Conflicts Resolution and Efficient Hardware Utilization," in *Transactions on Circuits and Systems I*, Vol. 67, No. 12, pp. 5454–5467, Dec. 2020.

[8] Anuj Verma, and Rahul Shrestha, "A New VLSI Architecture of Next-Generation QC-LDPC decoder for 5G New-Radio Wireless-Communication Standard," in *Int. Symp. on circuits and Systems*, Oct. 2020.

[9] J. Nadal, and A. Baghdadi, "Parallel and Flexible 5G LDPC Decoder Architecture Targeting FPGA," in *Transactions on VLSI Systems*, Vol. 29, No. 6, pp. 1141–1151, Jun. 2021.

[10] Anuj Verma, and Rahul Shrestha, "Hardware Efficient and High Throughput LLRC Segregation based Binary QC-LDPC Decoding Algorithm and Architecture," in *IEEE Transactions on Circuits and Systems II: Express briefs*, vol. 68, no. 8, pp. 2835–2839, Aug. 2021.

[11] Anuj Verma, and Rahul Shrestha, "A New Partially-Parallel VLSI-Architecture of Quasi-Cyclic LDPC Decoder for 5G New-Radio," in *Int. Conference on VLSI Design*, Jan. 2020.

[12] C. Lin, L. Liu, Y. Liao, and H. Chang, "A 33.2 Gbps/Iter. Reconfgurable LDPC Decoder Fully Compliant with 5G NR Applications," in *Int. Symp. on circuits and Systems*, May 2021.

[13] Sangbu Yun, Dongyun Kam, Jeongwon Choe, Byeong Yong Kong, and youngjoo Lee, "Ultra-Low-Latency LDPC Decoding Architecture Using Reweighted Offset Min-Sum Algorithm," in *Int. Symp. on circuits and Systems*, Oct. 2020.