Efficient VLSI-Architectures and ASIC-Fabrication of Channel Decoder for Contemporary Wireless-Communication Systems

Thesis Submitted

SCHOOL OF COMPUTING AND ELECTRICAL ENGINEERING

in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

by

Anuj Verma

D17029

Under the Supervision of

Dr. Rahul Shrestha

INDIAN INSTITUTE OF TECHNOLOGY MANDI

September, 2024

कर्मण्येवाधिकारस्ते मा फलेषु कदाचन।
मा कर्मफलहेतुर्भूर्मा ते सङ्गोऽस्त्वकर्मणि॥

karmaṇy-evādhikāraste mā phaleṣu kadācana
mā karma-phala-hetur bhūr mā te saṅgo'stv-akarmaṇi

कर्म पर ही तुम्हारा अधिकार है, कर्म के फलों में कभी नहीं... इसलिए कर्म को फल के लिए मत करो।

Your right is to perform your work, but never to the results. Never be motivated by the results of your actions, nor should you be attached to not performing your prescribed duties.

Shree Bhagwat geeta, Chapter 2, Verse 47

aisi vani boliye, man ka aapa khoye,
auran ko sheetal kare, aaphu sheetal hoye.

ऐसी वाणी बोलिए, मन का आपा खोय।
औरन को शीतल करे, आपहु शीतल होय॥

Speak in words so sweet, that fill the heart with joy,
Like a cool breeze in summer, for others and self to enjoy.

---- *Kabir*

Be Happy and Make Others Happy.

Dedicated
to
Omnipresent God,
My mother Sushila Devi, father Ram Niwas Verma,
My dear wife Prachi Gupta, cutie daughter Leena Soni and
My family members.

DECLARATION BY THE RESEARCH SCHOLAR

Thesis Title	Efficient VLSI-Architectures and ASIC-Fabrication of Channel Decoder for Contemporary Wireless-Communication Systems.
Author	Anuj Verma
Student ID	D17029
Supervisor	Dr. Rahul Shrestha

I hereby declare that the entire work embodied within this thesis entitled **"Efficient VLSI-Architectures and ASIC-Fabrication of Channel Decoder for Contemporary Wireless-Communication Systems"** is the result of investigations carried out by me in the School of Computing and Electrical Engineering, Indian Institute of Technology Mandi, Mandi, Himachal Pradesh, India, under the supervision of **Dr. Rahul Shrestha**, Associate Professor, School of Computing and Electrical Engineering, Indian Institute of Technology Mandi, Himachal Pradesh, India.

I also declare that it has not been submitted elsewhere for the award of any degree or diploma. In keeping with general practice, due acknowledgments have been made wherever the work described is based on the findings of other investigators. Any omissions that might have occurred due to oversight or error in judgement are regretted.

Mr. Anuj Verma
Enrollment ID:- D17029

School of Computing and Electrical Engineering,
Indian Institute of Technology Mandi,
Mandi, Himachal Pradesh, India - 175075

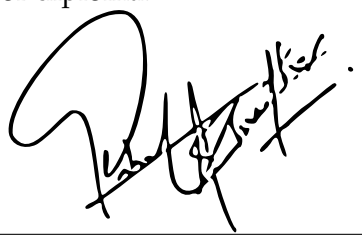
Date: 05-09-2024
Place: IIT Mandi,
Himachal Pradesh, India

DECLARATION BY THE RESEARCH ADVISOR

Thesis Title	Efficient VLSI-Architectures and ASIC-Fabrication of Channel Decoder for Contemporary Wireless-Communication Systems.
Author	Anuj Verma
Student ID	D17029
Supervisor	Dr. Rahul Shrestha

It is certified that the entire work embodied within this thesis entitled **"Efficient VLSI-Architectures and ASIC-Fabrication of Channel Decoder for Contemporary Wireless-Communication Systems"** has been carried out by **Mr. Anuj Verma** Enrollment No. D17029, under my supervision and guidance for the degree of Doctor of Philosophy in the School of Computing and Electrical Engineering, Indian Institute of Technology Mandi, Mandi, Himachal Pradesh, India.

To the best of my knowledge and belief, the present thesis completed by Mr. Anuj Verma fulfills the requirements of the Ph.D. ordinance of the Indian Institute of Technology Mandi, Mandi, Himachal Pradesh, India. It contains the original work of the candidate and no part of it has been submitted elsewhere for any degree or diploma.



Dr. Rahul Shrestha
Associate Professor

School of Computing and Electrical Engineering,
Indian Institute of Technology Mandi,
Mandi, Himachal Pradesh, India - 175075

Date: 05-09-2024

Place: IIT Mandi,
Himachal Pradesh, India

ACKNOWLEDGEMENTS

I heartily bow to the *divine nature* present everywhere. At first, I would like to express my deep and sincere gratitude to my research supervisor, *Dr. Rahul Shrestha*, for allowing me to do research and providing invaluable guidance throughout this work. His dynamism, vision, sincerity and motivation have deeply inspired me. He has taught me the methodology to carry out the work and to present the works as clearly as possible. It was a great privilege and honor to work under his guidance. I feel an energy from him that inspires me to become a kind and decent human being.

I would like to express my heartfelt gratitude.....

To *Dr. Hitesh Shrimali*, for his constant encouragement, support and inspiration throughout the thesis journey. I am thankful to him for sharing his immense knowledge about VLSI design and valuable feedback on my thesis work.

To *Dr. Shubhajit Roy Chowdhury*, for his constructive insights towards my research work.

To *Dr. Sunny Zafar*, for his valuable feedback during the doctoral meetings.

To the *anonymous reviewers* of IEEE TCAS-I, TCAS-II, TVLSI, TVT, VLSID and ISCAS, for their critical assessments of the work carried out in my thesis.

To my mother *Sushila Devi*, father *Ram Niwas Verma*, dear wife *Prachi Gupta*, cutie daughter *Leena Soni*, father-in-law *Gajender Prasad Gupta*, mother-in-law *Poonam Gupta*, brothers *Sachin Verma, Manoj Verma*, sisters-in-law *Ira Verma, Neeti Verma*, nephews *Urjit Soni, Vedansh Verma, Paritosh Soni*, brother-in-law *Bhaskar Goyal, Naman Goyal* and sister-in-law *Shreya Gupta* for their unconditional love, endless prayers, caring and immense sacrifices for educating and preparing me for my future.

To *Sunita Didi and Reshma Didi*, from Kataula (Mandi, Himachal Pradesh) for all the household chores during my daily life routine.

To *Shekhar Arya* and *Vinit Verma* from Qualcomm and Amazon, respectively, for their kind support during the physical designing of our ASIC.

To *Nandit Kaushik, Rahul Sharma, Minku* and *Najrul Islam*, for his self-less help, loving-kindness, technical discussions and his valuable time in creating beautiful memories throughout the thesis journey.

To *Rohit B. Chaurasiya*, *Dinesh Kumar*, *Ashish Joshi*, *Indu Yadav*, *Vijender Kumar Sharma*, and *Saurabh Dhiman* for his loving-kindness towards me throughout the whole thesis journey,

for all the technical discussions, knowledge sharing, clearing my basic doubts of design and help during ASIC-Chip tape-out and testing journey.

To *Rushiraj Jawale* for his immense help during the PCB fabrication.

To *Tarun Verma* from Digital Analog Electronics Lab, *Shivam Prajapati* from Control System Laboratory for their kind cooperation during the ASIC-Chip testing journey.

To *Karunanidhan Pandey, Shubham Telgote, Vartika, Muneeb, Suhail Illikkal, Ashish Tiwari, Kanchan Singh Rana, Shashwat, Shakti, Prashant, Monu* and *Shubham* for all the fruitful discussions during the course work and for their time in creating unforgettable memories.

I am greatly indebted to all the honorable teachers who taught me during the various stages of my life. Without any doubt, their teaching and guidance have completely transformed me into the person that I am today.

Finally, I would like to thank all the people who have supported me to complete the thesis work directly or indirectly.

Anuj Verma
Indian Institute of Technology Mandi
Mandi, Himachal Pradesh, India - 175075

ABSTRACT

The channel decoder exploits the data redundancy to facilitate the detection and correction of bit errors. Recently, the third generation partnership project (3GPP) advocated quasi-cyclic low-density parity-check (QC-LDPC) and polar codes as the standard channel codes (for data transmission and control signaling) in the physical layer specifications of fifth-generation new radio (5G-NR) standard, due to its near Shannon-limit performance and lower-decoding complexity with parallelism. In this doctoral thesis work, we present a partially-parallel QC-LDPC decoder architecture for 5G-NR technology, based on a layered min-sum decoding algorithm. This architecture exploits the iterative decomposition of combined variable-node and check-node processing unit which leads to area optimization and lower hardware complexity. This VLSI architecture of QC-LDPC decoder has been field-programmable logic-array (FPGA) prototyped and its implementation results are compared with various reported works that shows lower hardware-utilization up to 87%. In addition, a new fully-parallel high-throughput VLSI architecture of QC-LDPC decoder that is compliant to the 5G-NR wireless communication standard has been proposed in this work. Its FPGA implementation shows that our decoder can operate at a clock frequency of 102 MHz and delivers a throughput of 2.9 Gbps which is 20× better than existing LDPC decoder architectures. Furthermore, we proposed a hardware-friendly QC-LDPC decoding algorithm based on the new logarithmic-likelihood-ratio compound (LLRC) segregation technique. Based on this technique, a novel hardware-efficient QC-LDPC decoder architecture has been presented in this thesis. Performance analysis has shown that the suggested LLRC-segregation based decoding algorithm delivers an adequate frame-error rate (FER) of 10^{-5} between 1 to 6.5 dB of SNR range for various standard code-rates. Suggested QC-LDPC decoder is post-route simulated and implemented on the FPGA platform. It operates at a maximum clock frequency of 135 MHz and delivers a peak throughput of 11.02 Gbps. On comparing with the relevant works, our decoder delivers 2.2× higher throughput and 8.3× better hardware-efficiency.

Subsequently, this thesis presents an implementation-friendly simplified offset min-sum (SOMS) decoding algorithm for QC-LDPC code that alleviates the computational complexity of the QC-LDPC channel decoding. A novel parallel and hardware-efficient architecture of the QC-LDPC decoder based on the suggested SOMS algorithm has been

presented here. It alleviates the routing-complexity, delivers lower decoding-latency and higher data-throughput. The suggested SOMS algorithm delivers an adequate FER of 10^{-5} at SNR of 1.3 dB with a code-rate of 1/3. Subsequently, our QC-LDPC decoder has been hardware-implemented on the FPGA platform that operates at the maximum clock frequency of 128.36 MHz. This proposed QC-LDPC decoder delivers a peak throughput of 13.3 Gbps and latency of 0.77 µs. On comparing with the reported implementations, the proposed decoder delivers 7.5× higher data-throughput and 34% better hardware-efficiency. Eventually, a hardware-efficient and high-throughput reconfigurable channel decoder architecture has been proposed for unified decoding of LDPC and polar code. It has been designed based on the new dataflow technique for reconfigurable decoding that requires fewer hardware-resources in the decoder design. Furthermore, this reconfigurable LDPC/polar decoder has been application-specific integrated-circuit (ASIC) fabricated, occupying an area of 1.96 mm^2. This ASIC chip supports multiple code-rates and code-lengths that are compliant to massive machine type communication (mMTC) and ultra-reliable and low-latency communication (URLLC) applications of the 5G-NR wireless communication standard. At the supply voltage of 1.2 V, the proposed decoder chip operates at the measured clock frequency of 72.7 MHz and delivers a data-throughput of 3.35 Gbps which is 4× higher than the state-of-the-art implementations. It also consumes 15.8% lesser silicon area and achieves 2.5× better hardware-efficiency in comparison to the contemporary works.

Keywords: Channel Decoder, LDPC Codes, Polar Codes, 5G New-Radio, Digital VLSI Architecture, Field-Programmable Gate-Array (FPGA), Application-Specific-Integrated-Circuit (ASIC) Design and Fabrication, FPGA Prototyping, Wireless communication.

List of Abbreviations

1G: First Generation

2G: Second Generation

3G: Third Generation

3GPP: Third Generation Partnership Project

4G: Fourth Generation

5G: Fifth Generation

ACM: A-posteriori LLR Computation Module

ASIC: Application Specific Integrated Circuits

AWGN: Additive White Gaussian Noise

BER: Bit Error Rate

BG: Base Graph

BP: Belief Propagation

BPSK: Binary Phase Shift Keying

BRAM: Block Random Access Memory

CE-LLRC: Compressed Extrinsic LLRC

CF: Carrier Frequency

CMBU: Combiner Unit

CN: Check Node

CRC: Cyclic Redundancy Check

CRT: Cyclic Rotational Technique

CVPU: Check Nodes & Variable Nodes Processing Unit

DSMN: Data Selection Multiplexer Network

ELSM: Extrinsic LLR Storage Memory

eMBB: enhanced Mobile Broadband

FEC: Forward Error Correction

FER: Frame Error Rate

FLP: Floating Point

FOM: Figure Of Merit

FP: Fixed Point

FPGA: Field Programmable Gate Array

FSK: Frequency Shift Keying

GEs: Gate Equivalents

gNB: Next generation Node B

HDL: Hardware Description Language

HDM: Hard Decision Memory

HDRM: Hard Decision Register Memory

HUE: Hardware Utilization Efficiency

HWPN: Hard Wired Permutation Network

ILA: Integrated Logic Analyzer

IMB: Initial Memory Bank

IMT: International Mobile Telecommunications

IMU: Initial Memory Unit

IMUU: Initialization & Memory Updating Unit

IoT: Internet-of-Thing

IP: Intellectual Property

IRMB: Initial Register Memory Bank

ITU: International Telecommunication Union

LBC: Linear Block Code

LDPC: Low Density Parity Check

LLR: Logarithmic Likelihood Ratio

LLRC: Logarithmic Likelihood Ratio Compound

LMB: Left Memory Bank

LMR: Left Memory Routing

LMS: LDPC Min Sum

LNA: Low Noise Amplifier

LSB: Least Significant Bit

LTE: Long Term Evolution

LUT: Look Up Table

MCU: Magnitude Computation Unit

MDR: Memory & Data Routing

MMN: Memory Multiplexer Network

mMTC: massive Machine Type Communication

MS: Min Sum

MSAU: Min Sum Approximation Unit

MSB: Most Significant Bit

MSESM: Memory Selection and Extrinsic LLR Storage Memory

MSMN: Memory Selection Multiplexer Network

MSO: Mixed Signal Oscilloscope

MSR: Memory Selection Router

MST: Memory Selection Technique

MSU: Min Sum Unit

mVG: Minimum Value Generator

MWR: Memory Writing Router

MWT: Memory Writing Technique

NR: New Radio

OMS: Offset Min Sum

PCB: Printed Circuit Board

PCM: Parity Check Matrix

PE: Processing Element

PMS: Polar Min Sum

PSK: Phase Shift Keying

PTLR: Peak Throughput to Latency Ratio

QAM: Quadrature Amplitude Modulation

QC: Quasi Cyclic

QFN: Quad Flat No-lead

QoS: Quality of Service

QPSK: Quadrature Phase Shift Keying

RAN: Radio Access Network

RAT: Radio Access Technology

RF: Radio Frequency

RMR: Right Memory Routing

SM: Sign Magnitude

SNR: Signal to Noise Ratio

SOMS: Simplified Offset Min Sum

STA: Static Timing Analysis

SU: Sign Unit

TC: Two's Complement

TCSM: Two's Complement To Sign Magnitude

UMC: United Microelectronics Corporation

URLLC: Ultra Reliable Low Latency Communication

VCUM: Variable Node And Check Node Updation Module

VLSI: Very Large Scale Integration

VN: Variable Node

Wi-Fi: Wireless Fidelity

WiMAX: Worldwide Interoperability for Microwave Access

Notations

k: Information bits

m: Parity bits

n: Codeword bits

R: Code-rate

H: Parity check matrix

B: Base graph matrix

$\mathcal{G}$: Generator matrix

$\mathcal{L}$: Left-to-right message

$\mathcal{R}$: Right-to-left message

$\Upsilon(x, y)$: Min-sum operation on x and y

z: Expansion factor

L: Quantized logarithmic likelihood ratio

$\widehat{X}$: Estimated codeword

τ: Matrix transpose

T: Number of all 'non -1' elements in the entire B matrix

σ^2: Noise variance

E: Extrinsic LLRC matrix

e: Extrinsic LLRC sub-matrix

V: Variable-node message computation

C: Check-node message computation

$\mathbb{A}$: A-priori LLR matrix

I: Current iteration

r: Base graph matrix row

i_{max}: Maximum number of iterations

η_H: Base graph matrix element

Q: Quantization-bits

P: Index vector

Θ_T: Data-throughput

f_{max}: Maximum clock frequency

η_{clk}: Number of clock cycles consumed for single decoding iteration

S: Selected LLRC matrix

D: A-posterori LLRC matrix

W: Weighted elements

syn: Syndrome check

$R(x)$: x^{th} rotated LLRC

$Re(x)$: x^{th} re-rotated LLRC

L_c: LLR compound

$min1$: First minimum value

$min2$: Second minimum value

idx: Index of first minimum value

t_b: Number of all 'non -1' elements in the a single B matrix row

Λ_D: Latency

$\mathbb{H}_u$: Hardware utilization

$\mathbb{C}$: Number of clock cycles consumed for the processing of single base graph matrix layer

$\mathbb{L}$: Base graph matrix processed layers

m_b: Rows (check-nodes) in base graph matrix

n_b: Columns (variable-nodes) in base graph matrix

k_b: Information bit columns in base graph matrix

d_c: Average degree of check-node

d_v: Average degree of variable-node

β: Offset value

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Communication Model

A typical communication model is the combination of the transmitter, channel, receiver, as shown in Fig. 1.1. A typical transmission path of a wireless communication system is comprised of transducer, source encoder, encryption, channel encoding, bandpass modulation and multiplexing. A real-world information sources, viz. speech, images, text, video, etc., are transformed into electrical signals using an input transducer. Subsequently, the source encoder removes the redundant information from signals to maximize the resource utilization. These signals are encrypted using encryption standard for the security purpose and unauthorized access. Eventually, these signals are passed through channel encoder unit that introduces some amount of redundancy to make signal more robust against noise. Afterwards, these signals are modulated by suitable modulation technique (like PSK, FSK, and QAM etc.) for antenna transmission. Finally, the modulated signals are multiplexed with other signals using various multiplexing techniques like time division multiplexing (TDM) or frequency division multiplexing (FDM) to share the valuable bandwidth.

A channel in wireless communication networks is a signal transmission medium that is unpredictable about noise, interference, distortion, scattering etc., resulting the noisy and distorted received signals due to fading interference. On the other side, a receiver is the

collection of de-multiplexing , demodulation, channel decode, decryption, source decoder, and transducer. The noisy/distorted received signal is separated from other signals by using de-multiplexing. These individual signals are demodulated and the original informative signal are recovered. Subsequently, the channel decoder eliminates the redundant bits from demodulated signal. The decryption module removes the security and convert the encrypted signal into bit-streams. Afterwards, the source decoder helps to retrieve the signal into estimated transmitted signal. At last, the output transducer converts the estimated received signal into its corresponding real-world sources (speech, image, text, video, etc.), as shown in Fig. 1.1.

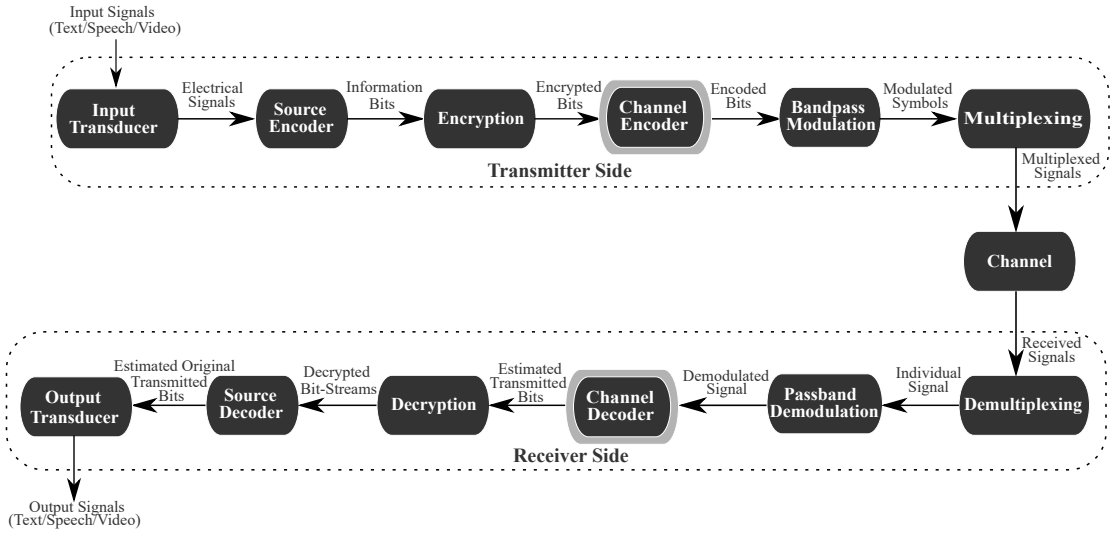


Figure 1.1: Typical system model for wired and wireless communication networks.

There are various types of communication: mobile, satellite, deep-space, radio, and television communications. All these communication models are operated with the aid of high data-rates and low-latency devices. Hence, communication systems can be categorized into various types based on their characteristics, as follows:

- **Analog and Digital Communications:** The Analog communication system transmits the continuous signal over the analog communication channels. The analog modulation techniques such as amplitude modulation (AM), frequency modulation (FM), and phase modulation (PM) encode information over carrier signals. For example,

traditional analog telephony and AM/FM radio broadcasting. Digital communication systems convert all signals into discrete binary signals and transmit them in digital format. Digital modulation techniques such as amplitude shift keying (ASK), frequency shift keying (FSK), phase shift keying (PSK), binary phase shift keying (BPSK), quadrature phase shift keying (QPSK), and quadrature amplitude modulation (QAM) offer improved noise-immunity and greater bandwidth-efficiency. For example, digital television (DTV) and digital data transmission over computer networks.

- **Wired and Wireless Communications:** Wired networks utilize the physical medium, such as wires or cables, for the signal transmissions between the transmitter and receiver sides. These types of networks provide reliable, secure, and high-speed communication, such as landline telephones and fiber optic systems. On the contrary, wireless-communication networks establish connections among devices by using wire-free networks such as cellular and broadband networks.

- **Broadcasting and Satellite Communications:** In broadcast communication, systems transmit audio and video signals over radio-frequency (RF) channels for radio and television receivers, respectively. Satellite communications provide coverage over large geographic areas. These types of communication are used for the global positioning system (GPS), satellite internet access and remote sensing systems, etc.

A telecommunications system comprises individual interconnected devices or components designed to facilitate information exchange between multiple locations or users. Overall communication systems are necessary for fostering connectivity, collaboration, efficiency, and social interactions in both personal and professional contexts. The communication standard plays an essential role in various applications, including mobile and satellite communications, broadcasting, remote-working, and data-driven transmissions. Modern communication provides a significant advancement due to the transformations in technologies and societal dynamics. All the networks, such as digital communication, space communications, social media and multimedia communication, virtual reality (VR)

and augmented reality (AR), mobile communication, and various other real-time communications, enhancing the productivity of business and academic organizations. The proliferation of high-speed and low-latency reliable data communications has transformed the organizations to personalize communication and target connectivity more efficiently.

1.2 Evolution of Wireless Communication Standard

The evolution of wireless communication standards has marked a revolutionary journey through technological innovations and relentless pursuit of connectivity. From the humble beginnings of analog cellular networks in the early 1980s to cutting-edge fifth-generation (5G) technology, wireless communication has dramatic transformation of technology shifting, faster data speeds, low-latency, and greater connectivity. A pictorial representation for the evolution of wireless communication standards has been illustrated in Fig. 1.2 which shows how technologies have evolved over the past decades. The evolution of wireless communication standards continue to unfold promising technologies, even finding greater possibilities and opportunities for connectivity over the years. Each generation has introduced a significant improvement in performance, efficiency, and functionality to revolutionize services and applications.

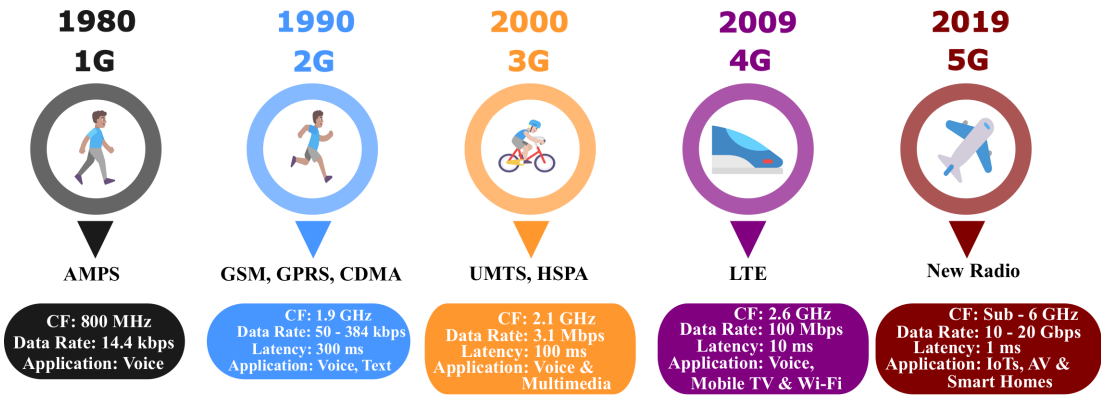


Figure 1.2: Pictorial representation for the evolution of wireless-communication standard over the past decades.

1.2.1 First Generation

At the beginning of the 1980s, the first-generation (1G) wireless communication standard enabled the mobile voice network and liberated it from the constraints of wired communication networks. These systems were operating on analog signals over the wireless-communication channels. Several technologies, such as the advanced mobile phone system (AMPS), nordic mobile phone system (NMTS), and European total access communication system (ETACS) transmit the signals over the 800 MHz carrier frequency (CF) as specified in Fig. 1.2. The frequency-modulated (FM) signals have duplex channel capacity with a 30 kHz narrow bandwidth. The transmitted data-rates for the 1G communication standard is 14.4 kbps, which are primarily designed for voice communications and have limited data transmission capabilities. Signal algorithms were employed to filter out unwanted noise and distortions from the received signals. Hence, the hardware implementations for analog modulations/demodulations and channel equalizers require precise tuning and calibration to ensure reliable communication in diverse environments. Despite all these challenges, the successful deployment of the 1G communication standard laid the foundation of wireless communication technology in the interconnected world.

1.2.2 Second Generation

Global standardization and advanced features like call waiting and forwarding have enhanced the functionality and convenience of mobile networks. A new communication standard, named second-generation (2G), has been introduced to drive several new technologies. The digital modulation of all transmitted signals occurs over the carrier frequency ranges of 850 MHz to 1900 MHz. The transitions from 1G to 2G wireless communication standards incorporate digital modulations (ASK, FSK, and PSK), higher data-services, increased spectral-efficiency, and improved power-management. In 1990s, the 2G communication networks introduced a new digital technology named as global system for mobile (GSM) communication. The GSM standard supports the 14.4−60 Kbps data-rates that is

sufficient for short message services (SMS) and email services in wireless systems. However, the 2G wireless-communication standard has employed voice as well as data-services with GSM technology.

In mid-1990s, the code division multiple access (CDMA) technique was introduced in 2G that supported high data-rate, better spectral-efficiency, and more number of users connectivity. In early 2000, a new 2.5G and 2.75G were incorporated into 2G, where the general packet radio service (GPRS) technique was successfully deployed with data-rates of 171 kbps. The most popular technology, CDMA2000 supports higher data-rates than CDMA networks and also can provide 384 Kbps of data-rate and bandwidth of 200 kHz, as shown in Fig. 1.2. GSM and CDMA technologies employ convolutional coding technique that add some redundancy parity bits in information bits at the transmission side. Such redundant bits are used to detect and correct errors in signals at the receiver side. However, convolutional decoding-algorithms like Viterbi algorithm was used at the receiver side to recover error-free transmitted data. Such coding schemes were used to trade-off between error correction capability, spectral-efficiency, and data-rates. Therefore, the hardware implementation for the 2G communication standard involves a complex realization of components and substations to offer reliable, secure, and efficient wireless-communication services for mobile users. Each hardware component contributes seamless connectivity and functionality to the overall deployment of 2G mobile networks.

1.2.3 Third Generation

In 2004, the third-generation (3G) wireless communication standard was developed due to an increment in connectivity and network services. The transitions from 2G to 3G introduced mobile internet access, email & multimedia messages, video streaming, and improved quality of services (QoS). A wider range of services and applications have been supported by the 3G wireless-communication systems. It also supports a universal mobile terrestrial/telecommunication system (UMTS) to offer faster speed, a robust platform for the mobile industry, and enhanced capabilities. The transmitted signals for 3G standard

were digitally modulated on the 850 MHz to 2100 MHz carrier frequency, as presented in Fig. 1.2. Such higher frequency bands support multimedia services and also allow the data-rates of 500−700 Kbps. Typically, the bandwidth supported by 3G UMTS technology is 5 MHz which allows to connect more number of systems in the network.

Further, 3G networks have been transformed into 3.5G and 3.75G systems with high-speed downlink packet access (HSDPA) and high-speed uplink packet access (HSUPA) technique to support 3.1 Mbps maximum data-rates, as illustrated in Fig. 1.2. This 3G HSPA technology provides all the services with enhanced speed and more mobility. The physical layer for 3G wireless communication standard supports QPSK and QAM digital-modulated signals with convolutional and turbo channel-coding schemes for error detection & corrections. These convolutional and turbo channel-coding algorithms helped to improve reliable communication and data transmission over noisy and fading channel conditions. However, the hardware implementation for the 3G wireless-communication standard has costly infrastructures and requirements to achieve reliable, high speed data communication over wireless channels. Therefore, the key features of the 3G mobile network are various smartphone applications, faster web-browsing, video-calling, TV streaming, spectrum-efficiency.

1.2.4 Fourth Generation

Fourth-generation (4G) wireless communication standard is an enhanced version of 3G networks. This network has been developed by the institute of electrical and electronics engineers (IEEE) organization, called long-term evolution (LTE) or advanced LTE (LTE-A), allowing higher data-rates and handling advanced multimedia services. It supports wider bandwidth, typically up to 20 MHz to accommodate higher data speeds, improved spectral-efficiency, and online gaming services. The 4G LTE networks are a combination of complex modulation schemes and carrier aggregation to multiple uplink and downlink capabilities. The 4G LTE standard was introduced in 2010 with a carrier frequency range from 700 MHz to 2600 MHz, as demonstrated in Fig. 1.2. These frequency bands include

both lower bands (e.g., 700 MHz, 800 MHz) for wider coverage and higher bands (e.g. 1700 MHz, 2600 MHz) for increased capacity and data-speeds. However, the 4G LTE standard supports worldwide interoperability for microwave access (WiMAX), and wireless fidelity (Wi-Fi) technologies with data-rates of 100 Mbps.

The physical layer of the 4G LTE wireless-communication standard has QPSK and QAM digital modulation schemes with turbo, convolutional, and low-density parity-check (LDPC) channel-coding techniques. These error detection and correction techniques provide reliable data transmission between transmitter and receiver by adding some redundancy to transmitted data. These redundant bits are mitigated at the receiver side by employing sophisticated decoding algorithms. These decoding algorithms can achieve high data-rates, low error-rates, and robust performance in diverse communication environments. Therefore, the hardware implementation of these channel decoding algorithms plays a crucial role in the deployment of the physical layer of 4G LTE wireless communication standard. However, there is a trade-off between hardware resource consumption and achieved data throughput from a hardware perspective. Hence, the 4G LTE system supports high-definition (HD) video streaming, online gaming, reduced latency for mission critical applications, and enhanced security.

1.2.5 Fifth Generation

Fifth-generation (5G) has opened up a new range of possibilities for network connection and data transmission applications within the area of wireless-communication standards. The international telecommunication union (ITU) and third generation partnership project (3GPP) specified all the technical details for 5G communication systems. The 5G radio access technology (RAT) supports various new technology drivers such as self-organizing networking, scalable orthogonal frequency division multiplexing (OFDM), advanced channel coding, low latency design, adaptive beam framing, non-orthogonal multiple access (NOMA) and device-to-device (D2D) networks. Due to the introducing of these new technology drivers, the 5G radio access networks (RANs) are identified by 5G new radio (NR).

However, the transformation of 4G-LTE to 5G-NR is classified into two categories: (a) non-standalone mode (b) standalone mode. In non-standalone mode, both 4G-LTE spectrum and 5G-NR spectrum will be utilized, and controlling signals will be connected to 4G-LTE networks. On the contrary, there will be a dedicated 5G-NR core network for a higher bandwidth spectrum in 5G-NR standalone mode.

Such standardization must be compliant with the specifications of international mobile telecommunications (IMT)-2020 that has been supported by ITU. Therefore, these specifications are more profoundly advanced than the 4G-LTE technologies. These high-end 5G-NR specifications are distributed over three use cases: enhanced mobile broadband (eMBB) that addresses data-driven cases of accessing multimedia content, massive machine type communication (mMTC) which is characterized by a large volume of connected devices (like narrow-band IoT applications) have lower cost and longer battery life, and ultra-reliable low-latency communications (URLLC) targets for the delivery of critical communication. To support these high-end cases, the 5G-NR standard has a specified peak data-rates of 10–25 Gbps, latency $\leq$ 1 ms, spectral-efficiency of 15–30 bits/s/Hz, connection-density of 106 devices/km^2, bandwidth of 400 MHz, carrier frequency ranges sub-6 GHz and mmWave bands focus from 24.25 GHz to 52.6 GHz etc, as presented in Fig. 1.2. The 5G-NR technology is envisioned to render enhanced connectivity to users and materialize the digitization of diverse industrial verticals. Therefore, all the hardware blocks of analog RF frontend and digital baseband in the 5G-NR physical layer must be designed efficiently to match the aforementioned specifications of high-throughput (for eMBB), enhanced hardware/spectral-efficiency (for mMTC), and low latency (for URLLC).

In recent release-16 of 5G-NR standardization for the physical layer, the 3GPP has advocated LDPC and polar code as standard channel codes for data transmission and control signaling, respectively. The near Shannon limit performance of LDPC codes guarantees that channel bandwidth and capacity are used effectively via the utilization of these codes. These codes are more dynamic and diversified in terms of recent wireless-communication standards due to their high coding-gain, and resistance to error bursts capabilities. In 5G-

NR communication, hardware implementations for LDPC and polar decoders play a crucial role in achieving high-speed and reliable data transmission. Hence, the hardware aspects of LDPC and polar decoders in 5G-NR systems involve sophisticated architectures, processing units, memory structures, and control logic optimized for high-speed, low-latency decoding of these codes.

1.3 System Level Overview of Wireless Communication

A schematic representation of physical layer for the wireless-communication system, is presented in Fig. 1.3. On the transmitter side, real-world information sources viz. speech, image, text, video, etc are transduced to analog electrical signals and subsequently, digitized into '$\mathbb{K}$' binary digits or bits. Such $\mathbb{K}$ bits are segregated into multiple frames of k-bits each (where $k \in \mathbb{K}$) and these k information bits must be reliably communicated between transmitter & receiver while transmitting a frame. As shown in Fig. 1.3, such k-bits are fed to the channel encoder where additional bits (generated by processing k information bits) are suffixed or prefixed with k information bits resulting into a codeword of n-bits (i.e. code-length) such that $n > k$. Thus, $n-k$ bits are referred to as parity bits. These parity bits mitigate the adverse effects of external noise, multi-path fading, and modulation interferences on the transmitted signal across wireless channels. Consecutively, the codeword of n-bits from the encoder is passed into a rate-correlator for puncturing this n-bits sequence based on various code-rates, as specified by wireless standards. Such a rate-correlator enhances channel bandwidth-efficiency and flexibility of the communication system without incrementing its complexity. Furthermore, these punctured bits are modulated by a bandpass modulator and transmitted after being processed by an RF up-conversion mixer along with a power amplifier and antenna, via additive-white Gaussian-noise (AWGN) channel, as shown in Fig. 1.3.

At the receiver side, a band-selection filter selects the signal frequency band from the continuous-time received signals (at the output of the receiving antenna) corresponding

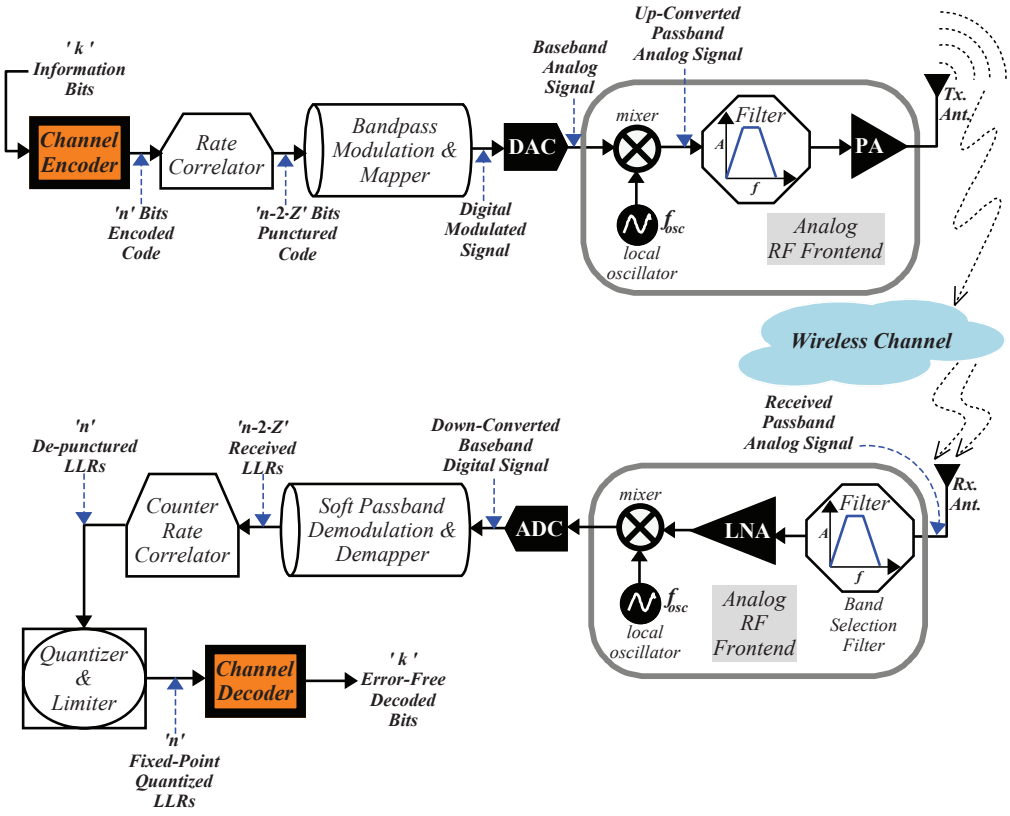


Figure 1.3: System-level overview for the conventional physical layer of a wireless communication system illustrating the roles of channel encoder & decoder at the transmitter & receiver side, respectively.

to the specifications of wireless standards. High frequency RF signal from this selected frequency band is passed to a low noise amplifier (LNA) for suppressed noise amplification (to restore the signal strength) and subsequently, RF down-converted to baseband signal using the mixer, as shown in Fig. 1.3. Such continuous-time baseband signal is digitized by an analog-to-digital converter (ADC) and its output is subsequently soft-demodulated to generate the logarithmic-likelihood-ratios (LLRs) for all the transmitted bits. The soft-demodulated generated LLRs can be mathematically formulated by (1.1)

$$L(x) = log\left(\frac{P(x=0)}{P(x=1)}\right). \tag{1.1}$$

It states that if $P(x{=}0){>}P(x{=}1)$ then $L(x)$ is positive, indicating the decoded bit is '1', else $L(x)$ is negative that indicates the decoded bit is '0' or vice-versa. These LLRs are further processed by a counter rate-correlator that depunctures the bit-positions and some redundant bit-positions by allocating '0' (null) values, for code-rate synchronization with the rate-correlator at the transmitter side, as shown in Fig. 1.3. These de-punctured LLRs from counter rate-correlator are represented in (r_q, s_q) quantization format where r_q and s_q bits represent integer and fractional parts, respectively, after passing through quantizer & limiter" module. Eventually, these quantized LLRs are fed to the channel decoder that generates k error-free decoded bits at the receiver corresponding to the source information of the transmitter.

1.4 Channel Coding

In telecommunication systems, channel coding or forward error correction (FEC) is a technique that is used for detecting and correcting the error in received bits at the receiver end. Error detections and corrections can be accomplished by adding some redundant bits in the transmitted information bits. These redundancy bits allow the receiver not only to detect errors that may occur anywhere in the message but often to correct the errors in received codes. Hence, the channel coding technique uses the FEC codes for detecting and correcting the error at the receiver end. The ITU and 3GPP has standardized specific FEC codes for the different communication standards to achieve specific requirements. Thus, 3GPP impels the convolutional codes for 3G wireless-communication standard, turbo codes has been incorporated with 4G-LTE communication systems, and linear block codes (LBCs) such as LDPC and polar codes are pivoted for 5G-NR contemporary wireless-communication standard. In 5G-NR networks, LDPC codes carry data and paging information while polar code conveys control signals, scheduling decisions and grants.

1.4.1 Low-Density Parity-Check Codes

LDPC code was pioneered by Robert G. Gallager in 1963 [6]. Due to its near Shannon limit performance, LDPC codes have been adopted by various wireless-communication standards like DVB-S2, Wi-Fi, WiMAX, and 5G-NR [1, 7, 8, 9]. LDPC codes can be characterized by sparse parity-check-matrix (PCM), typically meaning that the majority of elements are zeros, as presented in Fig. 1.4 (a). This PCM has the size of $(n-k) \times n$ where n represents the codeword bits, k denotes the information bits, and the ratio of information bits and codeword bits is known as code-rate (R) as expressed in (1.2).

$$R = \frac{k}{n} \tag{1.2}$$

These LDPC codes can be graphically represented by a bipartite graph, also known as Tanner graph [10]. The Tanner graphs are constructed with the aid of PCM, as shown in Fig. 1.4 (b). The Tanner graph is represented with check nodes (CNs) and variable nodes (VNs) and edges denote the connections between CNs and VNs, as presented in Fig. 1.4 (b). For example: CN1 has the edge connections with VN1, VN2, VN3, and VN5 that represent CN1 propagates message to all connected VNs.

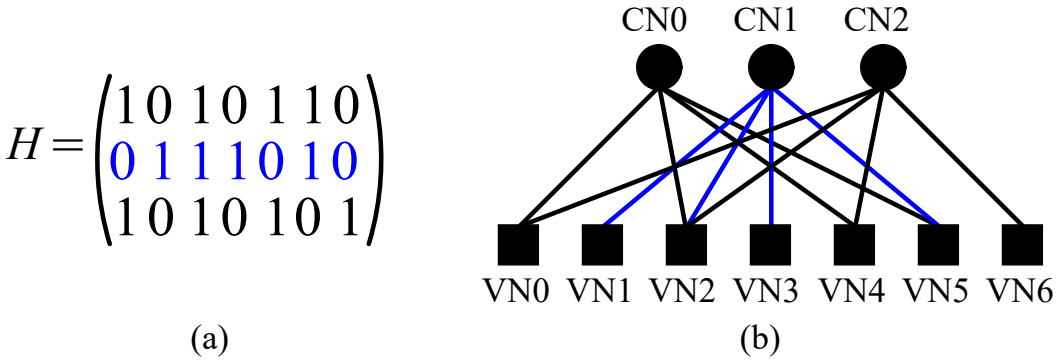


Figure 1.4: (a) Parity check matrix (PCM) that has 3 rows and 7 columns for (7, 4) LDPC code. (b) Basic representation of Tanner graph that has been derived by PCM H matrix with 3 check nodes (CNs) and 7 variable nodes (VNs).

Due to the advancement in wireless-communication technology, LDPC codes have been updated to quasi-cyclic (QC) LDPC to achieve better performance and hardware-efficiency.

Hence, QC-LDPC code is the enhanced version of LDPC code, represented using PCM (denoted by H) that is an expanded version of base graph B matrix where every element of this matrix is replaced by permutation identity matrix ($P_0/P_1/P_2$) or zero matrix P_{-1} with an expansion factor of z, as shown in Fig. 1.5 (a) & (b) where $z = 3$. The base graph B matrix and expansion factor (z) have been standardized and compliant with the specific wireless-communication standards. Thereby, the expansion factor z for 5G-NR has been quantified based on $z = a \times 2^j$ where $a \in \{2, 3, 5, 7, 9, 11, 15\}$ and $0 \leq j \leq 7$. The maximum value of z is limited to 384 with $a = 3$ and j =7, based on 5G-NR standard [1, 2].

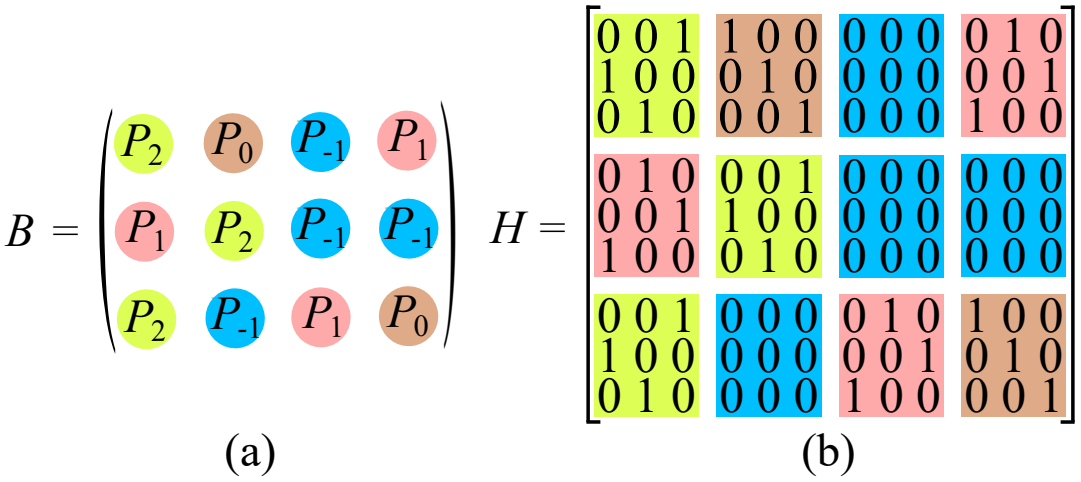


Figure 1.5: (a) Schematic illustration of an example base graph ($B_{3\times4}$) matrix. (b) Parity check matrix (H) that is the expanded version of base graph ($B_{3\times4}$) matrix with an expansion factor (z) of 3, replacing B matrix elements with permutational identity and zero matrices.

In the 5G-NR standard, LDPC code pertains to the QC-LDPC codes structure that has been represented in Fig. 1.6. The base graph matrix B has been divided into sub-matrices as follows:

- **A:** denotes the information bits that are systematically used for data transmission.
- **B:** represents the dual-diagonal matrix that corresponds to parity bits. This matrix is a square matrix with the bi-diagonal structure where the weight of the first column is 3 and the remaining three columns consist of an upper bi-diagonal structure.
- **E:** corresponds to the extension check matrix that is used to decide parity bits for

different 5G-NR standard code-rates.

- **O:** represents as zero matrix.
- **I:** denotes identity matrix.

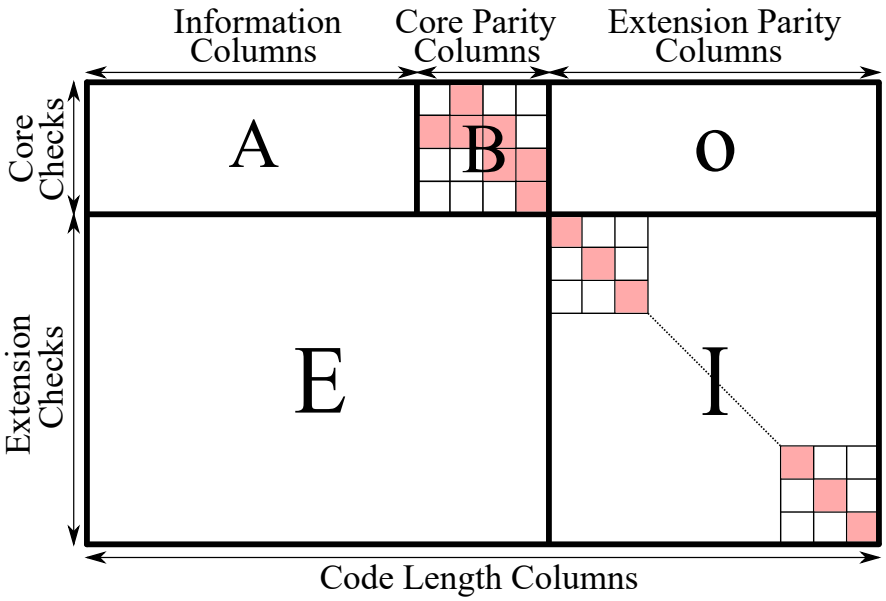


Figure 1.6: Structure diagram of base graph B matrix for QC-LDPC codes in 5G-NR wireless communication standard.

Therefore, the 5G-NR communication standard supports two different base graph matrices: $B1$ and $B2$. Both $B1$ and $B2$ matrices have identical structures, as presented in Fig. 1.6. The base graph $B1$ matrix has 46 rows and 68 columns, as presented in Fig. 1.7, has maximum information bits of 8448 (37.5% more bits than 4G-LTE standard) and supports code-rates from $1/3$ to $8/9$ whereas $B2$ has 42 rows and 52 columns with a maximum of 3840 information bits and supports code-rates from $1/5$ to $2/3$. In addition, dimensions of $B1$ and $B2$ matrices for various code-rates are listed in Table 1.1.

1.4.2 Polar Codes

Polar codes are (n, k) linear block codes where $n{>}k$ [11]. To begin with, k information bits are transmitted over the reliable channel, referred as free bits whereas the remaining $(n{-}k)$

307	19	50	369	-1	181	216	-1	-1	317	288	109	17	357	-1	215	106	-1	242	180	330	346	1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
76	-1	76	73	288	144	-1	331	331	178	-1	295	342	-1	217	99	354	114	-1	331	-1	112	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
205	250	328	-1	332	256	161	267	160	63	129	-1	-1	200	88	53	-1	131	240	205	13	-1	-1	-1	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
276	87	-1	0	275	-1	199	153	56	-1	132	305	231	341	212	-1	304	300	271	-1	39	357	1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
332	181	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1
195	14	-1	115	-1	-1	-1	-1	-1	-1	-1	-1	166	-1	-1	-1	241	-1	-1	-1	-1	51	157	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1
278	-1	-1	-1	-1	-1	257	-1	-1	-1	1	351	-1	92	-1	-1	-1	253	18	-1	225	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1
9	62	-1	-1	316	-1	-1	333	290	-1	-1	-1	-1	-1	114	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1
307	179	-1	165	-1	-1	-1	-1	-1	-1	-1	-1	18	-1	-1	-1	39	-1	-1	224	-1	368	67	-1	170	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1
366	232	-1	-1	-1	-1	-1	-1	-1	-1	321	133	-1	57	-1	-1	-1	303	63	-1	82	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	101	339	-1	274	-1	-1	111	383	-1	-1	-1	-1	-1	354	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
48	102	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	8	-1	-1	-1	47	-1	-1	-1	-1	188	334	115	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
77	186	-1	-1	-1	-1	-1	-1	-1	-1	174	232	-1	50	-1	-1	-1	-1	74	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
313	-1	-1	177	-1	-1	-1	266	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	115	-1	-1	370	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
142	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	248	-1	-1	137	89	347	-1	-1	-1	12	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
241	2	-1	-1	-1	-1	-1	-1	-1	-1	210	-1	-1	318	-1	-1	-1	-1	55	-1	-1	-1	-1	-1	-1	269	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	13`	-1	338	-1	-1	-1	-1	-1	-1	-1	57	-1	-1	-1	-1	-1	-1	-1	-1	289	-1	57	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
260	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	303	-1	81	358	-1	-1	-1	375	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	130	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	163	280	-1	-1	-1	-1	132	4	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
145	213	-1	-1	-1	-1	-1	344	242	-1	197	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
187	-1	-1	206	-1	-1	-1	-1	-1	264	-1	341	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	59	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	205	-1	-1	-1	102	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	328	-1	-1	-1	213	97	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
30	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	11	233	-1	-1	-1	22	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	24	89	-1	-1	-1	-1	-1	-1	-1	61	-1	-1	-1	-1	-1	-1	-1	27	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
298	-1	-1	158	235	-1	-1	-1	-1	-1	-1	339	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	234	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	72	-1	-1	-1	-1	17	383	-1	-1	-1	-1	-1	-1	312	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
71	-1	81	-1	76	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	136	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	194	-1	-1	-1	-1	194	-1	101	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
222	-1	-1	-1	19	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	244	-1	274	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	252	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	5	-1	-1	-1	147	-1	-1	-1	-1	-1	-1	78	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
159	-1	-1	-1	-1	-1	-1	-1	-1	-1	229	-1	-1	260	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	90	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	100	-1	-1	-1	-1	-1	215	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	258	-1	-1	256	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
102	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	201	-1	175	-1	-1	-1	-1	-1	-1	-1	-1	-1	287	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	323	8	-1	-1	-1	-1	-1	-1	-1	-1	361	-1	-1	-1	-1	-1	-1	-1	-1	-1	105	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
230	-1	-1	-1	-1	-1	-1	148	-1	-1	-1	-1	-1	-1	-1	202	-1	312	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	320	-1	-1	-1	-1	335	-1	-1	-1	-1	-1	2	-1	-1	-1	-1	-1	-1	-1	-1	-1	266	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
210	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	313	297	-1	-1	21	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	269	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	82	-1	-1	-1	-1	-1	-1	-1	-1	-1	115	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
185	-1	-1	-1	-1	-1	-1	-1	-1	177	289	-1	214	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	258	-1	93	-1	-1	-1	346	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	297	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
175	-1	-1	-1	-1	-1	-1	-1	37	-1	-1	-1	-1	-1	-1	-1	-1	312	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	52	-1	314	-1	-1	-1	-1	-1	139	-1	-1	-1	-1	-1	-1	-1	-1	288	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1
113	-1	-1	-1	14	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	218	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1
-1	113	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	132	-1	114	-1	-1	-1	-1	-1	-1	168	-1	-1	-1	-1	-1	-1	-1	0	-1	-1
80	-1	-1	-1	-1	-1	-1	78	-1	163	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	274	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1
-1	135	-1	-1	-1	-1	149	-1	-1	-1	15	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0

Figure 1.7: Base graph ($B1$) matrix for 5G-NR wireless communication standard that has 46 rows and 68 columns. It carries the maximum information bits of 8448 and also supports multiple standard code-rates from 1/3 to 8/9. [1, 2]

Table 1.1: Sizes of base graph matrices corresponding to various code-rates of LDPC code that is compliant to the specifications of 5G-NR wireless communication standard.

Code-rates	**Size of $B1$ Matrix**	**Size of $B2$ Matrix**
1/5	–	42×52
1/3	46×68	22×32
2/5	35×57	17×27
1/2	24×46	12×22
2/3	13×35	7×17
3/4	10×32	–
5/6	7×29	–
8/9	5×27	–

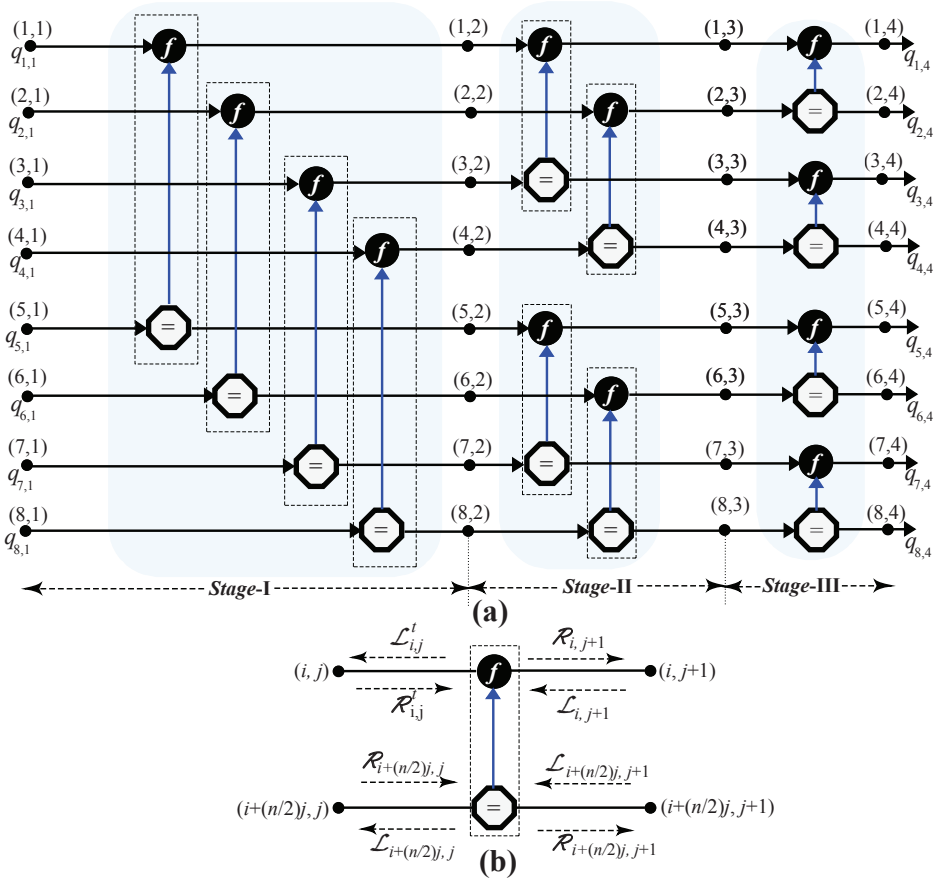


Figure 1.8: (a) Factor graph with 32 nodes, representing (n=8, k=4) polar code, and (b) schematic representation of a processing element.

parity bits (i.e. frozen bits) are transmitted over an unreliable channel. Such distribution of bits over reliable and unreliable channels is termed channel polarization [11]. Furthermore, n codeword bits can be generated by matrix multiplication of k information bits with the generator matrix $\mathcal{G}$. Here, $\mathcal{G}$ is m^{th} Kronecker power of $\mathcal{F}=\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}$ and $m=\log_2(n)$ that can be mathematically expressed as $\mathcal{G}=\mathcal{F}^{\otimes m}$. Polar codes can be graphically represented as a factor graph which plays a pivotal role in decoding of the polar codes, using the BP based decoding algorithm. Generally, (n, k) polar codes can be represented with $m-$stage factor graph that contains $(m+1)\cdot n$ nodes. Furthermore, factor graph shown in Fig. 1.8 (a), represents (n=8, k=4) polar code and it has three stages with $(3+1)\cdot 8=32$ nodes. Here, each processing element (PE) has been presented in Fig. 1.8 (b). Furthermore, the BP decoding algorithm is associated with $(i, j)-$index node where LLR messages are iteratively propagated from left-to-right ($\mathcal{L}^I_{i,j}$) and right-to-left ($\mathcal{R}^I_{i,j}$) among their adjacent index node, corresponding to the I^{th} iteration. The mathematical expressions for $\mathcal{L}_{i,j}$ and $\mathcal{R}_{i,j}$ messages are

$$\mathcal{L}^I_{2i,j} = \Upsilon(\mathcal{L}_{i,j+1}, \mathcal{L}_{i+n/2^j,j+1} + \mathcal{R}_{2i+1,j}); \tag{1.3}$$

$$\mathcal{L}^I_{2i+1,j} = \Upsilon(\mathcal{L}_{i,j+1}, \mathcal{R}_{2i,j}) + \mathcal{L}_{i+n/2^j,j+1}; \tag{1.4}$$

$$\mathcal{R}^I_{i,j+1} = \Upsilon(\mathcal{R}_{2i+1,j} + \mathcal{L}_{i+n/2^j,j+1}, \mathcal{R}_{2i,j}); \tag{1.5}$$

$$\mathcal{R}^I_{i+n/2^j,j+1} = \Upsilon(\mathcal{R}_{2i,j}, \mathcal{L}_{i,j+1}) + \mathcal{R}_{2i+1,j}; \tag{1.6}$$

where

$$\Upsilon(x, y) = sign(x)\cdot sign(y)\cdot min(|x|, |y|). \tag{1.7}$$

The adjacent nodes iteratively propagate LLR messages based on (1.3)−(1.6) and update LLRs on PEs. Therefore, such an iterative process continues, until all parity-check equations are not matched or a maximum number of iterations is not reached.

1.5 Channel Decoding Algorithms

Numerous decoding algorithms provide insights regarding various diverse advancements and developments to improve the channel decoding capabilities. Thus, it has become crucial for the channel decoding algorithms to enhance the performance of communication networks. Therefore, this analysis provides comprehensive insight into the challenges, and advancements, and suggests a crucial aspect for the decoding of error correction codes. Table 1.2 shows the comparisons of various LDPC decoding algorithms that has been reported in the literature. In LDPC decoding, the sum-product (SP) algorithm, also known as the belief-propagation (BP) algorithm propagates the message between CNs and VNs. It provides excellent bit-error-rate (BER) performance and can achieve Shannon limit error correction [12]. However, there is slow convergence and an increase in complexity with higher code-lengths [13]. Further, an improved SP algorithm [14] has lower computational-

Table 1.2: Comparisons for various LDPC decoding algorithms.

Algorithms	Performance	Hardware complexity	SNR (dB) for 10^{-5} BER
Sum product (SP)	Excellent	Very Complex	1.6
Improved SP	Excellent	Complex	1.65
Min-sum (MS)	Moderate	Low	1.9
Offset MS	Good	Low	1.7
Normalized MS	Good	Moderate	1.68
SAMS	Moderate	Moderate	1.95

complexity and also computes the Fourier transform faster than conventional SP algorithms. All the complex operations of the SP algorithm have been simplified by using min-sum (MS) approximation in MS algorithm [15]. However, the MS algorithm [16] reduces the complexity in the LDPC decoding process and sacrifices the BER performance compared to the SP algorithm [13]. Further, offset [13] and normalized [17] MS decoding algorithms have been introduced to optimize the performance by adding and scaling of offset value, respectively. This optimization provides improved error correction performance, especially at higher signal-to-noise ratios [13] and requires an additional parameter for tuning the optimal performance. Further, the second minimum approximation min-sum (SAMS)

algorithm [18] incurs lower complexity at the level of CN to VN message computations, at the cost of coding loss. Similarly, Yun et al. [17] suggested min-sum (MS) algorithm based LDPC decoder that delivers better hardware-compatibility and lower computational-complexity [13]. In addition, the scheduling technique for LDPC decoding is classified as layered and flooding [19, 20]. The layered scheduling achieves better coding performance with longer latency and moderate data-throughput [19], and the flooding scheduling delivers lower latency and higher data-throughput, at the cost of higher area occupation [20]. Each LDPC decoding algorithm has its own advantages and disadvantages, and the algorithm choice depends on the specific requirement of applications and trade-off between performance and complexity.

Table 1.3: Comparisons for various polar decoding algorithms.

Algorithms	**Performance**	**Throughput**	**Latency**	**Parallelism**	**Hardware Complexity**
BP	Moderate	Very High	Low	High	Low
SC	Good	Moderate	Moderate	Low	Moderate
SC list	Excellent	Moderate	Moderate	Low	High
SC-flip	Good	Moderate	Moderate	Low	High

On the other side, polar codes are another kind of error correcting code that has become more popular as a result of channel capacity abilities and efficient decoding algorithms. Polar codes are LBCs that can achieve the channel capacity for symmetric binary-input memoryless channels [11]. It is mainly adopted by the 5G-NR standard due to its lower computational-complexity and parallelism [1, 21]. Table 1.3 shows the comparisons of various polar decoding algorithms reported in the literature. Belief propagation (BP), successive cancellation (SC), SC list (SCL), and SC-flip algorithms [22, 23, 24, 25, 26, 27] are primarily used for the decoding of polar codes. Here, the BP-based polar decoding algorithm is inherently suitable for parallel computations and to achieve higher data-throughput [22]. On the other hand, the SC decoding algorithm delivers excellent coding performance and consumes longer latency [23]. The SCL decoding algorithm improves the coding performance at the price of higher hardware-requirement and energy-consumption

[25]. Further, the SCL decoding algorithm incorporates tree pruning techniques to reduce computational-complexity and offers balanced performance [28]. The fast shortening technique has been merged into the SC algorithm, named fast-SSC achieves good error correction performance and is more efficient for longer code-lengths and higher code-rates [27]. Its performance is further enhanced with the formulation of the SC flip decoding algorithm based on an alternative technique that relies on the multiple decoding sequences to identify correct and incorrect codewords [26]. Each polar decoding algorithm offers a different balance between error correction performance, computational-complexity, and memory requirements, making them suitable for various applications based on specific constraints and requirements.

1.6 VLSI Architecture of Channel Decoders

Design of LDPC decoder architecture has been classified based on the degree of parallelism viz. fully-parallel architecture that delivers higher throughput [29], partially-parallel architecture targets hardware/energy-efficiency [30], and serial architecture is suitable for lower throughput applications [31]. Comparisons of various channel decoder architecture based on different metrics has been represented in Table 1.4. Huge basegraph B matrix and higher code-lengths exacerbate hardware-efficiency and data-throughput of the LDPC decoder [1]. The routing technique proposed in [32] increases the implementation complexity when applied to higher code-lengths of the 5G-NR standard. Similarly, parallel routing network and high-speed architecture for LDPC decoder have been reported in [33] and [34], respectively. On the other side, the performance and routing-complexity of the LDPC decoder also depend on two scheduling techniques: layered and flooding [35], [36]. A hybrid-NMS algorithm based on layered scheduling, proposed by Zhao et al. [37], delivered a double convergence rate in comparison with the flooding schedule based decoding algorithm. Similarly, Zhang et al. [38] suggested a parallel layered decoding architecture based on a puncturing scheme that supports all code-rates of IEEE 802.16e (WiMAX) standard. On the

one side, the LDPC decoder based on the flooding schedule delivers degraded performance, higher-throughput, and lower-latency at the cost of huge switching-complexity [36], [39]. On the other side, a layered scheduling based LDPC decoder delivers excellent performance, medium-throughput, moderate routing-complexity, and longer-latency. However, this routing-complexity escalates while decoding the LDPC code with higher code-lengths and multiple code-rates, which exacerbates hardware-efficiency and data-throughput of LDPC decoder. Recently, layer merging methods for small code-length and high-throughput LDPC decoders of 5G-NR standard have been reported in [40] and [41], respectively.

Table 1.4: Comparisons for various LDPC/polar channel decoder architectures.

Architecture	Area	Power	Throughput	Latency	Hardware Complexity	Routing Complexity
Fully Parallel	High	High	High	Low	High	High
Partially Parallel	Moderate	Moderate	Moderate	High	Moderate	Moderate
Serial	Low	Low	Low	High	Low	Low
Layered	Moderate	Moderate	Low	Low	Moderate	High
Flooding	Moderate	Moderate	high	Low	Moderate	Moderate

On the other side, the polar decoder architecture that has been designed on the BP decoding algorithm provides high-parallelism and data-throughput at the cost of BER performance [22]. Similarly, SC and SC-List algorithms based polar decoder provide better decoding performance with higher-latency [23], [25]. Further, an area-saving technique has been proposed by Hashemi et al. [42] based on interpolation code construction and layered decoding technique. Mousavi et. al. presented another approach by using a partial sum network that efficiently estimates the list of codewords with more hardware resource usage [43]. A fast-SSC algorithm based polar decoder architecture [44] showcased higher data-throughput with moderate area-consumption.

1.7 Problem Definition

The ITU and 3GPP have advocated LDPC codes for data signaling whereas polar code has been specialized for control signaling in 5G-NR wireless communication systems. As discussed in section 1.5 and 1.6, there are various distinct LDPC and polar decoding algorithms and architectures have been reported in the literature. However, it is important to design such an LDPC and polar decoder that will fulfill all the high-end specifications for 5G-NR networks. Therefore, we are targeting to achieve all 5G-NR standard specifications like throughput $\approx$ 10–20 Gbps, latency $\leq$ 1 ms, etc. Therefore, some LDPC decoder architectures

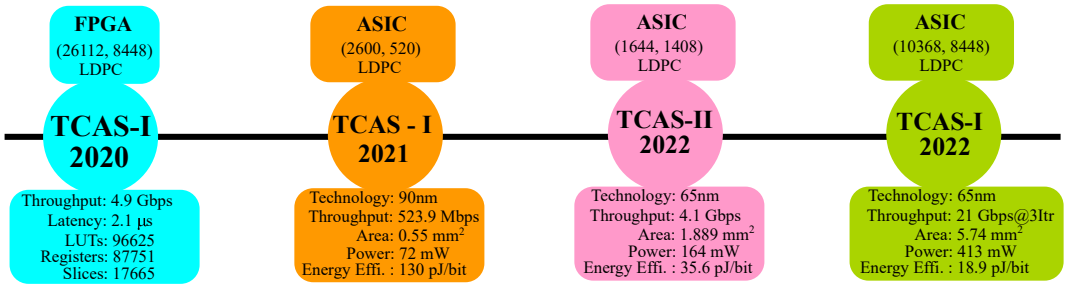


Figure 1.9: Illustration of various state-of-the-art LDPC decoder architectures with their corresponding specifications.

have been reported in the literature [5, 40, 45, 46]. All the specifications for these LDPC decoders have been shown in Fig. 1.9. To take these references, V.L. Petrovic et. al. proposed an LDPC decoder architecture that has lower hardware-consumption whereas it provides the data-throughput of 4.9 Gbps [46] which does not satisfy the specifications for 5G-NR eMBB applications. Similarly, a small 5G-NR code-length based LDPC decoder architecture has been reported by H. Cui et. al. [40] where the authors focused on better algorithmic performance. Further, (1644, 1408) QC-LDPC decoder architecture has been presented by S. Yun [45] in the literature, delivering the data-throughput of 4.1 Gbps. In addition, multi-mode LDPC decoder architecture is reported to support high-parallelism to achieve higher data-throughput [5], also supports memory alignment scheduling for CN and VN operations whereas this decoder architecture compromises with BER performance due to its lesser number of decoding iterations.

Similarly, there are various hardware implementations that are separately reported for LDPC and polar decoders where [5, 45, 47] are some of the recent works in the literature. These individual LDPC and polar decoder architectures swallow very high hardware resources and also consume more power. By consideration of these limitations for individual LDPC and polar decoder, it has been observed the LDPC and polar decoding algorithms have few identical computations such as min-sum approximation and addition operations, etc. The LDPC and polar decoding memory requirement for iterative processing of belief computations can be formulated and shared between LDPC and polar codes. Therefore, it has been analyzed that LDPC and polar decoding algorithm and architectures has a possibility to combine the algorithmic operations and operate on a new unified reconfigurable LDPC/polar technique that is used for the decoding of LDPC as well as polar codes. However, based on the contemporary demand to incorporate dual channel codes for evolving wireless-communication technologies like 5G-NR and 6G. It is high time to proliferate the development of efficient and unified hardware architectures for flexible channel decoders that support such requirements. In this context, one of the major research challenges is to design hardware-efficient decoder chip with unified reconfigurable LDPC/polar channel decoder architecture that meets high-end specifications of evolving technologies and flexibly supports more than one channel decoding scheme. Presently, there is a surge in the implementation of reconfigurable channel decoders for both LDPC and polar codes that are adopted by the 5G-NR standard. Fig. 1.10 clearly illustrates the

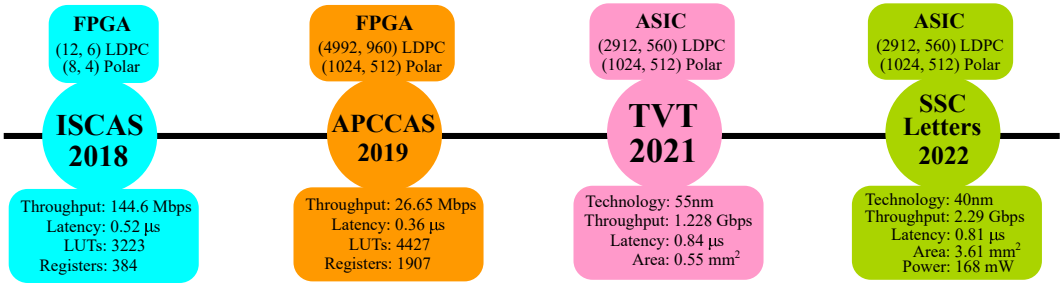


Figure 1.10: Illustration of various state-of-the-art unified reconfigurable LDPC/polar decoder architectures with their corresponding specifications.

state-of-the-art works for the unified channel decoder architectures with their specifications. With this notion, Bei-Sheng et al. recently reported energy-efficient dual-mode decoder for LDPC and polar codes based on the BP technique for 5G-NR standard [3]. Similarly, Shan et al. developed a pipelined architecture for a reconfigurable channel decoder for LDPC/polar code that supports eMBB applications of 5G-NR standard [4]. Additional hardware-efficient architectures of reconfigurable LDPC/polar channel decoders and their FPGA implementations are reported by Ningyuan et al. [30] and Ting et al. [48].

In this context, one of the major challenges to design a hardware-efficient and high data-throughput LDPC decoder architecture. We can achieve hardware-efficient architectures by using the iterative decomposition technique. However, it has been noticed that such decoder delivers moderate data-throughput and higher decoding-latency. On the other hand, it is challenging to design high-throughput channel decoder architecture, compliant with the high-end specifications of the 5G-NR wireless standards. Since, its achievable throughput is the ratio of code-length and latency, higher code-length and lower-latency are the most preferable factors for the design of high-throughput decoder architecture. Therefore, this work employs a high-parallelism methodology to achieve lower-latency and high data-throughput channel decoder architectures. However, such methodology surges the hardware consumption; nevertheless, the proposed channel decoder algorithms and architectures have lower computational-complexity and alleviate the hardware consumption. Hence, it results a new reconfigurable channel decoder architecture that is hardware-efficient and delivers higher data-throughput. Alternatively, our work exploits common operations between LDPC and polar decoding like min-sum approximation, memory sharing of intrinsic as well as extrinsic information, and message routing (data-flow), to propose a new hardware-efficient architecture of unified reconfigurable architecture of LDPC/polar decoder that surpasses the state-of-the-art results.

1.8 Summary and Contributions

This thesis comprehensively delivers various channel decoder architectures that is compliant to the specifications of 5G-NR wireless communication standard. Firstly, we proposed a hardware-efficient QC-LDPC decoder architecture based on the conventional min-sum LDPC decoding algorithm. An iterative decomposition technique has been incorporated into this architecture that affects an excessive number of clock-cycles to decode the QC-LDPC codes. Further, a new fully-parallel QC-LDPC decoder architecture has been suggested that delivers high data-throughput and lower decoding-latency with excessive consumption of hardware-resources. Subsequently, we proposed an algorithm based on the log-likelihood-ratio compound (LLRC) segregation technique that performs the CN and VN operation in matrix form. This parallel computation technique alleviates the routing-complexity for CN and VN operations. Furthermore, a novel QC-LDPC decoder architecture has been presented in this thesis that offers lower hardware-consumption due to the LLRC segregation technique. It is to be noted that the limitation of LLRC segregation based decoding algorithm is its high computational-complexity.

To overcome this limitation, a new simplified offset min-sum (SOMS) algorithm has been suggested that has lower computational-complexity and memory storage ability. A novel QC-LDPC decoder has been suggested based on the SOMS decoding algorithm. This decoder architecture alleviates the computational-complexity and also consumes lower hardware-resources. Further, a latest reconfigurable LDPC/polar decoding technique has been presented in this thesis that can be used to decode LDPC as well as polar codes. Afterward, we proposed a new unified LDPC/polar decoder architecture based on the reconfigurable LDPC/polar decoding technique. This suggested unified decoder architecture is complaint to mMTC and URLLC applications of 5G-NR wireless communication standards. The suggested LDPC/polar decoder architecture shares the memory which leads to lower hardware-efficiency. In this thesis, an ASIC fabrication of unified reconfigurable LDPC/polar channel decoder has been presented that has been designed in 110 nm-CMOS

technology node. Eventually, this thesis presents a detailed chip characterization process of reconfigurable channel decoder using the real-world test setup of 5G-NR wireless communication standard. A comprehensive list of contributions from this thesis are enumerated, as follows:

1. A partially-parallel hardware-efficient QC-LDPC decoder architecture for iterative message passing mechanism based on layered min-sum decoding algorithm for 5G-NR wireless communication standard has been first presented here. In addition, this proposed QC-LDPC decoder architecture has been hardware prototyped in a FPGA platform. Furthermore, BER versus SNR performance analysis of the min-sum decoding algorithm has been carried out that delivers an adequate coding performance corresponding to specific SNR for various 5G-NR standard code-rates.

2. Aforementioned decoder architecture delivers a lower data-throughput due to its higher decoding-latency. Therefore, a novel fully-parallel QC-LDPC decoder has been proposed for eMBB applications of 5G-NR wireless standard. This decoder has been designed based on an offset min-sum (OMS) decoding algorithm with layered scheduling. Subsequently, extensive performance analyses of our min-sum decoding algorithm have been carried out based on 5G-NR specifications. Furthermore, hardware implementation of the proposed LDPC decoder is performed on the FPGA platform and its results are compared with the reported works in the literature.

3. In addition, a novel decoding algorithm has been proposed based on the LLR compound (LLRC) segregation technique. This simplifies and allows faster accessibility of LLRs and mitigates the data-overcrowding issues while performing CN and VN operations. Further, comparative performance analyses of the proposed QC-LDPC decoding algorithm with the existing ones are presented in this thesis. As a result, it incurred lesser data-congestion, higher data-throughput and hardware-efficient QC-LDPC decoder architecture. The suggested QC-LDPC decoder has been hardware implemented on the FPGA platform and its implementation results are compared

with the state-of-the-art contributions.

4. Above suggested LLRC segregation based OMS algorithm has high computational-complexity and hardware-efficiency. Consequently, the earlier proposed QC-LDPC decoding algorithm has been simplified to lower computational-complexity, and this new algorithm is termed as simplified OMS (SOMS) algorithm. Consecutively, comprehensive FER performance analyses of the proposed SOMS decoding algorithm are presented in this thesis. Corresponding to the proposed SOMS algorithm, a new QC-LDPC decoder architecture has been proposed that alleviates hardware-efficiency and has achieved higher data-throughput than the proposed QC-LDPC decoder. Eventually, real-world FPGA-prototype test setup for functional verification of the proposed QC-LDPC decoder has been carried out in this thesis.

5. Finally, a new unified reconfigurable channel decoder architecture for the decoding of LDPC and polar codes has been proposed in this thesis. This reconfigurable decoder is designed based on SOMS and belief propagation decoding algorithm for LDPC and polar codes, respectively. Here, we suggested a unified LDPC/polar decoding technique that shares the memory storage and computational operations of LDPC and polar decoding algorithm. Subsequently, a proposed unified LDPC/polar decoder has been designed based on the suggested reconfigurable technique. At last, we fabricated an ASIC for this reconfigurable channel decoder in the united microelectronics corporation (UMC) 110 nm-CMOS technology node. This chip has been characterized and functionally validated using the real-world test setup. All the measured results of our ASIC chip have been analyzed and compared with state-of-the-art implementations from the literature.

Chapter 2

Hardware-Efficient and High-Throughput QC-LDPC Decoder Architectures

---

2.1 Introduction

It is notably in the realm of high-speed communication standards that LDPC codes have been used as a foundational component in advanced error-correcting-codes. The BP decoding algorithm [49] for LDPC codes improves the BER performance by using an iterative message-passing technique. This decoding algorithm facilitates excellent error correction and also encourages reliable communications over the adverse channel conditions. The hardware implementation of BP algorithm is a cumbersome task due to its highly-complex computational units. The computing unit and memory-intensive operations in the hardware implementations of the BP algorithm are extremely complex, resulting in excessive resource usage. Therefore, the hardware implementation of BP algorithm is a challenging endeavor. A unique MS decoding technique has been developed to optimize the hardware

complexity of the BP algorithm [49]. This approach streamlines and performs the message transmission operations at the CNs and VNs. Although the MS decoding algorithm has lesser computational-complexity than the BP algorithm, it is obtained at the cost of degraded BER performance of the MS algorithm [15]. Therefore, normalized and offset approximations have been introduced in the MS decoding algorithm to improve the BER performance with lesser computational-complexity and hardware-utilization [40, 50]. The hardware implementation of the MS decoding algorithm with offset approximation is deemed acceptable due to its reduced hardware-complexity.

As discussed earlier, an offset min-sum (OMS) decoding algorithm-based LDPC decoder architecture consumes fewer hardware resources. However, 5G-NR wireless communication networks demand high-end specifications, such as higher data throughput and lower decoding latency, to meet standardization requirements. Consequently, designing an LDPC decoder architecture that balances hardware efficiency with high data throughput and low decoding latency is a challenging task. However, our work is focused on designing hardware-efficient and high-throughput QC-LDPC decoders that are compliant with the enhanced mobile broadband (eMBB) specifications of 5G NR technology.

In this chapter, we proposed two QC-LDPC decoders: 1) hardware-efficient and 2) high-throughput, based on conventional OMS decoding algorithm that is compliant to 5G-NR wireless communication applications. Firstly, the partially-parallel hardware-efficient LDPC decoder architecture is designed based on the conventional OMS with layered scheduling. We incorporate an iterative decomposition technique to achieve the hardware-efficiency. The CN & VN operations are sequentially processed in this decoder that leads to lower hardware-consumption and high decoding-latency. Hence, this partially-parallel QC-LDPC decoder architecture has moderate data-throughput due to its higher decoding-latency. Further, a new QC-LDPC decoder architecture has been presented in this chapter that delivers high data-throughput with the aid of high-parallelism. Here, a replication technique has been applied to CN & VN operation unit in this fully-parallel QC-LDPC decoder architecture. Hence, hardware-efficient as well as high data-throughput QC-LDPC

decoder architectures have been presented in this chapter. The key contributions of this chapter as follows:

- Firstly, a conventional layered scheduling based OMS decoding algorithm for LDPC codes is presented in this chapter. Consequently, a BER performance analysis with non-scaling and scaling factors in every iteration has been presented here that is compliant to the 5G-NR standard. We also analyzed the effects for various number of decoding-iterations and quantization-bits on BER performance.

- In addition, a new partially-parallel QC-LDPC decoder architecture for an iterative message-passing mechanism based on a layered OMS decoding algorithm has been proposed that is compliant to the 5G-NR wireless communication standard. Subsequently, this QC-LDPC decoder architecture has been placed and routed on the FPGA platform. Furthermore, we performed the static timing analysis and compared our hardware implementation results with the state-of-the-art works.

- A fully-parallel QC-LDPC decoder architecture has been presented that is designed based on the conventional OMS decoding algorithm. This QC-LDPC decoder is designed to achieve high data-throughput and is compliant to the 5G-NR wireless communication standard. Similarly, this decoder architecture is also FPGA prototyped on the Xilinx ultrascale+ FPGA platform. Finally, we compared the hardware-implemented results with reported works in the literature.

2.2 System Level Overview for 5G New Radio

In the 5G-NR physical layer, various user equipments (UEs) communicate with the base stations (gNBs), as shown in Fig. 2.1. All nearby UEs initially generate a scheduling request (SR) signal to access the uplink resources for data transmission through the physical random access channel (PRACH). Once gNBs receive the SR signal, a decision for data transmission is made, based on the channel quality measurement (CQM) [21]. After decoding the channel

information from the uplink (UL) grant, UEs transmit the UL data according to specified parameters in uplink control information (UCI). On the other side, gNBs transmit the downlink (DL) data information through the physical downlink shared channel (PDSCH) by using specified parameters of downlink control information (DCI), and beam-forming technique [21], as presented in Fig. 2.1. Here, the communication among UEs and gNBs take place based on periodically generated positive acknowledgment (ACK) and negative acknowledgment (NACK) UCI signals.

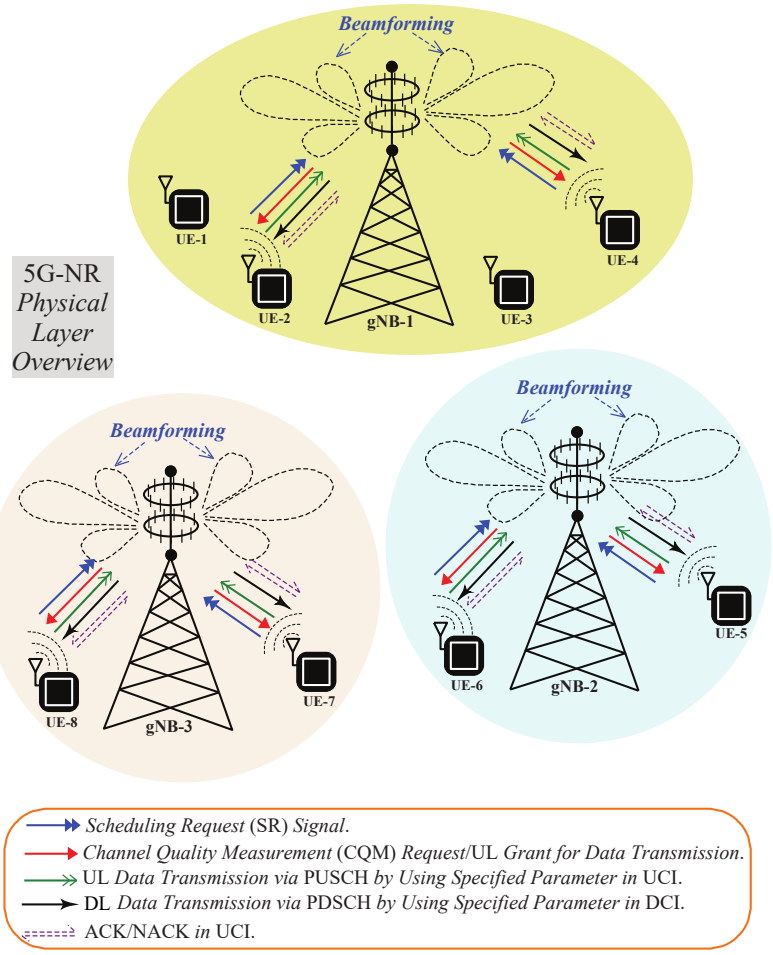


Figure 2.1: System-level overview of 5G-NR physical layer that shows data transmission between user equipments (UEs) and base stations (gNBs).

Overall system level architecture of the 5G-NR wireless communication network has been schematically presented in Fig. 2.2. To begin with, transmit bits from transport and

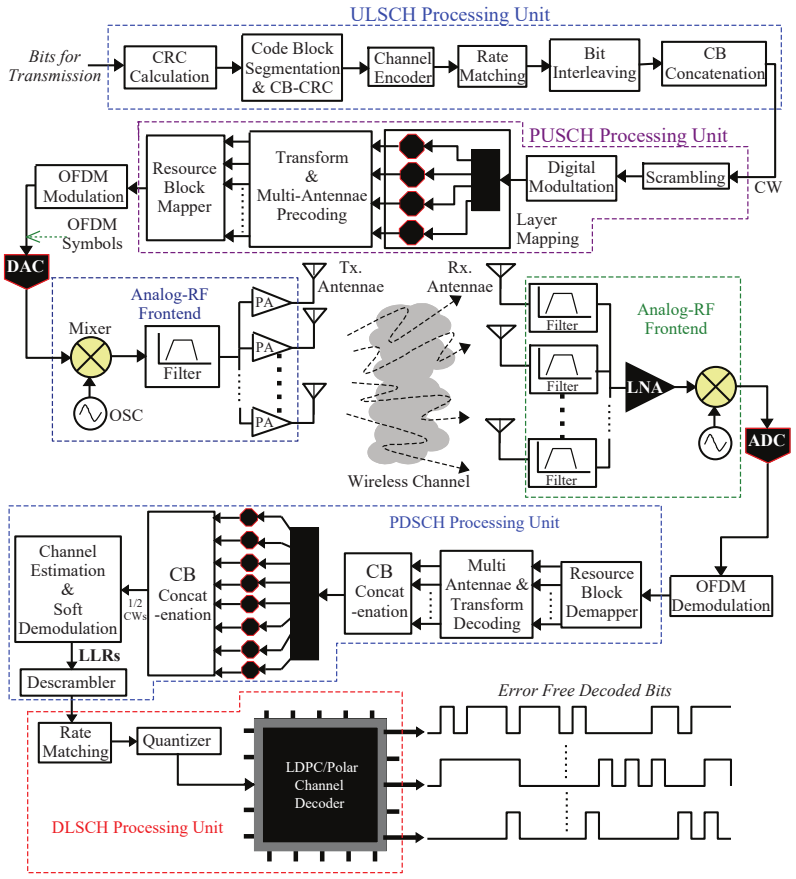


Figure 2.2: High-level architecture of physical layer for 5G-NR wireless systems depicting the roles of channel encoder and decoder at transmitter and receiver sides, respectively.

data-link layers are passed through the uplink shared channel (ULSCH) processing unit that generates a codeword. It comprises cyclic redundancy check (CRC) and parity bits that are generated by the code-block segmentation and the channel encoder (i.e. LDPC or polar encoder for 5G-NR standard) units, respectively [1]. Such parity bits aid in mitigating the adverse distortion on the transmitted signals across wireless channels due to external noise, multi-path fading, and modulation interferences. Further, these encoded bits are processed by the rate matching unit for puncturing, in order to support specific code-rate that are compliant to 5G-NR standard [1], as shown in Fig. 2.2. This unit enhances channel bandwidth efficiency and flexibility of the communication network without increasing the complexity. Now, the bit interleaver disperses the sequence of the

bit-stream for transmission to alleviate the burst error. Thereafter, the generated codeword is transmitted to the physical uplink shared channel (PUSCH) processing unit where the ULSCH-generated codeword is scrambled to prevent the unauthorized access of data, as shown in Fig. 2.2. It also shows that such scrambled codeword are digitally-modulated via the QPSK/QAM scheme and are mapped into 1 to 4 layers, which are processed by the transform pre-coding for the UL data transmission with lesser peak-to-average power ratio (PAPR). These layers are pre-coded to one or multiple antenna ports by the multiplication of a multi-antenna matrix with pre-coder matrix [1]. This process achieves beam-forming and spatial multiplexing for the line-of-sight data transmission between UEs and gNBs. Therefore, all these PUSCH symbols are non-interleaved/interleaved mapped with the physical resource blocks for the OFDM modulation, as illustrated in Fig. 2.2. Here, non-interleaved mapping is the direct mapping of symbols to the resource blocks. On the other side, interleaved mapping provides frequency diversity by distributing the resource blocks over the entire bandwidth [1, 21]. Consecutively, all OFDM symbols are processed by the analog radio-frequency (RF) frontend and transmitted via massive multiple-input multiple-output (MIMO) scheme over the PRACH, as presented in Fig. 2.2.

At the receiver side, MIMO antennas capture continuous signals from the PRACH where a band-selection filter selects the specified frequency band that is compliant to 5G-NR standard [21]. Now, these filtered high-frequency signals are passed through the low noise amplifier (LNA) to suppress noise and restore the signal strength. Further, such continuous-time analog signals are digitized with the aid of an analog-to-digital converter (ADC), whose output is OFDM-demodulated and passed to the physical downlink shared channel (PDSCH) processing unit, as shown in Fig. 2.2. Here, OFDM demodulated symbols are de-framed into various radio-frame symbols, after passing through the resource block de-mapper [1]. Subsequently, the multi-antenna decoder selects the radio-frame symbols from these OFDM demodulated radio-frame symbols with the aid of demodulation reference signals (DMRSs). Furthermore, such decoded symbols are mapped into 1 to 8 layers (4 layers for one codeword and 8 layers for two codewords), as illustrated in Fig. 2.2. After

layer mapping, the channel estimation process is carried out that improves the coding performance and enhances the spectral-efficiency as well as reliability of the received OFDM symbols [1]. Thereafter, these symbols are soft-demodulated to generate the LLRs, corresponding to all the transmitted bits. Further, all these received LLRs are descrambled into 1 or 2 codeword(s), and are processed by the inverse rate matcher. Here, initial $2\times z$ bits (where z refers to expansion factor) are punctured and assigned with zero values to match the code-rate, corresponding to the transmitted 5G-NR code-rate. Such de-punctured LLRs are transferred to bit-quantized and such quantized LLRs are fed to LDPC/polar channel-decoder which processes these LLRs to generate the error-free decoded bits. Eventually, the CRC of decoded bits is carried out and if it is the same as the transmitted CRC bits then these decoded bits are considered as the true transmitted bits.

2.3 QC-LDPC Layered Offset Min-Sum Decoding Algorithm

QC-LDPC code has been standardized for 5G-NR enhanced-mobile-broadband (eMBB) communication system [1]. The 5G-NR parity check matrix (H) for QC-LDPC code is constructed by using base graph matrix (B) (as presented in Fig. 1.7) with the expansion factor (z) of 384. The decoding of LDPC codes is based on the BP technique which is iteratively performed by message-passing between CNs and VNs until the valid codeword is found or the maximum number of iterations is achieved. The computations carried out by these CNs and VNs operations are based on the OMS algorithm which is widely adopted, as it is implementation-friendly in comparison to its counterpart SP algorithm. The OMS iterative decoding with layered scheduling has been presented in Algorithm 1. In the initialization process (from lines 3-7), quantized channel LLRs represented as L are received for all n bits of codeword.

Subsequently, the iterative message exchange process begins between CNs and VNs via a layer-by-layer process of B matrix. In such an iterative process, VNs initiate by sending messages to all CNs in the first layer of B, where CNs will receive their corresponding

Algorithm 1 Conventional OMS-layered decoding algorithm for proposed QC-LDPC decoder architecture.

1: **Inputs:** $L = \{l_1, l_2, \cdots l_n\}$, σ^2; **Output:** $\widehat{X} = \{\chi_1, \chi_2, \cdots \chi_n\}$;
2: **Initialization:**
3: **for** $n \in L$
4: $\quad \mathbb{A}_n = log\left(\frac{P(l_n|\chi_n = 0)}{P(l_n|\chi_n = 1)}\right) = \frac{2.L}{\sigma^2}$
5: $\quad E = \{e_1, e_2, \cdots, e_n\} = 0$
6: $\quad V_{p,n} = log\left(\frac{P(l_n|\chi_n = 0)}{P(l_n|\chi_n = 1)}\right), p \in H(n)$
7: **end for**
8: **CN and VN Message Exchange Iteratively:**
9: **for** $I = 1$ to i_{max}
10: $\quad$ **for** $r = n_l$
11: $\quad\quad$ **for** $m \in B(r)$
12: $\quad\quad\quad$ **for** $n \in H(m)$
13: $\quad\quad\quad\quad V_{m,n} = \mathbb{A}_n$ - $E^{I-1}_{m,n}$ ▷ **VN Update**
14: $\quad\quad\quad\quad \alpha_{m,n} = \min_{n' \epsilon H(m) \backslash n} |x_{m,n'}| - offset$
15: $\quad\quad\quad\quad \beta_{m,n} = \begin{cases} \alpha_{m,n} & \alpha_{m,n} \geq 0 \\ 0 & otherwise \end{cases}$
16: $\quad\quad\quad\quad \tilde{C}_{m,n} = \prod_{n' \epsilon H(m) \backslash n} sign(x_{m,n'}) \cdot \beta_{m,n}$ ▷ **CN Update**
17: $\quad\quad\quad\quad \tilde{\mathbb{A}} = V_{m,n} + \tilde{C}_{m,n}$ ▷ **Updated LLR**
18: $\quad\quad\quad$ **end for**
19: $\quad\quad$ **end for**
20: $\quad$ **end for**
21: **Hard Decision for all n:**
22: $\quad \widehat{X}^I_n = \begin{cases} 0 & \chi^{(i)}_n \geq 0 \\ 1 & otherwise \end{cases}$
23: $\quad$ **if** $\widehat{X} \cdot H^\tau = 0$ **then break**;
24: **end for**

messages and apply MS approximation to achieve the updated belief from the first layer of B matrix, as shown in Algorithm 1 from lines 9-20. Such updated beliefs are propagated through the second layer of the B matrix and achieve new beliefs for the next decoding iterations. This process continues until the last layer of B that produces the output belief. Finally, its hard decision values are obtained that are expressed as $\widehat{X}$ vector in line 22 of Algorithm 1. Such hard decision checks the syndrome $\widehat{X} \cdot H^\tau$ and if it is zero then early termination is achieved, as shown in line 23 of Algorithm 1 and this process completes first iteration of this min-sum LDPC decoding algorithm. Such aforementioned iterative process

presented between lines 9-20 of Algorithm 1 continues until the syndrome $\widehat{X} \cdot H^T$ is zero or the maximum numbers of decoding iteration are exhausted.

2.4 BER Performance Analysis

We present a comprehensive BER performance analysis of the conventional QC-LDPC decoding algorithm (from Algorithm 1) compliant to the 5G-NR wireless communication standard. Comparative performance analyses of conventional and scaled LDPC decoding algorithms are also included in this section. Fig. 2.3 shows the BER versus E_b/N_0

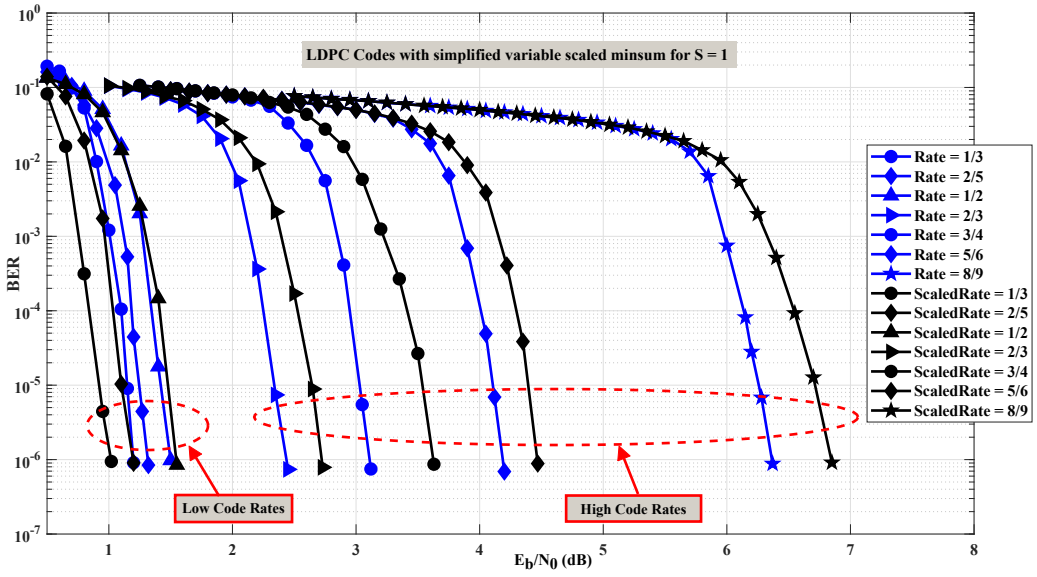


Figure 2.3: Comparative BER performance analyses of QC-LDPC layered min-sum decoding algorithm for decoding (n, k) = (26112, 8448) of QC-LDPC codes with various standardized code-rates that are compliant to 5G-NR wireless communication standard.

plots of 5G-NR compliant LDPC decoding algorithm which is extensively simulated under additive-white Gaussian-Noise (AWGN) channel environment where the information is BPSK modulated and demodulated for all the standard code-rates of 5G-NR standard. It clearly shows two categories of plots for different code-rates: conventional decoding performance without scaling and hardware-implemented performance with scaling, where LLRs

are scaled by a factor of 1-2$^{-\lceil I/S \rceil}$ in every iteration [51]. Such scaling factor exponentially increases with decoding iterations (I) where its final value is 1 and therefore, the scaling factor is approximated by a constant horizontal step (S) [51].

Table 2.1: Net coding gain for the decoding of 5G-NR LDPC codes by using conventional and scaled OMS decoding algorithm to attain 10^{-6} BER.

Code-rate	Conventional OMS	Scaled OMS	Net Coding gain
1/3	1.0	0.98	0.02
2/5	1.32	1.2	0.12
1/2	1.5	1.7	0.20
2/3	2.45	2.73	0.28
3/4	3.12	3.63	0.51
5/6	4.2	4.47	0.27
8/9	6.37	6.85	0.48

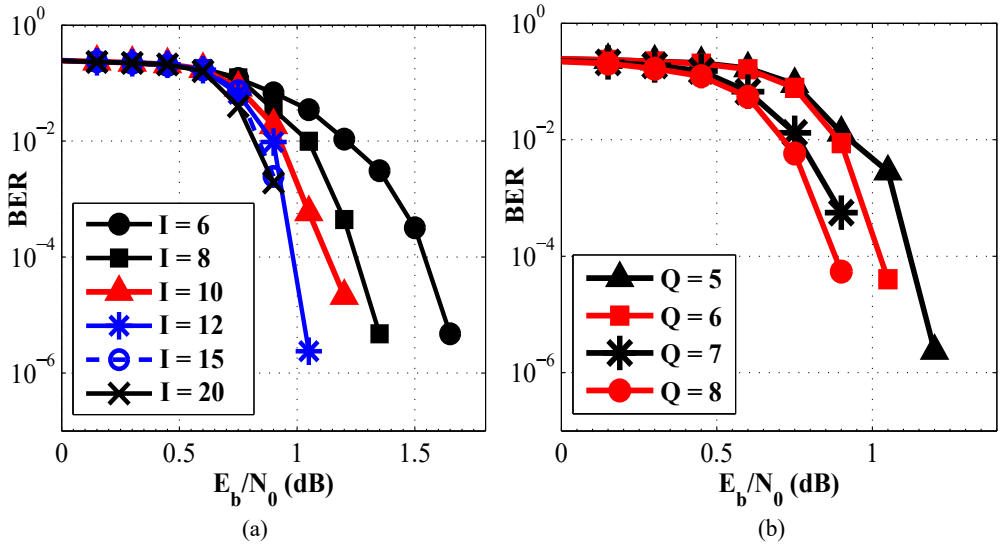


Figure 2.4: BER performance analyses of the offset min-sum QC-LDPC decoding algorithm for various (a) decoding iterations (I) and (b) bits quantization (Q).

For the 5G-NR standard, there are two base-graph (B) matrices: $B1$ and $B2$. Our simulations are based on the $B1$ matrix for code-length (n) of 26112 bits and information length (k) of 8448 bits. Table 2.1 shows the net coding gain achieved by conventinal and iterative

scaled OMS decoding algorithm to attain the 10^{-6} BER. Input LLRs fed to the decoding algorithm are quantized in 6-bit fixed-point format for simulation and it delivers a BER of 10^{-6} at 1.2 dB of E_b/N_0, for a code rate of 1/3, that has coding loss of 0.02 dB in comparison with the conventional decoding algorithm at same code rate, as shown in Fig. 2.3. It shows the BER performance simulation for different 5G-NR standardized code-rates of 1/3, 2/5, 1/2, 2/3, 3/4, 5/6, and 8/9. It can be observed from Fig. 2.3 and Table 2.1 that the fixed-point BER-performance plots that are scaled in every iteration deliver better performance for lower code-rates like 1/3 and 2/5. However, this performance degrades for higher code-rates like 3/4, 5/6, and further.

The BER performance analyses of the comprehensive QC-LDPC decoding algorithm have been carried out in AWGN channel environment with 26112 encoded bits generated using $B1$ matrix of 5G-NR (shown in Fig. 1.7) are transmitted and decoded for a code-rate of 1/3. We performed a comprehensive analysis for various decoding-iterations (ranging from 6−20) with 7-bits fixed quantization (Q) and 10 decoding iterations, as shown in Fig. 2.4 (a). It is to be noted that the BER plot converges with the increment of decoding-iterations. We also analyzed the BER performance analysis for various quantization bits (Q) at fixed iterations as shown in Fig. 2.4 (b). As the quantization bits increase, the BER performance is improved in comparison to lower quantization bits. It can be observed that the LDPC decoding with Q = 7 bits and 10 decoding iterations deliver a BER of 10^{-6} at 1 dB. There is definitely an improvement in performance with the increasing values of quantization-bits (Q) and iterations (I). However, it incurs huge hardware consumption and longer decoding-latency with high quantization bits (Q) and decoding-iterations (I), respectively.

2.5 Hardware-Efficient QC-LDPC Decoder Architecture

This section starts with a top-level architectural description of a layered-scheduling based QC-LDPC decoder. A partially-parallel decoder architecture has been presented that is complaint to 5G-NR wireless communication standard. It has been designed based on the

iterative decomposition technique that leads to hardware-efficiency. Subsequently, the VLSI architecture of various sub-modules of this hardware-efficient QC-LDPC decoder are also carried out.

2.5.1 Overall Decoder Architecture

The suggested architecture of the QC-LDPC decoder comprises of three basic modules: various memory blocks, CN & VN combined processing unit (CVCPU), and a network of multiplexers, as shown in Fig. 2.5. At the input side of this decoder, 8 bits of quantized LLRs are received and subsequently stored in the initial memory bank (IMB) which is a stack of 68 initial memory units (IMUs). Here, each IMU has a depth of 384 memory locations with 8-bits of word-length. Thus, IMB has 68×384 = 26112 memory locations which store 26112 received LLRs of 8-bits each. It is to be noted that 26112 bits is the standardized frame size of 5G-NR wireless standard and the corresponding LLRs from soft demodulator are fed to our decoder. In the real-world scenario, these received LLRs are fed sequentially where single LLR is stored in a memory location of IMB in every clock cycle and this consumes 26112 clock cycles to store all LLRs.

These LLRs are read from all 68 IMUs and their locations are based on B matrix elements that are stored in the base-matrix ROM, as shown in Fig. 2.5, and thereby, IMB reads out 68 LLRs in every clock cycle. These 68 LLRs (that comprises 68×8 = 544 bits) are passed through an iterative multiplexer (IMUX) that routes one of its LLR inputs (i.e. *Inp0*) for the first iteration and routes another LLR input (i.e. *Inp1*) via register for remaining iterations. Similarly, SUB-MUX passes *Inp0* and *Inp1* for the first and rest of the iterations,respectively. Outputs from IMUX and SUBMUX are fed as inputs to CVCPU which performs the subtraction, MS approximation, and addition for selected LLRs (where non -1's entries exist in a single layer of B matrix). A detailed discussion on the design of VLSI architecture for CVCPU has been presented in the next sub-section.

As shown in Fig. 2.5, CVCPU generates updated LLRs (*out_ini_mem*) for the next layer of B matrix and intermediate LLRs (*out_inter_mem*) which are stored in intermediate memory

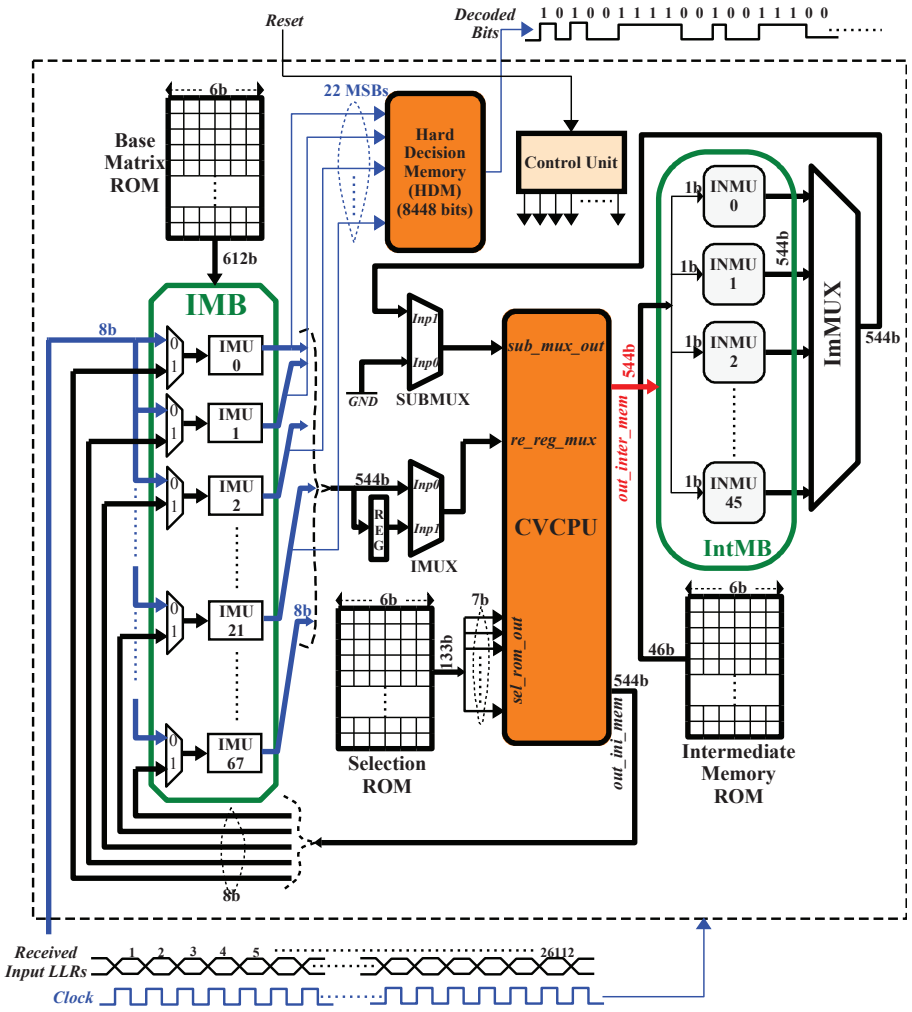


Figure 2.5: Overall VLSI-architecture of partially-parallel QC-LDPC decoder that is compliant to 5G-NR wireless communication standard.

bank (IntMB). All these 68 8-bits updated LLRs are fed back and overwritten in all 68 IMUs sequentially in single clock cycle. Hence, IMB requires 384 clock cycles to get updated with all generated LLRs for single layer of the B matrix. In the same clock cycle, 544-bits intermediate LLRs (68 LLRs where each LLR of 8-bits) are stored in IntMB. This IntMB comprises 46 intermediate memory units (INMUs) where each of these has 544-bits word 384 memory locations. Every INMU stores the intermediate LLRs corresponding to single layer of B matrix. As soon as the IMB is updated with the generated LLR of single layer, IntMB simultaneously stores the intermediate LLRs (which are used for next iterations) in

INMU in the same clock cycle. Both IMB and IntMB need 384 clock cycles to update all 68 IMUs and single INMU, respectively, for processing single layer of the B matrix. Such updated values from IMB are used for the next layer of B matrix. Subsequently, 68 LLRs are again read from the memory location corresponding to the next-layer elements of the B matrix. This layer also consumes 384 clock cycles for storing updated LLRs to IMB, and intermediate LLRs to the next INMU of IntMB.

Aforementioned process of LLR updation in IMB and IntMB continues till 46^{th} layer of B. At the end of 46^{th} layer processing, all INMUs of IMMB are filled with their corresponding B matrix layers and this marks the completion of the first iteration of the proposed QC-LDPC decoder, consuming 384×46 = 17664 clock cycles. Thereafter, the next decoding iteration starts with the selection of *Inp1* input of IMUX in which 68 LLRs pass through the registers, as shown in Fig. 2.5. Simultaneously, *Inp1* input of SUBMUX is selected that is read-out intermediate LLRs from IntMB via an intermediate-memory multiplexer (ImMUX) corresponding to specific B layer. For example: if IMB sends the updated LLRs corresponding to 9^{th} layer of B then ImMUX routes the data corresponding to the 9^{th} INMU of IntMB. This ImMUX passes the intermediate LLRs for 384 clock cycles to process single layer. Such iterative process is repeated for 10 iterations and this incurs a latency of 17664×10 = 176640 clock cycles.

Thus, our LDPC decoder has been designed to support 10 iterations and it stores all the updated LLRs in IMB thereafter. The hard decision process begins in-parallel fashion for the first 22 IMUs only; since from received LLRs, the first 22×z represents the information bits and the rest 46×z corresponds to parity bits (z = 384 is the expansion factor). As shown in Fig. 2.5, most-significant-bits (MSBs) of the first 22 IMUs are buffered in hard decision memory (HDM) which takes 384 clock cycles to store all 8448 decoded bits (i.e. 22×384 bits). Eventually, they are fetched sequentially in every clock cycle and hence the latency of our decoder is 185088 clock cycles (i.e. 176640 + 8448 cycles).

2.5.2 CN & VN Combined Processing Unit

The section proposed a new CVCPU architecture that is designed to integrate multiple decoding operations within a single module to simplify steering logic and streamline the pipeline process. The CVCPU is fed with 68 received/updated (via IMUX) and 68 intermediate (via SUBMUX) LLRs read-out from IMB and IntMB, respectively, at every clock cycle for their corresponding *B* matrix layer and decoding iteration, as shown in Fig. 2.5. The suggested VLSI architecture of CVCPU has been presented in Fig. 2.6 where the two sets of 68 LLRs (where each LLR has a bit-width of 8 bits) are routed through two different multiplexer networks (MUX68by19) where each network comprises of 19 68:1 multiplexers. The 7-bits selection lines for these multiplexers are generated by selection ROM (i.e. 7×19 = 133 bits), as discussed earlier in Fig. 2.5. Both MUX68by19 networks select only 19 LLRs each, corresponding to the values of select lines from Selection ROM. These LLRs are fed to intermediate max multiplexer (IMX) and intermediate min multiplexer (IMM) networks, as shown in Fig. 2.6, for selecting LLRs corresponding to non -1's elements in different *B* layers.

Here, IMX unit is the network of 19 parallel 2:1 multiplexers where one of the inputs is selected LLR and the other one is the decimal number '31'. Similarly, selected 19 intermediate LLRs from the second Mux68by19 network are fed to IMM network (comprising of 19 parallel 2:1 multiplexers) where intermediate LLR and '0' decimal value are the multiplexer inputs. These IMX and IMM select inputs based on the number of non -1's elements in the *B* matrix. For eg., layer 1 of the *B* matrix has 19 non -1's elements and thereby, all 19 parallel multiplexers route these 19 LLRs to the subtractor unit. However, for layer 6 of the *B* matrix, it has 8 non -1's elements so the first 8 multiplexers send LLRs, and the remaining 11 multiplexers route the 31 LLRs (via IMX) and 0 decimal value (via IMM) to subtractor network. It is a parallel network of 19 8-bits 2-input subtractors for subtracting intermediate LLRs with updated LLRs. Its output is limited to 6-bits quantization (via simple limiter unit) which fixes -32 and +31 as the lowest and highest values of LLRs, respectively.

Now, these limited LLRs are fed to the min-sum approximation unit (MSAU) for min-

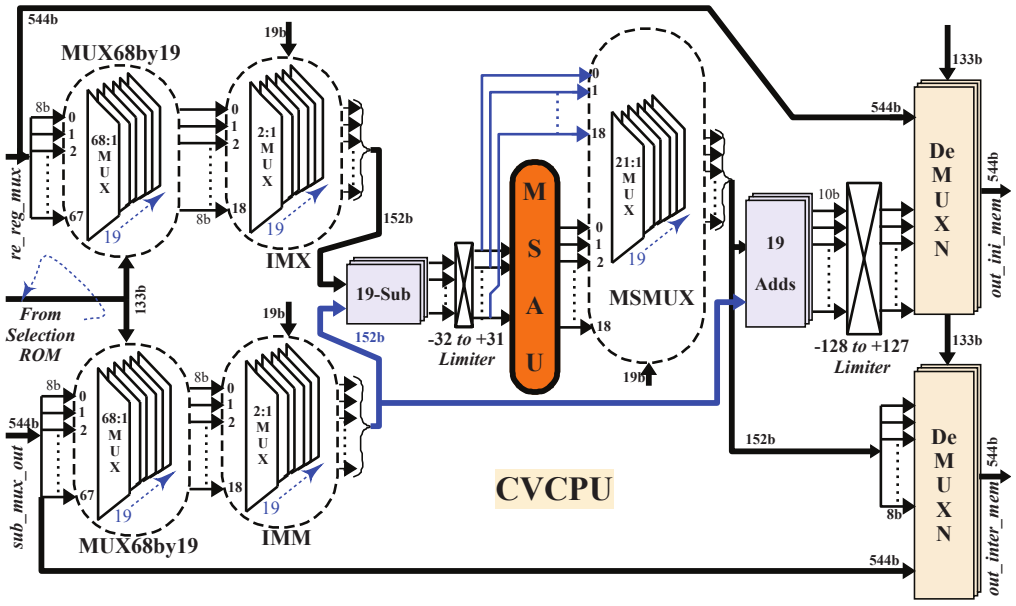


Figure 2.6: Internal VLSI micro-architecture of CN & VN combined processing unit (CVCPU) that is an integral part of overall QC-LDPC decoder architecture.

sum approximation and min-sum selection multiplexer (MSMUX) network for the selection of LLRs based on the *B* layer. This MSMUX network is a parallel stack of 19 2:1 multiplexers whose one of the inputs is the output from MSAU and another input is the output from min-sum limiter, as shown in Fig. 2.6. The 19-bits of selection input for the MSMUX network (1-bit for each 2:1 multiplexer of MSMUX) are generated based on the non -1's element in *B* matrix layers. Hence, the MSMUX network passes selected 19 LLRs (19×8 = 152 bits) and is fed as one of the inputs to the adder network (ADN). Another 152-bits input to ADN is the selected output from the second IMM network. Such ADN is a collection of 19 parallel 8-bits adders that add updated min-sum and intermediate LLRs. Subsequently, the sum is limited to -128 to +127 via quantization unit, as shown in Fig. 2.6. At the output side of CVCPU, there are two de-multiplexer networks (DeMUXNs) comprising of 19 de-multiplexers for replacing the previous LLRs with the updated LLRs based on the selection inputs generation from selection ROM, as discussed earlier in Fig. 2.5. One of the DeMUXNs updates the LLRs in *re_reg_mux* input and passes updated LLRs (*out_ini_mem*) to IntMB for the next layer processing of *B* matrix. Another DeMUXN updates the intermediate LLRs in *sub_mux_out*

input and sends new intermediate LLRs (*out_inter_mem*) to their corresponding INMU of IntMB, as shown in Fig. 2.5 and 2.6.

2.5.3 Min-Sum Approximation Unit

It is fed with 19 LLRs from the '-32 to +31 Limiter' which are in 2's complement format and thereby, these LLRs are converted to 8-bits signed-magnitude (SM) format using two's-complement to sign-magnitude (TCSM) units, as shown in Fig. 2.7. Here, MSBs of 19 SM-LLRs are fed to the sign unit (SU) and the rest 7-bits of these 19 SM-LLRs are processed by magnitude computation unit (MCU). In SU, all input MSBs are XORed and this multiplies all signed bits and generates a product bit. This bit is further XORed with every 19 MSBs of LLRs and produces 19 sign bits corresponding to the MSBs of LLRs, as shown in Fig. 2.7. Finally, SU delivers 19 signs for the min-sum approximation. In MCU, two minimum magnitudes (represented as *min1* and *min2*) are first selected among 19 magnitudes of SM-LLRs. Thereafter, an offset of decimal value 2 (i.e. 7'd2) is subtracted with *min1* as well as *min2*.

These subtracted values must be non-negative; however, if this value is less than zero then both *min1* and *min2* are assigned with zero by using 2:1 multiplexers at the subtractor outputs. Now, all the 19 magnitudes are replaced by *min1*, except the *min1* magnitude; this magnitude of *min1* is replaced by *min2*. Thus, the min-sum approximation of these magnitudes is performed by 19 parallel equalizers and 2:1 multiplexers in MCU, as shown in Fig. 2.7. Therefore, MCU delivers 19 min-sum approximated magnitudes. In addition, the sign bit (1-bit) and their corresponding magnitude (7-bits) are concatenated to obtain all 19 8-bits LLRs. Eventually, these SM-LLRs are converted to 2's complement format by signed-magnitude to two's-complement (SMTC) units.

2.5.4 Hardware Implementation and Comparisons

The proposed QC-LDPC decoder architecture has been hardware level coded using Verilog hardware-descriptive-language (HDL) and functionally verified using the Vivado design

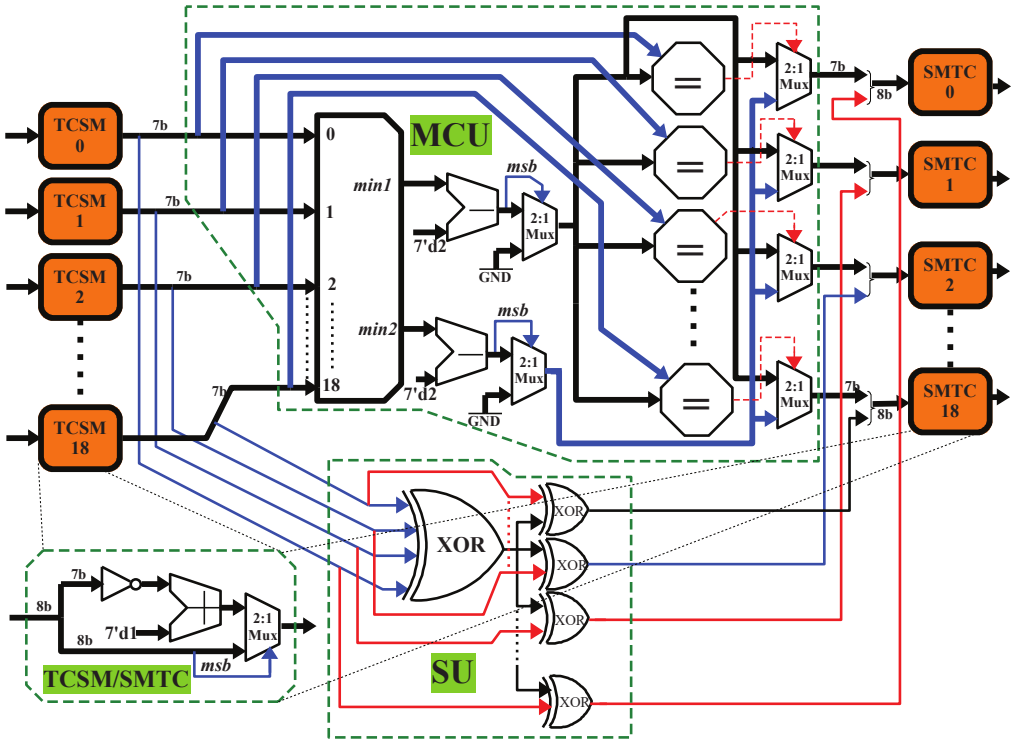


Figure 2.7: Internal digital micro-architecture of Min-Sum Approximation Unit (MSAU) that performs the check nodes (CNs) operations.

suite from Xilinx. This verified design is synthesized, placed, and routed in Xilinx Kintex-VII series of FPGA board. Static timing analysis of our design indicates that the critical path delay is 26.7 ns which corresponds to maximum clock frequency of 34 MHz. Table 2.2 shows the comparison result of our implementation with the reported LDPC decoder architectures where our suggested design consumes lesser hardware by ≈ 87% compared to [29]. Similarly, the suggested LDPC decoder architecture incurs better hardware utilization of ≈7% with respect to [52]. Hence, this QC-LDPC decoder architecture has been designed to achieve adequate BER performance with lower hardware-consumption for the 5G-NR communication standard.

Table 2.2: Hardware Utilization Comparisons of the Partially-Parallel QC-LDPC Decoder Architecture with the state-of-the-art implementations compliant to 4G-LTE and 5G-URLLC.

Resources	Proposed	[29]	[52]
FPGA Board	**Kintex-VII**	Kintex-VII	Virtex-VII
Code-length	**26112**	1296	1944
LUTs	**28450**	247411	30605
Slice Registers	**3749**	7128	-NA-
MUXes	**858**	-NA-	-NA-
Throughput (Mbps)	**4.9**	1100	900

2.6 High-Throughput QC-LDPC Decoder Architecture

In this section, a novel fully-parallel QC-LDPC decoder architecture that is compliant to 5G-NR wireless communication standard has been presented. This decoder is designed based on the conventional layered OMS decoding algorithm in order to achieve adequate BER performance. This decoder has been designed based on the replication technique to deliver high data-throughput.

2.6.1 Overall Decoder Architecture

A top-level design of the proposed high-throughput QC-LDPC decoder architecture is shown in Fig. 2.8 where the input is logarithmic-likelihood-ratios (LLRs) of 7-bits quantization that is generated by soft-demodulator module at the receiver side. At first, input LLRs are buffered in the initial register memory bank (IRMB) whose specific configuration is presented later in the next subsection. There are 68 output ports of IRMB and each one of them represents 384 7-bits LLRs (i.e. 384×7-bits = 2688 bits), as shown in Fig. 2.8. All these 68 outputs are fed to memory selection multiplexer network (MSMN) in a single clock cycle. Therefore, MSMN is a stack of 19 parallel 68:1 multiplexers (MUXes) that selects 19 LLR bunches (19-LLRBs) based on the columns (where non -1's elements exist) of the selected BG B matrix layer. For example, layer-1 to layer-4 of B matrix has 19 non -1's elements; on the other side, layer-5 has only three non -1's elements (1^{st}, 2^{nd} & 27^{th} columns). Thus,

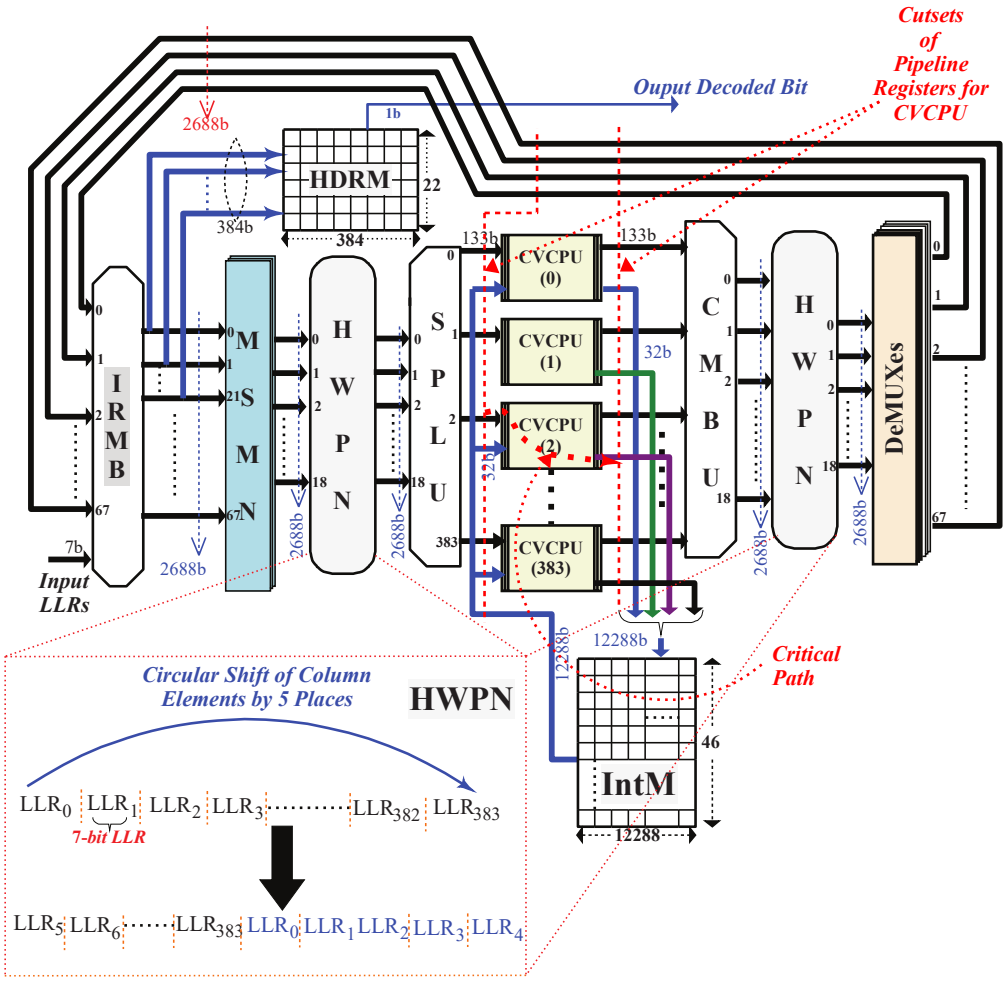


Figure 2.8: VLSI architecture of QC-LDPC decoder compliant to 5G-NR wireless communication standard supporting a frame length of 26112 bits which is encoded using $B1$ base graph matrix.

three initial MUXes of MSMN select LLRCs corresponding to 1^{st}, 2^{nd} & 27^{th} columns and the remaining 16 MUXes pass the maximum values (7-bits LLR) of '63'. Consecutively, these 19-LLRBs is rotated by the hard wired permutation network (HWPN) based on their corresponding B layer column element. The detailed operation of HWPN is schematically illustrated in Fig. 2.8. Furthermore, it shows that the 2688-bits 19-LLRBs are converted into 133-bits 384-LLRBs with the aid of a splitter unit (SPLU). The detailed schematic configuration of SPLU has been presented in 2.9 (a). The SPLU output is defined by $y(i) = \{A_i(0,i), A_i(1,i), ..., A_i(18,i)\}\ \forall\ i = \{0,1,2,....,383\}$ where $y(i)$ is the 133-bits LLRBs and 'A'

represents single 7-bits LLR. These 133-bits 384 LLRBs and the output data fetched from intermediate memory (IntM) are applied to 384-replicated combined VN & CN processing units (CVCPUs) through the input registers of the first-pipeline stage that enforces the critical path to fall in CVCPU. Hence, each of these 384 CVCPUs produces 133-bits updated

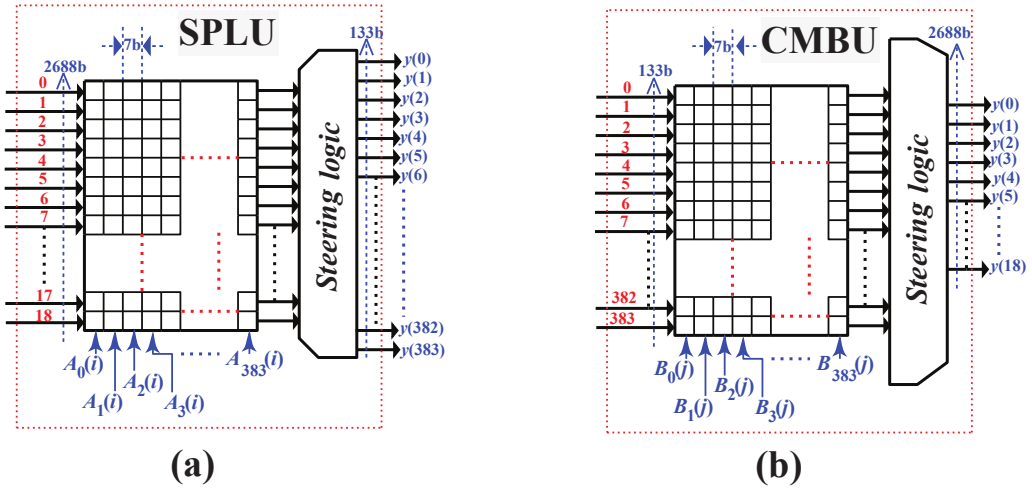


Figure 2.9: (a) Micro-architecture description of splitter unit (SPLU) architecture. (b) Digital VLSI description of combiner unit (CMBU).

LLRs and 32-bits compressed-extrinsic LLR (CE-LLR) by performing CN and VN processes. Therefore, 384-parallel CVCPUs send 133-bits updated 384-LLRBs to combiner unit (CMBU) for the next B layer, and concatenation of 32-bits 384 CE-LLRs to IntM for the next iteration through the second-pipeline stage, as shown in Fig. 2.8. Hence, these pipelining stages make the critical path of the proposed decoder lie within CVCPU architecture.

The CMBU performs the reverse operation of SPLU that transforms 133-bits updated 384-LLRBs into 2688-bits updated 19-LLRBs, as shown in Fig. 2.9 (b). Hence, the outputs from CMBU are defined by $y(j) = \{B_j(0,j), B_j(1,j), ...B_j(383,j)\}\ \forall\ j = \{0,1,2,....,18\}$ for the LLRs conversion, where each $y(j)$ is 2688-bits updated LLRBs and B is the updated LLR of 7-bits. Subsequently, these updated 19-LLRBs from CMBU are again circularly shifted (via HWPN shown in Fig. 2.8) by the factor of (z-η_B) where z represents expansion factor (i.e. 384 in our design) and η_H is corresponding B layers column-elements. The 2688-bits 19-LLRBs are fed back to IRMB through the network of de-multiplexers (De-MUXes), as

shown in Fig. 2.8. In addition, few specific register memory units (RMUs) of IRMB (where non -1's column elements exist in corresponding B layer) are revised with the 2688-bits updated LLRBs in a single clock cycle. Hence, the proposed decoder consumes two clock cycles to revise the entire memory locations of IRMB and IntM with the updated LLRs and CE-LLRs, respectively. Such updating of LLRs in IRMB as well as CE-LLRs in IntM are necessary for the processing of the next B matrix layer and iteration, respectively. Therefore, these LLRs are again read from the IRMB in a clock cycle for the next B layer, processed for LLR updation and sent back to IRMB in the next clock cycle. The aforementioned LLR updating process continues till 46^{th} layer of B matrix that marks the completion of the first decoding-iteration and it consumes 2×46 = 92 clock cycles. Eventually, at the end of the first iteration, IntM of the proposed decoder is filled with B layer CE-LLRs.

In the next decoding iteration, IntM reads out the CE-LLRs to 384-replicated CVCPUs corresponding to new B matrix layer. For example, if IRMB is involved in the 4^{th} layer of B matrix then IntM also reads out CE-LLRs corresponding to 4^{th} B layer. Such CE-LLRs and upcoming LLRs (i.e. SPLU outputs) are processed by CVCPU to generate updated 384-LLRs for the next layer and 384 new CE-LLRs for next iteration (which is stored in IntM), as shown in Fig. 2.8. Our LDPC decoder has been designed to support 10 decoding iterations which delivers adequate performance. Thereafter, the hard decision register memory (HDRM) is activated to store all sign bits −MSBs− of the stored updated LLRs from IRMB and this consumes takes single clock cycle. Eventually, these stored bits in HDRM are fetched sequentially in every clock cycle.

2.6.2 Memory Unit Configuration

The standardized code-length for 1/3 code-rate in the B matrix is 26112 bits for the maximum expansion factor (z = 384), based on 5G-NR specifications. In our design, IRMB is a stack of 68 register memory units (RMUs) in which the initial 22 RMUs correspond to information data and remaining 46 RMUs store parity data. Each RMU comprises 384 memory locations with 7-bits of word-length and these stored 384 LLRs are read from RMU in single clock

cycle. Thus, IRMB has 26112 memory locations (i.e. 384×68) that sequentially store the quantized input LLRs (7-bits each) and read simultaneously from RMUs in single clock cycle. Thus, the overall memory size of IRMB is 26112×7b = 182.784 kb. Here, IntM memory stores all 384 replicated CVCPU CE-LLR combinations, for all 46 matrix layers, where each CE-LLR is of 32-bits in sign-magnitude format. Thereby, CE-LLR combinations have bit-

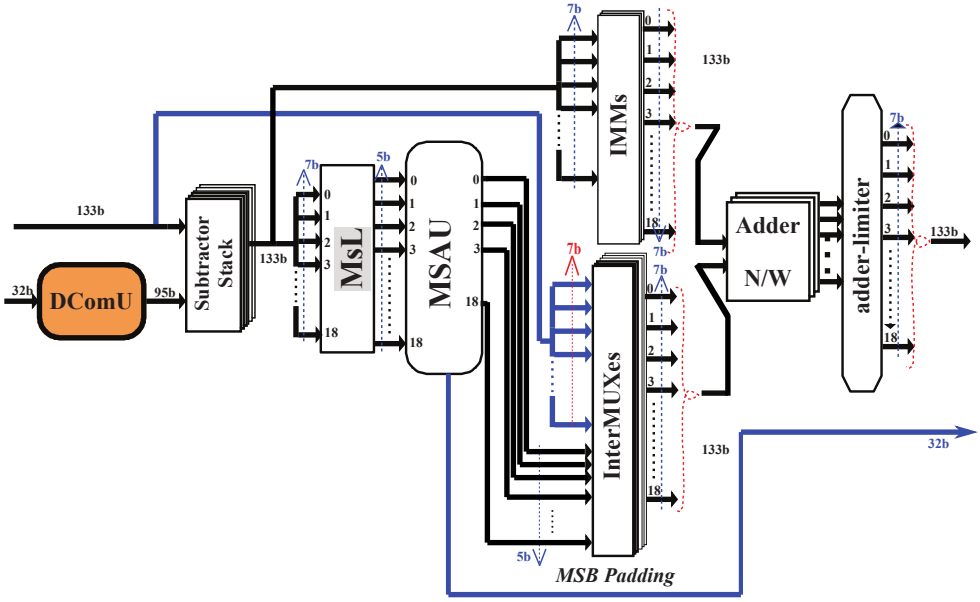


Figure 2.10: Suggested VLSI architecture of CVCPU used in fully-parallel QC-LDPC decoder architecture design.

width of 12288-bits (i.e. 384×32b). Hence, IntM has a memory depth of 46 with 12288-bits of word-length for storing the CE-LLR combinations for all B matrix layers. Each word-length of IntM stores the CE-LLR combination corresponding to single B matrix layer and is read out within one clock cycle. Further, IntM has a size of 565.248 kb (i.e. 46×12288 bits). In addition, HDRM has a depth of 22 with word-length of 384-bits in which all the sign-bits of updated LLRs after the 10^{th} iterations are stored in a single clock cycle and it has a size of 8.448 kb (i.e. 22×384 bits). Therefore, the overall memory size used in the suggested LDPC decoder is 765.48 kb.

2.6.3 Proposed CVCPU Architecture

The microarchitecture of CVCPU has been shown in Fig. 2.10 where 133-bits LLRs and 32-bits CE-LLRs are fed as inputs, as discussed earlier. Firstly, the received CE-LLRs are processed by a decompression unit (DComU) which converts this compressed form of LLRs into 19 extrinsic LLRs (ex-LLRs) with the aid of 19 equalizers, 2:1 MUXes, and sign-magnitude to 2's complement units (SMTCUs), as shown in Fig. 2.11. These 19 ex-LLRs (5-bits each) are subtracted with incoming LLRs (7-bits each), as described in 13^{th} line of Algorithm 1 using the stack of subtractors. This operation is also known as variable node updation process. Subsequently, these 19 subtracted LLRs are fed to the min-sum limiter (MsL) as well as intermediate-minimum MUXes (IMMs). The MsL truncates 7-bits of LLRs into 5-bits within the range of -15 to 15. These truncated LLRs are processed by min-sum approximation unit (MSAU) which is comprehensively presented in the next subsection. This unit performs the check node updation process that is mathematically expressed in lines $14-17$ of Algorithm 1.

Thereby, this MSAU delivers 19 updated LLRs (each of 5-bits word-length) and 32-bits CE-LLR to a network of 19 2:1 intermediate MUXes (InterMUXes) and IntM, respectively, of our decoder architecture presented in Fig. 2.8 & Fig. 2.10. These InterMUXes and IMMs select the updated LLRs and subtracted LLRs, respectively, and their selection processes are based on the B matrix layers. For example, from layer 1-4 of the B matrix, all the updated and subtracted LLRs are selected by InterMUXes and IMMs, respectively, because these layers have 19 non -1's elements in the B matrix layer. However, 5^{th} layer of B matrix has only three non -1's elements; hence, three initial InterMUXes and IMMs select three updated LLRs and three subtracted LLRs, respectively. Hence, the remaining 16 InterMUXes and IMMs choose the incoming LLRs and '0' decimal value, respectively. These 19 updated and subtracted LLRs are added to generate 8-bits updated beliefs for the specific layer. These beliefs pass through the adder-limiter which converts them to 7-bits that range from -63 to 63 and transfers them to CMBU of the proposed decoder architecture, as shown in Fig. 2.8.

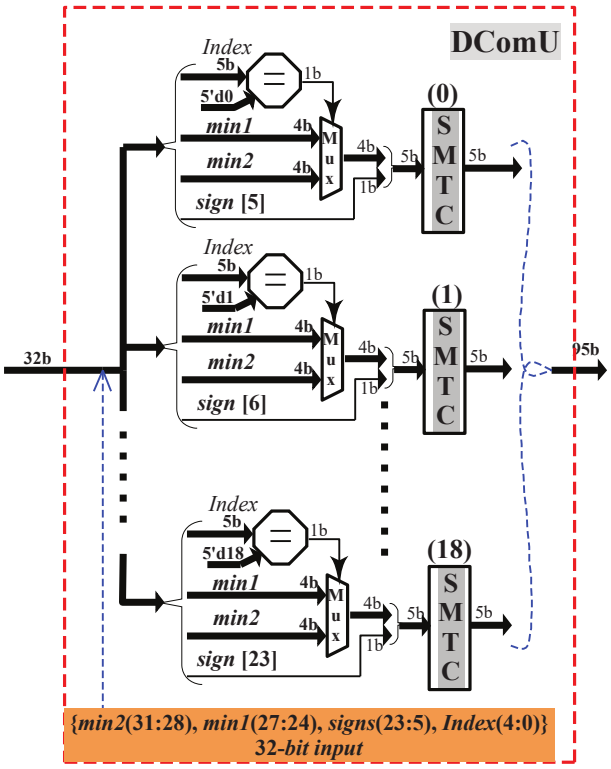


Figure 2.11: Suggested VLSI architectures of decompression unit used in CVCPU module of QC-LDPC decoder design.

2.6.4 Suggested MSAU Architecture

As discussed earlier, MSAU delivers the CN updation process from the Algorithm 1. The MSAU microarchitecture is fed with 19 LLRs generated by MsL, as shown in Fig. 2.10. These 5-bits LLRs are in the 2's complement format which is converted into 5-bits sign-magnitude (SM) format by 2's complement-to-sign-magnitude units (TCSMUs), as illustrated in Fig. 2.12 where the proposed architecture of MSAU has been presented. It shows that MSBs of these 19 SM-LLRs are XORed to generate a product bit and the remaining 4-bits of these 19 SM-LLRs are processed by magnitude comparison unit (MCU). The product bit is XORed with every 19 MSBs to produce the 19 updated sign-bits and are applied to their corresponding SMTCUs, as shown in Fig. 2.12. The remaining 4-bits magnitude of 19 SM-LLRs are fed to MCU where two minimum values (*min1* and *min2*) and the *index* of second minima which is chosen by the tree-structure (TS) approach in minimum-value generator

(mVG-19) unit [53]. These *min2* (4-bits width), *min1* (4-bits width), sign-bits (19-bits width) and *index* (5-bits width) are aggregated to generate 32-bits CE-LLR value, which is stored in IntM, and used for next iterations in our decoder. Furthermore, an offset binary value

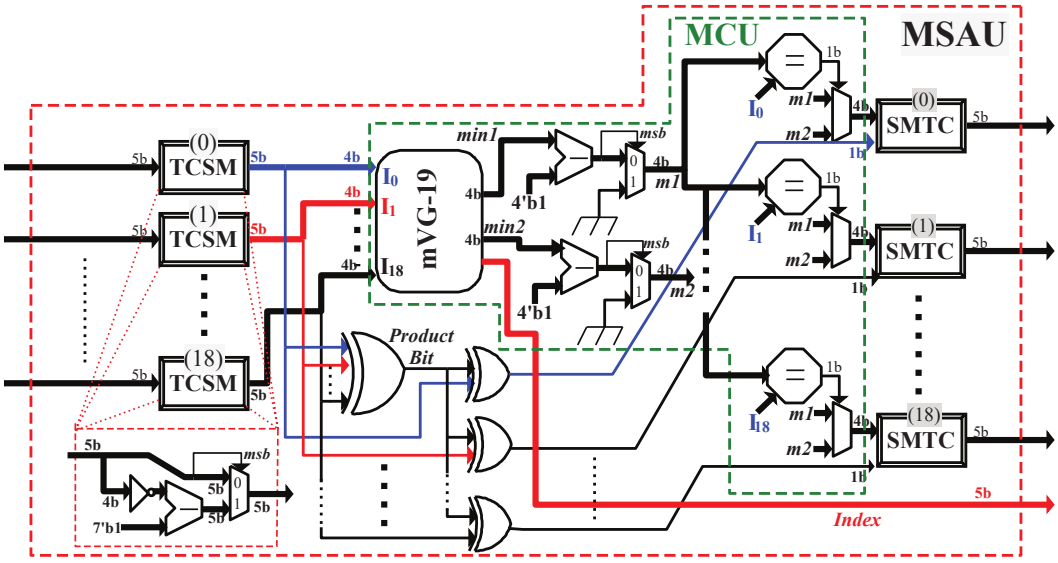


Figure 2.12: Suggested VLSI architecture of min-sum approximation unit (MSAU) used in our fully-parallel QC-LDPC decoder design.

'00001' is subtracted from both *min1* as well as *min2* and (these subtracted values must be non-negative) are passed through two 2:1 MUXes, as shown in Fig. 2.12. Therefore, all 19 magnitudes are replaced by *min1* except the index of *min2*. This replacement has been performed by 19 equalizers and 2:1 MUXes shown in Fig. 2.12. Finally, these 19 4-bits updated magnitude from MCU and their corresponding 19 updated sign bits from SU are applied to 19 parallel SMTCUs that convert these SM-LLRs into 2's complement LLRs.

2.6.5 Hardware Implementations and Comparisons

The proposed decoder has been synthesized and post-route simulated on the Xilinx Ultra-scale+ FPGA platform. Moreover, the static timing analysis indicates that our design attains timing-closure sign-off at 102.45 MHz of maximum clock frequency (f_{max}), i.e. 9.76 ns of critical path delay. Table 2.3 presents the comparison of our implementation results with other state-of-the-art decoders. Here, the throughput (Θ_T) of our LDPC decoder is obtained

Table 2.3: Comparisons of fully-parallel QC-LDPC Decoder Architecture With the Reported FPGA-Implementation Results.

Specifications	This work	[54]	[30]	[55]
Quantization	**7-bits**	4-bits	7-bits	8-bits
FPGA Board	**Zynq Ultrascale+**	Altera Stratix	–	–
Max. Throughput (Mbps)	**2900**	2000	144.6	625
LDPC code	**Irregular**	Irregular	Regular	Irregular
Code-length	**26112**	24576	12	1280
Exp. factor	**384**	1024	–	32
Max. frequency (MHz)	**102.45**	100	120.5	125
Critical path delay (ns)	**9.76**	10	8.29	8
Scheduling	**Layered**	flooding	Layered	Layered
Code rate	**1/3**	5/6	1/2	4/5
Decoder architecture	**Fully parallel**	Partially parallel	Partially parallel	Fully parallel
Base Matrix size	**46×68**	4×24	–	8×40
Standard	**5G-NR**	LDPC-CC	–	Space data system
Layers/iteration	**46**	–	–	3
Maximum iteration	**10**	18	10	20
Pipeline stages	**2**	3	–	–
Memory size (kb)	**765.48**	4402.268	–	328.744
Decoding Algorithm	**Offset Min-Sum**	Sum Product	Min-sum	Scaled Min-Sum

using the mathematical formulation is expressed in (2.1)

$$\Theta_T = \frac{n \times f_{max}}{I \times \eta_{clk}} \tag{2.1}$$

where the code-length (n) is 26112 bits and I represents the number of iterations (i.e. 10 iterations in our work). In addition, the proposed decoder requires 2 clock cycles to process each B layer and there are 46 such layers to be processed in each iteration. Therefore, the number of clock cycles consumed for single iteration (η_{clk}) is 92 clock cycles. Therefore, the suggested QC-LDPC decoder achieves a peak throughput of 2.9 Gbps. Subsequently, FPGA implementation results indicate the hardware utilization as follows: 1448762 lookup tables (LUTs), 211238 registers, 312497 F7 MUXes, and 189168 F8 MUXes. Table 2.3 indicates

that our design delivers 31.03% better throughput than the highest throughput achieved by the similar implementation [54] and 20× better than the state-of-the-art LDPC decoder implementation [30]. This decoder architecture delivers the high data-throughput for the code-rate of 1/3 with enhanced utilization of hardware resources.

2.7 Summary

This chapter presented a conventional OMS decoding algorithm with layered scheduling for 5G-NR wireless communication systems. This decoding algorithm performs various decoding operations such as initialization, VN & CN updations, and belief computations. Based on a conventional OMS decoding algorithm, this chapter presents a comprehensive BER performance analysis for all the standardized code-rates that are compliant to the 5G-NR applications. We also analyzed the decoding performance of QC-LDPC codes for different quantization-bits and various number of decoding iterations. This decoding algorithm successfully achieves the 10^{-6} BER at 1.1 dB SNR with 10 decoding iterations for the standard code-rate of 1/3.

Consequentially, a new partially-parallel QC-LDPC decoder architecture based on a layered OMS decoding algorithm has been presented that is compliant to 5G-NR wireless communication systems. This proposed QC-LDPC decoder has been designed based on iterative decomposition technique that leads to hardware-efficiency. Further, this QC-LDPC decoder architecture has been FPGA prototyped on the Xilinx ultrascale+. It is observed that the operating clock frequency of this decoder is 34 MHz and delivers the data-throughput of 965 Mbps. We compared its place and route implementation results with state-of-the-art works that show 87% reduction in hardware utilization.

Iterative decomposition technique improves the hardware-efficiency but enhances the decoding-latency of the decoder. Hence, it is necessary to design a digital VLSI architecture that exhibits lower decoding-latency and is suitable for higher data-throughput in 5G-NR applications. Furthermore, a new fully-parallel QC-LDPC decoder architecture has been

designed to deliver lower decoding-latency and higher data-throughput. This decoder

Table 2.4: Comparisons of Partially-parallel and Fully-parallel QC-LDPC Decoder Architecture with their FPGA Implementation Results.

Specifications	Partially-Parallel	Fully-Parallel
Quantization-bits	8	7
FPGA Board	Kintex-VII	Zynq Ultrascale+
Lookup Tables	28450	1448762
Registers	3749	211238
Maximum Throughput (Mbps)	965	2900
Maximum Clock Frequency (MHz)	34	102.45
Critical Path Delay (ns)	26.7	9.76
Maximum Iteration	10	10
Pipeline Stages	NA	2

architecture has been constructed based on a replication technique to achieve higher data-throughput. In addition, the decoder architecture provides a comprehensive description of architectural aspects for all the internal modules of the QC-LDPC decoder that are compliant to the 5G-NR standard. This fully-parallel QC-LDPC decoder has been prototyped on the Xilinx ultrascale+ platform using FPGA technology. According to the static timing analysis (STA), the CVCPU module is the primary component that is responsible for the critical path, which has a duration of 9.76 ns and operates at the maximum clock frequency of 102.45 MHz. It delivers a data-throughput of 2.9 Gbps while decoding at the 5G-NR standardized code-rate of 1/3. Thus, the takeaway point from this chapter is in-depth understanding of methodology as well as the challenges of designing the next-generation LDPC channel-decoder for the 5G-NR physical layer.

In this chapter, both LDPC decoder architectures can be enhanced to accommodate all standardized code-rates by regulating the layer processing mechanism on the base graph B matrix. These architectures are reconfigurable and can be expanded to support the various PCMs of different wireless-communication standards. The overall memory bank consumption can be reduced by processing a lesser number of LLRs or by using the LLRs puncturing mechanism. Table 2.4 presents the comparisons of partially-parallel and fully-

parallel QC-LDPC decoder architecture. Here, the partially-parallel architecture delivers moderate data-throughput with lower hardware-consumption whereas the fully-parallel decoder architecture delivers higher data-throughput at the cost of hardware. Therefore, it is imperative to develop a digital QC-LDPC decoder architecture that minimizes the hardware-consumption and maximizes the data-throughput of LDPC decoder. Further, this thesis also introduces a new decoder architecture that has been designed using a new log-likelihood-ratio compound (LLRC) technique based LDPC decoding algorithm in the upcoming chapters. This proposed mechanism enhances the data-throughput with lower hardware-consumption.

Chapter 3

LLRC-Segregation based Binary QC-LDPC Decoding Algorithm and Architecture

---

3.1 Introduction

It is important to maintain low-latency for 5G-NR applications like internet-of-things (IoTs), autonomous vehicles (AVs), and ultra-reliable low-latency communications (URLLCs). Hence, an efficient decoding algorithm for LDPC codes is crucial for practical implementation in wireless-communication systems. As discussed in sections 2.1 and 2.3 of this thesis, efficient decoding algorithms such as MS, offset MS, and normalized MS [49, 15, 50] have low hardware implementation complexity and are suitable for real-time applications in 5G-NR systems. Consequently, these algorithms may undergo adverse phenomena of error floor with the increasing value of SNR. These consequences can limit the performance of systems in ultra low error-rates scenarios like deep-space communications. Further, these LDPC decoding algorithms have high data-congestion and routing-complexity issues while

performing the CN and VN operations that lead to data-overcrowding during hardware-implementation. In this chapter, we aim to address these problems and introduce a novel LLRC-segregation based OMS decoding algorithm with layered scheduling. This approach alleviates the error floor and routing-complexity in the QC-LDPC decoder and allows it to achieve higher data-throughput and better hardware-efficiency. Further, a Monte-Carlo simulation for the LLRC-segregation based OMS decoding algorithm has been presented, and we have also compared our proposed algorithm with the existing algorithms in the literature.

In addition, a novel QC-LDPC decoder architecture has been proposed that is compliant to 5G-NR wireless communication standard. This decoder has been designed based on the proposed LLRC-segregation technique. We suggested a new LLRC routing method that reduces routing-complexity and data-congestion in the QC-LDPC decoder architecture. This LLRC mapping approach improves the hardware-efficiency and decreases the critical path of the overall QC-LDPC decoder architecture. It supports all the standard code-rates and code-lengths specified in the 5G-NR wireless communication standard [1]. The proposed QC-LDPC decoder is resilient and can be integrated into all other 4G-LTE protocols, including WiFi, WiGig, and WiMax. We have included some additional Boolean optimizations in the decoder architecture sub-modules to further improve hardware-efficiency. Finally, the overall QC-LDPC decoder has been placed and routed on the Xilinx Zynq Ultrascle+ FPGA platform. The key highlights of this chapter are as follows:

- A hardware-friendly OMS decoding algorithm based on the LLR compound (LLRC) segregation technique has been proposed in this chapter. It performs CN and VN operations in matrix form and also alleviates routing-complexity as well as data-congestion in hardware implementations. Further, we carried out comparative and comprehensive performance analyses of the proposed QC-LDPC algorithm with the existing algorithms in the literature.

- In addition, a novel QC-LDPC decoder architecture based on the proposed LLRC-

segregation based decoding algorithm has been presented in this chapter. It routes LLRC based on a specific routing technique that enhances hardware-efficiency and data-throughput. This decoder has been designed for the 5G-NR specifications that support all the standard code-lengths and code-rates of the contemporary 5G-NR wireless communication standard.

- Further, additional logical optimizations have been applied while designing a hardware-efficient QC-LDPC decoder architecture for the 5G-NR standard. Finally, the suggested QC-LDPC decoder has been hardware implemented on the FPGA platform, and its results are compared with relevant state-of-the-art implementations.

3.2 LLRC-Segregation Based Offset Min-Sum QC-LDPC Decoding Algorithm

The proposed LLRC-segregation OMS based QC-LDPC decoding algorithm with layered scheduling is presented in Algorithm 2. This algorithm alleviates the hardware routing-complexity as well as data-congestion of the QC-LDPC decoder architecture by using the new LLRC segregation technique. Here, all the computations of CN and VN messages are performed in the matrix form, where each row of the matrix represents single LLRC. This matrix based LLRC-segregation technique provides the simplification and fast accessibility of LLRs and also mitigates the problem of data-overcrowding while performing CN and VN operations. All the CN and VN operations are executed in the fully-parallel mode to achieve faster data computations.

Therefore, the initialization process for LLRC-segregation based OMS decoding algorithm for QC-LDPC codes is mathematically expressed in lines 2–7 of Algorithm 2 where the extrinsic LLRC E matrices are assigned by null values i.e. null matrices, as shown in line 3, where T represents the number of non -1 elements in entire B matrix. Subsequently, z number of LLRs from l_α to l_γ are investigated as LLRC, assigned to j^{th} row of V matrix, as shown in line 6 of Algorithm 2. Thereby, iterative exchange and computations of LLRCs

Algorithm 2 Proposed LLRC-Segregation based QC-LDPC Decoding Algorithm.

1: **Inputs:** $L = \{l_1, l_2, \cdots l_n\}$; **Output:** $\widehat{X} = \{\chi_1, \chi_2, \cdots \chi_n\}$;
2: **Initialization:**
3: $V_{n_b \times z} = [0]_{n_b \times z}$; $E_{T \times z} = [0]_{T \times z}$;
4: **for** $j = 1$ to n_b
5: $\alpha = (j-1) \times z + 1$; $\gamma = j \times z$;
6: $V(j) = [l_\alpha : l_\gamma]$; ▷ **LLRC updation in** V **matrix**
7: **end for**
8: **Iteration loop**
9: **for** $j = 1$ to m_b
10: *- - - **Selected** S **LLRCs** as shown in (3.1) - - - * \
11: $\alpha = \gamma + 1$; $\gamma = \alpha + t_b - 1$;
12: *- - - **VN** V **LLRCs** updation by (3.3) - - - * \
13: *- - - **CN** C **LLRCs** updation by (3.4) - - - * \
14: *- - - **AP** D **LLRCs** updation by (3.6)- - - * \
15: $V(p_i) \leftarrow D(j)\ \forall\ p_i \in P, j \in \{1, 2, \cdots, tb\}$;
16: **end for**$^{(j)}$
17: $A_{1 \times n} = [V(1), V(2), \cdots\ V(n_b)] = [a_1, a_2 \cdots\ a_n]$;
18: *- - - **Hard-decision** by (3.7) - - - * \
19: *- - - **Syndrome-check** by (3.7) - - - * \
20: **end Iteration loop**

between CN and VN are shown in lines 8−19 of Algorithm 2. Firstly, the LLRC selection process is performed based on the index-vector $P = \{p_1, p_2, \cdots, p_{t_b}\}$ where p_i and t_b are the index and number of non -1 elements in any j^{th} row of B matrix, respectively.

$$S_{tb \times z} = [V(p_1); V(p_2); \cdots; V(p_{t_b})] \tag{3.1}$$

Here, $V(p_i)$ is defined by p_i^{th} row of V matrix and {;} represents the row-separation of matrix. In the VN-LLRC updation process, α to γ LLRCs of E are assigned into e sub-matrix to compute the VN-to-CN LLRCs (V) by using the following mathematical formulation:

$$e_{tb \times z} = [E(\alpha) - E(\gamma)] \tag{3.2}$$

$$V_{t_b \times z} = S_{t_b \times z} - e_{t_b \times z} \tag{3.3}$$

Futhermore, the LLRs of $V(i)$ LLRC are cyclic-rotated based on w_i element $\forall\ i\ \in$

$\{1, 2, \cdots, t_b\}$, where weighted-vector $W = \{w_1, w_2, \cdots, w_{t_b}\}$ is the non -1 elements for j^{th} row of B matrix. Subsequently, each CN ($C_{d,c}$) in rotated V performs the CN updation process by the given expression:

$$C_{d,c} = \prod_{c' \in V(d) \setminus c} sign(V_{d,c'}) \cdot max\left\{\left(\min_{c' \in V(d) \setminus c}(|V_{d,c'} - \beta)|\right), 0\right\} \tag{3.4}$$

Here, $c' \epsilon V(d) \setminus c$ denotes that all CNs of d^{th} column in V matrix except CN c. Further, β represents an offset value that depends on the quantization format. It is noted that the value of β is typically fixed to 1 in fixed-point representation [40, 56].

Thereby, the LLRs of $C(i)$ LLRC are inversely rotated by $(z - W)$ vector $\forall\, i \in \{1, 2, \cdots, t_b\}$ and update the t_b rows of E by C matrix, as shown in (3.6). Subsequently,, the row a-posteriori LLRCs (AP-LLRCs) D corresponds to j^{th} row of B matrix is computed based on:

$$[E(\alpha) - E(\gamma)] = C_{t_b \times z} \tag{3.5}$$

$$D_{tb \times z} = V_{t_b \times z} + C_{t_b \times z} \tag{3.6}$$

Thereby, V matrix has been replaced ($\leftarrow$) by AP-LLRCs of D matrix based on the P vector, as shown in line 15 of Algorithm 2. Therefore, lines 10-15 represent the single-row processing of B matrix. Such m_b layer processes are computed in a single iteration.

$$\chi_i = \begin{Bmatrix} 0 & a_i \geq 0 \\ 1 & otherwise \end{Bmatrix} \quad a_i \in A, \chi_i \in \widehat{X}; \tag{3.7}$$

$$syn = \widehat{X} \cdot H^T \tag{3.8}$$

Here, the $A = \{a_1, a_2, \cdots, a_n\}$ is a vector of updated A-LLRCs $V(i)\ \forall\, i \in \{1, 2, \cdots n_b\}$ that result in hard-decision vector ($\widehat{X}$) and syndrome (syn) testing by using the (3.7) as shown in lines 17-18 of Algorithm 2. If the syndrome becomes '0' then the decoding iteration is terminated

otherwise it continues until the iterations are exhausted.

3.3 Performance Analysis of LLRC-Segregation Based QC-LDPC Decoding Algorithm

The performance analyses are extensive Monte-Carlo simulations of 10^6-10^8 QC-LDPC encoded information bits that are BPSK modulated/demodulated in an additive-white Gaussian-noise (AWGN) channel environment. These simulations are carried out based on the 5G-NR specifications [1], as discussed in previous chapters. Firstly, Fig. 3.1 shows the comparative frame-error-rate (FER) analysis (at a code-rate of 1/3 and 10 decoding iterations) of QC-LDPC decoding algorithm based on the LLRC-segregation technique in two formats: floating point (FLP) and fixed-point (FP). The FP format have internal LLRs (CN messages) and AP-LLR of 5-bits and 7-9 bits quantization, respectively.

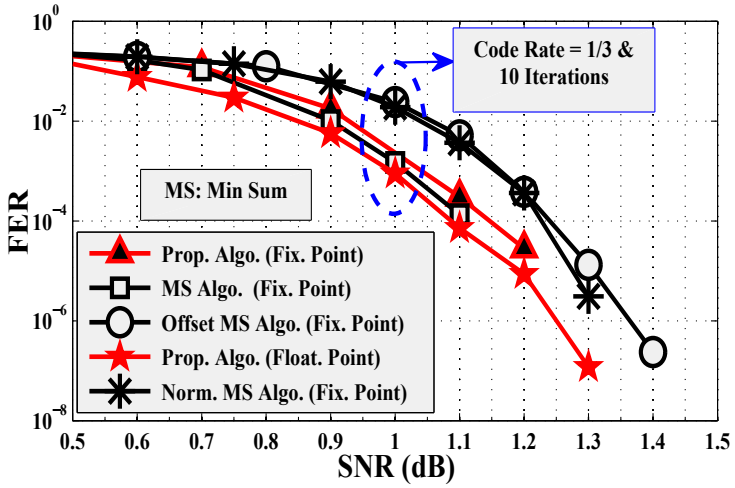


Figure 3.1: Comparative FER versus SNR plots of LLRC-segregation based offset min-sum and other QC-LDPC decoding-algorithms for the 5G-NR standard code-rate of 1/3 with 10 decoding iterations.

Here, Fig. 3.1 compares our proposed algorithm with other conventional QC-LDPC decoding algorithms which have FP simulation for internal LLRs (CN messages) of 5-bits

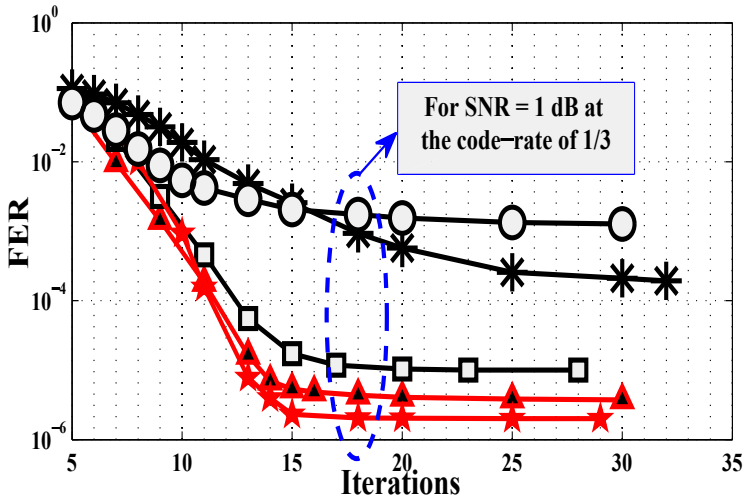


Figure 3.2: FER versus number of decoding-iterations plot for LLRC-segregation based offset min-sum QC-LDPC decoding-algorithm at the 5G-NR standard code-rate of 1/3 with SNR of 1 dB.

and LLR of 9-bits. The LLRC-segregation based algorithm (7-bits AP-LLRs) delivers 0.1 dB coding-gain at 10^{-4} FER, compared to OMS and NMS decoding algorithms. However, 0.02 dB of degradation with respect to the MS algorithm at the same FER with 5-bits internal LLR and 7-bits AP-LLR quantization format. Subsequently, FER versus iteration plots of all the aforementioned decoding-algorithms are presented in Fig. 3.2. It shows that the LLRC-segregation based QC-LDPC decoding algorithm delivers constant FER of $10^{-5.8}$, beyond 15 decoding iterations. Furthermore, the FER performance analysis of the proposed decoding algorithm for seven different code-rates, specified by 5G-NR standard [1], is presented in Fig. 3.3. It shows that adequate FER of 10^{-5} is achievable in the SNR range of 1.1 dB to 6.6 dB.

3.4 LLRC-Segregation based QC-LDPC Decoder Architecture

In the previous chapter, we discussed the distinct hardware-efficient and high-throughput QC-LDPC decoder architectures. They are designed based on the conventional LDPC decoding algorithm. Unlike, a new QC-LDPC decoder architecture has been suggested in this chapter that is designed based on a new LLRC-segregation based OMS decoding

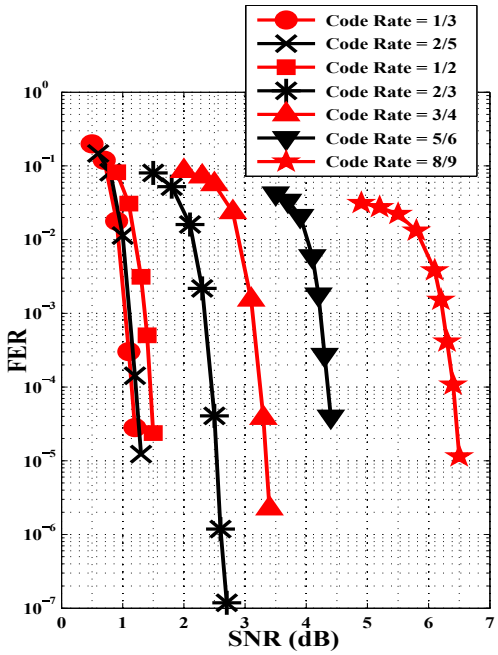


Figure 3.3: Comparative FER versus SNR plots of LLRC-segregatiom based offset min-sum QC-LDPC decoding-algorithm for all the standard code-rate of 5G-NR with 10 decoding iterations.

algorithm which is mathematically expressed in Algorithm 2. This QC-LDPC decoder architecture alleviates the data-congestion at CNs and VNs computation modules. This decoder also supports all the standardized code-rates of the 5G-NR wireless communication standard.

3.4.1 Overall Decoder Architecture

This section presents a detailed and generic architecture of the proposed QC-LDPC decoder, as shown in Fig. 3.4. It has been implemented for 5G-NR specifications [1] where the standard base graph B matrix (as shown in Fig. 1.7) has a size of $m_b \times n_b$ = 46×68 and maximum expansion factor z = 384 for decoding (n, k) QC-LDPC code where n ranges between 26112-10368 bits based on code-rates and k is 8448 bits. Nevertheless, it can also be efficiently implemented for any other standards/applications. Fig. 3.4 shows proposed QC-LDPC decoder architecture has been fed with primary inputs of n quantized channel-LLRs $L = \{l_1, l_2, l_3, \ldots, l_n\}$ which are routed via 1:z demultiplexer (DeMUX). Its outputs

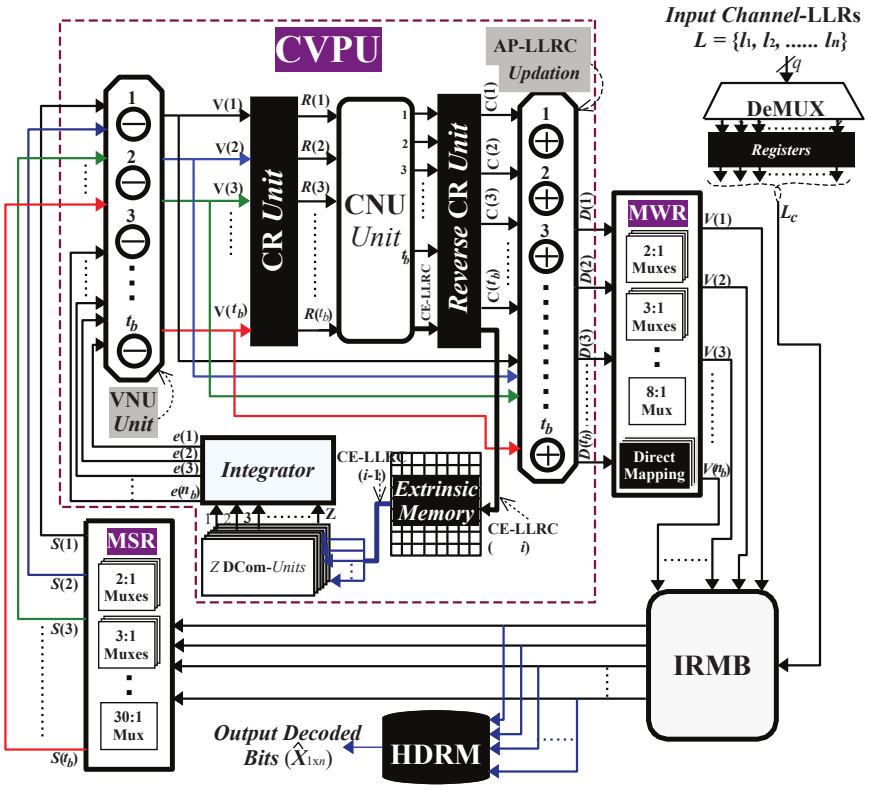


Figure 3.4: Proposed hardware-efficient architecture of QC-LDPC decoder and its submodules: MSR, CVPU, MWR, IRMB & HDRM that are designed with n_b = 68 and maximum t_b = 19.

are buffered (using registers) for every z clock cycles that accumulates z channel-LLRs which are concatenated into LLRC, referring Fig. 3.4, with the bit-width of Q×z bits where Q denotes bit-quantization of each channel LLR. Such generated LLRC denoted by L_c is directly fed to the initial register memory bank (IRMB), as shown in Fig. 3.4. These DeMUX buffering and concatenation processes repeat for n_b× that n_b LLRCs and stores LLRCs into IRMB. Hence, various sub-modules like memory selection router (MSR), check-nodes & variable-nodes processing unit (CVPU), memory writing router (MWR), and hard decision register memory (HDRM) process their corresponding data signals and route LLRC to next modules, as presented in Fig. 3.4.

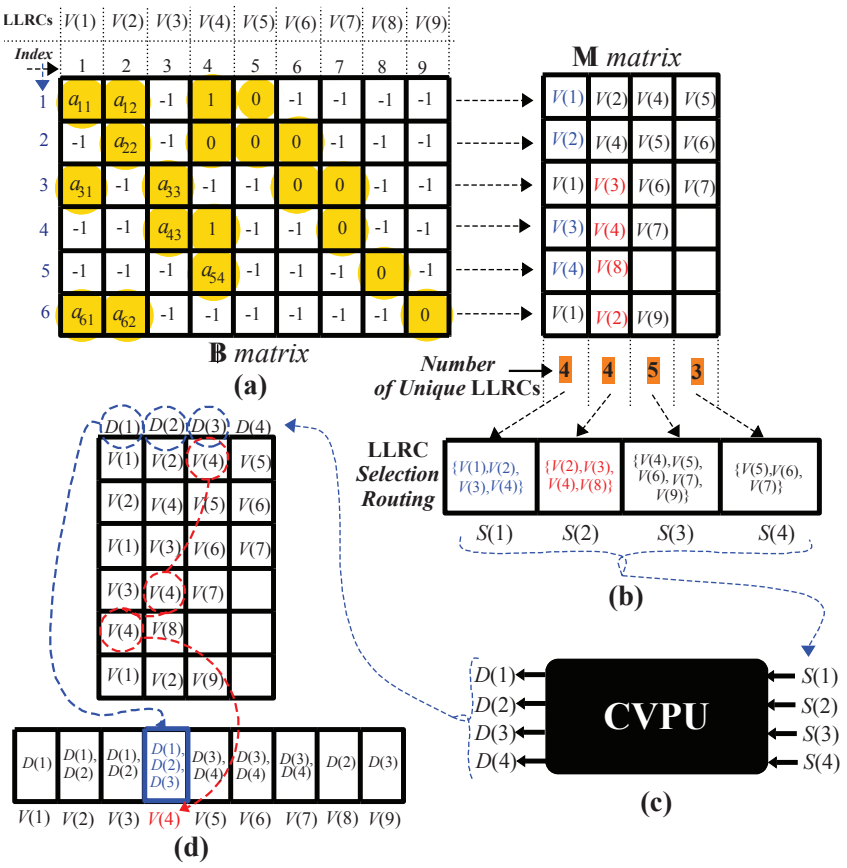


Figure 3.5: (a) Mapping of LLRCs $V(j)\ \forall\ j \in \{1,2,\cdots n_b\}$ with $\mathbb{B}$ matrix of the size $m_b \times n_b$ = 6×9 (b) generation of $\mathbb{M}$ matrix and routing of $S(j)\ \forall\ j \in \{1,2,\cdots t_b\}$ LLRCs (referring MUXes of MSR). (c) CVPU that processes S LLRCs to generate updated D LLRCs (d) mapping of D LLRCs with the column of $\mathbb{M}$ matrix (referring MUXes and direct mapping of MWR).

3.4.2 Memory Selection Router (MSR)

The V LLRCs are read from IRMB and are fed to MSR. It routes $V(j)\ \forall\ j \in \{1,2,\cdots n_b\}$ LLRCs into a specific combination of $S(k)\ \forall\ k \in \{1,2,\cdots t_b\}$ LLRCs. As shown in Fig. 3.4, MSR requires 19 parallel MUXes of various sizes viz. six 2:1 MUXes, five 3:1 MUXes, and one MUX of sizes: 5:1, 7:1, 12:1, 14:1, 17:1, 20:1, 23:1, and 30:1 which are based on proposed LLRCs selection routing in Fig. 3.5 (b). On the contrary, conventional MSR requires 19 68:1 multiplexers [41], resulting in 90% more hardware utilization and longer delay compared to the proposed MSR, as shown in Table 3.1. This subtask is mathematically represented in

line 10 of Algorithm 2 and functionally invoked by (3.1).

Table 3.1: Hardware utilization and delay comparisons of proposed and conventional decoder-implementations on FPGA board (Xilinx ZynQ-ZCU102 Ultrascale+).

Modules	Conventional Architecture [41]		Proposed Architecture		Hardware Saving (%)
	Number of MUXes	Delay (ns)	Number of MUXes	Delay (ns)	
MSR	19 68:1 MUXes	5.613	19 different MUXes (Max. Size: 30:1)	4.560	90
MWR	68 19:1 MUXes	5.226	25 different MUXes (Max. Size: 8:1)	3.091	91.40
MSU	452†	7.628	309†	8.2	31.60
Overall†	2161665	17.06	992367	7.4	54

†: Hardware utilization = (# of LUTs + # of F7 MUXes + # of F8 MUXes + # of Flip-Flops).

3.4.3 CN & VN Processing Unit (CVPU)

A new CVPU module has been introduced that leads reduction in hardware consumption as well as achieve a shorter logical delay due to hardware optimization. This unit updates CNs and VNs to compute row AP-LLRCs (denoted by D), as illustrated in (3.6) and line 14 of Algorithm 2. Fig. 3.4 shows that S LLRCs from MSR are subtracted with e extrinsic LLRCs in VN-LLRC updation (VNU) unit to generate V LLRCs. Here, e LLRCs are decompressed versions [41] of compressed-extrinsic LLRC (CE-LLRC) of $(I\text{-}1)^{th}$ iteration that is fetched from extrinsic memory E. Subsequently, V LLRCs from VNU unit are made to pass through the cyclic-rotational (CR) unit for the permutation of LLRs. Hence, rotated V LLRCs from CR unit are fed to check node updation (CNU) unit, as shown in Fig. 3.4. Here, CNU unit performs CN-computation (referring lines 13 from Algorithm 2) to compute C CN-LLRCs and a CE-LLRC, which is comprehensively explained later in upcoming section. Furthermore, t_b updated CN-LLRCs and a CE-LLRC are processed by reverse-CR unit that generates reversely rotated C LLRCs which are used for updating the AP-LLRCs D as mathematically expressed in (3.6). The updated D are generated using t_b parallel adders that sum reverse-CR C and VN-updated V LLRCs. Further, reverse-CR unit also computes

CE-LLRC for the I^{th} iteration that is stored in extrinsic memory, as shown in Fig. 3.4.

3.4.4 Memory Writing Router (MWR)

A network of parallel MUXes in MWR routes updated AP-LLRCs $D(j)\ \forall\ j \in \{1, 2, \cdots, t_b\}$ into V matrix based on the line 15 of Algorithm 2. The MWR architecture has been simply designed by using 25 multiplexers of various sizes, viz. three 2:1 MUXes, five 3:1 MUXes, three 4:1 MUXes, eight 5:1 MUXes, four 6:1 MUXes, one 7:1 and one 8:1 MUX. It has also 43 direct signal mappings based on the proposed technique shown in Fig. 3.5 (d) based on 5G standard B matrix. These 25 outputs of MUXes and 43 direct-mapped signals aggregate to 68 output signals from MWR that correspond to n_b = 68 columns of B base graph matrix that is compliant to 5G-NR specifications [1]. In the conventional design [41], such a MWR unit requires 68 19:1 MUXes that incur higher routing-complexity and alleviate hardware-efficiency. Comparison of FPGA synthesis results in Table 3.1 shows that the proposed MWR delivers 91.4% better hardware saving and 40% lesser delay, compared to the conventional one. Subsequently, all n_b = 68 number of $V(j)$ LLRCs from MWR are stored in the initial register memory bank (IRMB) in single clock-cycle, as shown in Fig. 3.4.

3.4.5 Hard Decision Process

As shown in Fig. 3.4 and illustrated in line 18 of Algorithm 2, the proposed QC-LDPC decoder consumes two clock cycles to generate updated D LLRCs for each layer of B-matrix and therefore, it requires $2{\times}m_b$ clock cycles for all the B-matrix row which is equivalent to single decoding iteration. As a result, the decoding latency is given by $2{\times}i_{max}{\times}m_b$ clock cycles where i_{max} represents total number of iterations. For the 5G-NR specified $B_{46\times68}$ matrix, computations of 46 and 5 layers are needed for minimum and maximum code-rates of 1/3 and 8/9, respectively. Therefore, the proposed decoder architecture takes 100 and 920 clock cycles to compute final A-LLRCs V for i_{max} = 10 decoding iterations. Eventually, the hard-decision register-memory (HDRM) stores all sign-bits of n LLRs, extracted from the outputs of IRMB, and sequentially delivers hard decoded bits at every clock cycle, as

shown in Fig. 3.4.

3.4.6 Hardware-Efficient CNU-Unit Architecture

The proposed CNU-unit architecture is presented in Fig. 3.6. It processes cyclically-rotated V LLRCs (referring to CVPU design in Fig. 3.4). These t_b V LLRCs are logically transformed into z LLRCs via splitter [41] and fed to z parallel min-sum units (MSUs) in Fig. 3.6. In each MSU, input LLRC is segregated into t_b LLRs and applied to two's-complement (TC) to sign-magnitude (SM) units. Here, we have suggested logically optimized 5-bits TCSM unit using a conversion unit (ConU) and a 2:1 multiplexer, as shown in Fig. 3.6. This ConU

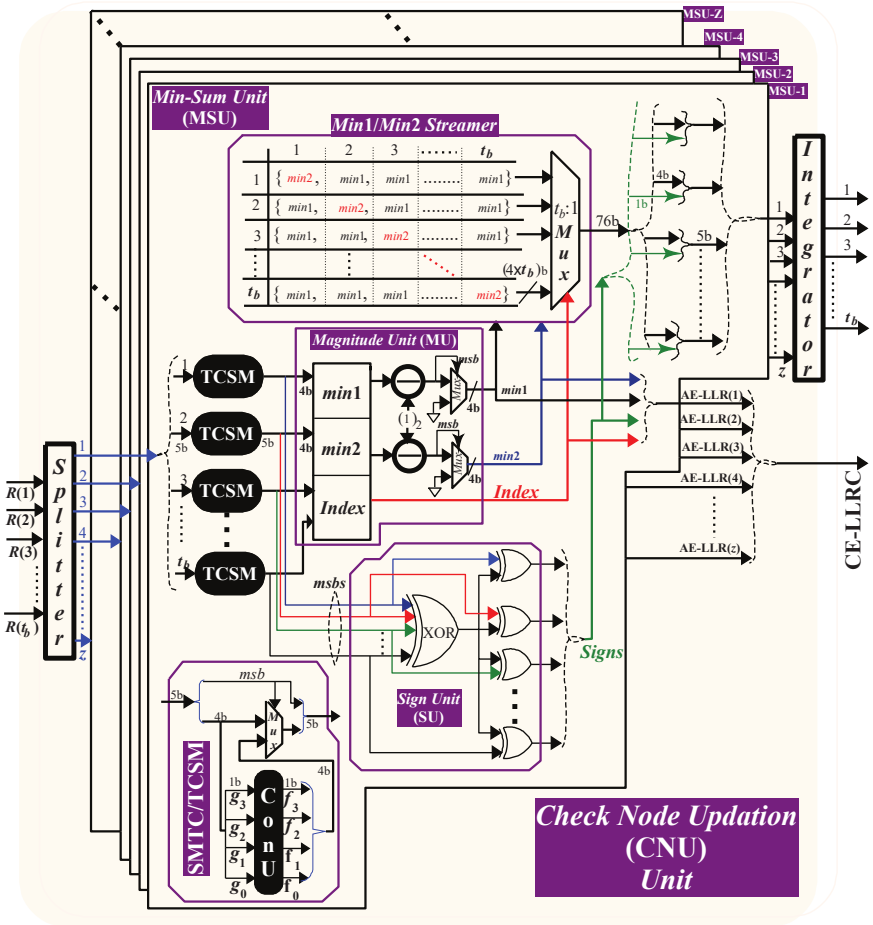


Figure 3.6: Hardware-efficient CNU Unit architecture of our QC-LDPC decoder that comprises of z MSUs in parallel.

processes only 4-bits magnitude because the sign-bit is identical in both TC and SM formats. The simplified Boolean equations for the outputs of ConU are expressed as

$$f_3 = \bar{g}_3 | (\bar{g}_2 \cdot \bar{g}_1 \cdot \bar{g}_0); \tag{3.9}$$

$$f_2 = (\bar{g}_3 \cdot \bar{g}_2) | (\bar{g}_2 \cdot g_1) | (\bar{g}_2 \cdot \bar{g}_0) | (g_2 \cdot \bar{g}_1 \cdot \bar{g}_0); \tag{3.10}$$

$$f_1 = g_1 \oplus g_0; \tag{3.11}$$

$$f_0 = g_0; \tag{3.12}$$

Such ConU –consuming only 16 gate-equivalents (GEs)– replaces 4-bit adder and an inverter in the conventional design [41] that is equivalent to 21 GEs, incurring a saving of 5 GEs in each TCSM unit. This design of the TCSM unit is also valid for SMTC converter that results $t_b + 2$ TCSM/SMTC units in each MSU which is replicated $z\times$ in CNU unit, as shown in Fig. 3.6. Hence, overall hardware saving in CNU is $5\times(t_b - 2)\times z$ GEs due to optimized ConU logic. After the TCSM conversions, signs and magnitudes of all LLRs are fed to sign-unit (SU) and magnitude-unit (MU), respectively. Here, SU and MU operations are compliant to (3.4) and line 13 of Algorithm 2. The MU generates offset-magnitudes of min1 and min2, and index of min1 (*Index*) [53], as shown in Fig. 3.6.

Such $min1$ and $min2$ values are consecutively processed by ConU logics and fed to Min1/Min2-Streamer that routes t_b-1 replications of $min1$ with a specific pattern of $min2$ via $t_b\times1$ multiplexer, as shown in Fig. 3.6. The proposed Min1/Min2-Streamer replaces the conventional logic that needs t_b units of equalizers and multiplexers [41]. Subsequently, all the updated-CN magnitudes and their corresponding signs (from SU) are concatenated, as shown in Fig. 3.6. These concatenated LLRs from z replicated MSUs are passed into an integrator [41] for the reverse-transformation logical operation to compute t_b number of CN-updated LLRCs. On the other side, $min1$, $min2$ and *Index* from MU, and signs from SU are concatenated to generate a compressed CN updation, referred as aggregated-extrinsic LLR (AE-LLR), that enhances the hardware-efficiency of extrinsic memory. Eventually, all the AE-LLRs from z MSUs are concatenated into a CE-LLRC which is applied to the reverse-

CR unit that is a subsequent module of CNU unit in CVPU, as shown in Fig. 3.4. Therefore, the implementation results in Table 3.1 shows that single MSU delivers a hardware saving of 31.6% and the proposed QC-LDPC decoder consumes 54% lesser hardware-resources in total, compared to the conventional decoder [41].

3.4.7 Hardware Implementations and Comparisons

Table 3.2: Comparison of Proposed QC-LDPC Decoder Implementation-Results with the State-of-the-Art Works.

Specifications	This work$^{\partial}$	TCAS-I 2020 [46]$^{\nabla}$	ISCAS 2020 [41]$^{\partial}$	APCCAS 2019 [48]	TCAS-II 2018 [57]$^{\Re}$
Tech. (nm)	**16**	16	16	40	40
Q-bits	**7**	8	7	6	4
f_{max} **(MHz)**	**135**	404.8	102.45	333.33	100
Θ_T **(Gbps)**	**3.56–11.02**	3.2–4.9	2.9	0.026	1.55
Λ_D **(μs)**	**0.94–7.33**	2.1–8.0	8.97	1.2	63.4
Iterations	**10**	10	10	5	10
Expansion factor (z)	**384**	384	384	96	4096
Code-rates	**1/3–8/9**	22/68–22/27	1/3	1/5	5/6
Net TP$^{✠}$ (Gbps)	**1.1**	0.49	0.29	0.005	0.155
Multi-rates	**Yes**	Yes	No	No	No
Memory	**Registers**	BRAM	Reg.	SP-RAM‡	BRAM‡
Memory Size (kb)	**756.48**	4914	765.8	–	2589.8
LUTs	**260765**	96625	1448762	4427	87749
Registers	**526972**	87751	211238	1907	49322
Hardware Utilization†	**992367**	5233977	2161665	6334	2726692
HUE$^{♮}$	**1.95**	23.12	16.20	47.5	439.78
PTLR$^{♠}$	**11723.4**	2331.75	323.29	22.2	24.44
SNR (dB)¶	**1.0–6.5**	0.4–7.4	1.0	4.5	2.7

✠: Net TP = Throughput/Iterations; †: Hardware Utilization = (# of LUTs + # of F7 MUXes + # of F8 MUXes + # of Register + # of Slices + FPGA BRAM (in bits)); ♠: *PTLR* (in Mbps/μs); ♮: *HUE* (in hardware-resources/layer/Mbps); ‡: SP-RAM = Single-port Random Access Memory; ¶: SNR required to achieve FER of 10^{-5}; ∂ : 5G-NR; ∇: 5G-NR, Wi-Max, and DVB-S2X ; $\Re$: Regular QC-LDPC

The proposed QC-LDPC decoder (based on 5G-NR specifications [1]) has been post and route (PAR) implemented on Xilinx Zynq-Ultrascale+ FPGA-board. The data-throughput

(Θ_T) and latency (Λ_D) are computed by (3.13) and (3.14), respectively.

$$\Theta_T = \frac{n \times f_{max}}{\delta + (i_{max} \times \mathbb{L} \times \mathbb{C})} \tag{3.13}$$

$$\Lambda_D = \frac{n}{\Theta_T} \tag{3.14}$$

where $\mathbb{L}$ represents the number of B matrix layers. Further, $\mathbb{C}$ denotes clock cycles required to process each B matrix layer (where $\mathbb{C} = 2$). Therefore, our QC-LDPC decoder implementation delivers a peak throughput of 11.02 Gbps while decoding (n = 10368 bits) at 8/9 code-rate with $\mathbb{L}$ = 5 layers. Our LDPC decoder consumes 120392 F7, 56004 F8 MUXes, and 28234 Slices. On the other side, two figure-of-merits (FOMs): hardware utilization efficiency (HUE) and peak-throughput to latency ratio (PTLR) are expressed by (3.15) and (3.16), respectively.

$$HUE = \frac{\mathbb{H}_u}{(\mathbb{L} \times \Theta_T)} \tag{3.15}$$

$$PTLR = \frac{\Theta_T}{\Lambda_D} \tag{3.16}$$

Here, $\mathbb{H}_u$ represents the FPGA hardware-utilization of the QC-LDPC decoder. Note that $\mathbb{L}$ is proportional to $\mathbb{H}_u$ because larger value of $\mathbb{L}$ surges the steering-logic size that enhances hardware-utilization of the decoder. Here, *PTLR* represents throughput achieved (in Mbps) by decoder per 1μs of processing time.

Table 4.2 shows the comparison of our implementation results with the relevant state-of-the-art works. It shows that the proposed QC-LDPC decoder has 2.2× higher throughput compared to the state-of-the-art Petrović et al. decoder [46]. Furthermore, our decoder architecture has shown 2.7× better hardware utilization compared to the implementations of Jiang et al. in [57]. The proposed LDPC decoder delivers a HUE of 1.95 which is 8.3× more efficient than the contemporary work of [41]. Subsequently, Table 4.2 shows that the PTLR of the proposed decoder architecture is 7.4× better than the highest value reported in [46]. Thus, our QC-LDPC decoder is a hardware-efficient design that delivers higher throughput and it is comparable to state-of-the-art implementations.

3.5 Summary

This chapter presented a new LLRC-segregation based OMS decoding algorithm for QC-LDPC codes that is compliant to the 5G-NR wireless communication standard and also resilient for 4G-LTE applications like WiFi, WiMAX, etc. Here, all CN and VN operations are performed in vector and matrix forms, which alleviates the hardware routing as well as the data-congestion in the QC-LDPC decoder architecture. Further, a comprehensive FER performance analysis of the LLRC-segregation based OMS decoding algorithm has been presented in this chapter.

Subsequently, a generic and detailed QC-LDPC decoder architecture was suggested based on the LLRC-segregation technique. This decoder architecture has been implemented on a standard base graph B matrix that has a size of 46×68 with a maximum expansion factor (z) of 384. It supports standard code-lengths ranging between 10368-26112 bits and also decode various 5G-NR standardized code-rates of QC-LDPC code. A detailed VLSI architecture of various sub-modules like MSR, CVPU, and MWR has been presented in this chapter. The MSR and MWR modules were designed based on the LLRC-segregation mapping technique that enhances hardware-efficiency and reduces decoding-latency. Hence, our proposed QC-LDPC decoder architecture has been designed with 5-bits internal LLRs and 7-bits AP-LLRs quantization formats.

Further, this LLRC-segregation based overall QC-LDPC decoder architecture has been FPGA prototyped on Xilinx Zynq Ultrascale+. This 5G-NR based QC-LDPC decoder architecture has a maximum clock frequency of 135 MHz and delivers a peak data-throughput of 11.02 Gbps while decoding at a code rate of 8/9. The proposed QC-LDPC decoder architecture has achieved a hardware utilization efficiency of 1.95 hardware-resources/layer/Mbps and renders the highest peak data-throughput to latency ratio of 11723.3 Mbps/μs. Hence, this QC-LDPC decoder architecture supports all the standard code-rates and also satisfies the 5G-NR physical layer specifications. The hardware consumption of the LLRC-segregation based OMS decoding algorithm is high due to its higher computational-

complexity. Hence, a new simplified offset min-sum (SOMS) decoding algorithm has been presented in the next chapter. Consequently, a novel QC-LDPC decoder architecture is designed based on the SOMS decoding algorithm. This SOMS QC-LDPC decoder architecture is compatible with 5G-NR systems, facilitates lower hardware-consumption, and increases the overall data-throughput.

Chapter 4

SOMS QC-LDPC Decoding Algorithm and Decoder Architecture

4.1 Introduction

As discussed earlier, 5G-NR technology offers high-speed data communication, lower-latency, high spectral-efficiency, improved reliability and energy-efficient devices. Therefore, it is essential to fulfill all the requirements for the 5G-NR standard and develop applications that support all the new specifications. To support all the 5G-NR applications such as eMBB, URLLC, and mMTC, error correction codes are playing a vital role for reliable communication networks. Hence, efficient QC-LDPC decoding algorithms and architectures are profoundly important for such high-speed and low decoding-latency applications.

In chapter 3 of this thesis, a LLRC-segregation based OMS decoding algorithm has been presented in section 3.2. This algorithm reduced the error floor phenomenon and routing-complexity in the QC-LDPC decoder and allowed it to achieve higher data-throughput with improved hardware-efficiency. Consequently, the computational-complexity of LLRC-segregation based OMS decoding algorithm is eminent that limits this technique for high-

level standards. There are too many computational operations like comparisons, additions, and memory storage to decode QC-LDPC codes in the LLRC-segregation based decoding algorithm. Due to its higher computational-complexity, the LLRC-segregation based algorithm is tedious and requires more hardware utilization. However, our major focus is to design a new hardware-friendly decoding algorithm that alleviates the computational-complexity without degrading the decoding performances.

In this chapter, we present a novel simplified offset min-sum (SOMS) decoding algorithm for QC-LDPC codes that alleviates the computational-complexity with the same FER performance. The suggested approach focuses on the faster accessibility for the data processing of LLRs between CNs and VNs at higher code-lengths with multiple code-rates. This algorithm synchronizes and simplifies the computations carried out by CNs and VNs. The proposed SOMS decoding algorithm decodes the QC-LDPC code that is compliant to the 5G-NR wireless communication standard whereas it can also be used for 4G-LTE applications. In addition, this algorithm primarily mitigates the problem of data-overcrowding while performing the selection of LLRs from the memory bank and storing updated LLRs back into the memory. In addition, a comprehensive and comparative FER performance analyses for the SOMS and existing decoding algorithms have been presented in this chapter.

Further, we proposed a new QC-LDPC decoder architecture based on the SOMS decoding algorithm that is compliant to 5G-NR wireless communication systems. This decoder is primarily designed based on the 5G-NR standard base-graph matrix with a maximum expansion factor. Eventually, our QC-LDPC decoder delivers the high data-throughput, lower decoding-latency, and reduces the routing-complexity with lower computational operations based on the suggested SOMS decoding algorithm. In addition, this SOMS based QC-LDPC decoder has been synthesized, placed and routed on Xilinx Zynq Ultrascale+ FPGA platform. Here, we performed a real-world test validation for QC-LDPC decoder and verified the decoded bits with the MATLAB simulation results. Finally, an ASIC synthesis and post-layout simulations of SOMS based QC-LDPC decoder in UMC 90 nm-CMOS technology node have been carried out in this chapter. The key highlights of this chapter

are follows as:

- A hardware-friendly SOMS decoding algorithm with layered scheduling has been proposed to decode QC-LDPC codes. This algorithm alleviates the computational-complexity of the QC-LDPC decoding and renders faster accessibility for the data processing of LLRs between CNs and VNs at higher code-lengths and multiple code-rates. Various techniques have been incorporated into the proposed LDPC decoding algorithm viz. memory selection technique (MST), cyclic rotational technique (CRT), and memory writing technique (MWT). They primarily mitigate the problem of data-overcrowding while performing the selection of LLRs from the memory bank and storing updated LLRs back into the memory.

- A Monte-Carlo simulation has been performed for the proposed SOMS decoding algorithm. Subsequently, a comprehensive FER performance analysis of the proposed QC-LDPC decoding algorithm is shown in this chapter. Further, performance comparison plots of our proposed SOMS with other state-of-the-art decoding algorithms has been illustrated in the upcoming sections of this chapter.

- Corresponding to the proposed SOMS algorithm, a new architecture of the QC-LDPC decoder and its submodules are presented in this chapter. These architectures are designed based on the IMT-2020 physical-layer specifications of the 5G-NR standard. Additional optimizations of these architectures have been carried out to reduce the routing-complexity, lower the decoding-latency, and improves data-throughput of the proposed QC-LDPC decoder.

- Comprehensive hardware utilization of the proposed decoder architecture has been presented in this chapter. Consecutively, hardware implementation of our decoder architecture has been performed in the FPGA platform and its results are compared with the relevant state-of-the-art implementations. Eventually, FPGA-prototype based real-world hardware testing for functional verification of the proposed QC-LDPC decoder has been carried out. An ASIC synthesis and post-layout simulation of the

proposed QC-LDPC decoder architecture have been carried out in UMC 90nm-CMOS technology. All the results of our ASIC design have been analyzed and compared with the contemporary implementations from the literature.

4.2 Simplified Offset Min-Sum Decoding Algorithm

This section commences with the presentation of an implementation-friendly QC-LDPC decoding algorithm that is based on offset min-sum decoding with layered scheduling and it is referred to as the SOMS algorithm in this chapter. Its mathematical formulation and corresponding flow diagram of the proposed SOMS decoding algorithm are presented in Algorithm 3 and Fig. 4.1, respectively. Here, the received LLRs in L vector are initially divided and stored in a-priori log-likelihood-ratios (A-LLRs) $\mathbb{A}$ matrix where every row of this matrix has z number of LLRs. Such LLR sorting representation alleviates the computational-complexity of the decoding algorithm and simplifies its implementation, which will be clarified later in the upcoming section 4.5.2. As discussed earlier in section 2.3, the conventional QC-LDPC decoding algorithm updates CNs and VNs element-wise which is a rigorous process for higher code-lengths that extend with the huge-sized B matrices of QC-LDPC codes for 5G-NR standard [1, 58].

Thus, exacerbating the data-flow synchronization between CNs and VNs leads to higher computational-complexity and data-congestion from an implementation perspective. The proposed SOMS algorithm alleviates such ramifications with the aid of a new simplified LLR-grouping technique that synchronizes the computations of CNs and VNs in matrix format. As a result, it inculcates lower computational-complexity, simplifies data-formation for CNs & VNs, and simplifies the CN's computation. The suggested SOMS algorithm has been segregated into various sub-tasks (i.e. presented in Algorithms: 3 – 6) and their detailed descriptions are presented in the remaining parts of this section.

1) **Initialization**: This sub-task of the proposed SOMS algorithm has been represented by lines 3–8 in Algorithm 3. At first, the received LLRs (i.e. $l_1, l_2, l_3 \cdots l_n$) of L vector are

Algorithm 3 Proposed SOMS Decoding Algorithm.

Inputs: Quantized-LLRs $(L) = \{l_1, l_2, l_3, \ldots, l_n\}$;
Output: Decoded bits $(\widehat{X}) = \{\chi_1, \chi_2, \chi_3, \ldots, \chi_n\}$;
Initialization:
$\mathbb{A}_{n_b \times z} = 0; \quad E_{T \times z} = 0 \; ; \quad I = 0 \; ;$
for $j = 1$ to n_b
  $\alpha = (j-1) \times z + 1 \; ; \quad \gamma = j \times z \; ;$
  $A(j) = [l_\alpha : l_\gamma] \; ;$ ▷ **Grouping of z LLRs**
end for$^{(j)}$
Message exchange between CNs and VNs iteratively:
for $I = 0$ to $(i_{max} - 1)$
  $\gamma = 0 \; ;$
  for $r = 1$ to m_b
    $P_{1 \times t_b} = [p_1, p_2, p_3, \ldots, p_{t_b}] \; ;$
    $S_{t_b \times z} =$ **MST (**$\mathbb{A}_{n_b \times z}, P_{1 \times t_b}$**)** $;$ ▷ **Algorithm 4**
    $\alpha = \gamma + 1 \; ; \quad \gamma = \alpha + t_b - 1 \; ;$
    $e_{t_b \times z} = E[(\alpha : \gamma); (1 : z)] \; ;$ ▷ **Sub-matrix**
    $V_{t_b \times z} = S_{t_b \times z} - e_{t_b \times z};$
    $W_{1 \times t_b} = [w_1, w_2, \ldots, w_{t_b}] \; ;$
    $e_{t_b \times z} =$ **CRT (**$V_{t_b \times z}, W_{1 \times t_b}$**)** $;$ ▷ **Algorithm 5**
    $R_{z \times t_b} = [e_{t_b \times z}]^\tau \; ;$
    for $j = 1$ to z
      $min1 = \min\left(|R(j)|_{1 \times t_b}\right) \; ;$
      $idx = \zeta(min1).|R(j)|_{1 \times t_b} \; ;$ ► $\zeta(\cdot)$ **function:** Determines the index of $min1$ from $|R(j)|_{1 \times t_b}$ vector.
      $min2 = \min\left(|R(j) \backslash R(j, idx)|\right) \; ;$
      $D_{1 \times t_b} = \text{sign}\left(R(j)_{t \times t_b}\right) \; ; \quad m = \prod_{k=1}^{t_b} D_k \; ;$
      $min1 = \max\{(min1 - \beta), 0\} \; ;$
      $min2 = \max\{(min2 - \beta), 0\} \; ;$
      $C(j)_{1 \times t_b} = min1 \; ; \quad C(j, idx) = min2 \; ;$
      $C(j) = (m \times D_{1 \times t_b}) \circ C(j)_{1 \times t_b};$ ▷ **Hadamard product**
    end for$^{(j)}$
    $e_{t_b \times z} = [C_{z \times t_b}]^\tau;$
    $Re_{t_b \times z} =$ **CRT (**$e_{t_b \times z}, (z - W_{1 \times t_b})$**)**; ▷ **Algorithm 5**
    $E[(\alpha : \gamma); (1 : z)] = Re_{t_b \times z};$
    $a_{t_b \times z} = Re_{t_b \times z} + V_{t_b \times z};$
    $\mathbb{A}_{n_b \times z} =$ **MWT(**$\mathbb{A}_{n_b \times z}, a_{1 \times t_b}, P_{1 \times t_b}$**)**; ▷ **Algorithm 6**
  end for$^{(r)}$
  $\widehat{v}_{1 \times n} = [A(1), A(2) \ldots A(n_b)] = [v_1, v_2 \cdots v_n];$
  $\chi_i = \begin{Bmatrix} 0 & v_i \geq 0 \\ 1 & otherwise \end{Bmatrix}, \quad v_i \in \widehat{v}, \quad \chi_i \in \widehat{X}$ ▷ **Hard-Decision**
  $Syn = \widehat{X} \cdot H^\tau$ ▷ **Syndrome Check**
end for$^{(i)}$
$\widehat{X}_{1 \times n} = \{\chi_1, \chi_2, \chi_3, \ldots, \chi_n\}.$

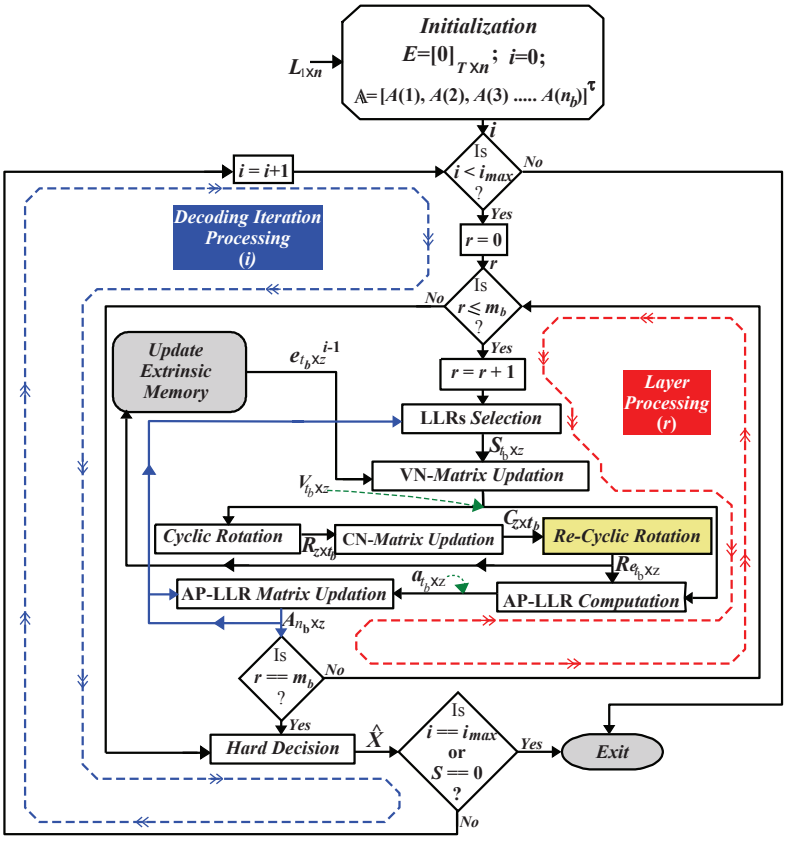


Figure 4.1: Flow diagram representing the data flow of proposed SOMS QC-LDPC decoding algorithm with layered scheduling.

stored in A-LLR $\mathbb{A} \in \mathbb{C}^{n_b \times z}$ matrix that incorporates $B \in \mathbb{C}^{m_b \times n_b}$ BG-matrix (referred from Fig. 1.7) where $m_b = m/z$ and $n_b = n/z$. Here, z number of LLRs are placed in every row of $\mathbb{A}$ matrix and each r^{th} row of AP-LLR matrix is represented as $A(r)$ row-vector. Hence, $\mathbb{A}$ matrix comprises of n_b such LLR groups which correspond to n_b columns of B matrix. Furthermore, an extrinsic-LLR matrix $\mathrm{E} \in \mathbb{C}^{T \times z}$ has been initialized as a zero matrix where T is the number of all 'non -1' elements in the entire B matrix. Eventually, an iteration variable I is reset and is later incremented by one after every decoding iteration.

2) **LLRs Selection and VN-matrix Updation**: This subtask has been segregated into two processes: LLRs selection and VN-matrix updation, as shown in Fig. 4.1. In the former process, $P \in \mathbb{C}^{1 \times t_b}$ index vector is computed that comprises the indices of 'non -1' elements (denoted by $p_1, p_2, p_3 \cdots p_{t_b}$ in line 13 of Algorithm 3) in ongoing r^{th} row of B matrix where

t_b refers to the number of 'non -1' elements in such $B(r)$ row-vector. For example, if a row-vector $B(3) = \{23, 67, -1, 0, -1, 4, 3\}$ then the value of $t_b = 5$ and the index vector $P_{1\times5} = \{1, 2, 4, 6, 7\}$ which excludes the indices 3 and 5 of '-1' elements in $B(3)$ vector. Such P index-vector and $\mathbb{A}$ matrix are processed by the proposed memory selection technique (MST) for computing $S \in \mathbb{C}^{t_b \times z}$ selection-matrix, as illustrated in line 14 of Algorithm 3 that is separately presented in Algorithm 4. This technique simplifies the mathematical computations and lowers the hardware-complexity for determining the S selection matrix that is explained in the next section. In the VN-matrix updation process, sub-matrix $e \in \mathbb{C}^{t_b \times z}$ represents α to γ row-vectors of E matrix. Subsequently, the VN-matrix updation $V \in \mathbb{C}^{t_b \times z}$ is generated by subtracting e sub-matrix from S selection matrix. Hence, aforementioned LLRs selection and VN-matrix updation subtask are represented between lines 13 – 17 in Algorithm 3.

Algorithm 4 Memory Selection Technique (MST).

1: **Inputs:** $\mathbb{P}_{n_b \times z}$, $\mathbb{Q}_{1\times t_b} = \{q_1, q_2, q_3, \ldots, q_{t_b}\}$;
2: **Output:** $\mathbb{S}_{t_b \times z}$;
3: **Initialization:**
4: $\mathbb{T}_{t_b \times z} = [0]_{t_b \times z}$;
5: **Row updation:**
6: **for** $i = 1$ to t_b
7: $\mathbb{T}(i) = \mathbb{P}(q_i)$; ▷ **Row-updation**
8: **end for**
9: $\mathbb{S}_{t_b \times z} = \mathbb{T}_{t_b \times z}$.

3) **LLRs Rotation**: The cyclic rotational technique (CRT) has been proposed in our work to perform the LLR-rotation subtask which is presented in Algorithm 5 and indicated by line 19 in Algorithm 3. At first, the weighted-vector $W \in \mathbb{C}^{1\times t_b}$ is generated that contains all 'non -1' elements of r^{th} row vector of B matrix. For example, if row vector $B(3) = \{23, 67, -1, 0, -1, 4, 3\}$ then $W_{1\times5} = \{23, 67, 0, 4, 3\}$. In line 7 of Algorithm 5, the '>>>' operator represents circular right shift in any $\mathbb{P}(i)$ row-vector. For example, if $\mathbb{P}(2) = \{9, -7, 4, -2, 6, -3, 8\}$ then $\mathbb{P}(2) >>> 3$ will result $\{6, -3, 8, 9, -7, 4, -2\}$. Such LLRs rotation subtask has been expressed by lines 18 – 20 in Algorithm 3 where $e \in \mathbb{C}^{t_b \times z}$ is computed

using CRT on the VN-updation matrix (V) and weighted-vector (W). Eventually, a rotational matrix $R \in \mathbb{C}^{z \times t_b}$ stores the transpose (τ) of CRT-computed e matrix.

Algorithm 5 Cyclic Rotational Technique (CRT).

1: **Inputs:** $\mathbb{P}_{t_b \times z}$, $\mathbb{Q}_{1 \times t_b} = \{q_1, q_2, q_3, \ldots, q_{t_b}\}$;
2: **Output:** $\mathbb{S}_{t_b \times z}$;
3: **Initialization:**
4: $\mathbb{T}_{t_b \times z} = [0]_{t_b \times z}$;
5: **Row rotation and updation:**
6: **for** $i = 1$ to t_b
7: $\mathbb{T}(i) = \mathbb{P}(i) >>> q_i$; ▷ **Circular right-shift**
8: **end for**
9: $\mathbb{S}_{t_b \times z} = \mathbb{T}_{t_b \times z}$.

4) **CN and Extrinsic Matrix Updation**: In Algorithm 3, the CN updation has been row-wise processed on R rotational matrix, as shown in lines 21 – 30. Here, all the elements of j^{th} row-vector $R(j)$ are replaced by '$min1$' value, except in idx index of this row vector where '$min2$' value is assigned as an element, as illustrated in lines 22 – 29 of Algorithm 3. Here, β is an offset value that depends on the quantization bits format under the fixed point representation [40, 56]. In our work, β has been assigned an optimized value of '2' based on the extensive FER performance analysis that has been presented in section 4.4. These mathematical computations of CN-updation alleviate the implementation complexity of CN-unit architecture and also reduce the size of extrinsic memory which is further explained in section 4.5.2. In the extrinsic memory updation, from lines 31 – 33 of Algorithm 3, the transpose of $C_{t_b \times z}$ matrix is again cyclically-rotated by $(z - W_{1 \times t_b})_{1 \times t_b}$ vector using CRT, represented in Algorithm 5. Finally, the output matrix $Re_{t_b \times z}$ generated by CRT is stored along $\alpha - \gamma$ rows of extrinsic memory matrix E for the next decoding-iteration, as presented in line 33 of the Algorithm 3 and shown in Fig. 4.1.

5) **AP-LLR Computation and Matrix Updation**: This subtask is exclusively performed by the suggested memory-writing-technique (MWT), as presented in Algorithm 6. To begin with, $a \in \mathbb{C}^{t_b \times z}$ matrix is computed by adding $Re_{t_b \times z}$ and VN $V_{t_b \times z}$ matrices. Such a matrix and index-vector P are used for updating the AP-LLR matrix $\mathbb{A}$ (using the MWT)

Algorithm 6 Memory Writing Technique (MWT).

1: **Inputs:** $\mathbb{P}_{n_b \times z}$, $\mathbb{Q}_{t_b \times z}$, $\mathbb{R}_{1 \times t_b} = \{r_1, r_2, r_3, \ldots, r_{t_b}\}$;
2: **Output:** $\mathbb{S}_{n_b \times z}$;
3: **Initialization:**
4: $\mathbb{T}_{n_b \times z} = \mathbb{P}_{n_b \times z}$;
5: **Row updation:**
6: **for** $i = 1$ to t_b
7: $\mathbb{T}(r_i) = \mathbb{Q}(i)$; ▷ **Row-updation**
8: **end for**
9: $\mathbb{S}_{n_b \times z} = \mathbb{T}_{n_b \times z}$.

which is expressed by lines 34 and 35 in Algorithm 3. This updated AP-LLR matrix $\mathbb{A}$ is subsequently used for the computation of hard-decision vector $\widehat{X}_{1 \times n}$, after accomplishing all the decoding iterations (i.e. when $I = i_{max}$).

6) **Hard Decision and Syndrome Test**: Eventually, the hard-decision vector $\widehat{X}$ is generated after the conversion of AP-LLR matrix $\mathbb{A}$ into a AP-LLR vector $\widehat{v}_{1 \times n}$. Such AP-LLR vector $\widehat{v}_{1 \times n}$ is used for the computation of estimated hard-decision vector $\widehat{X}_{1 \times n}$ which is performed in lines 37 and 38 of Algorithm 3. Thereafter, the syndrome $Syn = \widehat{X} \cdot \mathbf{H}^T$ is checked and if $Syn = 0$ then the decoding process is terminated else it continues till the maximum iteration i_{max} is reached, as clearly shown in Fig. 4.1.

4.3 Computational-Complexity Analysis of SOMS Decoding Algorithm

Here, the computational-complexity of the proposed SOMS decoding algorithm has been estimated and compared with the computational-complexities of other QC-LDPC decoding algorithms. Assume that the average CN and VN degrees of parity-check-matrix H are represented by d_c and d_v, respectively. computational-complexity and memory requirements of the various QC-LDPC decoding algorithms are quantified in Table 4.1. It shows the analysis of computational loads of all the decoding algorithms for a single decoding iteration. As illustrated in Table 4.1, improved SP and adjusted MS algorithms require

Table 4.1: Comparison of Computational Complexities Incurred by Various QC-LDPC decoding algorithms.

Algorithms	Special Functions	Comparisons	Multiplications	Addition	Memory
2D-SC MS [59]	–	$(2{\cdot}d_c{-}3){\cdot}m + d_v{\cdot}k$	–	$d_v{\cdot}n$	$2{\cdot}m + n + d_v{\cdot}k$
Adjusted MS [60]	$\ln^{\mathfrak{R}}$, $\mathrm{Coth}^{\wp}$	$(2{\cdot}d_c{-}3){\cdot}m$	–	$d_v{\cdot}\ n{+}2(d_c{-}2){\cdot}m$	$3{\cdot}m + n$
Improved SP [14]	$\mathrm{FFT}^{\ddagger}$, $\mathrm{IFFT}^{\S}$	–	$(d_c - 2){\cdot}m$	$2{\cdot}d_v{\cdot}m$	$d_c{\cdot}m + n$
Min-Sum (MS) [16]	–	$2{\cdot}m{\cdot}d_c{-}1$	n	$2(d_c{-}1){\cdot}m$	$2{\cdot}m + n$
Normalized MS [17]	–	$(2{\cdot}d_c{-}3){\cdot}m$	$2{\cdot}m$	$(2{\cdot}d_c{-}1){\cdot}m$	$2{\cdot}m + n$
Proposed SOMS	–	$(2{\cdot}d_c{-}3){\cdot}m$	–	$2{\cdot}d_v{\cdot}n$	$2{\cdot}m + n$

- $\mathfrak{R}$: Natural Logarithm; $\wp$: Hyperbolic Cotangent Trigonometry Function; ‡: Fast Fourier Transform; §: Inverse Fast Fourier Transform.

special operations to be performed. Unlike, the proposed SOMS algorithm does not require such special operations. Furthermore, our SOMS algorithm incurs $d_v{\cdot}k$ lesser comparisons than the 2D self-corrected (SC) decoding algorithm [59]. However, the SOMS algorithm requires more additional operations than 2D-SC decoding algorithms. In Table 4.1, the memory requirement of the SOMS algorithm has been expressed by summing the memory consumed by n received LLRs and the extrinsic-memory storage. Therefore, the total memory requirement of the SOMS algorithm is $2{\cdot}m + n$ which is comparatively lower than improved-SP, adjusted MS, and 2D-SC min-sum decoding algorithms[60, 59], as listed in Table 4.1.

Comparative plots of the computational-complexities for proposed SOMS and other decoding-algorithms with varying m values are shown in Fig. 4.2, 4.3, and 4.4. They are obtained for the code rate of $1/3$ and fixed VN as well as CN degrees (i.e. d_c=8 and d_v=5). The variations of comparisons, additions, and memory consumptions of decoding algorithms have been plotted while scaling the m value between 100−30000 bit [58]. It can be observed that our SOMS algorithm incurs a significant reduction in overall computational-complexity with respect to the state-of-the-art decoding algorithms. Specifically, Fig. 4.2 shows the comparison computations of various decoding-algorithms where the SOMS algorithm requires lesser comparison operations than the state-of-the-art 2D-SC MS decoding-algorithm [59]. Similarly, the variations of addition computations needed by different algorithms are

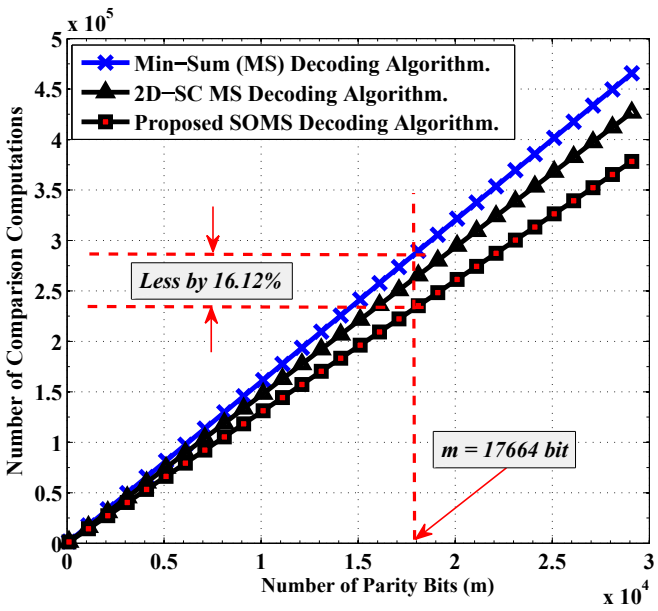


Figure 4.2: Comparative analysis of computational complexities among the proposed SOMS and other decoding algorithms in terms of comparison computations with the increasing m value.

presented in Fig. 4.3. Here, the suggested SOMS algorithm consumes more additions than the 2D-SC MS decoding algorithm and lesser additions than the MS algorithm [16]. Further, plots of memory requirements for different decoding algorithms are represented in Fig. 4.4. From Table 4.1, the proposed SOMS algorithm needs n memory storage for channel-LLRs, and two minimum LLRs, corresponding to each of the check-nodes which is referred to as extrinsic-LLR storage. Thus, the overall memory required by SOMS algorithm is $2 \cdot m + n$ and it is lesser compared to other decoding algorithms. Note that the maximum m value for the 5G-NR standard has been specified to be 17664 bit for a lower code rate of 1/3 [58]. Hence, for m = 17664 bit, our SOMS decoding algorithm consumes ≈16% lesser comparison computations, 25% more addition operations, and ≈41% lesser memory storage, compared to the state-of-the-art 2D-SC MS decoding algorithm [59]. Aforementioned analysis shows that the aggregated percentage of reductions in comparison computations and memory requirement is greater than the percentage of increment in addition operations. Therefore, the proposed SOMS algorithm delivers alleviated computational-complexity in comparison

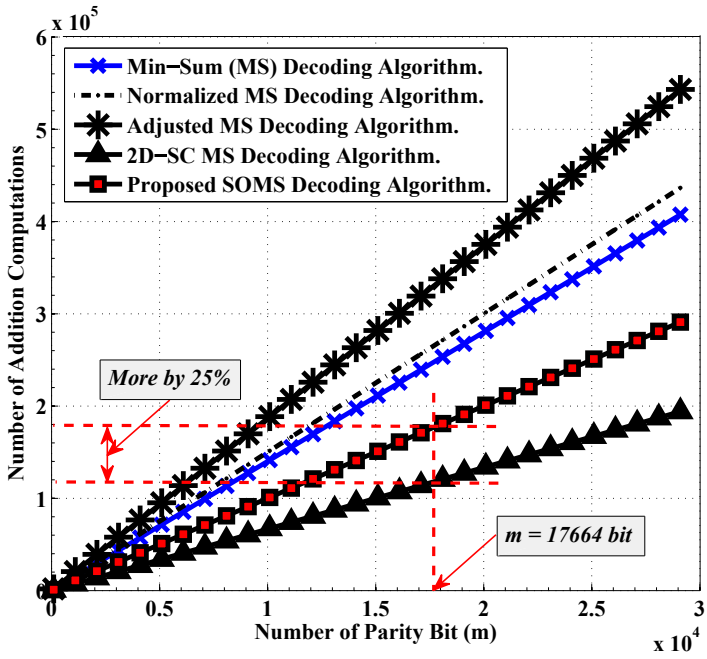


Figure 4.3: Comparative analysis of computational complexities among the proposed SOMS and other decoding algorithms in terms of addition operation with the increasing m value.

to other decoding-algorithms.

4.4 Performance Analysis

This work presents the comprehensive performance analysis of the proposed SOMS decoding algorithm from Algorithm 3 based on the specifications that are compliant to the 5G-NR standard [1, 58], as discussed in section 4.2. For each SNR value, an extensive Monte Carlo simulation has been performed to decode $\approx 10^6$ transmitted bit, using a BPSK modulation scheme, via AWGN channel environment. The FER performance analysis of the proposed SOMS algorithm for different bit quantization of input LLRs has been shown in Fig. 4.5 (a). These multiple plots from fixed point simulations are obtained for the bit width range of 5−13 bit while decoding at minimum, moderate, and maximum code-rates of 1/3, 3/4, and 8/9, respectively, for 10 decoding iterations. In Fig. 4.5 (a), the FER performance of our decoding algorithm gradually improves with the increasing bit width values. However,

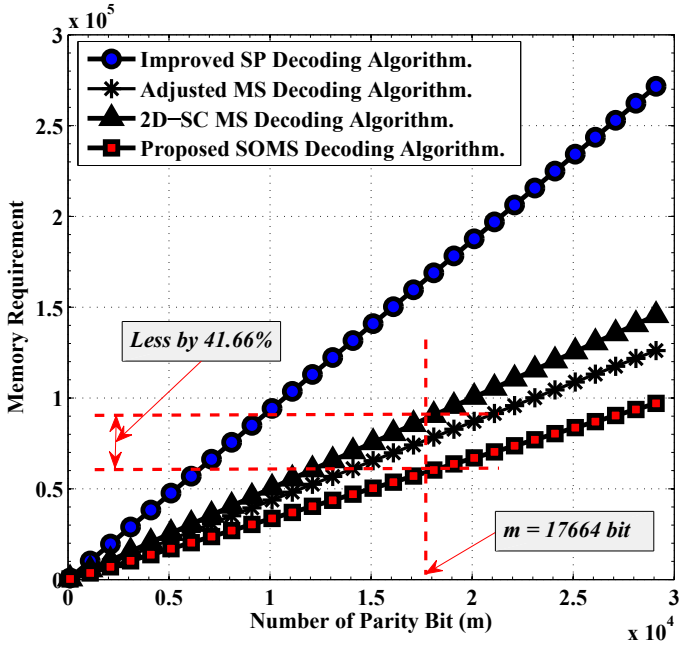


Figure 4.4: Comparative analysis of computational complexities among the proposed SOMS and other decoding algorithms in terms of memory requirement with the increasing m value.

such upscaling of bit width adversely increases the hardware requirements and power consumption of the proposed decoder architecture. Nevertheless, our decoding algorithm with 7-bit quantization delivers an adequate FER of 10^{-5} at 0.9 dB of SNR for a code rate of 1/3, as shown in Fig. 4.5 (a).

Furthermore, Fig. 4.5 (b) shows the comparative FER performance analysis between proposed and conventional [19] QC-LDPC decoding algorithms for all the code-rates of the 5G-NR standard. These code-rates range between 1/3 to 8/9 and are compliant with the $B1$ base graph matrix of 5G-NR standard [1, 58]. In Fig. 4.5 (b), these FER plots of different code-rates 1/3, 2/5, 1/2, 2/3, 3/4, 5/6, and 8/9 are obtained for the 5G-NR specified maximum code-lengths (in bits) of 26112, 21888, 17664, 13440, 12288, 11136, and 10368, respectively, for an expansion factor of z=384. In comparison with the conventional algorithm [19], the proposed SOMS decoding algorithm delivers a minimum coding loss of 0.025 dB and maximum coding loss of 0.34 at 10^{-4} FER for 1/3 and 8/9 code-rates,

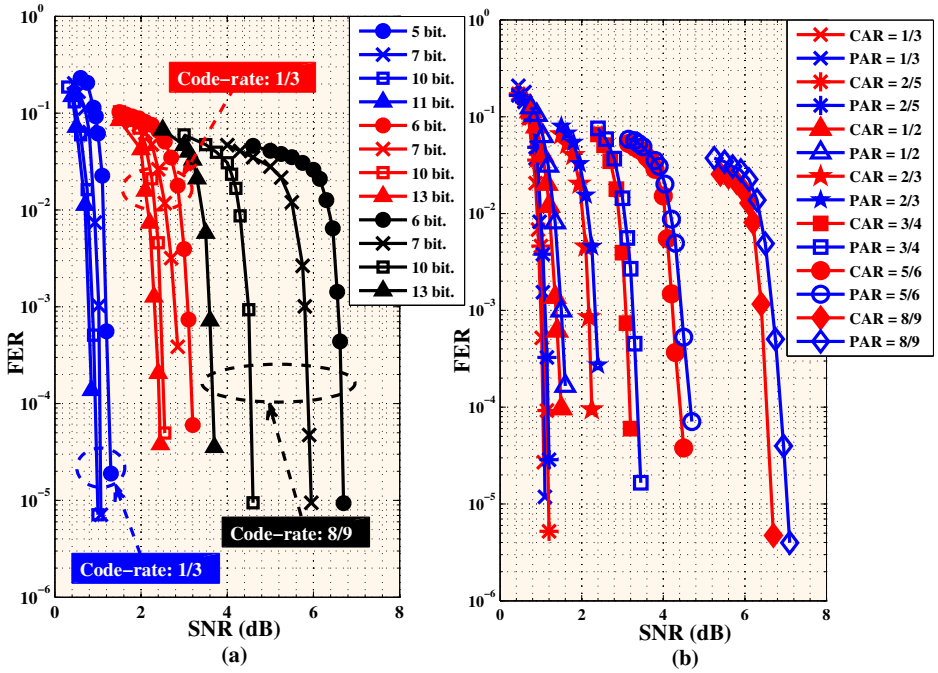


Figure 4.5: (a) Fixed-point FER performance-analysis of proposed SOMS decoding algorithm for various bit-quantization. (b) Comparative FER performance-analysis of proposed-SOMS and conventional QC-LDPC decoding algorithms. Here, CAR: conventional-algorithm-rate and PAR: proposed-algorithm-rate.

respectively. The proposed SOMS decoding-algorithm is hardware-friendly in nature and incurs lower computational-complexity with minor coding loss and it consequently enables the design of hardware-efficient architecture for LDPC decoder.

Comprehensive and comparative FER performance analysis of the proposed SOMS decoding algorithm with the various standard decoding-algorithms has been presented in Fig. 4.6 (a). Such analysis is carried out at a code rate of 1/3 for 10 decoding iterations. It is to be noted that the suggested SOMS algorithm shows minor coding losses of 0.020 dB, 0.032 dB, 0.025 dB with respect to MS [16], normalized MS [17], and offset MS [13] decoding algorithms at the FER of 10^{-4}, respectively, as illustrated in Fig. 4.6 (a). Subsequently, Fig. 4.6 (b) presents the FER versus SNR plots of our SOMS decoding algorithm at various offset (β) values that depend on the LLR quantization bit under the integer representations [40, 56].

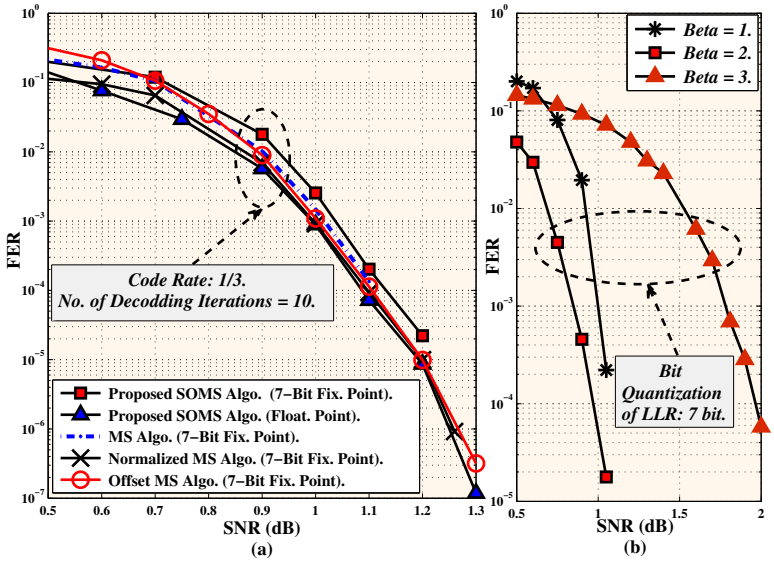


Figure 4.6: (a) Comparative FER performance analysis of the proposed SOMS algorithm with respect to the existing standard algorithms in the fixed point bit-quantization of 7 bit. (b) FER versus SNR plots of SOMS algorithm for various offset β values.

Note that the β value increases with the LLR quantization bits. Here, β is mathematically computed as $\beta = \mathbb{E}[C_{BP}] - \mathbb{E}[C_{MS}]$ where $\mathbb{E}[C_{BP}]$ and $\mathbb{E}[C_{MS}]$ represent expectations of CN updated messages for BP and MS algorithms, respectively. Furthermore, an optimized value of β must be obtained that allows the SOMS algorithm to deliver adequate FER performance for a specific fixed point format. Thereby, Fig. 4.6 (b) shows that an optimum FER performance of the SOMS algorithm has been achieved at β=2 for 7-bit quantization format.

It is imperative to understand the effects of various modulation schemes on the FER performance of the decoding algorithm. Thereby, Fig. 4.7 (a) presents the FER performances of the proposed SOMS decoding algorithm (with 7-bit fixed-point quantization) for various modulation schemes at a code rate of $1/3$. It can be observed that our algorithm delivers a FER of 10^{-5} at 1.05 dB and 1.3 dB SNRs for BPSK and 16-QAM schemes, respectively. Subsequently, FER versus iteration plots of our decoding algorithm based on the aforementioned modulation schemes are presented in Fig. 4.7 (b). Here, the proposed SOMS algorithm

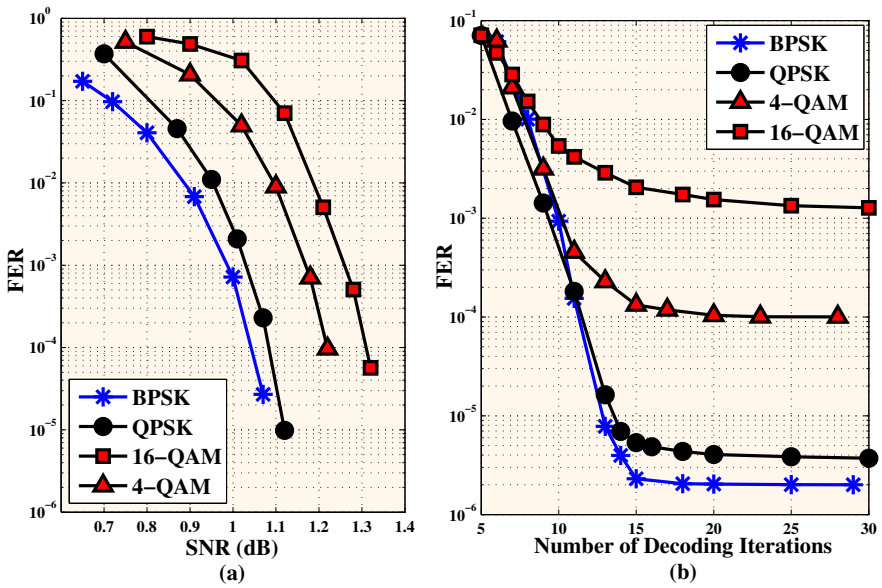


Figure 4.7: (a) Comparative FER performance-analysis of our SOMS decoding algorithm for various modulation schemes. (b) FER versus decoding-iterations plots of SOMS decoding algorithm for different modulation schemes. Note that QPSK and 16-QAM schemes are specified by the 5G-NR wireless communication standard.

delivers a constant FER of $10^{-5.8}$ beyond 15 decoding iterations for BPSK modulation.

Likewise, the 16-QAM scheme renders a constant FER of $10^{-2.9}$ beyond 18 iterations, as shown in Fig. 4.7 (b). The BPSK modulation technique having the fewest bits per symbol that offers best FER performance, while 16-QAM, 4-QAM, and QPSK, which involve multiple bits per symbol, introduce bit interference that degrades the FER performance.

These performance analyses are carried out for the maximum code length of 26112 bits with the highest expansion factor of z=384 at the fixed code rate of 1/3, compliant to the 5G-NR specification. In addition, there are four more code-lengths at 1/3 code rate: 22032 bit, 15232 bit, 11958 bit, and 8160 bit, with the corresponding z values of 324, 224, 176, and 120, respectively, that the LDPC decoder must support for the 5G-NR wireless-communication standard [1, 58]. Therefore, Fig. 4.8 presents the FER plots of the suggested SOMS decoding algorithm for such five code-lengths with varying expansion factors at a fixed code rate of 1/3 where the one with the longest code-length delivers superior performance among all.

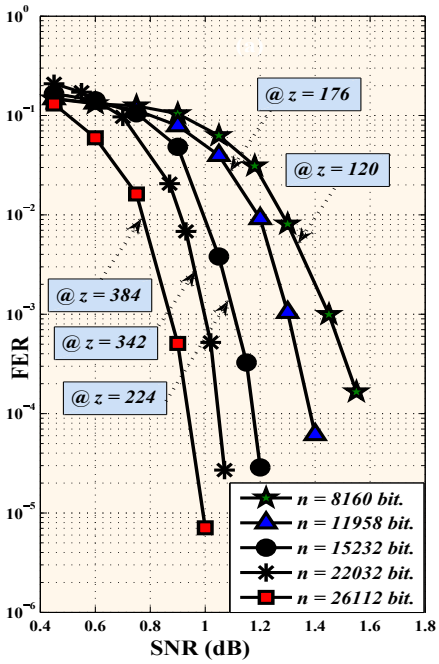


Figure 4.8: Comparative FER performance-analysis of the SOMS decoding algorithm for various code-lengths at different values of expansion factor (z).

4.5 SOMS based QC-LDPC Decoder Architecture

This section proposes a high-throughput QC-LDPC decoder architecture based on the suggested SOMS decoding algorithm. Here, designs of various decoder submodules are first presented and they are eventually aggregated into an overall QC-LDPC decoder architecture. Note that the design specifications of the proposed decoder architecture are compliant to the 5G-NR wireless communication standard [1, 58]. Standardization of QC-LDPC codes for 5G-NR is primarily based on two prototype BG matrices: $B1$ and $B2$. The expansion factor z for BG matrices has been quantified based on $z = a \times 2^j$ where $a \in \{2, 3, 5, 7, 9, 11, 15\}$ and $0 \leq j \leq 7$. Thus, peak value of z is 384 with $a = 3$ and $j = 7$, based on 5G-NR specification [2]. Therefore, maximum information bits supported by $B1$ matrix is 22×384 = 8448 bit (i.e. k = 8448 bit) with the code-rates (R) ranging between 1/3–8/9 [2]. Since the initial two columns of these BG matrices are extremely congested, they are punctured while encoding and are assigned zero values while decoding [58].

Such puncturing is performed on parity bits to support various code-rates of the 5G-NR standard. Therefore, the proposed QC-LDPC decoder architecture has been designed to decode the received channel-LLRs corresponding to the maximum value of code-length, which is n = 68×384 = 26112 bit (with 8448 information-bits) that is compliant to the $B1$ matrix and supports all the standard code-rates and other code-lengths of 5G-NR wireless communication standard.

4.5.1 Initialization and Memory Updating Unit

The quantized channel-LLRs $L = \{l_1, l_2, l_3 \cdots l_n\}$ are first processed by a submodule, referred as initialization & memory updating unit (IMUU), of the proposed QC-LDPC decoder. The suggested IMUU architecture has been presented in Fig. 4.9. It accomplishes two subtasks: initialization (presented in lines 3−8 of Algorithm 3) and AP-LLR matrix updation (referring line 35 in Algorithm 3 and Algorithm 6) of the proposed flow-diagram from Fig. 4.1. In IMUU architecture, 7-bit quantized-LLRs $l_i \; \forall \; i \in n$ are foremost fed to 1:384 de-multiplexer (DeMUX1) whose outputs are buffered by registers (*REG*s) for 384 clock cycles, as shown in Fig. 4.9. Subsequently, 384 LLRs at the outputs of *REG*s are concatenated (':') and denoted by $L_c = \{l_1 : l_2 : l_3 \cdots l_{384}\}$ that has the bit width of 2688 bit (i.e. 384×7-bit). These processes of buffering and concatenation for 384 clock cycles repeat 68× in IMUU architecture to obtain the initialized $\mathbb{A}$ matrix.

The buffered DeMUX1 output L_c is transferred to the data selection multiplexer network (DSMN) that decides whether to perform initialization or AP-LLR matrix updation subtask. Here, *REG*s at the DeMUX1-outputs sequentially deliver 68 L_c values one after the other, corresponding to 68 different columns of BG matrix (as shown in Fig. 1.7), to the DSMN that comprises of 19 parallel 2:1 multiplexers. Each of these 2:1 multiplexers is fed with L_c and updated AP-LLRs $a(i)$ (for i^{th} 2:1 multiplexer) values, as shown in Fig. 4.9. Here, every i^{th} multiplexer of DSMN routes either L_c or $a(i)$ to perform initialization or AP-LLR matrix updation, respectively. The DSMN outputs are referred as $D = \{d(1), d(2) \cdots d(19)\}$ and are processed by a memory multiplexer network (MMN). This MMN comprises 25 multiplex-

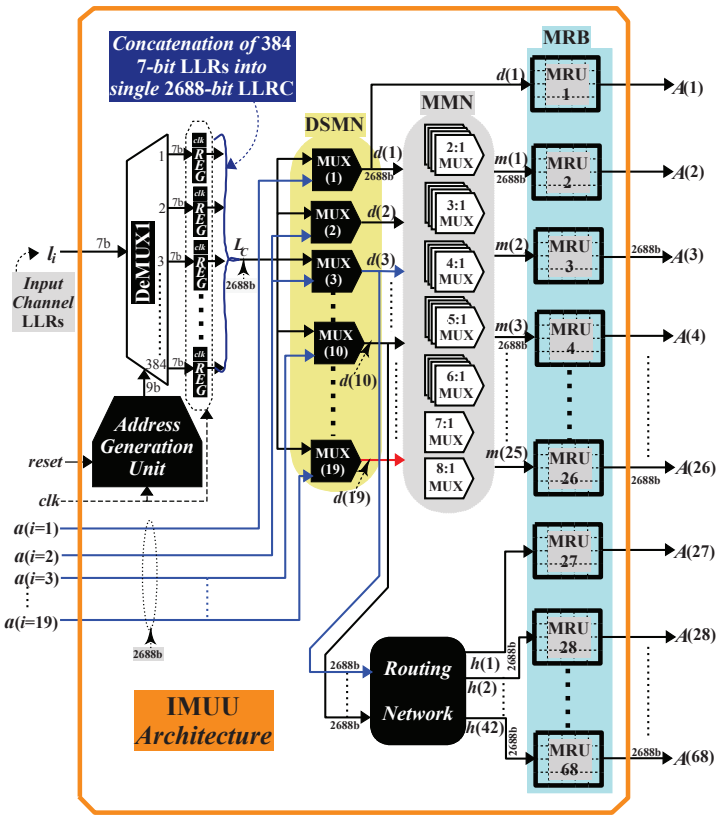


Figure 4.9: Proposed architecture of initialization & memory updating unit (IMUU) that executes initialization and VN-matrix updation (based on memory writing technique from Algorithm 6) tasks in Algorithm 3.

ers of various sizes ranging between 2:1−8:1 multiplexers to generate 25 LLR-groupings denoted by $M = \{m(1), m(2) \cdots m(25)\}$, as shown in Fig. 4.9. Furthermore, $d(3)-d(10)$ DSMN outputs are fed to the routing network which directly maps these eight signals to 42 $H = \{h(1), h(2) \cdots h(42)\}$ signals. Collectively, one $d(1)$, 25 M, and 42 H LLR groups (in total 68 LLR groups) are stored in 68 memory register units (MRUs) of the memory register bank (MRB). Here, each MRU has a size of 384×7-bits that stores 384 LLRs of 7-bit each, and thus, MRB has 68×384 = 26112 memory-locations for storing input channel-LLRs corresponding to 26112 QC-LDPC encoded-bits. Such 68 LLR groups are stored in 68 MRUs of MRB as shown in Fig. 4.9. It shows that all 68 outputs from MRB are denoted by $\mathbb{A} = \{A(1), A(2) \cdots A(68)\}$ and all these LLRs are generated in a single clock-cycle from IMUU

architecture. Therefore, the IMUU submodule provides the output data based on the high-parallelism technique that aids our QC-LDPC decoder to achieve high data-throughput and also experience lower data-congestion.

4.5.2 Memory Selection and Extrinsic-LLR Storage Memory

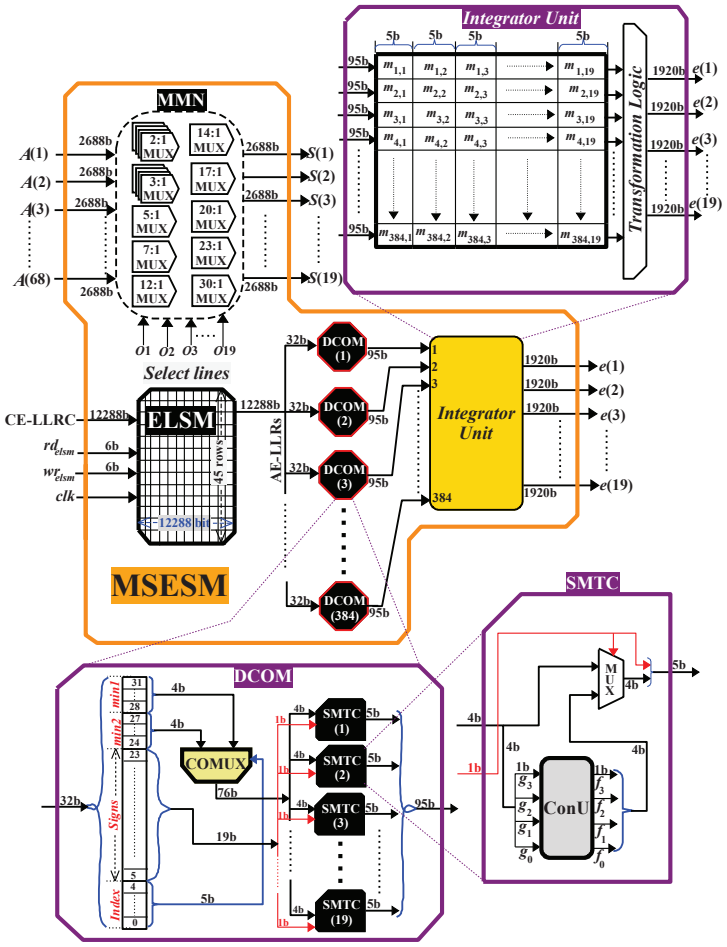


Figure 4.10: Proposed architectures of memory selection & extrinsic-LLR storage memory (MSESM), DCOM, SMTC, and integrator unit.

The proposed architecture of memory selection and extrinsic-LLR storage memory (MSESM) is presented in Fig. 4.10. It shows that 68 $\mathbb{A}(i)$ LLR groups from IMUU are fed to MMN of MSESM that incorporates 19 parallel multiplexers (MUXes) of various sizes between 2:1−30:1 multiplexers. They perform the LLRs selection subtask of the proposed

algorithm, as shown in Fig. 4.1. This subtask is mathematically represented as memory selection technique (MST) in Algorithm 4 that is invoked in line 14 of Algorithm 3. Each of these MMN MUXes routes specific combination of $\mathbb{A}(i)$ LLRs based on the B matrix columns to generate $S(j)\ \forall\ j = \{1, 2, 3, \cdots 19\}$ LLRs. In parallel, an extrinsic-LLR storage memory (ELSM) stores the E matrix in the compressed form of extrinsic-LLRs (e-LLRs) to enhance the hardware-efficiency of our decoder design. Each location of ELSM is reserved for the e submatrix that is used for VN-matrix updation subtask (referring to Fig. 4.1). Thus, ELSM has a depth of 46 memory locations corresponding to all 46 layers/rows of the $B1$ matrix [58] that is shown in Fig. 1.7. As shown in Fig. 4.10, ELSM architecture is fed with compressed extrinsic-LLRs (CE-LLRs) of 12288 bit (i.e. 384×32 bit) which is a concatenation of 384 aggregated extrinsic-LLRs (AE-LLRs) where each AE-LLR has a bit width of 32 bit (sign-magnitude format).

Hence, ELSM has a word-length of 12288 bits that are fetched at its output which is segregated into 384 different 32-bit AE-LLRs and are fed to 384 parallel decompression (DCOM) units, as shown in Fig. 4.10. Such AE-LLR has four parts: four most-significant-bits (MSBs) between 31−28 bit-positions represent the magnitude of *min1*, next four bits (27−24 bit-positions) hold *min2* value, 19 bit across 23−5 bit-positions correspond to signs of 19 eLLRs from CNs, and last five least-significant-bits (LSBs) of 4−0 bit-positions represent index for the placement of *min2*, as illustrated in Fig. 4.10 (*min1* and *min2* values will be discussed in the next subsections). The suggested architecture of the DCOM unit comprises a multiplexer (COMUX) and 19 parallel sign-magnitudes to two's-complement converters (SMTCs), as shown in Fig. 4.10. The 76-bit output of COMUX has 19 parts (of 4 bits each) that represent 18 replications of *min1* magnitudes and one *min2* magnitude in the specific pattern illustrated in (4.1).

$$\begin{pmatrix} min2 & min1_1 & min1_2 & \cdots & min1_{17} & min1_{18} \\ min1_1 & min2 & min1_2 & \cdots & min1_{17} & min1_{18} \\ min1_1 & min1_2 & min2 & \cdots & min1_{17} & min1_{18} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ min1_1 & min1_2 & min1_3 & \cdots & min2 & min1_{18} \\ min1_1 & min1_2 & min1_3 & \cdots & min1_{18} & min2 \end{pmatrix} \tag{4.1}$$

The selection among such states at the COMUX output depends on the 5-bit index value tapped from 32-bit AE-LLR. Earlier, this COMUX has been designed by using various equalizers, MUXes, and by-pass networks [32]. This COMUX unit renders the lower routing-complexity because it routes only two data signals (*min1* and *min2*). These 19 *min1/min2* magnitudes and their corresponding 19 sign-bit those are tapped from AE-LLR (between 23−5 bit-positions) are fed to 19 parallel SMTCs. Each SMTC converts the sign-magnitude value into its equivalent two's complement format using a 2:1 multiplexer and a conversion unit (ConU), based on the following simplified Boolean equations:

$$f_3 = \bar{g}_3 | (\bar{g}_2 \cdot \bar{g}_1 \cdot \bar{g}_0) \tag{4.2}$$

$$f_2 = (\bar{g}_3 \cdot \bar{g}_2) | (\bar{g}_2 \cdot g_1) | (\bar{g}_2 \cdot \bar{g}_0) | (g_2 \cdot \bar{g}_1 \cdot \bar{g}_0) \tag{4.3}$$

$$f_1 = g_1 \oplus g_0 \tag{4.4}$$

$$f_0 = g_0 \tag{4.5}$$

Every DCOM unit produces 95-bit (i.e. 19×5 bit) which is a concatenation of 19 e-LLRs where each e-LLR is represented in 5-bit two's-complement format. Thus, all 384 DCOMs generate 384 e-LLRs of 95-bit each and are further applied to the integrator unit, as shown in Fig. 4.10. In this integrator unit, input 384 e-LLRs are stored row-wise in a memory array of size 384×19 where each element of this array is segregated into e-LLRs of 5-bit, which is represented as $m_{i,j}$ such that $i \in 384$ and $j \in 19$. These elements are processed by transformation logic to generate 19 output of 1920 bit each, those are represented as e

= $\{e(1), e(2), \cdots e(19)\}$ where $e(j) = \{m_{1,j} : m_{2,j} : \cdots : m_{384,j}\}\ \forall\ j = \{1, 2, 3, \cdots 19\}$ where ':' indicates concatenation. These e-LLRs (e) are corresponding to the submatrix of E matrix, represented by line 16 of Algorithm 3. This submodule subsequently receives all the routed data and processes in parallel fashion, as shown in Fig. 4.10. Therefore, aforementioned parallel processing of *min1* as well as *min2* signals and ConU based SMTC have alleviated data-routing and hardware-consumption of the proposed QC-LDPC decoder architecture.

4.5.3 Variable-Nodes and Check-Nodes Updation Module

The suggested design of variable-nodes and check-nodes updation module (VCUM) is fed with two output LLRs from MSESM: 2688-bit selected-LLRs i.e. $S(j)\ \forall\ j = \{1, 2, 3 \cdots 19\}$ and 1920-bit extrinsic-LLRs i.e. $e(j)\ \forall\ j = \{1, 2, 3 \cdots 19\}$, as shown in Fig. 4.11. These LLRs are processed by VCUM to generate 95-bit updated CN-LLRs i.e. $C(j)\ \forall\ j = \{1, 2, 3 \cdots 384\}$, 1920-bit VN-LLRs $V(j)\ \forall\ j = \{1, 2, 3 \cdots 19\}$, and a 12288-bit CE-LLRs which is the concatenation of 384 32-bit AE-LLRs, as represented by lines 17–30 in Algorithm 3. Foremost, the subtraction unit of VCUM performs VN-matrix updation using S and e LLRs to generate VN-LLRs. Such subtraction unit comprises 19 subtractor submodules (SSs) and each of them consists of 384 subtractors. VN updation is conventionally performed by data-processing in a sequential manner [4]; unlike, this work simultaneously computes these VN-updated values with lower-latency and surge in hardware requirement.

These $V(j)$ VN-LLRs are fed to the rotational unit that performs the cyclic rotational technique (CRT), referring to Algorithm 5. This unit is a network of 18 MUXes ranging between 3:1–46:1 MUXes to perform the LLR cyclic-rotation of Fig. 4.1 and generates 19 rotated $e(j)\ \forall\ j = \{1, 2, 3 \cdots 19\}$ LLRs of 1920-bit each, as shown in line 19 of Algorithm 3. These 19 rotated 1920-bit e LLRs are subsequently fed to a splitter unit that transposes them (referring to line 20 of Algorithm 3) into 384 95-bit R LLRs. Fig. 4.12 (a) shows splitter-unit architecture where 19 input e LLRs (of 1920 bit each) are stored as segregated $n_{i,j}$ LLRs of 5-bit each such that $i \in \{1, 2, 3 \cdots 19\}$ and $j \in \{1, 2, 3 \cdots 384\}$. Such segregated LLRs are processed by transformation logic to generate 384 output values of 95-bit $R(j)\ \forall$

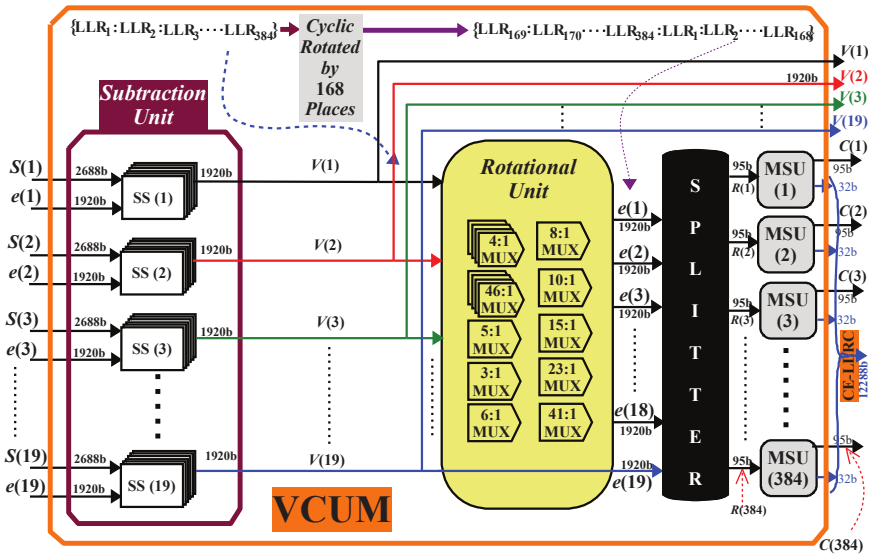


Figure 4.11: Proposed architecture of VCUM that is an aggregation of SU, rotational unit, splitter, and MSUs.

$j = \{1, 2, 3 \cdots 384\}$ LLRs where each $R(j) = \{n_{1,j} : n_{2,j} : n_{3,j} : \cdots : n_{19,j}\}\ \forall\ j = \{1, 2, 3 \cdots 384\}$. Now, these R LLRs are fed to 384 min-sum units (MSUs) for check nodes updation subtask, as shown in lines 21−30 of Algorithm 3.

In MSU architecture, 95-bit rotated 19 $R(j)$ LLRs are segregated into 19 LLRs of 5-bit each and are processed by corresponding two's complement to the sign-magnitude converter (TCSMC), as shown in Fig. 4.12 (b). The TCSMC architecture is identical to the SMTC design. The sign-bits (MSBs) −i.e. represented as $D_{1\times t_b}$ in line 25 of Algorithm 3− and magnitude bits of all 19 SM-LLRs are processed by sign-unit (SU) and magnitude-unit (MU), respectively. Here, SU XORs all 19 sign-bits of D vector to generate a product bit m which is XORed with input 19 sign-bits of D vector to generate an updated signs for corresponding 19 SM-LLRs, as presented in line 25 of Algorithm 3. Simultaneously, minimum value generator (mVG-19) [53] of MU computes two minima values (*min1* and *min2*) and position of *min2* (i.e. *idx*) for the 4-bit magnitudes of all 19 SM-LLRs. Both *min1* and *min2* values are subtracted with an optimized offset value of $(2)_{10}$ [40, 56], and further routed via two 2:1 MUXes to maintain its non-negativity, as shown in Fig. 4.12 (b). Thereafter, *min1*, *min2*, and *idx* values are fed

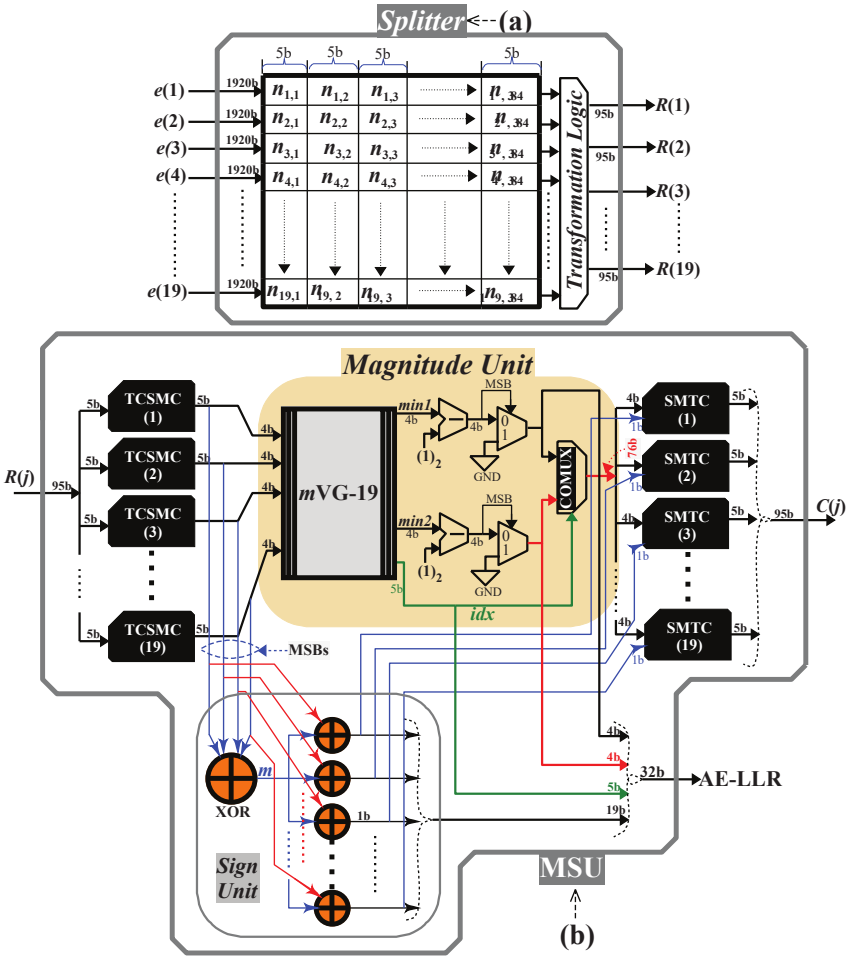


Figure 4.12: Proposed architecture of (a) splitter internal architecture (b) single MSU internal architecture.

to COMUX that generates 76-bit (19×4-bit) output which is an aggregation of 19 updated 4-bit SM-LLR magnitudes of $R(j)$ vector, as shown in line 28 of Algorithm 3. Furthermore, such 76-bit COMUX output is segregated into 19 updated SM-LLR magnitudes of 4-bit each. This COMUX unit is the replacement of various equalizers and MUXes [32] to achieve lower hardware-consumption and routing-complexity. These 19 magnitudes and their corresponding 19 updated sign-bits from SU are simultaneously fed to 19 SMTCs which convert the updated SM-LLRs into TC-LLRs. Subsequently, 19 such 5-bit TC-LLRs are concatenated into a 95-bit $C(j)$ CN-LLR. Furthermore, *min1* and *min2* values of 4-bit each,

19 updated sign-bits, and 5-bit *idx* values are concatenated to generate 32-bit AE-LLR. Thus, each MSU generates a 32-bit AE-LLR and a 95-bit updated $C(j)$ CN-LLR, as illustrated in Fig. 4.12 (b). This updated MSU module underwent gate-level optimization, resulting the reduction in logical delay that is the critical path of overall LDPC decoder architecture.

4.5.4 A-posteriori LLR Computation Module

The suggested architecture of a-posteriori LLR computation module (ACM) executes lines 31−34 in Algorithm 3, as shown in Fig. 4.13. It shows that 384 $C(j)$ CN-LLRs from VCUM are first processed by the integrator unit that generates $e(j)\ \forall\ j = \{1, 2, 3 \cdots 19\}$ LLRs of 1920 bit each. These $e(j)$ LLRs are the transpose of $C(j)$ LLRs. Further, such 19 $e(j)$ LLRs are cyclically rotated by the re-rotational unit −based on the proposed CRT from Algorithm 5− to generate $Re(j)$ LLRs. The re-rotational unit architecture is identical to the rotational unit. Furthermore, 19 VN-LLRs $V(j)$ from VCUM along with 19 re-rotated $Re(j)$ LLRs from the

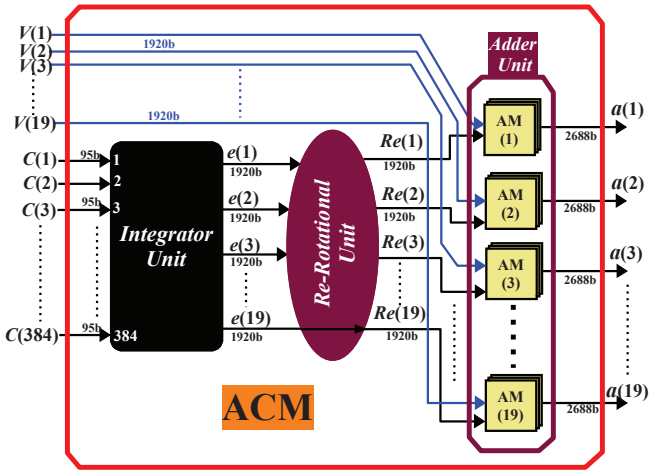


Figure 4.13: The suggested architecture of a-posteriori-LLR computation module (ACM) for the proposed QC-LDPC decoder.

re-rotational unit are fed to adder unit that updates AP-LLRs, based on line 34 in Algorithm 3, as shown in Fig. 4.13. Such addition-unit comprises of 19 addition modules (AMs) where each AM consists of 384 adders. Such AMs add concatenated-LLRs of $V(j)$ VN-LLRs with corresponding LLRs from $Re(j)$ LLRs to produce 19 AP-LLRs of 2688-bit each i.e. $a(j)\ \forall$

$j = \{1, 2, 3 \cdots 19\}$ which are fed back to IMUU, as shown in Fig. 4.9. These 19 $a(j)$ LLRs are used for the next $(r+1)^{th}$ layer-processing of BG matrix in the proposed LDPC decoding algorithm, as shown in Fig. 4.1.

4.5.5 Overall QC-LDPC Decoder Architecture

The proposed QC-LDPC decoder architecture that aggregates aforementioned submodules (i.e. IMUU, MSESM, VCUM and ACM) with a hard decision register memory (HDRM) is shown in Fig. 4.14. The decoding process initiates by feeding 7-bit channel LLRs l_i $\forall$ $i = \{1, 2, 3 \cdots n\}$ to IMUU where n can range upto a maximum value of 26112 bit, based on the 5G-NR specification [1, 58]. Here, IMUU performs initialization and AP-LLR matrix updation to generate these stored LLRs in concatenated form of $A(j)$. These $A(j)$ LLRs and a 12288-bit CE-LLR from VCUM are subsequently processed by MSESM that performs LLRs selection (using MST from Algorithm 4) and updates e-LLR matrix of flow-diagram from Fig. 4.1. Further, MSESM computes $S(j)$ and $e(j)$ LLRs $\forall$ $j = \{1, 2, 3 \cdots 19\}$. Such LLRs are routed by VCUM to perform VN-matrix updation, cyclic-rotation from (5), and CN-matrix updation to generate $V(j)$, $C(j)$, and a CE-LLR, as shown in Fig. 4.14. These $C(j)$ and $V(j)$ LLRs from VCUM are processed by ACM to compute $a(j)$ AP-LLRs which are fed back to IMUU for performing the AP-LLR matrix updation using MWT –referring Algorithm 6– that is corresponding to r^{th} layer-processing of B matrix, as shown in Fig. 4.14. Such layer-processing consumes two clock cycles in IMUU for r^{th} B matrix layer-processing. This updated memory of IMUU is used in the next $(r+1)^{th}$ layer-processing of B matrix and such layer-processing continues till the last layer of B matrix which is computed in single decoding iteration. Therefore, an overall decoder architecture is designed based on the high parallelism property of LDPC codes and also achieves lower routing-complexity or data congestion, as discussed in the aforementioned submodules.

The timing schedule of the overall QC-LDPC decoder architecture has been presented in Fig. 4.15. Initially, LLRs are fed into IMUU by asserting high value to *LLR-writing* signal which takes n clock cycles to load the LLRs. Thereafter, *LLR-reading* signal is asserted high

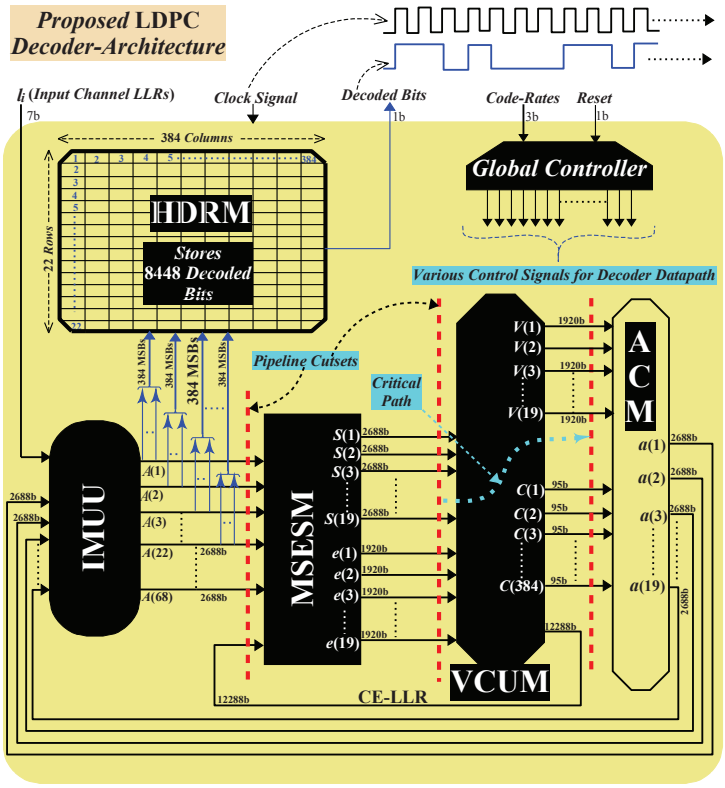


Figure 4.14: Proposed QC-LDPC decoder architecture that is compliant to 5G-NR specifications, supporting all the code-rates of 5G-NR and decodes 8448 bits from 26112 received channel-LLRs.

to provide the LLRs that correspond to an ongoing B matrix layer. Consecutively, VN updation, CN updation, and AP-LLR updation processes are started that correspond to various layers of B matrix, as shown in Fig. 4.15. These updation processes follow an overlapping message scheme [61] to reduce the latency of decoder architecture. After the computations of AP-LLRs, they are written back to IMUU by triggering high the *LLR-writing* signal, as illustrated in Fig. 4.15. Furthermore, the aforementioned processing continues till the last m_b layer of B matrix, and this marks the end of a single decoding iteration. Such processes are repeatedly carried out for 10 decoding-iterations. The proposed decoder architecture operates on 46 processing layers for a 1/3 code rate that consumes 92 clock cycles (i.e. 46×2 clock cycles) for a single iteration. Therefore, it requires 920 clock cycles (i.e. 92×10 clock cycles) to complete 10 iterations. After the 10^{th} iteration, HDRM is activated

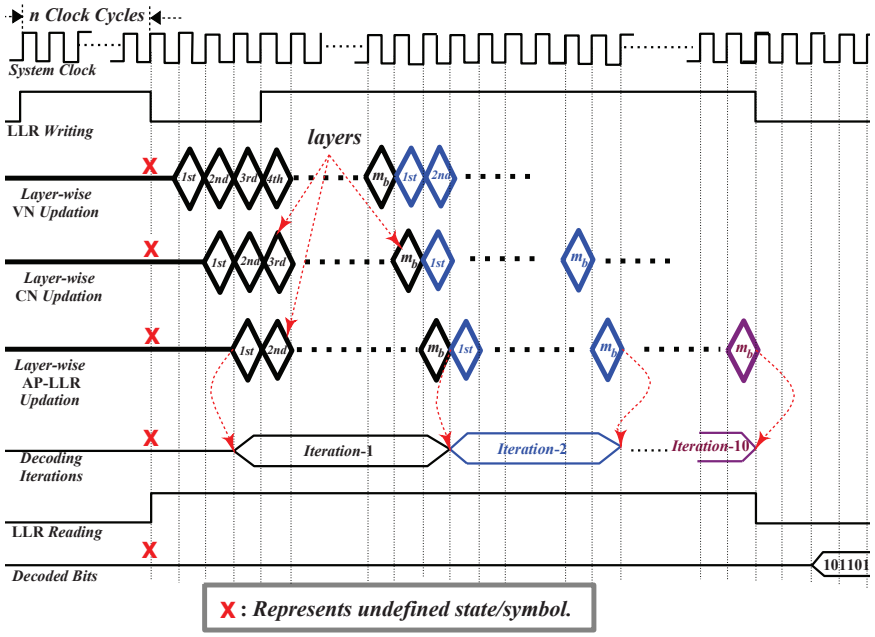


Figure 4.15: Timing diagram that represents scheduling of layer computation in overall QC-LDPC decoder architecture.

and stores all the sign-bits of updated AP-LLRs that correspond to $A(j)$ LLRs in the MRB of IMUU, as shown in Fig. 4.9 and 4.14. However, the LDPC decoder must generate 8448 decoded bits corresponding to information bits, compliant to 5G-NR specifications. Hence, only 8448 LLRs are encapsulated by an initial 22 $A(j)$ LLRs of IMUU to generate the decoded information bits. Therefore, 384 MSBs (corresponding to 384 7-bit updated AP-LLRs) from each $A(j)$ LLR are tapped as a word of 384 bits which is stored in HDRM, as depicted in Fig. 4.14. It also shows that 22 such 384-bits words are tapped and concatenated from $A(j)$ LLRs and stored in HDRM, which has a size of 22×384 bit (i.e. equivalent to 8448 decoded bits). Eventually, these decoded bits from HDRM are fetched out sequentially in every clock cycle.

4.5.6 Implementation Results and Comparisons

The suggested QC-LDPC decoder architecture has been synthesized, post-route simulated, and hardware prototyped in the FPGA platform of the Xilinx Zynq-Ultrascale+ board. This

prototype is capable of decoding a maximum code length of 26112 bits with a mother code rate of 1/3 and also supports all other code-lengths and code-rates (specified by the 5G-NR standard [1, 58] with the aid of global controller, as shown in Fig. 4.14. This proposed QC-LDPC decoder achieves a maximum throughput (Θ_T), expressed as

$$\Theta_T = \frac{n \times f_{max}}{i_{max} \times \mathbb{L} \times \mathbb{C}} \tag{4.6}$$

of 13.3 Gbps while decoding a code with n = 10368 bits, $\mathbb{L}$ = 5 base matrix layers at a code rate of 8/9. On the other side, Θ_T = 3.64 Gbps of minimum throughput is obtained by this design for n = 26112 bits, $\mathbb{L}$ = 46 base matrix layers at a code rate of 1/3. Here, i_{max} represents the number of decoding iterations (i_{max} = 10 iterations in this work), and $\mathbb{C}$ denotes the number of clock cycles consumed while processing each base matrix layer ($\mathbb{C}$ = 2 clock cycles, as discussed earlier in section 4.5.5). Our proposed decoder architecture achieves the high data-throughput due to its higher code length (n for 5G-NR) and lower-latency which is achieved by the balanced amount of parallelism technique. However, due to such parallelism, hardware consumption tends to increase. Nevertheless, this adverse effect has been calibrated with the aid of our lower computational-complexity SOMS decoding algorithm resulting in hardware-efficient decoder architecture. Thus, our QC-LDPC decoder achieves higher data-throughput with lower hardware-consumption.

Static timing analysis indicates that the suggested decoder operates at a maximum clock frequency of f_{max} = 128.36 MHz. Further, it has the shortest and longest latencies (computed as $\Lambda_D = n/\Theta_T$) of 0.77 μs at 8/9 code rate and 7.17 μs at 1/3 code rate, respectively. Apart from these combinations of code-lengths and code-rates, there are other such combinations specified by the 5G-NR standard [2] and are also supported by our decoder architecture. For these pairs of code-lengths and code-rates, Fig. 4.16 (a) and (b) present throughput and decoding-latency values achieved by the proposed LDPC decoder, respectively.

This chapter presents two figure-of-merits (FOMs) for fair comparison: hardware uti-

Table 4.2: Comparison of Proposed LDPC-Decoder Implementation-Results with the State-of-the-Art Works.

Comparison Metrics	**This work**		**[4]-TVT 2021**	**[62]-TVLSI 2021**	**[63]-Xilinx 2021**	**[64]-TCAS-II 2020**	**[65]-TCAS-I 2019**
FPGA Board	**Xilinx Zynq Ultrascale+**	**Intel Altera Stratix-IV**	Intel Altera Stratix-IV	Xilinx Kintex-7	Xilinx Kintex Ultrascale+	Xilinx Kintex-7	Xilinx Virtex-7
Tech. Node (nm)	**16**	**40**	40	28	20	28	28
Quant. Bits	**7-bit**		6-bit	8-bit	8-bit	5-bit	4-bit
Throughput (Mbps)	**3643 – 13309**	**2200 – 8037**	522.41	391 – 1096	1760	3600	634.7
Latency (μs)	**0.77 – 7.167**	**1.29 – 11.8**	2.08	18	14	0.42	3.63
Code Type	**Irregular**		Irregular	Irregular	Iregular	Regular	Irregular
Code Length	**10368 – 26112**		1088	3456	26112	155	2304
Exp. Factor	**384**		46	120	384	31	96
Clock Freq. (MHz)	**128.36**	**77.5**	333	160	440	700	75
Cri. Path Delay (ns)	**7.79**	**12.9**	3	6.25	2.27	1.42	13.3
Scheduling	**Layered**		Flooding	Flooding	Flooding	Layered	Layered
Code Rate	**1/3 – 8/9**		1/2 – 5/6	1/3 – 5/6	1/2 – 8/9	–	3/4
Base Matrix Size	**46 × 68**		8 × 24	46 × 68	46×68	3×5	8×24
Standard	**5G-NR**		WiMAX	5G-NR	5G-NR	–	WiMAX
No. of Layers	**5 – 46**		8	5 – 46	46	3	8
No. of Iter.	**10**		5	8	8	10	10
Pipeline Stages	**3**		5	–	–	–	2
Memory Size (kb)	**765.8**		173.8	7146	4212	486	2430
Decoding Algorithm	**Offset Min-sum**		Min-sum	Offset Min-sum	Offset Min-sum	Min-sum	Min-sum
Hard. Utili. †	**1200394**	**1010128**	12423	7438394	4415146	525764	2564543
Memory Type	**Registers**		Registers	BRAM	BRAM	BRAM	BRAM
BER of 10^{-5} (dB)	**0.9 – 7.1**		–	-0.5 – 4.2	–	–	0 – 5.6
Permutation Network	**Bit-rotational Mapping**		–	Barrel Shifter	–	–	Barrel Shifter
***PTLR* (Mbps/μs)**⋆	**17284.4**	**6230.2**	251.75	60.88	125.7	8571	174.84
HUE♮	**1.63**	**2.73**	2.97	147.54	54.53	48.68	505

†: Hardware Utilization = (No. of LUTs + No. of F7 MUXes + No. of F8 MUXes + No. of Flip-flops + BRAM (in bits)).
⋆: Peak-Throughput Latency Ratio (*PTLR*) = (Peak-Throughput / Latency) Mbps/μs.
♮: Hardware Utilization Efficiency (*HUE*) = Hardware Utilization / (No. of Layers × Peak-Throughput) hardware-resources/layer/Mbps.

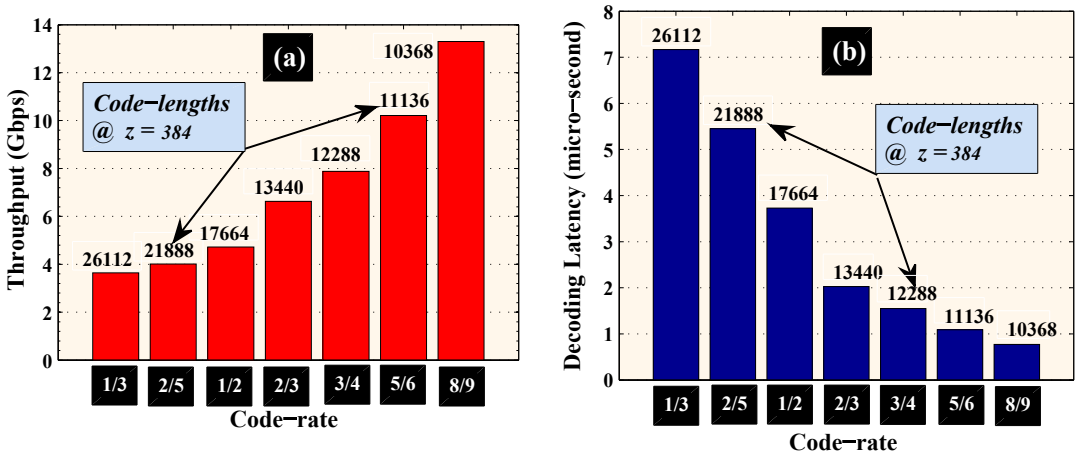


Figure 4.16: Timing diagram that represents scheduling of layer computation in overall QC-LDPC decoder architecture.

lization efficiency (HUE) and peak-throughput to latency ratio ($PTLR$) as

$$HUE = \left(\frac{\mathbb{H}_u}{\mathbb{L} \times \Theta_T}\right) \tag{4.7}$$

$$PTLR = \left(\frac{\Theta_T}{\Lambda_D}\right) \tag{4.8}$$

where $\mathbb{H}_u$ represents the FPGA hardware utilization of the LDPC decoder. Here, HUE denotes the amount of hardware consumed per layer of the basegraph B matrix to attain a throughput of 1 Mbps and it has a unit of hardware-resources/layer/Mbps. Thereby, a lower value of HUE is desirable for hardware-efficient design. It is to be noted that the number of base matrix layers ($\mathbb{L}$) is proportional to hardware consumption, as it increases the size of steering logic with a larger $\mathbb{L}$ value in the LDPC decoder architecture. On the other side, $PTLR$ represents the peak throughput achieved by the decoder (expressed in Mbps) in every 1 μs of the processing time. Desirably, $PTLR$ value must be higher and it has a unit of Mbps/μs. Implementation results of our QC-LDPC decoder architecture are compared with the relevant state-of-the-art works in Table 4.2. It shows that the proposed QC-LDPC decoder architecture has 7.5× higher data-throughput compared to the state-of-the-art Logicore intellectual-property (IP) of the LDPC decoder from Xilinx [63].

Subsequently, it can be observed that our decoder delivers 3.6× better throughput than the implementations reported by Y. Liu et al. in [64]. Based on the aforementioned formulations of FOMs from (4.7) and (4.8), the proposed LDPC decoder delivers a HUE of 1.96 hardware-resources/layer/Mbps which is 34% lower than the most-efficient contemporary work, reported by Shan Cao et al. [4]. Furthermore, Table 4.2 presents $PTLR$s of proposed and reported designs. Thus, FPGA implementation of the suggested LDPC decoder has achieved competitive $PTLR$ among all the reported works and specifically 2× better than the highest value reported by Y. Liu et al. in [64], as shown in Table 4.2. Therefore, the proposed LDPC decoder is hardware-efficient design that delivers higher throughput. For fair comparison with the state-of-the-art work from [4], we have implemented our decoder architecture in the same platform of Intel Altera Startix-IV FPGA board and its results are compared with [4] in Table 4.2. It shows that the proposed decoder delivers 93.5% higher data-throughput, 95.96% better $PLTR$, and 8.1% lower HUE than the implementation results of [4].

4.5.7 Hardware Verification and ASIC Design

The proposed QC-LDPC decoder that is hardware implemented on Xilinx Zynq UtraScale+ FPGA-board has been functionally verified with the aid of MATLAB simulation environment, as shown in Fig. 4.17. It shows a schematic representation and snapshot of our real-world test environment. In the MATLAB environment, the QC-LDPC encoder converts k information-bit into n codeword-bit (such that $n>k$). Subsequently, these n bits are modulated and transmitted via AWGN channel environment. At the receiver end, n LLR samples are fetched from the channel that corresponds to n noisy-bit. Furthermore, these n LLR samples are fixed-point quantized into a 7-bit value which is fed to the FPGA-implemented LDPC decoder, as shown in Fig. 4.17. These quantized input LLR samples from the MATLAB environment are written into the co-efficient (*.coe*) file, which is interfaced with the FPGA prototype of our decoder. Specifically, this *.coe* file is used for initializing the block random access memory (BRAM) of the FPGA board. Consecutively, the QC-LPDC decoder on FPGA processes these quantized LLR samples and generates the hard decoded-bits.

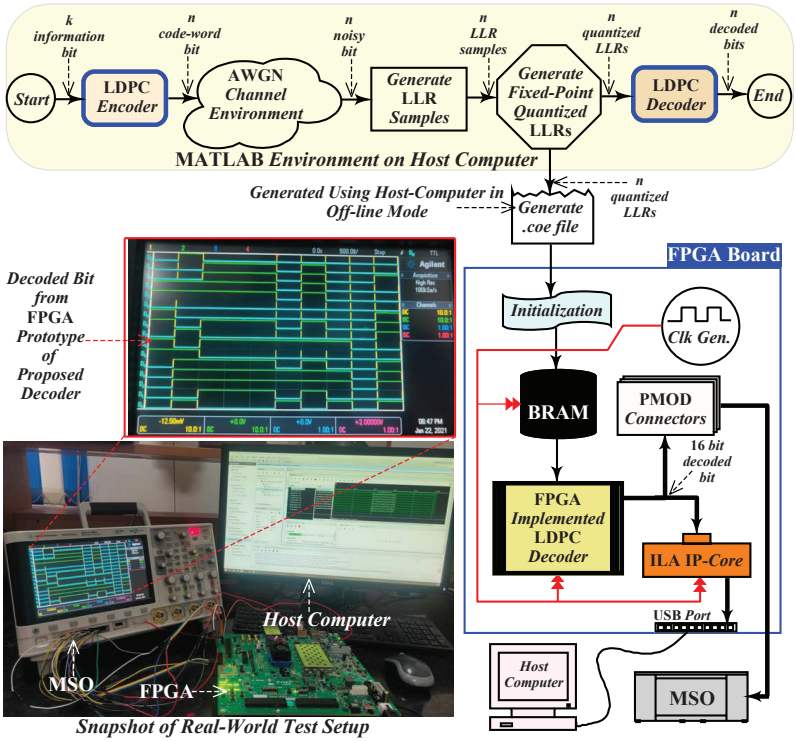


Figure 4.17: Schematic-representation and snapshot of the real-world test setup for the FPGA-prototype of proposed LDPC decoder.

The integrated logic analyzer (ILA) IP-core tracks the updated LLR values and outputs the decoded bits of the LDPC decoder. Therefore, initialized BRAM and ILA IP-core are instantiated along with the QC-LDPC decoder on the FPGA board that are synchronously operated by onboard system clock, as shown in Fig. 4.17. Thereafter, the output decoded bit of the QC-LDPC decoder is configured to interact with ILA IP-core and PMOD connector of the FPGA board. Thus, ILA IP-core and PMOD connectors are digitally interfaced with the monitor of the host computer and the mixed signal oscilloscope (MSO), respectively, as shown in Fig. 4.17. Eventually, the output decoded bits from QC-LDPC FPGA prototyped are displayed on the MSO/host-computer screen and are verified with the simulated outputs from the MATLAB environment.

Furthermore, the proposed QC-LDPC decoder has been ASIC synthesized and post-layout simulated in united microelectronics corporation (UMC) 90 nm-CMOS technology

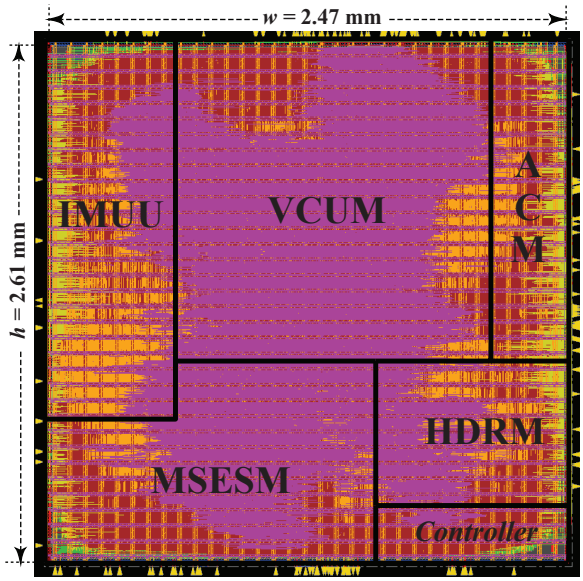


Figure 4.18: ASIC chip layout of the proposed QC-LDPC decoder in 90 nm-CMOS technology node, occupying a core area of $h \times w$=6.45 mm^2.

node, with the supply voltage of 1 V. At first, our design is coded using the Verilog HDL which is functionally verified, synthesized, and static-timing analyzed as well as power estimated with the aid of Verilog compiler-&-simulator, design compiler and prime time EDA-tools from Synopsys, respectively. Following that, its gate-level netlist has been imported to the Cadence Innovus tool where the physical design processes like floorplan, placement, power routing, nano routing, clock tree synthesis, and timing analysis have been carried out. Thus, the final chip layout of our LDPC decoder is shown in Fig. 4.18 that consumes an area of 6.45 mm^2. Based on the timing analysis, our decoder operates at the maximum clock frequency of 192.3 MHz and delivers a peak throughput of 9.6 Gbps while decoding a code with n = 2592 bits, $\mathbb{L}$ = 5 base matrix layers at a code rate of 8/9.

Further, complete implementation results of the suggested QC-LDPC decoder are listed in Table 4.3 where they are compared with the results of relevant reported works from the literature. The proposed decoder has achieved 57.3% higher data-throughput than the state-of-the-art implementation, reported by S. Yun et al. [45]. For a fair comparison, our implementation results are also compared with the recently reported work of H. Cui et al.

Table 4.3: Comparison of ASIC Implementation Results of Proposed LDPC Decoder with Relevant State-of-the-Art Works.

Metrics	This work	TCAS-II 2022 [45]	TCAS-I 2021 [40]	Elsevier 2019[14]
Technology (nm)	**90**	65	90	65
Standard	**5G-NR**	5G-NR	5G-NR	WiMAX, WiLAN
Quantization Bits	**7**	8	6	6
Code Length	**3072**	1664	2600	2304
Expansion Factor	**96**	64	52	96
Code Rate	**3/4–5/6**	11/13	1/5	1/2–5/6
Clock Frequency (MHz)	**192.3**	244	120.9	414
Decoding Iterations	**10**	10	15	10
Throughput (Mbps)	**9600**	4100	523.9	1528
Area (mm^2)	**6.45**	$2.22^{\dagger}$	1.889	$5.69^{\dagger}$
Power (mW)	**3456**	115	72	389
Energy Effn.$^{\$}$ (nJ/bit)	**0.36**	0.28	0.13	0.25

†: Scaled Area = Area/s^2 where s = scaling factor; S = 65/90;
$: Energy Efficiency = Power/Throughput nJ/bit.

[40], which has been implemented in the same 90 nm-CMOS process, as illustrated in Table 4.3. It shows that the proposed work achieved 18.3× higher data-throughput.

4.6 Summary

This chapter proposed a novel SOMS decoding algorithm for QC-LDPC codes and its corresponding decoder architecture that is compliant to the 5G-NR wireless communication standard. This algorithm has been designed to reduce the overall computational-complexity like comparisons, additions, and memory storage consumption. Further, the SOMS decoding algorithm is compared with other existing decoding algorithms in the literature such as 2D-MS [59], adjusted MS [60], Improved SP [14], MS [15], and Normalized MS [50]. It is to be noted that the proposed algorithm consumes ≈ 41% lesser memory storage and ≈16% lesser comparison computations with respect to the state-of-the-art 2D-MS decoding algorithm [59]. In Addition, a comprehensive performance analysis of the proposed

SOMS decoding algorithm has been presented for the specifications of the 5G-NR wireless communication standard. We have compared the FER performance of the SOMS algorithm with existing decoding algorithms and observed minor coding loss with existing decoding algorithms. It shows that the suggested SOMS algorithm delivers an adequate FER of 10^{-5} at SNR of 1.3 dB while decoding 16-QAM modulated QC-LDPC code with a code-rate of 1/3 and the code-length of 26112 bits.

Subsequently, we proposed a high data-throughput QC-LDPC decoder architecture that has been designed based on the suggested SOMS decoding algorithm. This decoder architecture is compliant to 5G-NR wireless communication standard that decodes (26112, 8448) QC-LDPC codes based on the standard base-graph matrix. Moreover, our proposed QC-LDPC decoder architecture supports all the 5G-NR standard code-rates and code-lengths with the maximum expansion factor (z) of 384. A detailed architecture for the overall QC-LDPC decoder has been presented in this chapter where various sub-modules are aggregated. This proposed decoder architecture enhances the hardware-efficiency and also follows the parallelism that alleviates the high data-throughput with lower decoding-latency. Some additional architectural transformations have been carried out to reduce the routing-complexity of the proposed decoder in this chapter.

Further, this SOMS algorithm based overall QC-LDPC decoder architecture has been FPGA prototyped on Xilinx Zynq Ultrascale+ and Intel Altera Stratix IV. This 7 quantization-bits 5G-NR based QC-LDPC decoder architecture operates at the maximum clock frequency of 128.36 MHz and delivers a peak data-throughput of 13.3 Gbps while decoding at an 8/9 code-rate. Our decoder has been functionally verified with the real-world test setup on the FPGA platform. Additionally, overall QC-LDPC decoder architecture has been ASIC synthesized and post-layout simulated in UMC 90 nm-CMOS technology node with the power supply voltage of 1V. It occupies an area of 6.45 mm^2 with a maximum operating clock frequency of 192.3 MHz. The (2592, 2304) QC-LDPC decoder delivers the maximum data-throughput of 9.6 Gbps while decoding of 8/9 code-rate. 5G-NR applications support two types of error correction codes for channel decoding: QC-LDPC codes for data signaling

and polar codes for control signaling. So, we focus on designing such a reconfigurable channel decoder that has the ability to decode the LDPC and polar codes by using a unified decoder architecture. In the upcoming chapter, we presented a reconfigurable channel decoder that is used to decode LDPC and polar codes for 5G-NR wireless communication applications.

Chapter 5

Unified Reconfigurable LDPC/Polar Channel Decoder

5.1 Introduction

In the physical layer of 5G-NR networks, error-correcting-codes are playing a critical role in reliable data transmission over unpredictable channel conditions and fading possibilities. The latest technological development in the telecommunication sector is 5G-NR technology that is envisioned to render enhance connectivity to users and materialize the digitization of diverse industrial verticals. In recent release-16 of 5G-NR standardization of the physical layer, the 3GPP has advocated LDPC and polar codes as standard channel codes in the physical layer specifications of 5G-NR wireless communication standard [1, 21]. Hence, 3GPP release-16 for 5G radio access network (RAN) has officially standardized LDPC and polar codes as channel-coding techniques for data and control channel information, respectively.

In section 4.2 of chapter 4, we proposed a hardware-friendly SOMS decoding algorithm for LDPC codes that has been designed based on layered scheduling. It allevi-

ates computational-complexity for the LDPC decoding algorithm and also improves the hardware-efficiency and data-throughput due to its parallelism process. As discussed in section 1.2.5 of chapter 1, LDPC and polar channel codes have been employed for reliable data transmission in the physical layer of 5G-NR wireless communication standard [1]. Therefore, this chapter presents the formulation of an unified decoding technique that is used for the decoding of LDPC and polar codes. In addition, design of a new reconfigurable LDPC/polar channel-decoder based on the unified decoding technique, compliant to the specifications of 5G-NR mMTC and URLLC applications, has been presented here.

The proposed decoding technique alleviates the storage requirement by mutually sharing the memory and processing element for LDPC and polar code. It is the combination of SOMS and belief-propagation decoding-algorithms for LDPC and polar codes, respectively, that delivers the high data-throughput with lower decoding-latency due to its parallelism nature. Further, a comprehensive performance analysis of the proposed decoding technique has been carried out in this chapter. In addition, a novel unified LDPC/polar decoder architecture has been presented that is compliant to mMTC and URLLC applications of 5G-NR wireless communication standard. It supports SOMS and belief propagation decoding-algorithms for LDPC and polar code, respectively. This decoder architecture enhances data-throughput and hardware-efficiency for supporting multiple code-rates of 5G-NR standard. Eventually, an ASIC chip of the proposed unified LDPC/polar decoder has been fabricated. Subsequently, the fabricated ASIC chip of channel-decoder has been characterized and functionally validated in the real-world test setup scenario. We also analyzed the power and data-throughput measurements for the proposed fabricated ASIC chip. This reconfigurable channel-decoder has been designed for the high-end specifications of 5G-NR wireless communication systems. Our contributions in this chapter are highlighted as follows:

- A new technique for reconfigurable LDPC/polar decoding process has been proposed in the chapter. This technique is represented in a data flow diagram that presented a unified decoding for LDPC and polar codes.

- A new flexible and hardware-efficient architecture of reconfigurable channel-decoder for LDPC and polar codes based on the aforementioned technique has been presented in this chapter. It supports multiple standardized code-rates and code-lengths that are specified for the mMTC and URLLC applications of the 5G-NR wireless communication standard.

- In addition, a new memory-shared architecture has been presented that improves the data-flow of information and excludes the requirement of various computation modules in the proposed channel-decoder hardware. Furthermore, this chapter also proposes hardware architecture for the processing element with lesser area requirement that is used in the polar decoding process.

- The proposed unified LDPC/polar reconfigurable decoder has been fabricated in united microelectronics corporation (UMC) 110 nm-CMOS technology node. This chip has been characterized and functionally validated using the real-world test setup. All the measured results of our ASIC chip have been analyzed and compared with the contemporary implementations from the literature.

- Finally, a post-silicon power measurement and throughput estimation while operating at various SNR and clock frequencies, respectively, have been analyzed in this chapter. We have performed this analysis for all the 5G-NR standard LDPC and polar code-rates.

5.2 Reconfigurable Channel Decoding for LDPC/Polar code

This section presents the proposed reconfigurable technique and its performance analysis for the decoding of LDPC/polar code. In addition, it can flexibly decode both LDPC and polar codes with multiple code-rates as well as code-lengths of the aforementioned 5G-NR applications. Furthermore, 5G-NR standardization of LDPC code is primarily based on two prototype BG matrices: $B1$ and $B2$ [1, 2]. Our decoder circuitry has been designed to decode

$B1$ matrix (referred as Fig. 1.7) based (n=560, k=352), (n=512, k=352), (n=464, k=352), and (n=432, k=352) LDPC codes at multiple code-rates of 2/3, 3/4, 5/6, and 8/9, respectively, with a fixed expansion factor of z=16. The suggested reconfigurable decoder also decodes (n=128, k=64) polar code with a code-rate of 1/2, which has been standardized by 5G-NR standard for control signaling.

5.2.1 Reconfigurable Decoding Technique

Schematic flow of the proposed unified LDPC/polar decoding technique has been presented in Fig. 5.1. It illustrates LDPC, polar, and reconfigurable decoding processes. Specifically, it depicts storage and processing elements for LDPC as well as polar decoding that are mutually shared and accessed, depending on whether the input channel LLRs belong to LDPC or polar code. To begin with, these input channel-LLRs are first stored in the left memory, as shown in Fig. 5.1. Further, these LLRs are transferred to BG-selected layer or polar left-routing for LDPC or polar decoding, respectively. Simultaneously, right memory passes the intermediate LLRs to the subtraction unit for subtracting these LLRs with BG-selected LLRs for LDPC decoding based on line 17 in Algorithm 3, as discussed in section 4.2 of chapter 4. Subsequently, these subtracted values are permutation rotated by the cyclic rotation unit and are routed to the processing element for LDPC decoding. On the other hand, data from right memory and polar left-routing units are processed by the same processing element for polar decoding, as shown in Fig. 5.1. This unit performs min-sum approximation for both LDPC and polar decoding process. Following that, outputs from the processing element are routed to cyclic re-rotation or polar right-routing units for LDPC or polar decoding, respectively. Such cyclically rotated data are added and fed back to left memory via a memory updating unit for LDPC decoding to update the left memory for next iteration, as illustrated by Fig. 5.1. It also shows that the right-rotated data are transferred back to both left and right memories for polar decoding, to update them for the next iteration. Hence, this data flow for reconfigurable decoding of LDPC or polar code is continuously processed until the maximum number of iterations is exhausted.

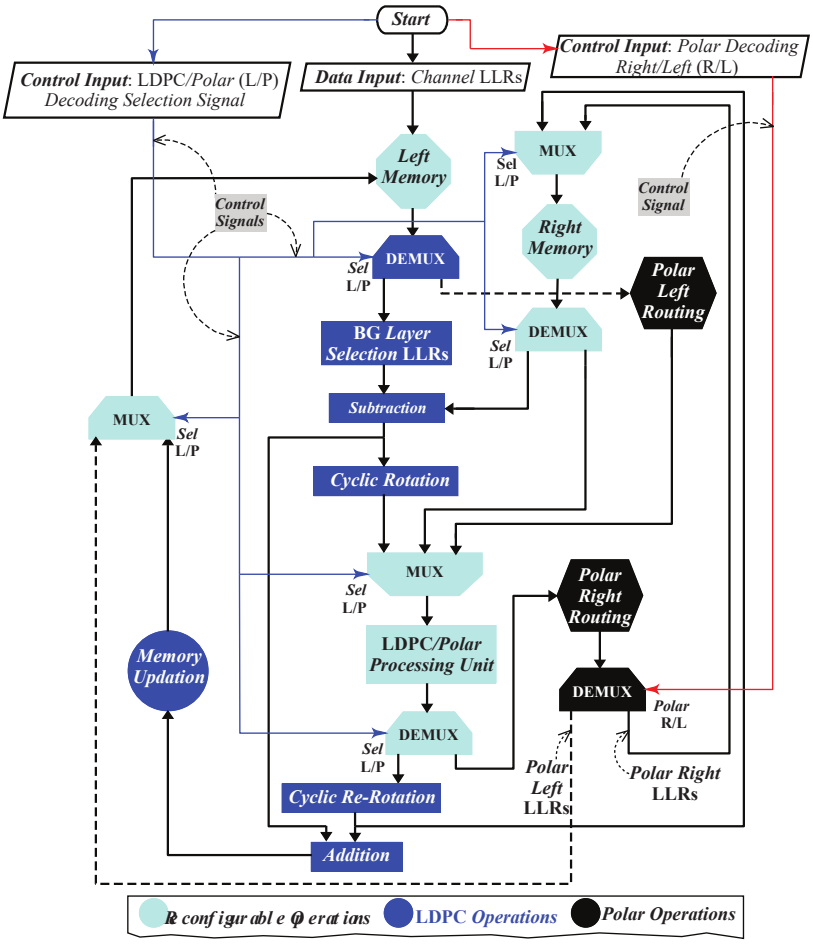


Figure 5.1: Flow diagram of the proposed reconfigurable technique for unified decoding of LDPC and polar codes.

5.2.2 Performance Analysis

Based on the aforementioned decoding technique, frame-error-rate (FER) performance analysis of both LDPC and polar decoding algorithms has been carried out with 5G-NR specifications [1, 2]. Here, we present Monte-Carlo simulations for the transmission of 10^8 bits into LDPC and polar frames with binary phase shift keying (BPSK) modulation scheme over the additive white Gaussian noise (AWGN) channel environment. The LDPC frame size has been standardized (for 5G-NR technology) to 560, 512, 464, and 432 for the code-rates of 2/3, 3/4, 5/6, and 8/9, respectively, with an expansion factor of z=16.

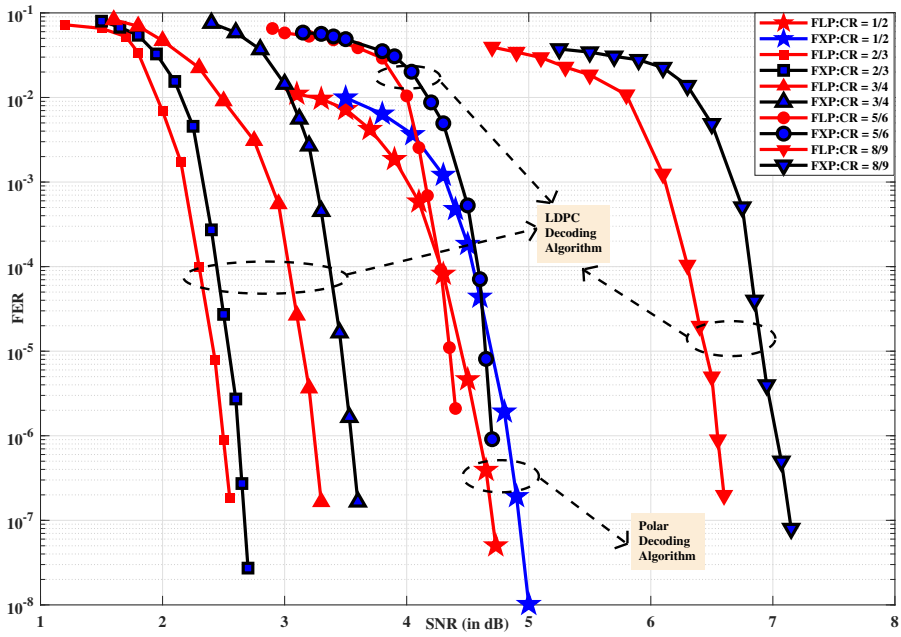


Figure 5.2: Performance analysis (FER versus SNR plots) of the proposed reconfigurable LDPC/polar decoding algorithms under floating and fixed point scenarios for various code rates, compliant to the 5G-NR wireless communication standard.

Similarly, standard frame size for polar code is 128 with the code-rate of 1/2. As a result, multiple FER versus SNR plots are presented in Fig. 5.2. These plots are obtained for LDPC and polar decoding algorithms for 10 iterations, using floating and fixed-point quantization schemes, as shown in Fig. 5.2. The quantization bits format for fixed point representations is denoted by (r_q, s_q) where r_q and s_q represent integer and fraction bits, respectively. Here, we perform the FER performance analysis for (6, 0) AP-LLRs and (5, 0) CN messages for LDPC codes whereas the polar decoding message have (4, 0) fixed point quantization-bits format. It clearly shows that the FER versus SNR plots of floating-point scheme have higher coding gain than the fixed-point scheme which is adopted in the design of proposed reconfigurable decoder. On the other side, effect of decoding iterations on the FER performance has been depicted in Fig. 5.3. It is observed that the FER performance improves with decoding iterations over the higher value of SNR. Specifically, decoding of LDPC code with the code-rate of 3/4 provides the constant FER of $10^{-4.5}$ after 25 iterations at SNR of 3.2 dB,

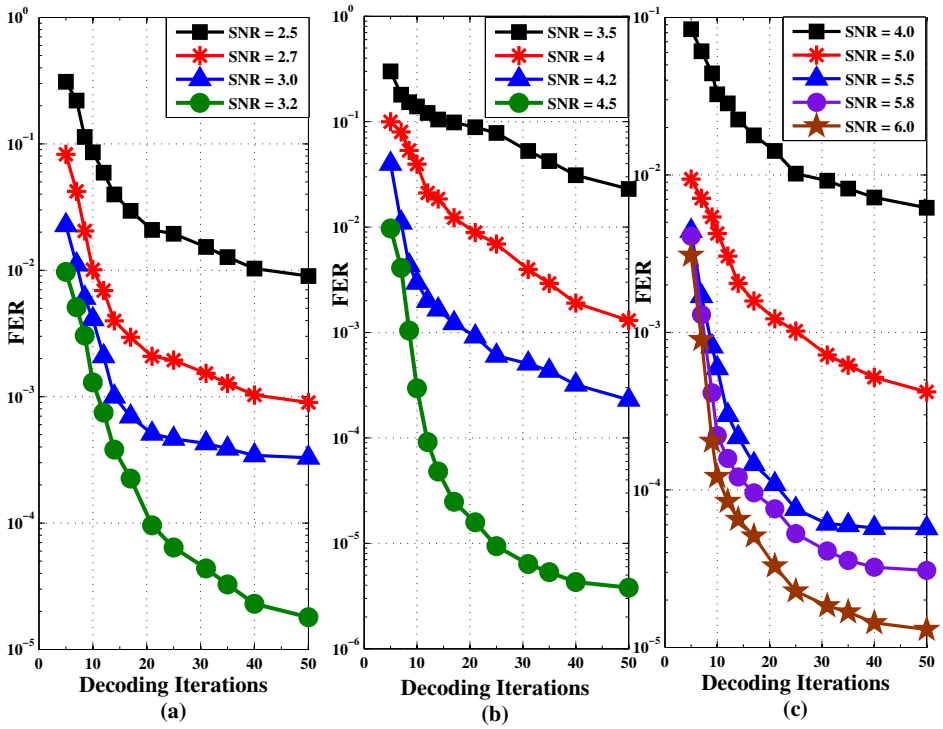


Figure 5.3: FER versus iteration plots of the proposed (a & b) LDPC and (c) polar decoding algorithms.

as shown in Fig. 5.3 (a). Furthermore, LDPC code-rate of 5/6 delivers the constant FER of $10^{-5.3}$ beyond 25 decoding iterations at the SNR of 4.5 dB, as illustrated in Fig. 5.3 (b). Finally, FER versus iteration plot for polar decoding (code-rate of 1/2) shows a FER of $10^{-4.8}$ beyond 25 iterations at a SNR of 6 dB, as shown in Fig. 5.3 (c).

5.3 Reconfigurable Channel Decoder

In this chapter, a reconfigurable channel-decoder architecture has been presented that is compliant to the specifications of the 5G-NR wireless communication standard. This decoder architecture supports the SOMS and belief-propagation based decoding algorithms for LDPC and polar codes, respectively. Various 5G-NR standard code-rates for LDPC and polar codes are supported by the proposed reconfigurable channel-decoder for the physical

layer implementation of 5G-NR networks.

5.3.1 Overall Reconfigurable Channel Decoder Architecture

This section presents an overall hardware design of the proposed unified reconfigurable LDPC/polar decoder architecture, as shown in Fig. 5.4. It has three key modules: memory & data routing (MDR) unit, processing element (PE), and hard decision memory (HDM). To begin with, a collection of four quantized channel-LLRs has been received at the input side and are stored in the memories of MDR unit, as shown in Fig. 5.4. The MDR unit has been designed with the memories to store and route input channel-LLRs as well as intermediate-LLRs for both LDPC and polar decoding processes.

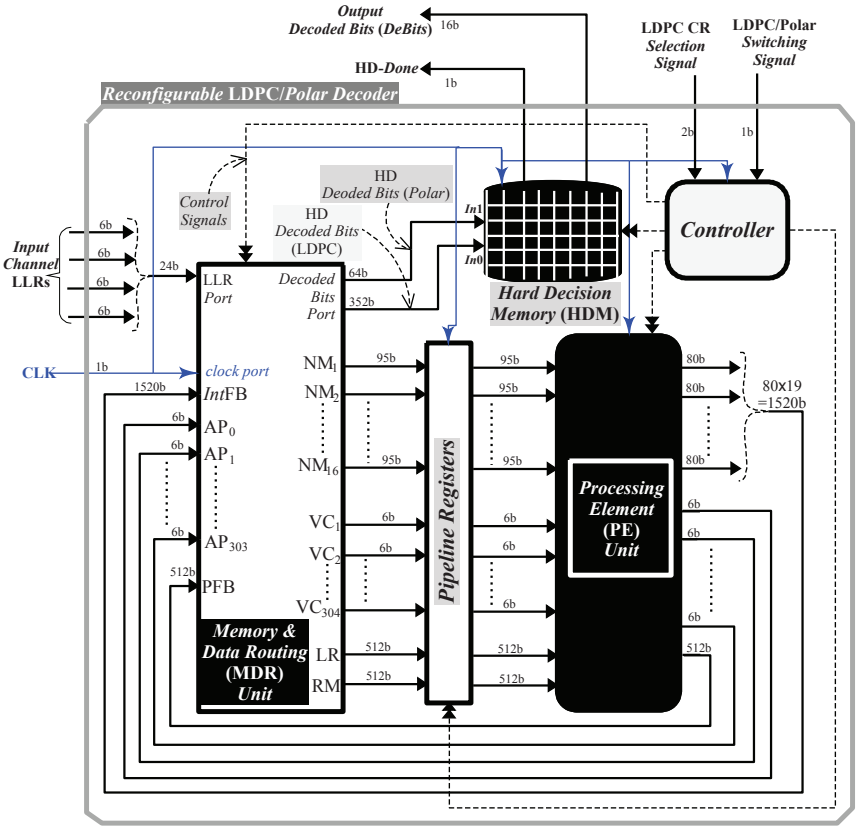


Figure 5.4: Overall hardware-efficient architecture of the proposed reconfigurable channel-decoder for decoding LDPC or polar code.

For LDPC decoding, MDR unit processes the incoming channel-LLRs for the first iteration and it later processes intermediate feedback (IntFB) LLRs for the remaining decoding iterations. Thus, it is fed with IntFB and a-posteriori (AP) LLRs from PE to generate variable-to-check node LLRs (i.e. denoted as VC) and normalized LLRs (i.e. NMs), as shown in Fig. 5.4. In addition, this MDR unit also facilitates data-routing with synchronization in order to fulfill the setup and hold timing requirements of our decoder micro-architecture. Following that, both VC and NM LLRs are applied to the PE through pipeline registers to alleviate critical-path delay of the proposed reconfigurable decoder, as presented in Fig. 5.4. It computes IntFB and AP-LLRs that are used for the next decoding iterations for LDPC decoding process.

For polar decoding, memories of the same MDR unit initially stores channel-LLRs and polar feedback (PFB) LLRs, as shown in Fig. 5.4. It also assigns maximum and zero values for free and frozen bits, respectively, in memory locations. Further, this module routes left-to-right and right-to-left messages based on the stage processing in the ongoing decoding iteration and generates left routed (LR) and right memory (RM) LLRs. Additionally, the PE processes LR and RM LLRs for polar decoding to generate the PFB LLRs.

In the reconfigurable decoder architecture, the polar decoding modules remain in standby mode during the LDPC decoding process to conserve energy and vice-versa for the polar decoding process. In this work, such LDPC/polar decoding process continues for 10 decoding iterations to deliver the hard-decision (HD) decoded bits. Eventually, as the maximum number of decoding iterations is exhausted, the HDM unit is activated to store decoded bits that are obtained from MDR unit, corresponding to the transmitted LDPC/polar codes, as shown in Fig. 5.4.

5.3.2 Memory and Data Routing Unit

The proposed hardware architecture of MDR unit has been presented in Fig. 5.5. Here, four quantized channel-LLRs (each of 6 bit) are processed by the normalization unit to generate normalized-LLRs whose values range between -31 to 31. These LLRs are steered into the

clocked de-multiplexer unit (CDU) that routes and buffers all the incoming normalized-LLRs until a full LDPC/polar frame is not received at the input side of CDU.

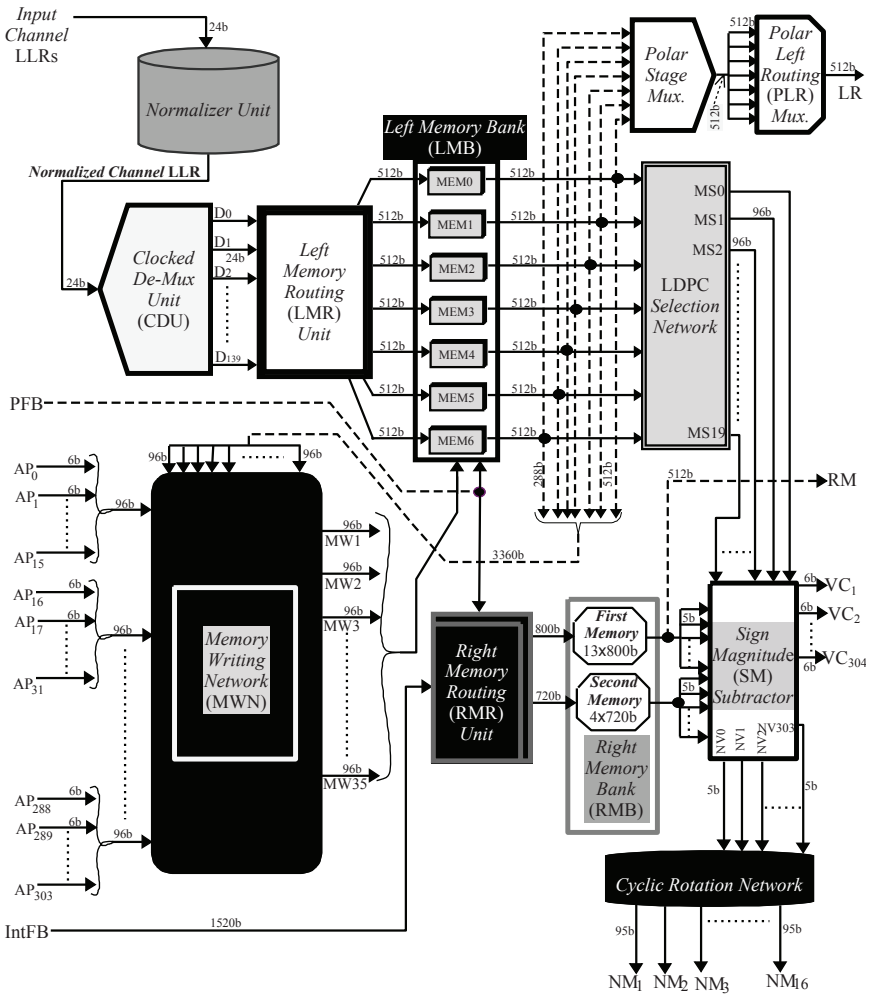


Figure 5.5: Proposed memory-efficient architecture of memory and data routing (MDR) unit for reconfigurable LDPC/polar decoder.

For LDPC decoding, all the a-posteriori LLRs (represented as AP_0–AP_{303}) from the PE and left memory bank (LMB) outputs are fed to memory writing network (MWN), as shown in Fig. 5.5. This MWN unit comprises of various multiplexers that updates all the LMB outputs with the updated AP-LLRs based on the current row processing of BG matrix and the updated LLRs (MW1–MW35) that are concatenated (referred as LDPC feedback

(LFB) LLRs) and fed to the left memory routing (LMR) unit. Therefore, all buffered-LLRs at the output side of CDU and LFB LLRs from MWN are simultaneously routed by the left memory routing (LMR) unit. This LMR passes channel-LLRs for first decoding iteration whereas it routes LFB LLRs for the remaining decoding iterations. Specifically, this LMR unit segregates these incoming LLRs into several groups of LLRs, as presented in Fig. 5.5. These segregated LLRs are stored in left memory bank (LMB) which is the combination of six 512-bit registers, and single memory with depth and data-width of 2 and 512 bit, respectively. Thus, LMB has cumulative depth of 8 and width of 512 bit. On the other side, intermediate feedback (IntFB) LLRs are routed by right memory routing (RMR) unit which segregates these data into two parts, as illustrated in Fig. 5.5. It also shows that such segregated LLRs are stored in right memory bank (RMB) that comprises of two memories: first memory has depth of 13 (corresponding to each row of BG matrix) with data-width of 800 bit, and second memory has depth of 4 with data-width of 720 bit. Unlike, entire RMB (including both first and second memories) is used for storing the intermediate LLRs for LDPC decoding. This method of memory organization eliminates the requirement of decompression units in conventional LDPC decoder [41] and thus leads to lesser memory consumption in the proposed unified LDPC/polar decoder-architecture. Similarly, LMB outputs are fed to LDPC selection network (LSN) that selects LLRs based on the BG-matrix layer scheduling. Hence, RMB and LSN outputs are segregated into LLRs and thereafter, applied to the sign-magnitude (SM) subtractors for subtraction (VN updation) and normalization of such LLRs. These SM subtractors also mitigate the overflow problem of our reconfigurable decoder-architecture. Subsequently, the normalized subtracted-output is applied to the cyclic rotational network that performs the cyclic rotation based on the BG matrix layer elements. Hence, VC_1–VC_{304} and NM_1–NM_{16} values from SM-subtractors and CR-unit, respectively, are fed to PE as outputs from MDR, as shown in Fig. 5.4 and 5.5 for LDPC decoding process.

In the polar decoding process, all buffered-LLRs at the output side of CDU and PFB LLRs from PE are routed by the LMR unit, as shown in Fig. 5.5. This LMR unit routes the

CDU-LLRs for first decoding iteration, whereas PFB-LLRs are routed for the remaining decoding iterations. Further, the LMB stores these LMR LLRs into the memory locations corresponding to the polar stage processing. As discussed earlier, LMB has cumulative depth of 8 with width of 512 bit. Its organization depends on the m+1 stages of polar code and provides the provision of memory sharing between LDPC and polar decoding processes. In addition, such LMB has the storage of LDPC/polar channel-LLRs and left-to-right processed LLRs for polar codes. In the MDR architecture, LMB simultaneously provides all the LLRs within a single clock cycle to be processed by polar-stage (PS) and polar-left-routing (PLR) multiplexers, corresponding to the polar decoding stage. Here, the output of PLR multiplexer (i.e. LR) is transferred to PE for polar decoding, as shown in Fig. 5.5. On the other side, the RMR unit routes the max LLR, zero, and PFB value for last, first, and intermediate stages, respectively. Further, these RMR-routed values are stored in RMB memory corresponding to the polar stage processing. Only first RMB memory is used to store right-to-left processing LLRs for polar decoding. Additionally, the outputs from the first memory of RMB are also transferred to PE for polar decoding process.

Hence, suggested LMB and RMB memory organizations also facilitate the sharing of storage requirements for channel as well as intermediately-processed LLRs for LDPC decoding and right-to-left cum left-to-right LLRs for polar decoding, respectively. Therefore, aforementioned storage sharing of LLRs and memory organization for LDPC and polar decoding has enhanced the memory efficiency of the proposed reconfigurable channel-decoder architecture.

5.3.3 Processing Element

For the LDPC decoding, the proposed PE has been designed to compute check-nodes (CNs) operations, as shown in Fig. 5.6. The normalized (NM) LLRs from MDR unit are fed to z number of LDPC min-sum (LMS) modules in the PE. Each of these LMS units is fed with 19 LLRs (concatenated into 95-bit bus where each LLR has a width of 5 bit). The NM LLRs are splitted into magnitude and sign values that are processed by magnitude-unit and

sign-unit, respectively, as shown in Fig. 5.7. In the former module, magnitudes (i.e. 4 bits

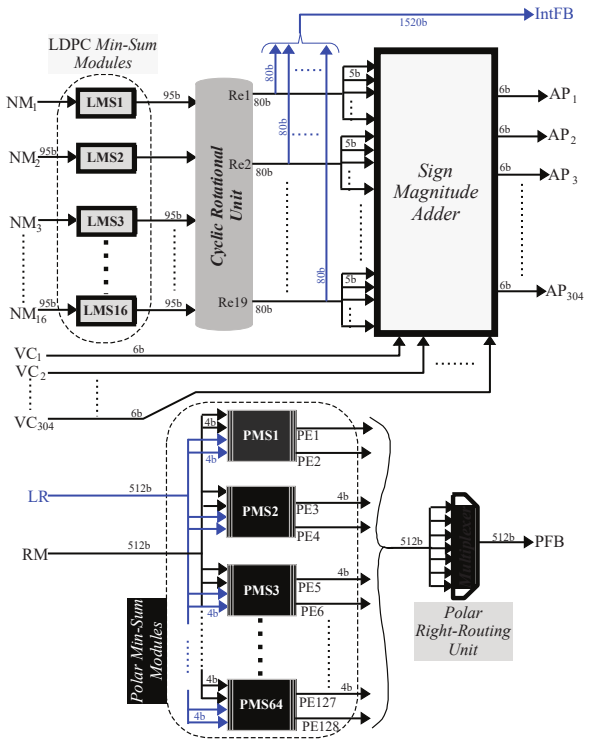


Figure 5.6: Proposed hardware architectures of processing element for our LDPC/polar reconfigurable decoder.

from each LLR) of all the LLRs are compared to generate two minimum values: min_1 and min_2, among all the 19 LLR magnitudes. Following that, equalizers and multiplexers are used for replacing the entire LLR magnitudes index with the minimum value, except the minimum value index that will be replaced by second minimum value. Simultaneously, the sign unit performs XOR operation of all sign bits (i.e. most significant bits (MSBs)) to generate a product bit. This product bit is XOR-ed with all sign-bits to update the signs of output LLRs, as presented in Fig. 5.7. Eventually, these signs are concatenated with their corresponding updated-LLR magnitudes. This updating process is known as CN updation in LDPC decoding. Afterwards, these CN-updated LLRs are applied to the cyclic rotational unit. It rotates all these LLRs, corresponding to the BG matrix layer elements, and sends them to the sign magnitude (SM) adder that generates the AP-LLRs (i.e. belief updations). These updated AP-LLRs are fed back to MDR unit for the next decoding iteration.

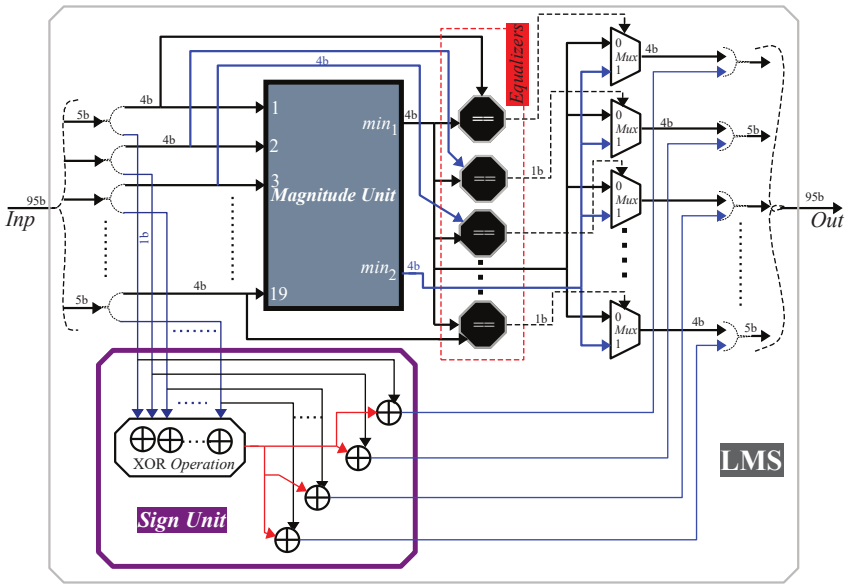


Figure 5.7: Proposed hardware architectures of LDPC min-sum unit for our LDPC/polar reconfigurable decoder.

For polar decoding, the proposed PE computes left-to-right and right-to-left operations, as shown in Fig. 5.8. The LR and RM LLRs generated from MDR unit are segregated into 256 LLRs (each LLR of 4 bit) and processed by 64 polar min-sum (PMS) modules, as illustrated in Fig. 5.6. Every PMS module performs the min-sum operation for left-to-right or right-to-left stages, depending on selection signal value of preceding multiplexer. Eventually, all the updated LLRs for polar decoding from the PMS modules are concatenated and applied to the polar right-routing unit for the data synchronization. Such polar synchronized data is then routed back to MDR unit for the next decoding iterations in the proposed reconfigurable decoder.

5.3.4 Timing Analysis

For the LDPC decoding, an overall timing diagram for VN, CN, and AP-LLR operations has been presented in Fig. 5.9 (a). Initially, the channel-LLRs are loaded in the memories of MDR by asserting high the L_{Mems} signal. Thereafter, L_{Mems} and R_{Mems} signals are asserted high to provide the LLRs that are corresponding to ongoing BG-matrix layer. Furthermore,

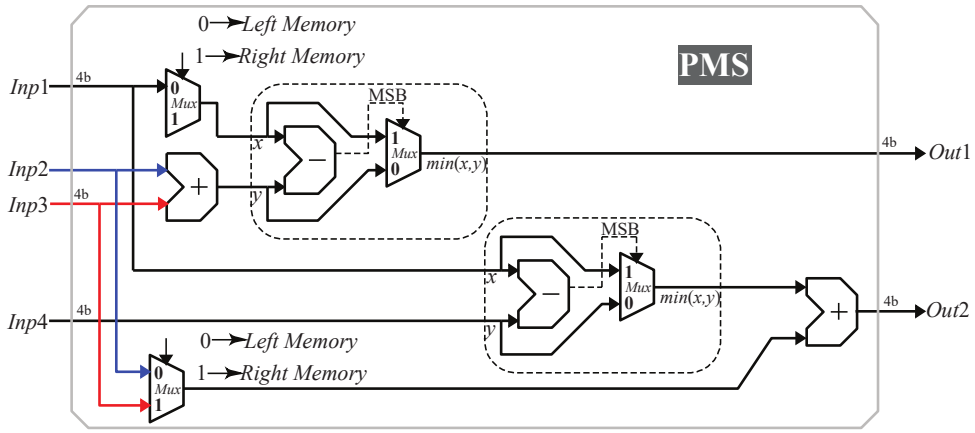


Figure 5.8: Proposed hardware architectures of polar min-sum unit for our LDPC/polar reconfigurable decoder.

VN, CN, and AP-LLRs operations are performed that correspond to specific BG-matrix layers, as shown in Fig. 5.9 (a). Such updating process follows the non-overlapping message scheme to deliver enhanced decoding performance. After the computations of AP-LLRs, they are written back to MDR memories by triggering low the L_{Mems} signal, as illustrated in Fig. 5.9 (a). Hence, the aforementioned processing continues till the last m_b layer of BG matrix, and this marks the end of a single decoding iteration. Such processes are repeatedly carried out for 10 decoding iterations. After which, the HDM is activated for storing all the sign-bits of updated AP-LLRs in a single clock-cycle, referring Fig. 5.4. Finally, this HDM delivers 16 decoded bits in every clock cycle, as presented in Fig. 5.9 (a).

Similarly for polar decoding, another timing diagram is shown in Fig. 5.9 (b) where the channel LLRs are first stored into MDR memories by triggering high the L_{Mems} signal. Following that, on applying the high values to L_{Mems} and R_{Mems} signals, MDR memories send LLRs for the computation of right-to-left messages that are stage-wise propagated from m^{th} to 1^{st} stage, which marks the completion of right-message computations. Thereafter, the left-to-right messages are stage-wise propagated from 1^{st} to $(m+1)^{th}$ stage that indicates the completion of left-message computations, as illustrated in Fig. 5.9 (b). Thus, a combination of right-message and left-message computations are considered as the completion of a single decoding iteration for polar decoding. Our channel decoder repeats 10 such decoding

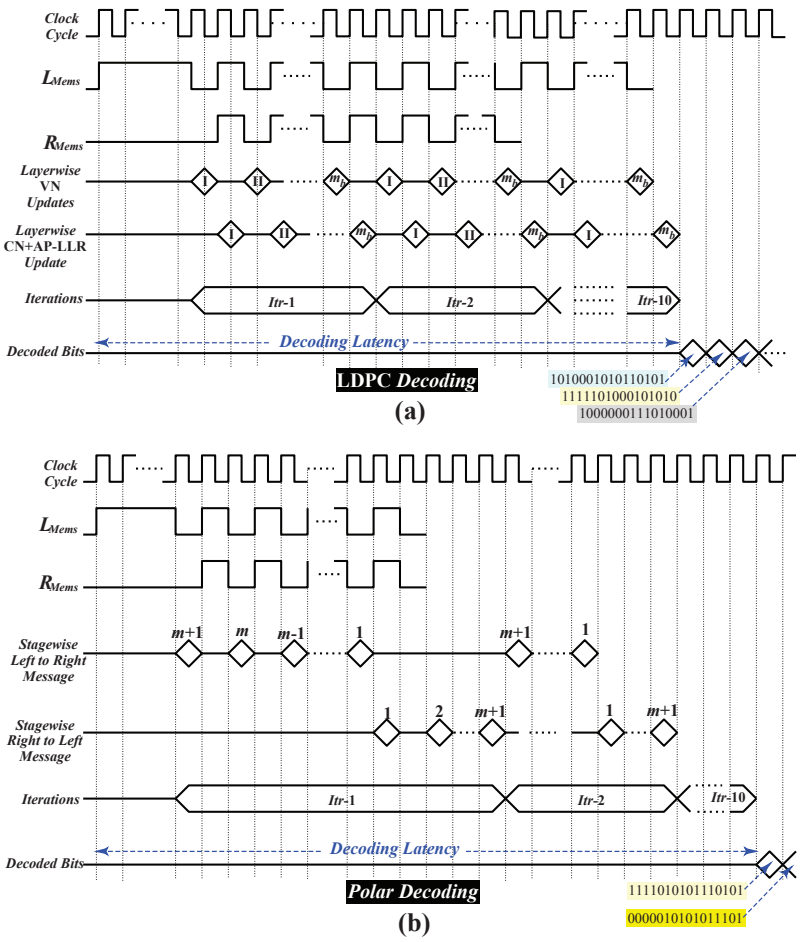


Figure 5.9: Timing diagrams for (a) LDPC and (b) polar decoding processes of the proposed reconfigurable channel-decoder architecture.

iterations to decode the polar codes. Subsequently, HDM is activated and stores all the sign-bits of updated LLRs, and finally provides 16 decoding bits at the output in every clock cycle, as presented in Fig. 5.9 (b).

5.4 Fabricated Chip Measurement Results

5.4.1 Chip Prototype and Packaging

In this work, an ASIC chip of the proposed reconfigurable LDPC/polar decoder has been fabricated in the united microelectronics corporation (UMC) 110 nm-CMOS tech-

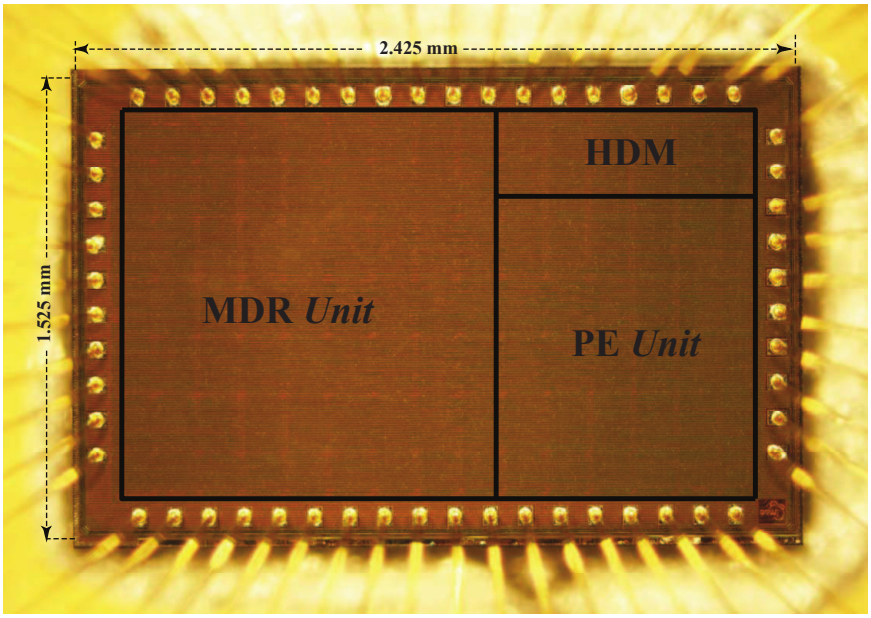


Figure 5.10: Die-micrograph of the fabricated ASIC chip for the proposed reconfigurable LDPC/polar decoder in UMC 110nm-CMOS technology node.

nology node. This decoder architecture is designed using the standard Verilog hardware-description-language (HDL) code which is functionally verified and gate-level synthesized, using the standard cell libraries of UMC 110nm-CMOS process. Here, functional verification, synthesis, static timing analysis, and power estimation have been performed with the aid of standard EDA tools from Synopsys. Furthermore, the physical design process is carried out with the Cadence EDA tools. The fabricated decoder chip is capable of decoding four code-rates for LDPC code and a single code-rate for polar code, based on the specifications of 5G-NR standard for URLLC and mMTC applications. [1]. Die-micrograph of the fabricated ASIC chip is shown in Fig. 5.10. It occupies a die area of 3.69 mm^2 and a core area of 1.96 mm^2. This chip is capable of operating up to a maximum clock-frequency of 72.7 MHz. Detailed features of our fabricated decoder chip are presented in Table 5.1.

On the other hand, our aforementioned ASIC die has been QFN packaged with 56 digital input/output (I/O) pads, as shown in Fig. 5.11 (a). These pads are used for interfacing various I/O signals of our chip, as clearly illustrated in Table 5.2. It also presents type and

Table 5.1: ASIC Chip Features of the Proposed Reconfigurable LDPC/Polar Decoder for 5G-NR Standard.

Chip Architecture	Reconfigurable Channel Decoder
Standard	5G New-Radio
Technology Node	UMC 110 nm-CMOS
IO Pads	56
Maximum Clock Frequency	72.7 MHz
Die Area	3.69 mm^2
Core Area	1.96 mm^2
IO Power Supply	3.3 V
Core Power Supply	1.2 V
Chip Packaging	Quad Flat No-lead (QFN)

Table 5.2: Classification and Quantification of Digital I/O Pads of Our ASIC Chip.

Signalling Type	**Pad Type**	**No. of Digital Pads**
Clock	Input	1 Pad
Initialize Reset	Input	1 Pad
LDPC/Polar Switching	Input	1 Pad
LDPC CR Selection	Input	2 Pads
Input LLRs	Input	24 Pads
Pads Supply Voltage	Input	2 Pads
Pads Ground Voltage	Input	2 Pads
Core Supply Voltage	Input	3 Pads
Core Ground Voltage	Input	3 Pads
Decoded Bits	Output	16 Pads
Done	Output	1 Pad

quantity of pads used for interfacing different I/O signals. This chip interfaces 39 input signals including power supplies, and generates 16 decoded-bits with a done signal, as the final outputs of the reconfigurable decoding process. Finally, the proposed decoder chip has been mounted on the printed circuit board (PCB) with its 2.54 mm connectors, as shown in Fig. 5.11 (b), for testing and validation in real-world test scenarios.

5.4.2 Test Setup and Functional Validation for Chip Characterization

Both schematic and real-world snapshots of the test setup for post-silicon validation of our fabricated decoder ASIC have been presented in Fig. 5.12. Detailed operation and information flow are clearly depicted in the schematic, as presented in Fig. 5.13. In MATLAB environment, the 5G-NR test cases are generated with k information-bits that are LDPC or

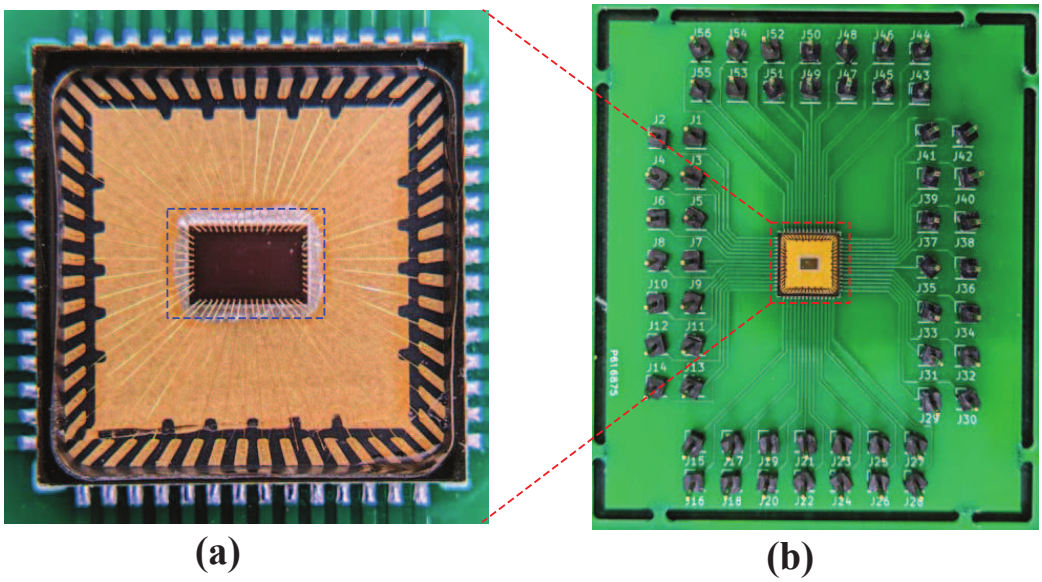


Figure 5.11: Fabricated ASIC chip of the proposed reconfigurable LDPC/polar decoder in UMC 110 nm-CMOS technology node: (a) QFN packaged chip, and (b) ASIC-chip mounted on the PCB for characterization.

polar encoded into n codeword-bits such that $n>k$. These n encoded bits are transmitted over the AWGN channel environment that introduces noise in the encoded bits which are received at the receiver side as noisy LLR samples, as shown in Fig. 5.13. Here, each of these noisy LLRs is fixed-point quantized into 6-bit value and is stored in the group of four LLRs (i.e. 24-bits) into the coefficient file (with an extension of *.coe*), which is interfaced with the FPGA-1 board. This file initializes the block random-access-memory (BRAM) of FPGA-1 (Xilinx Nexys FPGA-board), as shown in Fig. 5.13. From such BRAM, LLR groups (each of four LLRs) are fetched sequentially at the clock frequency of 72.7 MHz. It is an on-board clock signal that is generated with the help of the FPGA clocking wizard intellectual property (IP) core.

Furthermore, BRAM outputs and generated clock signal are configured to interact with PMOD connectors of FPGA-1 board, as presented in Fig. 5.12 and Fig. 5.13. Similarly, another FPGA-2 board has been configured with all the input controlling signals (like LDPC/polar switching, LDPC CR-selection and reset signals, referring Table 5.2) for the

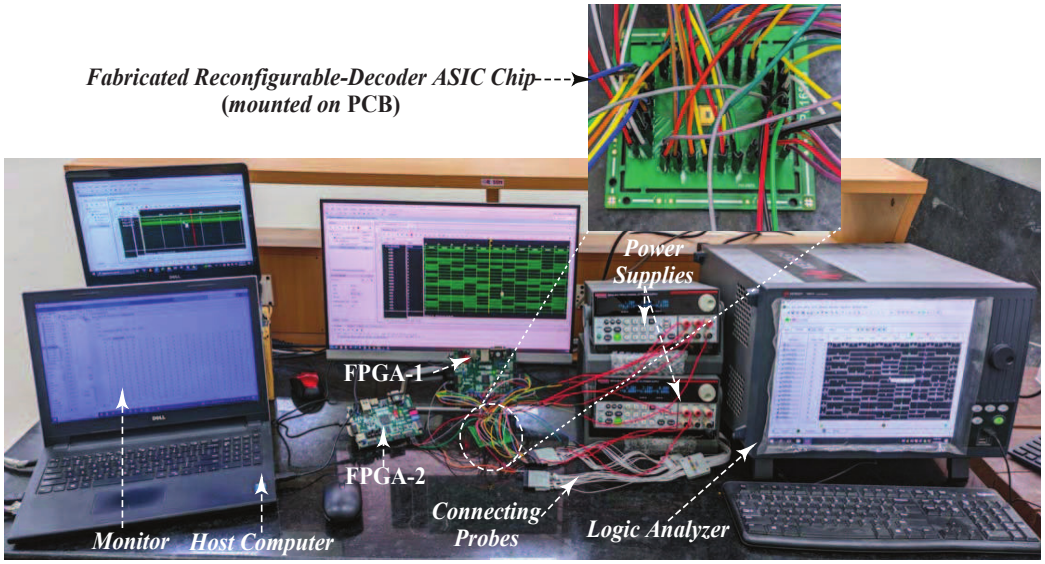


Figure 5.12: Snapshot of real-world test setup for characterizing the fabricated ASIC chip of the proposed reconfigurable LDPC/polar decoder.

decoder chip at 72.7 MHz of clock frequency to interact with PMOD connectors. Hence, FPGA-1 board delivers all the input LLRs and a clock signal of 72.7 MHz clock frequency to our decoder ASIC-chip. In addition, all input control signals are fed to this chip by FPGA-2 board, as shown in Fig. 5.13. Finally, supply voltages of 3.3 V and 1.2 V are provided to I/O pads and core of the chip, respectively.

Henceforth, the proposed reconfigurable decoder-chip processes all these input signals (from FPGA-1 and FPGA-2 boards) at an operating clock frequency of 72.7 MHz and generates error-free decoded-bits, as outputs which are displayed on the screen of Keysight-16861A 32-channels logic-analyzer via connecting probe (Keysight N2140A probe), as shown in Fig. 5.14. Once these decoded-bits are obtained from our chip, they are validated with the simulated decoded-bits from the Matlab environment (i.e. displayed on the monitor of host computer).

Such validation is carried out for all the standard code-rates of LDPC and polar codes, compliant to the URLLC and mMTC 5G-NR applications. Specifically, for the functional validation, quantized input LLRs (from the channel) are fed to the proposed decoder chip via BRAM of the FPGA-1 board, as discussed earlier. Simultaneously, the necessary control

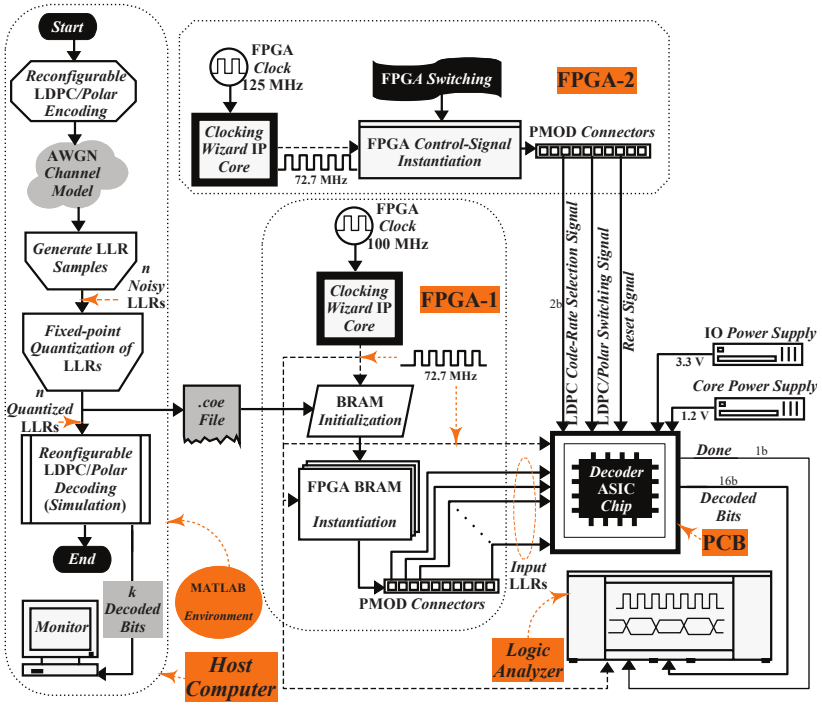


Figure 5.13: Schematic-representation of the test setup for characterizing the fabricated ASIC chip of the proposed reconfigurable LDPC/polar decoder.

signals for reconfiguration are generated from FPGA-2 board and fed to the decoder chip. At the clock frequency of 72.7 MHz, our decoder ASIC-chip processes such input LLRs for either LDPC or polar decoding, resulting into decoded-bits that are viewed on the logic analyzer screen. Such functionality of the fabricated ASIC-chip has been verified for more than 40 5G-NR standard based test cases with various SNR values at different code-rates for LDPC code, and also separately for the polar code.

However, this thesis presents the decoded bits of LDPC code with the code-rate of 3/4, as shown in Fig. 5.14 (a & b). Former Fig. 5.14 (a) shows the simulated decoded bits from the Matlab environment that are displayed on the monitor of the host computer, as depicted in the test-setup of Fig. 5.12. Furthermore, the measured values of decoded bits from our decoder chip are displayed on the screen of 32-channel logic analyzer, as shown in Fig. 5.14 (b). Here, the proposed decoder is capable of delivering 16 decoded bits in every clock cycle.

As shown in Fig. 5.14 (a & b), both simulated and measured values of decoded bits are matching, and hence this validates the correct functionality of our decoder ASIC. A similar validation process has been carried out for the rest of the code-rates of LDPC and polar codes.

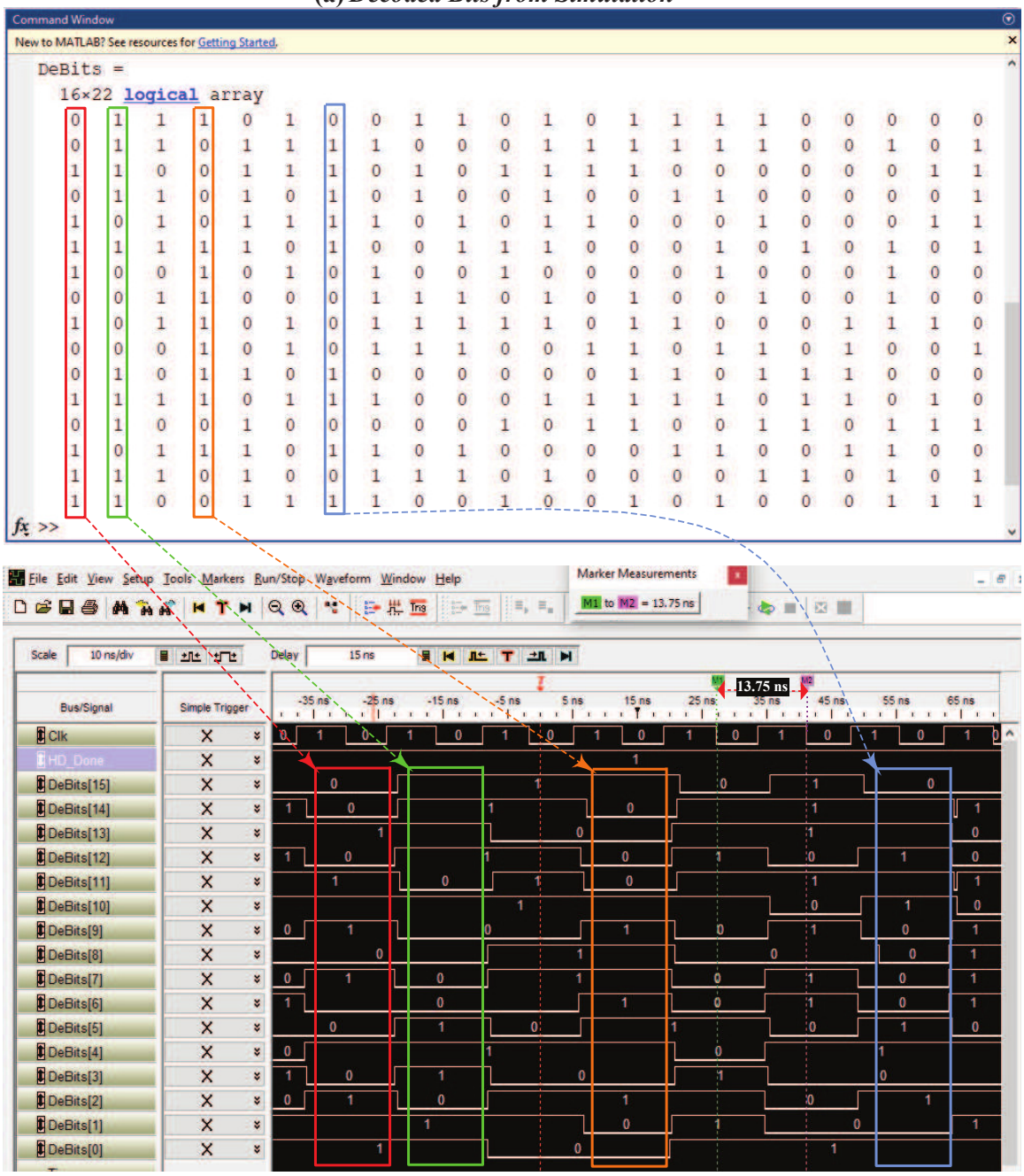
5.4.3 Post-Silicon Power Measurement and Throughput Estimation

In this work, power measurement is performed at the supply of 1.2 V while operating at 72.7 MHz of clock frequency. Both power and throughput measurements are carried out, based on the LDPC standard frame-sizes of 560, 512, 464, and 432 for 2/3, 3/4, 5/6, and 8/9 code-rates, respectively. For the polar code, these measurements are performed with a frame length of 128 and a code-rate of 1/2. Thus, various bar graphs indicate the measured power consumptions of our decoder chip, while decoding under different channel-SNR values for LDPC (at 5G-NR code-rates) and polar codes are presented in Fig. 5.15.

It also indicates that the total power consumed for decoding the LDPC code with a 2/3 code-rate at a SNR of 1.6 dB can be reduced up to 2.31× with respect to code-rate of 8/9 at 7.1 dB. It shows that the power requirement reduces, as the code-rate of LDPC code increases due to lower active-processing layers. Consequently, chip power consumption alleviates at the higher values of channel SNR which clearly indicates that the good environmental conditions result in lesser noisy decoded-bits. Note that the power consumption of higher code-rates is lesser than the lower code-rates due to the lower code-length of the LDPC codes. Similarly, the plots of power consumption versus operating clock frequencies that depict the surge in power requirement with increasing clock frequency have been presented in Fig. 5.16.

As demonstrated earlier, our decoder chip is capable of operating with a clock frequency of 72.7 MHz, at which it delivers a maximum throughput of Θ_T=3.350 Gbps. Such throughput has been estimated as $\Theta_T = (\lambda \times n \times f_{Clk})/(I \times \mathbb{C} \times \mathbb{L})$. Here, λ=16 number of output decoded-bits are generated in every single clock cycle, n is the code length, F_{Clk} indicates the clock frequency, I=10 decoding iterations, $\mathbb{C}$=3 number of clock cycles to complete the

(a) ***Decoded Bits from Simulation***



(b) ***Decoded Bits from Fabricated Decoder ASIC-Chip***

Figure 5.14: Snapshots of (a) simulated decoded bits from the Matlab simulation, displayed on the monitor of the host computer, and (b) measured output decoded-bits (generated at 72.7 MHz of clock frequency) from the fabricated ASIC-chip of the proposed reconfigurable LDPC/polar decoder, displayed on the logic analyzer screen. Note: above outputs are obtained after decoding the LDPC code of the code-rate 3/4.

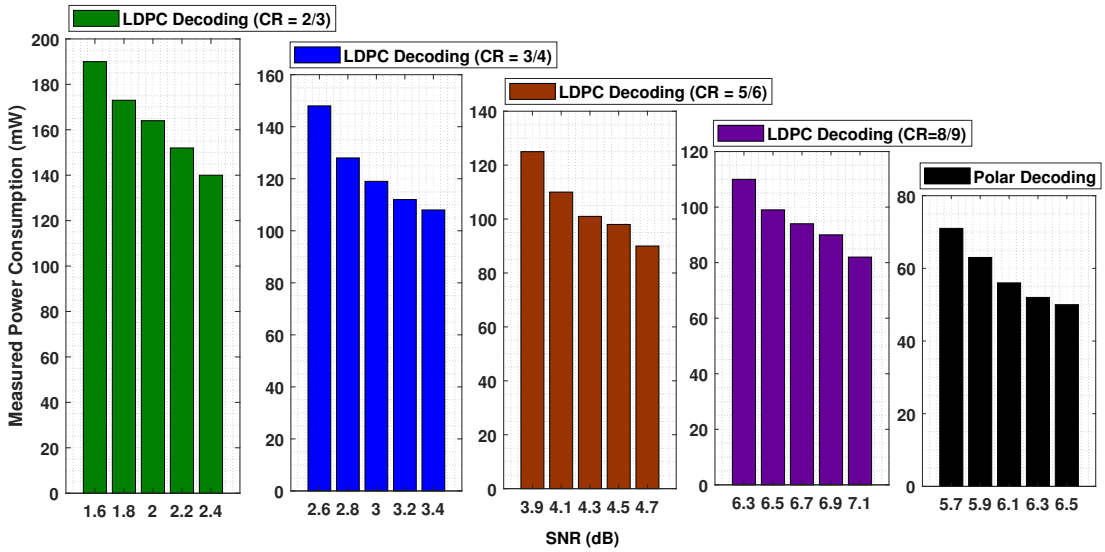


Figure 5.15: Analysis of the measured power consumptions of the proposed LDPC/polar decoder ASIC-chip at different channel SNRs, while decoding LDPC (at 5G-NR specified code-rates) and polar codes. Note: these are the total power requirements for our decoder-chip to achieve a FER of 10^{-4}. As seen in Fig. 5.2, decoding of various LDPC/polar codes with multiple code-rates achieves such a FER at different range of SNRs.

computations of single layer, and $\mathbb{L}$ represents number of active layers that varies with respect to code-rates. Hence, $\mathbb{L}$ has the values of 13, 10, 7, and 5 for LDPC code-rates of 2/3, 3/4, 5/6, and 8/9, respectively, for 5G-NR standard [41]. Furthermore, Fig. 5.17 represents the measured throughput of the proposed reconfigurable decoder chip with respect to various clock frequencies for different code-rates. These plots indicate that the throughput increases with higher code-rates due to the lower active layers ($\mathbb{L}$). Hence, based on the aforementioned achievable throughput for polar code and at different code-rates of LDPC code, our decoder is suitable for the mMTC and URLLC applications for 5G-NR wireless communication standards.

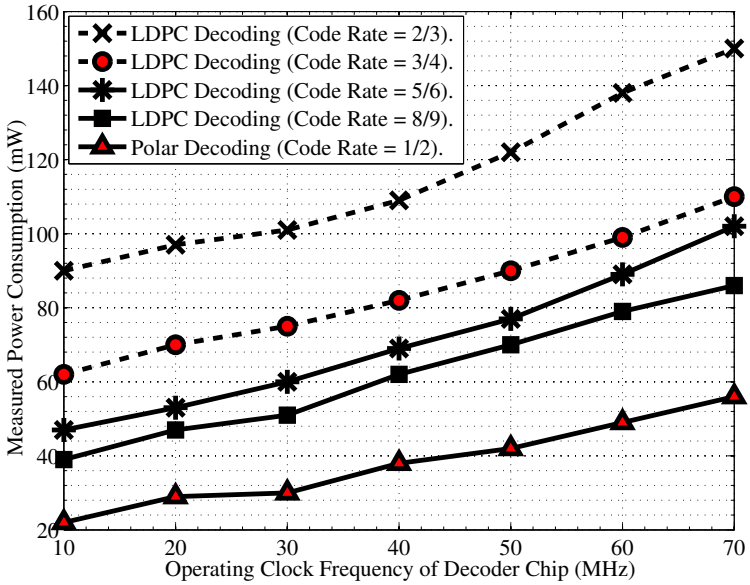


Figure 5.16: Analysis of the measured power consumptions of the proposed LDPC/polar decoder ASIC at different operating clock frequencies.

5.5 Comparisons

The measured results obtained from our fabricated reconfigurable LDPC/polar decoder-chip (in UMC 110 nm-CMOS technology node) are compared with the relevant state-of-the-art works, as presented in Table 5.3. It lists both types of implementations from the literature: post-fabrication measured results (represented as 'Silicon' in Table 5.3), and synthesized cum post-layout simulated outcomes (represented as 'Synthesis' in Table 5.3). For fair comparison, all the technological parameters have been scaled with reference to 110 nm-CMOS technology node. Referring Table 5.3, it can be observed that our reconfigurable decoder delivers 4× and 5.64× better throughput than the implementations of reconfigurable decoders, reported by Bei et al. in [3] and S. Cao et al. in [4], respectively. In addition, the proposed decoder chip consumes 1.9× lesser latency than the reported work [47] and occupies 15.8% lower area than the smallest silicon area, reported by a contemporary work [4]. Furthermore, referring Table 5.3, total power consumption of our decoder is 50.2% lower than the reconfigurable decoder of [3].

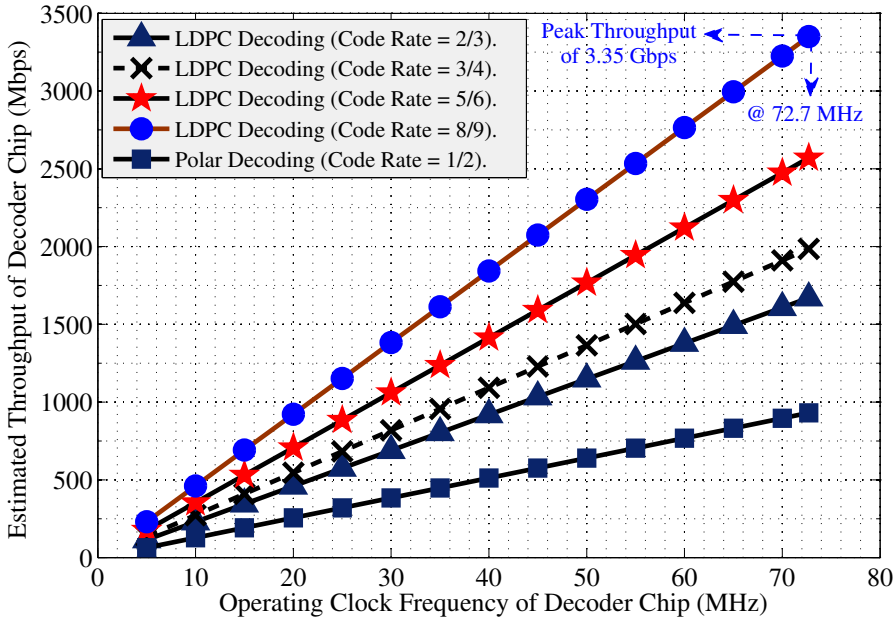


Figure 5.17: Variation of the achievable throughput of the proposed LDPC/polar decoder ASIC-chip at different operating clock frequencies.

To justify the fair comparison, significant figure-of-merits like energy and hardware efficiencies are presented in Table 5.3. Here, energy-efficiency indicates the amount of energy required to decode a single bit. Hence, our decoder chip possesses better energy-efficiency than all the reported implementations in Table 5.3 and it has specifically 10% higher energy-efficiency than the state-of-the-art work [47]. Similarly, hardware-efficiency is another FOM that represents the peak achievable throughput of the decoder per unit area. Referring Table 5.3, it can be observed that the proposed decoder chip delivers competitive hardware-efficiency among all the reported works. Specifically, it has achieved 2.5× better hardware-efficiency than the highest value of state-of-the-art work, reported by S. Yun et al. [45]. Based on the aforementioned results and discussion, our reconfigurable decoder delivers better hardware as well as energy efficiencies, and it is also compliant to the specifications of the 5G-NR wireless communication standard.

Fundamentally, the total area occupied by the channel-decoder is a function of code-

Table 5.3: Comparisons of Measured Results of Proposed Reconfigurable Channel-Decoder with State-of-the-Art Works.

Specifications	This Work	SSC Letters 2022 [3]	TVT 2021 [4]	JSSC 2020 [47]	TCAS-II 2022 [45]	TCAS-I 2022 [5]
Implementation Type	**Silicon**	Silicon	Synthesis	Silicon	Synthesis	Synthesis
CMOS Technology Node (nm)	**110**	40	55	40	65	65
Decoding Codes	**LDPC & Polar**	LDPC & Polar	LDPC & Polar	Polar	LDPC	LDPC
Quantized Bits (bits)	**6**	6	6	6	8	8
Total Memory (kb)	**12.4**	192	173.8	–	–	208.8
Pipeline Stages	**2**	4	5	8	–	6
Supply Voltage (V)	**1.2**	0.9	1.08	0.9	1.1	1.1
Max. Clock Freq. (MHz)	**72.7**	180	300	430	238	500
Code Length (bits)	**560**	6400	2912	1024	1664	10368
Decoding Schedule	**Layered**	Layered	Flooding	–	Layered	Layered
Iterations	**10**	5	5	–	10	3
Throughput (Mbps)	**3350**	$832^{\dagger}$	$594^{\dagger}$	$1180^{\dagger}$	$2360^{\dagger}$	$3682^{\dagger}$
Latency (μs)	**0.12**	0.81	0.84	0.23	–	2.81
Core Area (mm^2)	**1.96**	$15.65^{\ddagger}$	$2.2^{\ddagger}$	$4.81^{\ddagger}$	$3.43^{\ddagger}$	$16.43^{\ddagger}$
Power (mW)	**198**	$297^{\oplus}$	–	$76.08^{\oplus}$	$162^{\oplus}$	$491.5^{\oplus}$
Energy Efficiency (pJ/bit)	**59**	356	–	64	72	133.48
Nor. Ener. Eff. (pJ/bit/itr)	**5.9**	71.2	–	–	7.2	13.34
Hardware Eff. (Gb/s/mm^2)	**1.70**	0.053	0.55	0.245	0.68	0.22
Decoding Algorithm	**OMS$^{\alpha}$**	BP$^{\beta}$	Min-Sum	ST SCL$^{\gamma}$	NMS$^{\delta}$	OMS$^{\alpha}$
Multiple Code-Rates	**Yes**	Yes	Yes	Yes	No	Yes
BER for LDPC Rate 2/3	**10^{-4} at 2.1 dB**	10^{-4} at 1.4 dB	–	–	–	–
BER for LDPC Rate 3/4	**10^{-4} at 3 dB**	–	–	–	–	–
BER for LDPC Rate 5/6	**10^{-4} at 4.6 dB**	–	–	–	10^{-6} at 4.5 dB	–
BER for LDPC Rate 8/9	**10^{-4} at 6.2 dB**	–	–	–	–	10^{-4} at 4.3 dB
BER for Polar Rate 1/2	**10^{-5} at 5.5 dB**	10^{-4} at 4.6 dB	–	10^{-6} at 4 dB	–	–

- †: Normalized throughput, scaled for 110 nm-CMOS process, with the factor of S where S is a ratio of 110 nm with lower technology node (in nm); ‡: Core area has been scaled for 110 nm by the factor of S^2; ⊕: Power has been scaled for 1.2 V, by the factor of u^2 where u is a ratio of 1.2 V with the lower technology-node supply voltage (in V); α: Offset Min-Sum; β: Belief Propagation; γ: Split-Tree Successive Cancelation List; δ: Normalized Min-Sum.

length. Hardware requirements of computational unit and memory consumption (for storing LLRs) of the decoder surge with the increasing value of code-length. In the proposed reconfigurable decoder, the hardware resources of the computational unit remain constant and it does not increase with the code-length, as explained earlier in Section III. However, its hardware requirement is a function of memory for storing the LLRs. Hence, the memory consumption of our decoder is mathematically formulated as

$$M_P = n_P \left(1 + 2 \cdot q_b (1 + \log_2 n_P)\right), \tag{5.1}$$

$$M_L = n_L (1 + q_b) + 2(q_b + 10)(n_L - k_L), \tag{5.2}$$

$$M_R = max(M_P, M_L). \tag{5.3}$$

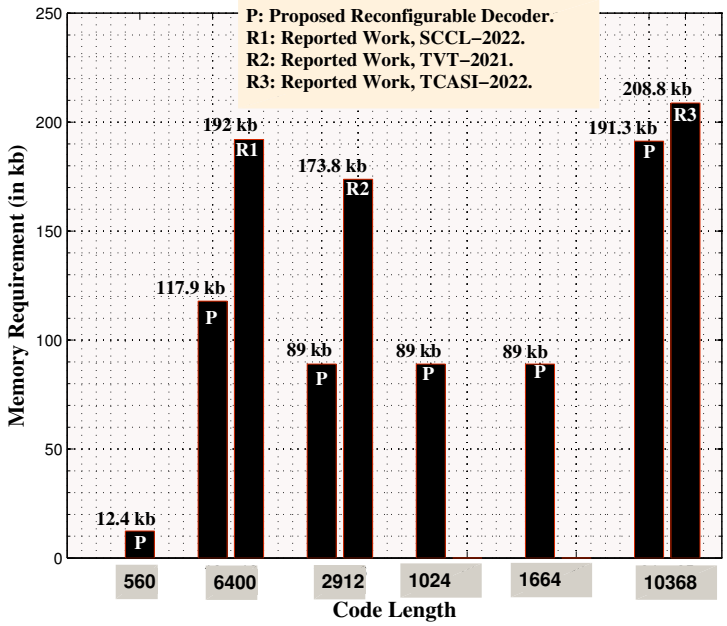


Figure 5.18: Comparative analysis of memory requirements versus code-lengths of the proposed and the reported (R1:[3], R2:[4], and R3:[5]) channel-decoders.

Here, M_P and M_L represent the memories needed for standalone polar and LDPC decoding, respectively. Nevertheless, suggested reconfigurable decoder consumes the

memory (denoted by M_R) that is maximum M_P and M_L. Number of quantization bits has been denoted by q_b. In addition, n_L and n_P represent the code-lengths of LDPC and polar codes, respectively. The memory requirements of reported works for various code-lengths are presented in Table 5.3. It also shows that the proposed decoder consumes the least memory, supporting the code-length of 560. Nevertheless, based on (5.3), memory required by our design to support these reported code-lengths have been computed and compared in Fig. 5.18. It is to be noted that the memory consumption of our decoder for the storage of LLRs is lowest among the reported architectures. Therefore, area consumption of the proposed decoder architecture remains comparatively lesser at these higher reported code-lengths, as listed in Table 5.3. Hence, we can extend the suggested decoder-architecture to support any higher code-lengths and it still consumes less area. Nevertheless, the fabricated decoder chip has been designed for supporting mMTC and URLLC applications of 5G-NR wireless commnication standard that requires a code-length of 560 [1].

5.6 Summary

This chapter presented a new reconfigurable LDPC/polar channel-decoder architecture that has been designed for 5G-NR mMTC and URLLC applications. We presented a reconfigurable channel-decoding technique that illustrates LDPC, polar and reconfigurable decoding processes. It also depicted the usage of mutually shared memory and operational units of LDPC and polar decoding operations. Further, a Monte Carlo simulation for the transmission of 10^8 LDPC and polar encoded bits has been carried out in this chapter. We performed a comprehensive FER performance analysis for (560, 352), (512, 352), (464, 352), and (432, 352) LDPC codes for the code-rates of 2/3, 3/4, 5/6, and 8/9, respectively, with an expansion factor of z = 16. Similarly, a FER performance analysis for (128, 64) polar code for the code rate of 1/2 has been carried out. It is observed that the FER performance improves with decoding iterations over the higher value of SNR. Subsequently, this chapter presented a novel unified reconfigurable LDPC/polar channel-decoder architecture that

has been designed based on the suggested reconfigurable technique. This channel-decoder architecture is complaint to the 5G-NR mMTC as well as URLLC applications, and supports the multiple standard code-rates for LDPC codes and a single code-rate for polar codes. This channel decoder enhances hardware-efficiency by sharing memory organization. It also improves the data-flow of information and excludes the requirement of various computation modules in the proposed channel-decoder hardware. Some additional optimizations in the processing element of polar decoding has also been carried out in this chapter.

In addition, overall reconfigurable LDPC/polar channel-decoder architecture has been ASIC synthesized, post-layout simulated, and fabricated in UMC 110 nm-CMOS technology node. This fabricated ASIC is capable of decoding the four code-rates of LDPC code and a single code-rate of polar code, based on the exact specification of 5G-NR wireless communication standard. It occupies core area of 1.96 mm^2 and die area of 3.69 mm^2 that is 15.8% lower area than the reported contemporary work. The proposed ASIC has been mounted on the PCB with 2.56 mm connectors. Consequently, our reconfigurable channel-decoder fabricated chip has been functionally verified with the real-world test setup environment on the screen of Keysight-16861A 32-channels logic-analyzer via connecting Keysight N2140A probe. Therefore, these decoded bits that is obtained from fabricated chip are validated with the simulation of MATLAB environment. We have performed such validations for all the standard code-rates of LDPC and polar codes, compliant to the 5G-NR wireless communication standard.

Furthermore, we measured the power consumption for all the standard code-rates of LDPC and polar decoding. A comprehensive bar graph analysis of measured power consumption has been depicted for suggested LDPC/polar channel-decoder ASIC chip to achieve a FER of 10^{-4} at different range of SNRs. It also indicates that the total power consumed for decoding the LDPC code with 2/3 code-rate at a SNR of 1.6 dB can be reduced up to 2.31× with respect to code-rate of 8/9 at 7.1 dB. Consequently, chip power consumption at the higher values of channel SNR clearly indicates that the good environmental conditions result in lesser noisy decoded bits. Similarly, the power consumption of

fabricated ASIC at various operating clock frequencies also presented in this chapter. It is to be noted that decoder consumes more power by increasing the operating clock frequency. Our reconfigurable LDPC/polar channel-decoder consumes power of 198 mW at maximum clock frequency of 72.7 MHz for LDPC decoding of 2/3 code-rate. Similarly, the measured throughput of the proposed reconfigurable decoder chip at various clock frequencies for different code-rates has been represented in this chapter. These plots indicate that the throughput increases with higher code-rates due to the lower active layers. Hence, it can be observed that our reconfigurable decoder delivers the maximum data-throughput of 3.35 Gbps with the latency of 0.12 μs. Based on the aforementioned discussion, the suggested reconfigurable decoder delivers better hardware as well as energy efficiencies, and it is also compliant to the specifications of the 5G-NR wireless communication standard. As a result, we addressed to deliver high data-throughput and better hardware-efficient reconfigurable channel-decoder for the contemporary wireless-communication technologies.

Chapter 6

Conclusion and Future Scope

6.1 Thesis Summary

The channel decoder empowers the wireless-communication systems for reliable transmission of data over the noisy channel environments. This unit of transceiver module not only mitigates the impacts of channel impairment but also provides the enhancement in spectral-efficiency and overall network performances. Therefore, such channel decoder is a crucial component in overall wireless-communication systems. In this thesis, we focused towards the designing of various channel-decoder architectures that satisfy the high-end specifications of 5G-NR wireless communication standard. The high-end specifications of 5G-NR applications/systems have been diversified into several categories: eMBB applications are targeting to achieve high data-throughput and lower-latency devices, high spectrum efficiency is the key parameter for mMTC applications whereas the URLLC applications are focusing for high-reliability and lower-latency systems. Based on the requirements for 5G-NR applications, we addressed these research challenges of delivering the high-throughput channel decoders for 5G-NR eMBB applications.

Firstly, a conventional OMS algorithm for the decoding of 5G-NR QC-LDPC codes

has been presented in this thesis. A comprehensive BER performance analysis for this decoding algorithm that adheres to 5G-NR wireless communication standard. Further, this work also addressed the BER performances for all the standard code-rates of 5G-NR communication systems. Based on the conventional decoding algorithm, we proposed an iterative decomposition based partially-parallel QC-LDPC decoder architecture that improved the hardware-efficiency and delivered moderate data-throughput. It operated at the maximum clock frequency of 34 MHz and delivered the data throughput of 965 Mbps. In addition, a fully-parallel QC-LDPC decoder architecture has been suggested that has been designed based on replication technique where CN and VN operation unit has been replicated to reduce the latency of decoder. Following that, we presented a comprehensive description of architectural aspects for all the internal modules of this new QC-LDPC decoder which is also compliant to 5G-NR technology. It achieved a data throughput of 2.9 Gbps while operating with the 5G-NR standardized code-rate of $1/3$. Both partially-parallel and fully-parallel QC-LDPC decoder architectures support single standardized code-rate of $1/3$. These QC-LDPC decoder architectures can also be accommodate for the regulation of various standard base graph matrices such as 4G-LTE, WiMAX, Wi-Fi, etc.

Based on the proposed QC-LDPC decoder architectures, we have observed that aforementioned QC-LDPC decoders consume excessive latency and hardware-resources to satisfy the high-end specifications like latency $\leq$ 1 ms and throughput $\approx$ 10-20 Gbps, of 5G-NR wireless communication standard. Hence, only architectural optimizations are not sufficient to achieve such sofisticated 5G-NR specifications. Hence, decoding algorithmic optimizations are necessary that provide an parallel operations for CNs & VNs as well as reduces the routing complexity for the decoding of LDPC codes. Therefore, we suggested an LLRC-segregation based decoding algorithm that reduced the routing-complexity and incurred an significant reduction in hardware-efficiency. Subsequently, generic QC-LDPC decoder architecture has been presented based on suggested LLRC-segregation technique. It supports standard code-lengths ranging between 10368-26112 bits and decoded at various 5G-NR standardized code-rates of QC-LDPC code. Further, this LLRC-segregation

based overall QC-LDPC decoder architecture has been FPGA prototyped on Xilinx Zynq Ultrascale+. It could operate at a maximum clock frequency of 135 MHz and delivered a peak data-throughput of 11.02 Gbps while decoding at a code rate of 8/9. It also achieved a hardware-utilization-efficiency (HUE) of 1.95 hardware-resources/layer/Mbps and renders a highest peak data-throughput to latency ratio (PTLR) of 11723.3 Mbps/μs. So, this QC-LDPC decoder architecture supported all the standard code-rates and also satisfied the 5G-NR physical layer specifications.

Moreover, we have suggested a simplified offset min-sum (SOMS) QC-LDPC decoding algorithm to achieve hardware-efficient and high data-throughput in this thesis. This proposed algorithm alleviates the computational-complexity and also have significant reduction in the operations like memory-storage, addition, and comparisons. In addition, comprehensive performance analysis of SOMS decoding algorithm has been presented for the specification of 5G-NR wireless communication standard. It is to be noted that the proposed SOMS decoding algorithm delivered adequate FER performance for the decoding of LDPC codes. Subsequently, a high-throughput QC-LDPC decoder architecture has been presented that is designed based on the proposed SOMS decoding algorithm. This QC-LDPC decoder architecture improved the hardware-efficiency and also inculcates parallelism to enhance data-throughput with lower decoding-latency. Our proposed decoder operated at the maximum clock frequency of 128.36 MHz and delivered a peak data-throughput of 13.3 Gbps with the hardware efficiency of 1.63 hardware-resources/layer/Mbps. Additionally, this SOMS based (2592, 2304) QC-LDPC decoder has been ASIC synthesized and post-layout simulated in the UMC 90 nm-CMOS technology node. As a result, this ASIC architecture occupied an area of 6.45 mm^2 and is capable of operating at a clock frequency of 192.3 MHz to deliver a data-throughput of 9.6 Gbps. So, the proposed QC-LDPC decoding algorithm and decoder architecture are significant for the deployment of recent and future wireless-communication systems like 5G-NR and 6G technologies.

The 5G-NR wireless network enhanced the user-connectivity and digitized the diverse industrial verticals. So, the 3GPP and ITU have advocated LDPC and polar codes for

channel-coding in the release-16 for 5G-NR standard. Consequently, we also proposed a reconfigurable LDPC/polar decoding technique that has various salient features like memory-storage sharing, unified data-routing and combined min-sum operations for the decoding of LDPC and polar codes in this thesis. Further, a comprehensive FER performance analysis for LDPC and polar codes has also been presented. Moreover, a novel unified reconfiguable channel-decoder has been designed based on suggested reconfigurable LDPC/polar decoding technique that supports multiple code-rates and code-lengths of 5G-NR mMTC and URLLC applications. In addition, the proposed reconfigurable LDPC/polar decoder architecture has been ASIC fabricated in UMC 110 nm-CMOS technology node. It occupies the core area of 1.96 mm^2 and die-area of 3.69 mm^2 that is the lowest in state-of-the-art works. Moreover, this fabricated ASIC has been validated and characterized in real-world test setup environment. It operates at the maximum clock frequency of 72.7 MHz and delivers the maximum data-throughput of 3.35 Gbps with a latency of 0.12 μs. We performed comprehensive analysis of measured power consumption of our proposed reconfigurable channel-decoder to achieve a FER of 10^{-4} at different range of SNRs. This indicates that the total power consumed for decoding the LDPC code with 2/3 code-rate at a SNR of 1.6 dB can be reduced up to 2.31× with respect to code-rate of 8/9 at 7.1 dB. Consequently, ASIC power consumption alleviates at the higher values of channel SNR which clearly indicates that the good environmental conditions result in lesser noisy decoded bits. Our reconfigurable LDPC/polar channel decoder consumes maximum power of 198 mW at 72.7 MHz for LDPC decoding. In this thesis, we addressed the research challenge of delivering high-throughput and better hardware-efficient reconfigurable LDPC/polar channel decoder for 5G-NR applications.

6.2 Conclusions

In this thesis, we presented several hardware-efficient and high-throughput QC-LDPC decoder architectures that are complaint to exact high-end specifications of 5G-NR wire-

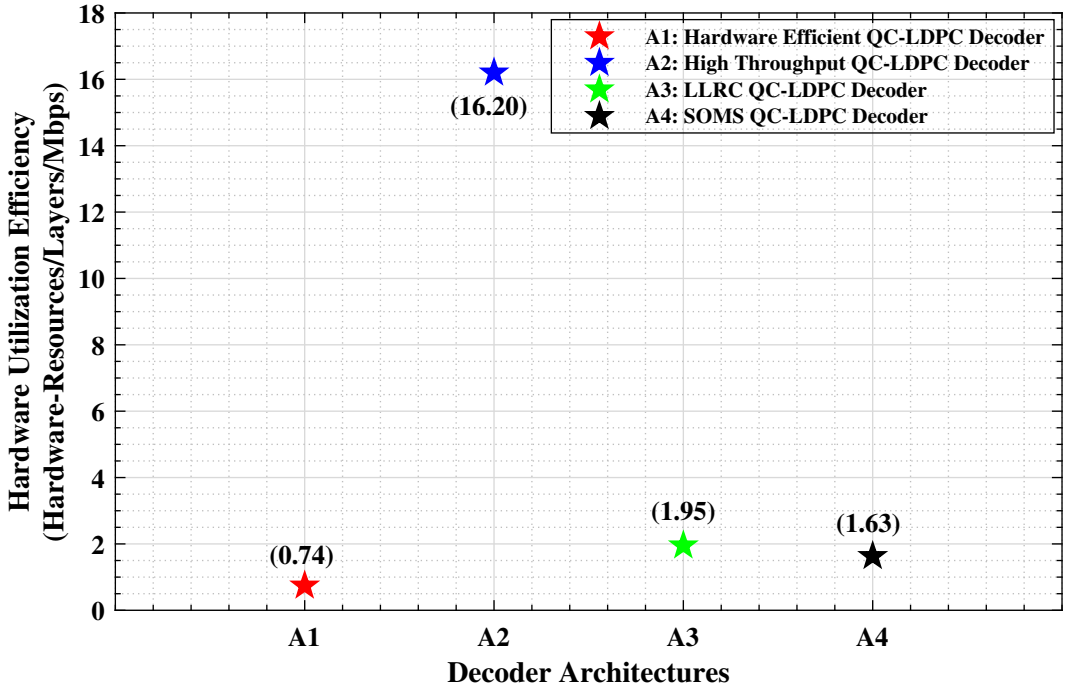


Figure 6.1: Comparisons of our proposed QC-LDPC decoder architectures based on the hardware utilization efficiency (HUE). Here, **A1** and **A2** architectures have been described in chapter 2 whereas **A3** and **A4** architectures are explained in chapter 3 and chapter 4, respectively.

less communication standard. Here, we are comparing our proposed QC-LDPC decoder architectures based on the hardware utilization efficiency (HUE). Fig. 6.1 shows the HUEs of all QC-LDPC decoders have been suggested in this thesis. This comparisons indicate that the **A1** architecture consumes the lowest HUE of 0.74 hardware-resources/layers/Mbps whereas **A2** architecture consumes the highest HUE of 16.20 hardware-resources/layers/Mbps. Similarly, data-throughput comparison of our proposed QC-LDPC decoder architectures has been presented in Fig. 6.2. It shows that **A1** decoder architecture delivers the lowest data-throughput of 965 Mbps and **A4** decoder architecture is capable of achieving the highest data-throughput of 13350 Mbps.

In this thesis, we were focusing to deliver a channel-decoder that has better hardware-efficiency as well as higher data-throughput. Hence, we merge hardware-efficiency and

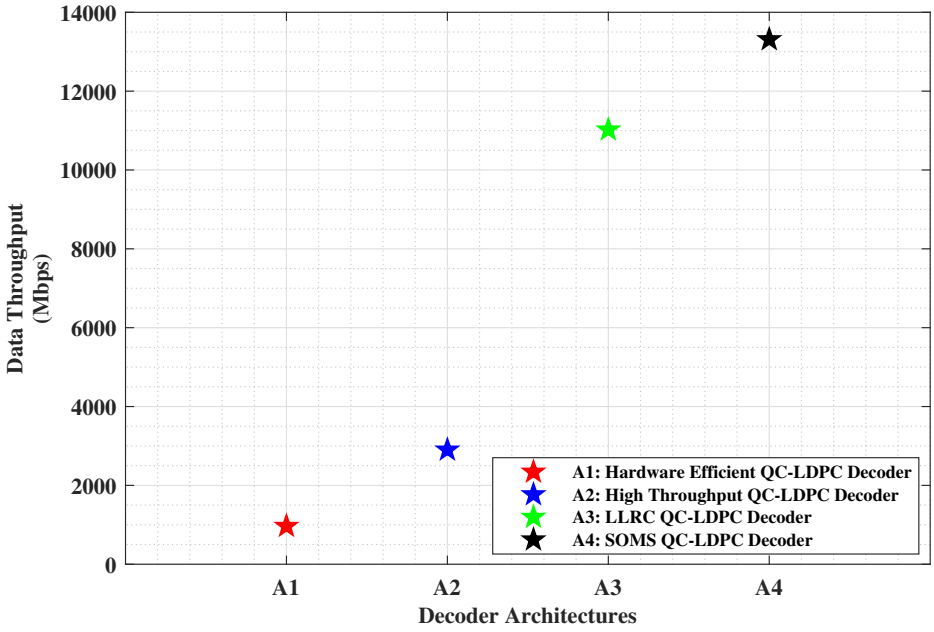


Figure 6.2: Illustrations of thesis contributions for various QC-LDPC decoder architectures based on data throughput. **A1** and **A2** architectures have been described in chapter 2 whereas **A3** and **A4** architectures are explained in chapter 3 and chapter 4, respectively.

data-throughput for various QC-LDPC decoder architectures that are achieved in this work. Fig. 6.3 represents the hardware-utilization-efficiency (HUE) and data-throughput for all the proposed QC-LDPC decoder architectures. This indicates that **A1** architecture (referring chapter 2) consumes lowest hardware-resources and delivers moderate data-throughput. Further, **A2** decoder (referring chapter 2) provides high data-throughput with the aid of more hardware-consumption. Moreover, **A3** architecture (referring chapter 3) occupies lower hardware-resources and achieves higher data-throughput. Finally, **A4** decoder architecture (referring chapter 4) provides significant improvement in HUE and delivers the highest data-throughput. So, we concluded that **A4** architecture delivers the best specifications among all the proposed QC-LDPC decoder architectures.

In this thesis, we were targeting to design an unified reconfigurable LDPC/polar channel-decoder that is hardware-efficient and delivers high data-throughput. So, we proposed an unified reconfigurable channel-decoder that has been designed based on

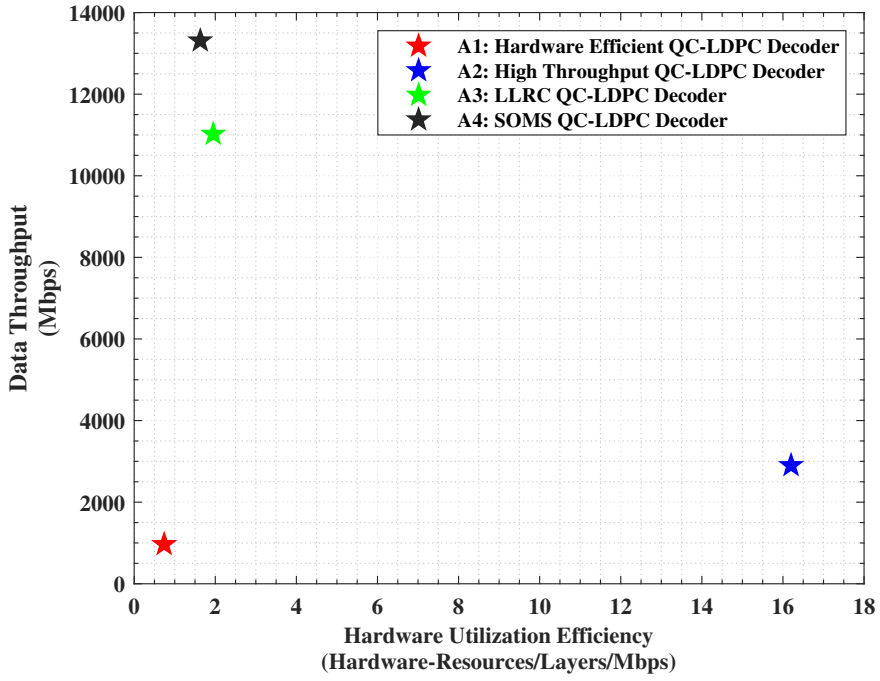


Figure 6.3: Illustrations of hardware utilization efficiency and throughput for various QC-LDPC decoder architectures that has been described in this thesis.

A4 QC-LDPC and a polar decoder architecture that is complaint to 5G-NR mMTC and URLLC specifications. Finally, we have fabricated an ASIC for this unified reconfigurable LDPC/polar channel-decoder architecture. This proposed channel-decoder is one the most hardware-efficient reconfigurable designs, reported till date in the literature. We addressed the research challenge of delivering high data-throughput and better hardware-efficiency for the contemporary wireless communication technologies in this thesis. It possesses profound importance in the efficient deployment of physical layers for current and future wireless-communication systems.

6.3 Future Scope

The channel decoding holds an immense impacts for the revolution of wireless-communication standard. Therefore, this decoding technique can be an integral part of emerging technolo-

gies such as quantum computing and quantum error correction techniques. Therefore, the quantum inspired channel decoders paving the way for ultra reliable and ultra low-latency applications for the communication in next generation networks and systems. Furthermore, the performance convergence of channel decoding reveals the exciting opportunities for the enhancement in network efficiency and scalability in edge computing and distributing networks. Future channel decoders alleviates computational as well as routing complexity in the central networking infrastructure to achieve faster response times and reliable decision making abilities. Implementation of flexible channel decoder can be integrated in various emerging technologies like machine learning and quantum computing, provide scalability and adaptability in evolving wireless-communication standard.

BIBLIOGRAPHY

[1] "Multiplexing and channel coding," *3GPP Release 15 Technical Specifications ETSI TS 138 212*, vol. 15.2.0, pp. 1–100, 2018.

[2] H. Li, B. Bai, X. Mu, J. Zhang, and H. Xu, "Algebra-assisted construction of quasi-cyclic ldpc codes for 5g new radio," *IEEE Access*, vol. 6, pp. 50 229–50 244, 2018.

[3] B.-S. Su, C.-H. Lee, and T.-D. Chiueh, "A 58.6/91.3 pj/b dual-mode belief-propagation decoder for ldpc and polar codes in the 5g communications standard," *IEEE Solid-State Circuits Letters*, vol. 5, pp. 98–101, 2022.

[4] S. Cao, T. Lin, S. Zhang, S. Xu, and C. Zhang, "A reconfigurable and pipelined architecture for standard-compatible ldpc and polar decoding," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 5431–5444, 2021.

[5] S. Lee, S. Park, B. Jang, and I.-C. Park, "Multi-mode qc-ldpc decoding architecture with novel memory access scheduling for 5g new-radio standard," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 5, pp. 2035–2048, 2022.

[6] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[7] "Digital video broadcasting (dvb); second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications; part 1: Dvb-s2," pp. 1–80, European Broadcast Union, ETSI EN 302 307-1, 2014.

[8] "Ieee standard for local and metropolitan area networks - part 16: Air interface for fixed and mobile broadband wireless access systems - amendment for physical and medium access control layers for combined fixed and mobile operation in licensed bands," *IEEE Std 802.16e-2005 and IEEE Std 802.16-2004/Cor 1-2005 (Amendment and Corrigendum to IEEE Std 802.16-2004)*, pp. 1–822, 2006.

[9] "Ieee standard for information technology–telecommunications and information exchange between systems - local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, pp. 1–4379, 2021.

[10] R. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[11] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.

[12] X.-Y. Hu, E. Eleftheriou, D.-M. Arnold, and A. Dholakia, "Efficient implementations of the sum-product algorithm for decoding ldpc codes," in *GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No.01CH37270)*, vol. 2, San Antonio (USA), 25-29 November 2001, pp. 1036–1036E vol.2.

[13] J. Chen, A. Dholakia, E. Eleftheriou, M. Fossorier, and X.-Y. Hu, "Reduced-complexity decoding of ldpc codes," *IEEE Transactions on Communications*, vol. 53, no. 8, pp. 1288–1299, 2005.

[14] J. R. Roberts, Michaelraj Kingston, "An improved low-complexity sum-product decoding algorithm for low-density parity-check codes," *Frontiers of Information Technology and Electronic Engineering*, vol. 16, 2015.

[15] H. Li, J. Guo, C. Guo, and D. Wang, "A low-complexity min-sum decoding algorithm

for ldpc codes," in *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, Chengdu (China), 27-30 October 2017, pp. 102–105.

[16] ——, "A low-complexity min-sum decoding algorithm for ldpc codes," in *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, Chengdu (China), 27-30 October 2017, pp. 102–105.

[17] S. Yun, B. Y. Kong, and Y. Lee, "Area- and energy-efficient ldpc decoder using mixed-resolution check-node processing," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 3, pp. 999–1003, 2022.

[18] H. Lopez, H.-W. Chan, K.-L. Chiu, P.-Y. Tsai, and S.-J. J. Jou, "A 75-gb/s/mm2 and energy-efficient ldpc decoder based on a reduced complexity second minimum approximation min-sum algorithm," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 4, pp. 926–939, 2020.

[19] O. Boncalo and A. Amaricai, "Ultra high throughput unrolled layered architecture for qc-ldpc decoders," in *2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, Bochum (Germany), 03-05 July 2017, pp. 225–230.

[20] O. Boncalo, A. Amaricai, and S. Nimara, "Memory-centric flooded ldpc decoder architecture using non-surjective finite alphabet iterative decoding," in *2018 21st Euromicro Conference on Digital System Design (DSD)*, Prague (Czech Republic), 29-31 August 2018, pp. 104–109.

[21] "Study on new radio (nr) access technology," *3GPP Release 14 Technical Specifications ETSI TR 138 912*, vol. 14.0.0, pp. 1–76, 2016.

[22] R. Shrestha, P. Bansal, and S. Srinivasan, "High-throughput and high-speed polar-decoder vlsi-architecture for 5g new radio," in *2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)*, New Delhi (India), 05-09 January 2019, pp. 329–334.

[23] J. Muramatsu, "Successive-cancellation decoding of binary polar codes based on symmetric parametrization," in *2021 IEEE International Symposium on Information Theory (ISIT)*, Melbourne (Australia), 12-20 July 2021, pp. 2364–2368.

[24] M. H. Ardakani, M. Hanif, M. Ardakani, and C. Tellambura, "Fast successive-cancellation-based decoders of polar codes," *IEEE Transactions on Communications*, vol. 67, no. 7, pp. 4562–4574, 2019.

[25] B. Yuan and K. K. Parhi, "Successive cancellation list polar decoder using log-likelihood ratios," in *2014 48th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove (USA), 02-05 November 2014, pp. 548–552.

[26] F. Ercan, C. Condo, and W. J. Gross, "Improved bit-flipping algorithm for successive cancellation decoding of polar codes," *IEEE Transactions on Communications*, vol. 67, no. 1, pp. 61–72, Jan 2019.

[27] Q. Jan, S. Hussain, L. Zechen, P. Zhiwen, L. Nan, and Y. Xiaohu, "Improved partitioned fast-ssc-flip decoding for polar codes," in *2022 7th International Conference on Computer and Communication Systems (ICCCS)*, Wuhan (China), 22-25 April 2022, pp. 382–386.

[28] M. Moradi and A. Mozammel, "A tree pruning technique for decoding complexity reduction of polar codes and pac codes," *IEEE Transactions on Communications*, vol. 71, no. 5, pp. 2576–2586, 2023.

[29] O. Boncalo and A. Amaricai, "Ultra high throughput unrolled layered architecture for qc-ldpc decoders," in *2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, Bochum (Germany), 03-05 July 2017, pp. 225–230.

[30] N. Yang, S. Jing, A. Yu, X. Liang, Z. Zhang, X. You, and C. Zhang, "Reconfigurable decoder for ldpc and polar codes," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, Florence (Italy), 27-30 May 2018, pp. 1–5.

[31] T. Bhatt, V. Sundaramurthy, V. Stolpman, and D. McCain, "Pipelined block-serial decoder architecture for structured ldpc codes," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 4, Toulouse (France), 14-19 May 2006, pp. IV–IV.

[32] M. Milicevic and P. G. Gulak, "A multi-gb/s frame-interleaved ldpc decoder with path-unrolled message passing in 28-nm cmos," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 10, pp. 1908–1921, 2018.

[33] Q. Lu, Z. Shen, C.-W. Sham, and F. C. M. Lau, "A parallel-routing network for reliability inferences of single-parity-check decoder," in *2015 International Conference on Advanced Technologies for Communications (ATC)*, Ho Chi Minh City (Vietnam), 14-16 October 2015, pp. 127–132.

[34] C.-W. Sham, X. Chen, W. M. Tam, Y. Zhao, and F. C. Lau, "A layered qc-ldpc decoder architecture for high speed communication system," in *2012 IEEE Asia Pacific Conference on Circuits and Systems*, Kaohsiung (Taiwan), 02-05 December 2012, pp. 475–478.

[35] M. Mansour, "A turbo-decoding message-passing algorithm for sparse parity-check matrix codes," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4376–4392, 2006.

[36] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, 2001.

[37] M. Zhao, X. Zhang, L. Zhao, and C. Lee, "Design of a high-throughput qc-ldpc decoder with tdmp scheduling," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 1, pp. 56–60, 2015.

[38] K. Zhang, X. Huang, and Z. Wang, "A high-throughput ldpc decoder architecture with rate compatibility," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 4, pp. 839–847, 2011.

[39] A. Darabiha, A. Chan Carusone, and F. R. Kschischang, "Power reduction techniques for ldpc decoders," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 8, pp. 1835–1845, 2008.

[40] H. Cui, F. Ghaffari, K. Le, D. Declercq, J. Lin, and Z. Wang, "Design of high-performance and area-efficient decoder for 5g ldpc codes," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 2, pp. 879–891, 2021.

[41] A. Verma and R. Shrestha, "A new vlsi architecture of next-generation qc-ldpc decoder for 5g new-radio wireless-communication standard," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, Seville (Spain), 12-14 October 2020, pp. 1–5.

[42] S. A. Hashemi, M. Mondelli, S. H. Hassani, C. Condo, R. L. Urbanke, and W. J. Gross, "Decoder partitioning: Towards practical list decoding of polar codes," *IEEE Transactions on Communications*, vol. 66, no. 9, pp. 3749–3759, 2018.

[43] M. Mousavi, Y. Fan, C.-Y. Tsui, J. Jin, B. Li, and H. Shen, "Efficient partial-sum network architectures for list successive-cancellation decoding of polar codes," *IEEE Transactions on Signal Processing*, vol. 66, no. 14, pp. 3848–3858, 2018.

[44] S. A. Hashemi, C. Condo, M. Mondelli, and W. J. Gross, "Rate-flexible fast polar decoders," *IEEE Transactions on Signal Processing*, vol. 67, no. 22, pp. 5689–5701, 2019.

[45] S. Yun, B. Y. Kong, and Y. Lee, "Area- and energy-efficient ldpc decoder using mixed-resolution check-node processing," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 3, pp. 999–1003, 2022.

[46] V. L. Petrović, M. M. Marković, D. M. E. Mezeni, L. V. Saranovac, and A. Radošević, "Flexible high throughput qc-ldpc decoder with perfect pipeline conflicts resolution and efficient hardware utilization," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 12, pp. 5454–5467, 2020.

[47] Y. Tao, S.-G. Cho, and Z. Zhang, "A configurable successive-cancellation list polar decoder using split-tree architecture," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 2, pp. 612–623, 2021.

[48] T. Lin, S. Cao, S. Zhang, S. Xu, and C. Zhang, "A reconfigurable decoder for standard-compatible ldpc codes and polar codes," in *2019 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, Bangkok (Thailand), 11-14 November 2019, pp. 73–76.

[49] M. K. Roberts and R. Jayabalan, "An improved low-complexity sum-product decoding algorithm for low-density parity-check codes," *Frontiers of Information Technology & Electronic Engineering*, vol. 16, pp. 511–518, 2015.

[50] S. Yun, B. Y. Kong, and Y. Lee, "Area- and energy-efficient ldpc decoder using mixed-resolution check-node processing," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 3, pp. 999–1003, 2022.

[51] A. A. Emran and M. Elsabrouty, "Simplified variable-scaled min sum ldpc decoder for irregular ldpc codes," in *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*, Las Vegas (USA), 10-13 January 2014, pp. 518–523.

[52] J.-C. Liu, H.-C. Wang, C.-A. Shen, and J.-W. Lee, "Low-complexity ldpc decoder for 5g urllc," in *2018 IEEE Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia)*, Chengdu (China), 26-30 October 2018, pp. 43–46.

[53] C.-L. Wey, M.-D. Shieh, and S.-Y. Lin, "Algorithms of finding the first two minimum values and their hardware implementation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 11, pp. 3430–3437, 2008.

[54] C.-W. Sham, X. Chen, F. C. M. Lau, Y. Zhao, and W. M. Tam, "A 2.0 gb/s throughput decoder for qc-ldpc convolutional codes," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 7, pp. 1857–1869, 2013.

[55] J. Su and J. Han, "Design of energy efficient ldpc decoders with low-voltage strategy," in *2015 IEEE 11th International Conference on ASIC (ASICON)*, Chengdu (China), 03-06 November 2015, pp. 1–4.

[56] T. T. Nguyen-Ly, V. Savin, K. Le, D. Declercq, F. Ghaffari, and O. Boncalo, "Analysis and design of cost-effective, high-throughput ldpc decoders," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 3, pp. 508–521, 2018.

[57] S. Jiang, F. Mo, F. C. M. Lau, and C.-W. Sham, "Tree-permutation-matrix based ldpc codes," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 8, pp. 1019–1023, 2018.

[58] "Chairman's notes of ai 7.1.5 on consideration on ldpc design for nr,doc. r1-1611112, 3gpp," http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_87/Docs/R1-1611112.zip, nov. 2016.

[59] P. Kang, Y. Xie, L. Yang, and J. Yuan, "Enhanced quasi-maximum likelihood decoding based on 2d modified min-sum algorithm for 5g ldpc codes," *IEEE Transactions on Communications*, vol. 68, no. 11, pp. 6669–6682, 2020.

[60] T. Richardson, "Adjusted min-sum decoder," U.S. Patent 0 109 269 A1, April 2018. [Online]. Available: https://www.freepatentsonline.com/y2018/0109269.html"

[61] Y. Chen and K. Parhi, "Overlapped message passing for quasi-cyclic low-density parity check codes," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 6, pp. 1106–1113, 2004.

[62] J. Nadal and A. Baghdadi, "Parallel and flexible 5g ldpc decoder architecture targeting fpga," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 6, pp. 1141–1151, 2021.

[63] "Xilinx logicore ip: Ldpc encoder/decoder v2.0," https://www.xilinx.com/member/ldpc-enc-dec/ldpc.html#kintexuplus, feb. 2021.

[64] Y. Liu, W. Tang, and D. G. M. Mitchell, "Efficient implementation of a threshold modified min-sum algorithm for ldpc decoders," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 9, pp. 1599–1603, 2020.

[65] O. Boncalo, G. Kolumban-Antal, A. Amaricai, V. Savin, and D. Declercq, "Layered ldpc decoders with efficient memory access scheduling and mapping and built-in support for pipeline hazards mitigation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 4, pp. 1643–1656, 2019.

List of Publications

Refereed Journals

1. **Anuj Verma** and Rahul Shrestha, "High-Throughput and Hardware-Efficient ASIC-Chip Fabrication of Reconfigurable LDPC/Polar Decoder for mMTC and URLLC 5G-NR Applications," ***in IEEE Transactions on Circuits and Systems I: Regular Papers***, DOI: 10.1109/TCSI.2024.3429174, Volume: 71, Issue: 9, pp. 4284-4297, September-2024 (**Click Here**).

2. **Anuj Verma** and Rahul Shrestha,"Low Computational-Complexity SOMS-Algorithm and High-Throughput Decoder Architecture for QC-LDPC Codes," ***in IEEE Transactions on Vehicular Technology,*** DOI: 10.1109/TVT.2022.3203802, Volume: 72, Issue: 1, pp. 66-80, January-2023 (**Click Here**).

3. **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,*** DOI: 10.1109/TCSII.2021.3071804, Volume: 68, Issue: 8, pp. 2835-2839, August-2021 (**Click Here**).

Peer-Reviewed Conferences

1. **Anuj Verma** and Rahul Shrestha, "A New VLSI Architecture of Next-Generation QC-LDPC Decoder for 5G New-Radio Wireless-Communication Standard," ***2020 IEEE International Symposium on Circuits and Systems (ISCAS),*** DOI: 10.1109/ISCAS45731.2020.9181188, pp. 1-5, October-2020, Spain (Seville) (**Click Here**).

2. **Anuj Verma** and Rahul Shrestha, "A New Partially-Parallel VLSI-Architecture of Quasi-Cyclic LDPC Decoder for 5G New-Radio," ***2020 33rd International Conference on VLSI Design and 2020 19th International Conference on Embedded Systems (VLSID)***, DOI: 10.1109/VLSID49098.2020.00018, pp. 1-6, January-2020, India (Bangalore) (**Click Here**).

Anuj VERMA

Ph.D. Research Scholar (Google Scholar)

linkedin.com/in/anuj-verma-416715123 anujsusniwas@gmail.com

+91 9958-612862 Bengaluru, Karnataka Indian Citizen

Passionate and accomplished PhD in VLSI, with a strong foundation in Digital VLSI architecture and hands-on experience in cutting-edge ASIC (RTL-to-GDS) designing. Eager to contribute to cutting-edge research and development, cutting-edge projects, and inspire the next generation of engineers in the evolving landscape of VLSI technology. Be Happy and Make Others Happy.

ACADEMIC QUALIFICATIONS

February 2018 – August 2024 | Doctorate of Philosophy, SCHOOL OF COMPUTING AND ELECTRICAL ENGINEERING, IIT Mandi

- **Efficient VLSI-Architectures and ASIC-Fabrication of Channel Decoder for Contemporary Wireless-Communication Systems:-** This thesis exploits various hardware-efficienct and high-throughput LDPC decoder archietctures for 5G-NR networks. Various hardware-friendly LDPC decoding algorithms has been proposed that reduces the routing and computational complexity. Finally, a reconfigurable channel (LDPC/polar) decoder has been **ASIC fabricated** that are compliant to mMTC and URLLC applications of the 5G-NR wireless-communication standard.
- **Final Grade** : 8.69 out of 10.
- Supervisor : **Dr. Rahul Shrestha**.

Digital Architectures | ASIC and FPGA Methodologies | RTL Coding Techniques | Timing Constraints | Physical Designing | Algorithms-to-Architectures Designing | 5G New Radio

July 2015 – June 2017 | Master of Technology | VLSI Design, ECE DEPARTMENT, DCRUST Murthal

- Design methodology of analog and digital circuits designing in 180nm CMOS Technology. FPGA implementation of peripheral component interconnect express (PCIe) interface using verilog HDL.
- **Final Grade** : 8.24 out of 10.
- Supervisor : **Dr. Pawan Kumar Dahiya**.

Physical Layer | Point-to-Point Communication | PCIe | PCI | PCI-X | AGP

July 2009 – June 2013 | Bachelor of Technology | ECE Department, IPEC, UPTU Lucknow

- Fundamental aspects of electronics devices, circuits and systems. Study of various modulation techniques & MATLAB implementation of various modulation schemes for communication applications.
- **Final Percentage** : 66.13.
- Supervisor : **Dr. Priyanka**.

FSO communication | MATLAB | ASK | FSK | BPSK | QPSK

July 2008 – June 2009 | Senior Secondary Education, S.D. INTER COLLEGE, UP Board

- **Final Percentage** : 69.00.

Mathematics | Physics | Chemistry

July 2006 – June 2007 | Secondary Education, S.D. INTER COLLEGE, UP Board

- **Final Percentage** : 64.00.

Mathematics | English

PROFESSIONAL EXPERIENCES

July 2024 – Present | Chief Engineer, SAMSUNG RESEARCH & DEVELOPMENT INSTITUTE, Bengaluru, India

- The project aims to leverage the flexibility and extensibility of RISC-V to create a highly efficient, scalable, and low-power solution for various image processing tasks. By optimizing the instruction set and hardware accelerators, the architecture is designed to handle complex image processing algorithms with improved performance, power, area (PPA), with reduced latency.

RISC V | Custom Instructions | Image Processing | Contrast Limited Adaptive Histogram Equalization (CLAHE)

March 2023 – September 2023 | Ph.D. Research Intern, SAMSUNG RESEARCH INSTITUTE, Bengaluru, India

- I designed a framework for architectural-level area and energy estimation tailored for custom hardware accelerators and digital architectures, involves integrating RTL and analytical modeling across various technology nodes. The goal is to develop a comprehensive area/energy estimator framework that optimizes both the design and performance of these digital architectures.

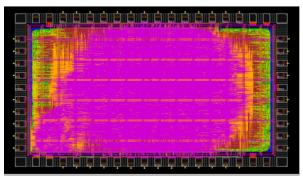
Area/Energy Estimators | RTL modelling | Analytical Modelling

July 2017 – January 2018 | **Senior Associate, Genpact Headstrong Capital Markets, Noida, India**

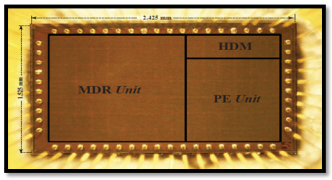
- I worked on configuring computer software or robots with Robotic Process Automation (RPA) to replicate human actions. These RPA robots interact with the user interface to capture data, manipulate applications, and perform tasks just as humans would.

Automation Anywhere | Openspan | Blue Prism

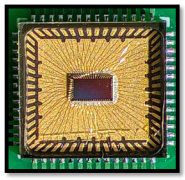
Ph.D. Research Work



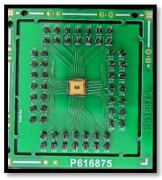
(a)



(b)



(c)



(d)

Our research work is mainly focused on the **reconfigurable low-density parity-check (LDPC) and polar channel decoder** with the exact specifications for **fifth-generation New Radio (5G NR) wireless communication standand**. We have implemented a **hardware-efficient VLSI architecture** (Click here) and **high-throughput digital architecture** (Click here) for LDPC decoder that is complaint to 5G NR standard and these designs are place and routed on the field-programmable logic array (FPGA) platform. Futher, We proposed hardware-efficient and high-throughput **log-likelihood-ratio compound (LLRC) segregation technique based algorithm and digital architecture** (Click here) of LDPC decoder. We also suggested a **simplified offset-minsum (SOMS) LDPC decoding algorithm** that alleviate the lower computational-complexity and its corresponding high-throughtput LDPC decoder architecture is also presented. The SOMS based LDPC decoder has been **FPGA prototyped and ASIC synthesized** (Click here) in the **90nm technology**. Finally, We **fabricated an ASIC for reconfigurable LDPC and polar channel decoder architecture** (Click here) that is complaint to **mMTC and URLLC application** of 5G-NR wireless communication standard. The **ASIC fabricated chip layout design, die-micrograph, QFN-packaged die, and chip printed circuit board (PCB) of reconfigurable channel decoder** are represented in **Figure (a)-(d), respectively.**

Ph.D. Thesis Title	**Efficient VLSI-Architectures and ASIC-Fabrication of Channel Decoder for Contemporary Wireless-Communication Systems**
Supervisor	**Dr. Rahul Shrestha**, Associate Professor, School of Computing & Electrical Engineering (SCEE), **IIT Mandi**, (Website)

Publications

July 2024	**Anuj Verma**, Rahul Shrestha, "High-Throughput and Hardware-Efficient ASIC-Chip Fabrication of Reconfigurable LDPC/Polar Decoder for mMTC and URLLC 5G-NR Applications", **IEEE Transactions on Circuits and Systems I : Regular Papers**, Jul. 2024 (**Early Access**). (Click here)
January 2023	**Anuj Verma**, Rahul Shrestha, "Low Computational-Complexity SOMS-Algorithm and High-Throughput Decoder Architecture for QC-LDPC Codes", **IEEE Transactions on Vehicular Technology**, vol. 72, no. 1, pp. 66-80, Jan. 2023. (Click here)
August 2021	**Anuj Verma**, Rahul Shrestha, "Hardware-Efficient and High-Throughput LLRC Segregation Based Binary QC-LDPC Decoding Algorithm and Architecture", **IEEE Transactions on Circuits and Systems II : Express Briefs**, vol. 68, no. 8, pp. 2835-2839, Aug. 2021. (Click here)
May 2020	**Anuj Verma**, Rahul Shrestha, "A New VLSI Architecture of Next-Generation QC-LDPC Decoder for 5G New-Radio Wireless-Communication Standard", **IEEE International Symposium on Circuits and Systems (ISCAS)**, pp. 1-5, May-2020.(Click here)
January 2020	**Anuj Verma**, Rahul Shrestha, "A New Partially-Parallel VLSI-Architecture of Quasi-Cyclic LDPC Decoder for 5G New-Radio", **33rd IEEE International Conference on VLSI Design and 19th International Conference on Embedded Systems (VLSID)**, pp. 1-6, January-2020.(Click here)
August 2017	**Anuj Verma**, Pawan Kumar Dahiya, "PCIe BUS : A State-of-the-Art-Review", **IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)**, Volume 7, Issue 4, PP 24-28, August-2017. (Click here)
March 2017	Kapil Kumar Sain, **Anuj Verma**, Priyanka, "Design and Implementation of Low Power D Flip-Flop in Cadence Virtuoso Tool", **2nd National Conference MANTHAN-2016 on Recent Trends in Computing and Communication Technologies (RCCT)**, March-2017.
September 2016	**Anuj Verma**, Pawan Kumar Dahiya, "Addressing the Key Dispersion Algorithm and Security in RFID Tag-Reader MAP", **1st National Conference MANTHAN-2016 on Recent Trends in Computing and Communication Technologies (RCCT)**, September-2016.

Technical Skills

VLSI Design Tools	Design Compiler and **Genus** for RTL Synthesis, VCS and **NCLaunch** for Functional Verification, **Innovus** for Physical Design flow, **Virtuoso schematic editor**, **Vivado (2018.2)** and ISE Design Suite for FPGA Based System Design, **ModelSim**, Symica, Magic VLSI Layout, Qflow.
Subjects	**Digital Architectural Designing**, **Digital MOS LSI Circuits**, **CMOS Designing**, **Digital Communication**, Digital & Analog Electronics, Electronics Devices and Circuits.
Hardware Language	**Verilog**, **System Verilog**, **SystemC**, VHDL.
Programming Language	C, Microsoft .Net, My SQL.
System Simulation Tool	MATLAB
Documentation Tool	Latex

Academic Teaching Assitantship

Feb'18-Aug'22 **Teaching Assistant**, School of Computing and Electrical Engineering, **Indian Institute of Technology Mandi**.

- **Digital VLSI Architecture Design** course and laboratory sessions for **M.Tech and Ph.D.** students.
- **Digital MOS LSI Design** course and laboratory sessions for **M.Tech and Ph.D.** students.
- **Digital System Design** and **Digial IC Design** laboratory sessions for **B.Tech** students.
- **Computer Architecture and Organisation** course and laboratory sessions for **B.Tech** students.
- **Applied Electronics** laboratory sessions for **B.Tech** students

Aug'15-May'17 **Teaching Assistant**, ECE Department, **Deenbandhu Chhotu Ram University of Science and Technology (DCRUST), Murthal, Sonipat**.

- **Digital CMOS IC Design** course and laboratory sessions for **M.Tech** students.
- **Verilog based Digital System Design** course and laboratory sessions for **M.Tech** students.
- **Digital Electronics** laboratory sessions for **B.Tech** students.

Membership of Professional Bodies

Membership	**Graduate Student Member**, Institute of Electrical and Electronics Engineers (IEEE).
Membership	Institute of Electrical and Electronics Engineers (IEEE) **circuits and systems (CAS) society**.
Membership	Institute of Electrical and Electronics Engineers (IEEE) **solid-state circuits society (SSCS)**.
IEEE Reviewer	**Transactions on Circuits and Systems I : Regular Papers**.
IEEE Reviewer	**Transactions on Circuits and Systems II : Express Briefs**.
IEEE Reviewer	**International Symposium on Circuits and Systems (ISCAS) conference** organized by **IEEE circuits and systems (CAS) society**.
IEEE Reviewer	**International MidWest Symposium on Circuits and Systems (MWSCAS) conference** organized by **IEEE circuits and systems (CAS) society**.
IEEE Reviewer	**Asia Pacific Conference on Circuits and Systems (APCCAS) conference** organized by **IEEE circuits and systems (CAS) society**.

Certifications

January 2023	International Conference on **36th IEEE International Conference on VLSI Design and 22nd International Conference on Embedded Systems (VLSID)** organized by AMD Semiconductor and VLSI seciety of India.
December 2022	International Workshop on **5G and Beyond : What is next ?** certified by Department of Electrical Engineering, IIT Kanpur.
October 2020	International Conference on **IEEE International Symposium on Circuits and Systems (ISCAS)** organized by IEEE circuits and systems (CAS) society.
January 2020	International Conference on **33rd IEEE International Conference on VLSI Design and 19th International Conference on Embedded Systems (VLSID)** organized by Qualcomm Semiconductor company.
January 2019	International Conference on **32nd IEEE International Conference on VLSI Design and 18th International Conference on Embedded Systems (VLSID)** organized by NXP Semiconductor company.
December 2018	International Workshop on **A Hands-on Introduction to Modern Wireless Systems : Theory and Simulation** certified by Department of Electrical Engineering, IIT Kanpur.
November 2018	Certification of Excellence for **International Workshop on Nano/Micro 2D-3D Fabrication, Manufacturing of Electronic-Biomedical Devices & Applications (IWNEBD)** certified by C4DFED, IIT Mandi.
July 2016	Summer Training in **VLSI design using VERILOG and FPGA** certified by DKOP Labs Private Limited.
July 2012	Industrial Training on **Communication Techniques** certified by Bharat Sanchar Nigam Limited (BSNL).
April 2012	Workshop on **Cyborg-The Robotics Workshop** conducted by KYRION ROBOTICS CLUB, IIT Kharagpur.

Honors and Awards

January 2023	**Fellowship Awarded** by AMD semiconductor and VLSI society of India at **36th IEEE International Conference on VLSI Design and 22nd International Conference on Embedded Systems (VLSID'2023)**.
March 2017	**Graduate Aptitude Test Examination (GATE)**, Qualified and cleared interview at Indian Institute of Technology (IIT) Mandi for admission in Ph.D program.
March 2014	**Graduate Aptitude Test Examination (GATE)**, Qualified and got addmission at Deen Bandhu ChhottuRam University of Science and Technology (DCRUST) Murthal in M.Tech program.
January 2013	Awarded as **Organizer** in ECE and EN forum : "Gentronix" at InderPrastha Engineering College (IPEC) affilated by UPTU.
January 2012	Awarded as **Promoter Team Leader** in ECE and EN forum:"Gentronix" at Inderprastha Engineering College (IPEC) affilated by UPTU.

Outreach and Volunteering

November 2018	International Workshop on Nano/Micro 2D-3D Fabrication, Manufacturing of Electronic-Biomedical Devices and Applications organized by **Centre for Design & Fabrication of Electronics Devices(C4DFED)** at Indian Institute of Technology (IIT) Mandi.
April 2012	Workshop on Cyborg-The Robotics Workshop conducted by **Kyrion Robotics Club (IIT Kharagpur)** at InderPrastha Engineering College (IPEC).

References

Dr. Rahul Shrestha
Associate Professor, IIT Mandi
@ rahul_shrestha@iitmandi.ac.in
+91 1905-267271

Dr. Hitesh Shrimali
Associate Professor, IIT Mandi
@ hitesh@iitmandi.ac.in
+91 1905-267902

Dr. Shubhajit Roy Chowdhury
Associate Professor, IIT Mandi
@ src@iitmandi.ac.in
+91 1905-267915