A New VLSI Architecture of Next-Generation QC-LDPC Decoder for 5G New-Radio Wireless-Communication Standard

Anuj Verma† and Rahul Shrestha‡, *Member*, *IEEE*
School of Computing and Electrical Engineering
Indian Institute of Technology Mandi
d17029@students.iitmandi.ac.in†; rahul_shrestha@iitmandi.ac.in‡

***Abstract*—In this paper, we present a new microarchitecture of low-density parity-check (LDPC) decoder compliant to the specifications of 5G new-radio (NR) wireless-communication standard. This work suggests a fully-parallel VLSI architecture for this decoder to achieve high throughput. The digital architecture of internal modules as well as system-level design of the LDPC decoder are presented here. The comprehensive bit-error-rate (BER) performance analyses of our LDPC decoder has been performed in additive-white Gaussian-noise (AWGN) channel environment for various number of decoding iterations and bit-quantization. It delivers a BER of 10^{-6} at 1 dB of E_b/N_0 while decoding for 10 iterations with 7-bits quantization. In addition, FPGA implementation and post-route simulation of the proposed LDPC decoder are carried out that can decode an encoded LDPC code of 26112 code-length for 1/3 code-rate. Our decoder has achieved a throughput of 2.9 Gbps while operating at a clock frequency of 102 MHz. These implementation results are compared with the reported works where our design delivered 20× better throughput compared to the state-of-the-art LDPC decoders.**

***Index Terms*—Wireless communication, 5G new-radio standard, channel coding, LDPC codes, very-large scale-integration (VLSI) architectures, and field-programmable gate-array (FPGA).**

I. Introduction

Low-density parity-check (LDPC) code coined by Robert G. Gallager, is one of the Shannon-limit approaching codes apart from turbo and polar codes [1]. The key features of LDPC codes are higher coding gain, lower error floor and simple decoder architecture that makes it the most widely adopted as forward-error-correction (FEC) code by various wireless standards [2]. Such near-Shannon limit performing [3] attributes proliferated the use of LDPC codes in contemporary wireless standards like DVB-S2, IEEE 802.11n (Wi-Fi) and IEEE 802.16e (WiMAX) [4] [5] [6]. Recently, LDPC code has been standardized for wireless enhanced-mobile broad-band (eMBB) communication in 5G new radio (NR) by the 3^{rd}-generation partnership-project (3GPP) due to its multi-edge and flexible shift-coefficient decoder design which can achieve area-efficient and higher-throughput features [7]. Such LDPC codes can be graphically represented by bipartite Tanner graph where the internal message propagations are achieved between the check nodes (CNs) and variable nodes (VNs) that provide the lower entanglement from the hardware perspective [8]. The standardized LDPC codes for 5G NR has adopted two base-graph (BG) matrices: BG1 and BG2 [9] [10] [11]. The BG1 matrix structure has been shown in Fig. 1 which has 46 rows and 68 columns where the maximum number of information-bits are 8448 and supports the code rate from 1/3 to 8/9. On the other side, BG2 matrix has 42 rows and 52 columns with 3840 maximum number of information-bits and supports the code rate from 1/2 to 2/3 [9] [11]. Efficient LDPC decoder implementation is significant task while designing the physical layer for any wireless standard. It possess various research challenges at the early stage of wireless standard deployment. Therefore, it is high-time to conceive efficient LDPC decoder-architecture that is compliant to next generation of 5G standard. As shown in Fig. 1, the major challenge is to design decoder architecture to support huge BG1 matrix that generates the decoded bits of 26112 encoded information-bits. To the best of our knowledge, LDPC decoder-implementations compliant to the specifications of 5G-NR has not been reported in literatures. Hence, our work intends to bridge this gap and present systematic design of efficient LDPC decoder for the next generation of wireless communication systems.

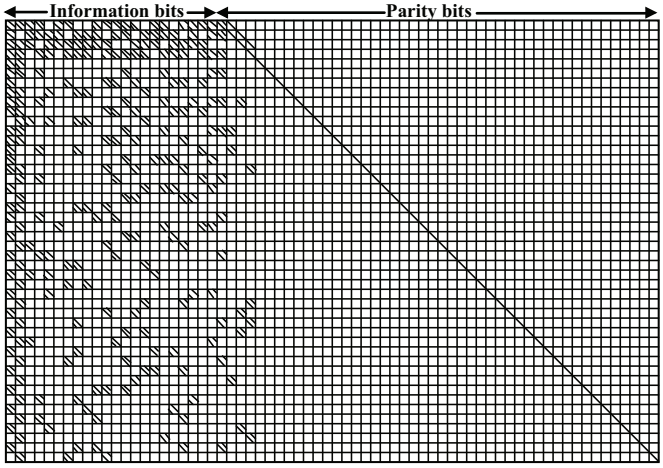

Fig. 1. Schematic representation of H_b matrix (BG1) compliant to 5G-NR wireless-communication standard for generating a QC-LDPC code of code-length (n) = 26112 bits for a code-rate of 1/3 with z = 384.

The highlights of our research contributions are as follows. We propose fully-parallel microarchitecture of Quasi-Cyclic (QC) LDPC decoder for eMBB specifications in 5G-NR wireless standard. Our decoder has been designed based on offset min-sum decoding algorithm with layered scheduling for the BG1 matrix which supports the code-rate of 1/3. Overall as well as internal architectures of the suggested LDPC decoder has been presented here. In addition, extensive performance analyses of our decoding algorithm has been carried out based on 5G-NR specifications. Furthermore, hardware implementation of our LDPC decoder is performed on FPGA platform and its results are compared with the state-of-the-art works.

Organization of this paper is as follows. Mathematical background of the LDPC decoding algorithm has been presented in section II. Section III includes the proposed architectures and the details of 5G-NR specifications. Performance analyses and FPGA implementations are illustrated in section IV. Finally, our paper concludes in section V.

II. Preliminaries

The QC-LDPC codes are (n, k) linear block codes which are encoded using $(n-k)\times k$ sparse parity-check matrix H, where k

978-1-7281-3320-1/20/$31.00 ©2020 IEEE

is information bit length and n is code-length ($n > k$). Fig. 1 shows the base-graph matrix (H_b) structure for the code rate 1/3 which is standardized for 5G-NR wireless-communication standard. The H is constructed by replacing filled cells with $z \times z$ circular permutation identity matrix and blank cells with $z \times z$ zero matrix in H_b matrix.

Algorithm 1 Offset Min-Sum Layered Decoding Algorithm for the Proposed QC-LDPC Decoder-Architecture.

Inputs: L, σ^2; **Output:** $\hat{C}$
Initialization:
for `all n ∈ V` **do**
  $z_n = log\left(\frac{P(L_n|C_n=0)}{P(L_n|C_n=1)}\right) = \frac{2.L_n}{\sigma^2}$
  $y_{m,n}^{(i=0)} = 0$
  $x_{p,n}^{(i=0)} = log\left(\frac{P(L_n|C_n=0)}{P(L_n|C_n=1)}\right), p \in H(n)$
end for
CN and VN Message Exchange Iteratively:
for `i = 0,` Itr_{max} **do**
  for `l =` n_l **do**
    for `m ∈ B(l)` **do**
      for `n ∈ H(m)` **do**
        $x_{m,n} = z_n - y_{m,n}$
        $\alpha_{m,n} = \min_{n' \epsilon H(m) \setminus n} |x_{m,n'}| - offset$
        $\beta_{m,n} = \begin{cases} \alpha_{m,n} & \alpha_{m,n} \geq 0 \\ 0 & otherwise \end{cases}$
        $\tilde{y}_{m,n} = \prod_{n' \epsilon H(m) \setminus n} sign(x_{m,n'}) \cdot \beta_{m,n}$
        $\tilde{z} = x_{m,n} + \tilde{y}_{m,n}$ ▷ Updated belief
      end for
    end for
  end for
Hard Decision for all n ∈ V:
  $\hat{c_n}^{(i)} = \begin{cases} 0 & z_n^{(i)} \geq 0 \\ 1 & otherwise \end{cases}$
  if $\hat{C} \cdot H^T = 0$ **then** `break;`
end for

The encoded n bits are transmitted over noisy channel and z_n represents the received soft-demodulated logarithmic-likelihood-ratios (LLRs), corresponding to codeword bits, that is applied as input to our LDPC decoder. Our LDPC decoder architecture is based on the offset min-sum algorithm with layered scheduling which generates a estimated codeword $\hat{C}$ after every iteration and its details has been illustrated in Algorithm 1. It can be observed that $\tilde{y}_{m,n}$ is message from CN to all connected VNs and $\tilde{z}$ represents message from VN to all the connected CNs which are updated after every H_b layer process.

III. Proposed LDPC Decoder-Architecture

Here, we present the new microarchitecture of QC-LPDC decoder −compliant to 3GPP 5G-NR wireless standard− which performs the offset min-sum decoding based on layer scheduling algorithm.

A. Overall Decoder Architecture

A top-level design of the proposed LDPC decoder is shown in Fig. 2 where the input is logarithmic likelihood ratios (LLRs) of 7-bits quantization that is generated by soft-demodulator at the receiver side. At first, input LLRs are buffered in the initial-register memory-bank (IRMB) whose specific configuration is presented later in 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. All these 68 outputs are fed to memory-selection multiplexer-network (MSMN) in a single clock cycle. MSMN is a stack of 19 parallel 68:1 multiplexers (MUXes) which selects 19 LLR-combinations (19-LLRCs) based on the columns (where non -1's elements exist) of selected H_b layer. For example, layer-1 to layer-4 of H_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, three initial MUXes of MSMN select LLRCs corresponding to 1^{st}, 2^{nd} & 27^{th} columns and remaining 16 MUXes pass the maximum values (7-bits LLR). Consecutively, these 19-LLRCs is rotated by the hard-wired permutation-network (HWPN) based on their corresponding H_b layer column element. The detailed operation of HWPN is schematically illustrated in Fig. 2. Furthermore, it shows that the 2688-bits 19-LLRCs are converted into 133-bits 384-LLRCs with the aid of splitter unit (SPLU). 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 LLRCs and A represents a 7-bits LLR. These 133-bits 384 LLRCs and the output data fetched from intermediate memory (IntM) are applied to 384-replicated combined check-node & variable-node processing-units (CVCPUs) through the input registers of first-pipeline stage that enforces the critical path to fall in CVCPU. Each of these 384 CVCPUs produces 133-bits updated LLRs and 32-bits compressed-extrinsic LLR (CE-LLR) by performing check-node and variable node processes. Therefore, 384-parallel CVCPUs send 133-bits updated 384-LLRCs to combiner unit (CMBU) for next H_b layer and the concatenation of 32-bits 384 CE-LLRs to IntM for next iteration through the second-pipeline stage as shown in Fig. 2. Hence, these pipelining stages make the CVCPU unit into the critical path.

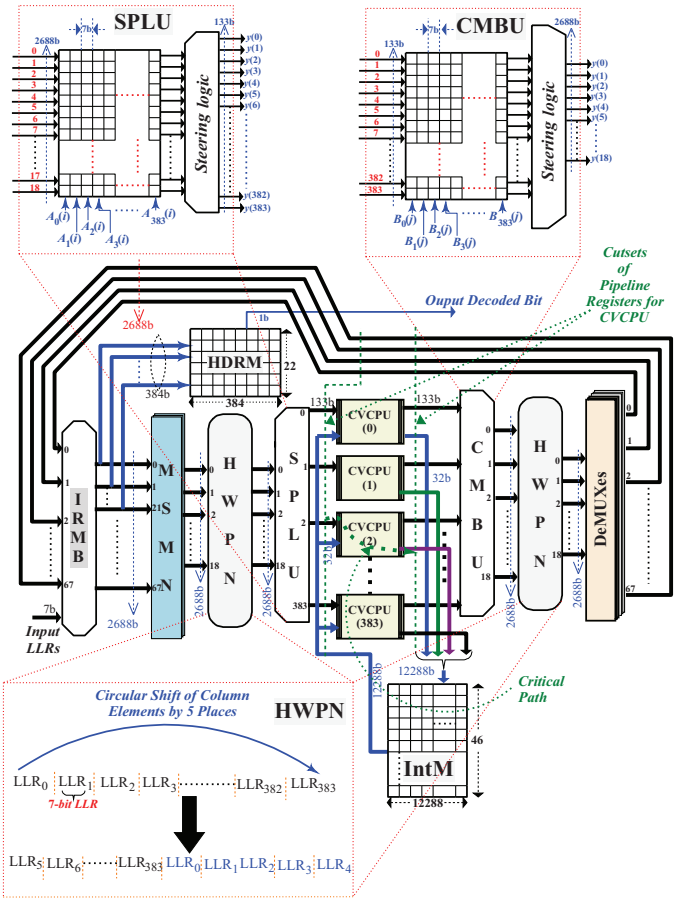

Fig. 2. Proposed VLSI architecture of QC-LDPC decoder compliant to 5G-NR wireless-communication standard supporting a frame length of 26112 bits which is encoded using BG1 matrix.

The CMBU performs reverse operation of SPLU that transforms 133-bits updated 384-LLRCs into 2688-bits updated 19-LLRCs, as shown in Fig. 2. 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 LLRCs and B is the updated LLR of 7-bits. Subsequently, these updated 19-LLRCs from CMBU are again circularly shifted (via HWPN shown in Fig. 2) by the factor of (z-η_H) where z represents expansion factor (i.e. 384 in our design) and η_H is corresponding H_b layers column-elements. The 2688-bits 19-LLRCs are fed back to IRMB through the network of de-multiplexers (De-MUXes), as shown in Fig. 2. In addition, few specific register-memory-units (RMUs) of IRMB (where non -1's column elements are exist in corresponding H_b layer) are revised with the 2688-bits updated LLRCs in a single clock-cycle. Hence, 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 next H_b layer and iteration respectively. Therefore, these LLRs are again read from the IRMB in a clock cycle for next H_b layer, processes for LLR updation and send back to IRMB in next clock cycle. Aforementioned LLR updating process continues till 46^{th} layer of H_b matrix that marks the completion of first decoding-iteration and it consumes 2×46 = 92 clock cycles. Eventually at the end of first iteration, IntM of our decoder is filled with H_b layer CE-LLRs.

In the next decoding iteration, IntM reads out the CE-LLRs to 384-replicated CVCPUs corresponding to new H_b layer. For example, if IRMB is involved in the 4^{th} layer of H_b matrix then IntM also reads-out CE-LLRs corresponding to 4^{th} H_b layer. Such CE-LLRs and upcoming LLRs (i.e. SPLU outputs) are processed by CVCPU to generate updated 384-LLRCs for the next layer and 384 new CE-LLRs for next iteration (which is stored in IntM), as shown in Fig. 2. 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 −most significant bits (MSBs)− of the stored updated LLRs from IRMB and it takes single clock cycle. Eventually, these stored bits in HDRM are fetched sequentially in every clock cycle.

B. Memory Unit Configuration

The standardized frame-length for 1/3 code-rate in BG1 H_b matrix is 26112 bits for the maximum expansion factor (z = 384), based on 5G-NR specifications [11]. In our design, IRMB is a stack of 68 register memory units (RMUs) in which 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 stores the quantized input LLRs (7-bits each) sequentially and read simultaneously from RMUs in single clock cycle. Thus, 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 H_b matrix layers, where each CE-LLR is of 32-bits in sign-magnitude format. Thereby, CE-LLR combinations has the bit-width of 12288-bits (i.e. 384×32b). Hence, IntM has the memory depth of 46 with 12288-bits of word-length for storing the CE-LLR combinations for all H_b layers. Each word length of IntM stores the CE-LLR combination corresponding to single H_b layer and are read out within one clock cycle. Further, IntM has the size of 565.248 kb (i.e. 46×12288 bits). In addition, HDRM has the 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, overall memory size used in our LDPC decoder is 765.48 kb.

C. CVCPU Architecture

Microarchitecture of proposed CVCPU has been shown in Fig. 3 where 133-bits LLRs and 32-bits CE-LLRs are fed as inputs, as discussed earlier. Firstly, the received CE-LLRs are processed by 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 compliment units (SMTCUs), as shown in Fig. 3. These 19 ex-LLRs (5-bits each) is

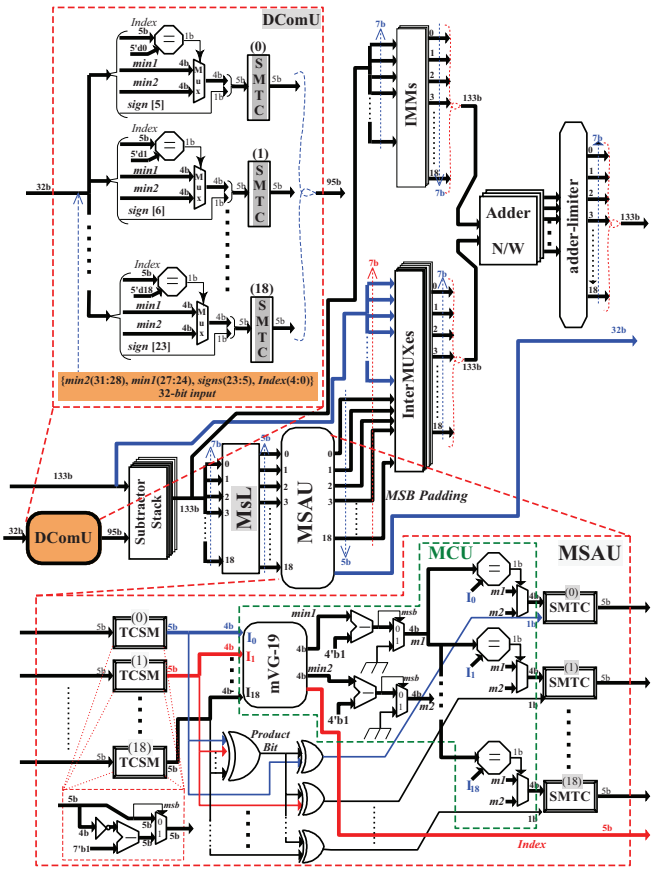

Fig. 3. Suggested VLSI architectures of CVCPU (middle figure), DComU (top figure), and MSAU (bottom figure) used in our QC-LDPC decoder design.

subtracted with incoming LLRs (7-bits each), as described in 13^{th} line of Algorithm 1 using the stack of subtractors. Subsequently, these 19 subtracted LLRs are fed to 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 MSAU delivers 19 updated LLRs (each of 5-bits word-length) and 32-bits CE-LLR to a network of 19 2:1-intermedite MUXes (InterMUXes) and IntM, respectively, of our decoder architecture presented in Fig. 2 & 3. These InterMUXes and IMMs select the updated LLRs and subtracted LLRs, respectively, and their selection processes are based on the H_b layers. For example, from layer 1-4 of H_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 H_b matrix. However, 5^{th} layer of H_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, remaining 16 InterMUXes and IMMs choose the incoming LLRs and

'0' decimal value respectively. These 19 updated and subtracted LLRs are added, as illustrated in 17^{th} line of Algorithm 1, to generate 8-bits updated beliefs (LLRs) for the specific layer. These beliefs pass through the adder-limiter which converts them to 7-bits that ranges from -63 to 63 and transfer them to CMBU of our proposed decoder architecture, as shown in Fig. 2.

D. MSAU Architecture

As discussed earlier, the MSAU microarchitecture is fed with 19 LLRs generated by MsL, as shown in Fig. 3. These 5-bits LLRs are in the 2's compliment format which is converted into 5-bits sign-magnitude (SM) format by 2's compliment-to-sign-magnitude units (TCSMUs). The MSBs of these 19 SM-LLRs, are XORed to generate a product bit, and 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. 3. 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 [12]. 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 '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. 3. Therefore, all 19 magnitudes are replaced by *min1* except the index of *min2*. This replacement is done by 19 equalizers and 2:1 MUXes shown in Fig. 3. Finally, these 19 4-bits updated magnitude from MCU and there corresponding 19 updated sign bits from SU are applied to 19 parallel SMTCUs that convert these SM-LLRs into 2's compliment LLRs.

IV. Performance Analysis, Hardware Implementation and Comparison

The BER performance analyses of the suggested QC-LDPC decoder have been carried out in AWGN channel environment where 26112 encoded bits generated using BG1 matrix of 5G-NR (shown in Fig. 1) are transmitted and decoded for a code-rate of 1/3. We

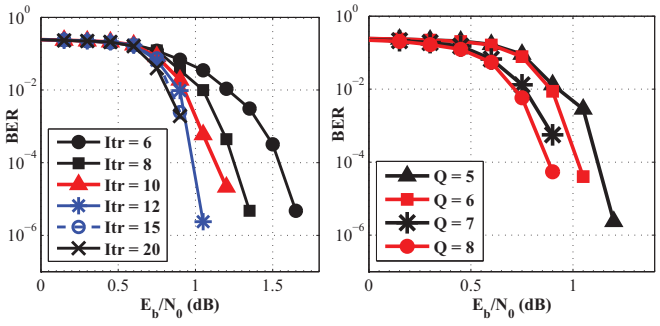

Fig. 4. BER performance analyses of the proposed QC-LDPC decoder for various decoding iterations (Itr.) and bits quantization (Q).

performed comprehensive analyses for various decoding-iterations (ranging from 6−20) and different fixed-point quantization (Q), in the range of 5−8 with 10 decoding iterations, as shown in Fig. 4. It can be observed that the proposed LDPC decoder 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 Q and iterations; however, it comes at the cost of huge area requirement and latency. The proposed decoder has been synthesized and post-route

TABLE I
Comparison of Proposed LDPC Decoder and Reported FPGA-Implementation Results.

Specifications	This work	[13]	[14]	[15]
Quantization	**7-bits**	4-bits	7-bits	8-bits
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 arch.	**Fully-parallel**	partially-parallel	partially-parallel	fully-parallel
H_b 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 algo.	**offset min-sum**	sum-product	min-sum	scaled min-sum

simulated on FPGA platform (Virtex Ultrascale+). Furthermore, the static timing analysis indicates that our design attend timing-closure sign-off at 102.45 MHz of maximum clock frequency (f_{max}), i.e. 9.76 ns of critical path delay. Table I 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 using the formulation $\Theta_T = (N_f \times f_{max}) / (I \times \eta_{clk})$ where the frame-length (N_f) is 26112 bits and I represents number of iterations (i.e. 10 iterations in our work). In addition, proposed decoder requires 2 clock cycles to process each H_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 indicates the hardware utilization as follows: 1448762 lookup tables (LUTs), 211238 registers, 312497 F7 MUXes and 189168 F8 MUXes. Table 1 indicates that our design delivers 31.03% better throughput than the highest throughput achieved by the similar implementation [13] and 20× better than the state-of-the-art LDPC decoder implementation [14].

V. Conclusion

This paper presented a new VLSI architecture of fully-parallel LDPC decoder compliant to 5G-NR wireless-communication standard. To the best of our knowledge, it is the first work that describes detail architectural aspects of all the internal modules of QC-LDPC decoder that is compliant to 5G technology using the specific BG1 matrix. In addition, we presented the comprehensive performance analyses as well as FPGA implementation results. The takeaway point from this paper is in-depth understanding of methodology as well as challenges of designing the next-generation LDPC channel-decoder for the 5G-NR physical layer.

VI. Acknowledgment

The authors like to thank for the support from ICPS Program (IoT Research) of the DST, Government of India.

REFERENCES

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

[2] A. Jimenez Felstrom and K. S. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. on Info. Theo.*, vol. 45, no. 6, pp. 2181-2191, 1999.

[3] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 33, no. 6, pp. 457-458, 1997.

[4] "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", *ETSI EN 302 307-1*, V1.4.1, pp.1-80, 2014.

[5] "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. 802.11-2016 (Revision of IEEE Std 802.11-2012)* , vol., no., pp.1-3534, 14 Dec. 2016.

[6] "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)* , vol., no., pp.1-822, 28 Feb. 2006.

[7] "High performance and area efficient LDPC code design with compact protomatrix" *R1-1613059, 3GPP*, Nov. 2016. [Online], Available:https://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_87/Docs/R1-1613059.zip

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

[9] 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. 50229-50244, 2018.

[10] T. Richardson and S. Kudekar, "Design of low-density parity check codes for 5G new radio," *IEEE Comm. Magazine*, vol. 56, no. 3, pp. 28-34, 2018.

[11] "Chairmans Notes of Agenda Item 7.1.4 Channel coding for NR", *document R1-1709681, 3GPP*, May. 2017. [Online], Available: http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_89/Docs/R1-1709681.zip

[12] C. Wey, M. Shieh and S. Lin, "Algorithms of finding the first two minimum values and their hardware implementation," *IEEE Trans. on Cir. and Sys. I: Regular Papers*, vol. 55, no. 11, pp. 3430-3437, 2008.

[13] C. 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 Trans. on Cir. and Sys. I: Regular Papers*, vol. 60, no. 7, pp. 1857-1869, 2013.

[14] N. Yang et al., "Reconfigurable decoder for LDPC and polar codes,"*IEEE Inter. Sym. on Cir. and Sys. (ISCAS)*, pp. 1-5, 2018.

[15] J. Su and J. Han, "Design of energy efficient LDPC decoders with low-voltage strategy," *IEEE 11th International Conference on ASIC (ASICON)*, pp. 1-4, 2015.