Flexible Encoder and Decoder of Low-Density Parity-Check Codes

Vladimir L. Petrović
School of Electrical Engineering
University of Belgrade
Belgrade, Serbia
Tannera LLC
Belgrade, Serbia
ORCID: 0000-0002-2268-6547

***Abstract*—Hardware implementation aspects of highly flexible low-density parity-check (LDPC) encoder and decoder are presented. The paper covers algorithmic and architectural approaches in achieving flexible and yet very efficient LDPC codec solutions in terms of hardware usage efficiency (HUE). LDPC encoder is implemented to support all possible codes from the 5G New Radio (NR) communication standard, whereas LDPC decoder supports any quasi-cyclic (QC) LDPC or QC-LDPC like code family. Both architectures were optimized using genetic algorithm (GA) for higher throughput or better error correcting (EC) performance results.**

***Keywords*—genetic algorithm optimization, hardware usage efficiency, high throughput, hybrid decoding schedule, low-density parity-check codes, pipeline**

I. Introduction

Flexibility is one of the main requirements for many today's applications of LDPC codes, especially in 5G NR. However, it does not relax the need for efficient and high throughput solutions. On the contrary, the throughput demands are even more challenging. Even though the high throughput can be achieved by using high parallelization techniques, these methods usually impact flexibility or severely increase usage of hardware resources. Therefore, in industrial applications it is a common case that partially parallel architectures are used and optimized for as much high throughput as possible for acceptable amounts of hardware resources [1]. Consequently, the hardware usage efficiency (HUE), expressed as the throughput divided with the chip area or field programmable gate array (FPGA) resources, is one of the main parameters for optimization during the LDPC codec design. This paper presents main achievements of the research on the HUE increase in partially parallel flexible LDPC encoder and decoder architectures.

II. Flexible LDPC encoder for 5G NR

5G NR codes are convenient for forward substitution-based encoding where parity bits are calculated by efficiently solving the system of equations in Galois field GF(2) [2]. Since 5G NR codes are quasi-cyclic, the most time/resource consuming process in LDPC encoding comes down to circular shifts of the groups of bits and exclusive or (XOR) operations. Hardware architecture needs to first calculate so-called core parity bits by shifting and XOR-ing only groups of Z_c information bits, where Z_c is the size of circularly shifted identity matrices (circulants) in the LDPC code parity check matrix (PCM). The next step is calculation of core parity bits by using intermediate results obtained by processing the first four rows of circulants in the PCM. The final step is calculation of so-called additional parity bits, which is architecturally the same as the first step with the only difference that core parity bits are also included in the calculation.

LDPC codes in 5G NR can be of very different lengths, but are all derived from two base graph (BG) matrices lifted by one of many possible lifting factors Z_c [3], which means that the main structure of short and long codes is the same and that the same architecture can be used for encoding of all 5G NR codes. The only necessity is to provide a flexible cyclic shifter network (CSN) that can shift bit vectors of multiple sizes. However, if short codes are used, the blocks of information bits that are circularly shifted are short, which means that a potentially large part of CSN is not used at all. The research presented in this paper, [2], and in [4] proposed even higher flexibility in the CSN, so that it can work as a large CSN when codes with high Z_cs are used, but also as multiple smaller CSNs when codes with lower Z_cs are used. This way, it is possible to encode multiple circulant rows in the PCM simultaneously and hence increase the throughput with potentially small overhead for additional CSN flexibility. This has been achieved by using the property that large shift size cyclic shifts can be divided to multiple smaller shift size cyclic shifts with the simple additional permutations before and after the shifts. Therefore, the proposed architecture effectively uses multiple smaller processing elements for short codes and one large processing element for long codes.

Even though multiple processing elements are available during the encoding of short codes, due to the sparse nature of 5G NR LDPC PCM, it is not common that all elements are used for useful cyclic shifts. Therefore, it has been proposed to rearrange the processing order so that the utilization of CSN increases. The rearrangement is achieved by permuting the rows of the PCM circulants using the GA based optimization, with the objective to shift as much information bit groups as possible at the same time. The flexible CSN architecture and the GA optimization provided significant improvements in encoding throughput for short codes of up to 183%. Consequently, the overall HUE has increased, despite the additional overhead in hardware resources that has been necessary for additional flexibility.

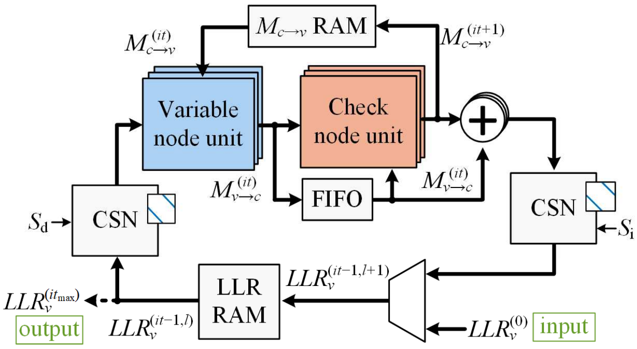
III. Flexible LDPC decoder with hybrid scheduling

LDPC decoders for wireless communications usually use iterative decoding algorithms based on the belief-propagation [5] and its simplifications. Besides the PCM, LDPC codes are frequently represented by the Tanner's graph where PCM columns correspond to the variable

Digital Object Identifier: 10.1109/TELFOR56187.2022.9983744; Final, published article available at: https://ieeexplore.ieee.org/document/9983744

nodes, which keep track of the current codeword values during the decoding, and PCM rows to the check nodes, which act as a parity check modules that give the information about potential updates in variable nodes. The decoding uses the message passing approach where variable and check nodes iteratively send messages to each other. Variable nodes keep probabilities that the codeword bits are "0" or "1" usually stored in the form of the logarithm likelihood ratio (LLR). Iteration is complete whenever variable nodes receive messages from check nodes and update their LLR values accordingly. The described approach is called the simultaneous decoding or flooding decoding schedule [6]. However, the most common approach, especially for QC-LDPC codes, uses the layered schedule [7] where the PCM is processed in layers, e.g. one circulant row after another. This effectively means that variable and check nodes communicate in groups, thus making LLR updates more frequent and consequently increasing the convergence speed.

The architecture of a layered LDPC decoder is shown in Fig. 1. LLR values are stored in LLR RAM. Variable node units (VNUs) calculate variable node messages based on the LLR values and the check nodes messages calculated during the previous iteration and stored to $M_{c\to v}$ RAM memory. LLR updates are done based on the VNU messages and new messages from check node units (CNU). In order to adequately connect variable and check nodes, it is necessary to cyclically shift LLRs according to the PCM.



$$M_{v\to c}^{(it)} = LLR_v^{(it-1,l)} - M_{c\to v}^{(it)}$$

$$M_{c\to v}^{(it+1)} = f\left(\left\{M_{v'\to c}^{(it)} \mid v' \in V_c \setminus v\right\}\right)$$

$$LLR_v^{(it-1,l+1)} = M_{v\to c}^{(it)} + M_{c\to v}^{(it+1)}$$

Fig. 1. The architecture of layered LDPC decoder; M: message; c: index of a check node, v: index of a variable node, (it): denotes a value after iteration it; (it, l): denotes a value after iteration it and layer l in the iteration $it + 1$; S_d: value of the cyclic shift; S_i: the inverse value of the cyclic shifter; V_c: a set of all variable nodes connected to the check node c [4].

In order to obtain high throughput, it is necessary to design the LDPC decoder to work at high operating clock frequency. One of the main techniques for achieving that goal is pipelining. The clock frequency is higher if the number of pipeline stages is higher. However, heavily pipelined layered LDPC decoders suffer from data hazards that need to be avoided in order to maintain valid LLR updates. Variable node must not use its LLR value for message calculation in layer i if the LLR update for layer $i-1$ has not yet happened. Consequently, it is necessary to introduce stall cycles whenever a hazard would occur to wait for the update, which, as opposed to the clock frequency increase, reduces the decoder throughput.

This research proposes a novel approach and architecture for LLR updates called hybrid decoding schedule [4], [8] that decouples the relation between the clock frequency increase and stall cycles by completely removing necessity for stalls. Whenever the out-of-date LLR should be read, the decoder does not wait for the update, but reads the old value. However, the contributions of check nodes to the LLR values in those cases are consequently propagated differently than in situations where LLRs are up-to-date. As a result the algorithm behaves as a flooding decoder when conflicts should be resolved and as a classic layered decoder when there are no pipeline hazards. Hybrid schedule thus enables introducing an arbitrary number of pipeline stages and the clock frequency increase, and yet keeps flexibility so that decoder supports any QC or QC-like LDPC code.

However, occasional switches between layered and flooding schedules can affect EC performance especially if the number of hazards is high. Therefore, it is proposed to rearrange the processing order of the PCM so that the number of out-of-dated LLR reads is minimized. The GA is used for this purpose too. After the optimization, the proposed decoder has obtained almost negligible EC performance loss when compared with the conventional layered decoder. Whenever it is not possible to obtain the same EC performance, it has been proposed to add one or a few more decoding iterations to compensate for the EC performance loss. Even though the iteration number was increased, it has been shown, on the example of 5G NR codes, that the benefits of removing stall cycles are huge and that effectively the hybrid decoder works between 30% and 110% faster than the conventional layered decoder for the same EC performance.

As a case study, the FPGA implementation for WiMAX, DVB-S2X, and 5G NR provided coded throughput of up to 1.77 Gbps, 4.32 Gbps, and 4.92 Gbps at 10 iterations, respectively. The design has been compared with the state of the art solutions and it has been shown that the HUE is significantly increased without losing flexibility.

REFERENCES

[1] Z. Wang and Z. Cui, "Low-complexity high-speed decoder design for quasi-cyclic LDPC codes," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 15, pp. 104–114, January 2007.

[2] V. L. Petrović, D. M. El Mezeni, and A. Radošević. "Flexible 5G New Radio LDPC Encoder Optimized for High Hardware Usage Efficiency," Electronics, vol. 10, p. 1106, May 2021.

[3] *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; NR; Multiplexing and channel coding (Release 16)*, 3GPP TS 38.212 V16.1.0 (2020-03), 2020.

[4] V. L. Petrović, "Fleksibilni koder i dekoder kodova sa proverama parnosti male gustine," University of Belgrade – School of Electrical Engineering, 2021.

[5] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," Electron. Lett., vol. 32, no. 18, pp. 1645–1646, August 1996.

[6] F. R. Kschischang and B. J. Frey, "Iterative decoding of compound codes by probability propagation in graphical models," IEEE J. Select. Areas Commun., vol. 16, no. 2, pp. 219– 230, February 1998.

[7] D. E. Hocevar, "A reduced complexity decoder architecture via layered decoding of LDPC codes," in Proc. IEEE Work. Signal Process. Syst., Austin, TX, USA, October 2004.

[8] V. L. Petrović, M. M. Marković, D. M. El Mezeni, L. V. Saranovac, and A. Radošević, "Flexible High Throughput QC-LDPC Decoder with Perfect Pipeline Conflicts Resolution and Efficient Hardware Utilization," IEEE Trans. Circuits Syst. I: Reg. Papers, vol. 67, pp. 5454-5467, December 2020.