HARDWARE-FRIENDLY LDPC DECODING SCHEDULING FOR 5G HARQ APPLICATIONS

Cing-Yi Liang†, Mao-Ruei Li†, Huang-Chang Lee‡, Hsin-Yu Lee†, Yeong-Luh Ueng†∗

†Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan
‡Department of Electrical Engineering, Chang Gung University, Taipei, Taiwan
∗Institute of Communications Engineering, National Tsing Hua University, Hsinchu, Taiwan

ABSTRACT

This paper presents hardware-friendly LDPC decoding schedules for 5G hybrid automatic repeat request (HARQ) applications. Since there are built-in punctured blocks in the parity check matrix (PCM), a scheduling technique is proposed that allows the punctured nodes to be efficiently recovered. For HARQ using the Chase combining (CC), the previous decoding results corresponding to the punctured part are retained, and the proposed layered decoding is arranged according to the row weight. For HARQ using the incremental redundancy (IR) approach, the parity bits corresponding to the pure-row-orthogonal part of the PCM are transmitted first. The hardware implementation shows that the throughput can be increased by 21.37% for the first decoding attempt, 56.9% for CC-HARQ and 14.51% for IR-HARQ when the code rate reaches 0.303.

***Index Terms*—** LDPC codes, 5G, HARQ, Chase combining, incremental redundancy

1. INTRODUCTION

Low-density parity-check (LDPC) codes [1][2] can provide near-capacity performance using a simple iterative belief propagation algorithm and are suitable for parallel implementation. Recently, LDPC codes have been chosen for the enhanced mobile broadband data channel in the new 5G radio. In order to increase the possibility of successful decoding, rate-compatible LDPC codes for hybrid automatic repeat request (HARQ) applications is desired for time-varying channels. A variety of schedules have been proposed for LDPC decoding, such as two-phase message passing (TPMP) [2], layered message passing decoding (LMPD) [3][4], shuffled BP decoding (SBPD) [5][6], optimized fixed schedules [7][8], and dynamic schedules [9]. Both SBPD and LMPD are hardware-friendly and can accelerate the decoding convergence such that it is about two times faster than for TPMP. Since the 5G LDPC codes are constructed by extension, LMPD is a more suitable schedule, and hence LMPD is considered in this work.

For 5G LDPC codes, the variable nodes corresponding to the first two block columns of the parity-check matrix (PCM) can be punctured. Many previous studies show that for punctured codes, more iterations are required in order to achieve the decoding convergence. The authors of [10] proposed a layered decoding technique to cope with the issue, where the decoding order is arranged according to the survived check node of each layer in the PCM. However, it is difficult to identify the survived check nodes in the 5G PCMs. In this paper, we propose an efficient LMPD for the 5G codes, where the processing of layers is scheduled based on the number of punctured edges and the check-node degree. The purpose is to efficiently recover the punctured nodes, and to reduce the required number of iterations.

When the decoding for the first received packet fails to converge, retransmission is required. When HARQ using the Chase combining (CC) [11] approach is adopted, in addition to directly combining the two received packets, the results of the first two punctured block columns generated by the first decoding attempt are also used in the second decoding attempt. It can be observed that the required number of iterations for the second decoding attempt can be reduced. For HARQ using the incremental redundancy (IR) approach, in order to provide the most contribution to the second decoding attempt, the order that the remaining parity bits are transmitted is investigated. The investigation shows that when the parity bits corresponding to the pure-row-orthogonal part of the PCM are transmitted earlier than the quasi-row-orthogonal part, the convergence speed of the second decoding attempt can be significantly improved. Based on the hardware implementation, the decoder throughput can be increased by 21.37% in the first decoding attempt. Moreover, the throughput can be increased by 56.9% for CC-HARQ and 14.51% for IR-HARQ when the code rate reaches 0.303.

2. PRELIMINARY

2.1. 5G rate-compatible LDPC coding

The PCM of the quasi-cyclic LDPC codes can be constructed using a base matrix. Each non-zero element can be expanded to a $z \times z$ circulant permutation matrix with a defined shift index, and each zero element is expanded to a $z \times z$ zero matrix. Fig. 1 shows the structure of the 5G PCM: part A is a compact

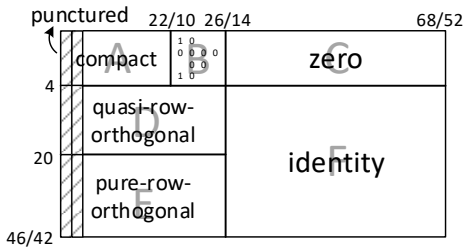

Fig. 1. The structure of the PCM in the 5G standard. The numbers next to the matrix indicate the number of block columns or rows, and they are the boundaries of different part for base graph 1 or 2.

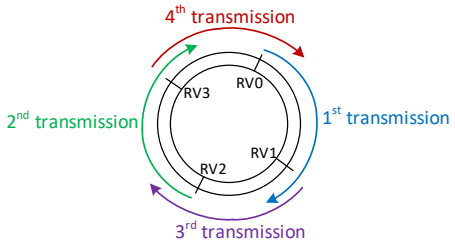

Fig. 2. The HARQ codeword circular buffer.

matrix, the first block column of part B has weight 3 and the remaining part is dual-diagonal, part C is an all-zero matrix, part D is a quasi-row-orthogonal matrix for which the position of the non-zero element between two consecutive block rows are not the same except for the first two block columns, part E is a pure-row-orthogonal matrix, and part F is an identity matrix. Moreover, the row weight of D is less than A, and E is less than D. There are 2 base graphs and each has 8 sets of base numbers. The circulant size (z) is equal to the base numbers multiplied by 2^j, where $0 \leq j \leq 7$. Note that the first two block columns corresponding to the information bits can be punctured. To decode the complete codeword, the decoder pads zeros at the punctured locations since the receiver does not receive any information at the punctured locations.

2.2. HARQ mechanism

To increase the transmission efficiency, HARQ which is a combination of forward error-correcting coding and ARQ error-control, is adopted in the 5G standard. The transmitted data is encoded with a forward error-correcting code, and the parity bits are either sent immediately together with the message, or only transmitted upon request when a receiver detects an erroneous message. After the receiver receives the retransmitted data, the receiver combines the retransmitted data and the prior transmitted data to enhance the decoding performance of the retransmissions.

The simplest version of HARQ is CC-HARQ, which directly combines the previous received message and the new message [11]. The retransmission packet is the same as the previous packet. The decoder uses the new packet to increase the probability of successful decoding. Another version of HARQ is IR-HARQ. All information bits are encoded using the lowest rate, and the resultant codeword is stored in a codeword circular buffer, as shown in Fig. 2. The transmitter only transmits a codeword with the highest rate in the initial step, as part A and B shown in Fig. 1. At every retransmission, the transmitter only retransmits the remaining parity bits to save energy compared to the CC approach. In the 5G standard, the information is separated into four redundancy versions (RVs), and they are at fixed locations in the codeword circular buffer. RVs can be located at any part, except for positions A and B indicated in Fig. 1.

3. PROPOSED EFFICIENT LAYERED DECODING SCHEDULING

In this section, we discuss the proposed scheduling techniques designed to reduce the number of required iterations. Both stand alone LDPC codes and HARQ are considered.

3.1. Decoding scheduling for 5G LDPC codes

3.1.1. Without puncturing in the systematic part

It is known that the low row-weight (check node degree) layer connects to fewer variable nodes. Consequently, the received codeword using the initial parity bits can be gradually corrected when decoding the lowest row-weight layer first. Then, the decoding proceeds using the more reliable *a posteriori* log-likelihood ratio (LLR) values. Therefore, the decoded word is more likely to be legal, so that the average number of iterations can be reduced. Therefore, scheduling from the lowest row-weight layer first is proposed and denoted as *LD*. Although the use of *LD* scheduling can not improve the average number of iterations for the highest rate 0.846 since the row weights are the same, it can significantly improve the lowest-rate case, i.e., rate = 0.324. Fig. 3 shows the BER curve and the average number of iterations for the conventional (*Conv.*) scheduling and the proposed *LD* scheduling. The proposed *LD* scheduling reduces the average number of iterations by 15.6% at $E_b/N_0 = 4.0$ dB. Note that the channel model used is the independent Rayleigh fading channel, and it is the same in the rest of the discussion.

3.1.2. With puncturing in the systematic part

Considering the punctured case in the 5G standard, *LD* scheduling also accelerates the convergence by 16.36% as

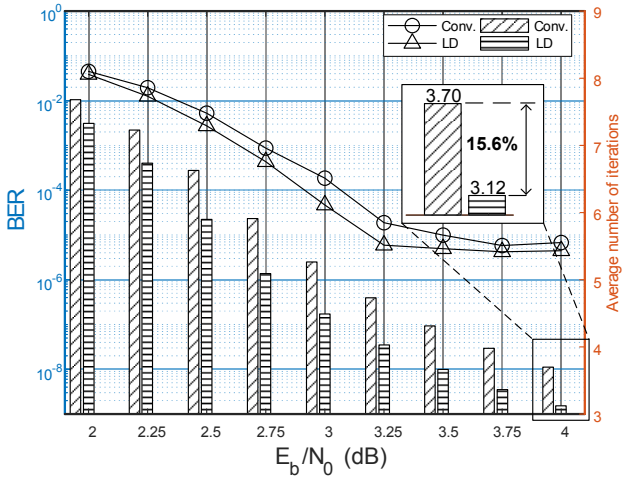

Fig. 3. BER performance and the average number of iterations for non-punctured codes. (rate = 0.324, length = 384×68 bits)

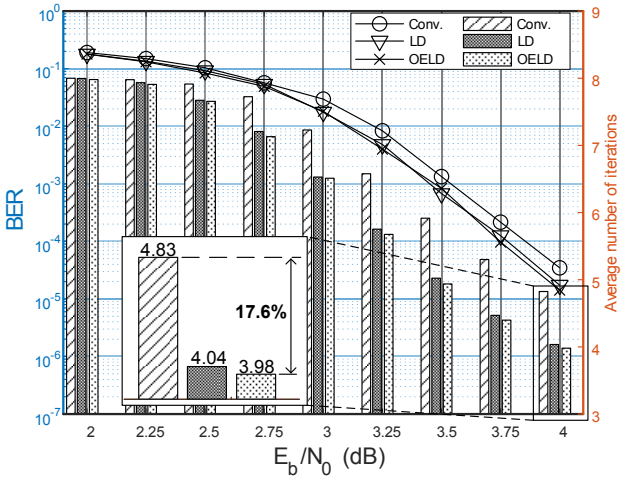

Fig. 4. BER performance and the average number of iterations for punctured codes. (rate = 0.303, length = 384×68 bits)

shown in Fig. 4. As there are two punctured blocks in the transmission, and the decoder pads zeros at the punctured locations, the punctured bits are unreliable, so they should not be used for decoding. Therefore, if we decode the layer with more punctured edges first, the unreliable punctured nodes may influence the correctness of the complete codeword. Conversely, if we decode the layer with less punctured edges first, the results are more likely to be legal. Therefore, we propose decoding from the one-punctured-edge layers first, which is denoted as *OE*. We can combine the *OE* and *LD* techniques to obtain the *OELD* scheduling. As shown in Fig. 4, the proposed *OELD* scheduling reduces the average number of iterations by 17.6% at E_b/N_0 = 4.0 dB. Moreover, the BER performance is also improved by 0.125 dB at E_b/N_0 = 4.0 dB.

3.2. Decoding scheduling for 5G LDPC codes in HARQ

3.2.1. CC-HARQ

For CC-HARQ, the decoder combines the old and new packets by adding the previous and the retransmitted channel values which is denoted as $Conv._{cc}$. Similarly, the *OELD* scheduling can reduce the average number of iterations by up to 17.39%. In the $Conv._{cc}$ scheme, the first two blocks are punctured for both transmissions, so it is similar to the case of decoding the punctured codes. Therefore, the *OELD* scheduling has the least average number of iterations when using the $Conv._{cc}$ scheme. Because of the built-in puncturing blocks, the retransmission packet still does not contain any information for the first two block columns. Therefore, we use the *a posteriori* LLR values for the first two block columns obtained in the first decoding attempt to increase the amount of information available in the second decoding attempt. The remaining variable nodes are combined as the conventional CC scheme. This scheme is denoted as $Prop._{cc}$. A system buffer is included to store the channel values and, hence, additional memory is not required. In addition, the average number of iterations can be further reduced if the low row-weight layer takes precedence over the one punctured-edge layer. The resultant scheduling is called *LDOE* scheduling. The $Prop._{cc}$ scheme can be viewed as a case of decoding without puncturing, so the row weight affects the average number of iterations more than the number of punctured edges. The results are shown in Fig. 5. The overall improvement from $Conv._{cc}$ to $Prop._{cc}$+*LDOE* is up to 36.27%.

3.2.2. IR-HARQ

Intuitively, the more retransmitted parity bits, the better the BER performance, so we fix the length of the retransmission, which we assume to be 14 blocks, and the part which is not received is padded with zeros. Based on the results mentioned above, the row weight for the decoding layer can significantly affect the number of iterations. So, considering the row-weight conditions, we set two locations for the retransmitted parity bits, including from the quasi-row-orthogonal part (*QO*) and from the pure-row-orthogonal part (*PO*), where the row-weight for *PO* is less than for *QO*. As shown in Fig. 6, retransmitting the parity-bits corresponding to the low row-weight layer first i.e., the *PO* part, can improve the performance by 0.2 dB at BER = 10^{-3}.

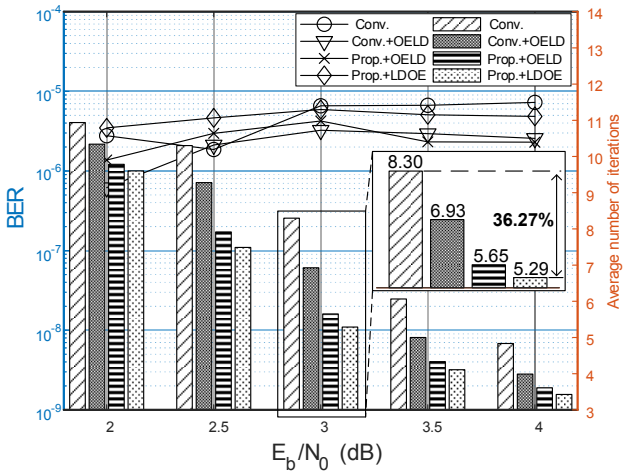

Fig. 5. BER performance and the average number of iterations for CC-HARQ. (rate = 0.303, and length = 384×68 bits)

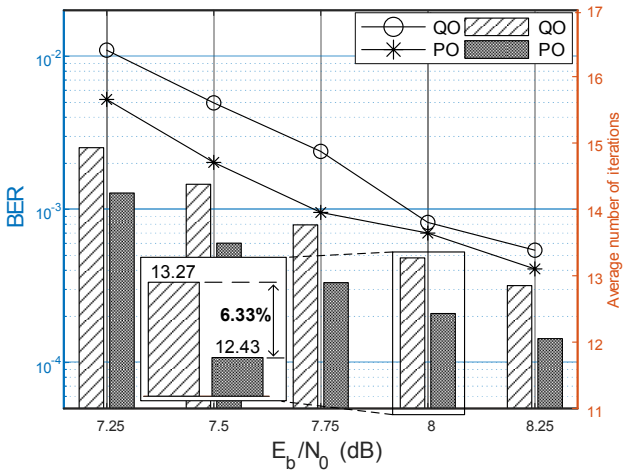

Fig. 6. BER performance and the average number of iterations for IR-HARQ. (rate = 0.303, and length = 384×26 bits for the initial transmission and 384×68 bits for the second decoding)

3.3. Hardware design for the proposed scheduling

Fig. 7 shows the proposed hardware architecture, where the LDPC decoder follows the conventional layered decoding [12] approach. The CC-HARQ mode requires the LLR memory in order to store the *a posteriori* LLR values for the first two block columns. The IR-HARQ mode uses a concatenator to concatenate the previous and the retransmitted parity-check bits. To apply the proposed combining scheme, we need an additional $384 * 2$ adders and MUXs before the LDPC decoder. As for the reserved blocks, there are sufficient system buffers that we can utilize, so additional memory space is not needed. Furthermore, only a small number of ROMs are needed in order to store the proposed decoding order. So, to implement the decoder based on the proposed scheduling, the additional hardware resources required is minor.

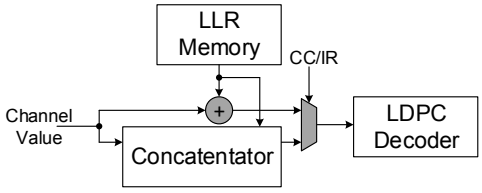

Fig. 7. The hardware architecture for HARQ. The shaded part is added to support the proposed scheduling.

Table 1. The synthesized results for the proposed hardware on an FPGA device (Virtex-7 xc7vx485t).

	LUTs	FF Pairs	Throughput (Mb/s)		
			no HARQ	CC	IR
Conv.[12]	215673	69335	288.42	167.83	210.05
Proposed	224888	69335	350.06	263.33	240.52

Based on the hardware architecture described above, we implemented the decoder on an FPGA device, Virtex-7 xc7vx485t, and the estimated frequency is 46.5 MHz after being synthesized using Synplify, as shown in Table 1. Using the scheduling described in Section 3.1, the throughput can be increased by 21.37%, 18.6%, 3.84%, and 1.12% for rate-0.303, 0.324, 0.833, and 0.846 codes, respectively. The throughput for the CC-HARQ scheme can be increased by 56.9% and 4.28% for rate-0.303 and 0.833 codes, respectively. Finally, the throughput can be increased by 14.51% for IR-HARQ.

4. CONCLUSION

The throughput of the 5G LDPC codes has been improved using the proposed scheduling techniques. For the first standalone decoding attempt, the average number of iterations can be reduced by decoding from the layers connecting to the single punctured variable node or the low row weight layers. For CC-HARQ, the decoding results for the first two punctured block columns of the first decoding attempt can be maintained to assist the second decoding attempt, and accelerate the convergence. Finally, the best location of the redundancy version which requires the least average number of iterations for IR-HARQ has been evaluated. Based on the simulation and the hardware synthesized results, the throughput can be increased obviously.

5. REFERENCES

[1] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, pp. 21–28, 1962.

[2] D. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, 1999.

[3] J. C. J. Chen and P. Fossorier, "Density evolution for BP-based decoding algorithms of LDPC codes and their quantized versions," *IEEE Global Telecommunications Conference*, vol. 2, no. 5, pp. 1378–1382, 2002.

[4] M.-R. Li, C.-H. Yang and Y.-L. Ueng, "A 5.28-Gb/s LDPC decoder with time-domain signal processing for IEEE 802.15.3c applications," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 2, pp. 592-604, Feb. 2017.

[5] J. Zhang and M. Fossorier, "Shuffled belief propagation decoding," *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 8–15, 2002.

[6] C. A. Aslam, Y.-L. Guan, K. Cai, "Improving the belief-propagation convergence of irregular LDPC codes using column-weight based scheduling," *IEEE Communications Letters*, vol. 19, no. 8, pp. 1283–1286, AUG. 2015.

[7] H.-C. Lee and Y.-L. Ueng, "LDPC decoding scheduling for faster convergence and lower error floor," *IEEE Transactions on Communications*, vol. 62, no. 9, pp. 3104–3113, 2014.

[8] H.-C. Lee and Y.-L. Ueng, "Incremental decoding schedules for puncture-based rate-compatible LDPC codes," *IEEE 83rd Vehicular Technology Conference (VTC Spring)*, pp. 1–5, 2016.

[9] H.-C. Lee, Y.-L. Ueng, S.-M. Yeh, and W.-Y. Weng, "Two informed dynamic scheduling strategies for iterative LDPC decoders," *IEEE Transactions on Communications*, vol. 61, no. 3, pp. 886–896, 2013.

[10] J. Ha, D. Klinc, J. Kwon, and S. W. Mclaughlin, "Layered BP decoding for rate-compatible punctured LDPC codes," *IEEE Communications Letters*, vol. 11, no. 5, pp. 440–442, 2007.

[11] Y. Wu and H. Yang, "Optimising energy efficiency of LDPC coded chase combining HARQ system," *Electronics Letters*, vol. 51, no. 6, pp. 490–492, 2015.

[12] H.-C. Lee, M.-R. Li, J.-K. Hu, P.-C. Chou, and Y.-L. Ueng, "Optimization techniques for the efficient implementation of high-rate layered QC-LDPC decoders," *IEEE Transaction on Circuits and Systems I: Regular Papers*, vol. 64, no. 2, pp. 457–470, 2017.