# A Low Valid Throughput Loss LDPC Codec Architecture with Variable Code Rate

Jiaming Liu, Zhijian Hao, Lingxiu Zhao, Jiazheng Lian, Jun Tao, *Senior Member, IEEE,* Yibo Fan, *Member, IEEE,* and Quanyuan Feng, *Senior Member, IEEE*

***Abstract*—Shortening is a rate-compatible technology that can improve LDPC error correction performance by modifying the code rate. However, as the code rate decreases, the throughput of valid data drops rapidly. Reducing the attenuation of valid throughput at low code rates is challenging. For this reason, this brief proposes two contributions. First, a Block Tag-based encoding method is presented with a non-fixed code length, which reduces the latency caused by fixed-length communications. Second, a novel Time Division Multiplexing (TDM) architecture is developed to improve valid throughput, reducing resource utilization compared to conventional multiplexers. Based on the above contributions, the codec is implemented on FPGA and the results show that the valid throughput loss is low in changing from high to low code rates.**

***Index Terms*—LDPC codec, throughput loss, time division multiplexing, non-fixed code length, FPGA.**

## I. Introduction

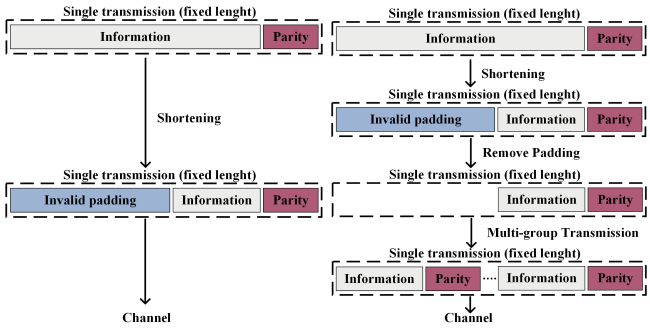
RATE-COMPATIBLE technology, a method for tuning LDPC error correction performance, is widely used in Internet of Things (IoT) applications [1]. Particularly, IoT edge platform requires dynamically adjusting the code rate to accommodate time-varying channels during mobility. LDPC base code is modified by puncturing, extending, or shortening to generate multi-rate LDPC codes with different performances [2]. In recent years, rate-compatible LDPC codecs have been widely proposed. Multi-rate codecs [3] [4] have been optimized through logic-sharing [5], algorithm improvements [6], and the construction of LDPC codes [7], among others.

However, in resource-constrained IoT devices, reducing the code rate by shortening can lead to a pronounced decline in throughput. Shortening introduces invalid padding bits that occupy part of the bandwidth in the fixed-length transmission of the rate-compatible codec, as shown in Fig. 1(a). Adding invalid information increases the latency of receiving and decoding. In the proposed method, invalid padding bits are removed, as shown in Fig. 1(b). The remaining bandwidth is used to transmit multiple encoded frames in the proposed non-fixed-length codec, improving the valid throughput and reducing the valid throughput loss.

Manuscript created February, 2023; This work was supported in part by the Important Project of the National Natural Science Foundation of China under Grant 62090012. *(Corresponding author: Quanyuan Feng.)*

Jiaming Liu, Zhijian Hao, Jiazheng Lian, Jun Tao, and Yibo Fan are with State Key Laboratory of Integrated Chips and Systems, Fudan University, Shanghai 201203, China (e-mail: liujm22@m.fudan.edu.cn; taojun,fanyibo@fudan.edu.cn).

Lingxiu Zhao and Quanyuan Feng are with the Institute of Microelectronics, Southwest Jiaotong University, Chengdu 611756, China (email: fengquanyuan@swjtu.edu.cn).




(a) Conventional method (b) Proposed method

Fig. 1. The transmission method with shortening technology.

In this brief, two methods are proposed to address the reduction of valid bandwidth and the multiple decoding latencies. **First**, a shortening technology with Block Tag can identify information block and parity check block to achieve multi-group transmission with non-fixed code length, which removes invalid padding bits to reduce throughput loss for block-based QC-LDPC code. **Second**, a miniaturized TDM decoder architecture with a shorter iteration calculation chain is presented to enhance valid throughput and reduce valid throughput loss in multi-group transmission.

The remainder of this brief is organized as follows: valid throughput loss is described in Section II, and the non-fixed-length transmission method and the TDM decoder optimization scheme are proposed in Section III. The low valid throughput loss LDPC codec architecture is implemented and verified in Section IV; finally, Section V concludes this brief.

## II. Valid Throughput Loss for LDPC Codec

The throughput is fixed for the circuit of conventional LDPC codecs, but the invalid padding bits of shortening reduce the amount of valid information in a single transmission. The evaluation of LDPC codecs with variable code rates cannot rely on throughput alone. The system must be evaluated by valid throughput, i.e., the throughput of valid information.

The throughput $TP$ is defined as:

$$TP = \frac{I}{T_{codec}} \times f_{clock}, \tag{1}$$

where $I$ denotes the number of information bits, $T_{codec}$ means the latency of the codec, and $f_{clock}$ shows the system's clock frequency. As can be further known, the valid throughput $TP_v$ is shown as:

$$TP_v = \frac{I_{valid}}{T_{codec}} \times f_{clock} = \frac{I - I_{pad}}{T_{codec}} \times f_{clock}, \tag{2}$$

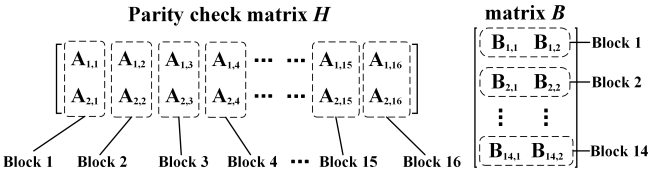



Fig. 2. The parity check matrix $H$ of C2 code and the submatrix $B$ of the generator matrix $G$. In the proposed method, matrix $H$ is divided into 16 blocks, and matrix $B$ is divided into 14 blocks.

where $I_{pad}$ represents the number of invalid information bits. Valid throughput ($TP_v$) can quantify the performance of codec in transmitting valid information ($I_{valid}$).

For multi-rate LDPC codecs, the valid throughput ($TP_v$) drops from high to low code rate ($r$). Therefore, the valid throughput loss ($TP_{Loss}$) is proposed to measure the throughput attenuation when transforming the code rate, shown as:

$$TP_{Loss} = \frac{TP_{vmax} - TP_{vmin}}{TP_{vmax} \times (r_{max} - r_{min})}, \tag{3}$$

where $TP_{vmax}$ denotes the maximum valid throughput at the maximum available code rate ($r_{max}$), $TP_{vmin}$ represents the minimum valid throughput at the minimum available code rate ($r_{min}$). $TP_{Loss}$ means the attenuation ratio of valid throughput at normalized code rates, which is utilized to quantify valid throughput attenuation as the code rate varies. Generally, a lower valid throughput loss indicates a more stable throughput in rate-compatible system.

## III. Proposed Method

### A. Non-fixed-length Transmission based on Block Tag

The conventional codecs only support fixed-length transmission, which means that a large number of invalid padding bits ($I_{pad}$) are filled in a single transmission when shortening. For this reason, a Block Tag-based transmission method is proposed to remove the invalid padding bits and reduce circuit complexity, wherein the block serves as the smallest unit.

The Consultative Committee for Space Data Systems (CCSDS) Near-Earth standard LDPC code (C2 code) is used as an example. The code rate of C2 basic LDPC code ($H$) is 7/8, as shown in Fig. 2. The parity check matrix ($H$) is composed of $2 \times 16$ sparse submatrices ($A_{i,j}$) of dimension $511 \times 511$. The calculation of encoding is expressed as $[s, p] = s \times G$, $s$ means the information vector (7154 bits), and $p$ denotes the parity vector (1022 bits), i.e., the encoded data has a total of 8176 bits. The matrix $G$ is defined as the generator matrix satisfying $G \cdot H^T = \mathbf{0}$, which is composed of an identity matrix $E$ and a matrix $B$. The dimension of the matrix $E$ is $7154 \times 7154$, and the matrix $B$ is composed of $14 \times 2$ cyclic sub-matrices.

In the proposed method, the parity check matrix $H$ is divided into 16 blocks according to the size of the submatrix, and the matrix $G$ is divided into 14 blocks, as shown in Fig. 2. When shortening, one or more blocks with the same serial number in matrix $H$ and $G$ are used as invalid padding blocks, i.e., the smallest unit of shortening is a block. There are a total of 14 available code rates. The dense location of the invalid

TABLE I
The Encoded Vector Transmitted at Available Code Rates

| Original Rate($r_o$) | New Rate($r_n$) | Vector Length | $\frac{I_{valid}(Non-fixed)}{I_{valid}(fixed)}$ |
|---|---|---|---|
| 7/8 | $7\beta/8$ | 14 Blocks($I$)+2 Blocks($p$) | $1.000\beta$ |
| 13/15 | $13\beta/15$ | 13 Blocks($I$)+2 Blocks($p$) | $1.067\beta$ |
| 6/7 | $6\beta/7$ | 12 Blocks($I$)+2 Blocks($p$) | $1.143\beta$ |
| 11/13 | $11\beta/13$ | 11 Blocks($I$)+2 Blocks($p$) | $1.231\beta$ |
| 5/6 | $5\beta/6$ | 10 Blocks($I$)+2 Blocks($p$) | $1.333\beta$ |
| 9/11 | $9\beta/11$ | 9 Blocks($I$)+2 Blocks($p$) | $1.455\beta$ |
| 4/5 | $4\beta/5$ | 8 Blocks($I$)+2 Blocks($p$) | $1.600\beta$ |
| 7/9 | $7\beta/9$ | 7 Blocks($I$)+2 Blocks($p$) | $1.778\beta$ |
| 3/4 | $3\beta/4$ | 6 Blocks($I$)+2 Blocks($p$) | $2.000\beta$ |
| 5/7 | $5\beta/7$ | 5 Blocks($I$)+2 Blocks($p$) | $2.286\beta$ |
| 2/3 | $2\beta/3$ | 4 Blocks($I$)+2 Blocks($p$) | $2.667\beta$ |
| 3/5 | $3\beta/5$ | 3 Blocks($I$)+2 Blocks($p$) | $3.200\beta$ |
| 1/2 | $1\beta/2$ | 2 Blocks($I$)+2 Blocks($p$) | $4.000\beta$ |
| 1/3 | $1\beta/3$ | 1 Block($I$)+2 Blocks($p$) | $5.333\beta$ |

[1] $\beta = L/(L + N_{tag})$, where $L$ is the original block length.

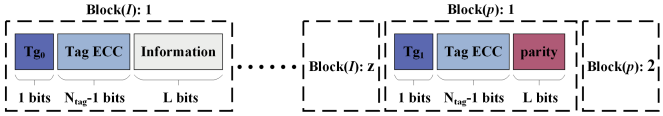



Fig. 3. The encoded data frame with code rate $\frac{z \times \beta}{z+2}$, $z$ is the number of Blocks($I$).

padding bits retains the properties of the cyclic submatrices when removing invalid blocks. In the proposed method, the vectors of invalid blocks are not transmitted, and the length of encoded frame transmitted at available code rates is shown in Table I. Block($I$) represents the information vector and Block($p$) indicates the parity check vector. Under the encoded vector with the same length (16 blocks), the valid information of the proposed non-fixed length method $I_{valid}$(Non-fixed) is more than that of the fixed length method $I_{valid}$(fixed) at a low code rate. Increasing the valid information can significantly improve the valid throughput.

Block Tag is introduced as the block identification (1 bit) to obtain frame length. The tag of the Block($I$) ($Tg_0$) is used as the start bit of the block to indicate that the block is an information vector, and the tag of the Block($p$) ($Tg_1$) signifies that the block is a parity vector. Since the parity blocks are two Blocks($p$) at any code rate, the number of $Tg_0$ ($M_{tg0}$) is counted by decoder until two consecutive $Tg_1$s are detected, then the number of blocks in a single frame is $M_{tg0} + 2$.

However, the channel's noise may impact the accuracy of Block Tag. Therefore, it is necessary to introduce forward error correction (FEC) techniques in Block Tag to improve the accuracy of Block Tag. In the brief, N-modular redundancy is employed as error correction for Block Tag, i.e., the encoder sends $N_{tag}$ identical Block Tags, and the decoder uses the most frequent value as the true Block Tag. Hence, the encoded data frame with code rate $\frac{z \times \beta}{z+2}$ is shown in Fig. 3.

In principle, the error correction performance of Block Tag is much more excellent than that of the data. As shown in Fig. 4, the experimental results show that the number of Tag errors decreases as the length of Block Tag increases. When $EbNo > -4.5dB$, no error occurs with Block Tag (33 bits) from the Additive White Gaussian Noise (AWGN) channel.

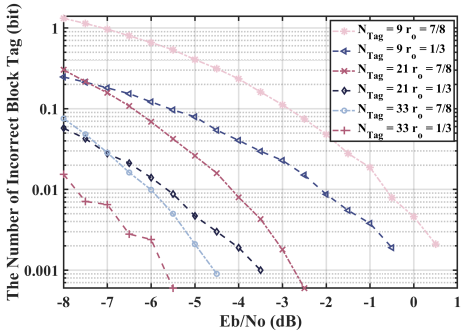
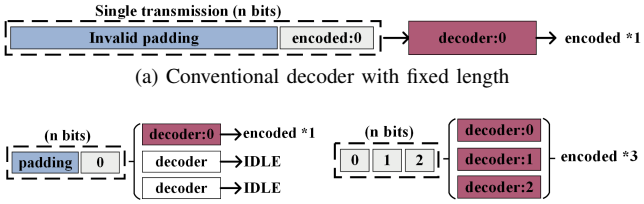



Fig. 4. The number of incorrect Block Tags under AWGN channel.




(a) Conventional decoder with fixed length

(b) Multiplexer (fixed length) (c) Multiplexer (non-fixed length)

Fig. 5. The decoders with variable methods.

Thus, a suitable length of Block Tag ($N_{tag}$) can ensure the accuracy of Block Tag.

### *B. A Miniaturized TDM Decoder Scheme*

Fixed-length decoders face challenges in parallel processing within multiplexers (Fig. 5(b)). The main reason is the long receiving latency of invalid padding bits, which leads to extended intervals between two frames in multiplexers and results in low improvement of multiplexers compared to a single decoder (Fig. 5(a)). However, in the proposed non-fixed-length method (Fig. 5(c)), the multiplexer computes multiple frames with no intervals in parallel, improving the throughput in a single transmission. When the code rate is lower, the number of parallelisms is higher, which reduces valid throughput loss. The resource utilization of conventional multiplexers is exceptionally high. In the brief, a miniaturized TDM decoder is presented to reduce decoding latency and resource utilization, using the Bit Flipping (BF) algorithm as an example.

A number of symbols are defined in this section. The dimension of the parity check matrix $H$ is $m \times n$. Let $N(i)$ and $M(j)$ be $N(i) = \{j|h_{i,j} = 1, j \in [1,n]\}$ and $M(j) = \{i|h_{i,j} = 1, i \in [1,m]\}$. $h_{i,j}$ is the element in the $i$-th row and $j$-th column in the matrix $H$.

Assuming that the bipolar code $(+1,-1)$ of the data with noise obtained at the decoder is $X$, the least credible variable node is flipped in each iteration of BF algorithm. According to:

$$f^{(t)} = \arg\min_{f}\Big(\sum_{i\in M(f)}\prod_{j\in N(i)} X_j^{(t)}\Big), \tag{4}$$

$f^{(t)}$ denotes the serial number of the least credible variable node in the $t$-th iteration. The calculation chain of BF algorithm is illustrated in Fig. 6. The single iteration chain contains multiplication, accumulation, *argmin*, and flipping in **4** levels.

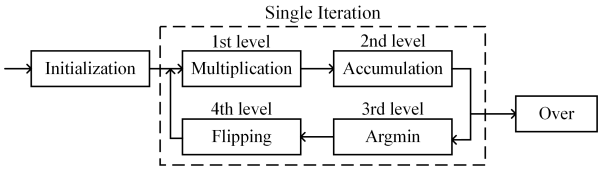



Fig. 6. The calculation chain of BF algorithm.

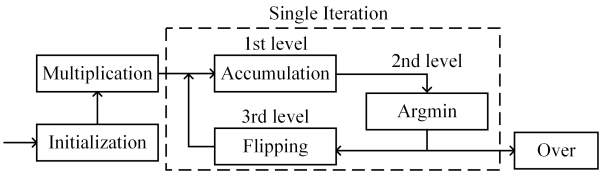



Fig. 7. The calculation chain of the proposed single-bit decoding algorithm.

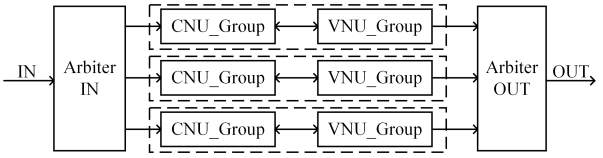



Fig. 8. The conventional multiplexer for BF algorithm.

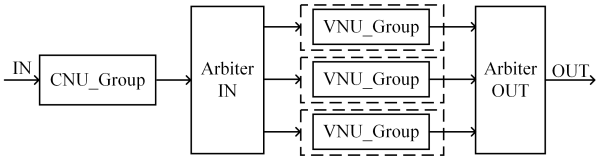



Fig. 9. The proposed TDM decoder for the proposed algorithm.

As the BF algorithm is a single-bit decoding algorithm, only one bit ($X_f$) is flipped in a single iteration. Although the variable nodes $X^{(t)}$ of the $t$-th iteration differs from the variable nodes $X^{(t-1)}$ of the ($t$-1)-th iteration only by 1 bit, the reliability still needs to be recalculated, and the unreasonable calculation increases the delays in decoding.

In this brief, a proposed single-bit decoding algorithm is developed based on a shorter iteration chain. The vector $V^{(t)}$ is defined as the value of the $t$-th iterative multiplication as:

$$V_i^{(t)} = \prod_{j\in N(i)} X_j^{(t)}. \tag{5}$$

Since only 1 bit differs between $X_j^{(t)}$ and $X_j^{(t-1)}$, then only the number of $M(j)$ bits differ between $V^{(t)}$ and $V^{(t-1)}$, so Eq. (4) can be updated to:

$$V_i^{(t)} = -V_i^{(t-1)}(i \in M(f^{(t-1)})), \tag{6a}$$

$$f^{(t)} = \arg\min_{f}\Big(\sum_{i\in M(f)} V_i^{(t)}\Big)(t>1). \tag{6b}$$

Obviously, the calculation of the flipping is simpler than the multiplication, the flipping of $V$ and $X$ can be calculated in parallel. The calculation chain of the proposed single-bit decoding algorithm is shown in Fig. 7, and the single iteration chain contains accumulation, *argmin*, and flipping in **3** levels.

TABLE II
HARDWARE IMPLEMENTATION OF ENCODER

| Work | Rate | Length (bits) | Multi Rate | FPGA technology | Resource utilization | Cycles | Clock (MHz) | $TP_v$ (Gbps) | $TP_{Loss}$ |
|---|---|---|---|---|---|---|---|---|---|
| [8] | 7/8 | 7154 | × | Virtex-5 /XC5VLX110T-1 | 3338 LUTs / 1340 FFs / 0 BRAM | 510 | 260 | 4.16 | N/A |
| [9] | 7/8 | 7154 | × | Virtex-5 /XC5VLX30T-1 | N/A LUTs / 9.2K FFs / N/A BRAM | 1020 | 164 | 1.14 | N/A |
| [10] | 1/2-5/6 | 2304 | √ 4 rates | Altera Stratix EP1S80F1508C6 | 11430 LEs / 391,1680 bits Memory | N/A | 60 | 0.12-0.36 | 2.00 |
| [11] | 1/3-3/4 | 6912 | √ 4 rates | Virtex-4 /XC4VLX25 | 614 Slices / 11 BRAMs | N/A | 278 | 1.09-3.34 | 1.62 |
| [12] | 1/3-7/8 | 511-7154 | √ 14 rates | Virtex-5 /XC5VLX110T-1 | 1658 LUTs / 1038 FFs / 0 BRAM | 512 | 335 | 0.33-4.68 | 1.71 |
| Proposed Fig. 10 | 0.313-0.822 | 511-7154 | √ 14 rates | Virtex-7 /XC7VX485T-1 | 15426 LUTs / 2520 FFs / 15 BRAMs | 51-272 | 167 | 1.67-4.39 | 1.22 |

TABLE III
HARDWARE IMPLEMENTATION OF DECODER

| Work | Rate | Length (bits) | Multi Rate | $p_d$ | FPGA technology | Resource utilization | Clock (MHz) | $TP_v$ (Gbps) | $TP_{Loss}$ |
|---|---|---|---|---|---|---|---|---|---|
| [13] | 0.868 | 9520 | × | 1 | Virtex-6 /XC6VLX240T-1 | 94978 LUTs / 19063 FFs / N/A BRAMs | N/A | N/A | N/A |
| [14] | 7/8 | 7156 | × | 1 | Virtex-4 /XC4VLX160 | 27046 LUTs / 27210 FFs / 80 BRAMs | 212.2 | 0.051-0.713 | 1.71 |
| [15] | 1/2-5/6 | 648-1944 | √ 4 rates | 1 | Virtex-4 /XC4VLX160 | 67204 LUTs / 35105 Slices | 116 | 0.612-1.808 | 1.98 |
| Fig. 5(a) | 1/3-7/8 | 511-7154 | √ 14 rates | 1 | Virtex-7 /XC7VX485T-1 | 26892 LUTs / 5736 FFs / 15 BRAMs | 100 | 0.040-0.562 | 1.71 |
| Proposed Fig. 8 | 0.313-0.822 | 511-7154 | √ 14 rates | 6 | Virtex-7 /XC7VX485T-1 | 123870 LUTs / 27134 FFs / 50 BRAMs | 100 | 1.003-2.634 | 1.22 |
| Proposed Fig. 9 | | | | | | 95563 LUTs / 17843 FFs / 50 BRAMs | | | |

The conventional multiplexer (3-lane) is shown in Fig. 8. The check node unit (CNU) is used to calculate the multiplication ($V^{(t)}$), and the variable node unit (VNU) is used to calculate the accumulation, *argmin* and flipping. The multiplexer distributes data via arbiters with the same number of CNU groups as VNU groups.

The proposed algorithm has a shorter iteration delay and less resource utilization in the multiplexer. As the multiplication is not in the iteration chain, the role of the CNU is only to calculate $V^{(1)}$ according to Eq. (5) and not to participate in subsequent iterations. A single-CNU multi-VNU TDM decoder is proposed, as shown in Fig. 9, which has a lower resource utilization than the conventional multiplexer.

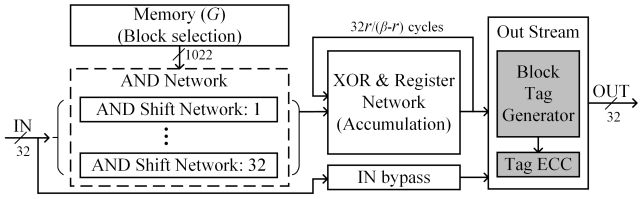



Fig. 10. The proposed LDPC encoder architecture.

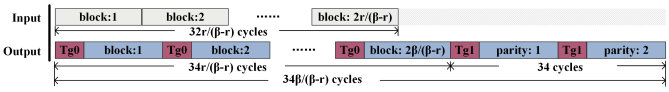



Fig. 11. The single-frame sequence diagram of proposed LDPC encoder.

## IV. IMPLEMENTATION RESULT

### A. Encoder Architecture

A variable code rate LDPC encoder architecture based on C2 code is proposed, as shown in Fig. 10, which uses inter-block serial intra-block parallel computation until the last block. The shift network can generate the non-first row vector of the quasi-cyclic matrix, reducing memory capacity. The accumulation delay between blocks is related to the code rate. The Tag Generator and Tag Error Correction Coding (ECC) produces Block Tag ($N_{tag} = 33$). Fig. 11 illustrates the single-frame sequence diagram. The parity vector is generated after block accumulation, and the Tags cost 2 cycles. The total encoding latency is $34\beta/(\beta - r)$ cycles.

Several LDPC encoders [8] [9] based on C2 code are shown in Table II, the above encoders with low resources do not change the code rate. Some multi-rate encoders with 4 rates [10] [11] have been proposed with a $TP_{Loss}$ of 2.00 and 1.62, according to Eq. (3). The miniaturized encoder with the proposed non-fixed method can transform 14 code rates through pre-coded data with a $TP_{Loss}$ of 1.71. Fig. 14(a) demonstrates that the proposed encoder has a high valid throughput at low code rates, with the lowest $TP_{Loss}$ (1.22) compared to other encoders.

### B. Decoder Architecture

A variable code rate LDPC TDM decoder architecture is presented, as shown in Fig. 12. Block Tag is detected to obtain the code length and removed by *Tag Check & Removal* module. The proposed TDM architecture has only one CNU group, the number of VNU groups and RAMs depends on the number of concurrency $p_d$. Each VNU group includes a set of registers for storing the vector $V^{(t-1)}$ of the last iteration,

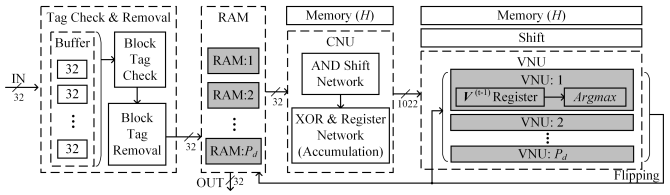



Fig. 12. The proposed LDPC decoder architecture.

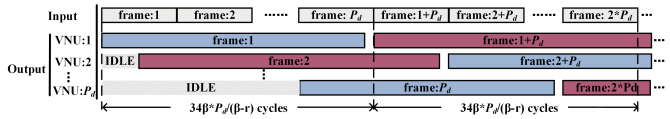
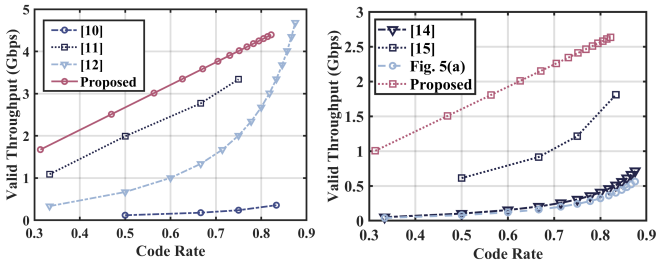



Fig. 13. The sequence diagram of proposed TDM decoder, $p_d \geq 6$.




(a) encoder (b) decoder

Fig. 14. The valid throughput of codec architectures at various code rates.

$V^{(t-1)}$ and matrix $H$ can obtain the vector $V^{(t)}$ of the current iteration according to Eq. (6a). As shown in Fig. 13, in the worst case, each VNU group completes the decoding of the current frame before the arrival of the next frame when $p_d = 6$. The total latency is independent of the delay of a single VNU group when $p_d \geq 6$. In this brief, the $p_d$ is 6.

The entire codec is tested in the AWGN channel to verify the error correction performance, as shown in Fig. 15. Binary Phase Shift Keying (BPSK) is the modulation method. When the code rate decreases, the bit error rate (BER) gradually decreases, and the codec has excellent error correction performance. When $Eb/No > 3dB$, no errors were observed for each simulation case when the code rate is 0.313.

In Table III, the hardware resources of the single-rate BF-based decoder [13] are similar to those of the proposed multi-rate decoder. The $TP_{Loss}$ of the decoder [14] is 1.71 when used the proposed non-fixed method. The decoder in [15] has 4 code rates with a $TP_{Loss}$ of 1.98.

Ablation experiments demonstrate the advantages of the proposed architecture. The $TP_{Loss}$ of decoder (Fig. 8) with the proposed non-fixed-length method is lower than that of the conventional decoder (Fig. 5(a)) with fixed code length. The proposed TDM decoder (Fig. 9) reduces 22.9% of the LUTs and 34.2% of the Flip-Flops compared to the proposed decoder (Fig. 8). Fig. 14(b) exhibits that the proposed TDM decoder has a high valid throughput at low code rates, with the lowest $TP_{Loss}$ (1.22) compared to other decoders.

## V. Conclusion

The proposed low valid throughput loss codec architecture is developed for multi-rate scenarios, with higher throughput at low code rates than conventional codecs. The non-fixed length encoding method exploits the advantages of the multiplexers. Moreover, the proposed TDM architecture has a lower resource utilization than the proposed multiplexer, which is suitable for IoT mobile edge platform.

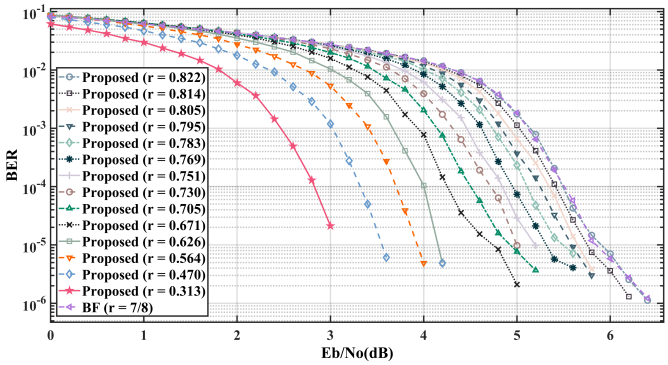



Fig. 15. Performance comparison of the proposed codec and BF algorithm, maximum iteration number is 50.

## References

[1] T. V. Nguyen *et al.*, "The Design of Rate-Compatible Protograph LDPC Codes," *IEEE Trans. Commun.*, vol. 60, no. 10, pp. 2841–2850, 2012.
[2] X. Liu, X. Wu, and C. Zhao, "Shortening for irregular QC-LDPC codes," *IEEE Commun. Lett.*, vol. 13, no. 8, pp. 612–614, 2009.
[3] J. Xie *et al.*, "High throughput multi-code ldpc encoder for ccsds standard," in *2019 IEEE 13th Int. Conf. on ASIC (ASICON)*, 2019, pp. 1–4.
[4] A. Verma and R. Shrestha, "Hardware-Efficient and High-Throughput LLRC Segregation Based Binary QC-LDPC Decoding Algorithm and Architecture," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 68, no. 8, pp. 2835–2839, 2021.
[5] A. Mahdi, N. Kanistras, and V. Paliouras, "A Multirate Fully Parallel LDPC Encoder for the IEEE 802.11n/ac/ax QC-LDPC Codes Based on Reduced Complexity XOR Trees," *IEEE Trans. Very Large Scale Integr. VLSI Syst.*, vol. 29, no. 1, pp. 51–64, 2021.
[6] Y. Sun, M. Karkooti, and J. R. Cavallaro, "VLSI Decoder Architecture for High Throughput, Variable Block-size and Multi-rate LDPC Codes," in *2007 IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2007, pp. 2104–2107.
[7] K. Zhang, X. Huang, and Z. Wang, "A high-throughput ldpc decoder architecture with rate compatibility," *IEEE Trans. Circuits Syst. I: Regul. Pap.*, vol. 58, no. 4, pp. 839–847, 2011.
[8] D. Theodoropoulos *et al.*, "Efficient Architectures for Multigigabit CCSDS LDPC Encoders," *IEEE Trans. Very Large Scale Integr. VLSI Syst.*, vol. 28, no. 5, pp. 1118–1127, 2020.
[9] Small World Communications, "LCE01C CCSDS (8160,7136) LDPC Encoder, 1st ed." [EB/OL], http://www.sworld.com.au/products/lce01c.html Accessed Mar 28, 2013.
[10] H. Yasotharan *et al.*, "A flexible hardware encoder for systematic low-density parity-check codes," in *2009 52nd IEEE Int. Midwest Symp. Circuits Syst.*, 2009, pp. 54–57.
[11] C. P. Fewer, M. F. Flanagan, and A. D. Fagan, "A Versatile Variable Rate LDPC Codec Architecture," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 54, no. 10, pp. 2240–2251, 2007.
[12] J. Liu and Q. Feng, "A Miniaturized LDPC Encoder: Two-Layer Architecture for CCSDS Near-Earth Standard," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 68, no. 7, pp. 2384–2388, 2021.
[13] F. Ghaffari and B. Vasic, "Probabilistic gradient descent bit-flipping decoders for flash memory channels," in *2018 IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2018, pp. 1–5.
[14] X. Chen *et al.*, "Memory System Optimization for FPGA-Based Implementation of Quasi-Cyclic LDPC Codes Decoders," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 58, no. 1, pp. 98–111, 2011.
[15] S. Kumawat *et al.*, "High-Throughput LDPC-Decoder Architecture Using Efficient Comparison Techniques Dynamic Multi-Frame Processing Schedule," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 62, no. 5, pp. 1421–1430, 2015.