IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 2 February 2024, accepted 21 February 2024, date of publication 27 February 2024, date of current version 7 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3371188

RESEARCH ARTICLE

Universal High-Throughput and Low-Complexity LDPC Decoder for Laser Communications

JING KANG, JUNSHE AN, AND YAN ZHU
Key Laboratory of Electronics and Information Technology for Space Systems, National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China

Corresponding author: Jing Kang (k_naive@163.com)

This work was supported by the Space Science Leading Satellite Project under Grant XDA15320100.

ABSTRACT To address the challenges posed by propagation channel impairments and to meet the high data rate requirements of laser communications, this study introduces a pioneering low-density parity-check (LDPC) decoder characterized by its high throughput and low complexity. The unique design of this decoder, based on an inter-frame pipeline and intra-frame parallel (IFPP-IFP) scheme, is specifically tailored to maximize the efficiency of processing units, leading to a substantial increase in decoding throughput. The implementation of IFPP is realized through a novel full-overlap message passing (FOMP) scheme and a dynamic address access (DAA) algorithm, distinguishing it from current solutions. Additionally, the decoder employs a message packing strategy and low-complexity data alignment units to effectively achieve IFP. Compared to existing solutions, our hardware implementation on the Xilinx XCKU060 FPGA demonstrates significant progress. The decoder achieves a decoding throughput of 2.67 Gb/s at 10 iterations and 350MHz. Remarkably, when five decoders are used on a single FPGA device, the throughput soars to 13.3 Gb/s, outperforming state-of-the-art designs by 1.3 times and concurrently reducing resource consumption by half. This combination of resource efficiency and enhanced throughput highlights the innovative and superior nature of our proposed approach.

INDEX TERMS LDPC decoder, laser communications, high-throughput, low-complexity, FPGA.

I. INTRODUCTION

As space exploration advances, the need for transmitting extensive scientific data has become urgent. Traditional radio frequency (RF) communications are nearing their bandwidth limit, driving the demand for laser communications, which offer higher bandwidth (up to multi-gigabit per second), reduced weight, enhanced flexibility, and improved security. However, the propagation channel of laser communications is less than ideal due to environmental factors such as free space loss, atmospheric attenuation, and turbulence of the refraction index, resulting in channel fading [1], [2]. Therefore, channel coding is essential to protect the transmitted signal against the effects of propagation channel impairments [3]. Furthermore, achieving high data rates in laser communications requires a dedicated physical layer design. However, conventional channel coding tends to consume significant computational and storage resources, presenting a bottleneck for achieving ultra-high throughput. Therefore, there is a pressing demand for innovative, high-speed, and low-complexity codecs, in particular for the on-board side in which embedded constraints compatible with a space environment are very demanding.

Low-density parity-check (LDPC) codes proposed by Gallager [4] are widely used for their performance is close to the Shannon limit. Many communication standards chose LDPC as the channel coding codes, including DVB-S2 [5], CCSDS [6], and 5G [7], etc. The efficiency of couples of LDPC codes and decoders has already been recently proven for this application field [8], [9], [10], [11]. A new forward error correction (FEC) technology has been proposed to the CCSDS optical working group (SLS-OPT) as a competitive

The associate editor coordinating the review of this manuscript and approving it for publication was Luca Barletta.

alternative to current standardized solutions for RF telemetry downlinks such as DVB-S2, SCCC, and ARJA/C2 LDPC codes [12].

QC-LDPC [13] are a sub-class of structured codes that should achieve performances close to unstructured LDPC codes [14] while being much more convenient for hardware implementations. LDPC codes usually use min-sum algorithm (MSA) [15] for iterative message passing (MP) decoding, which utilizes an approximate method to perform check node computation and thus has a lower complexity. To improve the decoding performance, attenuated MSA (AMSA) and offset MSA (OMSA) was proposed in [16].

A multitude of decoder architectures for QC-LDPC codes has been proposed in the literature. For current satellite communications, Jet Propulsion Laboratory designed a long frame DVB-S2 decoder [17] that fills 5 FPGA, achieving a data rate of 1.25 Gb/s. Our previous work proposed an enhanced partially-parallel LDPC decoder, which achieves a decoding throughput of 1.02 Gb/s [18]. In [19], a decoder based on the storage addressing scheme suitable for high parallelism decoders is presented, with a throughput of 1.1Gb/s. In [20], compact memory strategies for partially parallel QC-LDPC decoder architecture are proposed and achieve a throughput up to 2 Gb/s. The decoder in [21] is implemented on the Xilinx XCVU9P FPGA and achieves a throughput of 2.65 Gb/s.

However, new laser communications in space expect higher throughput and there comes the need of innovative methods to reach more than 10 Gb/s [22]. Very promising results are presented in [22] in which it is shown that a 10.6 Gb/s decoder can be implemented based on an Application Specific Instruction Set (ASIP) architecture. However, this high hardware efficiency comes at the cost of reduced programmability. In [12], a decoder IP was developed with a data rate of 10 Gb/s, but it consumes a significant amount of resources, making it less suitable for resource-constrained devices.

In this paper, we propose an universal, high-throughput, low-complexity LDPC decoder for future laser communications. The decoder is designed based on an inter-frame pipeline and intra-frame parallel (IFPP-IFP) scheme and addresses the limitations of previous works. Our contributions are as follows:

1) A full-overlap message passing (FOMP) scheme and a dynamic address access (DAA) algorithm are proposed to maximize the utilization of processing units, thereby achieving inter-frame pipeline (IFPP).
2) To further enhance throughput, an effective message packing and low-complexity data alignment units are employed to achieve intra-frame parallel (IFP).
3) A detailed FPGA implementation of the proposed decoder on Xilinx FPGA is showed, and the performance of various designs are analytically estimated and evaluated.

The remainder of the paper is organized as follows. Section II summarizes the notations, AMSA for iterative MP decoding, and the typical partially-parallel decoder architecture. Section III makes a detailed description of the proposed LDPC decoder. The implementation results are reported in section IV. Section V concludes the paper.

II. BACKGROUND

A. NOTATIONS

QC-LDPC codes are a class of LDPC codes obtained by expanding the base matrix B by an expansion factor L [3]. The LDPC code has a length of $N = L \times n$, with n denoting the number of columns of B, and an information bit length $K = N - M$, where $M = L \times m$, with m representing the number of rows in B. The parity check matrix H can be obtained in the following way: every element of B is replaced by a square matrix that is either the zero matrix for -1 entries of B, or the identity shifted matrix by the non-negative entry value of B, with $b_{i,j}$ is the cyclic shift value. Each column of H has constant weight γ (the number of 1-entries in a column) and each row has constant weight ρ (the number of 1-entries in a row). An equivalent view of the H matrix is the graph representation referred to as the Tanner graph. It is a bipartite graph, with two types of nodes, interconnected by edges. Lines from H correspond to check nodes (CN), and columns in H correspond to variable nodes (VN). The decoding process is an iterative process. Each iteration requires two types of processing performed on the messages VN and CN. The messages exchanged between the two types of units are quantized on a few bits, where Q is defined as the number of quantization bits.

B. THE AMSA FOR ITERATIVE MP DECODING

The AMSA is an iterative decoding algorithm that allows updating any check node or variable node in parallel and contains two phases. Let $m_{c_i \to v_j}$ (C2V) represent the extrinsic message from check node c_i to variable node v_j and $L_{v_j \to c_i}$ (V2C) represent the extrinsic message from the variable node v_j to check node c_i. These two messages are computed and exchanged along the edges in the corresponding Tanner graph. $N(c_i) \backslash v_j$ denote the neighboring variable nodes of c_i except for v_j and $N(v_j) \backslash c_i$ denote the neighboring check nodes of v_j except for c_i. α is the modification factor. C_{v_j} denote the intrinsic log-likelihood ratio (LLR) from the channel of variable node v_j. u_j represent the decoded bits. The decoding process is as follows:

1) Initializing: $L_{v_j \to c_i} = C_{v_j}, m_{c_i \to v_j} = 0$.
2) Iteration:

$$m_{c_i \to v_j} = \alpha \cdot \min_{v_b \in N(c_i) \backslash v_j} \left(\left| L_{v_b \to c_i} \right| \right) \times \prod_{v_b \in N(c_i) \backslash v_j} sign(L_{v_b \to c_i}). \quad (1)$$

$$L_{v_j \to c_i} = \sum\nolimits_{c_a \in N(v_j) \backslash c_i} m_{c_a \to v_j} + C_{v_j}. \quad (2)$$

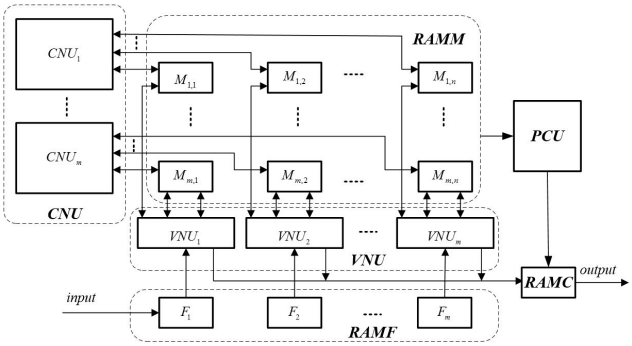


FIGURE 1. Typical partially-parallel decoder architecture.

3) Hard Decision:

$$u_j = \begin{cases} 0, \textit{if } L_{v_j} = \sum_{c_a \in N(v_j)} m_{c_a \to v_j} + C_{v_j} > 0; \\ 1, \textit{otherwise}. \end{cases} \tag{3}$$

If the decoded bits satisfy all the parity-check equations or a maximum number of iterations is reached, the decoding is finished. Otherwise, go back to 2).

C. TYPICAL PARTIALLY-PARALLEL DECODER ARCHITECTURE

A typical partially-parallel decoder architecture, as depicted in Fig.1, primarily comprises m check node update (CNUs), n variable node update (VNUs), m parity-check equation calculation module (PCUs), and three memories RAMF, RAMM, RAMC, which are used to store received LLRs, extrinsic messages, and hard decisions, respectively. The parallel degree is m or n. In other words, m rows (n columns) from different block rows (columns) are updated simultaneously in the C2V (V2C) message update phase. All rows (columns) of each block are updated serially in L clock cycles. The extrinsic messages in RAMM is managed using a static addressing approach, where read and write addresses for each node remain fixed. In the C2V message update phase, the RAMM access address is $b_{i,j}, b_{i,j}+1, \cdots 0, 1, \cdots b_{i,j}-1$. In the V2C message update phase, the memory access address is $0, 1, \cdots, L-1, L$.

III. THE UNIVERSAL, HIGH-THROUGHPUT, LOW-COMPLEXITY LDPC DECODER

In the typical decoding process, VNU and CNU operate alternately, with each of them being idle for half of the time. This leads to a reduced utilization efficiency of FPGA hardware resources. To overcome this limitation, this paper proposed a FOMP scheme and a DAA algorithm, enabling simultaneous storage and access of two frames within a single block of RAM.

To further enhance throughput, multiple messages are packed into the same memory word. We define the number of messages packed into one memory word as P. Potentially, the throughput can be P times that of a typical decoder, given that there are P times more functional units operating simultaneously. However, without a proper message packing scheme, memory access conflicts can arise, as multiple messages are accessed per cycle. In addition, efficient data alignment units are required to minimize the additional logic overhead. The techniques to address these challenges are described next.

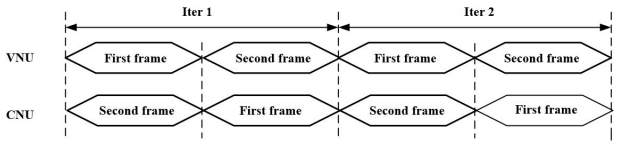


FIGURE 2. FOMP scheme.

A. FOMP SCHEME AND DAA METHOD

In a related study [19], an overlapped-NMSA decoder architecture was proposed, allowing VNU and CNU to work concurrently. Nevertheless, in FPGA simple dual-port RAM mode, each RAM port can only read or write data at a single address at a time. With the conventional static addressing approach, the extrinsic message read and write operation for VNU and CNU requires using separate ports. As a result, the RAMM resource requirements for this decoder are twice that of the traditional approach, leading to increased resource utilization demands. We address this challenge with a combination of three techniques, enabling simultaneous storage and access of two frames within a single block of RAM.

First, FOMP is produced, enabling the full parallel operation of VNU and CNU. As depicted in Fig.2, during each iteration, we initially update V2C message for the first-frame data and C2V message for the second-frame data. Subsequently, we update C2V message for the first-frame data and V2C message for the second-frame data. This process is based on the following initialization strategy: We initialize the first-frame extrinsic messages in RAMM to all zeros, while the second-frame extrinsic messages are initialized with channel-received LLRs.

Second, during each iteration, we employ DAA algorithm for reading and writing extrinsic messages in RAMM. We utilize l_v, l_h to denote the pipeline stages for VNU and CNU respectively. Fig.3 illustrates an example of the DAA algorithm. For a variable node in the first frame, suppose its initial storage address for extrinsic messages is d_v. The address access process for its extrinsic message iterative update proceeds as follows: During the first VNU, the read and write addresses for this extrinsic message are designated as d_v and $d_v + l_v$, respectively. Subsequently, in the first CNU, the read and write addresses become $d_v + l_v$ and $d_v + l_v + l_h$, respectively. Finally, in the second VNU, the read and write addresses transition to $d_v + l_v + l_h$ and $d_v + 2 \times l_v + l_h$. In subsequent iterations, the write address for

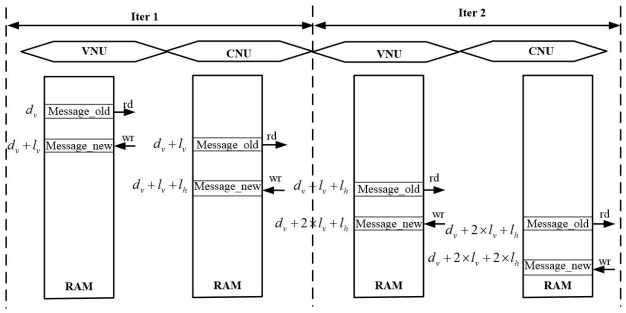


FIGURE 3. An example of the DAA algorithm.

each CNU is calculated by adding l_h to its read address and applying modulo L, while the write address for each VNU is determined by adding l_v to its read address and similarly taking the modulo L.

Algorithm 1 Dynamic Address Access

Initialization: $Addr_i^{(0)} = i$; $i = 0, 1, \cdots, L-1$ for the first frame; $i = L, L+1, \cdots, 2L-1$ for the second frame;
for *i 1* **to** I_{max} *-1* **do**
  V2C message update phase:;
  for *j 0* **to** $2L-1$ **do**
    $Addr_j^{(i)new} = \text{mod}(Addr_j^{(i-1)old} + l_v, L)$;
  end
  C2V message update phase:;
  for *j 0* **to** $2L-1$ **do**
    $Addr_j^{(i)new} = \text{mod}(Addr_j^{(i-1)old} + l_h, L)$;
  end
end

Third, RAMM operates in a true dual-port mode, featuring two read-write ports configured in a read-first mode, which guarantees no collisions (read will access prior memory contents safely). Each of these read-write ports is connected to both VNU and CNU and is responsible for reading and writing extrinsic messages for one frame of decoded data.

In light of the details mentioned above, DAA ensures that the address for C2V (V2C) message that needs to be read matches the address for the V2C (C2V) message that has just been updated. Furthermore, by configuring the RAM in read-first mode, it becomes possible to concurrently perform both the readout and write-in processes for two V2C or C2V messages. This approach enables the simultaneous storage and access of two frames within a single RAM block. In comparison to conventional design methodologies, the adoption of this technique leads to a twofold increase in the utilization of BRAM and computational resources within the decoder. Additionally, employing the method described above, we have derived a general formula for calculating the throughput, as shown in equation (4). The throughput for a conventional design is represented by equation (5).

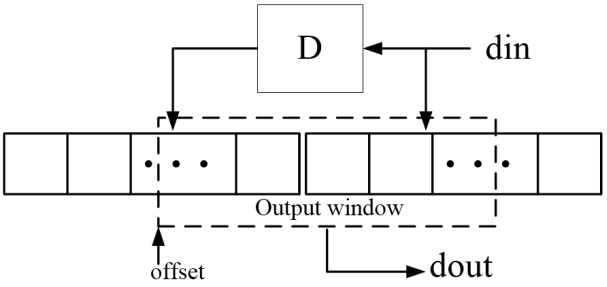


FIGURE 4. The architecture of data aligner.

$$T_{pro} = \frac{2 \times K \times f_{clk}}{2 \times max(\lceil \frac{L}{P} \rceil + l_v, \lceil \frac{L}{P} \rceil + l_h) \times I_{max}} \tag{4}$$

$$T_{conv} = \frac{K \times f_{clk}}{(\lceil \frac{L}{P} \rceil + l_v + \lceil \frac{L}{P} \rceil + l_h) \times I_{max}} \tag{5}$$

where f_{clk} denotes the clock frequency, and I_{max} represents the maximum number of iterations. According to equations (4) and (5), we arrive at equation (6).

$$\frac{T_{pro}}{T_{conv}} = \begin{cases} 2 + \frac{l_h - l_v}{\lceil \frac{L}{P} \rceil + l_v}, \textit{if } l_v > l_h; \\ 2 + \frac{l_v - l_h}{\lceil \frac{L}{P} \rceil + l_h}, \textit{otherwise}. \end{cases} \tag{6}$$

Given that $|l_h - l_v|$is significantly smaller than $\lceil \frac{L}{P} \rceil$, the ratio $\frac{T_{pro}}{T_{conv}}$ is approximately 2, thereby effectively doubling the throughput of the decoder.

B. MASSAGE PACKING AND DATA ALIGNMENT UNIT

To further enhance throughput, multiple messages are packed into the same memory word of RAMM and processed concurrently. Memory conflicts could arise if the CNU and VNU try to access the same location simultaneously. In [23], a solution involving a switching network and two data buffers was implemented to select the target data group. The switching network comprises a large number of multiplexers, which consume considerable hardware resources, and the design of its output logic is particularly challenging. To simplify the multiplexers and output logic, and shorten the critical path to reduce the decoding delay, we designed an optimized low-complexity data aligner as shown in Fig.4. The offset of each RAMM is precomputed according to $mod(b_{i,j}, P)$. To achieve data alignment, data read out from RAMMs (or data updated by processing units) is spliced with data read out from RAMM (or data updated) of the previous clock, and the location of the output window is determined by the offset value to obtain adjacent P messages. Subsequently, the adjacent P messages can be forwarded to the processing units (or written back to the RAMM). Overall, our implementation requires two Q-bit registers and the extra latency is one clock, which greatly simplifies the alignment task.

As the memories typically constitute a significant portion of the overall hardware in an LDPC decoder, and it has

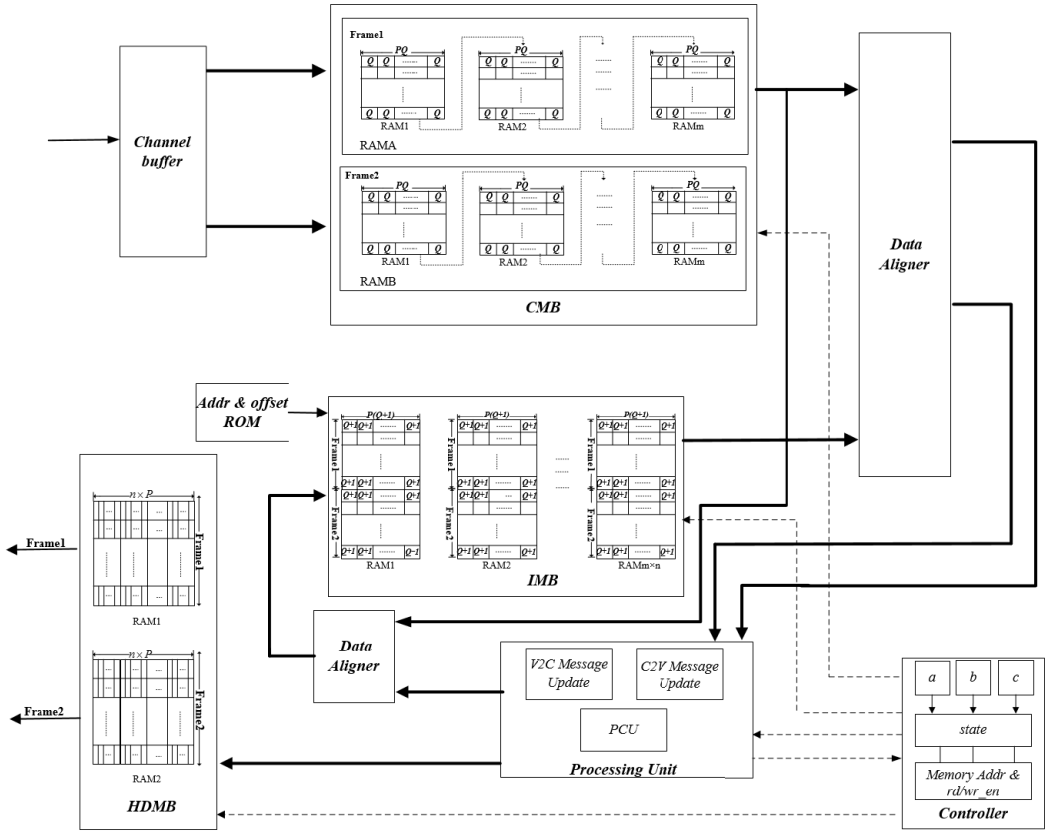


FIGURE 5. The architecture of the proposed decoder.

been observed that the required extrinsic messages and hard decisions for CNUs and PCUs can be simultaneously generated by VNUs. Notably, there is no data processing overlap between CNUs and PCUs, enabling them to operate in parallel. In order to further reduce memory usage, we pack hard decisions and extrinsic messages. In this strategy, during the VNU phase, the 1-bit hard decision (as the most significant bit), combined with the Q-bit extrinsic message, is written back into RAMM. This approach results in the repeated storage of hard decisions $P \times m$ times. However, it enables the decoder to concurrently read $P \times m$ copies of hard decisions from different offsets, allowing the PCUs to compute parity-check equations with the same level of parallelism as the CNUs. Furthermore, both PCUs and CNUs can share the same set of address information, eliminating the need for an extra address generation unit for PCUs.

Nonetheless, the previously mentioned strategy for packing hard decisions and extrinsic messages presents a challenge. When the check equations are satisfied or a preset maximum iteration number is reached, the hard decisions represent the final decoded results of the last input frame, which will subsequently be outputted. It is worth noting that two ports of RAMM are consistently occupied by the input and output of CNUs (PCUs) or VNUs. If the final output (hard decision) is retrieved from RAMM, it may lead to

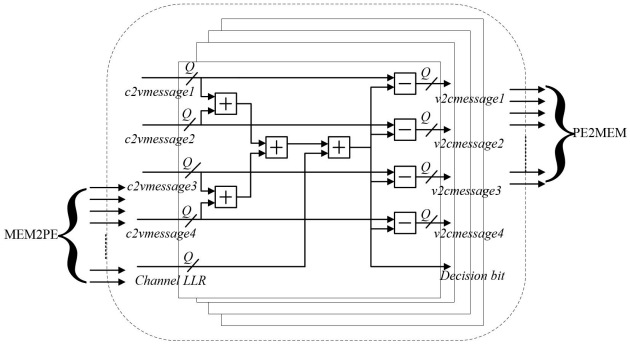


FIGURE 6. The architecture of V2C update module.

memory access conflicts. To address this issue, during the VNU phase, $P \times n$ hard decisions are simultaneously written to the RAMC memory banks while the outputs are being written to RAMM. Benefitting from this strategy, we reduce the number of RAMC from n to 1.

C. OVERALL DECODER ARCHITECTURE

To validate the proposed techniques, hardware decoder implementations are designed. The primary strengths of this decoder architecture lie in its universal, low hardware

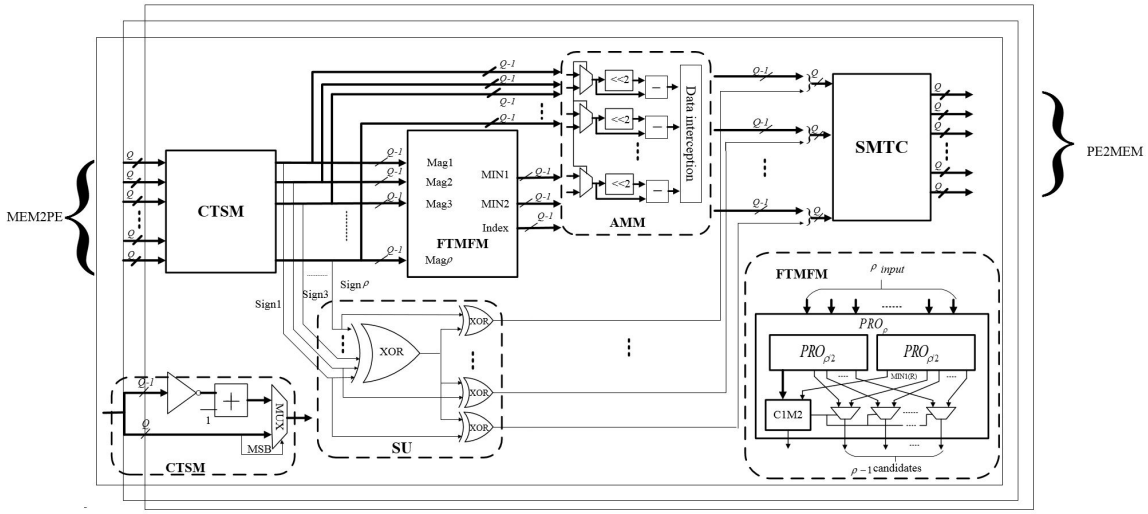


FIGURE 7. The architecture of C2V update module.

complexity, and high throughput. This architecture is applicable to all QC-LDPC codes and is particularly well-suited for resource-constrained devices. As depicted in Fig.5, the main components consist of processing units (VNU, CNU, PCU), various memory blocks (CMB, IMB, HDMB), data aligners, and a controller.

The processing units contain three modules to perform the crucial process of decoding. The V2C message update module (VNU) is responsible for performing equation (2), and comprises $n \times P$ individual variable node processing unit, which has $\gamma + 1$ inputs and outputs. γ is the column weight, and the additional input and output are used for channel-received LLR and the hard decision bit. The cascaded adders of the V2C message update module are shown in Fig.6. The C2V message update module (CNU) is composed of $m \times P$ individual check node processing units to perform the relevant operations according to equation (1), and its architecture is shown in Fig.7. Initially, the input LLRs in 2's complement format are converted to sign-magnitude (SM) format by using 2's-complement to sign-magnitude (CTSM) modules. Subsequently, the most significant bits (MSBs), representing signs, of the SM-LLRs are sent to the sign unit (SU), and the magnitudes of the SM-LLRs are processed by the first two minima find module (FTMFM) and attenuated modification module (AMM). The FTMFM employs an area-efficient recursive tree architecture to find the first two minima, along with the index of the first minimum [24], This approach reduces the number of comparators by reusing the intermediate comparison results computed for the first minimum to collect the candidates of the second minimum. The AMM is designed for multiplying the modification factor $\alpha = 0.75$ to the first two minima through left-shift, subtraction, and interception operation. In SU, all input MSBs are XORed and the resulting product bit is further XORed with every input MSBs to generate the sign bits of the updated SM-LLRs. Finally, the updated SM LLRs are converted back to 2's complement format using sign-magnitude to 2's complement (SMTC) module. The parity-check equation calculation unit (PCU) is responsible for determining whether the decoded hard decision is a valid codeword.

The channel memory bank (CMB) is a stack of $2 \times n$ simple dual-port RAMs responsible for storing the channel-received LLRs of two frames. Each RAM has a width of $P \times Q$ and a depth of L/P. The intermediate memory bank (IMB) is used to store extrinsic messages exchanged between variable nodes and check nodes during the decoding process. The RAMs within IMB operate as true dual-port RAMs with a read-first mode. Each of these read-write ports is connected to both VNU and CNU and is responsible for reading and writing extrinsic messages for one frame of decoded data. The width and depth of the RAM are $P \times (Q + 1)$ for P hard decisions and P extrinsic messages packing and $2 \times L/P$ for two-frame message storage. The hard decision memory bank (HDMB) stores the hard decisions of the decoder. It consists of a stack of 2 hard decision RAM units for two frames, with a depth of L/P and a word width of $n \times P$ bits.

The controller controls the entire decoding process. The decoding process is as follows:

1) Initialization: At the decoder's input side, Q bits of quantized channel LLRs are received and passed through the channel buffer. Subsequently, the LLRs for two frames are segmented and stored in the RAMs located within the CMB based on the columns of the parity-check matrix H. RAMA stores the first-frame data, while RAMB stores the second-frame data and the address range for all these RAMs is from 0 to $L/P - 1$. It is noteworthy that the extrinsic messages for the first frame in the CMB are initialized to all zeros, while the extrinsic messages for the second frame are initialized with the channel-received LLRs.

TABLE 1. Implementation results of the proposed decoder.

LDPC Code	Quantization	Iterations	Hardware Comlexity			One Decoder Performance		Max throughput on Kitnex UltraScale XCKU060
			LUTs	FFs	BRAMs	Throughput	Latency	
(8176,7154)	6	10	48353	49768	98	2.67Gb/s	1875 clocks	13.3Gb/s (5decoders)

TABLE 2. Comparison with other state-of-the-art designs.

Work	[12]	[18]	[19]	[21]	[22]	This paper
Processor	Virtex UltraScale+ XCVU9P	Virtex-5 XC5VLX330	Zynq 7000	Virtex UltraScale+ XCVU9P	Zynq XCZU9eg	Kitnex UltraScale XCKU060
Code length	30720	8176	8176	8176	16384	8176
Code rate	1/2	7/8	7/8	7/8	3/4	7/8
Algorithm	OMSA	AMSA	AMSA	NMSA	OMSA	AMSA
Iterations	25	10	15	10	10	10
f_{clk}(MHz)	250	250	210	253	250	350
LUTs	499986	46294	38499	194479	49121	48353/241760
FFs	460389	39103	50673	252306	26903	49768/248840
BRAMs	55.6Mb	88	144	/	50	98/490
Throughput/Gbps	10	1.02	1.1	2.65	2.13	2.67/13.3

Additionally, the iteration count is set to 0, with a maximum iteration count defined as I_{max}.

2) Updating variable nodes for the first-frame data and check nodes for the second-frame data: In VNU, extrinsic messages are sequentially updated in column order, and hard decisions are made. Simultaneously, in CNU, extrinsic messages are sequentially updated in row order. DAA for reading and writing extrinsic messages in CMB is employed (described in the previous section). Moreover, during the process, data read out from CMB (or data updated by processing units) is aligned by the data aligner and then can be forwarded to the processing units (or written back to the IMB and HDMB). It's noteworthy that PCU proceeds concurrently with CNU, completing the calculation of parity-check equations.
3) Updating check nodes for the first-frame data and variable nodes for the second-frame data: the process details are the same as step 2).
4) Termination decision: If the number of iterations reaches the maximum iteration count I_{max} or the syndromes are zero, the iterations terminate, and proceed to step 5). Otherwise, go back to step 2) to continue with the decoding iterations.
5) Decoding output: Read the hard decisions of two decoded frames separately from RAMA and RAMB in HDMB.

IV. FPGA IMPLEMENTATION AND ANALYSIS

The hardware architecture is dedicated on CCSDS C2 LDPC code, and synthesized and implemented on Xilinx XCKU060 FPGA. We use a factor $\alpha = 0.75$ and $Q = 6$ bits to quantize the LLR message, including 1-bit sign, 3-bit integer, and 2-bit fraction. Fig.8 shows the bit error rate (BER) test results for 10 maximum iterations from an FPGA implementation. Note that the difference between simulations and hardware tests is 0.1 dB or less. And we pack $P = 7$ neighboring rows (columns) in a memory address. The decoding throughput can be calculated as follows:

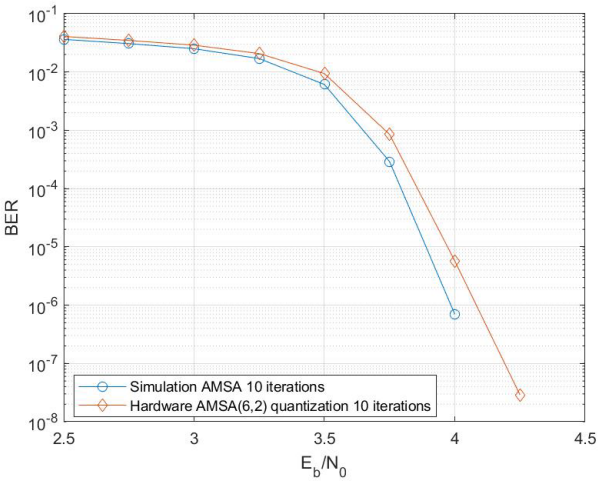


FIGURE 8. BER test results.

$$T = \frac{K \times f_{clk}}{t_{init} + (t_{c2v} + t_{v2c}) \times I_{max}} \tag{7}$$

where f_{clk} is the system clock frequency, K is the information bit length and I_{max} is the iteration number. $t_{init} = 75$, $t_{c2v} = t_{v2c} = 90$ is the clock cycles needed for initialization, CNU, and VNU, respectively. Table 1 presents the implementation results of the proposed decoder. Table 2 shows the comparison of the state-of-the-art designs.

As shown in Table 2, our proposed decoder occupies slice look-up-tables (LUTs) at 48353, flip-flops (FFs) at 49768, block RAMs (BRAMs) at 98 and achieves a decoding throughput of 2.67Gb/s at 10 iterations at a maximum clock frequency 350MHz. Throughput values are achieved by applying a parallelism level of 2 frames in a decoder. To achieve the maximum throughput on the FPGA, several decoders are allocated to exploit available resources. Promising results show that 13.3Gb/s is achieved with 5 decoders on one FPGA device, with less than 75% resources. Decoders proposed in [12] and [21] achieve high throughput at the cost of high resource consumption. The throughput of our proposed decoder, when utilizing a combination of five decoders, surpasses that of the decoder in [12] by 1.3 times, while also achieving a 50% reduction in resource utilization. In comparison to the designs presented in [18] and [19], our architecture, even with a single decoder, achieves a throughput of more than two times while maintaining a similar level of resource consumption. In [22], the proposed decoder is based on an ASIP architecture, which leads to reduced hardware complexity and enhanced decoding throughput. However, it's noteworthy that the decoder in [22] consumes a significant amount of LUTs, while FF resources are relatively low. In the realm of hardware design, a well-balanced allocation of resources contributes to achieving shorter critical path delays, and this point is further validated by the comparison of implementation results. In [22], the critical path limits the FPGA frequency up to 260 MHz. In contrast, the utilization of both LUTs and FFs resources of our proposed decoder is more balanced, resulting in a shorter critical path delay and the potential for higher decoding throughput. Comparisons demonstrate that the proposed decoder achieves high throughput, while maintaining low hardware complexity.

V. CONCLUSION

In this paper, we have introduced a novel LDPC decoder for future laser communications, supported by the innovative IFPP-IFP framework. This decoder leverages the FOMP scheme and DAA algorithm to maximize the utilization of processing units, resulting in efficient IFPP. Furthermore, we have incorporated effective message-packing techniques and utilized low-complexity data alignment units to achieve IFP. The decoder's key strengths lie in its universality, high throughput, and low hardware complexity, as exemplified by its remarkable ability to achieve a throughput of 13.3 Gb/s using only five decoders on a single FPGA device. This achievement signifies a significant advancement, effectively halving resource utilization compared to existing state-of-the-art designs. Furthermore, our proposed algorithm and architecture can adapt to different QC-LDPC code families, but customized adjustments based on the unique characteristics of relevant matrices may introduce complexity and potential compatibility challenges in practical implementation. Additionally, in practical laser communication environments, communication channels have uncertainties, and optical signals may encounter various interferences like atmospheric disturbances and scattering, potentially affecting decoding performance. Exploring adaptive decoding strategies is a promising avenue for further advancing laser communication technology.

REFERENCES

[1] P.-D. Arapoglou and N. Mazzali, "System-level benefit of variable data rate in optical LEO direct-to-Earth links," in *Proc. Int. Conf. Space Opt.*, Jun. 2021, p. 1876.

[2] A. A. Youssef, M. Abaza, and A. S. Alatawi, "LDPC decoding techniques for free-space optical communications," *IEEE Access*, vol. 9, pp. 133510–133519, 2021.

[3] A. Dixit and V. K. Jain, "SNR-and rate-optimized LDPC codes for free-space optical channels," *IEEE Access*, vol. 9, pp. 13212–13223, 2021.

[4] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[5] *Second Generation Framing Structure, Channel Coding and Modulation Systems for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Applications*, document ETSI EN 302 307, Nov. 2014.

[6] *CCSDS 131.0-B-2 TM Synchronization and Channel Coding*, CCSDS, Washington, DC, USA, 2011.

[7] 3GPPTSGRANWG1, *Multiplexingandchannelcoding(Release15)*, Standard TS 38.212, V15.1.1, France, 2018.

[8] *CCSDS 142.0-B-1 Optical Communications Coding and Synchronization*, CCSDS, Washington, DC, USA, 2019.

[9] V. Pignoly, B. Le Gal, C. Jego, B. Gadat, and L. Barthe, "High speed LDPC decoding for optical space link," in *Proc. 27th IEEE Int. Conf. Electron. Circuits Syst. (ICECS)*, Nov. 2020, pp. 1–4.

[10] S. Poulenard, B. Gadat, J. F. Chouteau, T. Anfray, C. Poulliat, C. Jego, O. Hartmann, G. Artaud, and H. Meric, "Forward error correcting code for high data rate LEO satellite optical downlinks," in *Proc. Int. Conf. Space Opt.*, Jul. 2019, pp. 2029–2038.

[11] J. Lochard, N. de Guembecker, P. Chéoux-Damas, X. Calmet, E. Giraud, A. Jullien, M. Ghezal, L. Barthe, J.-F. Chouteau, L. Coret, G. Artaud, L. Perret, P. Viallefont, and E. Cazala-Hourcade, "LASIN optical link on-board CO3D constellation," in *Proc. Int. Conf. Space Opt.*, Jul. 2023, pp. 375–383.

[12] G. Artaud, J.-F. Chouteau, L. Barthe, B. Gadat, T. Anfray, S. Poulenard, A. D. Thomas, and A. Quentel, "Design and validation of a new coding and synchronization layer for space optical communications," in *Proc. Int. Conf. Space Opt.*, Jul. 2023, pp. 2055–2068.

[13] T. M. N. Ngatched and F. Takawira, "Improved generalized low-density parity-check codes using irregular graphs," *Trans. South Afr. Inst. Electr. Eng.*, vol. 94, no. 4, pp. 43–49, Dec. 2003.

[14] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[15] M. P. C. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Trans. Commun.*, vol. 47, no. 5, pp. 673–680, May 1999.

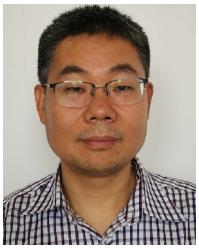
[16] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X.-Y. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Trans. Commun.*, vol. 53, no. 8, pp. 1288–1299, Aug. 2005.

[17] K. Andrews, "Implementation of a DVB-S2 encoder and decoder for the LCRD mission," in *Proc. CCSDS Spring Meeting*, 2013. [Online]. Available: https://docplayer.net/235340396-Implementation-of-a-dvb-s2-encoder-and-decoder-for-the-lcrd-mission.html

[18] J. Kang, B. Wang, Y. Zhang, and J. An, "Enhanced partially-parallel LDPC decoder for near Earth applications," in *Proc. IEEE 11th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC) IEEE 11th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Beijing, China, Jun. 2021, pp. 12–16.

[19] H. Bai, K. Zhai, G. Liu, and J. Zou, "A design method of high parallelism QC-LDPC decoder based on FPGA," in *Proc. 5th Int. Conf. Inf. Commun. Signal Process. (ICICSP)*, Shenzhen, China, Nov. 2022, pp. 441–447.
[20] T. Xie, B. Li, M. Yang, and Z. Yan, "Memory compact high-speed QC-LDPC decoder," in *Proc. IEEE Int. Conf. Signal Process. Commun. Comput. (ICSPCC)*, Oct. 2017, pp. 1–5.
[21] K. M. S. Nampoothiri, K. N. Menon, S. Muhsin, S. Manoj, P. Sooraj, K. J. Dhanaraj, and P. P. Deepthi, "A high throughput QC-LDPC decoder architecture for near-Earth satellite communication," in *Proc. 4th Int. Conf. Circuits, Syst. Simul. (ICCSS)*, Kuala Lumpur, Malaysia, May 2021, pp. 23–27.
[22] V. Pignoly, B. Le Gal, C. Jego, and B. Gadat, "High data rate and flexible hardware QC-LDPC decoder for satellite optical communications," in *Proc. IEEE 10th Int. Symp. Turbo Codes Iterative Inf. Process. (ISTC)*, Dec. 2018, pp. 1–5.
[23] 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, no. 1, pp. 104–114, Jan. 2007.
[24] Y. Lee, B. Kim, J. Jung, and I.-C. Park, "Low-complexity tree architecture for finding the first two minima," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 1, pp. 61–64, Jan. 2015.

JING KANG received the Ph.D. degree in technology of computer application from the University of Chinese Academy of Sciences, Beijing, China, 2021. After graduation, she joined the National Space Science Center, Chinese Academy of Sciences, where she is currently an Assistant Research Fellow with the Key Laboratory of Electronics and Information Technology for Space Systems. Her primary research interests include channel coding, satellite communications, laser communications, 5G, and FPGA implementation.



JUNSHE AN received the Ph.D. degree in computer software and theory from Northwestern Polytechnical University, Xi'an, China, in 2004. In 1995, he joined the National Space Science Center, Chinese Academy of Sciences, where he is currently a Professor with the Key Laboratory of Electronics and Information Technology for Space Systems. His current research interests include space integrated electronic technology, space computer hardware and software, system architecture, and data processing technology.

YAN ZHU received the Ph.D. degree in technology of computer application from the University of Chinese Academy of Sciences, Beijing, China, 2006. In 1997, he joined the National Space Science Center, Chinese Academy of Sciences, where he is currently a Professor with the Key Laboratory of Electronics and Information Technology for Space Systems. His main research interests include space-integrated electronic technology, data storage, processing, and transmission technology.