

A FPGA based Satellite Onboard High Speed Decoding Architecture for Internet of Vehicle

KaiShi Wang^{1,2}, Cheng Wang^{1,2}, Songsong Cai^{1,2}, Xi Gong^{1,2} and Weidong Wang^{1,2}

1. Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing

2. Information & Electronics Technology Lab, School of Electronic Engineering

Email: wangkaishi@bupt.edu.cn

Abstract—Users of the Internet of Vehicles (IoV) have the characteristics of uneven distribution and large quantity while the terrestrial network cannot fully cover them due to technical and cost issues. Satellite-Terrestrial Integrated Network (STIN) is paving innovative technological pathways for IoV by combining terrestrial network advantages including ultra-low latency and high throughput with satellite-enabled wide-area coverage capabilities. However, computational resource constraints present a critical performance bottleneck for Satellite. To solve the problem, a high speed decoding architecture with low hardware complexity and resource consumption based on Field Programmable Gate Array (FPGA) to improve the processing speed of physical layer on the Satellite has been proposed in this paper. The architecture is simulated and implemented based on Xilinx Virtex-690T platform, and the results proved that it achieves an uplink decoding rate of 625 Mbps with lower resource consumption.

Index Terms—Satellite, Internet of Vehicle, Decoding Architecture, Field programmable gate array, Aurora high-speed interface

I. INTRODUCTION

Since LDPC (Low density parity check) codes were proposed, it has been widely studied and applied because of their excellent error-correcting ability. Due to the special structure of the check matrix, the performance of the decoding algorithm is an important factor affecting the performance of LDPC codes. At present, there have been many research results on LDPC decoding algorithms.

However, because of problems such as the dynamic topology of the STIN, the diversity of user service requirements, and the high deployment cost of edge servers, the network requirements of users for computation-intensive services cannot be met [3], [4]. For instance, [5], [6] proposed static offloading schemes that transfer computational tasks from satellite-terrestrial networks to ground-based computing centers. [7] introduced a dynamic offloading scheme by employing high-altitude Unmanned Aerial Vehicle (UAV) as an intermediate layer to share partial satellite-terrestrial computing tasks. Despite alleviating computational challenges, these terrestrial network solutions suffer from high deployment costs and significant system design complexity. Therefore, to solve problems mentioned above, enhancing the uplink processing capabilities of satellite onboard communication systems, particularly improving decoding capabilities at the physical layer receiver of satellites, remains crucial for further research.

To enhance the processing capability of the satellite onboard physical layer (PL), it is imperative to reduce hardware complexity and processing latency. Taking the 5G physical layer as an example, scholars have made the following efforts: Given that LDPC decoding exhibits high algorithmic complexity, implementation overhead, resource consumption, and prolonged processing latency, which are the key factors limiting satellite PL performance, some scholars have focused on improving traditional LDPC decoding algorithms to enhance 5G PL efficiency [8], [9], [10], [11], [12]. Meanwhile, recent works [13], [14] propose AI-driven decoding algorithms that achieve superior decoding speeds compared to conventional methods such as Offset Minsum (OMS) and Normalization–Offset Minsum (NOMS). However, their hardware realization remains highly intricate, posing significant challenges for satellite-deployment feasibility. Another branch of research focuses on optimizing non-LDPC processing modules in 5G PL. For instance, [15] proposes a Multi-Polynomial Circuit design to enhance CRC computation speed. [16] develops a fast Deinterleaver and De-rate Matcher architecture to accelerate signal processing. [17] introduces an FFT accelerator to reduce the computational burden of orthogonal frequency-division multiplexing (OFDM) systems. While these innovations demonstrate localized performance improvements in specific sub-modules, their impact on the whole PL remains marginal. Additionally, some researchers explore novel hardware platforms for implementing 5G processing pipelines. [18] demonstrates a 5G architecture based on multi-core general-purpose processors (GPPs), achieving accelerated data streaming through high-speed Ethernet interfaces. [19], [20], [21] offload portions of the 5G processing chain to high-speed GPUs, leveraging their superior floating-point computational throughput to significantly enhance processing speeds. An advantages of these approaches lies in their ability to exploit specialized hardware characteristics for acceleration. Nevertheless, how to deploy new devices on the satellite requires more work to verify its feasibility.

Considering the limitations of existing solutions and the challenges, a FPGA based satellite onboard high speed decoding architecture for IoV is proposed, and a data flow control mechanism based on the Aurora high-speed interface (AHSI) has been designed. The architecture was implemented and simulated on the Xilinx Virtex-690T

platform, demonstrating that the proposed architecture achieves an uplink decoding rate of 625Mbps with low resource consumption while enhancing parallel processing efficiency in satellite receivers. The complete decoding system is aimed to be deployed on low-Earth orbit (LEO) satellites to receive and decode uplink data from multiple IoV users. Highlights of our contributions are enumerated as follows.

1) A FPGA-based High Speed Decoding Architecture has been proposed to improve the processing speed of satellite onboard PL. The system has restructured and decoupled conventional 5G receiver processing into two independent stages deployed across different Field Programmable Gate Array (FPGA) chips, alleviated the resource contention problem caused by processing large amounts of data on a single FPGA.

2) Hardware implementation of our decoding architecture has been performed in the Xilinx Virtex-690T FPGA platform. The link simulation results, placing and routing results, the hardware complexity analysis and the resource consumption analysis of our decoding architecture proposed have been demonstrated.

This paper is organized as follows. Part II introduces the scenario and system model. Part III presents the details of the proposed decoding architecture. Part IV demonstrates the simulation results and performance analysis of the hardware implementation. Finally, Part V concludes the paper with key findings.

II. SCENARIO AND SYSTEM MODEL

A. SCENE DESCRIPTION

Our decoding architecture will be deployed on the LEO satellites. As illustrated in Fig. 1, this deployment scenario utilizes multi-layer LEO constellations with varying orbital altitudes to provide wide-area coverage of ground service regions. The satellites will receive and process uplink data from multiple ground vehicular users (IoV users). The system is deployed in the Ka band.

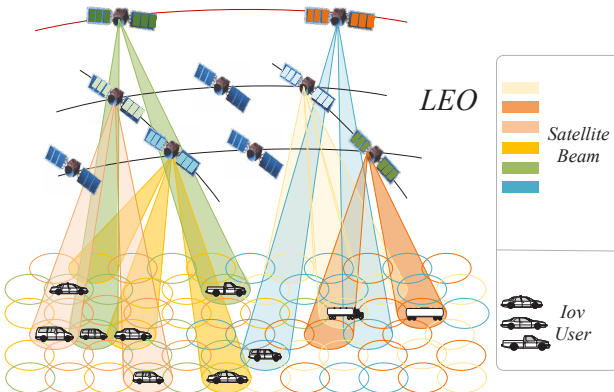


Fig. 1. Scenario of the proposed architecture.

B. SYSTEM MODEL

The proposed architecture is implemented using two FPGA chips. As depicted in Fig. 2, after receiving uplink data from ground vehicular users via the satellite air interface, the data undergoes packetization by Medium Access Control (MAC) layer before entering PL for decoding. The physical layer incorporates multiple instances of FPGA-1 → AURORA → FPGA-2 links. These instances operate in parallel, controlled by control signals dispatched from the MAC device. The data control bus handles both the distribution of control information and the routing of feedback signals, as well as enabling information exchange between different instances.

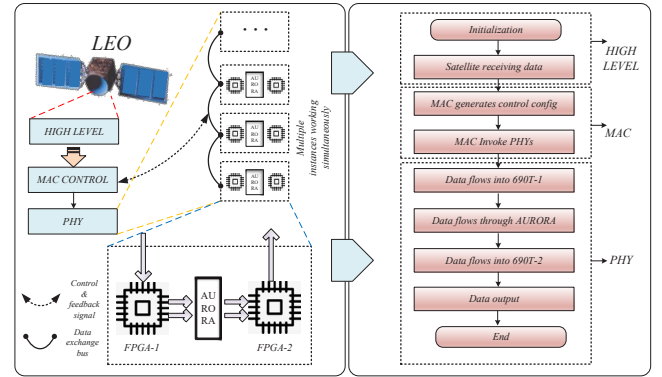


Fig. 2. Hardware architecture of two FPGA chips and AHSI & Architecture workflow.

C. PHYSICAL LAYER PROCESS & PARAMETERS

This study focuses on satellite onboard processing capacity of STIN, so we demonstrated a basic satellite onboard physical layer processing pipeline in Fig. 3. The transmitter processing involves two principal stages: symbol-level and bit-level processing. Specifically, the bit-level processing pipeline handles raw bitstream, while the symbol-level pipeline modulates bits into symbols and maps them to the time-frequency domain. At the receiver side, time synchronization module ensures symbol alignment, frequency synchronization module compensates for frequency offsets, and channel estimation module reduces the impact of transmission noise on signal integrity. The functions of the remaining modules are opposite to those of the sender.

D. AURORA HIGH-SPEED SERIAL INTERFACE

AHSI is a high-speed serial interface designed specifically for data transmission between FPGA chips, which is used in the architecture we proposed. The data processing mechanism we designed for the proposed architecture can allocate decoding channels for multiple data blocks through data flow control state machines. Data and instruction configuration information will be sent to another FPGA for decoding through AHSI, thus avoiding resource contention issues between different decoding channels.

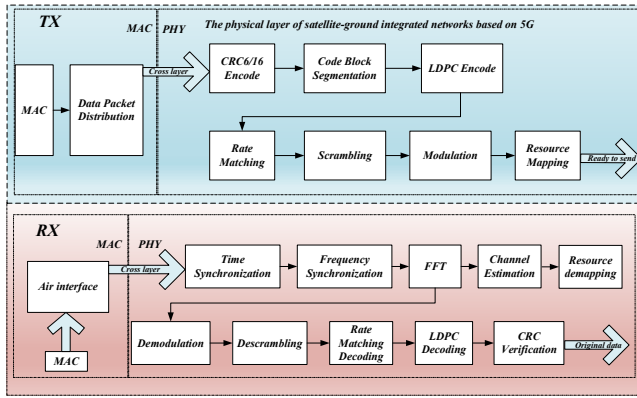


Fig. 3. Diagram of physical layer processing.

III. MULTI-USER DECODING ARCHITECTURE

This section presents the overall architecture of the proposed decoding hardware, as illustrated in Fig. 4. The green background in the diagram denotes the data processing components, while the red background highlights the data processing module which is a critical part for high speed data processing.

A. Functional modules

The data processing components incorporate signal processing modules such as time synchronization, frequency synchronization, channel estimation, etc.

In our architecture, an uplink time synchronization module is deployed within the satellite payload. The employed time synchronization algorithm combines coarse and fine synchronization due to significant timing errors in calculations based on GNSS, caused by the satellite’s high-speed mobility which result in substantial variations in multi-user data arrival times. Without fine synchronization, demodulation at the satellite receiver would be impaired. This module detects user-specific Timing Advance (TA) errors through local Zadoff-Chu (ZC) sequence detection, computes TA error values between users and the satellite, and delivers precise synchronization results for subsequent demodulation.

The Frequency Synchronization module compensates for Doppler frequency offsets. After initial ground compensation using GNSS and ephemeris data, residual frequency offsets of ± 20 kHz remain at the satellite side. For subcarriers with 120 kHz spacing in our architecture, this corresponds to a normalized fractional frequency offset of ± 0.167 . We employ the Schmid-Cox (SC) algorithm for fractional frequency offset estimation, leveraging phase differences between the two halves of training symbols for compensation.

The Channel Estimation module minimizes transmission noise impact on signal integrity. We implement the Least Squares (LS) algorithm by solving partial derivatives of the minimum mean square error equation to obtain channel frequency responses at pilot locations. LS estimation's hardware efficiency which is achieved by dividing received

pilot carriers by original pilot data, makes it ideal for resource-constrained satellite payloads.

The Resource Demapping module extracts data blocks from specified time-frequency locations. The soft demodulator processes received signals for $\pi/2$ -BPSK, QPSK, 8PSK, and 16APSK modulation schemes. The descrambling module performs despreading operations.

The Transport Data Flow Control (TDFC) module distributes user data to two Transmission Control Units (TCU) per MAC-layer instructions. The ATCU verifies the availability of parallel decoding pipelines in FPGA-2 during congested data traffic, manages an Aurora data channel, and transmits data via stream format to FPGA-2 for parallel decoding.

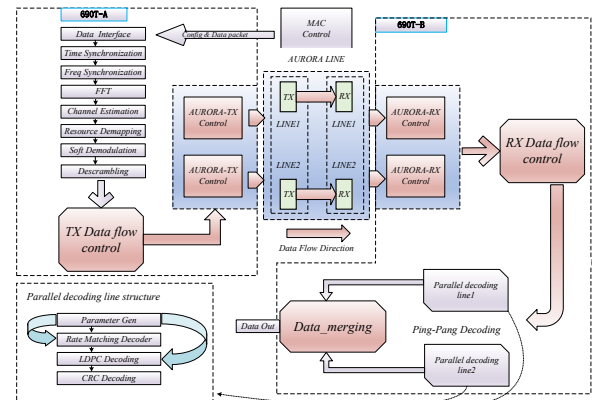


Fig. 4. Diagram of the overall decoding architecture.

The Reception Control Unit (RCU) is responsible for receiving CONFIG sequences and user data packets from FPGA-1. Accurately identify and cache control/data flow, while managing specific channel operations for each Aurora channel. The function of the Received Data Flow Control (RDFC) module is to analyze user CONFIG data (user ID, data length MCS), Route the data packet to the parallel decoding pipeline. The function of the parameter generation module is to use the CONFIG parameters to calculate parameters, including block length, code count, ZC, kb, mb, and encoding rate, and send these parameters to different decoding pipelines before the data is clock wise.

The rate matching module performs symmetric puncturing/zero-padding operations as implemented at the transmitter. Our LDPC decoding module executes layered normalized min-sum decoding algorithm which accelerating convergence versus flooding schedules while maintaining near-optimal belief propagation performance with hardware-friendly complexity. This approach is optimal for our on-board processing architecture. The CRC verification module validates packet integrity, and the data merging module reconstructs original packet sequences before outputting decoded data.

B. Data flow control mechanism

Section B provides a detailed structure of the dataflow control modules and aurora control units, which are showed

in Fig. 5. They are the key modules of our architecture. The lower left corner of Fig. 5 shows the state machine of Transmit Data flow Control (TDFC) Module, and the lower right corner shows the processing flow chart of RDFC. Fig. 5 also describes the structure of Reception Control Module and Transmit Control Module.

The combination of data flow control mechanism and functional modules is the key to achieving high speed decoding. The time synchronization, frequency synchronization, channel estimation and other modules deployed in FPGA1 alleviate the damage caused by satellite ground channel transmission, while the data flow control mechanism initializes multiple LDPC decoding pipelines in FPGA2. Due to the long processing time of LDPC decoding, congestion may occur when multiple data blocks are crowded on a decoding pipeline, resulting in decoding failure or data loss. We use feedback information designed to determine whether the pipeline is crowded, and use the high speed data processing state machine of TDFC in FPGA1 to send multiple user data packets to an idle LDPC decoding pipeline for decoding, improving this problem. The LDPC decoding pipeline used in this article has 2 lines, but this architecture has scalability and can open multiple LDPC decoding pipelines according to actual needs to achieve higher transmission rates.

TDFC manages two TCUs and operates via a three-stage state machine with four distinct states illustrated in Fig. 5: IDLE, CONFIG-GET, CONFIG-SEND, and DATA-SEND. The details of state transitions are as follows:

- IDLE State: Upon initialization, the module outputs an Aurora-reset signal to reset the ATCU. This signal triggers a hardware reset of the Aurora interface.
- CONFIG-GET State: When the Config-in-valid signal (indicating valid CONFIG data reception) is asserted, the module transitions to CONFIG-GET. Here, it captures and latches CONFIG parameters and activates an appropriate Aurora lane based on these parameters.
- CONFIG-SEND State: Upon receiving the AURORA-line-ready signal (acknowledging Aurora interface activation), the module transitions to CONFIG-SEND. It forwards the latched CONFIG data to the ATCU's FIFO buffer, which is directly connected to the active Aurora lane.
- DATA-SEND State: After transmitting CONFIG data (via Config-send-over signal), the module enters DATA-SEND to transmit user packets. Upon completion (Data-send-over is asserted), it returns to IDLE, resetting the ATCU for subsequent transmissions.

Fig. 5 shows the TCU comprises three functional blocks: Control Module, Reset Module, and FIFOs. Control Module is used to control one of the Aurora lanes. When it receives an activation signal, it will initiate the Reset Module to initialize the FIFO and sends a preparation signal to activate the designated Aurora lane. Reset Module is used to reset the FIFO buffer to prepare for new data.

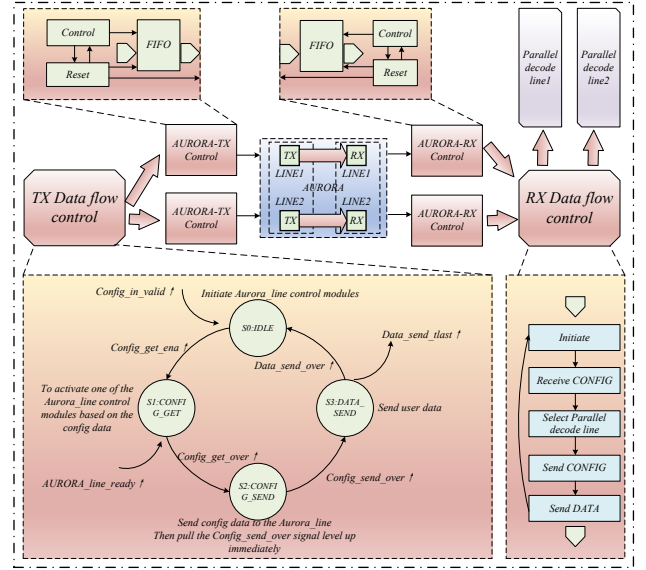


Fig. 5. A detailed structure of Data flow Control Module and Aurora control Unit.

One TCU consists of two depth-matched FIFOs that one for transmitting data payloads and another for streaming valid signals. This dual-buffered design ensures the receiving end module receives both the data stream and the valid data signal simultaneously. In order to reduce transmission power consumption, we abandoned the scheme of directly transmitting high-level effective signals. Instead, we calculated the length of the data block to be transmitted in the TDFC module and agreed with the RCU to transmit this length of data ahead of a specific clock. The RCU generated a data effective indication signal to ensure alignment between the indication signal and the data.

RCU functions similarly to its counterpart in the transmitter but must read data from the Aurora interface, which outputs only raw data and valid signals. Since the interface lacks explicit headers distinguishing data and configuration information, the RCU must buffer incoming data and differentiate between user data and preceding CONFIG metadata. Its architecture also comprises Control Module, Reset Module, and FIFO, but unlike the transmitter's design, its control module incorporates a feature-based discrimination mechanism to differentiate datastreams and CONFIG. Valid data entering the FIFO buffer is subsequently forwarded to RDFC for further processing.

The RDFC employs a ping-pong distribution logic to optimize decoding resource utilization. For instance, when data blocks DB1, DB2, and DB3 are received, the RDFC evaluates the minimum decodable latency threshold T for each block. Only when the transmission delay of data block DB1 is less than or equal to the threshold value T , it alternates data distribution between two parallel decoding pipelines, ensuring continuous processing.

The latency threshold T is derived as:

$$T = T_{g1} + T_{t2} + T_{g2}, T_{DB1} \leq T \quad (1)$$

where:

- T_{DB1} : Transmission latency of data block DB1.
- T_{t2} : Transmission latency of subsequent data block DB2.
- T_{g1} : Interval between DB1 and DB2 transmissions.
- T_{g2} : Interval between DB2 and DB3 transmissions.

IV. RESULT ANALYSIS

A. Simulation and Implementation Result Analysis

This section provides a detailed introduction to the simulation and implementation results of the parallel decoding pipeline architecture on the VIVADO platform. The testing environment is configured with a channel bandwidth of 400 MHz, of which 360 MHz is the effective bandwidth. The figure conducted link level simulations under various modulation methods and bit rates. Based on the given satellite communication payload and user terminal transmission and reception parameters, the communication link was calculated using data from the sub satellite point (with the best signal-to-noise ratio) and edge point (with the worst signal-to-noise ratio). Fig. 6 shows the block error rate curves under various modulation and coding methods.

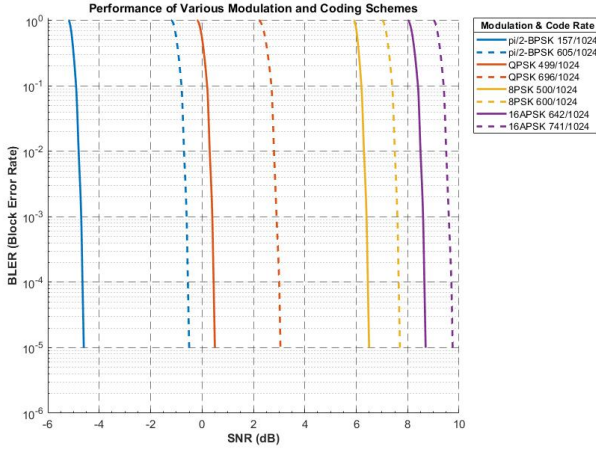


Fig. 6. Link level simulation results

The theoretical maximum throughput T was calculated as:

$$T = S_{c_n} * S_{ym_n} * S_{f_n} * Se \quad (2)$$

where S_{c_n} , S_{ym_n} , S_{f_n} and Se represent active subcarriers, symbols per subframe, subframes per second, and spectral efficiency, respectively.

Our physical layer architecture has a bandwidth of 400 MHz. After removing 10% of the protection bandwidth, the effective bandwidth is 360 MHz. To adapt to the long latency and high frequency offset characteristics of satellite scenarios, we adopted the CP-OFDM system with a subcarrier spacing of 120 KHz and an effective number of 3000 subcarriers. The physical layer system operates at a clock frequency of 200

MHz. The business channel is divided into frame, subframe, and time slot structures, with a subframe length of 0.5 ms, and one subframe containing four time slots. A time slot contains 14 symbols, of which 12 are data symbols and 2 are reference signal symbols. We first generate ten virtual user data packets, each with the size of 32768 bits and modulated by 16APSK ($Se=2.89/4$). By substituting the parameters, the maximum transmission rate can be calculated as 625 Mbps.

How the complete architecture implementation is placed and routed on the V-690T FPGA hardware platform are presented in Fig. 7 with placement positions of all functional components highlighted by different colors. It should be noted that the aurora modules incorporate the designed TCU, RCU, TDFC, RDFC, and AHSI components. The data processing modules correspond to the green background modules depicted in Fig. 4. It should be noted that DDR3 is used for caching air interface data, while PCIE is used for MAC and PHY communication.

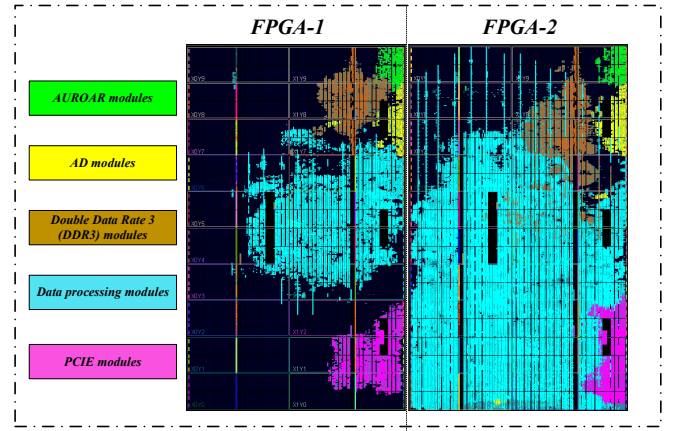


Fig. 7. Place and route: implemented by VIVADO 2018.3.

B. Hardware Complexity Analysis

TABLE I shows the comparison between our proposed architecture and existing decoding architectures. Our architecture has increased LUT consumption by 18% and memory consumption by 14% due to the introduction of a new data processing architecture, but with the same clock frequency, the maximum rate has increased by 69%. This improvement is due to our architecture shortening the decoding waiting time between large code blocks, significantly increasing the efficiency of data processing.

TABLE I
COMPARISON BETWEEN PROPOSED DECODER ARCHITECTURE AND OTHER ARCHITECTURE.

	Proposed	A. Katyushnyj [10]
Code Length	576–8424	576–8424
Frequency	200MHz	204.8MHz
LUTs	38942	32896
Block RAM&FIFO	21028	24543
Throughput (Mbps)	625	369

V. CONCLUSION

This paper proposes a novel decoding architecture for IoV. The simulation and implementation results show that by offloading the data processing processes of multiple data blocks to different FPGAs, the architecture proposed in this paper improves the decoding capability of on-board data and achieves a high transmission rate of 625Mbps with lower resource consumption. This architecture is fully implemented on traditional FPGA hardware platforms, achieving a balance between moderate implementation complexity and high deployment flexibility. For high-speed communication scenarios that require increased decoding capabilities, the system can be adjusted by making targeted modifications to the control module and increasing the number of parallel decoding lines.

In terms of compatibility with traditional satellite-terrestrial physical layer standards, the processing flow of the physical layer signal transmitter in the DVB-S2 protocol is roughly as follows: Mode Adaptation → FEC Encoding → Mapping and Physical layer framing → Base Band Filtering and Quadrature Modulation, while the processing flow of the receiver is the opposite. Our architecture is implemented based on 5G-NTN, and the above simulation and implementation results demonstrate the feasibility of the architecture. The physical layer processing flow of DVB-S2 is similar to that of 5G-NTN, so that the idea proposed in this architecture of dividing the receiver processing module into dual FPGAs for collaborative decoding can also be applied to DVB-S2 to enhance its data processing capability.

In summary, by enabling satellites to simultaneously process multiple vehicular user data streams, this work improves the uplink physical layer processing capability of satellite ground vehicular networks. The proposed architecture provides valuable reference for the design of the onboard physical layer of current and future IoT satellites, especially in high speed and low latency transmission scenarios.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (NSFC) under the Grant No.62371054.

REFERENCES

- [1] M. LiWang, S. Dai, Z. Gao, X. Du, M. Guizani and H. Dai, "A Computation Offloading Incentive Mechanism with Delay and Cost Constraints under 5G Satellite-Ground IoV Architecture," in *IEEE Wireless Communications*, vol. 26, no. 4, pp. 124-132, August 2019.
- [2] B. Ji et al., "A Vision of IoV in 5G HetNets: Architecture, Key Technologies, Applications, Challenges, and Trends," in *IEEE Network*, vol. 36, no. 2, pp. 153-161, March/April 2022.
- [3] H. Li, K. Ota and M. Dong, "Learning IoV in 6G: Intelligent Edge Computing for Internet of Vehicles in 6G Wireless Communications," in *IEEE Wireless Communications*, vol. 30, no. 6, pp. 96-101, December 2023.
- [4] M. Liao, X. Li and W. Luo, "Computing Offloading Based on Path Planning in Satellite Terrestrial Integrated Network," 2024 5th International Symposium on Computer Engineering and Intelligent Communications (ISCEIC), Wuhan, China, 2024, pp. 521-526.
- [5] L. Liu, M. Zhao, M. Yu, M. A. Jan, D. Lan and A. Taherkordi, "Mobility-Aware Multi-Hop Task Offloading for Autonomous Driving in Vehicular Edge Computing and Networks," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 2169-2182, Feb. 2023.
- [6] Y.Zhang, C. Chen, L. Liu, D. Lan, H. Jiang and S. Wan, "Aerial Edge Computing on Orbit: A Task Offloading and Allocation Scheme," in *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 1, pp. 275-285, 1 Jan.-Feb. 2023.
- [7] S. Gu, X. Sun, Z. Yang, T. Huang, W. Xiang and K. Yu, "Energy-Aware Coded Caching Strategy Design With Resource Optimization for Satellite-UAV-Vehicle-Integrated Networks," in *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 5799-5811, 15 April 2022.
- [8] B. N. Tran-Thi, T. T. Nguyen-Ly, H. N. Hong and T. Hoang, "An Improved Offset Min-Sum LDPC Decoding Algorithm for 5G New Radio," 2021 International Symposium on Electrical and Electronics Engineering (ISEE), Ho Chi Minh, Vietnam, 2021, pp. 106-109.
- [9] A.Verma and R. Shrestha, "Low Computational-Complexity SOMS-Algorithm and High-Throughput Decoder Architecture for QC-LDPC Codes," in *IEEE Transactions on Vehicular Technology*, vol. 72, no. 1, pp. 66-80, Jan. 2023.
- [10] A. Katyushnyj, A. Krylov, A. Rashich, C. Zhang and K. Peng, "FPGA implementation of LDPC decoder for 5G NR with parallel layered architecture and adaptive normalization," 2020 IEEE International Conference on Electrical Engineering and Photonics (EExPolytech), St. Petersburg, Russia, 2020, pp. 34-37.
- [11] Jang, H. Jang, S. Kim, K. Choi and I. -C. Park, "Area-Efficient QC-LDPC Decoding Architecture With Thermometer Code-Based Sorting and Relative Quasi-Cyclic Shifting," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 71, no. 6, pp. 2897-2910, June 2024.
- [12] H. Cui, F. Ghaffari, K. Le, D. Declercq, J. Lin and Z. Wang, "Design of High-Performance and Area-Efficient Decoder for 5G LDPC Codes," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 2, pp. 879-891, Feb. 2021.
- [13] Shah and Y. Vasavada, "Neural Layered Decoding of 5G LDPC Codes," in *IEEE Communications Letters*, vol. 25, no. 11, pp. 3590-3593, Nov. 2021.
- [14] T. Du, H. Ju, Y. Xu, D. He and W. Zhang, "A Deep Learning based Multi-edge-type decoding algorithm for 5G NR LDPC codes," 2023 International Wireless Communications and Mobile Computing (IWCMC), Marrakesh, Morocco, 2023, pp. 224-228.
- [15] -H. Pan, S. Zhong, P. Zhang and T. Yuan, "A Multi-Polynomial CRC Circuit for 5G Standard Using Parallel Pipelining Architecture," 2021 Cross Strait Radio Science and Wireless Technology Conference (CSRSWTC), Shenzhen, China, 2021, pp. 307-309.
- [16] N.Filipovic, D. El Mezeni and A. Radošević, "Hardware Implementation of 5G NR Deinterleaver and De-rate Matcher," 2021 15th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS), Nis, Serbia, 2021, pp. 57-60.
- [17] Y.Wu, P. Wang and J. McAllister, "Programmable Dataflow Accelerators: A 5G OFDM Modulation/Demodulation Case Study," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020, pp. 1728-1732.
- [18] X. Yang et al., "RaPro: A Novel 5G Rapid Prototyping System Architecture," in *IEEE Wireless Communications Letters*, vol. 6, no. 3, pp. 362-365, June 2017.
- [19] C.Tarver, M. Tonnemacher, H. Chen, J. Zhang and J. R. Cavallaro, "GPU-Based, LDPC Decoding for 5G and Beyond," in *IEEE Open Journal of Circuits and Systems*, vol. 2, pp. 278-290, 2021.
- [20] A. Aronov, L. Kazakevich, J. Mack, F. Schreider and S. Newton, "5G NR LDPC Decoding Performance Comparison between GPU & FPGA Platforms," 2019 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, USA, 2019, pp. 1-6.
- [21] N. -Y. Kuo and T. -Y. Hsu, "AMD GPU-Based Selected Acceleration of PRACH and PUSCH in 5G NR," in *IEEE Access*, vol. 13, pp. 30365-30376, 2025.