AI-assisted Modem Implementation for Intelligent Wireless Access Systems

Yunjoo Kim, Jungbo Son, Yuro Lee, JungSook Bae
Terrestrial & Non-Terrestrial Integrated Telecommunications Research Laboratory
ETRI (Electronics and Telecommunications Research Institute)
Daejeon, Korea
{yunjoo, jbson, yurolee, jsbae}@etri.re.kr

***Abstract*—This paper proposes a modem architecture that integrates a hardware modem based on a field-programmable gate array (FPGA) and a software modem that uses a graphics processing unit (GPU) to support machine learning (ML)-based channel estimation. In a proof-of-concept (PoC) implementation for an intelligent wireless access system, the proposed modem achieved an improvement of approximately 3.5 dB in signal-to-noise ratio (SNR) under real-time conditions. In addition, parallel processing reduced latency and ensured stable operation.**

***Index Terms*—ML, parallelization, system interface, FPGA, modem, deep learning acceleration, intelligent wireless access**

I. Introduction

As artificial intelligence (AI) technologies are introduced to a wireless access system, intelligent modems that support data training or decision-making functions become more important. To implement this modem, high-performance and heterogeneous systems include graphics processing unit (GPU)-based deep learning accelerators and neural processing units (NPUs) [1] [2]. However, a machine learning (ML) application in the physical layer remains limited due to strict real-time requirements. While the physical layer is commonly implemented on platforms like field-programmable gate array (FPGA) and system-on-chip (SoC), FPGA design is unsuitable for software-oriented AI module development [3] [4]. Therefore, an architecture suitable for ML algorithms and capable of efficient development is needed [5].

This paper presents an AI-assisted modem where the channel estimation function runs on GPU-based software and other physical layer functions run on an FPGA. The architecture was applied to a proof-of-concept (PoC) system, and its reception performance was evaluated based on signal-to-noise ratio (SNR) and real-time performance.

TABLE I: System Parameters

Variable	Value
System bandwidth (MHz)	400
Subcarrier spacing (kHz)	120
FFT size	4096
Number of subcarriers	3072
Occupied bandwidth (MHz)	368.64
OFDM symbol duration (μs)	8.33
CP length	288
CP duration (μs)	0.59
Number of OFDM symbols per slot	14
Slot duration (μs)	125
Subframe duration (ms)	1
Midframe duration (ms)	2
Radio frame duration (ms)	10
System clock (MHz)	122.88
Intermediate frequency (MHz)	3300
DAC/ADC sampling frequency (MHz)	3932.16

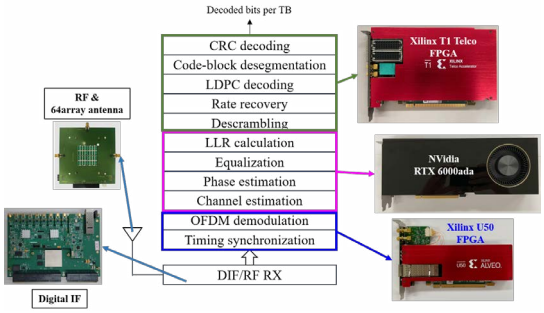


Fig. 1: Block diagram of ML-based reception modem

II. Intelligent Wireless Access System Design

A. System description

An intelligent wireless access system operates on the 28 GHz millimeter-wave band. The physical layer uses orthogonal frequency division multiplexing (OFDM) and adopts time division duplexing (TDD) to separate the uplink and downlink transmissions. The main parameters are listed in Table I.

A radio frame consists of 10 subframes (1 ms each), each subdivided into 8 slots with 14 OFDM symbols per slot. Two subframes make one mid-frame, totaling 16 slots. Within these 16 slots, 12 are for downlink, 3 for uplink, and 1 is a switch slot. This system supports a downlink speed of up to 1 Gbps.

B. Implementation Details

An AI-assisted modem consists of a hardware (FPGA) modem for basic wireless signal reception and data delivery, and a software (GPU) modem that uses an ML module for channel estimation to improve system performance. Fig.1 illustrates the reception flow between the hardware and the ML-based software modem. The set of GPU processes is defined as ML demodulator, which adopts a Transformer decoder-based neural network architecture.

During this process, in-phase/quadrature (I/Q) data and log-likelihood ratio (LLR) data are exchanged in real time between the two computing resources (FPGA and GPU)

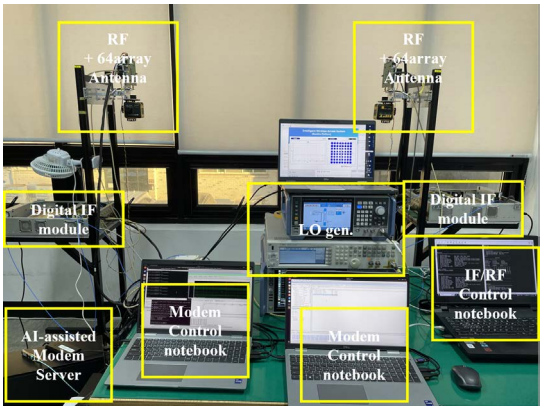


Fig. 2: AI-assisted modem system PoC testbed

via the software interface. The software interface handles a parallel instance-based approach to ensure stable real-time performance, even under slot processing delays.

III. EVALUATION AND DISCUSSION

The HW and ML demodulators were compared and their performance was measured under identical conditions.

A. Experimental Setup

The AI-assisted modem server is equipped with an NVIDIA RTX 6000 Ada GPU and Xilinx U50 and T1 boards. The HW demodulator is implemented on the T1 RFSoC FPGA. The ML demodulator is implemented on TensorRT and the compute unified device architecture (CUDA) 12.1 kernel.

The number of slots was set to one when measuring the performance of received signals. To analyze system latency and throughput, the number of slots was set to 4, 6, and 7 within a mid-frame. The measurement period was from the point at which the IQ input arrived at the AI module to the point at which the LLR output started.

B. Experimental Results and Analysis

Fig. 3 shows the processing time for 4 and 8 instances under slot condition 4, 6, and 7. More instances improve parallelism but also increase CPU load. With 4 or 8 instances, the ML demodulator processed 7 out of 12 slots in real time, achieving a throughput of 58.3%, which is approximately 3.5 times higher than the 16.7% with a single instance.

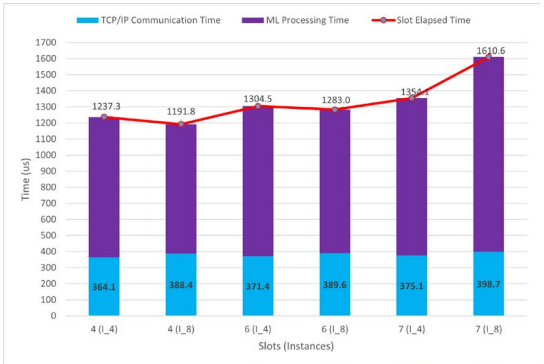
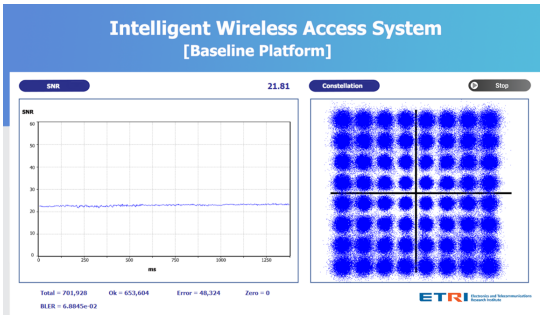
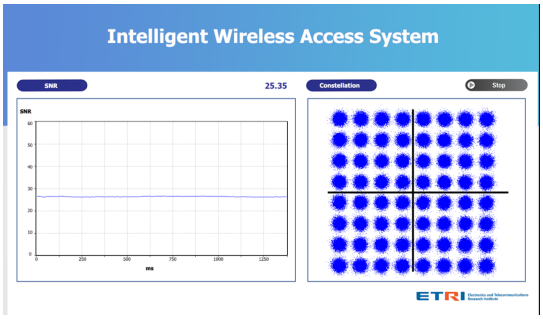


Fig. 3: Latency in ML demodulation (instances = 4, 8)



(a) HW-Modem System



(b) AI-assisted Modem System

Fig. 4: Received Constellation and SNR

In the testbed, when the 64-quadrature amplitude modulation (64-QAM) data was transferred, Fig. 4 (a) and (b) show the constellations of the received signals and the corresponding SNR values. The ML demodulator achieved an SNR of 25.35 dB, which is an improvement of approximately 3.5 dB over the HW demodulator's value of 21.81 dB.

IV. CONCLUSIONS

The proposed AI-assisted modem uses an FPGA for hardware processing and a GPU for ML-based channel estimation. The experiments showed better signal quality and increased throughput with parallel instances. These results support the practical feasibility in real-time wireless systems, and future work will address scalability and FGPA-GPU latency.

ACKNOWLEDGMENT

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-00972, Development of Intelligent Wireless Access Technologies).

REFERENCES

[1] E. Jeong, J. Kim, S. Tan, J. Lee, and S. Ha, "Deep learning inference parallelization on heterogeneous processors with TensorRT," *IEEE Embed. Syst. Lett.*, vol. 14, no. 1, pp. 15–18, 2022.

[2] Y. Zhou and K. Yang, "Exploring TensorRT to improve real-time inference for deep learning," in *Proc. IEEE HPCC*, 2022, pp. 2011–2018.

[3] S. A. U. Haq et al., "Deep neural network augmented wireless channel estimation for preamble-based OFDM PHY on Zynq SoC," *IEEE Trans. VLSI Syst.*, vol. 31, no. 7, pp. 1026–1038, 2023.

[4] A. Boutros, E. Nurvitadhi, and V. Betz, "Specializing for efficiency: Customizing AI inference processors on FPGAs," in *Proc. Int. Conf. Microelectron.*, 2021, pp. 62–65.

[5] F. Yan, A. Koch, and O. Sinnen, "A survey on FPGA-based accelerators for ML models," 2024. [Online]. Available: arXiv:2412.15666