Evaluation of FPGA Based Speech Recognition Hardware for Edge Devices

Qiyue Tu*

Witing School of Engineering, Johns Hopkins University,Baltimore, MD21218, United States

*Corresponding author: qtu6@jh.edu

Abstract. Current wearable devices are limited by power consumption and size, and voice recognition mostly relies on cell phone processing. Localized deployment has the potential advantages of improving privacy security and reducing latency. This paper synthesizes several research in this field to analyze the low power and latency advantages of Field Programmable Gate Array (FPGA) hardware over traditional Central Processing Unit (CPU) and CPU plus Graphics Processing Unit (GPU) solutions for speech recognition tasks. It also demonstrates the significant potential of FPGAs in terms of energy efficiency and real time performance in conjunction with Spiked Neural Networks (SNNs) and their hardware optimization strategies. Experimental data from studies shows FPGA hardware and SNN combination scheme can reach 11.5× and 40× energy efficiency compare to CPU and GPU which provides a feasible path for implementing native speech processing in future wearable devices. Potential future optimization directions such as further optimizing the on-chip memory layout and alternative of CPU controller are also analyzed based on the current state of technology development.

Keywords: Speech Recognition; Natural Language Process; FPGA.

1. Introduction

Nowadays, smart wearable devices are becoming increasingly commonplace. From the investing perspective, the wearable market is anticipated to keep on growing exponentially in the coming years. According to market analysis forecasts wearable devices market is expected to maintain a growth rate of more than 20% per year [1]. Smart devices such as true wireless earbuds, smart glasses, and smartwatches are undergoing rapid development and gaining numerous new features. Enabling more convenient and natural interaction with these wearable devices has become a key objective in product development and voice interaction is a significant component of human device communication.

Current mainstream machine voice interaction for smart audio devices are primarily three major steps: Automatic Speech Recognition (ASR), Large Language Models (LLM), and Text to Speech (TTS) [2,3]. ASR processes audio data captured by microphones through techniques such as Fast Fourier Transform (FFT), then converts it into text via algorithms. A LLM then interprets the text content and generates responses such as text replies or executing corresponding commands [2]. Finally, in the TTS step, the text feedback is converted into audio segments and output through speakers [3].

However, the vast majority of voice recognition jobs in modern smart wearables still mainly entail carrying out simple preprocessing and returning audio data to the primary smart device, such a smartphone, for additional ASR processing. Mainstream wearable device systems like Android provide communication APIs for transmitting audio signals enabling voice interaction on such devices, allowing applications deployed on smartphones to execute operations such as LLMs. This is due to the dual constraints of power consumption and size within smart wearable devices [4]. Currently, the battery capacity of smart wireless earbuds and smart wristbands generally falls below 500 mAh at 3.7 volts. The power consumption of voice recognition systems based on CPU or GPU platforms typically exceeds 5 watts which means the device's battery would only last approximately 18 minutes [5-7]. So, the chips equipped in such products still lack the computational power required to run localized language recognition models locally. If edge devices can deploy local speech recognition, voice interaction between smartphones and smart wearables could gain the option to

transmit messages in text format. Then, on high-performance processing devices such as smartphones, the LLM makes decisions or executes the required operations, and finally returns feedback via TTS. Text-based transmission can significantly reduce the required bandwidth, thereby reducing response delays caused by transmission to the smartphone.

Additionally, achieving local deployment of speech recognition on wearable devices could offer key advantages in data privacy, storage and response latency for wearable and edge systems. It reduces the amount of Random Access Memory (RAM) occupied by the phone itself and lowers the processing power requirements. Locally captured audio data encompasses not only voice information but also a lot of other sonic information. If language recognition can be performed locally, private audio information will no longer pass through directly internet connected devices like smartphones or computers, thereby reducing the risk of this critical data being stolen.

This paper will primarily explore the deployment of speech recognition technology on wearable devices. The low latency, low power consumption, and adaptable parallelism of FPGA architectures enable advanced on device processing such as real time recognition even translation tasks. The article first analyzes how FPGAs, as deployment hardware platforms, offer significant architectural advantages for language recognition tasks in wearable devices. The paper compares SNNs with the currently more popular Artifical Neural Networks (ANNs) and Convolutional Neural Networks (CNNs). Then, multiple researches and applications like performance testing of Efficient Speech Recognition Engine demonstrates that low power FPGA architectures coupled with SNNs and low-level hardware optimization attain impressive performance and efficiency gains. Such architectures demonstrate a remarkable decrease in latency, power, and memory profile of the solutions with respect to their accuracy on speech related applications. These results illustrate the great potential of neuromorphic FPGA platforms, enabling real time low power processing for future edge and wearable applications. Future directions for optimization in this area could include optimizing operational logic to guarantee low power consumption while maintaining performance in a variety of speech recognition environments, as well as fine-tuning the location and size of on-chip storage based on task types to further reduce read/write latency.

2. Theory

2.1. Advances in FPGA over Traditional Hardware

FPGAs differ significantly in structure from traditional CPUs, making them particularly advantageous for operation in edge devices like headphones where power and size constraints are severe. Traditional CPUs face the challenge that, due to the serial pipeline processing mode adopted by individual CPU cores, when handling audio signals with substantial real time data throughput, they must either increase frequency to reduce processing latency or vice versa. In other words, processors with this architecture face a trade-off between increasing operating frequency and extending operating time. The other current dominant hardware configuration, CPU with GPU, is even less suitable on edge devices. GPUs rely heavily on off-chip memory, which consumes significant power. GPUs also have high static power and are shared across many cores, reducing energy efficiency for small batch or real time inference [5]. What's more, currently most of GPUs are optimized for 16/32-bit floating-point, which is overkill for inference and wastes energy [8]. This means the performance of GPU will be greatly sacrificed on edge devices.

FPGAs, which can adopt a parallel architecture, offer a solution that balances both power consumption and processing latency. In the current electronic audio research and industry, FPGA hardware has long been very widely used in real time audio Digital Signal Processing (DSP) field. The low-level architecture of FPGAs delivers outstanding real time audio processing performance, providing extremely low latency and high sampling rates. Their reconfigurable nature enables deep optimization for specified applications, allowing for with adaptable levels of parallelism and pipelining to fully exploit the system's performance potential [9,10]. This enables them to break the

current reliance of wireless audio devices on directly transmitting audio data back to high-performance computers for speech recognition.

Beyond strictly defined computational capability, FPGA platforms also offer the flexibility to modify their design dynamically according to task requirements, is easily scalable in a modular approach. This means that product designers of wearable devices utilizing this hardware architecture can freely scale components to achieve the required speech recognition hardware processing power. Its high flexibility hardware architecture also provides possibility for customize modification to meet diverse task requirements for instance different accent or languages. Besides that, FPGAs can be specialized to exploit sparsity and event driven behavior. In speech recognition tasks, audio signals are typically sparse. That means FPGAs can process only active speech segments, avoiding redundant computations thereby further significantly reducing energy consumption.

The advantages mentioned above, demonstrate that the FPGA platform can provide a suitable hardware foundation for the edge device speech recognition application scenarios predefined in this paper.

2.2. Spiking Neural Networks

Innovations in hardware architecture also need to be complemented by new neural network models on the principal end of deep learning. Existing mainstream neural networks require massive amounts of computational resources and extremely high system configuration. It is of low efficiency and not easy to implement on hardware especially on portable devices. Spiking neural networks (SNNs) are often called the third generation of neural networks because they process mainly timing information, making them closer to how the brain works. Existing cases of deploying speech recognition on FPGAs like the Spiking Long Short-Term Memory (LSTM) Accelerator and Efficient Speech Recognition Engine (ESE) project both utilize SNNs as their deployment neural network solution for running their speech recognition models. This is primarily due to the exceptional compatibility of its deployment on FPGA hardware architecture.

Unlike traditional artificial or convolutional neural networks (ANNs/CNNs), where every neuron is activated in each layer, SNN’s event driven mechanism mimics the way biological neurons communicate through spike trains, which only activate neurons when their membrane potential reaches a certain threshold.. As a result, SNNs require less computation, reduce data transfer between layers, and are more energy efficient. And the sparse, event driven nature of SNNs allows them to work well with FPGA hardware, making them highly suitable for hardware implementation of edge device speech recognition [11,12].

SNN revolutionizes the data format of existing neural networks in both data storage and data processing perspective, which will make them more suitable for operation on FPGA hardware structures. Unlike conventional neural networks that use real numbers to represent activations, SNNs adopt sequences of binary spikes, making them more biologically plausible [13]. In contrast, popular CNNs rely on continuous values that demand high computational resources, memory bandwidth, and multipliers, leading to high power consumption and low hardware efficiency. While SNNs require fewer multipliers to process spikes, offering much potential for energy savings and making them highly suitable for FPGA implementations. Among various SNN models, the Hodgkin-Huxley model closely mimics biological neurons but is too computationally expensive, while the simpler Integrate-and-Fire model trades off some biological accuracy for better hardware compatibility[13].

SNN itself also facilitates more flexible memory access. Combined with the structural advantages of FPGAs, it addresses data throughput bandwidth bottlenecks at the architectural level. Within Effective Automated Spiking Neural Network (EASpiNN), develop team optimizes the SNN implementation by improving both computation and memory access[12]. For computation, it applies loop pipelining and parallelization to key computing engines. For memory, EASpiNN automatically determines the best cutoff to buffer as much of the weight matrix as possible, based on the model parameters and the FPGA's memory resources, it also enables burst access for all off-chip Dynamic RAM operations[12]. That overcomes challenge in storing the large weight matrix, which often

exceeds the available on-chip Block RAM or Ultra RAM capacity in embedded FPGAs for networks like MNIST and CIFAR-10[12].

3. Existing Relevant Case Studies

Both the Spiking LSTM Accelerator and ESE projects utilize FPGA hardware to run recognition algorithms, supplemented by traditional CPU architecture as the control chip. Their test data provides strong support for comparing the performance of two hardware platforms in running deep learning speech recognition tasks.

3.1. The Spiking LSTM Accelerator

The Spiking LSTM Accelerator is a deep learning accelerator for speech recognition, specifically designed for spike neural networks and deployed on FPGA hardware platforms. The team uses a low-cost circuit in the LSTM gate to drastically cut power consumption and hardware costs. Then, in order to minimize inference latency, quantize, and effectively deploy the synapses of the spiking LSTM network, they propose a serial parallel processing architecture and hardware implementation[5]. The team's evaluation of speech recognition acceleration performance shows: The FPGA-based spiking LSTM accelerator achieved a significant speedup of 8.3× over CPU and 6.6× over GPU. In addition, it consumed 728 times less energy per inference than the CPU, and 1442 times less than the GPU, when running on the Artix-7 XC7A200t platform [5]. The team's performance evaluation of speech recognition acceleration shows: The FPGA's adaptable hardware architecture makes use of high level data quantization and the combined benefits of serial and parallel processing, allowing it to avoid ineffective multiplication operations and redundant memory accesses (Table 1 and Table 2).

Table 1. Power Efficiency of Different Platforms

Platform	Energy/Inf. (μJ/I)
Intel Xeon Silver 4214R CPU	65.5
NVIDIA A100 GPU	129.8
Xilinx Artix-7 XC7A200t	0.087
Xilinx Zynq-7000 XC7Z020	0.066

Table 2. Summary of Comparative Data Provided by the Spiking LSTM Accelerator Team

	Energy Consumption(mJ)	Operation Latency(us)
CPU	65.5	654
GPU	129.8	519
FPGA	0.09	78.93

3.2. Efficient Speech Recognition Engine

ESE is a speech recognition engine based on FPGA hardware that employs a load-balancing sensing pruning technique. The hardware architecture that directly utilizes the sparse LSTM model was created by the team. There's a host CPU that runs software, manages the system, and communicates via peripheral component interconnect express (PCIe). The FPGA accelerator contains processing elements (PEs) for parallel computation, on-chip buffers, and a controller to manage the data flow and computation. The On-board Dynamic RAM stores large LSTM model parameters and data, accessed by the FPGA. It employs a load-balance-aware pruning method that compresses the LSTM model by 20×, with negligible accuracy loss. The compressed model is then partitioned and scheduled across multiple processing elements using a dedicated scheduler, enabling parallel execution and efficient data flow management. Evaluated on an LSTM speech recognition benchmark, ESE achieved speedups of 43× over an Intel Core i7-5930K CPU and 3× over a Pascal Titan X GPU, while being 40× and 11.5× more energy efficient, respectively [8] (Figure 1).

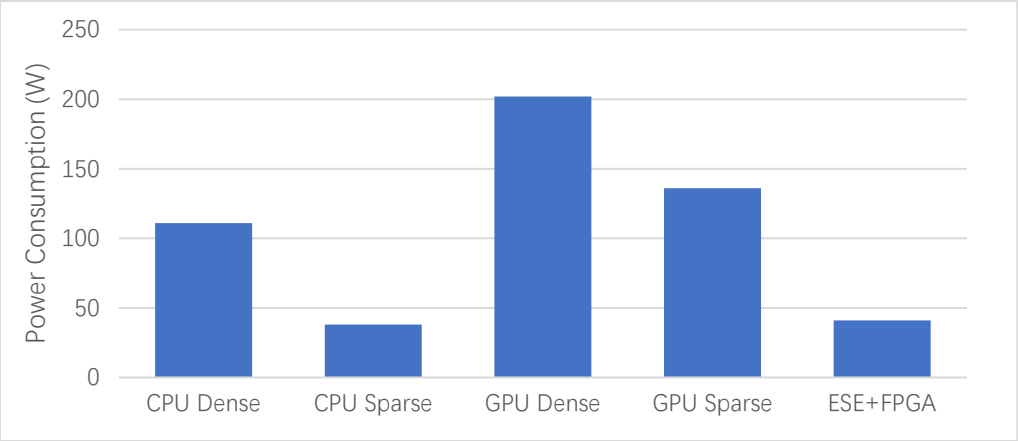

Figure 1. ESE is 40× and 11.5× higher energy efficient than Core i7 5930k CPU and Pascal Titan X GPU. (Picture credit: Original)

4. Direction of Further Improvements

There are many optimization directions in the FPGA plus SNN speech recognition framework described in the previous section.

As previously mentioned, the multiplier-free design offers latency advantages when integrated with FPGA-based SNN architectures. Under this hardware design premise, data quantization processing represents one of the key factors for further reducing power consumption. Optimizing the processing length of segments based on the size of paragraph information in the target language represents a straightforward and effective optimization approach.

Besides that, further optimizing the on-chip memory layout to enhance data input and output efficiency is also a top priority for further performance optimization. In deep learning applications, on-chip memory is essential for reducing latency. The energy used to access external memory, such as Double Data Rate (DDR) Memory, is greater than that of the chip's internal computation. In addition to reducing the power consumption and expense related to high memory bandwidth, a larger on-chip cache helps alleviate the memory bottleneck brought on by frequent external memory access. FPGA can drastically minimize reads and writes to external DDR memory because of its large on-chip memory capacity and adaptable setup[7]. On the other hand, GPUs need an extra processor to sustain them while they are operating, and using more external hardware significantly slows down data processing. Furthermore, FPGA's powerful raw data processing capability and reconfigurability enable it to handle data at any custom precision, while GPU's data processing is often limited by its development platform [7].

Another major factor limiting hardware latency is that the CPU controller in the current project still handles a significant number of tasks. In traditional designs, the hardware module for collecting voice data and the machine learning software application are often designed as discrete components of an edge device. Because hardware and software may be designed and tested separately, this split method streamlines the design process, but it frequently results in slower processing speeds when computations are performed by software [14]. The Von Neumann architecture, which necessitates frequent instruction fetching, is the primary source of this delay. Most audio preprocessing activities in speech recognition, such as FFT, are DSP operations that largely include adds and multiplications, therefore it is more efficient to execute them directly in hardware rather than depending on sophisticated software-based instructions [14]. Although voice recognition in wearable devices cannot entirely eliminate the need for controllers in principle, introducing dedicated DSPs to handle input output and data preprocessing represents a major optimization direction for reducing latency in the future.

5. Conclusion

FPGAs with a parallel architecture can process longer audio segments simultaneously. Its lower frequency and more optimized on-chip memory layout offers superior power consumption control and latency compared to GPUs.

The next generation neural network architecture SNNs mimic the mechanism of human brain so that neurons in a network biologically connect to each other through links or synapses has high potential working on FPGA platforms. The key innovations of SNNs including “pseudo- simultaneity” and topology in hardware that especially suitable for FPGAs.

Through analyzing case studies from several papers that contains projects using SNNs on FPGAs, to demonstrate their advantages over CNNs in terms of latency and power consumption when deployed in FPGA environments. In speech recognition tasks, ESE achieves energy efficiency improvements of 11.5 times and 40 times compared to CPU and GPU implementations respectively. The Spiking LSTM replaces traditional multipliers with multiplexers and incorporates on-chip storage to further optimize energy efficiency and latency.

And at the end, the possible future optimization strategies for these tasks. The paper explored structural efficiency improvements in FPGA based SNNs. Architectural optimizations such as multiplier less neuron design, low bit quantization, and reorganized memory greatly enhance performance. According to the developers of ESE engine, these techniques reduce DSP usage by over 60%, save up to 40× memory, and improve speed by 14×. Overall, event driven and hardware aware designs enable over 20× lower power consumption than traditional accelerators.

At the same time, the use of this type of FPGA still faces multiple challenges: Under the current industrial technology development environment, custom FPGAs require high costs, which cannot be sufficiently amortized in small scale production. Therefore, for this type of wearable smart device product, a unified hardware solution must be adopted to spread hardware development and production costs as evenly as possible. This, however, runs counter to the highly customized nature that defines this field. Furthermore, the highly customized nature of FPGAs also imposes certain limitations on software development. Currently,existing experience indicates that deep learning models usually undergo rapid iteration over time, which means software updates will far outpace hardware lifespans. This places significant demands on the software adaptation and compatibility capabilities of FPGAs.

References

[1] A. Ometov et al., “A survey on wearable technology: history, State-of-the-Art and current challenges,” Computer Networks, vol. 193, p. 108074, Apr. 2021, doi: 10.1016/j.comnet.2021.108074.

[2] T. Chen, Y. Yang, C. Qiu, X. Fan, X. Guo, and L. Shangguan, “Enabling Hands-Free Voice Assistant Activation on Earphones,” Jun. 03, 2024, pp. 155–168. doi: 10.1145/3643832.3661890.

[3] B. V. Varun, S. M. Kusuma, and G. Reddy, “AI-EDge based voice responsive smart headphone for user context-awarenes,” 2020 IEEE International Conference on Electronics, Computing and Communication Technologies, pp. 1–5, Jul. 2020, doi: 10.1109/conecct50063.2020.9198484.

[4] “Get started with Wear OS,” Android Developers. https://developer.android.com/training/wearables

[5] T. Yin, F. Dong, C. Chen, C. Ouyang, Z. Wang, and Y. Yang, “A spiking LSTM accelerator for automatic speech recognition application based on FPGA,” Electronics, vol. 13, no. 5, p. 827, Feb. 2024, doi: 10.3390/electronics13050827.

[6] W.-J. Luo, C. B. D. Kuncoro, and Y.-D. Kuan, “Wireless Power Hanger Pad for portable wireless audio device Power charger application,” Energies, vol. 13, no. 2, p. 419, Jan. 2020, doi: 10.3390/en13020419.

[7] C. Wang and Z. Luo, “A Review of the Optimal Design of Neural Networks Based on FPGA,” Applied Sciences, vol. 12, no. 21, p. 10771, doi: 10.3390/app122110771.

[8] S. Han et al., “ESE: Efficient Speech Recognition Engine with Sparse LSTM on FPGA,” arXiv (Cornell University), Dec. 2016, doi: 10.48550/arxiv.1612.00694.

[9] R. Michon, M. Ducceschi, P. Cochard, T. Skare, C. J. Webb, and R. Russo, “Evaluating CPU, GPU, and FPGA performance in the context of modal reverberation: a comparative analysis,” Frontiers in Signal Processing, vol. 5, Apr. 2025, doi: 10.3389/frsip.2025.1522604.

[10] M. Popoff, R. Michon, T. Risset, Y. Orlarey, and S. Letz, “Towards an FPGA-Based compilation flow for Ultra-Low latency audio signal processing,” 2023. https://api.semanticscholar.org/CorpusID:262133771

[11] Q. T. Pham, T. Q. Nguyen, P. C. Hoang, Q. H. Dang, D. M. Nguyen, and H. H. Nguyen, “A review of SNN implementation on FPGA,” 2021 International Conference on Multimedia Analysis and Pattern Recognition (MAPR), pp. 1–6, Oct. 2021, doi: 10.1109/mapr53640.2021.9585245.

[12] S. Panchapakesan, Z. Fang, and N. Chandrachoodan, “EASpiNN: Effective Automated Spiking Neural Network Evaluation on FPGA,” 2020 IEEE 28th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), p. 242, May 2020, doi: 10.1109/fccm48280.2020.00075.

[13] P. Plagwitz, F. Hannig, J. Teich, and O. Keszocze, “To spike or not to spike? A quantitative comparison of SNN and CNN FPGA implementations,” arXiv (Cornell University), Jun. 2023, doi: 10.48550/arxiv.2306.12742.

[14] J. Kwon and D. Park, “Hardware/Software Co-Design for TinyML Voice-Recognition application on resource frugal edge devices,” Applied Sciences, vol. 11, no. 22, p. 11073, Nov. 2021, doi: 10.3390/app112211073.