electronics

Editorial

Advanced AI Hardware Designs Based on FPGAs

Joo-Young Kim

School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Korea; jooyoung1203@kaist.ac.kr

Artificial intelligence (AI) and machine learning (ML) technology enable computers to run cognitive tasks such as recognition, understanding, and reasoning, which are believed to be processes that only humans are capable of, using a massive amount of data. As more industries adopt the technology, we face an ever-increasing demand for new hardware that can perform faster and more energy-efficient ML processing.

In recent years, traditional hardware vendors such as Intel and Nvidia, as well as new start-up companies such as Graphcore, Groq, and HabanaLabs, have tried to offer the best computing platform for complex ML algorithms. Although GPU is still the preferred computing platform due to its large userbase and well-established programming interface, its top spot will not be safe forever due to its low hardware utilization and lousy energy efficiency. In addition to performance and energy efficiency, adapting fast-changing AI/ML algorithms is another hot topic in AI hardware. FPGA has a clear benefit on this point, as it can reprogram to amend its processing quickly with a relatively low-power budget. In this Special Issue, we invite the latest developments in the field of advanced AI hardware design based on FPGA, which can show the device's strengths, including software-hardware co-design, customization, fast-prototyping, and scalability. We have accepted a total of 12 interesting papers that cover important research topics in the domain.

One of the most distinctive features of FPGA that sets it apart from other hardware platforms is that its logic is reconfigurable with programming. Wang et al. [1] built a dynamically partial reconfigurable (DPR) system model that only meets the actual application requirements to improve the execution efficiency. Further research for FPGA infrastructure was undertaken by Rios-Navarro et al. [2]. They propose a practical software API solution that efficiently organizes the memory to prevent reallocating data from one memory area to another in the programmable SoC on FPGA.

There are a few interesting papers about convolutional neural network (CNN) acceleration on FPGA, using software–hardware co-design approaches. Li et al. [3] propose the adaptive pointwise convolution and 2D convolution joint network (AP2D-Net). Using the network, they present an ultra-low-power and relatively high-throughput system combined with dynamic precision weights and activation for unmanned aerial vehicle's object detection scenarios. Han et al. [4] propose a hardware-amenable network model based on the dilate gated convolutional neural network optimized via data representation and quantization. They also present the first FPGA-based event detection accelerator based on the proposed model. Likewise, Bouguezzi et al. [5] introduce Ad-MobileNet, an advanced CNN model from the MobileNet model. Using an Ad-depth engine, an improved version of the depth-wise separable convolution unit, they achieved an efficient FPGA implementation.

Many works use data and model quantization in deep neural network (DNN) accelerators, but the following two papers experimented with extreme cases. Gao et al. [6] propose a binary neural network (BNN) inference accelerator suitable for FPGA with pruning the massive redundant operations while maintaining the original accuracy of the networks. Kwan and Nunez-Yanez [7] addressed the low-accuracy issue of BNN by adaptively choosing the number of frames used during inference, exploiting the high frame rates that binarized neural networks can achieve. It presents a novel entropy-based

Citation: Kim, J.-Y. Advanced AI Hardware Designs Based on FPGAs. *Electronics* **2021**, *10*, 2551. https://doi.org/10.3390/electronics10202551

Received: 11 October 2021
Accepted: 18 October 2021
Published: 19 October 2021

adaptive filtering technique that improves accuracy by varying the system's processing rate based on the entropy present in the neural network output.

Software programmability is the central issue in DNN accelerators, deciding the accelerator's adoption in the industry. Wu et al. [8] designed a reconfigurable and lightweight coprocessor of the RISC-V for better programmability than array-based accelerators. Gadea-Gironés et al. [9] utilized OpenCL, one of the most popular software frameworks in parallel computing. They propose an OpenCL-based design methodology for FPGAs that could catch up with the performance of GPU. Novickis et al. [10] developed a specialized tool to facilitate different accelerator implementations. It splits a feed-forward neural network into elementary layers, allocates computational resources, and generates high-level C++ descriptions for high-level synthesis (HLS).

Finally, we have a couple of non-trivial application works in this Special Issue. Alcolea and Resano [11] propose an accelerator architecture for gradient-boosting decision trees. Goswami and Bhatia [12] attempt to address the physical design problem of FPGA. They suggest a methodology for a post-placement and machine learning-based routing congestion prediction model.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Wang, Z.; Tang, Q.; Guo, B.; Wei, J.; Wang, L. Resource Partitioning and Application Scheduling with Module Merging on Dynamically and Partially Reconfigurable FPGAs. *Electronics* **2020**, *9*, 1461. [CrossRef]
2. Rios-Navarro, A.; Gutierrez-Galan, D.; Dominguez-Morales, J.P.; Piñero-Fuentes, E.; Duran-Lopez, L.; Tapiador-Morales, R.; Dominguez-Morales, M.J. Efficient Memory Organization for DNN Hardware Accelerator Implementation on PSoC. *Electronics* **2021**, *10*, 94. [CrossRef]
3. Li, S.; Sun, K.; Luo, Y.; Yadav, N.; Choi, K. Novel CNN-Based AP2D-Net Accelerator: An Area and Power Efficient Solution for Real-Time Applications on Mobile FPGA. *Electronics* **2020**, *9*, 832. [CrossRef]
4. Han, Z.; Jiang, J.; Qiao, L.; Dou, Y.; Xu, J.; Kan, Z. Accelerating Event Detection with DGCNN and FPGAs. *Electronics* **2020**, *9*, 1666. [CrossRef]
5. Bouguezzi, S.; Fredj, H.B.; Belabed, T.; Valderrama, C.; Faiedh, H.; Souani, C. An Efficient FPGA-Based Convolutional Neural Network for Classification: Ad-MobileNet. *Electronics* **2021**, *10*, 2272. [CrossRef]
6. Gao, J.; Liu, Q.; Lai, J. An Approach of Binary Neural Network Energy-Efficient Implementation. *Electronics* **2021**, *10*, 1830. [CrossRef]
7. Kwan, E.Y.L.; Nunez-Yanez, J. Entropy-Driven Adaptive Filtering for High-Accuracy and Resource-Efficient FPGA-Based Neural Network Systems. *Electronics* **2020**, *9*, 1765. [CrossRef]
8. Wu, N.; Jiang, T.; Zhang, L.; Zhou, F.; Ge, F. A Reconfigurable Convolutional Neural Network-Accelerated Coprocessor Based on RISC-V Instruction Set. *Electronics* **2020**, *9*, 1005. [CrossRef]
9. Gadea-Gironés, R.; Herrero-Bosch, V.; Monzó-Ferrer, J.; Colom-Palero, R. Implementation of Autoencoders with Systolic Arrays through OpenCL. *Electronics* **2021**, *10*, 70. [CrossRef]
10. Novickis, R.; Justs, D.J.; Ozols, K.; Greitāns, M. An Approach of Feed-Forward Neural Network Throughput-Optimized Implementation in FPGA. *Electronics* **2020**, *9*, 2193. [CrossRef]
11. Alcolea, A.; Resano, J. FPGA Accelerator for Gradient Boosting Decision Trees. *Electronics* **2021**, *10*, 314. [CrossRef]
12. Goswami, P.; Bhatia, D. Congestion Prediction in FPGA Using Regression Based Learning Methods. *Electronics* **2021**, *10*, 1995. [CrossRef]