

A 0.13 mJ/Prediction CIFAR-100 Fully Synthesizable Raster-Scan-Based Wired-Logic Processor in 16-nm FPGA

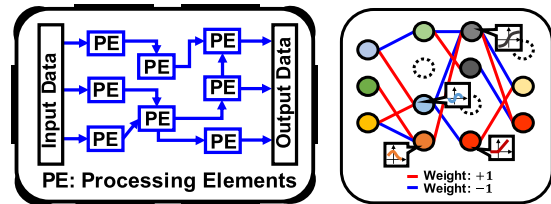
Dongzhu LI^{†a)}, Zhijie ZHAN[†], Rei SUMIKAWA[†], Mototsugu HAMADA[†], Atsutake KOSUGE[†], *Nonmembers,* and Tadahiro KURODA[†], *Fellow*

SUMMARY A 0.13mJ/prediction with 68.6% accuracy wired-logic deep neural network (DNN) processor is developed in a single 16-nm field-programmable gate array (FPGA) chip. Compared with conventional von-Neumann architecture DNN processors, the energy efficiency is greatly improved by eliminating DRAM/BRAM access. A technical challenge for conventional wired-logic processors is the large amount of hardware resources required for implementing large-scale neural networks. To implement a large-scale convolutional neural network (CNN) into a single FPGA chip, two technologies are introduced: (1) a sparse neural network known as a non-linear neural network (NNN), and (2) a newly developed raster-scan wired-logic architecture. Furthermore, a novel high-level synthesis (HLS) technique for wired-logic processor is proposed. The proposed HLS technique enables the automatic generation of two key components: (1) Verilog-hardware description language (HDL) code for a raster-scan-based wired-logic processor and (2) test bench code for conducting equivalence checking. The automated process significantly mitigates the time and effort required for implementation and debugging. Compared with the state-of-the-art FPGA-based processor, 238 times better energy efficiency is achieved with only a slight decrease in accuracy on the CIFAR-100 task. In addition, 7 times better energy efficiency is achieved compared with the state-of-the-art network-optimized application-specific integrated circuit (ASIC).

key words: *wired-logic, non-linear neural network, FPGA, high-level synthesis, software and hardware co-design*

1. Introduction

Artificial intelligence (AI) has become an integral part of our lives. Neural networks are a central technology in AI processing. In particular, convolutional neural networks (CNNs) achieve excellent recognition performance; thus, they are becoming a key technology for digital transformation. In applications such as industrial robots, autonomous driving, and self-checkout stores, CNN is generally processed in a power-constrained edge environment. Since these edge devices have strict power and heat constraints, energy-efficient CNN processors based on application-specific integrated circuits (ASICs) [1]–[15] are being developed. In recent years, research has been conducted to improve power efficiency by optimizing both the network structure and hardware architecture. For example, an ASIC optimized for hidden neural network (HNN) structure was presented at ISSCC'22 [1]. This hardware applies random numbers



(a) Wired-logic architecture processor (b) Non-linear neural network

Fig. 1 Wired-logic architecture-based processor using non-linear neural network (NNN).

as weight coefficients. This eliminates the need to store weight information and reduces memory usage and power consumption for memory access. While such network and task-optimized AI processors achieve good energy efficiency, they have high non-recurring engineering (NRE) costs because their applications are fixed. The HNN example [1] is dedicated to image classification tasks. Developing a large number of dedicated network-optimized AI processors to cover a wide range of applications requires high cost due to expensive photomask design and fabrication. On the other hand, field-programable-gate-arrays (FPGAs) can be customized for specific tasks without large NRE costs. Moreover, FPGA implementation only needs hardware description language (HDL) codes and post-HDL processes are highly automated by FPGA tools. Therefore, compared with ASIC development, FPGA implementation requires much lower design cost and less development time. Furthermore, along with high-level synthesis (HLS) technology, HDL coding can also be automated, and designers only need to write software codes like C++ or Python to implement AI models on FPGAs.

However, their energy efficiency is lower than ASICs due to the large capacitance of reconfigurable signaling wires and the large amount of leakage current from unused circuit blocks. Compared with ASIC-based implementation, conventional FPGA-based hardware is less energy-efficient by approximately two orders of magnitude [17].

In this work, a single-board FPGA-based raster-scan-based wired-logic processor is developed, which can process a 10-layer CNN for the CIFAR-100 task with 7 times better energy efficiency than the state-of-the-art ASIC [1]. All the processing elements (PEs) and signaling wires including weight information are implemented on the FPGA to eliminate memory access (Fig. 1 (a)). A technical challenge

Manuscript received June 12, 2023.

Manuscript revised October 4, 2023.

Manuscript publicized November 24, 2023.

[†]The authors are with Graduate School of Engineering, The University of Tokyo, Tokyo, 113–8656 Japan.

a) E-mail: ldzdongzhu@gmail.com

DOI: 10.1587/transele.2023LHP0001

of the wired-logic architecture is the large amount of hardware resources. Multi-board implementation is not a good way to achieve energy-efficient hardware because the power consumption for the board-to-board interface of FPGAs is much larger than that of the PEs [17]. To solve this problem, two technologies are used. One is a non-linear neural network (NNN) (Fig. 1 (b)). NNN presented in [18] achieves high expressive capability by individually optimizing the non-linear activation function of each neuron. As a result, even a 97% pruned ultra-sparse binary-weight neural network can achieve a high accuracy that is comparable to prior works [19]. The other is a newly proposed raster-scan wired-logic architecture. By reusing the same filter circuit in a time-shared manner, the amount of required hardware resources is reduced to 1/7.6, compared with the conventional wired-logic architecture [19]. The developed processor achieves a classification accuracy of 68.6% on the CIFAR-100 dataset and an energy efficiency of 0.13mJ/prediction. This energy efficiency is 238 times better than the state-of-the-art FPGA-based AI processor [20] and achieved with only a slight drop in accuracy. Moreover, even when compared with the state-of-the-art ASIC optimized for HNN [1], the proposed raster-scan wired-logic processor achieves 7 times better energy efficiency.

This paper is an extended version of the conference paper [31] with a new discussion about HLS tool chains to reduce the design cost of the wired-logic processor. The rest of this paper is organized as follows. In Sect. 2, we introduce related works regarding non-Von-Neumann processors and FPGA implementations. In Sects. 3 and 4, our proposed raster-scan-based wired-logic processor and HLS tool chains are presented, respectively. In Sect. 5, experimental results are shown and finally, we conclude this paper with the summary in Sect. 6.

2. Related Works

2.1 Non-von-Neumann ASIC Implementations

The most common approach for low energy consuming non-von-Neumann type AI hardware is to create ASICs. Energy efficiency can be maximized to the limit by designing ASICs with hardware structures optimized for the task to be performed.

A major non-von-Neumann ASIC architecture for AI processing is a wired-logic type implementation. Wired-logic architecture is a method in which a sufficient number of processing elements (PEs) are implemented on a chip. The PEs are then wired according to the order of tasks to be processed. Such a wired-logic type can be further divided into two types of implementation methods.

One is the neuromorphic processor [2]–[10], which uses a spiking neural network (SNN) model that mimics cerebellum neural activity and expresses signals using spikes. The SNN-based neuromorphic processor is actively studied using knowledge from the field of neuroscience. The other is a wired-logic processor that uses NNN, which expresses

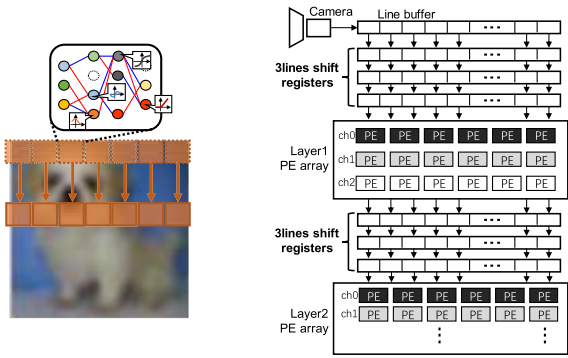
signals using digital values. NNN is a method that can build models with high accuracy using a small number of neurons and synapses. This is achieved by optimizing the activation function of each neuron individually. An example of ASIC design using NNN is reported Refs. [11] and [12].

A common technical challenge for the wired-logic architecture including neuromorphic processors is that they require a large amount of hardware for large-scale neural networks. In the example of the neuromorphic processor [2], it was necessary to use eight chips to implement the SNN which has sufficiently high accuracy on the CIFAR-10 and CIFAR-100 datasets. The chip-to-chip interface between ASICs is the new performance bottleneck due to the large power consumption and latency. The energy consumed by the chip-to-chip interface is more than two orders of magnitude larger than the energy consumed by the PEs in the neuromorphic processor [17]. To mitigate this problem, single chip implementations are needed for energy-efficient computing.

2.2 FPGA Implementations

While these methods based on ASIC implementation enable the reduction of energy consumption by designing task-optimized hardware, they lack post-implementation programmability and have high NRE costs. Therefore, a method of hardware implementation of neural networks on FPGAs, which can achieve low NRE costs, is being studied [18]–[24]. Methods are being studied to overcome the drawback that FPGA-based hardware has difficulties in achieving low energy consumption compared to ASICs. Reference [20] conducted structured pruning of CNN filters to reduce the energy consumption of external DRAM accesses. A wired-logic and NNN-based FPGA implementation, which is similar to the ASIC design method in [11], [12], has also achieved the reduction of energy consumption [21]. A shift-register-based wired-logic architecture has been reported in [19], which can implement multiple CNN layers onto a single FPGA board. In CNNs, the filter slides by one pixel at a time from the top left corner of the input image, and the processing proceeds in a raster-scan fashion until it reaches the bottom right corner. In conventional wired-logic processors, all these raster scan operations are loop-unrolled in the horizontal direction of the image or both vertical and horizontal directions. In a shift-register-based wired-logic implementation, all PEs are implemented in parallel as shown in Fig. 2. Such conventional loop-unrolled wired-logic architecture enables short latency and good energy efficiency.

However, this implementation has an issue with the huge hardware implementation size for large-scale neural networks. When implementing a CNN which has a large number of channels, the hardware amount increases rapidly. Therefore, in the conventional work [19], the number of channels is limited to 64 for a 14-layer CNN for the CIFAR-10 dataset. For more difficult tasks, such as classifying 100 classes of images in the CIFAR-100 dataset, more than 200 channels would be required. The hardware size would in-



(a) Neural-network operation (b) Width-parallel wired-logic processor

Fig. 2 Conventional loop-unrolled wired-logic architecture.

crease about nine times. Since the 64-channel 14-layer CNN for CIFAR-10 uses 1.3M lookup tables (LUTs), 11.7M LUTs would be needed when the number of channels increases beyond 200. This number far exceeds the size of available FPGAs on the market. Therefore, multiple FPGA boards are required for the implementation. As mentioned in [17], the board-to-board interface is not suitable for implementations pursuing low energy consumption due to its poor energy efficiency, so further circuit area reduction is required.

3. Raster-Scan-Based Wired-Logic Processor

NNN [19] is a kind of binary neural network where the weight coefficients are quantized to +1/-1. NNN [19] can achieve high recognition accuracy even with binary weight coefficients and an ultrasparse structure by optimizing the nonlinear function to various shapes for each channel through training. The raster-scan-based wired-logic processor using NNN developed in this work is shown in Fig. 3. A line buffer that can store one line of image data (a feature map) is connected to other line buffers and configured as a large shift register. In this work, the filter kernel size is set to 3×3 for all convolutional layers. Therefore, the shift register of all convolutional layers can be regarded as three lines of line buffers connected in series. The data for the filter kernel is taken out from this shift register and output to each PE (Fig. 4). By inserting data shifted by one pixel in every clock cycle, a convolution operation with Stride=1 can be realized. The shift register is connected to each PE by synapse wiring that reflects the training results of the neural network (NNN). If the weight coefficient is +1 as the result of training, it is connected to the positive input of the PE (Fig. 4); and if the weight coefficient is -1, it is connected to the negative input of the PE. Each PE consists of adders and a lookup table (LUT). As well as the conventional method [19], non-linear functions of NNN are implemented using FPGA 6-LUTs.

Compared with the conventional loop-unrolled wired-logic architecture (Fig. 2 (b)), while the proposed raster-scan architecture increases the number of clock cycles for processing due to the reduced parallelism, the hardware size is reduced to the inverse of the width of the data. For example, for the first convolutional layer, the input and output

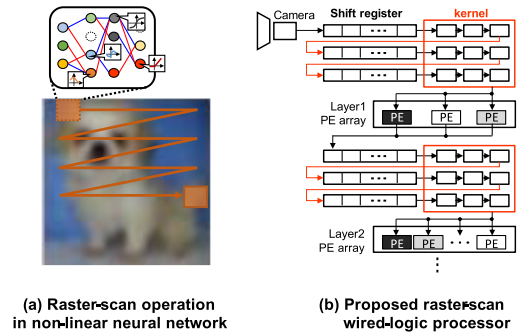


Fig. 3 Proposed raster-scan-based wired-logic architecture.

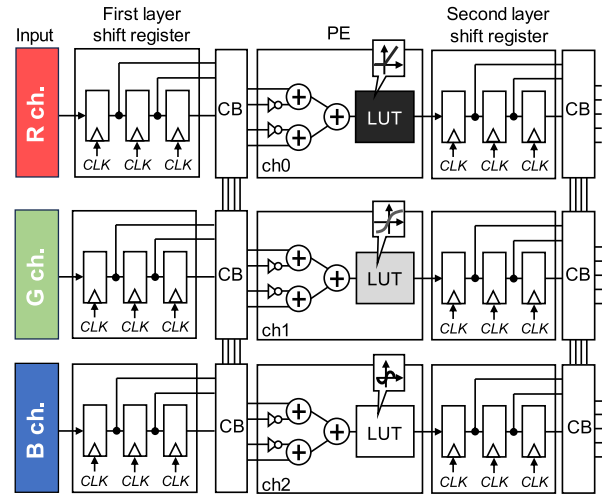


Fig. 4 Detailed processing element implementation of the neural network.

data are 32×32 pixels in the CIFAR-100 dataset. Therefore, applying the raster-scan wired-logic architecture increases the number of clock cycles by 32 times, but reduces the number of PEs required by a factor of 1/32. The larger the data size, the greater the effect of hardware usage reduction. Therefore, the effect of hardware usage reduction decreases in the later layers beyond the pooling layers where the feature map becomes smaller. Since the reduction effect is large in the first several layers with a large feature map, a sufficient hardware resource reduction effect can be achieved. The challenge with the raster-scan wired-logic processor is the zero-padding operation in convolution layers. In the conventional loop-unrolled architecture, where the circuit needs to insert zeros is always fixed and so it does not require complicated control circuit. In the proposed raster-scan architecture, it is necessary to implement zero padding according to the network structure. A counter is placed to check the number of pixels applied to the circuit. After one line of data has been input to the shift register, two pixels of zeros are inserted. When it detects that data for all lines have been input, it inserts zero data for two lines. All such controllers are implemented with a state machine based on a counter. The clock is applied synchronously with data, and the clock frequency is divided by 2 using a divider at the pooling layer.

4. High-Level Synthesis Tool for Raster-Scan-Based Wired-Logic Processor

To accelerate the development process of the raster-scan wired-logic processor, we have developed a HLS tool that can directly translate CNN models from PyTorch to Verilog. This allows us to quickly reflect any design changes to the neural networks made in PyTorch in the post-synthesis hardware design. Moreover, our HLS tool has the added capability of generating a test-bench in System Verilog that (1) reads an arbitrary input image from the CIFAR-10 and CIFAR-100 datasets, (2) performs inference using both PyTorch and the generated HDL code, and (3) validates the result of inference by comparing the output feature map (OFM) values of individual layers. This comprehensive functionality of our HLS tool has the advantage of greatly saving time and effort in debugging and validating the wired-logic processor.

Although commercially available HLS tools such as Xilinx Vitis [27] and Vitis AI [28] are strong at quick and high-performance FPGA deployment for many applications, they fall short for our usage because Ref. [27] does not support PyTorch and Ref. [28] is optimized for the on-board Deep Learning Processor Units (DPUs) in Xilinx products rather than FPGAs. On the other hand, recent HLS research works [29], [30] have come to support PyTorch in their frameworks as shown in Fig. 5 (a). The translation from PyTorch to HDL is achieved with Multi-Level Intermediate Representation (MLIR) and hence these tools support most of the existing PyTorch modules. However, to support this wide range of PyTorch modules using MLIR, the designer needs to manually plan dataflow and scheduling ahead. Furthermore, no previous HLS tools can generate HDL test-bench from PyTorch dataset.

The proposed raster-scan-based wired-logic architecture enables the implementation of a large-scale NNN on a single chip. As shown in Fig. 5 (b), this eliminates the manual labor for dataflow design and scheduling because there is no memory access during inference. This is achieved by the Python code of the NNN model training results, which already contains the entire data flow, shift registers, and PE array scheduling of each layer.

Hence, the design process using our tool is faster compared to conventional HLS tools. The proposed HLS tool, while specifically designed for wired-logic architecture, offers the flexibility of generating the required hardware design code for various networks.

With the proposed HLS tool, we manage to unify (1) the generation of new NNN models in HDL, (2) the generation of a new testbench in HDL, and (3) automated testing of all models with one simple script. For model generation, the proposed HLS tool can translate HDL code for (a) convolution (CONV), (b) max pooling, and (c) fully connected (FC) layers. For each type of layer, the calculation is element-wise unrolled in Verilog with appropriate weight data hard-coded as wired-logic. The bit width of input and output are assigned by the quantized network accordingly.

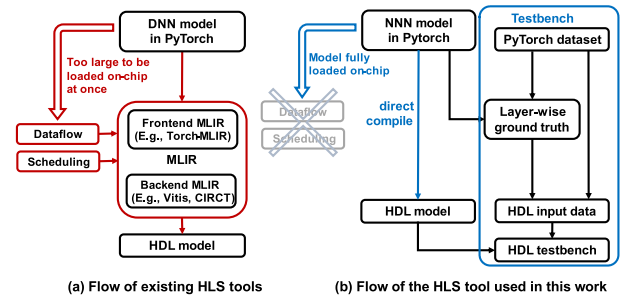


Fig. 5 Comparison of high-level synthesis flows between (a) previous works and (b) this work.

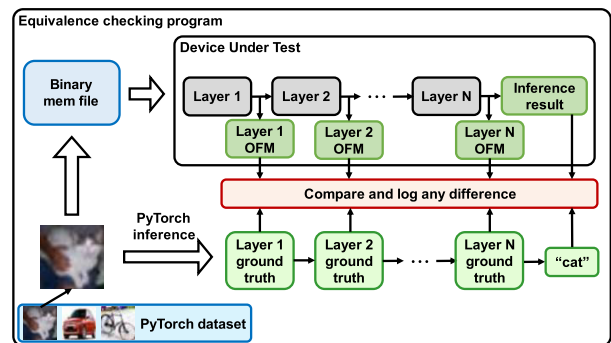


Fig. 6 Conceptual sketch of the equivalence checking program.

The testbench generated by the proposed HLS tool verifies the output feature map (OFM) values of each layer and identifies any incorrect layers, as shown in Fig. 6. Automating this process is necessary because the bit width, filter size, and input image size at any layer may change during development. Since HDLs are not suitable for handling this dynamic typing nature, we utilize HLS in Python to dynamically generate new HDL testbenches. First, a ground truth file is generated by passing an arbitrary image from the dataset through the NNN in PyTorch. All intermediate OFM values are recorded as expected answers for the HLS-generated model. These values are then converted into binaries and hardcoded as LUTs to be accessed by the testbench program. Next, a binary file is created using the image from PyTorch as input data. Finally, a testbench program is generated in SystemVerilog, incorporating updated shape and bit width information obtained from the NNN model. The testbench program performs the following tasks: (1) reads the file for the generated HDL model, (2) compares the OFM values between simulation and ground truth, and (3) records any differences in a log.

By allowing quick evaluation and revision of the initial CNN model at the hardware level, the proposed approach significantly enhances the efficiency of testing new models.

This enables AI model designers to focus more on innovative and creative tasks.

5. Experiment Results

5.1 Raster-Scan Wired-Logic Processor Using NNN

As shown in Fig. 7, a 10-layer VGG-like CNN that consists of 6 convolution layers, 3 max-pooling layers, and 1 fully connected layer was used as the initial structure to train the NNN. Eight implementations of the networks having different numbers of channels were trained, and all have 80% of their synapses pruned. Compared to conventional CNNs, the individual optimization of the nonlinear activation function in each neuron during backpropagation within the NNN enables smaller-scale and highly pruned neural networks to achieve a recognition accuracy similar to that of much larger CNNs. The resulting recognition accuracy ranges from approximately 63.5% to 69.3%, showcasing only a minor reduction in accuracy compared to that of state-of-the-art CNN processors [1], [20].

The proposed area-efficient raster-scan-based wired-logic processor was implemented on the Xilinx UltraScale+

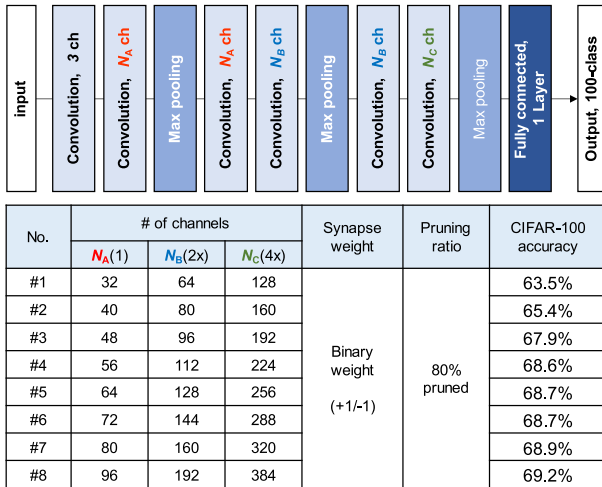


Fig. 7 NNN implementations and their training result on CIFAR-100 dataset.

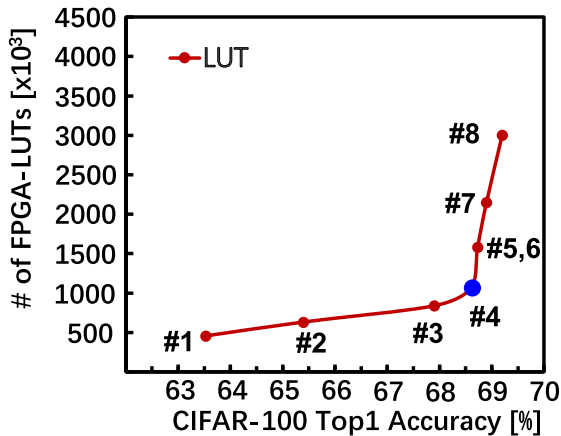


Fig. 8 Trade-off between hardware resource and accuracy of NNN.

Virtex series FPGA. Figure 8 summarizes the implementation results, and the trade-off between recognition accuracy and hardware usage was verified. In the NNN with a large number of channels, the number of adders and signaling wires increases with the number of channels, so the hardware usage (LUTs) increases rapidly. On the other hand, because the hardware usage increases non-linearly with the recognition accuracy of CIFAR-100, there is a point at which the hardware resource is used most efficiently. In this case, implementation #4 provides the best hardware usage efficiency. The relationship between hardware usage and recognition accuracy depends on the initial neural network structure. In this work, implementation #1 has a recognition accuracy that is closest to the recognition accuracy of previous research [2], [16], and implementation #4 shows only a slight drop in recognition accuracy compared to previous works [1], [20].

Using implementations #1 to #4 models, the total amount of hardware resources used in the conventional method [19] is compared with the proposed raster-scan-based wired-logic architecture as shown in Fig. 9. In this experiment, the conventional loop-unrolled wired-logic architecture uses a larger amount of hardware as the accuracy increases because the number of channels increases. The hardware resources required for the raster-scan architecture also increase, but the increase is much smaller than the loop-unrolled architecture. When implementing a 68.6% accuracy NNN for the CIFAR-100 dataset, the number of LUTs required is reduced from 8.04M LUTs (1) to 1.06M LUTs (1/7.6).

5.2 Performance Comparison

Figure 10 shows the details of the #1, #4, and #8 implementation results. They were implemented using Xilinx’s Virtex series, VU9P and VU19P respectively. The clock frequency for all cases was set to 100 MHz. Notice that all implementations employ wired-logic architecture, so there is no BRAM or DSP. The power consumption is 4.5W, 11.0W and 20.4W, respectively, including leakage power.

A commercial general-purpose neural processing unit (NPU) with high energy efficiency through cutting-edge pro-

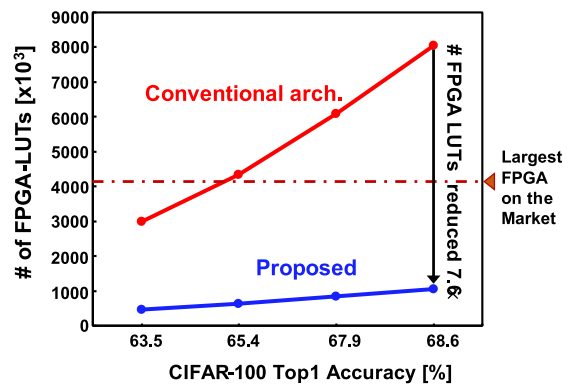
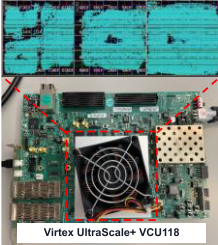


Fig. 9 Hardware resource utilization comparison on the CIFAR-100 dataset.



Model name	#1	#4	#8
Accuracy [%]	63.5%	68.6%	69.2%
FPGA platform	VU9P	VU19P	VU19P
# of LUTs	459,462	1,062,095	3,001,633
# of FFs	44,842	65,582	186,199
# of DSPs	0	0	0
BRAM	0	0	0
Power[W]	4.5	11.0	20.4
Clock frequency [MHz]	100	100	100
Throughput [Kfps]	86.5	86.5	86.5
Energy Efficiency [$\mu\text{J}/\text{frame}$]	52	127.2	235.8

Fig. 10 FPGA implementation results on the CIFAR-100 dataset.

Table 1 Comparison with the state-of-the-art FPGA implementations.

	Ref. [20]	This work (#4)	This work (#8)
Data set	CIFAR-100		
Accuracy [%]	70.30	68.64	69.20
Platform	Stratix-V FPGA	Virtex UltraScale+ XCVU19P	Virtex UltraScale+ XCVU19P
Network model	CNN	NNN	NNN
# of LUTs	N/A	1,062,095	3,001,633
# of FFs	N/A	65,582	186,199
Clock freq. [MHz]	100	100	100
Throughput [Kfps]	6.5×10^{-2}	86.5	86.5
GOPS/frame	0.89	1.43 (before pruning) 0.29 (after pruning)	3.69 (before pruning) 0.74 (after pruning)
Power [W]	2.0	11.0	20.4
Energy [mJ/frame]	31 (1)	0.13 (1/238)	0.24 (1/129)

Table 2 Comparison with the state-of-the-art ASIC implementation.

	Ref. [16]	This work (#1)	Ref. [1]	This work (#4)	This work (#8)
Data set	CIFAR-100				
Accuracy [%]	55.36	63.50	70.15	68.64	69.20
Platform	Synopsys HSPICE Tool with 45nm CMOS	Virtex UltraScale+ XCVU13P	Digital ASIC 40nm CMOS	Virtex UltraScale+ XCVU19P	Virtex UltraScale+ XCVU19P
Network model	CNN	NNN	HNN	NNN	NNN
# of LUTs	-	459,462	-	1,062,095	3,001,633
# of FFs	-	44,842	-	65,582	186,199
Clock freq. [MHz]	N/A	100	147	100	100
Throughput [Kfps]	N/A	86.5	9.6×10^{-2}	86.5	86.5
GOPS/frame	N/A	0.57 (before pruning) 0.11 (after pruning)	52.5	1.43 (before pruning) 0.29 (after pruning)	3.69 (before pruning) 0.74 (after pruning)
Power [W]	N/A	4.5	0.09	11.0	20.4
Energy [mJ/frame]	0.09 (1)	0.05 (1/1.8)	0.9 (1)	0.13 (1/7)	0.24 (1/3.75)

cess nodes has been proposed [32]. However, realizing the full extent of its theoretical performance is not always attainable. Depending on the specific neural networks or tasks, power efficiency will fall short of these theoretical expectations. In contrast, the state-of-the-art network-optimized ASIC [1] has provided superior performance when compared to commercial NPUs. To conduct a comprehensive assessment of energy efficiency, we compared our proposed approach to the ASIC [1] using the CIFAR-100 dataset, including system-level implementation from image input to classification. Furthermore, we made comparisons with the state-of-the-art FPGA-based AI accelerator [20]. It is worth noting that in all cases presented in Table 1 and Table 2, the input size of network is consistently 32×32 pixels, owing to the use of the CIFAR-100 dataset. The energy efficiency of the #1 implementation is $52 \mu\text{J}/\text{prediction}$. Compared with the previous work [16], the recognition accuracy is higher by 8 percentage points, and the energy efficiency is 1.8 times better. The #4 implementation has an energy efficiency of $127 \mu\text{J}/\text{prediction}$, which is 238 times better than that of an FPGA-based accelerator [20] with only a small decrease in

recognition accuracy as shown in Table 1. Compared with the state-of-the-art network-optimized ASIC [1], the energy efficiency is 7 times better. In addition, the energy consumption for model #8 is $235.8 \mu\text{J}/\text{prediction}$, which represents a 129 times improvement over the state-of-the-art FPGA-based accelerator [20] and a 3.75 times improvement compared to the network-optimized ASIC [1]. The most energy-efficient AI processor ever for the CIFAR-100 image classification task has been realized as shown in Table 2. Meanwhile, the experimental results presented above indicate that by using the proposed method, FPGAs can achieve superior energy efficiency when compared to ASICs.

6. Conclusion

An area-efficient raster-scan-based wired-logic processor that realizes high energy efficiency and recognition accuracy is presented. A CIFAR-100 NNN model with a recognition accuracy of 68.64% was implemented on a single FPGA. The proposed wired-logic processor processes the image data with a high efficiency of $0.13 \text{mJ}/\text{prediction}$. Compared with a state-of-the-art FPGA-based CNN accelerator with high recognition accuracy, the energy efficiency is improved by two orders of magnitude. Even compared with a state-of-the-art network-optimized ASIC AI processor, the energy efficiency is improved by 7 times, all while incurring only a minor dip in accuracy. In addition, the design cycle is greatly reduced through the proposed HLS tool, and testbench code can be automatically generated to facilitate debugging.

Our research seeks to establish a solid foundation for NNN and raster-scan wired-logic architecture, and our network and hardware architecture can be easily expanded to larger deep neural networks for different applications. NNN training can be carried out using various larger network models or architectures as the initial structure, depending on the user's application such as keyword spotting task [33]. On the other hand, depending on the size of the input data, the length of shift registers within the hardware architecture varies accordingly. Furthermore, the number of channels and PEs in raster-scan wired-logic architecture changes depending on the implemented neural network model. However, for highly complex tasks such as ImageNet classification, when employing the VGG13-base model with a maximum of 512 channels in the convolutional layers, the required hardware resources (LUTs) increase to 1.5×10^7 . Even though benefiting from NNN, achieving high accuracy with fewer than 512 channels in the convolutional layers is feasible, the required hardware resources still exceed the maximum capacity of currently available commercial FPGAs on the market. This limitation implies that there exists an upper limit for scalability to larger networks in the hardware implementation with a single FPGA. To address the challenge, further research into novel circuit technologies is necessary.

Acknowledgments

This work was supported by JST, PRESTO grant number

JPMJPR21B4, Japan.

References

- [1] K. Hirose, J. Yu, K. Ando, Á.L. García-Arias, J. Suzuki, T. Van Chu, K. Kawamura, and M. Motomura, "Hiddenite: 4K-PE Hidden Network Inference 4D- Tensor Engine Exploiting On-Chip Model Construction Achieving 34.8-to-16.0TOPS/W for CIFAR-100 and ImageNet," IEEE International Solid-State Circuits Conference (ISSCC), Dig. Tech. Papers, San Francisco, CA, USA, pp.252–253, Feb. 2022.
- [2] S.K. Esser, P.A. Merolla, J.V. Arthur, A.S. Cassidy, R. Appuswamy, A. Andreopoulos, D.J. Berg, J.L. McKinstry, T. Melano, D.R. Barch, C. di Nolfo, P. Datta, A. Amir, B. Taba, M.D. Flickner, and D.S. Modha, "Convolutional networks for fast, energy-efficient neuromorphic computing," Proc. National Academy of Sciences of the United States of America (PNAS), vol.113, no.41, pp.11441–11446, Oct. 2016.
- [3] J. Pu, W.L. Goh, V.P. Nambiar, M.M. Wong, and A.T. Do, "A 5.28-mm² 4.5-pJ/SOP Energy-Efficient Spiking Neural Network Hardware With Reconfigurable High Processing Speed Neuron Core and Congestion-Aware Router," IEEE Trans. Circuits Syst. I, Reg. Papers, vol.68, no.12, pp.5081–5094, Dec. 2021.
- [4] A. Shukla, V. Kumar, and U. Ganguly, "A Software-equivalent SNN Hardware using RRAM-array for Asynchronous Real-time Learning," 2017 International Joint Conference on Neural Network (IJCNN), Anchorage, AK, USA, pp.4657–4664, 2017.
- [5] J. Luo, L. Yu, T. Liu, M. Yang, Z. Fu, Z. Liang, L. Chen, C. Chen, S. Liu, S. Wu, Q. Huang, and R. Huang, "Capacitor-less Stochastic Leaky-FeFET Neuron of Both Excitatory and Inhibitory Connections for SNN with Reduced Hardware Cost," 2019 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, USA, pp.6.4.1-6.4.4, 2019.
- [6] D.-A. Nguyen, X.-T. Tran, and F. Iacopi, "A review of algorithms and hardware implementations for spiking neural networks," Journal of Low Power Electronics and Applications, vol.11, no.2, p.23, May 2021.
- [7] Y. Jang, G. Kang, T. Kim, Y. Seo, K.-J. Lee, B.-G. Park, and J. Park, "Stochastic SOT device based SNN architecture for On-chip Unsupervised STDP Learning," IEEE Trans. Comput., vol.71, no.9, pp.2022–2035, Oct. 2021.
- [8] A.R. Young, M.E. Dean, J.S. Plank, and G.S. Rose, "A Review of Spiking Neuromorphic Hardware Communication Systems" IEEE Access, vol.7, pp.135606–135620, Sept. 2019.
- [9] J. Schemmel, D. Brüderle, A. Grübl, M. Hock, K. Meier, and S. Millner, "A wafer-scale neuromorphic hardware system for large-scale neural modeling," IEEE International Symposium on Circuits and Systems (ISCAS), Paris, France, pp.1947–1950, 2010.
- [10] B. Rajendran, A. Sebastian, M. Schmuker, N. Srinivasa, and E. Eleftheriou, "Low-Power Neuromorphic Hardware for Signal Processing Applications: A review of architectural and system-level design approaches," IEEE Signal Process. Mag., vol.36, no.6, pp.97–110, Nov. 2019.
- [11] R. Sumikawa, K. Shiba, A. Kosuge, M. Hamada, and T. Kuroda, "A 1.2nJ/Classification 2.4mm² Wired-Logic Neuron Cell Array Using Logically Compressed Non-Linear Function Blocks in 0.18μm CMOS," JSAP International Conference on Solid State Devices and Materials, Chiba, Japan, pp.750–751, 2022.
- [12] R. Sumikawa, K. Shiba, A. Kosuge, M. Hamada, and T. Kuroda, "1.2 nJ/classification 2.4 mm² asynchronous wired-logic DNN processor using synthesized nonlinear function blocks in 0.18 μm CMOS," Japanese Journal of Applied Physics, vol.62, no.SC, p.SC1019, Jan. 2023.
- [13] D. Bankman, L. Yang, B. Moons, M. Verhelst, and B. Murmann, "An Always-On 3.8 μJ/86% CIFAR-10 mixed-signal binary CNN processor with all memory on chip in 28-nm CMOS," IEEE J. Solid-State Circuits, vol.54, no.1, pp.158–172, Oct. 2018.
- [14] Y.H. Chen, T. Krishna, J.S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," IEEE J. Solid-State Circuits, vol.52, no.1, pp.127–138, Nov. 2016.
- [15] K. Ando, K. Ueyoshi, K. Orimo, H. Yonekawa, S. Sato, H. Nakahara, S. Takamaeda-Yamazaki, M. Ikebe, T. Asai, T. Kuroda, and M. Motomura, "BRein memory: A single-chip binary/ternary reconfigurable in-memory deep neural network accelerator achieving 1.4 TOPS at 0.6 W," IEEE J. Solid-State Circuits, vol.53, no.4, pp.983–994, Dec. 2017.
- [16] T. Luo, L. Yang, H. Zhang, C. Qu, X. Wang, Y. Cui, W.-F. Wong, and R.S.M. Goh, "NC-Net: Efficient Neuromorphic Computing Using Aggregated Subnets on a Crossbar-Based Architecture With Non-volatile Memory," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol.41, no.9, pp.2957–2969, Sept. 2022.
- [17] M. Horowitz, "Computing's energy problem (and what we can do about it)," IEEE International Solid-State Circuits Conference (ISSCC), Dig. Tech. Papers, San Francisco, CA, USA, pp.10–14, 2014.
- [18] A. Kosuge, Y.-C. Hsu, M. Hamada, and T. Kuroda, "A 0.61-μJ/Frame Pipelined Wired-logic DNN Processor in 16-nm FPGA Using Convolutional Non-Linear Neural Network," IEEE Open Journal of Circuits and Systems, vol.3, pp.4–14, Jan. 2022.
- [19] Y.-C. Hsu, A. Kosuge, R. Sumikawa, K. Shiba, M. Hamada, and T. Kuroda, "A 13.7μJ/prediction 88% Accuracy CIFAR-10 Single-Chip Wired-logic Processor in 16-nm FPGA using Non-Linear Neural Network," IEEE Hot Chips Symposium (HCS), Cupertino, CA, USA, pp.1–14, 2022.
- [20] S. Moon, H. Lee, Y. Byun, J. Park, J. Joe, S. Hwang, S. Lee, and Y. Lee, "FPGA-Based Sparsity-Aware CNN Accelerator for Noise-Resilient Edge-Level Image Recognition," IEEE Asian Solid-State Circuits Conference (A-SSCC), Macau, Macao, pp.205–208, 2019.
- [21] A. Kosuge, M. Hamada, and T. Kuroda, "A 16 nJ/Classification FPGA-based Wired-Logic DNN Accelerator Using Fixed-Weight Non-Linear Neural Net," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol.11, no.4, pp.751–761, Dec. 2021.
- [22] R. Zhao, W. Song, W. Zhang, T. Xing, J.-H. Lin, M. Srivastava, R. Gupta, and Z. Zhang, "Accelerating Binarized Convolutional Neural Networks with Software-Programmable FPGAs" Proc. ACM International Symposium on Field-Programmable Gate Arrays (FPGA), New York, NY, USA, pp.15–24, Feb. 2017.
- [23] Q. Xiao, Y. Liang, L. Lu, S. Yan, and Y.-W. Tai, "Exploring heterogeneous algorithms for accelerating deep convolutional neural networks on FPGAs," Proc. 54th Annual Design Automation Conference, Austin, TX, USA, pp.1–6, June 2017.
- [24] L. Gong, C. Wang, X. Li, H. Chen, and X. Zhou, "MALOC: A fully pipelined FPGA accelerator for convolutional neural networks with all layers mapped on chip," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol.37, no.11, pp.2601–2612, July 2018.
- [25] A. Gaier and D. Ha, "Weight agnostic neural networks," Advances in Neural Information Processing Systems, Vancouver, BC, Canada, pp.5364–5378, June 2019.
- [26] Y.-C. Hsu, A. Kosuge, R. Sumikawa, K. Shiba, M. Hamada, and T. Kuroda, "A Fully Synthesized 13.7μJ/prediction 88% Accuracy CIFAR-10 Single-Chip Data-Reusing Wired-Logic Processor Using Non-Linear Neural Network," 28th Asia and South Pacific Design Automation Conference (ASP-DAC'23), pp.182–183, Jan. 2023.
- [27] X. Inc., Vitis High-Level Synthesis User Guide: UG1399 (v2022.2), 2022.
- [28] X. Inc., Vitis AI User Guide: UG414 (v3.0), 2023.
- [29] H. Ye, C. Hao, J. Cheng, H. Jeong, J. Huang, S. Neuendorffer, and D. Chen, "ScaleHLS: A New Scalable High-Level Synthesis Framework on Multi-Level Intermediate Representation," 2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA), pp.741–755, April 2022.
- [30] M. Urbach and M.B. Petersen, "HLS from PyTorch to System Verilog with MLIR and CIRCT," 2022 Workshop on Languages, Tools, and Techniques for Accelerator Design (LATTE), March 2022.

- [31] D. Li, Y.-C. Hsu, R. Sumikawa, A. Kosuge, M. Hamada, and T. Kuroda, "A 0.13mJ/Prediction CIFAR-100 Raster-Scan-Based Wired-Logic Processor Using Non-Linear Neural Network," IEEE International Symposium on Circuits and Systems (ISCAS), May 2023.
- [32] J.-S. Park, C. Park, S. Kwon, H.-S. Kim, T. Jeon, Y. Kang, H. Lee, D. Lee, J. Kim, Y.J. Lee, S. Park, J.-W. Jang, S.H. Ha, M.S. Kim, J. Bang, S.H. Lim, and I. Kang, "A Multi-Mode 8K-MAC HW-Utilization-Aware Neural Processing Unit with a Unified Multi-Precision Datapath in 4nm Flagship Mobile SoC," IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, pp.246–248, Feb. 2022.
- [33] A. Kosuge, R. Sumikawa, Y. -C. Hsu, K. Shiba, M. Hamada, and T. Kuroda, "A 183.4nJ/inference 152.8 μ W Single-Chip Fully Synthesizable Wired-Logic DNN Processor for Always-On 35 Voice Commands Recognition Application," IEEE Symposium on VLSI Circuits, June 2023.



Dongzhu Li received the B.S. degree in information and electrical engineering from China Agricultural University, Beijing, China in 2020. He is currently pursuing the M.S. degree in electrical engineering and information systems at The University of Tokyo, Tokyo, Japan. His research interests include energy-efficient hardware and software co-design for emerging AI processing.



Zhijie Zhan received the B.S. degree in information and communication engineering from The University of Tokyo, Tokyo, Japan in 2023. He is currently pursuing the M.S. degree in electrical engineering and information systems at The University of Tokyo, Tokyo, Japan. His research interests include energy-efficient AI processing, circuit design with emerging technologies, and heterogeneous architecture.



Rei Sumikawa received the B.S. degree in electronics and electrical engineering from The University of Tokyo, Tokyo, Japan in 2022. He is currently pursuing the M.S. degree in electrical engineering and information systems at The University of Tokyo, Tokyo, Japan. His current research interests include energy-efficient computing, hardware architecture and circuit design.



Professor of Systems Design Lab (d.lab).

Mototsugu Hamada received the B.S., M.S., and Ph.D. degrees in electronic engineering from the University of Tokyo, Tokyo, Japan, in 1991, 1993, and 1996, respectively. In 1996, he joined Toshiba Corporation and has been engaged in wireless and low power electronic circuits design with Center for Semiconductor Research and Development, Kawasaki, Japan. In 2016, he joined Keio University and was a Project Professor. In 2020, he joined the University of Tokyo, where he is currently a Project



Atsutake Kosuge received the Ph.D. degree from Keio University in Yokohama, Japan, in 2016. From 2017 to 2020, he held research positions at Hitachi Ltd. and Sony Corporation. In 2021, he joined The University of Tokyo, where he is currently an Assistant Professor of Systems Design Lab (d.lab). His research interests include energy efficient computing, computational sensing, and 3-D integration technologies.



IEICE Fellow, and a chair of VLSI Symposia.

Tadahiro Kuroda received the Ph.D. degree in electrical engineering from the University of Tokyo. In 1982, he joined Toshiba Corporation. He left Toshiba to join Keio University in 2000, and became a full professor in 2002. He was a visiting researcher from 1988 to 1990 and the Mackay Professor in 2007 at the University of California, Berkeley. He has been a professor at the University of Tokyo since 2019. He is the director of Systems Design Lab (d.lab) and the chairman of RaaS. He is an IEEE Fellow, an