

# A 16nm Fully Integrated SoC for Hardware-Aware Neural Architecture Search

Yu-Cheng Lin<sup>1</sup>, Ming-Shan Huang<sup>1</sup>, Jeng-Bang Wang<sup>1</sup>, Wen-Ching Chen<sup>2</sup>, Nian-Shyang Chang<sup>2</sup>,  
Chun-Pin Lin<sup>2</sup>, Chi-Shi Chen<sup>2</sup>, Tzi-Dar Chiueh<sup>1</sup>, and Chia-Hsiang Yang<sup>1</sup>

<sup>1</sup>National Taiwan University, Taipei, Taiwan <sup>2</sup>Taiwan Semiconductor Research Institute, Hsinchu, Taiwan

**Abstract**—Neural architecture search (NAS) is a technique that can automatically design and optimize neural network architectures. It aims to find a better balance between AI performance and hardware efficiency, at the cost of excessively high computational complexity. This work presents the *first* fully integrated system-on-chip (SoC) specialized for accelerating hardware-aware NAS. The SoC enables efficient exploration on diverse network architectures in the accuracy-latency space. It supports commonly-used networks, including convolutional neural network (CNN), recurrent neural network (RNN), and Transformer. Fabricated in 16nm FinFET, the chip dissipates 255mW at a clock frequency of 500MHz from a 0.8V supply. Compared to an NVIDIA A40 GPU, this work achieves a 27× speedup at a 2.6× lower clock frequency, given 1176× less power and 166× smaller silicon area.

## I. INTRODUCTION

The rapid advancement in AI has driven diverse applications, including image analysis and natural language processing. Neural networks lay the foundation of AI and it is critical to reduce the network size while maintaining AI performance. Conventionally, searching for such a compact network requires considerable expertise and trials. NAS has emerged to explore the feasible network topologies automatically [1]. Additionally, hardware-aware NAS enables to consider hardware metrics during the search process.

NAS-generated networks have been demonstrated to achieve higher accuracy with smaller network size than human-crafted designs for major networks [2], including CNN, RNN, and Transformer. Fig. 1 shows the three main components of NAS: search space, performance evaluation, and optimization algorithm [1]. A search space is first created to encompass all feasible network architectures. A network’s performance can be evaluated using a multi-objective reward. A learning-based optimization algorithm then selects a network architecture given the design constraints. This process continues until a termination condition is met, yielding an optimized network architecture for the target performance. Once the network architecture is selected, training is conducted to determine the network weights.

However, the expanded search space with more criteria and joint optimization goals introduces significantly high computational demand, requiring  $10^2$  to over  $10^4$  GPU hours even for image classification [2], as shown in Fig. 2. Hardware acceleration is a promising solution, but it asks for full integration of efficient search algorithm and flexible learning accelerator architecture. To the best of our knowledge, dedicated accelerators for NAS have not been demonstrated on

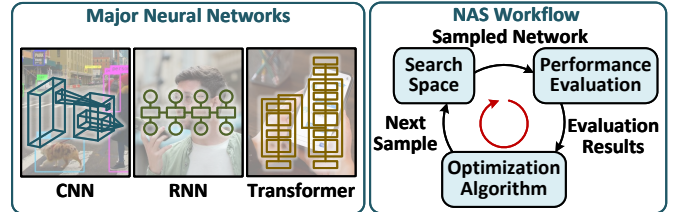


Fig. 1: Main components of NAS for major networks.

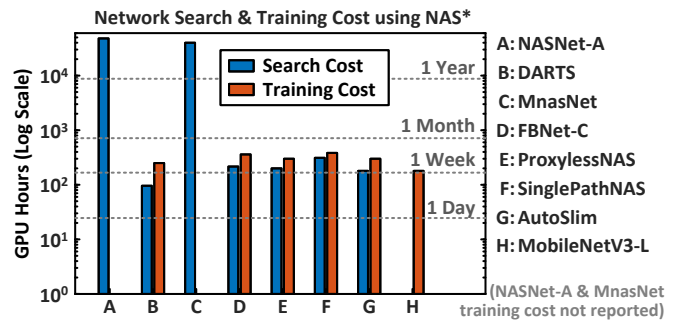


Fig. 2: Design issues for NAS (adapted from [2]).

silicon. This work presents the first fully integrated SoC for hardware-aware NAS. Compared to an A40 GPU, the chip achieves a 27× speedup at a 2.6× lower clock frequency with much lower implementation cost in both power and area.

The remainder of this paper is organized as follows. Section II shows the workflow of NAS and algorithm mapping. Section III describes the architecture of the proposed SoC. Chip implementation and performance evaluation are elaborated in Section IV. Finally, Section V concludes this paper.

## II. NAS WORKFLOW

Fig. 3 shows the adopted NAS workflow, based on the framework in [1]. Designs associated with NAS components from [3]–[5] are incorporated to improve the performance. For search space, a network is partitioned into multiple blocks, each with a specific set of configurations [3]. The configurations include factors for the neural network, such as the number of layers, layer type, and activation type. Data arithmetic is also considered for hardware-aware search.

For performance evaluation, the network is evaluated using a multi-objective score that considers key hardware metrics, such as latency and power. The total path count (TPC) [4] is

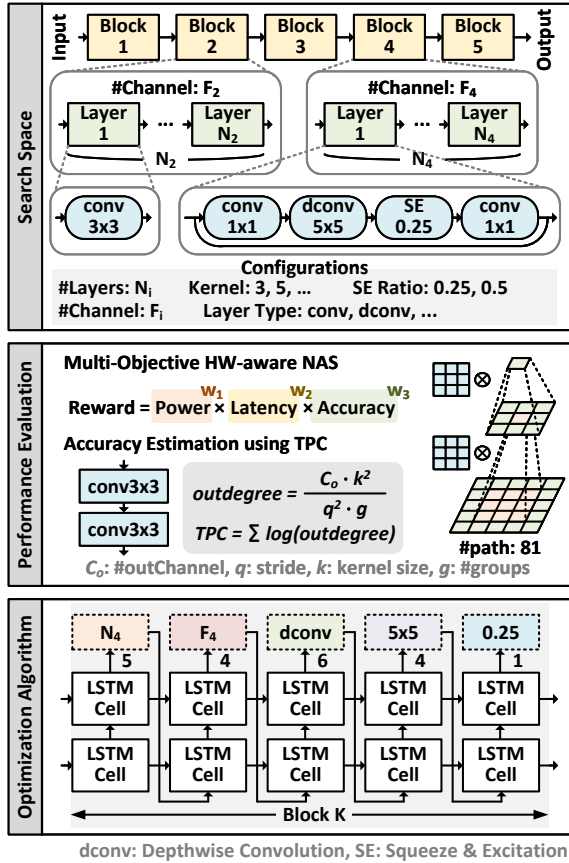


Fig. 3: Details of the NAS workflow.

adopted as an accuracy estimator to achieve fast performance evaluation. It considers the outdegree (total number of paths between neurons) in the neural network and predicts the accuracy solely on the network architecture. A strong correlation of 0.96 between the TPC score and the accuracy of neural network has been verified on image classification task in [4].

For optimization algorithm, reinforcement learning (RL) is employed to enable hardware-aware NAS based on a multi-objective reward [5]. In the RL process, a long short-term memory-based (LSTM-based) network, in which each LSTM cell outputs a configuration that defines the neural architecture, is utilized to select a network from the search space. The weights of the selected network are then determined by performing training on the entire dataset.

### III. SYSTEM ARCHITECTURE

Fig. 4 shows the system architecture of the proposed fully-integrated SoC, which includes a matrix processing engine (MPE), a vector processing engine (VPE), an inter-module synchronizer, and an Arm Cortex-M3 MCU. The MPE supports compute-intensive multiply-accumulate (MAC) operations for various neural networks. Its dataflow can be reconfigured to achieve higher hardware utilization by dynamically adjusting the level of parallelism. Furthermore, mixed-precision arithmetic (with FP8 and BF16 data formats [6]) is considered to explore more network architectures. The VPE incorporates vector function units (VFUs) that support

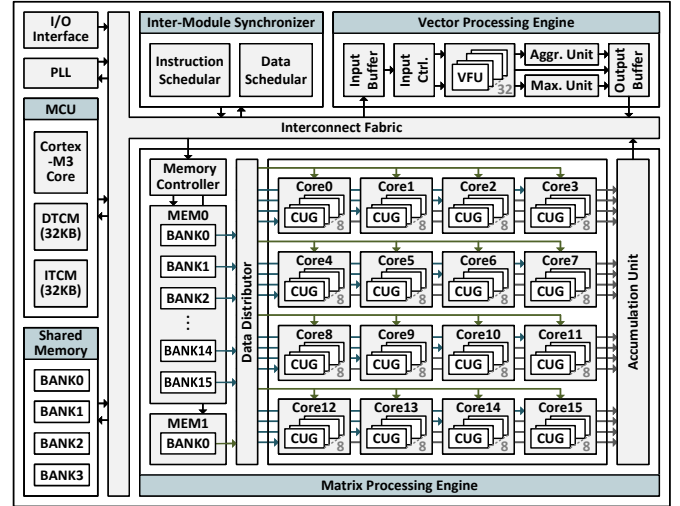


Fig. 4: System architecture of the proposed NAS SoC.

commonly used activations. The inter-module synchronizer coordinates data exchange between processing engines through a shared memory bank. The Cortex-M3 MCU handles I/O interfacing and hardware configuration.

#### A. Matrix Processing Engine

Fig. 5 depicts the hardware optimizations on the MPE. This engine supports MAC operations for matrix-matrix and matrix-vector multiplication. Standard, depthwise, and pointwise convolutions are also supported to expand the search space. The MPE consists of 16 cores, each containing 8 computing unit groups (CUGs). The level of parallelism is dynamically adjusted by a switch network for array configuration, maximizing hardware utilization for various layer types. A memory controller is designed to support the varying shape of the neural network. In the proposed architecture, input and weight are stored in two memory banks and accessed through unicast and broadcast interconnects. The roles of the two memory banks are exchanged conditionally for accommodating the varying network shape. The memory address is aligned with the data access sequence, enabling a two-port SRAM implementation for data preloading instead of double buffering. The proposed memory control mechanism reduces the memory size by 60%. For data accumulation unit, a shared adder tree architecture is designed to support diverse operations, reducing hardware resources by 53% compared to a fixed adder structure.

#### B. Mixed-Precision Computing Unit

Fig. 6 shows the details of the mixed-precision computing unit (CU). This work supports network layers using FP8 (E4M3 and E5M2) and BF16 formats, which are leveraged to expand the search space with respect to accuracy and network size. The mixed-precision CU integrates FP8 multipliers with reconfigurability to compute BF16 operands in four cycles. The processing cycle for BF16 can be reduced by 25% by replacing the least-significant 8-bit partial sum with a stochastically rounded number. The area of the CU architecture is reduced by 62% compared to that of direct-mapped design.

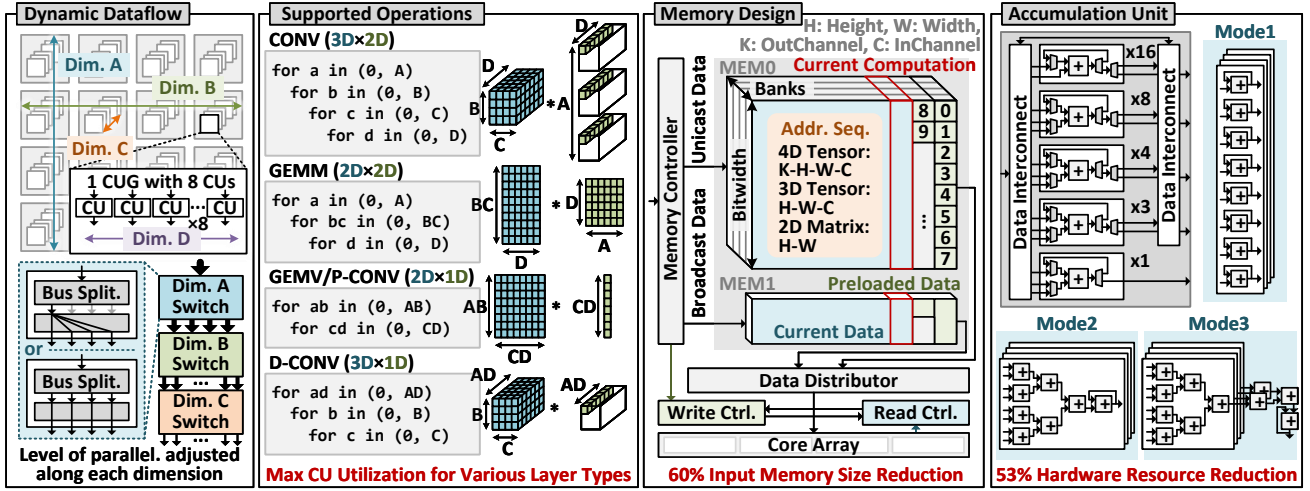


Fig. 5: Hardware optimizations for the MPE.

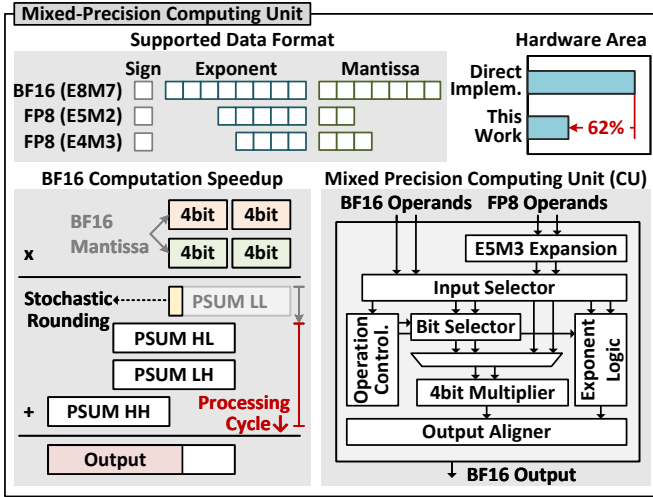


Fig. 6: Detailed architecture of the mixed-precision CU.

### C. Vector Processing Engine

Fig. 7 shows the design of the VPE. The VPE features an array of VFUs, each including a coordinate rotation digital computer (CORDIC)-based module for diverse activation functions. The area-efficient CORDIC-based hardware architecture is feasible since all the required operations can be performed by add-and-shift operations. The hardware architecture can be reconfigured to support linear and nonlinear operations in two modes. Such a design reduces hardware complexity by 33% compared to the implementation with dedicated multipliers and special function operators. A negative indexing CORDIC technique [7] is adopted to increase the input data range by employing an extended iteration index range.

### D. Inter-Module Synchronizer

Fig. 8 shows the interconnect topology. The interconnect between the MPE and the VPE needs to be reconfigurable to support the diverse architecture of network layers. A unidirectional ring facilitates intermediate data transfer with minimized

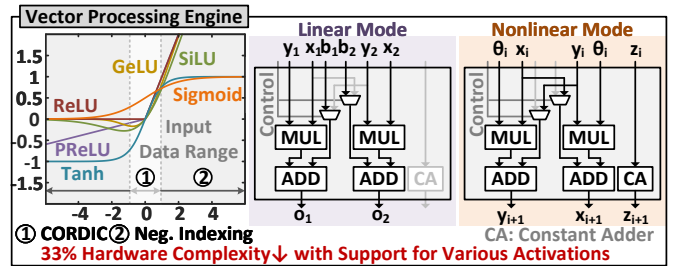


Fig. 7: Hardware design for the VPE.

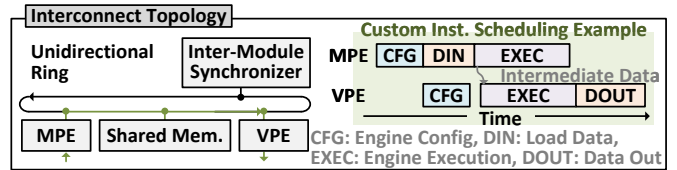


Fig. 8: Design of the interconnect topology.

external memory access. The inter-module synchronizer is designed to align data and instructions by processing sub-networks in a pipelined manner.

## IV. CHIP IMPLEMENTATION

The supported NAS configurations of the proposed SoC are shown in Fig. 9. Fig. 10 and Table I show the chip micrograph and chip summary. Fabricated in a 16nm FinFET CMOS technology, the chip integrates 6.4M logic gates in a core area of 2.83mm<sup>2</sup>. The proposed SoC dissipates 255mW at a clock frequency of 500MHz from a 0.8V supply. Due to the limited tapeout resource, a 1024-CU and 32-VFU version is designed and implemented. However, the hardware architecture is scalable to support more PEs and VFUs for searching larger neural networks.

Fig. 11 shows the FPGA evaluation platform and experimental results. The functionality of the SoC is verified by a custom FPGA evaluation platform (with an AMD ZYNQ UltraScale+ ZU19EG device) with 8GB DDR4 DRAM. The

CNN	RNN	Transformer
Number of Layers	Number of Layers	Number of Layers
Convolution Layer Type	RNN Cell Type	Attention Type
Pooling Layer Type	Activation Function	Activation Function
Activation Function	Hidden Size	Number of Heads
Kernel Size, Pooling Size	Bidirectionality	Sequence Length
Stride Size, Dilatation Size		Hidden Dimension
SE Ratio		Feedforward Size
Skip Connection		Normalization Type

Fig. 9: Supported NAS configurations.

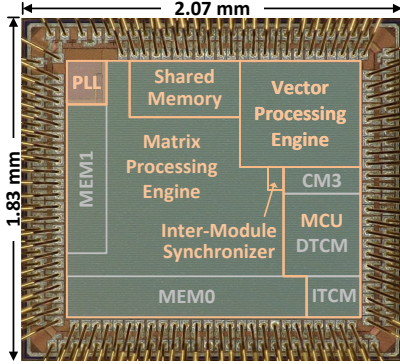


Fig. 10: Chip micrograph.

TABLE I: Chip summary

Technology	16nm	
Network Type	CNN, RNN, Transformer	
Data Format	BF16, FP8 (E5M2/E4M3)	
Area (mm <sup>2</sup> )	Die	2.07 × 1.83
	Core	1.80 × 1.57
On-Chip SRAM (KB)	322	
Core Voltage (V)	0.8	
Clock Frequency (MHz)	500	
Power (mW)	255	

executables compiled by the ARM toolchain are loaded by the FPGA and executed on the SoC. Image classification on CIFAR-10 is used for performance evaluation. The proposed SoC explores 310 architectures (samples) in 4.3 hours, including time for search and training, in the accuracy-latency space.

Fig. 12 shows the performance comparison with a mid-range data center GPU. Compared to the NVIDIA A40 GPU, this SoC demonstrates a 27× speedup at a 2.6× lower clock frequency, given 166× smaller silicon area and 1176× less power, at a less advanced technology node. The chip achieves an energy efficiency of 78.5 samples/kJ and an area efficiency of 19.0 samples/hr-mm<sup>2</sup>, outperforming the GPU by 31,416× in energy and 4,419× in area.

## V. CONCLUSION

This work presents the first fully integrated SoC for hardware-aware NAS, enabling efficient exploration of optimized network architectures. Our workflow efficiently explores design candidates by employing RL on partitioned search space, under hardware constraints. The TPC is incorporated as an accuracy estimator for fast performance

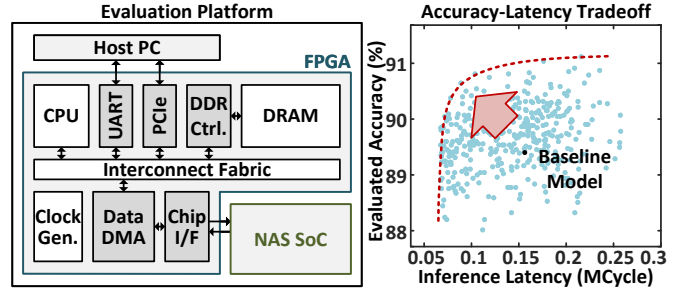


Fig. 11: Experimental verification and evaluation platform.

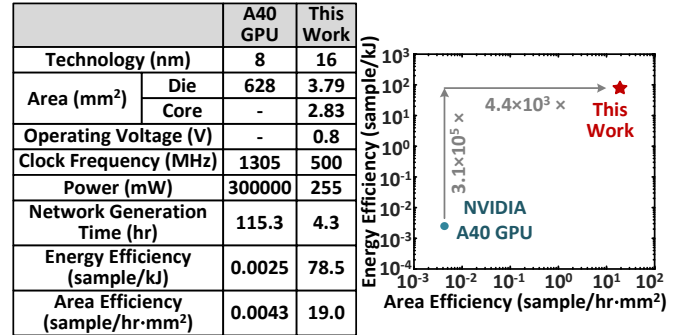


Fig. 12: Performance comparison.

evaluation. The SoC integrates a matrix processing engine, a vector processing engine, and an MCU, achieving a flexible and scalable architecture for NAS. Key hardware optimizations include dynamically reconfigurable dataflow, customized mixed-precision arithmetic, and area-efficient hardware design for diverse activation functions. The chip demonstrates a 27× speedup compared to an NVIDIA A40 GPU while operating at a 2.6× lower clock frequency, with significantly lower power and area requirements. This work achieves orders-of-magnitude improvements in both energy efficiency and area efficiency, demonstrating a promising solution for fast neural architecture search in an energy-efficient way.

## ACKNOWLEDGEMENT

This work is supported by NSTC of Taiwan. The authors also thank TSRI for support on chip design and fabrication.

## REFERENCES

- [1] V. Sze *et al.*, *Efficient Processing of Deep Neural Networks*, Springer, 2020.
- [2] H. Cai *et al.*, “Enable Deep Learning on Mobile Devices: Methods, Systems, and Applications,” *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 27, no. 3, pp. 1-50, 2022.
- [3] M. Tan *et al.*, “MnasNet: Platform-Aware Neural Architecture Search for Mobile,” *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2815-2823, 2019.
- [4] M.-S. Huang *et al.*, “TPC-NAS: Simple and Effective Neural Architecture Search Based on Total Path Count,” *IEEE International Conference on AI Circuits and Systems (AICAS)*, pp. 542-546, 2024.
- [5] S. Li *et al.*, “Searching for Fast Model Families on Datacenter Accelerators,” *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8081-8091, 2021.
- [6] P. Micikevicius *et al.*, “FP8 Formats for Deep Learning,” *arXiv*, 2022. Available: <https://arxiv.org/abs/2209.05433>
- [7] X. Hu *et al.*, “Expanding the Range of Convergence of the CORDIC Algorithm,” *IEEE Transactions on Computers*, vol. 40, no. 1, pp. 13-21, 1991.