Breakthrough edge AI inference performance using NorthPole in 3U VPX form factor

Filipp Akopyan, William P. Risk, John V. Arthur, Andrew S. Cassidy, Michael V. Debole, Carlos Ortega Otero, Jun Sawada, Evan Colgan, Michael E. Criscolo, Phillip V. Mann, Heinz Baier, Kai Schleupen, Arnon Amir, Alexander Andreopoulos, Rathinakumar Appuswamy, Deepika Bablani, Peter J. Carlson, Pallab Datta, Steven K. Esser, Myron D. Flickner, Rajamohan Gandhasri, Guillaume J. Garreau, Megumi Ito, Jennifer L. Klamo, Jeffrey A. Kusnitz, Nathaniel J. McClatchey, Neil McGlohon, Jeffrey L. McKinstry, Yutaka Nakamura, Tapan K. Nayak, Jay Sivagnaname, Daniel F. Smith, Rafael Sousa, Brian Taba, Ignacio Terrizzano, Takanori Ueda, Dharmendra S. Modha*

IBM Research

*dmodha@us.ibm.com

***Abstract*—We present preliminary results demonstrating AI (artificial intelligence) inference using the IBM AIU NorthPole Chip [1], [2] incorporated into a compact, rugged 3U VPX form factor module (NP-VPX) [3]. NP-VPX allows NorthPole to be used in edge applications with stringent cooling requirements, high-speed switch fabrics, and rugged environments. NP-VPX processes 965 frames per second (fps) with a Yolo-v4 network with 640 × 640 pixel images at 73.5 W at full-precision accuracy, achieving 13.2 frames / J (fps / W). NP-VPX processes over 40,300 fps with a ResNet-50 network with 224 × 224 pixel images at 65.9 W at full-precision accuracy, achieving 611 frames / J.**

***Index Terms*—VPX, AI accelerator, Yolo, ResNet, HPEC**

I. Introduction

Today, high-performance AI runs primarily in the datacenter and—while training may remain there—great opportunity exists to migrate inference out to the edge, reducing transmission energy as well as bandwidth, mitigating concerns regarding privacy as well as security, and enabling previously impossible applications. To enable inference outside the datacenter, users need AI accelerators with both high performance and high energy efficiency, embodied in a form factor optimized for deployment at the edge.

To address the need to widely deploy inference, we have developed the research prototype NorthPole VPX board (NP-VPX), which embeds the IBM AIU NorthPole inference processor in a 3U VPX form factor module (Figs. 1 & 2). Compared to NorthPole's earlier PCIe form factor board [1], [2], NP-VPX achieves similar throughput with improved energy efficiency due to a lower power FPGA (Fig. 2), leading the shift from "train and deploy in the datacenter" to "train in the datacenter and deploy everywhere."

II. NP-VPX Board

3U VPX is a standard and flexible form factor for modular rugged systems [3], often used in defense, aerospace, and other applications that must operate in challenging environments [4]. Numerous modules have been developed previously in this form factor to implement computational and signal processing

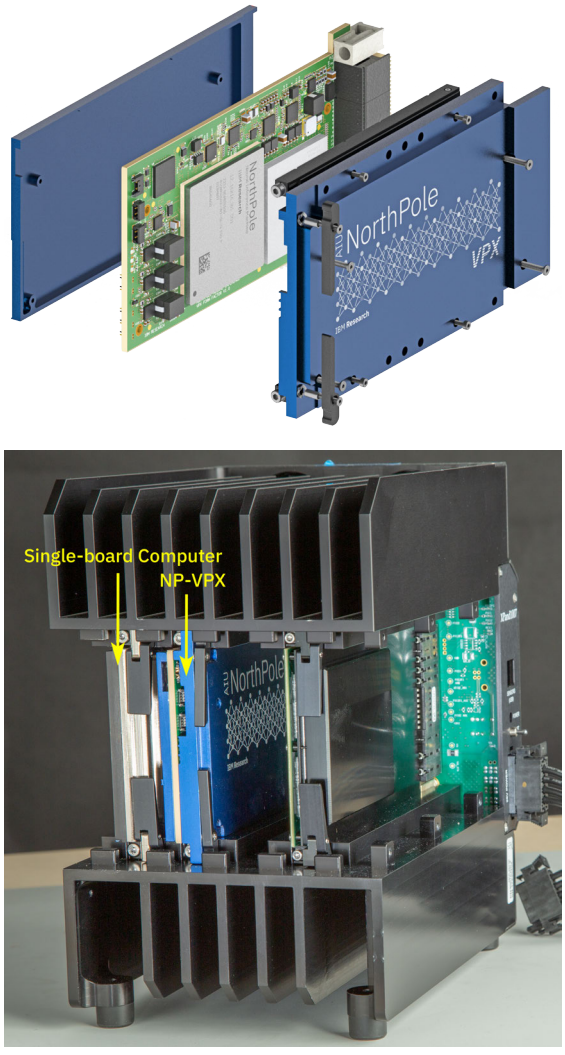

Fig. 1. *Top*: Exploded-view rendering of NorthPole VPX assembly. *Bottom*: Photo of the fully functional, fabricated, and assembled NorthPole VPX module, inserted into a VPX chassis with a single-board computer.

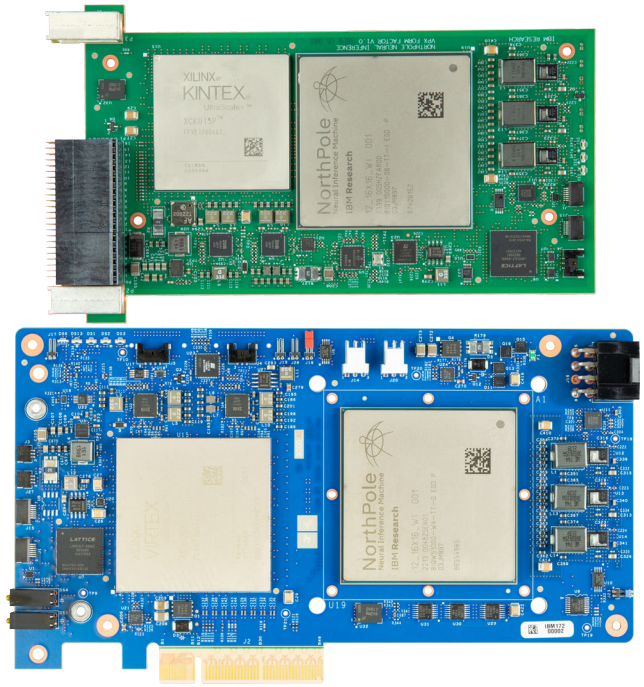

Fig. 2. *Top*: NorthPole VPX board, optimized for area and density in the 3U VPX form factor. *Bottom*: NorthPole PCIe board, designed for a server and datacenter form factor [1].

functions, using conventional CPUs, GPUs, and FPGAs. The NP-VPX board enables the novel NorthPole processor to be used in 3U VPX systems for high-performance, high-efficiency neural network inference.

A. AIU NorthPole

The NorthPole inference processor has demonstrated exceptional performance and energy, space, and time efficiency [1], [2]. Its architecture is based on a parallel, distributed core array, emphasizing on-chip distributed memory, high parallelism, low-precision compute, and deterministic control as well as data locality (Fig. 3). Further, NorthPole control is fully encapsulated on-chip, in the sense that it runs an entire neural network without orchestration from a host CPU. It enables a simple Put–Run–Get interaction model:

1) Put input tensor (such as an image)
2) Run inference model (such as a ResNet network)
3) Get output tensor (such as a classification)

To expand the scope for exploiting NorthPole's capabilities in edge deployments, we migrated the design to the 3U VPX form factor, providing an unmatched size, weight, and power (SWaP) system.

B. NP-VPX Design & Layout

The board includes the NorthPole Chip, a Xilinx Kintex Ultrascale+ FPGA, 3U VPX backplane connector and guide sockets, power, measurement, and (passive and active) support components (Fig. 1) in a 32-layer, 100 mm $\times$ 160 mm printed circuit board compliant with the 3U VPX standard (with a

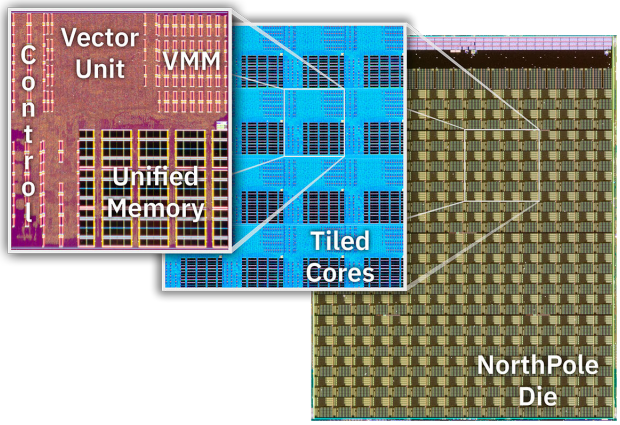

Fig. 3. NorthPole chip micrograph. Each NorthPole core (*left*) has a vector-matrix multiplication (VMM) unit, a vector unit, unified memory, and control logic. Cores are tiled (*middle*) in a 16×16 array on the NorthPole die (*right*).

usable area for components and routing of approximately 80 mm $\times$ 160 mm due to the cutout to accommodate the cover and wedgelocks). The FPGA acts as a bridge between the NorthPole Chip and the PCIe Gen 3 $\times 8$ bus that links to a host over the VPX backplane. It also allows control of power and measurement components via I^2C. The FPGA can also perform pre- and post-processing if required.

The power control system taps off the 12 V power domain and provides all of the voltage domains needed by the board. A Lattice MachXO3 FPGA controls the power bring-up sequence, observes fault signals, and—if detected—controls power-down. A power measurement analog-to-digital converter measures the voltage drop across a shunt resistor on the 12 V PCIe power input, sampled at approximately 1 kHz, enabling whole-board power to be calculated.

The NP-VPX board design is based on a previous research prototype PCIe form-factor board (described in [1]), which used a larger FPGA with more I/O pins and a larger board area of 100 mm $\times$ 192 mm plus PCIe connector (Fig. 2 *Bottom*). Shrinking the design's usable area by nearly one third involved aggressive reduction of NorthPole-to-FPGA connections to the minimum required for operation. For example, as chips are previously tested before assembly, all connections to design-for-test pins were removed. Further, components had to be moved closer and routing compressed into the smaller area.

Because the board is intended for deployment in a conduction-cooled VPX chassis, we performed thermal modeling to guide the design of aluminum covers that are used with standard commercial wedgelocks to conduct heat from components on the board to the chassis.

C. NP-VPX Performance and Efficiency

We have previously demonstrated NorthPole's unsurpassed energy efficiency on benchmark neural network inference [1] while maintaining high throughput. Here we show that efficiency is maintained and even improved, showing the highest throughput and energy efficiency in the 3U VPX form factor.

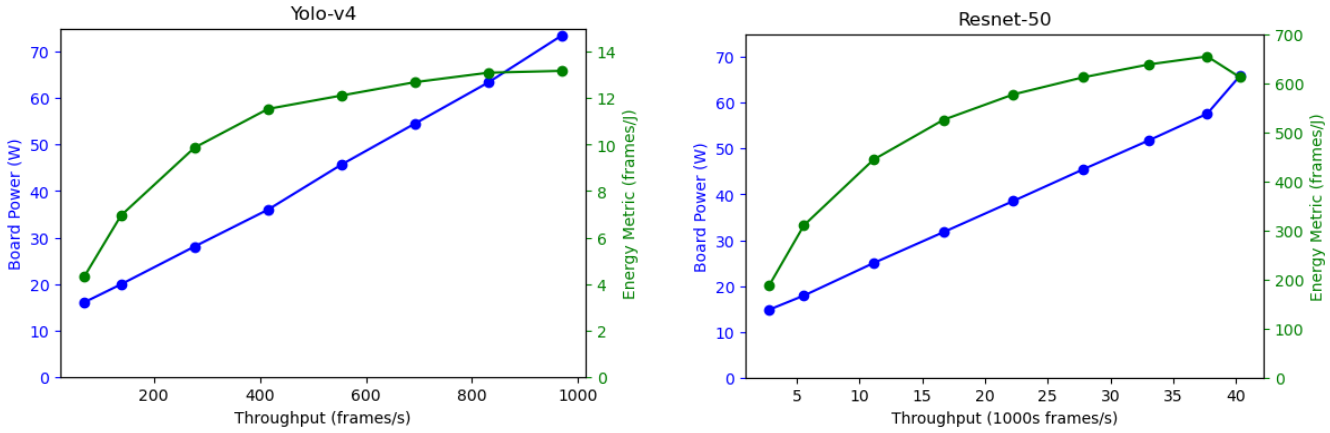

Fig. 4. Measured NorthPole VPX board power, throughput, and energy efficiency. *Left*: Running Yolo-v4 at 350 MHz, the board processed 969 fps at 640×640 pixels per image, consuming 73.5 W for a board-level efficiency of 13.2 frames / J. *Right*: Running ResNet-50 at 400 MHz, the board processed $40,340$ fps at 224×224 pixels per image, consuming 65.9 W for a board-level efficiency of 612 frames / J.

We characterized NP-VPX's operation by running two benchmark neural networks: Yolo-v4 [5] and ResNet-50 [6], trained to match full-precision accuracy. For this initial testing, we connected the board through a VPX-to-PCIe interface board to a standard server. In this arrangement, power is provided through the PCIe slot and is limited to 75 W (NP-VPX supports higher power operation in a VPX chassis, powered through the backplane.) We ran a mixed $4b/8b$-precision Yolo-v4 model with a batch size of two, at NorthPole clock frequencies ranging from 25 to 350 MHz, which was the maximum given the power limit (Fig. 4 *Left*). We started with a NorthPole core voltage of 0.8 V and raised it as needed for higher speeds, reaching 0.825 V at 350 MHz. At 350 MHz, the board processed 969 frames per second (fps) at 640×640 pixels per image, consuming 73.5 W for an efficiency of 13.2 frames / J (fps / W) (at the board level). NorthPole performance at 350 MHz is sufficient to process 32 real-time (30 fps) image streams on a single board. For highly power-constrained systems, limited to 25 and 50 W, we estimate throughput of 225 and 621 fps, respectively.

We ran a mixed $2b/4b/8b$-precision ResNet-50 model with a batch size of 32, at NorthPole clock frequencies from 25 to 400 MHz (Fig. 4 *Right*). We started with a NorthPole core voltage of 0.76 V and raised it as needed for higher speeds, reaching 0.82 V at 400 MHz. At 400 MHz, the board processed $40,340$ frames per second (fps) at 224×224 pixels per image, consuming 65.9 W for an efficiency of 612.0 frames / J (at the board level). Again, for highly power-constrained systems, limited to 25 and 50 W, we estimate throughput of $11,115$ and $31,574$ fps, respectively.

To estimate SWaP, in a power-constrained application, we retrained the Yolo-v4 network on a subset of the xView Dataset [11]. NP-VPX has a throughput of 225 (1280×1280 pixel images) fps in 70.7 W with the NorthPole clock at 300 MHz (see Fig. 5).

III. NorthPole Software Stack

The NorthPole Software Development Kit (SDK) provides NorthPole users with standard workflows for training, compiling, and running neural network inference models [1]. No changes were necessary to the SDK to use NorthPole in the VPX form factor.

The SDK supports quantization-aware training (using a GPU), compiling the resulting neural network into a hardware-compatible binary file, and running the network with the simple Put–Run–Get interface that minimally loads the host CPU.

A. SDK Training

The SDK provides a training flow (run on a GPU) to quantize models for NorthPole supported $2b$, $4b$, and $8b$ precisions, as well as tools to select precisions to retrain without loss of accuracy. Training is done in PyTorch [7] and can quantize existing models or train from scratch. Two quantization-aware training algorithms are provided: Fine-tuning After Quantization (FAQ) [8] and Learned Step-size Quantization (LSQ) [9]. Additional algorithms guide precision selection to optimize network performance, memory usage, and throughput [10].

B. SDK Compiling

Once a model is trained to NorthPole precisions, PyTorch exports the model to the compiler, which generates a hardware-compatible binary file in the standard Executable and Linkable Format (ELF). The ELF file includes all model parameters and all instructions to sequence all memory, communication, and compute operations.

C. SDK Runtime

After model compilation, a runtime application interacts with the board. It loads the ELF file onto the NorthPole Chip, loads input tensors (such as images) into the on-chip frame

Fig. 5. Examples of frames processed by a NorthPole VPX board running a Yolo-v4 network trained on the xView dataset. Labeled classes include buildings (*yellow*), aircraft (*white*), trucks (*green*), boats (*magenta*), and cars (*orange*).

buffer memory, and sends application packets to start operation. Then, after execution, the runtime application receives the neural network results. The runtime application (and thus the host CPU) does not schedule any layer-by-layer operations; instead, it puts input tensors into the device, starts execution of the entire network, and gets the resulting output tensors.

IV. Summary

The NorthPole VPX Board provides high-performance and high-efficiency embedded AI inference acceleration in an optimized SWaP form factor. It provides a platform that can deliver previously impossible neural network compute power to VPX-based systems [4], bringing AI inference out of the datacenter.

Acknowledgment

This work builds on previously published research [1] that was supported by the United States Air Force under Contract No. FA8750-19-C-1518. The authors are grateful to Qing Wu and Chris Capraro for valuable discussions.

REFERENCES

[1] D. Modha et al., "Neural inference at the frontier of energy, space, and time," Science, vol. 382, no. 6668, pp. 329–335, 2023.

[2] A. S. Cassidy et al., "IBM NorthPole: An Architecture for Neural Network Inference with a 12nm Chip," 2024 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 2024, pp. 214-215.

[3] https://www.vita.com/Standards

[4] M. Barnell, C. Raymond, C. Capraro and D. Isereau, "Agile Condor: A scalable high performance embedded computing architecture," 2015 IEEE High Performance Extreme Computing Conference (HPEC), Waltham, MA, 2015, pp. 1-5.

[5] A. Bochkovskiy, C. Y. Wang, H. Y. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv:2004.13934, 2020.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in IEEE CVPR, 2016, pp. 770-778.

[7] PyTorch, Available online: https://pytorch.org.

[8] J. McKinstry, et al., "Discovering Low-Precision Networks Close to Full-Precision Networks for Efficient Inference," Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition, 2019.

[9] S. Esser, et al., "Learned Step Size Quantization," arXiv:1902.08153, 2019.

[10] D. Bablani, et al., "Efficient and effective methods for mixed precision neural network quantization for faster, energy-efficient inference," arXiv:2301.13330, 2023.

[11] D. Lam, et al., "xview: Objects in context in overhead imagery," arXiv:1802.07856, 2018.