PETRA: A 22nm 6.97TFLOPS/W AIB-Enabled Configurable Matrix and Convolution Accelerator Integrated with an Intel Stratix 10 FPGA

Sung-Gun Cho[1,2], Wei Tang[1], Chester Liu[1], Zhengya Zhang[1]
[1] University of Michigan, Ann Arbor, MI, USA [2] Intel Corporation, San Jose, CA, USA

Abstract

PETRA is a configurable FP16 matrix multiplication and convolution accelerator designed to be 2.5D integrated using Advanced Interface Bus (AIB). PETRA is built upon four 16×16 systolic arrays, but it employs a configurable H-tree accumulation to improve both the latency and the utilization by up to 8×. A 22nm 3.04mm^2 PETRA prototype provides 1.433TFLOPS in computing matrix-matrix multiplication (MMM) and convolution (conv) at 0.88V, and it achieves a 6.97TFLOPS/W peak efficiency at 0.7V. PETRA is integrated with an Intel Stratix 10 FPGA in a multi-chip package (MCP) to provide the flexibility of FPGA and the performance and efficiency of PETRA.

Introduction

Machine learning (ML) and communication DSP are the driving applications of the next-generation computational hardware. Dedicated ML and DSP ASICs have been built, but the workloads are evolving at a fast pace, demanding increasing flexibility. We present a configurable FP16 MMM and conv accelerator named PETRA that is integrated with an Intel Stratix 10 FPGA over an 8-channel, 640Gb/s, sub-pJ/b AIB (Fig. 1) [1] on an Embedded Multi-die Interconnect Bridge (EMIB) [2]. The heterogeneous integration provides FPGA's flexibility to handle control flow, data arrangement and simple pre-/post-processing, while allowing the most computation-intensive kernels to be offloaded to the performance- and efficiency-optimized accelerator.

A matrix accelerator is commonly implemented in a 2D systolic array [3] for its regular structure and efficient data reuse. An n-by-n systolic array computes n^2 products in parallel and the partial sums are accumulated in n steps. Increasing n linearly increases the accumulation latency and makes it less efficient for smaller workloads. We present Processing Element TRee Array (PETRA) (Fig. 2) based on a systolic array but enhanced by an H-tree accumulation to shorten the latency from $O(n)$ to $O(\log(n))$. The tree is configurable to support concurrent workloads sharing an array to improve utilization. PETRA supports MMM; and by leveraging vertical and horizontal input shifting and reuse, it efficiently computes conv. PETRA is designed in FP16 to extend its application to training and communication DSP that require a high dynamic range. Integrated with an FPGA over an efficient high-bandwidth AIB interface, the heterogeneous system achieves both flexibility as well as performance and efficiency by apportioning control to the FPGA and computation kernels to PETRA.

Low-Latency and High-Utilization PE Tree Array

An n-by-n systolic array is wired both vertically and horizontally (and sometimes diagonally) [3], one for loading inputs and one for loading weights. PETRA's PE array is weight-stationary, so we keep only vertical paths for both input and weight loading (Fig. 3). PETRA's PE consists of a FP16 multiplier, a d-element rotating weight buffer and a data buffer. Partial sum accumulation is via an H-tree Pipelined Adder Tree (HPAT) (Fig. 3) instead of a linear accumulation pipeline in a systolic array. HPAT shortens the accumulation latency to $O(\log(n))$. The H-tree allows evenly distributed fan-ins that can be easily scaled up. Splitting multiply and add also enables a higher clock frequency.

The upper levels of HPAT are made of a configurable adder tree (CAT) (Fig. 4). CAT allows partial sums to be summed in any adjacent combinations or forwarded to output. Hence PETRA's PE array accommodates multiple concurrent matrix computations. In the prototype, an 8-input CAT is implemented over a 16×16 PE array, providing options to compute up to 8 separate summations for an 8× higher utilization. The HPAT and CAT are scalable techniques applicable to larger PE arrays for improving latency and utilization.

Prototype Design and Mapping

The PETRA prototype contains four 16×16 PE arrays that can run in parallel (Fig. 2). The I/O paths support die-to-die stream processing and buffered processing. The cross-array dispenser and within-array dispenser provide I/O distribution and facilitate data reuse. The outputs of PE arrays are streamed out or stored in the accumulation buffer.

A PE stores 16 weights in its buffer, and a 16×16 PE array stores a weight matrix of size up to 256×16. A PE array computes one 256-element inner product per cycle. By keeping inputs stationary and rotating 16 weights in each PE, a PE array computes one (1×256) × (256×16) vector-matrix multiplication over 16 cycles. By loading a new row of 16 inputs per cycle, a PE array computes one (m×256) × (256×16) MMM over 16m cycles (Fig. 5). Input loading and computing are overlapped. Since the input can be streamed in, m is not limited. Larger weight matrices are divided into 256×16 submatrices to map onto multiple PE arrays, using the accumulation buffer to produce the MMM outputs. Smaller weight matrices can share one PE array. The cross-array dispenser supports multicast to PE arrays for input reuse and workload balancing.

A PE array computes 2D conv by keeping weights stationary and shifting inputs: along the vertical paths in a PE array or the horizontal paths in the dispenser FIFOs (Fig. 6). For instance, to compute a 3×3 conv, a 3×3 input tile X_1 is loaded in the dispenser and shifted down the PE array. A 3×3 PE tile computes the inner product $X_1 \cdot W$, where W is a 3×3 kernel stored in the PE tile. In every cycle, a new row is loaded from the top and the input is shifted down by one row (for stride-1); and the 3×3 PE tile receives a new input tile X_i and computes $X_i \cdot W$. When the input reaches the bottom, the dispenser shifts the input one column to the left. The vertical and horizontal shifts enable the 2D scanning of an input to complete conv.

Chiplet Integration and Measurement Results

A PETRA chiplet was designed and fabricated in an Intel 22nm FinFET CMOS process with an 8-channel, 640Gb/s, sub-pJ/b AIB as the chiplet's I/O interface (Fig. 7). The PETRA chiplet was integrated with a Stratix 10 FPGA using EMIB [2] in an MCP. An AXI-compatible bus interface named University of Michigan AIB Interface (UMAI) is defined to encapsulate AIB's multi-channel, free-flowing data transmission in address-based read and write bus transactions. In our implementation, one UMAI interface converts up to 8 AIB channels to a 512-b data bus interface. A UMAI bus master is instantiated in the FPGA while a UMAI bus slave is embedded in the PETRA chiplet. UMAI abstracts the die-to-die interface to an SoC-like bus, and hence simplifies a chiplet integration to an IP integration.

The 3.04mm^2 22nm PETRA is measured to consume 637.5mW at 701MHz in room temperature at the nominal supply of 0.88V, when all 4 PE arrays are fully utilized for random non-zero MMMs. The results translate to a power efficiency of 2.25TFLOPS/W (FP16) and a compute density of 0.472TFLOPS/mm^2 (Fig. 8). The peak power efficiency of 6.97TFLOPS/W is measured at 0.7V. PETRA provides a competitive FP16 power efficiency and compute density compared to state-of-the-art DNN accelerators [3]-[7] (Table 1), some of which are in more advanced process nodes [6], [7]. PETRA's latency is notably lower. The performance of sample workloads is listed in Fig. 8. PETRA is the first configurable matrix accelerator designed for an FPGA-based MCP that leverages heterogeneous integration and an advanced interface to enable new ML and DSP applications.

Acknowledgements

This work was supported in part by DARPA CHIPS and ONR under grant N00014-17-1-2992. DARPA HI3 provided chip fabrication and packaging. We would like to thank F. Sheikh, M. Flanigan, A. Chan, T. Hoang, T. Tran, D. Kehlet, and S. Shumarayev from Intel for advice and assistance.

References

[1] D. Greenhill, et al., ISSCC, 2017.
[2] R. Mahajan, et al., ECTC, 2016.
[3] N. P. Jouppi, et al., ISCA, 2017.
[4] S. Yin, et al., JSSC, 2018.
[5] J. Lee, et al., JSSC, 2019.
[6] C. Lin, et al., ISSCC, 2020.
[7] J. Oh, et al., VLSI, 2020.

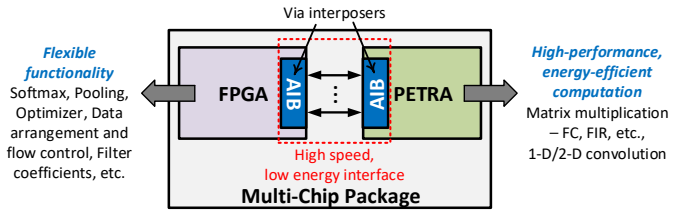


Fig. 1 A system in a multi-chip package integrating a PETRA accelerator chiplet and a FPGA.

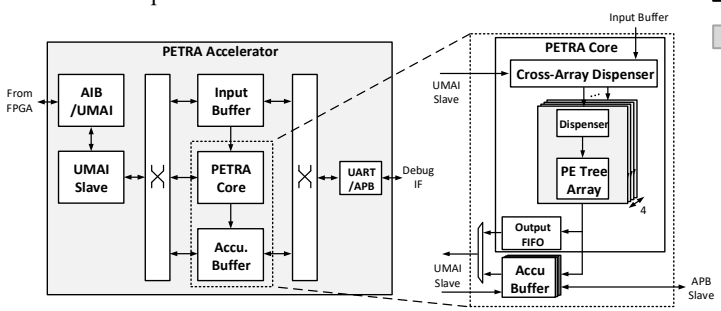


Fig. 2 Overall architecture of PETRA accelerator.

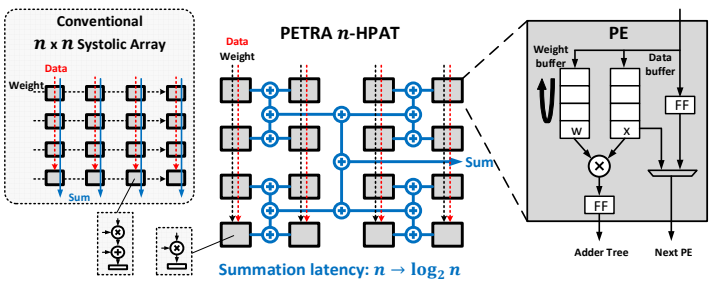


Fig. 3 Advantages of PETRA architecture and PE design.

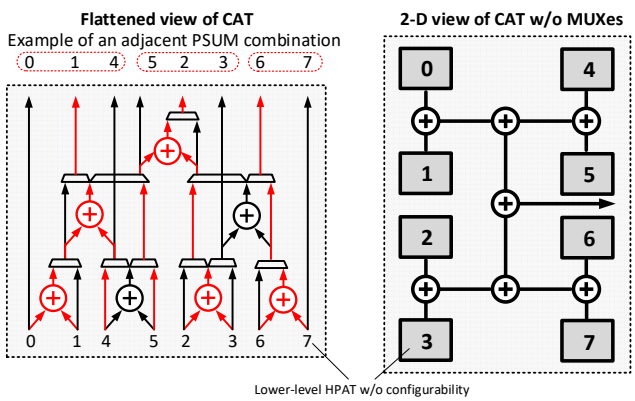


Fig. 4 1-D and 2-D views of CAT, enabling any partial sum (PSUM) output combination of adjacent PSUM inputs.

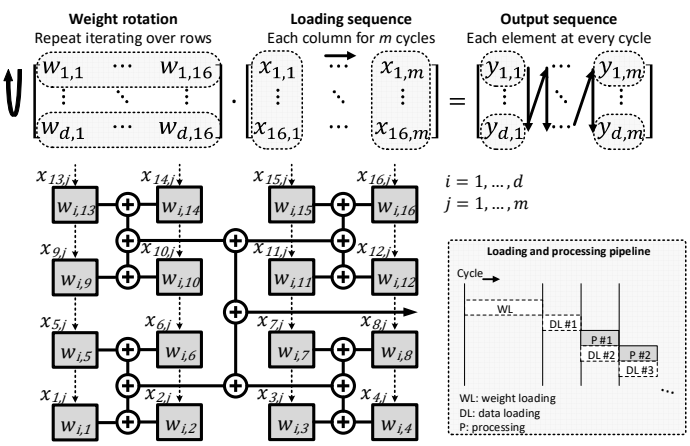


Fig. 5 MMM operation flow.

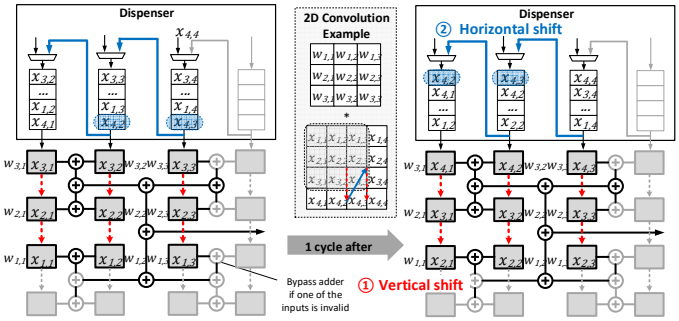
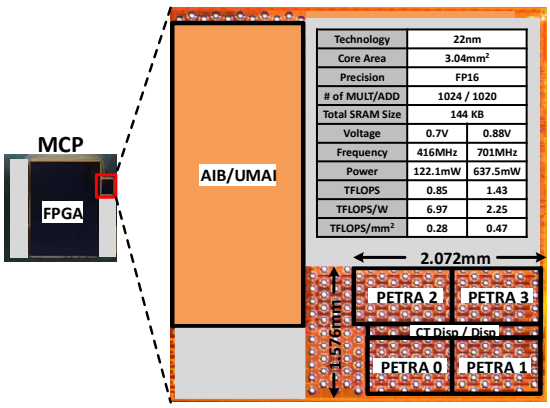
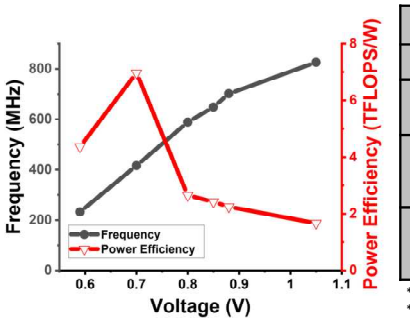


Fig. 6 A 3×3 convolutional layer operation flow example.



Technology	22nm	
Core Area	3.04mm²	
Precision	FP16	
# of MULT/ADD	1024 / 1020	
Total SRAM Size	144 KB	
Voltage	0.7V	0.88V
Frequency	416MHz	701MHz
Power	122.1mW	637.5mW
TFLOPS	0.85	1.43
TFLOPS/W	6.97	2.25
TFLOPS/mm²	0.28	0.47

Fig. 7 Die photo (unrelated designs are covered) and specifications.



Application	Throughput	Utilization ratio
VGG-16*	34.9 frame/s	0.87
Transformer Multi-head attention*	278.5 seq/s	1
MU-MIMO match filtering (128 antennas and 16 users)	1.40 GS/s**	1
MU-MIMO interference cancellation (16 users)	11.22 GS/s**	1

* Softmax and pooling excluded
** Giga symbols (complex) per second

Fig. 8 Frequency and power efficiency over voltage scaling (left) and performance on various workloads (right).

Table. 1 Comparison with the prior works.

	TPU [3]	Hybrid [4]	UNPU [5]	DLA [6]	Scalable [7]	This Work
Process [nm]	28	65	65	7	14	22
Die Area [mm²]	331	19.4	16	3.04	-	3.33
Core Area [mm²]	81	14.44	-	2.68	9.84	3.04
Voltage [V]	0.75	0.67 - 1.2	0.63 - 1.1	0.575 - 0.825	0.54 - 0.62	0.59 - 1.05
Frequency [MHz]	700	10 - 200	200	290 - 880	1000 – 1500	233 - 826
Power [mW]	40000	4.0 - 386	3.2 - 297	174 - 1053	-	109.0 - 1010.7
Bit Precision	INT8	INT8/16	INT1-16	ASYMM-Q8, INT8/16, FP16	DLFP16/32	FP16
Peak Performance	91.8 TOPS (INT8)	0.41 TOPS (INT8)	0.346 TOPS (INT16) - 7.372 TOPS (INT1)	3.6 TOPS (INT8) 0.9 TFLOPS (FP16)	2 - 3 TFLOPS (DLFP16)	0.476 - 1.688 TFLOPS (FP16)
Power Efficiency	2.295 TOPS/W (INT8)	1.06 - 5.09 TOPS/W (INT8)	3.08 TOPS/W (INT16) - 50.6 TOPS/W (INT1)	3.42 - 6.83 TOPS/W (INT8, Dense)	1.1 - 1.4 TFLOPS/W (DLFP16)	**1.67 - 6.971 TFLOPS/W (FP16)**
Area Efficiency	1.133 TOPS/mm² (INT8)	0.0211 TOPS/mm² (INT8)	0.0022 TOPS/mm² (INT16)	0.2965 TFLOPS/mm² (FP16) 1.186 TOPS/mm² (INT8)	0.2 - 0.3 TFLOPS/mm² (DLFP16)	**0.156 - 0.555 TFLOPS/mm² (FP16)**
Latency [cycles]	256	-	-	-	-	25