



THE CHIPS TO SYSTEMS CONFERENCE

SHAPING THE NEXT GENERATION OF ELECTRONICS

JUNE 23-27, 2024

MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA



JUNE 23-27, 2024

MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA

Hybrid Tiled Vector Systolic Architecture to Accelerate Convolution on FPGAs

Jay Shah, Nanditha Rao

International Institute of Information Technology, Bangalore, India



Motivation

Objective: Accelerate Convolution (Conv) and Matrix Multiplications (Matmul) for CNNs and Transformers: FPGAs contain inherent parallel computing resources which can parallelize these operations efficiently

FPGA-based hardware accelerator: Challenges we address:

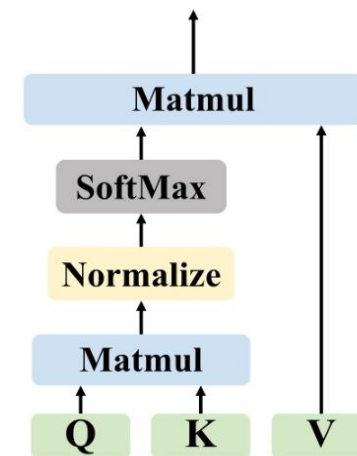
- Design a symmetric architecture with uniform systolic data flow and package it as a customisable IP
- Increase throughput
- Efficiently use all parallel hardware resources
- Reuse kernels and data: Reduce memory access latency

Systolic architecture IP: Questions we ask:

- How can we make the processing elements (PEs) more **parallel-computation capable and scalable**?
- Can **vectorization** help in reducing the memory redundancy while increasing the throughput?
- What **resources on the FPGA** should we map the Matrix multiplication or convolution operation to, considering the trade-off between utilization and throughput?
- **Automate the IP generation** for networks of different sizes: Can we **auto-map** the kernels and **partition the matrices** of different sizes to different systolic lane widths?

Matrix Convolution

Diagram illustrating Matrix Convolution. A 6x6 input matrix is convolved with a 3x3 kernel to produce a 6x6 output matrix. The input matrix is a 6x6 grid with values ranging from 0 to 9. The kernel is a 3x3 grid with values 1, 0, -1. The output matrix is a 6x6 grid with values ranging from -10 to 1. The diagram shows the input matrix, the kernel, and the resulting output matrix, with the operation labeled as $6 \times 6 \rightarrow 8 \times 8$.



Self-attention in Transformers

Key ideas

Parallel-computation capable and scalable PEs

- We propose a **Parameterizable Vector Systolic Architecture (VSA)** IP
- Accelerates image-kernel convolution and matrix multiplications

Increase throughput

- We design a **Tiled VSA**: Fully populates all FPGA resources
- Utilizes a novel convolution method [1] with reduced latency
- We explore PEs with **different vector lane widths**

Utilize FPGA resources efficiently

- We **partition the kernels and matrices** of larger sizes to operate on smaller VSAs
- We propose a **Hybrid Tile VSA**: Some tiles utilize LUTs and some DSPs

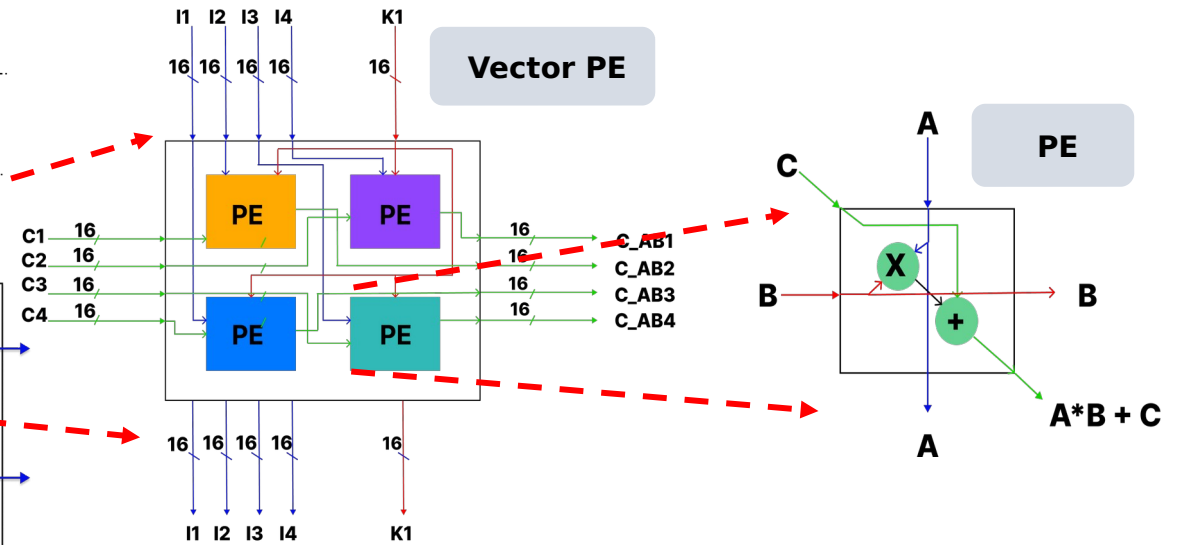
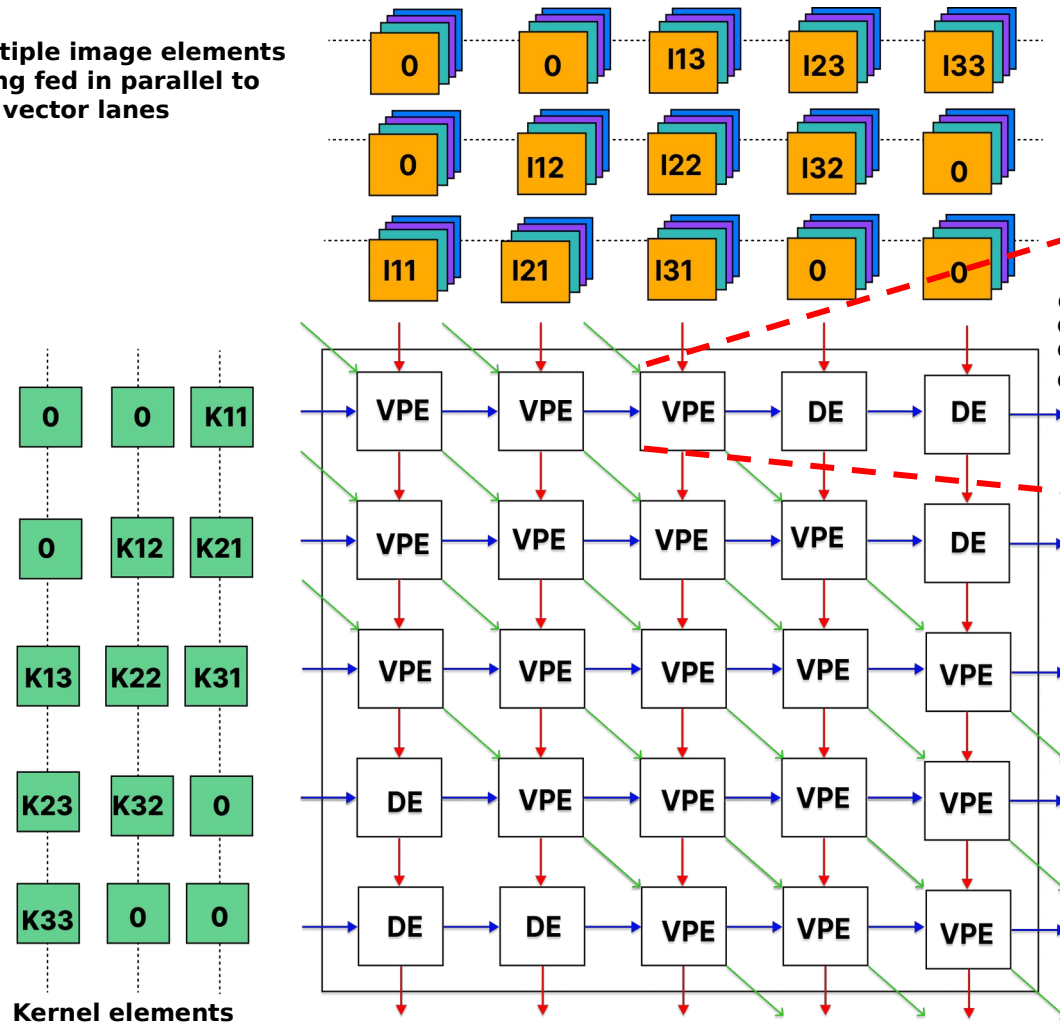
Automate the VSA IP generation

- We identify the **ideal lane width** of the Vector PE
- Kernels/matrices are **auto-mapped** to VSA tiles of different lane widths

Our **VSA** accelerator achieves **1165 GOPs and 1072 GOPs** for the optimal **Vector-6 and Vector-8** lane width Systolic architectures on the **Xilinx ZCU104** FPGA

Vector Systolic Accelerator

Multiple image elements being fed in parallel to the vector lanes

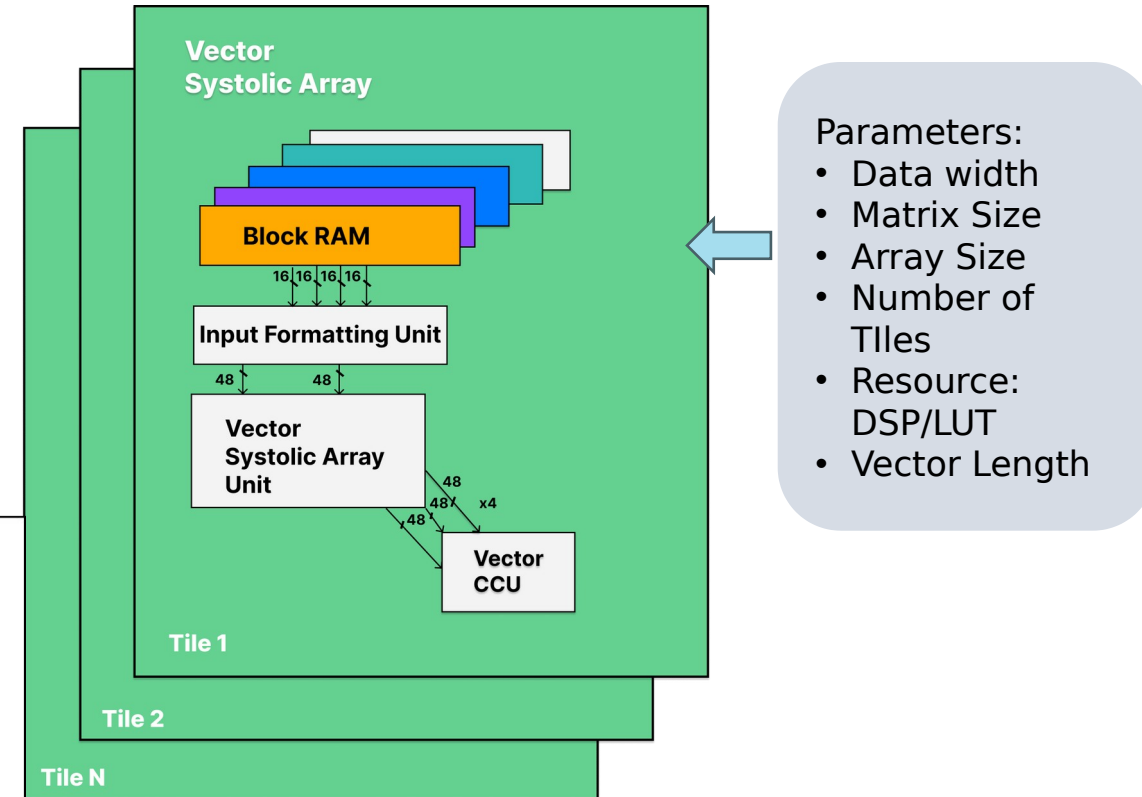
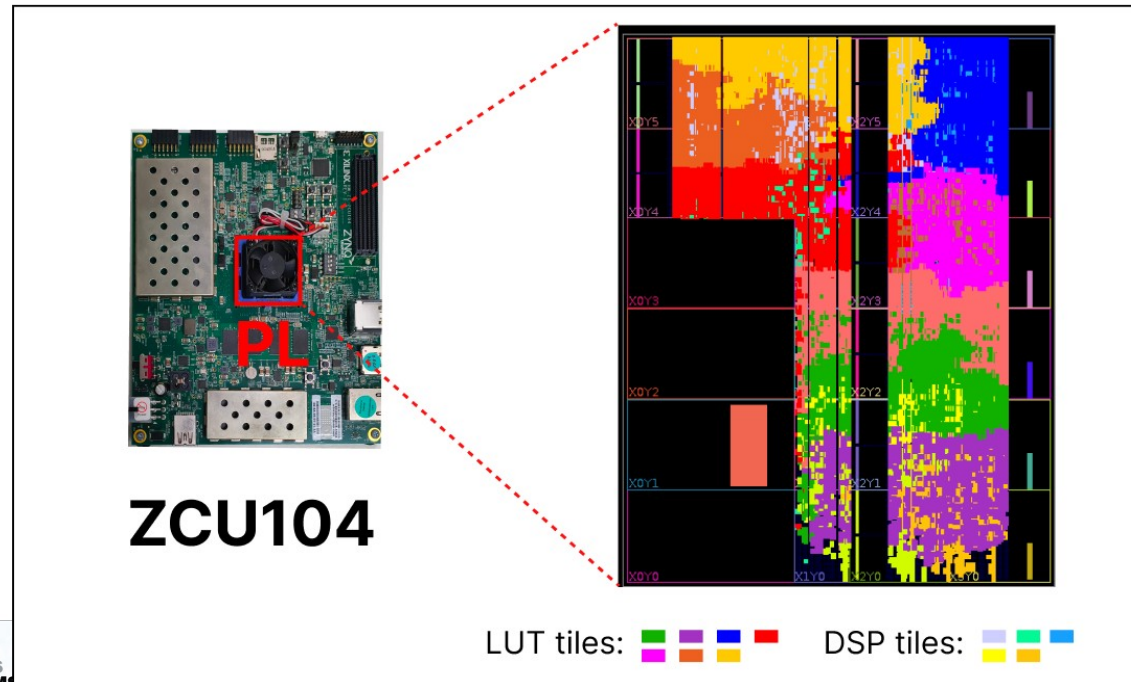


Parallel Computation:

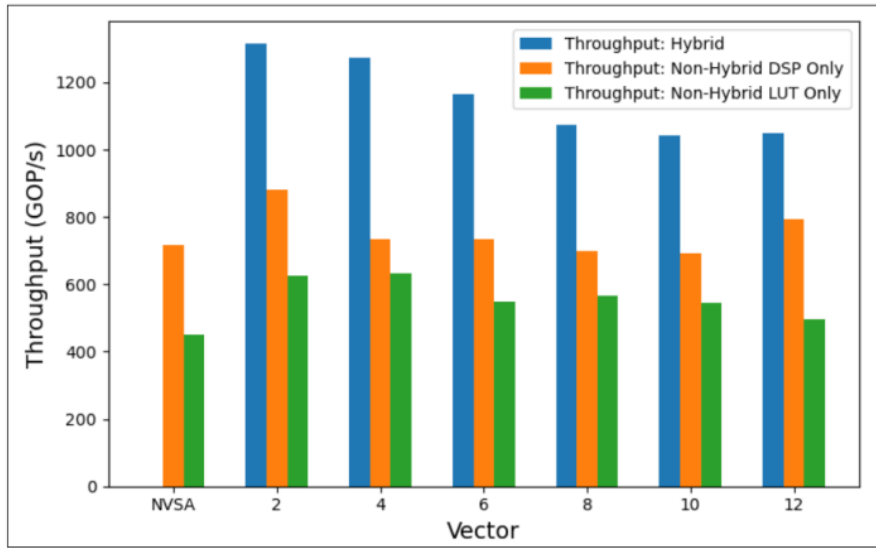
- Vector-PE that integrates multiple MAC units, operating on distinct image elements with the same kernels.
- Partition the image into multiple blocks, simultaneously processed by VSA
- Variable vector lane width: 2, 4, 6, 8, 10, and 12.

Hybrid Tiled VSA

- **Hybrid Tile-VSA:** DSPs being **scarce** resources on FPGA, it limits the number of tiles that can be instantiated
- Hybrid Structure: some times with DSP and others on LUTs, which increases the **parallel computation**, and increases the **overall throughput** at the cost of **reduced frequency**
- **Automation of IP Generation:** We identify the ideal lane width of the Vector PE
- Kernels/matrices are auto-mapped to VSA tiles of different lane widths



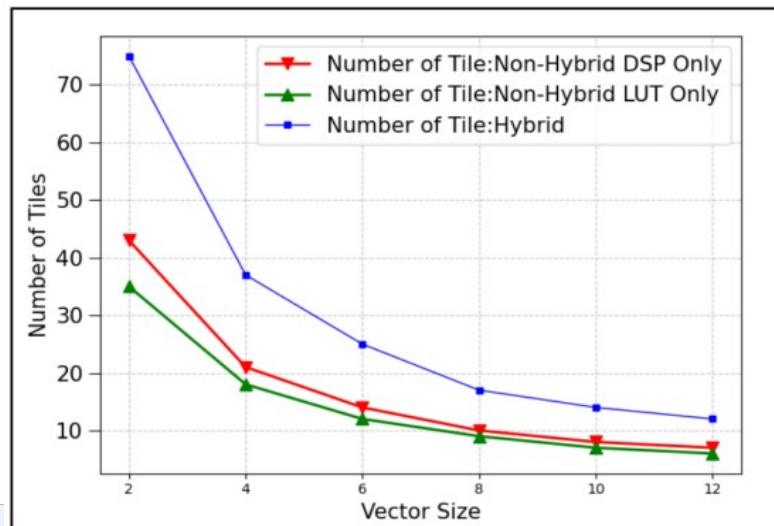
Experimental setup on FPGA and Results



Role of lane width on throughput

Hybrid approach demonstrates maximum throughput

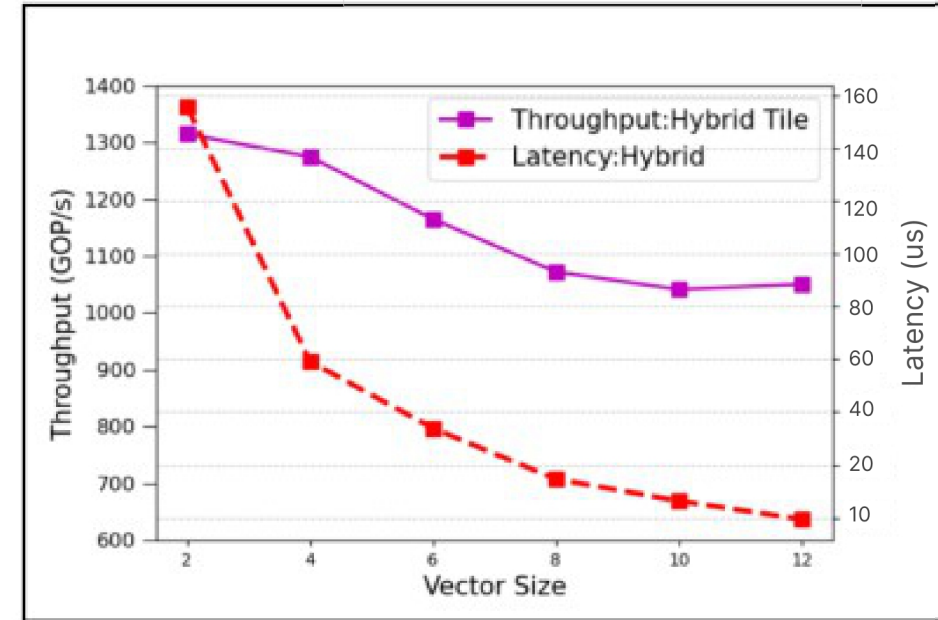
We recommend Vector-6 or Vector-8 with DSPs as an optimal configuration: Good throughput + enables 10-15 tiles



Hybrid multi-tile strategy: Some tiles with LUTs only and some with DSPs only

The Design1 with hybrid tiles can accommodate maximum tiles compared to the Design 2 and Design 3

Design 1: Hybrid Tile
Design 2: DSP-only Tile
Design 3: LUT-only Tile



Throughput versus latency across various vector lane widths

- Throughput decreases with an increase in lane-width due to reduction in the number of tiles.
- Latency decreases with an increase in lane-width, reducing the critical path, making it suitable for denser neural network implementations.

Related work and Conclusion

	Proposed Work	[1]	[7]	[3]	[10]	[8]	[17]	[18]
FPGA	Zynq ZCU104	ZCU104	ZCU102	VU9P	PYNQ-Z2	XC7K325T	ZCU102	ZCU102
Frequency (MHz)	203.81	150.015	214	263.1	120	150	300	200
Architectural novelty	Hybrid Vector Systolic	Novel Convolution	Multi-bit Booth vector	Systolic	Dual Line Systolic	Winograd 2D/3D	Mixed-pruning	Structured Sparse
LUTs	209K (91.01%)	185K (80.70%)	216K (78.90%)	1939K (75%)	-	3,490(13.7%)	72K(26.3%)	184K (67%)
DSPs	1634 (94.56%)	0 (0%)	115 (82.14%)	2060 (81.74%)	-	1,714 (89.3%)	654 (26%)	2520 (100%)
BRAMs	60 (19.39%)	67 (48.2%)		(17%)		1824 (95%)		1460 (80%)
Data type/bit width	int16 and 8	int16	int2-8	int32	int8	int8-16	int16	int8
Throughput (GOPs)	1050.3-1314.5	379.24	784.16	673.3	120	334.8	318.22	990

With Vector 2 and 4

Summary

- We introduced a Vector Systolic Accelerator IP with reconfigurable lane-width Processing Elements (PEs)
- We proposed a hybrid multi-tile strategy on FPGA, partitioning resources to have some tiles using DSPs and others utilizing remaining LUT resources --> Achieved improvements of 2.3x and 1.49x over its non-Hybrid counterparts for Vector-6 and Vector-8 respectively
- We replicated the Tiled-VSAs to entirely populate the FPGA -- > Throughputs achieved: 1165 GOPs and 1072 GOPs with convolution latency of 0.923 ns and 0.682 ns for Vector-6 and 8 on ZCU104

[1] Devaraddi, et al, DSD'22
 [3] L. Jia, et al, IEEE Micro'20 [7] M. Huang, et al, IEEE Trans. Circuits Syst.-22
 [8] Chengcheng Huang, et al, FPL'21
 [11] X. Liu, et al, ASAP'21 [17] X. Chang, et al, IEEE Trans. Circuits Syst.-21
 [10] P. Xue, et al, "Dual-Line-Systolic Array for High Performance CNN Accelerator," - FCCM
 [17] X. Chang, et al, "A mixed-pruning based framework for embedded convolutional neural network acceleration," - TCAS-I
 [18] C. Zhu, et al, "An efficient hardware accelerator for structured sparse convolutional neural networks on FPGAs," - IEEE VLSI

Future work

- Automate the vector tile section based on the kernel/matrix size, resource availability, and desired throughput.
- Reconfigure the VSA for large matrix multiplications