Hardware Acceleration of CNNs with Coherent FPGAs

Md Syadus Sefat
Texas State University
San Marcos, TX
m_s597@txstate.edu

Semih Aslan
Texas State University
San Marcos, TX
aslan@txstate.edu

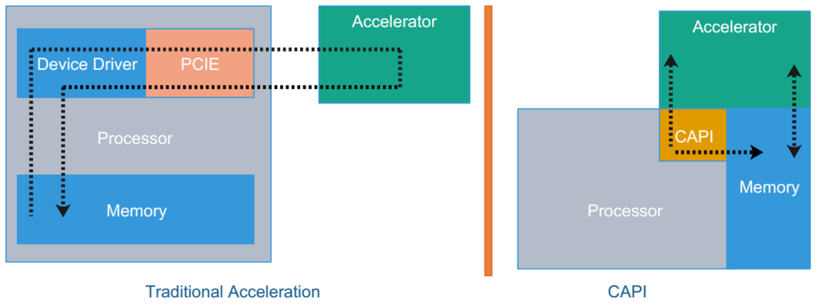
Apan Qasem
Texas State University
San Marcos, TX
apan@txstate.edu

This paper describes a new flexible approach to implementing energy-efficient CNNs on FPGAs. Our design leverages the Coherent Accelerator Processor Interface (CAPI) which provides a cache-coherent view of system memory to attached accelerators. Convolution layers are formulated as matrix multiplication kernels and then accelerated on a CAPI-supported Kintex FPGA board. Our implementation bypasses the need for device driver code and significantly reduces the communication and I/O transfer overhead. To improve the performance of the entire application, not just the convolution layers, we propose a collaborative model of execution in which the control of the data flow within the accelerator is kept independent, freeing-up CPU cores to work on other parts of the application. For further performance enhancements, we propose a technique to exploit data locality in the cache, situated in the CAPI Power Service Layer (PSL). Finally, we develop a resource-conscious implementation for more efficient utilization of resources and improved scalability.

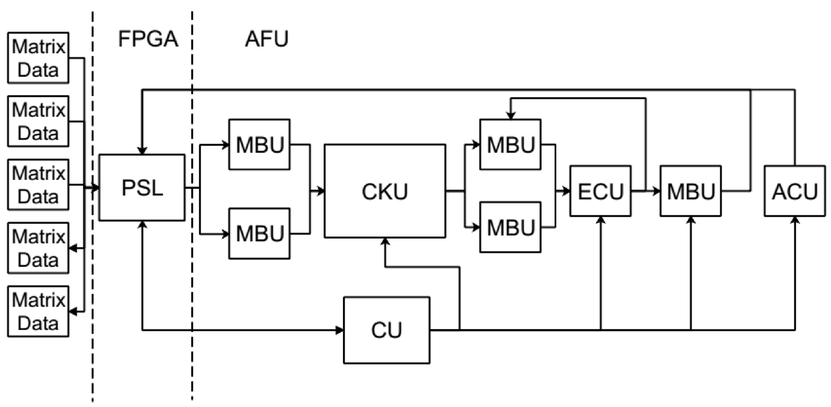
Contributions:

- New flexible approach to implement DNN in a heterogeneous system
- A new hardware architecture for collaborative DNN implementation
- Optimization of the hardware unit to implement a resource-conscious design for matrix-multiply on the FPGA
- Design of a batched computational unit to work with large weight matrices of DNN

Coherent and Non-Coherent Accelerator:

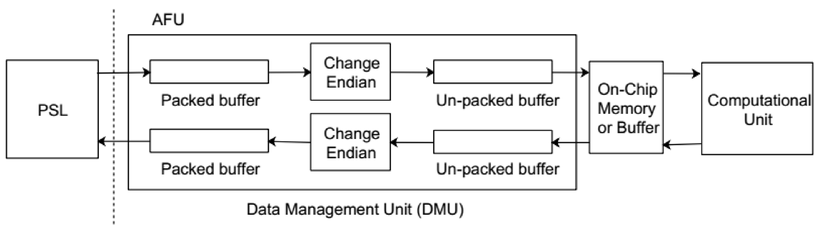


Hardware Architecture:

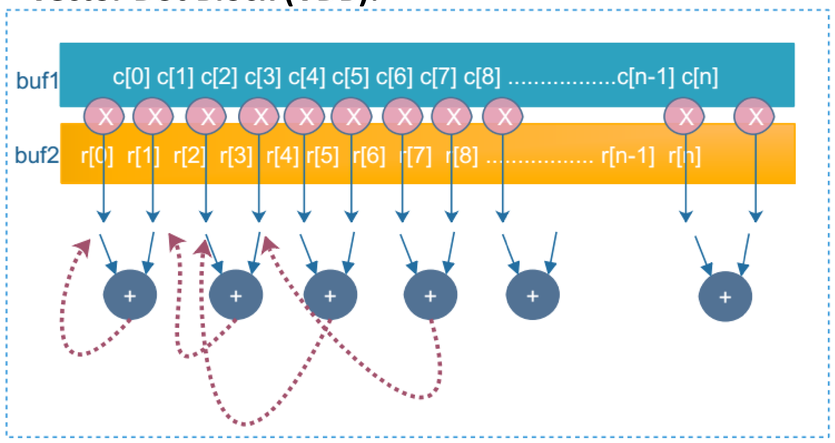


PSL – Power Service Layer
MBU – Multiport Buffer Unit
CKU – Computational Kernel Unit
ECU – Extra Computational Unit
ACU – Address Calculator Unit

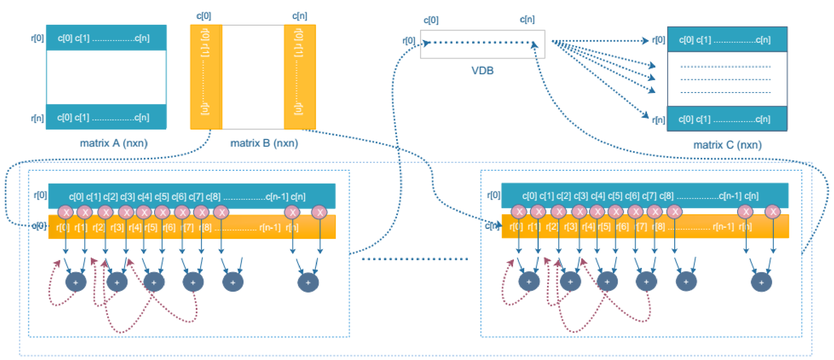
Simplified Dataflow:



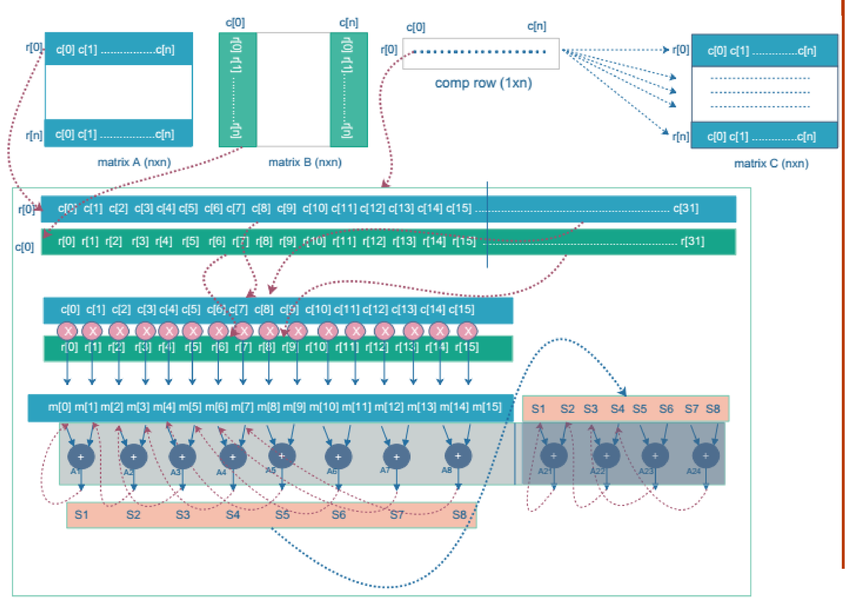
Vector Dot Block (VDB):



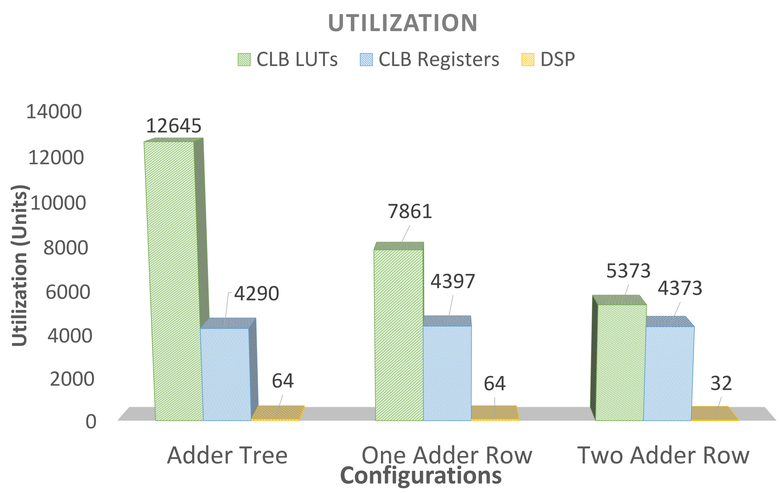
Computation with One Adder Row:



Computation with Two Adder Row, Resource Conscious Implementation (VDB_RC):



FPGA Resource Utilization:



Performance Comparison:

Table 1: Comparison with previous works

	[6]	[2]	Ours
year	2015	2016	2018
Platform	Virtex7 VX458t	Zynq XC7Z045	CAPI KU115
Clock (MHz)	100	150	250
Quantization	32-bit float	16-bit fixed	32-bit float
Performance (GOP/s)	61.62	136.97	155.08
Power(W)	18.61	9.63	9.82
Power Efficiency (GOP/s/W)	3.31	14.22	15.80

Conclusion and Future Work:

- To the best of our knowledge, this is the first paper that implements CNN operations on a CAPI enabled hardware accelerator
- Preliminary experiments show substantial performance gains over
- traditional methods and make the case for further exploration of hardware accelerated design of CNN and other deep learning applications.
- In future, we plan to deploy all performance-critical CNN operations in a multi-FPGA heterogeneous environment

[2] J. Qiu and et at. Going deeper with embedded fpga platform for convolutional neural network. In *ACM/SIGDA*, pages 26–35. ACM, 2016.

[6] C. Zhang and et al. Optimizing fpga-based accelerator design for deep convolutional neural networks. In *ACM/SIGDA*, pages 161–170. ACM, 2015.