# Optimizing FPGA-based Accelerator Design for Deep Convolutional Neural Networks

C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao and J. Cong.

Presented by Maya Gibson

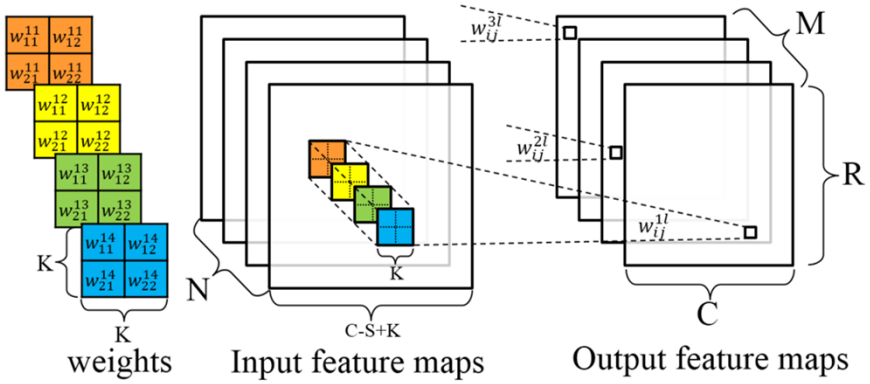
# Convolutional Neural Network (CNN)

- Feedforward process for recognition
- Backward path for training
- Composed of two components:
  - A feature extractor
  - A classifier
- Composed of multiple computation layers

[1,2]

# Computation Of A Convolutional Layer

## Graph of a convolutional layer




## Pseudo code

```
for(row=0; row<R; row++) {
 for(col=0; col<C; col++) {
  for(to=0; to<M; to++) {
   for(ti=0; ti<N; ti++) {
    for(i=0; i<K; i++) {
     for(j=0; j<K; j++) {
   L:  output_fm[to][row][col] +=
          weights[to][ti][i][j]*
          input_fm[ti][S*row+i][S*col+j];
} } } } } }
```

[1]

# CNN Applications

- Image Processing
- Video Surveillance
- Mobile Robot Vision
- Natural Language Processing

Fast growth of modern technology based on deep learning algorithms has generated new research & implementations [1,2,3]

# CNN Applications

- For any CNN algorithm implementation, there are a lot of potential solutions that result in a vast design space for exploration.
- There is up to a 90% performance difference between two different solutions with the same logic resource utilisation

An efficient method is of top priority for exploration of FPGA based CNN design space.

[1]

# Field-Programmable Gate Arrays (FPGA)

- General purpose processors are not efficient enough for CNN implementations.
- Integrated circuit with the following advantages
  - Good performance
  - High energy efficiency
  - Capability of reconfiguration

[1,2]

# Problem!

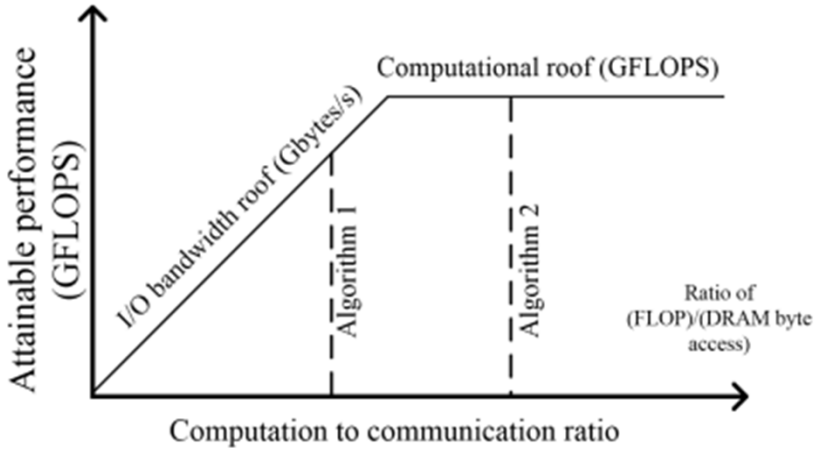
Although current FPGA accelerators have demonstrated better performance over generic processors, the accelerator design space has not been explored well enough - there are more efficient solutions that have yet to be discovered!

# Introduction Summary

- What is your topic?
- Where does it fit in the Computer Science discipline?
- What drives the development of the topic (or what are industry's needs), what is its background?
- What distinguishes the topic from associated topics?

# The Roofline Model

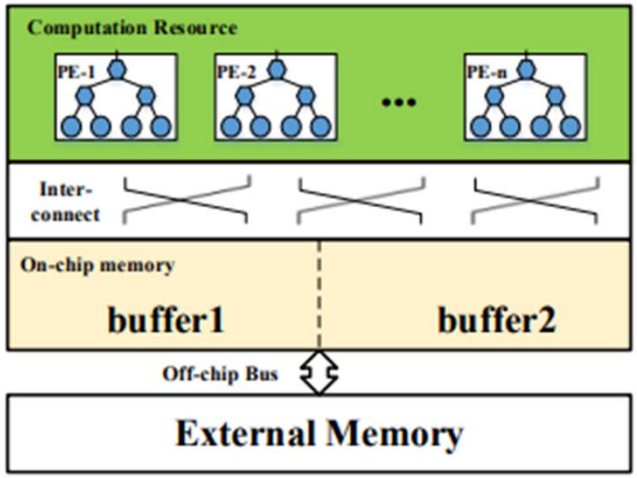
- Relates system performance to off-chip memory traffic and the peak performance provided by the hardware platform.




[1]

# Exploring Accelerator Designs

## Overview of our accelerator structure




[1]

# Exploring Accelerator Designs

Computation Optimisation

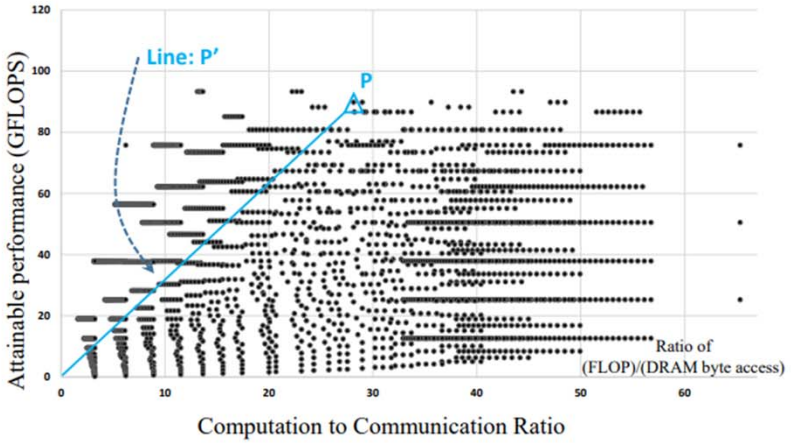
- Loop Unrolling
- Loop Pipelining
- Tile Size Selection

Memory Access Optimisation

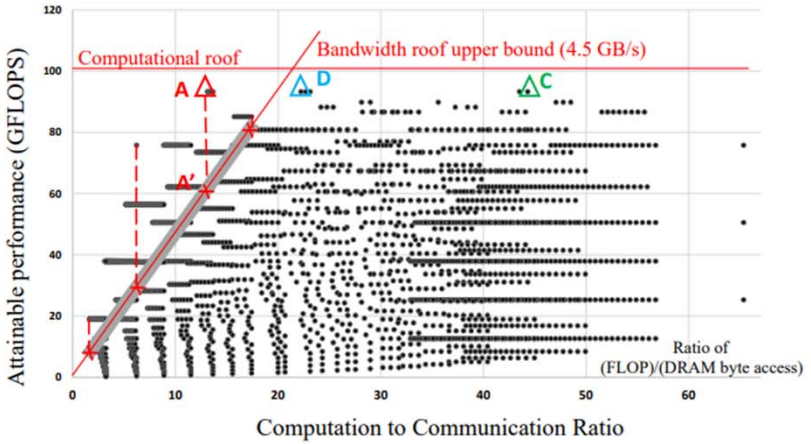
- Local Memory Promotion
- Loop Transformations for Data Reuse
- CTC Ratio.

[1]

# Design Space Exploration




(a) Design space of all possible designs



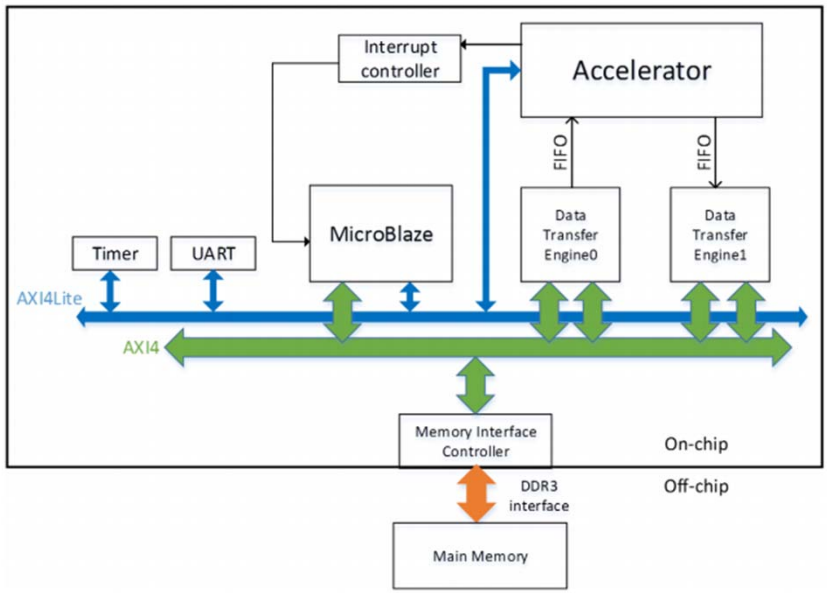

(b) Design space of platform-supported designs

All possible designs

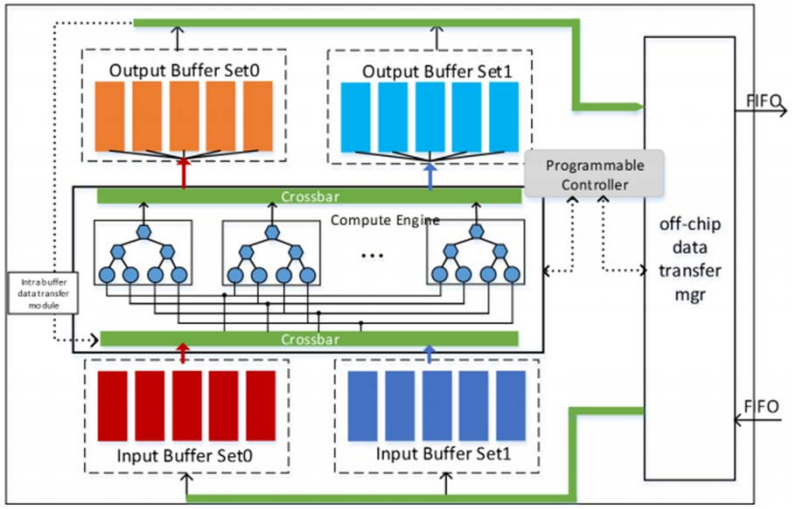
Platform-supported designs

[1]

# Implementation Details




Implementation Overview

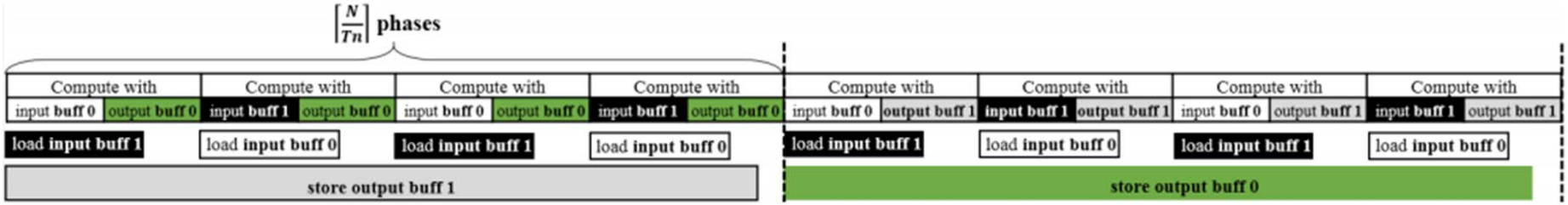



Proposed Accelerator

[1]

# Timing graph

- On-chip buffers overlap data transfer time with computation
- Used to increase the bandwidth utilization




[1]

# Evaluation

- Accelerator design implemented with Vivado HLS, on a VC707 board in C with a working frequency of 100MHz.
- The software implementation uses a Intel XEON CPU E5-2430 (@2.20GHz) with a 15MB cache.
- The results are very successful!

[1]

# Comparison to previous implementations

| | ICCD2013 [12] | ASAP2009 [14] | FPL2009 [6] | FPL2009 [6] | PACT2010 [2] | ISCA2010 [3] | Our Impl. |
|---|---|---|---|---|---|---|---|
| Precision | fixed point | 16bits fixed | 48bits fixed | 48bits fixed | fixed point | 48bits fixed | 32bits float |
| Frequency | 150 MHz | 115 MHz | 125 MHz | 125 MHz | 125 MHz | 200 MHz | 100 MHz |
| FPGA chip | Virtex6 VLX240T | Virtex5 LX330T | Spartan-3A DSP3400 | Virtex4 SX35 | Virtex5 SX240T | Virtex5 SX240T | Virtex7 VX485T |
| FPGA capacity | 37,680 slices 768 DSP | 51,840 slices 192 DSP | 23,872 slices 126 DSP | 15,360 slices 192 DSP | 37,440 slices 1056 DSP | 37,440 slices 1056 DSP | 75,900 slices 2800 DSP |
| LUT type | 6-input LUT | 6-input LUT | 4-input LUT | 4-input LUT | 6-input LUT | 6-input LUT | 6-input LUT |
| CNN Size | 2.74 GMAC | 0.53 GMAC | 0.26 GMAC | 0.26 GMAC | 0.53 GMAC | 0.26 GMAC | 1.33 GFLOP |
| Performance | 8.5 GMACS | 3.37 GMACS | 2.6 GMACS | 2.6 GMACS | 3.5 GMACS | 8 GMACS | 61.62 GFLOPS |
| | 17 GOPS | 6.74 GOPS | 5.25 GOPS | 5.25 GOPS | 7.0 GOPS | 16 GOPS | 61.62 GOPS |
| Performance Density | 4.5E-04 GOPs/Slice | 1.3E-04 GOPs/Slice | 2.2E-04 GOPs/Slice | 3.42E-04 GOPs/Slice | 1.9E-04 GOPs/Slice | 4.3E-04 GOPs/Slice | 8.12E-04 GOPS/Slice |

[1]

# Comparisons to CPU

| float | CPU 2.20GHz (ms) | | FPGA | |
|---|---|---|---|---|
| 32 bit | 1thd -O3 | 16thd -O3 | (ms) | GFLOPS |
| layer 1 | 98.18 | 19.36 | 7.67 | 27.50 |
| layer 2 | 94.66 | 27.00 | 5.35 | 83.79 |
| layer 3 | 77.38 | 24.30 | 3.79 | 78.81 |
| layer 4 | 65.58 | 18.64 | 2.88 | 77.94 |
| layer 5 | 40.70 | 14.18 | 1.93 | 77.61 |
| **Total** | 376.50 | 103.48 | 21.61 | - |
| **Overall GFLOPS** | **3.54** | **12.87** | **61.62** | |
| **Speedup** | **1.00x** | **3.64x** | **17.42x** | |

Performance Comparison

## Power Consumption and Energy

| | Intel Xeon 2.20GHz | | FPGA |
|---|---|---|---|
| | 1 thread -O3 | 16 threads -O3 | |
| Power (Watt) | 95.00 | 95.00 | 18.61 |
| Comparison | 5.1x | 5.1x | 1x |
| Energy (J) | 35.77 | 9.83 | 0.40 |
| Comparison | 89.4x | 24.6x | 1x |

[1]

## Pros and Cons

- Other application accelerators do not balance the physical limitations between bandwidth and computational power. This solution is balanced.
- The implemented CNN accelerator achieves a performance of 61.62 GFLOPS - the highest performance among existing accelerators.

[1]

## Conclusion

- The paper describes a “roofline-model-based method for convolutional neural network FPGA acceleration”.
  - Through optimising, modeling using the roofline model, finding the best layer/cross layer design and then through actual implementation.
- A unique implementation that sheds light into the design space, further solutions can build on this implementation.

# References

C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao and J. Cong, “Optimizing FPGA-based Accelerator Design for Deep Convolutional Neural Networks,” in Proc. of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA ’15), Association for Computing Machinery, New York, NY, USA, 2015, pp. 161–170, doi: 10.1145/2684746.2689060.

W. Rawat and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review", Neural Computation, vol. 29, no. 9, pp. 2352-2449, 2017. Available: 10.1162/neco_a_00990.

# References

C. Farabet, C. Poulet, J. Y. Han and Y. LeCun, "CNP: An FPGA-based processor for Convolutional Networks," 2009 International Conference on Field Programmable Logic and Applications, Prague, 2009, pp. 32-37, doi: 10.1109/FPL.2009.5272559.