Convolution Hardware Accelerator

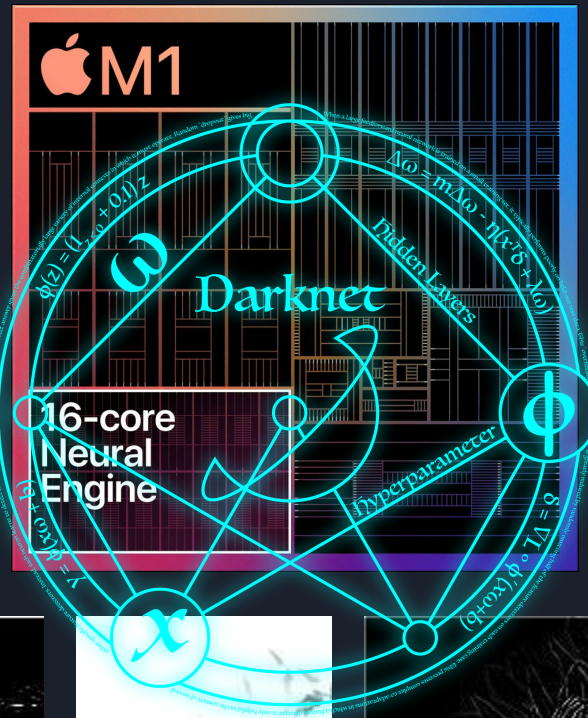
Oscar Kellner

Introduction

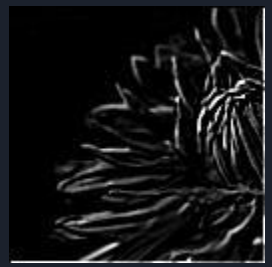
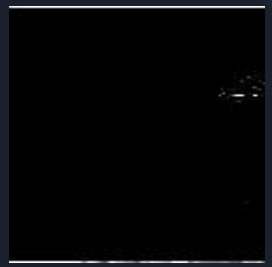
Machine Learning and Image Processing is cool

Kind of fast in optimized software, but can it be faster with the help of ***dedicated*** hardware?

Goal: design convolution hardware accelerator for use within existing machine learning framework (Darknet)



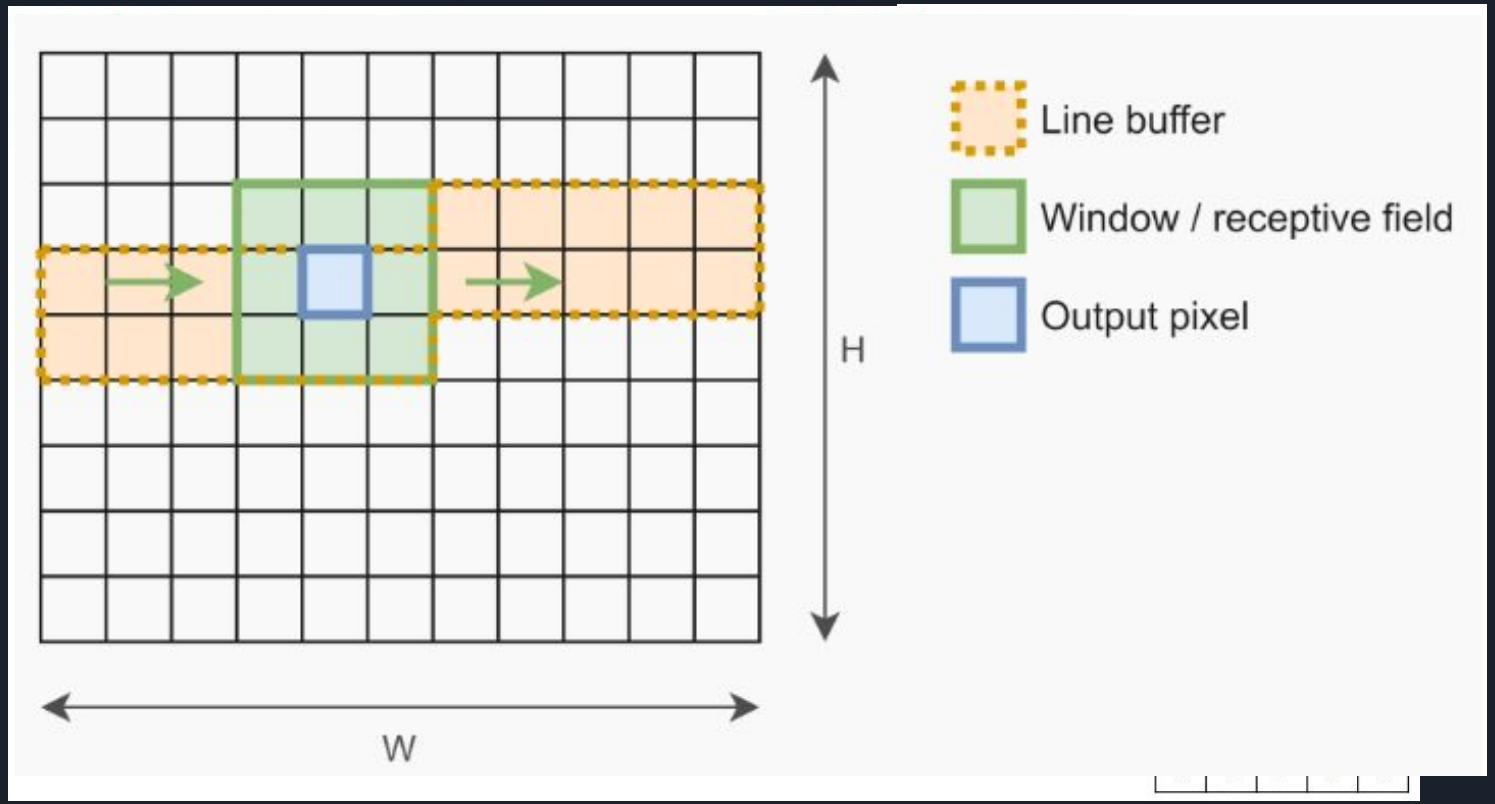


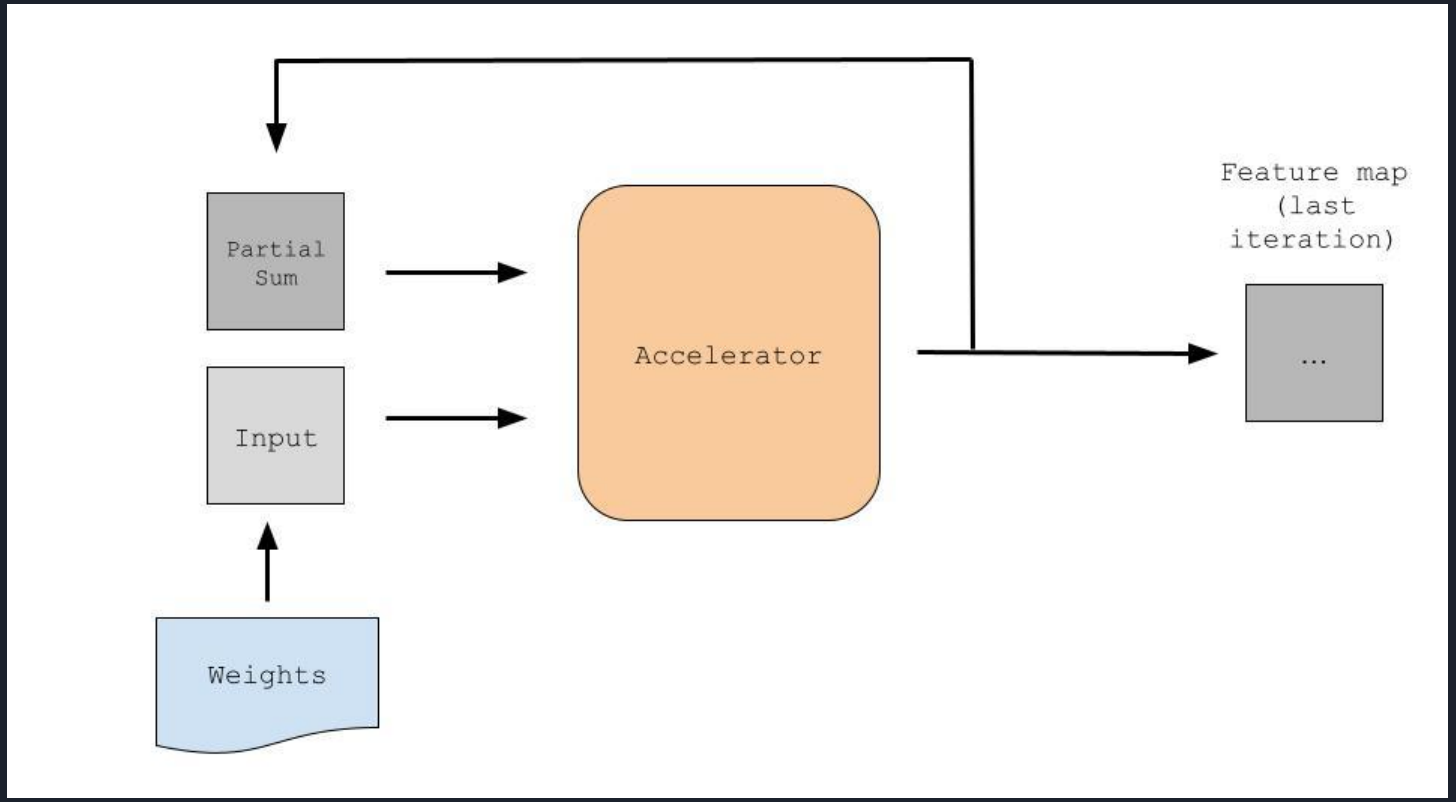
Tasks

- Design new HW friendly algorithm
- Synthesize and optimize hardware
- Verify with Python
- Interfacing in C
- Integrate with Darknet

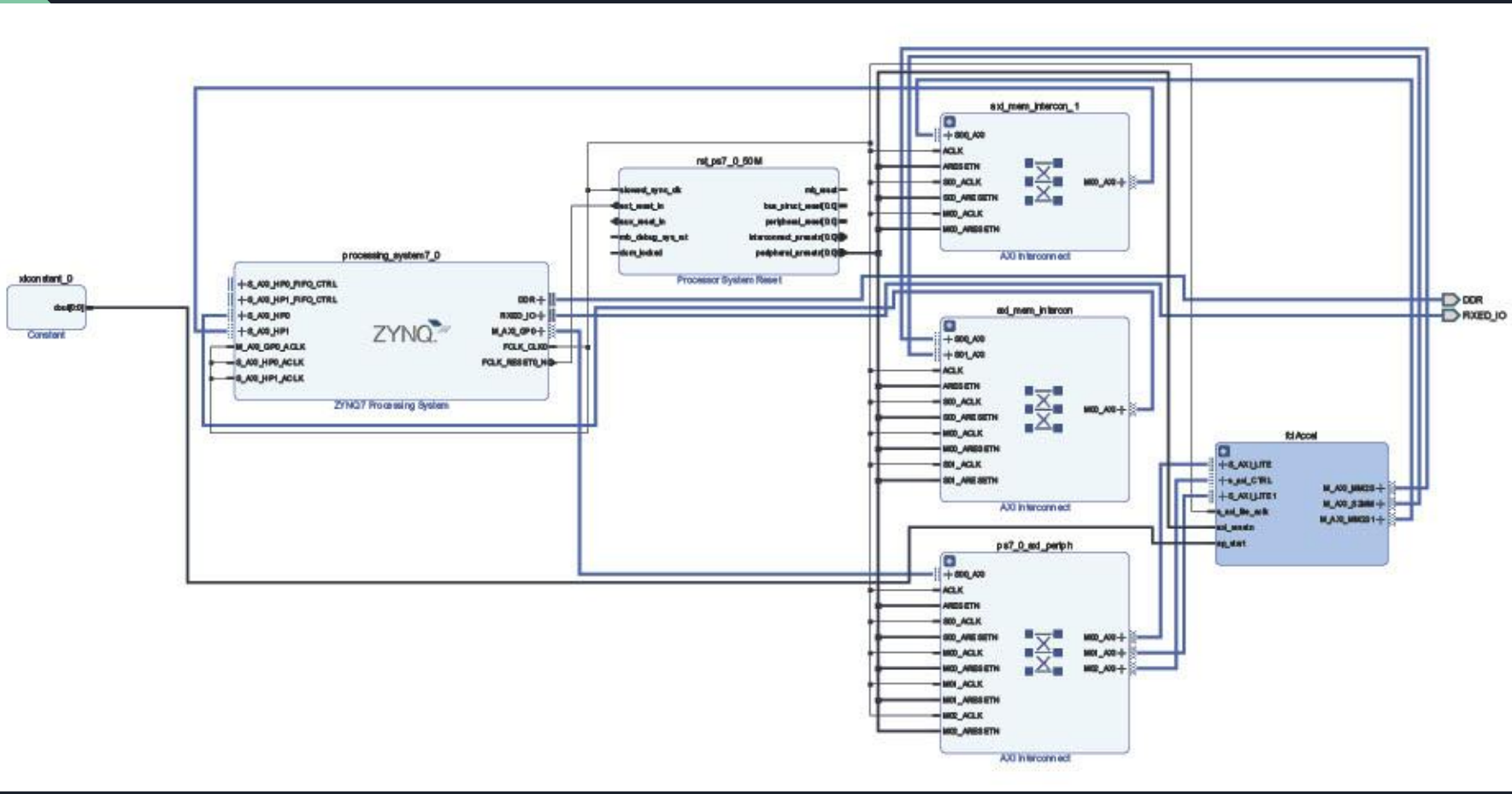
Convolution Overview



Designs


Partial Sum
Input
Weights
Accelerator
Feature map (last iteration)
...

Block Diagram



Performance (first layer of tiny.cfg only)

Unoptimized hardware w/ Python:

```
Hardware execution time:  12.059700012207031
1.7432472705841064 seconds were spent filling the buffers.
```

Optimized hardware w/ Python:

```
Hardware execution time:  3.5553882122039795
1.7212183475494385 seconds were spent filling the buffers.
```

About 3.4 times speedup after optimization, 5.6 times when discounting loading overhead

Optimized hardware in C:

```
Time taken: 2.6864829063 seconds.
```

About 1.3 times faster than Python (not accounting for filling buffers)

What about the whole network?

SEGMENTATION FAULTS!

Darknet will not run on the PYNQ!

When one segfault is fixed, more arise.

It may be possible to fix them all, but there wasn't enough time left.

Lessons Learned

- Test everything first before integration!
- Too much time spent hastily debugging ‘broken’ designs
- Communicate more often

References

https://medium.datadriveninvestor.com/convolutional-neural-networks-3b241a5da51e

https://m-alcu.github.io/blog/2018/01/13/neural-layers/

https://towardsdatascience.com/convolutional-layer-hacking-with-python-and-numpy-e5f64812ca0c

https://pjreddie.com/darknet/

https://basile.be/2019/03/18/a-tutorial-on-non-separable-2d-convolutions-in-vivado-hls/