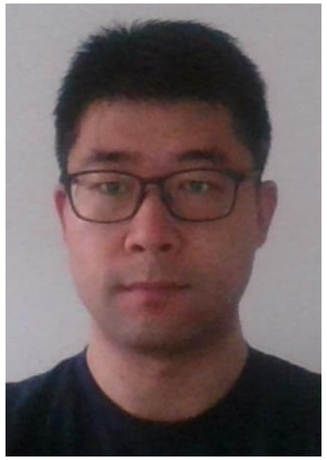
A Unified Hardware Architecture for Convolutions and Deconvolutions in CNN



Lin Bai, Yecheng Lyu and Xinming Huang

Worcester Polytechnic institute

2020 IEEE International Symposium on Circuits and Systems

Virtual, October 10-21, 2020

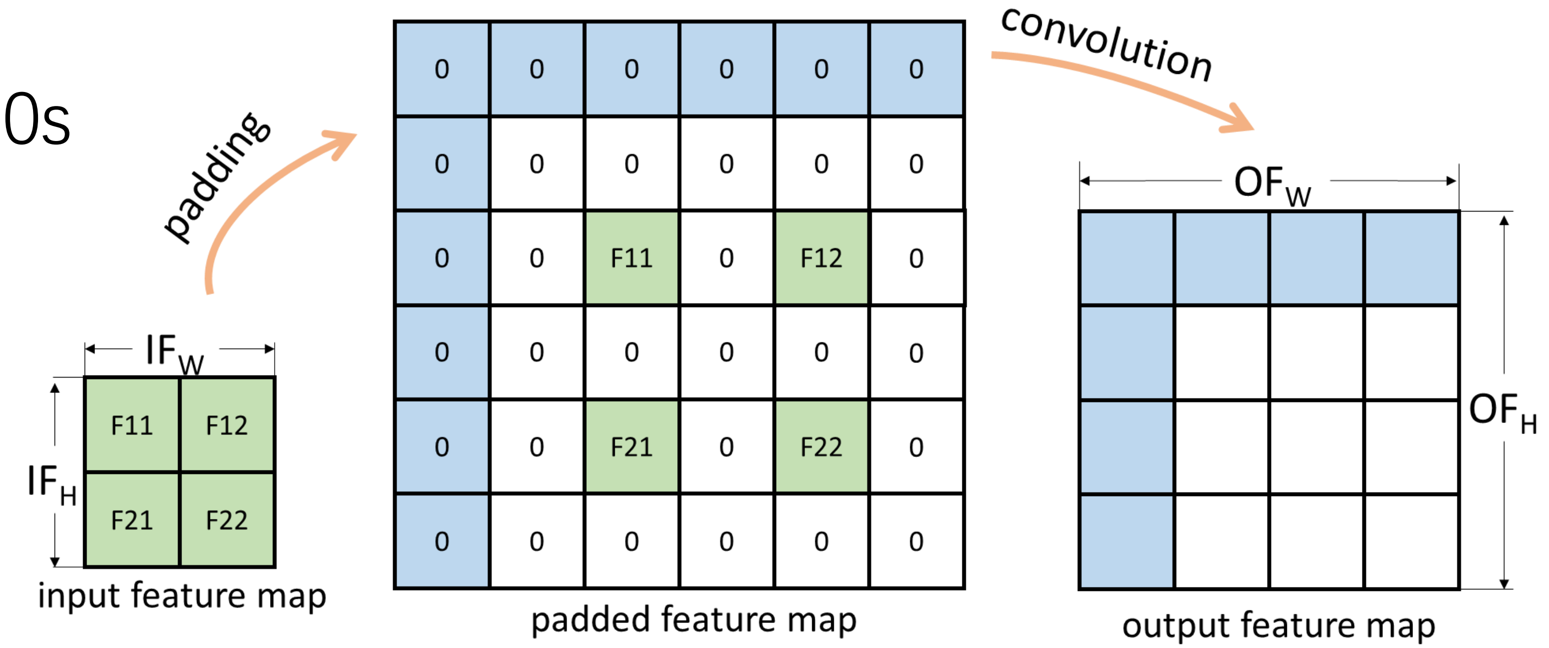
Overview

- Introduction
- Operation of deconvolution
- Optimization of deconvolution
- SegNet-basic Structure
- Hardware architecture

Operation of deconvolution

- Naïve deconvolution
  - Step 1: padding
  - Step 2: convolution

Too much multiply with 0s



Operation of deconvolution

- GPU deconvolution
  - Step 1: im2col
  - Step 2: matrix multiplication

Still much multiply with 0s

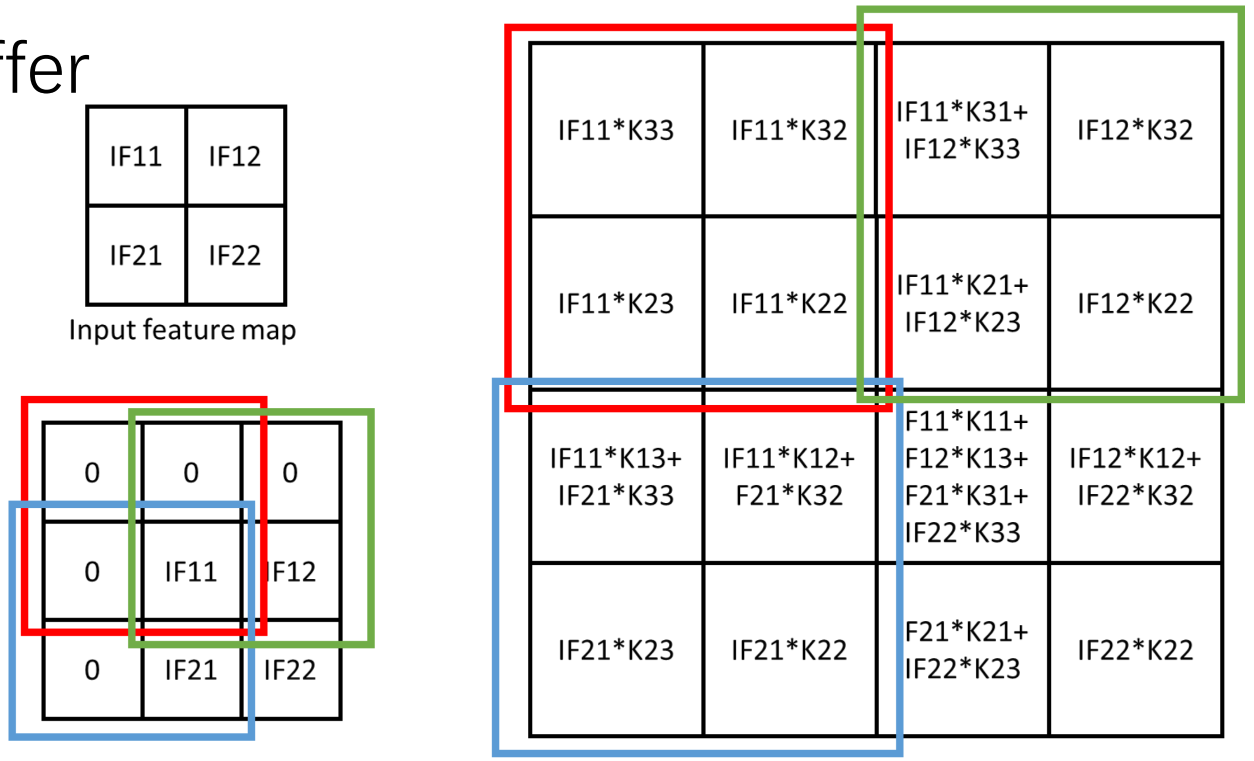
$$\begin{pmatrix}
w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 & 0 \\
0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 \\
0 & 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2}
\end{pmatrix}$$

Optimization for Hardware

- Re-use multiplication in SIMD accelerator
- Get rid of multiplication with 0s
- No extra memory needed to buffer

Optimization for Hardware

- Re-use multiplication in SIMD accelerator
- Get rid of multiplication with 0s
- No extra memory needed to buffer



IF11	IF12
IF21	IF22

Input feature map

0	0	0
0	IF11	IF12
0	IF21	IF22

padded feature map

IF11*K33	IF11*K32	IF11*K31+ IF12*K33	IF12*K32
IF11*K23	IF11*K22	IF11*K21+ IF12*K23	IF12*K22
IF11*K13+ IF21*K33	IF11*K12+ F21*K32	F11*K11+ F12*K13+ F21*K31+ IF22*K33	IF12*K12+ IF22*K32
IF21*K23	IF21*K22	F21*K21+ IF22*K23	IF22*K22

output feature map

Optimization for Hardware

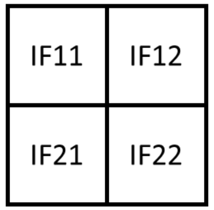
- Re-use multiplication in SIMD accelerator
- Get rid of multiplication with 0s
- No extra memory needed to buffer

$$OF_{11} = IF_{11} \cdot K_{11} + IF_{12} \cdot K_{13} + IF_{21} \cdot K_{31} + IF_{22} \cdot K_{33} \quad (1)$$

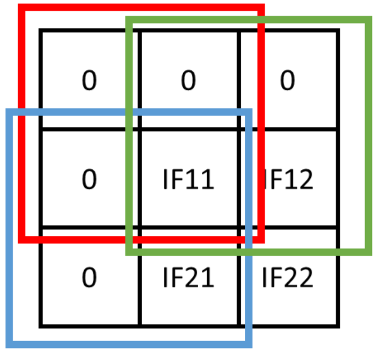
$$OF_{12} = IF_{12} \cdot K_{12} + IF_{22} \cdot K_{32} \quad (2)$$

$$OF_{21} = IF_{21} \cdot K_{21} + IF_{22} \cdot K_{23} \quad (3)$$

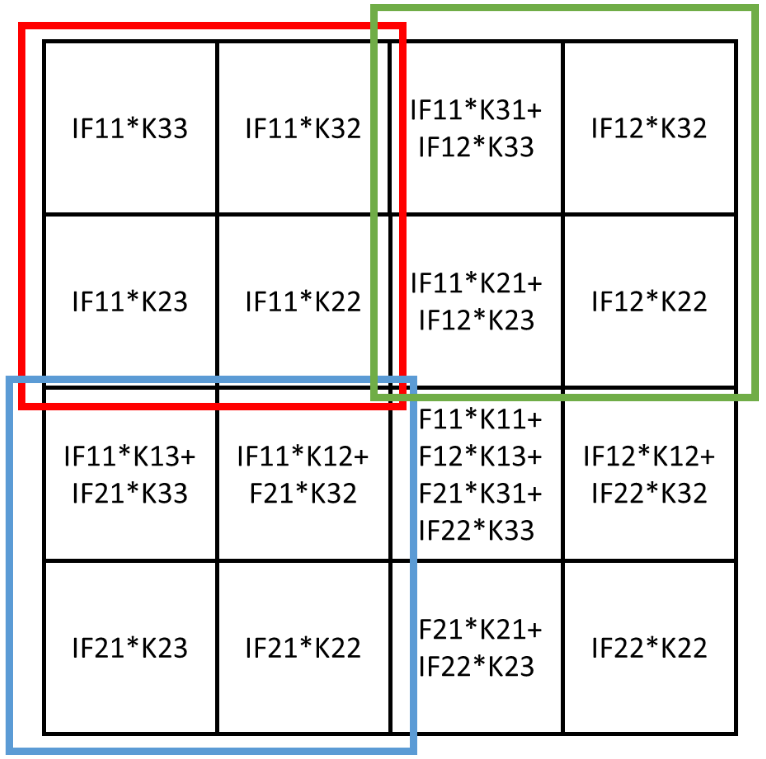
$$OF_{22} = IF_{22} \cdot K_{22} \quad (4)$$



Input feature map



padded feature map



output feature map

Optimization for Hardware

- Deconvolution
  - 9 multiplications
  - 5 adding

- Convolution
  - 9 multiplication
  - 8 adding

Optimization for Hardware

- Deconvolution
  - 9 multiplications
  - 5 adding

- Convolution
  - 9 multiplication
  - 8 adding

Possible to share multipliers !

Optimization for Hardware

- Deconvolution
  - 9 multiplications
  - 5 adding
- Convolution
  - 9 multiplication
  - 8 adding

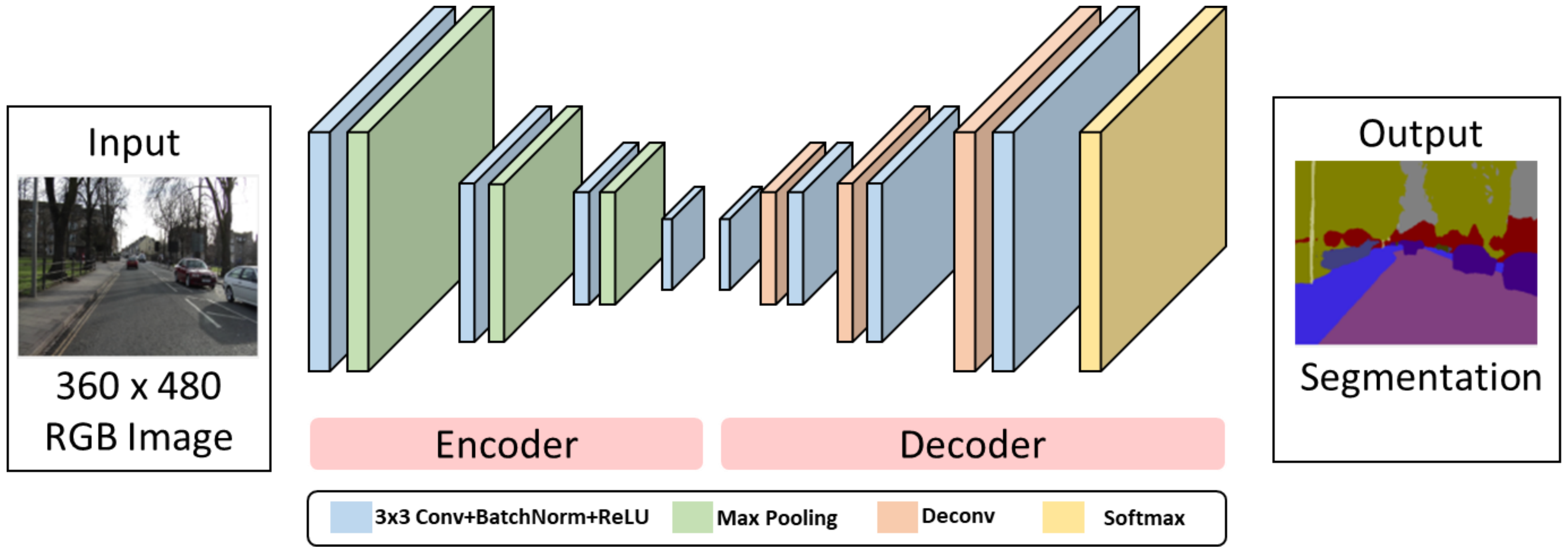
Possible to share multipliers !

- Deconvolution
  - 2x2 input
  - 2x2 output
- Convolution
  - 3x3 input
  - 1 output

Need reshape input and output!

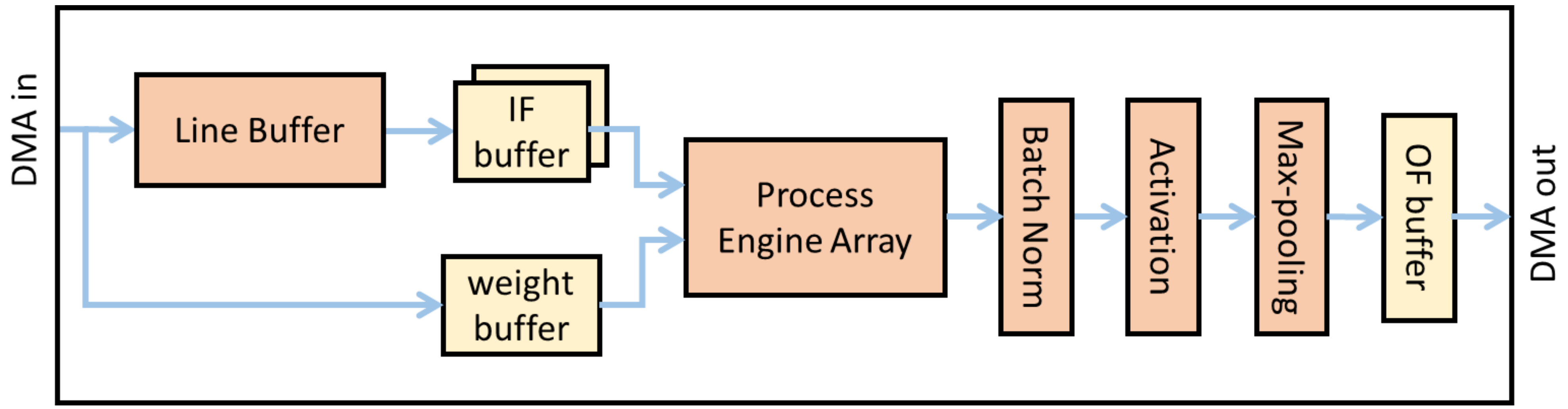
SegNet-Basic Structure

- Encoder-decoder structure, no skip connection
- Up-sampling is replaced by deconvolution
- Main operations: convolution, max pooling, deconvolution



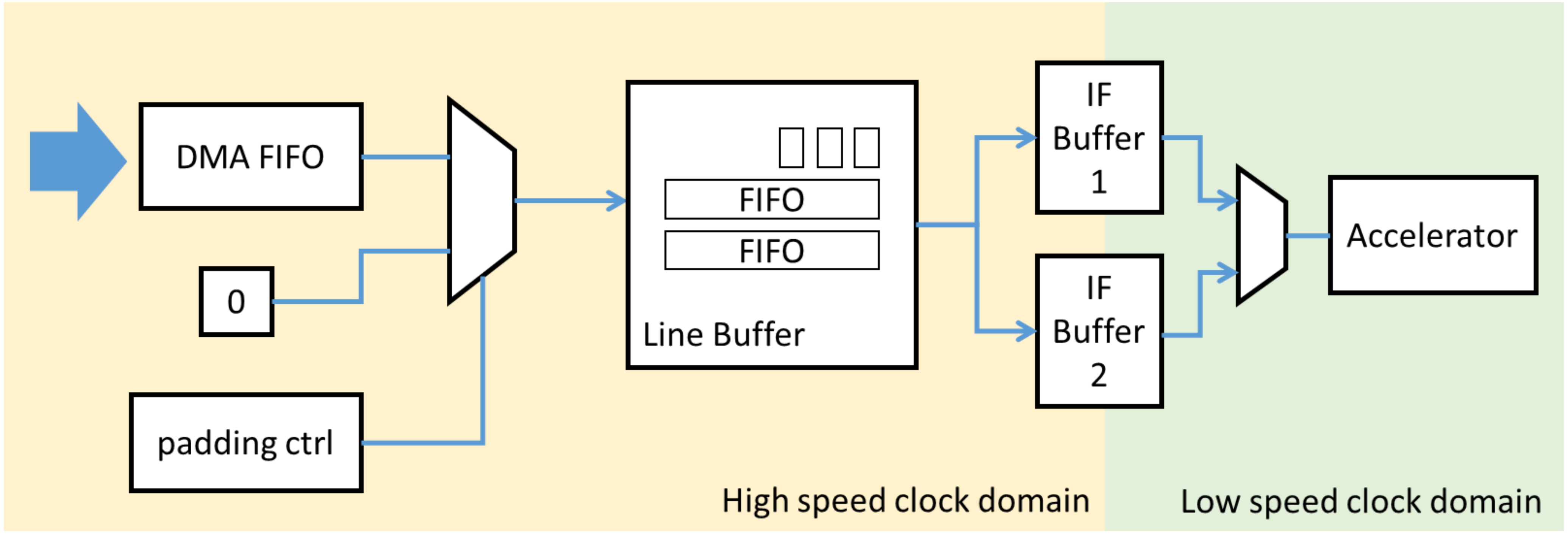
Hardware Architecture

- Overview of accelerator structure



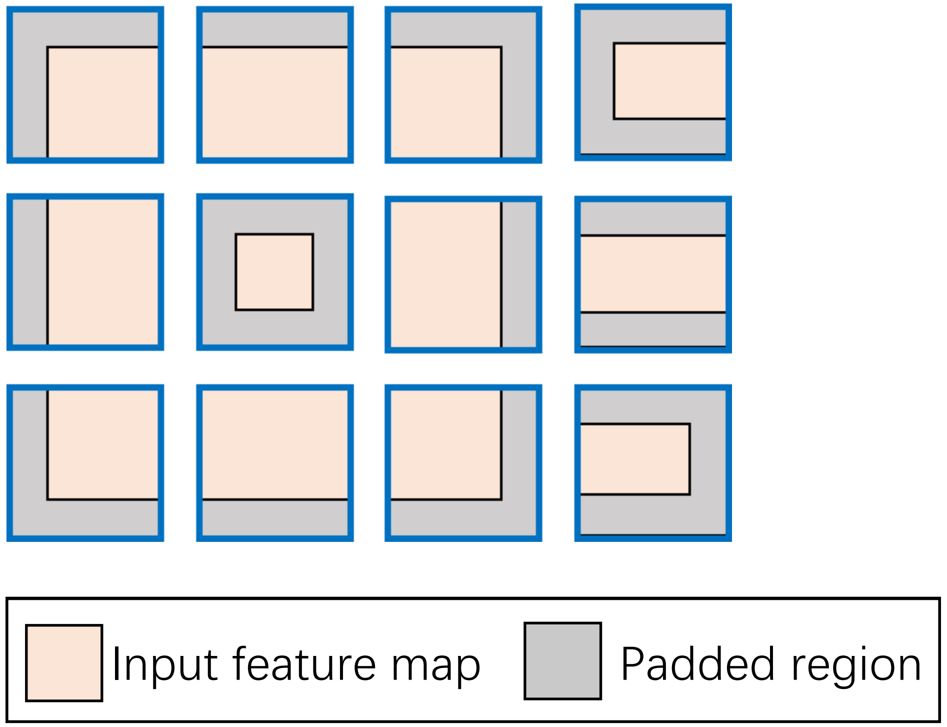
Hardware Architecture

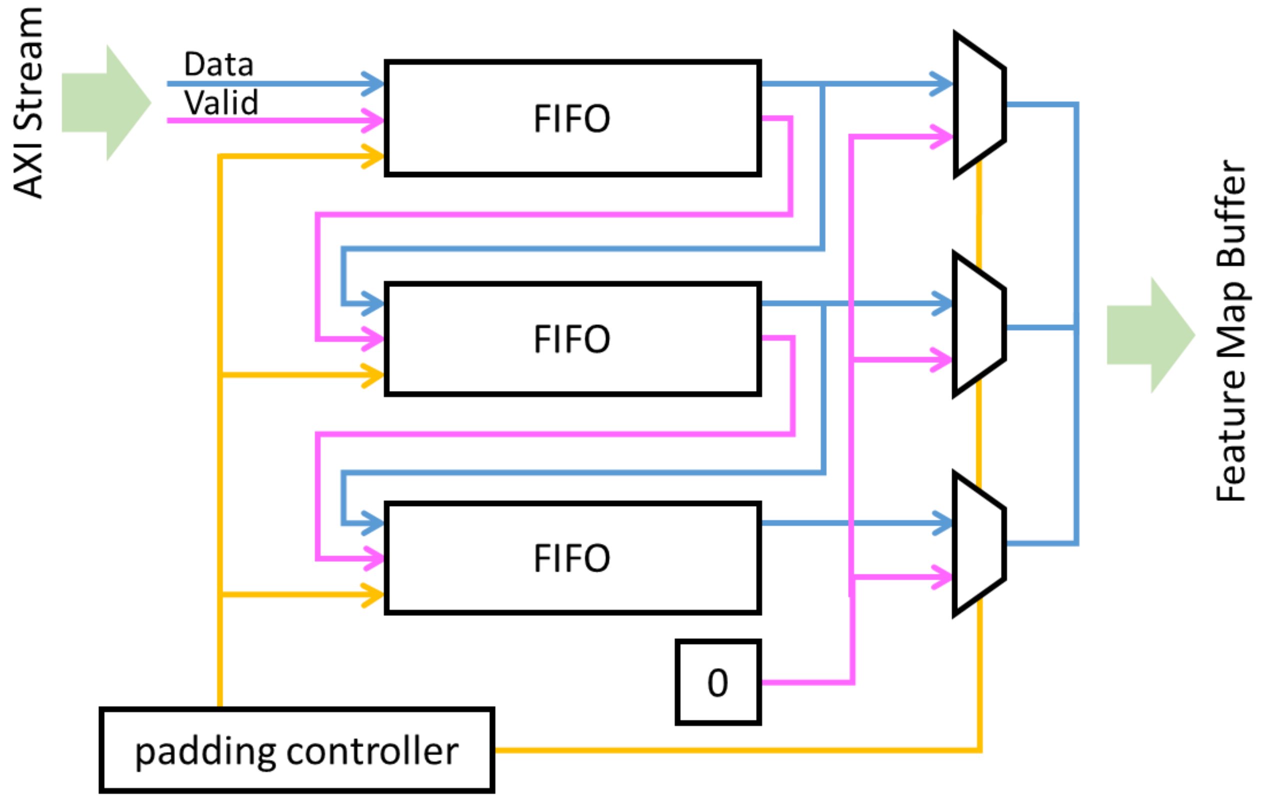
- Reshape input using Line Buffer
- Larger Input FeatureMap (IF) buffer needed



Hardware Architecture

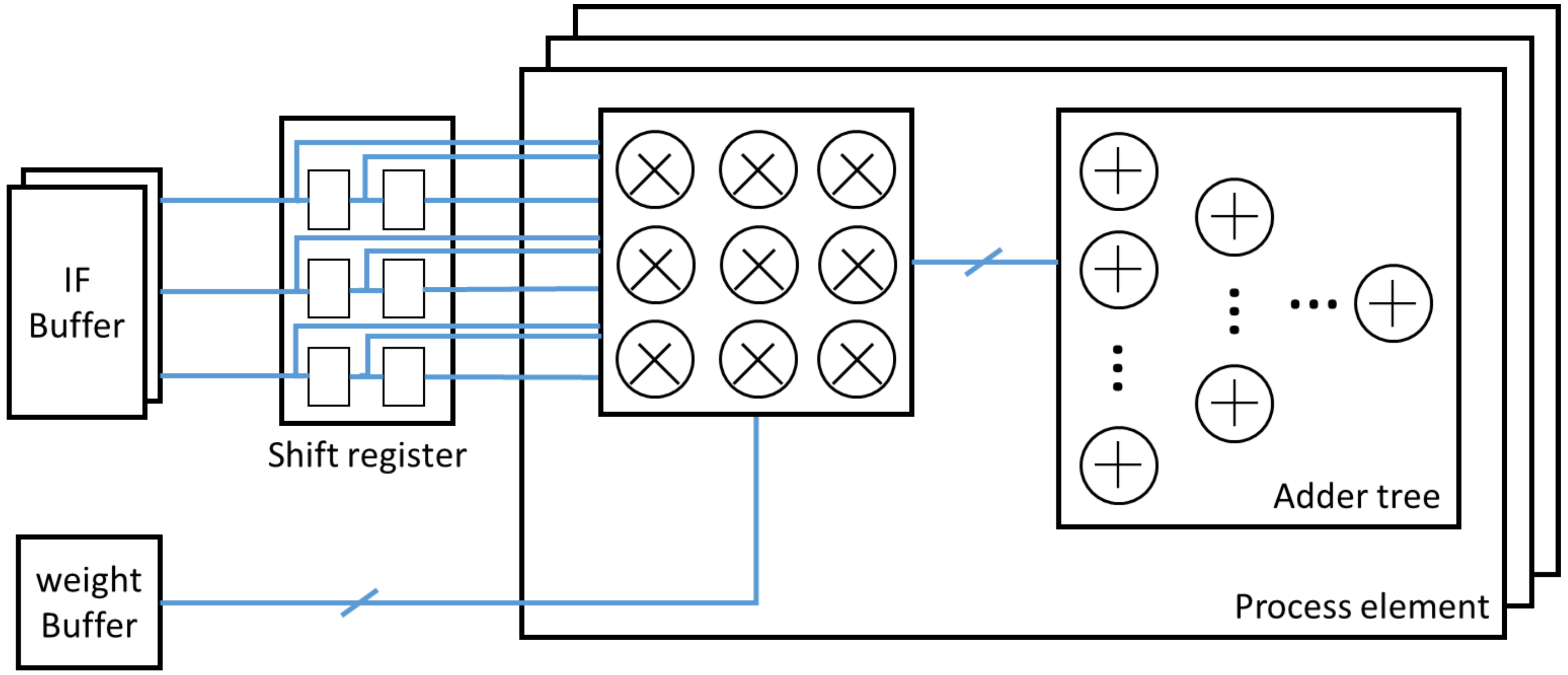
- Details of Line Buffer
- Capable of different paddings





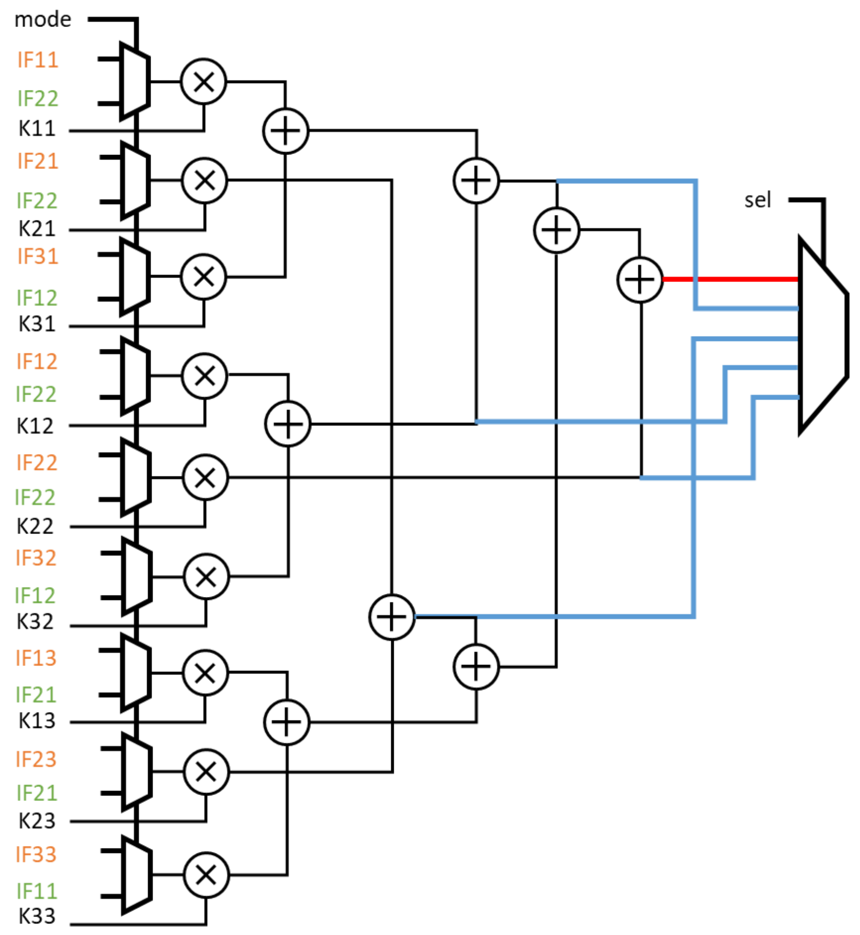
Hardware Architecture

- Process engine array



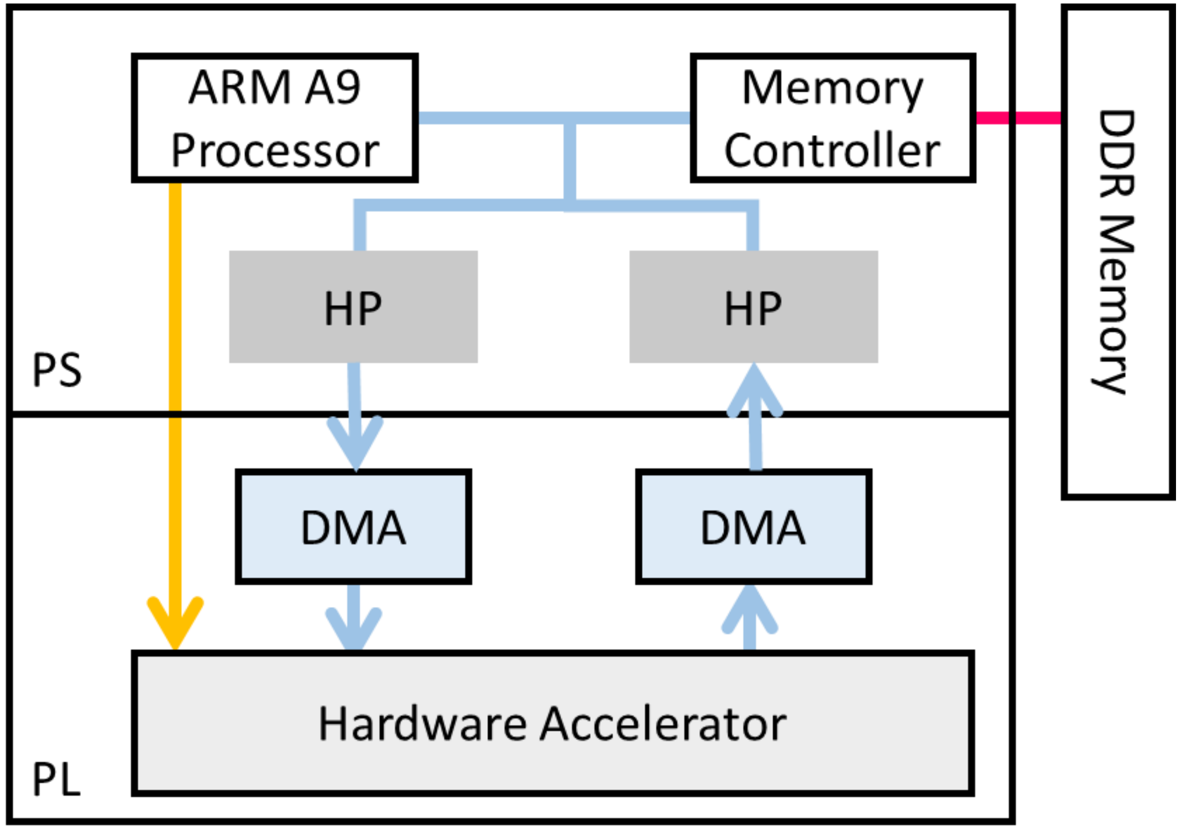
Hardware Architecture

- Process engine array



Overview of System

- Xilinx ZC706
- Camera is connected to PS side via USB interface
- High Performance (HP) for data transmitting



Performance and Results

Virtual, October 10-21

IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS

Performance comparison to others

	[18]	[19]	[10]	[11]	Ours
FPGA	ZC7Z045	ZC7Z045	ZC7Z045	ZC7Z020	ZC7Z045
Clock (MHz)	150	100	200	100	220
Operation	CONV	CONV	CONV+DECONV	DECONV	CONV+DECONV
Performance (GOPS)	187	229	125 (CONV) 29 (DECONV)	2.6	151.5 (CONV) 94.3 (DECONV)
Resource Efficiency (GOPS/DSP)	0.207	0.254	0.14 (CONV) 0.033 (DECONV)	0.012	0.168 (CONV) 0.104 (DECONV)

[18] J. Qiu, J. Wang, S. Yao et al, "Going deeper with embedded fpga platform for convolutional neural network," In Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays(FPGA), pp. 26-35, 2016.
[19] Q. Xiao, Y. Liang et al, "Exploring heterogeneous algorithms for accelerating deep convolutional neural networks on FPGAs," In 2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC), pp. 1-6, 2017.
[10] S. Liu, H. Fan, X. Niu, H. Ng, Y. Chu, and W. Luk, "Optimizing CNN based Segmentation with Deeply Customized Convolutional and Deconvolutional Architectures on FPGA," ACM Transactions on Reconfigurable Technology and Systems (TRETS), vol. 11, no. 3, 2018.
[11] X. Zhang, S. Das, O. Neopane, K. Kreutz-Delgado, "A Design Methodology for Efficient Implementation of Deconvolutional Neural Networks on an FPGA," arXiv preprint arXiv:1705.02583, 2017.

Thank you!