Accelerating Pooling through Im2col and Col2im Instructions in the DaVinci Architecture

Caio S. Rohwedder
Unicamp - Brazil

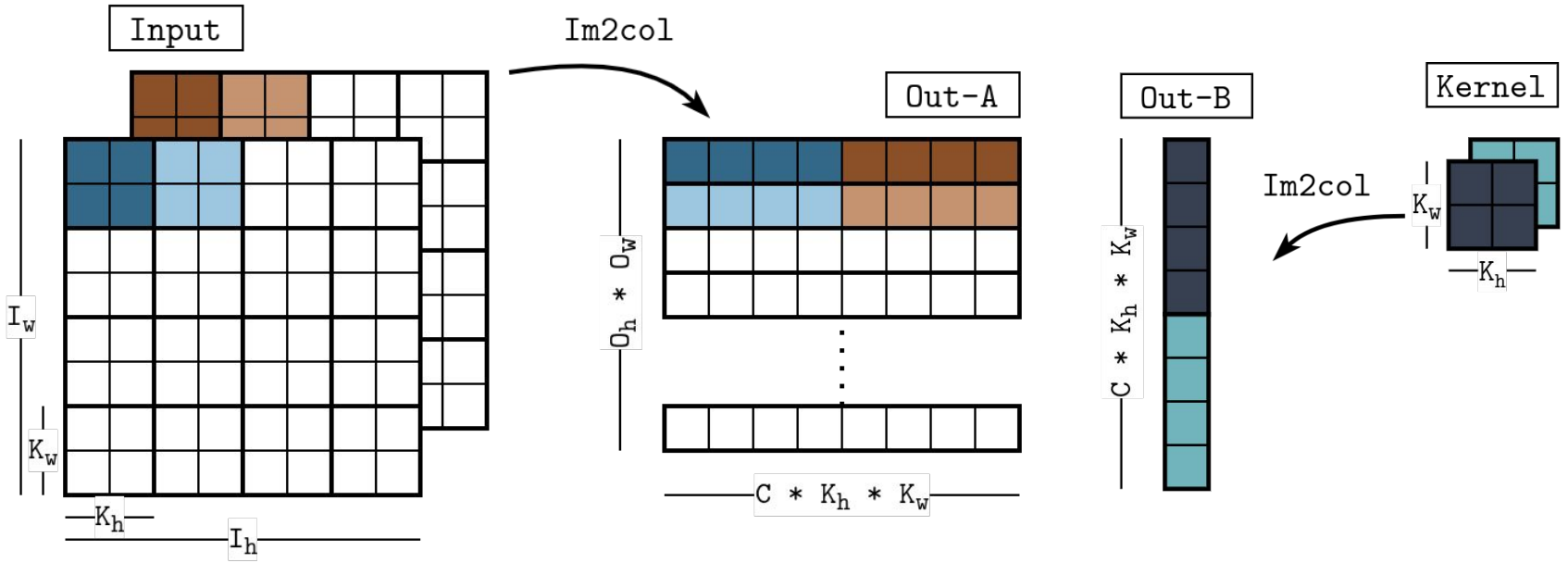
José N. Amaral
University of Alberta

Guido Araújo
Unicamp - Brazil

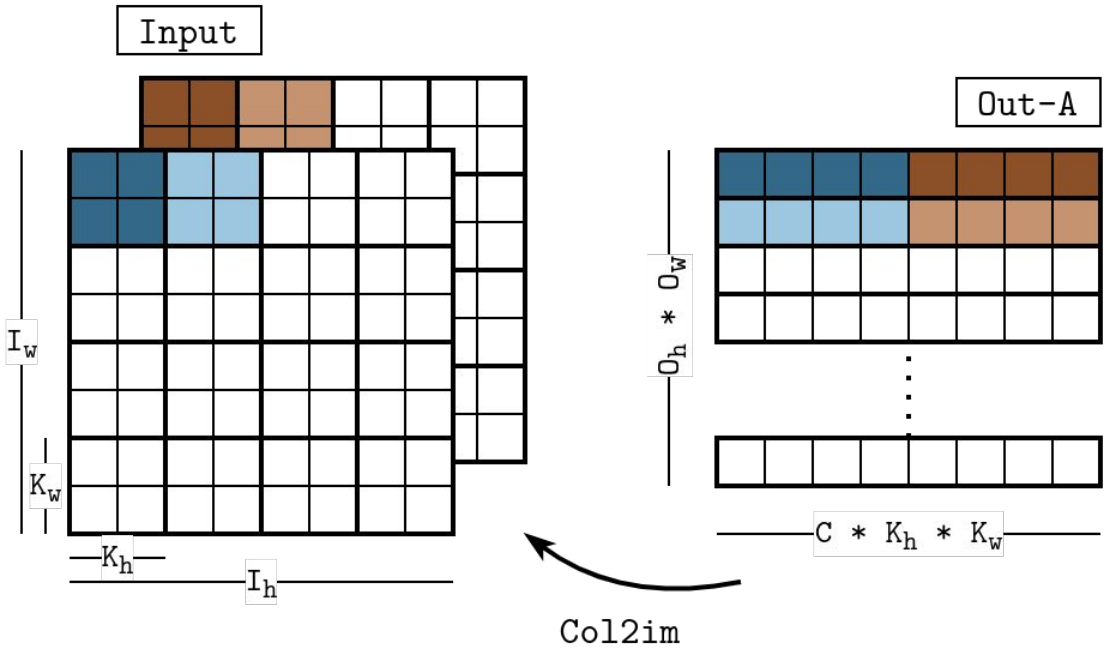
Amy Wang
Toronto Heterogeneous Compiler Lab

Giancarlo Colmenares
Toronto Heterogeneous Compiler Lab

Im2col + GEMM


Input
Im2col
Out-A
Out-B
Kernel
Im2col
I_w
K_w
K_h
I_h
$O_h * O_w$
$C * K_h * K_w$
$C * K_h * K_w$
K_w
K_h

Col2im: Backward Operator

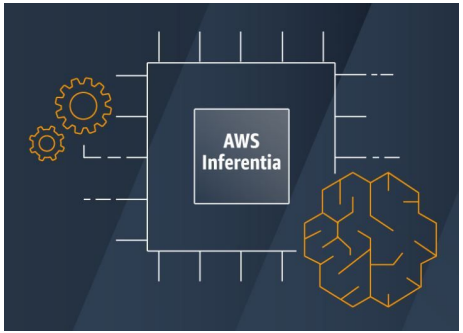


DL Accelerators

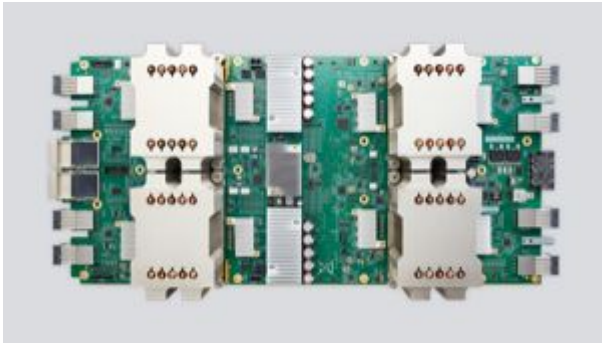
Huawei's Ascend 910[2]

Apple's A12[3]

Amazon's Inferentia[4]



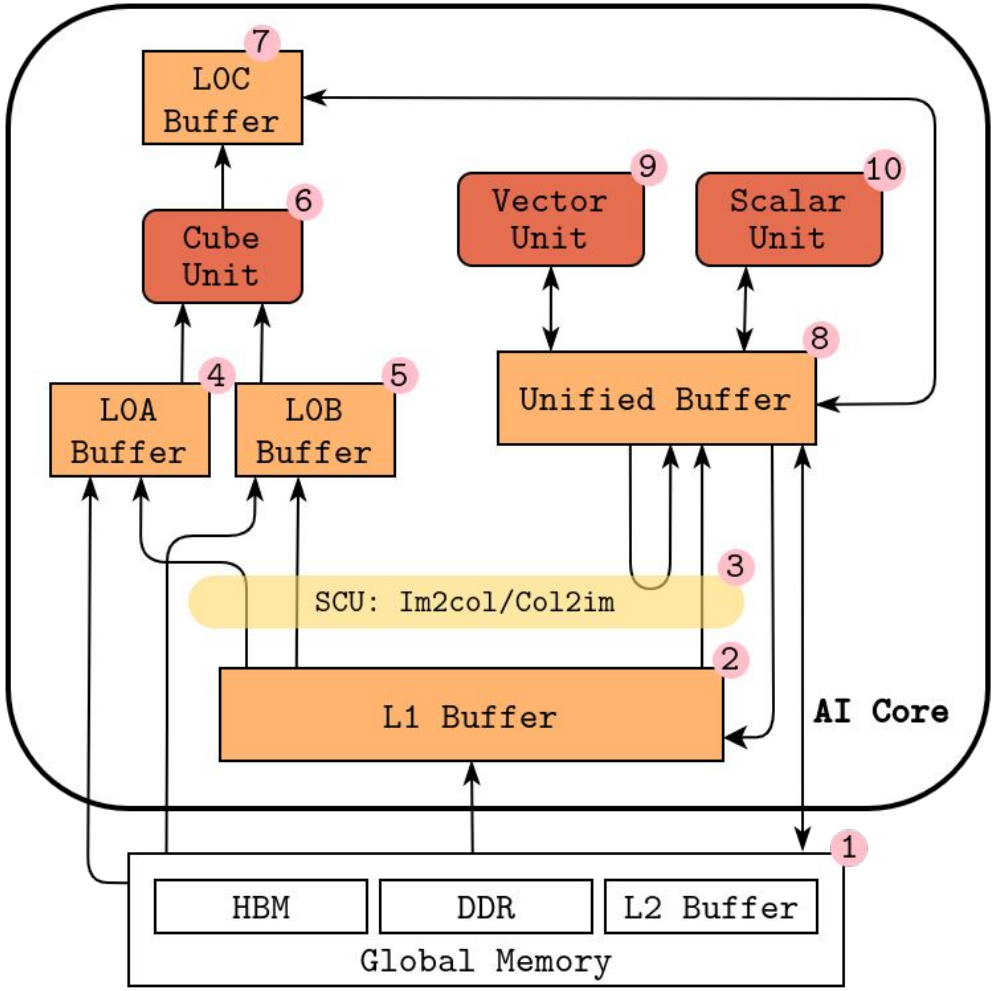
Google's TPU v2[1]



[1]https://cloud.google.com/tpu [2] https://e.huawei.com/en/products/cloud-computing-dc/atlas/ascend-910
[3]https://www.techinsights.com/blog/apple-iphone-xs-max-teardown [4]https://aws.amazon.com/machine-learning/inferentia/

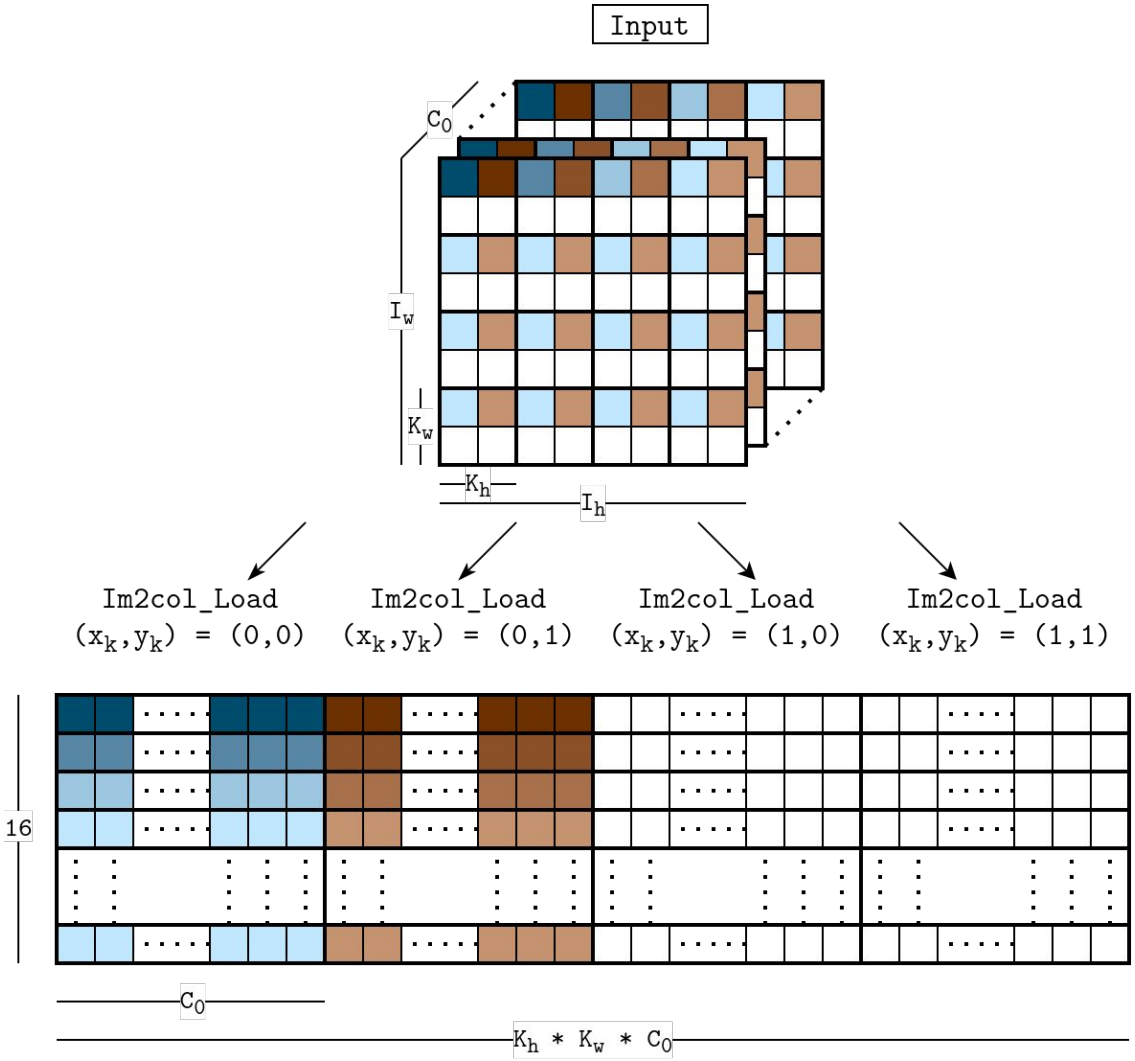
DaVinci - Ai Core

$$NCHW \rightarrow NC_1HWC_0$$



Im2col Instruction

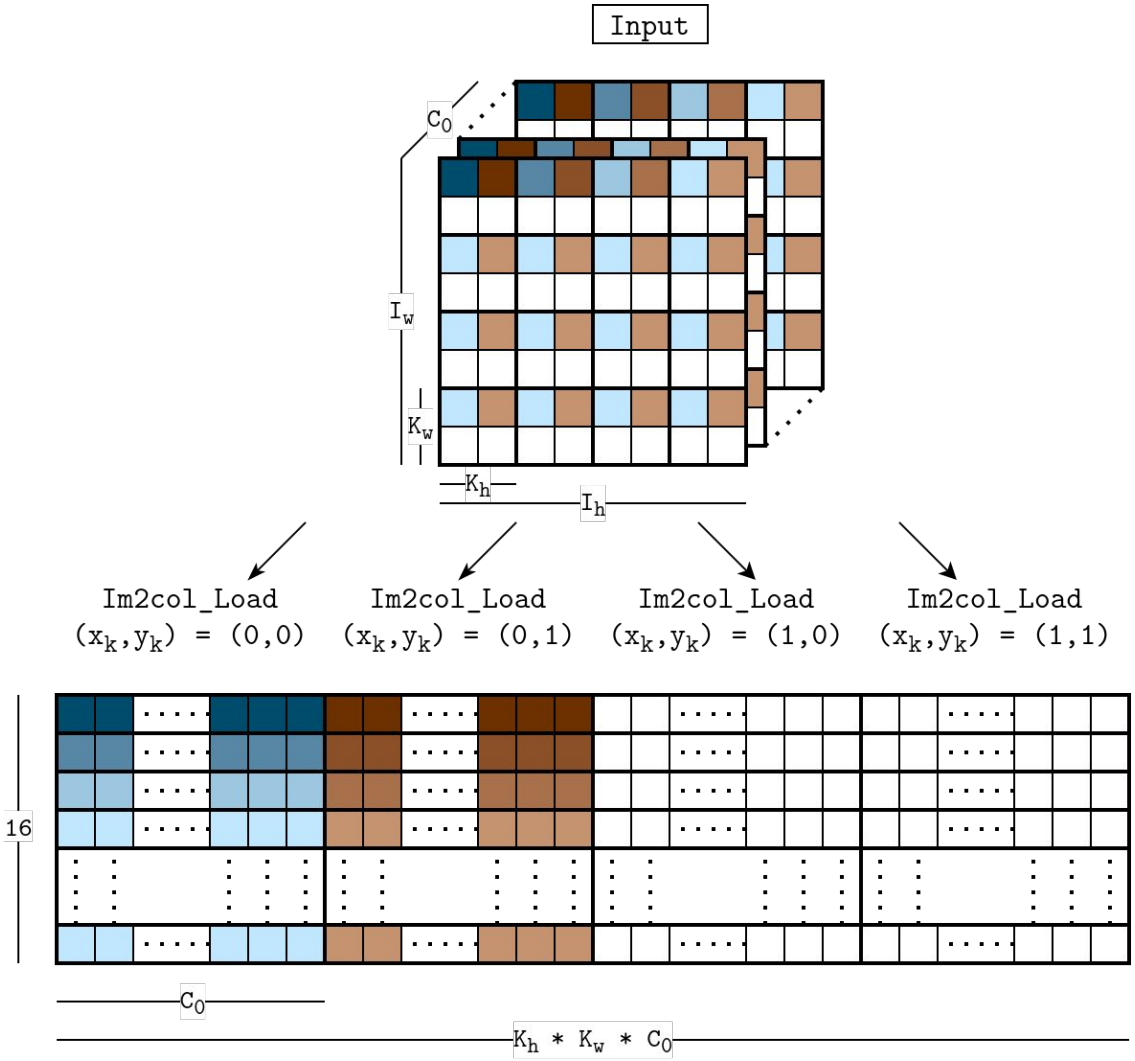
(x, y) = (0, 0)



Im2col Instruction

(x, y) = (0, 0)

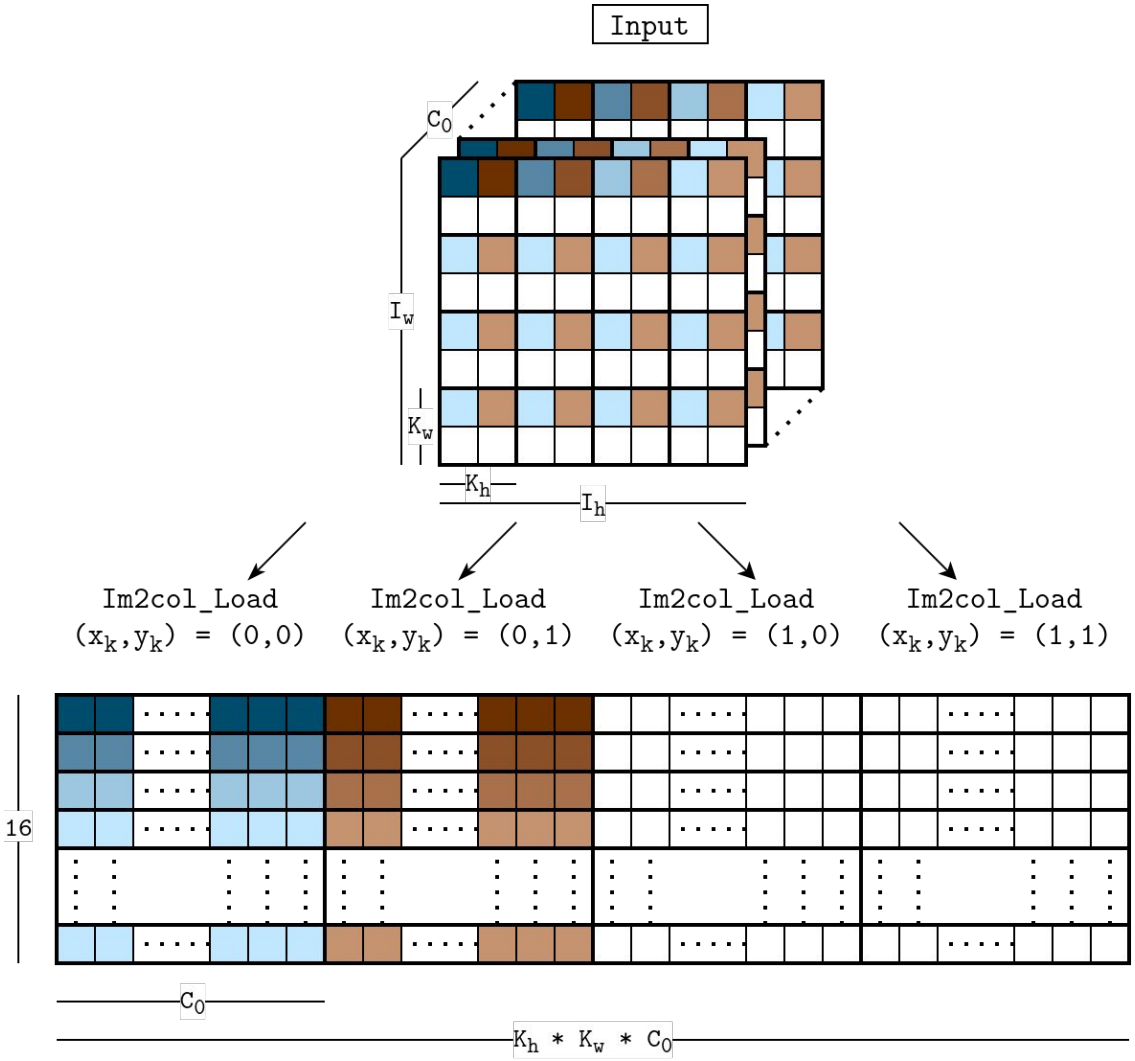
1. Select the next 16 patches from (x, y)



Im2col Instruction

(x, y) = (0, 0)

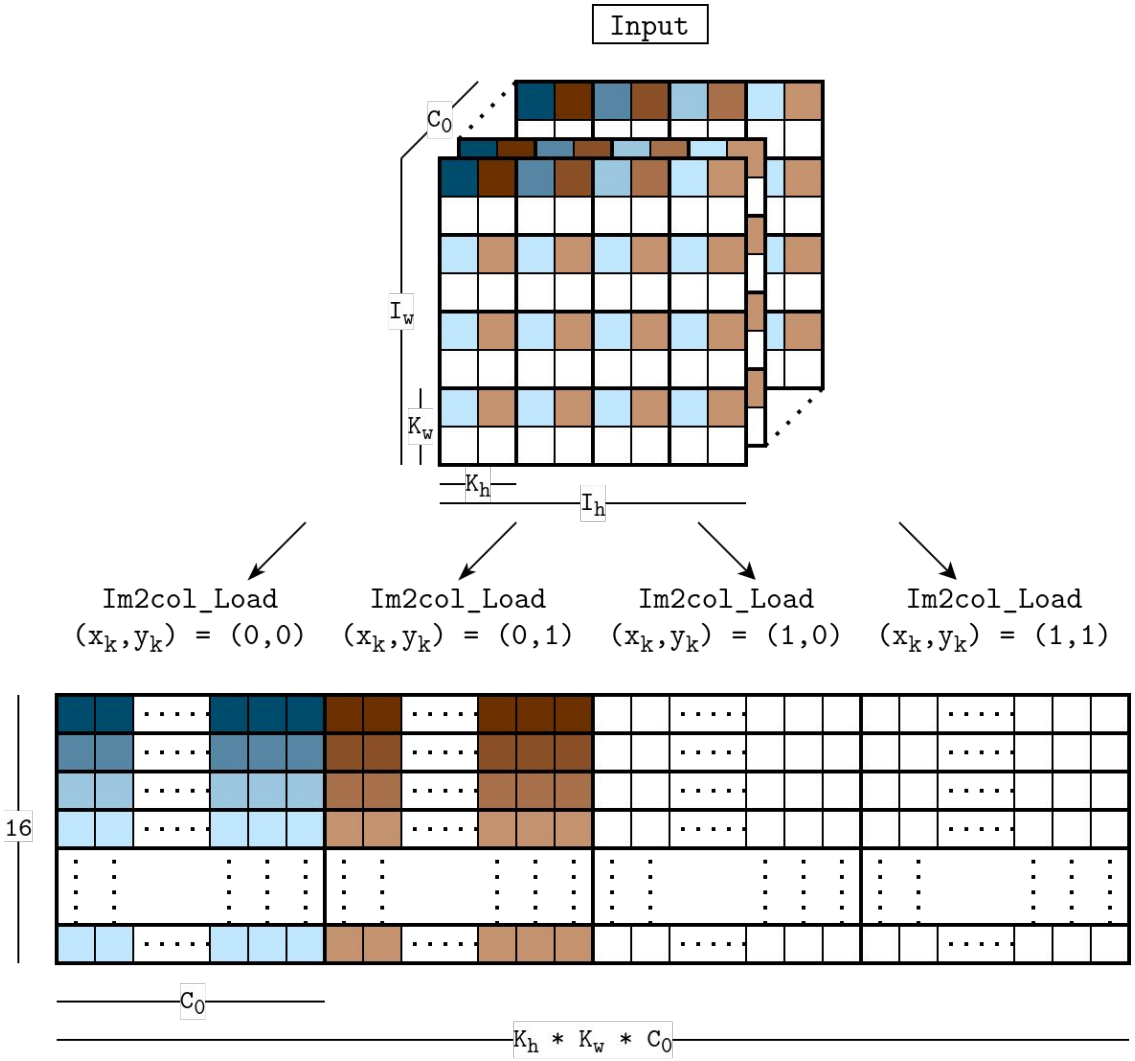
1. Select the next 16 patches from (x, y)
2. Select position (x_k, y_k) relative to the patches



Im2col Instruction

(x, y) = (0, 0)

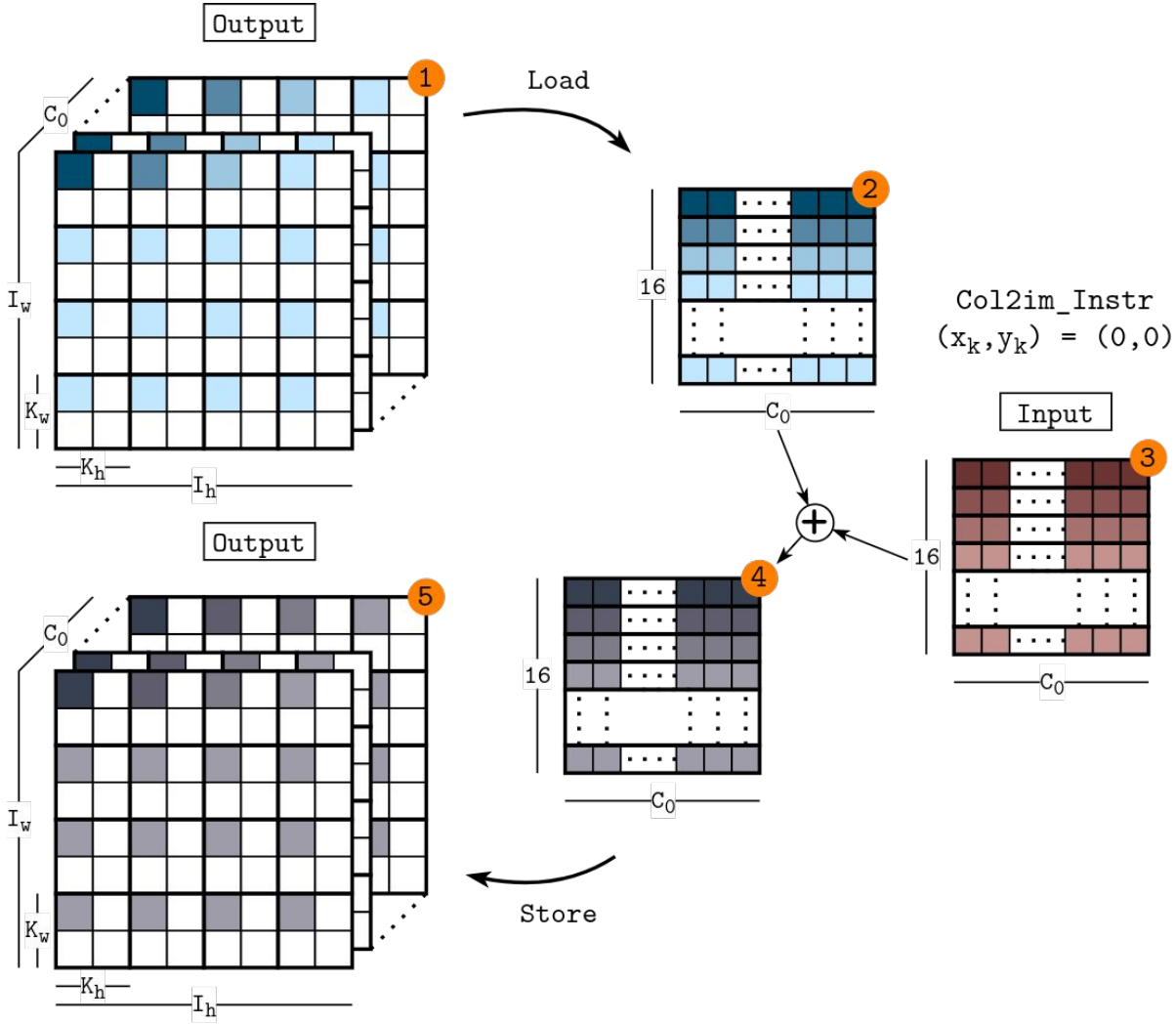
1. Select the next 16 patches from (x, y)
2. Select position (x_k, y_k) relative to the patches
3. Load all C_0 channels from the selected positions



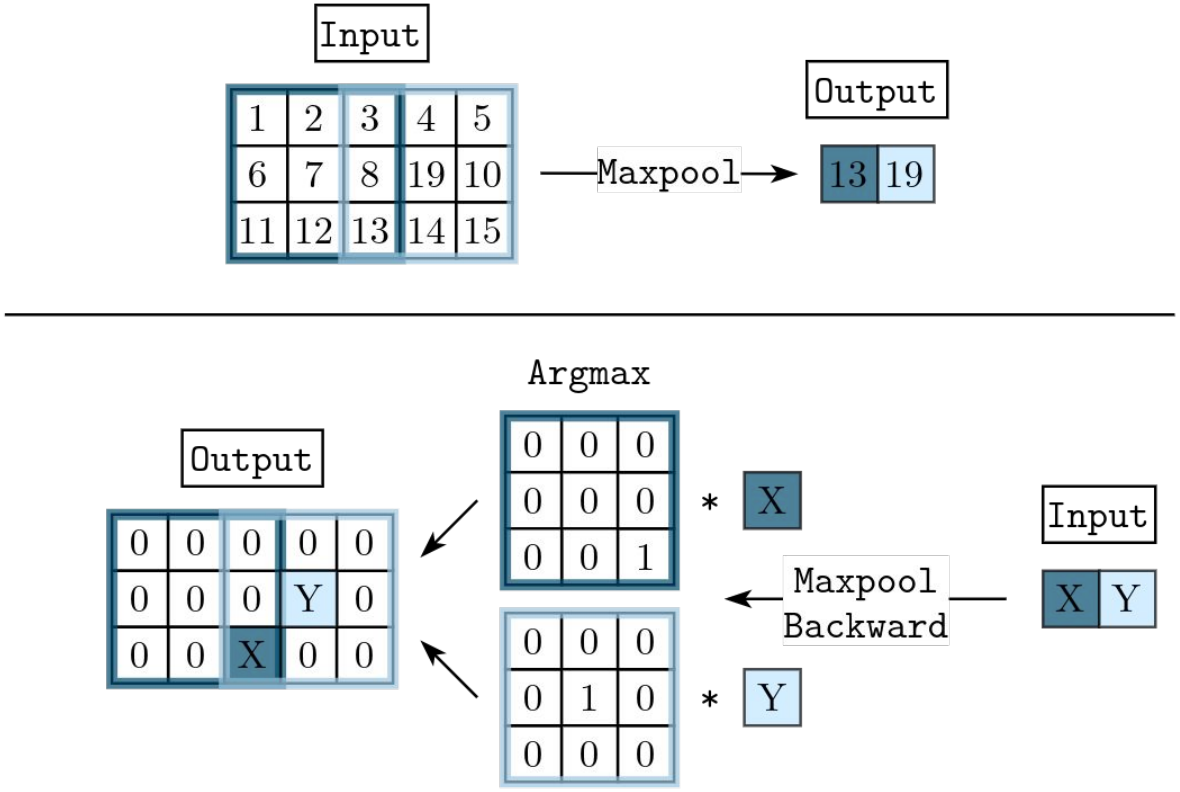
Col2im Instruction

Initialize output
with zeros

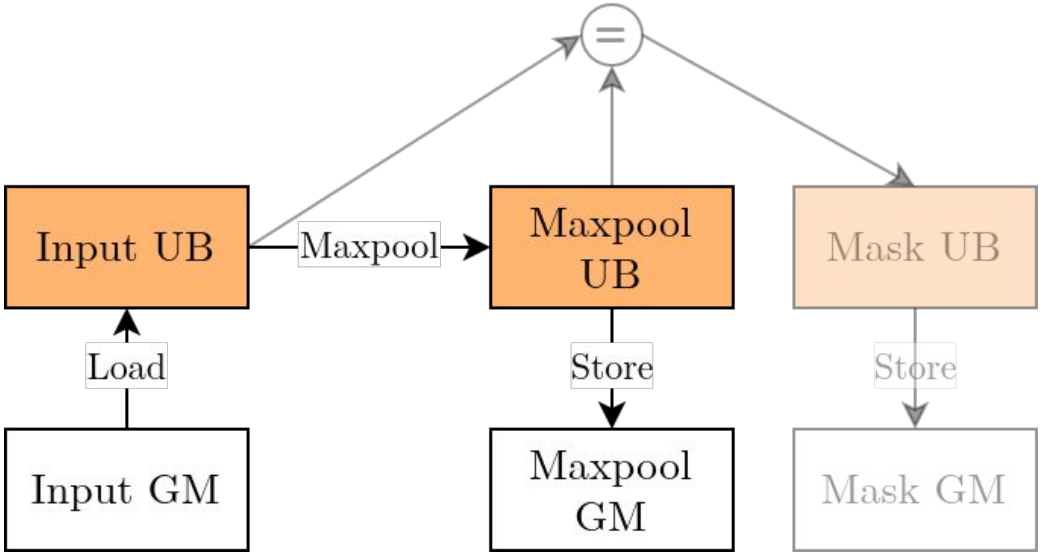
(x, y) = (0, 0)

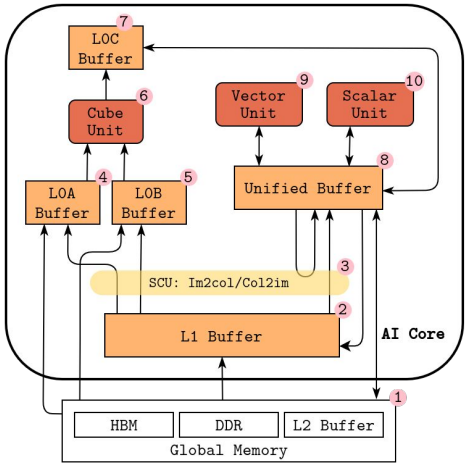


Pooling Operators

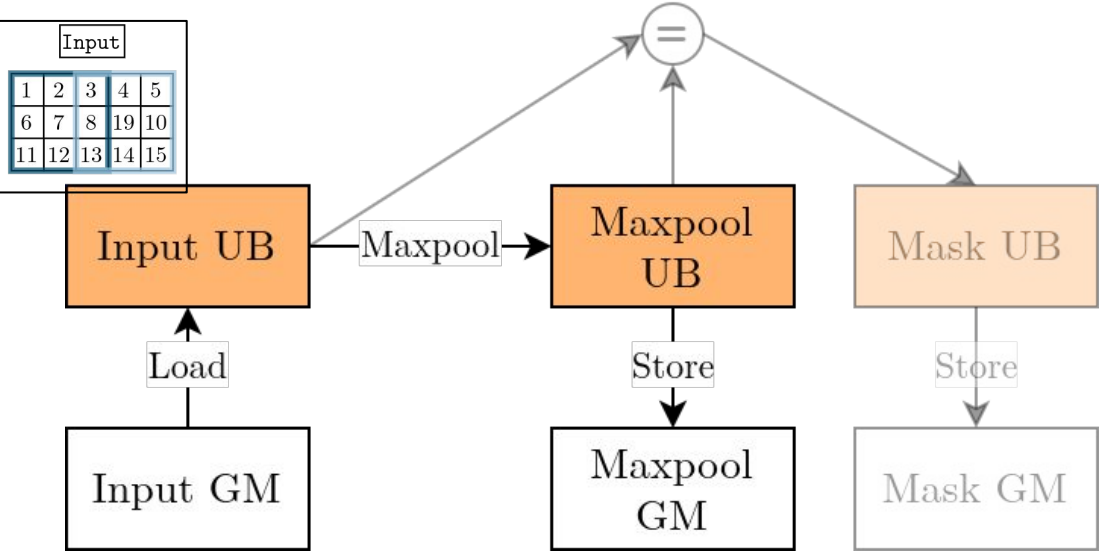


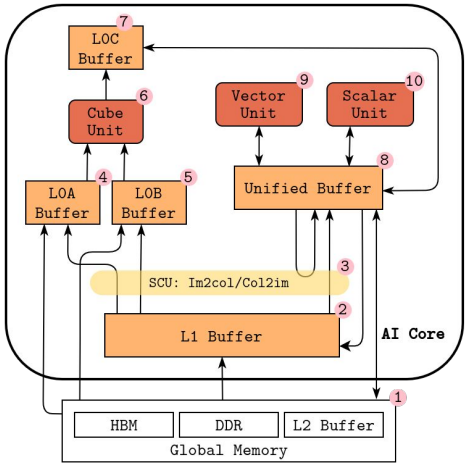
Pooling for DaVinci



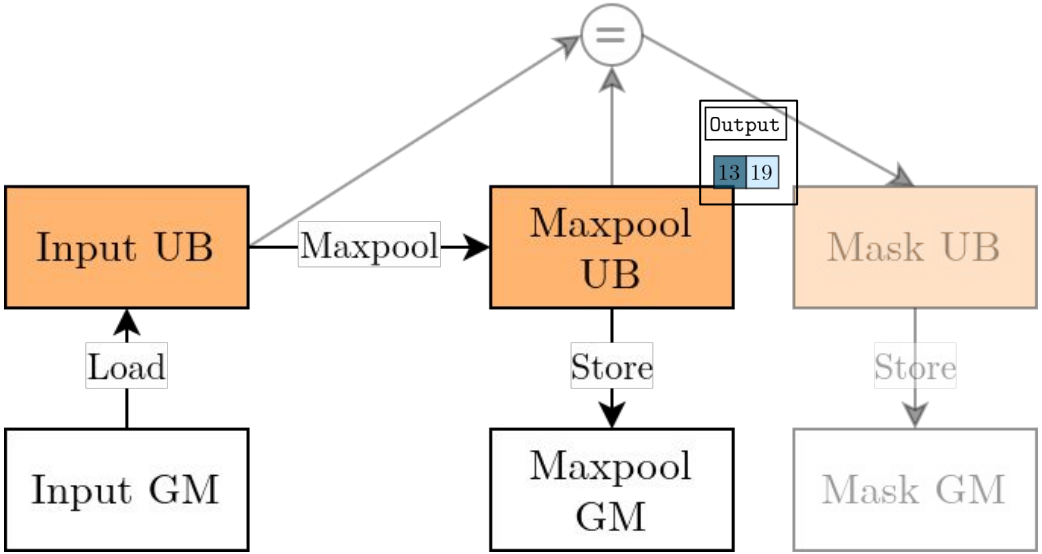


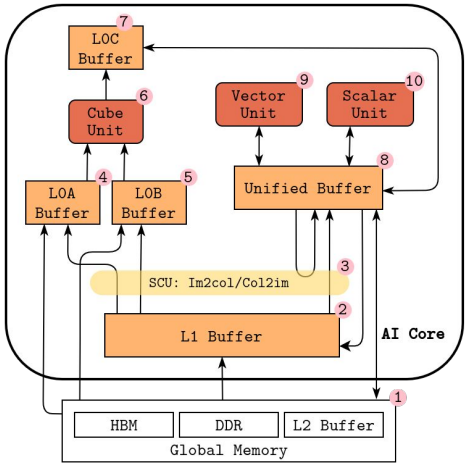
Pooling for DaVinci



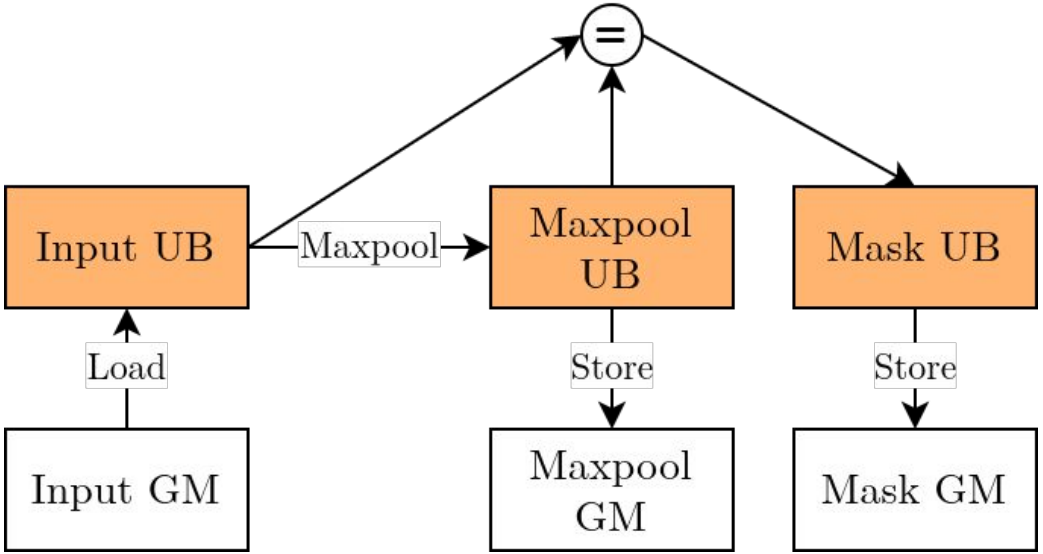


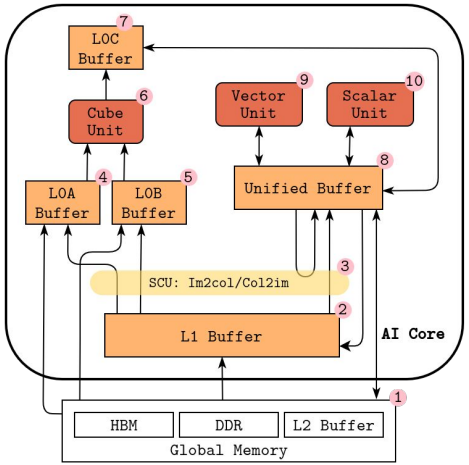
Pooling for DaVinci



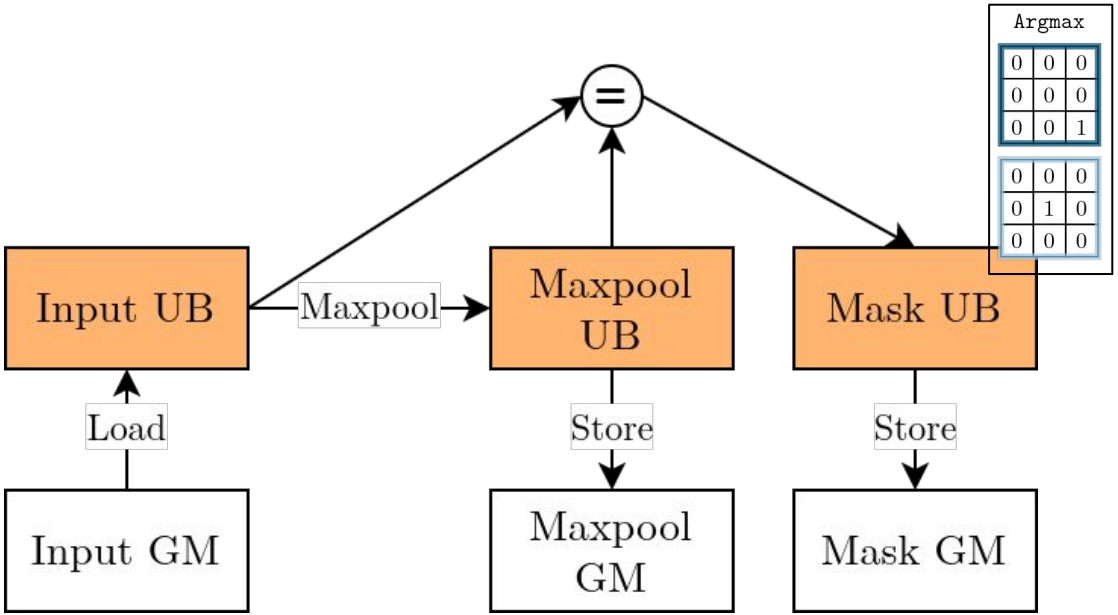


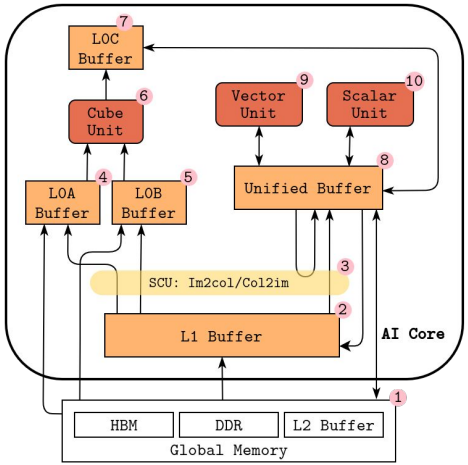
Pooling for DaVinci



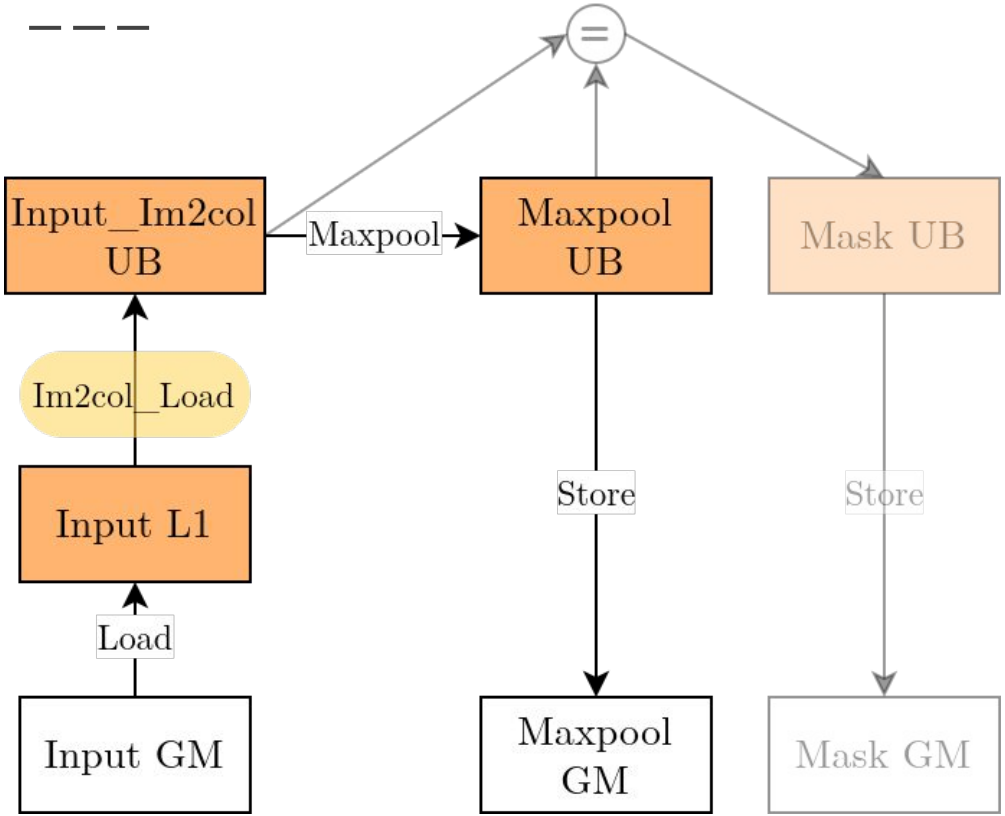


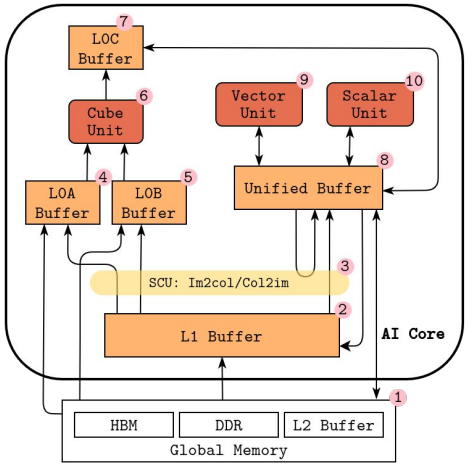
Pooling for DaVinci



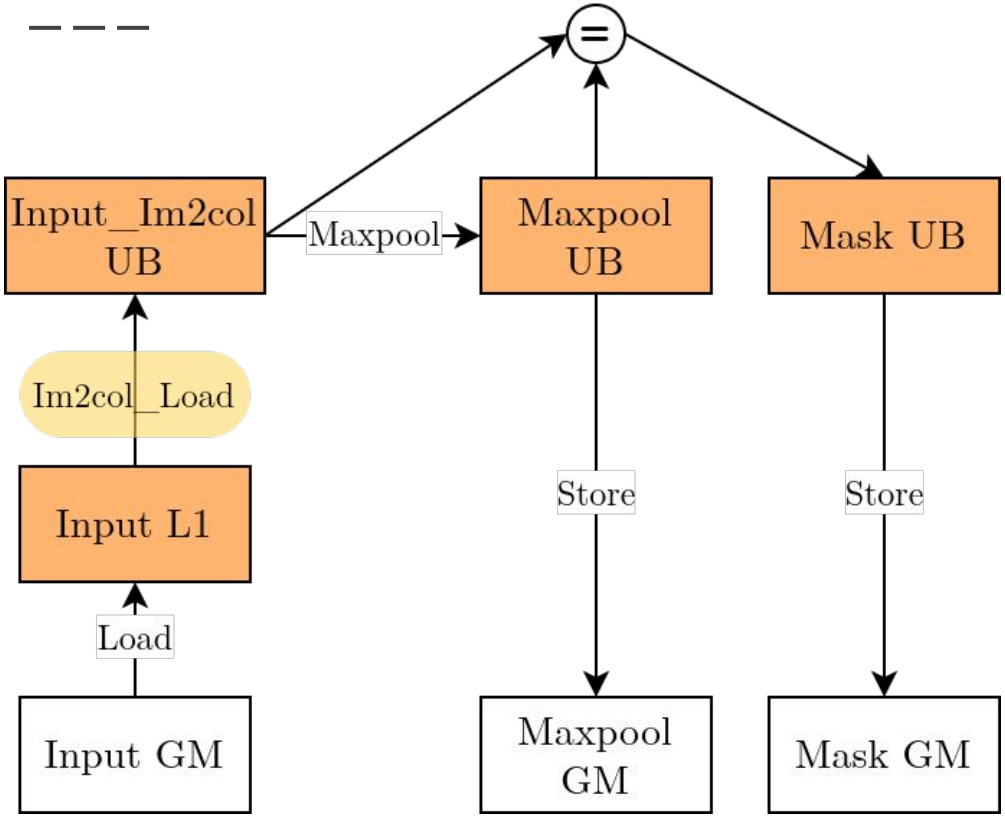


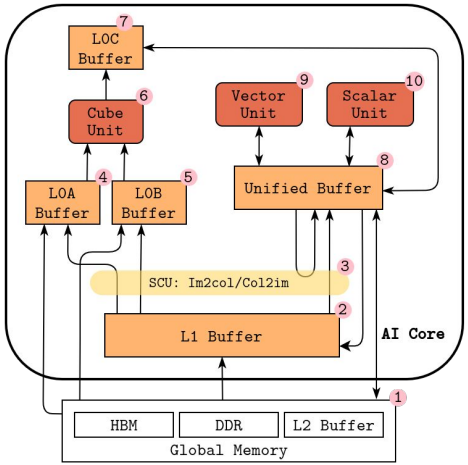
Im2col Based Pooling



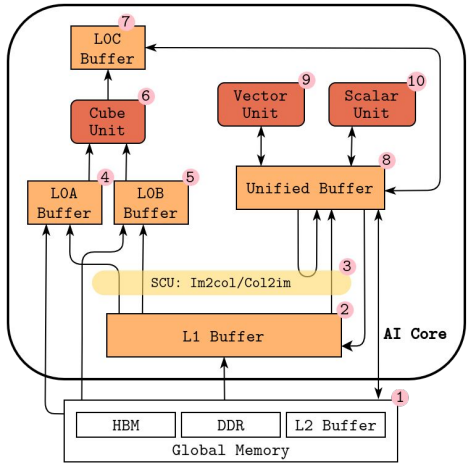


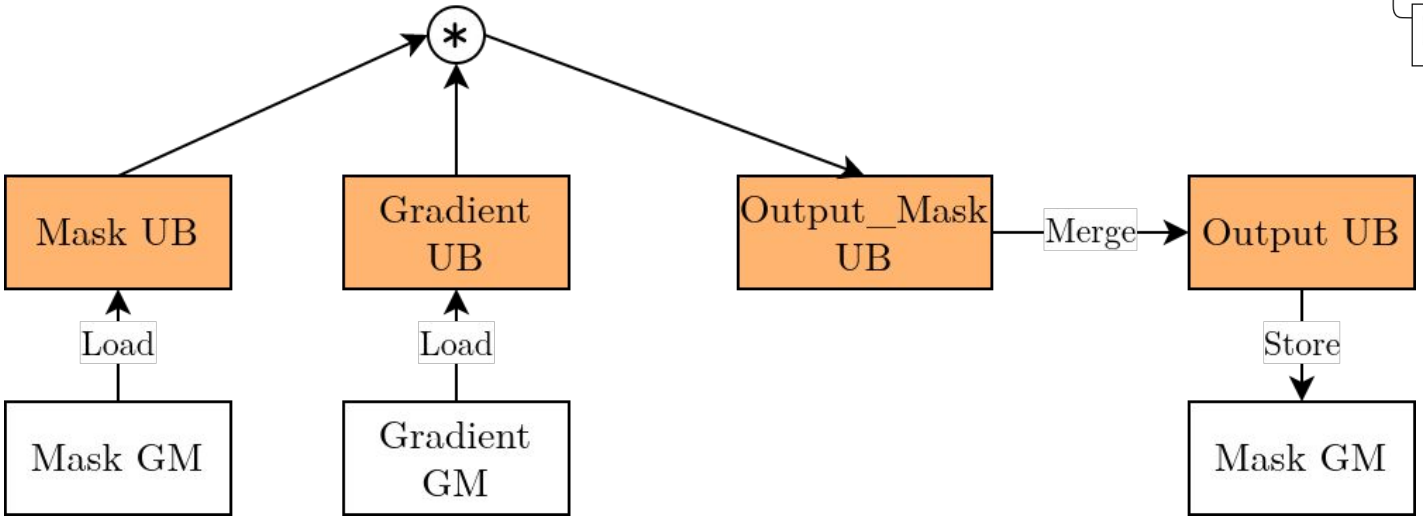
Im2col Based Pooling



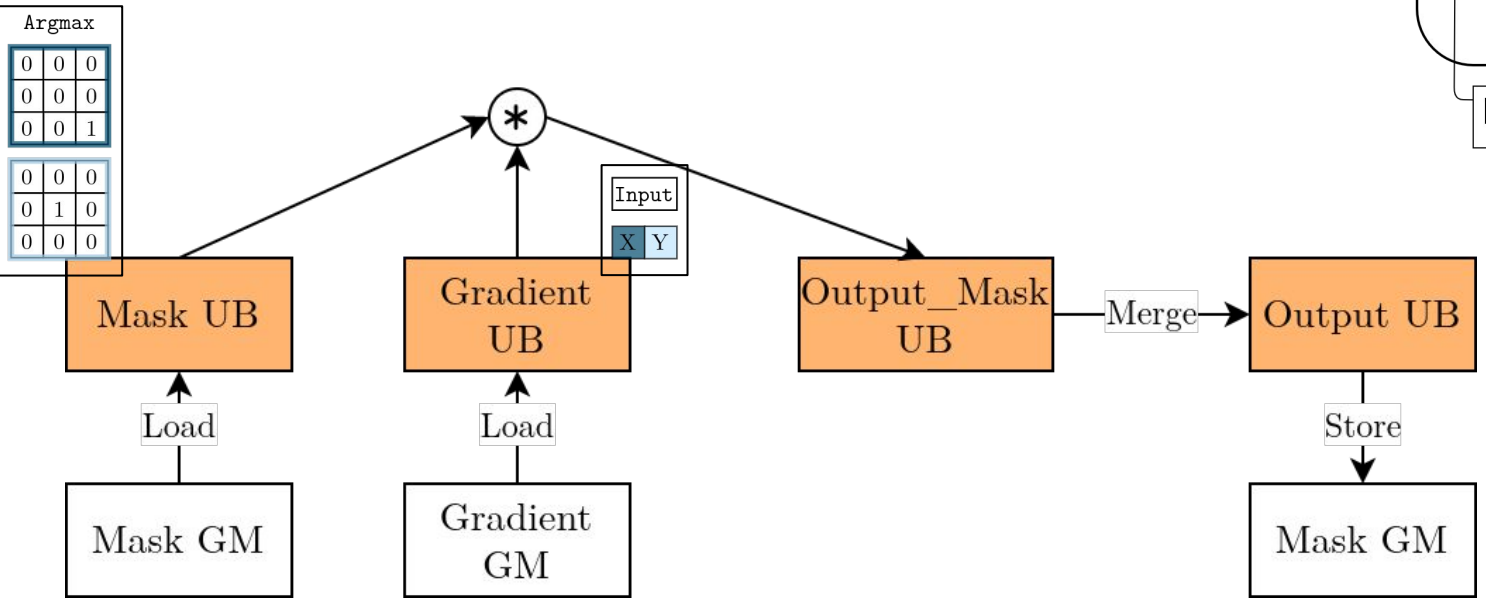


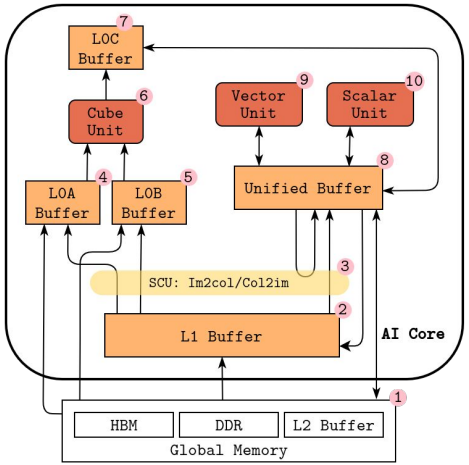
Backward Pooling for DaVinci



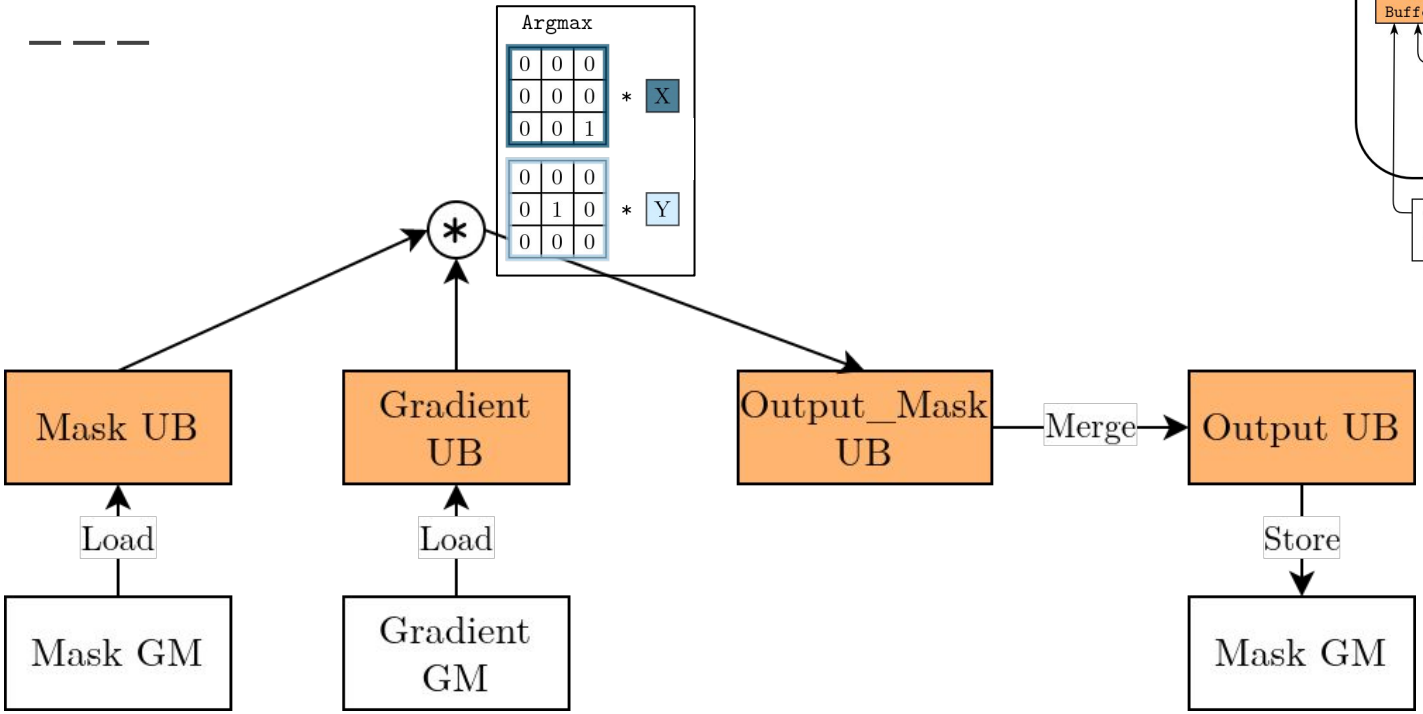


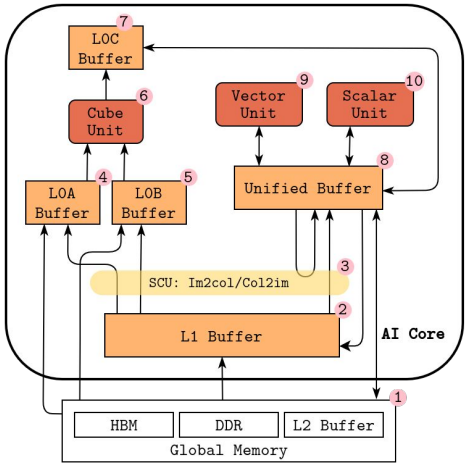
Backward Pooling for DaVinci



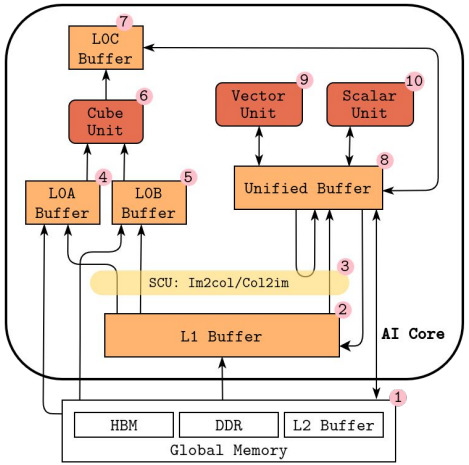


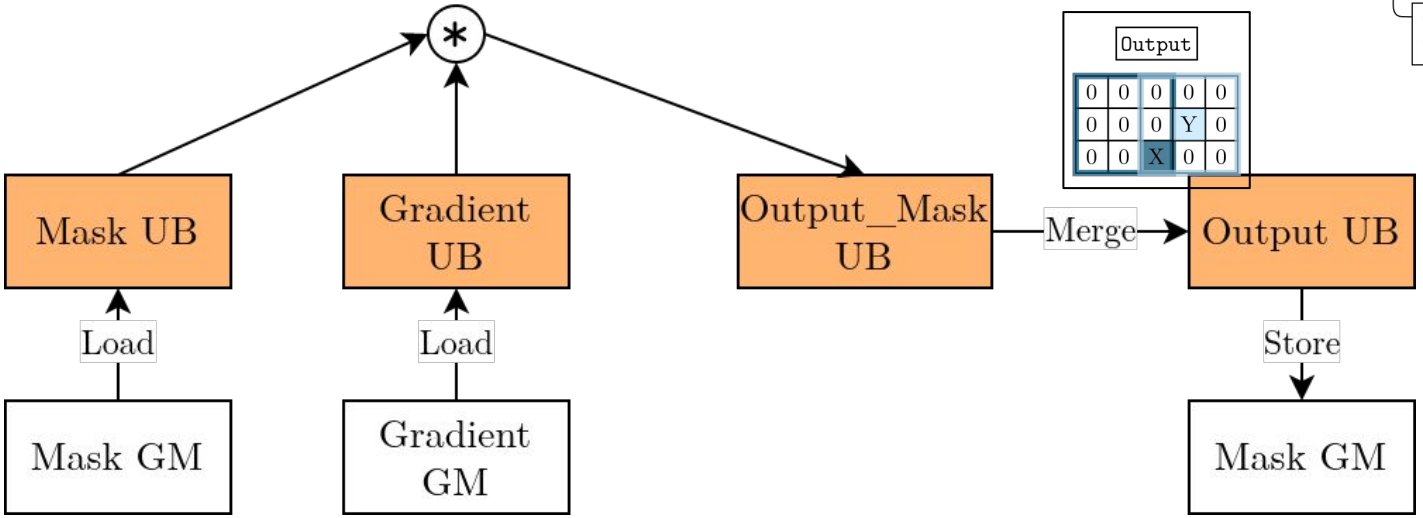
Backward Pooling for DaVinci



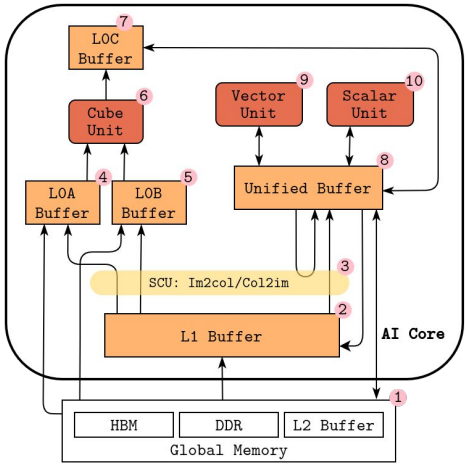


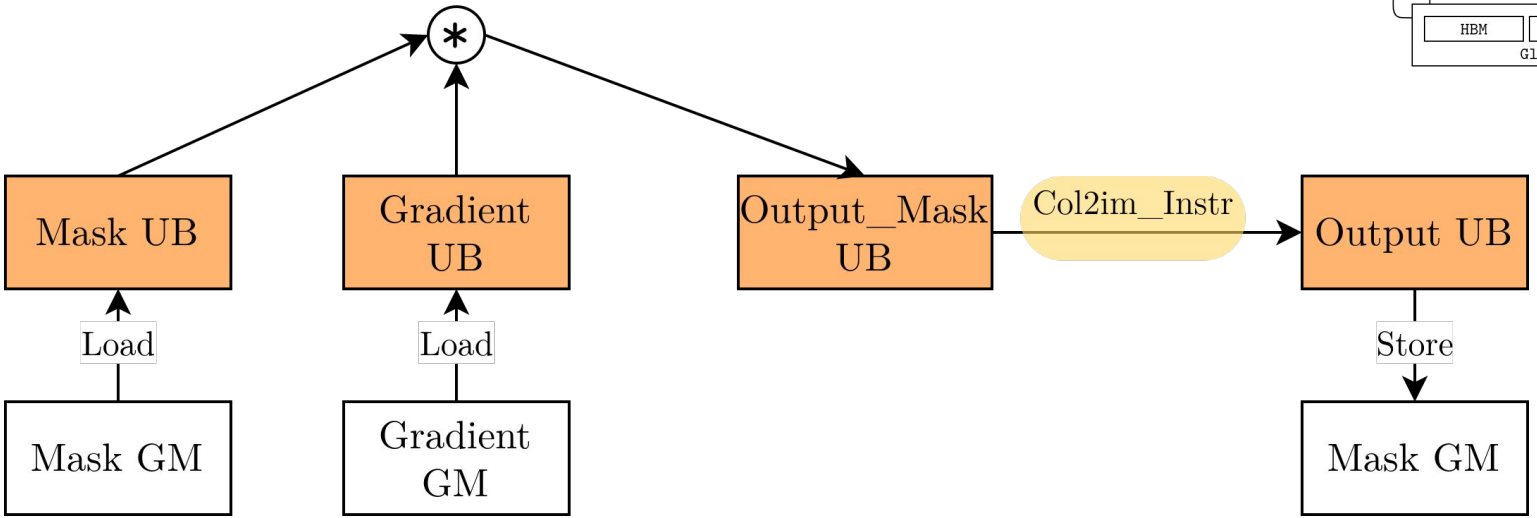
Backward Pooling for DaVinci





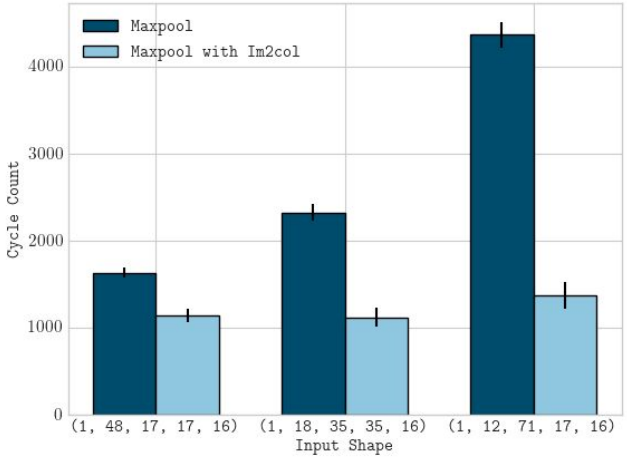
Col2im Based Backward Pooling

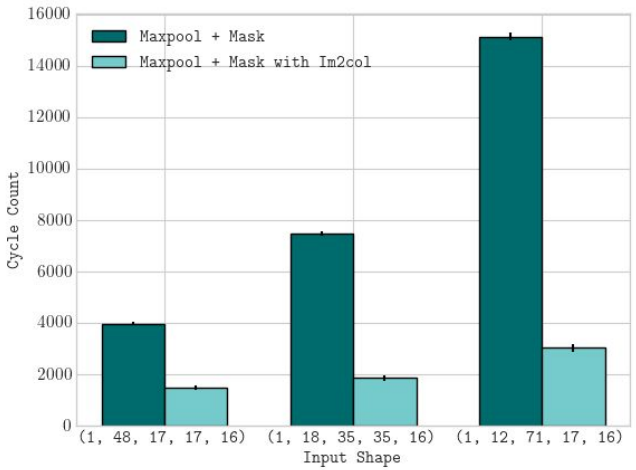


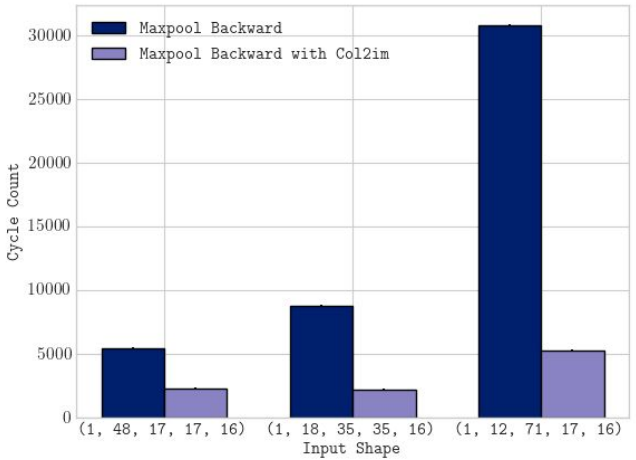


Maxpool Comparison

- Stride = (2, 2)
- Kernel size = (3,3)
- TVM



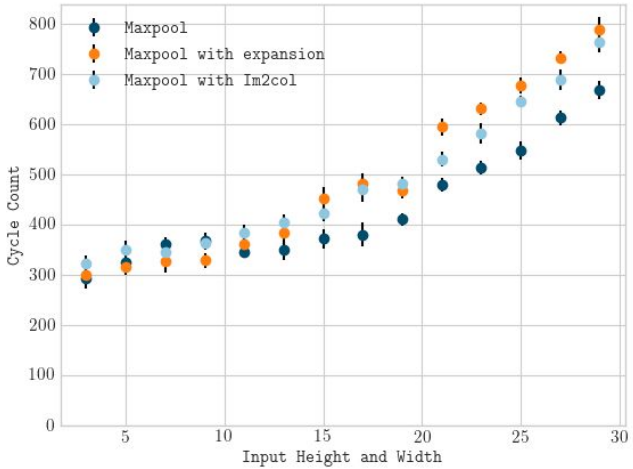




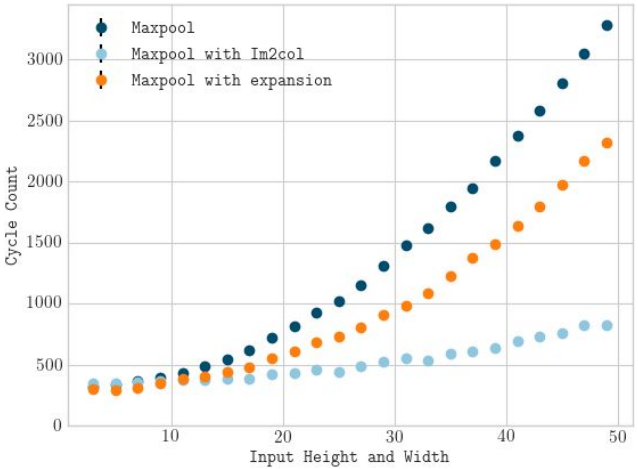
Maxpool Stride Comparison

- Single channel (C_1)
- Kernel size = (3,3)
- Up to tiling threshold

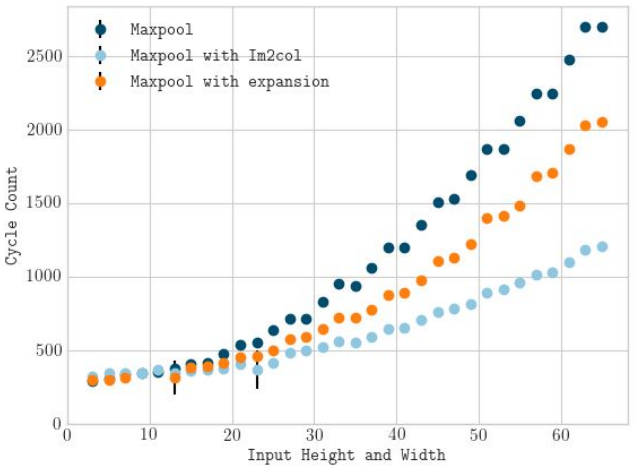
Stride = (1,1)



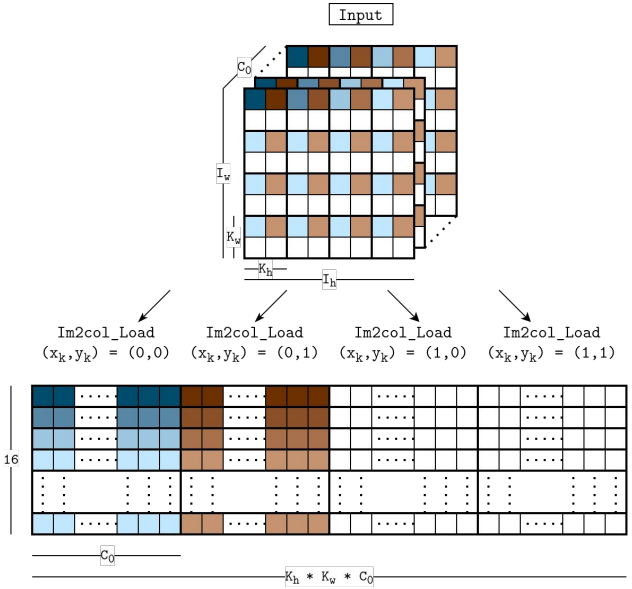
Stride = (2,2)

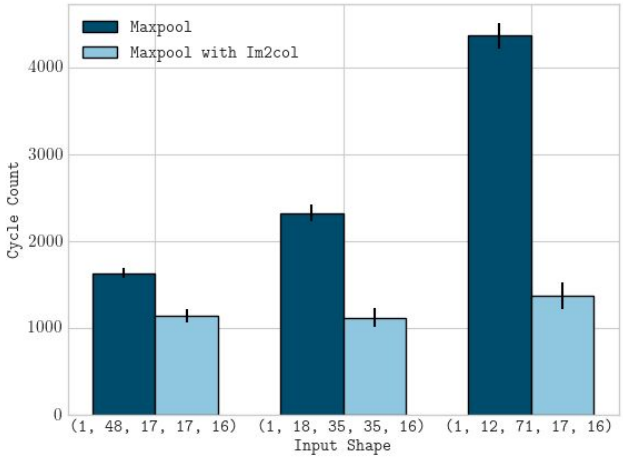


Stride = (3,3)



Conclusion


Input
C_0
I_w
K_w
K_h
I_h
Im2col_Load
$(x_k,y_k) = (0,0)$
Im2col_Load
$(x_k,y_k) = (0,1)$
Im2col_Load
$(x_k,y_k) = (1,0)$
Im2col_Load
$(x_k,y_k) = (1,1)$
16
C_0
$K_h * K_w * C_0$


Maxpool
Maxpool with Im2col
Cycle Count
4000
3000
2000
1000
0
(1, 48, 17, 17, 16)
(1, 18, 35, 35, 16)
(1, 12, 71, 17, 16)
Input Shape