ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

LAB06: Tiling on PULP

Davide Nadalini – d.nadalini@unibo.it
Lorenzo Lamberti – lorenzo.lamberti@unibo.it
Luka Macan – luka.macan@unibo.it
Francesco Conti – f.conti@unibo.it

Objective of the Class

Intro: Tiling

Tasks:

- 2D convolution in L1
- 2D convolution in L2
- Layer Tiling

Deadline:

Dec 6th 2024

Programming Language: C

Lab duration: 3h

Assignment:

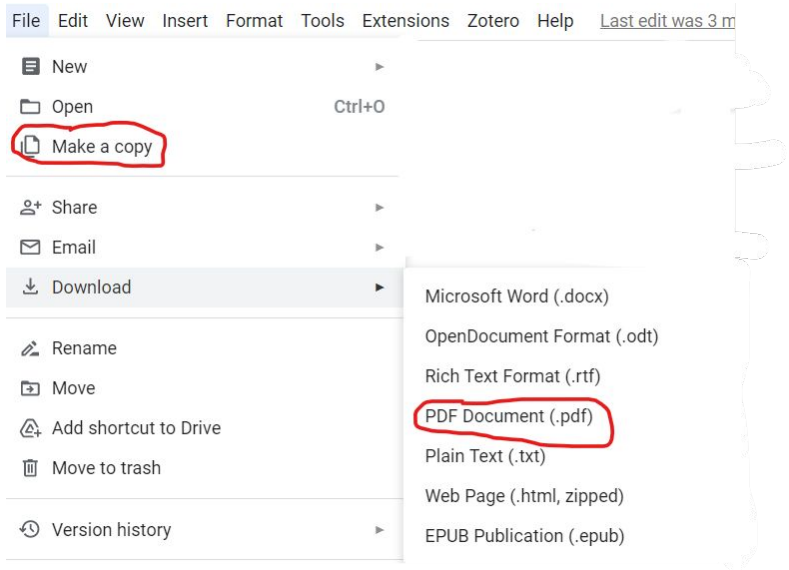
- Time for delivery: 2 weeks

The class is meant to be interactive: coding together and on your own!

How to deliver the Assignment

You will deliver ONLY the GDOC assignment, no code

- Copy the google doc to your drive, so that you can modify it. (File -> make a copy)
- Fill the tasks on this google doc.
- Export to pdf format.
- Rename the file to: LAB<number_of_the_lesson>_APAI_<your_name>.pdf
- Use Virtuale platform to load ONLY your .pdf file



Opening the VM and VSCode

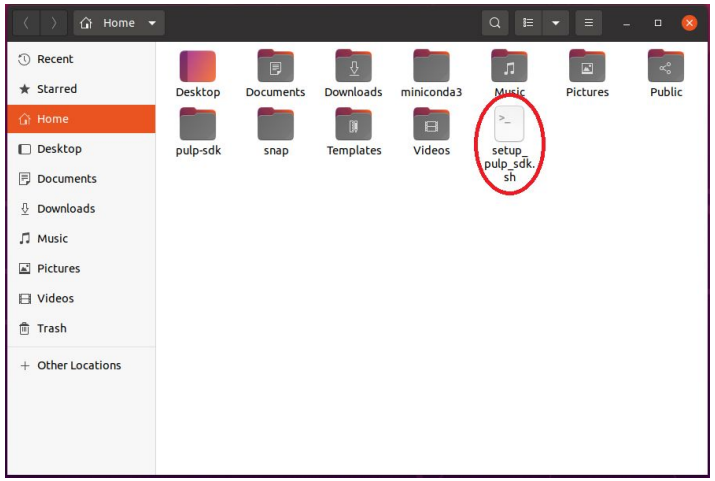
1. Open a terminal (right click – open a new terminal)

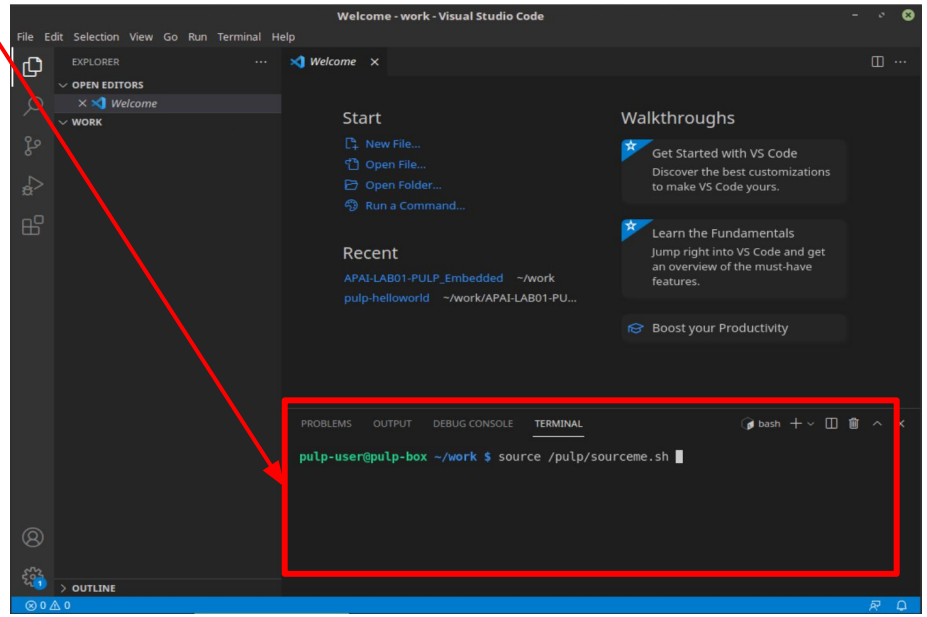
2. Open a text editor (For example “VSCode”):
   Now you can use the integrated terminal (open with CTRL+J) to run your applications!

```
$ code .
```

IMPORTANT: every time you open a new terminal to work on PULP, launch

```
$ source setup_pulp_sdk.sh
```





Getting Started

IMPORTANT: activate the pulp-sdk module file every time a new shell is open.

```
$ source setup_pulp_sdk.sh
```

For the lab of today, we need to:

```
$ pip install mako
```

HOW TO RUN THE CODE:

```
$ git clone https://github.com/EEESlab/APAI24-LAB06-PULP-Tiling-part1
$ cd APAI24-LAB06-PULP-Tiling-part1
$ python parameters_generate.py --channels=1 --spatial_dimension=1
$ make clean all run
```

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

INTRO

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

TASK1: fit in L1

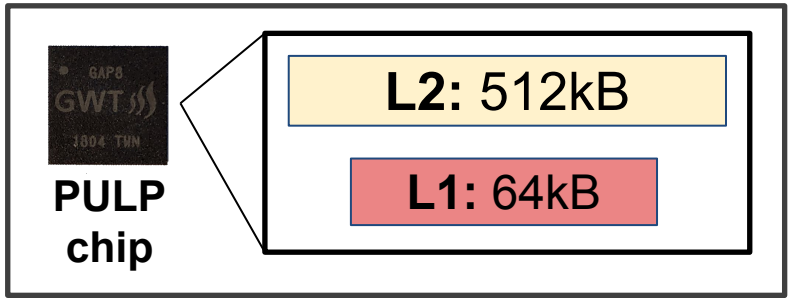
Case study: 1x1 conv2D

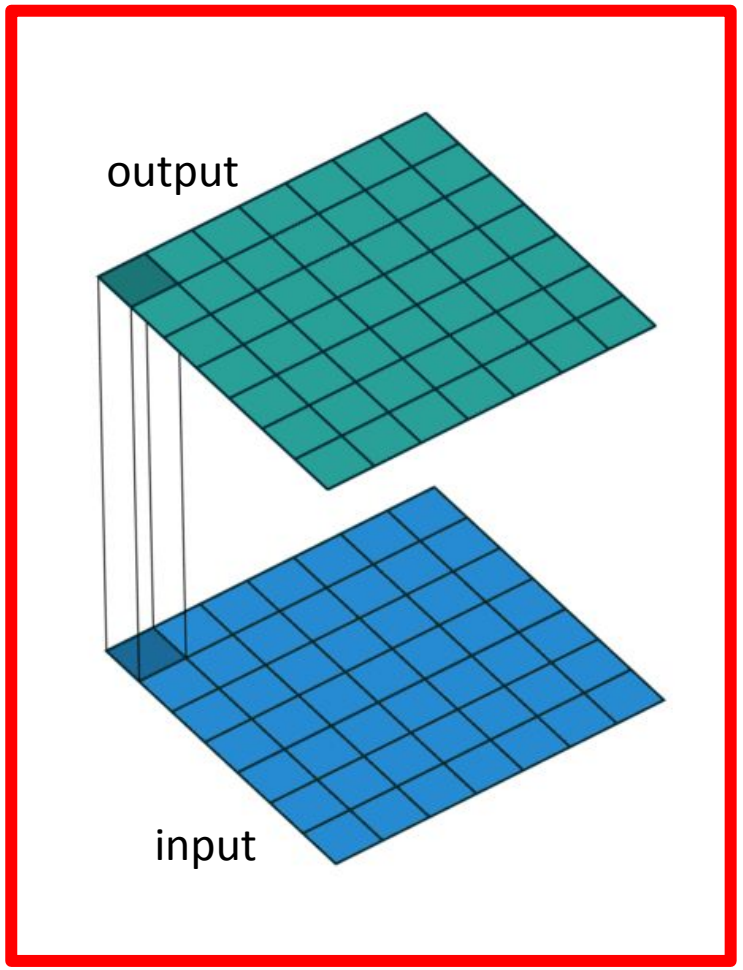
We tackle a 1x1 convolution with this sizes:

- Input = SPATIAL_DIM → defined by you
- Output = SPATIAL_DIM → defined by you
- Kernel = 1x1
- Stride = 1
- Padding = 0

NB: with conv1x1 the spatial size between input and output does not change!

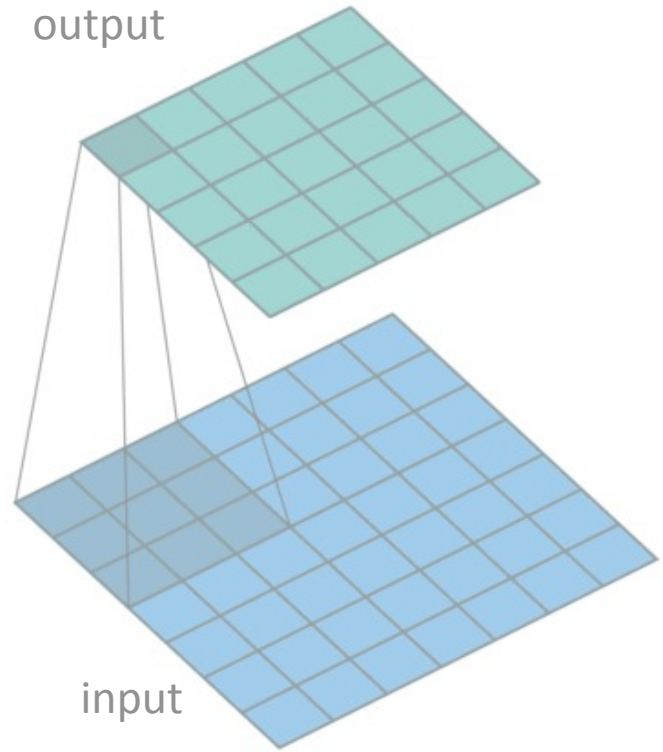
We want to fit into the L1 memory!





1x1 Convolution

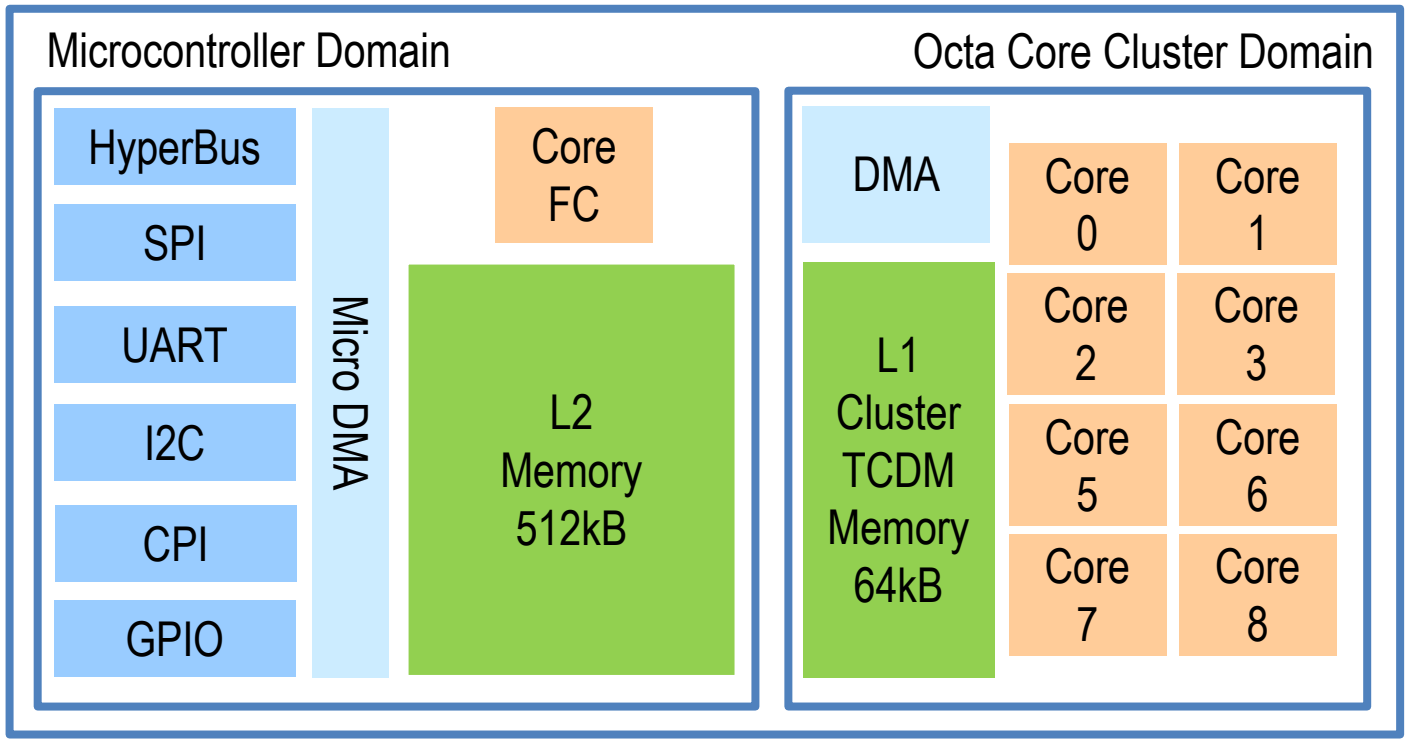
Used today!



3x3 Convolution

Used in lab04!

PULP Platform: today we focus on the 8-cores cluster

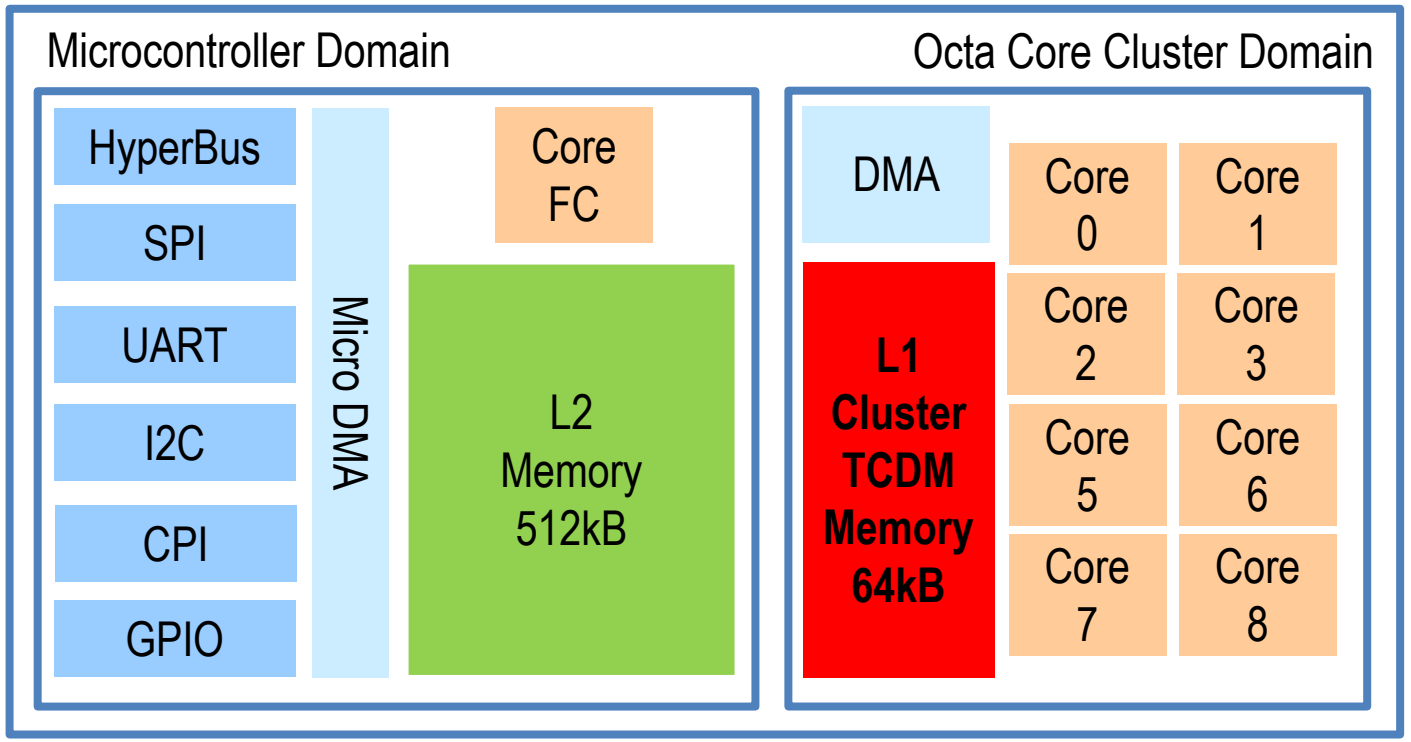


- **Cores**: 1 + 8
- **On-chip Memories**
  - A level 2 Memory, shared among all cores
  - A level 1 Memory, shared by the 8-cores cluster
- **cluster-DMA:** A multi-channel 1D/2D DMA, controlling the transactions between the L2 and L1 memories
- **micro-DMA**: A smart, lightweight and completely autonomous DMA () capable of handling complex I/O scheme
  - **Bus+Peripherals:** HyperBus, I2S, CPI, timers, SPI, GPIOs, etc…

NB: this is the architecture you find on the nano-drone!

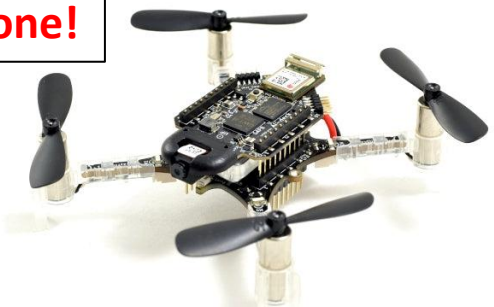
GitHub HW Project: https://github.com/pulp-platform/pulp
HW Documentation:
https://raw.githubusercontent.com/pulp-platform/pulp/master/doc/datasheet.pdf

PULP Platform: today we focus on the 8-cores cluster



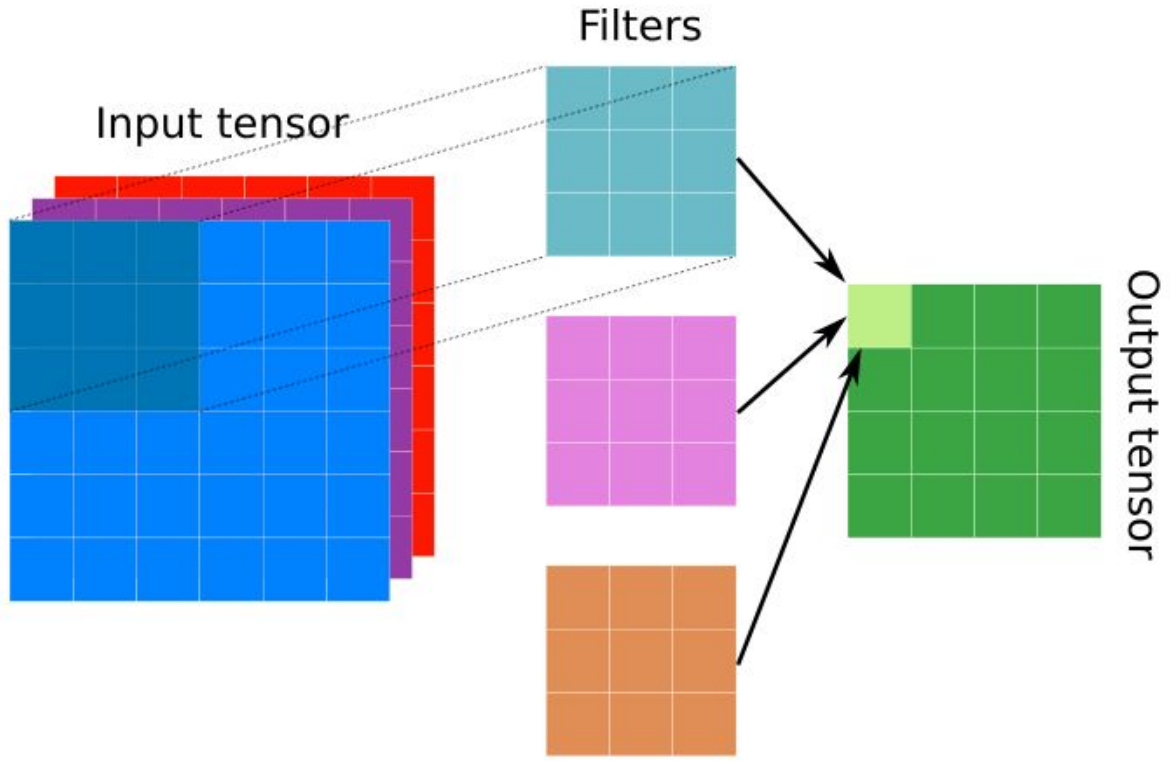
- **Cores**: 1 + 8
- **On-chip Memories**
  - A level 2 Memory, shared among all cores
  - A level 1 Memory, shared by the 8-cores cluster
- **cluster-DMA:** A multi-channel 1D/2D DMA, controlling the transactions between the L2 and L1 memories
- **micro-DMA**: A smart, lightweight and completely autonomous DMA () capable of handling complex I/O scheme
  - **Bus+Peripherals:** HyperBus, I2S, CPI, timers, SPI, GPIOs, etc…

NB: this is the architecture you find on the nano-drone!



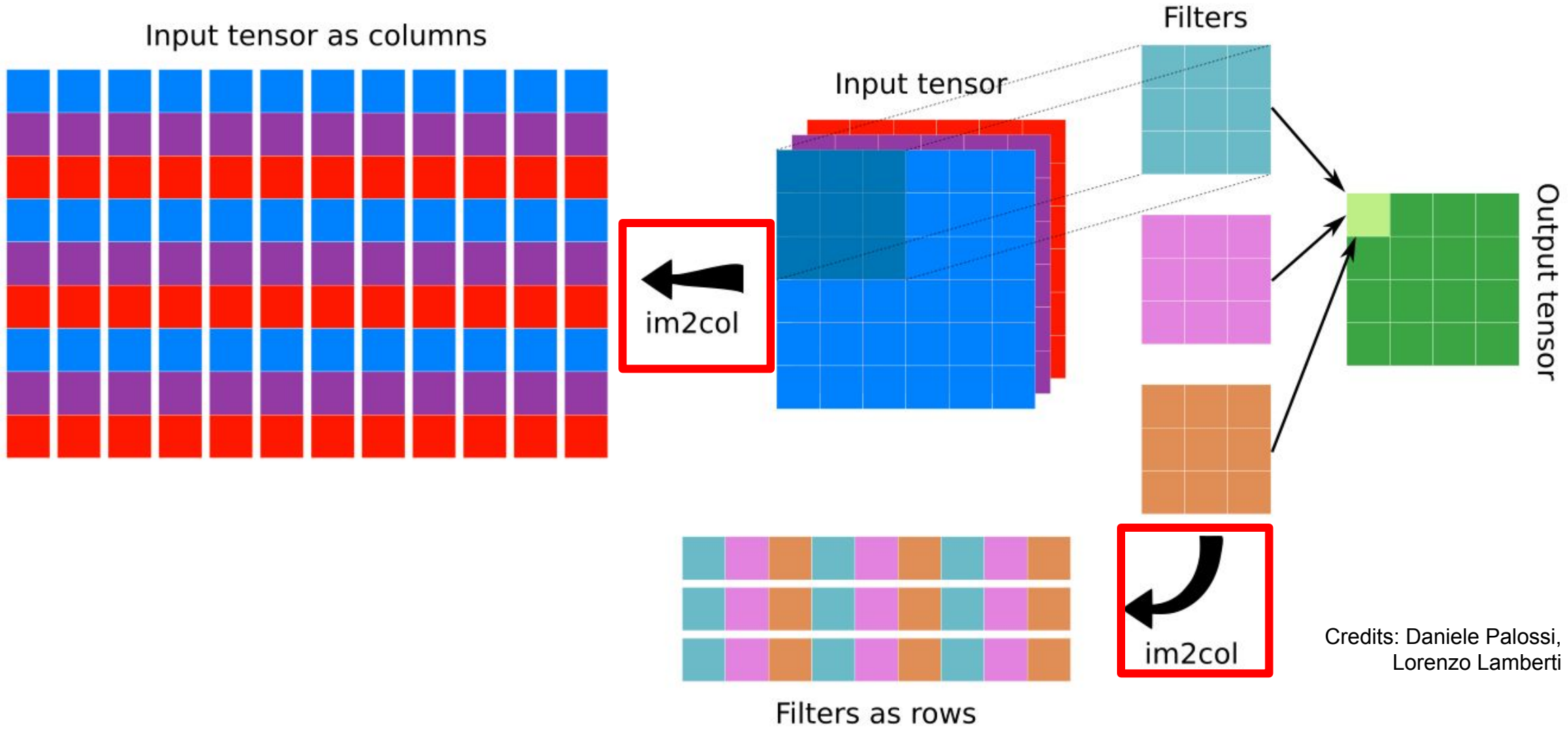
GitHub HW Project: https://github.com/pulp-platform/pulp
HW Documentation:
https://raw.githubusercontent.com/pulp-platform/pulp/master/doc/datasheet.pdf

Convolution Operation: naive



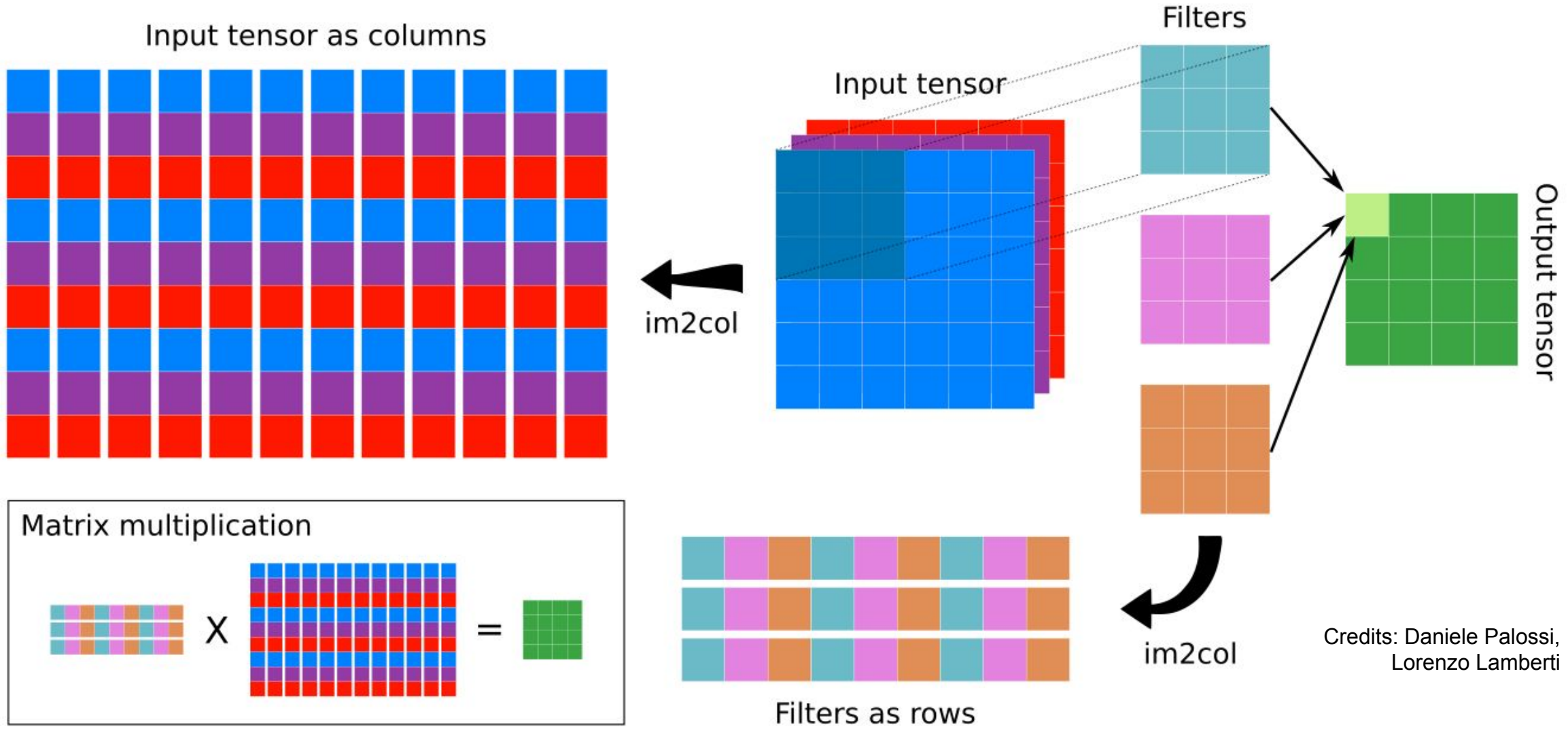
Credits: Daniele Palossi,
Lorenzo Lamberti

Convolution Operation: im2col and MatMul



Credits: Daniele Palossi, Lorenzo Lamberti

Convolution Operation: im2col and MatMul



Credits: Daniele Palossi, Lorenzo Lamberti

EX1: find maximum dimensions of layers fitting L1 without tiling

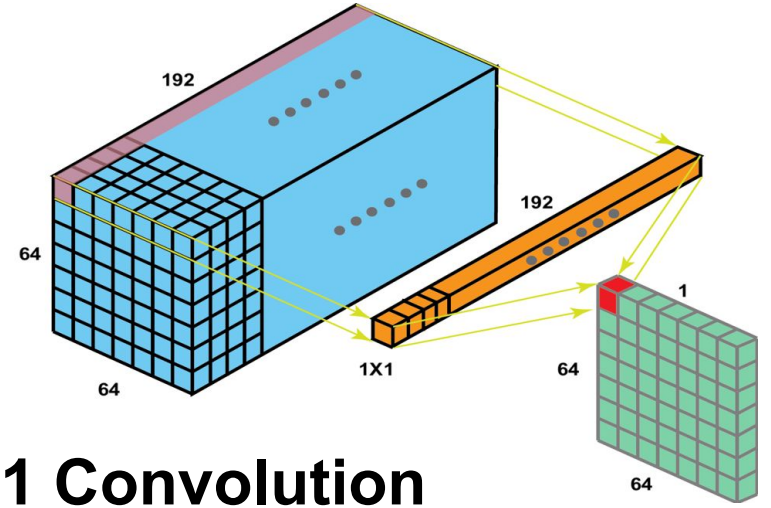
```
Prerequisites:
source setup_pulp_sdk.sh

Run the code:
$ python3 parameters_generate.py --channels=<add_here> --spatial_dimension=<add_here>
$ make clean all run
```

Follow the assignment document.

NB: Choose the exercise by uncommenting one of the following defines in `Inc/main.h`:

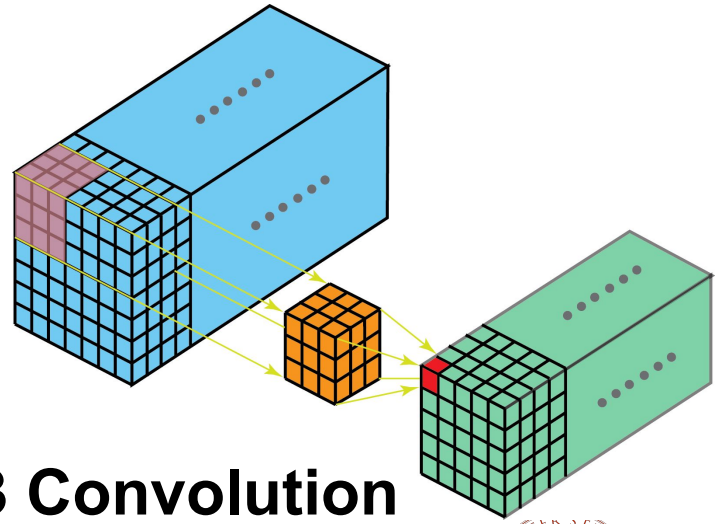
Remember to change “CHANNELS” and “SPATIAL_DIM” **in Src/main.c** when you change layer size!

```
#define EXERCISE1
//#define EXERCISE2
// #define EXERCISE3
```



1x1 Convolution

Used today!



3x3 Convolution

Used in lab04!

Exercise 1

L1 memory: 64kB = ~~64000~~ (consider 50KB +/- 2KB as Maximum).

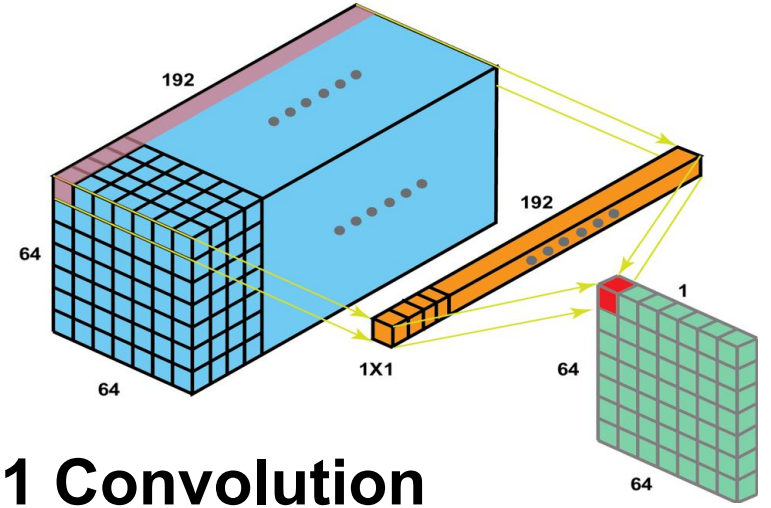
We must fit: input, kernel, output
Ch = 16
W, H = ?
Spatial dimension = W = H

Input size= W * H * Ch
Kernel_size= W_k * H_k * Ch_in * Ch_out = 1*1*16*16
Output_size= W * H * Ch
IM2COL size= 2 * 8 * W_k * H_k * Ch_in

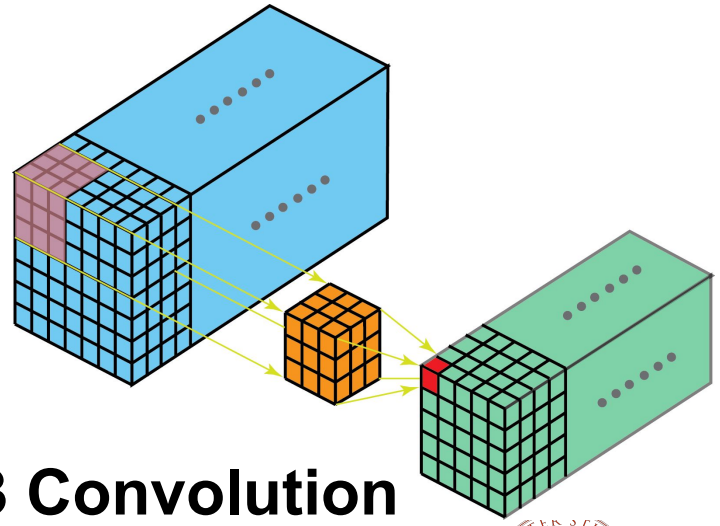
We want to solve this:
Input + kernel + output < L1

(W*H*16) + (16*16) + (W*H*16)=
(W^2*16) + (16*16) + (W^2*16) < 52000

16 W^2 + 256 + 16 W^2 < 52000
32 W^2 < 52000 - 256



1x1 Convolution
Used today!



3x3 Convolution
not used today

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

TASK2: fetch from L2

EX2: fetch data from L2

Run the code:

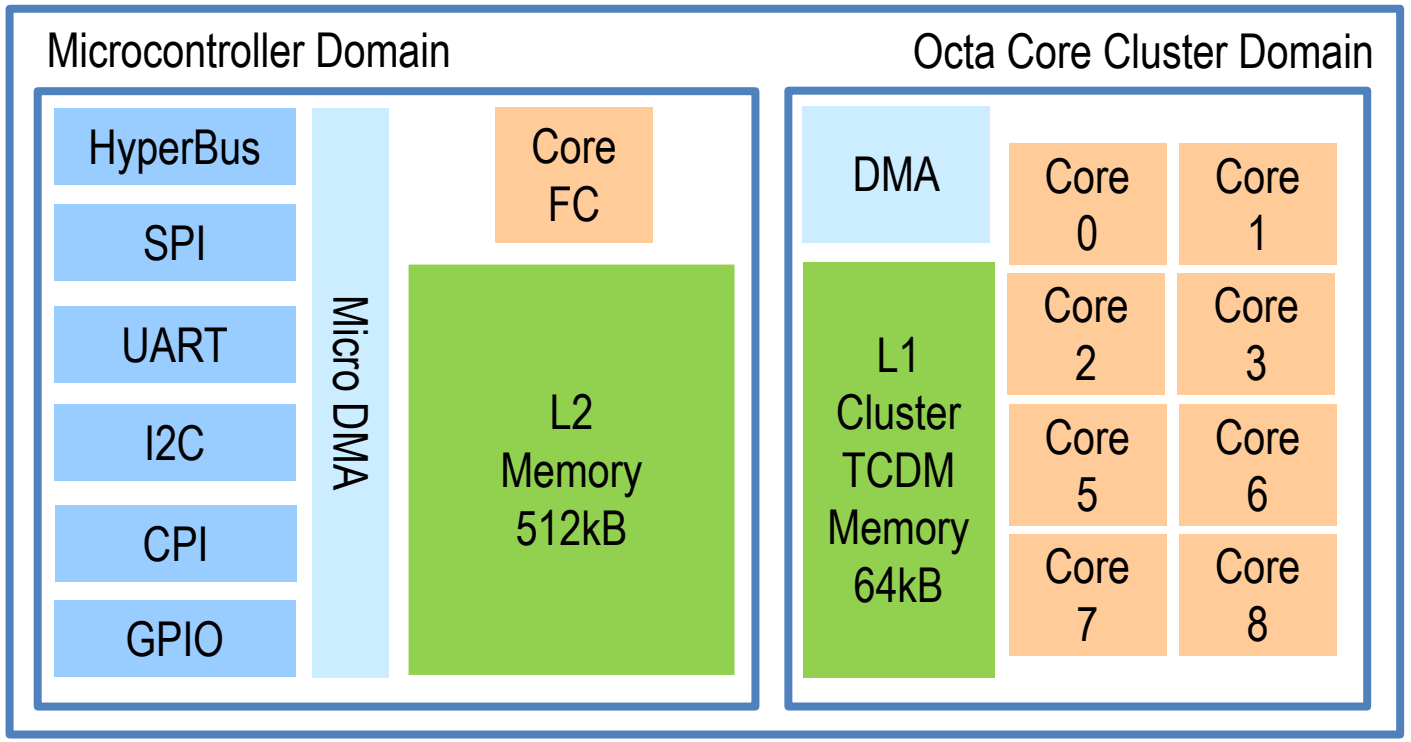
```
$ python3 parameters_generate.py --channels=<add_here> --spatial_dimension=<add_here>
$ make clean all run
```

Follow the assignment document.

NB: Choose the exercise by uncommenting one of the following defines in `Inc/main.h`:

```
// #define EXERCISE1
#define EXERCISE2
// #define EXERCISE3
```

PULP Platform: today we focus on the 8-cores cluster

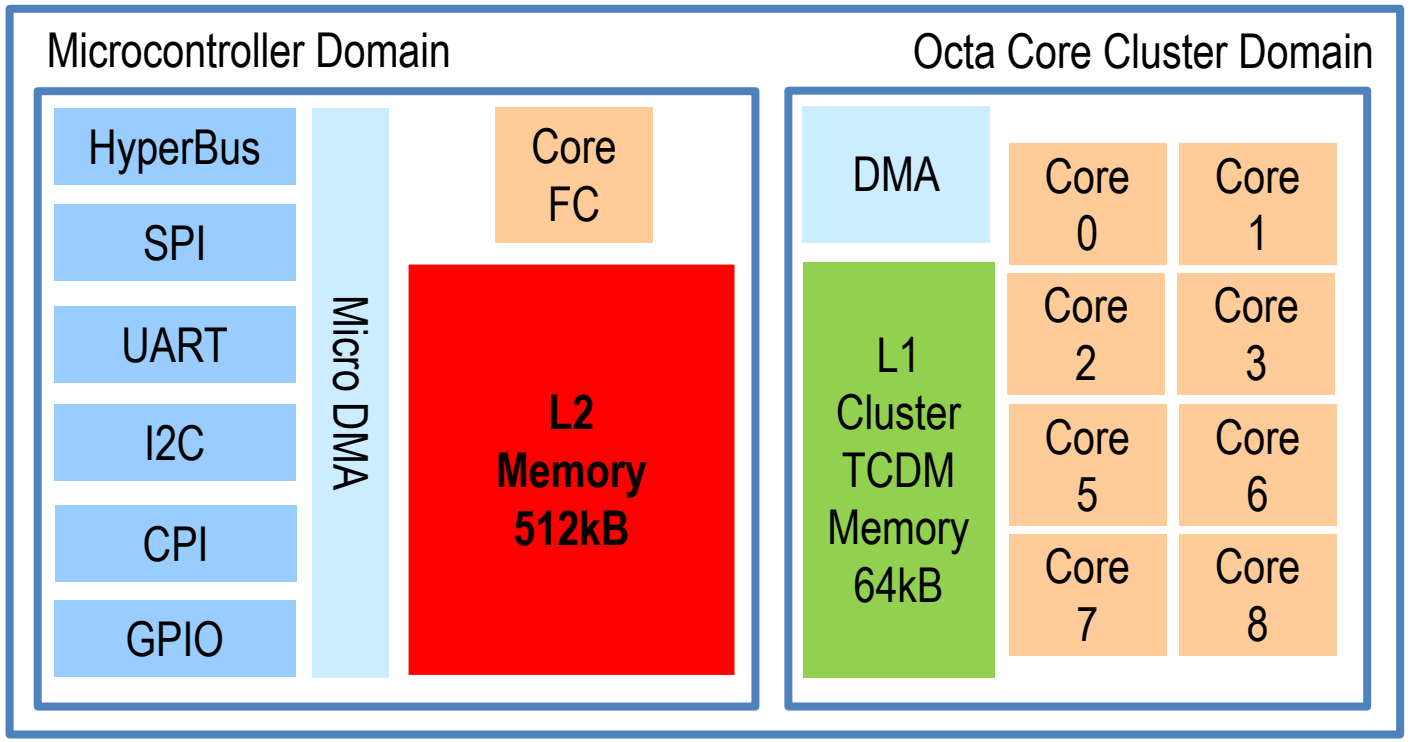


- **Cores**: 1 + 8
- **On-chip Memories**
  - A level 2 Memory, shared among all cores
  - A level 1 Memory, shared by the 8-cores cluster
- **cluster-DMA:** A multi-channel 1D/2D DMA, controlling the transactions between the L2 and L1 memories
- **micro-DMA**: A smart, lightweight and completely autonomous DMA () capable of handling complex I/O scheme
  - **Bus+Peripherals:** HyperBus, I2S, CPI, timers, SPI, GPIOs, etc…

NB: this is the architecture you find on the nano-drone!

GitHub HW Project: https://github.com/pulp-platform/pulp
HW Documentation:
https://raw.githubusercontent.com/pulp-platform/pulp/master/doc/datasheet.pdf

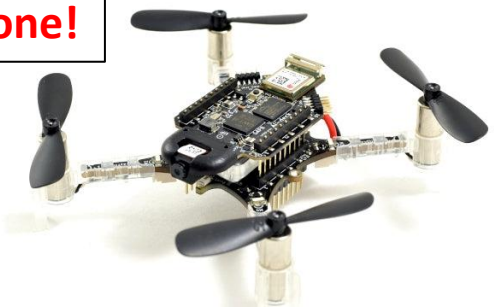
PULP Platform: today we focus on the 8-cores cluster



Slower access from cluster!!

- **Cores**: 1 + 8
- **On-chip Memories**
  - A level 2 Memory, shared among all cores
  - A level 1 Memory, shared by the 8-cores cluster
- **cluster-DMA:** A multi-channel 1D/2D DMA, controlling the transactions between the L2 and L1 memories
- **micro-DMA**: A smart, lightweight and completely autonomous DMA () capable of handling complex I/O scheme
  - **Bus+Peripherals:** HyperBus, I2S, CPI, timers, SPI, GPIOs, etc…

NB: this is the architecture you find on the nano-drone!



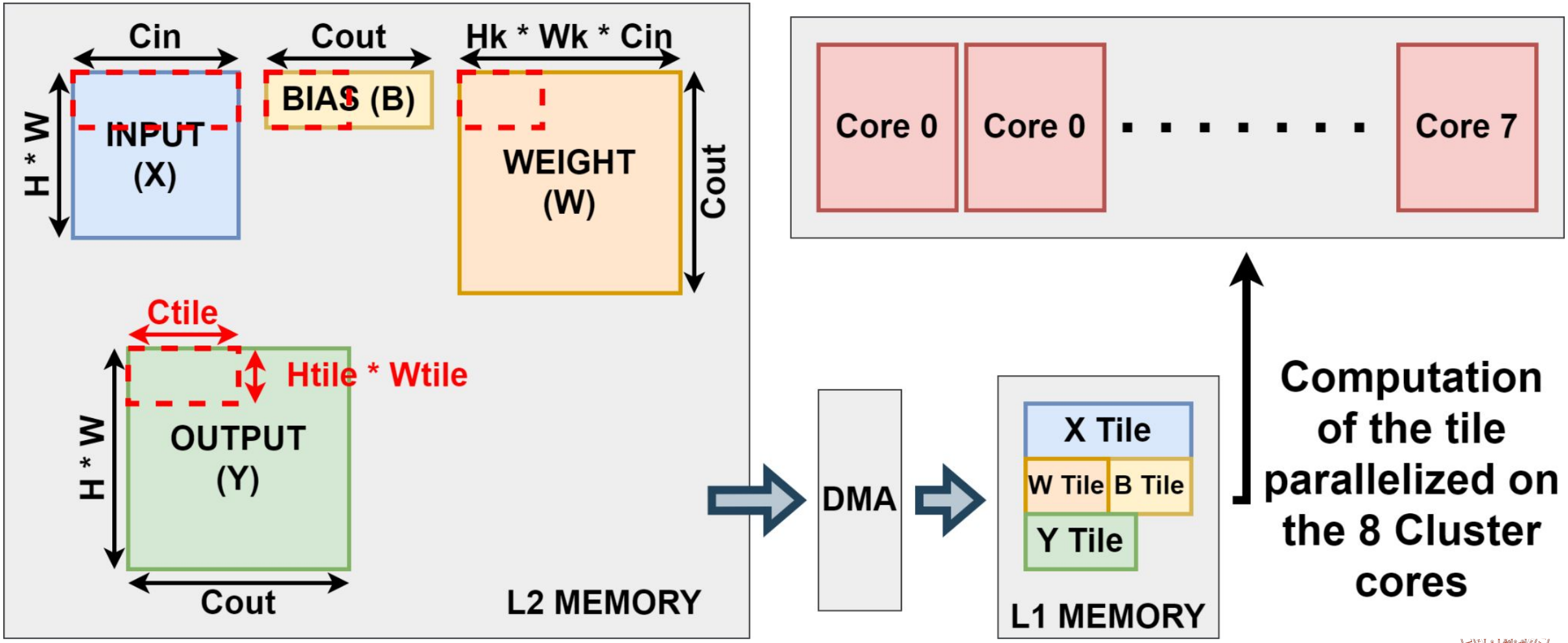
GitHub HW Project: https://github.com/pulp-platform/pulp
HW Documentation:
https://raw.githubusercontent.com/pulp-platform/pulp/master/doc/datasheet.pdf

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

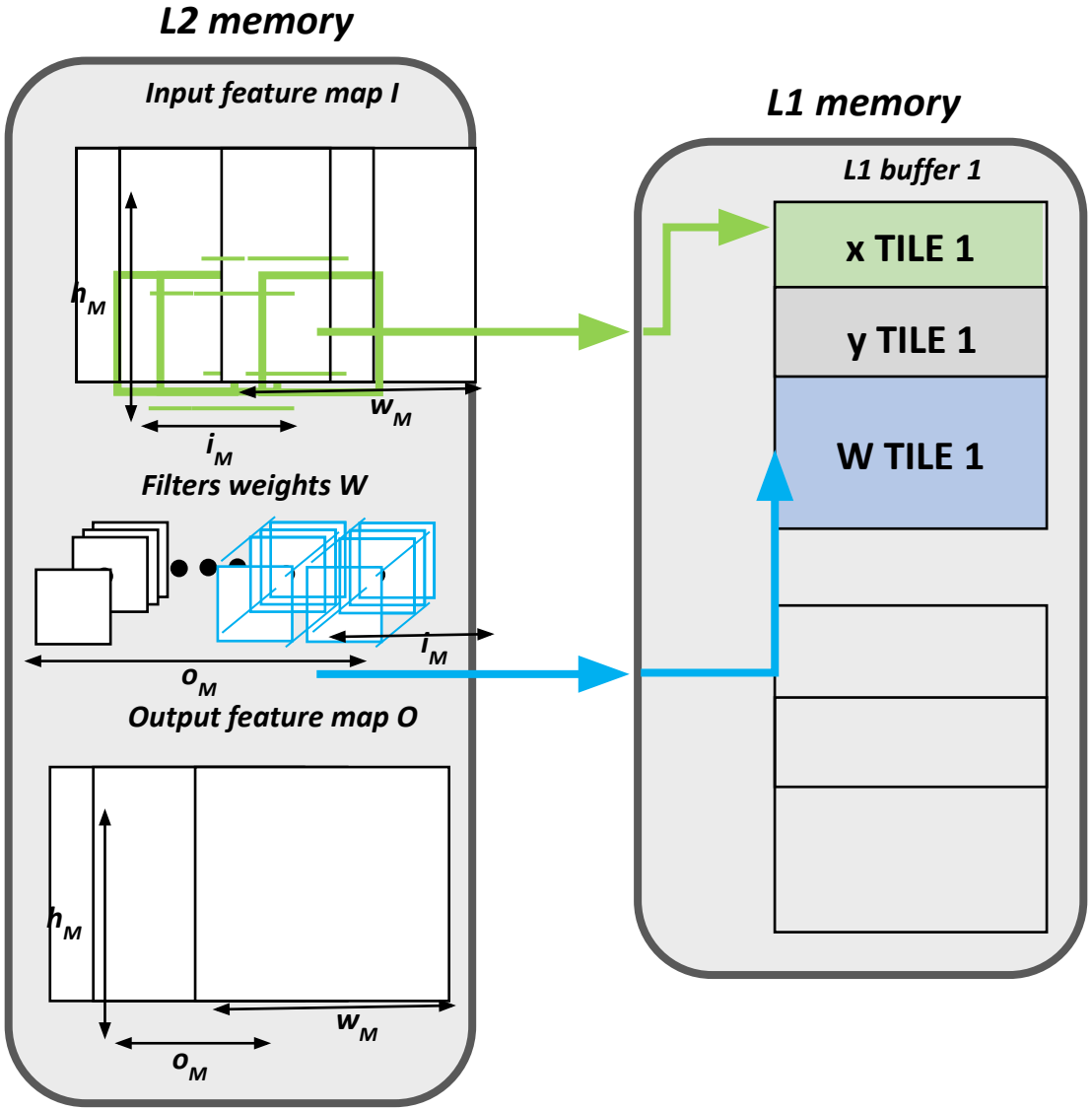
TASK3: Tiling

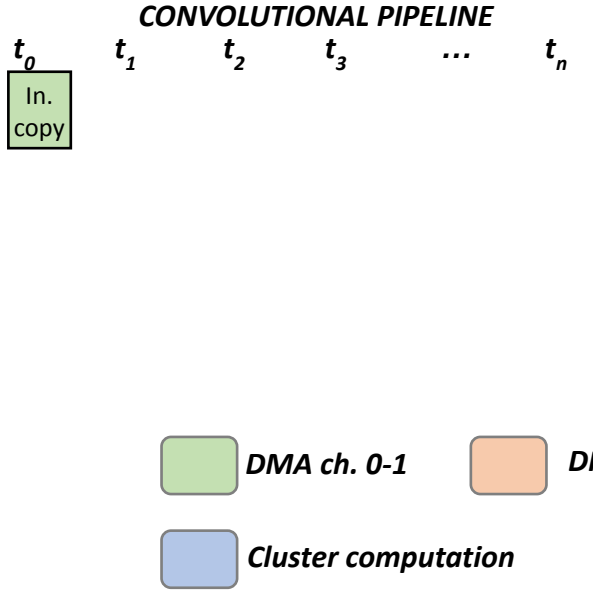
Tiling

Tiling: for each output chunk (tile), bring to L1 only the portion of the input/weight tensors necessary to compute it, then move the result to L2 an go on with the next tile!

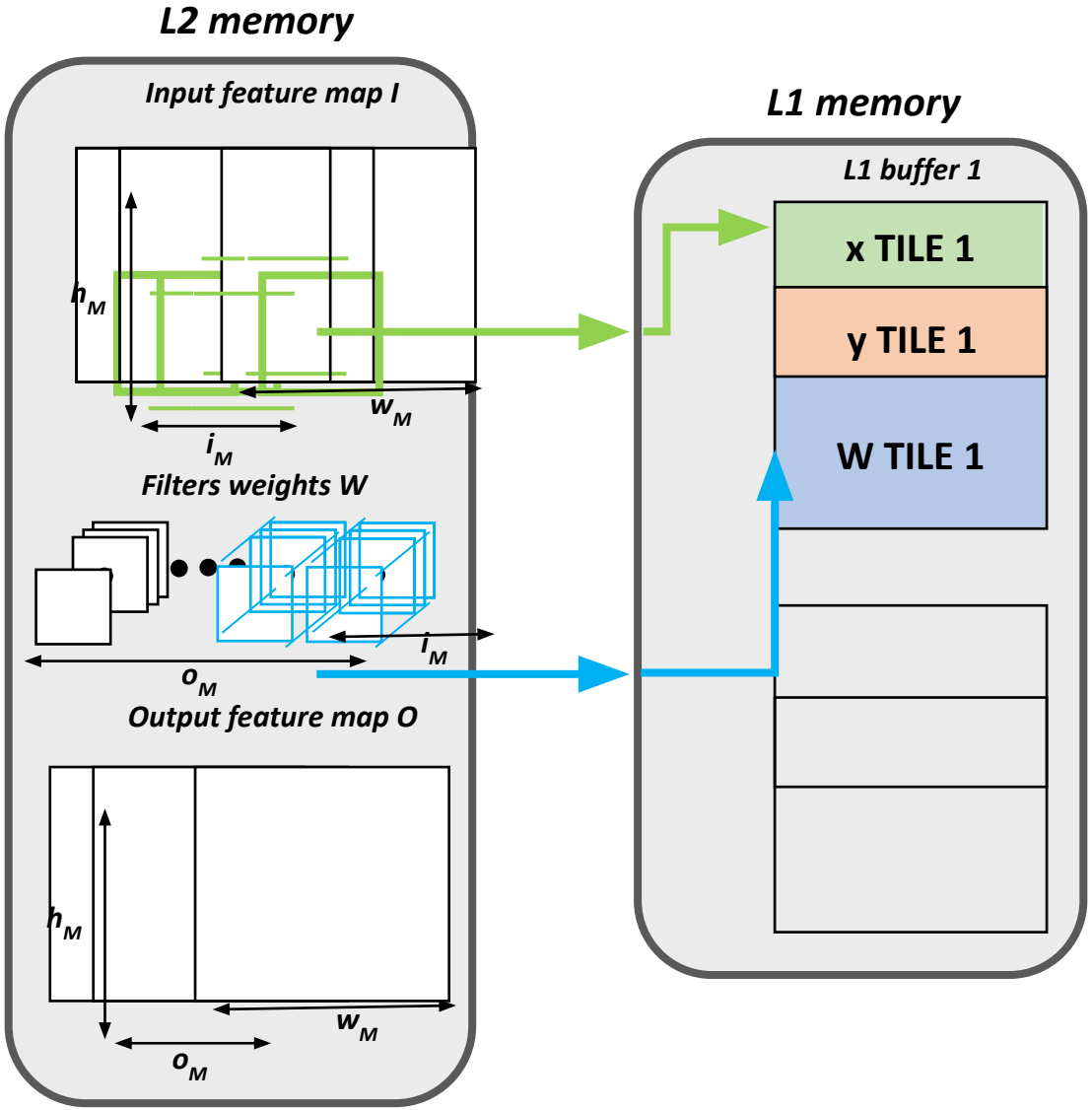


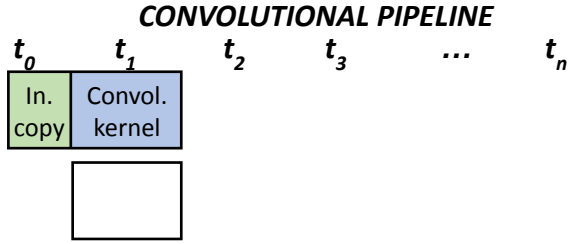
Tiling from L2 to L1



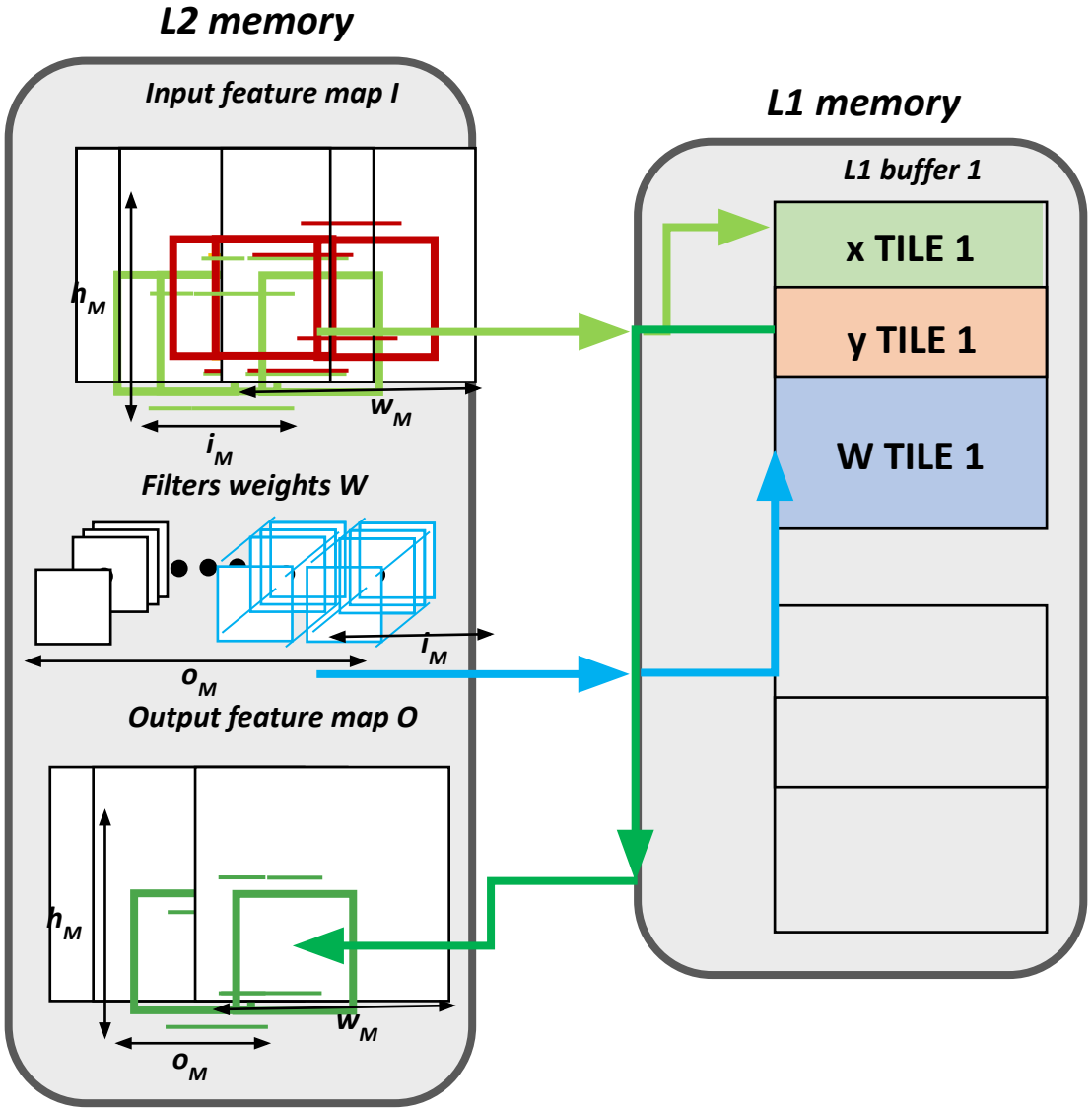


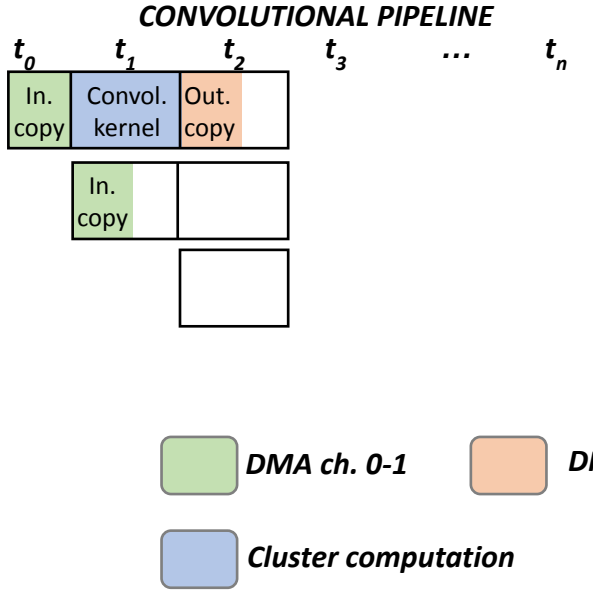
Tiling from L2 to L1





Tiling from L2 to L1





EX3: Tiling layer

Run the code:

```
$ python3 parameters_generate.py --channels=#### --spatial_dimension=####
$ make clean all run
```

Follow the assignment document.

NB: Choose the exercise by uncommenting one of the following defines in `Inc/main.h`:

```
// #define EXERCISE1
// #define EXERCISE2
#define EXERCISE3
```

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEI – Università di Bologna

www.unibo.it