Computer Architecture

Final Project

Outline

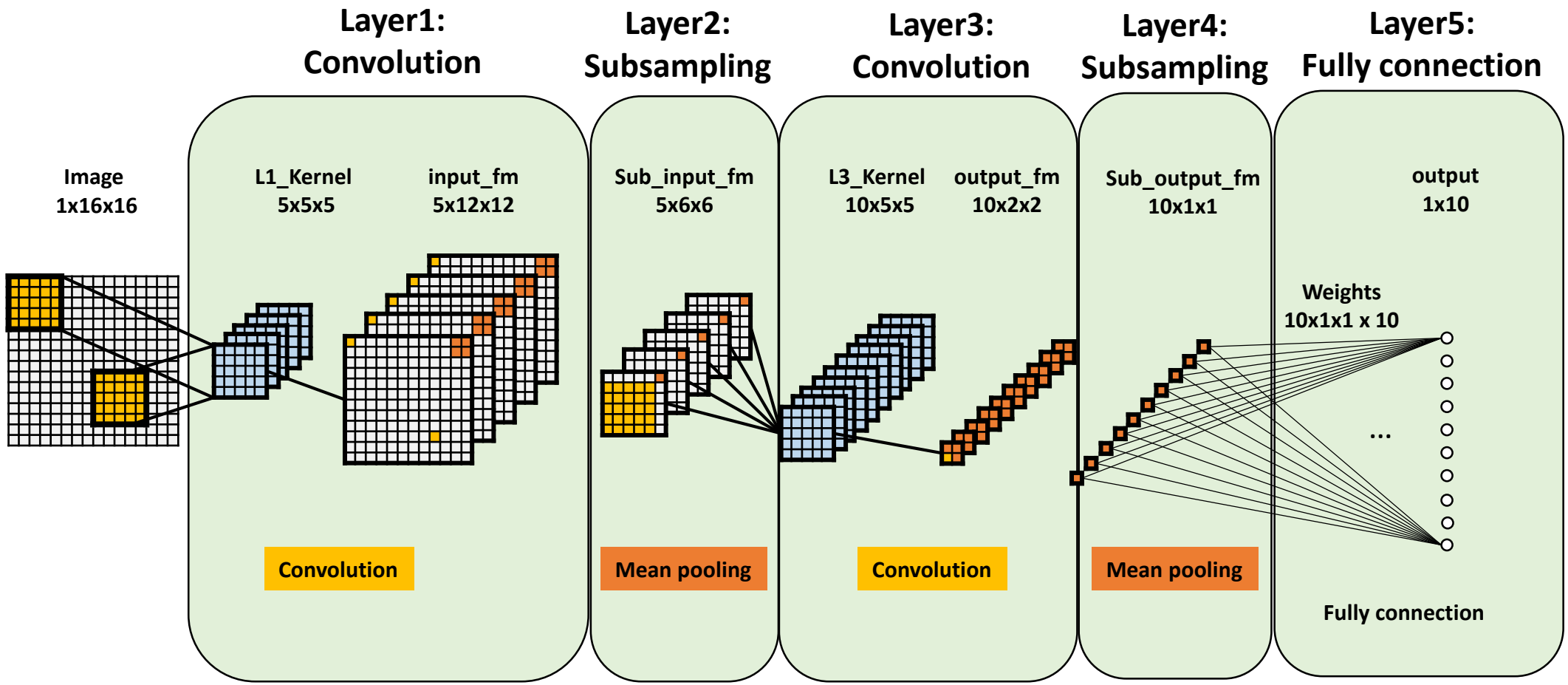
- Introduction
- The Architecture of CNN
- Spec.
- Grading Policy

Introduction

- **In this final project, you are going to design a simple version of CNN on the GPU. The simple version means that you only need to consider the feed-forward part and its architecture is just a 5-layer structure. Some of process are omitted such as activation function and bias. However, it still reserve the computational behavior.**

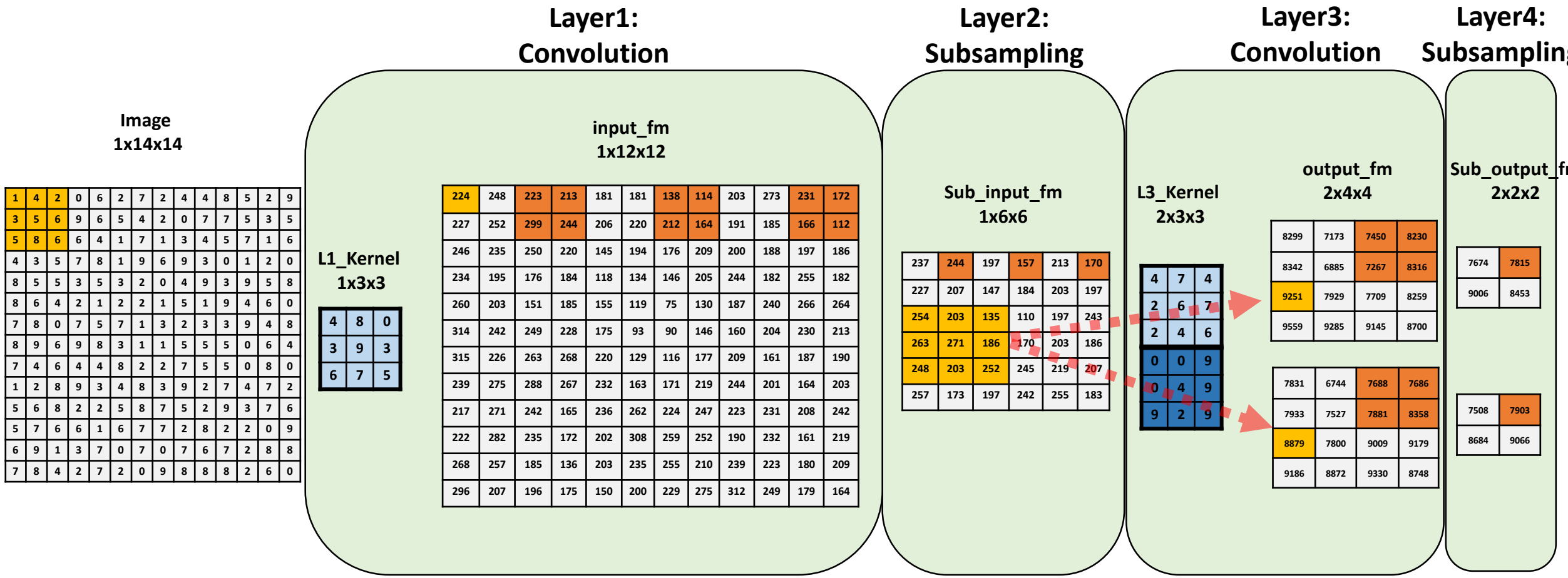
- **CNN is composed of three types of layers:**
  - Convolution layer
  - Subsampling layer
  - Fully connection layer

The Architecture of CNN



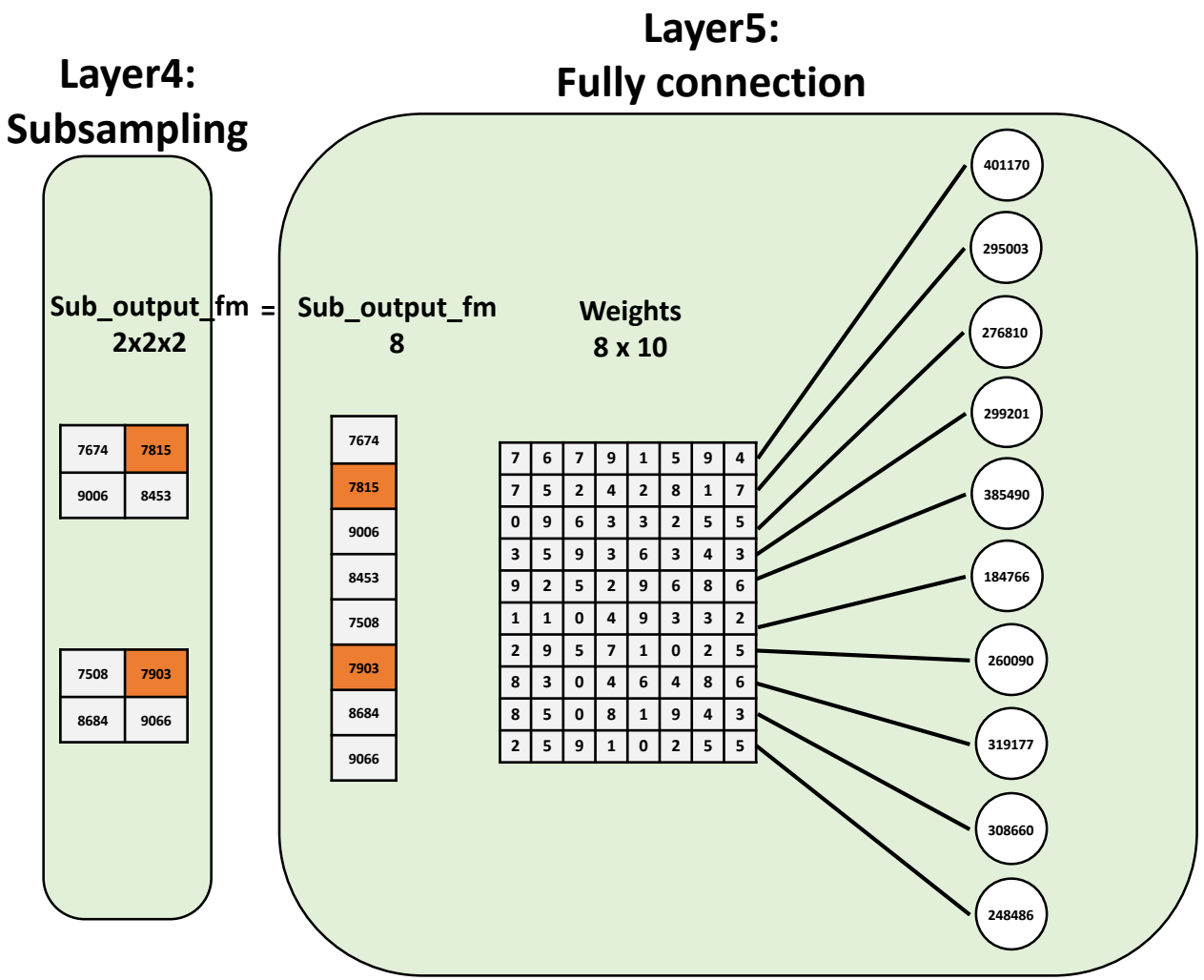
Example:

numImage = 1
imageSize = 14
l1_NumKernel = 1
l1_KernelSize = 3
l3_NumKernel = 2
l3_KernelSize = 3



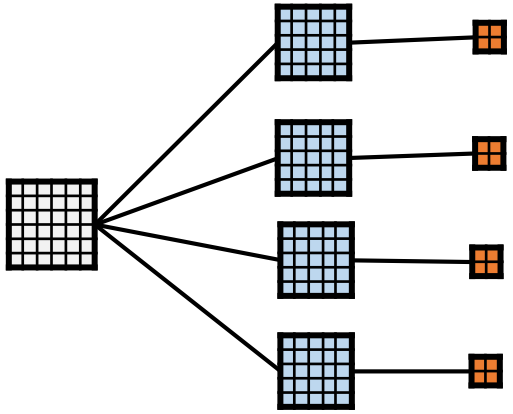
Example(cont.):

numImage = 1
imageSize = 14
l1_NumKernel = 1
l1_KernelSize = 3
l3_NumKernel = 2
l3_KernelSize = 3

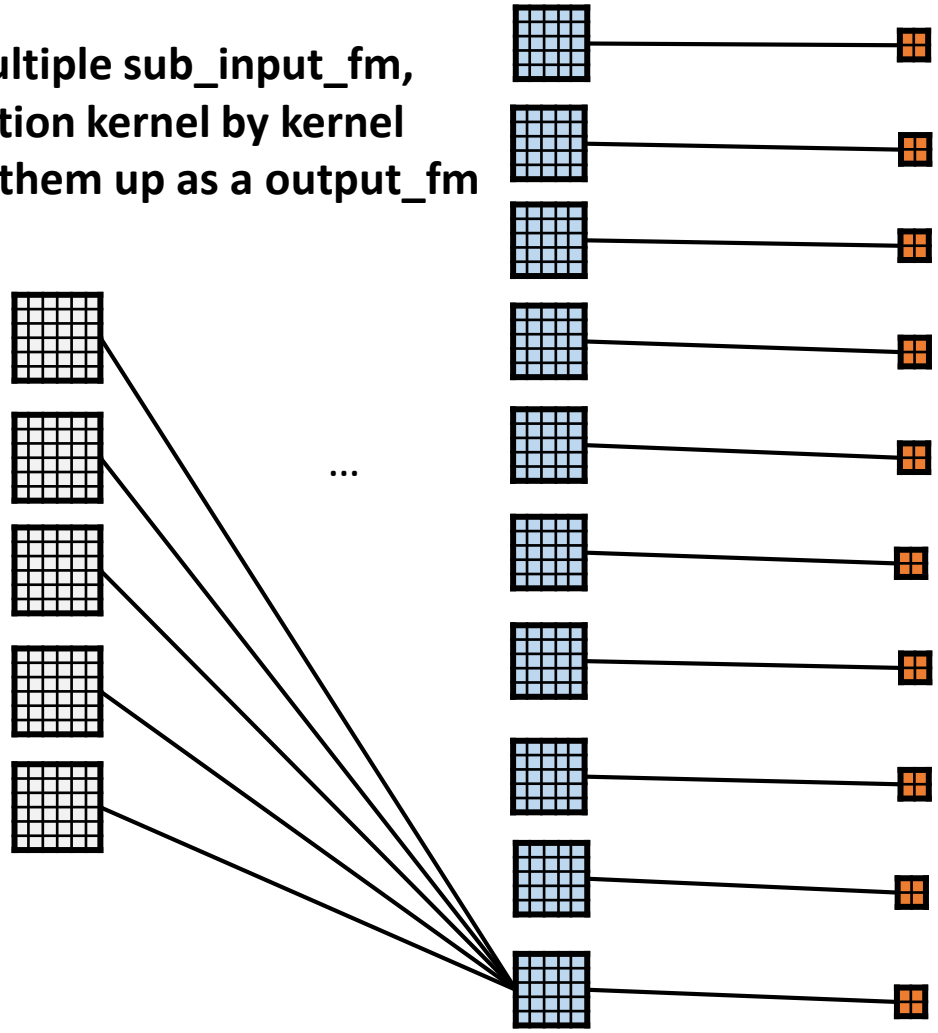


Example: Convolution Layer Case

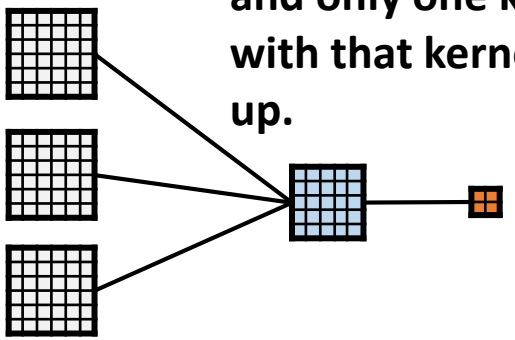
If there is one sub_input_fm, do convolution with each kernel.



If there are multiple sub_input_fm, All do convolution kernel by kernel and then sum them up as a output_fm



If there are multiple sub_input_fm and only one kernel, do convolution with that kernel and then sum them up.



Final Project: Convolution Neural Network

In The Final Project, You Have to…

- **Write OpenCL to design the implementation of CNN.**
  - Host program:
    - Create memory buffer
    - Send data
    - Enqueue task to GPU
    - Read result from GPU
    - …
  - Kernel
    - Convolution
    - Subsampling
    - Fully connection
- **Write a report to describe how you implement your design**
  - Parallelized algorithm
    - Work-item's task
    - Work-group size
    - Techniques
    - …
  - Describe your design
    - How to use your design
    - Use report files to verify your design
    - …
  - Show your statistics

Spec.

Software

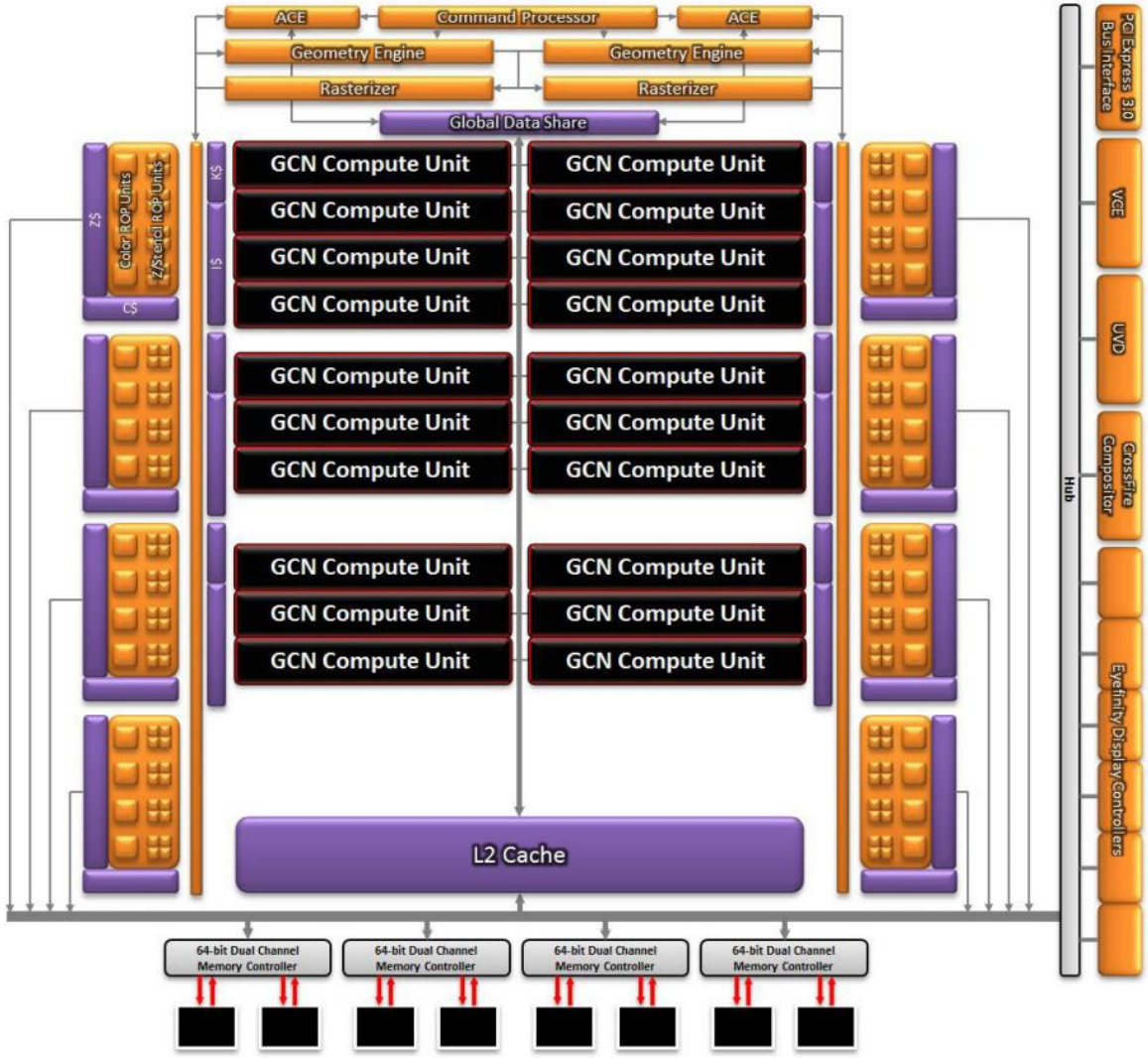
- **Data type:**
  - Every element of images is an integer from 0 to 9.
  - Every element of kernels is an integer from 0 to 9.
  - Every element of weights is an integer from 0 to 9.
  - Above three are stored into an 1D array by row-major, individually.
  - Number of image: 1 ~ 20
  - Image size: 16x16 ~ 128x128
  - Number of kernel: 1 ~ 10
  - Kernel size: 3x3 ~ 15x15
  - Output node: 10
- **Your code could run on native or M2S**

Hardware

- **Do not change hardware architecture**

AMD Radeon™ HD 7870	
Frequency	1000 MHz
Compute Unit	20 #
Stream Processors	1280
L1 Cache (one per CU)	16 KB
L1C Latency	1 clock
L2 Cache (one per 4 CUs)	128 KB
L2 Cache Latency	10 clocks

AMD Radeon™ HD 7870



←Architecture Overview

↓ Cache Hierarchy

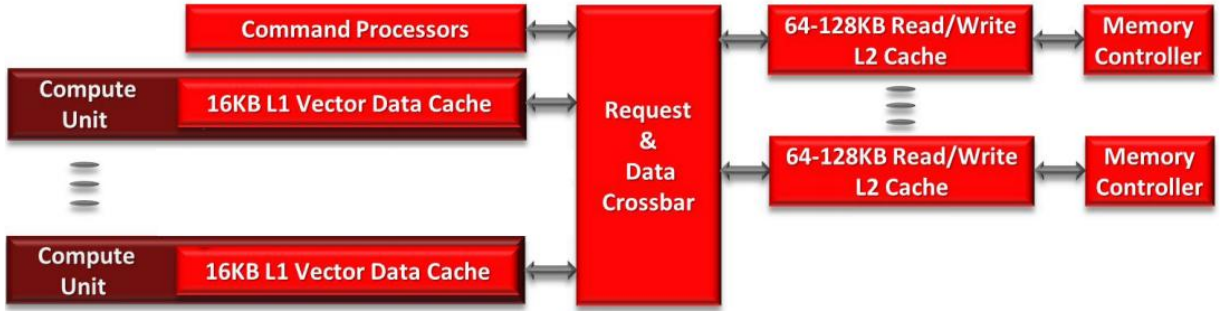


Figure from: https://www.amd.com/Documents/GCN_Architecture_whitepaper.pdf

Analysis on Multi2sim

Collect Report

- **In the Lab package, you will see**
  - EXAMPLE: CNN_C
  - CNN:
    - 00_Configuration: ＊Contain CPU, GPU, Memory configuration files
    - 00_Report ＊After 03 and 04, the reports will be generated here
      Note that the reports will be covered every running 04.
    - 01_COMPILE_ON_NATIVE.sh* ＊Could run on native after 01
    - 03_COMPILE_ON_M2S.sh* ＊Compile source code by using M2S's library
    - 04_RUN_ON_M2S.sh* ＊Start to simulate on M2S
    - 09_CLEAN_REPORT.sh* ＊Clean all report files
    - CNN.c
    - CNN.cl ＊Write your code here
    - parameter.h

Analysis on Multi2sim

How to Get The Statistics: CPU/GPU Report

```
;
; CPU Configuration
;
 Config
;
; Simulation Statistics
;
; Global statistics
[ Global ]

Cycles = 198145463
Time = 159.44
CyclesPerSecond = 1242776
MemoryUsed = 29409280
MemoryUsedMax = 29409280
```

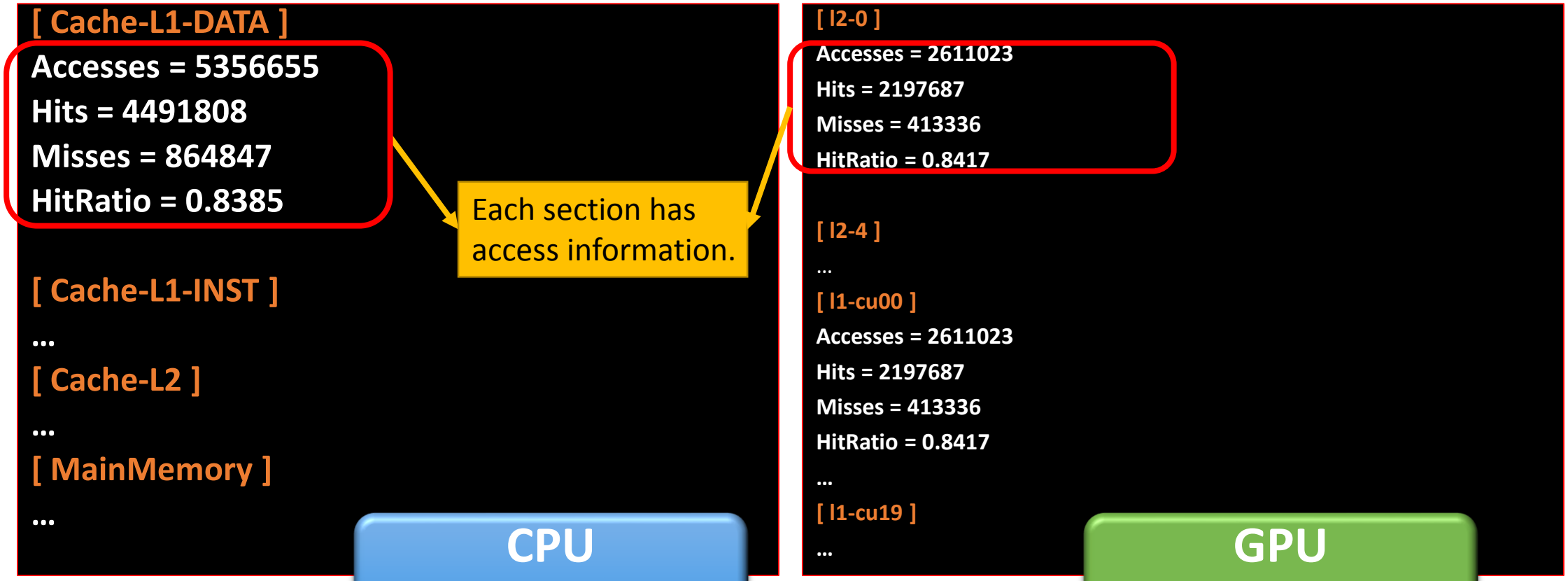
```
;
; GPU Configuration
;
Config
;
; Simulation Statistics
;
; Global statistics
[ Device ]
...
VectorMemInstructions = xxx
Cycles = 5600711
InstructionsPerCycle = xxx

[ ComputeUnit 0 ]
...
[ ComputeUnit 19 ]
...
```

"Cycles" is used to measure the performance!! Remember to record it!!!

Each CU has some other information.

Analysis on Multi2sim

How to Get The Statistics: Memory Report



Grading Policy

Two Parts:

- **[60%] OpenCL code:**
  - [45%] 3 cases to verify the functionality (small/middle/big image, kernel size)
  - [15%] Performance (on Multi2sim)
    - Each case accounts for 10pts:
      The best gets 10, the worst gets 1.
    - Measured by “Cycles” from GPU report.
- **[40%] Report (IEEE double column format):**
  - [20%] Description of your design including work-item’s task, techniques, and so on.
  - [10%] How to use your design
  - [10%] Show statistics

  Note: report do not exceed 4 pages.

(due 1/14)

About Submission

Please Observe The Following Rules:

- **ONE group submit ONE report.**
- **Please write down ALL the group members (1 ~ 3 members per group).**
- **The group is the same as previous lab.**
- **Each one will get the same score in the same group.**
- **Upload the package to E3 “作業列表 -> FP”, and it must contains**
  - Report (accept PDF/word).
  - CNN.c
  - CNN.cl
  - parameter.h
- **Deadline: 1/14 23:59:59**
- **No score for late submission.**

Login Into Workstation

Guideline

- **Please use SSH tool to connect to the workstation**
  - Such as “mobaXterm” , “Putty”…
- **Login**
  - Host name : 140.113.225.130
  - Port: 22
  - Type: SSH
  - Personal account is the same as previous lab.
- **Get the package**
  - Please type the command:
    $tar -xvf /tmp/FP.tar

Reference

- White paper - AMD GRAPHICS CORES NEXT (GCN) ARCHITECTURE
  https://www.amd.com/Documents/GCN_Architecture_whitepaper.pdf