

Computer Architecture

Final Project

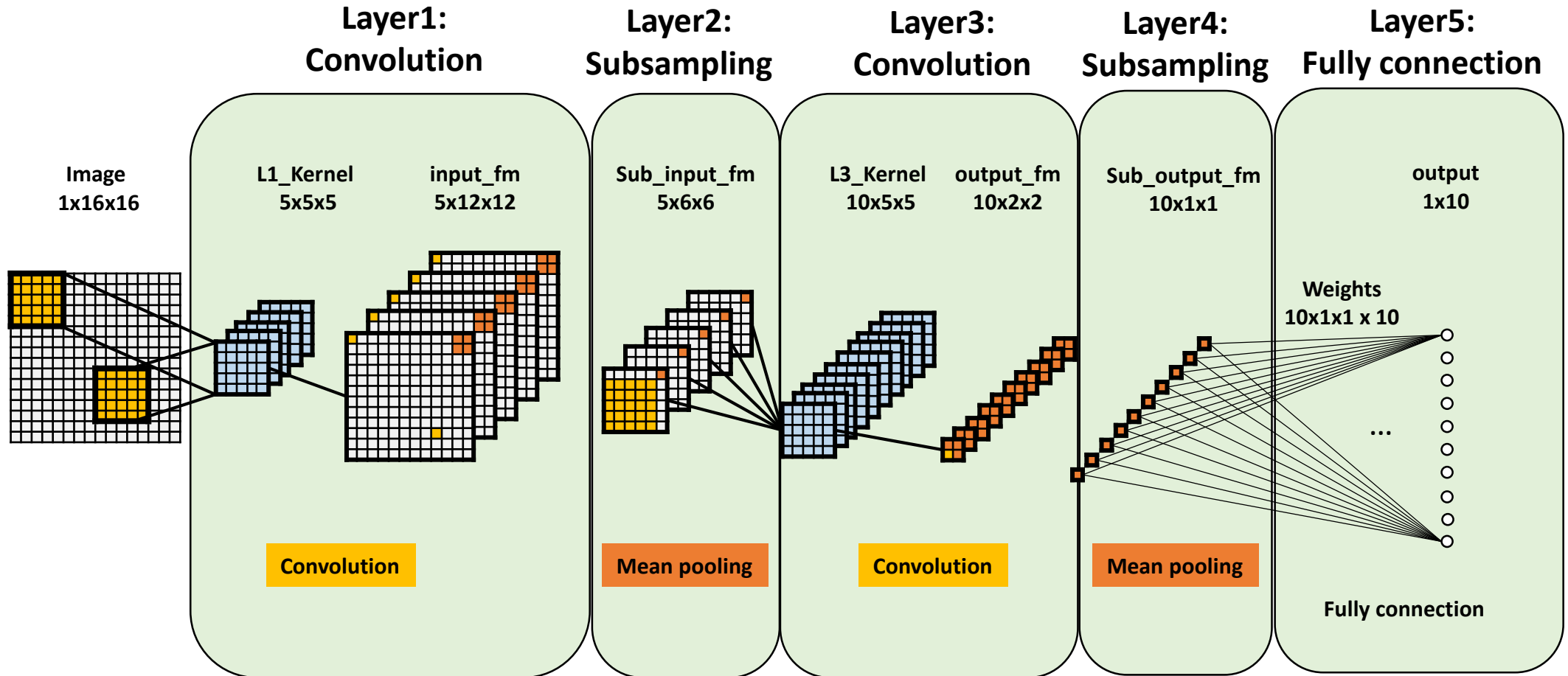
Outline

- **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

Layer1: Convolution

Image
1x14x14

1	4	2	0	6	2	7	2	4	4	8	5	2	9
3	5	6	9	6	5	4	2	0	7	7	5	3	5
5	8	6	6	4	1	7	1	3	4	5	7	1	6
4	3	5	7	8	1	9	6	9	3	0	1	2	0
8	5	5	3	5	3	2	0	4	9	3	9	5	8
8	6	4	2	1	2	2	1	5	1	9	4	6	0
7	8	0	7	5	7	1	3	2	3	3	9	4	8
8	9	6	9	8	3	1	1	5	5	5	0	6	4
7	4	6	4	4	8	2	2	7	5	5	0	8	0
1	2	8	9	3	4	8	3	9	2	7	4	7	2
5	6	8	2	2	5	8	7	5	2	9	3	7	6
5	7	6	6	1	6	7	7	2	8	2	2	0	9
6	9	1	3	7	0	7	0	7	6	7	2	8	8
7	8	4	2	7	2	0	9	8	8	8	2	6	0

L1_Kernel
1x3x3

4	8	0
3	9	3
6	7	5

input_fm
1x12x12

224	248	223	213	181	181	138	114	203	273	231	172
227	252	299	244	206	220	212	164	191	185	166	112
246	235	250	220	145	194	176	209	200	188	197	186
234	195	176	184	118	134	146	205	244	182	255	182
260	203	151	185	155	119	75	130	187	240	266	264
314	242	249	228	175	93	90	146	160	204	230	213
315	226	263	268	220	129	116	177	209	161	187	190
239	275	288	267	232	163	171	219	244	201	164	203
217	271	242	165	236	262	224	247	223	231	208	242
222	282	235	172	202	308	259	252	190	232	161	219
268	257	185	136	203	235	255	210	239	223	180	209
296	207	196	175	150	200	229	275	312	249	179	164

Layer2: Subsampling

Sub_input_fm
1x6x6

237	244	197	157	213	170
227	207	147	184	203	197
254	203	135	110	197	243
263	271	186	170	203	186
248	203	252	245	219	207
257	173	197	242	255	183

Layer3: Convolution

L3_Kernel
2x3x3

4	7	4
2	6	7
2	4	6
0	0	9
0	4	9
9	2	9

output_fm
2x4x4

8299	7173	7450	8230
8342	6885	7267	8316
9251	7929	7709	8259
9559	9285	9145	8700

7831	6744	7688	7686
7933	7527	7881	8358
8879	7800	9009	9179
9186	8872	9330	8748

Layer4: Subsampling

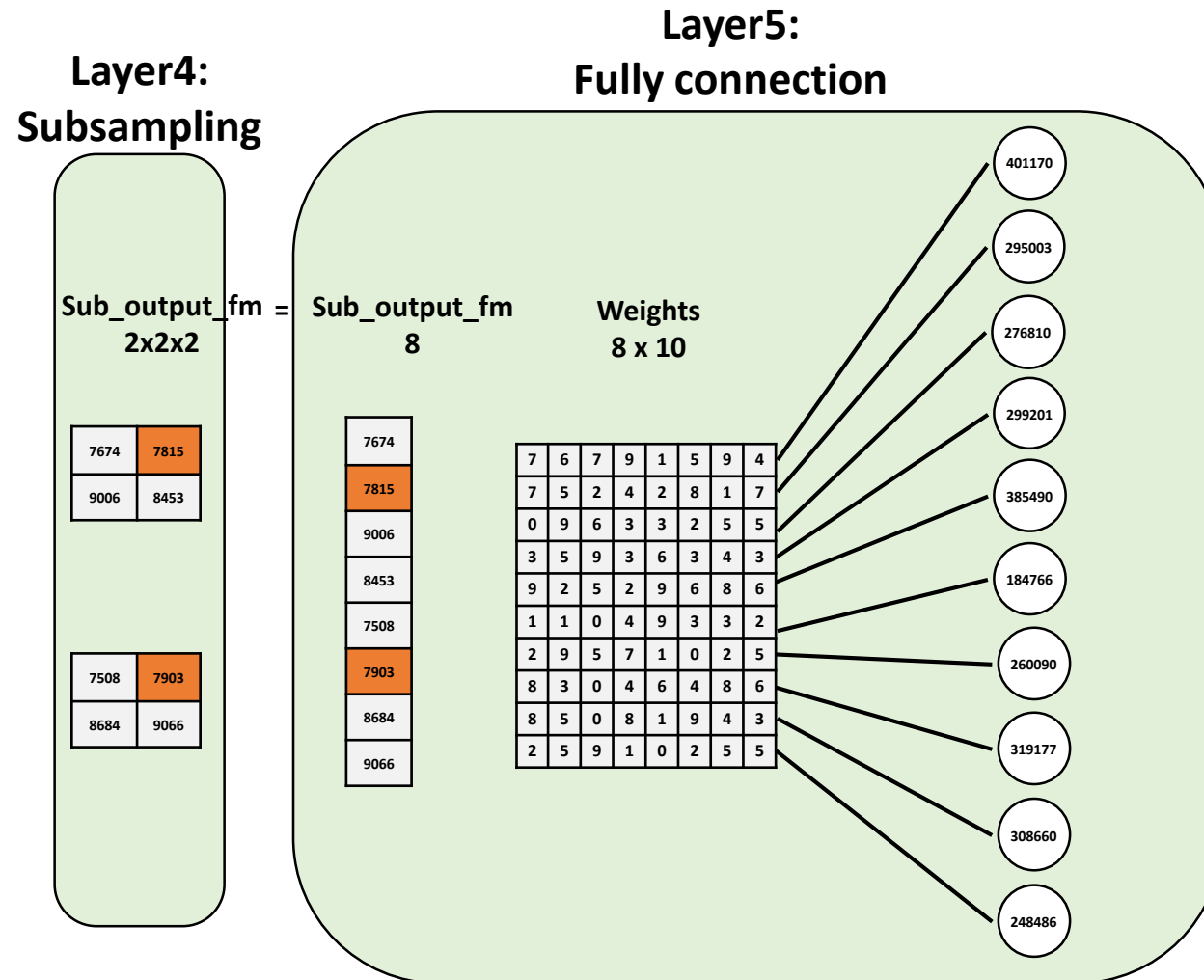
Sub_output_fm
2x2x2

7674	7815
9006	8453

7508	7903
8684	9066

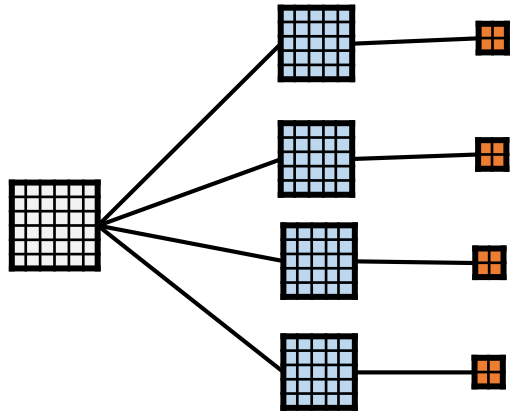
Example(cont.):

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

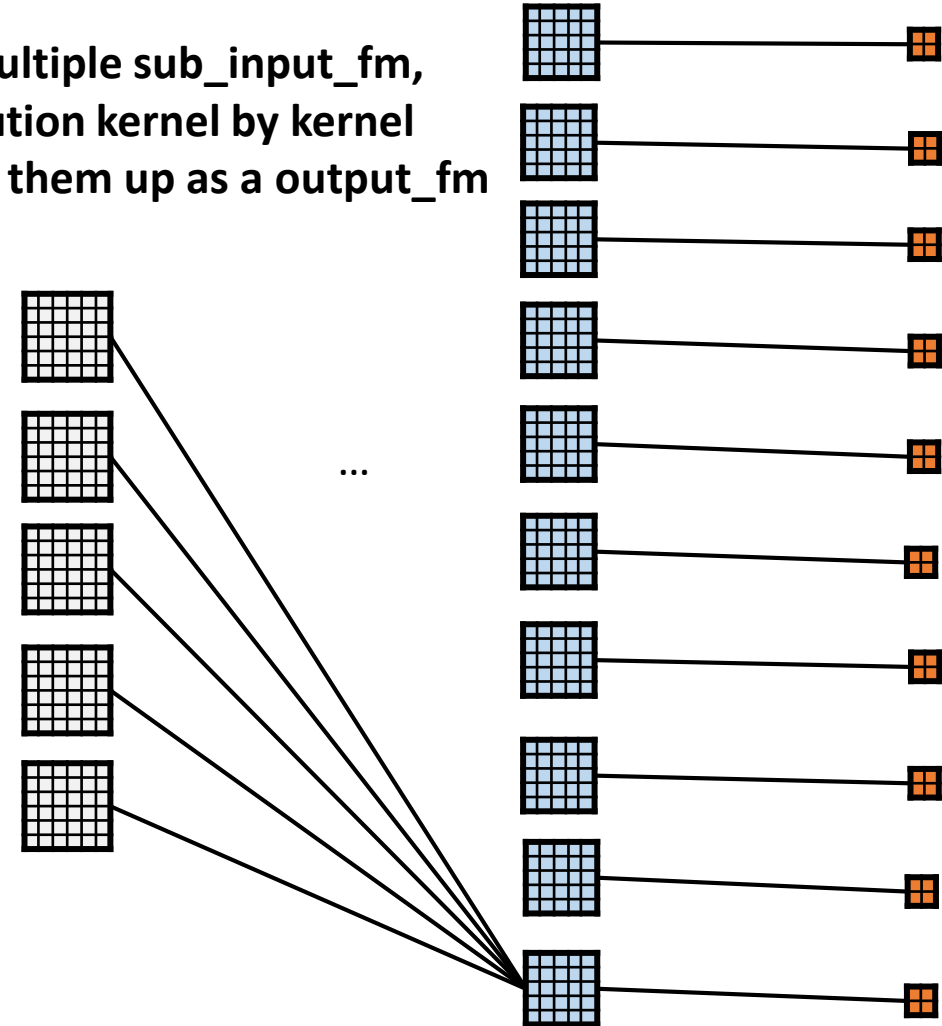


Example: Convolution Layer Case

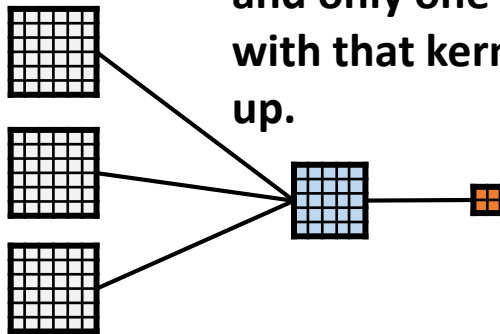
If there is one sub_input_fm,
do convolution with each kernel.



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

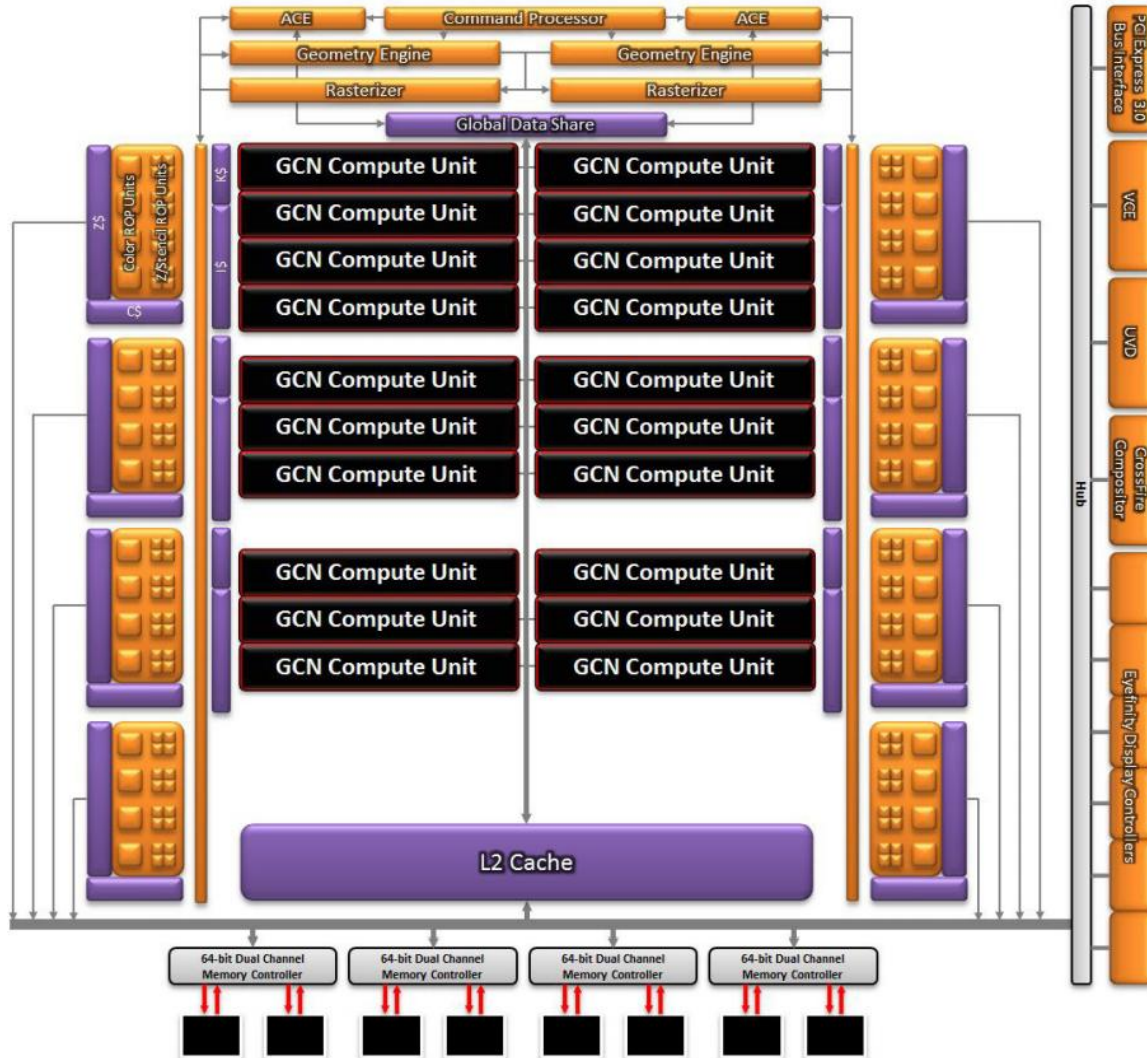
- **Data type:**
 - Every element of images is an integer from 0 to 9.
 - Every element of kernels is an integer from 0 to 9.
 - Every element of weights is an integer from 0 to 9.
 - Above three are stored into an 1D array by row-major, individually.
 - Number of image: 1 ~ 20
 - Image size: 16x16 ~ 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

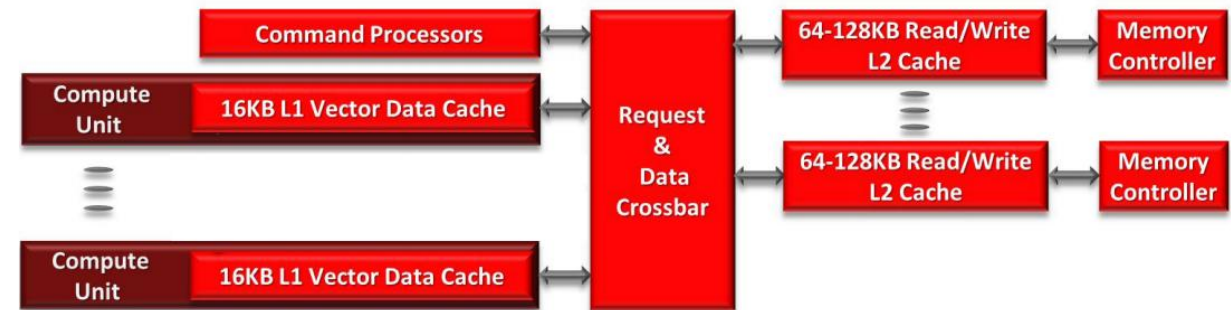


Figure from: https://www.amd.com/Documents/GCN_Architecture_whitepaper.pdf

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 1
 - CNN.cl * Write your code here
 - parameter.h 1

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
```

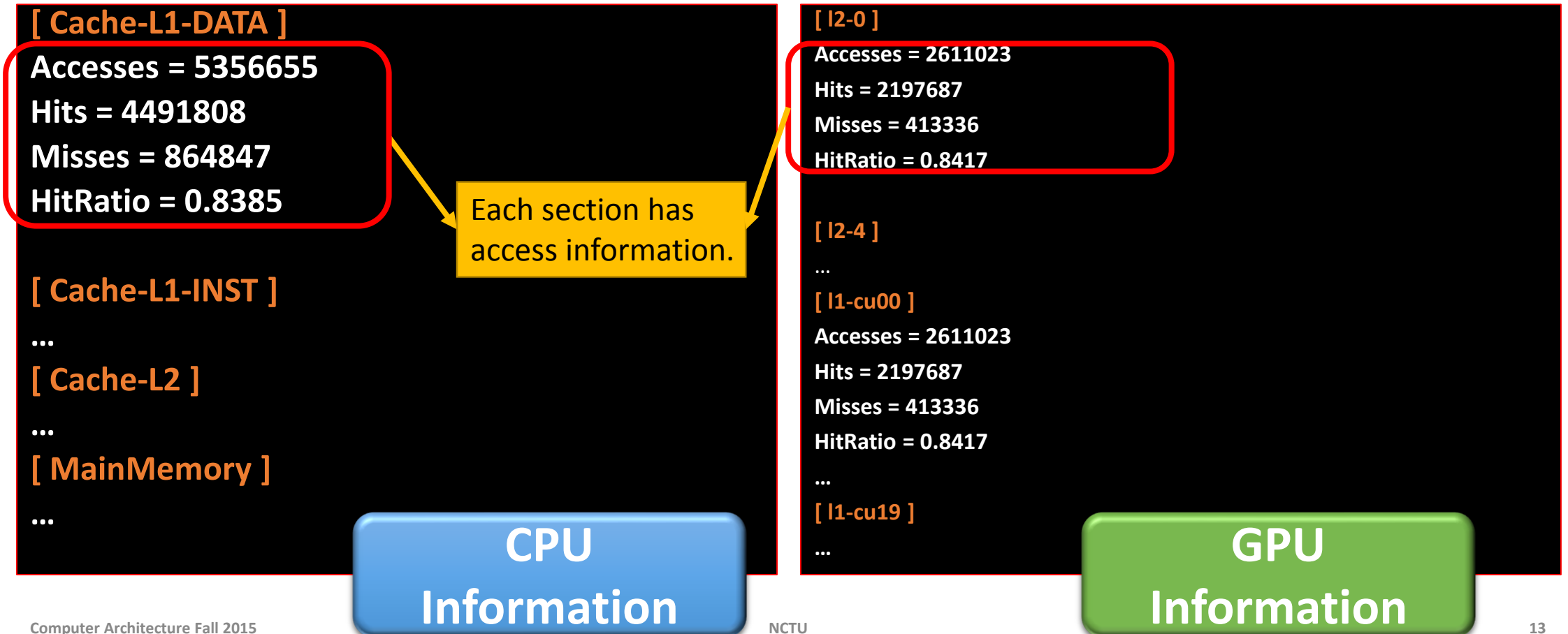
“Cycles” is used to measure the performance!! Remember to record it!!!

Each CU has some other information.

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

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

(due 1/14)



Note: report do not exceed 4 pages.

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