
[SoC Design: Term Project]
Hardware accelerator for CNN

Chester Sungchung Park
SoC Design Lab, Konkuk University
Webpage: <http://soclab.konkuk.ac.kr>

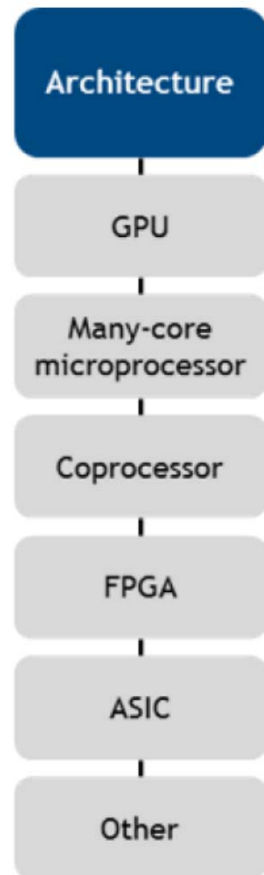
Teaching Assistants

- ❑ Joocho Wang (joohowang@konkuk.ac.kr), Ph.D. candidate
- ❑ Sunwoo Kim (sunwkim@konkuk.ac.kr), Ph.D. candidate

Outline

- ☐ Hardware accelerator
- ☐ CNN for MNIST
- ☐ Design flow
- ☐ Design constraints
- ☐ Evaluation
- ☐ Submission
- ☐ Presentation
- ☐ Appendix

Hardware Accelerator

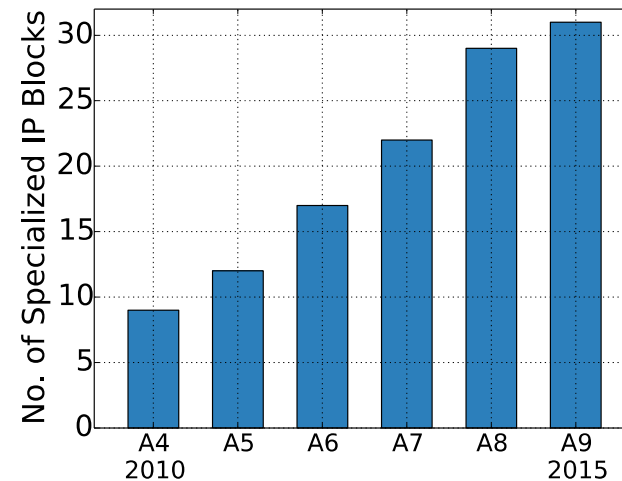
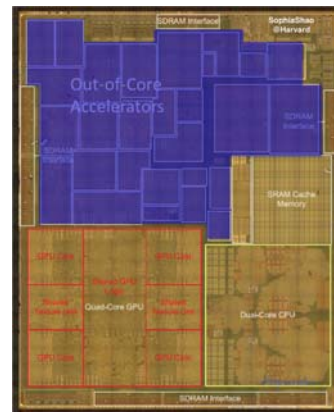


... accelerate applications and workloads by offloading a portion of the processing onto adjacent silicon subsystems such as **graphics processing units (GPUs)** and **field-programmable gate arrays (FPGAs)**.

Hyper-Moore's Law (Nvidia)

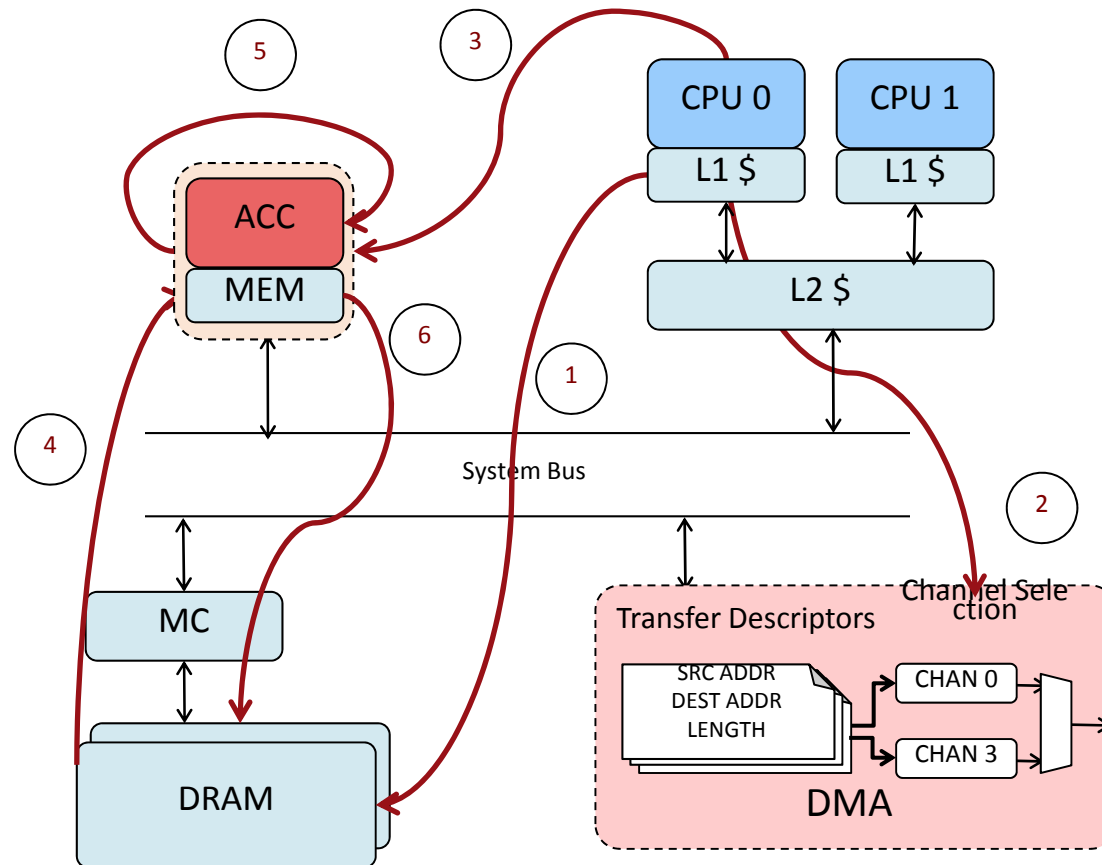
We are just on the precipice of **a new Moore's Law**
– one that is driven by traditional CPUs with **accelerator** kickers

Apple's AP

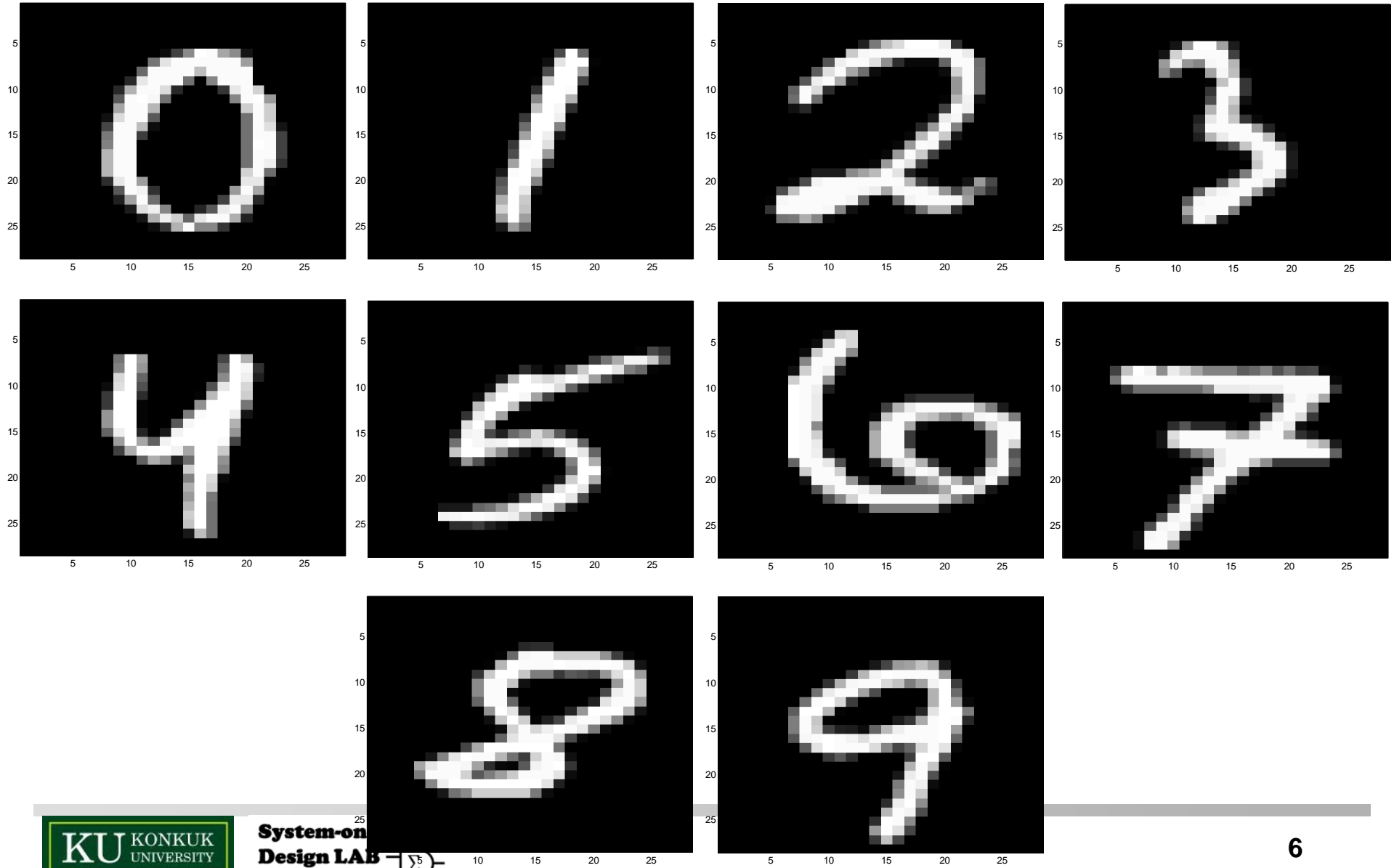


Hardware Accelerator

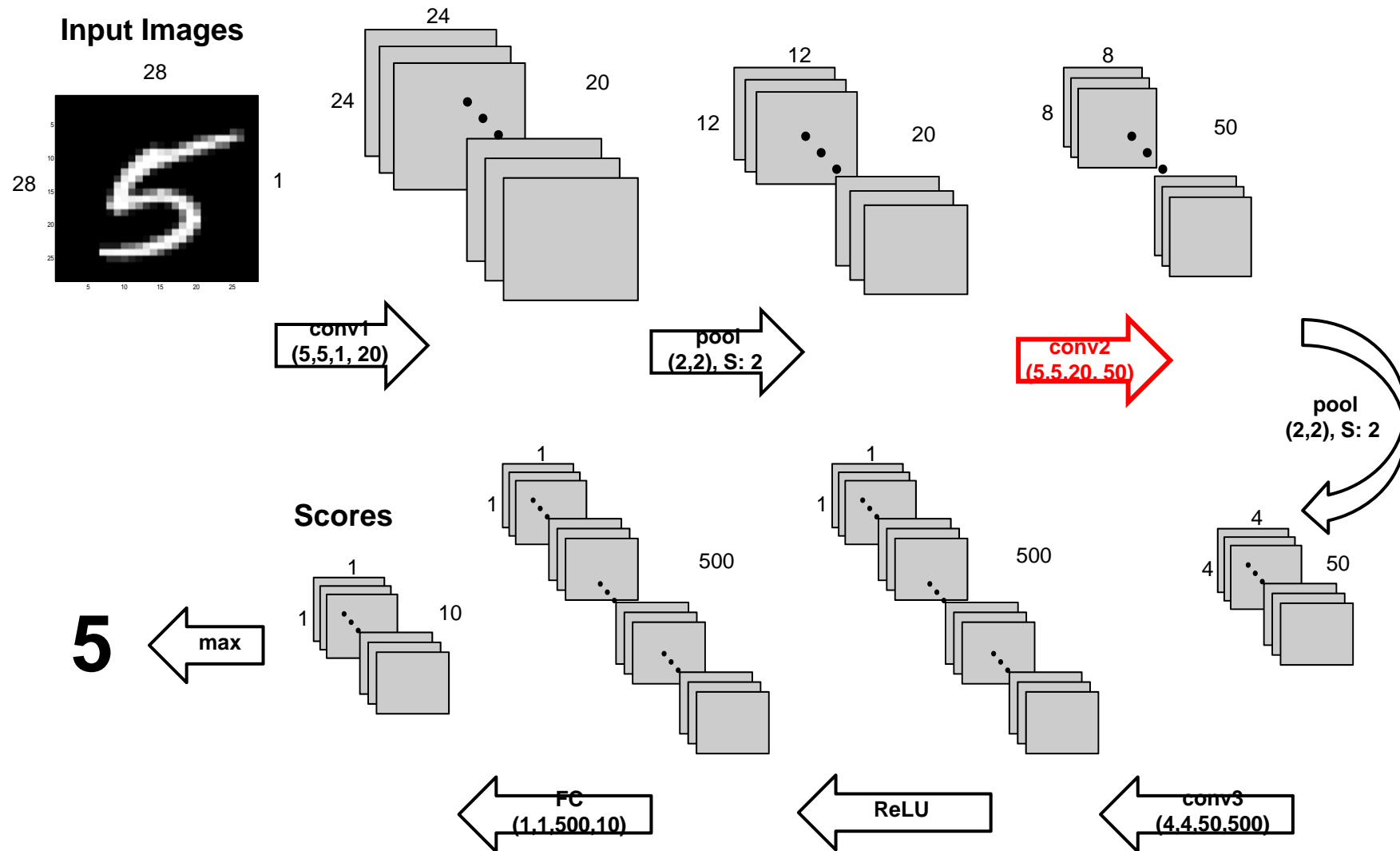
❑ SoC integration



MNIST (Input Images)



CNN for MNIST



CNN for MNIST

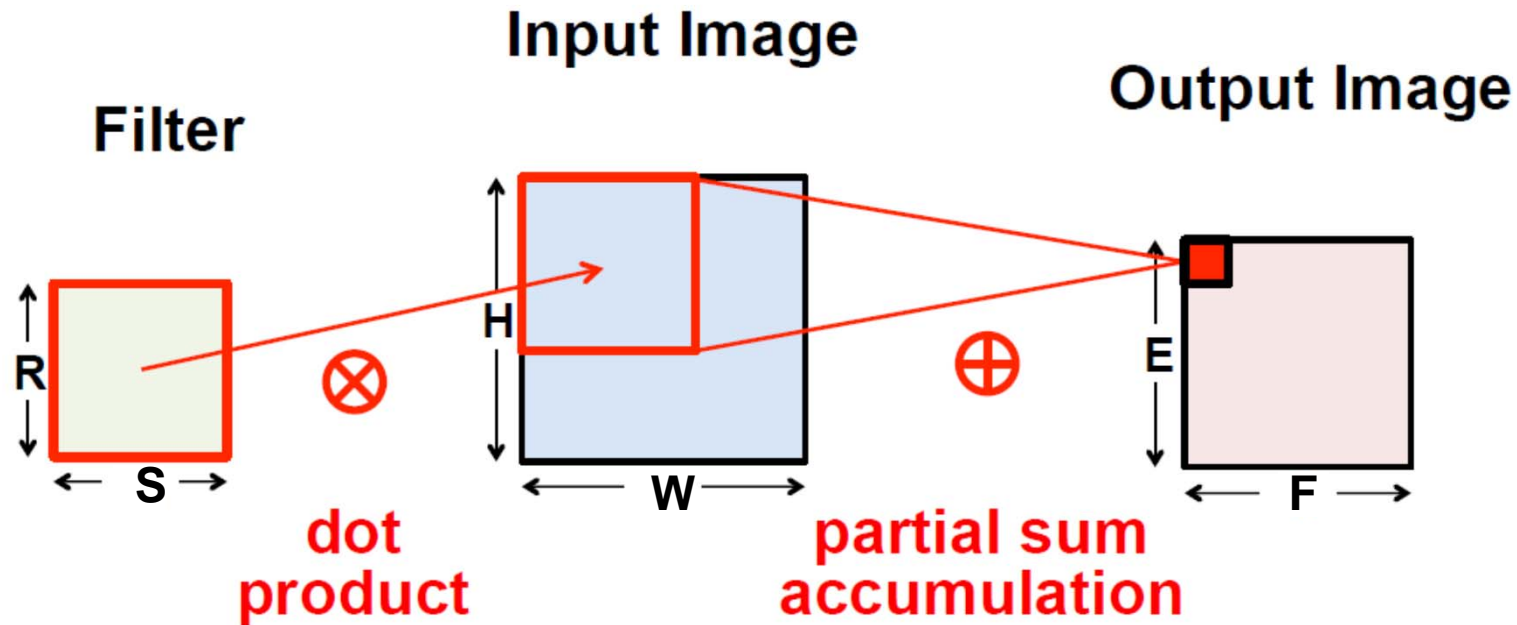
❑ Full-SW implementation (Jooho Wang)

- Set the stack and heap size as explained in the appendix
- Run the main program and check the results in the Console window



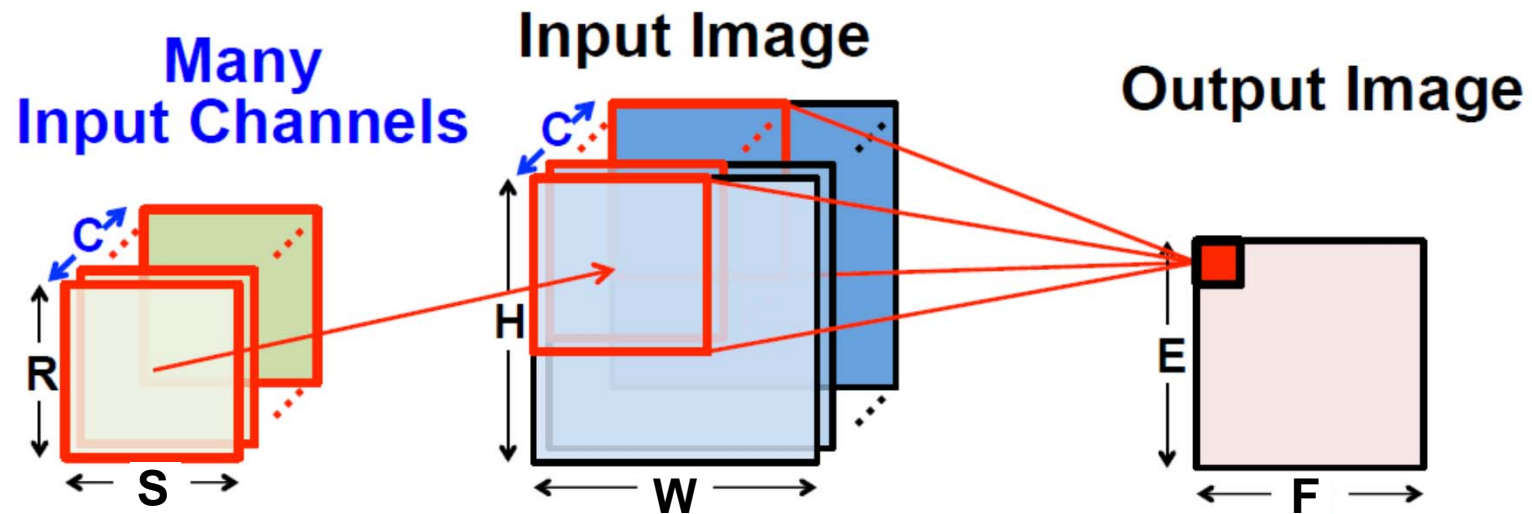
CNN_MNIST_SW.zip

Convolution Layer in CNN



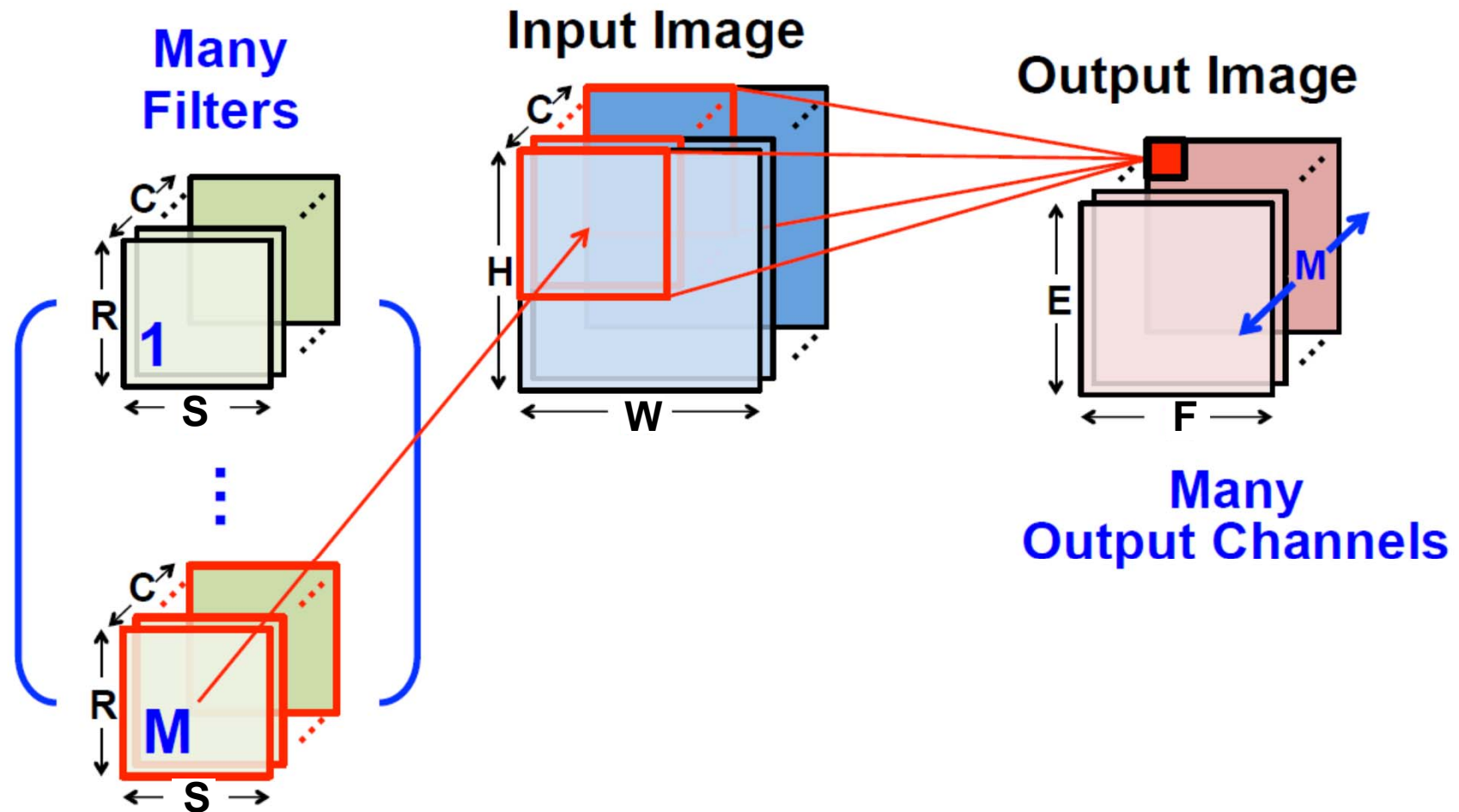
Source: Y-H. Chen

Convolution Layer in CNN



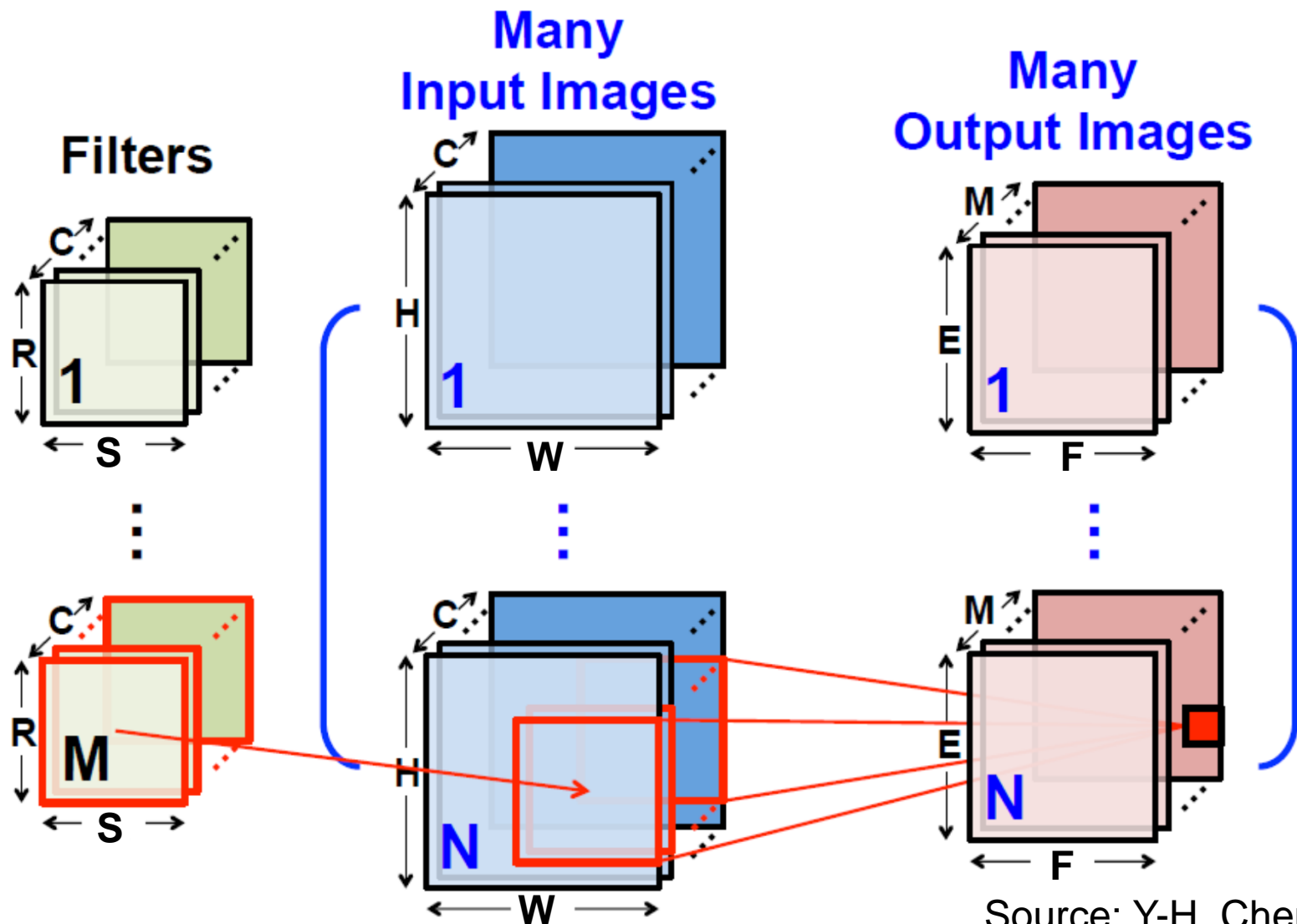
Source: Y-H. Chen

Convolution Layer in CNN



Source: Y-H. Chen

Convolution Layer in CNN



Source: Y-H. Chen

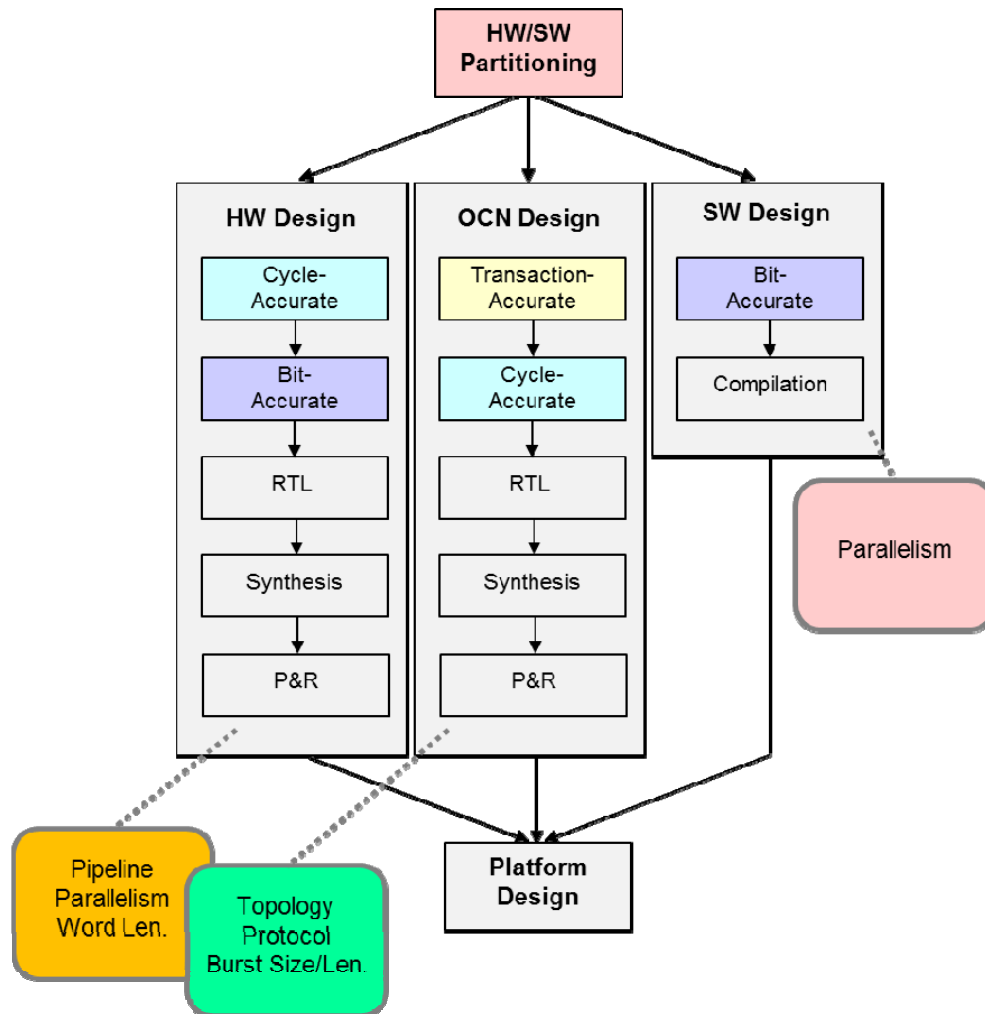
Convolution Layer in CNN

□ Pseudo code

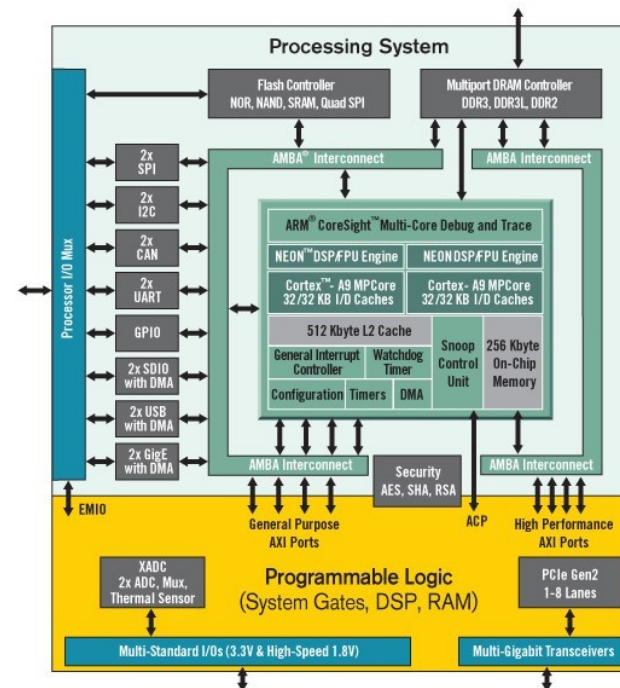
```
For (n = 0; n < N; n++) → Batch
  For (c = 0; c < C; c++) → Channel
    For (m = 0; m < M; m++) → Filter
      For (f = 0; f < F; f++) → of height
        For (e = 0; e < E; e++) → of width
          For (s = 0; s < S; s++) → f height
            For (r = 0; r < R; r++) → f width
              of[e][f][m][n] += if[r + e][s + f][c][n] · f[r][s][c][m]
```

Feature map out Feature map in Filter

Design Flow



ZYNQ7020



Design Constraints

- ❑ Modify **only** the convolution part of Layer 2 (**convolution2_hw**) in the main program
 - Implement the **hardware** accelerator to which you can offload **all** the arithmetic operations
 - Do not modify any other part in the main program

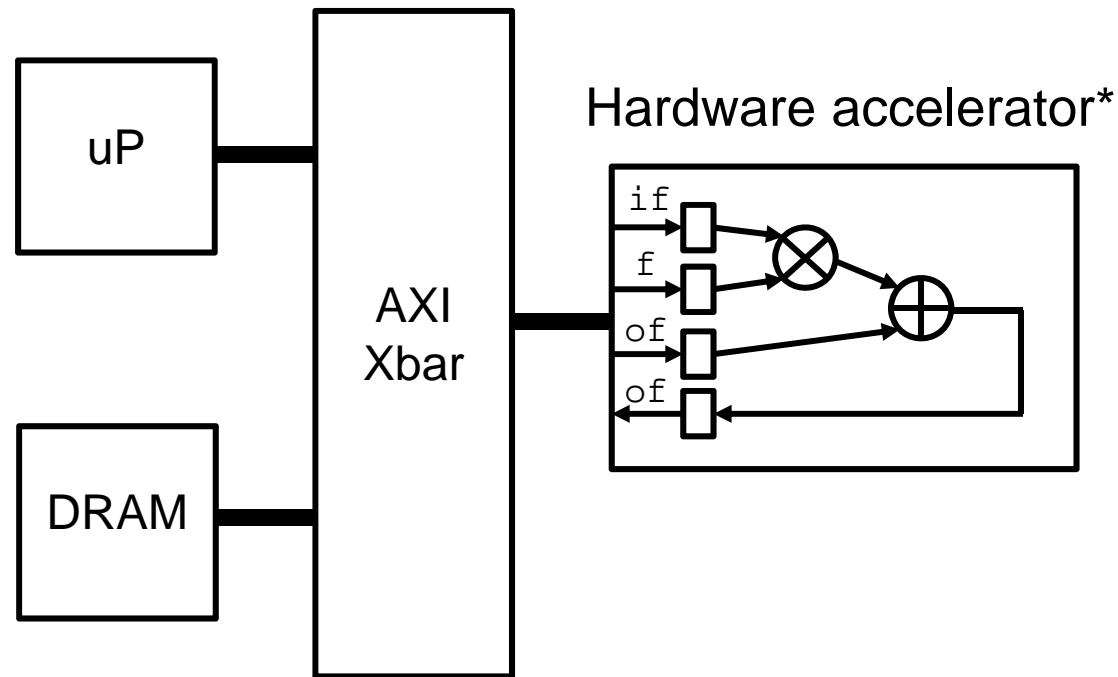
```
void convolution2_hw(float *ofmap, float *ifmap,... )
{
    int n = 0, c = 0, m = 0, f = 0, e = 0, r = 0, s = 0;

    for(n = 0; n < N; n++)
        for(c = 0; c < C; c++)
            for(m = 0; m < M; m++)
                for (f = 0; f < F; f++)
                    for (e = 0; e < E; e++)
                        for (r = 0; r < R; r++)
                            for (s = 0; s < S; s++)
                                ofmap[ ((n*M+m)*E+e)*F+f] += ifmap...;
}
```

You are allowed to modify only this part!

Design Constraints

- ❑ Example of hardware accelerator



* Single MAC hardware block assumed (no parallelism)

Evaluation

- ☐ Submission completeness
 - Reproducibility (10pt)
- ☐ Accuracy
 - Classification error (20pt)
 - Quantization error (10pt)
- ☐ Execution time
 - Convolution in Layer 2 (30pt)
- ☐ Novelty
 - Any good ideas (10pt)

Reproducibility

- ❑ Make sure that your submission is complete
 - In other words, it is possible to *reproduce* your design together with the results (accuracy and performance) using only the submitted files

Accuracy

❑ Run the main program and check the accuracy of `convolution2_hw` in the Console window

- Classification error
 - ✓ A new set of 10 test images will **not** be given in advance
- Quantization error
 - ✓ NSR measures the difference from the reference results (e.g., quantization noise)

```
<Reference design>
estimated label: 0 (0.1069348)
estimated label: 1 (0.0899757)
estimated label: 2 (0.1065318)
estimated label: 3 (0.0880197)
estimated label: 4 (0.0886517)
estimated label: 5 (0.1049255)
estimated label: 6 (0.0922938)
estimated label: 7 (0.1232479)
estimated label: 8 (0.1233093)
estimated label: 9 (0.1174142)
Average time (usec): 258928.7757874
Total time (usec): 375775.9928703
```

```
<HW-based design>
estimated label: 0 (0.1069348)
estimated label: 1 (0.0899757)
estimated label: 2 (0.1065318)
estimated label: 3 (0.0880197)
estimated label: 4 (0.0886517)
estimated label: 5 (0.1049255)
estimated label: 6 (0.0922938)
estimated label: 7 (0.1232479)
estimated label: 8 (0.1233093)
estimated label: 9 (0.1174142)
Average time (usec): 258929.1334152
Total time (usec): 375774.6517658
```

```
Measure performance: NSR(dB) = -inf
```

Performance

- ❑ Run the main program and check the performance of `convolution2_hw` in the Console window
 - Execution time

```
<Reference design>
estimated label: 0 (0.1069348)
estimated label: 1 (0.0899757)
estimated label: 2 (0.1065318)
estimated label: 3 (0.0880197)
estimated label: 4 (0.0886517)
estimated label: 5 (0.1049255)
estimated label: 6 (0.0922938)
estimated label: 7 (0.1232479)
estimated label: 8 (0.1233093)
estimated label: 9 (0.1174142)
Average time (usec): 258928.7757874
Total time (usec): 375775.9928703
```

```
<HW-based design>
estimated label: 0 (0.1069348)
estimated label: 1 (0.0899757)
estimated label: 2 (0.1065318)
estimated label: 3 (0.0880197)
estimated label: 4 (0.0886517)
estimated label: 5 (0.1049255)
estimated label: 6 (0.0922938)
estimated label: 7 (0.1232479)
estimated label: 8 (0.1233093)
estimated label: 9 (0.1174142)
Average time (usec): 258929.1334152
Total time (usec): 375774.6517658
```

Measure performance: NSR(dB) = -inf

Any Good Ideas

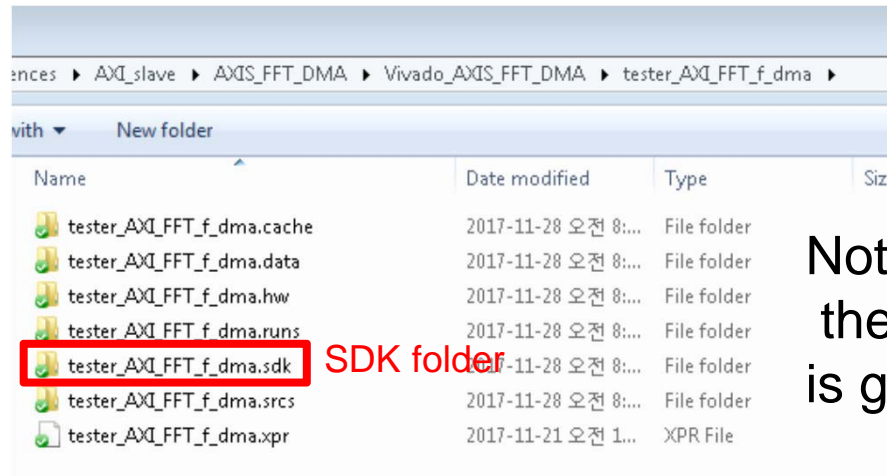
- ❑ Anything that can reduce the execution time such as
 - DMA-based data transfer
 - Array of multiple MAC hardware blocks
 - Row-stationary dataflow
- ❑ For the state-of-the-art hardware accelerator for CNN, refer to the following paper (and those citing it – available at <http://ieeexplore.ieee.org>):
 - *Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks*, IEEE JSSC, Jan. 2017

Submission

- ☐ Due date: **Dec. 11 (Mon), 2017**, 23:59:59 GMT+9
- ☐ Zip the following files into a single file and send it to chesterku2013@gmail.com
 - **C/Verilog source/header files** and any other files that are needed to *reproduce* your design
 - Copy of the entire **SDK folder** in your project (including the bitstream)
 - **Slideset file** (PPT) including the results in the Console window
- ☐ You can post a question in the Q&A of the course page (<https://www.sites.google.com/site/kusocdesignlab/q-a-2>)
- ☐ **No delay will be acceptable!**

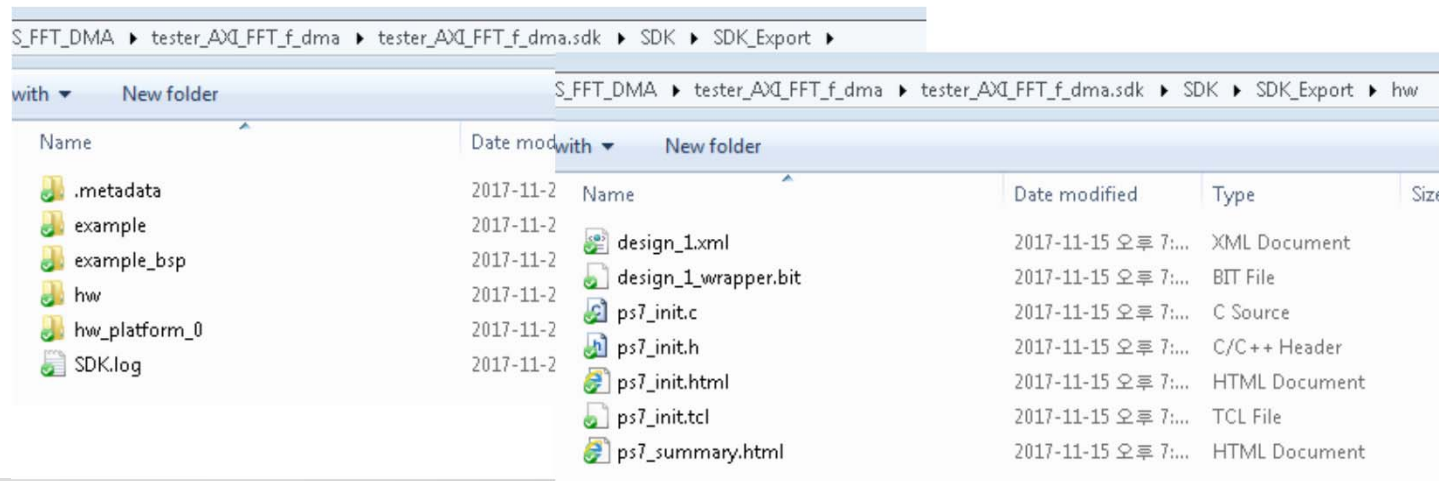
Submission

❑ SDK folder in your project



Name	Date modified	Type	Size
tester_AXI_FFT_f_dma.cache	2017-11-28 오전 8:...	File folder	
tester_AXI_FFT_f_dma.data	2017-11-28 오전 8:...	File folder	
tester_AXI_FFT_f_dma.hw	2017-11-28 오전 8:...	File folder	
tester_AXI_FFT_f_dma.runs	2017-11-28 오전 8:...	File folder	
tester_AXI_FFT_f_dma.sdk	2017-11-28 오전 8:...	File folder	
tester_AXI_FFT_f_dma.srds	2017-11-28 오전 8:...	File folder	
tester_AXI_FFT_f_dma.xpr	2017-11-21 오전 1:...	XPR File	

Note that the name of the SDK folder generally is given as “[project name].sdk”



Name	Date modified	Type	Size
.metadata	2017-11-2		
example	2017-11-2		
example_bsp	2017-11-2		
hw	2017-11-2		
hw_platform_0	2017-11-2		
SDK.log	2017-11-2		

Name	Date modified	Type	Size
design_1.xml	2017-11-15 오후 7:...	XML Document	
design_1_wrapper.bit	2017-11-15 오후 7:...	BIT File	
ps7_init.c	2017-11-15 오후 7:...	C Source	
ps7_init.h	2017-11-15 오후 7:...	C/C++ Header	
ps7_init.html	2017-11-15 오후 7:...	HTML Document	
ps7_init.tcl	2017-11-15 오후 7:...	TCL File	
ps7_summary.html	2017-11-15 오후 7:...	HTML Document	

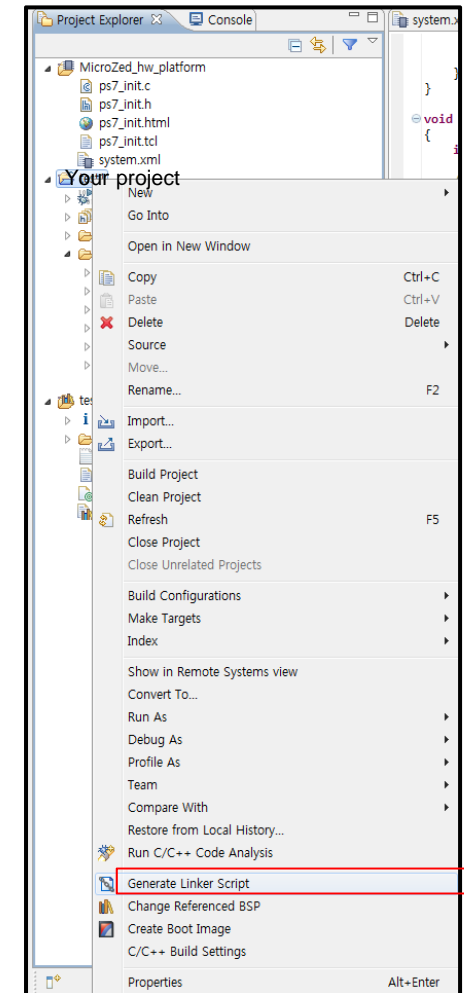
Presentation & Demo

- Dec. 12 (Tue) 10:30~12:00 / 14 (Thu) 09:00~10:30, 2017, New Eng. Bldg #1113
 - Team-presentation with exactly the same **slideset file** as submitted on Dec. 11
 - Team-demo with exactly the same **SDK folder** as submitted on Dec. 11
- 1. *Bring a storage device (e.g., HDD) having the entire project folder just in case (e.g., when the SDK folder does not work)*
- 2. *Note that any progress made later than **Dec. 11** **cannot** be counted for evaluation*

Appendix

Generating a linker script

- ❑ Select the application project in the **Project Explorer** or **C/C++ Projects** view
- ❑ Right-click **Generate Linker Script** or click **Xilinx Tools > Generate Linker script**.



Generating a linker script

- ❑ Set both the heap and stack sizes in the **Basic** tab to **104857600** as shown below

