[SoC Design – Term Project]

Accelerator for CNN

Chester Sungchung Park

SoC Design Lab, Konkuk University

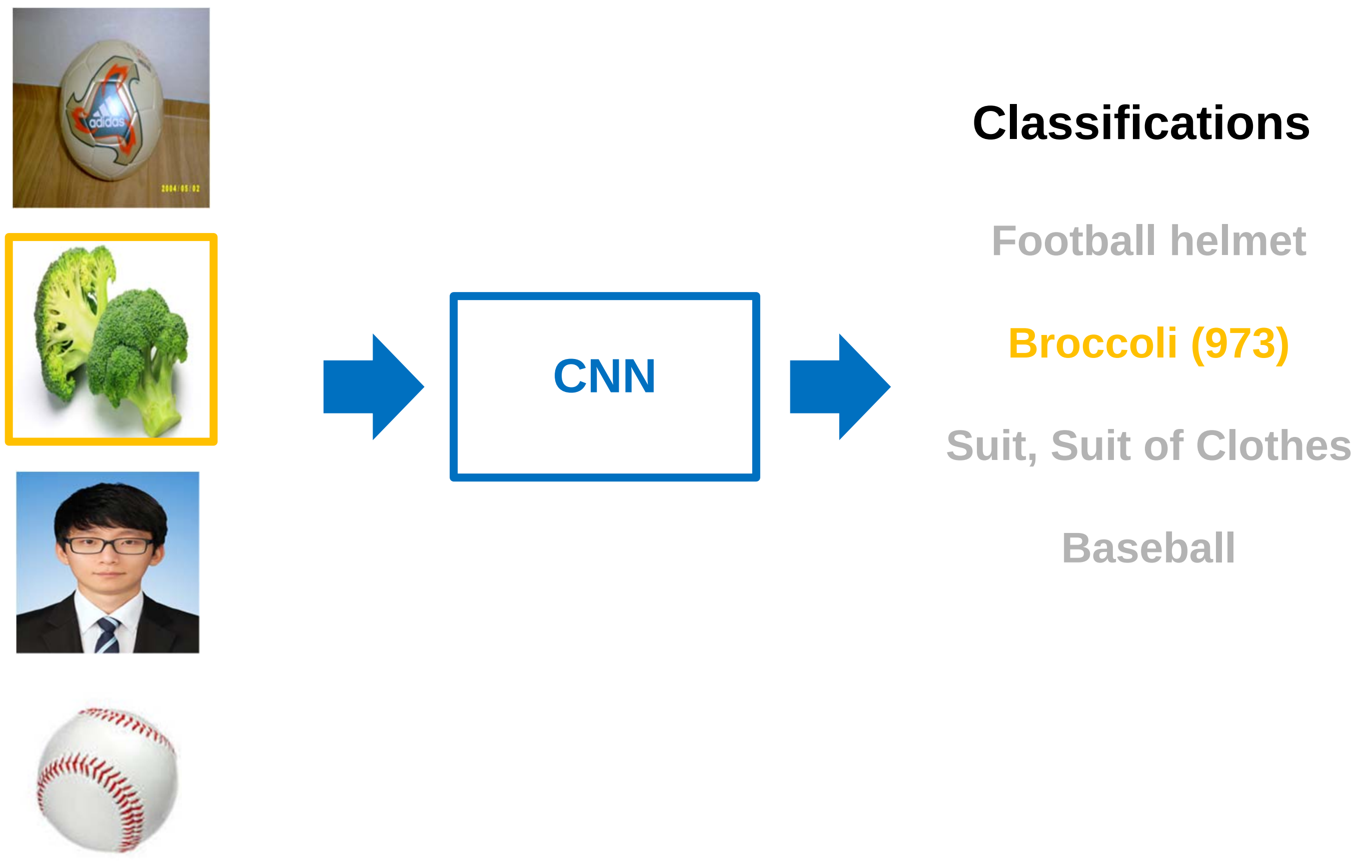
Webpage: http://soclab.konkuk.ac.kr

Outline

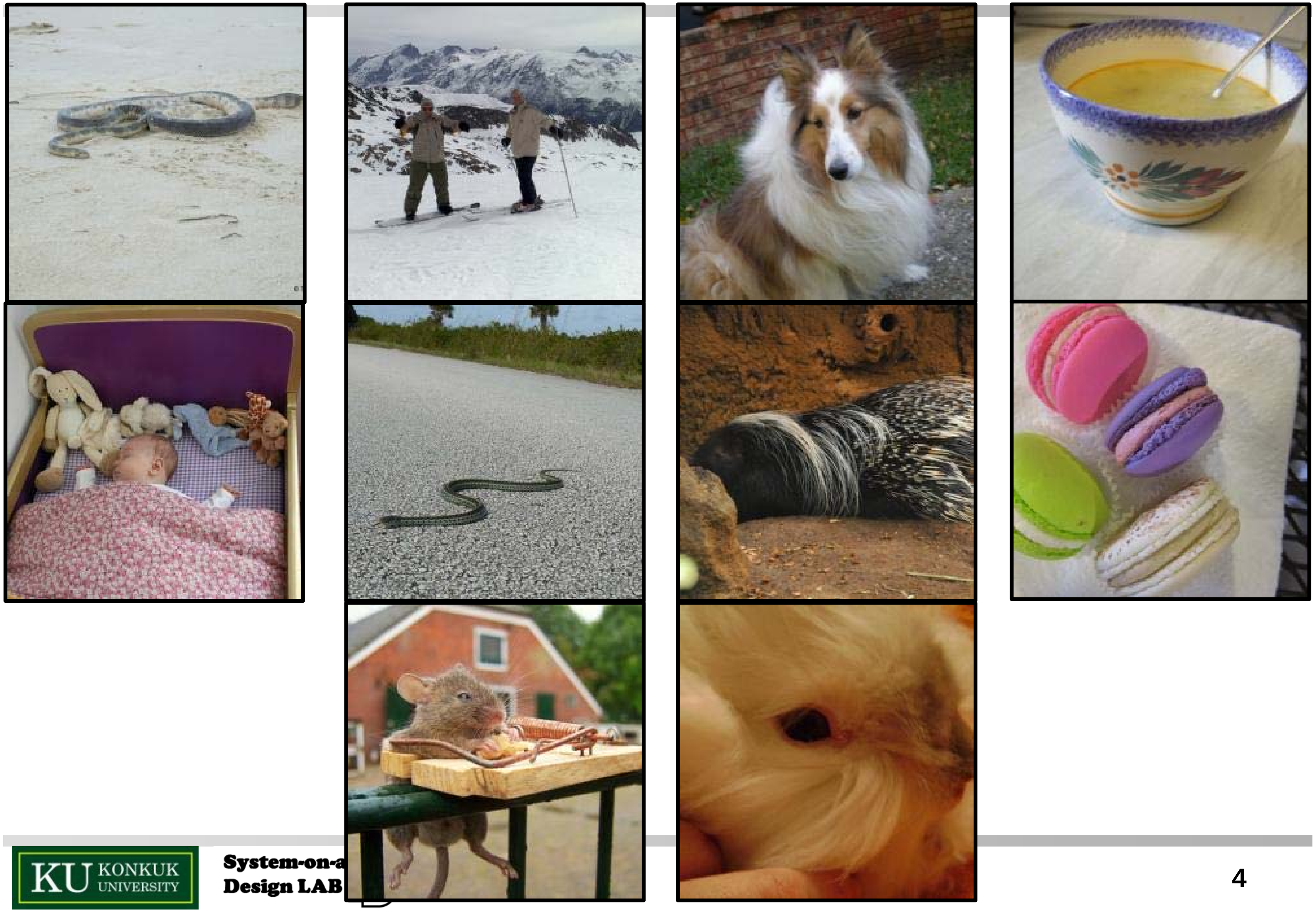
- ❑ Hardware accelerator
- ❑ Convolution in CNN
    - AlexNet for ImageNet
- ❑ Design flow
- ❑ Design constraints
- ❑ Evaluation
- ❑ Submission
- ❑ Presentation
- ❑ Appendix

Convolutional Neural Networks (CNN)

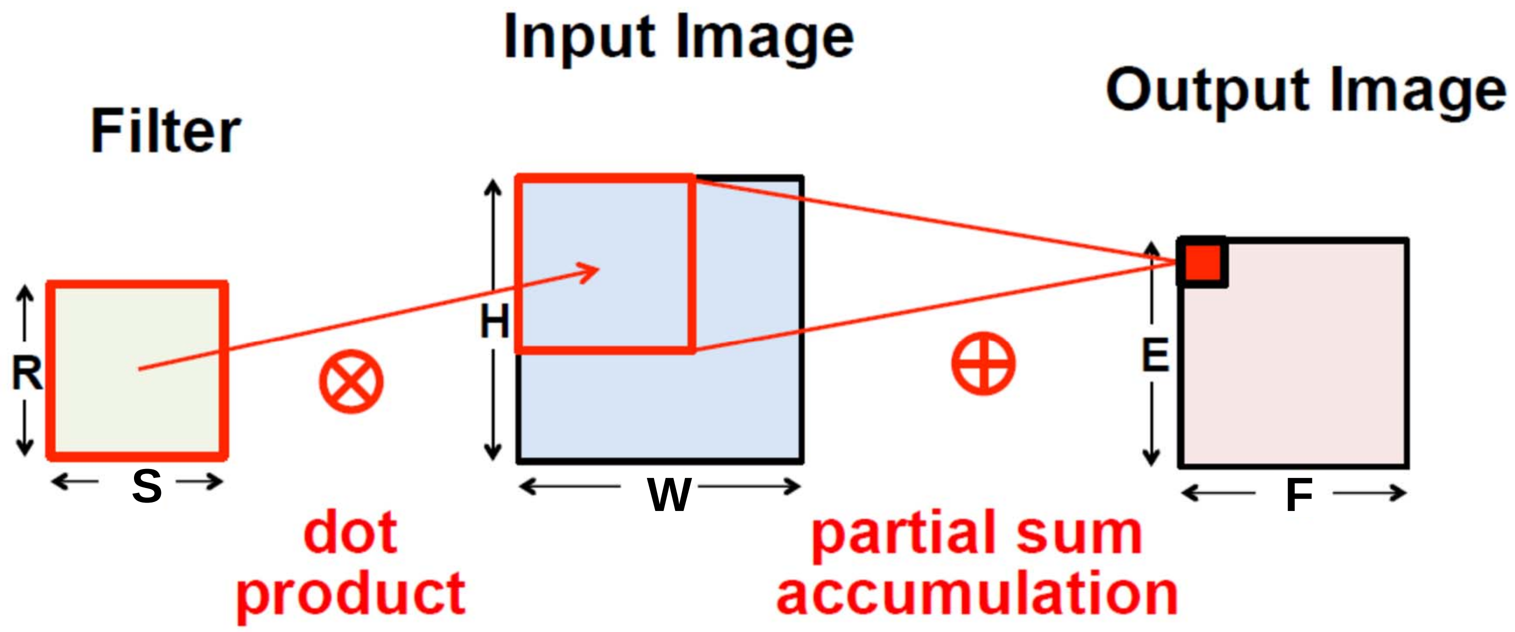
- ❑ Example: AlexNet for ImageNet



ImageNet (Input Images)

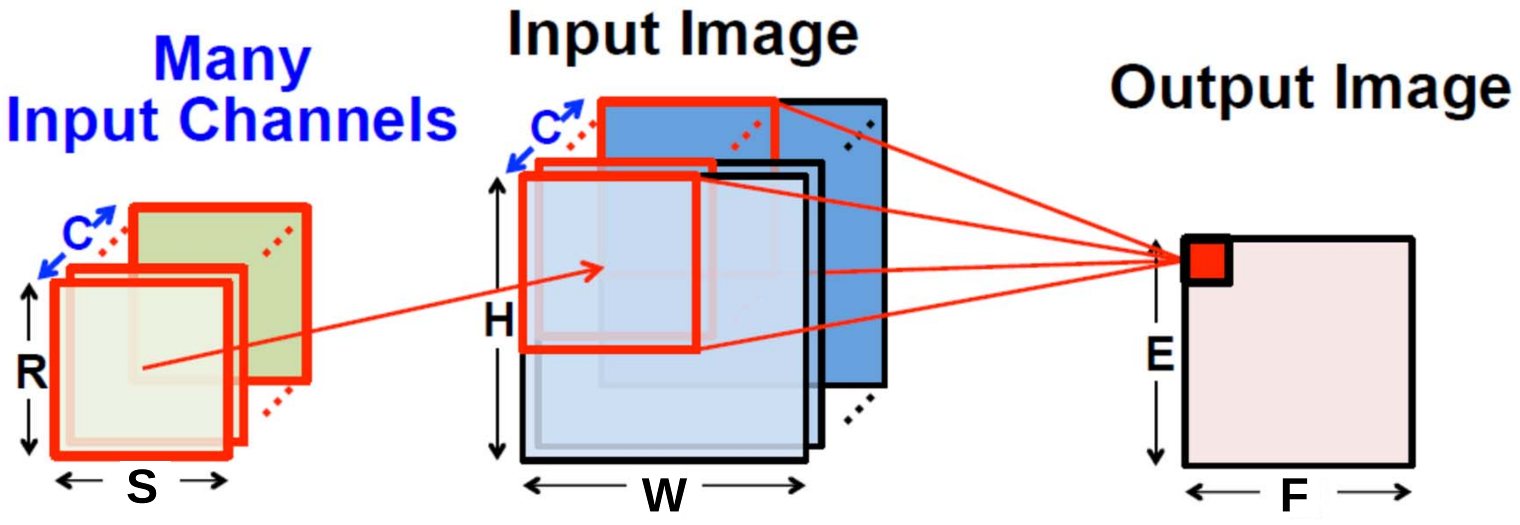


Convolution in CNN



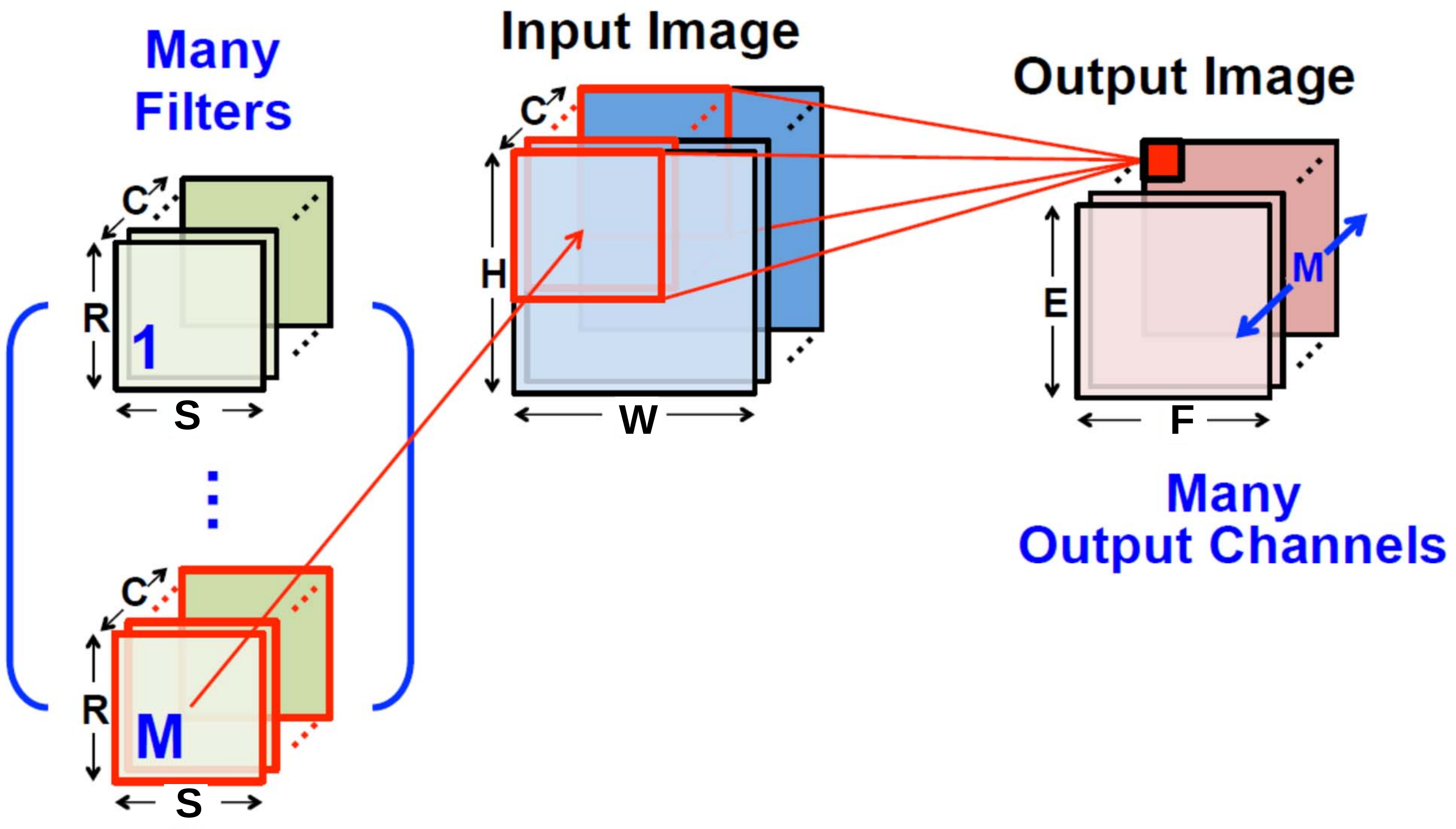
Source: Y-H. Chen

Convolution in CNN



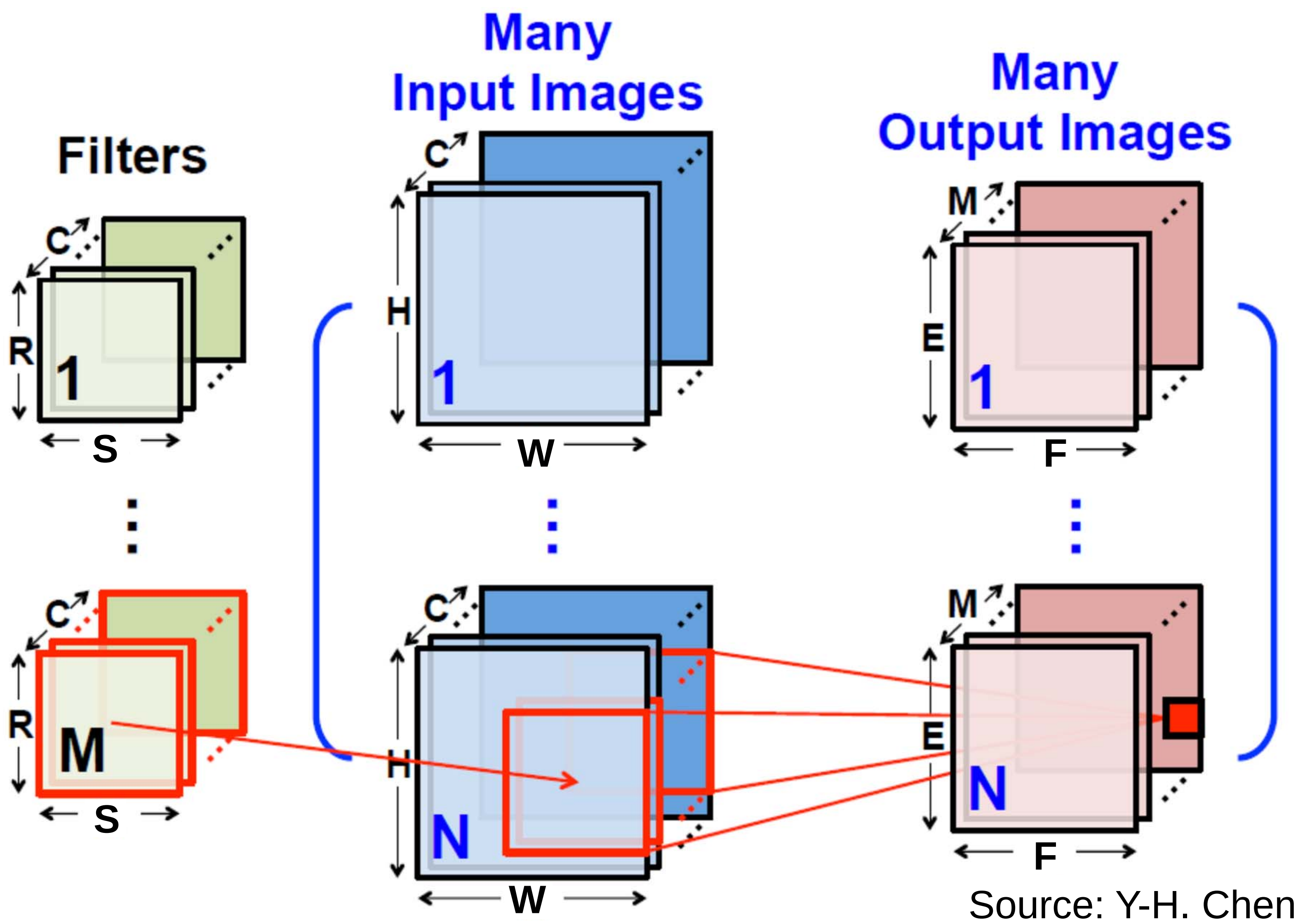
Source: Y-H. Chen

Convolution in CNN



Source: Y-H. Chen

Convolution in CNN



Source: Y-H. Chen

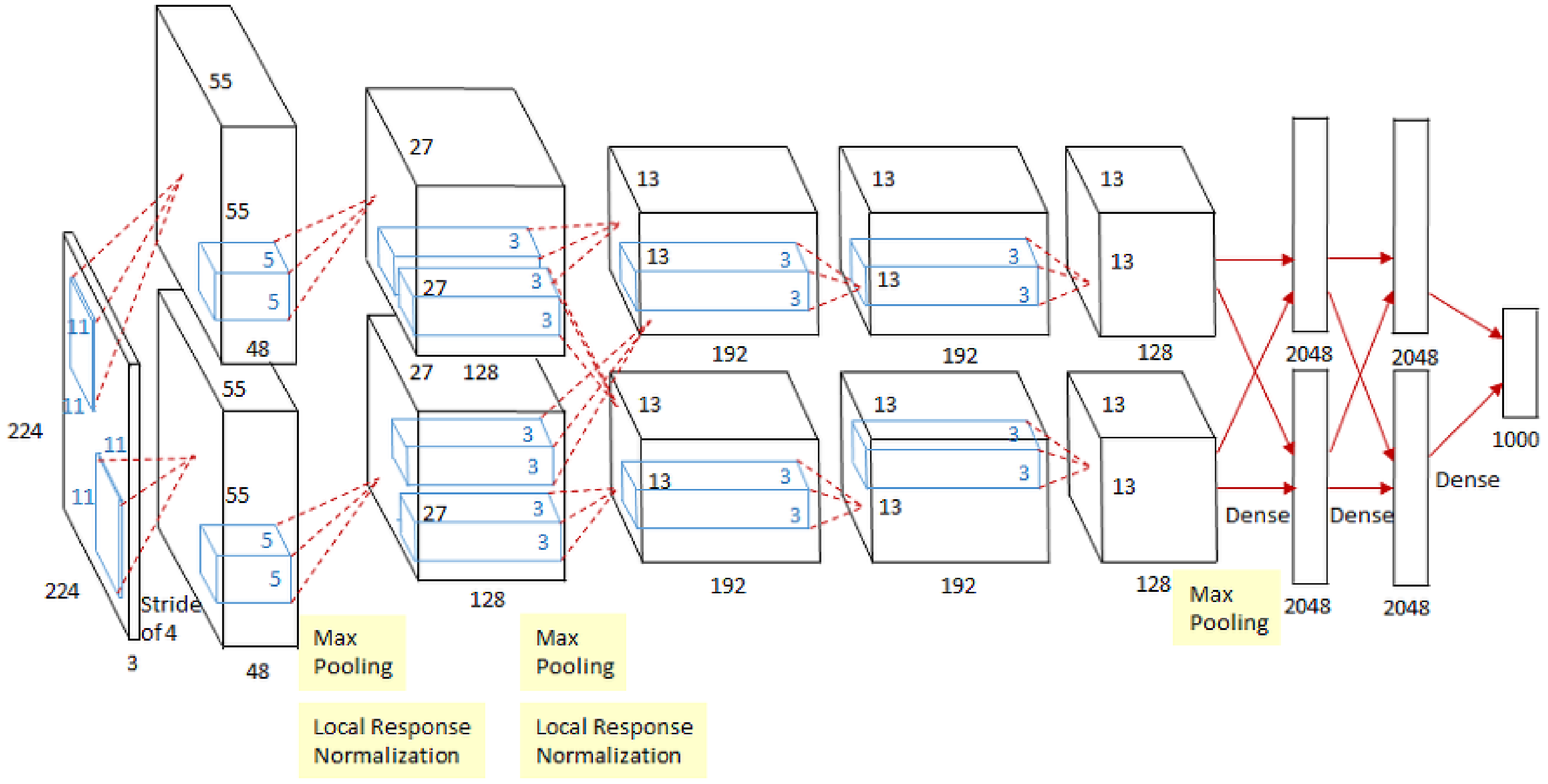
Convolution in CNN

- Pseudo code

```
For $(n = 0; n < N; n++)$ $\rightarrow$ Batch
  For $(c = 0; c < C; c++)$ $\rightarrow$ Channel
    For $(m = 0; m < M; m++)$ $\rightarrow$ Filter
      For $(f = 0; f < F; f++)$ $\rightarrow$ $\boldsymbol{of}$ height
        For $(e = 0; e < E; e++)$ $\rightarrow$ $\boldsymbol{of}$ width
          For $(s = 0; s < S; s++)$ $\rightarrow$ $\boldsymbol{f}$ height
            For $(r = 0; r < R; r++)$ $\rightarrow$ $\boldsymbol{f}$ width
              $\underbrace{\boldsymbol{of}}_{\text{Feature map out}}[e][f][m][n] \mathrel{+}= \underbrace{\boldsymbol{if}}_{\text{Feature map in}}[r+e][s+f][c][n] \cdot \underbrace{\boldsymbol{f}}_{\text{Filter}}[r][s][c][m]$
```

Convolution in CNN

❑ AlexNet for ImageNet



Convolution in CNN

❑ AlexNet for ImageNet (cont'd)

		Width (W/S)	**Height (H/R)**	**# I-Ch (C)**	**# O-Ch (M)**
Input Image		227	[redacted]	3	-
Filter	Layer 1	11	[redacted]	3	96
	Layer 2	5	[redacted]	48	256
	Layer 3	3	3	256	384
	Layer 4	3	3	192	384
	Layer 5	3	3	192	256
	Layer 6	6	6	256	4096
	Layer 7	1	1	4096	4096
	Layer 8	1	1	4096	1000

Reference Implementation

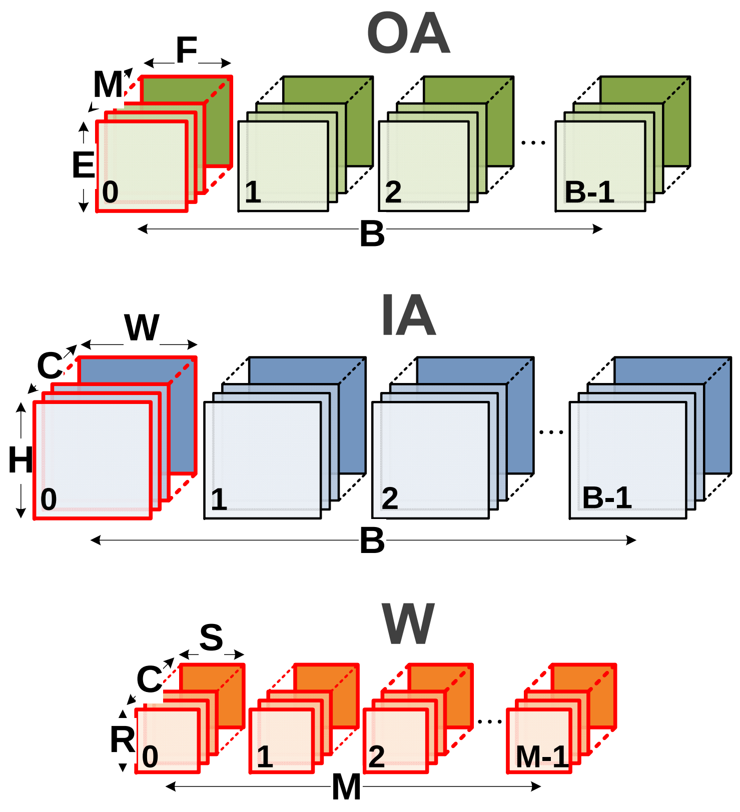
- ❑ Run the attached reference implementation as explained in the appendix
  - AlexNet with layer 3 in HW and the others in SW
  - No local reuse (NLR) chosen as dataflow

source_TP_[redacted]

- ❑ For details of NLR, refer to the following paper
  - “Optimizing FPGA-based Accelerator Design for Deep Convolutional Neural”, C. Zhang et al., FPGA 2015

No Local Reuse (NLR)

- Original algorithm

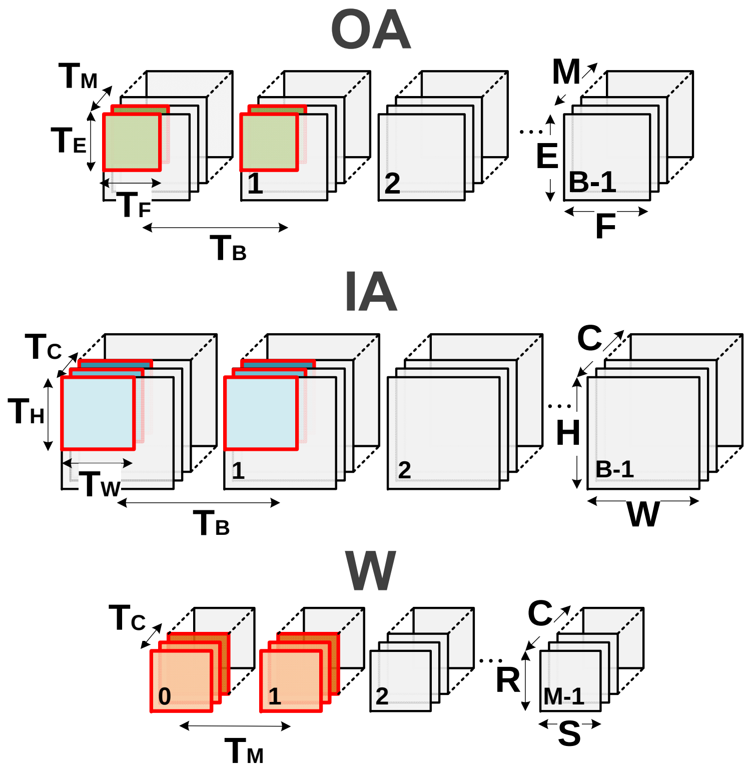


```
for (b = 0; b < B; b++) {
 for (e = 0; e < E; e++) {
  for (f = 0; f < F; f++) {
   for (m = 0; m < M; m++) {
    for (c = 0; c < C; c++) {
     for (r = 0; r < R; r++) {
      for (s = 0; s < S; S++) {
        O[b][m][e][f]+=
        W[m][c][r][s]*I[b][c][e+r][f+s];
} } } } } } }
```

B:	No.of images
E:	Output height
F:	Output width
M:	No.of filters
C:	No.of features
R:	Filter height
S:	Filter width

No Local Reuse (NLR)

- Loop tiling for NLR

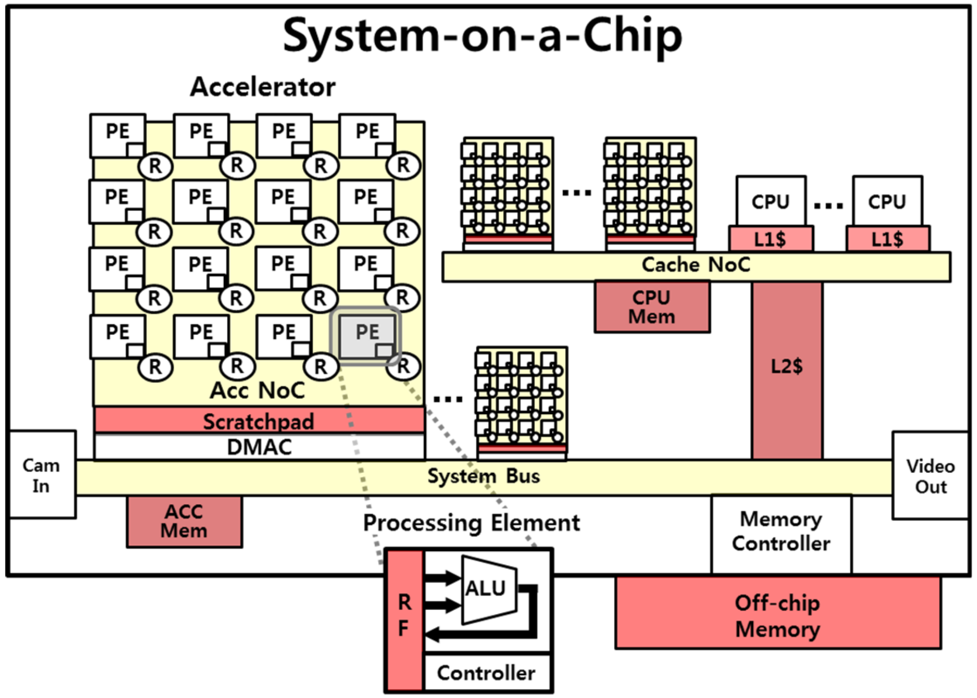


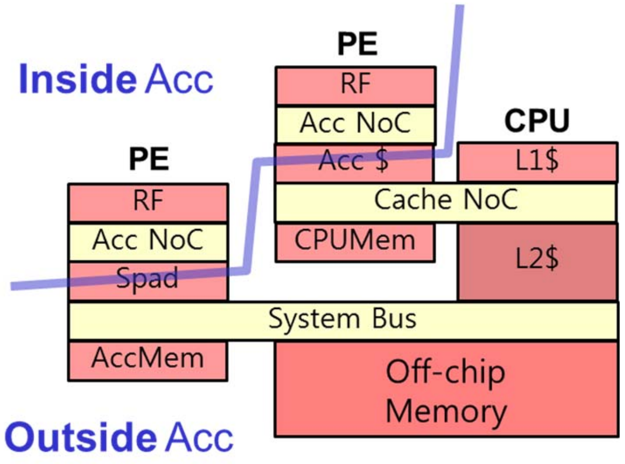
```
for (b = 0; b < B; b += TB) {
 for (e = 0; e < E; e += TE) {
  for (f = 0; f < F; f += TF) {
   for (m = 0; m < M; m += TM) {
    for (c = 0; c < C; c += TC) {
     Ibuf = I[b:b+TB-1][c:c+TC-1][e:e+TE+R-2][f:f+TF+S-2];
     Wbuf = W[m:m+TM-1][c:c+TC-1][0:R-1][0:S-1];
     for (r = 0; r < R; r++) {
      for (s = 0; s < S; s++) {
       for (tb = 0; tb+b < min(B,b+TB); tb++) {
        for (te = 0; te+e < min(E,e+TE); te++) {
         for (tf = 0; tf+f < min(F,f+TF); tf++) {
          for (tm = 0; tm < TM; tm++) {
           for (tc = 0; tc < TC; tc++) {
            wx[tm][tc] = Wbuf[tm][tc][r][s];
            ix[tc] = Ibuf[tb][tc][te+r][tf+s];
            ox[tm] = Obuf[tb][tm][te][tf];
            Obuf[tb][tm][te][tf] = wx[tm][tc]*ix[tc]+ox[tm];
    }}}}}}}}
    O[b:b+TB-1][m:m+TM-1][e:e+TE-1][f:f+TF-1] = Obuf;
}}}}
```

```
TB: Batch size
TE: Output height /tile
TF: Output width /tile
TM: No.of filters /tile
C: No.of features /tile
R: Filter height /tile
S: Filter width /tile
```

No Local Reuse (NLR)

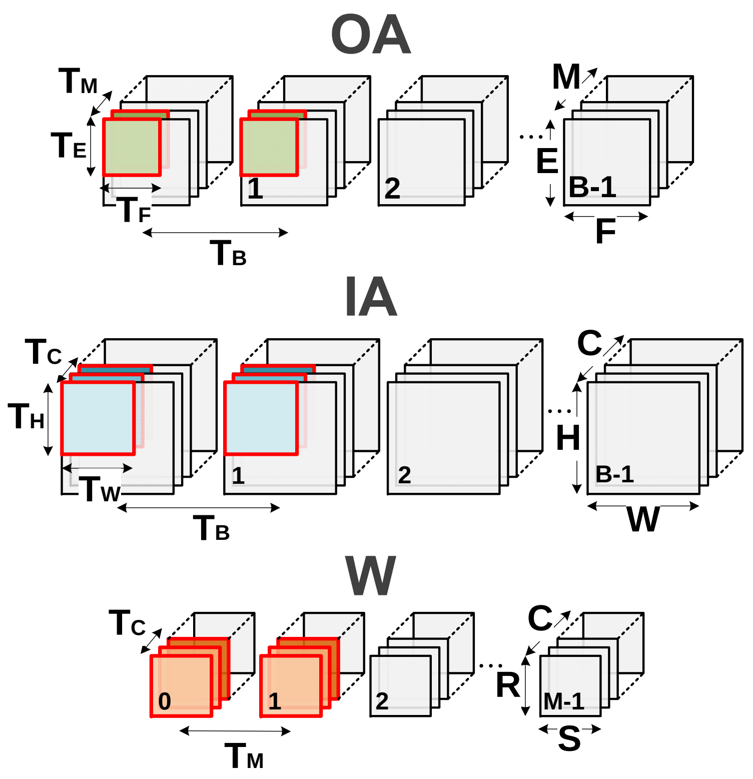
- Memory hierarchy

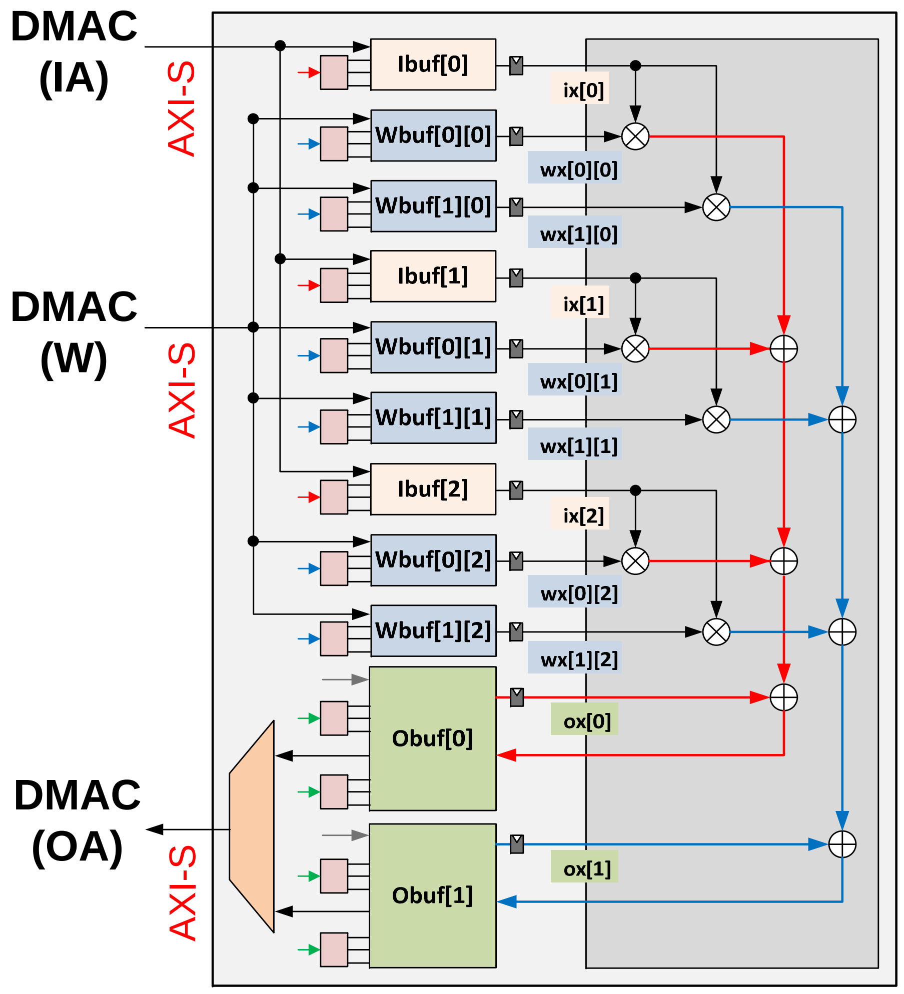




No Local Reuse (NLR)

❑ Accelerator hardware



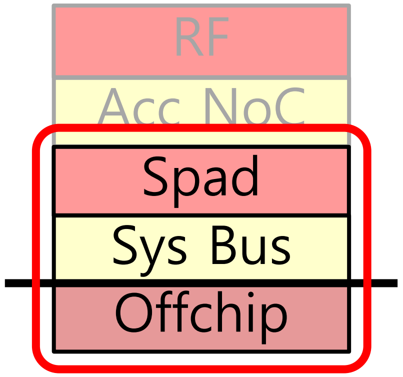


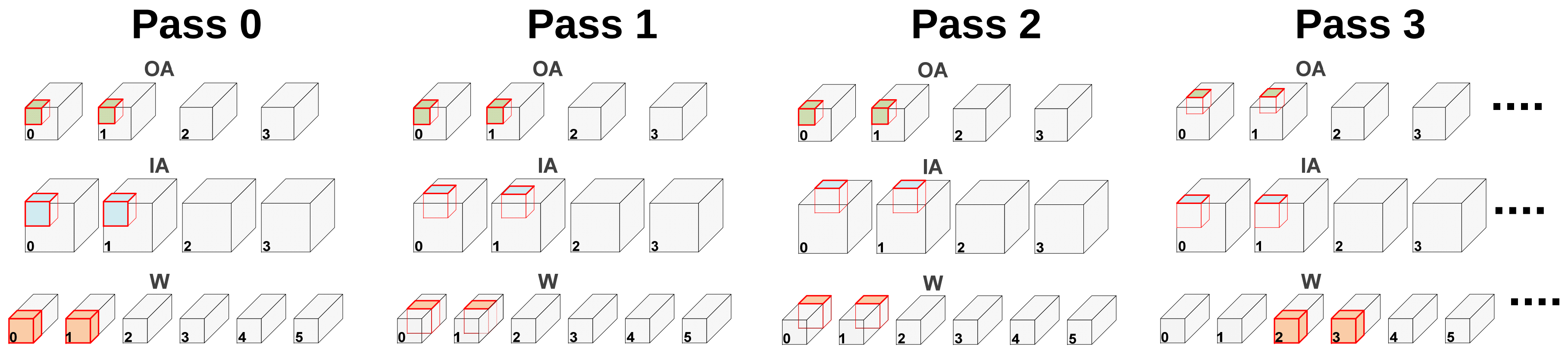
* $T_C = 2$, $T_M = 3$

No Local Reuse (NLR)

```
for (b = 0; b < B; b += TB) {
 for (e = 0; e < E; e += TE) {
  for (f = 0; f < F; f += TF) {
   for (m = 0; m < M; m += TM) {
    for (c = 0; c < C; c += TC) {
     Ibuf = I[b:b+TB-1][c:c+TC-1][e:e+TE+R-2][f:f+TF+S-2];
     Wbuf = W[m:m+TM-1][c:c+TC-1][0:R-1][0:S-1];
     ............................................
    }}}}}}}}
    O[b:b+TB-1][m:m+TM-1][e:e+TE-1][f:f+TF-1] = Obuf;
}}}}
```

Outer-Pipeline
(**Inter**-Pass)



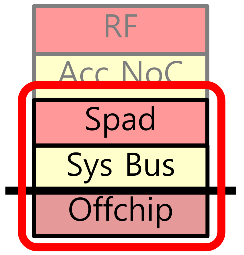


* C = 6, M = 6, TC = 2, TM = 3

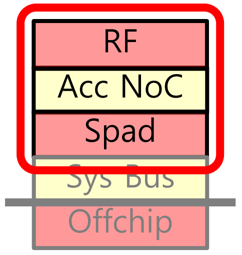
No Local Reuse (NLR)

```
for (b = 0; b < B; b += TB) {
 for (e = 0; e < E; e += TE) {
  for (f = 0; f < F; f += TF) {
   for (m = 0; m < M; m += TM) {
    for (c = 0; c < C; c += TC) {
     Ibuf = I[b:b+TB-1][c:c+TC-1][e:e+TE+R-2][f:f+TF+S-2];
     Wbuf = W[m:m+TM-1][c:c+TC-1][0:R-1][0:S-1];
     for (r = 0; r < R; r++) {
      for (s = 0; s < S; s++) {
       for (tb = 0; tb+b < min(B,b+TB); tb++) {
        for (te = 0; te+e < min(E,e+TE); te++) {
         for (tf = 0; tf+f < min(F,f+TF); tf++) {
          for (tm = 0; tm < TM; tm++) {
           for (tc = 0; tc < TC; tc++) {
            wx[tm][tc] = Wbuf[tm][tc][r][s];
            ix[tc] = Ibuf[tb][tc][te+r][tf+s];
            ox[tm] = Obuf[tb][tm][te][tf];
            Obuf[tb][tm][te][tf] = wx[tm][tc]*ix[tc]+ox[tm];
     }}}}}}}}
    O[b:b+TB-1][m:m+TM-1][e:e+TE-1][f:f+TF-1] = Obuf;
}}}}
```

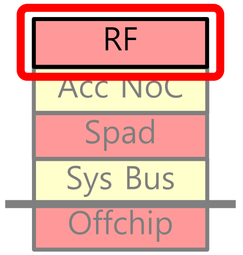
Outer-Pipeline (**Inter**-Pass)



Inner-Pipeline (**Intra**-Pass)



Unroll



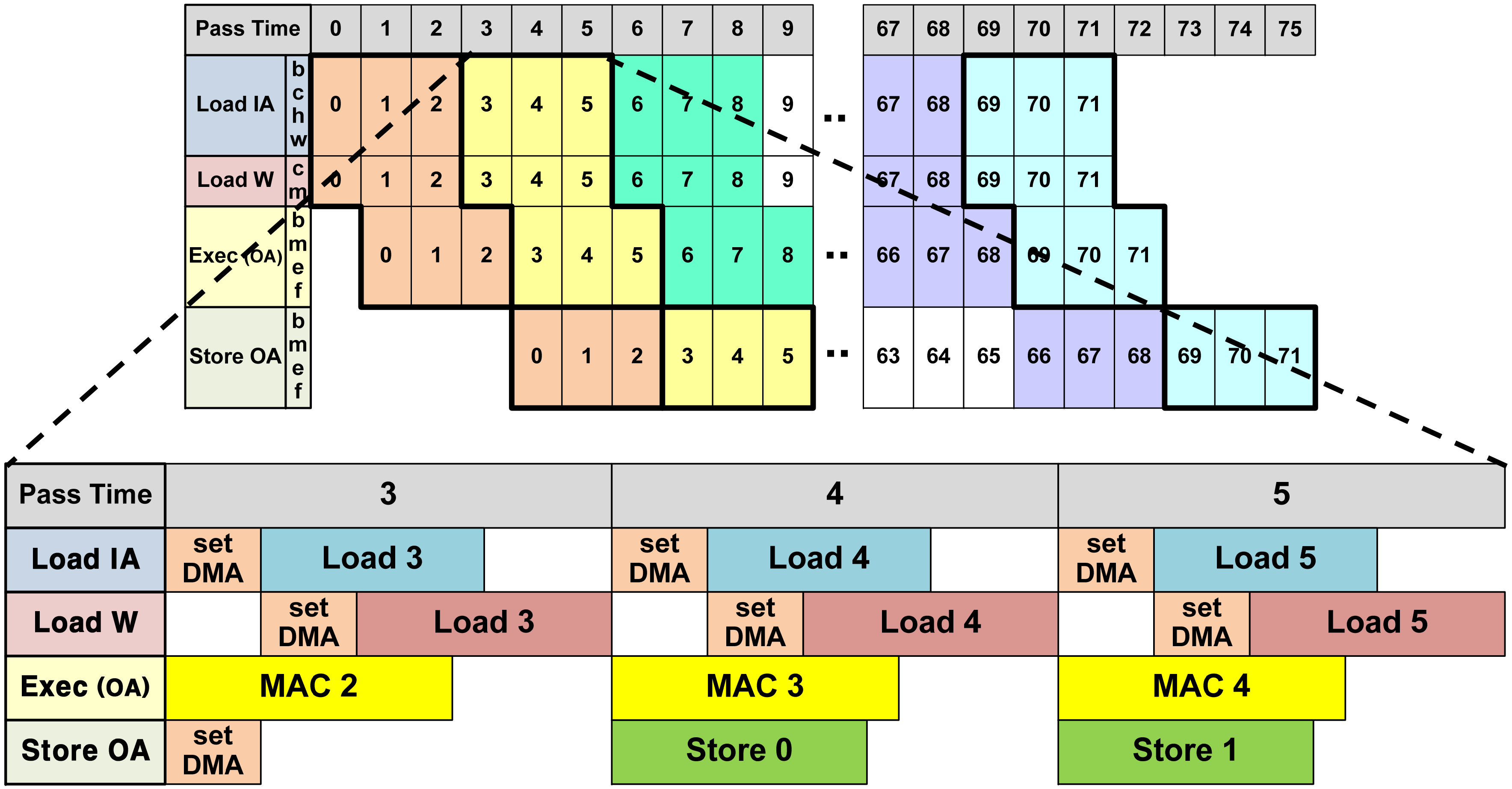
No Local Reuse (NLR)

❑ Outer pipeline (inter-pass)

Pass Time		0	1	2	3	4	5	6	7	8	9		67	68	69	70	71	72	73	74	75
Load IA	b	0	0	0	0	0	0	0	0	0	0	..	2	2	2	2	2				
	c	0	3	6	0	3	6	0	3	6	0		3	6	0	3	6				
	h	0	0	0	0	0	0	0	0	0	0		3	3	3	3	3				
	w	0	0	0	0	0	0	0	0	0	3		3	3	3	3	3				
Load W	c	0	3	6	0	3	6	0	3	6	0		3	6	0	3	6				
	m	0	0	0	2	2	2	4	4	4	0		2	2	4	4	4				
Exec (OA)	b			0			0			0		..		2			2				
	m			0			2			4				2			4				
	e			0			0			0				3			3				
	f			0			0			0				3			3				
Store OA	b						0			0		..		2			2			2	
	m						0			2				0			2			4	
	e						0			0				3			3			3	
	f						0			0				3			3			3	

No Local Reuse (NLR)

- ❑ Outer pipeline (inter-pass) (cont'd)



Pass Time		0	1	2	3	4	5	6	7	8	9		67	68	69	70	71	72	73	74	75
Load IA	b c h w	0	1	2	3	4	5	6	7	8	9	..	67	68	69	70	71				
Load W	c m	0	1	2	3	4	5	6	7	8	9		67	68	69	70	71				
Exec (OA)	b m e f		0	1	2	3	4	5	6	7	8	..	66	67	68	69	70	71			
Store OA	b m e f					0	1	2	3	4	5	..	63	64	65	66	67	68	69	70	71

Pass Time	3			4			5		
Load IA	set DMA	Load 3		set DMA	Load 4		set DMA	Load 5	
Load W		set DMA	Load 3		set DMA	Load 4		set DMA	Load 5
Exec (OA)	MAC 2			MAC 3			MAC 4		
Store OA	set DMA			Store 0			Store 1		

No Local Reuse (NLR)

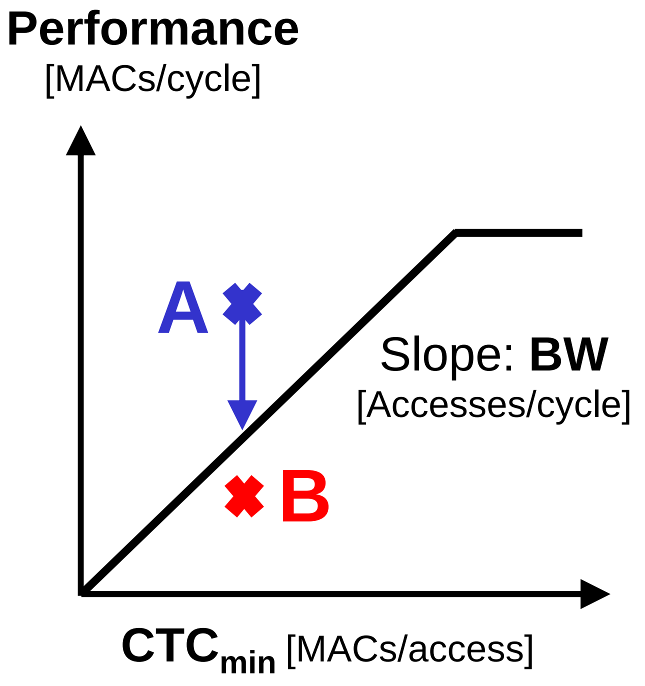
❑ Roofline model (revisited)

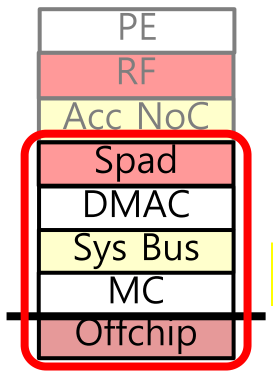
A: Communication -Limited

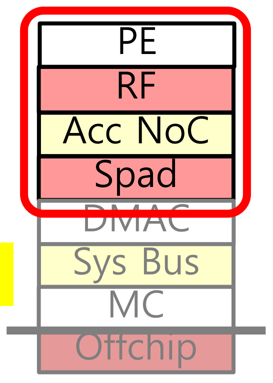
Pass Time	3n+1		
Load IA	set DMA	Load 3n+1	
Load W		set DMA	Load 3n+1
Exec (OA)	MAC 3n		
Store OA	Store 3n-3		

B: Computation -Limited

Pass Time	3n+1		
Load IA	set DMA	Load 3n+1	
Load W		set DMA	Load 3n+1
Exec (OA)	MAC 3n		
Store OA	Store 3n-3		

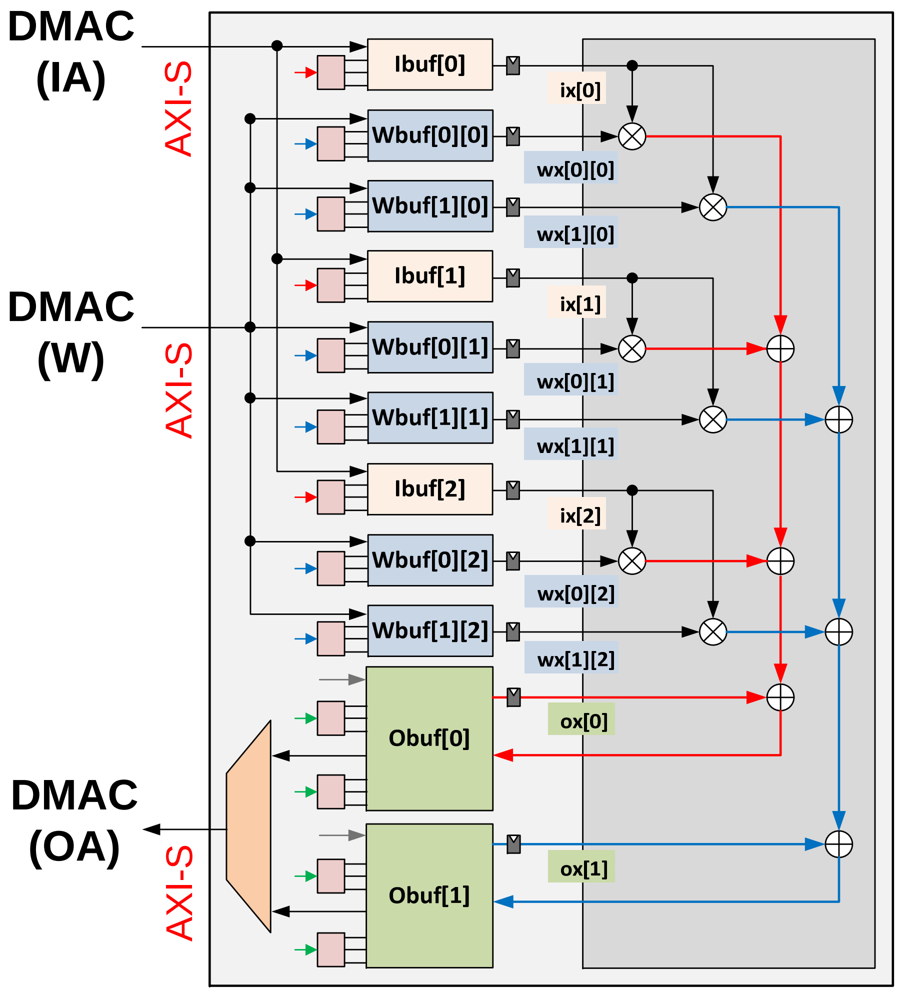






$$CTC_{min} := \min(CTC_{Load}, CTC_{Store})$$

Available Hardware Blocks



Denoted as "hls23x" for layer x

* $T_C = 2$, $T_M = 3$

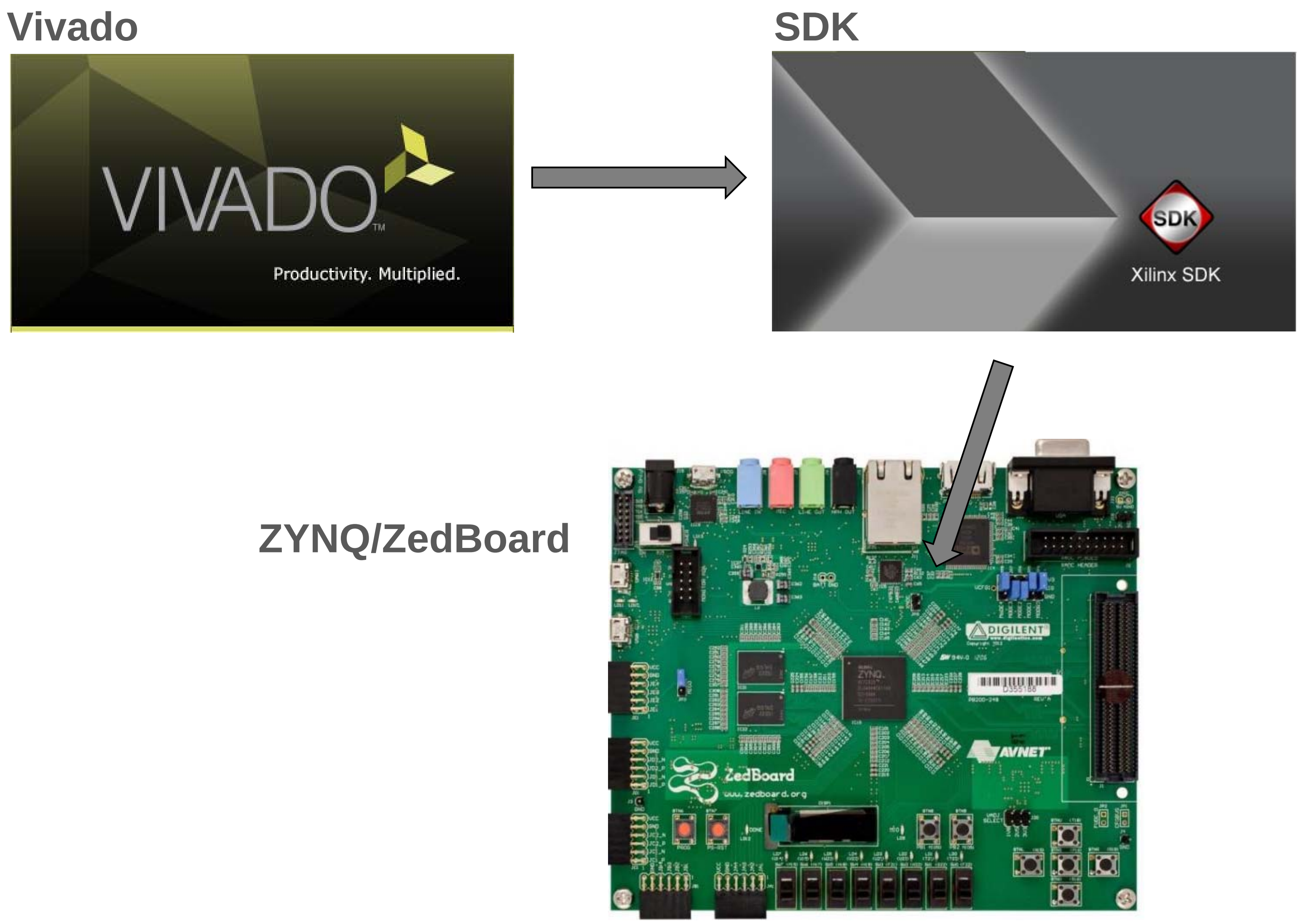
Available Hardware Blocks

You're allowed to use whatever hardware blocks (even to use all the blocks) to improve the performance

$T_C \times T_M$	3 x 4	2 x 8	8 x 2	2 x 4	4 x 2
L1	hls341	NA	NA	NA	NA
L2	NA	hls282	hls822	hls242	hls422
L3	NA	hls283	hls823	hls243	hls423
L4	NA	hls284	hls824	hls244	hls424
L5	NA	hls285	hls825	hls245	Hls425

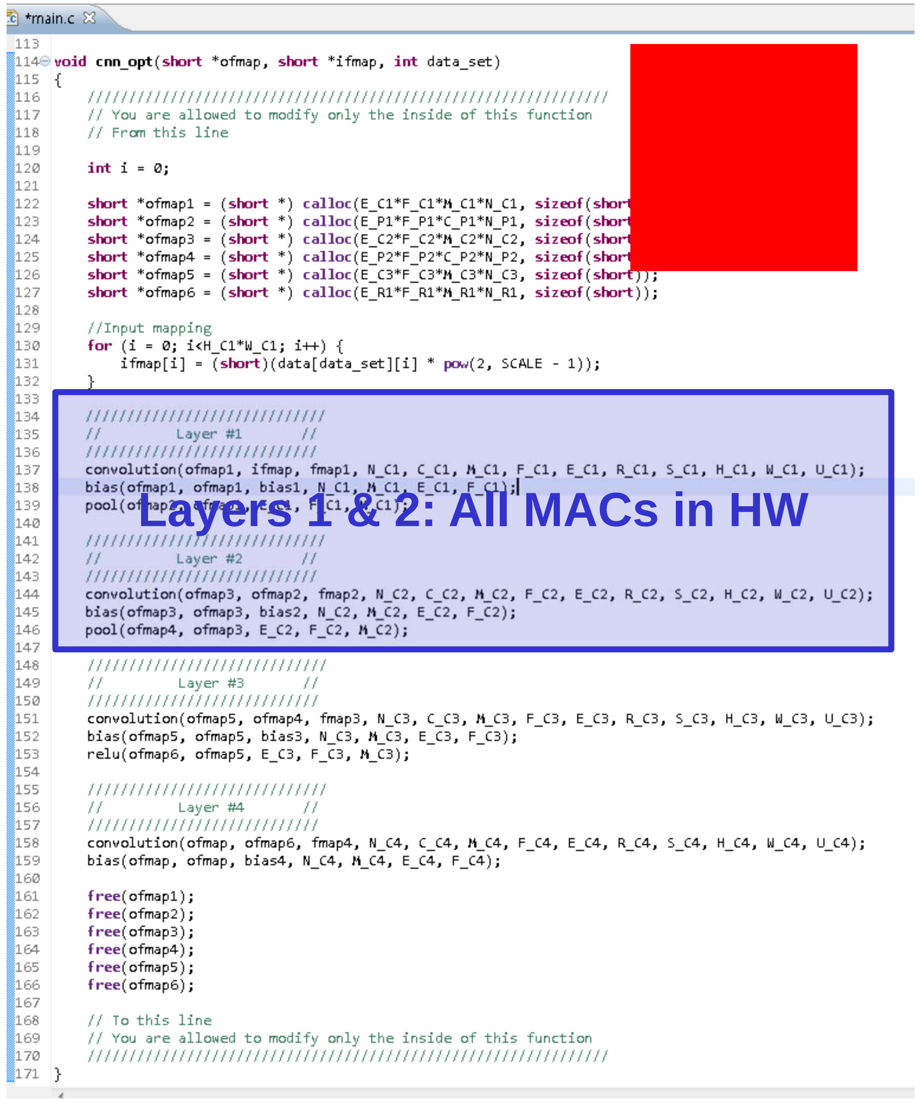
Will be available in the course webpage soon

Design Flow


Vivado
VIVADO
Productivity. Multiplied.
SDK
SDK
Xilinx SDK
ZYNQ/ZedBoard
ZedBoard
DIGILENT
AVNET

Design Constraints

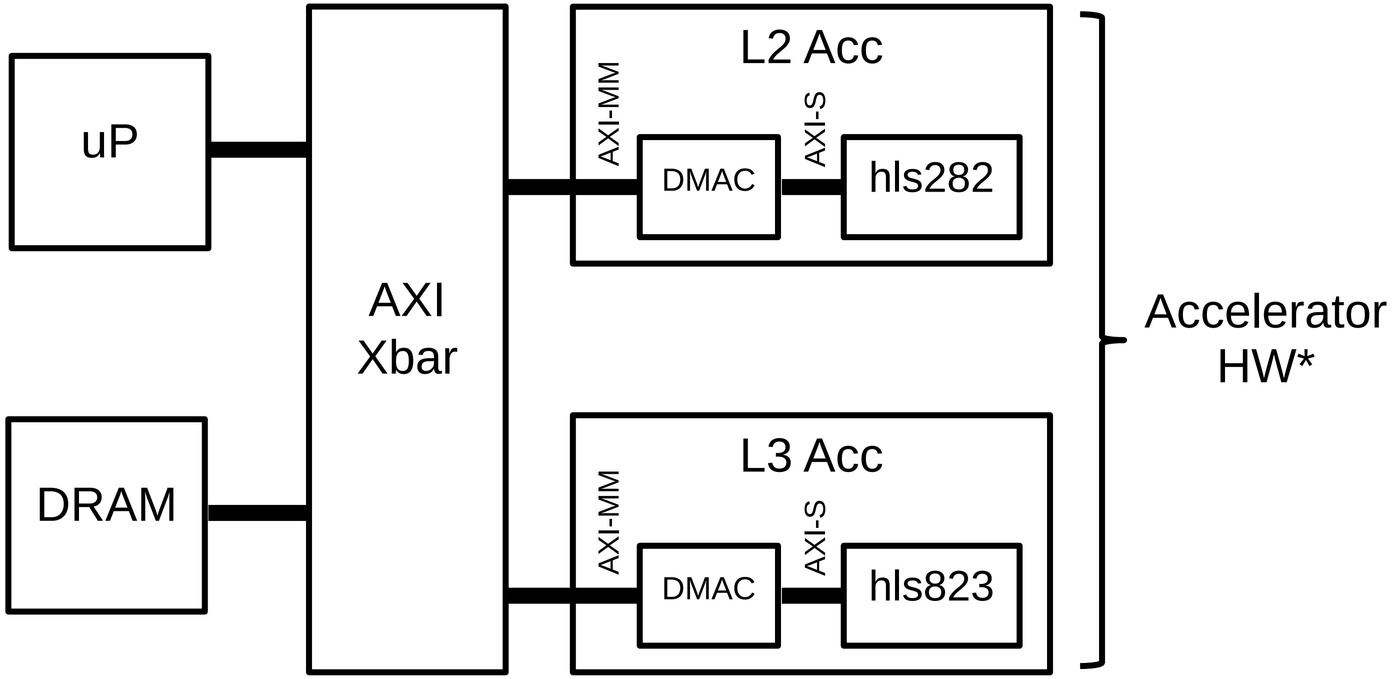
- ❑ You are allowed to modify only the following function where all the multiply-accumulate (MAC) operations are implemented in HW
  - **conv_hw**
- ❑ You are not allowed to change anything outside the above function



```c
void cnn_opt(short *ofmap, short *ifmap, int data_set)
{
    /////////////////////////////////////////////////////////////////
    // You are allowed to modify only the inside of this function
    // From this line

    int i = 0;

    short *ofmap1 = (short *) calloc(E_C1*F_C1*M_C1*N_C1, sizeof(short [redacted]
    short *ofmap2 = (short *) calloc(E_P1*F_P1*C_P1*N_P1, sizeof(short
    short *ofmap3 = (short *) calloc(E_C2*F_C2*M_C2*N_C2, sizeof(short
    short *ofmap4 = (short *) calloc(E_P2*F_P2*C_P2*N_P2, sizeof(short
    short *ofmap5 = (short *) calloc(E_C3*F_C3*M_C3*N_C3, sizeof(short));
    short *ofmap6 = (short *) calloc(E_R1*F_R1*M_R1*N_R1, sizeof(short));

    //Input mapping
    for (i = 0; i<H_C1*W_C1; i++) {
        ifmap[i] = (short)(data[data_set][i] * pow(2, SCALE - 1));
    }

    /////////////////////////////
    //          Layer #1      //
    ////////////////////////////
    convolution(ofmap1, ifmap, fmap1, N_C1, C_C1, M_C1, F_C1, E_C1, R_C1, S_C1, H_C1, W_C1, U_C1);
    bias(ofmap1, ofmap1, bias1, N_C1, M_C1, E_C1, F_C1);
    pool(ofmap2, ofmap1, E_C1, F_C1, M_C1);

    /////////////////////////////
    //          Layer #2      //
    ////////////////////////////
    convolution(ofmap3, ofmap2, fmap2, N_C2, C_C2, M_C2, F_C2, E_C2, R_C2, S_C2, H_C2, W_C2, U_C2);
    bias(ofmap3, ofmap3, bias2, N_C2, M_C2, E_C2, F_C2);
    pool(ofmap4, ofmap3, E_C2, F_C2, M_C2);

    /////////////////////////////
    //          Layer #3      //
    ////////////////////////////
    convolution(ofmap5, ofmap4, fmap3, N_C3, C_C3, M_C3, F_C3, E_C3, R_C3, S_C3, H_C3, W_C3, U_C3);
    bias(ofmap5, ofmap5, bias3, N_C3, M_C3, E_C3, F_C3);
    relu(ofmap6, ofmap5, E_C3, F_C3, M_C3);

    /////////////////////////////
    //          Layer #4      //
    ////////////////////////////
    convolution(ofmap, ofmap6, fmap4, N_C4, C_C4, M_C4, F_C4, E_C4, R_C4, S_C4, H_C4, W_C4, U_C4);
    bias(ofmap, ofmap, bias4, N_C4, M_C4, E_C4, F_C4);

    free(ofmap1);
    free(ofmap2);
    free(ofmap3);
    free(ofmap4);
    free(ofmap5);
    free(ofmap6);

    // To this line
    // You are allowed to modify only the inside of this function
    /////////////////////////////////////////////////////////////////
}
```

Design Constraints

- Example of accelerator
  - Layers 2 & 3 in HW and the others in SW



* The FPGA considered here (ZC7020) is assumed to accommodate no more than **220 MAC hardware units** and **570KBs BRAM**

Evaluation

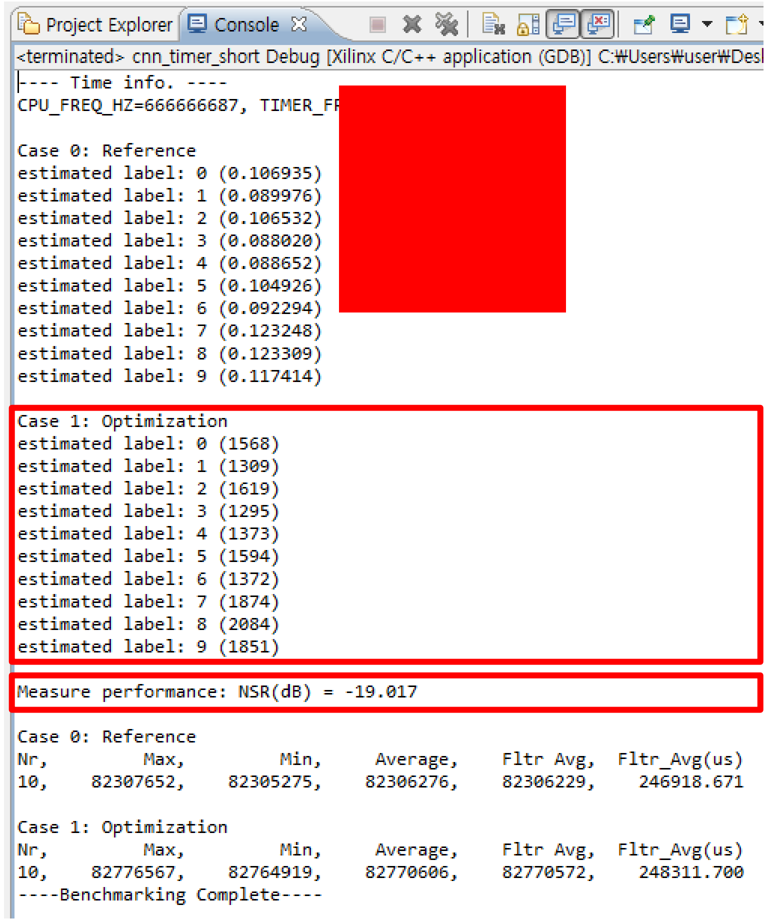
- ❑ Submission completeness
  - Reproducibility (10pt)
- ❑ Accuracy
  - Classification error (20pt)
  - Quantization error (10pt)
- ❑ Performance
  - Execution time (30pt)
- ❑ Technical solidness
  - Any good ideas or contributions (30pt)

Reproducibility

- Make sure that your submission is reproducible
  - In other words, it is possible to reproduce your design together with the claimed accuracy and performance using only <u>the files that you submitted</u>

Accuracy

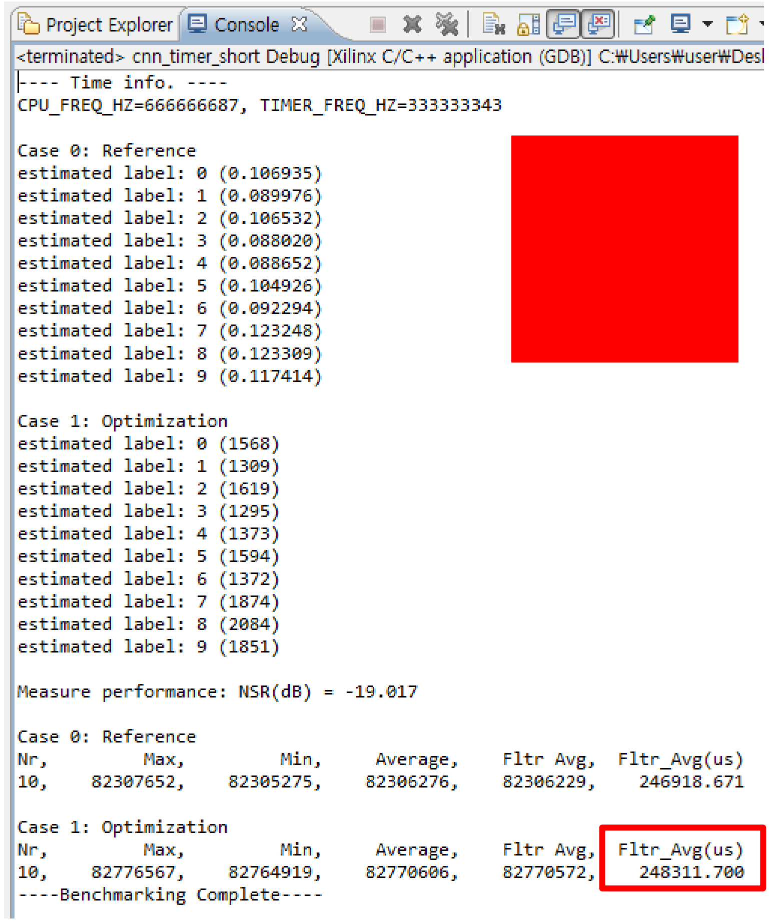
- ❑ Run the main program to check the accuracy of **`conv_hw`**
  - Classification error
    - ✓ A new set of images will be given onsite
  - Noise-to-signal ratio (NSR)
    - ✓ Difference from **`conv_ref`**



```
<terminated> cnn_timer_short Debug [Xilinx C/C++ application (GDB)] C:₩Users₩user₩Desl
---- Time info. ----
CPU_FREQ_HZ=666666687, TIMER_F [redacted]

Case 0: Reference
estimated label: 0 (0.106935)
estimated label: 1 (0.089976)
estimated label: 2 (0.106532)
estimated label: 3 (0.088020)
estimated label: 4 (0.088652)
estimated label: 5 (0.104926)
estimated label: 6 (0.092294)
estimated label: 7 (0.123248)
estimated label: 8 (0.123309)
estimated label: 9 (0.117414)

Case 1: Optimization
estimated label: 0 (1568)
estimated label: 1 (1309)
estimated label: 2 (1619)
estimated label: 3 (1295)
estimated label: 4 (1373)
estimated label: 5 (1594)
estimated label: 6 (1372)
estimated label: 7 (1874)
estimated label: 8 (2084)
estimated label: 9 (1851)

Measure performance: NSR(dB) = -19.017

Case 0: Reference
Nr,          Max,         Min,     Average,     Fltr Avg,  Fltr_Avg(us)
10,    82307652,    82305275,    82306276,    82306229,    246918.671

Case 1: Optimization
Nr,          Max,         Min,     Average,     Fltr Avg,  Fltr_Avg(us)
10,    82776567,    82764919,    82770606,    82770572,    248311.700
----Benchmarking Complete----
```

Performance

- Run the main program to check the performance of **conv_hw**
  - Execution time (-O3)



```
<terminated> cnn_timer_short Debug [Xilinx C/C++ application (GDB)] C:₩Users₩user₩Desl
---- Time info. ----
CPU_FREQ_HZ=666666687, TIMER_FREQ_HZ=333333343

Case 0: Reference
estimated label: 0 (0.106935)
estimated label: 1 (0.089976)
estimated label: 2 (0.106532)
estimated label: 3 (0.088020)
estimated label: 4 (0.088652)
estimated label: 5 (0.104926)
estimated label: 6 (0.092294)
estimated label: 7 (0.123248)
estimated label: 8 (0.123309)
estimated label: 9 (0.117414)

Case 1: Optimization
estimated label: 0 (1568)
estimated label: 1 (1309)
estimated label: 2 (1619)
estimated label: 3 (1295)
estimated label: 4 (1373)
estimated label: 5 (1594)
estimated label: 6 (1372)
estimated label: 7 (1874)
estimated label: 8 (2084)
estimated label: 9 (1851)

Measure performance: NSR(dB) = -19.017

Case 0: Reference
Nr,         Max,         Min,     Average,    Fltr Avg,  Fltr_Avg(us)
10,    82307652,    82305275,    82306276,    82306229,    246918.671

Case 1: Optimization
Nr,         Max,         Min,     Average,    Fltr Avg,  Fltr_Avg(us)
10,    82776567,    82764919,    82770606,    82770572,    248311.700
----Benchmarking Complete----
```

Technical Solidness

- Any good ideas or contributions that help to improve the performance such as
  - Optimum allocation of accelerator resources (MAC units & BRAMs) across layers
  - Maximum utilization of accelerator resources
- 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):
  - “Maximizing CNN accelerator efficiency through resource partitioning”, Y. Shen et al., ISCA 2017

Important Notes

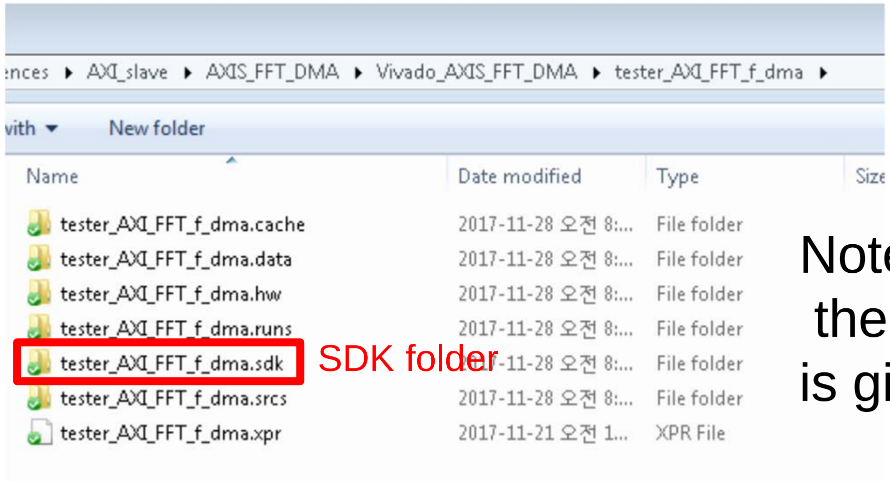
- You should keep the reference implementation (**`conv_ref`**) unchanged
- The compiler optimization level should be set to -O3 when the performance is measured
- You will have some bonus points if you implement some of the paper ideas cited in the previous slide
- You may be able to provide two versions of the convolution function, e.g.,
  - 1st version with the best performance
  - 2nd version implementing paper idea(s)

Submission

- ❑ Due date: **Dec. 15 (Sun), 2019, 15:00:00** 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

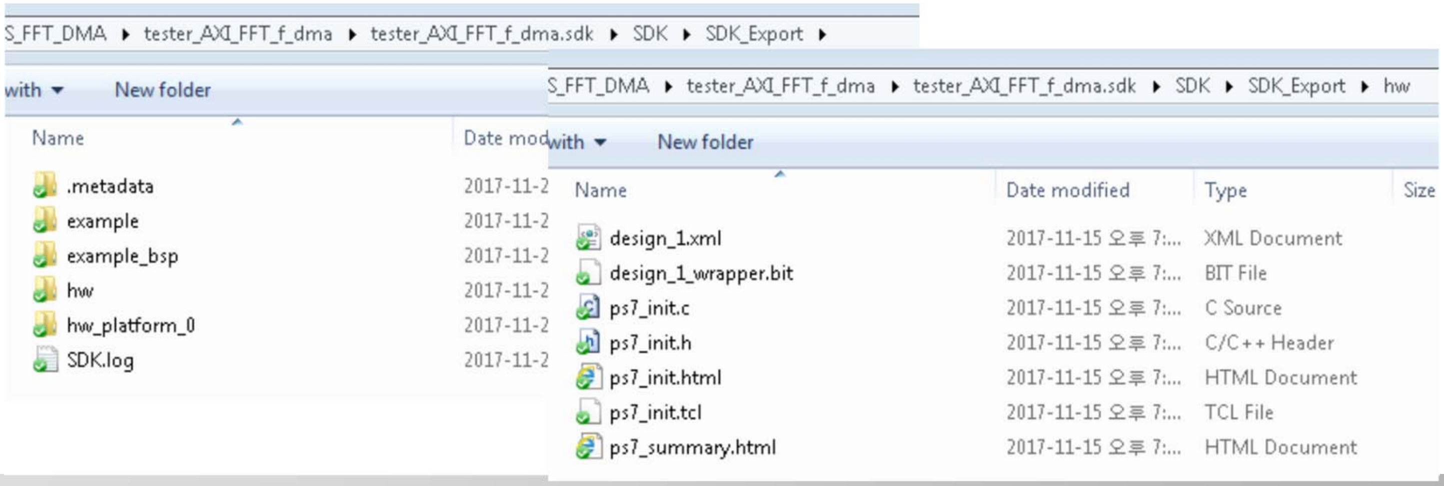


ences ▸ AXI_slave ▸ AXIS_FFT_DMA ▸ Vivado_AXIS_FFT_DMA ▸ tester_AXI_FFT_f_dma ▸

with ▾ New folder

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.srcs	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"



S_FFT_DMA ▸ tester_AXI_FFT_f_dma ▸ tester_AXI_FFT_f_dma.sdk ▸ SDK ▸ SDK_Export ▸

with ▾ New folder

Name	Date mod
.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

S_FFT_DMA ▸ tester_AXI_FFT_f_dma ▸ tester_AXI_FFT_f_dma.sdk ▸ SDK ▸ SDK_Export ▸ hw

with ▾ New folder

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. 16 (Mon) 10:30~13:30, New Eng. Bldg. #1113**
  - Team-presentation with exactly the same **slideset files** as submitted on **Dec. 15**
  - Team-demo with exactly the same **SDK folder** as submitted on **Dec. 15**

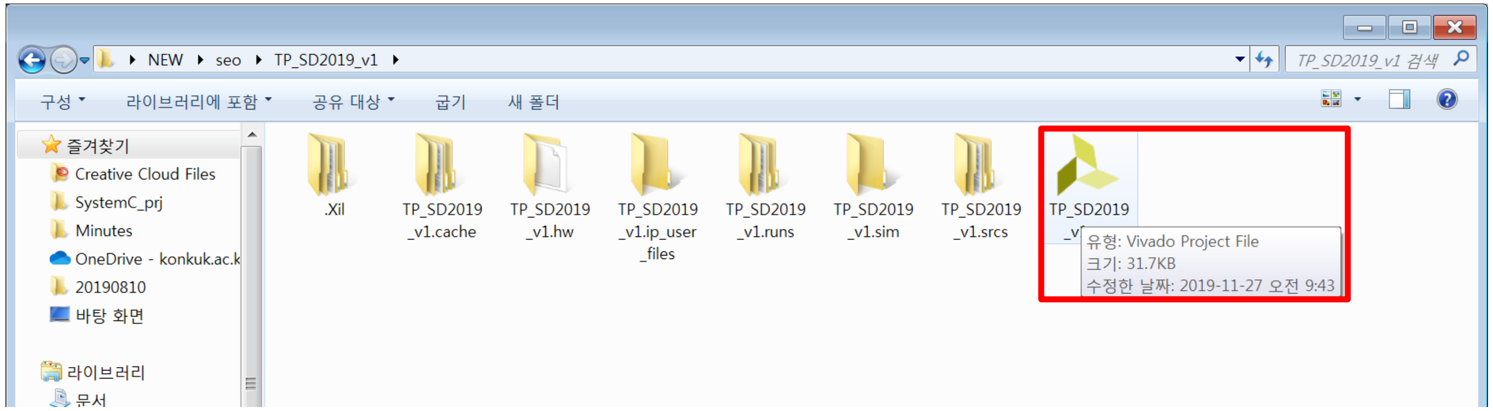
1. *Bring a storage device (e.g., HDD) having the entire **project folder** just in case (e.g., when the SDK folder does not work)*
2. *Note that any progress made later than **Dec. 15** can**not** be counted for evaluation*

Appendix

Open Vivado projects

❑ Open Vivado projects

- Unzip and open '***TP_SD2019_v1***' folder and run file **'TP_SD2019_v1.xpr'**
  Download link:
  https://drive.google.com/open?id=1zwLSdnKHZNEAO4TTdc0Yk2tlPeKX14SV

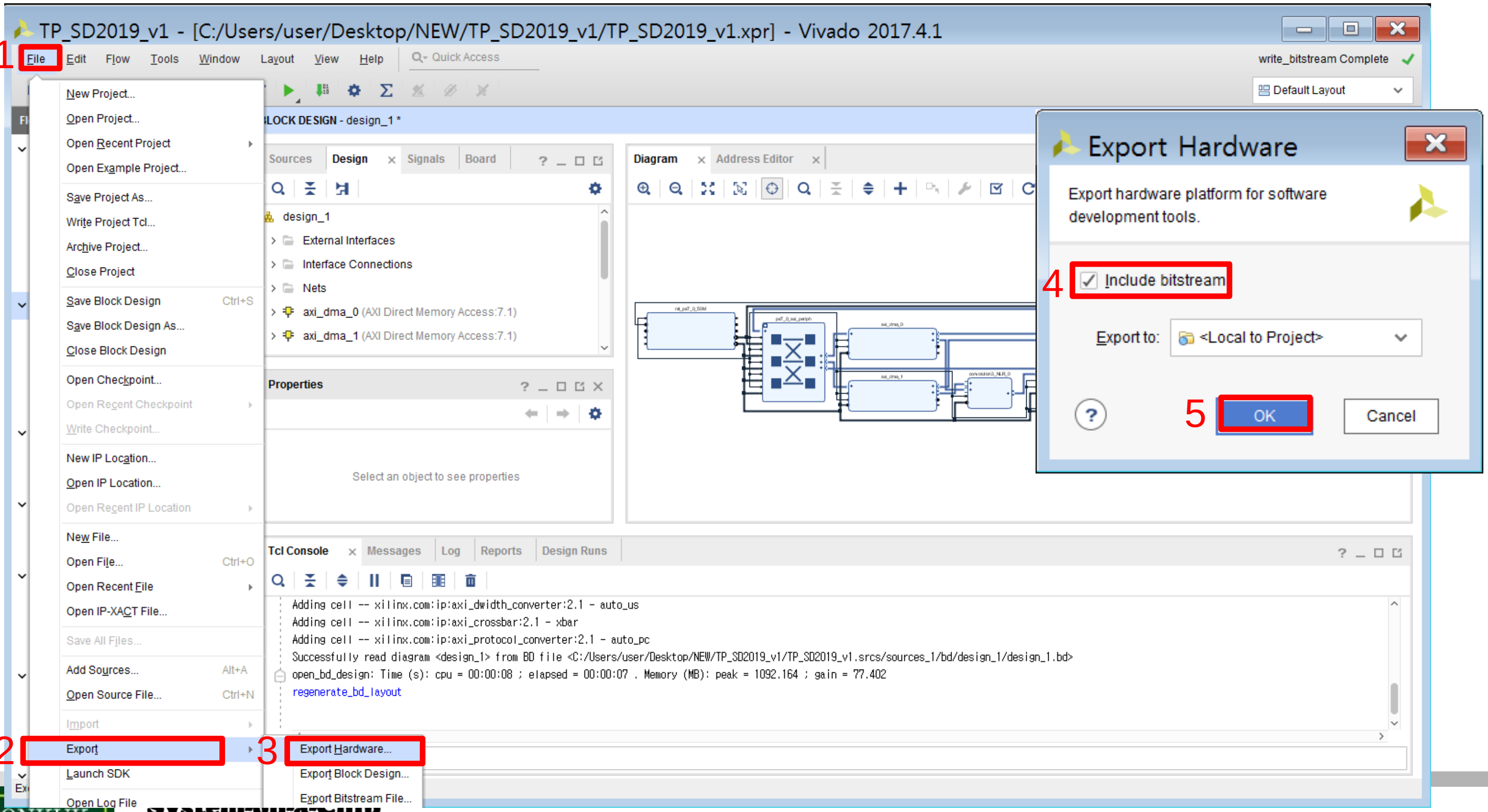


유형: Vivado Project File
크기: 31.7KB
수정한 날짜: 2019-11-27 오전 9:43

Open Vivado projects

❑ Export Hardware

- Click ***'Export > Export Hardware'*** in ***'File'*** menu
- Check ***'Include Bitstream'*** and then click ***'OK'***



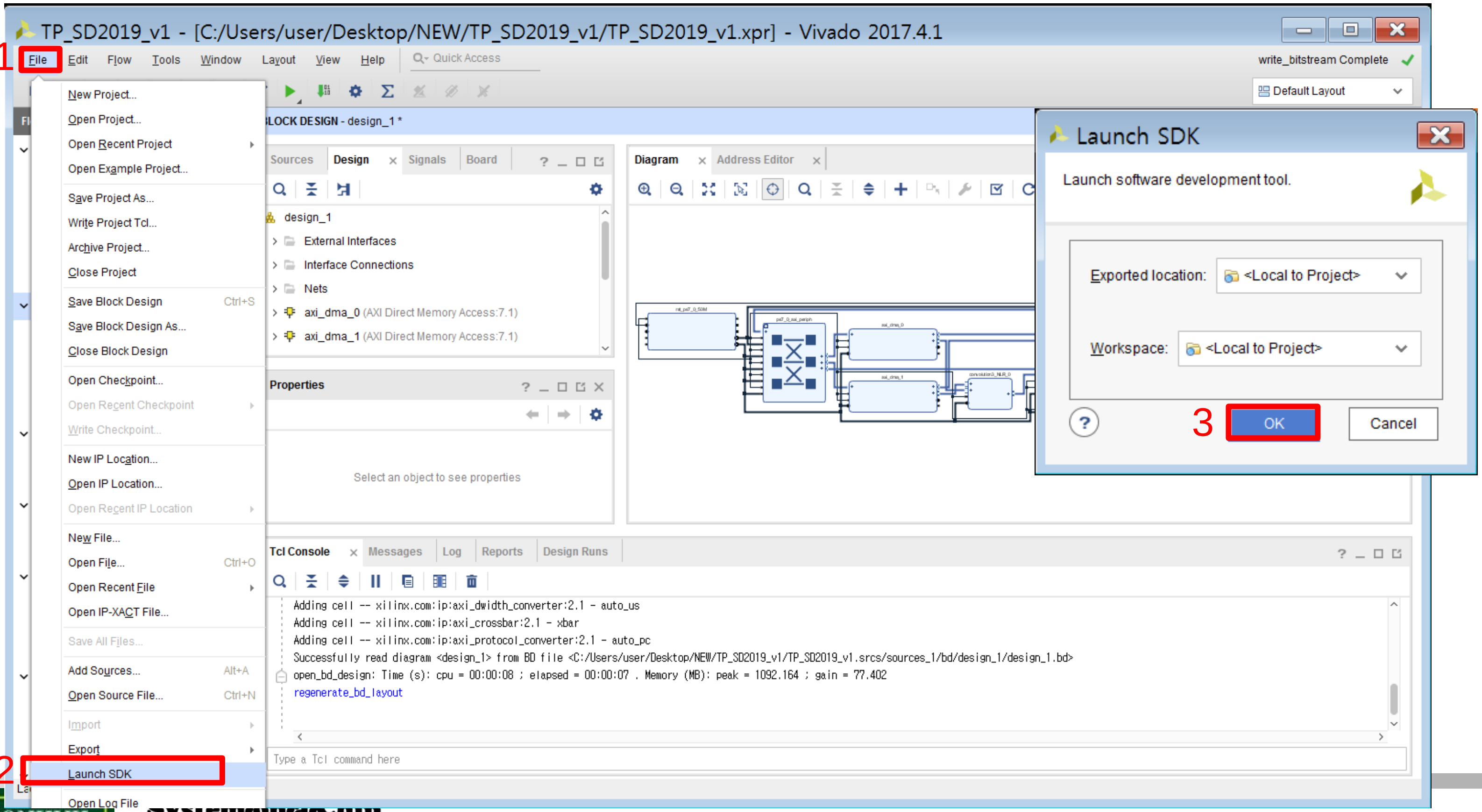
Export hardware platform for software development tools.

```
Adding cell -- xilinx.com:ip:axi_dwidth_converter:2.1 - auto_us
Adding cell -- xilinx.com:ip:axi_crossbar:2.1 - xbar
Adding cell -- xilinx.com:ip:axi_protocol_converter:2.1 - auto_pc
Successfully read diagram <design_1> from BD file <C:/Users/user/Desktop/NEW/TP_SD2019_v1/TP_SD2019_v1.srcs/sources_1/bd/design_1/design_1.bd>
open_bd_design: Time (s): cpu = 00:00:08 ; elapsed = 00:00:07 . Memory (MB): peak = 1092.164 ; gain = 77.402
regenerate_bd_layout
```

Open Vivado projects

❑ Launch SDK

- Click ***'Launch SDK'*** in ***'File'*** menu
- Click ***'OK'***


TP_SD2019_v1 - [C:/Users/user/Desktop/NEW/TP_SD2019_v1/TP_SD2019_v1.xpr] - Vivado 2017.4.1

write_bitstream Complete

Launch SDK

Launch software development tool.

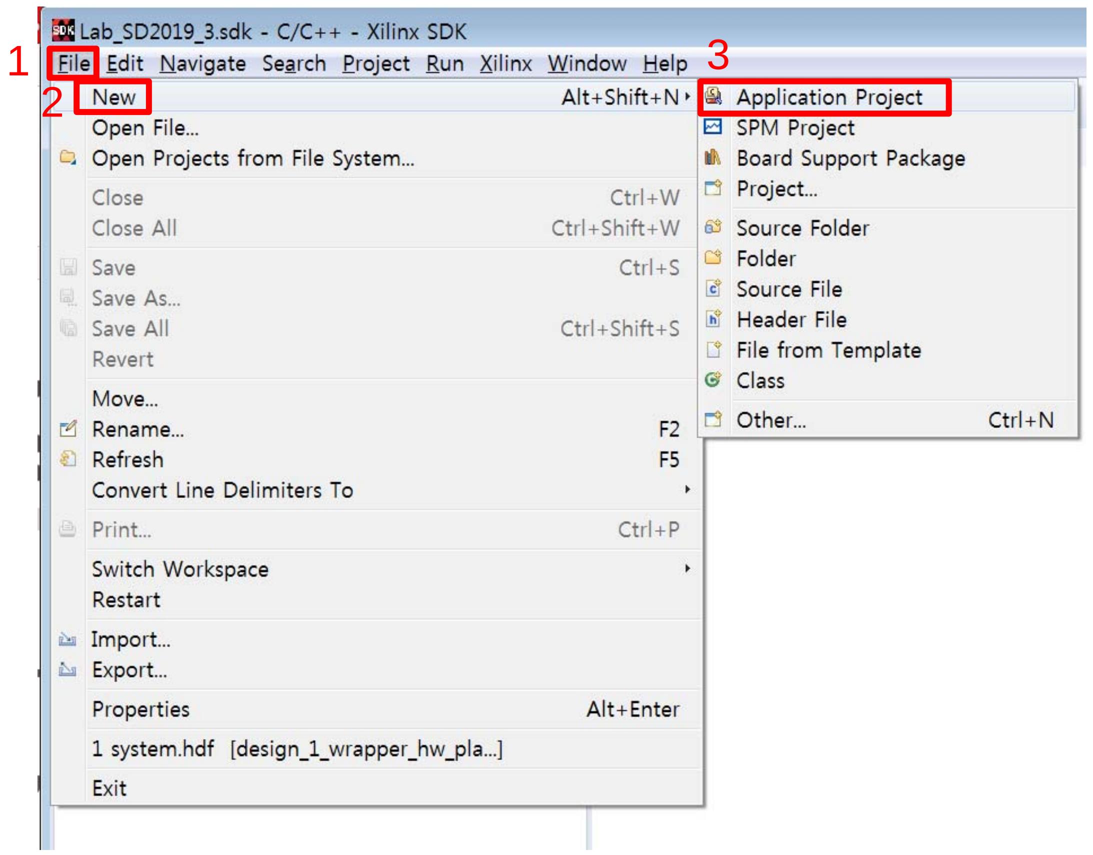
Exported location: <Local to Project>

Workspace: <Local to Project>

```
Adding cell -- xilinx.com:ip:axi_dwidth_converter:2.1 - auto_us
Adding cell -- xilinx.com:ip:axi_crossbar:2.1 - xbar
Adding cell -- xilinx.com:ip:axi_protocol_converter:2.1 - auto_pc
Successfully read diagram <design_1> from BD file <C:/Users/user/Desktop/NEW/TP_SD2019_v1/TP_SD2019_v1.srcs/sources_1/bd/design_1/design_1.bd>
open_bd_design: Time (s): cpu = 00:00:08 ; elapsed = 00:00:07 . Memory (MB): peak = 1092.164 ; gain = 77.402
regenerate_bd_layout
```

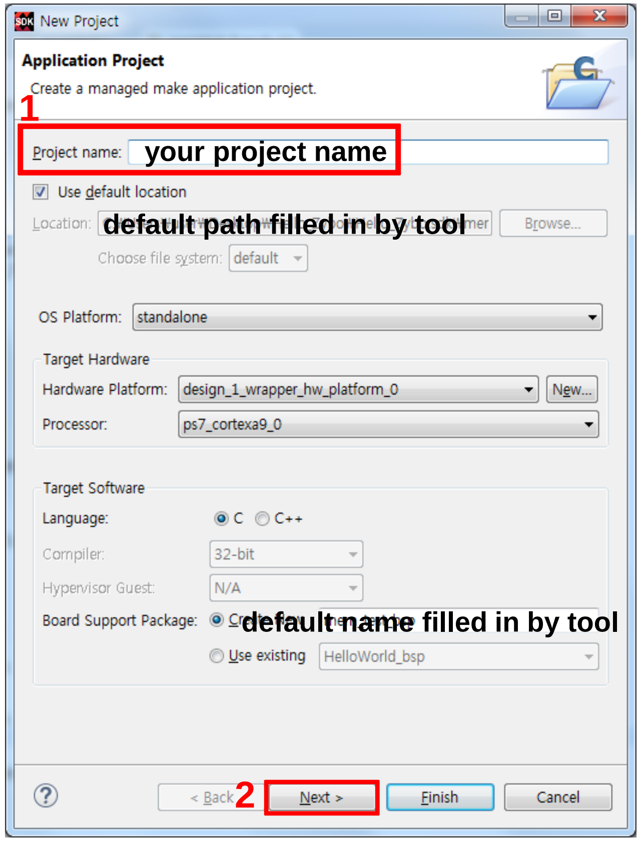
Running C Applications

- ❑ Create a C application project
  - Click ***'File' > 'New' > 'Application Project'***



Running C Applications

- ❑ Create a C application project (cont'd)
  - Type the project name
  - Click ***'Next'***


New Project

Application Project

Create a managed make application project.

1

Project name: your project name

[✓] Use default location

Location: default path filled in by tool Browse...

Choose file system: default

OS Platform: standalone

Target Hardware

Hardware Platform: design_1_wrapper_hw_platform_0 New...

Processor: ps7_cortexa9_0

Target Software

Language: [✓ C] C++

Compiler: 32-bit

Hypervisor Guest: N/A

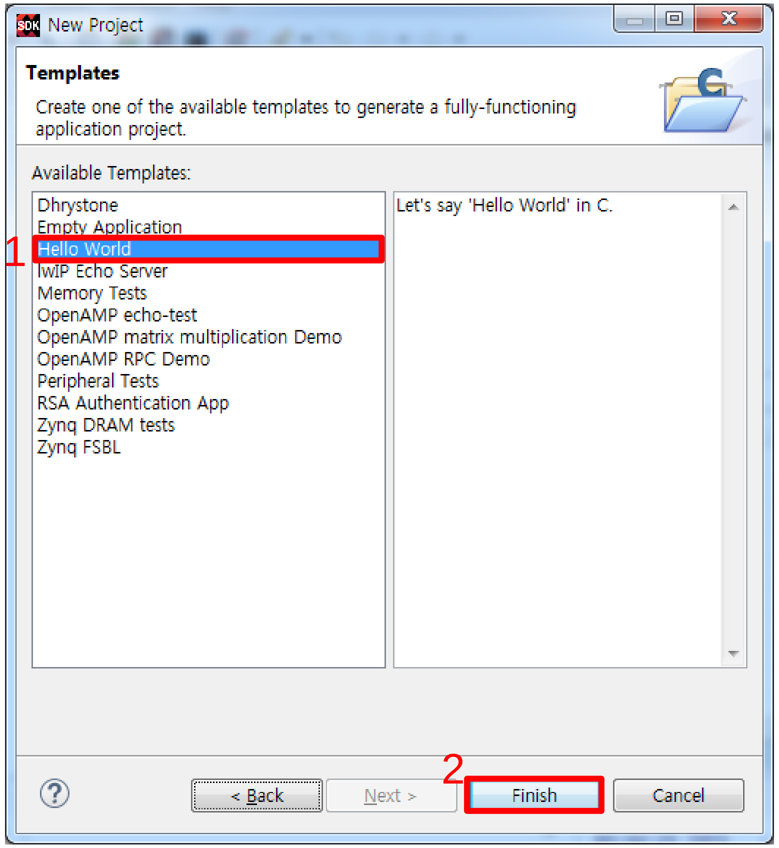
Board Support Package: [✓ Create New] default name filled in by tool

Use existing HelloWorld_bsp

Running C Applications

❑ Create a C application project (cont'd)

- Choose ***'Hello World'*** and then click ***'Finish'***



New Project

Templates

Create one of the available templates to generate a fully-functioning application project.

Available Templates:

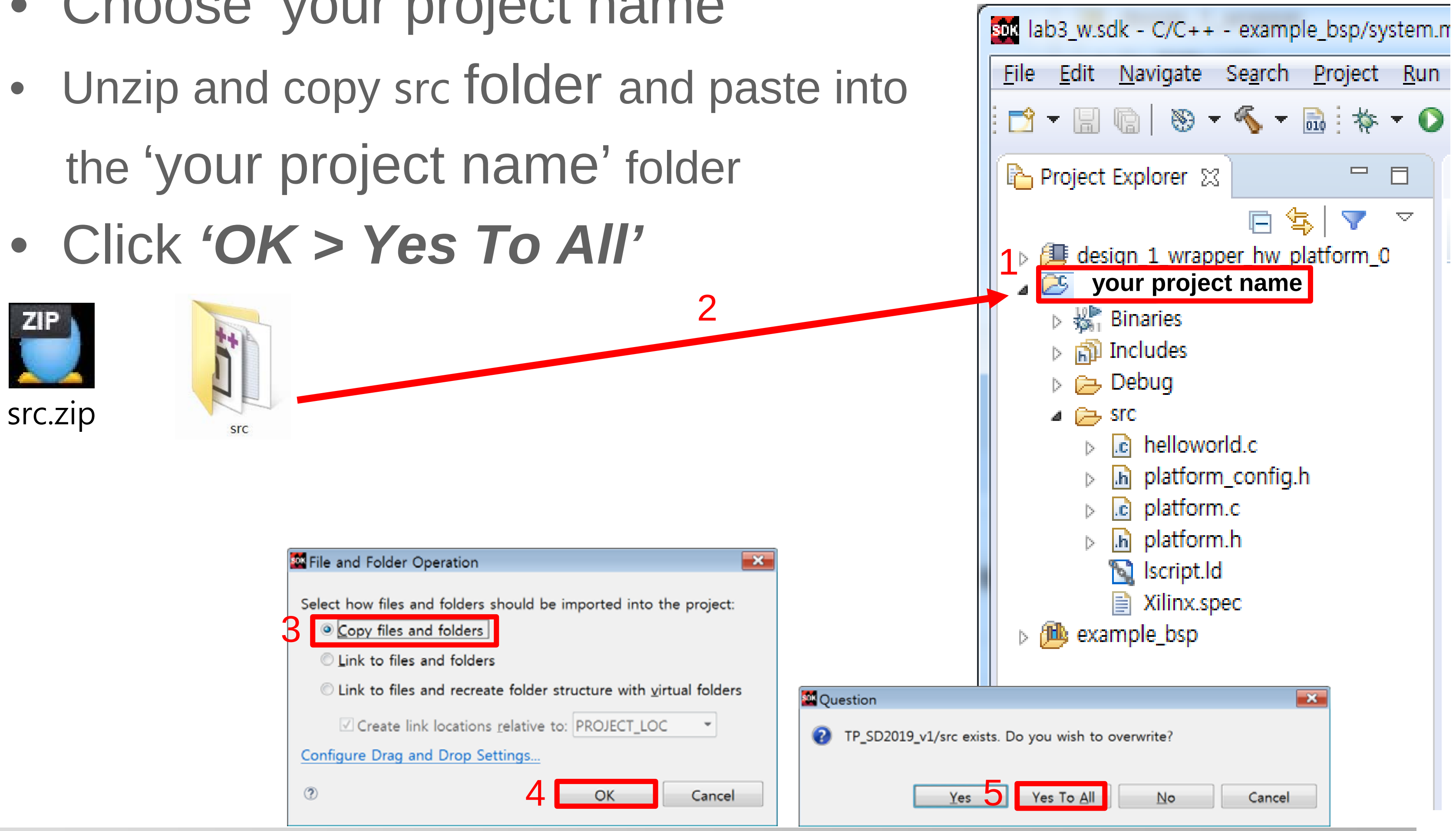
- Dhrystone
- Empty Application
- Hello World
- lwIP Echo Server
- Memory Tests
- OpenAMP echo-test
- OpenAMP matrix multiplication Demo
- OpenAMP RPC Demo
- Peripheral Tests
- RSA Authentication App
- Zynq DRAM tests
- Zynq FSBL

Let's say 'Hello World' in C.

Running C Applications

❑ Add source files

- Choose 'your project name'
- Unzip and copy src folder and paste into the 'your project name' folder
- Click ***'OK > Yes To All'***



File and Folder Operation

Select how files and folders should be imported into the project:

- [✓ Copy files and folders]
- Link to files and folders
- Link to files and recreate folder structure with virtual folders

☑ Create link locations relative to: PROJECT_LOC

Configure Drag and Drop Settings...

OK Cancel

Question

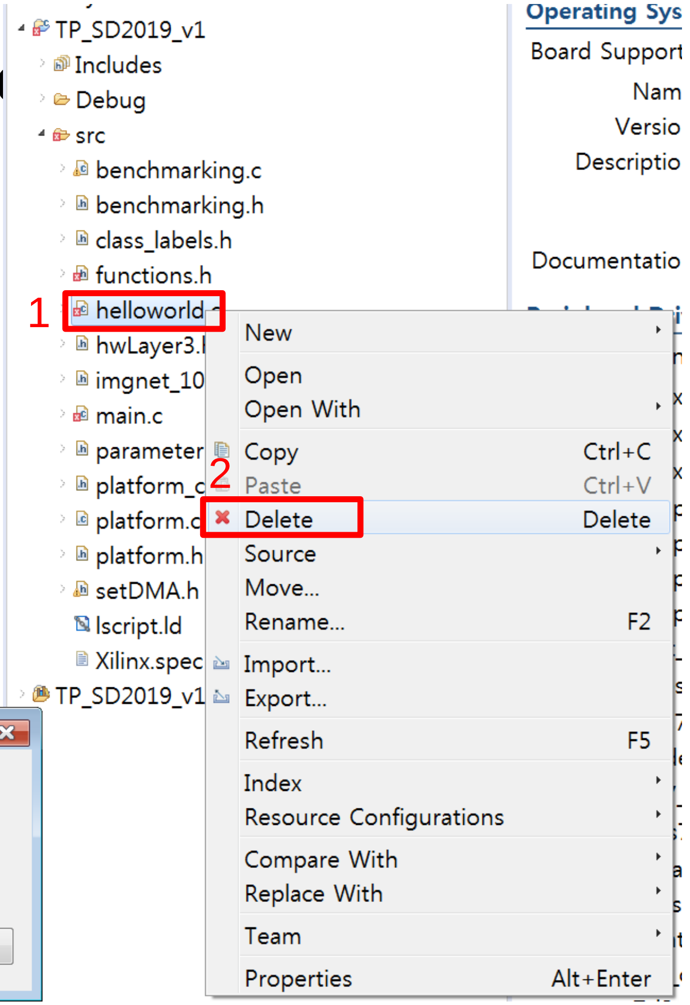
TP_SD2019_v1/src exists. Do you wish to overwrite?

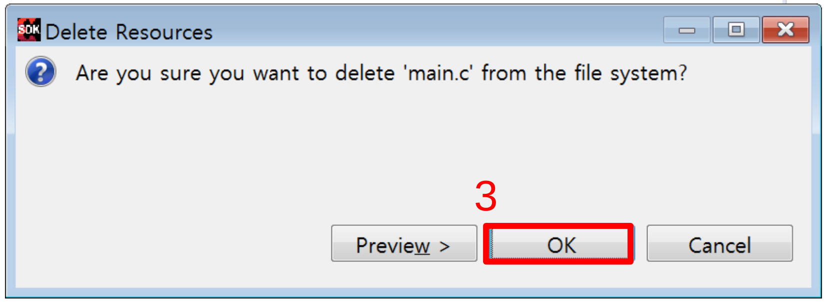
Yes [✓ Yes To All] No Cancel

Running C Applications

❑ Add source files (Cont'd)

- Right-click 'hellowrld.c' in src f
- Click ***'Delete > OK'***

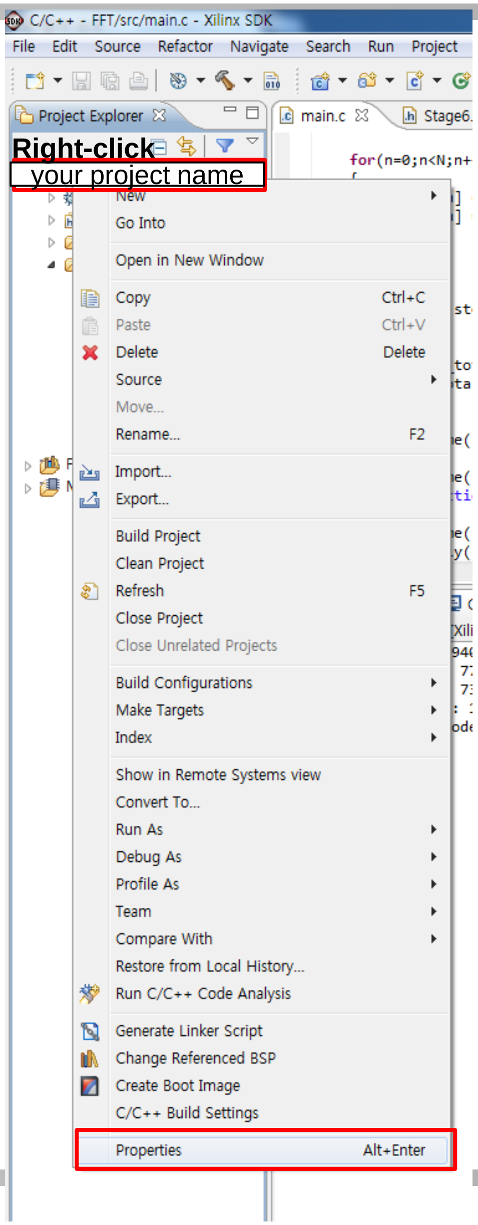




Are you sure you want to delete 'main.c' from the file system?

Running C Applications

- ❑ Adding Math Library
  - Project must have ***'–lm library'*** to use ***'math.h'*** header file.
  - Right-click ***'your project name'*** and then click ***'Properties'***



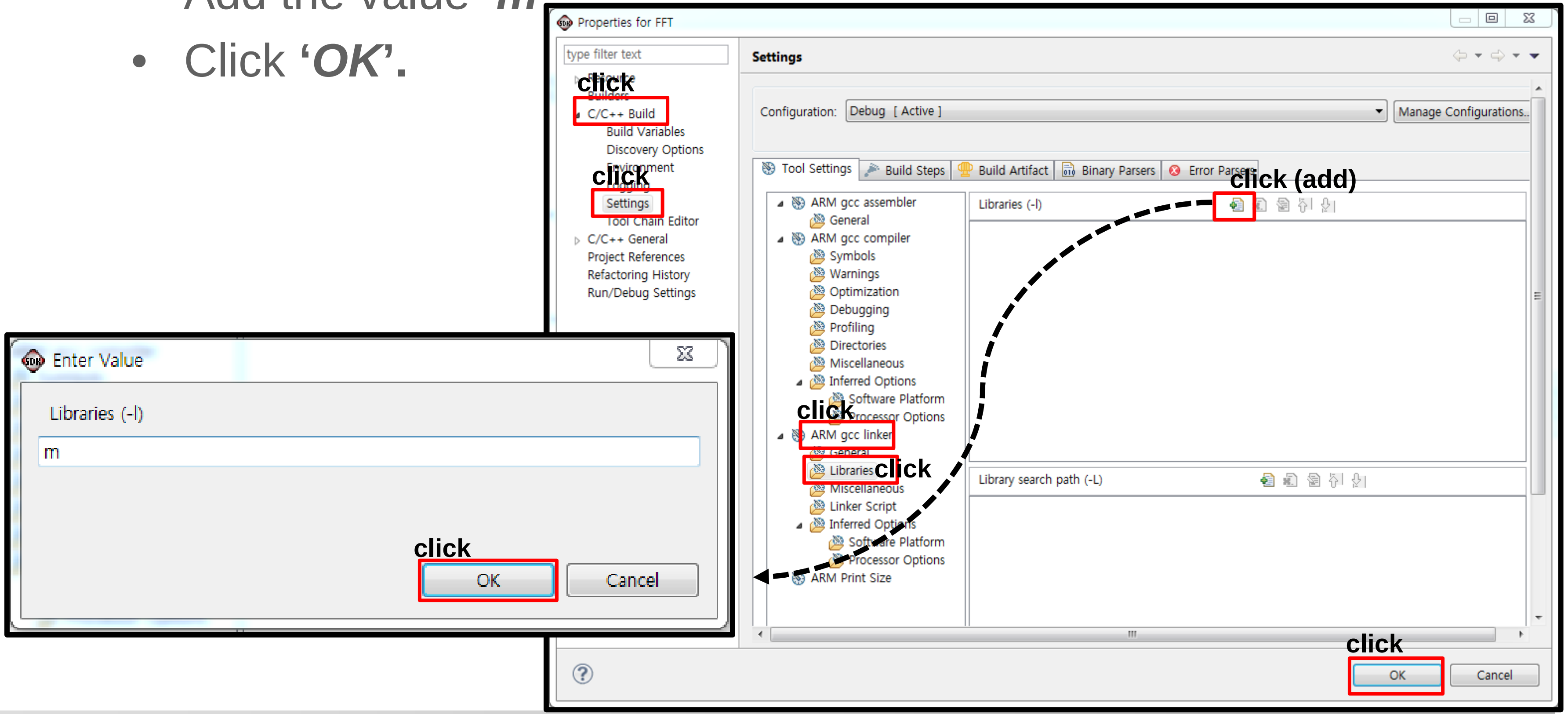
```
for(n=0;n<N;n+
```

KU KONKUK UNIVERSITY
System-on-a-Chip Design LAB

Running C Applications

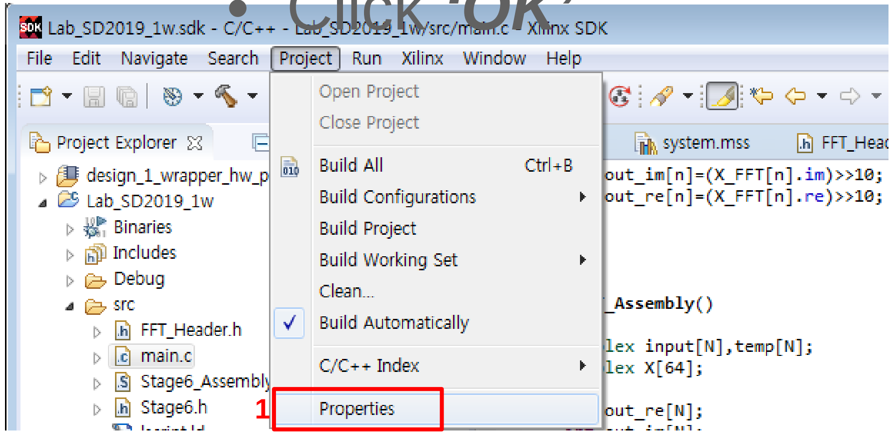
❑ Adding Math Library (Cont'd)

- Click ***'C/C++ Build*** > ***Settings*** > ***ARM gcc linker*** > ***libraries*** > ***add'***
- Add the value ***'m'***
- Click **'*OK*'.**

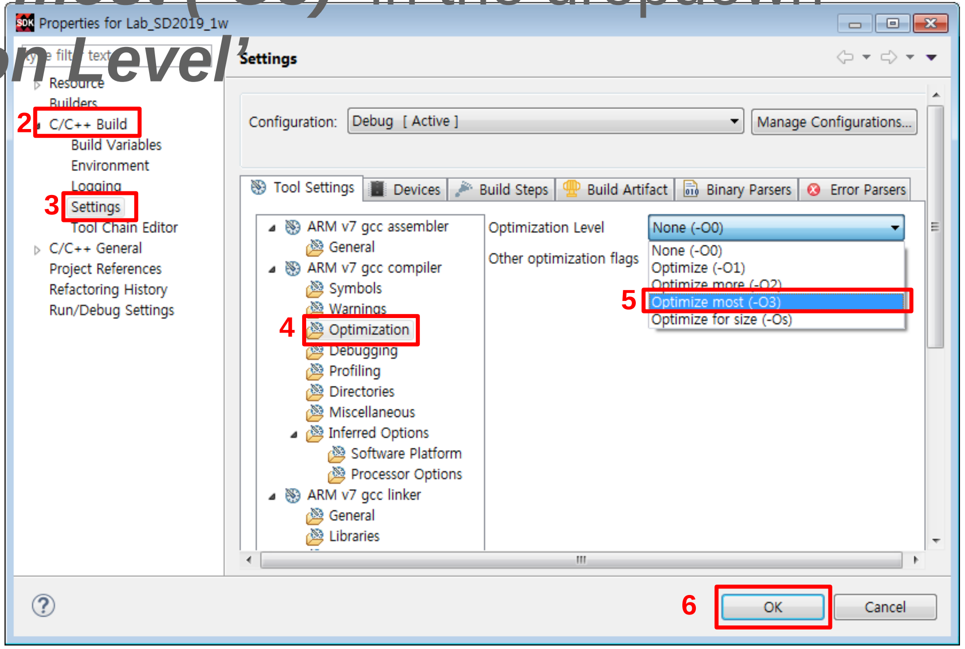


Running C Applications

- Set the compiler optimization level
  - Select ***'Project'*** menu and click ***'Properties'***
  - Select ***'Settings'*** tap and click ***'ARM v7 gcc compiler > Optimization'***
  - Select ***'Optimization most (-O3)'*** in the dropdown menu of ***'Optimization Level'***
  - Click ***'OK'***

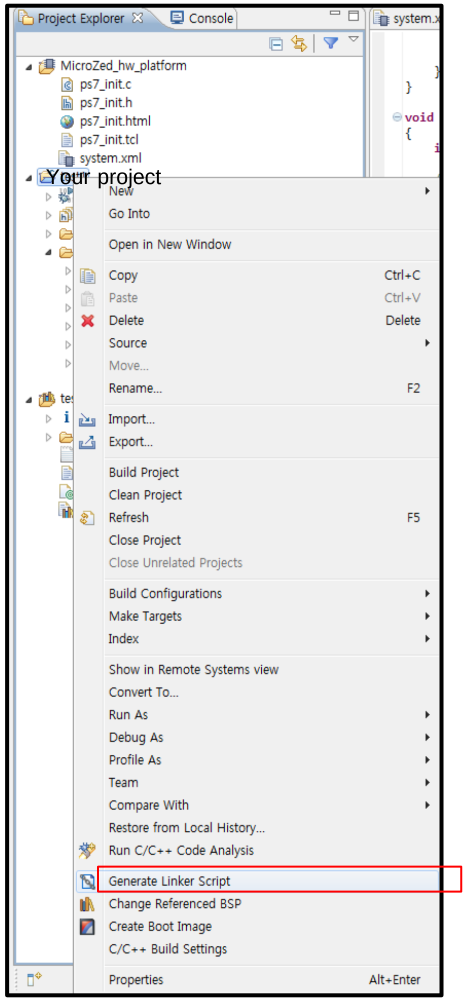


```
out_im[n]=(X_FFT[n].im)>>10;
out_re[n]=(X_FFT[n].re)>>10;
_Assembly()
lex input[N],temp[N];
lex X[64];
out_re[N];
```



Running C Applications

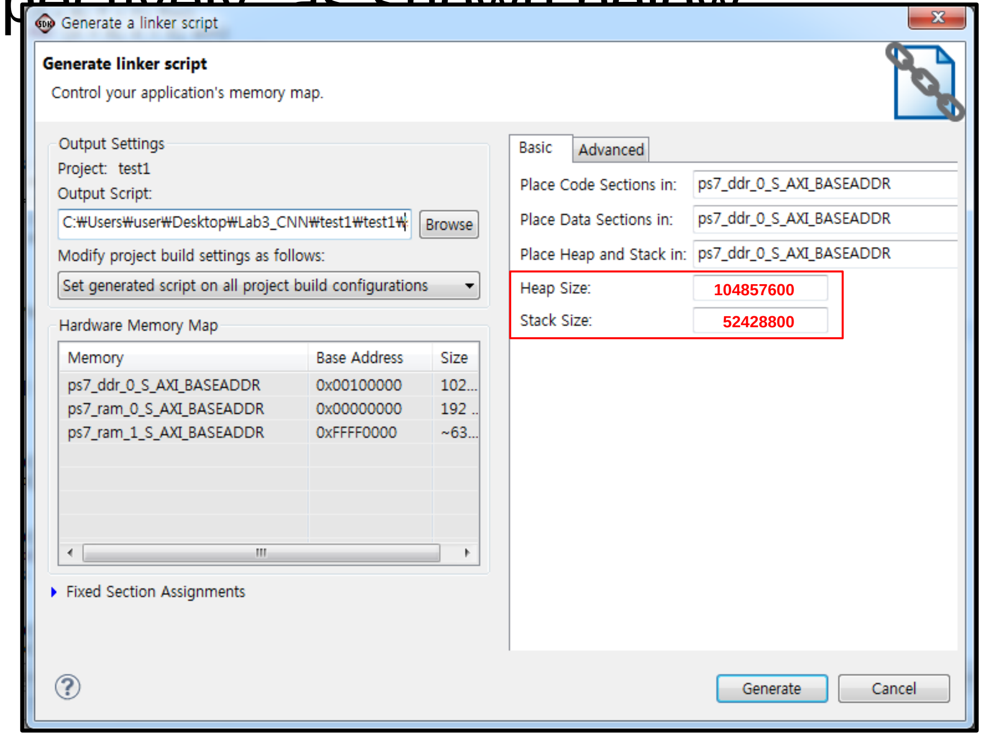
- ❑ Setting Stack & Heap Size
  - Select the application project in the ***Project Explorer*** or ***C/C++ Projects*** view
  - Right-click ***'your project name'*** and then click '***Generate Linker Script'*** or click '**Xilinx Tools > Generate Linker script'**



Running C Applications

❑ Setting Stack & Heap Size (Cont'd)

- Set both the heap and stack sizes in the ***Basic*** tab to ***104857600*** (100 MB) and ***52428800*** (50 MB) respectively, as shown below



Generate linker script
Control your application's memory map.

Output Settings
Project: test1
Output Script:
C:₩Users₩user₩Desktop₩Lab3_CNN₩test1₩test1₩
Modify project build settings as follows:
Set generated script on all project build configurations

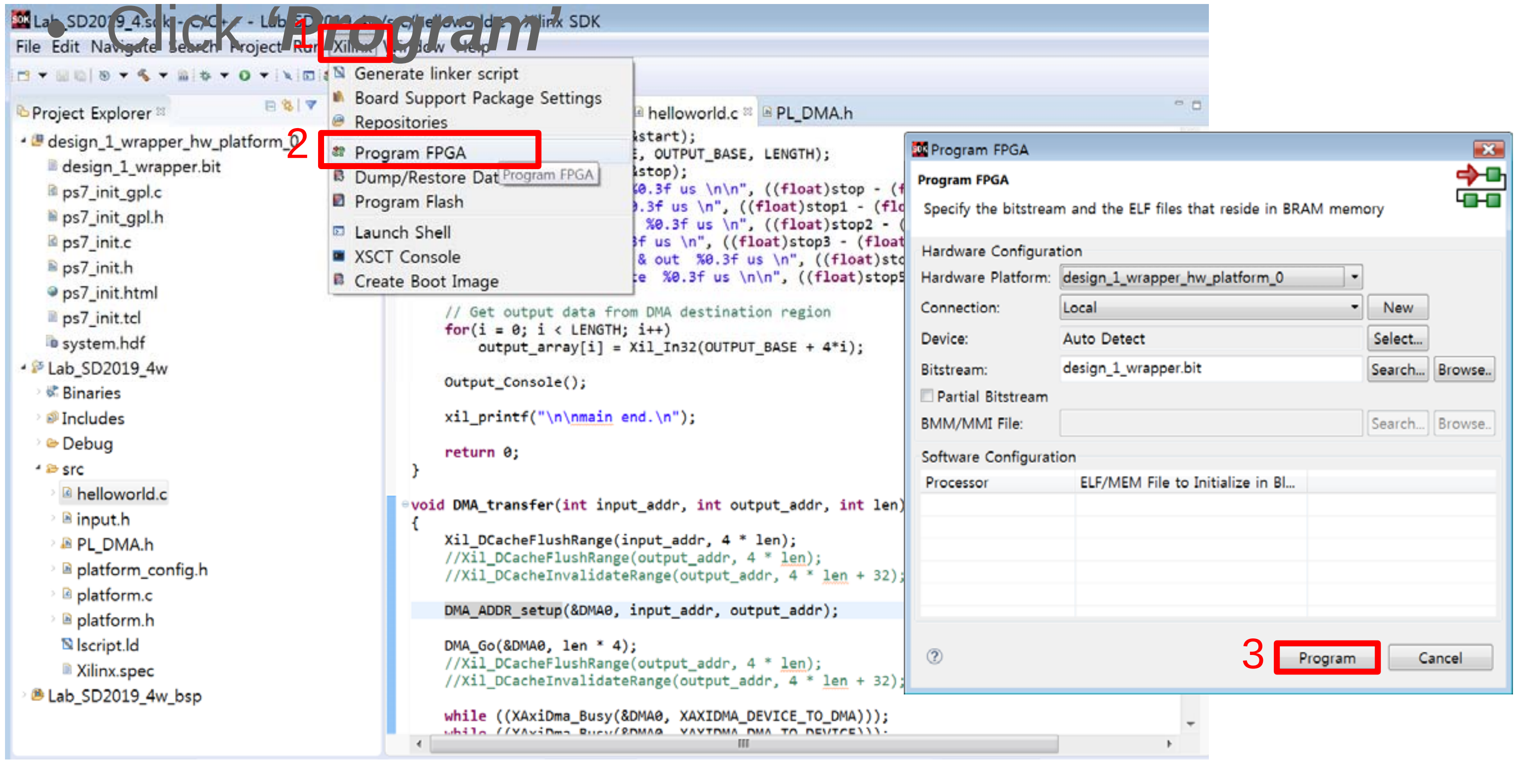
Hardware Memory Map

Memory	Base Address	Size
ps7_ddr_0_S_AXI_BASEADDR	0x00100000	102...
ps7_ram_0_S_AXI_BASEADDR	0x00000000	192 ..
ps7_ram_1_S_AXI_BASEADDR	0xFFFF0000	~63...

Place Code Sections in:	ps7_ddr_0_S_AXI_BASEADDR
Place Data Sections in:	ps7_ddr_0_S_AXI_BASEADDR
Place Heap and Stack in:	ps7_ddr_0_S_AXI_BASEADDR
Heap Size:	104857600
Stack Size:	52428800

Running C Applications

- Program FPGA
  - Choose the ***'Xilinx'*** menu and then click ***'Program FPGA'***
  - Click ***'Program'***



```
&start);
E, OUTPUT_BASE, LENGTH);
&stop);
%0.3f us \n\n", ((float)stop - (f
).3f us \n", ((float)stop1 - (flo
 %0.3f us \n", ((float)stop2 - (
3f us \n", ((float)stop3 - (float
& out  %0.3f us \n", ((float)sto
te  %0.3f us \n\n", ((float)stop5

    // Get output data from DMA destination region
    for(i = 0; i < LENGTH; i++)
        output_array[i] = Xil_In32(OUTPUT_BASE + 4*i);

    Output_Console();

    xil_printf("\n\nmain end.\n");

    return 0;
}

void DMA_transfer(int input_addr, int output_addr, int len)
{
    Xil_DCacheFlushRange(input_addr, 4 * len);
    //Xil_DCacheFlushRange(output_addr, 4 * len);
    //Xil_DCacheInvalidateRange(output_addr, 4 * len + 32);

    DMA_ADDR_setup(&DMA0, input_addr, output_addr);

    DMA_Go(&DMA0, len * 4);
    //Xil_DCacheFlushRange(output_addr, 4 * len);
    //Xil_DCacheInvalidateRange(output_addr, 4 * len + 32);

    while ((XAxiDma_Busy(&DMA0, XAXIDMA_DEVICE_TO_DMA)));
```

Program FPGA

Specify the bitstream and the ELF files that reside in BRAM memory

Hardware Configuration

Hardware Platform:	design_1_wrapper_hw_platform_0		
Connection:	Local	New	
Device:	Auto Detect	Select...	
Bitstream:	design_1_wrapper.bit	Search...	Browse...
☐ Partial Bitstream			
BMM/MMI File:		Search...	Browse...

Software Configuration

Processor	ELF/MEM File to Initialize in Bl...

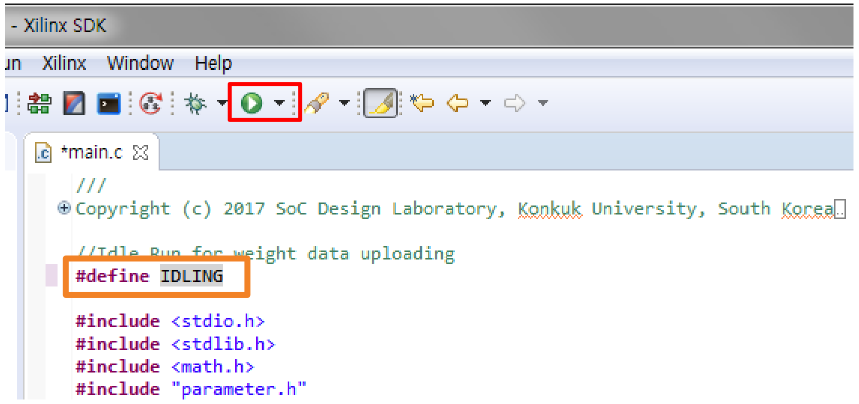
Running C Applications

❑ Check the source code: **main.c**

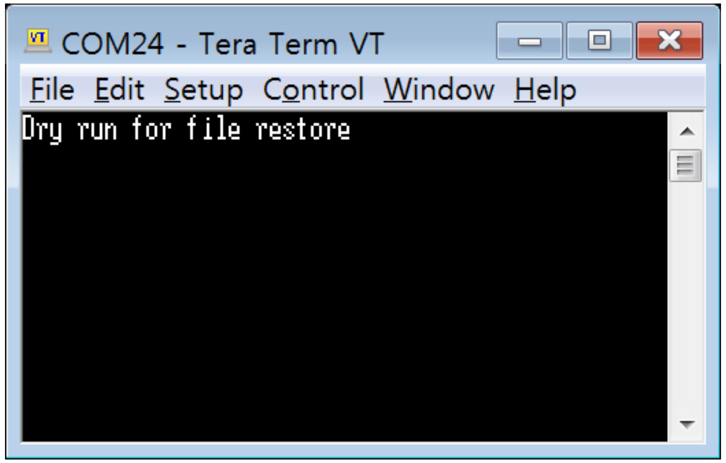
- Make sure that **'#define IDLING'** is written at the top of the source code

❑ Run the application

- This application is simply for uploading the convolution weights into the DDR



```
///
Copyright (c) 2017 SoC Design Laboratory, Konkuk University, South Korea

//Idle Run for weight data uploading
#define IDLING

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "parameter.h"
```

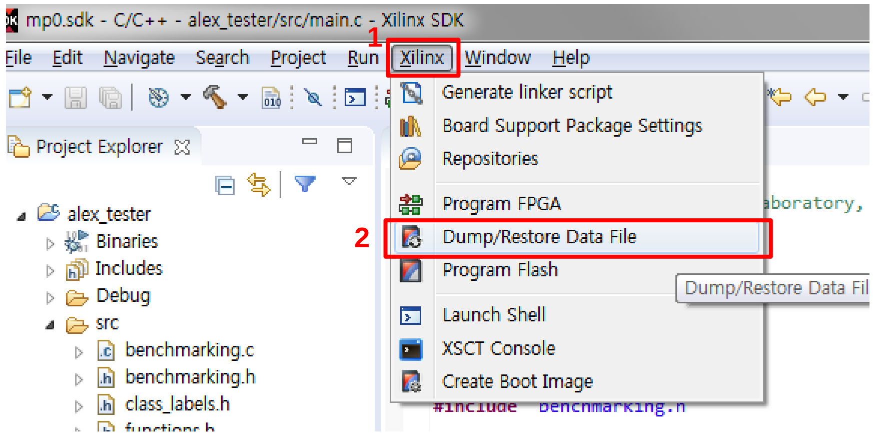


```
Dry run for file restore
```

Running C Applications

❑ Restore Memory

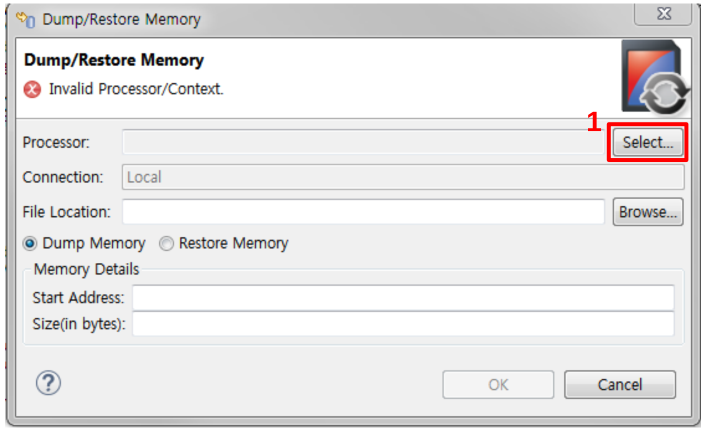
- Click ‘***Xilinx > Dump/Restore Data File***’



Running C Applications

❑ Restore Memory (Cont'd)

- Click '***Select***'
- Select '***Name=Xilinx Hardware Server***'
- Select '***ARM Cortex-A9 MPCore #0***'
- Click '***OK***'



Dump/Restore Memory

Dump/Restore Memory

Invalid Processor/Context.

Processor: [Select...]

Connection: Local

File Location: [Browse...]

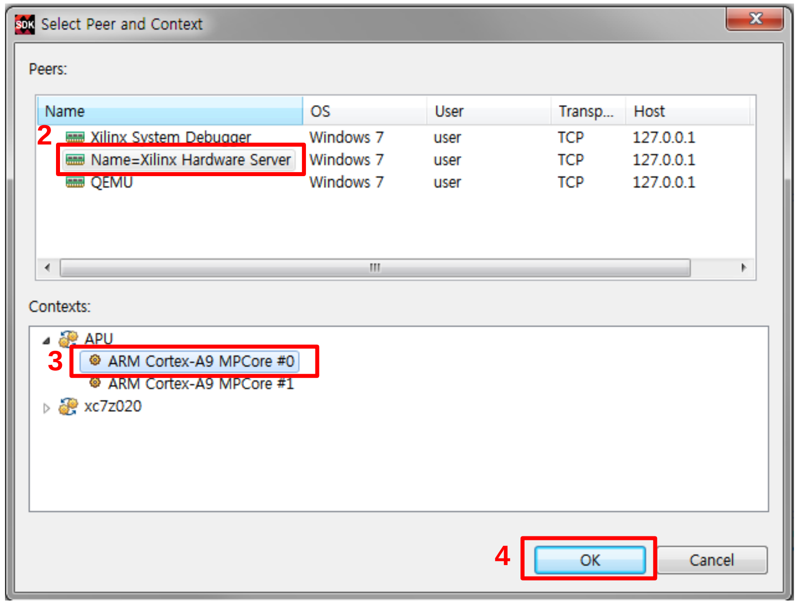
[● Dump Memory] ○ Restore Memory

Memory Details

Start Address:

Size(in bytes):

OK Cancel



Select Peer and Context

Peers:

Name	OS	User	Transp...	Host
Xilinx System Debugger	Windows 7	user	TCP	127.0.0.1
Name=Xilinx Hardware Server	Windows 7	user	TCP	127.0.0.1
QEMU	Windows 7	user	TCP	127.0.0.1

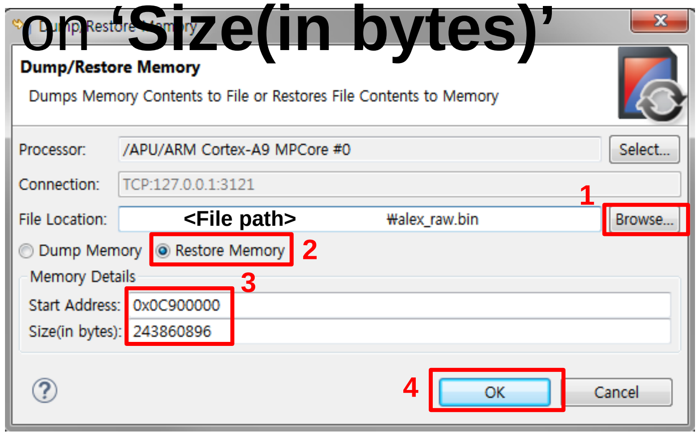
Contexts:

- APU
  - ARM Cortex-A9 MPCore #0
  - ARM Cortex-A9 MPCore #1
- xc7z020

OK Cancel

Running C Applications

- ❑ Restore Memory (Cont'd)
  - Click '***Browse***' and select file **alex_raw.bin** Download link: https://drive.google.com/open?id=122wv9t3FVrXlQYgc9bk7eK_L5-Q8ansA
  - Select ***'Restore Memory'***
  - Type **0x0C900000** on **'Start Address'** and **243860896** on **'Size(in bytes)'**


Dump/Restore Memory

Dump/Restore Memory

Dumps Memory Contents to File or Restores File Contents to Memory

Processor: /APU/ARM Cortex-A9 MPCore #0

Connection: TCP:127.0.0.1:3121

File Location: <File path> ₩alex_raw.bin

Dump Memory [✓ Restore Memory]

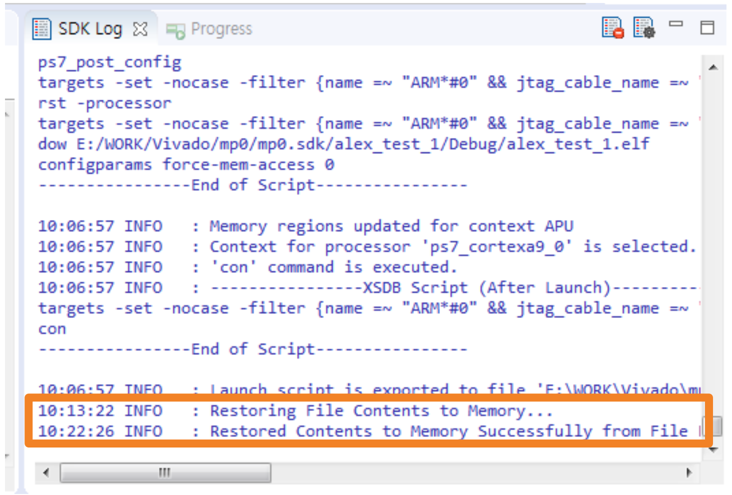
Memory Details

Start Address: 0x0C900000

Size(in bytes): 243860896

Running C Applications

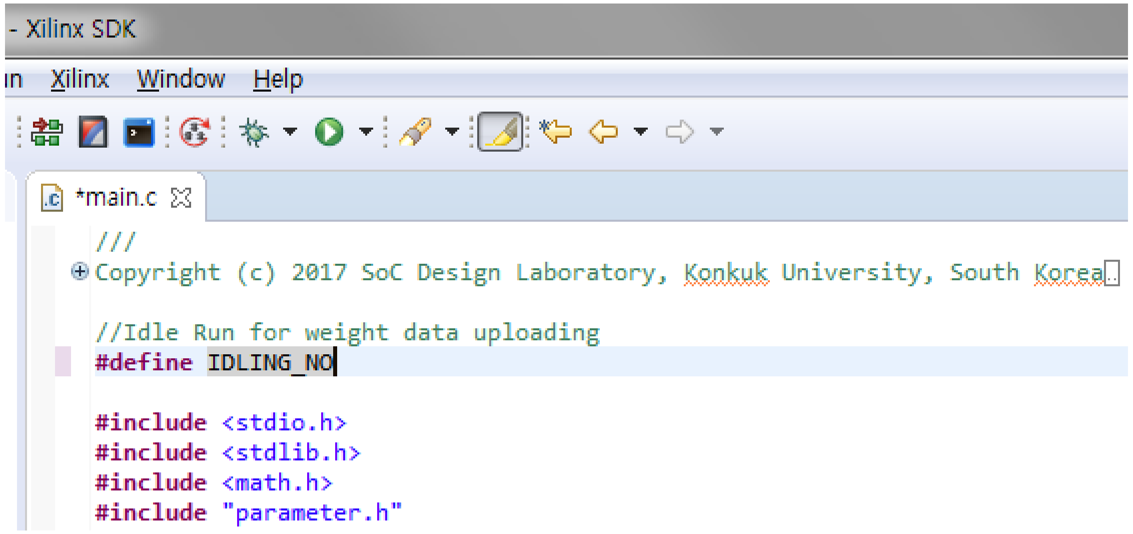
- ❑ Restore Memory (Cont'd)
  - Check the SDK log window (which may take about 9 minutes) to see the message ***'Restored Contents to Memory Successfully from File ~'***



```
ps7_post_config
targets -set -nocase -filter {name =~ "ARM*#0" && jtag_cable_name =~
rst -processor
targets -set -nocase -filter {name =~ "ARM*#0" && jtag_cable_name =~
dow E:/WORK/Vivado/mp0/mp0.sdk/alex_test_1/Debug/alex_test_1.elf
configparams force-mem-access 0
----------------End of Script----------------

10:06:57 INFO   : Memory regions updated for context APU
10:06:57 INFO   : Context for processor 'ps7_cortexa9_0' is selected.
10:06:57 INFO   : 'con' command is executed.
10:06:57 INFO   : ----------------XSDB Script (After Launch)---------
targets -set -nocase -filter {name =~ "ARM*#0" && jtag_cable_name =~
con
----------------End of Script----------------

10:06:57 INFO   : Launch script is exported to file 'E:\WORK\Vivado\m
10:13:22 INFO   : Restoring File Contents to Memory...
10:22:26 INFO   : Restored Contents to Memory Successfully from File
```

Running C Applications

- ❑ Modify the source code: ***main.c***
  - Change **‘#define IDLING’** to **‘#define IDLING_NO’** (or anything different from **‘IDLING’**)
- ❑ Run the application again
  - This application is for actually running the CNN al

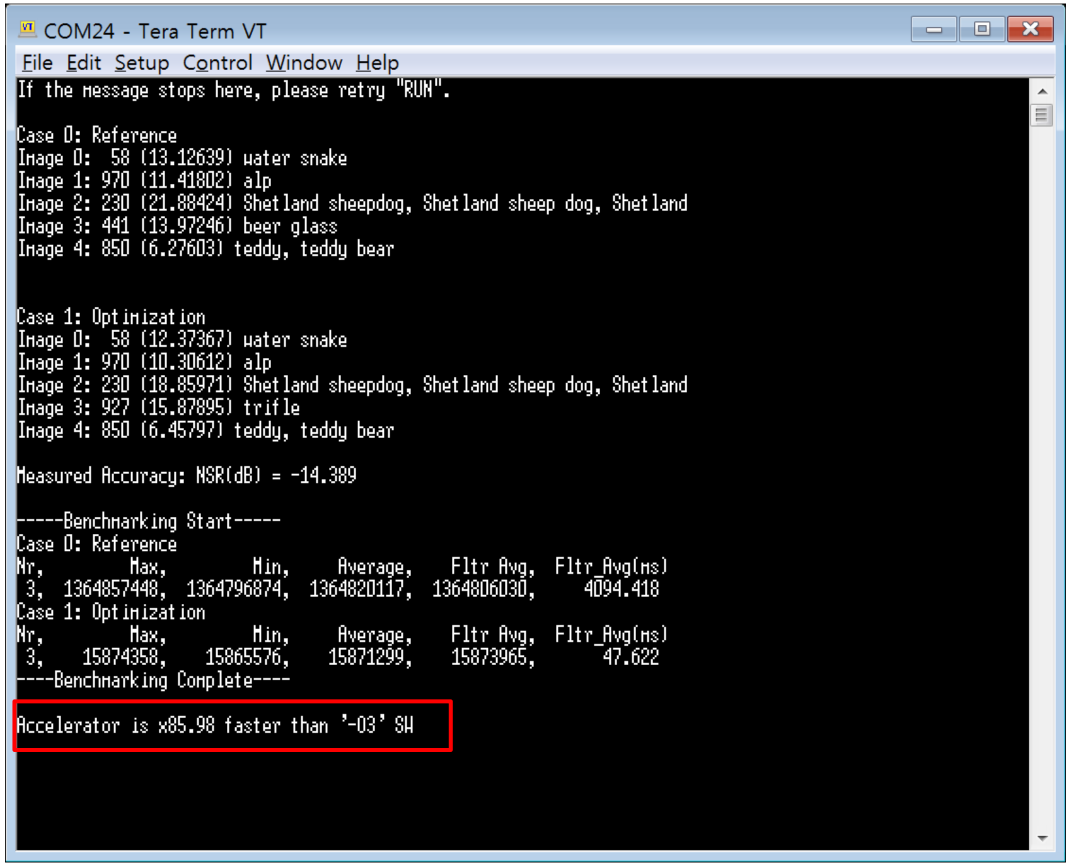


```
///
Copyright (c) 2017 SoC Design Laboratory, Konkuk University, South Korea

//Idle Run for weight data uploading
#define IDLING_NO

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "parameter.h"
```

Running C Applications

❑ Run the application again (Cont'd)

- Check the output of the application on ***'Tera Term'***



```
If the message stops here, please retry "RUN".

Case 0: Reference
Image 0:  58 (13.12639) water snake
Image 1: 970 (11.41802) alp
Image 2: 230 (21.88424) Shetland sheepdog, Shetland sheep dog, Shetland
Image 3: 441 (13.97246) beer glass
Image 4: 850 (6.27603) teddy, teddy bear


Case 1: Optimization
Image 0:  58 (12.37367) water snake
Image 1: 970 (10.30612) alp
Image 2: 230 (18.85971) Shetland sheepdog, Shetland sheep dog, Shetland
Image 3: 927 (15.87895) trifle
Image 4: 850 (6.45797) teddy, teddy bear

Measured Accuracy: NSR(dB) = -14.389

-----Benchmarking Start-----
Case 0: Reference
Nr,         Max,         Min,     Average,    Fltr Avg,  Fltr_Avg(ms)
 3,  1364857448,  1364796874,  1364820117,  1364806030,     4094.418
Case 1: Optimization
Nr,         Max,         Min,     Average,    Fltr Avg,  Fltr_Avg(ms)
 3,    15874358,    15865576,    15871299,    15873965,       47.622
----Benchmarking Complete----

Accelerator is x85.98 faster than '-O3' SW
```