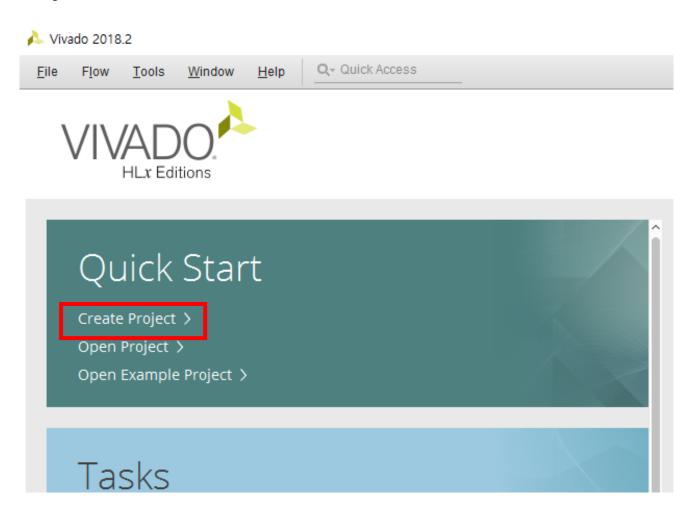
Practice 3

- How to use Vivado & Basic syntax

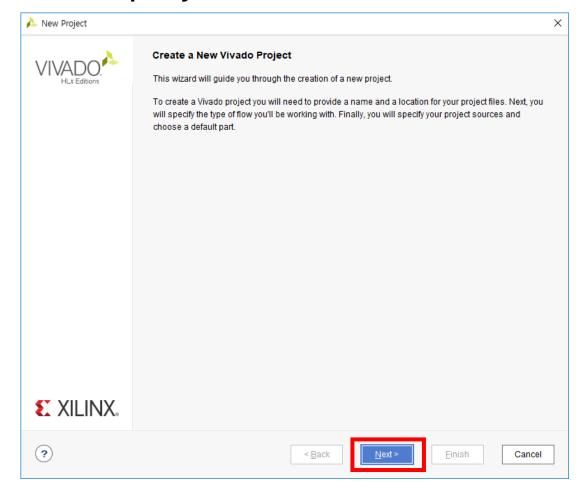
Computing Memory Architecture Lab.

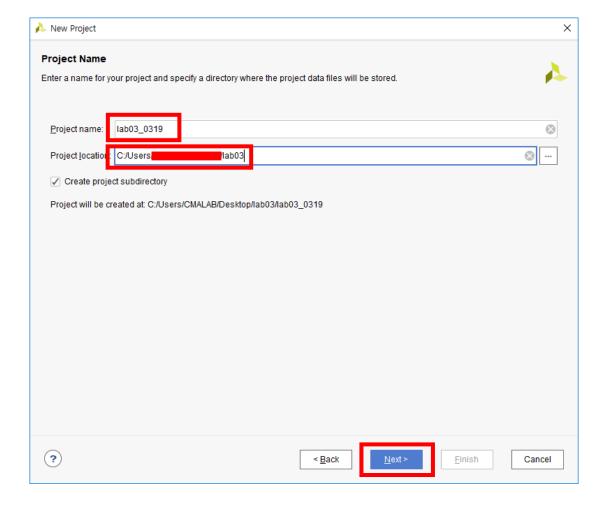
Vivado Tutorial

Create New Project

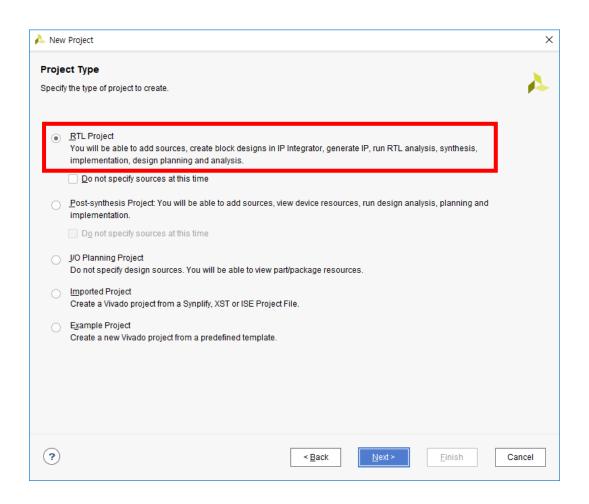


Enter project name and location





- Select project type
 - RTL Project



Copy sources into project

?

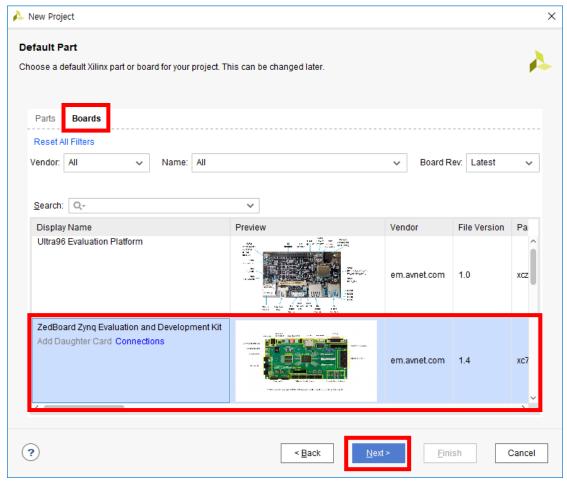
 (Optional) add sources or IPs, constraints A New Project Add Constraints (optional) **Add Sources** Specify or create constraint files for physical and timing constraints. Specify HDL, netlist, Block Design, and IP files, or directories containing those files, to add to your project. Create a new source file on disk and add it to your project. You can also add and create sources later + = 1 1 1 +New Project figurable IP, DSP composite, and Embedded sub-design files to add to your project Use Add Files, Add Directories or Create File buttons below Use Add Files or Create File buttons below Add Files Add Directories Create File Scan and add RTL include files into project Use Add Files or Add Directories buttons below ✓ Copy sources into project Add Files Create File ✓ Add sources from subdirectories ✓ Copy constraints files into project Target language: Verilog 🗸 Simulator language: Mixed ? (?) <u>F</u>inish Cancel Finish Cancel Add Files Add Directories

< 뒤로(<u>B</u>)

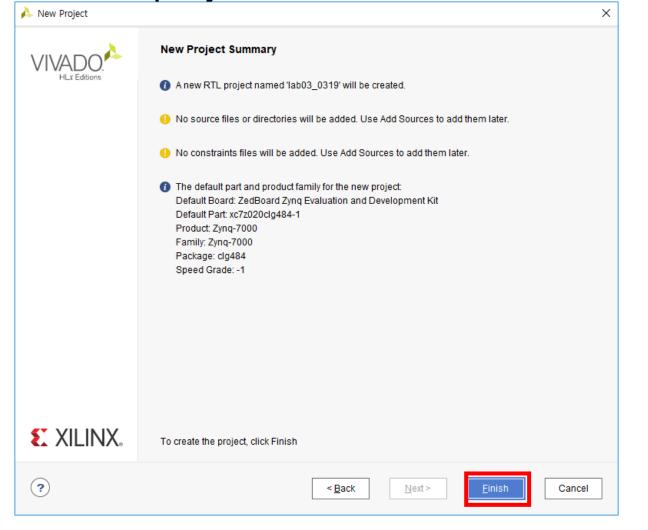
Next>

Cancel

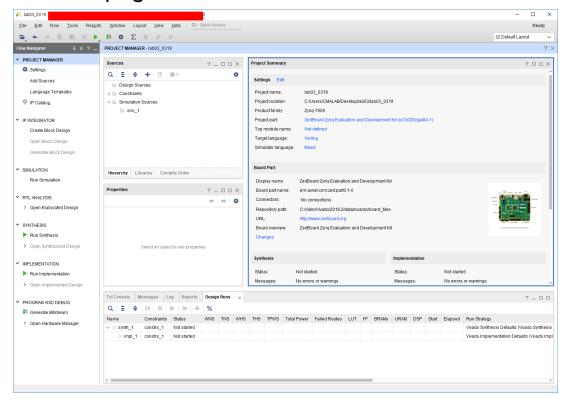
- Choose part or board
 - We are going to use ZedBoard



Finish project creation

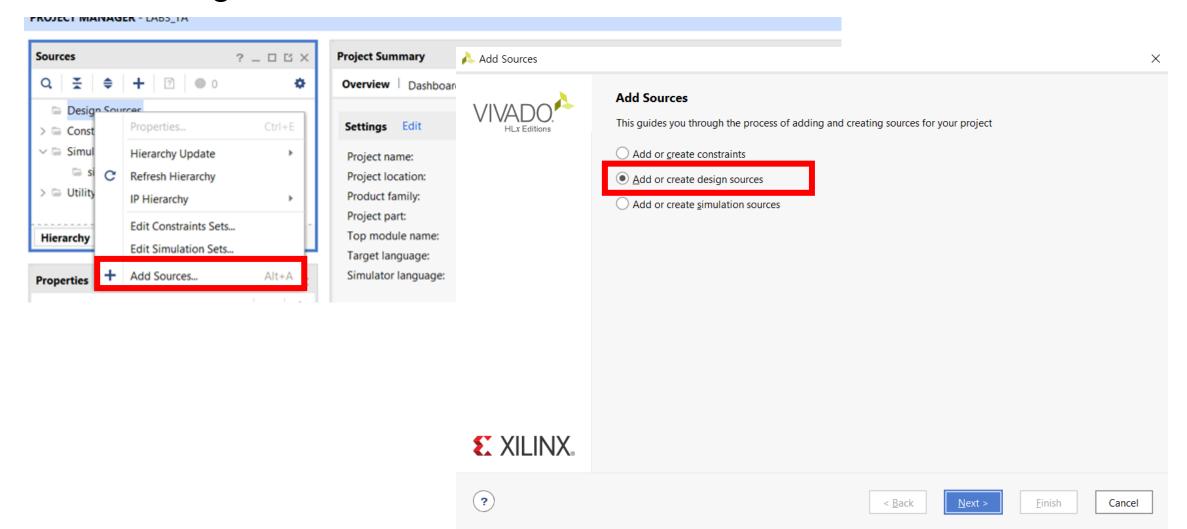


Main page



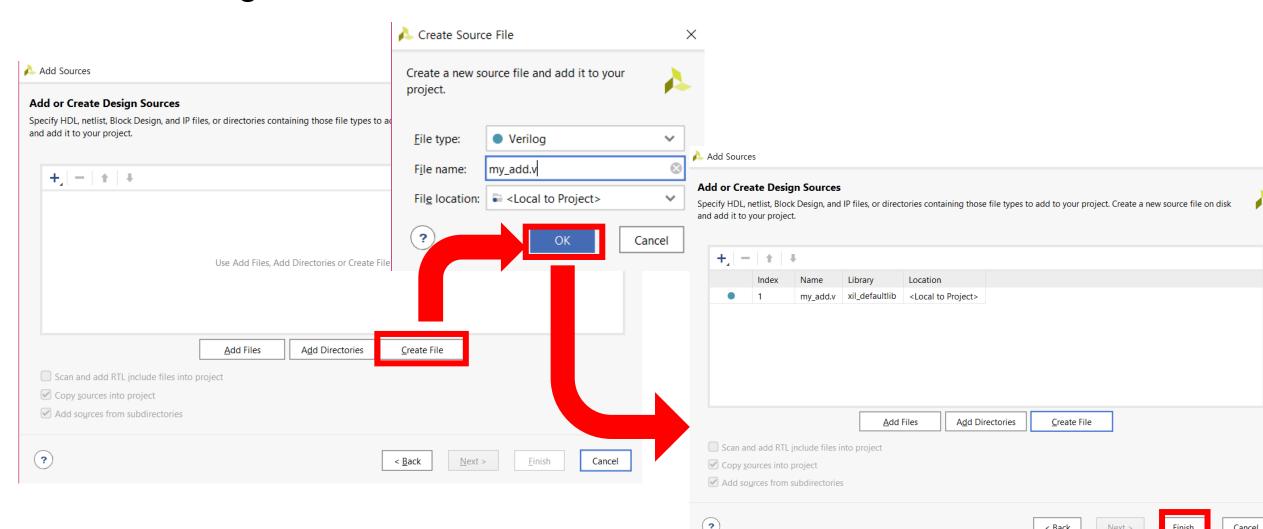
Create design sources

Create design source

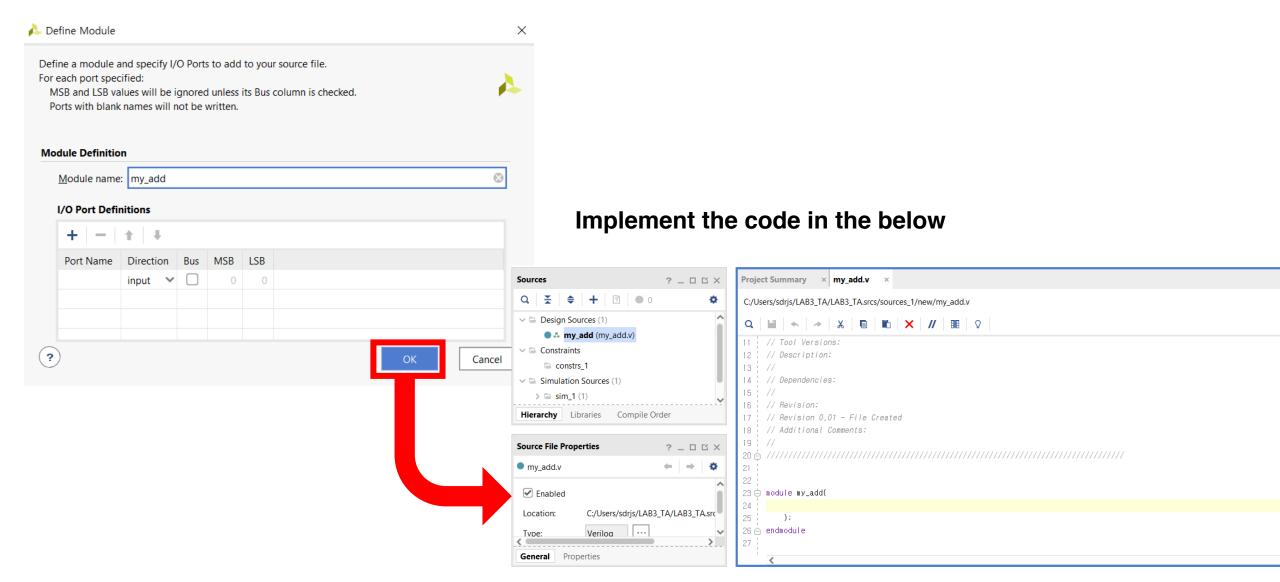


Create design sources

Create design source

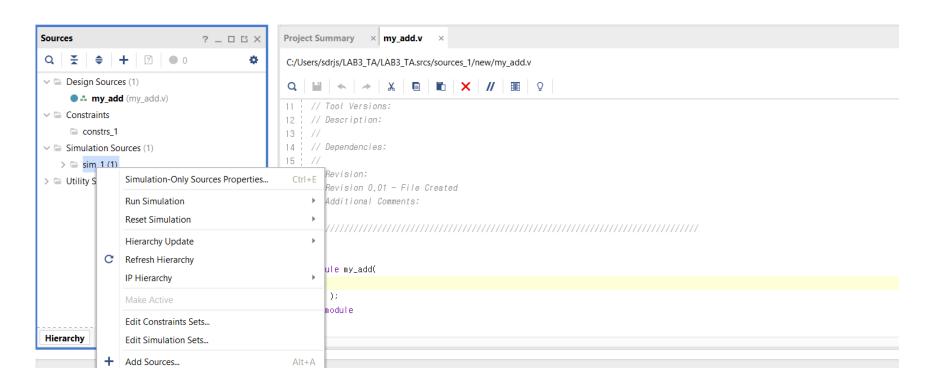


Create design sources



Create simulation sources

Create simulation source



Create simulation sour Add or Create Simulation Sources Specify simulation specific HDL files, or directories containing HDL files, to add to your project. Create a new source file on disk and add it to ~ Specify simulation set: sim_1 Create simulation source Library Location Modules and testbenches xil_defaultlib <Local to Project> Add Files Add Directories Create File Scan and add RTL include files into project ✓ Copy sources into project Add sources from subdirectories ✓ Include all design sources for simulation Add Sources **Add or Create Simulation Sources** ? Specify simulation specific HDL files, or directories containing HDL files, to add to your project. Create a new source file on disk and add it to your project. Specify simulation set: | = sim_1 A Create Source File X A Define Module +, - | 1 | 1 Create a new source file and add it to your Define a module and specify I/O Ports to add to your source file. project. For each port specified: MSB and LSB values will be ignored unless its Bus column is checked. Ports with blank names will not be written. Verilog File type: File name: Module Definition Module name: tb File location: - <Local to Project> I/O Port Definitions Cancel + - + + Scan and add RTL include files into pro Port N... Directi... Copy sources into project input V 0 0 Add sources from subdirectories ✓ Include all design sources for simulation

Cancel

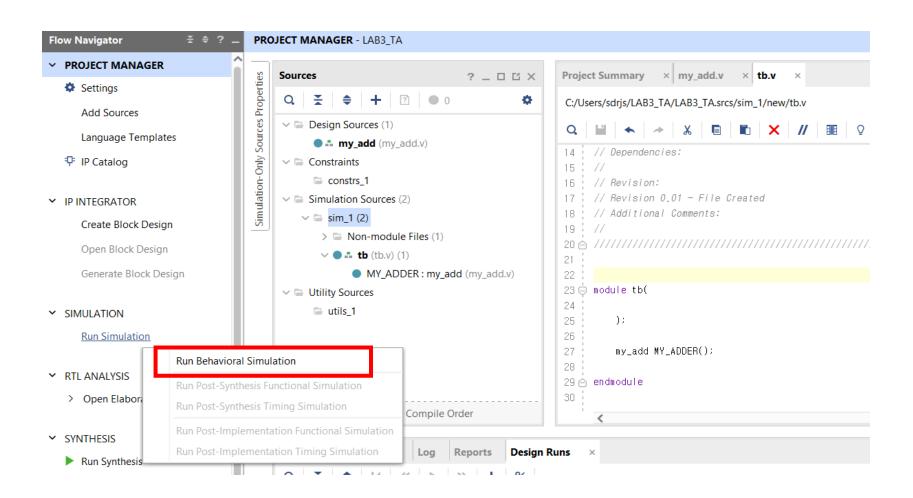
< Back

Next >

(?

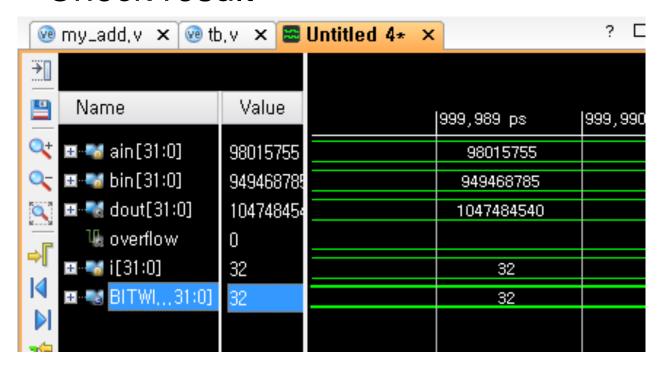
Simulation

Run Simulation -> Run Behavioral Simulation



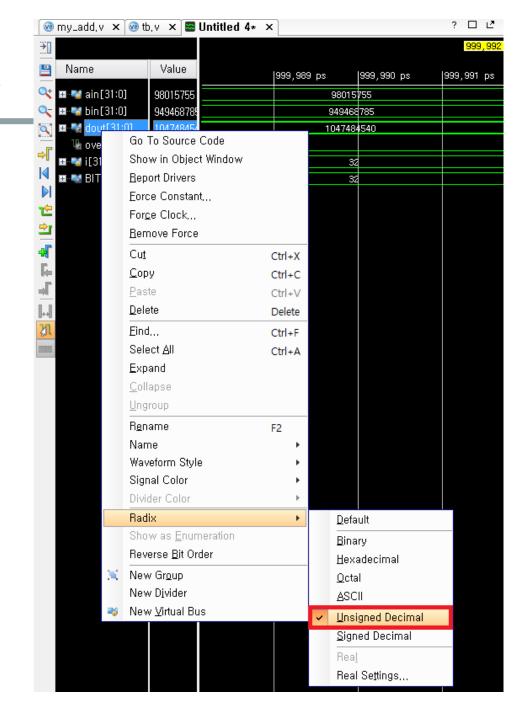
Simulation results

Check result



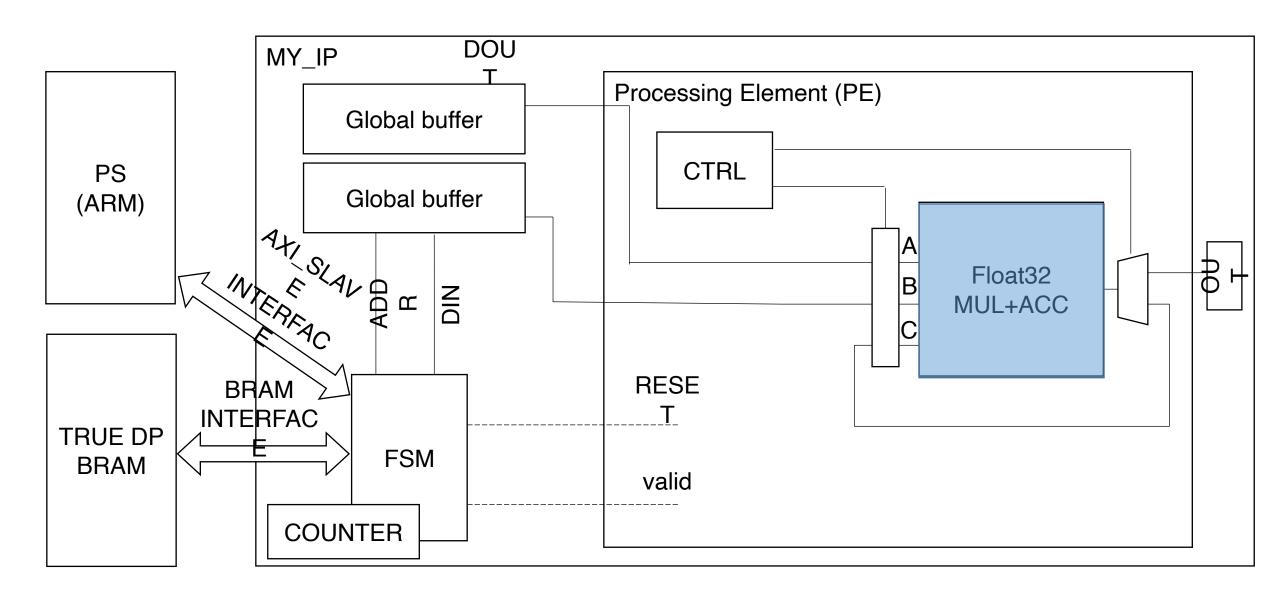
Simulation results

Change radix

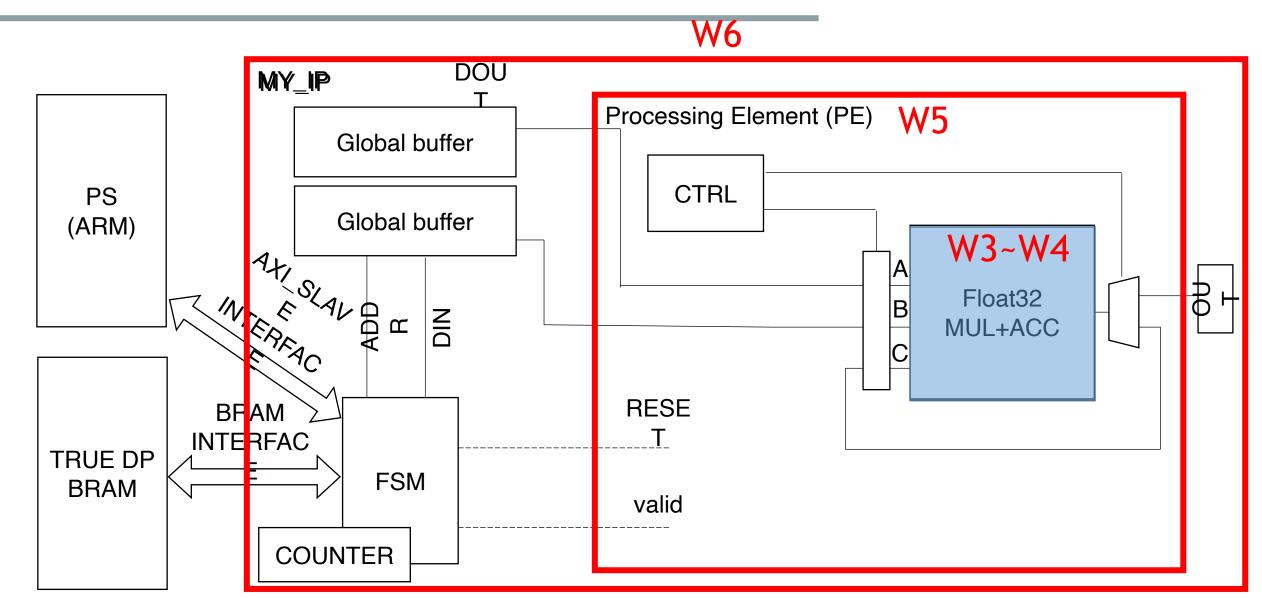


Main Practice

Final Project Overview: Matrix Multiplication IP



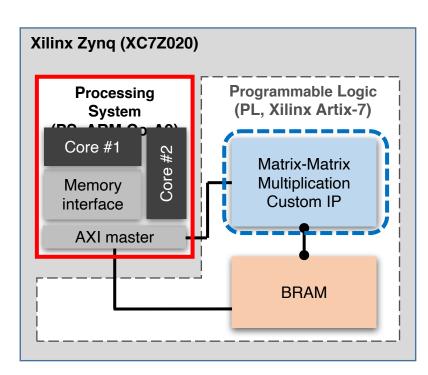
Final Project Overview: Matrix Multiplication IP



Final Project Overview: Matrix Multiplication IP

- CNN is our application
 - Convolution layer becomes a matrix-matrix multiplication after convolution lowering e.g., 32x64 matrix * 64x64 matrix → 32x64 matrix
- ARM CPU runs the main function which calls your MM IP on PL
 - MM for 32x64 weight matrix * 64x64 input matrix multiplication
- BRAM is used for data transfer between SW and HW

MM function on Hardware (Software running on CPU)



Practice

Implement following three combinational blocks and test them (i.e., show wave form). Design your own test bench for your blocks (test at least 32 vectors). You should follow given input and output declaration format.

- 1. Design n-bit integer adder (default 32 bit)
- 2. Design n-bit integer multiplier (default 32 bit)
- 3. Design n-bit integer fused multiplier (default 32 bit)

n-bit Integer Adder (default 32 bit)

```
module my add #(
      parameter BITWIDTH = 32
       input [BITWIDTH-1:0] ain,
       input [BITWIDTH-1:0] bin,
       output [BITWIDTH-1:0] dout,
       output overflow
   IMPLEMENT HERE! */
endmodule
```

- ain: 1st operand (unsigned)
- bin: 2nd operand (unsigned)
- dout: summation result
- overflow:
 - ==1: if overflow is detected
 - ==0: otherwise.

n-bit Integer Multiplier (default 32 bit)

```
module my mul #(
       parameter BITWIDTH = 32
       input [BITWIDTH-1:0] ain,
       input [BITWIDTH-1:0] bin,
       output [2*BITWIDTH-1:0] dout
   IMPLEMENT HERE! */
•••
endmodule
```

- **ain:** 1st operand (unsigned)
- bin: 2nd operand (unsigned)
- dout: multiplication result

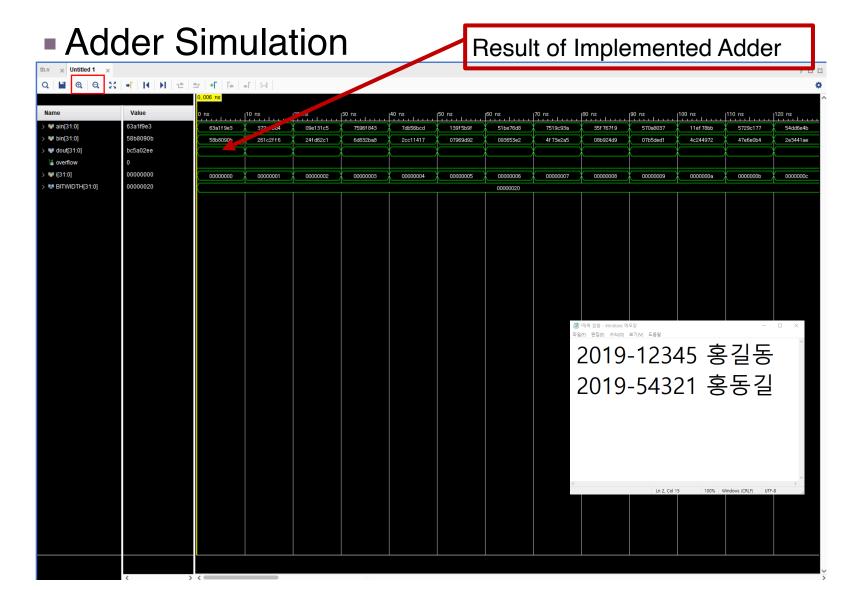
n-bit Integer fused multiplier (default 32 bit)

```
module my_fusedmult #(
       parameter BITWIDTH = 32
       input [BITWIDTH-1:0] ain,
       input [BITWIDTH-1:0] bin,
       input en,
       input clk,
       output [2*BITWIDTH-1:0] dout
   IMPLEMENT HERE! */
endmodule
```

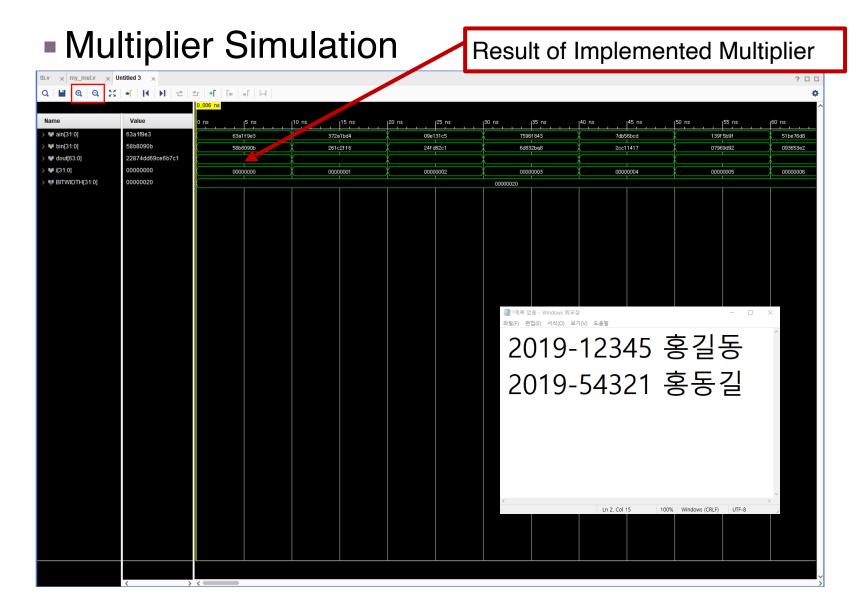
- ain: 1st operand (unsigned)
- bin: 2nd operand (unsigned)
- dout: multiplication and accumulation result
- en:
 - ==1: fused multiplier computes output
 - ==0: fused multiplier initialize output as 0
- clk: clock

Requirements

- Result
 - Attach your project folder with all your Verilog codes (e.g., Adder, Multiplier, fused-multiplier, testbench)
 - Attach your waveform(simulation result) with [student_number, name]
 - Refer to slide 21-23 for details
 - Use the test-bench provided
- Report
 - Explain Adder, Multiplier, fused-multiplier that you implemented
 - In your own words
 - Either in Korean or in English
 - # of pages does not matter
 - PDF file name: Week3_2020_12345_홍길동.pdf
 - PDF only!!
- Result + Report to one .zip file
- Upload (.zip) file on ETL
 - Submit one (.zip) file Week3_2020_12345_홍길동.zip
 - zip file name :
 - Due: 3/24(WED) 23:59
 - No Late Submission

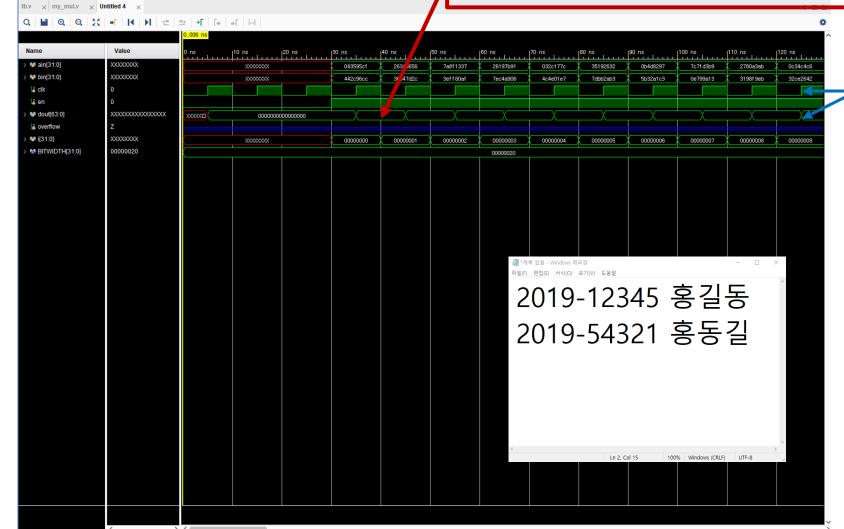


- dout = ain + bin
- Match time-scale using +/- buttons



- dout = ain * bin
- Match time-scale using +/- buttons

■ Fused-multiplier Simulations of Implemented fused-multiplier



 Every positive clock, fused-multiplier operation

Operation must be executed at positive clock

dout = dout + ain * bin