Lab 2 Practice:

Prerequisities

Concept	Video Lectures	in this lab
Data flow modeling	CT Verilog series 04	Wire, reg, signed, etc
Behavioral modeling	CT Verilog series 05 ~ 06	Always block
Sequential blocks	CT Verilog series 07	Synchronous output with clock

It is strongly recommended to complete the practice exercises to be well-prepared for the basic lab.

1. ALU Design

Design a synchronous ALU (Arithmetic Logic Unit).

1. To prevent timing issue, you have to add a **flip-flop (FF)** before the output port (the output should be delayed for 1 cycle).
2. Change value at the **positive edge** of each clock cycle and be reset synchronously.
3. Output = 0 when the reset is triggered.
4. A Verilog template is given.

IO List and Specification:

Signals	**I/O**	**Bit Width**	**Desciption**
clk	input	1	Clock (positive-edge triggered)
rst	input	1	Synchronous active-high reset
A	input	8	Signed ALU input
B	input	8	Signed ALU input
ctrl	input	1	ALU control signal
out	output	16	Signed ALU output

Name	**ctrl**	**Function**
Function 1	0	out = A * B
Function 2	1	If (A < B) out = 1 with sign extension else out = -1 with sign extension

Example:

1. (Cycle 1) ctrl = 0, A = 8'd5, B = 8'd4
   (Cycle 2) out = 16'd32
2. (Cycle 1) ctrl = 1, A = 8'd3, B = 8'd10

(Cycle 2) out = 16'd1

Note:

Be aware of the signed numbers.

2. Counter

Implement a specific counter with the following rules.

1. Change value at the **positive edge** of each clock cycle and be reset synchronously.
2. Output = 0 when the reset is triggered.
3. A Verilog template is given.

IO List and Specification:

Signals	**I/O**	**Bit Width**	**Desciption**
clk	input	1	Clock (positive-edge triggered)
rst	input	1	Synchronous active-high reset
out	output	16	The current value of the counter

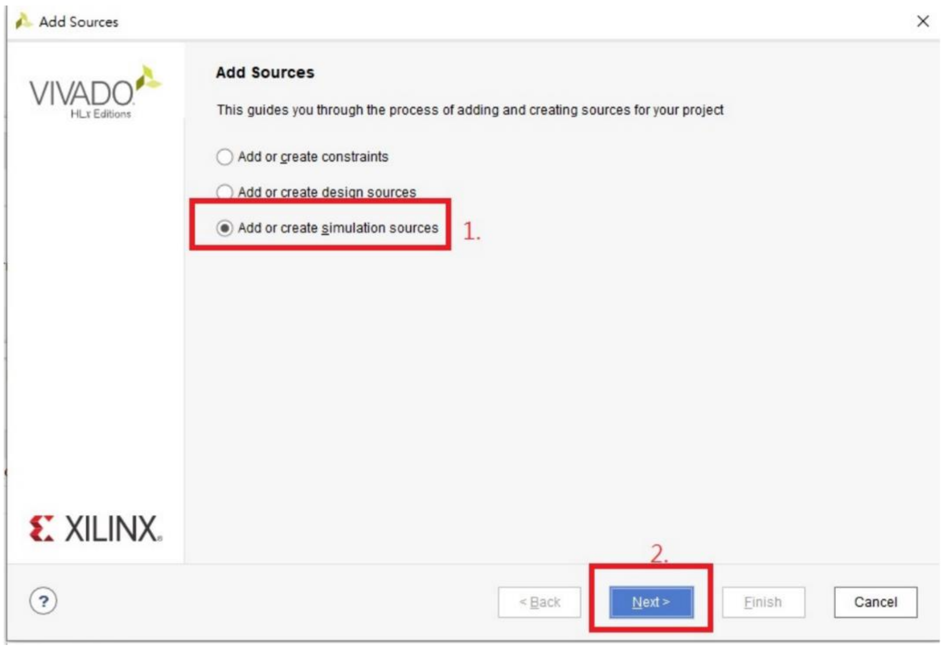
The counter will count upward from 1 until being reset.
Let i denote the i-th step. The counter will follow the rules:

$$\begin{cases} if\ a_i\ is\ odd: a_{i+1} = a_i * 2 \\ if\ a_i\ is\ even: a_{i+1} = a_i + i \end{cases}$$

Your output should look like the following sequences:

i	1	2	3	4	5	6	7	8	9
a_i	1	2	4	7	14	19	38	45	90

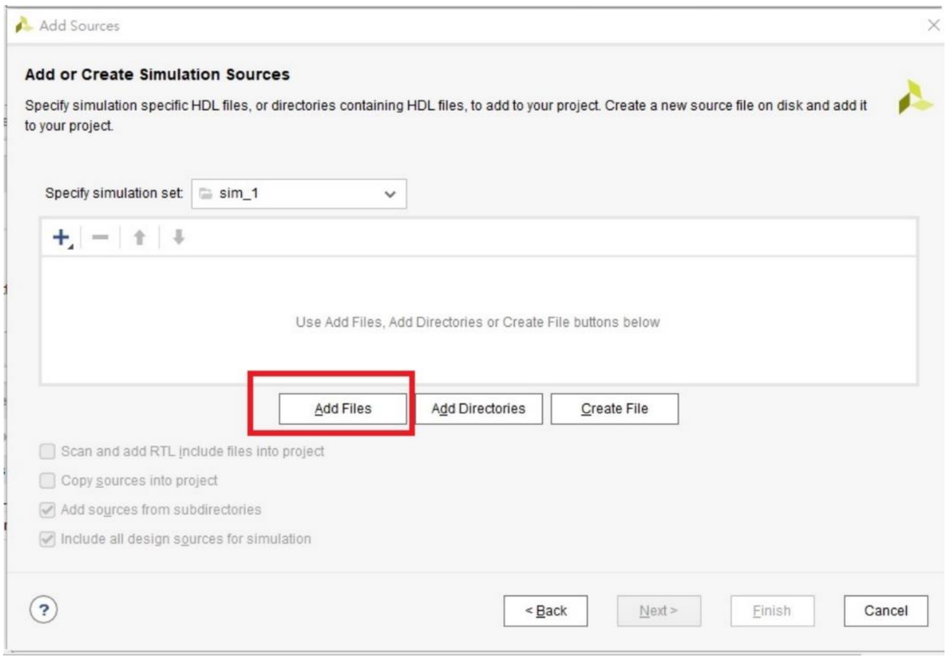
Appendix: How to add pattern.dat to the simulation sources.


Add Sources

Add Sources

This guides you through the process of adding and creating sources for your project

( ) Add or create constraints
( ) Add or create design sources
(●) Add or create simulation sources


Add Sources

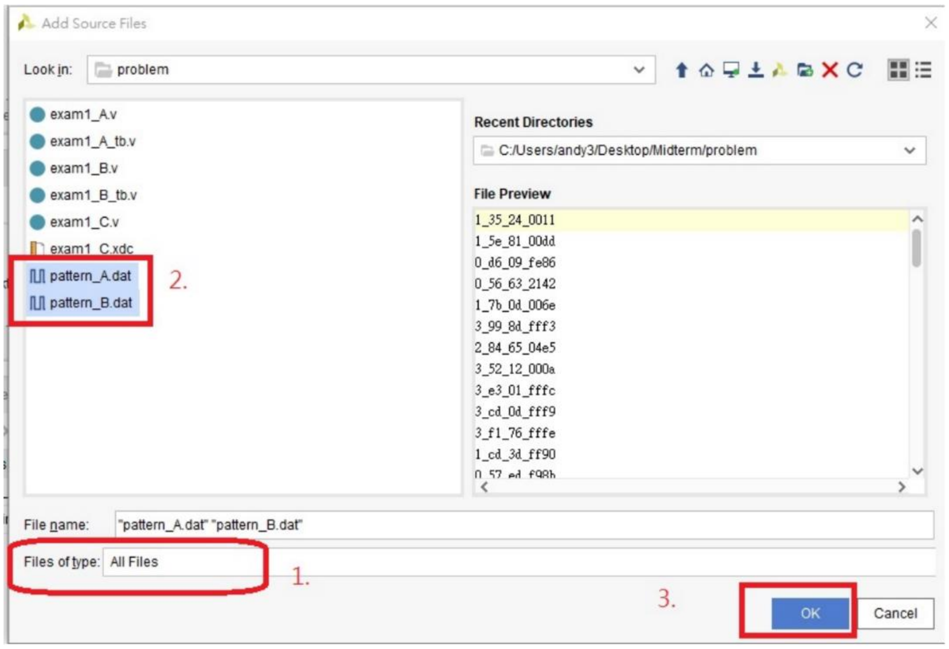
Add or Create Simulation Sources

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

Use Add Files, Add Directories or Create File buttons below

☐ Scan and add RTL include files into project
☐ Copy sources into project
☑ Add sources from subdirectories
☑ Include all design sources for simulation



Add Source Files

Look in: problem

Recent Directories: C:/Users/andy3/Desktop/Midterm/problem

File Preview

```
1_35_24_0011
1_5e_81_00dd
0_d6_09_fe86
0_56_63_2142
1_7b_0d_006e
3_99_8d_fff3
2_84_65_04e5
3_52_12_000a
3_e3_01_fffc
3_cd_0d_fff9
3_f1_76_fffe
1_cd_3d_ff90
0_57_ed_f98b
```

File name: "pattern_A.dat" "pattern_B.dat"

Files of type: All Files