KYUNGPOOK NATIONAL UNIVERSITY

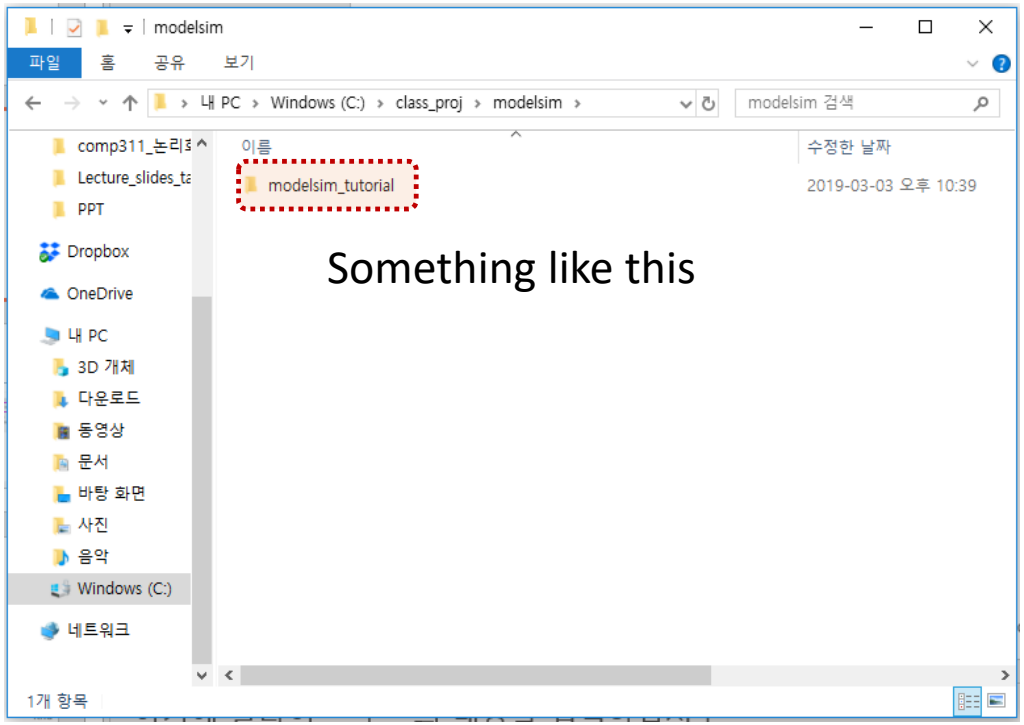
COMP311-1 Logic Circuit Design Chap. 2

Instructor: Taigon Song (송대건)

2019 Fall

Modelsim Tutorial [1/10]

- Create a folder in the location you desire

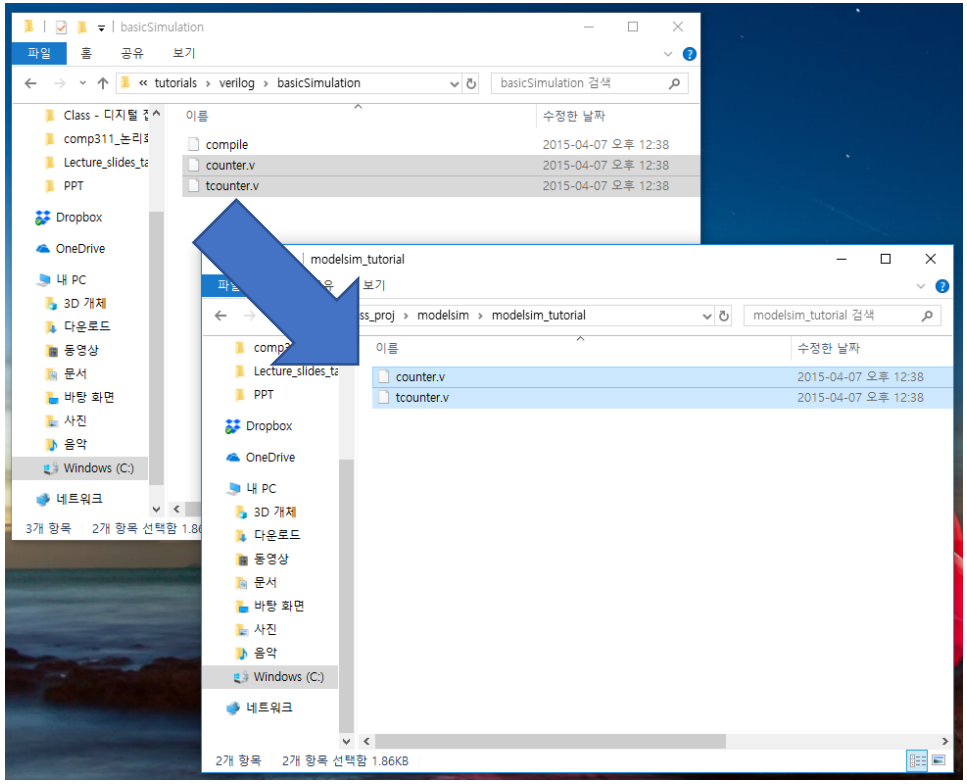


- Note)
  - Always in (D:) when using the lab desktop
  - No Korean
  - No spaces

Modelsim Tutorial [2/10]

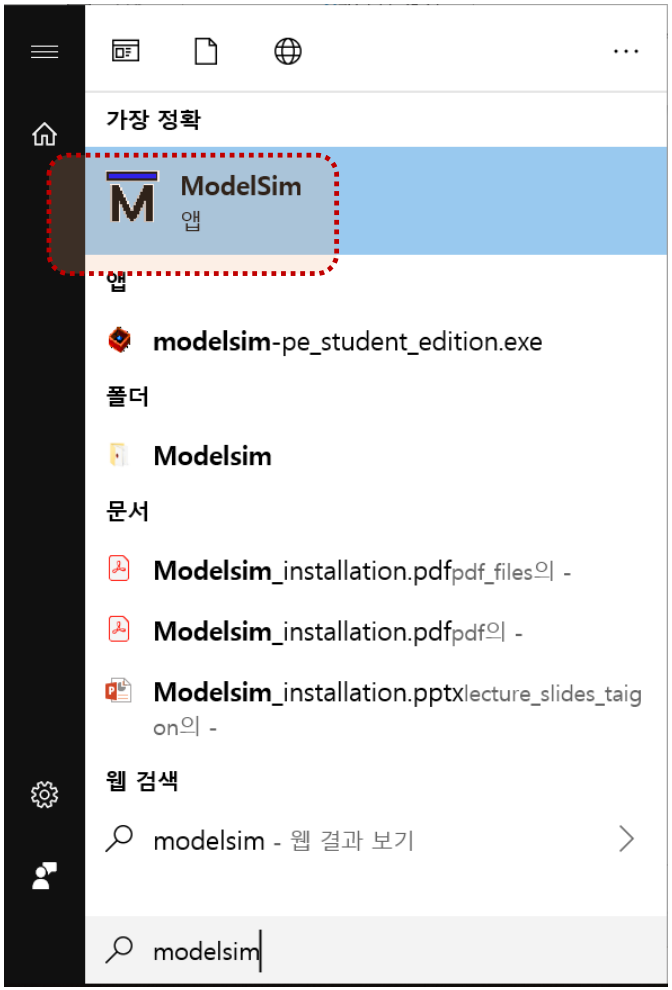
- Copy two files from the Modelsim installation directory
  - counter.v and tcounter.v

Verilog – *<install_dir>/examples/tutorials/verilog/basicSimulation/counter.v* and *tcounter.v*



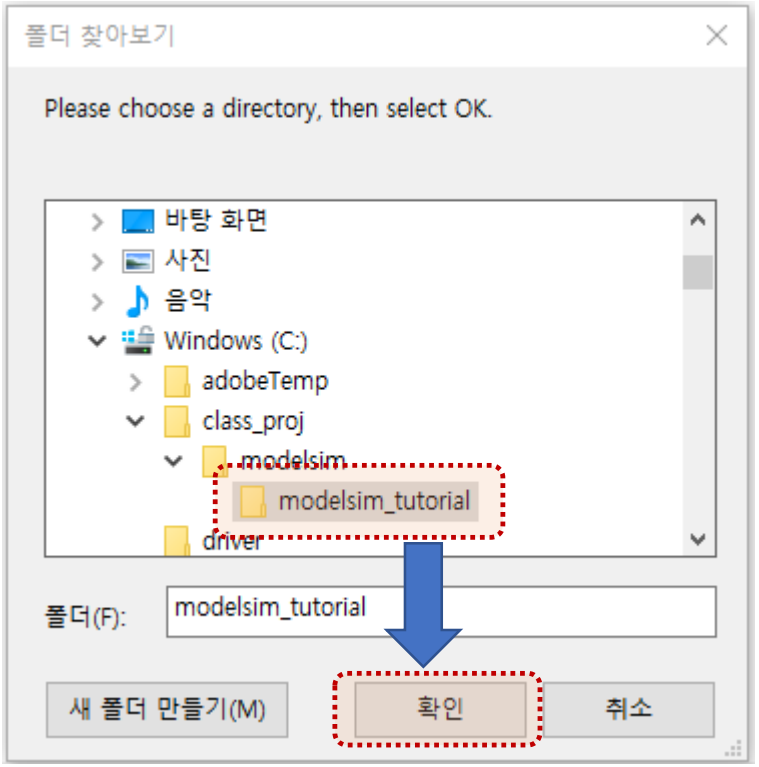
Modelsim Tutorial [3/10]

- Run modelsim
  - Run in administrative mode
    when you're having problems executing
    in default permission settings



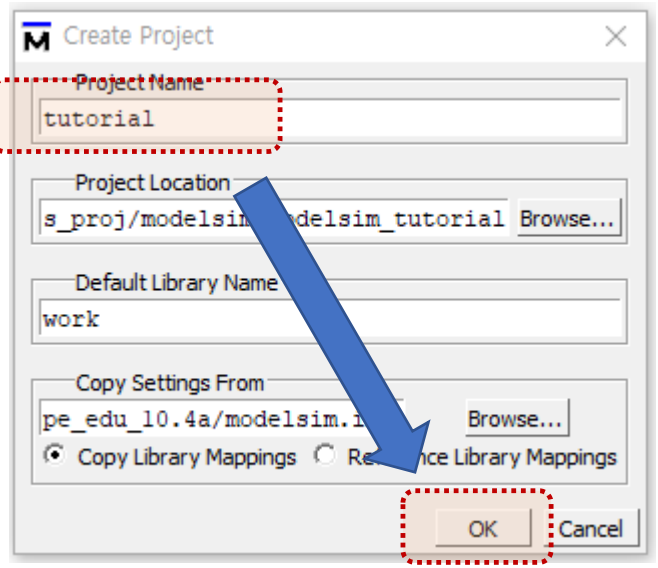
Modelsim Tutorial [4/10]

- File → Change directory
  - Change it to the directory that you created and copied the two Verilog files. (counter.v and tcounter.v)
  - Then OK

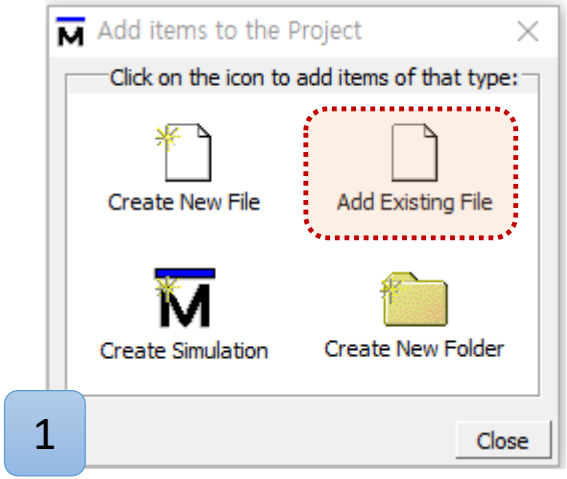


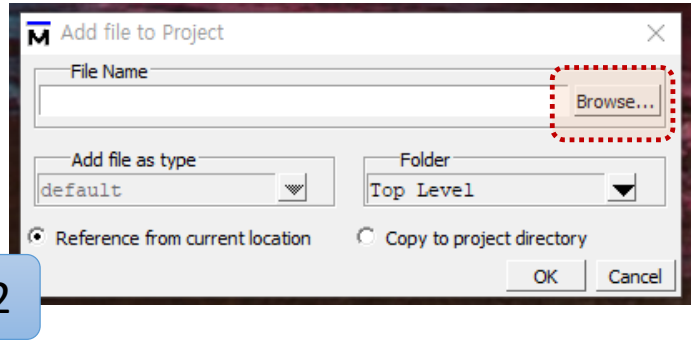
Modelsim Tutorial [5/10]

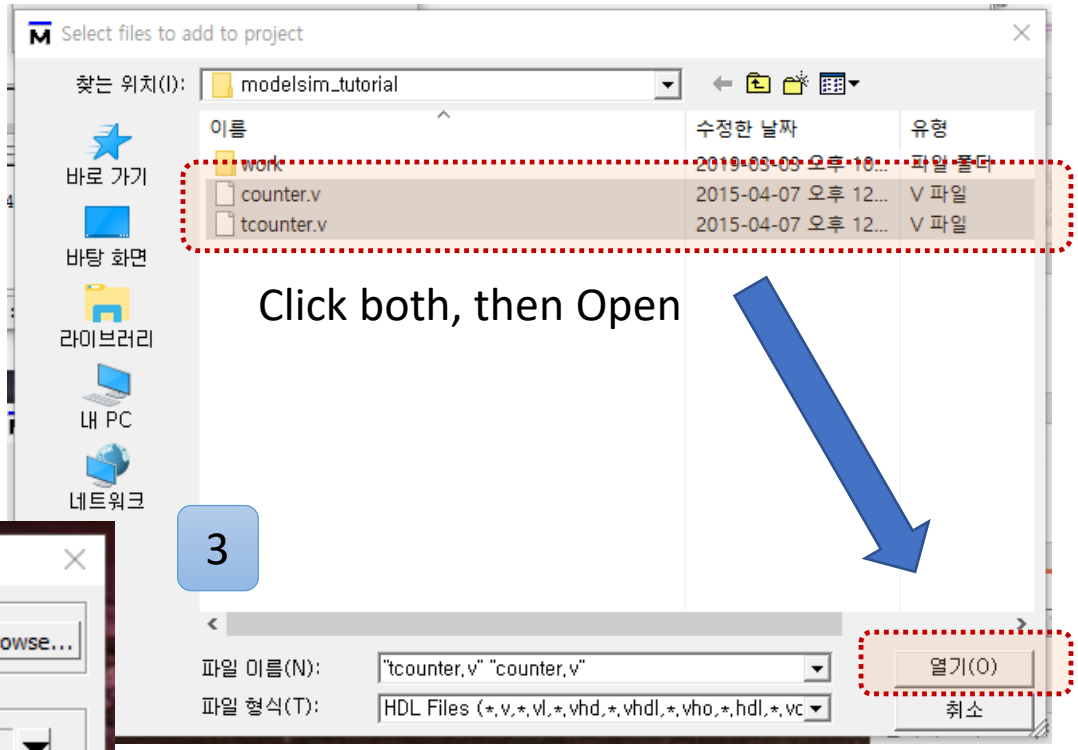
- File → New → Project
  - Change the Project Name, and don't change any other items
  - Then OK

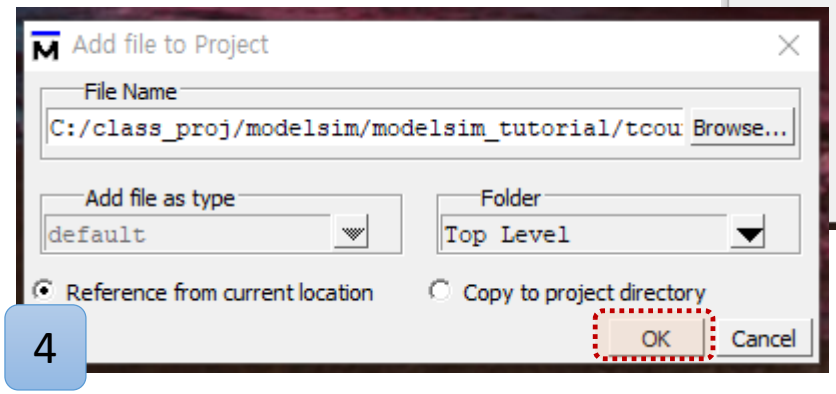


Modelsim Tutorial [6/10]



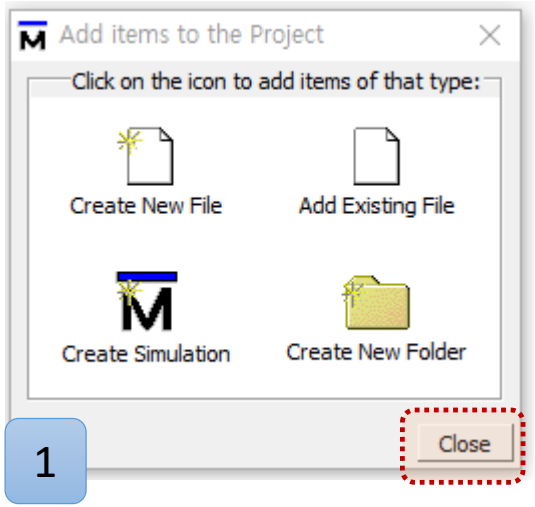


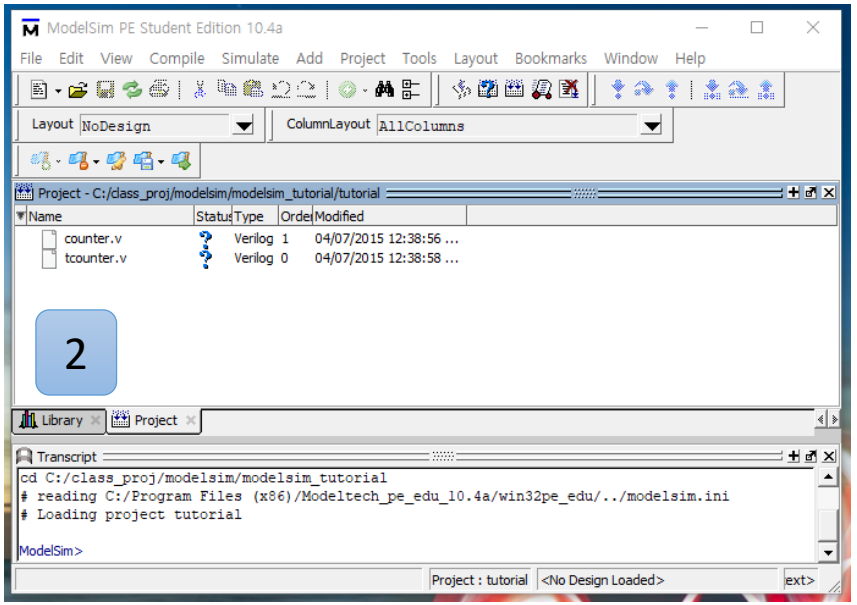


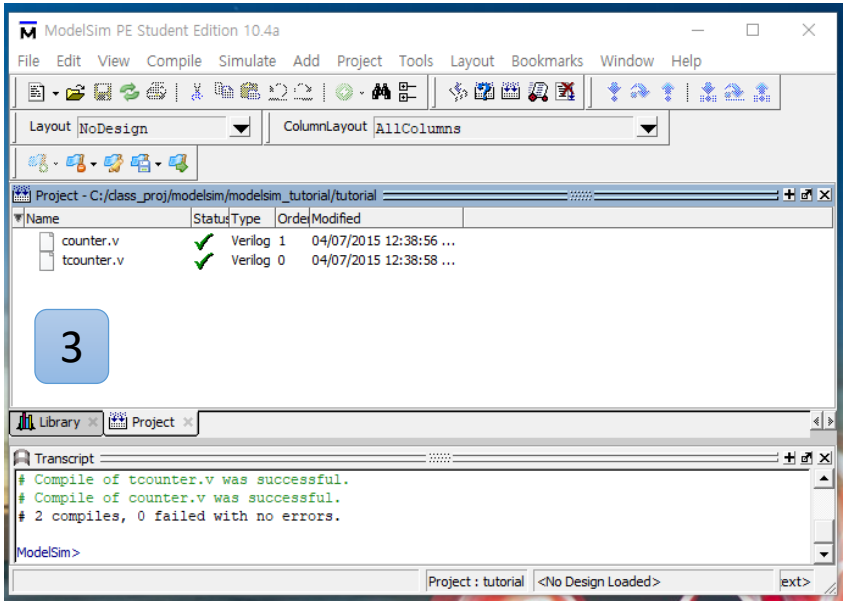


Modelsim Tutorial [7/10]

- Close “Add items to the Project”
- Compile → Compile all
  - Make sure “?” changes to “√”

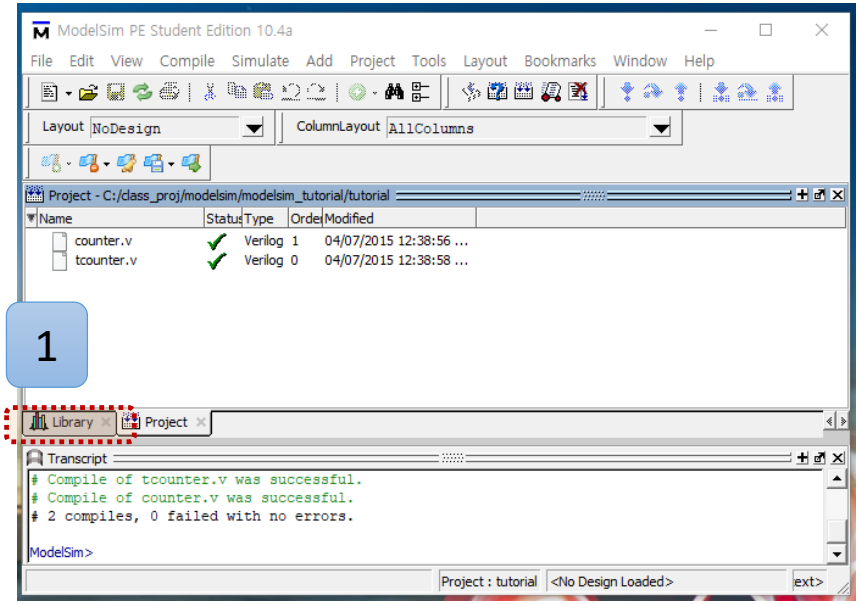


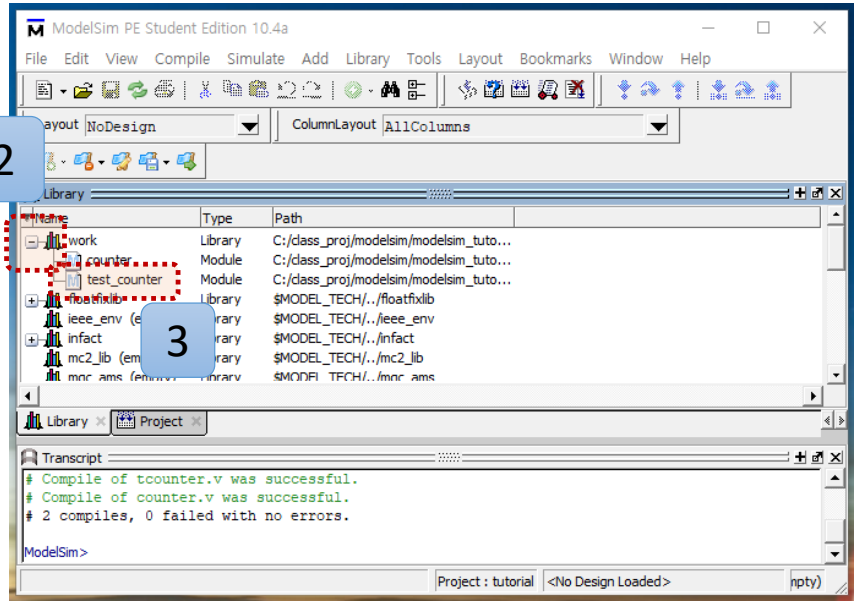




Modelsim Tutorial [8/10]

- Click “Library tab”, then click the “+” icon next to the “work”

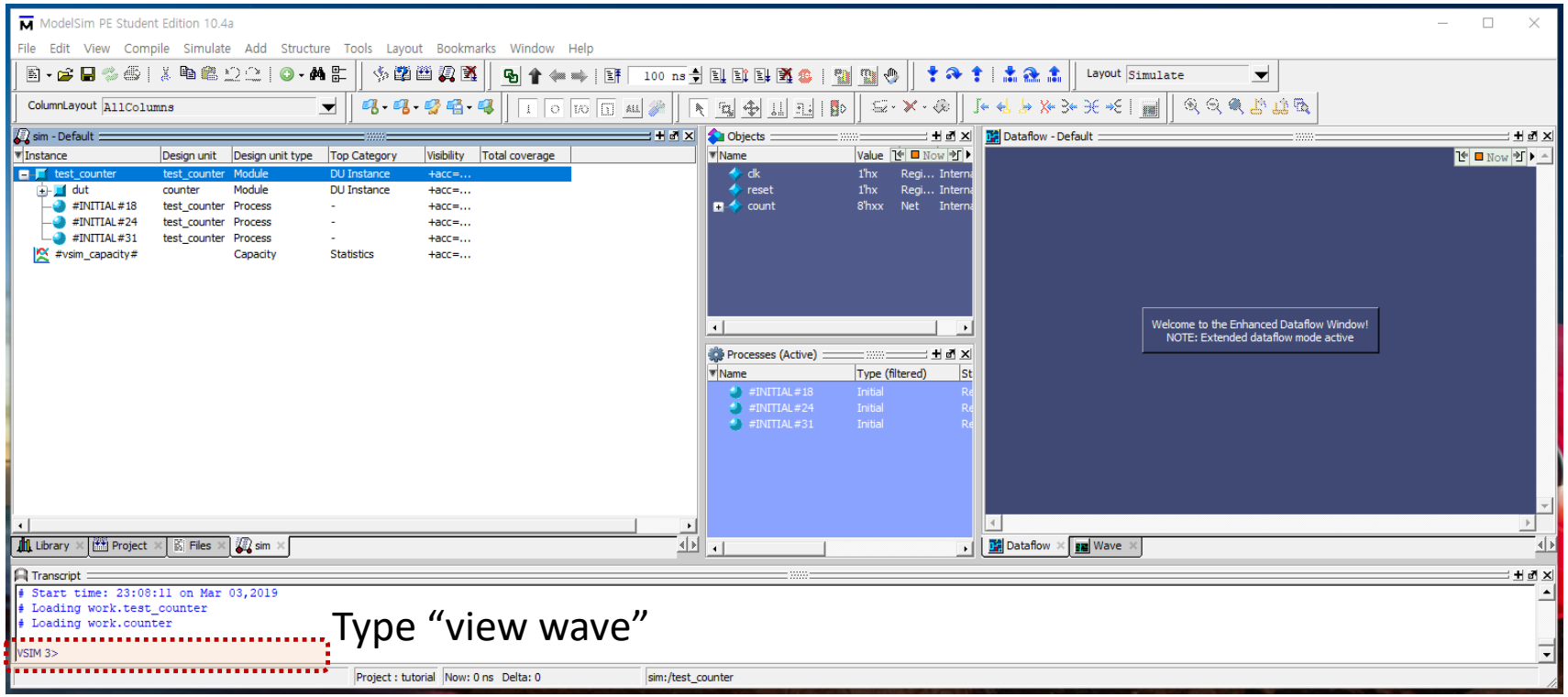




- Then, right click “test_counter” → simulate

Modelsim Tutorial [9/10]

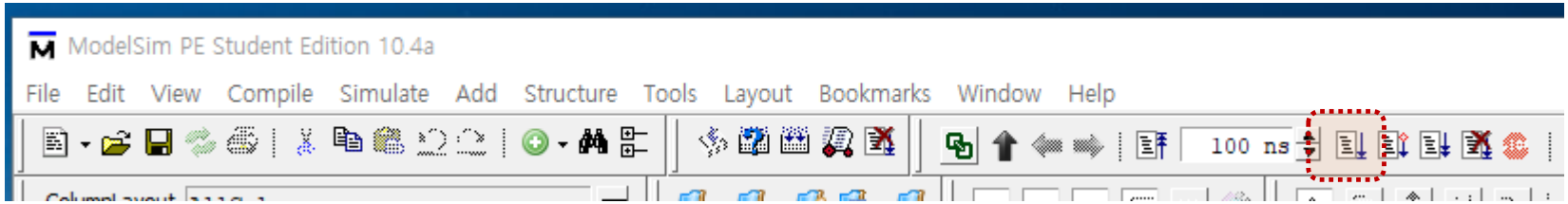
- After (9), you will see something like this:



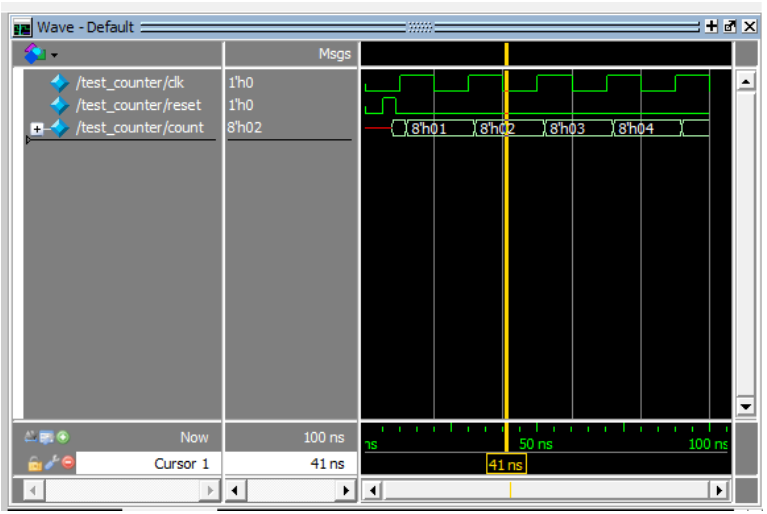
- Type “view wave” in the command prompt.
- Then, Add → To Wave → All items in Region

Modelsim Tutorial [10/10]

- Click the “Run” icon

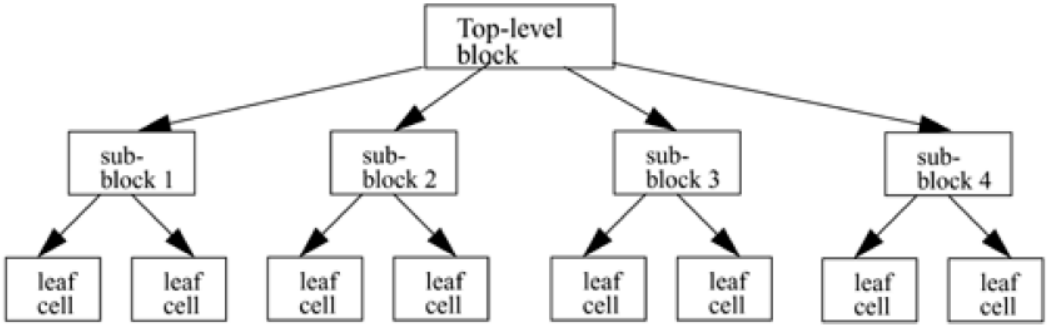


- In the wave view, right click → zoom full
- Then, you’ll see something like this

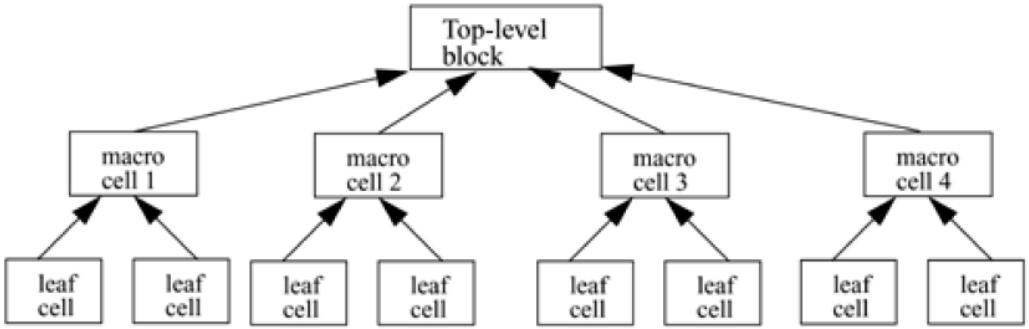


Design Methodologies

- Top-down

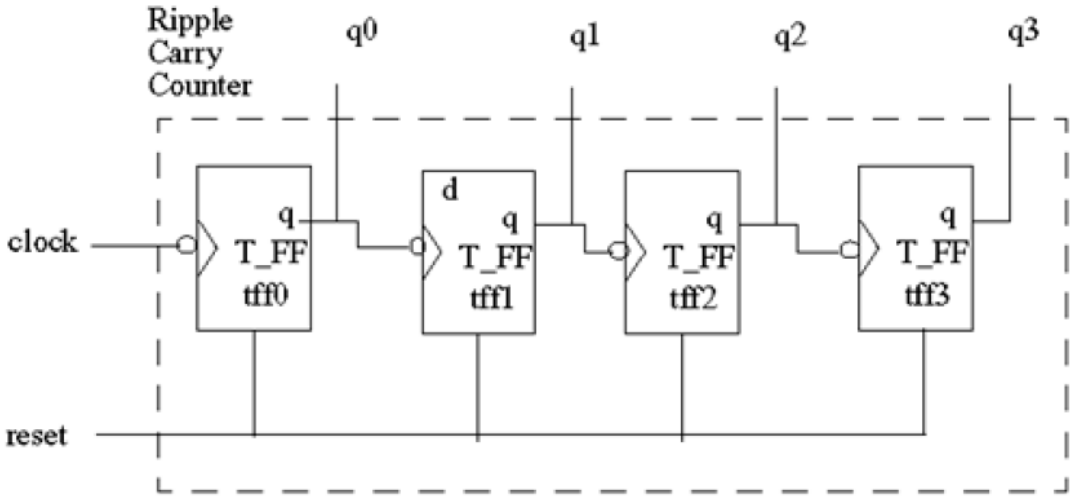


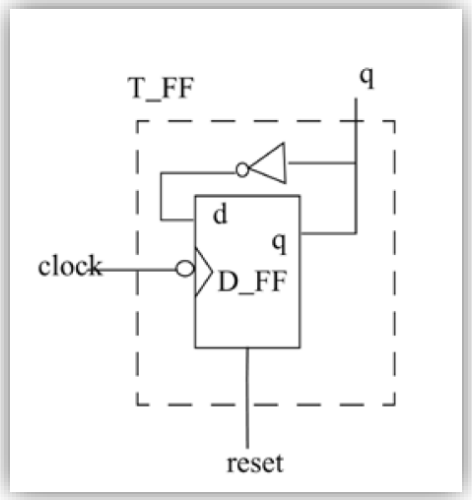
- Bottom-up



A 4-bit Ripple Carry Counter Example

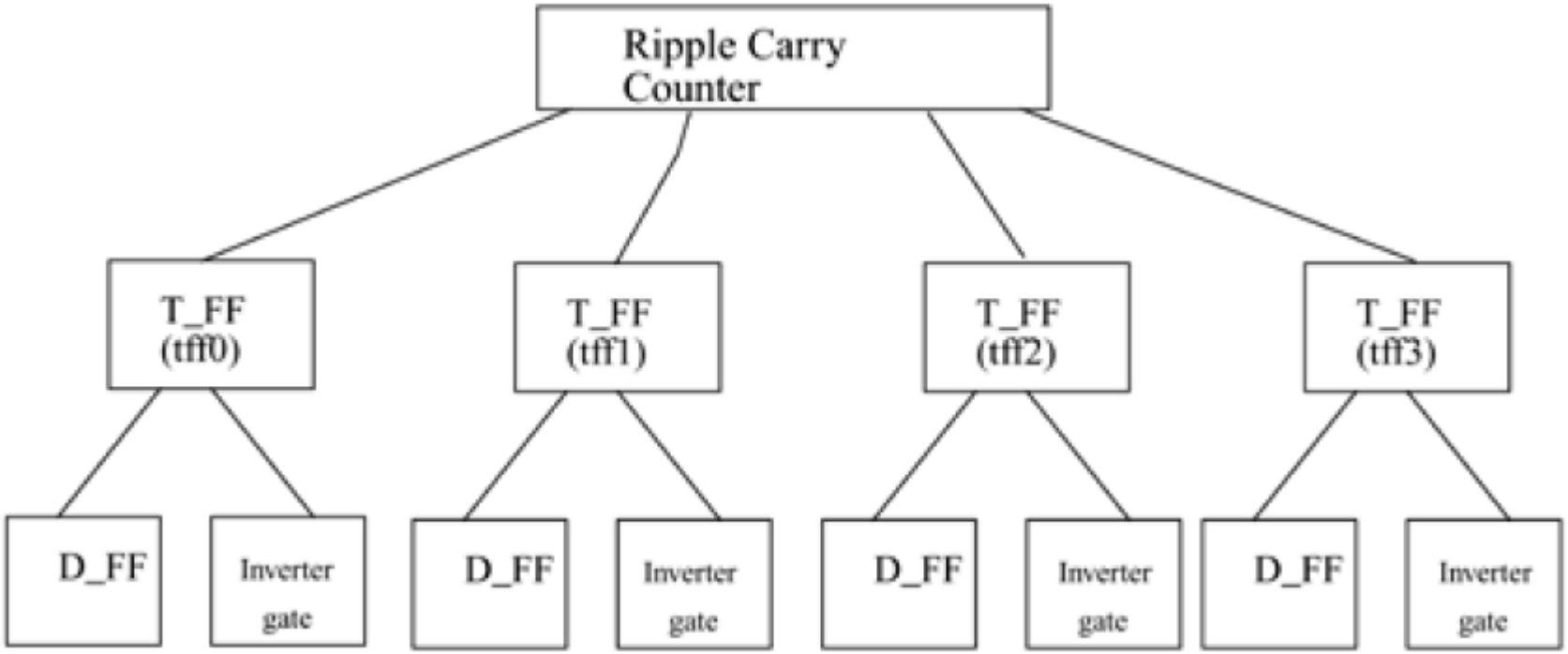
- Based on negative edge-triggered toggle flipflops (T_FF)





Design Hierarchy

Example of the "Ripple Carry Counter"



Syntax of a Module Definition

```
module <module_name>
(<module_terminal_list>);
...
<module internals>
...
...
endmodule
```

```
module T_FF (q, clock, reset);
.
.
<functionality of T-flipflop>
.
.
endmodule
```

Levels of Design Abstraction

- Behavioral or algorithmic level
  - Highest level of abstraction. Almost as similar to C programming
- Dataflow level
  - How data flows between HW registers and how data is processed
- Gate level
  - AND, OR, INV…etc
- Switch level
  - Transistor level

Instances

- A unique object summoned from a module definition

```
// Define the top-level module called ripple carry
// counter. It instantiates 4 T-flipflops. Interconnections are
// shown in Section 2.2, 4-bit Ripple Carry Counter.
module ripple_carry_counter(q, clk, reset);

output [3:0] q; //I/O signals and vector declarations
              //will be explained later.
input clk, reset; //I/O signals will be explained later.

//Four instances of the module T_FF are created. Each has a unique
//name.Each instance is passed a set of signals. Notice, that
//each instance is a copy of the module T_FF.
T_FF tff0(q[0],clk, reset);
T_FF tff1(q[1],q[0], reset);
T_FF tff2(q[2],q[1], reset);
T_FF tff3(q[3],q[2], reset);

endmodule
```

```
// Define the module T_FF. It instantiates a D-flipflop. We assumed
// that module D-flipflop is defined elsewhere in the design. Refer
// to Figure 2-4 for interconnections.
module T_FF(q, clk, reset);

//Declarations to be explained later
output q;
input clk, reset;
wire d;

D_FF dff0(q, d, clk, reset); // Instantiate D_FF. Call it dff0.
not n1(d, q); // not gate is a Verilog primitive. Explained later.

endmodule
```

Illegal Module nesting

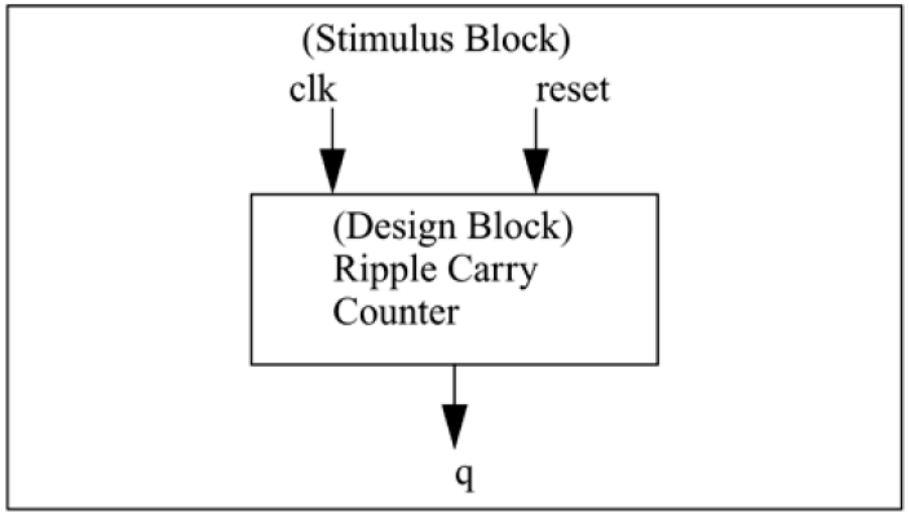
- What is the problem here?

```
// Define the top-level module called ripple carry counter.
// It is illegal to define the module T_FF inside this module.
module ripple_carry_counter(q, clk, reset);
output [3:0] q;
input clk, reset;

   module T_FF(q, clock, reset);// ILLEGAL MODULE NESTING
   ...
   <module T_FF internals>
   ...
   endmodule // END OF ILLEGAL MODULE NESTING

endmodule
```

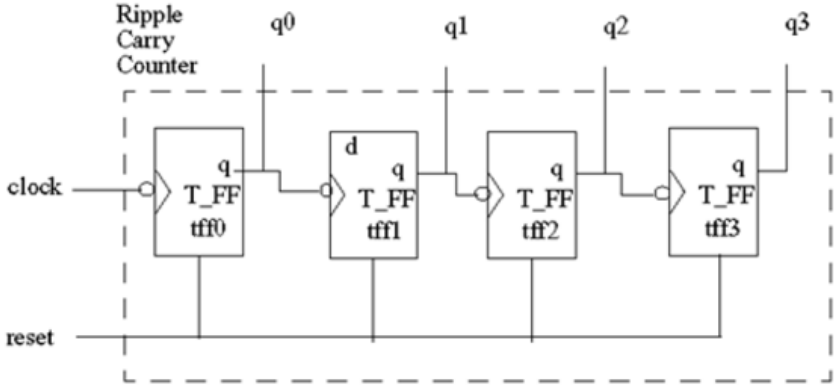
A Verilog Experiment

Understanding the concept of "testbench"

- What's the difference between a normal module and a testbench module?



Testbench module



Ripple Carry Counter module

A Verilog Experiment

What are needed in a Testbench.v