Welcome to The Hardware Lab!

Fall 2016
Lab 3: FPGA and Sequential Circuits

Prof. Chun-Yi Lee

Department of Computer Science
National Sting Hua University

Verilog Lab 3

- **Five basic questions** (**50%**)
  - Demonstration on **11/3/2016 (Thu)**, **in class**

- **Lab 3 report** (**30%**)
  - **Circuit** or **block diagram** of your design, **and explain how it works**
  - Answer the questions listed in the handout (if any)
  - Contribution of each team member
  - What you have learned from Lab 3

- **Two optional questions** (**20%**)
  - Challenge it if you have extra time
  - Include in your Lab 3 report if you can make it
  - **Describe your design and how it works**

Verilog Lab 3 Rules

- You can use **ANY** modeling techniques
- Demonstrate your work by **waveforms (Q1~Q3) and FPGA (Q4~Q5, OQ1~OQ2)**
- Please follow the **module names** and **I/O ports** of the provided templates
  - Submit **only one testbench file (Q1~Q3)** and **one design file**
  - **Avoid commenting out** modules in your submitted Verilog files
  - **Please keep in mind that we will deduct points if the module names and I/O port mismatch with the template**
- If not specifically mentioned, we assume the following SPEC
  - **CLK** is **positive edge triggered**
  - Reset the Flip-Flops when **RESET == 1'b0**

Verilog Lab 3 Submission

- **Demonstration** and **code** submission due date & time:

  5:30pm, 11/3/2016 (Thu)

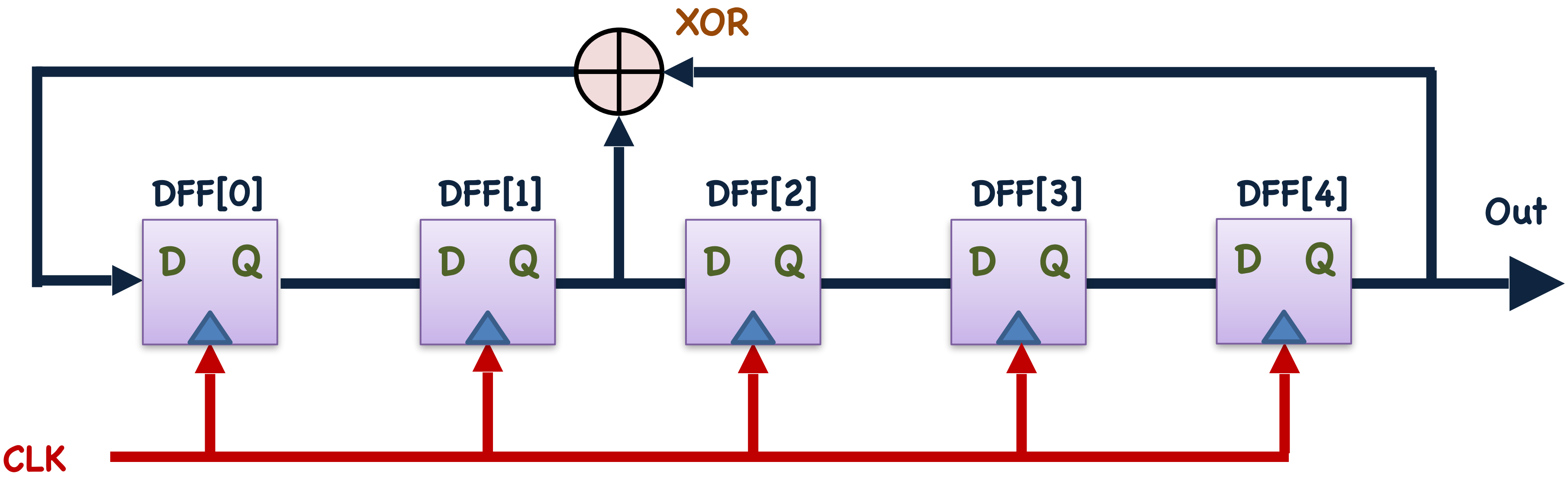
  This is a HARD deadline

- Please submit your report and Verilog codes to **ILMS**
  - Format: **Lab3_Team1_Codes.v, Lab3_Team1_Tb.v, Lab3_Team1_Report.**pdf
  - **We will deduct points if the submitted files mismatch the requirements**
  - **Please don’t uncomment the modules in the template file**

- We will test your codes by our own testbench

Question 1

- Implement a Linear-Feedback Shift Register (LFSR)

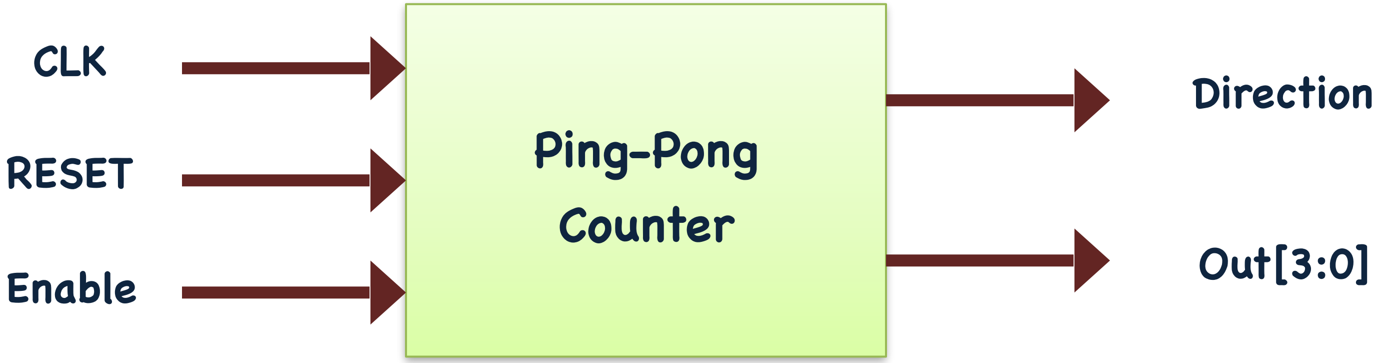


- When **RESET == 1’b0**, reset DFF[4:0] to **5’b00001**
- Please draw the **state transition diagram** of the DFFs in LFSR
- Please describe what happens if we reset the DFFs to **5’b00000**

Question 2

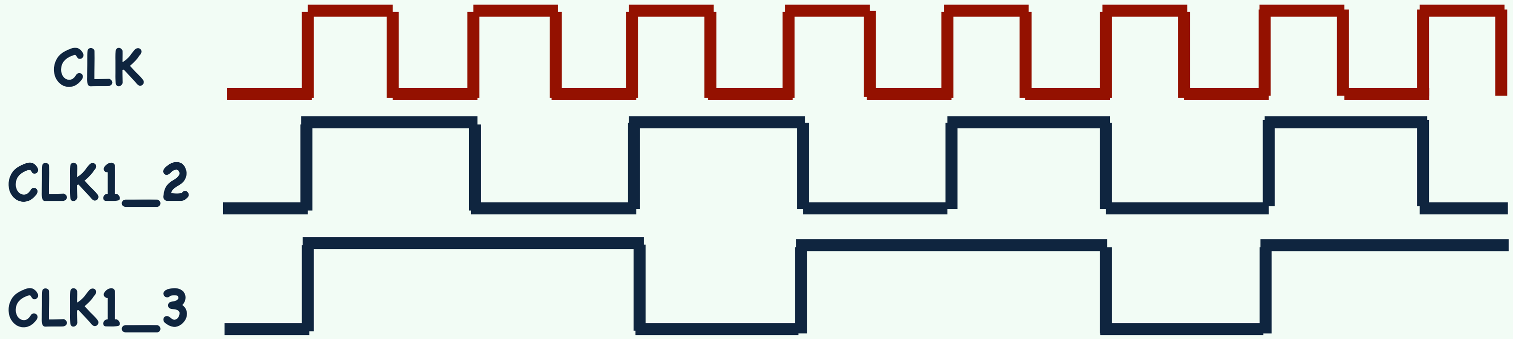
- Design a 4-bit Ping-Pong Counter
  - **Out:** 0,1,2,…,13,14,15,14,13,…,2,1,0,1,2,…
  - **Direction:** 0,0,0,…..,0, 0, 0, 1, 1,…,1,1,1,0,0,…

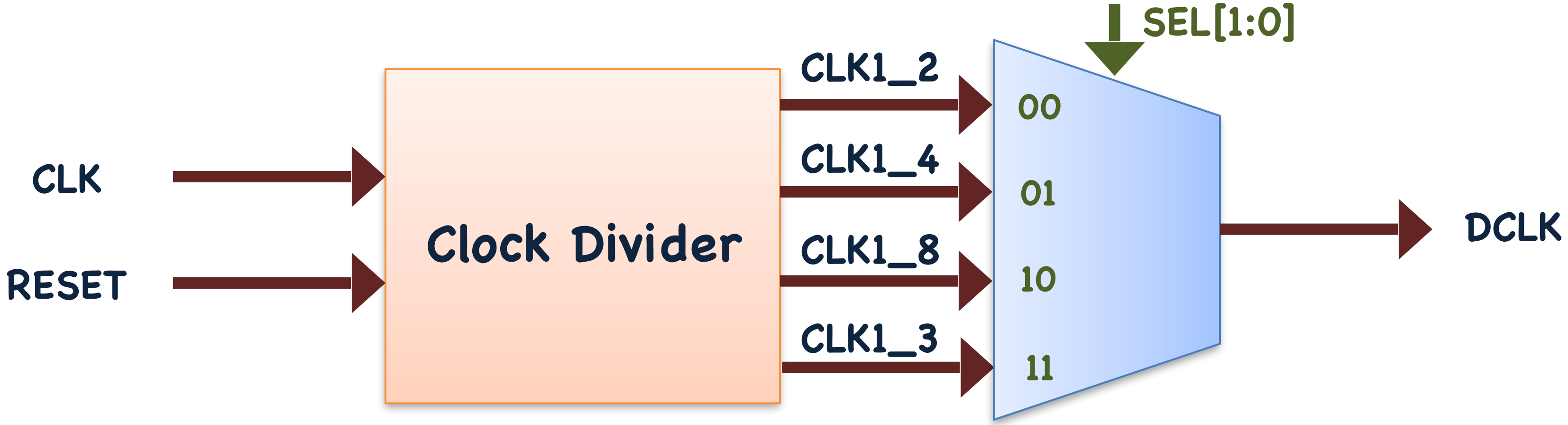
- SPEC
  - When **RESET == 1’b0**, the counter resets its value to 4’b0000
  - When **Enable == 1’b1**, the counter begins its operation. Otherwise, the counter holds its current value



Question 3

- Design a Clock Divider
  - SEL[1:0] and the mux are combinational, not triggered by CLK
  - **Outputs: CLK1_2, CLK1_4, CLK1_8, CLK1_3, DCLK**





Question 4

- Practice **switches** and **LEDs** on your FPGA

- Use your **switches** to provide binary inputs
  - Please use **SW[15:0]** as your inputs

- Display that value on **LEDs**
  - Please use **LED[15:0]** as your outputs

- Demonstrate your FPGA work on **11/3/2016 (Thu)**

Question 5

- Create a 6-bit Full Adder on FPGA
  - You can use either Ripple-Carry Adder or CLA Adder

- Use Switches as your **Cin**, **A[5:0]**, and **B[5:0]**
  - **Cin**: **SW[12]**
  - **A[5:0]**: **SW[11:6]**
  - **B[5:0]**: **SW[5:0]**

- Use LEDs to show your Sum and Cout
  - **Sum[5:0]**: **LED[5:0]**
  - **Cout**: **LED[6]**

- Demonstrate your FPGA work on **11/3/2016 (Thu)**

Questions to Answer in the Report

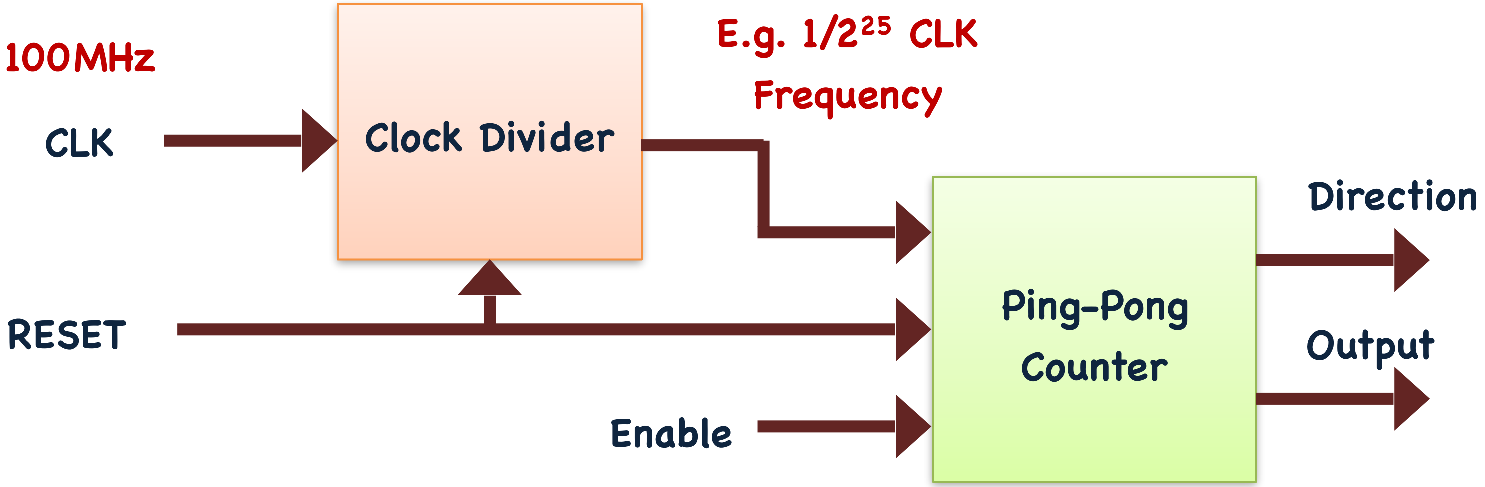
- You are in an interview. Now your interviewer is giving you an 8-bit binary number A[7:0] and asking you to design a circuit to divide A by 8. How would you design your circuit? (You can ignore the remainder)

- Your 4-bit adder can also be a 4-bit subtractor. Let's assume A = 4'b1110 and B=4'b0111, how would you use your 4-bit adder to compute (A - B)?

- Your customer complains that your LFSR in Question 1 is broken. For some unknown reason, DFF[1] is always equal to 1 no matter what happens. How would it affect the state transition diagram?

Optional Question 1

- Configure your 4-bit Ping-Pong Counter on FPGA
  - Use Switches as your RESET (**SW[0]**) and Enable (**SW[1]**)
  - Use LEDs to show your Output (**LED[3:0]**) and Direction (**LED[4]**)
  - **You can use asynchronous RESET in your OQ1 Ping-Pong counter**
- Divide frequency by a clock divider
  - **In your report, please tell us the clock divider frequency you used**
- Demonstrate your FPGA work on **11/3/2016 (Thu)**



Optional Question 2

- Try to get familiar with 7-Segment Display
- Use your switches to provide binary inputs
- Display the values on 7-Segment Display
  - Please use **Seg[6:0] and AN[3:0]** as your outputs
  - The four decimal digits should correspond to **SW[15:12]**, **SW[11:8]**, **SW[7:4]**, and **SW[3:0]**
  - **Report the clock divider frequency that you used** to display the four digits concurrently
  - Use **SW[0]** as your **RESET** (**synchronous RESET**) as well

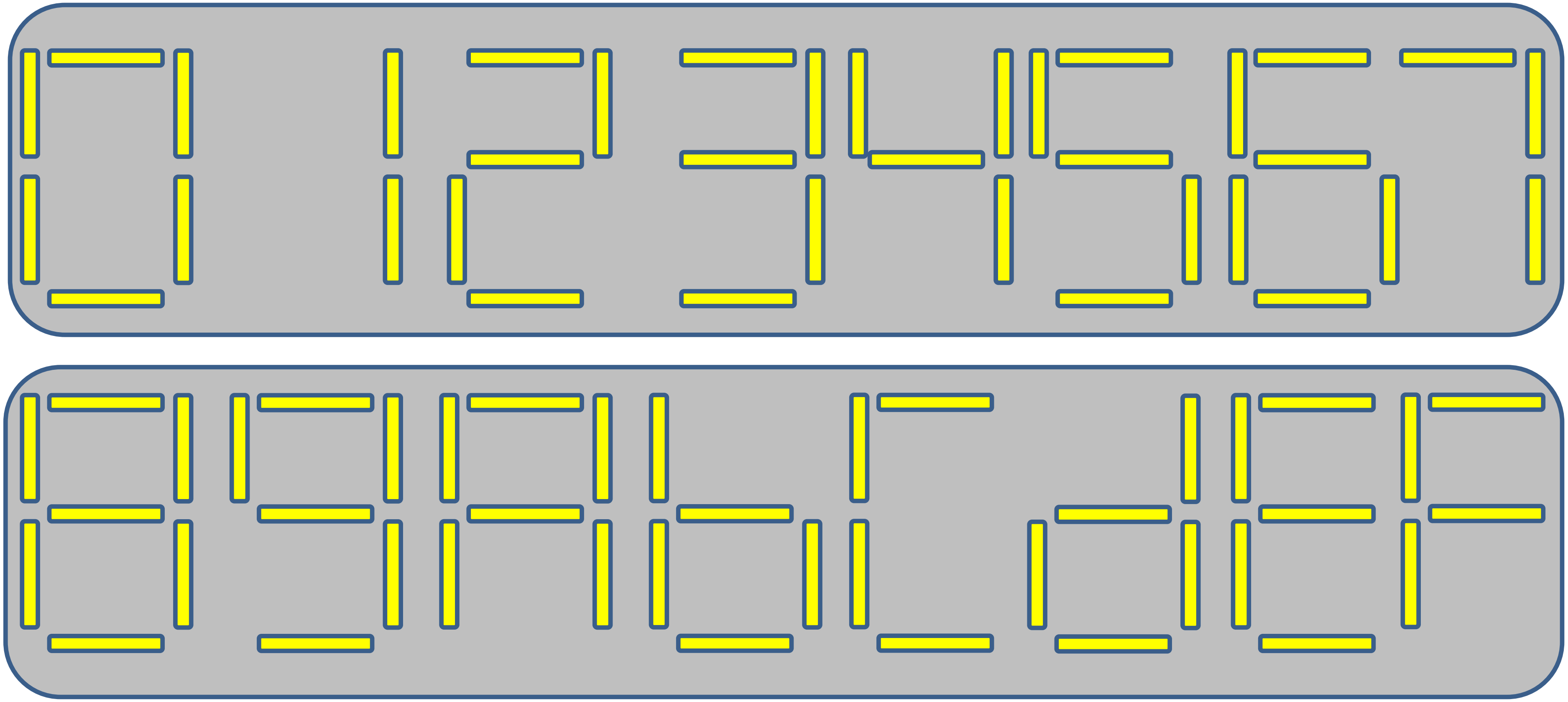
When AN[3] == 1'b0	Display SW[15:12]
When AN[2] == 1'b0	Display SW[11:8]
When AN[1] == 1'b0	Display SW[7:4]
When AN[0] == 1'b0	Display SW[3:0]

- Demonstrate your FPGA work on **11/3/2016 (Thu)**

Optional Question 2 (Cont’d)

- **Input** to **output** digit mappings

0000 0001 0010 0011 0100 0101 0110 0111



1000 1001 1010 1011 1100 1101 1110 1111

Thank you for your attention!

*Balloon Festival at Reno, Nevada, USA
This picture is taken by Chun-Yi Lee himself, who is also a fan of photography