Welcome to The Logic Design Lab!

Fall 2020

Lab 3: Sequential Circuits

Prof. Chun-Yi Lee

Department of Computer Science

National Tsing Hua University

Agenda

- Lab 3 Outline
- Lab 3 Basic Questions
- Lab 3 Advanced Questions

Lab 3 Outline

- Basic questions (1.5%)
  - Individual assignment
  - Due on 10/15/2020. In class.
  - Only demonstration is necessary. Nothing to submit.

- Advanced questions (5%)
  - Group assignment
  - ILMS submission due on 10/22/2020. 23:59:59.
  - Demonstration on your FPGA board (In class)
  - Assignment submission (Submit to ILMS)
    - Source codes and testbenches
    - Lab report in PDF

Lab 3 Rules

- Please note that grading will be based on **NCVerilog**
- You can use **ANY** modeling techniques

- If not specifically mentioned, we assume the following SPEC
  - **CLK** is **positive edge triggered**
  - Synchronously reset the Flip-Flops when **RESET == 1'b0**

Lab 3 Submission Requirements

- Source codes and testbenches
  - Please follow the templates EXACTLY
  - We will test your codes by TAs' testbenches
- Lab 3 report
  - Please submit your report in a single PDF file
  - Please draw the block diagrams of your designs by yourself
  - Please explain your designs in detail
  - Please list the contributions of each team member clearly
  - Please explain how you test your design
  - What you have learned from Lab 3

Agenda

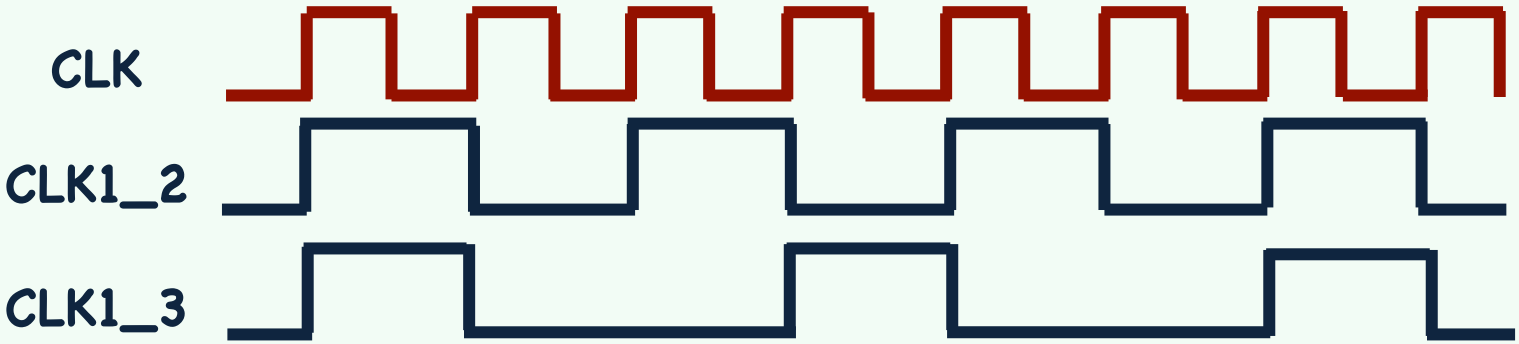
- Lab 3 Outline
- Lab 3 Basic Questions
- Lab 3 Advanced Questions

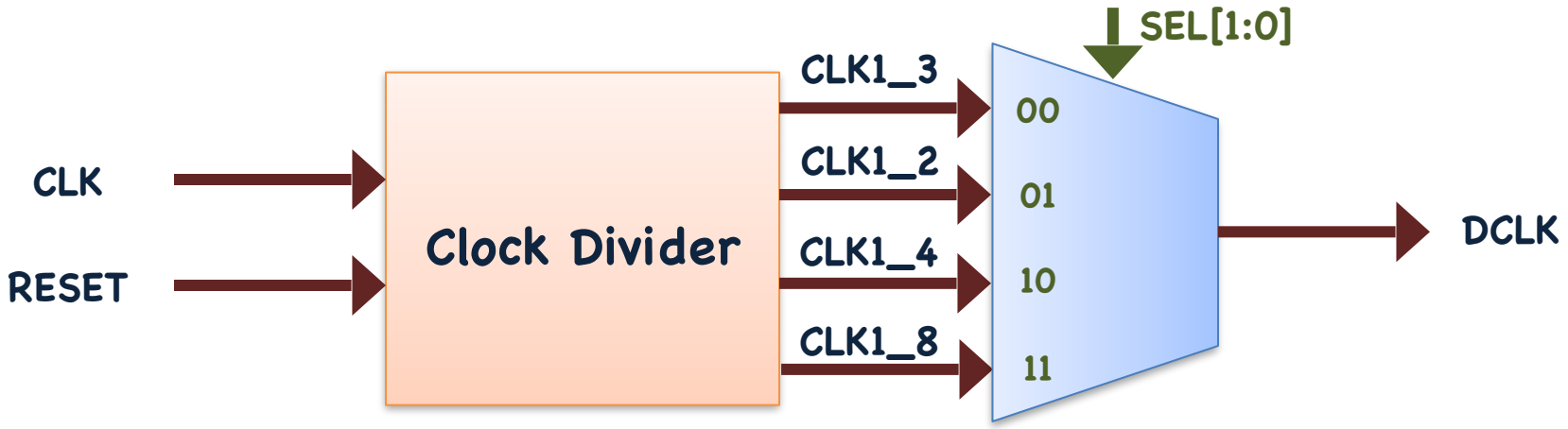
Basic Questions

- Individual assignment
- Verilog questions (due on 10/15/2020. In class.)
  - Clock Divider
  - 4-bit Ping-Pong Counter
- Demonstrate your work by waveforms

Verilog Question 1

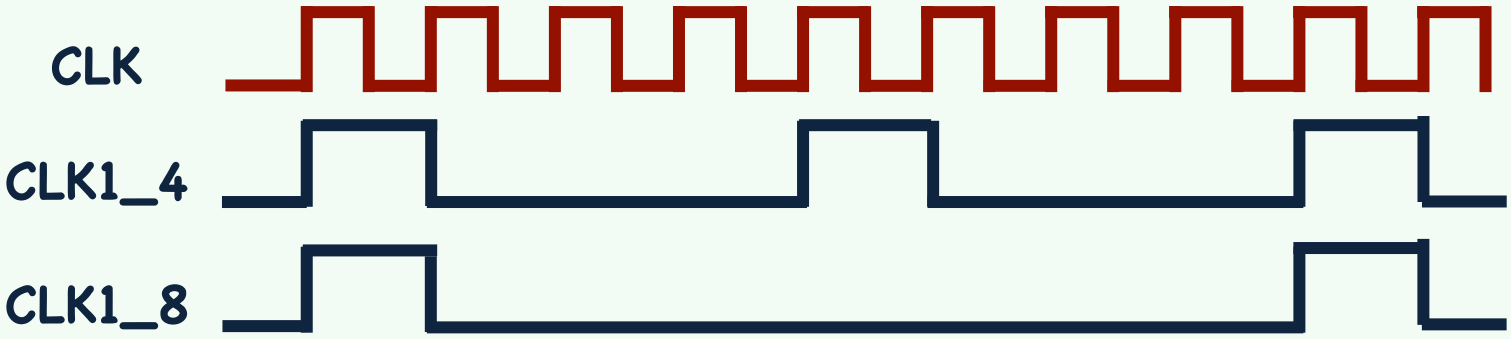
- Clock Divider
  - SEL[1:0] and the mux are combinational, not triggered by CLK
  - **Outputs: CLK1_2, CLK1_4, CLK1_8, CLK1_3, DCLK**

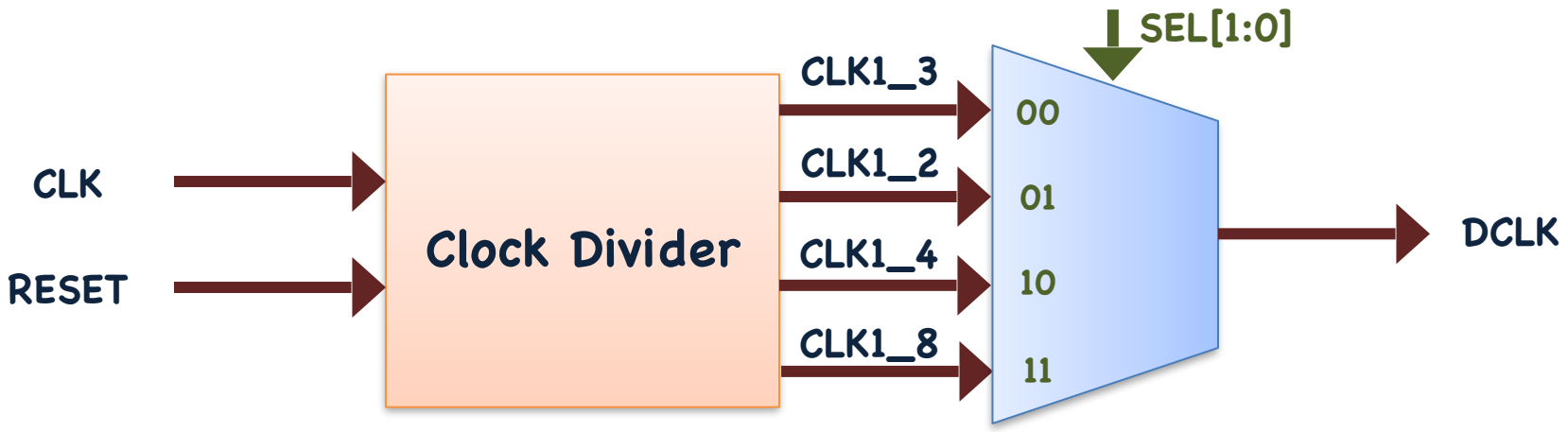




Verilog Question 1(Con't)

- Clock Divider
  - SEL[1:0] and the mux are combinational, not triggered by CLK
  - When RESET == 1'b0, all signals out the clock divider are **zero**
  - **Outputs: CLK1_2, CLK1_4, CLK1_8, CLK1_3, DCLK**





Verilog Question 2

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

- SPEC
  - When **RESET == 1'b0**, the counter resets its value to 4'b0001, and the **Direction** to 1'b1.
  - When **Enable == 1'b1**, the counter begins its operation. Otherwise, the counter holds its current value

Agenda

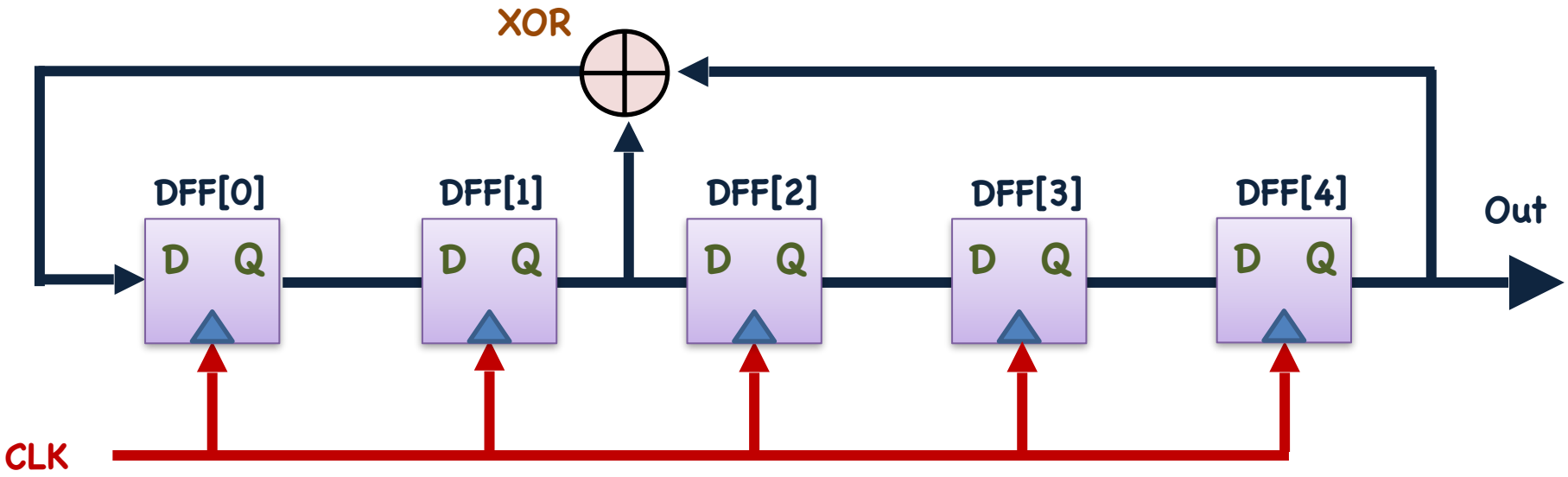
- Lab 3 Outline
- Lab 3 Basic Questions
- Lab 3 Advanced Questions

Advanced Questions

- Group assignment
- Verilog questions (due on 10/22/2020. 23:59:59.)
  - Linear–Feedback Shift Register
  - 64 x 8 memory array
  - 4–bit Paramterized Ping–Pong Counter
- FPGA demonstration (due on 10/22/2020. In class.)
  - 4–bit Paramterized Ping–Pong Counter on FPGA

Verilog Question 1

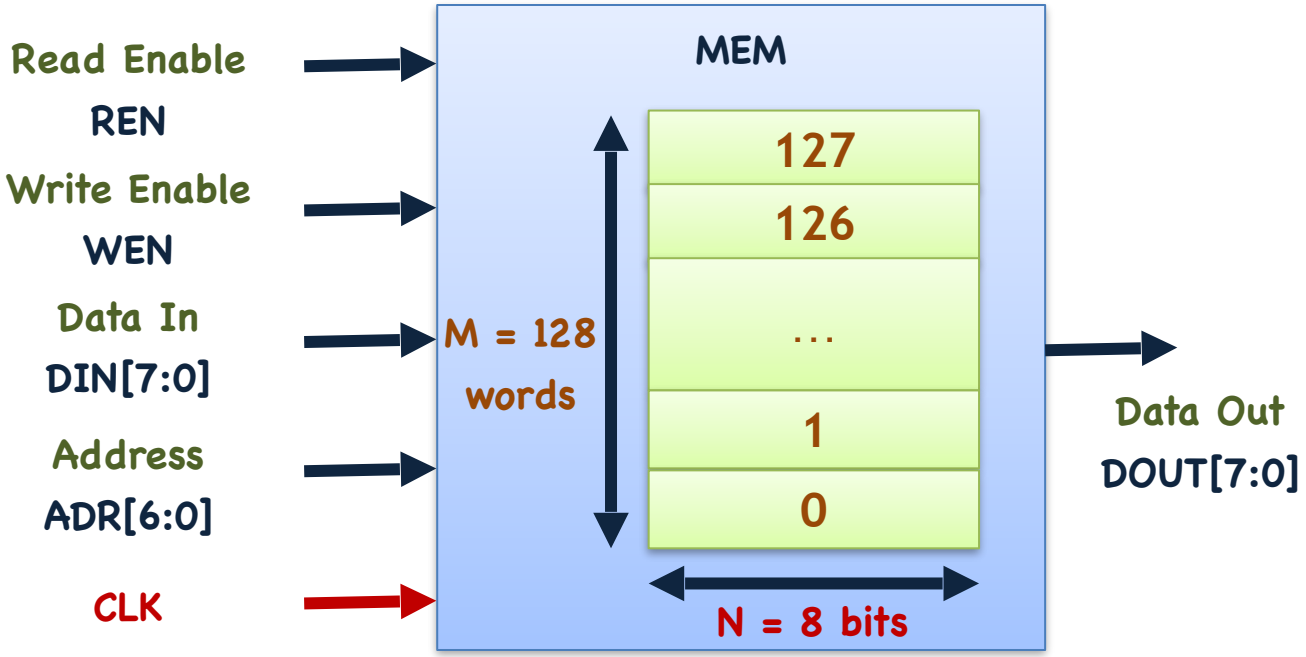
- Linear-Feedback Shift Register (LFSR)



- When **RESET == 1'b0**, reset DFF[4:0] to **5'b11111**
- Please draw the **state transition diagram** of the DFFs in LFSR
- Please describe what happens if we reset the DFFs to **5'b00000**

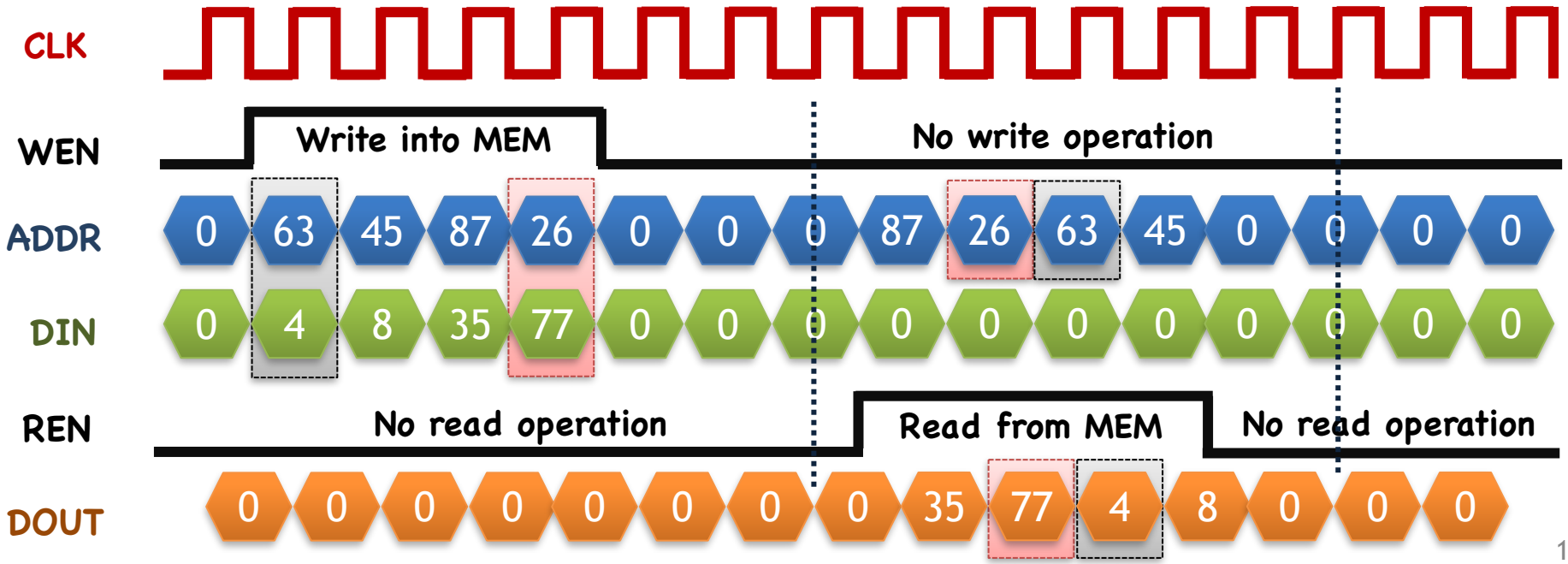
Verilog Question 2

- 128 x 8 memory array **MEM**
  - **M = 128, N = 8**
  - **Inputs**: CLK, REN, WEN, ADDR[6:0], DIN[7:0]
  - **Outputs**: DOUT[7:0]



Verilog Question 2 Specification

- Specification
  - When WEN == 1'b1, write DIN to MEM[ADDR]
  - When REN == 1'b1, output MEM[ADDR] to DOUT; otherwise DOUT = 8'd0
  - REN and WEN won't be one at the same time. **If both are 1, do only the read operation**
  - **MEM** does not need to be reset



Verilog Question 3

- Design a **4-bit Ping-Pong Counter** with **MAX** and **MIN**
  - **Input:** CLK, RESET, Enable, FLIP, MAX[3:0], MIN[3:0]
  - **Out:** 0,1,2,…,7,8,9,8,7,…,2,1,0,1,2,…
  - **Direction:** 1,1,1,…,1,1,1,0,0,…,0,0,0,1,1,…
  - In the above example, **MAX** is 9 and **MIN** is 0

Verilog Question 3 Specification

- **4-bit Ping-Pong Counter** SPEC
  - When **RESET == 1'b0**, the counter resets its value to **MIN**
  - When **Enable == 1'b1**, the counter begins its operation. Otherwise, the counter holds its current value

- **MAX** and **MIN**
  - **MAX** and **MIN** values are the maximum and minimum values for the counter
  - **MAX** > **MIN**. Otherwise, the counter holds its current value
  - When counter > **MAX** or counter < **MIN**, counter holds its current value

- **FLIP**
  - When **FLIP** == 1'b1, counter flips its direction
  - Flip is **only one cycle** in length
  - Flip occurs when counter < **MAX** and counter > **MIN**

Verilog Question 3 Specification

- **RESET**
  - When **RESET** == 1'b0, the value of the counter is set to **MIN**
  - When **RESET** == 1'b0, the value of Direction is set to **1'b1**
- **Notes**
  - Be careful that **MAX** and **MIN** will change during counting
  - Once the value of the counter is out of range, hold the value and direction
  - If **MAX** == **MIN** == **output**, please hold the **output** and **direction**

Advanced Questions

- Group assignment
- Verilog questions (due on 10/22/2020. 23:59:59.)
  - Linear–Feedback Shift Register
  - 64 x 8 memory array
  - 4–bit Paramterized Ping–Pong Counter
- FPGA demonstration (due on 10/22/2020. In class.)
  - 4–bit Paramterized Ping–Pong Counter on FPGA

FPGA Demonstration 1

- 4−bit Paramterized Ping−Pong Counter on FPGA
- **Behavior specification**
  - In the beginning, the digits showing on the 7-segment display should be **MIN**
  - Once **Enable** is on, the Ping-Pong Counter starts counting
  - When **Enable** is off, the Ping-Pong Counter holds its value
  - The Ping-Pong Counter only counts when **MAX** > **MIN**
- **Switches**
  - **SW[15]** stands for **Enable**
  - **SW[14:11]** stand for **MAX**
  - **SW[10:7]** stand for **MIN**

FPGA Demonstration 1

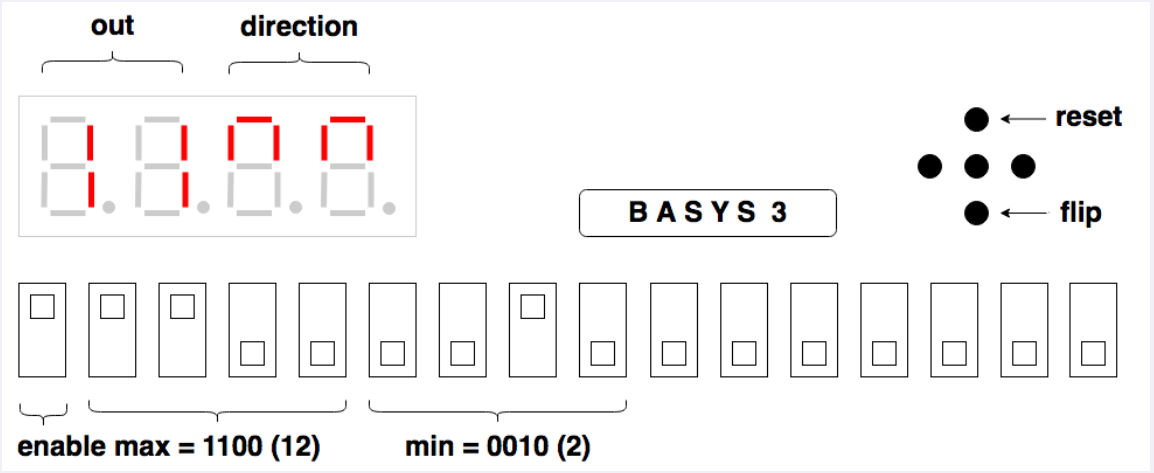
- **Buttons**
  - "**DOWN**" button stands for Flip
    - Once flip occurs, you should change your direction
    - Flip only occurs when **MIN** <= output <= **MAX**
  - "**UP**" button stands for **RESET**
    - Once the button is pushed, the output is set to **MIN**, which is determined by **SW[10:7]**
    - The direction is set to "**counting up**"
- **Please present your output signal on the two leftmost 7-segment displays**

FPGA Demonstration 1

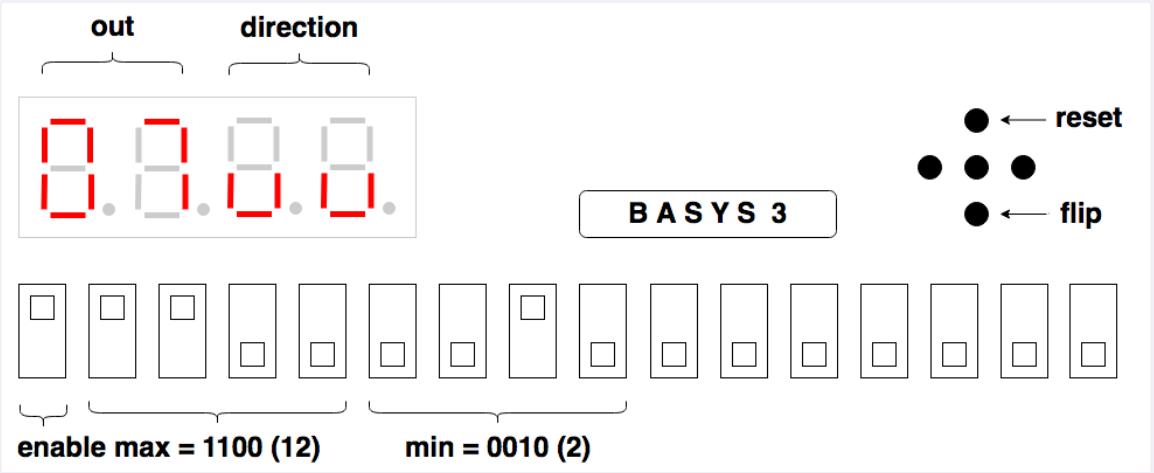
- **7-segment display**
  - The rightmost two digits of the 7-segment displays stand for **direction**
  - Please illuminates the upper three segments when counting up, and illuminates the lower three segments otherwise
  - Please see the figure on the next page for more details

FPGA Demonstration 1

Counting Up



Counting Down



FPGA Demonstration 1

- **Notes**
  - Be careful that **MAX** and **MIN** will change during counting
  - Once the value of the counter is out of range, hold the value and direction
  - If **MAX** == **MIN** == **output**, please hold the **output** and **direction**
  - You **MUST** add debounce and one-pulse circuits for your buttons
  - **Remember to add debounce and one-pulse circuits to your design**
  - We use the **100MHz** clock which is provided by the FPGA board. Please set **clk** as input and connect it with the **W5** port on the FPGA borad.
  - Your counter should count in an observable frequency so that TAs can tell whether your design is correct or not

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