Welcome to The Hardware Lab!

Fall 2016

Lab 5: Advanced Design Problems

Prof. Chun-Yi Lee

Department of Computer Science
National Sting Hua University

Verilog Lab 5

- **Three basic questions** (**50%**)
  - Demonstration on **12/8/2016 (Thu)**, **in class**
  - Report due on **12/13/2016 (Tue)**
- **Lab 5 report** (**30%**)
  - **Circuit** or **block diagram, and FSM state transition 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 5
- **Two optional questions** (**20%**)
  - Challenge it if you have extra time
  - Include in your Lab 5 report if you can make it
  - **Describe your design and how it works**

Verilog Lab 5 Rules

- You can use **ANY** modeling techniques
- Demonstrate your work by **waveforms (Q1~Q3) and FPGA (OQ1~OQ2)**
- Please follow the **module names** and **I/O ports** of the provided templates
  - Submit **only one testbench file** 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 5 Submission

- **Demonstration and code** submission due date & time:
  5:30pm, 12/8/2016 (Thu)
  This is a HARD deadline

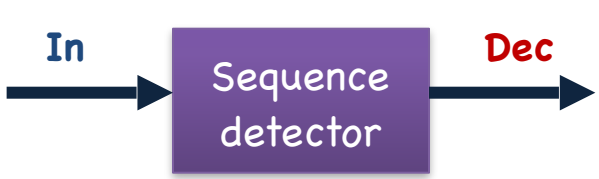
- **Report** submission due date & time:
  5:30pm, 12/13/2016 (Tue)
  This is a HARD deadline

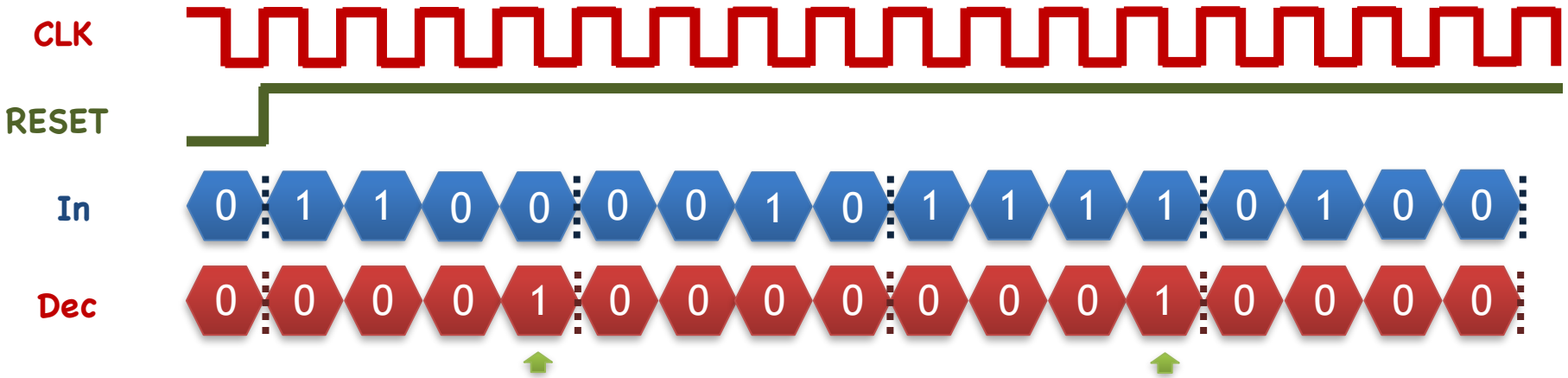
- Please submit your report and Verilog codes to **ILMS**
  - Format: **Lab5_Team1_Codes.v, Lab5_Team1_Tb.v, Lab5_Team1_Report.pdf**
  - **We will deduct points if the submitted files mismatch the requirements**
  - **Please don't uncomment the modules in the template file**

- We will test your codes by our own testbench

Question 1

- **Mealy machine sequence detector**
  - 1-bit input **In** and 1-bit output **Dec**
  - When the four bit sequence is either **1100** or **1111**, **Dec** is set to **1**
  - Re-detect the sequence **every four bits**
  - Please draw your state diagram in your report

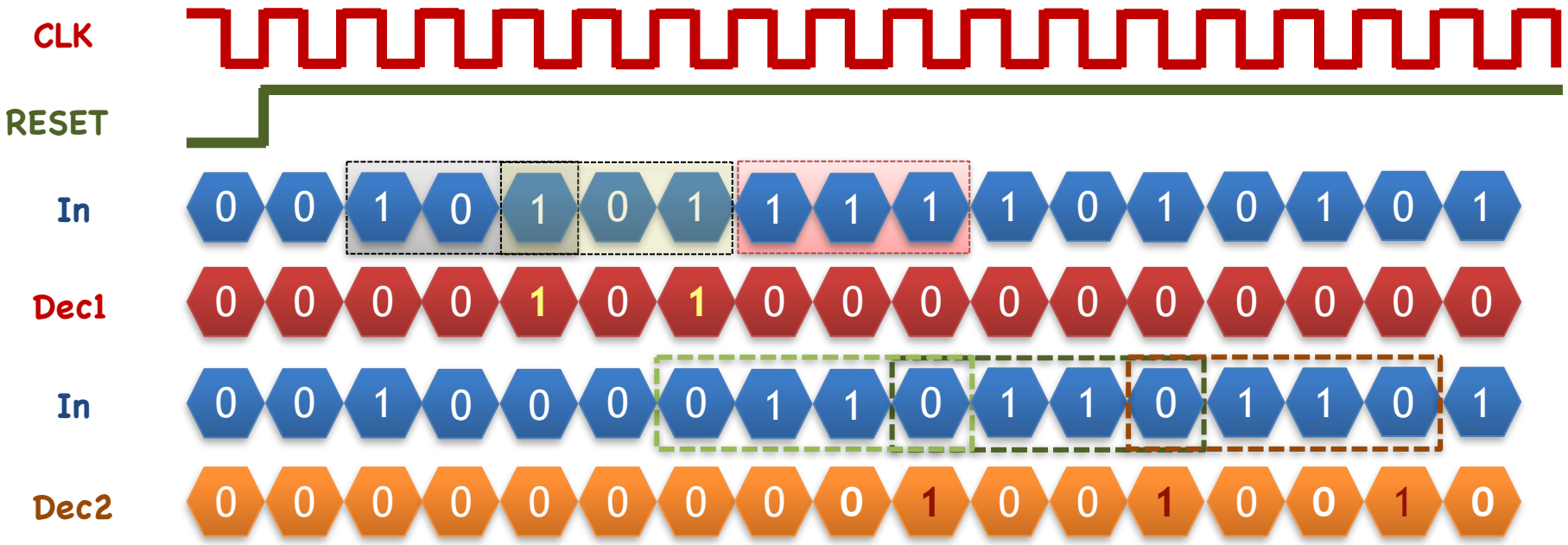




Question 2

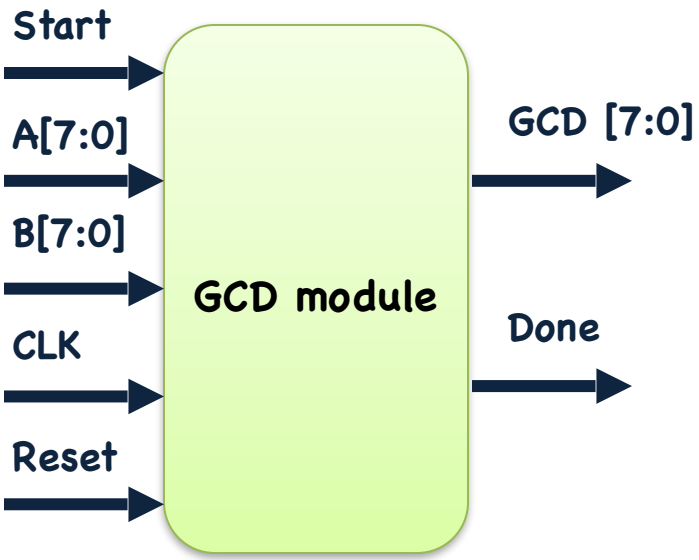
- **Sliding Window** sequence detector
  - **Dec1** == 1'b1 when input is **101** **AND no** **111** occurs before
  - **Dec2** == 1'b1 when input sequence is **0110**

- **Continuous detection**
  - Detect the sequences whenever they occur
  - Please draw a state transition diagram in your report



Question 3
Greatest Common Divisor (GCD)

- Calculate the greatest common divisor of two numbers **A** and **B**
- Block diagram and pseudo code are as follow
  - You **shall not** use **while statements** in your Verilog codes

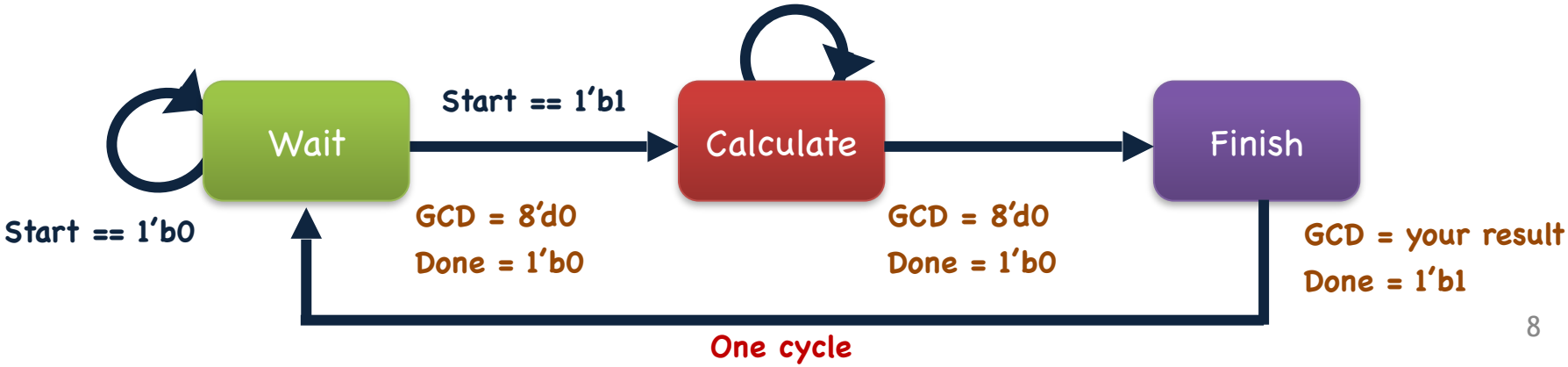


GCD pseudo code

```
Function gcd (a, b)
begin
    if (a == 0)
        return b;
    while (b != 0)
    // Do the following operation once per clock cycle
    begin
        if (a > b)
            a = a - b;
        else
            b = b - a;
    end
    return a;
end
```

Question 3 (Cont'd)
Greatest Common Divisor (GCD)

- Three states are used: **Wait**, **Calculate**, and **Finish**
- **Wait state**
  - Wait for Start == 1'b1 (**one cycle**) to begin the operation
  - When Reset == 1'b0, reset the module to the **Wait state**
- **Calculate state**
  - Calculate the **subtraction operations once per cycle**
- **Finish state**
  - Output the GCD result for **one cycle**
  - Done == 1'b1 for **one cycle**



Optional Question 1
Vending Machine

- Four options available: **Coffee**, **Coke**, **Oolong**, and **Milk Tea**
  - Prices are: **Coffee** (**NT$ 40**), **Coke** (**NT$ 15**), **Oolong** (**NT$ 20**), **Milk Tea** (**NT$ 30**)
- The **two rightmost 7-segment** displays show the money inserted into the machine
  - When RESET == 1'b0, please display "00"
  - The maximum value is **NT$ 70**
- Use **three buttons** to insert money
  - You can insert **NT$ 5**, **NT$ 10**, and **NT$ 50**

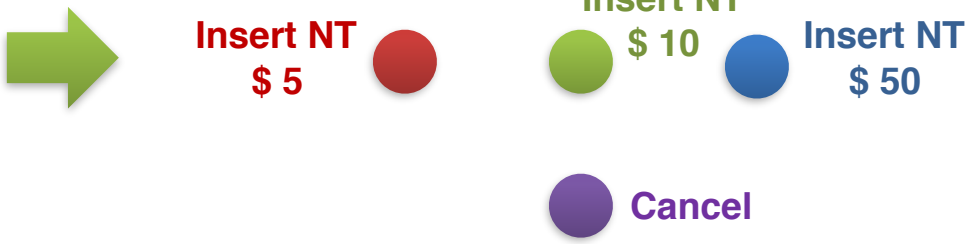
Optional Question 1 (Con'd)
Vending Machine

- Use **four LEDs** to indicate which drinks you can buy
  - **LED[3:0]** corresponds to Coffee, Coke, Oolong, and Milk Tea, respectively
- Use **four Switches** to buy a drink
  - **SW[3:0]** corresponds to Coffee, Coke, Oolong, and Milk Tea, respectively
  - You can only buy one drink at a time
- Use the **two rightmost 7-segment display** to show the change after buying a drink
  - E.g., if you inserted **NT$ 40** and bought a can of Coke (**NT$ 15**), the 7-segment display will show **NT$ 25**

Optional Question 1 (Con'd)
Vending Machine

- Assume that the machine allows you to buy **ONLY ONE DRINK** at a time
- Decrement the **7-segment display** by **NT$ 5** every second to mimic the process of returning changes
  - Return the changes until it becomes zero
- If the buyer does not want to buy a drink, he/she can use a **cancel button** to cancel it
  - The inserted money will be returned the same way (**NT$ 5** per second)

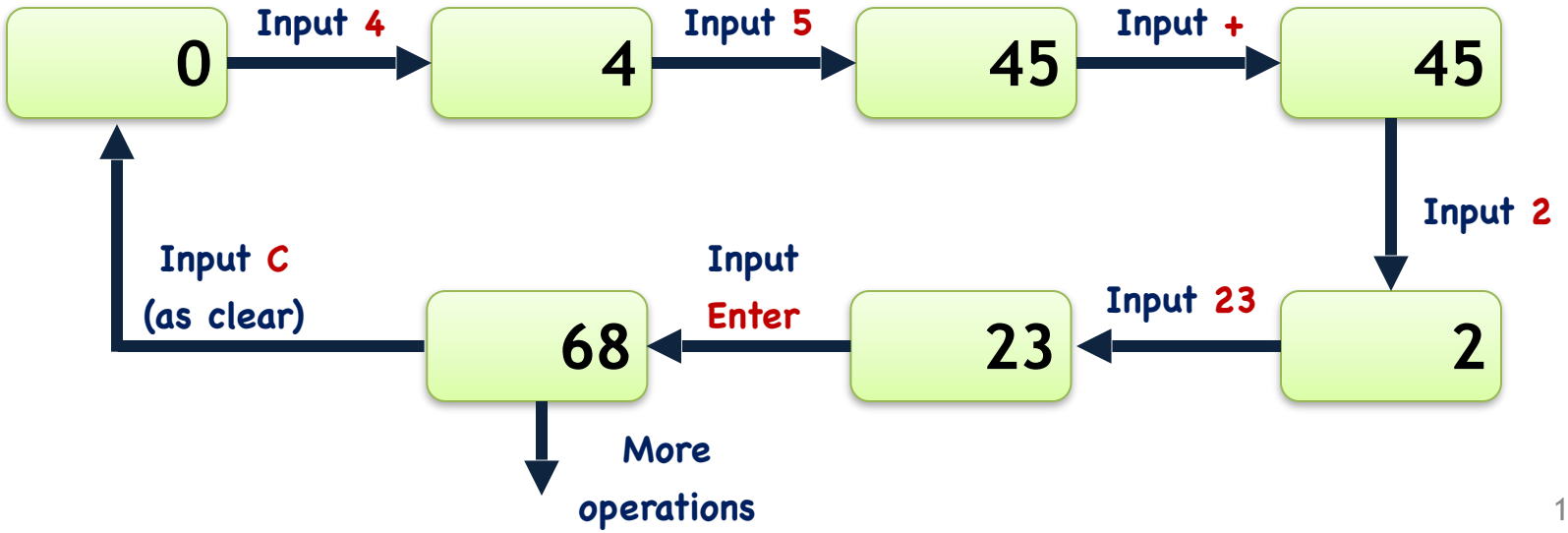
The layout of the buttons used in this question



Optional Question 2
Calculator

- Please use your keyboard to implement a calculator
  - The calculator can perform **+**, **-**, and **x**
  - To simplify your design, please use the letter "**x**" for multiplication operation
  - Use your keyboard to provide inputs

- The flow of the calculator is as follows:



Optional Question 2 (Cont'd)
Calculator

- Clear
  - Use the "**C**" key as clear
  - **You can hit this clear anytime to reset the calculator to zero**
- Overflow
  - If the result is larger than **999** or less than **-999**, simply show **999** and **-999**
- Cascade operations
  - You can cascade your results with more operations
  - E.g., You can continue with **+**, **-**, and **x** from the previous result **68** in the above example

Thank you for your attention!

*Lake Helen at Lassen Volcanic National Park, Shasta County, California, USA
This picture is taken by Chun-Yi Lee himself, who is also a fan of photography