Computer Organization

RTL Design

Hunjun Lee, Suheon Bae
High Performance Computer System (HPCS) Lab
March 27, 2018

Objective

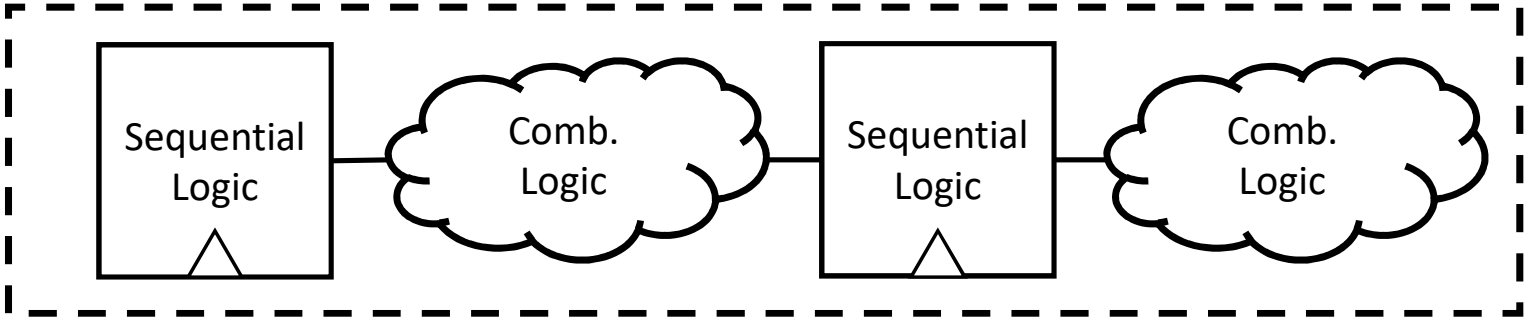
- To review register transfer level (RTL) designs

Previous Labs

- Lab 01: Introduction to Verilog (1)
  - Modeling combinational circuits in Verilog
  - Implementing a 16-bit ALU
- Lab 02: Sequential Logic & RTL
  - Modeling a simple FSM (010 detector)
  - Simple register file with read/write
- Now we are ready to implement a more complicated FSM (vending machine)

RTL design (review)

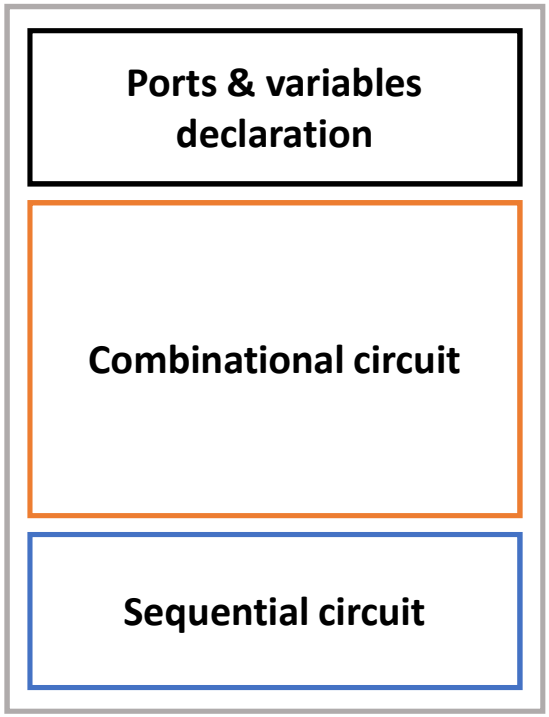
- Design logic to implement synchronous circuit
  - Dataflow between registers (Sequential Logic)
  - Logical operation with register values (Combinational Logic)

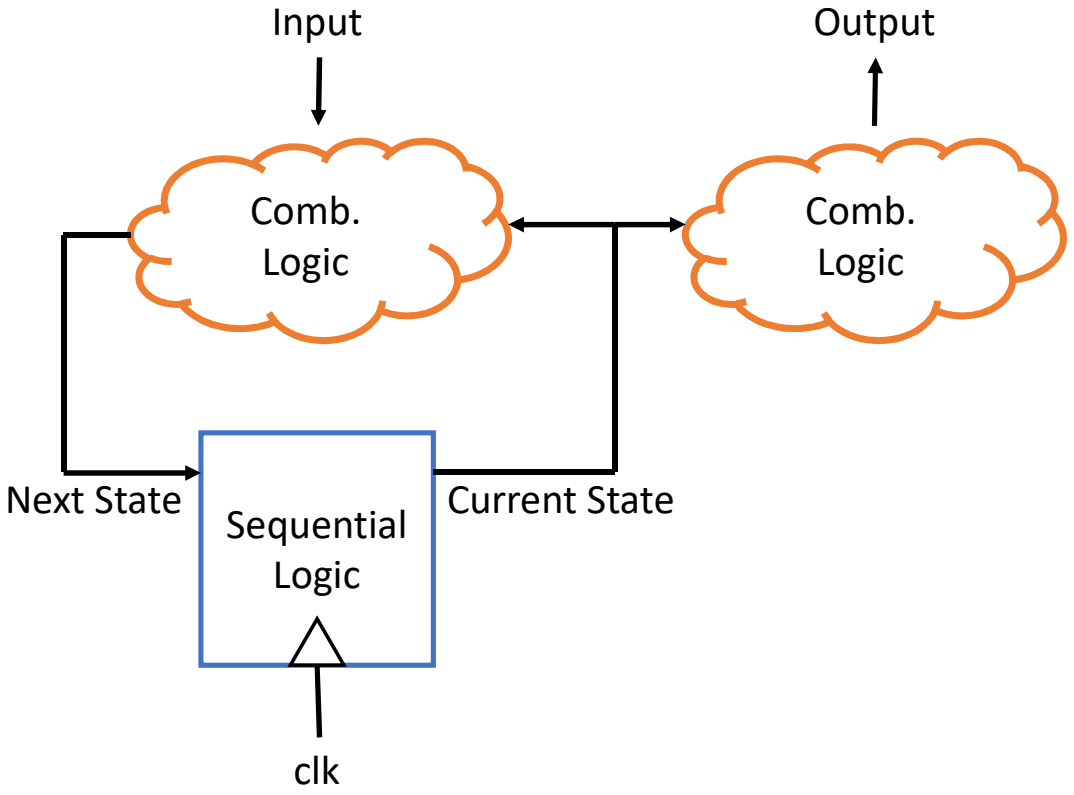


RTL design (review)

- Understand what should be stored in each clock signal
  - Ex) **Current state** of FSMs
- How stored data should be transferred
  - Ex) **Current state** -> **Next state**
- Find out and understand combinational logic for stored data and output
  - Ex) **Next state** and **Output** of FSMs

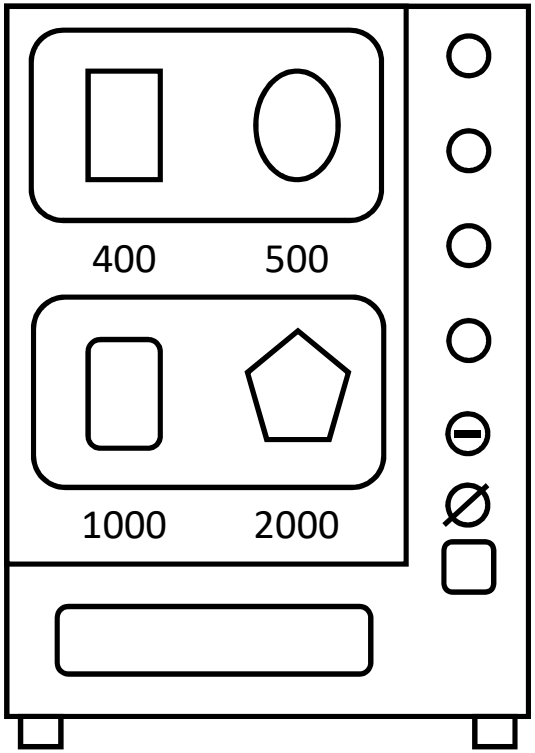
RTL design (review)





Assignment #3:
Vending Machine

Assign #3: Vending Machine



- There is a vending machine with **4 items** in it
  - Price: 400, 500, 1000, 2000
- It receives **3 types of coins** (100, 500, 1000)
- There is a **return button**
  - When triggered, the machine returns change

Assign #3: Vending Machine Functionality

- When a user inserts a coin
  - **Available** item is shown
- When a user selects an item
  - If the selected item is available -> item is sold
  - If the selected item is not available -> nothing happens
  - Assume **infinite amount** of items
- When a user presses return button
  - The total **number of coins** for the change is returned (with the least amount)
    - Ex) When the change is 1700 -> it is returned with 1000*1, 500*1 and 100*2
  - Assume **infinite amount** of coins

Assign #3: Vending Machine Ports

- Implement a simple vending machine with 4 items and 3 types of coins (**do not modify port name of the skeleton code**!! Otherwise, you will get **0 point** for this lab)
  - Input
    - i_input_coin: 3bit input (if x^{th} bit is set to 1, x^{th} coin is inserted)
    - i_select_item: 4bit input (if x^{th} bit is set to 1, x^{th} item is selected)
    - i_trigger_return: trigger return button of the vending machine
  - Output
    - o_available_item: 4bit output (shows availability of each item, 1 if x^{th} item is not out of stock && current change >= price of the item)
    - o_output_item: 4bit output (if x^{th} item is sold, x^{th} bit is set to 1)
    - o_return_coin: the number of coin returned (only if the machine has enough change) when received i_trigger_return
    - o_current_total: the sum of money a user has inserted and used up (total amount of change)

Assign #3: Vending Machine (Extra Credit)

- Two extra credit designs

- 1) Assume **finite amount** of items
  - Assume an item is not available if is out of stock

- 2) Assume **finite amount** of coins
  - Make your **own functionality** for returning coin if the amount of change is finite

- **Make your own test bench** to prove your functionality is error free to receive the full extra credit