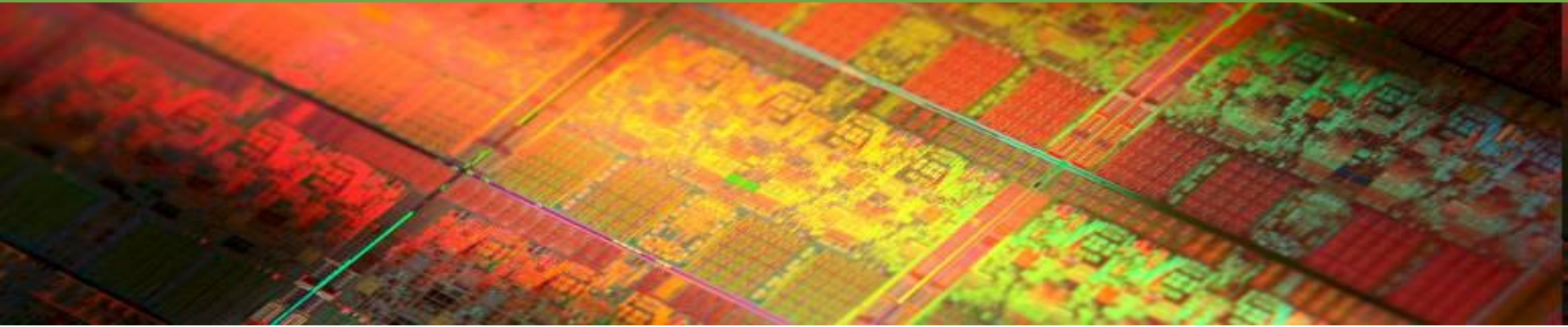


EE 2000

Logic Circuit Design (Fall 2018)



Tutorial 7

An in-depth State Machine
Logic Design

Instructor: Dr. Ray C.C. Cheung

Objectives

- Learn to design a control unit based on a state machine.
- Describe a practical design with VHDL.
- Understand Mealy and Moore Machine.
- Simulate and test your design.

Reviews

Some important concepts:

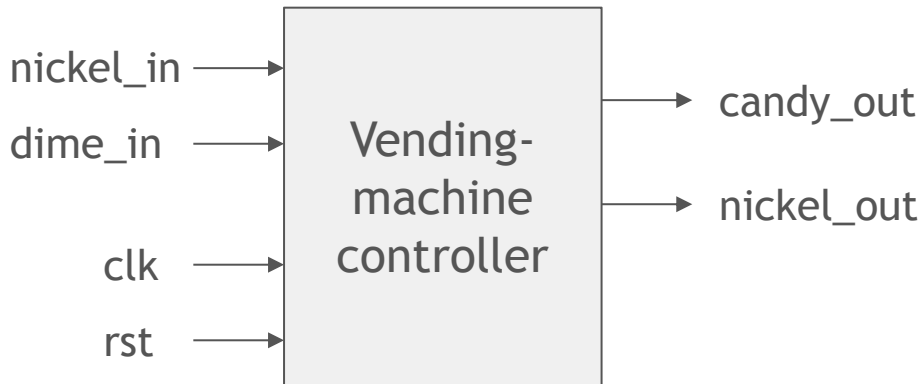
- Combinational & sequential logic
- Synchronous & asynchronous systems
- Mealy & Moore machine

Problem

- We will design a controller for a vending machine.
- It sells candy costing 15 cents and accepts 5 cents and 10 cents.
- It should be able to give candy and correct change as the output.

Analysis

- I/O ports



Descriptions	
clk	Clock signal
rst	Reset signal
nickel_in	5 cents coin is deposited.
dime_in	10 cents coin is deposited.
candy_out	Set to high when input reaches 15.
nickel_out	Set to high when change is due.

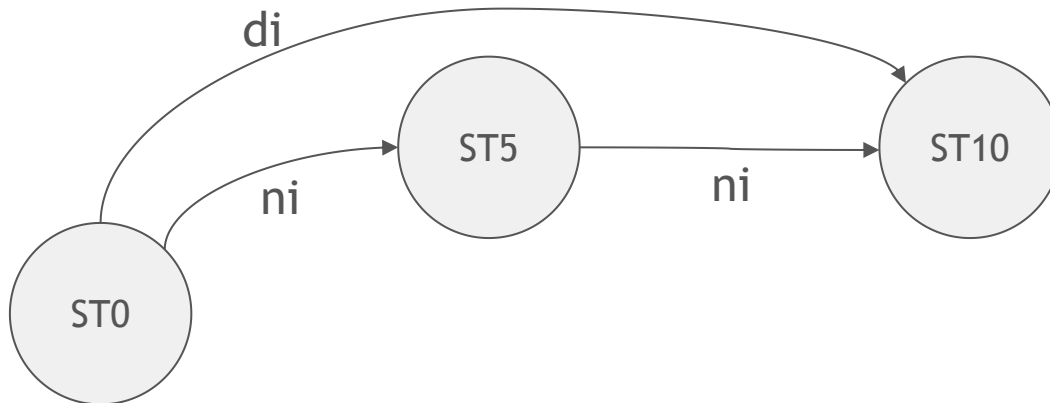
State transition

- We use number of cents stored in the controller as the states.
- Initially, we have not deposited coins, the state is **ST0**.



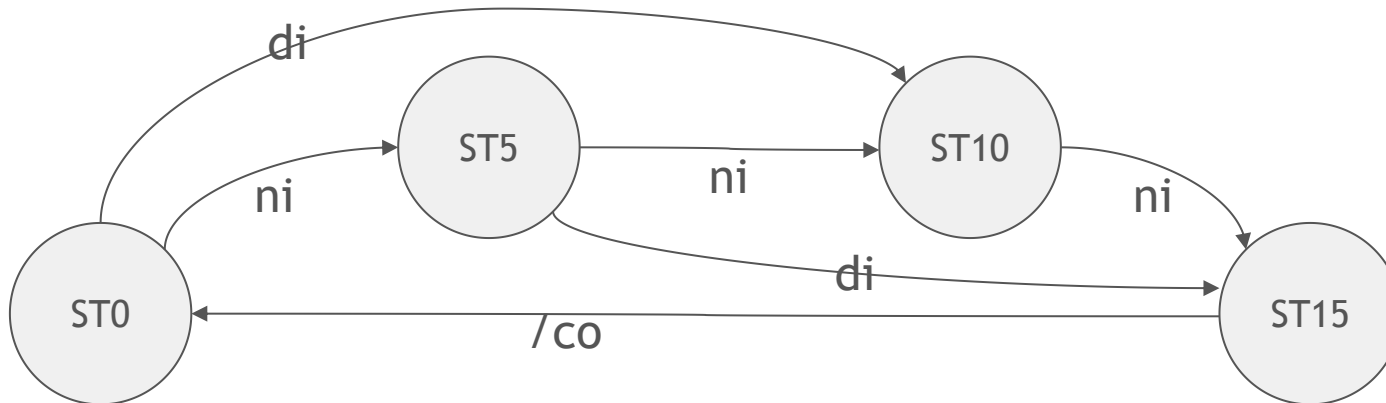
State transition

- We can input either 5 cents or 10 cents.
- One assumption, *dime_in* and *nickel_in* won't be high at the same time.



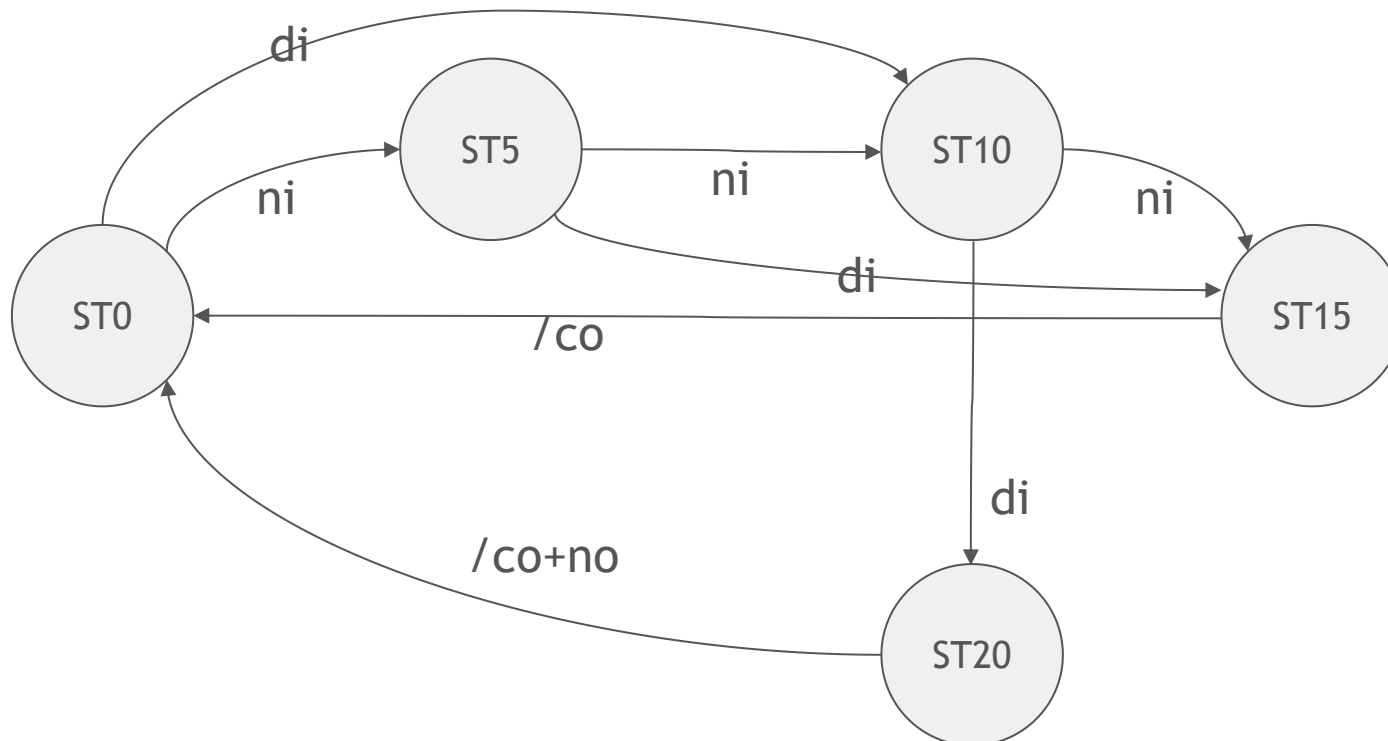
State transition

- If current money reaches 15, candy_out is set to high and go back to **ST0**.



State transition

- Sometimes, you may input more than 15 cents, the controller should give candy as well as the change.



VHDL Step by step

Define states



Describe state transition



Describe transition condition (nx_state) and output



Simulation, Synthesis and Implementation

Define states and state transition

- It is the same as your last tutorial.

```
type state is (ST0, ST5, ST10, ST15, ST20);  
signal pr_state, nx_state: state;
```

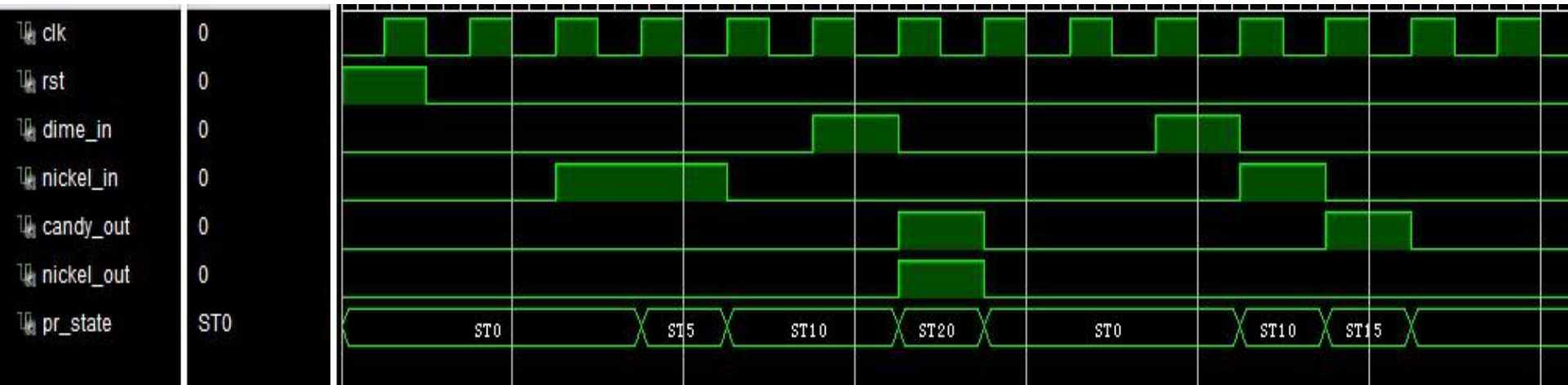
```
process (clk, rst)  
begin  
    if (rst = '1') then  
        pr_state <= ST0;  
    elsif (clk'event and clk='1') then  
        pr_state <= nx_state;  
    end if;  
end process;
```

Combinational logic

Here is a template
for you.

```
process (pr_state, nickel_in, dime_in)
begin
  case pr_state is
    when ST0 =>
      candy_out <= '0';
      nickel_out <= '0';
      if (nickel_in='1') then
        nx_state <= ST5;
      elsif (dime_in='1') then
        nx_state <= ST10;
      else
        nx_state <= ST0;
      end if;
      -- Complete your code here.
    end case;
  end process;
```

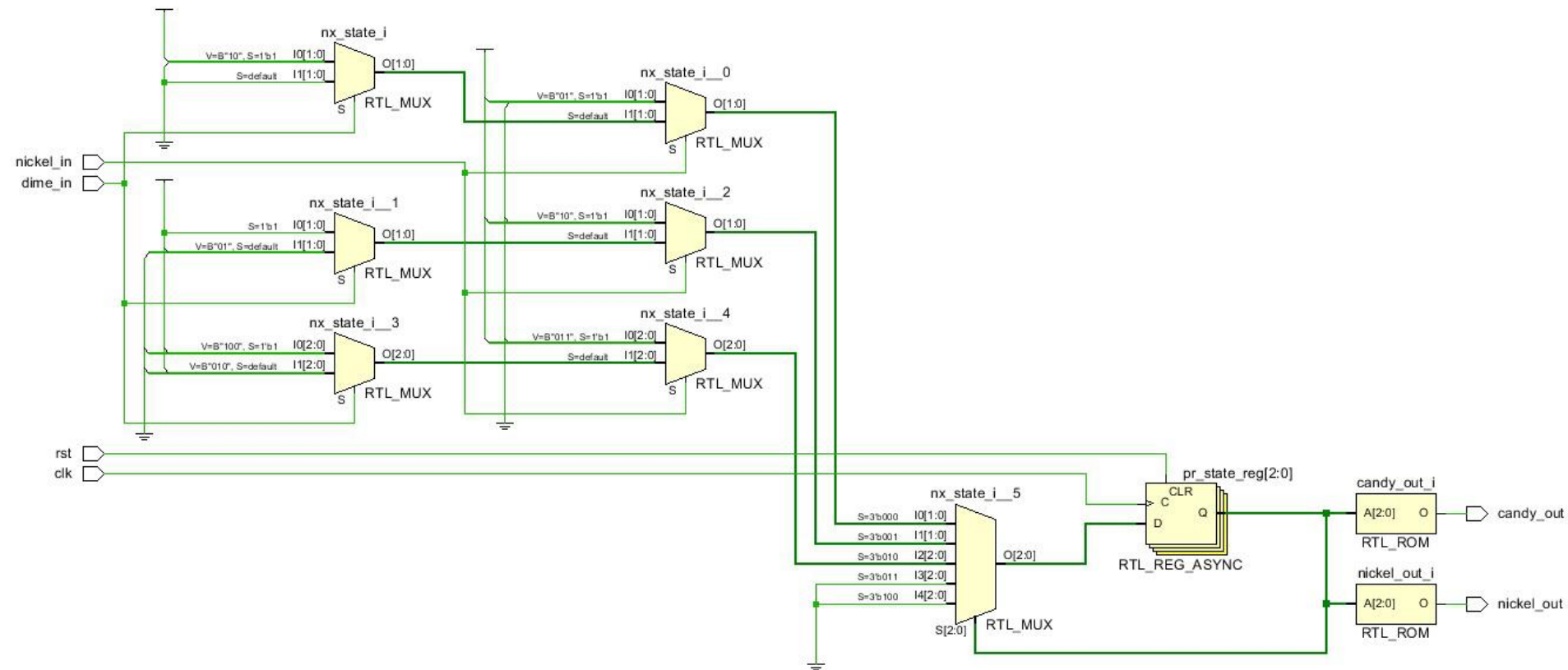
Simulation results



Return candy
and change

Only return
candy

RTL analysis



Reference

- More details about finite state machine
 - <https://www.youtube.com/watch?v=-WlrfGUg6tk>
- Xilinx finite state machine lab workbook
 - <https://www.xilinx.com/support/documentation/university/Vivado-Teaching/HDL-Design/2015x/VHDL/docs-pdf/lab10.pdf>

Tutorial - Remarks

- Please raise your hand if you have questions!
- You can also refer to the course Canvas homepage.
- Get familiar with state machine design.
- Learn a good coding style for state machine description.
- Practice at home with more hands-on experience.