Welcome to The Logic Design Lab!

Fall 2021
Sequential Circuit II

Prof. Chun-Yi Lee

Department of Computer Science
National Tsing Hua University

Agenda

- Announcement
- Clarification
- Sequential circuits
- Finite state machine concept

Today's class will help you:

1. Understand the difference between synchronous reset and asynchronous reset
2. Understand concept of the global clock signal
3. Understand the concept of finite state machine

Picture cited from http://www.princeton.edu for educational purpose only

Announcements

- Lab 3
  - Basic questions demonstration on **10/21/2021 (Thu)**
  - Advanced questions and FPGA demonstration due on **10/28/2021 (Thu)**
- Midterm Exam
  - Computer-based exam on **11/4/2021 (Thu)**

Agenda

- Announcement
- Clarification
- Sequential circuits
- Finite state machine concept

Today's class will help you:

1. Understand the difference between synchronous reset and asynchronous reset
2. Understand concept of the global clock signal
3. Understand the concept of finite state machine

Picture cited from http://www.princeton.edu for educational purpose only

Variable Assignment

- Only assign values to a variable in a **single always block**

☺

```
module Counter (clk, Reset, Out);
input       clk, Reset;
output      Out;
reg         [3:0] counter;


always @(posedge clk) begin
    if (!Reset)
        counter <= 4’b0000;
    else
        counter <= counter + 1’b1;
end

assign out = counter [3];

endmodule
```

✗

```
module Counter (clk, Reset, Out);
input       clk, Reset;
output      Out;
reg         [3:0] counter;


always @(posedge clk) begin
    if (!Reset)
        counter <= 4’b0000;
    else
        counter <= counter + 1’b1;
end

always @ (*) begin
        if (counter == 4’b1010)
                counter = 4’b0000;
end

assign out = counter [3];

endmodule
```

Agenda

- Announcement
- Clarification
- Sequential circuits
  - Reset
  - Clock
- Finite state machine concept

Today's class will help you:

1. Understand the difference between synchronous reset and asynchronous reset
2. Understand concept of the global clock signal
3. Understand the concept of finite state machine

Picture cited from http://www.princeton.edu for educational purpose only

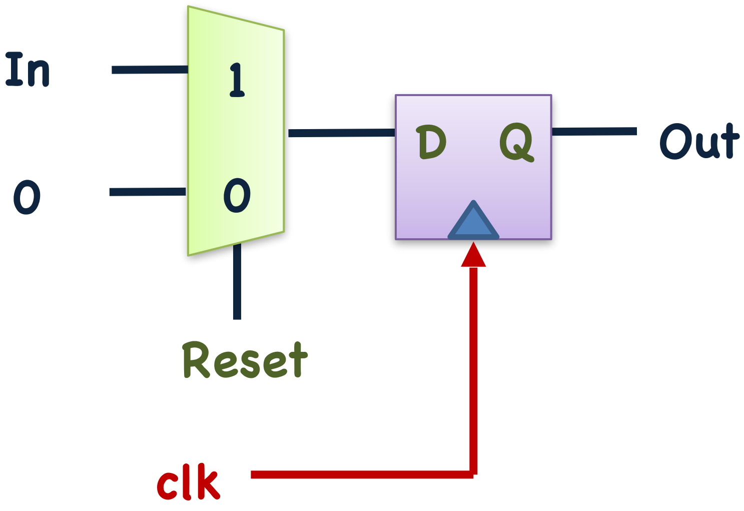
Why Do We Need a Reset?

- Initialize your flip-flops
- If not properly reset, your flip-flops may get unknown values (e.g., **XXXX** in your counters)

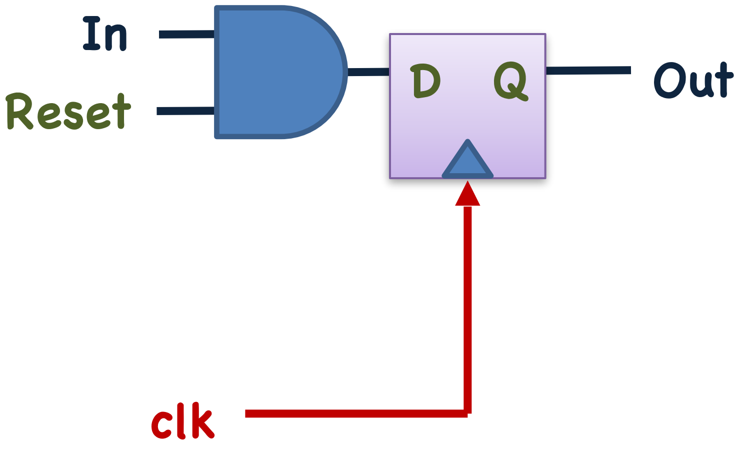
Synchronous Reset

- Reset the value stored in a D Flip-Flop to 0
- Triggered by clock edges
- Assume that **Q = 1’b0** when Reset is set to **1’b0**

Scheme 1



Scheme 2

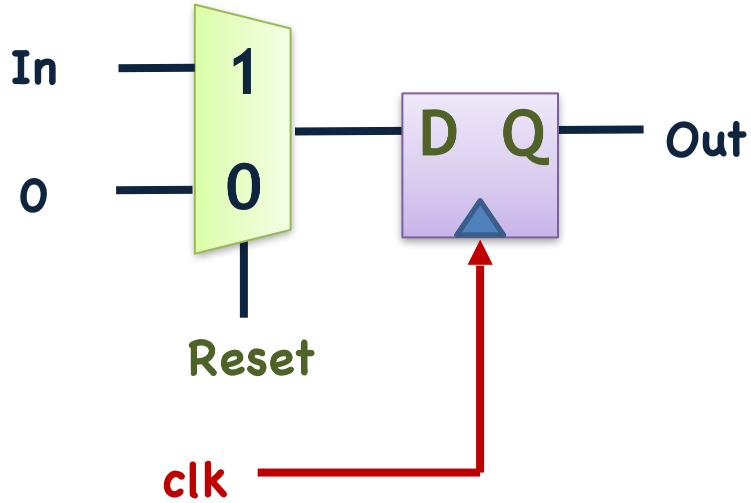


Synchronous Reset in Always Block

- **Prioritize** your reset signal over the rest of the inputs
- In your testbench, **initialize your reset signal first**
- You **shall not** use any initial block in your design modules

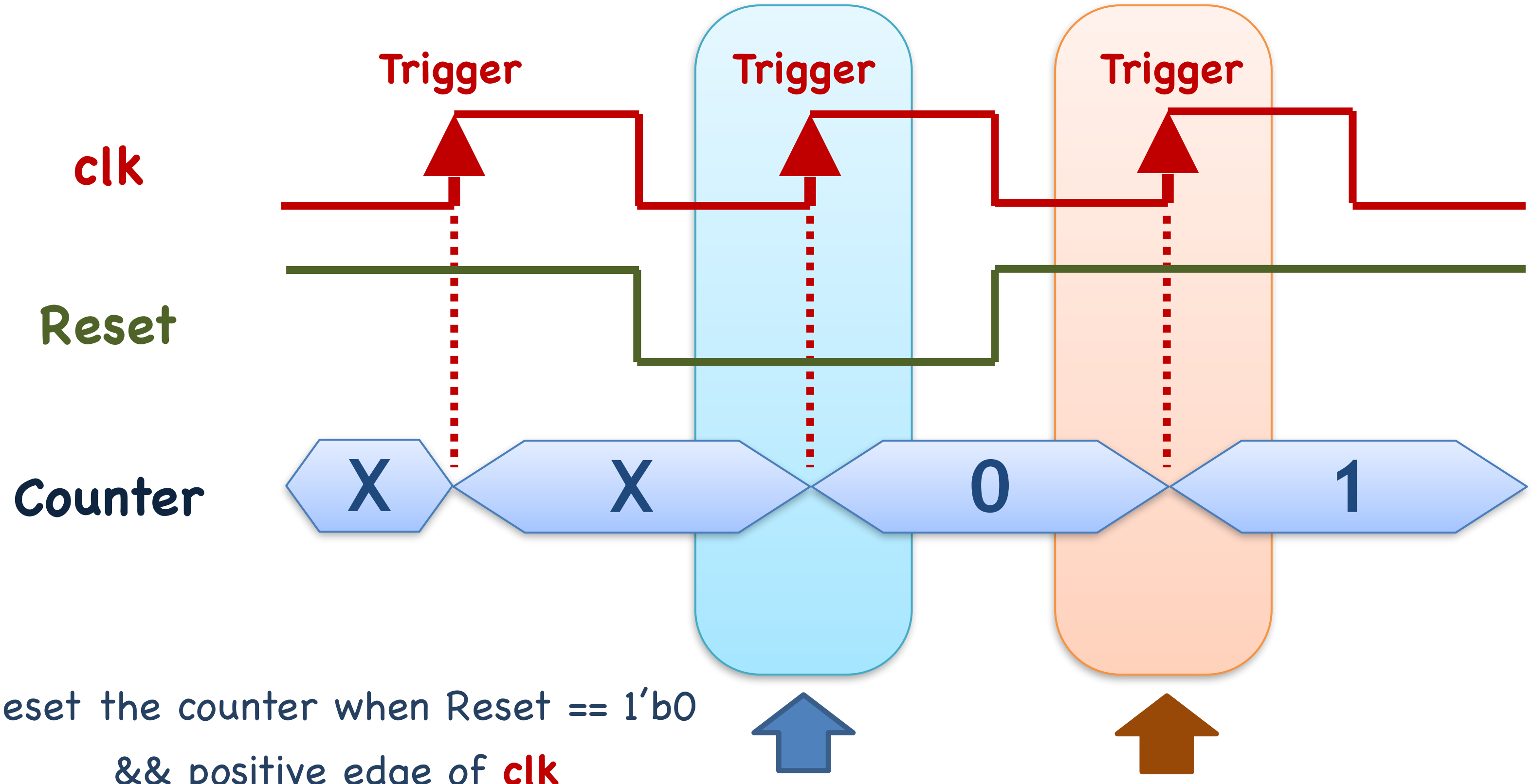
```
module My_Test_Flip_Flop (Out, In, clk, Reset);
  input     In, clk, Reset;
  output    Out;
  reg       Out;

always @ (posedge clk)
  begin
    if (Reset == 1'b0)            Out <= 1'b0;
    else                          Out <= In;
  end
endmodule
```

Declaration of Synchronous Behavior



Synchronous Reset in Waveform

- Reset the flip-flops at **clock edges**

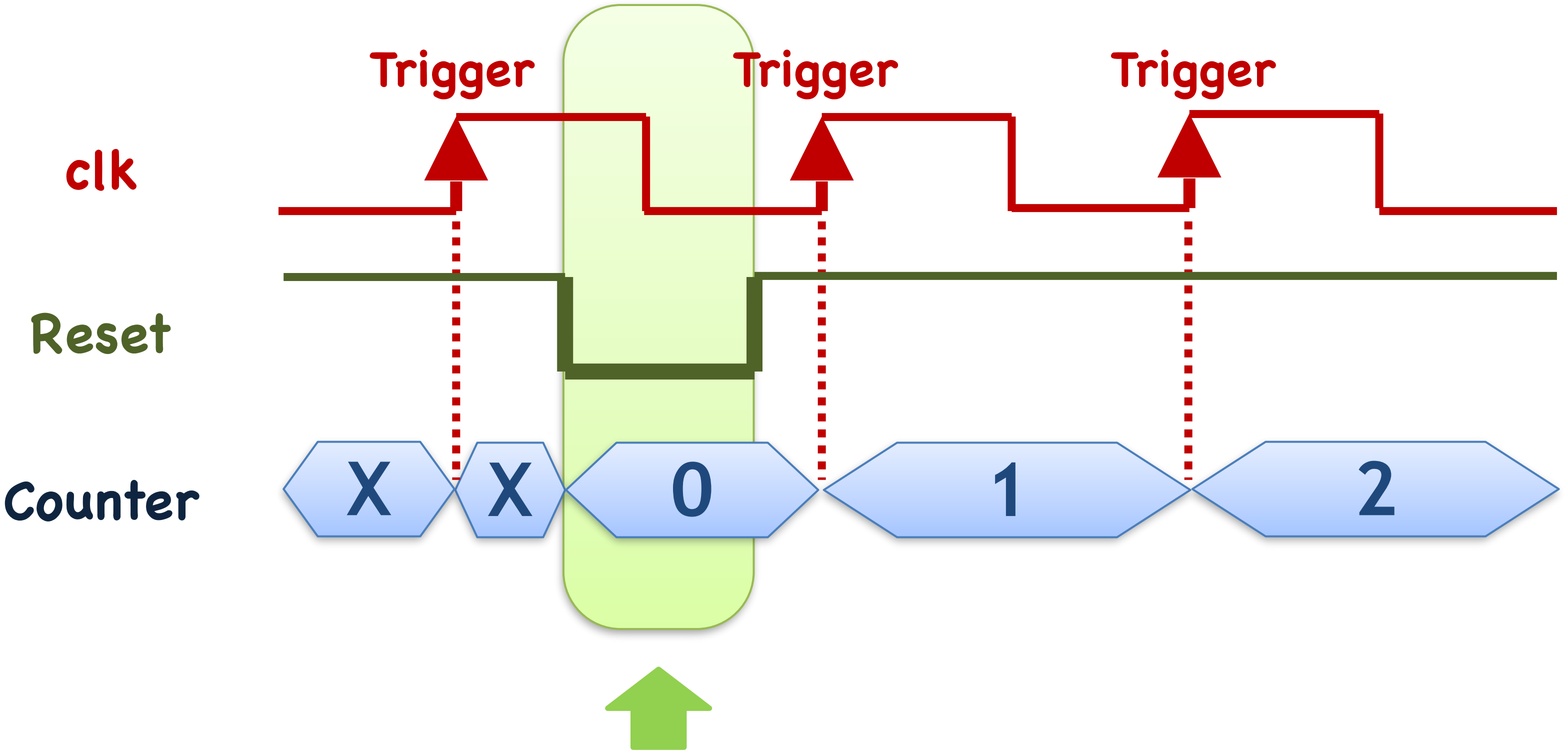


Reset the counter when Reset == 1'b0
&& positive edge of **clk**

Counter begins again

Asynchronous Reset in Waveform

- Reset the flip-flops **ANY TIME**
- Concept similar to **Latches**



Reset is done regardless of the **clk** signal

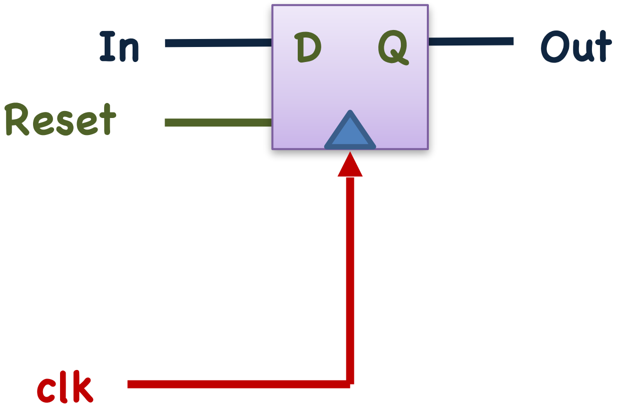
Asynchronous Reset in Always Block

- Add “**negedge Reset**” or “**posedge Reset**” in always statements
- Reset the flip-flops regardless of **clk**
- **Reset** signal gets the **highest priority**

```
module My_Test_Flip_Flop (Out, In, clk, Reset);
  input    In, clk, Reset;
  output   Out;
  reg      Out;

always @ (posedge clk or negedge Reset)
  begin
    if (Reset == 1’b0)           Out <= 1’b0;
    else                         Out <= In;
  end
endmodule
```

Declaration of Asynchronous Behavior ↓

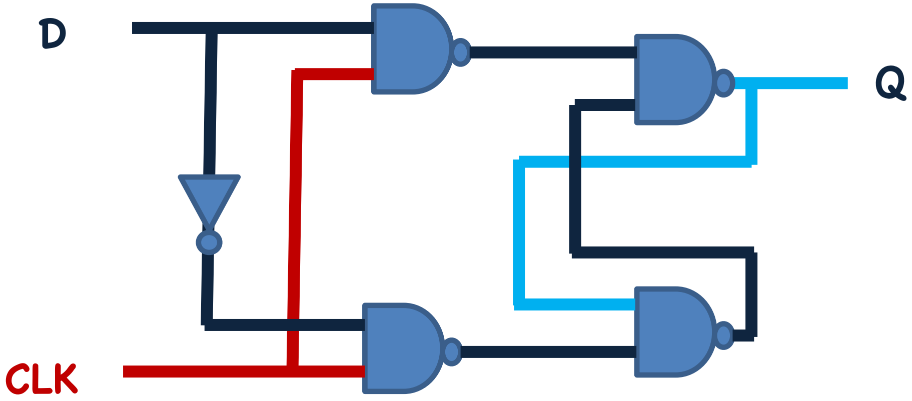
D flip-flop has an additional Reset input port



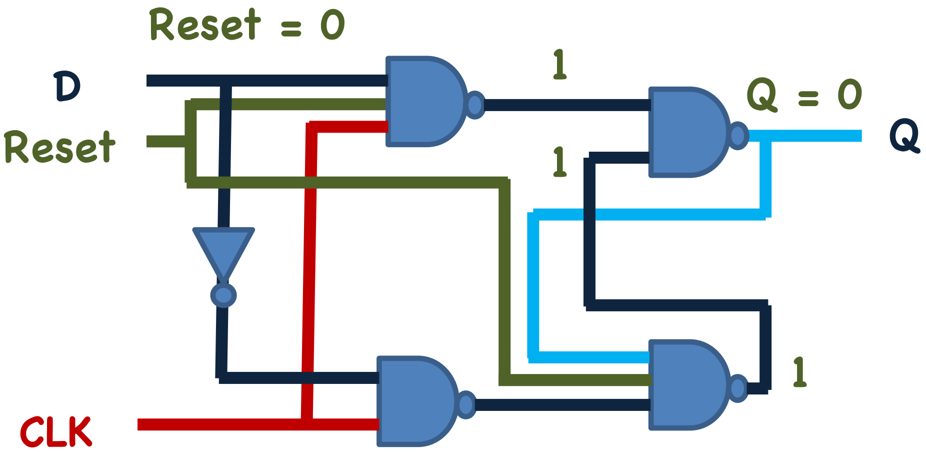
Out is reset to 0 when Reset goes down to zero

Asynchronous Reset in Latches

Latch in Lab 1



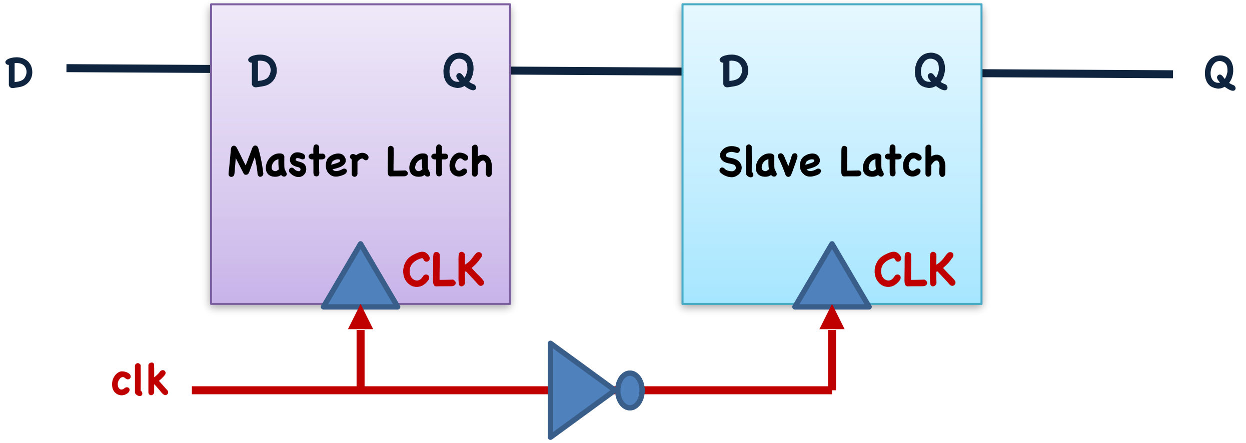
Latch with Asynchronous Reset



Asynchronous Reset in DFFs

DFF in Lab 1

(Synchronous)

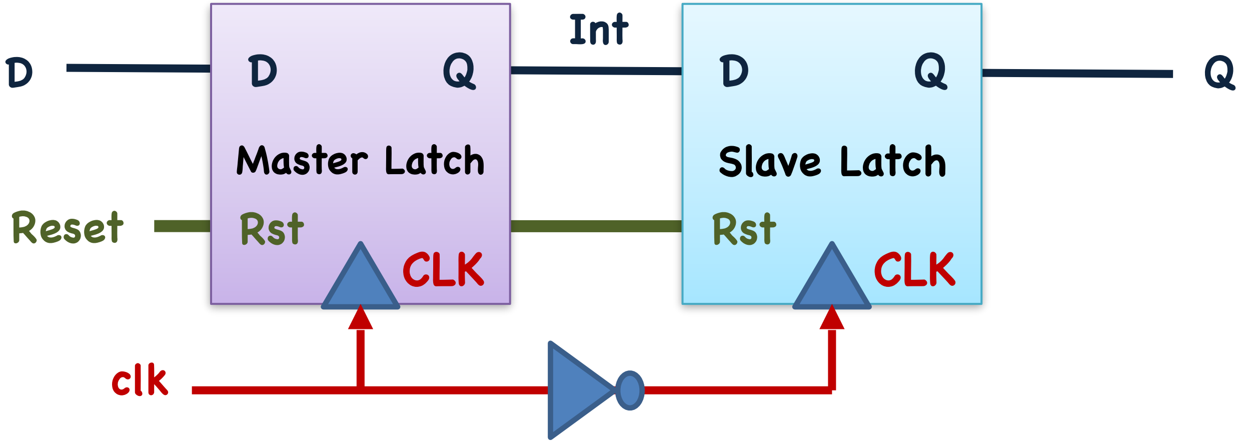


DFF with

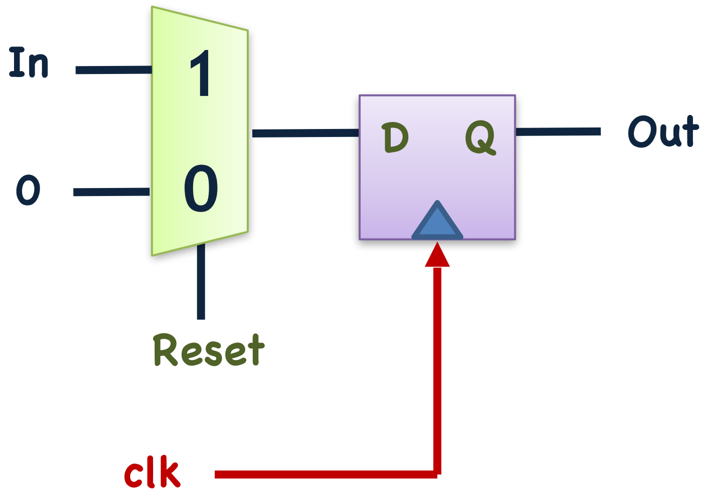
Asynchronous

Reset

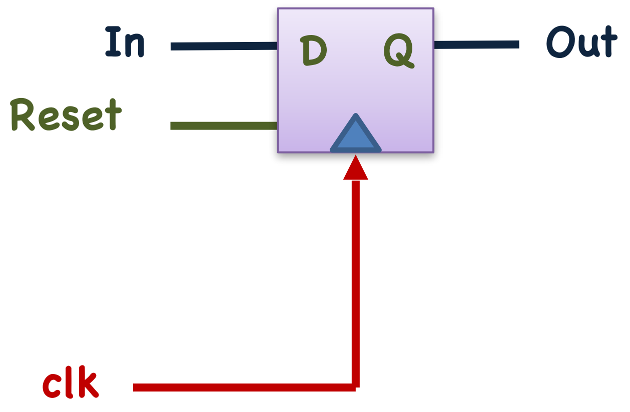
When Reset = 0 Int = 0 Q = 0



Synchronous vs. Asynchronous Reset



Synchronous Reset

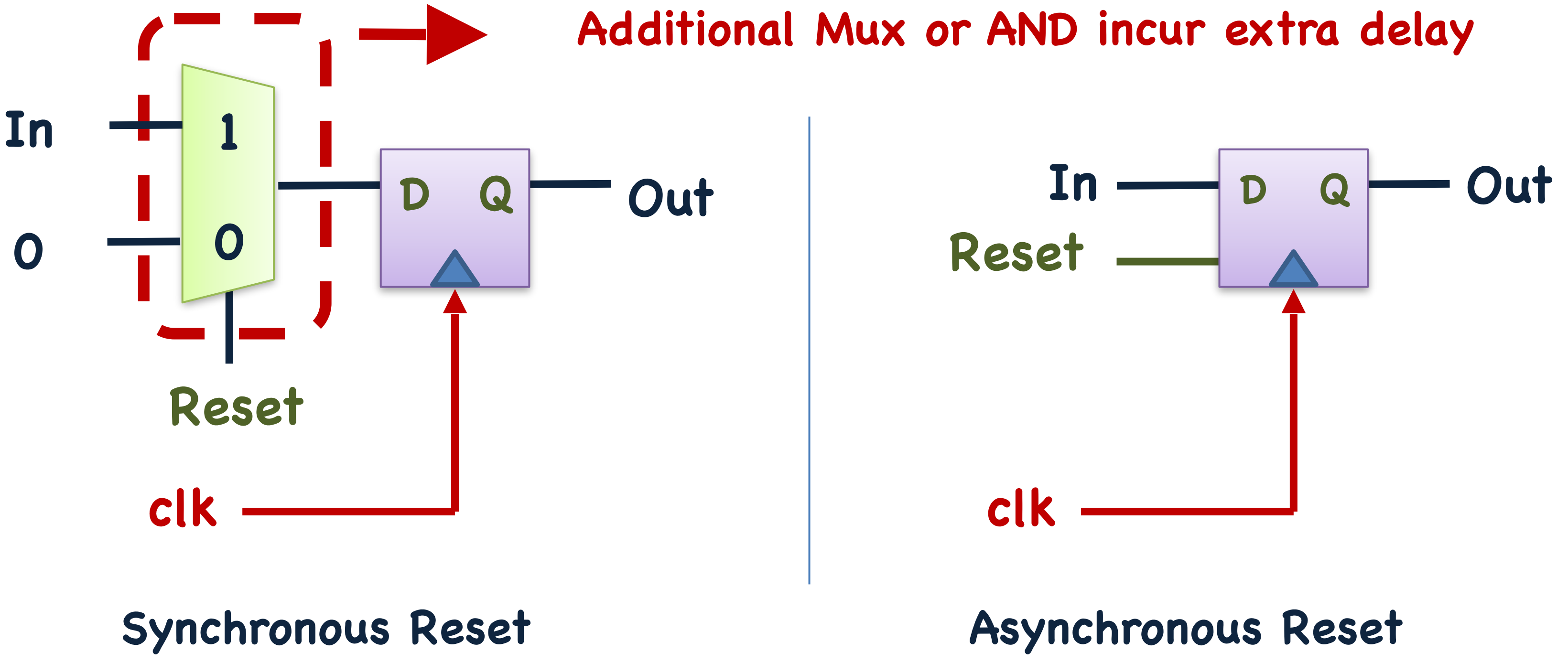


Asynchronous Reset

	Synchronous Reset	Asynchronous Reset
Clock	Reset at clock edges	Regardless of clock
Always block	always @ (posedge clk)	always @ (posedge clk or **negedge Reset**)
Logic	Additional Mux or AND gate	Additional reset input port in DFF

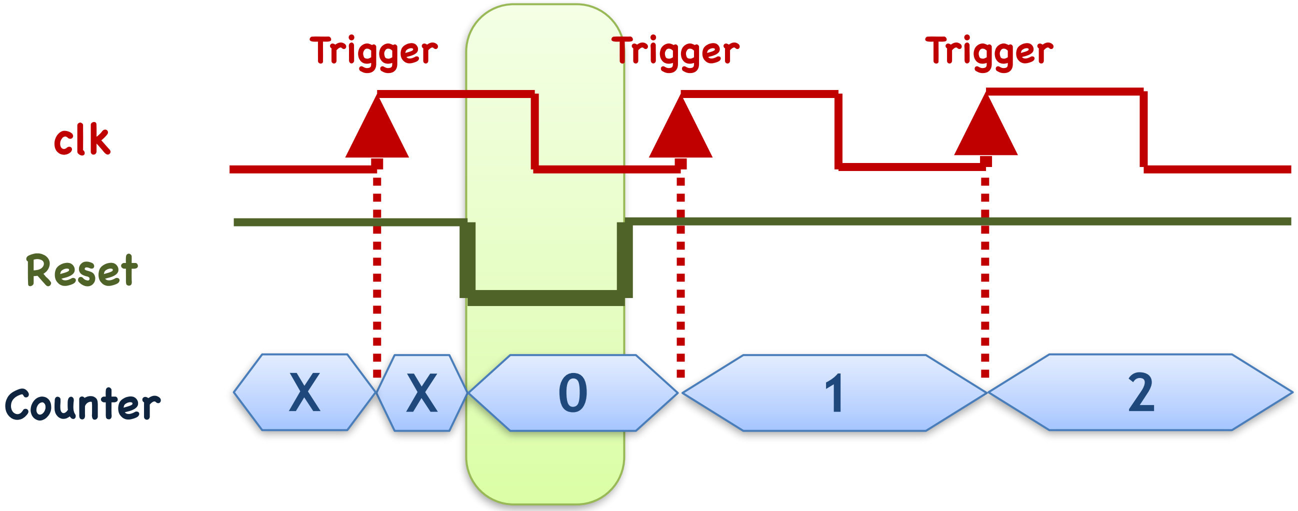
Advantages of Asynchronous Reset

- Fast
  - No additional Mux or AND
- Reset **ANYTIME**, no clock signal is required



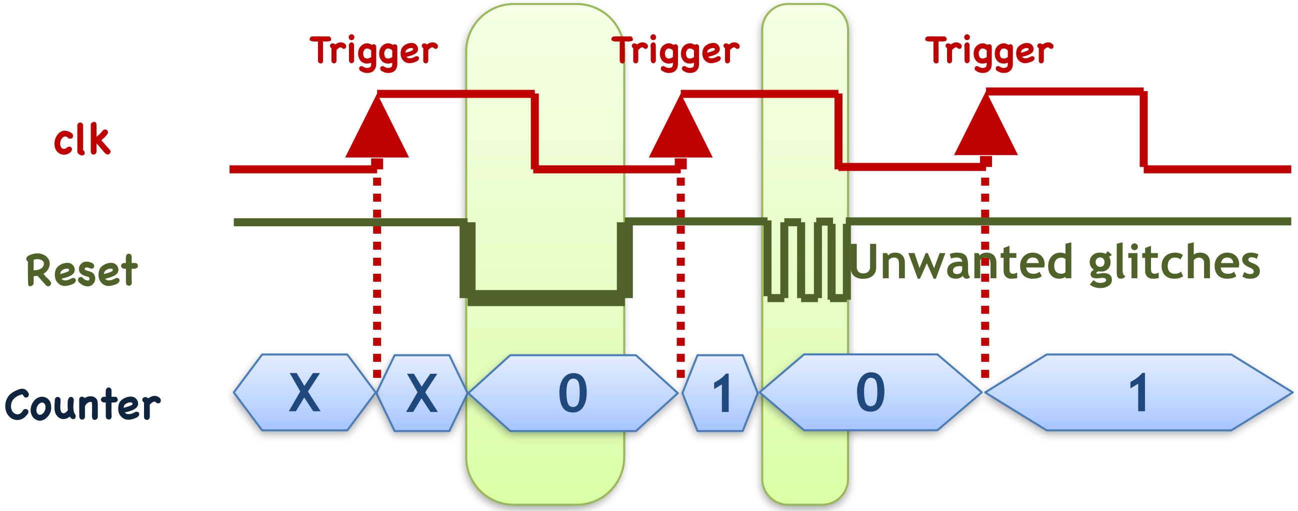
Disadvantages of Asynchronous Reset

Ideal



In reality

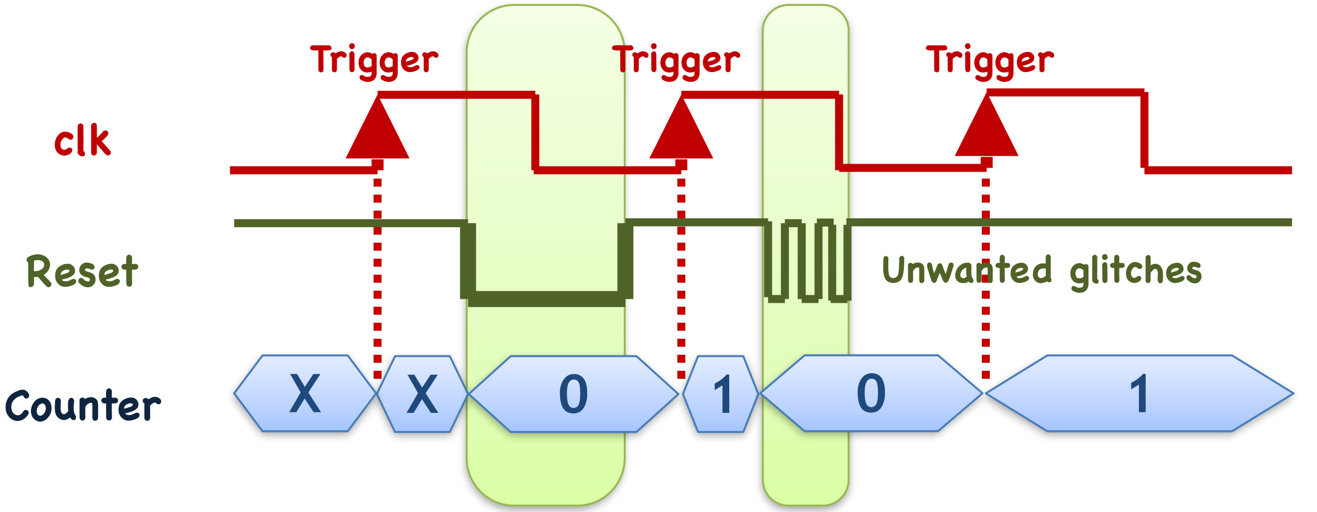
Asynchronous Reset is vulnerable to glitches



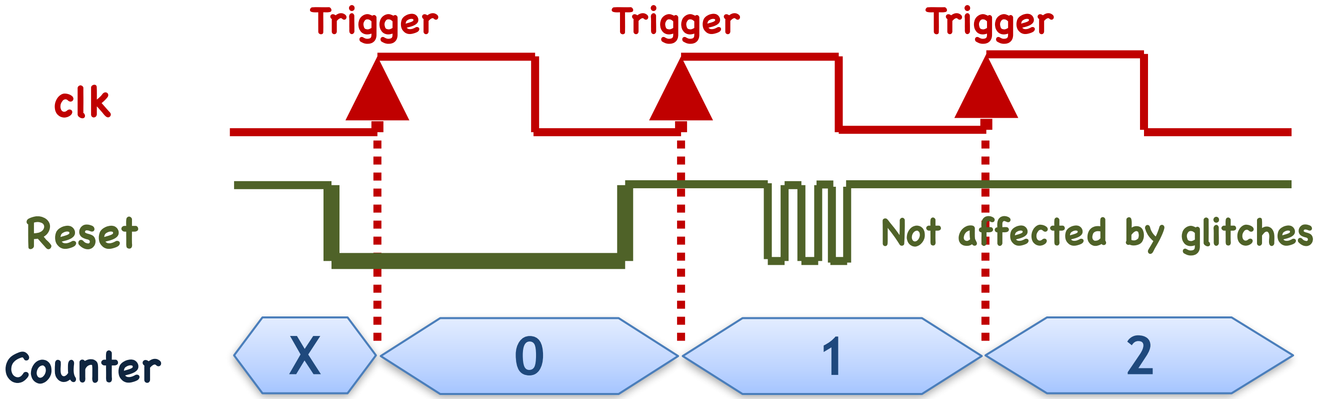
Advantages of Synchronous Reset

- Synchronous Reset is not affected by glitches
  - Unless glitches occur right at clock edges

Asynchronous Reset



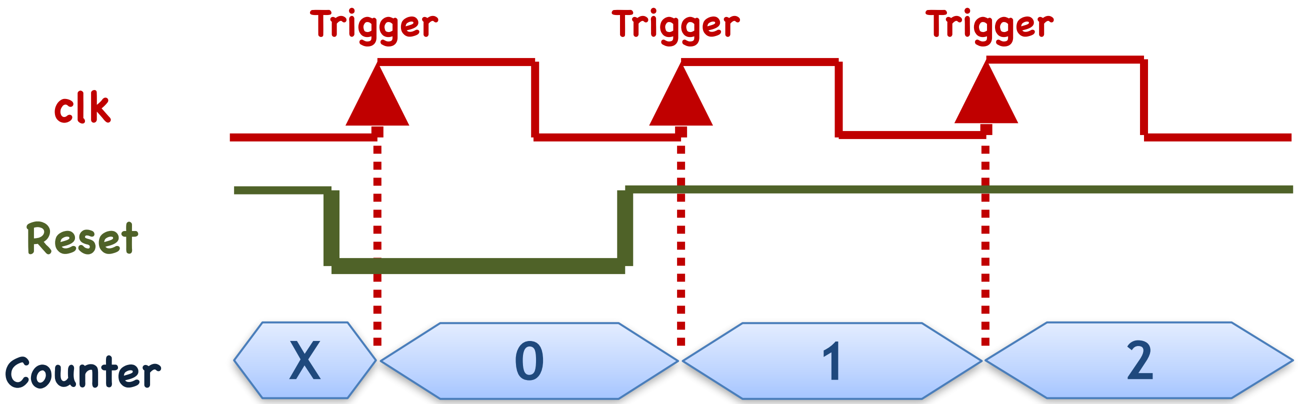
Synchronous Reset



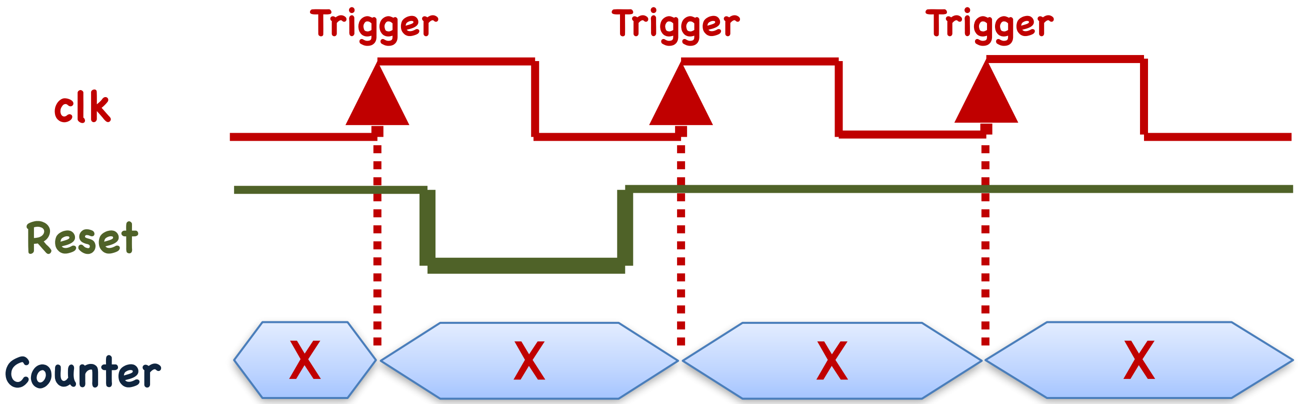
Disadvantages of Synchronous Reset

- Slower
- If Reset signal is too short, no DFFs are reset

Reset is long enough



Reset is too short



Summary of Pros and Cons

- Here we summarize the previous slides

Reset type	Advantages	Disadvantages
Synchronous	Not affected by glitches, Completely synchronous circuit	Slow, Reset cannot be too short
Asynchronous	Fast, Reset anytime	Vulnerable to glitches, Larger DFFs

Agenda

- Announcement
- Clarification
- Sequential circuits
    - Reset
    - Clock
- **Finite state machine concept**

Today's class will help you:

1. Understand the difference between synchronous reset and asynchronous reset
2. Understand concept of the global clock signal
3. Understand the concept of finite state machine

Picture cited from http://www.princeton.edu for educational purpose only

Why do We Need a Clock?

- In real world, we need clocks to tell us what time is it
- …so that we can collaborate no matter where we are

Bob in Antarctica

Stuart in Paris

Kevin in New York

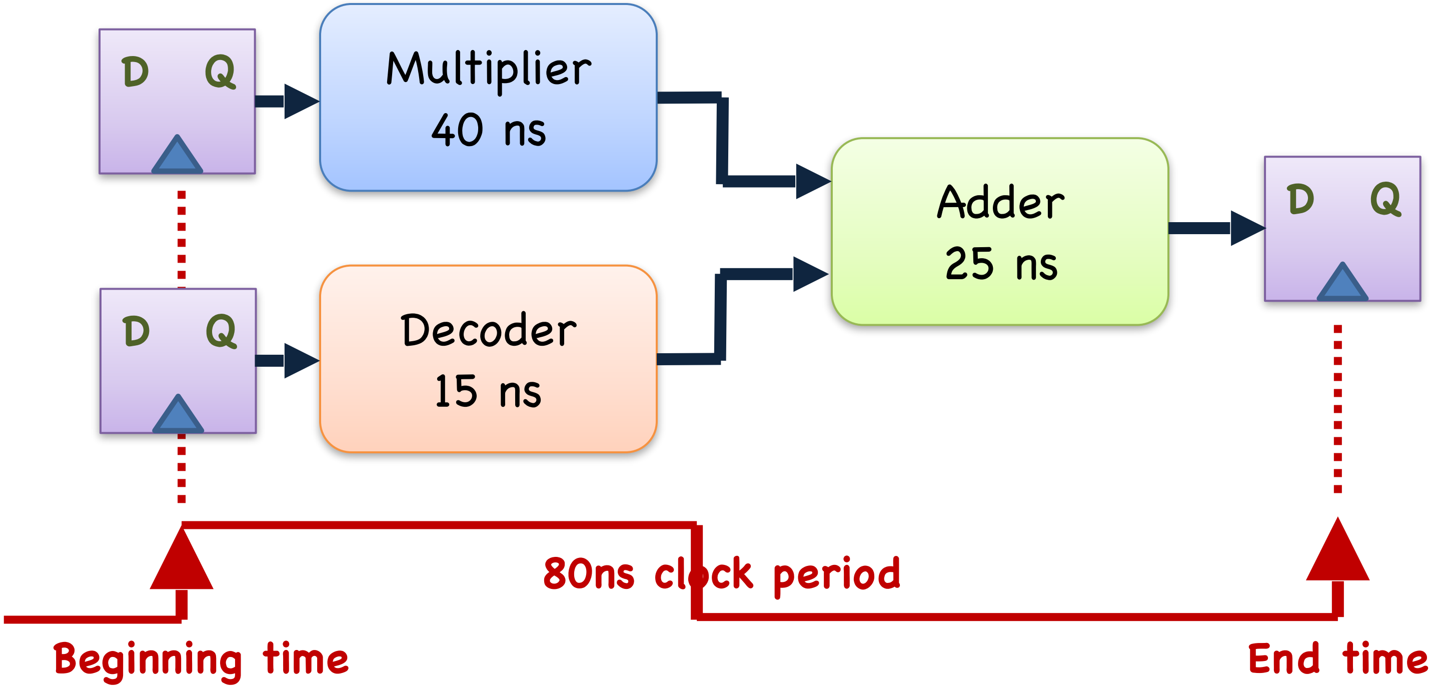
How do they collaborate?

They need to know each other's time

*Pictures cited from www.universalstudioshollywood.com for education purpose only

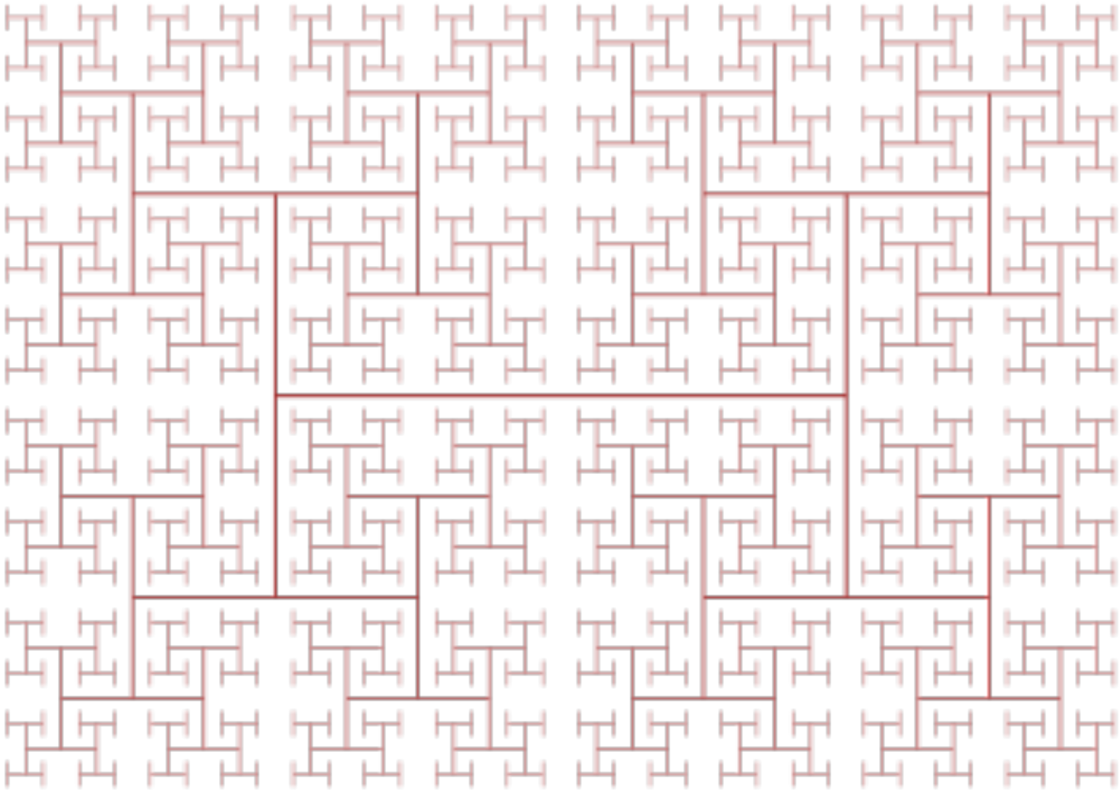
Clock Signal for Logic Blocks

- Clock signals and DFFs are used to synchronize the beginning and end time of a computation
- Ensure the progresses of all the blocks
  - Similar to the concept of our beginning date and deadline of homework



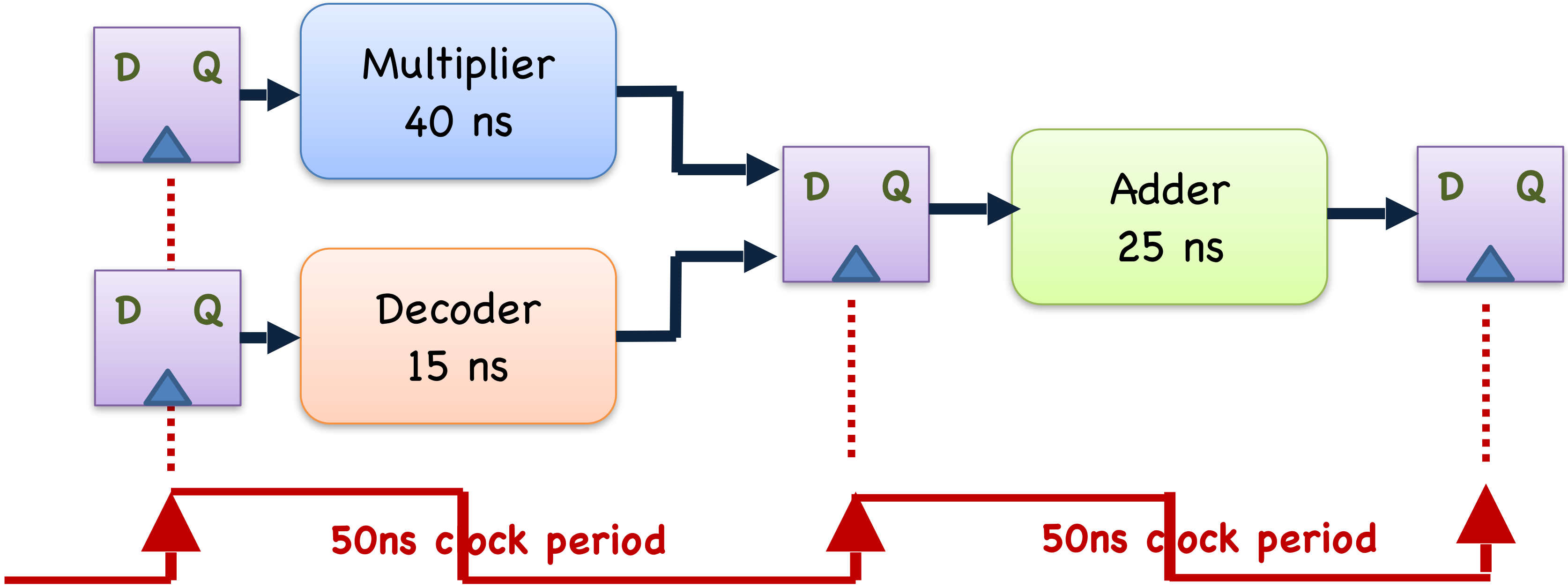
Clock Signal Distribution

- Clock signals are distributed all over the chip
- Different ways of distribution are possible



Deal with Faster Clock Requirements

- Divide your designs, and insert more pipeline stages
- Simply and redesign your logic
- Use faster devices



How Companies Decide Clock Freq?

- Basically, by **past experience** and **negotiation**

CEO:
We need 3GHz to compete
with our rivals

Higher management level:
We need to get everything
done in 100 clock cycles

Your manager:
We need to get our stuffs
done in 10 clock cycles

Your responsibility:

Mission impossible

*Pictures cited from disney.com and www.paramount.com for education purpose only

Agenda

- Announcement
- Clarification of lab 3
- Sequential circuits
  - Reset
  - Clock
- Finite state machine concept

Today's class will help you:

1. Understand the difference between synchronous reset and asynchronous reset
2. Understand concept of the global clock signal
3. Understand the concept of finite state machine

Picture cited from http://www.princeton.edu for educational purpose only

What is Finite State Machine (FSM)?

- Everything in the world has states, so are you

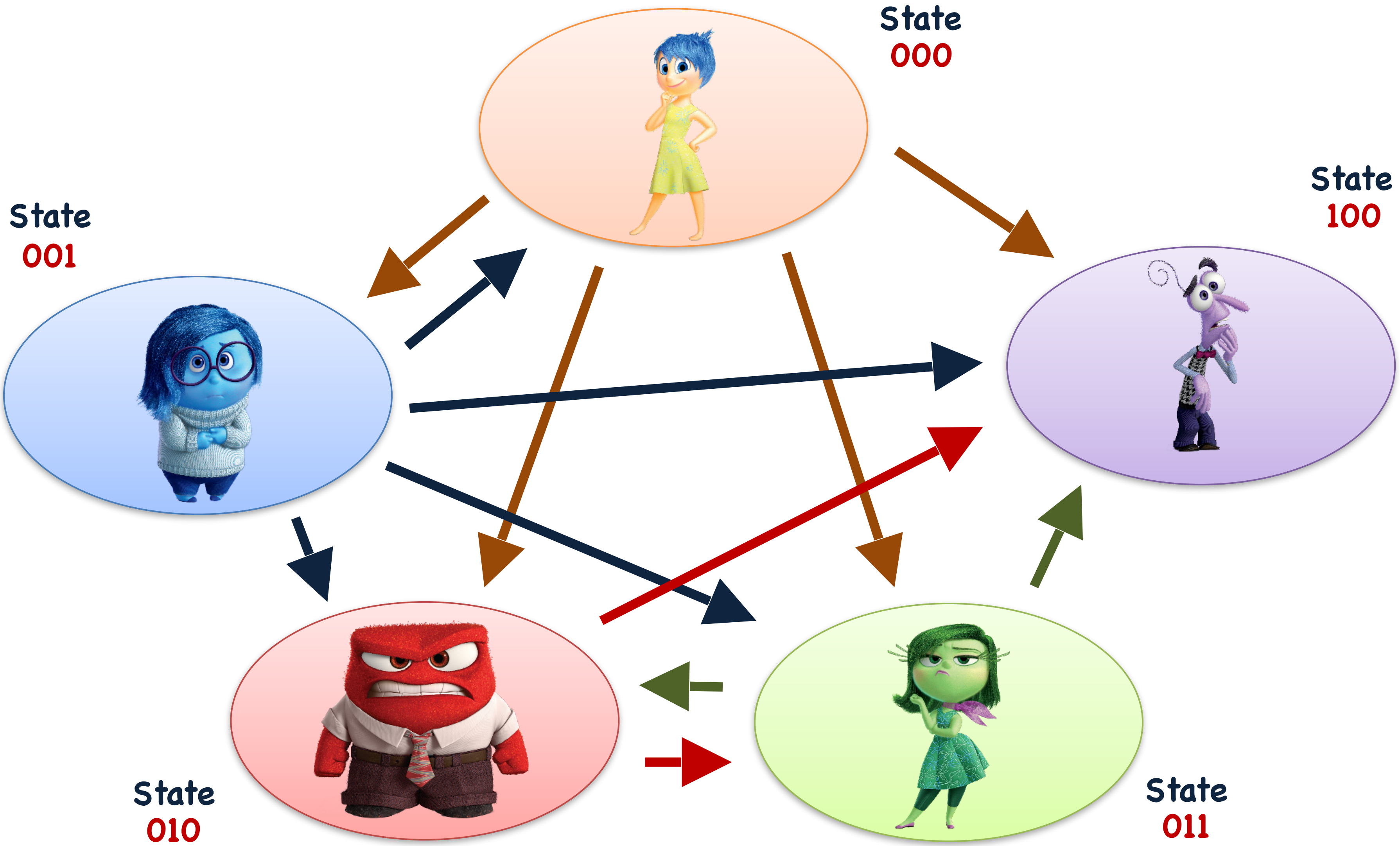
*Pictures cited from disney.com for education purpose only

FSM Can Change States

- States change when events happen
  - https://www.youtube.com/watch?v=WMlzCP5u2w8

*Pictures cited from disney.com for education purpose only

State Transition Diagram



*Pictures cited from disney.com for education purpose only

Thank you for your attention!

*Taken at Lassen Peak at Shasta Caverns, California, USA.
This picture is taken by Chun-Yi Lee himself, who is also a fan of photography