DCS Lab 3
Sequential Circuit 2

葉曜銘

Purpose

- Be familiar with Verilog syntax
  - Avoid multiple-driven net
  - Separate sequential and combinational block
- Be familiar with circuit specification
  - Output data should be strictly **RESET** !!!
- Sequential circuit

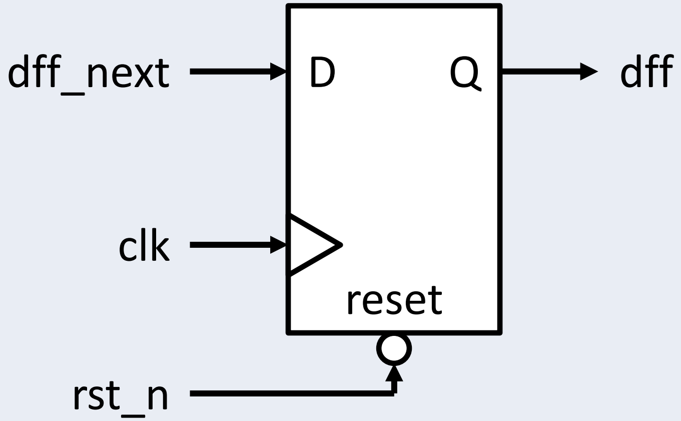
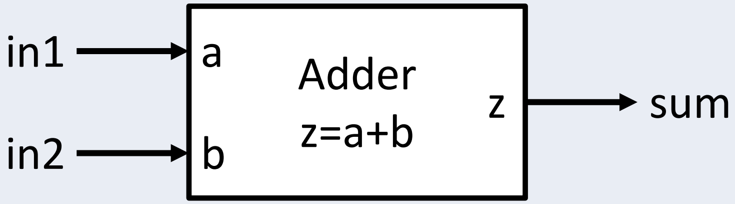
Lab03

- Run the command below to get files for this lab.
  - tar -xvf ~dcsTA01/Lab03.tar

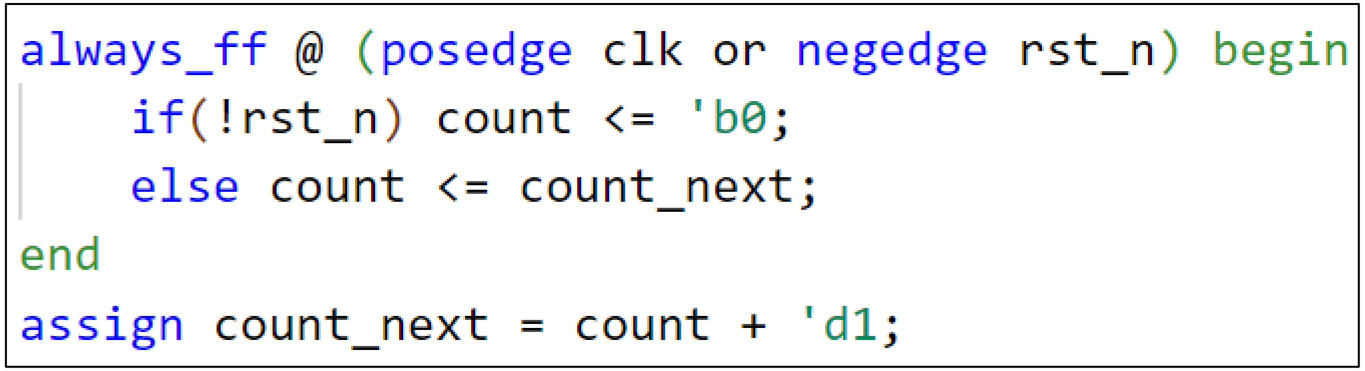
Sequential v.s. Combinational in Verilog

	Sequential Circuit (D Flip-Flop)	Combinational Circuit

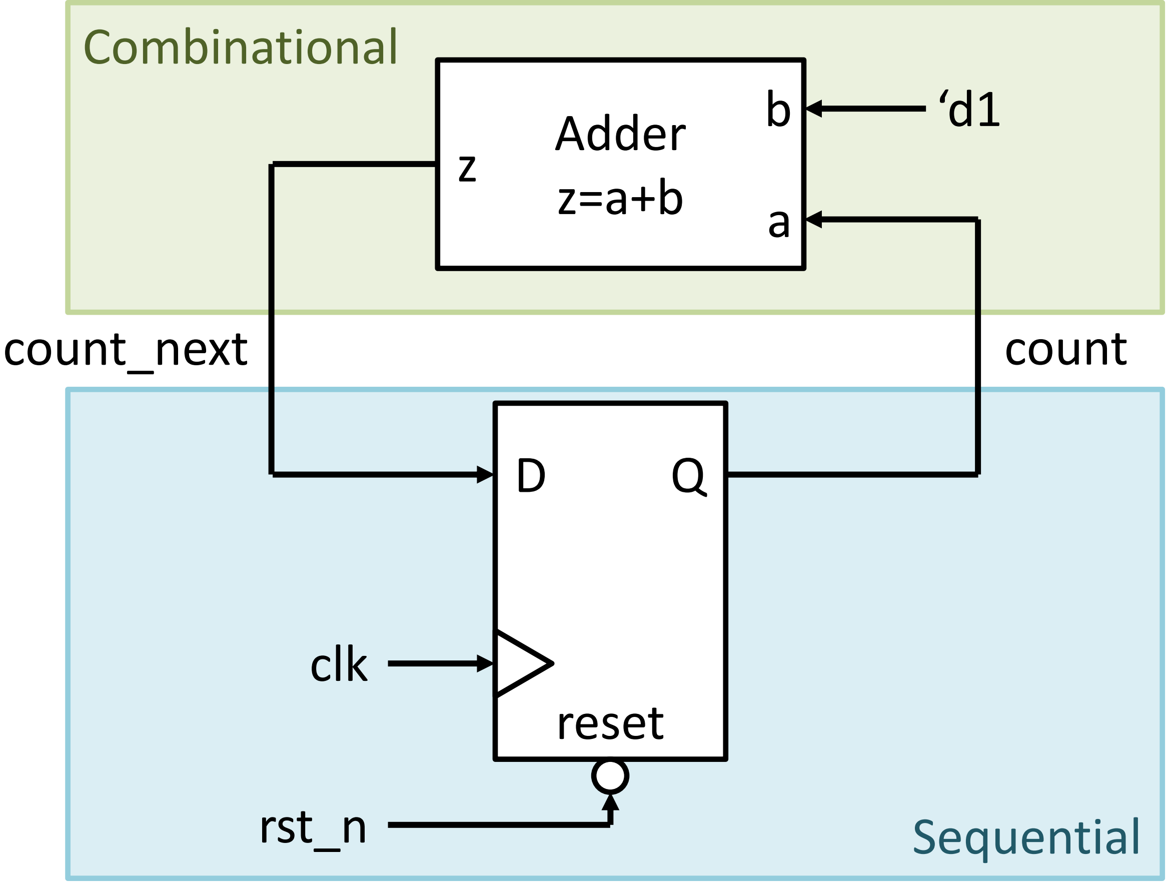
Verilog Code	`always @ (posedge clk or negedge rst_n) begin`
	`    if(!rst_n) dff <= 'b0;`
	`    else dff <= dff_next;`
	`end`
	
	`always_ff @ (posedge clk or negedge rst_n) begin`
	`    if(!rst_n) dff <= 'b0;`
	`    else dff <= dff_next;`
	`end`
		`always @ * begin`
		`    sum = in1 + in2;`
		`end`
		
		`always_comb begin`
		`    sum = in1 + in2;`
		`end`
		
		`assign sum = in1 + in2;`

Block Diagram		

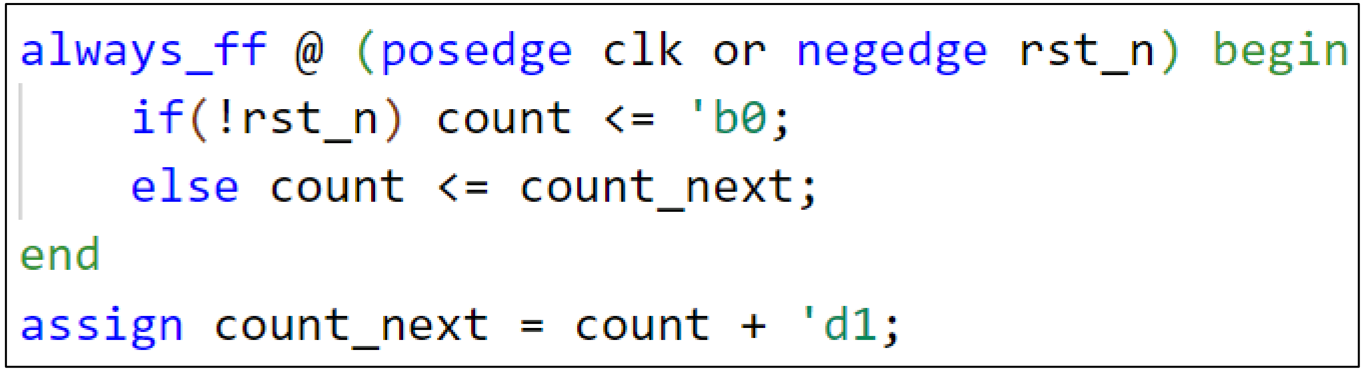
Example Counter



```
always_ff @ (posedge clk or negedge rst_n) begin
    if(!rst_n) count <= 'b0;
    else count <= count_next;
end
assign count_next = count + 'd1;
```

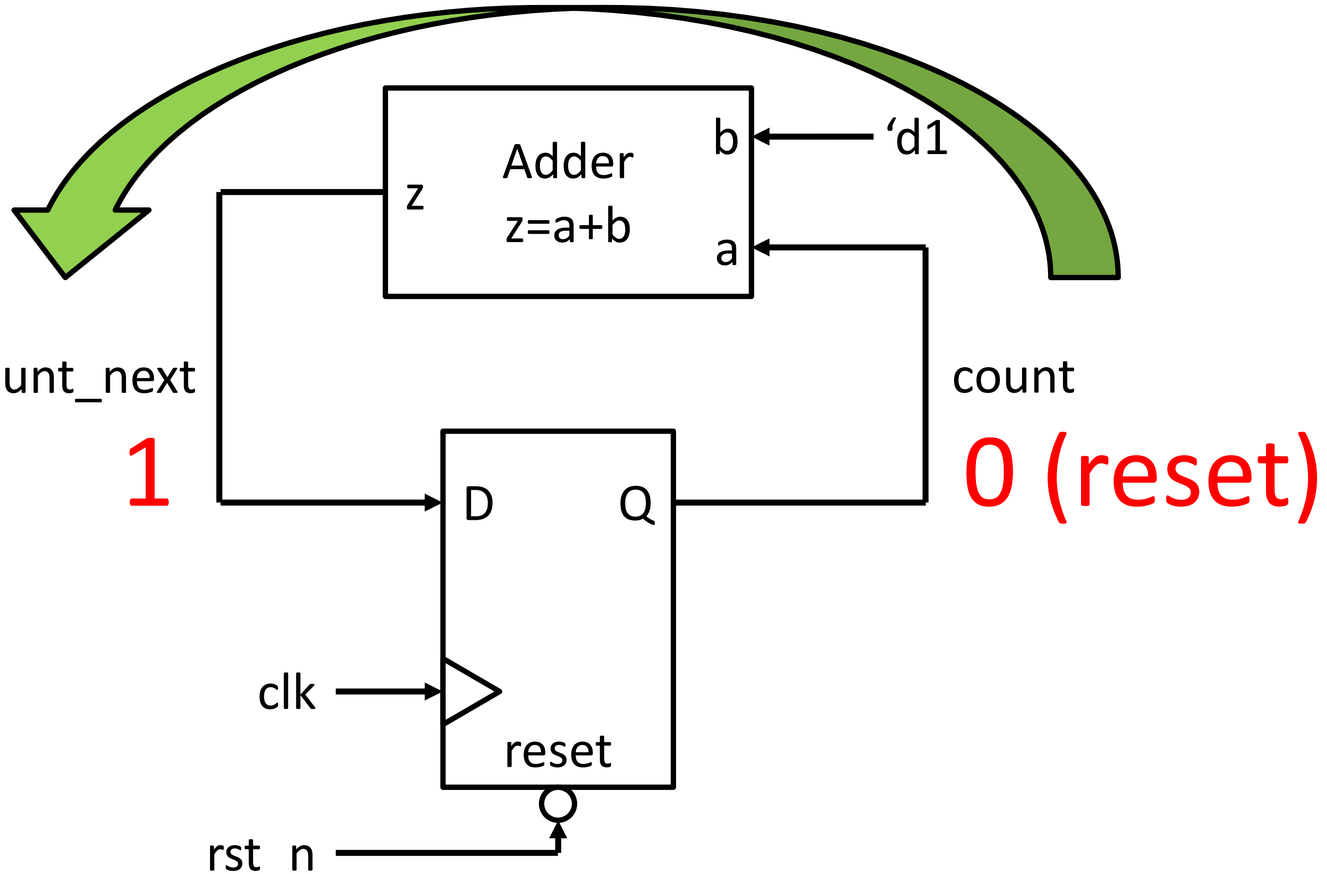


Example Counter



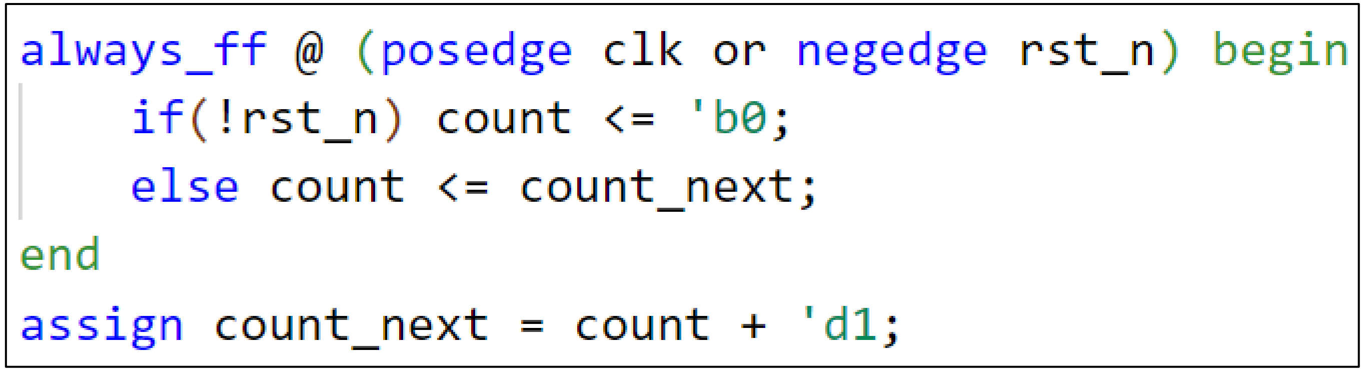
```
always_ff @ (posedge clk or negedge rst_n) begin
    if(!rst_n) count <= 'b0;
    else count <= count_next;
end
assign count_next = count + 'd1;
```

- Cycle 0:

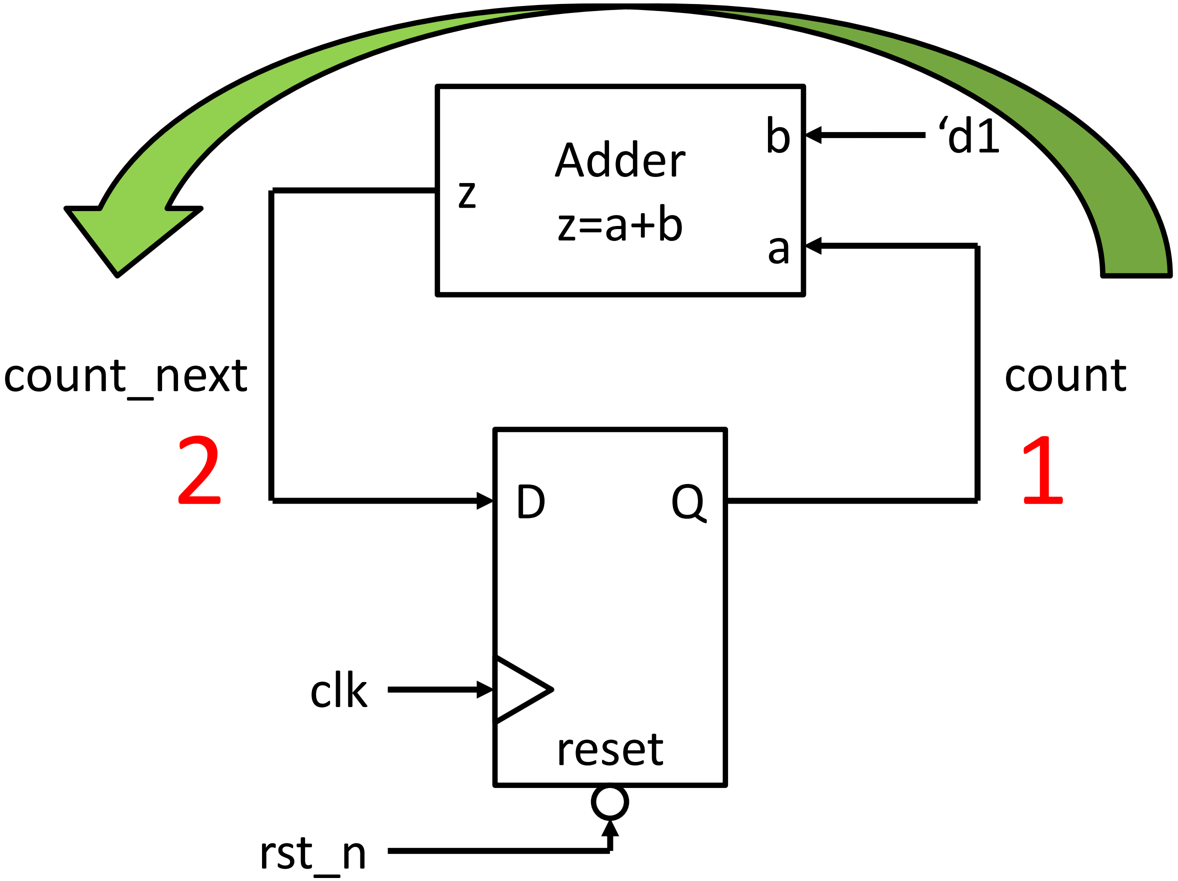


Example Counter

- Cycle 1:

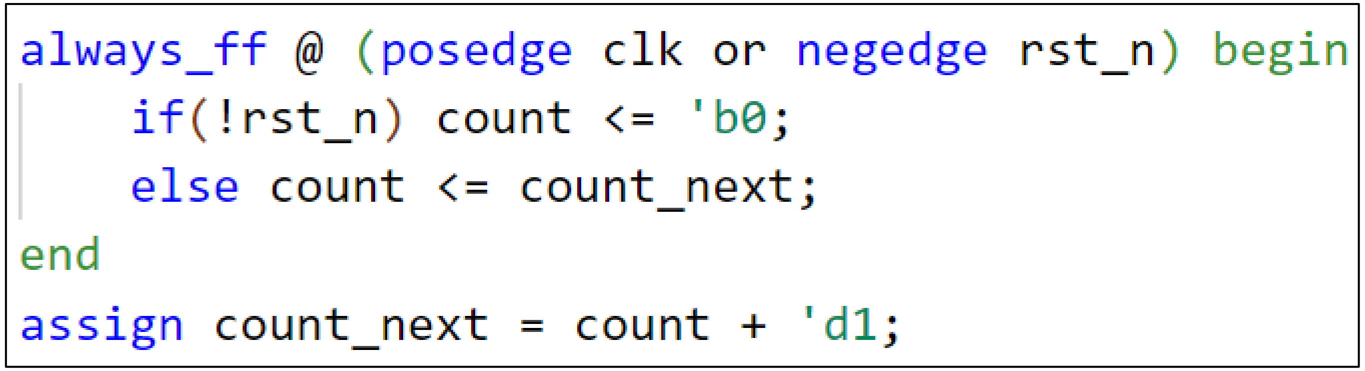


```
always_ff @ (posedge clk or negedge rst_n) begin
    if(!rst_n) count <= 'b0;
    else count <= count_next;
end
assign count_next = count + 'd1;
```

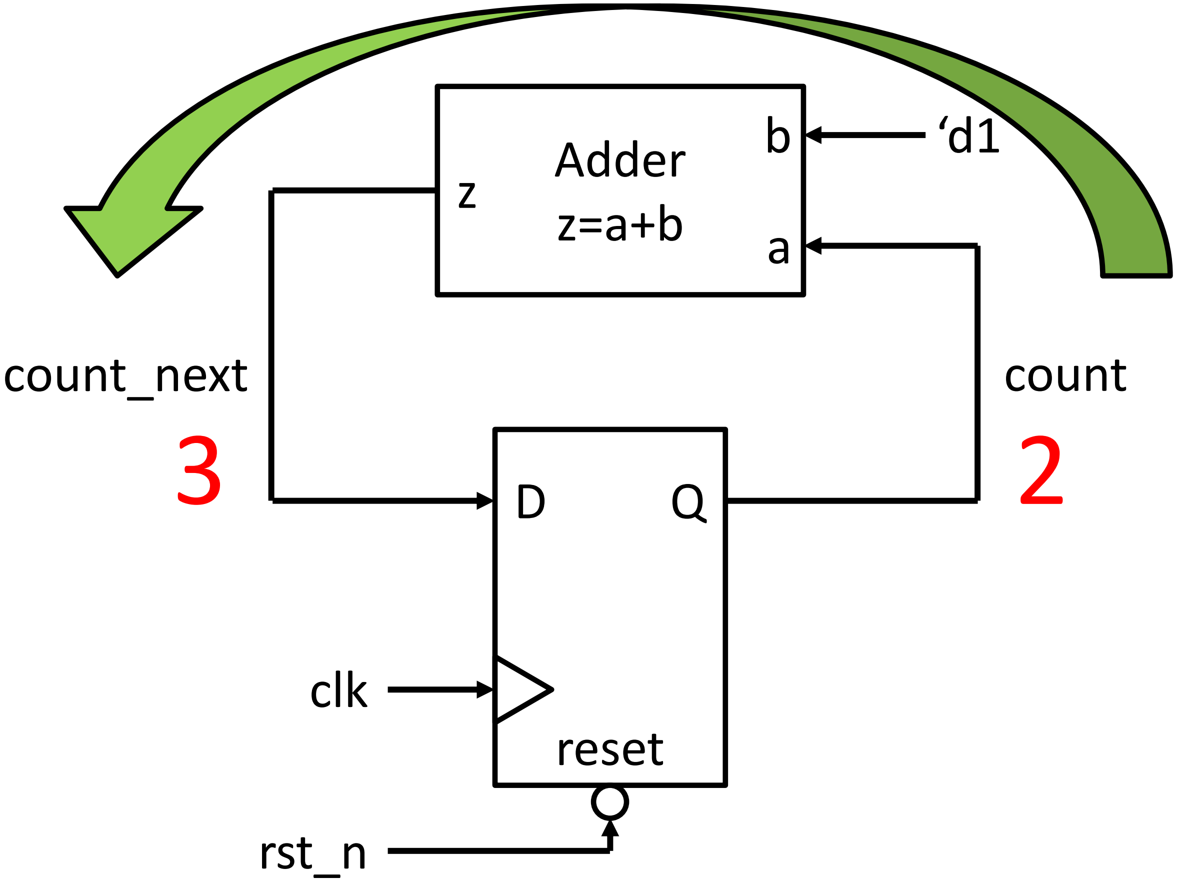


Example Counter

- Cycle 2:

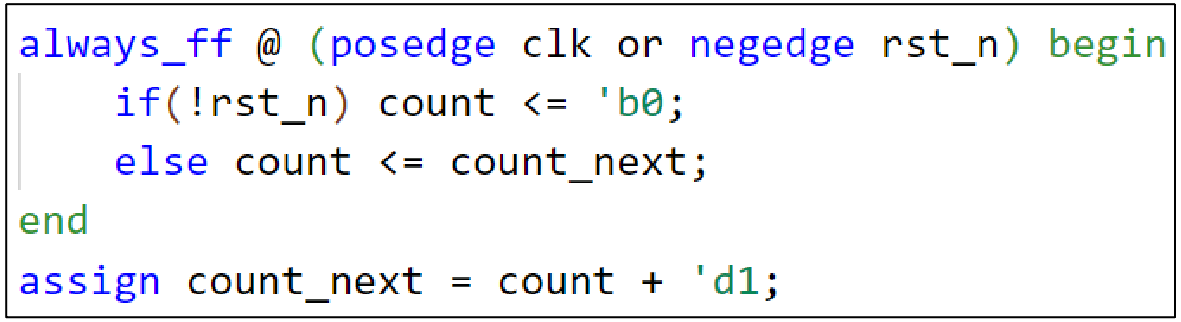


```
always_ff @ (posedge clk or negedge rst_n) begin
    if(!rst_n) count <= 'b0;
    else count <= count_next;
end
assign count_next = count + 'd1;
```

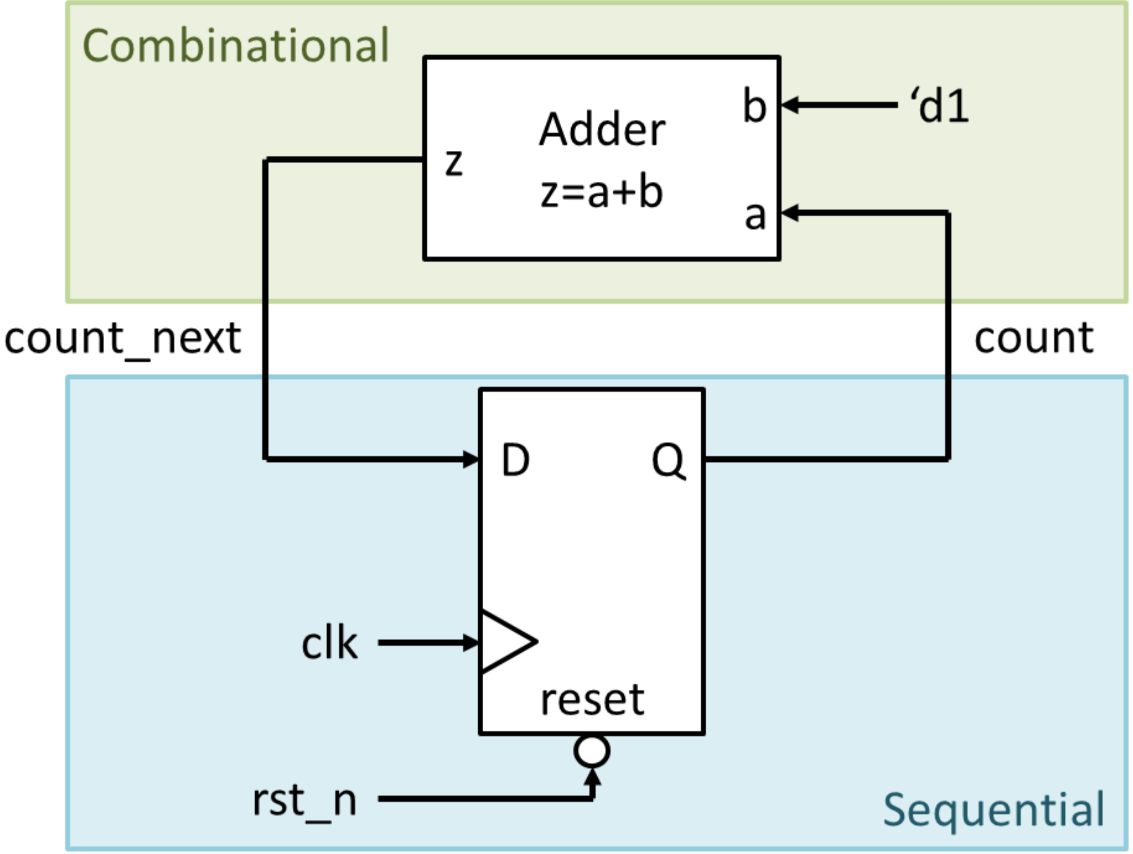


Reminder

- Reset all DFFs so you don’t start with unknown value.
- Separate combinational and sequential part in your Verilog code.
- Avoid multiple-driven net.



```
always_ff @ (posedge clk or negedge rst_n) begin
    if(!rst_n) count <= 'b0;
    else count <= count_next;
end
assign count_next = count + 'd1;
```



Problem for This Lab: Blackjack

- Card Values:
  - Number cards (1–10): Face value.
  - Face cards (Jack, Queen, King): Worth 10 points each.
- Rule:
  - “Hit" (take another card) when your hand value is no more than 16.
  - If your total exceeds 21, you bust and lose the round. Otherwise, you win!

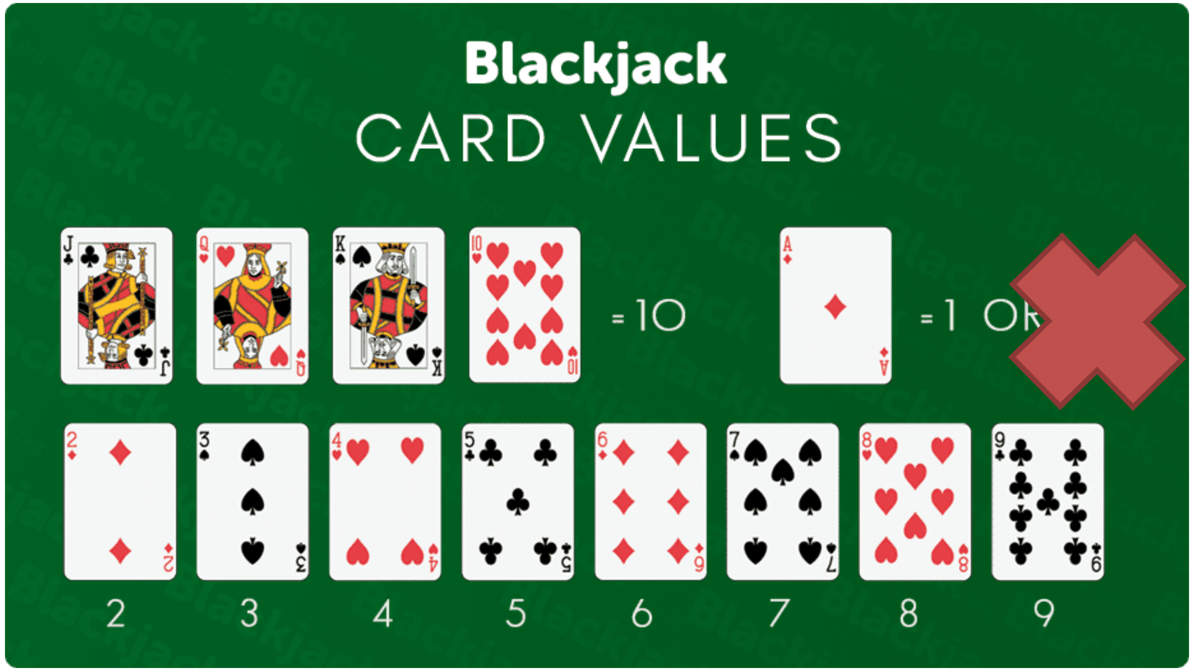


Image reference: https://www.blackjack.org/blackjack/how-to-play/

Seq.sv

Input Signal	Bit Width	Definition
clk	1	Clock.
rst_n	1	Asynchronous active-low reset.
in_valid	1	High when card is valid.
card	4	The value of a new card.

Output Signal	Bit Width	Definition
win	1	The end of a round, and the hand value is no more than 21. Should be reset when rst_n is low.
lose	1	The end of a round, and the hand value exceeds 21. Should be reset when rst_n is low.
sum	5	The hand value at the end of a round. Should be reset when rst_n is low.

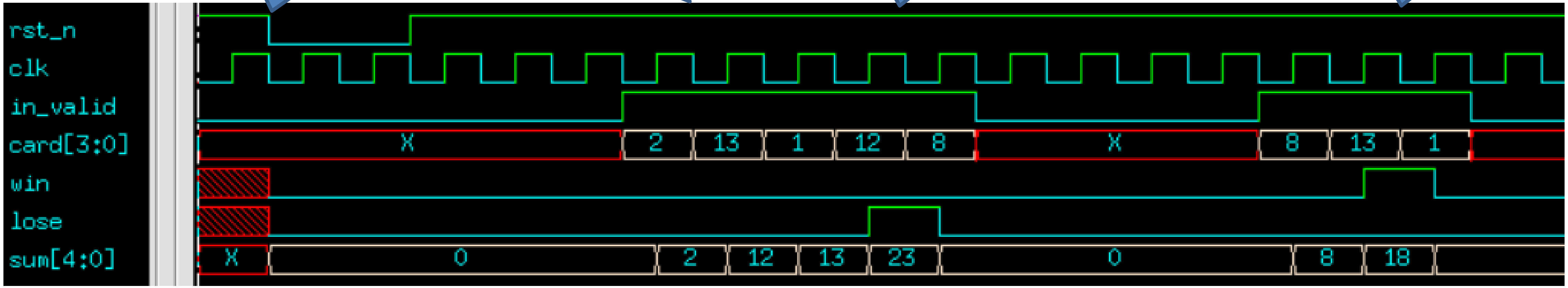
Seq.sv

Outputs should be reset when rst_n is low

Card>10 should be treated as 10

The card 8 is not taken

The card 1 is not taken



Set lose for one cycle when sum>21

Set win for one cycle when 16<sum<=21

PATTERN will only check sum when win or lose is high

Reminder

1. Output signals should be reset to zero when rst_n is low.
2. You should keep drawing card (don’t set win or lose to 1) when your hand value is no more than 16, and stop drawing card when your hand value is more than 16.
3. win and lose should not be set at the same time.
4. win should not be set if your hand value is no more than 21.
5. lose should not be set if your hand value is more than 21.
6. sum should be correct when win or lose is high.
7. win or lose should be set for only one cycle.
8. The cycle time is 10 ns.

Grading Policy

- Pass the RTL & Synthesis & Gate-level simulation: 100%
  - 合成結果: **(不能有Error、Timing report slack met、不能有Latch)**
- Demo 2 打7折

Upload

- 請將Lab03/01_RTL裡的Seq.sv依以下命名規則重新命名後上傳至E3
- 命名規則：Seq_dcsxxx.sv，xxx為工作站帳號號碼
- **命名錯誤扣5分!!!**
- **Deadline:**
  - **Demo 1: 3/13 17:25**
  - **Demo 2: 3/13 23:59**