# SC1005
# Digital Logic

## Recap and Discussion

## Lecture 17 and 18

## Sequential Circuits

# Summary of Lecture 17

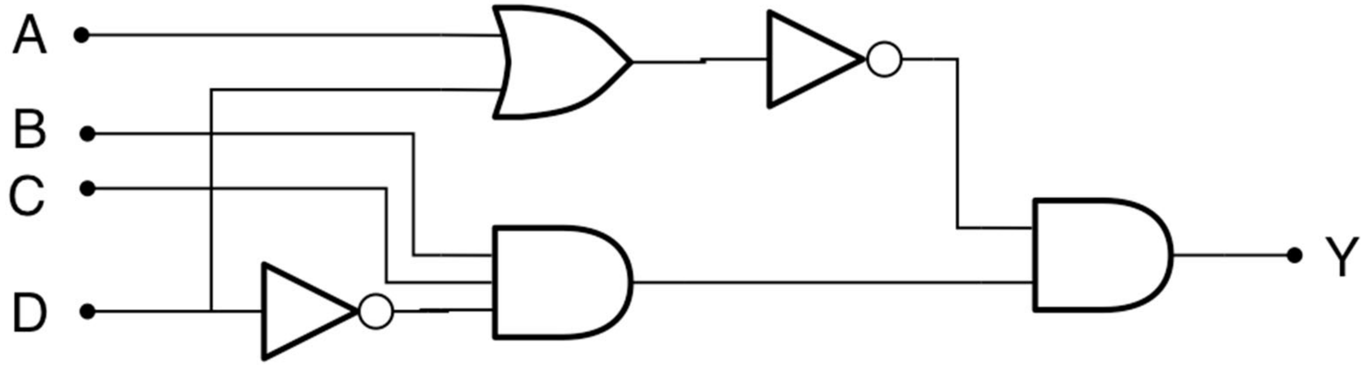
- Intro to Sequential Circuits
- Latches
- Timing

# Summary of Lecture 18

- Sequential Circuits
    - Gated/Enabled SR Latch
    - Transparent D-Latch
    - Edge-Triggered Flip-Flop
    - Registers

# Review of Combinational Circuits

- So far, we have investigated ***combinational*** circuits
- The output of a combinational circuit is purely a function of the inputs to it
  - *output = f (input)*
- When the inputs are changed, the output will change to that determined by those inputs (after a short propagation delay)
- Knowing only the ***current*** inputs, we can always determine the output



# Sequential Circuits

- ***Sequential*** circuits introduce the idea of ***state***
  - The state of a circuit encodes information about the past to determine how it will react with the inputs to produce an output

- The output of a sequential circuit depends on **both** the ***current*** inputs and what has previously happened (the ***previous*** inputs determine the “state”)

- In this sense, sequential circuits have ***memory***, whereas combinational circuits have no memory.

# Exercise 2

- Consider two styles of fan speed control:
  - Dial-based (Off, Lo, Med, Hi)
  - Button-based (+,–)
- Which of the following correctly define the type of circuit used for implementing the fan control for L and R?

L

R

A. L – Combinational
R – Combinational

B. L – Combinational
R – Sequential

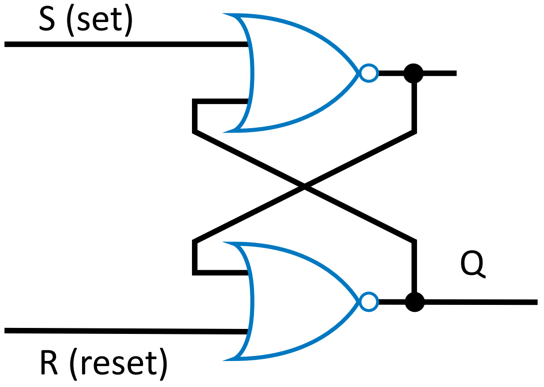
C. L – Sequential
R – Combinational

D. L – Sequential
R – Sequential

**Ans: B.** For the first, the fan speed depends on which input is selected. For the second, the fan speed depends on the input and the current fan speed

# Set-Reset (SR) Latch

- The most basic circuit for storing a bit is the **Set-Reset (SR) Latch**

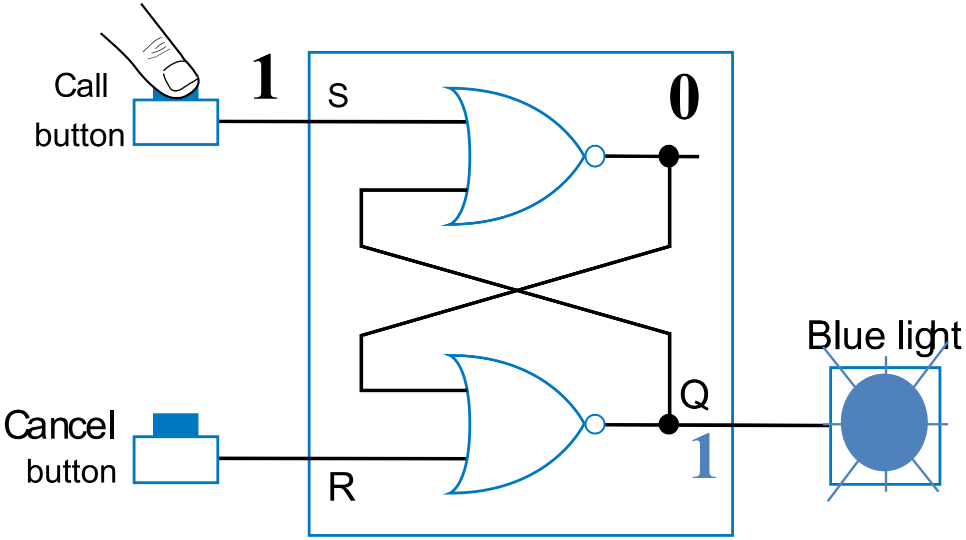


| S | R | Q+ | Function |
|---|---|---|---|
| 0 | 0 | Q | Store |
| 0 | 1 | 0 | Reset |
| 1 | 0 | 1 | Set |

Note: S = R =1 is not defined

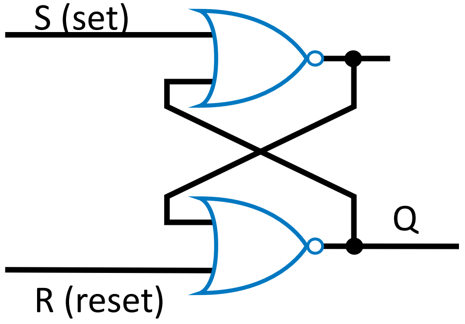
The SR latch allows us to build a flight attendant call system

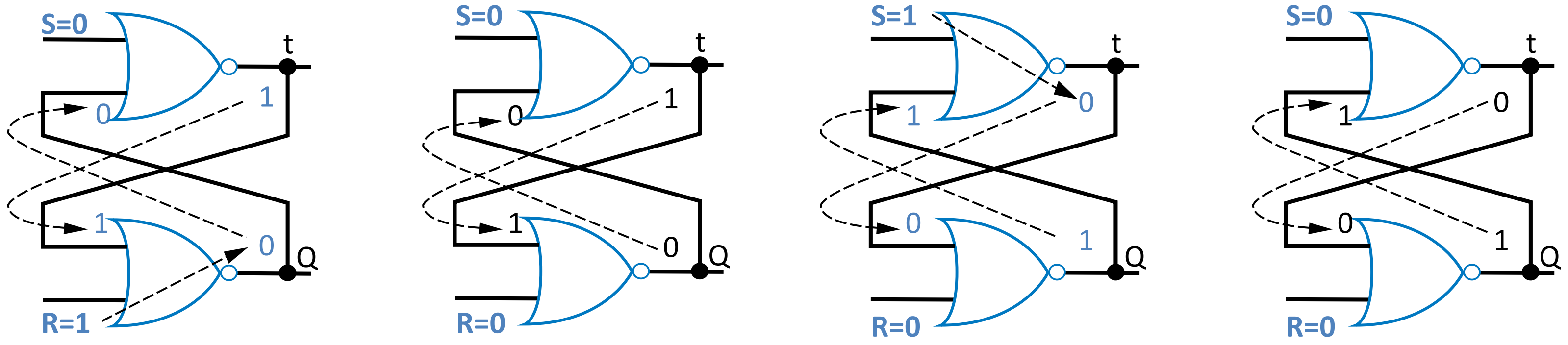
- Connect “call” to S, and “cancel” to R
- Q stays at 1 even after “call” is released

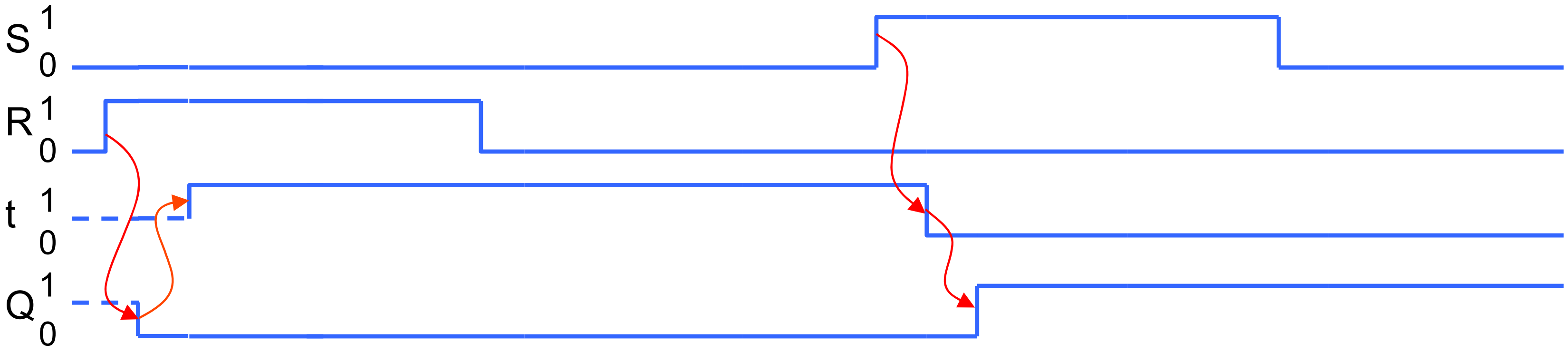


# Set-Reset (SR) Latch

- Operation:

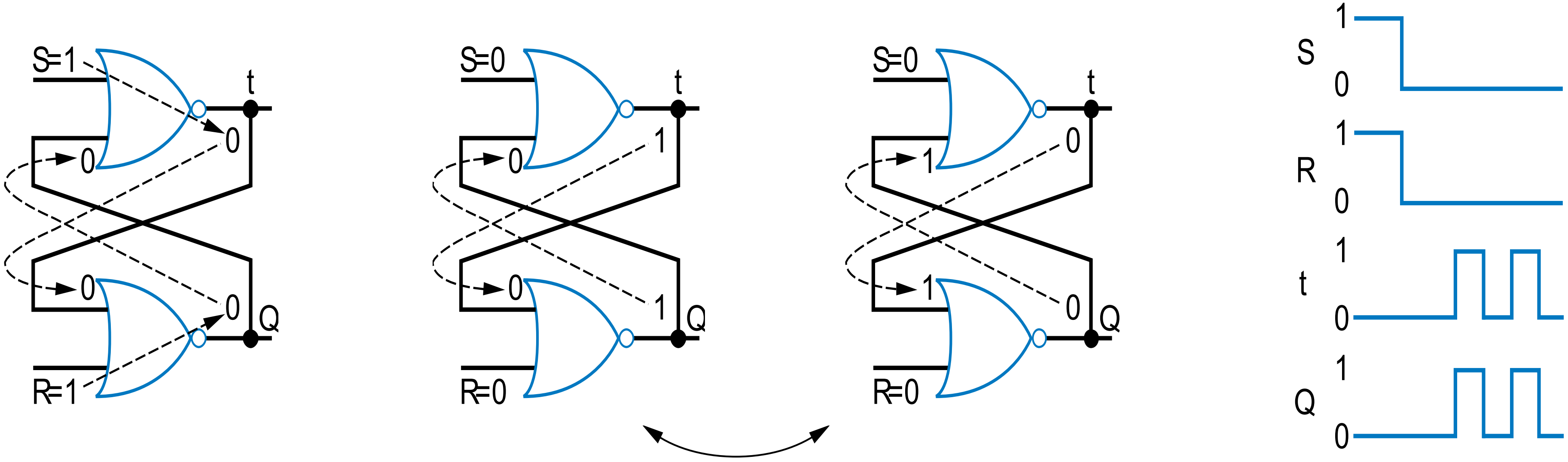






# Set-Reset (SR) Latch

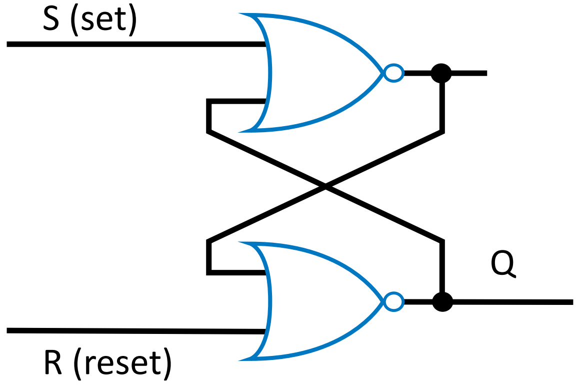
- There is, though, a problem with the SR Latch
- What happens when both S and R are asserted?



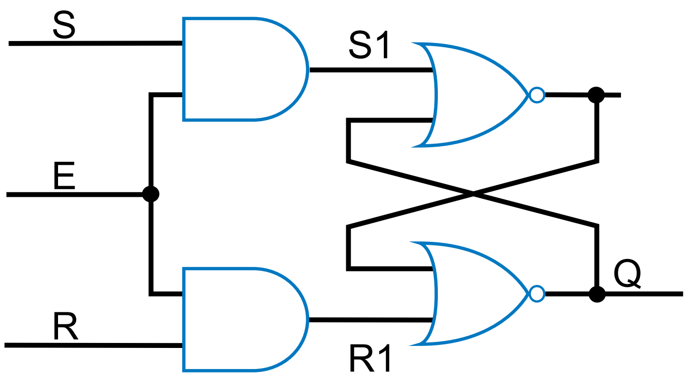
- When both S and R is then de-asserted at the same time, we may get oscillation at the output:
  - Constant swapping from 0 to 1 to 0 …
- Normally, one gate is a little faster so the output will eventually stabilize, but which?
  - Impossible to know

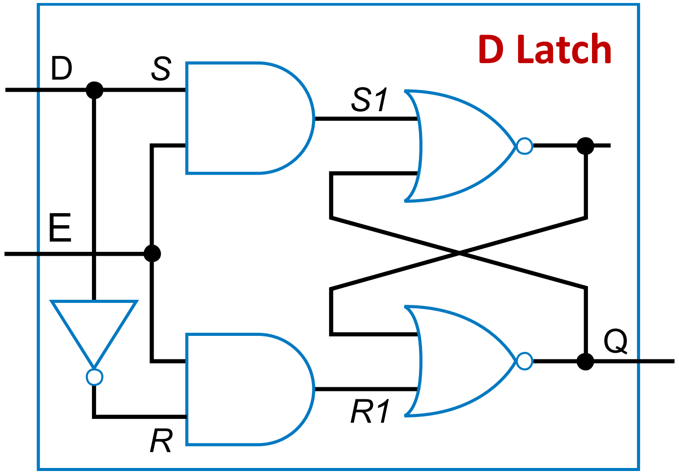
# Recap: Latches

## Set-Reset (SR) Latch



## Enabled SR Latch





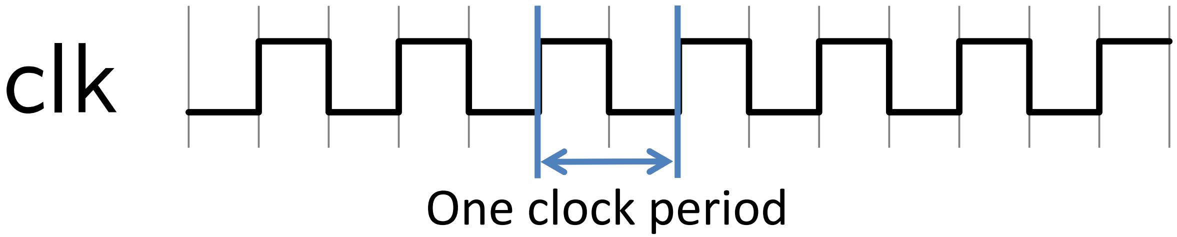
| S | R | Q+ | Function |
|---|---|---|---|
| 0 | 0 | Q | Store |
| 0 | 1 | 0 | Reset |
| 1 | 0 | 1 | Set |
| 1 | 1 | ? | Undefined |

| E | S | R | Q+ | Function |
|---|---|---|---|---|
| 0 | X | X | Q | Store |
| 1 | 0 | 0 | Q | Store |
| 1 | 0 | 1 | 0 | Reset |
| 1 | 1 | 0 | 1 | Set |
| 1 | 1 | 1 | ? | Undefined |

| E | D | Q+ | Function |
|---|---|---|---|
| 0 | X | Q | Store |
| 1 | 0 | 0 | Transparent |
| 1 | 1 | 1 | Transparent |

# The Clock

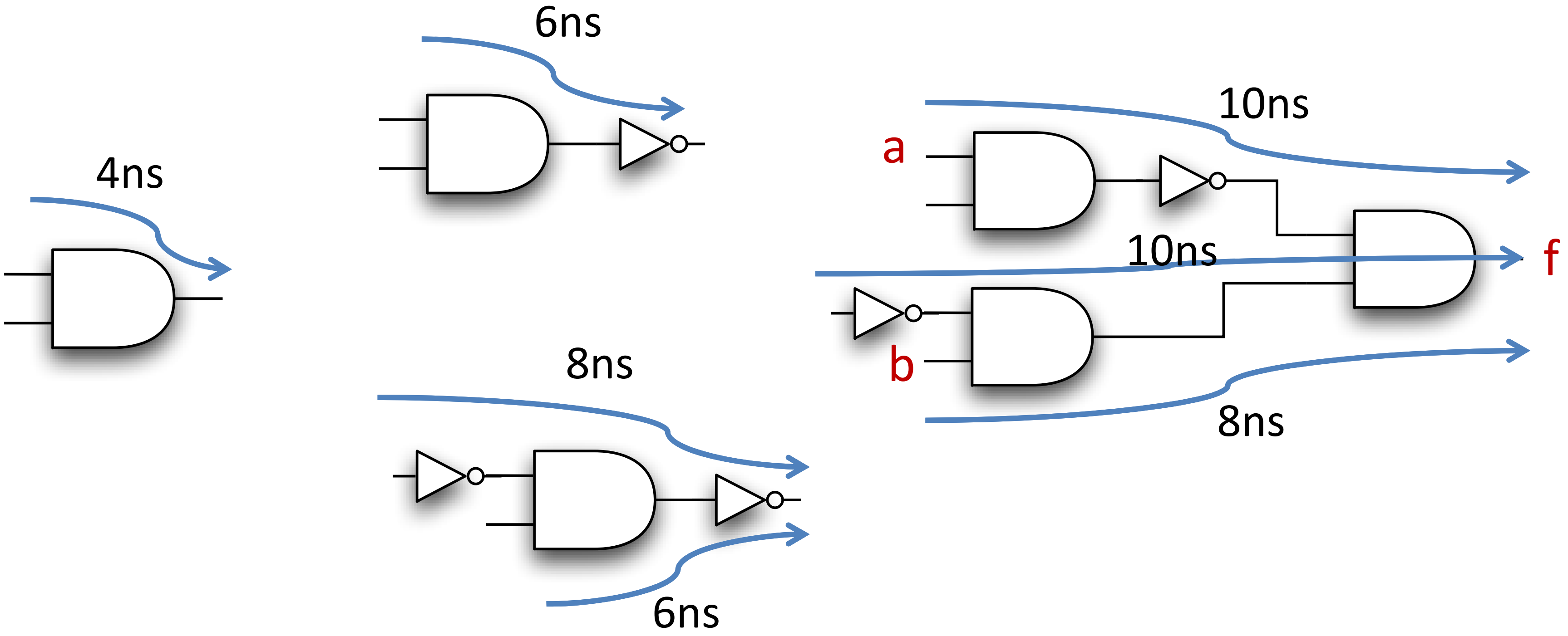
- In digital systems, sequential components are connected to what we call a "clock"
- The clock is a signal that continuously toggles between 0 and 1 at a fixed rate:



- The period of the clock is the time it takes to complete one complete cycle – we call this a cycle
- The frequency is the inverse of the period:
  - A 10ns ($10 \times 10^{-9}$s) period gives a frequency of $1/10 \times 10^{-9} = 10^8$Hz = 100MHz
  - This is where the MHz and GHz numbers for chips come from – how fast the clock oscillates

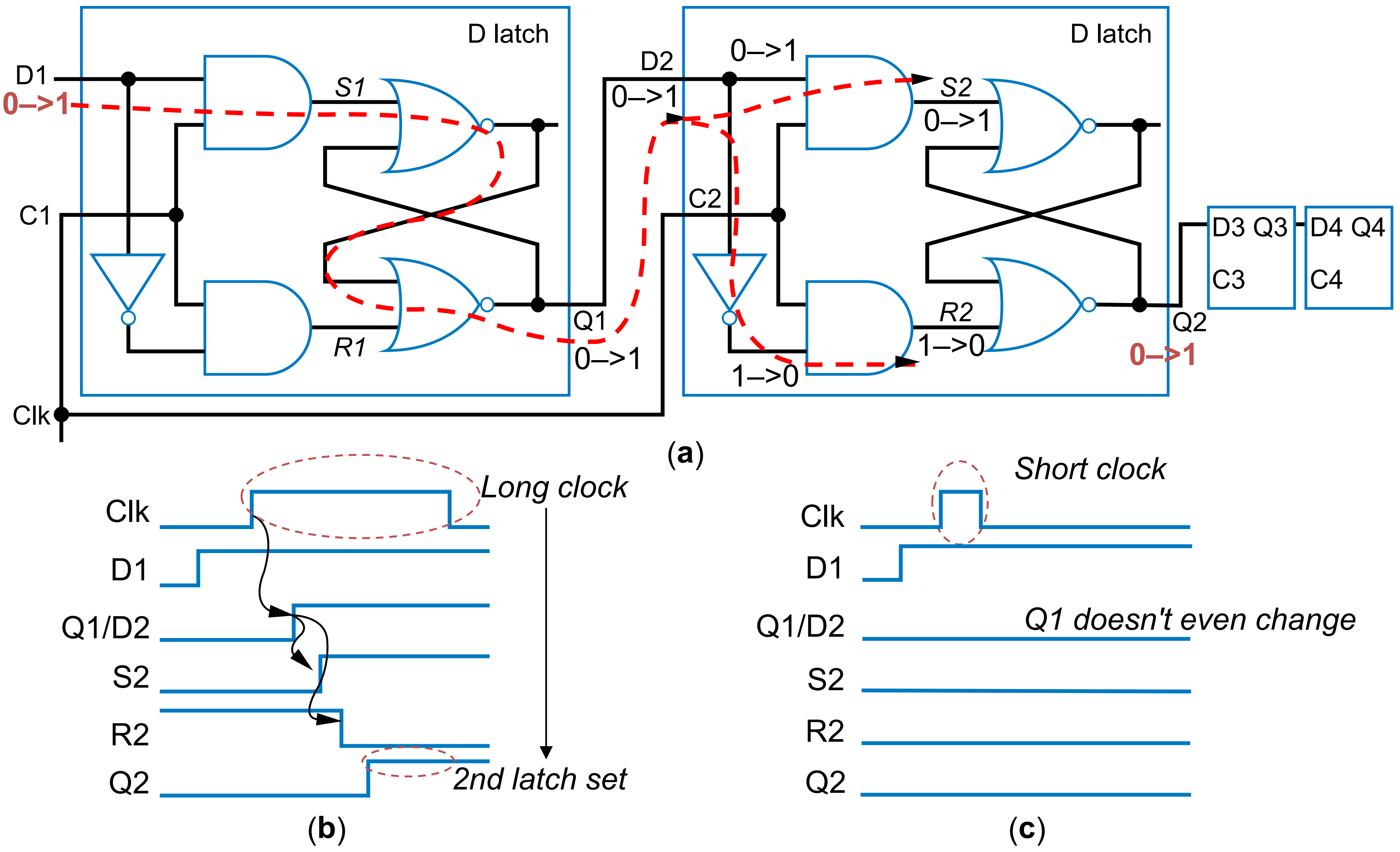
# Timing in Circuits

- Assume an **and** gate has a delay of 4ns and an **inverter** has a delay of 2ns:



- The propagation delay, $t_p$, may differ between different inputs and outputs in the same circuit
- If we report a single number, we always use the worst (i.e. longest path)

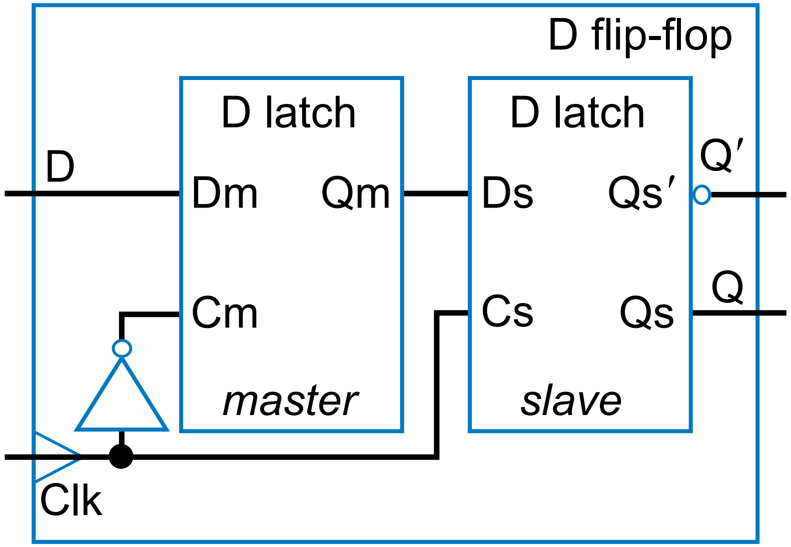
# Timing of Latches



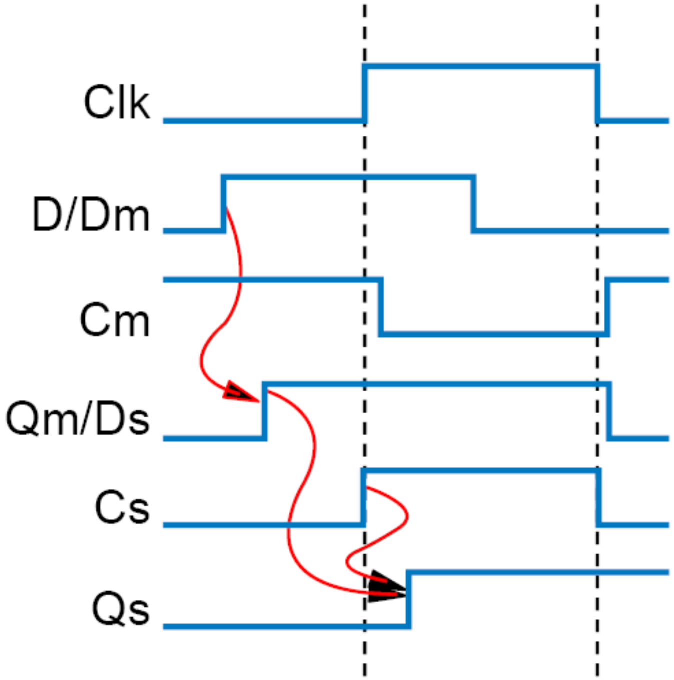
Transparent (level-sensitive) latches are problematic!!

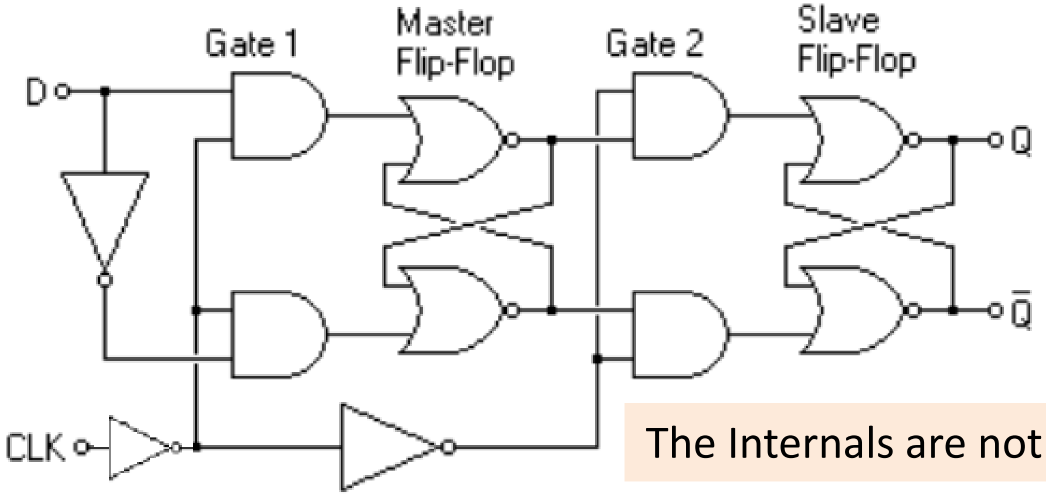
# Recap: Edge-Triggered Flip-Flops

- Use an edge triggered flip-flop to overcome the limitation of D latches
- We build edge-triggered flip-flops using two D-latches in a master-slave arrangement.



- A **flip-flop** is a circuit that changes its outputs only at the **control signal's edges**
  - **Positive edge-triggered**: output changes when control goes from 0 to 1
  - **Negative edge-triggered**: output changes when control goes from 1 to 0

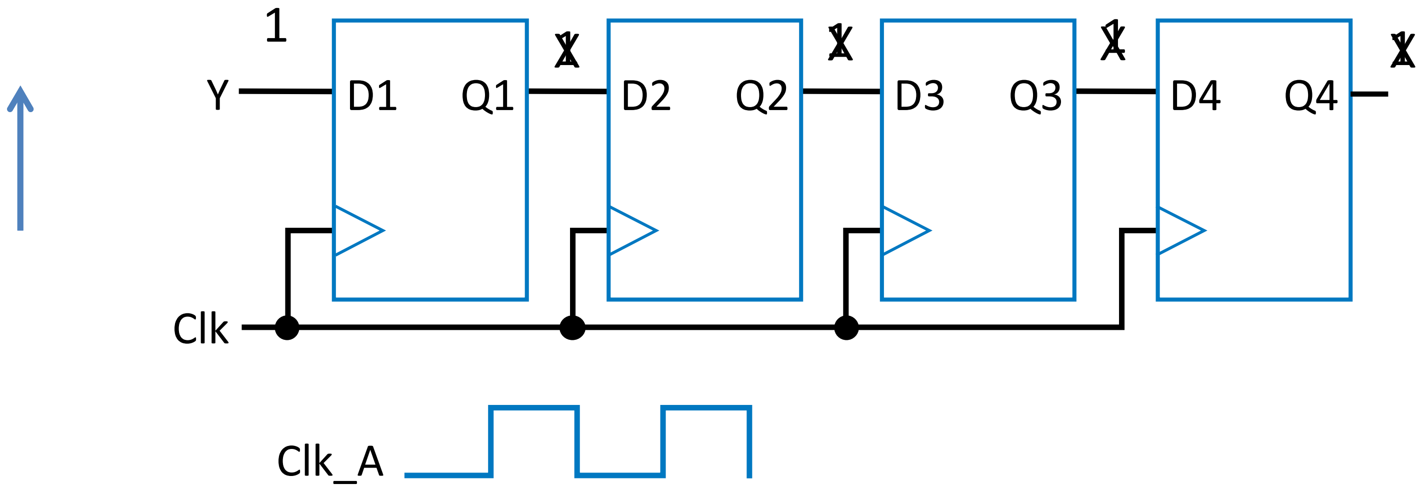




The Internals are not important

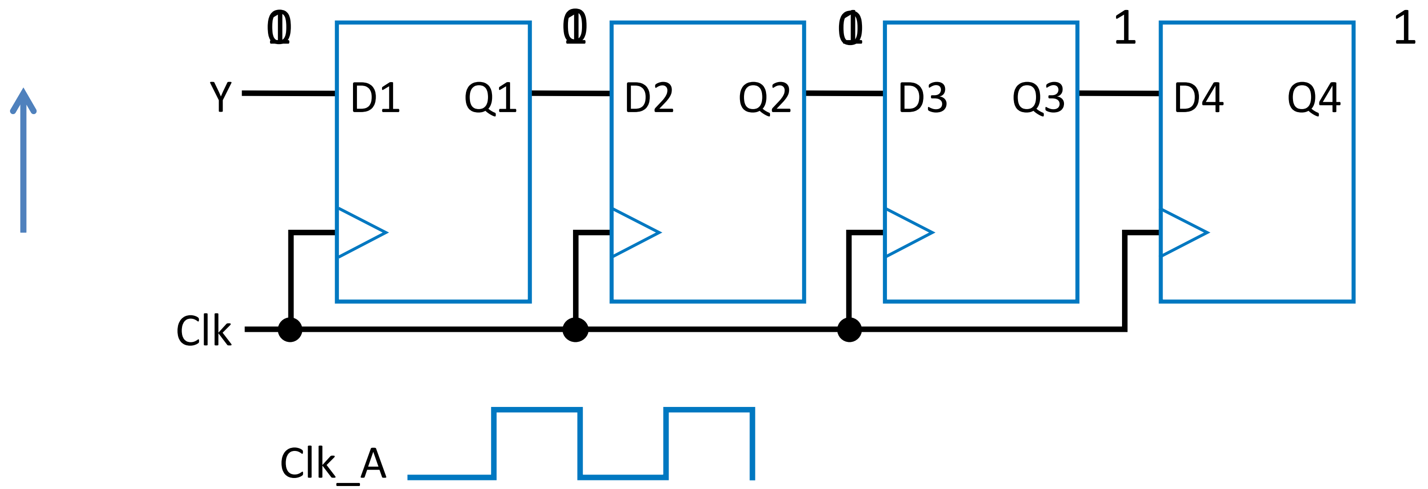
# Recap: Edge Triggered Flip-Flops

- Use edge triggered flip-flops to overcome the limitation of D latches
  - Apply a 1 to the input. The current state of the FFs is unknown (X)



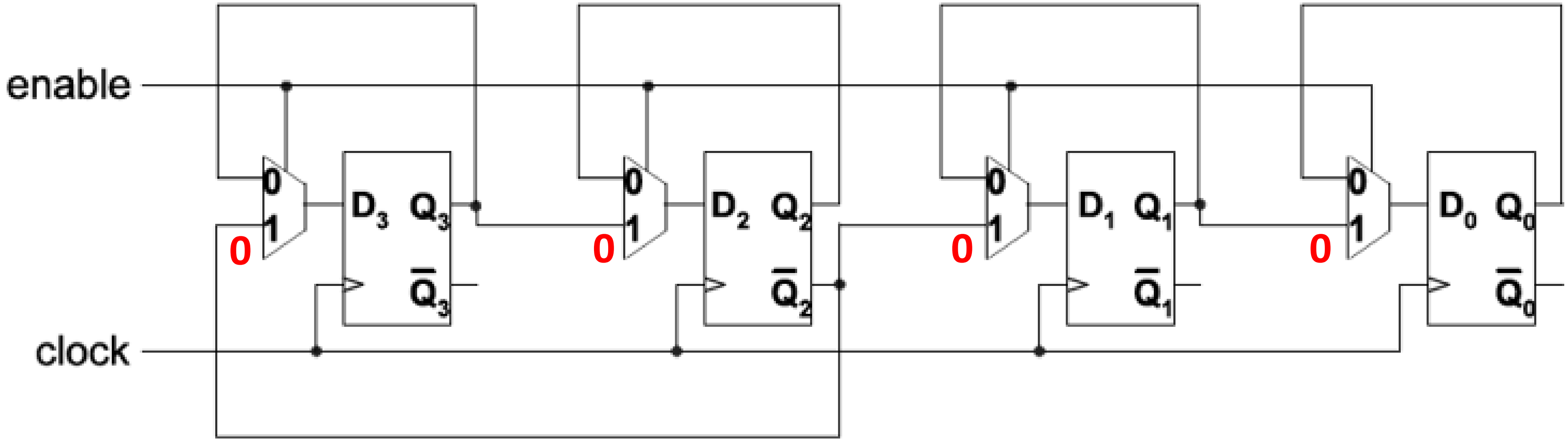
# Recap: Edge Triggered Flip-Flops

- Use edge triggered flip-flops to overcome the limitation of D latches



The duty cycle of the clock has no impact on the FF operation.

# Exercise 3



If *enable* = 1 and the current state of the circuit is $Q_3Q_2Q_1Q_0$ = 0101, what is the next state?

A. 1111

B. 1010

C. 1100

D. 0011

E. 0000

Ans: E = 0000

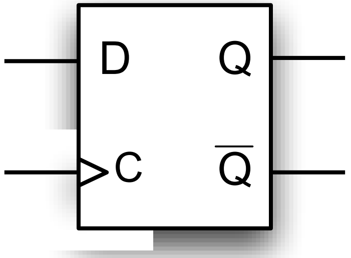
$Q_2$ is '1', so $Q_2$' = '0'. $Q_3$ and $Q_1$ will be '0' at the next rising edge. The other outputs will just propagate through.

What is the next state if enable=0 and $Q_3Q_2Q_1Q_0$ = 0101?

Ans: $Q_3Q_2Q_1Q_0$ = 0101 (No change)

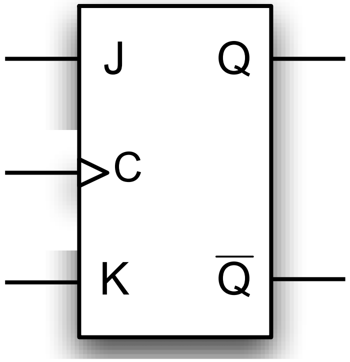
# Edge-Triggered Flip-Flops

- We've seen the D flip-flop:
  - Q takes the value on D at the rising edge of C



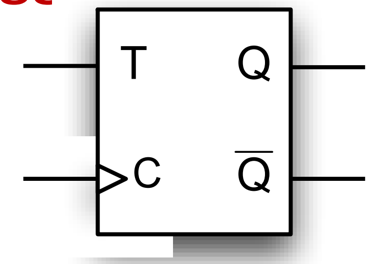
| clk | D | Q+ |
|---|---|---|
| ↑ | 0 | 0 |
| ↑ | 1 | 1 |
|  | X | Q |

- There's also a J-K flip-flop:
  - At the rising edge, Q becomes 1 if J is asserted and 0 if K is asserted
  - If both J and K are asserted, the output toggles at the rising edge



| clk | J | K | Q+ |
|---|---|---|---|
| ↑ | 0 | 0 | Q |
| ↑ | 0 | 1 | 0 |
| ↑ | 1 | 0 | 1 |
| ↑ | 1 | 1 | Q’ |
|  | X | X | Q |

- And a T flip-flop:
  - Q toggles at the rising edge if T=1

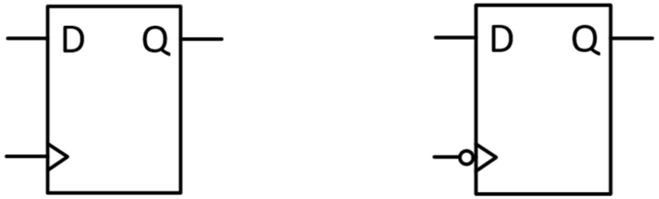


| clk | T | Q+ |
|---|---|---|
| ↑ | 0 | Q |
| ↑ | 1 | Q’ |
|  | X | Q |

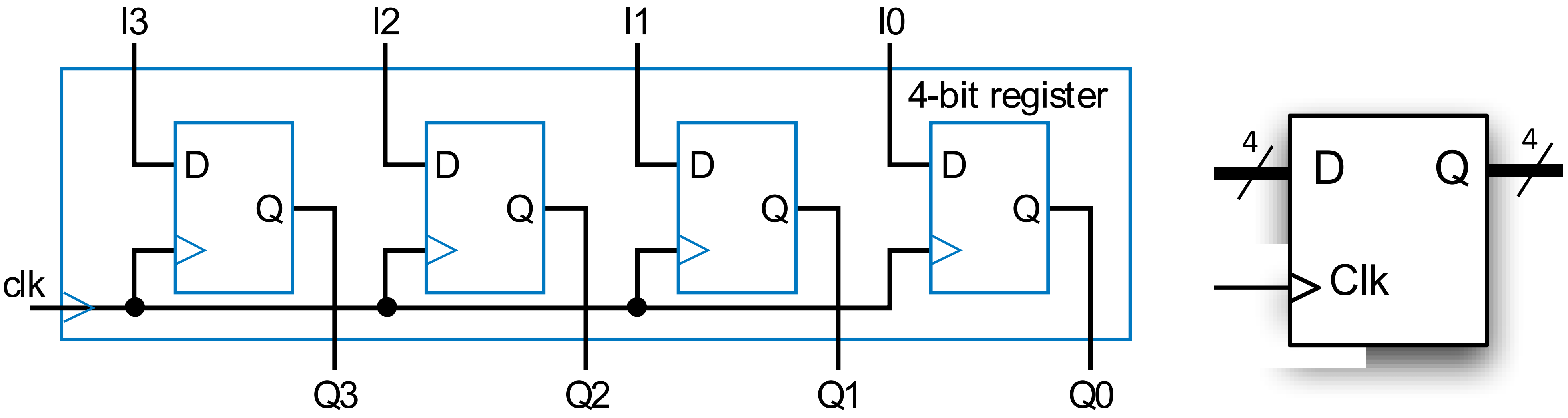
- (Also a lesser seen SR flip-flop)

# Registers

- A **flip-flop** is a circuit that changes its outputs only at the **control signal's edges**
  - **Positive edge-triggered**
  - **Negative edge-triggered**

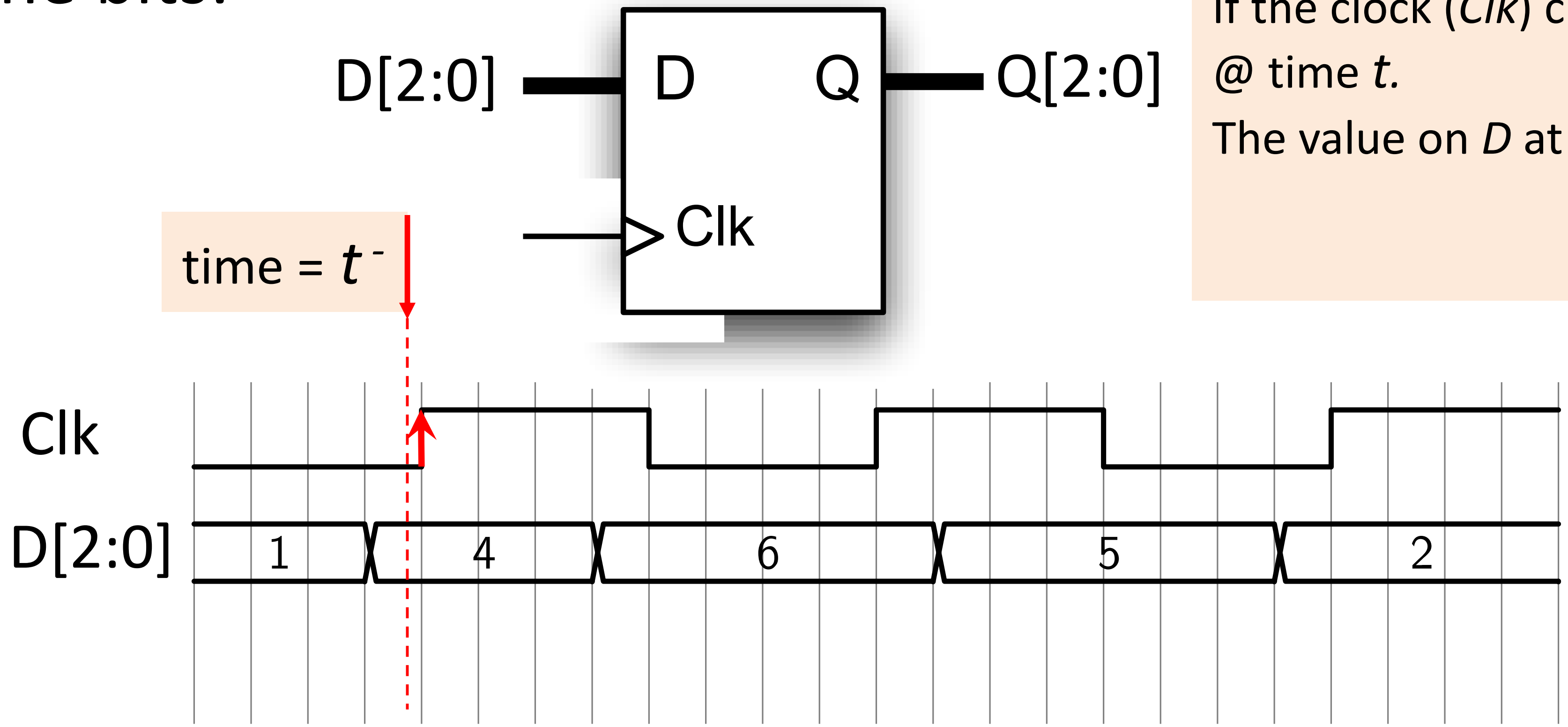


- The D-type flip-flop is the fundamental building block in synchronous design
- But we often deal with multi-bit signals
- When we combine multiple D flip-flops together to store multiple bits, we call this a **register**:



# Registers

- When dealing with multi-bit registers, we can show these values in a timing diagram without splitting up the bits:

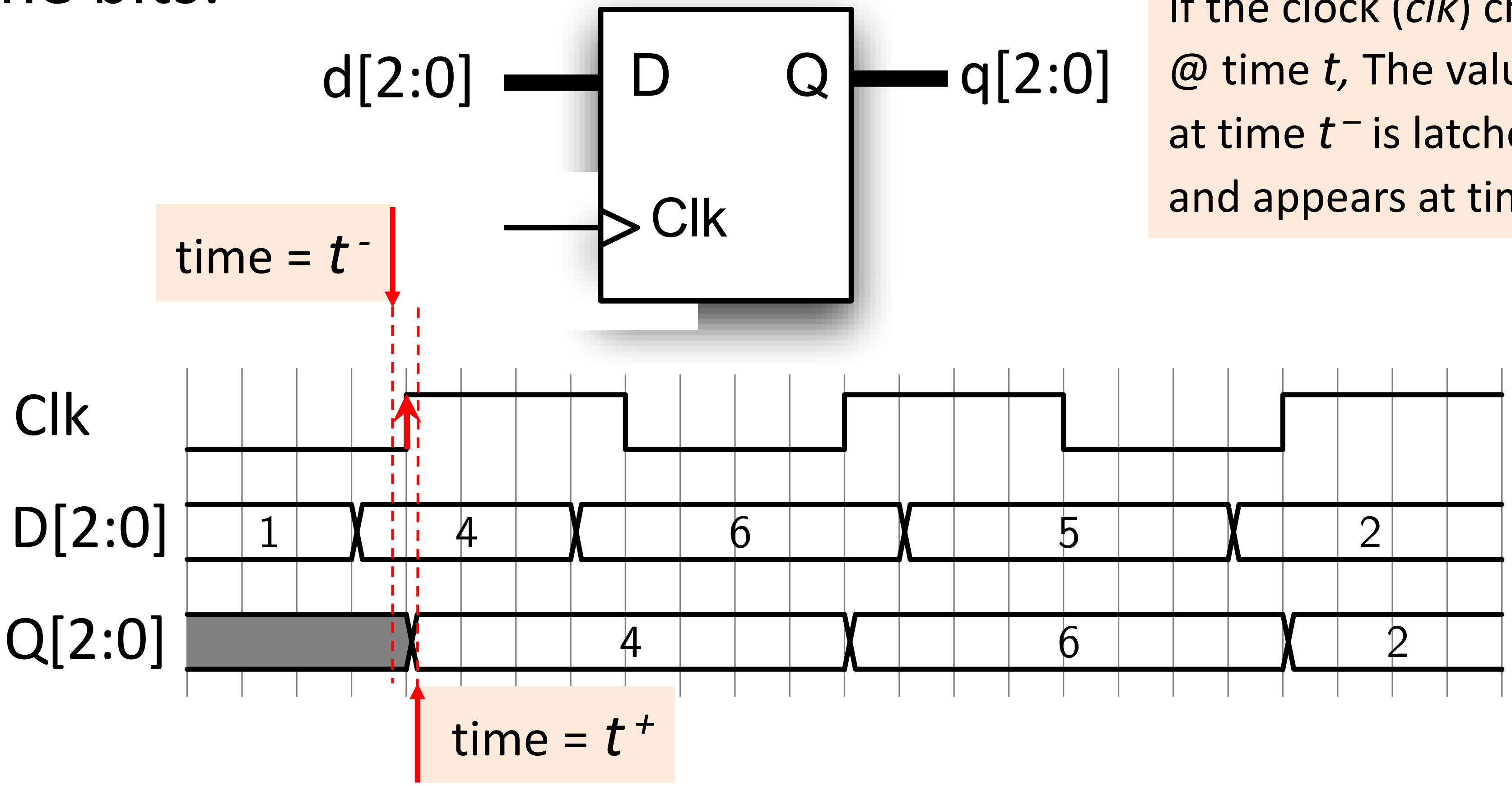


If the clock (*Clk*) changes @ time *t.*

The value on *D* at time $t^-$

# Registers

- When dealing with multi-bit registers, we can show these values in a timing diagram without splitting up the bits:

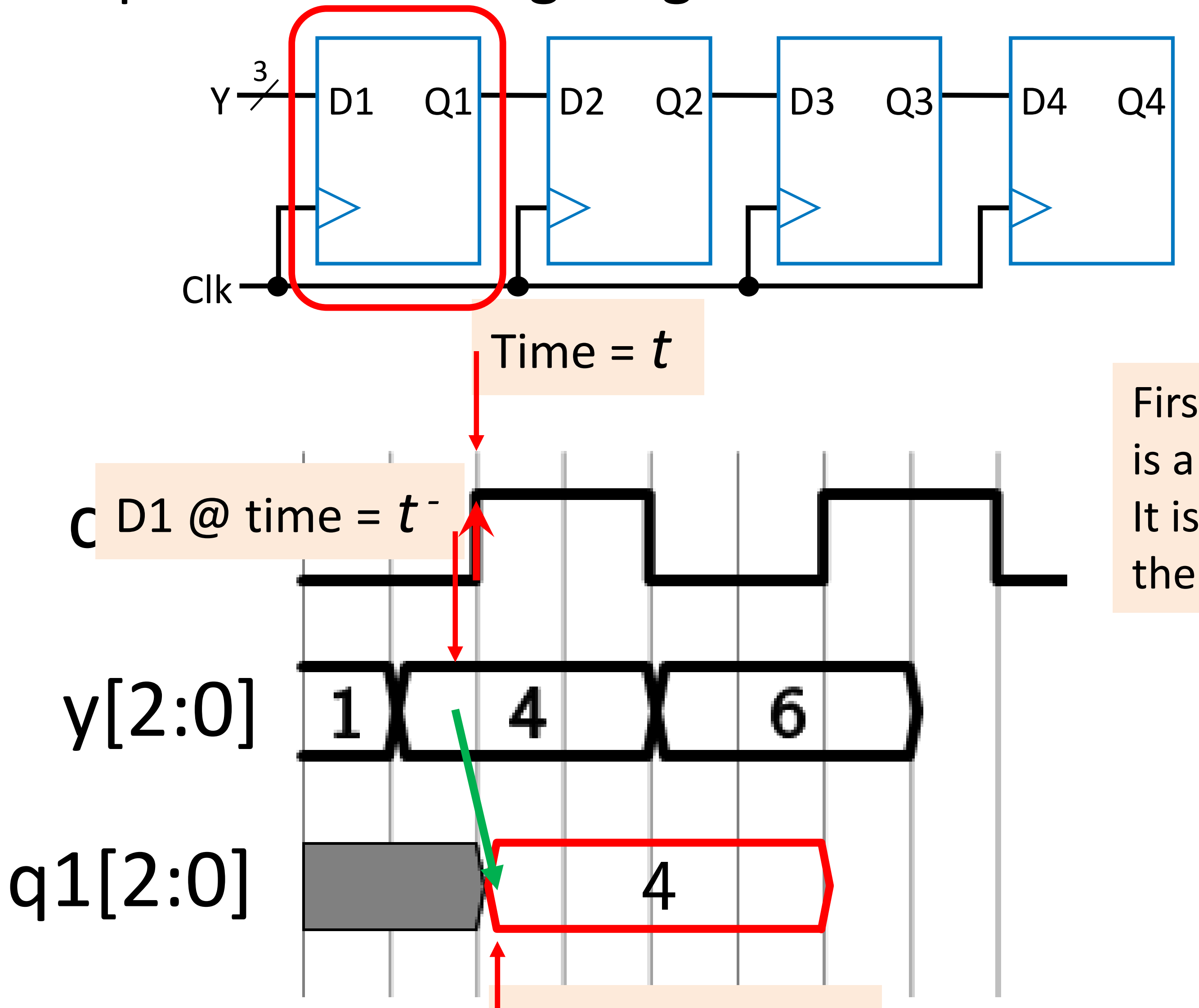


If the clock (*clk*) changes @ time *t*, The value on *D* at time $t^-$ is latched to Q and appears at time $t^+$

- We can use decimal or hex (or even binary), but make it clear
- Again, transitions only happen at the rising edge

# Lecture 18 (Task 1)

- Complete the timing diagram of the shift register



Time = $t$

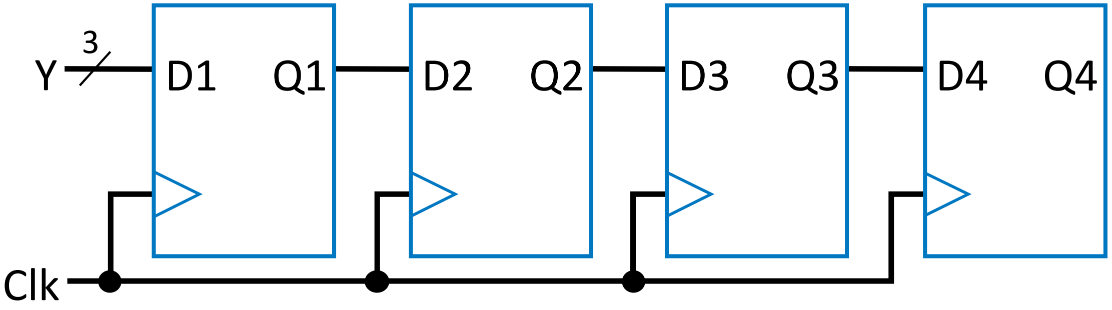
D1 @ time = $t^-$

Q1 @ time = $t^+$

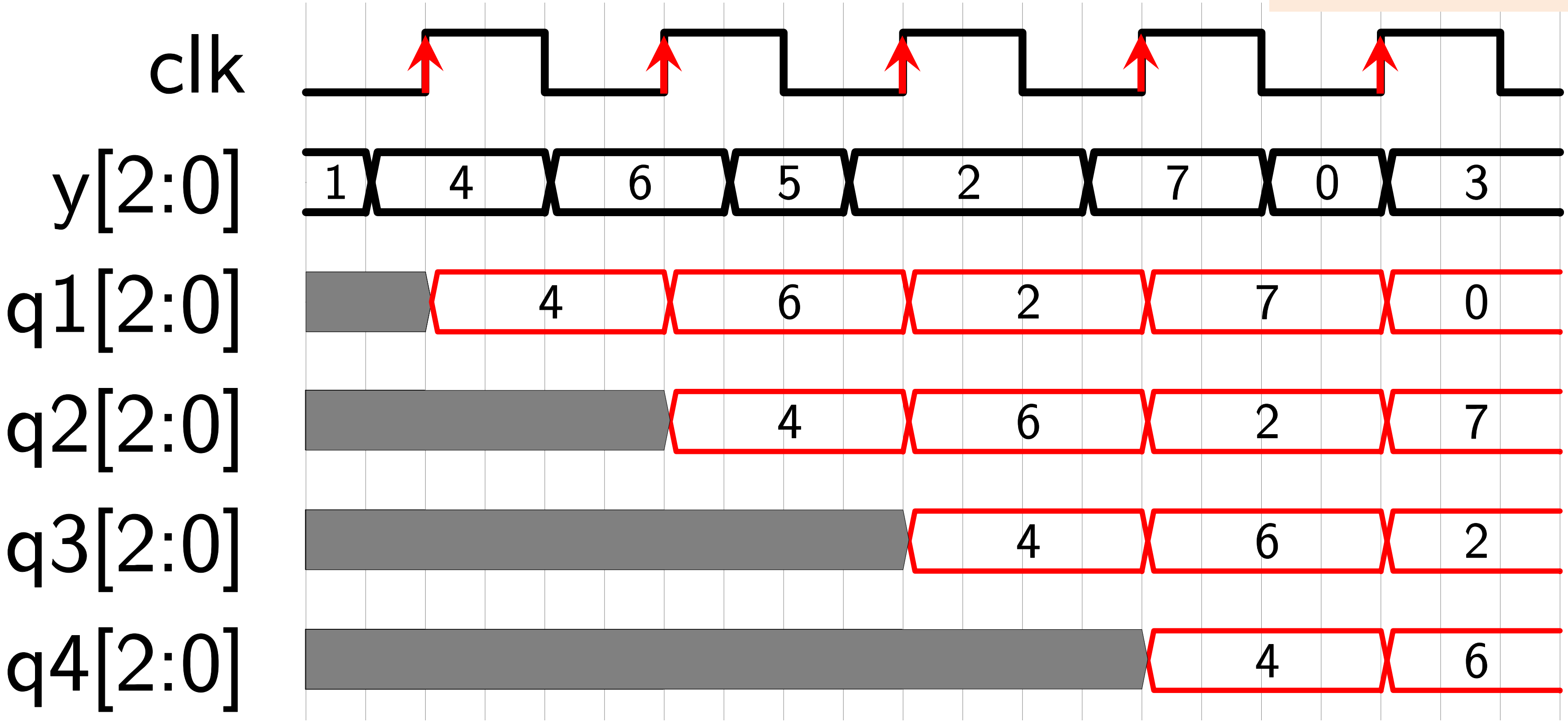
Firstly, in any circuit there is a propagation delay.
It is this delay that makes the FF work.

# Lecture 18 (Task 1)

- Complete the timing diagram of the shift register



Note all four 3-bit registers produce an output at the same time.
Just after each clk rising edge.



# Selected Past Exam Questions

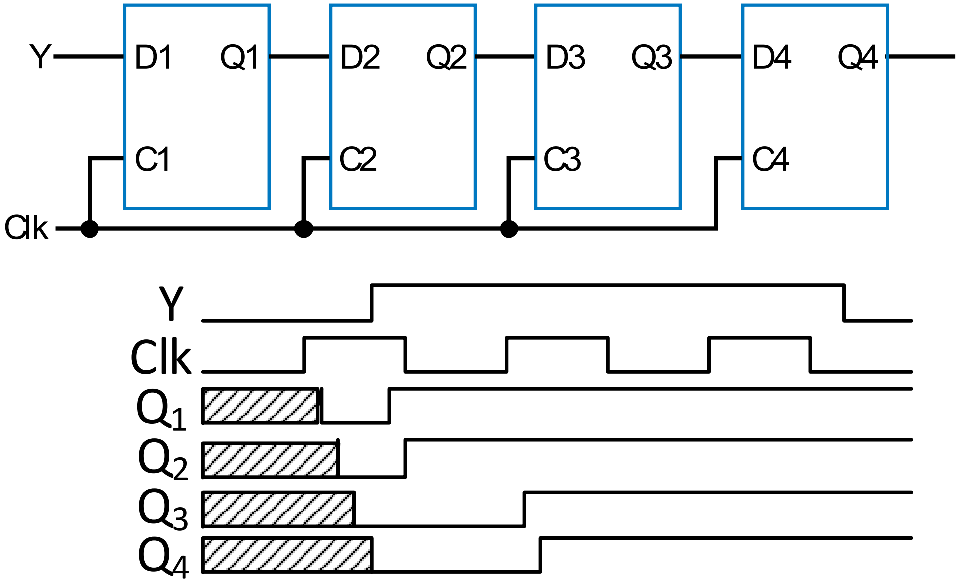
## CE/CZ1005 2017-2018 Semester 2 (Apr/May 2018)

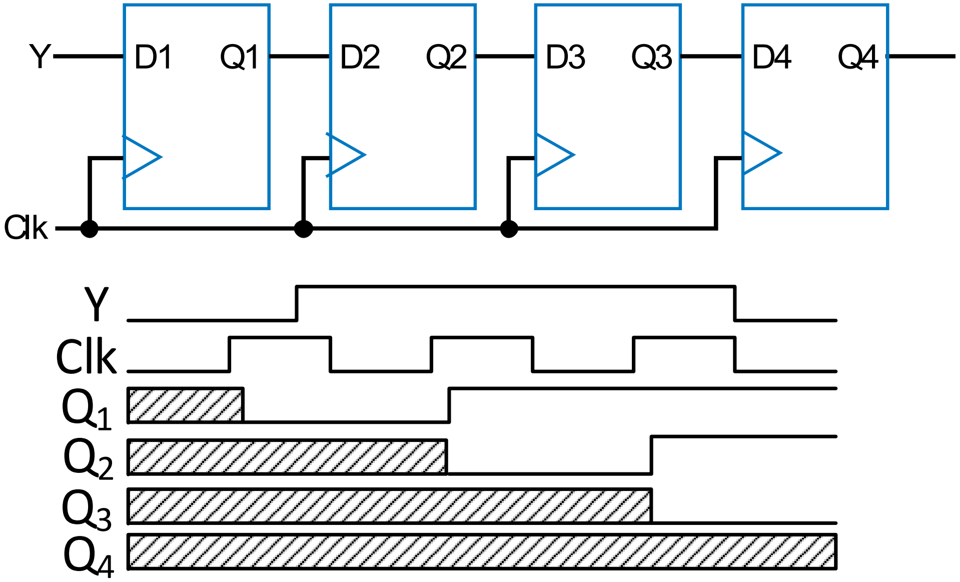
Q4(a) Differentiate between a level-sensitive D-latch and edge-triggered D-flip-flop with the help of a timing diagram.

(5 marks)

ANS: A level sensitive D-latch will propagate signals though the latch when the enable signal is high. An edge triggered D-FF uses a master-slave arrangement so that the signals move through the FF only during the clock transition period.

Consider:





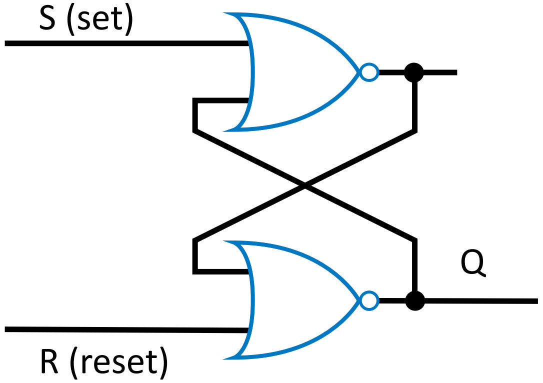
# Selected Past Exam Questions

## CE/CZ1005 2017-2018 Semester 1 (Nov/Dec 2017)

Q4(a) What is a SR latch? Briefly describe how it works.

(6 marks)

ANS: A latch has 2 stable states, Q=1 and Q=0. A SR latch sets Q to 1 when S=1 and resets Q to 0 when R=1. When S=R=0 the latch holds its value. The output is undefined when S=R=1. The simplest SR latch is constructed using cross-coupled NOR gates. It is defined by:



| S | R | Q+ | Function |
|---|---|---|---|
| 0 | 0 | Q | Store |
| 0 | 1 | 0 | Reset |
| 1 | 0 | 1 | Set |