IE1204 Digital Design

L8: Memory Elements: Latches and Flip-Flops. Counter

KTH Informations- och kommunikationsteknik

Elena Dubrova

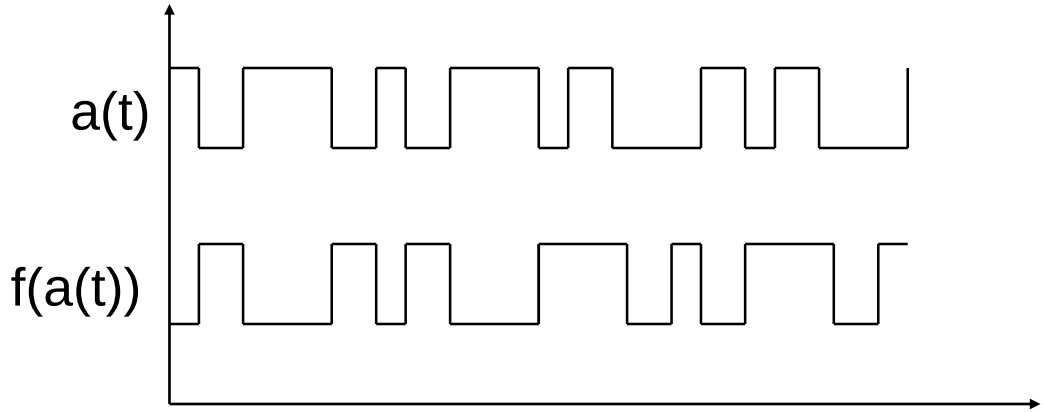
KTH / ICT / ES

dubrova@kth.se

This lecture

- BV pp. 383-418, 469-471

Sequential System



A sequential system has a built-in memory - the output depends therefore BOTH on the current and previous value(s) of the input signal

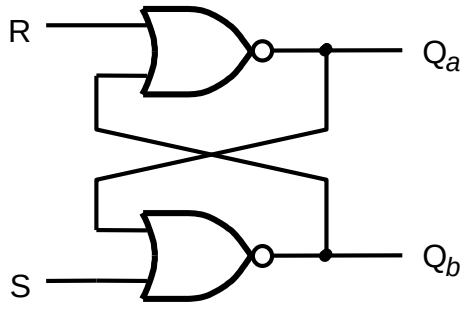
Lecture 8 - Lecture 13

How do we get the hardware to remember something?

- To remember something, we have to somehow retain the information
- One way is to store information in the form of a charge on a capacitance (DRAM)
- Another way is to let the information "run around in a circle and bit its own tail"

SR-latch (NOR)

- SR-latch can be implemented with NOR gates
- SET and RESET inputs are active high
- SET and RESET should not be active at the same time!



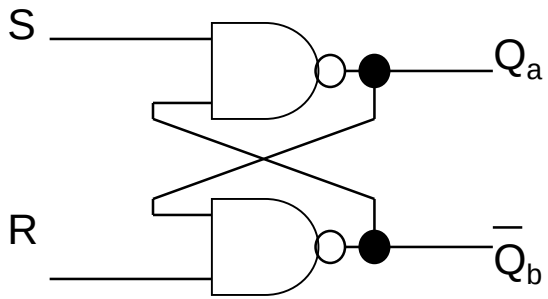
(A) Circuit

S	R	Q_a	Q_b	
0	0	0/1	1/0	(No change)
0	1	0	1	
1	0	1	0	
1	1	0	0	Prohibited input combination (causes oscillation)

(B) Characteristic Table

SR-latch (NAND)

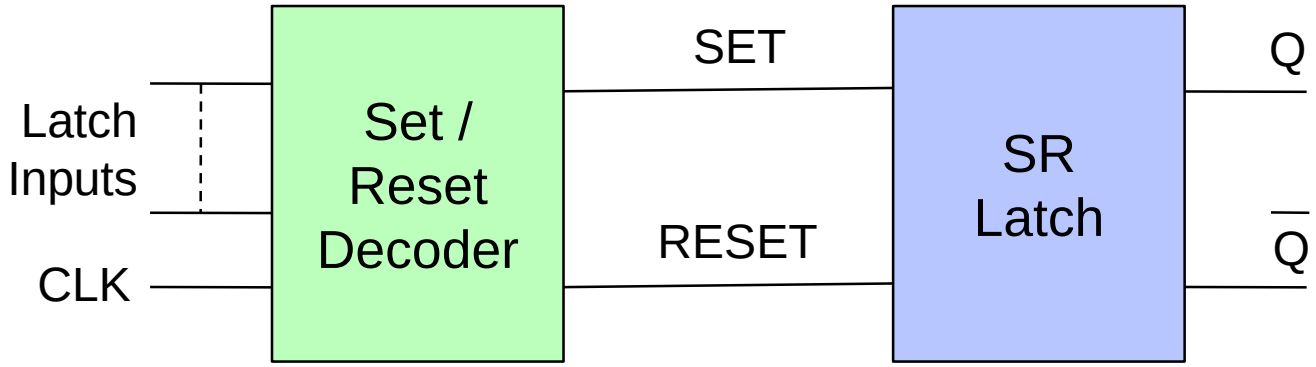
- SR-latch can also be implemented with NAND gates
- SET and RESET inputs are active low



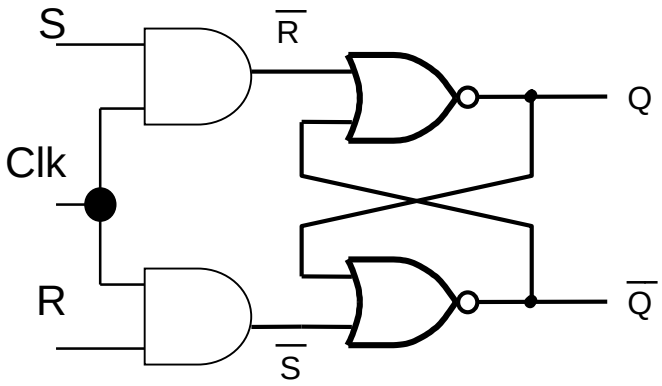
S	R	Q_a	Q_b	
0	0	1	1	Prohibited input combination
0	1	1	0	
1	0	0	1	
1	1	0/1	1/0	(No change)

Construction of clocked latch

- To ensure that the state can only be changed at certain points in time, a special *Clock* signal is used



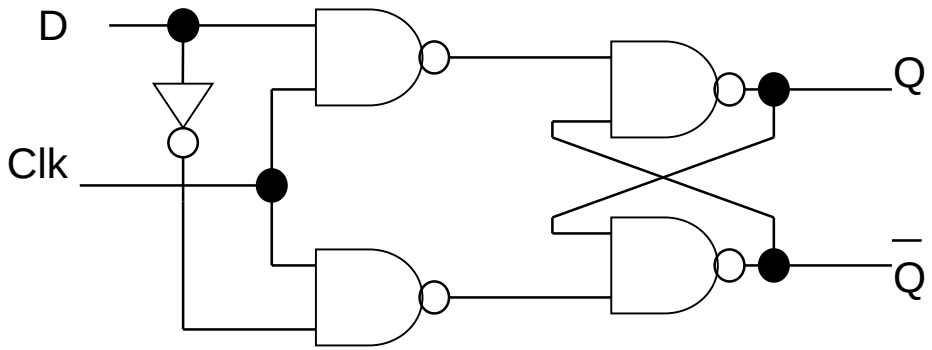
Gated SR-Latch



Clk	S	R	Q(t+1)
0	x	x	Q(t) (no change)
1	0	0	Q(t) (no change)
1	0	1	0
1	1	0	1
1	1	1	x

"Prohibited location"

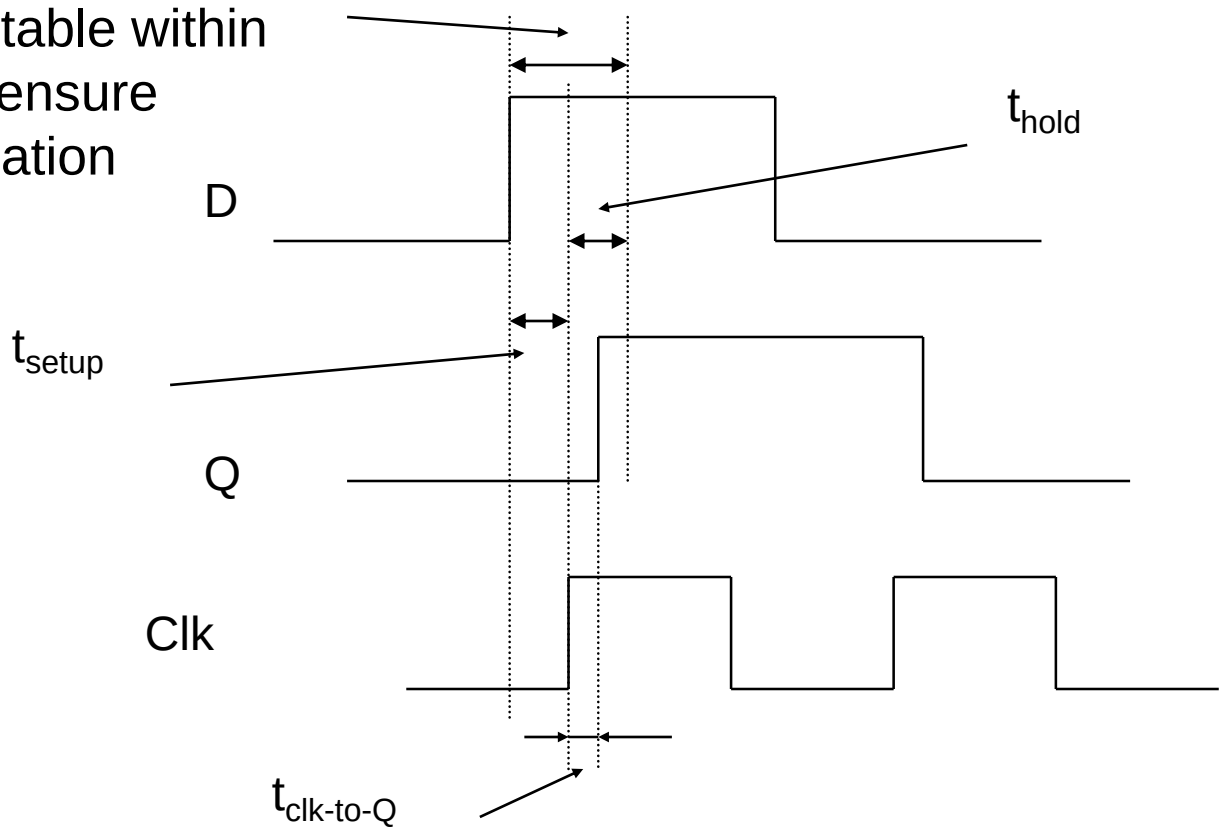
Gated D-Latch



Clk	D	Q(t+1)
0	x	Q(t)
1	0	0
1	1	1

Setup & Hold Time

D must be stable within this area to ensure correct operation



How do we create a sequence?

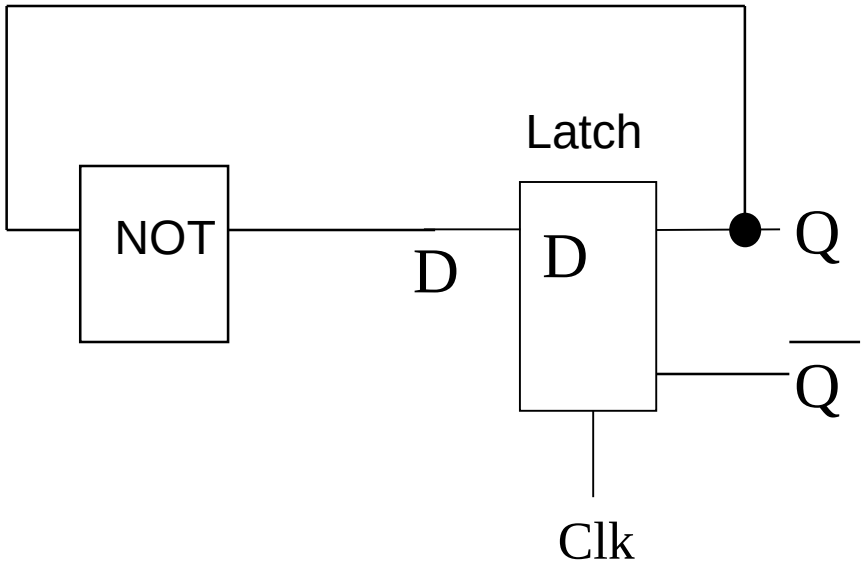
- We get a sequence if we take a value and then determine the next value based on the current value.

Ex: 0,1,0,1, ...

the next value = NOT (present value)

- We need to process (invert) the current value and then remember it until the next value is calculated

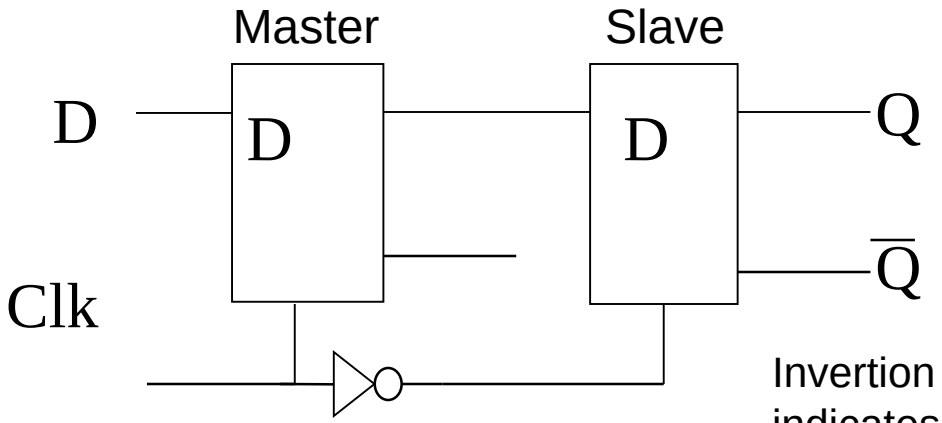
Sequential machines (cont.)



Problems !!! If CLK is 1 for a long time, values with a period of T_{latch}+ T_{NOT} just spin around

We want to design a storage element which changes its state no more than once during one clock cycle

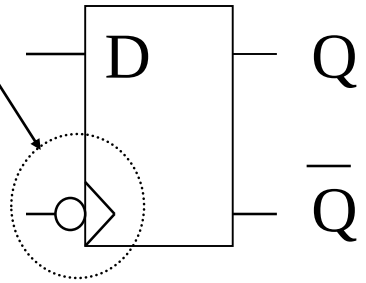
Master-slave D-flip flop



Clk	D	Q(t+1)
↓	0	0
↓	1	1
1	x	Q(x)
0	x	Q(x)
↑	x	Q(x)

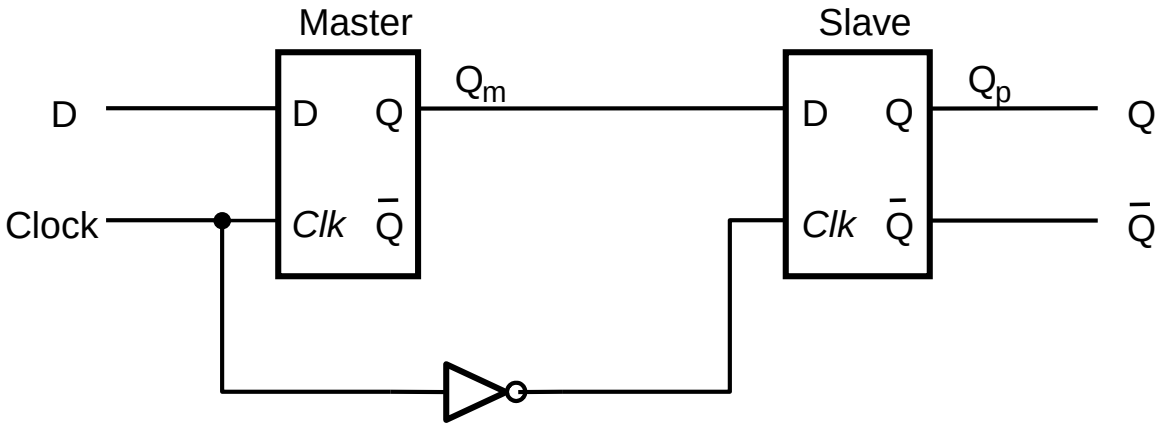
Invertion ring at CLK indicates a negative edge.

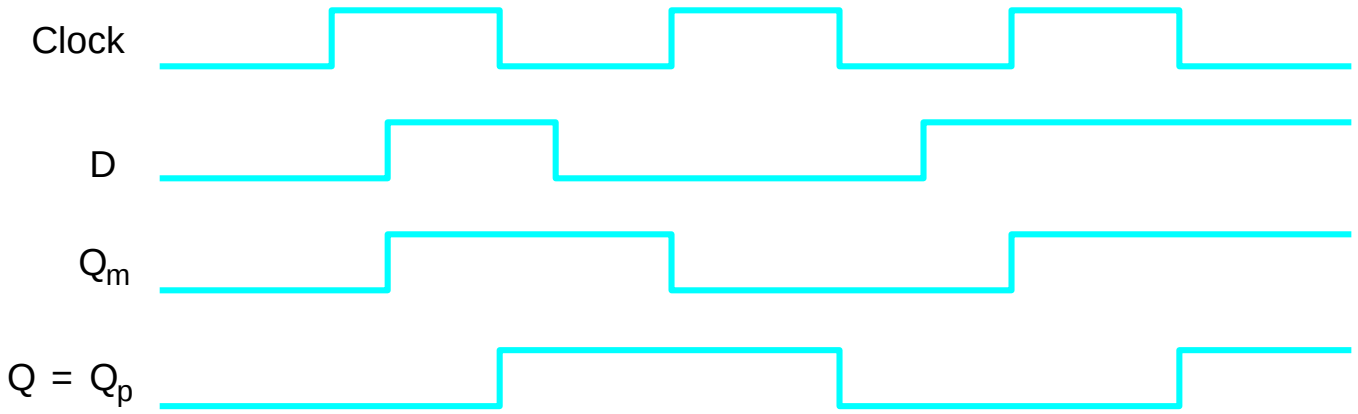
Negative edge triggered D-flip flop



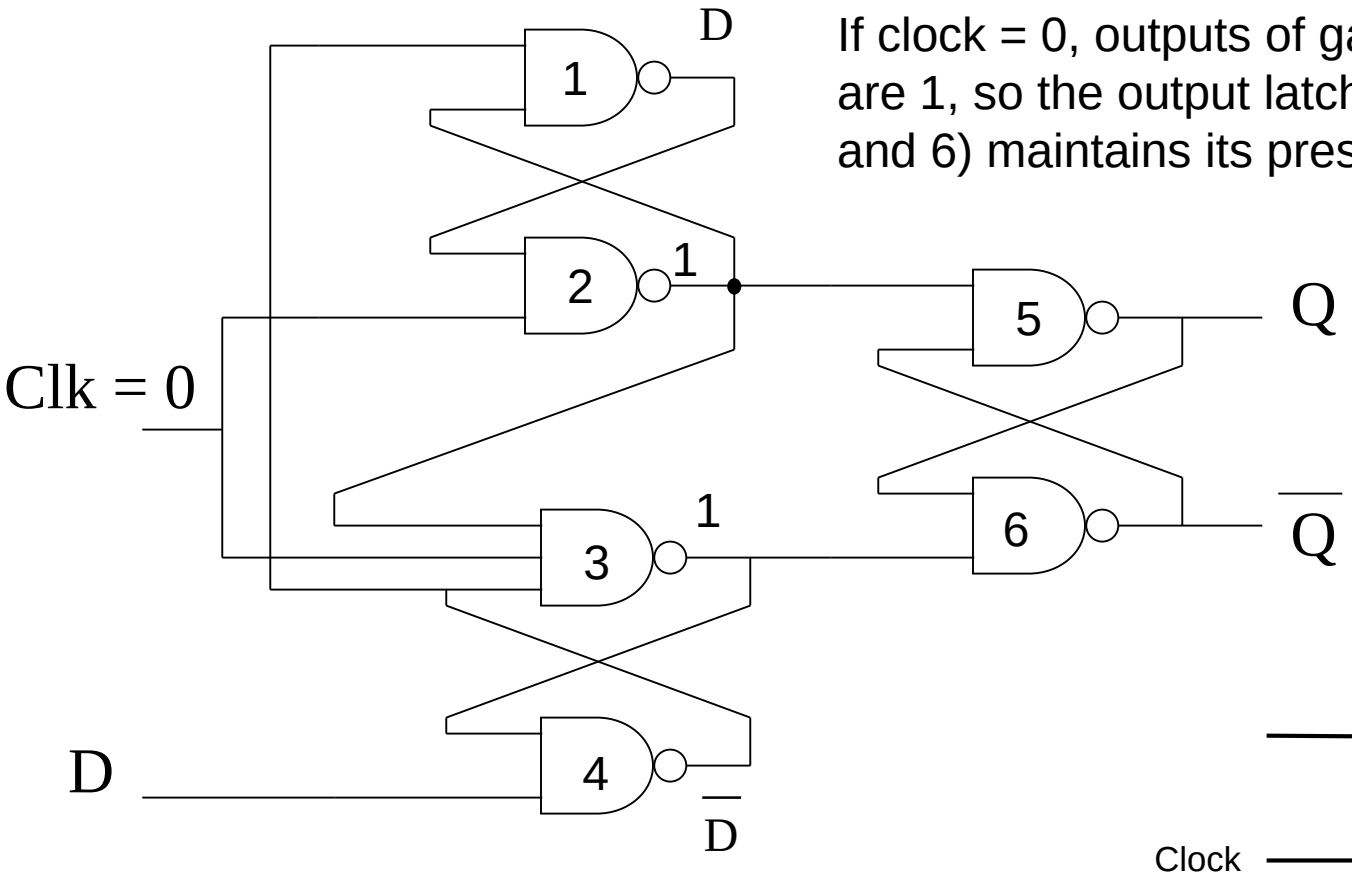
Solution: Connect the two D-latches after each other!

Timing Chart Master-slave

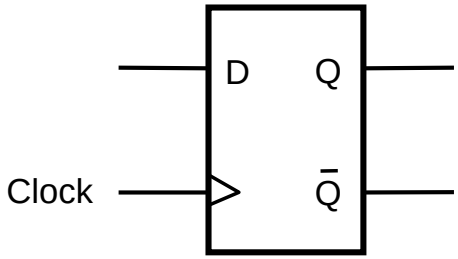




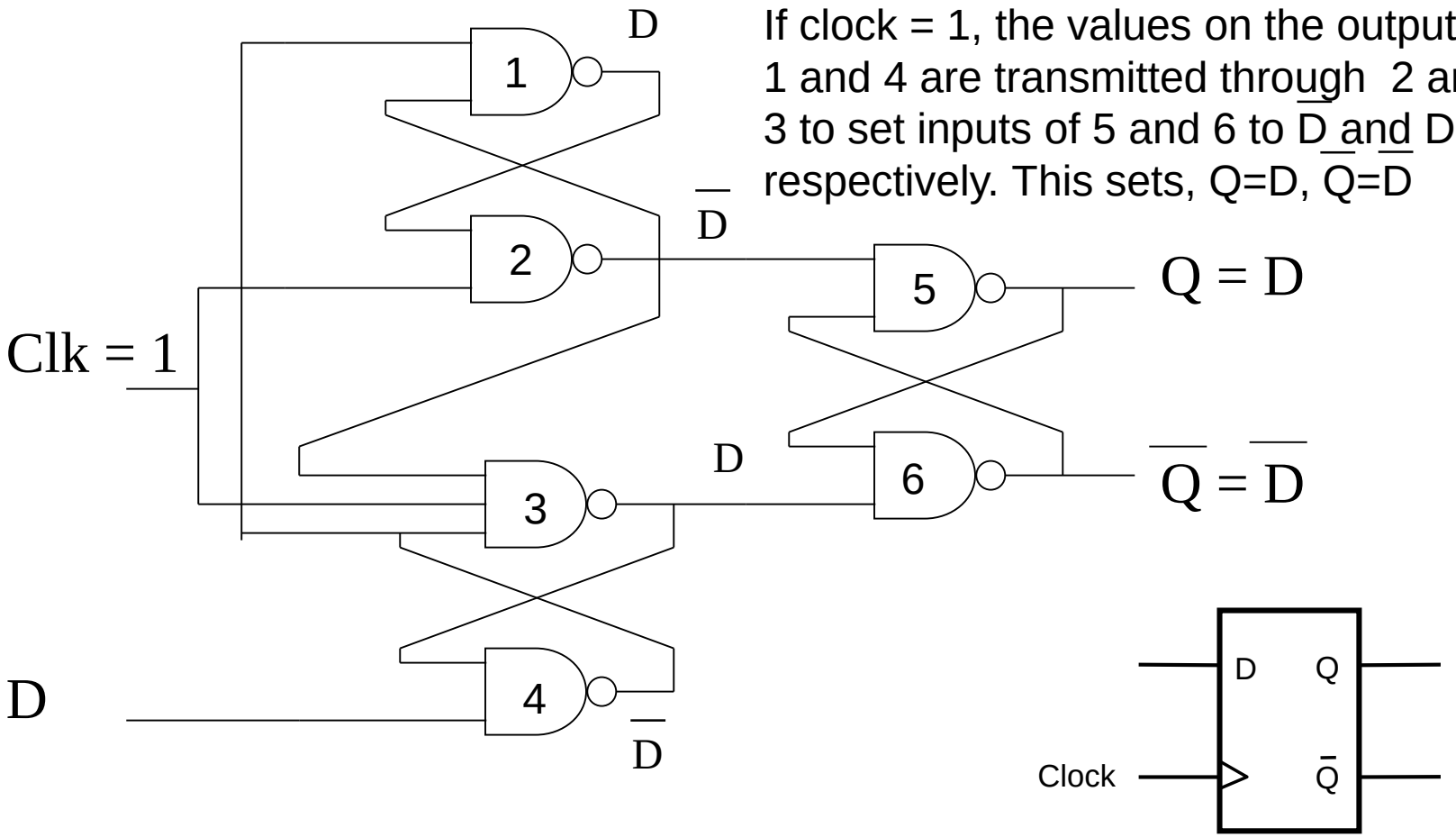
Positive edge-triggered D flip-flop



If clock = 0, outputs of gates 2 and 3 are 1, so the output latch (gates 5 and 6) maintains its present state



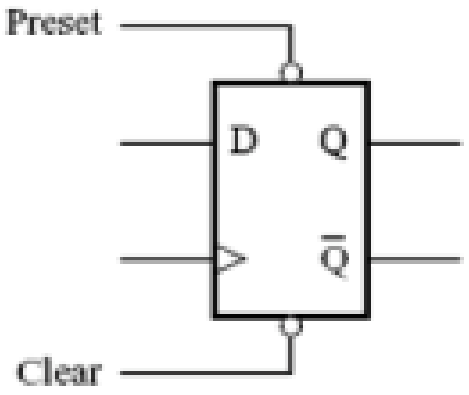
Positive edge-triggered D flip-flop



If clock = 1, the values on the outputs of 1 and 4 are transmitted through 2 and 3 to set inputs of 5 and 6 to $\overline{D}$ and D, respectively. This sets, Q=D, $\overline{Q}$=$\overline{D}$

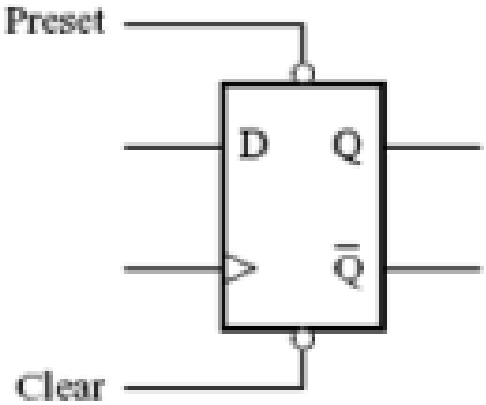
Flip-Flops with Clear and Preset inputs

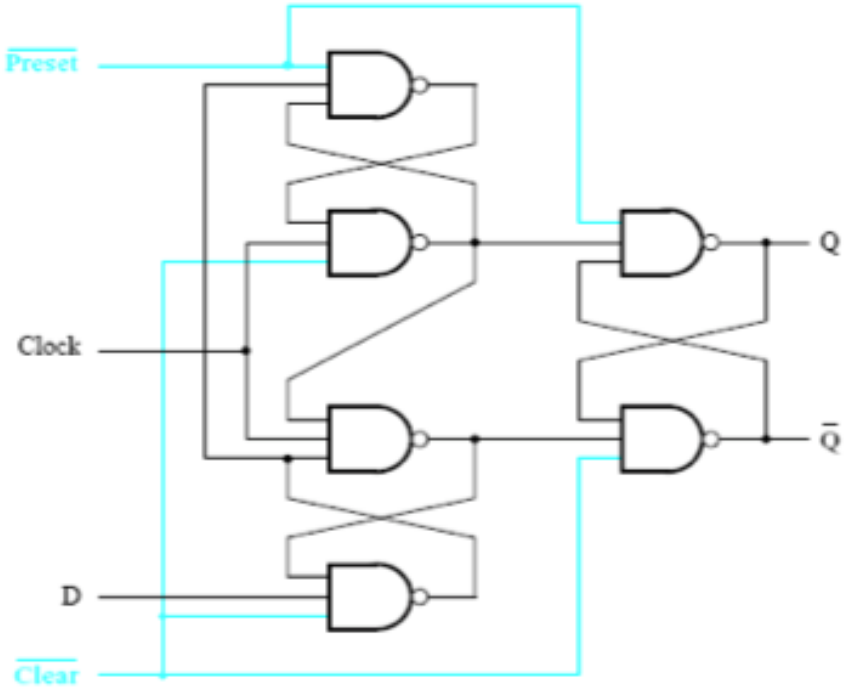
- It is important for the design sequential circuits to be able to set flip-flops to predetermined values
  - Preset: Sets the flip-flop to 1
  - Clear: Sets the flip-flop to 0



Asynchronous reset

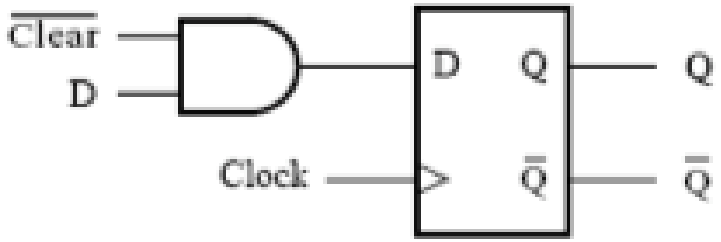
- An asynchronous reset (clear) means that the flip-flop will change its state to 0 immediately after the reset is active





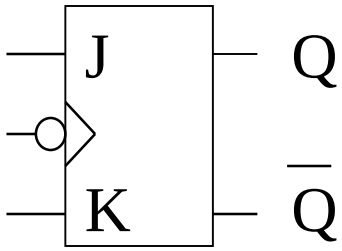
Synchronous reset

- A synchronous reset causes the flip-flop to take state 0 at the next clock edge
- Synchronous reset is implemented with an additional logic



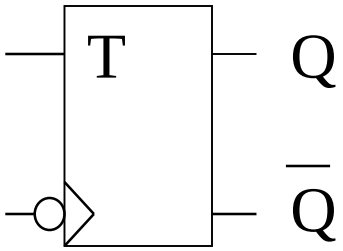
Other common types of flip-flops

JK flip-flop (by Jack Kilby - Nobel Prize 2000)



Clk	J	K	Q	$\overline{Q}$
↓	0	0	Q(t)	Q(t)
↓	0	1	0	1
↓	1	0	1	0
↓	1	1	$\overline{Q(t)}$	$\overline{Q(t)}$

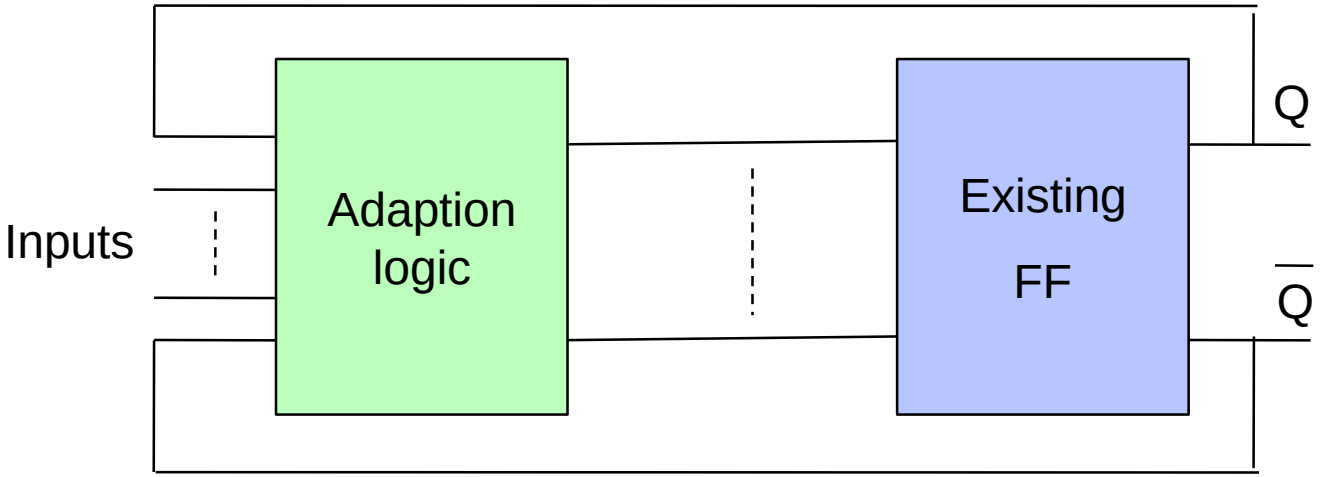
T-flip-flop (T = Toggle)



Clk	T	Q	$\overline{Q}$
↓	0	Q(t)	Q(t)
↓	1	$\overline{Q(t)}$	$\overline{Q(t)}$

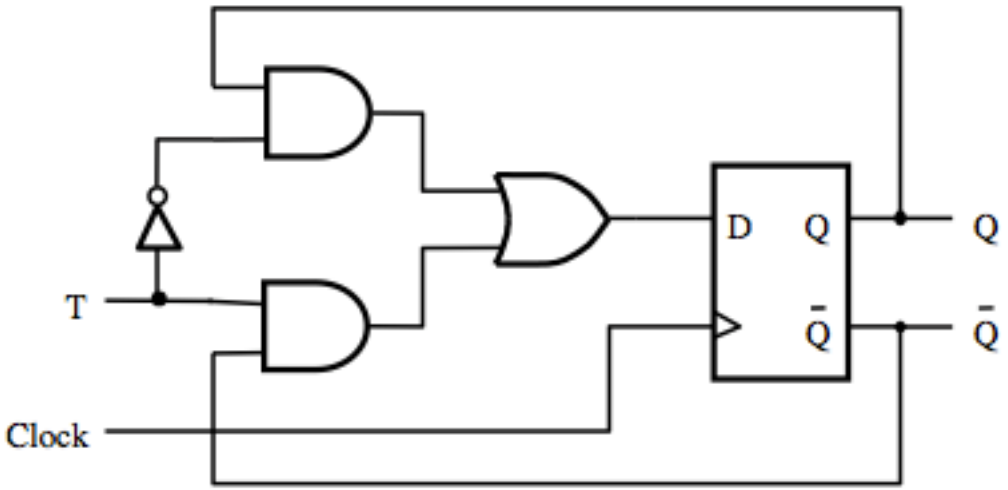
Construction of new flip-flops

- One can construct new flip-flops based on the existing types



Construction of the T flip-flop with D flip-flop

- One can construct the new flip-flops based on an existing type



Clk	D	Q(t+1)
↑	0	0
↑	1	1

Clk	T	Q(t+1)
↑	0	Q(t)
↑	1	$\bar{Q}$(t)

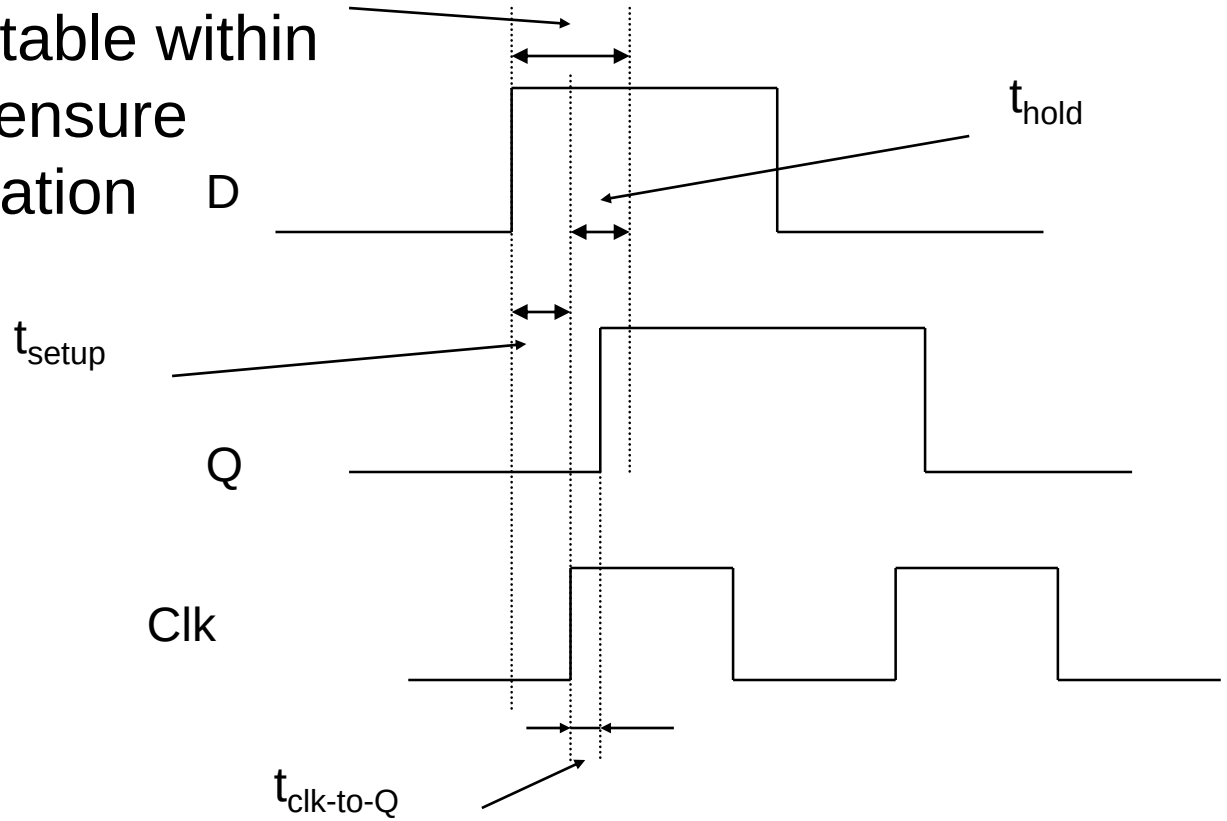
Toggles at each positive edge of clock

Timing Analysis

- It is possible to determine the maximum frequency in a sequential circuit by having information about
  - Gate delays t_{logic}
  - Setup time t_{su} of flip-flops
  - Hold time t_h of flip-flops
  - Clock-to-output t_{cQ} time

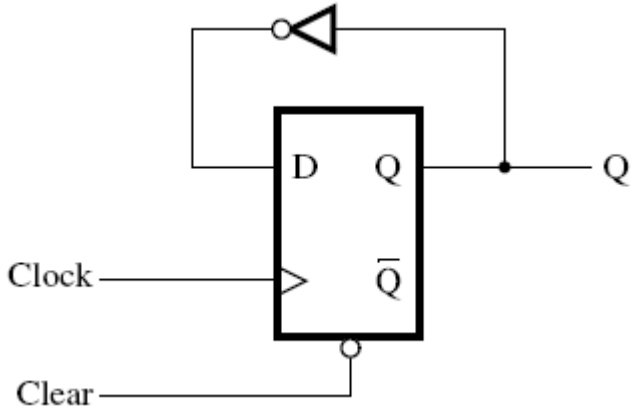
Setup & Hold Time

D must be stable within this area to ensure correct operation



What is the maximum frequency?

- Gate delays
  - $t_{logic} = t_{NOT} = 1.1$ ns
- Setup time
  - $t_{su} = 0.6$ ns
- Hold time
  - $t_h = 0.4$ ns
- Clock-to-output
  - $t_{cQ} = 1.0$ ns

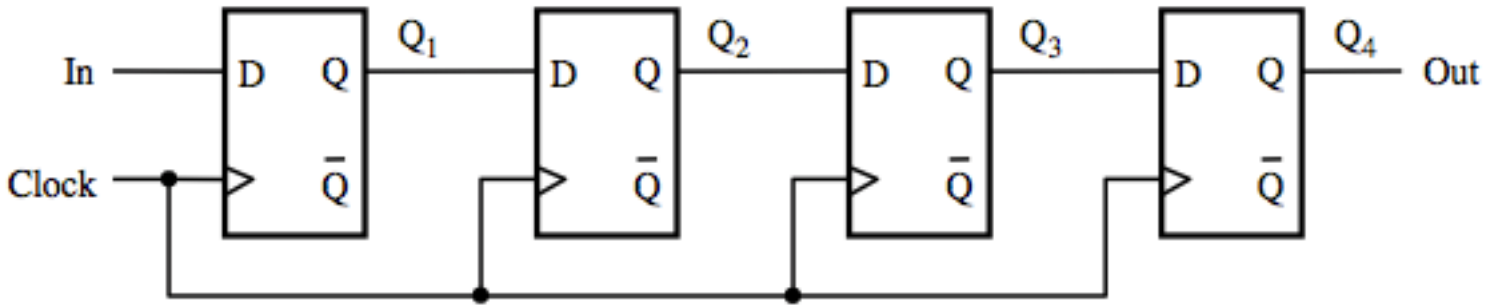


$T = t_{su} + \max(t_h, t_{cQ}) + t_{logic} = 2.7$ ns

$F = 1/T = 370$ MHz

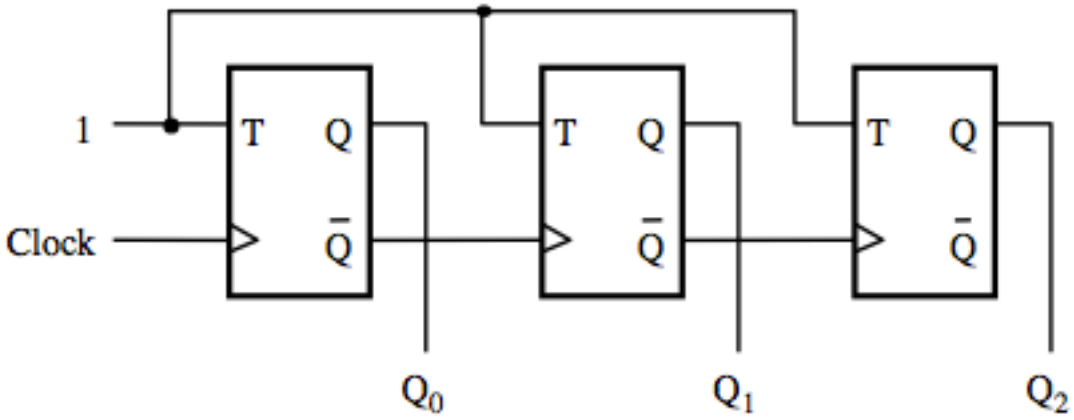
Shift Register

- A shift register contains several flip-flops
- For each clock cycle, we shift all values from left to right
- Many designs use shift registers and values Q_4, ..., Q_1 as input to other components



Asynchronous counter

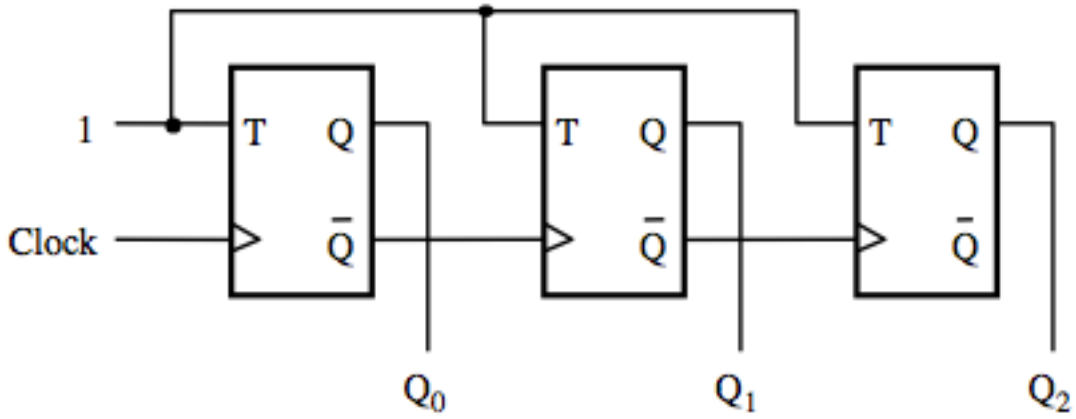
- One can realize a counter with flip-flops
- The examples below show an *asynchronous* counter
- Some clock inputs are coupled to the $\overline{Q}$ output of the previous flip-flop

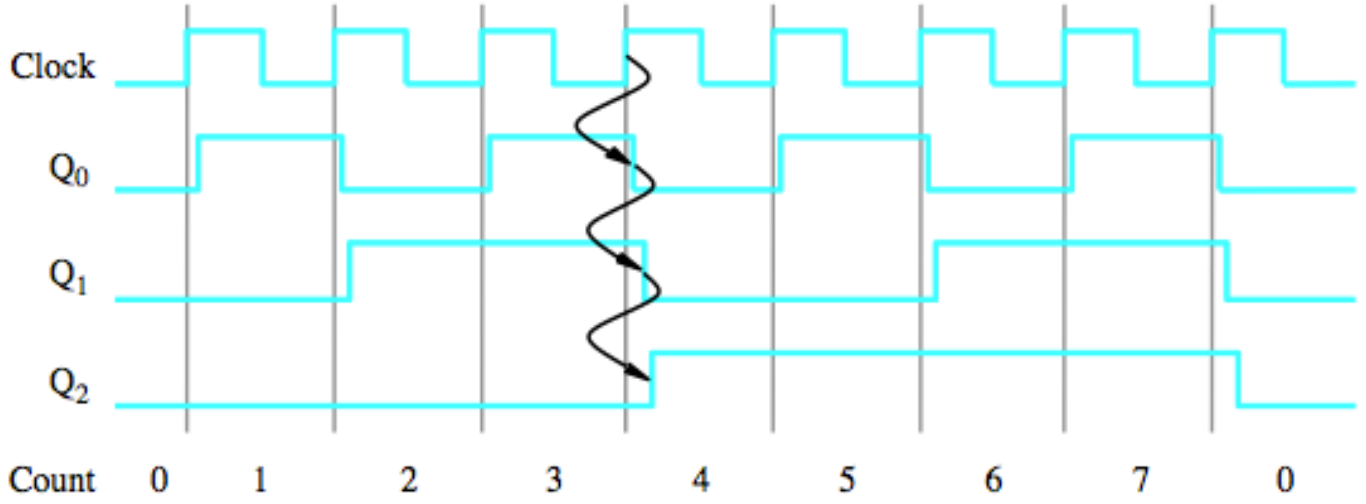


Asynchronous 3-bit counter

Clk	T	Q(t+1)
↑	0	Q(t)
↑	1	$\overline{Q(t)}$

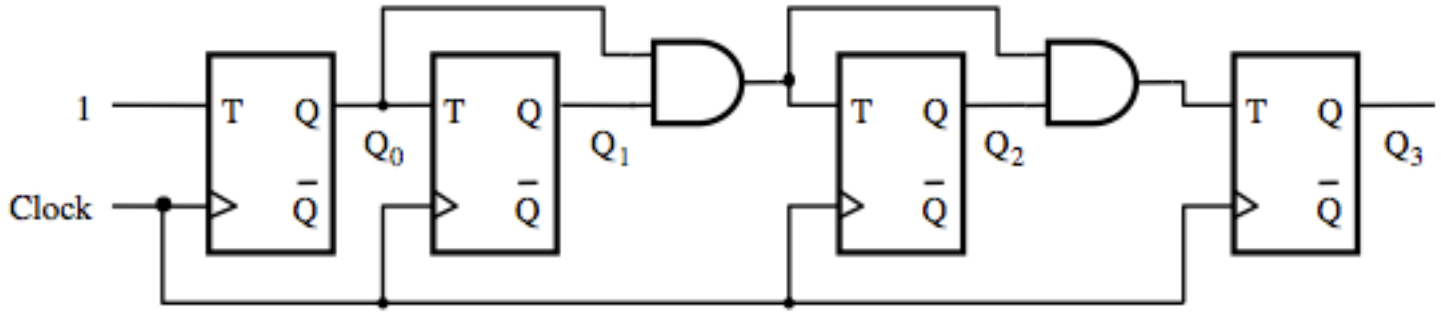
Toggles at each positive edge of clock





Synchronous counter

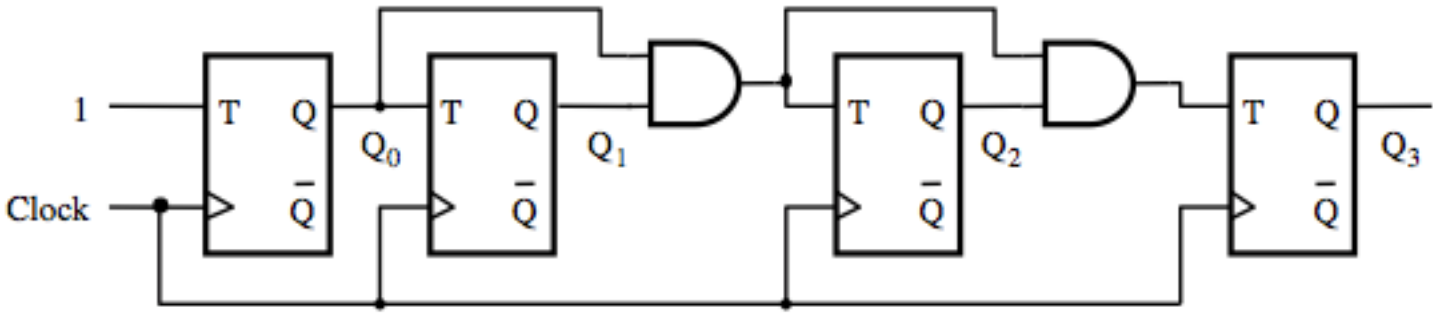
- In a *synchronous* counter clock inputs of flip-flops are connected to the same clock signal

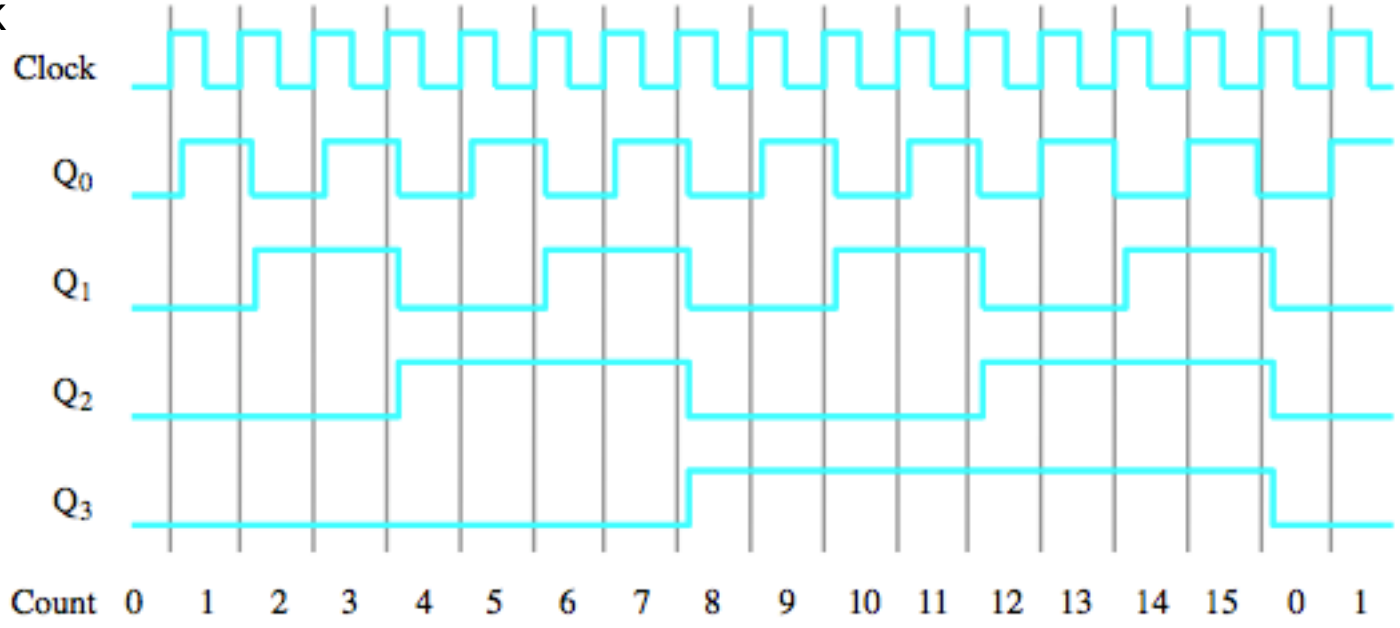


Synchronous counter

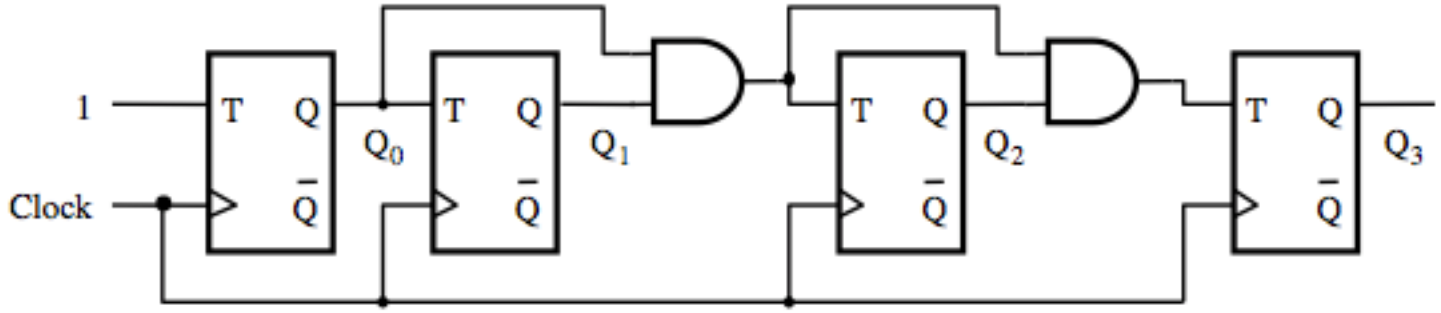
Clk	T	Q(t+1)
↑	0	$\underline{Q}(t)$
↑	1	$Q(t)$

Toggles at each positive edge of clock





What is the maximum frequency?



- The critical path determines the maximum frequency!
- This is the longest combinational path from Q_0 through the two AND gates to the input of flip-flop that computes Q_3
  - t_{logic} thus is equivalent to the delay of two AND gates

Summary

- Memory Elements
  - Latches
  - Flip-Flops
- Shift registers
- Counters
- Next lecture: BV pp. 485-507