Digital Electronics and Microprocessor Systems (ELEC211)

Dave McIntosh and Valerio Selis

dmc@liv.ac.uk

v.selis@liv.ac.uk

Digital 4: Introduction to sequential logic

UNIVERSITY OF LIVERPOOL

Outline

- Sequential circuits
- Propagation / gate delay
- Timing diagrams
- Latches
- Flip flops

Previous material

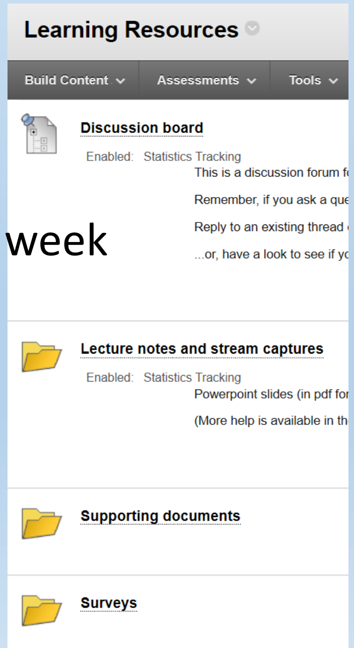
Basic gates ✓

Combinational and sequential logic ✓

Use VITAL!:

- Stream lectures
- Handouts
- Notes and Q&A each week
- Discussion Board
- Exam resources

www.liv.ac.uk/vital



Discussion board

Enabled: Statistics Tracking

This is a discussion forum f

Remember, if you ask a que

Reply to an existing thread

...or, have a look to see if yo

Lecture notes and stream captures

Enabled: Statistics Tracking

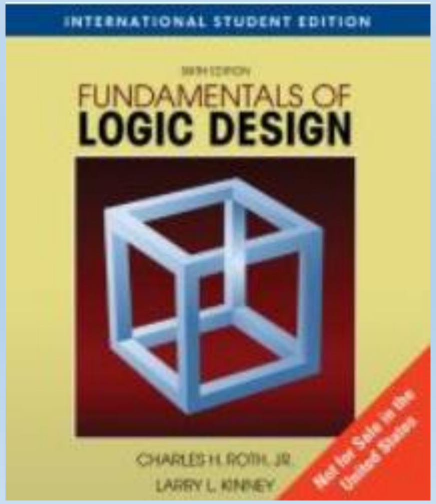
Powerpoint slides (in pdf for

(More help is available in th

Supporting documents

Surveys

Course textbook – please borrow and use it! ➡


INTERNATIONAL STUDENT EDITION
FUNDAMENTALS OF
LOGIC DESIGN
CHARLES H. ROTH, JR.
LARRY L. KINNEY
Not for Sale in the United States

Current channel depends on previous channel
… which depends on previous channel
… which depends on previous channel
…

Previous state is significant!

https://www.publicdomainpictures.net/en/view-image.php?image=117613&picture=remote-control

https://pixabay.com/illustrations/tv-television-watch-1673224/

Sequential logic – revision

Sequential switching circuits have the property that the output depends not only on the present input but also on the **past sequence** of inputs.

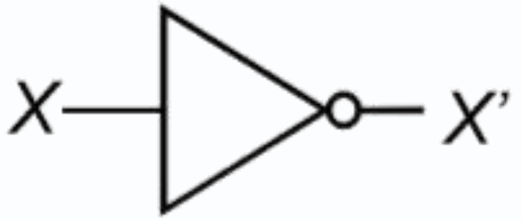
So the “state” becomes significant to determining the output.

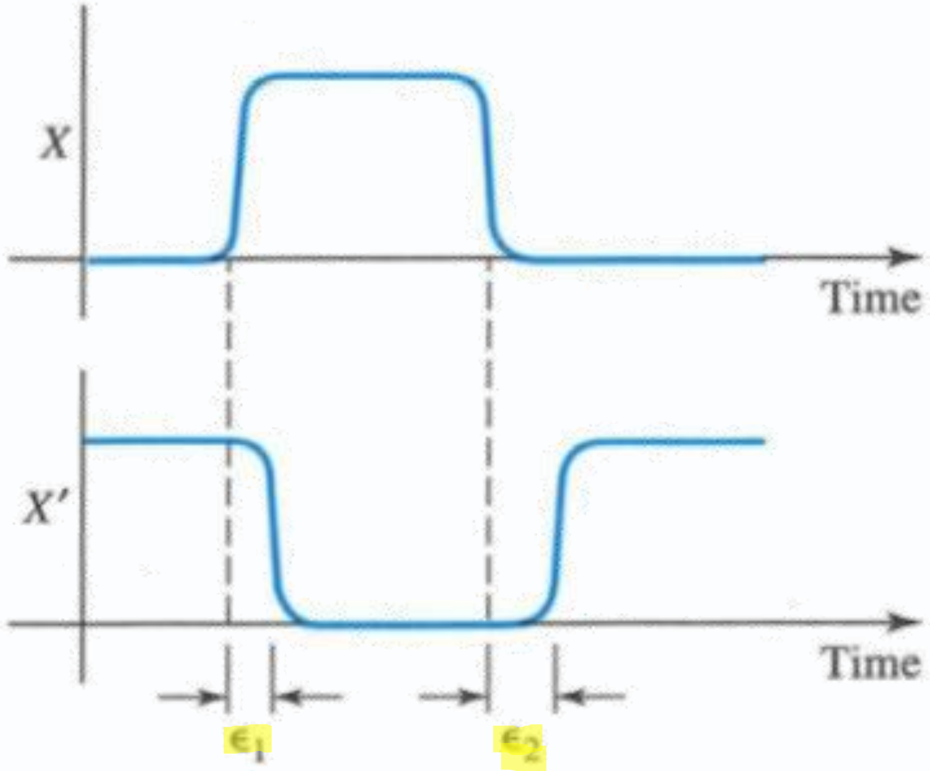
Latches and flip-flops are commonly used as memory devices in sequential circuits.

In order to construct a switching circuit that has memory, such as a latch or flip-flop, we must introduce feedback into the system.

Gate delays and timing diagrams

- The output of a logic gate does not change instantaneously.
  - The output change is going to be delayed with respect to the input change.
  - This delay is called propagation delay
    - Propagation delay from 0 to 1, ϵ_1.
    - Propagation delay from 1 to 0, ϵ_2.

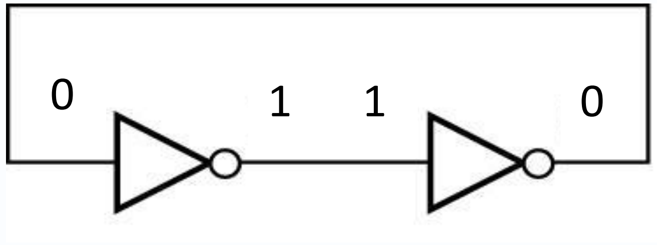


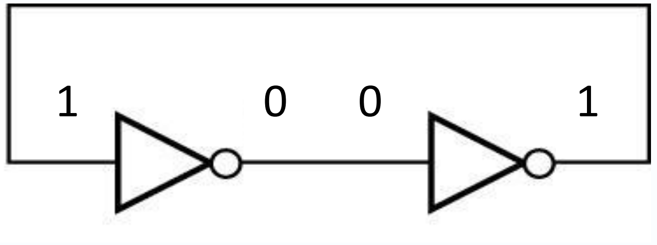


In practice ϵ1 and ϵ2 can be different

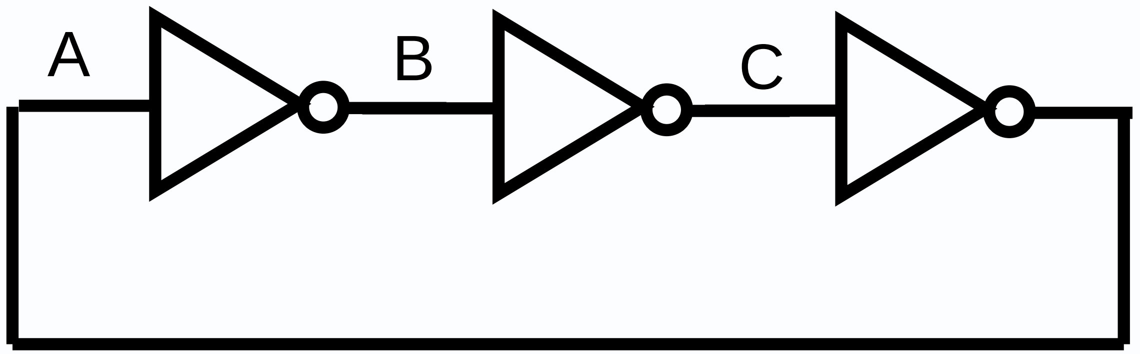
Feedback loop with two Inverters

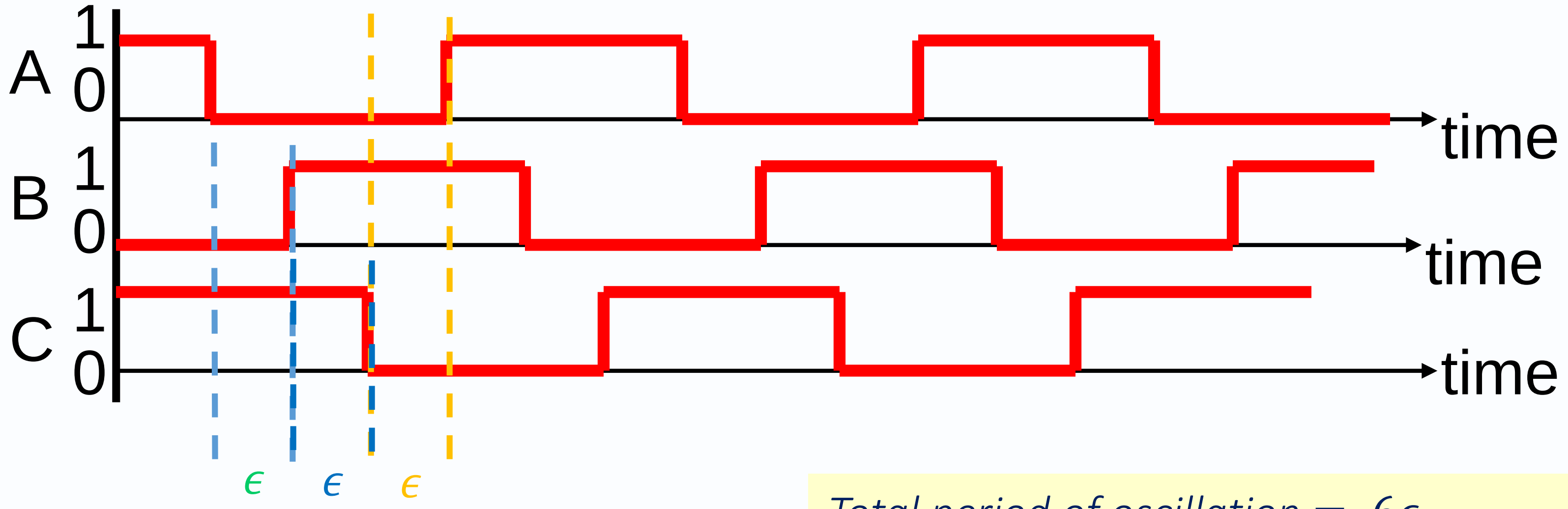
Consider a feedback loop circuit with two inverters.
This circuit has two stable states:





Ring Oscillator – timing diagram

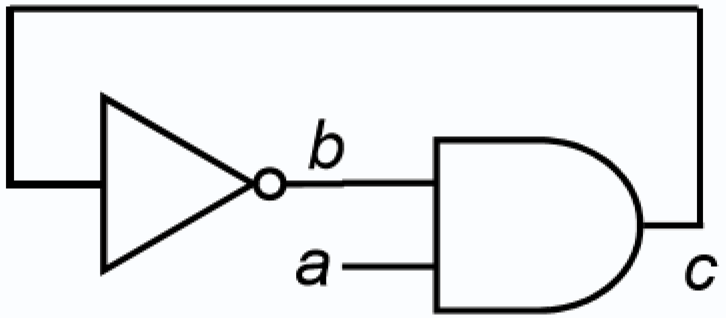




Total period of oscillation = 6ϵ
(assumes propagation delay is identical for each of the three gates)

Question

The inverter in the figure has a propagation delay of 4 ns and the AND gate of 8 ns. Draw a timing diagram for the circuit showing a, b, c. a and c are initially equal to 0, b is initially one. After 20 ns a becomes 1 for 90 ns and then 0 again.



https://www.pickpik.com/door-open-wooden-the-next-doorway-92347

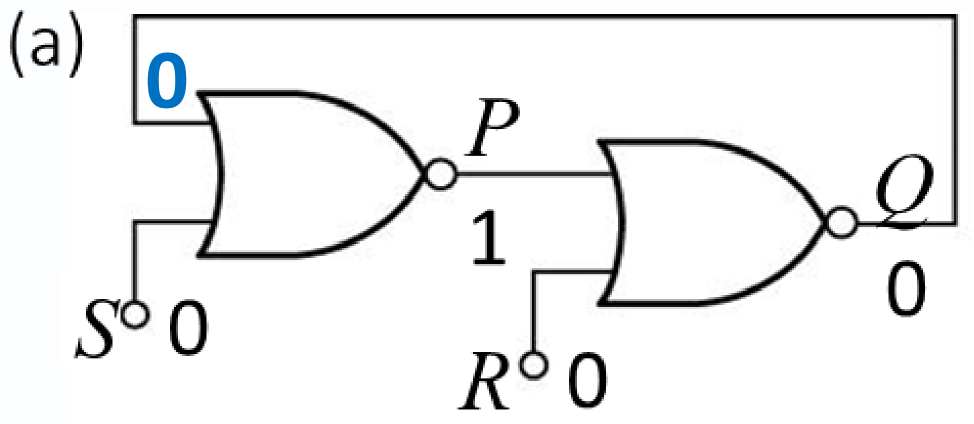
Latches – two stable states

https://www.flickr.com/photos/58026849@N03/7134296191/

Set-Reset Latch

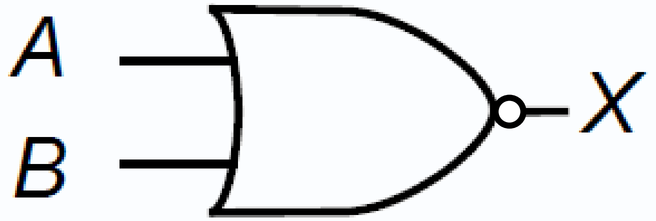
The S-R latch can be put into one of two stable output states.

These are triggered by an input pulse at input S (to **set** to one stable state, P=0, Q=1) or at R (to **reset** to the other stable state, P=1, Q=0)



Initial state: S=R=0, P=Q'=1

NOR



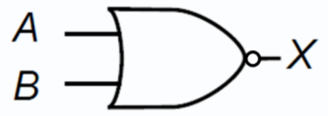
A	B	X
0	0	**1**
0	1	**0**
1	0	**0**
1	1	**0**

Set-Reset Latch

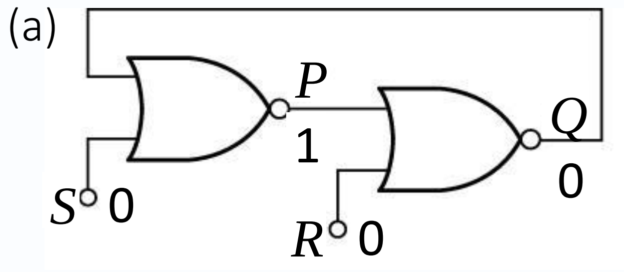
The S-R latch can be put into one of two stable output states.

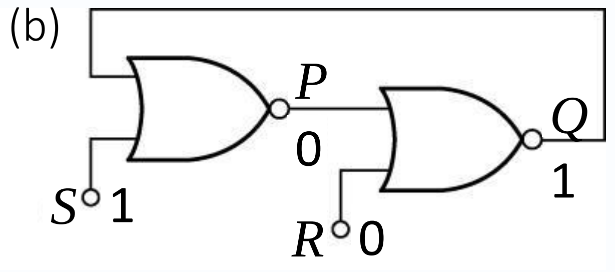
These are triggered by an input pulse at input S (to **set** to one stable state, P=0, Q=1) or at R (to **reset** to the other stable state, P=1, Q=0)

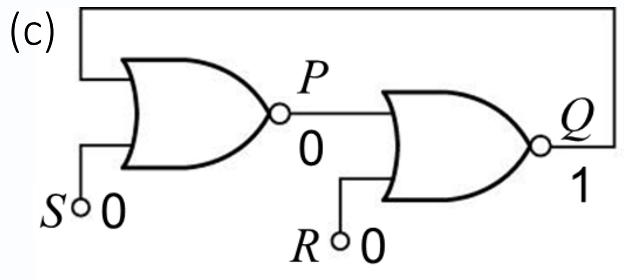
NOR

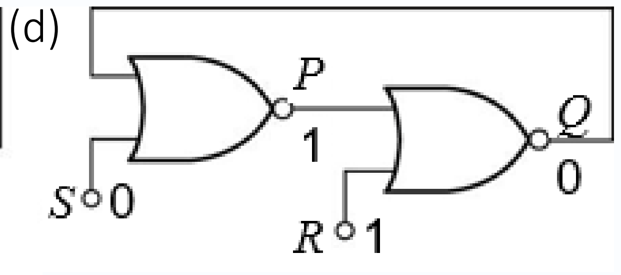


A	*B*	***X***
0	0	**1**
0	1	**0**
1	0	**0**
1	1	**0**









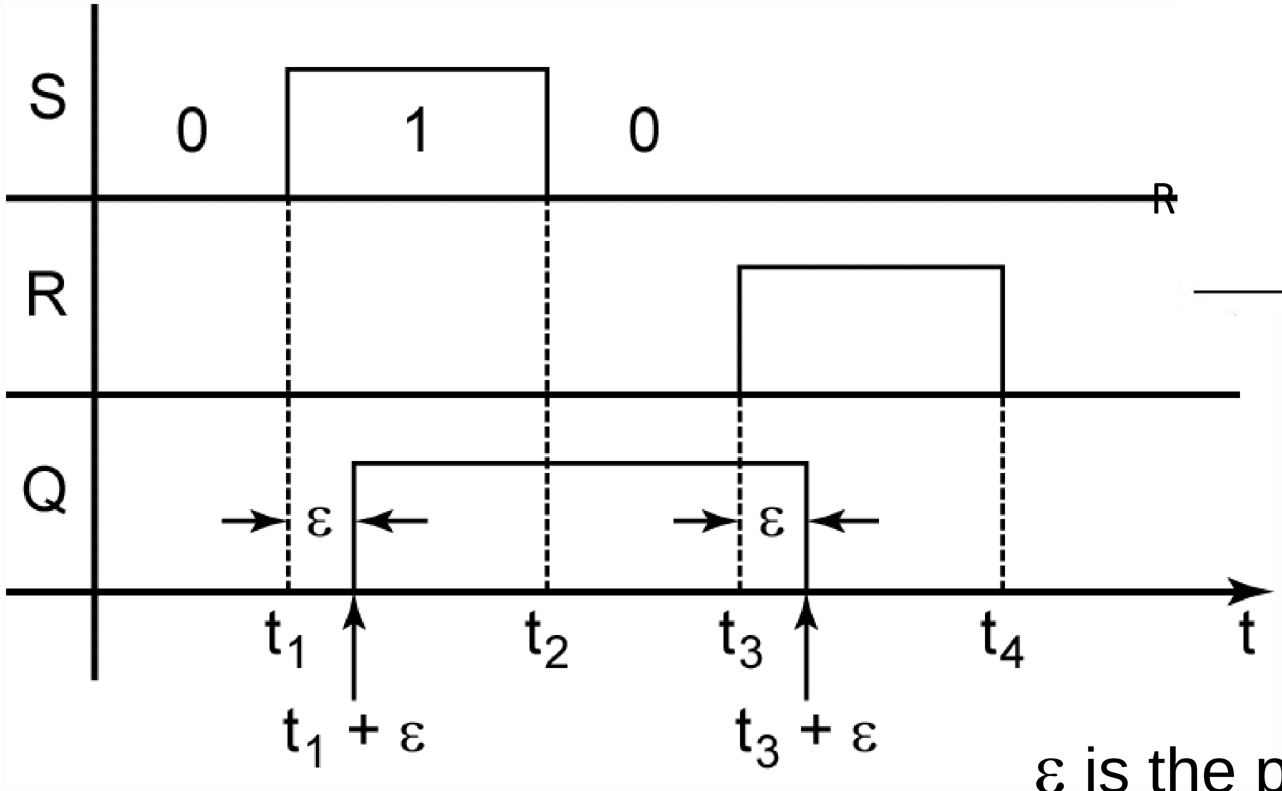
Pulse at S: S=1, R=0, P=Q'=0

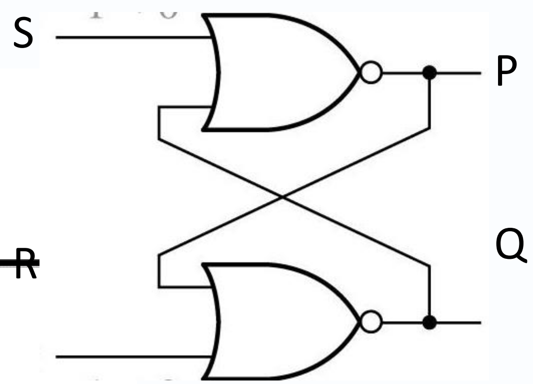
Pulse at R: S=0, R=1, P=Q'=1

Initial state: S=R=0, P=Q'=1

Pulse at S removed: S=0, R=0, P=Q'=0

S-R latch timing





ε is the propagation delay

S-R Latch table

The circuit is said to have "memory" because output depends not just on present inputs, but also on the previous sequence of inputs

S-R latch table

S(t)	R(t)	Q(t)	**Q(t+ε)**
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**0**
1	0	0	**1**
1	0	1	**1**
1	1	0	-
1	1	1	-

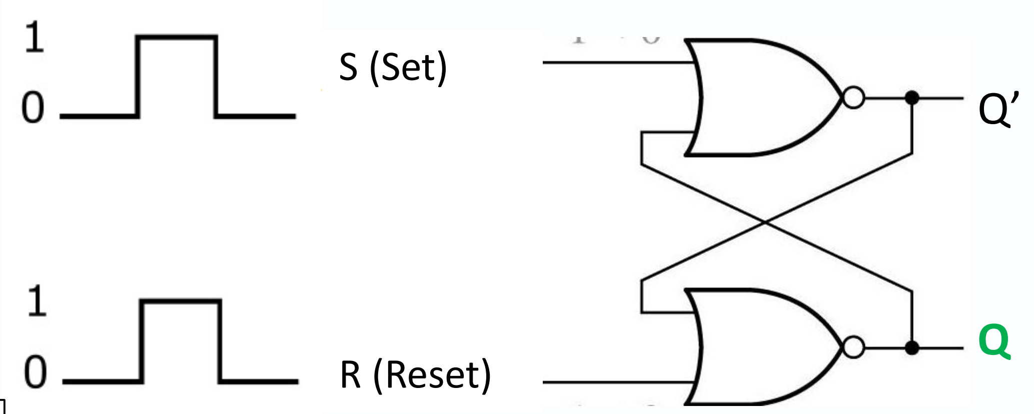
Inputs not allowed

ε is the propagation delay

Q is usually referred to as the output of the S-R latch

Q = 1 is a 'set' state
Q = 0 is a 'reset' state

Cross-coupled form emphasises the gates' symmetry:

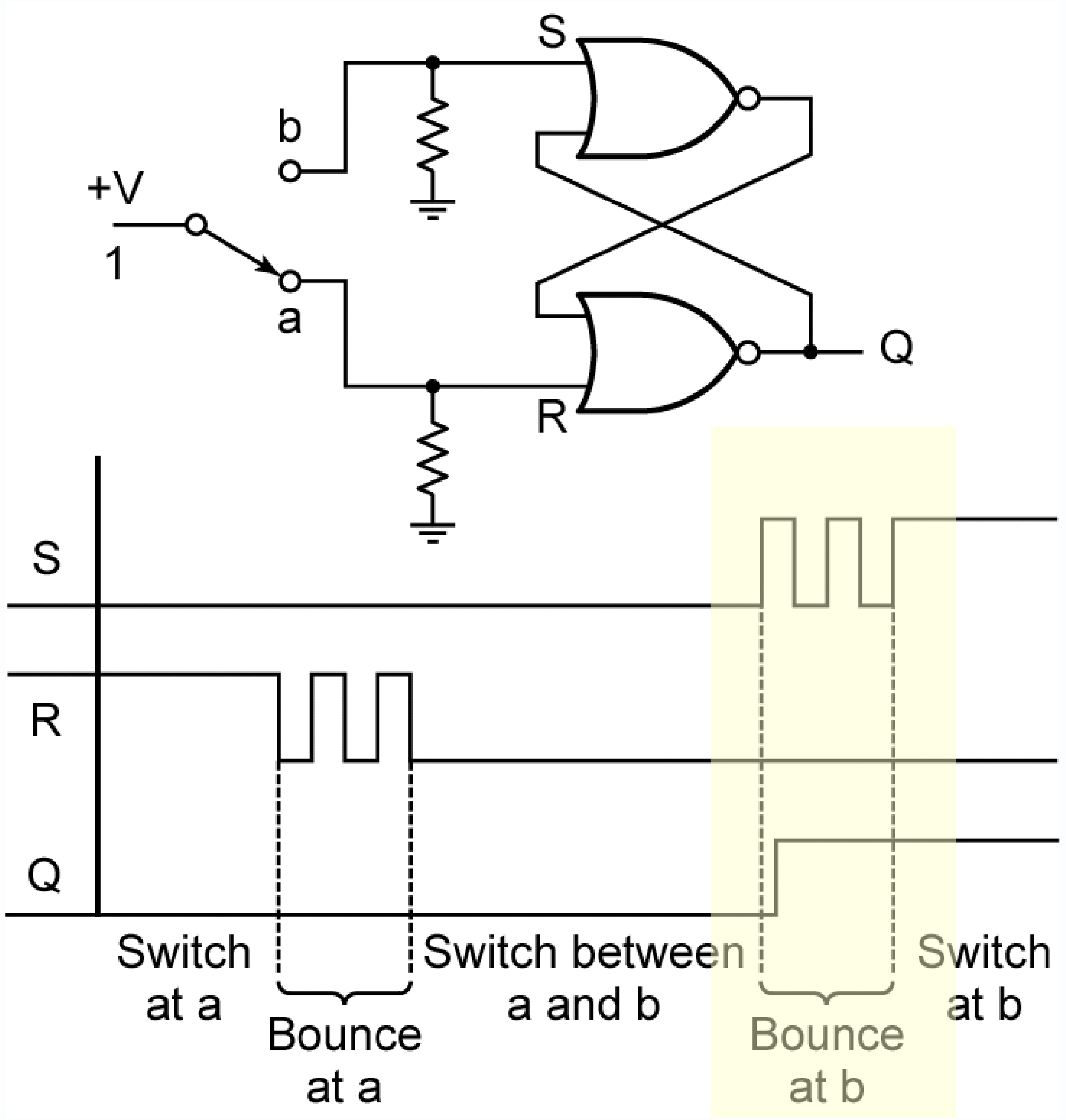


The set-reset latch is ***asynchronous***: output changes shortly after the input, rather than according to an external clock signal

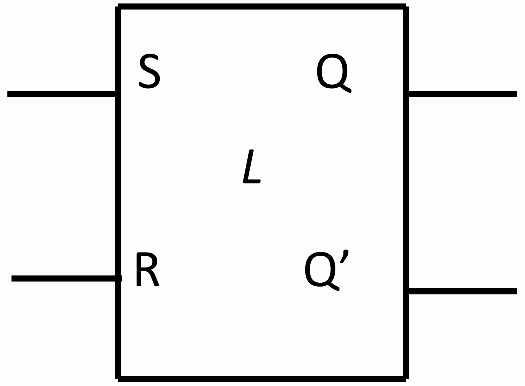
Switch Debouncing with an S-R Latch

When a mechanical switch is opened or closed, the switch contacts vibrate or bounce open & closed several times.

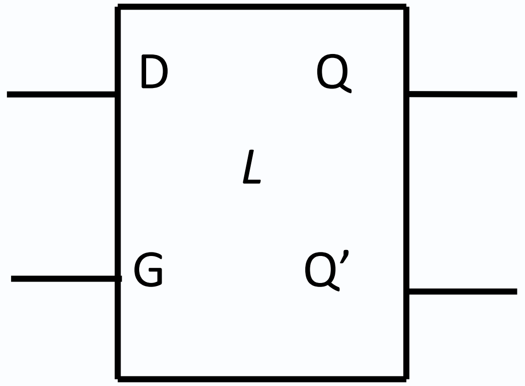
The pull-down resistors assure that when the switch is between *a* and *b*, the latch inputs S and R are at logic 0.



Latch shorthand



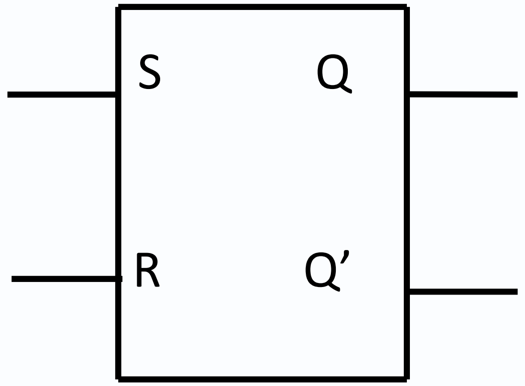
SR Latch



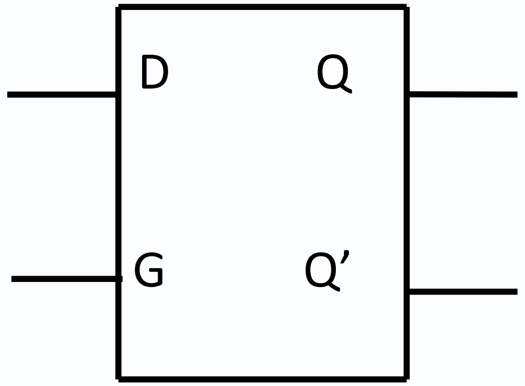
Gated D Latch

G	Q^+
0	mem
1	D

We won’t often include the “*L*”



SR Latch

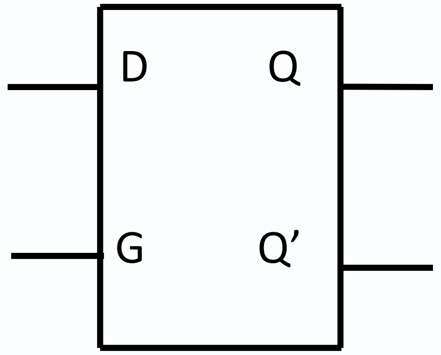


Gated D Latch

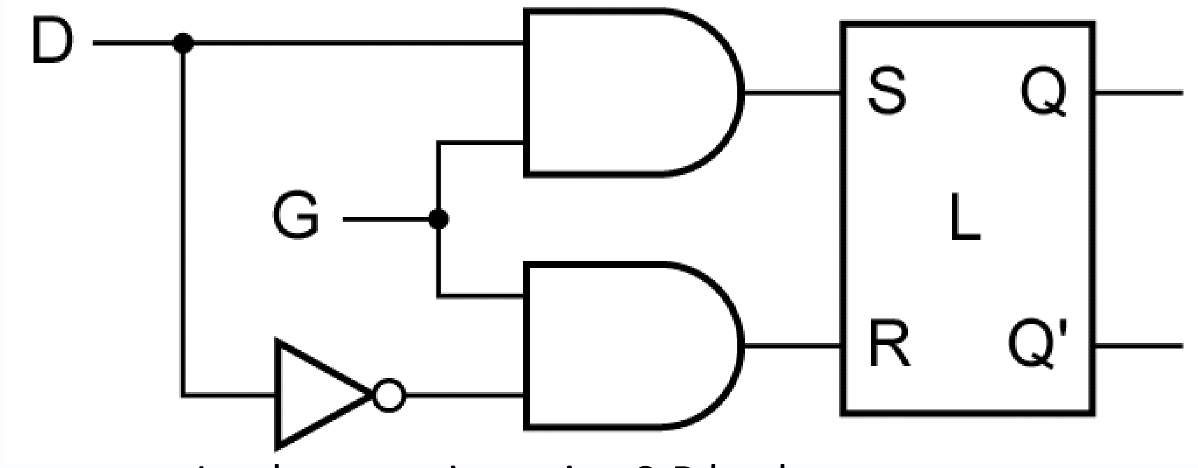
Gated D-Latch or Transparent Latch

A gated D-latch has two inputs, a data input (D) and a gate input (G).

This is a level-sensitive latch with an enabling input (G)



$\equiv$



Implementation using S-R latch

'Transparent' because **if G=1:**

output (Q) = D (data) input

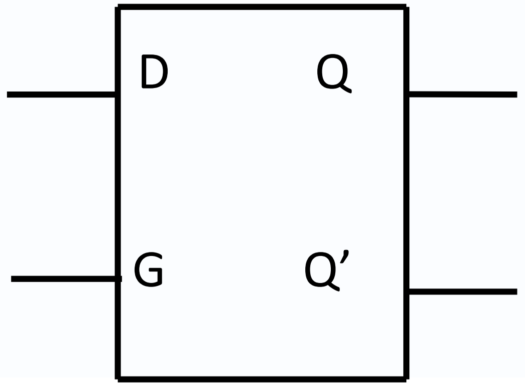
G=0 → S=R=0 → no change (memory)

G=1, D=0 → S=0, R=1 → RESET (Q^+ becomes or stays 0)

G=1, D=1 → S=1, R=0 → SET (Q^+ becomes or stays 1)

Gated D-Latch or Transparent Latch

G=0 → S=R=0 → no change
G=1, D=0 → S=0, R=1 → RESET (Q^+ becomes or stays 0)
G=1, D=1 → S=1, R=0 → SET (Q^+ becomes or stays 1)



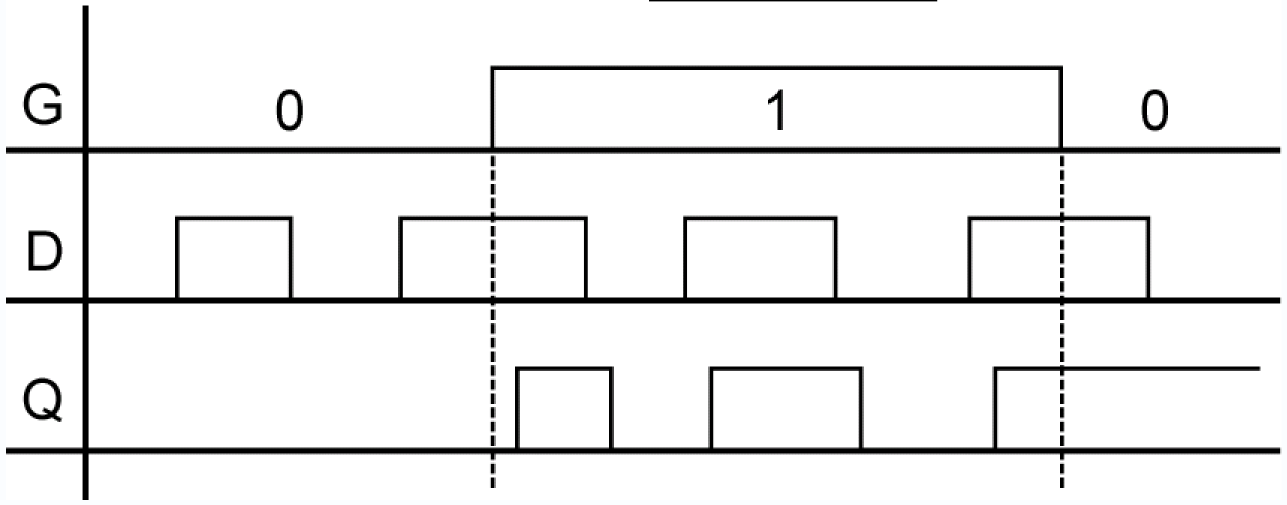
G	D	Q	Q^+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Q is existing state, Q^+ is the next state

“R” =GD’

“S” =GD

Output data **changes** only for these two rows.



G	Q^+
0	mem
1	D

UNIVERSITY OF LIVERPOOL

Flip-flops

Flip-flops:

- We define the flip-flop as being a memory device (e.g. latch) that changes its output state **in response to a clock input.** Clocked circuits prevent response to ‘glitches’.

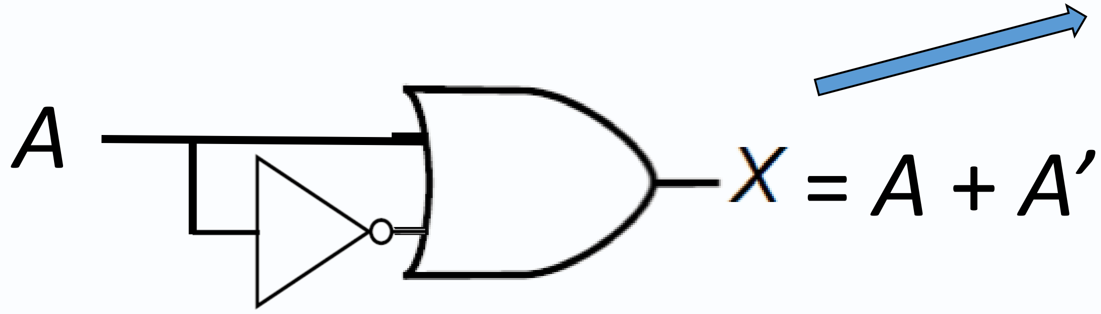
Don’t be confused…:

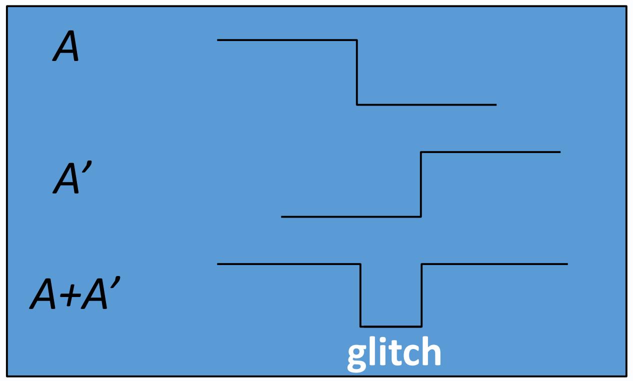
- SR latches are sometimes referred to by other sources as “SR flip-flops”
- In that case, our ‘SR flip-flop’ is referred to as a ‘gated’ or ‘clocked’ SR flip-flop

Why use flip-flops not latches?

- Timing avoids the effect of "glitches"
- Example of a glitch:

Logical outcome: X = TRUE for any A
Example of actual outcome:
A → 0, during inverter propagation delay, A' still 0 so X = FALSE temporarily. This is a **glitch**.





A glitch at D might trigger an unwanted change (latch), but this can be avoided if the clock is in control (flip-flop)

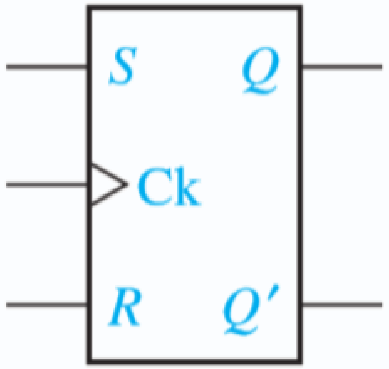
D	Action at D change
0	"RESET", $Q^+ = 0$
1	"SET", $Q^+ = 1$

vs

D	Action at next clock pulse
0	"RESET", $Q^+ = 0$
1	"SET", $Q^+ = 1$

UNIVERSITY OF LIVERPOOL

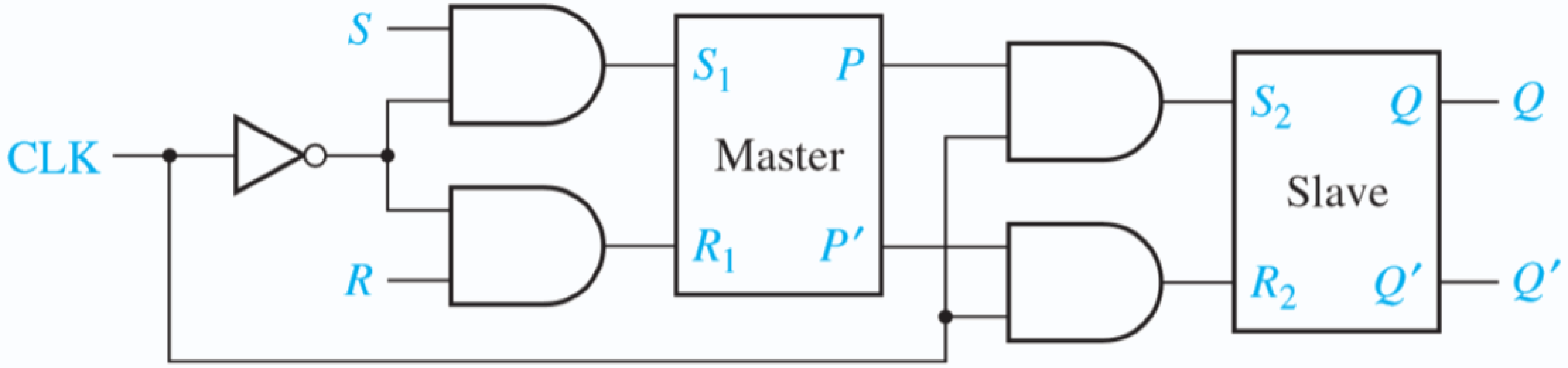
S-R flip-flop



Operation summary:

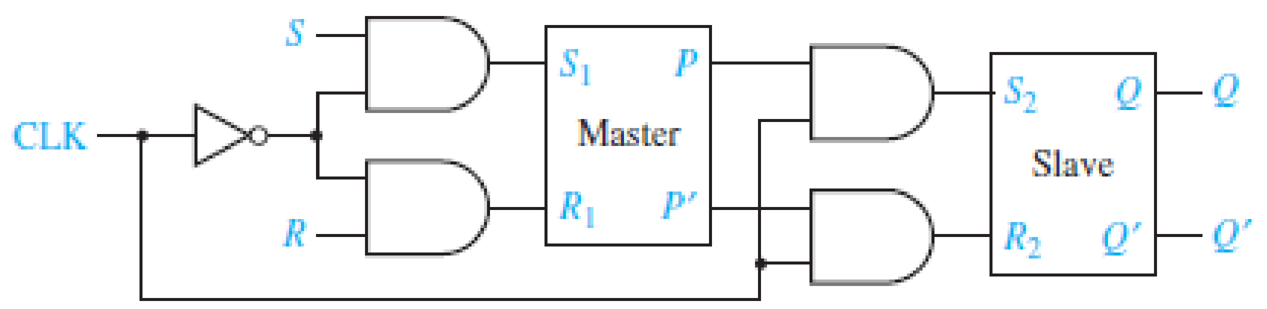
$S = R = 0$	No state change
$S = 1, R = 0$	Set Q to 1 (after active Ck edge)
$S = 0, R = 1$	Reset Q to 0 (after active Ck edge)
$S = R = 1$	Not allowed

This flip-flop changes state after the rising edge of the clock.

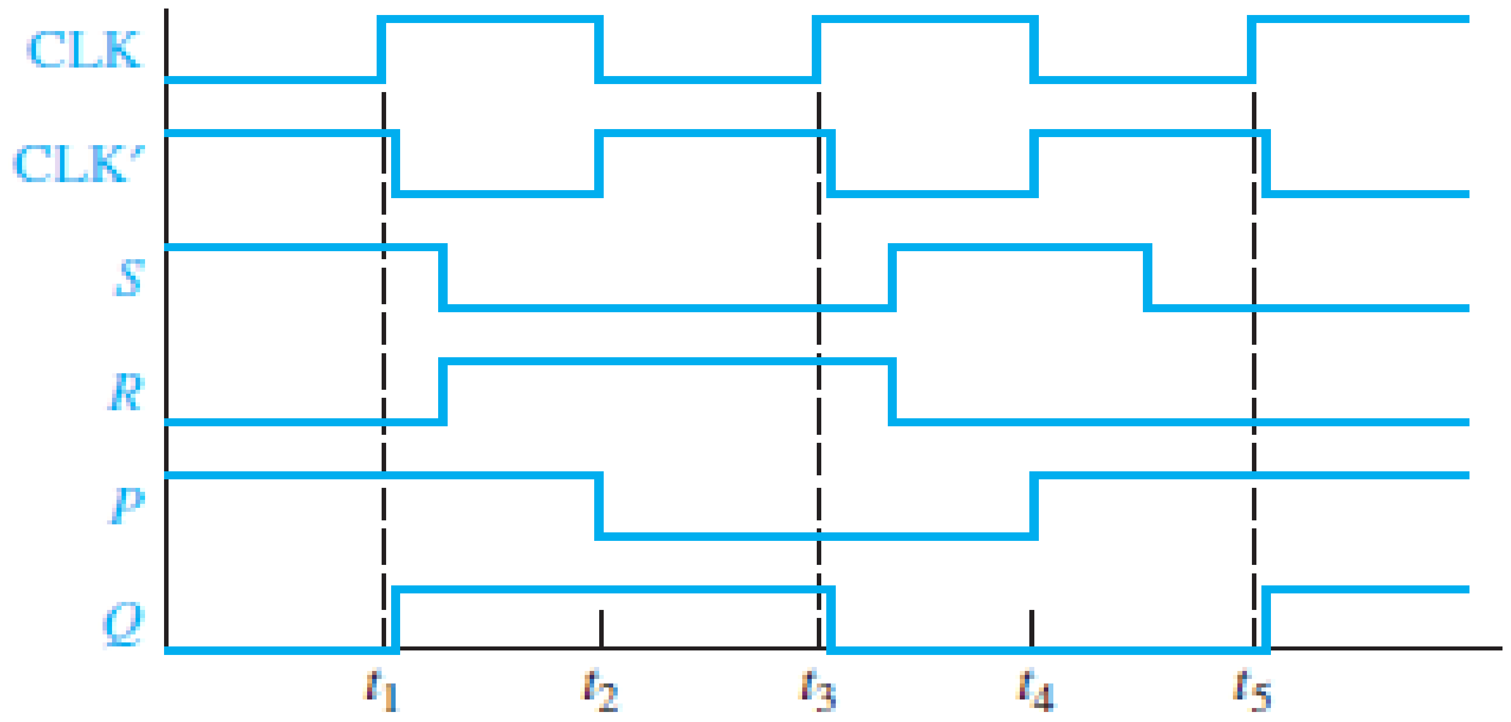


Implementation with two latches

S-R flip-flop timing analysis



S-R flip-flop implemented with two latches



Timing analysis

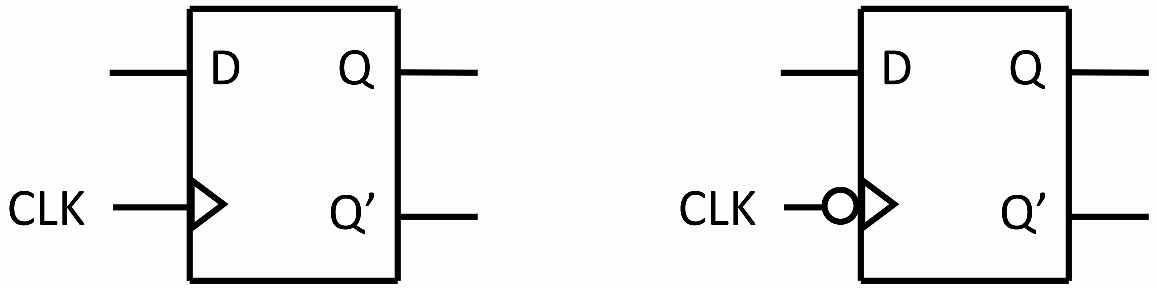
Edge-Triggered D-type Flip-Flop

The **flip-flop** output changes only in response to the clock (CLK), not to a change in D.

(It is still affected by D, but does not respond directly to the change in D.)

Arrowhead > identifies the clock input

Rising-edge:
0 to 1 clock
transition



Falling-edge:
1 to 0 clock
transition

rising-edge-trigger *falling-edge trigger*

An inversion bubble on the clock input indicates a falling-edge trigger.

The “active edge” refers to the clock edge that triggers the change (i.e. rising or falling)

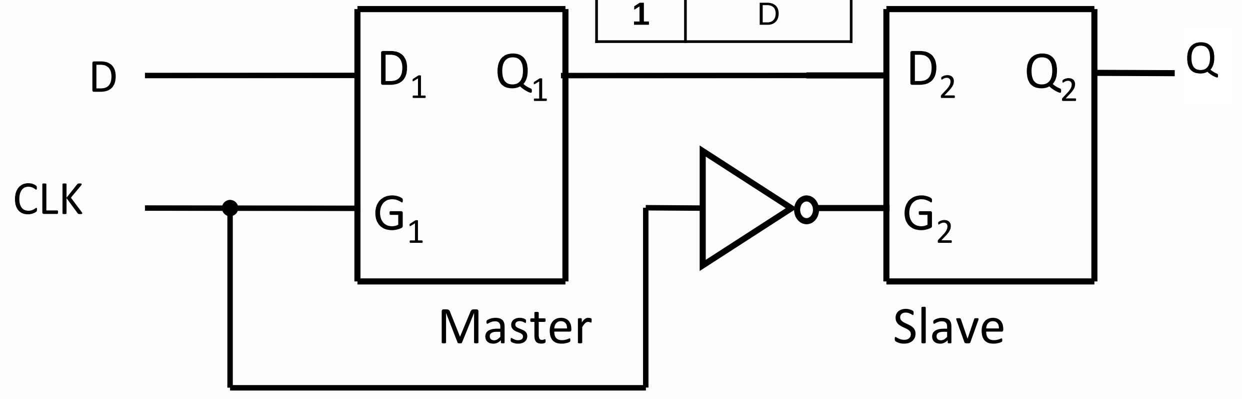
Edge-Triggered D-type Flip-Flop

A *falling-edge triggered* D flip-flop can be constructed from two gated D latches and an inverter – Master/Slave.

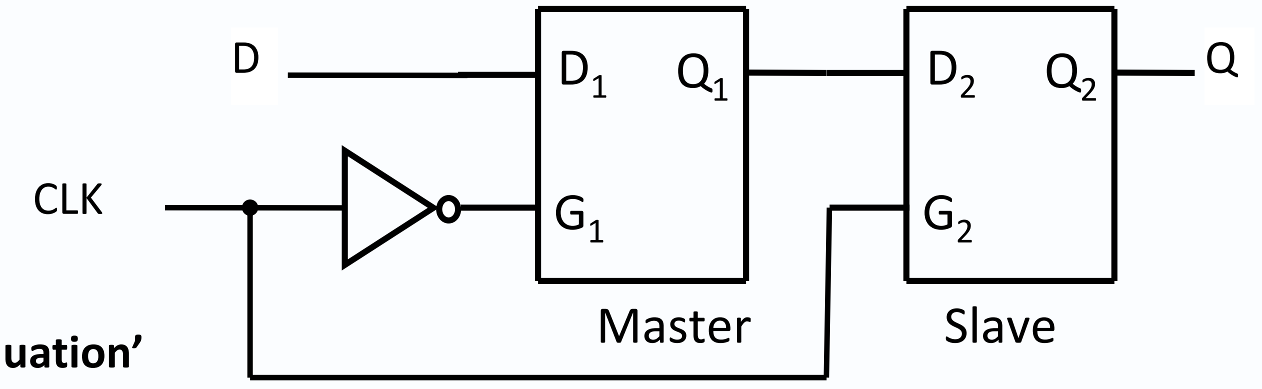
<u>CLK 1→0:</u>
$G_1 \rightarrow 0$ so $D_2 = Q_1$ = mem
Before the edge, $G_1 = 1$ so Q_1 was D.
So now Q_1 and D_2 store this.

$G_2 \rightarrow 1$ so $Q = Q_2 = D_2$ = mem so:
$Q^+ = D$, meaning Q^+ = whatever D was AT THE TIME OF the clock edge.

Each latch	
G_n	Q_n^+
0	mem
1	D



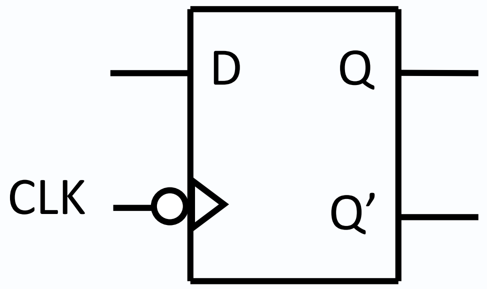
A *rising-edge-triggered* D flip-flop

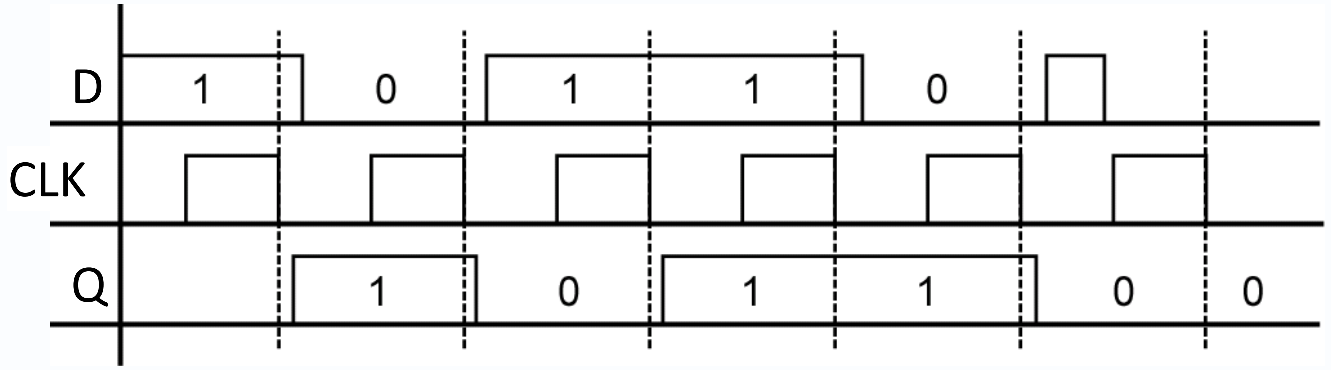


CLK 0→1:
Same applies: **$Q^+ = D$**

$Q^+ = D$ is the 'characteristic equation'

Edge-Triggered D-type Flip-Flop





The operation can be summarised in an action table.

D	Action at next clock pulse
0	“RESET”, $Q^+ = 0$
1	“SET”, $Q^+ = 1$

The output after the clock pulse, Q^+, is equal to the D input at the time of the clock pulse.

Setup and Hold Times for an Edge-Triggered D Flip-Flop

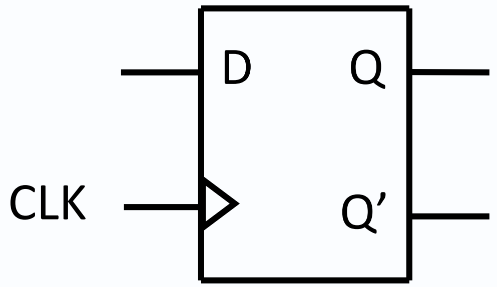
The **propagation delay** of a flip-flop ($\boldsymbol{t_p}$) is the time between the active edge of the clock and the resulting change in the output. ‘Flip-flop delay’.

<u>The D input also creates timing issues:</u>

Amount of time that D must be stable **before** the active edge, is called the **setup time ($\boldsymbol{t_{su}}$).**

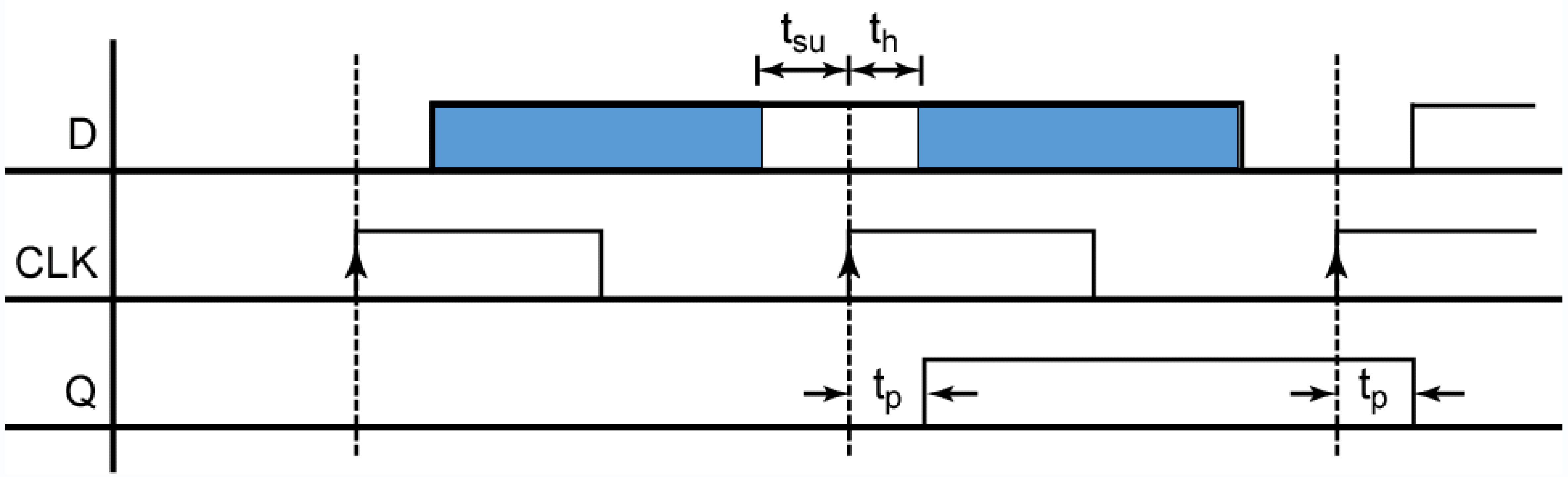
The amount of time that D must hold the same value **after** the active edge, is called the **hold time ($\boldsymbol{t_h}$).**

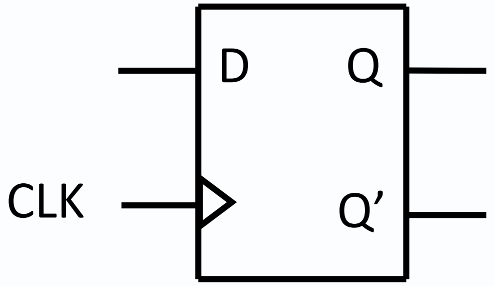
If D changes in the prohibited time interval, i.e. ‘at or too close to the time of the active edge’, the output is unpredictable.



UNIVERSITY OF LIVERPOOL

Setup and Hold Times for a Rising-Edge-Triggered D Flip-Flop

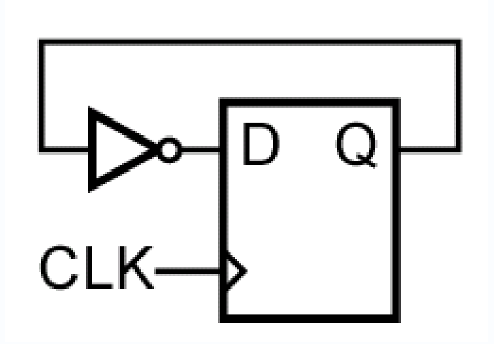


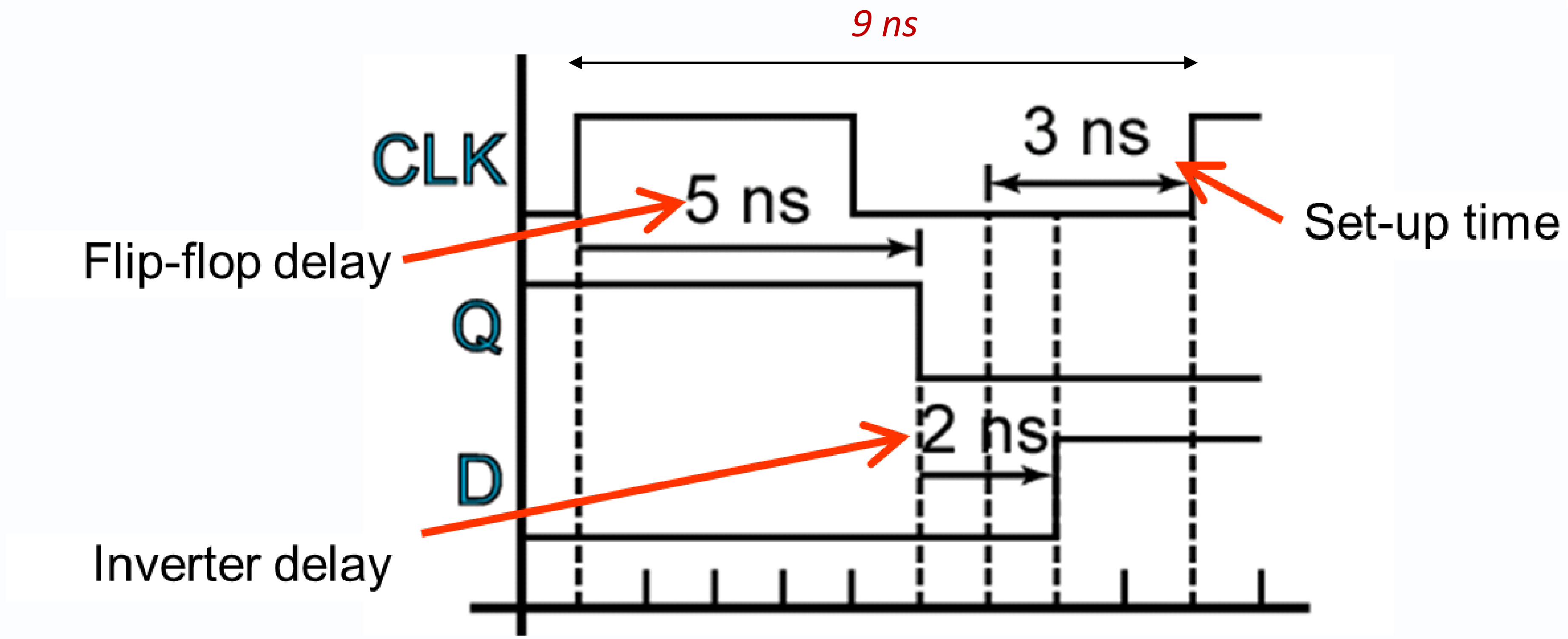


D only allowed to change in the shaded regions (not during t_{su} or t_h)

Determination of Minimum Clock Period

Suppose the inverter propagation delay is *2 ns*
the FF propagation delay is *5 ns* ,
and its setup time is *3 ns*.
(The hold time does not affect this calculation)

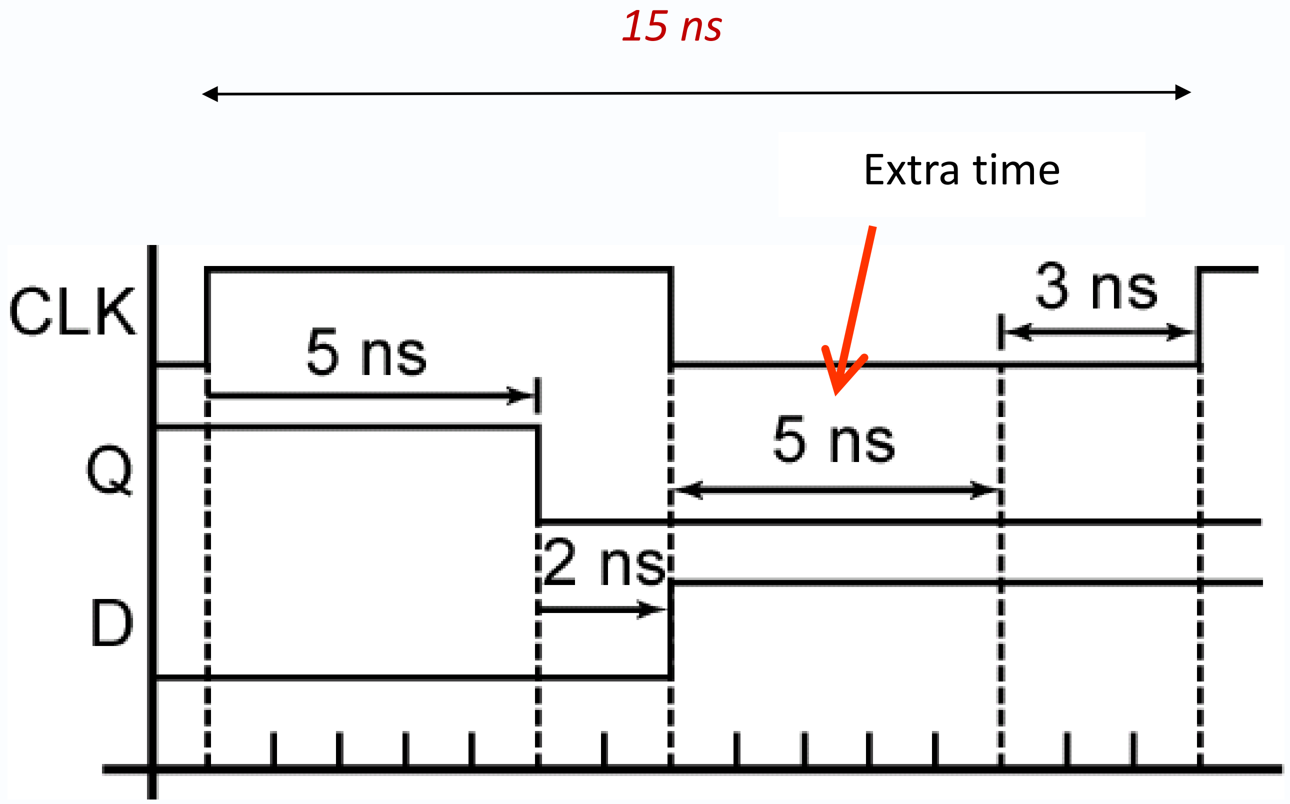




D is not in its new state long enough before the next rising edge (3ns set-up time required)

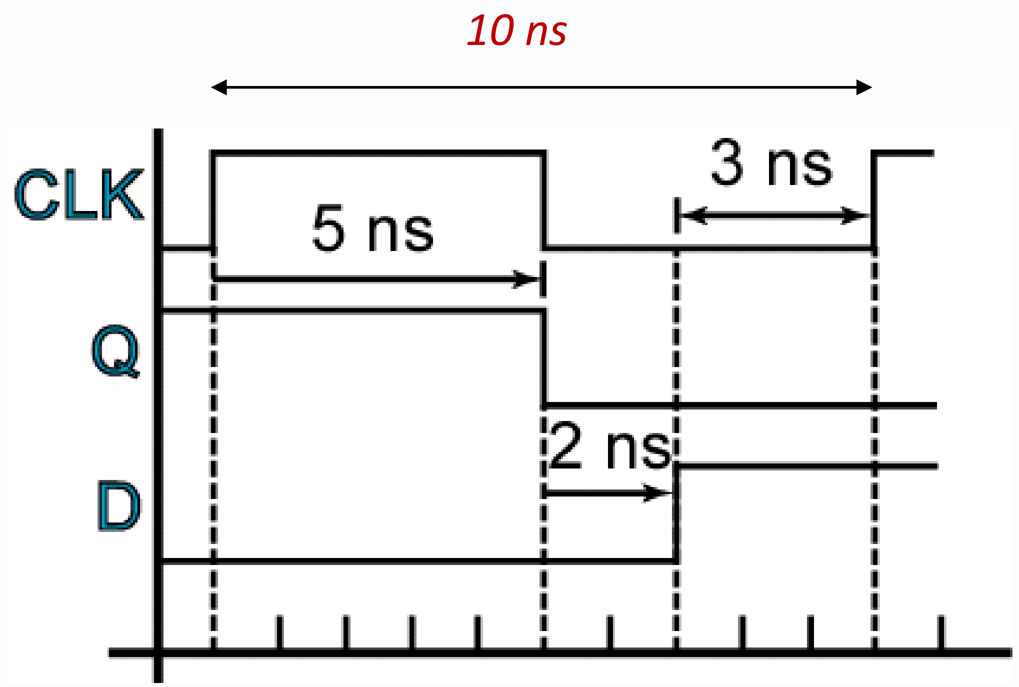
9 ns **is not enough for the clock period**

Determination of Minimum Clock Period



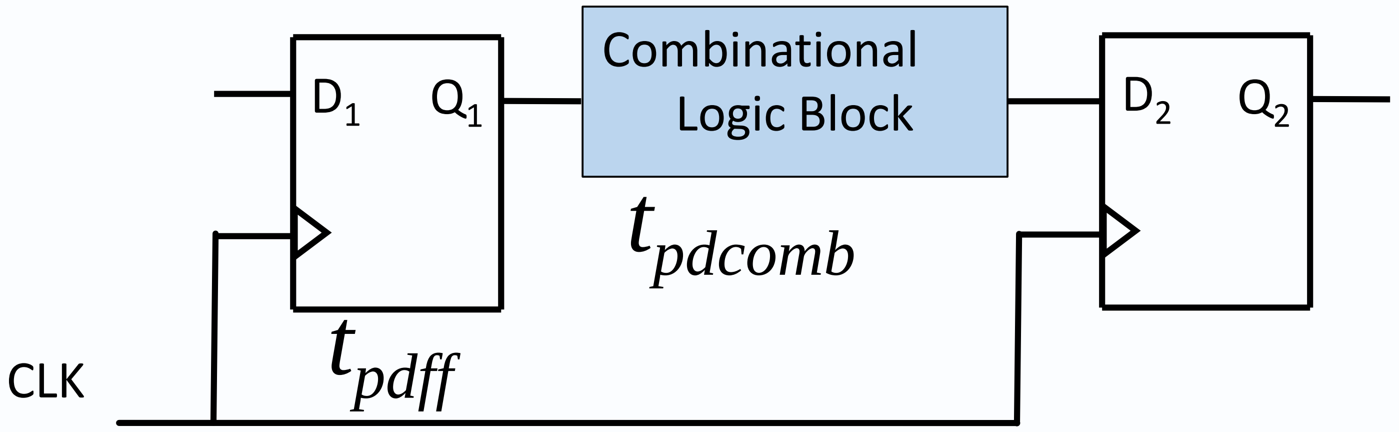
15 ns is more than enough for the clock period.

Determination of Minimum Clock Period



10 ns is just enough for the clock period.

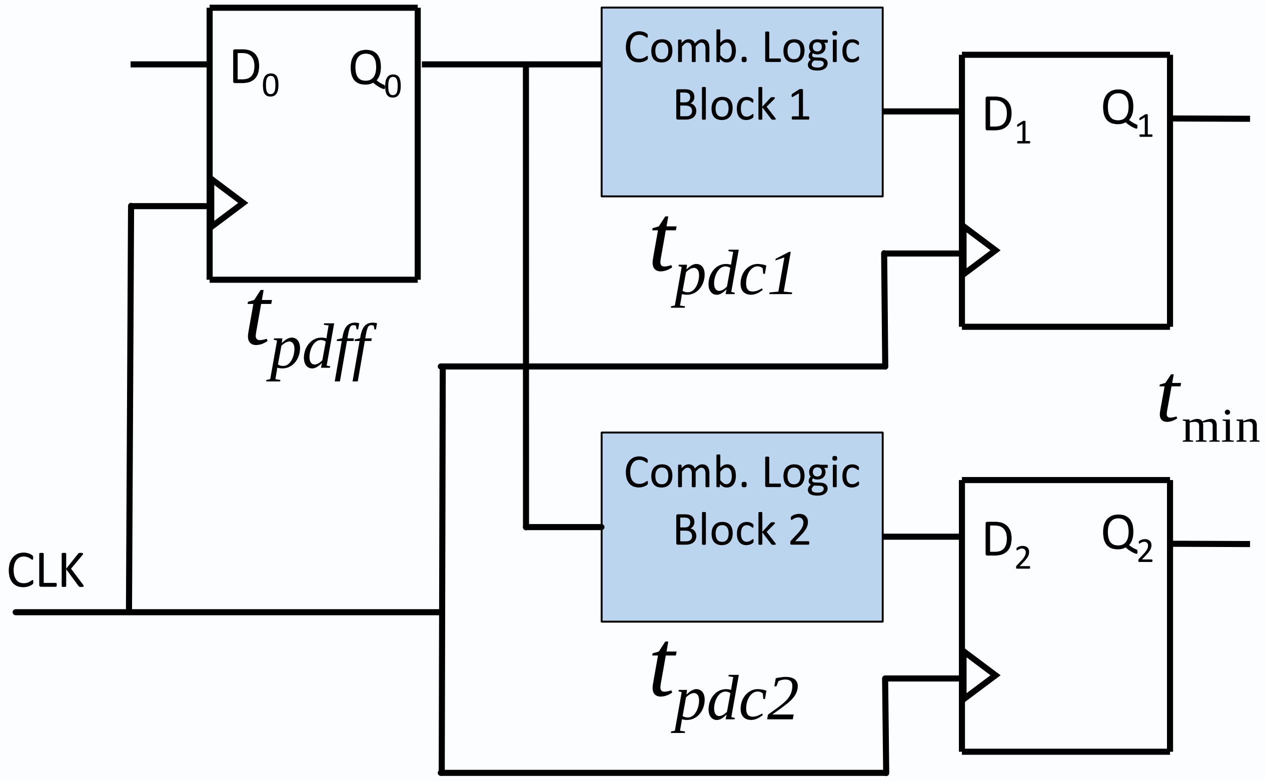
Min. Clock Period in 2 Flip-flop System



$$t_{min} = t_{pdff} + t_{pdcomb} + t_{su}$$

Set-up time and hold time are **not delays**. They are just timing constraints for the correct operation of flip-flops. The minimum clock period in a system does not depend on the hold time of flip flops.

Min. Clock Period in a complex circuit



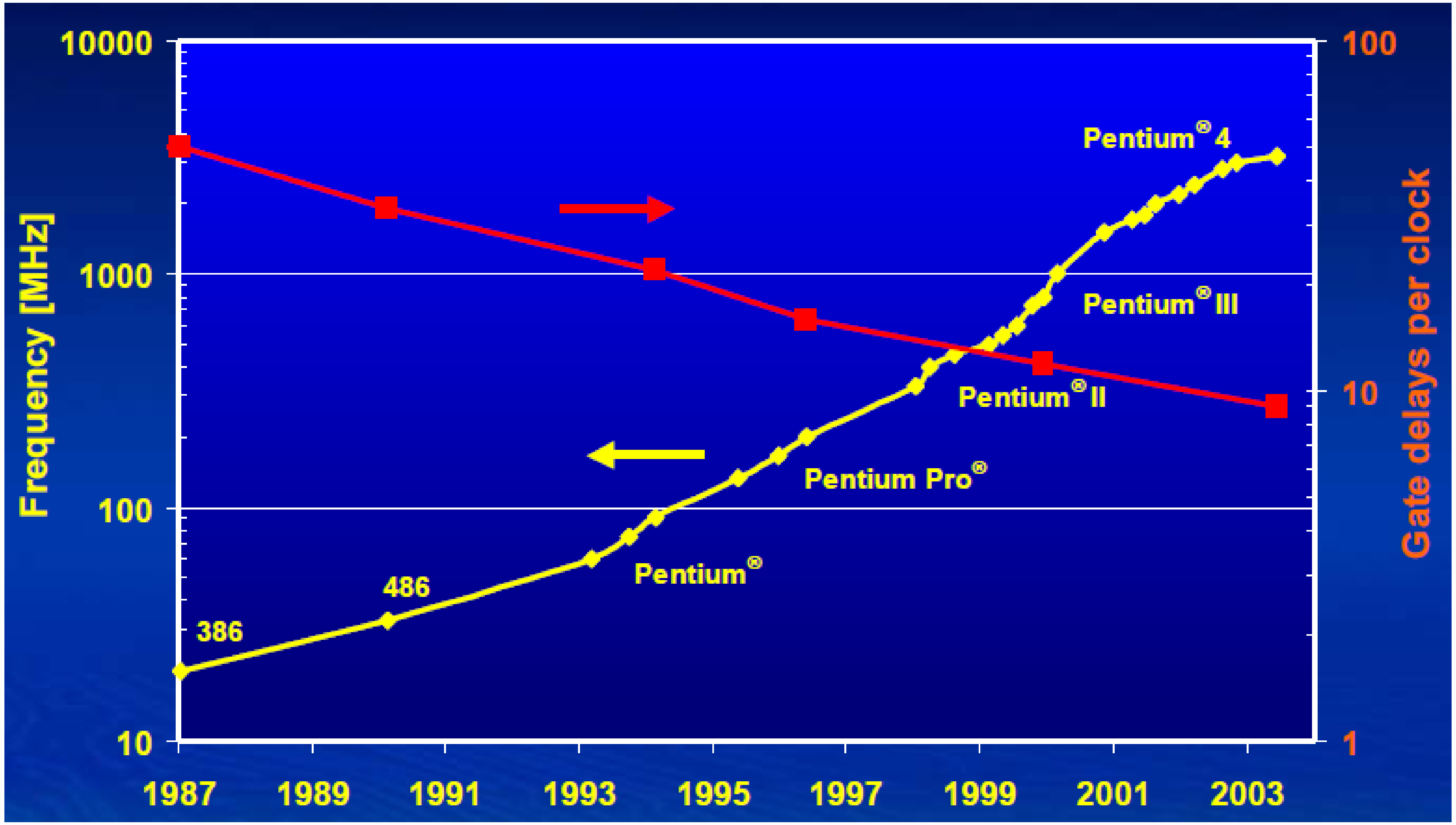
$$t_{min1} = t_{pdff} + t_{pdc1} + t_{su}$$

$$t_{\min} = \max\left(t_{\min1}, t_{\min2}\right)$$

$$t_{min2} = t_{pdff} + t_{pdc2} + t_{su}$$

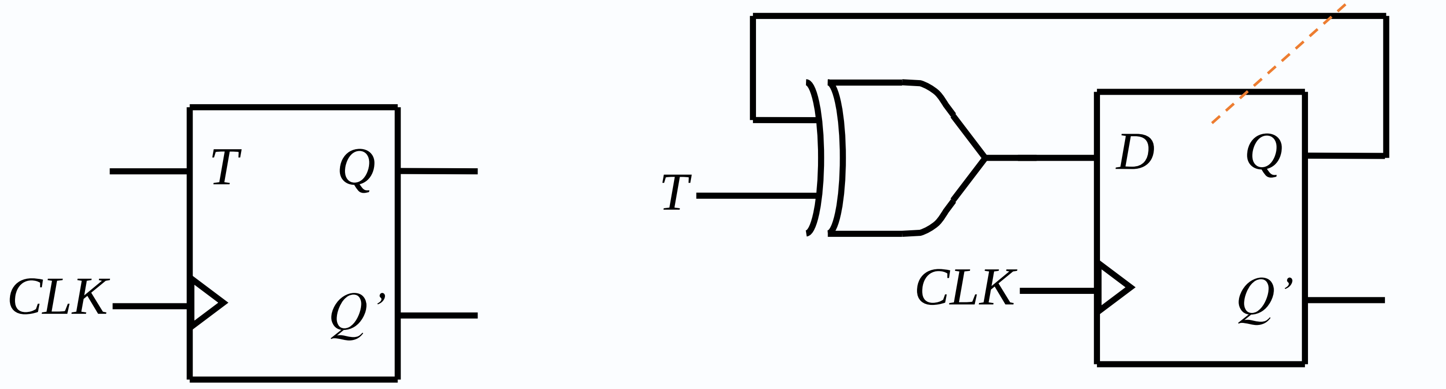
In a circuit with one clock & many flip-flops & combinational circuits, the longest propagation delay of the combinational circuits determines the minimum required clock period.

Clock frequency and gate delay



From Intel presentation, 2004.

T or Toggle Flip-Flop



(Edge-triggered D flip-flop:)

D	Action at next clock pulse
0	$Q^{+}=0$
1	$Q^{+}=1$

When T=1 the flip-flop changes state after the active edge of the clock.

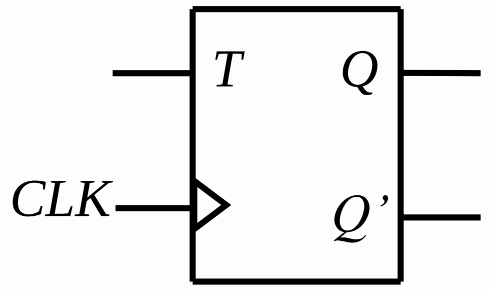
When T=0 no state change occurs.

From the truth table:

T	Q	Q^{+}
0	0	0
0	1	1
1	0	1
1	1	0

$$Q^{+} = D = \overline{T}Q + T\overline{Q} = T \oplus Q$$

T or Toggle Flip-Flop



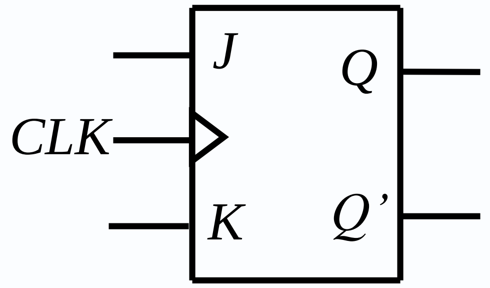
T	Q	Q^+
0	0	0
0	1	1
1	0	1
1	1	0

T	Action at next clock pulse
0	No change, $Q^+ = Q$
1	Toggle (change), $Q^+ = Q'$

Toggle flip-flop often used in building counters.
Most CPLDs and FPGAs can be programmed to implement toggle flip-flops.

J-K Flip-Flop

J	K	Q	Q^+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



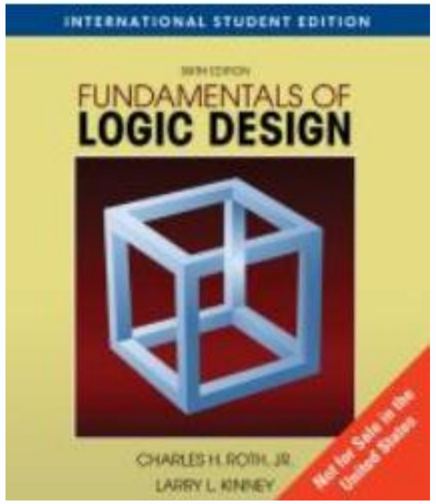
J	K	Action at next clock pulse
0	0	No change, $Q^+ = Q$
0	1	Reset, $Q^+ = 0$
1	0	Set, $Q^+ = 1$
1	1	Toggle, $Q^+ = Q'$

Extended version of the S-R flip-flop … simultaneous "1" allowed
J is like S, K like R.

Summary and suggested reading

- Gate delays, timing diagrams (Section 8.3)
- Set-Reset latch (Section 11.2)
- Gated D-latch (Section 11.3)
- Edge-triggered D flip-flop (Section 11.4)
- S-R, JK and T flip-flops (Sections 11.5-7)

Roth and Kinney *Fundamentals of Logic Design*



UNIVERSITY OF LIVERPOOL