

Memory Elements I

CS31

Pascal Van Hentenryck



Memory Elements (I)

Combinational devices are good for

- computing Boolean functions
- pocket calculator

Computers also need to remember things

- memory elements

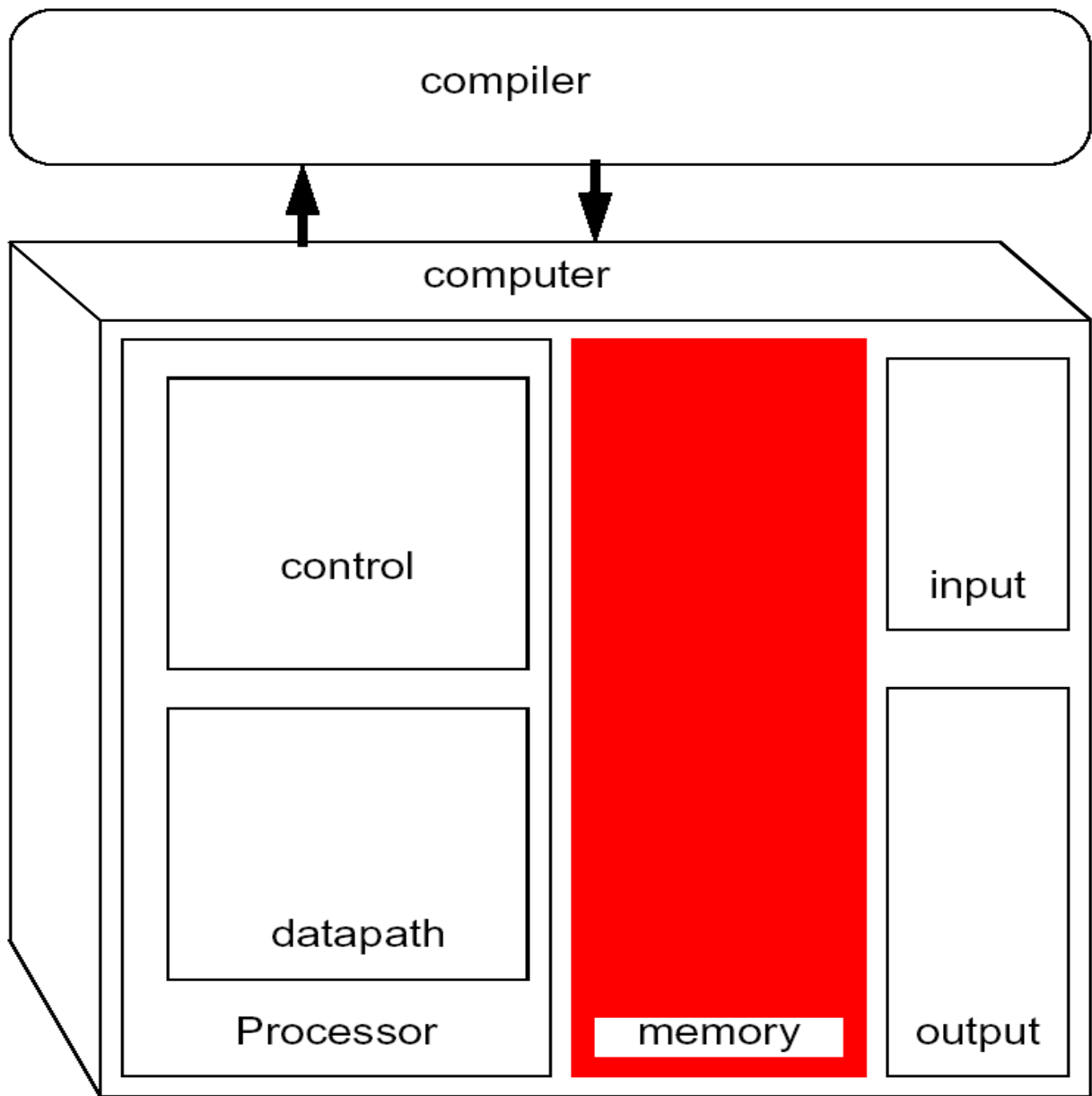
Today's topic

- how to make computers remember things

Overview

- latches
- transparent latches
- flip-flops

The Big Picture



Abstraction Hierarchy

Programming Language

Assembly Language

Machine Language

Sequential Circuit

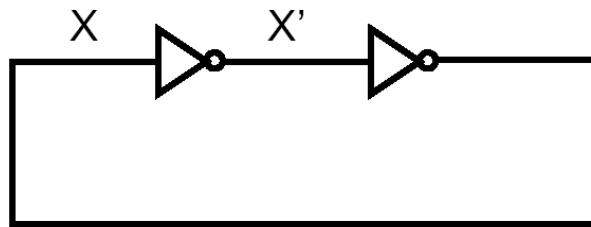
Combinational Circuit

Binary Value

Voltage

Memory Elements

How to remember things with gates and wires?



Two stable states

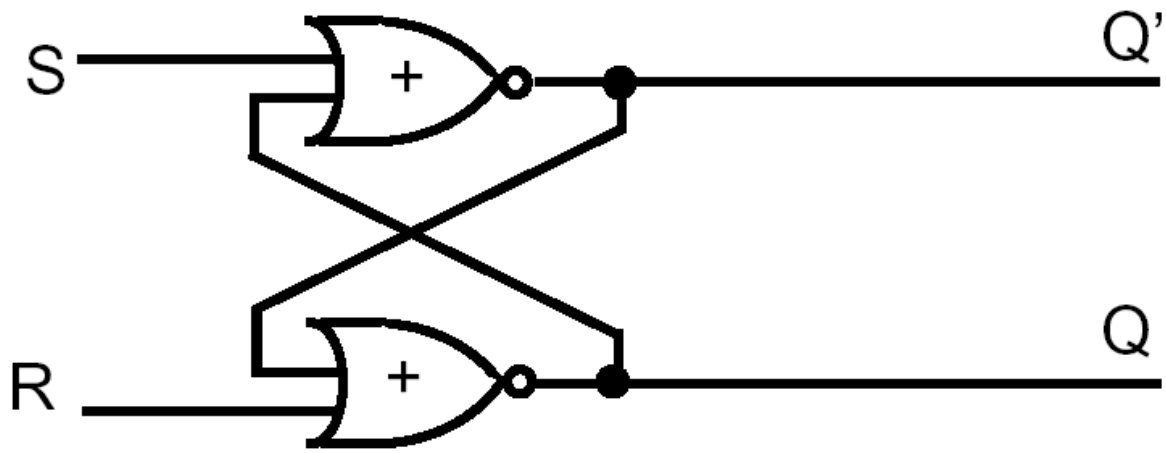
- $X = 1$ and $X' = 0$
- $X = 0$ and $X' = 1$

This circuit can store one-bit of information

What do we need now?

- How to assign a state?
- How to change the state?

R-S Latch

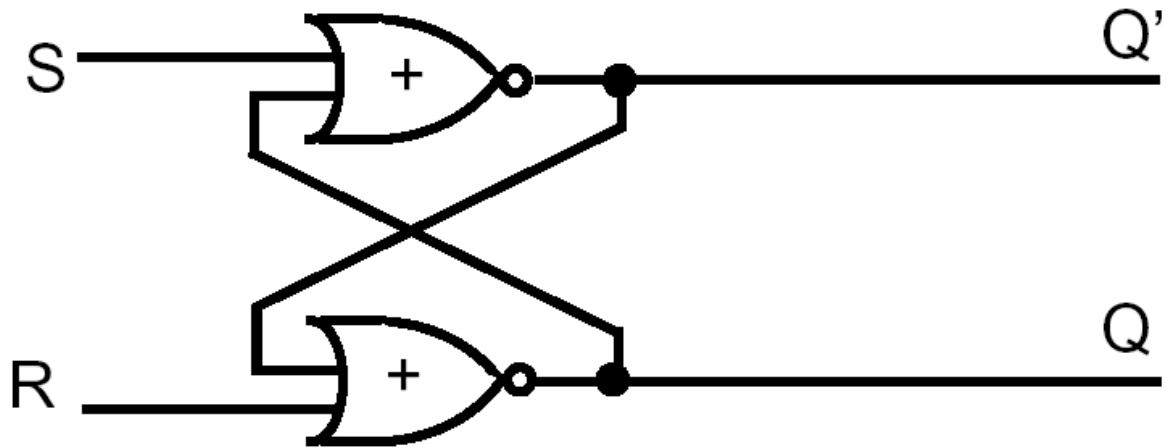


A word is



Each bit is represented by a latch

R-S Latch

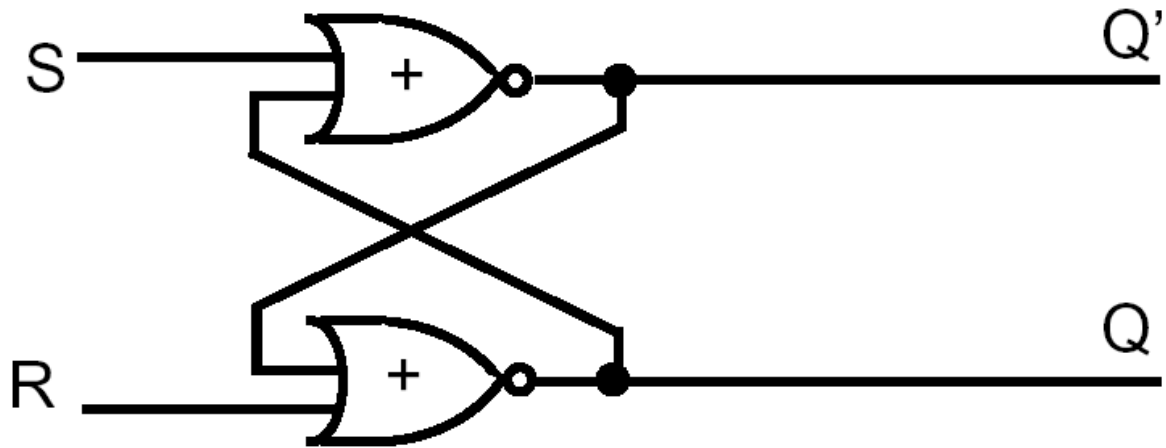


Initial State: [$Q = 1$ and $Q' = 0$]

What if Q Q'

$S=0$ & $R=1$

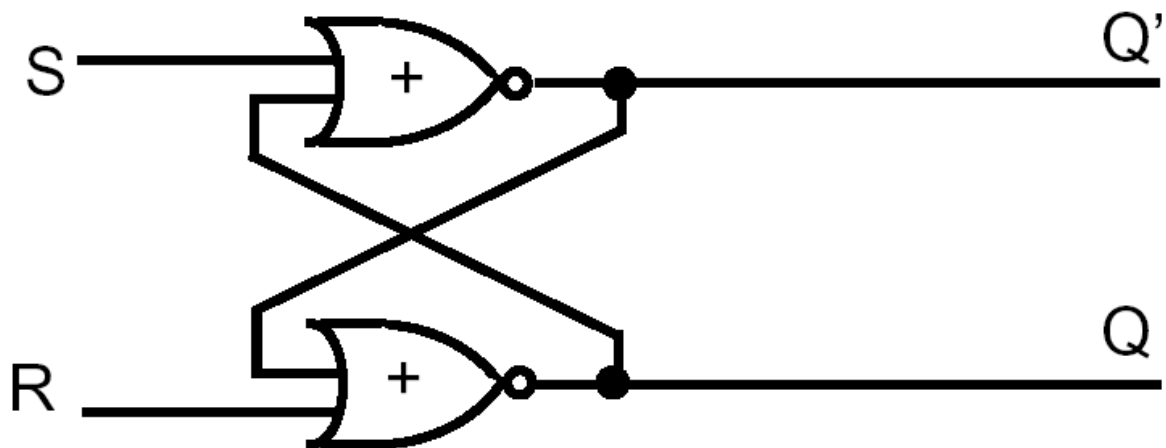
R-S Latch



Initial State: [$Q = 1$ and $Q' = 0$]

What if	Q	Q'
S=0 & R=1	0	1

R-S Latch

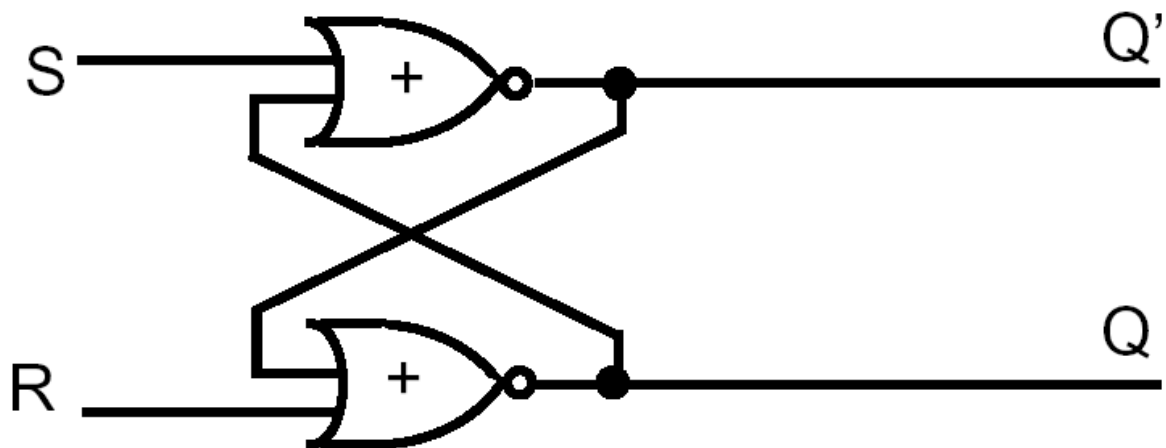


Initial State: [$Q = 1$ and $Q' = 0$]

What if Q Q'

$S=1$ & $R=0$

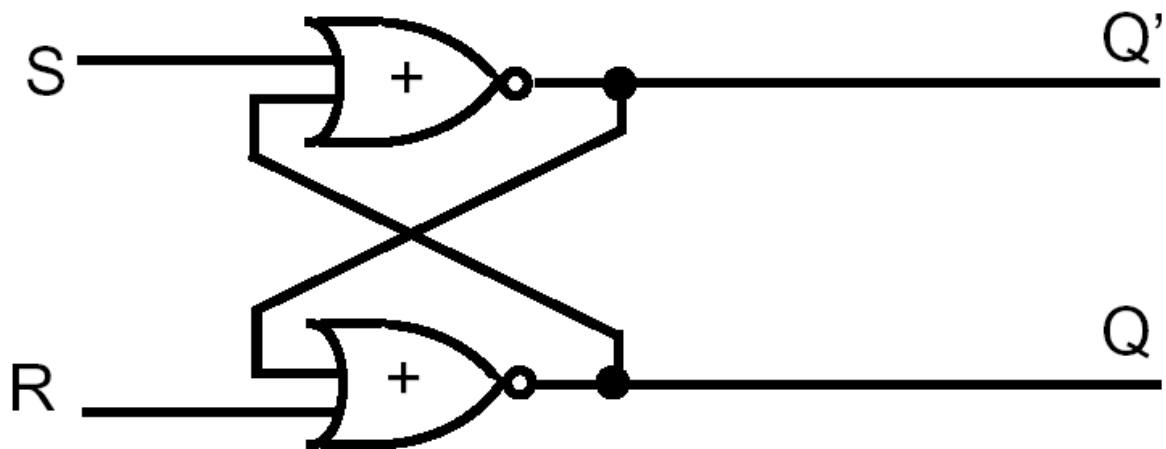
R-S Latch



Initial State: [$Q = 1$ and $Q' = 0$]

What if	Q	Q'
S=1 & R=0	1	0

R-S Latch

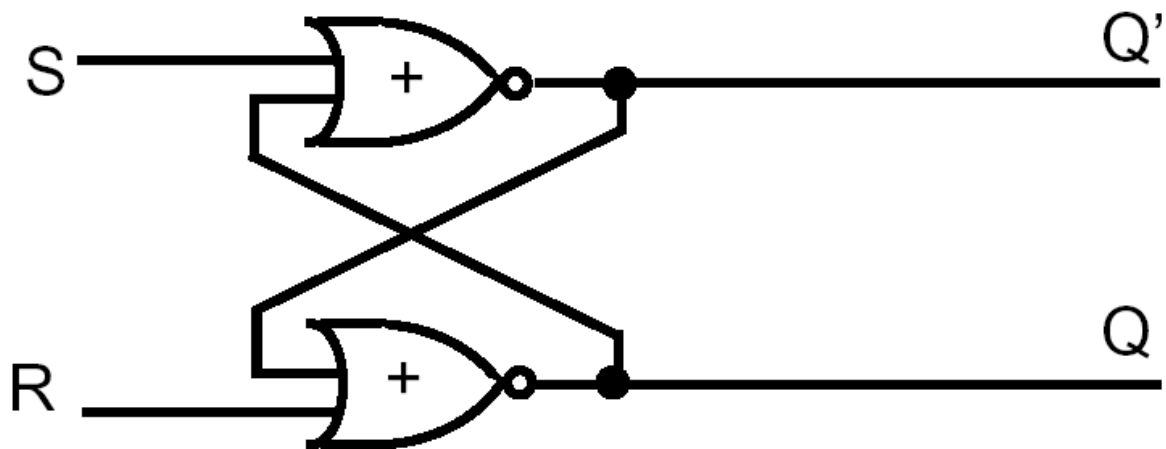


Initial State: $Q = 0$ and $Q' = 1$

What if Q Q'

$S=1$ & $R=0$

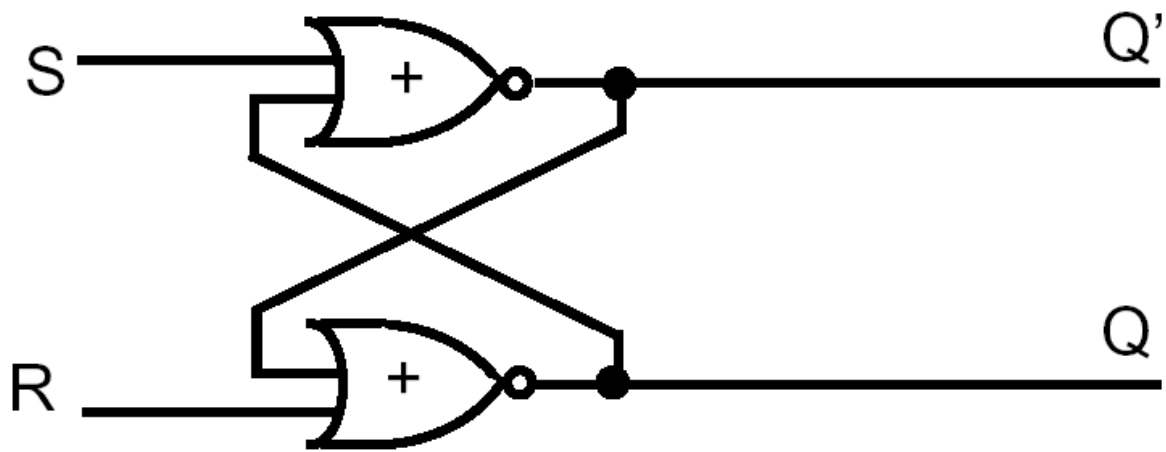
R-S Latch



Initial State: $Q = 0$ and $Q' = 1$

What if	Q	Q'
S=1 & R=0	1	0

R-S Latch

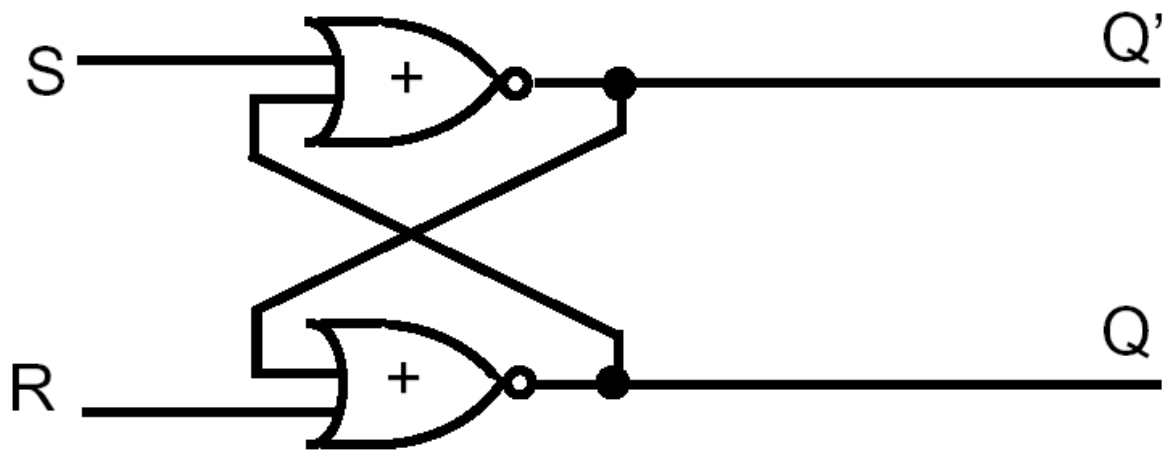


Initial State: $Q = 0$ and $Q' = 1$

What if Q Q'

$S=1$ & $R=1$

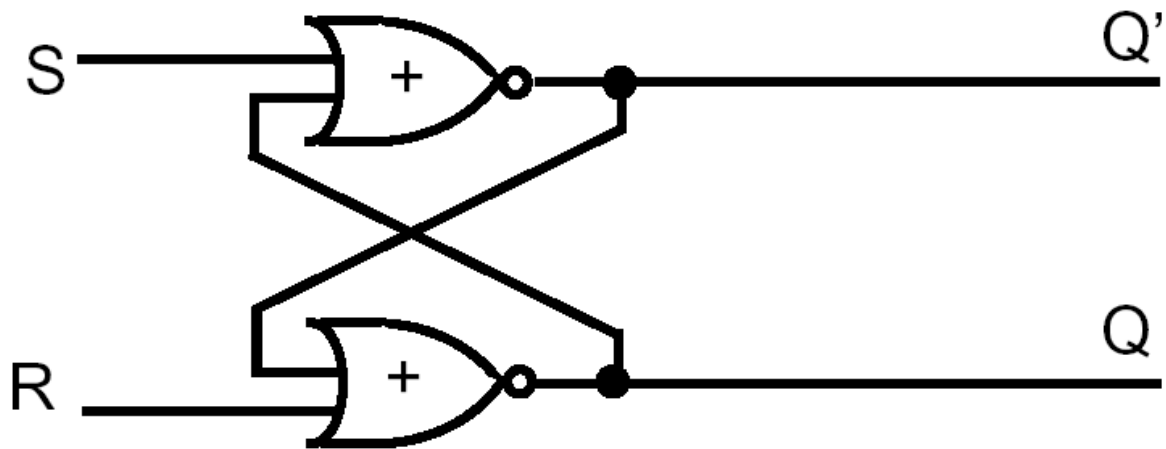
R-S Latch



Initial State: $Q = 0$ and $Q' = 1$

What if	Q	Q'
S=1 & R=1	0	0

R-S Latch

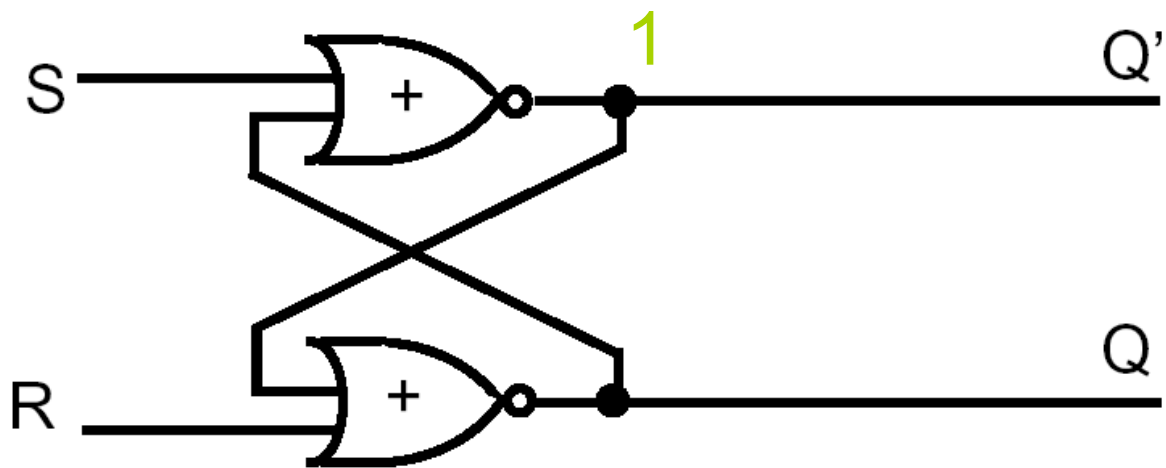


Initial State: $Q = 0$ and $Q' = 0$

What if Q Q'

$S=0$ & $R=0$

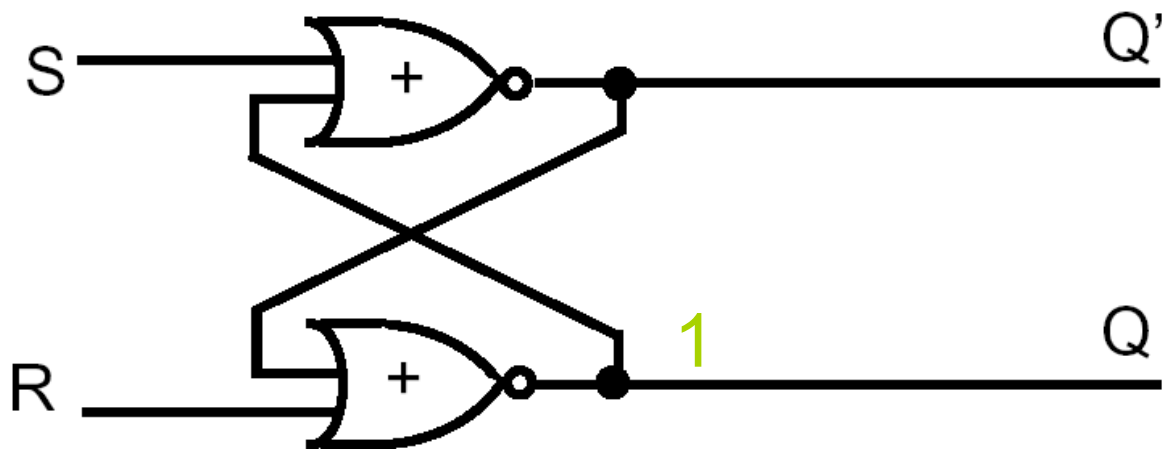
R-S Latch



Initial State: $Q = 0$ and $Q' = 0$

What if	Q	Q'
S=0 & R=0 (S 1 st)	0	1

R-S Latch

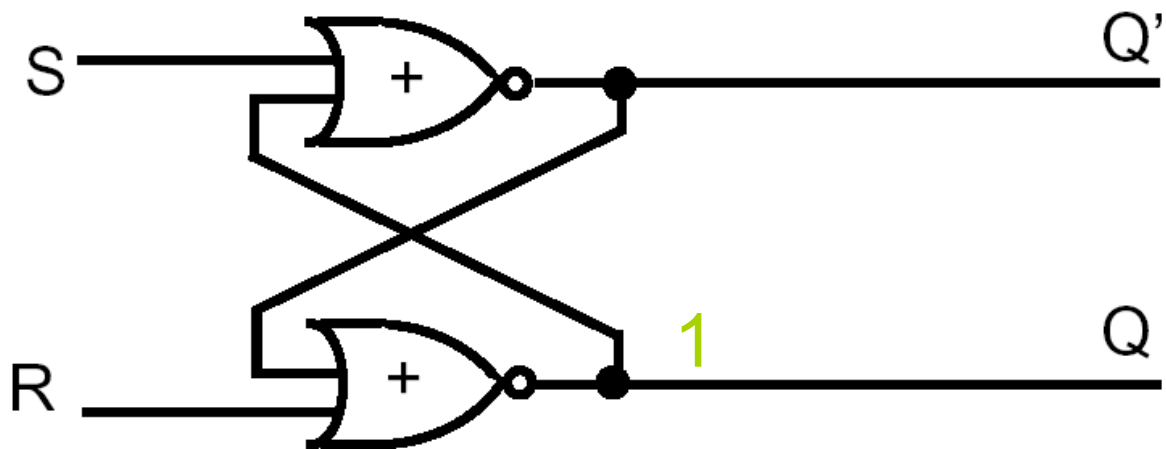


Initial State: $Q = 0$ and $Q' = 0$

What if Q Q'

$S=0$ & $R=0$ (R 1st)

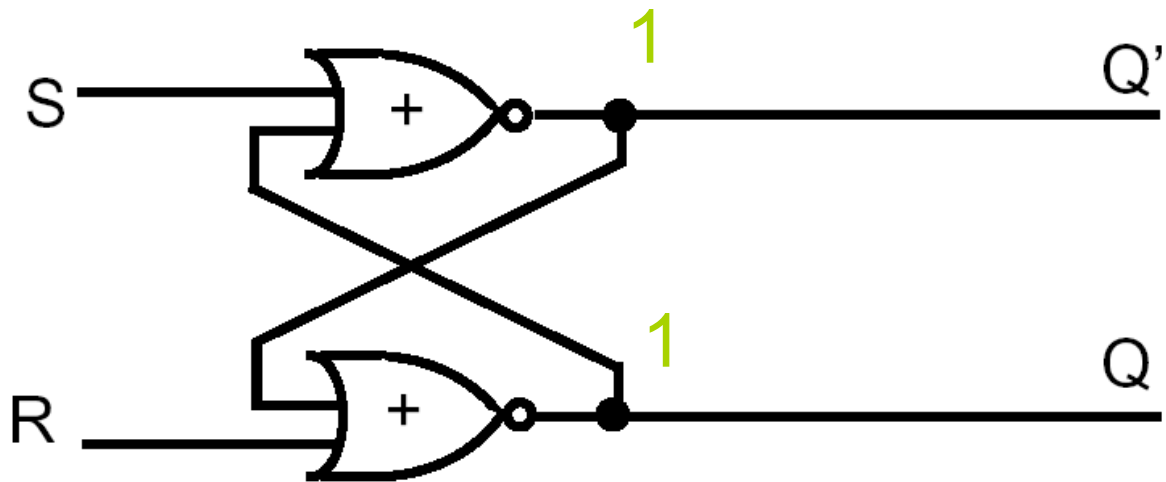
R-S Latch



Initial State: $Q = 0$ and $Q' = 0$

What if	Q	Q'
$S=0$ & $R=0$ (R 1 st)	1	0

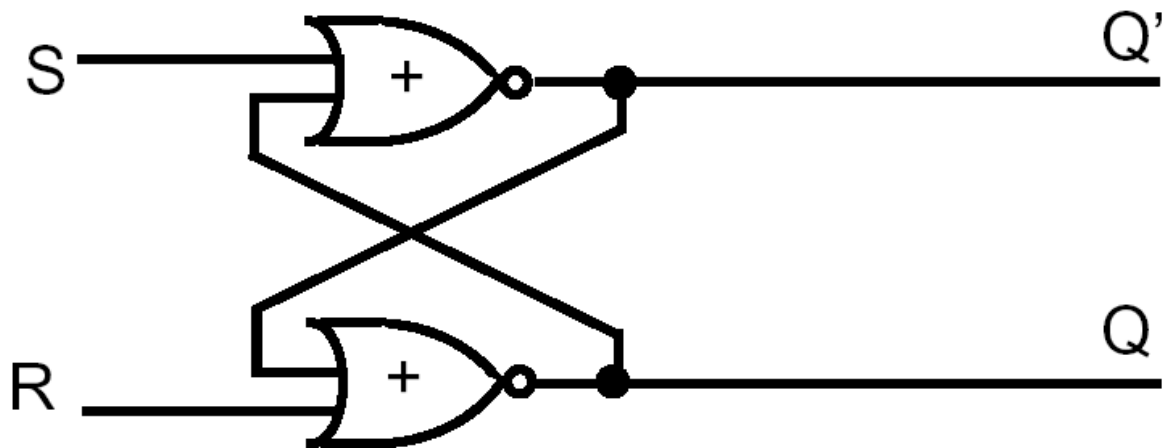
R-S Latch



Initial State: $Q = 0$ and $Q' = 0$

What if	Q	Q'
S=0 & R=0 (same time)	?	?

Problems with Latches



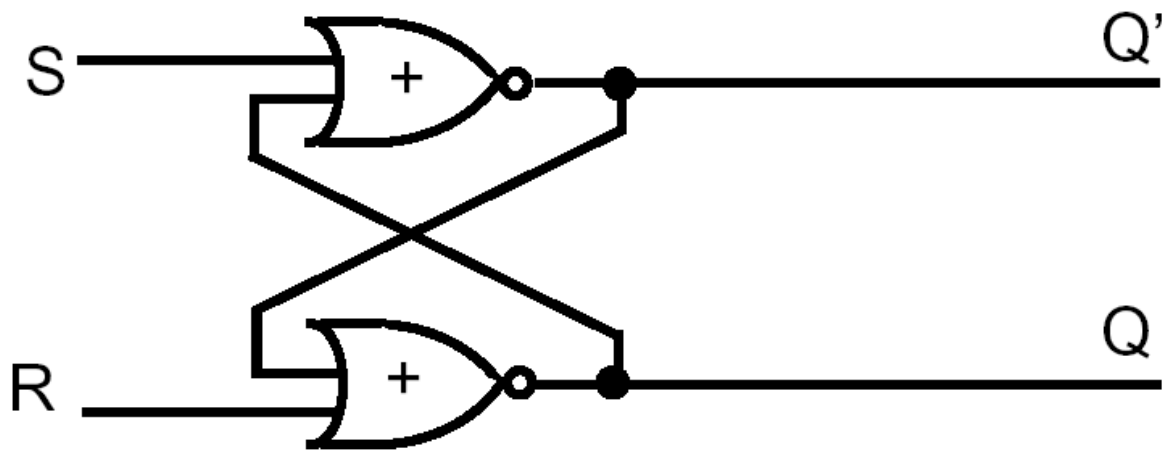
Race conditions

- unpredictability of results

Possibilities

- $Q'=0$ and $Q=1$
- $Q'=1$ and $Q=0$
- Instability

R-S Latch



What does all this mean?

- We have to worry about timing
- This is the last thing we want to do

Clocks

Circuits with feedback loops are difficult to control

- Need to worry about timing

Main abstraction in circuit design

- Ignoring propagation time

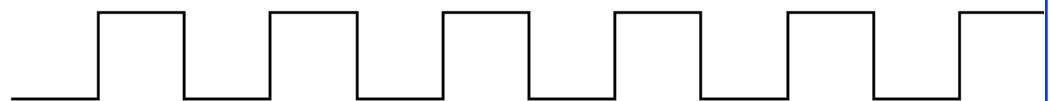
Basic solution

- Using a clock

What is a clock?

- It is a free running signal

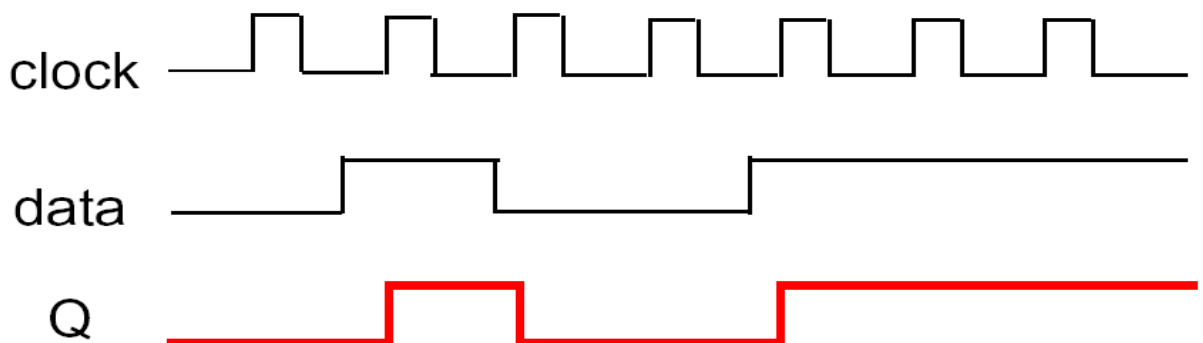
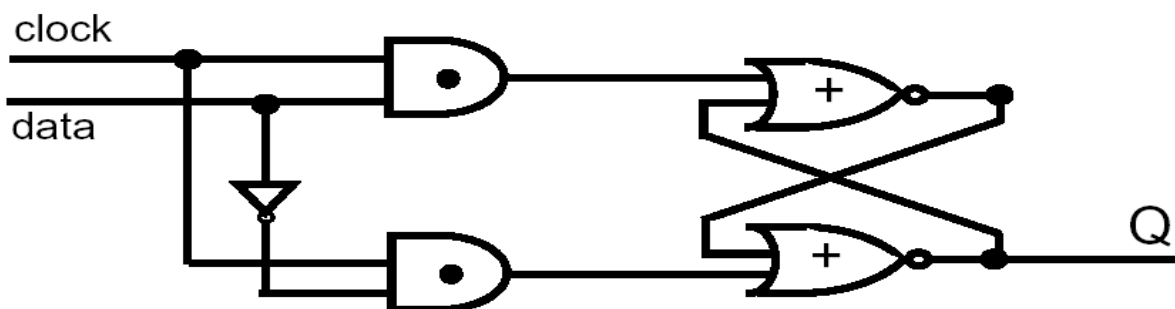
Clock



Transparent Latches

How to avoid timing problems?

- state changes only occur when the clock is asserted
- transparent latch that enforces $Q = \text{Data}$



Shifting

0	1	0	0
---	---	---	---



Shift right

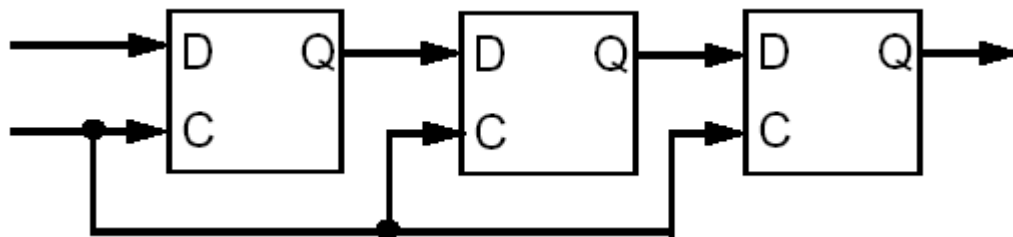
0	0	1	0
---	---	---	---

What does shifting right do to a number?

Divides the number by 2.

Look Ma, my latch is leaking

The results are still unpredictable



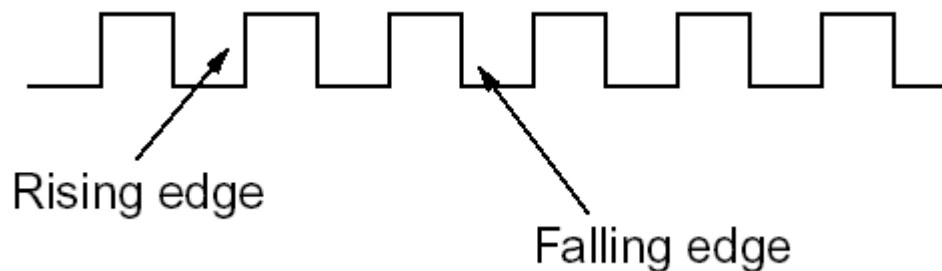
The value of the first latch can be assigned to the second, the third, or any of the connected latches depending on propagation times and the clock cycle.

Annoying to build shifting registers

How to remedy this problem?

Edge-Triggered Methodology

State Changes only occur on a clock edge



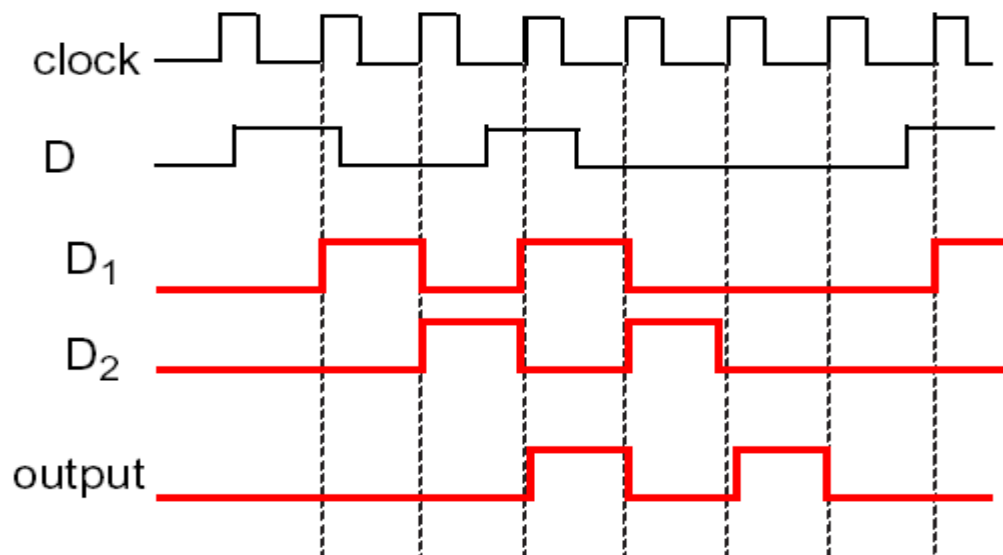
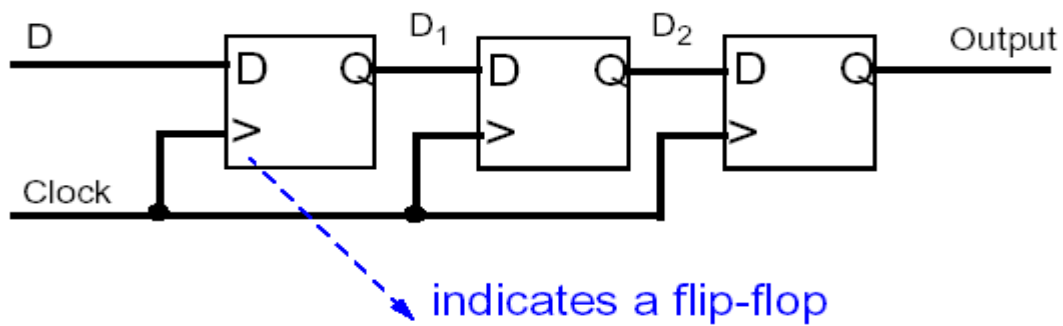
Flip-flops

- latches which are only updated on a clock edge
- bits in words are flip-flops

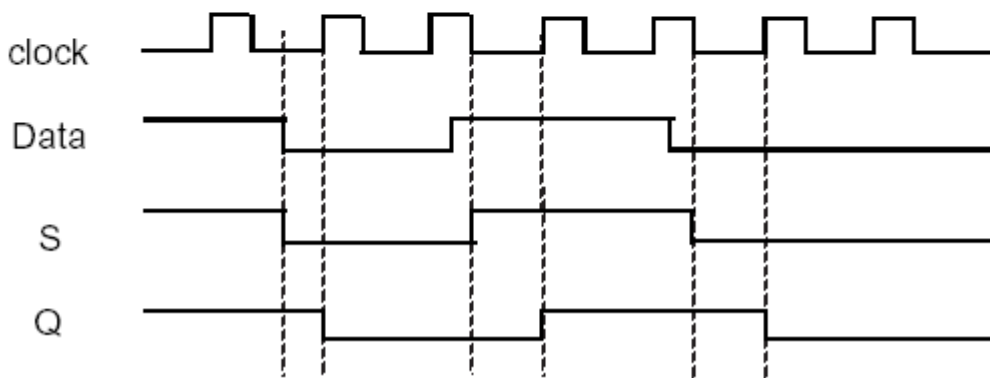
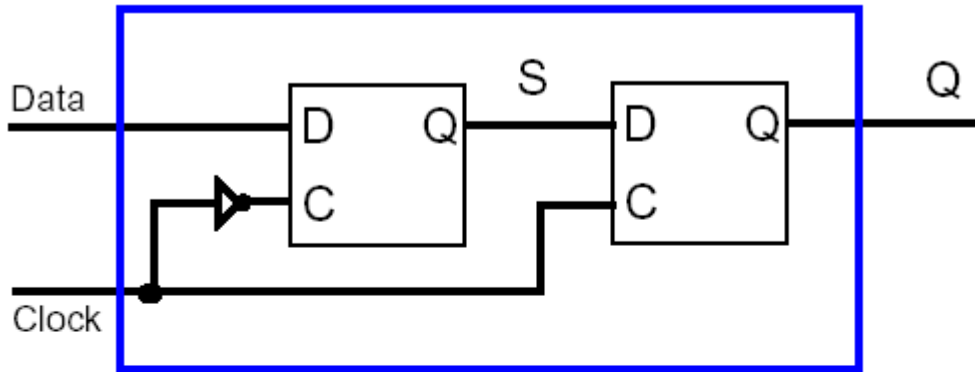


Flip-Flops

Shift Register Again



Edge-Triggered Flip-Flop

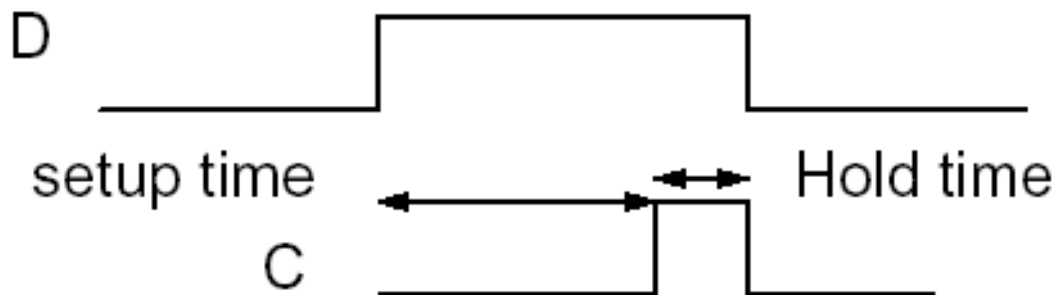


- S can only change when $C = 0$
- Q can only change when $C = 1$
- Q changes on next *rising* clock edge after Data changes

A flip-flop is made of two latches!

Edge-Triggered Flip-Flops

Basic constraint (rising edge flip-flop)

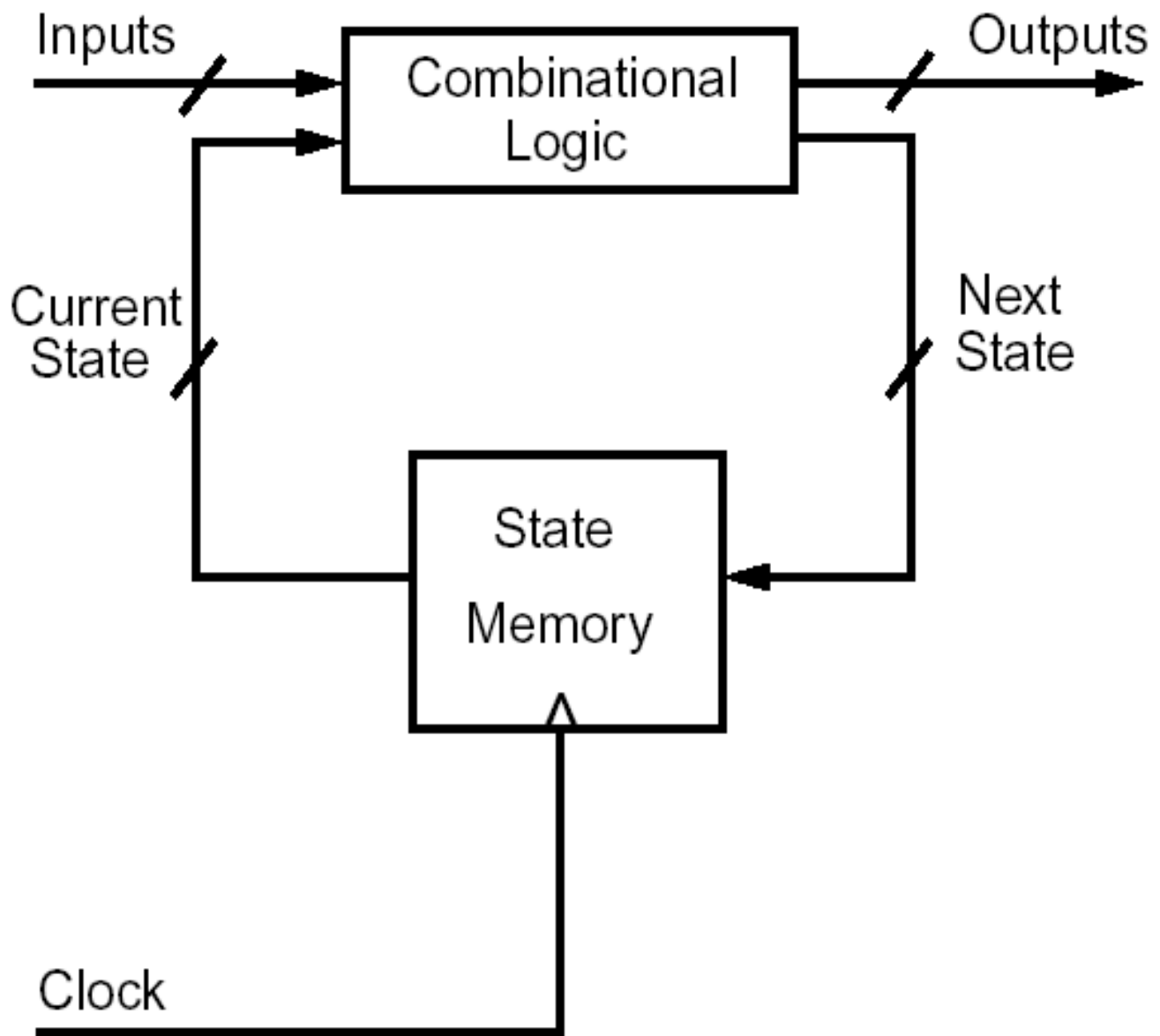


- The input D must be stable for a short period of time before the clock edge
- The input D must be stable for a short period of time after the clock edge

In practice

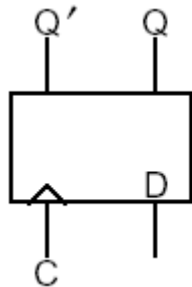
- Setup times are enforced by making sure that the clock cycle is long enough
- Hold times are generally 0 or very very small

(Clocked) Sequential Circuit



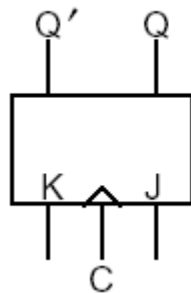
More Flip-Flops

D (data)



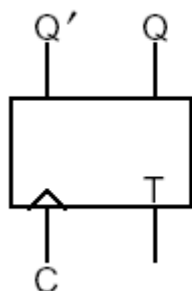
D	Q(t+1)
0	0
1	1

JK (refinement of RS)



J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	Q(t)'

T (toggle)



T	Q(t+1)
0	Q(t)
1	Q(t)'

Excitation Tables

If we want a flip-flop to make a certain state transition, what signals should we apply?

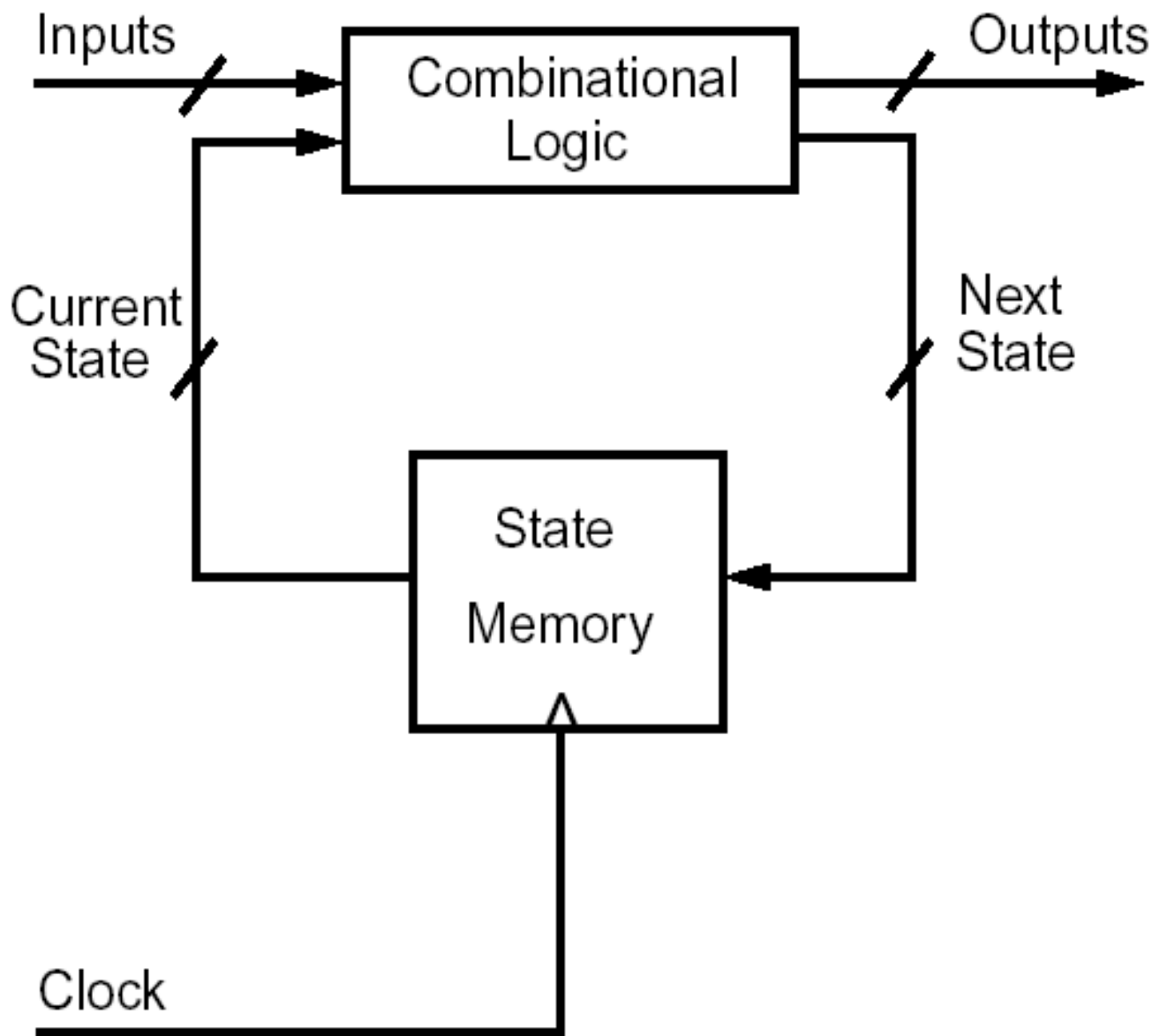
JK Characteristic Table

J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	Q(t)'

JK Excitation Table

Q(t)	Q(t+1)	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

(Clocked) Sequential Circuit



Dynamic Discipline

A (clocked) sequential circuit

- only clocked sequential devices and combinational devices
- no combinational cycles: every cycle contains a memory element
- the input values of every memory element must be stable before the clock edge, i.e. the clock cycle must be long enough for the combinational part to propagate