Sequential Circuits

Lecture 27
Section C.8

Robb T. Koether

Hampden-Sydney College

Mon, Nov 4, 2019

Outline

Sequential Circuits

- A sequential circuit has the ability to store a value.

Sequential Circuits

- A sequential circuit has the ability to store a value.
- Consequently, the output is not fully determined by the input.

Sequential Circuits

- A sequential circuit has the ability to store a value.
- Consequently, the output is not fully determined by the input.
- It is determined by the input together with the stored value(s).

Sequential Circuits

- A sequential circuit has the ability to store a value.
- Consequently, the output is not fully determined by the input.
- It is determined by the input together with the stored value(s).
- This forms the basis of the computer's ability to store data.

Outline

SR Latches

- An SR latch is a sequential circuit that has the ability to *store* a bit.

SR Latches

- An SR latch is a sequential circuit that has the ability to *store* a bit.
- Thus, it is a sequential circuit.

SR Latches

- An SR latch is a sequential circuit that has the ability to *store* a bit.
- Thus, it is a sequential circuit.
- It is also an unclocked circuit, meaning that the computer's clock (0 or 1) is not one of its inputs.

SR Latches

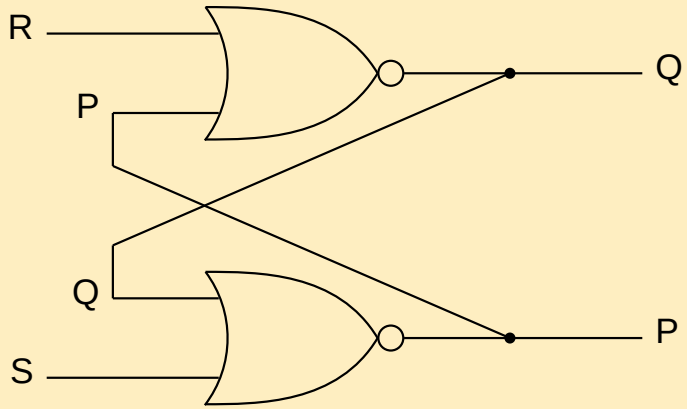
- An SR latch is a sequential circuit that has the ability to *store* a bit.
- Thus, it is a sequential circuit.
- It is also an unclocked circuit, meaning that the computer's clock (0 or 1) is not one of its inputs.
- It uses two NOR gates whose outputs are fed back in as inputs.

SR Latches

- An SR latch is a sequential circuit that has the ability to *store* a bit.
- Thus, it is a sequential circuit.
- It is also an unclocked circuit, meaning that the computer's clock (0 or 1) is not one of its inputs.
- It uses two NOR gates whose outputs are fed back in as inputs.
- (All sequential circuits use feedback.)

SR Latches

SR Latch



- *S* is “set” and *R* is “reset.”
- We will examine how the SR latch behaves for different values of *S* and *R*.

SR Latches

$R(t)$	$S(t)$	$Q(t-\varepsilon)$	$P(t-\varepsilon)$	$Q(t+\varepsilon)$	$P(t+\varepsilon)$
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		

Analyze the circuit when the inputs are different.

SR Latches

$R(t)$	$S(t)$	$Q(t-\varepsilon)$	$P(t-\varepsilon)$	$Q(t+\varepsilon)$	$P(t+\varepsilon)$
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	0	1

Analyze the circuit when the inputs are different.

SR Latches

$R(t)$	$S(t)$	$Q(t-\varepsilon)$	$P(t-\varepsilon)$	$Q(t+\varepsilon)$	$P(t+\varepsilon)$
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1		

Analyze the circuit when the inputs are both 0.

SR Latches

$R(t)$	$S(t)$	$Q(t-\varepsilon)$	$P(t-\varepsilon)$	$Q(t+\varepsilon)$	$P(t+\varepsilon)$
0	0	0	0	1	1
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	0

Analyze the circuit when the inputs are both 0.

Stability of the SR Latch

- Notice that when $R = S = 0$, the values of P and Q are unstable when both are 0 or both are 1.

Stability of the SR Latch

- Notice that when $R = S = 0$, the values of P and Q are unstable when both are 0 or both are 1.
- In reality, one of them receives the new value slightly ahead of the other, which then leads to stability.

Stability of the SR Latch

- Notice that when $R = S = 0$, the values of P and Q are unstable when both are 0 or both are 1.
- In reality, one of them receives the new value slightly ahead of the other, which then leads to stability.
- But it may take a few moments to stabilize.

SR Latches

$R(t)$	$S(t)$	$Q(t-\varepsilon)$	$P(t-\varepsilon)$	$Q(t+\varepsilon)$	$P(t+\varepsilon)$
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

Analyze the circuit when the inputs are both 1.

SR Latches

$R(t)$	$S(t)$	$Q(t-\varepsilon)$	$P(t-\varepsilon)$	$Q(t+\varepsilon)$	$P(t+\varepsilon)$
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

Analyze the circuit when the inputs are both 1.

SR Latches

SR Latch

R	S	Q	P
0	0	0	1
0	0	1	0
0	1	1	0
1	0	0	1
1	1	0	0

SR Latches

- If S is asserted (1) and R is deasserted (0), then Q is asserted (1), i.e., Q is set.

SR Latches

- If S is asserted (1) and R is deasserted (0), then Q is asserted (1), i.e., Q is set.
- If R is asserted (1) and S is deasserted (0), then Q is deasserted (0), i.e., Q is reset.

SR Latches

- If S is asserted (1) and R is deasserted (0), then Q is asserted (1), i.e., Q is set.
- If R is asserted (1) and S is deasserted (0), then Q is deasserted (0), i.e., Q is reset.
- Thus, the name, set-reset latch.

SR Latches

- If S is asserted (1) and R is deasserted (0), then Q is asserted (1), i.e., Q is set.
- If R is asserted (1) and S is deasserted (0), then Q is deasserted (0), i.e., Q is reset.
- Thus, the name, set-reset latch.
- If R and S are both 1, then the values of Q and P are 0. As we will see, this case will not arise.

SR Latches

- In all but the last case (which will not arise), $P = \overline{Q}$. Therefore, we will rename it $\overline{Q}$.

SR Latches

- In all but the last case (which will not arise), $P = \overline{Q}$. Therefore, we will rename it $\overline{Q}$.
- Note that when R and S are both deasserted (0), Q maintains its value (either 0 or 1) and $\overline{Q}$ maintains the complementary value.

SR Latches

- In all but the last case (which will not arise), $P = \overline{Q}$. Therefore, we will rename it $\overline{Q}$.
- Note that when R and S are both deasserted (0), Q maintains its value (either 0 or 1) and $\overline{Q}$ maintains the complementary value.
- Thus, the SR Latch is able to *store* the values of Q and $\overline{Q}$.

SR Latches

- In all but the last case (which will not arise), $P = \overline{Q}$. Therefore, we will rename it $\overline{Q}$.
- Note that when R and S are both deasserted (0), Q maintains its value (either 0 or 1) and $\overline{Q}$ maintains the complementary value.
- Thus, the SR Latch is able to *store* the values of Q and $\overline{Q}$.
- The S input (set) will set Q and the R input (reset) will reset Q.

SR Latches

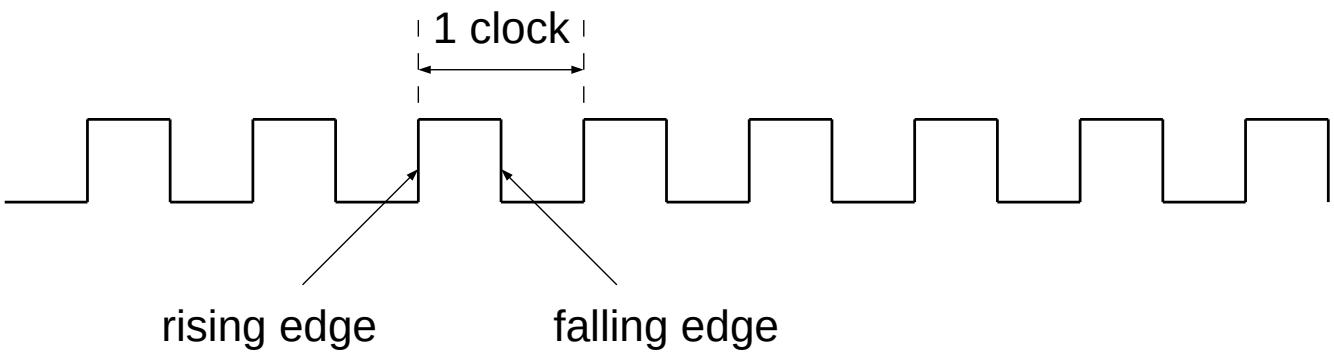
- How would the SR latch behave if we replaced the NOR gates with NAND gates?

Outline

The Clock

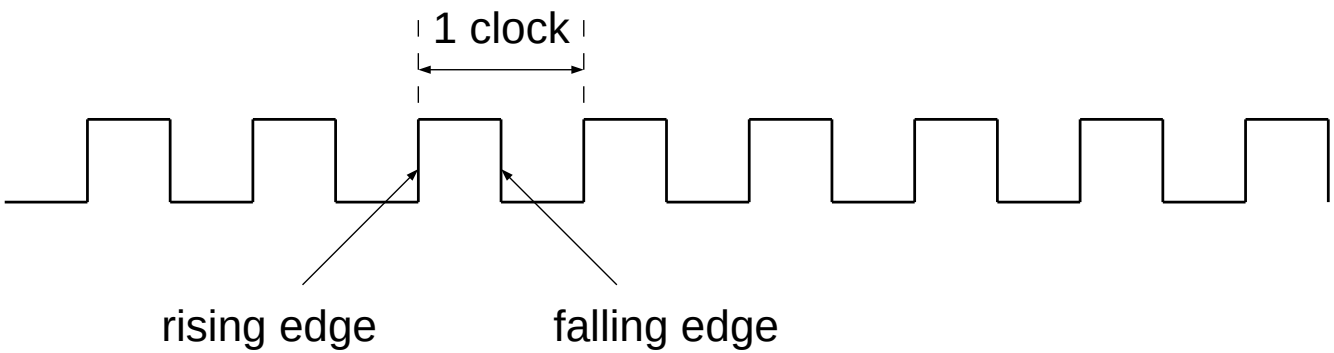
- The clock is a component of the processor whose value continuously oscillates between 0 and 1.
- As we will soon see, the clock is used to control all other components of the processor.

The Clock



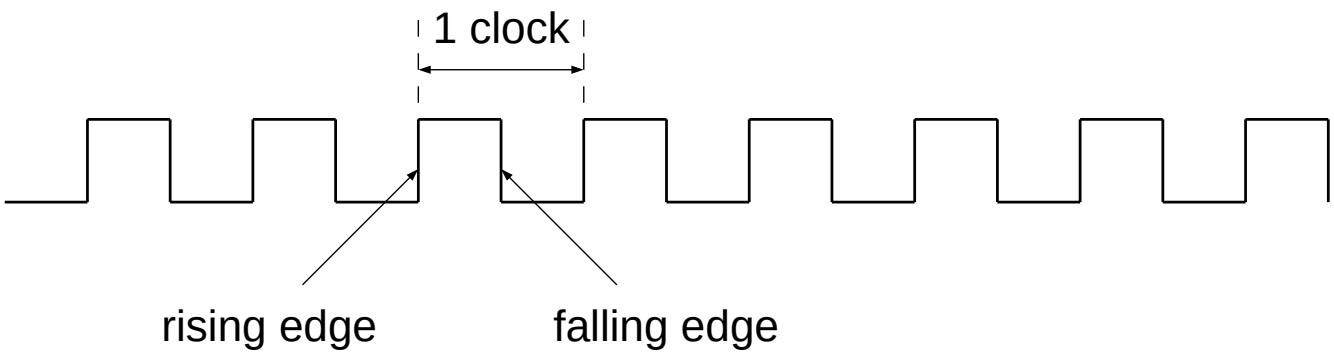
The behavior of the clock over time

The Clock



On the rising edge, the clock changes from 0 to 1

The Clock



On the falling edge, the clock changes from 1 to 0

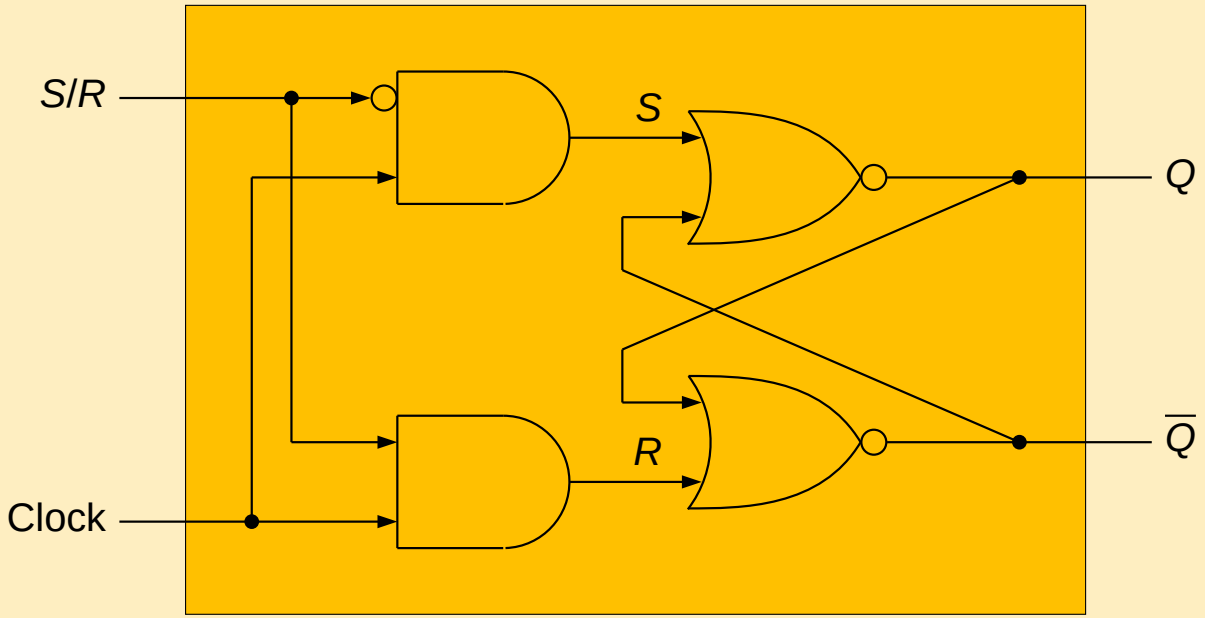
Outline

D Latches

- A D Latch is able to store a bit on the rising edge of the Clock.
- It uses an SR Latch to store the value and it uses a pair of AND-gates to initiate the storing when the clock switches from 0 to 1.
- It is a clocked circuit; one of its inputs it the clock.

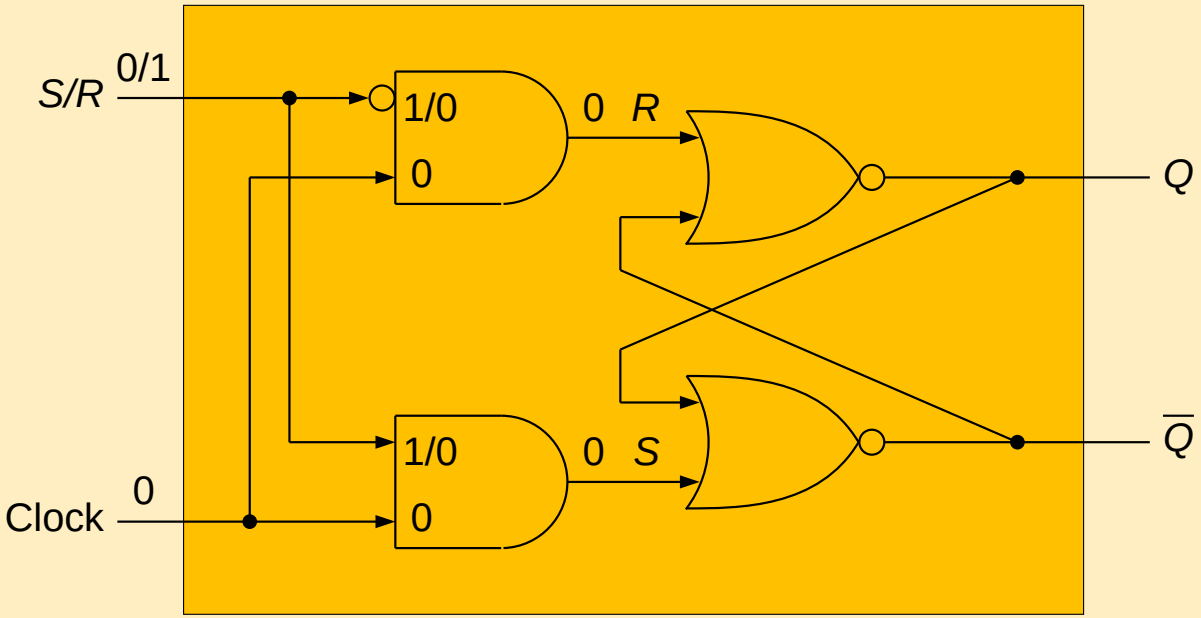
D Latches

D Latch



D Latches

D Latch



D Latches

- The input R/S is the data D to be stored.
- When Clock $= 0$, both AND gates will output 0.
- In this case, the latch is closed; the state of Q is unaffected.
- When Clock $= 1$, one of the AND gates will output 1, depending on the value of R/S (or D).
- In this case, the latch is open; the value of D is stored as Q.
- Note that the two AND gates cannot both output 1 simultaneously.

D Latches

- Suppose that initially Clock $= 0$ and $S/R = 1$.
- Consider a sequence of events where Clock switches to 1, then 0, then 1 again.
- Suppose that after Clock changes back to 0, S/R changes to 0, and then Clock changes back to 1.
- What happen to Q?

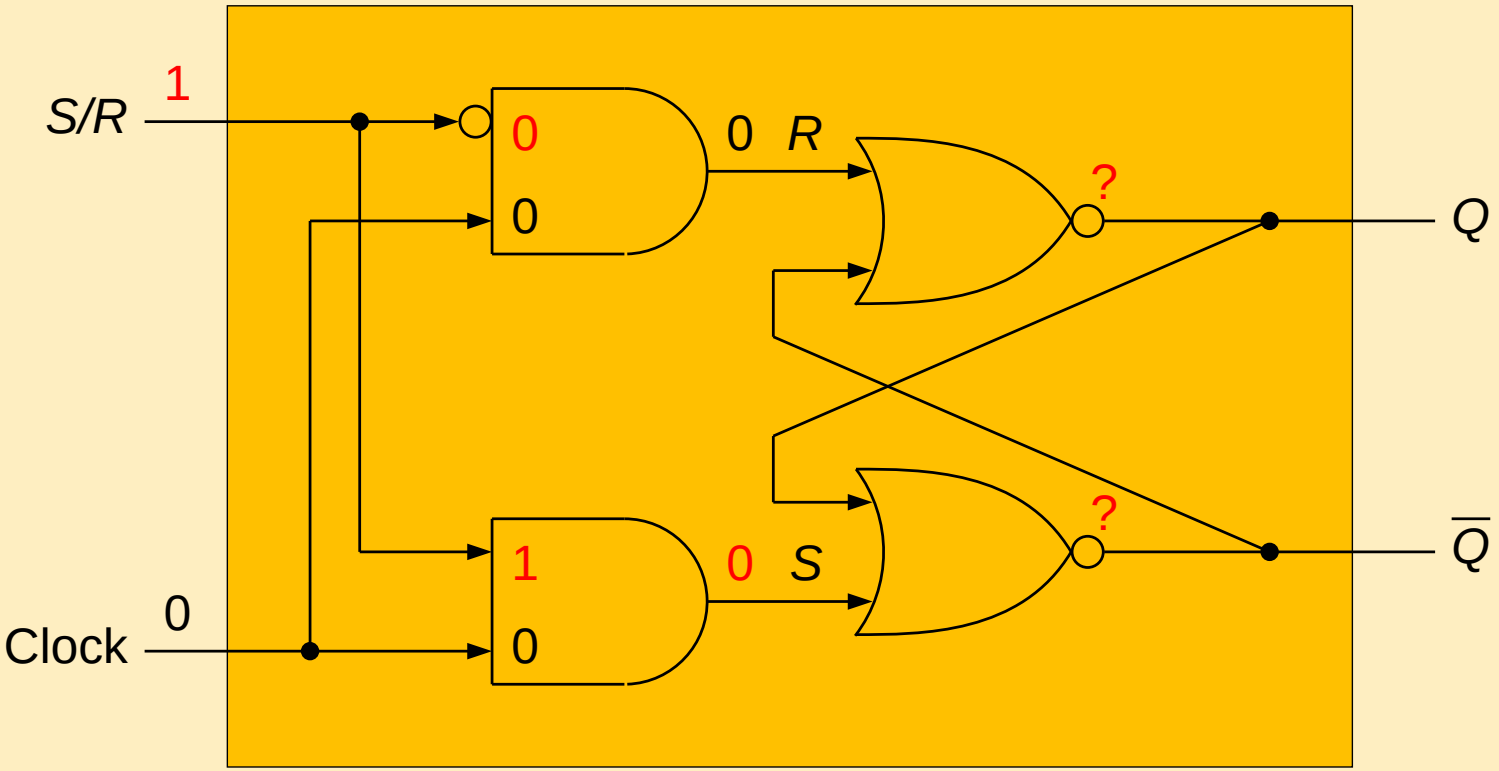
D Latches

D Latch

$$\text{Clock} = 0 \text{ and } D = ?(\textit{whatever})$$

D Latches

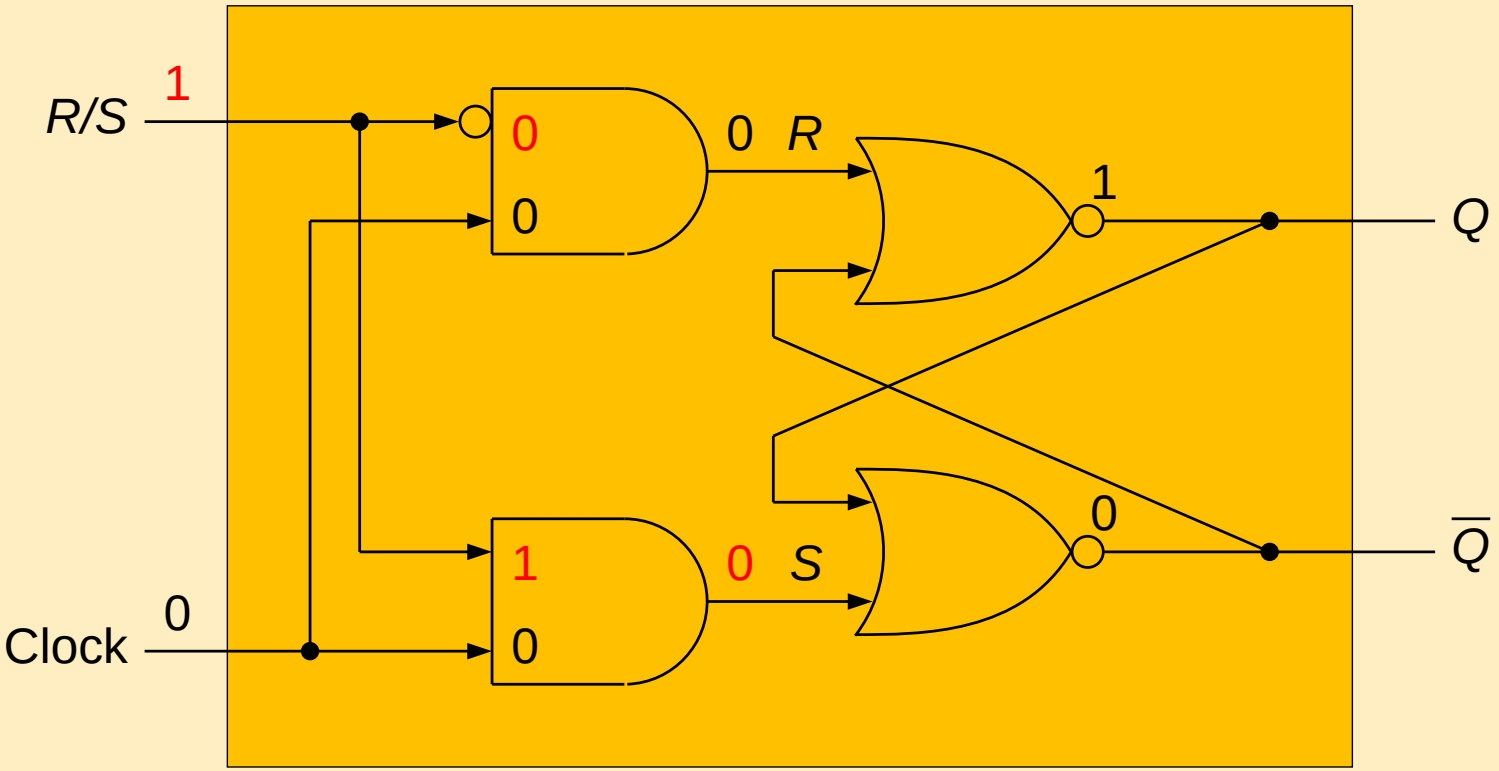
D Latch



Change Clock to 1; Set Q

D Latches

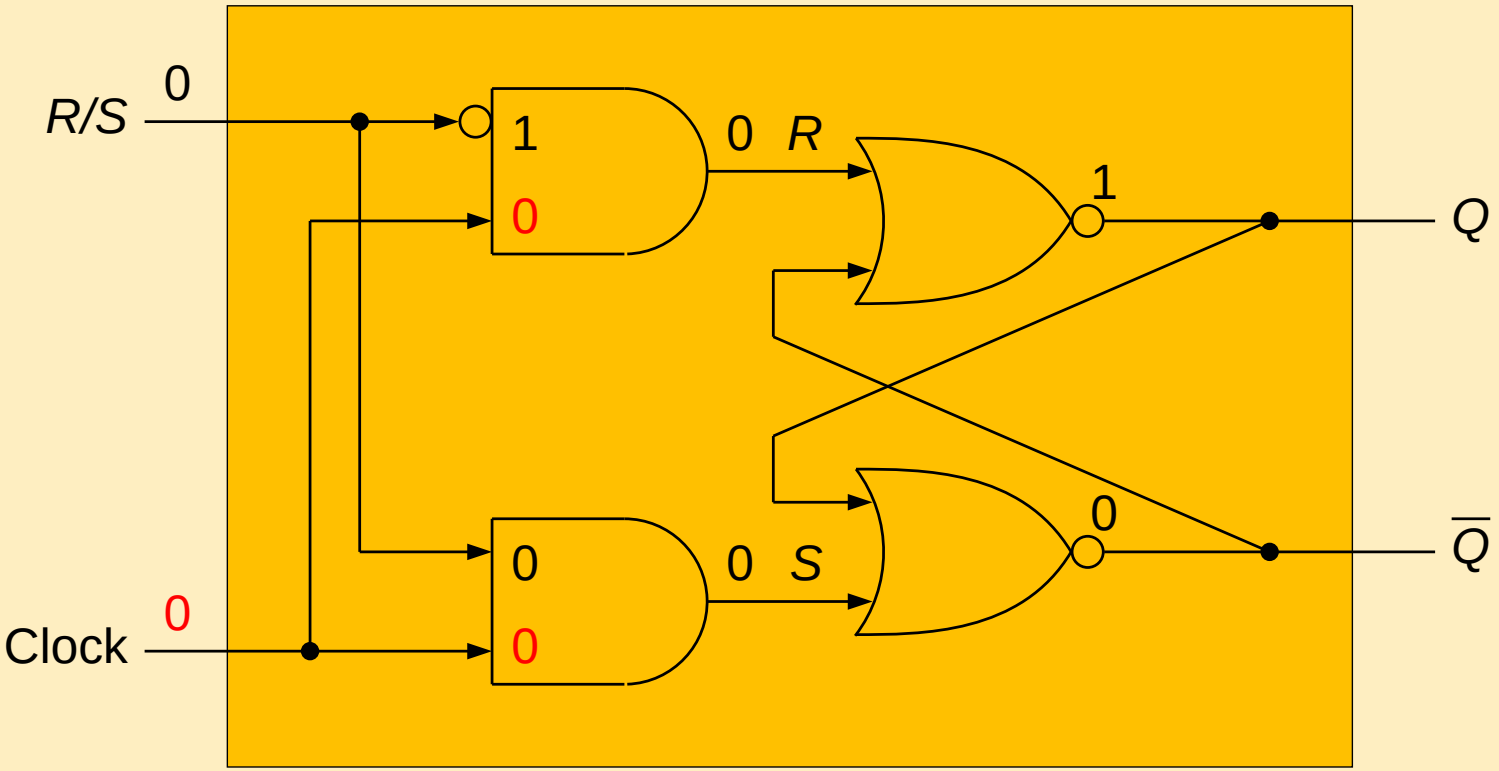
D Latch



Change Clock to 0; no change to Q

D Latches

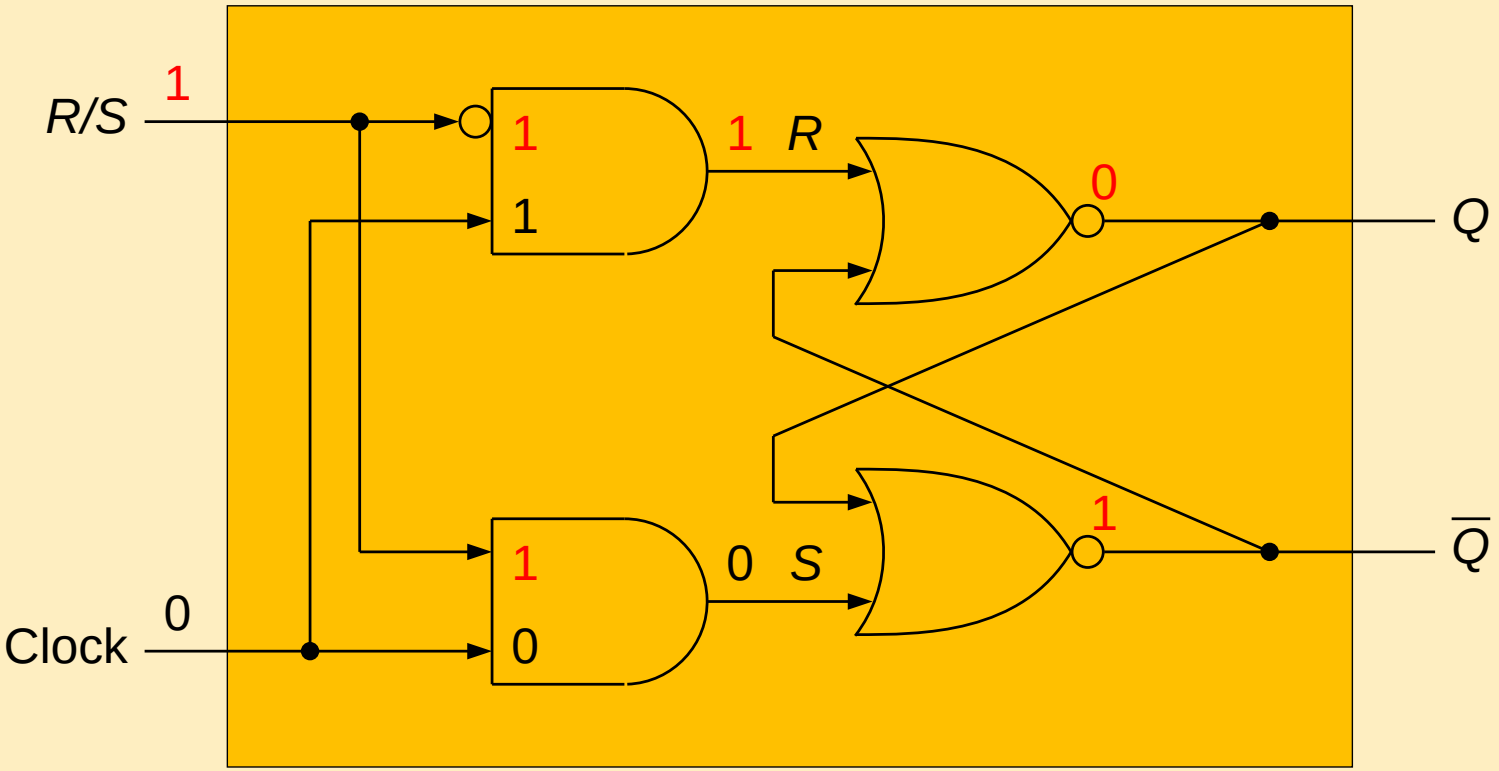
D Latch



Change S/R to 0; no change to Q

D Latches

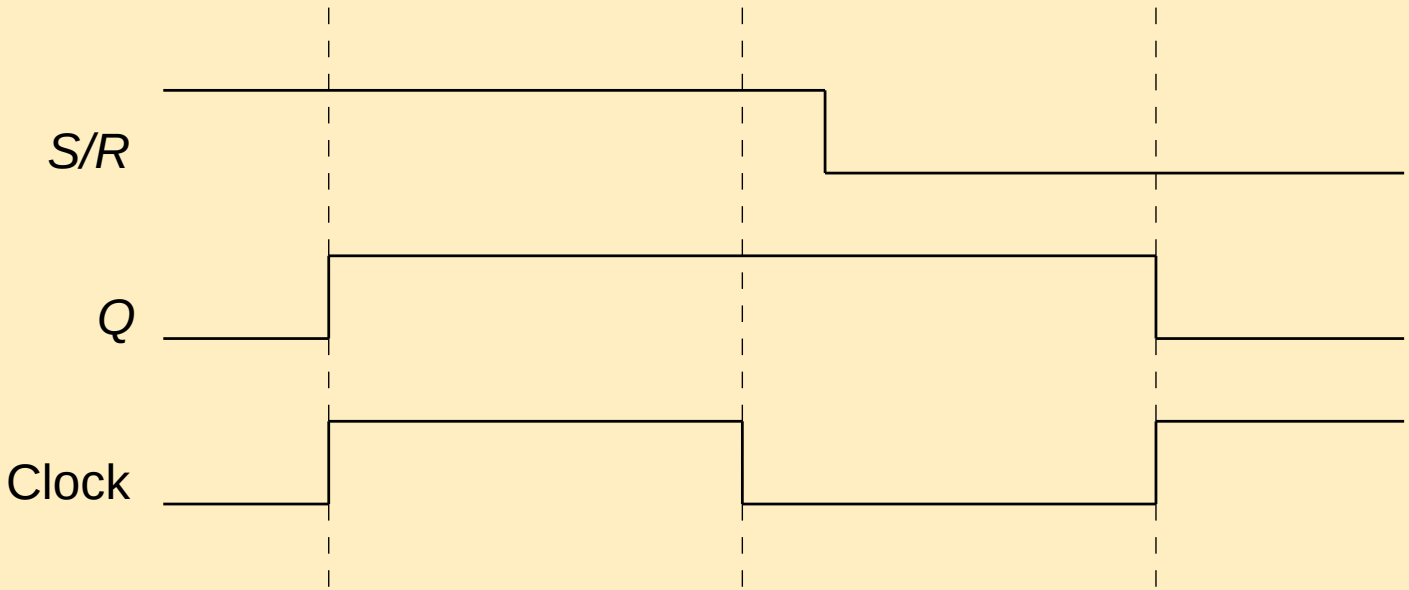
D Latch



Change Clock to 1; Reset Q

D Latches

D Latch



The clock and the input and output of the D latch over time.

D Latches

- The D Latch is transparent because when Clock $= 1$, the input D appears immediately as the output Q.

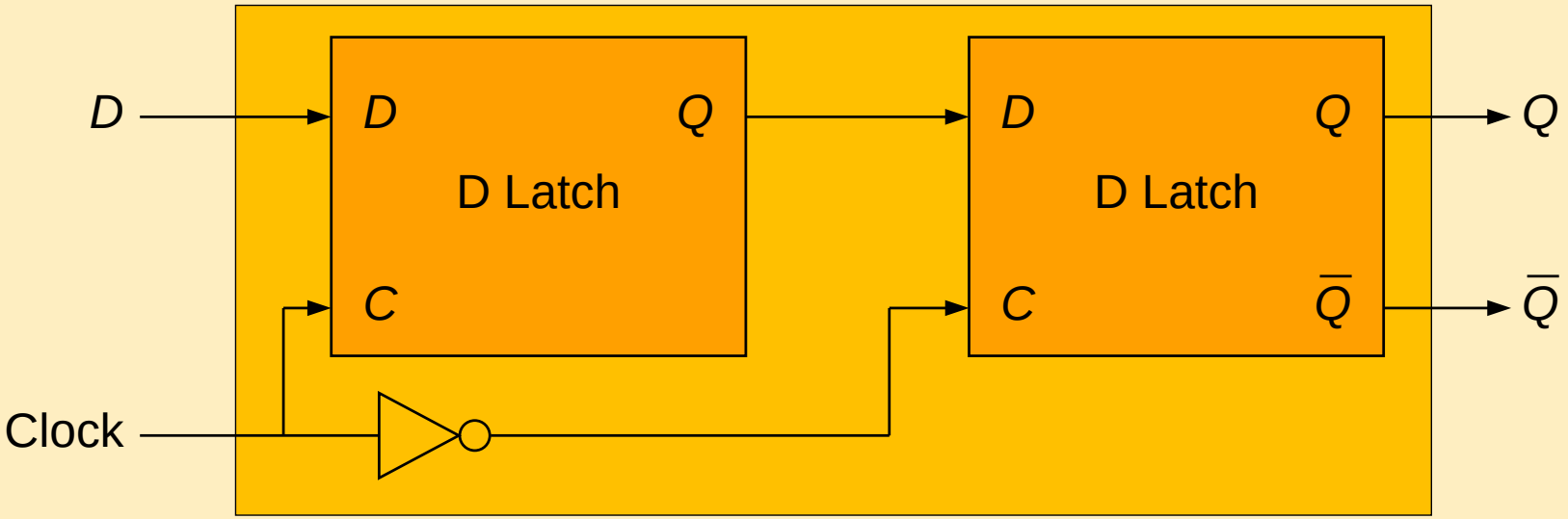
D Latches

- How would we design a latch that *closes* on the rising edge and *opens* on the falling edge?

Outline

D Flip Flops

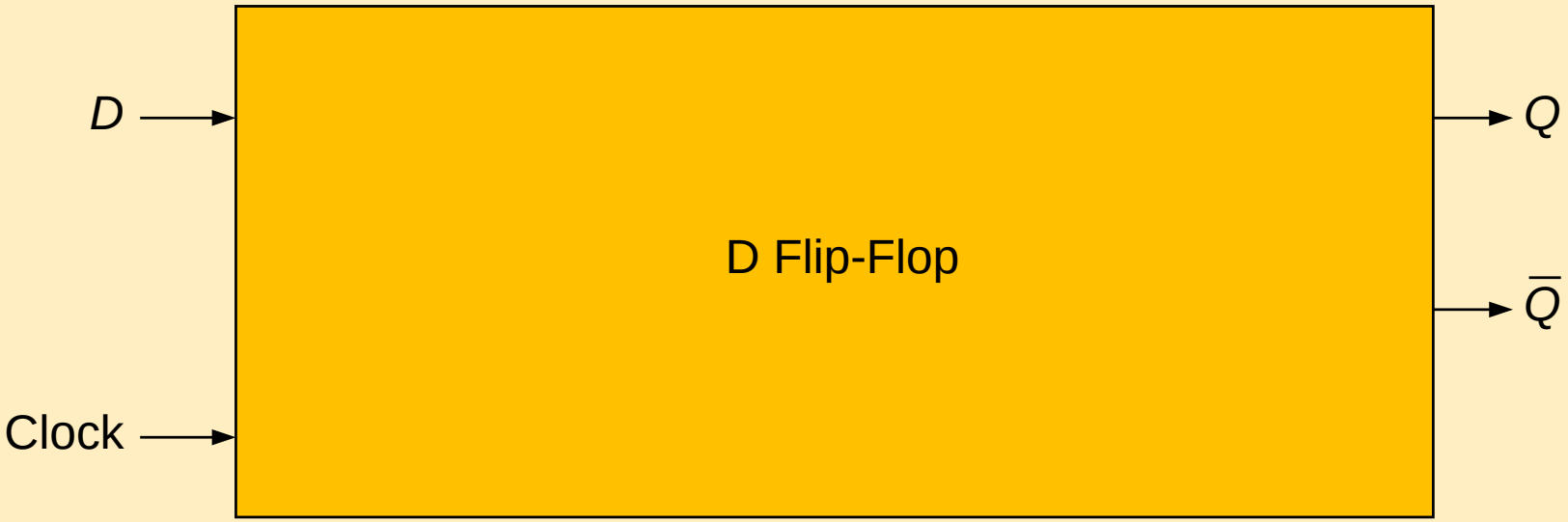
D Flip Flop



- The Clock will alternately set the value of the first latch or the second latch.

D Flip Flops

D Flip Flop



- Track the values of Q and $\overline{Q}$ over time as the clock C and the input D changes.

D Flip Flops

- Set-up time is the time it takes for data to stabilize before the rising clock edge.
- Hold time is the time the data must be stable after the rising clock edge.
- The value Q of the D Flip Flop is changed on the falling edge of the clock.

D Flip Flops

- The D Flip Flop is not transparent.
- The input is stored when Clock $= 1$, but it appears as output only when Clock $= 0$.

Outline

Assignment

Assignment

- Read Section C.8.