Latches

VOLKAN KURSUN

Some material from McGraw Hill

Outline

- Storage Elements
- Basic Latch
- Gated SR Latch
- Gated D Latch

Storage Elements

❑ In combinational circuits, the value of each output depends solely on the present values of signals applied to the inputs

❑ In sequential circuits, the values of outputs depend not only on the present values of the inputs but also on the **past behavior** of the circuit (past inputs and outputs)

❑ Sequential circuits include **storage elements** that store the values of logic signals: the contents of storage elements represent the state of the circuit

Sequence of States

❑ When the inputs change values, the new inputs either leave the circuit in the same state or cause the circuit to change to a new state

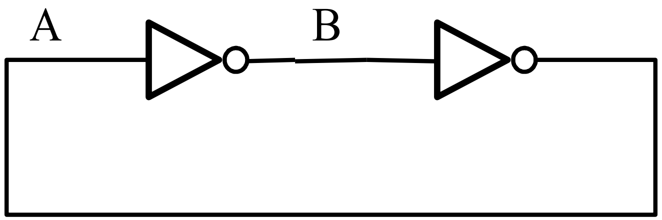
❑ Over time, a circuit with storage elements moves through a **sequence of states** in response to the changes in the inputs: **sequential circuits (state machines)**

Outline

- Storage Elements
- Basic Latch
- Gated SR Latch
- Gated D Latch

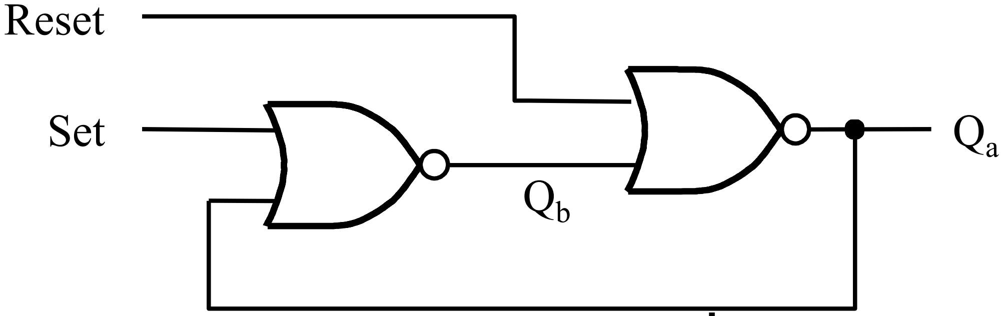
Simple Memory Element

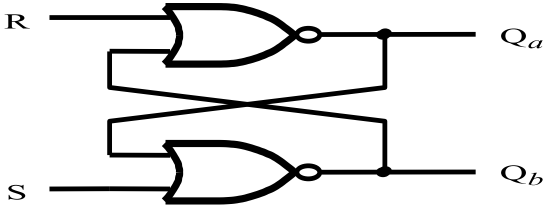
- ❑ Consists of a loop with two inverters (cross-coupled inverter pair)
- ❑ Circuit will maintain its state as long as the power supply is applied
- ❑ The circuit has two states (AB = 0b01 or AB = 0b10)
- ❑ The simple memory element **does NOT provide the means to control these two states**: circuit needs to be expanded to provide the capability to change state



Basic Latch

- ❑ A memory element built with two NOR gates: inputs Set and Reset provide the means to change state



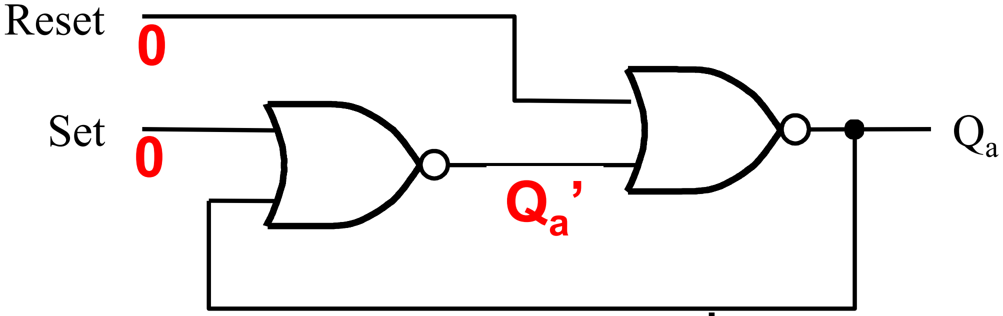


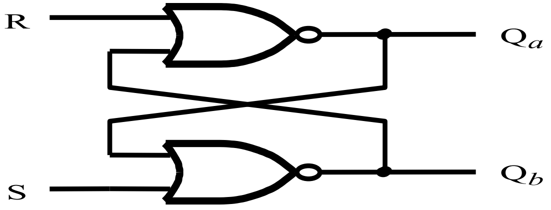
S	R	Q_a	Q_b
0	0	0/1	1/0
0	1	0	1
1	0	1	0
1	1	0	0

A more common way to draw the basic latch

Basic Latch

- ❑ When Set and Reset are both 0, **the latch maintains its state**





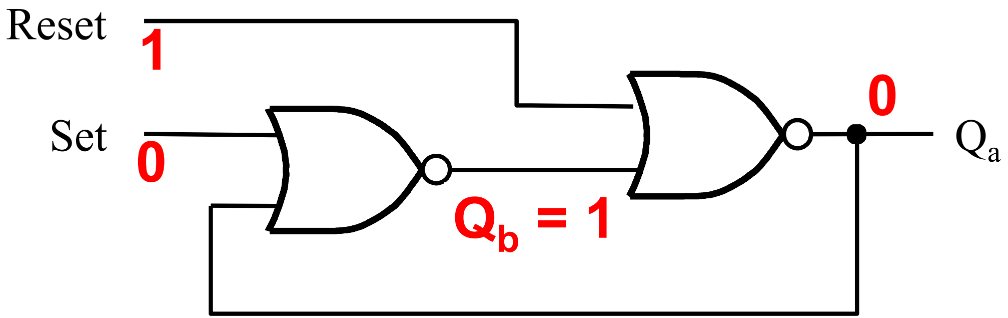
S	R	Q_a	Q_b
0	0	0/1	1/0
0	1	0	1
1	0	1	0
1	1	0	0

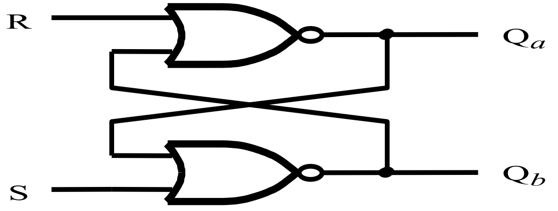
(no change of state)

A more common way to draw the basic latch

Basic Latch

- When Set = 0 and **Reset = 1**, **the latch is reset (Q_a = 0)**





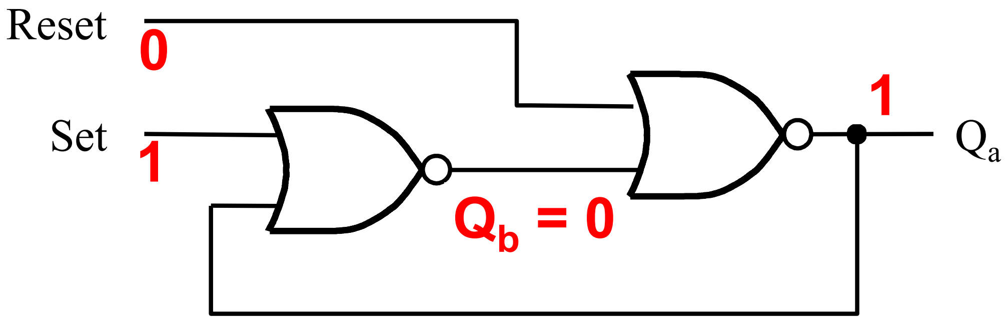
A more common way to draw the basic latch

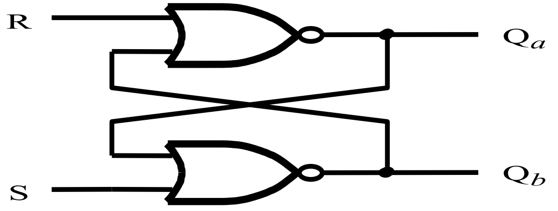
S	R	Q_a	Q_b
0	0	0/1	1/0
0	1	0	1
1	0	1	0
1	1	0	0

Reset

Basic Latch

- When **Set = 1** and Reset = 0, **the latch is set (Q_a = 1)**





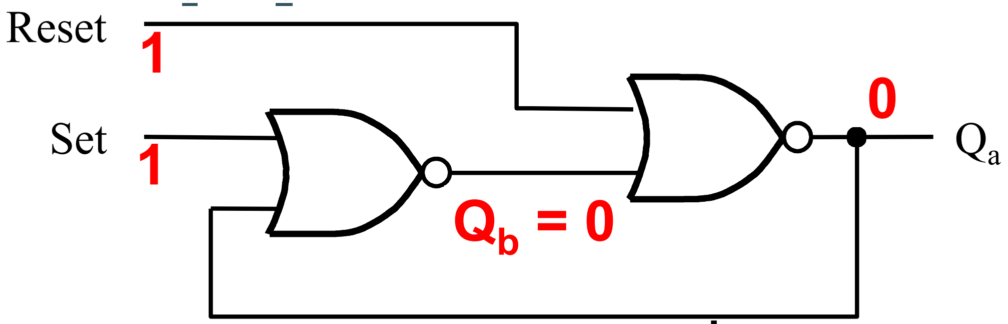
A more common way to draw the basic latch

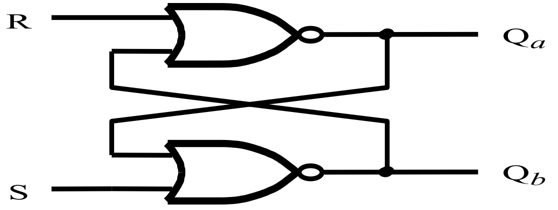
S	R	Q_a	Q_b
0	0	0/1	1/0
0	1	0	1
1	0	1	0
1	1	0	0

Set

Basic Latch

- When Set and Reset are both 1, both outputs are zero: **Q_a = Q_b = 0**





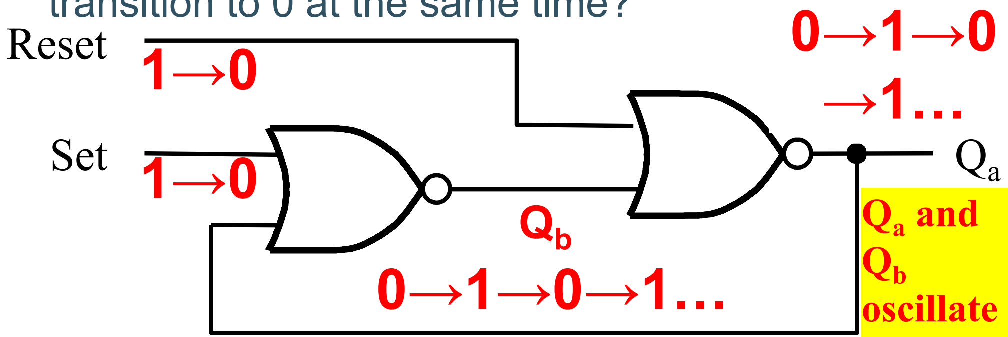
A more common way to draw the basic latch

S	R	Q_a	Q_b
0	0	0/1	1/0
0	1	0	1
1	0	1	0
1	1	0	0

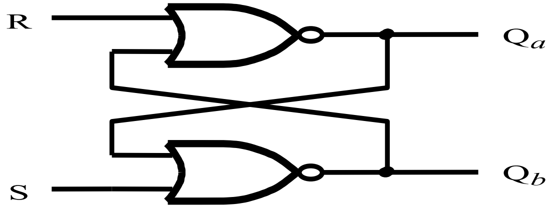
Both outputs are 0

Basic Latch Oscillation and Uncertainty

- After Set and Reset are both 1 and both outputs are zero: **Q_a = Q_b = 0**, what if both Set and Reset transition to 0 at the same time?



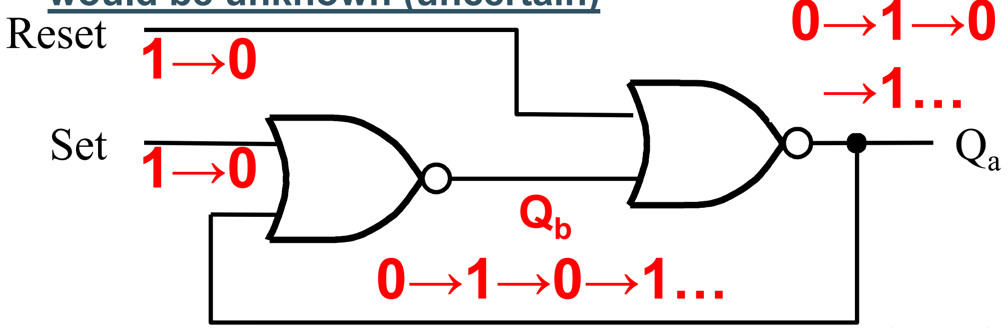
Q_a and Q_b oscillate between Q_a = Q_b = 1 and Q_a = Q_b = 0



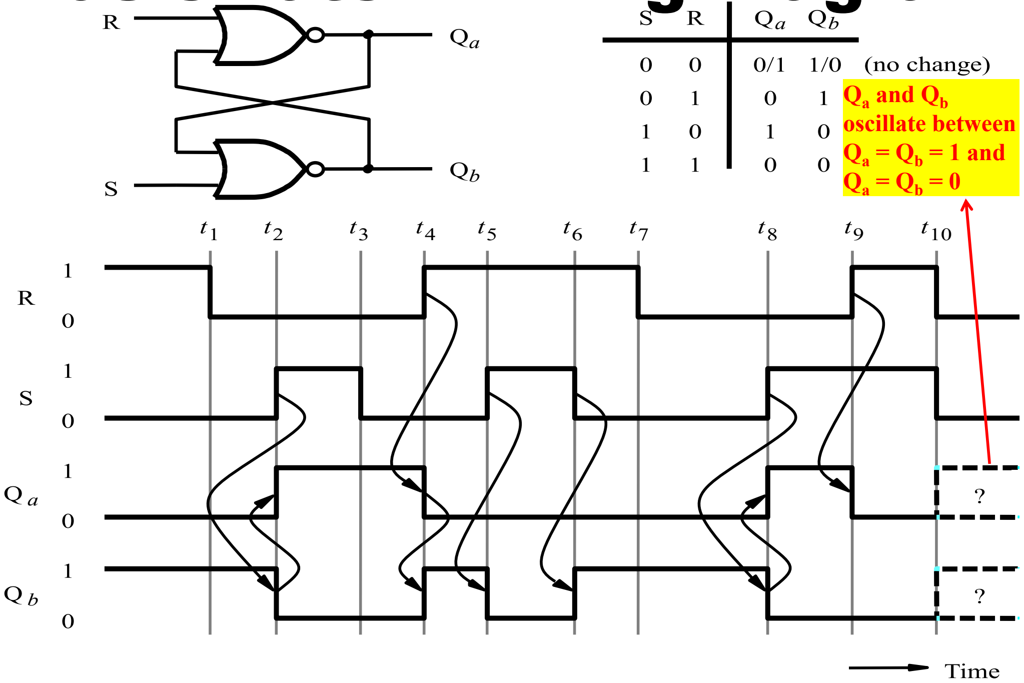
S	R	Q_a	Q_b
0	0	0/1	1/0
0	1	0	1
1	0	1	0
1	1	0	0

Basic Latch Oscillation and Uncertainty

- Qa and Qb oscillate between **$Q_a = Q_b = 1$** and **$Q_a = Q_b = 0$**
- **If** the delays through **the two NOR gates are identical**, the **oscillation will continue indefinitely**
- **In a real circuit, the delays of the two NOR gates will be different** and the latch will eventually settle down to one of its two stable states: **however, the final state would be unknown (uncertain)**



Basic Latch Timing Diagram



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	

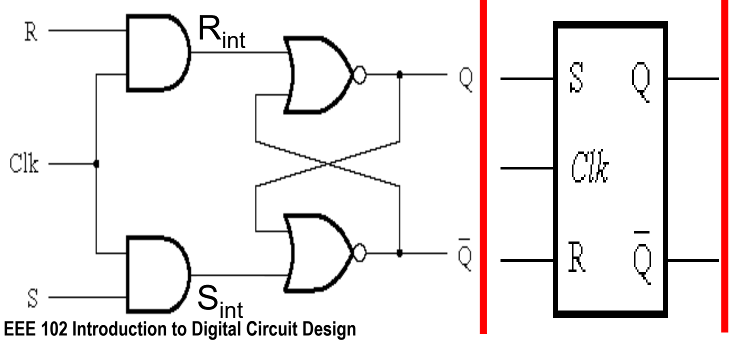
Q_a and Q_b oscillate between $Q_a = Q_b = 1$ and $Q_a = Q_b = 0$

Outline

- Storage Elements
- Basic Latch
- Gated SR Latch
- Gated D Latch

Gated SR Latch

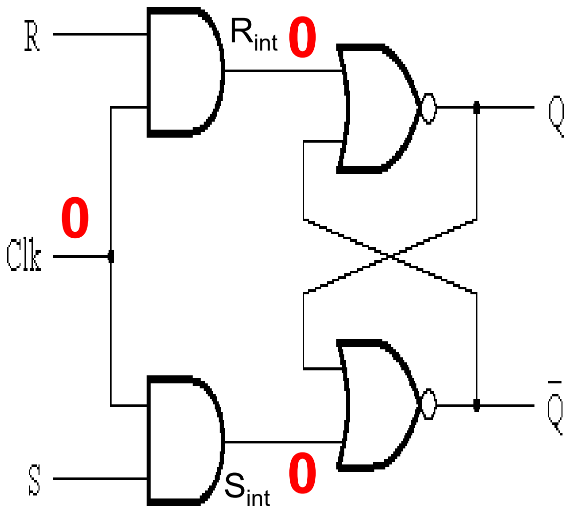
- In the basic SR latch, the state changes occur anytime that the set and reset inputs change
- Add an **enable signal** that would allow **controlling when** the latch would respond to the changes in its set and reset inputs: **when disabled, the changes in the set and reset signals would be ignored** by the latch and the existing state would be maintained



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	0

Gated SR Latch

- When **clk = 0**, the latch is disabled: the changes in the **set and reset signals** would be **ignored** by the latch and the **existing state** would be **maintained**

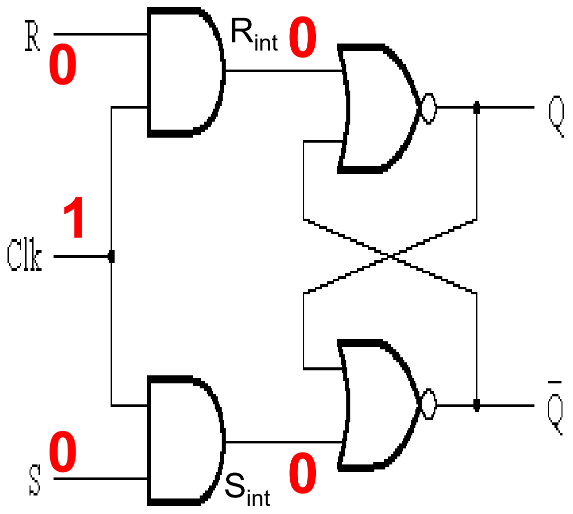


Clk	S	R	$Q(t+1)$
0	x	x	$Q(t)$
1	0	0	$Q(t)$
1	0	1	0
1	1	0	1
1	1	1	0

(no change of state)

Gated SR Latch: Level-Sensitive

- When clk = 1, the latch is enabled: R_{int} = R and S_{int} = S and the latch behaves like the basic SR latch
- If **S = R = 0**, the latch **maintains** its **state**

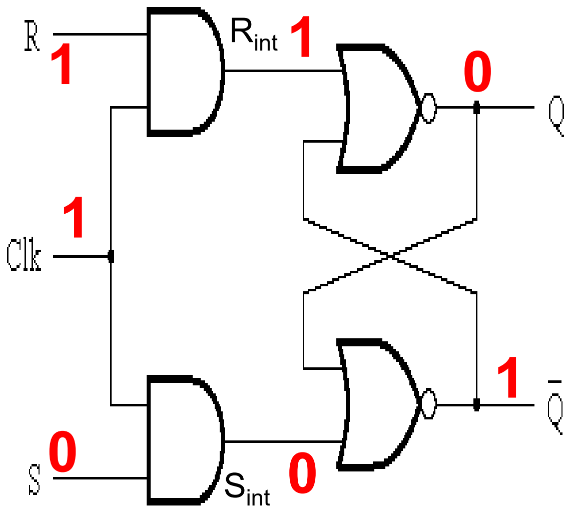


Clk	S	R	$Q(t+1)$
0	x	x	$Q(t)$ (no change)
1	0	0	$Q(t)$
1	0	1	0
1	1	0	1
1	1	1	0

(no change of state)

Gated SR Latch: Level-Sensitive

- When clk = 1, the latch is enabled: R_{int} = R and S_{int} = S and the latch behaves like the basic SR latch
- If S = 0 and **R = 1**, the latch is **reset**

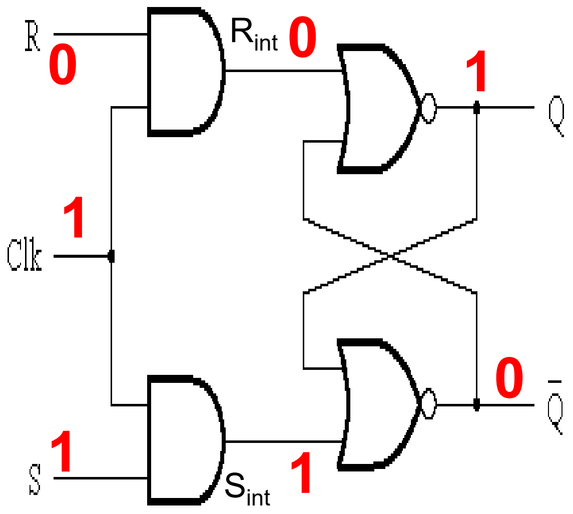


Clk	S	R	$Q(t+1)$
0	x	x	$Q(t)$ (no change)
1	0	0	$Q(t)$ (no change)
1	0	1	0
1	1	0	1
1	1	1	0

reset

Gated SR Latch: Level-Sensitive

- When clk = 1, the latch is enabled: R_{int} = R and S_{int} = S and the latch behaves like the basic SR latch
- If **S = 1** and R = 0, the latch is **set**

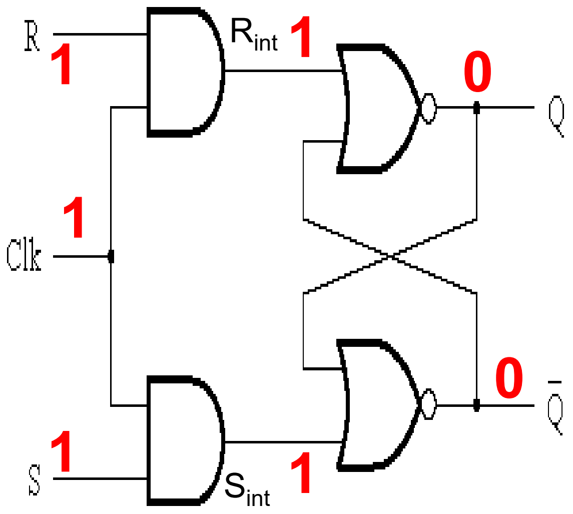


Clk	S	R	$Q(t+1)$
0	x	x	$Q(t)$ (no change)
1	0	0	$Q(t)$ (no change)
1	0	1	0
1	1	0	1
1	1	1	0

set

Gated SR Latch: Level-Sensitive

- ❑ When clk = 1, the latch is enabled: R_{int} = R and S_{int} = S and the latch behaves like the basic SR latch
- ❑ If **S = 1 and R = 1**, both **outputs are 0**



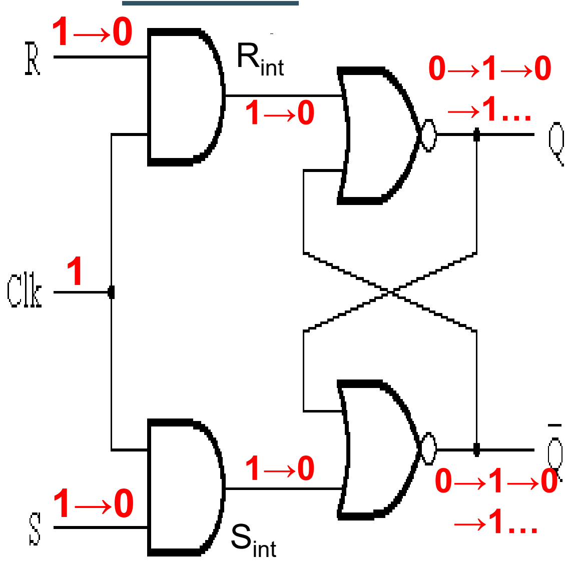
Clk	S	R	$Q(t+1)$
0	x	x	$Q(t)$ (no change)
1	0	0	$Q(t)$ (no change)
1	0	1	0
1	1	0	1
1	1	1	0

Undefined if both R and S or Clk transition to 0 afterwards

SR Latch Oscillation

- ❑ If **S = R = 1, Q = Q' = 0**. If S and R transition from 1 to 0 at the same time, Q and Q' oscillate between **Q = Q' = 0** and **Q = Q' = 1**

Oscillations after R&S 1→0 while Clk = 1

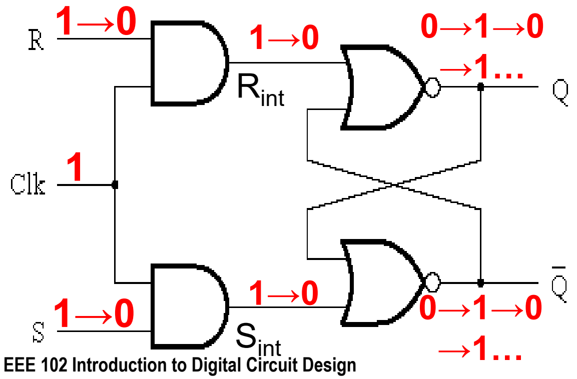


Clk	S	R	$Q(t+1)$
0	x	x	$Q(t)$ (no change)
1	0	0	Q(t
1	0	1	0
1	1	0	1
1	1	1	0

Oscillates if both R and S transition to 0 afterwards

SR Latch Oscillation

- ❑ Q and Q' oscillate between **Q = Q' = 1** and **Q = Q' = 0**
- ❑ **If** the delays through **the two NOR gates are identical**, the **oscillation will continue indefinitely**



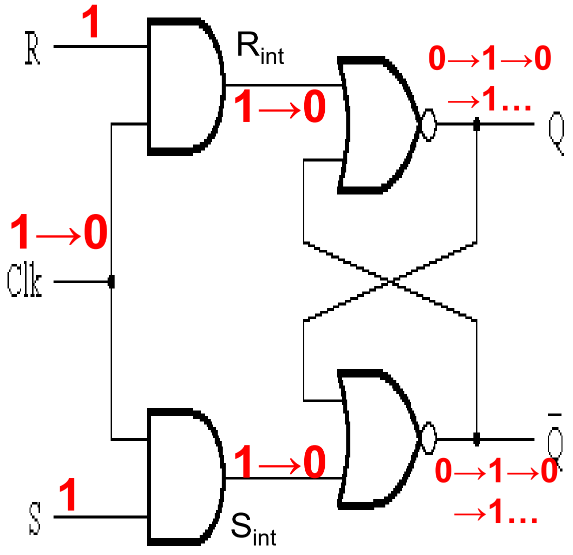
Clk	S	R	$Q(t+1)$
0	x	x	$Q(t)$ (no change)
1	0	0	Q(t
1	0	1	0
1	1	0	1
1	1	1	0

Oscillates if both R and S transition to 0 afterwards

SR Latch Oscillation

- ❑ Similarly**, if Clk transitions from 1 to 0 while S = R = 1**, Q and Q' oscillate between **Q = Q' = 0** and **Q = Q' = 1**

Oscillations after Clk 1→0

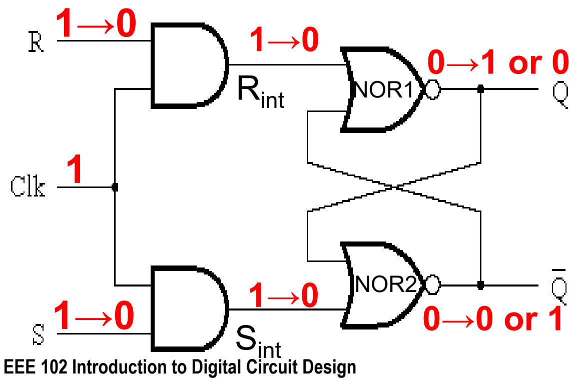


Clk	S	R	$Q(t+1)$
0	x	x	$Q(t)$ (no change)
1	0	0	Q(t
1	0	1	0
1	1	0	1
1	1	1	0

Oscillates if Clk transitions to 0 while S = R = 1

SR Latch Uncertainty

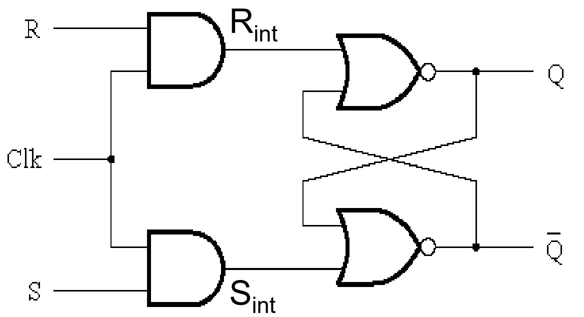
- In a real circuit, the delays of the two NOR gates will be different and the latch will eventually settle down to one of its two stable states: **<u>however, the final state would be unknown (uncertain)</u>**
- If $\text{Delay}_{\text{NOR1}} < \text{Delay}_{\text{NOR2}}$: will stabilize to Q = 1 and Q' = 0
- Alternatively, if $\text{Delay}_{\text{NOR2}} < \text{Delay}_{\text{NOR1}}$: will stabilize to Q = 0 and Q' = 1



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

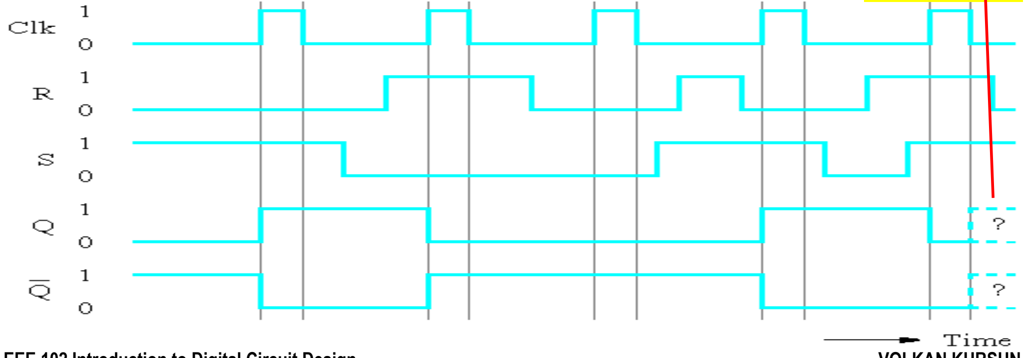
<u>Undefined</u> if both R and S or Clk transition to 0 afterwards

SR Latch Timing Diagram



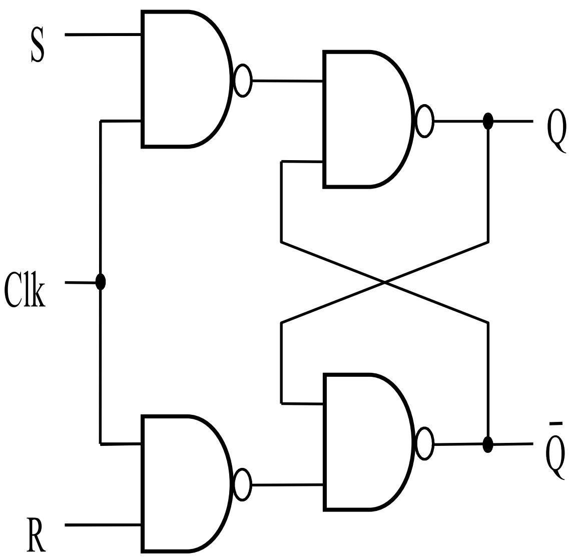
Clk	S	R	Q(t+1)
0	x	x	Q(t) (no change)
1	0	0	Q(t) (no change)
1	0	1	0
1	1	0	1
1	1	1	0

When Clk transitions from 1 to 0 while S = R = 1, Q and Q' oscillate between Q = Q' = 1 and Q = Q' = 0



Gated SR Latch with NAND

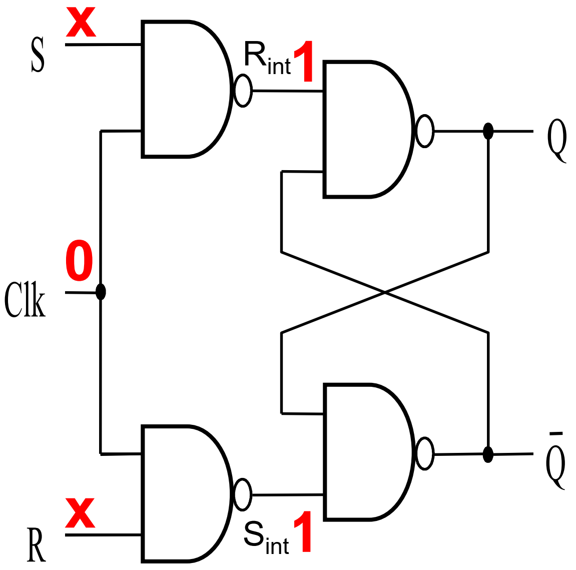
- S and R assignments have been reversed as compared to the AND-NOR implementation



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	1

Gated SR Latch with NAND

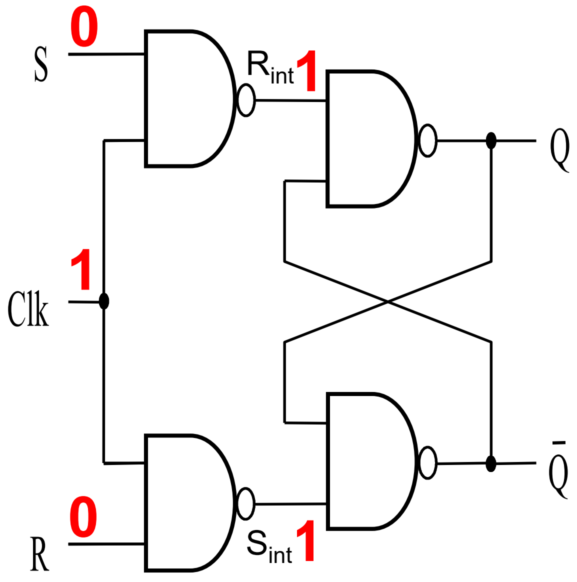
- Clk = 0: **<u>maintains state</u>**



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	1

Gated SR Latch with NAND

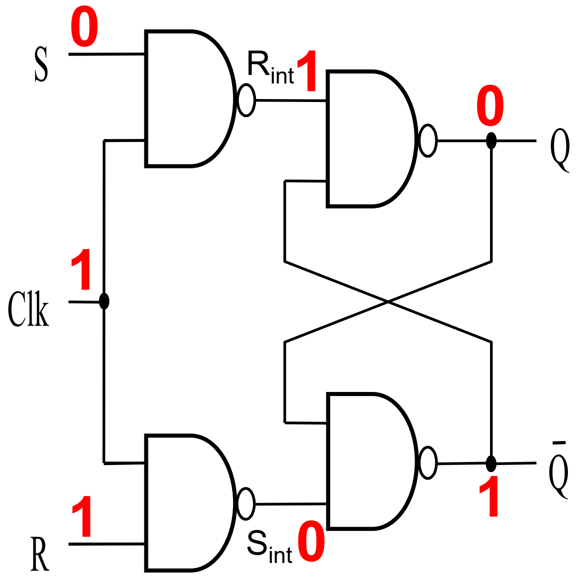
- Clk = 1, S = 0, R = 0: **maintains state**



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	1

Gated SR Latch with NAND

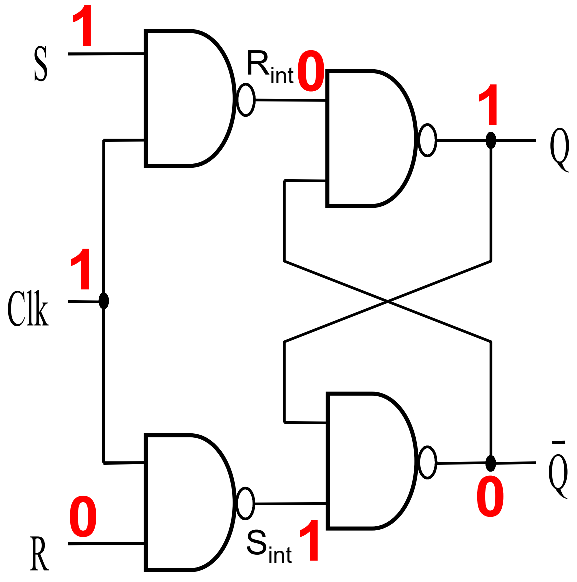
- Clk = 1, S = 0, R = 1: **reset** the latch



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	1

Gated SR Latch with NAND

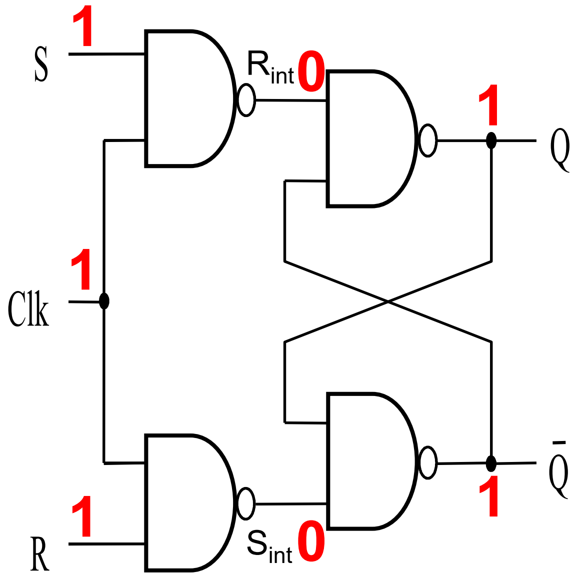
- Clk = 1, S = 1, R = 0: **set** the latch



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	1

Gated SR Latch with NAND

- Clk = 1, S = 1, R = 1: **Q = Q’ = 1**



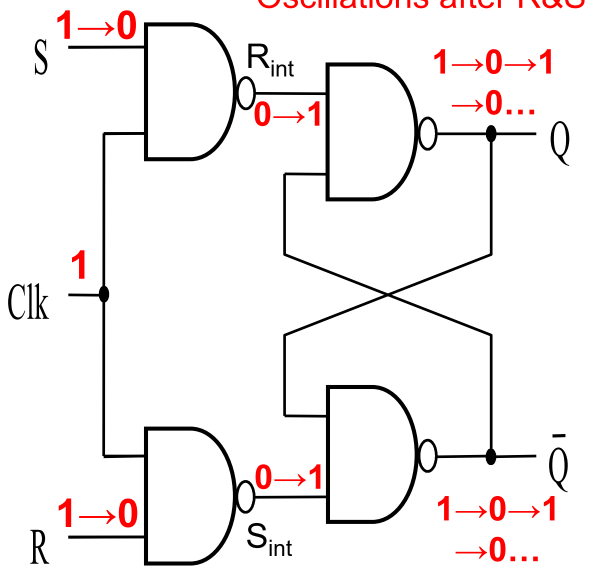
Clk	S	R	Q(t+1)
0	x	x	Q(t) (no change)
1	0	0	Q(t) (no change)
1	0	1	0
1	1	0	1
1	1	1	1

Undefined if both R and S or Clk transition to 0 afterwards

NAND SR Latch Oscillations

- If S and R transition from 1 to 0 at the same time, Q and Q' oscillate between **Q = Q' = 1** and **Q = Q' = 0**

Oscillations after R&S 1→0 while Clk = 1



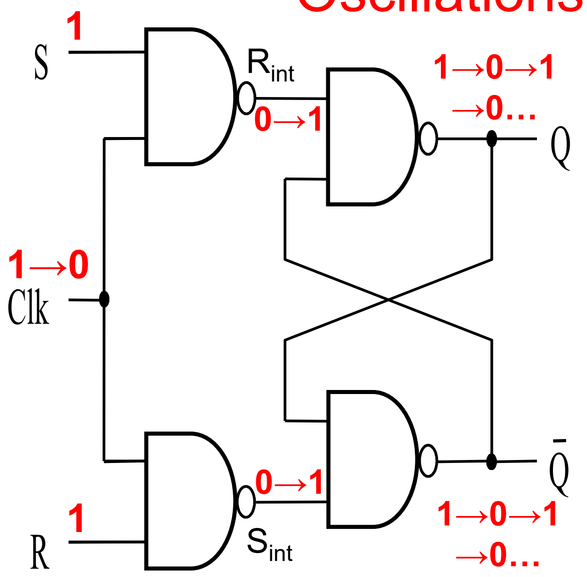
Clk	S	R	Q(t+1)
0	x	x	Q(t) (no change)
1	0	0	Q(t) (no change)
1	0	1	0
1	1	0	1
1	1	1	1

Oscillates if both R and S transition to 0 afterwards

NAND SR Latch Oscillations

- Similarly**, if Clk transitions from 1 to 0 while S = R = 1**, Q and Q' oscillate between **Q = Q' = 1** and **Q = Q' = 0**

Oscillations after Clk 1→0

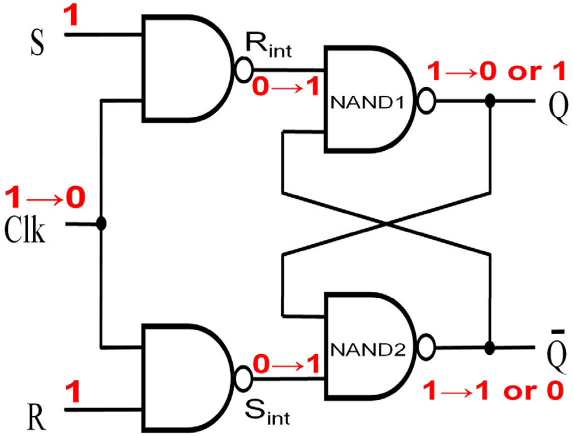


Clk	S	R	Q(t+1)
0	x	x	Q(t) (no change)
1	0	0	Q(t) (no change)
1	0	1	0
1	1	0	1
1	1	1	1

Oscillates if clk transitions to 0 afterwards

NAND SR Latch Uncertainty

- In a real circuit, the delays of the two NAND gates will be different and the latch will eventually settle down to one of its two stable states (set or reset state): **however, the final state would be unknown (uncertain)**
- If Delay$_{NAND1}$ < Delay$_{NAND2}$: will stabilize to Q = 0 and Q' = 1
- Alternatively, if Delay$_{NAND2}$ < Delay$_{NAND1}$: will stabilize to Q = 1 and Q' = 0

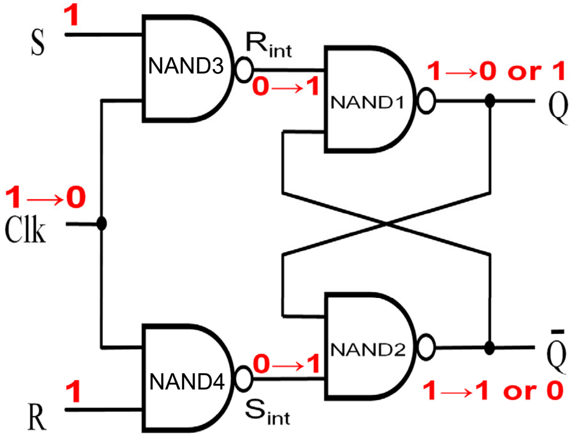


Clk	S	R	Q(t+1)
0	x	x	Q(t) (no change)
1	0	0	Q(t) (no change)
1	0	1	0
1	1	0	1
1	1	1	1

Undefined if both R and S or Clk transition to 0 afterwards

NAND SR Latch Uncertainty

- Similarly, if the delays of NAND3 and NAND4 gates are different, the latch will settle down to one of its two stable states (set or reset state): **however, the final state would be unknown (uncertain)**
- If Delay$_{NAND3}$ < Delay$_{NAND4}$: will stabilize to Q = 0 and Q' = 1
- Alternatively, if Delay$_{NAND4}$ < Delay$_{NAND3}$: will stabilize to Q = 1 and Q' = 0



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	1

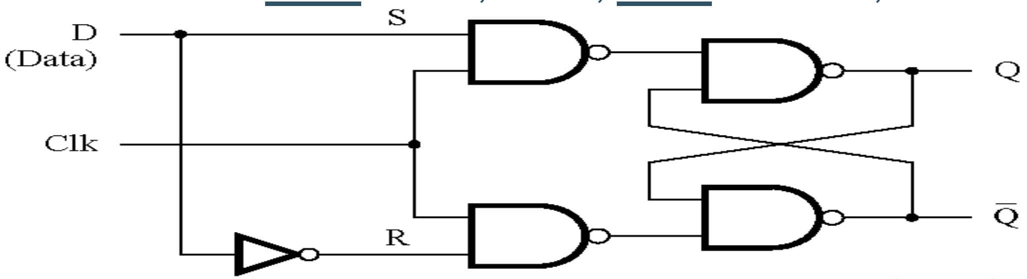
Undefined if both R and S or Clk transition to 0 afterwards

Outline

- Storage Elements
- Basic Latch
- Gated SR Latch
- Gated D Latch

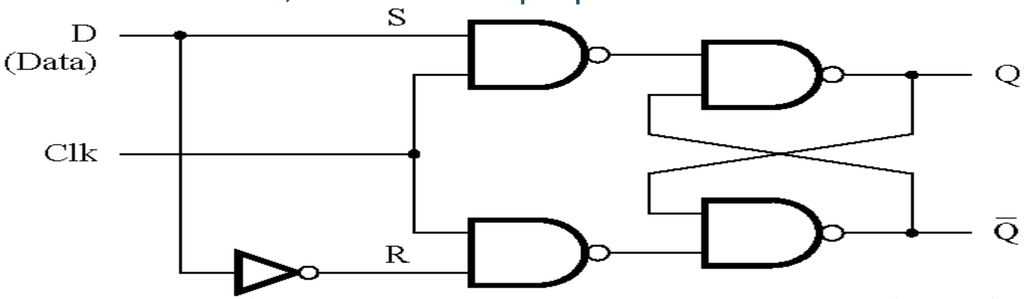
Gated D Latch

- Stores the value of single data input (bit storage)
- Data is stored under the control of a clock signal: level-sensitive
- There is only one data input in a D latch and the internal Set and Reset signals are generated from the single data input: S = D, R = D’
- Clk = 1 and **D = 1**: S = 1, R = 0, **set** the latch, Q = 1
- Clk = 1 and **D = 0**: S = 0, R = 1, **reset** the latch, Q = 0



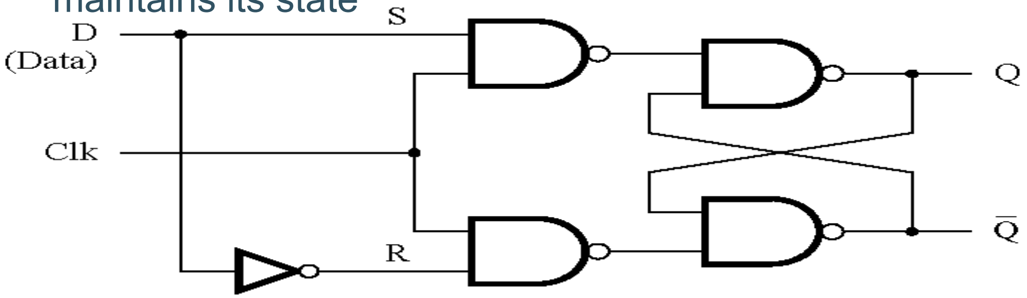
Gated D Latch

- Since S = D and R = D’, the troublesome situation where **S = R = 1 cannot occur in a D latch: no output uncertainty in a D latch**
- The output Q tracks the input D as long as Clk = 1 (latch is transparent when Clk = 1): level-sensitive behavior
- **D latch stores the value of D shortly before the Clk transitions from 1 to 0**
- When Clk = 0, the latch is opaque and maintains state

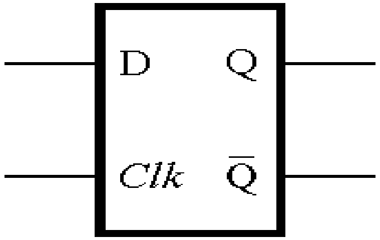


Gated D Latch Operation

- As long as Clk = 1, the Q output follows the D input
- When Clk = 0, the Q output cannot change and the latch maintains its state



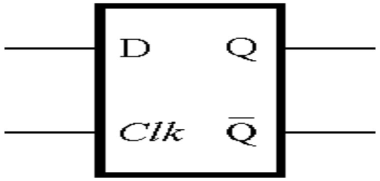
Clk	D	$Q(t+1)$
0	x	$Q(t)$
1	0	0
1	1	1

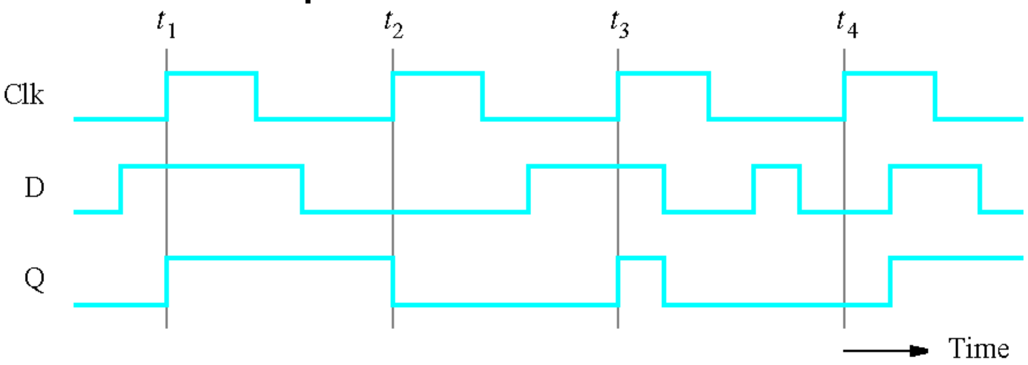


Gated D Latch Timing Diagram

- ❑ As long as Clk = 1, the Q output follows the D input
- ❑ When Clk = 0, the Q output cannot change: maintains state

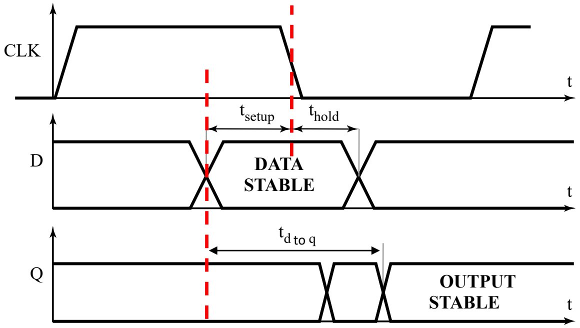
Clk	D	$Q(t+1)$
0	x	$Q(t)$
1	0	0
1	1	1



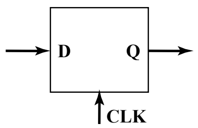


Positive Latch Timing Definitions

- ❑ **Transparent when clock is positive (clock = 1)**: Stores the value of D input that is present before the Clk 1 to 0 transition
- ❑ Opaque when the clock is negative (clock = 0)
- ❑ Set-up time
  - ❑ Valid data must arrive by **set-up time before the negative edge** of the clock (before the latch becomes opaque – end of the sampling period)
- ❑ Hold-time
  - ❑ Valid data must stay until **hold time after the negative edge** of the clock

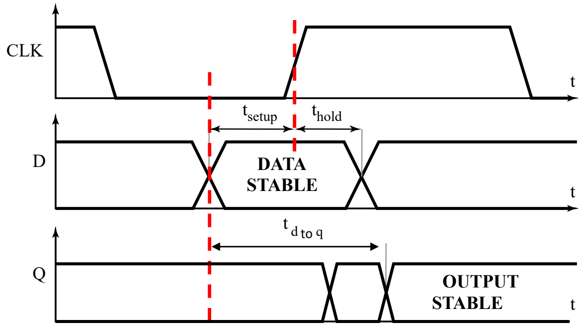


Positive latch

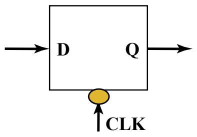


Negative Latch Timing Definitions

- ❑ **Transparent when the clock is negative**
  - ❑ Samples the input when clock = 0
- ❑ Opaque when the clock is positive (clock = 1)
- ❑ **Set-up time**
  - ❑ Valid data must arrive by set-up time **before the positive edge** of the clock (before the latch becomes opaque - end of the sampling period)
- ❑ **Hold-time**
  - ❑ Valid data must stay until hold time **after the positive edge of the clock**



Negative latch



Gated D Latch VHDL

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY latch IS
    PORT (        D, Clk      : IN      STD_LOGIC ;
                  Q           : OUT  STD_LOGIC) ;
END latch ;

ARCHITECTURE Behavior OF latch IS
BEGIN
    PROCESS ( D, Clk )
    BEGIN
        IF Clk = '1' THEN
            Q <= D ;
        END IF ;
    END PROCESS ;
END Behavior ;
```

The code does not specify what Q should be assigned when the condition for the if statement is not satisfied: implies that the Q should maintain its value. **IMPLIED MEMORY**