Unit 11

Latches and Flip-Flops

Logic Circuits (Spring 2022)

"Memory" Devices

- *Sequential* switching circuits
  - The output depends not only on *the present input* but also on *the past sequence of inputs*
  - The circuits must be able to "remember" something about the past history of the inputs in order to produce the present output
- *Latches* and *flip-flops*
  - Commonly used memory devices in sequential circuits
  - Assume one of two stable output states
  - Have one or more inputs that can cause the output state to change

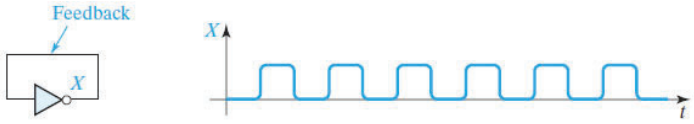
Latch vs. Flip-Flop

- Latch
  - A memory element that has no clock input
  - Each latch works *independently* with respect to timing
- Flip-flop
  - A memory element that has a clock input
  - The operation of all flip-flops are *synchronized* by a common clock or pulse generator
  - The flip-flops can only change output in response to a clock input, not data inputs
- Synchronous circuits
  - The changes in the state of memory devices are synchronized by a clock signal
  - Nearly all digital circuits are fully synchronous

  cf. Asynchronous circuits

Feedback

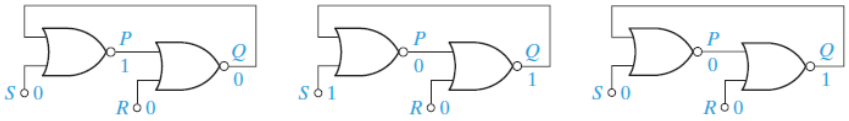
- The output of one of the gates is connected back into the input of another gate in the circuit so as to form a closed loop
- A feedback loop with an inverter



- A feedback loop with two inverters

S-R Latch (Set-Reset Latch)

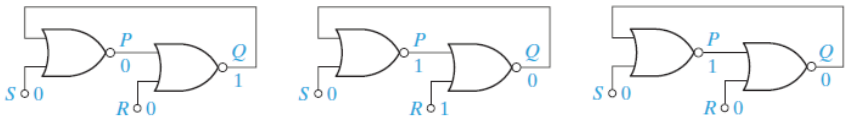
- A feedback loop with two NORs



- Initially, $P=1$, $Q=0$
- $S=0$, $R=0$
  - The circuit remains stable with $P=1$ and $Q=0$
- $S=1$, $R=0$
  - P will become 0, and then Q will become 1
- $S=0$, $R=0$ (S is changed back to 0)
  - The circuit will not change the state

S-R Latch (Set-Reset Latch)

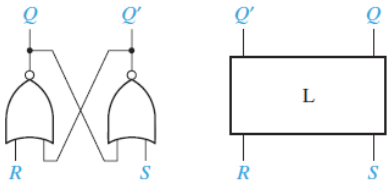
- Now $P=0$, $Q=1$



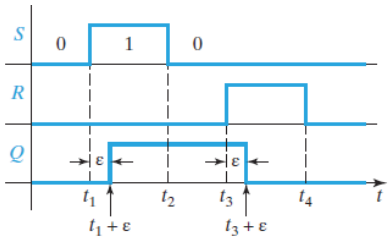
- $S=0$, $R=0$
  - The circuit remains stable with $P=0$ and $Q=1$
- $S=0$, $R=1$
  - Q will become 0, and then P will become 1
- $S=0$, $R=0$ (R is changed back to 0)
  - The circuit will not change the state
- Characteristics of S-R latch
  - The output depends not only on the present inputs, but also on the past sequence of inputs
  - P and Q are always complements $\Rightarrow P = Q'$

S-R Latch (Set-Reset Latch)

- Cross-coupled form of S-R latch

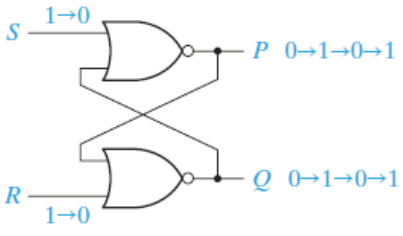


- Timing diagram



S-R Latch (Set-Reset Latch)

- $R=S=1$ is not allowed
  - Both P and Q are all 0's $\Rightarrow P \neq Q'$
  - If both inputs change 1 to 0 simultaneously, the latch may continue to oscillate



S-R Latch (Set-Reset Latch)

- *Present state* $Q(t)$
  - The state of the Q output at the time any input signal changes
- *Next state* $Q(t+\varepsilon)$ (or $Q(t+1)$)
  - The state of the Q output after the latch or flip-flop has reached to the input change and stabilized
- *Next-state equation*

Present State Q	Next State Q^+ SR 00	SR 01	SR 11	SR 10
0	⓪	⓪	⓪	1
1	①	0	0	①

The stable states are circled

$$Q(t+\varepsilon) = R(t)'[S(t) + Q(t)]$$
$$= R(t)'S(t) + R(t)'Q(t)$$
$$Q^+ = R'S + R'Q$$

S-R Latch (Set-Reset Latch)

- *Characteristic equation*
  - Takes into account any disallowed input equation ($S=R=1$)
  - Not necessarily the same as the next-state equation

Present State Q	Next State Q^+ SR 00	SR 01	SR 11	SR 10
0	⓪	⓪	X	1
1	①	0	X	①

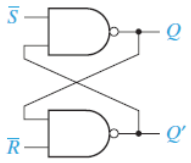
RQ \ S	0	1
00	0	1
01	1	1
11	0	X
10	0	X

S	R	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	–	Inputs not allowed
1	1	1	–	

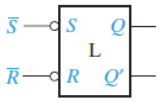
$$Q^+ = S + R'Q \qquad (SR = 0)$$

$\overline{S}$-$\overline{R}$ Latch using NAND Gates

- Alternative form of S-R Latch
  - Uses NAND gates
- Circuit representation
  - Set Q to 1 when $\overline{S} = 0$
  - Reset Q to 0 when $\overline{R} = 0$



- Symbol representation



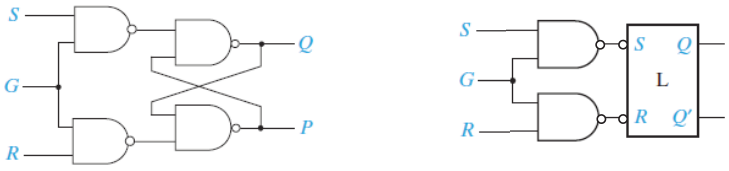
$\overline{S}$-$\overline{R}$ Latch using NAND Gates

- Truth table representation

$\overline{S}$	$\overline{R}$	Q	Q^+	
1	1	0	0	
1	1	1	1	
1	0	0	0	
1	0	1	0	
0	1	0	1	
0	1	1	1	
0	0	0	–	Inputs not allowed
0	0	1	–	

Gated S-R Latches

- *Gate* or *enable* input
  - *Gated latches* have an additional input called the *gate* or *enable* input
  - When the gate input is inactive, the state of the latch cannot change
  - When the gate input is active, the latch operates in an ordinary way
- NAND-gate gated S-R latch



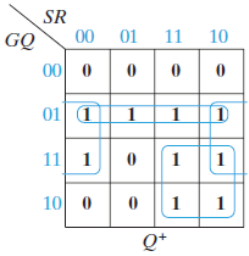
Gated S-R Latches

- Next state table

Present State	Next State Q^+							
	$G = 0$				$G = 1$			
	SR 00	SR 01	SR 11	SR 10	SR 00	SR 01	SR 11	SR 10
0	⓪	⓪	⓪	⓪	⓪	⓪	1	1
1	①	①	①	①	①	0	①	①

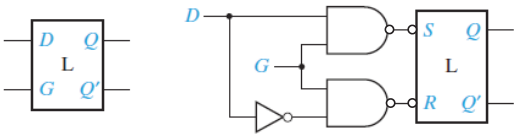
- Next-state equation

$$Q^+ = SG + Q(R' + G')$$

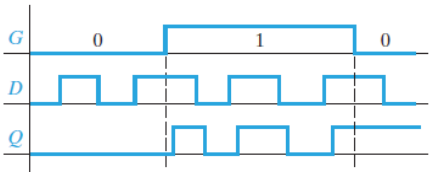


Gated D Latch

- Transparent latch
  - Q becomes equal to D while G is active



- Timing diagram



Gated D Latch

- Next-state equation / Characteristic equation

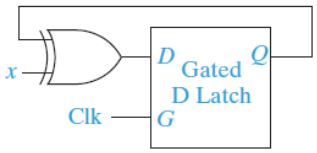
G D Q	Q^+
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	0
1 1 0	1
1 1 1	1

Q \ GD	00	01	11	10
0	0	0	1	0
1	1	1	1	0

$$Q^+ = G'Q + GD$$

Problems with the Latches

- Example circuit



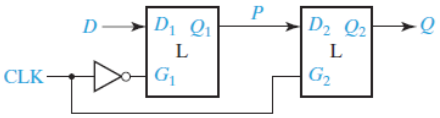
- The output changes any time
- The latch takes its input any time

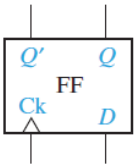
Flip-Flops

- *Flip-flop*
  - The **outputs** only change on a clock edge
- *Edge-triggered flip-flop*
  - The flip-flop take **inputs** on a clock edge
  - The **inputs** to the flip-flop only need to be stable for a short period of time around the clock edge
- *Master-slave flip-flop*
  - The flip-flop takes **inputs** during the clock period

Rising Edge-Triggered D Flip-Flop

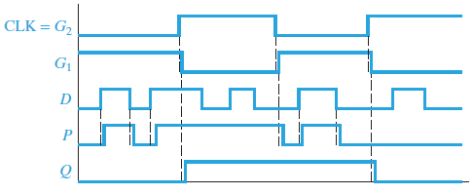
- The output can change on a 0 to 1 transition on the clock input
- Construction from two gated D latches





Rising edge-triggered =
positive edge-triggered

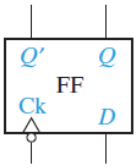
- Timing diagram



Falling Edge-Triggered D Flip-Flop

- The output can change on a 1 to 0 transition on the clock input
- Truth table representation

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

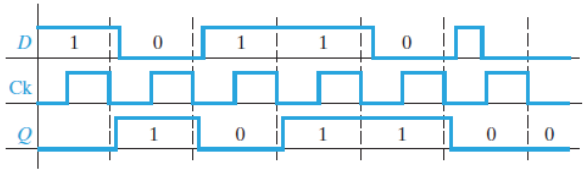


Falling edge-triggered =
negative edge-triggered

- Characteristic equation

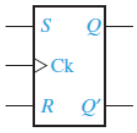
$$Q^+ = D$$

- Timing diagram

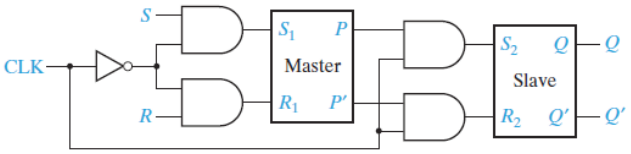


S-R Flip-Flop

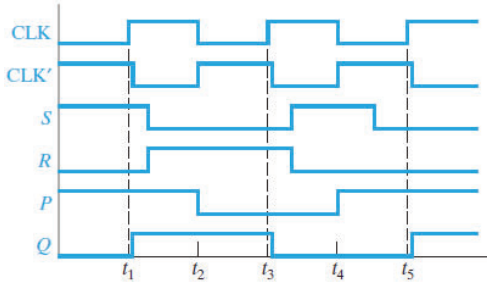
Symbolic representation



Implementation with two latches



Timing diagram

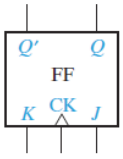


Characteristic equation

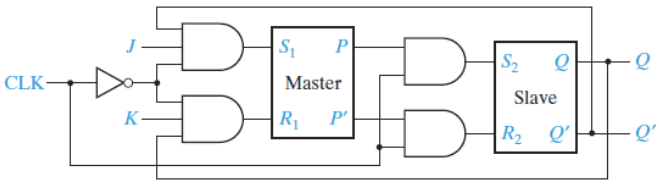
$$Q^+ = S + R'Q \; (SR = 0)$$

J-K Flip-Flop

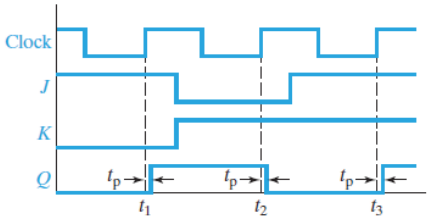
Symbolic representation



Implementation with two latches



Timing diagram



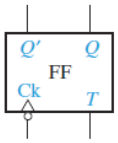
Characteristic equation

$$Q^+ = JQ' + K'Q$$

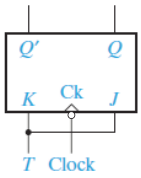
J K Q	Q^+
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	1
1 1 0	1
1 1 1	0

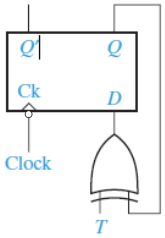
T Flip-Flop

Symbolic representation

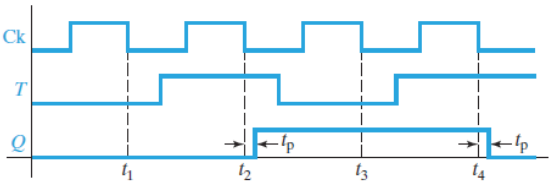


Conversion of *J-K*/*D* to *T*





Timing diagram



Characteristic equation

$$Q^+ = T \oplus Q = TQ' + T'Q$$

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

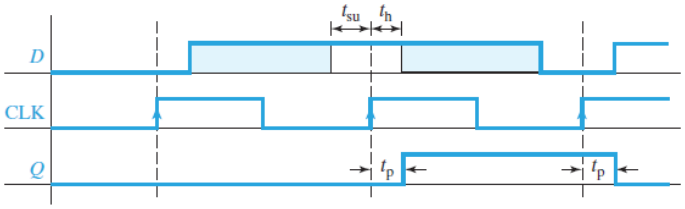
Characteristic Equations (Summary)

- Procedures to find a characteristic equation
  - Make a truth table that gives the next state (Q^+) as a function of the present state (Q) and the inputs. Any illegal input combinations should be treated as don't-cares.
  - Plot a map for Q^+ and read the characteristic equation from the map
- Characteristic equations: Summary

$Q^+ = S + R'Q \ (SR = 0)$	(S-R flip-flop)
$Q^+ = GD + G'Q$	(gated D latch)
$Q^+ = D$	(D flip-flop)
$Q^+ = D{\cdot}CE + Q{\cdot}CE'$	(D-CE flip-flop)
$Q^+ = JQ' + K'Q$	(J-K flip-flop)
$Q^+ = T \oplus Q = TQ' + T'Q$	(T flip-flop)

Setup and Hold Times

- Timing constraints of a flip-flop
  - A flip-flop changes state only on the active edge of the clock
  - The *D* input to an edge-triggered flip-flop *must* be held at a constant value for a period of time *before* and *after* the active edge of the clock



Propagation time (t_p):
the amount of time from the clock edge to the output change

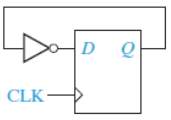
Setup time (t_{su}):
the amount of time that *D* must be stable before the active edge

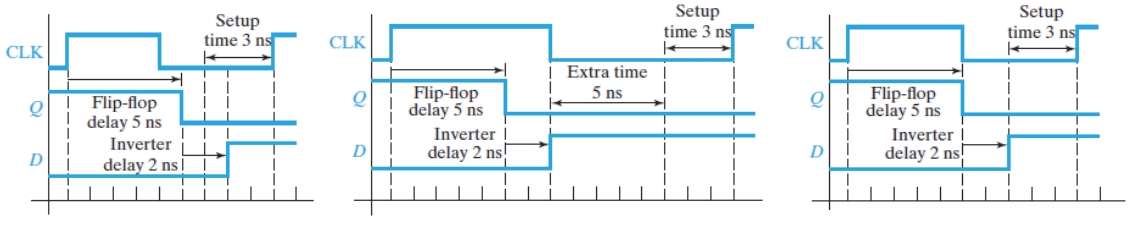
Hold time (t_h):
the amount of time that *D* must hold the same value after the active edge

Minimum Clock Period

- Why minimum clock period?

- Example case
  - Propagation delay of an inverter: 2 ns
  - Propagation delay of a flip-flop: 5 ns
  - Setup time of a flip-flop: 3 ns

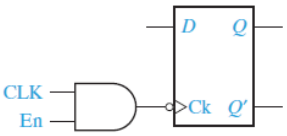




Setup time not satisfied | Setup time satisfied | Minimum clock period = 10 ns

D Flip-Flop with Clock Enable

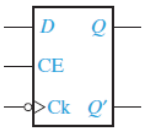
- How to hold existing data with changing data inputs
  - Even though the data input to the flip-flop may be changing, we want some flip-flops to hold existing data
- One solution



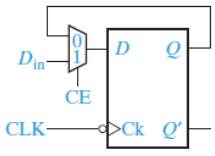
- Two potential problems with the above solution
  - Gate delay may cause the clock to arrive at different times
  - The flip-flop may trigger due to the change in En instead of the clock

D Flip-Flop with Clock Enable

- Symbolic representation



- Implementation using a MUX



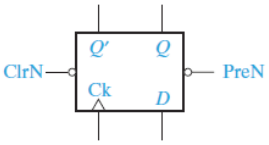
- Characteristic equation

$$Q^+ = D = Q \cdot CE' + D_{in} \cdot CE$$

D Flip-Flop with Clear and Preset

- Additional inputs
  - Can be used to set the flip-flops to an initial state independent of the clock
  - Asynchronous Clear (ClrN)
  - Asynchronous Preset (PreN)

- Symbolic representation



Ck	*D*	PreN	ClrN	Q^+
x	x	0	0	(not allowed)
x	x	0	1	1
x	x	1	0	0
↑	0	1	1	0
↑	1	1	1	1
0,1,↓	x	1	1	*Q* (no change)

D Flip-Flop with Clear and Preset

- Timing diagram