Digital Systems Design and Laboratory
[11. Latches and Flip-Flops]

Chung-Wei Lin

cwlin@csie.ntu.edu.tw

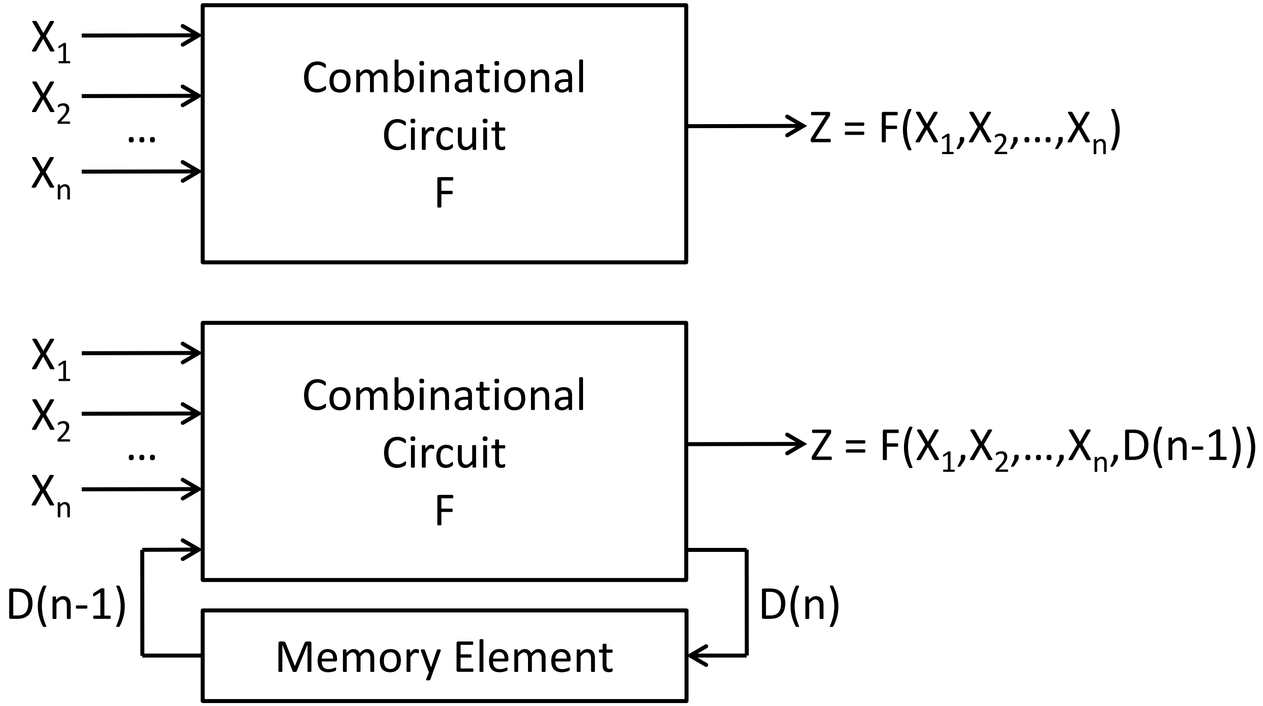
CSIE Department

National Taiwan University

We would like to thank Prof. Hui-Ru Jiang (NTU) for contributing the lecture material

Recap: Two Types of Switching Circuits

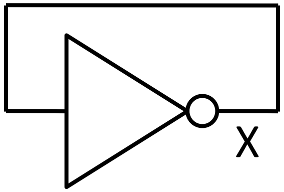
- Combinational circuits (memoryless)
  - Outputs depend only on **present** inputs
- Sequential circuits
  - Outputs depend on both **present and past** inputs
  - In general, sequential circuits = combinational circuits + memory

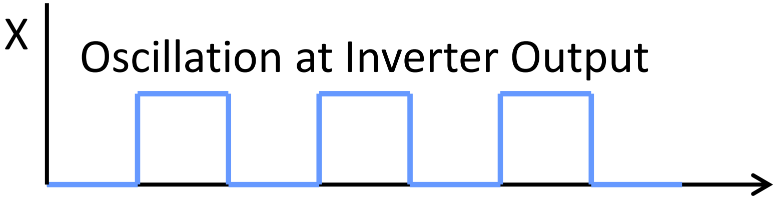


How to Remember the Past?

❑ Feedback

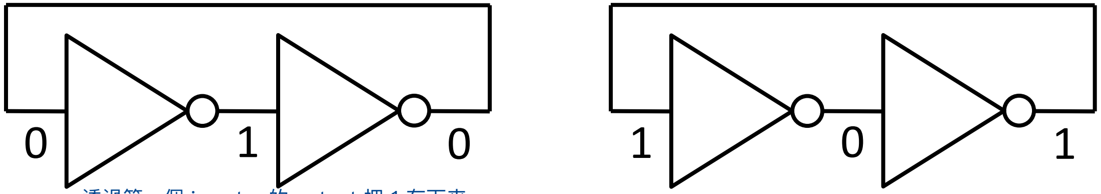
- The output of one of the gates is connected back into the input of another gate in the circuit so as to form a closed loop
- Example: inverter with feedback
  - Q: How fast does the circuit oscillate?
  - A: Determined by the propagation delay of the inverter



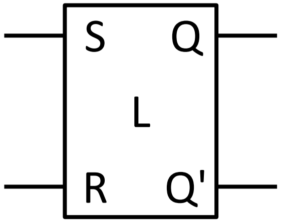


- Example: a feedback loop with two inverters
  - Two stable states
  - Latch: basic memory unit (store 1 bit)

存哪個 bit 看是看哪個點



透過第一個 inverter 的 output 把 1 存下來
透過第二個 inverter 的 output 把 0 存下來



Outline

❑ **Set-Reset Latch**

❑ Gated D Latch

❑ Edge-Triggered D Flip-Flop

❑ S-R Flip-Flop

❑ J-K Flip-Flop

❑ T Flip-Flop

❑ Flip-Flops with Additional Inputs

- SR latch 在說 S 是 force set to 1 代表的是 force Q = 1

- 因為 latch 和 flip flop 的差異在於 latch 沒有 clock，所以 latch 的值可以不斷地隨著 input 改變而改變（不像 flip flop 需要根據是 positive / negative edge trigger ，等 positive / negative edge 發生時才改變值）所以我們不會像 flip flop 裡一樣看到有 Q 和 Q+ (next state)，因為像 SR latch，他只會 hold current state（Q, Q’），除非 input force a change

- 即使 Q 和 Q’ 都是 current state ，在 SR latch 裡，Q’（Q）會接回去到 output 是接到 Q （Q’） 的 gate，這樣的接髮才能確保 S = R = 0 時 latch maintain current state（這就是 latch 的記憶功能）

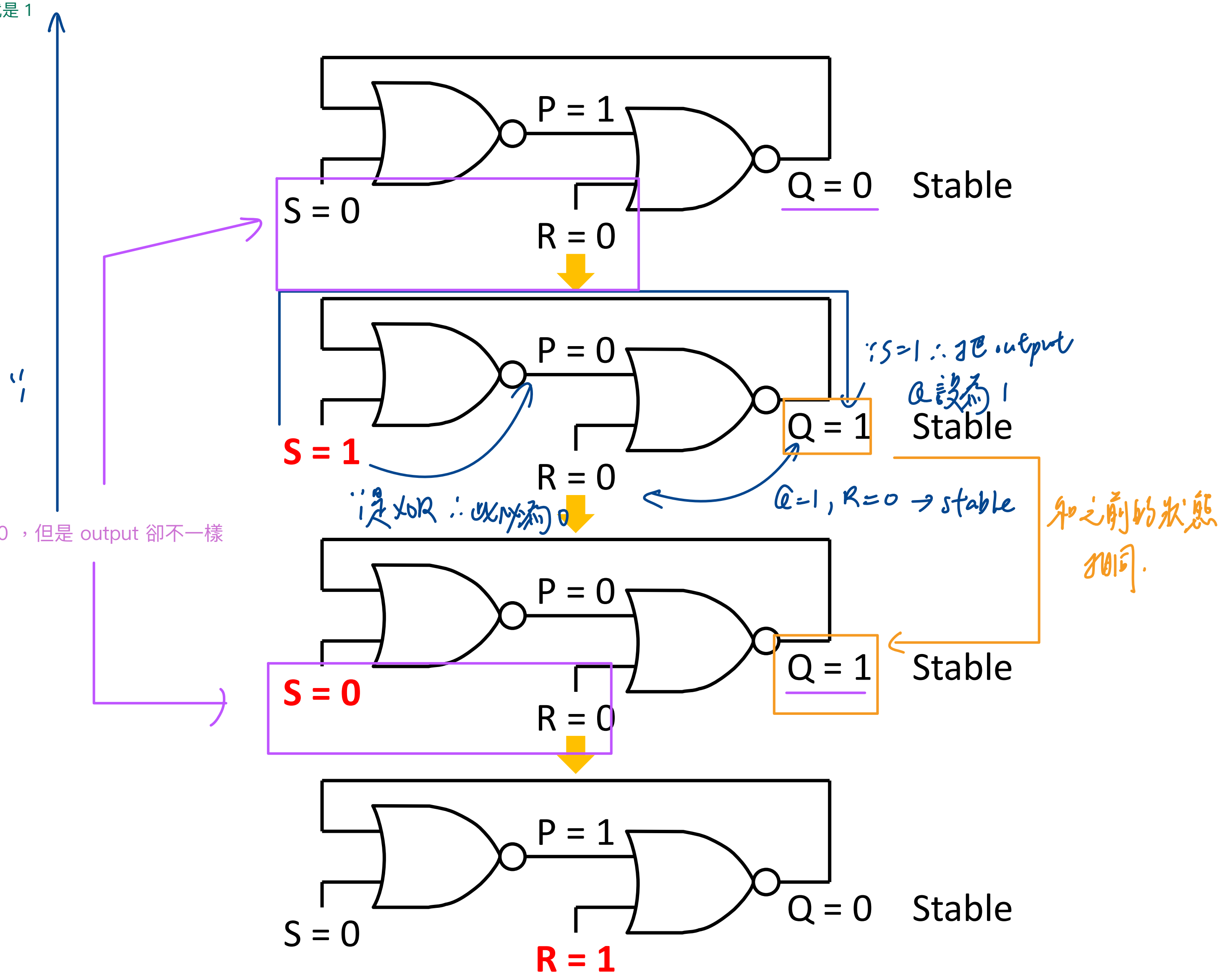
- 即使 we’re given Q 的值（還有 S, R） ，我們仍然不能直接從 Q 的值取 complement 來得到 Q’，因為 Q 和 Q’ 的值是同時被計算的（根據 S, R, feedback） ，Q 和 Q’ 的值會互相影響，所以不能直接 derive Q’ from Q （Q’ 只是 ideally NOT Q）

set：要把 output 的值設成 1
reset：要把 output 的值設成 0

>> S = R = 0 代表維持過去的 state
過去如果是 1 output 就是 1

- S, R 不能同時是 1

Set-Reset (S-R) Latch (1/2)

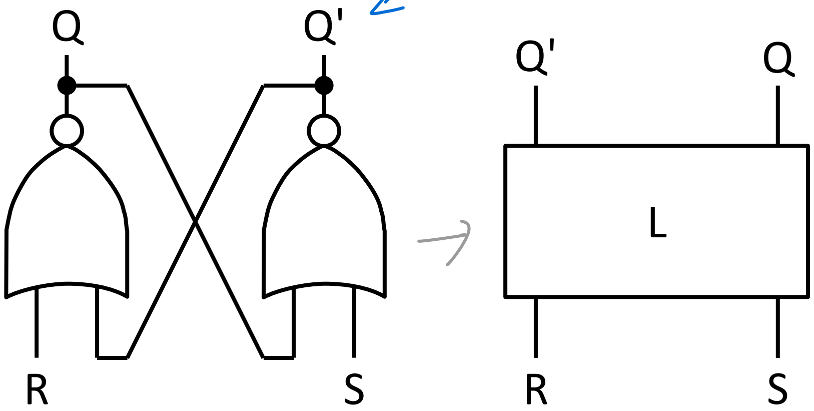


input 皆為 0, 0 ，但是 output 卻不一樣

Set-Reset (S-R) Latch (2/2)

如果 R = S = 1，因為經過 NOR gate ，所以
Q = Q’ = 0

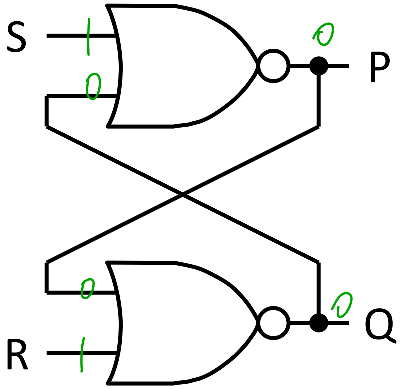
- ❑ Cross-coupled form



reset (Q=0)　　set (Q=1)

用L與長方形表示latch
(Latch)

- ❑ S = R = 1 not allowed!
  - ➢ Note that the outputs are Q and Q'
  - ➢ An oscillation scenario
    - Both S and R: 1 → 0
    - Both P and Q: 0 → 1 → 0 → 1
      - If the gate delays are equal



Next-State (Characteristic) Equation

❑ Operation

原本的state

新的state

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	X
1	1	1	X

Unchanged

∵ S=R=0 時

→ output 和前一個 state 相同

Reset to 0

R=1 代表 reset $Q^+=0$

Set to 1

S=1 代表 set $Q^+=1$

Inputs Not Allowed

S 和 R 不能同時是 1

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

$$Q^+ = \boxed{S} + \boxed{R'Q}$$

如果 S = 1
代表 output 設成 1

用 K-map 做出 characteristic equation

如果沒有 reset (R = 0)
代表 R’ = 1
因此 R’Q = Q，等同用原來的 state Q

S–R latch 的應用

Application: Switch Debouncing

- ❑ When a mechanical switch is opened or closed, switch contacts tend to vibrate before settling down
- ❑ Debounce with S-R latch
  - ➢ When the switch is flipped from a to b…
  - ➢ Work only with a "double throw" switch
    - Double throw: switch between two contacts
    - Single throw: switch between one contact and open

開關在開和關時會有一段時間搞不清楚是開的還是關的，所以訊號會跳來跳去，S–R latch 可以避免這個情形

一移到b時，S=1 → Q須設成1

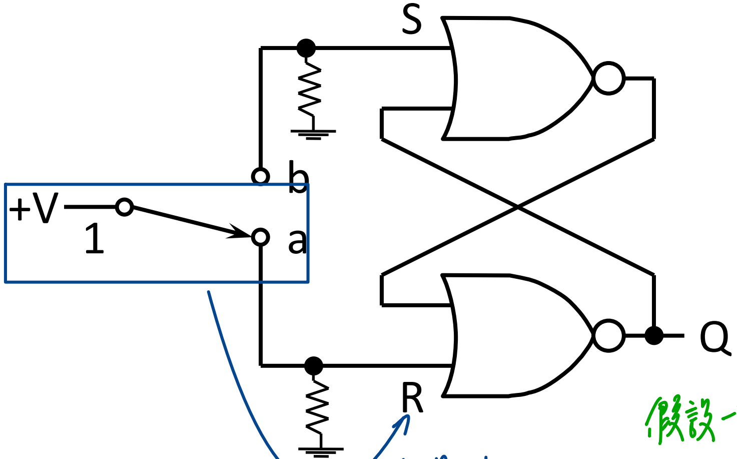
但不用擔心S的值震盪影響Q

∵ if S=1 → Q=1

S=0 → ∵ S=R=0 ∴ Q維持原狀態 → Q=1

在這段期間 R=S=0

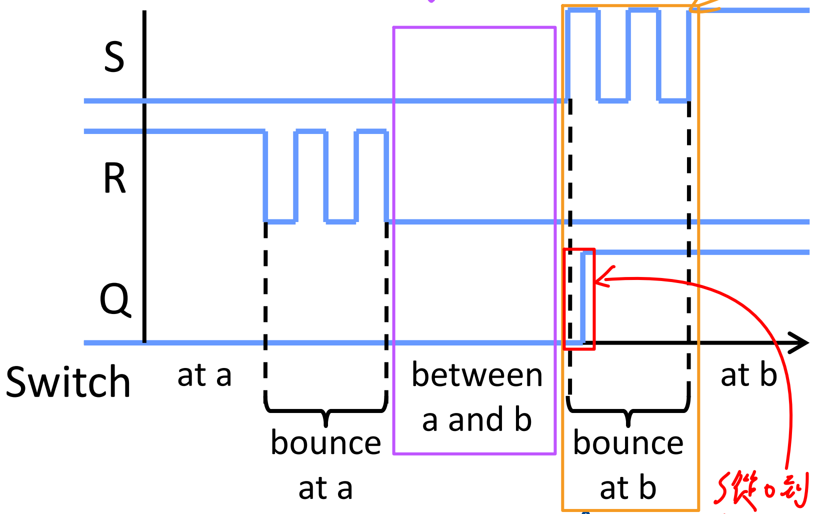
→ 維持原狀態 ∴ Q=0



∴ R=1

reset

假設一開始 Q=0



↑把switch從a移開

→ R的值無法確定

S從0到1

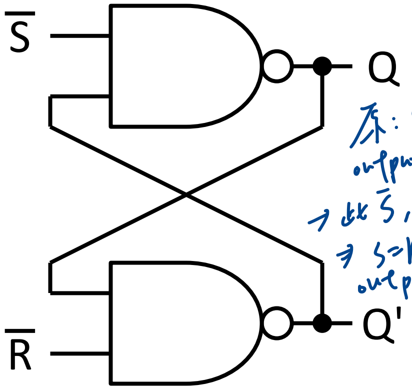
→ 有小小的 propagation delay

∴ Q一陣子後接 1

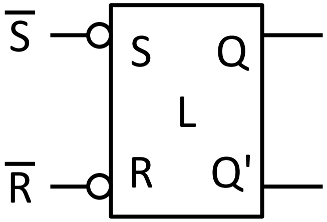
Alternative Form with NAND-Gates

→ 但不影响 ∵ 若 [R=1 → Q=0 (∵reset)
R=0 → ∵S=0 ∴ 维持原 state → Q=0
↓
∴ 无论R的值如何跳動.
→ Q=0

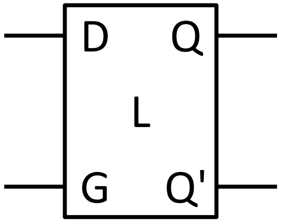
❑ $\overline{S}$-$\overline{R}$ latch: active-low inputs for S & R



原: S=R=0时
output 不变
→ 此 $\overline{S}$, $\overline{R}$
→ S=R=1 时
output 不变.



$\overline{S}$	$\overline{R}$	Q	Q^+	
1	1	0	0	Unchanged
1	1	1	1	
1	0	0	0	Reset to 0
1	0	1	0	
0	1	0	1	Set to 1
0	1	1	1	
0	0	0	X	Inputs Not Allowed
0	0	1	X	



Outline

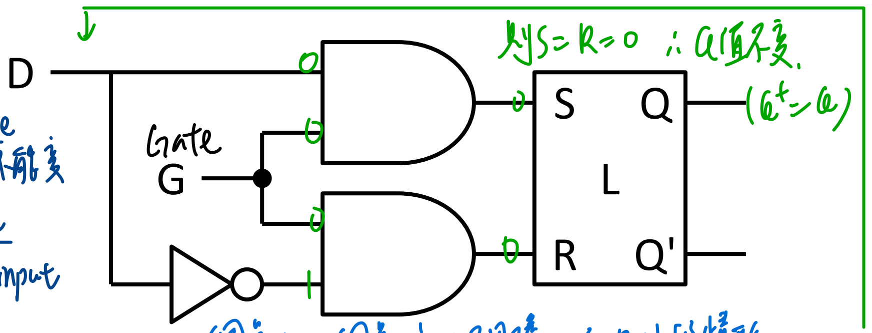
- ❑ Set-Reset Latch
- ❑ **Gated D Latch**
- ❑ Edge-Triggered D Flip-Flop
- ❑ S-R Flip-Flop
- ❑ J-K Flip-Flop
- ❑ T Flip-Flop
- ❑ Flip-Flops with Additional Inputs

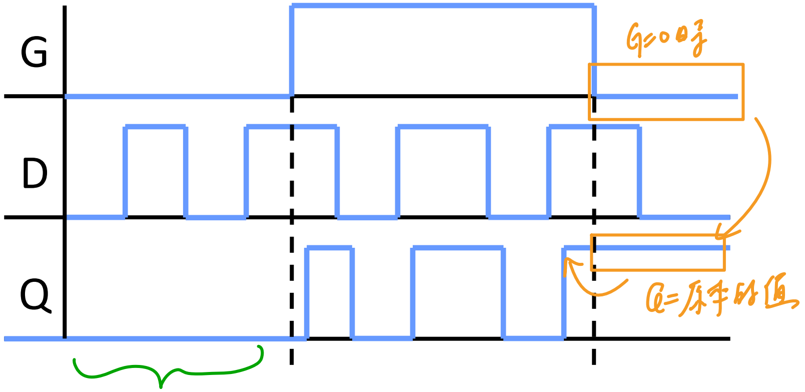
S-R latch 再加一些 gate，input 變 D, G.

Gated D Latch

Gate : inactive
→ latch state 不能變

Gate : active
→ latch 由其它 input 控制.





If D=G=0

∵一個是D 一個是D' ∴不用擔心 S=R=1 的情形

G	D	Q	Q^+	
0	0	0	0	Unchanged $Q^+ = Q$
0	0	1	1	
0	1	0	0	
0	1	1	1	
1	0	0	0	$Q^+ = D$
1	0	1	0	
1	1	0	1	
1	1	1	1	

G=0時
∵兩個 AND gate output 皆為0
∴ S=R=0
→ Q+=Q

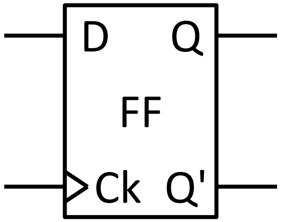
G=1時
兩個 AND gate output
→ depend on D

DQ \ G	0	1
00	0	0
01	1	0
11	1	1
10	0	1

$Q^+ = \boxed{G'Q} + \boxed{GD}$

G=0時 output depends on Q

G=1時是D



Outline

- ❑ Set-Reset Latch
- ❑ Gated D Latch
- ❑ **Edge-Triggered D Flip-Flop**
- ❑ S-R Flip-Flop
- ❑ J-K Flip-Flop
- ❑ T Flip-Flop
- ❑ Flip-Flops with Additional Inputs

Flip flop和latch最大不同：Flip-flop的output只有在特定時候會改變

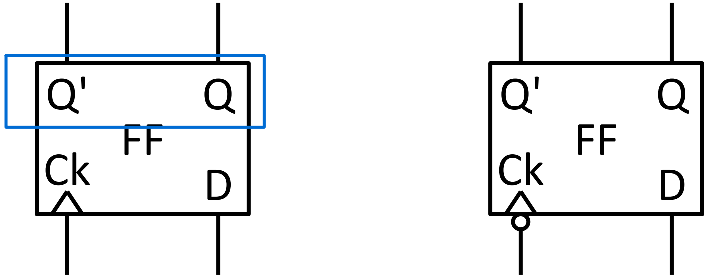
clock edge = clock signal

從0→1
從1→0

Edge-Triggered D Flip-Flops (1/2)

- ❑ Output changes are aligned with clock edges
  - ➢ Positive (rising-edge) trigger 0→1
  - ➢ Negative (falling-edge) trigger 1→0 (clock從1變0時，Q的值變成D)

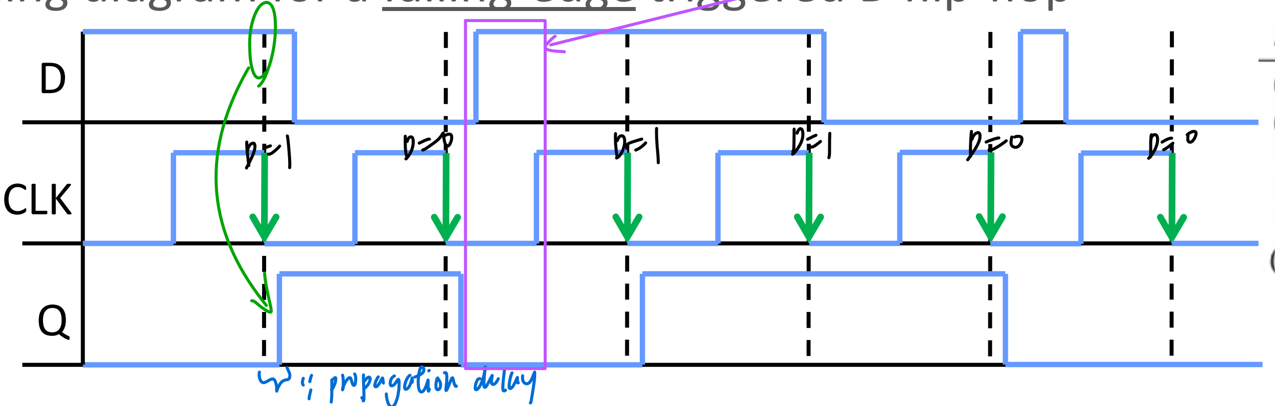
必是一個為0。一個為1



Rising-Edge Trigger   Falling-Edge Trigger

- ➢ Next-state (characteristic) equation: $\boxed{Q^+ = D}$
- ➢ Timing diagram for a falling-edge triggered D flip-flop

D改變但Q卻不變 ∵沒有falling edge



看到clock 1→0時
去看當時的D值=1
∴經過一點點delay
Q=1

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

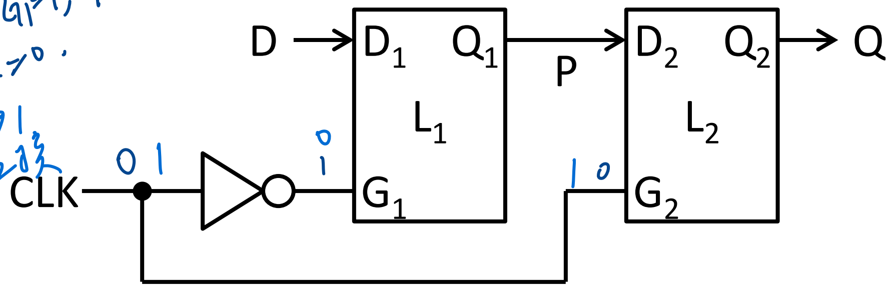
(c) Truth table

Edge-Triggered D Flip-Flops (2/2)

- ❑ Construct a rising-edge triggered D flip-flop from 2 gated D latches

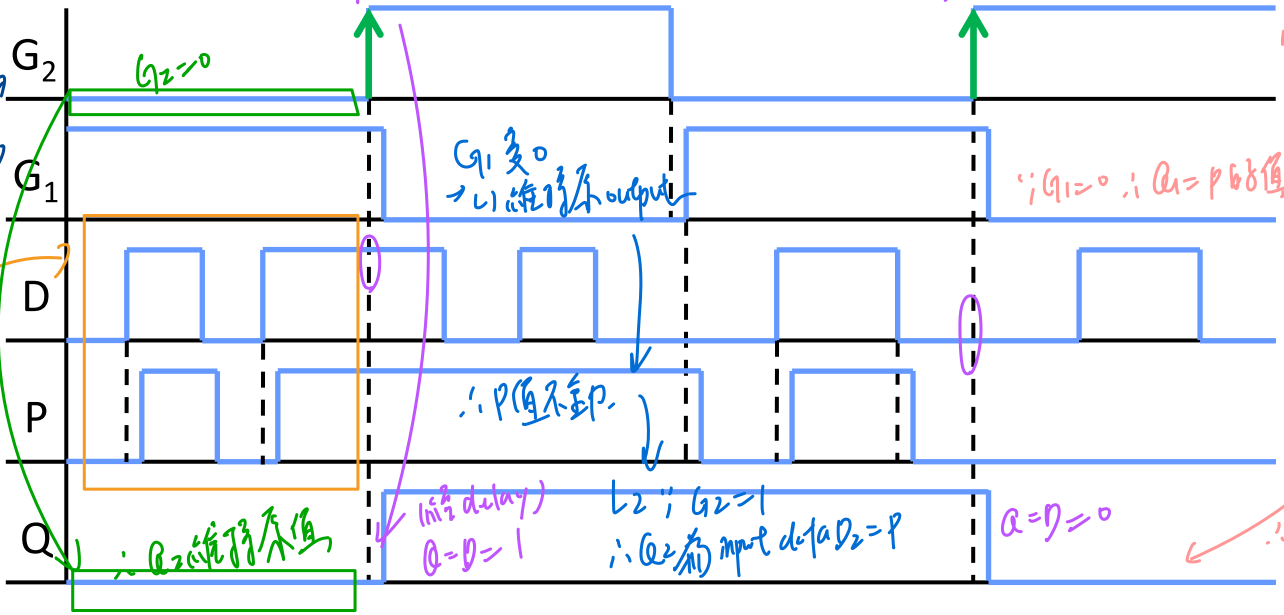
一開始, G1=1, G2=0,
代表CLK=0.

當CLK: 0→1
L1, L2角色互換



有一個input是D (data)
另一個input是gate
if gate=0
→ Q維持和之前一樣
gate=1
→ Q output為input data D

- ❑ Time diagram



G2: 0→1 (rise edge)

第2次rising edge

G2=0

不致相反
差一點propagation delay

∵G1=1
∴Q1=P=D
只是有微小delay

G1變0
→L1維持原output

∴P值不動

(含delay)
Q=D=1

L2: G2=1
∴Q2為input data D2=P

∴Q2維持原值

雖然此時G2=1
代表L2的output
Q2隨D變動.

∵G1=0 ∴Q1=P的值為原值

Q=D=0

但D2=P固定
∴Q=P固定為原值

Setup Times and Hold Times

(看是 rising / falling edge triggered)

在 active edge 時，Q=D，但考慮 propagation delay → 有可能 Q 的值非 rising edge 時 D 的值

∴ 須 setup time, hold time → D 須在 active edge 前後的一段時間是固定的值

↓

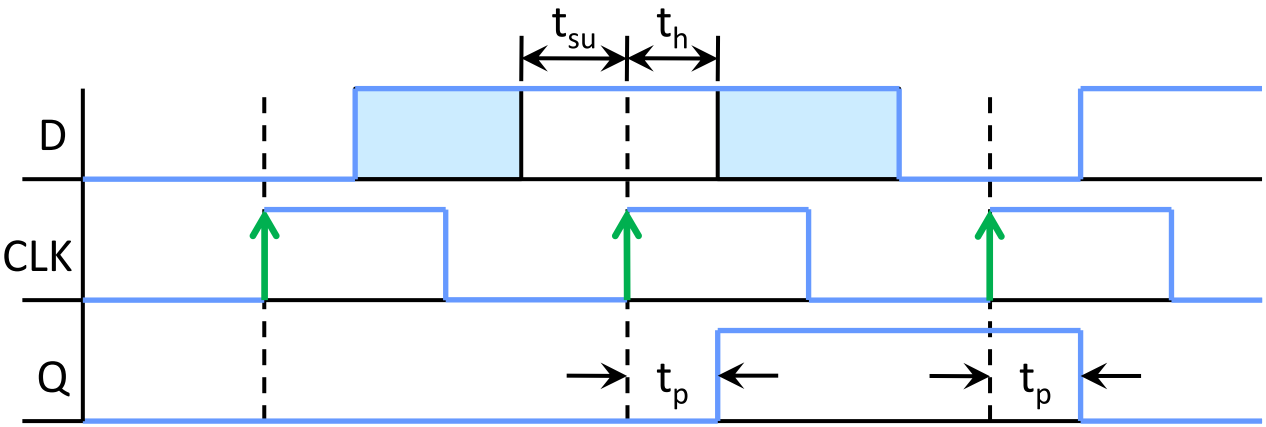
來確保 Q 的值是正確的 D 值

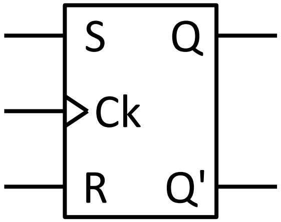
active edge 前

- ❑ Setup time
  - ➢ The amount of time that D must be stable before the active edge

- ❑ Hold time
  - ➢ The amount of time that D must hold the same value after the active edge

active edge 後.





Outline

- ❑ Set-Reset Latch
- ❑ Gated D Latch
- ❑ Edge-Triggered D Flip-Flop
- ❑ **S-R Flip-Flop**
- ❑ J-K Flip-Flop
- ❑ T Flip-Flop
- ❑ Flip-Flops with Additional Inputs

flipflop: state 只在 clock edge 改变
⇕ (latch 和 flipflop 差别)
latch: state 在 C=1 且 input 改变时
改变.

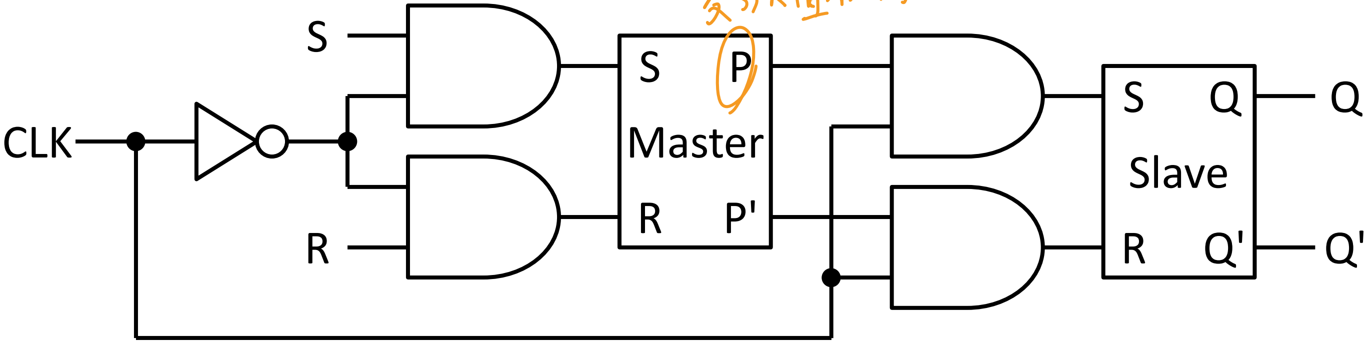
∵是 flip-flop ∴ output Q 在 active edge 改变

S-R Flip-Flop

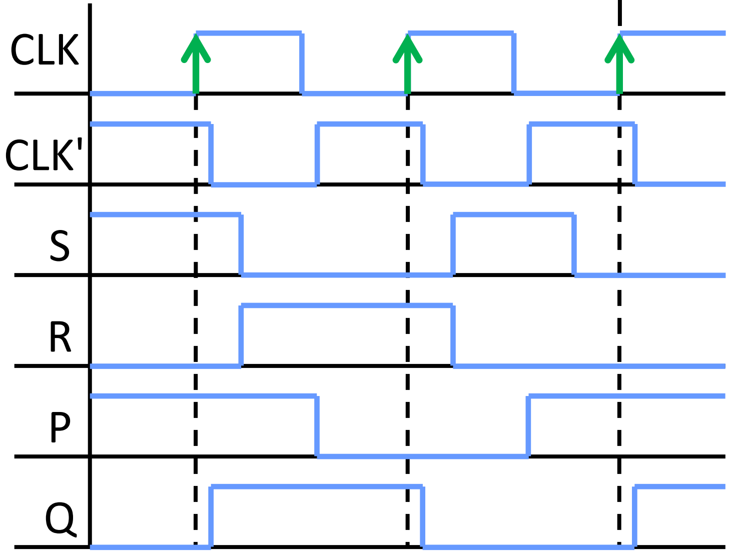
- ❑ Output changes at rising edges
- ❑ Construct from 2 latches ↙ 用 S-R latch

Not desired --- should only allow the S and R inputs to change while the clock is high

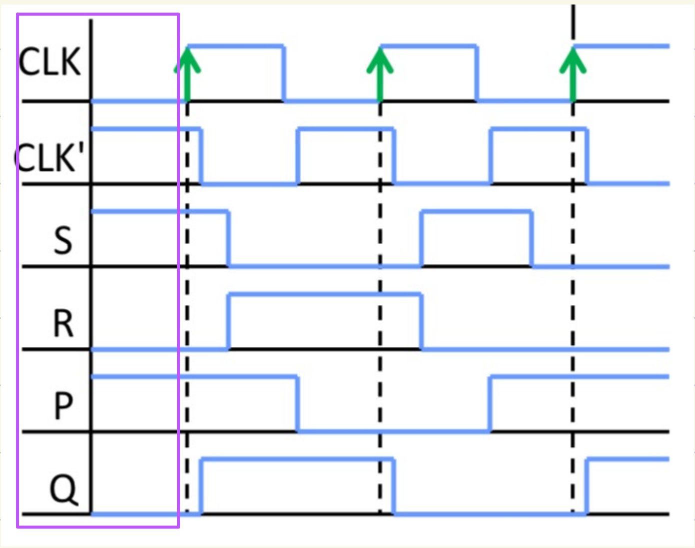
受 S, R 值影响.



S	R	Operation
0	0	No state change
0	1	Reset Q to 0 (after active CLK edge)
1	0	Set Q to 1 (after active CLK edge)
1	1	Not allowed



rising edge → output 改變



一開始 S=1, R=0

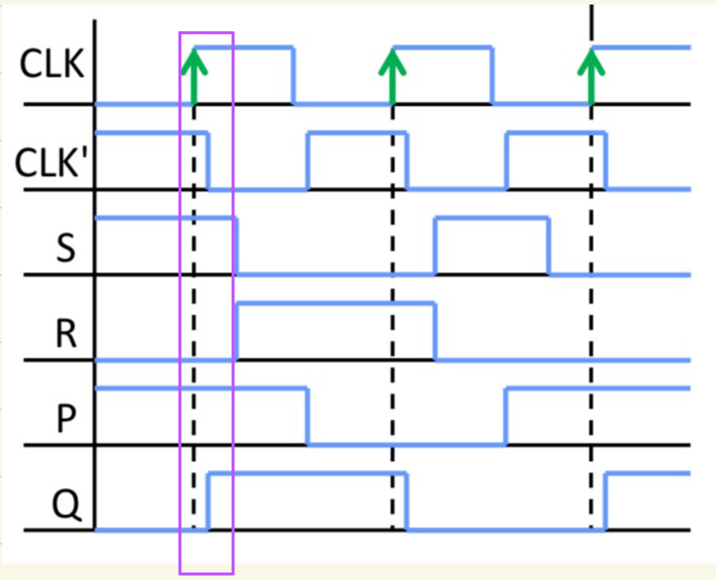
→ P=1

此時 CLK=0

∴ Q不會改變.

在 active edge 之前

→ Q被鎖定, P可跟著 input 動.



rising edge 發生時

→ S=R=0

∴ P的值為原值 → P=1

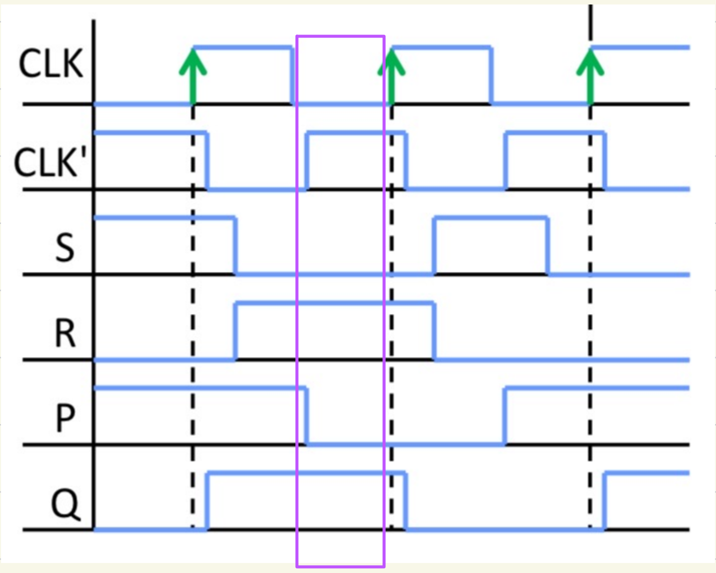
S=1, R=0

Q的值可跟著P動.

→ 在 propagation delay 後 Q=1

active edge 一發生

→ Q 就可跟著P動.



clock是0時,

→ slave 前面2個 AND gate output=0

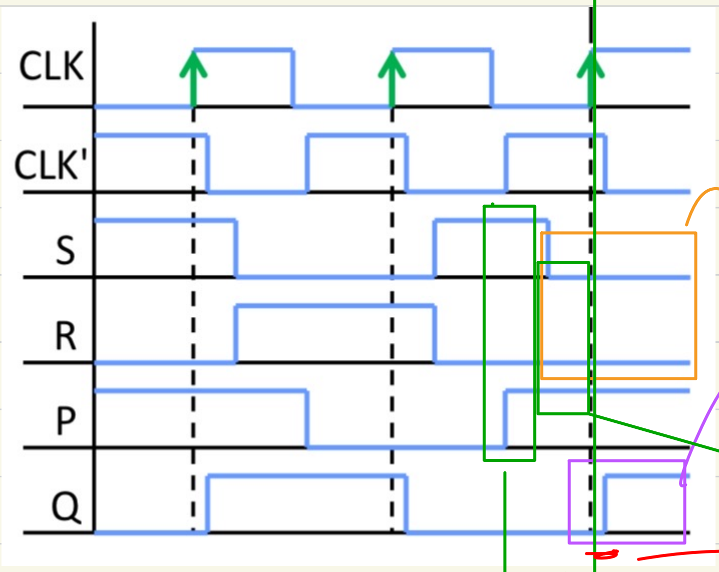
∴ S=R=0

→ Q的值被鎖定

但同時P的值可根據S,R的值改變

→ 此時 S=0, R=1 ∴ P=0

(reset to zero)



① S=1, R=0 ∴ set P=1

② 接著 S: 1→0, R=0
∴ P維持原值=1

S=R=0 → 理論上: 沒有 state change
但Q: 0→1

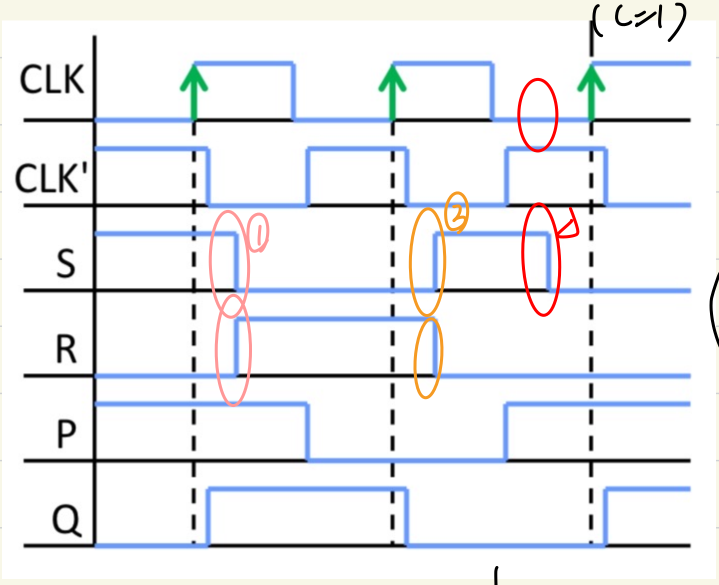
③ 直到 rising edge 發生 → Q變成P ∴ Q=1
→ 違反 S-R flip flop S=0, R=0時值不變.

S=R=0
但Q的值改變了

∴ 須限制 S, R只有在 clock=high (CLK=1) 時做改變,

前面兩次 ① ② 沒問題 ∵ S, R在 clk=high 時就做改變了, 在 C=0 時不做改變.

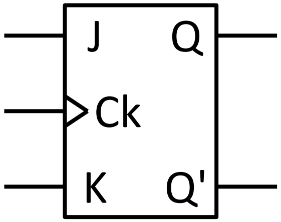
(C=1)



△ 發生問題的原因:
在 C=0 時改變 S: 1→0
此時 S=R=0 ∴ P的值不變.
但前面P的值已經在 S=1, R=0時設成 P=1
∴ P維持1

→ Q未受到 S=1, R=0 的改變. ∵ active edge 未到.
(S=1, R=0時Q應在 active edge後設成1)
但S又變成0 ∴ S=R=0 → Q不變

⇒ S=1, R=0 → Q=1 沒有被反應在Q上,



Outline

- ❑ Set-Reset Latch
- ❑ Gated D Latch
- ❑ Edge-Triggered D Flip-Flop
- ❑ S-R Flip-Flop
- ❑ **J-K Flip-Flop**
- ❑ T Flip-Flop
- ❑ Flip-Flops with Additional Inputs

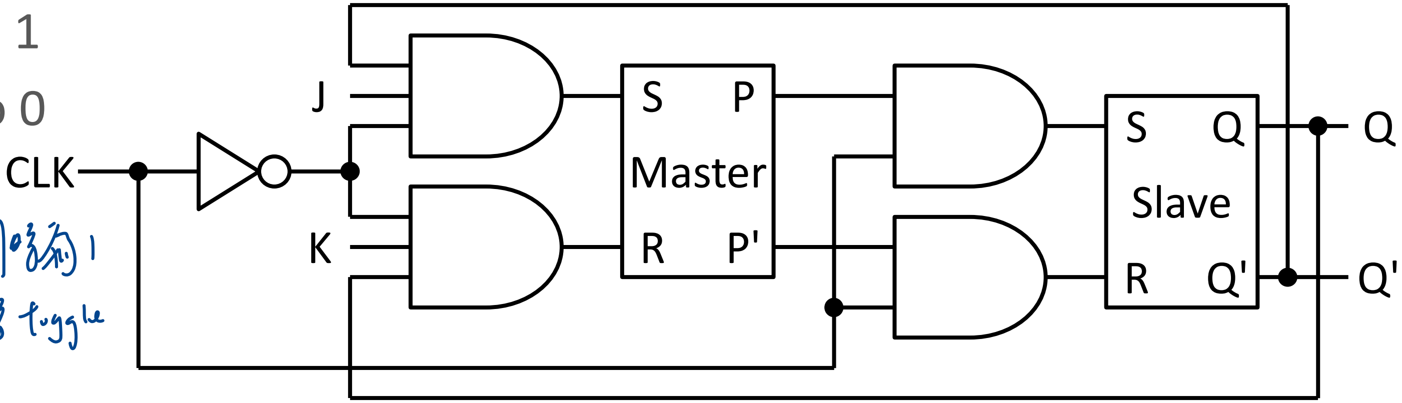
J-K Flip-Flop

❑ Extension of S-R flip-flop

類似S ➢ J: jump to 1

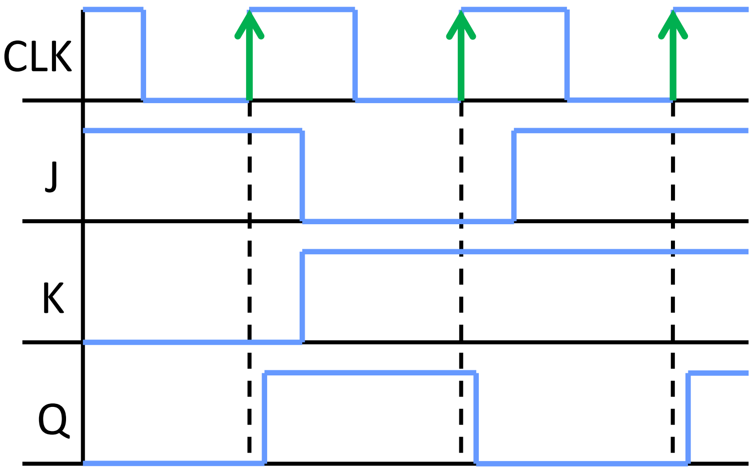
類似R ➢ K: clear to 0

→ JK, SR 不同:
S-R flip flop S,R 不能同時為1
但 J-K 可以, J=K=1 時 toggle

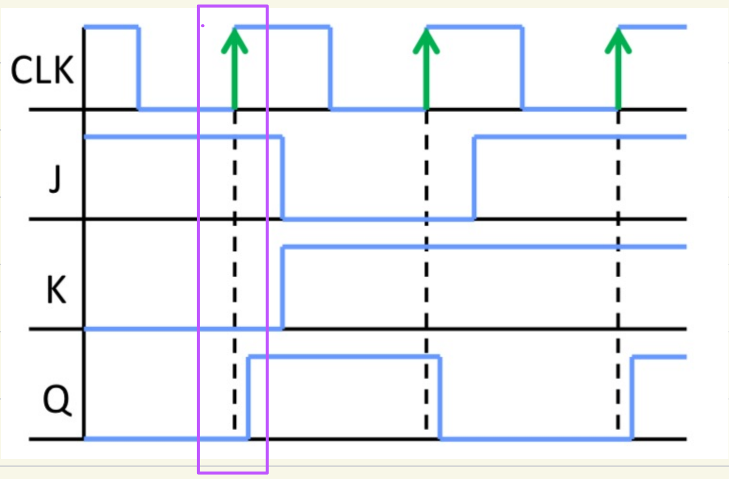


J (似S)	K (似R)	Q	Q^+	
0	0	0	0	Unchanged
0	0	1	1	
0	1	0	0	Clear to 0
0	1	1	0	
1	0	0	1	Jump to 1
1	0	1	1	
1	1	0	1	Toggle (反轉)
1	1	1	0	

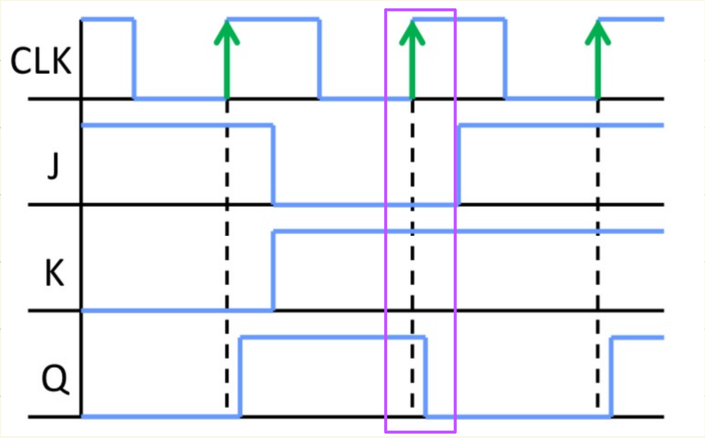
$$Q^+ = JQ' + K'Q$$



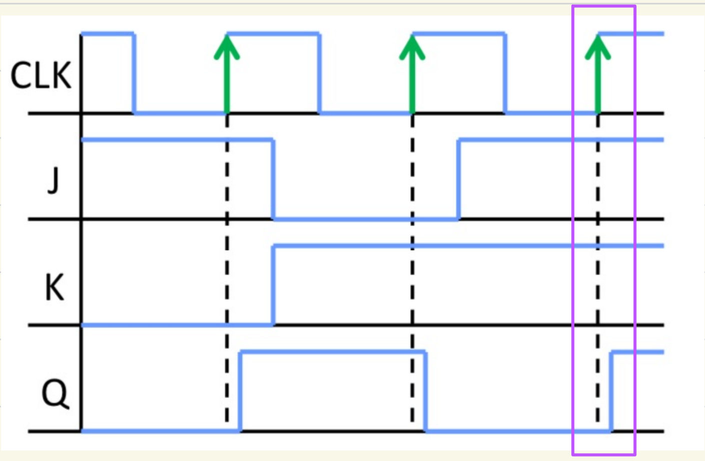
Rising-Edge Trigger



edge trigger 時，J=1, K=0
→ 代表 jump to 1 ∴ Q 從 0 變 1

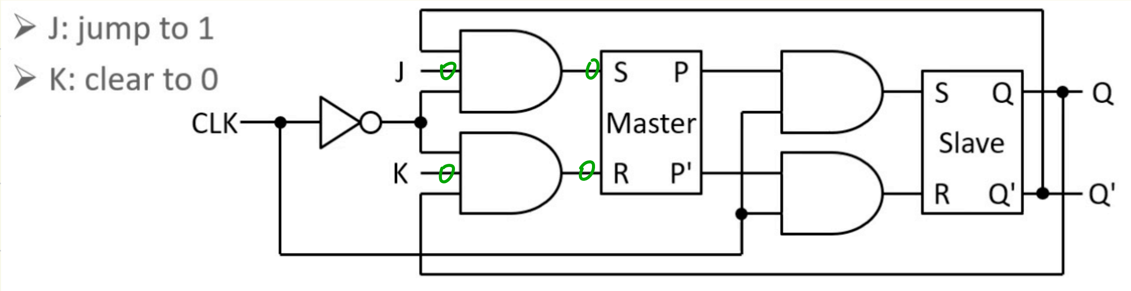


第二次 rising edge 時，J=0, K=1
→ 代表 clear to 0 ∴ Q 從 1 變 0



第三次 rising edge 時，J=K=1
→ 代表 toggle
∴ Q 原本=0，反轉變成 1

Case 1 : $\boxed{J=K=0}$

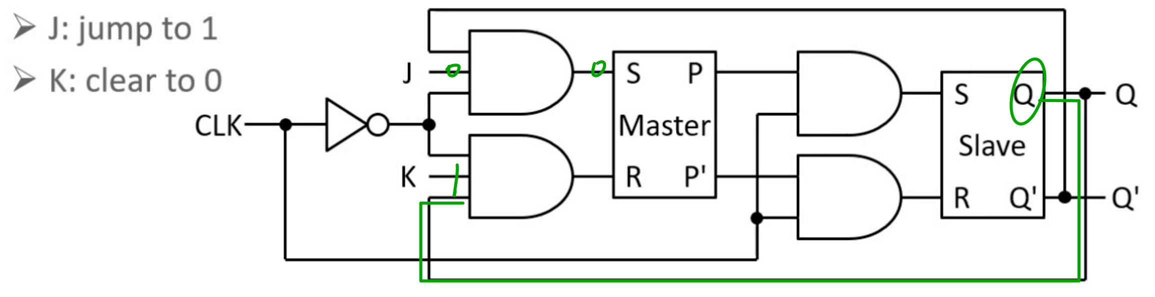


➢ J: jump to 1

➢ K: clear to 0

AND Gate output 皆為 0 ∴ $S=R=0$ → Q^+ 不動.

Case 2 : $\boxed{J=0, K=1}$



➢ J: jump to 1

➢ K: clear to 0

J的 AND Gate 永遠 $=0$, K的 AND Gate 看Q的值決定

若 $Q=0$

→ K的 AND Gate $=0$ ∴ $S=R=0$ → 不動.

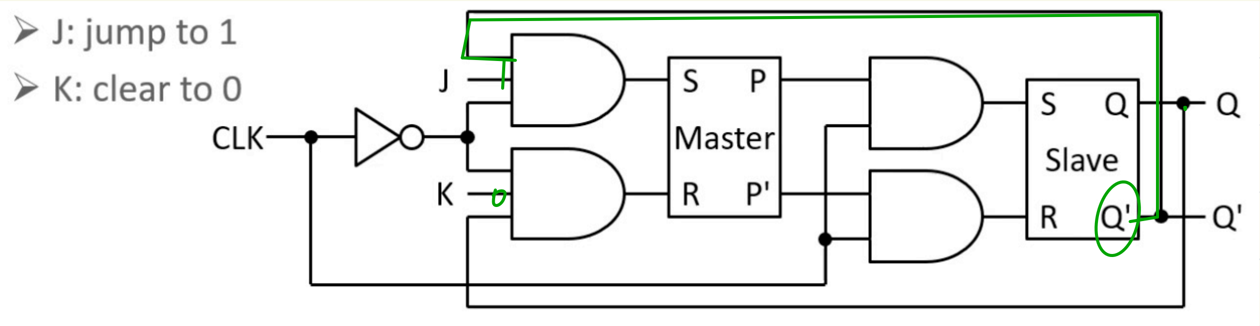
※假設了 $Q=0$ 的情形 ∴ Q^+ 維持 Q → $Q^+=0$

若 $Q=1$

→ K的 AND Gate $=1$ ∴ $R=1$ → reset to 0 → $Q^+=0$

∴無論 $Q=0$ or $Q=1$, 當 $J=0$, $K=1$ 時結果都是 0

Case 3 : $\boxed{J=1, K=0}$



➢ J: jump to 1

➢ K: clear to 0

K 的 AND Gate 永遠是 0，J 的 AND Gate 看 Q' 的值決定

If $Q'=1$

→ J 的 AND Gate output $=1$ → $S=1$ ∴ set to 1 → $Q^+=1$

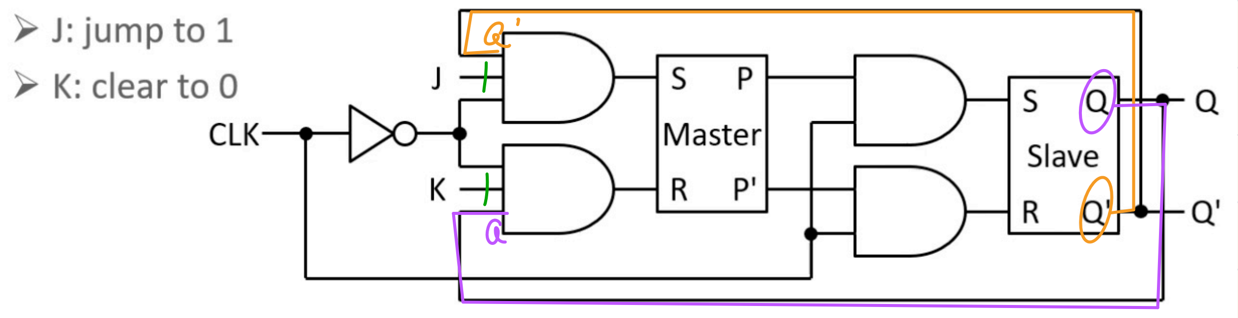
If $Q'=0$

→ J 的 AND Gate output $=0$ → $S=0$, $R=0$ (∵ $K=0$) → 不動

因為原本的 $Q=1$ ∴ Q^+ 保留 Q 的值 → $Q^+=1$

∴ 無論 $Q'=1$ or $Q'=0$，當 $J=1$, $K=0$ 時，Q^+ 的值都是 1

④ $\boxed{J=K=1}$



➢ J: jump to 1

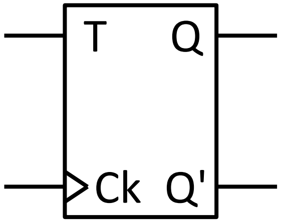
➢ K: clear to 0

If $\boxed{Q=1}$

→ K 的 AND Gate output $=1$ → $R=1$ (reset/成 0) → $\boxed{Q^+=0}$

If $\boxed{Q=0}$

→ ∵ $Q=0$ → $Q'=1$ ∴ J 的 AND Gate output $=1$ → $S=1$ (Set 成 1) → $\boxed{Q^+=1}$



Outline

- ❑ Set-Reset Latch
- ❑ Gated D Latch
- ❑ Edge-Triggered D Flip-Flop
- ❑ S-R Flip-Flop
- ❑ J-K Flip-Flop
- ❑ **T Flip-Flop**
- ❑ Flip-Flops with Additional Inputs

T Flip-Flop

toggle

- T: toggle T=0 不變，T=1 反轉

→ 只有T來控制 input

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

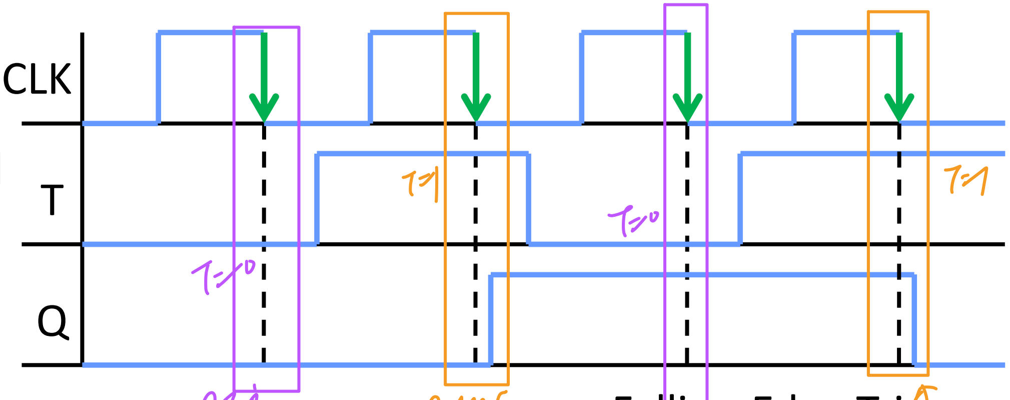
Unchanged

Toggle

XOR

$$Q^+ = \underline{TQ' + T'Q} = T \oplus Q$$

$$= JQ' + K'Q$$



Q不變 ∴Q保持0

Q相反 ∴Q:0→1

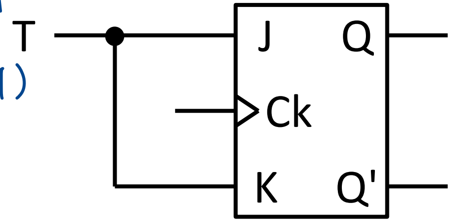
Falling-Edge Trigger

Q不變 ∴Q保持1

Q相反 ∴Q:1→0

- Implementations

作法1 (用JK→T)

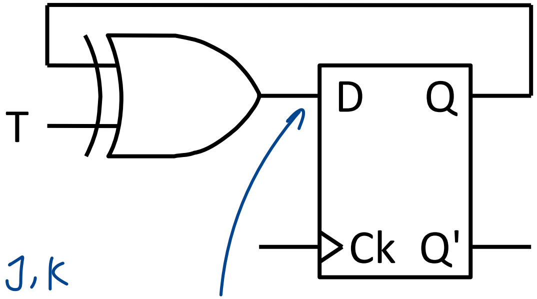


T flip flop：在原本的 J-K flip flop 的 J,K 都接到 T

→ T=0 ⇒ J=K=0 (不變)

T=1 ⇒ J=K=1 (反轉)

作法2 (用D flip-flop)



D input：$Q \oplus T$

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

Outline

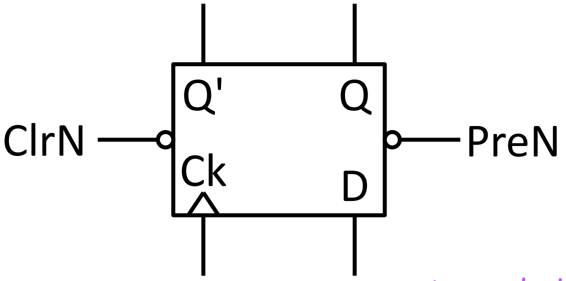
- ❑ Set-Reset Latch
- ❑ Gated D Latch
- ❑ Edge-Triggered D Flip-Flop
- ❑ S-R Flip-Flop
- ❑ J-K Flip-Flop
- ❑ T Flip-Flop
- ❑ **Flip-Flops with Additional Inputs**

Flip-Flops with Additional Inputs

如果 PreN, ClrN 一個0一個1
→ 強制把 Q^+ 設成某個值, (和 rising, falling edge 無關)

假設: rising edge flip flop.

- Set a flip-flop to an initial state independent of the clock
  - Example: asynchronous Clear and Preset



不允許 clear, preset 同時為0。

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 (Unchange)

rising edge trigger

Q^+ 值和D相同.

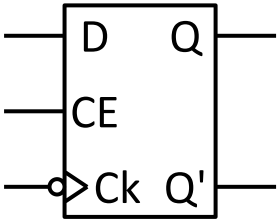
PreN=ClrN=1
→ D flip flop

- Let a flip-flop hold existing data even though the data input may be changing
  - Gated clock: gate the clock by Clock Enable (CE)

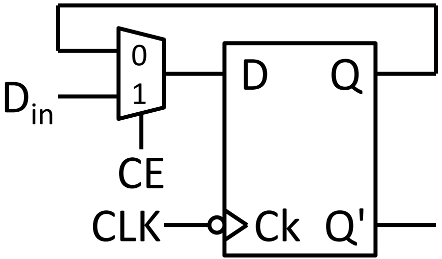
就算 input data 改變, 也不去改值

(假設 rising edge)
∵ D flip flop
∴ 只有在↑時可能去改變值
(變成D)

→ ↓ / 0 / 1 ⇒ $Q^+=Q$
(falling edge)
clock 維持在 0 or 1



$Q^+ = D(CE) + Q(CE)'$



clock enable 是1時 (CE=1)
→ 才允許 flip flop 做 D flip flop 做的事情

clock enable 是0時 (CE=0)
→ 不會做任何值的改變 (Q值保持)

Summary

Type	Q^+
S-R Latch	S + R'Q
Gated D Latch	G'Q + GD
D Flip-Flop	D
S-R Flip-Flop	S + R'Q
J-K Flip-Flop	JQ' + K'Q
T Flip-Flop	TQ' + T'Q
D-CE Flip-Flop	D(CE) + Q(CE)'

✗ S=R=1
S: set to 1
R: reset to 0

由G值決定 G=0 → Q不變
G=1 → 改D

給input D → next state就是D

似SR latch, 但只在rising / falling edge改變值

J=K=0 : 不變
J或K其中一個=1 改0/1
J=K=1 : 反轉

T=0 同 J=K=0 (不變)
T=1 同 J=K=1 (反轉)

flip-flop 只有在 0→1 / 1→0 時有值的改變、

只有在 CE=1 時 → 和D flip flop 運作相同
CE=0 時 → 保持原Q

Q&A