



UNIVERSITA' DEGLI STUDI DI PARMA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

# **Fondamenti di Informatica B**

---

Lezione n.5

# Fondamenti di Informatica B

---

## Lezione n.5

- **Circuiti Combinatori e Sequenziali**
- **Circuiti Ben Formati**
- **Introduzione ai Circuiti Sequenziali**
- **Elementi di Memoria**

In questa lezione verranno considerate le differenze tra circuiti combinatori e circuiti sequenziali e introdotti i primi elementi sui circuiti sequenziali

# Circuiti Combinatori e Sequenziali

I circuiti logici possono appartenere a due categorie:

- Circuiti Combinatori:

- Le uscite sono solo funzione del valore assunto nello stesso istante dagli ingressi

- Circuiti Sequenziali:

- Le variabili di uscita non dipendono solo dagli ingressi ma anche dalla storia precedente del circuito

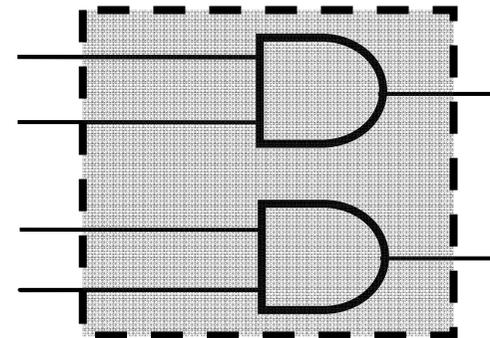
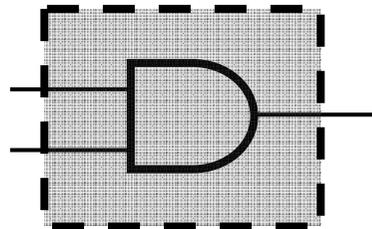
I circuiti sequenziali devono disporre di elementi in grado di mantenere *memoria* dello stato del sistema

Stato: riassunto finito della storia passata

# Circuiti Combinatori

---

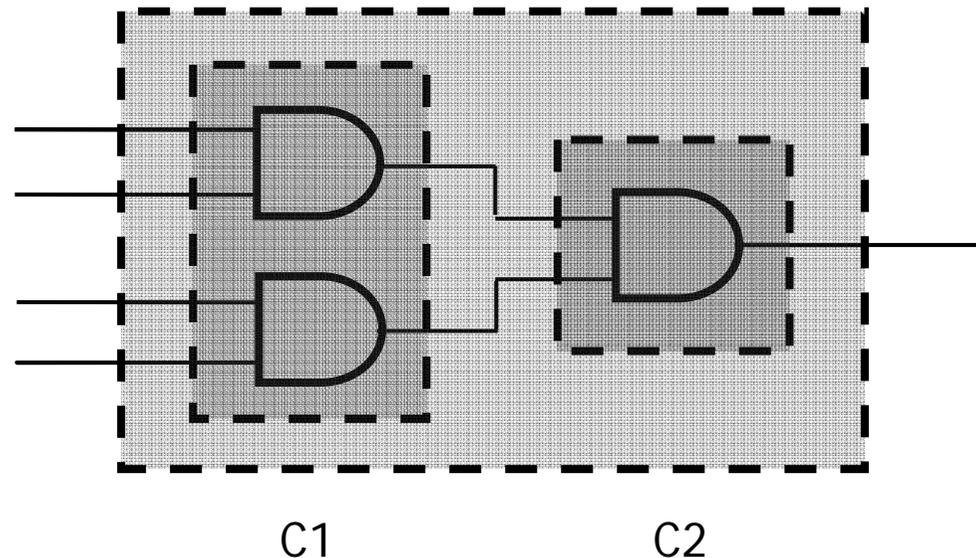
- I circuiti combinatori si dicono *well formed* (w.f.) o ben formati se si ottengono a partire da porte logiche primitive secondo le seguenti regole:
  - Una singola linea o una singola porta è un circuito w.f.
  - La giustapposizione di due circuiti w.f. genera un circuito w.f.



# Circuiti Combinatori

---

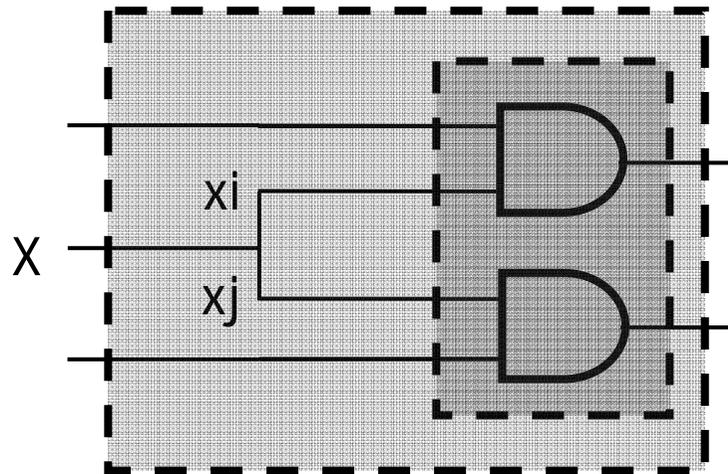
- Se C1 e C2 sono due circuiti w.f. separati, il circuito che si ottiene collegando linee di uscita di C1 con linee di ingresso di C2 è w.f.



# Circuiti Combinatori

---

- Se  $x_i$  e  $x_j$  sono linee di ingresso di un circuito w.f. e se si collega  $x_i$  e  $x_j$  tra loro in  $X$ , si ottiene un circuito w.f.



# Circuiti Combinatori

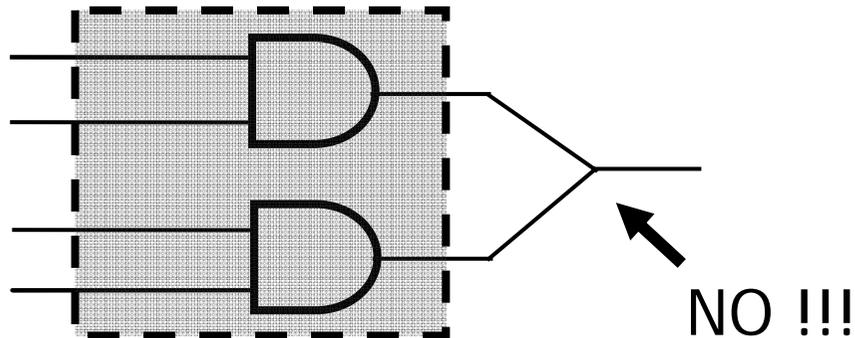
---

- I circuiti combinatori w.f. non contengono anelli e le uscite non sono mai collegate fra loro
- Eccezioni a questi comportamenti richiedono:
  - la comprensione di meccanismi a livelli di astrazione inferiore o
  - introducono meccanismi di memoria

# Circuiti Combinatori

---

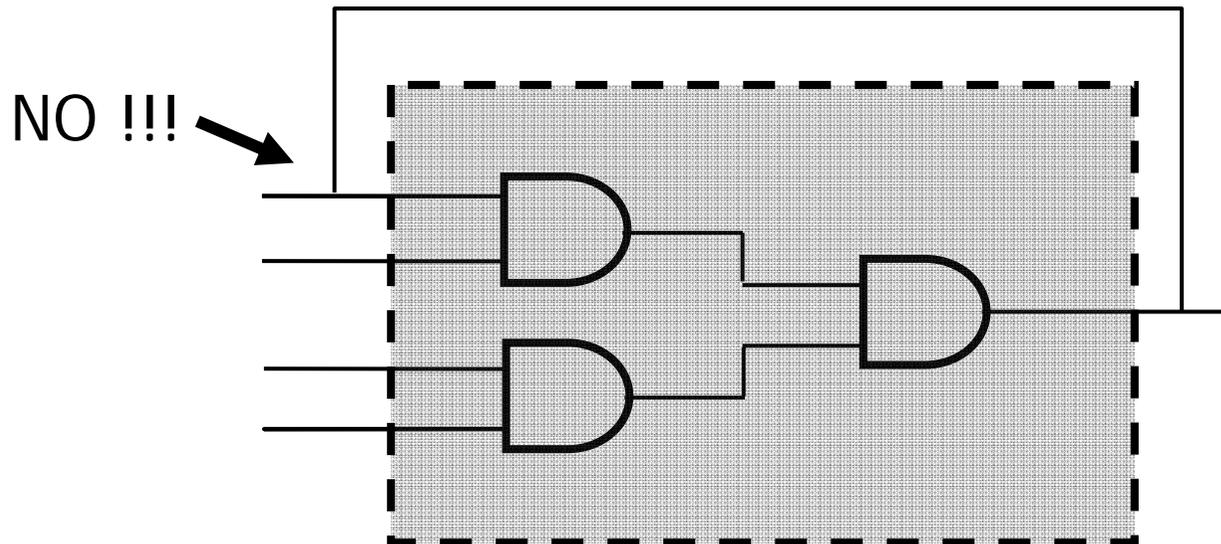
- Esempio di un circuito combinatorio con le uscite collegate fra loro:
  - richiede la comprensione di meccanismi a livelli di astrazione inferiore (tecnologia elettronica)



# Circuiti Combinatori

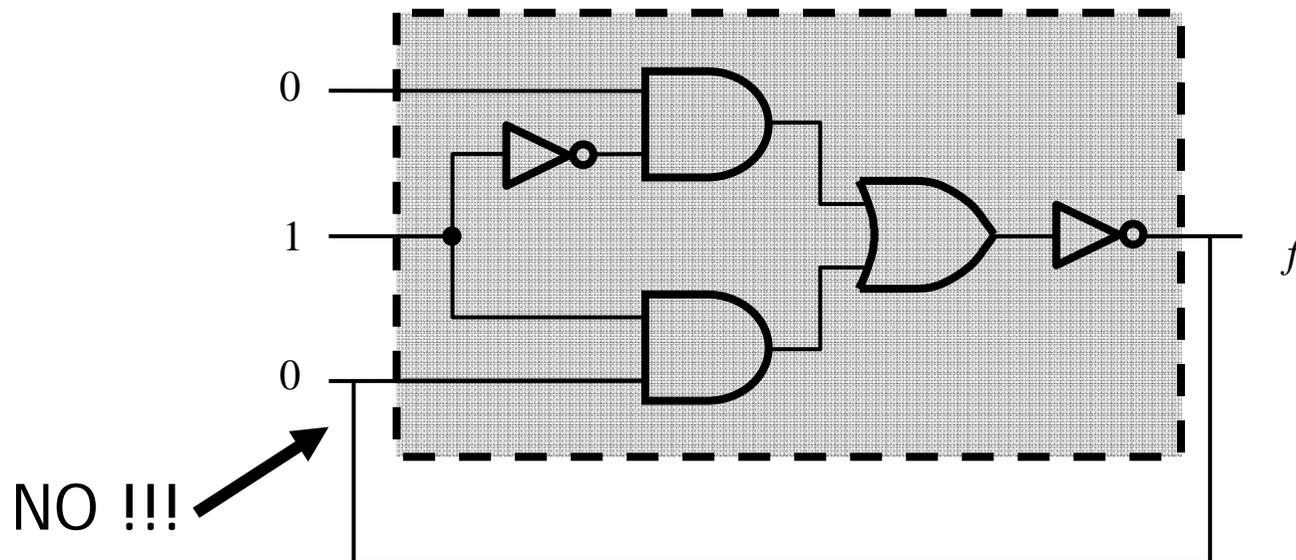
---

- Esempio di un circuito combinatorio con anelli:
  - Può introdurre meccanismi di memoria



# Circuiti Combinatori

- Esempio di un circuito combinatorio con anelli:
  - Può introdurre meccanismi di memoria o situazioni instabili



# Circuiti Sequenziali

---

- Un circuito sequenziale dispone di elementi di memoria per ricordare la storia passata
- Il più semplice elemento di memoria è il FLIP-FLOP, in grado di immagazzinare un bit di informazione per un tempo indefinito
- Il FLIP-FLOP costituisce una cella di memoria

# Circuiti Sequenziali

---

- I FLIP-FLOP o multivibratori bistabili sono elementi circuitali in grado di:
  - permanere stabili in due stati diversi (MEMORIA)
  - commutare da uno stato all'altro (SCRITTURA)
  - manifestare lo stato interno (LETTURA)

# Multivibratori

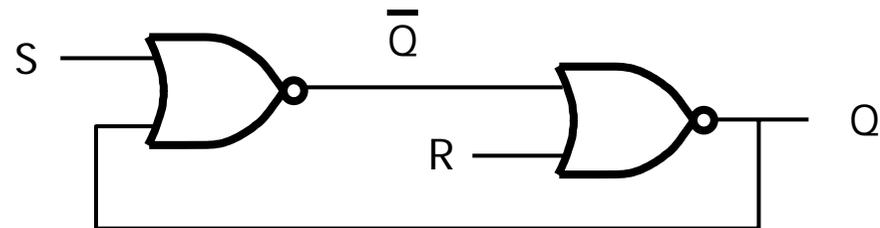
---

- Altri tipi di multivibratori sono gli:
  - Astabili:
    - Continuano indefinitamente a vibrare tra due configurazioni diverse
  - Monostabili:
    - Permangono in una configurazione stabile fino a che non vengono eccitati dall'esterno. Dopo un transitorio, ritornano alla configurazione iniziale.

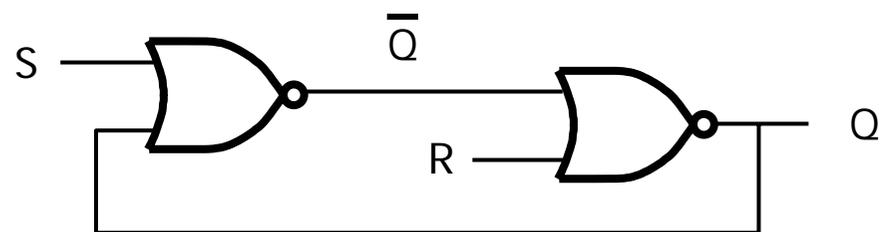
# Flip-Flop SR

---

- I due stati stabili sono:



Memorizza 1

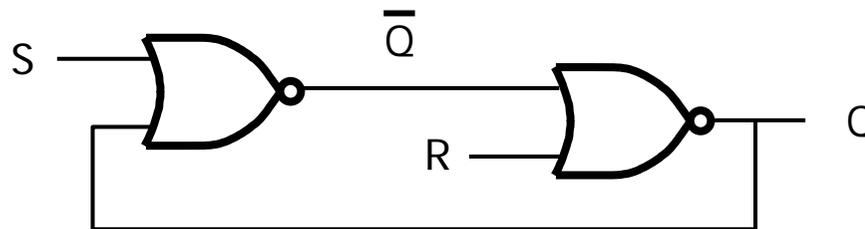


Memorizza 0

# Flip-Flop SR

---

- Esempio di funzionamento:
  - Da ingresso  $SR=00$  a  $SR=01$

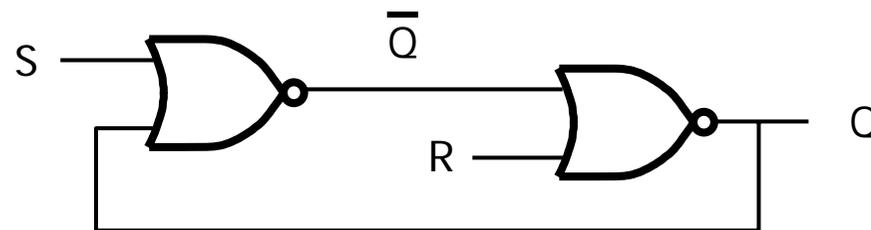


Memorizza 0

# Flip-Flop SR

---

- Funzionamento:
  - Ingresso  $SR=00$

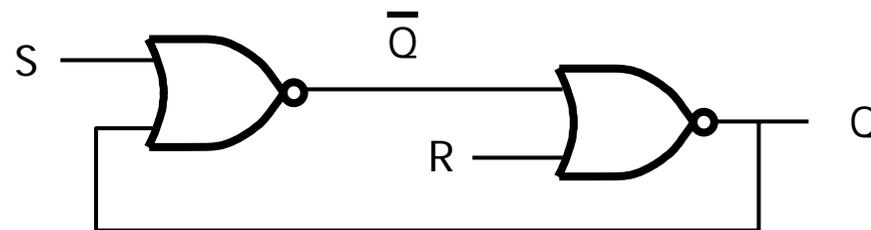


Memorizza 1

# Flip-Flop SR

---

- Funzionamento:
  - Ingresso SR=01

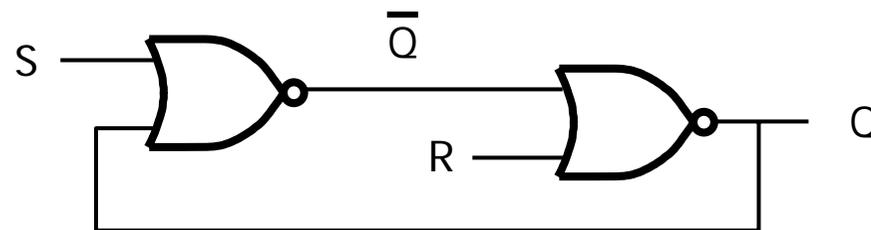


Memorizza 0

# Flip-Flop SR

---

- Funzionamento:
  - Ingresso  $SR=00$

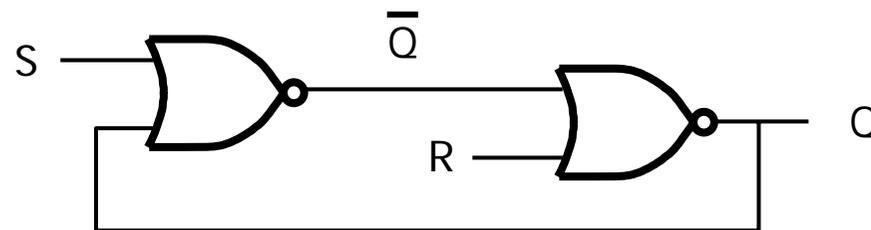


Memorizza 0

# Flip-Flop SR

---

- Funzionamento:
  - Ingresso SR=10

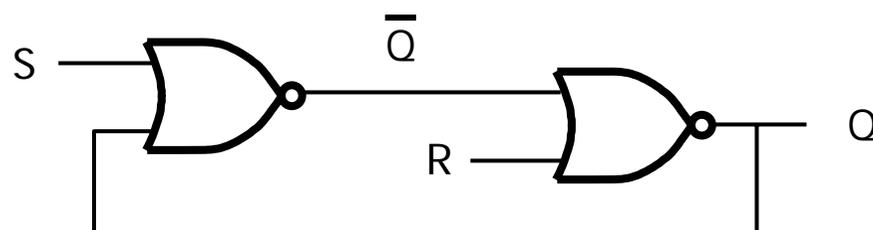


Memorizza 1

# Flip-Flop SR

---

- Funzionamento:
  - Ingresso SR=11



Memorizza ?

Il valore memorizzato quando S o R tornano a 0 dipende dalla specifica implementazione del circuito (ritorneremo a breve)

# Flip-Flop SR

---

- Scrittura:
  - $SET=1$  e  $RESET=0 \Rightarrow \overline{Q} = 0$  e  $Q = 1$
  - $RESET=1$  e  $SET=0 \Rightarrow Q = 0$  e  $\overline{Q} = 1$
- Quando entrambi tornano al valore 0, il circuito mantiene la stessa configurazione delle uscite
- Questi sono i modi per scrivere 0 o 1 nell'elemento di memoria

# Flip-Flop SR

---

## ■ Scrittura:

– SET=1 e RESET=0  $\Rightarrow \overline{Q} = 0$  e  $Q = 1$

– RESET=1 e SET=0  $\Rightarrow Q = 0$  e  $\overline{Q} = 1$

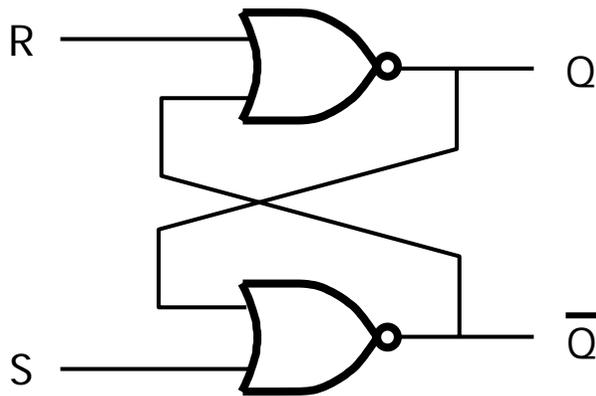
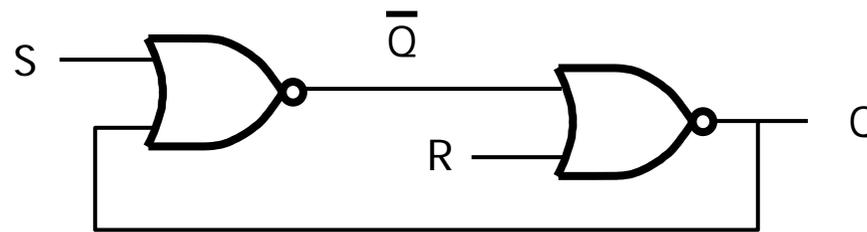
## ■ Quando entrambi tornano al valore 0, il circuito mantiene la stessa configurazione delle uscite

S	R	Q	$\overline{Q}$	STATO
1	0	1	0	SET
0	1	0	1	RESET
0	0	Q	$\overline{Q}$	MEMORIA
1	1	0	0	NON VALIDO

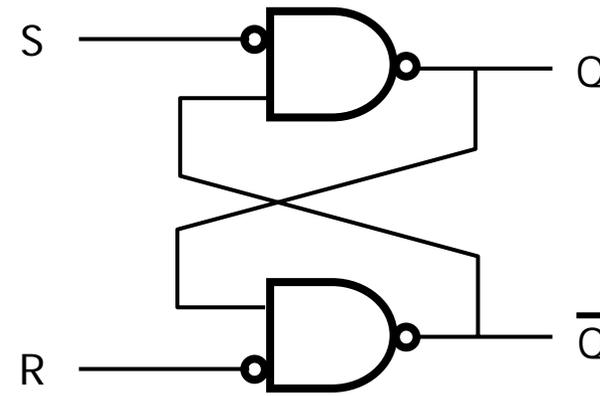
# Flip-Flop SR

---

- Diverse rappresentazioni:



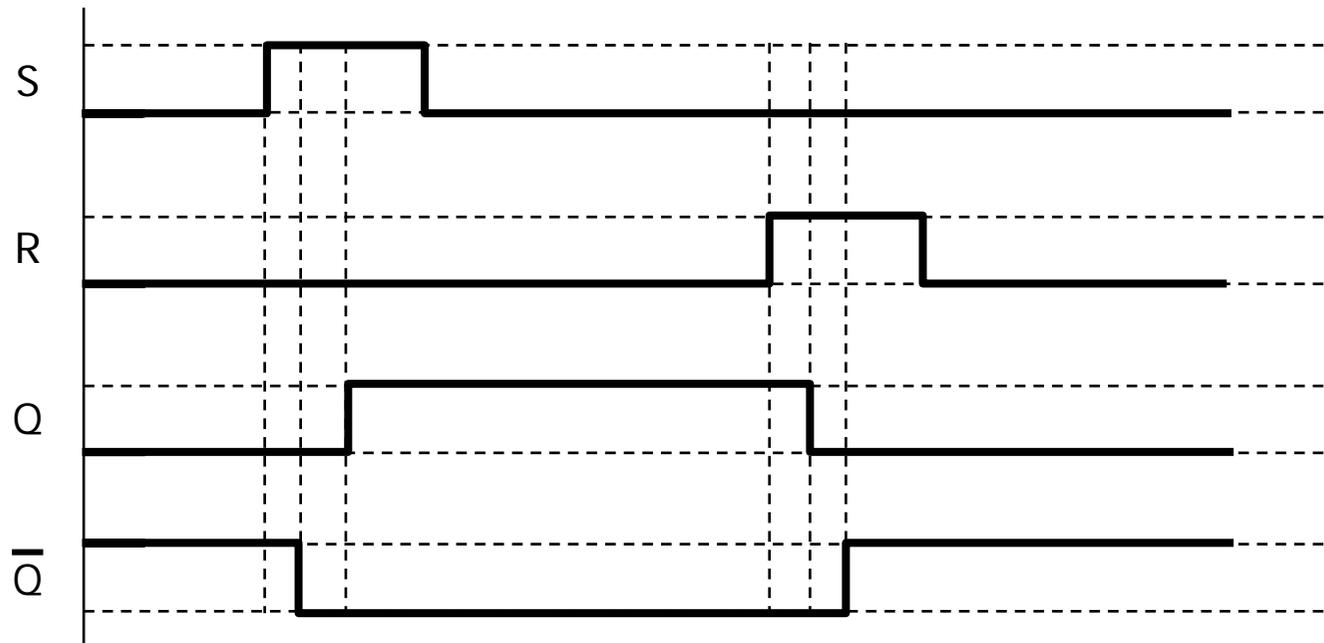
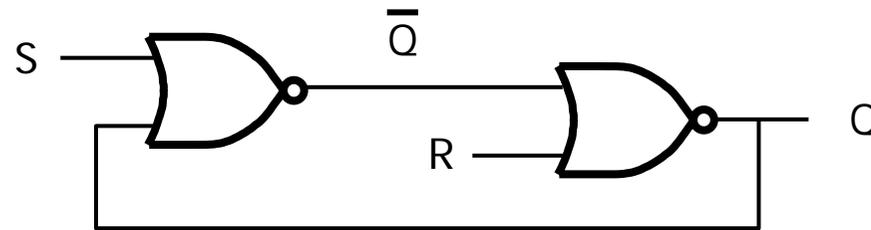
Con porte NOR



Con porte NAND

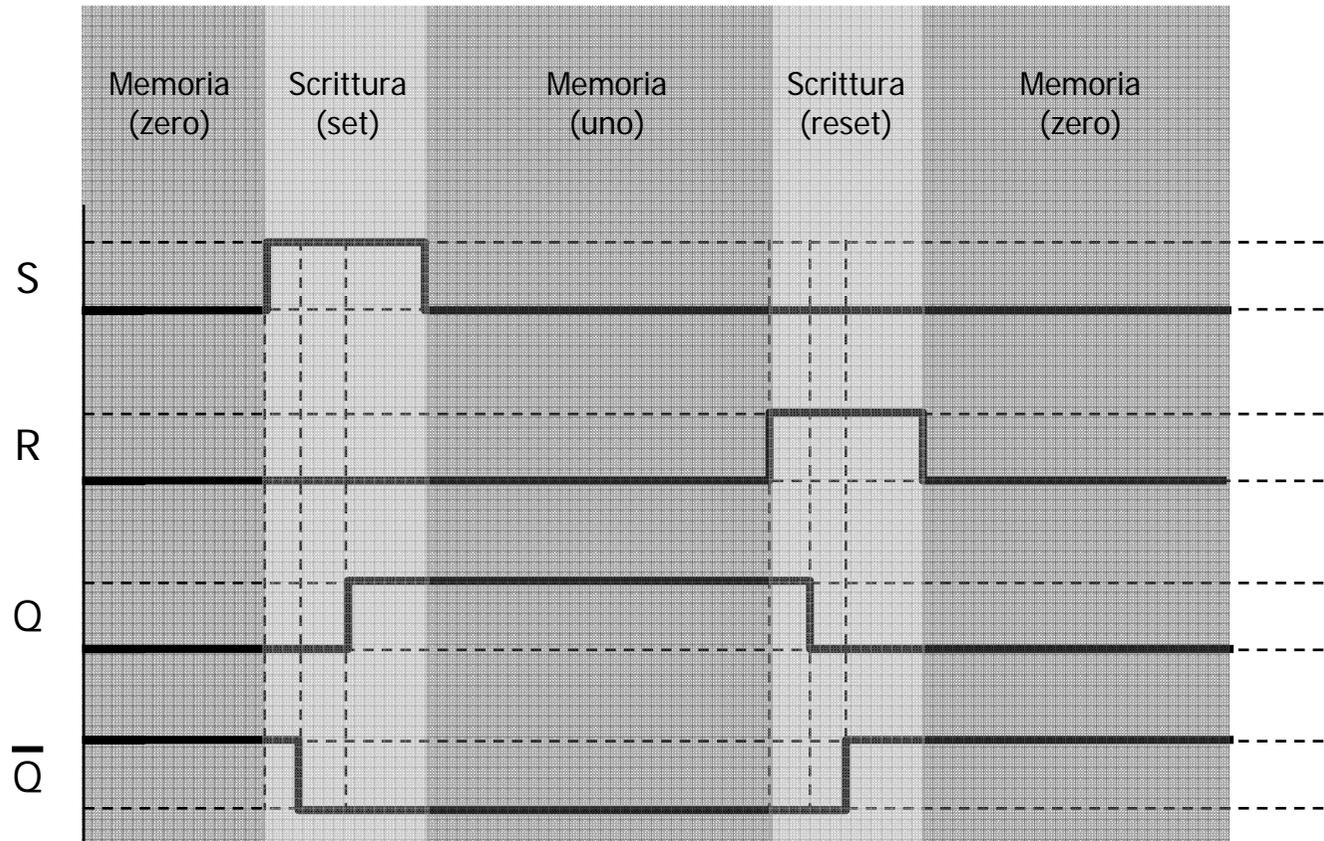
# Flip-Flop SR

- Andamento temporale dei segnali:



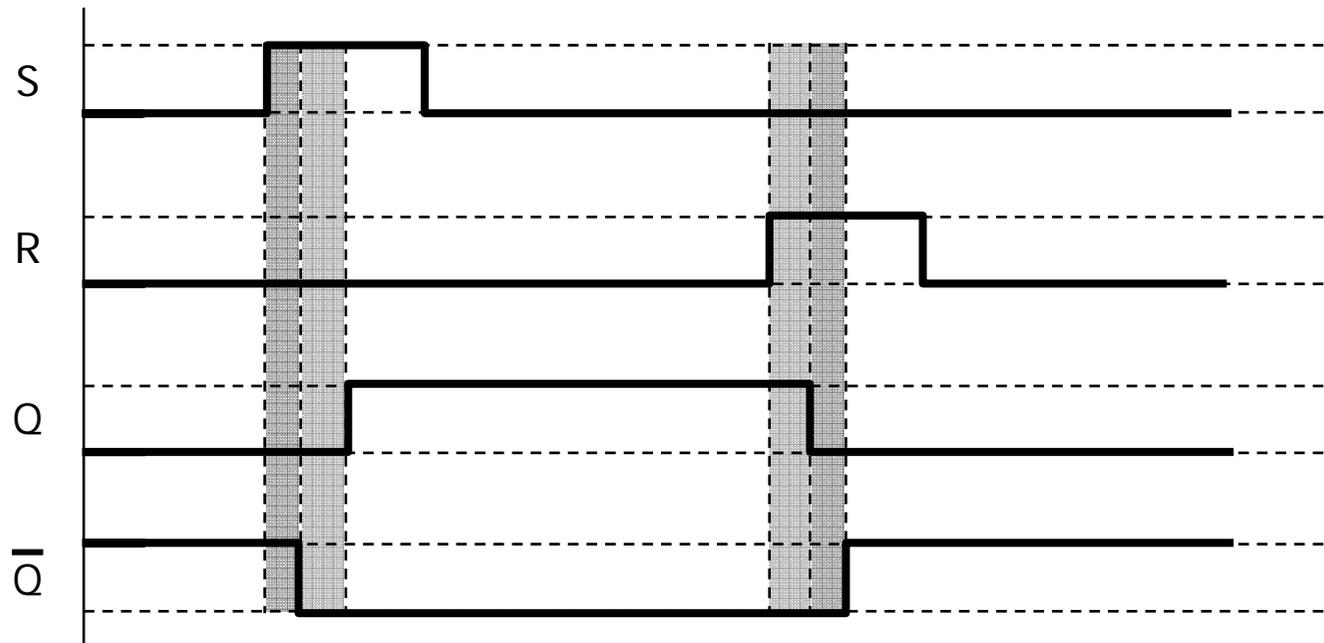
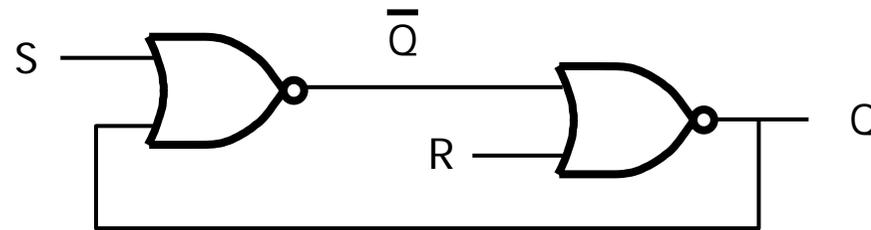
# Flip-Flop SR

## ■ Fasi del Flip-Flop:



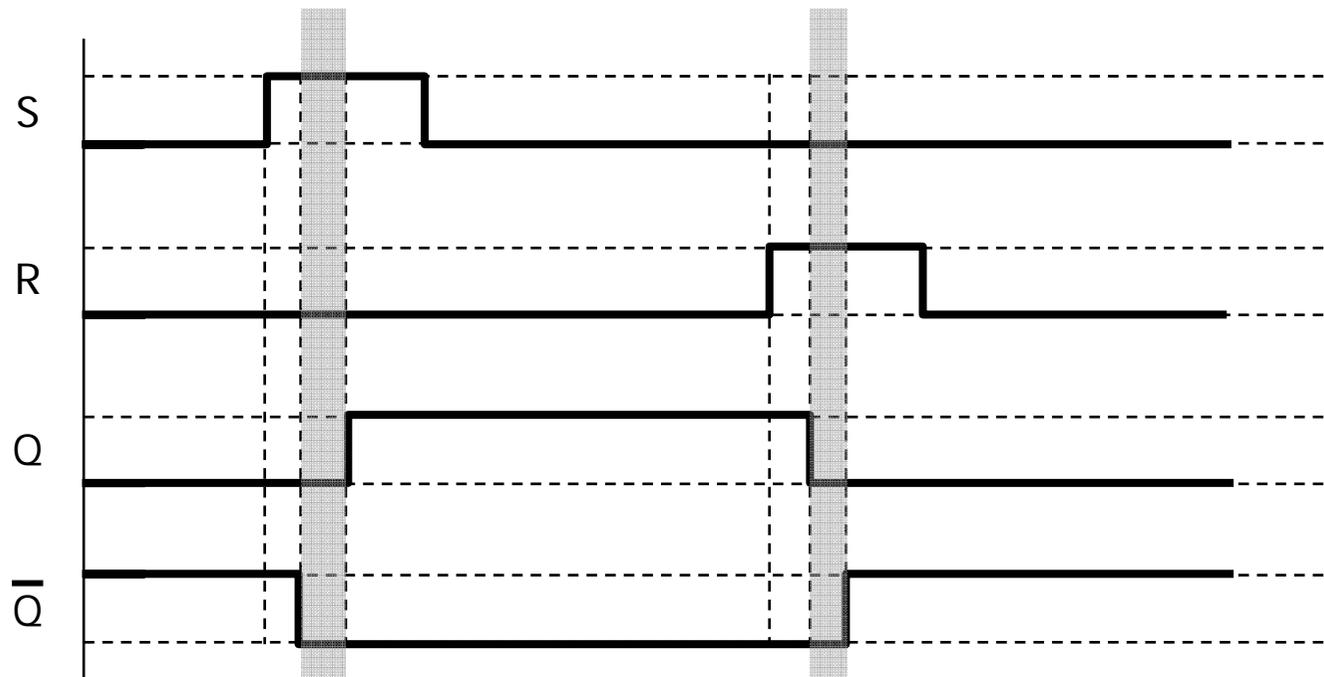
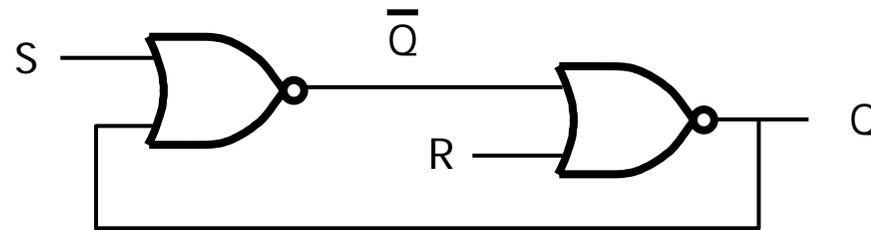
# Flip-Flop SR

- Ritardi delle porte logiche:



# Flip-Flop SR

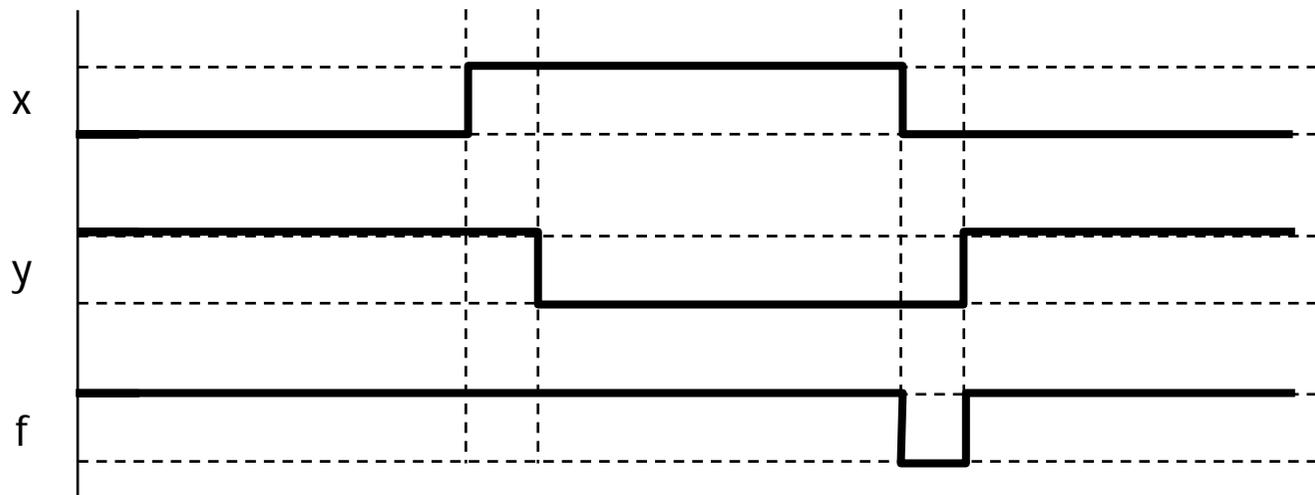
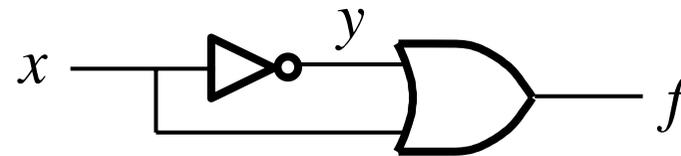
- Stati del Flip-Flop non consistenti:



# Flip-Flop SR

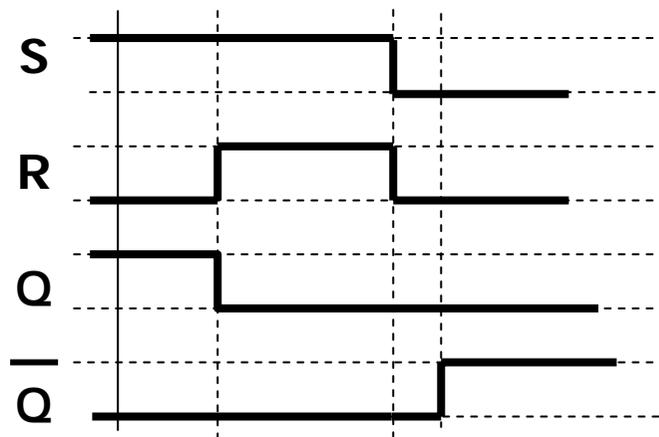
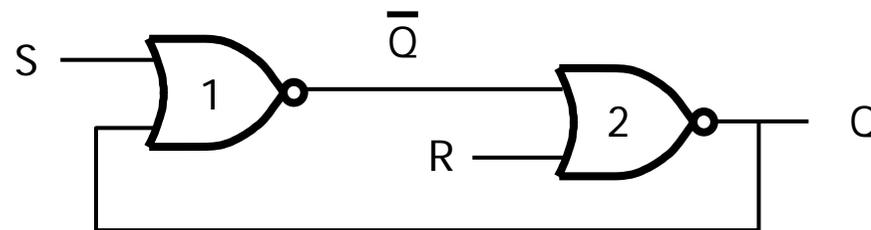
- I ritardi delle porte possono portare a comportamenti indesiderati:

$$f = (x + \bar{x})$$

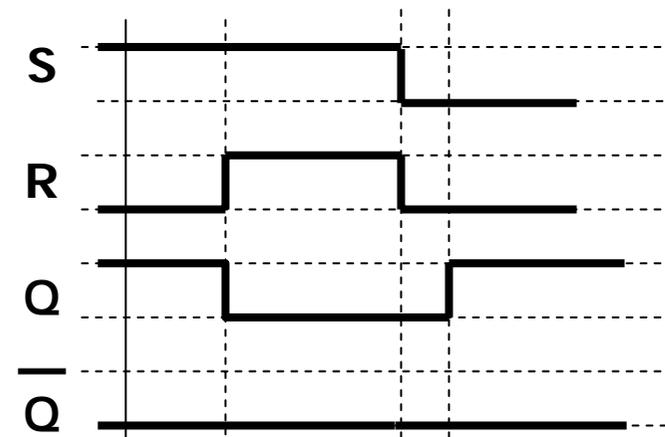


# Flip-Flop SR

- Configurazione di ingresso non consentita:



La porta 1 commuta per prima

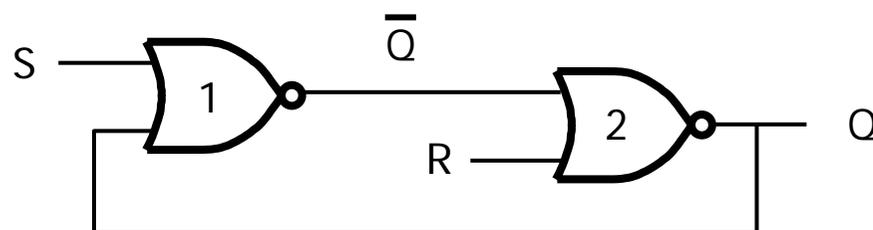


La porta 2 commuta per prima

# Flip-Flop SR

---

- Configurazione di ingresso non consentita:



- SR=11 non è consentita e non dovrà mai comparire all'ingresso del FLIP-FLOP, in quanto:
  - Forza uscite non complementari
  - La transizione SR=11  $\Rightarrow$  00 porta a situazioni diverse in funzione del ritardo introdotto dalle singole porte (al termine il valore delle uscite non è definito)

# Corse critiche

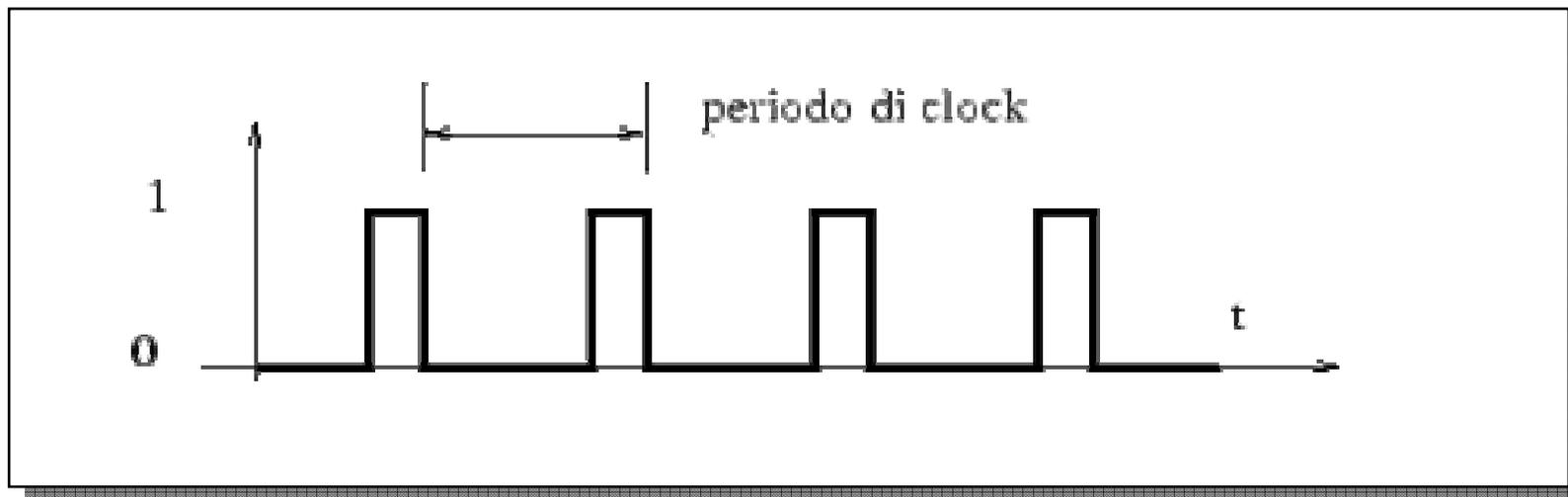
---

- I fenomeni di questo tipo prendono il nome di corse critiche:
  - Configurazioni non volute si presentano agli ingressi di circuiti di memoria
  - Differenze di comportamento tra i circuiti portano a evoluzioni casuali

# Corse critiche e Clock

---

- Per eliminare le corse critiche e per razionalizzare il progetto viene introdotto il segnale di cadenza o di clock



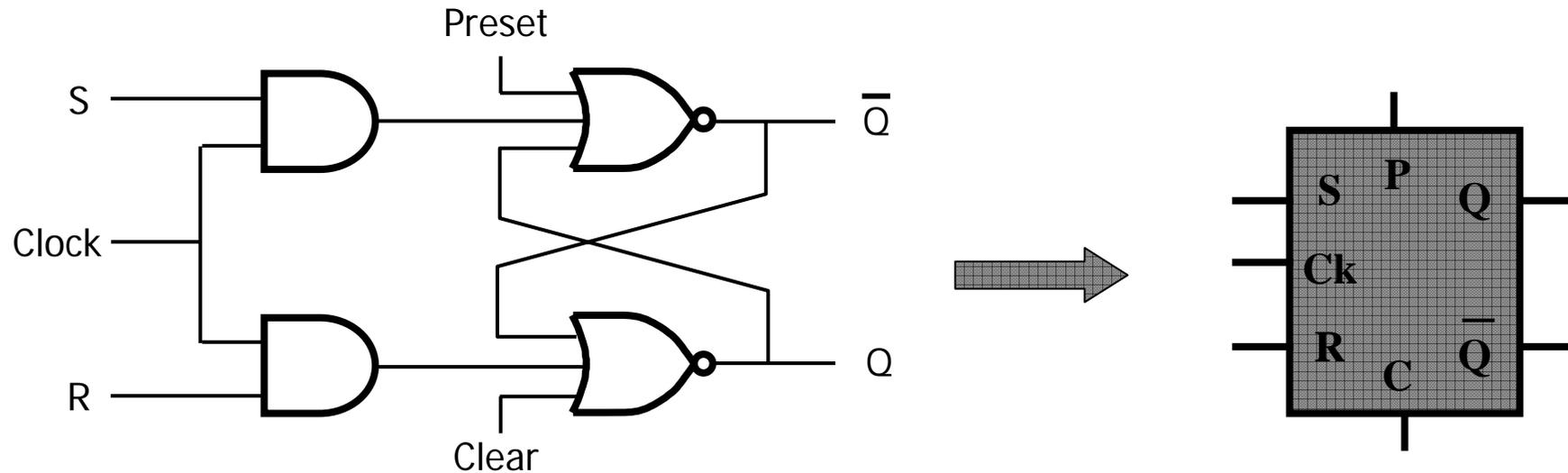
# Il Segnale di Clock

---

## CLOCK A LIVELLO

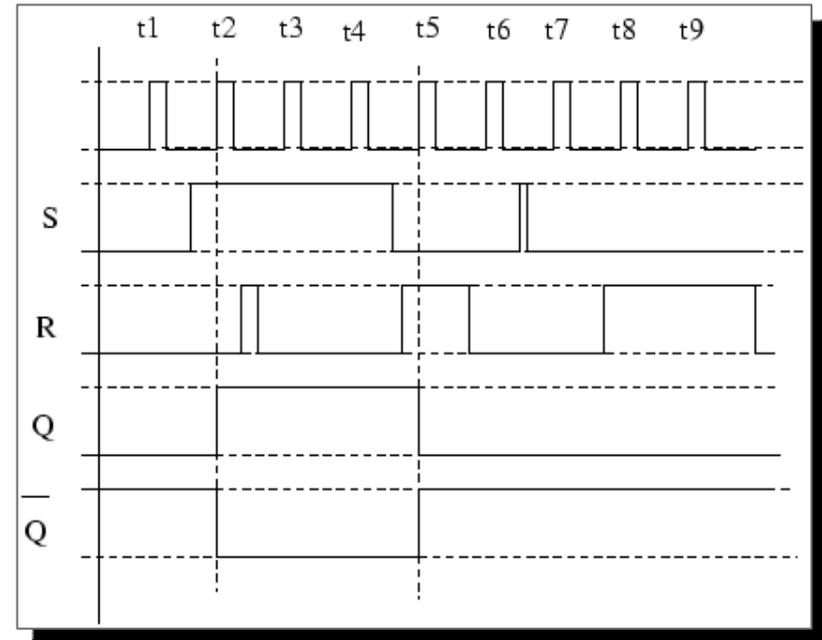
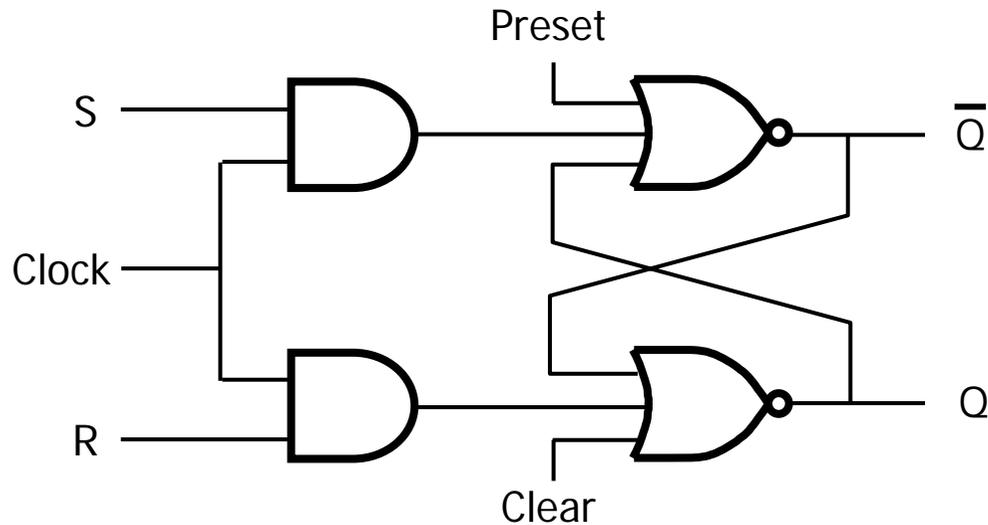
- Stato attivo  $\Rightarrow 1$ 
  - Vi avvengono le transizioni
  - Ha la durata minima necessaria a garantire il cambiamento di stato di tutti i circuiti di memoria
  
- Stato inattivo  $\Rightarrow 0$ 
  - Viene consentito alla rete combinatoria di modificare le uscite
  - La durata minima consente in ogni caso l'evoluzione completa della rete

# Filp-Flop SR Cadenzato



- **CLOCK:**
  - con  $CLOCK=0$  ingressi bloccati, il sistema evolve
  - con  $CLOCK=1$ , S e R sono attivi
- **Preset (P) e Clear (C) forzano il FF in uno dei due stati, indipendentemente dal clock**

# Filp-Flop SR Cadenzato

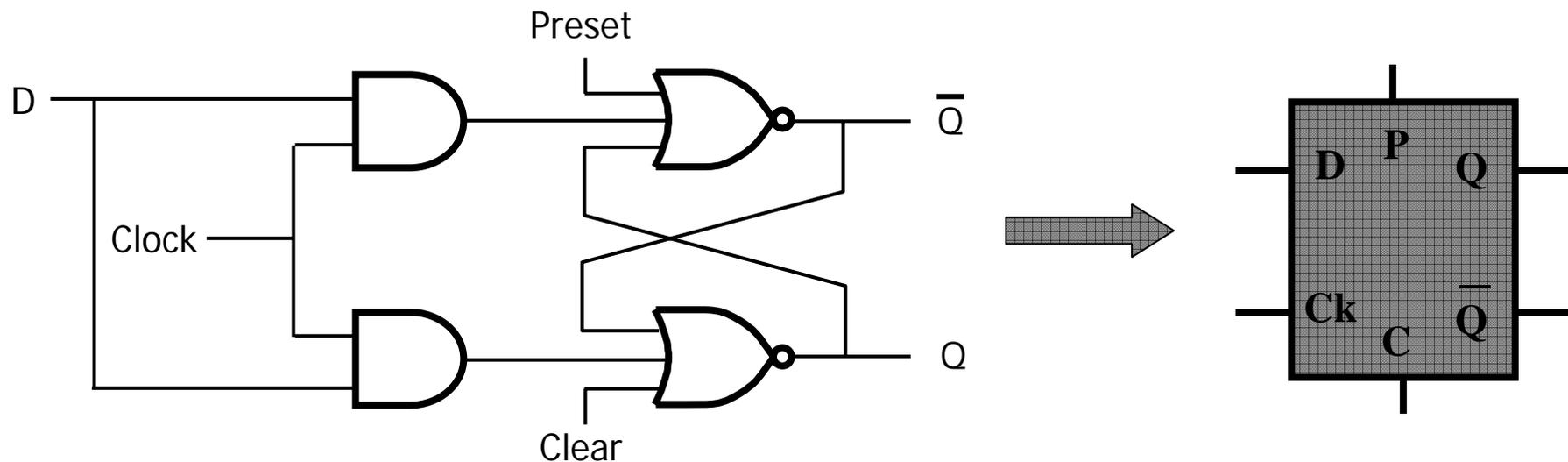


<b>S</b>	<b>R</b>	<b>Ck</b>	<b>Q(n+1)</b>
<b>X</b>	<b>X</b>	<b>0</b>	<b>Q(n)</b>
<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>
<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>
<b>0</b>	<b>0</b>	<b>1</b>	<b>Q(n)</b>
<b>1</b>	<b>1</b>	<b>1</b>	<b>x</b>

I segnali di ingresso S e R non cambiano quando il segnale di CLOCK vale 1 (per definizione)

# Flop-Flop D-Latch

- Il D-Latch memorizza il dato in ingresso



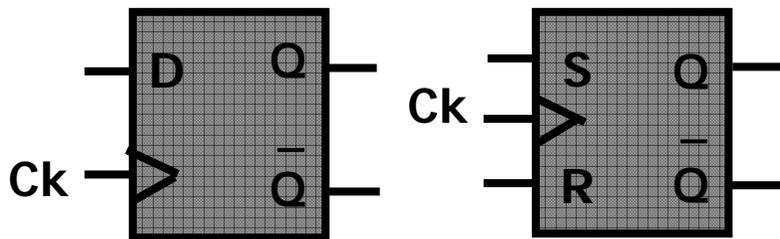
$D$	$Ck$	$Q(n+1)$
1	1	1
0	1	0
1	0	$Q(n)$
0	0	$Q(n)$

# Flip-Flop Edge-Triggered

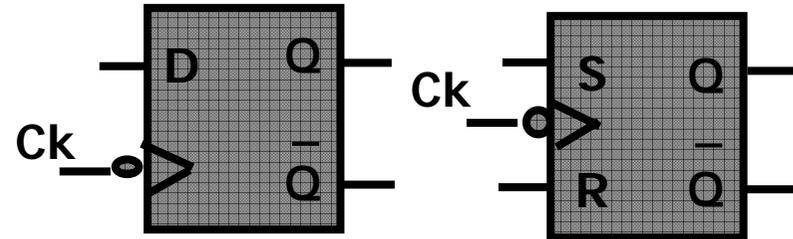
- I FF attivi sul livello richiedono un circuito complesso per la generazione del segnale di clock.
  - basso duty-cycle, definito come  $t_{(Ck=1)}/t_{(Ck=0)}$
- Sono soggetti a rischio di transizioni multiple nello stesso intervallo di CLOCK

# Flip-Flop Edge-Triggered

- I FF più sicuri (FF sincroni) commutano sui fronti di salita o sui fronti di discesa del segnale di CLOCK

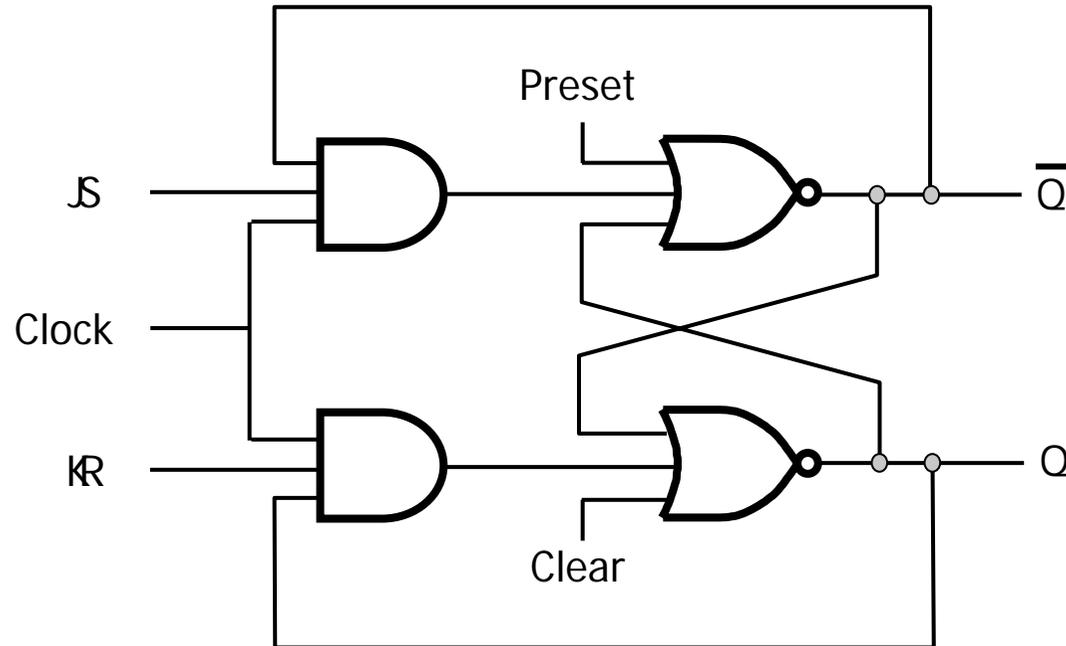


FF attivi sul fronte di salita



FF attivi sul fronte di discesa

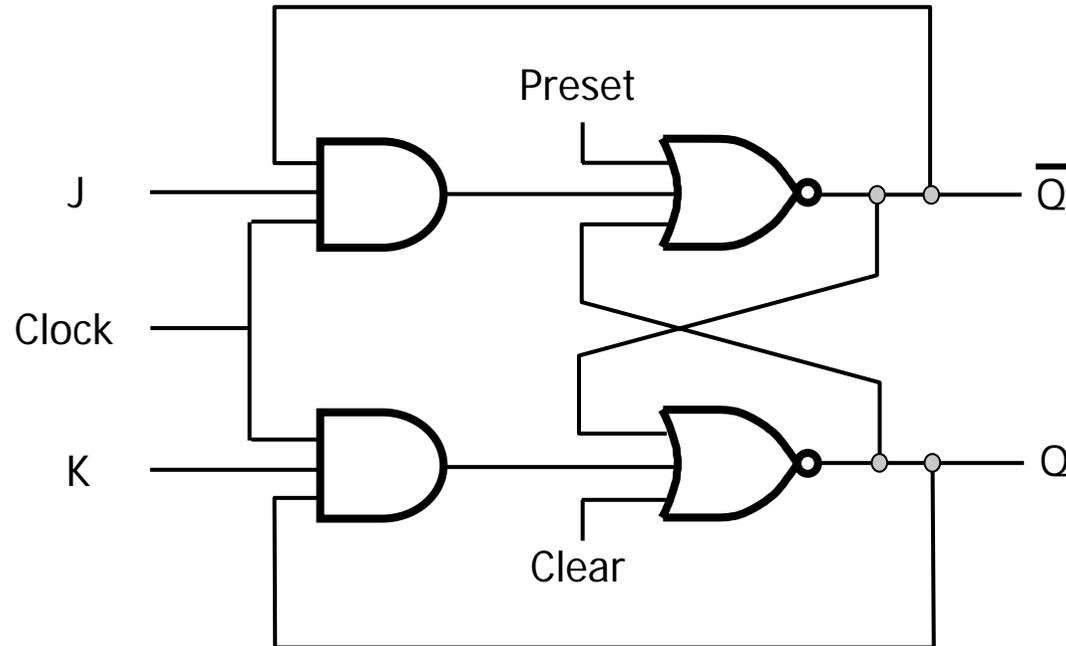
# Altri tipi di FF Sincroni



- Si aggiunge una retroazione e i segnali di ingresso SR diventano JK

# Flip-Flop JK

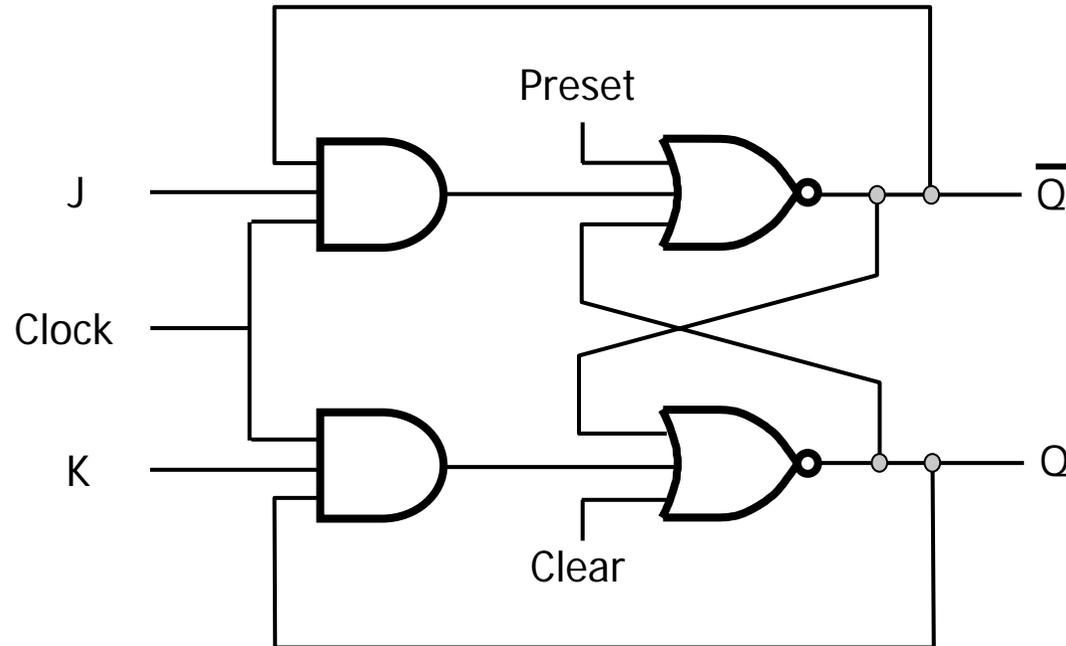
---



- Se il segnale di CLOCK è 0, gli ingressi vengono inibiti

# Flip-Flop JK

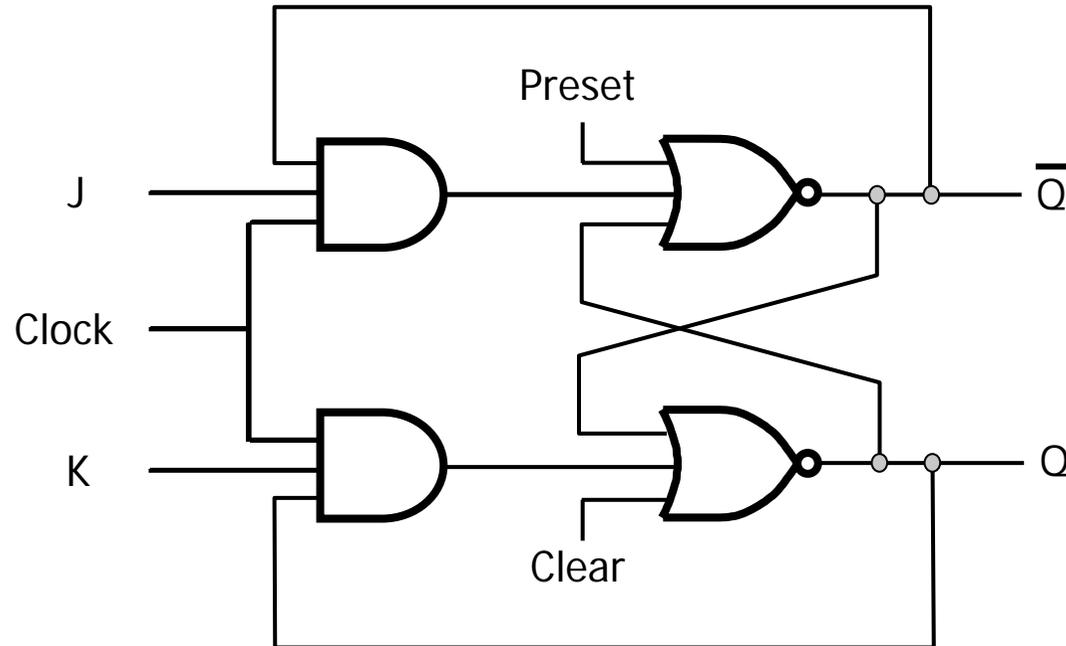
---



- Si comporta come un FLIP-FLOP S-R ma accetta in ingresso anche la combinazione  $JK=11$  che inverte il valore delle uscite

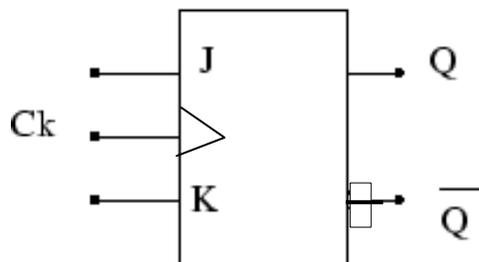
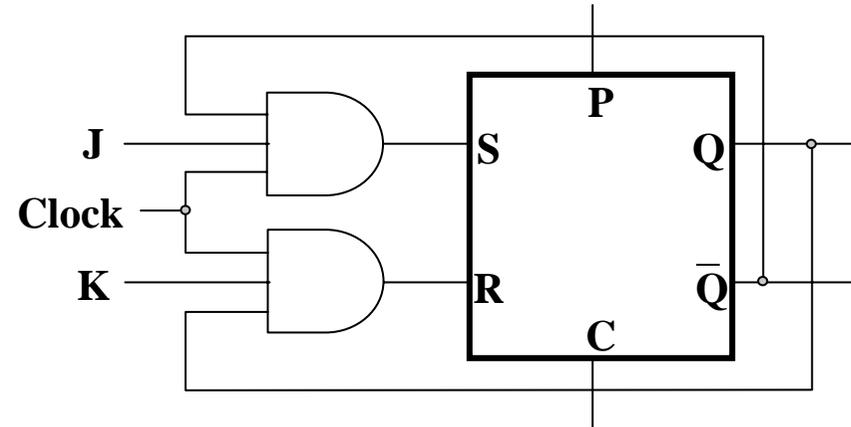
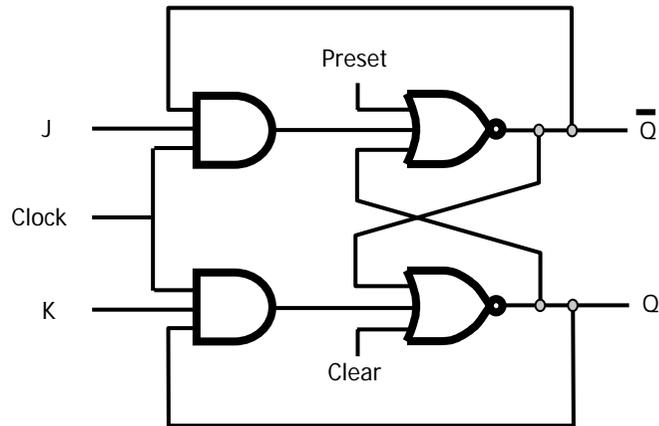
# Flip-Flop JK

---



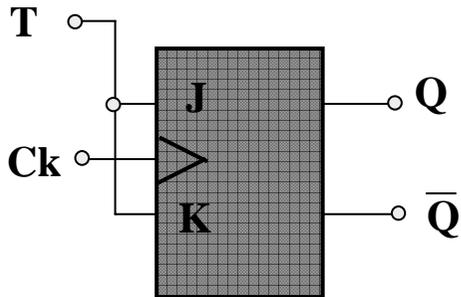
- Infatti: se stato=0 e JK=11, lo stato viene complementato

# Flip-Flop JK

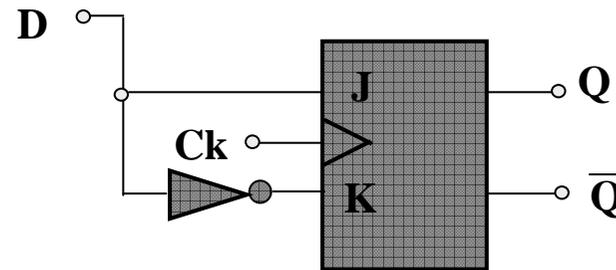


$J$	$K$	$Ck$	$Q(n+1)$
X	X	0	$Q(n)$
1	0	1	1
0	1	1	0
0	0	1	$Q(n)$
1	1	1	$\overline{Q(n)}$

# Altri Flop-Flop Cadenzati



$T$	$J$	$K$	$Q(n+1)$
-	1	0	1
-	0	1	0
0	0	0	$Q(n)$
1	1	1	$\overline{Q(n)}$

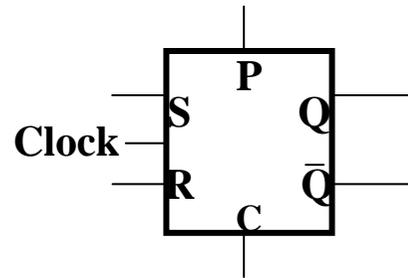


$D$	$J$	$K$	$Q(n+1)$
1	1	0	1
0	0	1	0
-	0	0	$Q(n)$
-	1	1	$\overline{Q(n)}$

- **FF-T: ad ogni colpo di clock commuta o no a seconda del valore di T**
- **FF-D: ad ogni colpo di clock scrive 1 o 0 a seconda del valore di D**

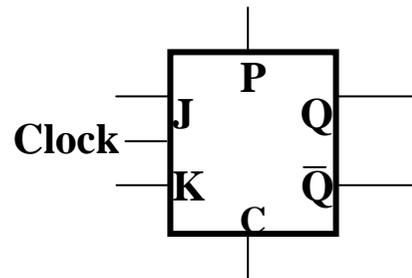
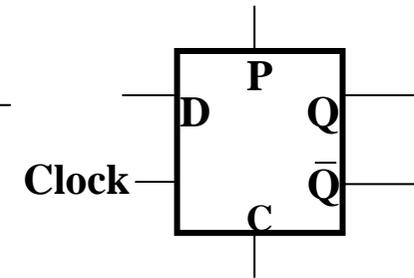
# Tabella riassuntiva

---



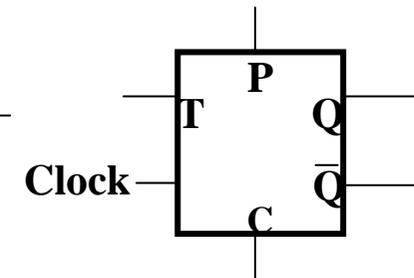
		Input SR			
		00	01	10	11
Uscita	0	0	0	1	-
	1	1	0	1	-

		Input D	
		0	1
Uscita	0	0	1
	1	0	1

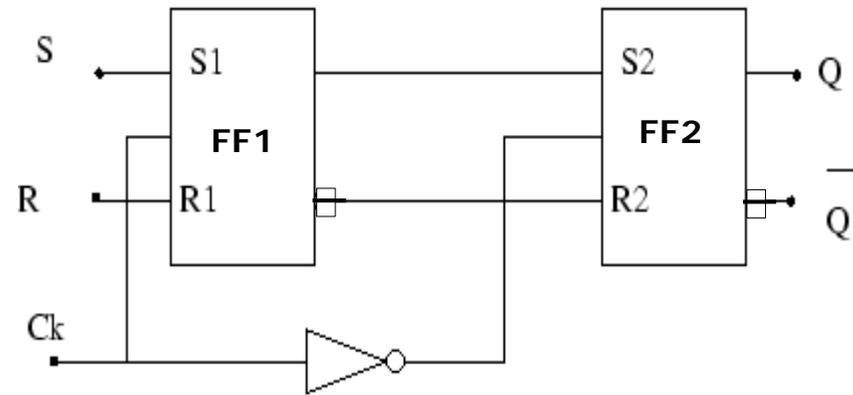
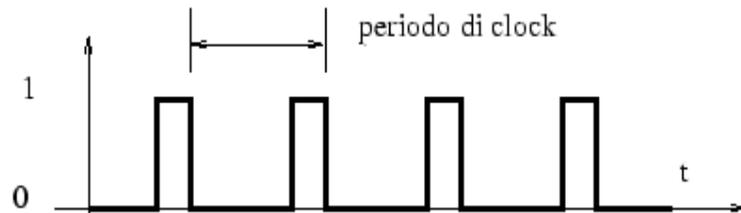


		Input JK			
		00	01	10	11
Uscita	0	0	0	1	1
	1	1	0	1	0

		Input T	
		0	1
Uscita	0	0	1
	1	1	0



# Flip-Flop Master/Slave



**Il segnale di clock separa le variazioni dei dispositivi di memoria dall'assestamento dei circuiti combinatori.**

**Ck=0: FF1 disabilitato, FF2 abilitato, le uscite commutano e potenzialmente tutti i circuiti che seguono**

**Ck=1: FF1 abilitato, FF2 disabilitato, le uscite sono stabili e FF1 si assesta in funzione del valore degli ingressi**

