
Gli elementi di memoria: i bistabili

I bistabili: dai bistabili asincroni ai Flip-Flop Edge-Triggered

Introduzione

Bistabili Asincroni

Bistabili Sincroni: Latch e Flip-Flop

Tabelle riassuntive

Utilizzo dei bistabili

- Nei circuiti sequenziali il valore delle uscite in un determinato istante dipende sia dal valore degli ingressi in quello stesso istante sia dal tempo.
 - ▶ Una stessa configurazione di ingresso applicata in due istanti di tempo successivi può produrre due valori d'uscita differenti.
- Un circuito sequenziale *ha memoria degli eventi passati* e, quindi, richiede degli elementi in grado di conservare informazioni.
 - ▶ In un generico istante t l'informazione relativa al “contenuto” di questa memoria è rappresentata nel concetto di *stato*.

- Gli elementi in grado di conservare informazioni sono detti *bistabili*.
 - ▶ Il termine *bistabile* deriva dal fatto che tale elemento è stabile in due stati (0 e 1) e che le transizioni di stato sono forzate da un segnale di ingresso.
 - Nota: i *bistabili* sono caratterizzati dalla volatilità cioè rispettano quanto indicato solo se alimentati.
- La differenza principale tra i vari tipi di elementi di memoria è costituita da:
 - ▶ Numero di ingressi dell'elemento di memoria.
 - ▶ Modo in cui tali ingressi ne determinano lo stato.

■ Classificazione dei *bistabili*:

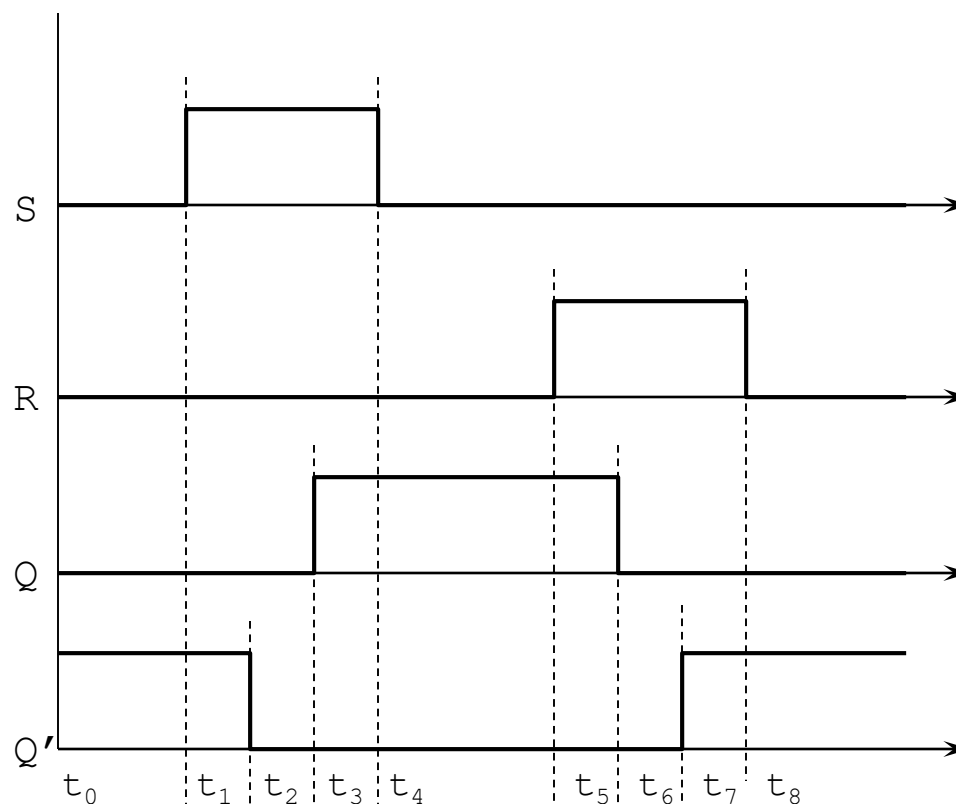
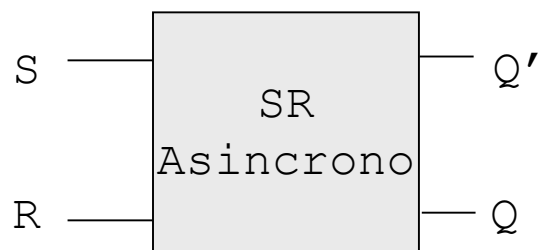
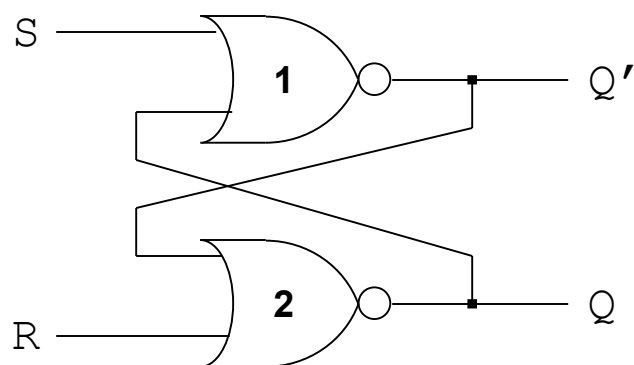
▶ Asincroni

- Sono privi di un segnale di sincronizzazione e modificano il loro stato rispondendo direttamente ad eventi sui segnali di ingresso.

▶ Sincroni

- sono sensibili ad un segnale di controllo (spesso il clock) e la transizione da uno stato all'altro avviene solo in corrispondenza di un impulso del segnale di controllo.
- Ulteriore classificazione dei bistabili sincroni:
 - bistabili sincroni controllati (*gated latch*);
 - flip-flop.
 - » flip-flop a livello (pulse-triggered detti anche o master-slave)
 - » flip-flop a fronte (edge-triggered)
 - » flip-flop con blocco dati (data lock-out) [non trattati].

- Il bistabile asincrono più semplice è il bistabile SR (*Set-Reset*)
 - ▶ Viene utilizzato come blocco base per realizzare bistabili più complessi.



■ Analisi di funzionamento:

- ▶ Tempo $t = t_0 = 0$
 - Condizione iniziale: $S=0$, $R=0$ e $Q=0$, $Q'=1$
- ▶ Tempo $t=t_1$: evento $S=1$
 - La porta 1 ha in ingresso $1, 0$ e in uscita, al tempo t_2 , $Q'=0$
- ▶ Tempo $t=t_2$
 - La porta 2 ha in ingresso $0, 0$ e in uscita, al tempo t_3 , $Q=1$
- ▶ Tempo $t= t_3$
 - La porta 1 ha in ingresso $1, 1$ e mantiene l'uscita a $Q'=0$ mentre la porta 2 ha in ingresso $0, 0$ e mantiene l'uscita a $Q=1$
- ▶ Tempo $t= t_4$: evento $S=0$
 - La porta 1 ha in ingresso $0, 1$ e quindi mantiene l'uscita $Q'=0$ mentre la porta 2 ha in ingresso $0, 0$ e quindi mantiene l'uscita a $Q=1$.
 - Il circuito è stabile nello stato $Q=1$, $Q'=0$

■ Analisi di funzionamento (cont.):

► Tempo $t = t_5$: Evento $R=1$

- La porta 2 ha in ingresso 1, 0 e in uscita, al tempo t_6 , $Q=0$.

► Tempo $t=t_6$

- La porta 1 ha in ingresso 0, 0 e in uscita, al tempo t_7 , $Q'=1$.

► Tempo $t=t_7$

- La porta 2 ha in ingresso 1, 1 e mantiene l'uscita a $Q=0$ mentre la porta 1 ha in ingresso 0, 0 e quindi mantiene l'uscita a $Q'=1$.

► Tempo $t = t_8$: evento $R=0$

- La porta 2 ha in ingresso 0, 1 e quindi mantiene l'uscita a $Q=0$ e la porta 1 ha in ingresso 0, 0 e quindi mantiene l'uscita a $Q'=1$
- Il circuito è stabile nello stato $Q=0$, $Q'=1$


- I segnali S e R prendono il nome di *Set* e *Reset*:
 - ▶ Un 1 su Set porta Q ad 1 mentre un 1 su Reset porta Q a 0.
- Riassumendo:
 - ▶ Un valore 1 sull'ingresso S quando R ha valore 0 porta le uscite allo stato stabile $Q=1$, $Q'=0$; riportando a 0 l'ingresso S lo stato delle uscite non cambia;
 - ▶ Un valore 1 sull'ingresso R con S a valore 0 porta le uscite allo stato stabile $Q=0$, $Q'=1$; riportando a 0 l'ingresso R lo stato delle uscite non cambia.
 - ▶ Un valore 0 sugli ingressi S e R non modifica lo stato;
 - ▶ La configurazione $S=1$ e $R=1$ è una configurazione non ammissibile.
- Osservazione: nelle configurazioni valide le uscite Q e Q' sono complementari per costruzione.

- Applicando contemporaneamente su S e R un valore 1 il circuito si porta in uno stato instabile 00; tale configurazione non è ammissibile.
 - ▶ Nel passaggio da 11 a 00, non è possibile identificare chi tra S o R cambia per primo;
 - ▶ Il bistabile asincrono ritorna in modo imprevedibile allo stato 01 o 10.
 - ▶ Questa condizione è chiamata corsa critica (*race condition*) o *transizione non-deterministica*

□ Rappresentazioni del comportamento di un bistabile SR

■ Tabella delle transizioni

SR					
		00	01	11	10
Q	0	0	0	-	1
	1	1	0	-	1



S	R	Q*
0	0	Q
0	1	0
1	0	1
1	1	-

□ Tabella delle eccitazioni

Q	Q*	S	R
0	0	0	-
0	1	1	0
1	0	0	1
1	1	-	0

□ Espressione logica

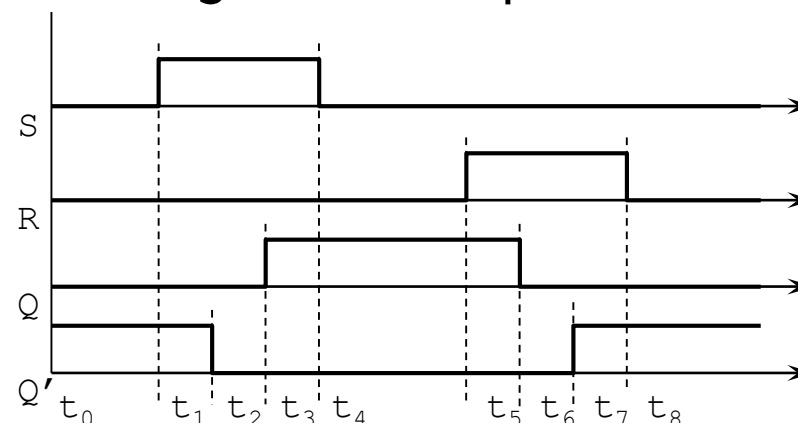
$$Q^* = S + R'Q$$

Con vincolo $S=R \neq 1$

Q*: stato futuro

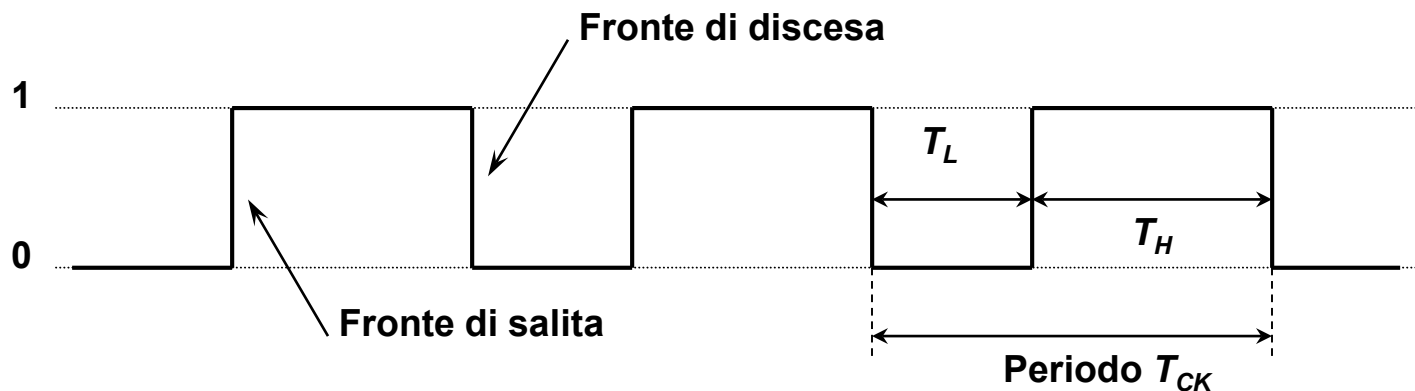
Q: stato presente

□ Diagramma Temporale



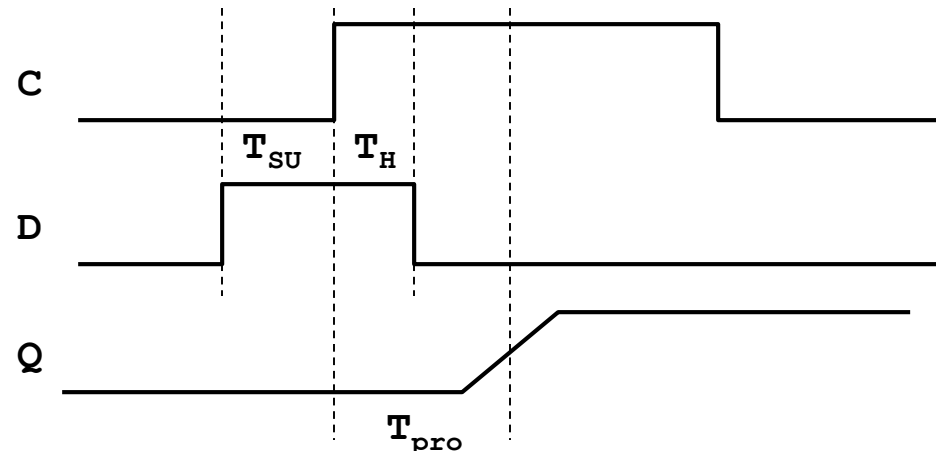
- Un bistabile asincrono modifica il proprio stato solo in relazione ad eventi sugli ingressi.
- Il progetto di circuiti digitali può richiedere che la modifica dello stato avvenga in modo controllato.
 - ▶ Ad esempio, solamente in istanti di tempo ben precisi cosicché eventi transitori non costituiscano eventi significativi.
- Questa esigenza impone l'aggiunta di un ingresso di controllo al bistabile.
- Il segnale applicato all'ingresso di controllo può essere:
 - ▶ Aperiodico
 - ▶ Periodico (denominato Clock)
 - nella maggior parte dei casi;

- Il *clock* è un segnale indipendente caratterizzato da un *periodo di clock* (o *ciclo di clock*) T_{CK} .
 - ▶ Frequenza del clock: $f_{CK} = 1/T_{CK}$;
- Nel periodo T_{CK} il segnale assume il valore logico 1 per un tempo T_H e il valore logico 0 per un tempo T_L
 - ▶ Il rapporto T_H / T_{CK} è detto *duty-cycle*
- Il passaggio dal valore 0 al valore 1 è detto *fronte di salita*
- Il passaggio dal valore 1 al valore 0 è detto *fronte di discesa*



- Per essere riconosciuto correttamente, l'ingresso deve rimanere stabile all'interno di una *finestra* di tempo nell'intorno di un fronte del clock
- Tempo di Set-Up (T_{su})
 - ▶ Intervallo minimo che precede l'evento di clock durante il quale l'ingresso deve essere mantenuto stabile;
- Tempo di Hold (T_H)
 - ▶ Intervallo minimo che segue l'evento di clock durante il quale l'ingresso deve essere mantenuto stabile
- Esempio:

T_H : tempo di *Hold*
 T_{su} : tempo di *Set-Up*
 T_{pro} : tempo di *propagazione*





Bistabili sincroni

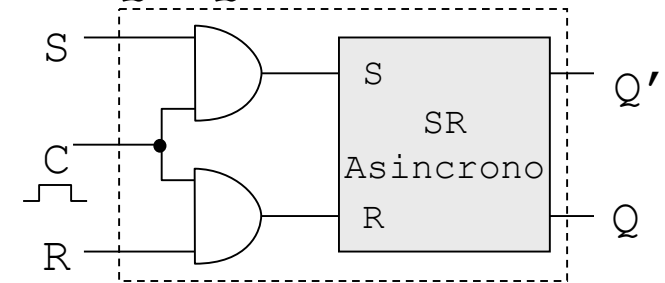
- I fattori che differenziano i bistabili riguardano due aspetti:
 - ▶ La relazione ingresso-stato;
 - ▶ La relazione stato-uscita;
- La relazione *ingresso-stato* (*tipo di temporizzazione*) definisce quando gli ingressi modificano lo stato interno del bistabile.
 - ▶ basato sul livello del segnale di controllo
 - Durante tutto l'intervallo di tempo in cui il segnale di controllo è attivo, qualsiasi variazione sui segnali di ingresso influenza il valore dello stato interno del bistabile.
 - bistabili con commutazione a livello.
 - ▶ basato sul fronte del segnale di controllo
 - Il valore dello stato interno del bistabile viene aggiornato solamente in corrispondenza di un fronte del segnale di controllo.
 - bistabili con commutazione sul fronte (di salita oppure di discesa).

- La relazione *stato-uscita* definisce quando lo stato aggiorna le uscite.
 - ▶ basato sul livello del segnale di controllo
 - Durante tutto l'intervallo di tempo in cui il segnale di controllo è attivo un cambiamento dei segnali di ingresso modifica oltre allo stato interno anche le uscite.
 - Bistabili con questa relazione stato-uscita sono denominati LATCH
 - Il segnale di controllo è solitamente chiamato *enable*.
 - ▶ basato sul fronte del segnale di controllo
 - Le uscite vengono aggiornate su di un fronte del segnale di sincronismo.
 - Bistabili con questa relazione stato-uscita sono denominati FLIP-FLOP

■ Tabella riassuntiva

		Relazione <i>Stato-Uscita</i>	
		Livello	Fronte
Relazione <i>Ingresso-Stato</i>	Fronte		Flip-Flop con commutazione sul fronte
	Livello	Latch con Enable	Flip-Flop con commutazione a livello (Master-Slave)

- Il *latch SR* è ottenuto aggiungendo al bistabile asincrono SR un circuito di controllo.
 - ▶ Sul livello alto di C una variazione sugli ingressi modifica lo stato interno e lo stato interno modifica le uscite Q e Q' .
 - $C=1$ modalità *trasparente*;
 - $C=0$ modalità *opaca*;



- Tabella delle transizioni
- Tabella delle eccitazioni
- Espressione logica

C	S	R	Q^*
0	-	-	Q
1	0	0	Q
1	0	1	0
1	1	0	1
1	1	1	-

Q	Q^*	C	S	R
0	0	0	-	-
1	1	0	-	-
0	0	1	0	-
0	1	1	1	0
1	0	1	0	1
1	1	1	-	0

$$Q^* = C' Q + C (S + R' Q)$$

Nota: l'espressione logica è ricavata dalla tabella delle transizioni

Latch: *D*

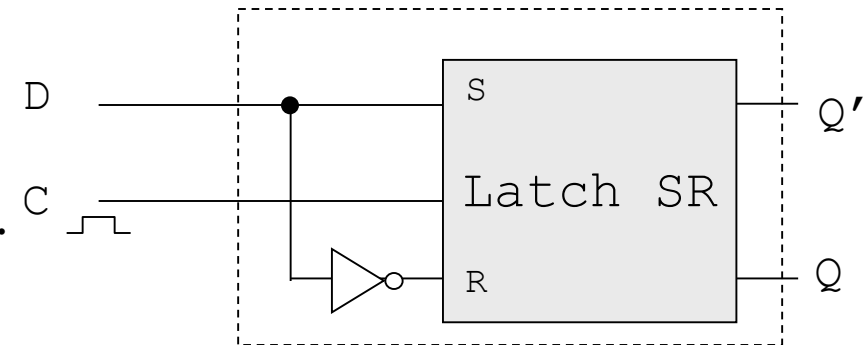


POLITECNICO
MILANO 1863

- Il *latch D* è ottenuto a partire da un *latch SR* imponendo che $S=R'$

- **D: Delay o Data**

- $C=1$ modalità *trasparente*;
 - Q segue l'ingresso.
- $C=0$ modalità *opaca*;
 - Q mantiene l'ultimo ingresso letto.



- Tabella delle transizioni

C	D	Q^*
0	–	Q
1	0	0
1	1	1

- Tabella delle eccitazioni

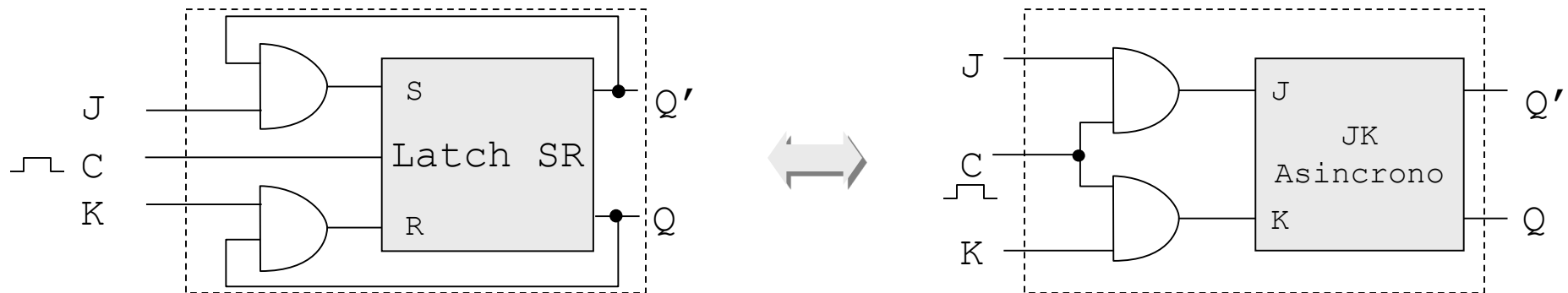
Q	Q^*	C	D
0	0	0	–
1	1	0	–
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	1

- Espressione logica

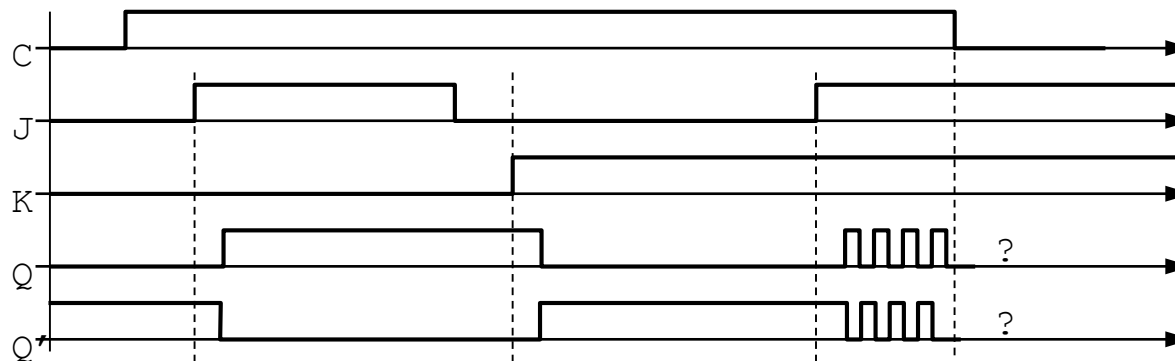
$$Q^* = C' Q + CD$$

Nota: l'espressione logica è ricavata dalla tabella delle transizioni

- I latch, spesso, non consentono di garantire un comportamento affidabile nella realizzazione di una data funzionalità.
- Esempio: Eliminare la configurazione non ammissibile del *latch SR*. Il nuovo *latch* è detto *JK*; imponendo che $J=K=1$, il valore dello stato viene invertito.
 - Per $J=K=1$ si ottiene $Q^*=Q'$;

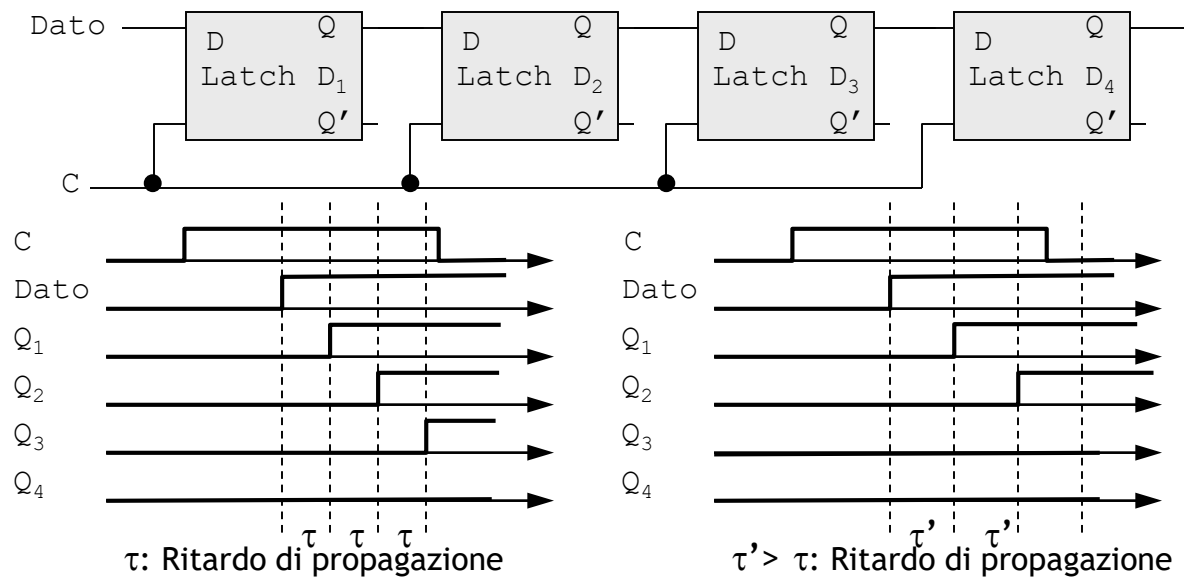


- Quello che si ottiene non è un nuovo *Latch*.
 - ▶ Analisi del comportamento del *latch JK* realizzato:



- Si osserva che per $J=K=1$ il latch ha un comportamento instabile
 - ▶ Le uscite Q e Q' hanno un comportamento oscillatorio ed il valore risultante quando J , K o C cambiano non è noto a priori.
 - corsa critica.
- Vincolo sulla complementazione:
 - ▶ Un solo cambiamento di stato per ciclo di clock per evitare l'effetto di propagazione indesiderato tra uscite ed ingresso.

■ Esempio 2: *shift-register*



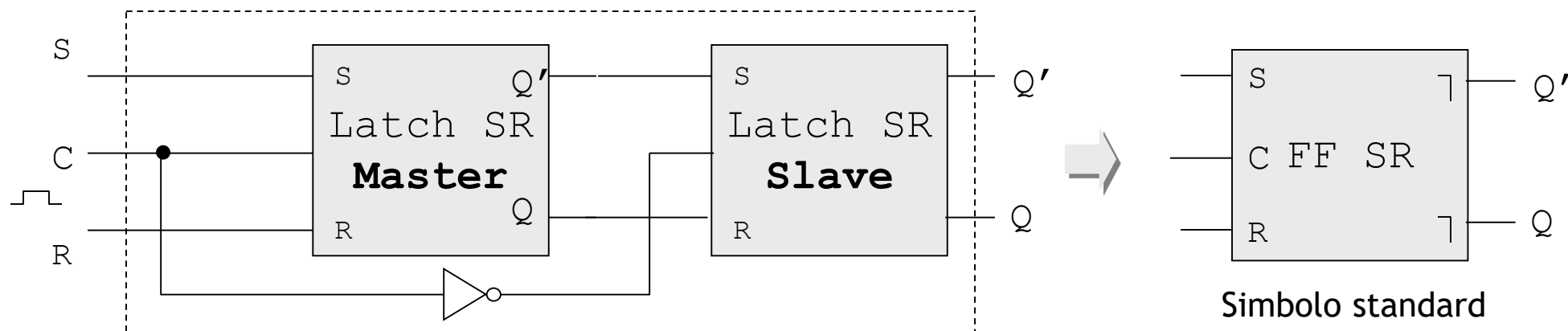
■ 2 Problemi:

- ▶ Non produce una singola traslazione di un bit
 - Non rispetta le specifiche;
- ▶ Il risultato dipende:
 - sia dal ritardo di propagazione dei latch;
 - sia dall'ampiezza del valore alto su C.

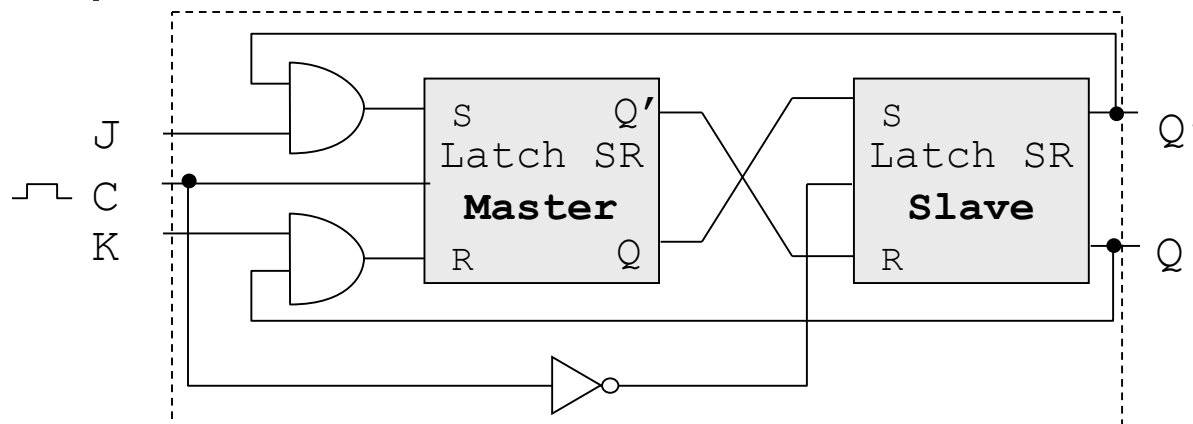
- Per evitare l'effetto di propagazione indesiderata, i bistabili sincroni vengono modificati in modo che lo stato possa modificare le uscite solo in corrispondenza di un evento del segnale di controllo.
- Flip-Flop:
 - ▶ Relazione *stato-uscita* (aggiornamento della uscita):
 - sul fronte.
 - ▶ Relazione *ingresso-stato* (aggiornamento dello stato):
 - a livello (Flip-Flop a livello o *pulse-triggered* o *master-slave*)
 - a fronte (Flip-Flop con commutazione sul fronte o *edge-triggered*).

- I flip-flop master-slave vengono realizzati utilizzando due latch in cascata che hanno il segnale di sincronismo in contrapposizione di fase.
 - ▶ Il primo latch sincrono è il latch principale (*master*).
 - ▶ Il secondo latch sincrono è il latch ausiliario (*slave*).
 - ▶ I due latch lavorano in contrapposizione di fase
 - Il percorso di propagazione ingresso uscita non è continuo

■ Flip-flop master-slave SR.



Flip-flop master-slave JK:



- Tabella delle transizioni
- Tabella delle eccitazioni
- Espressione logica

J	K	Q^*
0	0	Q
0	1	0
1	0	1
1	1	Q'

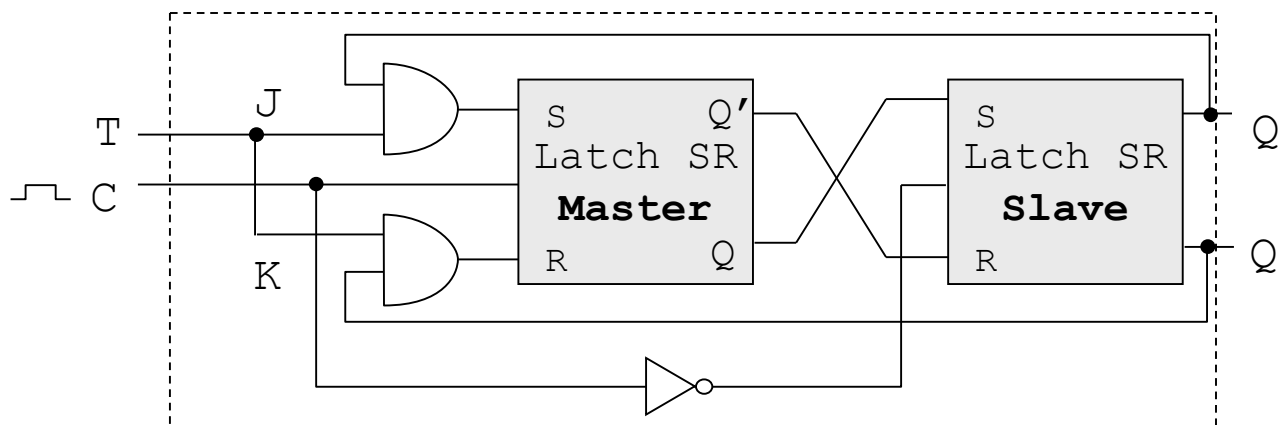
Q	Q^*	J	K
0	0	0	—
0	1	1	—
1	0	—	1
1	1	—	0

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

Nota: l'espressione logica è ricavata dalla tabella delle transizioni

Il cambiamento delle uscite avviene nel passaggio da 1 a 0 di C.

Flip-flop master-slave T:



- Tabella delle transizioni
- Tabella delle eccitazioni
- Espressione logica

T	Q^*
0	Q
1	Q'

Q	Q^*	T
0	0	0
0	1	1
1	0	1
1	1	0

$$Q^* = TQ' + T'Q$$

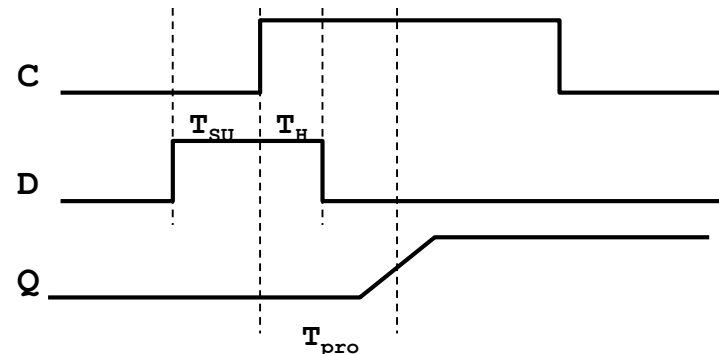
Nota: l'espressione logica è ricavata dalla tabella delle transizioni

Il cambiamento delle uscite avviene nel passaggio da 1 a 0 di C.

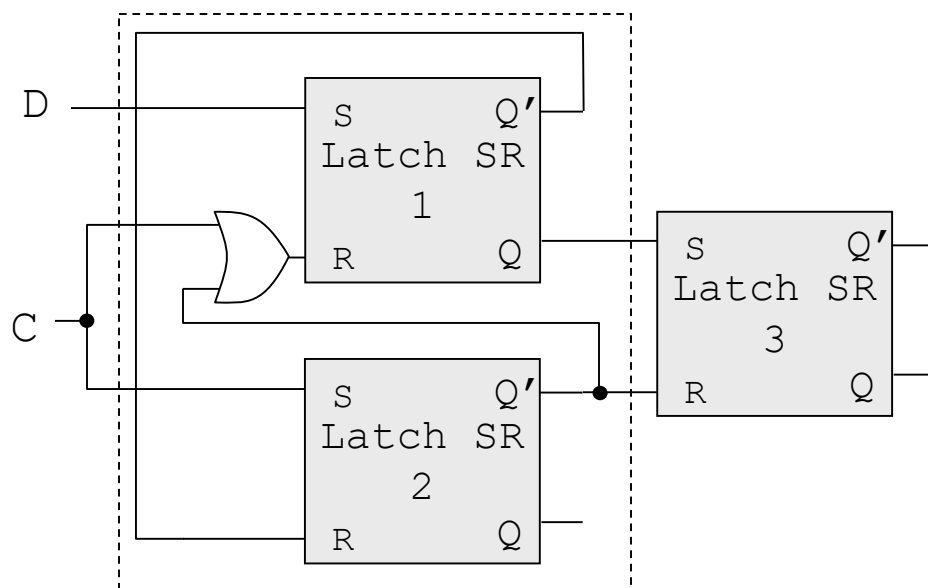
- Funzionamento:
 - ▶ Segnale di sincronismo sul livello alto.
 - Il latch *master* è trasparente e modifica il valore dello stato interno al Flip-Flop in relazione ai valori assunti dai segnali di ingresso.
 - Il latch *slave* è opaco e non consente che le uscite vengano modificate.
 - ▶ Segnale di sincronismo passa al livello basso (fronte di discesa)
 - Il latch *master* passa da trasparente a opaco mantenendo stabile il valore dello stato interno.
 - Il latch *slave* passa da opaco a trasparente e lo stato interno aggiorna le uscite.
- Il comportamento complessivo vede dunque due fasi:
 - ▶ Durante il livello attivo alto del segnale di sincronizzazione il valore degli ingressi (ad esempio, S e R) determinano il valore dello stato interno del latch *master*.
 - ▶ Sul fronte di discesa del segnale di clock viene aggiornato il valore delle uscite del bistabile che rimane fisso fino al successivo fronte di discesa.

- La modalità *master slave* è stata utilizzata per evitare problemi di sincronizzazione dovuti ad un tempo di *hold* maggiore del tempo di propagazione.
- Miglioramenti tecnologici hanno permesso di avere Flip-Flop che commutano sul fronte con tempi di hold pari praticamente a zero

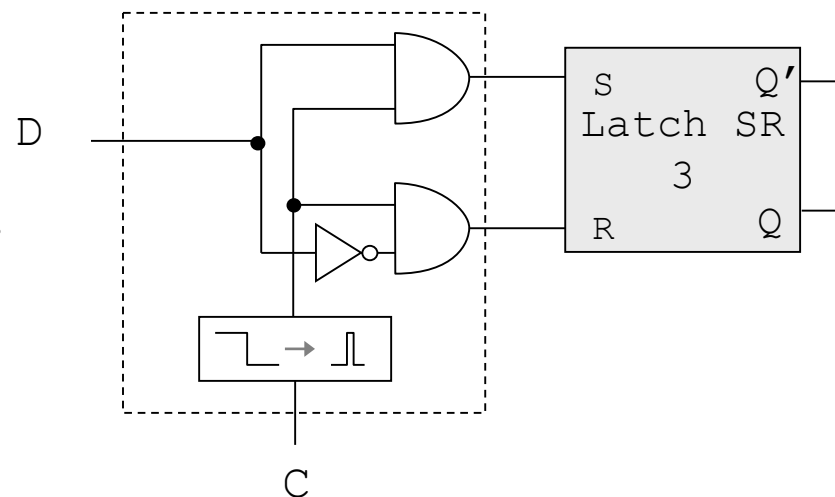
T_H : tempo di *Hold*
 T_{su} : tempo di *Set-Up*
 T_{pro} : tempo di *propagazione*



- I flip-flop che commutano sul fronte Edge-Triggered vengono realizzati producendo, o fisicamente o funzionalmente, la derivata del segnale di clock.
 - Genera un impulso (fisico o funzionale) su di un fronte.
- Flip-Flop D Edge-Triggered



Comportamento funzionalmente equivalente

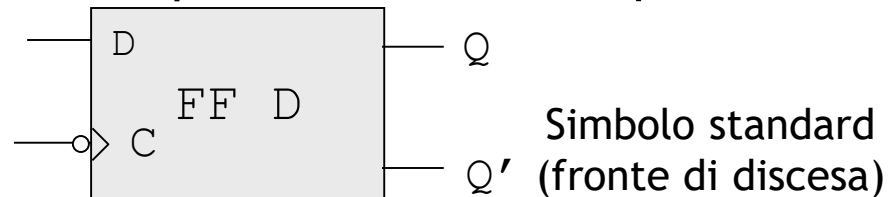


■ Funzionamento:

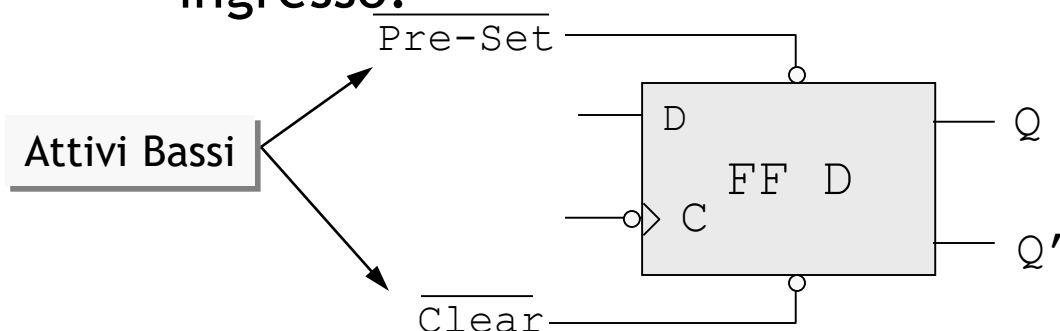
- ▶ Per $C=1$ gli ingressi di Latch SR 3 sono $S=0$ e $R=0$
- ▶ Durante $C=1 \rightarrow 0$, il valore su D attiva il latch SR 1 e, successivamente, il latch SR 2 viene attivato.
 - Se $D=1$, il segnale Q del latch SR 1 viene portato a 1; se $D=0$ il segnale Q del latch SR 1 resta a 0

■ Nota:

- ▶ per $C=1$ il Latch SR 1 può trovarsi nella condizione instabile 11 (a cui consegue $Q=Q'=0$); tale situazione viene risolta nel passaggio di C da 1 a 0 producendo uno stato stabile e deterministico che dipende solo dal valore assunto da D durante la transizione.
- ▶ I tempi di *Hold* e *Set-Up* devono essere rispettati.



- Spesso, nei Flip Flop e nei Latch sono presenti degli ingressi diretti che sono utilizzati per scavalcare gli ingressi dati.
- Gli ingressi diretti sono asincroni.
- Sono utili per:
 - ▶ Stabilire lo stato iniziale del Flip-Flop o del Latch;
 - ▶ Mantenere il Flip-Flop o il Latch in uno stato particolare indipendentemente dai dati presenti ai terminali di ingresso.



Esempio di simbolo standard con ingressi diretti di Pre-Set e Clear.
(FF D su fronte di discesa)

■ Tabella Riassuntiva conclusiva:

- Nota: i bistabili *Latch* e *M/S* considerati sono attivi a livello alto. Analoghe considerazioni possono essere effettuate per elementi attivi a livello basso.

Tipo	Quando campiona gli ingressi	Quando le uscite sono valide
latch senza clock	Sempre	Ritardo di propagazione dal cambiamento degli ingressi
Latch sensibile a livello	Clock alto (T_{SU} e T_H attorno al fronte di discesa)	Ritardo di propagazione dal cambiamento degli ingressi
Flip-Flop master/slave	Transizione 1→0 del Clock (T_{SU} e T_H attorno al fronte di discesa)	Ritardo di propagazione dal cambiamento dal fronte di discesa del clock
Flip-Flop attivo sul fronte di salita	Transizione 0→1 del Clock (T_{SU} e T_H attorno al fronte di salita)	Ritardo di propagazione dal cambiamento dal fronte di salita del clock
Flip-Flop attivo sul fronte di discesa	Transizione 1→0 del Clock (T_{SU} e T_H attorno al fronte di discesa)	Ritardo di propagazione dal cambiamento dal fronte di discesa del clock

Tabelle delle Transizioni e delle Eccitazioni

■ Tabelle delle Transizioni:

S	R	Q^*
0	0	Q
0	1	0
1	0	1
1	1	—

J	K	Q^*
0	0	Q
0	1	0
1	0	1
1	1	Q'

D	Q^*
0	0
1	1

T	Q^*
0	Q
1	Q'

■ Tabelle delle Eccitazioni:

Q	Q^*	S	R
0	0	0	—
0	1	1	0
1	0	0	1
1	1	—	0

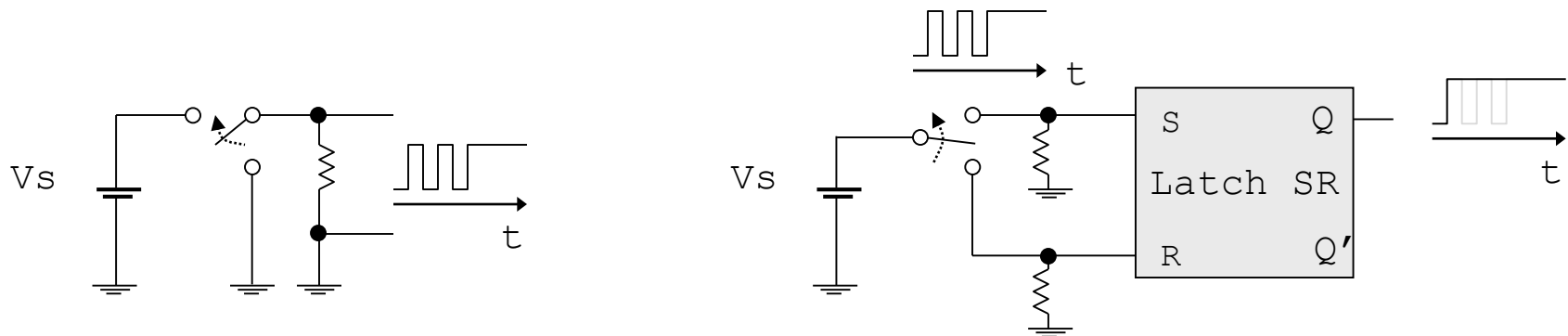
Q	Q^*	J	K
0	0	0	—
0	1	1	—
1	0	—	1
1	1	—	0

Q	Q^*	D
0	0	0
0	1	1
1	0	0
1	1	1

Q	Q^*	T
0	0	0
0	1	1
1	0	1
1	1	0

■ *Bistabile SR:*

- ▶ Utilizzato per filtraggi di segnali provenienti da dispositivi che possono generare transitori indesiderati.
 - Es: circuiti antirimbalzo.



■ *Latch RS e D:*

- ▶ Usati come elementi di memoria in sistemi a clock stretto: meglio non utilizzarli.
- ▶ Il latch RS è il blocco fondamentale per altri tipi di Flip-Flop.



Utilizzo dei bistabili

■ *Flip Flop JK:*

- ▶ Utile come blocco funzionale
- ▶ Usato per costruire *Flip Flop D* e *T*
- ▶ In logica TTL è il più semplice elemento di memoria per realizzare una funzione sequenziale $f(I_n, Q, Q^+)$
- ▶ Due ingressi complicano le connessioni
- ▶ Non usare mai i FF J-K se sono a disposizione FF attivi sul fronte, per il problema dei transitori (alee) - problema della *cattura degli 1*-

■ *Flipflop D:*

- ▶ Minimizza le connessioni, è il più utilizzato in dispositivi VLSI CMOS
- ▶ Il più facile da usare
- ▶ La miglior scelta per un progetto sequenziale

■ *Flipflop T:*

- ▶ In realtà non esiste (è fatto con JK)
- ▶ Va molto bene per realizzare contatori