



Gli elementi di memoria: i bistabili I registri

Bistabili Asincroni
Bistabili Sincroni: Latch e Flip-Flop
Registri

versione del 18/11/03



Circuiti sequenziali

- 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**.

- 2 -



Circuiti sequenziali e bistabili

- 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 gli ingressi ne determinano lo stato.

- 3 -



Bistabili: classificazione

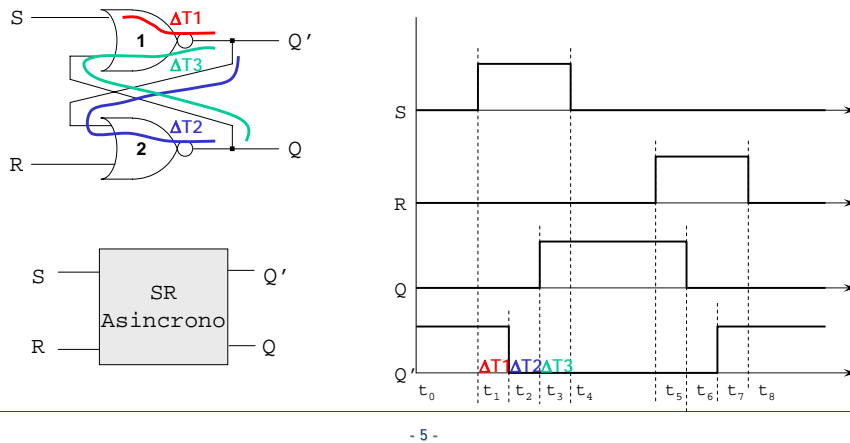
- Classificazione dei **bistabili**:
 - **Asincroni**
 - Sono privi di un segnale di sincronizzazione e **modificano il loro stato** rispondendo **direttamente ad eventi sui segnali di ingresso**.
 - **Sincroni**
 - sono **sensibili ad un segnale di controllo** (spesso il clock) e la transizione da uno 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 master-slave (a livello o pulse-triggered)
 - » flip-flop edge-triggered (a fronte)

- 4 -



Bistabili asincroni: SR

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



- 5 -



Bistabile asincrono SR - Analisi di funzionamento (1)

- 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 $Q=1, Q'=0$

- 6 -



Bistabile asincrono SR - Analisi di funzionamento (2)

- 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 $Q=0, Q'=1$

- 7 -



Bistabili asincroni: SR

- 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.

- 8 -



Bistabili asincroni SR: ingressi non ammissibili

- Applicando contemporaneamente su **S** e **R** un **valore 1** il circuito si porta in uno **stato instabile** con $Q=0, Q'=0$; tale configurazione non è ammissibile. Infatti
 - nel passaggio degli ingressi da 11 a 00, non è possibile identificare chi tra S o R cambia per primo;
 - il bistabile asincrono ritorna quindi **in modo imprevedibile** allo stato $Q=0$ e $Q'=1$ oppure allo stato $Q=1$ e $Q'=0$
 - Questa condizione è chiamata **corsa critica** (*race condition*) o **transizione non-deterministica**

- 9 -



Descrizione del comportamento dei bistabili

- Tabella delle transizioni** (o mappa di Karnaugh):
 - Per i bistabili sincroni spesso vengono riportate le sole configurazioni del clock attivo
 - **ingressi**: ingressi primari i^t , stato presente Q^t
 - **uscita**: stato prossimo Q^{t+1}
- Tabella delle eccitazioni**
 - Per i bistabili sincroni spesso vengono riportate le sole configurazioni del clock attivo
 - **ingressi**: stato presente Q^t , stato prossimo Q^{t+1}
 - **uscita**: configurazione degli ingressi primari che realizza la transizione
- Equazione di funzionamento** (espressione logica)
 - ricavata dalla tabella delle transizioni
 - se sono presenti anche le configurazioni per clock non attivo, in generale viene sintetizzata in modo separato per clock attivo e clock non attivo

- 10 -



Bistabili asincroni SR: descrizioni del comportamento

- Rappresentazioni del comportamento di un bistabile SR

Tabella delle transizioni

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

Tabella delle eccitazioni

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

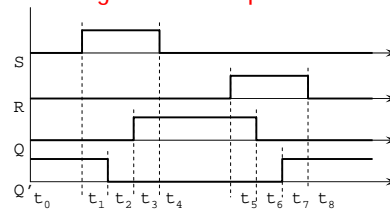
Espressione logica

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

Con vincolo $S=R \neq 1$

Q*: stato prossimo
Q: stato presente

Diagramma Temporale



- 11 -



Sincronismo

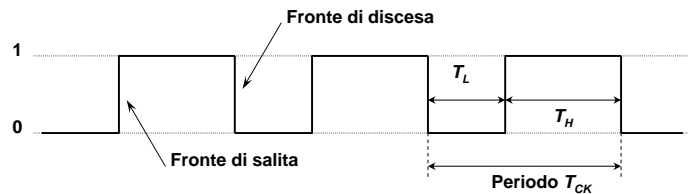
- Un bistabile asincrono modifica il proprio stato solo in relazione ad eventi sugli ingressi
- Il progetto di circuiti digitali può richiedere che la **modifica dello stato avvenga in modo controllato**
 - Ad esempio, solamente in istanti di tempo ben precisi cosicché eventi transitori non costituiscano 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

- 12 -



Segnale di clock

- 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*



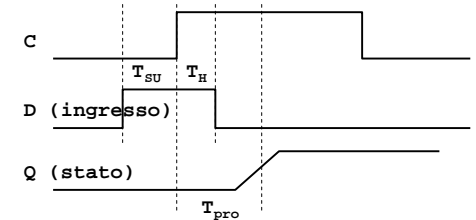
- 13 -



Tempi di Hold e Set-Up

- Per essere riconosciuto correttamente, un ingresso primario di un bistabile 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
- Ad esempio

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



- 14 -



Bistabili sincroni: *Relazione ingresso-stato*

- I fattori che differenziano i bistabili sincroni riguardano due aspetti:
- La relazione **ingresso-stato** (quando gli ingressi sono efficaci)
 - La relazione **stato-uscita** (quando vengono modificate le uscite)
- 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).

- 15 -



Bistabili sincroni: *Relazione stato-uscita*

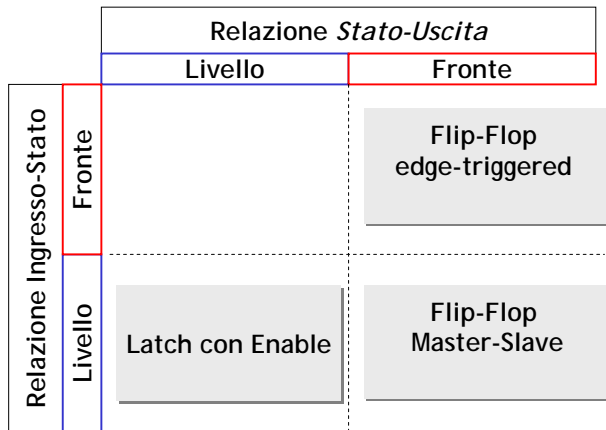
- La relazione **stato-uscita** definisce quando lo stato aggiorna le uscite
 - basato sul **livello** del segnale di controllo
 - Durante tutto l'intervallo di tempo in cui il segnale di controllo è attivo un cambiamento dei segnali di ingresso modifica oltre allo stato interno anche le uscite.
 - Bistabili con questa relazione stato-uscita sono denominati **LATCH**
 - Il segnale di controllo è solitamente chiamato *enable*.
 - Le uscite cambiano quando cambiano gli ingressi
 - 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**
 - Le uscite cambiano in corrispondenza di un evento del clock

- 16 -



Bistabili sincroni

- Tabella riassuntiva



- 17 -



Latch: SR

- 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*;

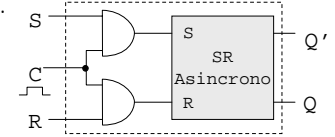


Tabella delle transizioni

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

hold
reset
set
not allowed

Tabella delle eccitazioni

| 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 |

Espressione logica

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

- 18 -



Latch: D

- 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.

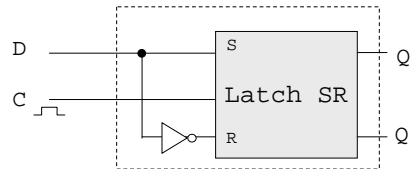


Tabella delle transizioni

| 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

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

- 19 -



Latch: JK

- Il *latch JK* è simile ad un SR, ma con la configurazione $J=K=1$ (con $C=1$) il valore dello stato viene invertito
 - Per $J=K=1$ si ottiene $Q^*=Q'$

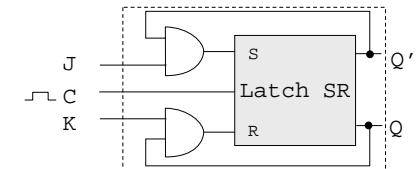


Tabella delle transizioni

| C | J | K | Q* |
|---|---|---|----|
| 0 | - | - | Q |
| 1 | 0 | 0 | Q |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | Q' |

hold
reset
set
toggle

Tabella delle eccitazioni

| Q | Q* | C | J | K |
|---|----|---|---|---|
| 0 | 0 | 0 | - | - |
| 1 | 1 | 0 | - | - |
| 0 | 0 | 1 | 0 | - |
| 0 | 1 | 1 | 1 | - |
| 1 | 0 | 1 | - | 1 |
| 1 | 1 | 1 | - | 0 |

Espressione logica

$$Q^* = C'Q + C(K'Q + JQ')$$

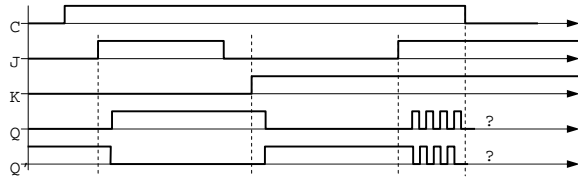
- 20 -



Flip-Flop

- I latch, spesso, non consentono di garantire un comportamento affidabile nella realizzazione di una data funzionalità.

- **Esempio 1: analisi del comportamento del latch JK**



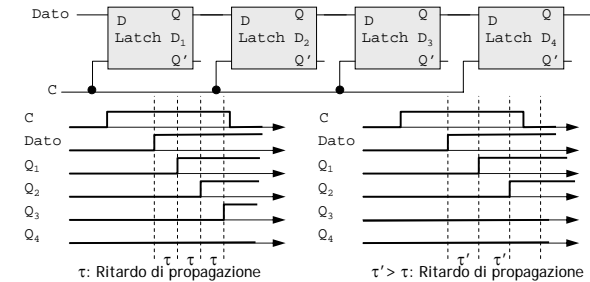
- Per $J=K=1$ il bistabile ha un comportamento instabile
 - Le uscite Q e Q' hanno un comportamento oscillatorio ed il valore risultante quando J , K o C cambiano non è noto a priori (corsa critica)
- Per un funzionamento corretto con gli ingressi $J=K=1$:
 - Un solo cambiamento di stato per ciclo di clock per evitare l'effetto di propagazione indesiderato tra uscite ed ingresso.

- 21 -



Flip-Flop

- **Esempio 2: shift-register**



- **Due problemi:**
 - Non produce una singola traslazione di un bit (non rispetta le specifiche)
 - Il risultato dipende:
 - sia dal ritardo di propagazione dei latch;
 - sia dalla durata del valore alto su C .

- 22 -



Flip-Flop

- 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 (fronte) del segnale di controllo**

- **Flip-Flop:**

- Relazione **stato-uscita** (aggiornamento della uscita):
 - sul fronte
- Relazione **ingresso-stato** (aggiornamento dello stato):
 - a livello (Flip-Flop *master-slave*)
 - a fronte (Flip-Flop *edge-triggered*)

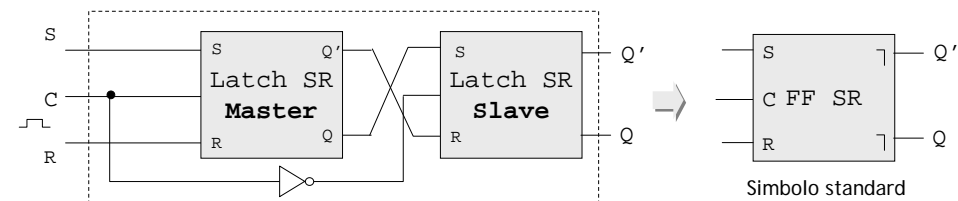
- 23 -



Flip-Flop: SR Master-Slave

- 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 chiamato latch principale (*master*).
 - Il secondo latch sincrono è chiamato 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 (fronte di discesa)



- 24 -



Flip-Flop: D Master-Slave

Flip-flop master-slave D (fronte di discesa)

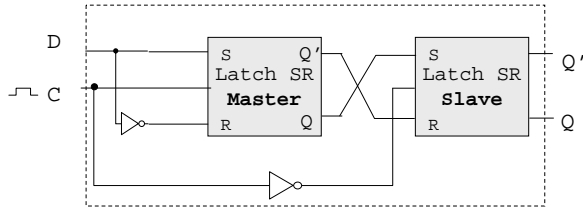


Tabella delle transizioni

| D | Q* |
|---|----|
| 0 | 0 |
| 1 | 1 |

C=1

Tabella delle eccitazioni

| Q | Q* | D |
|---|----|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Espressione logica

$$Q^* = D$$



Flip-Flop: JK Master-Slave

Flip-flop master-slave JK (fronte di discesa)

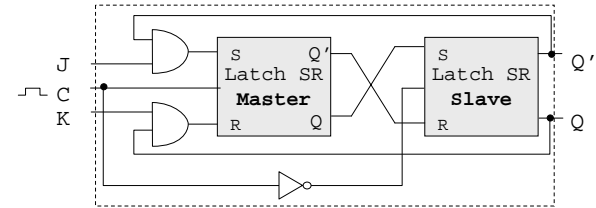


Tabella delle transizioni

| J | K | Q* |
|---|---|-----------|
| 0 | 0 | Q hold |
| 0 | 1 | 0 reset |
| 1 | 0 | 1 set |
| 1 | 1 | Q' toggle |

C=1

Tabella delle eccitazioni

| Q | Q* | J | K |
|---|----|---|---|
| 0 | 0 | 0 | - |
| 0 | 1 | 1 | - |
| 1 | 0 | - | 1 |
| 1 | 1 | - | 0 |

Espressione logica

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



Flip-Flop: T(oggle) Master-Slave

Flip-flop master-slave T (fronte di discesa)

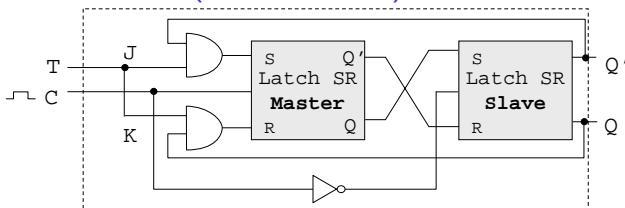


Tabella delle transizioni

| T | Q* |
|---|----|
| 0 | Q |
| 1 | Q' |

Tabella delle eccitazioni

| Q | Q* | T |
|---|----|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Espressione logica

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



Flip-Flop: Master-Slave

□ 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.

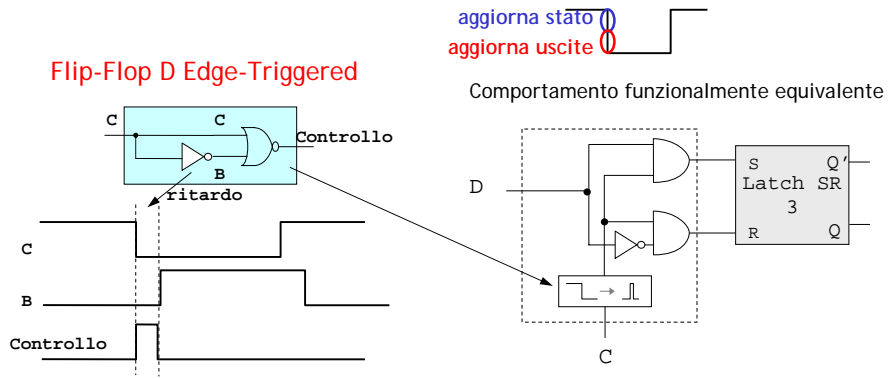
□ Il comportamento complessivo vede dunque due fasi:

- 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.



Flip-Flop: Edge-Triggered

- I flip-flop Edge-Triggered vengono realizzati producendo, o fisicamente o funzionalmente, la derivata del segnale di clock
 - Genera un impulso (fisico o funzionale) in corrispondenza di un fronte

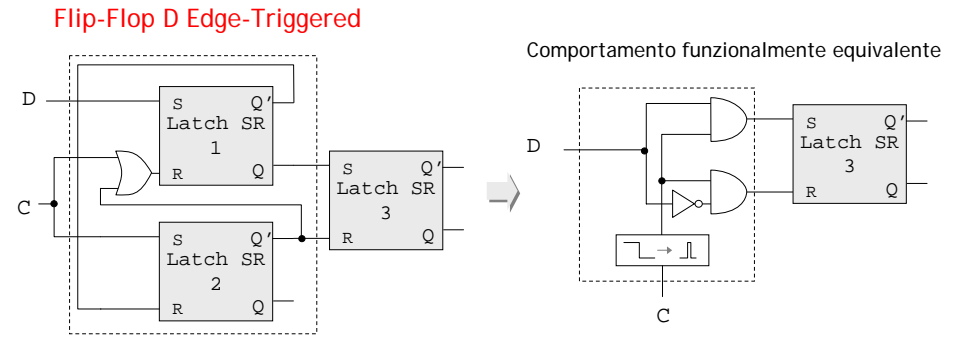


- 29 -



Flip-Flop: Edge-Triggered

- Flip-Flop D Edge-Triggered costituito da 3 latch con comportamento globale equivalente a quello prima visto

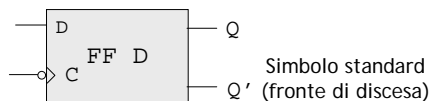


- 30 -



Flip-Flop: Edge-Triggered

- 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 conzione 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.

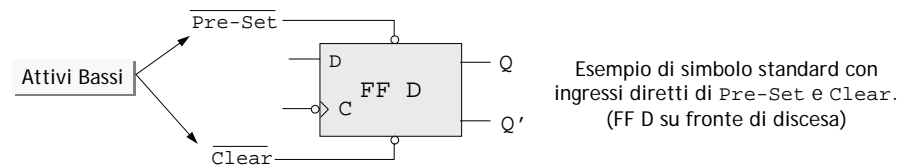


- 31 -



Latch & Flip-Flop: Pre-set e Clear

- 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.



- 32 -



Latch e flip flop

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 | Ingressi stabili | 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 prima del fronte di discesa) | Ritardo di propagazione dal cambiamento degli ingressi |
| Flip-Flop master/slave | Clock alto (T_{SU} e T_H prima del fronte di discesa) | Ritardo di propagazione 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 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 fronte di discesa del clock |

- 33 -



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 |

- 34 -



Registri



Introduzione

Circuiti sequenziali speciali

- Esiste una classe di circuiti sequenziali la cui progettazione potrebbe seguire il processo "classico" di sintesi ma che è più conveniente analizzare in altro modo.
- A questa classe appartengono:
 - **Registri**
 - Memorizzano una definita quantità di informazione
 - Possono operare sul contenuto una o più semplici trasformazioni.
 - » Shift destro/sinistro
 - » Caricamento parallelo/seriale
 - Contatori
 - Attraversano ripetutamente un numero definito di stati
 - » Contatori sincroni
 - » Contatori asincroni

- 36 -



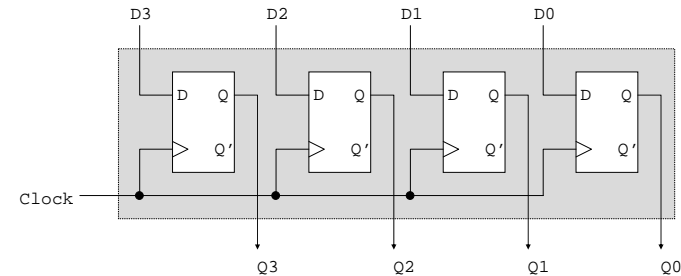
Registri

- Un *registro* è un elemento di memoria in grado di conservare un insieme di bit, denominato *parola*, su cui può eventualmente operare una o più semplici trasformazioni.
 - Benché si possa utilizzare un qualunque tipo di bistabile, per realizzare i registri si preferisce utilizzare *FF D* (*master-slave* o *edge-triggered*).
- I registri si distinguono sulla base dei seguenti aspetti:
 - **Modalità di caricamento dati**
 - Parallelo
 - Seriale
 - **Modalità di lettura dati**
 - Parallelo
 - Seriale
 - **Operazioni di scorrimento sui dati:**
 - a destra e/o a sinistra (aritmetico o non aritmetico) e circolare.



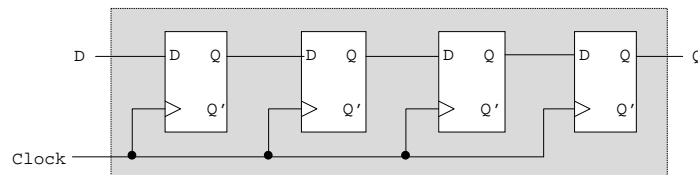
Registri

- Registro *parallelo-parallelo*
 - Esempio di registro a 4 bit.



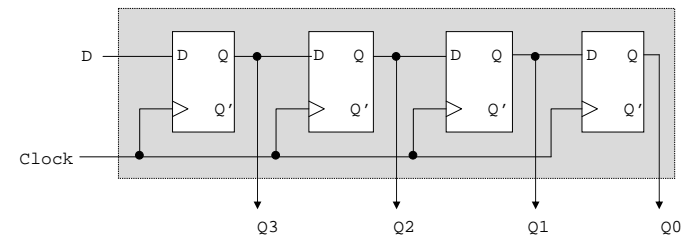
Registri

- Registro *serie-serie* (*Shift Register - Registro a Scorrimento*)
 - Esempio di registro a 4 bit



Registri

- Registro *serie-parallelo*
 - Esempio di registro a 4 bit

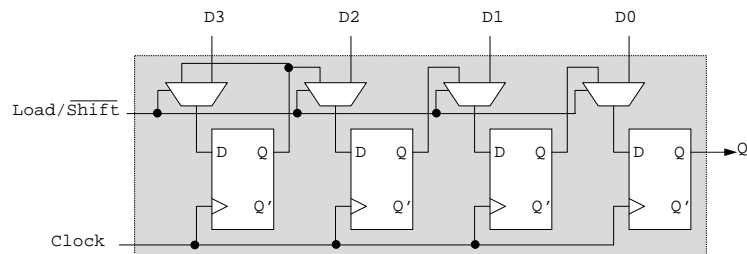




Registri

□ Registro *parallelo-serie*

- Esempio a 4 bit con shift-aritmetico (Shift Destro)
 - In fase di traslazione, ricopia il bit più significativo nella posizione più significativa (estensione del segno)



Registri

□ Registro *circolare* a 4 bit

- Esempio a 4 bit con rotazione a destra
 - In fase di traslazione, trasferisce il bit meno significativo al posto di quello più significativo, spostando i rimanenti di una posizione a destra.

