



Reti logiche sequenziali

Reti logiche sequenziali



Sono quelle reti logiche nelle quali il valore delle uscite in un determinato istante t_i dipende:

- sia dalla condizione di partenza della rete
- sia dal valore degli ingressi all'istante t_i sia dal valore degli ingressi agli istanti $t < t_i$

Per definire il comportamento della rete sequenziale è necessario tenere conto della sua storia passata.

Una rete logica sequenziale ha memoria degli eventi passati

Reti logiche sequenziali

In un generico istante t_i , l'informazione relativa al contenuto di questa memoria è rappresentata nel concetto di stato.

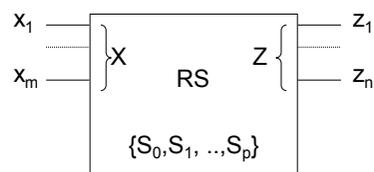
La rete sequenziale, per la fisicità dei suoi componenti, può trovarsi in un numero finito di stati.

Dato uno stato e una configurazione di ingresso il nuovo stato è univocamente determinato.

Considereremo *macchine a stati finiti*

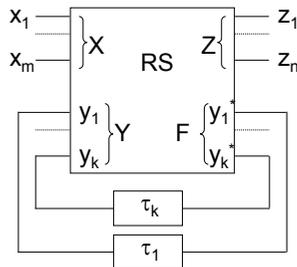
Reti logiche sequenziali

Se supponiamo presenti m ingressi x_1, x_2, \dots, x_m ed n variabili di uscita z_1, z_2, \dots, z_n e se denotiamo con S_0, S_1, \dots, S_p gli stati interni della rete sequenziale, si può tracciare il seguente schema:



Reti logiche sequenziali

Per rappresentare gli stati interni viene scelto un sottoinsieme delle variabili interne alla rete, dette *variabili di stato* e denotate con y_1, y_2, \dots, y_k



Le y_i^* , dette *variabili di eccitazione*, denotano il nuovo valore delle variabili di stato y_i

Le τ_i indicano dei ritardi che si suppone concentrati negli anelli di retroazione

Schema generale di una rete sequenziale

Reti logiche sequenziali

Da un punto di vista algebrico, detto t_0 un istante iniziale nel quale la rete si trova nello stato stabile $Y(t_0)$, e detti t_1, t_2, \dots, t_r gli istanti successivi, la relazione ingressi-uscite può essere espressa come:

$$Z(t_r) = Z[Y(t_0), X(t_0), X(t_1), \dots, X(t_r)]$$

Se si ricorre allo schema che tiene conto delle variabili di stato, si hanno le seguenti equazioni:

$$z_1 = z_1(x_1, \dots, x_m, y_1, \dots, y_k)$$

$$z_n = z_n(x_1, \dots, x_m, y_1, \dots, y_k)$$

$$y_1^* = y_1^*(x_1, \dots, x_m, y_1, \dots, y_k)$$

$$y_k^* = y_k^*(x_1, \dots, x_m, y_1, \dots, y_k)$$

Reti logiche sequenziali

Le equazioni precedenti possiamo riscriverle come:

$$Z=Z(X,Y)$$

$$Y^*=Y^*(X,Y)$$

Uno stato totale $T=\{X;Y\}$ è detto stabile se lo stato futuro e stato presente ad esso corrispondente coincidono, ossia se:

$$Y^*(X;Y)=Y$$

Elementi di memoria: i bistabili

Bistabili

Sono gli elementi in grado di memorizzare le informazioni

Tali elementi sono stabili nei due stati (0,1) e le transizioni da uno stato all'altro avvengono in corrispondenza di un cambiamento degli ingressi.

Le differenze principali tra i diversi bistabili dipendono:

- dal numero di ingressi
- dal modo in cui tali ingressi ne determinano lo stato

Classificazione dei bistabili

Asincroni

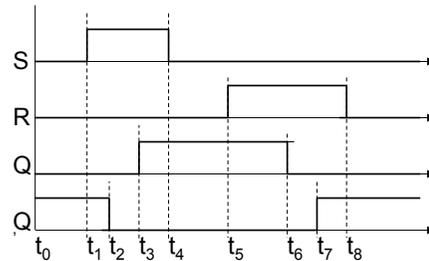
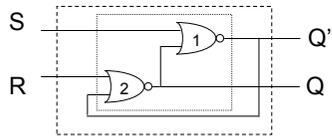
- Sono privi di un segnale di sincronismo e modificano il loro stato in seguito al cambiamento degli ingressi.

Sincroni

- Sono sensibili ad un segnale di controllo e i cambiamenti di stato avvengono solo in corrispondenza di un impulso del segnale di controllo.

Bistabili asincroni

SR



Analisi di funzionamento

$t=t_0$: $S=0, R=0, Q=0, Q'=1$

$t=t_1$: E: $(S=1) \rightarrow I_1(t_1)=(1,0) \rightarrow O_1(t_2)=0 \rightarrow Q'(t_2)=0$

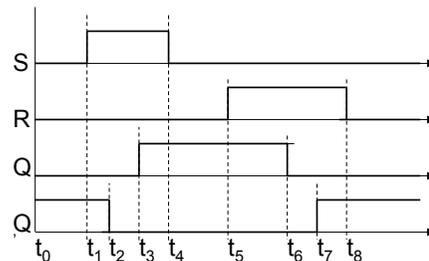
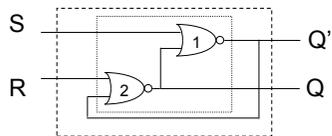
$t=t_2$: $I_2(t_2)=(0,0) \rightarrow O_2(t_3)=1 \rightarrow Q(t_3)=1$

$t=t_3$: $I_1(t_3)=(1,1) \rightarrow O_1(t_3)=0 \rightarrow Q'(t_3)=0$

$t=t_4$: E: $(S=0) \rightarrow I_1(t_4)=(0,1) \rightarrow O_1(t_4)=0 \rightarrow Q'(t_4)=0$

Bistabili asincroni

SR



Analisi di funzionamento

$t=t_5$: E: $(R=1) \rightarrow I_2(t_5)=(1,0) \rightarrow O_2(t_6)=0 \rightarrow Q(t_6)=0$

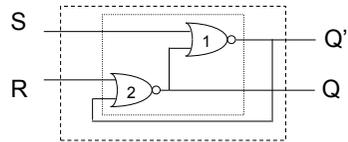
$t=t_6$: $I_1(t_6)=(0,0) \rightarrow O_1(t_7)=1 \rightarrow Q'(t_7)=1$

$t=t_7$: $I_2(t_7)=(1,1) \rightarrow O_2(t_7)=0 \rightarrow Q(t_7)=0$

$t=t_8$: E: $(R=0) \rightarrow I_2(t_8)=(0,1) \rightarrow O_2(t_8)=0 \rightarrow Q(t_8)=0$

Bistabili asincroni

SR



I segnali S e R prendono il nome di Set e Reset

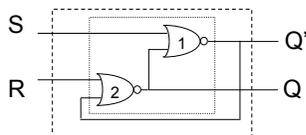
- La configurazione $S=1, R=0$ porta la rete allo stato stabile $Q=1, Q'=0$
- La configurazione $S=0, R=1$ porta la rete allo stato stabile $Q=0, Q'=1$
- La configurazione $S=0, R=0$ lasci inalterato lo stato della rete

La configurazione $S=1, R=1$ non è ammissibile

- La rete si porta nello stato instabile $Q=0, Q'=0$
- Passando da 11 a 00 non è possibile prevedere se la configurazione di uscita sarà 10 o 01
- Questa condizione è detta corsa critica

Bistabili asincroni

SR



Espressione logica

$$Q^* = (S+Q)' \Rightarrow Q^* = S+Q = S+(R+Q)'$$

$$Q^* = S+R'Q \quad \text{Con il vincolo } S=R \neq 1$$

Tabella delle transizioni

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

Q^*

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

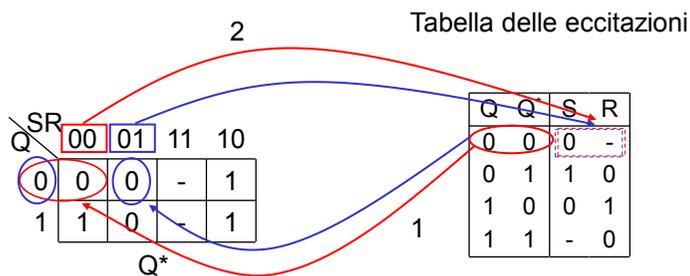
Q^* : Stato futuro

Q: Stato presente

Bistabili asincroni SR

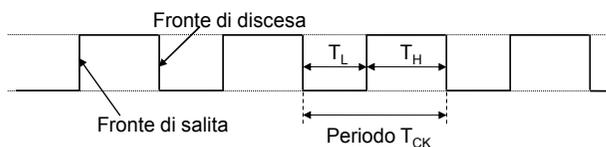
Tabella delle eccitazioni

■ La *Tabella delle Eccitazioni* consente di conoscere la configurazione degli ingressi da applicare affinché possa avvenire una certa transizione di stato



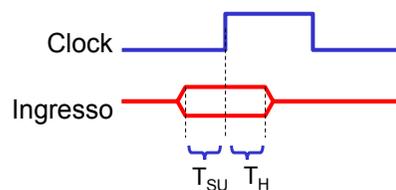
Sincronia

- Un bistabile asincrono modifica il proprio stato solo in presenza di un evento sugli ingressi.
- Il progetto delle reti può richiedere che il cambiamento avvenga in istanti di tempo ben precisi.
- Questa esigenza impone la presenza di un segnale di ingresso di controllo.
- Il segnale che prendiamo in considerazione è periodico, detto clock.



Tempi di Hold e Set-up

- Affinché venga riconosciuto correttamente, un ingresso deve rimanere stabile in una finestra temporale nell'intorno di un fronte del clock
- **Tempo di Set-up** (T_{SU})
 - Minimo intervallo di tempo che precede il fronte del clock durante il quale l'ingresso deve essere mantenuto stabile
- **Tempo di Hold** (T_H)
 - Minimo intervallo di tempo che segue il fronte del clock durante il quale l'ingresso deve essere mantenuto stabile



Bistabili sincroni

- Vengono classificati sulla base della relazione:
 - ingresso-stato
 - Stato-uscita
- La relazione ingresso-stato definisce quando gli ingressi modificano lo stato
 - Sul livello del segnale di controllo
 - Sul fronte del segnale di controllo
- La relazione stato-uscita definisce quando lo stato aggiorna le uscite:
 - Sul livello del segnale di controllo (LATCH)
 - Sul fronte del segnale di controllo (FLIP-FLOP)

Latch SR

È ottenuto aggiungendo un segnale di controllo al bistabile asincrono SR
 Sul livello alto di C una variazione sugli ingressi modifica lo stato e lo stato modifica le uscite

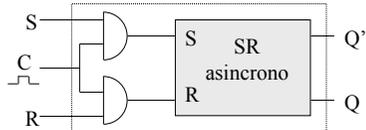


Tabelle delle Transizioni

C	S	R	Q
0	-	-	Q
1	0	0	Q
1	0	1	0
1	1	0	1
1	1	1	-

Tabelle 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

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

Latch D

E' ottenuto da un latch SR imponendo $S=R'$

Per $C=1$, $Q^*=D$

Per $C=0$, $Q^*=Q$

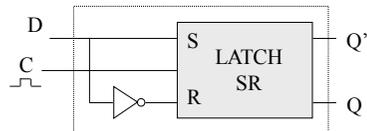


Tabelle delle Transizioni

C	D	Q
0	-	Q
1	0	0
1	1	1

Tabelle delle eccitazioni

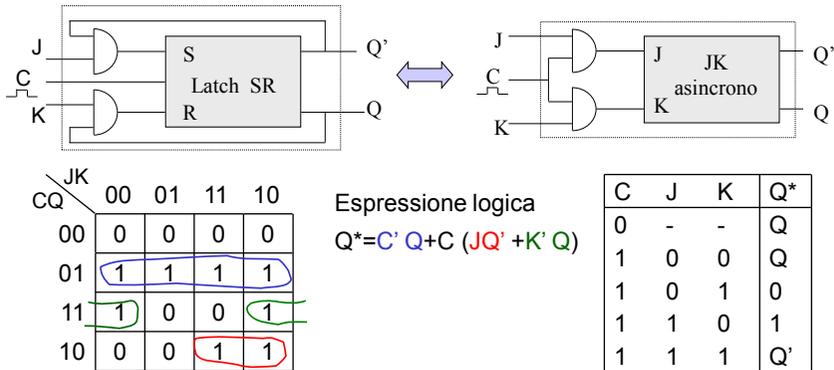
Q	Q*	C	D
0	0	0	-
1	1	0	-
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	1

Espressione logica

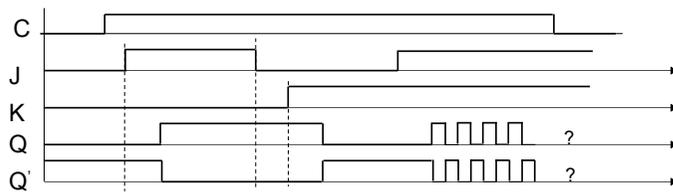
$$Q^* = C' Q + C D$$

Latch JK

Permette di eliminare la configurazione non ammissibile del Latch SR imponendo che in questo caso lo stato venga invertito



Problemi del latch JK



- Il Latch non consente di garantire un comportamento affidabile
- Se $S=1$, $R=1$ il bistabile ha un comportamento instabile, oscillatorio e il valore di Q e Q' quando J , K o C cambiano non è noto a priori
- Per evitare l'effetto di propagazione indesiderata tra uscite ed ingresso si deve imporre il vincolo di un solo cambiamento di stato per ciclo di clock

Flip-Flop

Per evitare l'effetto di propagazione indesiderata, bisognerebbe imporre una dimensione massima all'impulso di clock.

Questa necessità può essere eliminata modificando i bistabili sincroni in modo che lo stato possa modificare le uscite solo in corrispondenza di un fronte del segnale di controllo

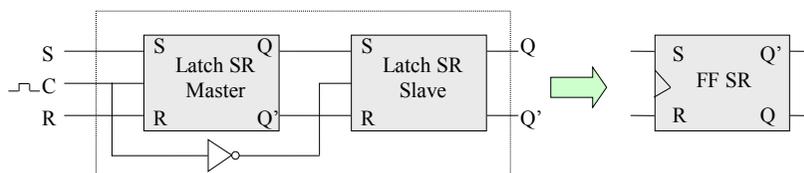
Flip-Flop:

- Relazione stato-uscita (aggiornamento della uscita) sul fronte
- Relazione ingresso-stato (aggiornamento dello stato)
 - A livello (Flip-Flop pulse-triggered o master-slave)
 - A fronte (Flip-Flop edge-triggered)

Flip-Flop Master-Slave

I Flip-Flop master-slave vengono realizzati utilizzando due latch in cascata che hanno il segnale di sincronismo in contrapposizione di fase.

Flip-Flop master-slave SR

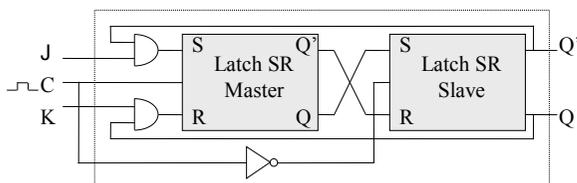


Flip-Flop Master-Slave: Funzionamento

- Segnale di sincronismo $C=1$
 - Il latch master è trasparente e modifica il valore dello stato interno al Flip-Flop in relazione al valore del segnale di ingresso
 - Il latch slave è opaco e non consente che venga modificato il valore delle uscite
- Fronte di discesa del segnale di sincronismo ($1 \rightarrow 0$)
 - Il latch master diventa opaco mantenendo il valore dello stato interno
 - Il latch slave diventa trasparente e lo stato interno aggiorna il valore della uscita

Dunque, durante il livello alto viene aggiornato lo stato interno e sul fronte di discesa viene aggiornata l'uscita che rimane stabile fino al successivo fronte di discesa.

Flip-Flop Master-Slave JK



Tablelle delle Transizioni

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

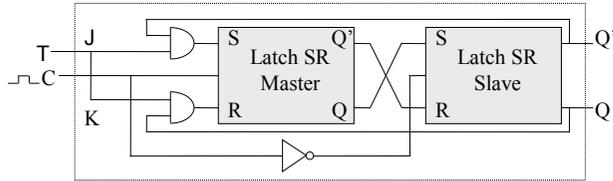
Tablelle delle eccitazioni

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

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

Flip-Flop Master-Slave T



Tablelle delle Transizioni

C	T	Q*
0	-	Q
1	0	Q
1	1	Q'

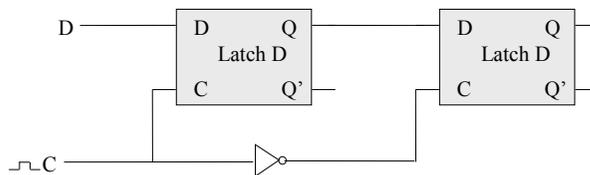
Tablelle delle eccitazioni

Q	Q*	C	T
0	0	0	-
1	1	0	-
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	0

Espressione logica

$$Q^* = C' Q + C (TQ' + T' Q)$$

Flip-Flop master-slave D



Tablelle delle Transizioni

C	D	Q*
0	-	Q
1	0	0
1	1	1

Tablelle delle eccitazioni

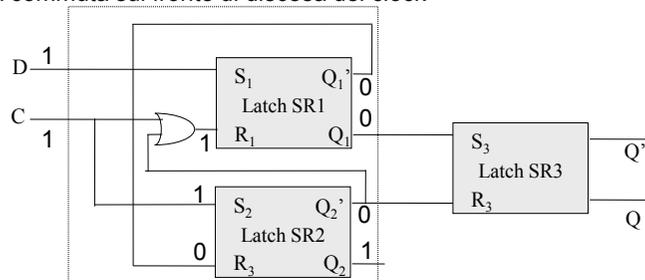
Q	Q*	C	D
0	0	0	-
1	1	0	-
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	1

Espressione logica

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

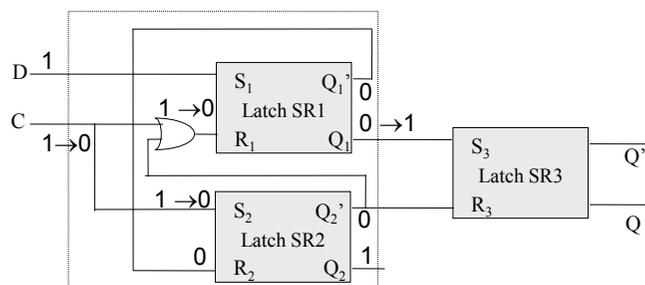
FLIP-FLOP edge-triggered

L'uscita commuta sul fronte di discesa del clock



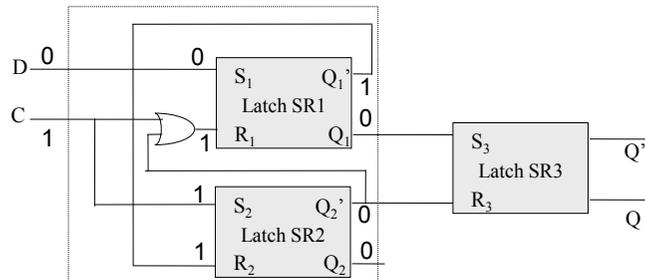
- Per $C=1$ gli ingressi del latch SR3 sono $S=0$, $R=0$ e Q^*
- Se $D=1$, durante l'impulso di controllo ($C=1$) il latch SR1 si trova in uno stato instabile con $Q_1=Q_1'=0$

FLIP-FLOP edge-triggered



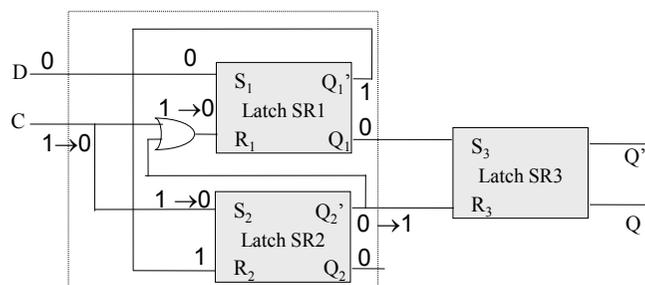
- Durante il fronte di discesa del clock ($C=1 \rightarrow 0$)
 - prima viene attivato SR1 (Q_1 passa a 1);
 - poi viene attivato SR2 (che mantiene i valori precedenti);
 - poiché S3 diventa 1 e R3=0, vengono aggiornati i valori di Q (=1) e Q' (=0).

FLIP-FLOP edge-triggered



- Per C=1 gli ingressi del latch SR3 sono S=0, R=0 e Q*
- Se D=0, durante l'impulso di controllo (C=1) il latch SR2 si trova in uno stato instabile con $Q_2=Q_2'=0$

FLIP-FLOP edge-triggered

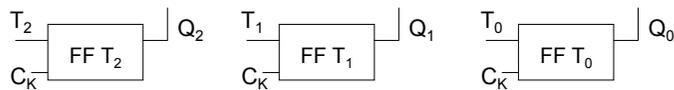


- Durante il fronte di discesa del clock (C= 1→0)
 - prima viene attivato SR1 (che mantiene i valori precedenti) ;
 - poi viene attivato SR2 (Q₂ passa al valore 1);
 - poiché S₃=0 e R₃ diventa 1, vengono aggiornati i valori di Q (=0) e Q' (=1).

Contatori Sincroni

Contatori sincroni

Contatore modulo 8 con FFT



Q_2	Q_1	Q_0	Q_2^*	Q_1^*	Q_0^*	T_2	T_1	T_0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

Q	Q^*	C	T
0	0	0	-
1	1	0	-
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	0

$$T_0 = 1$$

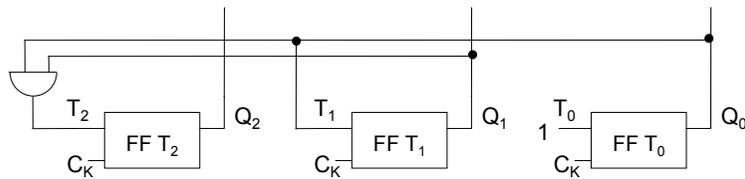
$$T_1 = Q_0$$

$$T_2 = Q_1 Q_0$$

Contatori sincroni

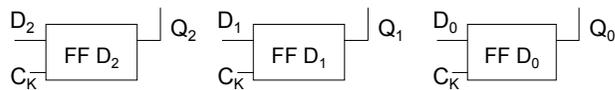
Contatore modulo 8 con FFT

$$T_2 = Q_1 Q_0 \quad T_1 = Q_0 \quad T_0 = 1$$



Contatori sincroni

Contatore modulo 8 con FF D



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

Q ₂	Q ₁	Q ₀	Q ₂ '	Q ₁ '	Q ₀ '	D ₂	D ₁	D ₀
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	1	0	1	1	0	1
1	0	1	1	1	0	1	1	0
1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0

Q ₁ Q ₀	00	01	11	10
Q ₂ 0	1	0	0	1
1	1	0	0	1

$$D_0 = Q_0'$$

Q ₁ Q ₀	00	01	11	10
Q ₂ 0	0	1	0	1
1	0	1	0	1

$$D_1 = Q_1'Q_0 + Q_1Q_0' = Q_1 \oplus Q_0$$

Q ₁ Q ₀	00	01	11	10
Q ₂ 0	0	0	1	0
1	1	1	0	1

$$D_2 = Q_2'Q_1Q_0 + Q_2Q_1' + Q_2Q_0' = Q_2 \oplus Q_1Q_0$$

Contatori sincroni

Contatore modulo 8 con FF D

$$D_2 = Q_2 \oplus Q_1 Q_0 \quad D_1 = Q_1 \oplus Q_0 \quad D_0 = Q_0'$$

