

Reti Logiche Sequenziali

Argomento:

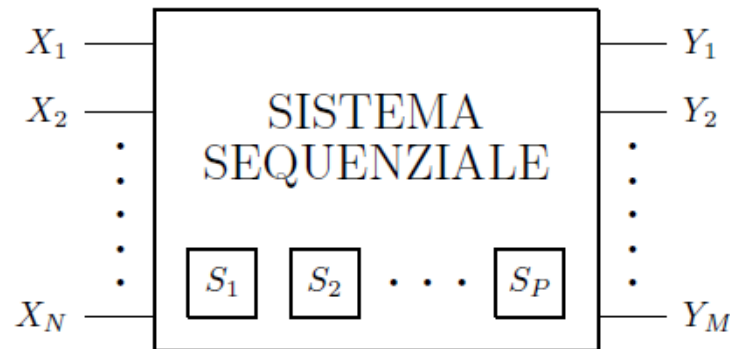
- Latch e flip-flop
- Sintesi

Materiale di studio:

- Capitolo 11.4

Rete logica sequenziale: definizione

Una **rete logica sequenziale** è una rete logica nella quale i valori presenti alle uscite sono **determinati dai valori presenti agli ingressi** e dal **valore dello stato** in cui si trova il sistema.



Una rete logica sequenziale:

- **contiene elementi di memoria**
- è descritta da una tabella di verità in cui si **tiene conto anche dello stato del sistema**

Circuiti sequenziali

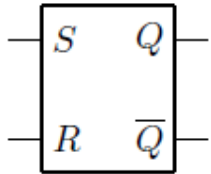
Latch: sensibili al **livello** dei segnali d'ingresso

- Latch di tipo R-S

Flip-flop: sensibili ai **fronti** dei segnali d'ingresso

- Flip-flop di tipo R-S
- Flip-flop di tipo D
- Flip-flop di tipo J-K

Latch di tipo R-S

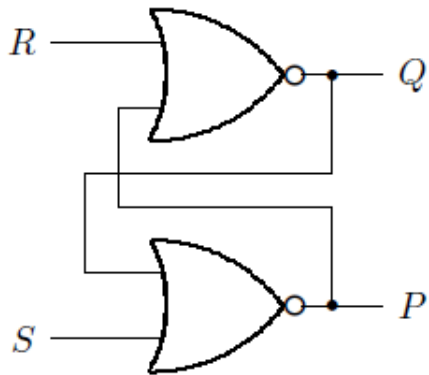


$R = 1 \rightarrow \text{Reset: } Q = 0$

$S = 1 \rightarrow \text{Set: } Q = 1$

Si assume R/S non siano mai entrambi ad 1

Q	R	S	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	0
1	1	1	0



$$Q = \overline{R + \overline{Q}}$$

$$P = \overline{S + Q}$$

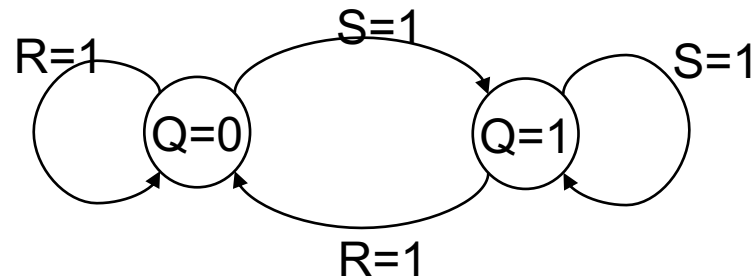
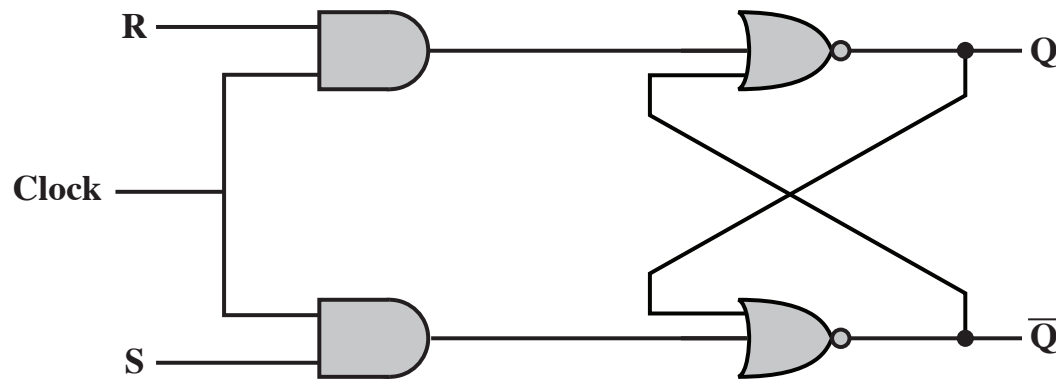


diagramma di stato

Clock

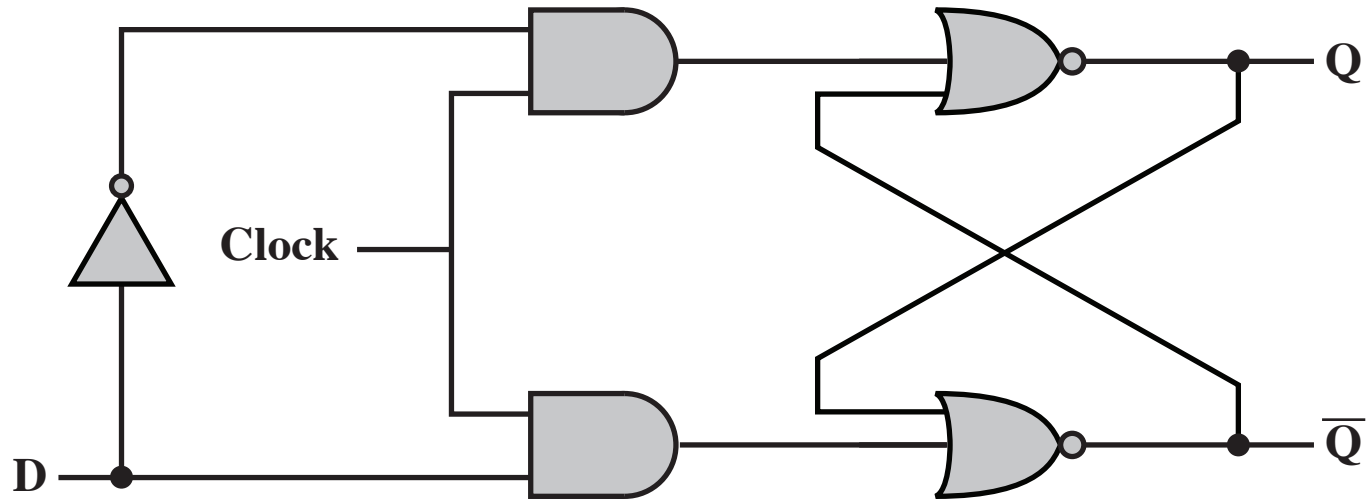
- Un circuito sequenziale può essere:
 - **Asincrono**: l'output è determinato da come cambiano i segnali in ogni istante di tempo.
 - Esempio: Latch R-S
 - **Sincrono**: l'output è determinato dai valori dei segnali in certi istanti di tempo.
 - Gli istanti di tempo sono dati dal segnale di **clock**
 - Esempio: Flip-flop R-S
- Il clock è un segnale che oscilla tra 0 e 1 con una data frequenza.
 - E' il direttore di orchestra che coordina tutti i circuiti sequenziali.

Flip-Flop R-S con clock



- Gli ingressi R e S esercitano il loro effetto solo quando il clock C ha valore logico 1.
- L'output si congela con clock $1 \rightarrow 0$
 - Latch: sensibile al **livello** 0/1 dell'input
 - Flip-flop: sensibile ai **fronte** di discesa del clock

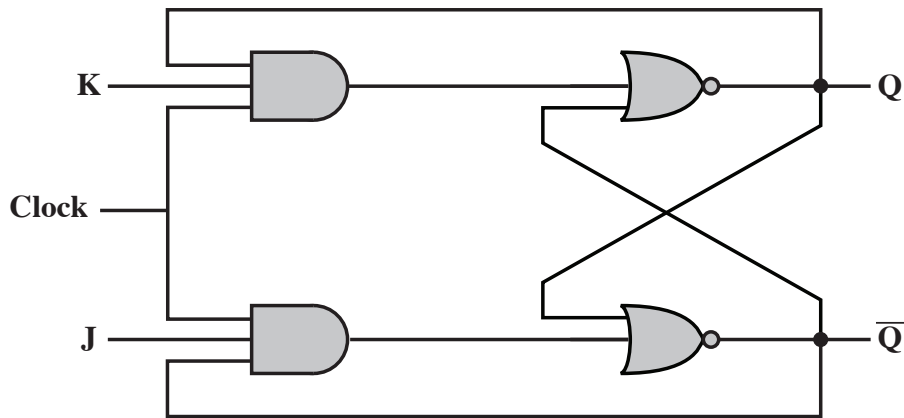
Flip-Flop di tipo D



Nel flip-flop S-R la configurazione $R=S=1$ non è valida.

Il latch D risolve questo problema.

Flip-Flop di tipo J-K



Clock=1

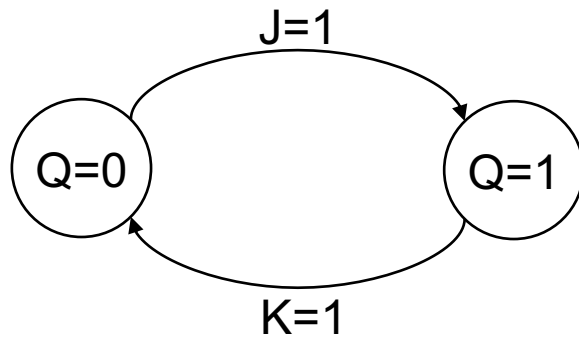


diagramma di stato

Se $Q=1$ e clock $C=1$, può esercitarsi solo l'azione di $K=1$ (**reset**)

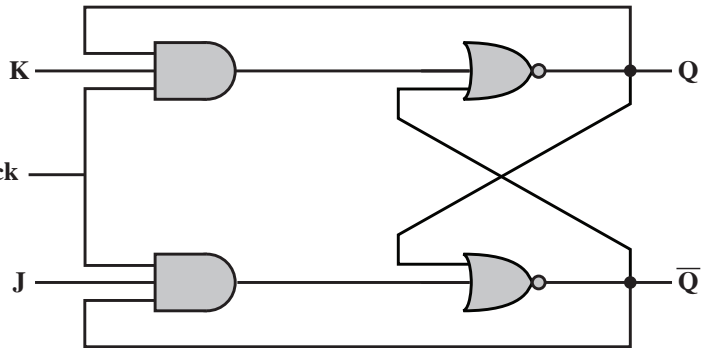
Se $Q=0$ e clock $C=1$, può esercitarsi solo l'azione di $J=1$ (**set**).

Clock=1

Q	K	J	Q'
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

tabella delle transizioni

Flip-Flop di tipo J-K



Se $J=K=Clock=1$: funzione di **toggle**

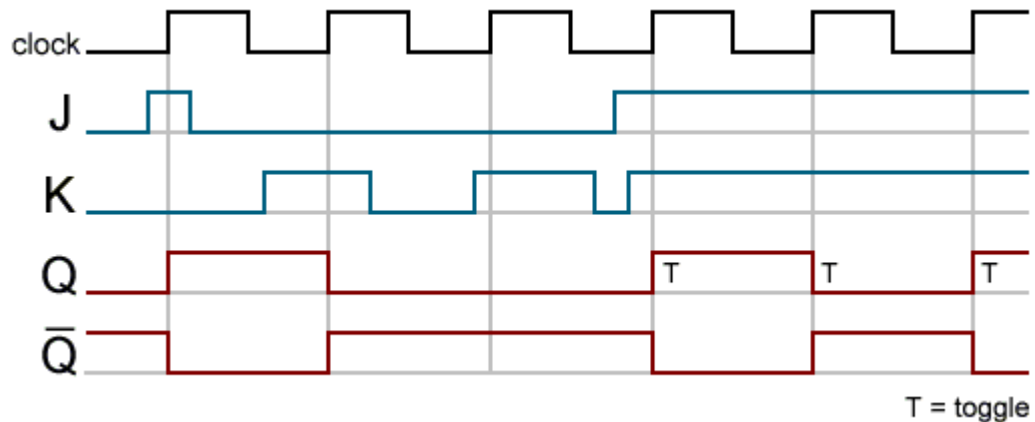
- L'uscita Q è instabile: si inverte **ripetutamente**.

La durata del clock $C=1$ può essere calibrata per ottenere una sola commutazione di Q .

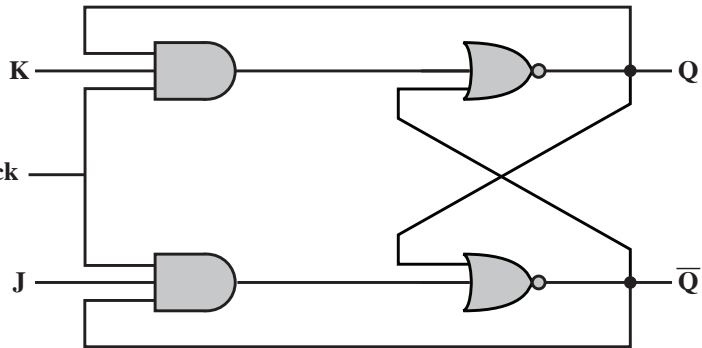
Clock=1

Q	K	J	Q'
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

tabella delle transizioni



Flip-Flop di tipo J-K



Se $J=K=Clock=1$: funzione di **toggle**

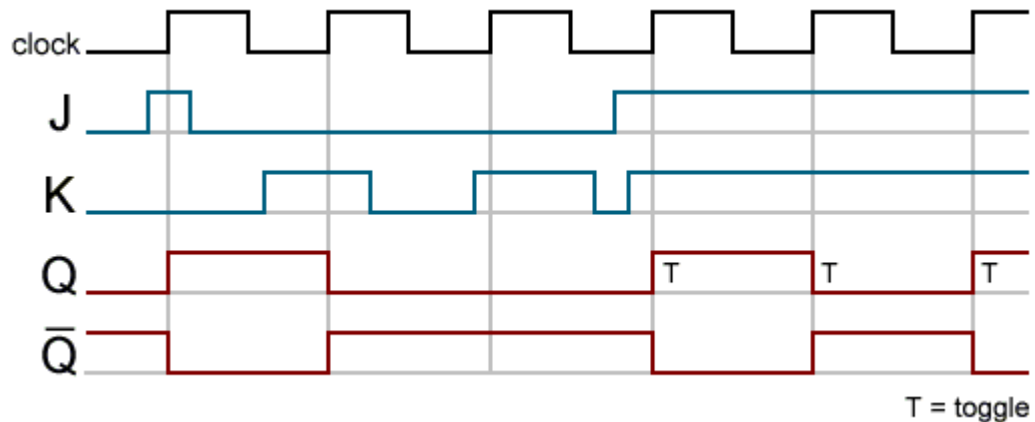
- L'uscita Q è instabile: si inverte **ripetutamente**.

La durata del clock $C=1$ può essere calibrata per ottenere una sola commutazione di Q .

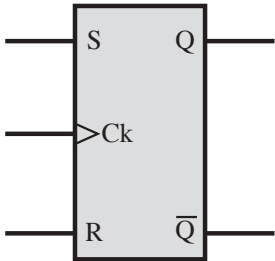
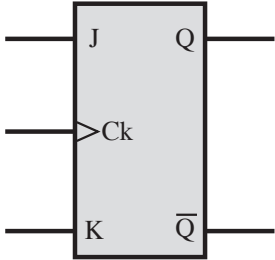
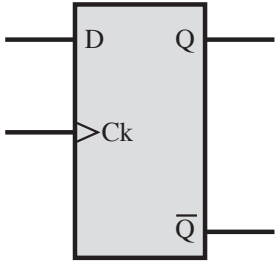
Clock=1

Q	K	J	Q'
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

tabella delle transizioni

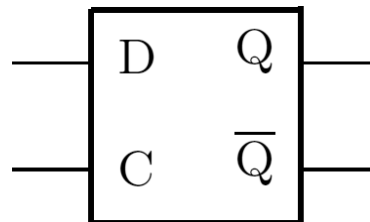


Sommario Flip-Flop

Name	Graphical Symbol	Truth Table															
S-R		<table><tr><th>S</th><th>R</th><th>Q_{n+1}</th></tr><tr><td>0</td><td>0</td><td>Q_n</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>–</td></tr></table>	S	R	Q_{n+1}	0	0	Q_n	0	1	0	1	0	1	1	1	–
S	R	Q_{n+1}															
0	0	Q_n															
0	1	0															
1	0	1															
1	1	–															
J-K		<table><tr><th>J</th><th>K</th><th>Q_{n+1}</th></tr><tr><td>0</td><td>0</td><td>Q_n</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>$\overline{Q_n}$</td></tr></table>	J	K	Q_{n+1}	0	0	Q_n	0	1	0	1	0	1	1	1	$\overline{Q_n}$
J	K	Q_{n+1}															
0	0	Q_n															
0	1	0															
1	0	1															
1	1	$\overline{Q_n}$															
D		<table><tr><th>D</th><th>Q_{n+1}</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	D	Q_{n+1}	0	0	1	1									
D	Q_{n+1}																
0	0																
1	1																

Registri

- Un **registro** è un dispositivo in grado di **memorizzare** (cioè conservare nel tempo) un valore logico. Tale capacità distingue i registri dalle porte logiche.
- Un registro da 1 bit può essere implementati con flip-flop di tipo D

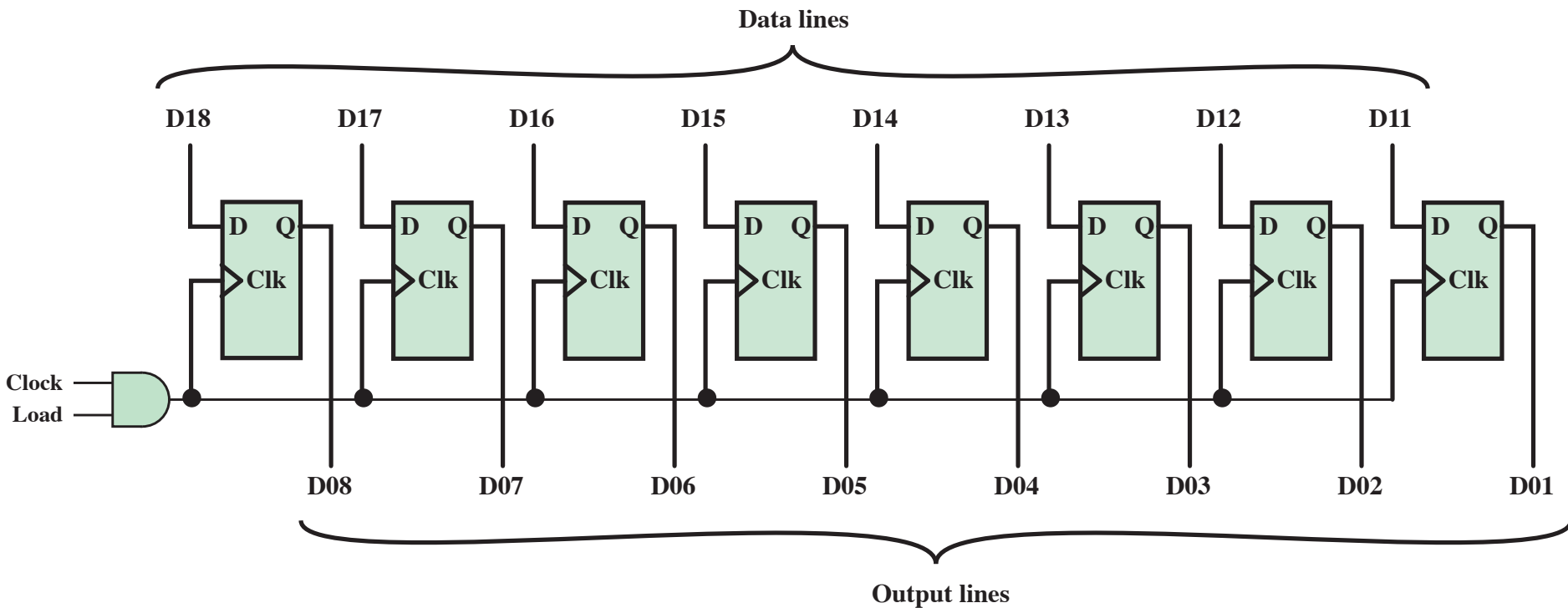


Registri a più bit

I registri possono essere:

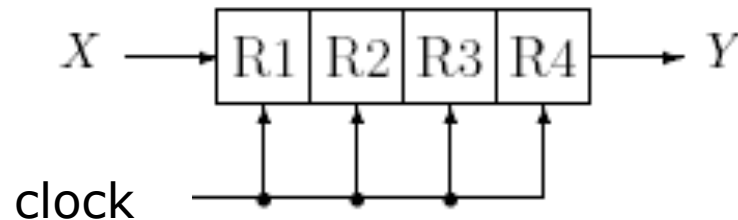
- **Registri paralleli**: un insieme di più memorie da 1 bit che possono essere lette/scritte in **parallelo**
- **Registri a scorrimento** (shift): un insieme di più memorie da 1 bit che possono essere lette/scritte **serialmente**

Registro parallelo a 8 bit



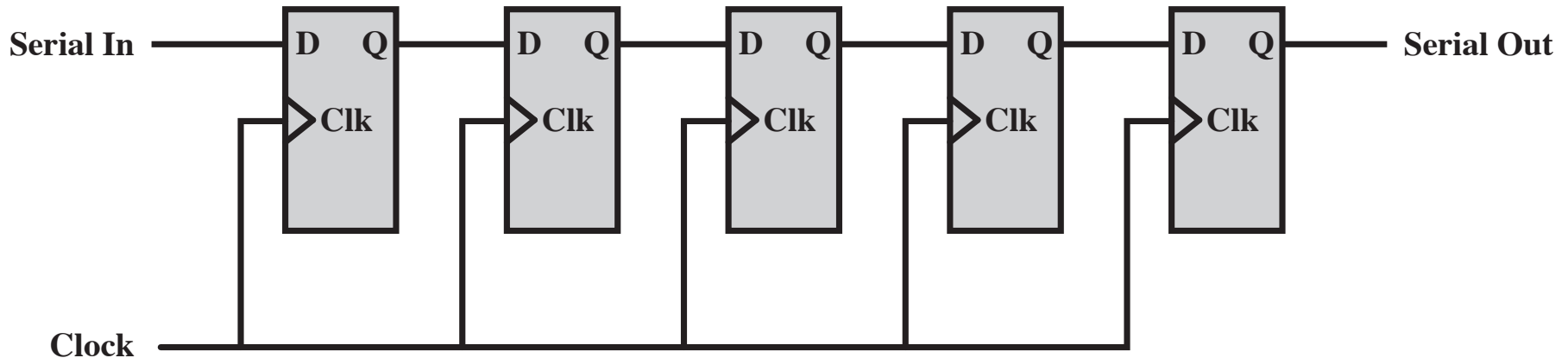
Shift register

- Per il trasferimento seriale si utilizza un dispositivo chiamato **registro a scorrimento** (**shift register**)
- A ciascun impulso (\square) del clock, i bit contenuti nel registro scorrono di una posizione verso destra:
 - R4 \rightarrow Y
 - R3 \rightarrow R4
 - R2 \rightarrow R3
 - R1 \rightarrow R2
 - X \rightarrow R1



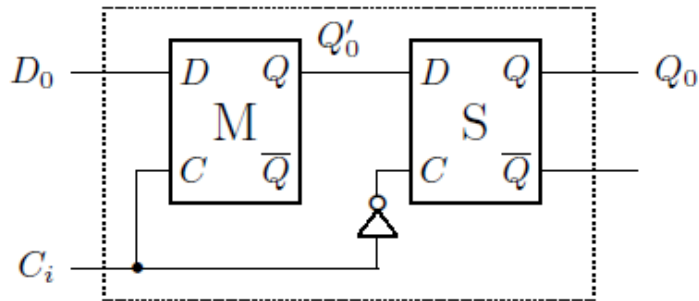
Esempio: se inizialmente è $X=0$ e lo shift register contiene 0111, dopo un impulso del clock lo shift register contiene 0011 \rightarrow **Divisione per 2**

Shift register a 5 bit



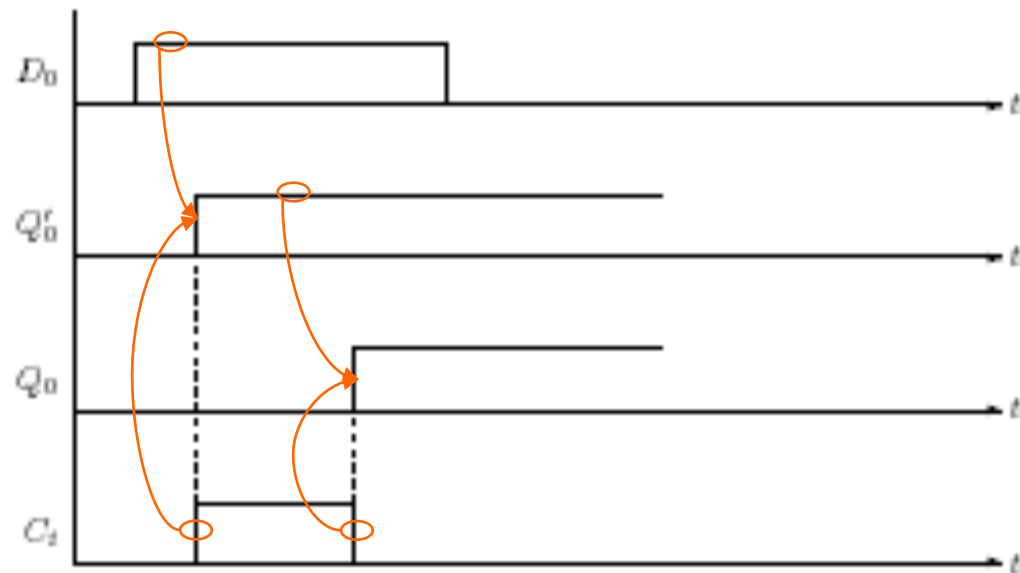
- Il clock deve essere tale da permettere il passaggio di un dato da un flip-flop al successivo
- Se il periodo di clock è troppo lungo si può usare un flip-flop di tipo D master-slave

Flip-flop di tipo D (Master-Slave)

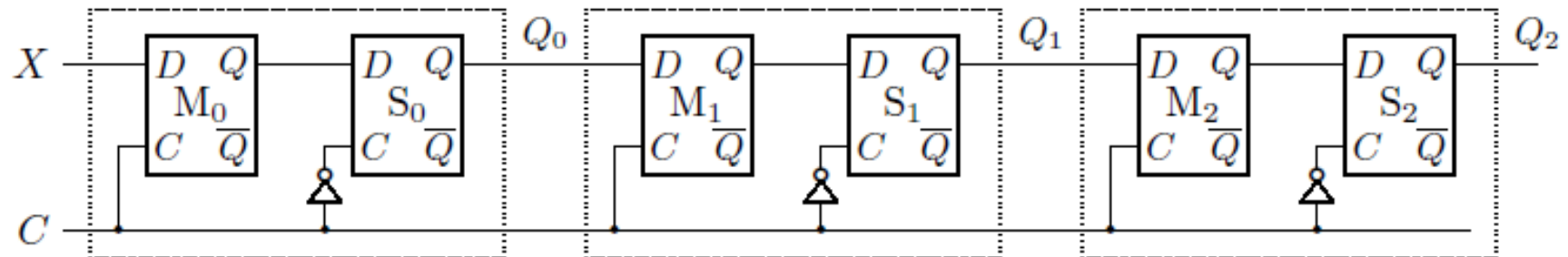


Ingresso e uscita: **disaccoppiati**

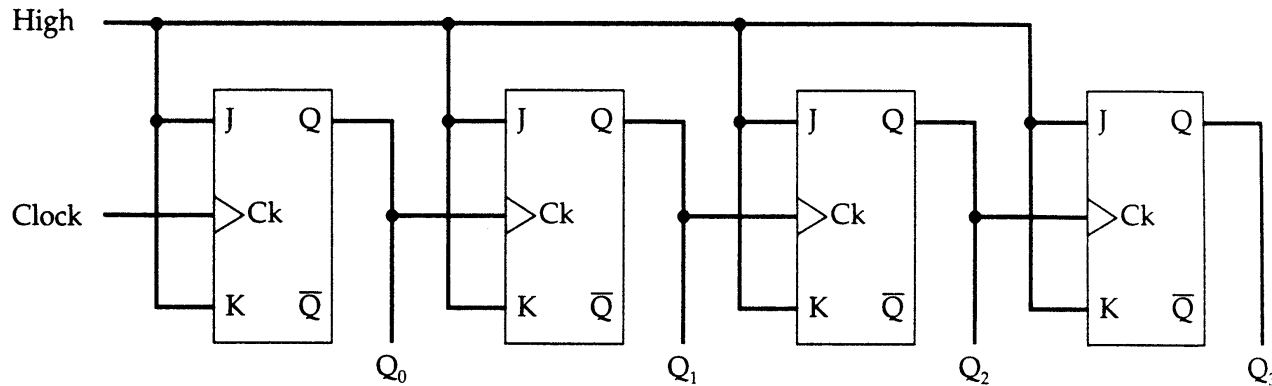
Il flip-flop slave memorizza il valore in ingresso sul fronte di discesa, con un ritardo di mezzo ciclo di clock



Shift register con flip-flop D Master-Slave

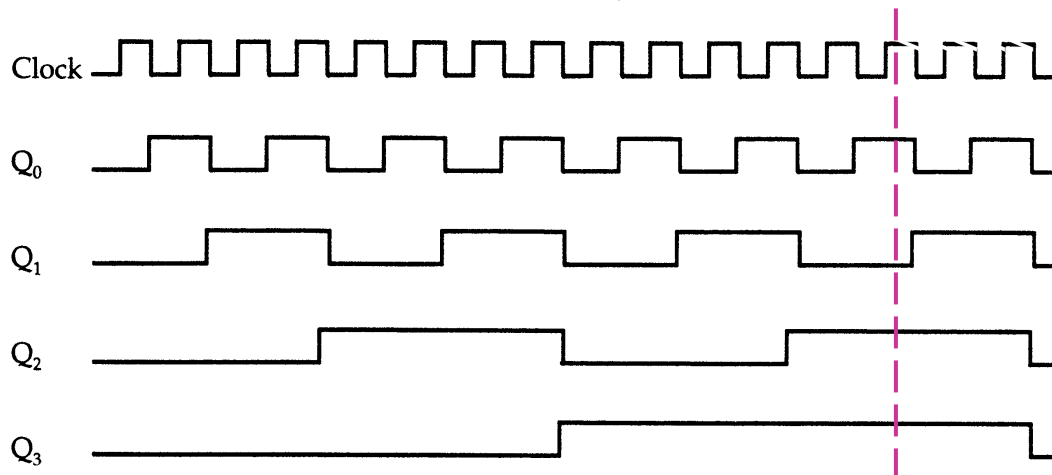


Esempio: contatore binario



(a) Sequential Circuit

Richiede edge-triggered flip-flop con clock $1 \rightarrow 0$

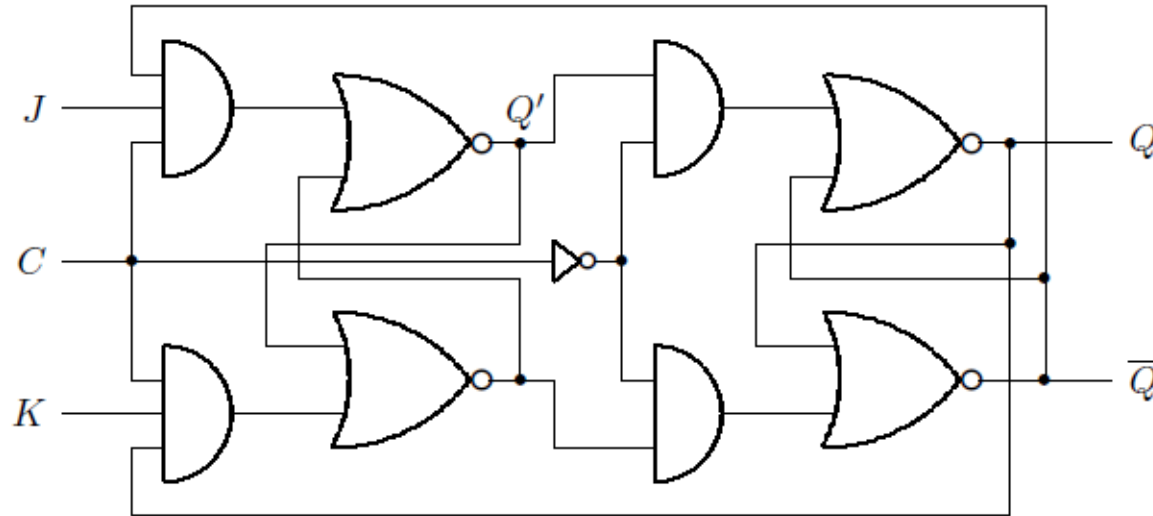


(b) Timing Diagram

t_x

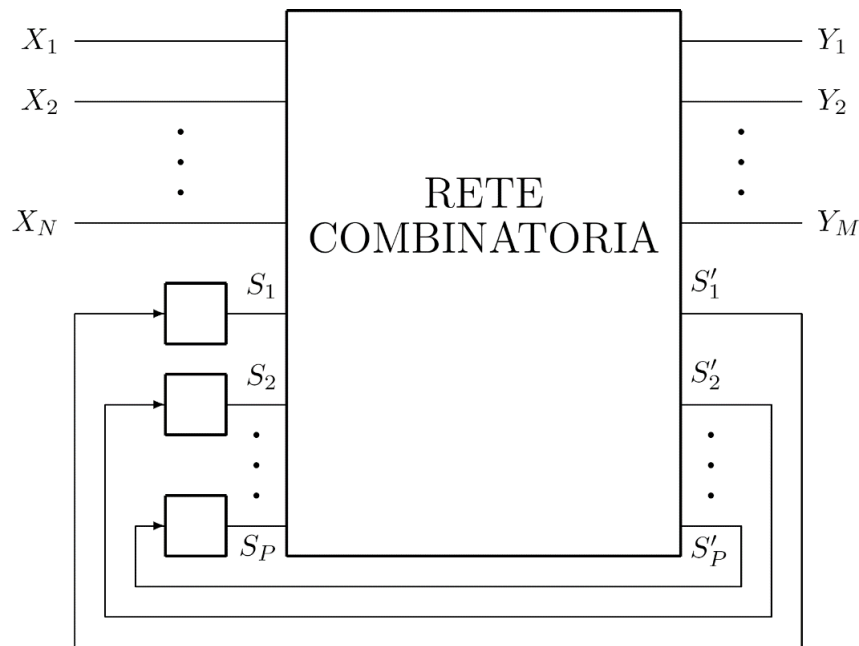
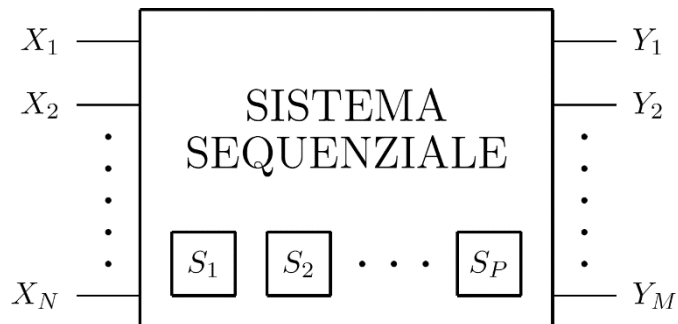
All'istante t_x si ha:
 $Q_3 Q_2 Q_1 Q_0 = 1 1 0 1_2 = 13$

Flip-flop JK sensibile ai fronti di discesa



- Configurazione di tipo Master-Slave
- Il funzionamento si distingue dal flip-flop J-K per:
 - La commutazione sui fronti di discesa anziché sul livello (**edge-triggered**)
 - Il disaccoppiamento uscita/ingresso tipico della configurazione Master-Slave

Sintesi delle reti sequenziali (1 di 2)

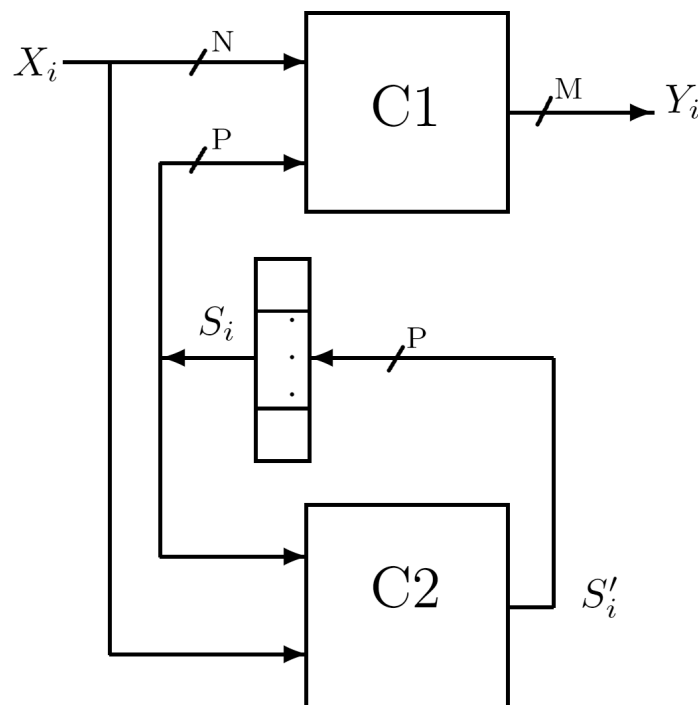


- Lo **stato del sistema** è contenuto in elementi di memoria: per Z stati servono $P = \lceil \log_2 Z \rceil$ bit di memoria.
- Ad ogni istante, una **rete combinatoria** fornisce le uscite e il nuovo valore dello stato in funzione degli ingressi.

Sintesi delle reti sequenziali (2 di 2)

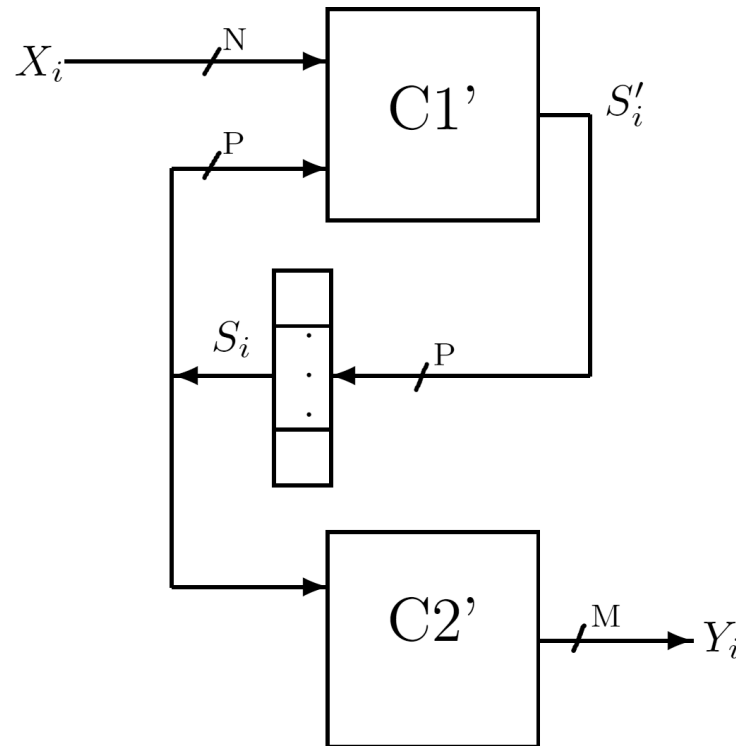
- Un'opportuna **scelta delle variabili di stato** riduce la complessità della parte combinatoria, ma non ci sono regole generali da seguire.
- Per la sintesi della rete combinatoria si applicano le tecniche già illustrate in precedenza.
- Spesso la rete viene realizzata in due parti separate, secondo uno dei due modelli seguenti:
 - modello della **macchina di Mealy** (1955);
 - modello della **macchina di Moore** (1956).
- I due modelli sono funzionalmente equivalenti.

Sintesi: macchina di Mealy



- **Rete combinatoria C1:** fornisce le **uscite** in funzione degli ingressi e dello stato.
- **Rete combinatoria C2:** fornisce il nuovo **stato** in funzione degli ingressi e del vecchio stato (= stato all'istante precedente).

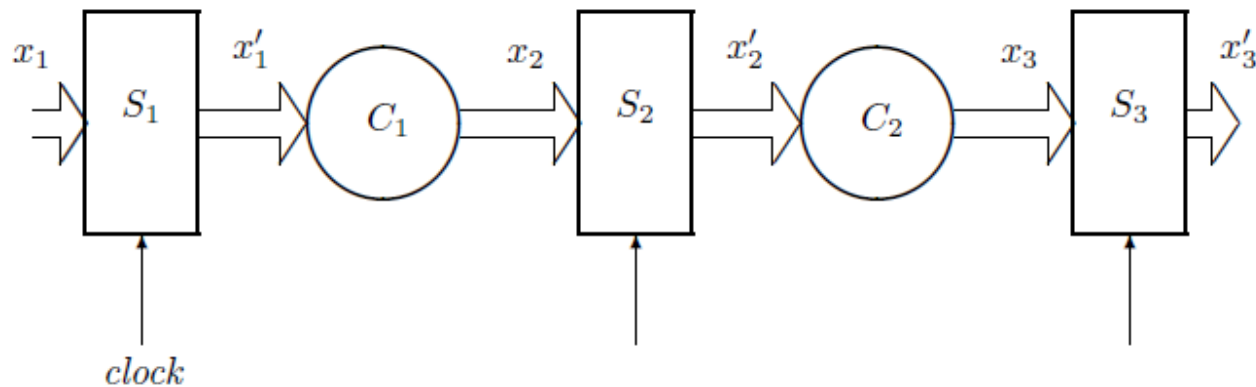
Sintesi: macchina di Moore



- **Rete combinatoria $C2'$** : fornisce le **uscite** in funzione del solo stato.
- **Rete combinatoria $C1'$** : fornisce il nuovo **stato** in funzione degli ingressi e del vecchio stato.

Sincronizzazione di più reti

Nelle reti logiche reali i **tempi di propagazione sono non nulli**: questo può essere uno svantaggio, ma consente anche di operare simultaneamente i diversi passi di una catena di elaborazioni:

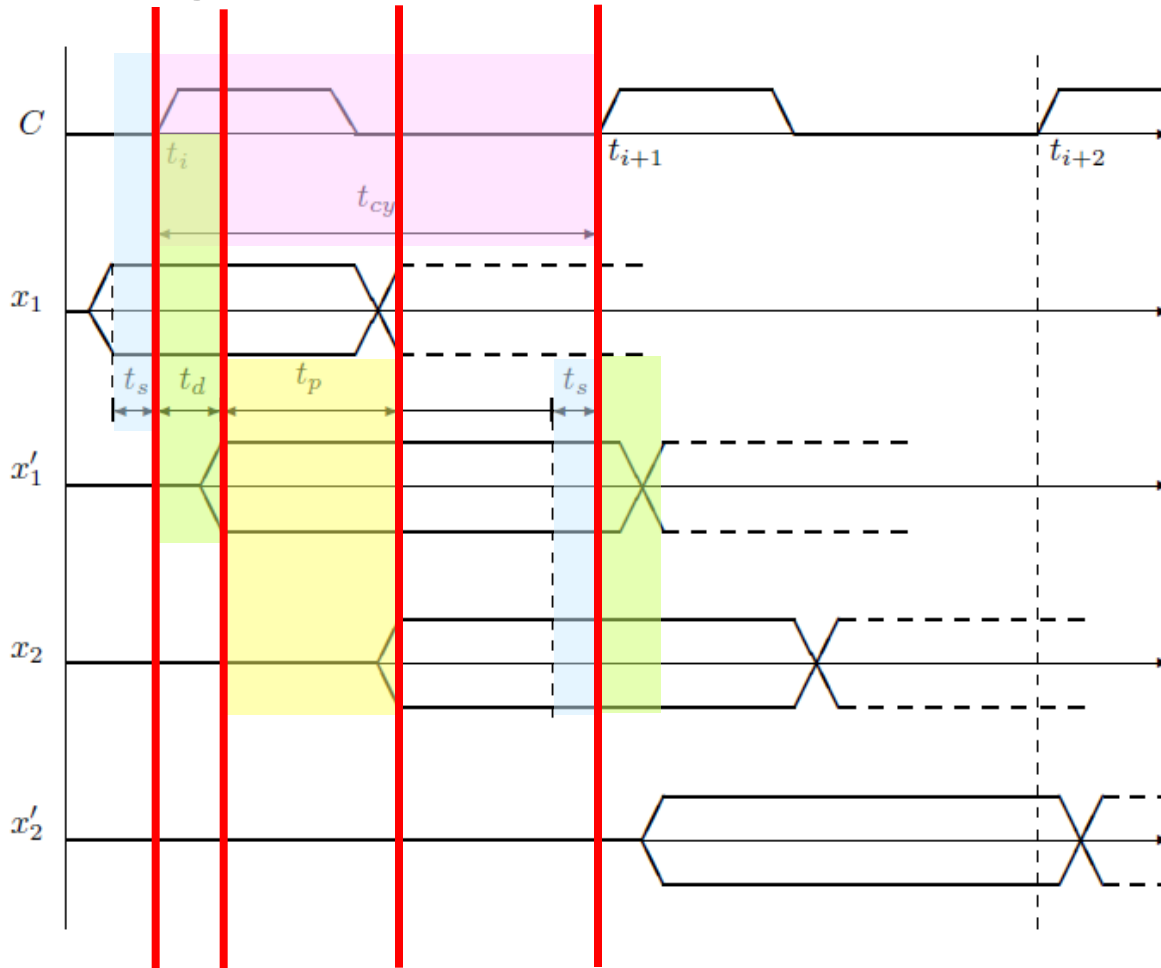


Le uscite all'istante t delle reti combinatorie C_i fungono da ingressi per la rete C_{i+1} all'istante t_{i+1} (**pipeline**).

Un **clock** sincronizza le elaborazioni, assicurando che gli elementi di memoria S_i acquisiscano valori corretti.

Limiti sul periodo del clock

Diagramma temporale del sistema precedente:

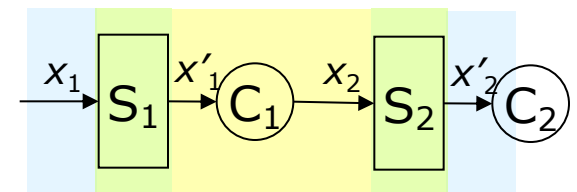


t_{cy} : periodo del clock

t_s : tempo "di set up"

t_d : tempo di memorizzaz.

t_p : tempo di propagazione



Deve essere:

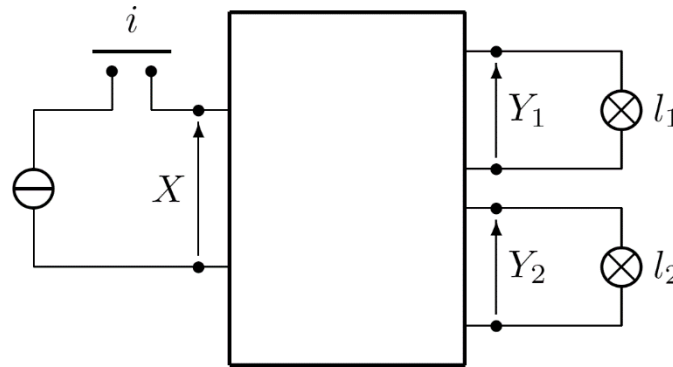
$$t_{cy} \geq t_s + t_d + t_p$$

Esempio di sintesi: interruttore

Un interruttore i , due lampadine l_1 e l_2 .

Assegnazione degli stati logici all'interruttore e alle lampadine:

- **0** = interruttore aperto, **1** = interruttore chiuso (premuto);
- **0** = lampadina spenta, **1** = lampadina accesa.

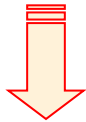


Specifica di funzionamento

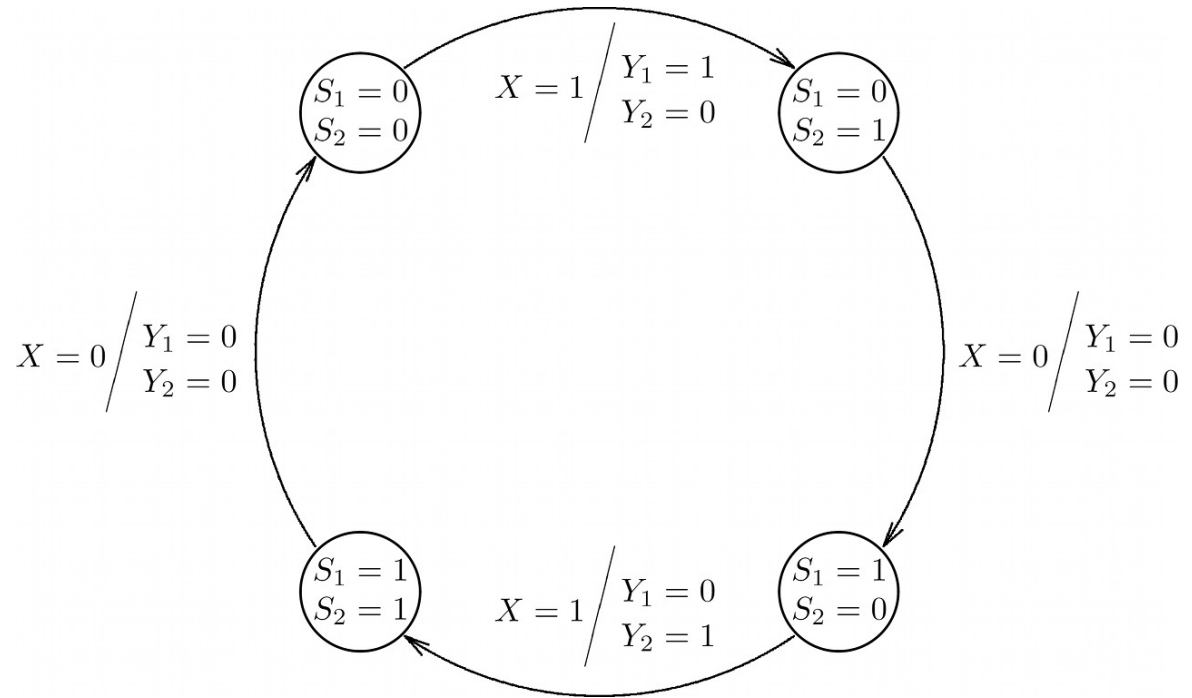
- **Interruttore aperto** ($X=0$):
le lampadine sono entrambe spente ($Y_1=Y_2=0$).
- **Interruttore chiuso** ($X=1$):
si accende a turno una delle due lampadine.

Esempio di sintesi: diagramma di stato

4 stati



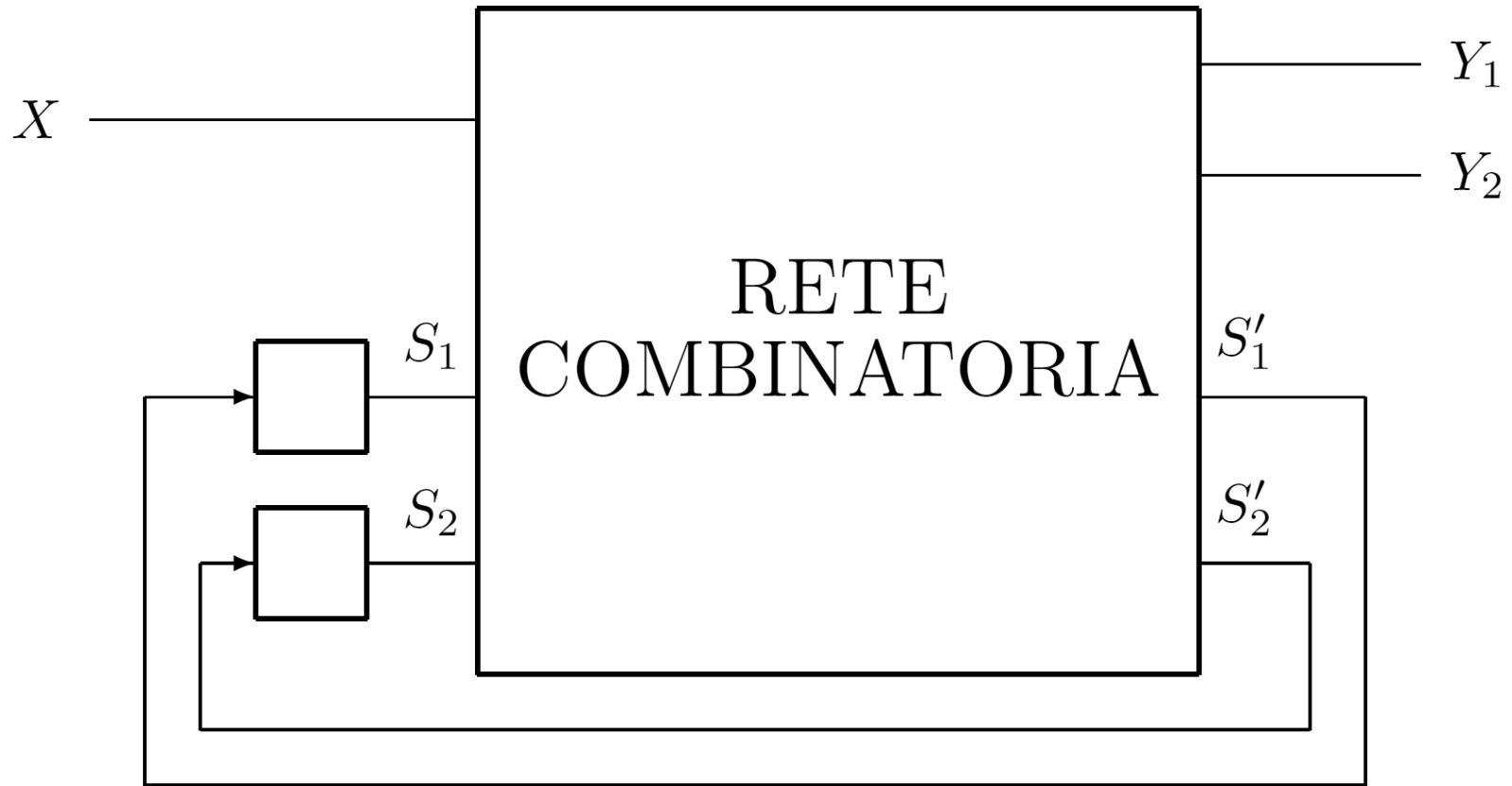
2 bit di stato



Specifica di funzionamento

- **Interruttore aperto** ($X=0$):
le lampadine sono entrambe spente ($Y_1=Y_2=0$).
- **Interruttore chiuso** ($X=1$):
si accende a turno una delle due lampadine.

Esempio di sintesi: modello



Sintesi con macchina di Mealy

Esempio di sintesi: parte combinatoria

S_1	S_2	X	Y_1	Y_2	S_1'	S_2'
0	0	0	0	0	0	0
0	0	1	1	0	0	1
0	1	0	0	0	1	0
0	1	1	1	0	0	1
1	0	0	0	0	1	0
1	0	1	0	1	1	1
1	1	0	0	0	0	0
1	1	1	0	1	1	1

$$\begin{aligned} Y_1 &= \overline{S_1} \cdot \overline{S_2} \cdot X + \overline{S_1} \cdot S_2 \cdot X \\ &= \overline{S_1} \cdot X \end{aligned}$$

$$\begin{aligned} Y_2 &= S_1 \cdot \overline{S_2} \cdot X + S_1 \cdot S_2 \cdot X \\ &= S_1 \cdot X \end{aligned}$$

$$\begin{aligned} S_1' &= \overline{S_1} \cdot S_2 \cdot \overline{X} + S_1 \cdot \overline{S_2} \cdot \overline{X} + S_1 \cdot \overline{S_2} \cdot X + S_1 \cdot S_2 \cdot X \\ &= S_1 \cdot X + (S_1 \oplus S_2) \cdot \overline{X} \end{aligned}$$

$$\begin{aligned} S_2' &= \overline{S_1} \cdot \overline{S_2} \cdot X + \overline{S_1} \cdot S_2 \cdot X + S_1 \cdot \overline{S_2} \cdot X + S_1 \cdot S_2 \cdot X \\ &= X \end{aligned}$$

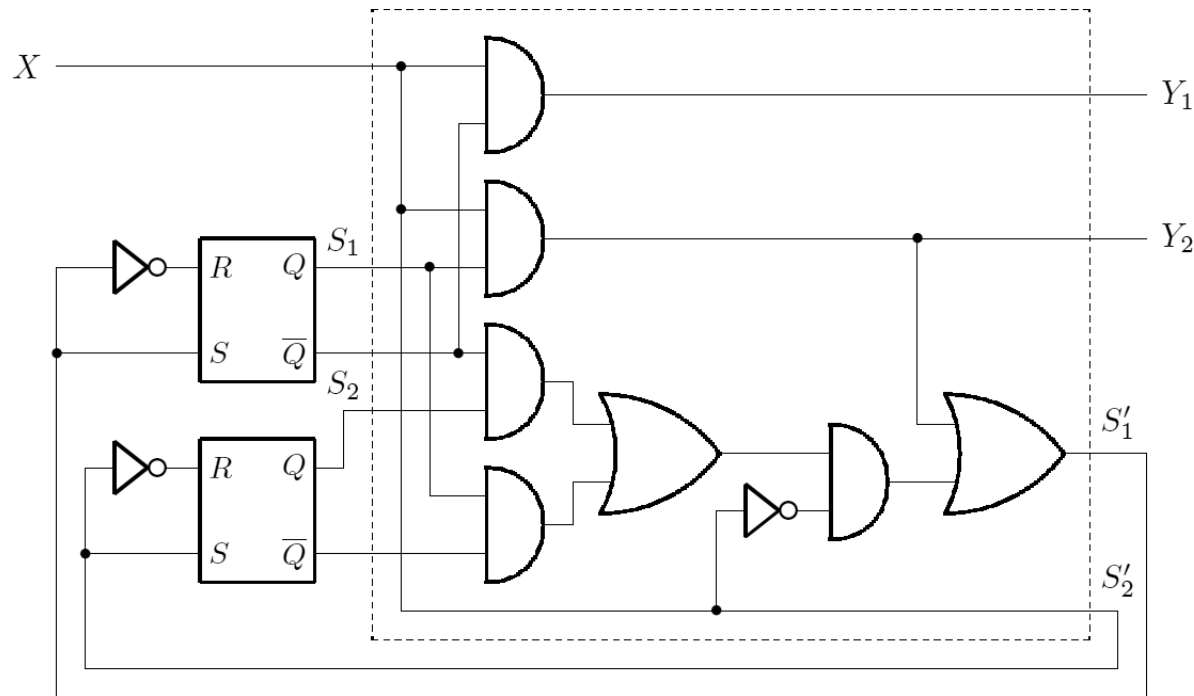
Esempio di sintesi: versione 1

$$Y_1 = \overline{S_1} \cdot X$$

$$Y_2 = S_1 \cdot X$$

$$S'_1 = Y_2 + (S_1 \oplus \overline{S_2}) \cdot X$$

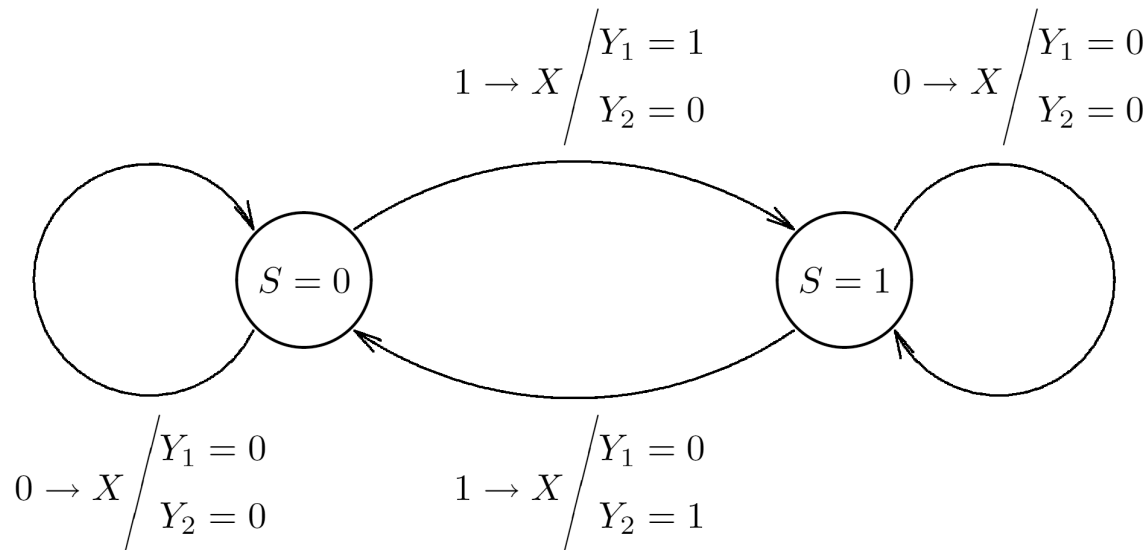
$$S'_2 = X$$



Latch R-S come elementi di memoria: rete sequenziale **asincrona**

Esempio di sintesi: "revisione"

S2 = X: si può fare a meno di S2? Sì!



- **S=0**: la prossima lampadina ad accendersi sarà l_1
- **S=1**: la prossima lampadina ad accendersi sarà l_2

Lo stesso valore ($X=1$) che porta da $S=0$ a $S=1$, porta anche da $S=1$ a $S=0$: le transizioni vanno sincronizzate da un clock.

- E' necessaria una rete sequenziale **sincrona**.

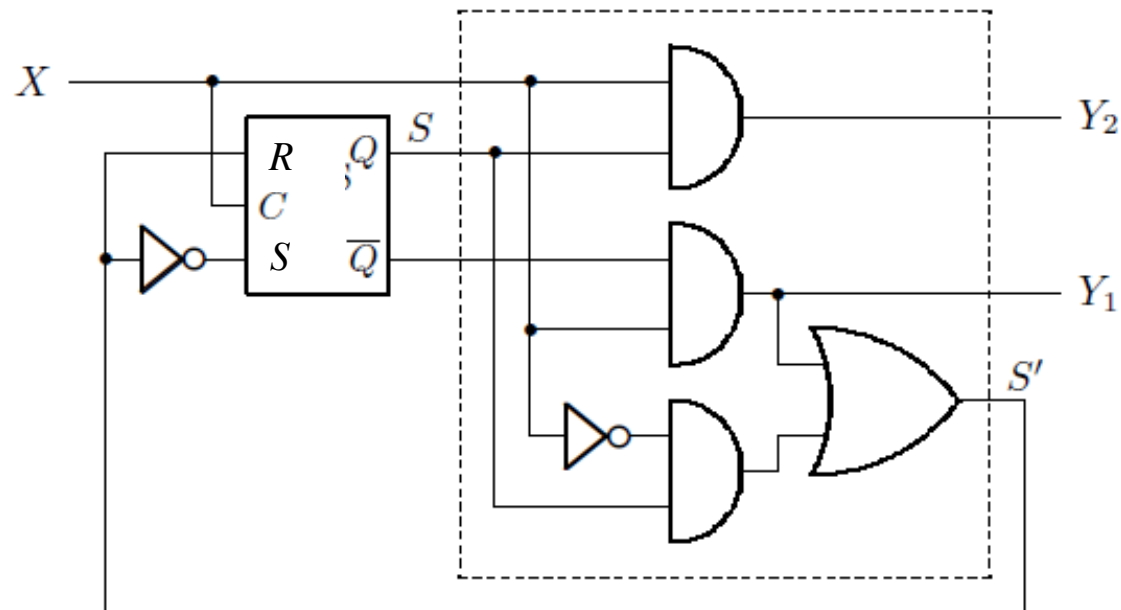
Esempio di sintesi: versione 2

S	X	Y_1	Y_2	S'
0	0	0	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	0

$$Y_1 = \bar{S} \cdot X$$

$$Y_2 = S \cdot X$$

$$S' = S \oplus X$$



Serve un flip-flop, ad es. di tipo R-S.

Le funzioni di clock, che sincronizza le transizioni, qui sono svolte da X .