Flip-flop e loro applicazioni

Reti sequenziali elementari

(6)

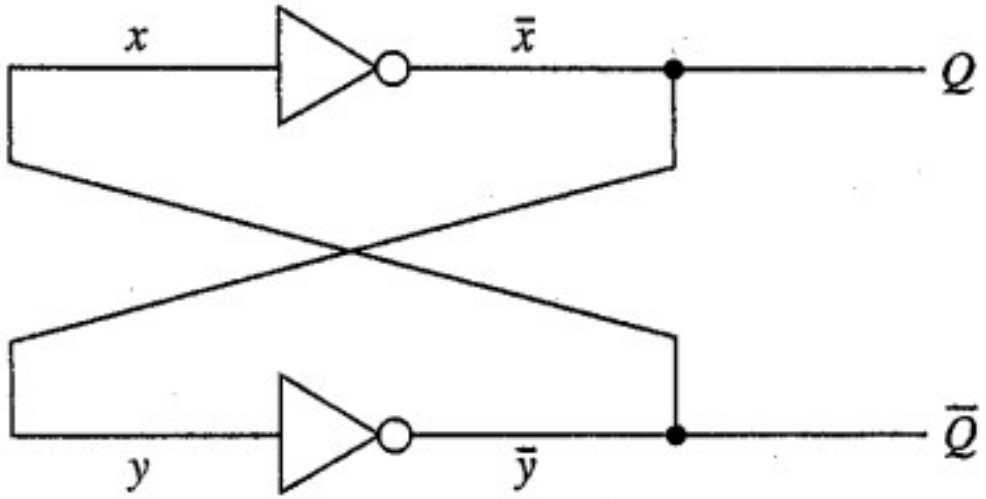
L'elemento bistabile
Latch o flip-flop trasparenti
Temporizzazione dei flip-flop trasparenti
Architettura master-slave
Flip-flop non trasparenti o edge-triggered
Registri, riconoscitori di sequenza
Contatori

Reti sequenziali

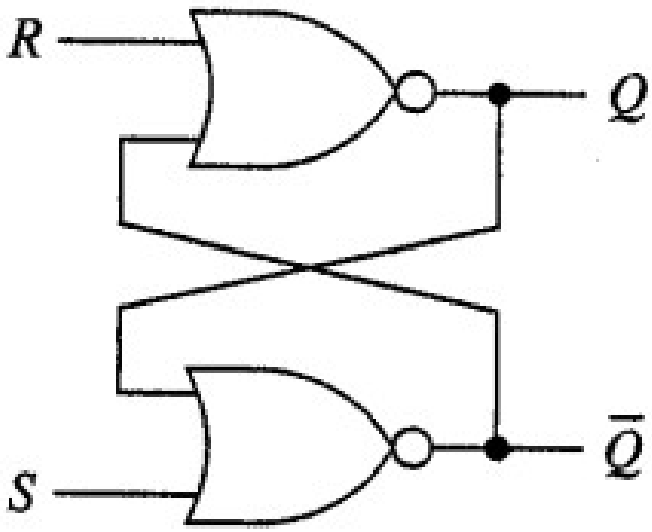
- L'uscita non dipende soltanto dagli ingressi nell'istante corrispondente, ma anche dalla sequenza dei valori precedenti
  - Alcuni nodi della rete possono assumenre valore diversi a parità di ingresso
- Due tipologie
  - Asincrone
    - Un cambiamento in un qualsiasi ingresso determina la valutazione di un nuovo stato della rete e delle uscite
  - Sincrone
    - Il calcolo di un nuovo stato viene attivato soltanto dalla commutazione di una particolare linea di ingresso (clock)

L'elemento bistabile

- La chiave per realizzare reti sequenziali è un circuito con più stati stabili
  - A parità di ingressi
  - Avendo un circuito con 2 stati stabili, si possono ottenere 2^n stati usando *n* repliche del circuito

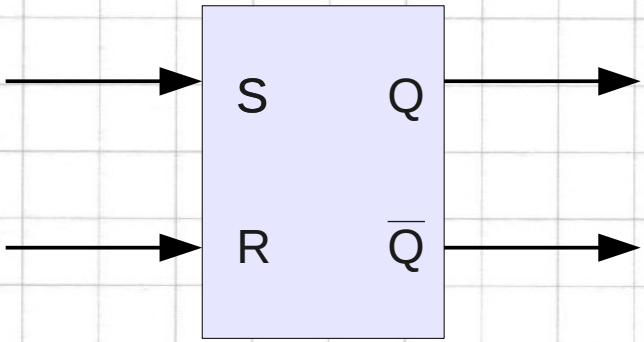


Il latch SR



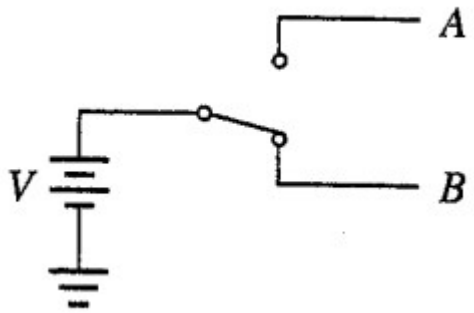
Inputs		Outputs	
S	R	Q^+	$\bar{Q}^+$
0	0	Q	$\bar{Q}$
0	1	0	1
1	0	1	0
1	1	0*	0*

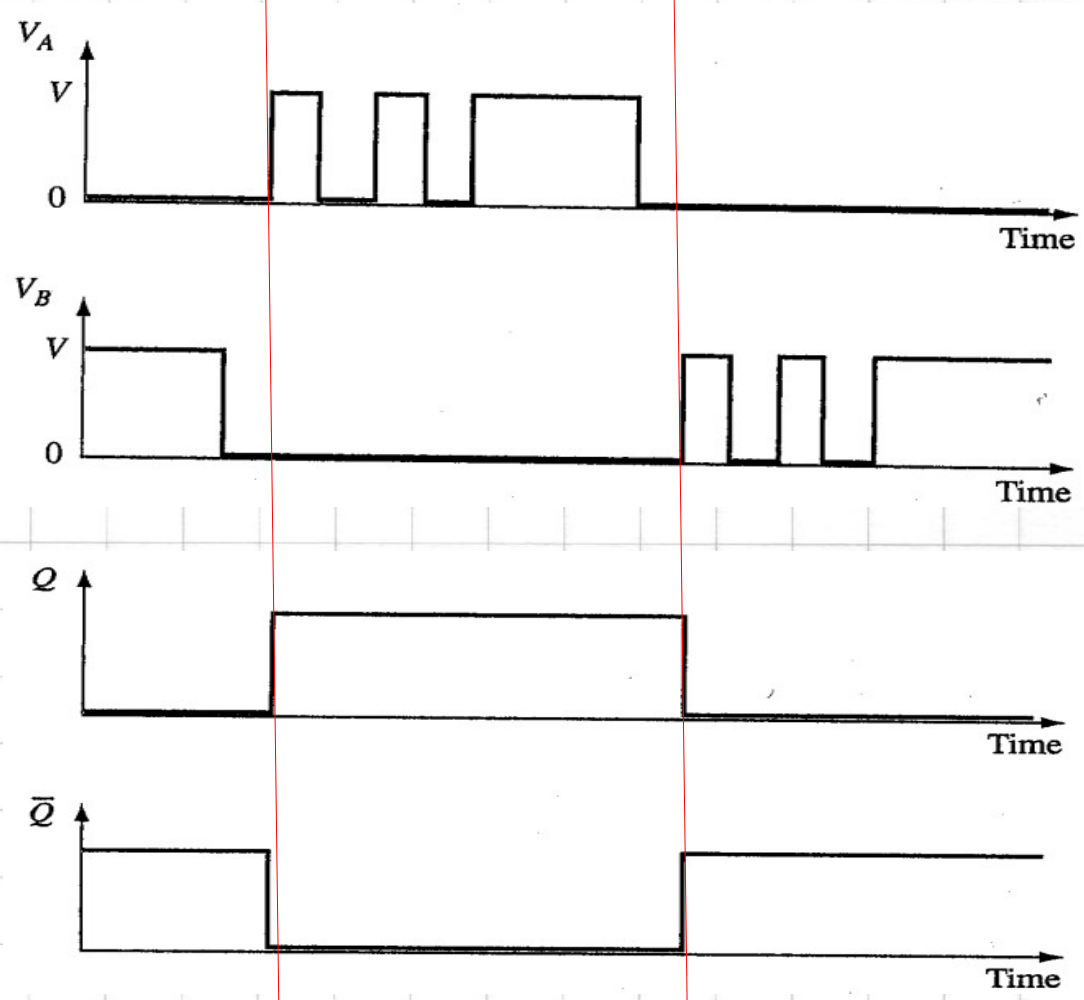
*Unpredictable behavior will result if inputs return to 0 simultaneously



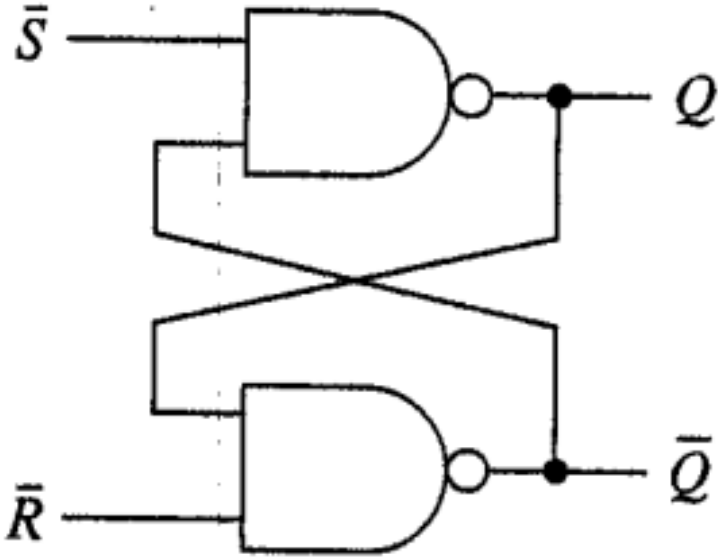
Un esempio di applicazione

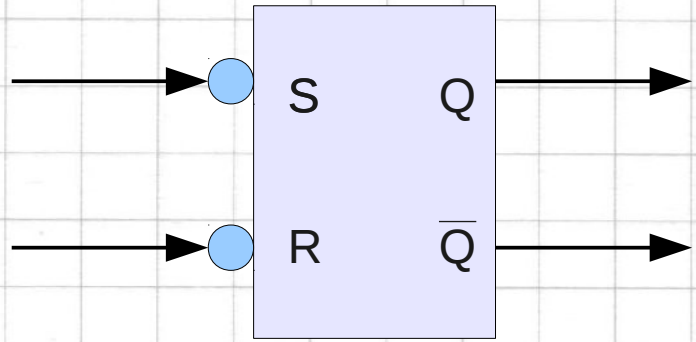
- La funzione antirimbalzo



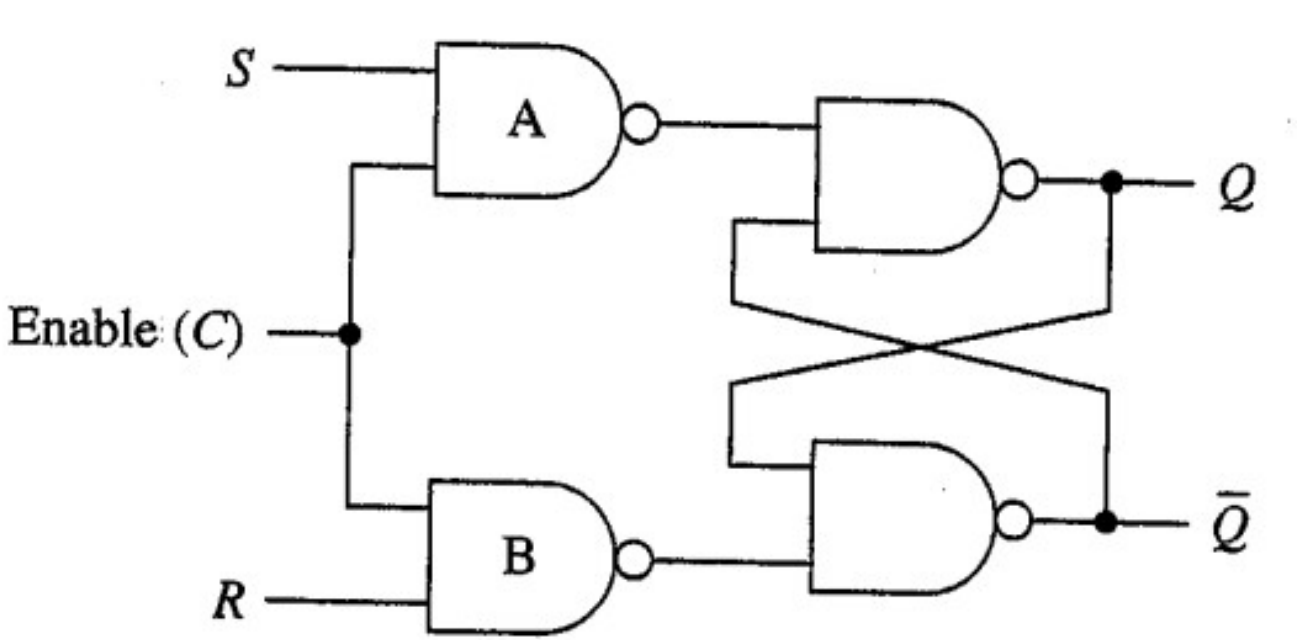


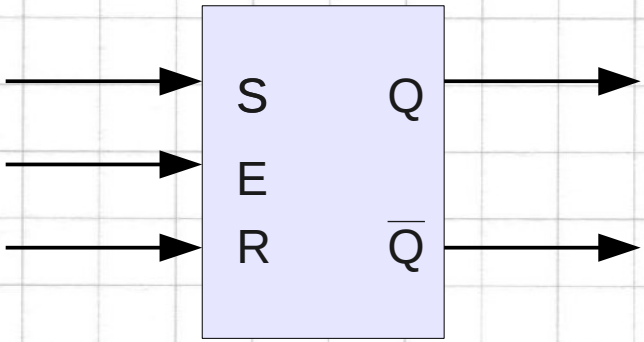
Il latch $\overline{S}$ $\overline{R}$





Il latch SR con abilitazione

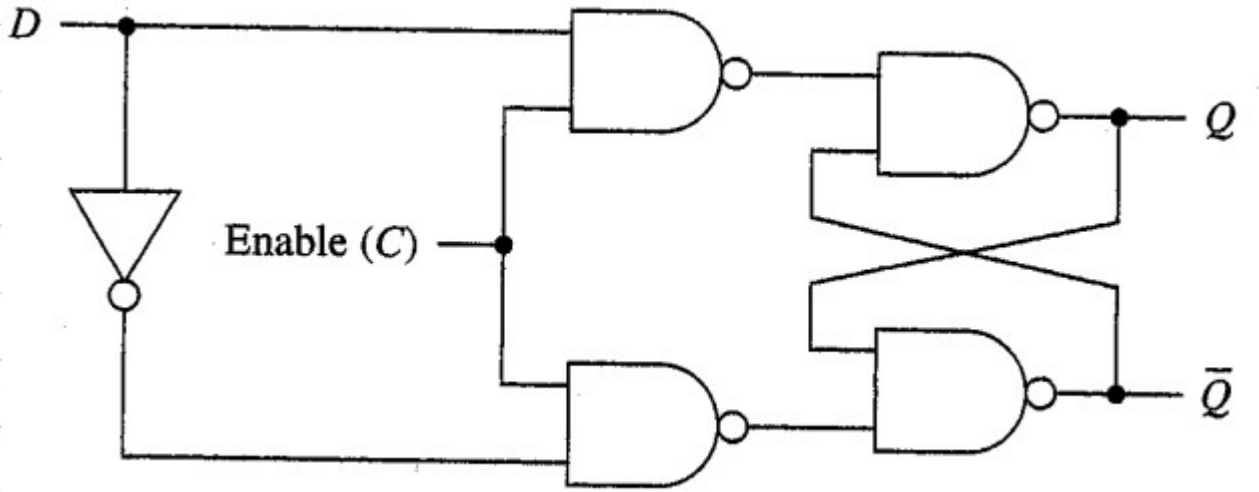


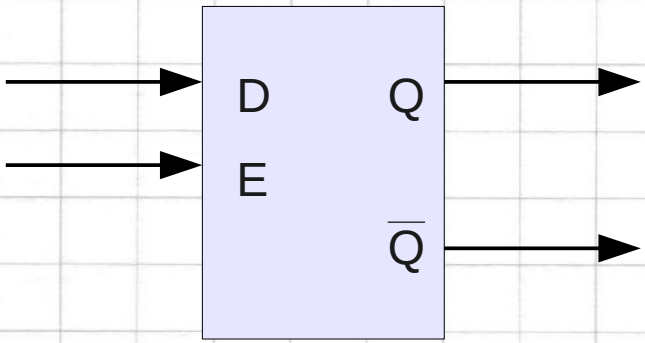


Inputs			Outputs	
S	R	E	Q^+	$\bar{Q}^+$
0	0	1	Q	$\bar{Q}$
0	1	1	0	1
1	0	1	1	0
1	1	1	1*	1*
X	X	0	Q	$\bar{Q}$

*Unpredictable behavior will result if S and R return to 0 simultaneously or C returns to 0 while S and R are 1

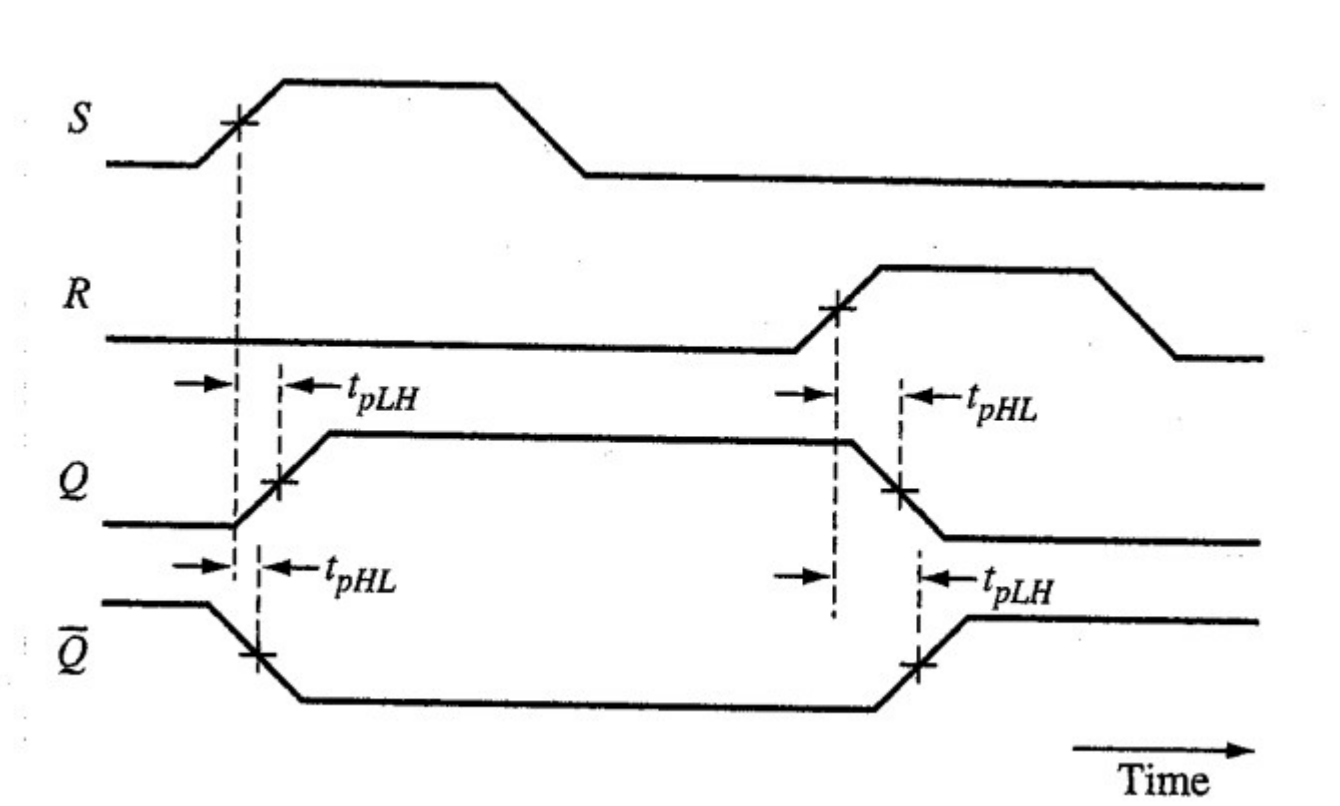
Il latch D



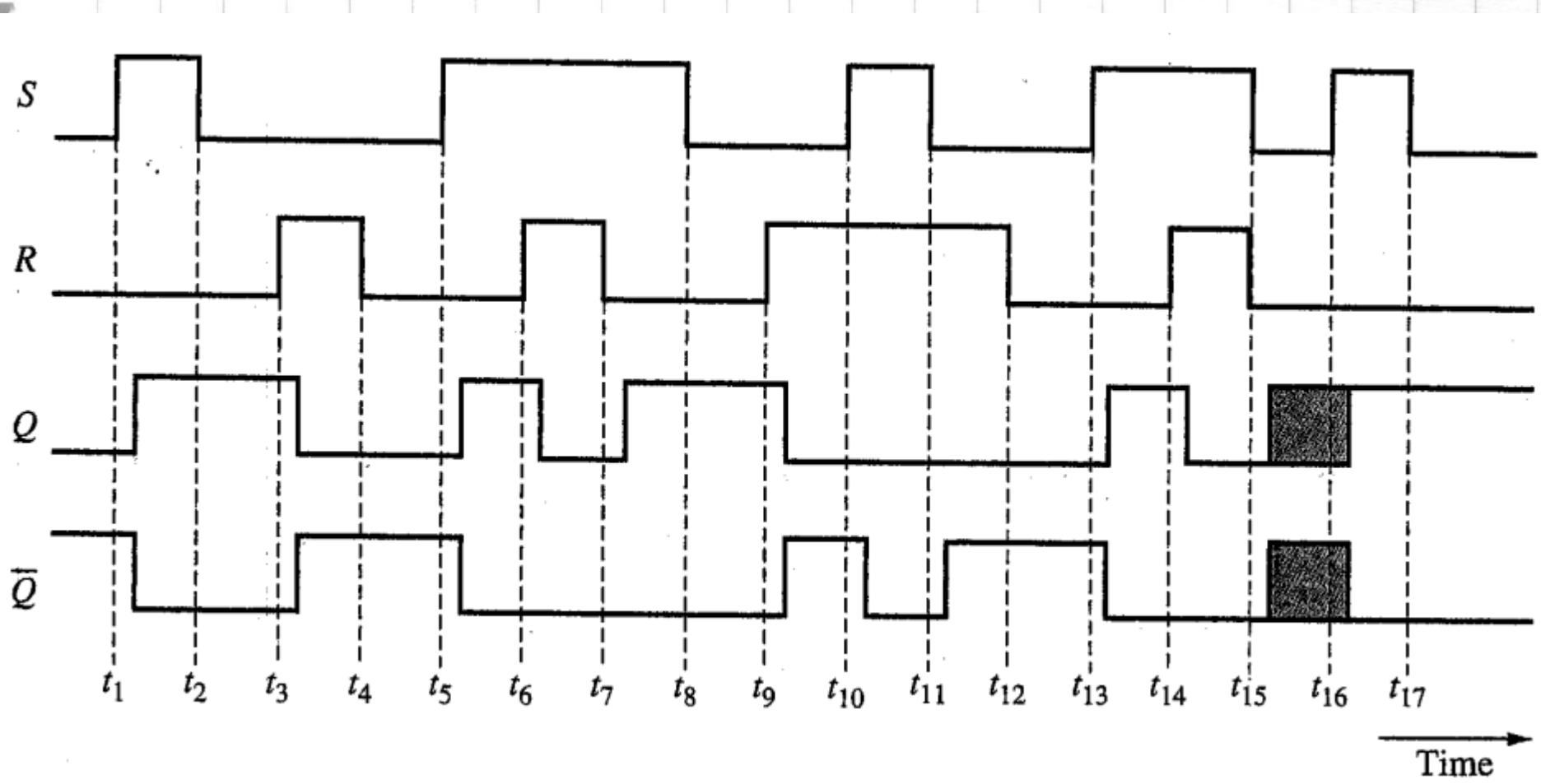


Inputs		Outputs	
D	E	Q^+	$\bar{Q}^+$
0	1	0	1
1	1	1	0
X	0	Q	$\bar{Q}$

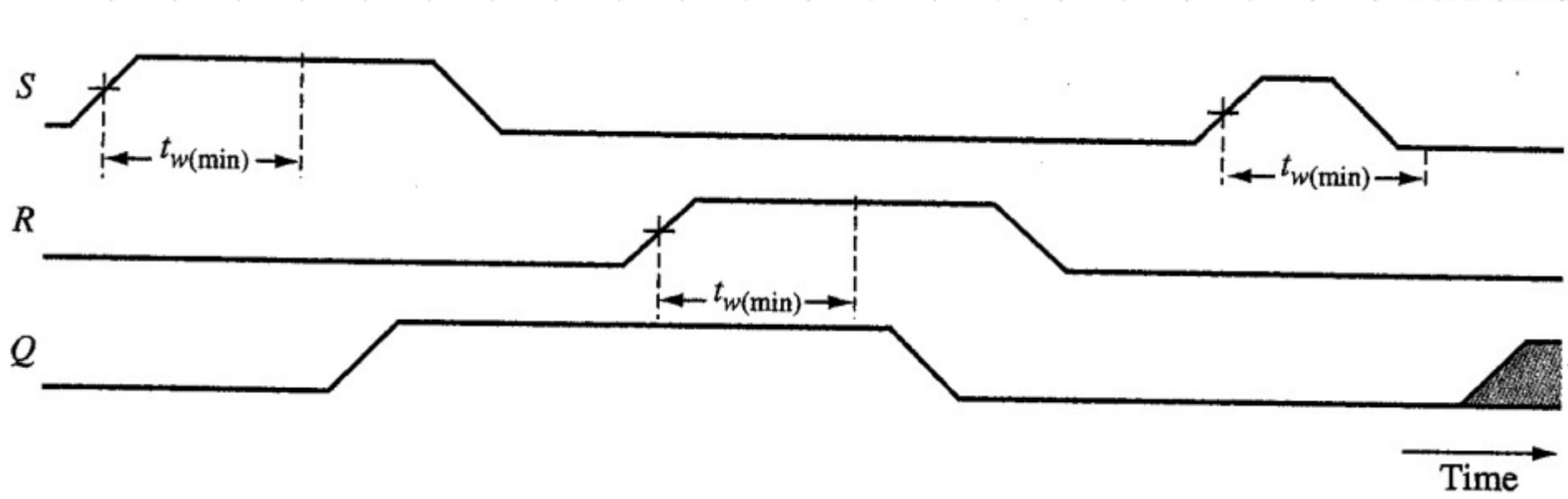
Temporizzazione dei latch (1)



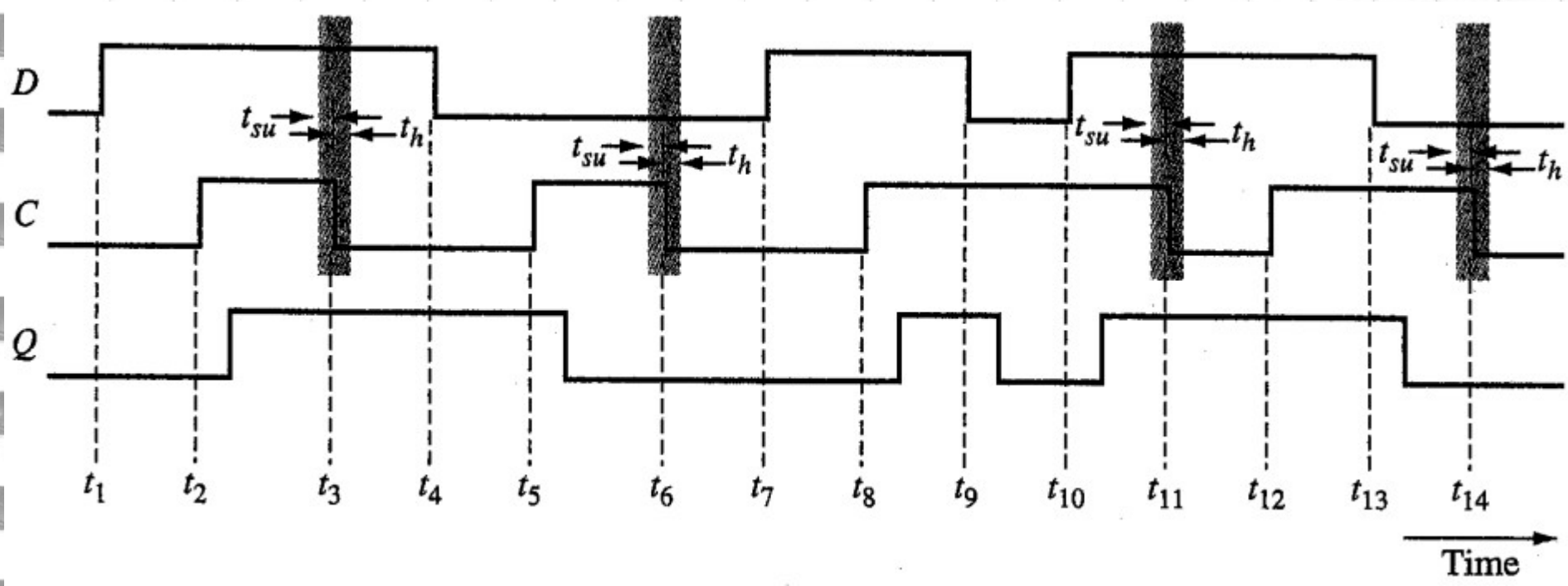
Temporizzazione dei latch (2)



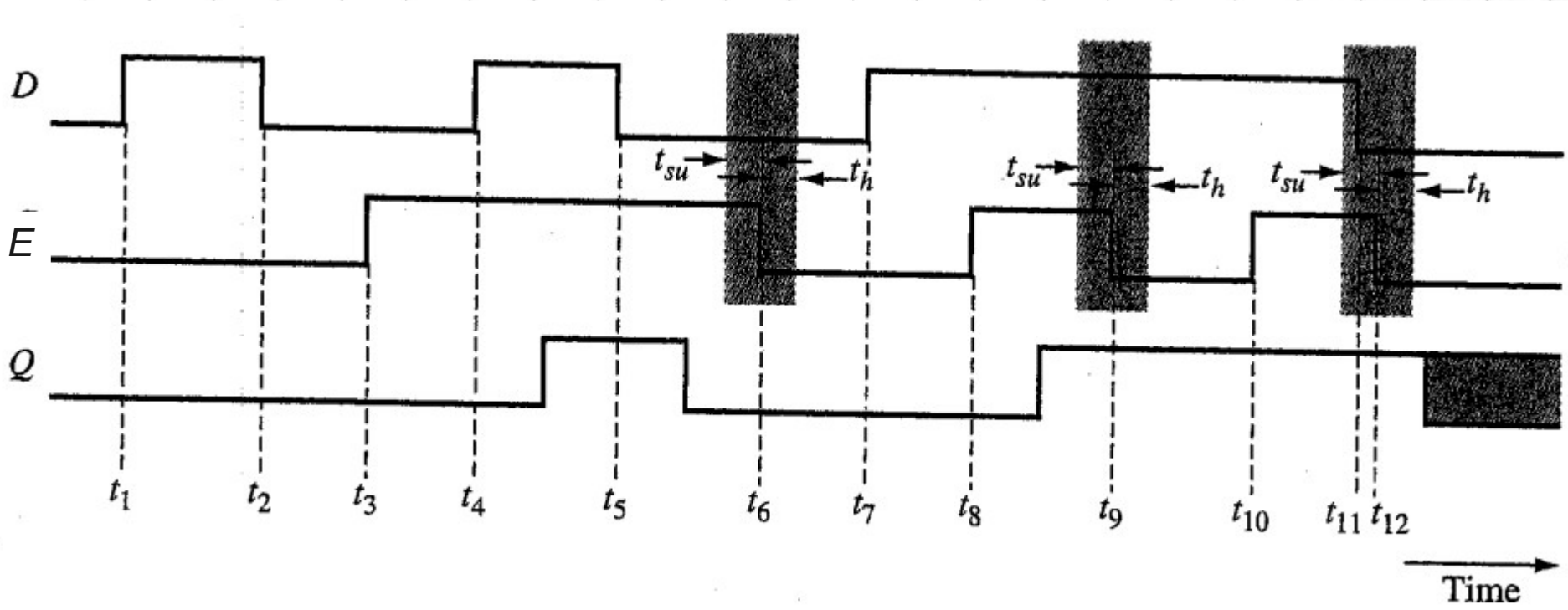
Durata minima dell'impulso



Tempo di impostazione (setup) e mantenimento (hold)



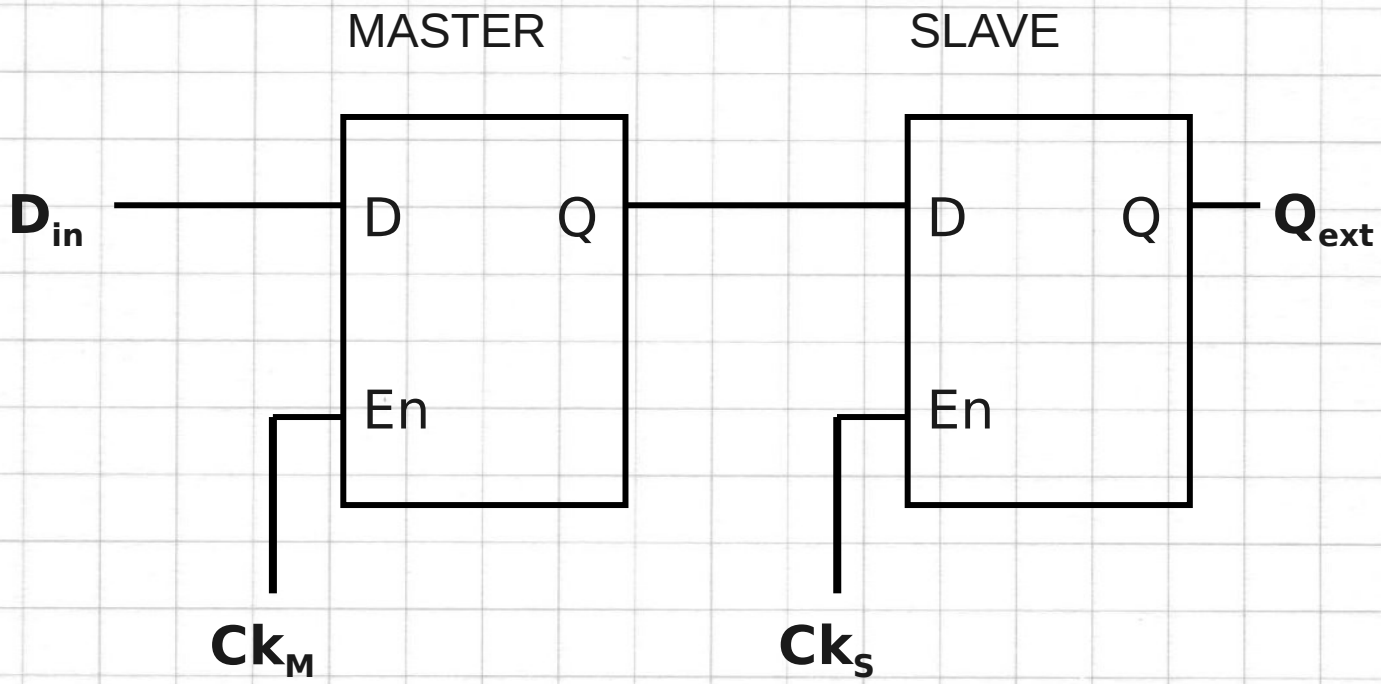
Metastabilità



Reti trasparenti e non

- I latch sono reti sequenziali sincrone che si definiscono **trasparenti** in quanto l'effetto degli ingressi generalmente si riflette sull'uscita immediatamente
  - Non è consigliabile riportare l'uscita su uno degli ingressi
    - Si rischia di innescare una situazione oscillatoria, se la commutazione modifica l'uscita
- Per questo servono reti **non trasparenti** in cui l'aggiornamento delle uscite avviene in un momento successivo alla lettura degli ingressi
  - Architettura **master-slave**
  - Reti **edge-triggered**

Architettura master-slave



Clock non sovrapposto

- Il clock master e il clock slave non devono mai essere attivi (alti, 1) contemporaneamente
- Non possono essere ottenuti con un inverter

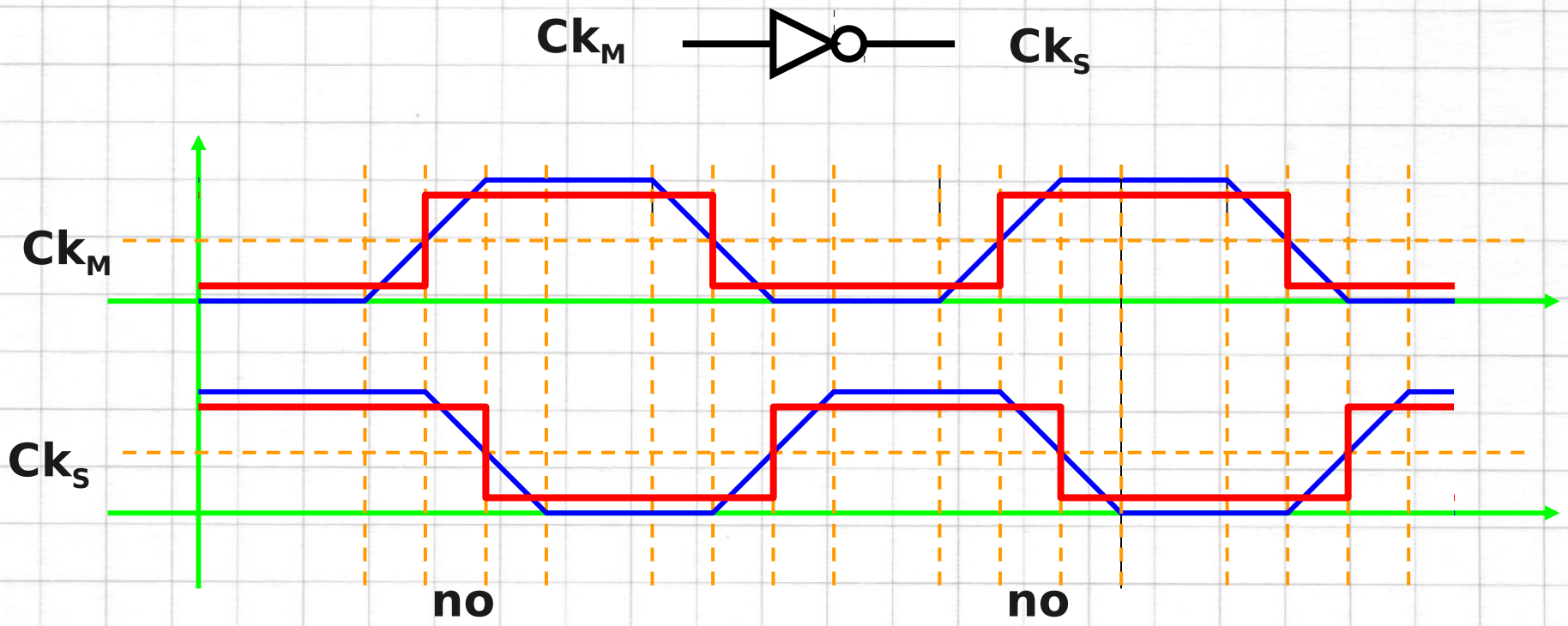
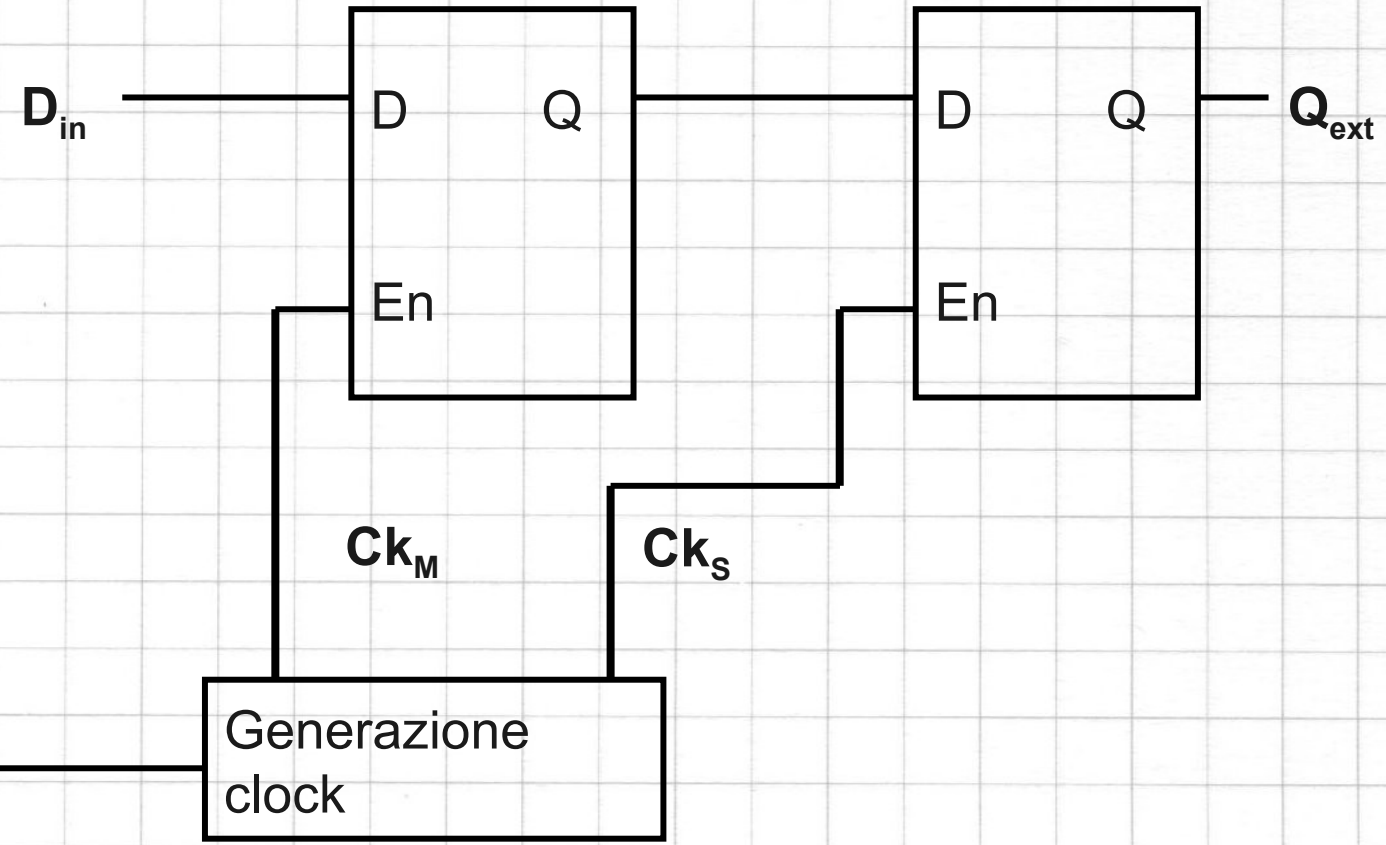


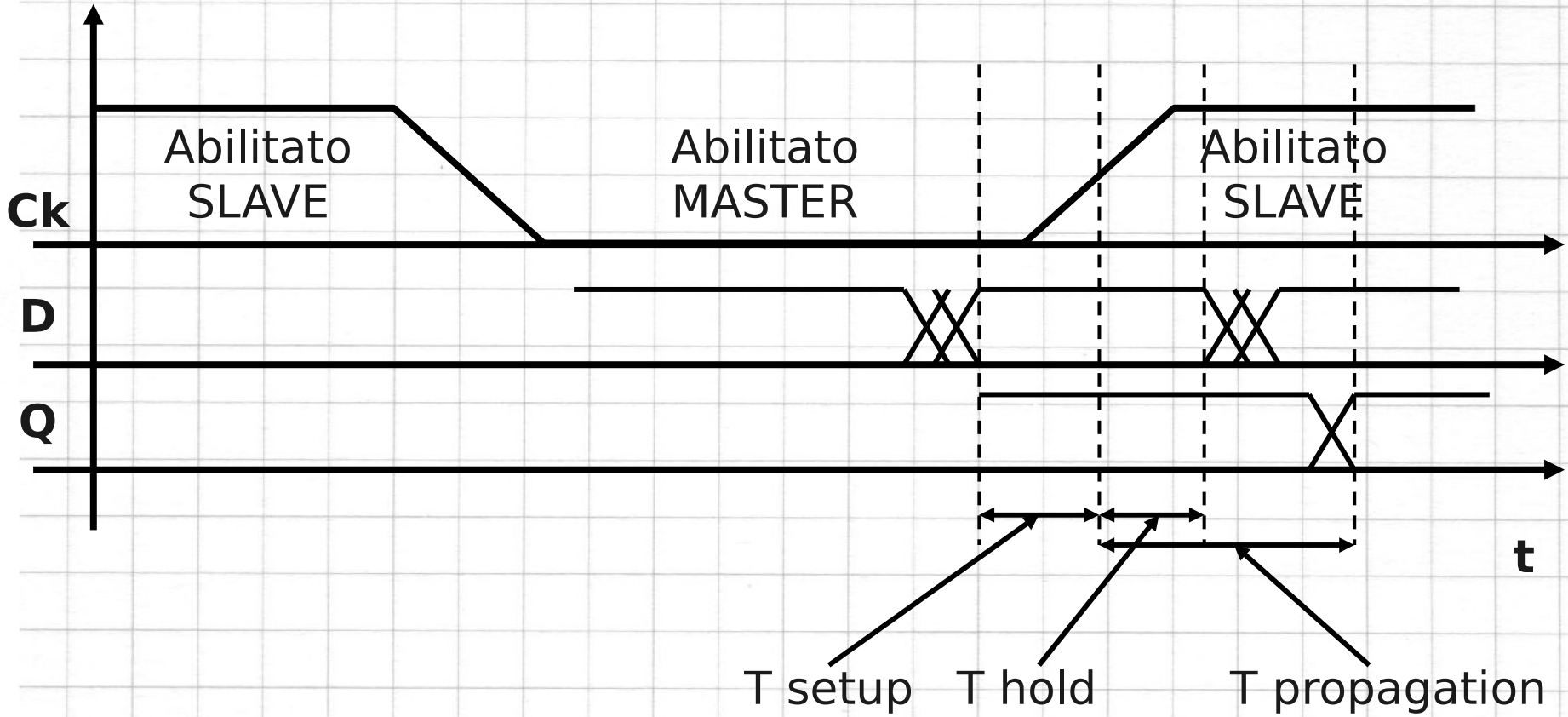
Tabella di verità

Ck	D	Q
0	X	Q
1	X	Q
↓	X	Q
↑	0	0
↑	1	1



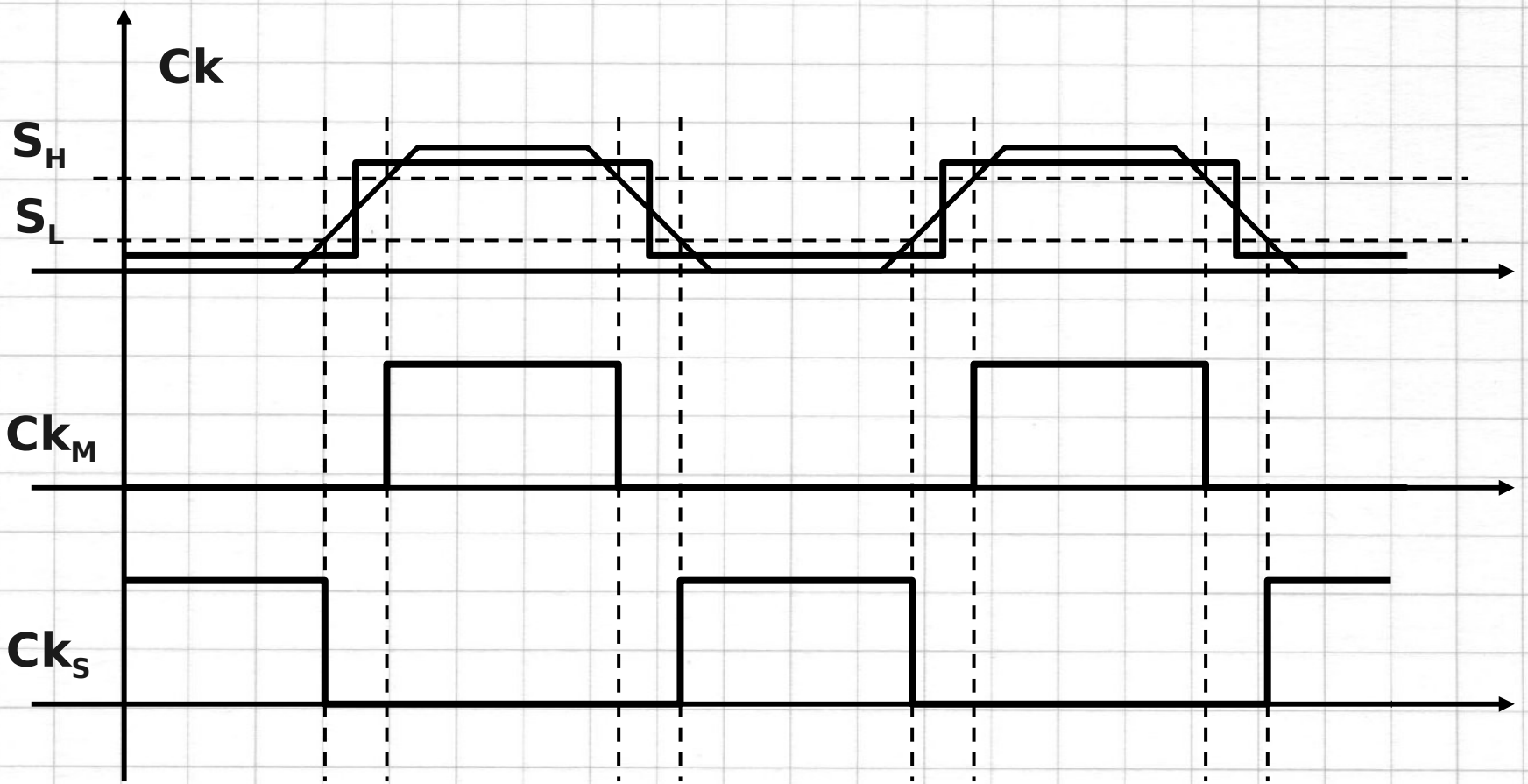
Tempi di rispetto

Per evitare errori in fase di memorizzazione è necessario che il dato sia stabile un po' prima e un po' dopo la commutazione del clock

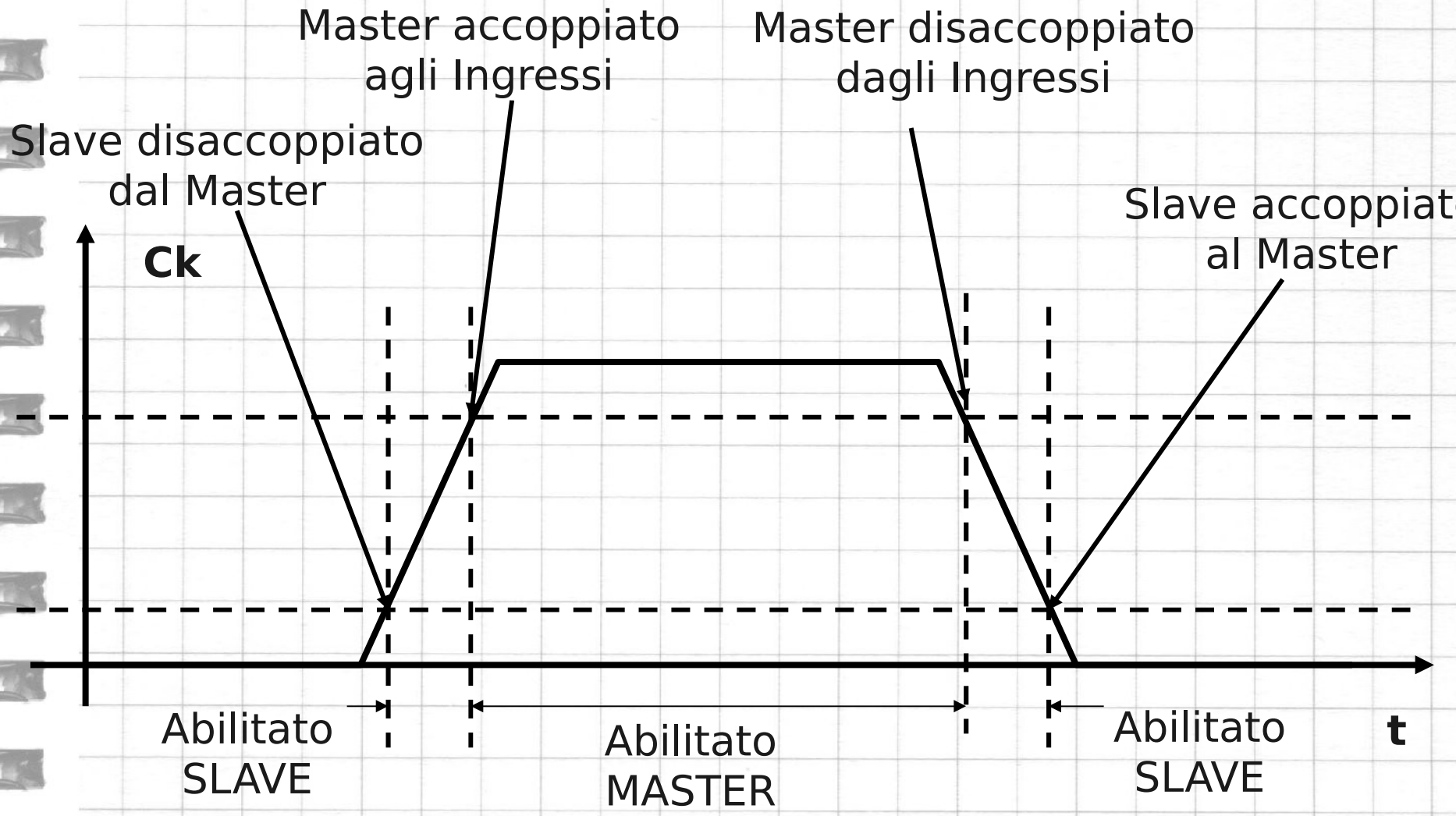


Clock a due fasi non sovrapposte

- Tecnica di generazione a soglia

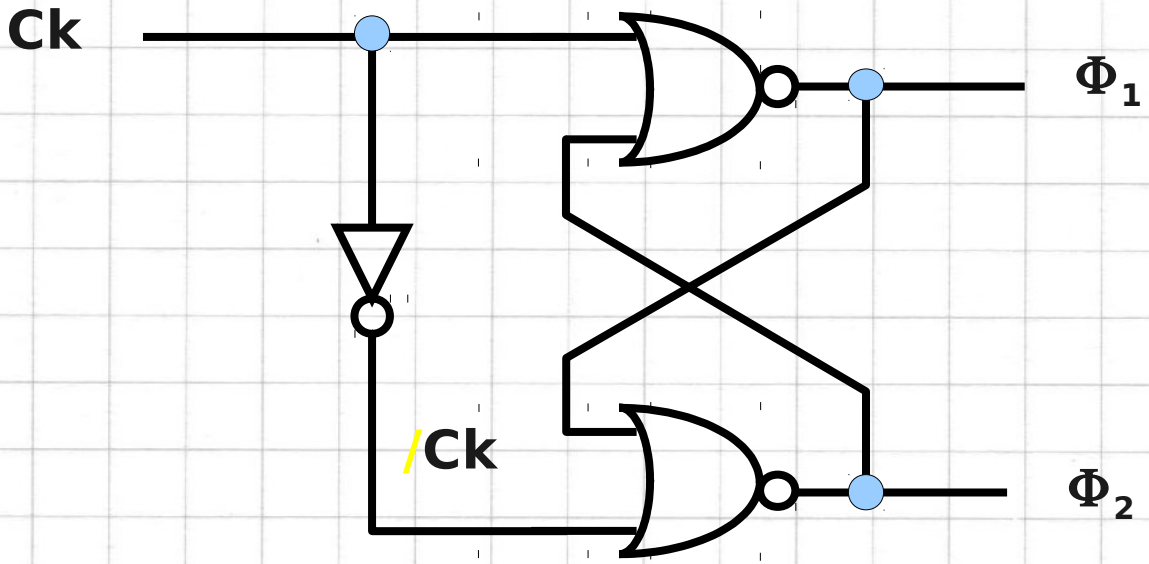


Sequenza di funzionamento

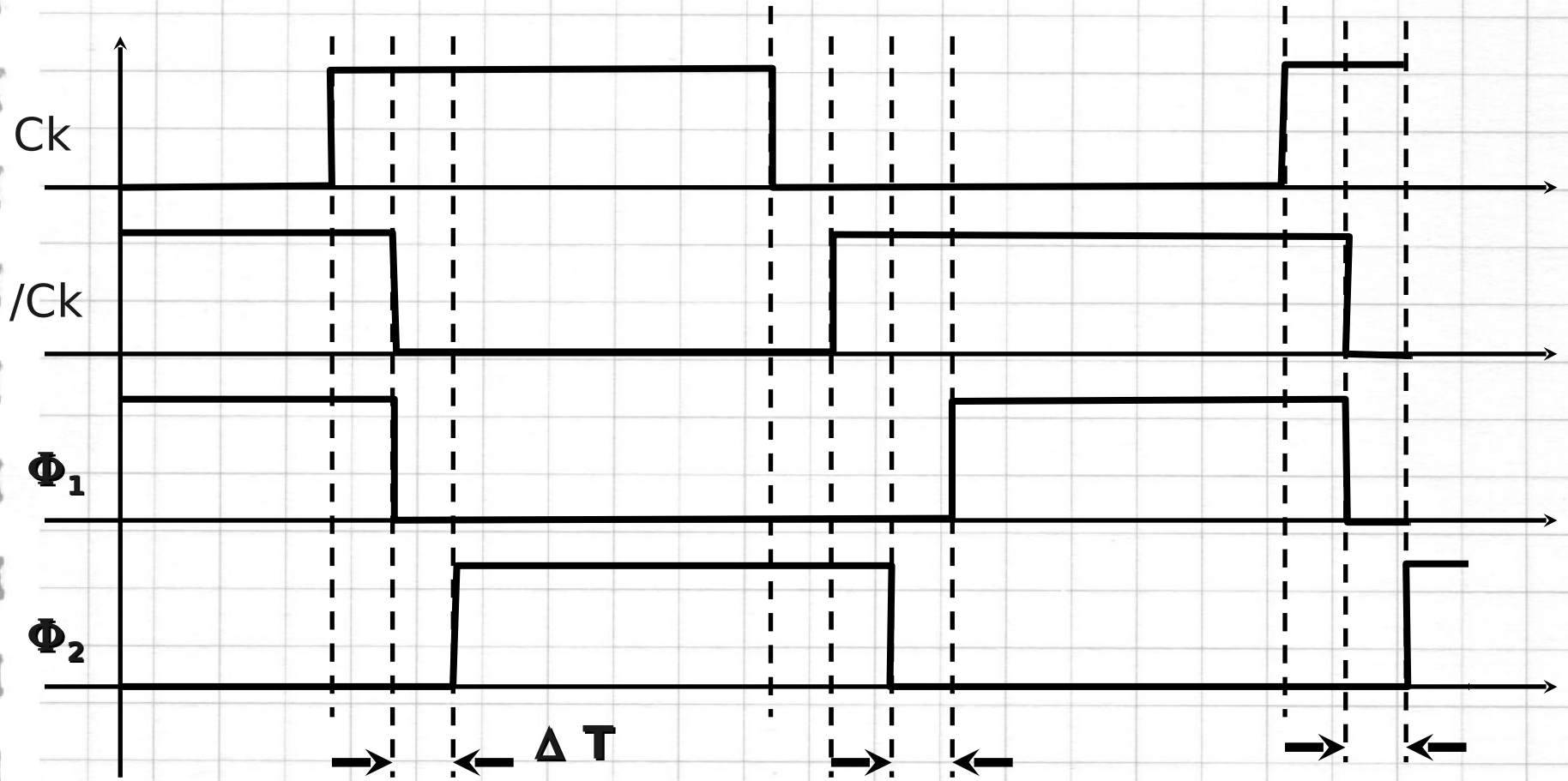


Clock a due fasi

•Un altro modo di generare il clock a due fasi non sovrapposte

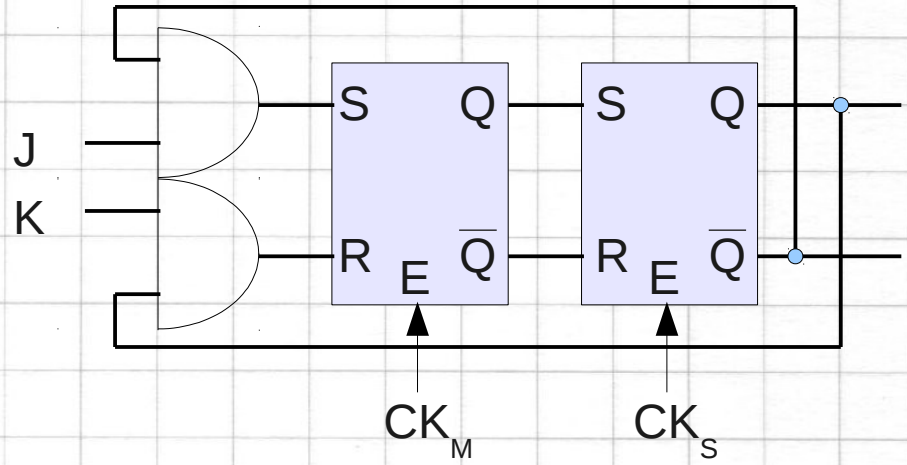


Forme d’onda

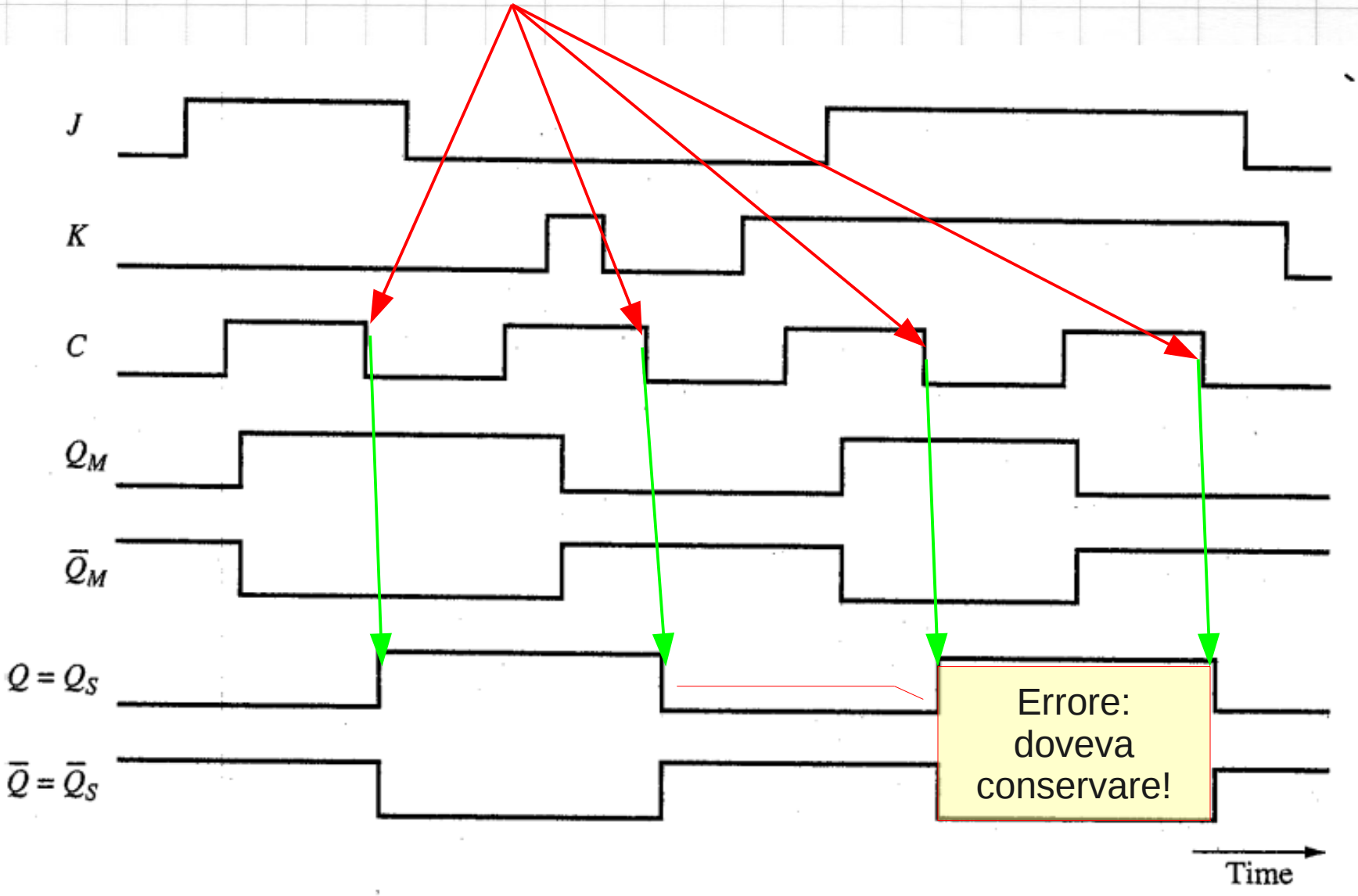


Il flip-flop JK

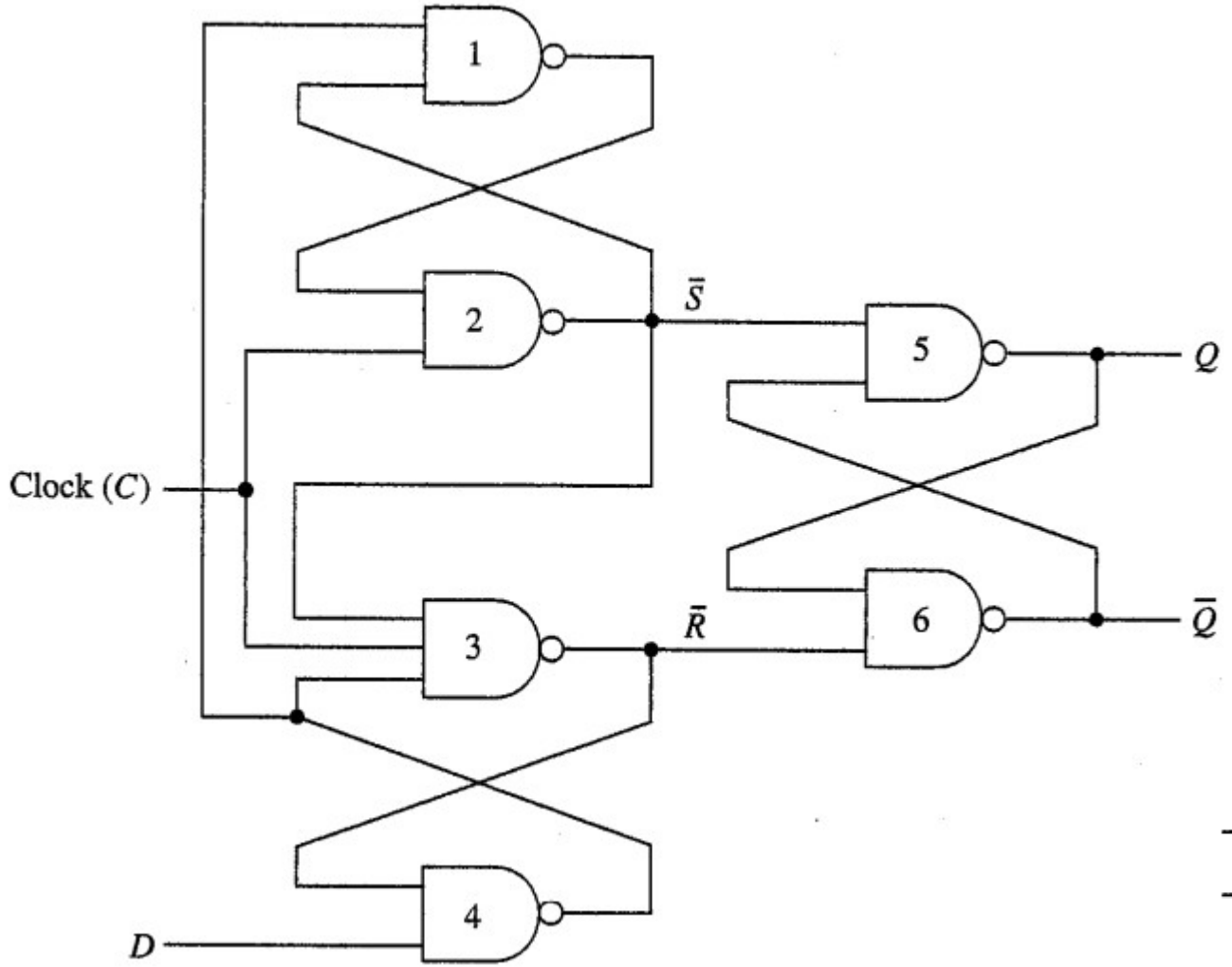
Inputs			Outputs	
J	K	C	Q^+	$\overline{Q}^+$
0	0	↑	Q	$\overline{Q}$
0	1	↑	0	1
1	0	↑	1	0
1	1	↑	$\overline{Q}$	Q
X	X	0	Q	$\overline{Q}$



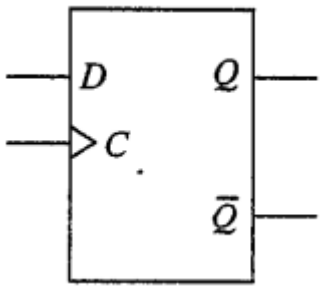
Problema: impulso in ingresso



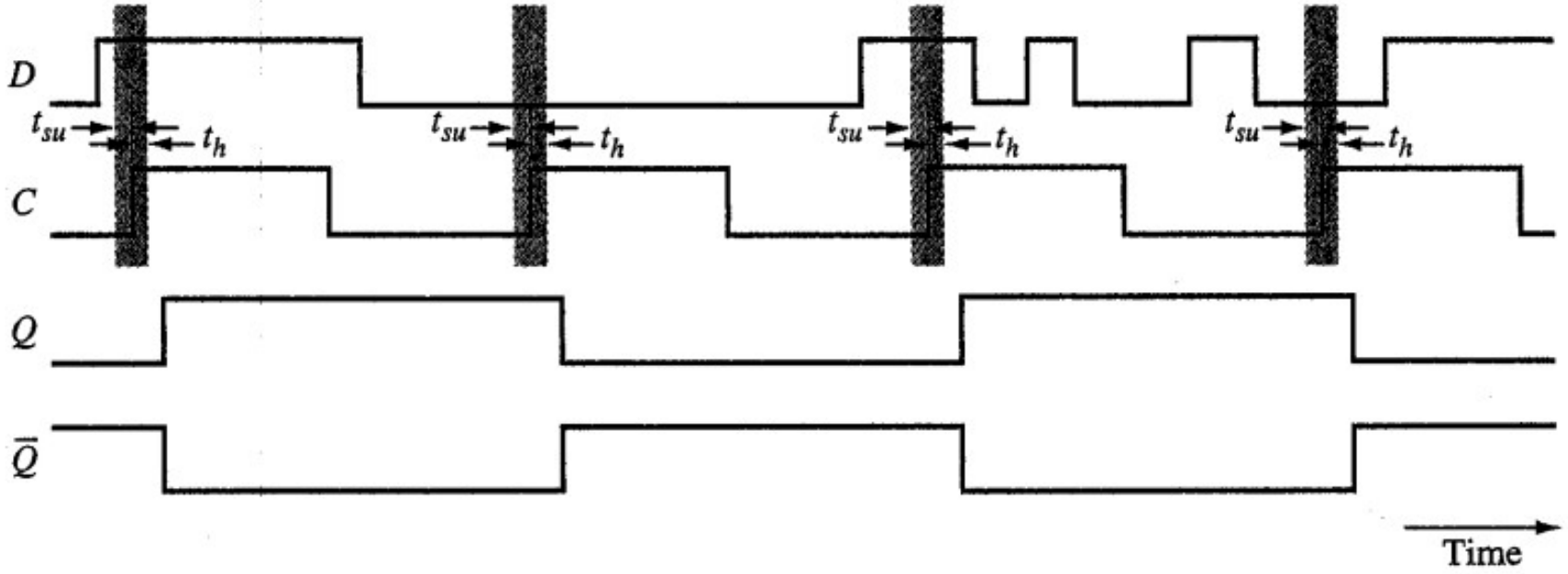
Flip-flop sensibili al fronte



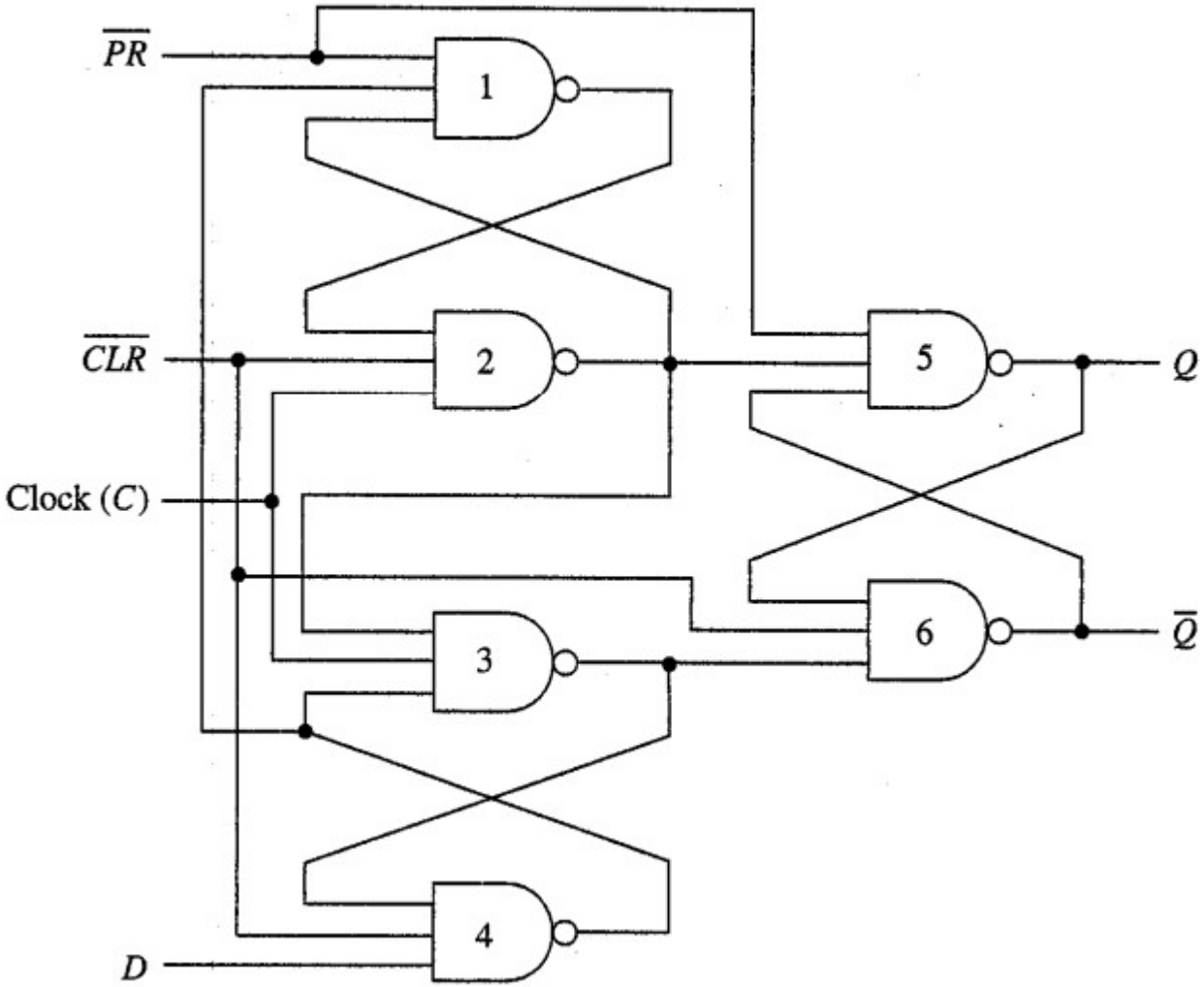
Inputs		Outputs	
D	C	Q^+	$\bar{Q}^+$
0	↑	0	1
1	↑	1	0
X	0	Q	$\bar{Q}$
X	1	Q	$\bar{Q}$



Timing del flip-flop D

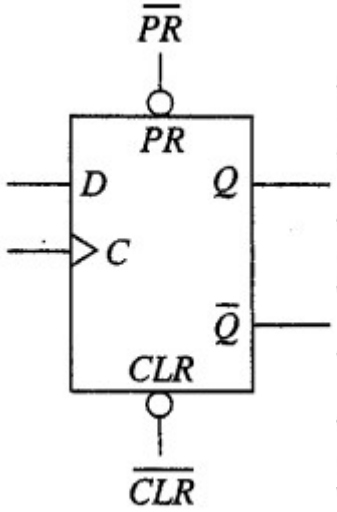


Ingressi asincroni



Inputs				Outputs	
$\overline{PR}$	$\overline{CLR}$	D	C	Q^+	$\overline{Q}^+$
0	1	X	X	1	0
1	0	X	X	0	1
0	0	X	X	1*	1*
1	1	0	↑	0	1
1	1	1	↑	1	0
1	1	X	0	Q	$\overline{Q}$
1	1	X	1	Q	$\overline{Q}$

*Unpredictable behavior will result if $\overline{PR}$ and $\overline{CLR}$ return



I flip-flop non trasparenti

- Tipi di flip-flop non trasparenti
  - Flip-flop D
    - L'uscita assume il valore di *D*
  - Flip-flop D con abilitazione o DE
    - Con *E* = 0 l'uscita si conserva
    - Con *E* = 1 l'uscita assume il valore di *D*
  - Flip-flop JK
    - Con *J* = 0 e *K* = 0 l'uscita si conserva
    - Con *J* = 0 e *K* = 1 l'uscita vale 0 (reset)
    - Con *J* = 1 e *K* = 0 l'uscita vale 1 (set)
    - Con *J* = 1 e *K* = 1 l'uscita si complementa
  - Flip-flop T
    - Con *T* = 0 l'uscita si mantiene
    - Con *T* = 1 l'uscita si complementa

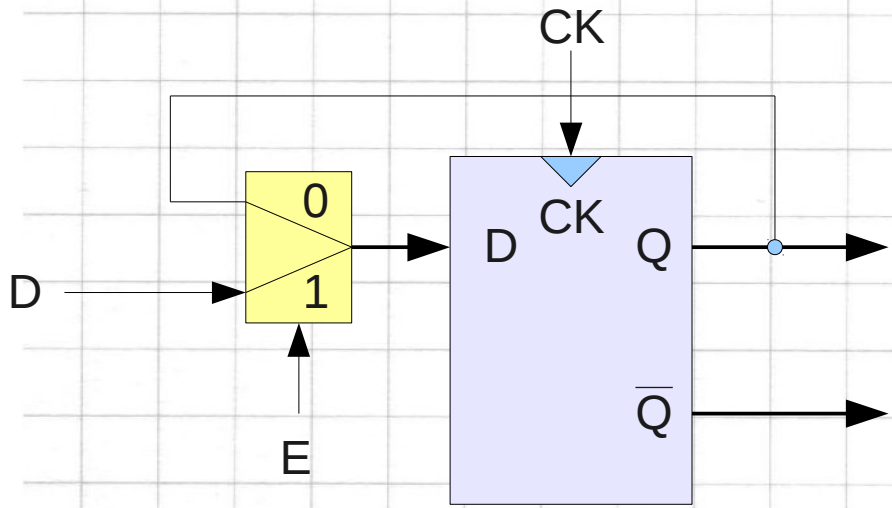
Equazioni caratteristiche

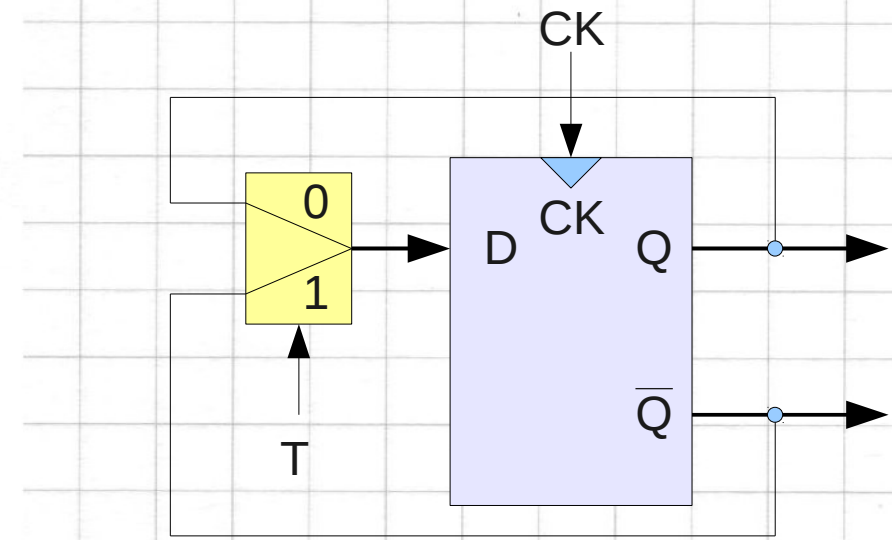
- Modo alternativo di esprimere il funzionamento di flip-flop non trasparenti
- Flip-flop D
  - $Q' = D$
- Flip-flop DE
  - $Q' = Q\overline{E} + DE$
- Flip-flop JK
  - $Q' = J\overline{K} + Q\overline{J}\,\overline{K} + \overline{Q}JK = J\overline{Q} + \overline{K}Q$
- Flip-flop T
  - $Q' = T\overline{Q} + \overline{T}Q = T \oplus Q$

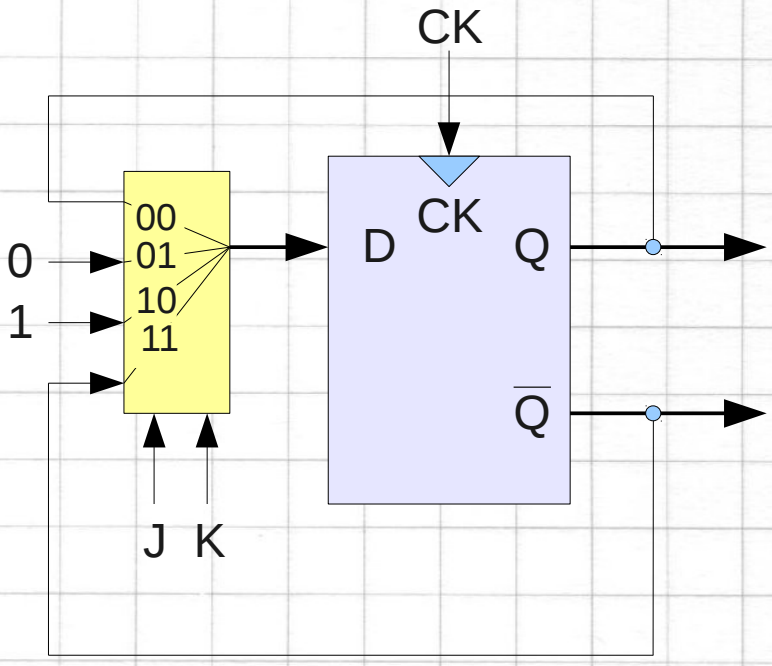
Da un tipo di flip-flop all'altro

- Ogni flip-flop può essere usato per realizzare le funzioni svolte dagli altri tipi
  - Con l'aggiunta eventuale di una rete logica
- Per dimostrare questa affermazione
  - 1) usiamo il flip-flop D per realizzare tutti gli altri tipi
  - 2) usiamo DE, JK e T per realizzare il D
- Osservazione
  - Il procedimento proposto ha significato concettuale, le realizzazione ottenute non sono ottime
  - Ogni tipo di flip-flop risulta più adatto a particolari tipi di applicazione

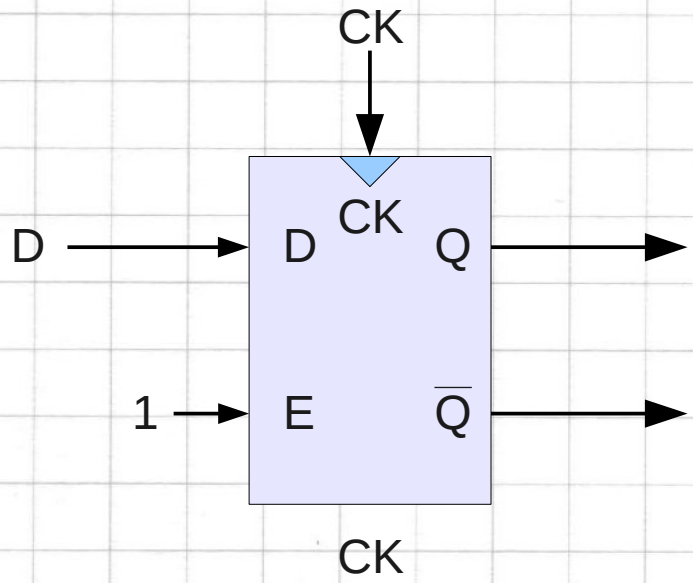
Da flip-flop D a DE, JK, T

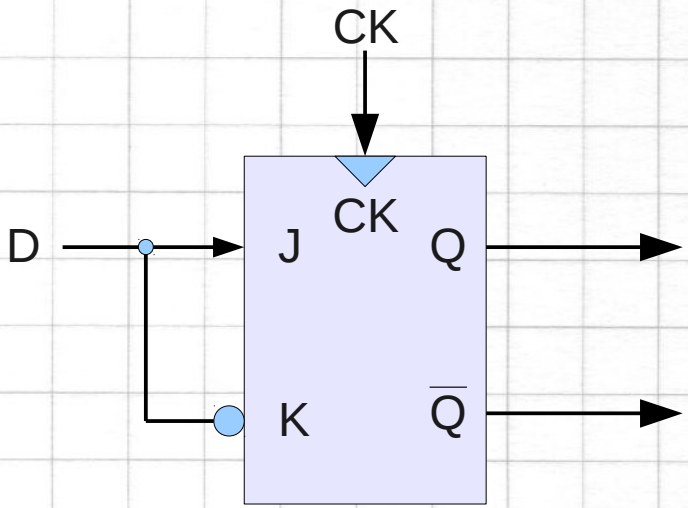


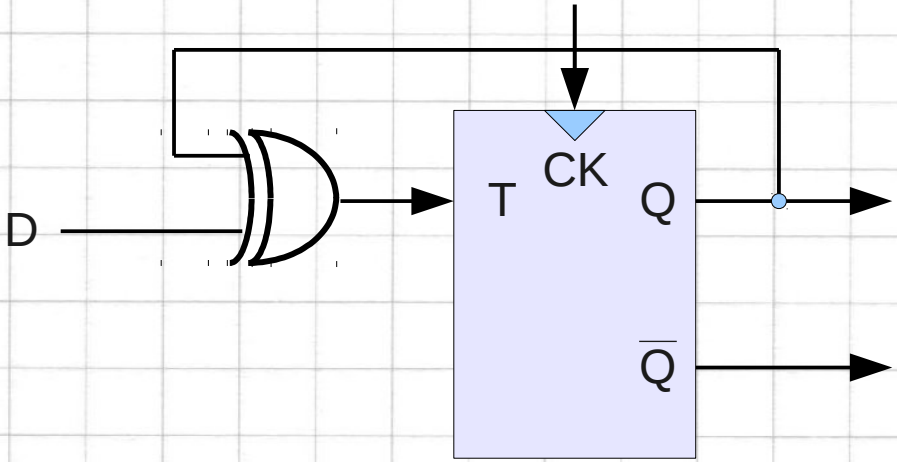




Da flip-flop DE, JK, T a D





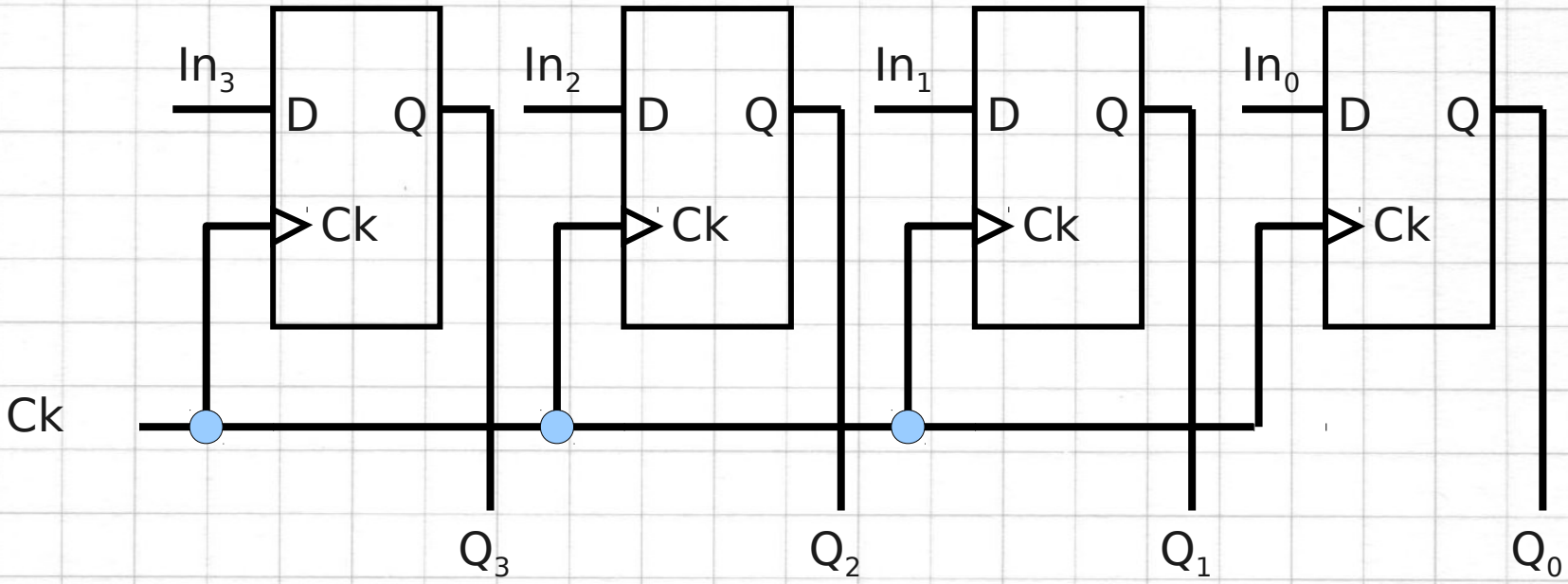


I registri

- Array di FF, in grado di memorizzare più bit di informazione
  - Si caratterizzano per il modo con cui avviene la memorizzazione e la lettura dei dati
    - Parallelo o seriale nel tempo
- Possono eseguire manipolazione dei bit
  - Scorrimenti (shift) destri o sinistri dei bit
- Più funzionalità possono essere incorporate in un unico dispositivo
  - Registro universale, usato anche all'interno dei microcontrollori

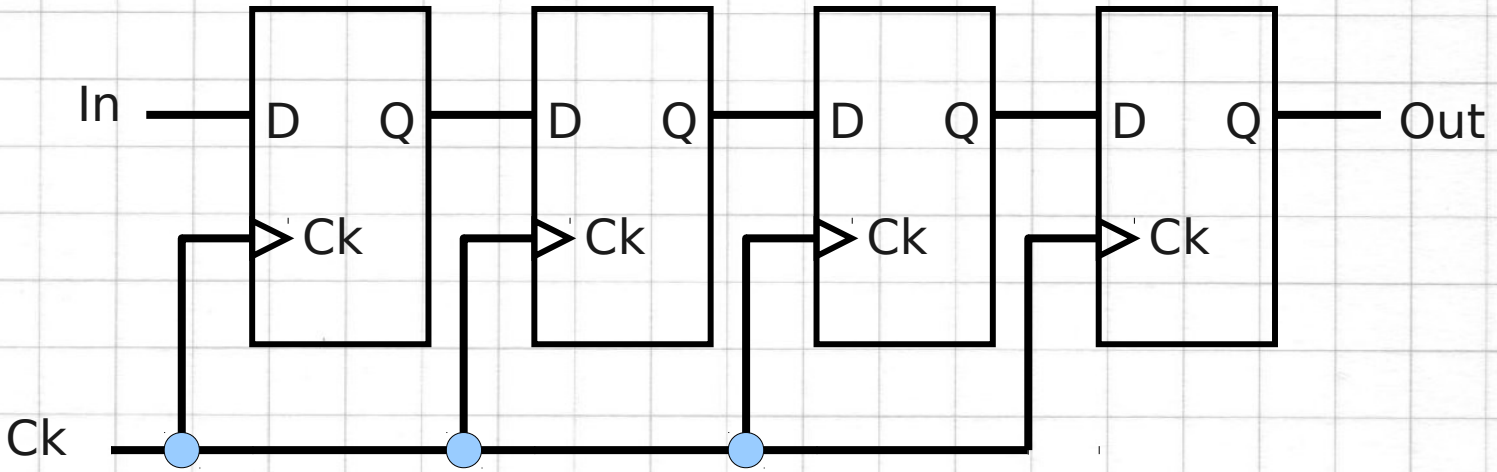
Registro parallelo

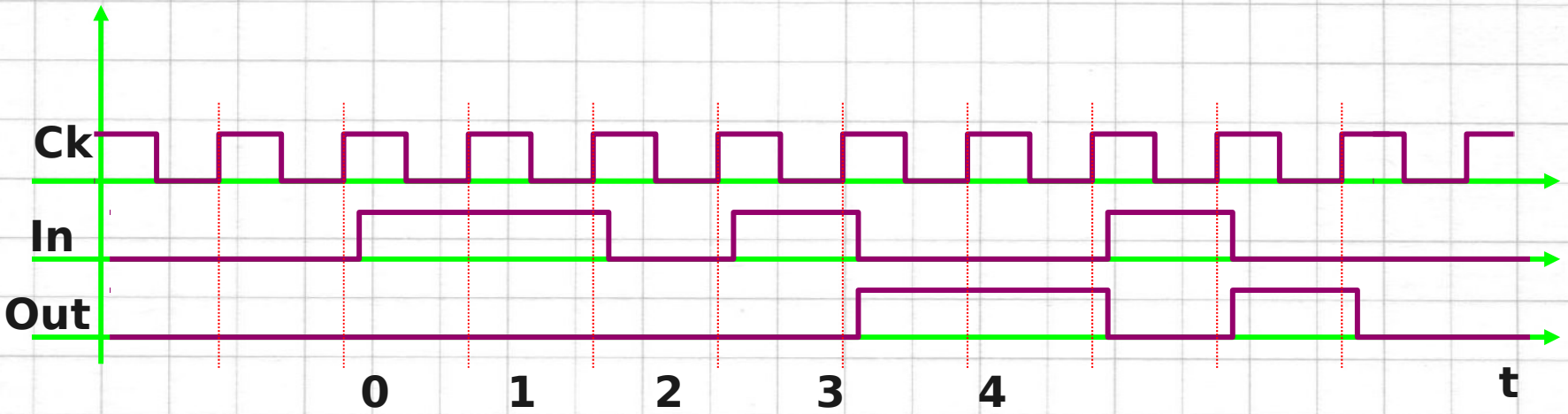
Parallel In - Parallel Out (PIPO)



Registro a scorrimento (1) (shift register)

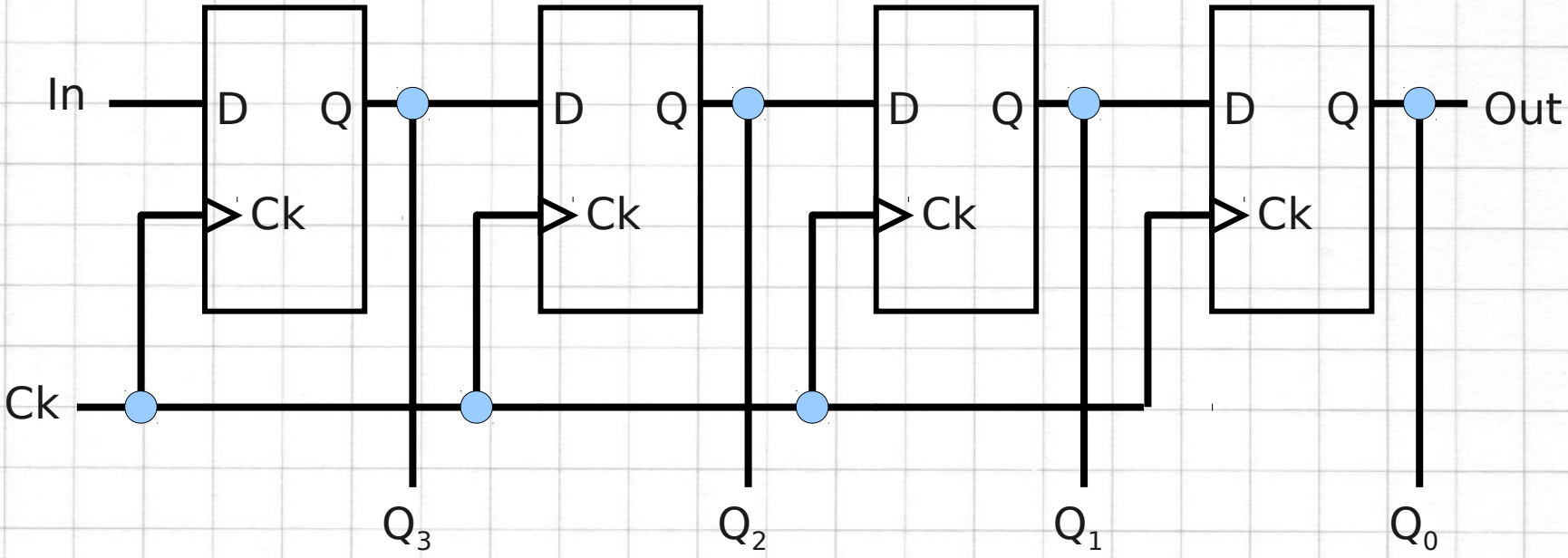
Serial In - Serial Out (SISO)





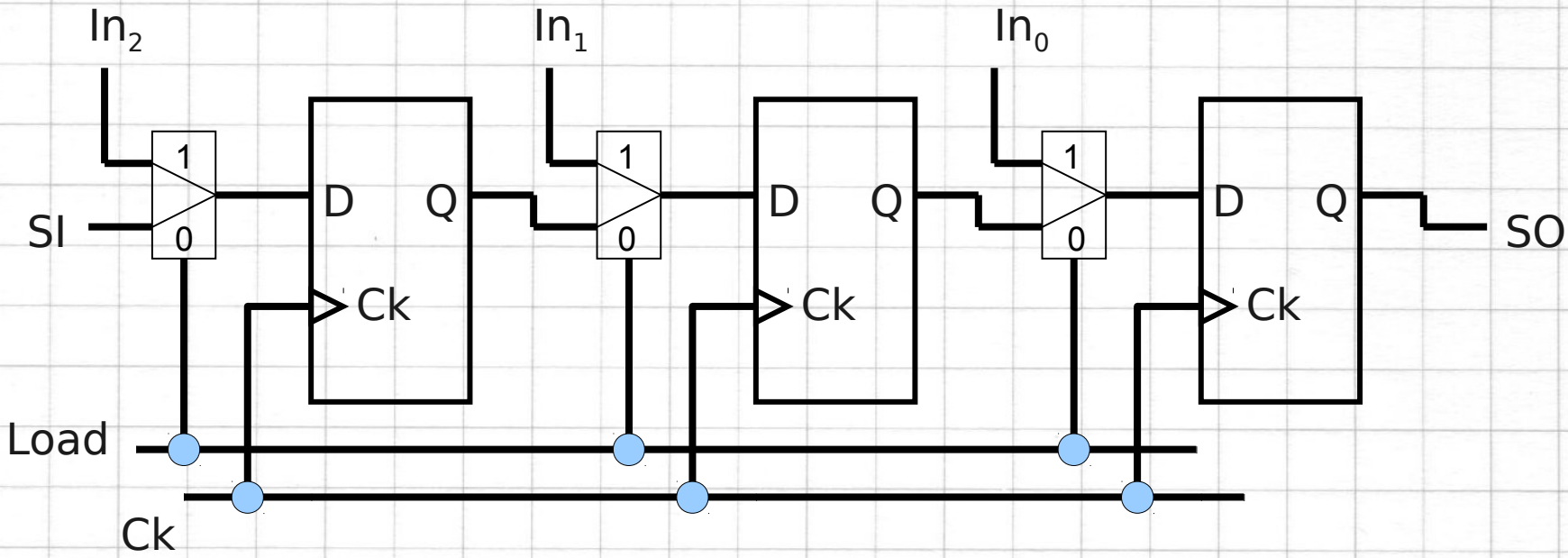
Registro a scorrimento (2) (shift register)

Serial In - Parallel Out (SIPO)

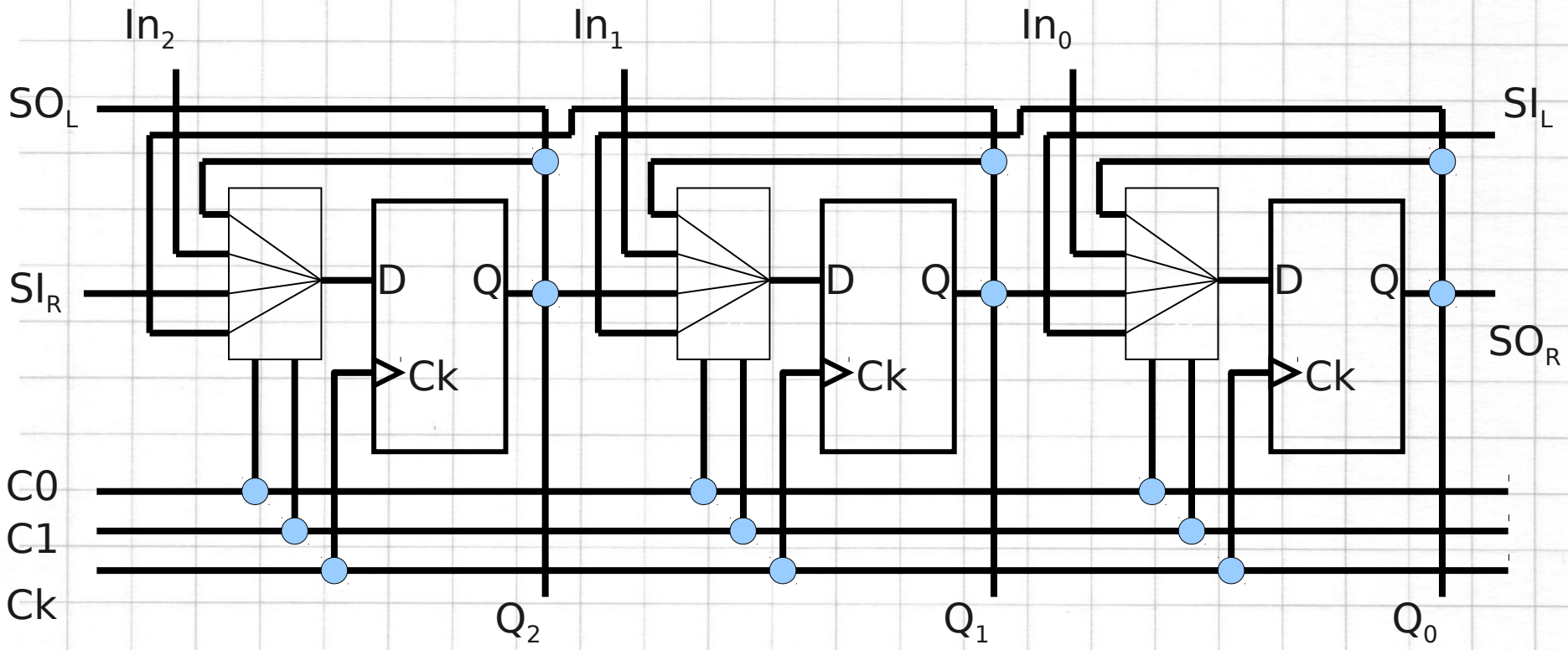


Registro a scorrimento (3) (shift register)

Parallel In - Serial Out (PISO)



Registro universale

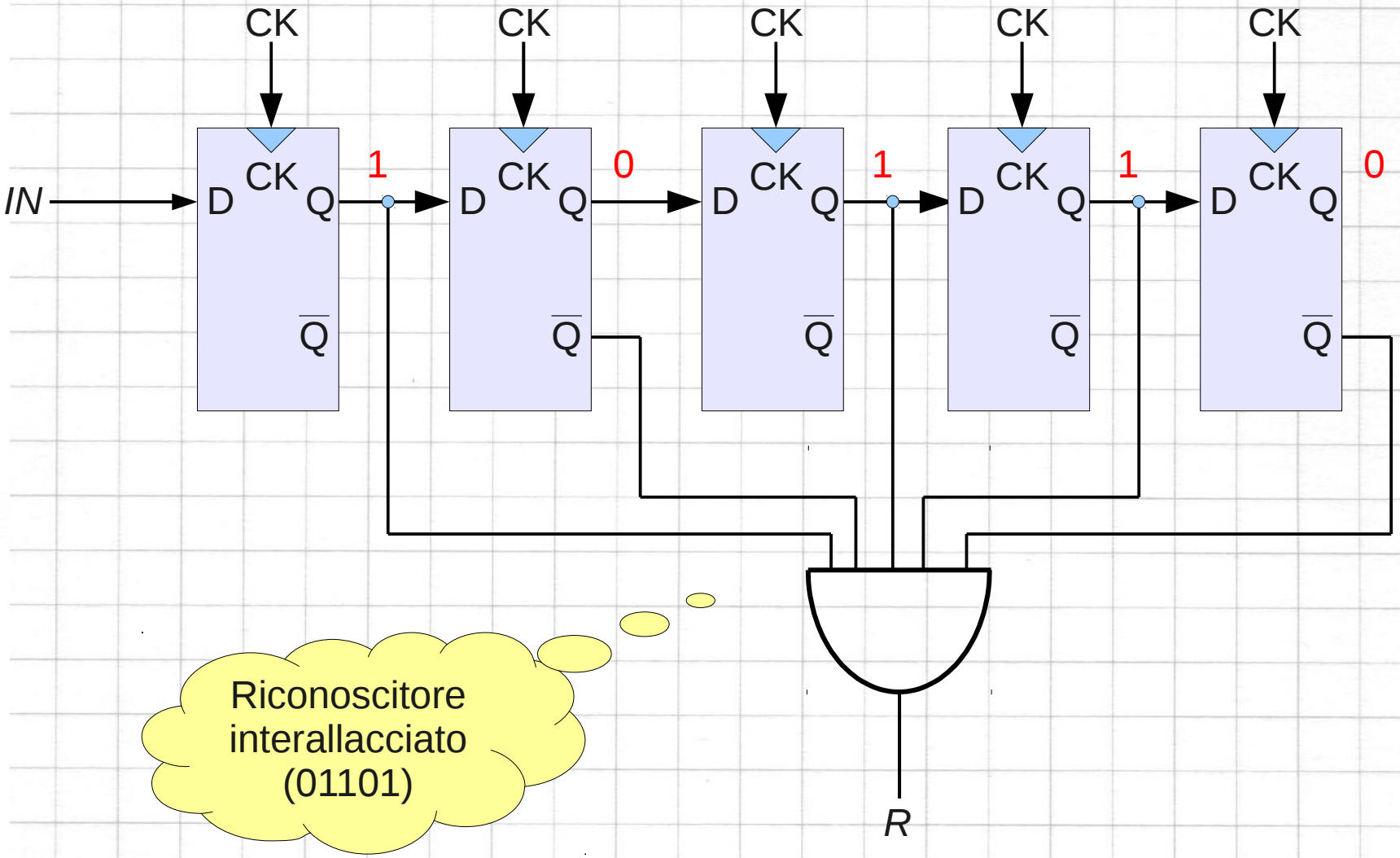


C_1	C_0	Effetto
0	0	Conserva
0	1	Ingresso parallelo
1	0	Shift destro
1	1	Shift sinistro

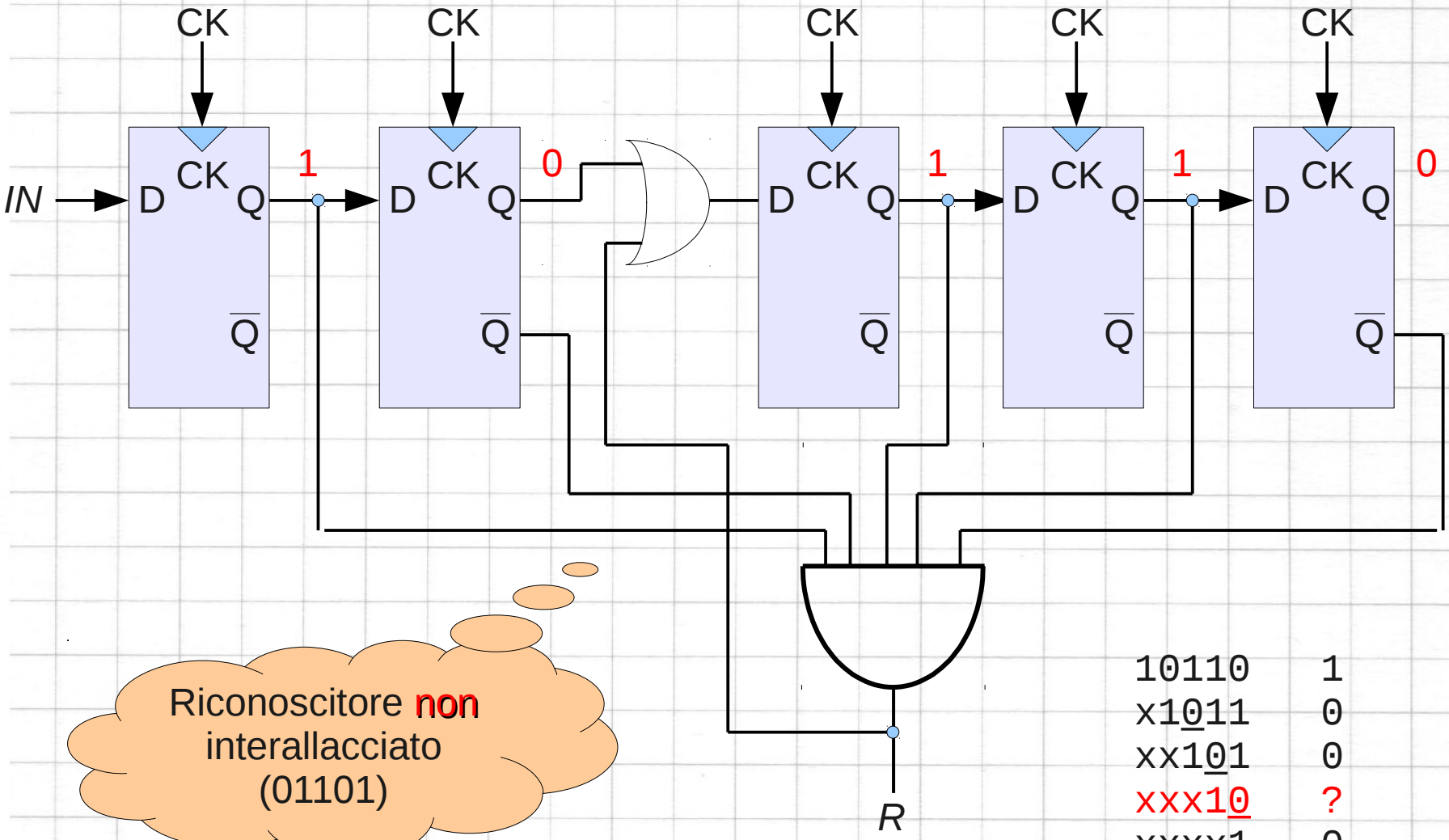
Riconoscitori di sequenza

- Macchine sequenziali con un uscita che si attiva quando l'ingresso presenta in sequenza una serie preordinata di valori
  - Esempio di sequenza: 01101
  - Riconoscimento di sequenze interallacciate
    - I bit di una sequenza riconosciuta possono essere considerati parte di una nuova sequenza
      - Nell'esempio, lo ...01 finale può essere considerato nuovamente
  - Sequenze non interallacciate
    - Una volta riconosciuti, i bit non si usano più
  - Esempio di funzionamento
    - `In:    ...0111011011011011011011010100 0...`
    - `Out_I: ...0000000100100100100001001000 00...`
    - `Out_NI:...0000000100000100000001000000 00...`

Riconoscitori con shift (1)



Riconoscitori con shift (2)



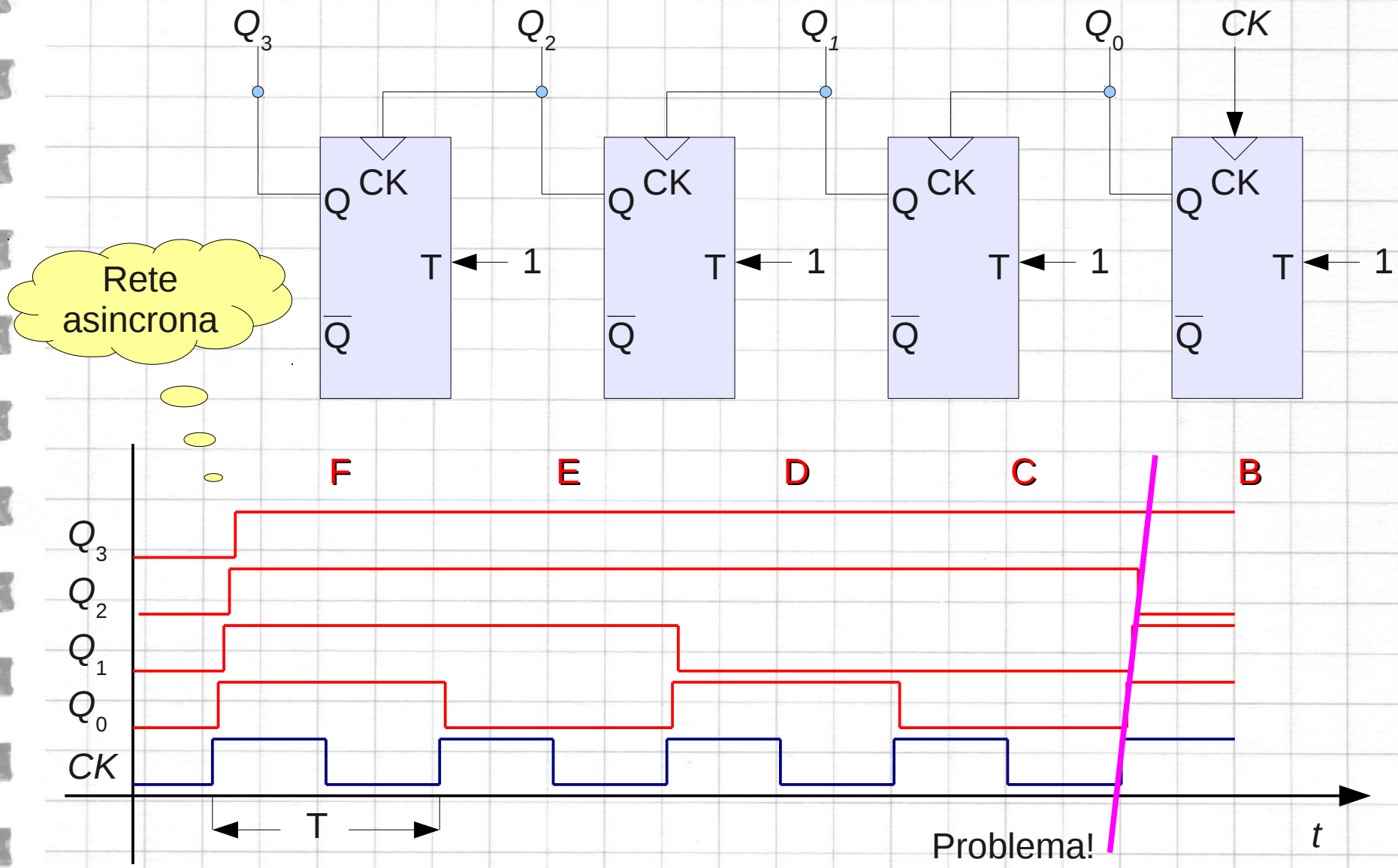
Riconoscitore non interallacciato (01101)

10110	1
x1011	0
xx101	0
xxx10	?
xxxx1	0

Contatori

- Macchine sequenziali
  - Sincrone o asincrone
- Interfaccia
  - Hanno in ingresso un'onda quadra
  - Generano una sequenza di valori predefiniti
    - Ogni valore è ricavato dal precedente
      - Sommando o sottraendo un valore costante
    - Alla fine del conteggio possono comportarsi diversamente
      - Ripartire da 0
      - Modulo *n*
      - Restare al valore massimo raggiunto (saturazione)
    - Possono avere diversi ingressi per funzioni opzionali
      - Azzeramento (Reset)
      - Abilitazione (Enable)
      - Caricamento parallelo (Count/$\overline{\text{Load}}$)
      - Direzione del conteggio (Up/$\overline{\text{Down}}$)

Divisori di frequenza



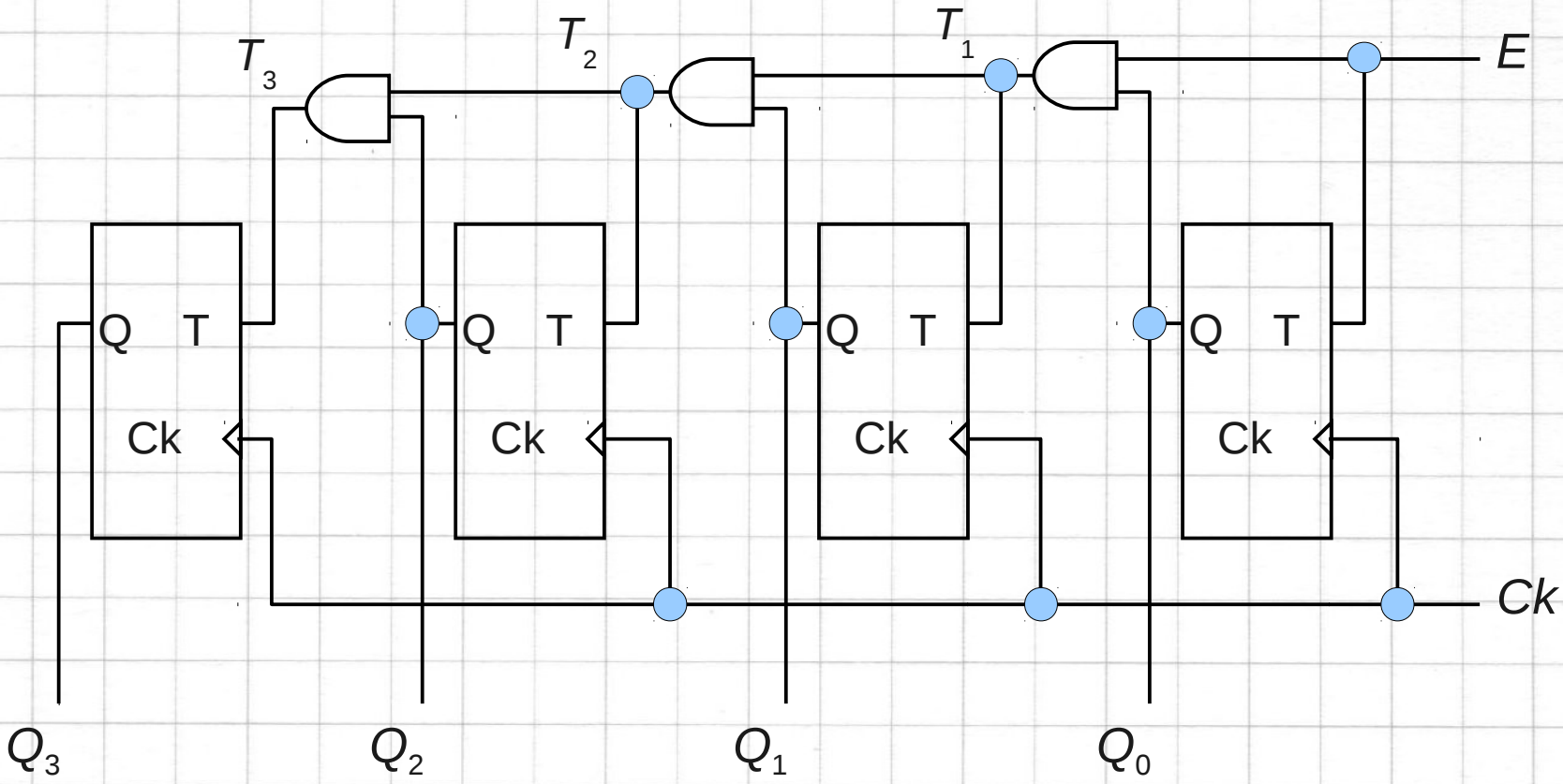
Contatori modulo *n*

- Macchine sequenziali che generano, ciclicamente e in ordine, valori tra 0 e *n* - 1
  - Al termine del conteggio ripartono da capo
  - Categoria particolare: contatori modulo 2^m
    - Richiedono *m* variabili di stato
- Possibili variabili di controllo
  - Enable: 1 (abilita il conteggio); 0 (conserva)
  - Reset: 1 (azzera); 0 (conta normalmente)
  - Preset: 1 (tutti i bit accesi); 0 (conta normalmente)
  - Up/$\overline{\text{Down}}$: 1 (conteggio crescente); 0 (decrescente)
  - Load: 1 (carica il valore); 0 (conta normalmente)

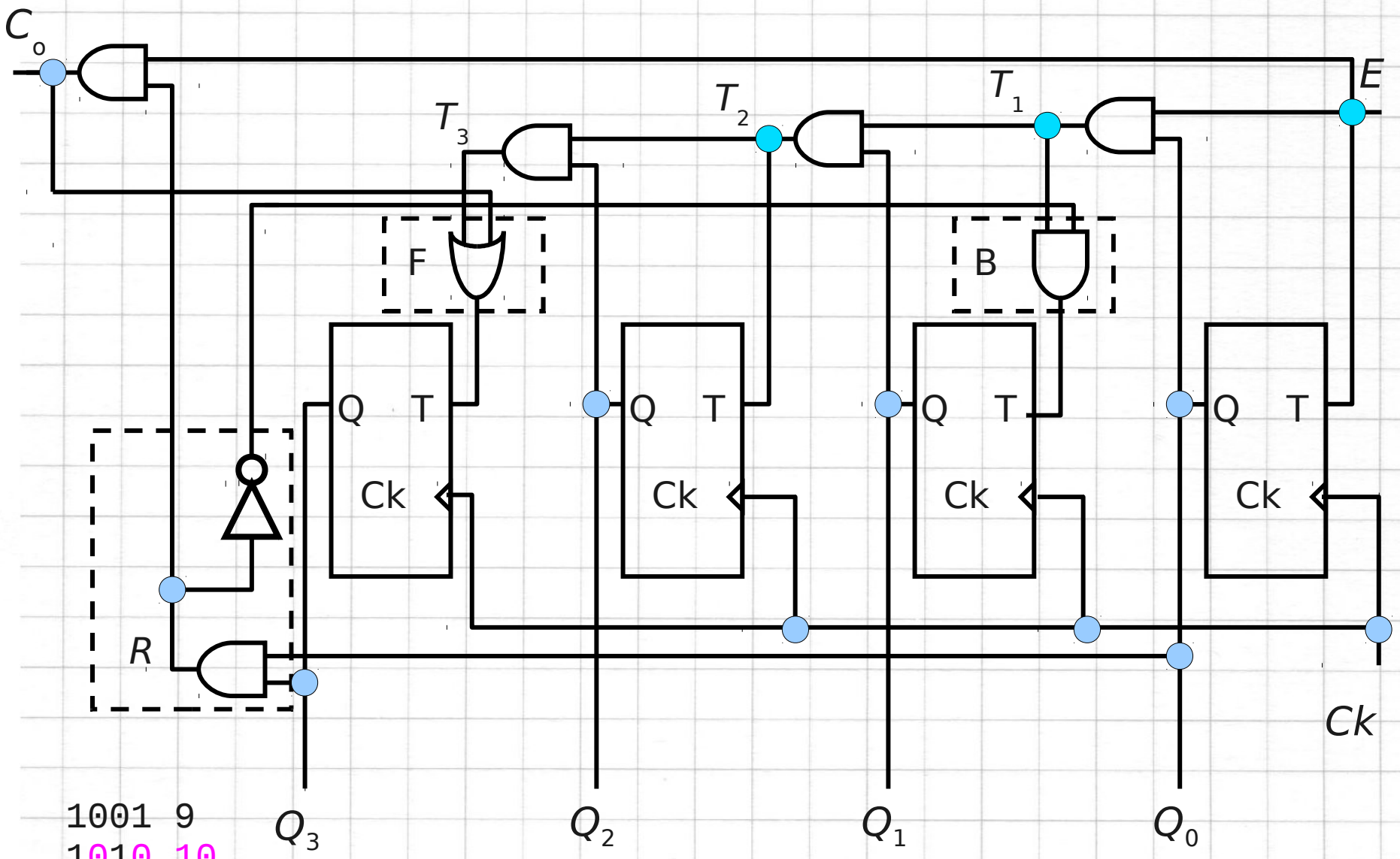
Contatore sincrono modulo 2^n

$$T_i = E\,Q_{i-1}\,Q_{i-2}\cdots Q_0$$
$$T_{i+1} = E\,Q_i\,Q_{i-1}\,Q_{i-2}\cdots Q_0 = T_i\,Q_i$$

Un bit commuta se:
1) il contatore è abilitato
2) i bit di peso inferiore al clock precedente sono tutti a 1

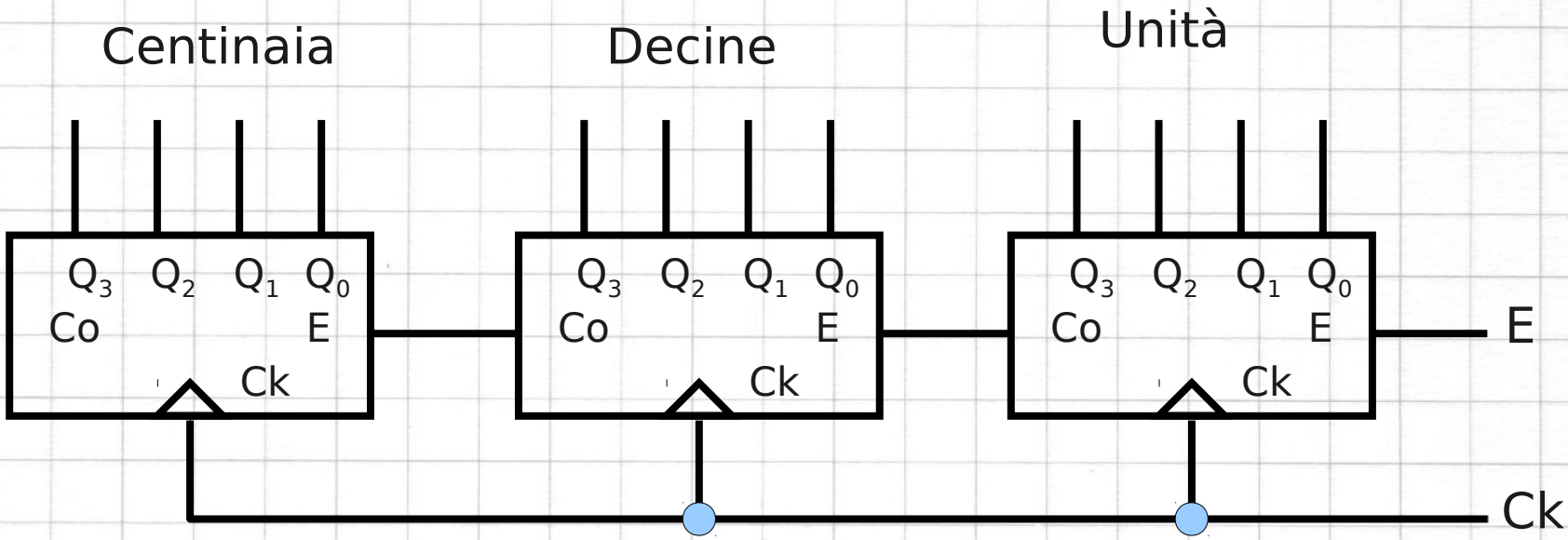


Contatore sincrono modulo 10

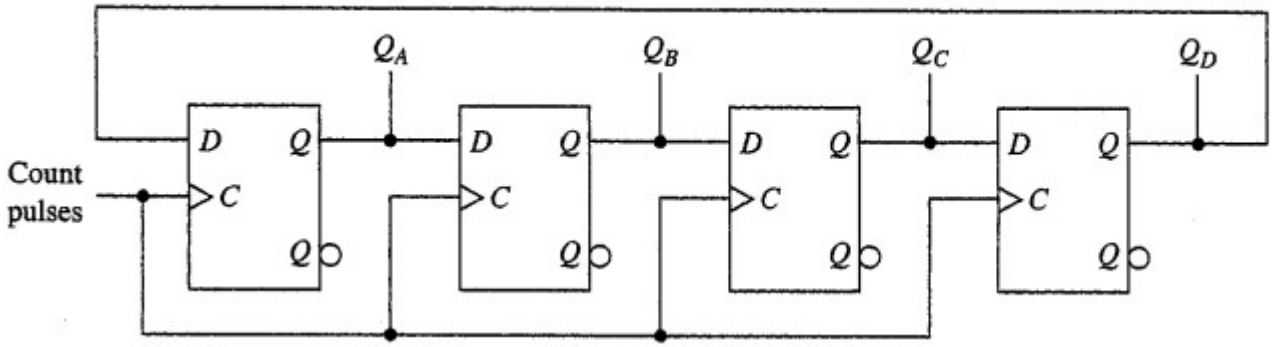



```
1001 9
1010 10
0000 0
F-B-
```

Contatori decadici in cascata

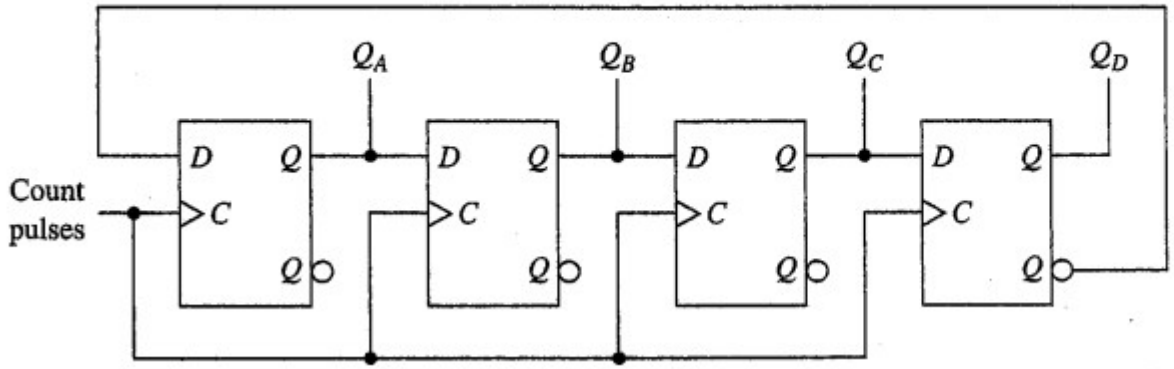


Contatori ad anello e Johnson



Q_A	Q_B	Q_C	Q_D
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0

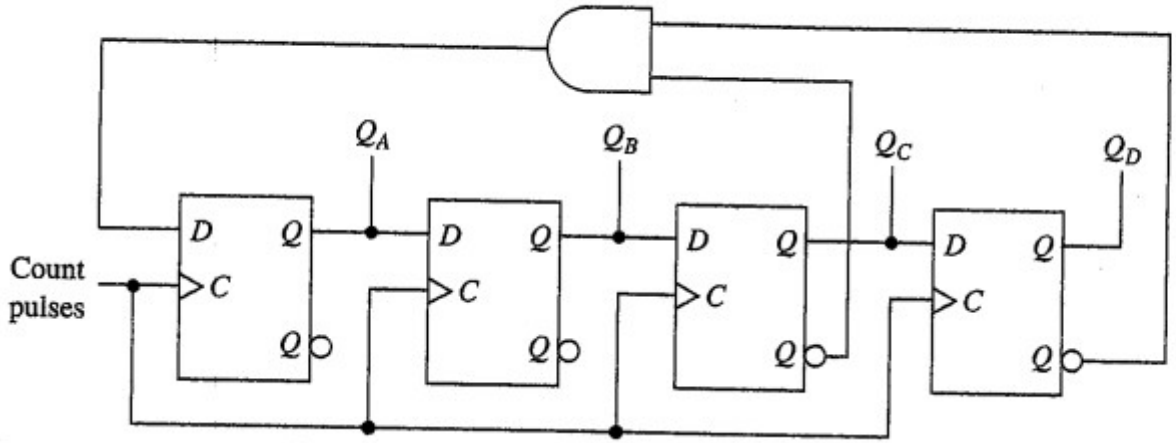
etc.



Q_A	Q_B	Q_C	Q_D	And-gate inputs
0	0	0	0	$\overline{Q}_A\overline{Q}_D$
1	0	0	0	$Q_A\overline{Q}_B$
1	1	0	0	$Q_B\overline{Q}_C$
1	1	1	0	$Q_C\overline{Q}_D$
1	1	1	1	Q_AQ_D
0	1	1	1	$\overline{Q}_AQ_B$
0	0	1	1	$\overline{Q}_BQ_C$
0	0	0	1	$\overline{Q}_CQ_D$
0	0	0	0	

etc.

Contatore Johnson mod 7



Q_A	Q_B	Q_C	Q_D	And-gate inputs
0	0	0	0	$\overline{Q}_A\overline{Q}_D$
1	0	0	0	$Q_A\overline{Q}_B$
1	1	0	0	$Q_B\overline{Q}_C$
1	1	1	0	$Q_C\overline{Q}_D$
0	1	1	1	$\overline{Q}_AQ_B$
0	0	1	1	$\overline{Q}_BQ_C$
0	0	0	1	$\overline{Q}_CQ_D$
0	0	0	0	

etc.

Il problema dell'inizializzazione

- In generale, una macchina sequenziale con n flip-flop ha 2^n stati
- Se la sequenza del contatore ha $m < 2^n$ valori, esistono valori non compresi nel conteggio
  - Occorre porsi la domanda: come procede la sequenza se accidentalmente il contatore viene a trovarsi in uno di questi $2^n - m$ valori?
  - Possibili risposte
    - In modo arbitrario, ma tornando dopo un numero finito di cicli a un valore appartenente alla sequenza
    - In modo arbitrario, e senza mai tornare a un valore valido della sequenza
      - Questa condizione è da evitare
      - All'accensione lo stato dei flip-flop è (in genere) casuale