



# Circuiti sequenziali e latch

Prof. Alberto Borghese  
Dipartimento di Scienze dell'Informazione  
[borgnese@dsi.unimi.it](mailto:borgnese@dsi.unimi.it)

Università degli Studi di Milano



## Sommario

### Circuiti sequenziali

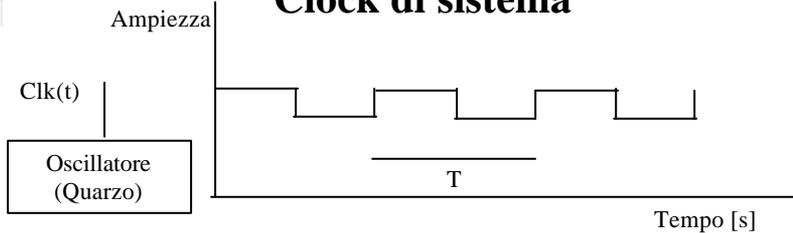
Latch asincroni

Latch sincroni SR

Latch sincroni D



# Clock di sistema



Frequenza: numero di cicli/s Si misura in Hertz, Hz).  
 Periodo: tempo necessario a completare 1 ciclo Si misura in secondi, s).

$$T = 1/f$$

Tempo di salita e discesa trascurabile rispetto a quello delle porte logiche.



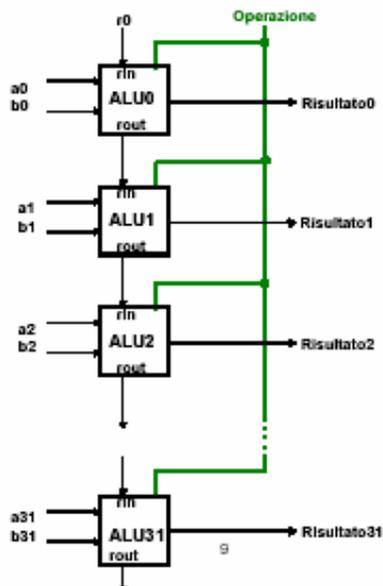
# Perchè esiste il clock?



Ogni quanto tempo possiamo presentare gli ingressi al sommatore?

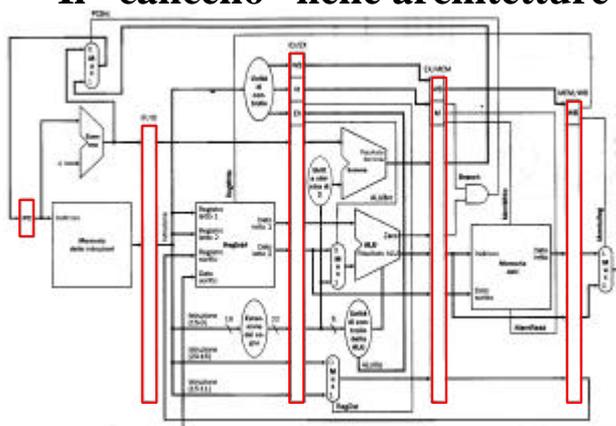
Dobbiamo essere ragionevolmente sicuri che il risultato sia stato calcolato ed utilizzato.

“Cancello” davanti ai bit di A e B.





## Il “cancello” nelle architetture



FF — Decod — Exec — R/W — WB

Il “cancello” viene inserito solamente in alcuni punti dell’architettura.  
In questi punti si **sincronizza** l’attività: “*nodi di sincronizzazione*”.

Il clock diventa indispensabile per sincronizzare il funzionamento delle varie componenti nelle architetture retro-azionate.



## Circuiti sincroni e asincroni



- **Architettura logica sincrona:**

Le fasi di elaborazione sono scandite da un orologio comune a tutto il circuito (**clock**).

- Ad ogni fase di clock, la parte combinatoria del circuito ha tempo di elaborare (i segnali di percorrere il cammino critico) e quindi il circuito ha il tempo di stabilizzarsi (transitori critici). Questo deve avvenire entro il “**tick**” successivo.
- **Progetto sincrono:** il controllo dei transitori/cammini critici è limitato alla parte di circuito tra due **cancelli** (porte di **sincronizzazione**)

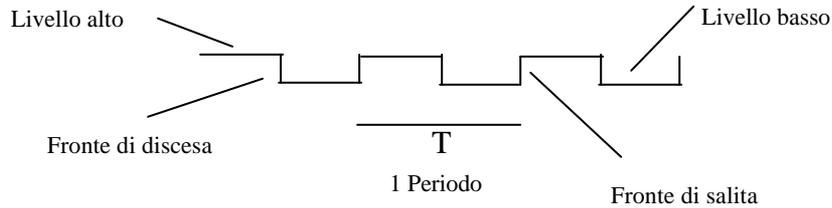
- **Architettura logica asincrona:**

L’elaborazione e propagazione dei segnali avviene in modo incontrollato, secondo le velocità di propagazione dei circuiti.

- Non devo mai aspettare il “**tick**” di un clock ? **massima velocità**
- **Progetto asincrono:** Devo progettare il circuito in modo che nessun transitorio/cammino critico causi problemi ? analisi di tutti i transitori critici possibili. Improprio per circuiti con feed-back.



## Utilizzo del clock



- **Metodologia sensibile ai livelli:**

Le variazioni di stato possono avvenire per tutto il periodo in cui il clock è al livello alto (basso).

- **Metodologia sensibile ai fronti:**

Le variazioni di stato avvengono solamente in corrispondenza di un fronte di clock. Noi adotteremo questa metodologia.



## Architetture sequenziali



- I circuiti combinatori **non hanno memoria**. Gli output al tempo  $t$  dipendono unicamente dagli input al tempo  $t$ .
- Sono necessari circuiti con memoria, per consentire comportamenti diversi a seconda della situazione dell'architettura. Nella memoria viene memorizzato lo **stato** del sistema.
- Un esempio classico è il distributore automatico, che deve ricordare quante e quali monete sono state inserite, e che deve comportarsi tenendo conto non solo delle monete inserite attualmente ma anche di quelle inserite in precedenza.
- I circuiti che hanno elementi di memoria consentono di eseguire operazioni sequenzialmente (scandite dal clock).



# Sommario



Circuiti sequenziali

Latch asincroni

Latch sincroni SR

Latch sincroni D



# Latch & bistabili



Elemento cardine dei circuiti sequenziali è lo **stato**. Lo stato riassume il funzionamento negli istanti precedenti e deve essere immagazzinato (memorizzato).

Necessità di elementi con memoria (bistabili -> registri -> memorie).

Elemento base della memoria è il **bistabile**: dispositivo in grado di mantenere *indefinitamente* un certo valore di input (0 o 1).

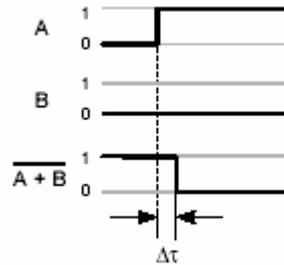
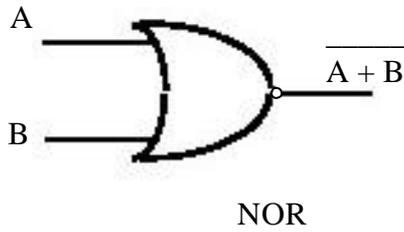
Il suo valore di uscita coincide con lo stato. L'uscita al tempo  $t$ , dipende dallo stato al tempo  $t-1$  e dal valore presente agli input.

## **Tipi di bistabili:**

- Bistabili non temporizzati (asincroni) / temporizzati (sincroni).
- Bistabili che commutano sul livello del clock (**latch**) o sul fronte (**flip-flop**).



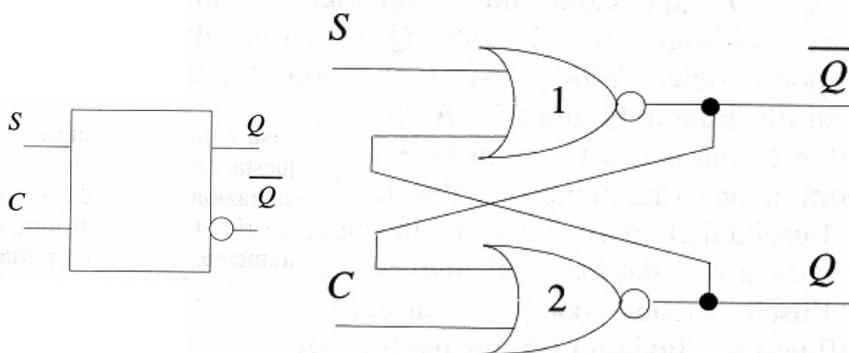
## Principio di funzionamento



Il ritardo,  $\Delta\tau$ , introdotto tra la commutazione dell'input e la commutazione dell'output è alla base del funzionamento di un bistabile.



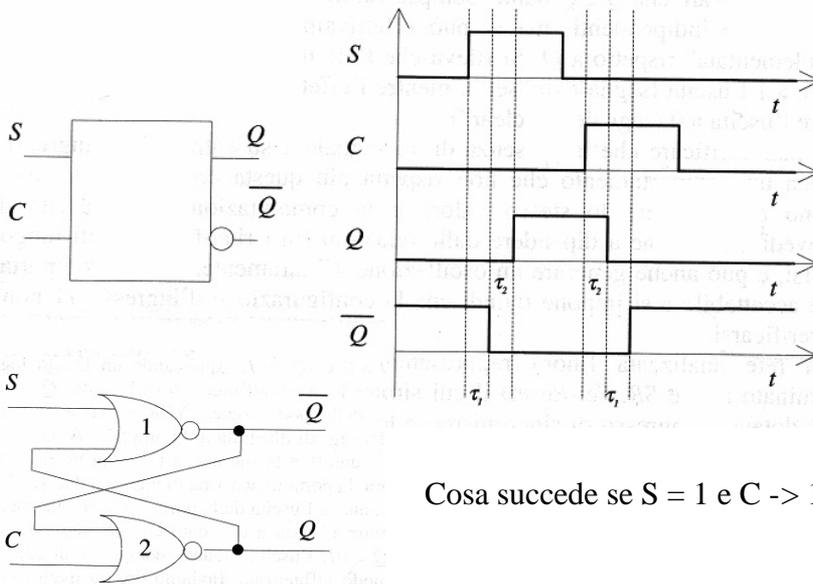
## Latch asincrono SC (o SR)



Una coppia di porte NOR retro-azionate può memorizzare un bit.



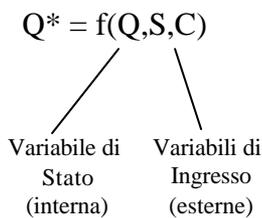
# Funzionamento del circuito SC



Cosa succede se  $S = 1$  e  $C \rightarrow 1$ ?



# Tabella delle transizioni



Q	SC = 00	SC = 01	SC = 11	SC = 10
0	0	0	X	1
1	1	0	X	1

No change ( $Q^* = Q$ )     
 Clear Reset     
 Set

Q è l'uscita del latch: **stato presente**.

Q\* è il valore dell'uscita al tempo successivo: **stato prossimo**.



## Tabella delle verità di SC



- Se considero Q (lo stato) e S e C come ingressi, ottengo la **tabella delle verità di Q\***:



Q	SC = 00	SC = 01	SC = 11	SC = 10
0	0	0	X	1
1	1	0	X	1

S	C	Q	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	X
1	1	1	X

$$Q^* = \bar{S}\bar{C}Q + S\bar{C}Q + S\bar{C}\bar{Q} = \bar{S}\bar{C}Q + S\bar{C}$$

Status Quo
Set/Reset



## Tabella delle eccitazioni



Q	Q*	S	C
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Data la transizione Q -> Q\*, qual'è la coppia di valori di ingresso che la determina?

$$(Q, Q^*) = f(S, C)$$



# Sommario



Circuiti sequenziali

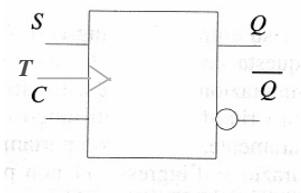
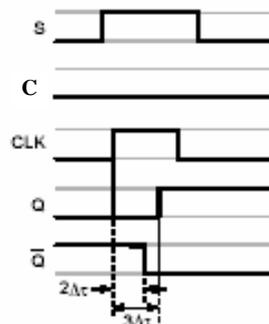
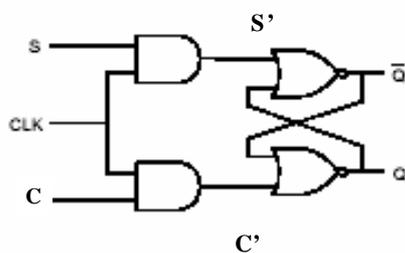
Latch asincroni

Latch sincroni SR

Latch sincroni D



## Il latch SR sincrono



Si inserisce un AND tra il clock e gli ingressi.  
Solo quando il clock è alto i “cancelli”  
rappresentati dagli AND fanno passare gli  
input. Cancelli di abilitazione del latch.



# Tabella delle transizioni



$$Q^* = f(S, C, Q, T)$$

TQ	SC = 00	SC = 01	SC = 11	SC = 10
00	0	0	X	0
01	1	1	X	1
11	1	0	X	1
10	0	0	X	1

Q è l'uscita del latch: **stato presente**.

Q\* è il valore dell'uscita al tempo successivo: **stato prossimo**.



# Tabella delle verità



$$Q^* = f(S, C, Q, T)$$

T	Q	S	C	Q*
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	X = 0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	X = 1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	X = 1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	X = 0

TQ	SC = 00	SC = 01	SC = 11	SC = 10
00	0	0	X	0
01	1	1	X	1
11	1	0	X	1
10	0	0	X	1

$$\begin{aligned}
 Q^* &= \overline{T}QSC + TQSC + \overline{T}QSC + TQSC + \overline{T}QSC + TQSC + \\
 &\quad \overline{T}QSC + \overline{T}QSC = \\
 &= \overline{T}Q\overline{C} + TQ\overline{C} + \overline{T}Q\overline{C} + TQ\overline{C} = \\
 &= \overline{T}Q + TQ\overline{C} + TQS = \overline{T}Q + T(Q\overline{C} + QS)
 \end{aligned}$$

Status quo

Cambia valore di uscita



## Analisi della funzione logica sintetizzata



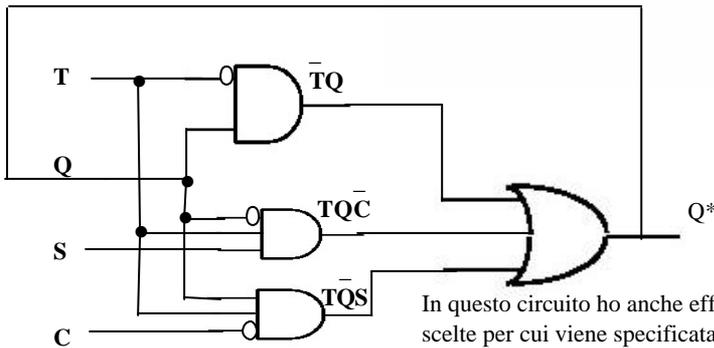
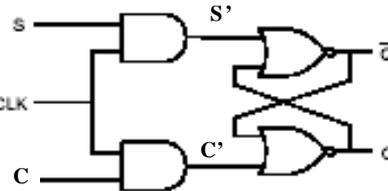
$$Q^* = \bar{T}Q + T(Q\bar{C} + \bar{C}S)$$

Status  
quo

If S=1 (C=0)  $Q^* = Q * !Q = 1$

If C=1 (S=0)  $Q^* = 0 + 0 = 0$

If C=S=1  $Q^* = 0 + !Q = !Q$   
cambio stato.



In questo circuito ho anche effettuato delle scelte per cui viene specificata l'uscita per S=C=1. E' un circuito più complesso.



## Sommario



Circuiti sequenziali

Latch asincroni

Latch sincroni SR

Latch sincroni D

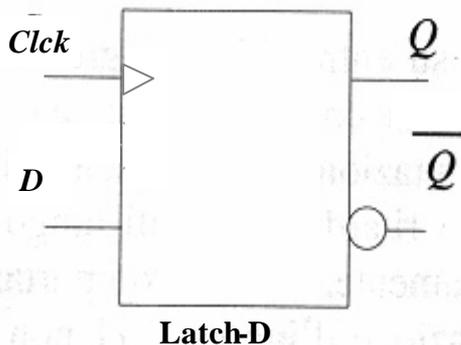


## Latch D sincrono

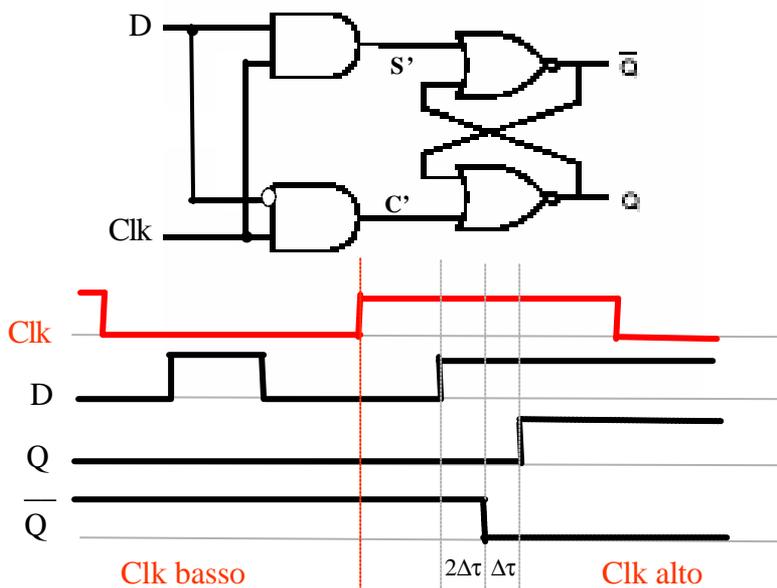


Memorizza il valore presente all'ingresso dati quando il clock è alto.

```
if clk = 1
  then
    Q* = D
  else
    Q* = Q
```



## La struttura del latch D





# Tabella delle transizioni

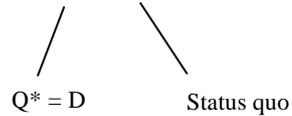


$$Q^* = f(T, Q, D)$$

TQ	D = 0	D = 1
00	0	0
01	1	1
11	0	1
10	0	1

La funzione logica corrispondente è:

$$Q^* = TD + \bar{T}Q$$



Q è l'uscita del latch: **stato presente**.

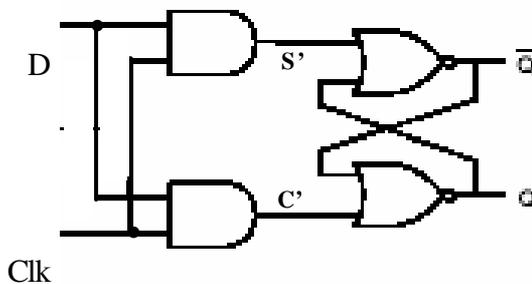
Q\* è il valore dell'uscita al tempo successivo: **stato prossimo**.



# Tabella delle verità



$$Q^* = f(T, Q, D)$$



$$Q^* = \bar{T}\bar{D}Q + \bar{T}DQ + T\bar{D}\bar{Q} + TDQ =$$

$$= \bar{T}Q + TD$$

Status quo (pointing to  $\bar{T}Q$ )  
 $Q^* = D$  (pointing to  $TD$ )

T	D	Q	Q*
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



# Sommario



Circuiti sequenziali

Latch asincroni

Latch sincroni SR

Latch sincroni D