



# Circuiti sequenziali e latch

Prof. Alberto Borghese  
Dipartimento di Scienze dell'Informazione

[borgnese@di.unimi.it](mailto:borgnese@di.unimi.it)

Università degli Studi di Milano

Riferimento Patterson: sezioni C.7 & C.8.



## Sommario

### Circuiti sequenziali

Latch asincroni

Latch sincroni SR

Latch sincroni D



## Perchè esiste il clock?



### Esempio:

$$C = A + B$$

$$E = D + C$$

Quando posso calcolare E con lo stesso sommatore?

“Cancello” davanti all’ingresso del sommatore prima della seconda somma.

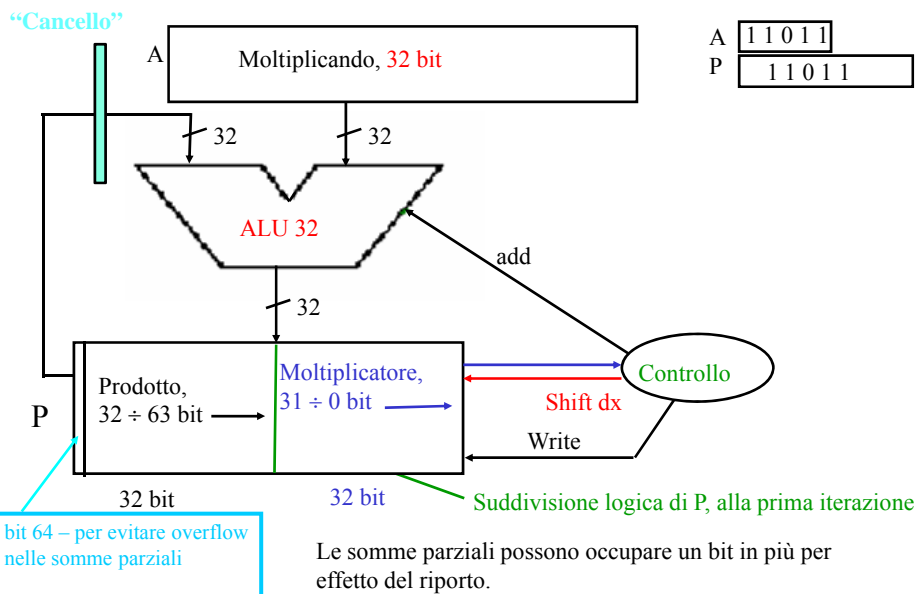
Ogni quanto tempo possiamo presentare gli ingressi al sommatore?

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

**Occore una sincronizzazione dell’attività del sommatore.**



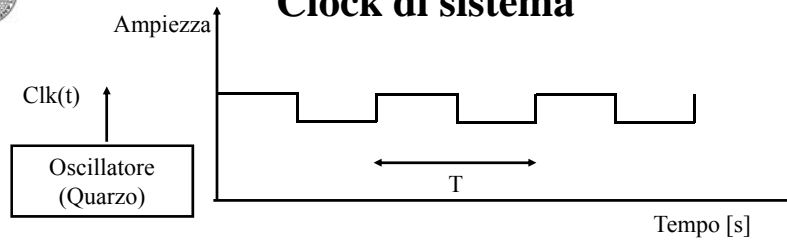
## Circuito finale – Moltiplicatore firmware







## Clock di sistema



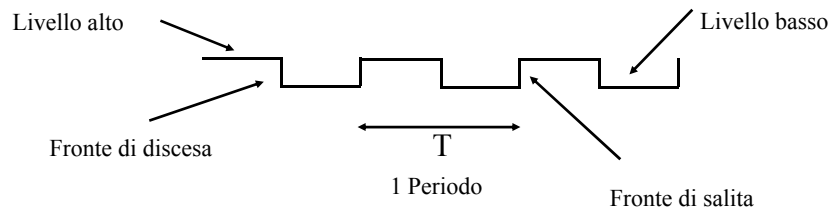
Frequenza: numero di cicli/s Si misura in Hertz, Hz.  
Periodo: tempo necessario a completare 1 ciclo Si misura in secondi, s.

$$T = 1/f$$

Tempo di salita e discesa trascurabile rispetto al tempo di commutazione delle porte logiche.



## 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.



## Circuiti sincroni

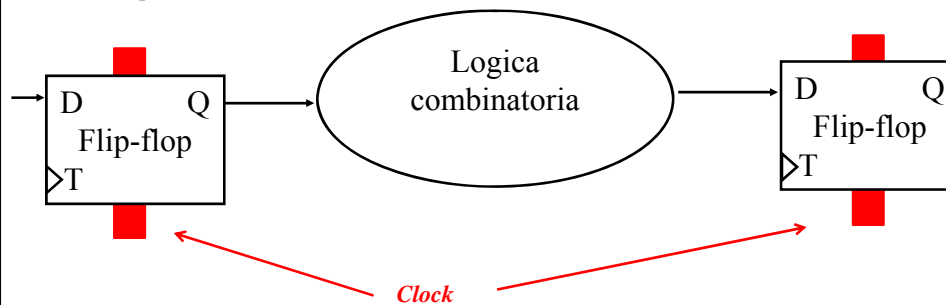


- **Architettura logica sincrona:**

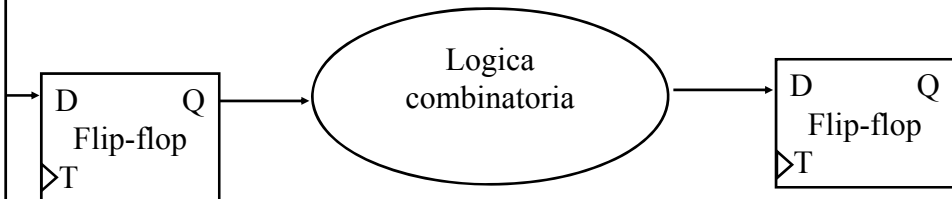
Le fasi di elaborazione sono scandite da un orologio comune a tutto il circuito (**clock**).

- 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**)

Esempio: CPU



## Struttura di un circuito sequenziale

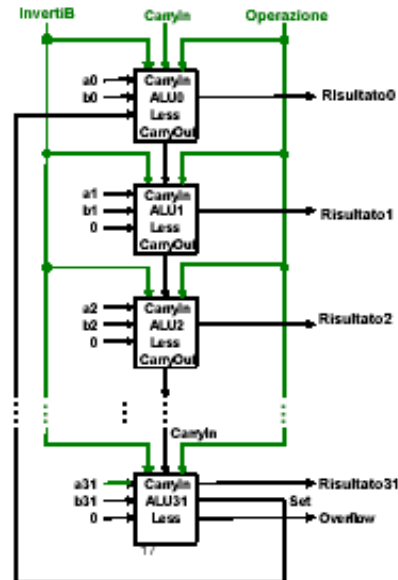




## Circuiti asincroni



- **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.  
Improponibile per circuiti con feed-back.Esempio: ALU.



## Architetture sequenziali



- I circuiti combinatori **non hanno memoria**. Gli output al tempo  $t$  dipendono unicamente dagli input al tempo  $t$  che provengono dall'esterno:  $y^{t+1} = f(u^{t+1})$
- Sono necessari circuiti con memoria, per consentire comportamenti diversi a seconda della situazione dell'architettura. Nella memoria viene memorizzato lo **stato** del sistema che riassume la storia passata e la sequenza delle situazioni precedenti.
- I circuiti che hanno elementi di memoria consentono di eseguire operazioni **sequenzialmente** (scandite dal clock (e.g. CPU) o meno (e.g. Distributore di bibite)).



## Sommario



Circuiti sequenziali

Latch asincroni

Latch sincroni SR

Latch sincroni D



## Bistabili: latch e flip-flop



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 output (0 o 1).

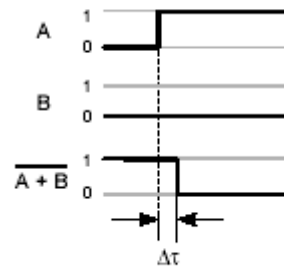
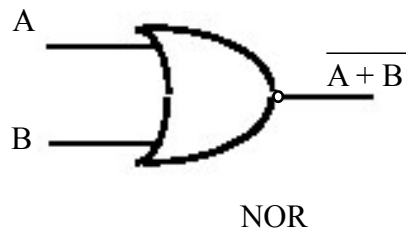
Il suo valore di uscita coincide con lo stato. L'uscita al tempo  $t$ , dipende dallo stato (uscita) al tempo  $t-1$  e dal valore presente agli input.

### **Tipi di bistabili:**

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



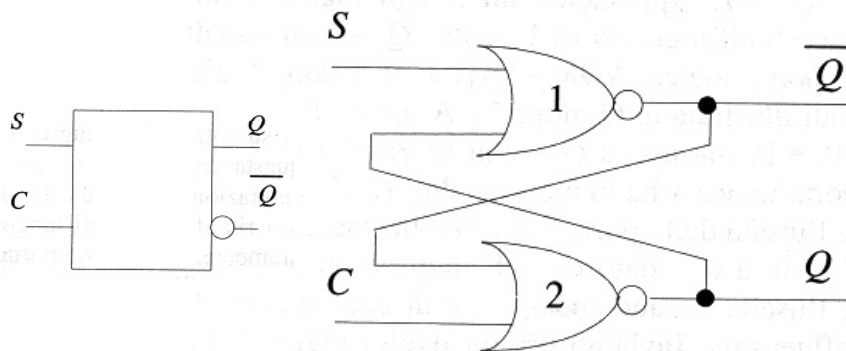
## Principio di funzionamento



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)

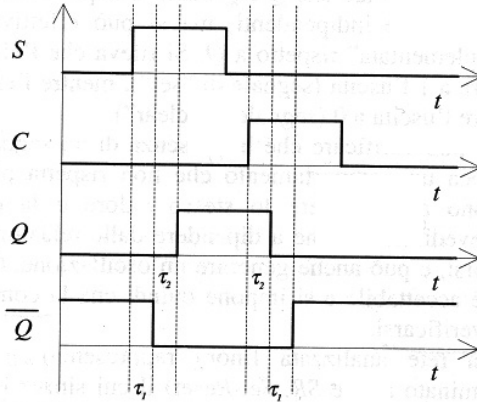
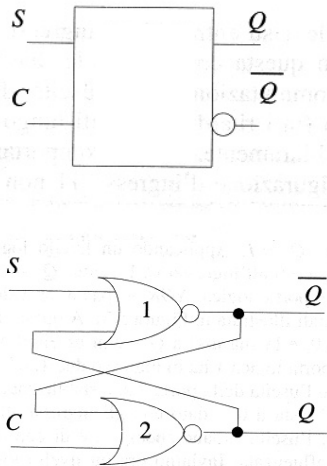


Una coppia di porte NOR retro-azionate può memorizzare un bit.





## Funzionamento del circuito SC



Cosa succede se  $S = 1$  e  $C = 0$ ,  
quando  $C \rightarrow 1$ ?

NB: Cammino critico = 2



## Tabella della verità di SC



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



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

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

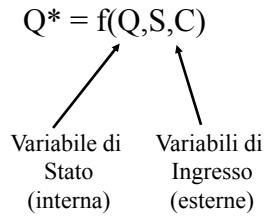
X è impostato = 0

Status Quo      Set

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



## Tabella delle transizioni



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

No change (Q\* = Q) Memory  
 Clear Reset Write 0  
 Set Write 1

Q è l'uscita del latch: **stato presente**,  $Q_t$   
 Q\* è il valore dell'uscita al tempo successivo: **stato prossimo**,  $Q_{t+1}$



## Tabella della verità di SC - II



Impostando X = 1 1, si ottiene:

$$Q^* = \bar{S}\bar{C}Q + S \quad (\text{per assorbimento}):$$

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

$\swarrow$  Status quo

SC = 11  $\rightarrow$  Q\* = 1  $\rightarrow$  Set

Impostando X = 0 1, si ottiene:

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

$$\overline{S\oplus C} Q + S\bar{C}$$

$\swarrow$  Status quo

SC = 11  $\rightarrow$  Q\* = 0  $\rightarrow$  Reset

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



## 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  $Q \rightarrow Q^*$ , qual'è la coppia di valori di ingresso che la determina?

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



## Sommario



Circuiti sequenziali

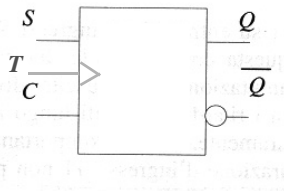
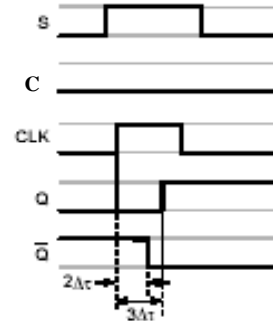
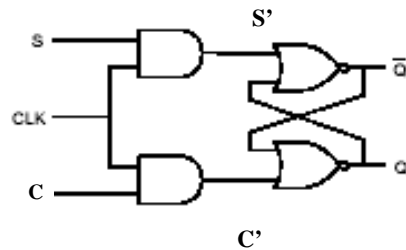
Latch asincroni

**Latch sincroni SR**

Latch sincroni D



## Il latch SC 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 (collegano l'altro ingresso dell'AND con l'uscita). Cancelli di abilitazione del latch.



## Tabella delle transizioni



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

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

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

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





## Sommario



Circuiti sequenziali

Latch asincroni

Latch sincroni SR

**Latch sincroni D**

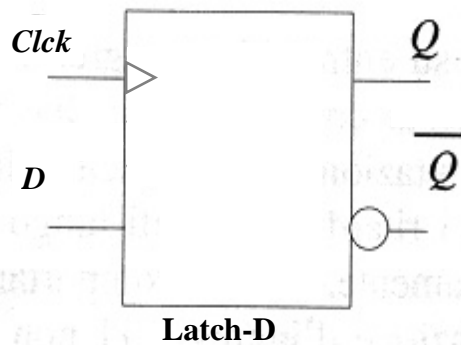


## Latch D sincrono



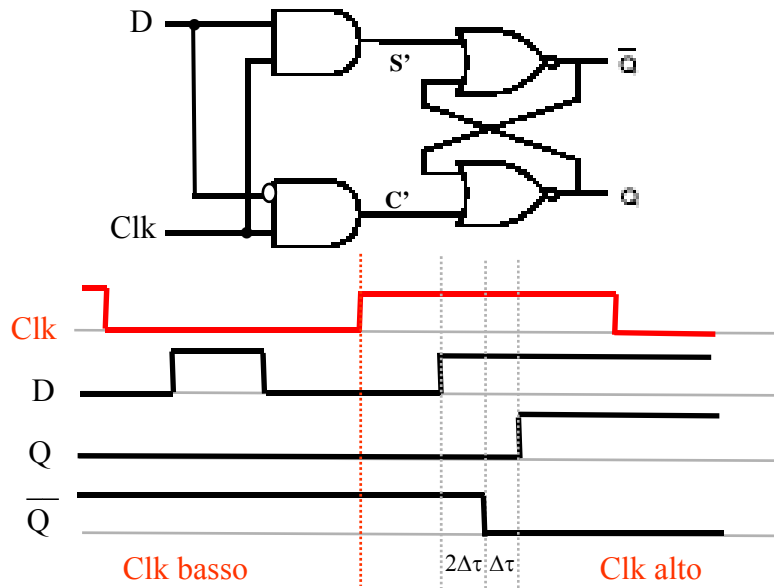
Memorizza il valore presente all'ingresso dati quando il clock è alto.

```
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:

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

$$Q^* = D$$

Status quo

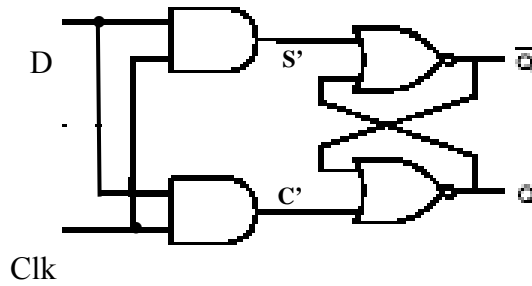
Q è l'uscita del latch: **stato presente**.

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



## Tabella delle verità

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



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

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

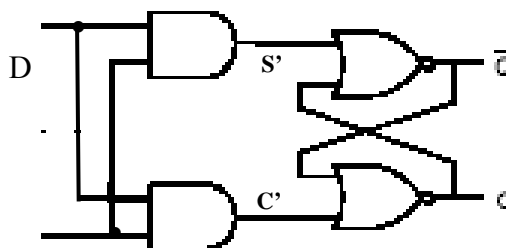
$$= \bar{T}Q + TD$$

Status  
quo

$$Q^* = D$$



## Elemento di memoria



T = segnale di scrittura – Write – attivo alto

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

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

Status Quo  
(memory)

$$Q^* = D$$

(write)

Se T = 1, scrivo D

Se T = 0, mantiene il dato in memoria





## Sommario



Circuiti sequenziali

Latch asincroni

Latch sincroni SR

Latch sincroni D