



# Latch sincroni e flip-flop

Prof. Alberto Borghese  
Dipartimento Informatica  
[borgnese@di.unimi.it](mailto:borgnese@di.unimi.it)

Università degli Studi di Milano

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



## Sommario

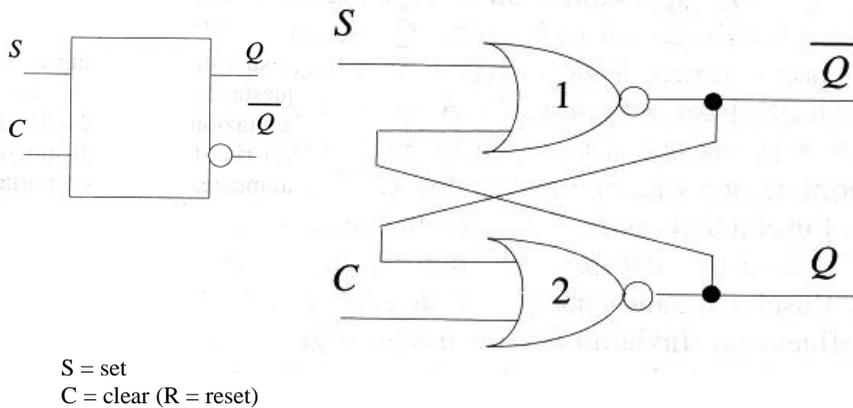
Latch sincroni SR

Latch sincroni D

Flip-flop



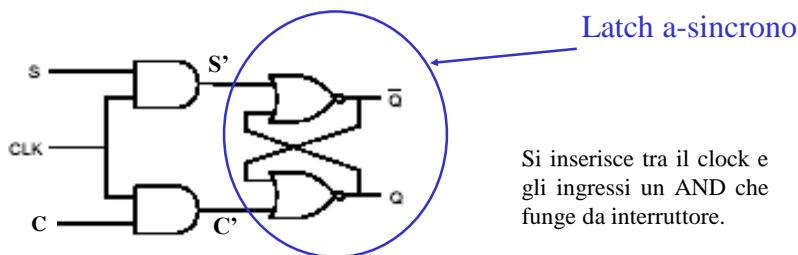
## Latch asincrono SC (o SR)



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



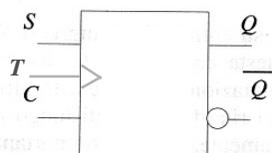
## Il latch SC sincrono



Si inserisce tra il clock e gli ingressi un AND che funge da interruttore.

If (CLK = H = 1) then  
 $S' = S; C' = C$

If (CLK = L = 0) then  
 $S' = C' = 0$

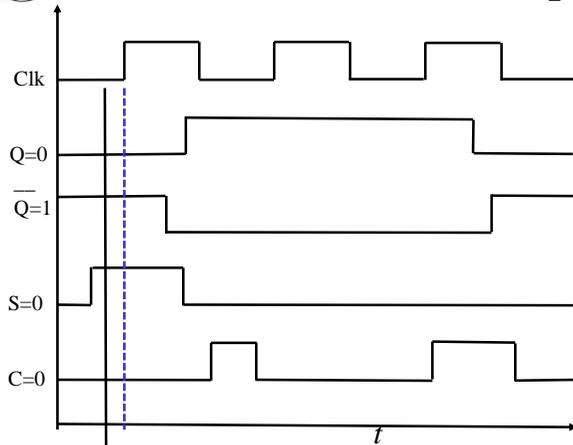


Solo quando il clock è alto i “cancelli” rappresentati dagli AND fanno passare gli input (collegano l’altro ingresso dell’AND con l’uscita). Cancelli di «abilitazione» del latch.

Latch asincrono, sincronizzato.

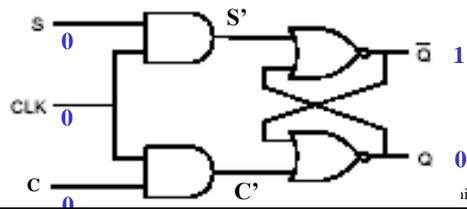


## Il latch sincrono nel tempo: clock basso



$Q_{t=0} = 0$

$\overline{Q}_{t=0} = 1$

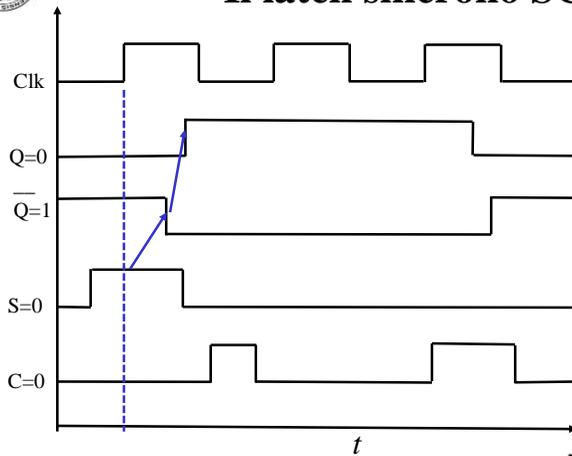


A.A. 2023-2024

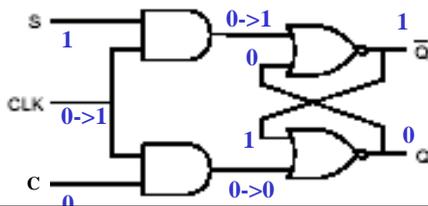
ii.it\



## Il latch sincrono SC - set

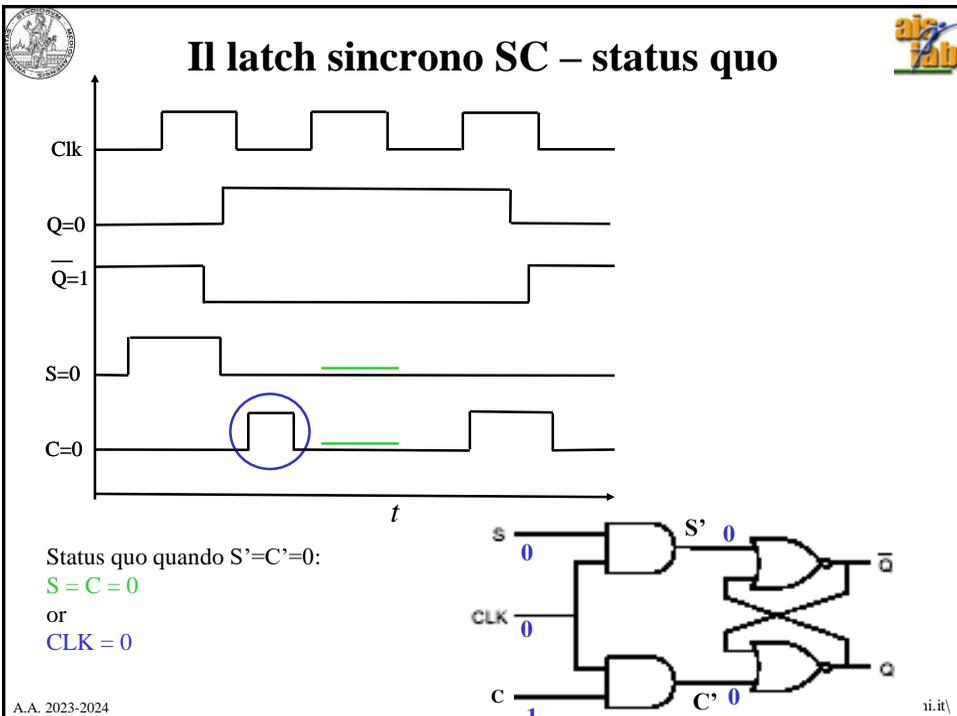
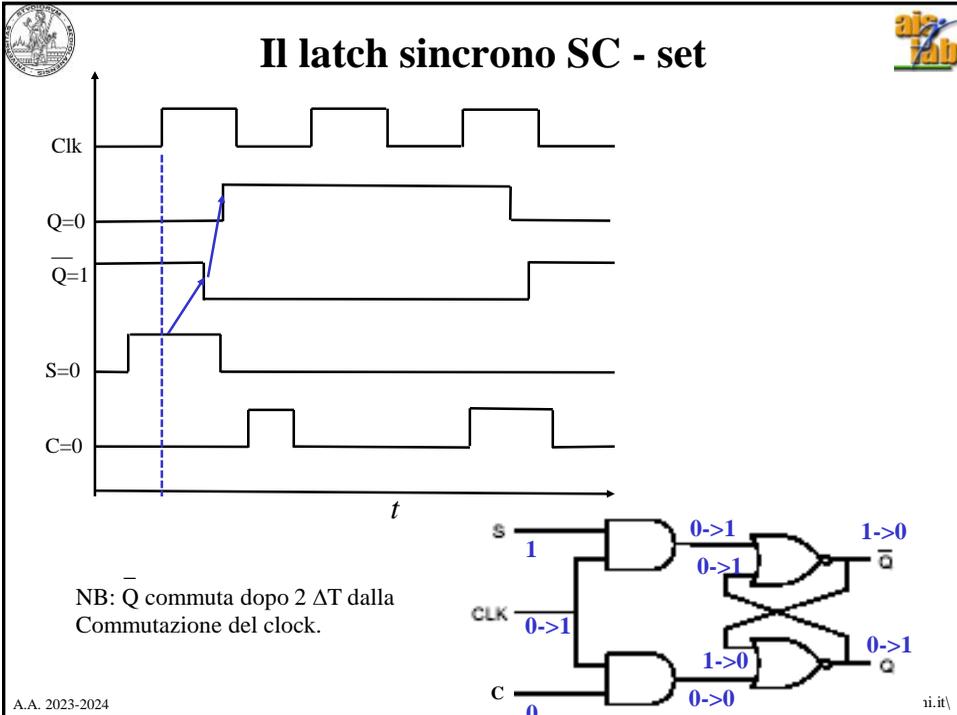


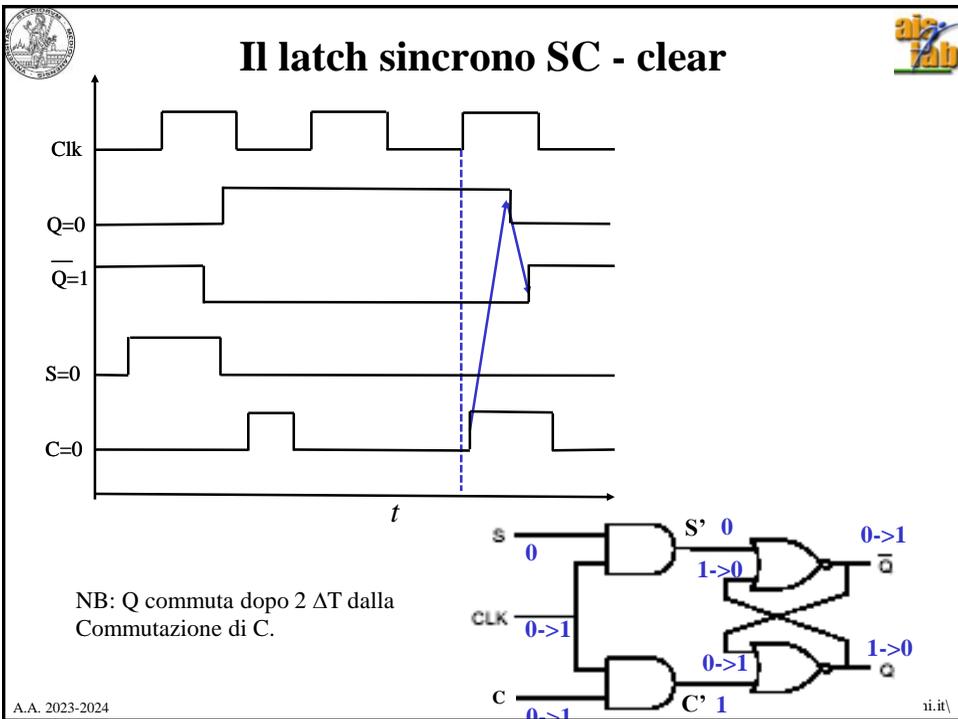
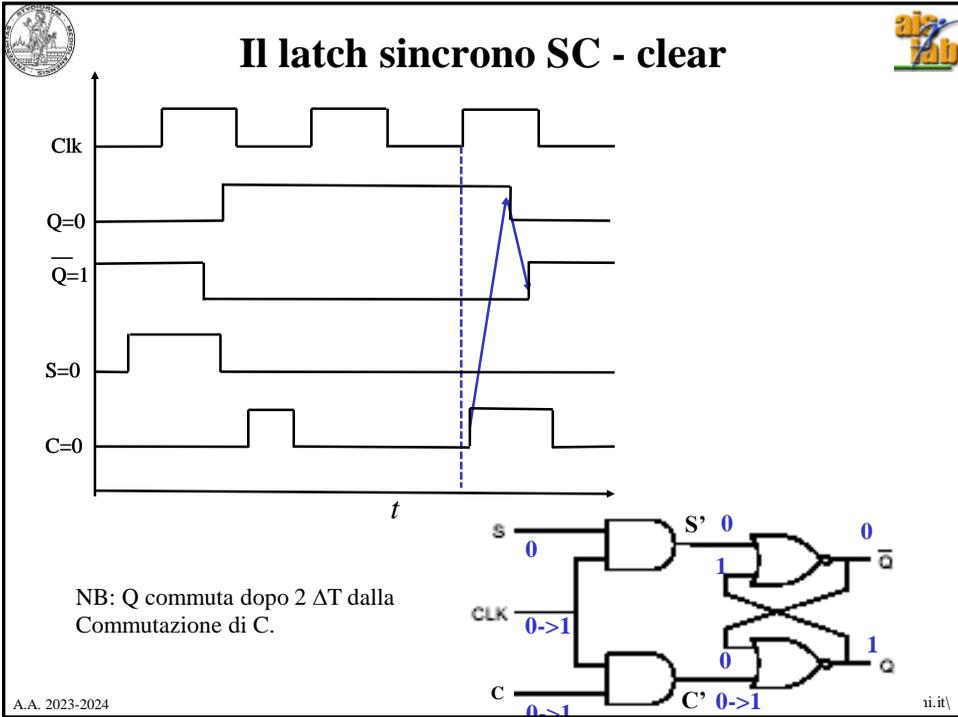
NB:  $Q$  commuta dopo  $2 \Delta T$  dalla Commutazione del clock.



A.A. 2023-2024

ii.it\

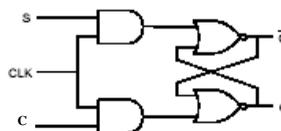






T	Q	S	C	Q*
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	X
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	X

## Tabella della verità e tabella di transizione



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

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

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



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

## Tabella della verità - I



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

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

$$Q^* = \overline{TQSC} + TQSC + \overline{TQSC} + TQSC + \overline{TQSC} + TQSC + \overline{TQSC} + TQSC$$

$$TQSC =$$

$$= \overline{TQ} + \overline{TQ}SC + TQ\overline{C} + TQSC =$$

$$= \overline{TQ} + TQ\overline{C} + TQSC =$$

$$\overline{TQ} + T(Q\overline{C} + SC)$$

Status quo (Memory)

Set

$$TSC = 1 \rightarrow Q^* = 0$$

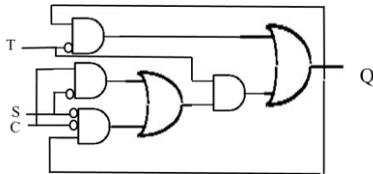


# Circuito SOP semplificata

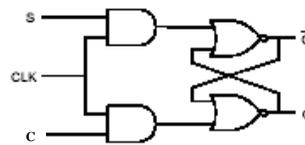


TQ	SC = 00	SC = 01	SC = 10	SC = 11
00	0	0	0	0
01	1	1	1	1
10	0	0	1	X=0
11	1	0	1	X=0

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



CC = 5 CO = 7



CC = 3 CO = 4

calcoliamo anche !Q



# Tabella della verità - II



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

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

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

$$Q^* = \bar{T}\bar{Q}SC + \bar{T}Q\bar{S}C + \bar{T}QSC + T\bar{Q}\bar{S}C + T\bar{Q}SC + TQ\bar{S}C + TQSC +$$

$$+ T\bar{Q}SC + TQ\bar{S}C + TQSC =$$

$$= \bar{T}\bar{Q}C + \bar{T}Q\bar{S}C + \bar{T}QSC + T\bar{Q}\bar{S}C + T\bar{Q}SC + TQ\bar{S}C + TQSC =$$

$$= \bar{T}Q + T\bar{Q}SC + TS = \bar{T}Q + T(QSC + S)$$

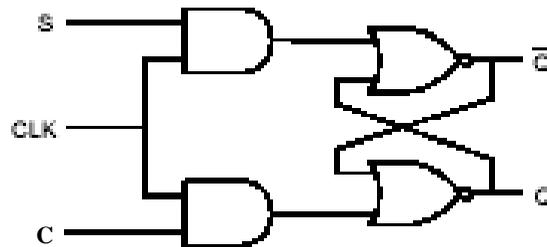
Status quo (Memory)

Cf. Latch asincrono

Set



## Latch sincrono



Latch attivo alto (commuta sul periodo alto del clock)



## Sommario

Latch sincroni SR

**Latch sincroni D**

Flip-flop

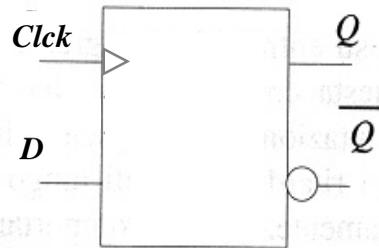
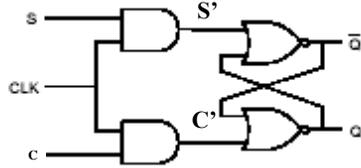


# Latch D sincrono

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

if (CLK = 1)  
then  
Q\* = D

If (CLK = 0)  
then  
Q\* = Q



Latch-D

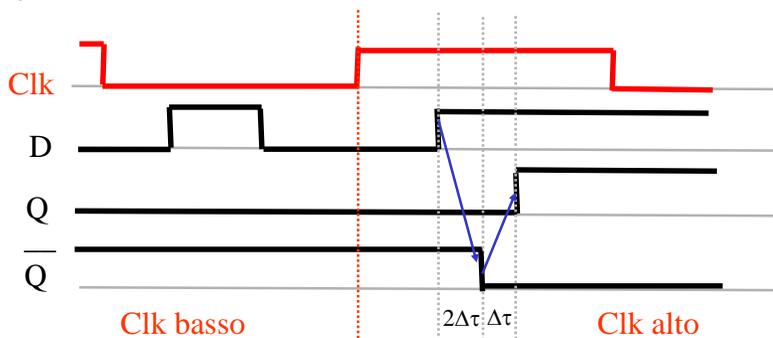
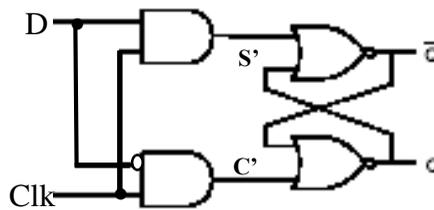
Latch trasparente sincrono



# La struttura del latch D

If (CLK==1)  
S' = D; C' = !D  
Q\* = D

If (CLK = 0)  
S' = C' = 0  
Q\* = Q





## Tabella delle transizioni

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

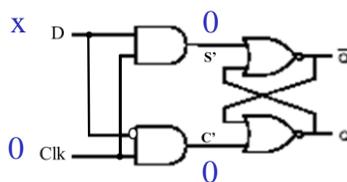
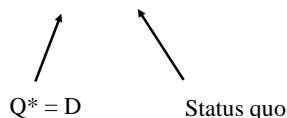
TQ	D = 0	D = 1
00	0	0
01	1	1
11	0	1
10	0	1

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

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

La funzione logica corrispondente è:

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



$$Q^* = Q$$



## Tabella delle transizioni

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

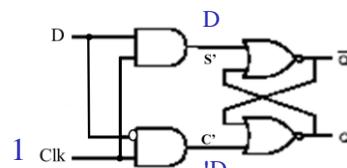
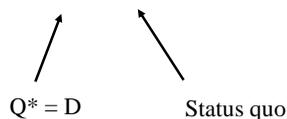
TQ	D = 0	D = 1
00	0	0
01	1	1
11	0	1
10	0	1

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

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

La funzione logica corrispondente è:

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



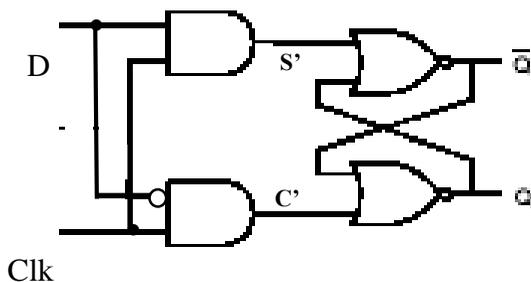
$$Q^* = D$$

Come mai qui non si verifica la situazione  $S'=C'=1$ ?



# Tabella della 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

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

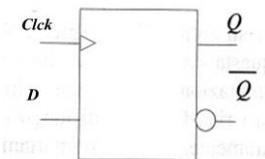
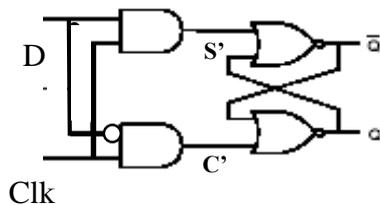
$$= \bar{T}Q + TD$$

Status quo       $Q^* = D$



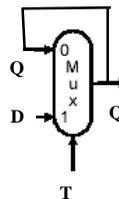
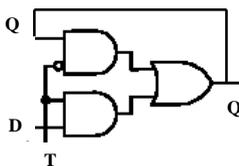
# Ottimizzazione del circuito

Complessità 4  
Cammino critico 3



Clk come interruttore che pilota un mux:

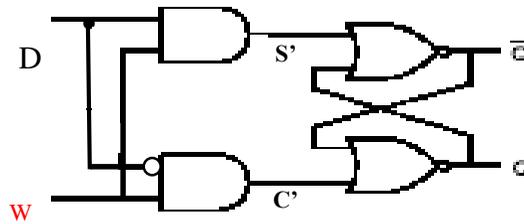
$$Q^* = \bar{T}Q + TD$$



Complessità 3  
Cammino critico 2



# Elemento di memoria



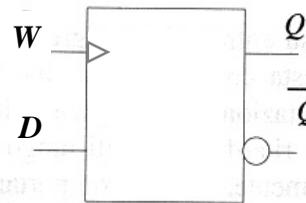
T = segnale di scrittura -> segnale Write – attivo alto

$Q^* = Q$   
 Status Quo  
 (memory)

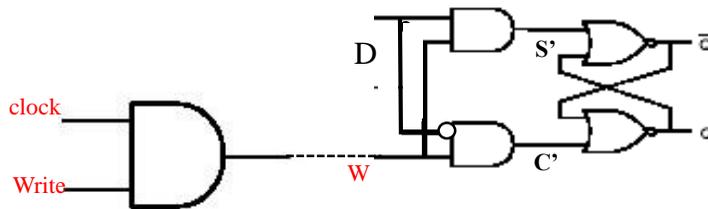
$$Q^* = \bar{T}Q + TD$$

$Q^* = D$   
 (write)

Se  $W = 1$ , scrivo D  
 Se  $W = 0$ , mantiene il  
 in memoria



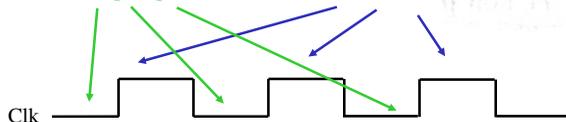
# Elemento di memoria



Write può essere sincronizzato dal clock

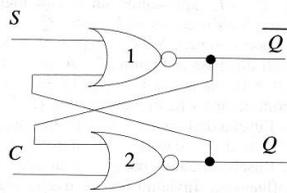
$Q^* = \bar{W}Q + WD$   
 $Q^* = Q$  Status Quo (memory)  
 $Q^* = D$  (write)

if (w=1)  $Q^* = D$

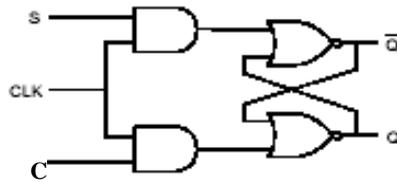




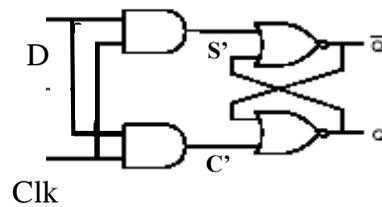
## I latch



Operazioni di Set/Reset



Operazioni di Set/Reset sincronizzate



Elemento di memoria



## Sommario

Latch sincroni SR

Latch sincroni D

**Flip-flop**



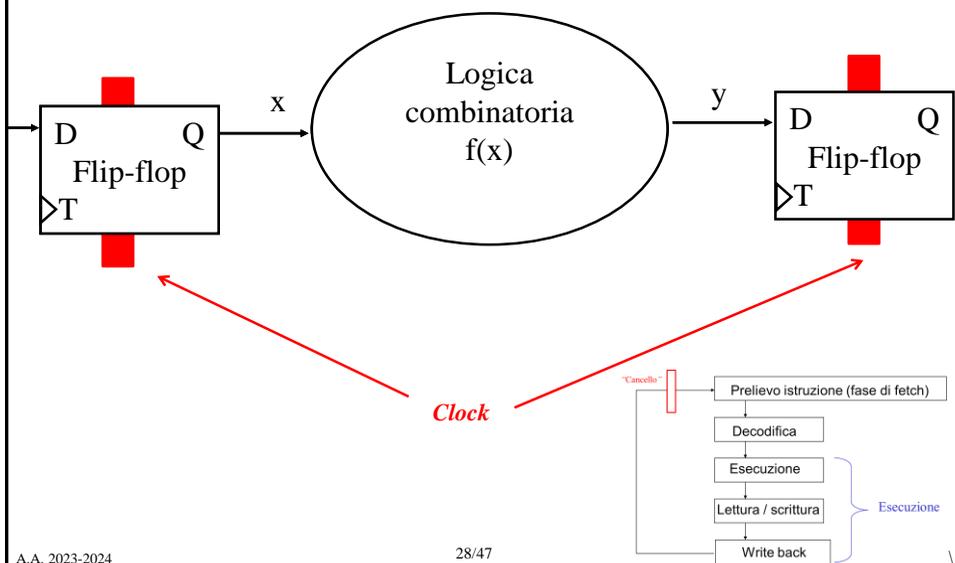
## I bistabili



- Elementi di memoria (latch)
  - Sincroni
  - A-sincroni
- “Cancelli” (flip-flop)

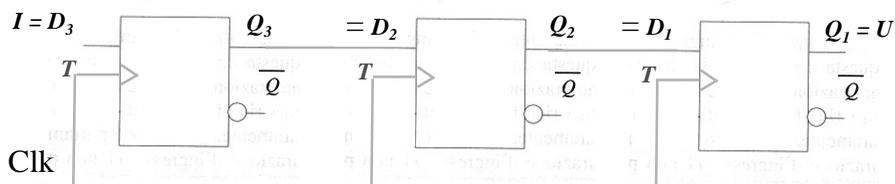


## Struttura di un circuito sequenziale



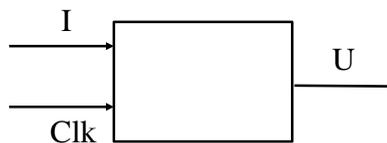


## Shift register

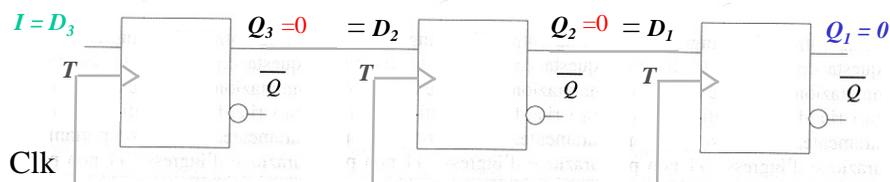


Registro a scorrimento (shift register o barrel shifter).

- Un unico ingresso I e un'unica uscita U.
- In presenza di un segnale attivo (clock alto), il contenuto viene spostato verso dx di **una posizione** (e.g. operazione di shift).
- Il valore contenuto nell'elemento più a dx dove va?
- Qual'è il problema con l'utilizzo dei latch sincroni?



## Shift register con i latch (i problemi)

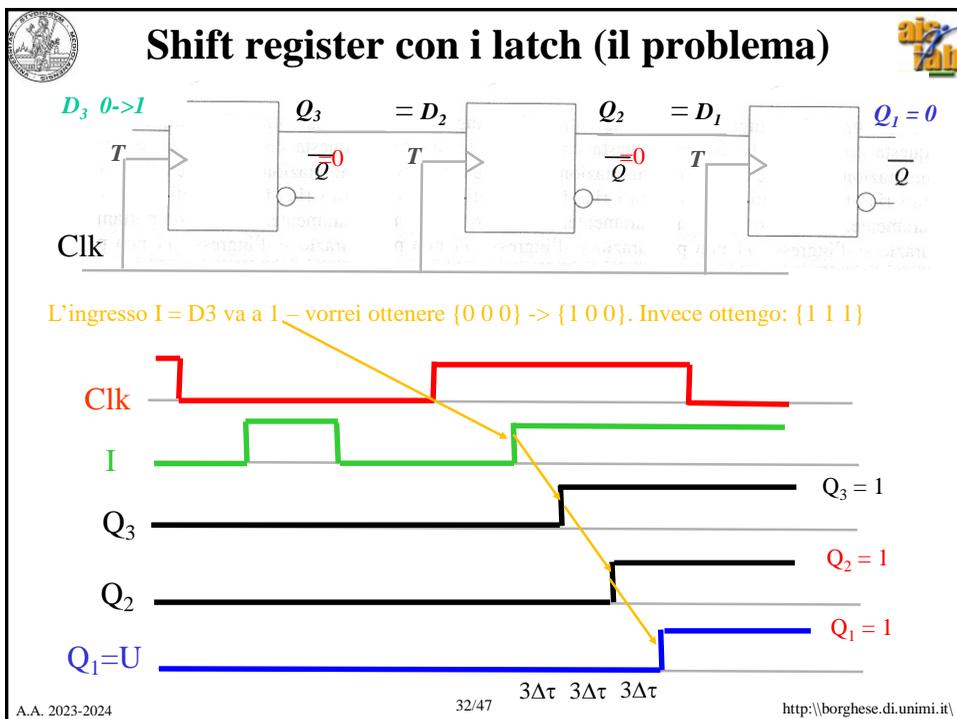
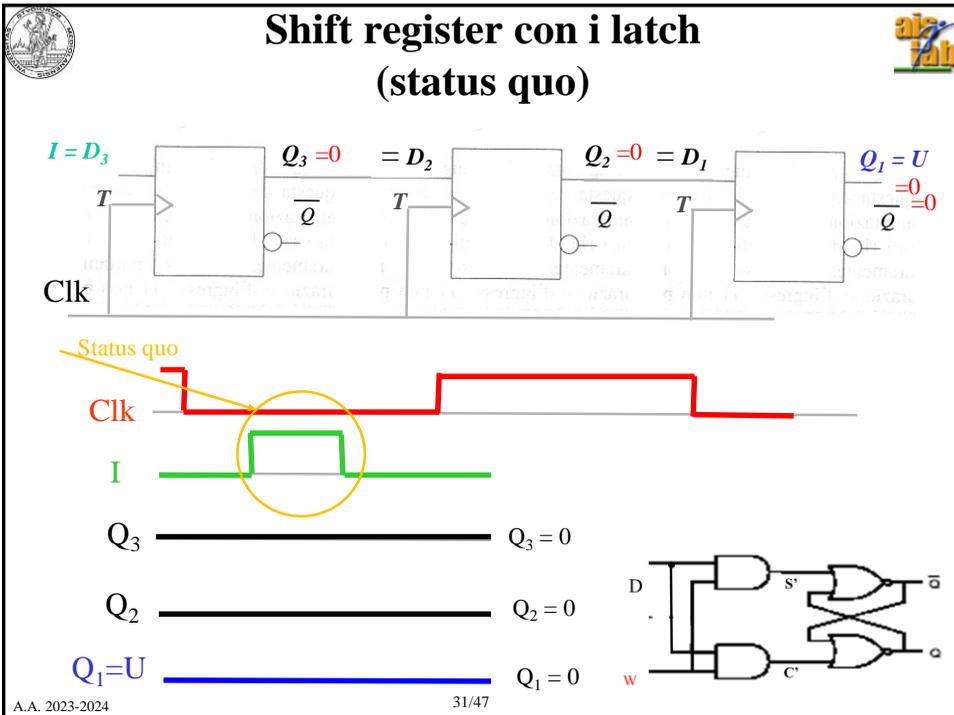


Fotografiamo la situazione iniziale:

- Clock basso
- $Q_3 = Q_2 = Q_1 = 0$
- $D_3 = 0$

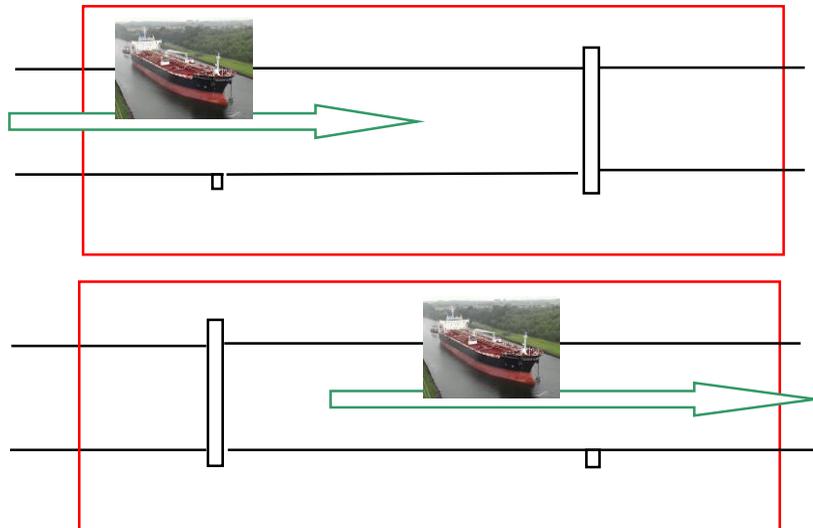
Shift di 1 posizione:

- $D_2 = Q_3$
- $D_1 = Q_2$





## Dispositivo di sincronizzazione

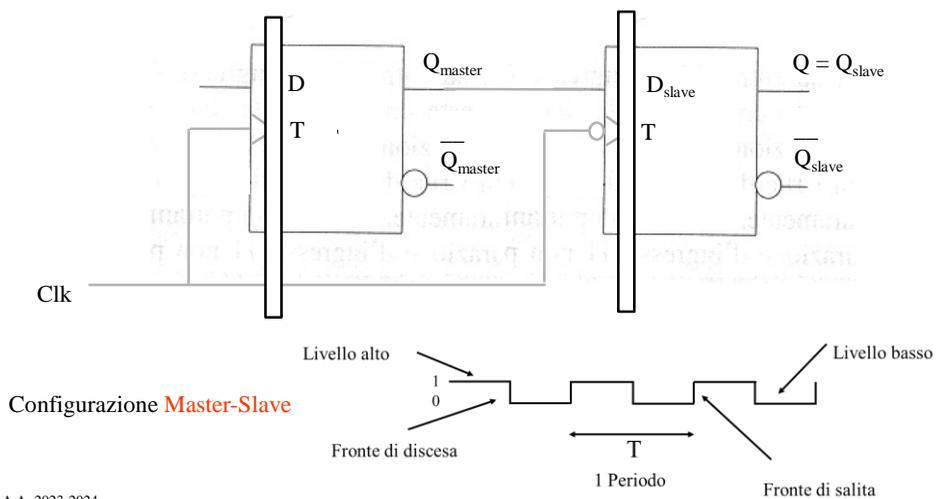


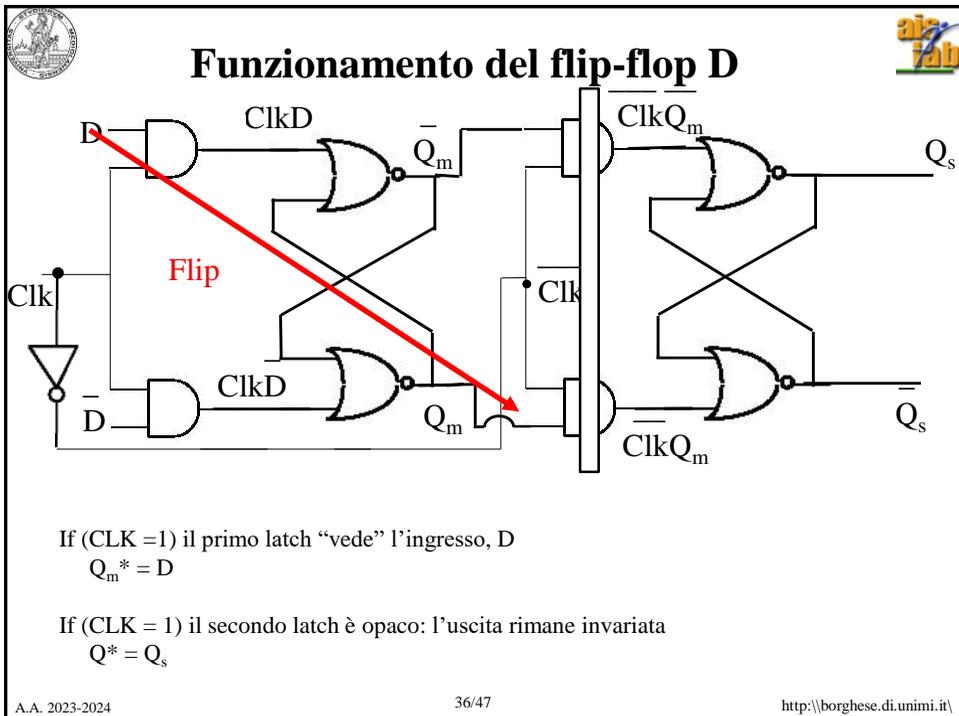
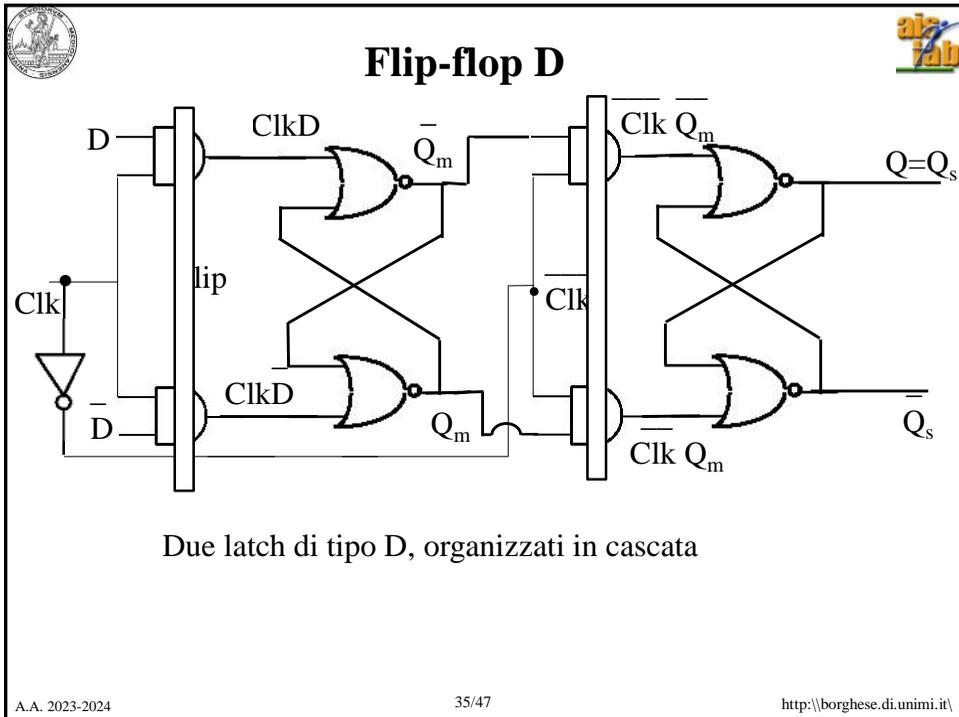
Sistema di "chiuse"

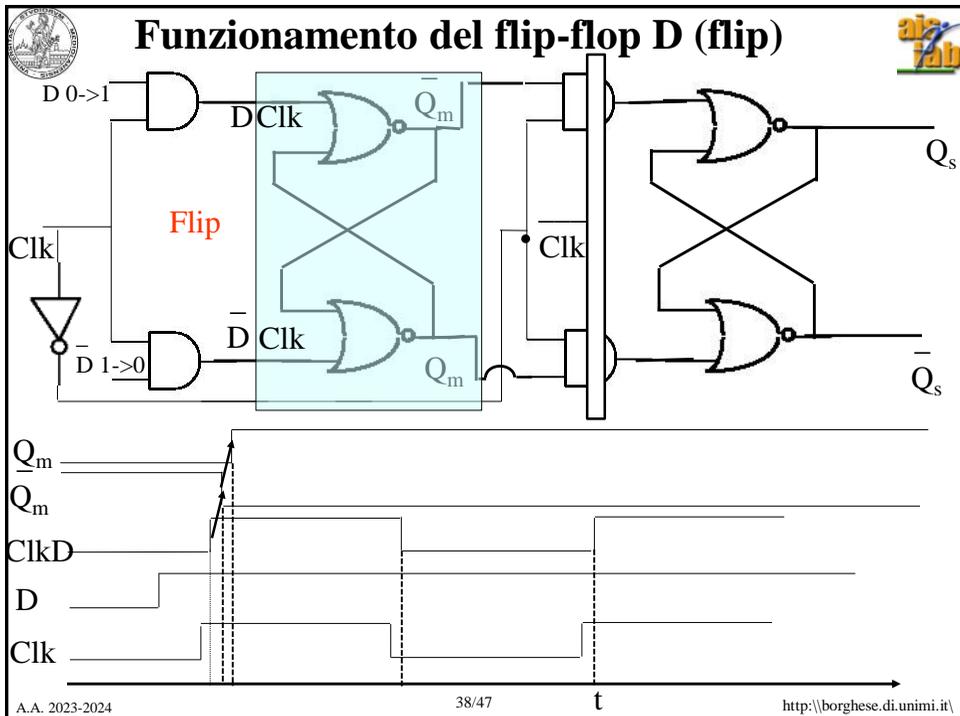
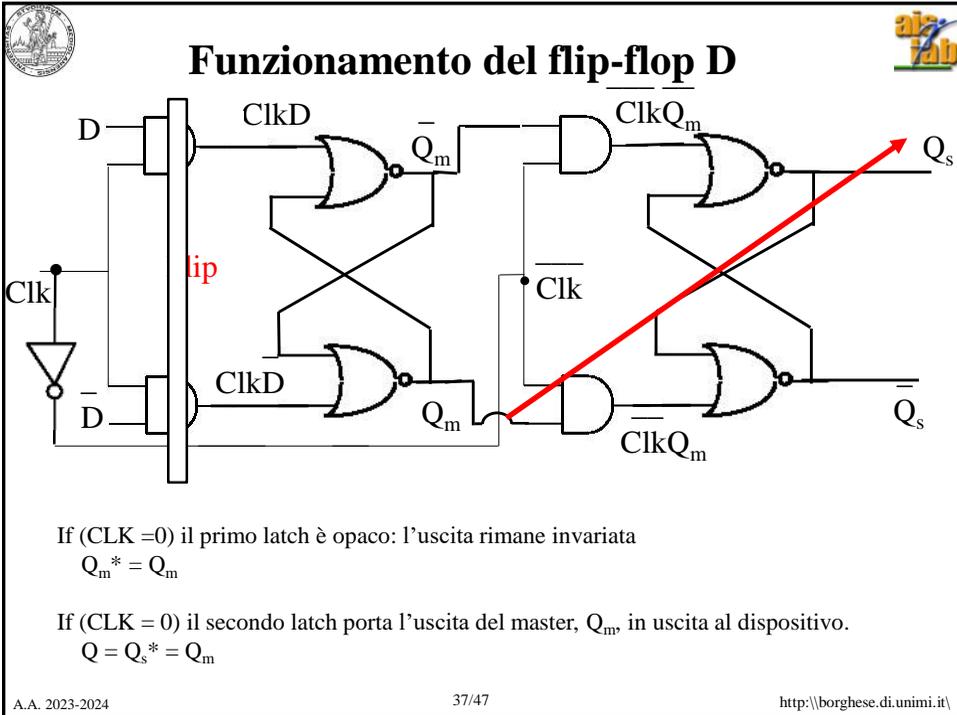


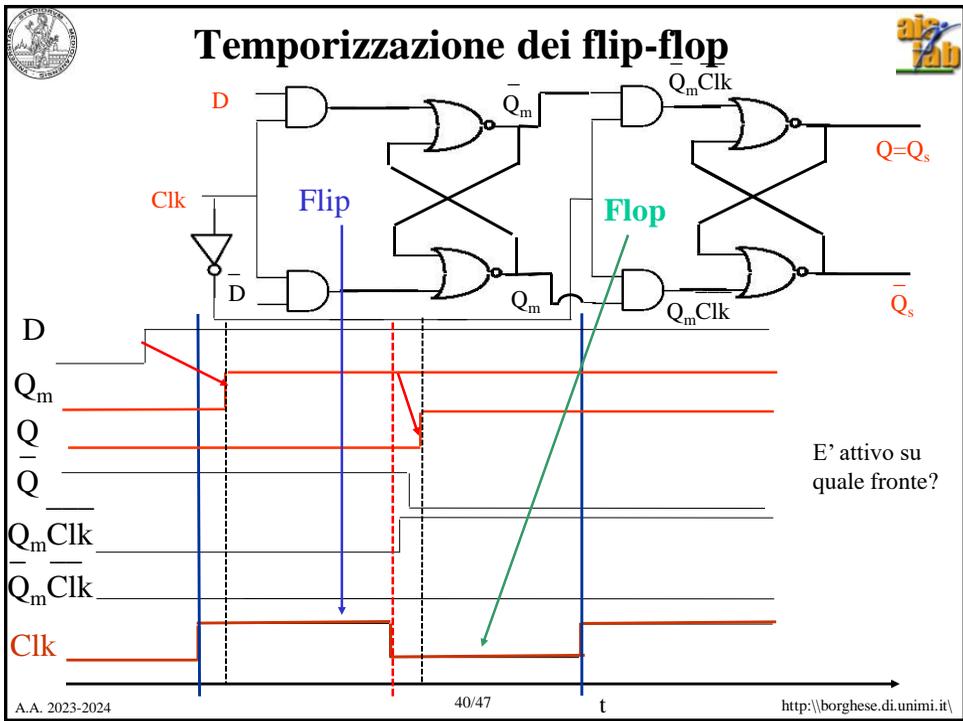
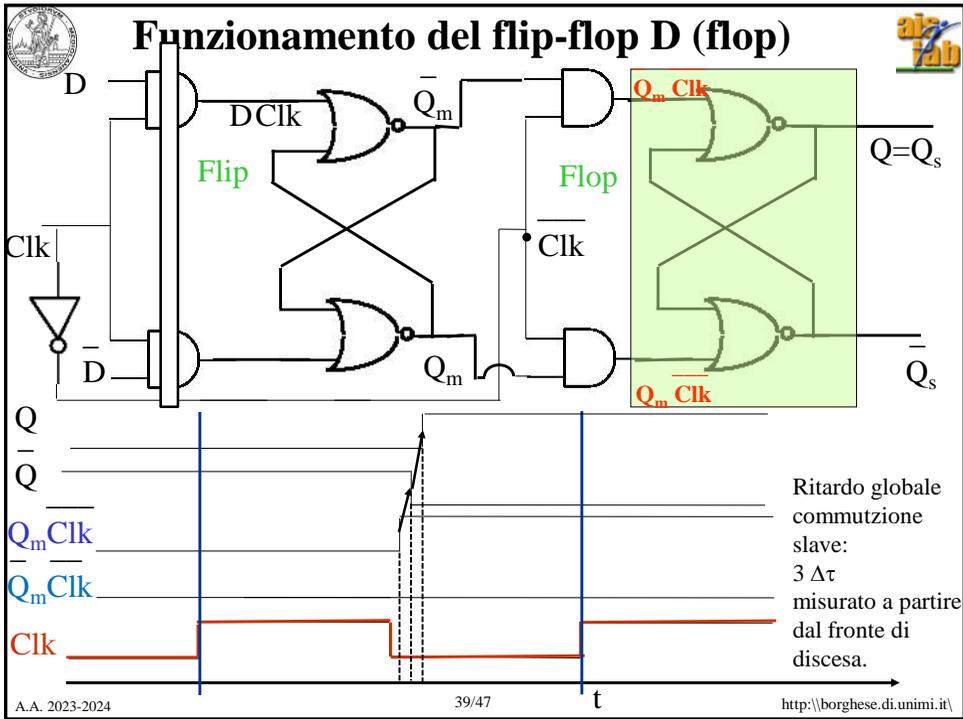
## Flip-flop

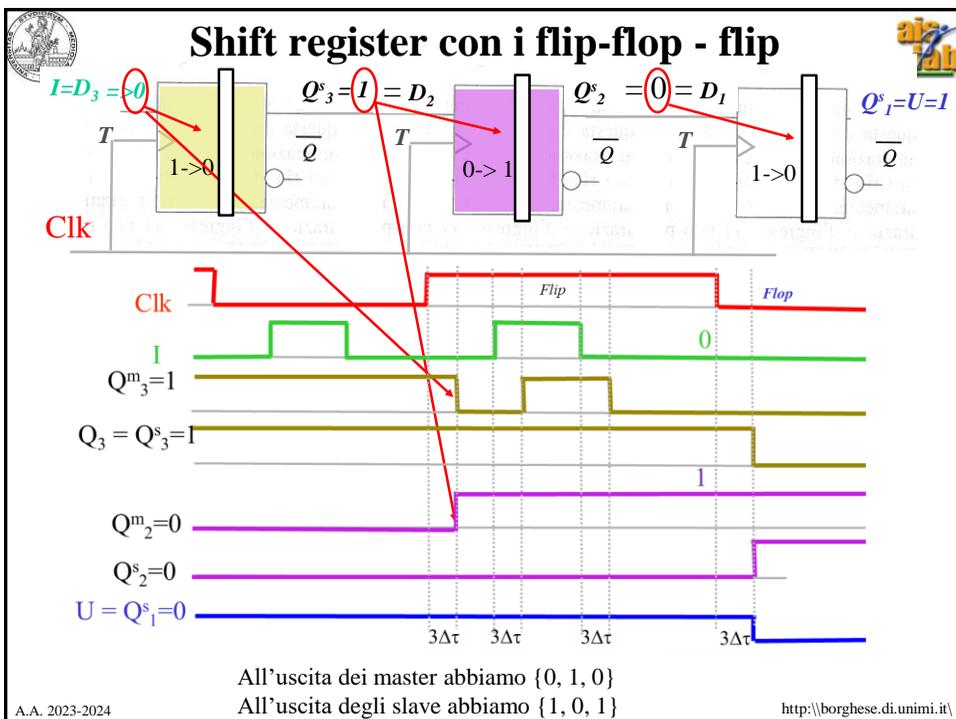
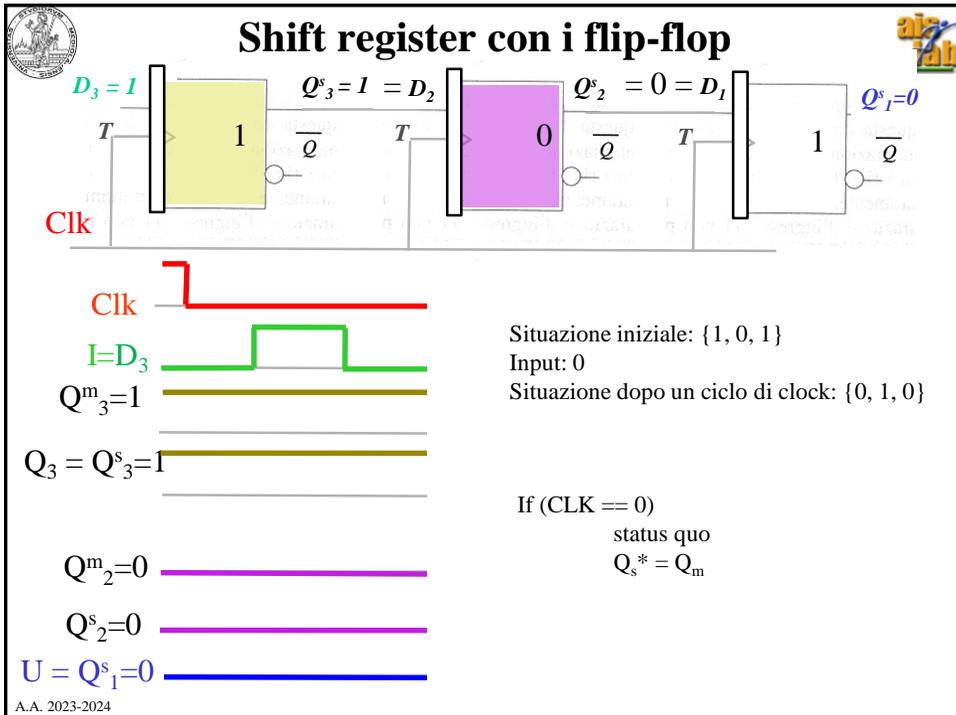
Dispositivi attivi sul fronte (di salita o discesa) del clock (edge sensitive): il loro stato (uscita) può commutare solo in corrispondenza della transizione alto->basso o basso->alto del clock.

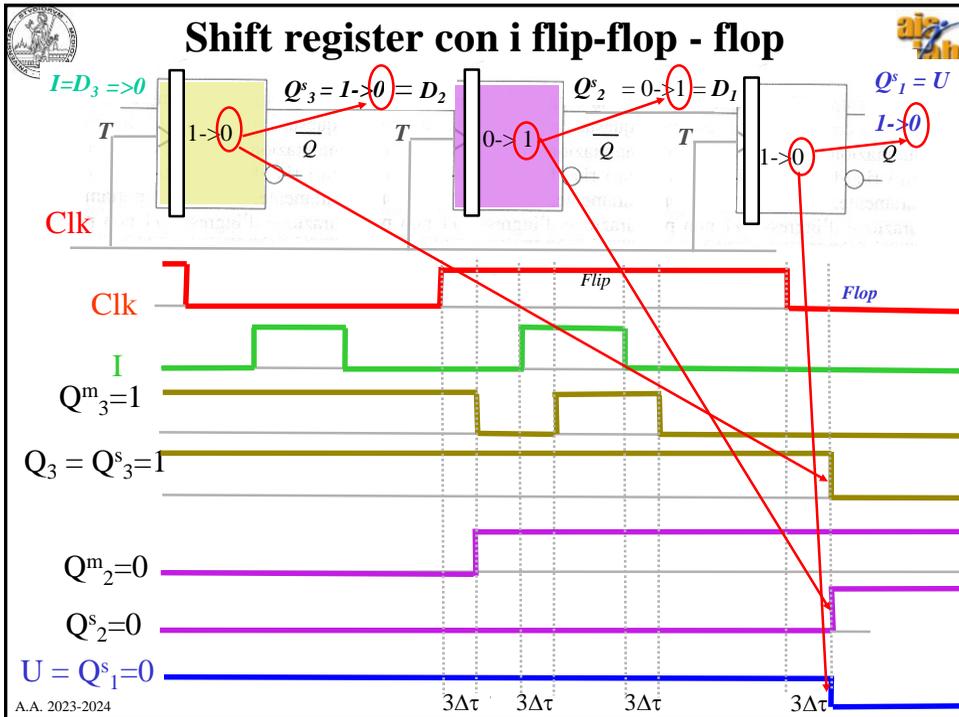












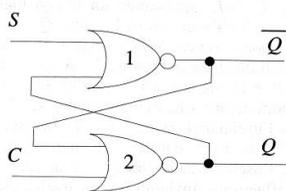
## Configurazione master-slave

- Semi-periodo di clock alto:
  - Il master è trasparente: uscita del latch master = ingresso.
  - Lo slave è “opaco” -> mantiene l’uscita.
  - Lo slave è “disaccoppiato” dal latch master.
    - Master e slave possono avere uscita diversa
- Semi-periodo di clock basso:
  - Il master è opaco: l’uscita del latch master si mantiene.
  - Il master è “disaccoppiato” dall’ingresso esterno.
  - Lo slave è trasparente: uscita del latch slave = uscita del latch master.
    - Master e slave hanno la stessa uscita

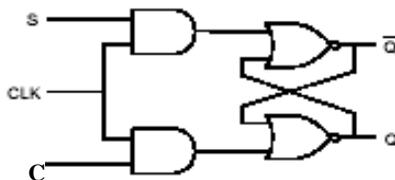
A.A. 2023-2024 44/47 <http://borghese.di.unimi.it/>



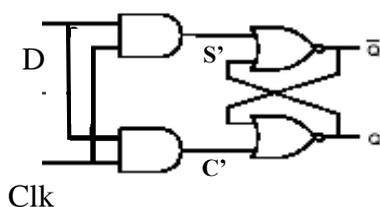
# I bistabili



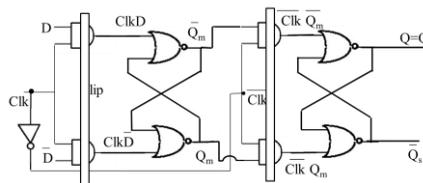
Latch SC asincrono (Set/Reset)



Latch SC sincrono (Set/Reset sincronizzato)



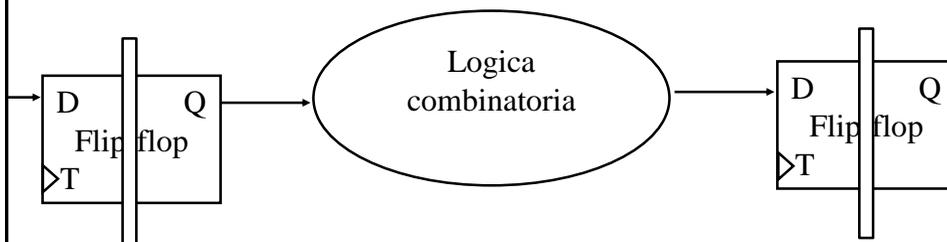
Latch D sincrono (Elemento di memoria)



Flip-flop di tipo D («cancello»)



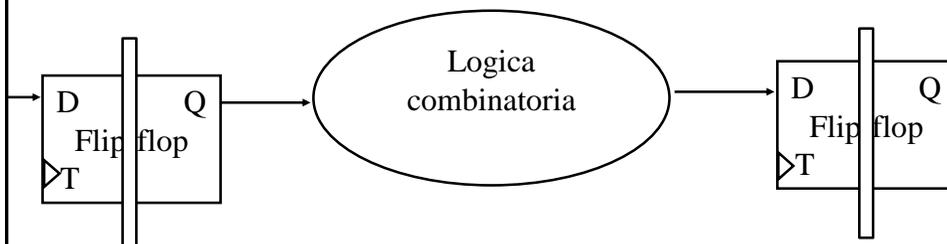
# Struttura di un circuito sequenziale



Pone dei problemi di sincronizzazione: la logica combinatoria deve terminare la commutazione in tempo utile.



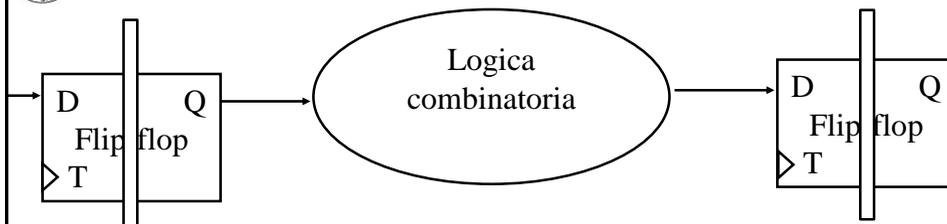
## Temporizzazione di un circuito sequenziale



- La logica ha tempo sufficiente per completare la commutazione.
- Il periodo di clock è tale, per cui la commutazione del clock avviene dopo che la logica combinatoria ha terminato tutte le commutazioni.
- Il tempo necessario alla logica combinatoria per commutare è  $\leq$  tempo associato al cammino critico.
- Il clock arriva contemporaneamente a tutti i dispositivi sincronizzati.

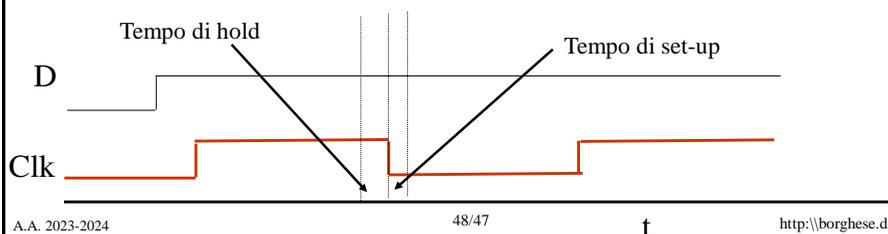


## Temporizzazione: problemi



L'input D deve essere stabile intorno alla commutazione del clock:

- Tempo di hold ( $t_h$ )**: è il tempo minimo per cui deve rimanere stabile l'input D prima del fronte di clock (tempo di attraversamento delle porte del master).
- Tempo di set-up ( $t_s$ )**: è il tempo minimo per cui deve rimanere stabile l'input D dopo il fronte di clock (tempo di attraversamento delle porte dello slave).



# Temporizzazione: Come si dimensiona il clock

$T = k * (t_p + t_h + t_s + t_w)$  k è margine di sicurezza  $k > 1$

**Tempo di propagazione:** è il tempo necessario per propagare il segnale nella logica combinatoria ( $t_p$ ). Il massimo tempo è rappresentato dal cammino critico.

**Tempo di skew:** ritardo massimo del clock ( $t_w$ ).

Tempo logica combinatoria ( $t_p$ )

A.A. 2023-2024 49/47 http://\borghese.di.unimi.it\

# Sommario

- Latch sincroni SR
- Latch sincroni D
- Flip-flop

A.A. 2023-2024 50/47 http://\borghese.di.unimi.it\