



# 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



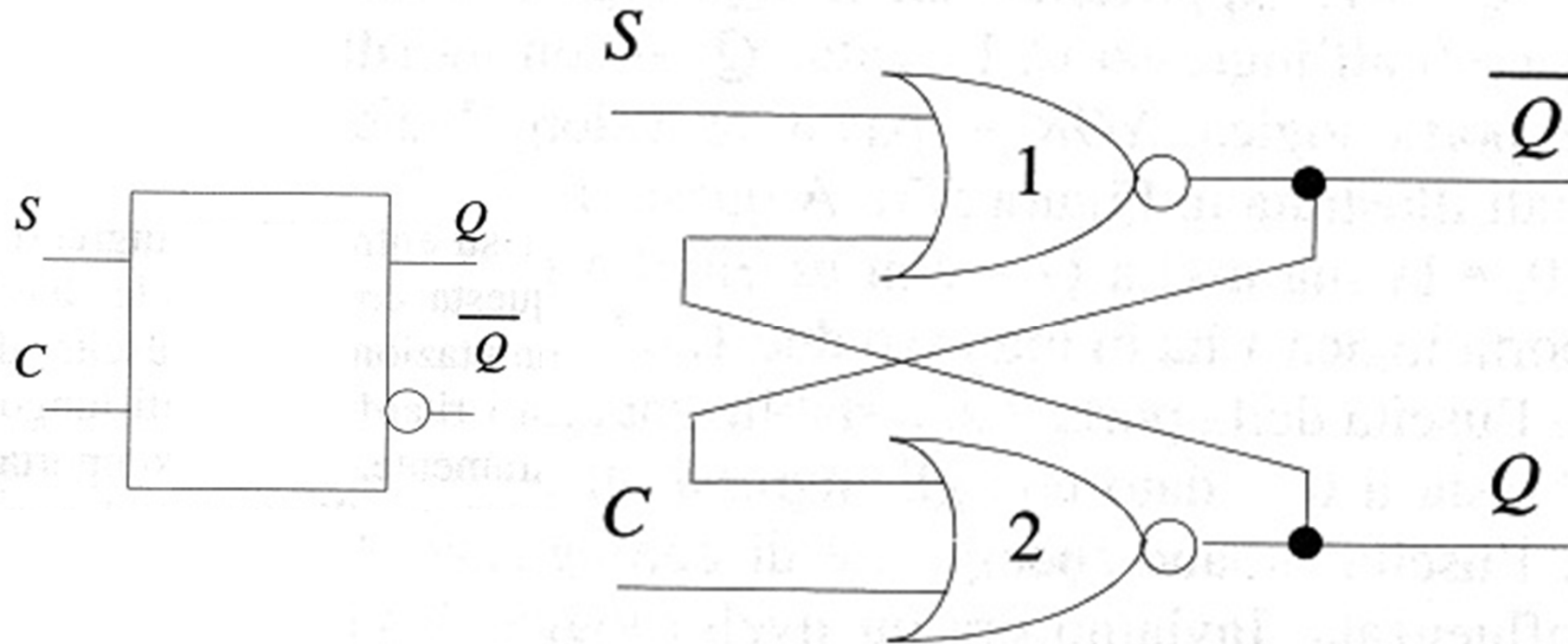
Latch sincroni SR

Latch sincroni D

Flip-flop



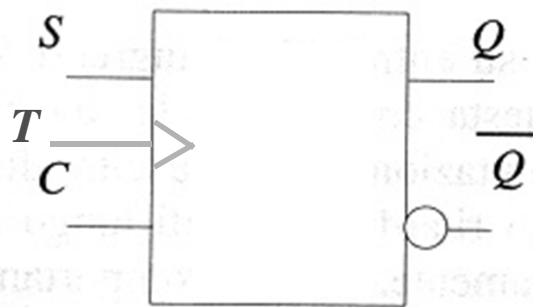
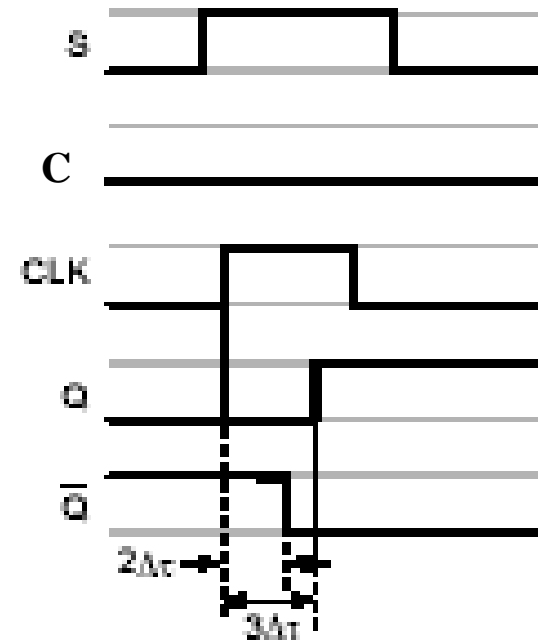
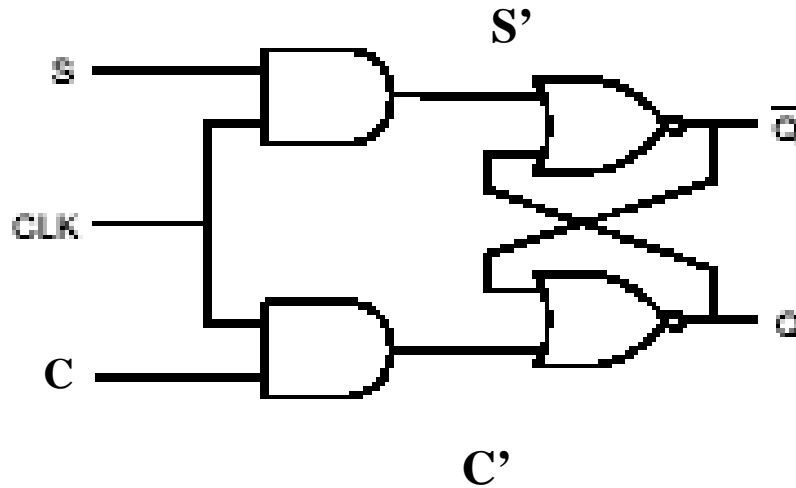
# Latch asincrono SC (o SR)



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



# Il latch SC sincrono



Si inserisce 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.



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

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



# Tabella delle transizioni

$$Q^* = f(S, C, Q, \mathbf{T})$$

<b>TQ</b>	<b>SC = 00</b>	<b>SC = 01</b>	<b>SC = 10</b>	<b>SC = 11</b>
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**.



# Tabella della verità - I

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

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

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

$$Q^* = \overline{T} \overline{Q} \overline{S} \overline{C} + \overline{T} \overline{Q} S \overline{C} + \overline{T} \overline{Q} S C + \overline{T} Q \overline{S} \overline{C} + \overline{T} Q S \overline{C} + \overline{T} Q S C +$$

$$T \overline{Q} \overline{S} C +$$

$$= \overline{T} \overline{Q} C + \overline{T} \overline{Q} S C + \overline{T} \overline{Q} S C + T \overline{Q} S C =$$

$$= \overline{T} \overline{Q} + \overline{T} \overline{Q} S C + T \overline{Q} S C =$$

$$\overline{T} \overline{Q} + T(Q S C + S C)$$

Status quo (Memory) : Cambia valore di uscita



# 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
11	1	0	1	X

$$Q^* = \overline{T}Q\overline{S}C + \overline{T}QSC + \overline{T}Q\overline{S}\overline{C} + \overline{T}QSC + \overline{T}Q\overline{S}C + \overline{T}QSC + \overline{T}QSC + \overline{T}QSC =$$

$$= \overline{T}Q\overline{C} + \overline{T}QSC + \overline{T}Q\overline{C} + \overline{T}QS + \overline{T}QS =$$

$$= \overline{T}Q + \overline{T}QSC + \overline{T}QS + \overline{T}QS = \overline{T}Q + T(QSC + S)$$

Status quo  
(Memory)

Cambia valore  
di uscita

Cf. Latch  
asincrono





# Sommario



Latch sincroni SR

Latch sincroni D

Flip-flop

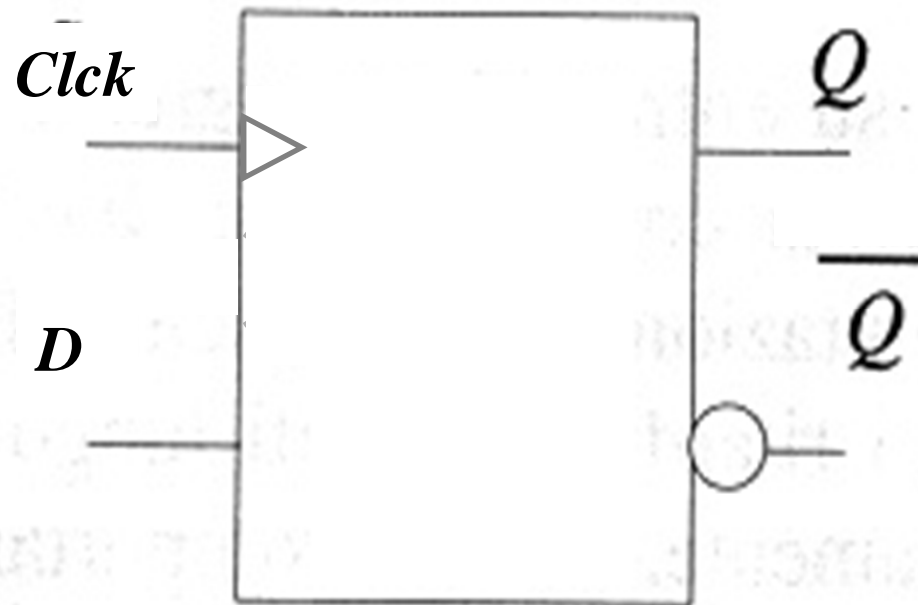


# Latch D sincrono



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

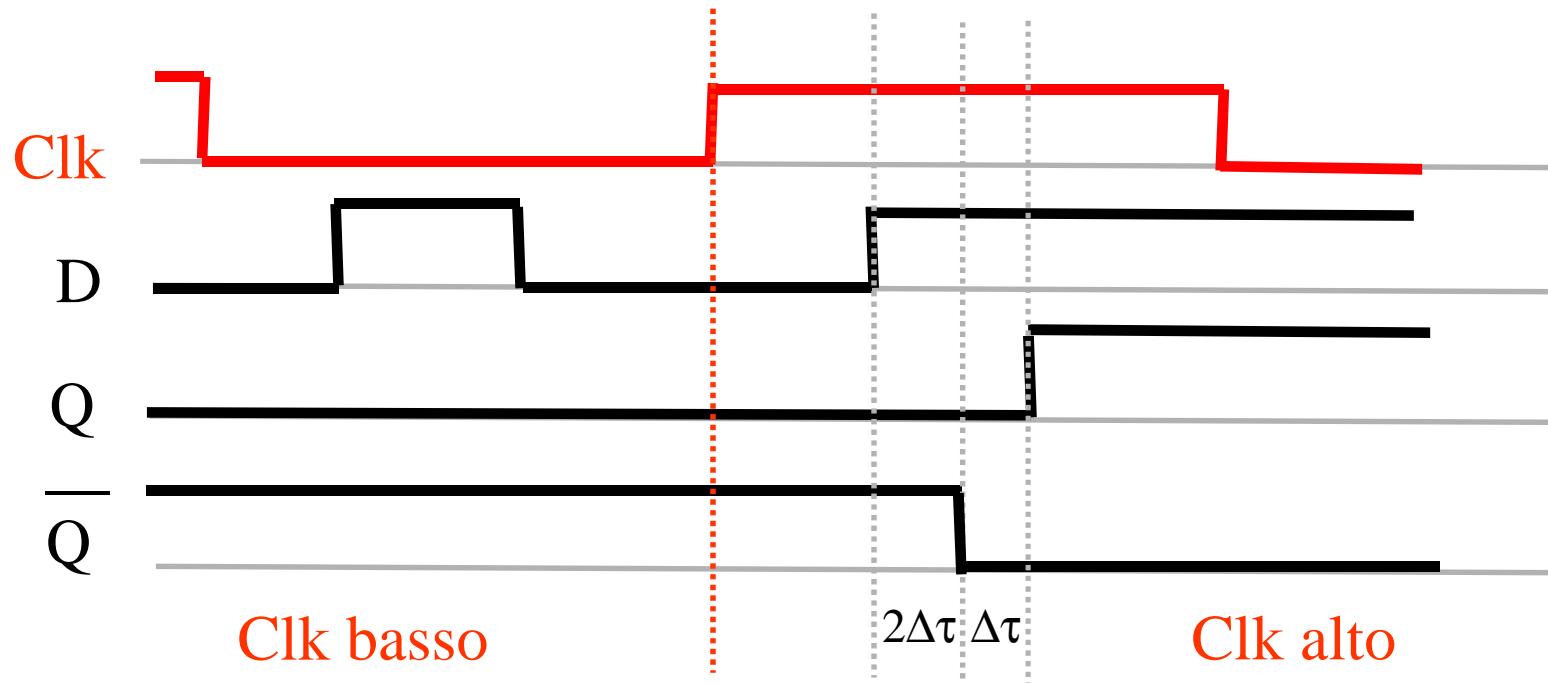
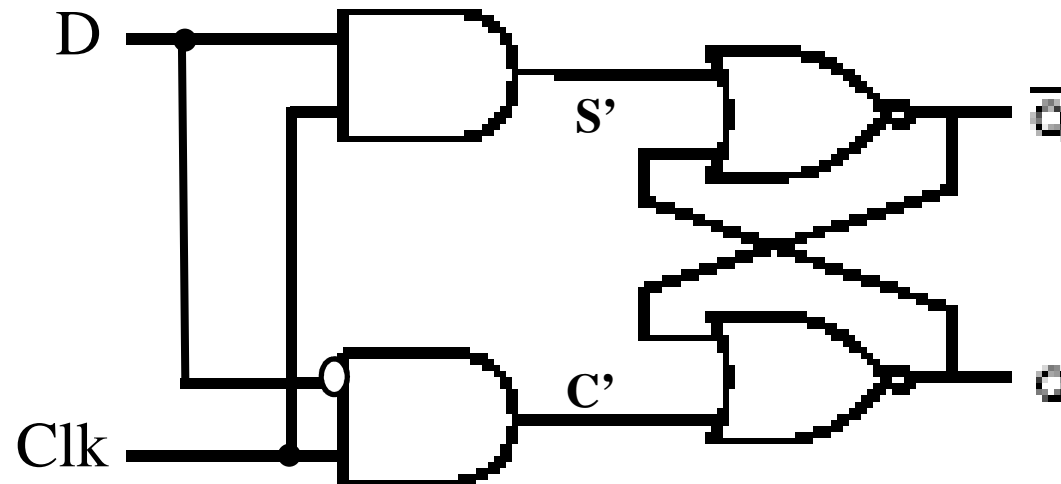
```
if clk = 1
  then
     $Q^* = D$ 
  else
     $Q^* = Q$ 
```



**Latch-D**



# 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

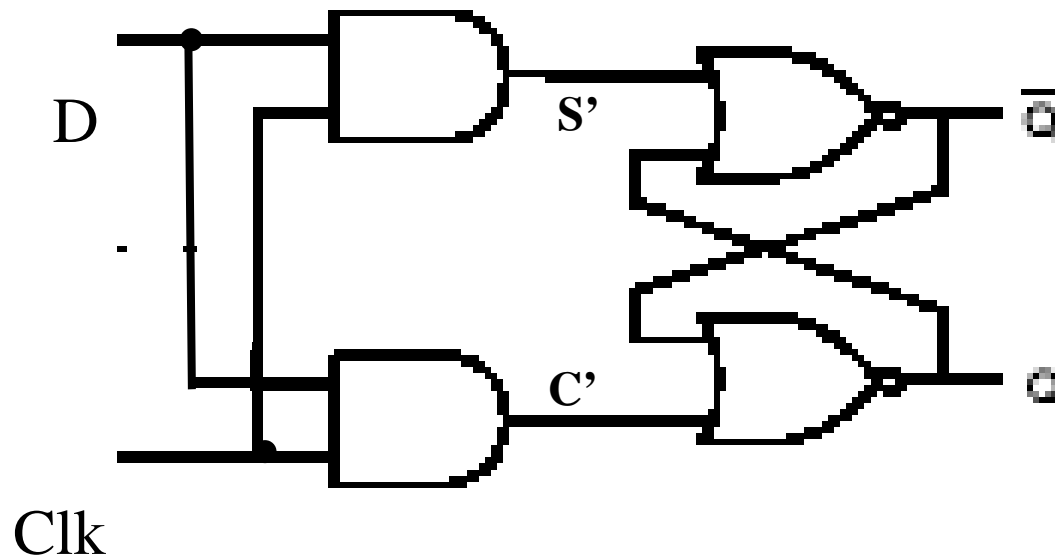
Q è l'uscita del latch: **stato presente**.

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



# Tabella 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

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

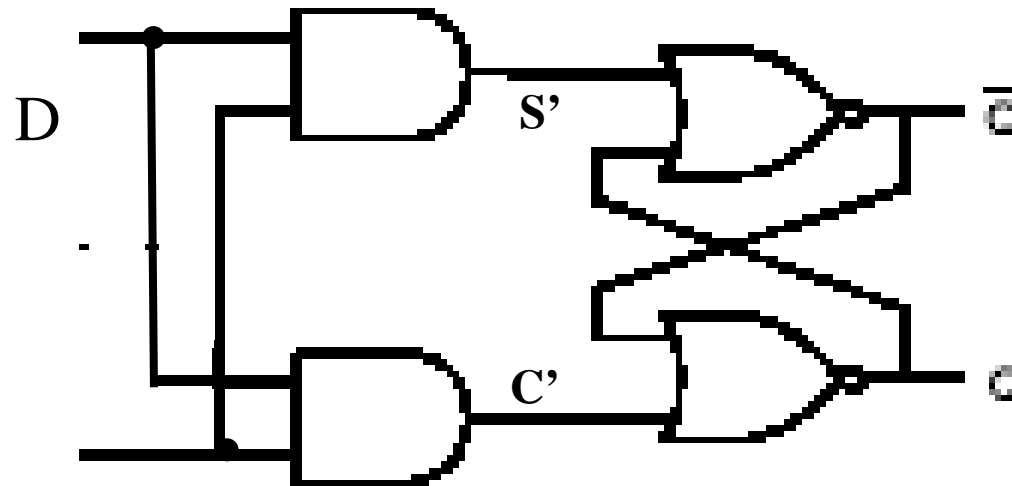
$$= \bar{T}Q + TD$$

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



Latch sincroni SR

Latch sincroni D

Flip-flop



# I bistabili

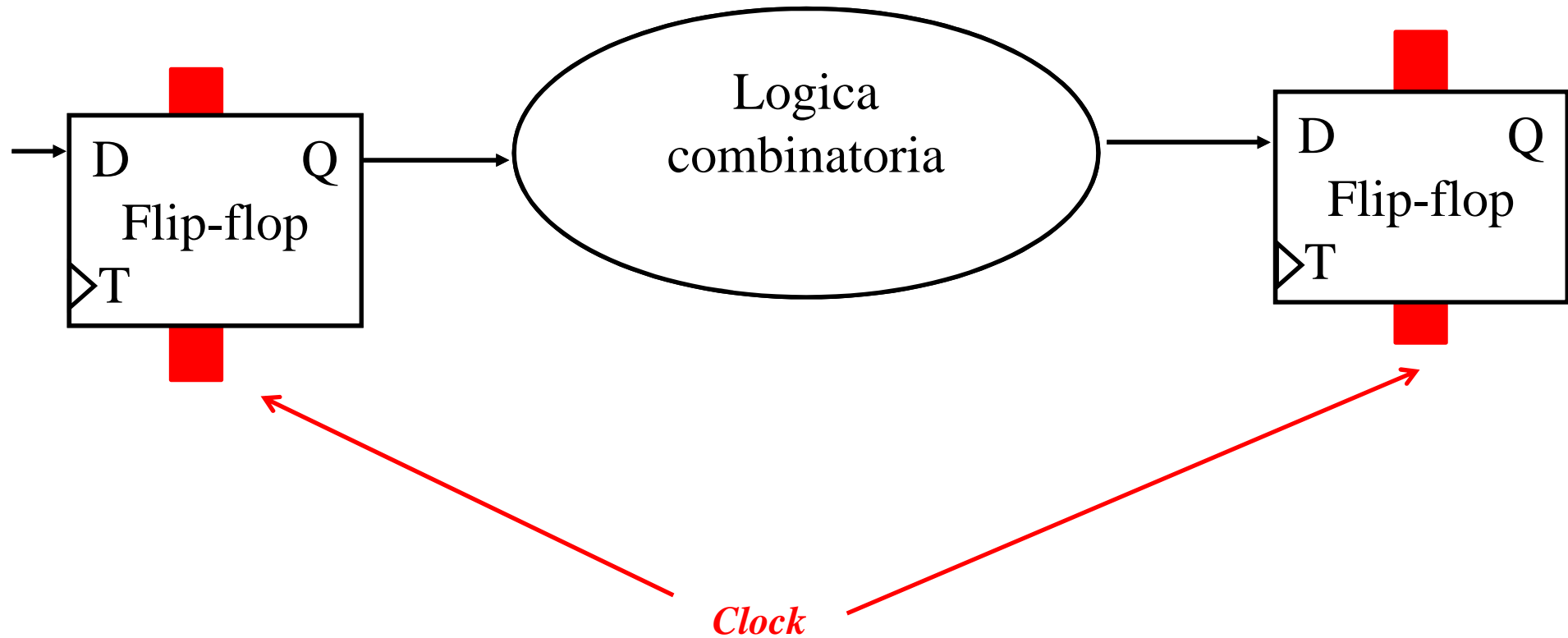


- Elementi di memoria (latch)
- “Cancelli” (flip-flop)



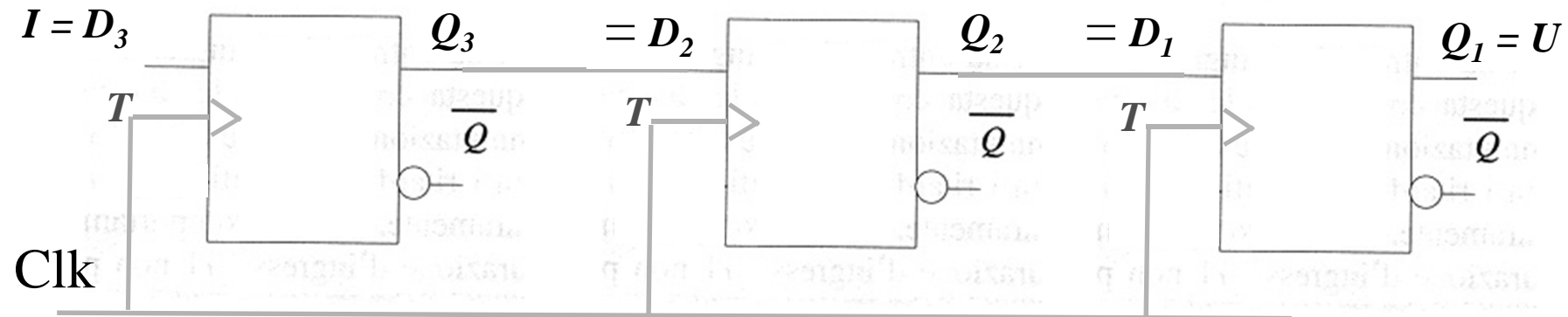


# Struttura di un circuito sequenziale





# Shift register

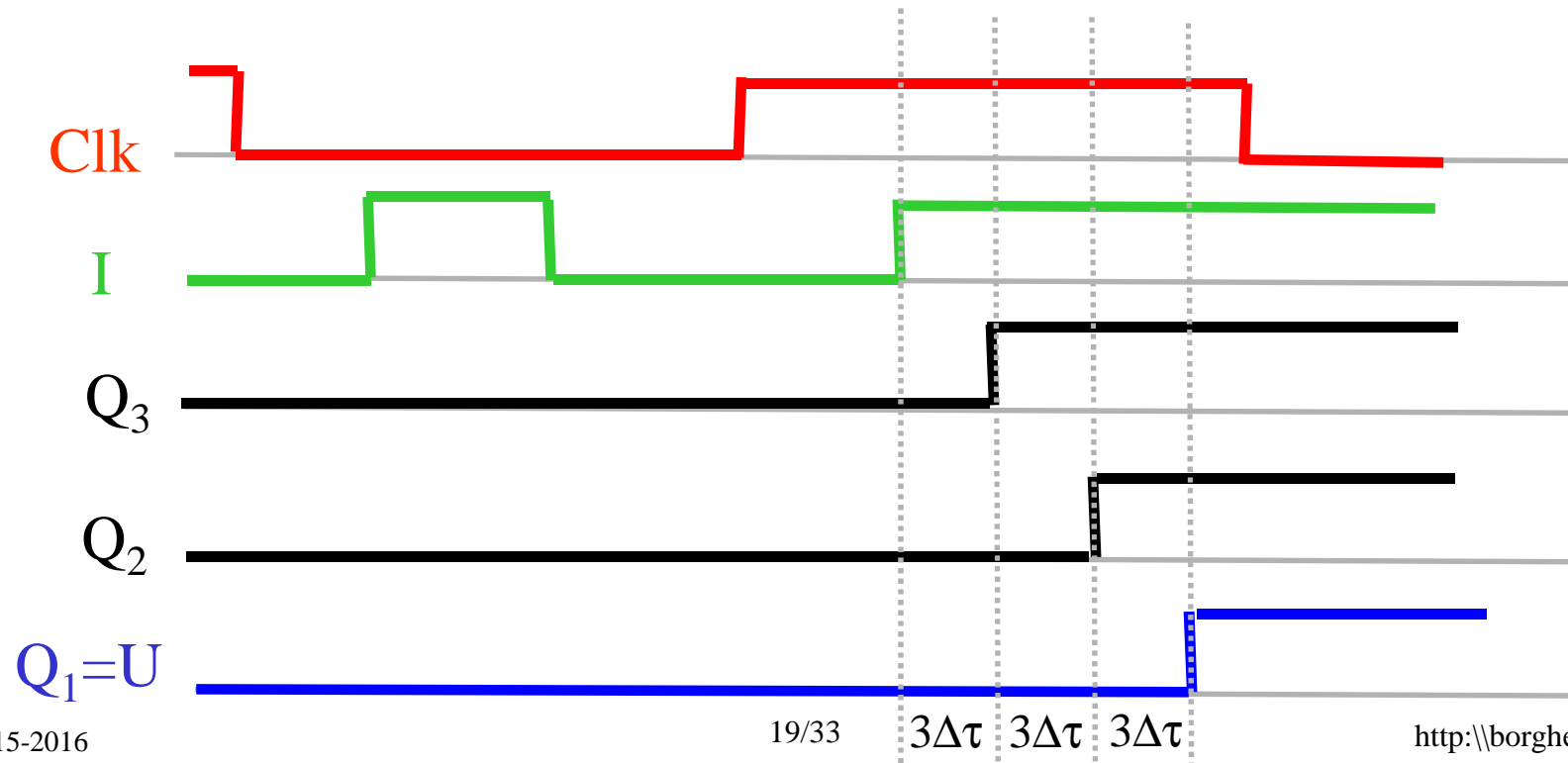
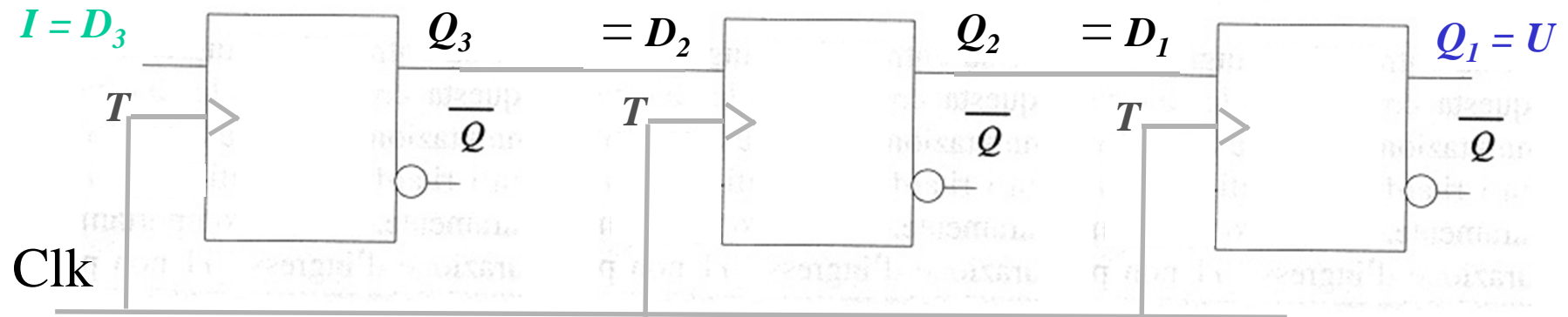


Registro a scorrimento (shift register o barrel shifter).

- 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.
- Il valore contenuto nell'elemento più a dx dove va?
- Qual'è il problema con l'utilizzo dei latch sincroni?

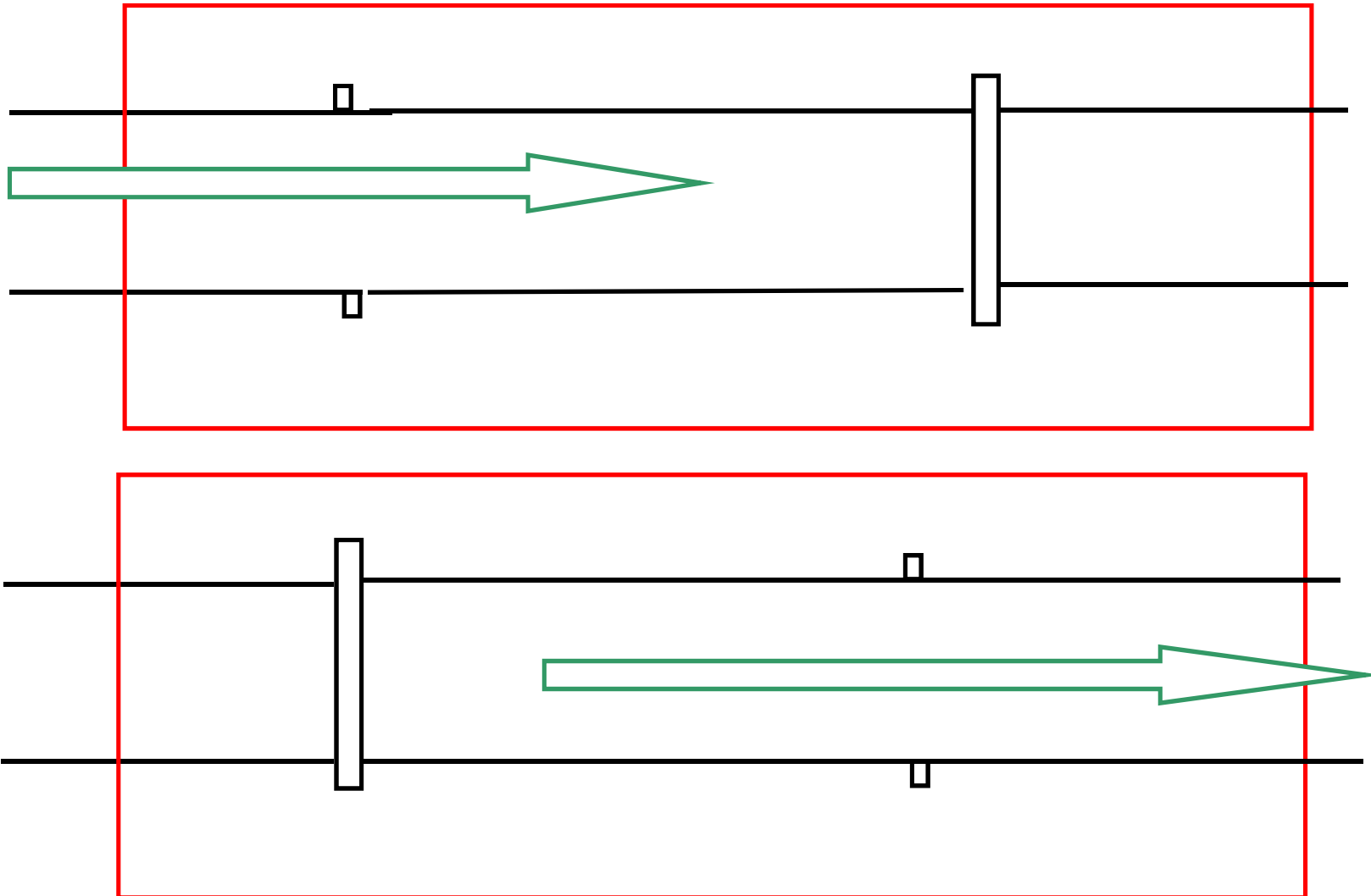


# Shift register con i latch (i problemi)





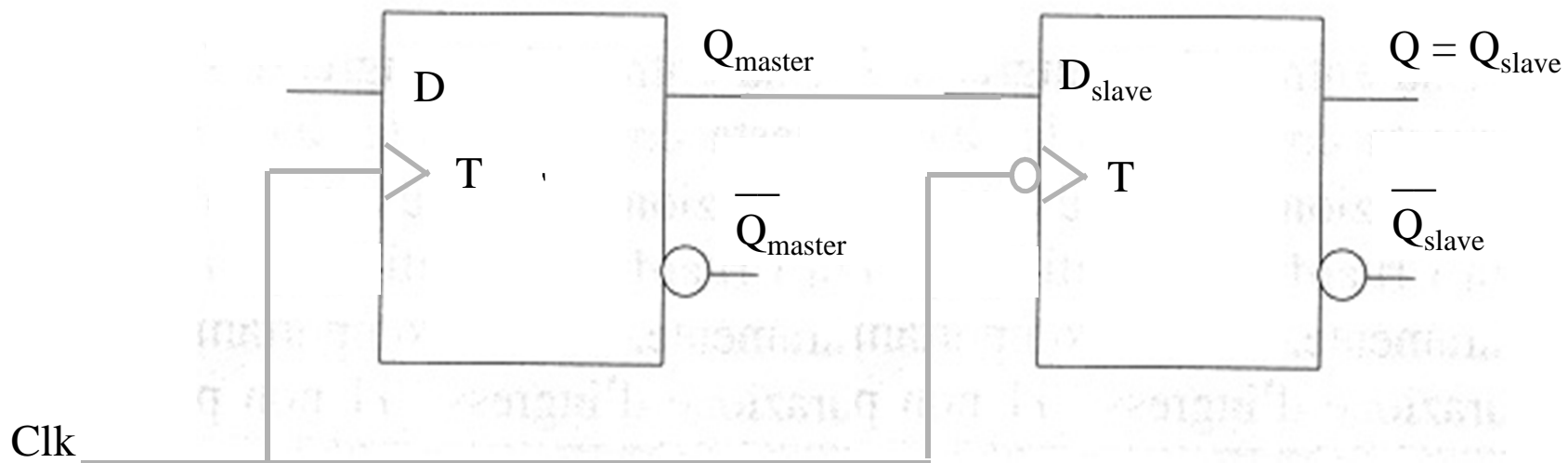
# Dispositivo di sincronizzazione





# Flip-flop

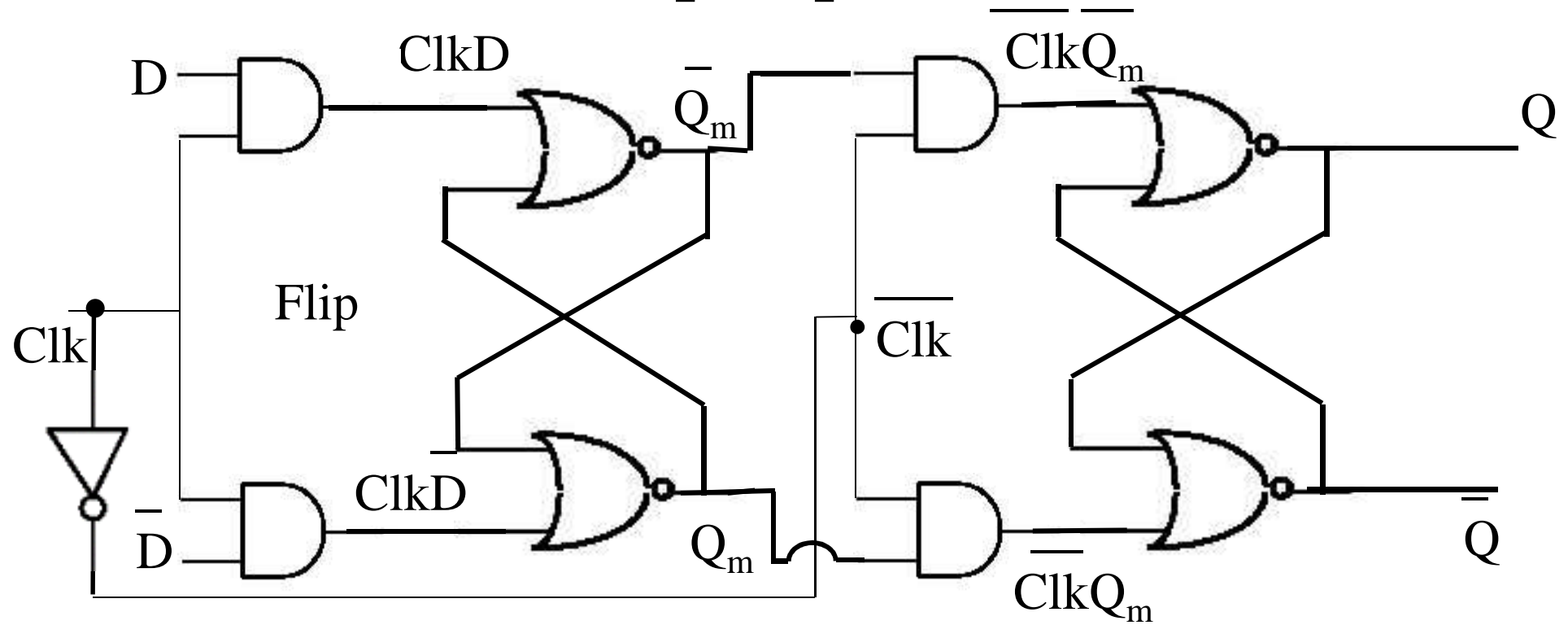
**Dispositivi attivi sul fronte** (di salita o discesa) **del clock** (edge sensitive): il loro stato (uscita) può commutare solo in corrispondenza della transizione alto->basso o basso->alto del clock.



Configurazione **Master-Slave**



# Flip-flop D



D

$\bar{Q}_m$

Q

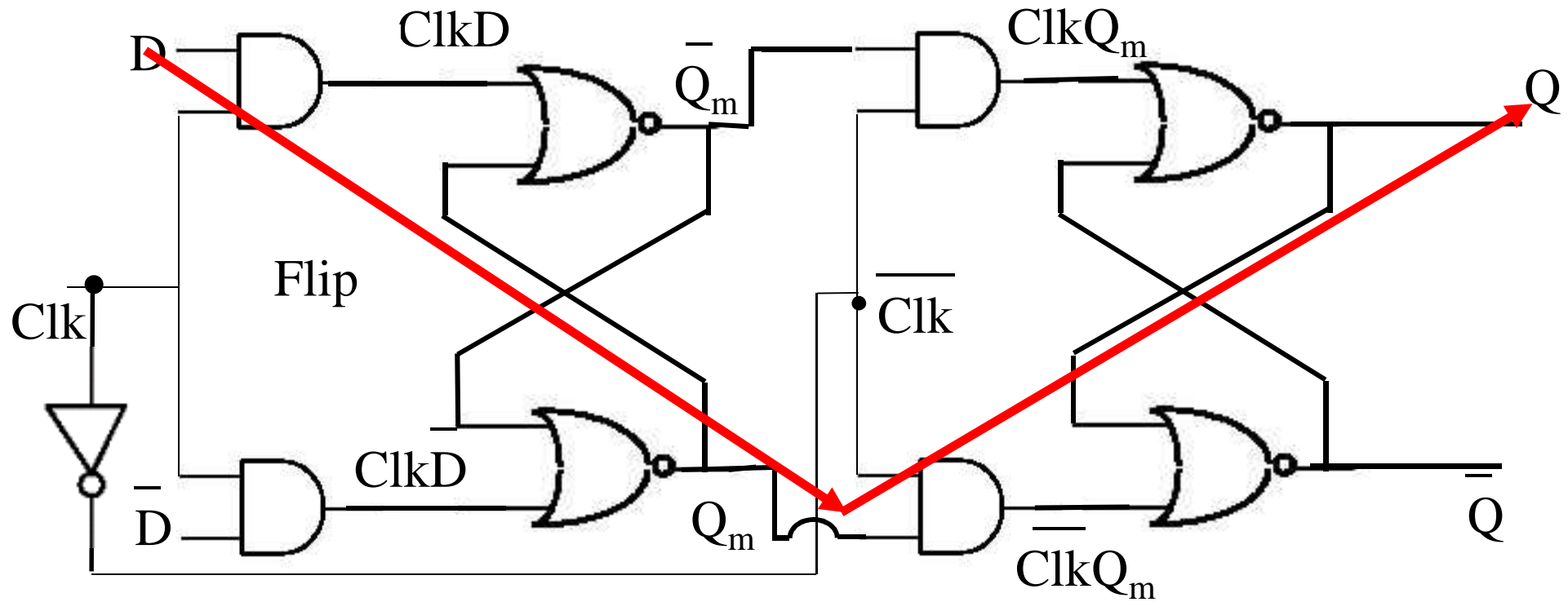
Clk

$Q_m$

$\bar{Q}$



# Funzionamento del flip-flop D



D

$\bar{Q}_m$

Q

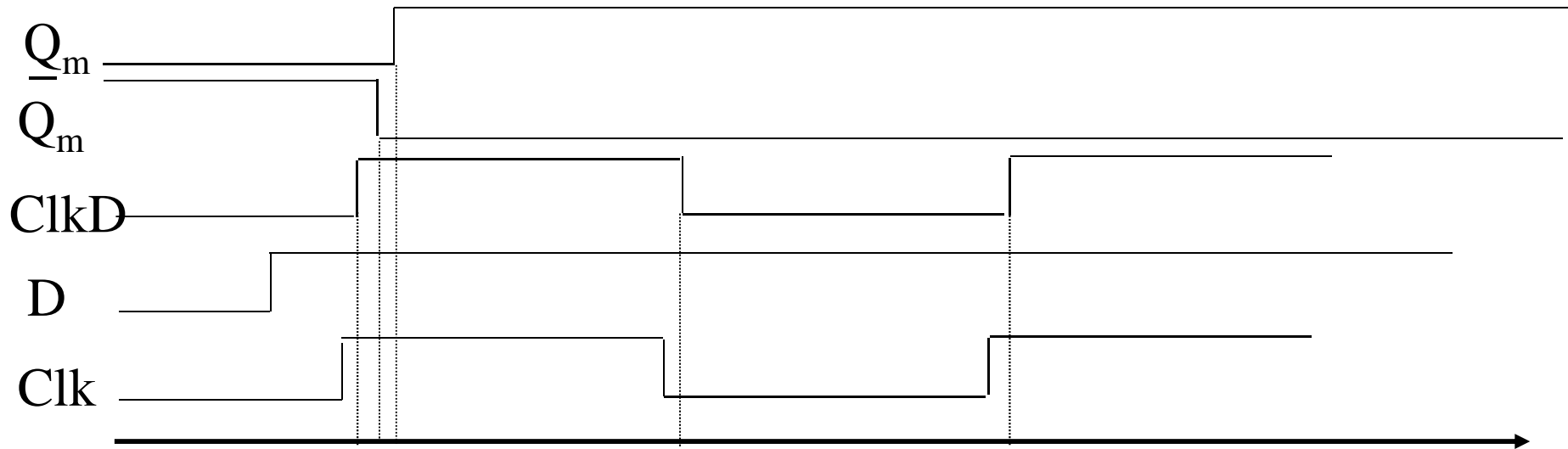
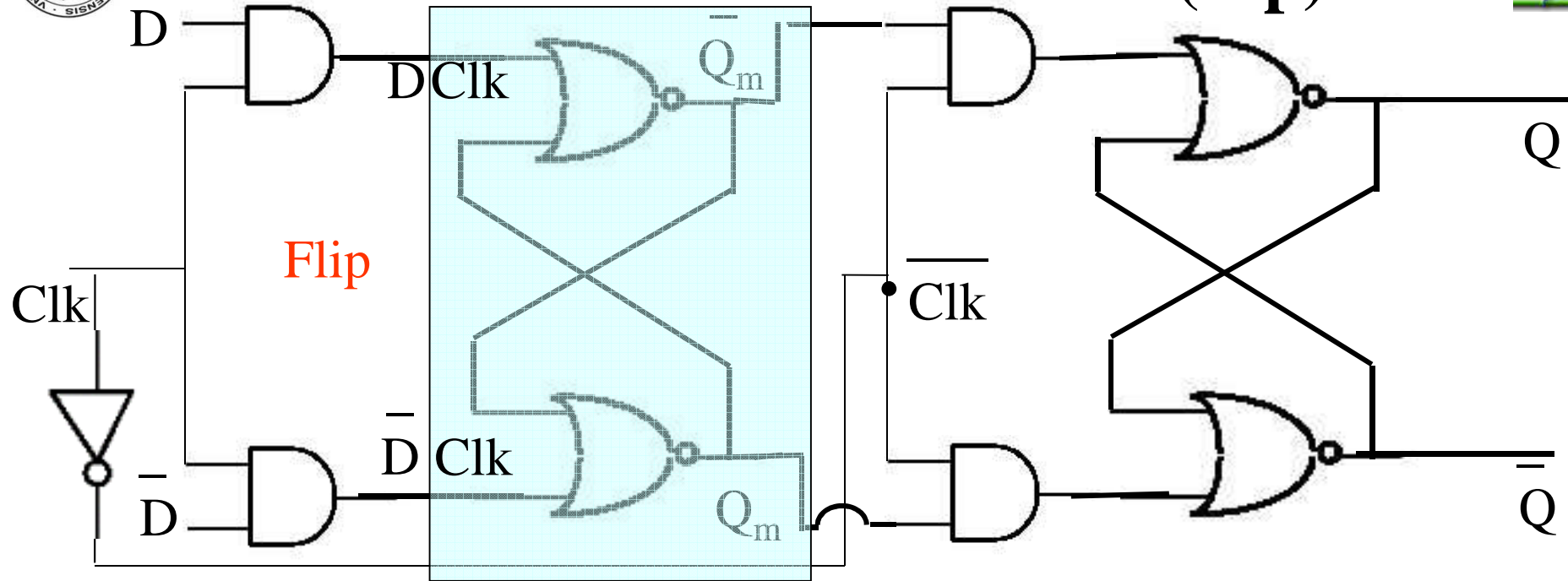
Clk

$Q_m$

$\bar{Q}$



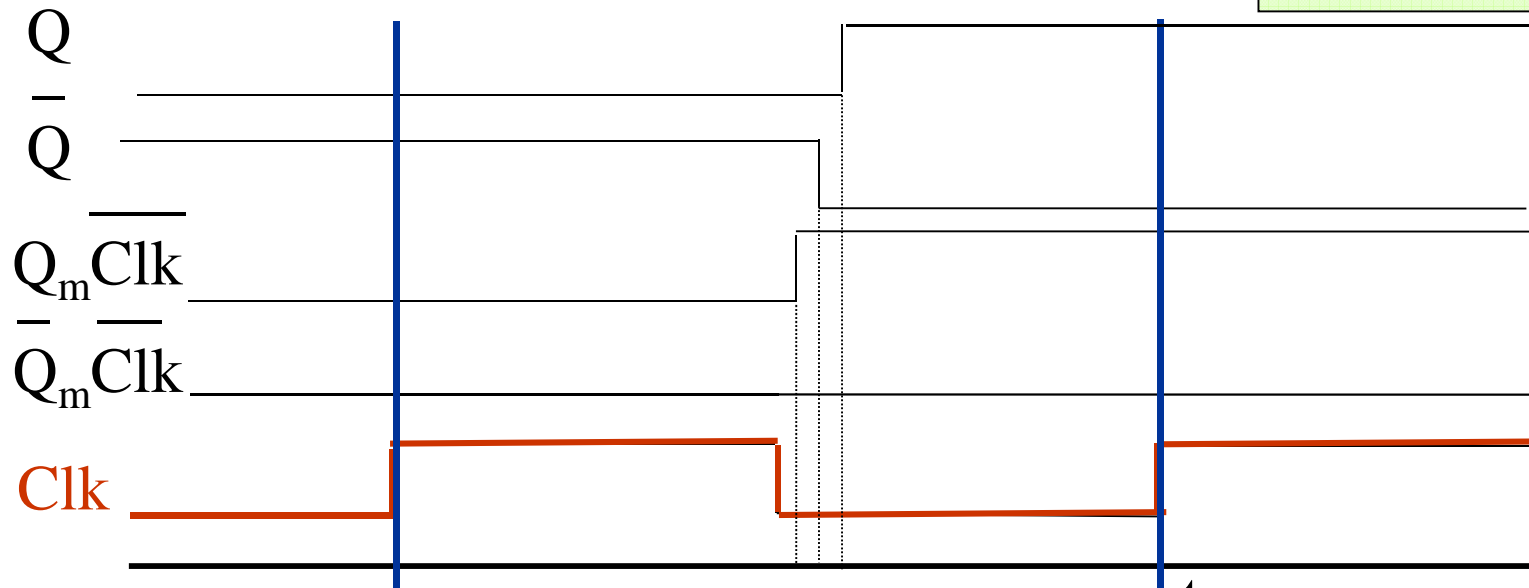
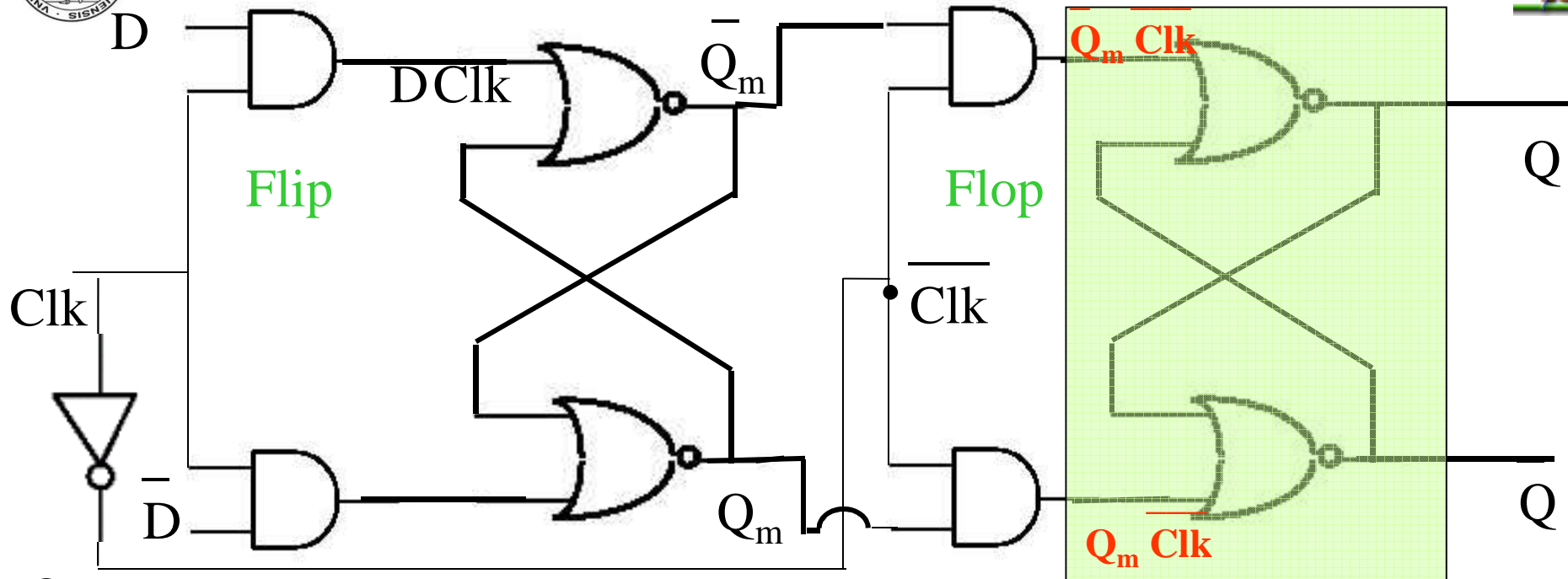
# Funzionamento del flip-flop D (flip)







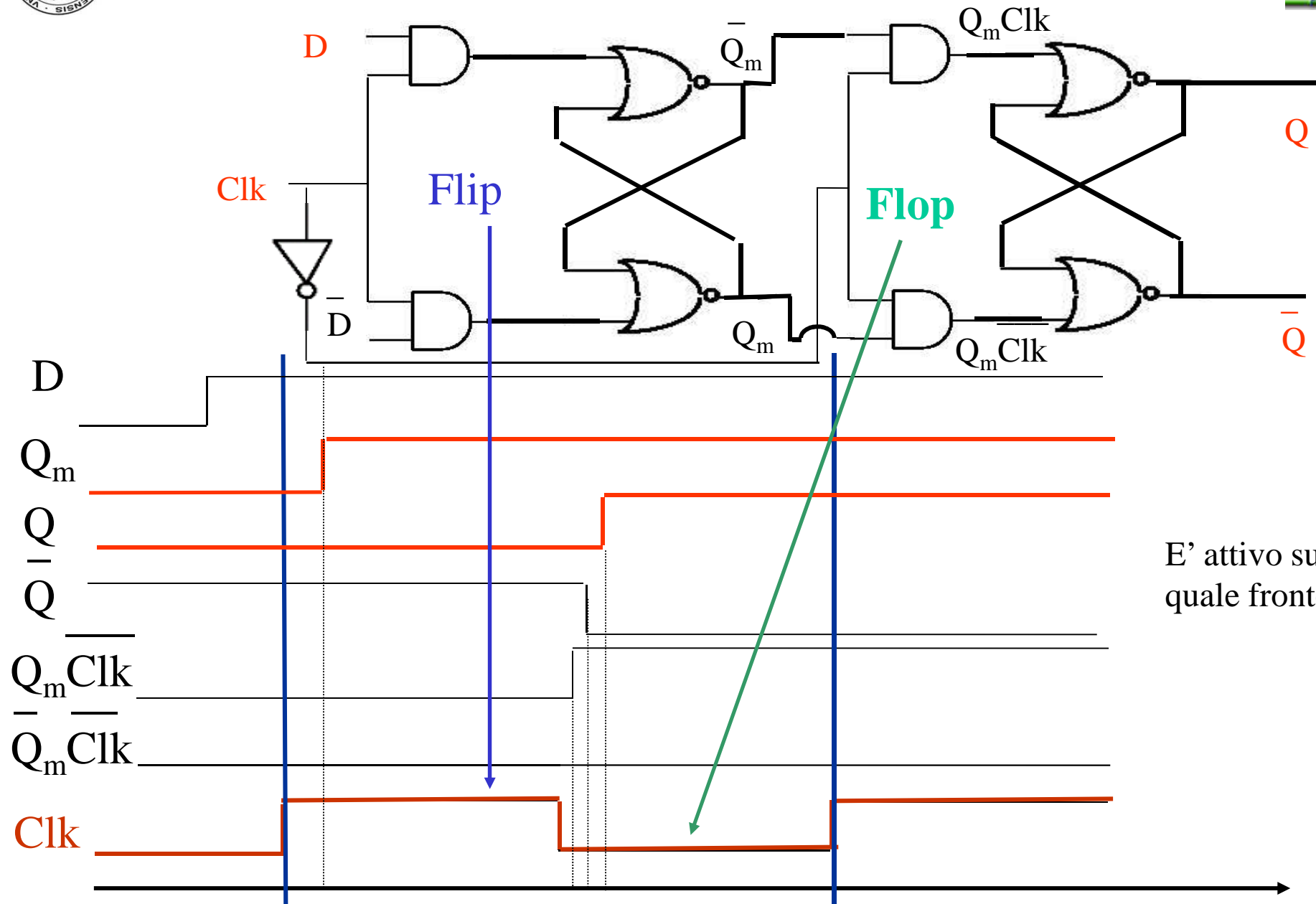
# Funzionamento del flip-flop D (flop)



Ritardo globale:  
 $3 \Delta\tau$   
misurato a partire  
dal fronte di  
discesa.

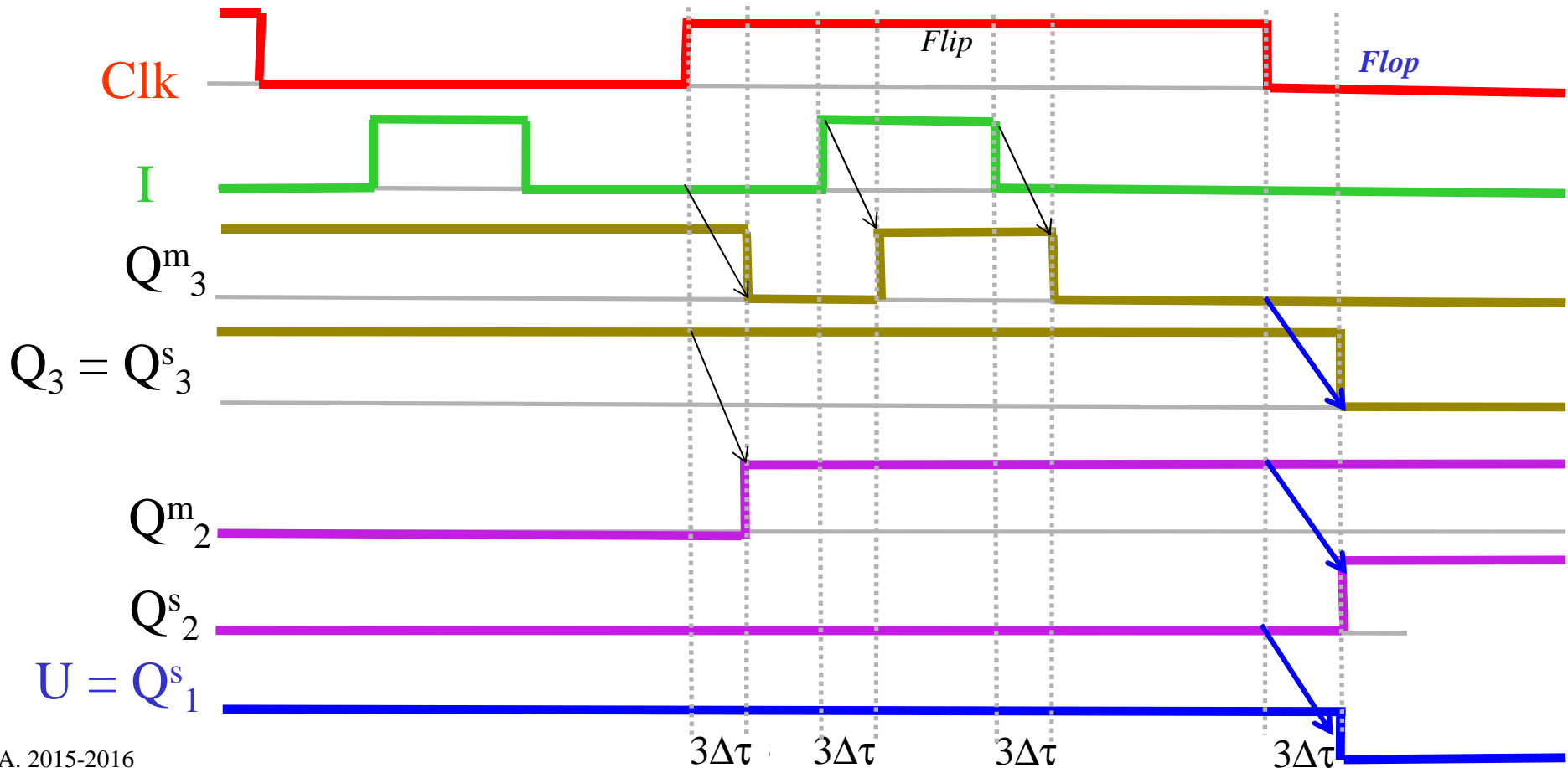
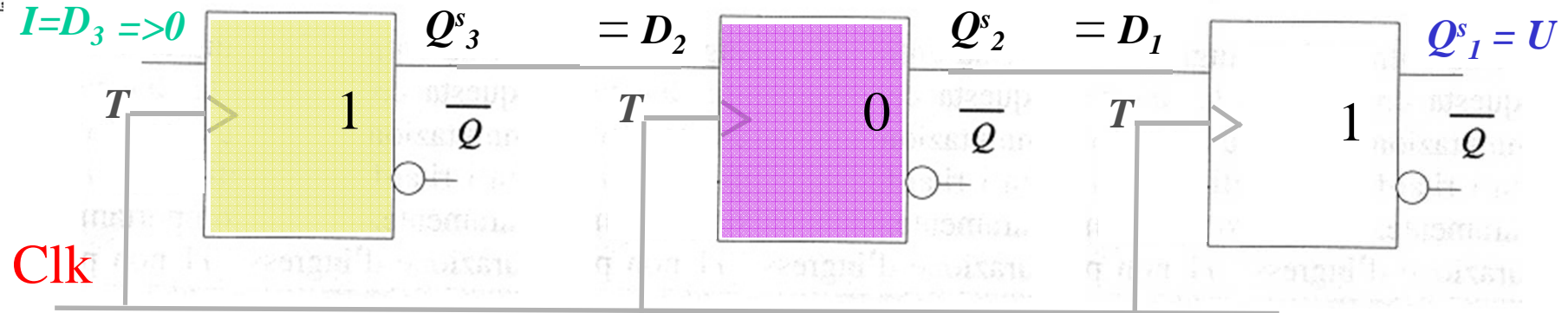


# Temporizzazione dei flip-flop





# Shift register con i flip-flop





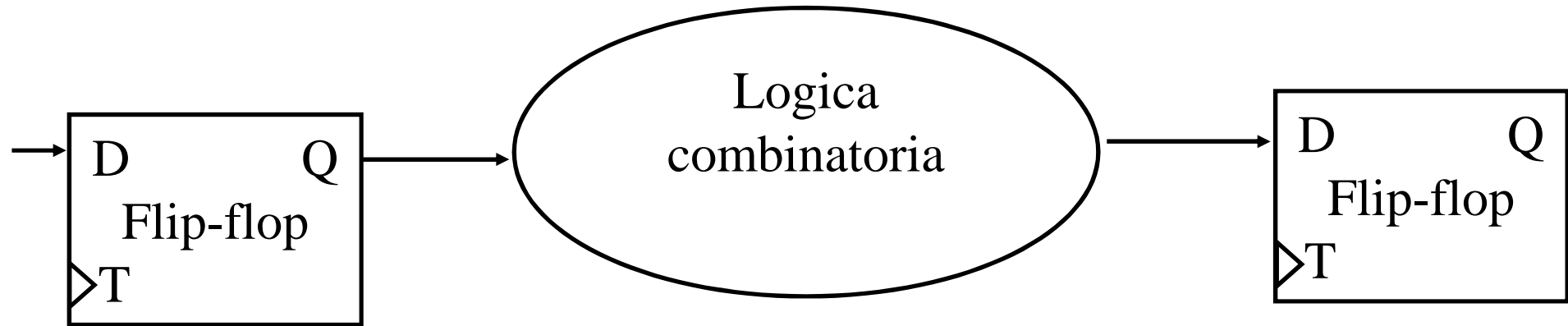
# Configurazione master-slave



- Semi-periodo di clock alto:
  - Il master è trasparente: ingresso -> uscita del latch master.
  - Lo slave è “opaco” -> mantiene l’uscita.
  - Lo slave è “disaccoppiato” dal latch master.
- 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 e porta in uscita l’uscita del latch master.



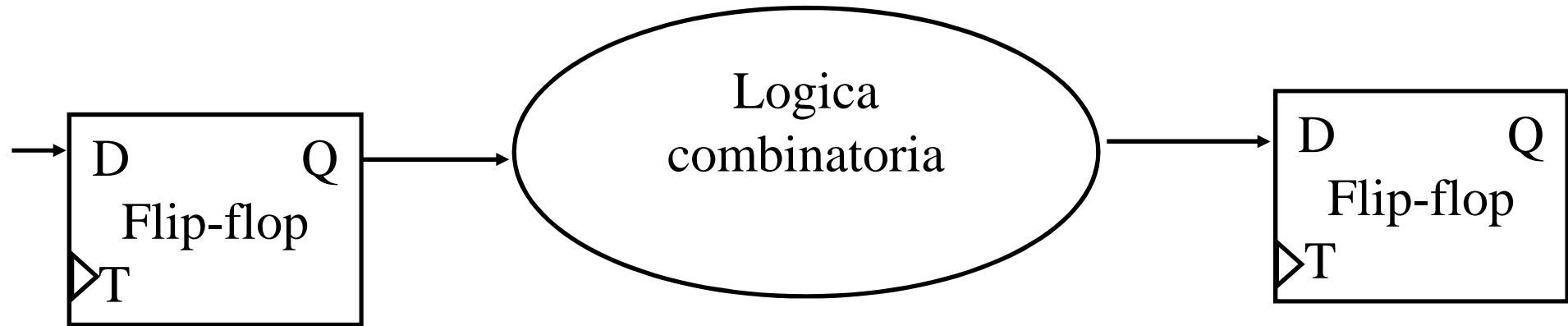
# Struttura di un circuito sequenziale



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



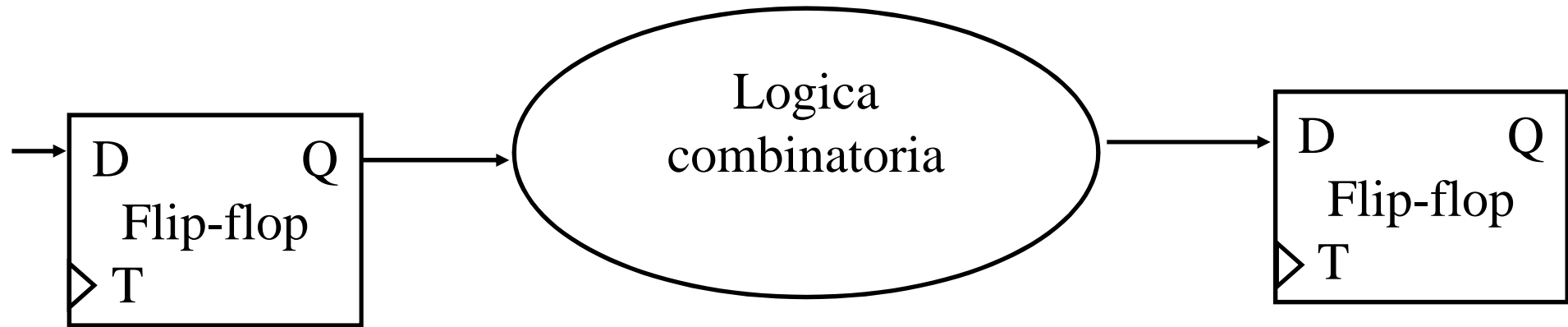
# Temporizzazione di un circuito sequenziale



- 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.

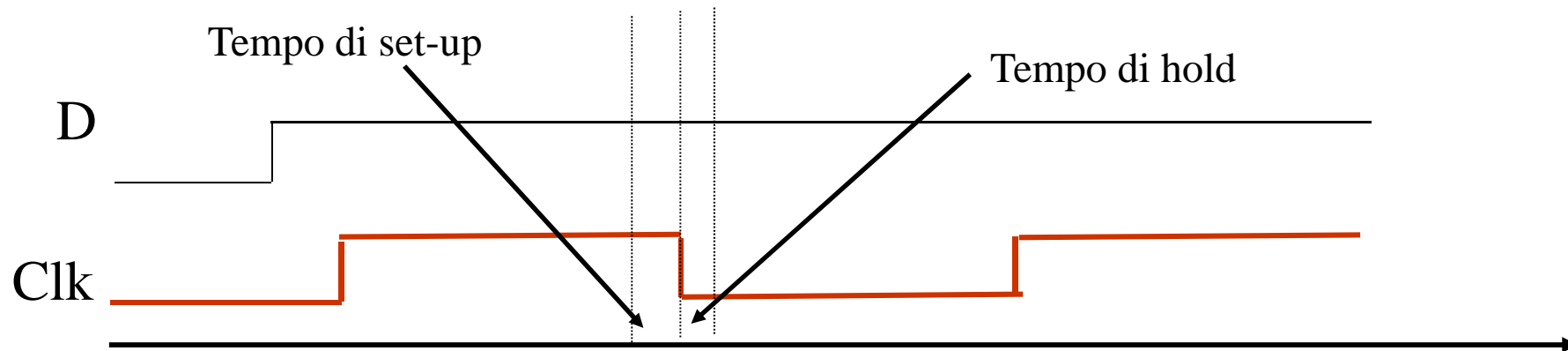


# Temporizzazione: problemi



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

- **Tempo di set-up:** è il tempo minimo per cui deve rimanere stabile l'input D prima del fronte di clock.
- **Tempo di hold:** è il tempo minimo per cui deve rimanere stabile l'input D dopo il fronte di clock (solitamente trascurabile).

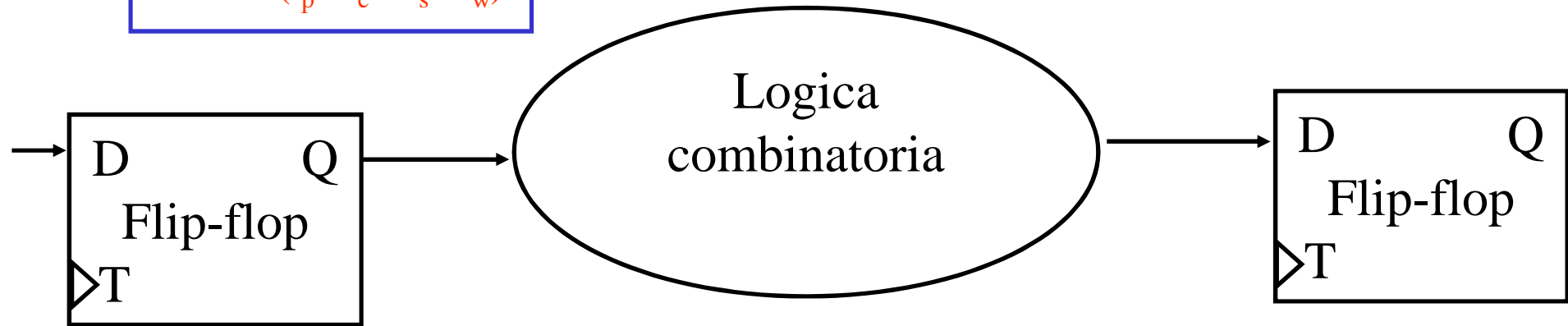




# Temporizzazione: Come si dimensiona il clock

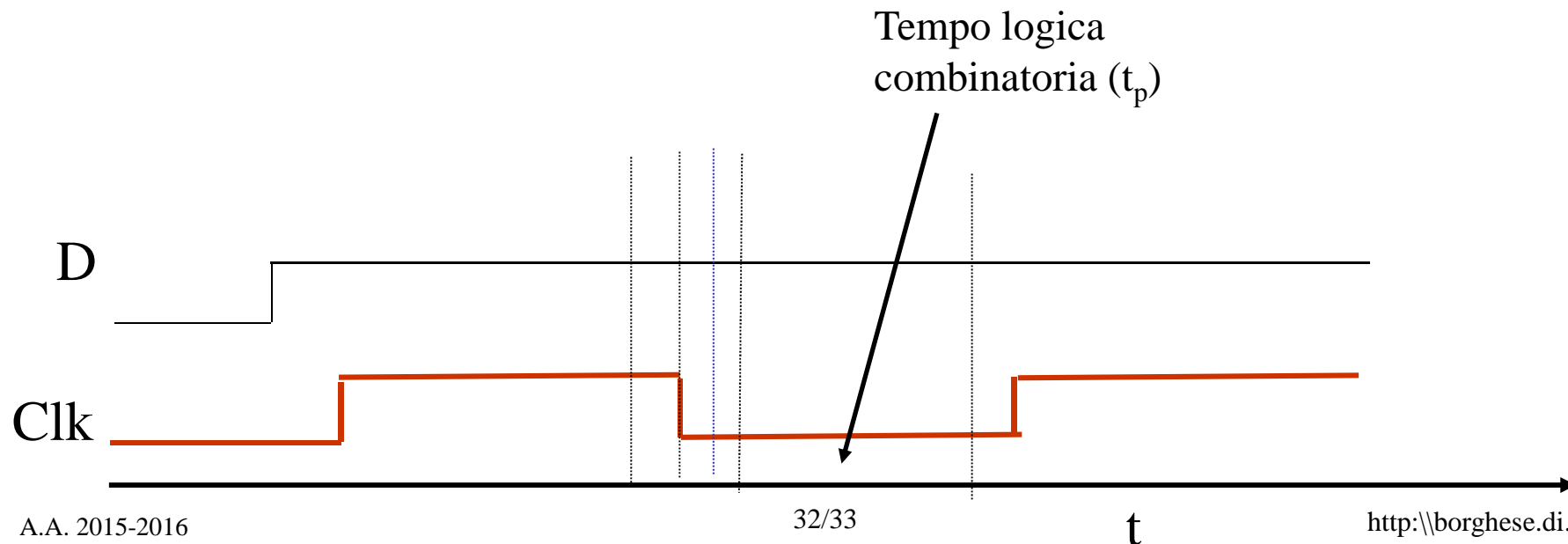


$$T > k * (t_p + t_c + t_s + t_w)$$



**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 ( $t_w$ ).







# Sommario



Latch sincroni SR

Latch sincroni D

Flip-flop