



# I flip-flop ed il register file

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

Università degli Studi di Milano

Riferimento sul Patterson: Sezioni C.9 e C.11



## Sommario

I registri ed il register file

I problemi dei latch trasparenti sincroni

I bistabili DT



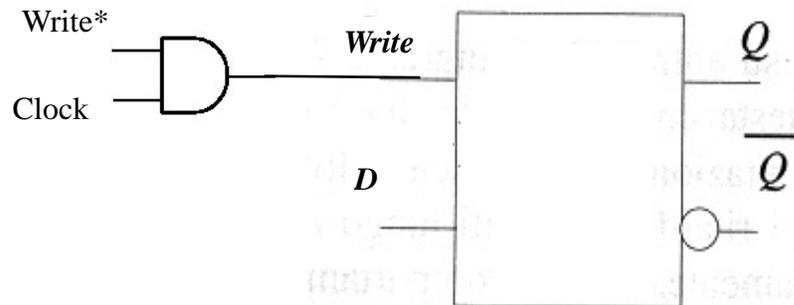
## Latch sincrono come elemento di memoria



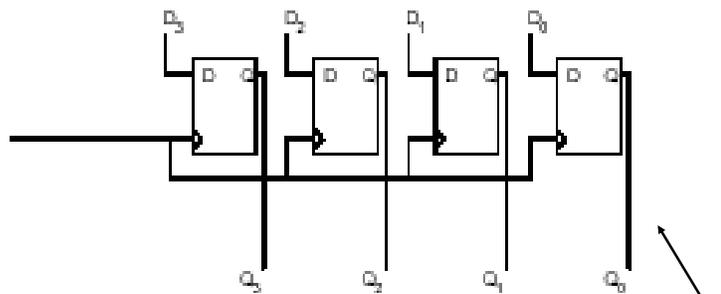
E' trasparente quando Write = 1

Se Write = 1  $Q_{t+1} = D$

Se Write = 0  $Q_{t+1} = Q_t$



## Registri



Un registro a 4 bit.

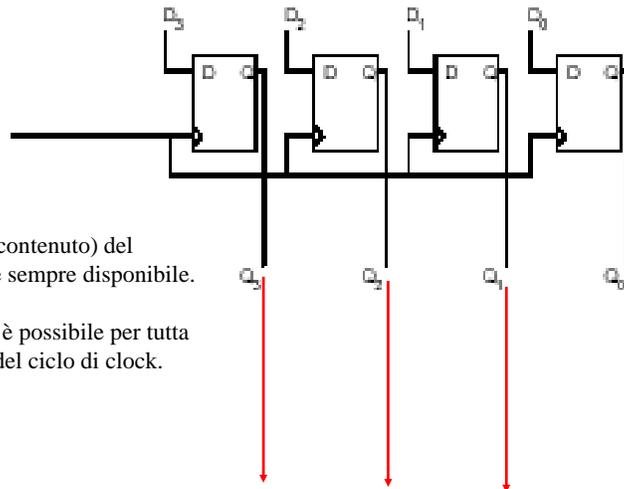
Memorizza 4 bit.

Latch di tipo D

NB Non è un registro a scorrimento (shift register!)



## Lettura di un registro



Lo stato (contenuto) del bistabile è sempre disponibile.

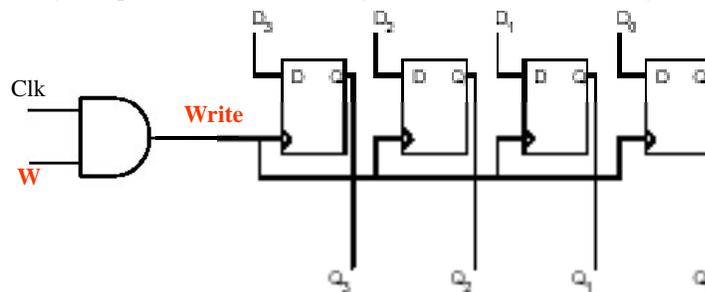
La lettura è possibile per tutta la durata del ciclo di clock.



## Scrittura di un registro



Ad ogni colpo di clock lo stato del registro assume il valore dell'ingresso dati.



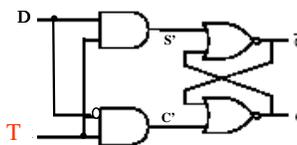
Cosa occorre modificare perchè il registro venga scritto quando serve?

Introdurre una sorta di *"apertura del cancello (chiusura circuito)"*.

Può essere sincronizzata o meno con il clock.

Il clock apre il passaggio al contenuto di D attraverso il latch.

Quando il segnale di Write è a zero, lo stato non varia.





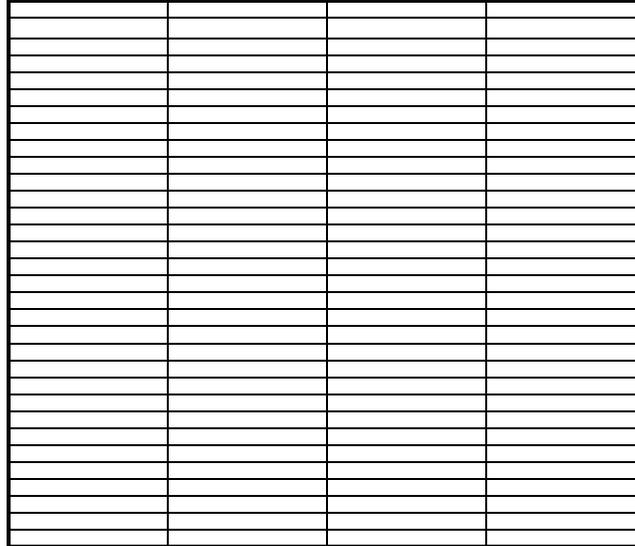
## Banco di Registri

32 bit

Struttura costituita da un insieme di registri

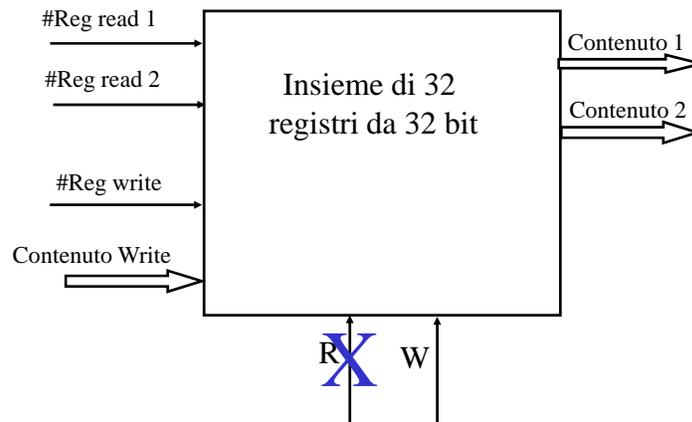


32 registers



## Register file

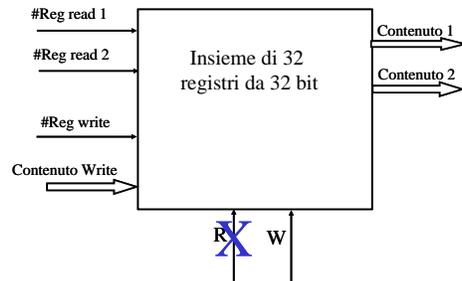
Banco di registri utilizzabile come memoria



Possono essere letti / scritti fornendo il numero del registro.



## Gestione del register file



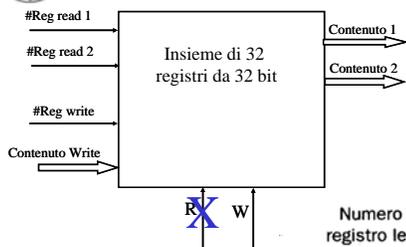
La lettura non modifica il contenuto di un registro (collego uscita Slave con il circuito combinatorio).

La scrittura invece richiede la modifica. Occorre il segnale W.

$$\#bit\_indirizzamento = \log_2 \#bit$$

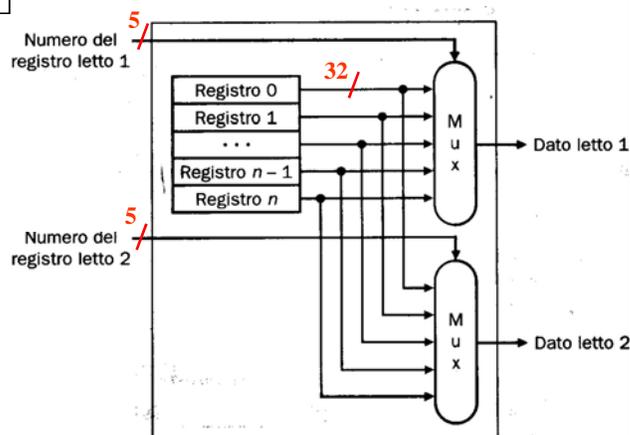


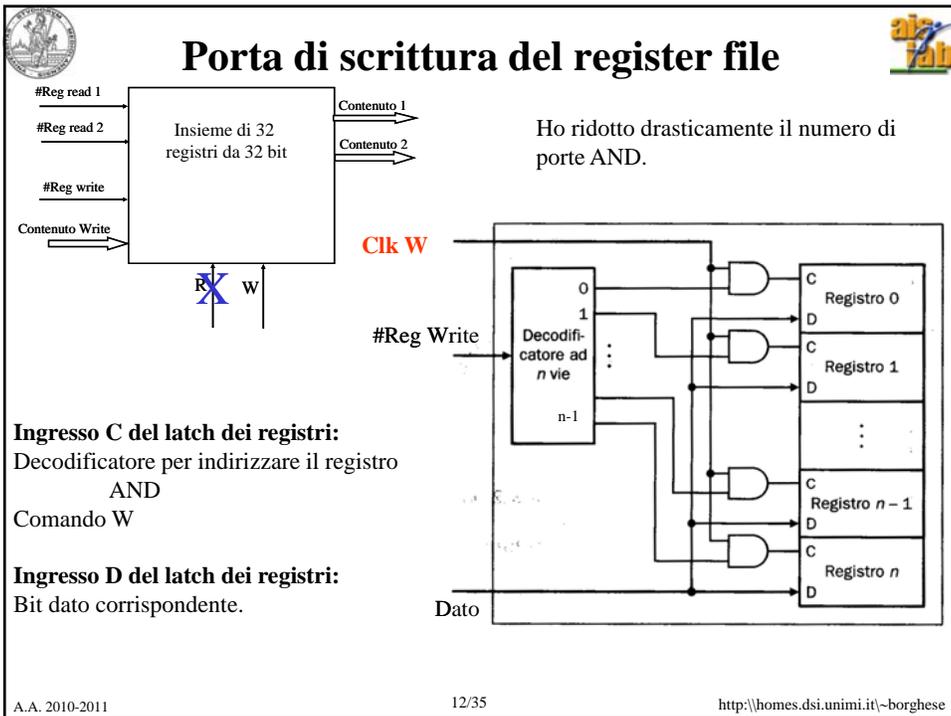
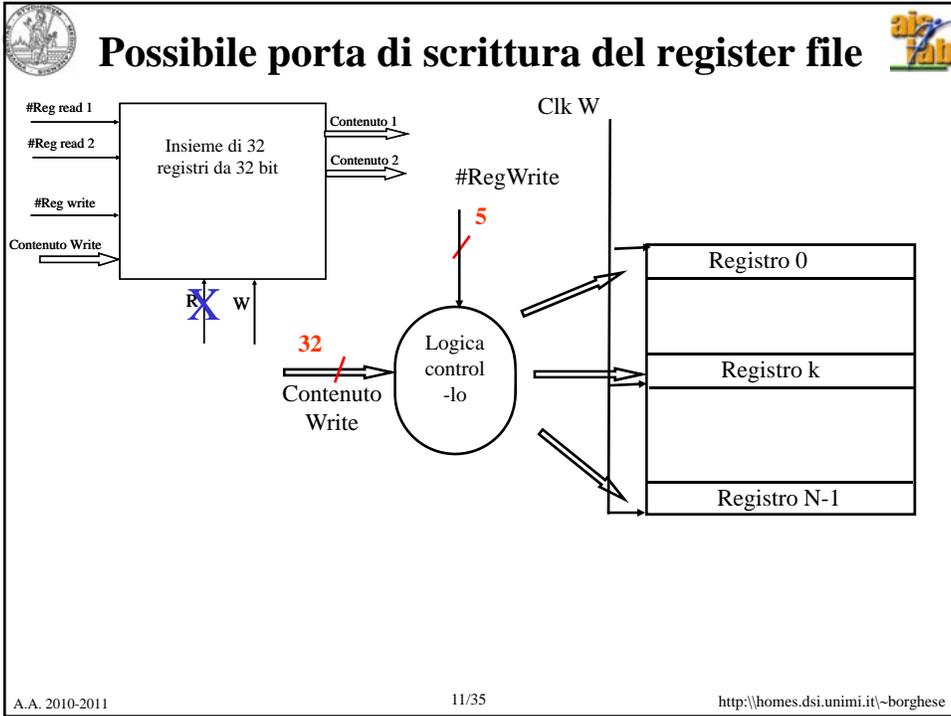
## Porta di lettura del register file



Un mux per ogni porta di lettura.

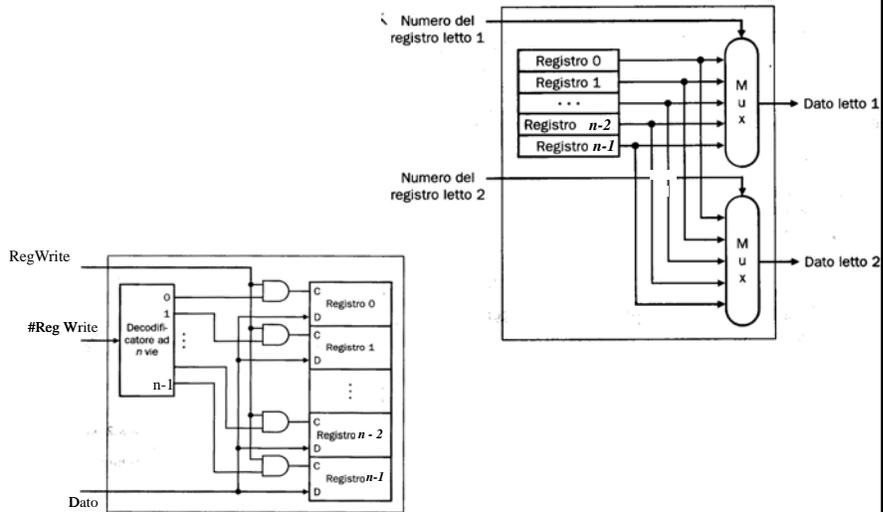
Ciascun Mux ha la complessità di 32 mux, uno per ogni bit.







## Register file



## Sommario



I registri ed il register file

I problemi dei latch trasparenti sincroni

I bistabili DT



## I bistabili



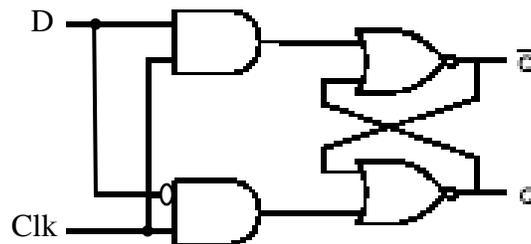
- Elementi di memoria (latch)
- “Cancelli” (flip-flop)



## I latch



I latch sono chiamati anche dispositivi trasparenti: quando il clock è alto, il valore di D viene riportato in uscita,  $Q = D$ .

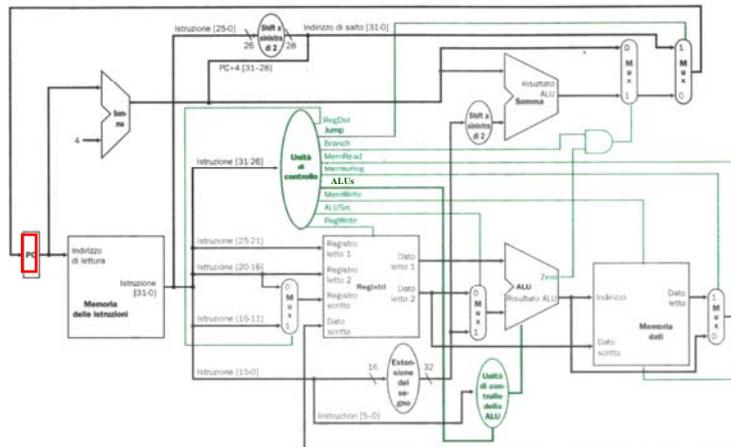


Per tutto il tempo in cui il clock è attivo, l'uscita è collegata all'ingresso D, dopo di che il “cancello” si chiude.

Il “cancello” è aperto per tutta la durata di un semiperiodo del clock. Problema?



## Il “cancello” nelle architetture

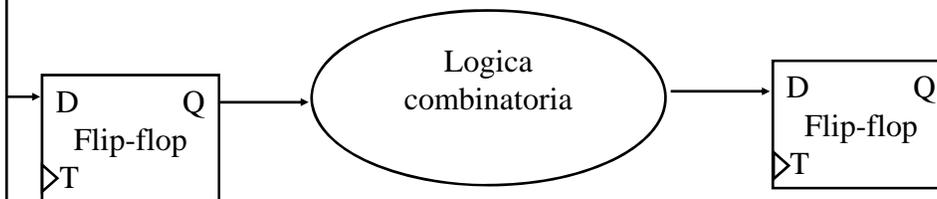


Il “cancello” viene inserito solamente in alcuni punti dell’architettura. In questi punti si **sincronizza** l’attività: “*nodì di sincronizzazione*”.

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

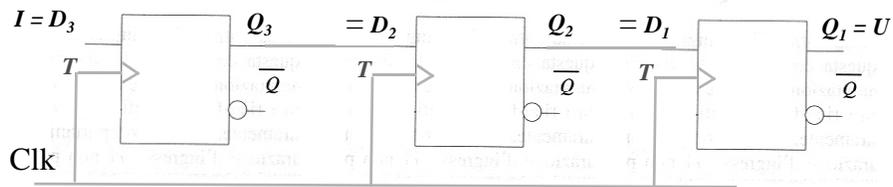


## Struttura di un circuito sequenziale





## Shift register

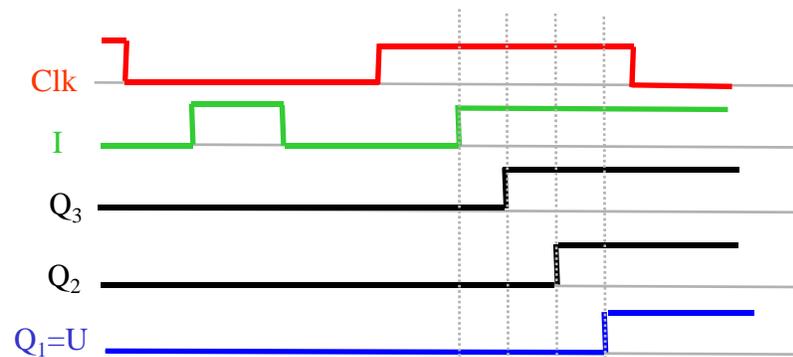
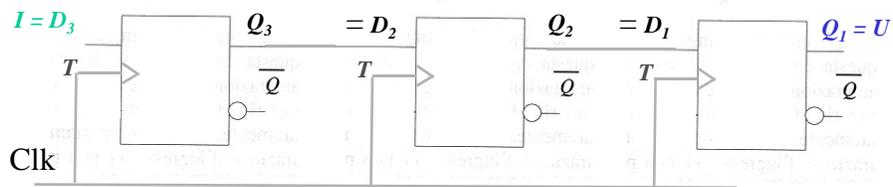


Registro a scorrimento (shift register o barrel shifter).

- 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?



## Shift register con i latch (i problemi)





## Sommario



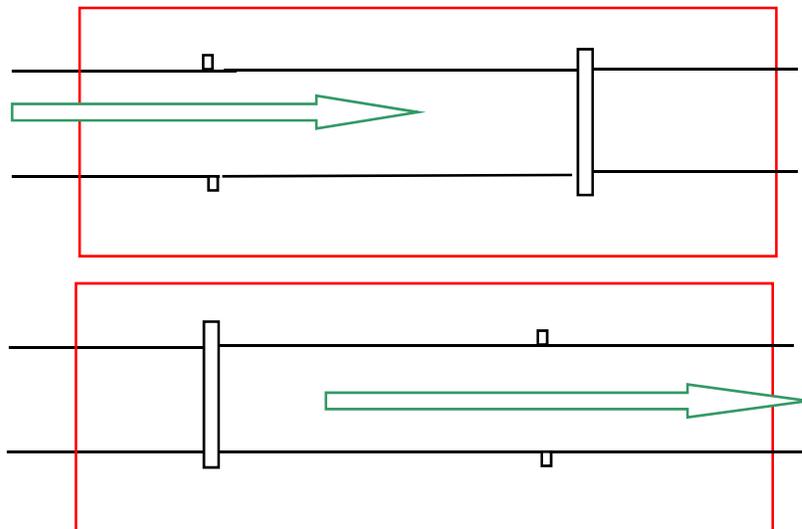
I registri ed il register file

I problemi dei latch trasparenti sincroni

**I bistabili DT**



## Dispositivo di sincronizzazione

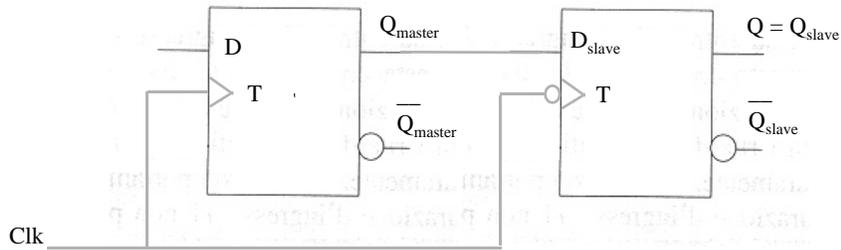




# Flip-flop



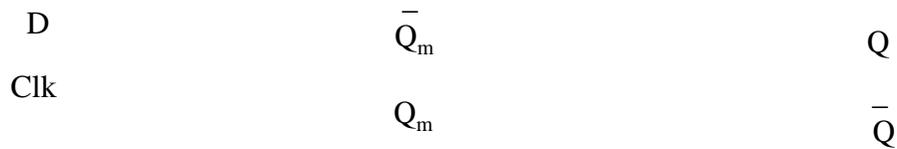
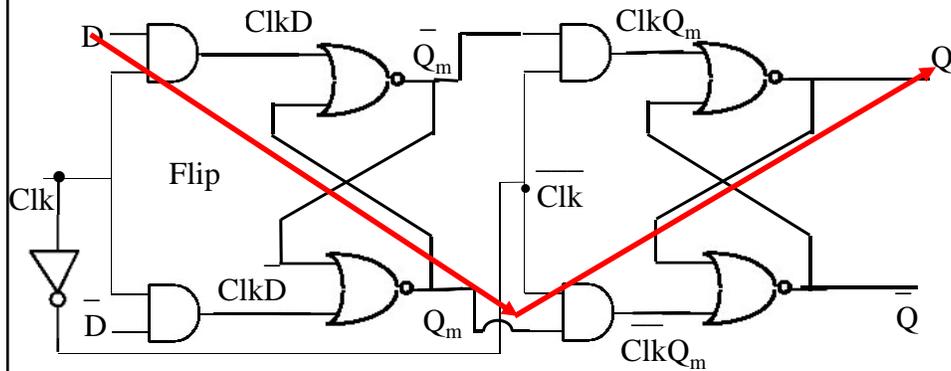
Dispositivi attivi sul fronte (di salita o discesa) del clock (edge sensitive): il loro stato (uscita) può commutare solo in corrispondenza della transizione alto->basso o basso->alto del clock.

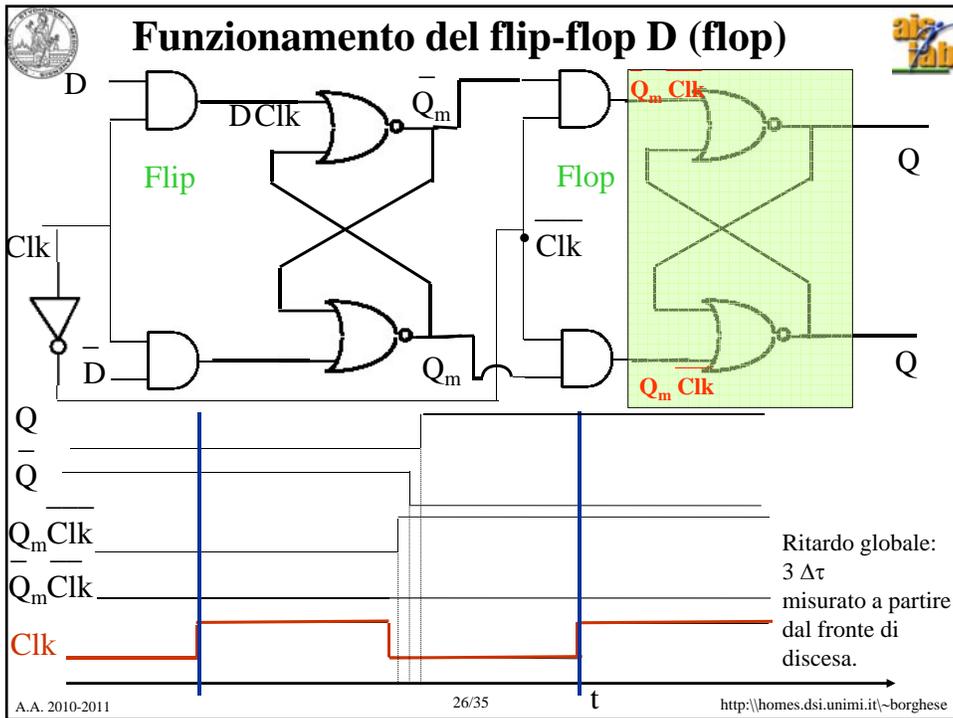
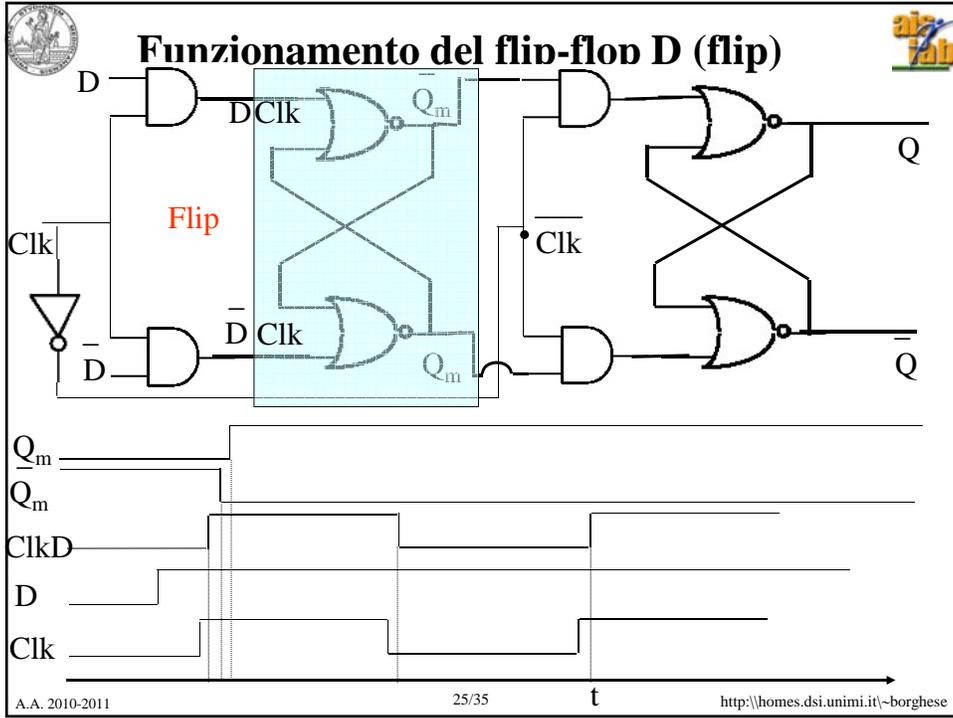


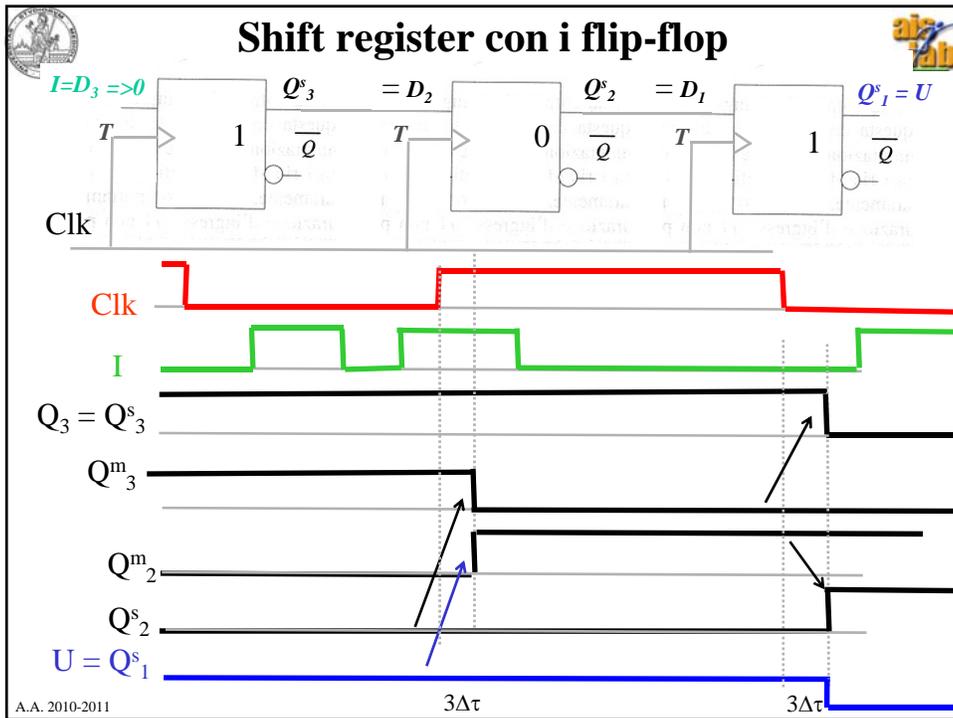
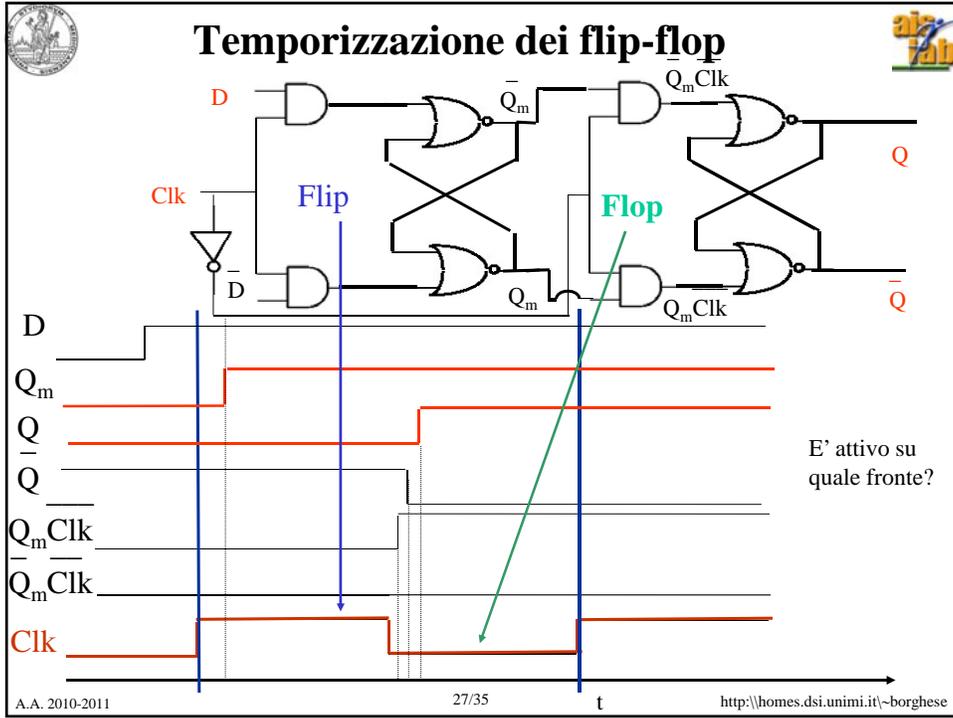
Configurazione Master-Slave



# Funzionamento del flip-flop D









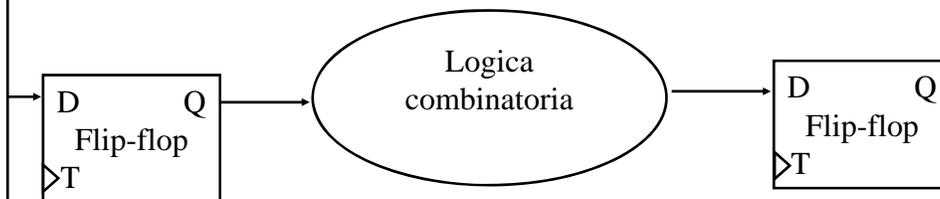
## Configurazione master-slave



- 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.



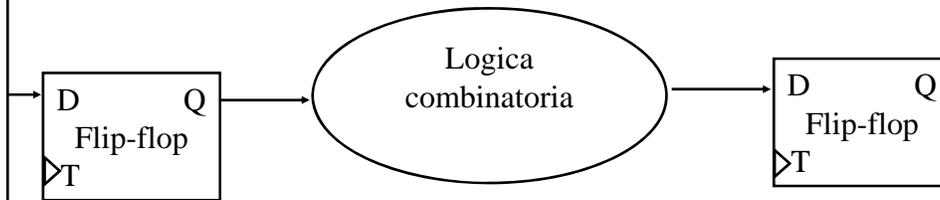
## Struttura di un circuito sequenziale



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



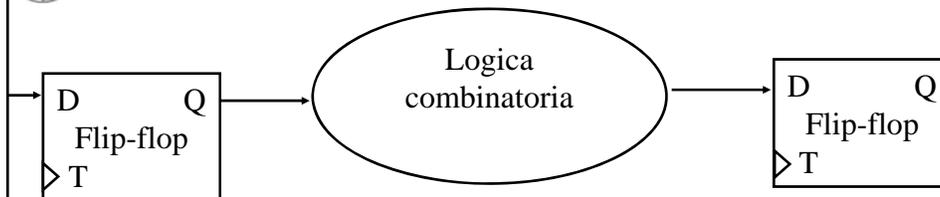
## Temporizzazione di un circuito sequenziale



- 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.

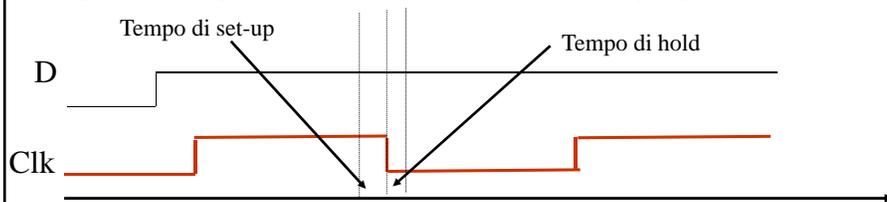


## Temporizzazione: problemi



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).
- *Tempo necessario per fare commutare l'uscita master di un flip-flop.*



**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 nel flip-flop e quindi alla logica combinatoria ( $t_p$ ).

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

Tempo di propagazione ( $t_p$ )  
(> Tempo di hold)

Tempo di set-up ( $t_s$ )

Tempo di hold ( $t_h$ )

Tempo logica combinatoria ( $t_c$ )

A.A. 2010-2011 33/35 http://homes.dsi.unimi.it/~borghese

**I bistabili: riassunto**

- Semplici elementi di memoria (1 bit)
- “Cancelli”
- Latch o flip-flop.

• I latch possono essere asincroni o sincroni.

• La differenza tra latch e flip-flop sta nel fatto che nel primo, lo stato può cambiare (anche più volte) se il clock è alto, mentre nel secondo può cambiare solo su un fronte di clock (salita o discesa secondo l’implementazione).

- Differenti tipi (S-R, D, J-K, T, sincroni o asincroni,..)
- Differenti realizzazioni (con porte NOR, NAND,...)

A.A. 2010-2011 34/35 http://homes.dsi.unimi.it/~borghese



## Sommario



I problemi dei latch trasparenti sincroni

I bistabili DT

I registri ed il register file