Architettura degli Elaboratori 2019-2020

Circuiti Sequenziali

Prof. Elisabetta Fersini
elisabetta.fersini@unimib.it

Circuiti combinatori vs sequenziali

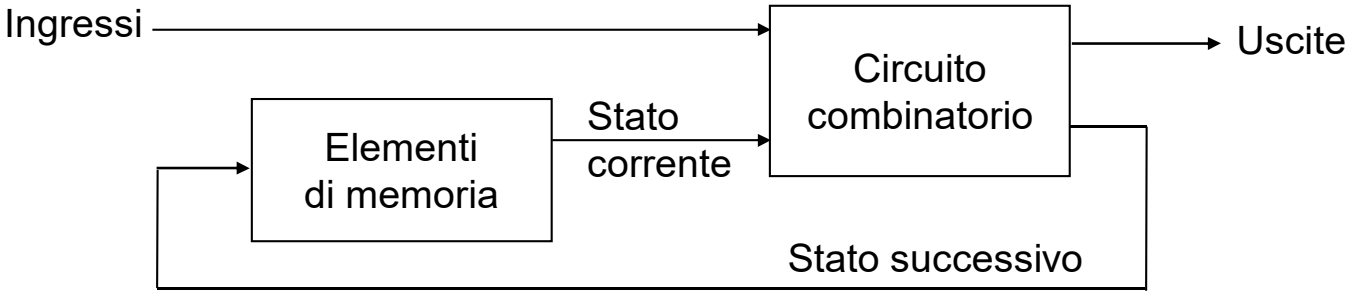
- I **circuiti combinatori** sono in grado di calcolare funzioni che dipendono **solo** dai dati in **input**
- I **circuiti sequenziali** sono invece in grado di calcolare funzioni che dipendono anche da uno **stato**
- => ovvero, che dipendono anche da informazioni memorizzate in elementi di memoria interni

Circuiti sequenziali: famiglie

- Esistono due famiglie di circuiti digitali sequenziali:
  - asincroni, non fanno uso di clock
  - sincroni, necessitano di clock
- Esempio di circuito sequenziale asincrono:
  - SR-latch
- Esempio di circuito sequenziale sincrono:
  - Flip-Flop

Circuiti sequenziali

- I circuiti sequenziali sono formati da:
  - Elementi di memoria (di vario tipo): **memorizzano** informazione
  - Reti combinatorie: **elaborano** informazione
- Un circuito sequenziale ha, in ogni dato istante, uno **stato** determinato dai bit memorizzati



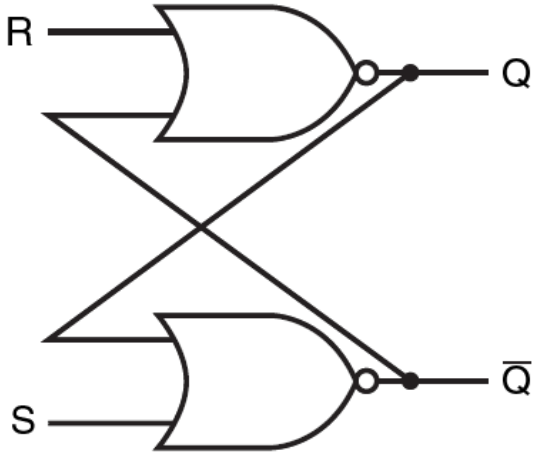
Circuiti sequenziali

- Per realizzare circuiti sequenziali è necessario un elemento di memoria per memorizzare lo stato
- Possiamo organizzare le porte logiche in modo da realizzare un elemento di memoria?
- Sì, un elemento in grado di memorizzare un singolo bit è il **latch**

S-R Latch

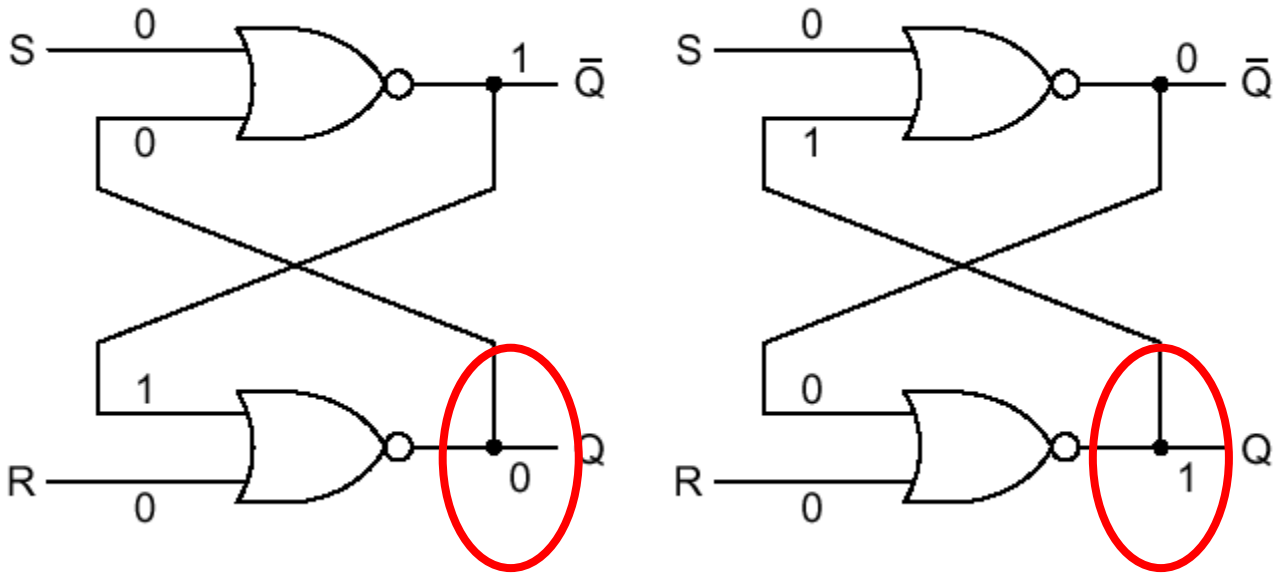
- S-R Latch è un circuito, composto da 2 porte NOR concatenate

S = Set e R = Reset.

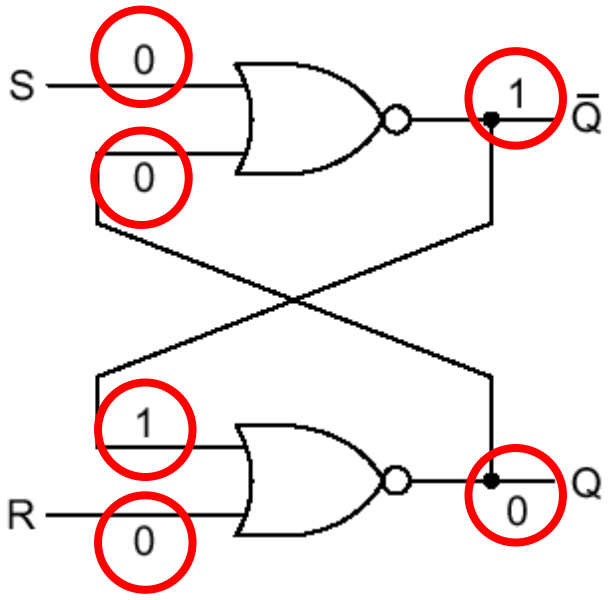


A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

S-R Latch

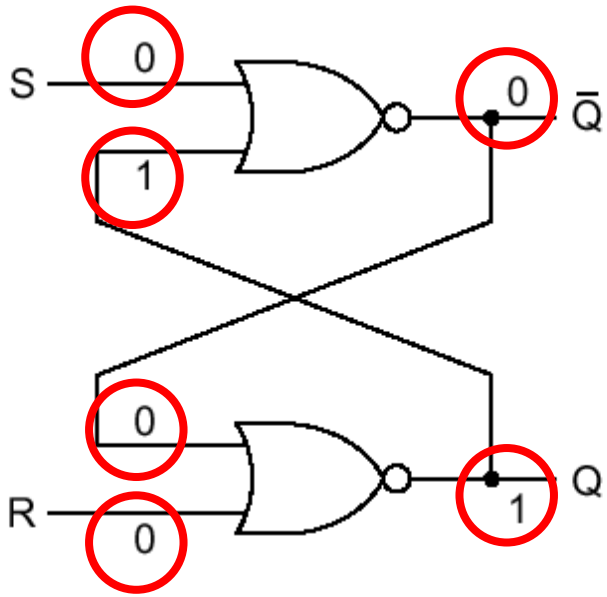


S-R Latch



Input S	Input R	Stato Interno Old Q	Output Q	Output $\bar{Q}$
0	0	0	0	1
0	0	1	1	0
0	1	1/0	0	1
1	0	1/0	1	0
1	1	1/0	1	1

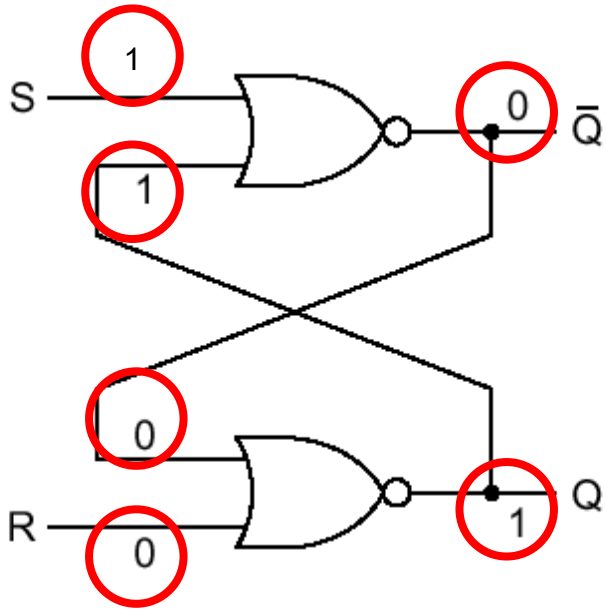
S-R Latch



Input S	Input R	Stato Interno Old Q	Output Q	Output $\bar{Q}$
0	0	0	0	1
0	0	1	1	0
0	1	1/0	0	1
1	0	1/0	1	0
1	1	1/0	1	1

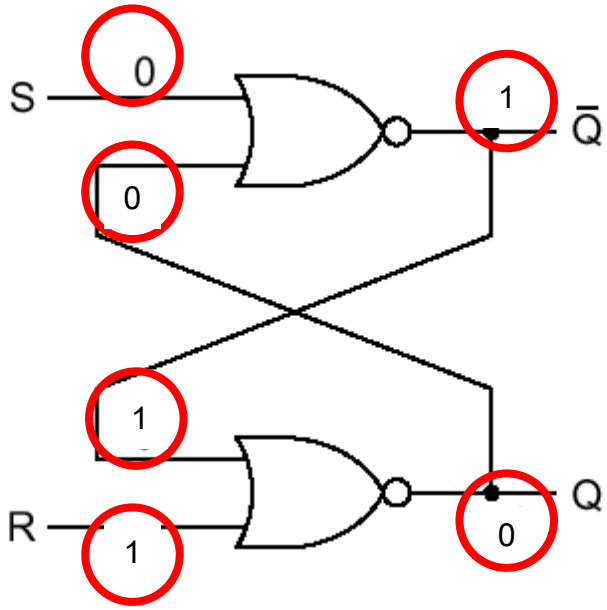
La combinazione (S,R) = (0,0) viene detta combinazione di riposo, perché semplicemente mantiene il valore memorizzato in precedenza

S-R Latch



Input S	Input R	Stato Interno Old Q	Output Q	Output $\bar{Q}$
0	0	0	0	1
0	0	1	1	0
0	1	1/0	0	1
1	0	1/0	1	0
1	1	1/0	1	1

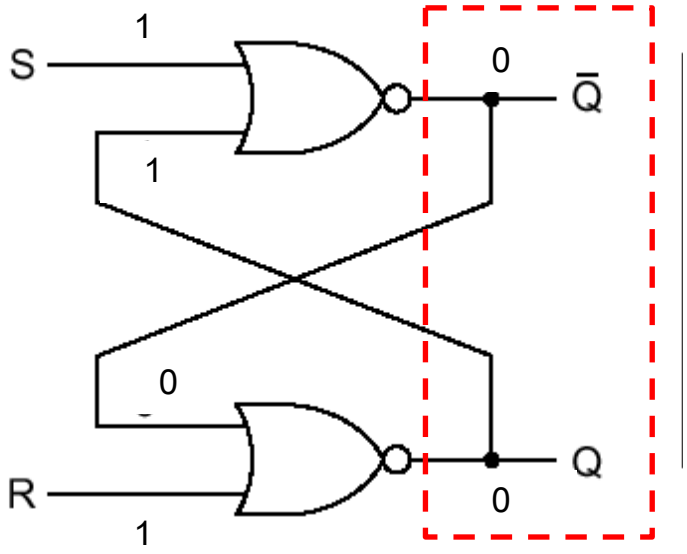
S-R Latch



Input S	Input R	Stato Interno Old Q	Output Q	Output $\bar{Q}$
0	0	0	0	1
0	0	1	1	0
0	1	1/0	0	1
1	0	1/0	1	0
1	1	1/0	1	1

S-R Latch

La configurazione di S=1 e R=1, viola la proprietà di complementarietà di Q e $\overline{Q}$, può portare ad una configurazione instabile.



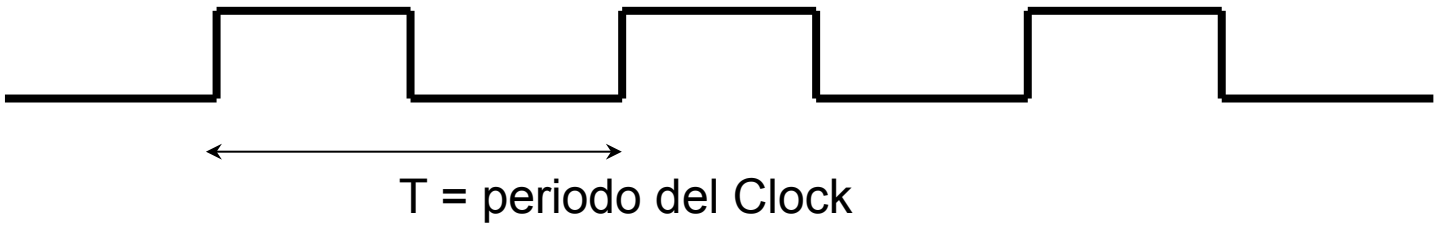
Input S	Input R	Stato Interno Old Q	Output Q	Output $\overline{Q}$
0	0	0	0	1
0	0	1	1	0
0	1	1/0	0	1
1	0	1/0	1	0
1	1	1/0	1	1

S-R Latch

- I segnali S e R devono essere stabili, e valere (1,0) o (0,1) per poter memorizzare un valore corretto
- (S,R) sono di solito calcolati da un circuito combinatorio
  - l'output del circuito diventa stabile dopo un certo intervallo di tempo
  - è possibile calcolare questo intervallo di tempo, che dipende dal numero di porte attraversate e dal ritardo delle porte
  - bisogna evitare che durante questo intervallo, gli output intermedi del circuito vengano memorizzati

Latch e clock

- Soluzione: clock
- Usiamo un segnale a gradino, il cui periodo viene scelto abbastanza grande da assicurare la stabilità degli output
- Usiamo il clock per abilitare la scrittura nei latch
  - Il clock determina il ritmo dei calcoli e delle relative operazioni di memorizzazione
- Il circuito diventa sincrono



D Latch

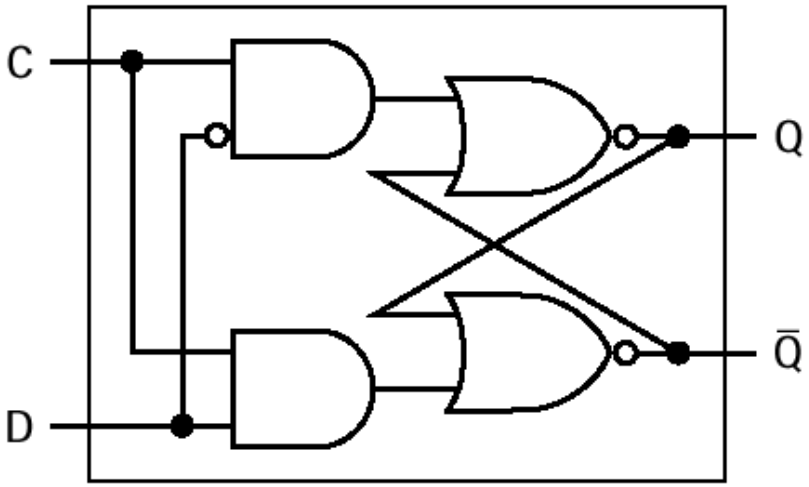
- Latch sincronizzato con il clock
- Il clock garantisce che il latch cambi stato solo in certi momenti

D=1 corrisponde al *setting*

- S=1 e R=0

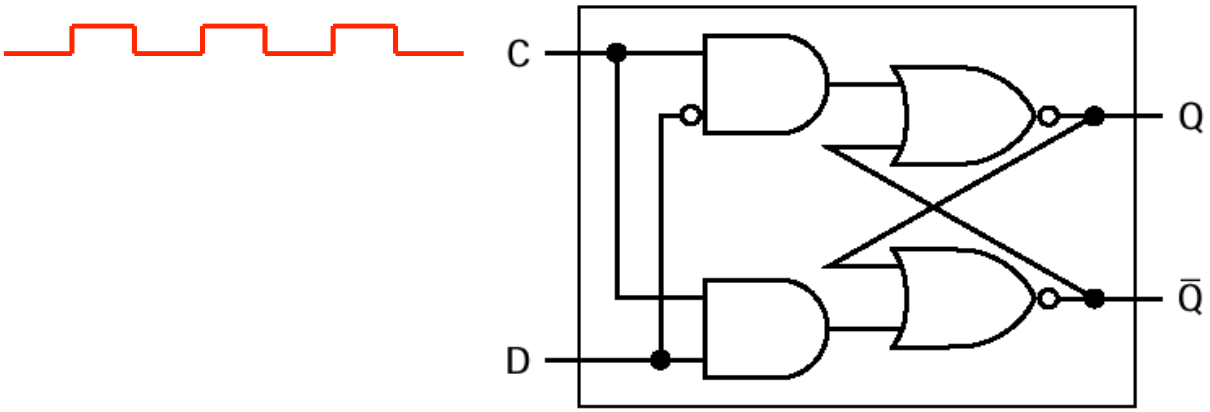
D=0 corrisponde al *resetting*

- S=0 e R=1



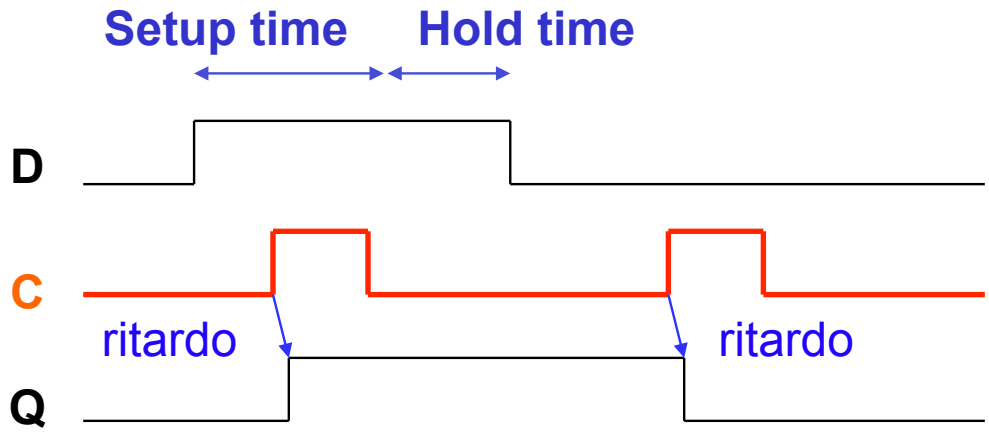
D Latch

- Quando il **clock è deasserted** non viene memorizzato nessuna valore:
  - S=0 e R=0 (viene mantenuto il valore precedentemente memorizzato)
- Quando il **clock è asserted** viene memorizzato un valore (in funzione del valore di D)



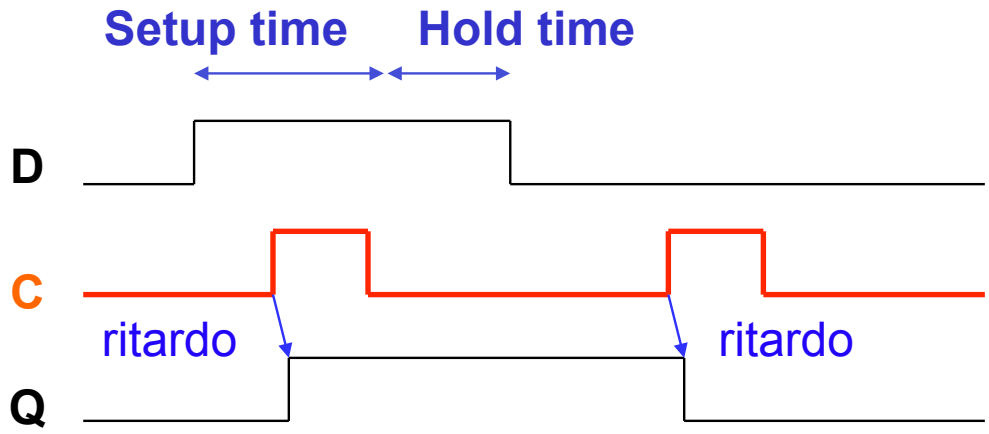
D Latch

- Il segnale D, ottenuto come output di un circuito combinatorio deve:
  - essere già **stabile** quando C diventa *asserted*
  - rimanere **stabile** per tutta la durata del livello alto di C (Setup time)
  - rimanere **stabile** per un altro periodo di tempo per evitare malfunzionamenti (Hold time)



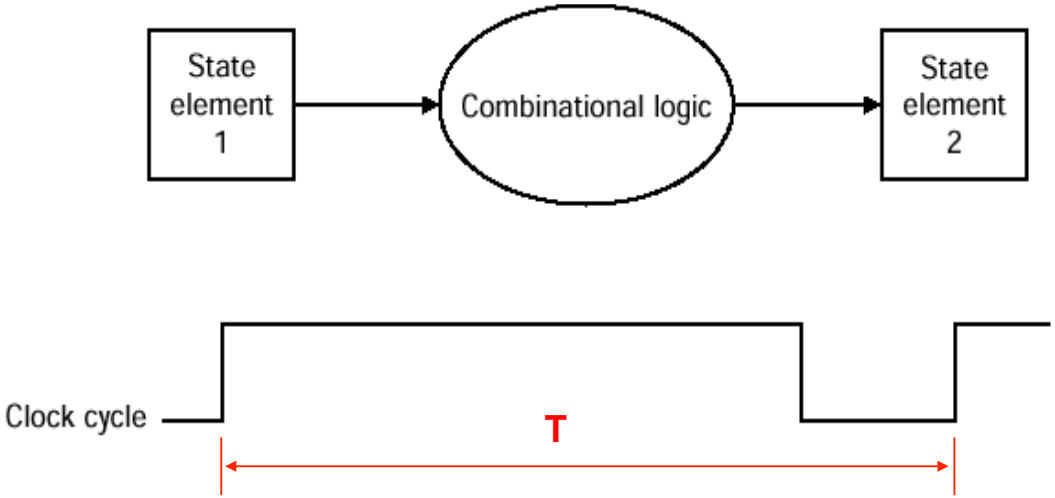
D Latch

- I circuiti reali hanno **ritardi non-nulli**
- Gli output possono temporaneamente cambiare da valori corretti a valori errati, e ancora a valori corretti ➔ **Glitch**
  - dopo un certo intervallo, con alta probabilità i segnali si stabilizzano



D Latch

- Il periodo di clock T deve essere scelto abbastanza lungo affinchè l'output del circuito combinatorio si stabilizzi
  - deve essere stabile un po' prima del periodo di apertura del latch (setup time), e lo deve rimanere per un certo tempo (hold time)



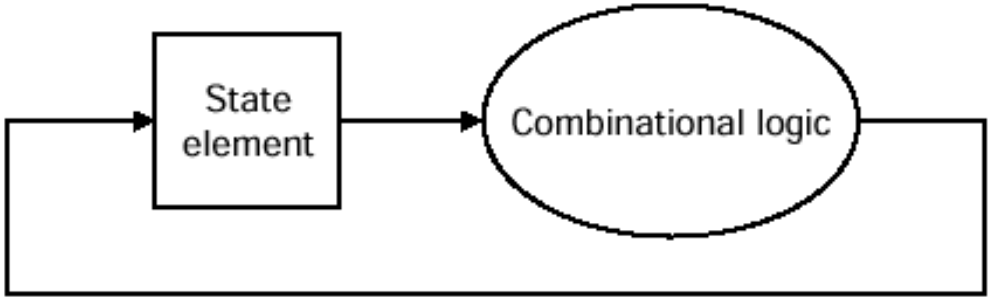
D Latch

- Il D-latch è caratterizzato dal seguente comportamento:
  - durante l'intervallo alto del clock il valore del segnale di ingresso D viene memorizzato nel latch
  - il valore di D si propaga quasi immediatamente all'uscita Q
  - anche le eventuali variazioni di D si propagano quasi immediatamente, con il risultato che Q può variare più volte durante l'intervallo alto del clock
  - solo quando il clock torna a zero Q si stabilizza
  - durante l'intervallo basso del clock il latch non memorizza

trasparenza del latch

D Latch – input/output

- Supponiamo che l'elemento di memoria debba essere usato sia come input che come output durante lo stesso ciclo di clock. Funzionerebbe il D Latch?



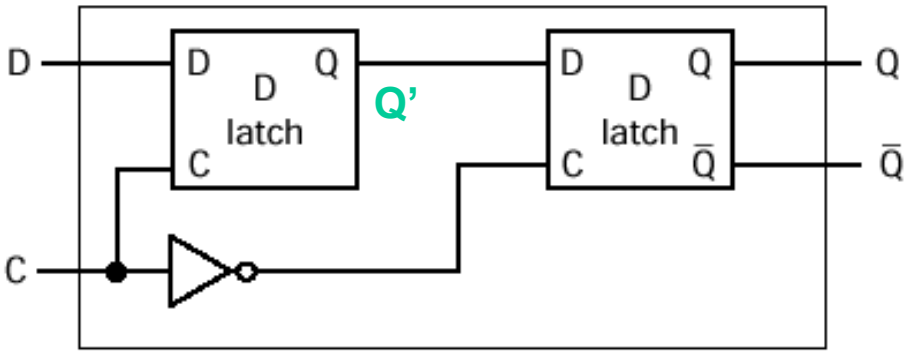
Metodologie di timing

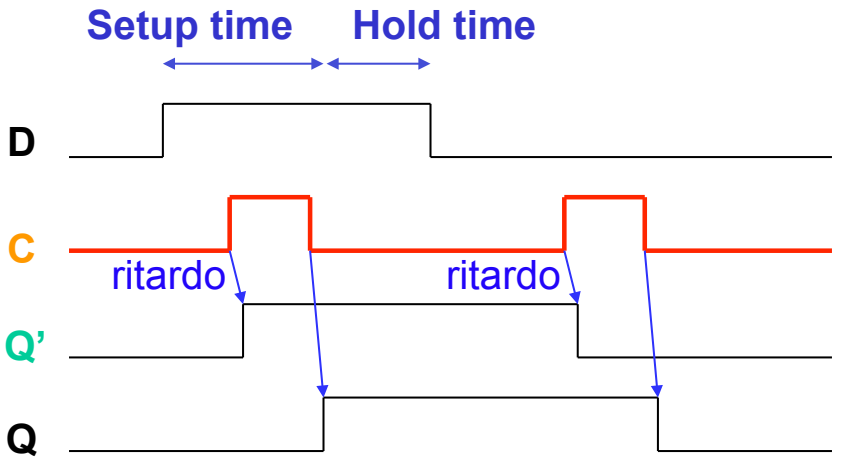
- Fino ad ora abbiamo visto una metodologia di timing detta **level-triggered** che avviene **sul livello alto** (o basso) del clock
- Esiste una metodologia di timing chiamata **edge-triggered** che avviene **sul fronte di salita** (o di discesa) del clock
  - la memorizzazione avviene istantaneamente
  - l'eventuale segnale di ritorno «sporco» non fa in tempo ad arrivare a causa dell'istantaneità della memorizzazione

flip-flop

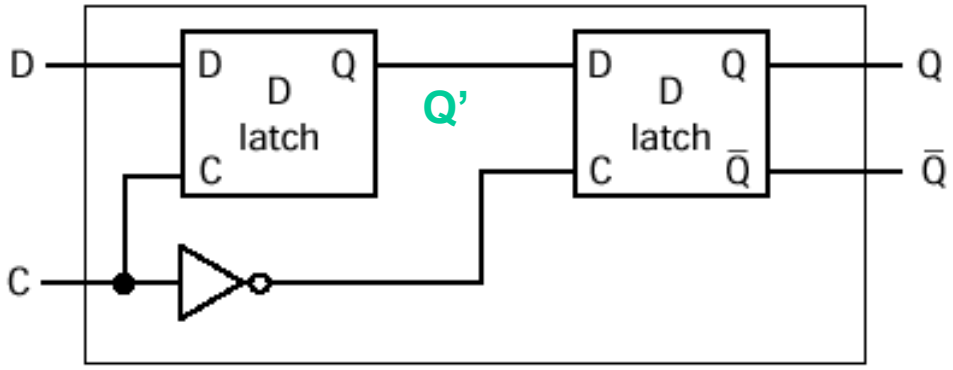
D Flip-Flop

- Il D Flip-flop può è usabile come input e output durante lo stesso ciclo di clock
- Realizzato ponendo in serie 2 D-latch: il primo viene detto **master** e il secondo **slave**



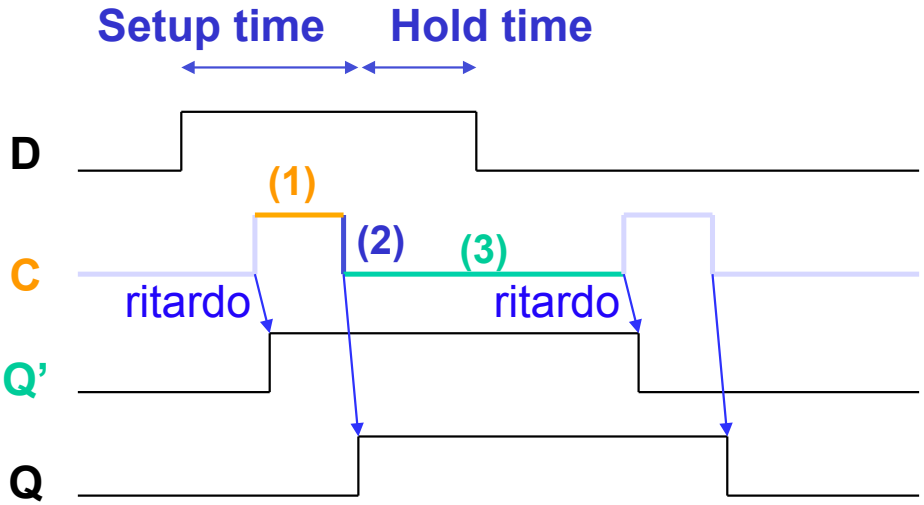


D Flip-Flop



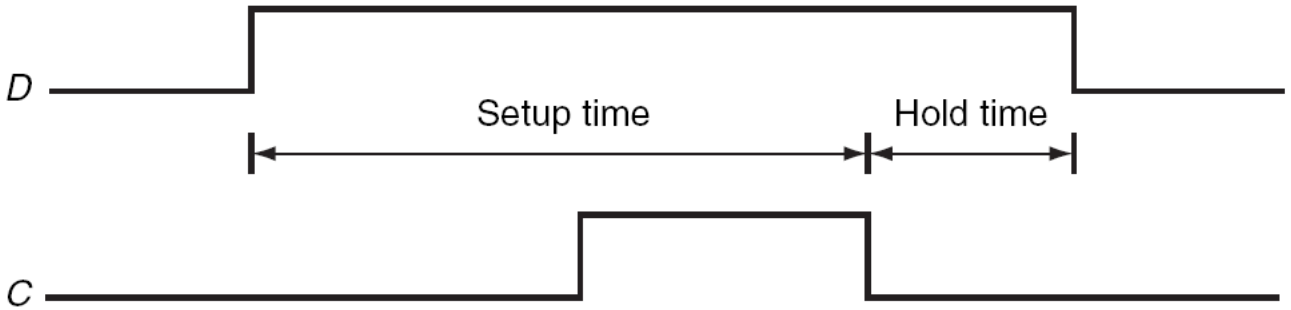
(1) Il primo latch è aperto e pronto per memorizzare D. Il valore memorizzato Q' fluisce fuori, ma il secondo latch è chiuso

- => nel circuito combinatorio a valle entra ancora il vecchio valore di Q



D Flip-Flop

- Il segnale D deve essere attivo per un periodo abbastanza lungo: setup time (prima del clock edge) + hold time (dopo il clock edge)



Nella prossima lezione

- Ora sappiamo come costruire gli elementi di memoria. Vediamo come utilizzarli per realizzare:
  - Register File
  - Memorie principali

Da leggere

- "Basics of Logic Design", appendice C del testo 4th ed. (nel testo 5th ed. questo argomento è trattato nella appendice B)
  pag. C-50 – pag. C-54