

# 4 - Circuiti sequenziali

## CIRCUITI SEQUENZIALI



I **circuiti sequenziali**, a differenza di quelli combinatori, permettono di calcolare funzioni che dipendono anche dallo **stato**, oltre che dai dati in input.

*circuito combinatorio + elementi di memoria (stato) = circuito sequenziale.*

Lo stato dipende dalle informazioni memorizzate negli elementi di memoria.

I circuiti sequenziali **sincroni**, diversamente da quelli **asincroni**, utilizzano il *clock*.

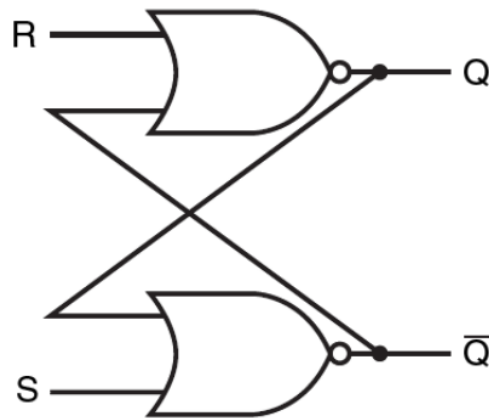
## S-R LATCH



Il **latch** è un circuito logico in grado di memorizzare un bit.



L'**S-R Latch** si compone di due porte NOR i cui output rientrano in ingresso nell'altra porta. Gli output dei due NOR devono essere **complementari** per garantire la stabilità del circuito.



S-R Latch

L'**output** (Q) rappresenta lo *stato* della macchina.

- La combinazione (0, 0) rappresenta lo **stato di riposo**, poiché conserva la Q precedente.
- La combinazione (1, 0) rappresenta lo stato di **setting**:  $Q=1$ ,  $\neg Q=0$ .
- La combinazione (0, 1) rappresenta lo stato di **resetting**:  $Q=0$ ,  $\neg Q=1$ .
- La combinazione (1, 1) **non è uno stato possibile**, dato che sarebbe *violata la proprietà di complementarità* di Q e  $\neg Q$ : non si può prevedere quale sarà la configurazione di memoria successiva.

*Esempio: se si volesse passare dalla configurazione (1, 1) allo stato di riposo, lo stato della macchina dipenderebbe da quale input si azzerava per primo (è fisicamente impossibile un azzeramento contemporaneo di S e R).*

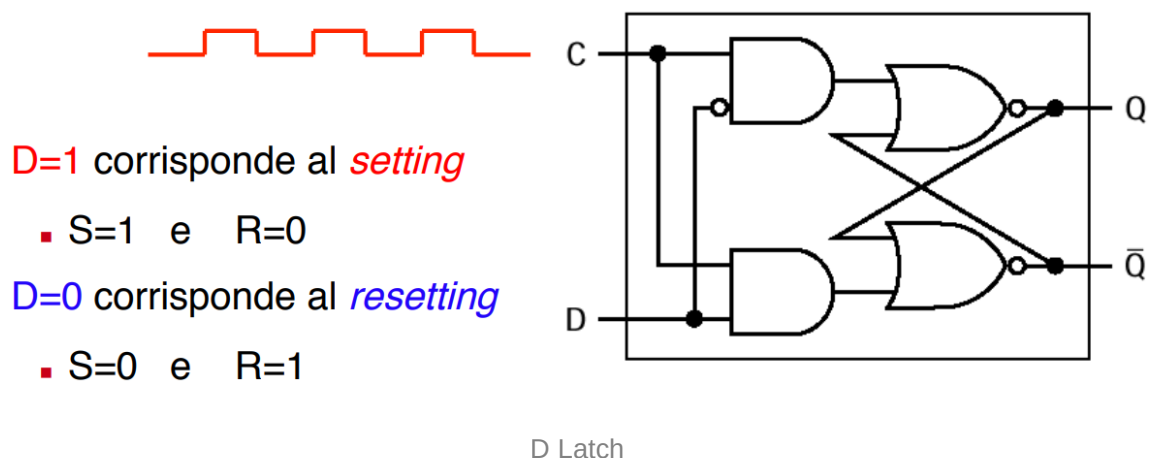
S e R sono di solito calcolati da un circuito combinatorio ed il latch da essi composto diviene **stabile** dopo un dato lasso di tempo. Tale tempo si calcola in base al **ritardo delle porte** ed al **numero delle stesse**. Durante questo lasso di tempo, i valori di output non devono essere memorizzati, in quanto non ancora stabili.

## CLOCK E D LATCH



Il **clock** è un segnale a gradino (onda quadra) di periodo abbastanza grande perché venga garantita la stabilità dei circuiti.

Sincronizzando il clock con l'S-R Latch, si ottiene il **D-Latch**, il cui stato cambia in base al clock, garantendo maggiore stabilità.



Il D Latch si basa su una metodologia di timing detta **level-triggered**:

- Quando il clock è **deasserted** (stato basso), il circuito è in *stato di riposo*.
- Quando il clock è **asserted** (stato alto), il circuito *cambia stato* in base al valore di D.

Il segnale D deve rimanere stabile prima dell'inizio dello stato asserted del clock (*setup time*), per tutta la durata dello stato alto e per un tempo successivo (*hold time*) che evita malfunzionamenti.

I segnali in output si stabilizzano dopo un dato intervallo di tempo, durante il quale i valori corretti possono mutare in valori errati, per poi ritornare corretti: tale fenomeno è detto **glitch**.

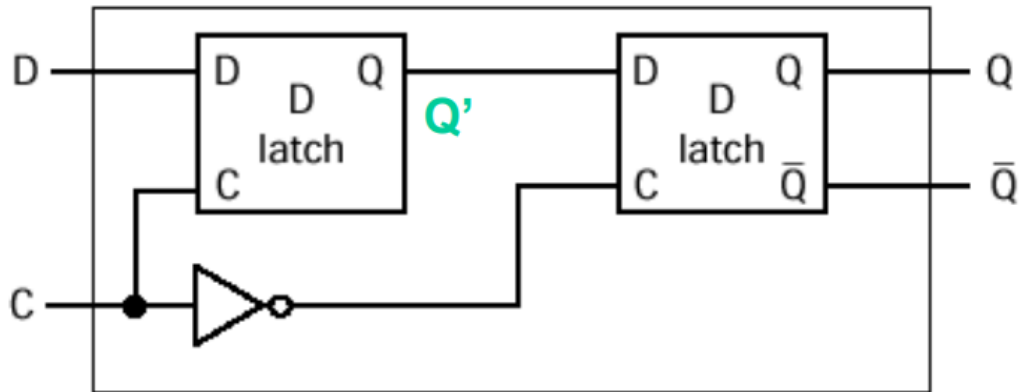
Il D Latch (e anche l'S-R Latch) è detto **trasparente** poiché qualunque cambiamento dei valori in input si propaga quasi simultaneamente ai valori in output.

## D FLIP-FLOP

Oltre alla metodologia di timing di cui sopra, esiste anche un metodo **edge-triggered**, che si basa sul fronte di discesa (*clock edge*) del segnale. La memorizzazione è, in questo caso, istantanea. Tale metodologia di timing è sfruttata dai **flip-flop**.



Il **D Flip-flop** si compone di due D Latch: nell'ordine, il *master latch* e lo *slave latch*.



D Flip-flop

Con il clock in stato *asserted*, l'output del *master latch* entra nello *slave latch*, che però è **chiuso**: alla **clock edge**, lo *slave latch* si **apre** e si ottiene l'output Q in base allo stato del *master latch*.