UNIVERSITÀ DEGLI STUDI DI TRIESTE

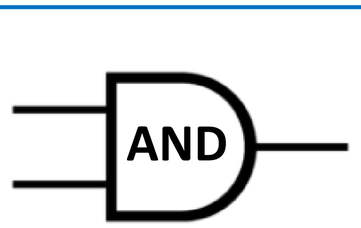
Circuiti in logica sequenziale

Prof.ssa Giulia Cisotto

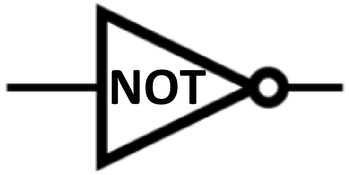
giulia.cisotto@units.it

Trieste, 19 marzo 2026

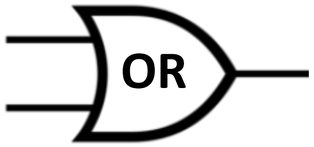
RECAP



A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

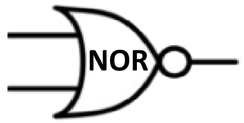


A	$\overline{A}$
0	1
1	0

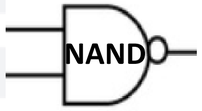


A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

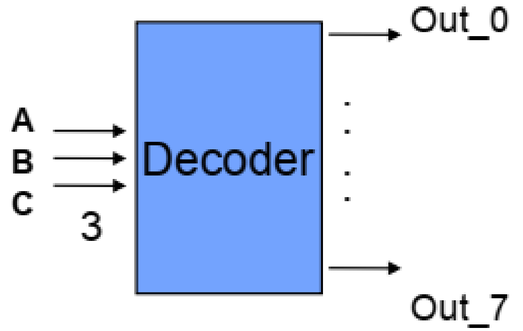
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

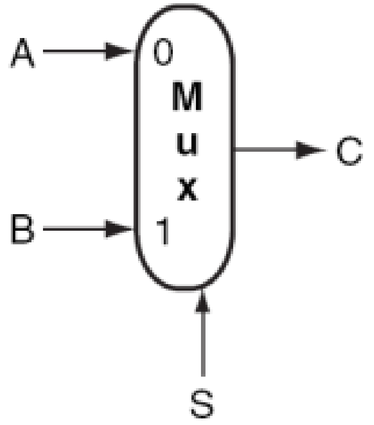


A	B	$A + B$	A NOR B
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0



A	B	$A \cdot B$	A NAND B
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0





UNIVERSITÀ DEGLI STUDI DI TRIESTE

RECAP

Qualunque funzione logica può essere costruita usando le porte logiche AND, OR e NOT.

Algebra booleana

N°	Nome	Proprietà
1	**Proprietà associativa**	$a + (b + c) = (a + b) + c;\ a \cdot (b \cdot c) = (a \cdot b) \cdot c$
2	**Proprietà commutativa**	$a + b = b + a;\ a \cdot b = b \cdot a$
3	**Proprietà distributiva**	$a + (b \cdot c) = (a + b) \cdot (a + c)$ $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$
4	**Assioma dell'annullamento**	$a \cdot 0 = 0;\quad a + 1 = 1$
5	**Assioma del Complemento**	$(a + \bar{a}) = 1;\quad (a \cdot \bar{a}) = 0$
6	**Assioma dell'idempotenza**	$a \cdot a = a;\quad a + a = a$
7	**Assioma doppia negazione**	$\bar{\bar{a}} = a$
8	**Teorema di DeMorgan**	$a + b = \overline{\bar{a} \cdot \bar{b}}\ ;\ a \cdot b = \overline{\bar{a} + \bar{b}}$
9	**Teorema dell'assorbimento**	**Se** $Y = a + ab$ **allora** $Y = a$
10	**Teorema del consenso**	**Se** $Y = ab + \bar{a}c + bc$ **allora** $Y = ab + \bar{a}c$

Il **Teorema dei Mintermini** afferma che **ogni funzione booleana può essere espressa come una somma (operazione OR) di mintermini**. Un **mintermine** è un prodotto (operazione AND) di tutte le variabili della funzione, ciascuna presente in forma diretta o negata, che rendono la funzione uguale a 1.

Il **Teorema dei Maxtermini** afferma che **ogni funzione booleana può essere espressa come un prodotto (operazione AND) di maxtermini**. Un **maxtermine** è una somma (operazione OR) di tutte le variabili della funzione, ciascuna presente in forma diretta o negata, che rendono la funzione uguale a 0.

RECAP

MULTIPLEXER

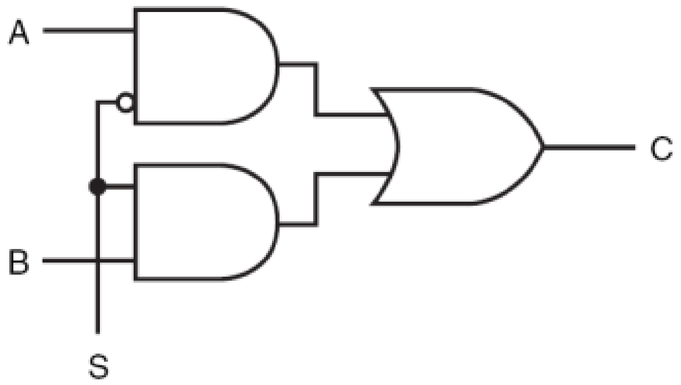
Esempio 1

Funzione logica

$$C = (A \cdot \bar{S}) + (B \cdot S)$$

Qual è la tabella di verità di questo circuito?

Circuito logico equivalente



S	A	B	C
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

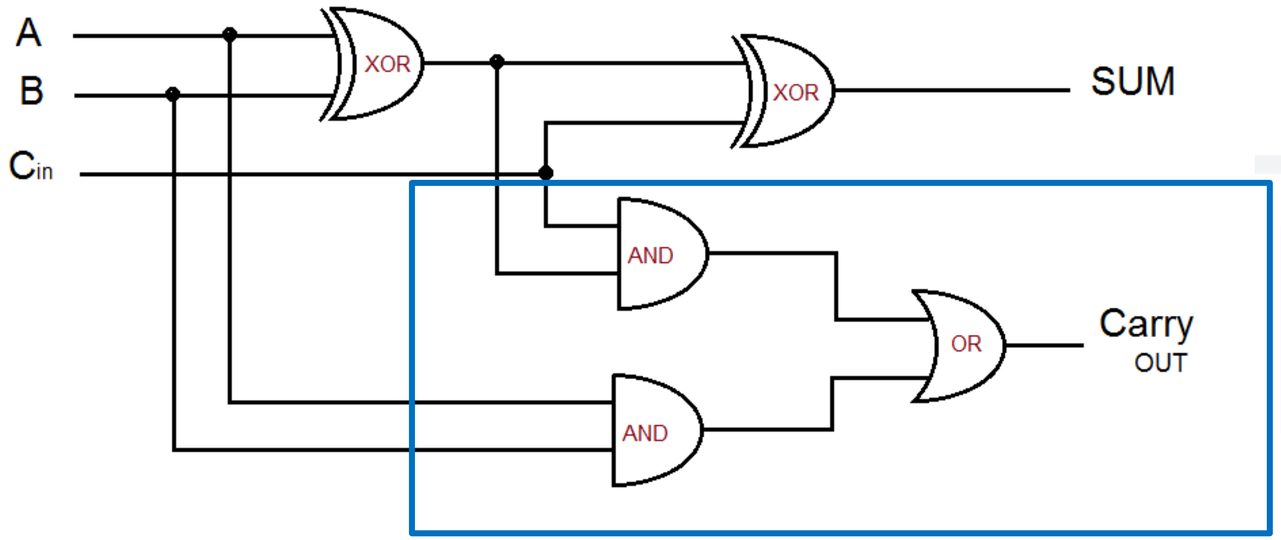
S ha la funzione di segnale di controllo

RECAP

IL PRIMO CIRCUITO LOGICO

Full adder

Inputs			Outputs	
A	B	C_{in}	S	C_{out}
0	0	0		0
0	0	1		0
0	1	0		0
0	1	1		1
1	0	0		0
1	0	1		1
1	1	0		1
1	1	1		1



Questo circuito realizza la **SOMMA BINARIA** tra due bit (con riporto).

SOLUZIONE

IL PRIMO CIRCUITO LOGICO

Full adder: circuito che realizza la SOMMA BINARIA tra due bit (con riporto).

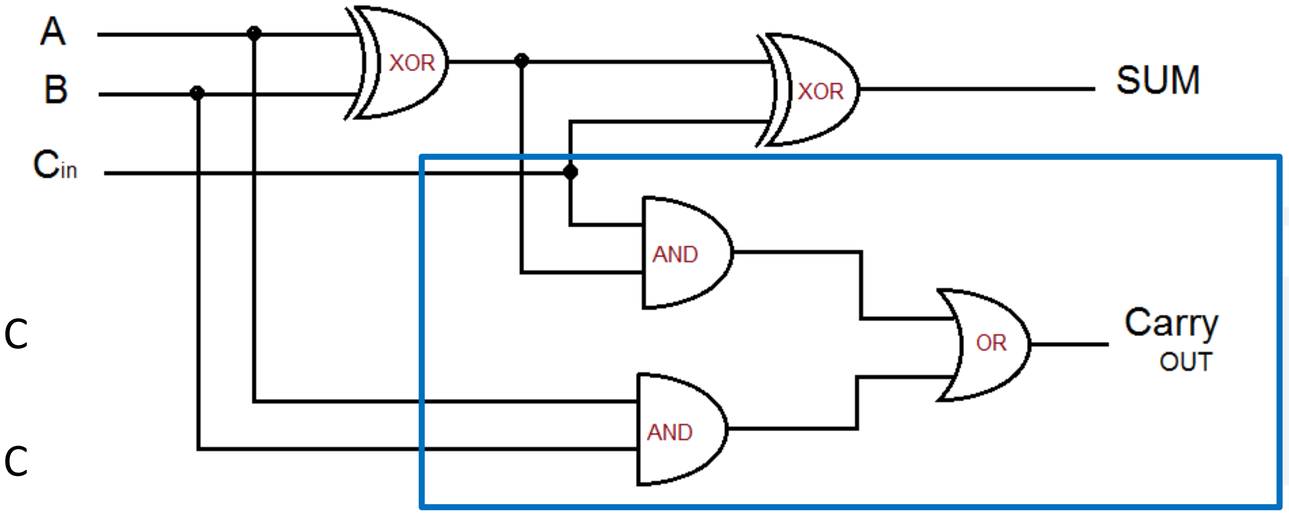
Inputs			Outputs	
A	B	C_{in}	S	C_{out}
0	0	0		0
0	0	1		0
0	1	0		0
0	1	1		1
1	0	0		0
1	0	1		1
1	1	0		1
1	1	1		1

not(A) · B · C

A · not(B) · C

A · B · not(C)

A · B · C



C_{out} = not(A) · B · C + A · not(B) · C + A · B · not(C) + A · B · C =

= (not(A) · B + A · not(B)) · C + A · B · (not(C) + C) =

= (**A** ⊕ **B**) · C + A · B · 1 = (**A** ⊕ **B**) · C + A · B

UNIVERSITÀ DEGLI STUDI DI TRIESTE

CIRCUITI COMBINATORI VS SEQUENZIALI

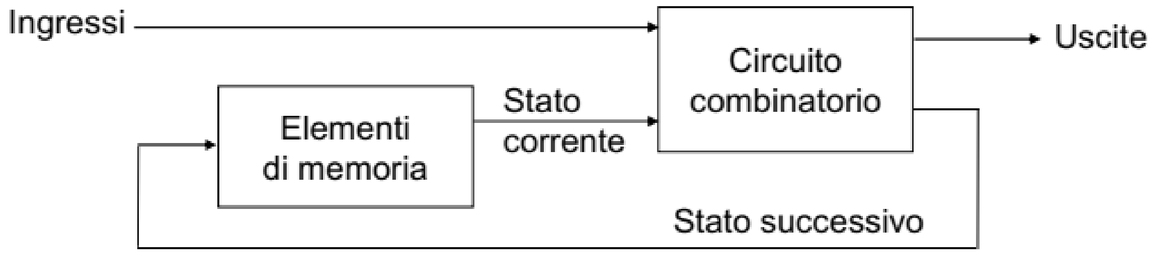
- I circuiti combinatori sono in grado di calcolare funzioni che dipendono solo dai dati in input
- I circuiti sequenziali sono invece in grado di calcolare funzioni che dipendono anche da uno stato
- => ovvero, che dipendono anche da informazioni memorizzate in elementi di memoria interni

CIRCUITI SEQUENZIALI

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?

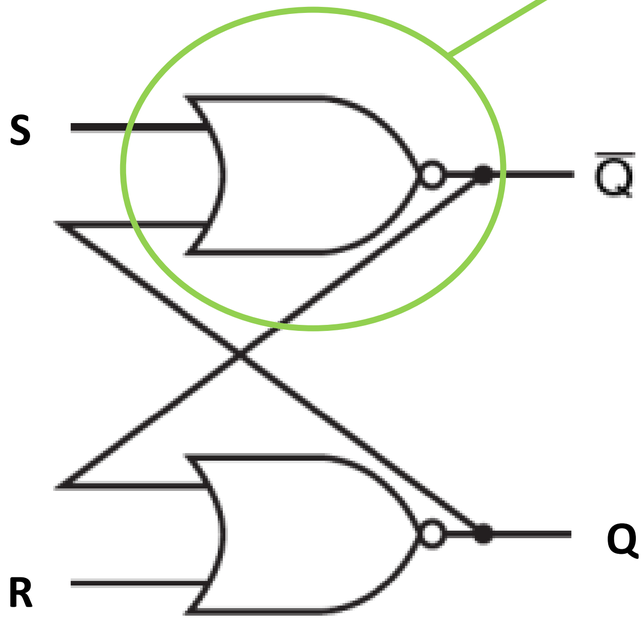
Sì, un elemento in grado di memorizzare un singolo bit è il latch

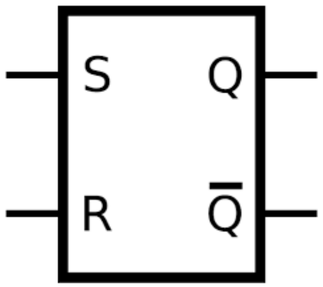
S-R LATCH

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

S = Set e R = Reset

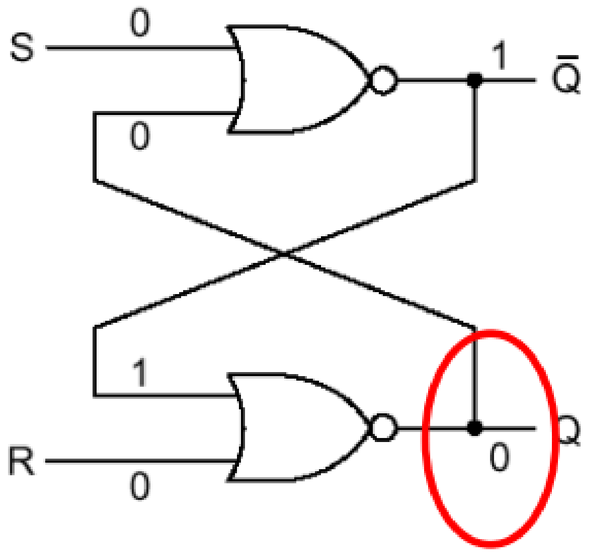
A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0





S-R LATCH

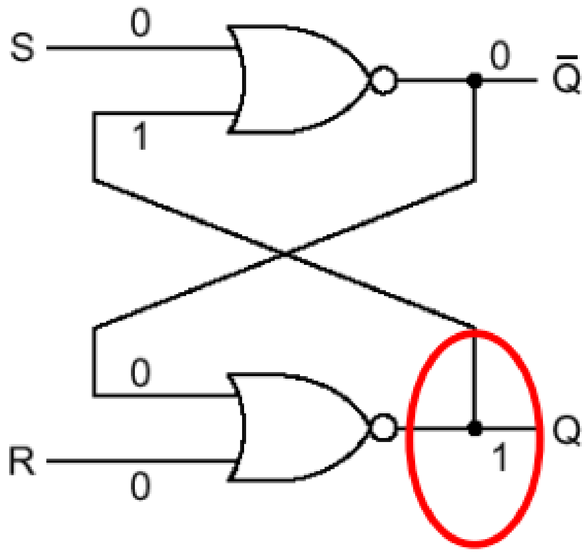
Ipotizziamo che lo stato corrente Q fosse a 0 e che gli input (S,R) siano entrambi 0.



In questa configurazione degli input (S=0, R=0), l'uscita Q non cambia.

S-R LATCH

Ipotizziamo che lo stato corrente Q fosse a 1 e che gli input (S,R) siano entrambi 0.



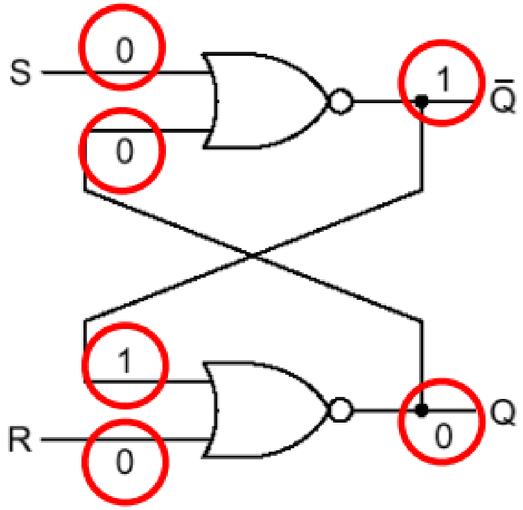
In questa configurazione degli input (S=0, R=0), l'uscita Q non cambia.

Quando S=0, R=0, viene mantenuto lo stato di prima (Q e notQ rimangono invariati)

S-R LATCH

Tabella di verità per un circuito sequenziale!

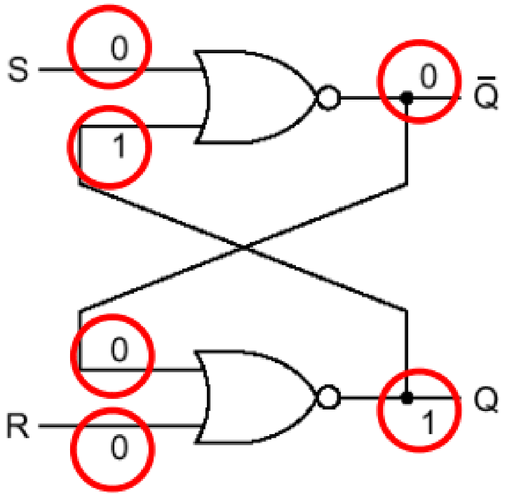
S=0, R=0



S-R LATCH

Tabella di verità per un circuito sequenziale!

S=0, R=0

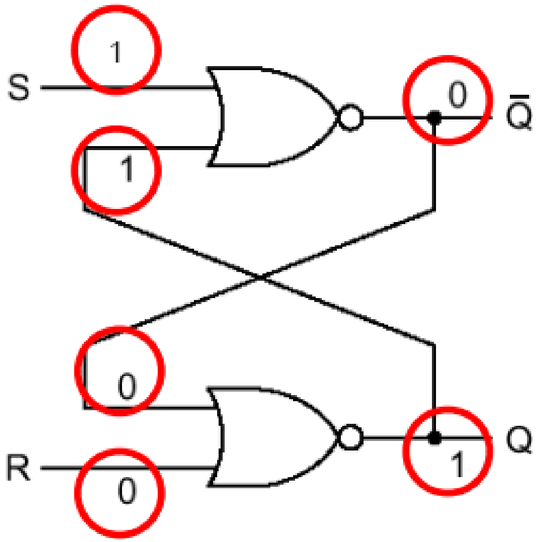


Input S	Input R	Stato Interno Old Q	Output Q	Output $\bar{Q}$
0	0	0	0	1
0	0	1	1	0

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

S-R LATCH

S=1, R=0

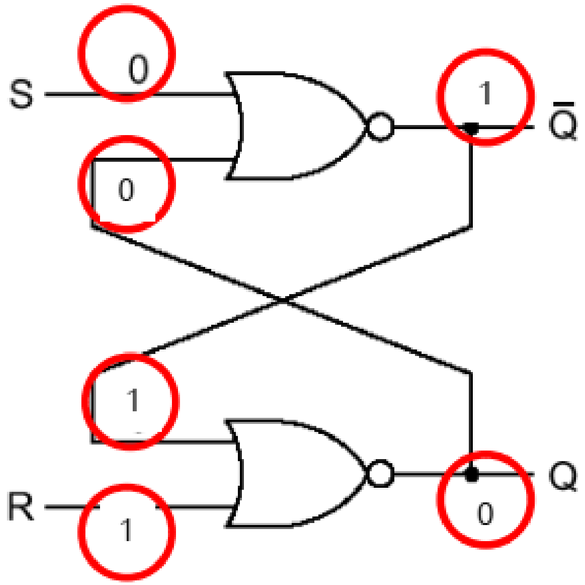


Input S	Input R	Stato Interno Old Q	Output Q	Output $\bar{Q}$
0	0	0	0	1
0	0	1	1	0
1	0	1/0	1	0

La combinazione (S,R) = (1,0) viene detta combinazione di SET, perché Q viene messo a 1.

S-R LATCH

S=0, R=1

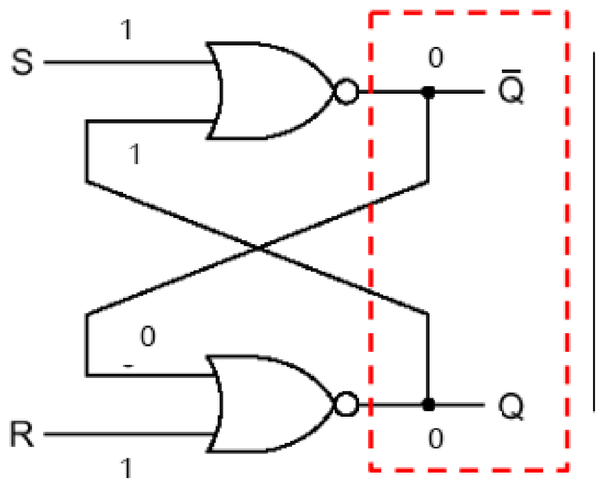


Input S	Input R	Stato Interno Old Q	Output Q	Output $\bar{Q}$
0	0	0	0	1
0	0	1	1	0
0	1	1/0	0	1
1	0	1/0	1	0

La combinazione (S,R) = (0,1) viene detta combinazione di ReSET, perché Q viene messo a 0.

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

S-R LATCH

(S,R) sono di solito calcolati da un circuito combinatorio

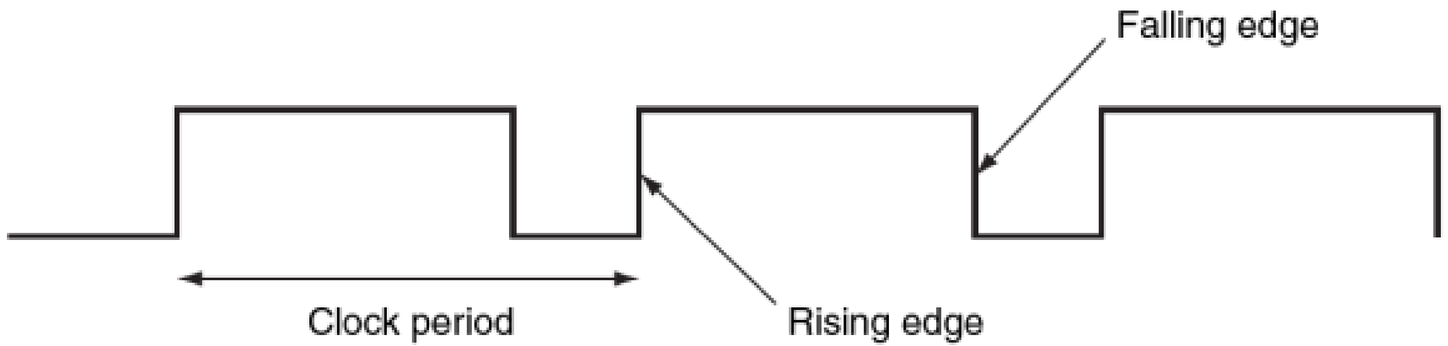
- l'output del circuito diventa stabile dopo un certo intervallo di tempo che dipende dal numero di porte attraversate
- bisogna evitare che durante questo intervallo, gli output intermedi del circuito vengano memorizzati

Soluzione: clock

CLOCK

Il segnale di clock è fondamentale per le reti sequenziali che sono caratterizzate da uno stato

Clock – definito come un segnale (onda quadra) con un periodo predeterminato e costante



Caratterizzato da un periodo T (oppure ciclo) di clock e dalla frequenza F(definita come 1/periodo) e misurata in Hertz F= 1/T.

LATCH E CLOCK

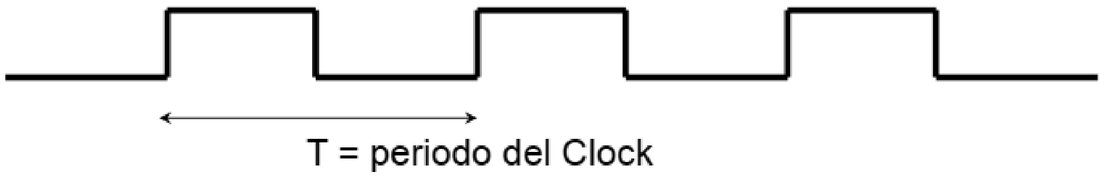
Soluzione: clock

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.



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:

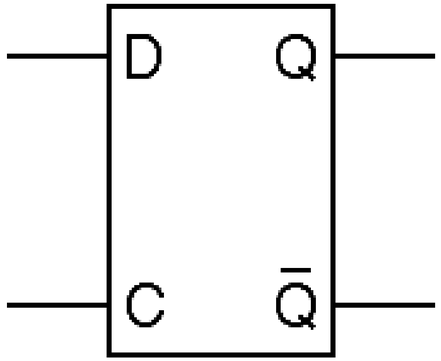
- SR-latch

Esempio di circuito sequenziale sincrono:

- D-latch, Flip-Flop

Errata corrige

D LATCH



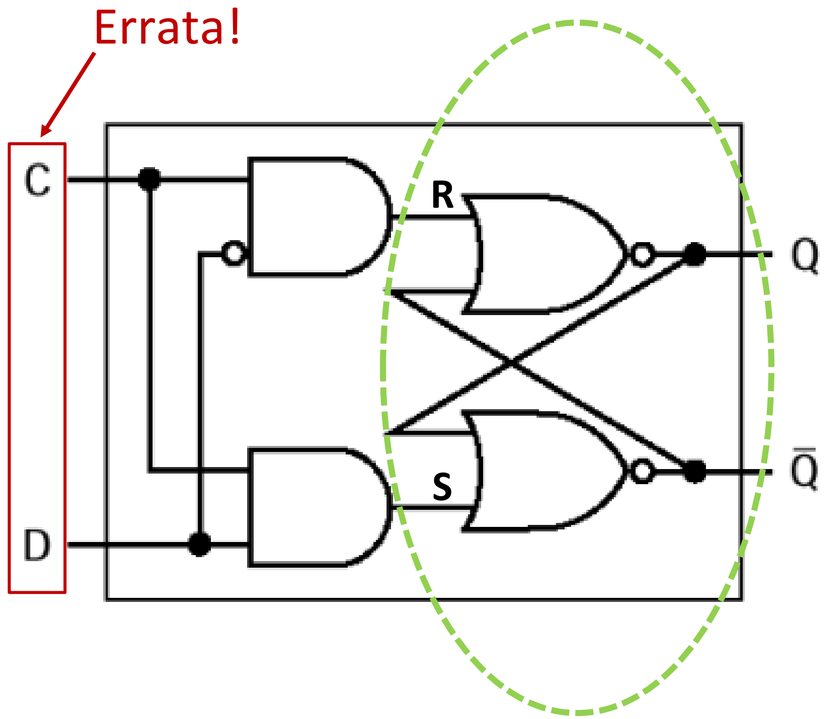
- Latch sincronizzato con il clock
- Il clock garantisce che il latch cambi stato solo in certi momenti

D=1 corrisponde al *setting*

- S=1 e R=0

D=0 corrisponde al *resetting*

- S=0 e R=1



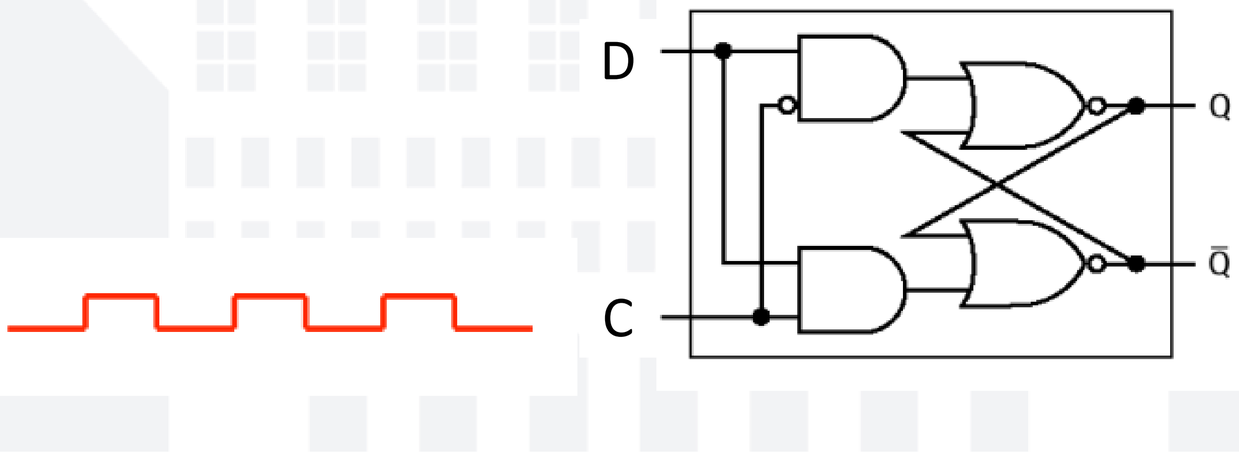
Diventa quindi *impossibile* trovarsi nella situazione S=R=1.

D LATCH

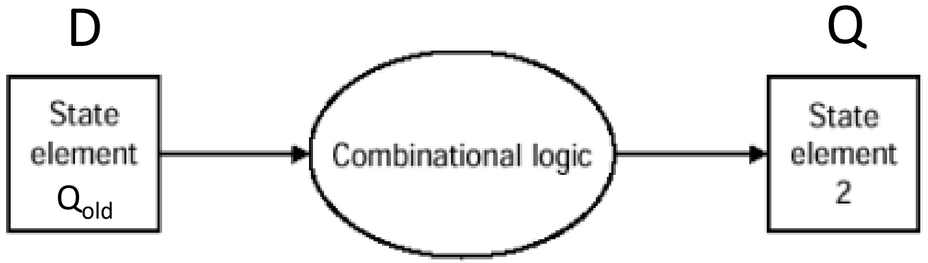
Quando il clock è deasserted non viene memorizzato nessuna valore:

- 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)

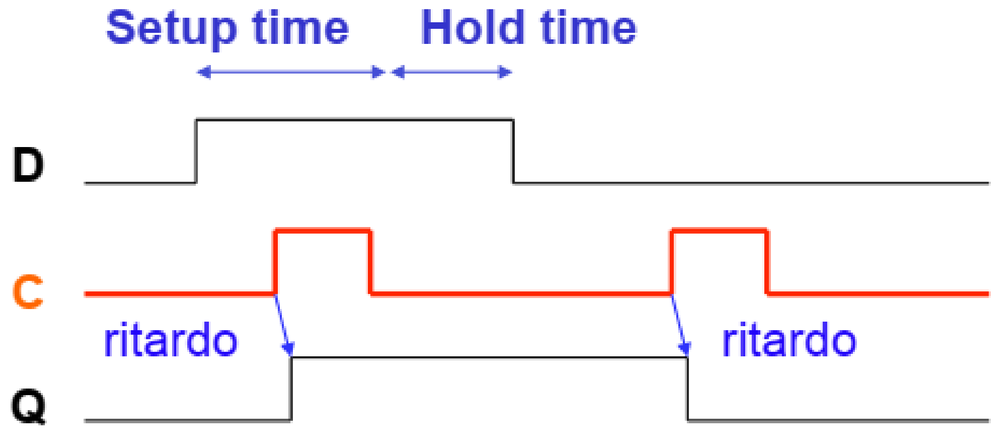


D LATCH



Il segnale D 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*)



Il periodo di clock T (ovvero il periodo tra i due fronti di salita del clock) deve essere scelto abbastanza lungo affinché l'output del circuito combinatorio si stabilizzi.

D LATCH

Il D-latch è caratterizzato dal seguente comportamento:

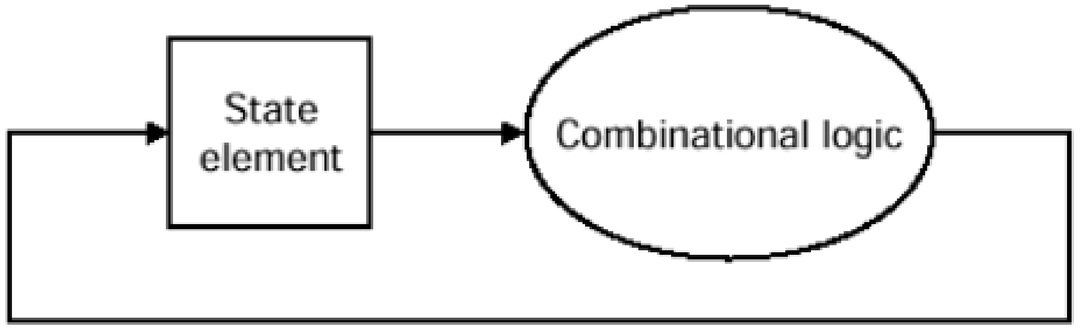
trasparenza del latch

- 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

- durante l'intervallo basso del clock il latch non memorizza
- solo quando il clock torna a zero Q si stabilizza
- se il segnale cambia troppo vicino al fronte di discesa del clock (C che torna a 0), il latch può non stabilizzarsi correttamente (metastabilità).

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?



Durante lo stesso ciclo (C=1):

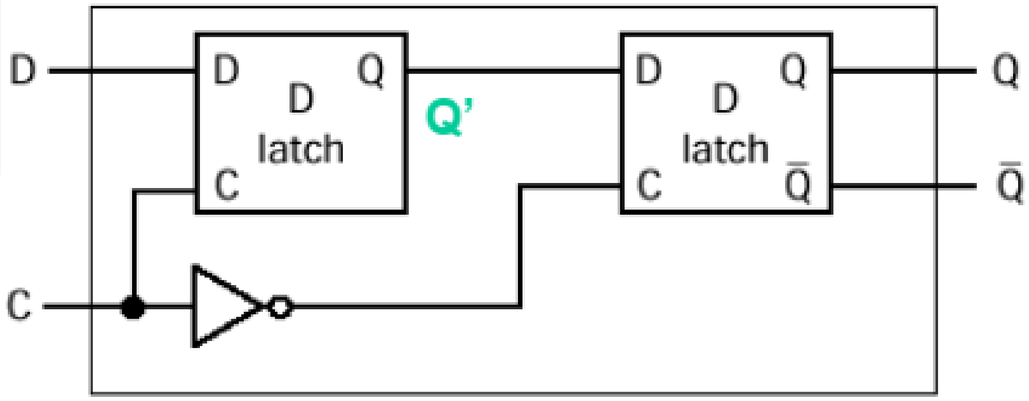
- nel D-latch l'uscita dipende subito dall'ingresso
- se quell'uscita rientra come input → si crea una dipendenza immediata (feedback)

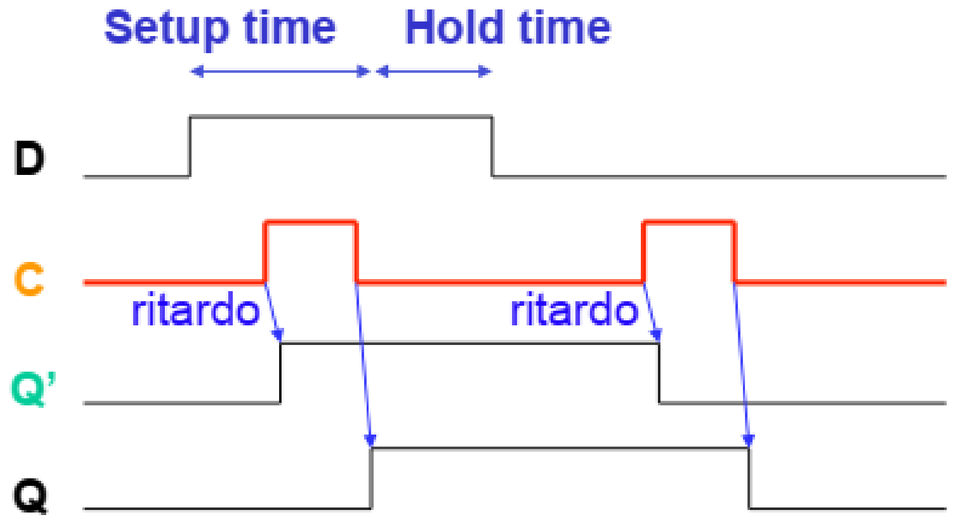
Conseguenza:

- il valore può cambiare più volte nello stesso ciclo
- *comportamento instabile / non controllato*

D FLIP-FLOP

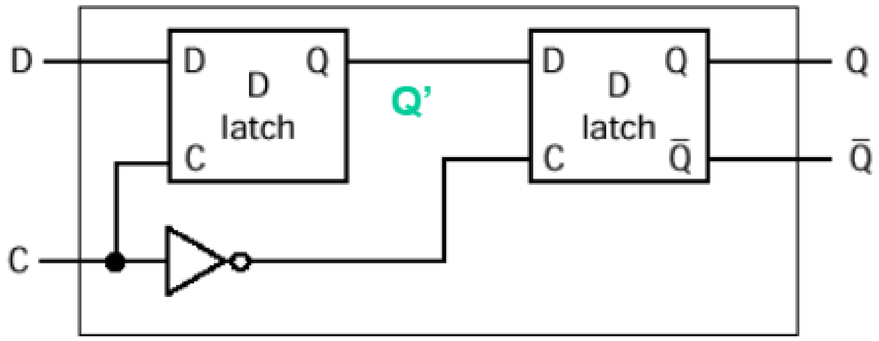
- Il D-latch è “troppo reattivo” nello stesso ciclo
  ➔ **serve il D flip-flop per avere un aggiornamento discreto e stabile**
- Realizzato ponendo in serie 2 D-latch: il primo viene detto master e il secondo slave.
- Legge l’ingresso solo al fronte di clock
- Mantiene l’uscita costante per tutto il ciclo

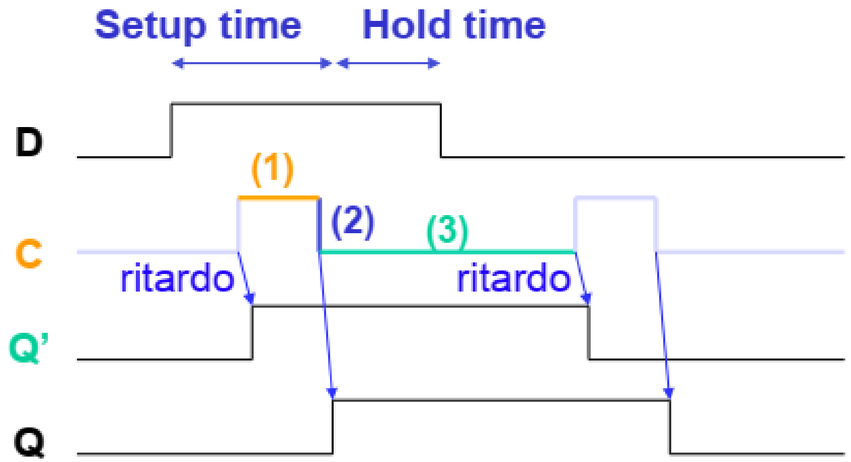




Nota. Si sfrutta sia il fronte di salita del clock che quello di discesa. Questo permette di **separare le fasi** di acquisizione del nuovo dato (D→Q’) e di aggiornamento dello stato (Q’ → Q)

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.

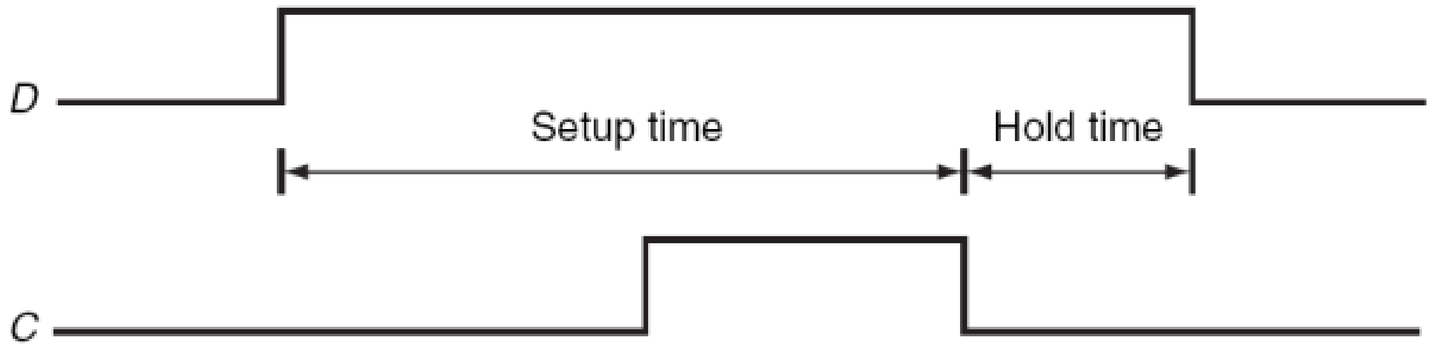
(2) Il segnale del clock scende, e in questo istante il secondo latch viene aperto per memorizzare il valore di Q'.

(3) Il secondo latch è aperto, memorizza D (Q'), e fa fluire il nuovo valore Q nel circuito a valle. Il primo latch è invece chiuso, e non memorizza niente.

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)



Calendario fino a Pasqua

- 24/03/2026: lezione
- 25/03/2026: lezione + lab (Assembly, procedure annidate)
- 26/03/2026: lezione/lab (da confermare)

- *31/03/2026: free (libero per preparazione al parziale)*
- *01/04/2026: free (libero per preparazione al parziale)*
- *08/04/2026: free (libero per preparazione al parziale)*
- 09/04/2026: PRIMO PARZIALE

- 14/04/2026: lezione

Prossima lezione: 24 marzo, h.14:00, aula 4C

Materiale per la lezione

- Patterson & Hennessy, Appendice B
- Patterson & Hennessy, §4.2

Prossima lezione: 24 ***marzo, h.14:00****, aula 4C*