RETI SEQUENZIALI/CIRCUITO LATCH

martedì 1 novembre 2022 17:44

RETI SEQUENZIALI

I CIRCUITI VISTI FINO AD ORA POSSONO ESSERE definiti combinatori, PERCHÈ dato UN INPUT elaborano UN OUTPUT e quindi ho un certo numero di variabili in INPUT e viene calcolato uno o più OUTPUT.
Questo calcolo dipende anche dall'INPUT.

COSA SUCCEDE SE CAMBIO L'INPUT? CAMBIERÀ L'OUTPUT.

SE questo è l'unico modo per realizzare i nostri circuiti, ci manca una cosa fondamentale, OSSIA LA MEMORIA.
PERCHÈ SE IO ho dei circuiti che dato UN INPUT calcolano un OUTPUT, E CAMBIA L'INPUT, CAMBIERÀ anche l'OUTPUT.

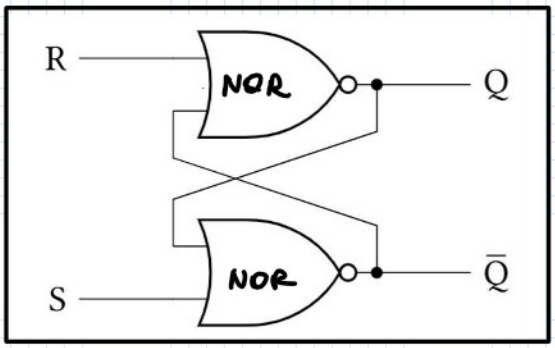
IO VOGLIO IMPLEMENTARE anche UNA memoria.
VOGLIO MEMORIZZARE UNA MOLE di INFORMAZIONI

POSSIAMO IMMAGAZZINARE UN CIRCUITO LATCH, UN CIRCUITO che UTILIZZA UN ANELLO di RETROAZIONE.

CIRCUITO LATCH

- È un circuito analogico che consente di immagazzinare un'informazione digitale

HO L'INPUT CHE DIPENDE ANCHE DALL'OUTPUT:



S	R	Q	Q' SAVE
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	indeterminato
1	1	1	indeterminato

a seconda dei valori che fornisco in INPUT, l'OUTPUT potrà cambiare.

NOR

x_1	x_2	y
0	0	1
0	1	0
1	0	0
1	1	0

CATTURA ogni valore fornito in INPUT.
UN CIRCUITO di questo TIPO fa girare l'informazione dall'output all'INPUT.
Riesce a MEMORIZZARE LE INFORMAZIONI che gli do, UN SINGOLO BIT, GRAZIE alla CATENA DI RETROAZIONE.

RES (RESET E SET) SONO INPUT.
Q E $\bar{Q}$ (Valore del BIT di INFORMAZIONE che vado a memorizzare all'INTERNO del LATCH, $\bar{Q}$ è il negato del BIT che vado a salvare)

UN CIRCUITO di QUESTO TIPO, EVOLVE NEL TEMPO, IN FUNZIONE dello STATO PRECEDENTE IN CUI ESSO SI TROVAVA;

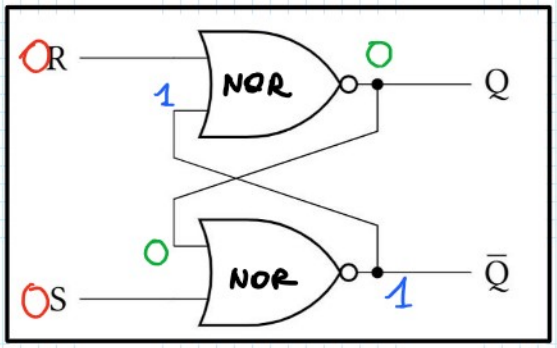
QUESTE RETI MANTENGONO L'INFORMAZIONE FISSATA.

POICHÉ ho delle linee che TORNANO INDIETRO dall'output all'INPUT, il valore di OUTPUT può cambiare nel tempo se si VERIFICA una determinata condizione, quindi diventa il NUOVO valore di INPUT.

SE MODIFICO l'INPUT OTTERRÒ UN NUOVO OUTPUT, che dipende dall'INPUT e dallo STATO PRECEDENTE.

$Q \in \{0,1\}$.

SE $q=0$, COSA STO FORNENDO IN INPUT?



S	R	Q	Q' SAVE
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	indeterminato
1	1	1	indeterminato

NOR

x_1	x_2	y
0	0	1
0	1	0
1	0	0
1	1	0

SE FISSATO $S=0, R=0$ e [$q=0$], questa è la configurazione.

↳ dato memorizzato

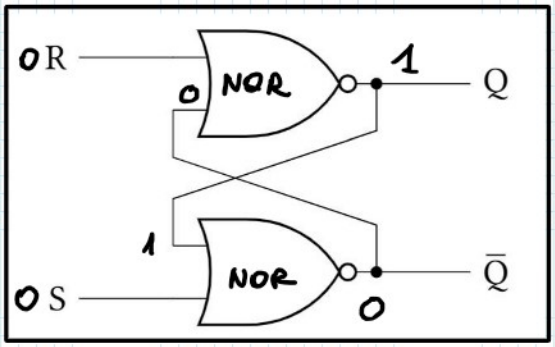
11:52

IL MIO STATO Q che era 0, SE applico un INPUT 0 0 MI FORNISCE ANCORA IN USCITA 0!

↓

NUOVO STATO

SE $q=1$



SI INVERTE.

L'USCITA SARÀ di NUOVO 1.

(questa è la seconda riga)

SE IO FORNISCO COME INPUT 00, IL MIO Q NON CAMBIA È SEMPRE QUELLO!

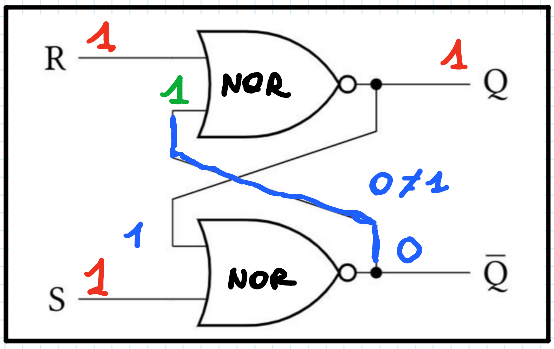
STO MEMORIZZANDO UN BIT di informazione.

QUINDI MANTIENE L'INFORMAZIONE che ha già memorizzato, SE IN INPUT FORNISCO 0 e 0.

SE SET VALE 1 e RESET VALE ZERO, IO FORZERÒ IL valore da salvare a 1. SE RESET VALE 1 e SET VALE ZERO IO andrò a forzare il valore del CIRCUITO a 0.

SE IMPOSTO 0 A SET e 1 a RESET, indipendentemente dallo STATO PRECEDENTE (Q) questo CIRCUITO commuterà a ZERO, e ME LO SEGNO.

SE SET E RESET VALGONO ENTRAMBI UNO SPOSTO IL MIO CIRCUITO IN UNO STALLO di OSCILLAZIONE indeterminato, 0 e 1, quando?



NOR

x_1	x_2	y
0	0	1
0	1	0
1	0	0
1	1	0

IL MIO SEGNALE NON SI STABILIZZERÀ MAI.
OSCILLERÀ.

UN CIRCUITO di questo TIPO HA QUESTO PROBLEMA, OSCILLA, NON POSSO UTILIZZARE gli INPUT.

Possiamo INVENTARCI qualche cosa PER MIGLIORARLO.

CAMPIONIAMO IL VALORE DI S E R.

PRIMA di QUESTO CIRCUITO HO ALTRI CIRCUITI CHE PRIMA o POI SI STABILIZZERÀ, Ci VUOLE TEMPO PRIMA CHE UN CIRCUITO COMPLESSO SI STABILIZZI, QUANDO SONO SICURO CHE I VALORI IN INPUT SI STABILIZZANO allora CERCO di CAMPIONARE IL VALORE. QUANDO VADO ad effettuare QUEL CAMPIONAMENTO, IL CIRCUITO SI CHIAMA FLIP FLOP, UN CIRCUITO digitale.

DOBBIAMO CERCARE di NON FAR ARRIVARE 1 e 1 IN INPUT a QUESTO CIRCUITO!