SAPIENZA Università di Roma
Dipartimento di Informatica

Latch e Flip-flop

Prof. Daniele Gorla

Reti sequenziali

SAPIENZA Università di Roma
Dipartimento di Informatica

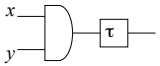
Finora abbiamo solo considerato circuiti *aciclici*.

Questo perché abbiamo implicitamente assunto che le porte siano *ideali*, nel senso che hanno un tempo di attraversamento nullo.

Con questa assunzione non ha senso avere

perché il valore di y dipenderebbe da se stesso (definizione malfondata).

In realtà, le porte hanno un tempo di attraversamento, tipicamente modellato avendo una porta ideale (ad attraversam. nullo) e in serie un ritardo τ:



Questo introduce il fattore **tempo** nei circuiti, che pertanto verranno chiamati *reti sequenziali*.

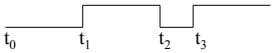
Variabili che variano nel tempo

SAPIENZA Università di Roma
Dipartimento di Informatica

Le variabili booleane non avranno più un semplice assegnamento di 0 o 1, ma tale assegnamento varierà nel tempo

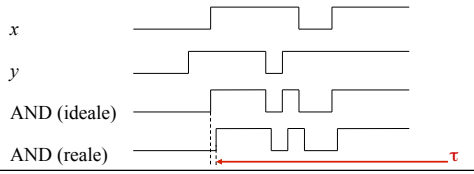
Ciò viene rappresentato tramite un *diagramma temporale*, rappresentazione grafica che consente di mostrare la variazione dei valori di una variabile nel corso del tempo.

Es.: x



Analogamente, si può rappresentare mediante un diagramma temporale l'evoluzione di una rete sequenziale:

Es.: x



Notazione temporale

SAPIENZA Università di Roma
Dipartimento di Informatica

Mentre per i circuiti combinatori non aveva senso scrivere $y = x \cdot y$, ora questo ha senso perché la prima y rappresenta il valore della variabile con un ritardo τ rispetto alla seconda y.

→ stessa linea nel diagramma temporale, campionata a istanti diversi

Per chiarezza scriveremo $Y = x \cdot y$, per evidenziare che la prima occorrenza (Y) rappresenta il valore all'istante $t+\tau$, mentre la seconda occorrenza (y) rappresenta il valore all'istante t.

Le reti sequenziali, pertanto, calcolano funzioni booleane che variano nel tempo in base a:

1. variazioni dei valori di input, e
2. valori di uscita della rete ad istanti precedenti

→ circuiti *con memoria!!*

Circuiti di memoria elementari

Il circuito sequenziale basilare è il Flip Flop, un circuito in grado di memorizzare **un bit** di informazione (ci arriveremo).

Latch SR

Le funzioni essenziali di tale circuito sono 3:

1. immagazzinare uno 1;
2. immagazzinare un 0;
3. mantenere il bit memorizzato.

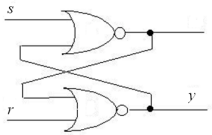
} Codificabili con 2 bit (s e r)

Le tre funzioni sono dette *set*, *reset* e *hold*.

s	r	y	Y
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	-
1	1	1	-

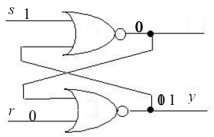
y \ sr	00	01	11	10
0	0	0	-	1
1	1	0	-	1

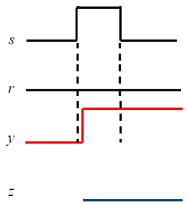
$$Y = \bar{r}(y+s) = r + \overline{(y+s)}$$



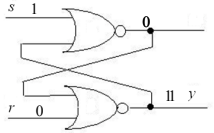
Stato di Set ($s = 1$, $r = 0$)

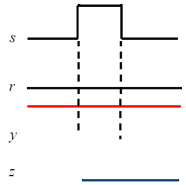
$y = 0$





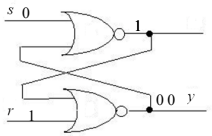
$y = 1$

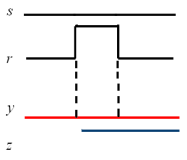




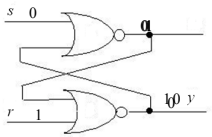
Stato di Reset ($s = 0$, $r = 1$)

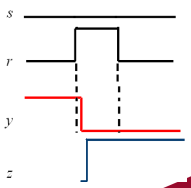
$y = 0$





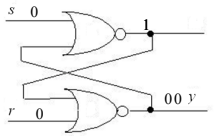
$y = 1$

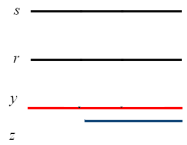




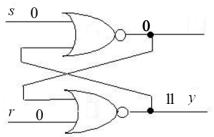
Stato di Hold ($s = 0$, $r = 0$)

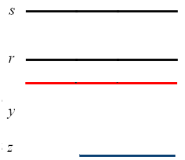
$y = 0$





$y = 1$





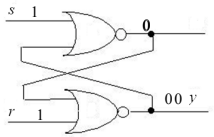
La combinazione s=r=1 "appiattisce" il latch

SAPIENZA Università di Roma Dipartimento di Informatica

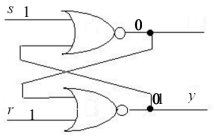
In tutti i casi visti finora, z è il complemento di y (dopo un opportuno τ).

Questa proprietà si perde quando $s = r = 1$; inoltre, in questo caso, sia z che y sono sempre a 0, indipendentemente dal valore iniziale di y.

$y = 0$



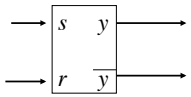
$y = 1$



$s = r = 1$ è un input proibito per il latch!!

Latch SR di porte NOR (riassunto)

SAPIENZA Università di Roma Dipartimento di Informatica



Rappresentazione circuitale

s r	Y
00	y
01	0
10	1
11	-

Descrizione del funzionamento

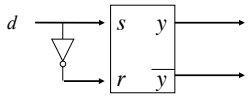
y Y	s r
0 0	0 -
0 1	1 0
1 0	0 1
1 1	- 0

Funzione di eccitazione

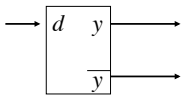
Evitare la configurazione s = r = 1 (latch D)

SAPIENZA Università di Roma Dipartimento di Informatica

Un modo per garantire che la configurazione $s = r = 1$ non si presenti mai è il seguente:



Questo è un nuovo latch, chiamato **latch D** (*delay*), che memorizza l'input della linea d e lo ripropone sull'output y con un ritardo dovuto all'attraversamento delle porte NOT e NOR.



Rappresentazione circuitale

d	Y
0	0
1	1

Descrizione del funzionamento

y Y	d
0 0	0
0 1	1
1 0	0
1 1	1

Funzione di eccitazione

Usare la configurazione s = r = 1

SAPIENZA Università di Roma Dipartimento di Informatica

Un altro modo per garantire che la configurazione $s = r = 1$ non si presenti mai è aggiungere una nuova funzionalità al latch SR:

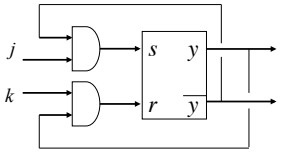
j	k	y	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

come SR

y \ j k	00	01	11	10
0	0	0	1	1
1	1	0	0	1

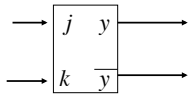
$Y = j\bar{y} + \bar{k}y$

oppure, usando un SR:



Latch JK

Questo è un nuovo latch, chiamato **latch JK**, descritto dalle seguenti informazioni:



Rappresentazione circuitale

j k	*Y*
00	*y*
01	0
10	1
11	$\bar{y}$

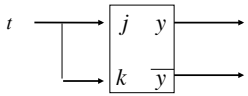
Descrizione del funzionamento

y Y	*j k*
0 0	0 -
0 1	1 -
1 0	- 1
1 1	- 0

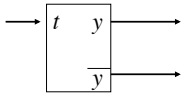
Funzione di eccitazione

Latch T

Cosa succede in un latch JK ponendo $j = k$?



Questo è un nuovo latch, chiamato **latch T** (*toggle*), che o mantiene il valore memorizzato ($t = 0$) o lo complementa ($t = 1$).



Rappresentazione circuitale

t	*Y*
0	*y*
1	$\bar{y}$

Descrizione del funzionamento

y Y	*t*
0 0	0
0 1	1
1 0	1
1 1	0

Funzione di eccitazione

Sistemi sincroni e asincroni

Le reti sequenziali possono operare in modo **sincrono** o **asincrono**.

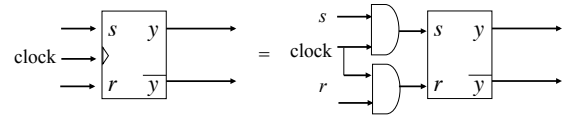
- Nei sistemi asincroni i circuiti logici cambiano ogni volta che uno o più ingressi cambiano;
- Nei sistemi sincroni, il momento in cui gli ingressi sono presi in considerazione è determinato da un segnale di "cadenza", detto *clock* (un treno di impulsi periodico ad onda quadra):

Le reti **asincrone** hanno il vantaggio di reagire istantaneamente alle variazioni degli input, ma questo rende la loro progettazione estremamente complessa e quindi usata solo in casi di assoluta necessità di una risposta immediata.

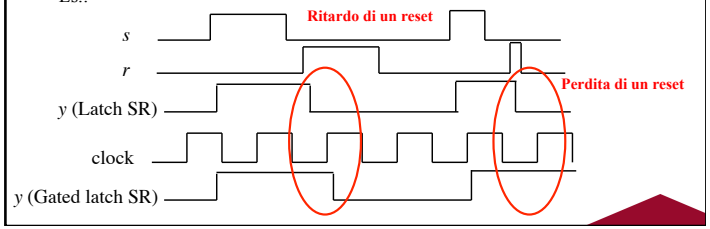
I moderni elaboratori utilizzano praticamente solo reti **sincrone**!

Gated latch

Dato un segnale di clock, possiamo metterlo in AND con tutti gli ingressi del latch; in questo modo gli ingressi verranno considerati solo negli istanti in cui il clock ha valore 1.



Es.:



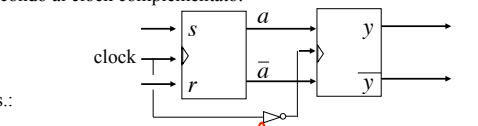
Flip-flop Master-slave

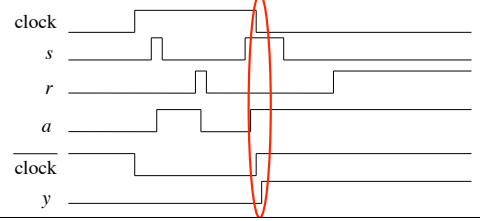
SAPIENZA UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

I gated latch funzionano bene solo se il clock è a 1 per pochissimo tempo.

Soluzione alternativa sono i Flip-flop master-slave (SR, per esempio), ottenuti mettendo in cascata due gated latch, il primo collegato al clock, il secondo al clock complementato:

Es.:





FF sensibile al fronte d'onda discendente del clock