



Algebra booleana e Reti logiche

Reti sequenziali
(prima parte)

Rete sequenziale

- Come abbiamo visto, in una rete combinatoria l'uscita è esclusivamente funzione dell'ingresso.
- In una rete sequenziale, invece, l'uscita è funzione, oltre che dell'ingresso, anche dello *stato* della rete.



La rete ha cioè “memoria”
dello stato interno raggiunto
per effetto degli ingressi
precedentemente applicati

L'esempio dell'ascensore

- *Ingresso*: comando impartito per raggiungere un certo piano
- *Stato corrente*: piano in cui si trova l'ascensore nel momento in cui viene impartito il comando
- *Stato futuro*: piano a cui si porta l'ascensore per effetto del comando impartito
- *Uscita (risposta della macchina)*: la cabina si muove o meno a seconda dell'ingresso, cioè del comando impartito, e dello stato corrente

L'esempio dell'ascensore

- Comportamento di un ascensore in un edificio a 4 piani:

<i>Ingresso</i> <i>Stato corrente</i>	I_0	I_1	I_2	I_3
S_0	S_0 / O_0	S_1 / O_1	S_2 / O_1	S_3 / O_1
S_1	S_0 / O_1	S_1 / O_0	S_2 / O_1	S_3 / O_1
S_2	S_0 / O_1	S_1 / O_1	S_2 / O_0	S_3 / O_1
S_3	S_0 / O_1	S_1 / O_1	S_2 / O_1	S_3 / O_0

Uscita: $O_0 \rightarrow$ “la cabina non si muove”
 $O_1 \rightarrow$ “la cabina si muove”

Modello generale di una rete sequenziale

■ Variabili di

- ✓ *ingresso:* x_1, x_2, \dots, x_n
- ✓ *uscita:* z_1, z_2, \dots, z_m
- ✓ *stato presente:* y_1, y_2, \dots, y_ℓ
- ✓ *stato futuro:* Y_1, Y_2, \dots, Y_ℓ

con x_i, z_i, y_i, Y_i definite su $\{0, 1\}$

Modello generale di una rete sequenziale

- Una n-pla (x_1, x_2, \dots, x_n) costituisce una *configurazione di ingresso* (o più semplicemente un ingresso)
- L'insieme delle $N = 2^n$ configurazioni di ingresso rappresenta l'*alfabeto di ingresso*

$$I = \{I_1, I_2, \dots, I_N\}$$



ogni configurazione I_k
costituisce un *simbolo dell'alfabeto*

Modello generale di una rete sequenziale

- In modo del tutto analogo si definiscono l'*alfabeto di uscita* e l'*alfabeto di stato*:

$$O = \{O_1, O_2, \dots, O_M\} \quad \text{con} \quad M = 2^m$$

$$S = \{S_1, S_2, \dots, S_L\} \quad \text{con} \quad L = 2^\ell$$

dove O_k ed S_k rappresentano rispettivamente una *configurazione di uscita* e una *configurazione di stato* (o più semplicemente un'uscita e uno stato del sistema)

Modello generale di una rete sequenziale

- Si definisce *macchina sequenziale* la quintupla:

$$M = (I, O, S, f, g)$$

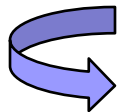
dove I , O e S rappresentano rispettivamente gli alfabeti di ingresso, di uscita e di stato, mentre f e g sono due funzioni così definite:

$$f: S \times I \rightarrow O \quad (\text{funzione di uscita o di trasferimento})$$

$$g: S \times I \rightarrow S \quad (\text{funzione di stato o di transizione})$$

Modello generale di una rete sequenziale

- Poiché l'alfabeto di stato è finito si parla anche di *macchina o automa a stati finiti*.



Due modelli:

- ✓ *automa di Mealy:*

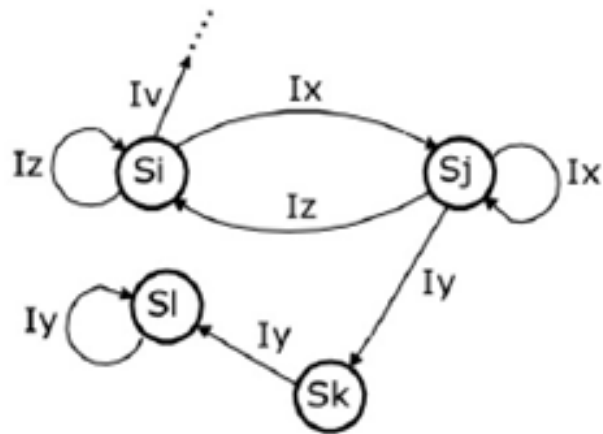
$$f: S \times I \rightarrow O$$

- ✓ *automa di Moore:*

$$f: S \rightarrow O$$

Modello generale di una rete sequenziale

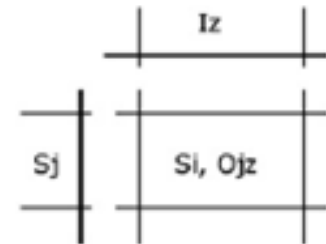
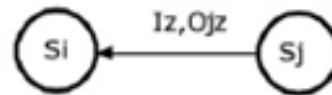
- Le funzioni di stato e di uscita possono essere rappresentate in termini di *diagrammi di stato* e *tabelle di flusso*:



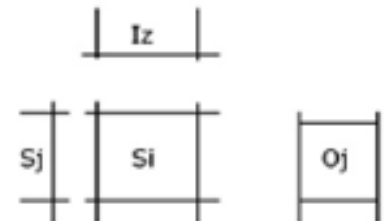
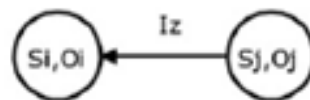
	I1	...	Ix	Iy	Iz	...	IN
S1							
::							
Si			Sj		(Si)		
Sj			(Sj)	Sk	Si		
Sk				Sl			
Sl				(Sl)			
SL							

Modello generale di una rete sequenziale

- Rappresentazione delle uscite con il modello di Mealy:



- Rappresentazione delle uscite con il modello di Moore:

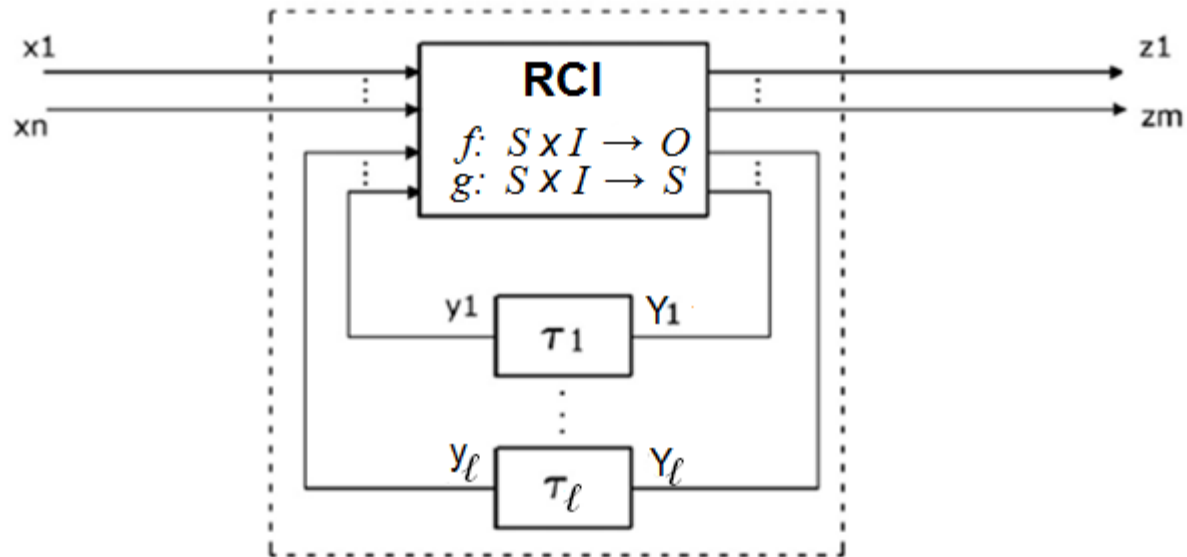


Modello generale di una rete sequenziale

- Una rete logica sequenziale è l'implementazione di una macchina a stati finiti.
 - ➡ Parte combinatoria che realizza le funzioni di uscita e di transizione
 - ➡ Stato futuro riportato in ingresso (y_i diventa uguale a Y_i dopo un intervallo di tempo τ_i)

Modello generale di una rete sequenziale

Rete combinatoria (ideale)
che realizza le funzioni
di uscita e di transizione



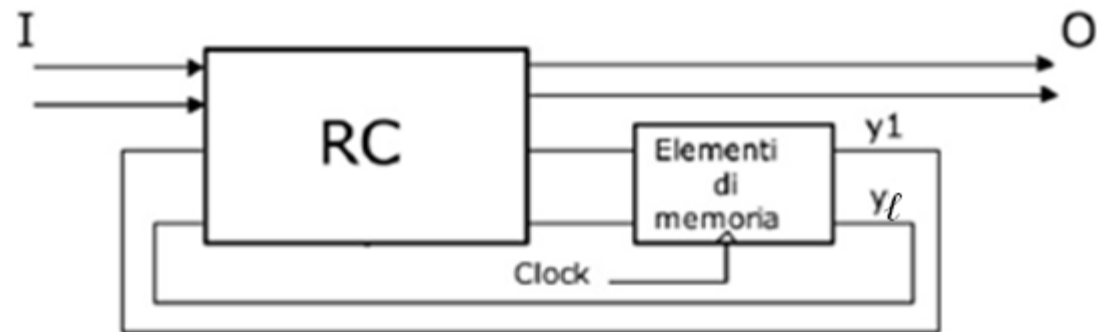
y_i diventa uguale a Y_i
dopo un tempo di ritardo τ_i

Comportamento sincrono e asincrono

- *Reti asincrone*: le variazioni delle configurazioni di ingresso vengono sentite e possono modificare lo stato e le uscite in qualsiasi istante.
 - *Reti sincrone*: le variazioni delle configurazioni di ingresso vengono sentite e possono modificare lo stato e le uscite solo in presenza di un opportuno evento di sincronizzazione.
- ➡ Ci occupiamo delle reti sincrone
(che hanno maggiore rilevanza nei calcolatori)

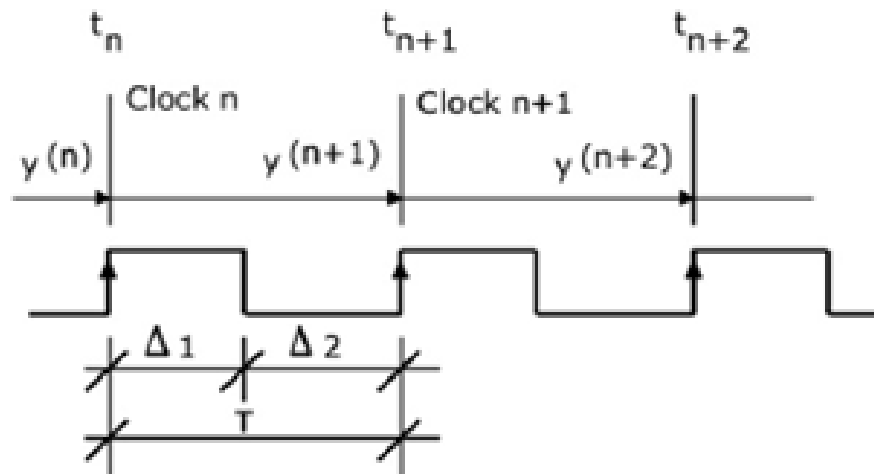
Reti sincrone

- Nelle reti sincrone, i ritardi τ_i sono realizzati da *elementi di memoria* dotati di un ingresso aggiuntivo (*Clock*) che permette di abilitare il cambiamento di stato:



Reti sincrone

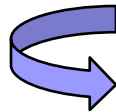
- Clock periodico:



I cambiamenti di stato avvengono
in corrispondenza degli istanti $t_n, t_{n+1}, t_{n+2}, \dots$
di una sequenza temporale discreta

Reti sincrone

- In una rete reale, la parte combinatoria (RC) è costituita da porte con tempi di commutazione non nulli.



La frequenza del clock deve essere scelta in modo da permettere, entro il periodo T , la stabilizzazione di RC e degli stessi elementi di memoria.

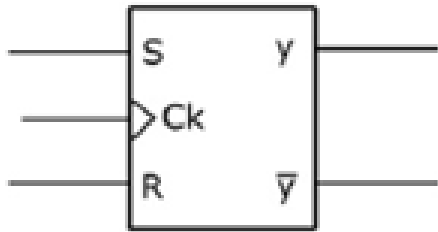
Elementi di memoria

- Circuiti sequenziali molto semplici detti *flip-flop*.

➡ Diversi tipi di flip-flop:

- *flip-flop SR (FF-SR)*
- *flip-flop JK (FF-JK)*
- *delay flip-flop (FF-D)*
- *trigger flip-flop (FF-T)*

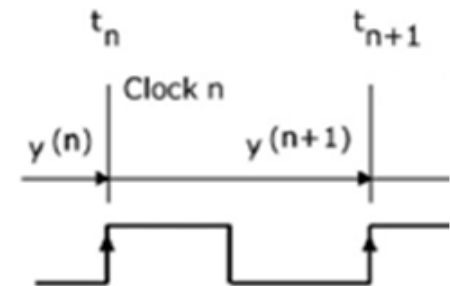
Flip-flop SR



S: *set*, R: *reset*
Ck: *clock*
y: *stato*

Tabella che descrive le variazioni di stato:

$y \backslash SR$	00	01	11	10
0	0	0	-	1
1	1	0	-	1



$$y^{(n+1)} = S + \bar{R}y^{(n)}$$

Flip-flop SR

- L'ingresso $S = 1, R = 1$ non è ammesso.
- L'ingresso $S = 1, R = 0$ porta il FF nello stato 1.
- L'ingresso $S = 0, R = 1$ porta il FF nello stato 0.
- L'ingresso $S = 0, R = 0$ non modifica lo stato (che dipende da quale tra S e R ha assunto per ultimo il valore 1)

Flip-flop SR

- Funzione di transizione:

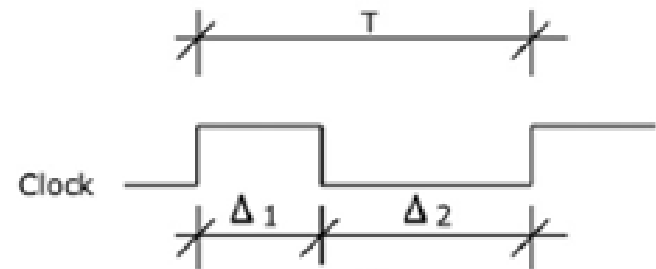
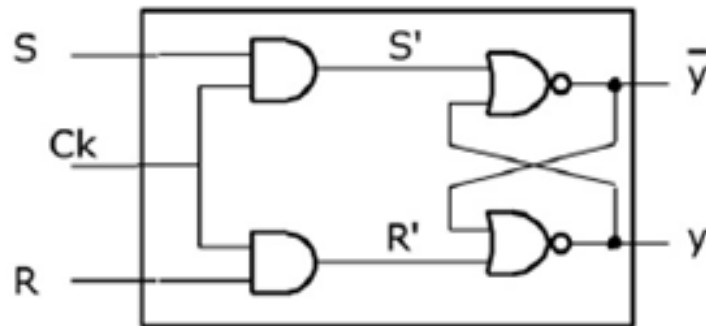
$$y' = S + \bar{R}y \quad \text{ovvero} \quad y^{(n+1)} = S + \bar{R}y^{(n)}$$



Lo stato in cui il flip-flop si troverà all'istante t_{n+1} dipende dallo stato in cui si trovava all'istante t_n e dal relativo ingresso

Flip-flop SR

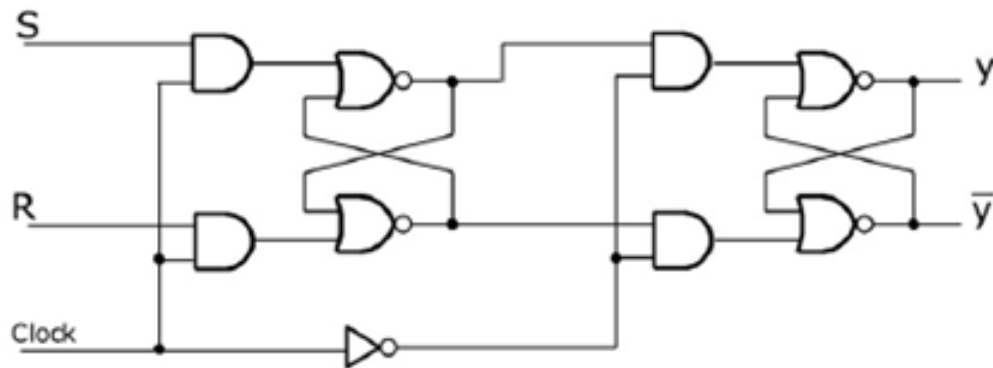
- Schema logico:



Si assume che i segnali S e R non varino durante Δ_1 e che Δ_1 sia sufficiente a far completare l'eventuale transizione di stato della rete (*latch*).

Flip-flop SR

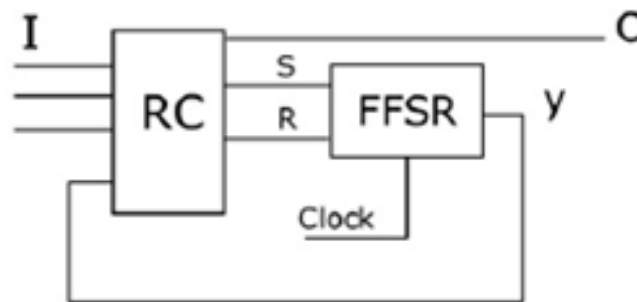
- Configurazione *master-slave*:



Durante Δ_1 ($Ck = 1$) cambia stato solo il latch di sinistra (*master*); durante Δ_2 ($Ck = 0$) il latch di destra (*slave*) si porta nello stato raggiunto dal master (che non può più commutare)

Configurazione master-slave

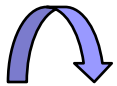
- Con tale configurazione è possibile garantire che, nella rete in cui il FF è inserito, nessun segnale cambi durante Δ_1 :



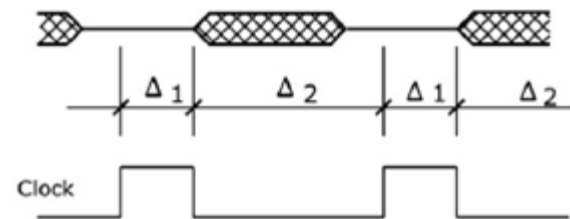
- le variabili di stato y cambiano solo durante Δ_2
- si fa in modo che anche l'ingresso primario I cambi solo durante Δ_2 (*sincronizzazione di I con il clock*)

Configurazione master-slave

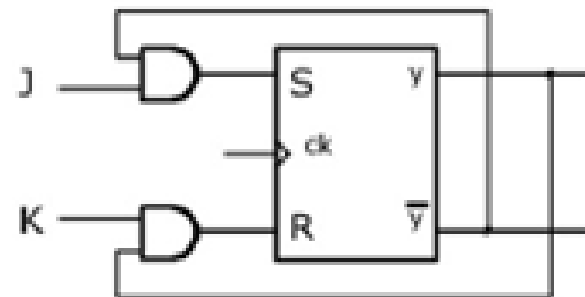
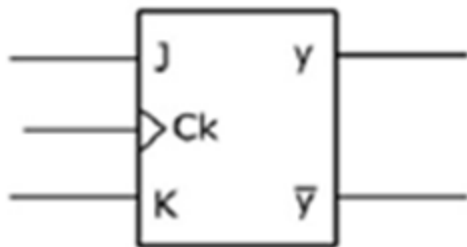
- Vincoli per Δ_1 e Δ_2 :
 - Δ_1 deve essere tale da permettere la commutazione dei master
 - Δ_2 deve essere tale da permettere la stabilizzazione della rete combinatoria



Ad ogni impulso di clock
vengono così campionati
ingressi ben definiti

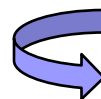


Flip-flop JK



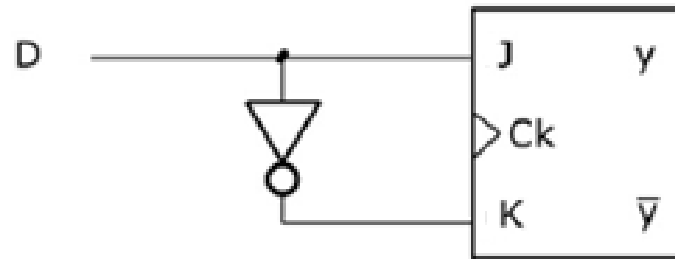
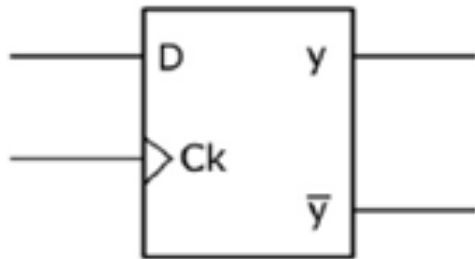
Il comportamento è analogo a quello di FF-SR
tranne che per l'ingresso 11:

$\begin{matrix} JK \\ y \end{matrix}$	00	01	11	10
0	0	0	1	1
1	1	0	0	1

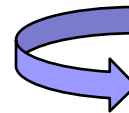


$$y' = \bar{y}J + y\bar{K}$$

Flip-flop D

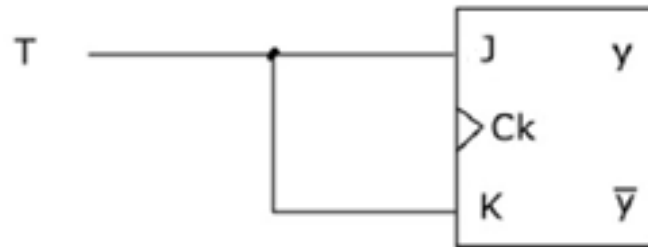
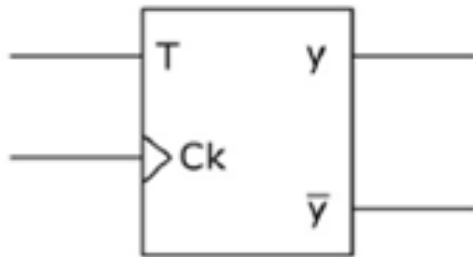


Realizza un blocco di ritardo
pari al periodo di clock T

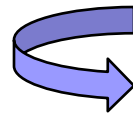


$$y' = D$$

Flip-flop T



Cambia stato se $T = 1$,
rimane nello stesso stato se $T = 0$.



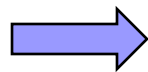
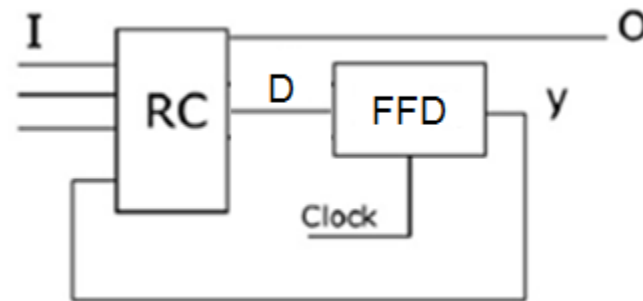
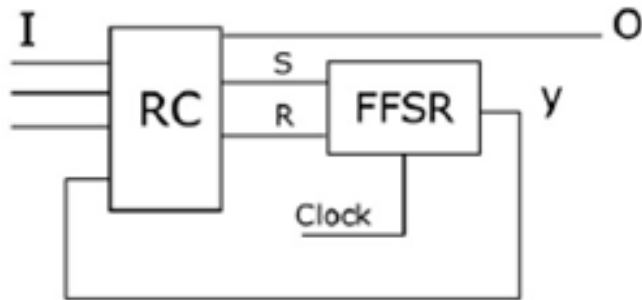
$$y' = \bar{T}y + T\bar{y} = T \oplus y$$

Flop-flop: osservazioni

- Gran parte dei flip-flop sono oggi realizzati in modo da commutare sui fronti di salita o di discesa del clock (*edge triggered flip-flop*).
- Inoltre, i flip-flop in commercio, oltre agli ingressi sincroni, spesso presentano anche ingressi asincroni (comunemente detti *Preset* e *Clear*), che possono essere usati per comandare la commutazione in maniera indipendente dal clock.

Funzioni di eccitazione dei flip-flop

- Come si determinano gli ingressi da applicare ai flip-flop per ottenere il funzionamento voluto della rete sequenziale in cui essi sono inseriti?



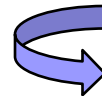
Occorre tener conto della modalità di funzionamento dei flip-flop impiegati.

Funzioni di eccitazione dei flip-flop

- A titolo d'esempio, supponiamo che la funzione di transizione della rete sequenziale sia la seguente:

x1 x2		00	01	11	10
y	0	1	0	1	1
	1	1	1	0	0

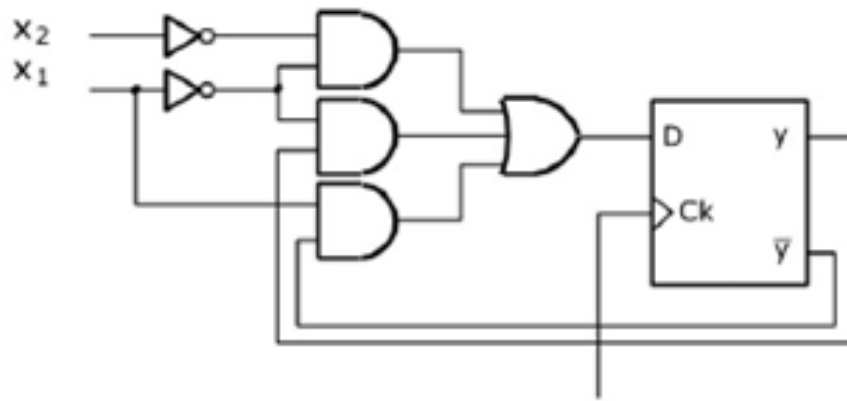
y'



$$y' = \bar{x}_1 \bar{x}_2 + \bar{x}_1 y + x_1 \bar{y}$$

Funzioni di eccitazione dei flip-flop

- Nel caso di FF-D, l'ingresso D è la realizzazione della funzione y' :



$$y' = \bar{x}_1 \bar{x}_2 + \bar{x}_1 y + x_1 \bar{y}$$

$$D = \bar{x}_1 \bar{x}_2 + \bar{x}_1 y + x_1 \bar{y}$$

Funzioni di eccitazione dei flip-flop

- Nel caso di FF-SR e FF-JK, occorre dare una coppia di ingressi tale da realizzare la transizione di stato prevista sulla mappa di y' :

		x1 x2				
			00	01	11	10
y	0		1	0	1	1
	1		1	1	0	0
			y'			

$y \rightarrow y'$	S	R
$0 \rightarrow 0$	0	-
$0 \rightarrow 1$	1	0
$1 \rightarrow 0$	0	1
$1 \rightarrow 1$	-	0

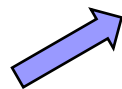
$y \rightarrow y'$	J	K
$0 \rightarrow 0$	0	-
$0 \rightarrow 1$	1	-
$1 \rightarrow 0$	-	1
$1 \rightarrow 1$	-	0

Funzioni di eccitazione dei flip-flop

- Dalla mappa di y' si passa alle mappe di S e R in caso di FF-SR:

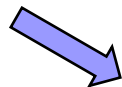
x1 x2		00	01	11	10
y	0	1	0	1	1
	1	1	1	0	0

y'



x1 x2		00	01	11	10
y	0	1	0	1	1
	1	—	—	0	0

$$S = \bar{x}_1 \bar{x}_2 + x_1 \bar{y}$$



x1 x2		00	01	11	10
y	0	0	—	0	0
	1	0	0	1	1

$$R = x_1 y$$

Funzioni di eccitazione dei flip-flop

- Idem per FF-JK:

x1 x2	00	01	11	10
y				
0	1	0	1	1
1	1	1	0	0

y'

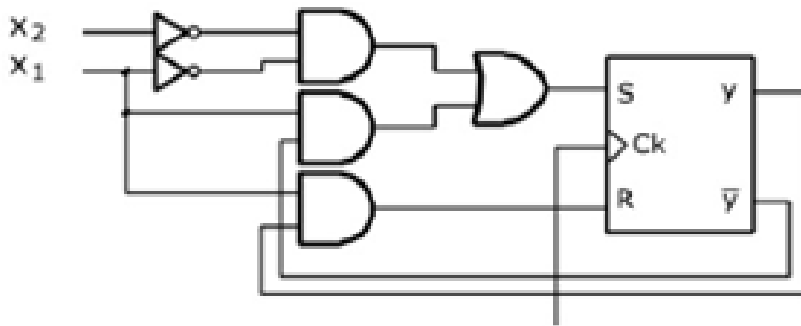
x1 x2	00	01	11	10
y				
0	1	0	1	1
1	—	—	—	—

$$J = x_1 + \bar{x}_2$$

x1 x2	00	01	11	10
y				
0	—	—	—	—
1	0	0	1	1

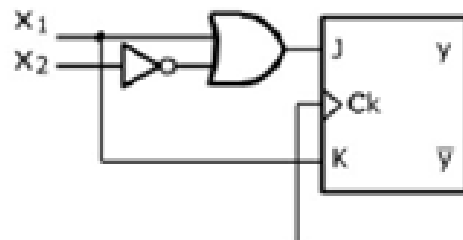
$$K = x_1$$

Funzioni di eccitazione dei flip-flop



$$S = \bar{x}_1 \bar{x}_2 + x_1 \bar{y}$$

$$R = x_1 y$$



$$J = x_1 + \bar{x}_2$$

$$K = x_1$$

(Figure parzialmente tratte da G. Bucci. *Calcolatori elettronici. Architettura e organizzazione*.
Copyright © 2009 - The McGraw-Hill Companies)