

Architetture dei Calcolatori

Reti Sequenziali

Prof. Francesco Lo Presti

Rete Sequenziale

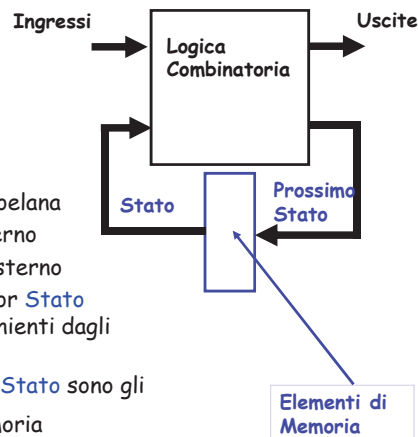
- Una rete sequenziale è caratterizzata da :

- Elementi di Memoria:

- ✓ Latch o Flip-Flop

- Una Rete Combinatoria:

- ✓ Implementa una funzione booleana
- ✓ **Ingressi** sono segnali dall'esterno
- ✓ **Uscite** sono segnali verso l'esterno
- ✓ Ci sono altri ingressi, **Stato** or **Stato corrente**, sono segnali provenienti dagli elementi di memoria
- ✓ Le restanti uscite, **Prossimo Stato** sono gli ingressi agli elementi di memoria



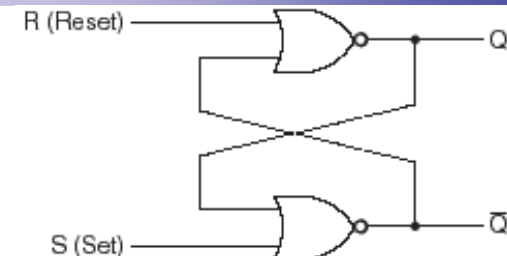
2

Circuiti Sequenziali

- Si introduce il concetto di **stato**
- I valori in uscita influenzano gli ingressi (**feedback**)
- **Asincroni**: il cambio di stato di un elemento influenza immediatamente gli ingressi del circuito
- **Sincroni**: il cambio di stato di uno o più elementi influenza gli ingressi del circuito solo in istanti predeterminati, uguali per tutti

1

Latch SR



- Gli ingressi dipendono dalle uscite!!
- È un elemento di memoria
 - Memorizza 1 bit
- Q è il bit memorizzato nel latch SR
- Ha due stati:
 - set: $Q=1$ ($\bar{Q}=0$)
 - reset: $Q=0$ ($\bar{Q}=1$)
- Gli ingressi S e R possono essere utilizzati per cambiare lo stato del latch

3

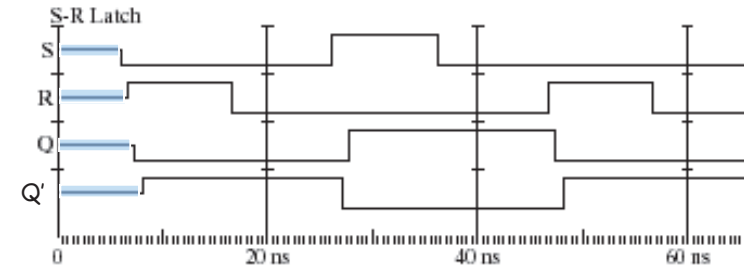
Latch SR

- Prossimo Stato in funzione degli ingressi e dello stato corrente

S	R	Q(k)	Q(k+1)	$\bar{Q}(k+1)$	
0	0	0	0	1	reset (stabile)
0	0	1	1	0	set (stabile)
0	1	0	0	1	reset (stabile)
0	1	1	0	1	reset (transiente)
1	0	0	1	0	Set (transiente)
1	0	1	1	0	Set (stabile)
1	1	0	0	0	!!
1	1	1	0	0	!!

4

Comportamento di un Latch SR nel tempo



5

Evitare S=R=1...

- Quando entrambi gli ingressi commutano da 1 a 0 avviene una **race condition**
- Entambe le uscite vengono portate da 0 a 1
- A causa delle differenze anche minime tra due NOR gates i tempi di commutazione possono differire tra loro → uno delle due uscite commuta prima dell'altra
 - La commutazione della prima porta impedisce la commutazione dell'altra
- **Conclusione:** il valore di uscita non è predicibile

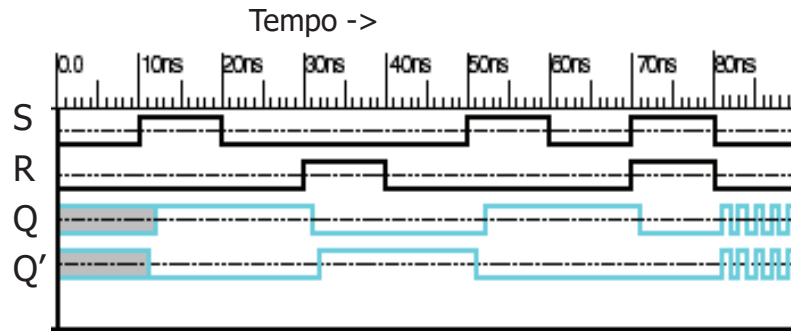
6

Evitare S=R=1...

- Quando entrambi gli ingressi commutano da 1 a 0 avviene una **race condition**
- Entambe le uscite vengono portate da 0 a 1
- Entrambe le porte NOR commutano nello stesso istante
 - Questo comporta la commutazione delle uscite da 0 a 1
 - ...che a sua volta comporta la commutazione delle uscite da 1 a 0
 - ...che a loro volta faranno commutare nuovamente le uscite da 0 a 1
 - ...etc. etc. etc.
- **Conclusione:** le uscite oscilleranno tra 0 e 1!

7

Esempio



8

Tabella di Transizione di un latch SR

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	---

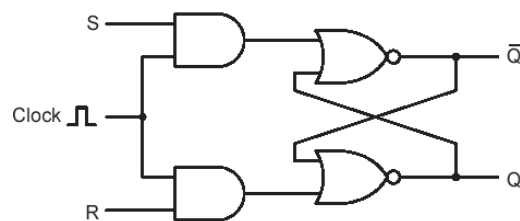
9

Aggiungiamo un segnale di clock al latch SR

- Un segnale di clock è usato per assicurarsi che il latch possa commutare solo in determinati momenti

Segnale di clock

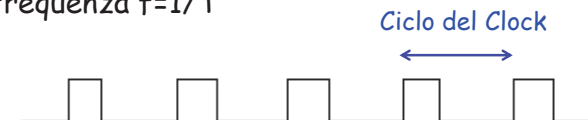
- Il latch reagisce ai valori di S e R solo quando il clock è pari a 1



10

Il ruolo del segnale di clock

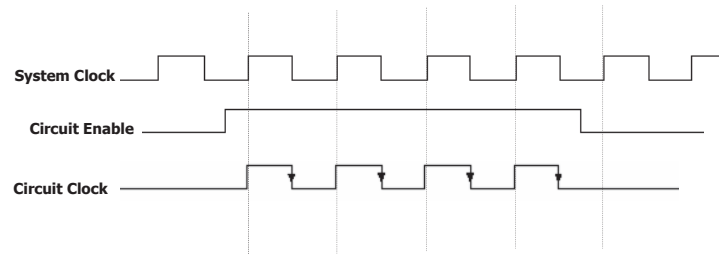
- Il segnale di clock assicura che le modifiche agli ingressi si propagino alle uscite solo quando richiesto
- Clock di Sistema: segnare periodico di periodo T e frequenza $f=1/T$



11

Clock di un circuito a partire dal clock di sistema

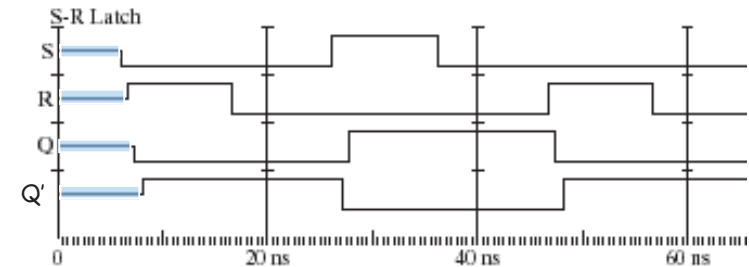
- Il clock di Sistema commuta ogni ciclo: come si può controllare un particolare circuito solo quando serve?



12

Un problema sottile...

- Con il clock=1, le uscite del latch continuano a cambiare ogni qual volta cambiano gli ingressi



13

...di conseguenza...

- Nel caso di commutazione da $(S=1, R=0)$ a $(S=0, R=1)$ o da $(S=0, R=1)$ a $(S=1, R=0)$ le uscite del latch SR per un piccolo intervallo tempo potrebbero trovarsi nello stato dove entrambe le uscite sono 0

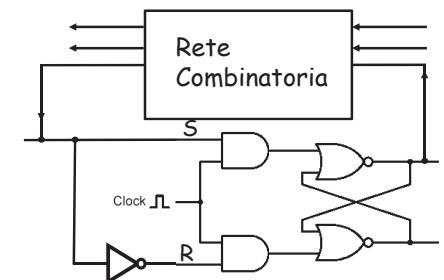
	Stato iniziale	Cambio ingresso	Stato Transiente	Stato Stabile
S	1	0	0	0
R	0	1	1	1
Q	1	1	0	0
$\bar{Q}$	0	0	0	1

- Se Q e $\bar{Q}$ sono ingressi ad un circuito, quest'ultimo riceve ingressi con valori errati (per un certo intervallo di tempo) causando uscite con valori errati (sempre per un certo intervallo di tempo)

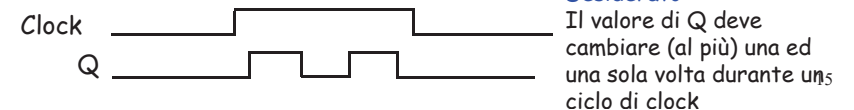
14

...una conseguenza...

- Consideriamo il sequence circuito



- Quando $C=1$, il valore di Q può continuare ad cambiare
 ◦ Quante volte?

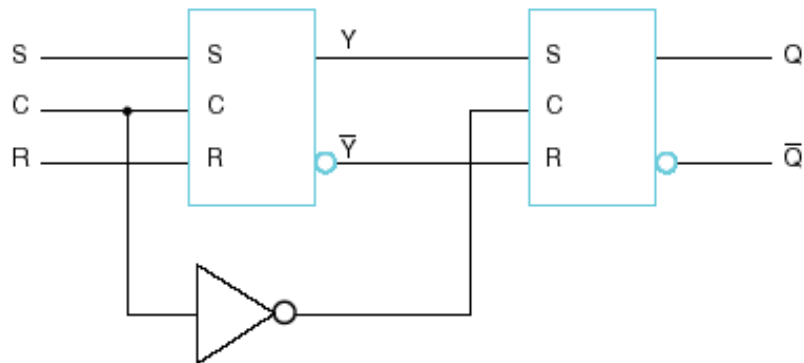


Comportamento Desiderato:
 Il valore di Q deve cambiare (al più) una ed una sola volta durante un ciclo di clock

Soluzione: Circuito Flip-Flop master slave

□ Due latch SR (master e slave)

- Il master (collegato agli ingressi del circuito) può cambiare il suo stato solo quando il suo stato quando il Clock=1
- Lo slave (collegato alle uscite) cambia il suo stato quando il clock=0



16

Soluzione: Circuito Flip-Flop master slave

□ Due latch SR (master e slave)

- Il master (collegato agli ingressi del circuito) può cambiare il suo stato solo quando il suo stato quando il Clock=1
- Lo slave (collegato alle uscite) cambia il suo stato quando il clock=0

□ Pertanto:

- Lo slave legge in ingresso l'output del master solo dopo che questo si è stabilizzato (clock=0)
- Un eventuale circuito a valle riceve in ingresso i valori in uscita dello slave quando le uscite saranno stabilizzate (clock=1)

- Questo permette di controllare quando l'uscita di un circuito a monte impatta gli ingressi del circuito a valle

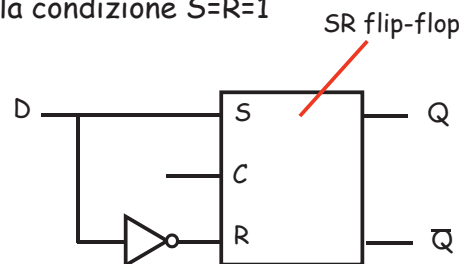
17

Flip Flop D (Delay)

□ Un unico ingresso D

- $R = \bar{S}$

□ Evita la condizione $S=R=1$



18

Flip Flop D (Delay)

□ Il flip-flop D rappresenta una cella di memoria

- Memorizza (e rende disponibile in uscita il valore presentato in ingresso)

Simbolo

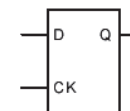


Tabella di Verità

D	Q_{n+1}
0	0
1	1

19

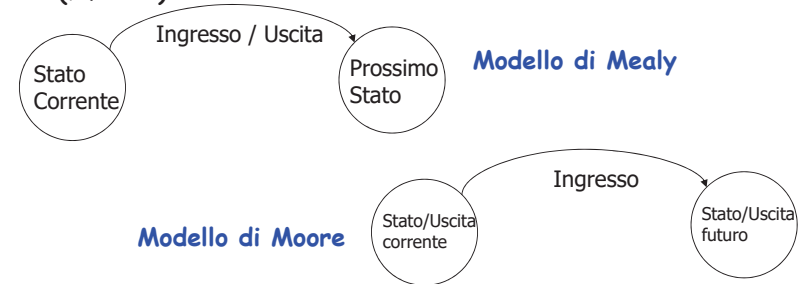
Automa a Stati Finiti (ASF) Finite State Machine (FSM)

- Modello Formale per descrivere un circuito sequenziale
- Descritto tramite una tabella di **transizioni** tra **stati** in funzione degli **ingressi**
- e una tabella di **uscite** in funzione degli **ingressi** e/o **stato**
- Due Modelli formali:
 - **Modello di Moore**: uscite sono funzione solo dello stato
 - **Modello di Mealy**: uscite sono funzione dello stato e degli ingressi

20

Rappresentazione Grafica

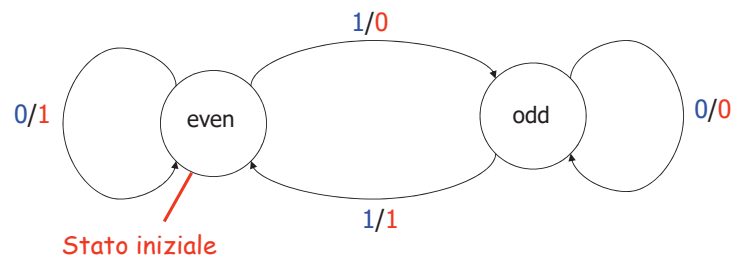
- Stati sono rappresentate da nodi, transizioni da archi orientate tra stati, ingressi da etichette degli archi, le uscite da etichette degli archi (Mealy) o stati (Moore)



21

Esempio

- ASF con ingresso binario **x**, e uscita binaria **z**. L'uscita è $z=1$ se e solo se il numero di 1 nella sequenza in ingresso è pari, 0 altrimenti



22

Descrizione tabellare dell'ASF

- Prossimo stato in funzione dello stato corrente e dell'ingresso

Stato Corrente	Ingresso	Prossimo Stato
even	0	even
even	1	odd
odd	0	odd
odd	1	even

- Uscita in funzione dello stato corrente e dell'ingresso

Stato Corrente	Ingresso	Prossimo Stato
even	0	1
even	1	0
odd	0	0
odd	1	1

23

Processo di Astrazione

- ASF **descrive** una rete sequenziale (RS)
- RS **implementa** un Automa a Stati Finiti
- L'**analisi** di una RS permette di risalire al corrispondente ASF
- La RS che implementa un dato ASF si ottiene tramite il processo di **sintesi**
- Simile a quanto fatto con funzioni booleane e circuiti combinatori
 - Una Funzione Booleana **descrive** una rete combinatoria
 - Una Rete Combinatoria **implementa** una Funzione Booleana

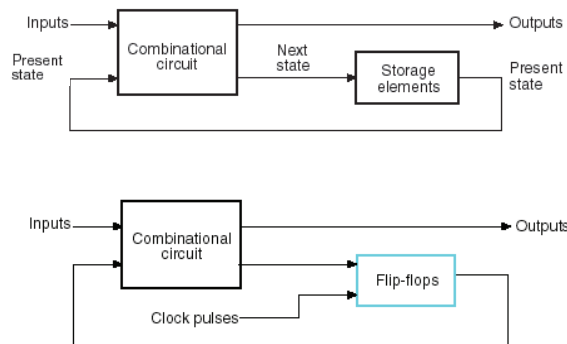
24

Sintesi di una Rete Sequenziale

- Identificare ingressi, uscite e lo stato
 - Quanti Flip Flop sono necessari?
 - Per un ASF con n stati sono necessari k FFs con $2^k \geq n$
 - Etichetta ogni stato con una stringa di k bit
- Identificare le tabella di verità per le uscite in funzione dello stato e degli ingressi e sintetizzare la relativa rete combinatoria
- Identificare le tabella di verità corrispondente alla tabella di transizione (prossimo stato in funzione dello stato e degli ingressi) e sintetizzare la relativa rete combinatoria
- Determinare la tipologia di Flip Flop per memorizzare lo stato
 - Flip Flop tipo D sono la scelta più semplice

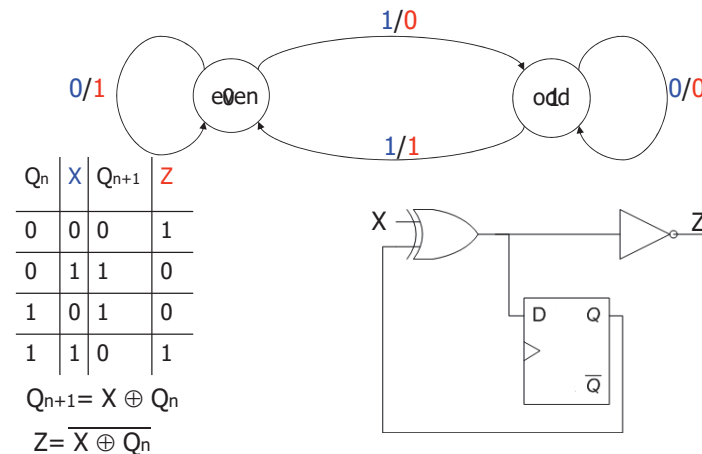
25

Schema di una Rete Sequenziale



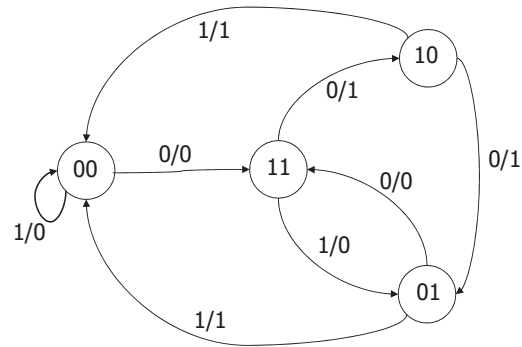
26

Esempio



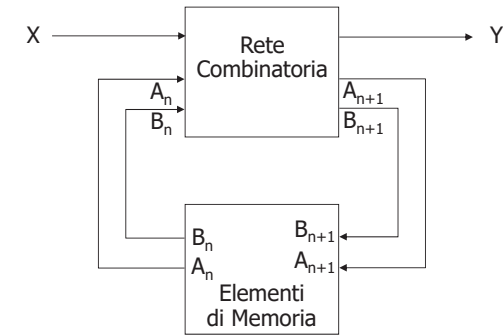
27

Esempio 1



28

Esempio 1: Schema Rete



29

Example 1: Tabelle di Transizione

A_n	B_n	X	Y	A_{n+1}	B_{n+1}
0	0	0	0	1	1
0	0	1	0	0	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	0	0
1	1	0	1	1	0
1	1	1	0	0	1

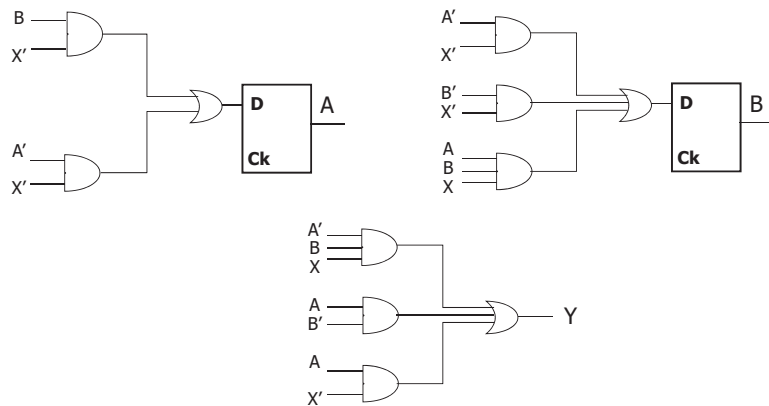
30

Esempio 1: Minimizzazione

Variabili di Stato				Uscita			
A_{n+1}		X		B_{n+1}		Y	
AB	11	0	1	AB	11	0	1
	01	1	0		01	0	1
	00	1	0		11	1	0
	10	0	0		10	1	1

31

Esempio 1: circuito logico



32

Esempio 2

□ Rete sequenziale con due ingressi

□ L'uscita è:

- 10 quando il numero di ingressi consecutivi 00 è dispari
- 01 quando il numero di ingressi consecutivi 11 è dispari
- 11 quando il numero di ingressi consecutivi 00 o 11 è pari
- 00 altrimenti

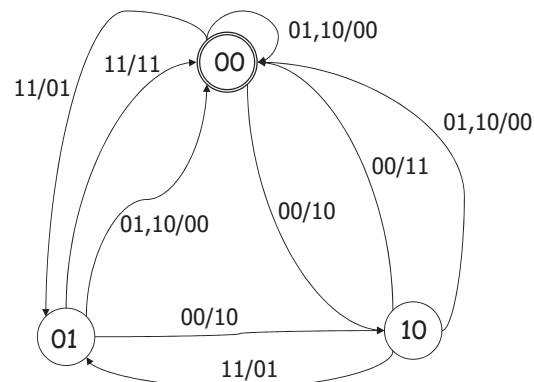
□ Esempio:

INPUT	01	01	00	00	00	11	11	10	11	11	11	11
OUTPUT	00	00	10	11	10	01	11	00	01	11	01	11

33

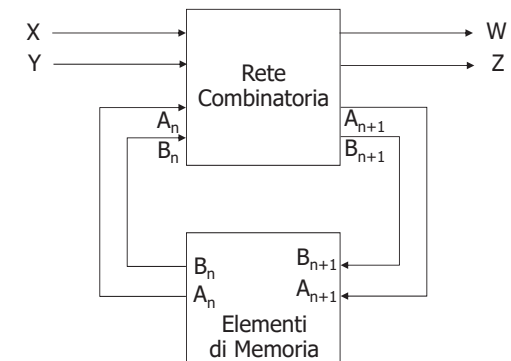
Esempio 2: Automa a Stati Finiti

□ Stato iniziale (doppio cerchio)



34

Esempio 2: Schema Rete Sequenziale



35

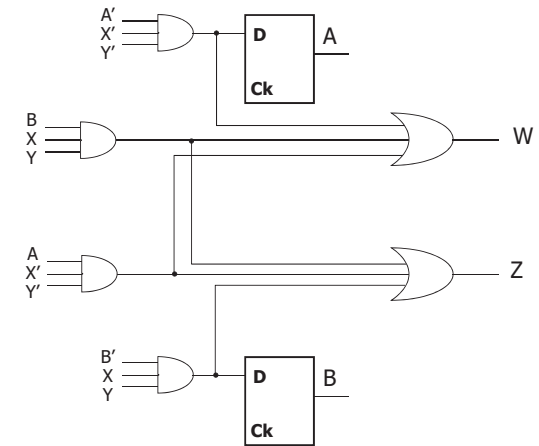
Esempio 2: tabella di transizione

A_n	B_n	X	Y	W	Z	A_{n+1}	B_{n+1}
0	0	0	0	1	0	1	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	1	0	1
0	1	0	0	1	0	1	0
0	1	0	1	0	0	0	0
0	1	1	0	0	0	0	0
0	1	1	1	1	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0
1	0	1	1	0	1	0	1
1	1	0	0	--	--	--	--
1	1	0	1	--	--	--	--
1	1	1	0	--	--	--	--
1	1	1	1	--	--	--	--

Non specificati in
quanto lo stato 11
non è presente

36

Esempio 2: circuito logico



37