

Livello dei componenti fisici: Circuiteria elettronica

Eugenio G. Omodeo



UNIVERSITÀ
DEGLI STUDI DI TRIESTE

Dip. Matematica e Geoscienze — DMI



UNIVERSITÀ
DEGLI STUDI DI TRIESTE

Trieste, 06/12/2022

- Le componenti a **2 stati** bastano per la rappresentazione digitalizzata dell'informazione:



- Le componenti a **2 stati** bastano per la rappresentazione digitalizzata dell'informazione:
- a cominciare dai numeri, in un sistema digitale di elaborazione tutto è rappresentato tramite **bit** (binary digit: **0**, **1**).



- Le componenti a **2 stati** bastano per la rappresentazione digitalizzata dell'informazione:
- a cominciare dai numeri, in un sistema digitale di elaborazione tutto è rappresentato tramite **bit** (binary digit: **0**, **1**).
- Gli esiti di confronti a **due vie** (successo / fallimento) possono guidare l'ordine di esecuzione delle istruzioni di un programma.





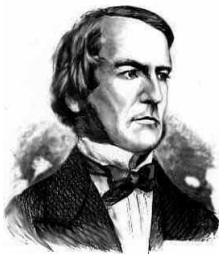
WIKIPEDIA
L'enciclopedia libera

[Pagina principale](#)
[Ultime modifiche](#)
[Una voce a caso](#)
[Vetrina](#)
[Aiuto](#)

Porta logica

Da Wikipedia, l'enciclopedia libera.

In **elettronica digitale** e **informatica**, una **porta logica** è un **circuito digitale** in grado di **implementare** (cioè di realizzare, simulandone la "logica matematica" mediante opportuni controlli su segnali elettrici) una particolare operazione logica di una o più **variabili booleane**.



(George Boole, 1815 – 1864)



UNIVERSITÀ
DEGLI STUDI DI TRIESTE

OPERAZIONI DELLA LOGICA PROPOSIZIONALE

'	"	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
f	&	>	'	<	"	+	∨	↓	↔	¬"	←	¬'	→		∨		
f	f	f	f	f	f	f	f	f	f	v	v	v	v	v	v	v	v
f	v	f	f	f	f	v	v	v	v	f	f	f	f	v	v	v	v
v	f	f	f	v	v	f	f	v	v	f	f	v	v	f	f	v	v
v	v	f	v	f	v	f	v	f	v	f	v	f	v	f	v	f	v

Intuitivamente parlando, **f** e **v** stanno per falso e vero.



OPERAZIONI DELLA LOGICA PROPOSIZIONALE

'	"	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
'	"	f	&	>	'	<	"	+	∨	↓	↔	¬"	←	¬'	→		∇
f	f	f	f	f	f	f	f	f	f	∇	∇	∇	∇	∇	∇	∇	∇
f	∇	f	f	f	f	∇	∇	∇	∇	f	f	f	f	∇	∇	∇	∇
∇	f	f	f	∇	∇	f	f	∇	∇	f	f	∇	∇	f	f	∇	∇
∇	∇	f	∇	f	∇	f	∇	f	∇	f	∇	f	∇	f	∇	f	∇

zOOMando entro
questa tavola, tro-
viamo che:

¬		
0		1
1		0

OPERAZIONI DELLA LOGICA PROPOSIZIONALE

'	"	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
f	&	>	'	<	"	+	∨	↓	↔	¬"	←	¬'	→		∇		
f	f	f	f	f	f	f	f	f	f	∇	∇	∇	∇	∇	∇	∇	∇
f	∇	f	f	f	f	∇	∇	∇	∇	f	f	f	f	∇	∇	∇	∇
∇	f	f	f	∇	∇	f	f	∇	∇	f	f	∇	∇	f	f	∇	∇
∇	∇	f	∇	f	∇	f	∇	f	∇	f	∇	f	∇	f	∇	f	∇

zOOMando entro questa tavola, troviamo che:

¬		&	
0	1	0	0
1	0	0	0
		1	1

OPERAZIONI DELLA LOGICA PROPOSIZIONALE

'	"	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
'	"	f	&	>	'	<	"	+	∨	↓	↔	¬"	←	¬'	→		∇
f	f	f	f	f	f	f	f	f	f	∇	∇	∇	∇	∇	∇	∇	∇
f	∇	f	f	f	f	∇	∇	∇	∇	f	f	f	f	∇	∇	∇	∇
∇	f	f	f	∇	∇	f	f	∇	∇	f	f	∇	∇	f	f	∇	∇
∇	∇	f	∇	f	∇	f	∇	f	∇	f	∇	f	∇	f	∇	f	∇

zOOMando entro questa tavola, troviamo che:

¬		&			∨	
0	1	0	0	0	0	0
1	0	0	1	0	0	1
1	1	1	0	0	1	0
		1	1	1	1	1



OPERAZIONI DELLA LOGICA PROPOSIZIONALE

'	"	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
f	&	>	'	<	"	+	∨	↓	↔	¬"	←	¬'	→		∨		
f	f	f	f	f	f	f	f	f	f	v	v	v	v	v	v	v	v
f	v	f	f	f	f	v	v	v	v	f	f	f	f	v	v	v	v
v	f	f	f	v	v	f	f	v	v	f	f	v	v	f	f	v	v
v	v	f	v	f	v	f	v	f	v	f	v	f	v	f	v	f	v

zOOMando entro questa tavola, troviamo che:



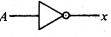
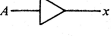
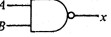
¬	
0	1
1	0

&		
0	0	0
0	1	0
1	0	0
1	1	1




∨		
0	0	0
0	1	1
1	0	1
1	1	1

→		
0	0	1
0	1	1
1	0	0
1	1	1



Name	Graphic Symbol	Algebraic Function	Truth Table														
AND	 $x = A \cdot B$ or $x = AB$	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>x</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	x	0	0	0	0	1	0	1	0	0	1	1	1
A	B	x															
0	0	0															
0	1	0															
1	0	0															
1	1	1															
OR	 $x = A + B$	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>x</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	x	0	0	0	0	1	1	1	0	1	1	1	1
A	B	x															
0	0	0															
0	1	1															
1	0	1															
1	1	1															
inverter	 $x = A'$	<table border="1"> <thead> <tr> <th>A</th> <th>x</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	x	0	1	1	0									
A	x																
0	1																
1	0																
buffer	 $x = A$	<table border="1"> <thead> <tr> <th>A</th> <th>x</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	x	0	0	1	1									
A	x																
0	0																
1	1																
NAND	 $x = (AB)'$	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>x</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	x	0	0	1	0	1	1	1	0	1	1	1	0
A	B	x															
0	0	1															
0	1	1															
1	0	1															
1	1	0															



<i>Name</i>	<i>Graphic Symbol</i>	<i>Algebraic Function</i>	<i>Truth Table</i>															
NOR		$x = (A + B)'$	<table border="1"> <thead> <tr> <th><i>A</i></th> <th><i>B</i></th> <th><i>x</i></th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	<i>A</i>	<i>B</i>	<i>x</i>	0	0	1	0	1	0	1	0	0	1	1	0
<i>A</i>	<i>B</i>	<i>x</i>																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
exclusive-OR (XOR)		$x = A \oplus B$ or $x = A'B + AB'$	<table border="1"> <thead> <tr> <th><i>A</i></th> <th><i>B</i></th> <th><i>x</i></th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	<i>A</i>	<i>B</i>	<i>x</i>	0	0	0	0	1	1	1	0	1	1	1	0
<i>A</i>	<i>B</i>	<i>x</i>																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
exclusive-NOR or equivalence		$x = A \odot B$ or $x = A'B' + AB$	<table border="1"> <thead> <tr> <th><i>A</i></th> <th><i>B</i></th> <th><i>x</i></th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	<i>A</i>	<i>B</i>	<i>x</i>	0	0	1	0	1	0	1	0	0	1	1	1
<i>A</i>	<i>B</i>	<i>x</i>																
0	0	1																
0	1	0																
1	0	0																
1	1	1																



Teorema

Qualsiasi tabella di verità ad n operandi (ve ne sono, in tutto, 2^{2^n}) può venir espressa combinando opportunamente n variabili a valori v / f tramite le operazioni logiche.



A titolo di esempio, esprimiamo in **forma normale congiuntiva** una celebre funzione booleana, la **Implicazione 'a tre vie'**

seAlloraAltrim			
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Prendendo le righe a risultato 0, leggiamo direttamente:

$$(A \vee B \vee C) \& (A \vee \neg B \vee C) \& (\neg A \vee B \vee C) \& (\neg A \vee B \vee \neg C)$$

Fra le altre, ecco una codifica ben piú sbrigativa:

$$(\neg A \vee B) \& (A \vee C)$$

Piú facile da seguire, la costruzione di una SOP

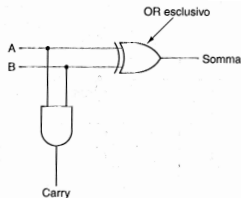
(= *Sum Of Products*)



Disegno, minimizzazione,
generazione automatica di
circuiti combinatori

A	B	Somma	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

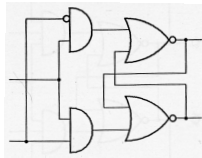
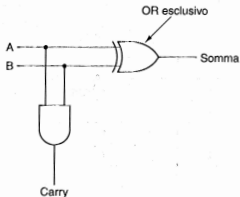
a)



Disegno, minimizzazione,
generazione automatica di
circuiti combinatori¹

A	B	Somma	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

a)

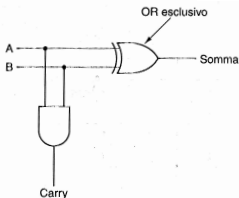


¹Per i circuiti sequenziali, serve di più...

Disegno, minimizzazione,
generazione automatica di
circuiti combinatori

A	B	Somma	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

a)



The Nobel Prize in Physics 1956
William B. Shockley, John Bardeen, Walter H. Brattain

Share this:    

The Nobel Prize in Physics 1956



William Bradford
Shockley
Prize share: 1/3



John Bardeen
Prize share: 1/3



Walter Houser
Brattain
Prize share: 1/3

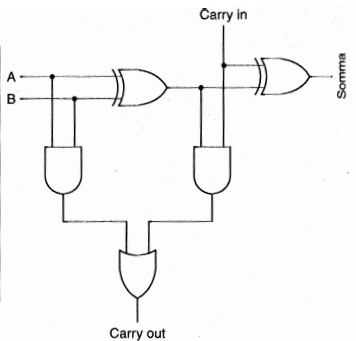
The Nobel Prize in Physics 1956 was awarded jointly to William Bradford Shockley, John Bardeen and Walter Houser Brattain "for their researches on semiconductors and their discovery of the transistor effect".

Photos: Copyright © The Nobel Foundation



Esempio (addizionatore di bit con riporto)

A	B	Carry in	Somma	Carry out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Esempio (addizionale di bit con riporto)

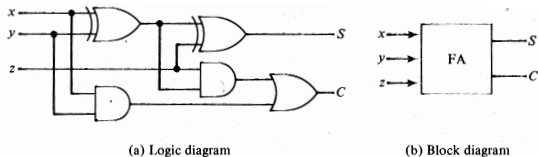


Fig. 1-19 Full-adder circuit.

(È solo disegnato diversamente!)



Esempio (addizionale di 4 bit con riporto)

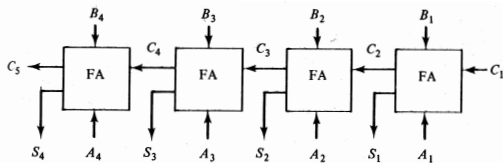


Fig. 4-8 Four-bit full-adders.



Flip-flop D (data) [\[modifica | modifica wikitesto \]](#)

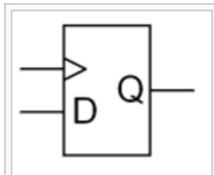
Ha un ingresso per il dato, un ingresso di sincronizzazione (**clock**) e un'uscita. In corrispondenza del comando di **clock**, trasferisce l'ingresso in uscita e ve lo mantiene fin quando non cambia il suddetto ingresso.


Equazione caratteristica: $Q^+ = D$

Applicazioni: per le sue caratteristiche è il componente base delle memorie (veloci) e **registri** (normali, a scorrimento, ad anello).

Tabella di verità:

D	Q+	Descrizione
0	0	reset
1	1	set



Simbolo circuitale  (ISO) per flip-flop di tipo D, dove \triangleright è l'ingresso del clock, D è l'ingresso del dato e Q è l'uscita del dato memorizzato.

Otteniamo così il circuito di figura 5. Quando il segnale di clock è zero sia S che R sono zero ed il circuito si trova nella fase di memorizzazione. Quando invece il clock è uguale ad 1 risulta: $S=D$ ed inoltre $R=\bar{D}$; in questo modo ogni variazione dell'ingresso D viene riprodotta in uscita ed il latch è trasparente.

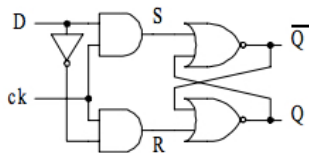


Fig. 5 D latch statico.

Organizzazione di una piccola memoria

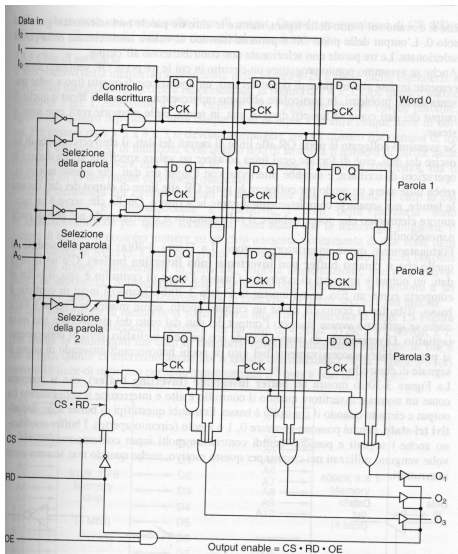


Figura 3-29 Diagramma logico di una memoria 4 x 3. Ogni fila rappresenta una delle quattro parole da 3 bit. Un'operazione read o write legge o scrive sempre una parola completa.

