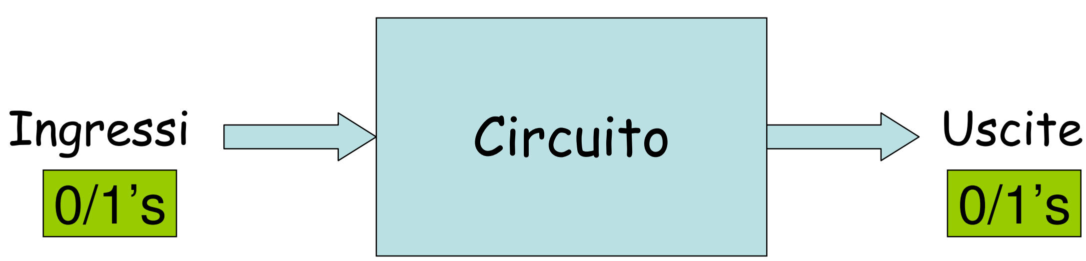
Parte III

Hardware

Circuiti Digitali



- Un circuito (digitale) puo' essere descritto
  - Mostrando i dettagli realizzativi in termini di circuiti elementari (porte)
  - Mediante una tavola di verita' che mostra i valori in uscita x tutti i possibili ingressi

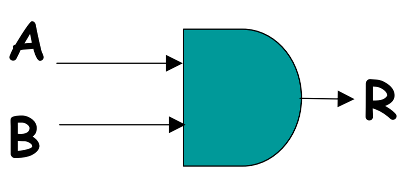
L' Hardware di un computer

I componenti di un computer sono realizzati con un **gran numero** di componenti elettroniche "molto semplici" dette *porte logiche*

Solamente 3 tipi di porte (base):

- AND ("e")
- OR ("o")
- NOT ("non")

AND



A	B	R
0	0	0
0	1	0
1	0	0
1	1	1

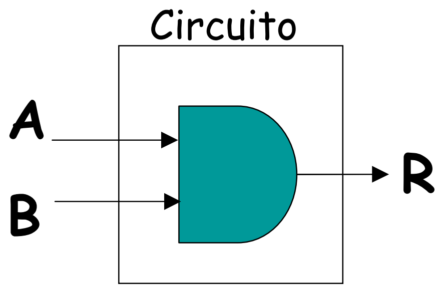
Tavola di verità

1 = VERO, 0 = FALSO

1 = tensione, 0 = no tensione

Fornisce tensione all'output **R** se e solamente se vi e` tensione in **entrambi** gli input **A** e **B**

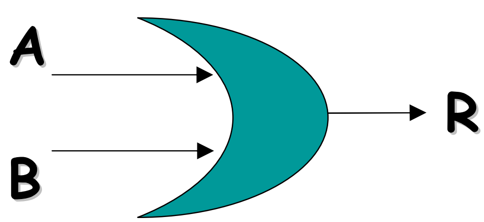
tavola di verità ⇔ circuito



A	B	R
0	0	0
0	1	0
1	0	0
1	1	1

- Dato un qualsiasi circuito e' sempre possibile definire la tavola di verita' (in un solo modo)
- Data una tavola di verita' si possono costruire in generale piu' circuiti che la realizzano

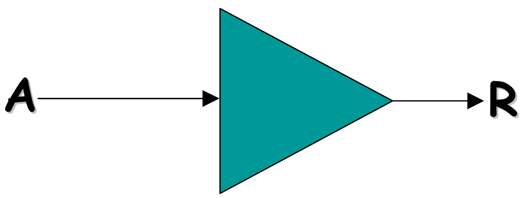
OR



A	B	R
0	0	0
0	1	1
1	0	1
1	1	1

Fornisce tensione all'output **R** se e solamente se vi e` tensione in **almeno uno** degli input ***A*** e **B**

NOT



A	R
0	1
1	0

Fornisce tensione all'output **R** se e solamente se **non** vi e` tensione all'input ***A***

Assemblando queste componenti possiamo costruire nuovi circuiti piu'complessi..

IMPLICAZIONE LOGICA

se *A* allora B $\qquad A \Rightarrow B$

ogni volta che *A* e' VERO anche B deve essere VERO

Tavola di verita'

$$A \Rightarrow B$$

A	B	$A \Rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

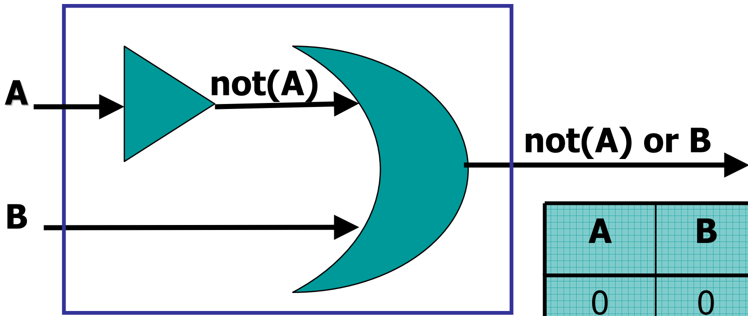
Come costruire un circuito che realizza questa tavola di verita`?

A	Ris
0	0/1
1	0/1

A	B	Ris
0	0	0/1
0	1	0/1
1	0	0/1
1	1	0/1

Ci sono $4 = 2^{(2^1)}$ possibili tabelle di verita` ad una entrata e $16= 2^{(2^2)}$ possibili tabelle di verita`a due entrate.
In generale, ci sono $2^{(2^N)}$ tabelle di verita` con N entrate.
I circuiti corrispondenti **si possono tutti realizzare** componendo i circuiti elementari AND, OR e NOT .

Il seguente circuito realizza l'implicazione logica $A \Rightarrow B$



A	B	not(A)	not(A) or B
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

Infatti vale che

$A \Rightarrow B \cong \text{not}(A) \text{ or } B$