

# Elettronica digitale – parte 1

Prof. Hajj Ali

Per info

[hajjali2000@yahoo.it](mailto:hajjali2000@yahoo.it)

# Elettronica digitale

L'**elettronica digitale** si occupa della **teoria** e dello **sviluppo** dei **circuiti logici** che, sono un insieme di dispositivi chiamati porte logiche opportunamente connessi. Progettate per essere resistente al rumore.

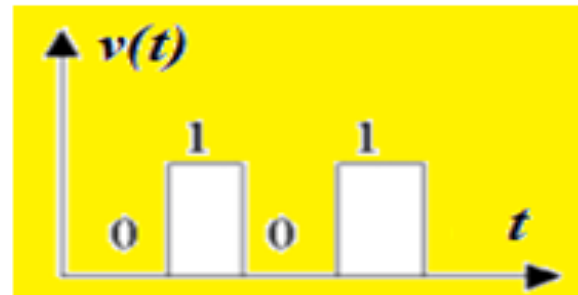
Il termine **digitale** viene dalla parola inglese **digit** che significa **cifra**.

**Le porte logiche** sono dispositivi capaci di eseguire operazioni logiche su **segnali binari**.

I **segnali binari** sono **livelli di tensione**, espressi mediante un numero definite come **cifre binarie (0 e 1)**, in inglese **binary digit**, più semplicemente **bit**.

0 logico = livello basso di tensione

1 logico = livello alto di tensione



# Elettronica digitale

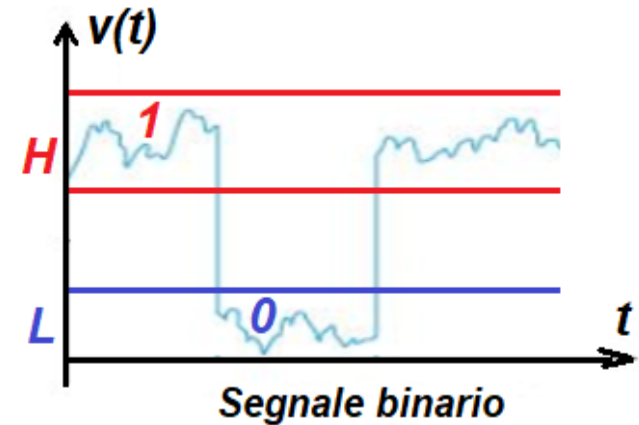
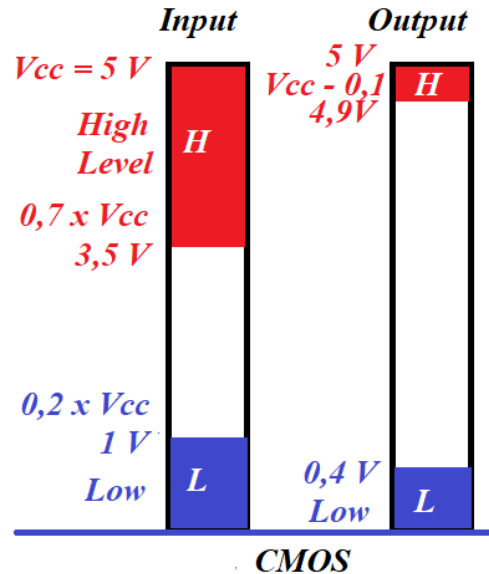
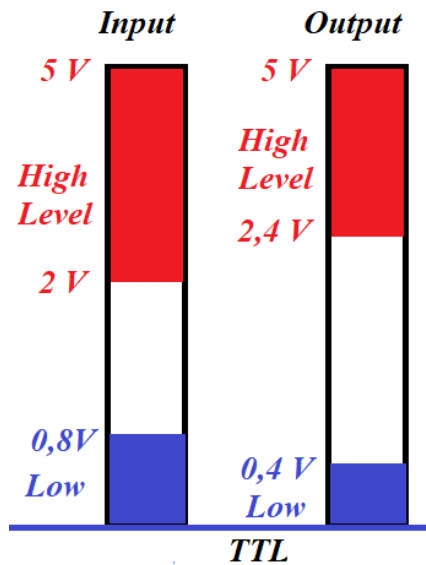
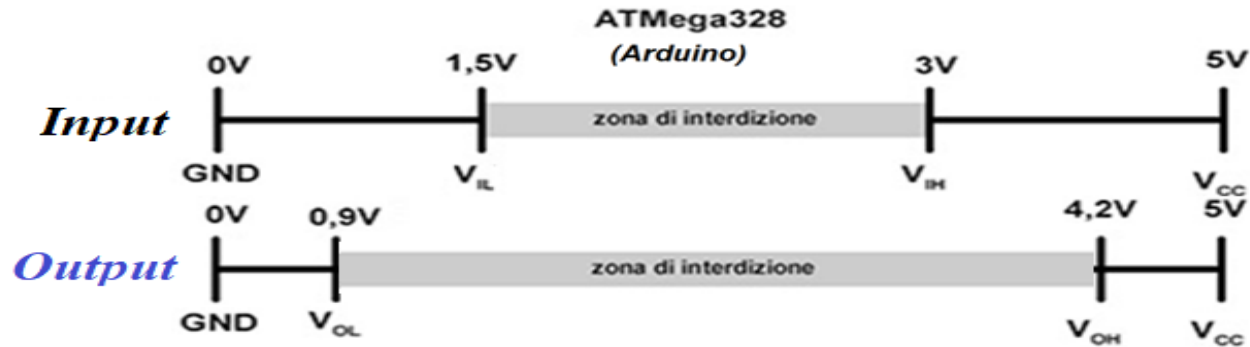
I **livelli logici** sono livelli di tensione elettrica che definiscono l'1 binario o lo 0 binario, possono differire in funzione della tecnologia utilizzata per la costruzione dei Dispositivi elettronici digitali.

Le **famiglie logiche** più utilizzate sono la **TTL (Transistor Transistor Logic)** e la **CMOS (Complementary MOS)**: (Transistor ad Effetto di Campo Metallo Ossido Semiconduttore).

I livelli logici sono identificati tramite una coppia di simboli:

0	Livello basso	Low level (L)	False (F)	Open
1	Livello alto	High level (H)	True (T)	Close

# Elettronica digitale



# Alcune definizioni

**Variabile logica**: è una variabile che può assumere solo due valori, che normalmente vengono denominati 0 e 1.

0	Livello basso	Low level (L)	False (F)	Open
1	Livello alto	High level (H)	True (T)	Close

**Funzione logica oppure funzione di commutazione**: rappresenta il legame esistente tra **variabili d'ingresso (indipendente)** e la **variabile di uscita (dipendente)** di una rete logica.

# Alcune definizioni

Una **rete logica** è un circuito costituito da **porte logiche**.

Le **porte logiche** (logic gates):

- sono i **circuiti minimi** per l'elaborazione di segnali binari;
- rappresentano la realizzazione elettronica degli operatori dell'**algebra di Boole**.

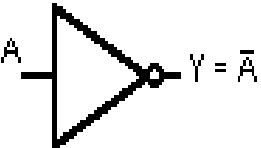

**Algebra di Boole:** è lo strumento matematico che consente di rappresentare le funzioni logiche semplificate attraverso opportune regole.

**Tabella della verità** (Truth table): è un modo alternativo alla funzione logica di commutazione, per **esprimere** la **relazione** tra **ingressi** e **uscite** di una rete logica. In una tabella di verità sono rappresentate in colonna tutte le possibili combinazioni di valori che possono assumere le variabili logiche ( variabili booleane ). La colonna finale della tabella di verità è dedicata a mostrare il risultato dell'espressione.

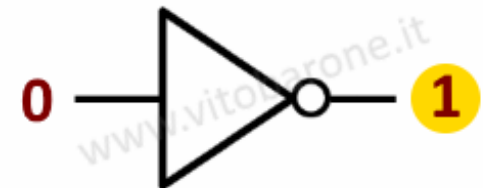
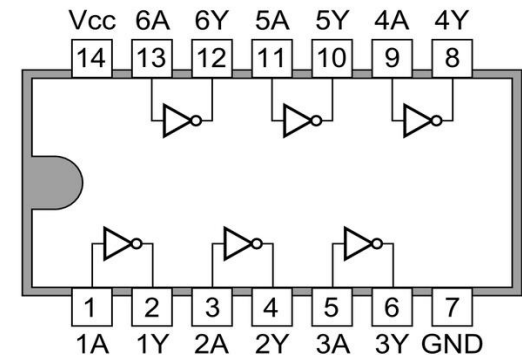
# Porte logiche fondamentali

Le **porte logiche fondamentali** rappresentano gli elementi principali di una rete logica e rappresentano i mattoni dell'elettronica digitale. Sono principalmente tre: **NOT, AND e OR.**

la **porta NOT**: ha un solo ingresso ed una sola uscita.  
Essa **inverte** lo stato logico presente all'ingresso.

Simbolo grafici -Tradizionale - IEEE/IEC	Funzione logica	Tabella di verità						
	$Y = \overline{A}$ <p>e si legge A negato oppure</p> <p>A complementato.</p>	<table border="1"><thead><tr><th>A</th><th>Y</th></tr></thead><tbody><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></tbody></table>	A	Y	0	1	1	0
A	Y							
0	1							
1	0							
								

7404 Hex Inverters



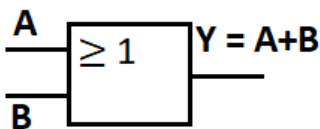
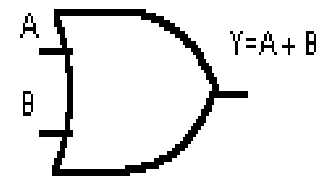
IEEE: Institute of Electrical and Electronics Engineers / IEC: International Electrotechnical Commission

# Porte logiche fondamentali

## La porta OR:

- effettua la somma logica;
- il minimo degli ingressi è due.
- *L'uscita assume lo stato logico 1 se almeno una variabile di ingresso è allo stato logico 1.*

Simbolo grafici  
-Tradizionale  
- IEEE/IEC



Funzione logica

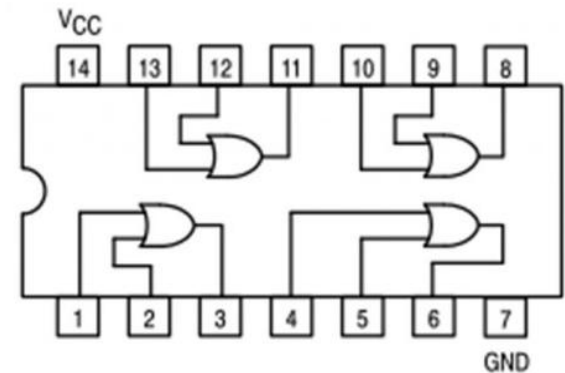
$$Y=A+B$$

si legge A or B.

Tabella di verità

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

SN74LS32



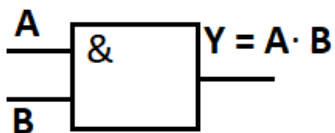


# Porte logiche fondamentali

## La porta AND:

- effettua il prodotto logica;
- il minimo degli ingressi è due.
- *l'uscita assume lo stato logico 1 solo se tutte variabile di ingresso sono allo stato logico 1.*

Simbolo grafici  
- Tradizionale  
- IEEE/IEC



Funzione logica

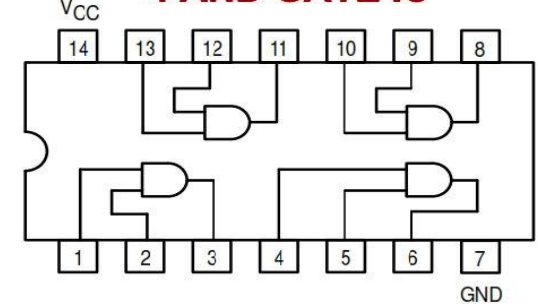
$$Y = A * B$$

si legge A and B.

Tabella di verità

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

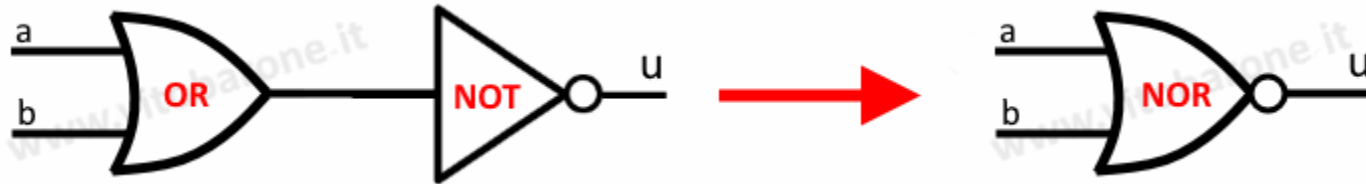
## 7408 4 AND GATE IC



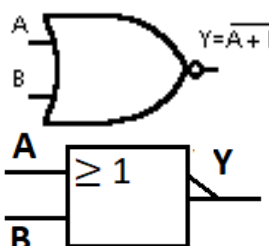
# Porte logiche derivate

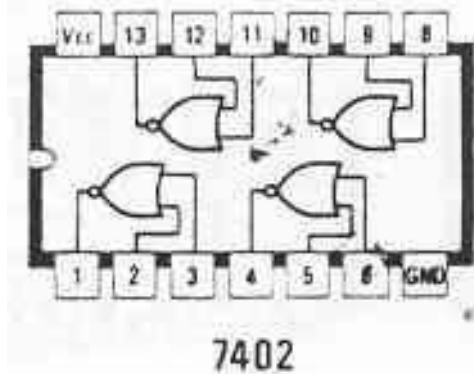
**Le porte logiche derivate:** Essi si ottengono dalla combinazione tra le porte logiche principali (NOT, OR e AND), sono: NOR, NAND, XOR e XNOR.

**porta logica NOR:** è una combinazione tra OR e NOT;



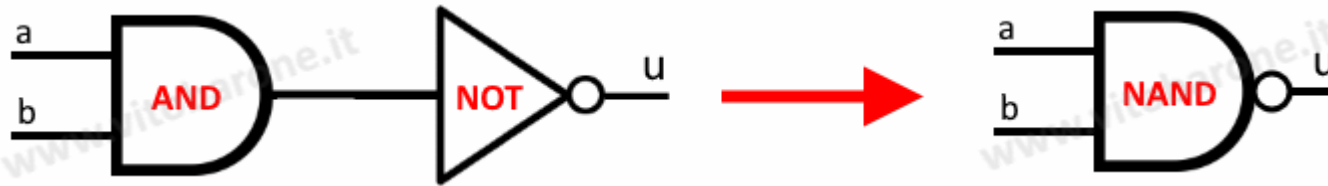
- *l'uscita assume lo stato logico 0 se almeno una variabile di ingresso è allo stato logico 1. In tutti gli altri casi  $Y=1$ .*

Simbolo	Funzione logica	Tabella di verità															
	$Y = \overline{A+B}$  si legge A nor B.	<table border="1"><thead><tr><th>A</th><th>B</th><th>Y</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>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	0
A	B	Y															
0	0	1															
0	1	0															
1	0	0															
1	1	0															

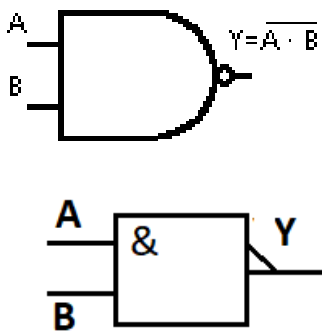


# Porte logiche derivate

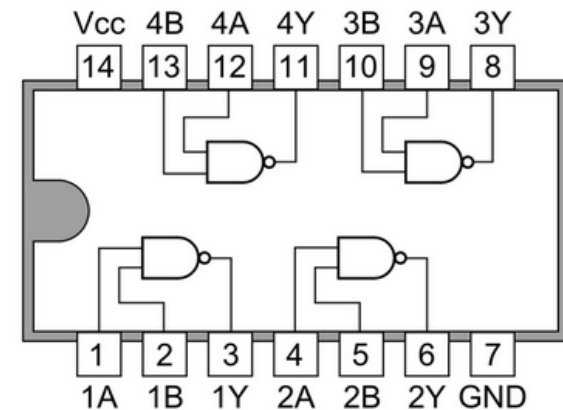
**La porta logica NAND:** è una combinazione tra AND e NOT;



- *l'uscita assume lo stato logico 0 se tutte le variabili di ingresso sono allo stato logico 1. In tutti gli altri casi  $Y=1$ .*

Simbolo	Funzione logica	Tabella di verità															
	$Y = A \cdot B$ <p>si legge <u>A nand B</u>.</p>	<table border="1"><thead><tr><th>A</th><th>B</th><th>Y</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	Y	0	0	1	0	1	1	1	0	1	1	1	0
A	B	Y															
0	0	1															
0	1	1															
1	0	1															
1	1	0															

7400 Quad 2-input NAND Gates

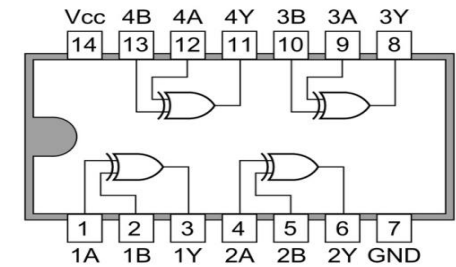


# Porte logiche derivate

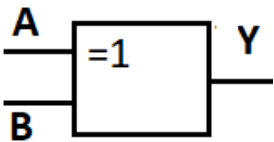
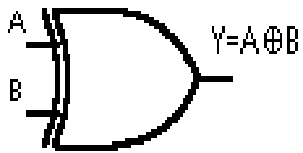
**La porta logica XOR:** l'XOR opera su due soli ingressi.

- L'uscita vale 1 se gli ingressi assumono valore diverso,
- L'uscita vale 0 se gli ingressi sono tra loro uguali.
- È definito un comparatore digitale.

7486 Quad 2-input ExOR Gates



**Simbolo**



**Funzione logica**

$$Y = A \oplus B$$

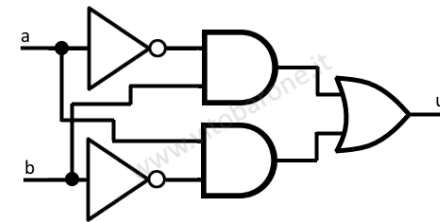
si legge *A or esclusivo B*  
oppure *A diverso da B*

**Tabella di verità**

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0



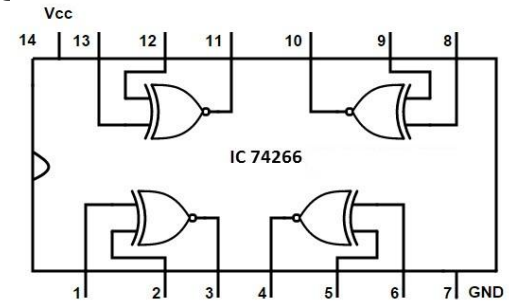
L'XOR si ottiene combinando le tre porte NOT, AND e OR, Come mostrato in figura.



# Porte logiche derivate

**La porta logica XNOR:** l'XNOR è la negazione della porta XOR.

- opera su due soli ingressi.
- L'uscita vale 0 se gli ingressi assumono valore diverso,
- L'uscita vale 1 se gli ingressi sono tra loro uguali.
- È definito un comparatore digitale.





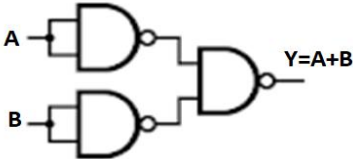
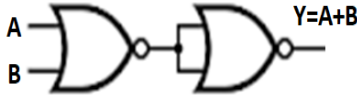
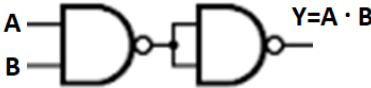
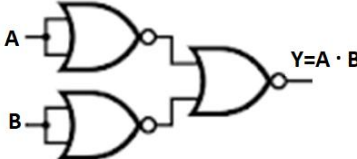
Simbolo	Funzione logica	Tabella di verità															
<p><math>Y = A \odot B = \overline{A \oplus B}</math></p> <p>si legge <i>A nor esclusivo B</i> oppure <i>A coincidente con B</i></p>		<table border="1"><thead><tr><th>A</th><th>B</th><th>Y</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>1</td></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>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	1
A	B	Y															
0	0	1															
0	1	0															
1	0	0															
1	1	1															

Si può ottenere una *porta XOR* combinando

le tre porte NOT, AND e OR, come mostrato in figura.

# Porte logiche universali

**Le porte logiche universali:** Il porte logiche **NAND** e **NOR** vengono dette anche **universali**, in quanto è possibile realizzare qualunque funzione di commutazione utilizzando solo NANA e solo NOR.

Porta logica	Con operatori NAND	Con operatori NOR
NOT	 $Y = \bar{A}$	 $Y = \bar{A}$
OR	 $Y = A + B$	 $Y = A + B$
AND	 $Y = A \cdot B$	 $Y = A \cdot B$

# Forme canoniche

**Forme canoniche:** Permette di trovare l'espressione logica (non minima) corrispondente ad una tabella di verità.

**1<sup>a</sup> forma canonica** detta anche **S.O.P.** ( Sum Of Products, **somma dei prodotti**)

Per rappresentare una generica tabella di verità mediante la prima forma canonica bisogna:

- considerare **solo le righe** della tabella in corrispondenza delle quali **l'uscita vale 1**;
- ognuna di tali righe corrisponde a un **prodotto logico AND** (detto **mintermine**) di tutte le variabili, vengono **negate** (NOT) se **valgono zero** e dirette se valgono uno;
- **l'uscita** è data dalla **somma logica OR** di tutti i mintermini così individuati.

**2<sup>a</sup> forma canonica** (Prodotto delle somme) Per la seconda forma canonica bisogna:

- considerare solo le righe della tabella in corrispondenza delle quali l'uscita vale 0;
- ognuna di tali righe corrisponde a una somma logica OR (detto *maxtermine*) di tutte le variabili, prese negate se valgono uno e dirette se valgono zero;
- l'uscita è data dal prodotto logico AND di tutti i maxtermini così individuati.

# 1<sup>a</sup> forme canoniche

**Esercizio risolto con la prima forma canonica:** data la seguente di tabella

Consideriamo tutte le righe(tre) in corrispondenza delle quali l'uscita vale 1 e scriviamo per ognuna di esse il mintermine corrispondente:

A	B	C	Y	Mintermine
0	0	0	0	
0	0	1	0	
0	1	0	1	$\bar{A} \cdot B \cdot \bar{C}$
0	1	1	0	
1	0	0	1	$A \cdot \bar{B} \cdot \bar{C}$
1	0	1	0	
1	1	0	1	$A \cdot B \cdot \bar{C}$
1	1	1	0	

L'espressione logica in forma canonica si ottiene facendo la somma (OR) dei mintermini trovati:

$$Y = \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot \bar{C} + A \cdot B \cdot \bar{C}$$

Esercizio: progettare il circuito della porta logica XNOR



# 1<sup>a</sup> forma canonica - minimizzazione

- ad una tabella di verita' possono corrispondere piu' circuiti
- da un punto di vista progettuale, occorre costruire circuiti con il **minimo numero** di porte logiche.

in generale, per la semplificazione/minimizzazione di una forma canonica, si ricorre spesso (usando i **teoremi dell'algebra booleana**).

Per esempio, considerando la prima forma canonica:  $Y = \bar{A} \cdot \bar{B} \cdot \bar{C} + A \cdot \bar{B} \cdot \bar{C} + A \cdot B \cdot \bar{C}$

raccogliendo  $\bar{C}$  abbiamo:  $Y = (\bar{A} \cdot \bar{B} + A \cdot \bar{B} + A \cdot B) \cdot \bar{C}$

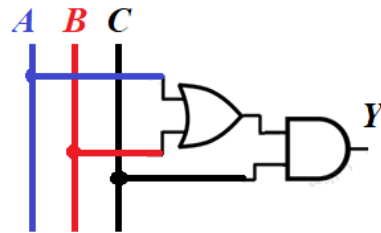
Procedendo ancora a raccogliere A nella parentesi tonda:  $Y = (\bar{A} \cdot \bar{B} + A \cdot (\bar{B} + B)) \cdot \bar{C}$

ma essendo  $\bar{B} + B = 1$  (assioma del completamento):  $Y = (\bar{A} \cdot \bar{B} + A \cdot (1)) \cdot \bar{C}$

$A \cdot (1) = A$  e allora:  $Y = (\bar{A} \cdot \bar{B} + A) \cdot \bar{C}$

da cui infine usando il secondo teorema dell'assorbimento abbiamo:  $Y = (A + \bar{B}) \cdot \bar{C}$

La rete logica corrispondente:



# Fonti

<https://giphy.com/gifs/>

<http://www.edutecnica.it/elettrotecnica/alternata/alternata.htm>

[http://www.vitobarone.it/elettronica/porte\\_logiche.htm](http://www.vitobarone.it/elettronica/porte_logiche.htm)

<http://www.dacrema.com/Informatica/portelogic.htm>

