

IL LIVELLO LOGICO DIGITALE

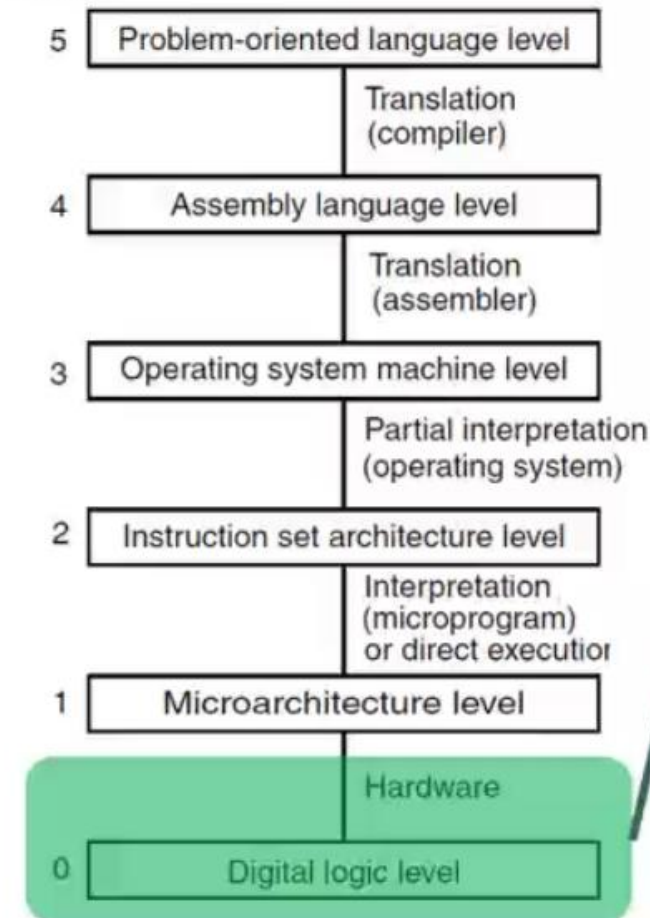
(Prima parte)

Argomenti

- Il livello logico digitale:
 - algebra di Boole;
 - le trasformazioni nel dominio di Boole.
- Circuiti logici digitali elementari:
 - circuiti Integrati;
 - circuiti combinatori e sequenziali;
 - circuiti combinatori:
 - Multiplexer;
 - Decoder;
 - Comparatori;
 - Programmable Logic Arrays.
 - Circuiti Aritmetici:
 - Shifter;
 - Adder;
 - Arithmetic Logic Units.
 - Clock.

Il livello logico digitale

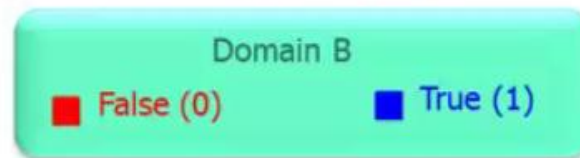
Livello



- Il livello logico digitale è l'hardware del calcolatore.
- È fatto di piccoli dispositivi elettronici chiamati porte logiche che lavorano con l'algebra di Boole.
- Utilizzando questi piccoli mattoncini si possono comporre funzioni complesse come sommatore, shifter, memorie,...

Algebra di Boole

- George Boole (1815–1864) è il matematico che ha ideato quest'algebra basata su un insieme (B) di due valori: true e false.
- Per convenzione al valore logico true corrisponde il simbolo 1 e a false il simbolo 0.



- Sul dominio B, si possono definire delle funzioni booleane:

funzioni unarie: $B \longrightarrow B$

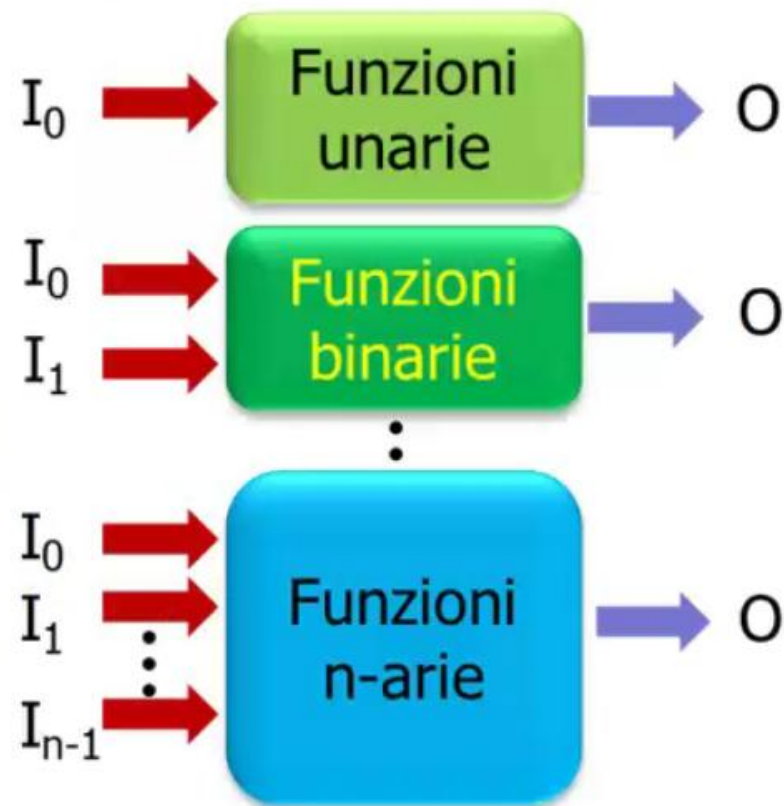
funzioni binarie: $B \times B \longrightarrow B$

...

funzioni n-arie: $B \underbrace{\times \dots \times}_n B \longrightarrow B$

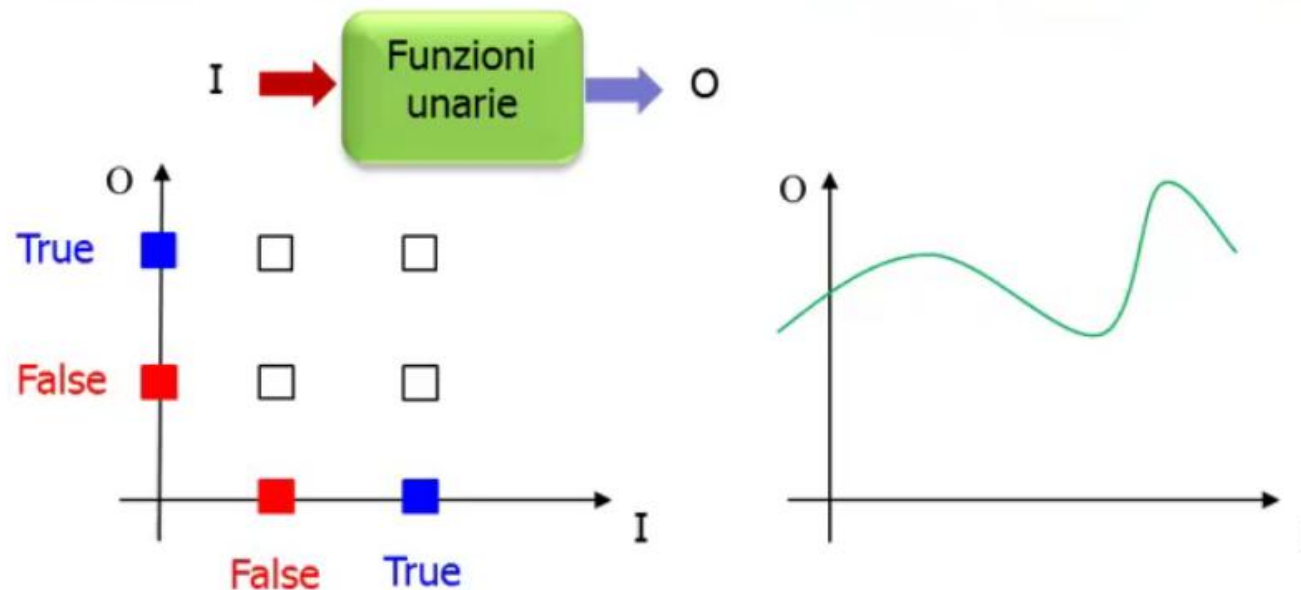
Funzioni Booleane come scatole nere

- Si possono immaginare le funzioni logiche come delle scatole nere che ricevono in ingresso e restituiscono in uscita variabili logiche.



Confronto tra funzioni booleane e funzioni reali

- Il comportamento di una funzione booleana è semplice da descrivere, perché gli ingressi/uscite variano in un insieme finito di valori.
- La **tavola di verità** è lo strumento che permette di esplorare esaustivamente tutte le possibili combinazioni.



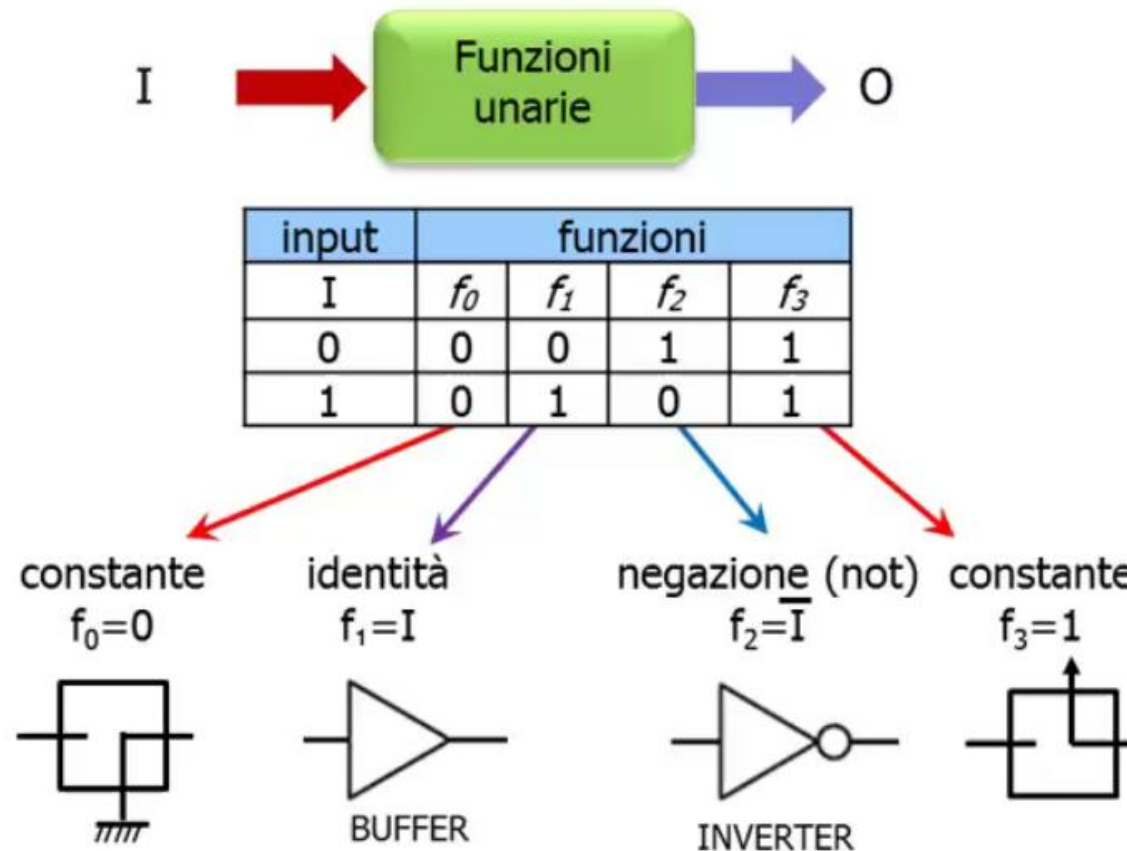
L'insieme delle possibili funzioni booleane

- Con n ingressi si hanno 2^n combinazioni che originano 2^{2^n} possibili funzioni.



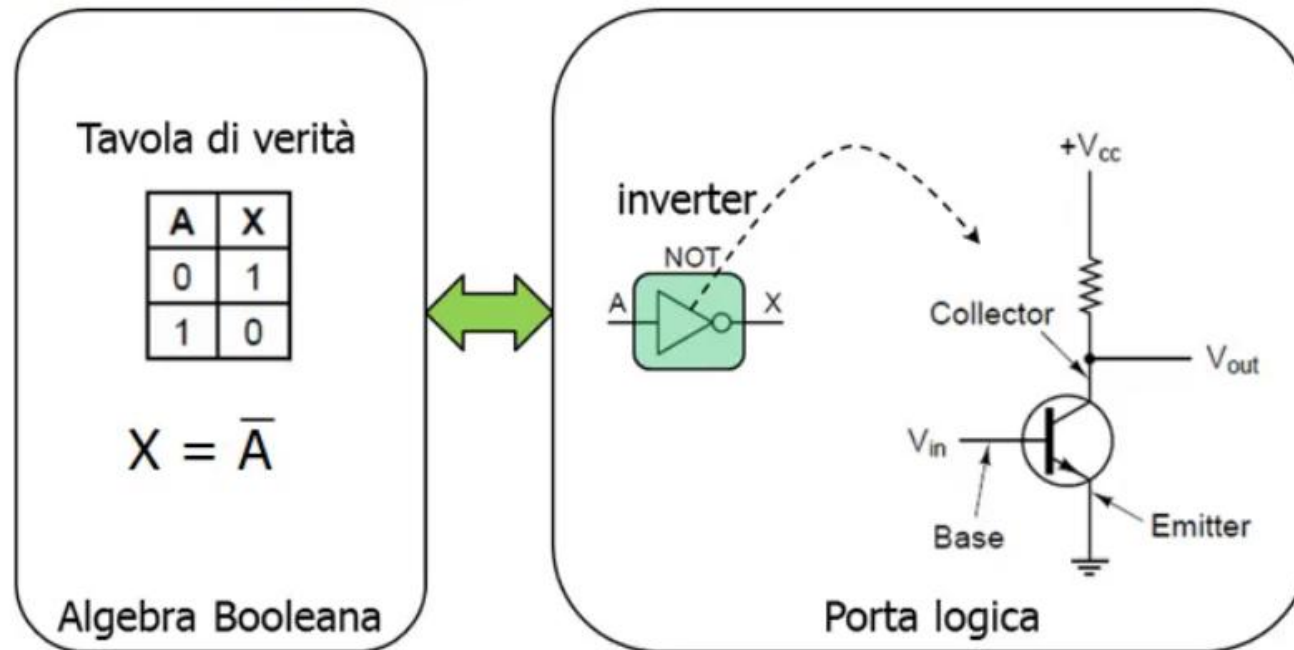
Unary Boolean functions

- Le funzioni ad un solo operando sono $2^2=4$.



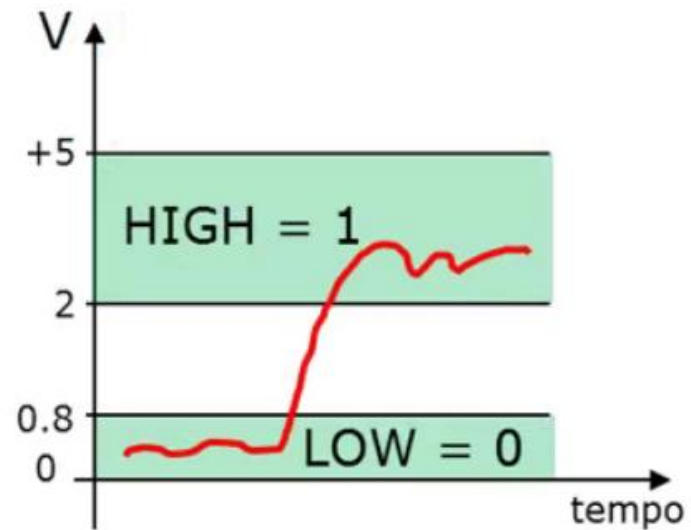
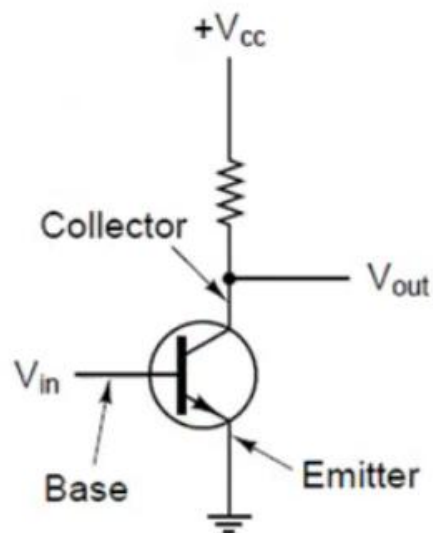
Negation (NOT operator)

- Esiste una stretta correlazione tra algebra booleana e circuiti digitali.
- La negazione è una funzione che inverte il valore della variabile in ingresso.



Segnali digitali

- La relazione che sussiste tra livelli logici digitali e reali segnali analogici:

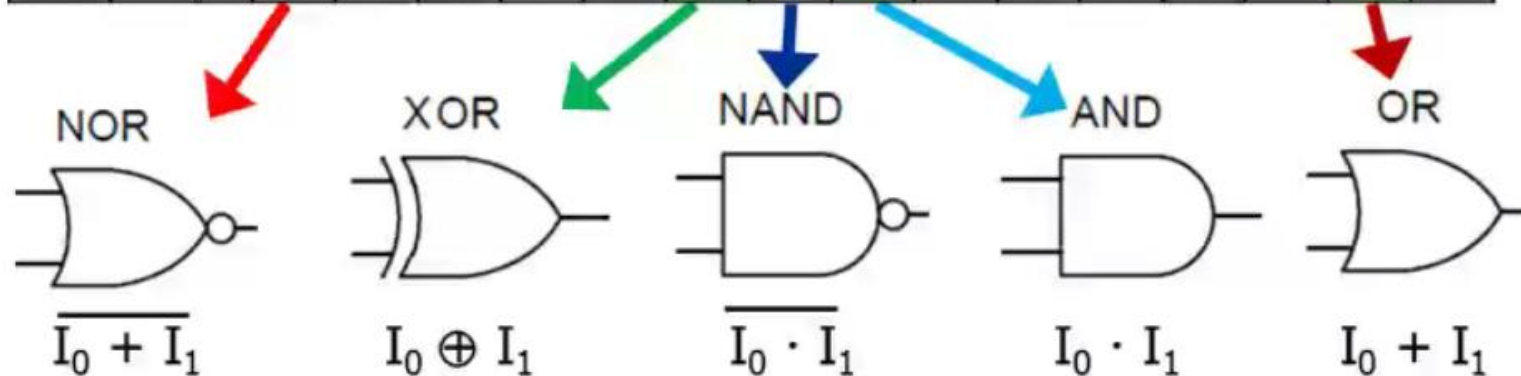


Livelli logici digitali dei TTL

L'insieme delle possibili funzioni booleane binarie



ingressi		funzioni															
I_0	I_1	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1



Proprietà dell'algebra di Boole

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\bar{A} = 0$	$A + \bar{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A}\bar{B}$

Come verificare la validità di un'equivalenza?

$$A + AB \stackrel{?}{=} A$$

- Utilizzando il metodo algebrico attraverso le proprietà:

$$A + AB = A(1 + B) = A$$

- Utilizzando la tavola di verità con il metodo esaustivo:

A	B	AB	A + AB	A
0	0	0	0	0
0	1	0	0	0
1	0	0	1	1
1	1	1	1	1

Le due espressioni sono identiche per ogni combinazione degli ingressi

- Utilizzando la logica e le definizioni: la variabile A è già presente a sinistra dell'operatore OR perciò la sua presenza nel gruppo di AND ha lo scopo di ridurre B ad A (quindi non aggiunge nulla a B).

Come verificare la validità di un'equivalenza?

$$A + \overline{A} B \stackrel{?}{=} A + B$$

- Utilizzando il metodo algebrico attraverso le proprietà (distributiva):

$$A + \overline{A} B = (A + \overline{A})(A + B) = A + B$$

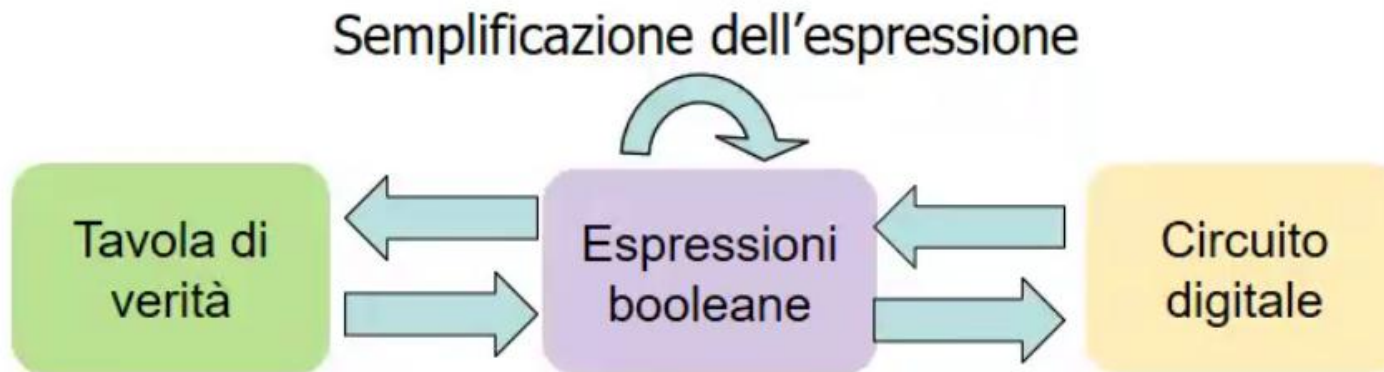
- Utilizzando la tavola di verità con il metodo esaustivo:

A	B	\overline{A}	$\overline{A} B$	$A + \overline{A} B$	$A + B$
0	0	1	0	0	0
0	1	1	1	1	1
1	0	0	0	1	1
1	1	0	0	1	1

Le due espressioni sono identiche per ogni combinazione degli ingressi

- Utilizzando la logica e le definizioni: la variabile A è già presente a sinistra dell'operatore OR perciò la sua presenza nel gruppo di AND in forma negata ha lo scopo di ridurre B ad A negato (quindi se A è true l'espressione è true mentre se A è falsa il risultato è B).

Le trasformazioni nel dominio di Boole

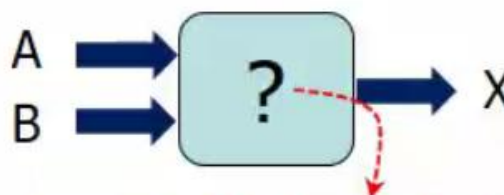


- Per ogni espressione logica c'è un circuito digitale equivalente e una colonna della tavola di verità.
- Per ogni colonna della tavola di verità c'è una espressione che la rappresenta e un corrispondente circuito digitale.
- Per ogni circuito digitale c'è una espressione che lo descrive ed una corrispondente colonna della tavola di verità.
- Ora si analizzeranno la trasformazione di una colonna della tavola di verità nella corrispondente espressione logica.

Tavola di verità → espressione logica

- Si vuole tradurre una colonna (X) della tavola di verità in funzione delle variabili A e B, cioè $X = F(A, B)$.

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0



Il comportamento del circuito è descritto completamente dalla tavola di verità

- Si considerano il minor numero di valori 0 oppure 1:
True 1: X è composta da un insieme di gruppi AND che contengono tutte le variabili (negate quelle che hanno il valore 0 nella riga corrispondente), legati con OR.
False 0: X è composta da un insieme di gruppi OR che contengono tutte le variabili (negate quelle che hanno il valore 1 nella riga corrispondente), legati con AND.

Tavola di verità → espressione logica (1)

- X è composto da tanti gruppi AND quanti sono i valori **true** nella colonna della tavola di verità.
- Ogni gruppo di AND contiene tutte le variabili in ingresso, sono negate quelle che sono **false** nella riga corrispondente.

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

$\bar{A} \cdot B$

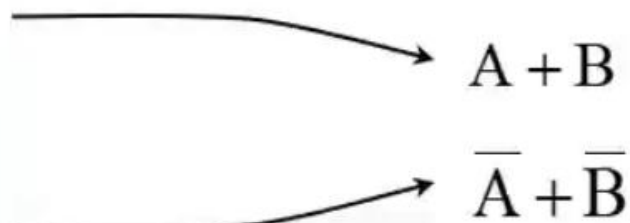
$A \cdot \bar{B}$

$$X = \bar{A} \cdot B + A \cdot \bar{B}$$

Tavola di verità → espressione logica (0)

- X è composto da tanti gruppi OR quanti sono i valori **false** nella colonna della tavola di verità.
- Ogni gruppo di OR contiene tutte le variabili in ingresso, sono negate quelle che sono **true** nella riga corrispondente.

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0



$A + B$

$\bar{A} + \bar{B}$

$$X = (A + B) \cdot (\bar{A} + \bar{B})$$

- ...so any Boolean function can be implemented using NOT, AND, and OR gates.

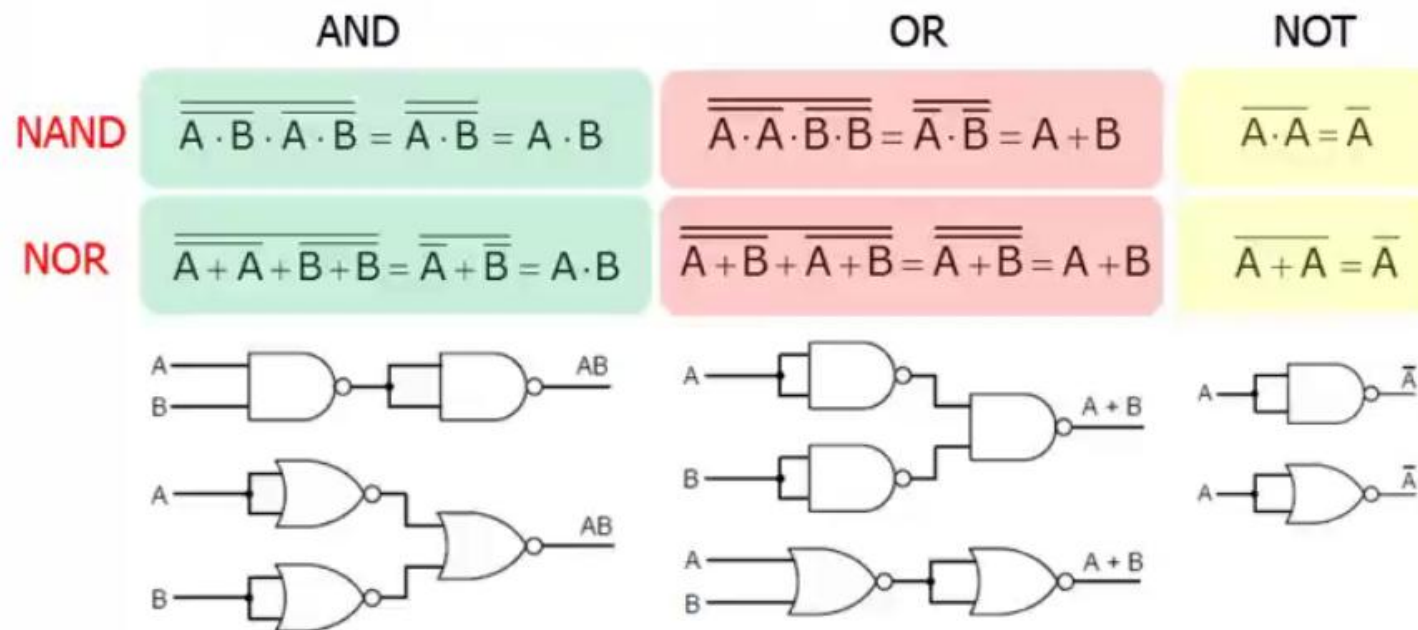
Gli operatori universali

- Abbiamo visto come ogni funzione logica booleana (una colonna della tavola di verità) si possa realizzare con gli operatori AND, OR e NOT
- È facile dimostrare che questi tre operatori possano essere realizzati con una sola porta NAND o NOR.
- Quindi ogni funzione logica booleana può essere realizzata con un solo operatore detto **universale**.



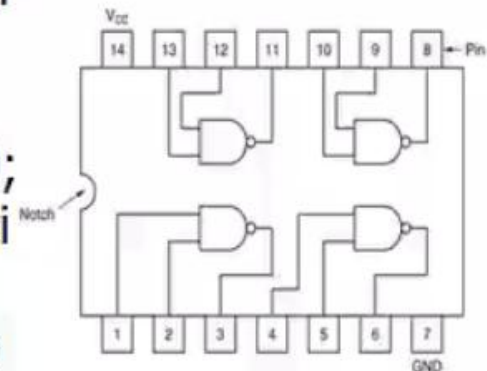
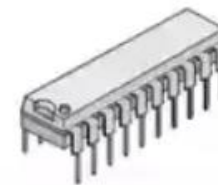
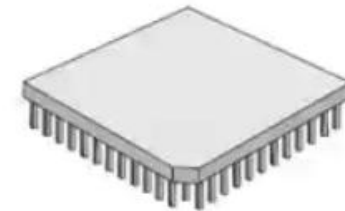
Gli operatori universali

- Tutte le porte logiche possono essere realizzate utilizzando solo porte NAND o NOR.
- La presenza di operatori universali è molto importante per la costruzione dei circuiti digitali.



Circuiti integrati

- Le porte logiche sono vendute individualmente ma dentro i **circuiti integrati** (or **chip**).
- Essi sono pezzi di silicio sul quale sono stampati dei circuiti.
- I chip possono essere divisi in classi in funzione del numero di porte logiche che contengono:
 - **SSI** (Small Scale Integrated) sono circuiti con meno di 10 porte;
 - **MSI** (Medium Scale Integrated) sono circuiti con più di 10 porte e meno di 100;
 - **LSI** (Large Scale Integrated) sono circuiti con più di 100 porte e meno di 100.000;
 - **VLSI** (Very Large Scale Integrated) sono circuiti con più di 100.000 porte.

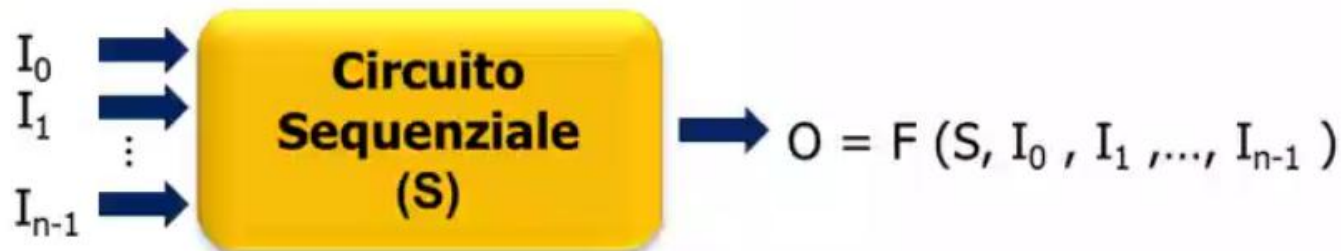


Circuiti combinatori e sequenziali

- I circuiti combinatori sono quei circuiti digitali nei quali l'uscita (O) dipende esclusivamente dagli ingressi (e non dallo stato del circuito, cioè dalla configurazione precedente degli ingressi):



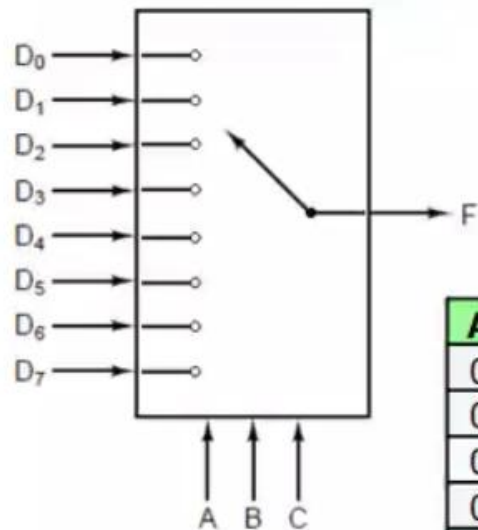
- In un circuito sequenziale l'uscita (O) dipende dagli ingressi e dallo stato (S) del circuito:



Multiplexer

- Il multiplexer è un circuito con 2^n ingressi, un' uscita e n ingressi di controllo (selettorie) che selezionano la linea in ingresso che verrà trasferita in uscita:

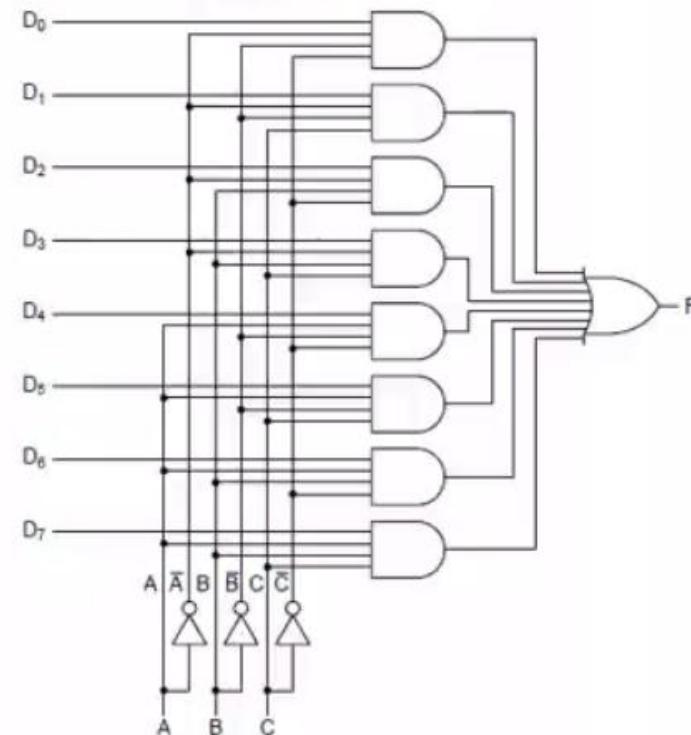
eight-input multiplexer



true table

A	B	C	F
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

inside a 8-input multiplexer



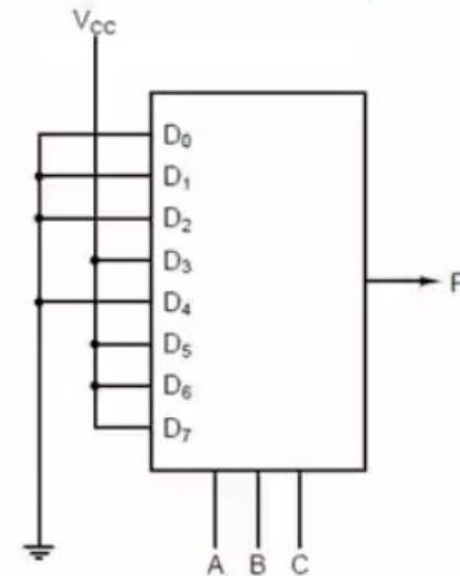
Multiplexer per realizzare funzioni booleane

- Il Multiplexer può essere utilizzato per realizzare qualsiasi funzione logica.
- Si consideri ad esempio la seguente funzione F che ha 3 ingressi: A, B e C:

Tavola di verità

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Il multiplexer è cablato per elaborare la funzione logica F



- Si può utilizzare un multiplexer con 3 ingressi di selezione come mostrato in figura.

Decoder (decodificatori)

- È un circuito che riceve un numero a n -bit come ingresso e seleziona in uscita l'unica linea corrispondente al suo valore numerico.
- È utile quando si dispone di un indirizzo e si vuole selezionare il chip di memoria corrispondente.

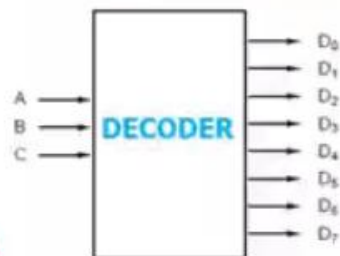
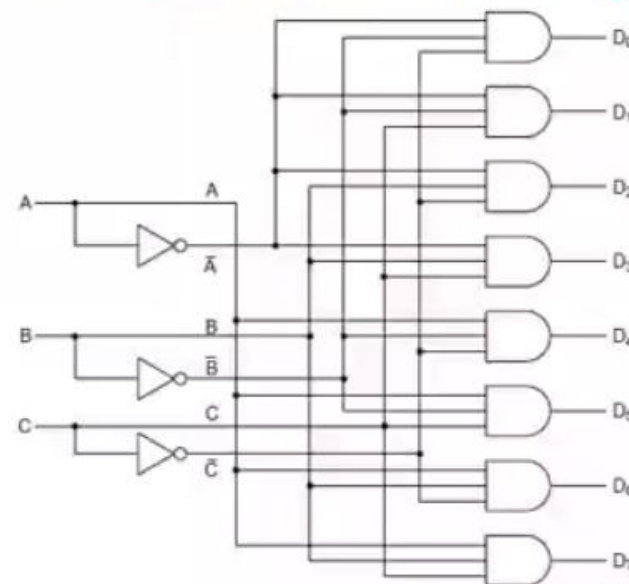


Tavola di verità

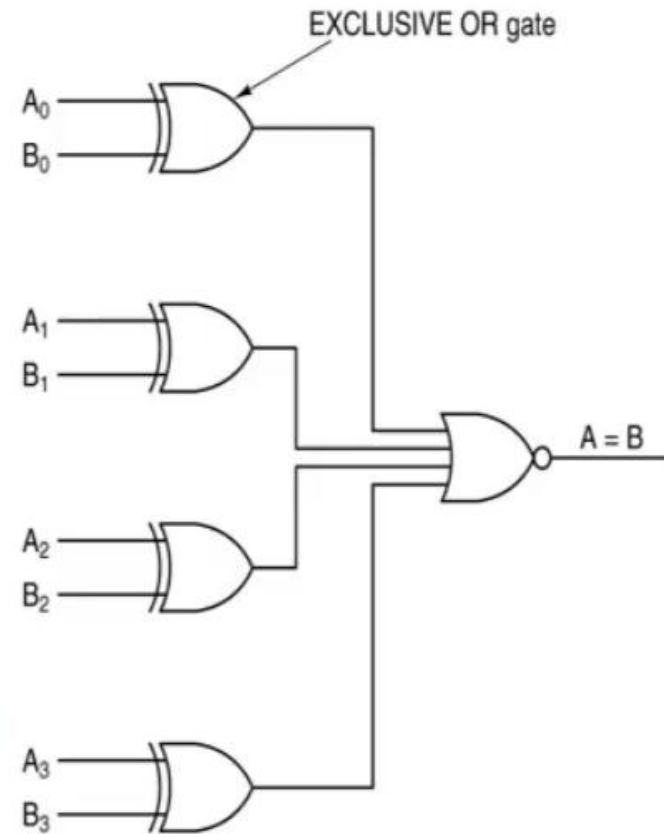
A	B	C	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Schema digitale di un decoder a 3 ingressi



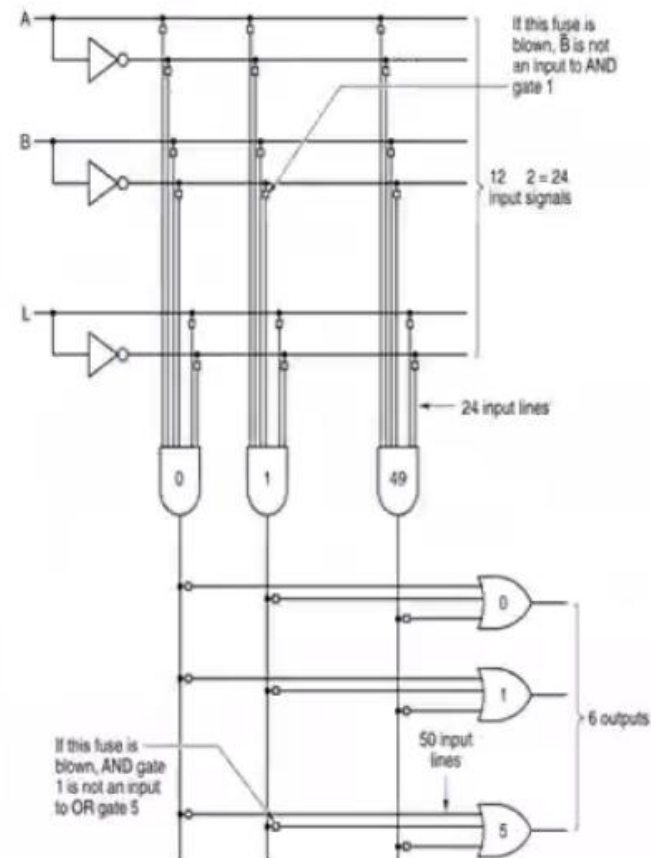
Comparatore

- Questo circuito compara due parole A e B di 4 bit.
- Il risultato è 1 se i bit della prima parola sono uguali a quelli della seconda, cioè: $A_i = B_i$ con $i=0, 1, 2$, e 3.
- Lo schema si basa sull'OR Esclusivo (XOR) delle due parole (eseguito bit-a-bit), i risultati parziali dei confronti tra i singoli bit (tutti 0 se le due parole sono identiche) sono inviati ad un NOR, che risponde 1 solo quando $A=B$.



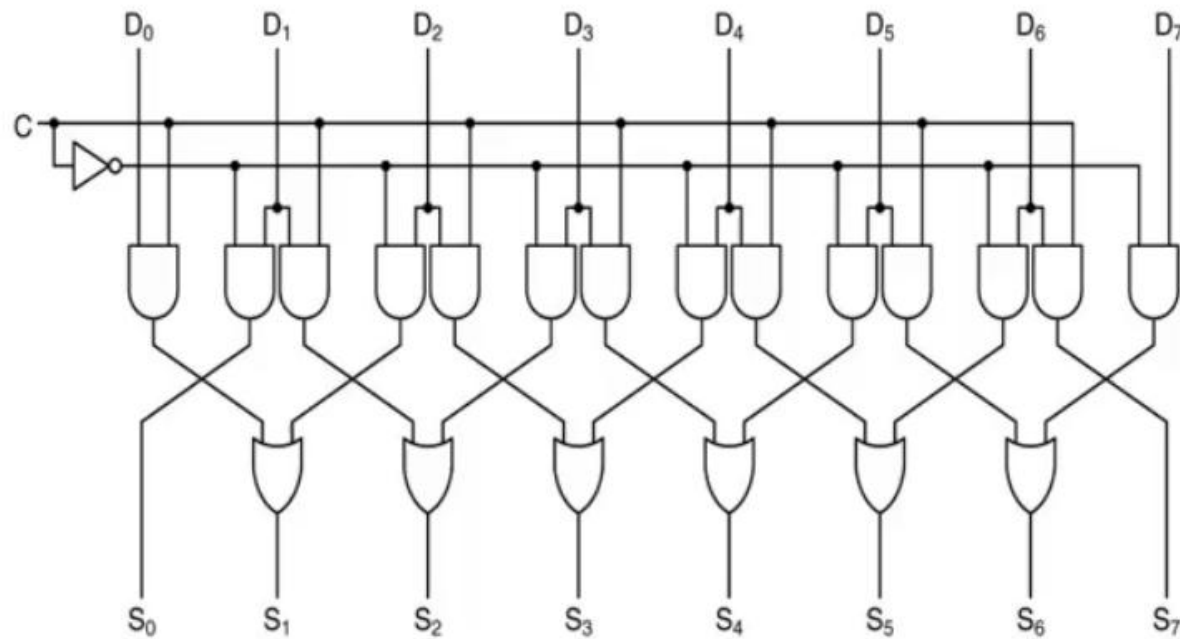
PLA (Programmable Logic Arrays)

- Un chip generico per la costruzione di una funzione arbitraria può essere costruito formando gruppi di AND e OR è la **PLA** (Programmable Logic Array).
- Un esempio di PLA con 12 ingressi è mostrato in figura.
- Il cuore del circuito è un array di 50 porte AND che creano una matrice 24×50 .
- Ogni linea di ingresso delle 50 porte AND contiene un fusibile.



Circuiti aritmetici – Lo shifter

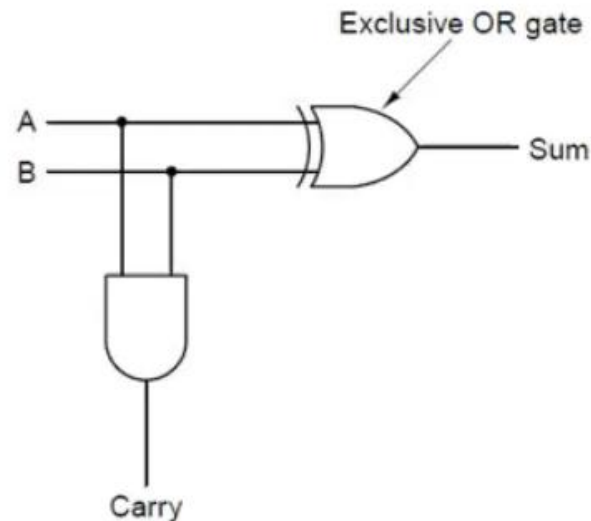
- L'uscita è una parola (nell'esempio a 8 bit) che verrà fatta scorrere a destra o sinistra di un bit.
- Il segnale C definisce la direzione dello scorrimento (0=sinistra, 1=destra).



Half Adder (Semi-sommatore)

- Un sommatore è un circuito in grado di eseguire la somma tra bit.
- Considerando due variabili binarie A e B, il circuito genera il risultato della somma e l'eventuale riporto.
- Questo circuito molto semplice è chiamato **half adder**, o semi-sommatore, poiché non gestisce un riporto in ingresso.

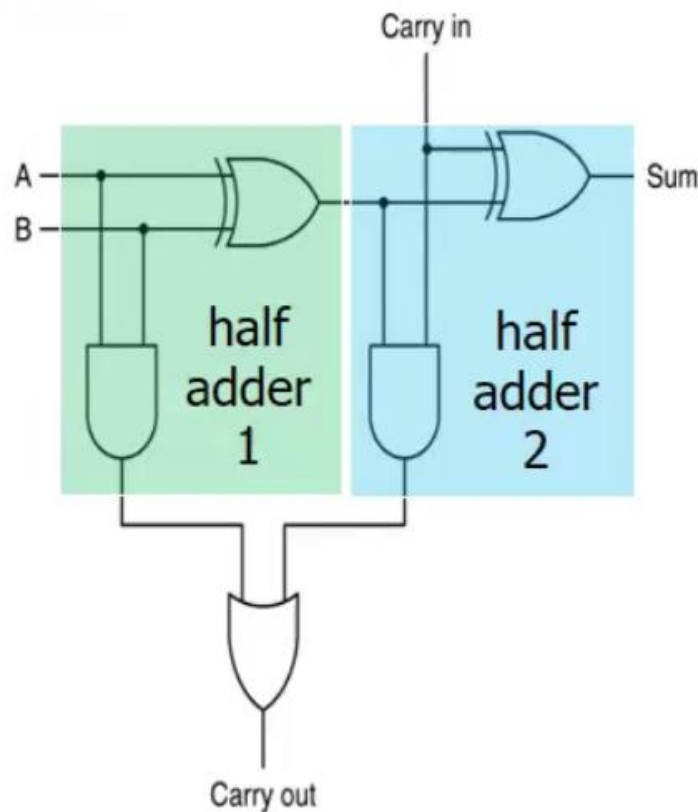
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Full Adder (Sommatore)

- Utilizzando due half-adders si può costruire un **full adder** che prende in ingresso tre bit (A, B e riporto in ingresso) e restituisce in uscita somma e riporto (carry out).

A	B	Carry in	Sum	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

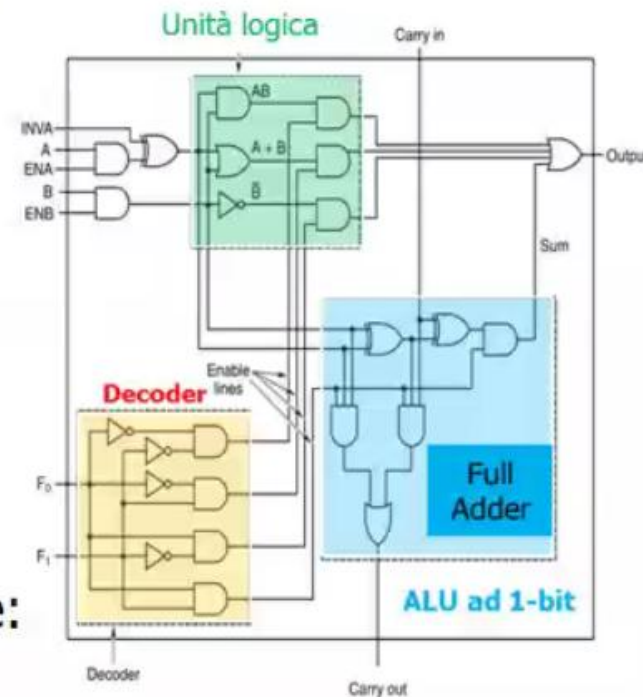


ALU (Arithmetic Logic Unit)

- L'unità ALU contiene 3 differenti unità: un decoder, una unità logica e un full adder.
- Il Decoder permette di selezionare l'operazione richiesta in base ai segnali F_0 e F_1 .

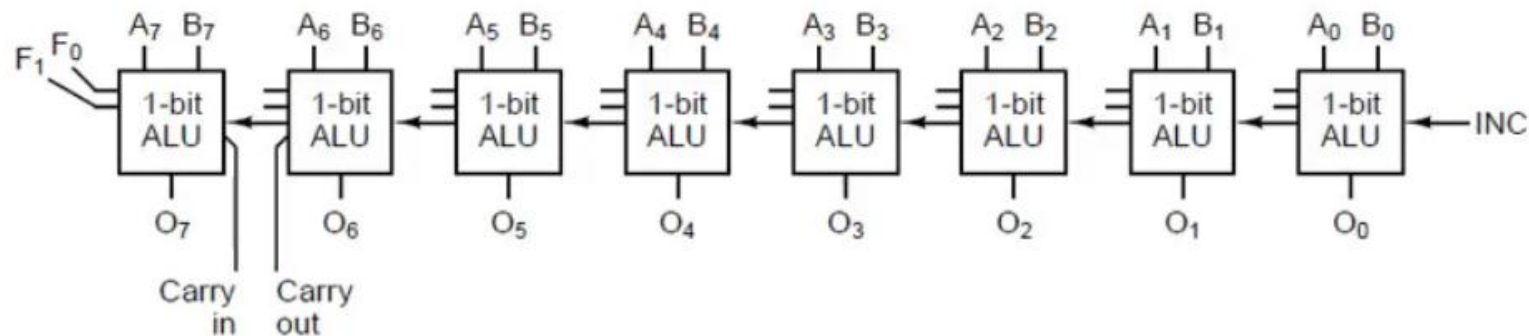
F_0	F_1	Output
0	0	A and B
0	1	A or B
1	0	not B
1	1	A + B + Carry in

- L'unità logica è in grado di calcolare: AB , $A+B$ e la negazione di B .
- Il full adder somma A , B e il riporto in ingresso e calcola il risultato e l'eventuale riporto (carry out).



ALU ad n-bit

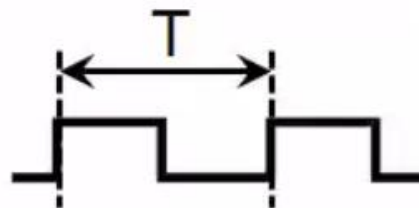
- Le ALU ad 1-bit possono essere assemblate insieme per costruire ALU di lunghezza variabile.
- Questa tecnica è detta **bit slice** (suddivisione di bit) e può essere applicata anche ai precedenti circuiti digitali che lavorano bit-a-bit.



Un ALU a 8-bit costruita con otto ALU ad un 1-bit.

Clock

- Il **clock** in questo contesto è un circuito che emette una serie di impulsi di larghezza predefinita a intervalli di tempo costanti.
- L'intervallo temporale compreso tra due fronti in salita (o discesa) di due impulsi consecutivi è detto **ciclo di clock** o **periodo (T)**.



Clock

- Una tecnica che permette di aumentare la risoluzione del segnale di clock (C1) è di effettuare un AND tra il segnale originario e una sua replica ritardata (C2).

