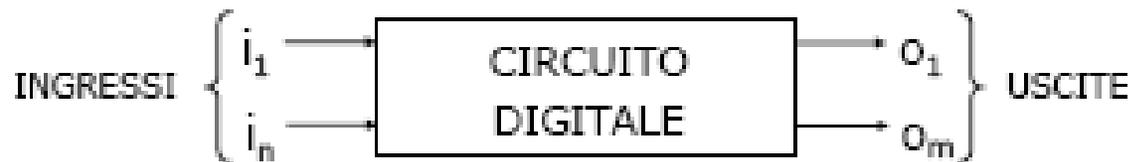




Cal Corso di Calcolatori Elettronici (per Elettronici)

Parte III: Circuiti digitali

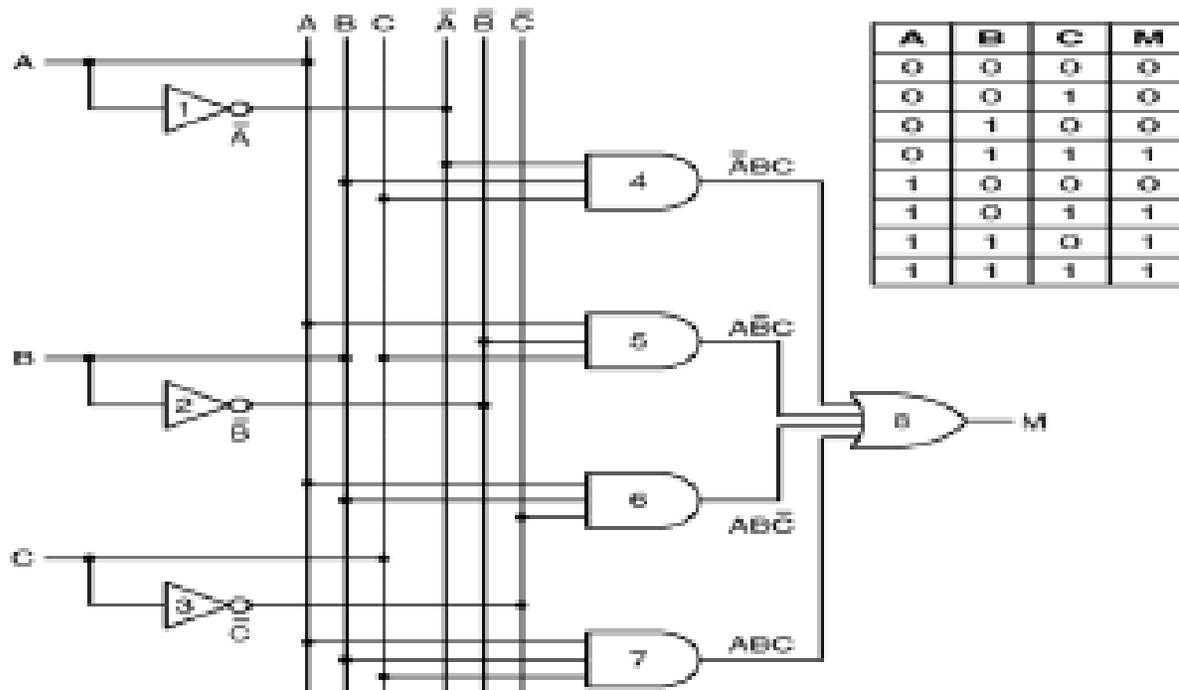
Circuiti digitali



- Circuiti elettronici i cui ingressi e le cui uscite assumono solo due livelli
- Al circuito sono associate le funzioni che calcolano le uscite a partire dagli ingressi

$$\begin{cases} o_1 = f_1(i_1, \dots, i_n) \\ \vdots \\ o_m = f_m(i_1, \dots, i_n) \end{cases}$$

Funzione di maggioranza

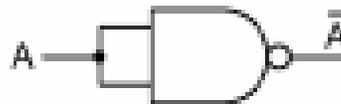
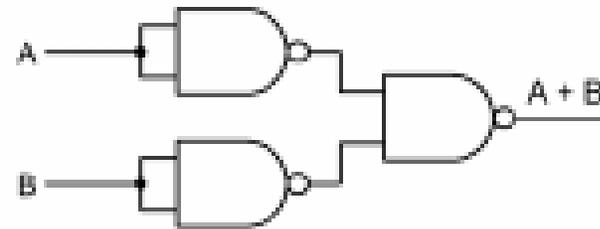
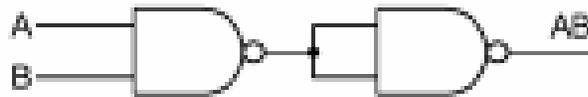


A	B	C	M
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

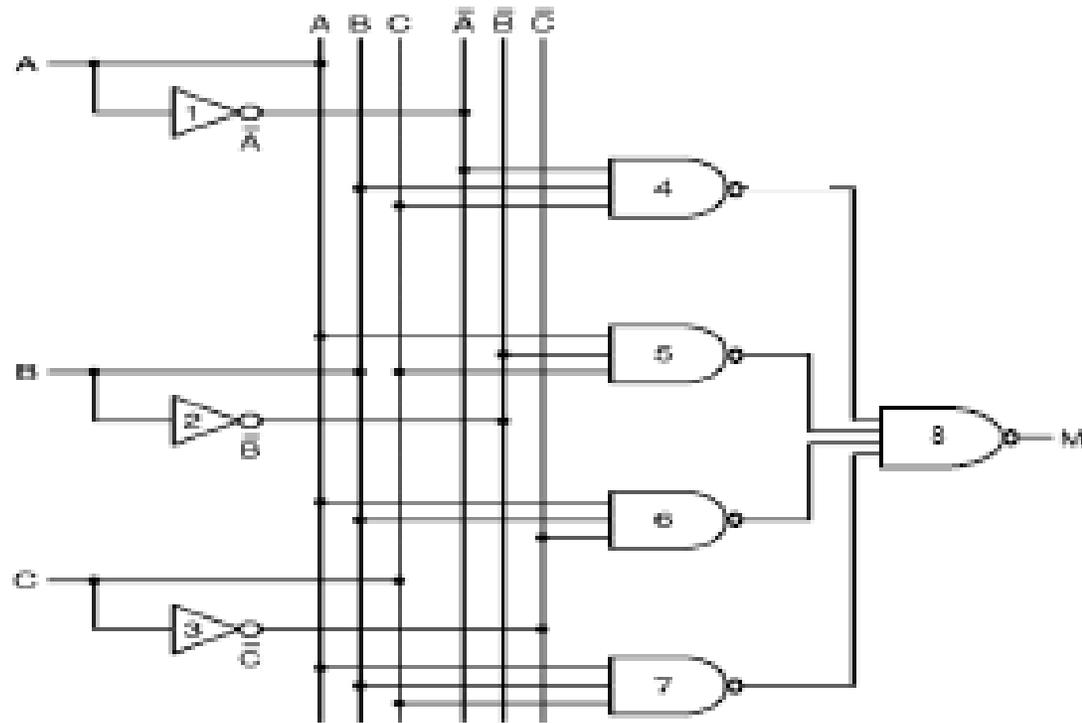
$$M = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

Completezza delle porte NAND e NOR

È possibile simulare AND, OR e NOT, e quindi realizzare qualsiasi circuito, usando soli NAND oppure soli NOR.

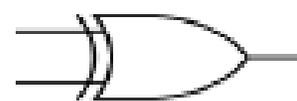


Realizzazione della maggioranza con solo NAND

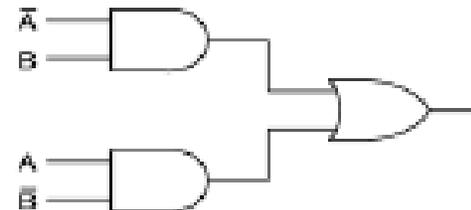
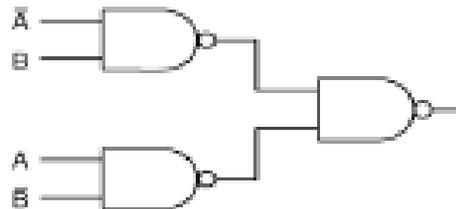


Porte XOR

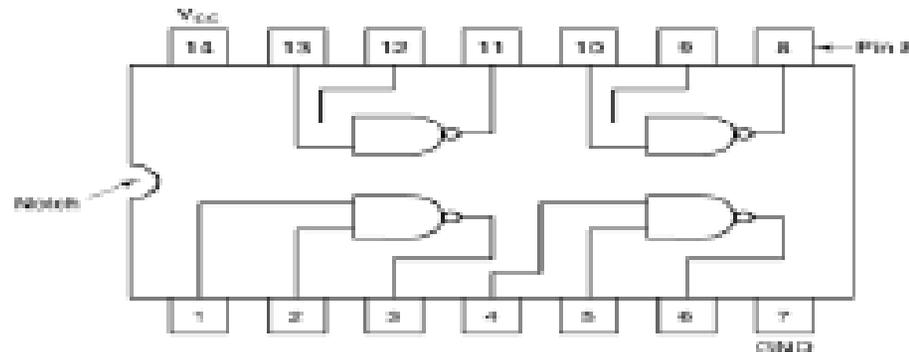
A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0



- Calcola la funzione *OR esclusiva*: dà uscita 1 (vero) quando uno solo degli ingressi (ma non entrambi) vale 1
- Facilmente realizzabile con porte AND, OR e NAND



Circuiti Integrati

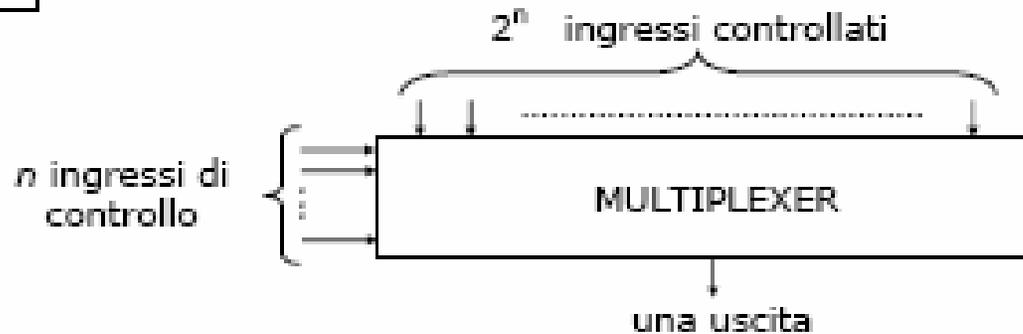


- Molte porte realizzate sulla stessa piastrina di silicio (*chip*)
- Contenitori da 14 a 68 piedini
- Vari livelli di integrazione:
 - SSI (*Small Scale*) 1-10 porte
 - MSI (*Medium Scale*) 10-100 porte
 - LSI (*Large Scale*) 10^2 - 10^5 porte
 - VLSI (*Very Large Sc.*) $> 10^5$ porte
- Tempi di commutazione 1-20 nsec

Circuiti combinatori

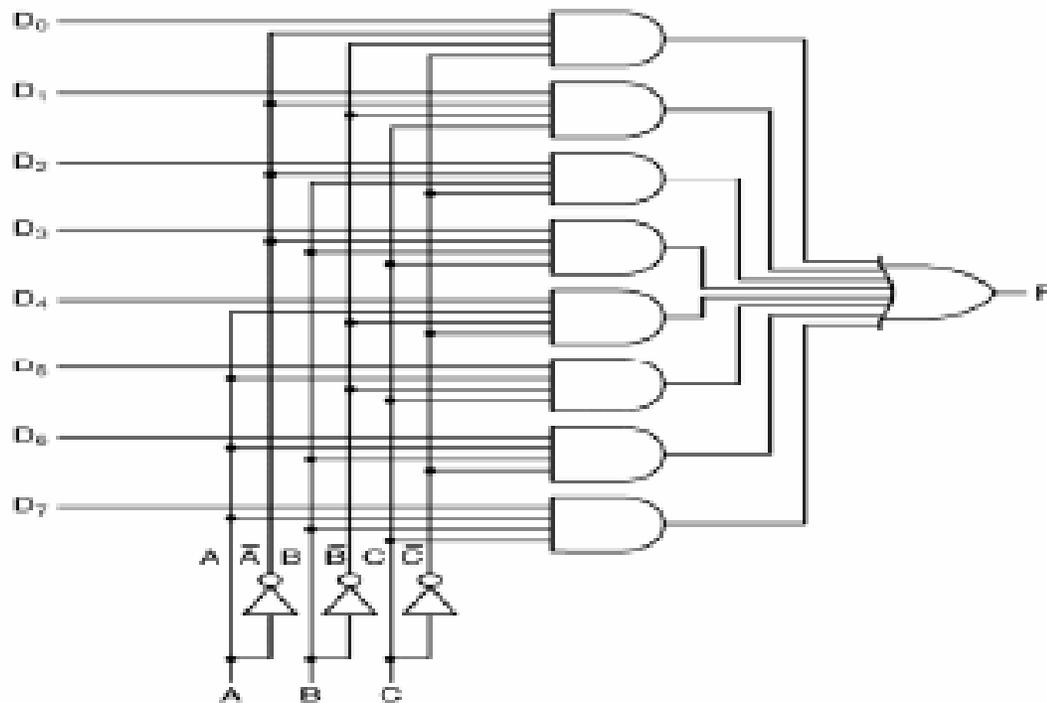
Circuiti in cui l'uscita dipende solo dagli ingressi, e non dallo stato cioè dalla storia passata

ES



- Gli ingressi di controllo selezionano quale degli ingressi controllati viene mandato in uscita

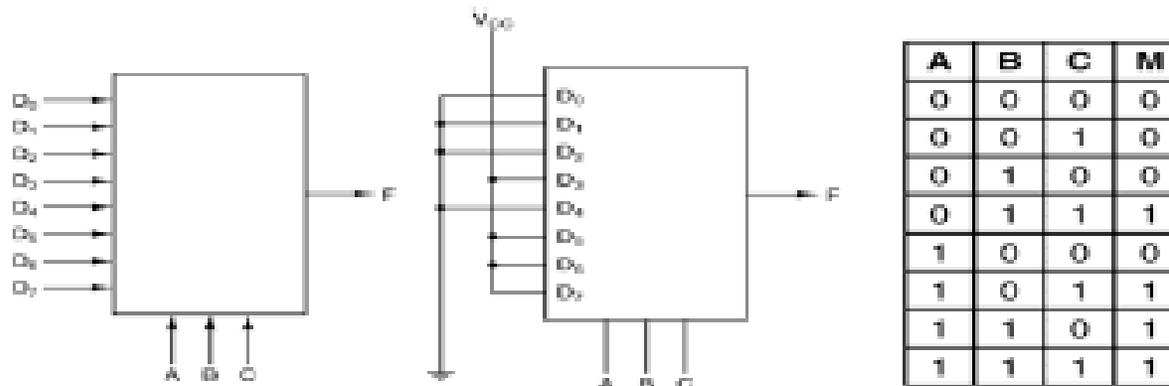
Multiplexer (circuito)



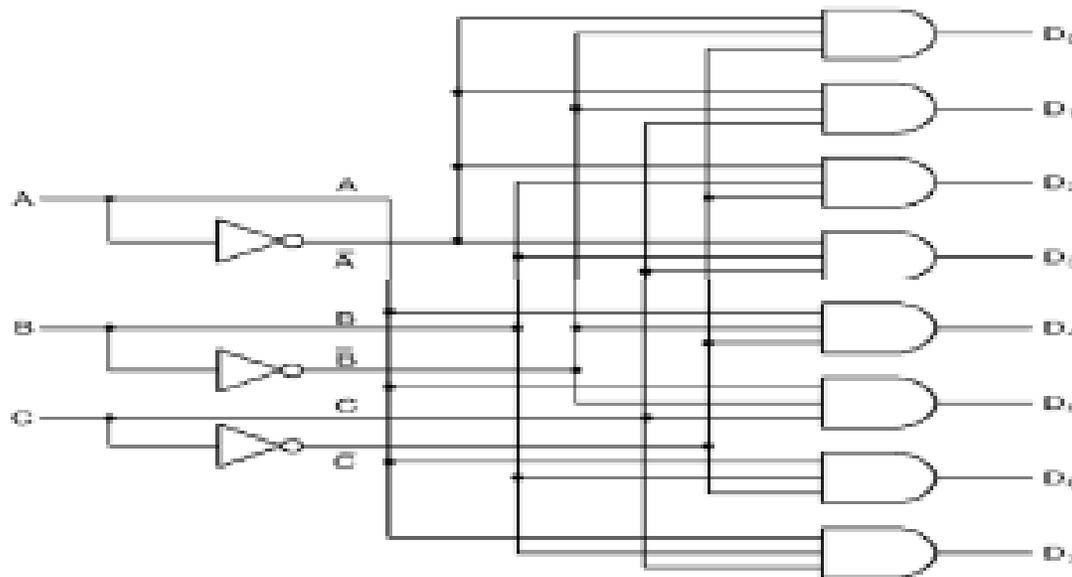
Realizzazione di funzioni booleane tramite multiplexer

- Con un multiplexer ad n bit si può calcolare qualsiasi funzione di n variabili
- Gli ingressi controllati corrispondono ai mintermini
- Si cablano a 0 o 1, a seconda che il mintermine compaia o meno nella forma canonica

ES (*Funzione di maggioranza*)

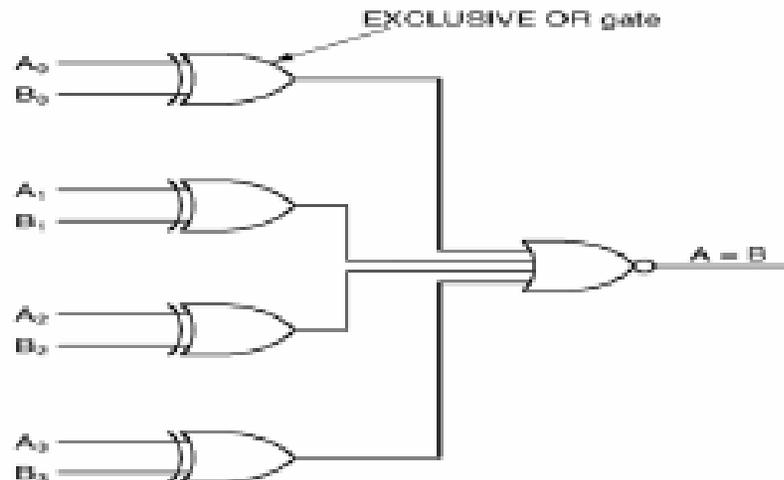


Decodificatore



- Circuito a n ingressi e 2^n uscite
- Una ed una sola delle n uscite assume valore vero in corrispondenza a ciascuna delle 2^n configurazioni di ingresso

Comparatore

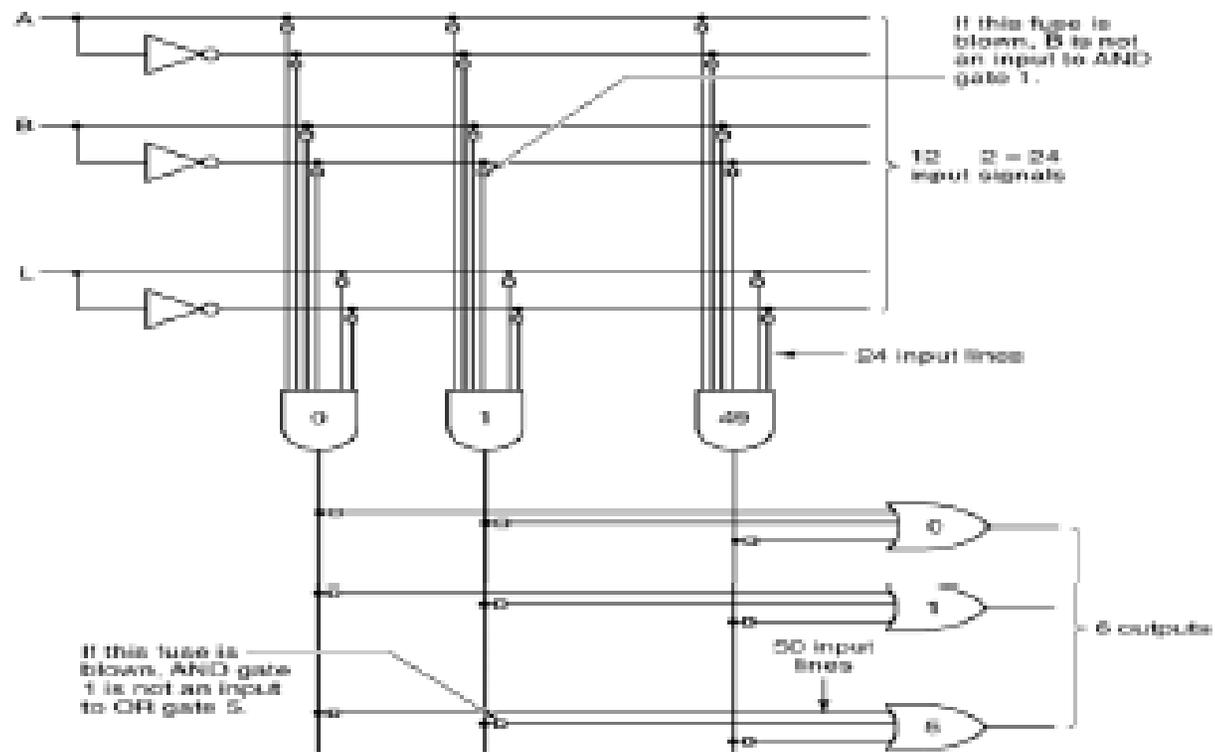


- Compara i bit omologhi di due stringhe
- L'uscita vale 1 se e solo se $A_i = B_i \forall i$
- Se $A_i = B_i$ allora $A_i \text{ XOR } B_i = 0$
- Il NOR da uscita 1 solo quando tutti i suoi ingressi valgono 0

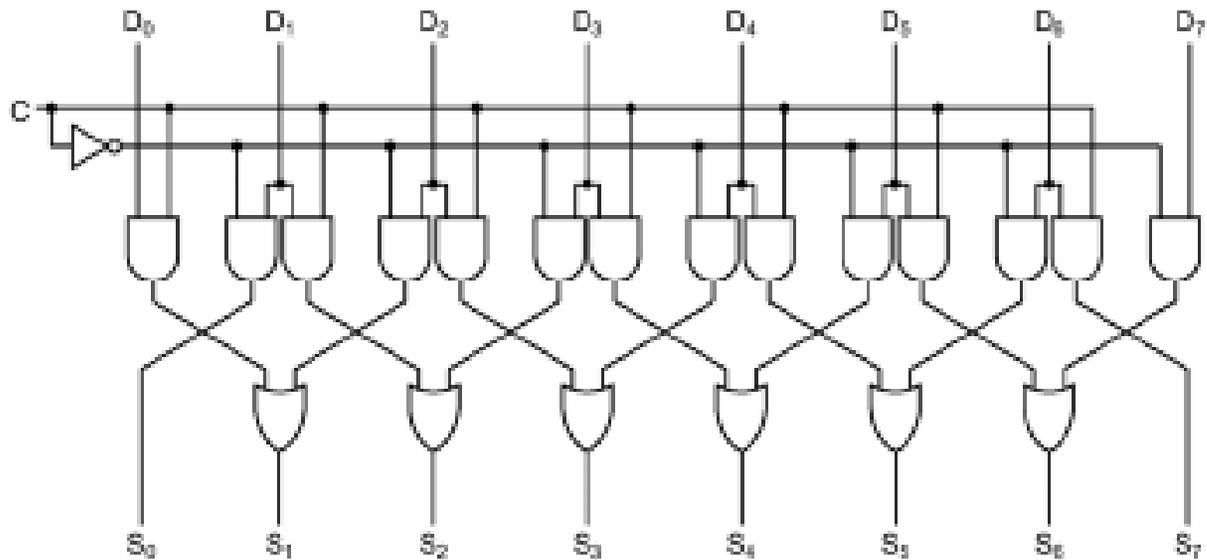
Programmable Logic Arrays (PLA)

- Permette di realizzare una "qualsiasi" funzione (in prima forma canonica)
- Circuito configurabile tramite bruciatura (interruzione) di connessioni
- Permette di calcolare più funzioni
- Due sezioni nel circuito:
 - a) generazione di un insieme di mintermini
 - b) selezione dei mintermini da inviare a ciascuna delle uscite
- Limitazioni
 - Numero limitato di mintermini generati $\ll 2^n$
 - Numero limitato di mintermini in ingresso a ciascuno degli OR di uscita

PLA: esempio di realizzazione circuitale



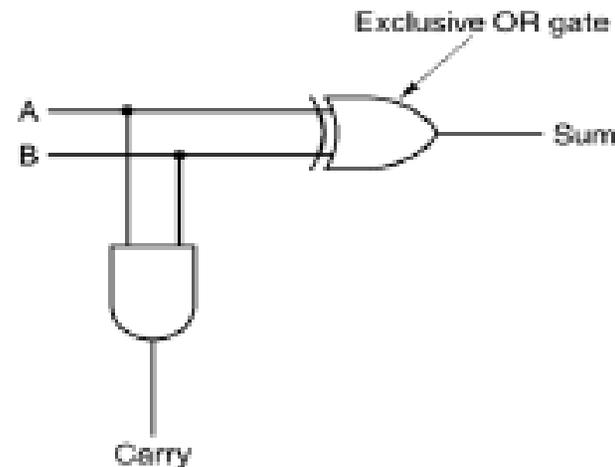
Shifter



- Il segnale C determina il verso dello shift (sinistra/destra)

Semiaddizionatore (Half adder)

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

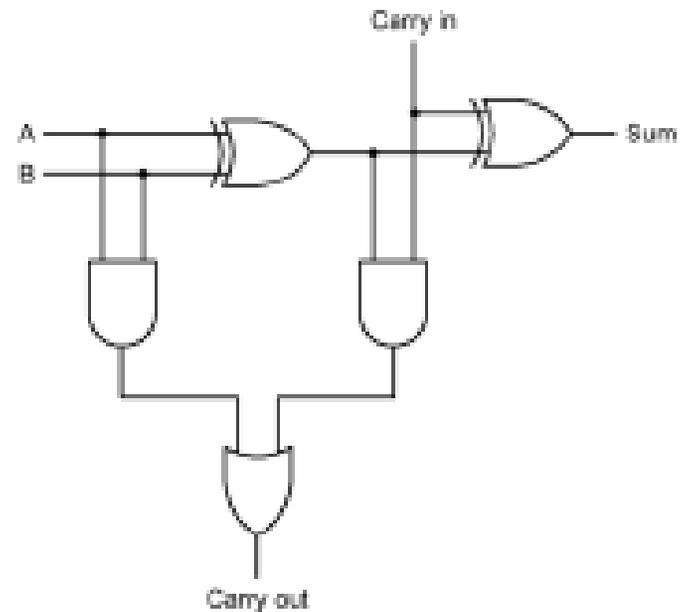


- Circuito a 2 ingressi e 2 uscite: somma e riporto (*carry*)
- Non può essere usato per la somma di numerali a più bit, dove occorre sommare anche il riporto della cifra precedente

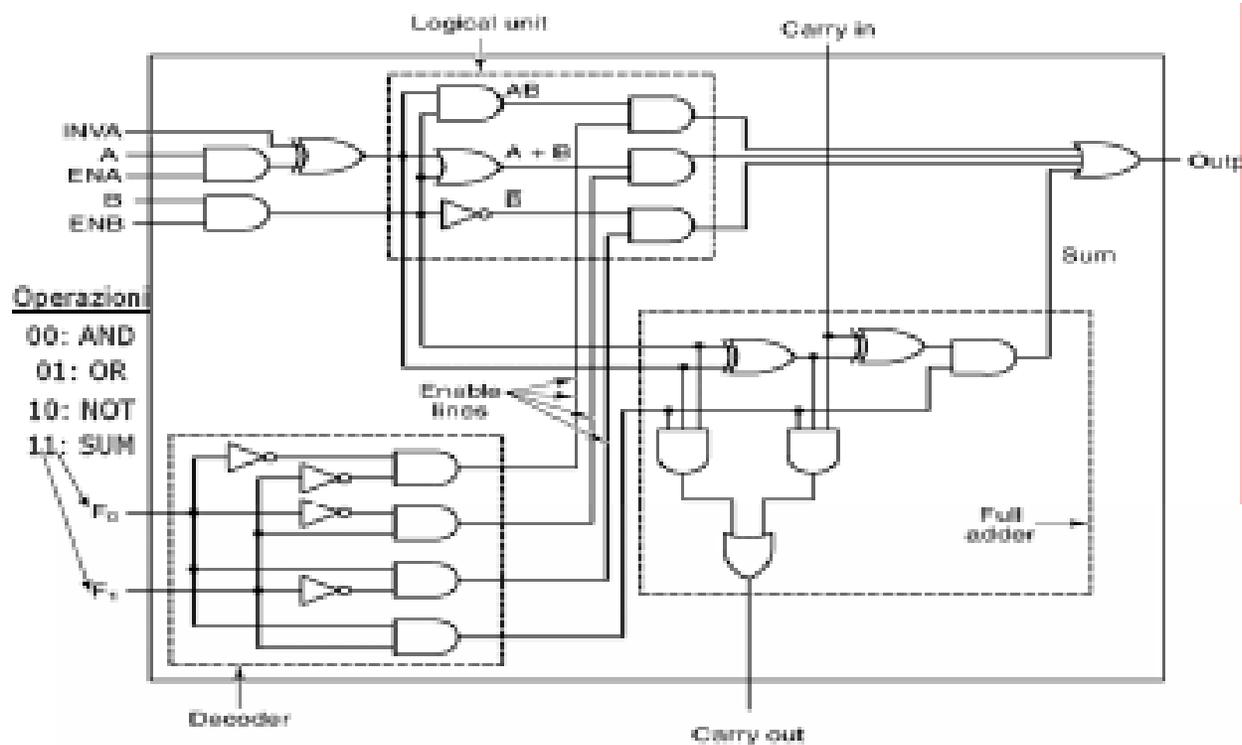
Addizionatore completo

- Circuito a 3 ingressi e 2 uscite
- Riceve il riporto dalla cifra precedente

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

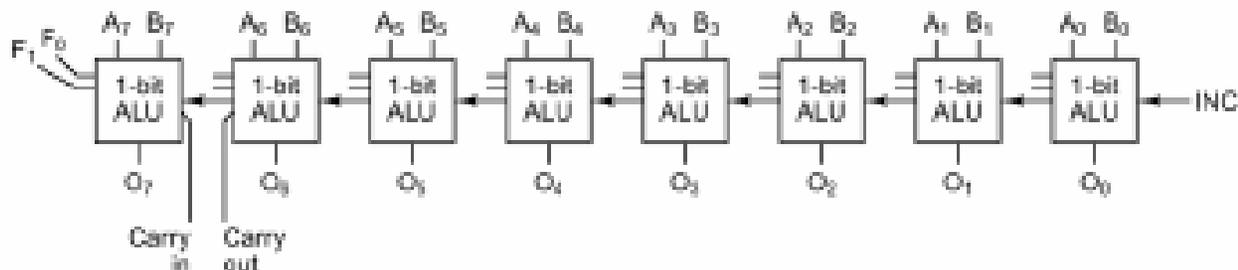


ALU a 1 bit (bit slice)



ALU a n bit

- Realizzata connettendo n ALU ad 1 bit (bit slices)
- Problema: propagazione dei riporti
- Ciascuno stadio deve attendere il riporto dal precedente
- Tempo di addizione lineare con n
- INC incrementa la somma di 1 ($A+1$, $A+B+1$)



Circuiti sequenziali



$$o_i = f_i(i_1, \dots, i_n, s_1, \dots, s_r) \quad i=1, \dots, m$$

$$s'_j = g_j(i_1, \dots, i_n, s_1, \dots, s_r) \quad j=1, \dots, r$$

- Le uscite del circuito dipendono dagli ingressi e dalla *storia passata*
- La storia passata è riassunta nello stato che è codificato nelle variabili di stato booleane s_1, \dots, s_r
- Le variabili di stato sono memorizzate in elementi di memoria binari
- Circuiti combinatori calcolano le uscite e il nuovo valore dello stato