

Architettura degli Elaboratori 2021-2022

Circuiti digitali
Logica combinatoria

Prof. Elisabetta Fersini
elisabetta.fersini@unimib.it

Cosa vedremo oggi

- Introduzione ai circuiti
- Porte logiche (gates)
- Circuiti notevoli
 - Decoder
 - Multiplexor
 - Programmable Logic Array (PLA)
 - Read Only Memory (ROM)
 - Array di elementi logici

- I circuiti logici sono realizzati come **circuiti integrati** realizzati su chip di silicio (piastrina)
- Porte (**gate**) e fili depositati su chip di silicio, inseriti in un package e collegati all'esterno con un certo insieme di pin (piedini)
- I circuiti integrati si distinguono per grado di integrazione
 - Da singole porte indipendenti, a circuiti più complessi

- Integrazione
 - SSI (Small Scale Integrated): 1-10 porte
 - MSI (Medium Scale Integrated): 10-100 porte
 - LSI (Large Scale Integrated): 100-100.000 porte
 - VLSI (Very Large Scale Integrated): > 100.000 porte
- Con tecnologia SSI, i circuiti integrati contenevano poche porte, direttamente collegate ai pin esterni
- Con tecnologia MSI, i circuiti integrati contenevano alcuni componenti base
 - circuiti comunemente usati nel progetto di un computer
- Con tecnologia VLSI, i circuiti integrati possono oggi contenere una CPU completa (o più)
 - microprocessore

- Nell'elettronica digitale sia gli ingressi che le uscite possono assumere solo i valori di **segnale alto** (1 per convenzione) o **basso** (0 per convenzione).
 - In un circuito digitale i valori binari sono ottenuti tramite discretizzazione dei segnali:
 - Falso: segnali con voltaggio basso ≤ 1 0 / FALSO / [0..1] Volt
 - Vero: segnali con voltaggio più alto > 1 1 / VERO / [2..5] Volt
- Un **circuito** (o rete) **combinatoria** è quel circuito il cui lo stato delle uscite dipende solo dalla funzione logica applicata allo stato istantaneo (cioè in un determinato istante di tempo) delle sue entrate.
- Un **circuito** (o rete) **sequenziale** è quel circuito il cui lo stato delle uscite non dipende solo dalla funzione logica applicata ai suoi ingressi, ma anche sulla base di valori pregressi collocati in memoria.

- Le **porte logiche** sono i componenti elettronici che permettono di svolgere le operazioni logiche **primitive** oltre che a quelle direttamente **derivate**.
 - Le porte logiche che realizzano le operazioni principali dell'algebra booleana
 - Una porta logica è un circuito elettronico che dati dei segnali 0 e 1 in input produce un segnale in output ottenuto effettuando una operazione booleana sugli ingressi
- Le porte logiche hanno n input e generalmente 1 output
 - A ogni combinazione di valori in ingresso corrisponde una e solo una combinazione in di valori uscita
 - Dati gli input, l'output corrispondente appare quasi istantaneamente
- Porte logiche **fondamentali**: AND, OR, NOT
- Porte logiche **derivate**: NAND, NOR, XOR

Porta logica AND

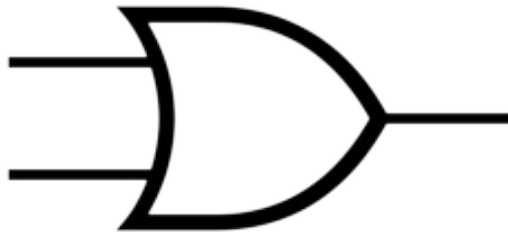
- La porta logica **AND** svolge l'operazione logica di AND tra due bit, detta anche **prodotto logico**.
- Considerando due entrate A e B, l'uscita $A \cdot B$ è data da:



A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

Porta logica OR

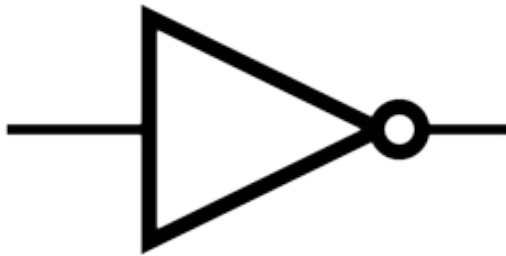
- La porta logica **OR** svolge l'operazione logica di OR tra due bit, detta anche **somma logica**.
- Considerando due entrate A e B, l'uscita $A + B$ è data da:



A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

Porta logica NOT

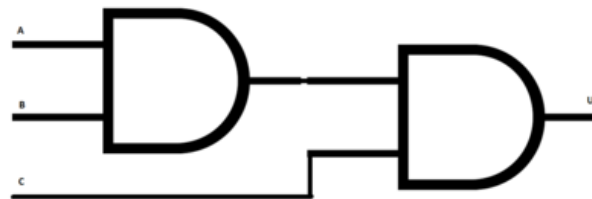
- La porta logica **NOT** svolge l'operazione logica di NOT su un bit, detta anche **negazione logica**.
- Considerando un'entrata A, l'uscita \bar{A} è data da:



A	\bar{A}
0	1
1	0

Porte con più di due ingressi

- Ad eccezione della porta NOT, le altre porte logiche possono esistere anche ad N ingressi (2, 3, 4, ..., N).
- Queste porte svolgono l'operazione logica associata su N bit invece che su 2.
 - Sono particolarmente comode nella rappresentazione grafica dei circuiti logici.
- Nella pratica, cioè nella realizzazione di circuiti, se si hanno a disposizione solo porte a 2 ingressi, è possibile realizzare porte a N ingressi collegando a **cascata** tra loro porte a 2 ingressi.
 - Un AND a 3 ingressi si può creare usando 2 AND a 2 ingressi come segue:

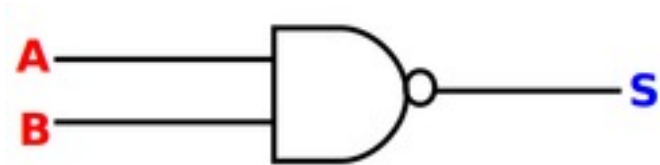


Porte logiche derivate

- Oltre alle porte logiche fondamentali (AND, OR, NOT) esistono altre porte che sono realizzate combinando le porte fondamentali, il cui principale scopo è la semplificazione dei circuiti (realizzando operazioni composte in un unico componente).
- Le porte logiche derivate sono NAND, NOR e XOR.

Porta NAND

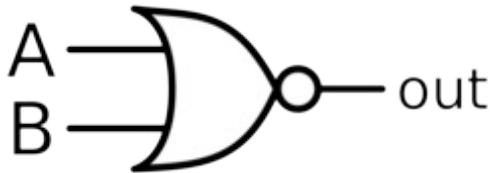
- La porta logica **NAND** svolge l'operazione logica di NOT sul bit risultante dall'operazione AND sui bit in ingresso.



A	B	$A \cdot B$	A NAND B
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Porta NOR

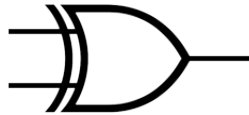
- La porta logica **NOR** svolge l'operazione logica di NOT sul bit risultante dall'operazione OR sui bit in ingresso.



A	B	$A + B$	A NOR B
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Porta XOR

- La porta logica **XOR** opera come disgiunzione esclusiva tra due input.



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

- Qualunque funzione logica può essere costruita usando le porte logiche AND, OR e NOT.
- Le porte NOR e NAND che svolgono la funzione di inverter, sono definite **universali**.
 - NAND → OR

$$A \text{ NAND } B = \text{NOT}(A \text{ AND } B) = (\text{NOT } A) \text{ OR } (\text{NOT } B)$$

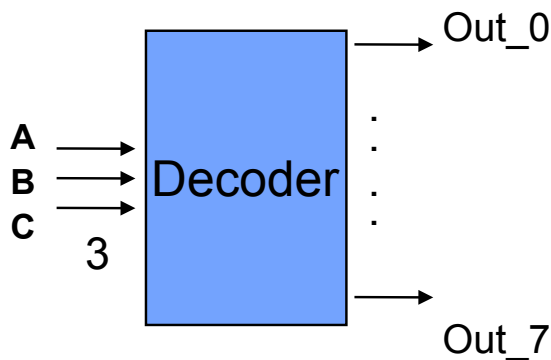
- NAND → NOT

$$A \text{ NAND } 1 = \text{NOT}(A \text{ AND } 1) = (\text{NOT } A)$$

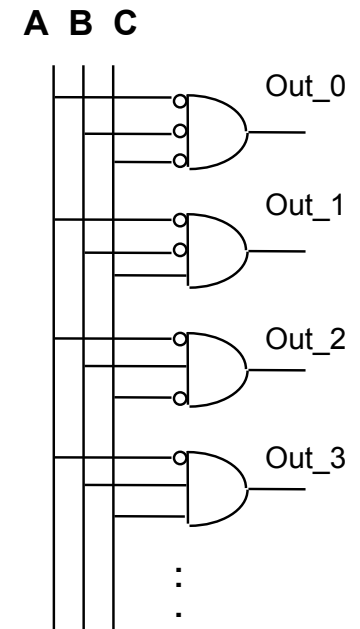
- Un decoder è un componente elettronico caratterizzato dall'avere n ingressi e 2^n uscite.
- Lo scopo del decoder è di impostare allo stato alto l'uscita corrispondente alla conversione in base 10 della codifica binaria a n bit ricevuta in input (e di impostare allo stato basso tutte le altre).
- In pratica:
 - gli n input sono interpretati come un numero unsigned
 - se questo numero rappresenta il numero i , allora
 - solo il bit in output di indice i ($i=0,1,\dots,2^n-1$) verrà posto ad 1
 - Tutti gli altri verranno posti a 0

Decoder

- 2^n uscite, solo 1 valore è attivo per ogni combinazione di input
- Quindi:
 - l'ingresso seleziona una delle uscite;
 - l'uscita selezionata ha valore 1 tutte le altre 0



A	B	C	0	1	2	3	4	5	6	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



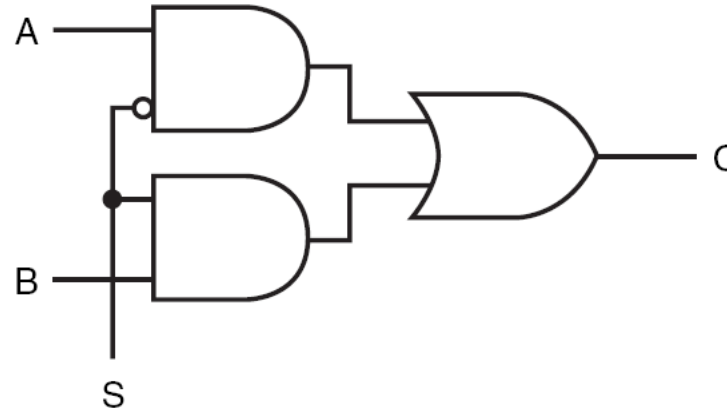
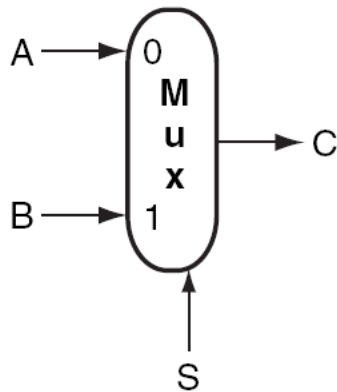
Multiplexor

- Un **multiplexor**, detto anche **selettore**, è un componente elettronico caratterizzato da:
 - 2^n entrate principali
 - n entrate di controllo (selettore)
 - 1 uscita
- Il valore del selettore determina quale input diviene output

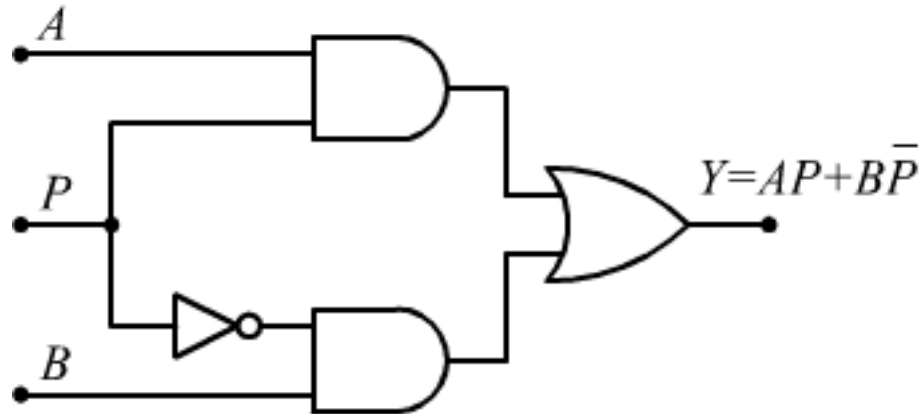
Multiplexor

- Esempio:

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



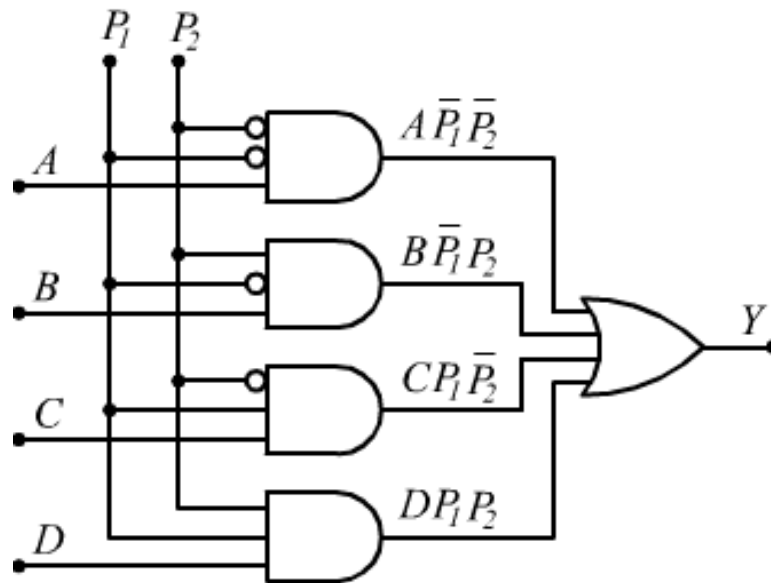
Multiplexor



P	Y
0	B
1	A

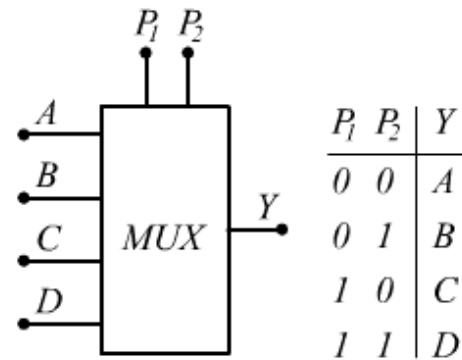
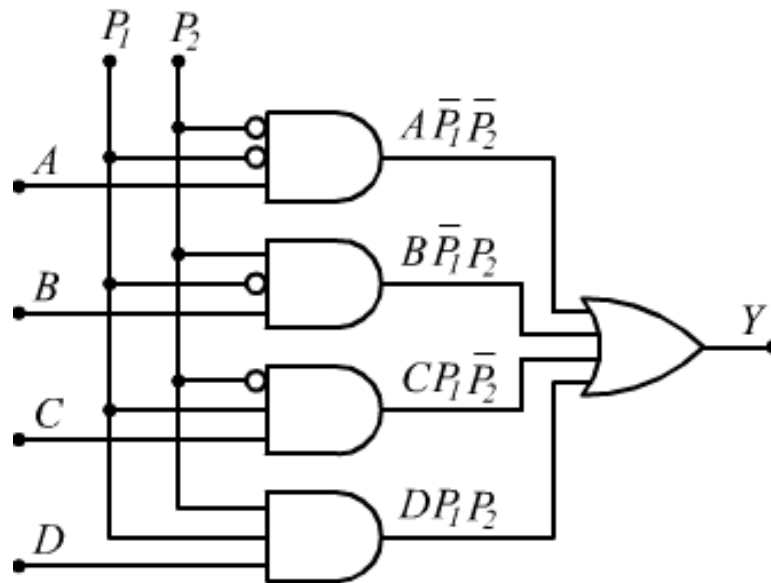
Multiplexor

Quale funzione logica viene implementata?



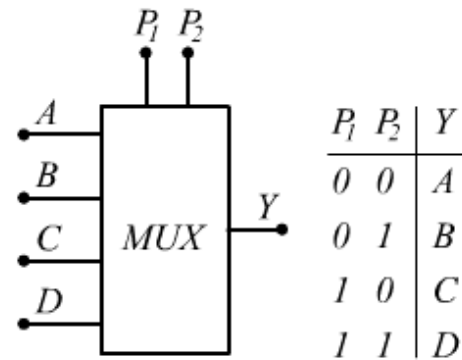
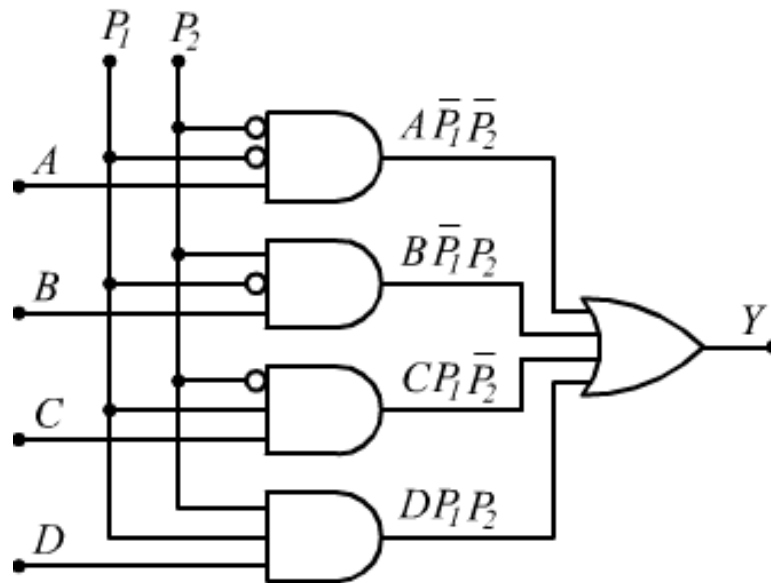
Multiplexor

Quale funzione logica viene implementata?



Multiplexor

Quale funzione logica viene implementata?



$$Y = A\bar{P}_1\bar{P}_2 + B\bar{P}_1P_2 + CP_1\bar{P}_2 + DP_1P_2$$

Multiplexor

- Se un multiplexor riceve in input n segnali, esso necessiterà di $\log_2 n$ selettori, e consisterà di:
 1. Un decoder che genererà n segnali
 2. Un array di n porte logiche AND
 3. Un'unica porta logica OR

$$n = 2^m \longleftrightarrow m = \lg_2 n$$

- Come abbiamo visto, utilizzando le porte logiche AND, OR e NOT è possibile implementare qualunque funzione logica più complessa.
- Possiamo creare logiche a due livelli:
 - Somma di prodotti: somma logica (OR) di prodotti (AND)
 - Prodotto di somme: prodotto (AND) di somme (OR)

$$E = ((A \cdot B) + (A \cdot C) + (B \cdot C)) \cdot \overline{(A \cdot B \cdot C)}$$

Somma di prodotti

$$E = \overline{(\bar{A} + \bar{B} + C) \cdot (\bar{A} + \bar{C} + B) \cdot (\bar{B} + C + A)}$$

Prodotto di somme

Logiche a due livelli e PLA

- Consideriamo la seguente tabella di verità per D, dati gli ingressi A,B,C ed esprimiamola come somma di prodotti.

Inputs			Output
A	B	C	D
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$$\bar{A} \cdot \bar{B} \cdot C$$

$$\bar{A} \cdot B \cdot \bar{C}$$

$$A \cdot \bar{B} \cdot \bar{C}$$

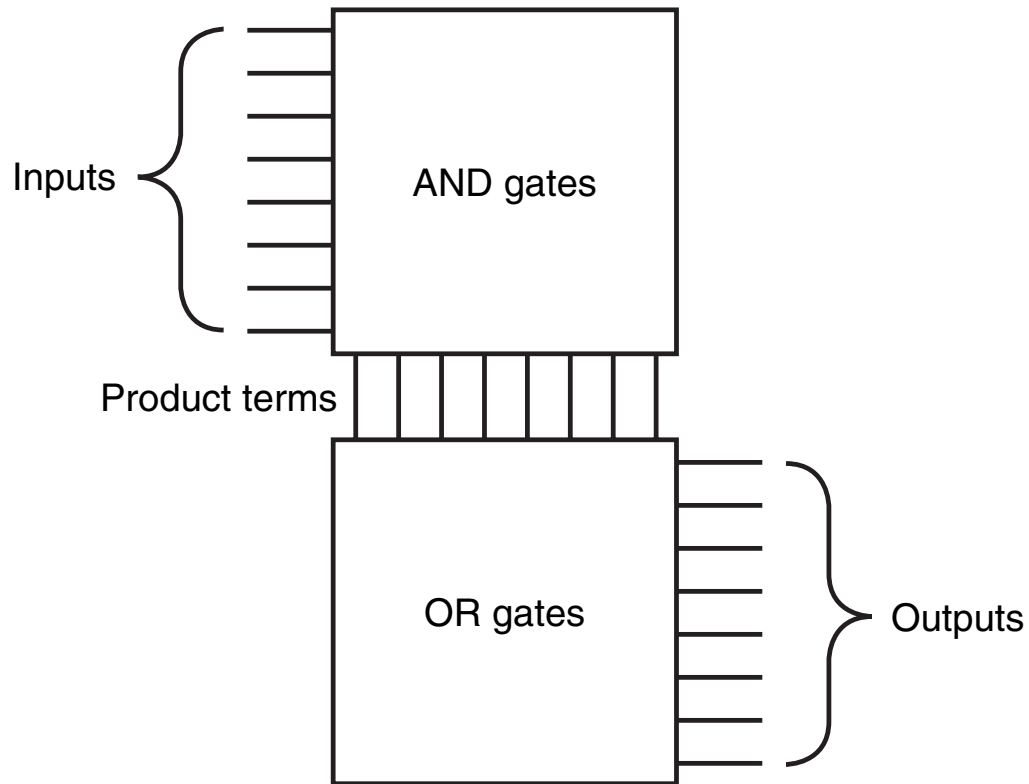
$$A \cdot B \cdot C$$

- Possiamo scrivere la funzione logica per D come somma di prodotti:

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

Logiche a due livelli e PLA

- La somma di prodotti corrisponde ad una implementazione comunemente nota come **Programmable Logic Array (PLA)**:
 - Un insieme di input
 - I corrispondenti input complementati (mediante inverter)
 - Una logica a due stage:
 - Primo stage: un array di porte logiche AND (prodotto)
 - Secondo stage: un array di porte logiche OR (somma)

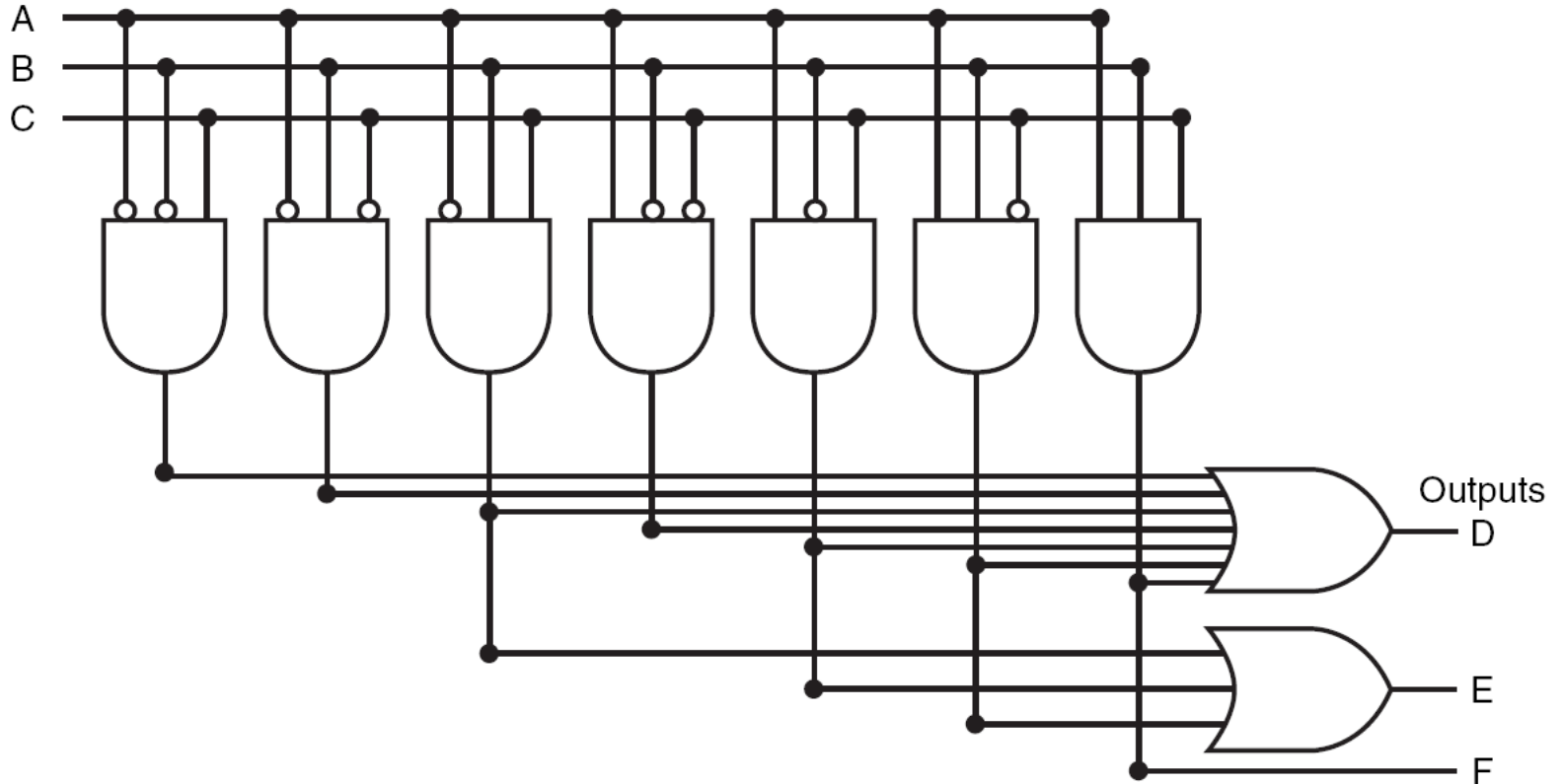


- Consideriamo la seguente tabella di verità.

Inputs			Outputs		
A	B	C	D	E	F
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	1	0	1

- Costruire il PLA corrispondente.

Inputs



ROM (Read Only Memory)

- Circuito combinatorio in cui ad ogni ingresso (indirizzo) corrisponde una uscita (contenuto della cella di memoria con quell'indirizzo)
- Osservazione: si chiama "memoria" perché ha delle locazioni di memoria che possono essere lette;
- Input: n bit
- Output: una tra le 2^n celle/locazioni di memoria (the memory height)
- Una locazione di memoria è su m bit (the memory width)

X_0	X_1	X_2	Z_0	Z_1
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

ROM vs PLA

1. ROM – fully decoded; PLA – partially decoded
2. ROM dimensione più grande rispetto a PLA
3. PLA più efficienti
4. ROM possono implementare qualsiasi funzione logica con il numero di input e output specificato senza dover modificare la sua dimensione

- PROM (Programmable ROM) – possono essere programmate se il designer conosce il loro contenuto
- EPROM (Erasable Programmable ROM)