



# Architettura degli Elaboratori I

Corso di Laurea Triennale in Informatica

Università degli Studi di Milano

Dipartimento di Informatica “Giovanni Degli Antoni”

Edizione 2 (Cognomi H-Z), A.A. 2024-2025, Nicola.Basilico@unimi.it

# Programmable Logic Arrays (PLA)

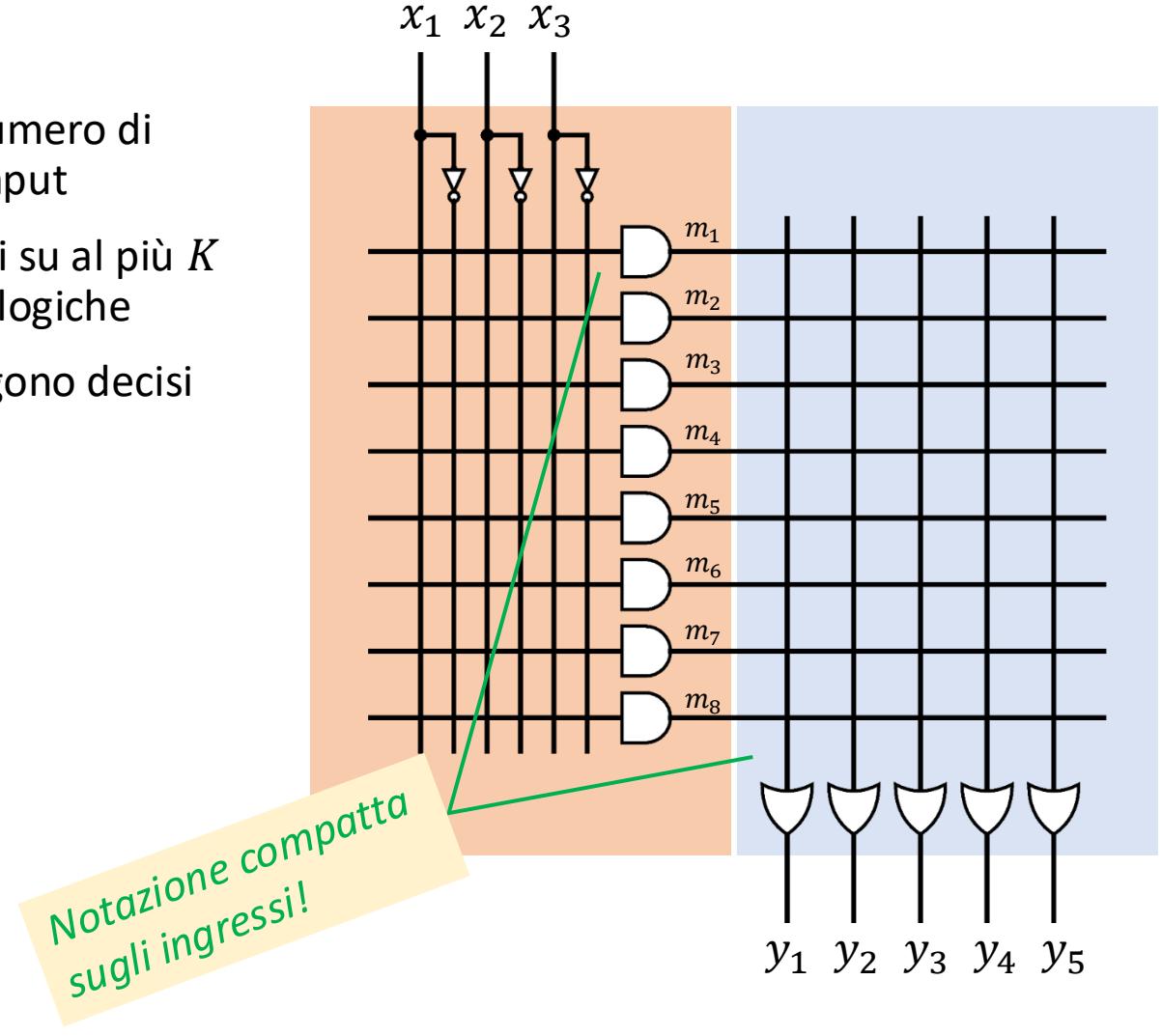
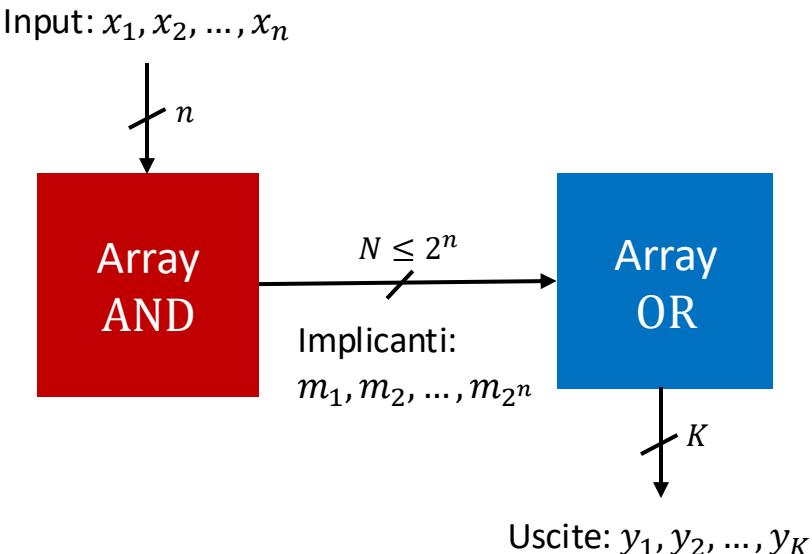
- I PLA sono un dispositivo logico programmabile che viene utilizzato per l'implementazione hardware dei circuiti combinatori
- È un circuito «grezzo» su cui, tramite un procedimento automatizzato, possiamo «forgiare» la nostra funzione logica



- Programmando una PLA, possiamo ottenere un circuito SOP che rappresenta la nostra funzione
- La PLA ricalca i principi di sintesi dei circuiti combinatori in **prima forma canonica** che sono basati su due stadi: stadio AND e stadio OR

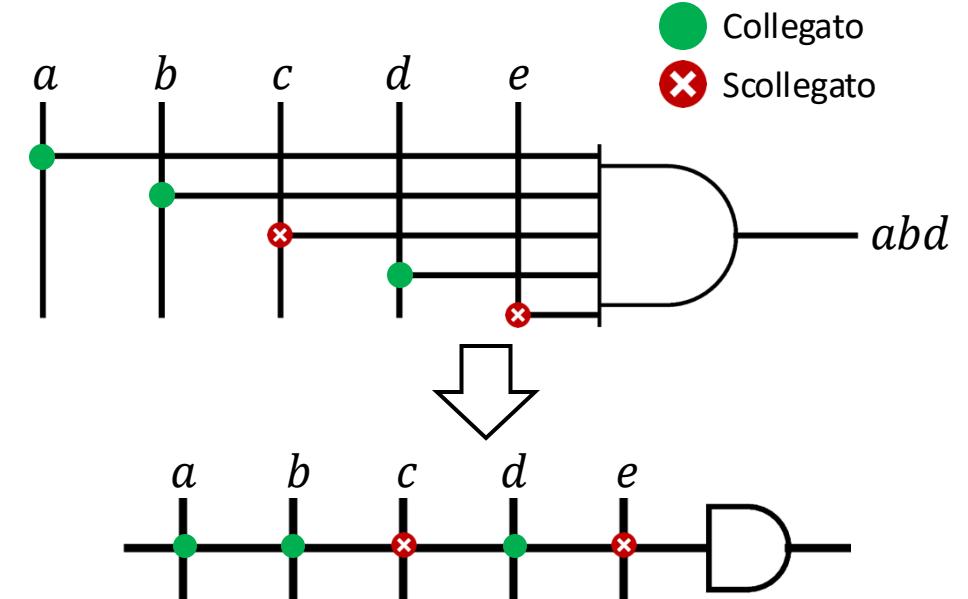
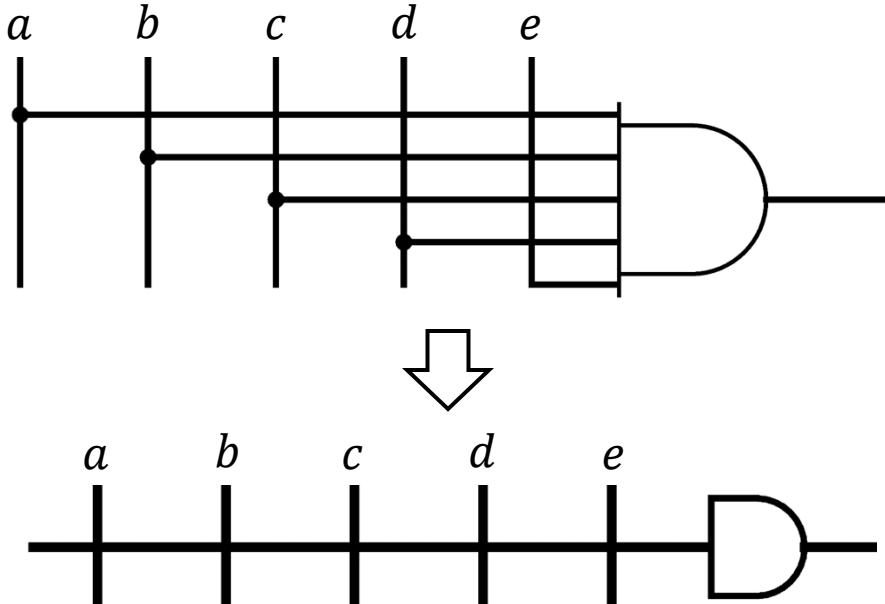
# Programmable Logic Arrays (PLA)

- Il layout a due stadi si basa su due array: AND e OR
- L'array AND viene programmato per calcolare un numero di implicanti  $N$  che è al più  $2^n$  dove  $n$  è il numero di input
- L'array OR consente di sommare gruppi di implicanti su al più  $K$  uscite, quindi possiamo codificare al più  $K$  funzioni logiche
- $n, N$  e  $K$  sono parametri dimensionali del PLA, vengono decisi durante la fabbricazione del dispositivo



# Programmable Logic Arrays (PLA)

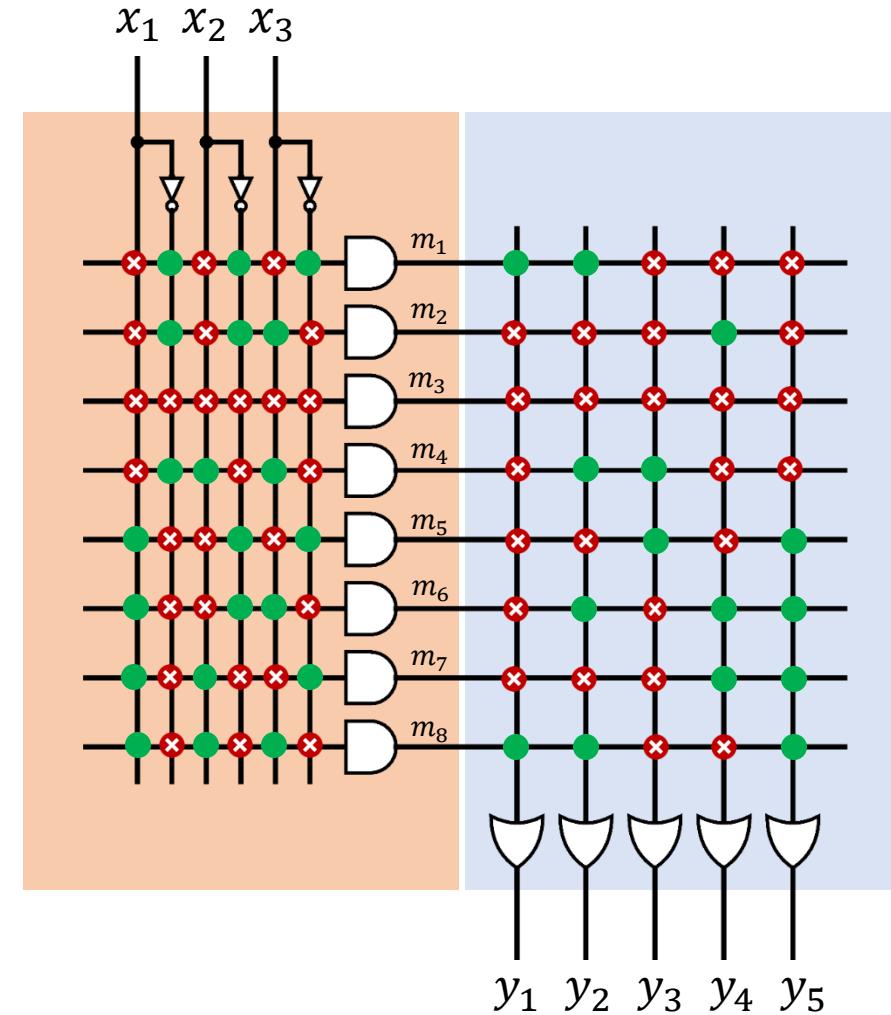
- Schema per la notazione compatta sugli ingressi
- La adottiamo sia per lo stadio AND che per lo stadio OR (l'esempio è per AND)
- Gli input  $a, b, c, \dots$  possono essere variabili naturali o negate
- Come si programma un PLA?
- «Bruciando» alcuni collegamenti e lasciandone altri collegati (viene di solito specificato tramite una matrice)
- Tecnologia possibile: un fusibile (o antifusibile) su ogni collegamento, applicando una tensione abbastanza alta «brucio» il fusibile e quindi blocca il collegamento



# Programmable Logic Arrays (PLA)

- **Esempio** di implementazione

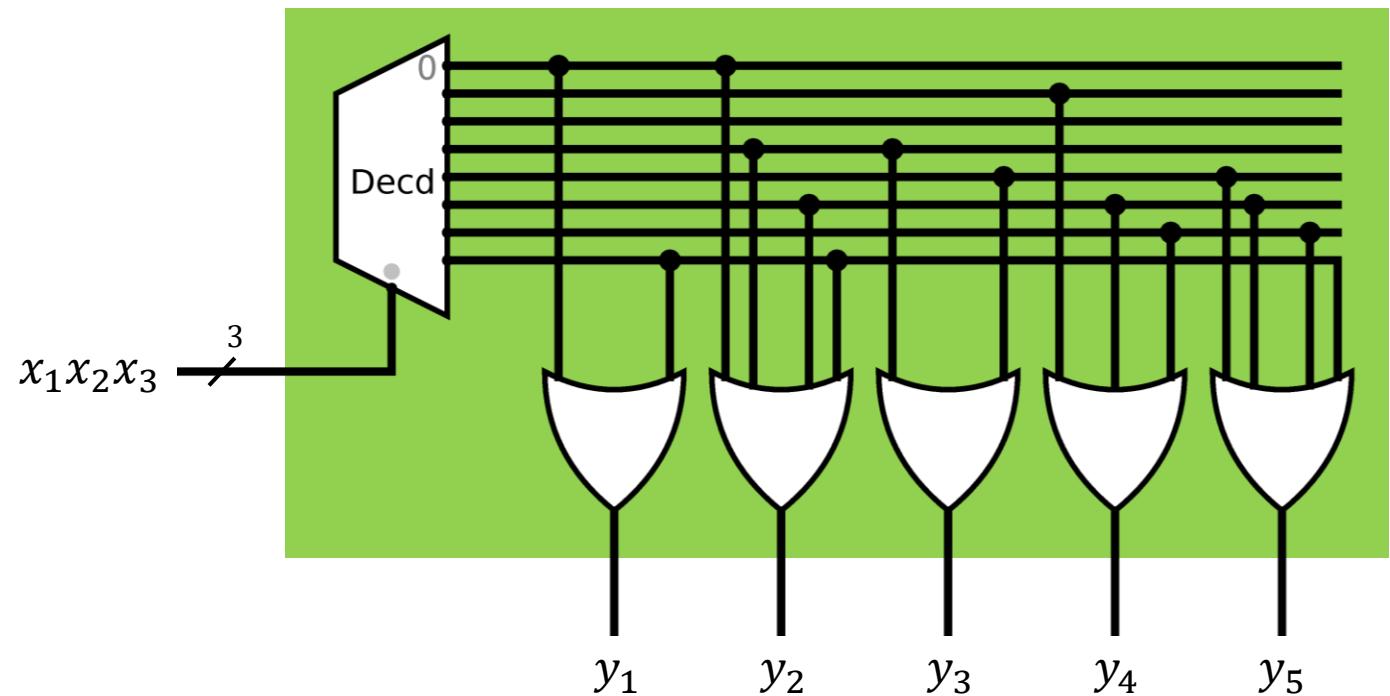
$x_1$	$x_2$	$x_3$	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$
0	0	0	1	1	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	0	0	0
0	1	1	0	1	1	0	0
1	0	0	0	0	1	0	1
1	0	1	0	1	0	1	1
1	1	0	0	0	0	1	1
1	1	1	1	1	0	0	1



# ROM

- Approccio alternativo: Read-Only Memory (ROM)
- Come PLA, ma l'array AND è **hardwired** in modo da poter generare tutti i  $2^n$  mintermini possibili
- L'input è visto come **l'indirizzo** di una cella di memoria a cui sono memorizzati i  $K$  bit delle uscite
- L'array OR è programmabile, nel caso delle EEPROM (Flash drives) è riscrivibile

$x_1$	$x_2$	$x_3$	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$
0	0	0	1	1	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	0	0	0
0	1	1	0	1	1	0	0
1	0	0	0	0	1	0	1
1	0	1	0	1	0	1	1
1	1	0	0	0	0	1	1
1	1	1	1	1	0	0	1



# PLA e ROM

## PLA

- Possono essere opportunamente dimensionati e il numero di implicanti può essere limitato
- Se però si hanno  $N \leq 2^n$  implicanti non si può rappresentare una qualsiasi funzione logica
- Potrebbe essere necessaria una semplificazione

## ROM

- Se si hanno  $n$  input si hanno sempre  $2^n$  celle di memoria
- Si ha a disposizione ogni mintermine quindi si può implementare una qualsiasi funzione logica
- Lo svantaggio delle ROM è la crescita esponenziale della loro dimensione con  $n$