

Sistemi logici complessi

I circuiti logici standard prevedono, per la realizzazione di un sistema complesso, i seguenti passi:

- definizione delle specifiche
- descrizione del sistema mediante interconnessione dei componenti
- assemblaggio su un circuito stampato

I componenti sono circuiti integrati ad esempio della serie 74 o della serie 54 (specifiche militari), realizzati in tecnologia TTL, TTL Schottky, TTL Schottky Low Power, CMOS etc. contenenti porte logiche.

Al crescere della complessità del sistema l'approccio precedente diventa poco pratico per l'aumento di costo, di dissipazione di potenza, di ritardo di propagazione e di ingombro ed inoltre per la relativa semplicità con cui un circuito può essere clonato

Dispositivi Logici Programmabili

Sono circuiti ad elevata scala di integrazione che possono essere opportunamente programmato (anche dall'utente finale) per realizzare una specifica funzione logica senza aggiungere o rimuovere componenti.

La programmazione avviene mediante descrizione ad alto livello in opportuni ambienti di sviluppo (**Hardware Description Language**)

Alta flessibilità ma alta volatilità

La programmazione può essere:

- **Reversibile**
- **Non Reversibile**
- **Volatile** finché c'è tensione
- **Non Volatile**

↳ Avviene a linguaggio ad alto livello, ma agisce sul chip.

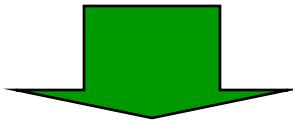
⇒ Non si ricorre più al circuito sulla board ma

è tutto dentro un chip.

Definizioni

↪ Programmabili sub campo dall'utente finale

PLD (Programmable Logic Device) o FPD (Field-Programmable Device)



SPLD (Simple Programmable Logic Device)



PLA (Programmable Logic Array)

[piani AND e OR programmabili]

CPLD (Complex Programmable Logic Device)



Sono costituiti da un insieme di SPLD sullo stesso chip

Basato su concetto di funzioni. SOMME DI PRODOTTI

PAL (Programmable Array Logic)

[piano AND programmabile e OR fisso]

FPGA (Field-Programmable Gate Array)



CPLD con meno ingressi e un maggior numero di Flip-Flop

Definizioni

Programmable Switch (Interruttore Programmabile)



consente di connettere tra loro elementi logici o fili mediante comando esterno

Logic Capacity (Capacità logica)



ammontare della logica esprimibile dal PLD in termini di numero equivalente di porte NAND a due ingressi

Numero di porte NAND equivalenti

Porte equivalenti

Logic Density (Densità logica)



Quanto è piccolo il chip rispetto a quanta potenza ha.

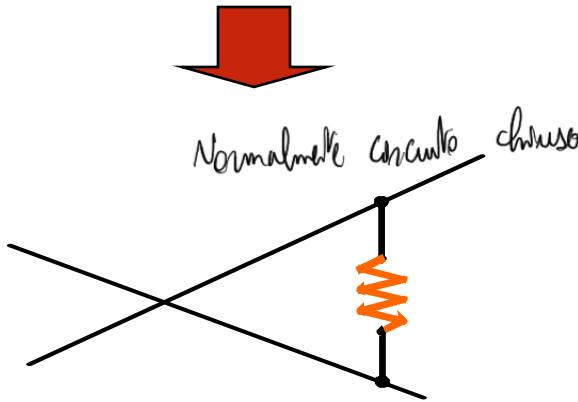
ammontare della logica per unità di area del PLD

*Logic Capacity
Area*

Interruttori Programmabili garantiscono la riprogrammabilità.

Non reversibili-Non Volatili

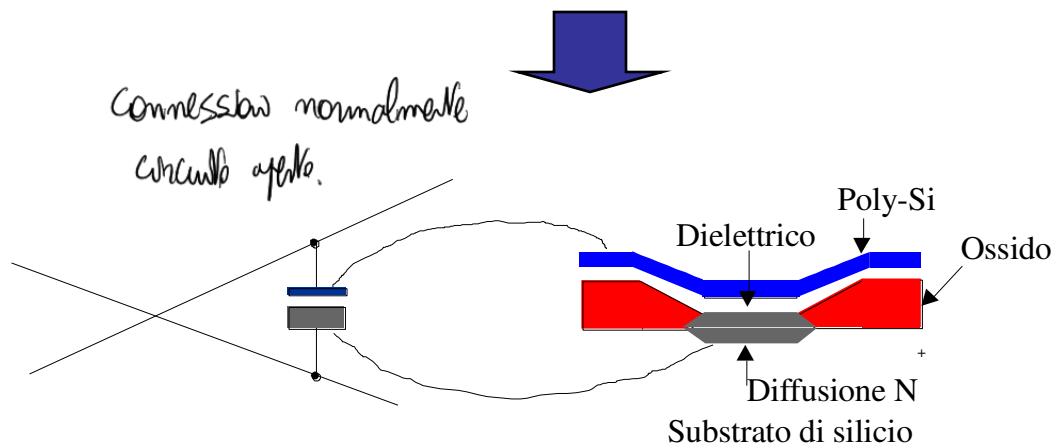
Fuse (Fusibili)



Aprono il circuito se attraversati da corrente elevata tale da bruciare il resistore

Non reversibile e non volatile

Antifuse (Antifusibili)

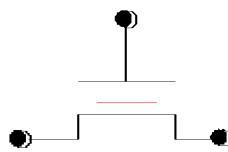


Chiudono il circuito se viene applicata una tensione elevata tale da perforare il dielettrico

Non reversibile e non volatile anche l'au.

Interruttori Programmabili

Reversibili-Non Volatili



Mos A DOPPIA GATE



Gá struttura

Punti che posso connettere e disconnettere

Transistor per EPROM



Programmabili fuori dal
circuito

Transistor per EEPROM



Programmabili nel
circuito

Interruttori Programmabili

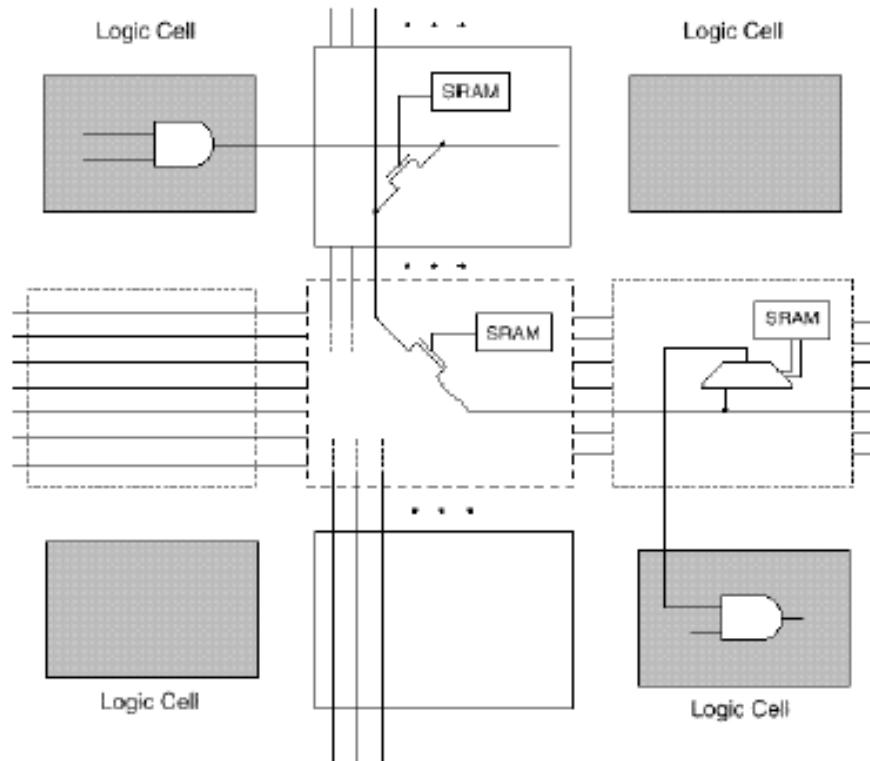
Reversibili-Volatili

Sono basati sull'utilizzo di memorie statiche SRAM per pilotare la gate di transistor (**pass-transistor**) usati per connettere due fili

uso pass transistors pilotati con
memorie dinamiche.

Alza segnale da gate quando
deve essere chiuso, altrimenti
lo chiuderà.

Volatile perché programmaz.
legguta a memorie volatile.



Interruttori Programmabili

Riepilogo

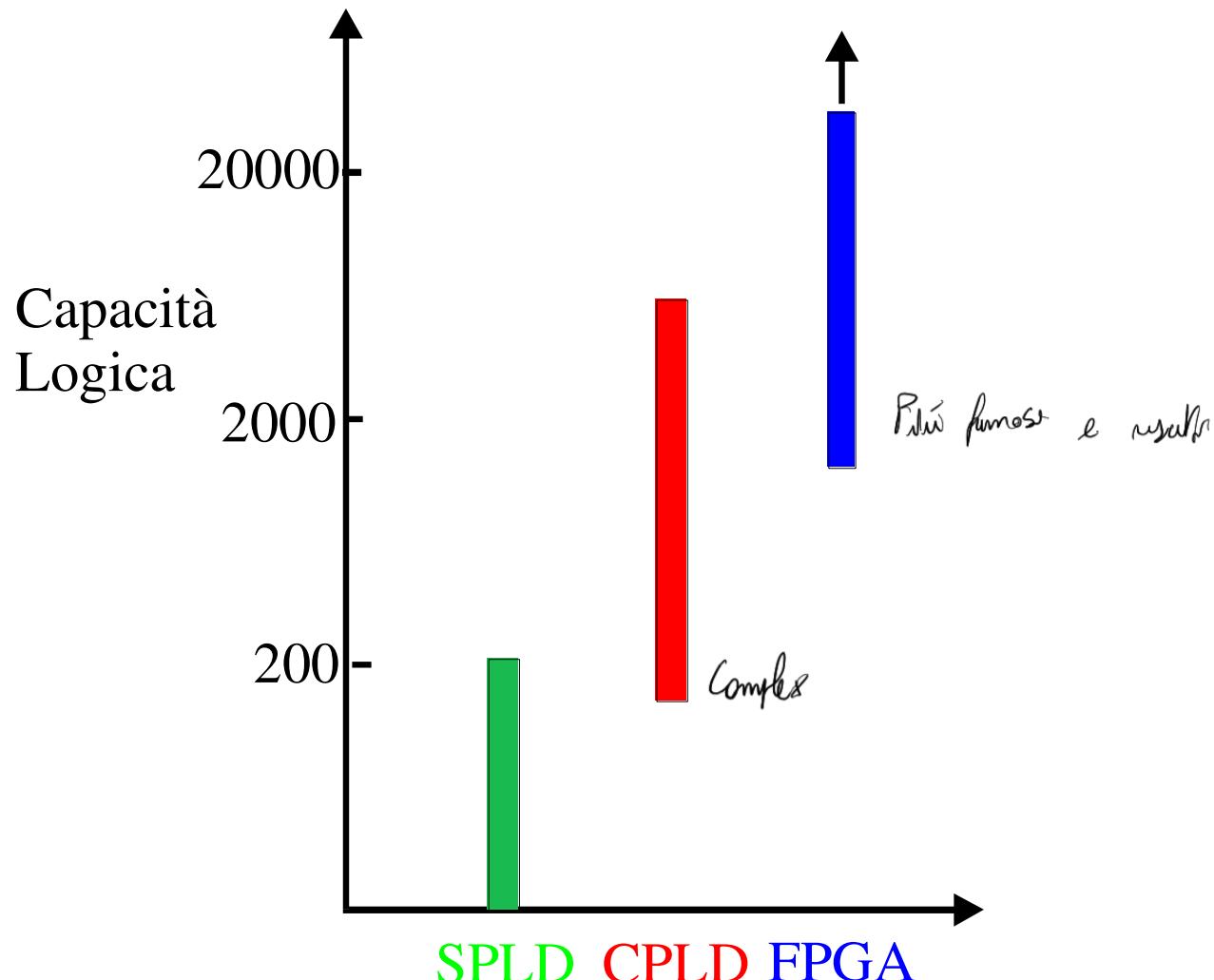
Interruttore	Riprogrammabile	Volatile	Tecnologia
Fusibile	No	No	Bipolare
Antifusibile	No	No	CMOS
EPROM	Si (fuori dal circuito)	No	UV-CMOS
EEPROM	Si (nel circuito)	No	EE-CMOS
SRAM	Si (nel circuito)	Si	CMOS

Electrical erasable programmable.

BASE: Connessione e disconnessione da pezzi da circuito
Quindi dove avrete tutte le possibili combinazioni

Capacità Logica

DOMANDA COM'È LA CAPACITÀ LOGICA?



Come si programma?

Programmazione dei PLD

Programmable Logic Device

Linguaggi più usati per la programmazione dei PLD:

ABEL Advanced Boolean Equation Language

E' utilizzato per piccoli moduli

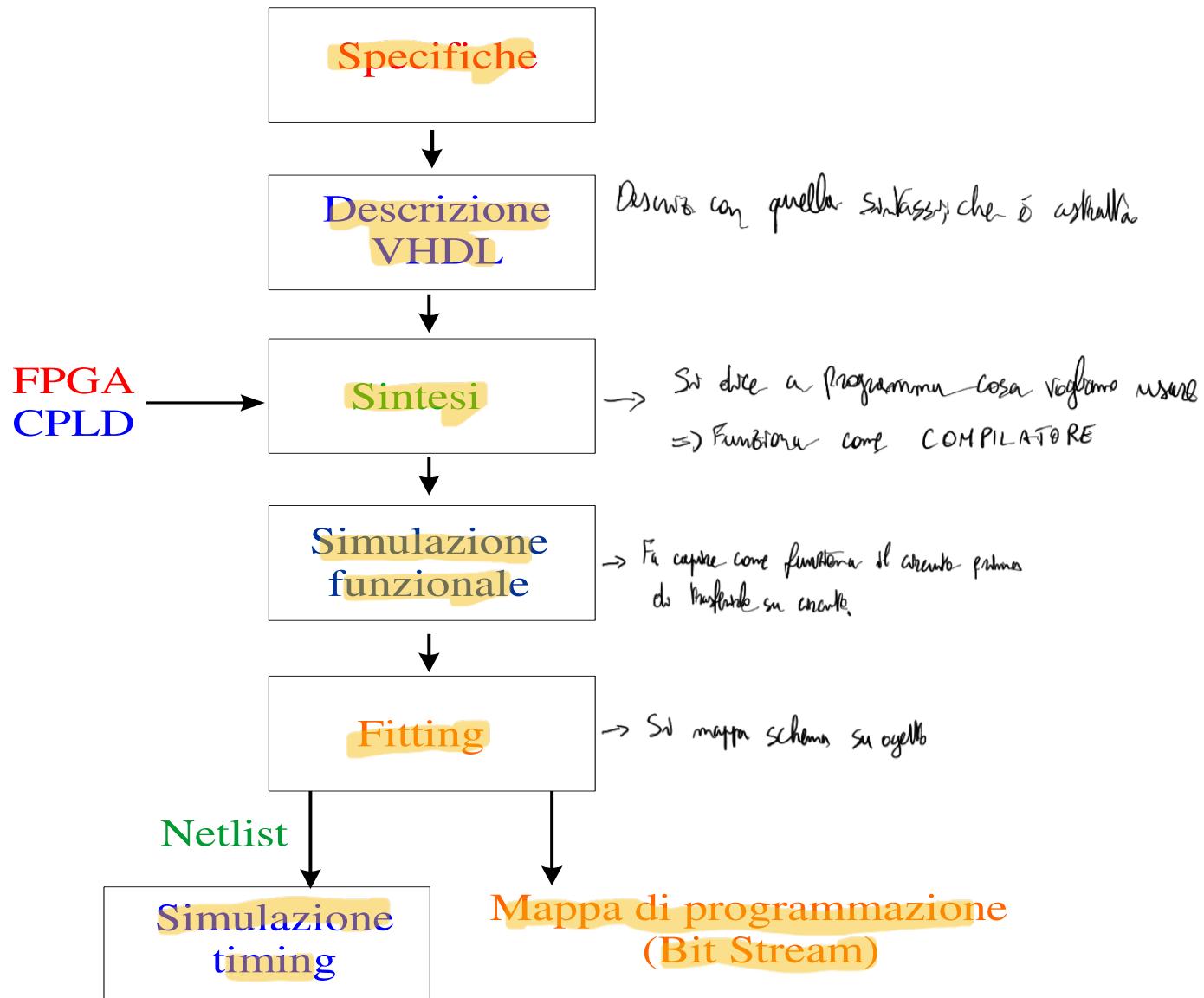
VHDL Very high speed integrated circuit Hardware Description Language

E' utilizzato per moduli complessi

I linguaggi producono un “bit stream” che viene inviato al modulo per la programmazione sul quale è inserito il PLD

Programmazione dei PLD

Programmable Logic Device



↳ Fa fare simulazioni sul tempo → Timb pronto! Generaz del BIT stream: apre o chiude gli interruttori programmabili.

Luigi Zeni DII-SUN
Fondamenti di Elettronica Digitale

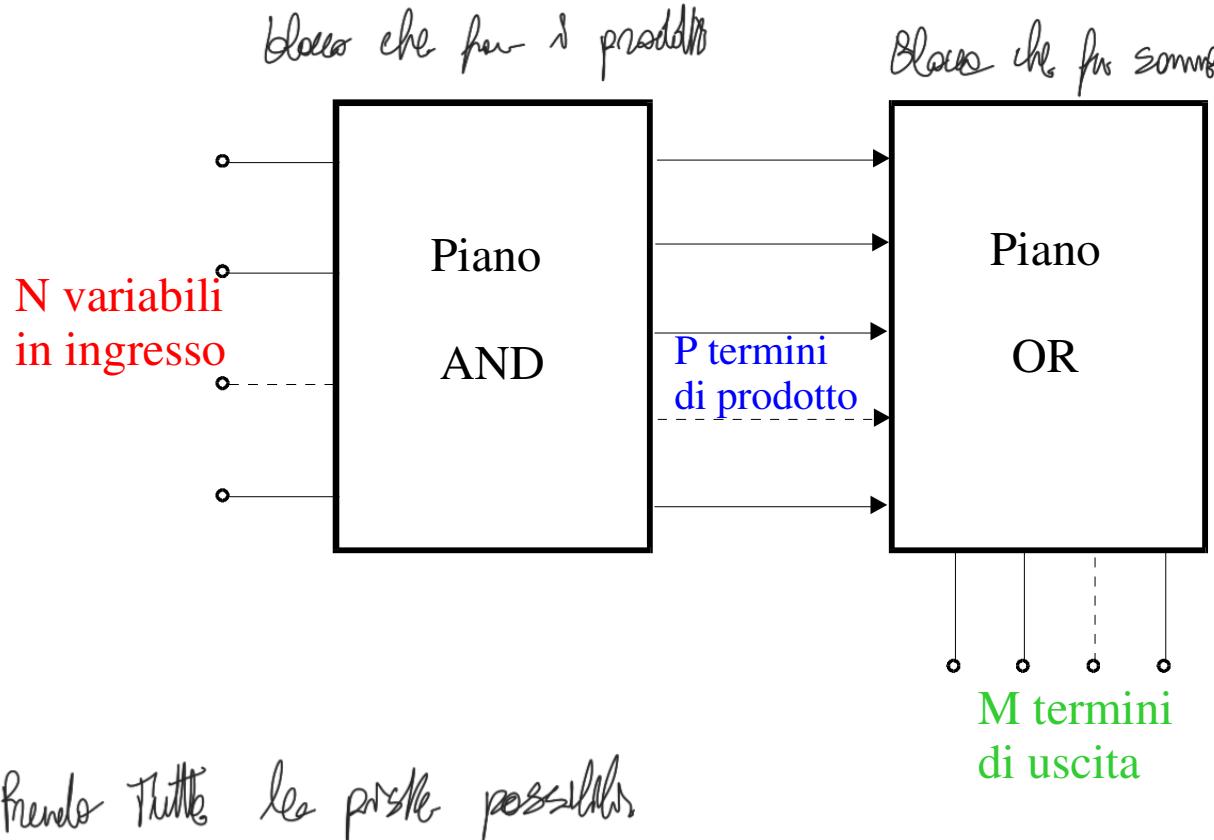
Hardware non lo vede proprio *

* Dispositivo si può staccare se non vuole.
Prensa

Struttura dei PLA

Programmable Logic Array

Crea somme di prodotti



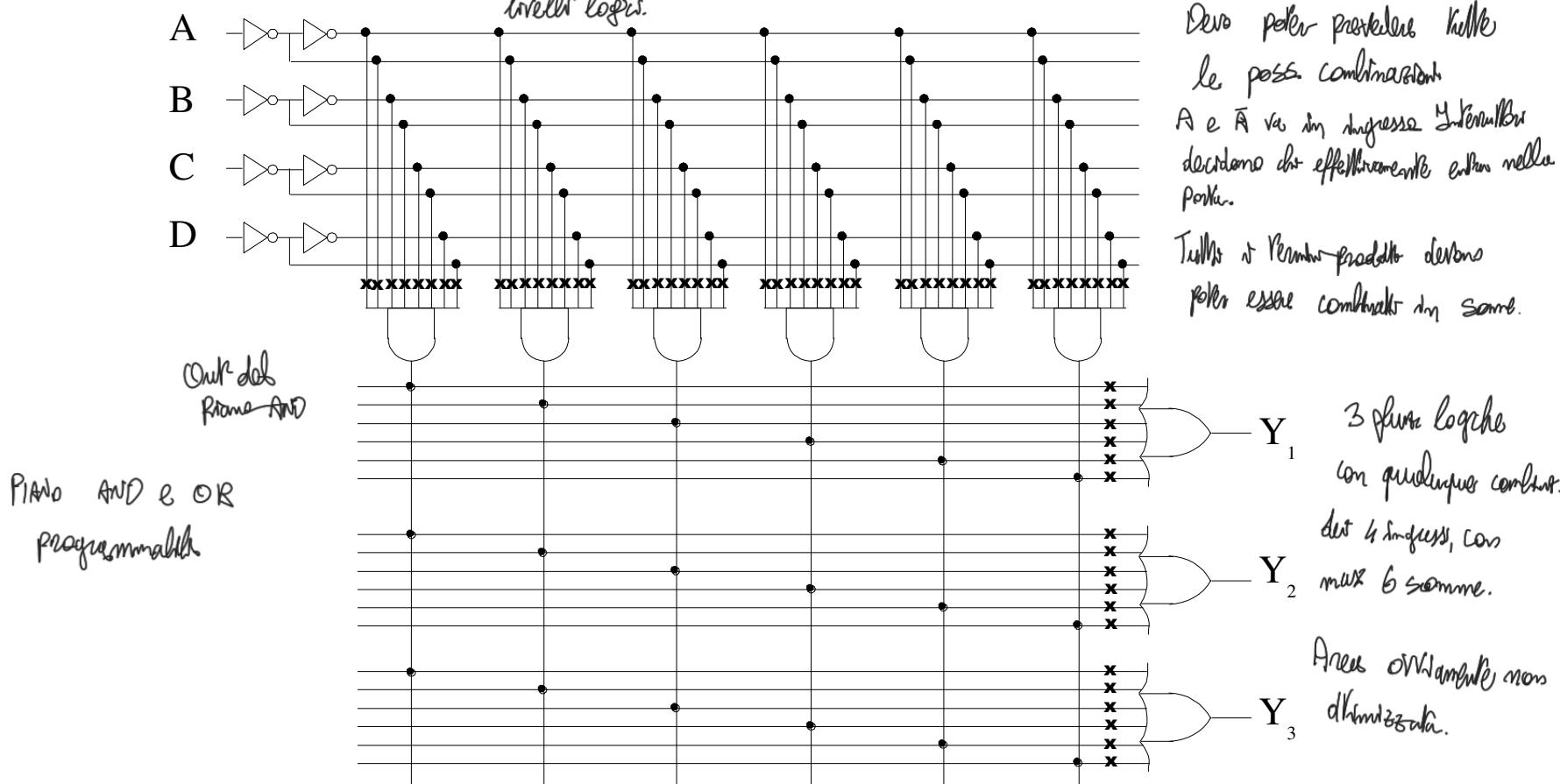
Si possono ottenere **M** espressioni logiche ognuna con **P** termini di somma e **N** prodotti per ciascun termine

Programmable Logic Array

Struttura dei PLA

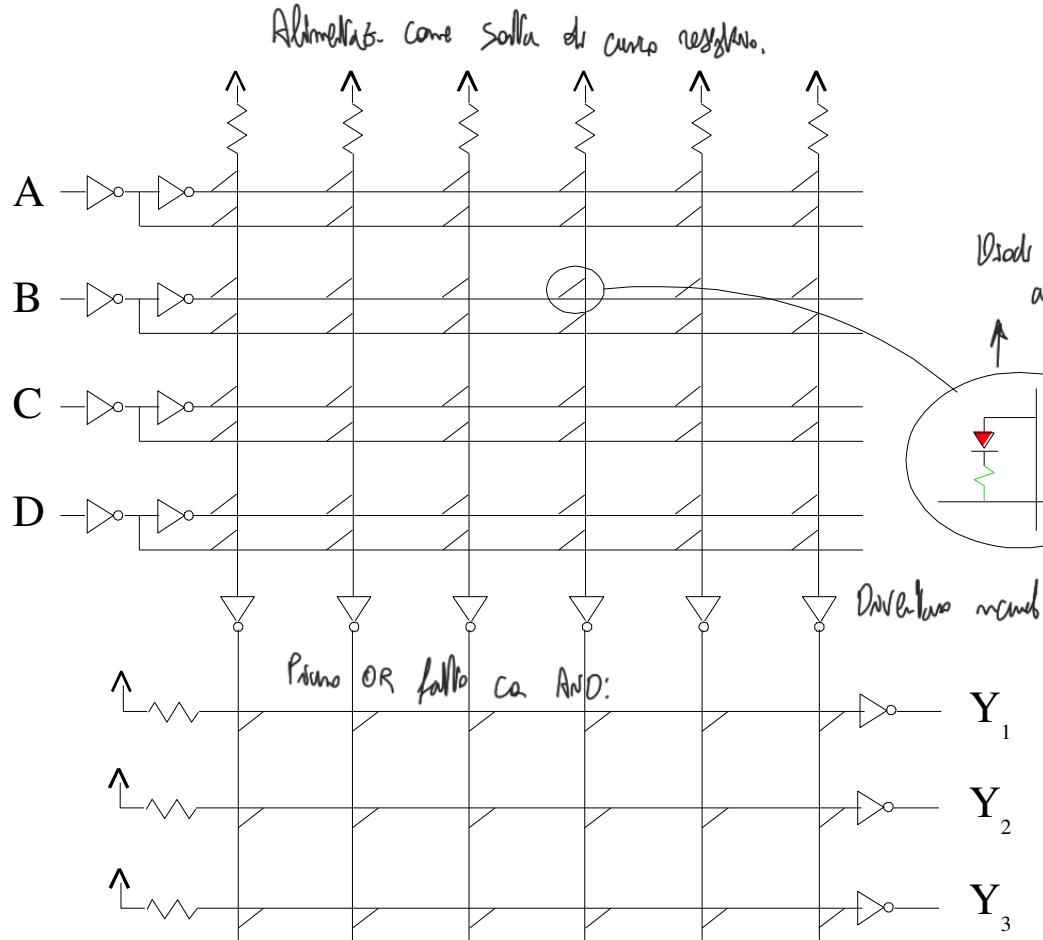
Come si subbologgi?

Dove ingressi negativi 2 volte: prevedo tutte le combinazioni, ma voglio rigenerare livelli logici.

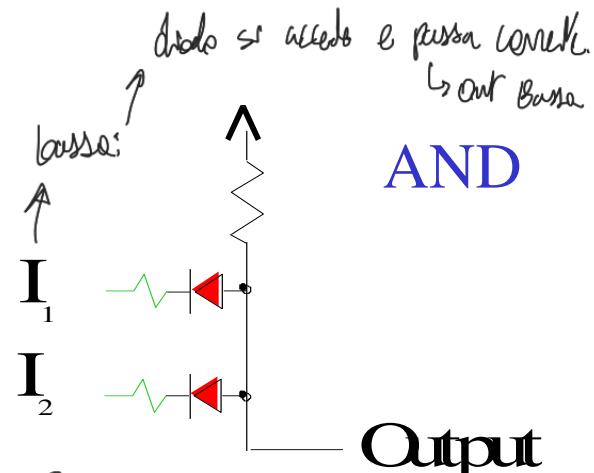


Schema logico di un PLA con 4 ingressi e 3 uscite
Le **x** indicano i punti di connessione programmabile

Realizzazione dei PLA



Programmable Logic Array



Se sono puliti e due bassi passi comelli
e ha OUT bassa.

Le due matrici implementano funzioni
AND che diventano NAND a causa degli
invertitori

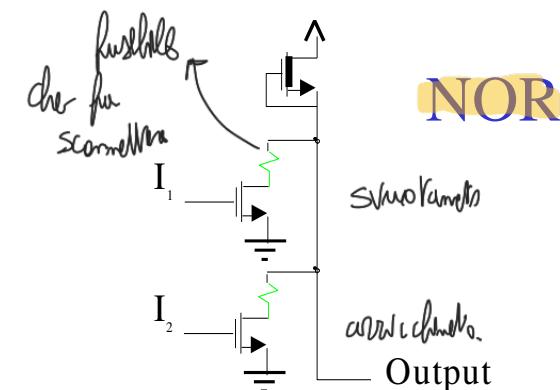
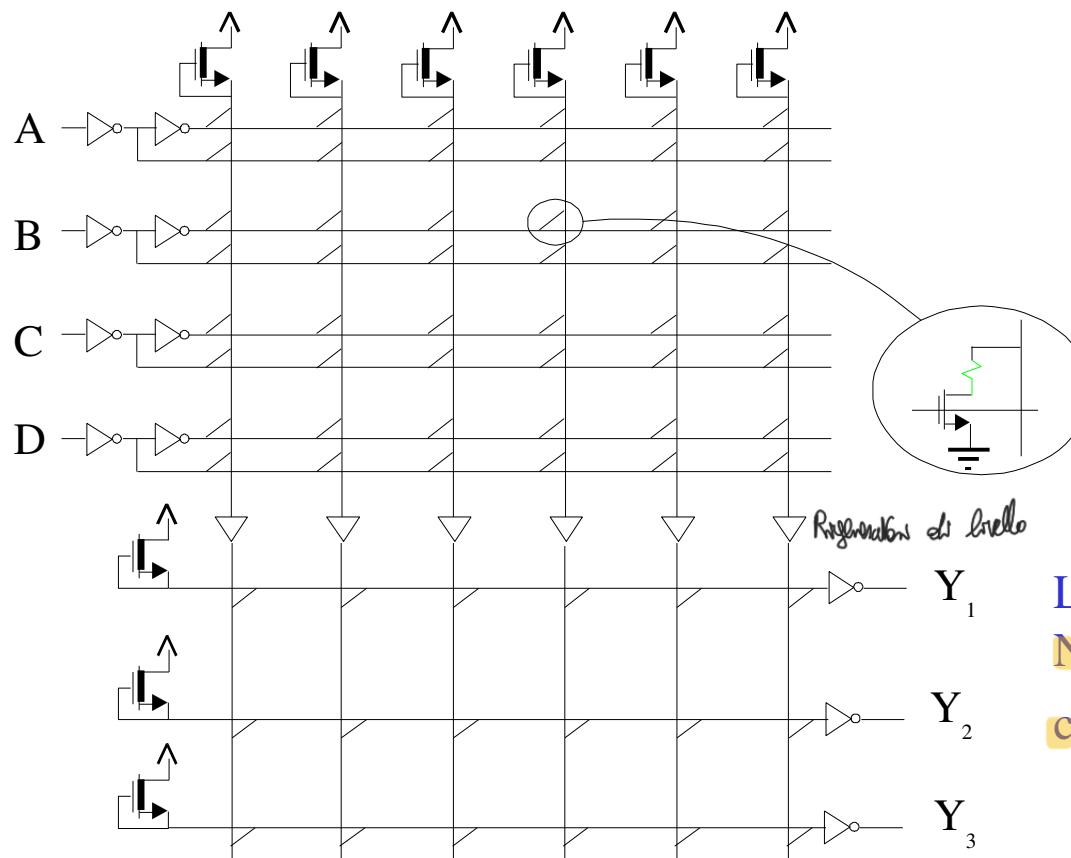
$$Y = \overline{(AB)} \cdot \overline{(AB)} = AB + AB$$

Realizzazione di un PLA mediante diodi e fusibili.

La funzione complessiva su ciascuna uscita Y è la NAND delle NAND degli ingressi cioè la OR delle AND degli ingressi

Realizzazione dei PLA

Realizzazione 2: ho carico
a svuotamento.



Le due matrici implementano funzioni NOR che, nella seconda, diventano OR a causa degli invertitori

$$Y = \overline{A+B} + \overline{A+B} = \overline{AB} + \overline{AB}$$

Realizzazione di un PLA mediante NMOS e fusibili.

La funzione complessiva su ciascuna uscita Y è la OR delle NOR degli ingressi cioè la OR delle AND degli ingressi negati (non è un problema perché gli ingressi sono disponibili anche in forma negata)

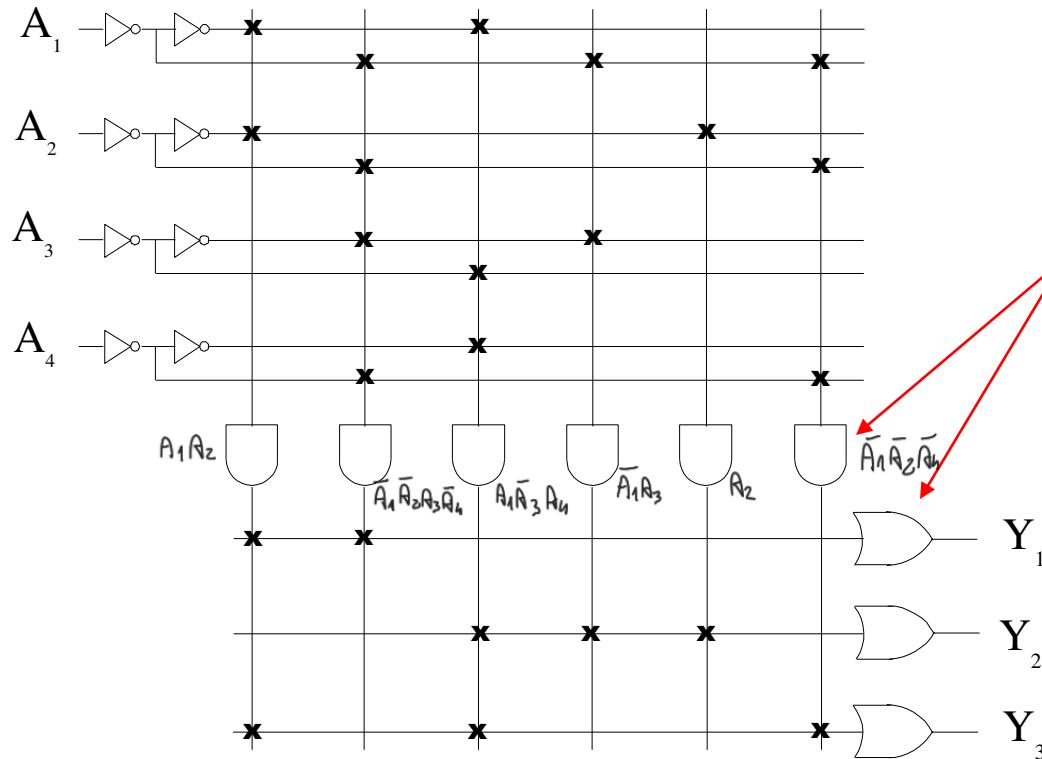
Esempio di funzioni con PLA

Programmable Logic Array

$$Y_1 = A_1 A_2 + \overline{A_1} \overline{A_2} A_3 \overline{A_4}$$

$$Y_2 = A_1 \overline{A_3} A_4 + \overline{A_1} A_3 + A_2$$

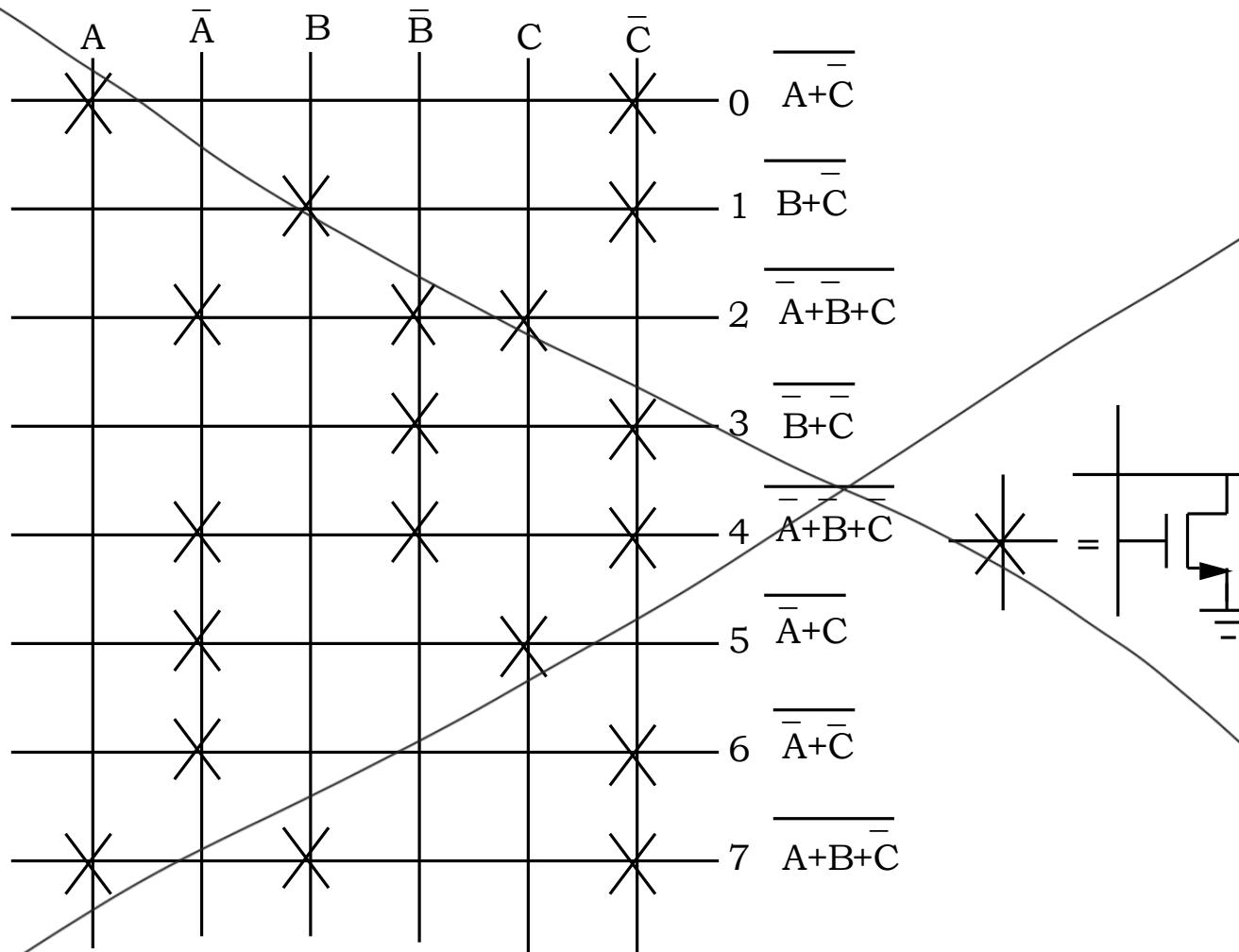
$$Y_3 = A_1 A_2 + A_1 \overline{A_3} A_4 + \overline{A_1} \overline{A_2} \overline{A_4}$$



Ovviamente le AND e le OR indicate nello schema sono solo simboliche

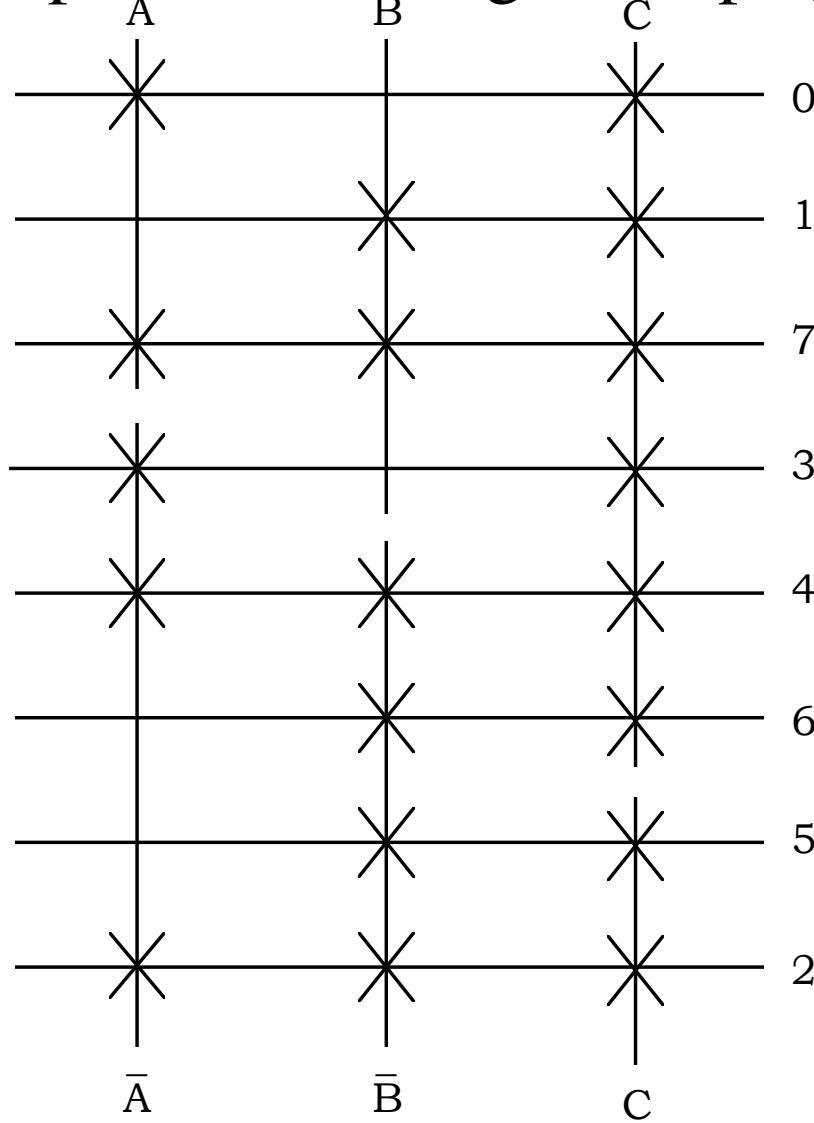
Realizzazione delle tre funzioni Y_1 Y_2 Y_3

Ottimizzazione dei PLA



Vengono utilizzate solo il 40% delle possibili connessioni

Operazione di taglio e ripiegamento nei PLA



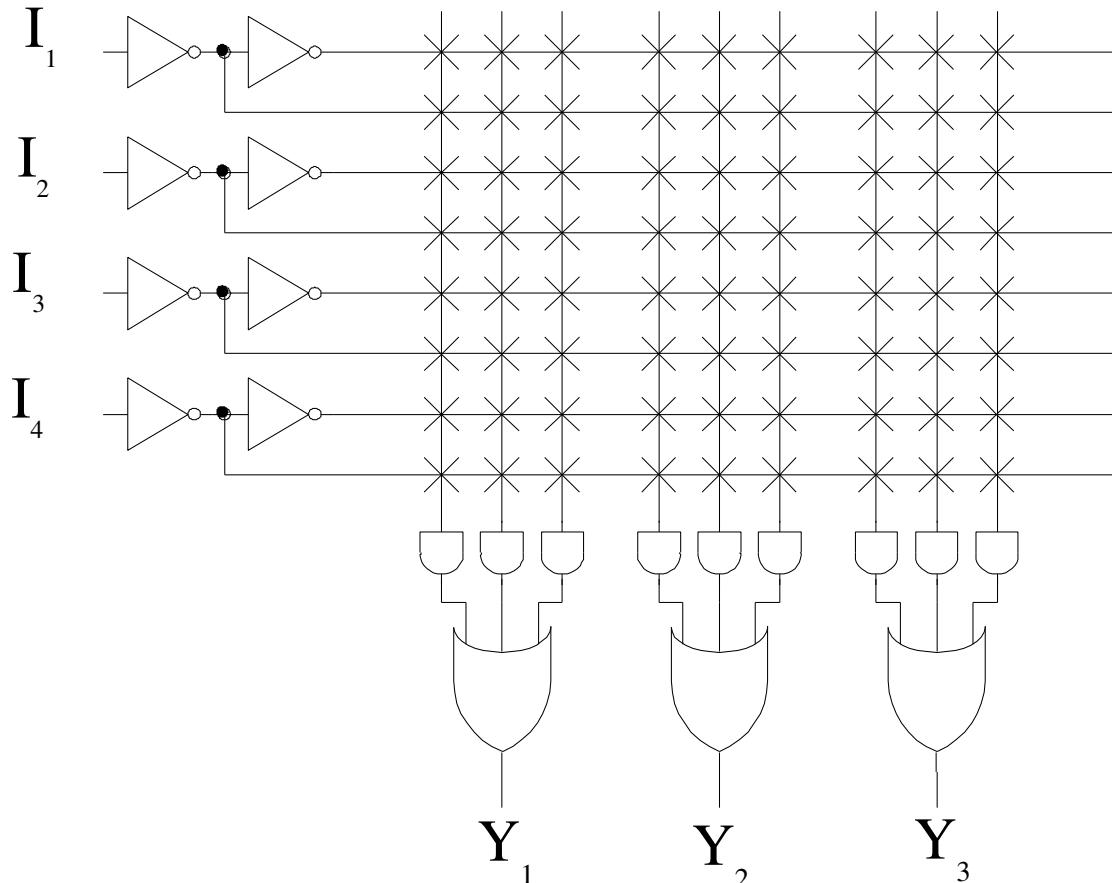
Si possono compattare
funzioni uguali.

Le righe vengono
spostate e vengono
praticati tagli per
compattare la matrice.
Ovviamente si perde di
generalità ma si
guadagna in area
occupata

Dopo la riorganizzazione delle righe si utilizza il 79% delle connessioni

Programmable Array Logic

Dispositivi PAL



Programmabile solo il piano AND; MENO VERSATILE

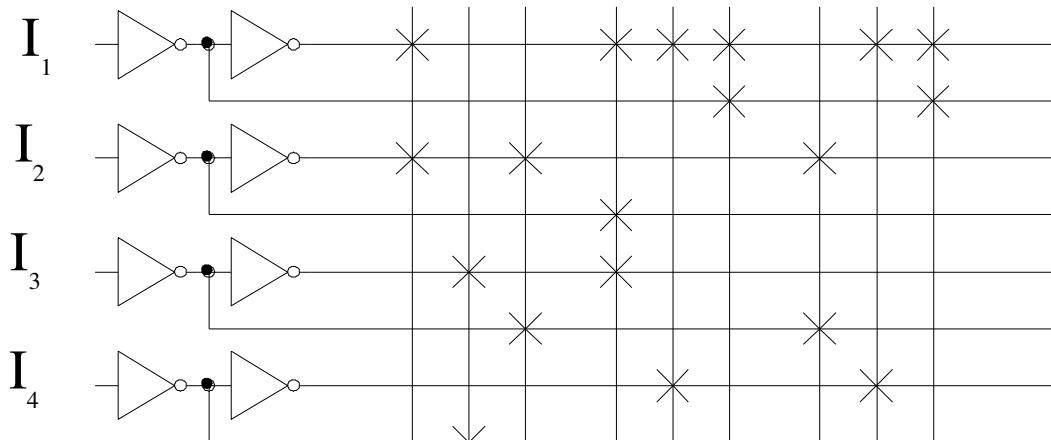
4 ingressi, 3 termini da prodotto. OR sono max a 3 termini.

E' programmabile solo il piano AND e, in questo caso, ogni uscita può avere solo 3 termini prodotto.

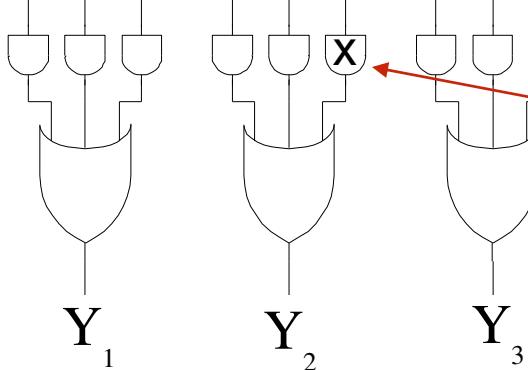
Nelle PAL commerciali il numero di termini prodotto varia da 6 a 8.

Esempio di PAL

Programmable Array Logic



Prima ingresso Variabile
Qua no, ho versione più
semplificata



$$Y_1 = I_1 I_2 + I_3 \bar{I}_4 + I_2 \bar{I}_3$$

$$Y_2 = I_1 \bar{I}_2 I_3 + I_1 I_4$$

$$Y_3 = I_2 \bar{I}_3 + I_1 I_4$$

Prima OR usata appena
Terzo ingresso mancante

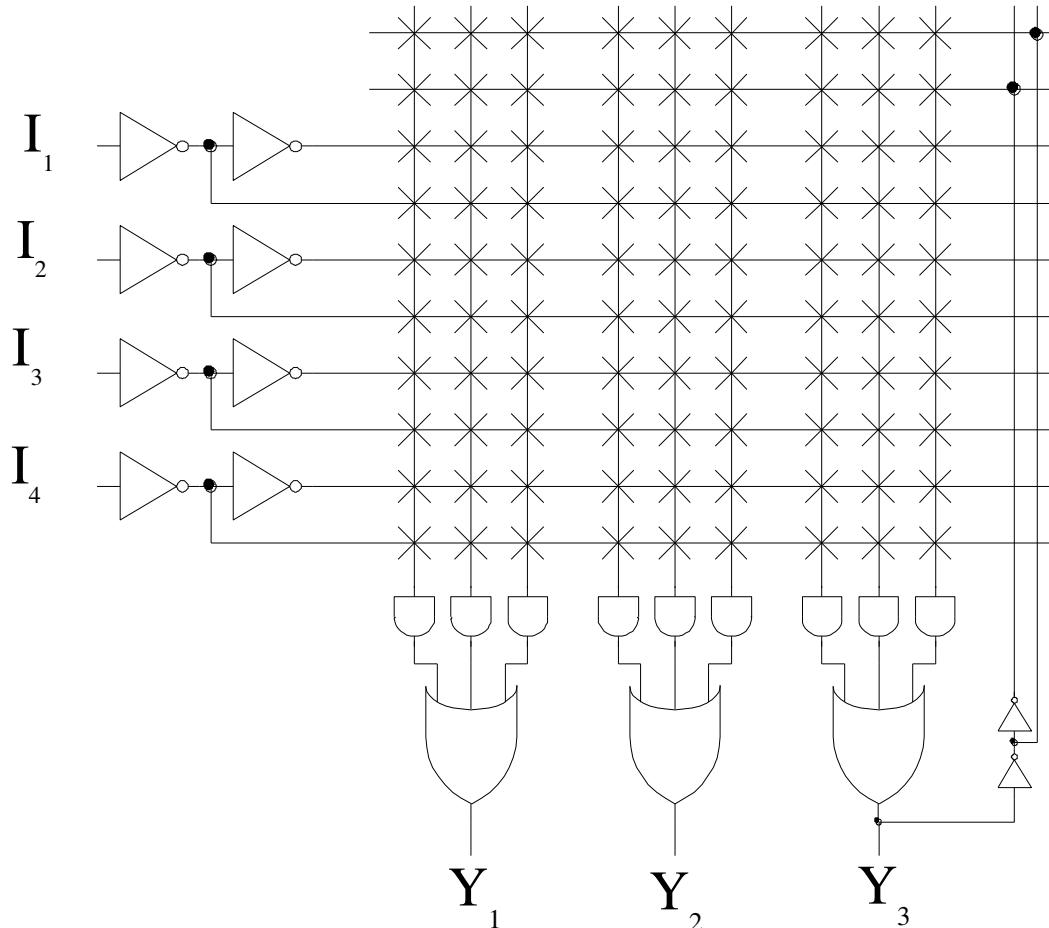
Valori di uscita sempre 0 in
quanto ottenuti dalla AND di
una variabile e la sua negata

Se un termine prodotto è necessario per due uscite viene calcolato due volte come i termini $I_1 I_4$ e $I_2 \bar{I}_3$

Logica in due passi per dispositivi PAL

Programmable Array Logic

Per consentire la realizzazione di funzioni con un numero maggiore di termini prodotto si ricorre alla logica in due passi, cioè alcune uscite vengono inviate come nuovi ingressi al piano AND



Si aumenta la flessibilità
ma anche il ritardo di
propagazione

Per mitigare l'effetto di non più 3 termini
di prodotto, uso questi. Un'uscita della
OR viene riportata all'ingresso e
si aggiunge valore all'ingresso.

Logica in due passi per dispositivi PAL

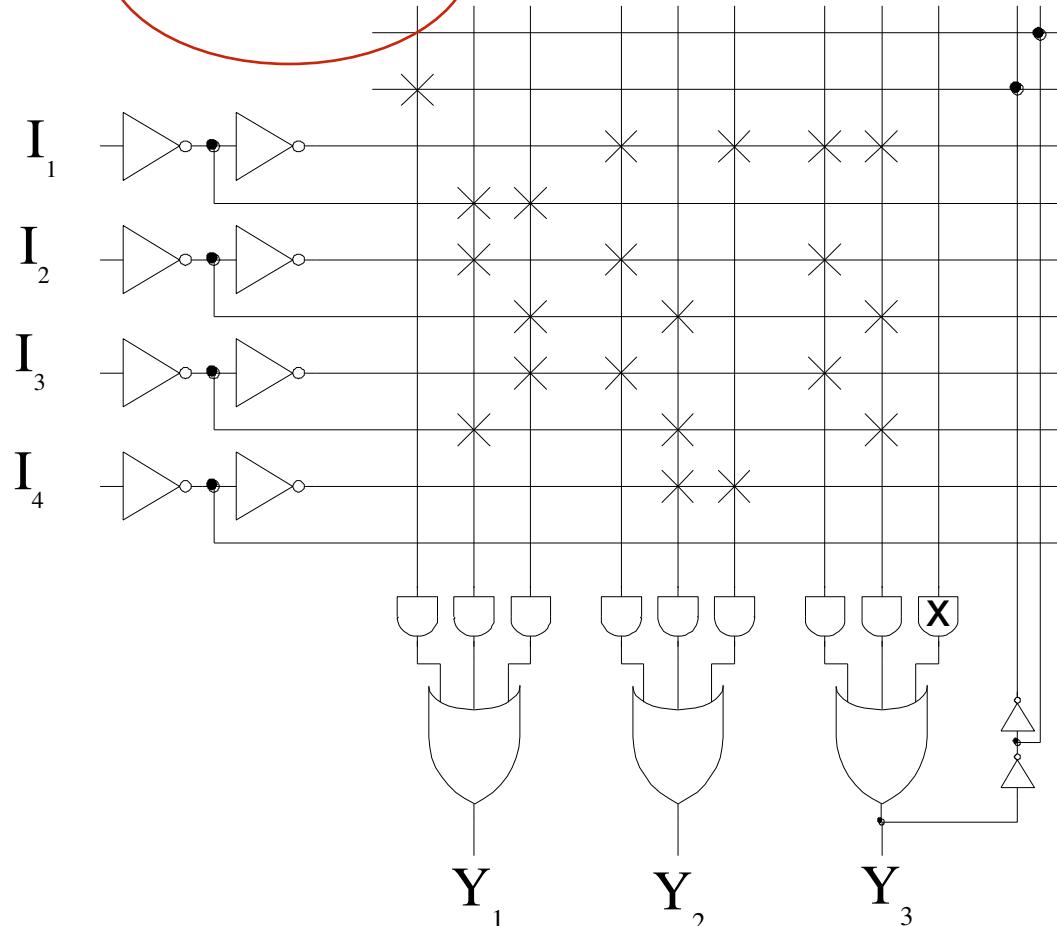
Programmable Array Logic

Realizzazione delle funzioni Y_1 e Y_2 con una PAL a 3 prodotti

$$Y_1 = I_1 I_2 I_3 + I_1 \overline{I}_2 \overline{I}_3 + \overline{I}_1 I_2 \overline{I}_3 + \overline{I}_1 \overline{I}_2 I_3 *$$

$$Y_2 = I_1 I_2 I_3 + \overline{I}_2 \overline{I}_3 I_4 + I_1 I_4$$

* Immagine di voler fare le somme.
Ma ho bisogno, dm 2 passi, posso generare.
Mi servono 2 somme: Y_1 e Y_2 . Usò Y_3
 Y_3 fu un pezzo di Y_1 .



$$Y_3 = I_1 I_2 I_3 + I_1 \overline{I}_2 \overline{I}_3$$

Si utilizza l'uscita Y_3 per generare i primi due termini della funzione Y_1

Selezione della polarità in uscita per dispositivi PAL

Realizzazione delle funzioni Y_1 e Y_2 con una PAL a 3 prodotti

$$Y_1 = I_1 I_2 I_3 + I_1 \bar{I}_2 \bar{I}_3 + \bar{I}_1 I_2 \bar{I}_3 + \bar{I}_1 \bar{I}_2 \bar{I}_3$$

$$Y_2 = I_1 + I_2 + I_3 + I_4$$

Non è possibile implementare le funzioni direttamente, neanche con la logica in due passi, perché sia Y_1 che Y_2 presentano più di 3 termini prodotto.

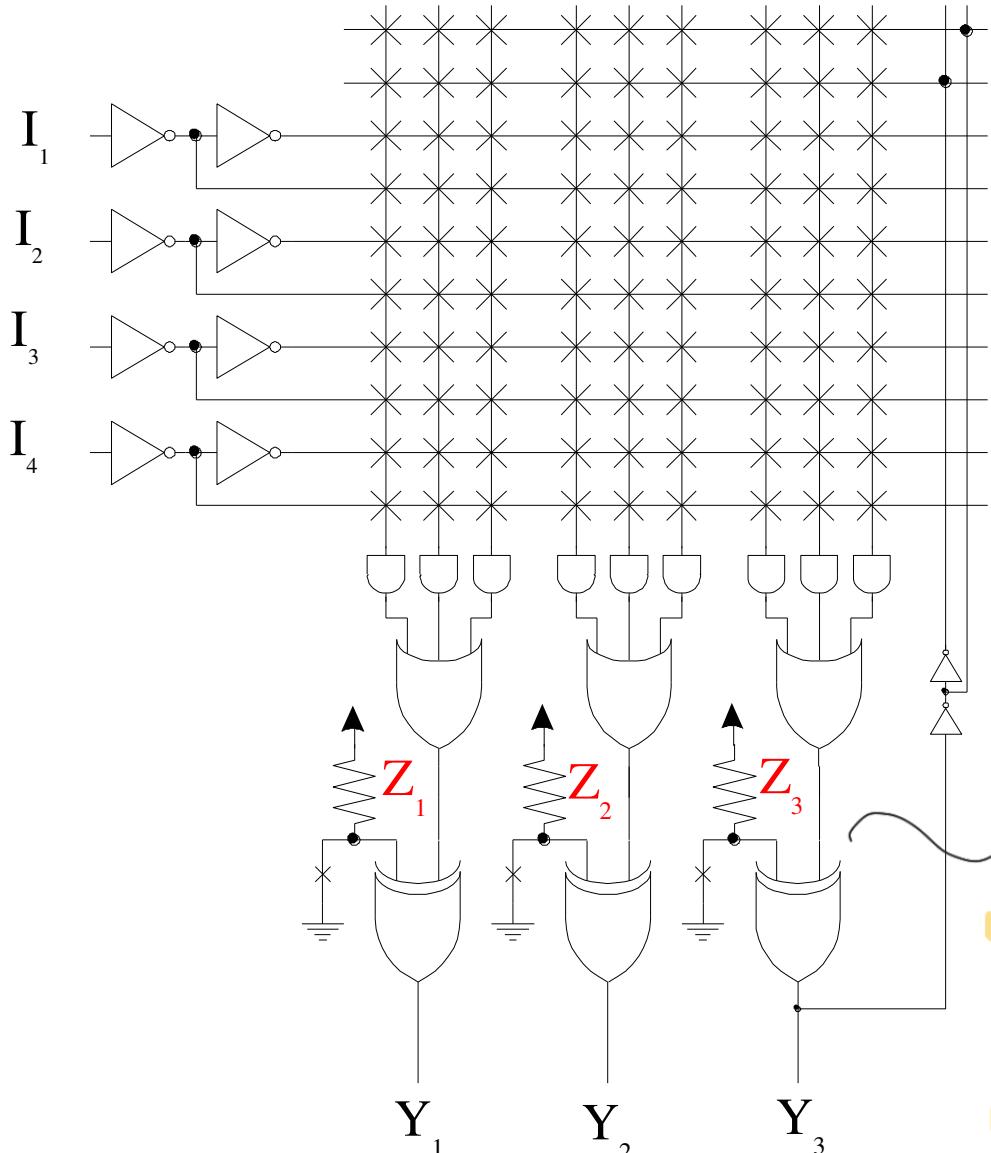
Sarebbe invece possibile implementare $\bar{Y}_2 = \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4$ che contiene un solo termine prodotto e usare:

$$Y_3 = I_1 I_2 I_3 + I_1 \bar{I}_2 \bar{I}_3$$

come nel caso precedente, in una logica a due passi.

Occorre quindi negare le uscite!

PAL con polarità dell'uscita programmabile

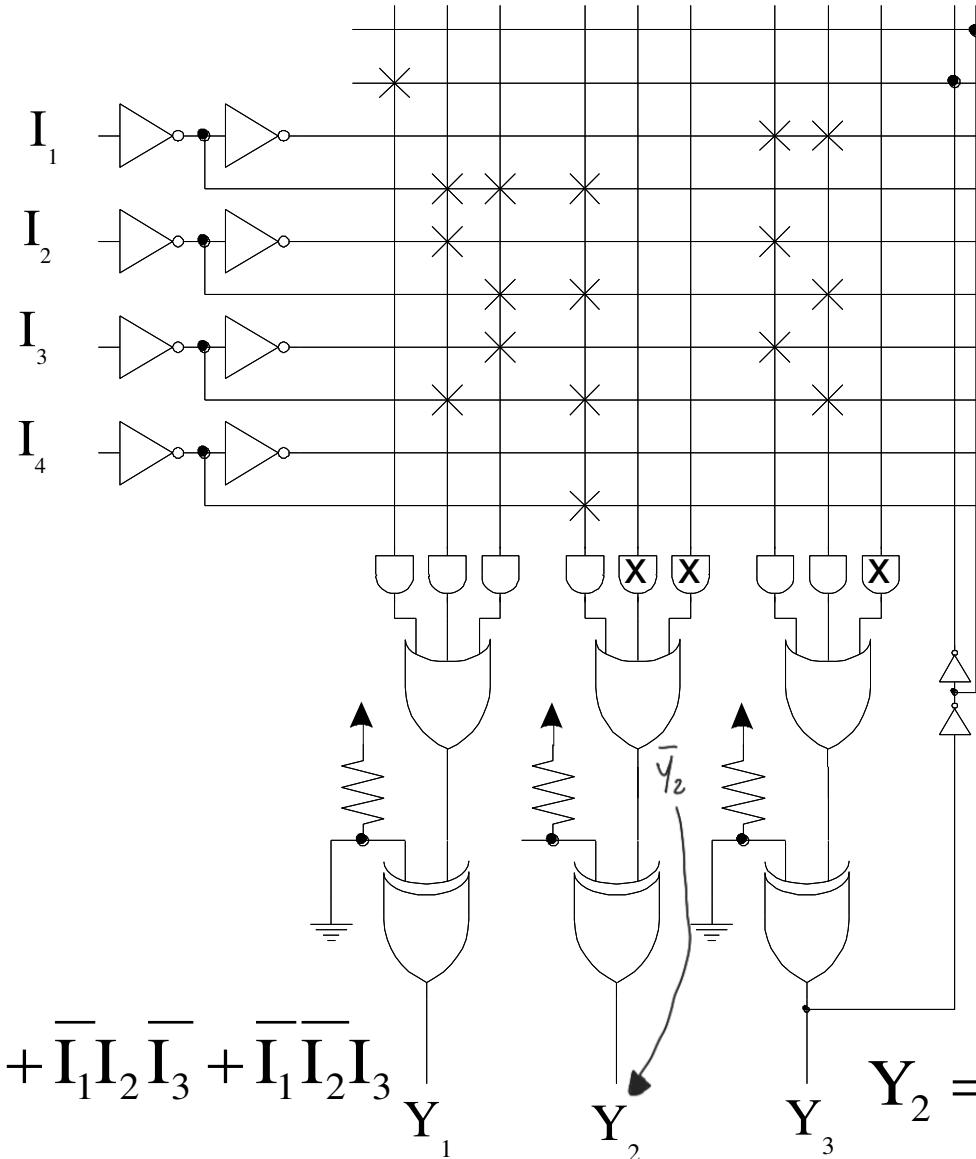


A	B	$A \text{ XOR } B$
0	0	0
0	1	1
1	0	1
1	1	0

Con scherzo di Z_3 , collegabile ad alimentaz.
o a massa a seconda di X_3 .

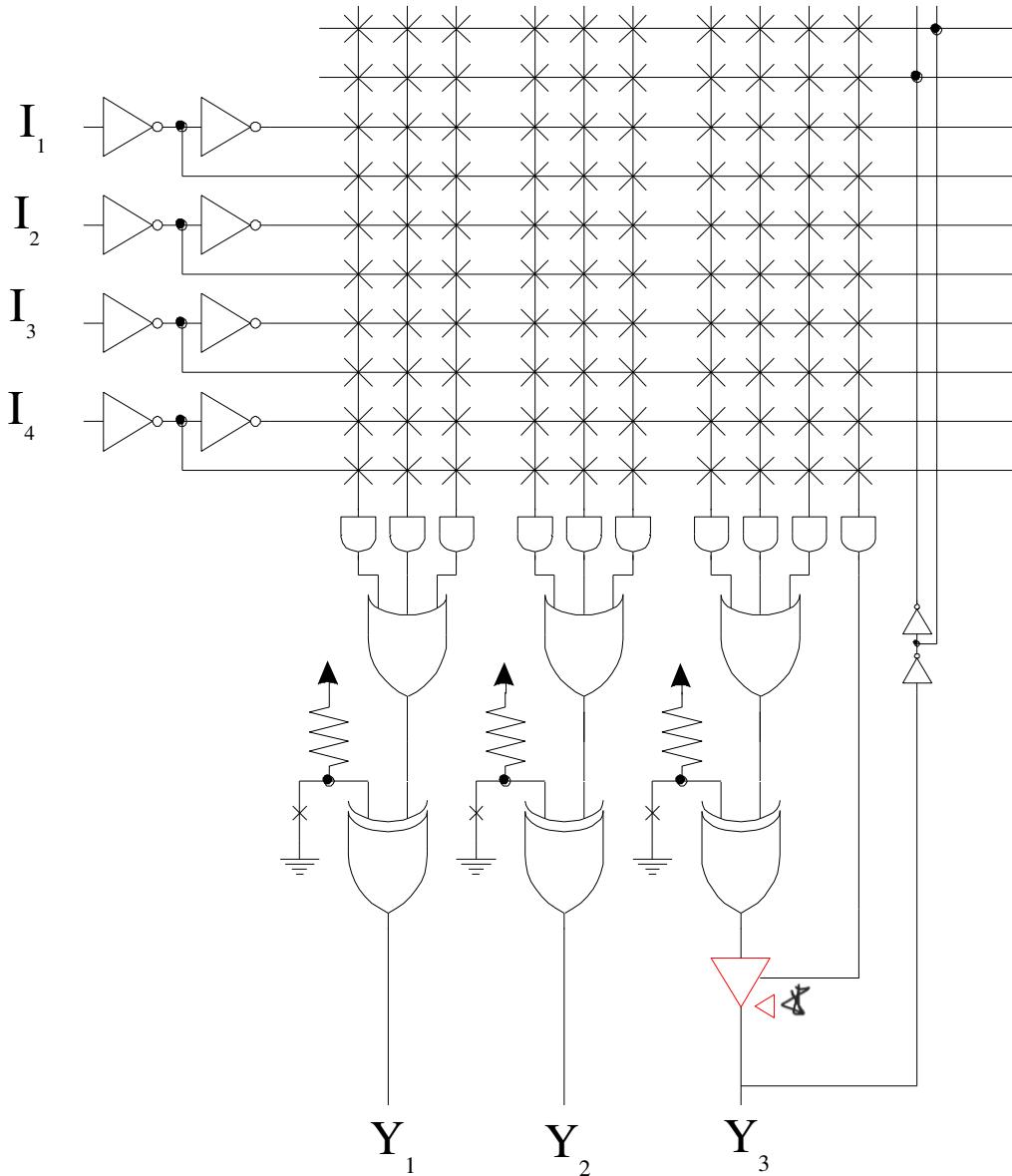
Si utilizzano porte XOR programmabili
in cui uno dei due ingressi può essere
posto o a 0 o a 1 ottenendo in uscita
 $Y = Z$ oppure $Y = \bar{Z}$, rispettivamente

PAL con polarità dell'uscita programmabile



Luigi Zeni DII-SUN
Fondamenti di Elettronica Digitale

PAL con terminali I/O programmabili

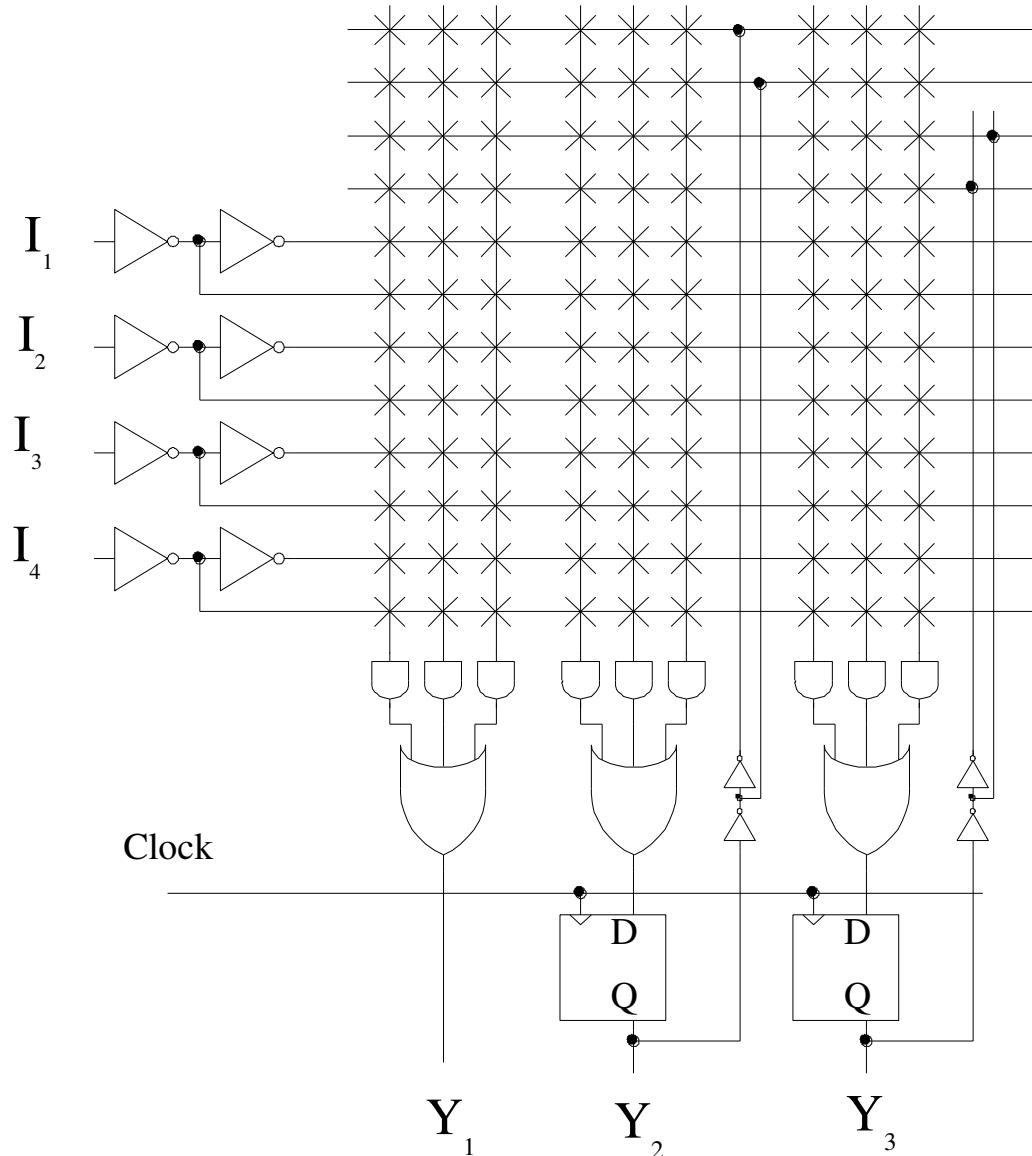


* Pin Y_3 del chip non vede più quello che viene dalla matrice perché appena, e posso portare dentro segnale che viene da fuori.
Se ho molti altri zeri appeso e butto dentro segnale.

Si utilizza un buffer **tristate**, pilotato da una uscita dedicata del piano AND, per avere Y_3 come terminale di I/O

L'abilitazione del **tristate** può essere funzione degli ingressi o essere fissata, a secondo dei casi.

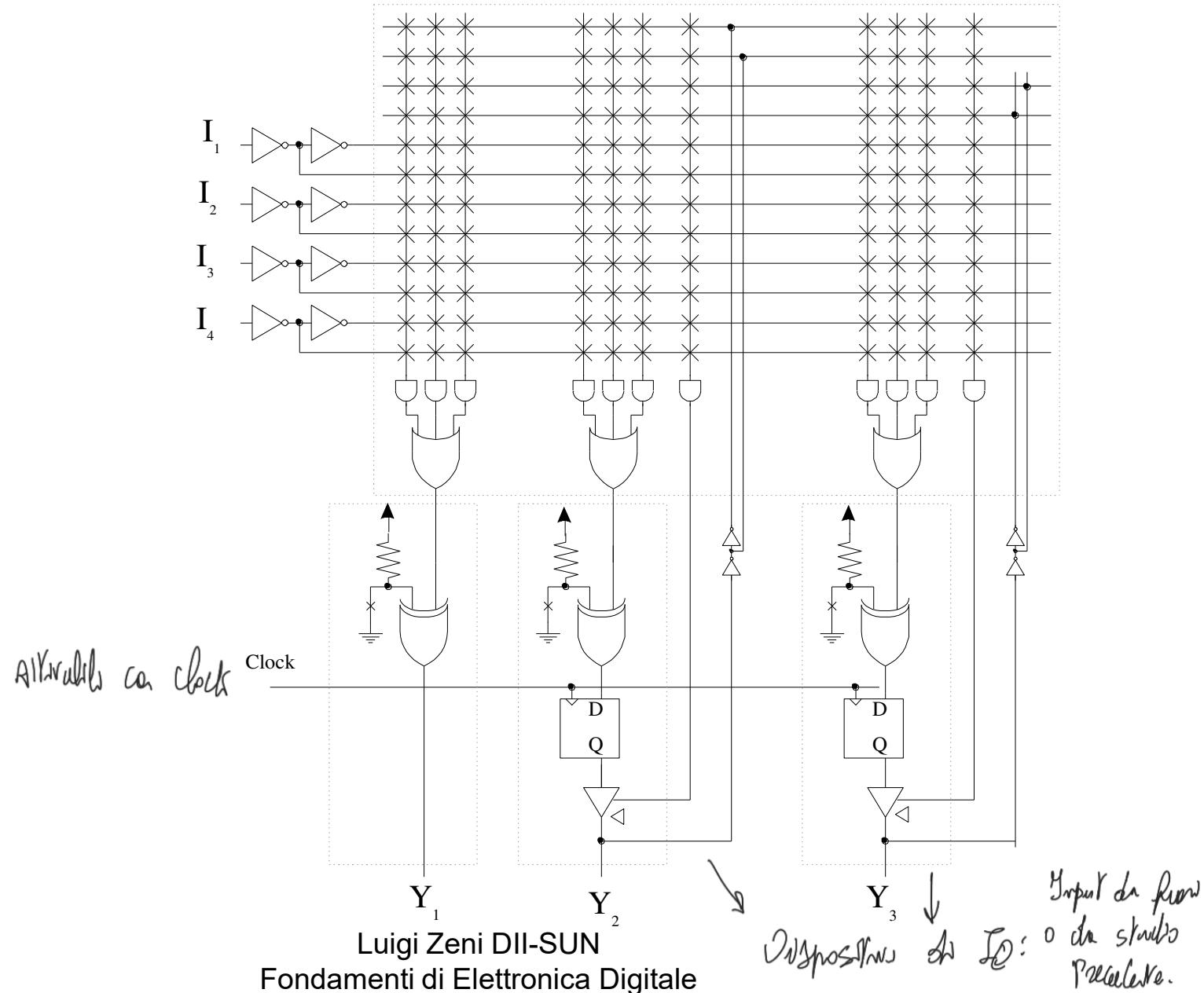
PLD sequenziali



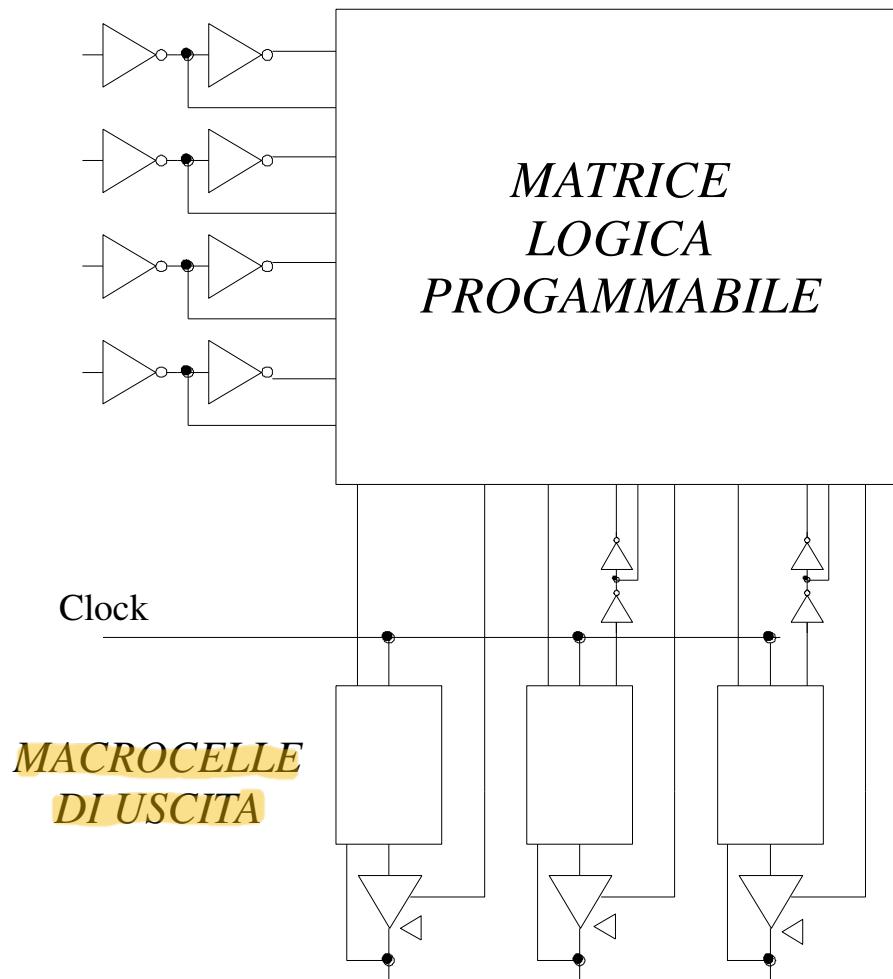
Contengono **Flip-Flop**, per consentire la realizzazione di circuiti sequenziali.

Usiamo anche dei flip flop. Uno degli ingressi lo posso prendere dallo stato precedente. Questo per compattare PAL per renderlo più potente.
Non si mette su tutte e quante. Voglio compresso per semplicità.

PAL sequenziali con terminali I/O programmabili



Schema a blocchi semplificato di PAL

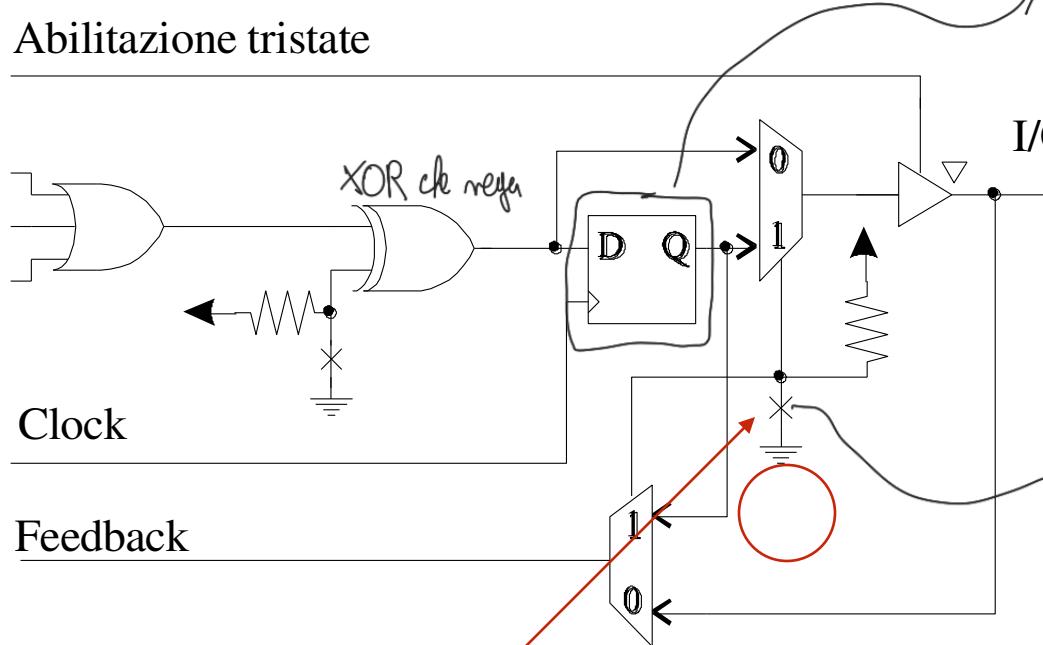


La struttura può essere vista come costituita da una matrice logica e diverse macrocelle in uscita.

Normalmente si configura macrocelle per realizz. circuiti o combinatori o sequenziali.

Possibile struttura delle macrocelle

Abilitazione tristate



Metto FF e uso 2 multiplexer
per tutti i controlli,
I/O pin posso costituire
modo da farla lavorare
sequenziale o combinatoria.

Se è chiuso, non c'è
busso. Passa carico di sopra.
Se aperto è comb. logica 0.

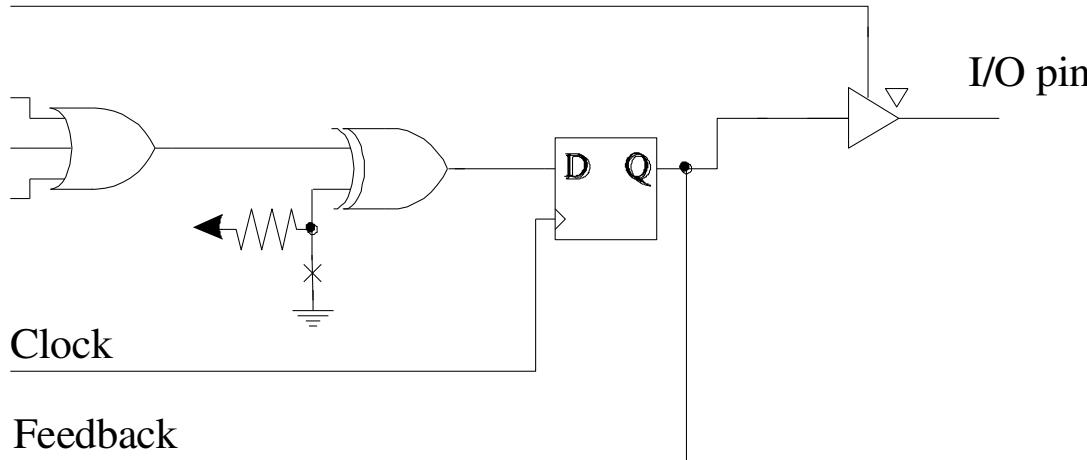
Evidentemente il feedback viene da lì oppure

I multiplexer 2-1 programmabili consentono di escludere completamente il Flip-Flop, conservando in ogni caso la possibilità di avere il terminale I/O programmabile

* lo uso come ingresso.

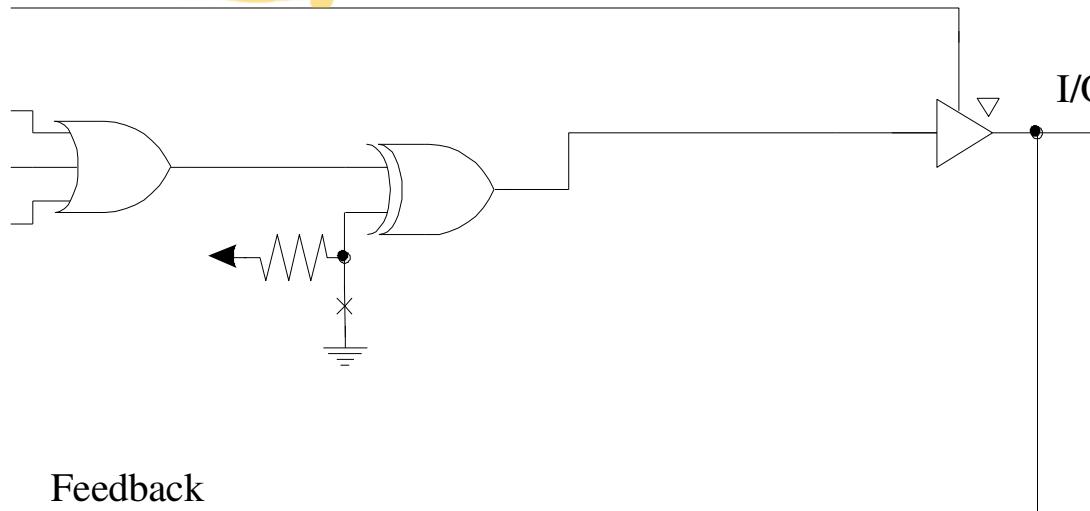
Macrocella in modalità sequenziale e combinatoria

Abilitazione tristate



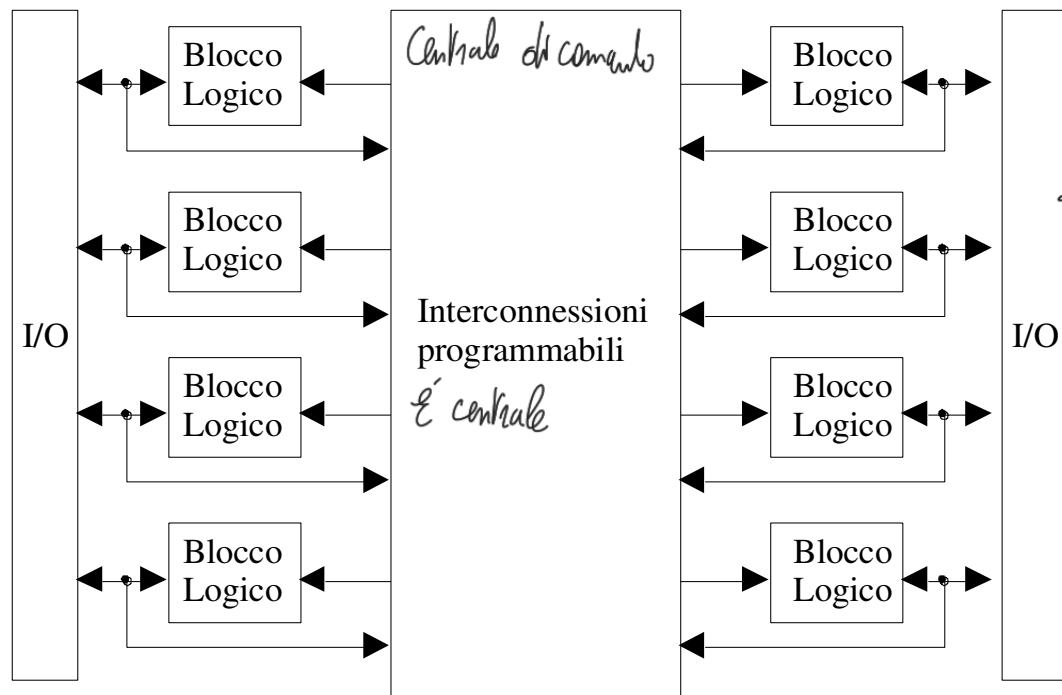
Feedback

Abilitazione tristate



Feedback

CPLD



Possible architecture:
blochi parlano fra loro
con matrice programmabile
→ Qui parla con più estens.
* Ciascun blocco parla
con la matrice interna. Blochi
parlano fra loro attraverso la
matrice. Mediante V_{DD} i
blochi scambiano info con
lo stesso percorso.

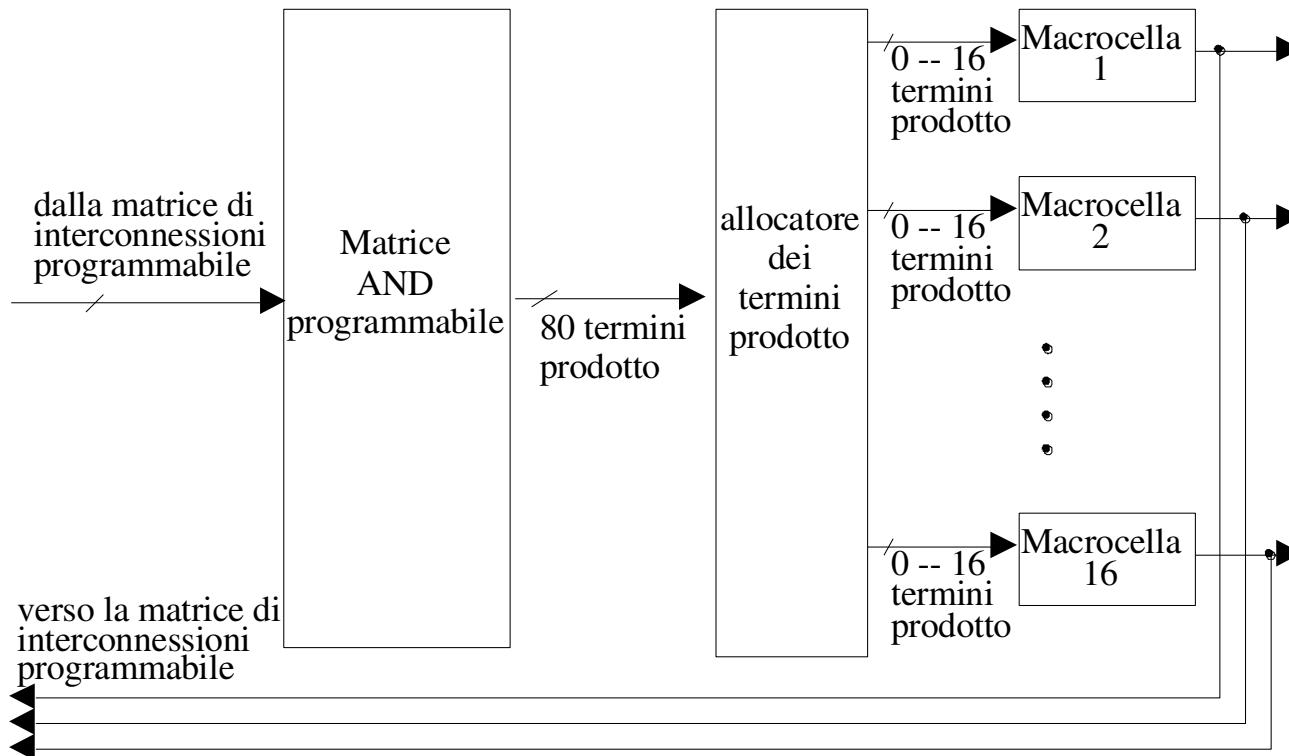
Consentono di aumentare le potenzialità del sistema senza aumentare eccessivamente il numero di ingressi dei singoli blocchi logici, ciascuno costituito da una matrice tipo PAL.

Infatti aumentare troppo il numero di ingressi del singolo blocco comporterebbe insostenibili problemi di FAN-IN delle porte che costituiscono il piano AND.

Il ritardo complessivo dipende poco dalla complessità del progetto. *

Schema di blocco logico di un CPLD

Matrice programmabile per indirizzare le porte



Cypress 370

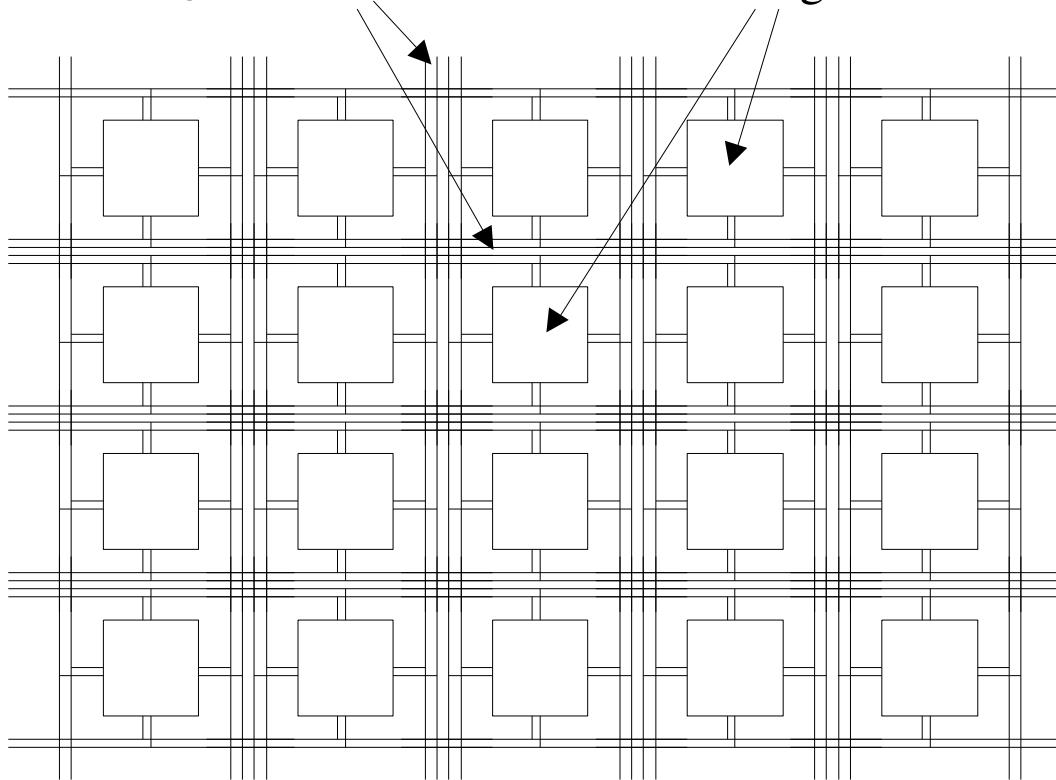
Caratteristiche dei CPLD della famiglia Altera FLEX10K

Sigla	EPF10K10	EPF10K30	EPF10K50	EPF10K100
Capacità Logica	10000	30000	50000	100000
N. Blocchi Logici	576	1728	2888	49992
N. Flip-Flop	720	1968	3184	5392
N. Pin I/O	150	246	310	406

Struttura generale degli FPGA

Canali di collegamento

Celle logiche

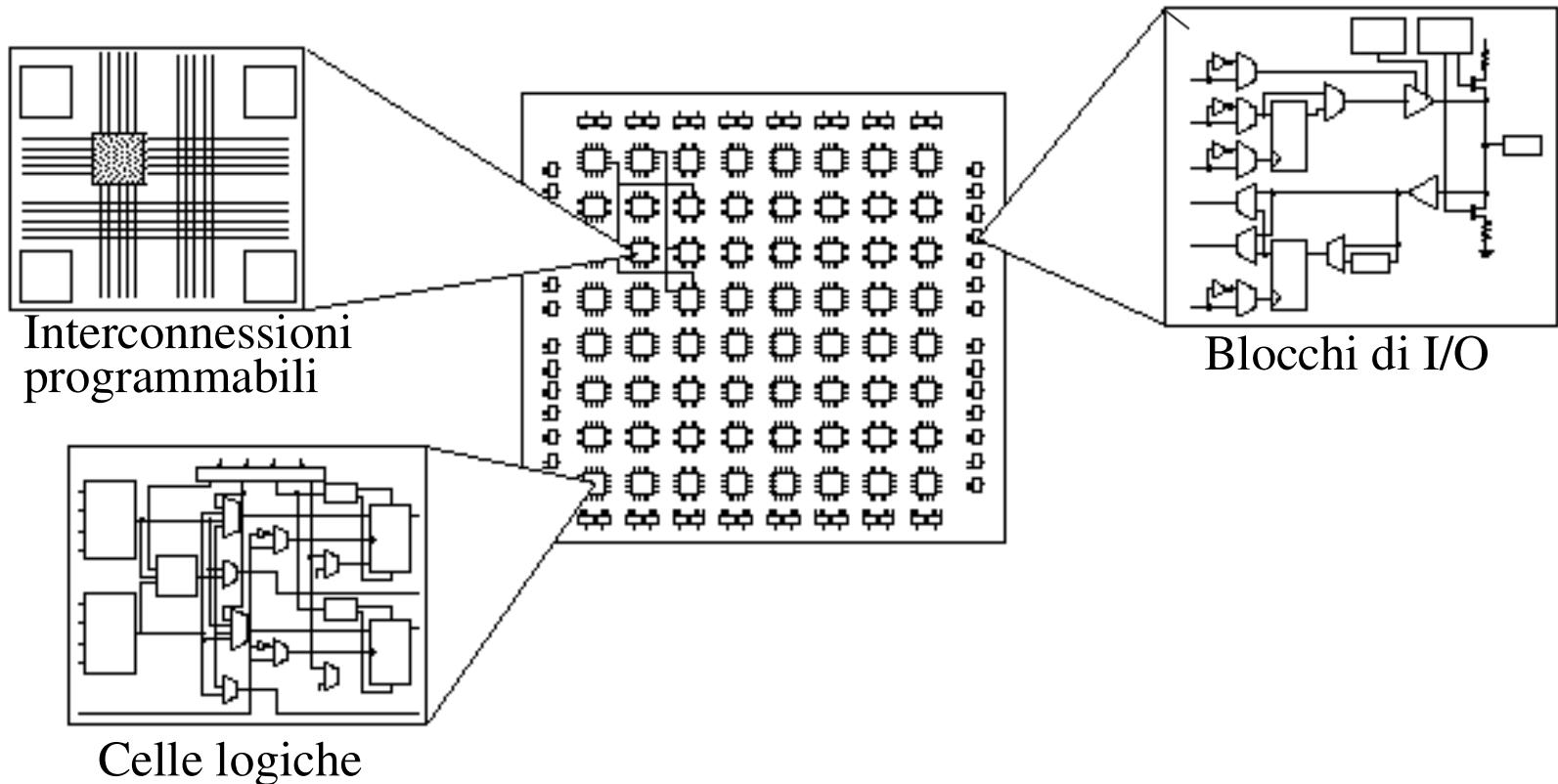


Molti ce passa per tutto il chip e i blocchi sono distribuiti uniformemente; quindi può dipendere dalla posizione

Le celle logiche hanno una funzionalità ridotta rispetto ai blocchi logici dei CPLD ma sono in numero molto maggiore. Gli FPGA presentano quindi una architettura a "grana fine" ed inoltre il numero elevato di celle consente di avere a disposizione un numero elevato di Flip-Flop.

.

Struttura interna di un FPGA



Il ritardo complessivo di un FPGA dipende in maniera marcata dallo schema che si vuole realizzare, a differenza dei CPLD.

- 1) Descrivere i diversi programmi
- 2) Implementare funzione PAL (con doppio passaggio)
- 3) Schema delle macrocella di memoria (slide 2 prima)
- 4) Cos'è la capacità logica
- 5) Schema RAM/Decodificatore (22-23-24)

