



DIPARTIMENTO DI ELETTRONICA E INFORMAZIONE



**POLITECNICO
DI MILANO**

Dispositivi Logici Programmabili



PLA e PAL avanzate
Martedì 13 Gennaio 2009

- Sono dispositivi hardware che mettono a disposizione componenti logici più o meno complessi che possono essere connessi tra loro (**programmazione delle connessioni**) a seconda delle esigenze di progetto
- Dispongono di:
 - Componenti logici (Porte logiche, Flip-flop, Buffer...)
 - Linee di connessione
- Tipologie di dispositivi programmabili
 - ROM (Read-Only Memory), PLA (Programmable Logic Array), PAL (Programmable Array Logic): **dispositivi logici programmabili a 2 livelli**
 - il termine *2 livelli* indica che il dispositivo base è costituito da 1 sezione AND e da 1 sezione OR disgiunte
 - CPLD
 - FPGA



- **Read-Only Memory (ROM)**
 - Piano AND fissato.
 - Implementa **tutti** i possibili mintermini (decoder).
 - Piano di OR adattabile.
- **Programmable Logic Array (PLA)**
 - Piano AND programmabile.
 - È dato il numero di termini prodotto generabili, si programmano solo i mintermini/implicanti necessari.
 - Piano OR programmabile.
- **Programmable Array Logic (PAL)**
 - Piano AND programmabile.
 - È dato il numero di termini prodotto generabili, si programmano solo i mintermini/implicanti necessari.
 - Piano di OR fissato.
 - Ogni funzione (OR) può essere costruita solo con un certo sottoinsieme (cablato) di termini prodotto.



Logiche programmabili a 2 livelli: *campi di applicazione*



DIPARTIMENTO DI ELETTRONICA E INFORMAZIONE

- Reti combinatorie a due livelli non ottimizzate:
 - Read-Only Memory (ROM)
 - Anche PLA e PAL
- Reti combinatorie a due livelli ottimizzate:
 - Programmable Logic Array (PLA), Programmable Array Logic (PAL)
- Reti combinatorie multi livello costituite da reti a due livelli ottimizzate:
 - PLA e PAL con retroazione
- Macchine Sequenziali Sincrone con reti combinatorie multi livello costituite da reti a due livelli ottimizzate:
 - PLA e PAL con retroazione e registri

Read-Only Memory (ROM)

- Un Memoria a Sola Lettura (ROM) implementa la prima forma canonica di m funzioni di uscita a n ingressi
 - Somma di Prodotti (SOP)
- In una ROM, una configurazione di ingresso, denominata *indirizzo*, è associata una configurazione di uscita, denominata *parola*



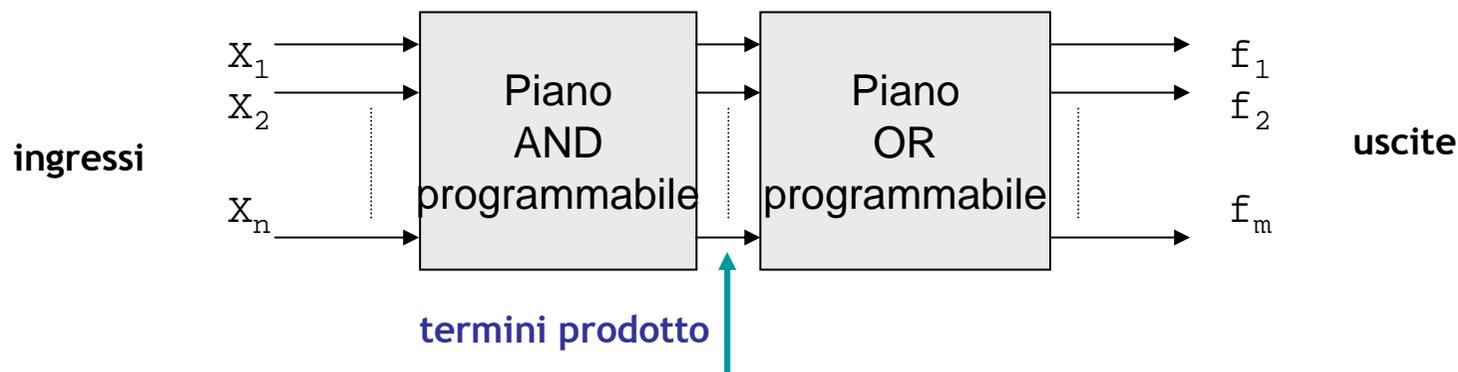


Programmable Logic Array (PLA)



DIPARTIMENTO DI ELETTRONICA E INFORMAZIONE

- Un Array Logico Programmabile (*Programmable Logic Array* - PLA) consente di implementare una somma di prodotti espressa in forma **minima** a due livelli (*somma di implicanti*)
 - Nota: l'estensione a forme non minime è naturale.
- In generale una PLA è definita da: **numero di ingressi** (n° variabili delle funzioni), **numero dei termini prodotto generabili**, **numero di uscite** (n° di funzioni realizzabili)



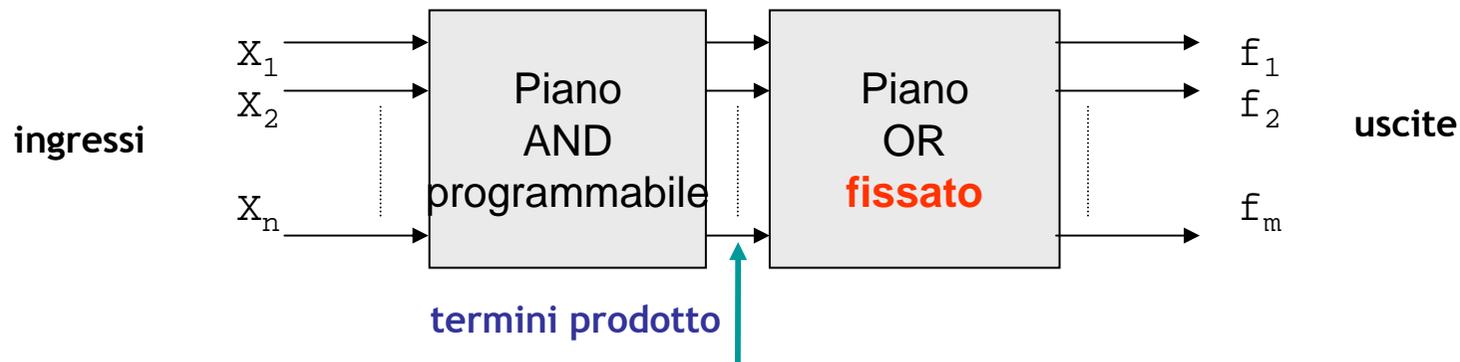


Programmable Array Logic (PAL)



DIPARTIMENTO DI ELETTRONICA E INFORMAZIONE

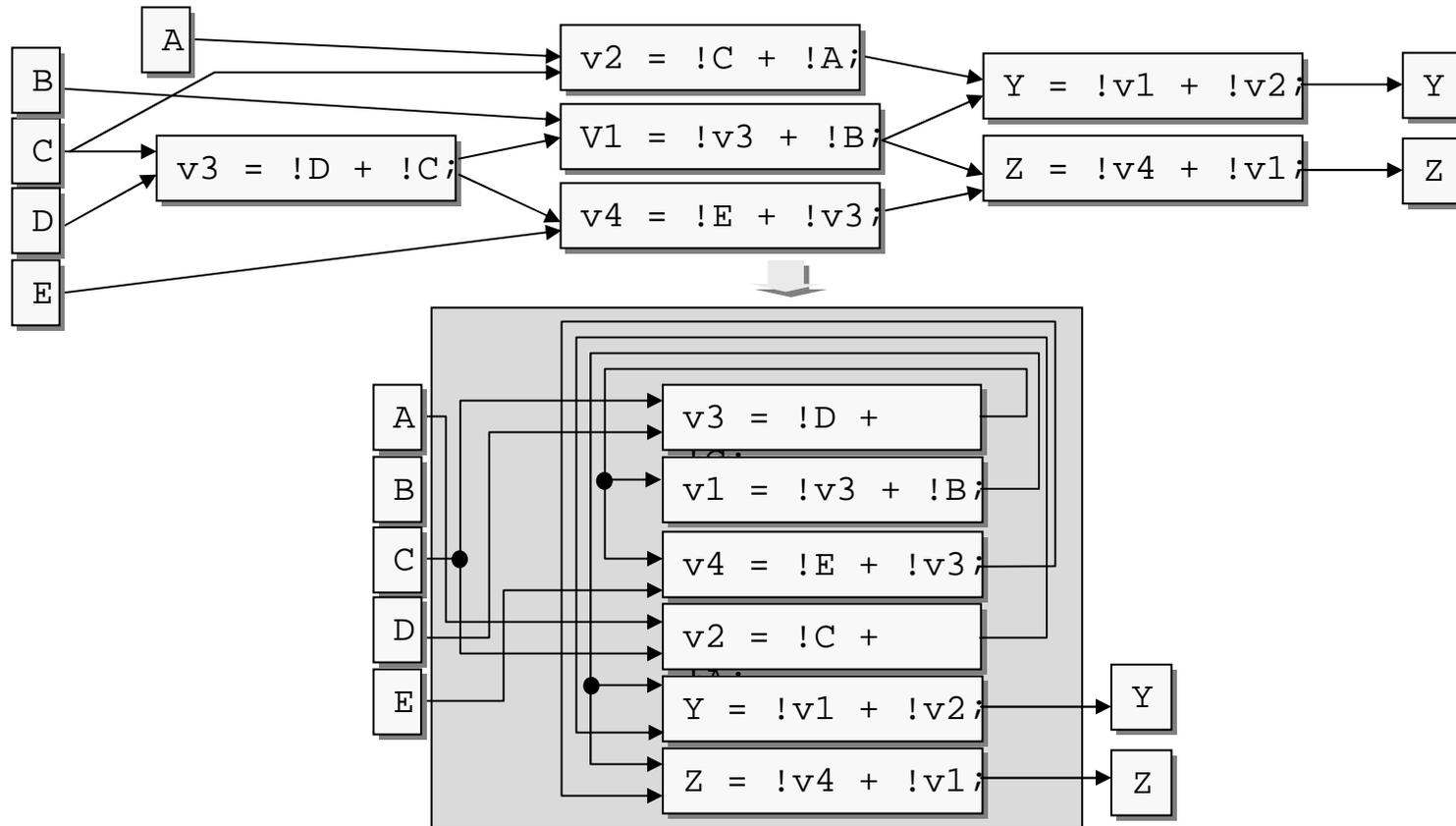
- Un Array Programmabile Logico (PAL) consente di implementare una somma di prodotti espressa in forma minima a due livelli (*somma di implicantì*)
 - PLA e PAL coprono lo stesso spazio d'applicazione.
- In generale una PAL è definita da: **numero di ingressi** (n° variabili delle funzioni), **numero dei termini prodotto** generabili, **numero di uscite** (n° di funzioni realizzabili)
 - Il piano OR fissato nasce da un trade-off tra il n° di ingressi per OR e il n° di OR



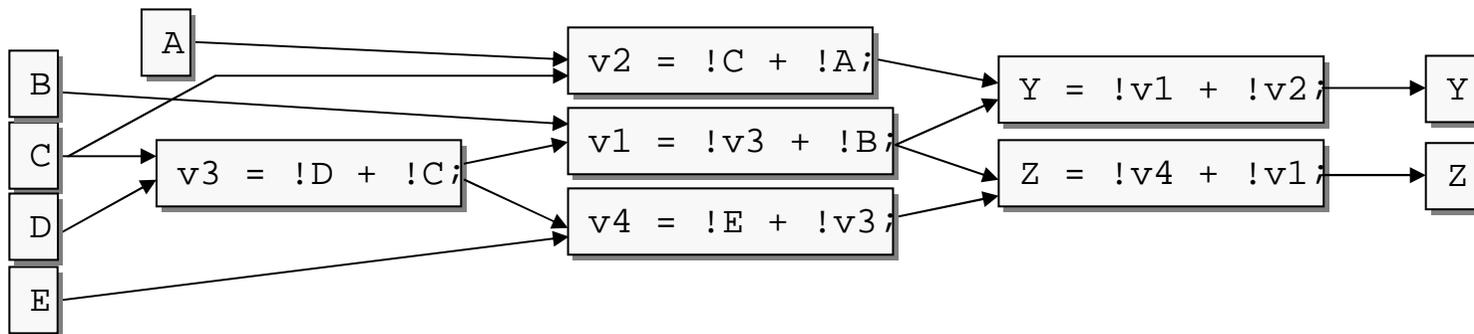
- Lo schema base mostrato consente di realizzare solo reti combinatorie a due livelli
- Questo limite è superato:
 - Introducendo delle linee di retroazione
 - Permette di implementare **reti combinatorie multi livello a più uscite**
 - Introducendo elementi di memoria (bistabili)
 - Permette di implementare **macchine sequenziali sincrone** in cui la parte combinatoria è costituita da una rete multi livello a più uscite



- Esempio di implementazione tramite PLA di una rete combinatoria multi livello a più uscite.



- Esempio di implementazione di una rete combinatoria multi livello a più uscite.



Funzioni da realizzare (**piano OR**)

$$v1 = !v3 + !B$$

$$v2 = !C + !A$$

$$v3 = !D + !C$$

$$v4 = !E + !v3$$

$$Y = !v1 + !v2 \text{ uscita}$$

$$Z = !v4 + !v1 \text{ uscita}$$

Termini prodotto da realizzare (**piano AND**)

$$p1 = !A$$

$$p2 = !B$$

$$p3 = !C$$

$$p4 = !D$$

$$p5 = !E$$

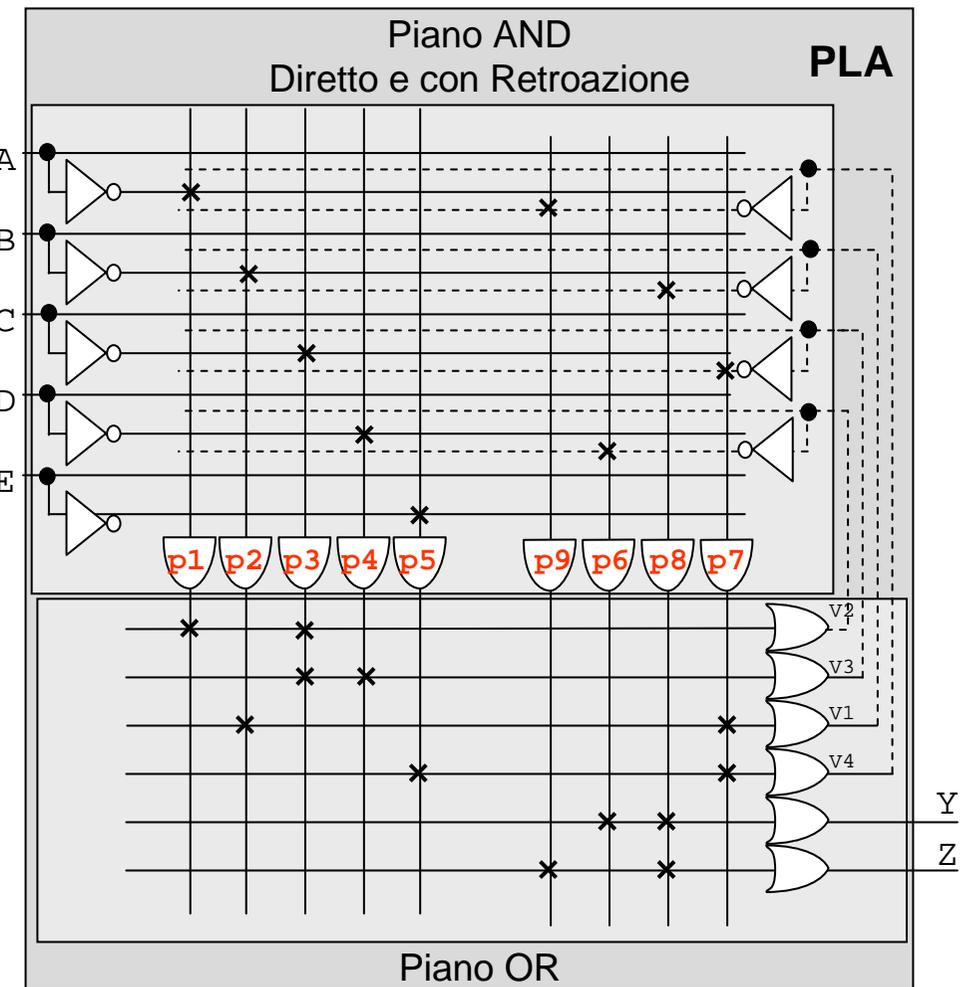
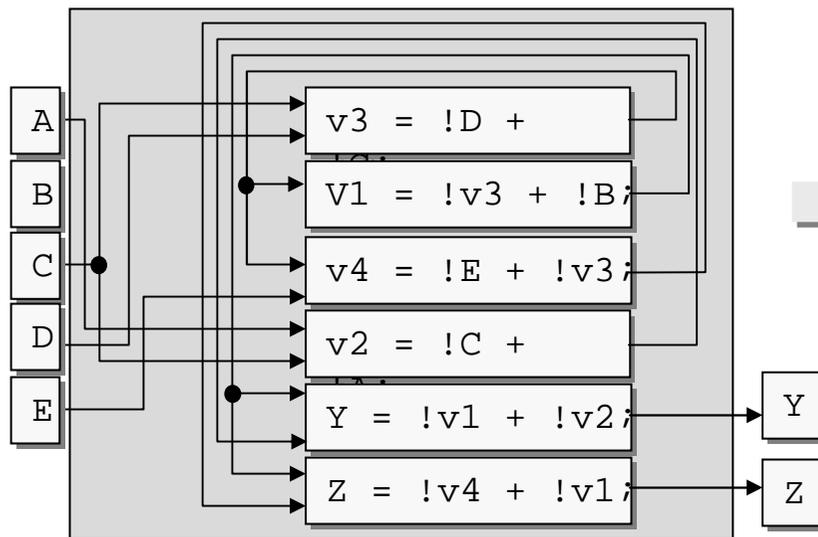
$$p6 = !v1$$

$$p7 = !v2$$

$$p8 = !v3$$

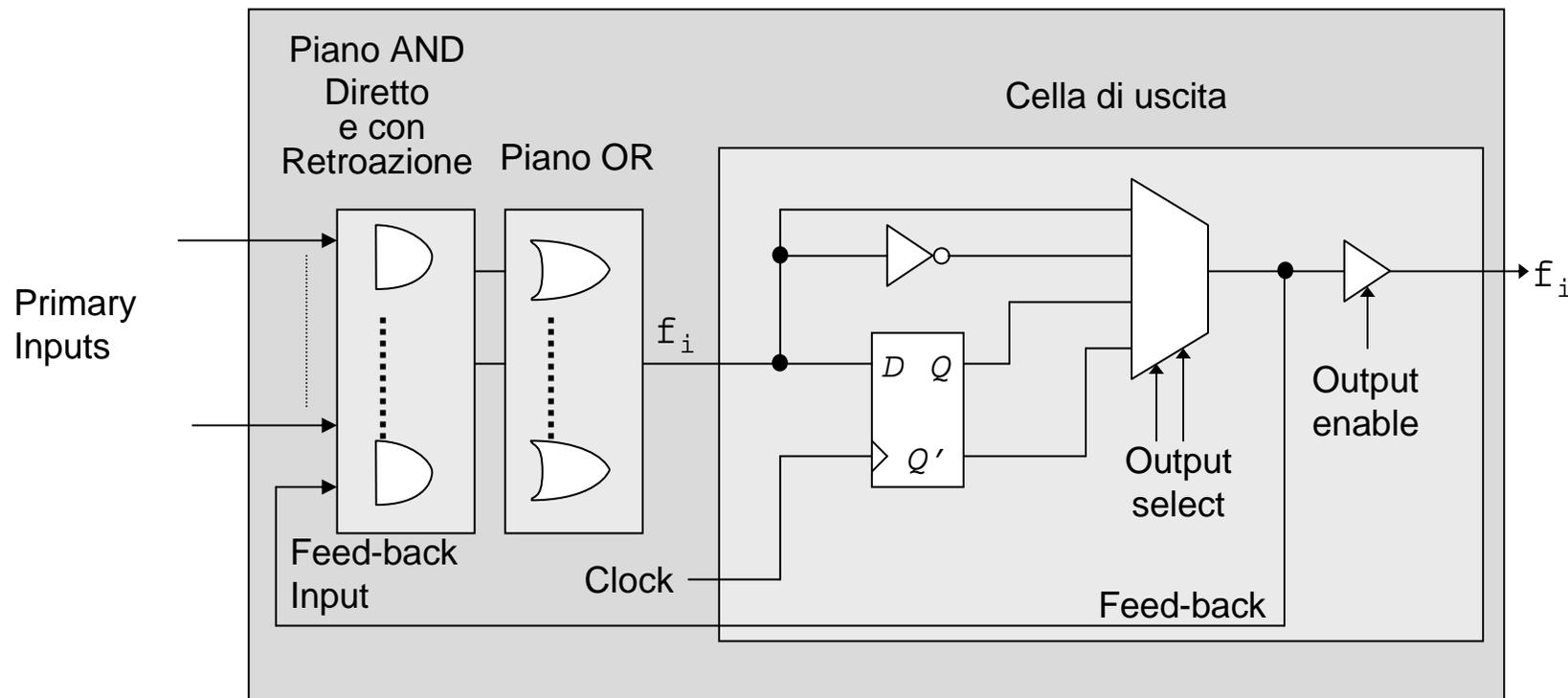
$$p9 = !v4$$

- Esempio (cont):





- Struttura logica dei dispositivi avanzati basati su PLA a PAL con **retroazione e bistabili** (per FSM)





- Realizzazione multi livello delle macchina a stati identificata dalle espressioni logiche sotto riportate (a, b ingressi primari, Y uscita della FSM, s_1, s_2 bit di stato):

$$Y = s_1 !ab + s_2 b + !s_2 \quad (\text{uscita } \lambda)$$

$$s_1^* = s_1 !a + s_2 + s_1 b \quad (\text{funzione di transizione } \delta_1)$$

$$s_2^* = s_1 !a !b + s_2 !b + !s_1 \quad (\text{funzione di transizione } \delta_2)$$

- Si raccoglie il fattore $g = s_1 !a + s_2$ ottenendo

$$Y = s_1 !ab + s_2 b + !s_2 = b(s_1 !a + s_2) + !s_2 = \mathbf{bg} + !s_2$$

$$s_1^* = s_1 !a + s_2 + s_1 b = (s_1 !a + s_2) + s_1 b = \mathbf{g} + s_1 b$$

$$s_2^* = s_1 !a !b + s_2 !b + !s_1 = !b(s_1 !a + s_2) + !s_1 = \mathbf{!bg} + !s_1$$

- Tale forma può essere realizzata grazie alla retroazione disponibile e agli elementi di memoria presenti.



- Le funzioni da realizzare, di transizione e d'uscita, sono:

$$g = s_1!a + s_2$$

$$Y = bg + !s_2$$

$$s_1^* = g + s_1b$$

$$s_2^* = !bg + !s_1$$

- La prima funzione è utilizzata in retroazione per realizzare le rimanenti
- La rete è a più livelli ed utilizza elementi di memoria FFD.
 - L'uso di FFD rende le funzioni di transizione uguali alle funzioni di eccitazione.
Stato presente: $Q_1 = s_1$; $Q_2 = s_2$ *Stato futuro:* $D_1 = s_1^*$; $D_2 = s_2^*$;

- Le funzioni da realizzare, di eccitazione e d'uscita, sono:

$$g = Q_1!a + Q_2$$

$$Y = bg + !Q_2$$

$$D_1 = g + Q_1b$$

$$D_2 = !bg + !Q_1$$



- Lo schema logico è:

Sezione AND

$$p1 = Q_1 !a$$

$$p2 = Q_2$$

$$p3 = bg$$

$$p4 = !Q_2$$

$$p5 = g$$

$$p6 = Q_1 b$$

$$p7 = !bg$$

$$p8 = !Q_1$$

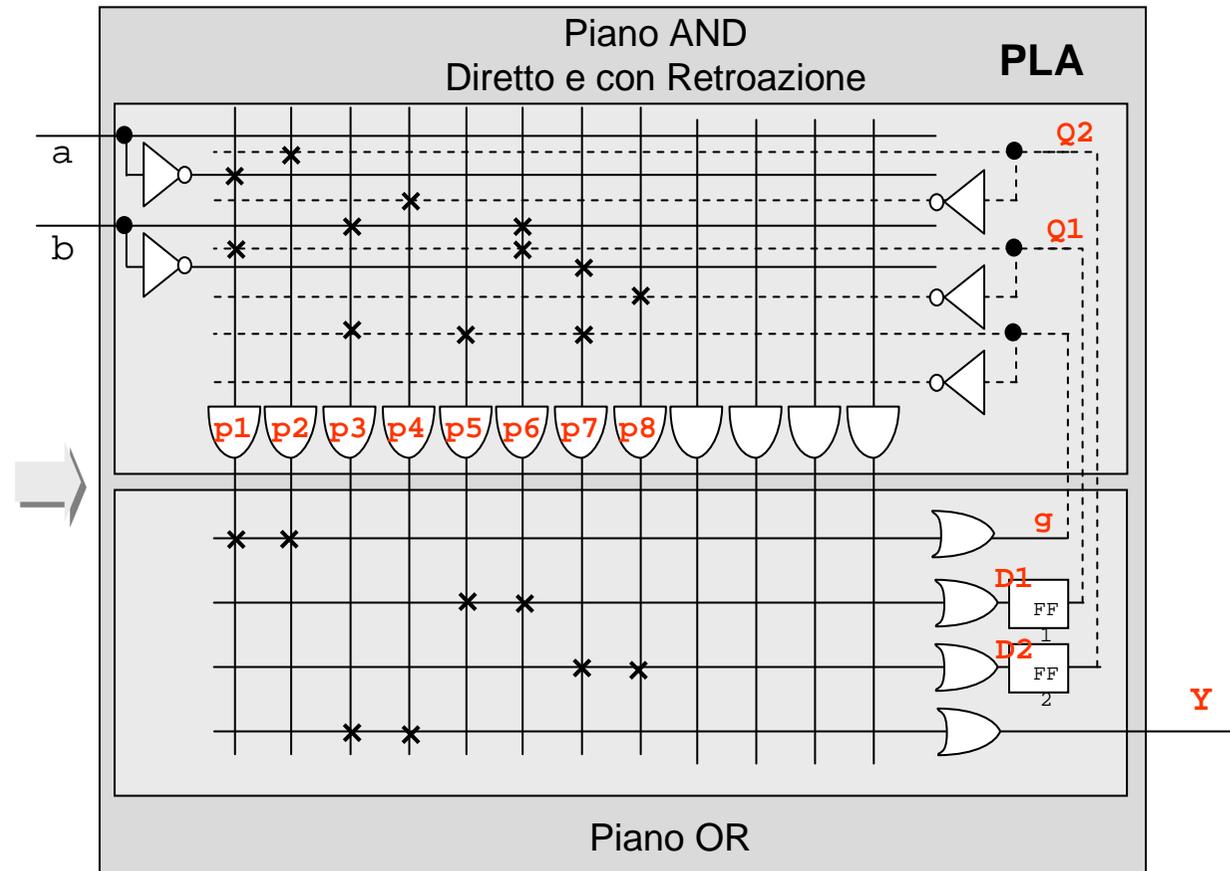
Sezione OR

$$g = Q_1 !a + Q_2$$

$$Y = bg + !Q_2$$

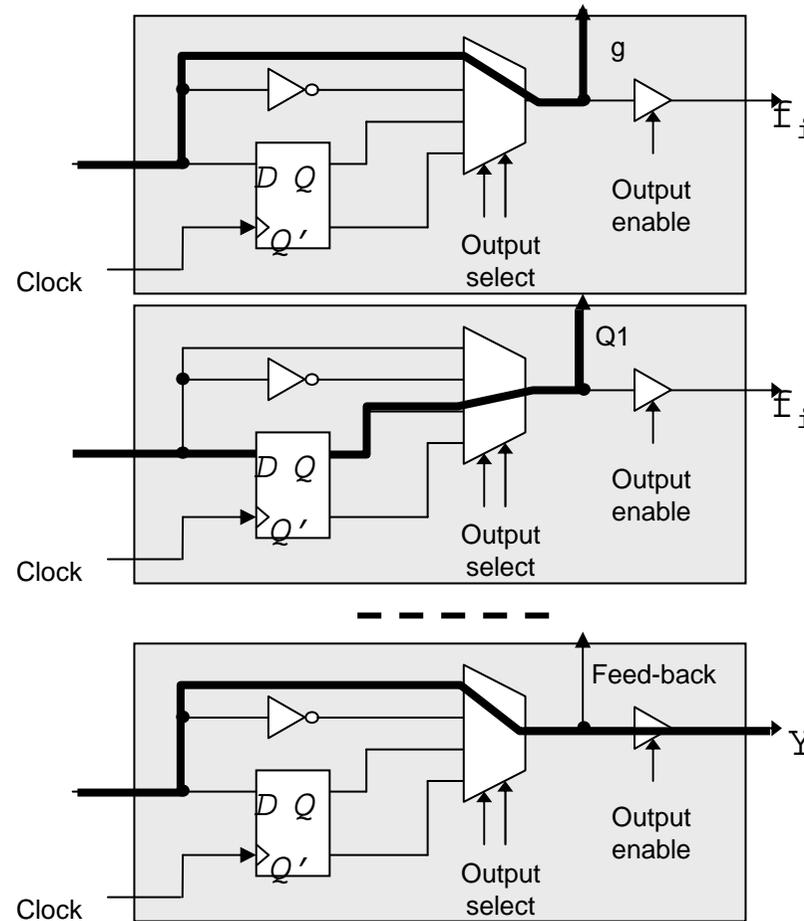
$$D_1 = g + Q_1 b$$

$$D_2 = !bg + !Q_1$$





- Dove le celle di uscita sono configurate nel modo seguente

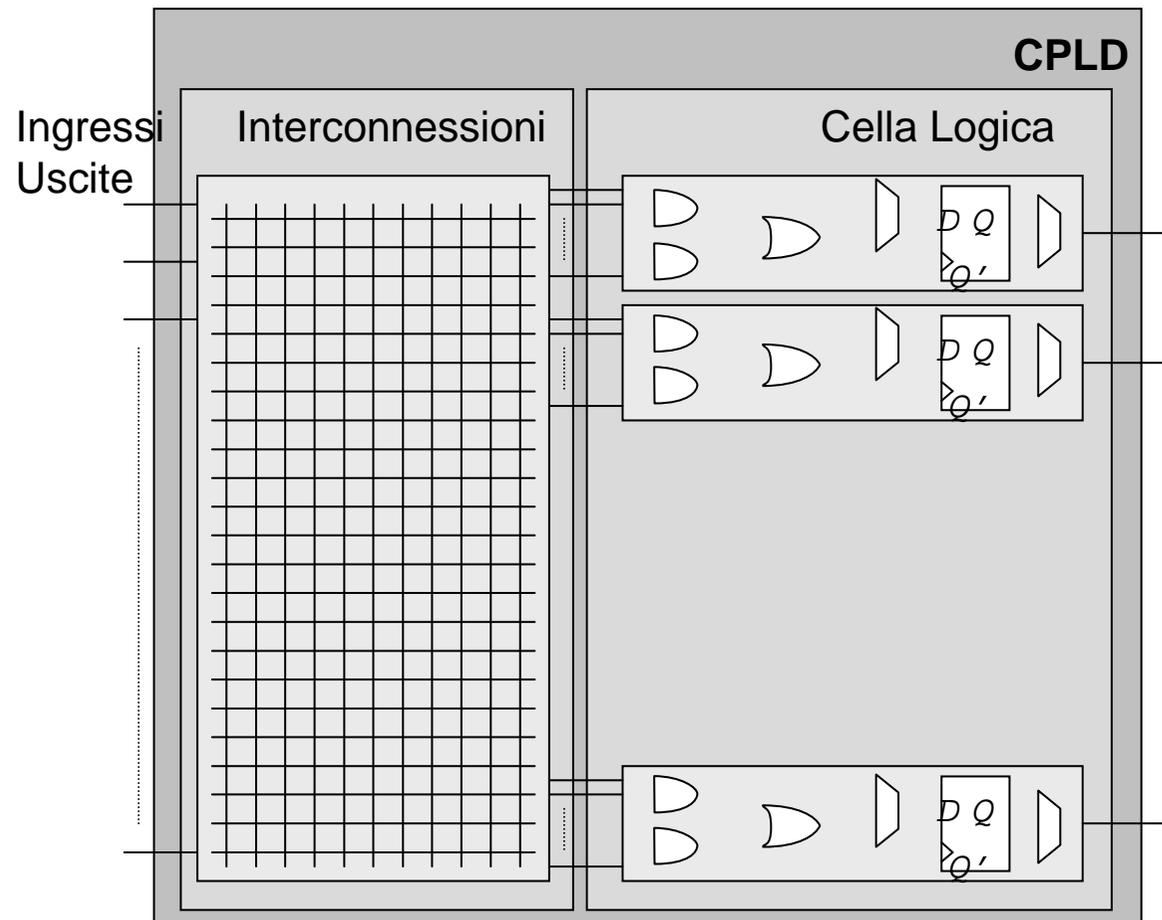




- I Dispositivi Logici Programmabili Complessi (*Complex Programmable Logic Devices - CPLD*) sono una evoluzione delle PLA e PAL
- Sono caratterizzati da:
 - Connessioni globali
 - Logica concentrata
- Rispetto a PAL e PLA:
 - Hanno dimensioni molto maggiori.
 - Sono costituite da celle più complesse.
 - Consentono di ottenere prestazioni più elevate.
 - Hanno una struttura più regolare e facilmente programmabile.



- L'architettura generale è la seguente





- Esempio: Cella logica della *Xilinx XC9500*

