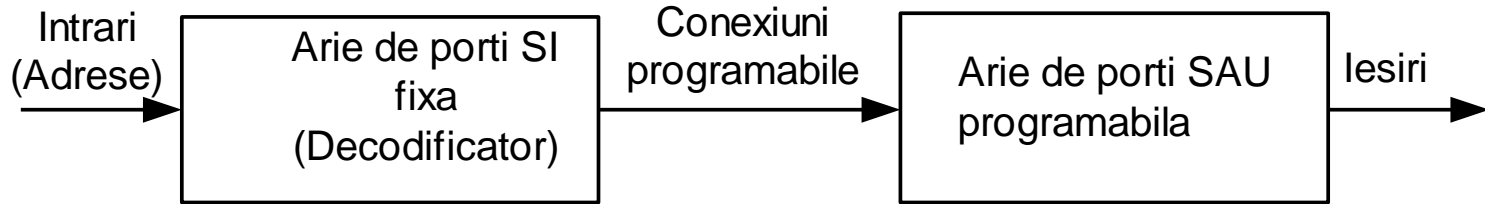


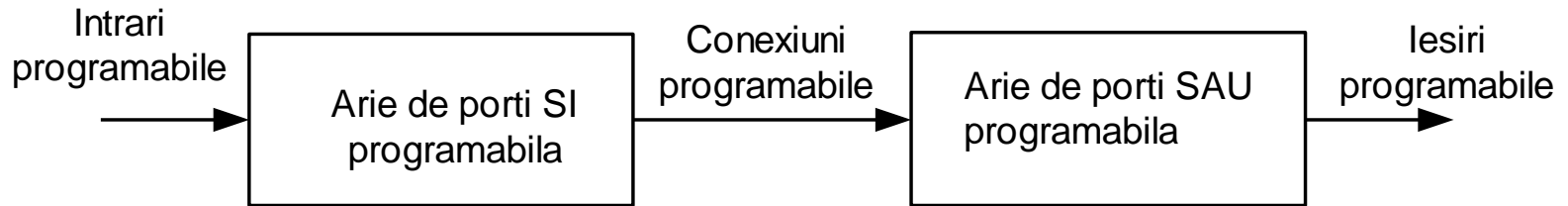
# Clasificarea PLD

- **SPLD** (Simple Programmable Logic Device)
  - PROM** (*Programmable ROM*)
  - PLA** (*Programmable Logic Array*)
  - PAL** (*Programmable Array Logic*)
- **CPLD** (Complex Programmable Logic Device)
- **FPGA** (Field Programmable Gate Array)

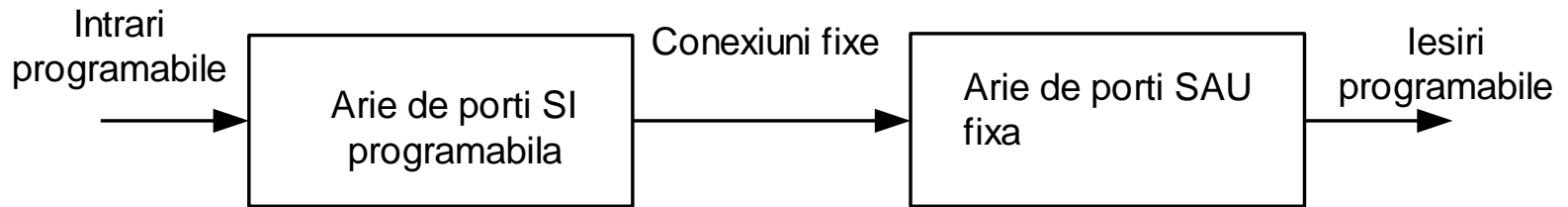
# Circuite logice programabile simple (SPLD)



Memorie PROM



Circuit PLA



Circuit PAL

# Memorii ROM

- **Memoria ROM** este un circuit combinațional care stochează permanent (memorie nevolatilă) informația binară, iar această informație poate fi numai citită.
- **Memoria PROM** (Programmable ROM) poate fi programată o singură dată. Metode de programare: prin măști (la etapa de fabricare) sau fuzibile (de către utilizator).

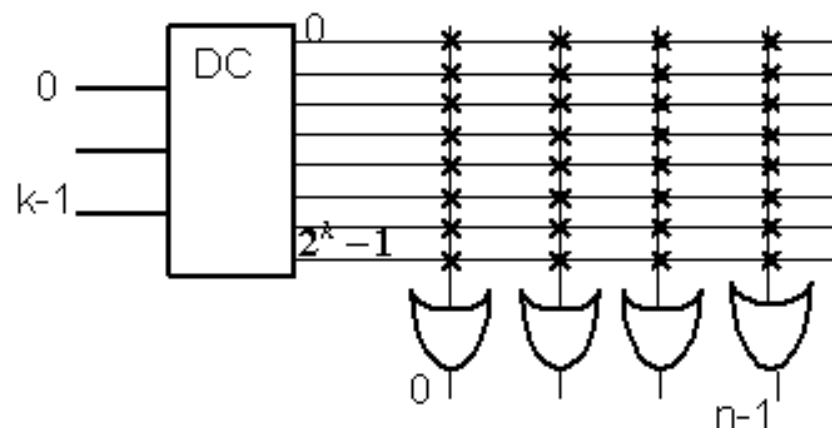
*Fuzibilul este o peliculă subțire de CrNi, care se vaporizează la trecerea unui curent suficient prin el. Programarea constă în selecția adresei și a liniei de date și aplicarea unui impuls de tensiune (10-30 V) pe un pin special de programare.*

Tehnologia de fabricare: bipolară sau MOS.

- **Memoria EPROM** (Erasable PROM) poate fi programată de mai multe ori, pentru că memoria poate fi „ștearsă” prin expunerea la raze ultraviolete. Matricea de memorie conține tranzistoare MOS cu poartă flotantă FAMOS (Floating Avalanche MOS).
- **Memoria EEPROM** (Electrically EPROM) poate fi ștearsă pe cale electrică.
- **Memoria FLASH** este o memorie EEPROM de capacitate mare .



O memorie ROM  $k \times n$  conține:  
 $k$  linii de intrare (adrese)  
 Decodificator (porți ȘI)  $k \rightarrow 2^k$   
 $n$  elemente SAU cu  $2^k$  intrări fiecare

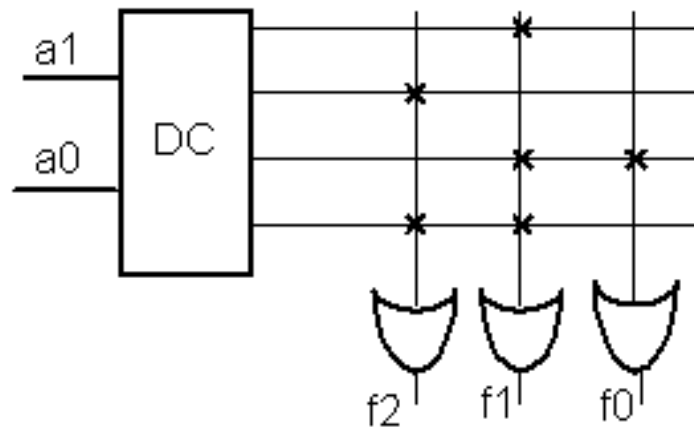


Decodificatorul este conectat la toate cele  $n$  porți SAU prin fuzibile. Astfel sunt  $2^k \times n$  conexiuni programabile.

Programarea ROM se face conform sumei canonice. Nu este necesară minimizarea.  
 Exemplu. Combinația de intrare din tabelul de adevăr se aplică la intrările decodificatorului, Valorile funcțiilor constituie conținutul memoriei.

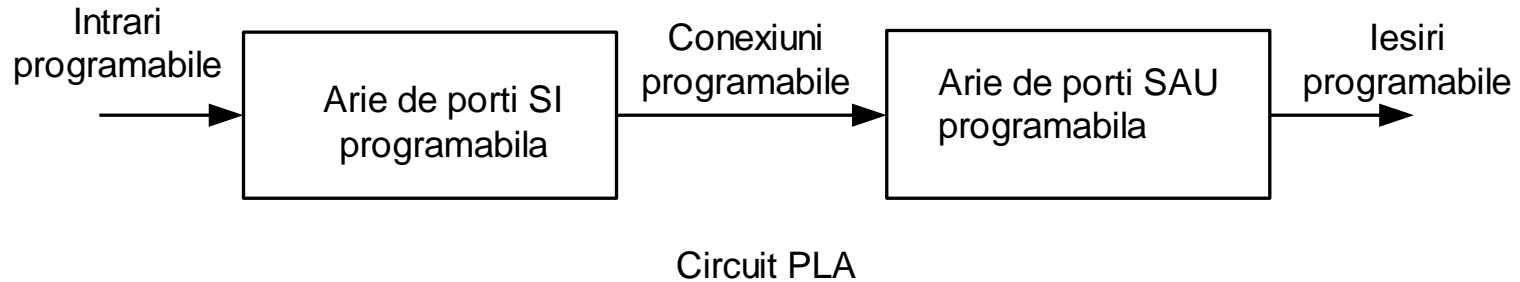
- 0 – fuzibilul se arde
- 1 – fuzibilul rămîne intact.

| a1 | a0 | f2 | f1 | f0 |
|----|----|----|----|----|
| 0  | 0  | 0  | 1  | 0  |
| 0  | 1  | 1  | 0  | 0  |
| 1  | 0  | 0  | 1  | 1  |
| 1  | 1  | 1  | 1  | 0  |



Dezavantajul implementării funcțiilor logice cu memorii ROM este creșterea foarte mare a capacității memoriei odată cu creșterea numărului de intrări în circuit datorită rigidității posibilităților de adresare.

# Circuite PLA



Dimensiunea unui dispozitiv PLA este dată de: numărul de intrări  $n$ , numărul de ieșiri  $m$  și numărul de termeni produs  $p$ . Numărul termenilor produs este mult mai mic decât numărul de mintermeni de  $n$  variabile ( $2^n$ ).

Porțile ȘI au câte  $2n$  intrări care pot fi conectate prin programare la oricare din cele  $n$  variabile de intrare, sub formă directă sau complementată.

Porțile SAU au câte  $p$  intrări care pot fi conectate prin programare la ieșirile oricărei porți ȘI.

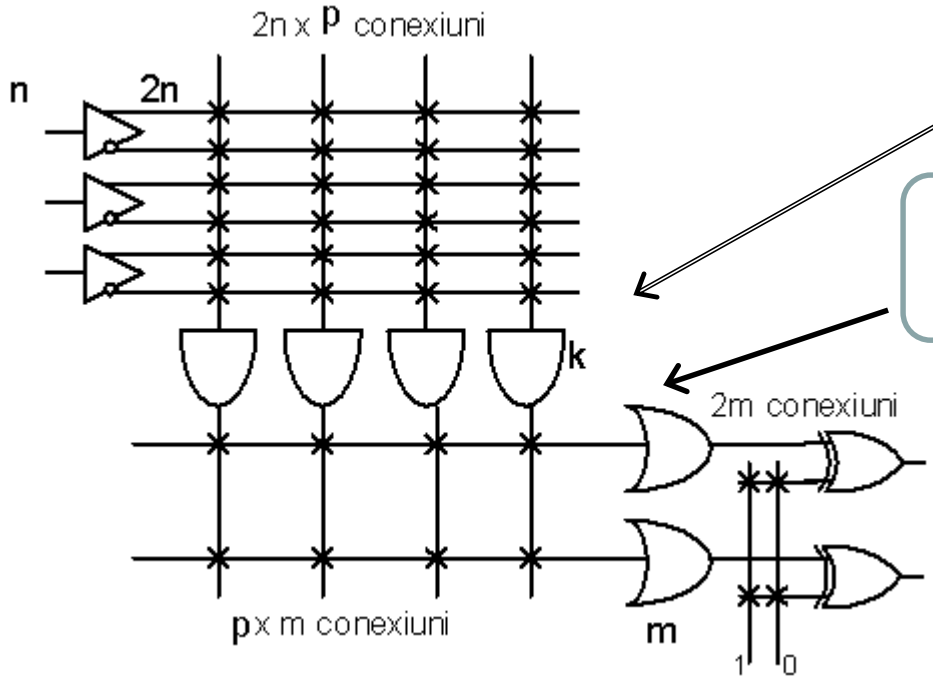
Ieșirile porților SAU sunt conectate la câte o poartă XOR cu o intrare programabilă, la care se aplică 0 pentru a obține funcția în formă directă, și 1 – pentru a obține funcția complementată.

Circuitul permite implementarea unui număr de  $m$  funcții logice, fiecare de câte  $n$  variabile, cu condiția că numărul termenilor produs să nu-l depășească pe  $p$ .

Astfel circuitul conține  $2n \times p + p \times m + 2m$  conexiuni programabile.

Dimensiunea unui dispozitiv PLA este dată de:  
 numărul de intrări **n**,  
 numărul de ieșiri **m**,  
 numărul de termeni produs **p**.  
 Numărul termenilor produs este mult mai mic decât  
 numărul de mintermeni de  $n$  variabile ( $2^n$ )

Porțile ȘI au câte **2n** intrări care pot fi conectate prin programare la oricare din cele **n** variabile de intrare, sub formă directă sau complementată.



Porțile SAU au câte **p** intrări care pot fi conectate prin programare la ieșirile oricărei porți ȘI.

Ieșirile porților SAU sunt conectate la câte o poartă XOR cu o intrare programabilă, la care se aplică 0 pentru a obține funcția în formă directă, și 1 – pentru a obține funcția complementată.

Circuitul permite implementarea unui număr de  $m$  funcții logice, fiecare de câte  $n$  variabile, cu condiția că numărul termenilor produs să nu-l depășească pe  $p$ .  
 Astfel circuitul conține  $2n \times p + p \times m + 2m$  conexiuni programabile.

Pentru implementarea funcțiilor logice în PLA este necesară minimizarea lor și elaborarea tabelului de programare. Tabelul de programare specifică termenii produs și termenii sumă pentru funcțiile care urmează a fi implementate.

Exemplu. Fie dat tabelul de adevăr pentru funcțiile F1 și F2

| A | B | C | F <sub>1</sub> | F <sub>2</sub> |
|---|---|---|----------------|----------------|
| 0 | 0 | 0 | 1              | 1              |
| 0 | 0 | 1 | 1              | 0              |
| 0 | 1 | 0 | 1              | 0              |
| 0 | 1 | 1 | 0              | 0              |
| 1 | 0 | 0 | 1              | 0              |
| 1 | 0 | 1 | 0              | 1              |
| 1 | 1 | 0 | 0              | 1              |
| 1 | 1 | 1 | 0              | 1              |

Se efectuează minimizarea funcțiilor.

|   |   |    |    |                             |
|---|---|----|----|-----------------------------|
|   |   | BC |    |                             |
|   | A | 00 | 01 | $\overbrace{11 \quad 10}^B$ |
| 0 | 1 | 1  | 1  | 0                           |
| 1 | 1 | 0  | 0  | 0                           |
|   |   | C  |    |                             |

$$F_1 = \overline{A}\overline{B} + \overline{A}\overline{C} + \overline{B}\overline{C}$$

$$\overline{F}_1 = AB + AC + BC$$

|   |   |    |    |                             |
|---|---|----|----|-----------------------------|
|   |   | BC |    |                             |
|   | A | 00 | 01 | $\overbrace{11 \quad 10}^B$ |
| 0 | 1 | 0  | 0  | 0                           |
| 1 | 0 | 1  | 1  | 1                           |
|   |   | C  |    |                             |

$$F_2 = AB + AC + \overline{A}\overline{B}\overline{C}$$

$$\overline{F}_2 = \overline{A}\overline{C} + \overline{A}\overline{B} + A\overline{B}\overline{C}$$



Analizând expresiile minimizate se aleg acelea care permit utilizarea termenilor comuni. Acestea sunt:

$$F_1' = AB + AC + BC \text{ or } F_1 = (AB + AC + BC)'$$

$$F_2 = AB + AC + A'B'C'$$

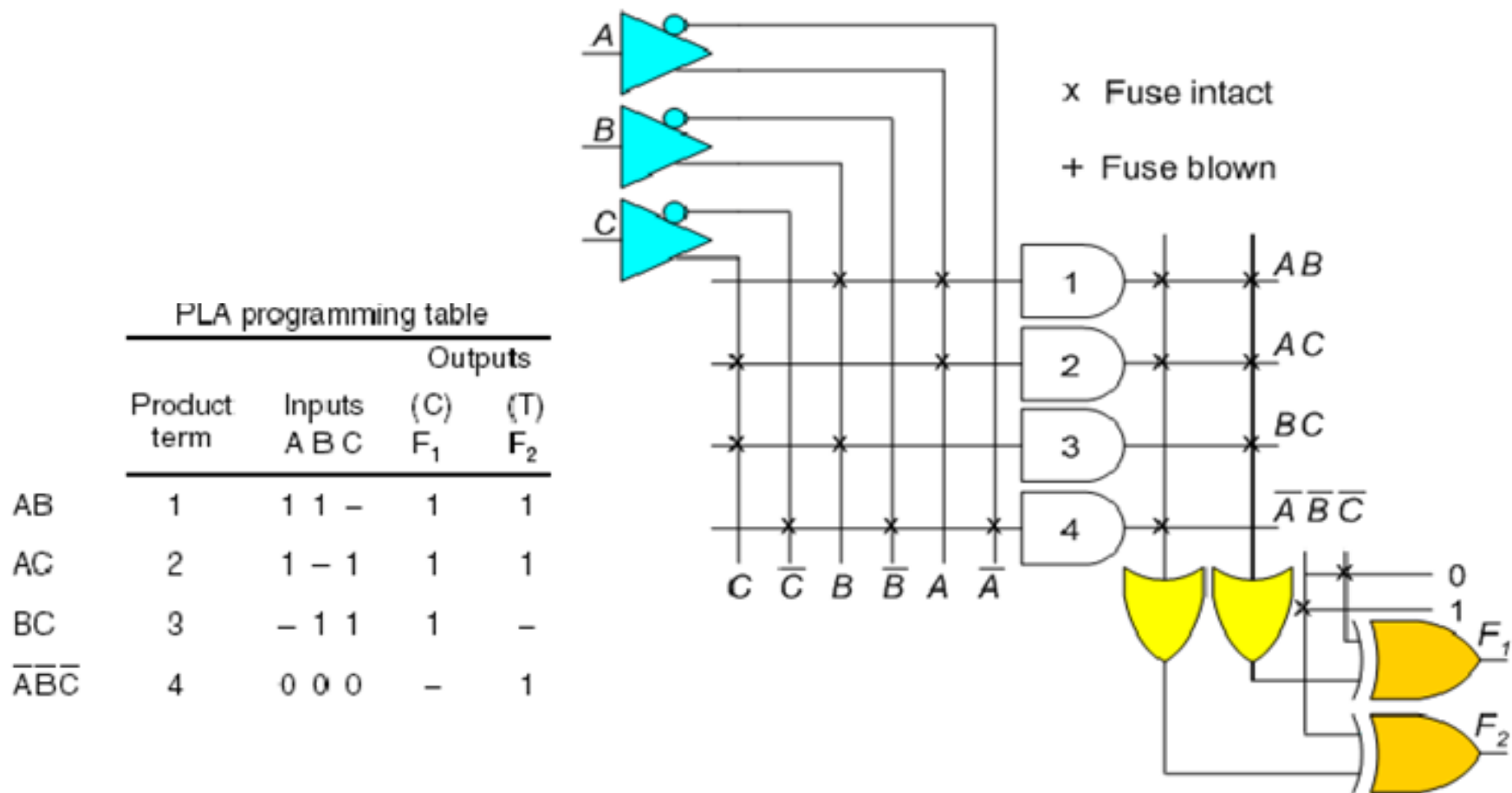
Astfel vom folosi doar 4 termeni produs :

$AB$ ,  $AC$ ,  $BC$ , and  $A'B'C'$ .

Tabelul de programare a PLA va arăta astfel:

| PLA programming table |                 |                 |                       |                       |  |
|-----------------------|-----------------|-----------------|-----------------------|-----------------------|--|
|                       |                 |                 | Outputs               |                       |  |
|                       | Product<br>term | Inputs<br>A B C | (C)<br>F <sub>1</sub> | (T)<br>F <sub>2</sub> |  |
| $AB$                  | 1               | 1 1 -           | 1                     | 1                     |  |
| $AC$                  | 2               | 1 - 1           | 1                     | 1                     |  |
| $BC$                  | 3               | - 1 1           | 1                     | -                     |  |
| $\overline{ABC}$      | 4               | 0 0 0           | -                     | 1                     |  |

## Circuitul PLA programat:

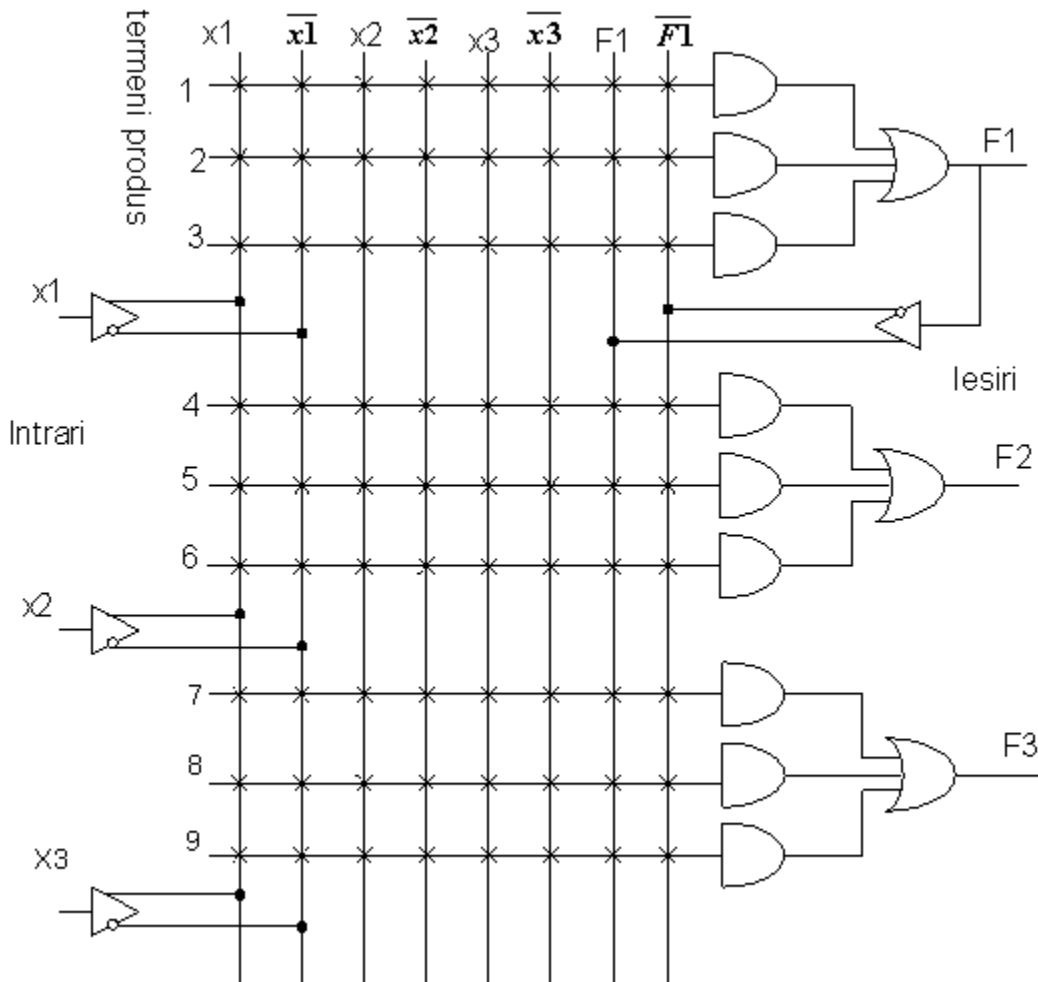
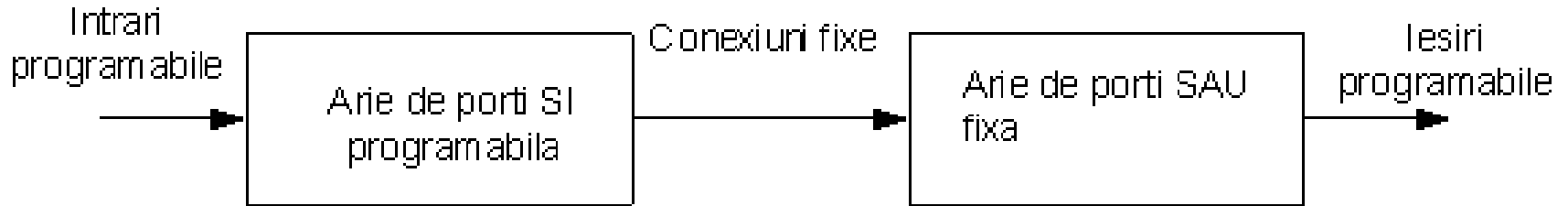


Cu toate că circuitele PLA sunt foarte flexibile, ele nu și-au găsit o aplicare practică mare, fiindcă este necesară programarea pe două nivele – ȘI și SAU.

Exemplu de dispozitiv PLA este Signetics 82S100, apărut la jumătatea anilor 70. Dispozitivul are 16 intrări, 48 porți ȘI și 8 ieșiri. Astfel el are  $2 \times 16 \times 48 = 1536$  conexiuni fuzibile în matricea de porți ȘI și  $8 \times 48 = 384$  în matricea de porți SAU.

Cu toate că circuitele PLA sunt foarte flexibile, ele nu și-au găsit o aplicare practică mare, fiindcă este necesară programarea pe două nivele – ȘI și SAU.

# Circuite PAL



Circuitele PAL sunt mai ușor programabile dar nu sunt atât de flexibile, ca circuitele PLA.

Circuitul conține 3 intrări și 3 ieșiri. Fiecare intrare e conectată la o poartă buffer (tampon)/inversor. Fiecare ieșire este generată de o poartă SAU fixă. Fiecare secțiune conține 3 porți ȘI programabile cu 6 intrări. Ieșirea  $F_1$  poate fi programată ca intrare la porțile ȘI.

Numărul porților ȘI în fiecare secțiune este fix, dar dacă funcția conține mai mulți termeni produs, există posibilitatea de a utiliza mai multe secțiuni.

Pentru implementarea funcțiilor logice în PAL este necesară minimizarea lor și elaborarea tabelului de programare. Tabelul de programare specifică doar termenii produs pentru funcțiile care urmează a fi implementate.

### Exemplu.

$$F_1 = \Sigma(1, 3, 4, 5, 7)$$

$$F_2 = \Sigma(1, 3, 4, 6, 7)$$

$$F_3 = \Sigma(1, 2, 3, 4, 5, 7)$$

F1

|    |       |    |    |    |    |
|----|-------|----|----|----|----|
|    | x1 x2 | 00 | 01 | 11 | 10 |
| x3 | 0     |    |    |    | 1  |
| 1  |       | 1  | 1  | 1  | 1  |

F2

|    |       |    |    |    |    |
|----|-------|----|----|----|----|
|    | x1 x2 | 00 | 01 | 11 | 10 |
| x3 | 0     |    |    | 1  | 1  |
| 1  |       | 1  | 1  | 1  |    |

F3

|    |       |    |    |    |    |
|----|-------|----|----|----|----|
|    | x1 x2 | 00 | 01 | 11 | 10 |
| x3 | 0     |    | 1  |    | 1  |
| 1  |       | 1  | 1  | 1  | 1  |

$$F_1 = x_1 \bar{x}_2 + x_3$$

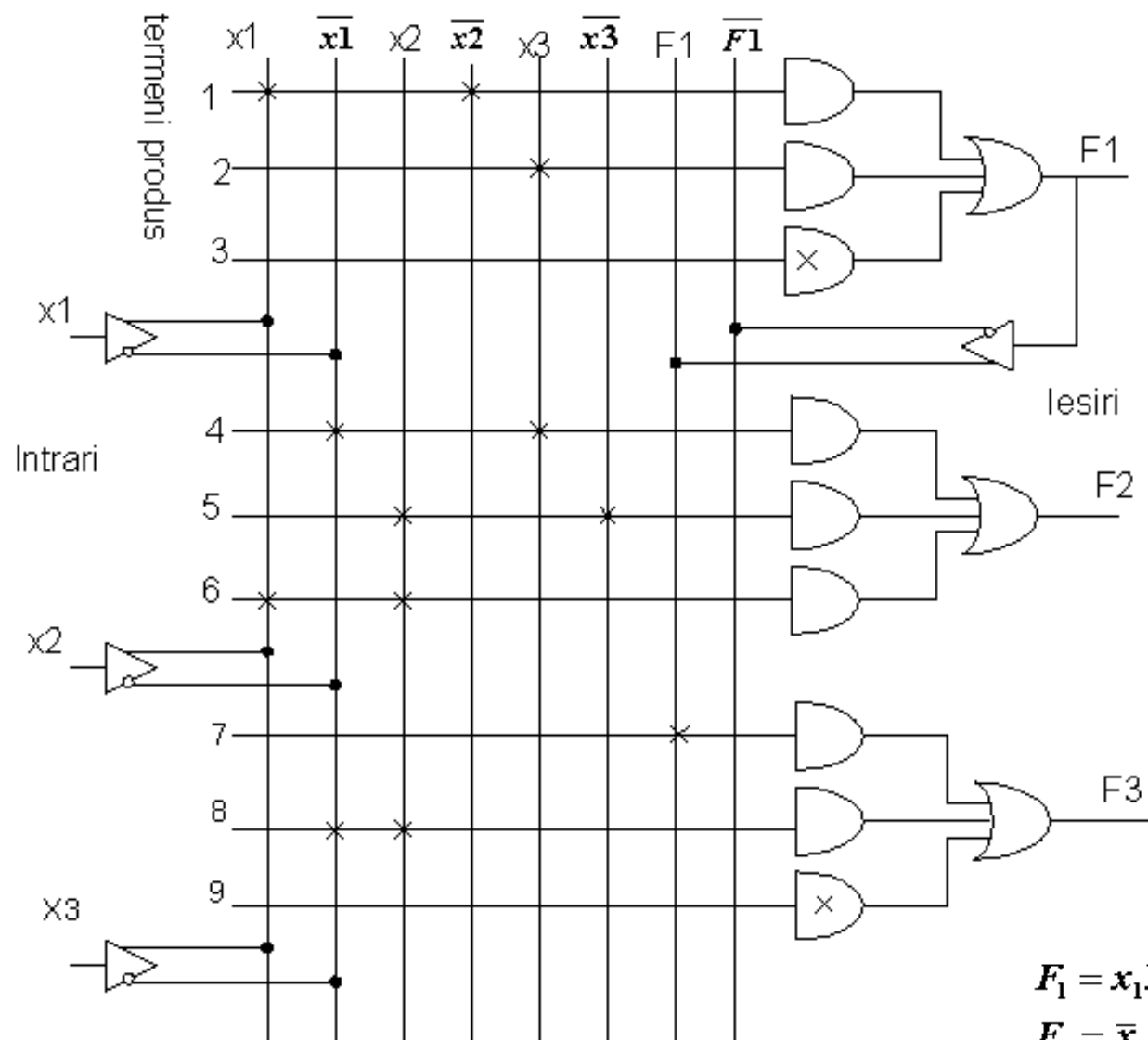
$$F_2 = \bar{x}_1 x_3 + x_2 \bar{x}_3 + x_1 x_2$$

$$F_3 = x_1 \bar{x}_2 + x_3 + \bar{x}_1 x_2 = F_1 + \bar{x}_1 x_2$$

Tabelul de programare a PLA va arăta astfel:

| Termeni produs | Intrari pentru porti SI |    |    |    | Iesiri  |
|----------------|-------------------------|----|----|----|---|
|                | x1                      | x2 | x3 | F1 |   |
| 1              | 1                       | 0  | -  | -  | $F_1 = x_1 \bar{x}_2 + x_3$                     |
| 2              | -                       | -  | 1  | -  |   |
| 3              | -                       | -  | -  | -  |   |
| 4              | 0                       | -  | 1  | -  | $F_2 = \bar{x}_1 x_3 + x_2 \bar{x}_3 + x_1 x_2$ |
| 5              | -                       | 1  | 0  | -  |   |
| 6              | 1                       | 1  | -  | -  |   |
| 7              | -                       | -  | -  | 1  | $F_3 = F_1 + \bar{x}_1 x_2$                     |
| 8              | 0                       | 1  | -  | -  |   |
| 9              | -                       | -  | -  | -  |   |

## Circuitul PAL programat:

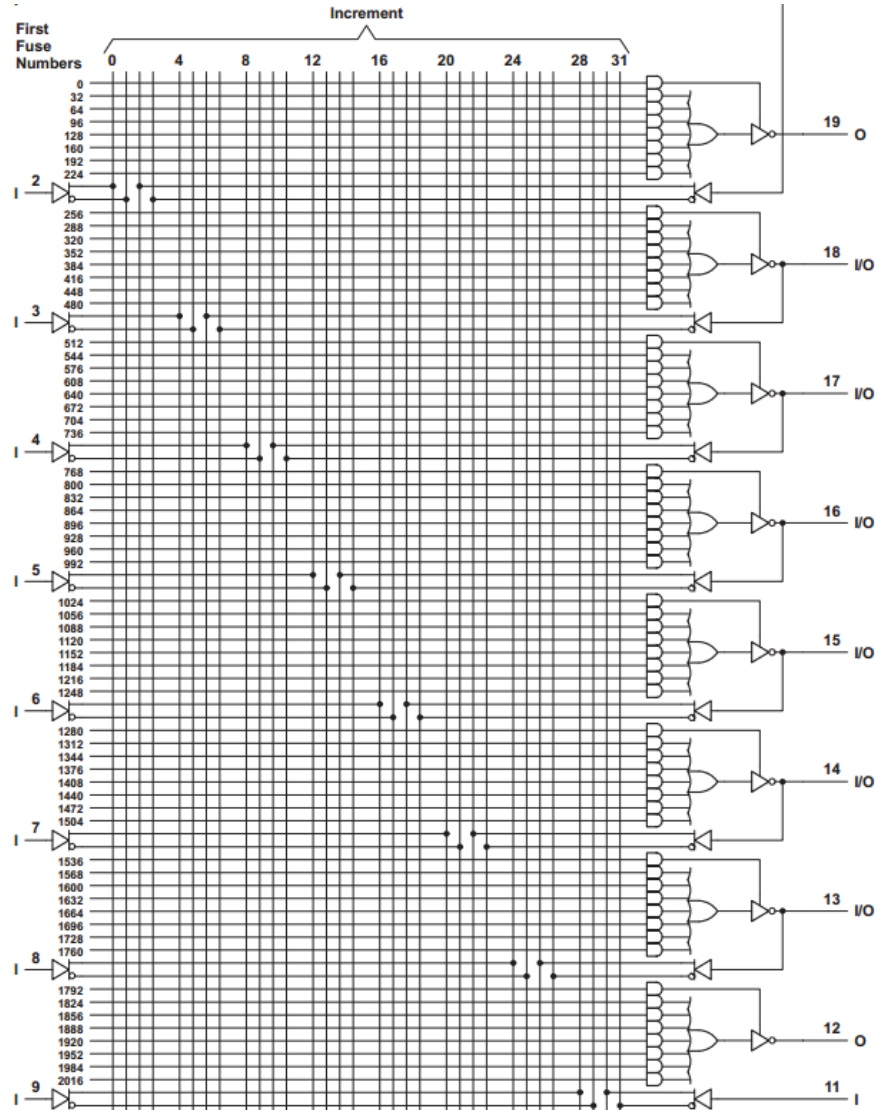
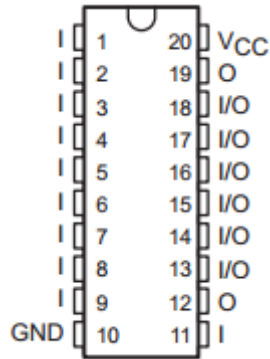


$$F_1 = x_1 \bar{x}_2 + x_3$$

$$F_2 = \bar{x}_1 x_3 + x_2 \bar{x}_3 + x_1 x_2$$

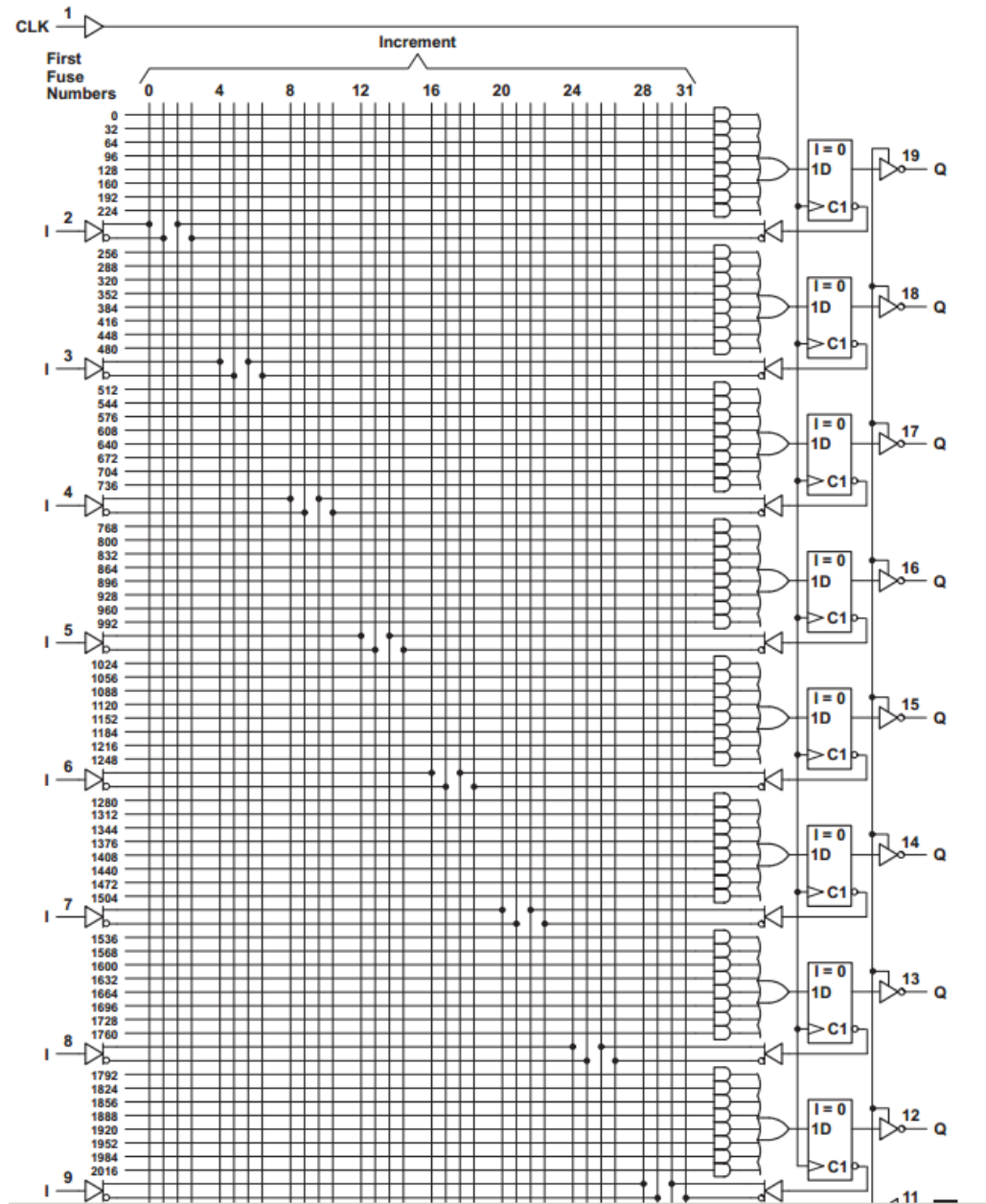
$$F_3 = x_1 \bar{x}_2 + x_3 + \bar{x}_1 x_2 = F_1 + \bar{x}_1 x_2$$

În prezent una din cele mai utilizate structuri de PLD combinaționale este **PAL16L8**, compania **AMD (Advanced Micro Devices)**. Dispozitivul are **16 intrări**, **64 de porți ȘI**, divizate în 8 secțiuni, **8 porți SAU** pentru generarea celor 8 ieșiri. Fiecare poartă ȘI are 32 intrări pentru variabilele de intrare în formă directă și inversă.



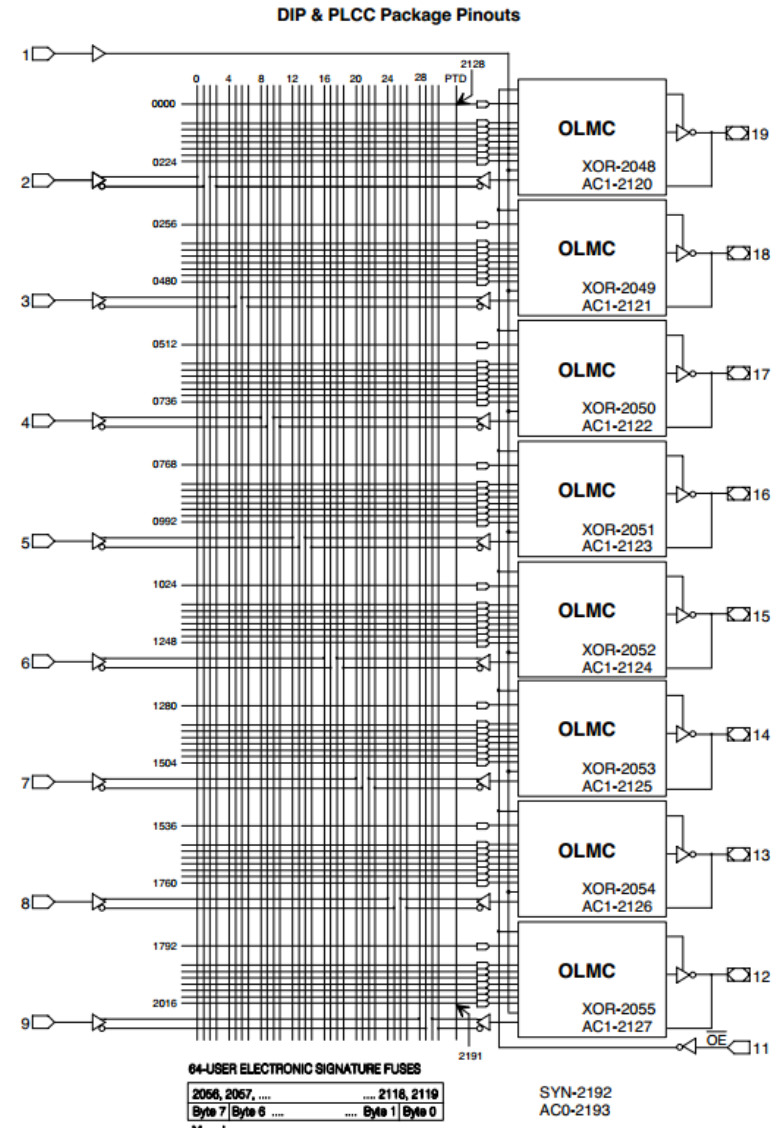
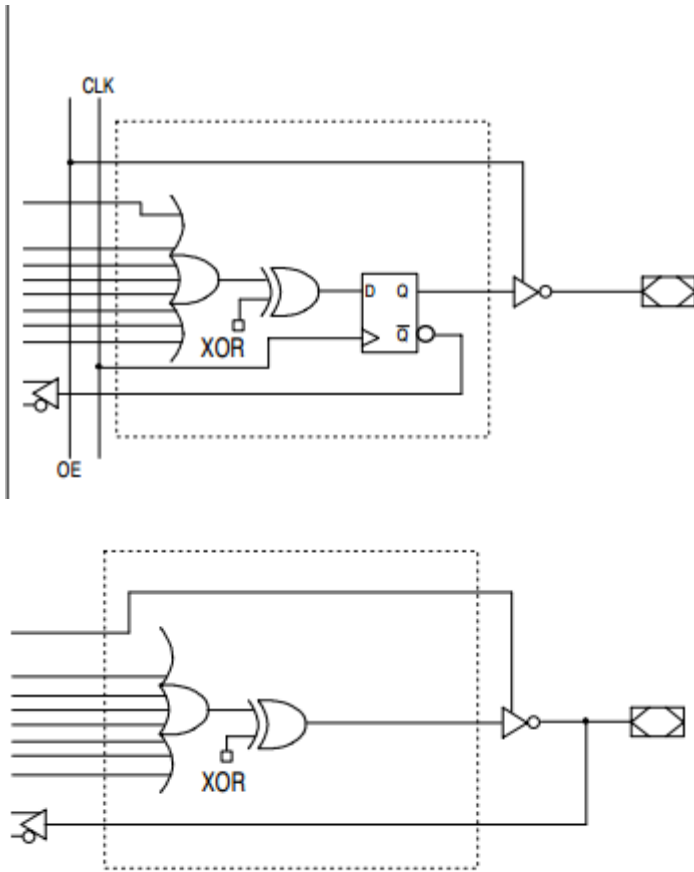


Circuitele PAL secvențiale  
conțin bistabile de tip D (ex.  
PAL16R8)



Circuitul **GAL16V8** cu ștergere electrică – Generic Array Logic, firma Lattice Semiconductor conține macrocelule Output Logic Macrocell (OLMC).

O macrocelă este formată din bistabile care pot funcționa în regimurile D și T și câteva multiplexoare pentru alegerea regimului (combinațional, secvențial, ieșire directă sau inversă).



Avantajele oferite de circuitele PLD sunt: consumul redus de putere, performanțe mai bune datorită lungimi mult reduse a interconexiunilor și o fiabilitate mai ridicată.

În prezent circuitele SPLD se utilizează ca componente de rețea care cer performanțe ridicate per ansamblu: hub-uri de rețea bridge-uri, routere, produse din zona telefoniei mobile, video game-urilor și a web browserelor.

Producători: **AMD,**  
**Philips Semiconductors,**  
**Lattice Semiconductor.**