

**DISPOSITIVOS**  
**LÓGICOS**  
**PROGRAMABLES**

# Dispositivos lógicos programables

## Tipos de PLDs

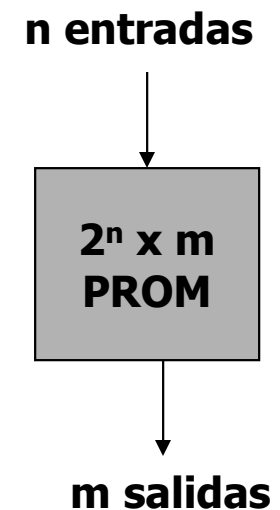
- Read Only Memory (ROM)
- Programmable ROM (PROM, etc.)
- Programmable Logic Arrays (PLA)
- Programmable Array Logic (PAL)
- Erasable PLD (EPLD)
- Field Programmable Gate Arrays (FPGA)

## Dispositivos lógicos programables

- **Arreglos lógicos sencillos**
  - Basados en circuitos lógicos de 2 niveles (AND/OR)
  - Tienen una estructura de arreglo regular
  - Distintos tipos
    - Memorias de solo lectura, Read Only Memories (ROMs, PROMs, etc.)
    - Programmable Logic Array (PLA)
    - Programmable Array Logic (PAL)
- **Arreglos programables in situ Field Programmable Gate Arrays (FPGA)**
  - Bloque básico común replicado muchas veces
  - Cada bloque puede configurarse para producir distintas funciones lógicas y en general incluyen biestables (flip-flops)
  - Interconexiones programables
  - las FPGAs más grandes tienen alrededor de 500K compuertas además de 500 Kb de SRAM

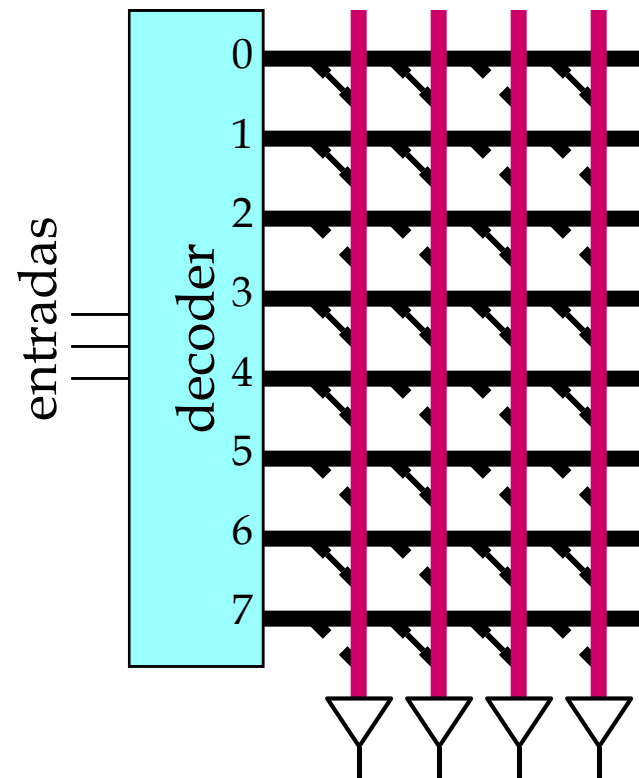
## Memoria de solo lectura, Read-Only Memory (ROM)

- Un decodificador genera para las  $n$  variables de entradas  $2^n$  **minitérminos**.
- Utilizando compuertas OR para unir los minitérminos de las funciones Booleanas, se puede generar cualquier circuito combinatorio.
- Las memorias ROM son dispositivos que incluyen el decodificador (compuertas AND) y las compuertas OR en un solo circuito integrado.
- Un depósito de información binaria fija
- Los CI ROM y PROM tienen uniones internas que pueden quemarse o romperse ("programando")
- Consisten en  $n$  líneas de entrada y  $m$  de salida
- Cada combinación de entrada se llama dirección
- Cada combinación de salida es una palabra
- La cantidad de bits por palabra es  $m$
- La cantidad de direcciones posibles es  $2^n$
- Una palabra de salida es seleccionada por una dirección única



## Construcción de una memoria ROM / PROM

- Las ROM y PROM pueden construirse utilizando arreglos ortogonales de cables.
  - Conexión opcional en cada intersección.
  - El decodificador coloca un 1 lógico en solo uno de los cables horizontales, que se puede detectar a la salida si hay una conexión realizada.
- Algunas PROMs se pueden configurar quebrando conexiones.
  - Alta tensión ubicada entre una entrada y una salida por vez.
  - La gran corriente causa que el fusible en la interconexión se queme.
- Otras PROMs pueden borrarse y reprogramarse (EPROMs, EEPROMs, Flash).



- **Funciones almacenadas**  
 $\Sigma(0,1,3,4,6)$ ,  $\Sigma(0,1,3,5,7)$ ,  
 $\Sigma(2,3,6,7)$ ,  $\Sigma(0,3,4,6)$

# Read-Only Memory (ROM / PROM)

Ejemplo: ROM 4x4 con compuertas AND-OR

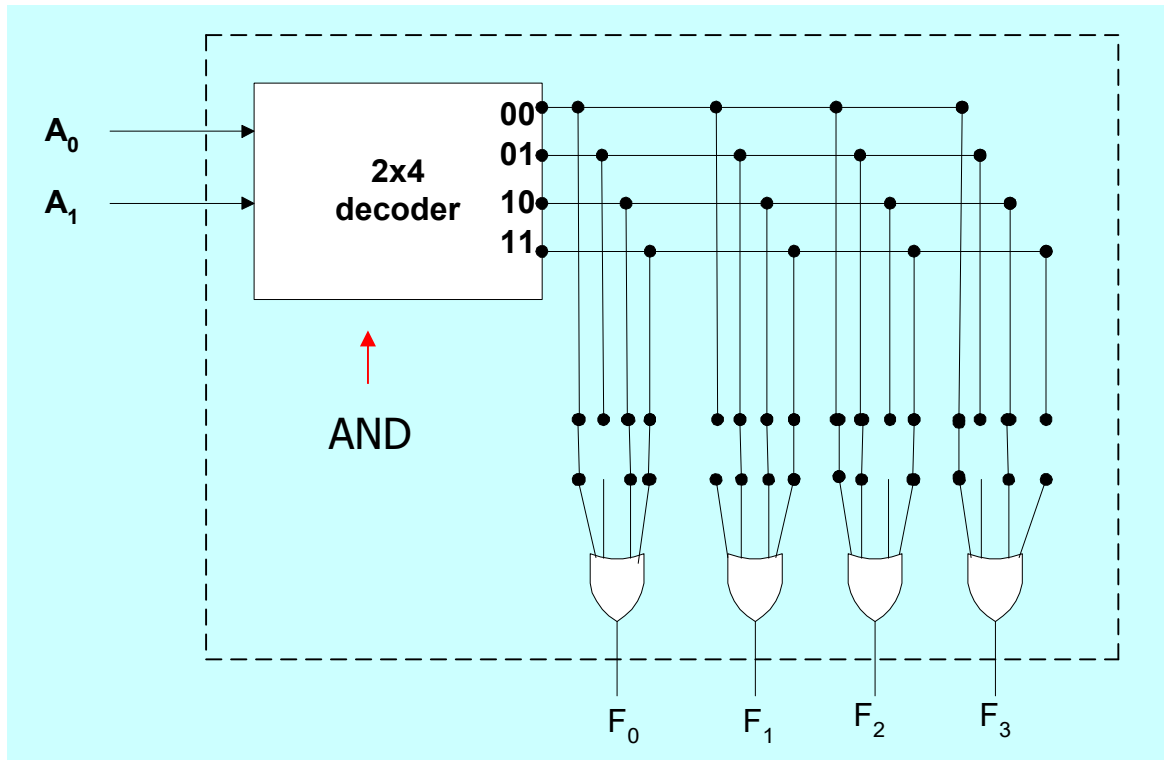
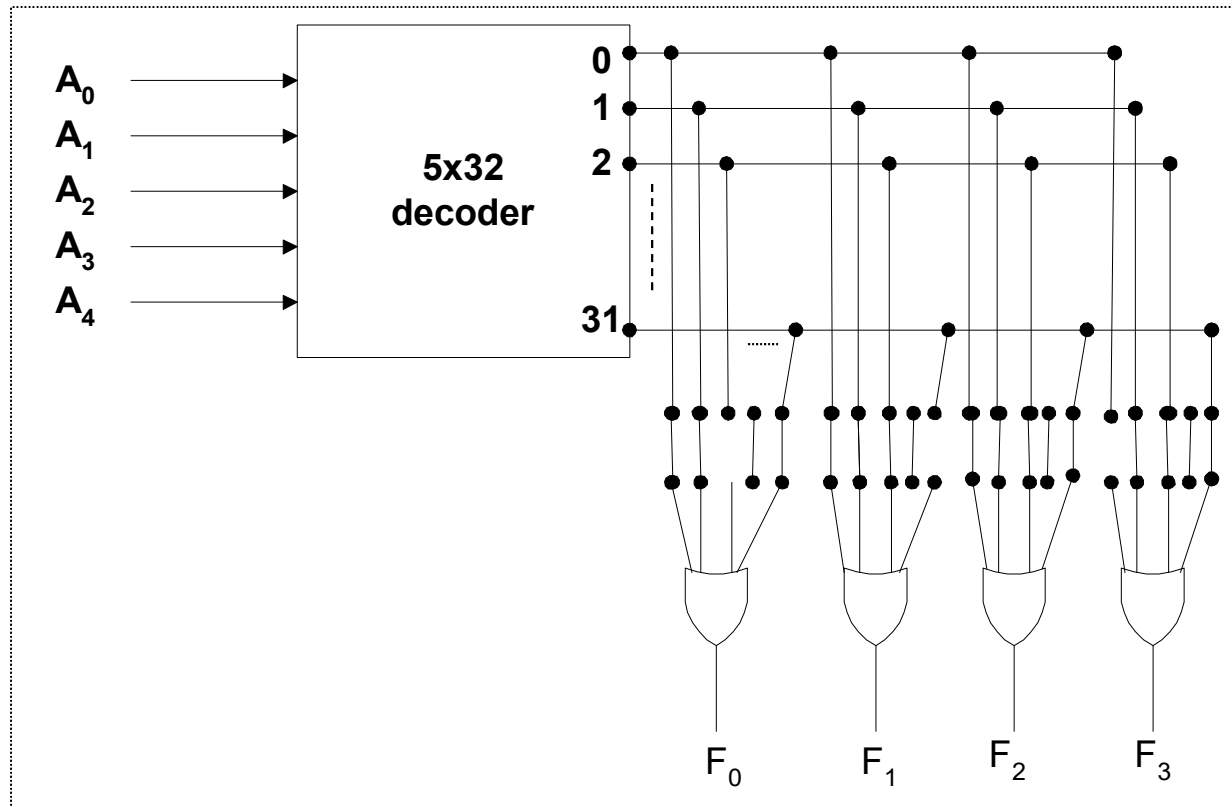


Tabla de verdad

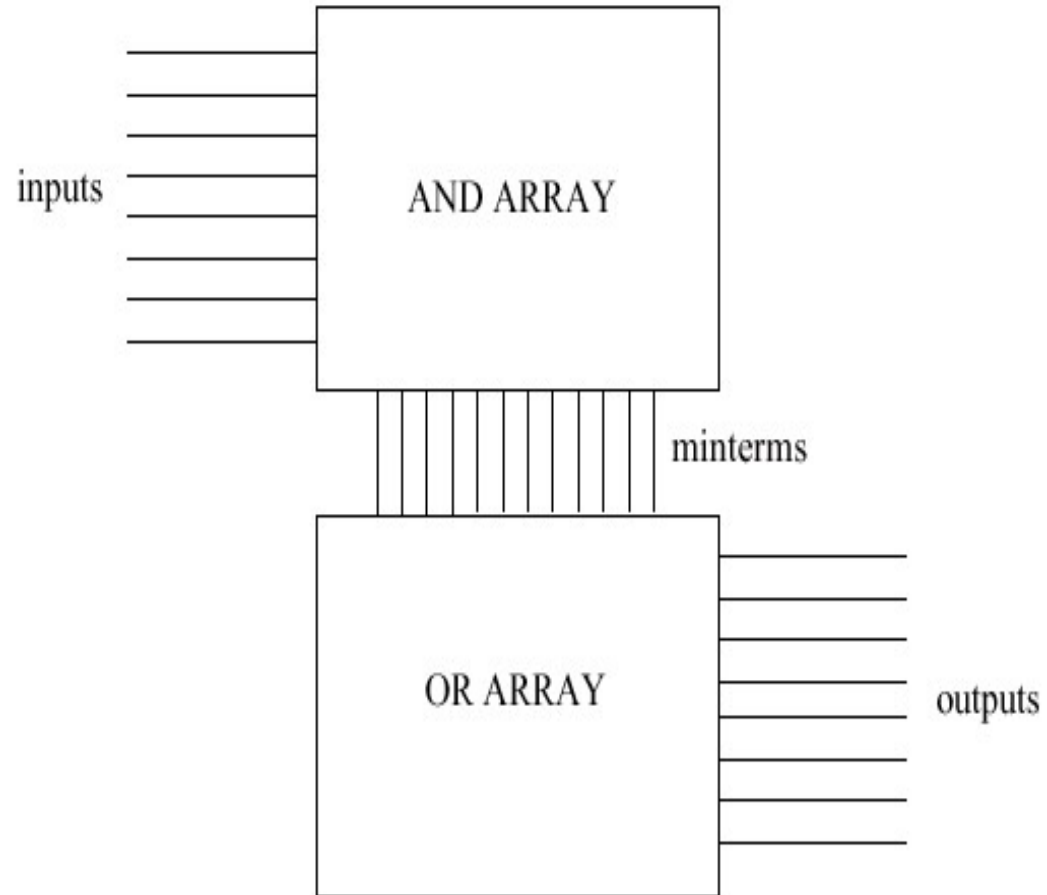
$A_0$	$A_1$	$F_0$	$F_1$	$F_2$	$F_3$
0	0	1	0	1	1
0	1	0	1	1	0
1	0	1	1	0	1
1	1	1	1	1	0

# Read-Only Memory (ROM / PROM)

Ejemplo: ROM 32x4



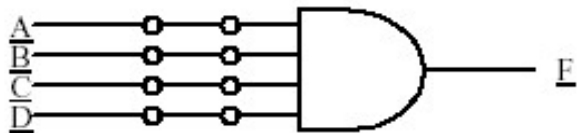
# Read-Only Memory (ROM / PROM)



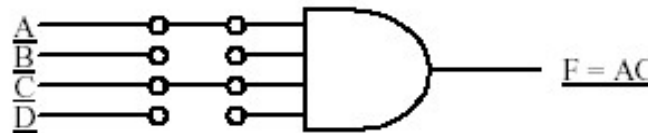


# Programmable Logic Array (PLA)

PLA: Notación de programación



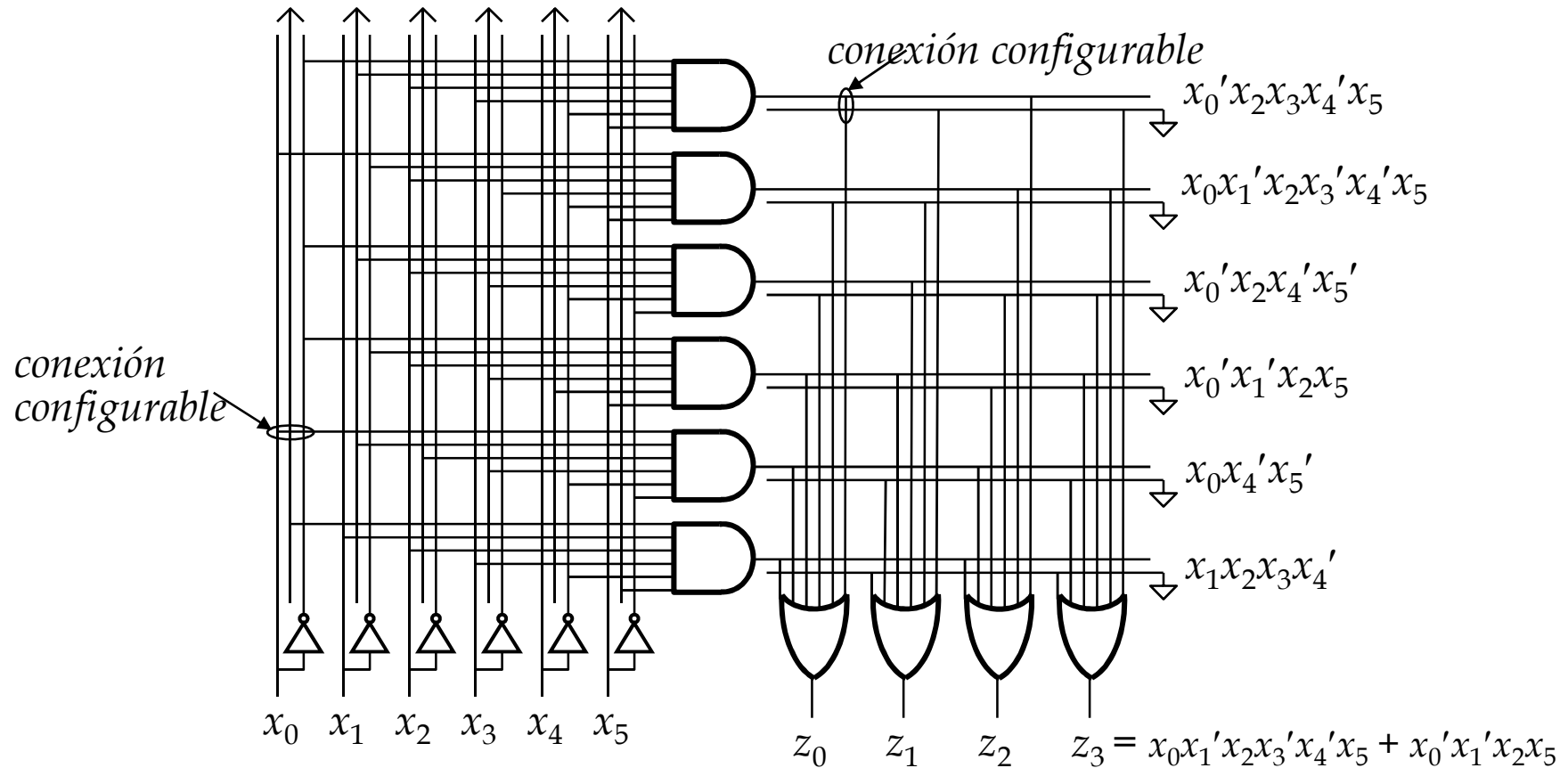
before programming: all fuses intact



after programming: fuses blown to remove unneeded connections

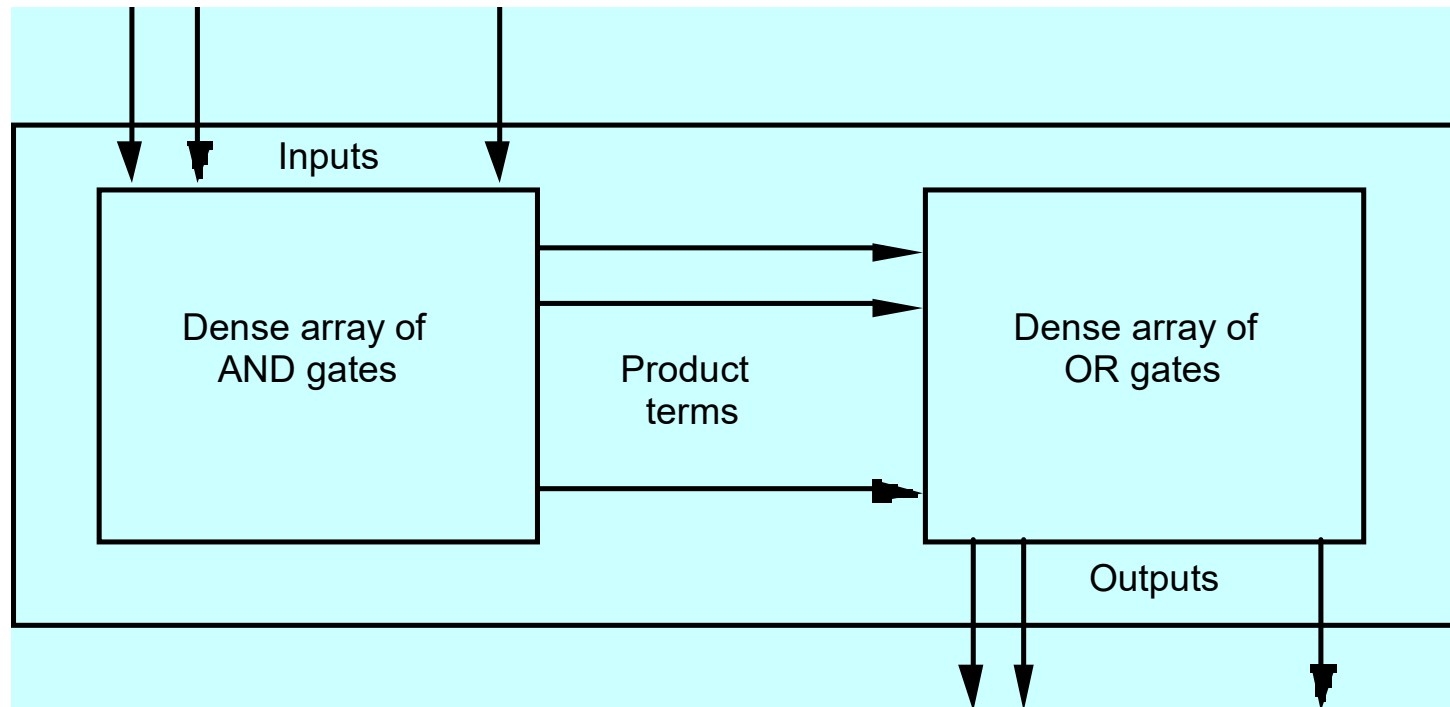
Las PLA son estructuras más generales que las PAL o ROM. También tienen un arreglo de ANDs y uno de ORs ambos programables . El tamaño del arreglo AND depende de la cantidad de entradas pero no tiene la cantidad de ANDs como una PROM ( $2^n$ ); mientras que el tamaño del arreglo OR depende de la cantidad de salidas.

## Programmable Logic Array (PLA)



- PLA: tiene configurable "AND-plane" y "OR-plane".
- "AND-plane" – entrada y "OR-plane" – salida
- Utilizado para implementar suma de productos de dos niveles

## Programmable Logic Array (PLA)



- Diagrama en bloques de una PLA (Suma de Productos)

# Programmable Devices

**F1 = A B C**

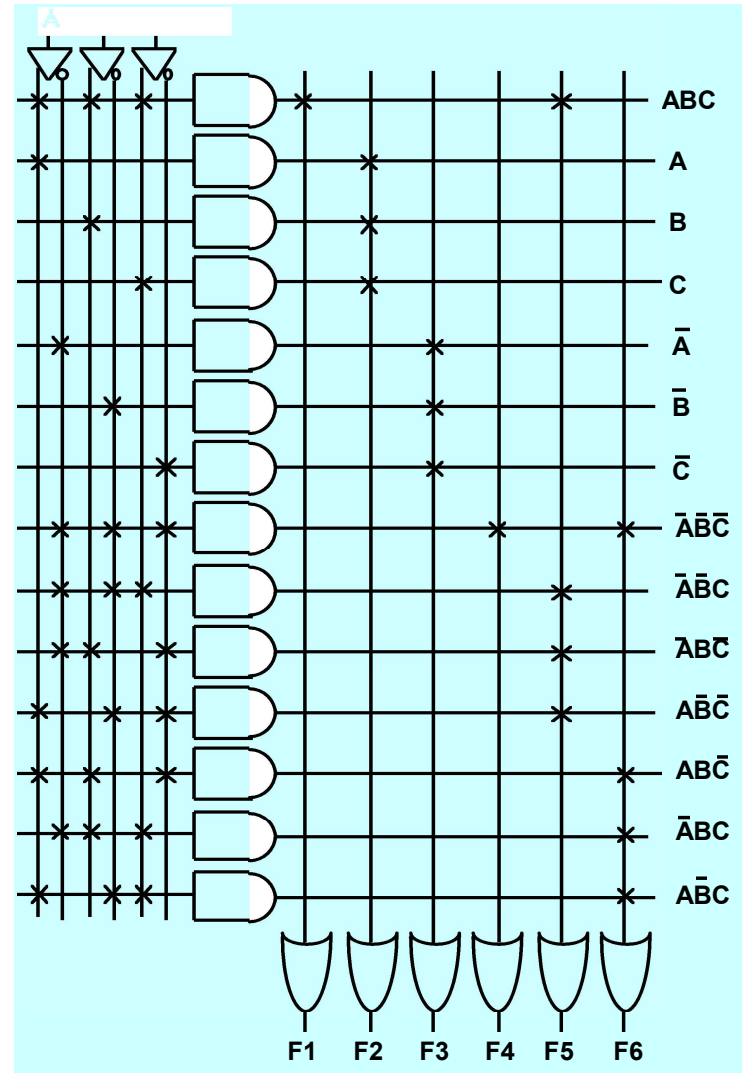
**F2 = A + B + C**

**F3 = A B C**

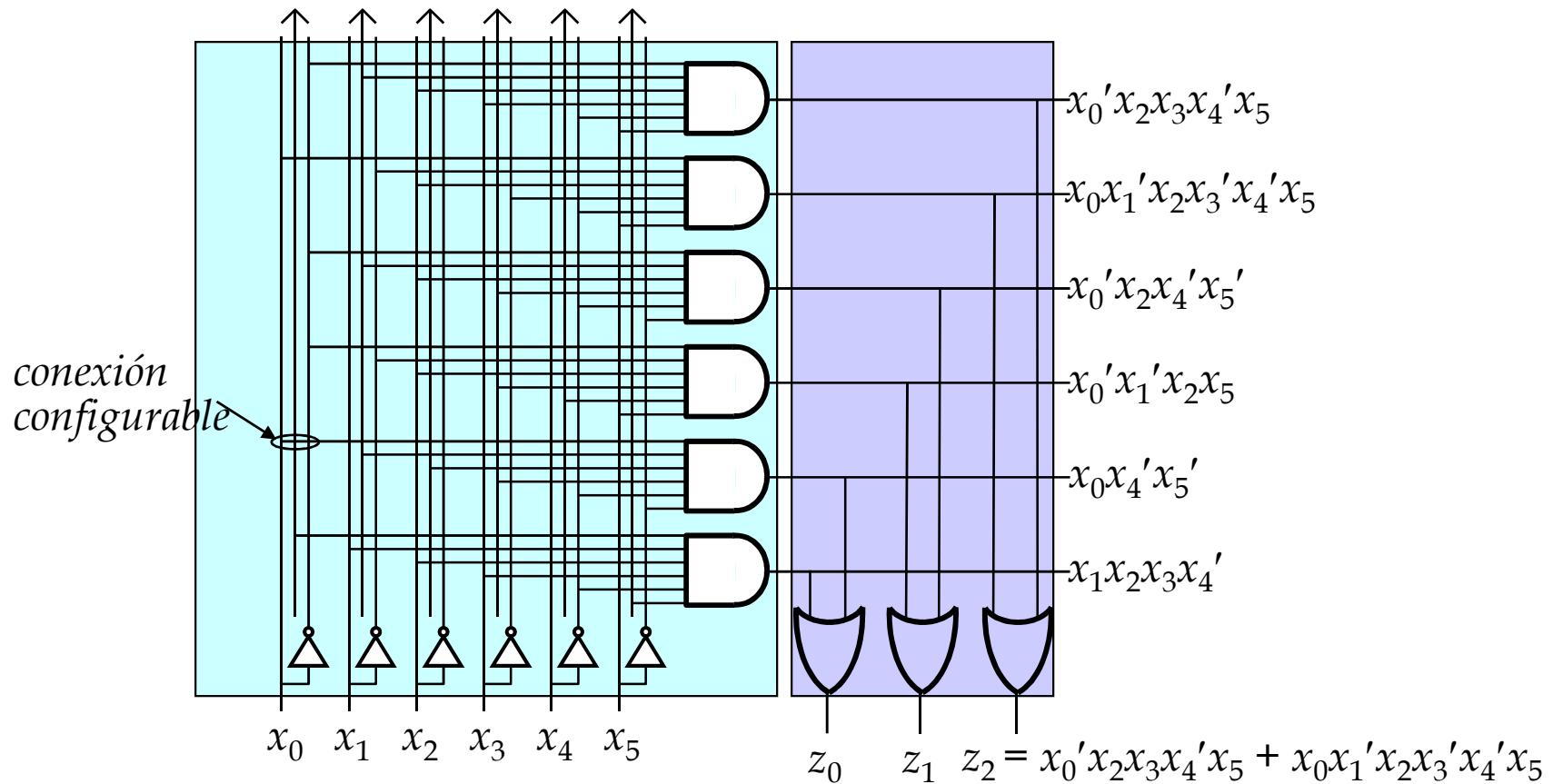
**F4 = A + B + C**

**F5 = A xor B xor C**

**F6 = A xnor B xnor C**



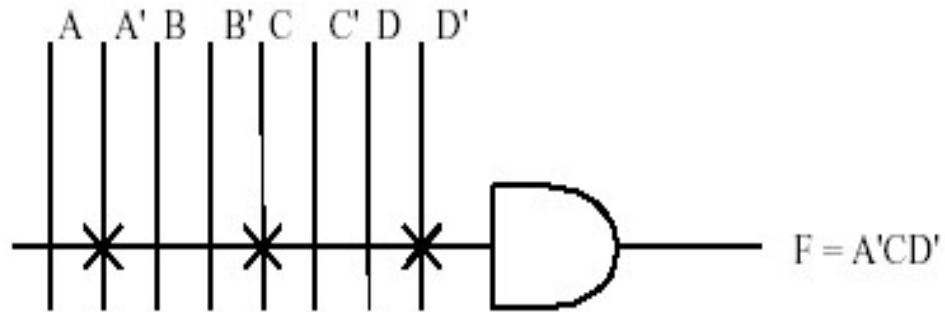
## Programmable Array Logic (PAL)



- PAL es similar a las PLAs pero tiene fijo el arreglo OR.
- Más simple de programar y de construcción más barata.
- Para cada salida está determinada la cantidad de términos.

# Programmable Array Logic (PAL)

PAL notación de programación



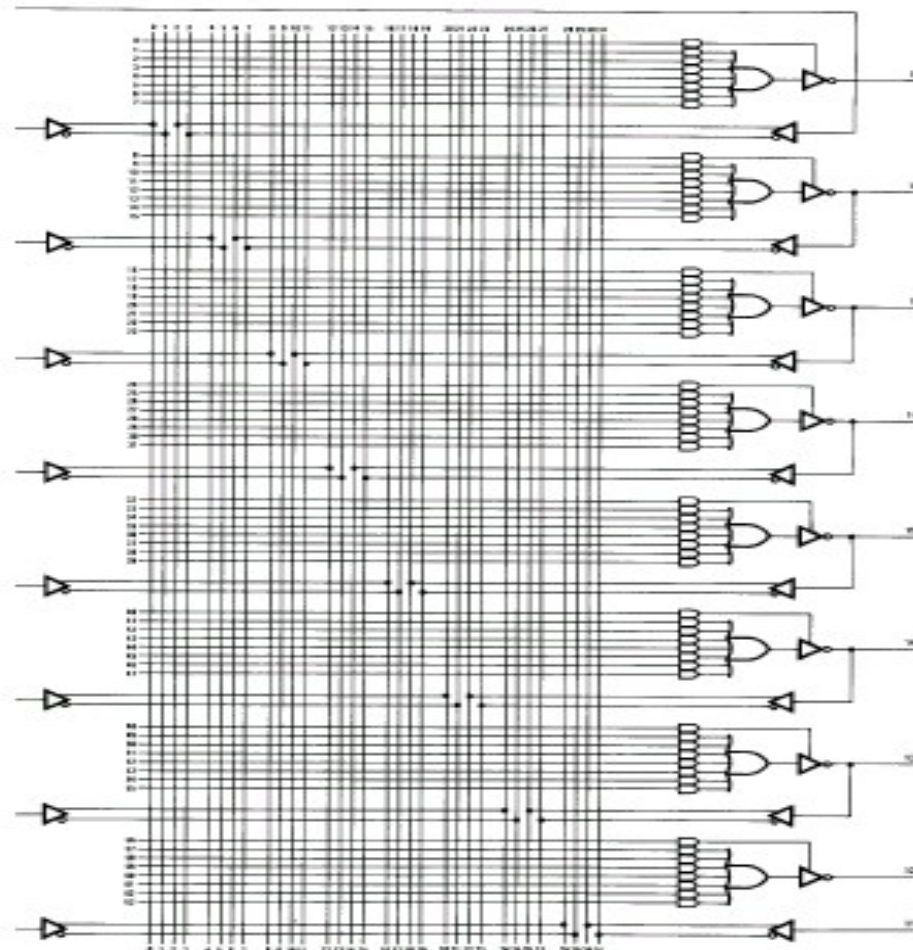
an X is used to indicate a programmed



complemented and uncomplemented outputs are available

# Programmable Array Logic (PAL)

La PAL de la figura consiste de un número fijo de compuertas AND de entradas múltiples combinadas con compuertas OR. Las conexiones a cada AND son programables, hay flexibilidad en las funciones lógicas realizables. La única limitación es la cantidad fija de miniterminos..



# Programmable Array Logic (PAL)

## Ejemplo de diseño: Conversor BCD a código de Gray

Tabla de verdad

A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	1	1	1	0
0	1	1	0	1	0	1	0
0	1	1	1	1	0	1	1
1	0	0	0	1	0	0	1
1	0	0	1	1	0	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

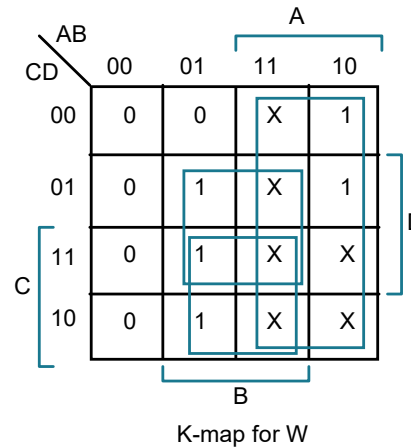
Funciones mínimas:

$$W = A + B D + B C$$

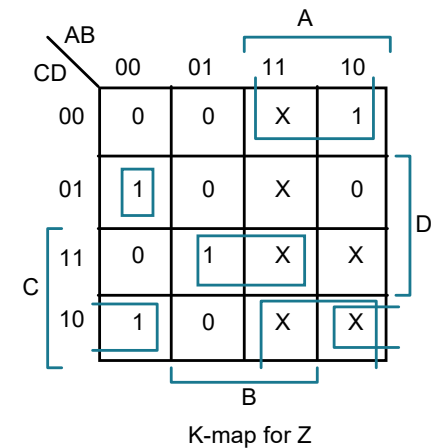
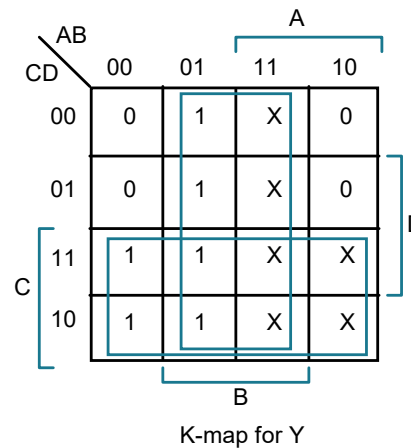
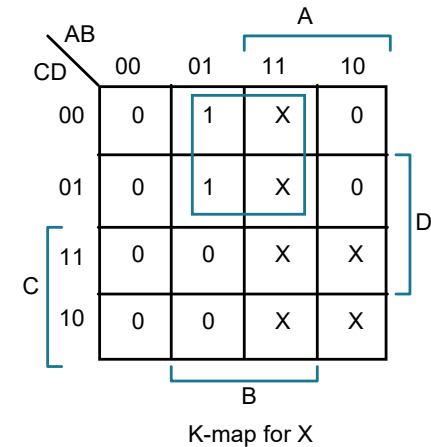
$$X = B C'$$

$$Y = B + C$$

$$Z = A'B'C'D + B C D + A D' + B' C D'$$



## Mapas K

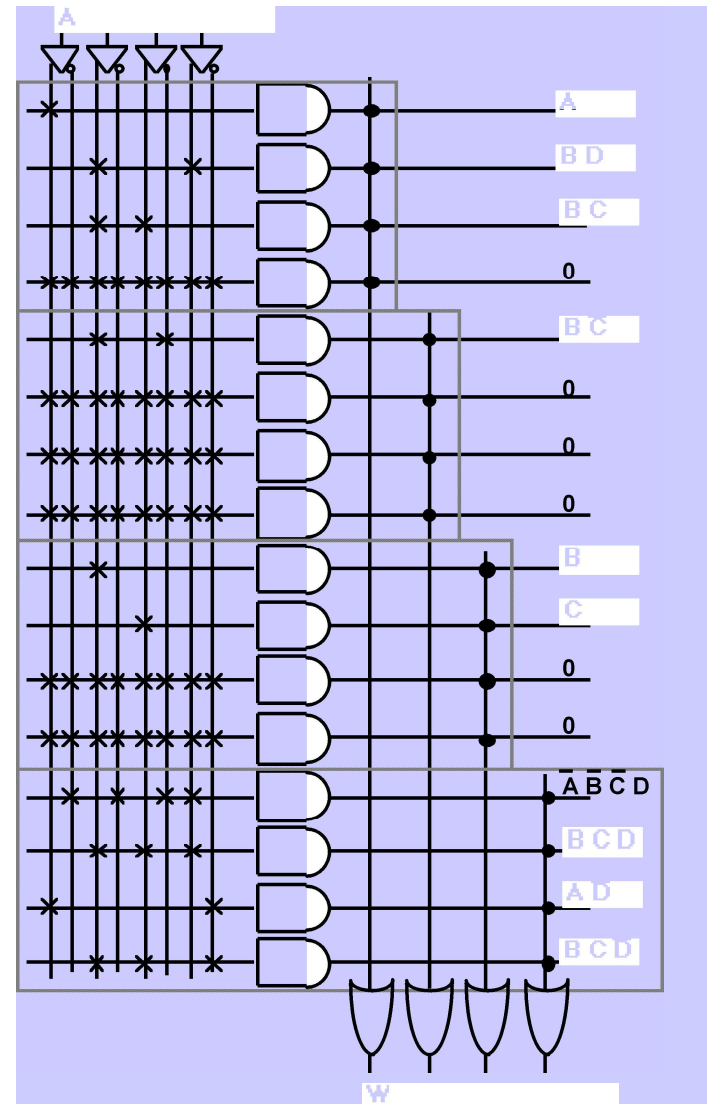




# Programmable Array Logic (PAL)

Ejemplo de diseño: Conversor  
BCD a código de Gray

PAL programada



4 términos de producto para  
cada compuerta OR

## Comparación de PROM, PLA y PAL

- **Se pueden considerar las PROMs y las PALs como versiones restringidas de PLAs.**
  - La PROM es una PLA con arreglo AND que genera todos los minterminos y un arreglo OR configurable
  - La PAL es una PLA con un arreglo OR fijo, y cada salida es un subconjunto de las ANDs
- **Selección de acuerdo a las necesidades.**
  - Se observará las ecuaciones a implementar
  - Tener en cuenta la cantidad de entradas y salidas
  - Comparar costos
- **Las PROMs son más flexibles, pero utilizar PALs y PLAs donde resulte posible.**
  - Las que incluyen biestables permiten realizar circuitos secuenciales sincrónicos

## **Herramientas de diseño**

- **Las herramientas CAD realizan diseño en base a especificaciones en lenguajes descriptores de hardware**
- **PALASM y ABEL elementales**
- **VHDL y Verilog**