

86.01 Técnica Digital

Memorias y Dispositivos Lógicos Programables

Ing. Jorge H. Fuchs

Objetivos de la clase:

Conocer los distintos tipos de memorias como también su organización interna.

Analizar el concepto de módulo universal en el sentido de un circuito capaz de permitir la síntesis de cualquier tipo de función lógica dada su facilidad de programación.

Conocer los distintos tipos de Dispositivos Lógicos Programables (PLD) y sus características.

Estudiar su utilización en la implementación de funciones lógicas de circuitos combinatoriales.

Clasificación de las memorias según su acceso:

Memorias de acceso secuencial

El tiempo de acceso a los datos depende de su ubicación. Ejemplos: Buffer, Stack, etc. Las veremos más adelante.

Memorias de acceso aleatorio (Random Access Memory)

El tiempo de acceso a los datos es independiente de su ubicación. Es una característica de cada una. Ejemplos: RAM, ROM, HDD, SSD, Pen Drive, etc.

Memorias de acceso aleatorio

Clasificación de las memorias de acceso aleatorio (RAM):

Memorias de Lectura / Escritura (conocidas como RAM)

Memorias de sólo lectura (ROM)

Memorias de mayormente lectura

Memorias de acceso aleatorio

Memorias de Lectura/Escritura (conocidas como RAM):

DRAM (Dynamic Random Access Memory):

Los datos se almacenan como en la carga de un capacitor. Tiende a descargarse y, por lo tanto, es necesario un proceso de refresco periódico. Son de mayor capacidad, más simples y baratas que las SRAM. Son volátiles.

SRAM (Static Random Access Memory):

Los datos se almacenan formando biestables (las veremos más adelante), por lo que no requiere refresco. Igual que la DRAM son volátiles. Son más rápidas que las DRAM y más caras.

Memorias de acceso aleatorio

Memorias de sólo lectura (ROM):

ROM (Read Only Memory):

Se usan principalmente en microprogramación de sistemas. Los fabricantes las suelen emplear cuando producen componentes de forma masiva.

PROM (Programmable Read Only Memory):

El proceso de escritura es eléctrico. Se puede grabar (por el usuario) posteriormente a la fabricación del chip, a diferencia de las anteriores que se graba durante la fabricación. Permite una única grabación y es más cara que la ROM.

Memorias de acceso aleatorio

Memorias de sobre todo lectura:

EPROM (Erasable Programmable Read Only Memory):

Se puede escribir varias veces de forma eléctrica, sin embargo, el borrado de los contenidos es completo y a través de la exposición a rayos ultravioletas (suelen tener una pequeña “ventanita” en el chip).

EEPROM o E²PROM (Electrically Erasable PROM):

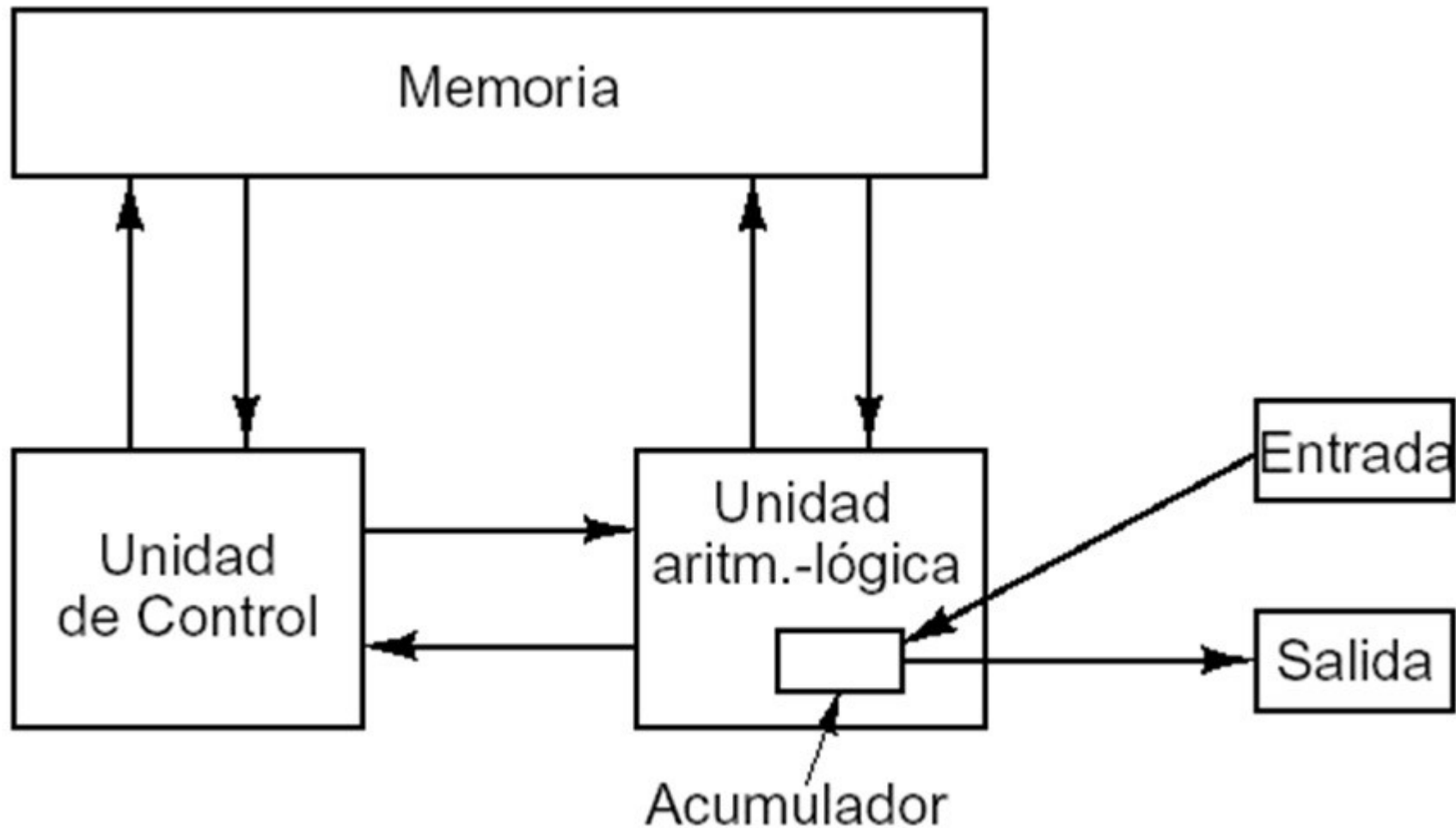
Se puede borrar selectivamente byte a byte con corriente eléctrica. Es más cara que la EPROM. El proceso de borrado es lento.

FLASH:

Está basada en las memorias EEPROM pero permite el borrado por bloques, por lo que este proceso es mucho más veloz. Es más barata y densa.

Memorias de acceso aleatorio

Esquema básico de una computadora:



Buses

Es un camino eléctrico común entre varios dispositivos y la CPU (Unidad Central de Procesamiento).

Ejemplos:

Bus de Direcciones (Address Bus)

Bus de Datos (Data Bus)

Bus de Control

Bus de Direcciones (Address Bus)

n líneas de Address → 2^n locaciones

n: 16, 20, 32, 64, etc.

Unidireccional

Tamaño de la memoria

Bus de Datos (Data Bus)

d líneas de Data → d bit / palabra

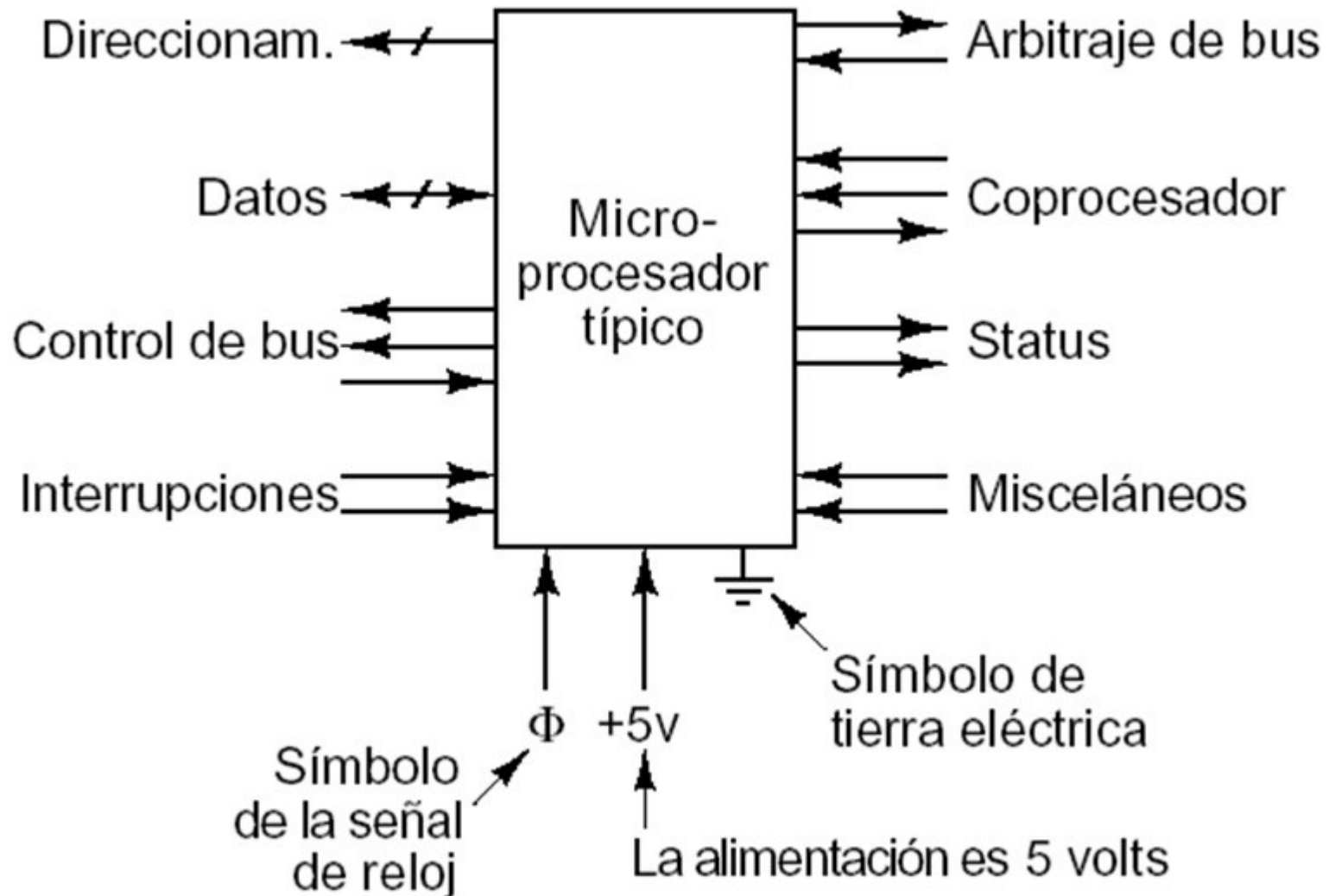
d: 8, 16, 32, 64.

Bidireccional

Ancho de banda

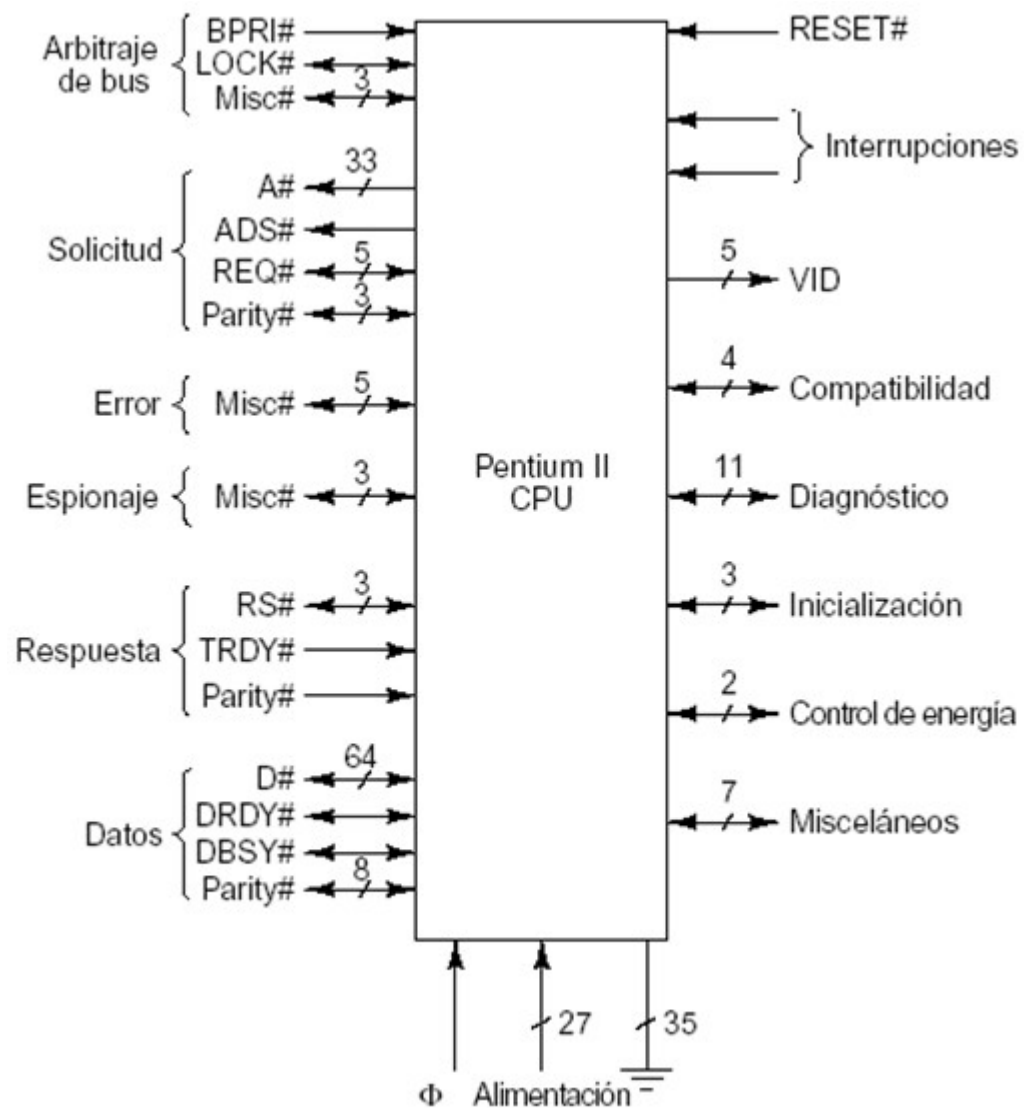
Buses

Buses externos de una CPU:



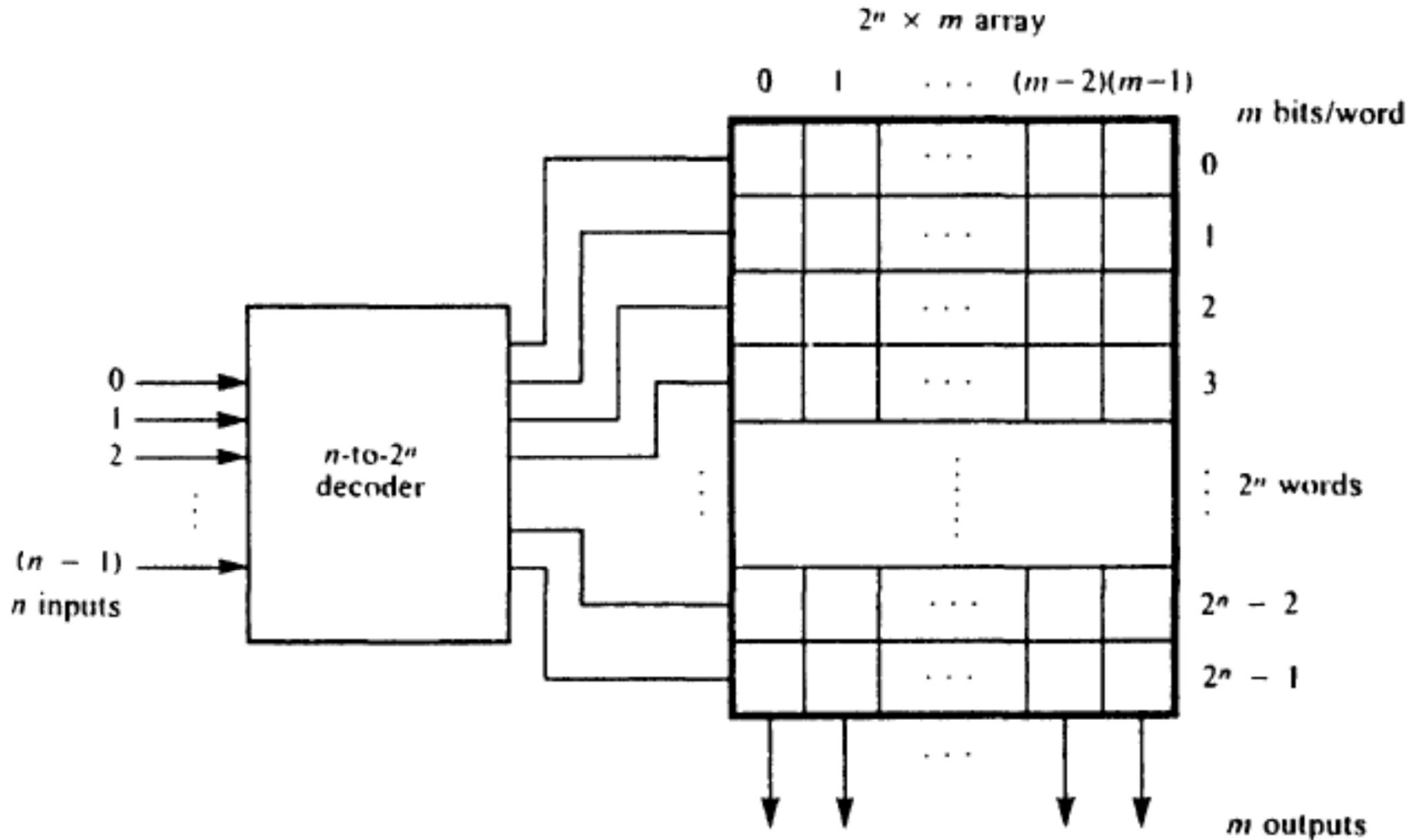
Buses

Ejemplo de buses externos de una CPU:



Organización de memorias

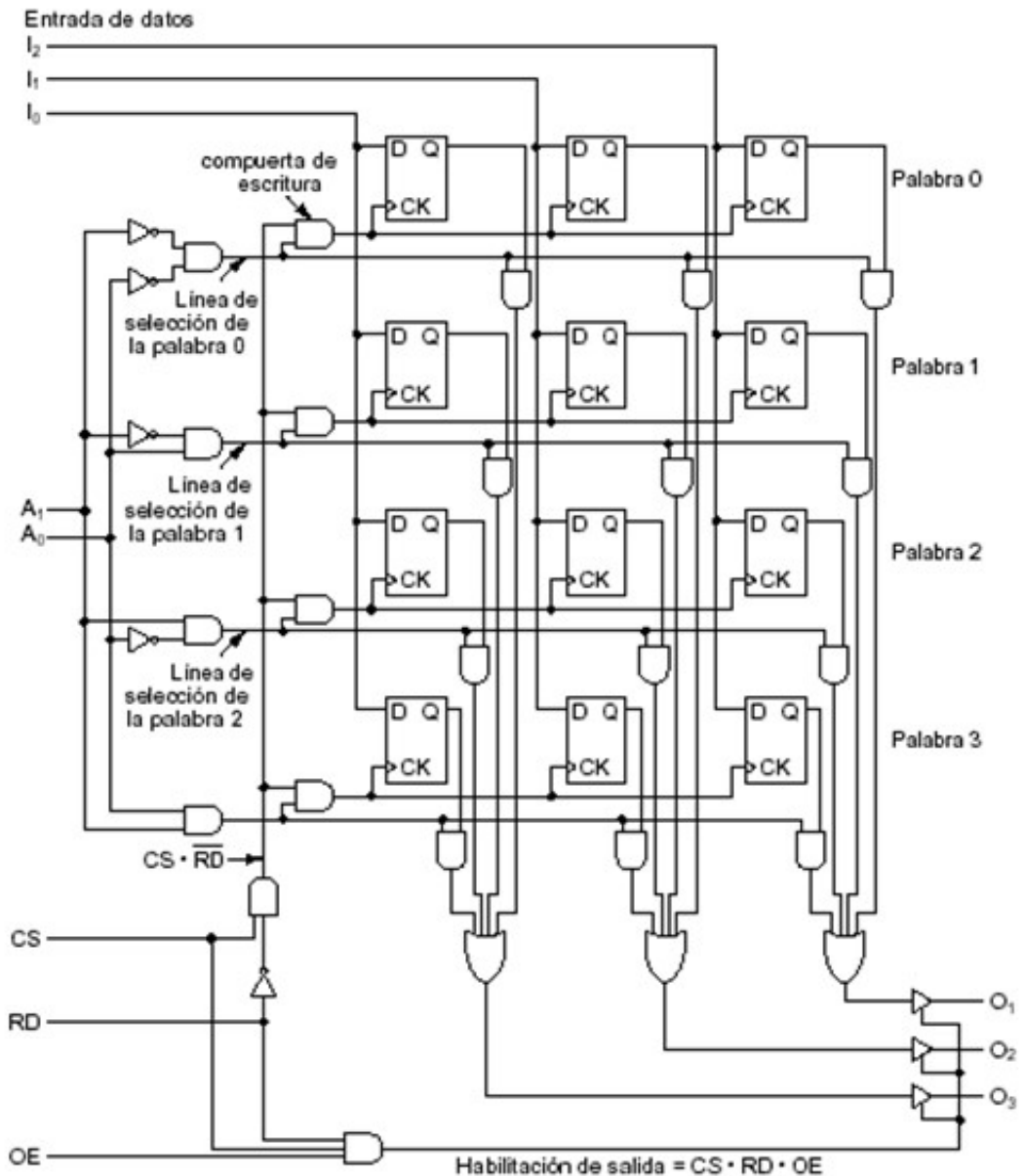
Organización de una memoria de acceso aleatorio:



Organización de memorias

Organización de una memoria de 4 palabras de 3 bits cada una.

Cada renglón tiene 3 bits, una operación de lectura o escritura siempre lee o escribe una palabra completa (3 bits).



Organización de memorias

Selección de las m palabras (Address Bus: n líneas)

- Cantidad de palabras: $m = 2^n$

Organización en palabras de L bits (Data Bus: d líneas)

- Longitud de palabra: $L = d$

Capacidad de la memoria

- Capacidad (cant. total de celdas): $c = m L$

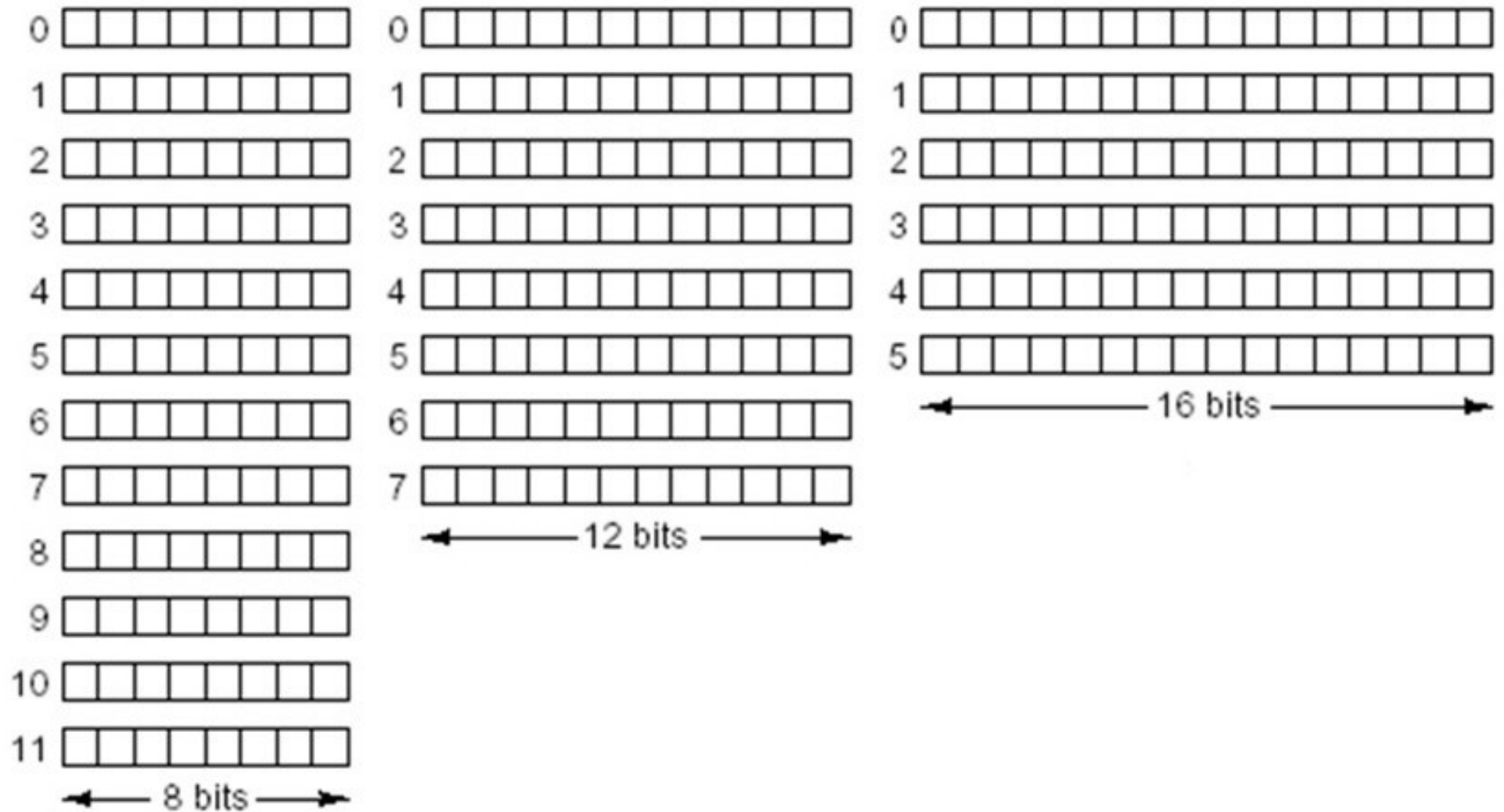
Organización de memorias

Más ejemplos:

	Bus de datos	Bus de direcciones	Capacidad en bits
1024 x 1	1	10	1024
256 x 4	4	8	1024
2048 x 8	8	11	16384
4096 x 1	1	12	4096
8K x 8	8	13	65568
1M x 1	1	20	1048576

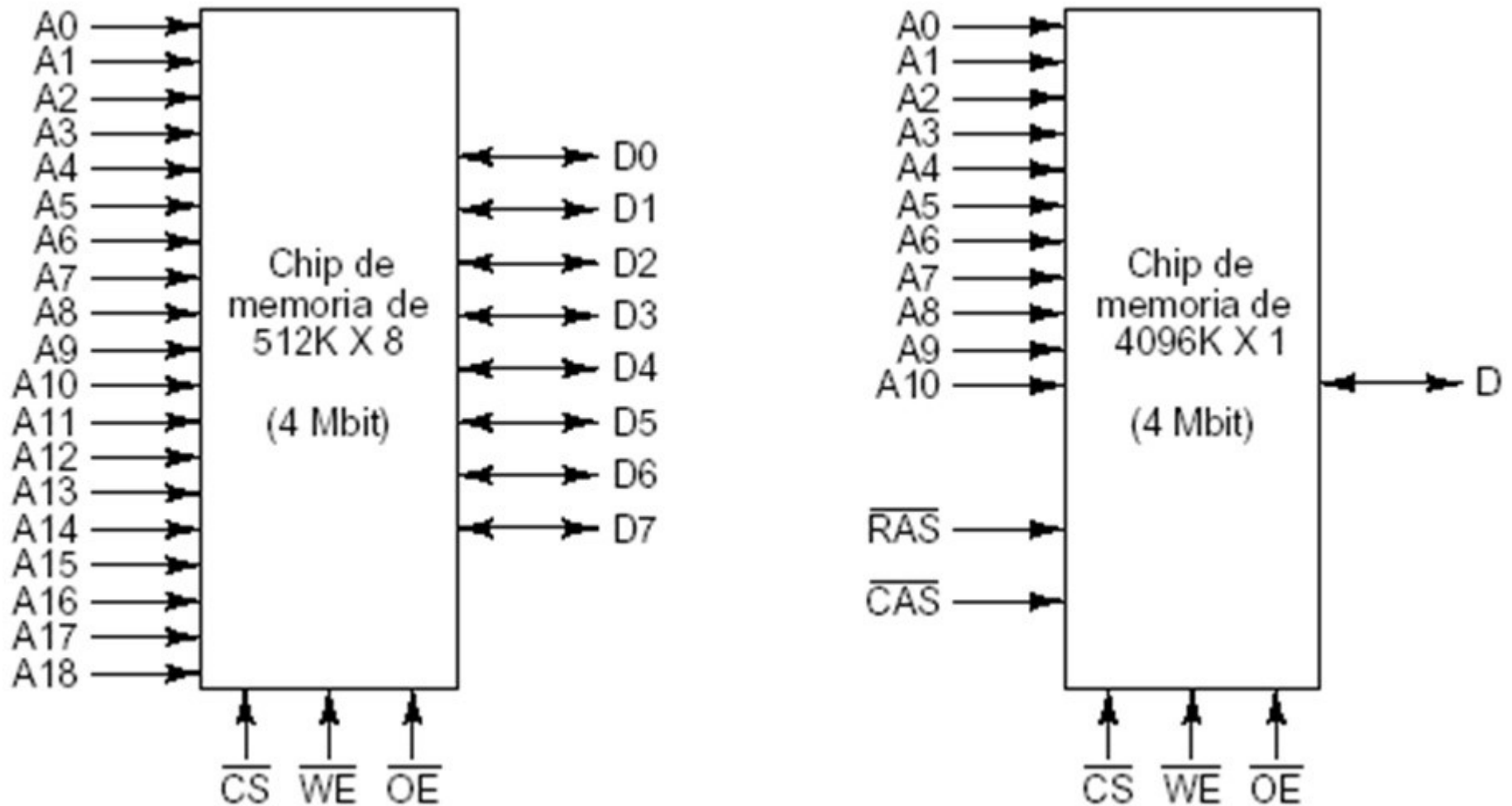
Organización de memorias

Varias formas de organización de una memoria de 96 bits:



Organización de memorias

Dos formas de organizar un chip de memoria de 4 Mbit:



Tipos de memorias y usos

Comparación de distintos tipos de memorias:

Tipo	Categoría	Borrado	Alterable por byte	Volátil	Uso típico
SRAM	Lect./escrit.	Eléctrico	Sí	Sí	Caché nivel 2
DRAM	Lect./escrit.	Eléctrico	Sí	Sí	Memoria principal
ROM	Sólo lectura	Imposible	No	No	Aparatos en vol. grandes
PROM	Sólo lectura	Imposible	No	No	Equipos en vol. pequeños
EPROM	Princ. lectura	Luz UV	No	No	Prototipos de dispositivos
EEPROM	Princ. lectura	Eléctrico	No	No	Prototipos de dispositivos
Flash	Lect./escrit.	Eléctrico	No	No	Película para cámara digital

Dispositivos Lógicos Programables (PLD)

Tipos de PLD:

Read Only Memory (ROM)

Programmable ROM (PROM, EPROM, E²PROM)

Programmable Logic Arrays (PLA)

Programmable Array Logic (PAL)

Erasable PLD (EPLD)

Field Programmable Gate Arrays (FPGA)

Dispositivos Lógicos Programables (PLD)

Arreglos lógicos sencillos

Basados en circuitos lógicos de 2 niveles (AND/OR).

Tienen una estructura de arreglo regular.

Distintos tipos:

Memorias de solo lectura (ROM, PROM, E²PROM)

Programmable Logic Array (PLA)

Programmable Array Logic (PAL)

Arreglos programables en campo Field Programmable Gate Arrays (FPGA)

Bloque básico común replicado muchas veces.

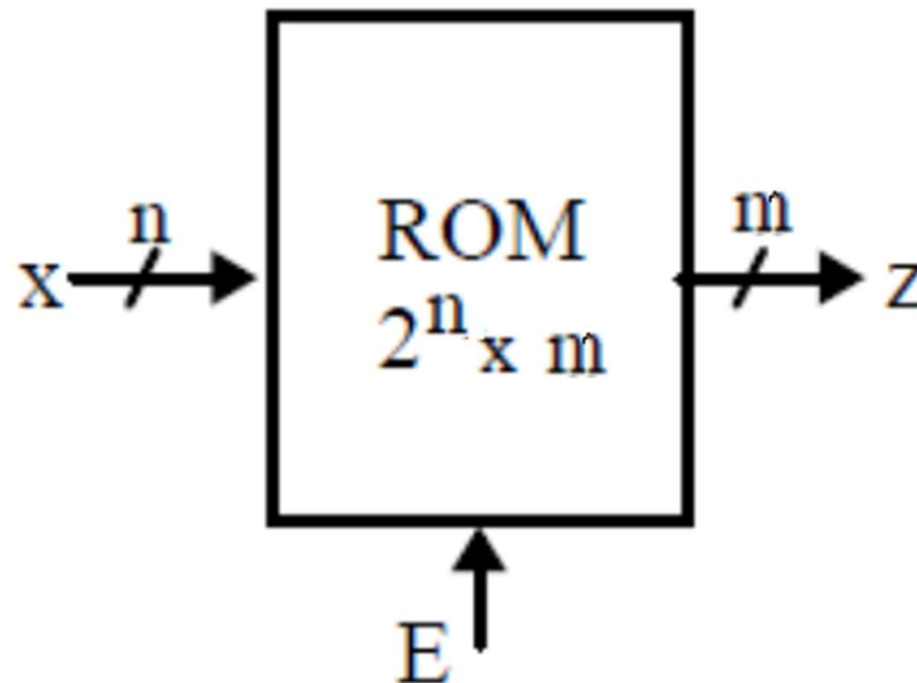
Cada bloque puede configurarse para producir distintas funciones lógicas y en general incluyen biestables (flip-flops).

Interconexiones programables.

Las FPGAs más grandes tienen alrededor de 500K compuertas además de 500 Kb de SRAM.

Read Only Memory (ROM)

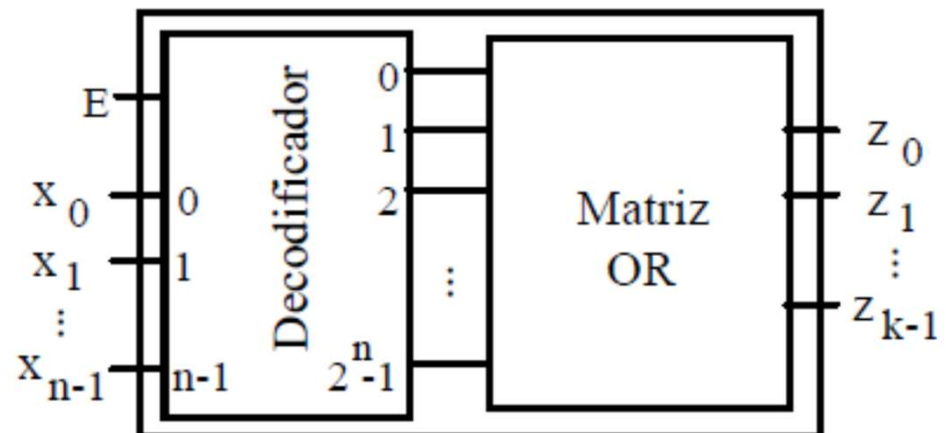
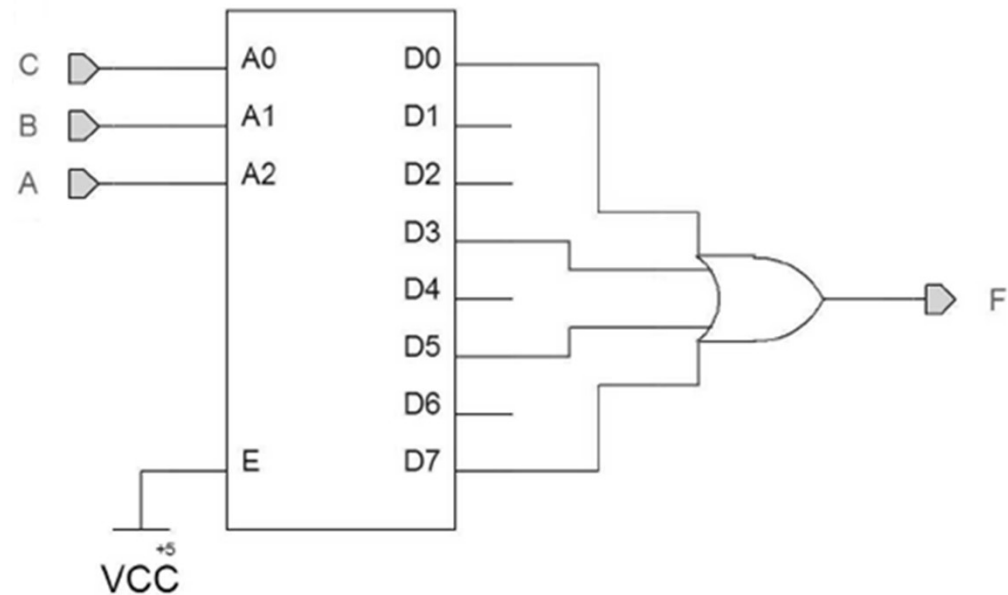
Una **ROM** de $2^n \times m$ bits es un circuito lógico combinacional con **n** entradas y **m** salidas que implementa **m funciones lógicas de n variables**. El circuito puede disponer también de entradas adicionales de habilitación.



Read Only Memory (ROM)

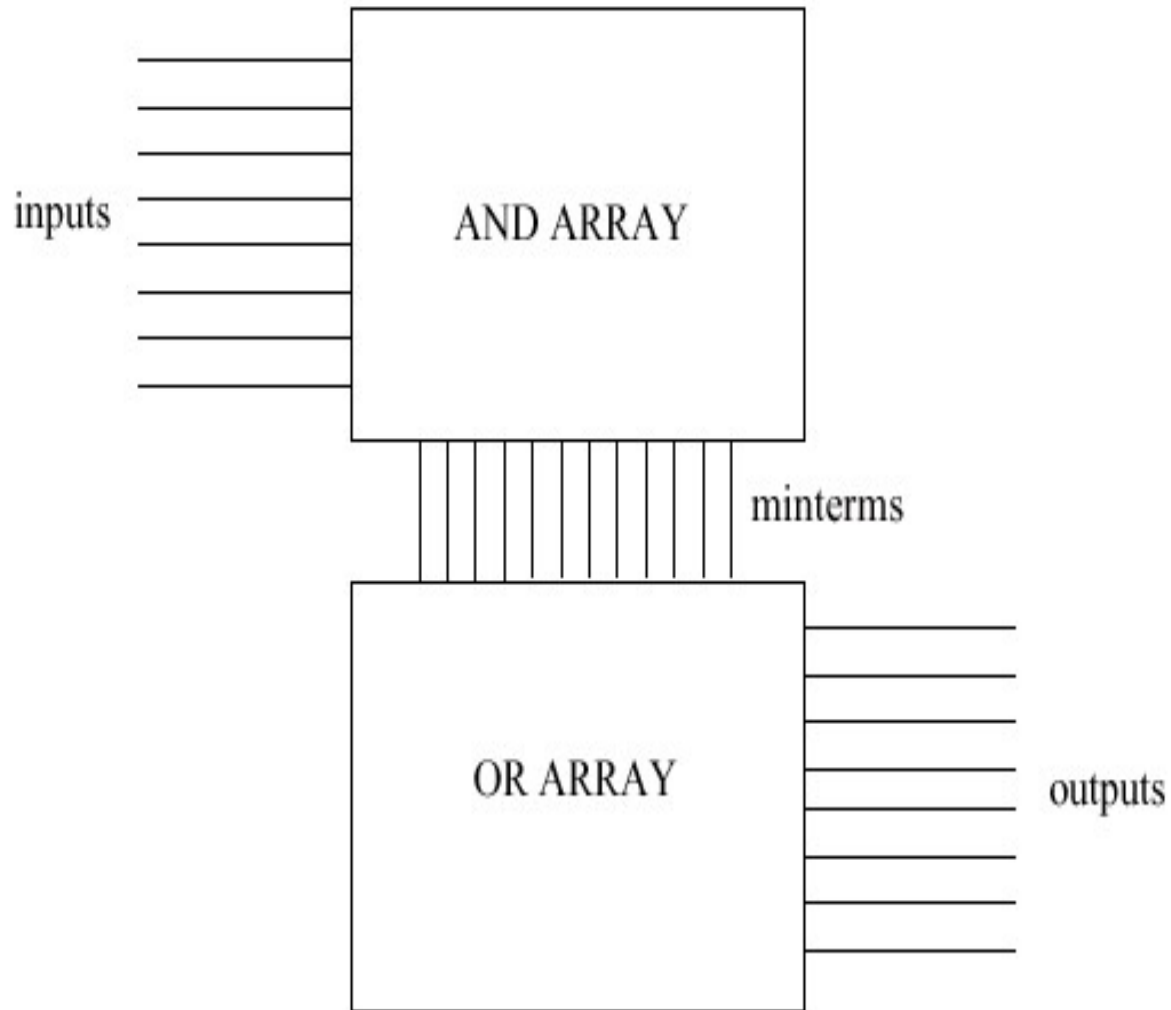
Recordando la implementación de funciones con Decodificadores y OR:

	A	B	C	F
0)	0	0	0	1
1)	0	0	1	0
2)	0	1	0	0
3)	0	1	1	1
4)	1	0	0	0
5)	1	0	1	1
6)	1	1	0	0
7)	1	1	1	1



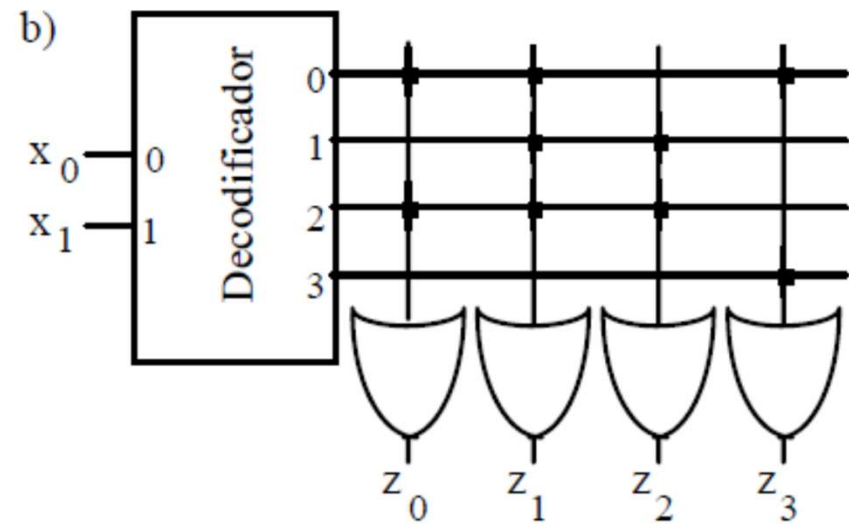
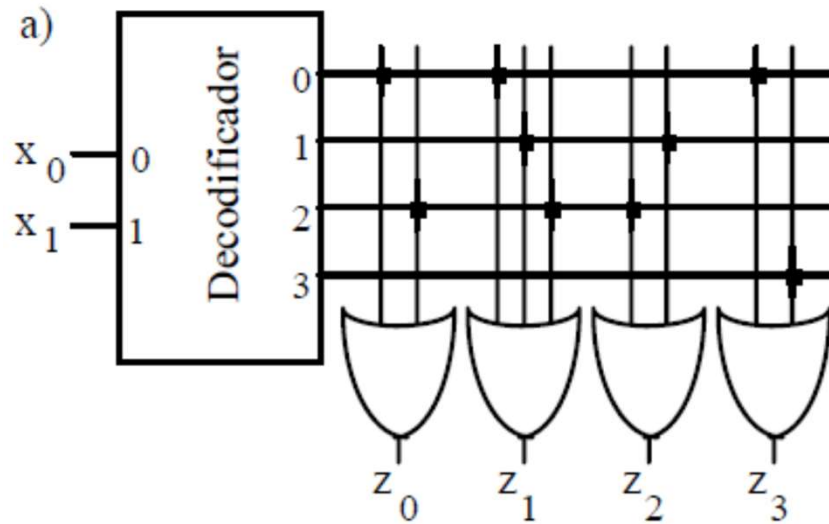
Read Only Memory (ROM)

Esquema genérico:



Read Only Memory (ROM)

a) ROM de 4 x 4 bits. b) Representación abreviada de la matriz OR.



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.

Incluyen el decodificador (compuertas **AND**) y las compuertas **OR** en un solo circuito integrado.

Es un depósito de información binaria fija (combinacional).

Los CI ROM y PROM tienen uniones internas que pueden quemarse o romperse programando.

Consisten en n líneas de entrada y m de salida.

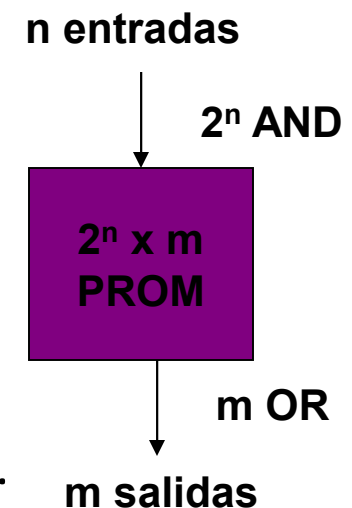
Cada combinación de entrada se llama **dirección**.

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.



Read Only Memory (ROM)

Las ROM y PROM pueden construirse utilizando arreglos ortogonales de cables.

Conexión opcional en cada intersección.

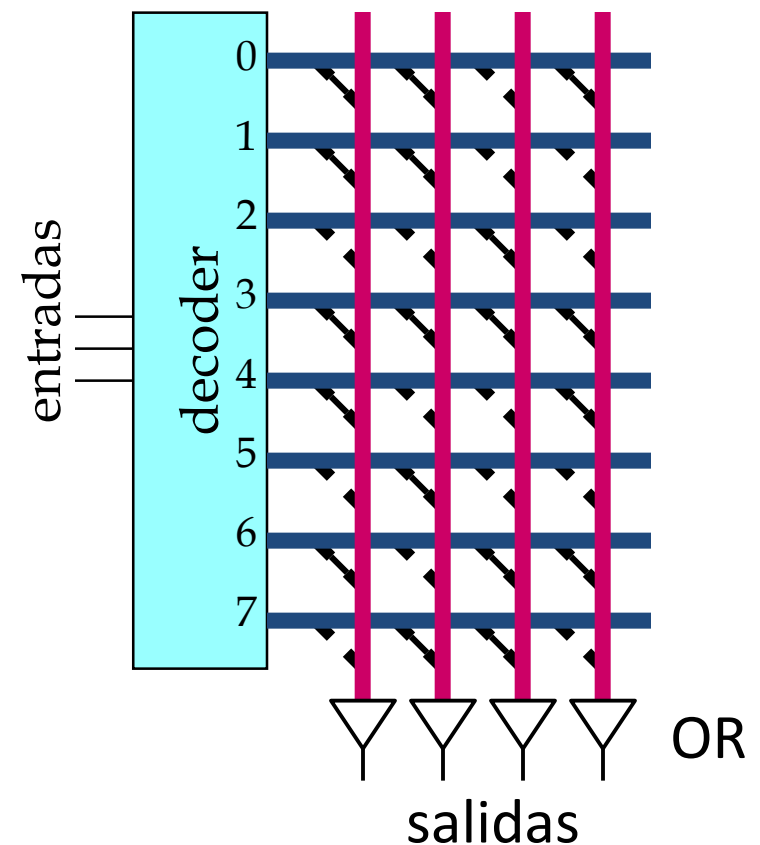
El decodificador coloca un 1 lógico en solo uno de los cables horizontales, que aparecerá a la salida de una de las OR si hay una conexión realizada.

Algunas PROM 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 quemara.

Otras "ROM" pueden borrarse y reprogramarse (EPROM, EEPROM, Flash).



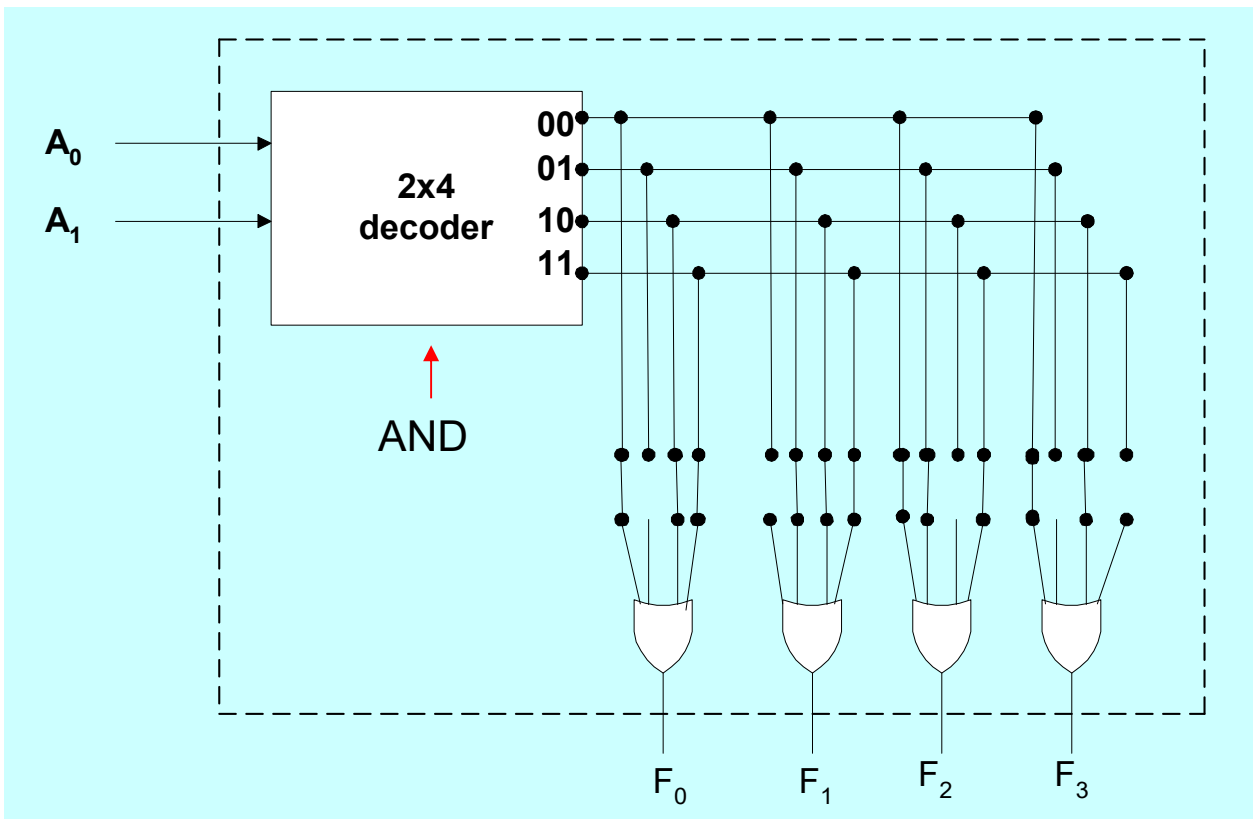
- **Funciones almacenadas**
 $\Sigma(0,1,3,4,6)$, $\Sigma(0,1,3,5,7)$,
 $\Sigma(2,3,6,7)$, $\Sigma(0,3,4,6)$

Read Only Memory (ROM)

Ejemplo: ROM 4 x 4 con funciones implementadas.

Tabla de verdad

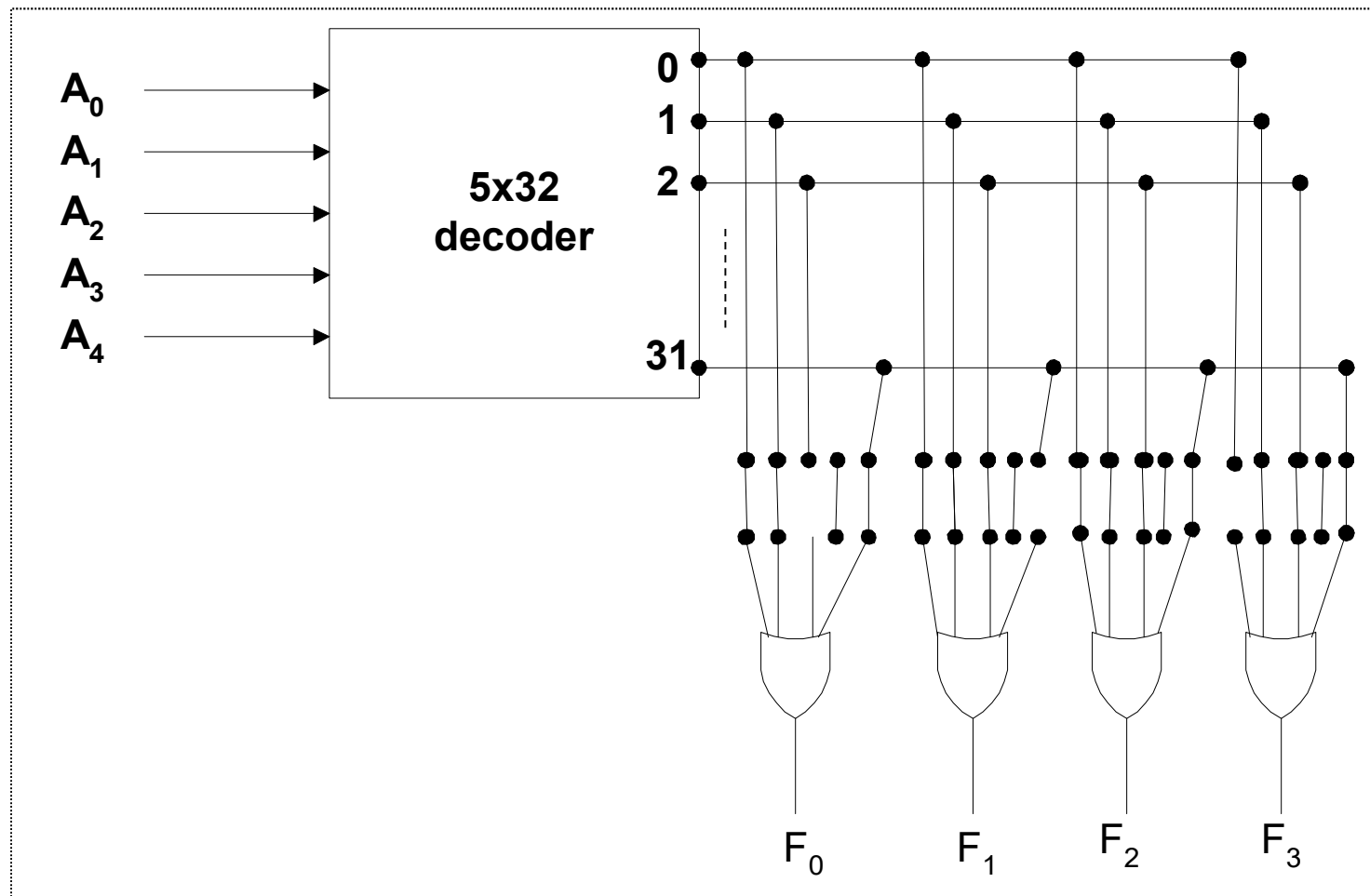
A_0	A_1	F_0	F_1	F_2	F_3
0	0	1	0	1	1
0	1	0	1	1	0
1	0	1	1	0	1
1	1	1	1	1	0



← OR

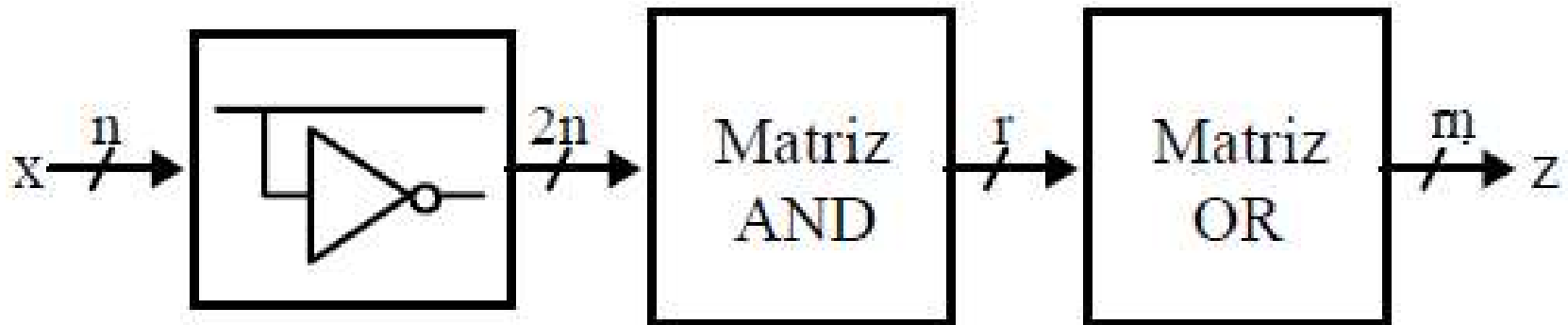
Read Only Memory (ROM)

Ejemplo: ROM 32 x 4



Programmable Logic Array (PLA)

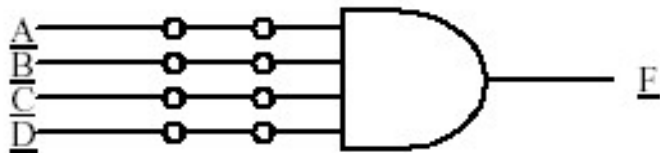
Un arreglo lógico programable (**PLA**) de $n \times r \times m$ es un circuito lógico combinacional con n entradas y m salidas que implementa **m funciones lógicas de n variables** utilizando r productos, siendo $r < 2^n$. Adicionalmente el circuito puede disponer de entradas de habilitación.



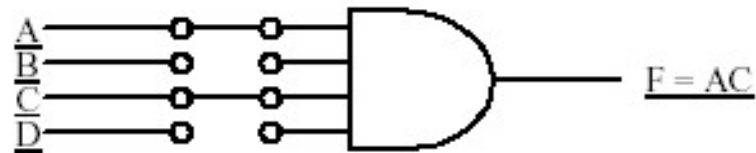
Programmable Logic Array (PLA)

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

Notación de programación:



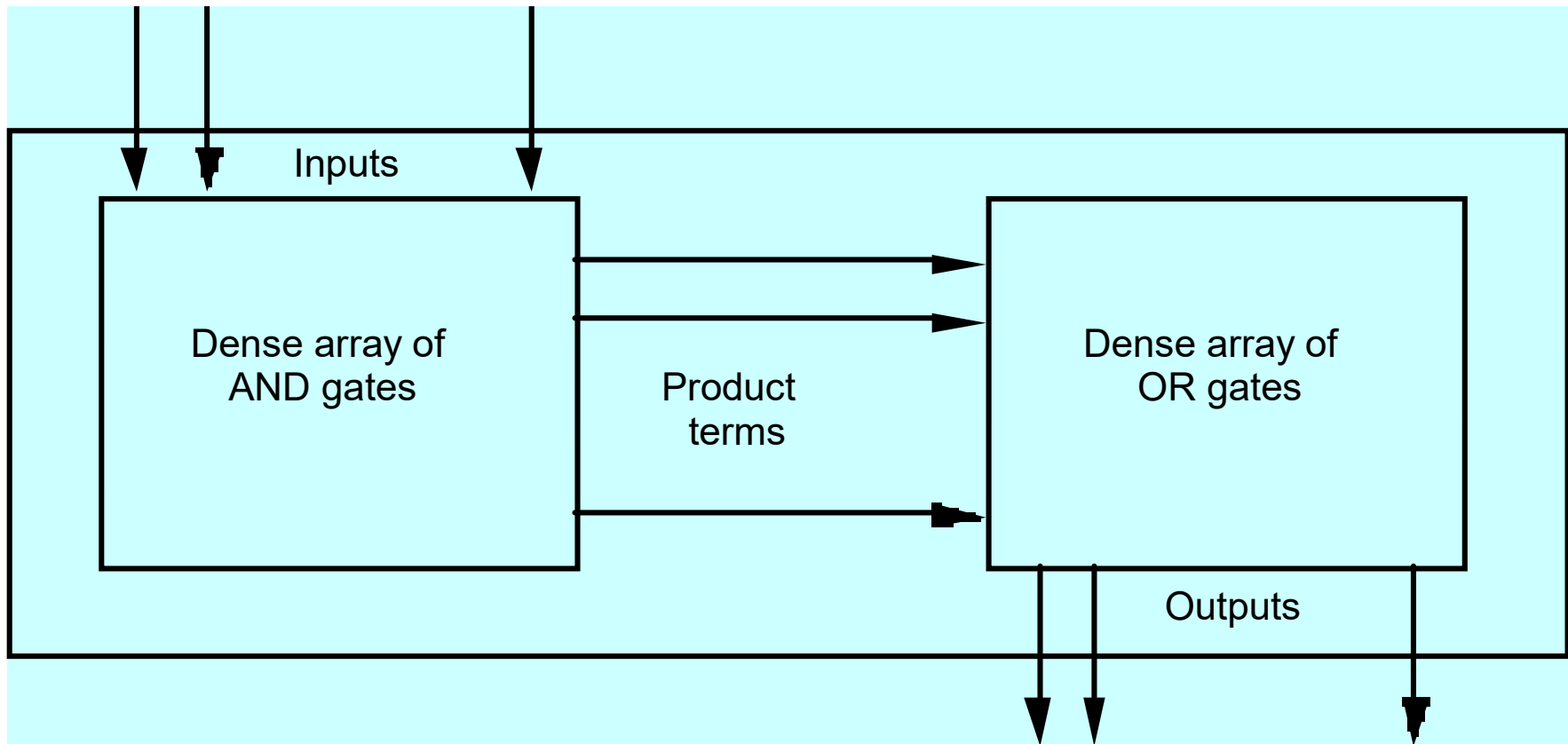
before programming: all fuses intact



after programming: fuses blown to remove unneeded connections

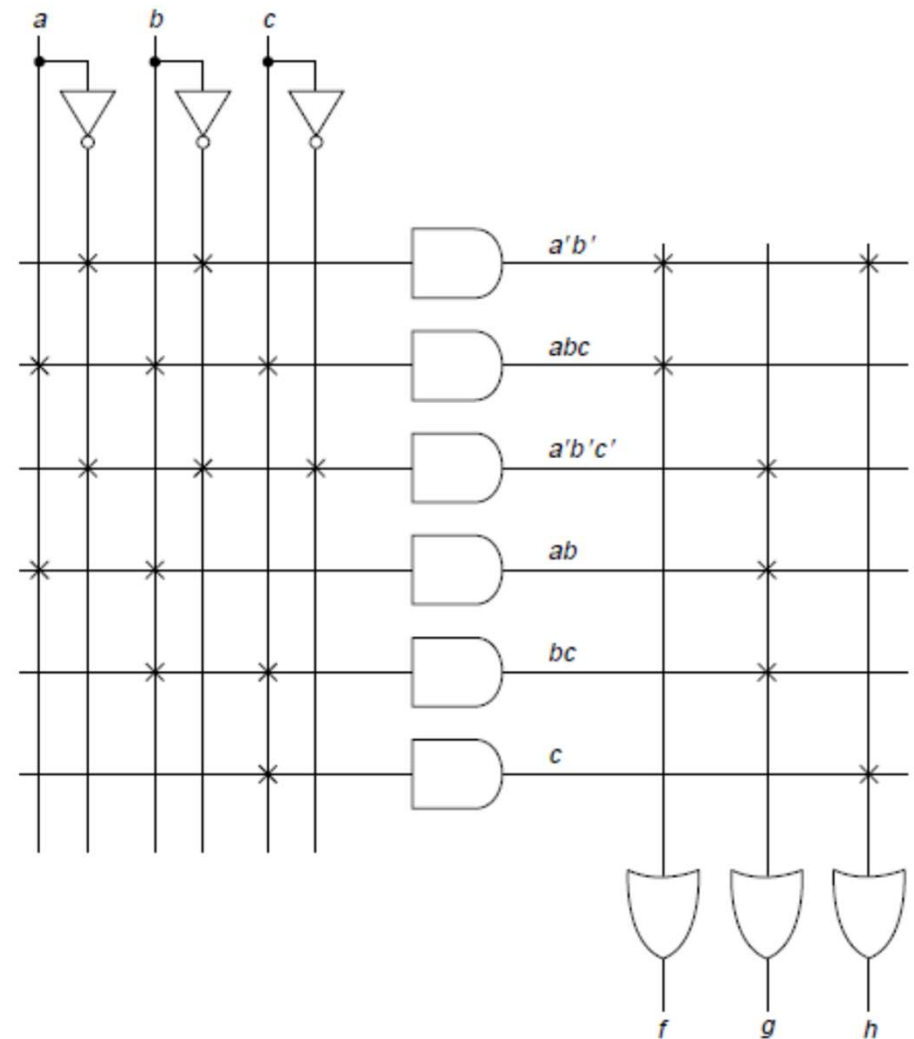
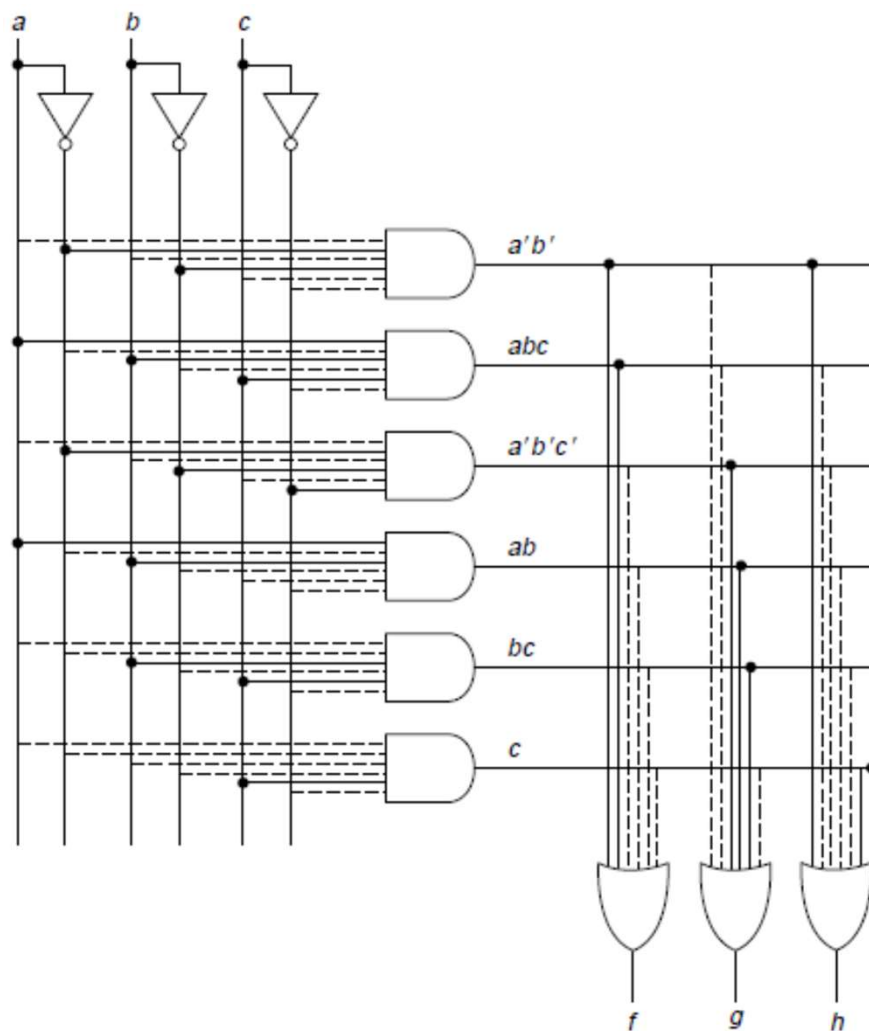
Programmable Logic Array (PLA)

Las PLA tienen configurable la matriz AND y también la matriz OR. Permiten implementar suma de productos (no necesariamente minterminos) de dos niveles.



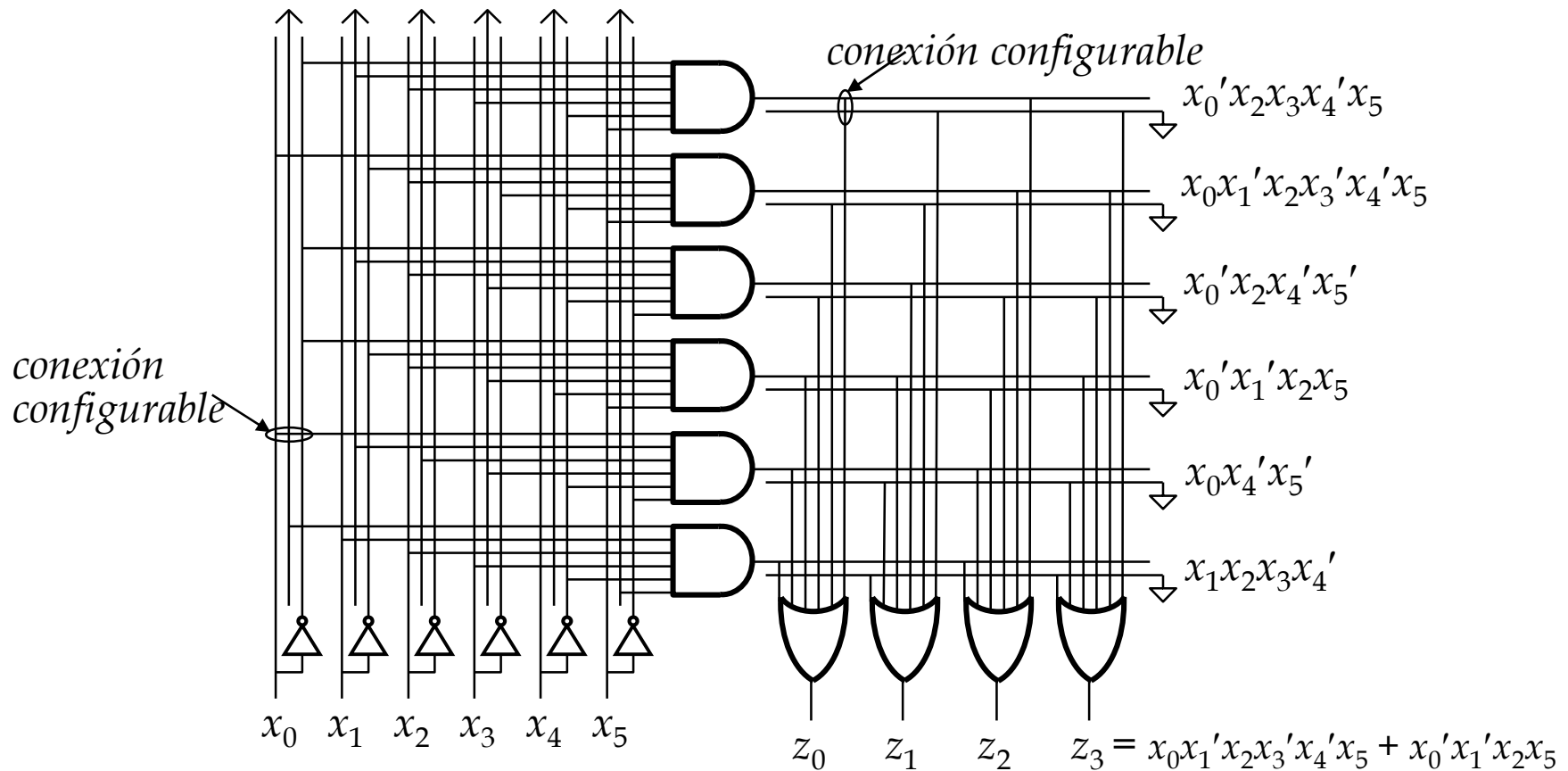
Programmable Logic Array (PLA)

Ejemplo de programación y notación de una PLA (Suma de Productos):



Programmable Logic Array (PLA)

Ejemplo:



Programmable Logic Array (PLA)

Ejemplo: Implementación de 6 funciones de 3 variables:

$$F1 = A B C$$

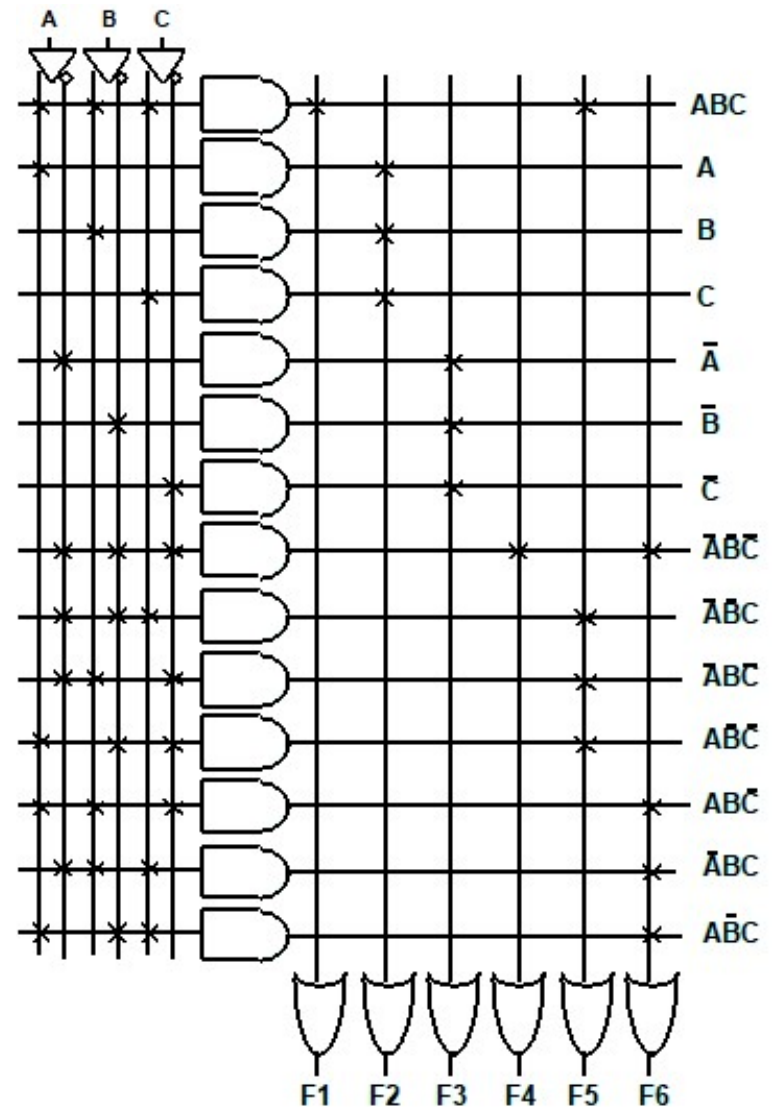
$$F2 = A + B + C$$

$$F3 = A B C$$

$$F4 = A + B + C$$

$$F5 = A \text{ xor } B \text{ xor } C$$

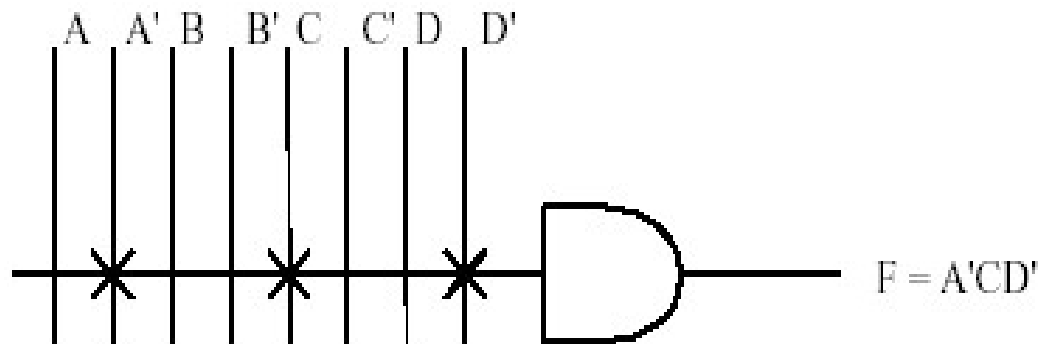
$$F6 = A \text{ xnor } B \text{ xnor } C$$



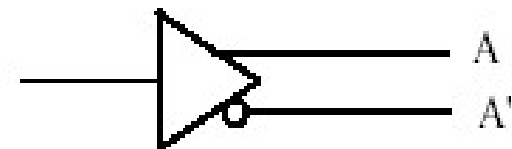
Programmable Array Logic (PAL)

PAL es similar a la PLA pero tiene fijo el arreglo OR.
Más simple de programar y de construcción más barata.
Para cada salida está determinada la cantidad de términos de la SP.
Generalmente se dispone también de las salidas complementadas.

Notación de programación:



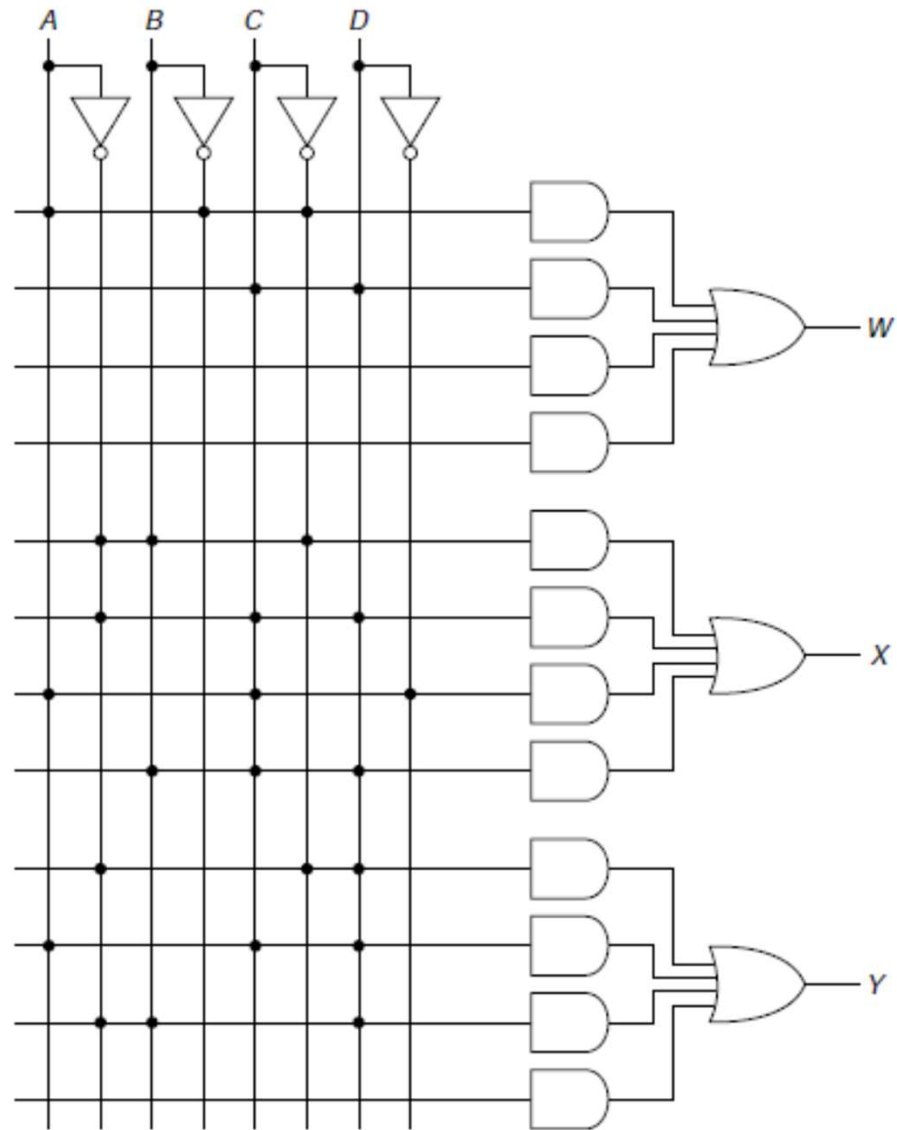
an X is used to indicate a programmed



complemented and uncomplemented
outputs are available

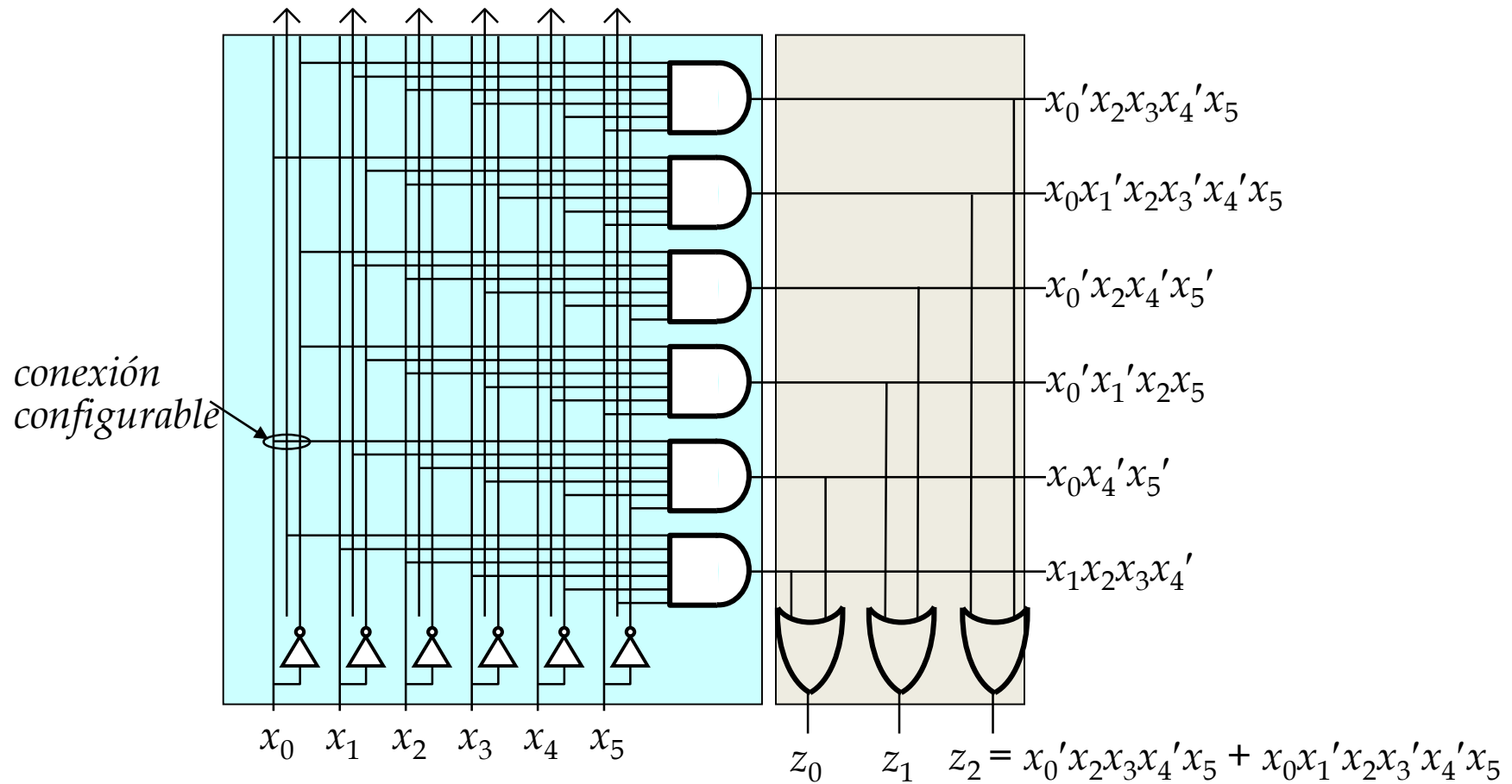
Programmable Array Logic (PAL)

Ejemplo:



Programmable Array Logic (PAL)

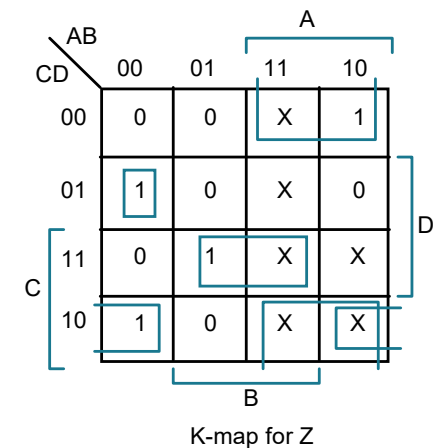
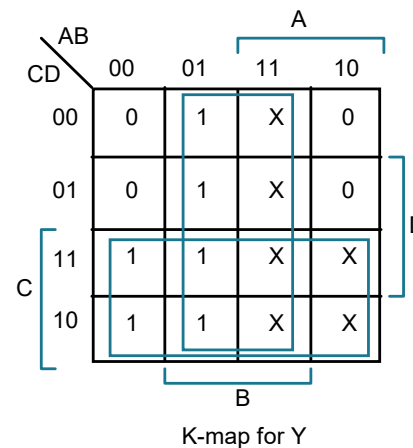
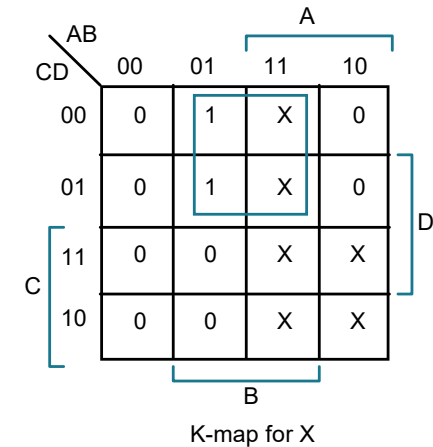
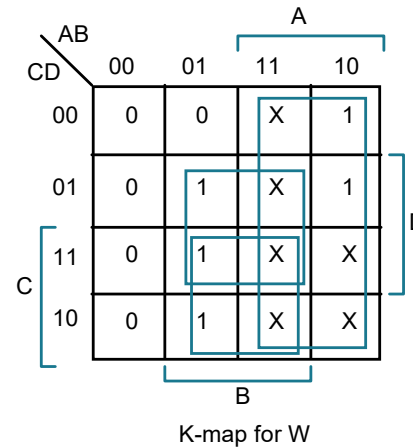
Ejemplo:



Programmable Array Logic (PAL)

Ejemplo de diseño: Conversor de código BCD a Gray

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:

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

$$X = B C'$$

$$Y = B + C$$

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

Programmable Array Logic (PAL)

Ejemplo de diseño: Conversor de código BCD a Gray

PAL programada

Funciones mínimas:

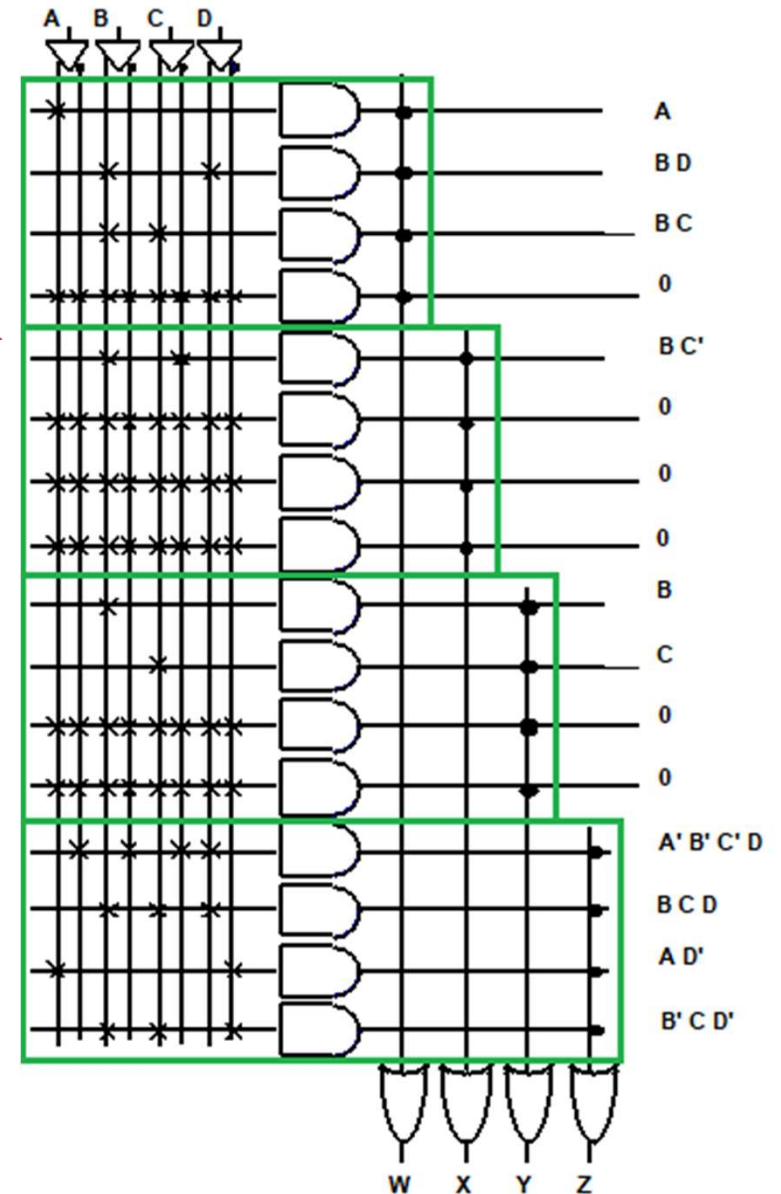
$$W = A + B D + B C$$

$$X = B C'$$

$$Y = B + C$$

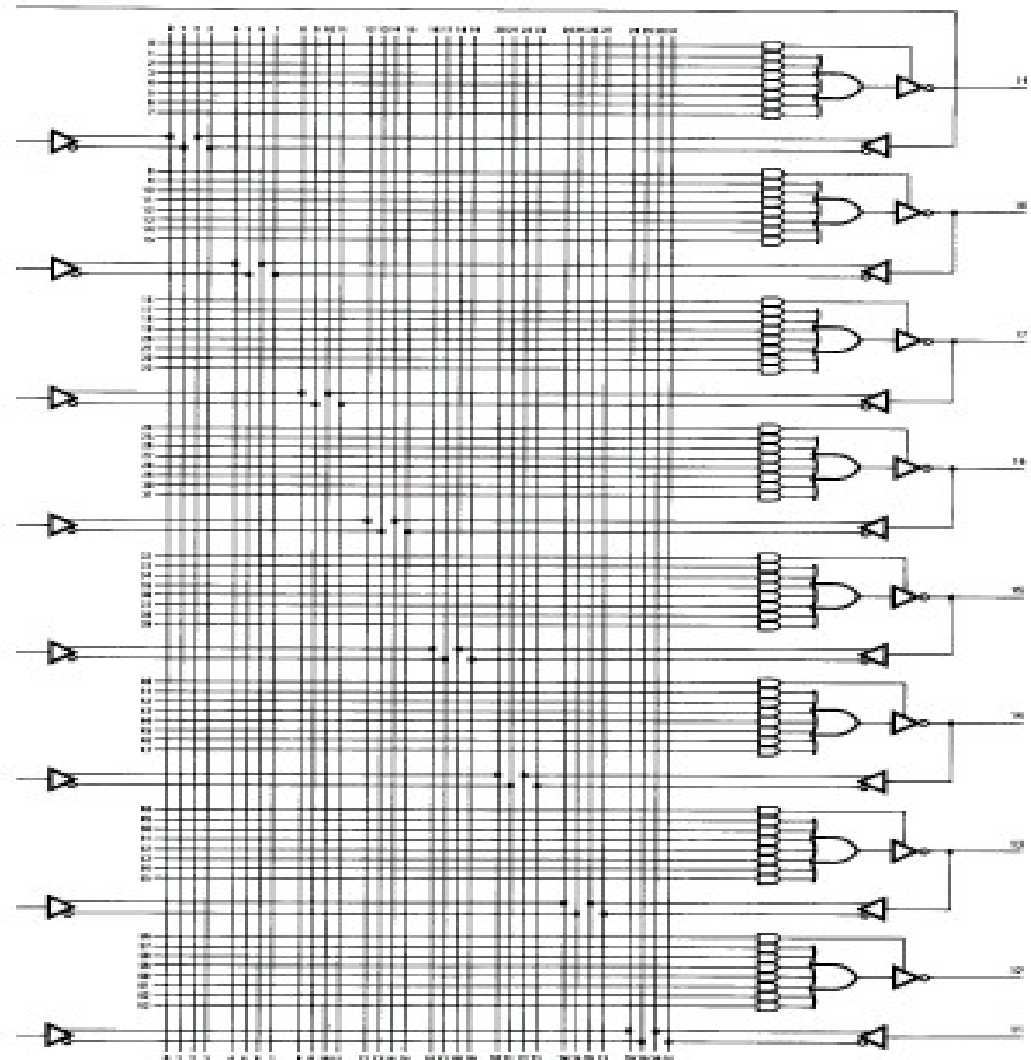
$$Z = A'B'C'D + B C D + A D' + B' C D'$$

4 términos de producto para cada compuerta **OR**



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 términos. Mayormente tienen la posibilidad de realimentar las salidas.



Comparación

PROM se programa a partir de los minitérminos de las funciones

A partir de la TV directamente, ya que el decodificador genera todos los minitérminos y sumo solo los que forman parte de la función.

PLA se programa a partir de la simplificación de las funciones

Se simplifican las funciones a implementar y a partir de allí se generan los términos producto.

PAL se programa a partir de la simplificación de las funciones

Se simplifican las funciones a implementar y a partir de allí se generan los términos producto considerando que debo tener como mínimo tantas AND como sumandos en cada función a implementar.

	Matriz AND	Matriz OR
PROM	Fija	Programable
PLA	Programable	Programable
PAL	Programable	Fija

Se pueden considerar a PROM y PAL como versiones restringidas de la PLA

La PROM es una PLA con arreglo AND fijo que genera todos los minitérminos y un arreglo OR configurable.

La PAL es una PLA con un arreglo OR fijo, y cada salida es un subconjunto de las AND.

Selección de acuerdo a las necesidades

Se observará las ecuaciones a implementar.

Tener en cuenta la cantidad de entradas y salidas.

Comparar costos.

La PROM es más fácil, pero utilizar PAL y PLA donde resulte posible

Las que incluyen biestables permiten realizar circuitos secuenciales sincrónicos.

Las herramientas CAD realizan diseño en base a especificaciones en lenguajes descriptores de hardware.

PALASM y ABEL elementales.

VHDL y Verilog avanzados (FPGA).

