TEMA 3

Dispositivos Lógicos Programables:
PAL, GAL, FPLA y EPLD. Programación
de los PLD

Dispositivos Lógicos Programables

"Programmable Logic Devices" o **PLDs**

Reemplazan a muchos dispositivos SSI

- ◊ **Ocupan menos espacio en tarjetas**
- ◊ **Reducen el número y el coste de dispositivos en un diseño**
- ◊ **Disminuyen las conexiones entre componentes**
  - **Ahorro de espacio**
  - **Conexionado interno fiable**
- ◊ **Simplifican las tareas de diseño (CAD)**

Clasificación

PROM (“**P**rogrammable **R**ead-**O**nly **M**emory”)

Almacenamiento de datos

Implementación de lógica combinacional

PLA (“**P**rogrammable **L**ogic **A**rray”)

Implementación de circuitos lógicos

FPLA (Programable por el usuario)

PAL (“**P**rogrammable **A**rray **L**ogic”)

Implementación de circuitos lógicos

GAL (“**G**eneric **A**rray **L**ogic”)

Implementación de circuitos lógicos

PLD (“**P**rogrammable **L**ogic **D**evice”)

Implementación de circuitos lógicos

EPLD (“Erasable PLD”)

CPLD (“Complex PLD”)

◊ **Programables por el usuario mediante equipos de bajo coste**

Matrices Programables

Matriz o "Array"

Red de conductores distribuidos en filas y columnas, con un 'fusible' en cada punto de cruce

El tipo más sencillo de matriz programable:

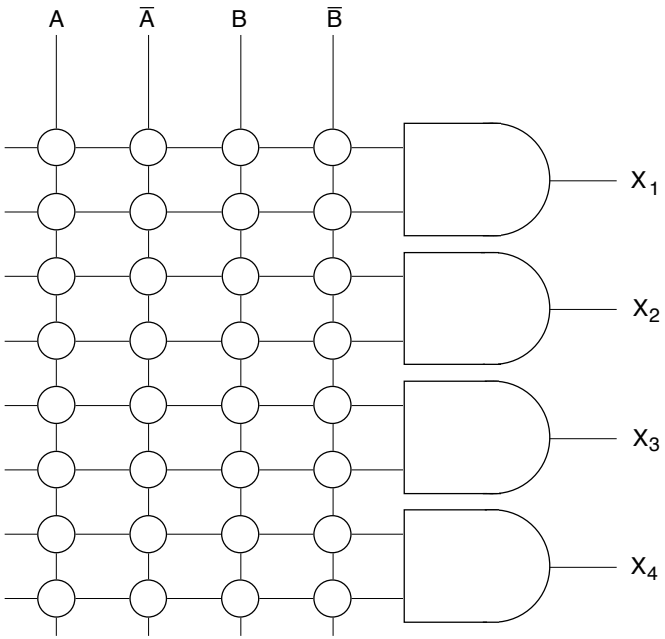
En cada punto de cruce las dos líneas están interconectadas mediante un diodo en serie con un fusible

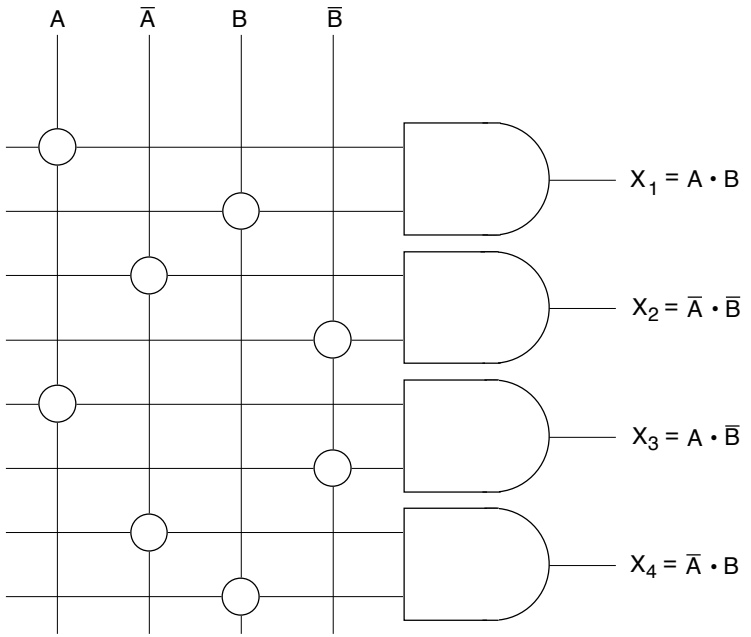
Matriz AND (OR)

Una serie de puertas AND (OR) cuyas entradas son algunas de las filas o de las columnas de una matriz programable

- **Antes de la programación cada entrada de cada puerta AND (OR) está conectada a todas las líneas de la matriz programable que se cruzan con ella**
- **Se programa fundiendo fusibles. Se eliminan selectivamente conexiones en los puntos de cruce de cada entrada de cada puerta AND (OR)**
- **Para cada entrada de una puerta AND (OR) sólo debe quedar intacto un fusible (cada entrada conectada a una línea de las que se cruzan con ella) como máximo**
- **Se programan los '0's**

Matriz AND



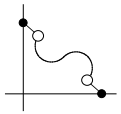


Sin programar

Todos los fusibles intactos

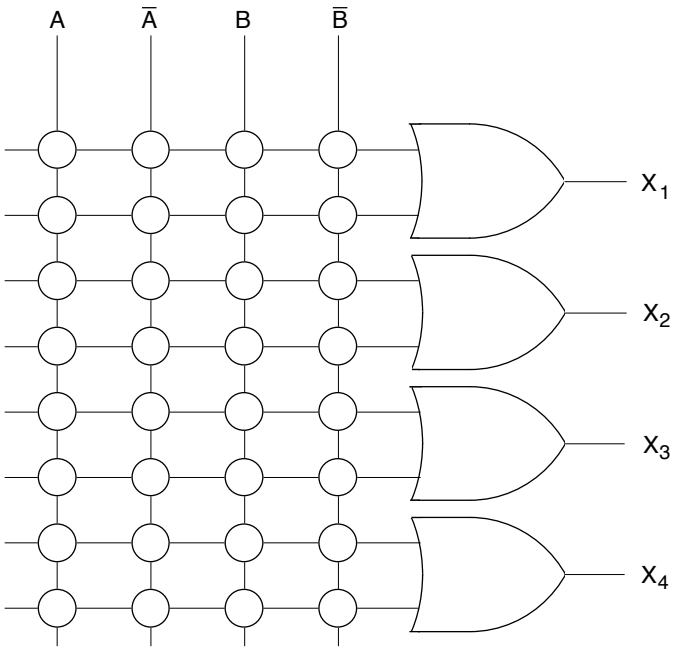
Programada

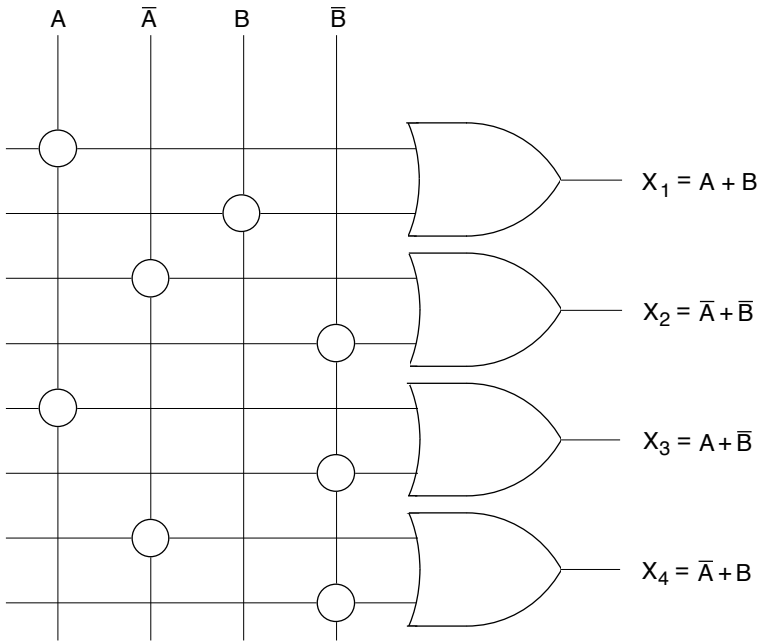
Algunos fusibles intactos (uno por entrada máximo)



Fusible

Matriz OR



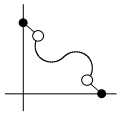


Sin programar

Todos los fusibles intactos

Programada

Algunos fusibles intactos (uno por entrada máximo)



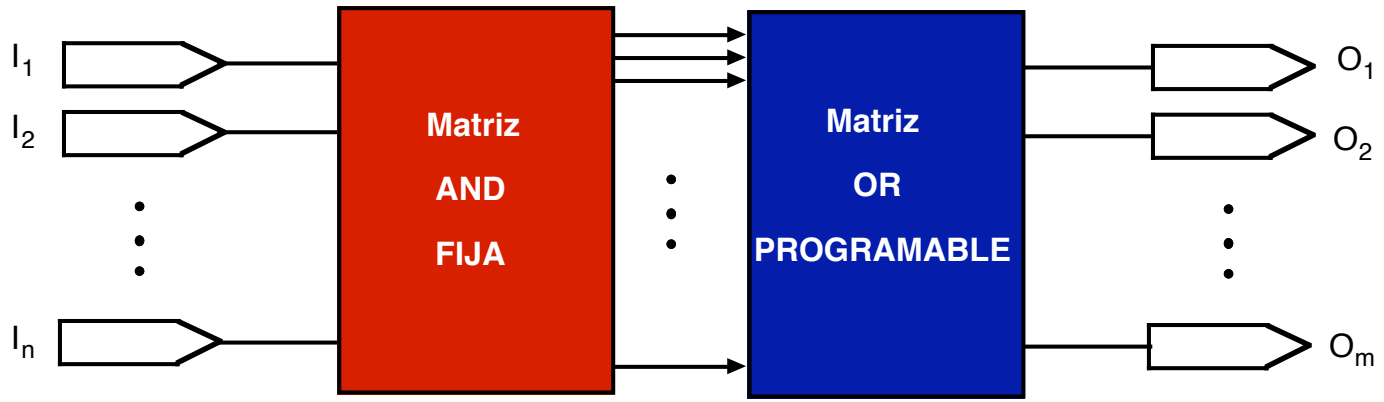
Fusible

PLDs: PROM

Clasificación por su arquitectura
(ordenación funcional de sus elementos internos)

PROM ("Programmable Read-Only Memory"

- ◊ **Primer nivel:** Conjunto fijo de puertas AND (Decodificador)
- ◊ **Segundo Nivel:** Matriz programable de puertas OR

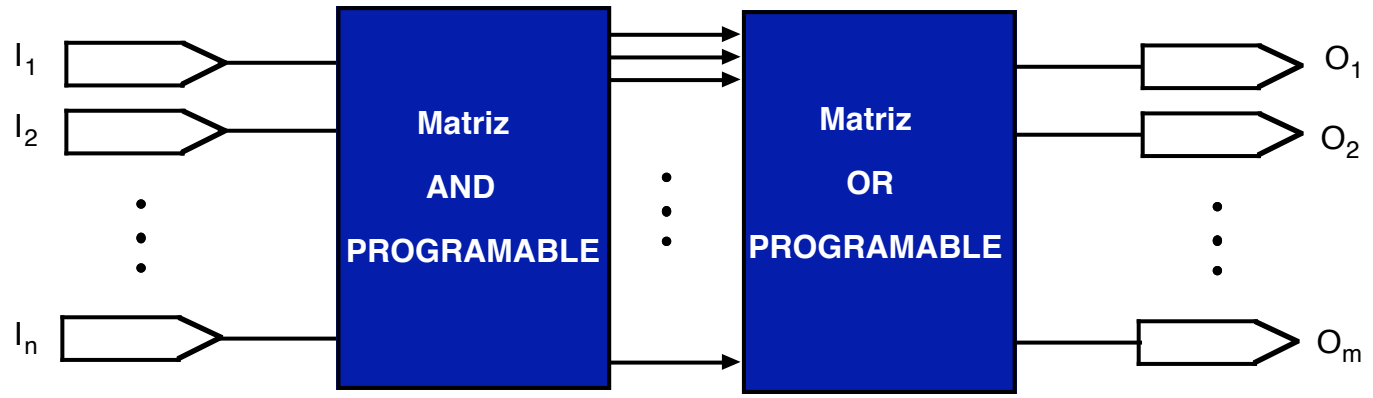


PLDs: FPLA

Clasificación por su arquitectura
(ordenación funcional de sus elementos internos)

PLA ("Programmable Logic Array")
FPLA ("Field Programmable Logic Array")

- **Primer nivel:** Conjunto programable de puertas AND (Decodificador)
- **Segundo Nivel:** Matriz programable de puertas OR



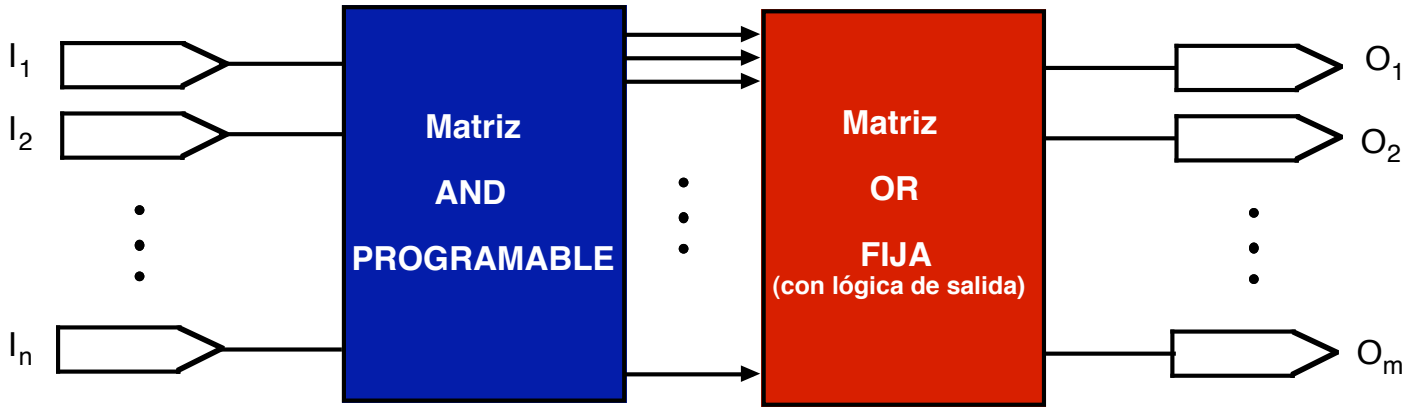
Buen aprovechamiento del área del chip (para aplicaciones lógicas)

PLDs: PAL

Clasificación por su arquitectura
(ordenación funcional de sus elementos internos)

PAL ("Programmable Array Logic")

- ◊ **Primer nivel:** Conjunto programable de puertas AND (Decodificador)
- ◊ **Segundo Nivel:** Matriz fija de puertas OR
- ◊ **Lógica de salida**



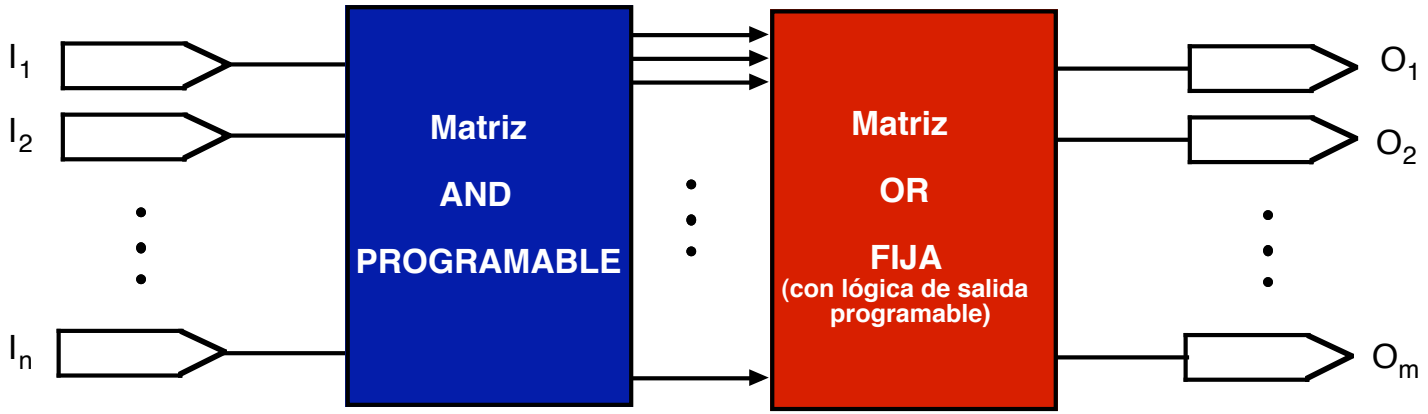
Buen aprovechamiento del área del chip (para aplicaciones lógicas)
Bajo retardo (un solo nivel programable, menos fusibles)

PLDs: GAL

Clasificación por su arquitectura
(ordenación funcional de sus elementos internos)

GAL ("Generic Array Logic")

- ◊ **Primer nivel:** Conjunto programable de puertas AND (Decodificador)
- ◊ **Segundo Nivel:** Matriz fija de puertas OR
- ◊ **Lógica de salida programable**

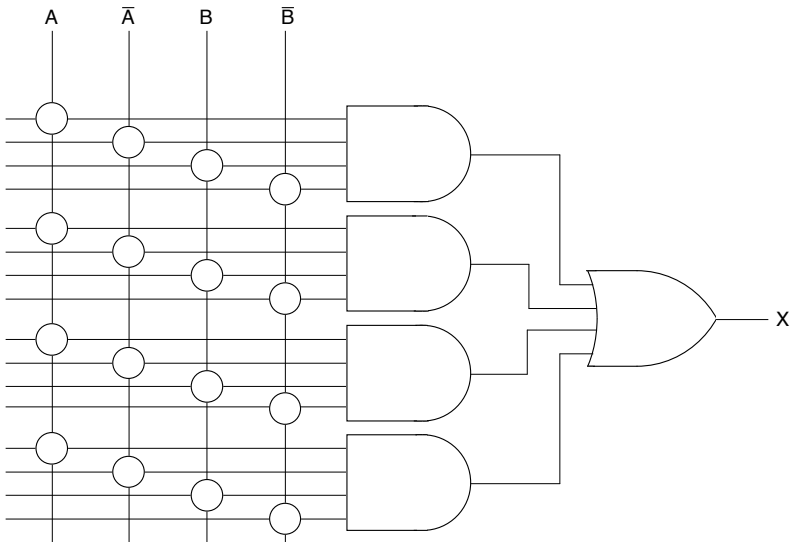


Buen aprovechamiento del área del chip (para aplicaciones lógicas)
Bajo retardo (un solo nivel programable, menos fusibles)
Reprogramables (E2CMOS)

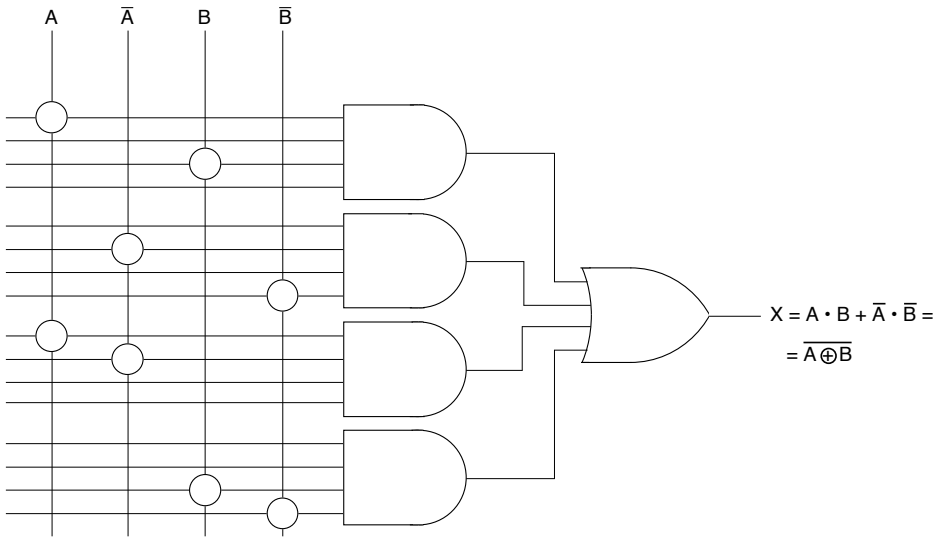
PAL

Estructura básica

- ◊ **Matriz AND programable**
- ◊ **Matriz OR fija**

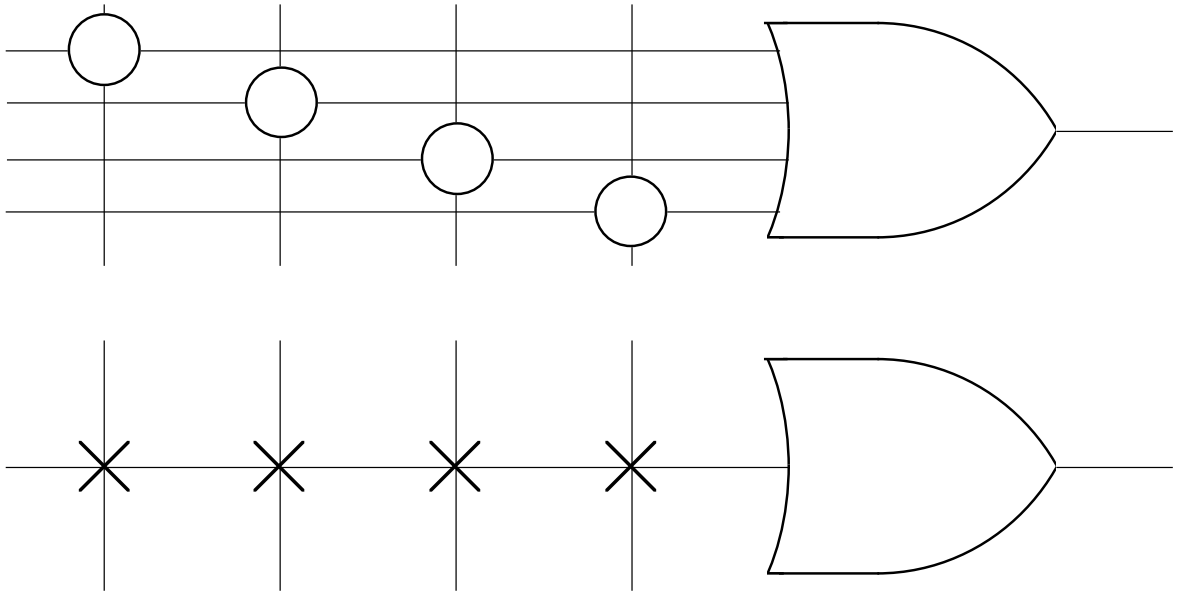


Estructura de una PAL



Implementación mediante una PAL de A EXNOR B

Representación simbólica

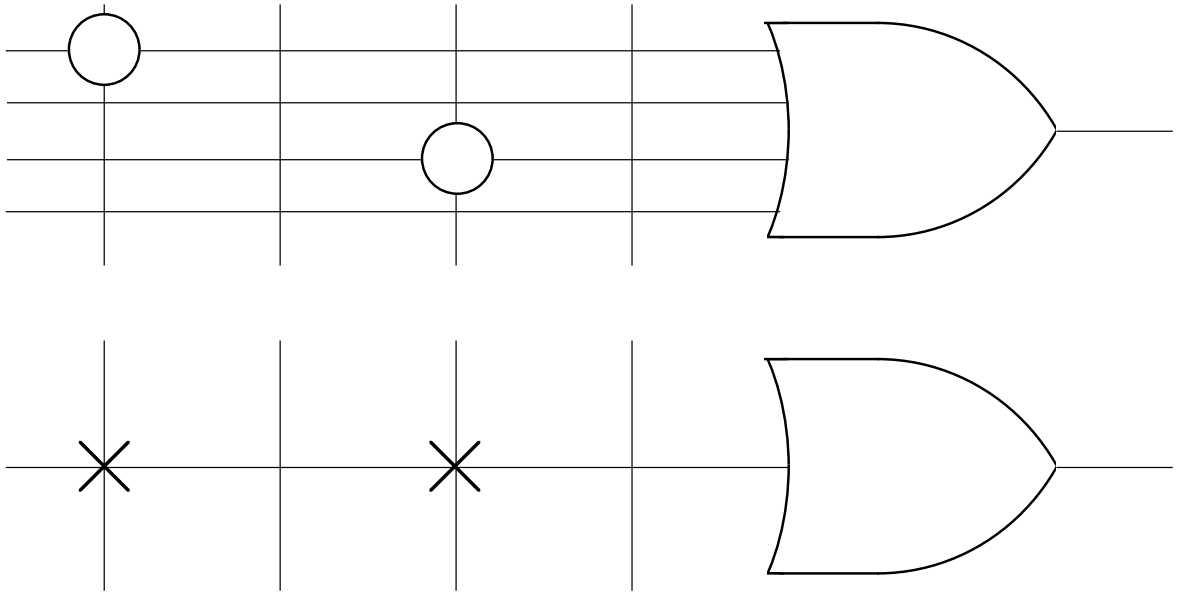


Sin programar

Todos los fusibles intactos

× **Fusible**

Representación simbólica

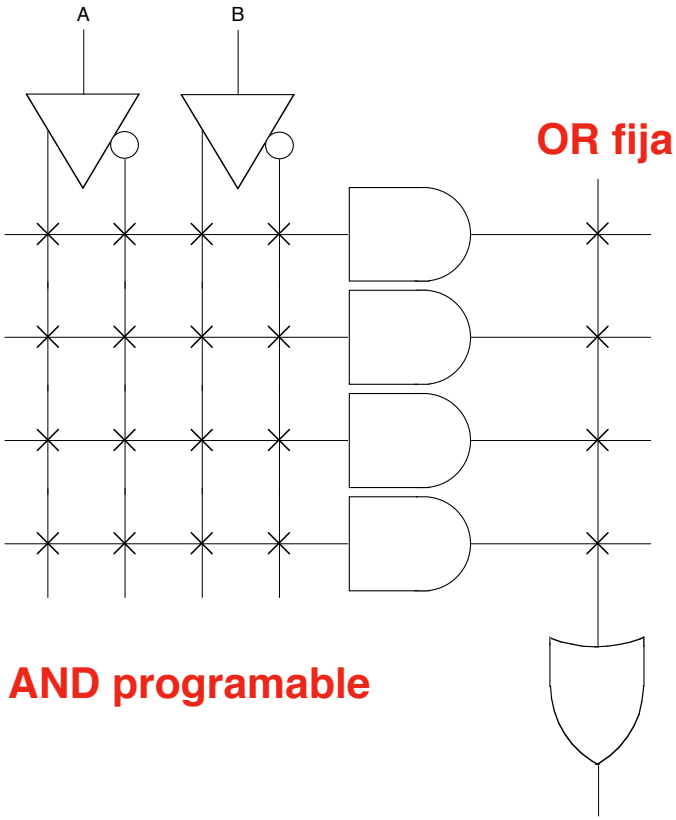


Programada

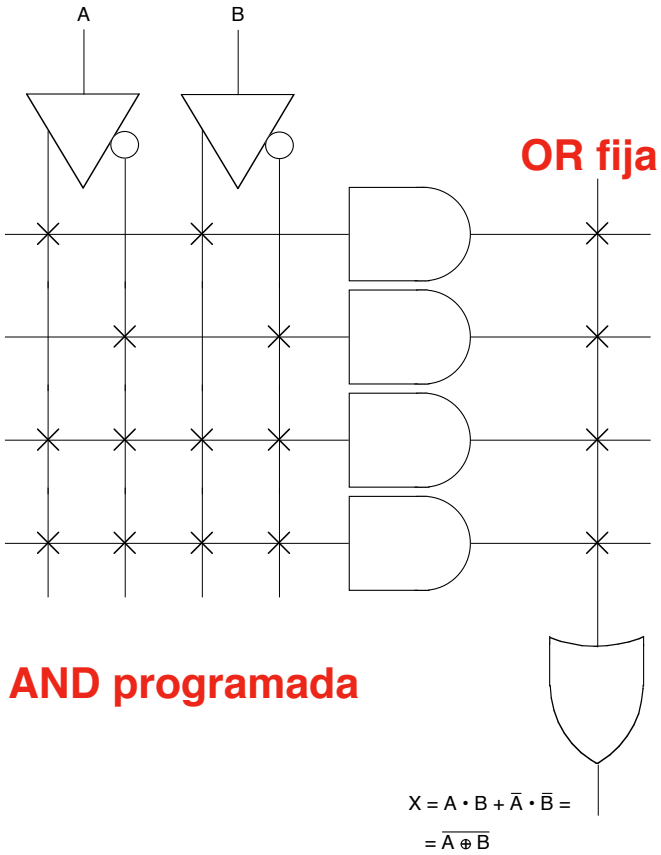
Algunos fusibles intactos

× **Fusible**

PAL

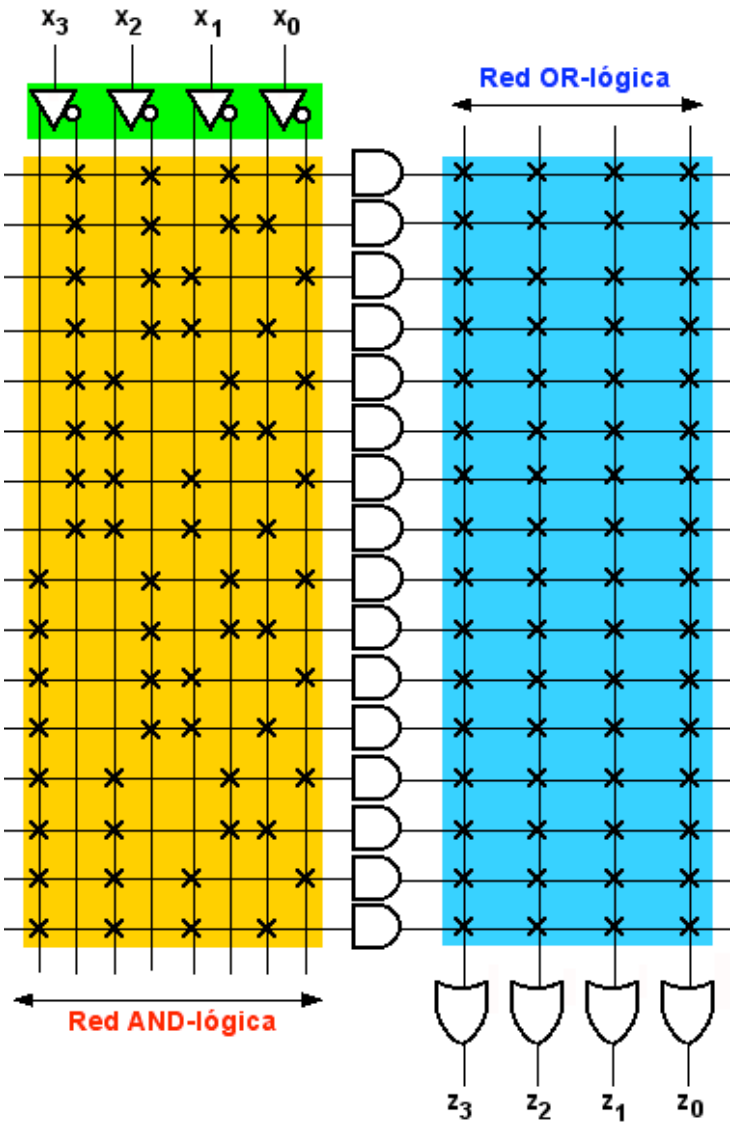


Estructura de una PAL



Implementación de A EXNOR B

PROM simbólica



AND fija

Todos los 2^n "minterms" que pueden formarse con las variables de entrada

Están implementados siempre, aunque no se requieran

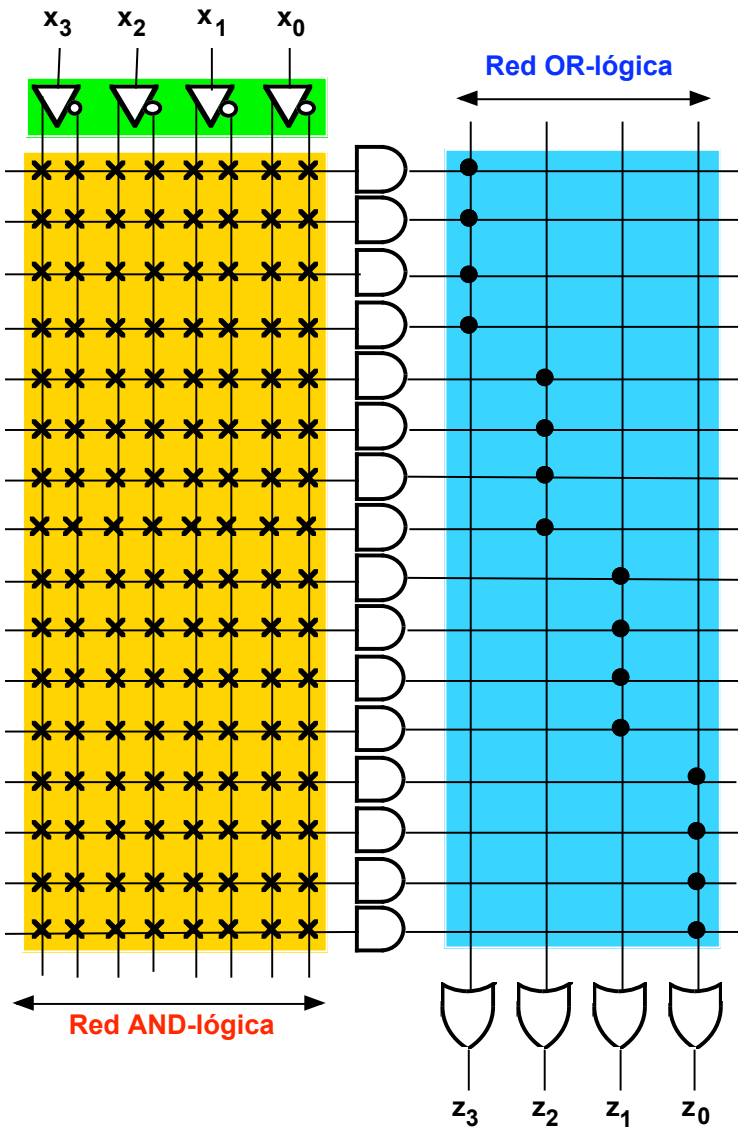
OR programable

Todos los "minterms" pueden ser compartidos por todas las funciones de salida

Los sumandos están disponibles siempre, aunque no se requieran

Todos los fusibles (área)

PAL



AND programable

Hasta 2^n términos producto que pueden formarse con las variables de entrada

Los productos están disponibles siempre, aunque no se requieran

Todos los fusibles (área)

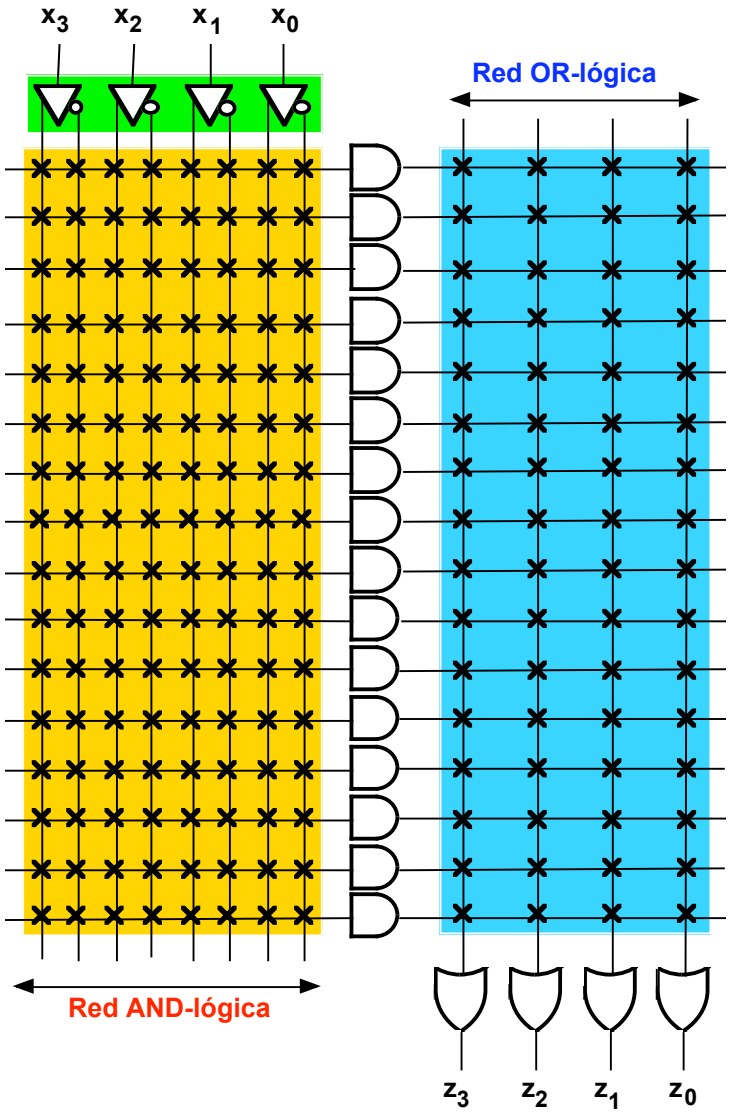
OR fija

Los términos producto no pueden ser compartidos por las funciones de salida

Los sumandos están disponibles siempre, aunque no se requieran

Limitado número de términos producto ($< 2^n$) disponibles por función

PLA (FPLA)



AND programable

Hasta 2^n términos producto que pueden formarse con las variables de entrada

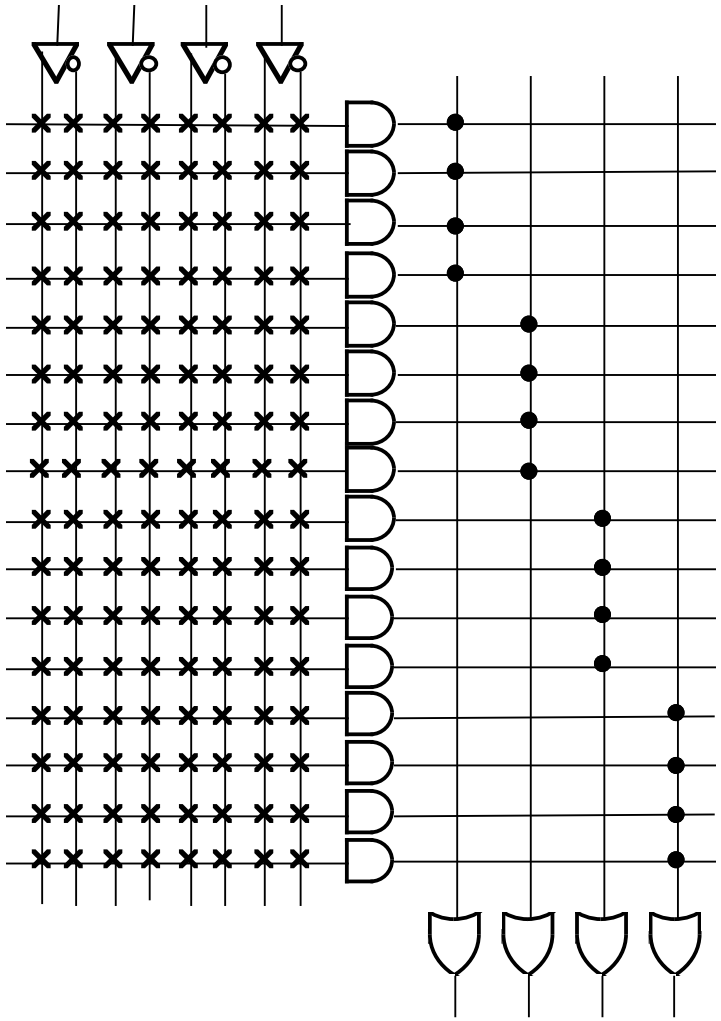
Los productos están disponibles siempre, aunque no se requieran

OR programable

Todos los términos producto pueden ser compartidos por todas las funciones de salida

Los sumandos están disponibles siempre, aunque no se requieran

Síntesis de funciones lógicas con PAL



Ejemplo

Implementar las funciones lógicas

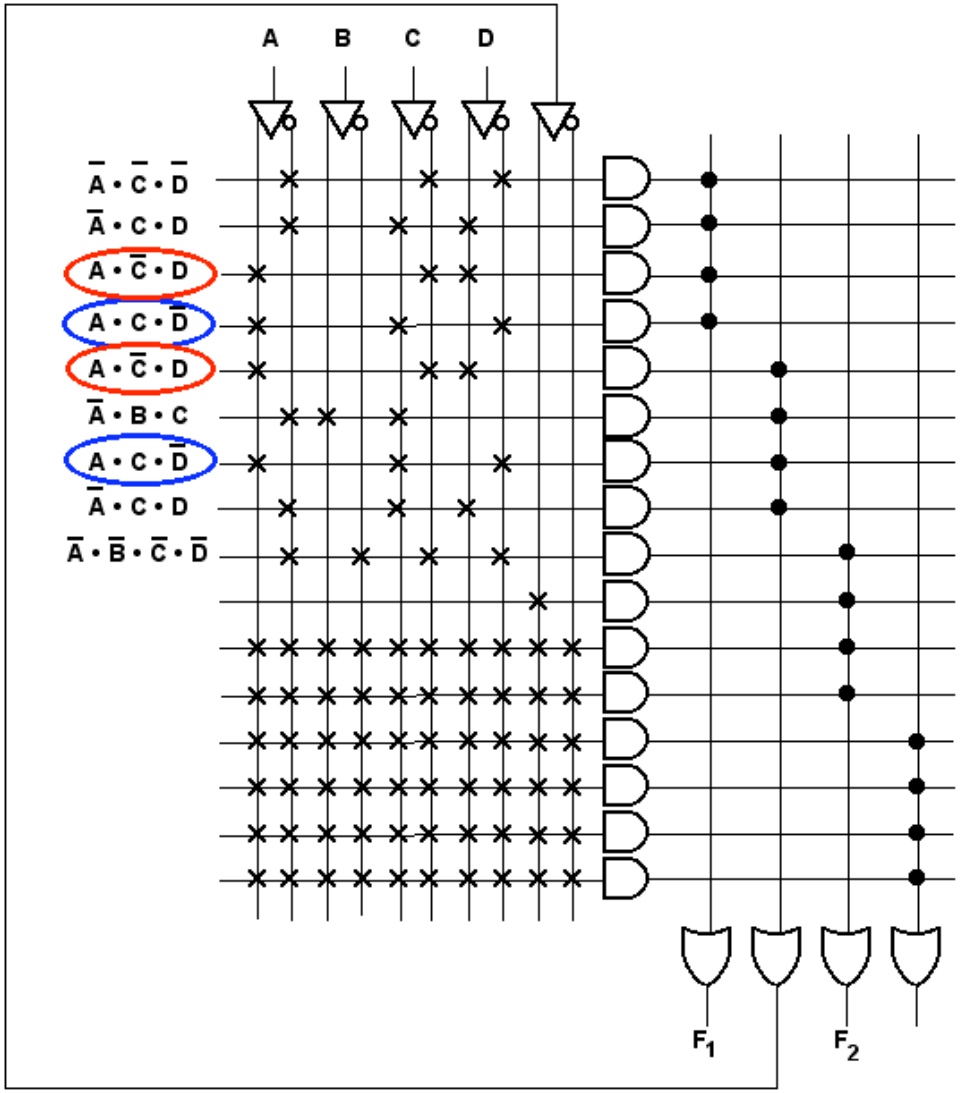
$$F_1 = /A \cdot /C \cdot /D + /A \cdot C \cdot D + A \cdot /C \cdot D + A \cdot C \cdot /D$$

$$F_2 = A \cdot /C \cdot D + /A \cdot B \cdot C + A \cdot C \cdot /D + /A \cdot C \cdot D + /A \cdot /B \cdot /C \cdot /D$$

Utilizando la PAL de la figura

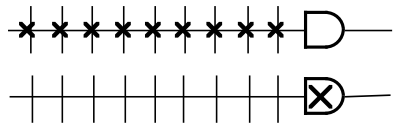
Las funciones están expresadas en la forma SOP mínima

Síntesis de funciones lógicas con PAL (2)

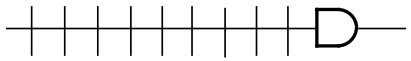


Ejemplo

Salida AND a ‘0’ lógico (L)



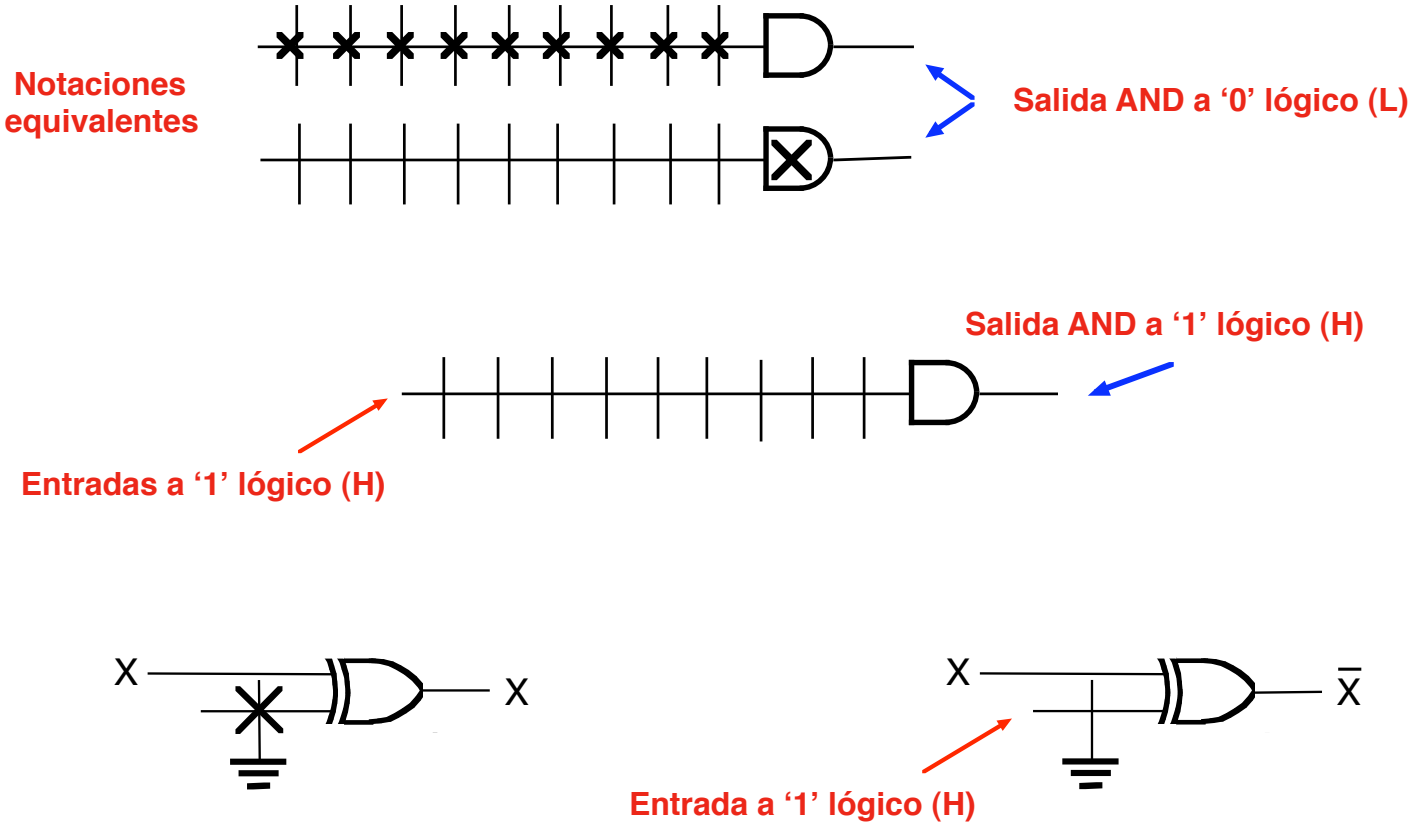
Notaciones equivalentes



Salida AND a ‘1’ lógico (H)

Dos funciones no pueden compartir términos producto en una PAL

Notación simbólica y tecnología



Síntesis lógica con PROM, PAL y PLA

Ejercicio

Dadas las 4 funciones lógicas:

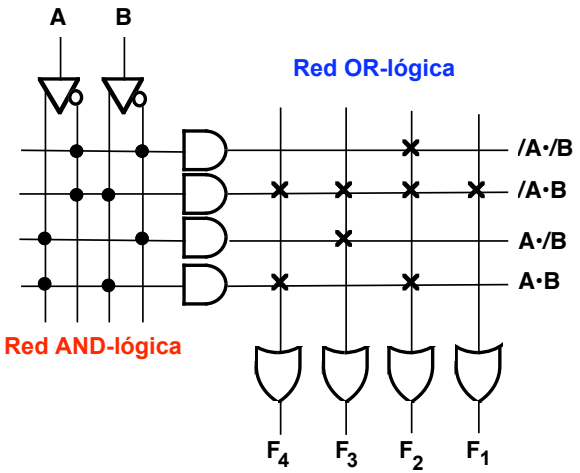
$F_1 = /A \cdot B$
$F_2 = /A + B$
$F_3 = A \text{ EXOR } B$
$F_4 = B$

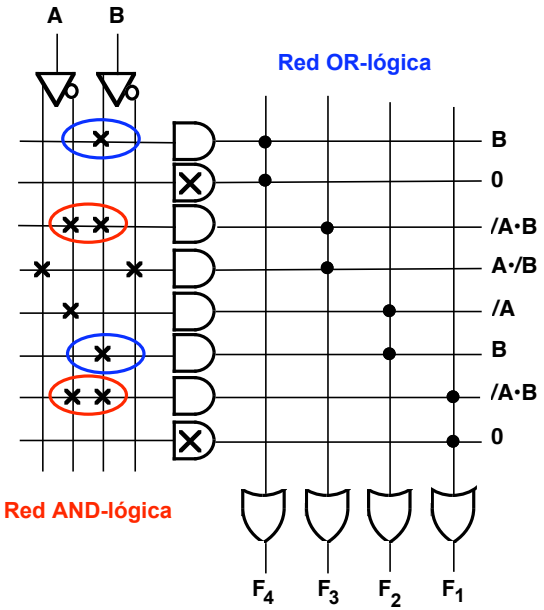
Mostrar su implementación mínima mediante PROM, PAL y PLA.

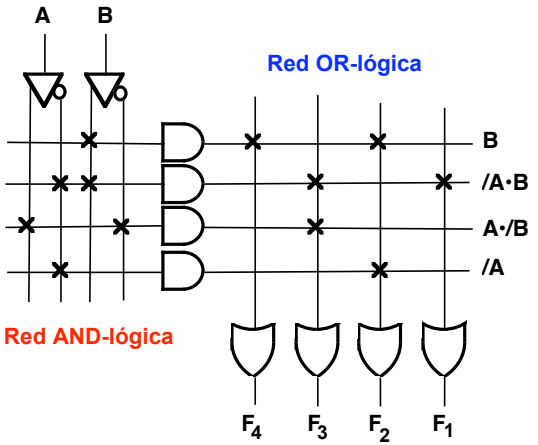
Con referencia a las implementaciones PAL y PLA ¿cuál de ellas requiere menor número de ANDs?¿cuál requiere menor número de fusibles intactos?

Síntesis lógica con PROM, PAL y PLA (2)

Ejercicio

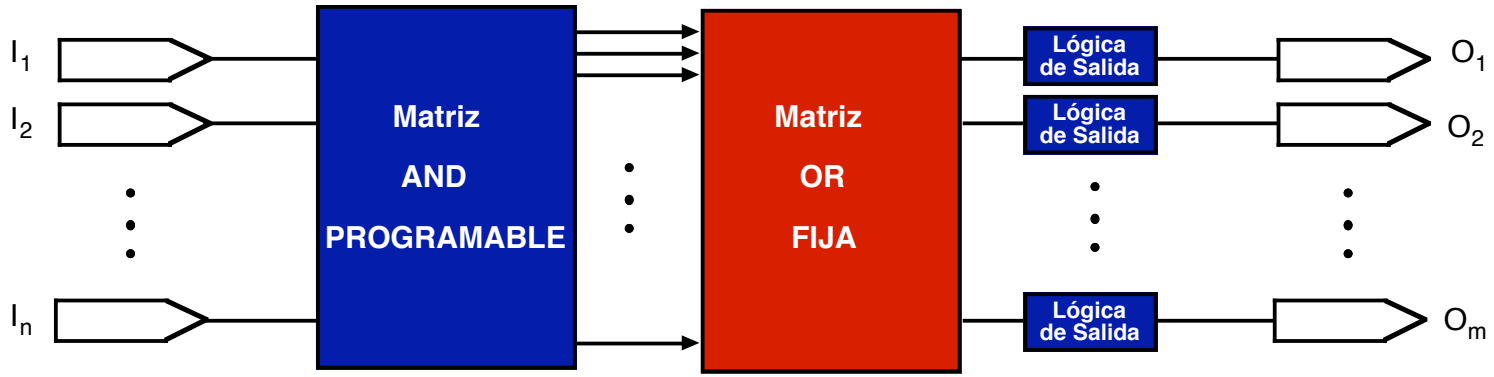






PROM **PAL** **PLA**

Diagrama de bloques de una PAL



PAL10L8

#Entradas (máx)

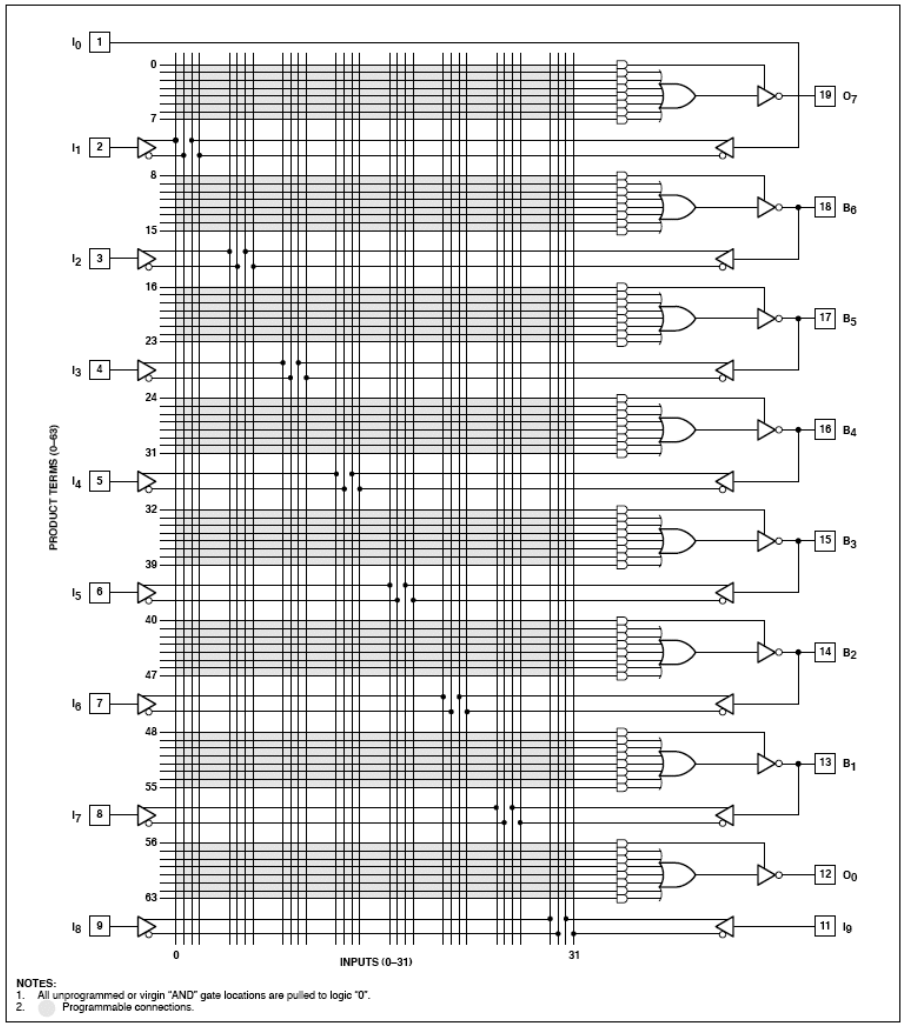
#Salidas (máx)

Salidas activas a nivel L (L, H, P, X, …)

PAL16R8

Salidas vía Flip-Flop y realimentación interna

Diagrama Lógico de una PAL



PAL16L8

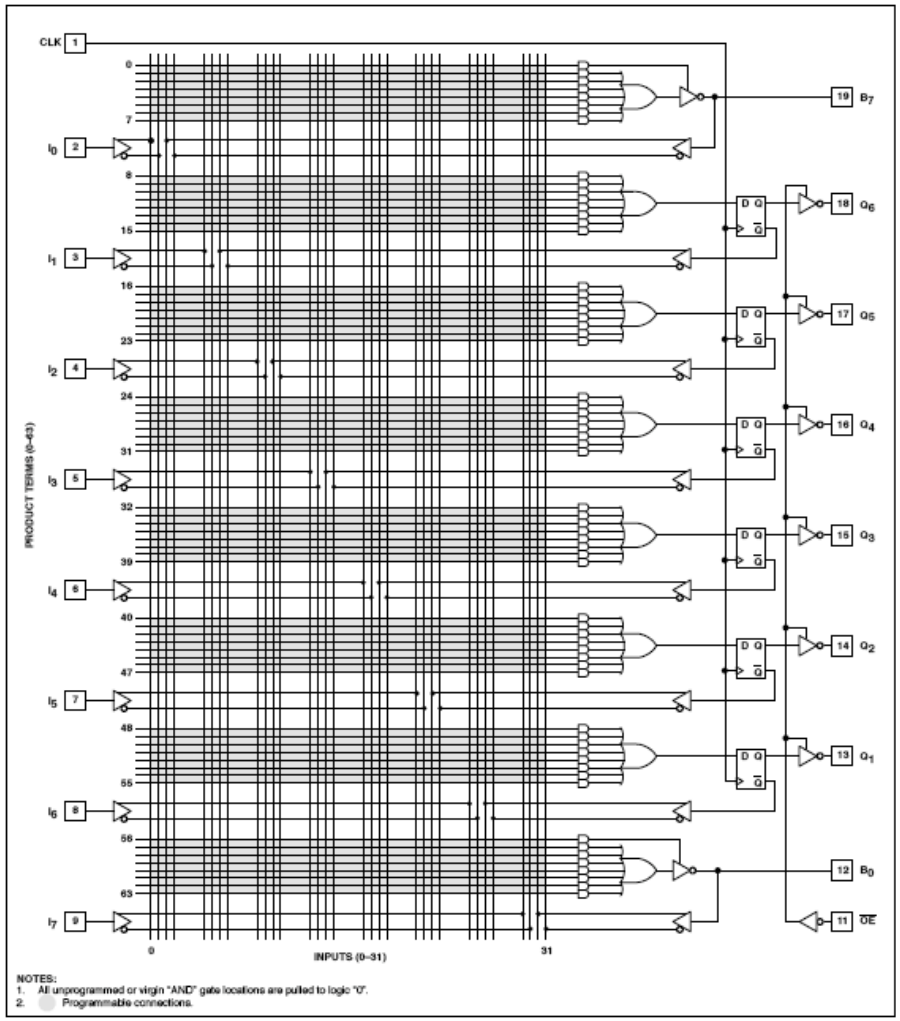
Hata 16 'pins' utilizables como entradas

Hasta 8 Salidas

Salidas activas a nivel L

- **8 bloques**
- **64 términos producto (7 utilizables como entradas de la OR por bloque)**
- **/OE programable (término producto) Individual para cada bloque**
- **Realimentación Salida-Zona de fusibles**

Diagrama Lógico de una PAL (2)



PAL16R6

Hata 16 'pins' utilizables como entradas (6 de realimentación vía Flip-flop)

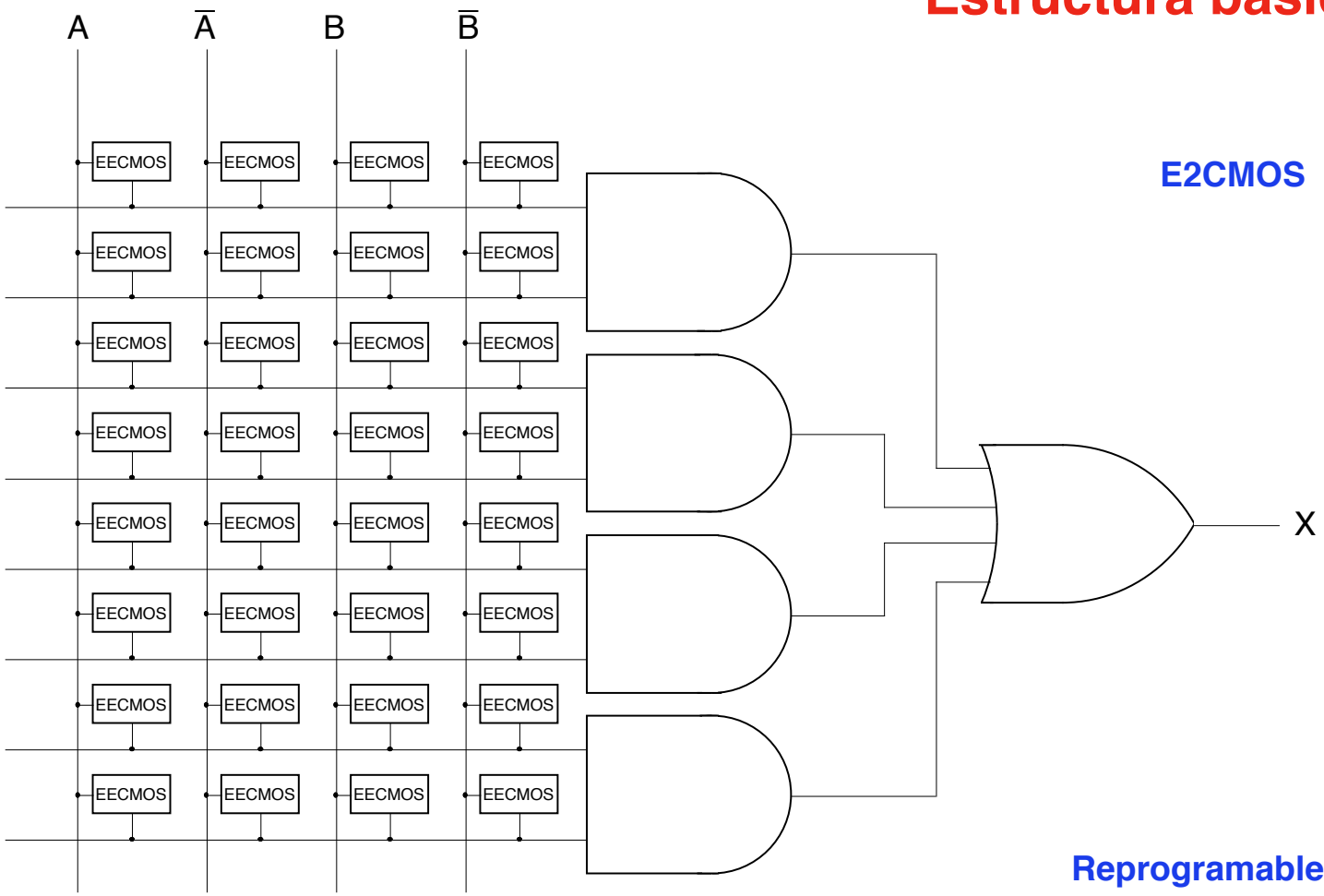
Salidas vía Flip-flop

Hasta 6 Salidas vía Flip-flop

- **8 bloques (2 con salida combinacional)**
- **6 bloques con salida vía Flip-flop**
- **64 términos producto (7 utilizables como entradas de la OR por bloque combinacional y 8 por bloque secuencial)**
- **/OE programable (término producto) individual para cada bloque combinacional**
- **/OE común para todas las salidas secuenciales ('pin' dedicado)**
- **CLK común para todos los Flip-flops ('pin'dedicado)**
- **Realimentación Salida-Zona de fusibles**

GAL

Estructura básica

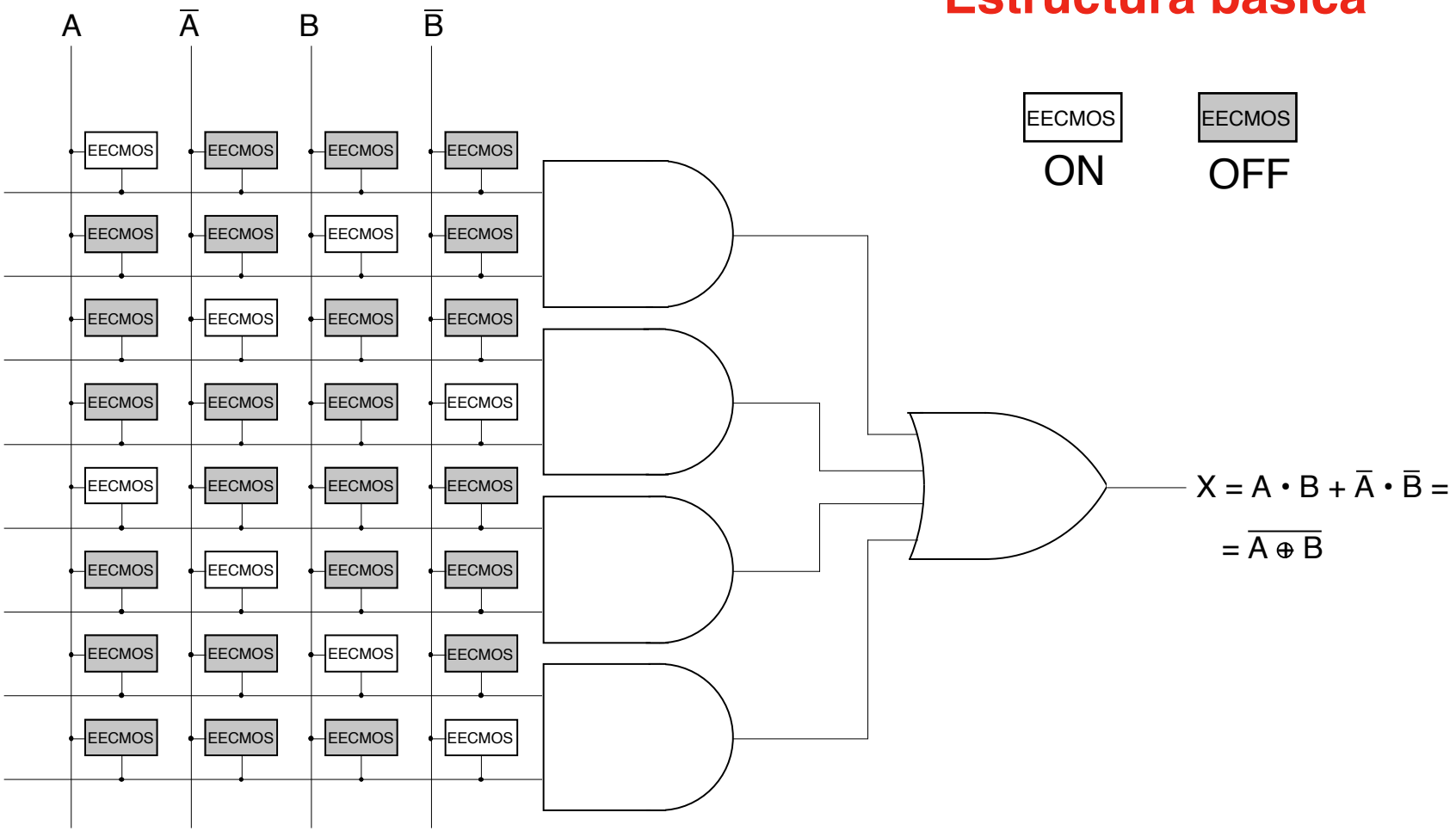


E2CMOS

Reprogramable

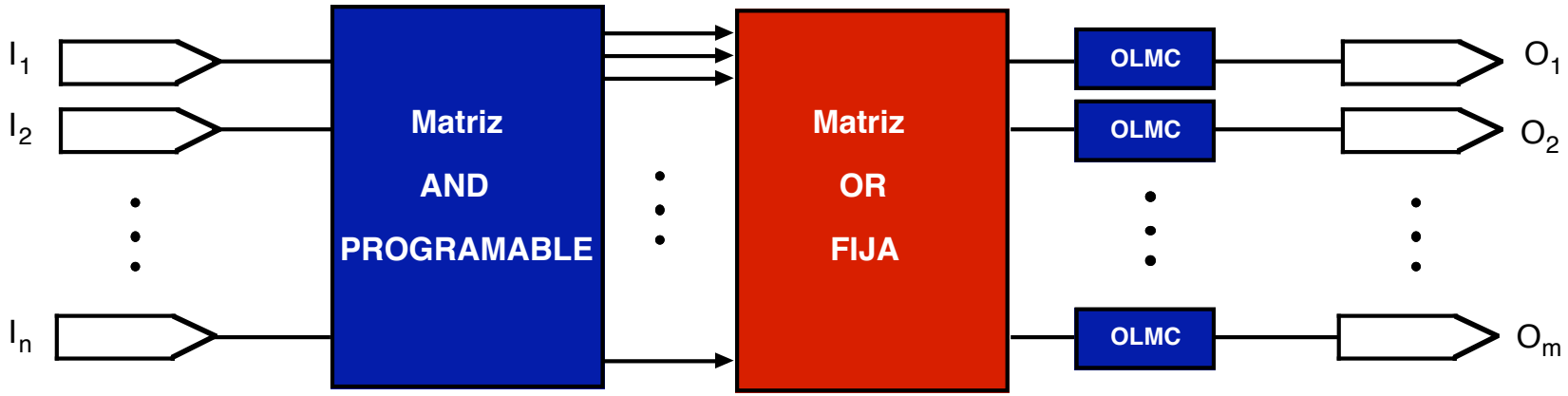
GAL (2)

Estructura básica



Implementación mediante una GAL de A EXNOR B

Diagrama de bloques de una GAL



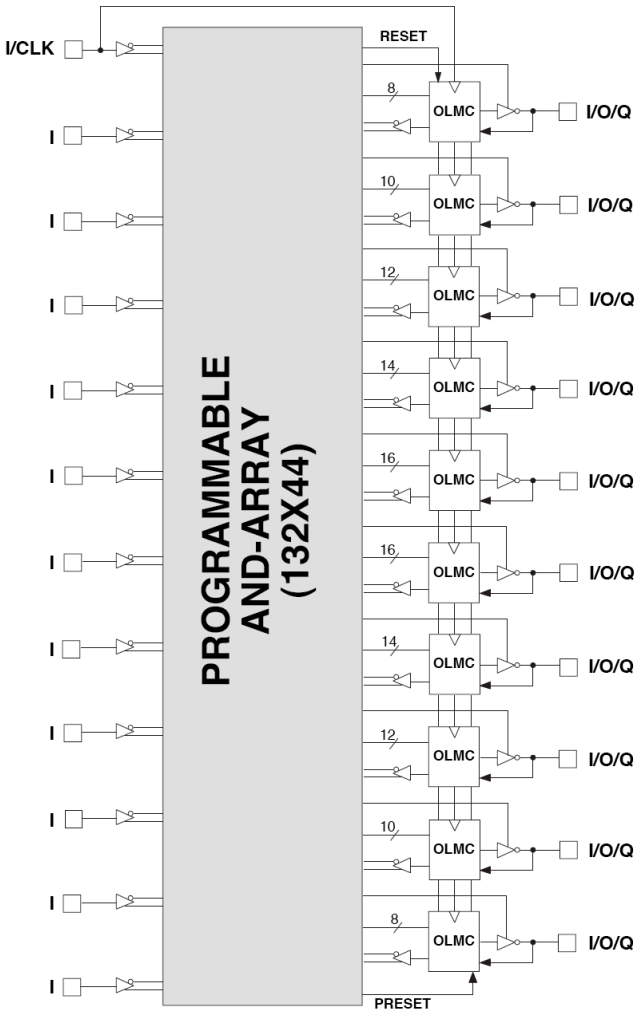
OLMC: "Output Logic Macrocell"

GAL16V8

#Entradas (máx) **#Salidas (máx)**

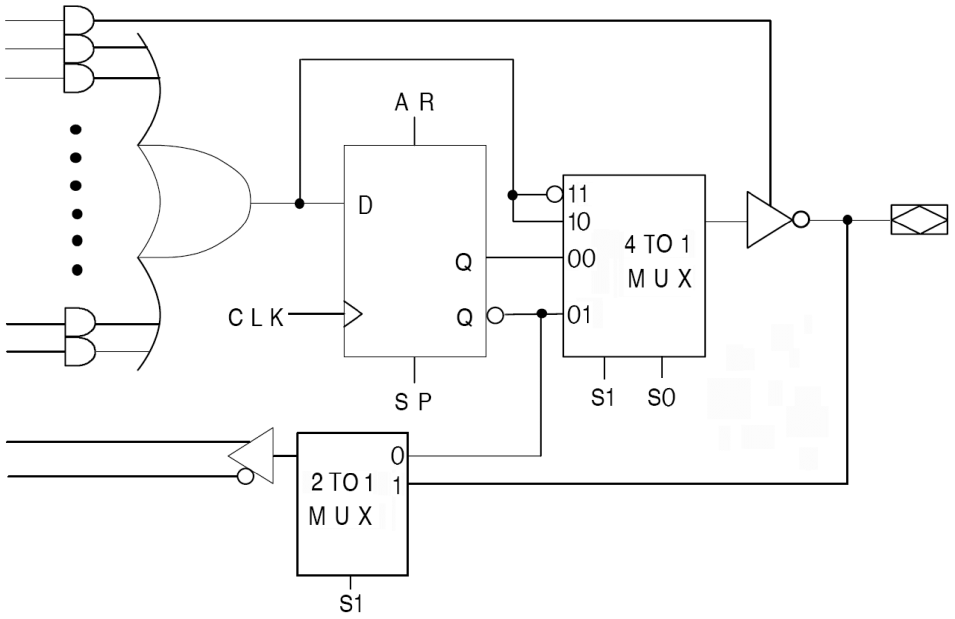
Configuración de salida variable

Diagrama de bloques funcional de las GAL



GAL22V10

OLMC: "Output Logic Macrocell"



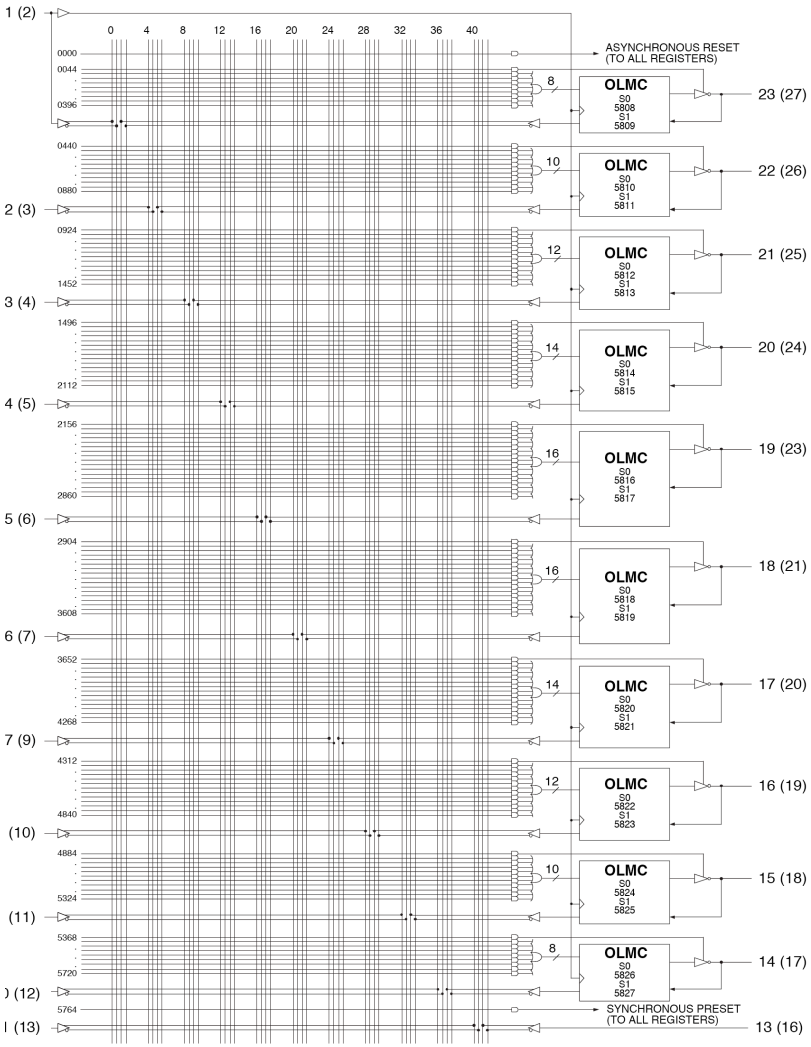
Hasta 22 entradas

Hasta 10 salidas (I/O)

SET (síncrono) y RESET (asíncrono) programables comunes a todas las OLMC

Reloj común a todas las OLMC

Mapa de fusibles de las GAL



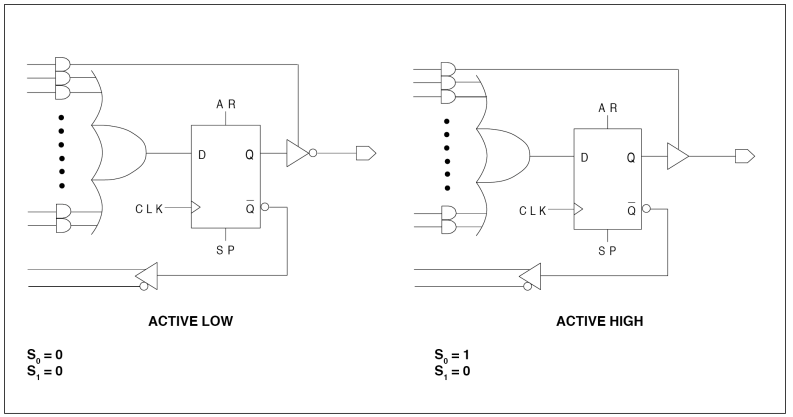
AND: 44 entradas

OR: 8, 10, 12, 14, 16, 16, 14, 12, 10, 8 entradas

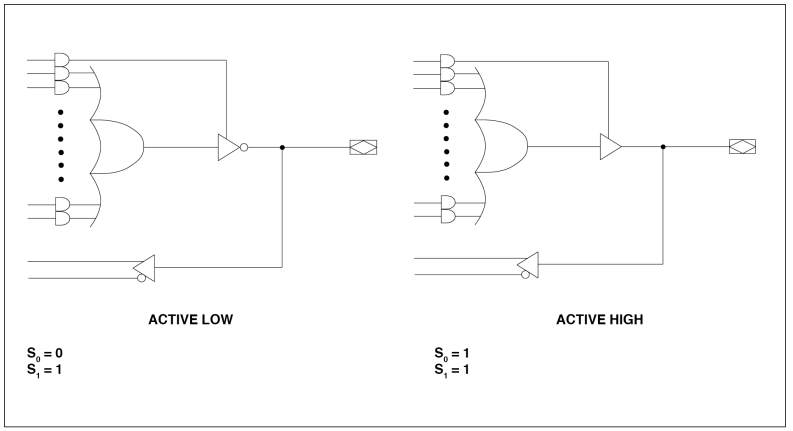
GAL22V10

2 MODOS: Combinacional y Registrado

Registered Mode



Combinatorial Mode



OLMC

Síntesis de una función lógica sobre GAL

Ejemplo implementación sobre una GAL

Mostrar una implementación de la función X, expresada en la forma SOP:

X = A·B·C·D·E·F + A·/B·C·/D·E·/F + /A·B·/C·D·/E·F + /A·B·C·D·E·/F +
+ A·/B·/C·/D·/E·F + /A·/B·/C·D·E·F + /A·/B·/C·/D·/E·/F

en una GAL22V10.

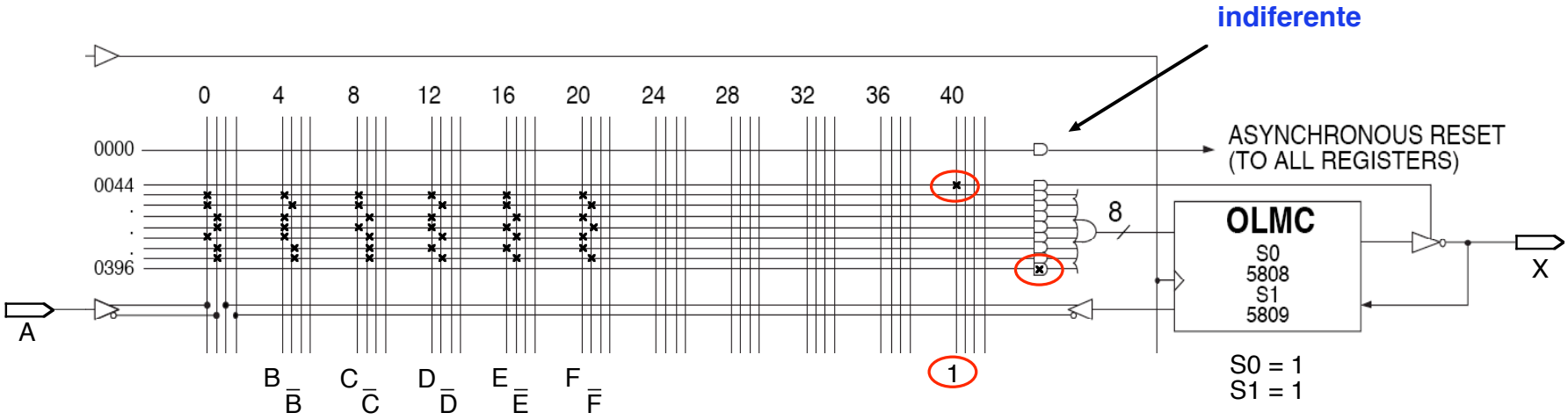
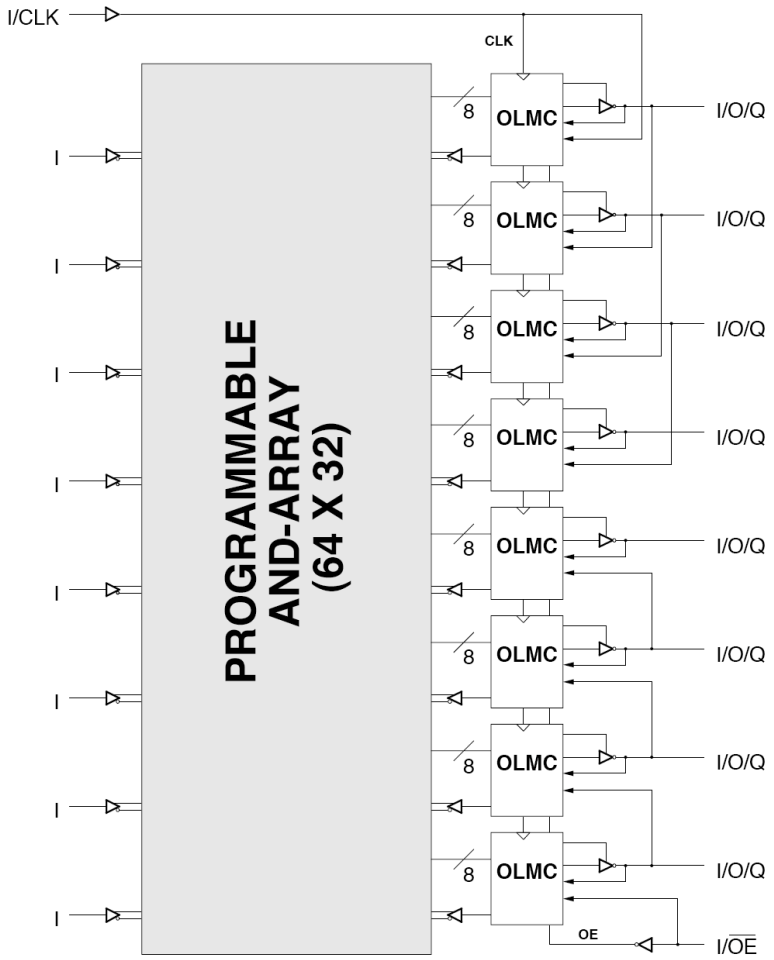


Diagrama de bloques funcional de las GAL

GAL16V8



Destinada a emular a gran parte de las PAL de igual o menor número de macroceldas

8 macroceldas

Hasta 16 entradas (+ CLK y /OE dedicadas)
Hasta 8 salidas

ANDs 32 entradas

ORs 7 u 8 entradas

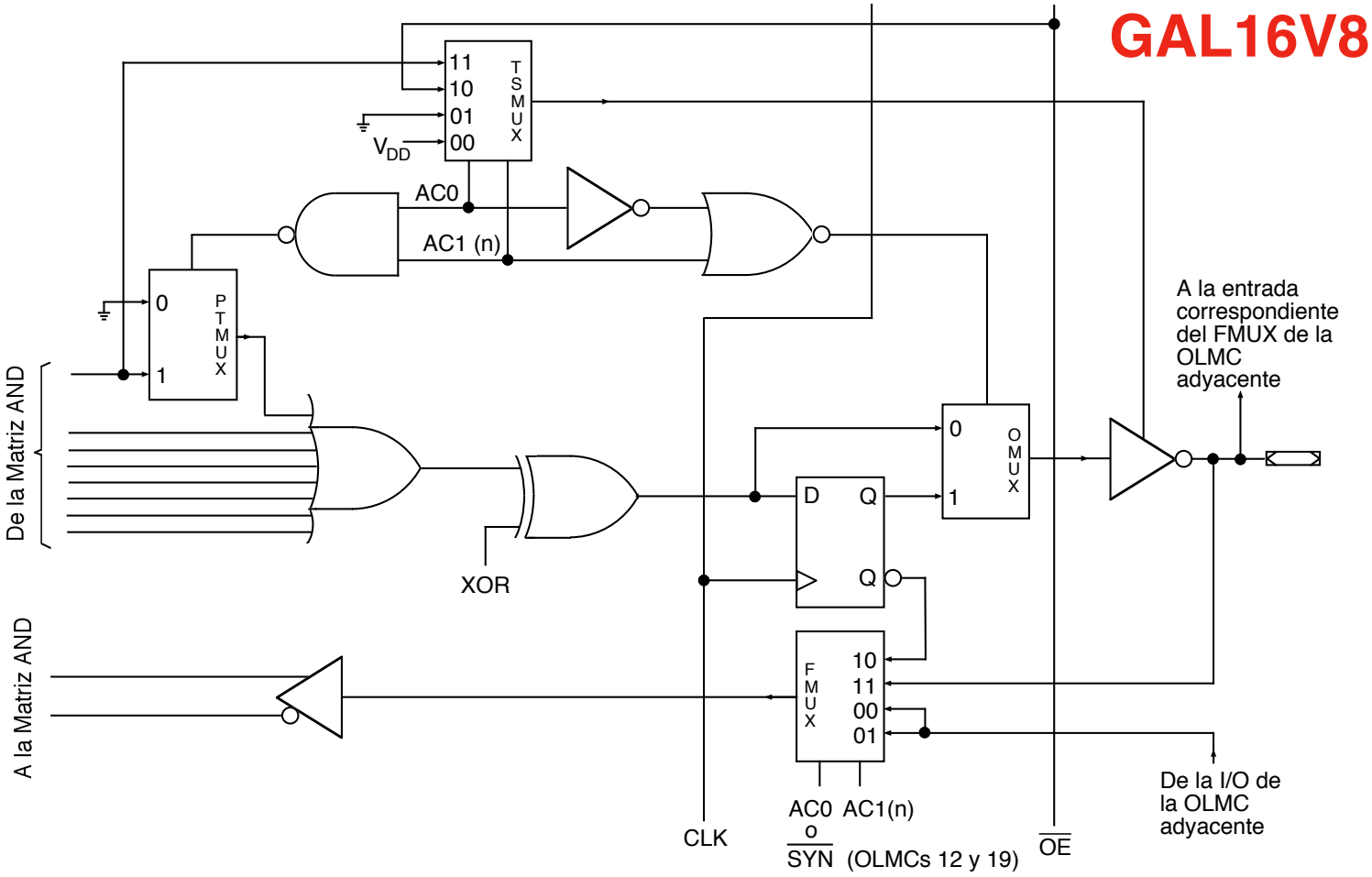
OLMC: Configuración de salida variable

3 MODOS de operación:

- **Simple**
- **Complejo**
- **Registrado**

El mismo MODO para todas las OLMC

GAL 16V8: OLMC



SYN, AC0, CLK, OE: globales (para todas las OLMC)

XOR, AC1 (n): locales (independientes para cada OLMC)

GAL 16V8: MODOS de operación

GAL16V8

SYN, AC0: **Controles globales. Configuran el MODO (igual para todas las OLMC).**
No todas las patillas se configuran igual en un determinado modo

Simple (‘10’)	**Salida combinacional dedicada (siempre activa) o Entrada dedicada**
Complejo (‘11’)	**Salida combinacional o Entrada/Salida combinacional**
Registrado (‘01’)	**Salida vía registro dedicada o Entrada/Salida combinacional**

XOR, AC1(n): **Controles locales**

XOR en cada AC1(n)	**Selecciona la polaridad de la salida modo (‘0’: activa baja, ‘1’: activa alta) Controla la configuración de Entrada/Salida en cada modo**

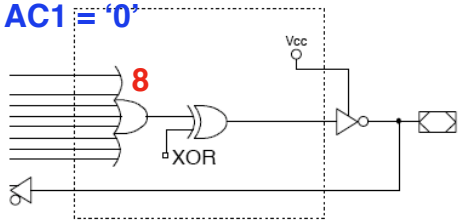
Diagrama de bloques de una GAL 16V8

Modo Simple

GAL10V8

SYN, AC0 = ‘1, 0’

AC1 = ‘0’

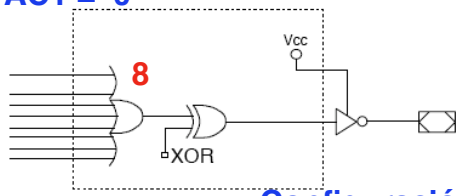


Combinatorial Output with Feedback Configuration for Simple Mode

- SYN=1.
- AC0=0.
- XOR=0 defines Active Low Output.
- XOR=1 defines Active High Output.
- AC1=0 defines this configuration.
- All OLMC **except** pins 15 & 16 can be configured to this function.

Excepto patillas 15 y 16

AC1 = ‘0’

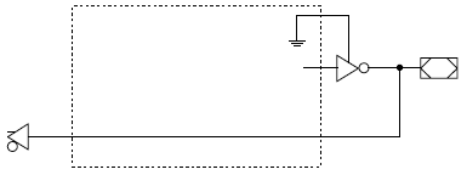


Combinatorial Output Configuration for Simple Mode

- SYN=1.
- AC0=0.
- XOR=0 defines Active Low Output.
- XOR=1 defines Active High Output.
- AC1=0 defines this configuration.
- Pins 15 & 16 are permanently configured to this function.

Configuración permanente patillas 15 y 16

AC1 = ‘1’



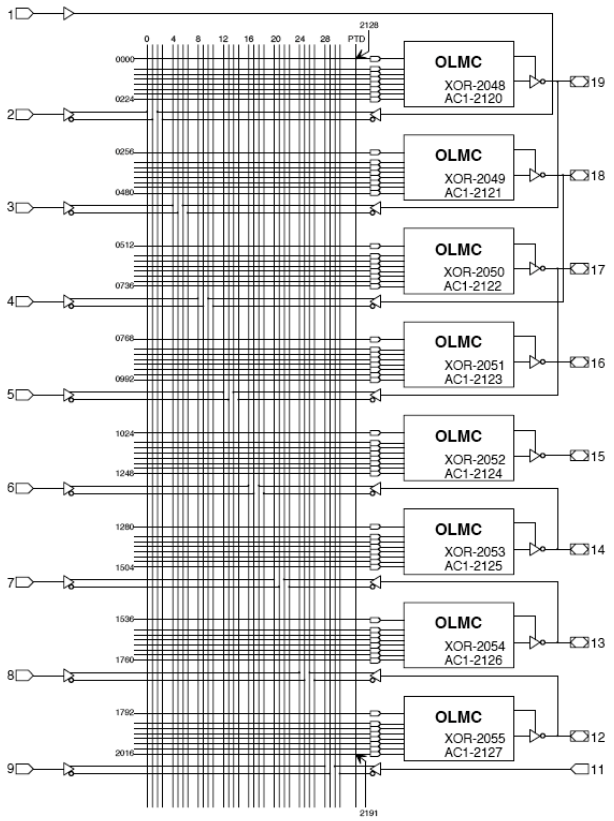
Dedicated Input Configuration for Simple Mode

- SYN=1.
- AC0=0.
- XOR=0 defines Active Low Output.
- XOR=1 defines Active High Output.
- AC1=1 defines this configuration.
- All OLMC **except** pins 15 & 16 can be configured to this function.

Excepto patillas 15 y 16

Simple Mode Logic Diagram

DIP & PLCC Package Pinouts



64-USER ELECTRONIC SIGNATURE FUSES

2056, 2057, ...	.... 2118, 2119
Byte 7 | Byte 6 ...	.... Byte 1 | Byte 0

M S B L S B

Hasta 2 Salidas y hasta 6 I/Os

La entrada o realimentación proviene de la OLMC vecina. Con SYN = ‘1’ y AC0 = ‘0’ todos los FMUXs seleccionan la señal que proviene de la OLMC (o entrada, patillas 11 y 19) adyacente

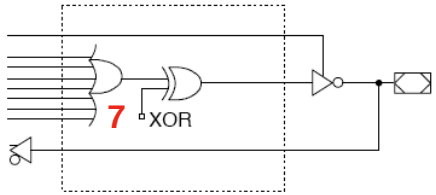
Diagrama de bloques de una GAL 16V8

Modo Complejo

GAL10V8

SYN, AC0 = ‘1, 1’

AC1 = ‘1’

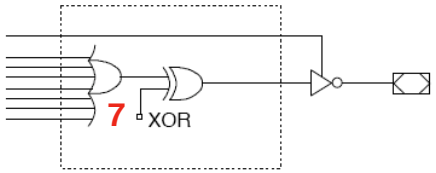


Combinatorial I/O Configuration for Complex Mode

- SYN=1.
- AC0=1.
- XOR=0 defines Active Low Output.
- XOR=1 defines Active High Output.
- AC1=1.
- Pin 13 through Pin 18 are configured to this function.

Excepto patillas 12 y 19

AC1 = ‘1’

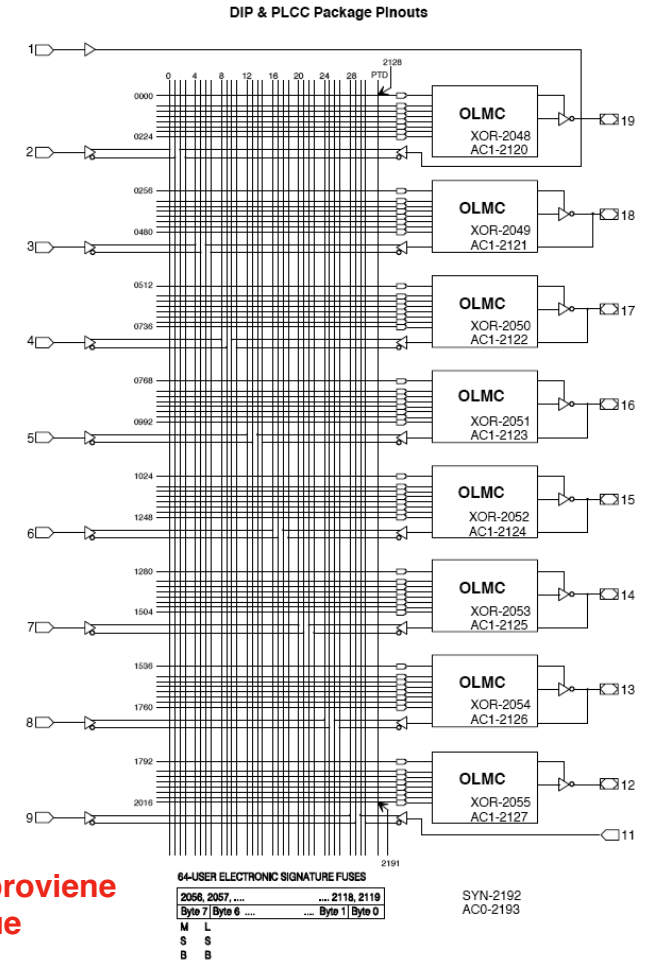


Combinatorial Output Configuration for Complex Mode

- SYN=1.
- AC0=1.
- XOR=0 defines Active Low Output.
- XOR=1 defines Active High Output.
- AC1=1.
- Pin 12 and Pin 19 are configured to this function.

patillas 12 y 19

Complex Mode Logic Diagram



Hasta 2 Salidas y hasta 6 I/Os

Con SYN = ‘1’ y AC0 = ‘1’ todos los FMUXs seleccionan la señal que proviene de la propia OLMC; excepto en en el caso de las patillas 12 y 19 en que seleccionan la señal que proviene de la “OLMC” (entrada) adyacente

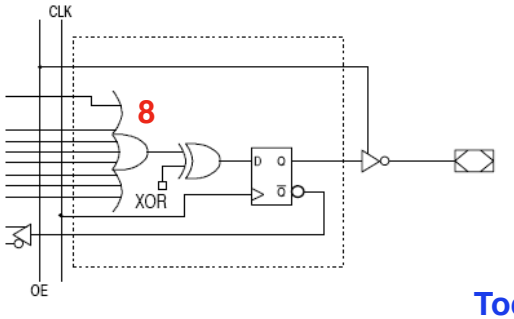
Diagrama de bloques de una GAL 16V8

Modo Registrado

GAL10V8

SYN, AC0 = '0, 1'

AC1 = '0'

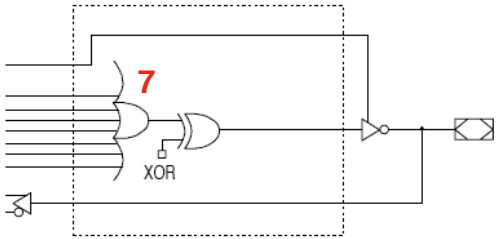


Registered Configuration for Registered Mode

- SYN=0.
- AC0=1.
- XOR=0 defines Active Low Output.
- XOR=1 defines Active High Output.
- AC1=0 defines this output configuration.
- Pin 1 controls common CLK for the registered outputs.
- Pin 11 controls common $\overline{OE}$ for the registered outputs.
- Pin 1 & Pin 11 are permanently configured as CLK & $\overline{OE}$ for registered output configuration.

Todas las OLMC (patillas 12 a 19)

AC1 = '1'



Combinatorial Configuration for Registered Mode

- SYN=0.
- AC0=1.
- XOR=0 defines Active Low Output.
- XOR=1 defines Active High Output.
- AC1=1 defines this output configuration.
- Pin 1 & Pin 11 are permanently configured as CLK & $\overline{OE}$ for registered output configuration.

Todas las OLMC (patillas 12 a 19)

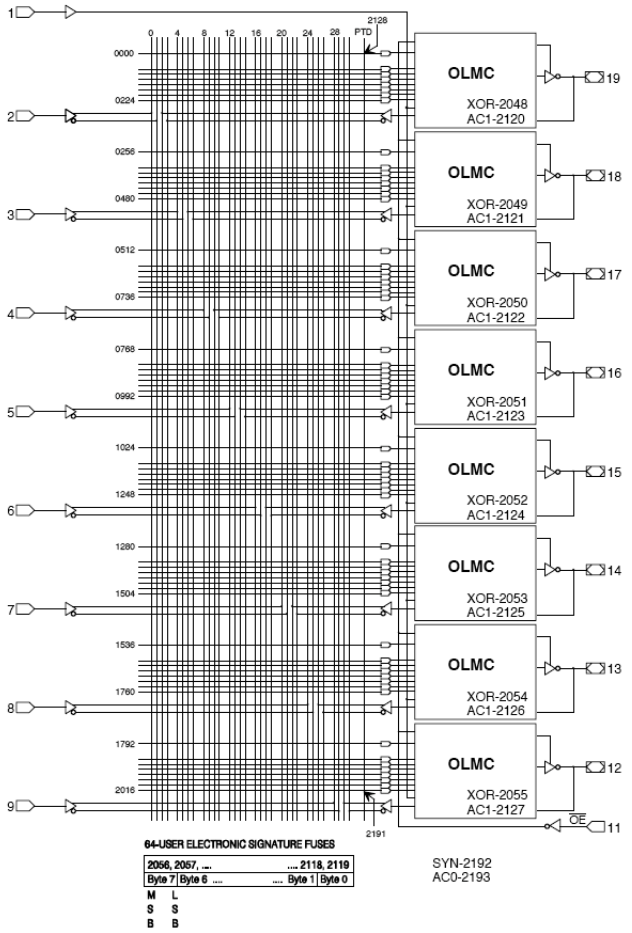
Las patillas 1 y 11 permanentemente configuradas para CLK y / OE, respectivamente, para la configuración de salida registrada

Hasta 8 Salidas vía registro. Hasta 8 I/Os

La entrada o realimentación es desde la propia patilla u OLMC

Registered Mode Logic Diagram

DIP & PLCC Package Pinouts



GAL 16V8: Emulación de PALs

Un dispositivo GAL16V8, utilizando los distintos modos, puede emular a los siguientes dispositivos PAL

PAL Architectures Emulated by GAL16V8	GAL16V8 Global OLMC Mode
16R8	Registered
16R6	Registered
16R4	Registered
16RP8	Registered
16RP6	Registered
16RP4	Registered
16L8	Complex
16H8	Complex
16P8	Complex
10L8	Simple
12L6	Simple
14L4	Simple
16L2	Simple
10H8	Simple
12H6	Simple
14H4	Simple
16H2	Simple
10P8	Simple
12P6	Simple
14P4	Simple
16P2	Simple

GAL 16V8: Síntesis expresiones lógicas

Ejercicio

Mostrar una implementación (*) sobre un dispositivo GAL16V8, de las siguientes funciones (sin desarrollar las expresiones):

$F_1.H = [\ /x_1 \cdot /x_2 + x_1 \cdot x_3 + /x_2 \cdot /x_3\ ].H$ **(SOP.H)**

$F_2.L = [\ /x_1 \cdot /x_2 + x_1 \cdot x_3 + /x_2 \cdot /x_3\ ].L$ **(SOP.L)**

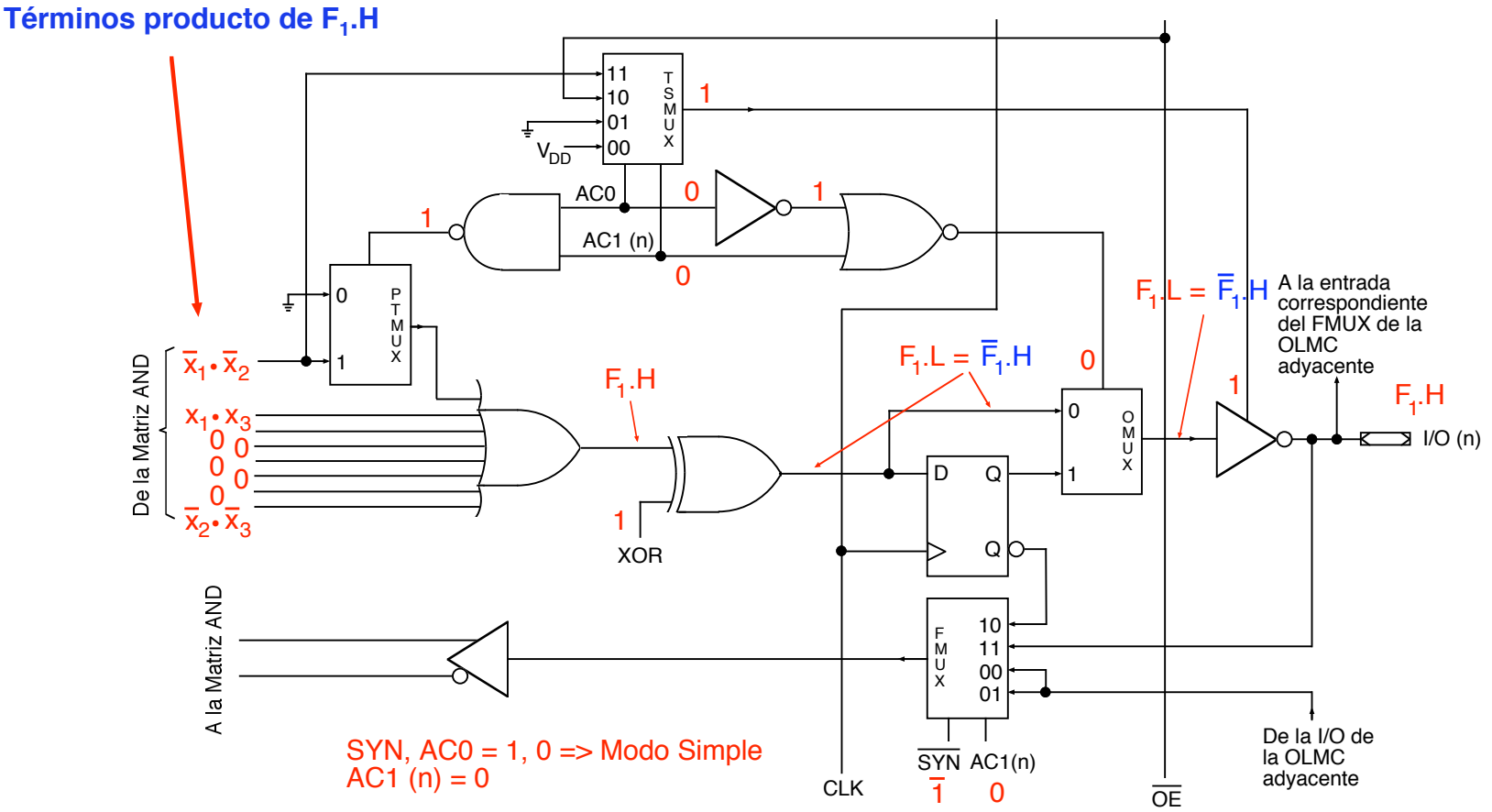
$F_3.L = [\ (/x_1 + x_2) \cdot (/x_1 + x_3) \cdot (x_2 + /x_3)\ ].L$ **(POS.L)**

$F_4.H = [\ (/x_1 + x_2) \cdot (/x_1 + x_3) \cdot (x_2 + /x_3)\ ].H$ **(POS.H)**

(*) Aún cuando la selección de modo y la programación de los fusibles SYN, AC0 y AC1 (n), la realiza el compilador software / hardware de forma "transparente" para el diseñador, mostrar como podrían programarse estos fusibles. Considérese la implementación en una macrocelda con entradas de selección del FMUX: S1 S0 = /SYN AC1 (OLMC 12 o 19).

GAL 16V8: Síntesis expresiones lógicas (2)

Ejercicio

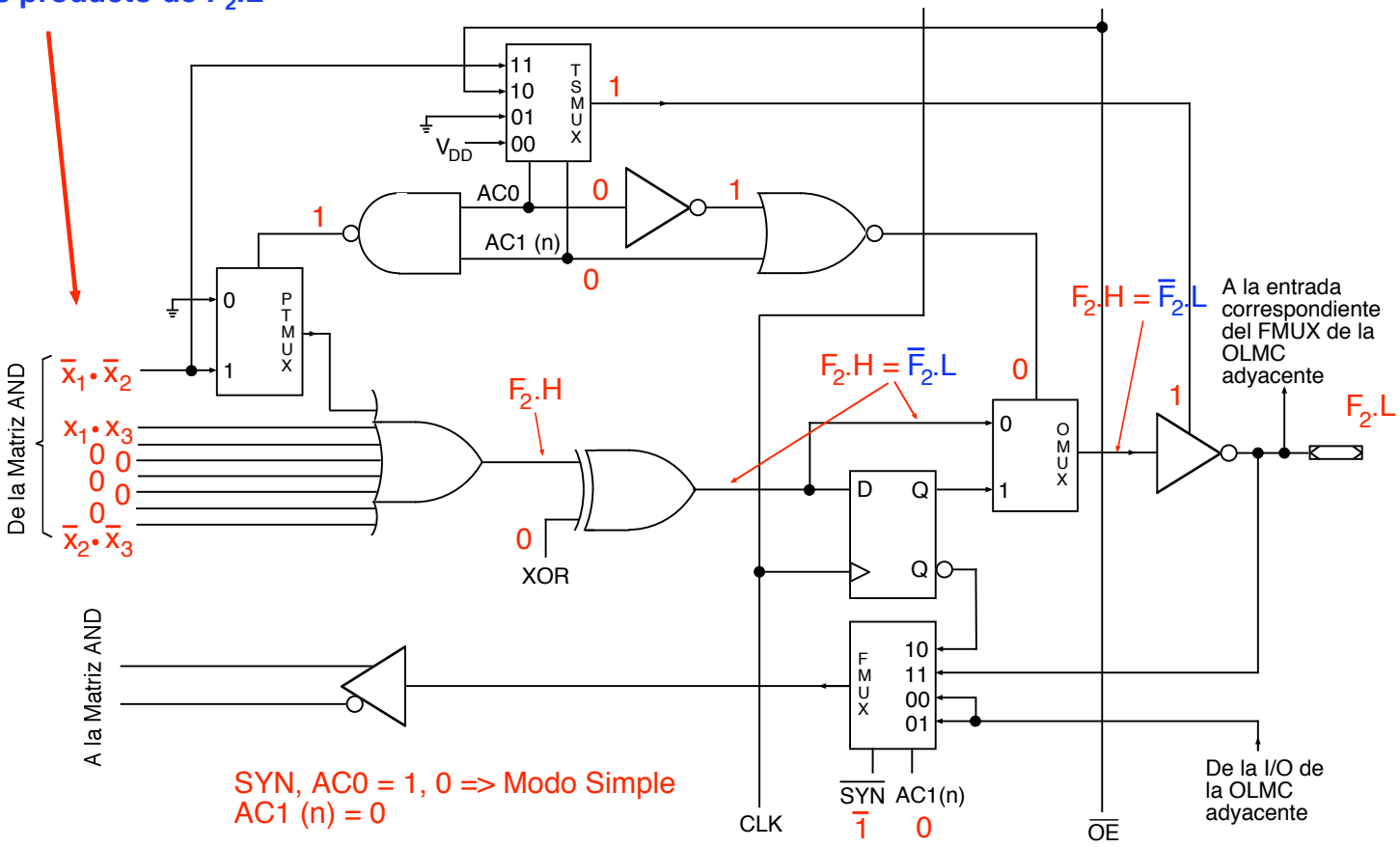


$F_1.H = [\ /x_1 \cdot /x_2 + x_1 \cdot x_3 + /x_2 \cdot /x_3\ ].H$ **(SOP.H)**

GAL 16V8: Síntesis expresiones lógicas (3)

Ejercicio

Términos producto de F_2.L



$F_2.L = [\ /x_1 \cdot /x_2 + x_1 \cdot x_3 + /x_2 \cdot /x_3\ ].L$ **(SOP.L)**

GAL 16V8: Síntesis expresiones lógicas (4)

Ejercicio

$F_3.L = [\ (/x_1 + x_2) \cdot (/x_1 + x_3) \cdot (x_2 + /x_3)\ ].L$ (POS.L)

$F_3.L = /F_3.H = /[\ (/x_1 + x_2) \cdot (/x_1 + x_3) \cdot (x_2 + /x_3)]\ .H =$

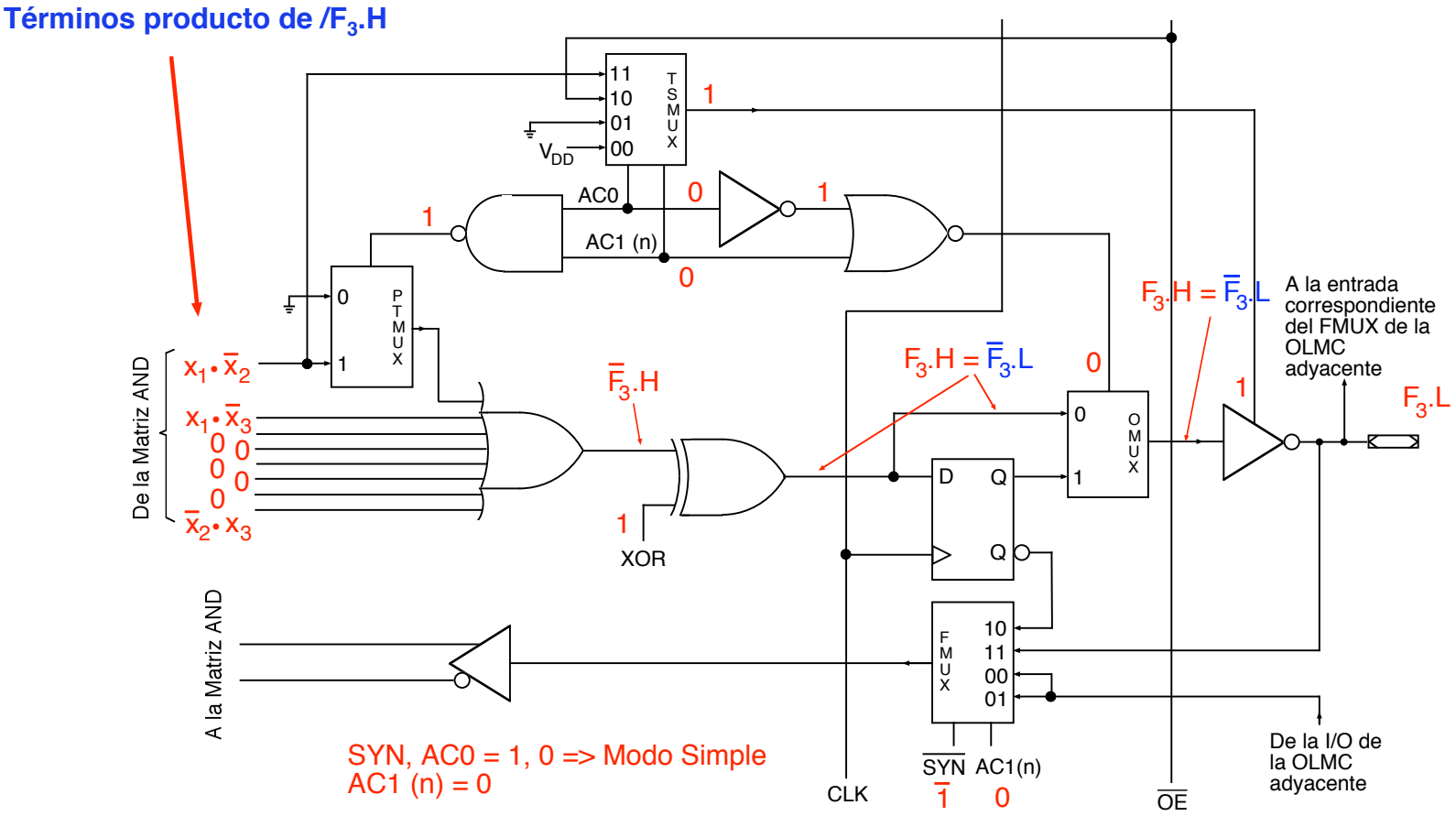
$= [\ /(/x_1 + x_2) + /(/x_1 + x_3) + /(x_2 + /x_3)\ ].H =$

$= [\ x_1 \cdot /x_2 + x_1 \cdot /x_3 + /x_2 \cdot x_3\ ].H$ (SOP.H)

Términos producto de $/F_3.H$

GAL 16V8: Síntesis expresiones lógicas (5)

Ejercicio



$F_3.L = [\ (/x_1 + x_2) \cdot (/x_1 + x_3) \cdot (x_2 \cdot\ /x_3)\ ].L$ **(POS.L)**

GAL 16V8: Síntesis expresiones lógicas (6)

Ejercicio

$F_4.H = [\ (/x_1 + x_2) \cdot (/x_1 + x_3) \cdot (x_2 + /x_3)\ ].H$ (POS.H)

$F_4.H = /F_4.L = /[\ (/x_1 + x_2) \cdot (/x_1 + x_3) \cdot (x_2 + /x_3)\ ].L =$

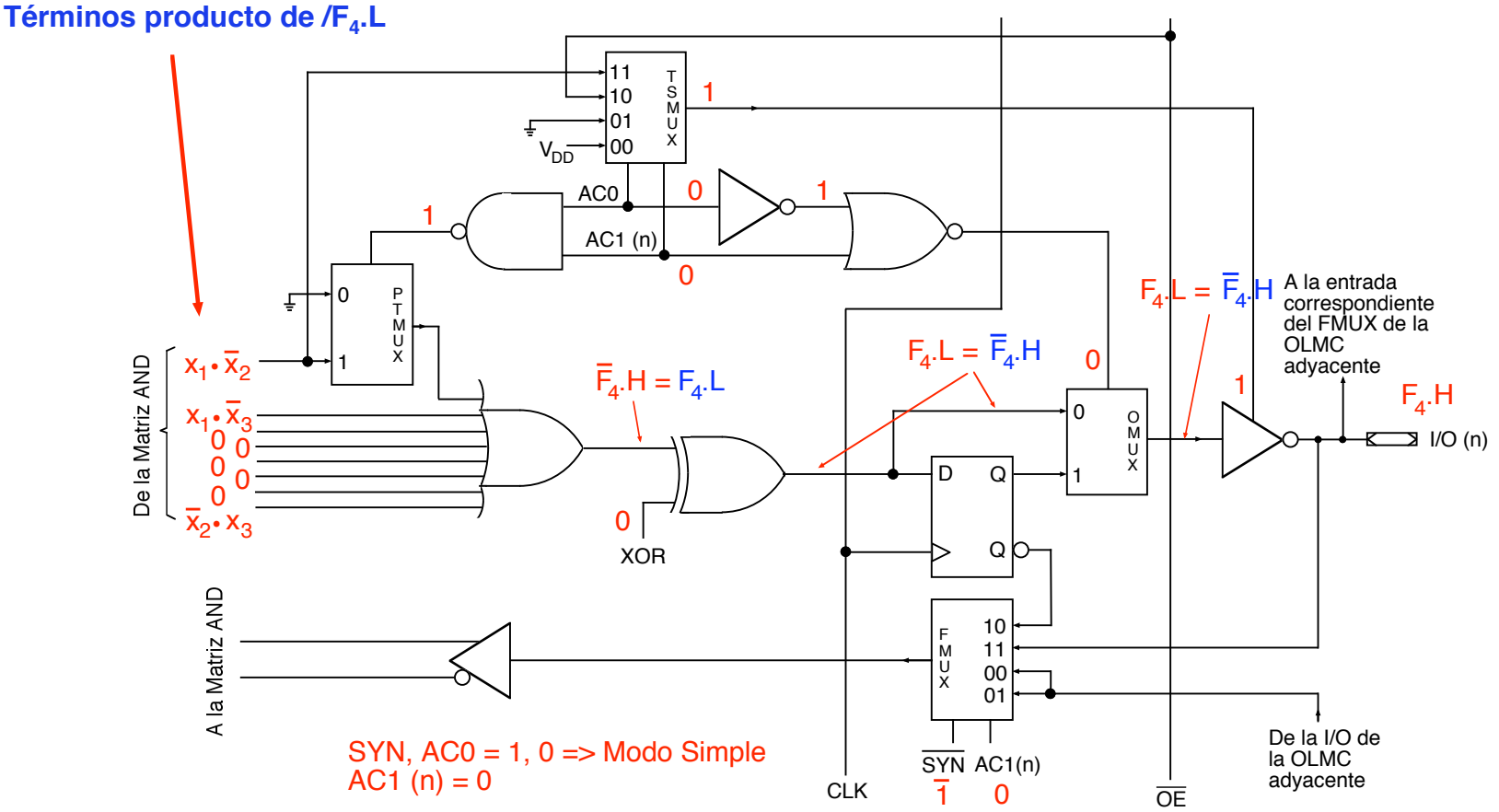
$= [\ /(/x_1 + x_2) + /(/x_1 + x_3) + /(x_2 + /x_3)\ ].L =$

$= [\ x_1 \cdot /x_2 + x_1 \cdot /x_3 + /x_2 \cdot x_3\ ].L$ (SOP.L)

Términos producto de $/F_4.L$

GAL 16V8: Síntesis expresiones lógicas (7)

Ejercicio



$F_4.H = [\ (/x_1 + x_2) \cdot (/x_1 + x_3) \cdot (x_2 + /x_3)\ ].H$ **(POS.H)**

CPLDs

"Complex PLD"

- Versiones más grandes y complejas de los PLDs simples

- Matriz de interconexión interna centralizada que se utiliza para conectar de forma óptima las celdas del dispositivo
  - máxima velocidad, funcionalidad, versatilidad, …

- Macroceldas configurables
  - Polaridad de salida
  - Combinacional o registro (D-FF, JK-FF, SR-FF, T-FF)
  - Triestado
  - Realimentación de la señal a la matriz de interconexión
  - Expansores

- Arquitectura

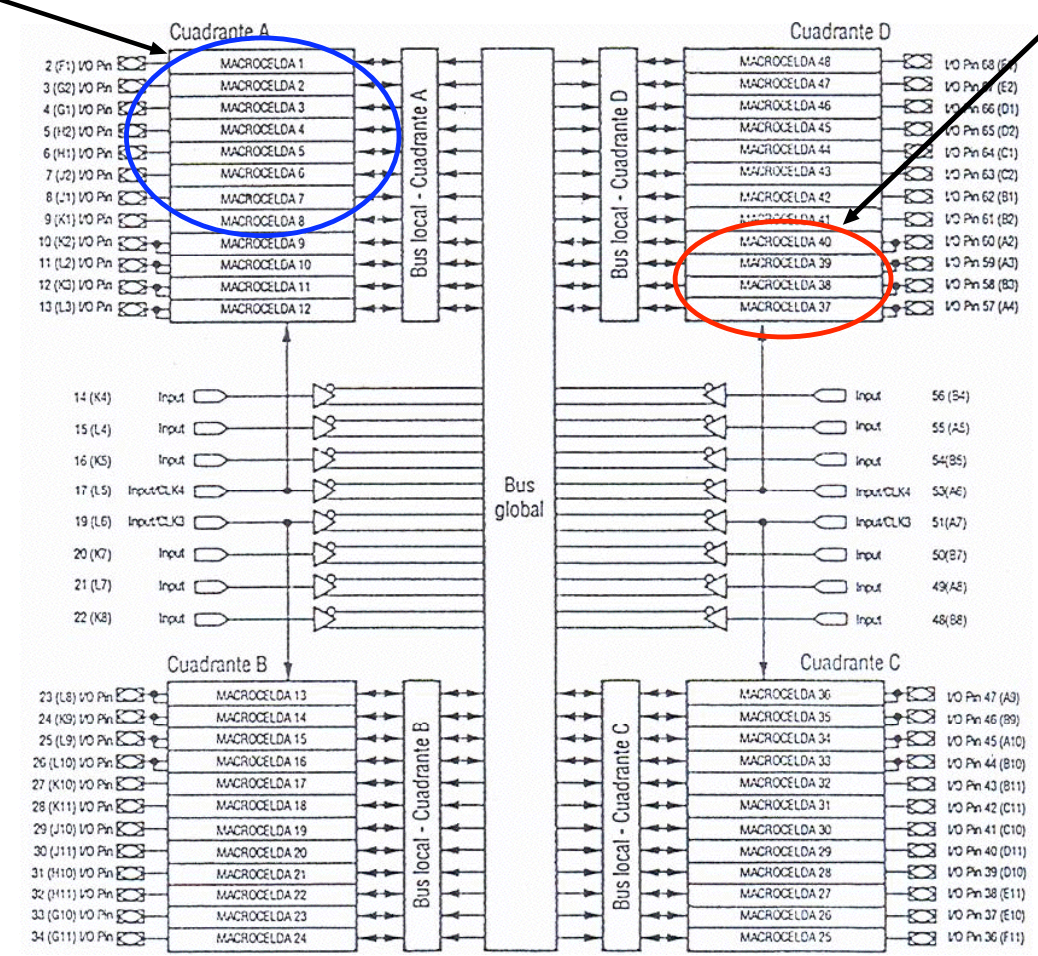
  PIN de Entrada --> buffer/inversor --> Matriz de interconexión -->
  --> Lógica AND-OR --> Macrocelda de Salida

Limita la flexibilidad de diseño, pero simplifica el análisis del mismo al ofrecer una temporización de señales desde entradas a salidas muy predecible.

EP1830

Macroceldas locales

Macroceldas globales



EP1830

ALTERA

48 Macroceldas

1 BUS global

12 Entradas dedicadas globales

4 Cuadrantes de 12 Macroceldas

4 BUSES locales

Cuadrante:

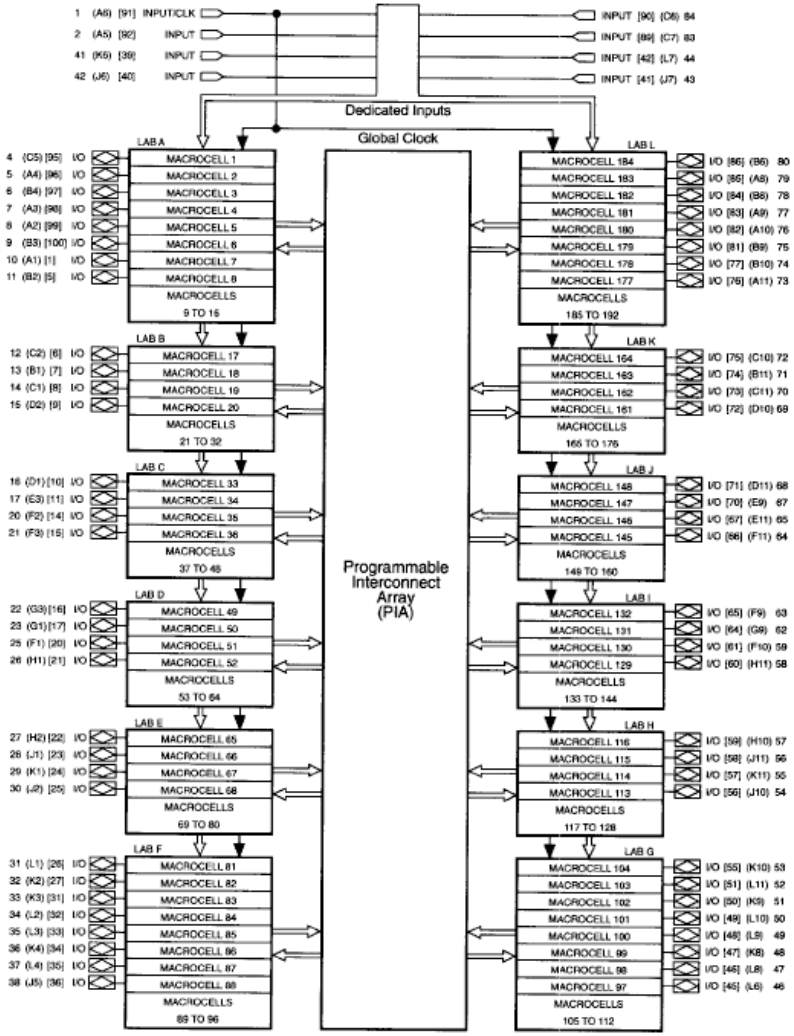
- **Reloj independiente**
- **8 Macroceldas locales**
- **4 Macroceldas globales**

UV EPROM-OTPROM

EPM5192

EPM5192

ALTERA MAX 5000



192 Macroceldas

12 LABs ("Logic Array Block") de 16 Macroceldas y 32 términos producto para expansión

388 términos producto para ampliación (Expansor)

64 I/Os: Entrada, Salida o bidireccional
Hasta 72 Entradas o 64 Salidas

15 Entradas dedicadas globales. 1 puede utilizarse como Reloj global

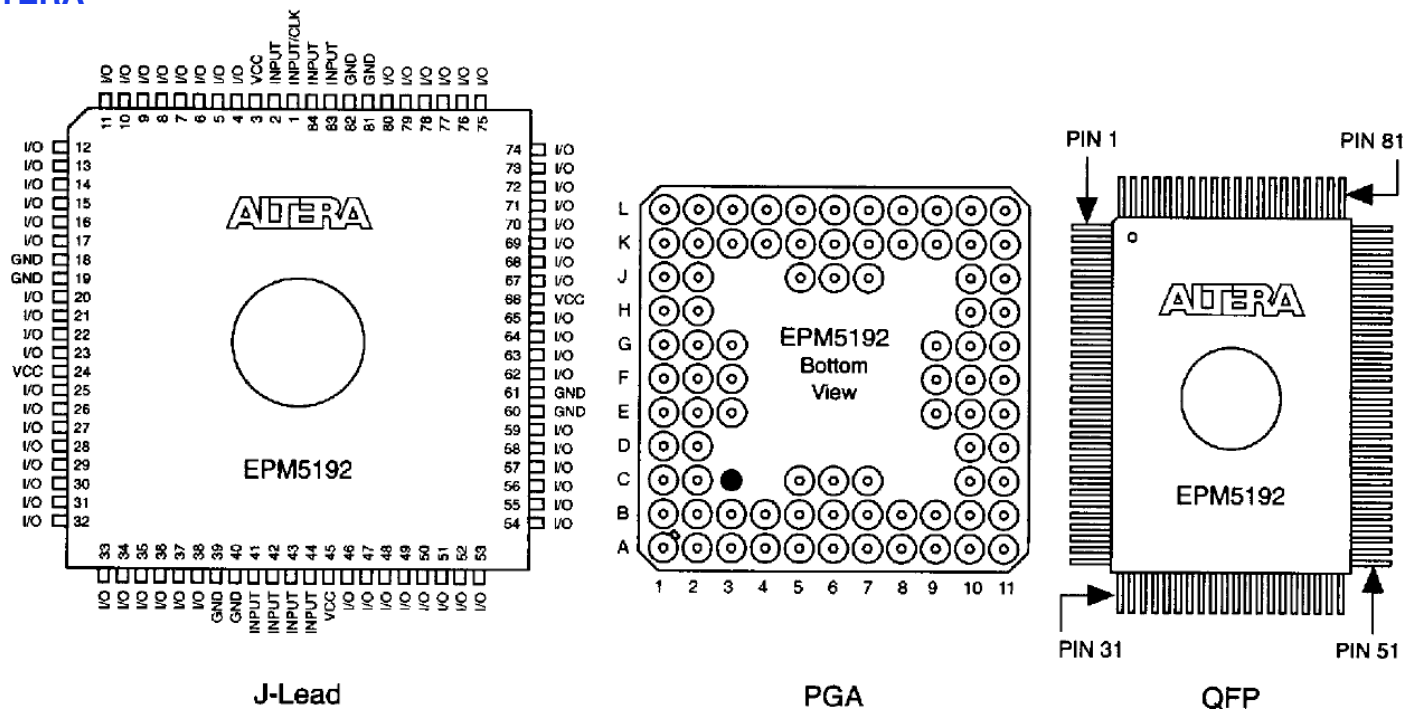
Pueden coexistir Reloj global y relojes locales

12 PIA
"Programmable Interconect Array"

EPM5192 (2)

EPM5192

ALTERA



J-Lead PGA QFP

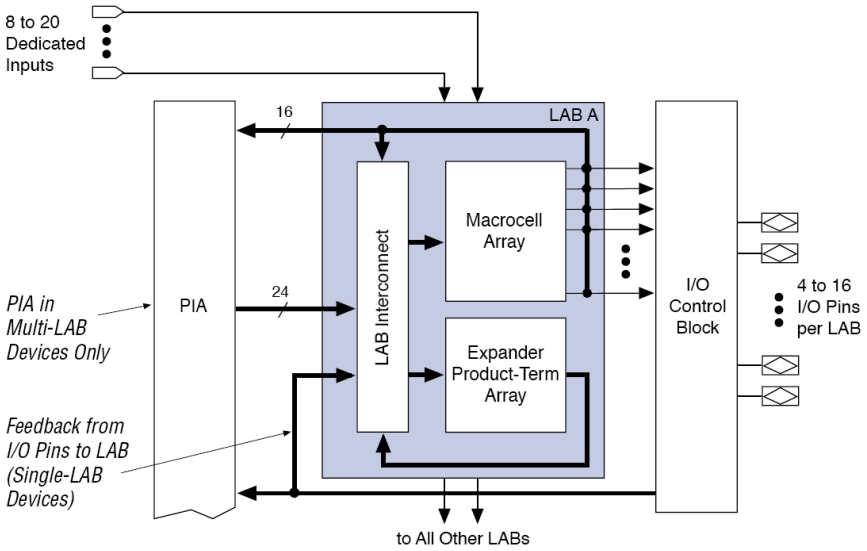
UV EPROM - OTPROM

EPM5192 (3)

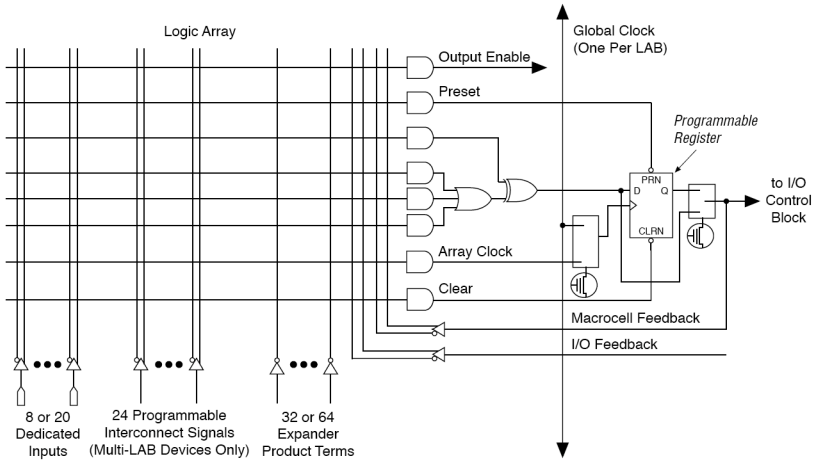
EP5192

Arquitectura

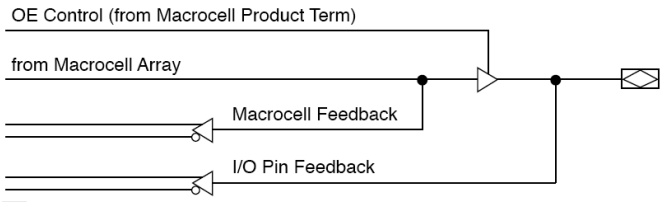
LAB: "Logic Array Block"



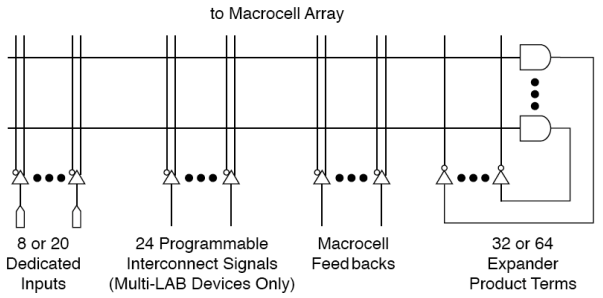
Macrocelda



Control I/O



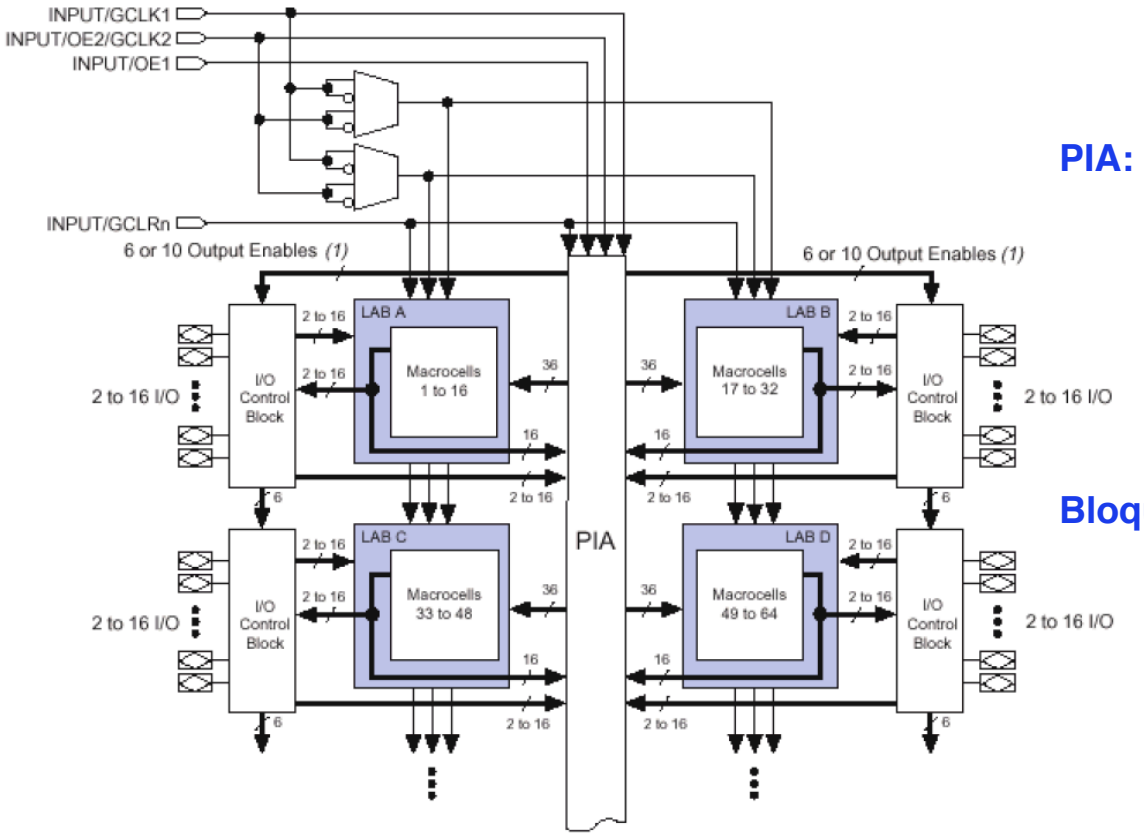
Expansor



CPLDs Eléctricamente borrables

Familia MAX 7000

Arquitectura



LABs: "Logic Array Blocks"

16 Macroceldas

PIA: "Progammable Interconect Array"

Estructura de conexionado jerárquica

Retardo predecible
~ 5 ns 'pin' a 'pin'

Bloques de control I/O

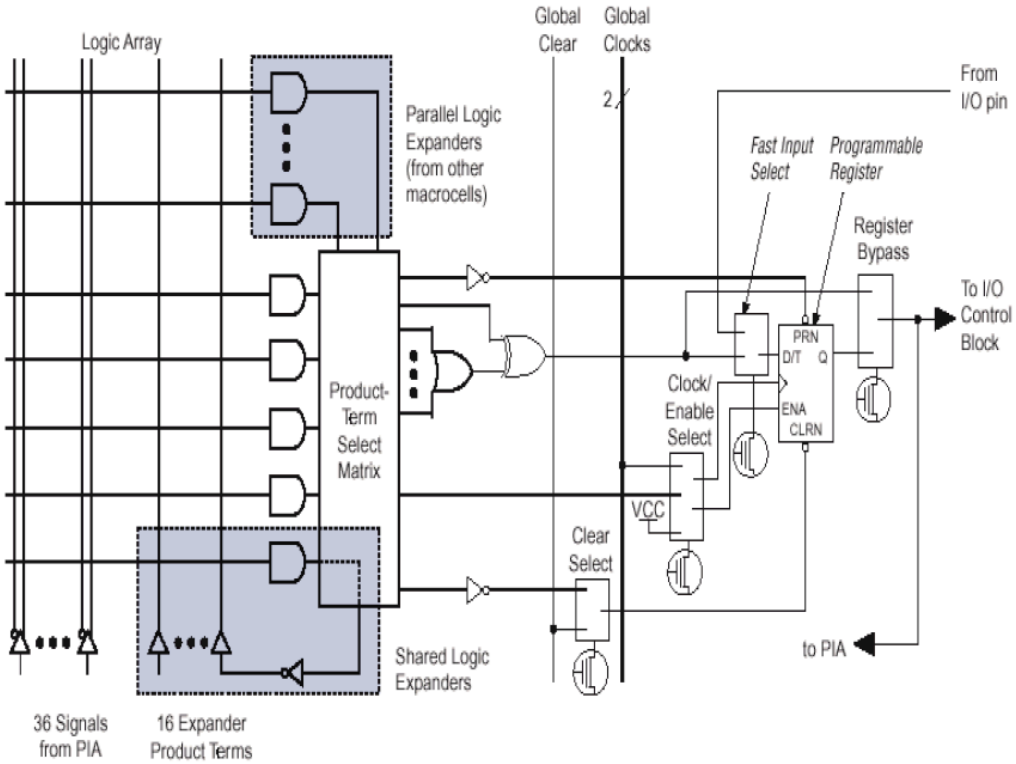
Ej.: EPM7128
8 LABs (128 macroceldas)
100 (hasta) I/O pins utilizables
6 (hasta) OE

EEPROM **ALTERA**

CPLDs Eléctricamente borrables (2)

Familia MAX 7000

Macrocelda



Macrocelda

- Matriz Lógica
- Matriz de selección de términos producto
  - Para entradas de la OR o XOR
  - Para Clear, Preset, CLK o habilitación de CLK del FF de la macrocelda
- Productos para expansión
  - De macroceldas adyacentes
  - Compartidos por cualquier macrocelda
- Registro programable (individual)
  - D, T, J-K, S-R
  - Control de reloj
- Control de 'slew-rate' de salida

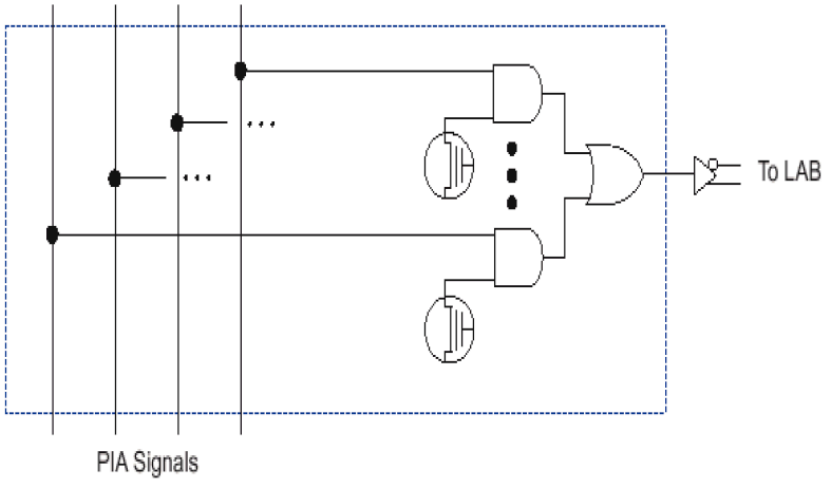
EEPROM

ALTERA

CPLDs Eléctricamente borrables (3)

Familia MAX 7000

Macrocelda



PIA

Admite como entradas:

- **Entradas dedicadas**
- **I/Os**
- **Salidas de macroceldas**

Salidas de PIA:

- **Disponibles por todos los LABs**
- **Sólo se conectan a cada LAB las señales requeridas por este.**
- **Controlado por una celda EEPROM**

Programación de PLDs

Requerimientos

◊ **Software de programación (compilador lógico, compilador del diseño)**
Interpreta entrada del diseño
Diagrama esquemático
HDL ("Hardware Description Language")
Expresiones Booleanas
Tablas Características, Tablas de Estados, …
Genera el fichero de salida para programación
Formato JEDEC ("Joint Electronic Device Engineering Council")
Paquetes: ABEL, CUPL, OrCAD-PLD, LOG/iC, Max+PLUSII, …

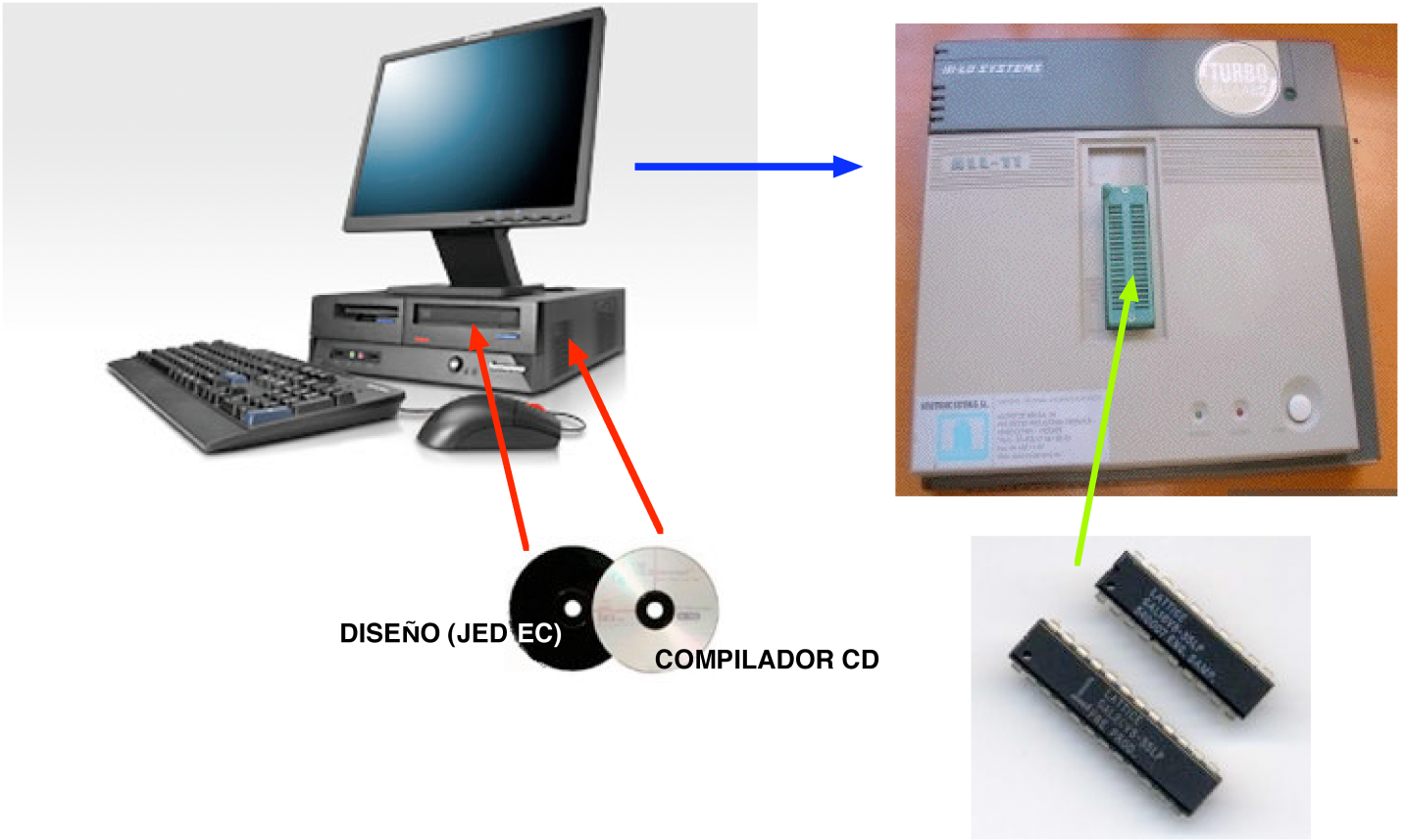
◊ **Computador**
Ejecuta el programa software de programación
Controla el hardware de programación

◊ **Programador controlado por software**
Bajo control del programa, a partir del fichero JEDEC, genera las señales de programación, adecuadas a cada dispositivo (voltajes, intensidades, temporización, …)
SPRINT, ALL11, …

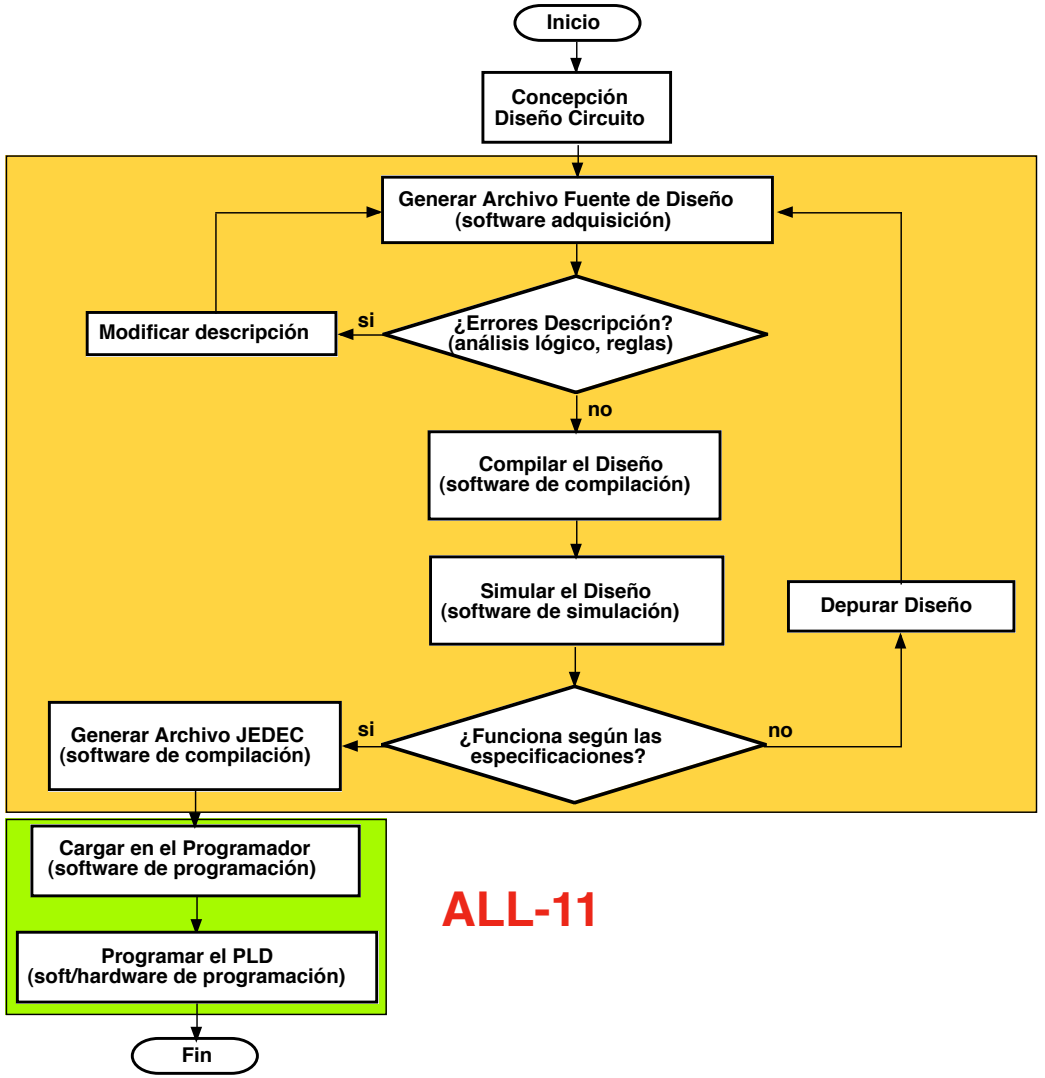
* **Algunos paquetes generan las señales de programación y las aplican al circuito "in situ", a través del puerto de salida paralelo o serie, mediante un simple dispositivo conector. Requiere que los PLDs dispongan de la facilidad de acceso "scan" (CPLDs)**
ispGAL ("In-System Programmable GAL"), MAX+PLUSII (Puerto paralelo-ByteBlaster MV)

Programación de PLDs

Programanción



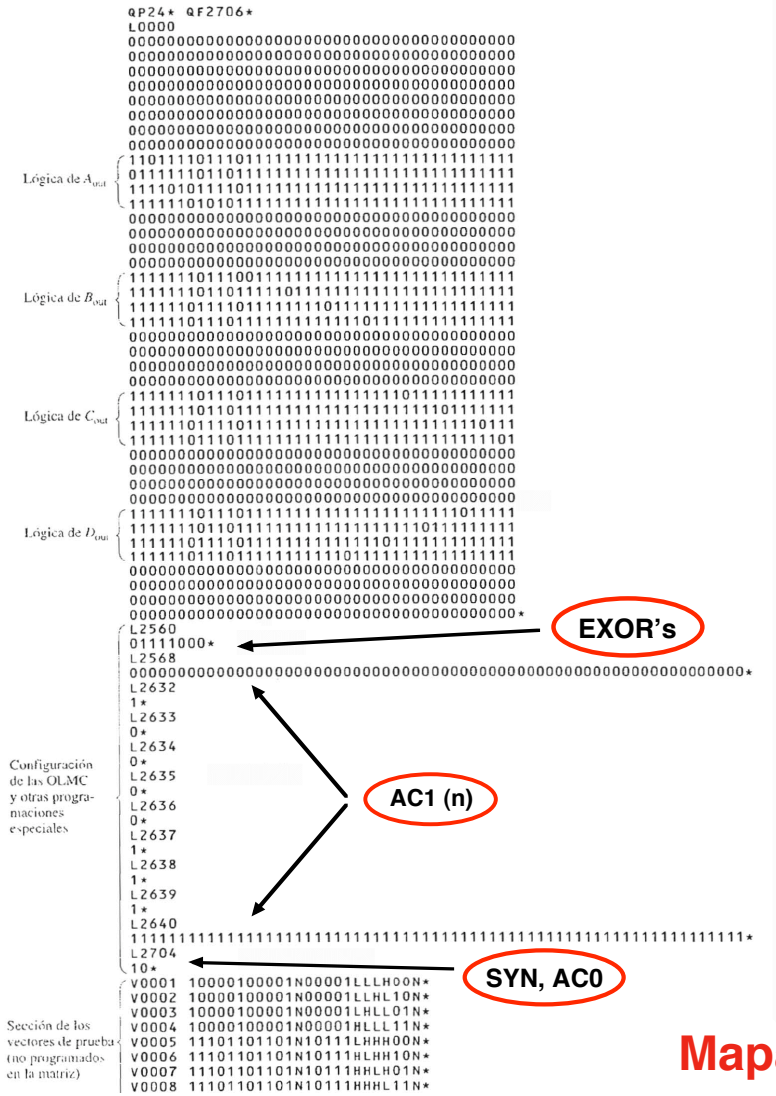
Programación de PLDs

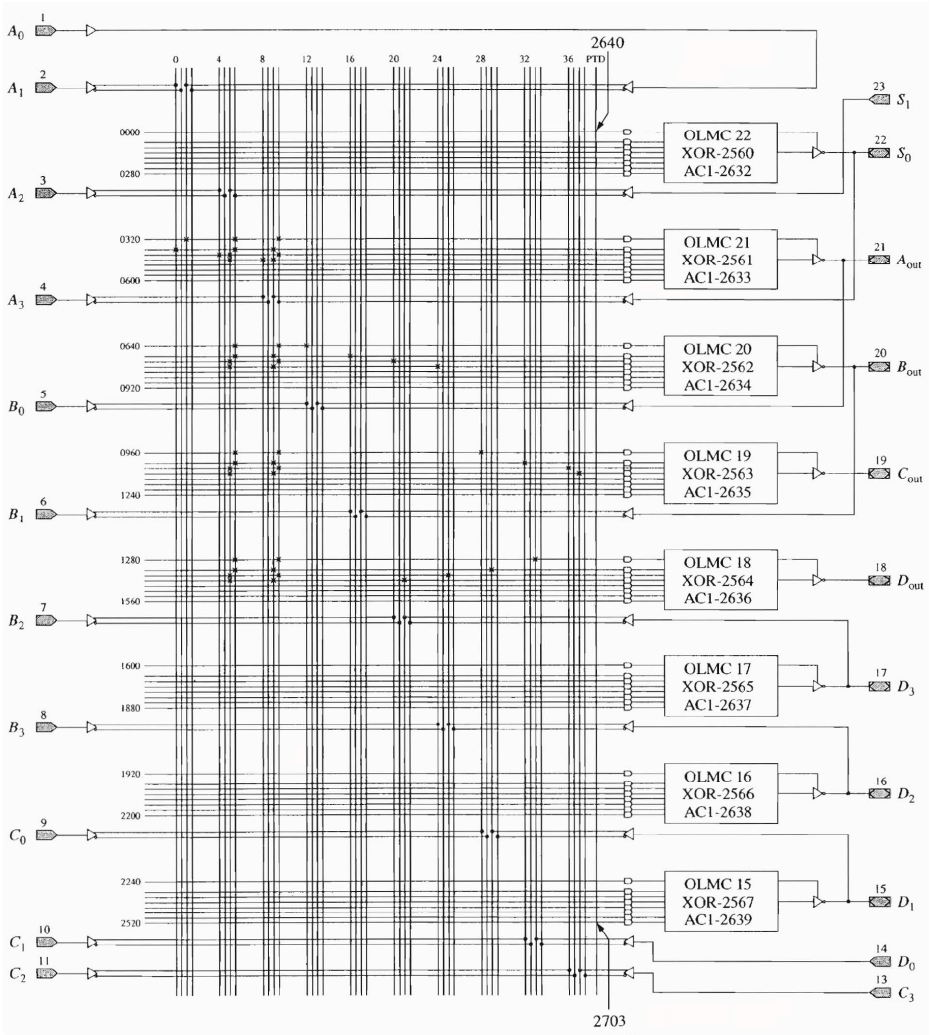


MAX+PLUSII

ALL-11

Formato JEDEC





Mapa de Fusibles

Programación de PLDs