

## Tema 5: Lógica Combinacional Programable

# Lógica Programable

Alternativamente a utilizar dispositivos lógicos de función específica existen muchos dispositivos que permiten su programación.

Con estos, pretendemos extender las posibilidades de “cierta programabilidad” que hasta el momento podíamos conseguir con los multiplexores, y otros MSI similares

Las ventajas principales de estos dispositivos son las siguientes:

- Son genéricos y por tanto fabricables con una mejor relación de costes-prestaciones
- Se puede conseguir una funcionalidad bastante compleja con un factor de forma pequeño: en un único chip se pueden conseguir una densidad lógica importante
- El diseño puede alterarse ¡sólo con software!.
  - Es un diseño más sencillo

Otras ventajas:

- Si se está haciendo un diseño en tarjetas se consigue además de un factor de forma mejor un menor número de conexiones entre las partes
  - Además el conexionado es fiable
  - Diseño soft, implementado con puertas.
  - Densidad de componentes cada vez más alta

# Lógica Programable

Existen multitud de variantes:

**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** (“**E**rasable **PLD**”)

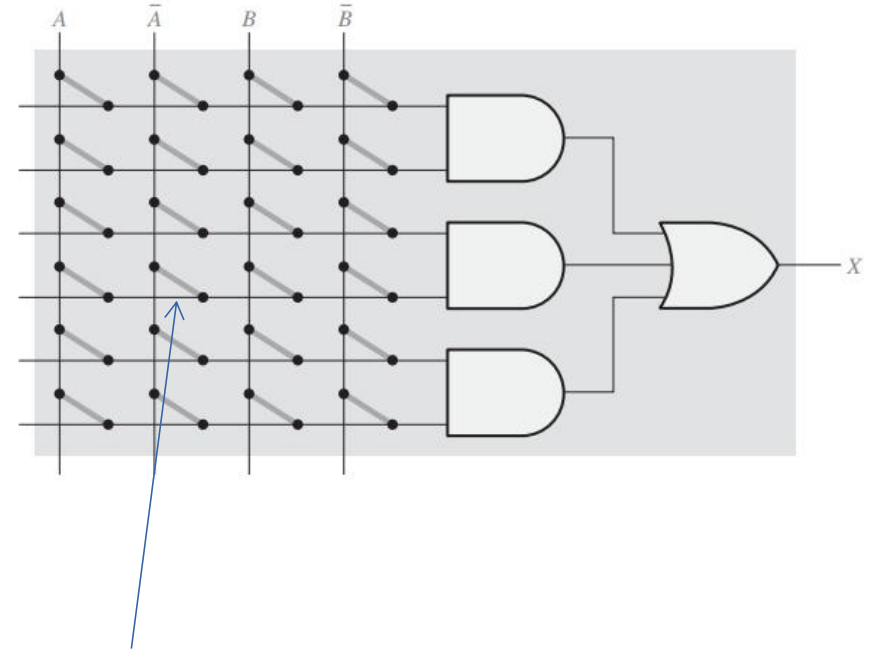
**CPLD** (“**C**omplex **PLD**”)

# Lógica Programable - SPLD

.Varias aproximaciones:

-SPLD (dispositivos más sencillos, *Simple Programmable Logic Devices*) : Ej: **PAL**

- PAL (*Programmable Array Logic*). El término algo ambiguamente se usa de forma genérica para referirse a PALs, GALs, ROMs, PLAs (no son estrictamente lo mismo)
- Es una “matriz” de conexiones AND que se puede programar para formar un circuito. Todos los AND se unen después por un OR. Son como una expresión en *hardware* de los *min-terms*: *suma de productos*
- En el caso más “puro” sólo se puede programar una vez mediante unos “fusibles” que conectan o desconectan cada entradas de las puertas (una señal a cada entrada de puerta). Las **GAL** sí son reprogramables múltiples veces.
- Estos dispositivos tienen en general un alto rendimiento en velocidad de operación. Por eso pueden aparecer en partes de un diseño más complejo, p.ej. para descargar de ciertas operaciones al microprocesador



Se eliminan algunos de estos según convenga  
En este caso TODAS las AND están conectadas  
a la salida por OR (la red OR es fija)

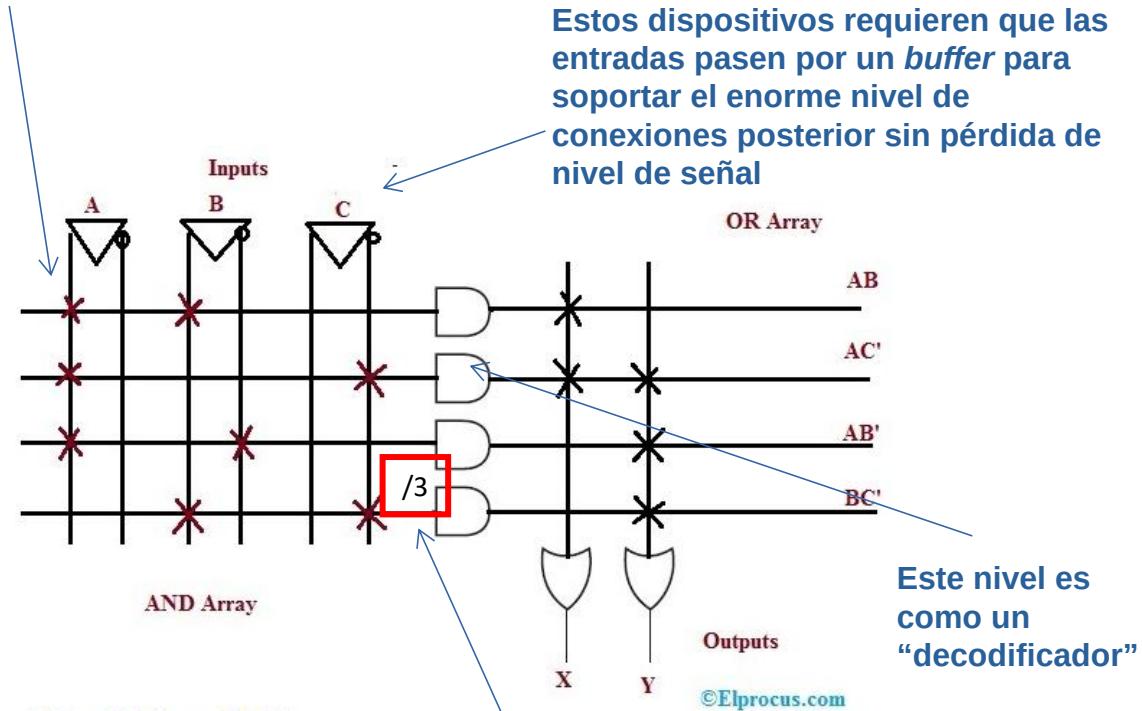
# PALs - PLA

•Conceptos que debemos tener:

-Notación compacta para los esquemáticos de PAL/PLA. Se evita indicar las múltiples entradas a una puerta AND

-La PLA (*programmable Logic Array*) permitirá normalmente además de especificar las conexiones en el nivel AND, también expresarlas en el nivel OR. Es un dispositivo bastante más complejo. Actualmente este dispositivo está cayendo en desuso a favor de las FPGA que veremos más adelante.

X = fusible intacto = conexión



$$X = AB + AC'$$

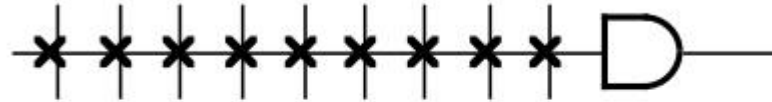
$$Y = AB' + BC'$$

Habitualmente se representa un sólo hilo pero se marca por ejemplo en este caso : "/ 3", lo que indica que son en realidad 3 conexiones

# PALs - PLA

Notaciones especiales:

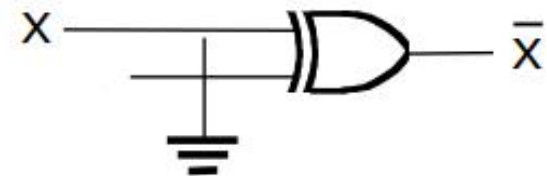
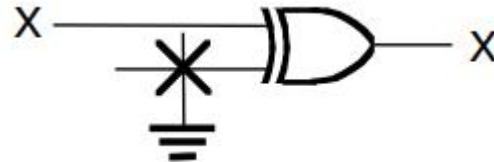
- Salida siempre a "L"



- Salida siempre a "H"

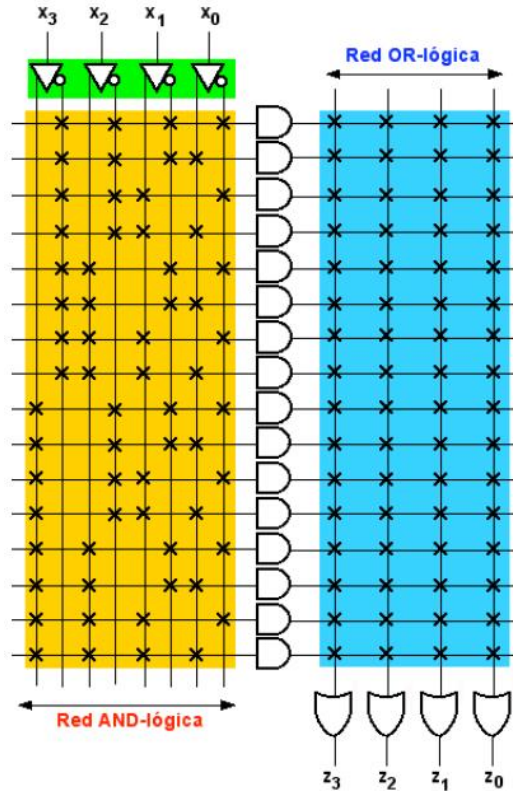


- Entrada especial



# PROM simbólica

Los  $2^n$  min-terms posibles ya están disponibles, se usen o no. Además, normalmente, la parte “AND” es fija



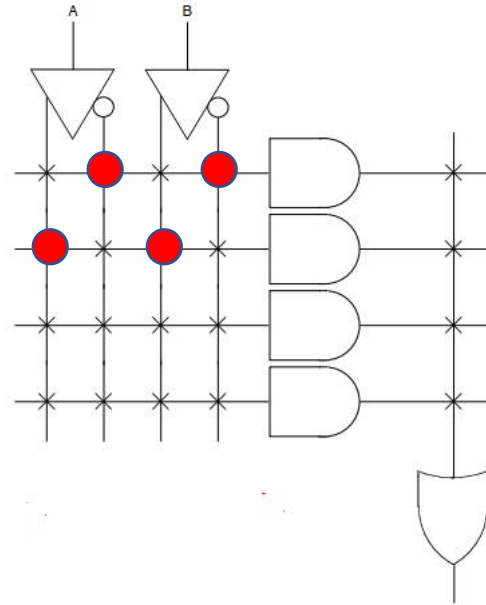
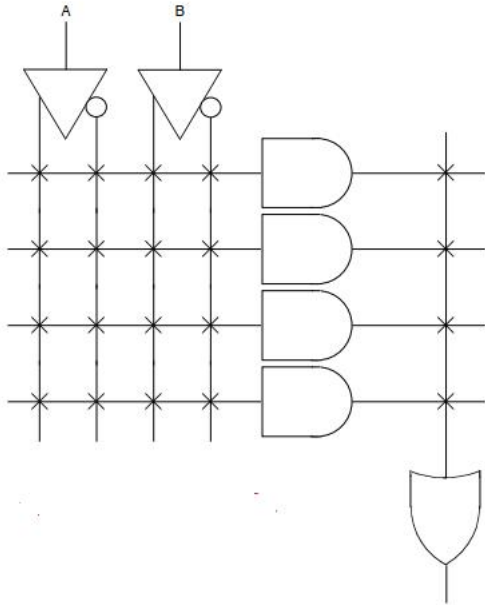
Todos los “sumandos” OR están disponibles, pero mediante el sistema de fusibles pueden estar o no conectados. Sólo este nivel es el programable.

En las PAL básicas suele ser al revés. La parte OR es la fija y la AND la programable



# Ejemplos de programación

- $F = A \text{ exnor } B$        $\overline{A \oplus B} = A \cdot B + \overline{A} \cdot \overline{B}$
- *Supongamos que tenemos como elemento programable una PAL básica con programabilidad sólo en el nivel de puertas AND.*



# Ejemplos de programación

- *Implementemos las siguientes funciones con distintos dispositivos:*

$$F_1 = \overline{A} \cdot B$$

$$F_2 = \overline{A} + B$$

$$F_3 = A \oplus B$$

$$F_4 = B$$

# Ejemplos de programación

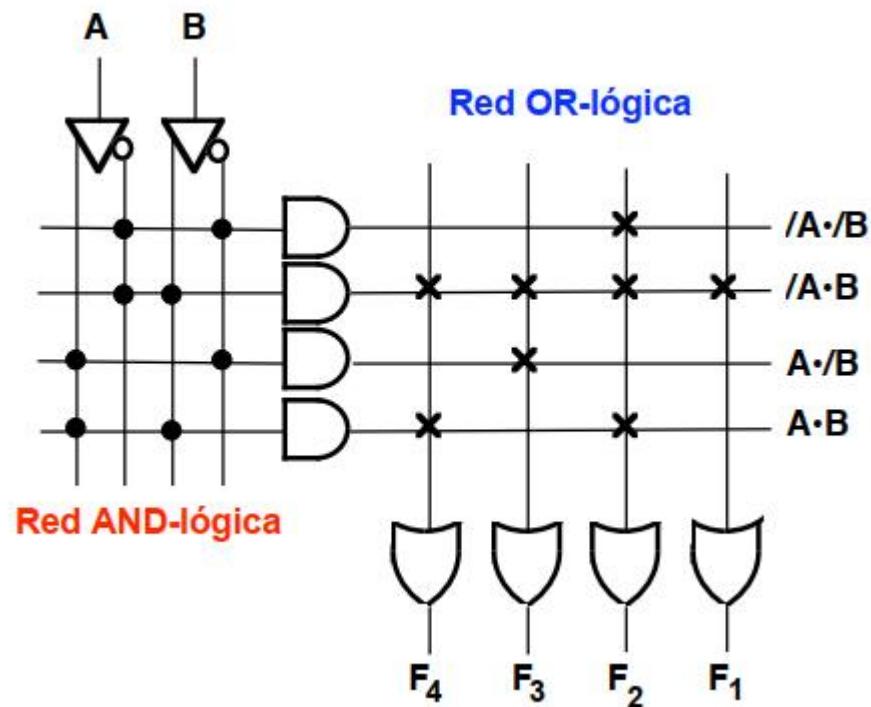
PROM

$$F_1 = \bar{A} \cdot B$$

$$F_2 = \bar{A} + B$$

$$F_3 = A \oplus B$$

$$F_4 = B$$



# Ejemplos de programación

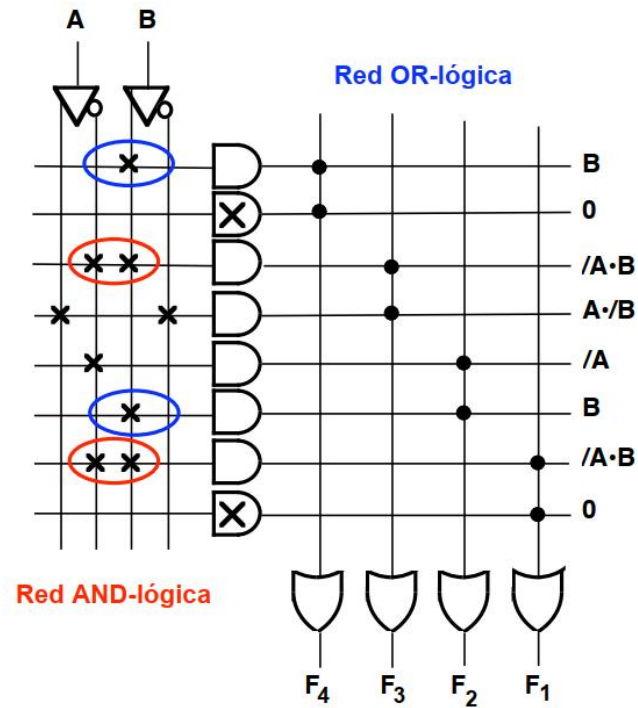
*PAL*

$$F_1 = \overline{A} \cdot B$$

$$F_2 = \overline{A} + B$$

$$F_3 = A \oplus B$$

$$F_4 = B$$



# Ejemplos de programación

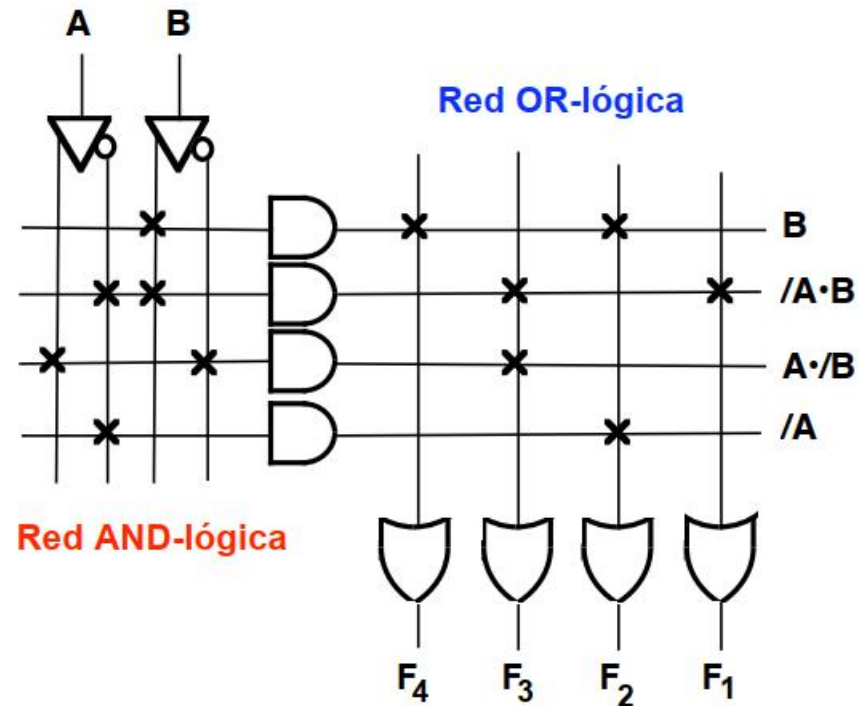
PLA

$$F_1 = \bar{A} \cdot B$$

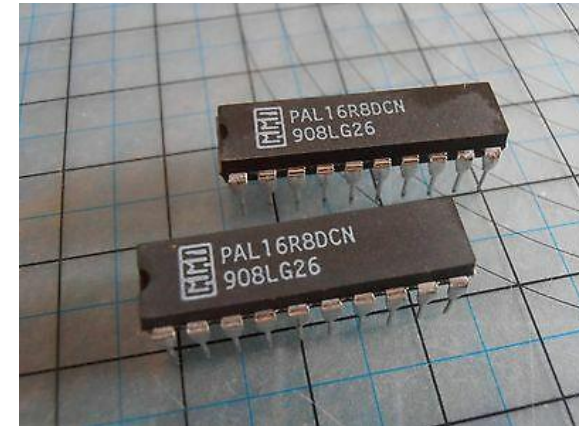
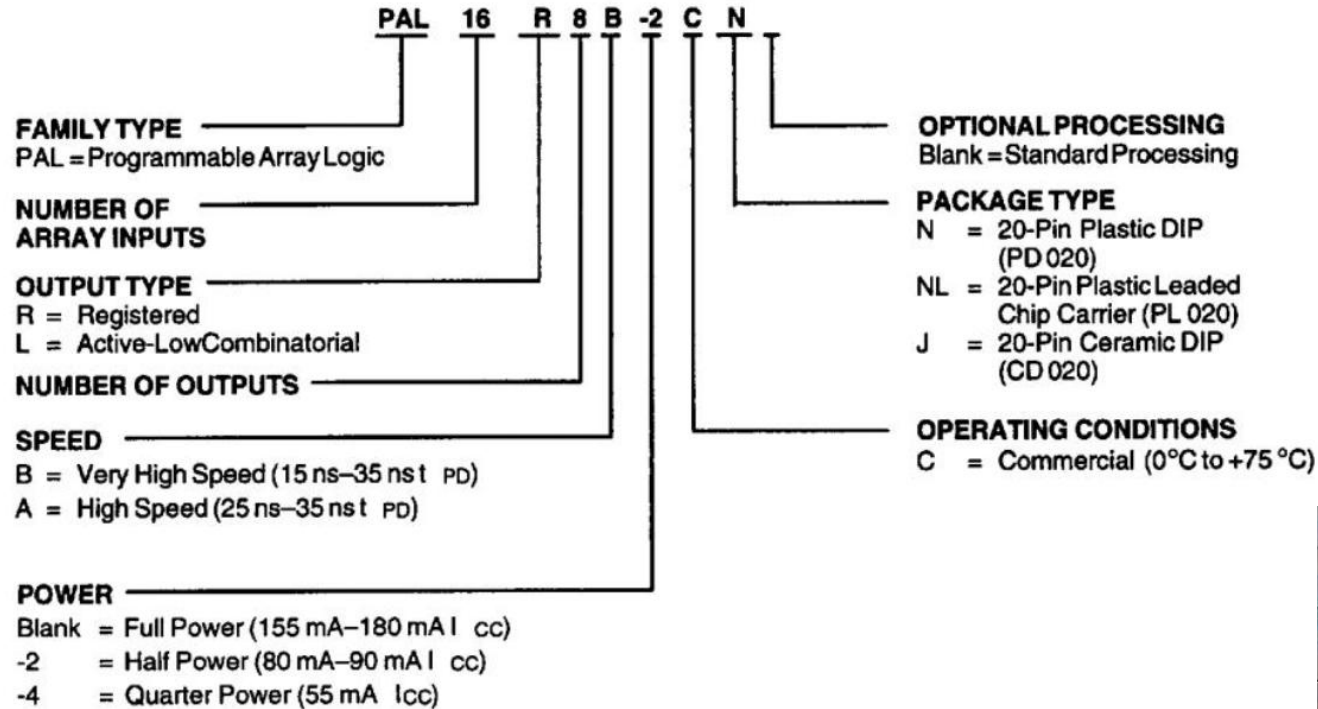
$$F_2 = \bar{A} + B$$

$$F_3 = A \oplus B$$

$$F_4 = B$$



# Selección de un dispositivo. Ejemplo



# Concepto de “macrocelda” en las GAL/PAL

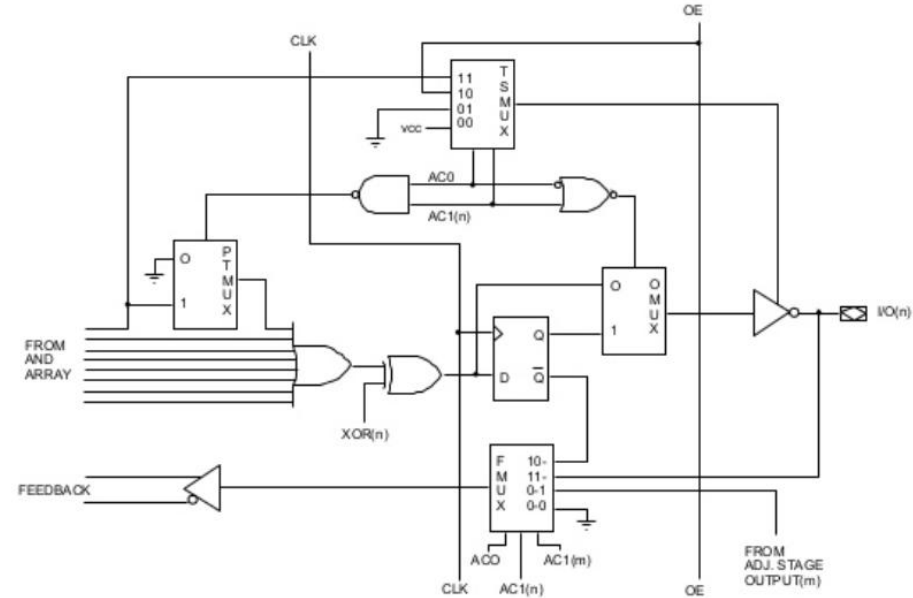
- A parte de las puertas “OR” en la salida estos dispositivos pueden contener algo más de circuitería para otras operaciones lógicas.
- Esta parte es la que constituye las **macrocells**.
- Las macroceldas extienden mucho la funcionalidad genérica del dispositivo.
- Las operaciones exactas que permiten difieren según el dispositivo o el fabricante. Típicamente:
  - Inversiones de lógica
  - Posibilidad de cambiar un pin de ser entrada a salida o viceversa
  - Reconectar de nuevo hacia adentro a otras partes del circuito la salida
  - etc.
- Es un parámetro más que se debe configurar en el dispositivo, a parte de las interconexiones.
- *Nota: Esta parte es muy dependiente del dispositivo concreto. Aunque existen unos conceptos generales que casi todos poseen, deben consultarse las hojas de características del dispositivo concreto a utilizar para conocer las funciones disponibles.*

# Ejemplo: Macroceldas en el dispositivo GAL16V8

•GAL16V8: 16 input, 8 output, pero con salidas configurables.

•Las funciones de salida pueden ser bastante sofisticadas. Algunas funciones posibles en esta macrocelda:

- Función normal combinacional
- Combinacional con realimentación hacia el interior (otras partes del circuito)
- Cambiar un puerto de salida a entrada
- Trabajar en modo “*registered*” : mantener valores en un registro de memoria
- Funcionar con lógica modo ALTO o BAJO (inversión de lógica)
- etc..



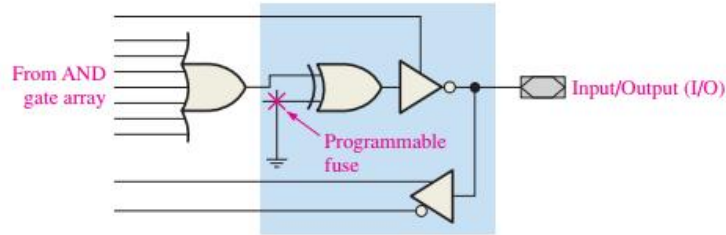
Buscar en internet y revisar un datasheet para este dispositivo:  
Por ej: <http://eeshop.unl.edu/pdf/gal16v8.pdf>



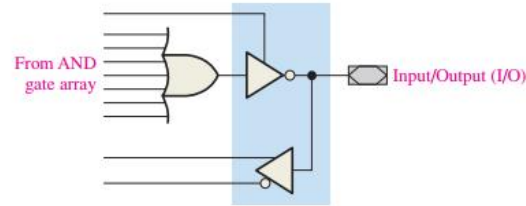
# Concepto de “macrocelda” en las GAL/PAL

Algunas funciones que aparecen en el esquema anterior:

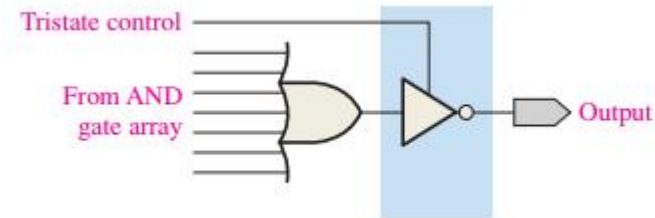
- ❑ Cambio de polaridad



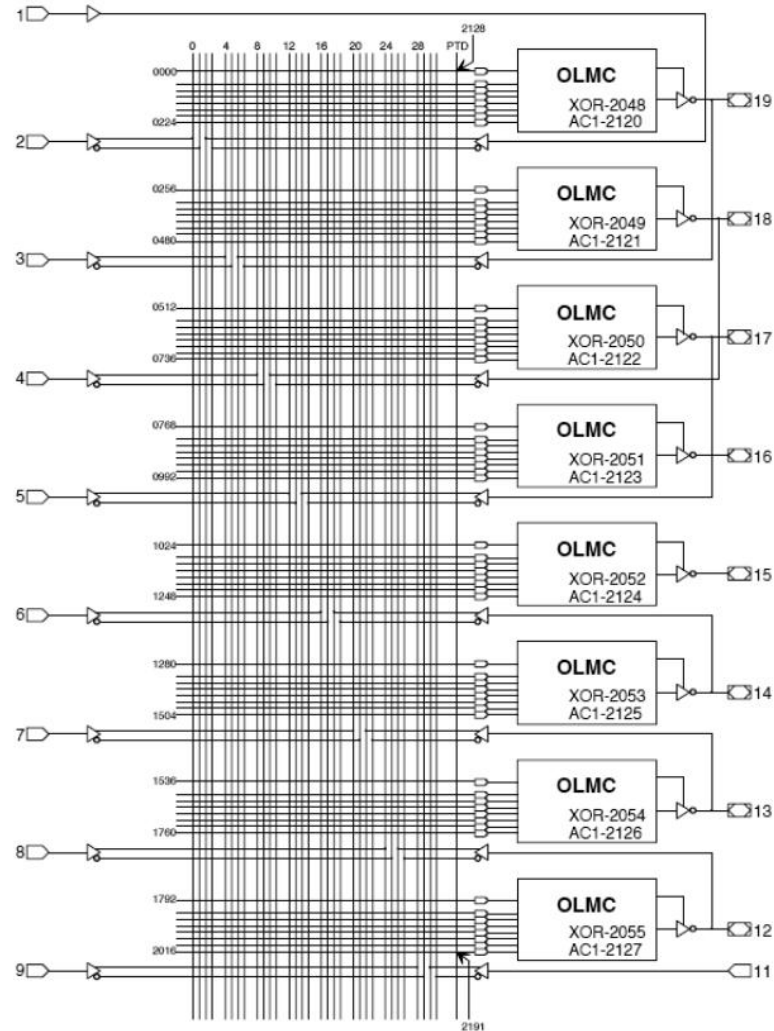
- ❑ Cambio función IN/OUT de un terminal



- ❑ Simple desconexión de la salida (*tristate buffer*)



# Esquemático de la GAL16V8



# Ejemplos comerciales y modo de programación PAL/GAL

.Para algunos fabricantes ya se consideran productos antiguos/demasiado básicos y sustituidos por los CPLDs

CPLD manufacturers.

Manufacturer	Series Name	Design Software	Website
Altera	MAX	Quartus II	Altera.com
Xilinx	Coolrunner	ISE Design Suite	Xilinx.com
Lattice	ispMACH	ispLEVER classic	Latticesemi.com
Atmel	ATF	ProChip Designer	Atmel.com

Ahora de Intel

Consultar posibilidades en un catálogo de uno de los ppales fabricantes de semiconductores:  
<http://www.latticesemi.com/>

# Lógica Programable - CPLD

•La “C” viene de *Complex*. Son dispositivos más sofisticados que las SPLDs.

•Son arquitecturas que básicamente consisten en múltiples SPLDs que pueden interconectarse.

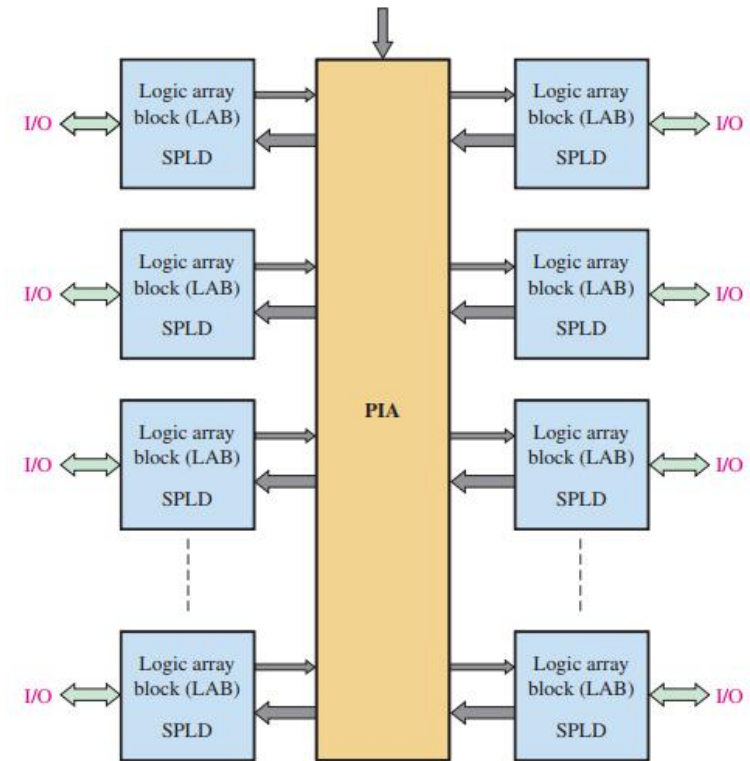
•Partes:

-PIA(*programmable interconnection array*), también llamada por algunos fabricantes AIM (*advanced interconnection matrix*)

-LAB (*logic array block*). Básicamente un SPLD.

•Son normalmente reprogramables.

•La programación (configuración de las conexiones) reside en algún tipo de memoria en el chip por lo que ya no son puramente *hardware*. También combinan una parte de *software*.



Consultar info sobre la Intel-Altera MAX 7000:

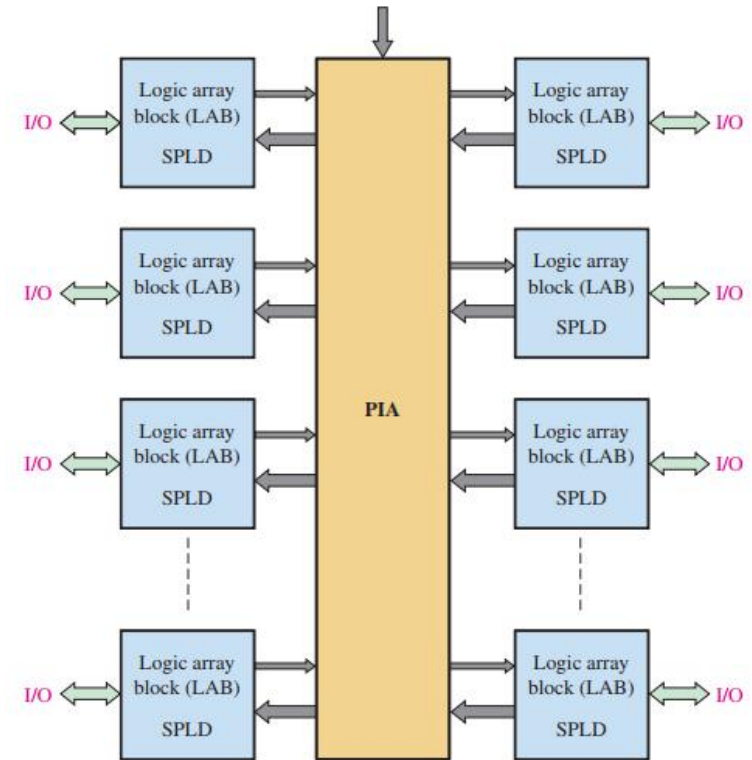
<https://www.intel.com/content/www/us/en/programmable/support/literature/lit-m7k.html>

# Lógica Programable - CPLD

•PIA = *programmable interconnect array*

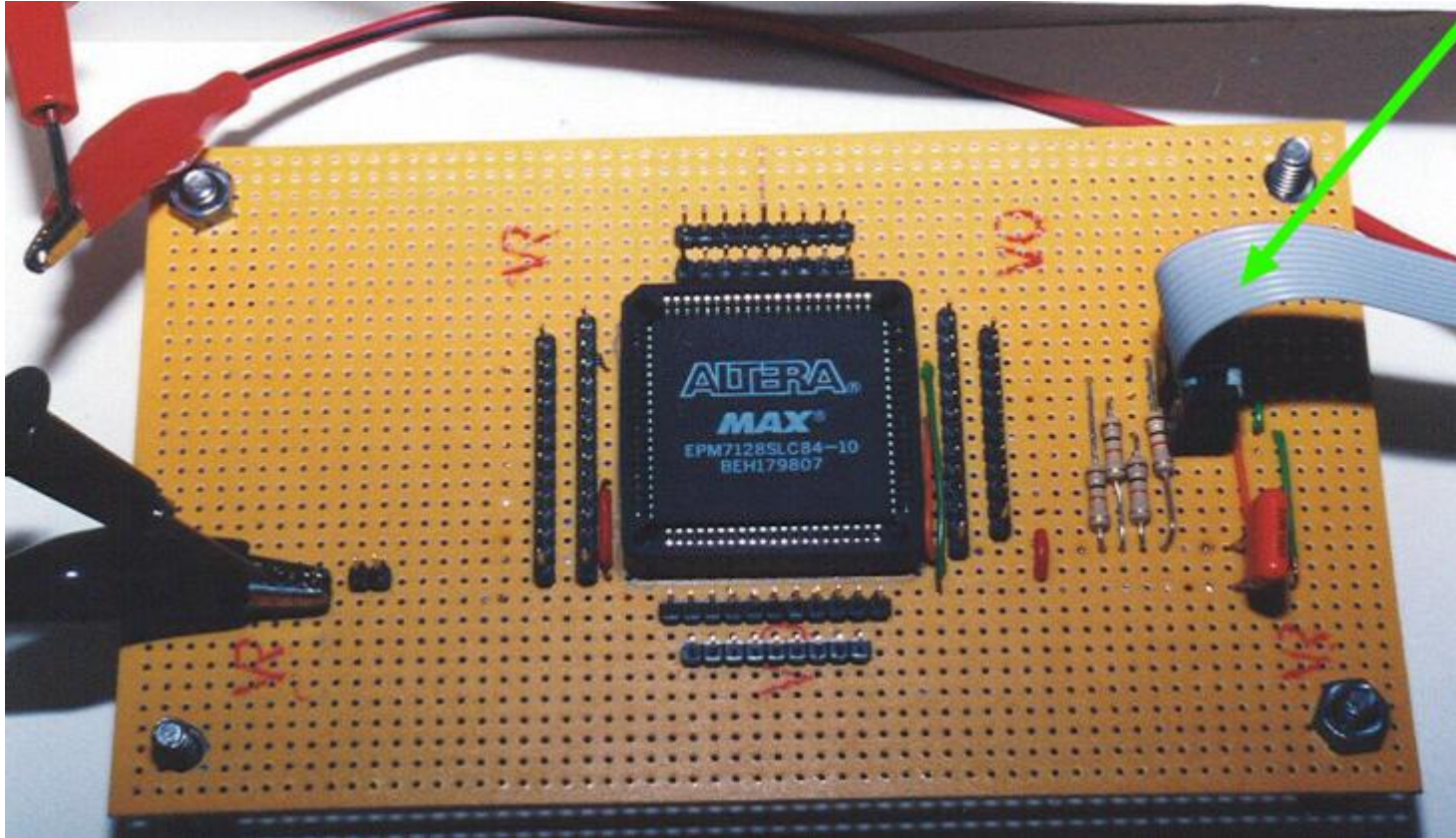
•La densidad de los CPLD puede ser bastante alta

- Cada LAB/SPLD puede tener dimensiones tan grandes como para implicar de docenas a MILES de macrocells
- Estamos hablando de chips con cientos de pines.



# Dispositivos reprogramables

Transferencia de las configuraciones



# Dispositivos reprogramables

¿Cómo mantienen los CPLD reprogramables las interconexiones?

Normalmente mediante algún tipo de memoria modificable: una EEPROM o SRAM

Ambos son sistemas no volátiles que mantendrán los estados aunque se quite potencia al circuito. Estos sistemas para mantener la configuración de las conexiones se utilizarán también en las FPGAs que serán analizadas más adelante.