

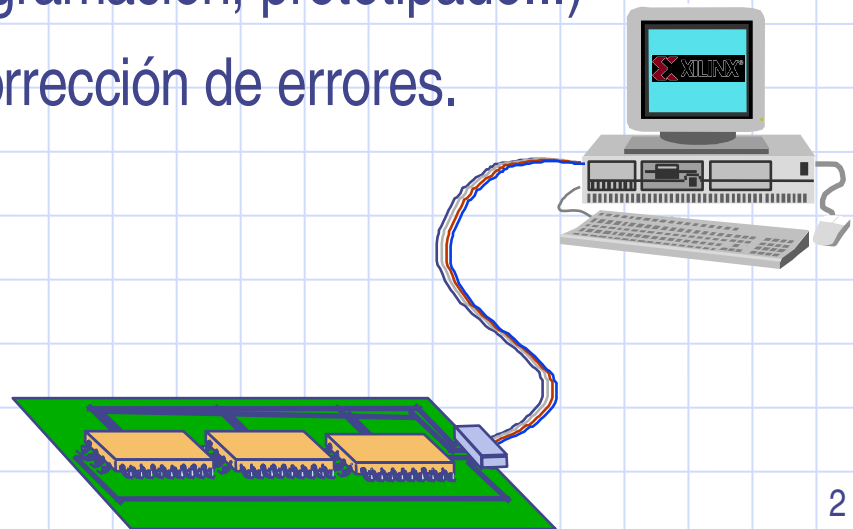
# Tema 1

## Dispositivos Lógicos Programables (PLD)

- **Introducción a los sistemas digitales**
  - ♦ Creación de un sistema digital
  - ♦ Partes de los sistemas digitales
  - ♦ Opciones de diseño
- **Dispositivos lógicos programables**
  - ♦ Criterios de clasificación
  - ♦ Tipos de dispositivos
- ***Field Programmable Gate Arrays (FPGA)***
  - ♦ Características fundamentales
  - ♦ Estructura y ejemplo
  - ♦ Ventajas e inconvenientes

# Creación de un sistema digital

- ◆ Idea, necesidad, oportunidad...
- ◆ Análisis de requisitos y especificación.
- ◆ Diseño (Captura de Esquemáticos, HDLs y otros).
- ◆ Simulación (verificación del circuito).
- ◆ Implementación (PCB, programación, prototipado...)
- ◆ Depuración, ampliación, corrección de errores.
- ◆ Fabricación.



# Partes de los sistemas digitales



# Opciones de diseño

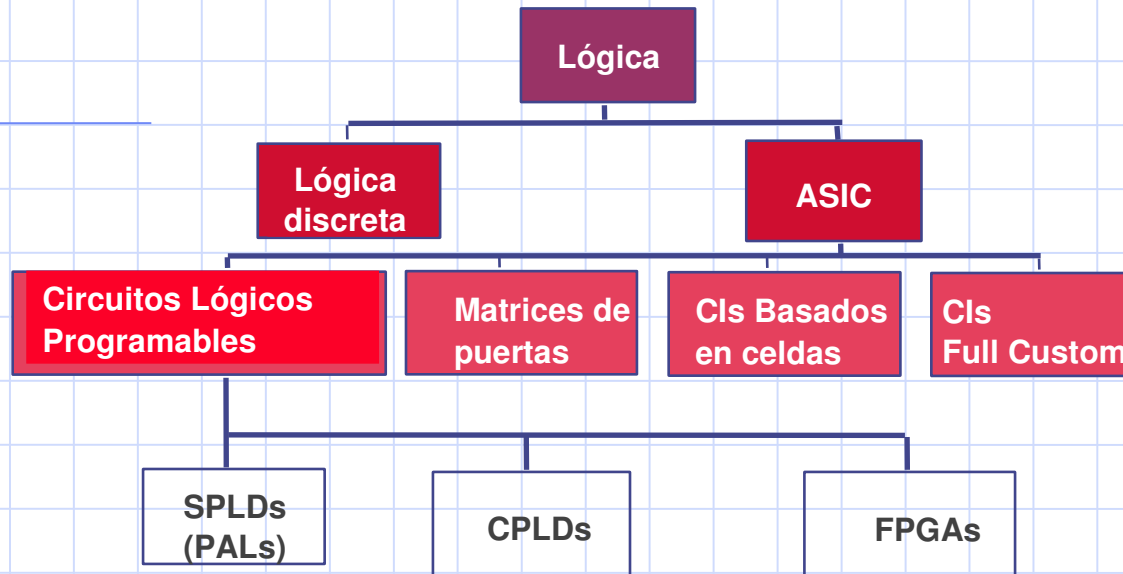
## ◆ Microcontrolador:

- Ventajas, inconvenientes...

## ◆ Lógica programable:

- Ventajas, inconvenientes...

# Diseño de lógica no programable



## Acrónimos

SPLD = PLDs Simples.

PAL = Matr. de Lógica Prog.

CPLD = PLDs Complejos.

FPGA = Circuito Prog. Usuario.

## Recursos Comunes

◆ Bloques Lógicos Configurables (CLB).

- Memoria por Tabla de búsqueda.
- Planos AND - OR
- Puertas Simples.

◆ Bloques Entrada / Salida (IOB).

- Bidireccional, latches, inversores, pullup/pulldowns.

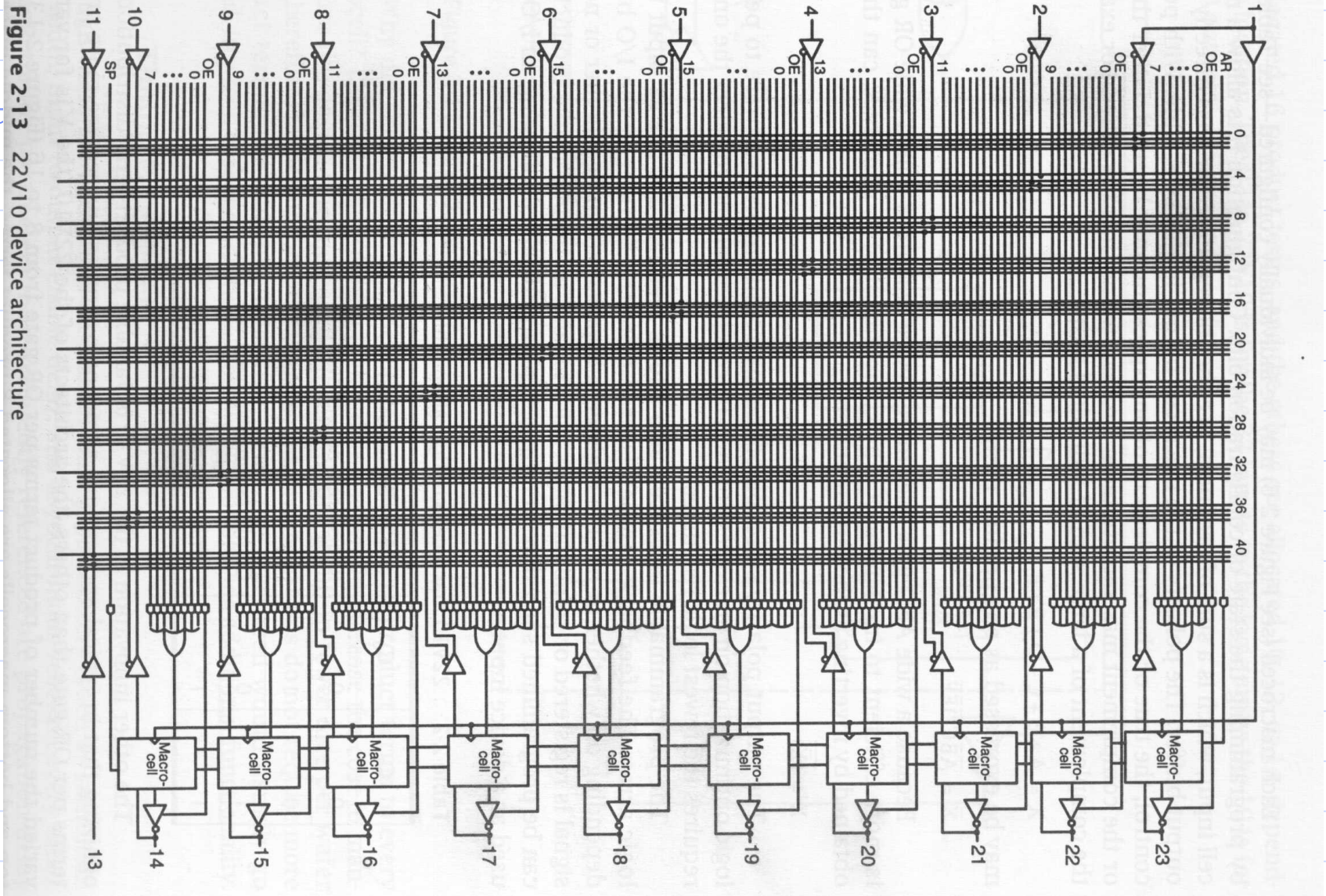
◆ Interconexión o Cableado.

- Realimentación interna local y global.

# Dispositivos lógicos programables

- ◆ Funciones lógicas configurables.
- ◆ Más versátiles que la lógica discreta.
- ◆ Mismo encapsulado, más...
- ◆ Estandarización, mejora con la integración.
- ◆ Nuevas formas de diseño lógico: HDL

# Estructura de una GAL: 22V10



# Clasificación según tecnología de grabación

## ◆ Programables una vez.

- Fusibles.
- Antifusibles.

## ◆ Reprogramables no volátiles.

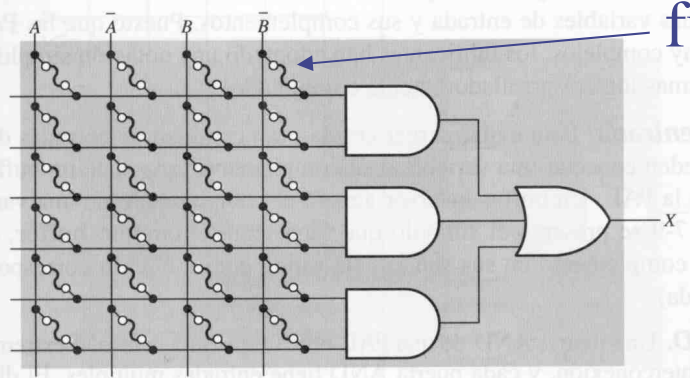
- Transistores MOS de puerta flotante.

## ◆ Reprogramables volátiles.

- Celdas SRAM.

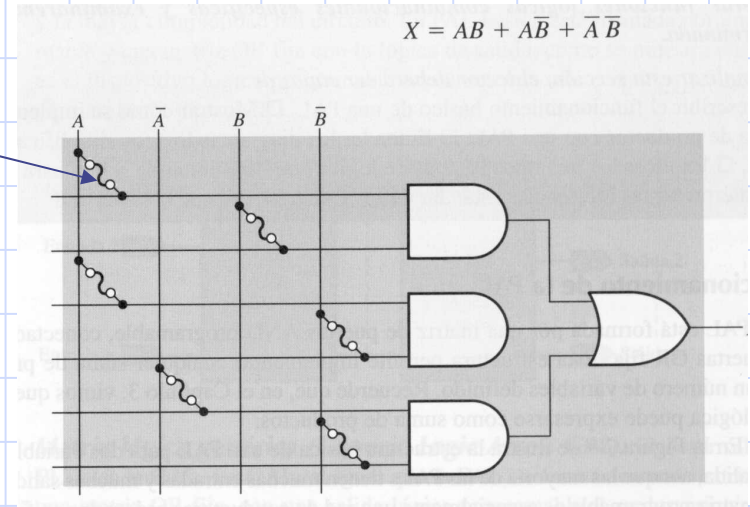


# Ejemplo de PLD



SIN PROGRAMAR

fusibles



PROGRAMADA

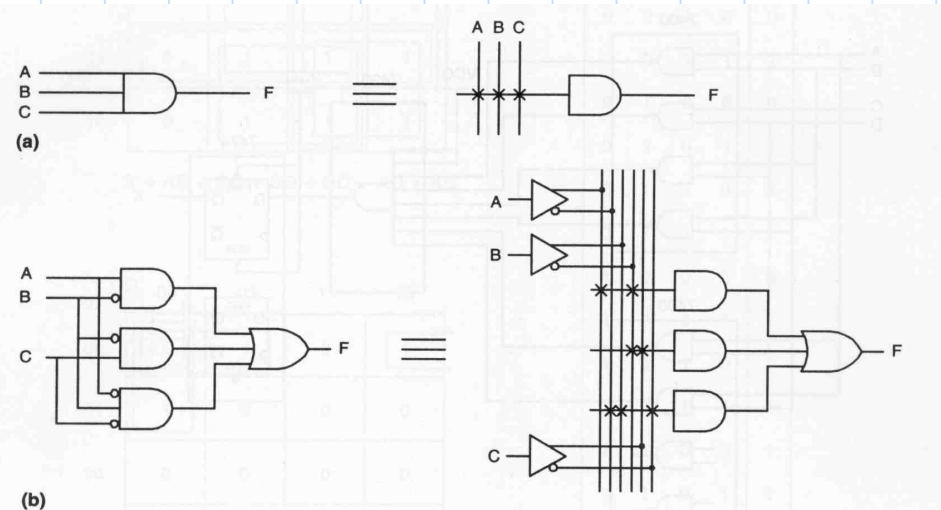


Figure 2-4 (a) Standard gate symbol and equivalent PAL diagram; (b) schematic and PAL diagram equivalence

# Antifusibles

◆ Antifusible: *Circuito abierto* que mediante programación genera un *corto circuito*

➤ Tipos OTP (One Time Programming).

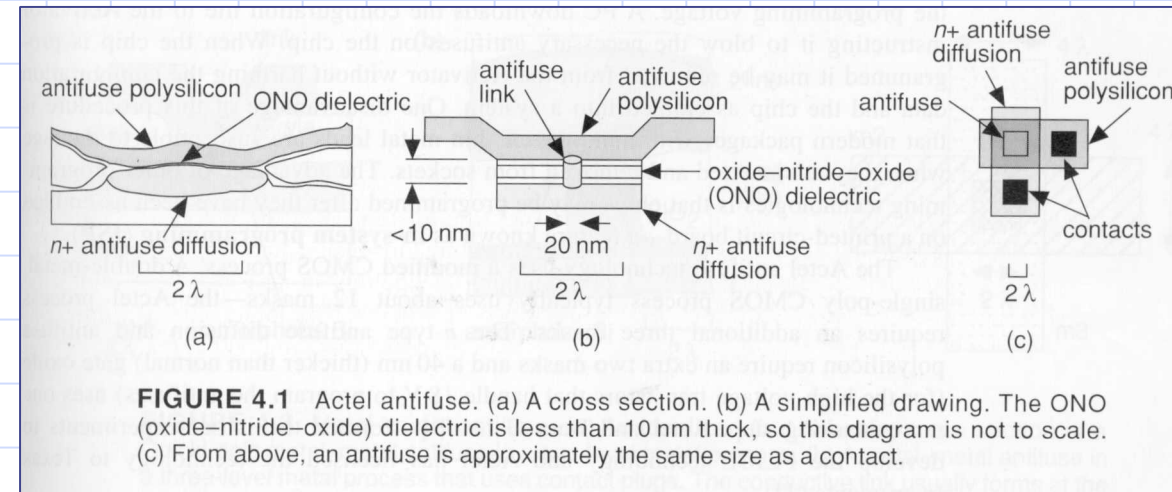
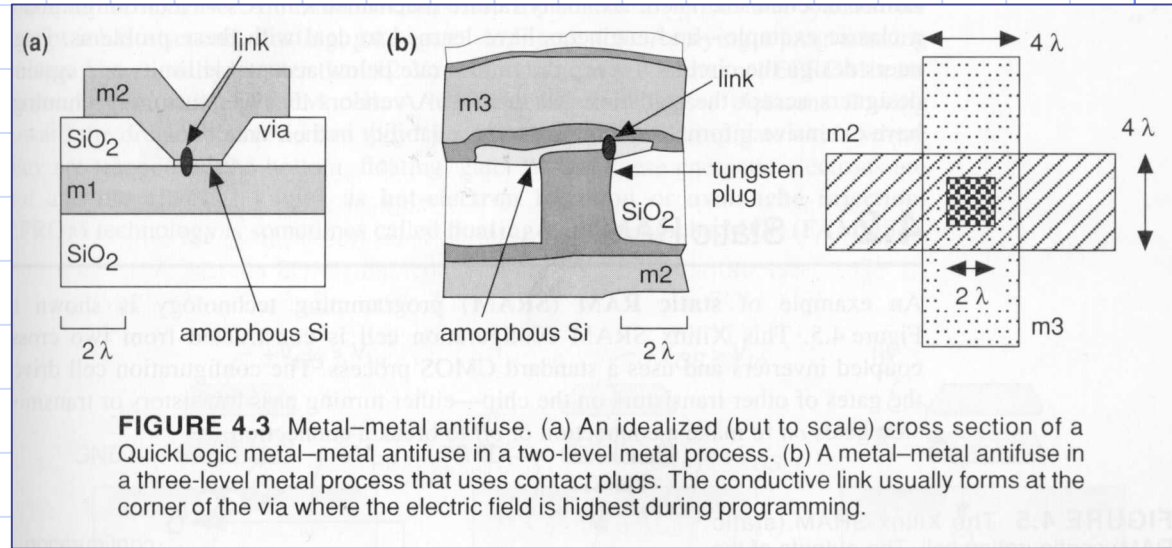
➤ Alta densidad de integración.

➤ Basada en Multiplexor.

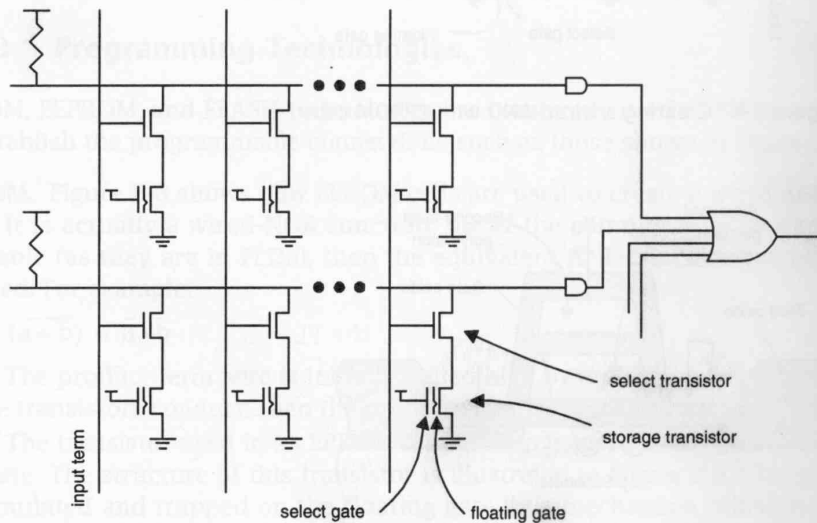
Tipos

Basados en a-Si

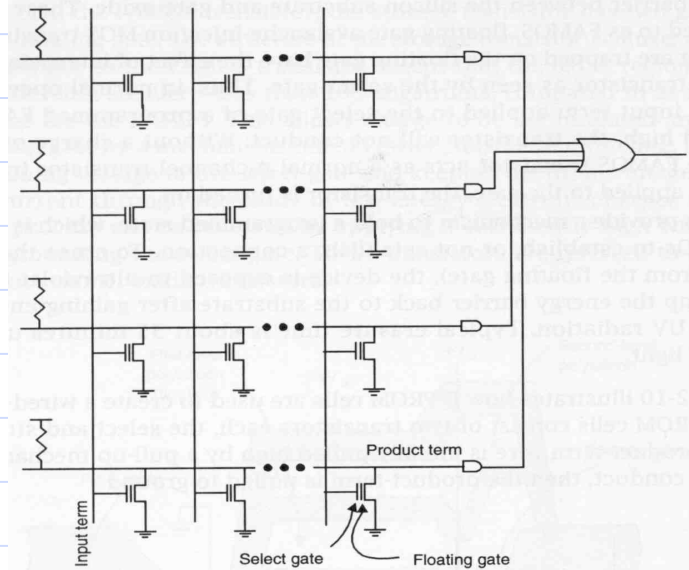
Basados en ONO



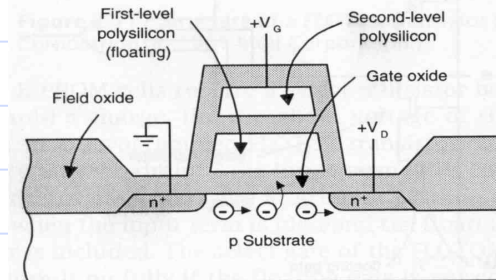
# Tecnologías EPROM y EEPROMs



**Figure 2-10** Creating a wired-AND with E<sup>2</sup>PROM cells



**Figure 2-8** Creating a wired-AND with EPROM cells



**Figure 2-9** Structure of a FAMOS transistor [Reprinted by permission of Intel Corporation. © 1981 Intel Corporation.]

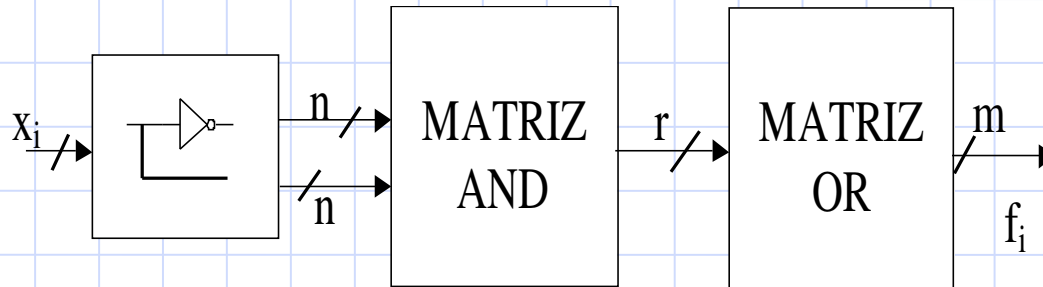
# Clasificación según funciones

- ◆ Matrices AND - OR.
- ◆ Matrices AND - OR + registros.
- ◆ Matrices + bloques E/S.
- ◆ PLD simples interconectados.
- ◆ Estructuras jerárquicas complejas.
  - Interconexión.
  - Bloques lógicos (LUT).
  - Bloques E/S.

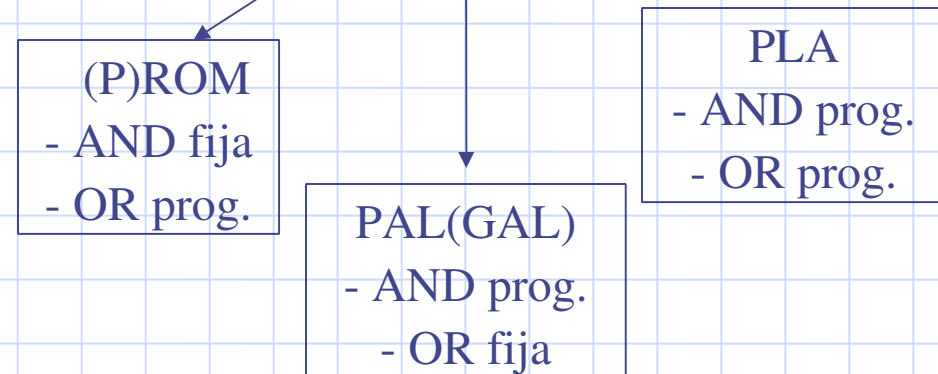
# Dispositivos Programables

C.I.	Fabricante	Año	Características
<b>PLA</b>	Philips	1970	Planos AND y OR Programables.
<b>PAL</b>	MMI (AMD)	1978	Plano AND programable y plano OR Fijo.
<b>EPLD</b>	Altera	1980	Borrables UV. Tipo PAL con registros. Salida versátil (VPAL).
<b>FPGA</b>	Xilinx	1984	Matrices de lógica programable por conexiones controladas por SRAM en el propio circuito. <u>Grano fino</u> : Puertas y/o multiplexores + registros. <u>Grano grueso</u> : Funciones complejas de varias variables y registros.
<b>CPLD</b>	Xilinx, Altera, etc	1995	Tipo VPAL, Borrable Flash-EEPROM y programable en circuito, conexiones programables (PIA).

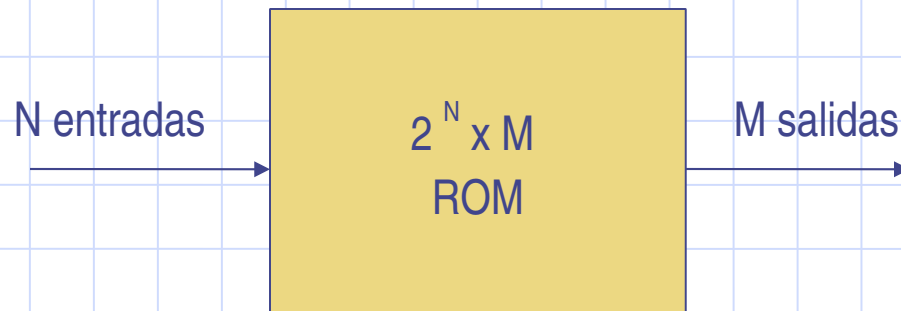
# SPLD (Simple PLD)



## Tipos de PLD:

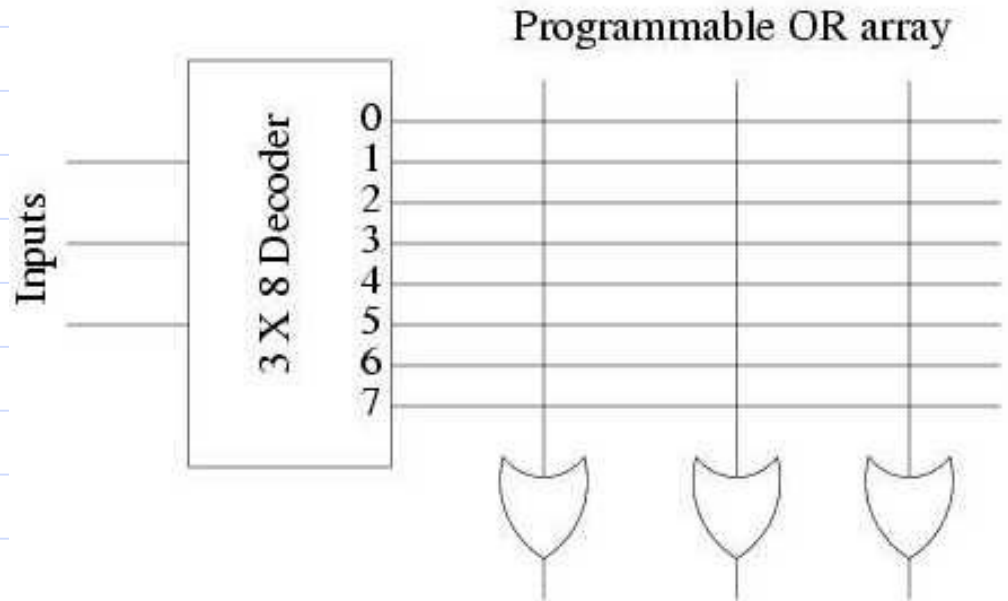
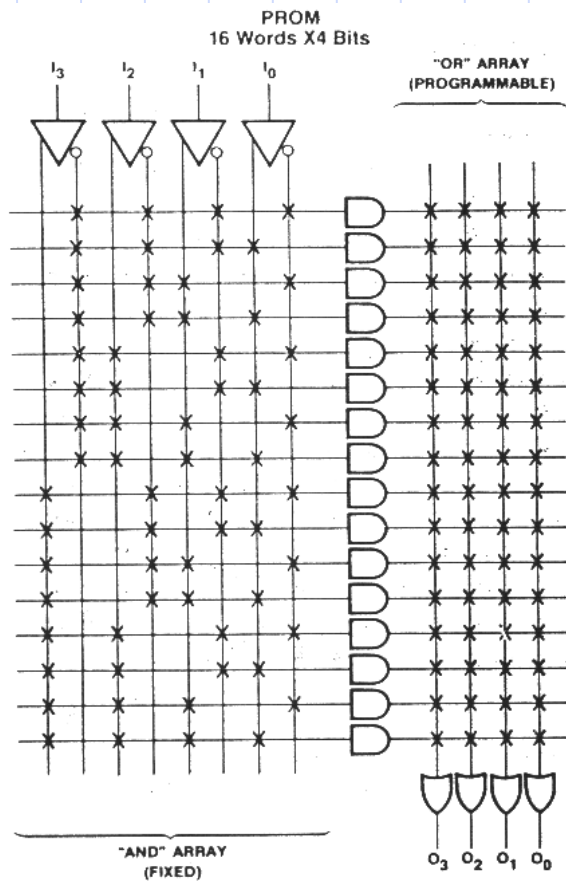


# Programmable ROM (PROM)



- Direcciones: N bits; Palabra de salida: M bits.
- ROM contiene  $2^N$  palabras de M bits.
- Los bits de entrada deciden la palabra particular que estará disponible en las líneas de salida.

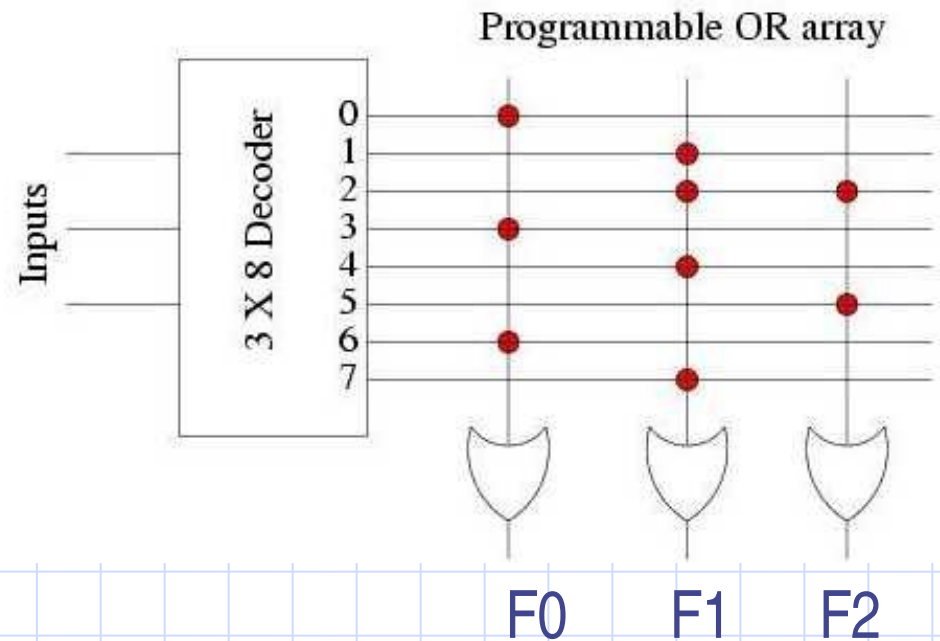
# Diagrama lógico de una PROM





# Circuito combinacional implementado usando PROM

I0	I1	I2	F0	F1	F2
0	0	0	1	0	0
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	1	0
1	0	1	0	0	1
1	1	0	1	0	0
1	1	1	0	1	0

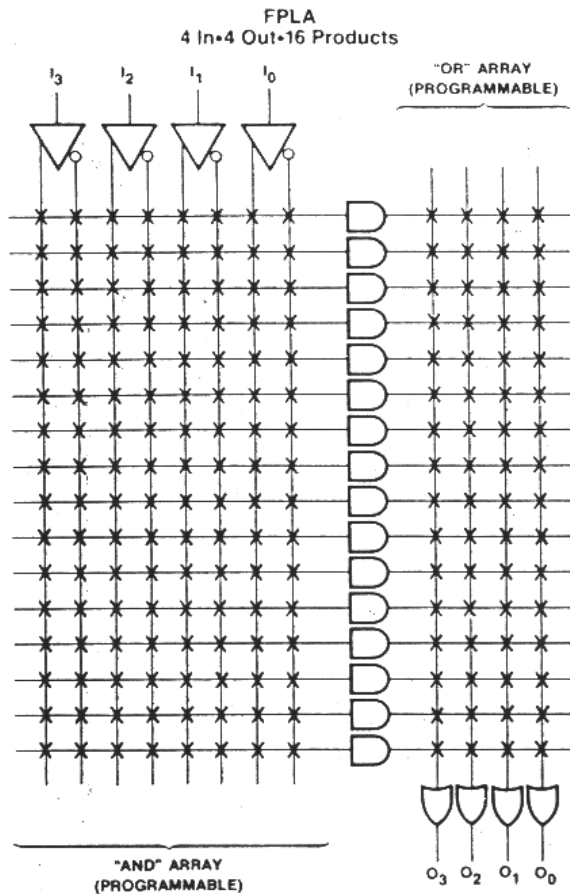


# PLA: Programmable Logic Array

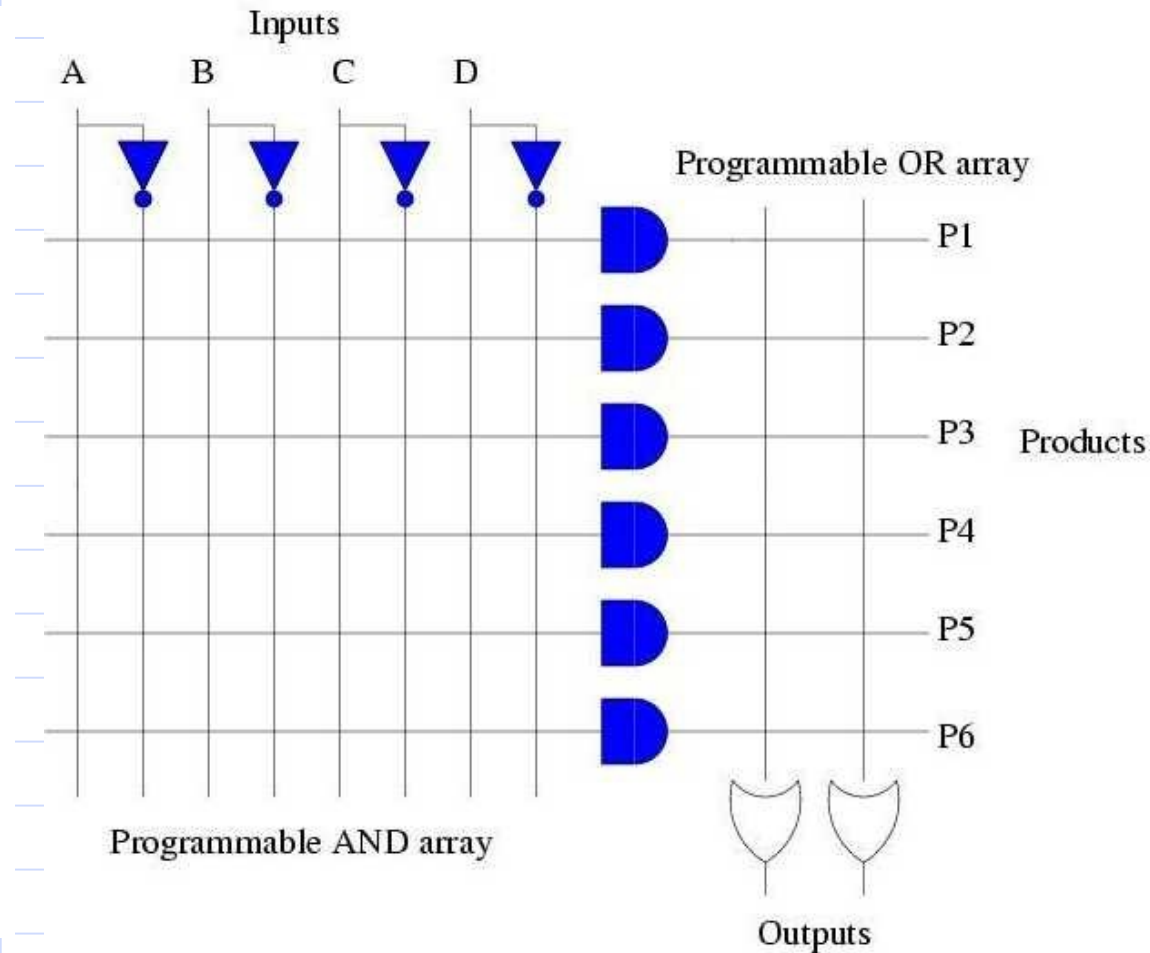


- Matriz programable de ANDs y de ORs.
- $n \times k \times m$  PLA tiene  $2n \times k + k \times m$  conexiones.
- Suma de productos.

# PLA



4 x 16 x 4

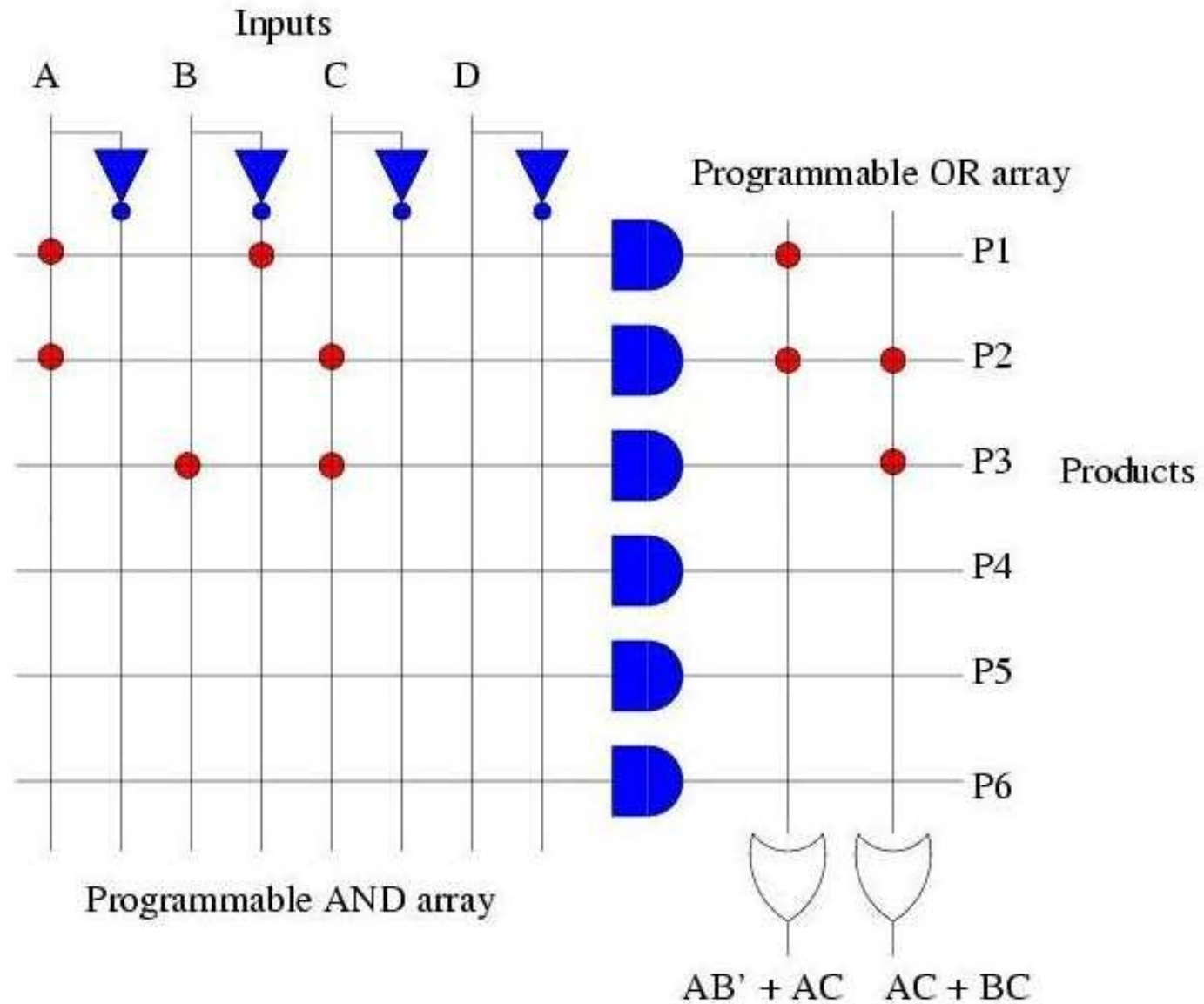


4 x 6 x 2

# Implementación lógica con PLA

- ◆ Número finito de puertas AND  $\Rightarrow$  simplificar la function al mínimo número de terminos productos.
- ◆ El número de variables en términos productos no es importante dado que tenemos todas las variables de entrada.
- ◆ Compartir términos entre diferentes OR  $\Rightarrow$  minimización de múltiples funciones.

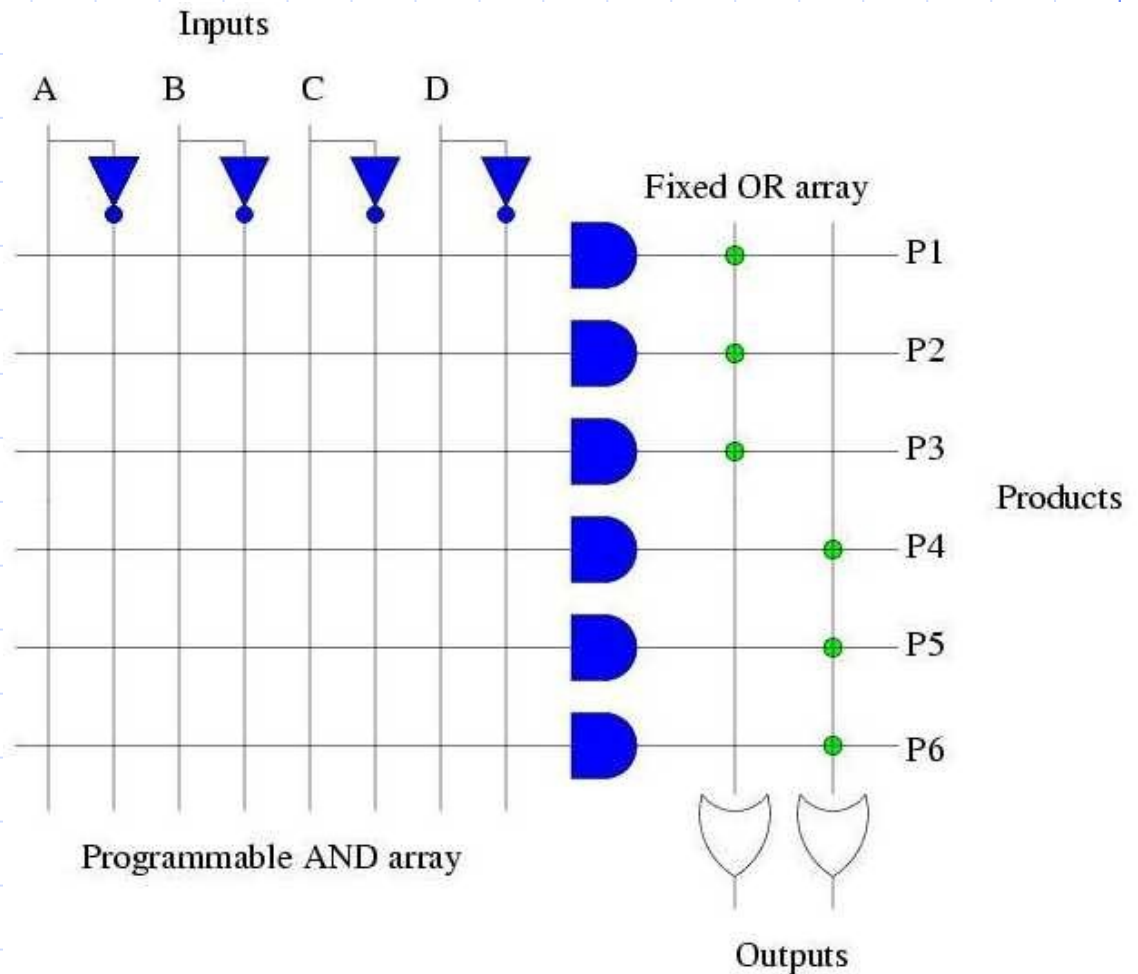
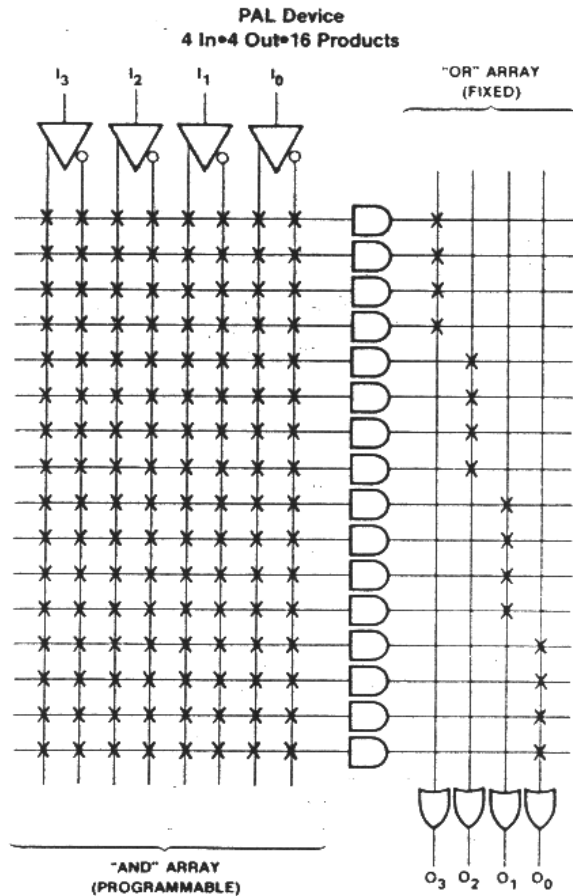
# Diseño con PLA



# PAL: Programmable Array Logic

- ◆ Matriz de ANDs programable.
- ◆ Matriz de ORs fija.
  - Cada línea salida está permanentemente conecta a un conjunto específico de términos productos.
- ◆ El número de funciones que pueden implementarse con PAL es mucho menor que con PROM y PLA.
- ◆ Mayor densidad y menores retardos que las PLAs.

# Diagrama Lógico PAL

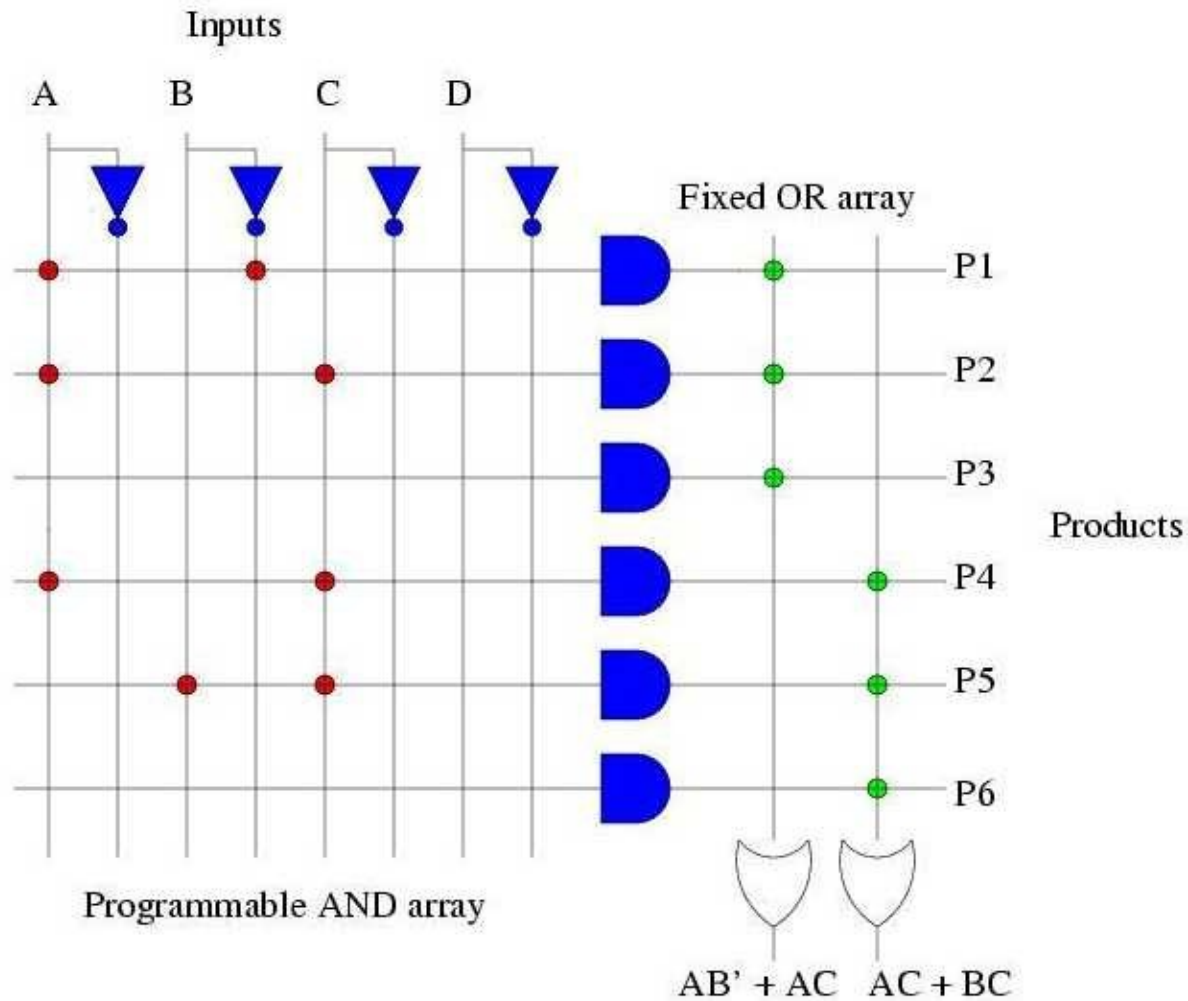


# Implicaciones PAL

- ◆ Número de términos productos por salida ha de ser mayor que el número de términos productos por cada suma de productos.
- ◆ No se pueden compartir términos productos entre diferentes salidas.



# Diseño con PAL



# Diagrama de bloques de una GAL: 22V10

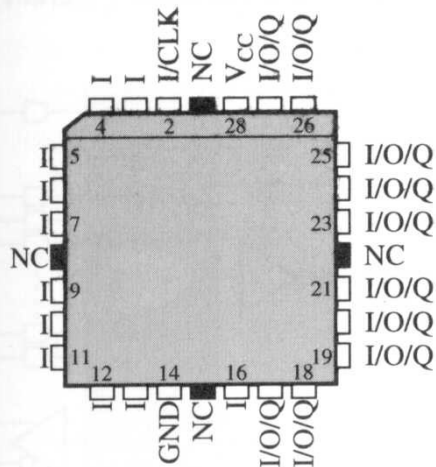
◆ Es el SPLD más popular

22V10

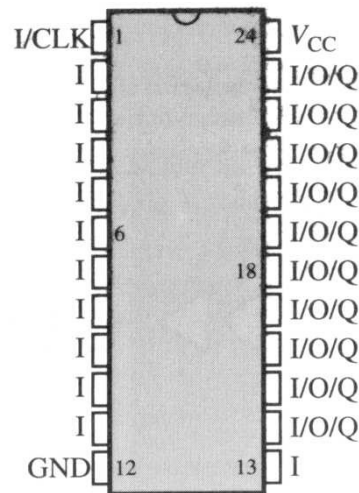
entradas  
o salidas

salidas

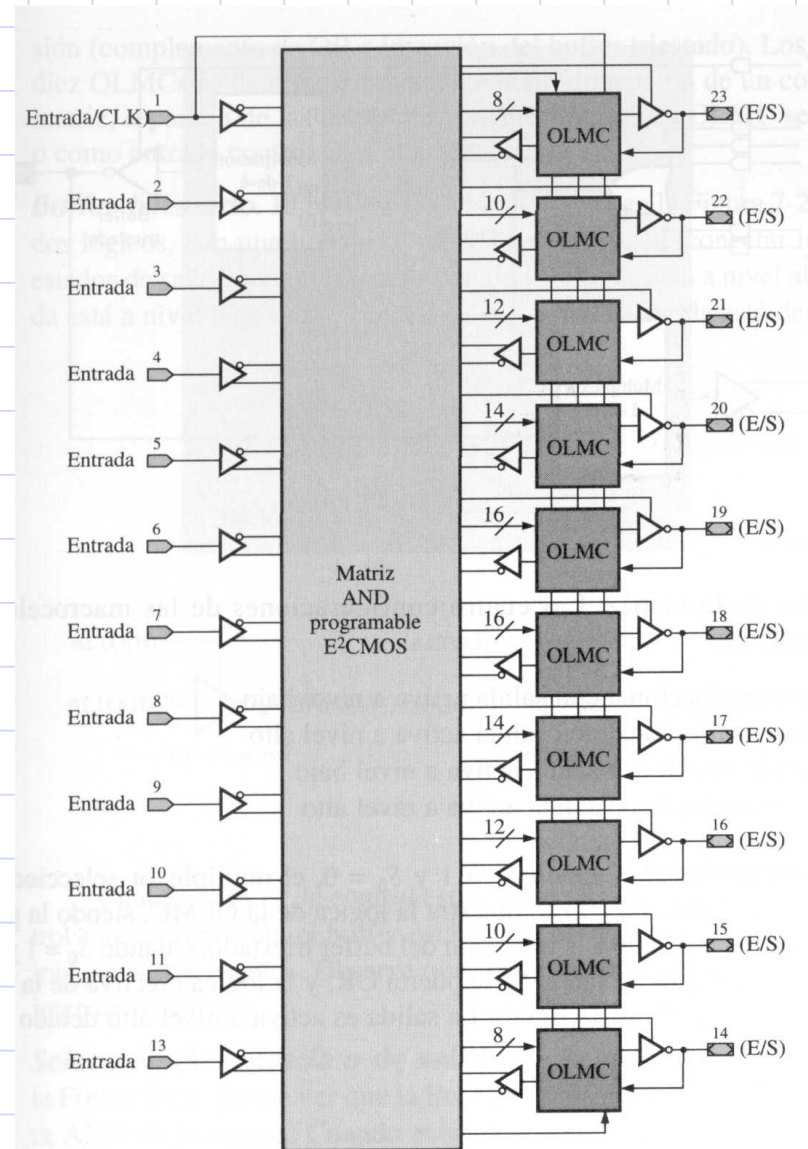
tipo de salida



(a) PLCC



(b) DIP



# Macroelda de salida

$C_1$	$C_0$	Modo de salida
0	0	Registro (activo 0)
0	1	Registro (activo 1)
1	0	Comb. (activo 0)
1	1	Comb. (activo 1)

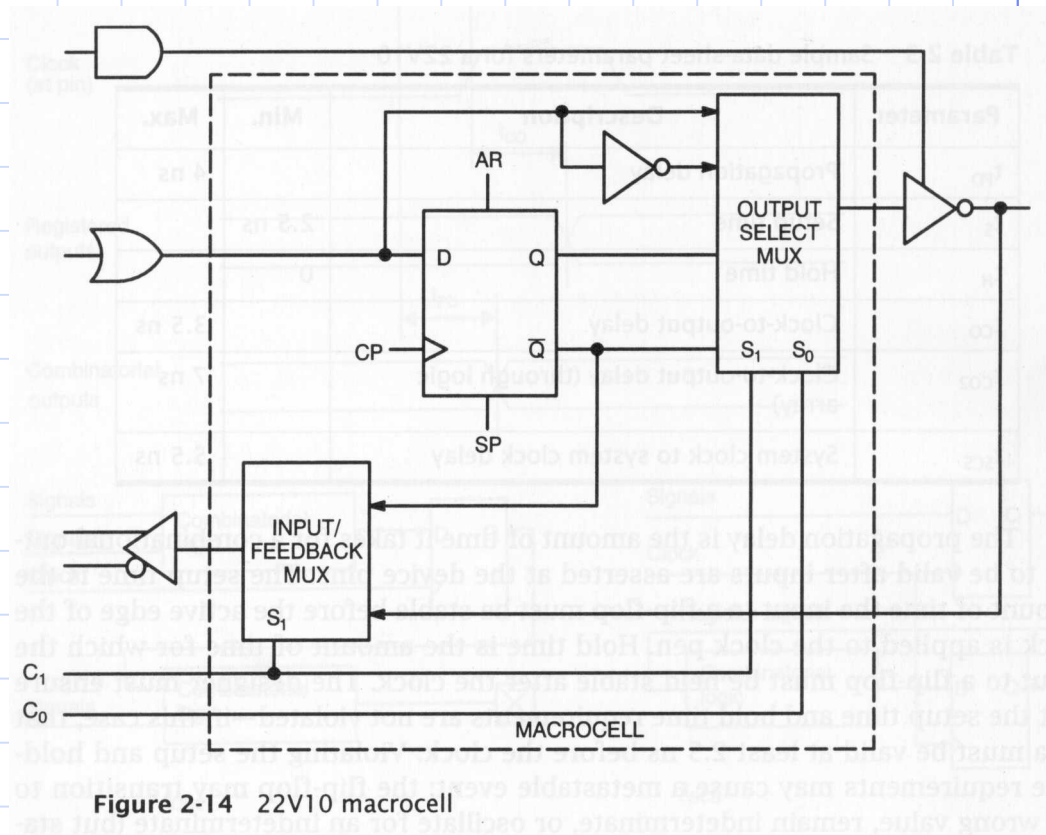


Figure 2-14 22V10 macrocell

# Retardos en una 22V10

- ◆  $t_{PD}$ : tiempo que tarda en cambiar una salida a partir de una entrada valida.
- ◆  $t_{CO}$ : tiempo que tarda en cambiar una salida a partir de un flanco de la señal de reloj.
- ◆  $t_{SU}$ : tiempo mínimo que los datos deben estar estables **antes** del flanco de la señal de reloj.
- ◆  $t_H$ : tiempo mínimo que los datos deben estar estables **después** del flanco de la señal de reloj.
- ◆  $t_{SCS}$ : periodo mínimo de reloj en operaciones registro/registro.
- ◆  $f_{max} = 1/t_{SCS}$

Table 2-3 Sample data-sheet parameters for a 22V10

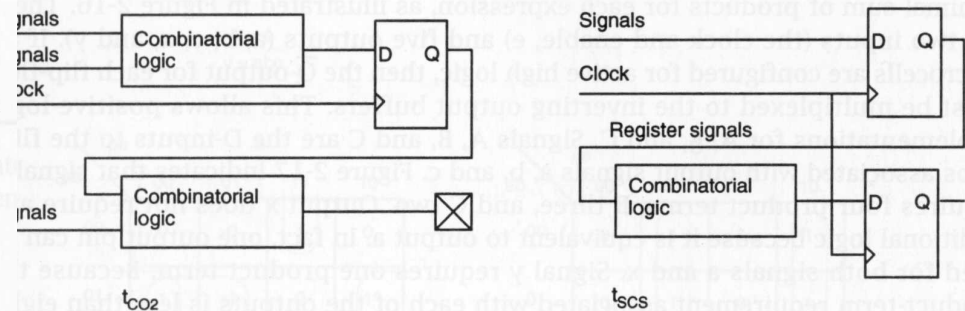
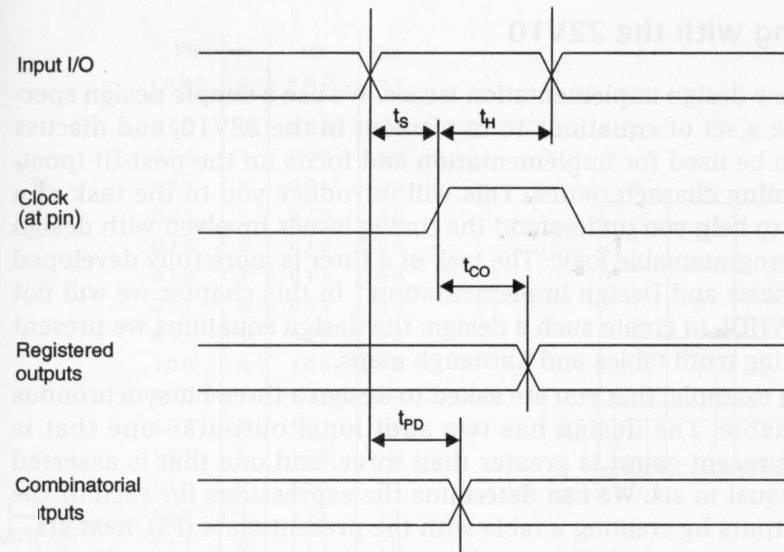
Parameter	Description	Min.	Max.
$t_{PD}$	Propagation delay		4 ns
$t_S$	Setup time	2.5 ns	
$t_H$	Hold time	0	
$t_{CO}$	Clock-to-output delay		3.5 ns
$t_{CO2}$	Clock-to-output delay (through logic array)		7 ns
$t_{SCS}$	System clock to system clock delay		5.5 ns

# Retardos en una 22V10 (2)



**Table 2-3** Sample data-sheet parameters for a 22V10

Parameter	Description	Min.	Max.
$t_{PD}$	Propagation delay		4 ns
$t_S$	Setup time	2.5 ns	
$t_H$	Hold time	0	
$t_{CO}$	Clock-to-output delay		3.5 ns
$t_{CO2}$	Clock-to-output delay (through logic array)		7 ns
$t_{SCS}$	System clock to system clock delay		5.5 ns



**Figure 2-15** Timing parameters

# De los SPLDs a los CPLDs

Ante el avance de la tecnología,

- ¿Por qué los fabricantes no escalaron las arquitecturas de la 16V8 y 22V10?
- ¿Por qué no pasar del 16V8 al 128V64?

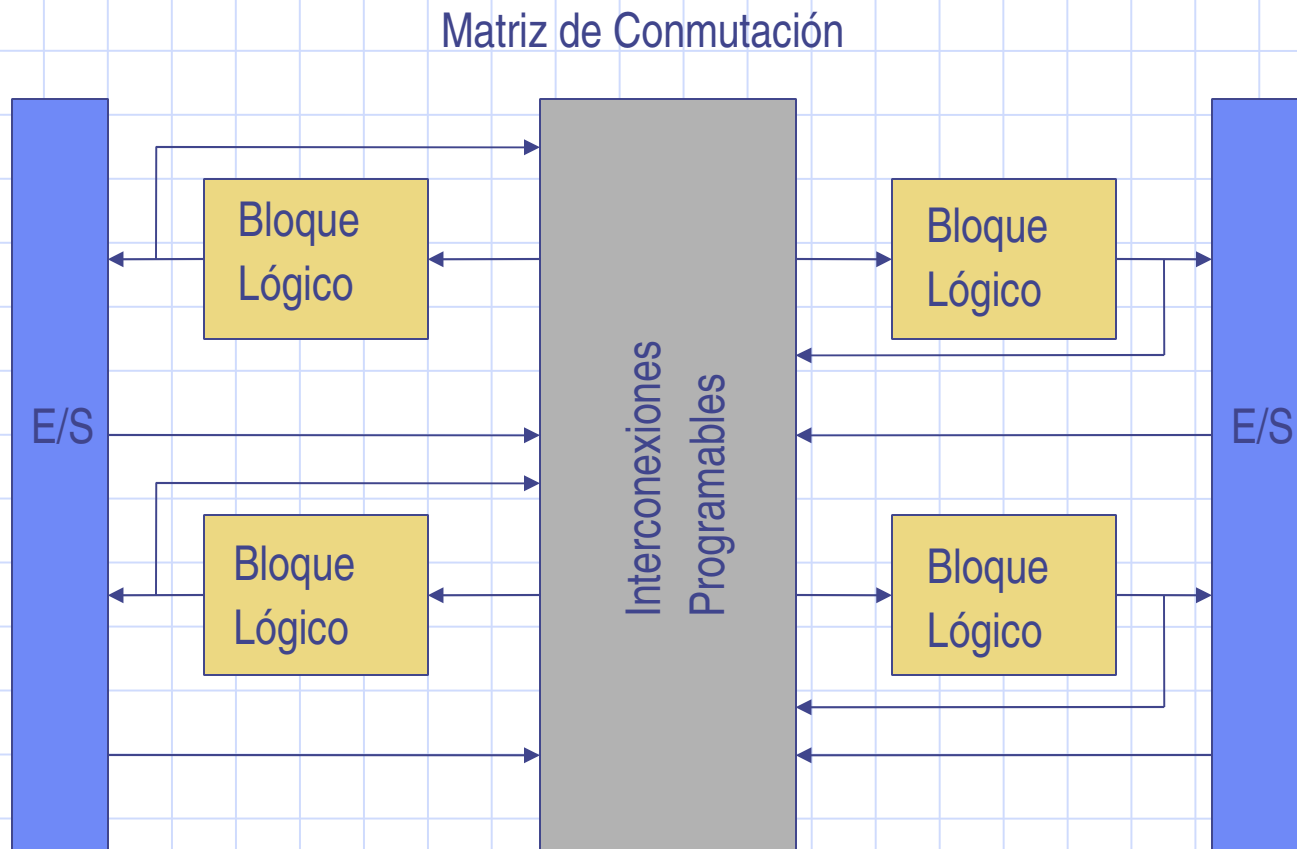
Características del hipotético 128V64:

- 64 entradas primarias/64 pines bidireccionales.
- 128 variables por término productos (fan-in de puertas AND).

Algunas limitaciones del hipotético 128V64

- Puertas AND al menos 8 veces más lentas que las del 16V8.
- El área de silicio se multiplica por 64.
- El número de E/S sólo se multiplica por 8.

# CPLD



# Bloques lógico CPLD

## ◆ PLD simple

- Entradas.
- Matriz de términos productos.
- Colocador de términos productos.
- Macro-celdas (registros).

◆ Bloques lógicos ejecutan expresiones de suma de productos, almacenan los resultados en microceldas.

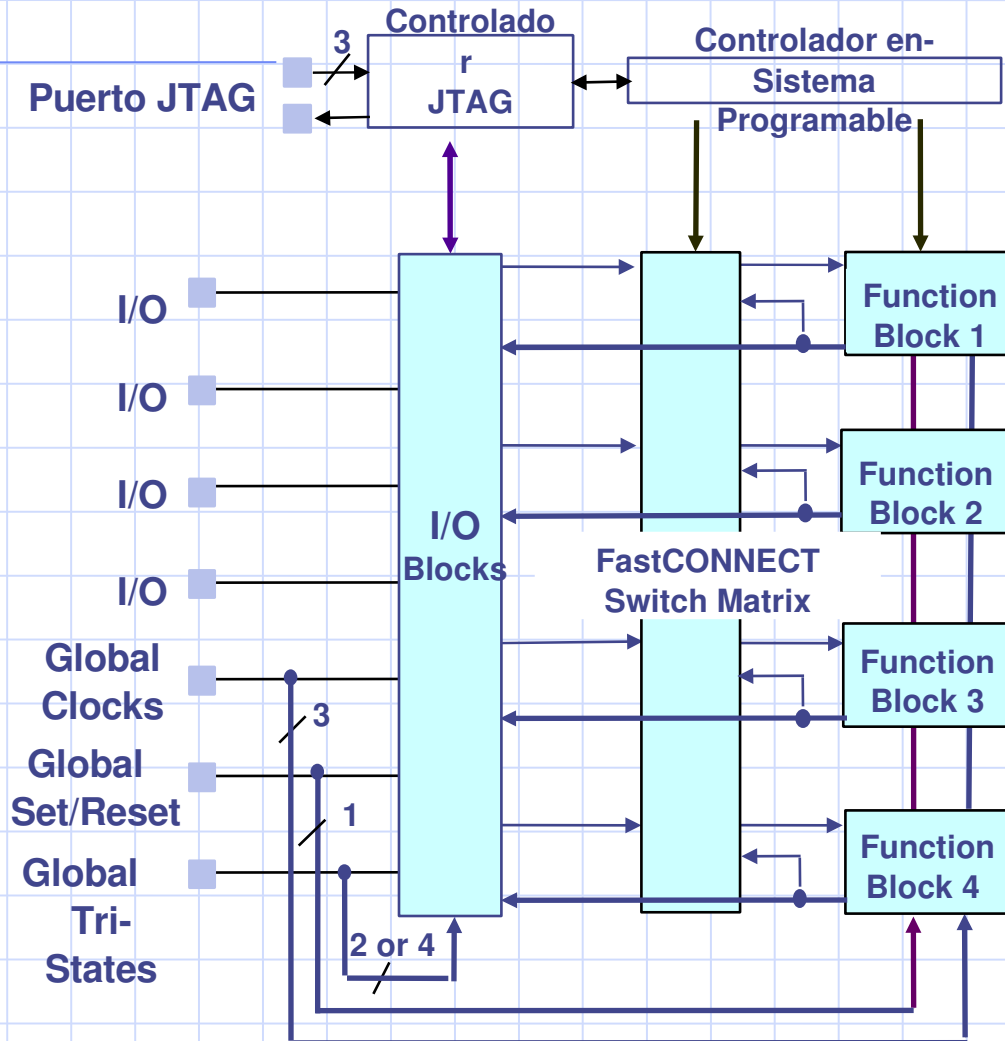
◆ Interconexión programable establece la ruta de las señales de y desde los bloques lógicos.



# Principales recursos CPLD

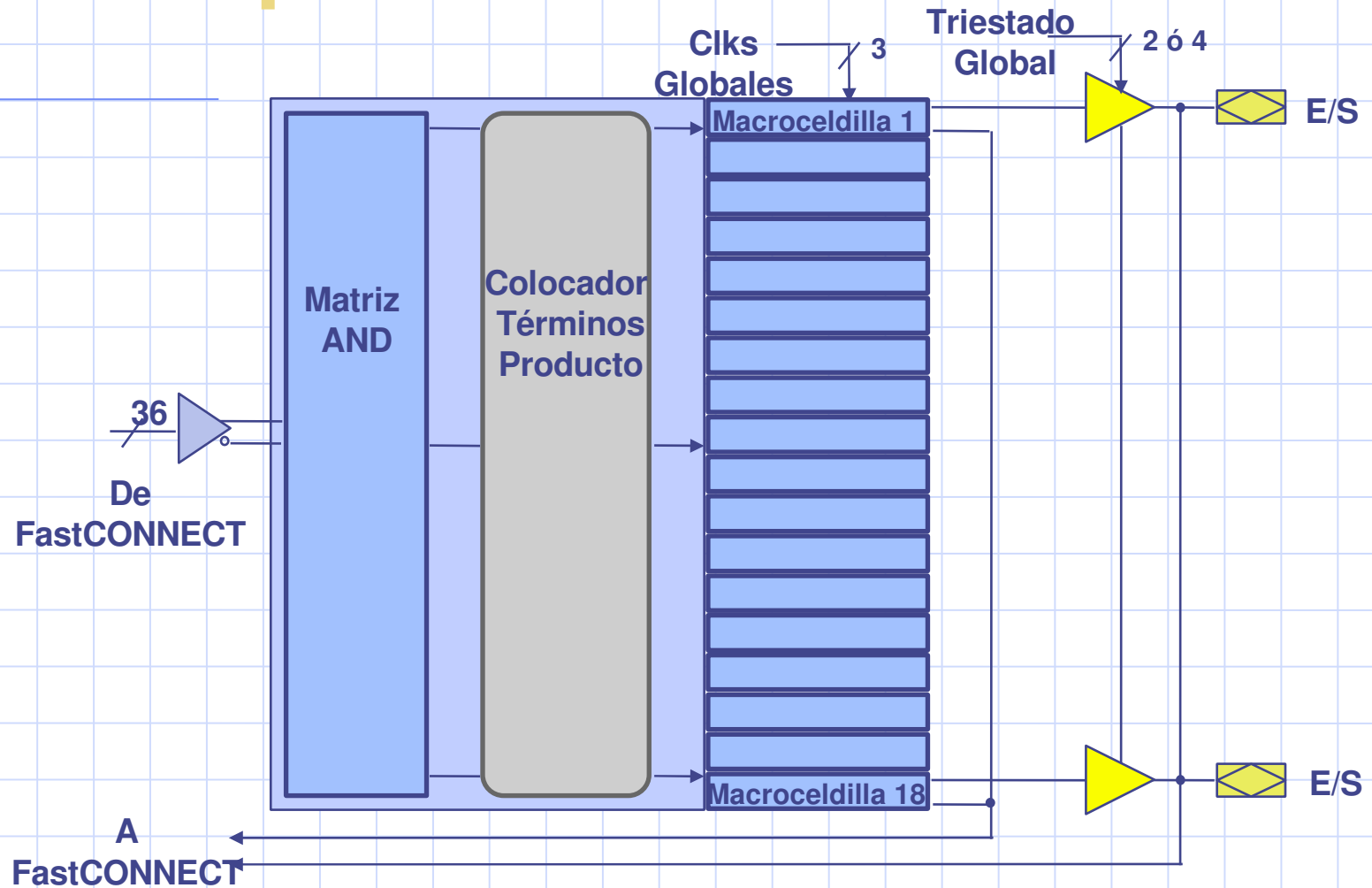
- ◆ Número de macroceldas por bloque lógico.
- ◆ Número de entradas de las interconexiones programables a los bloques lógicos.
- ◆ Número de términos producto en los bloques lógicos.

# CPLDs *FastFLASH*™ XC9500



- ◆ **CPLDs Programables en circuito impreso (ISP).**
- ◆ **Retardos de 5 ns.**
- ◆ **36 a 576 macroceldas.**
- ◆ **La mejor arquitectura del mercado para asignación de pines.**
- ◆ **10,000 ciclos programación/borrado.**
- ◆ **Soporta JTAG: IEEE 1149.1.**

# Bloque de Función XC9500



¡ Cada bloque de función es como una 36V18 !

# Familia MAX7000 (ALTERA)

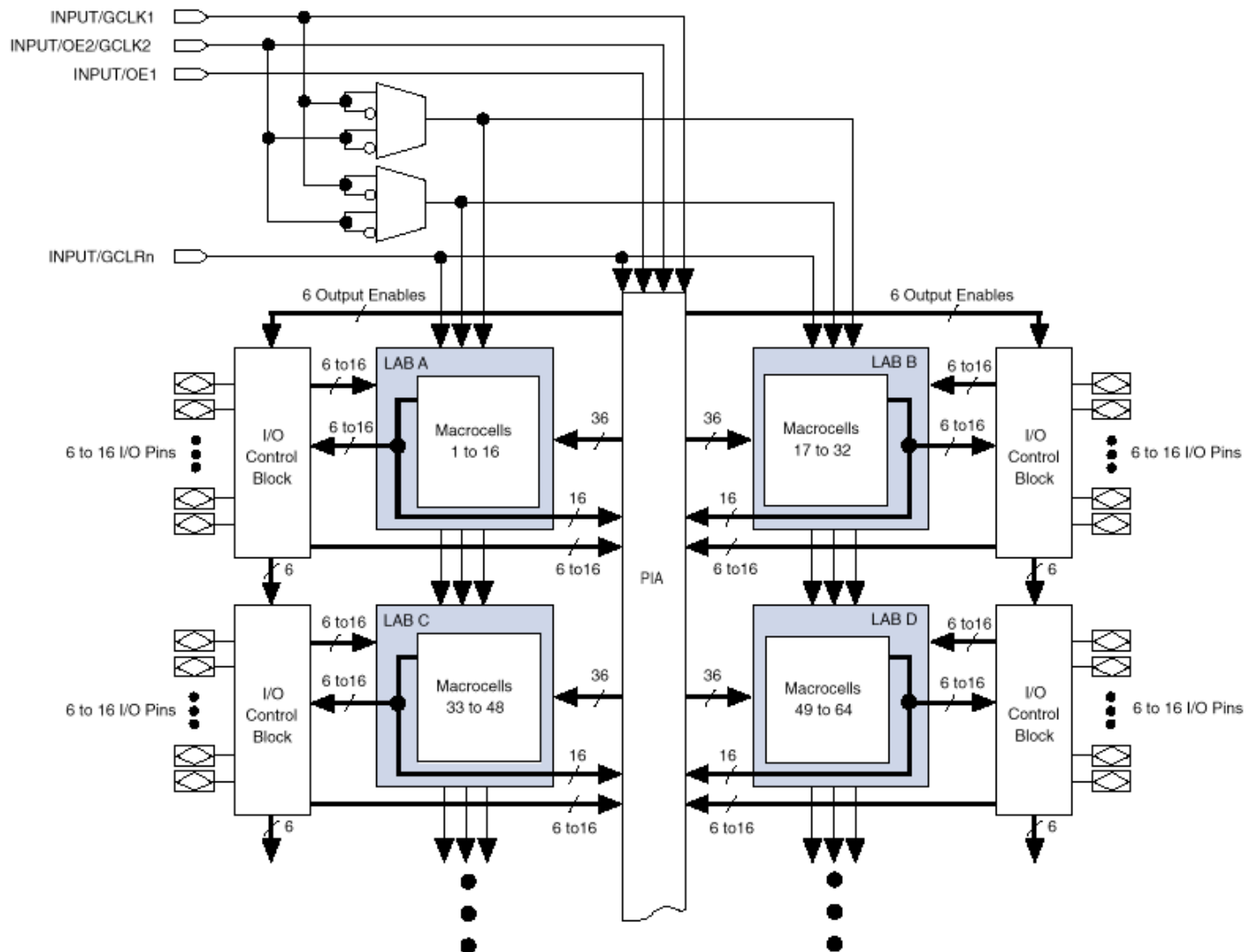
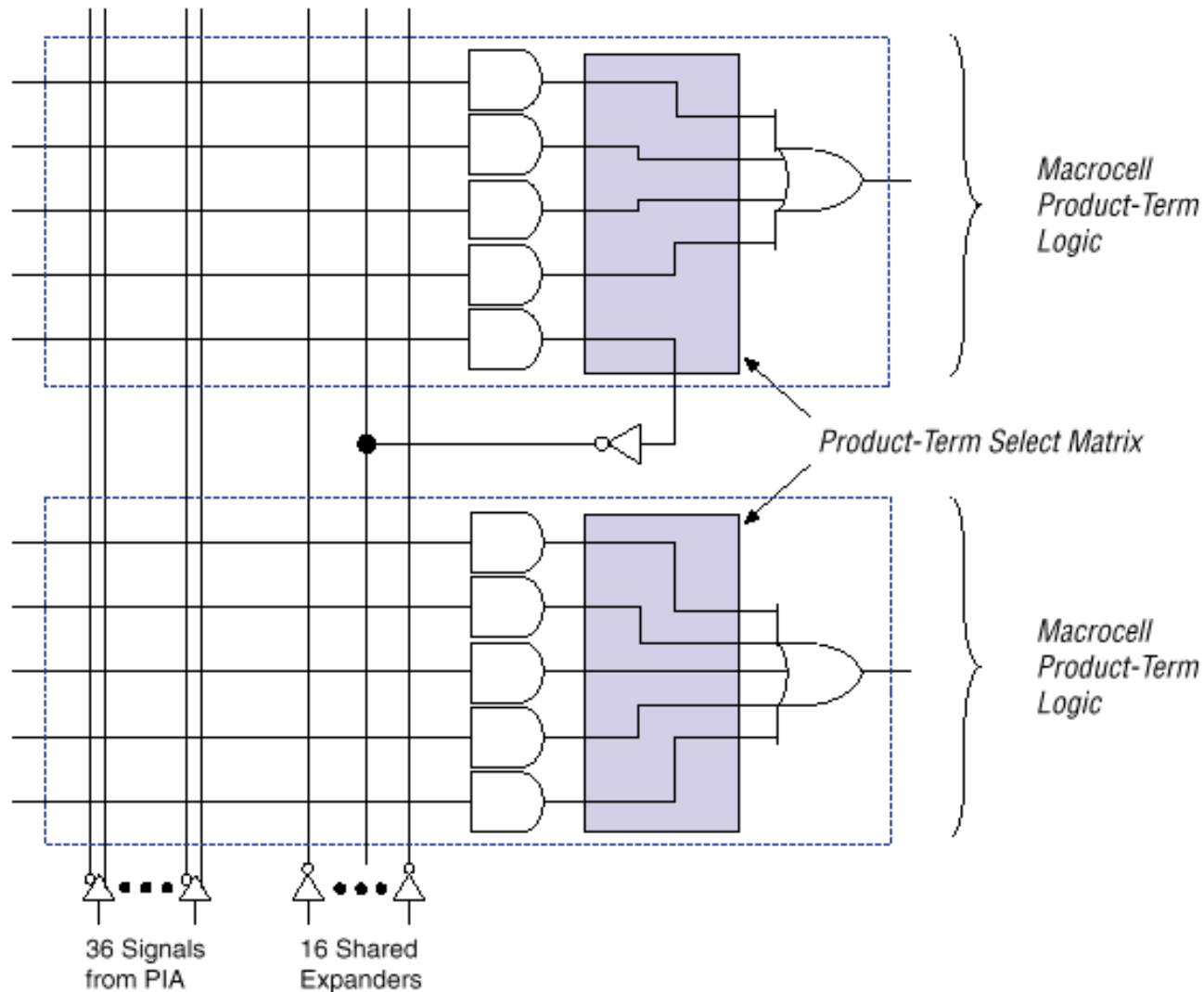


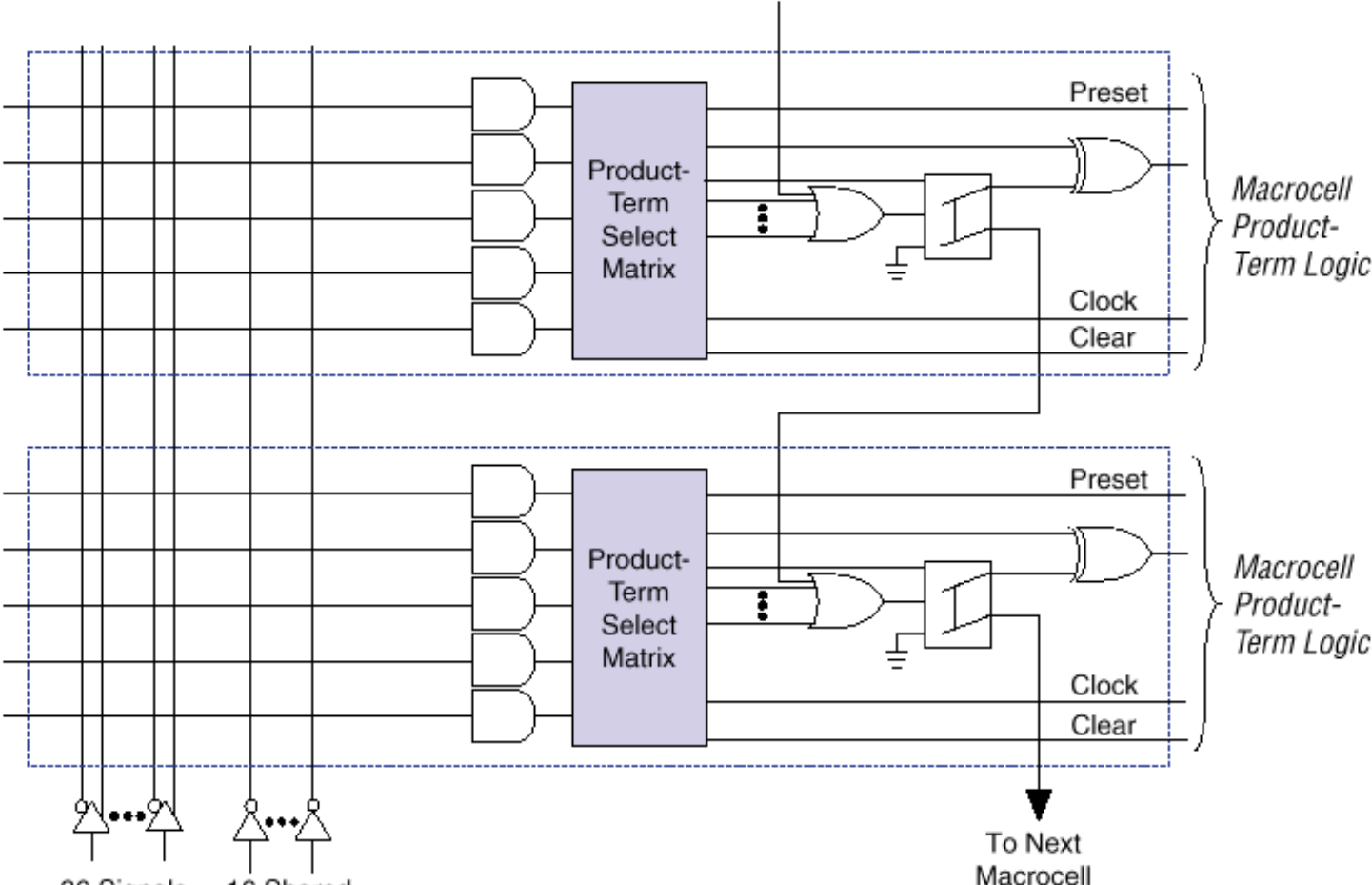
Diagrama de bloques de las MAX 7000E y MAX 7000S

# Familia MAX7000 (ALTERA)

*Shareable expanders can be shared by any or all macrocells in an LAB.*



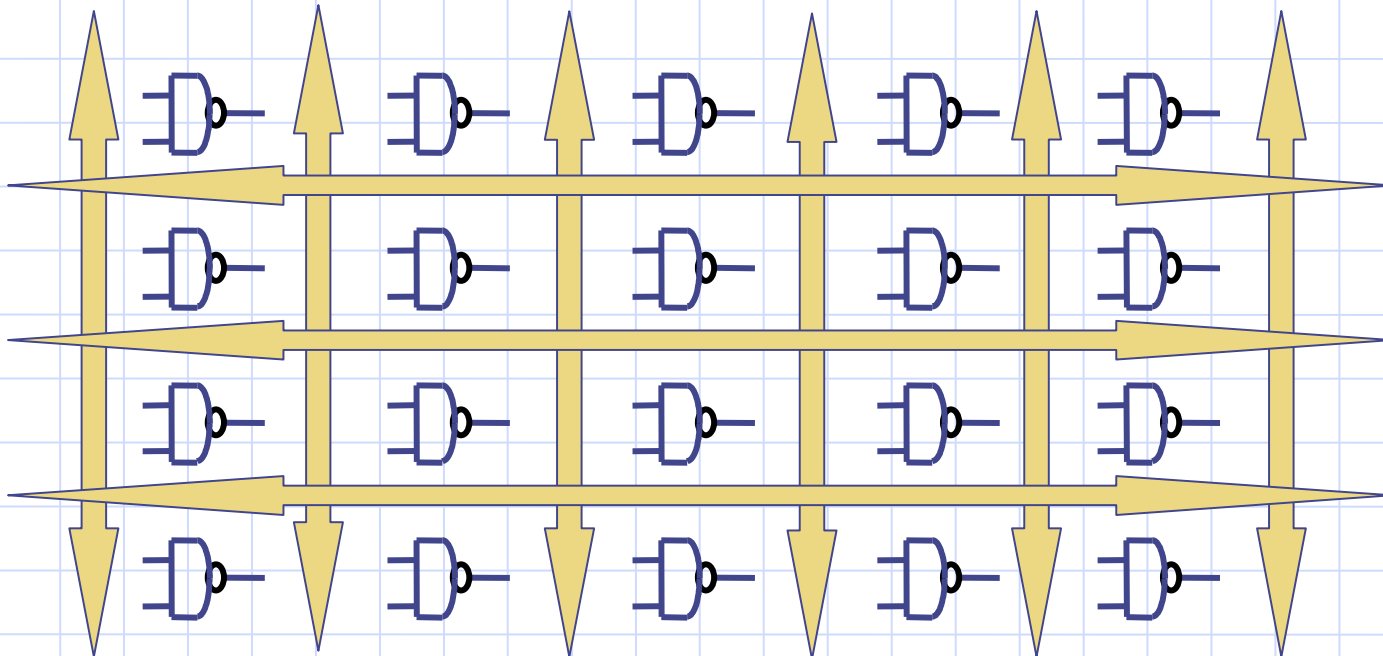
From  
Previous  
Macrocell



## Expansores paralelos

# Y otras soluciones?

- ◆ Para construir una PLD “mejorada” se distribuyen bloques lógicos más pequeños y se dispone de más recursos de interconexión menores y más repartidos.



# ¿Qué es una FPGA?

- ◆ Las FPGA (Field Programmable Gate Array) son *matrices de celdas lógicas* comunicadas entre si y/o con *bloques de E/S* mediante “*canales de comunicación*”.
- ◆ Las conexiones entre los diferentes elementos son programables.  
Tecnologías:
  - *SRAM* (Altera, Lucent Technol., Atmel, Xilinx)
  - *ANTIFUSIBLE* (Actel, Cypress, QuickLogic, Xilinx)
  - EPROM
- ◆ Bloques lógicos son LUT o MUX.

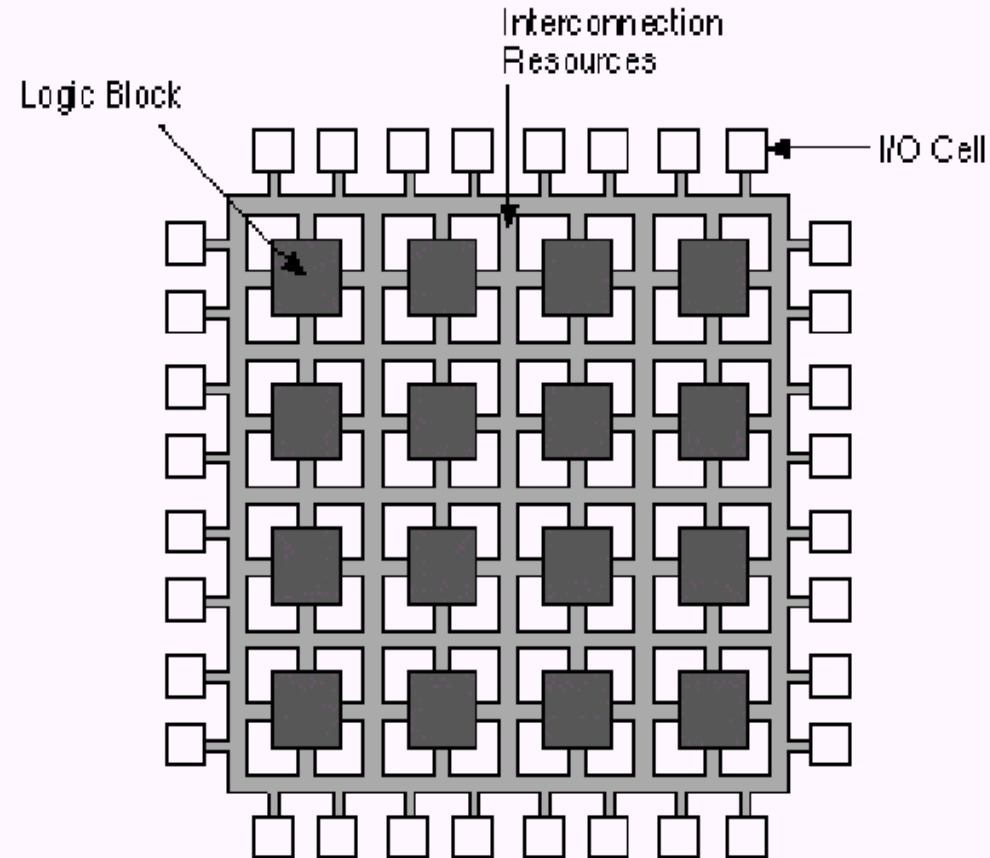


Figure 8 FPGA Architecture

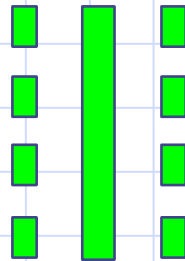


# CPLDs y FPGAs

## CPLD

**Complex Programmable Logic Devices**

(Circuitos Lógicos Prog. Complejos  
aplicación)



**Arquitectura:**

**tipo-PAL**

**Más Combinacional**

**Densidad:**

**Baja-a-media**  
**Muchas 22V10s**

**Prestaciones: Retardos Predecibles**

**Hasta 200 MHz**

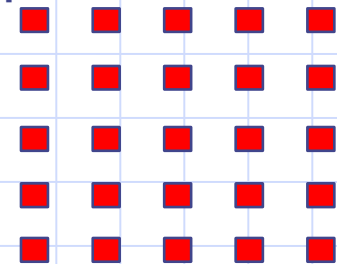
**Interconexión:**

**“Crossbar”**

## FPGA

**Field-Programmable Gate Array**

Circuitos Progr. por el usuario en la



**Tipo Matrices de puertas (GAL)**

**Más Registros**

**Medio-a-alta**

**1K-250K puertas lógicas**

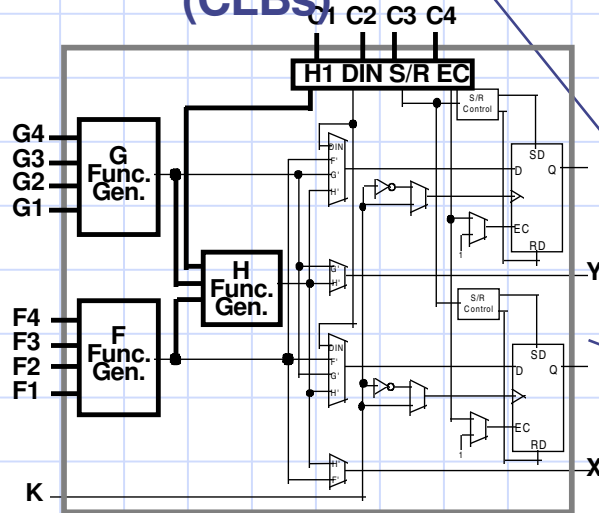
**Dependiente de aplicación**

**hasta 100MHz**

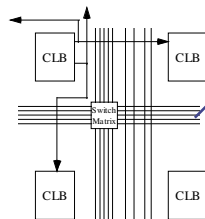
**Incremental**

# Arquitectura XC4000/Spartan

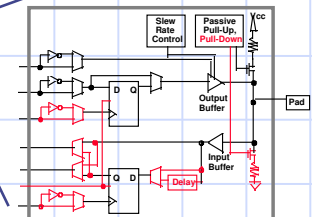
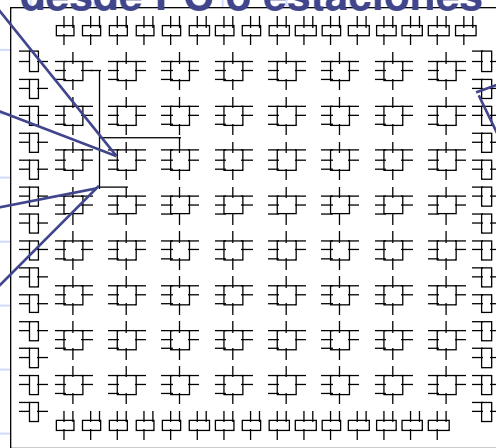
## Bloques Lógicos Configurables (CLBs)



## Interconexión Programable



- Alta Densidad -> 250K Puertas Lógicas
- LUT tipo SRAM para RAM Sincrona de Doble Puerto o Lógica
- Estructura tipo matriz de ASIC
- Salidas Tri-estados
- Reconfiguraciones infinitas, cargadas desde PC o estaciones de trabajo en ~1 s



## Bloques E/S (IOBs)

# Configurable Logic Blocks

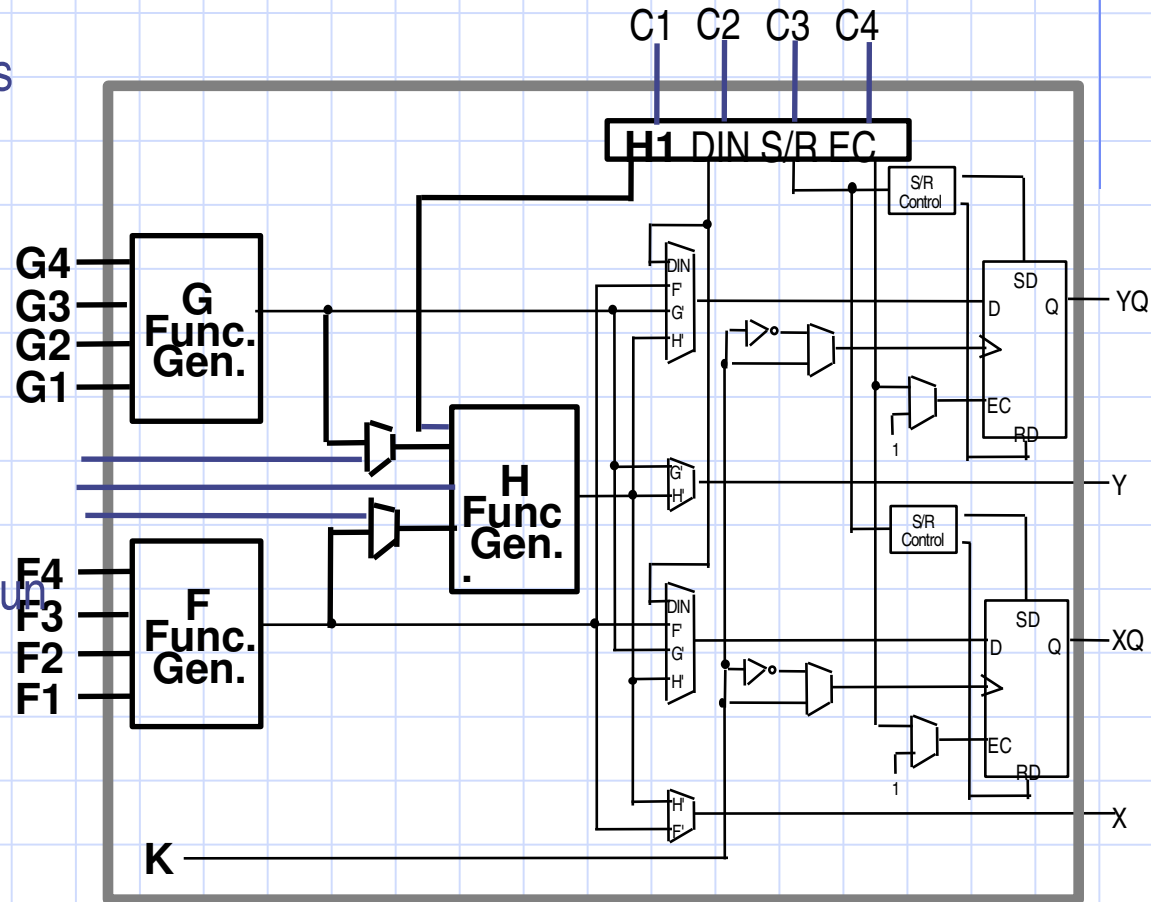
- ◆ 2 generadores de funciones de 4 entradas (Look Up Tables)

- 16x1 RAM o funciones lógicas.

- ◆ 2 Biestables

- Cada uno puede ser configurado independientemente como un biestable o como latch.

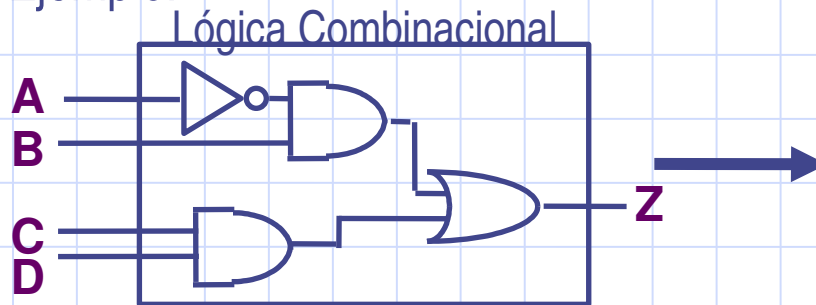
- Independiente polaridad de reloj.
- Set/Reset síncronos y asíncronos



# Look Up Tables

- ◆ Lógica Combinacional almacenada en una SRAM de 16x1 Look Up Tables (LUTs) en una CLB

- ◆ Ejemplo:

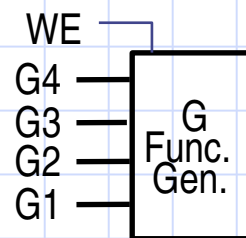


Look Up Table  
4-bit dirección

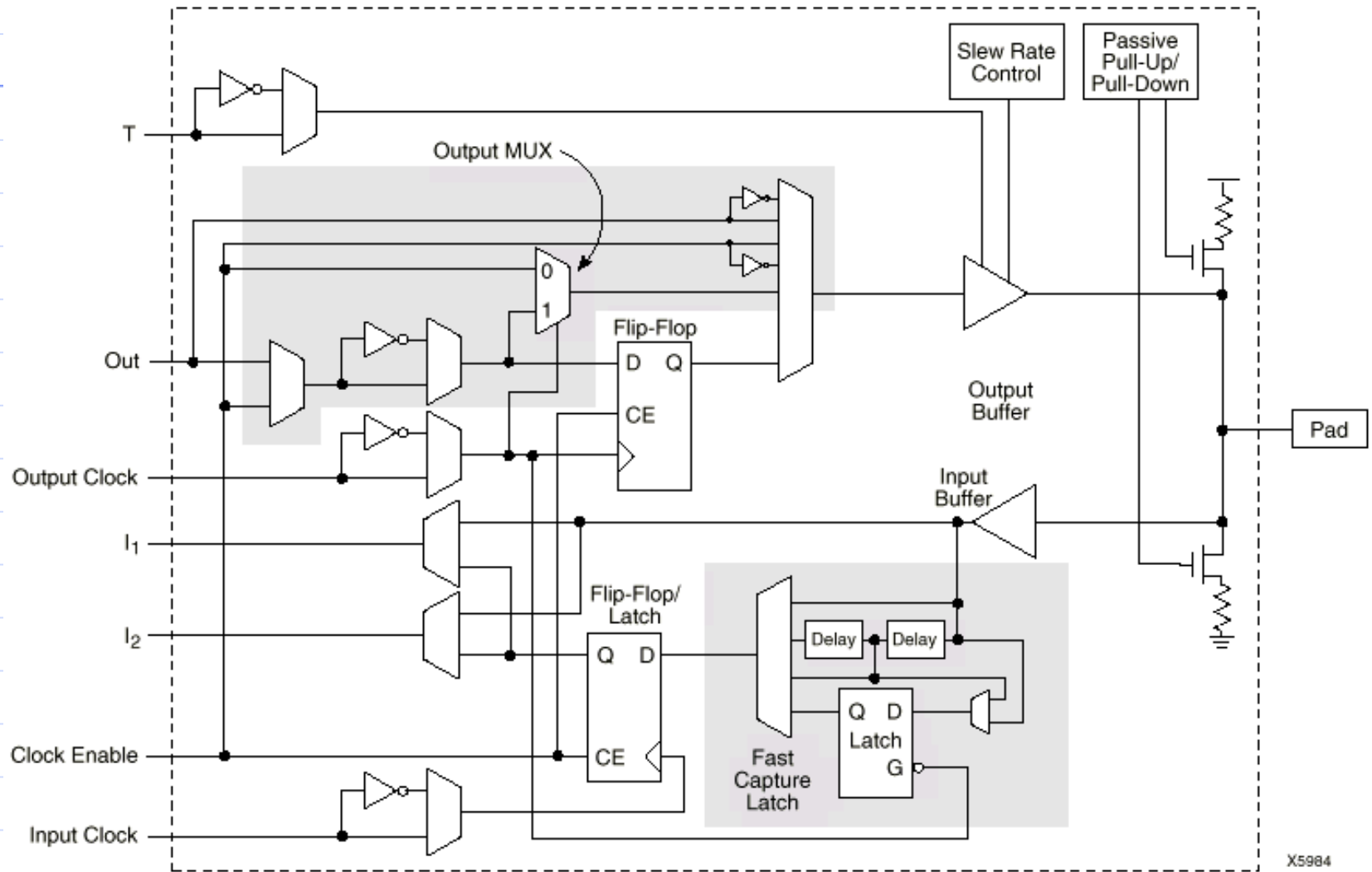
A	B	C	D	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

$$2^{(2^4)} = 64K !$$

- ◆ Elección para usar cada LUT como una generador de función de 4 entradas o como una RAM síncrona de doble puerto.



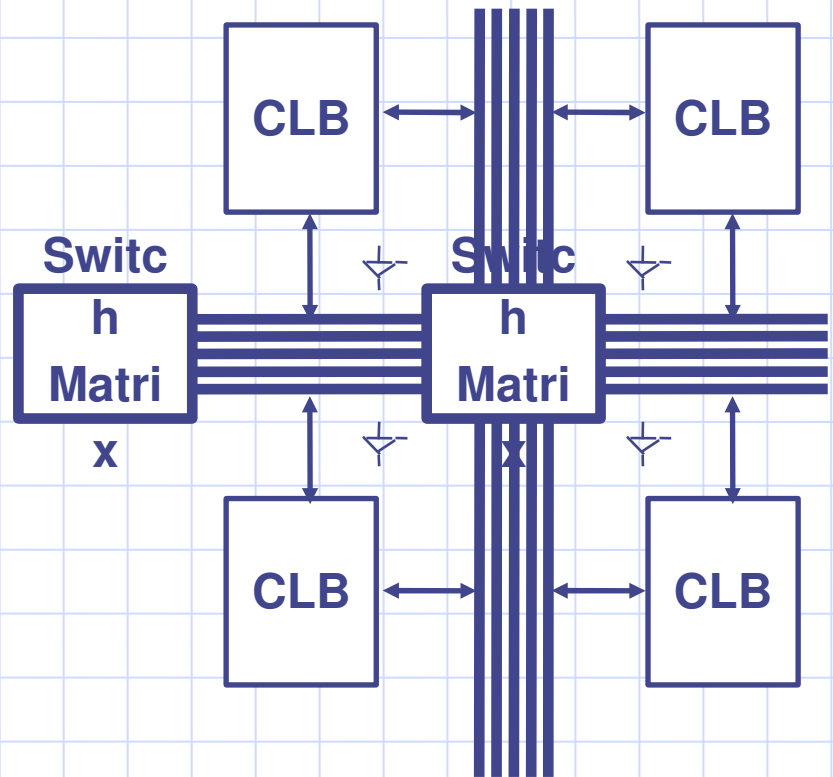
# Diagram I/O Block(IOBs)



Shaded areas are not included in XC4000E family.

# Rutado de Xilinx FPGA

- ◆ 1) Conexión directa y rápida - CLB to CLB
- ◆ 2) Conexión de propósito general –Uso matriz de conmutación (switch matrix)
- ◆ 3) Líneas largas.
  - Cruzan segmentos chip.
  - Relojes Globales, el menor desfase (skew)
  - 2 Tri-states por CLB para buses



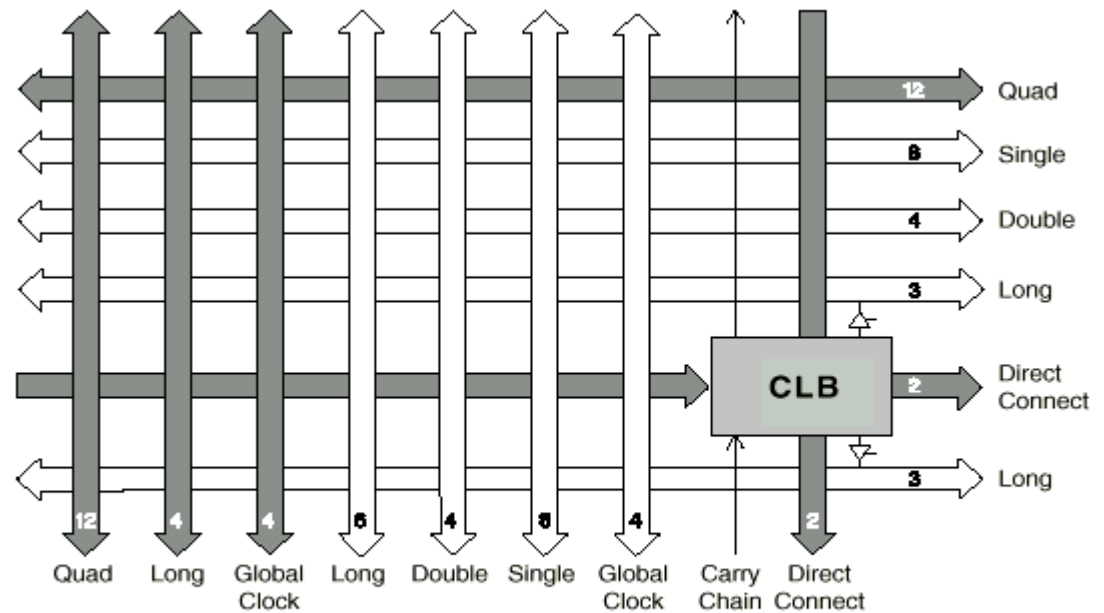
# Jerarquía de interconexión

## Tipos de interconexiones

- líneas simples
- líneas dobles
- líneas cuádruples
- líneas octales
- líneas largas

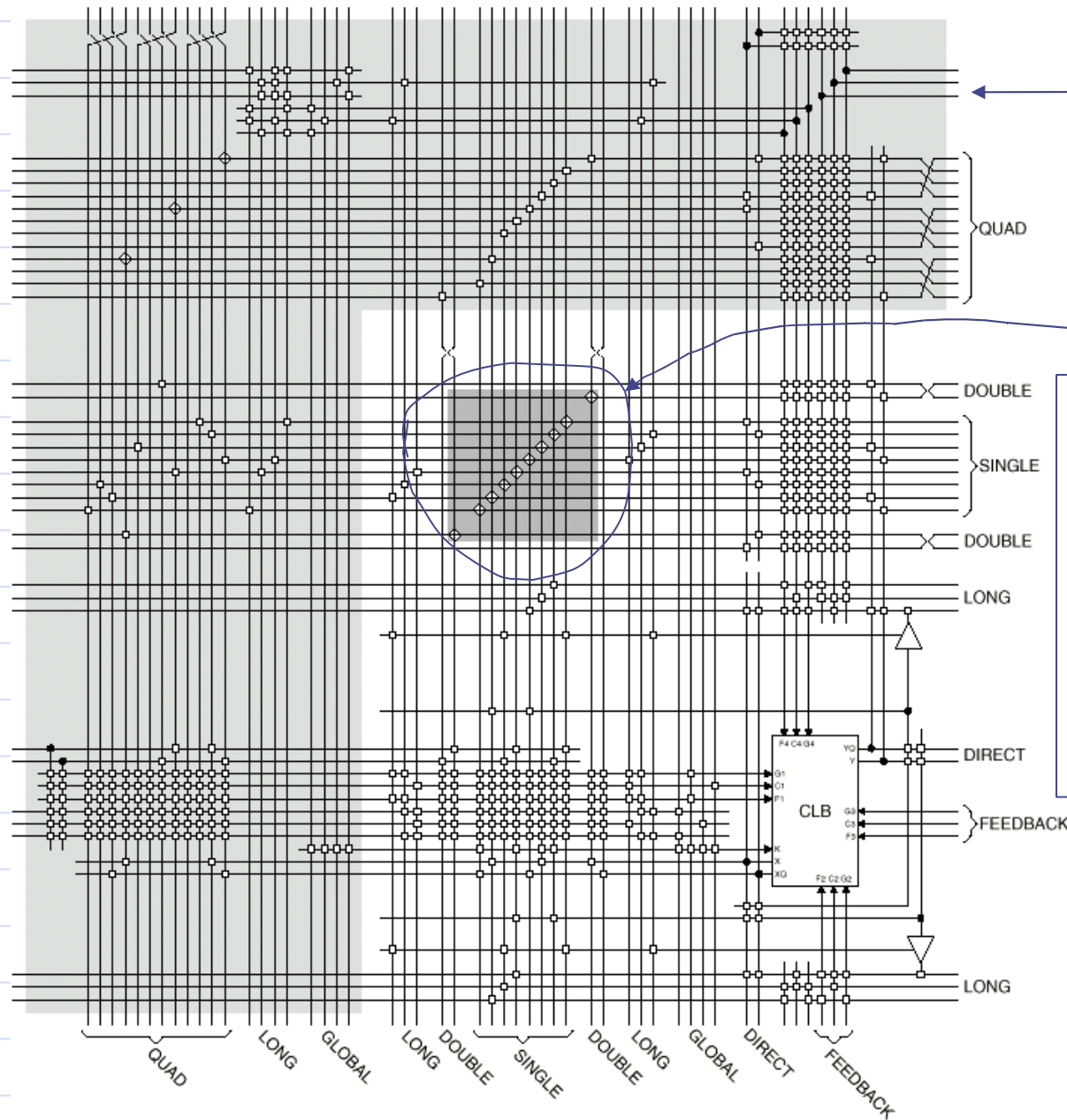
Table 15: Routing per CLB in XC4000 Series Devices

	XC4000E		XC4000X	
	Vertical	Horizontal	Vertical	Horizontal
Singles	8	8	8	8
Doubles	4	4	4	4
Quads	0	0	12	12
Longlines	6	6	10	6
Direct Connects	0	0	2	2
Globals	4	0	8	0
Carry Logic	2	0	1	0
Total	24	18	45	32



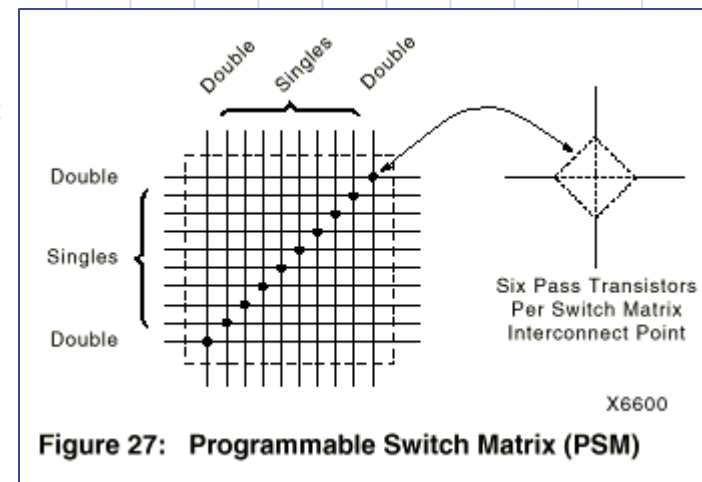
x5994  
High-Level Routing Diagram of XC4000 Series CLB (shaded arrows indicate XC4000X only)

# Interconexión programable



Sólo XC4000X

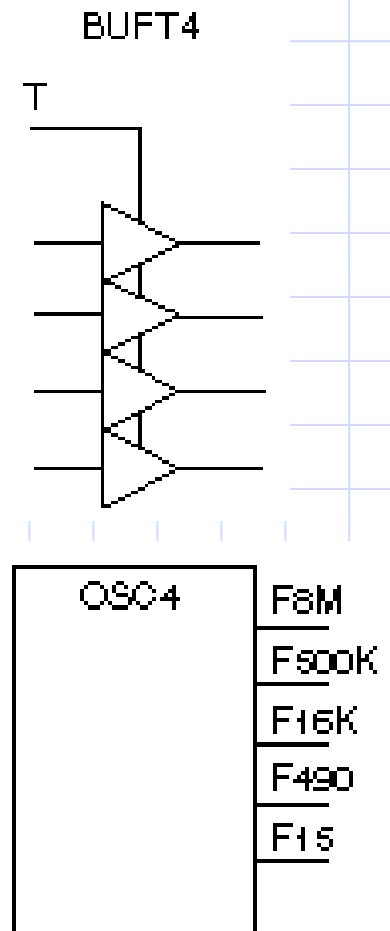
Matriz de conmutación





# Otros recursos de las FPGA

- ◆ Buffers Tri-state for buses (BUFT's)
- ◆ Buffers de reloj global & alta velocidad (BUFG's)
- ◆ Decodificadores extensos(DECODEx)
- ◆ Osciladores internos (OSC4)
- ◆ Reset global para todos los Flip-Flops, Latches (STARTUP)
- ◆ Recursos especial CLB especial
  - Lógica de Acarreo rápido
  - RAM de doble puerto síncrono.
  - Boundary Scan



# What's Really In that Chip?

Programmable Interconnect Points, PIPs (White)

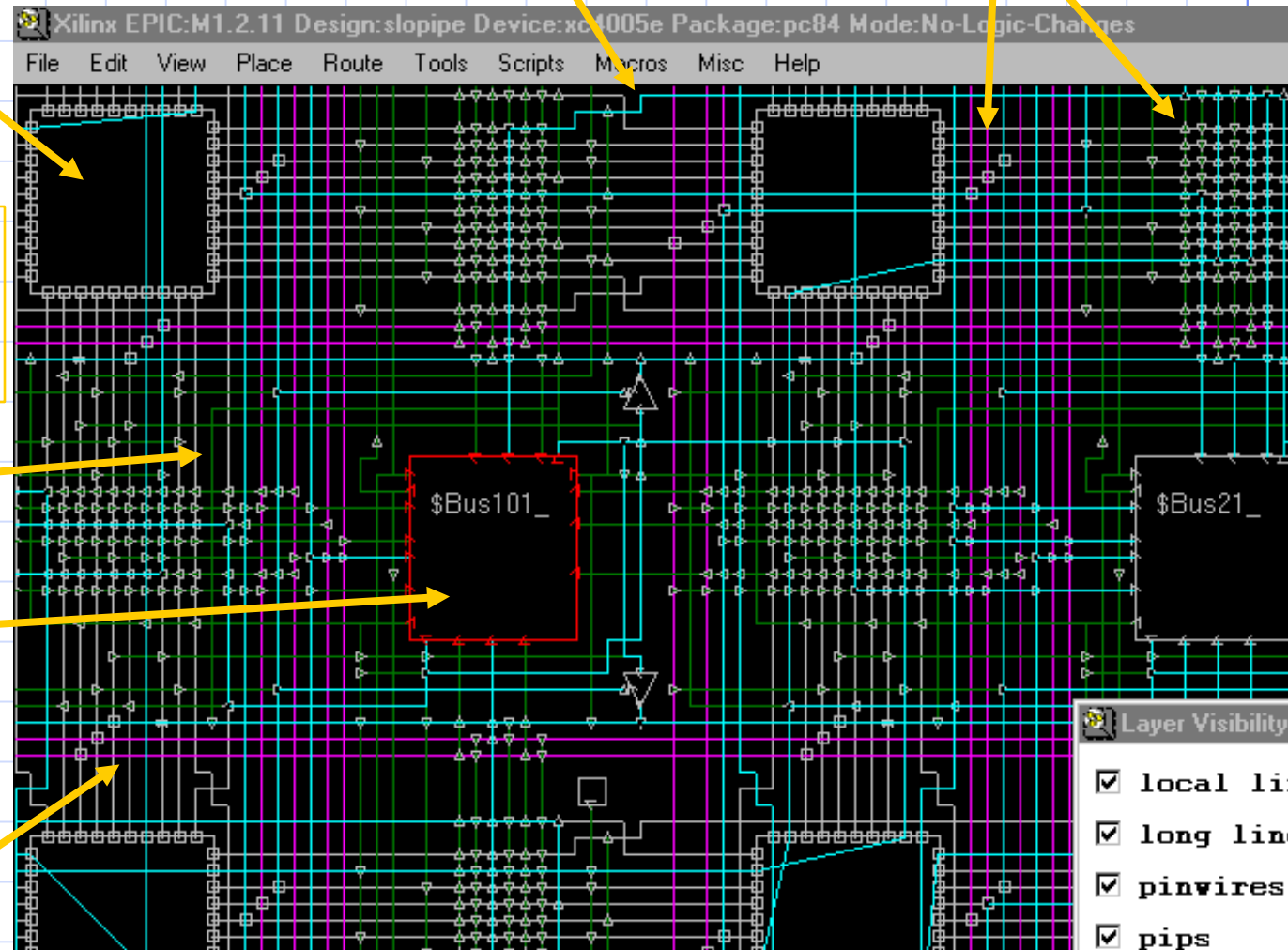
Switch  
Matrix

Routed Wires (Blue)

Direct  
Interconnect  
(Green)

CLB  
(Red)

Long Lines  
(Purple)



# Características adicionales FPGA avanzadas

- ◆ Memorias de gran tamaño empotrada
- ◆ Lógica dedicada para funciones aritméticas.
- ◆ Phase locked loops para sincronización de reloj.
- ◆ Circuitos de división y multiplicación.
- ◆ Reconfiguración parcial.

# CPLD o FPGA?

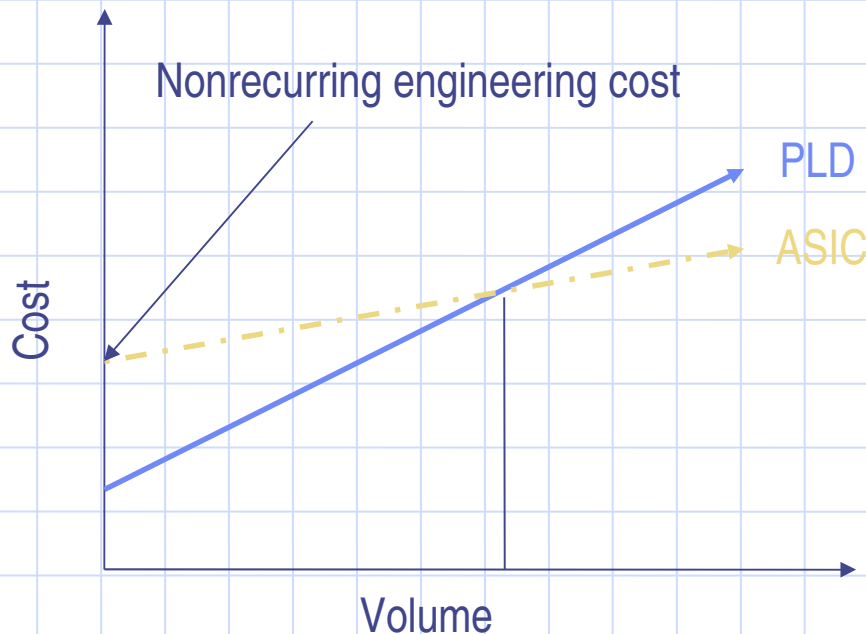
## ◆ CPLD

- ◆ No volatil.
- ◆ Verificación JTAG
- ◆ Alto fan-in
- ◆ Rápido contadores, maquinas de estado.
- ◆ Lógica Combinacional
- ◆ Lógica de Control

## ◆ FPGA

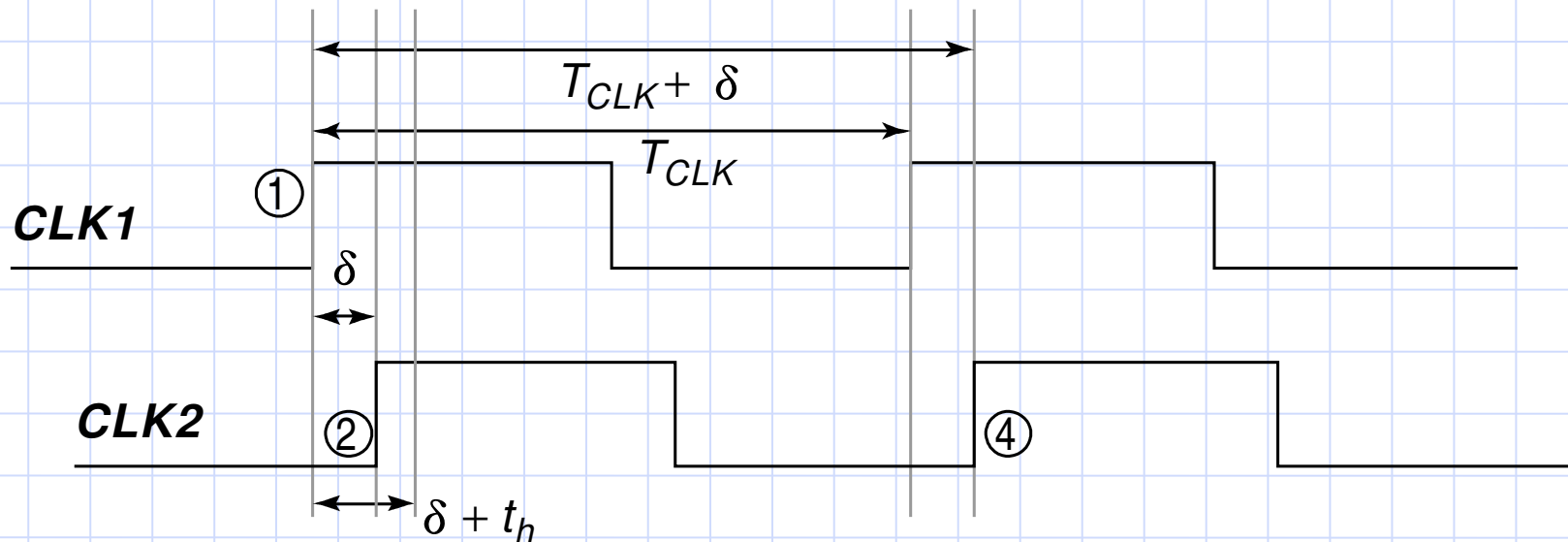
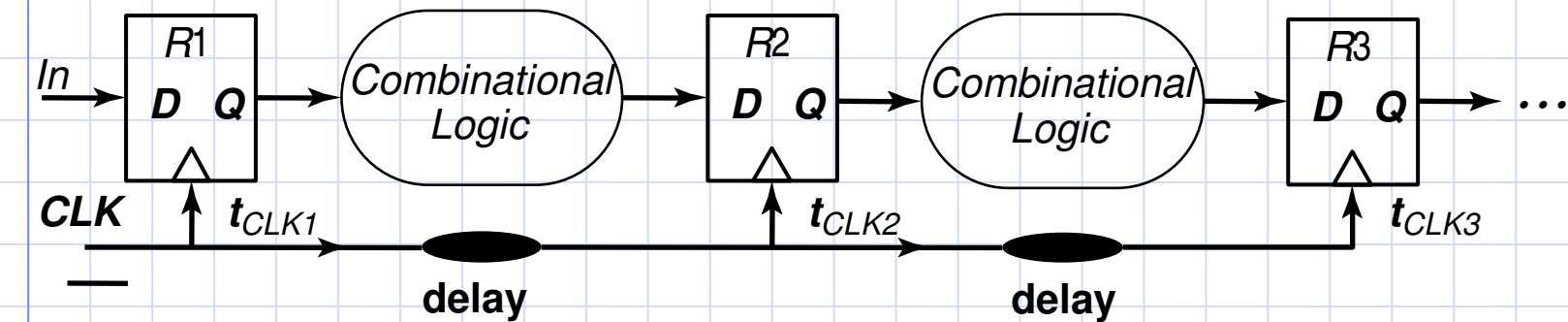
- ◆ Reconfiguración con SRAM
- ◆ Excelente para arquitectura de computadores, DSP,
- ◆ Flujo de diseño similar ASIC
- ◆ Necesita PROM para operaciones no volatiles

# Ventajas PLD

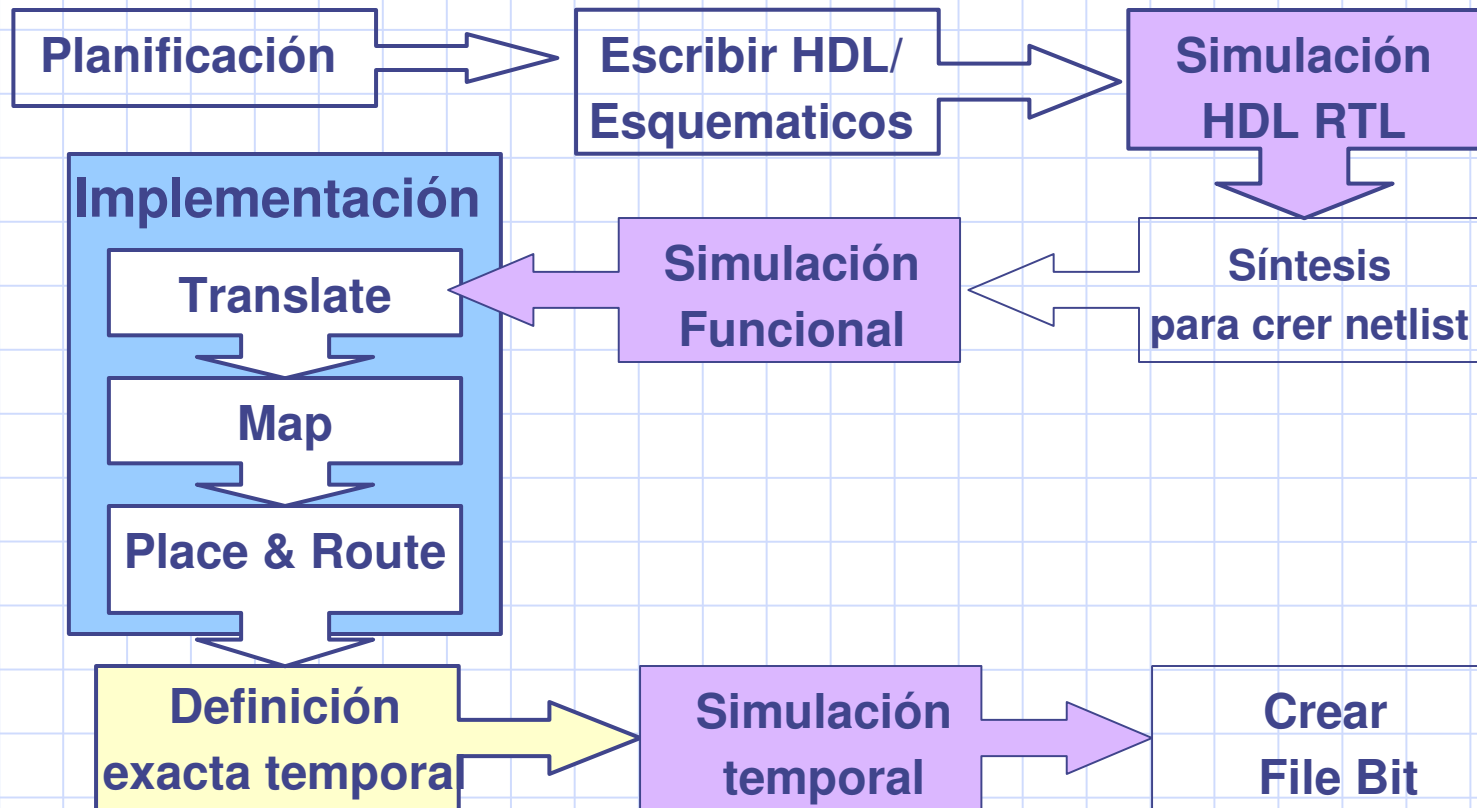


- ◆ Corto tiempo de diseño
- ◆ Mucho más barato para un volumen bajo de producción.

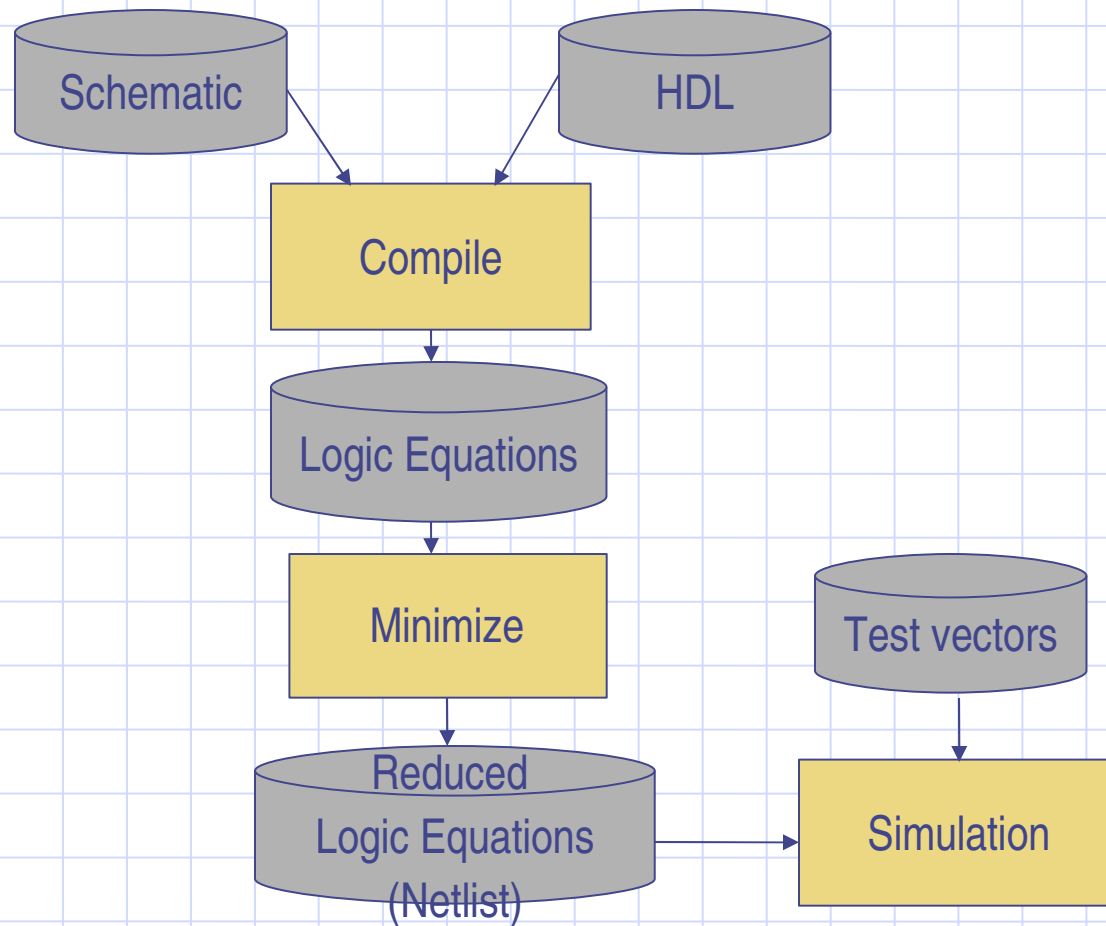
# Skew



# Flujo de Diseño



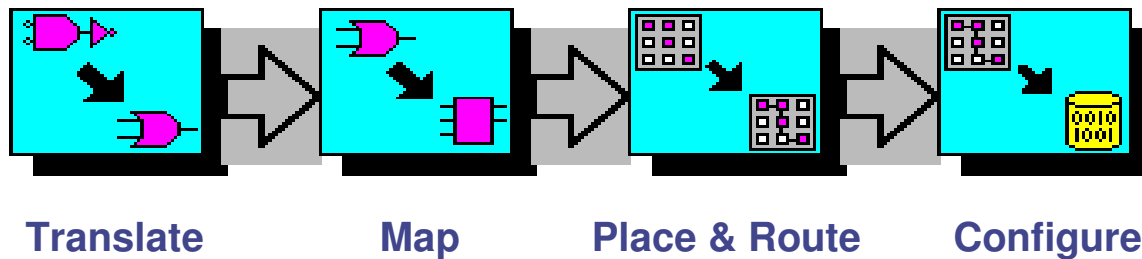
# Flujo diseño CPLD





# Implementación del Diseño

- ◆ Entrada: **Netlist** Salida: **bitstream**
- ◆ **Map**ear el diseño con los recursos de la FPGA
  - Descomponer el circuito en bloques que tenga máximo  $n$  entradas.
  - Problema NP-hard
  - Sin embargo, una solución óptima no se necesita.



# Implementación del Diseño (Cont.)

- ◆ **Place:** Colocar los bloques lógicos generados durante el mapeo a posiciones específicas dentro de la FPGA.
  - Objetivo: minimizar la longitud de los cables.
  - NP-hard
- ◆ **Route:** Establecer una ruta de interconexiones entre los bloques lógicos.
  - NP-hard

# Aplicaciones PLD

- ◆ Lógica de pegamento (Glue Logic )
  - remplazar SSI con SPLD y MSI CPLD)
- ◆ Rápido desarrollo. (Todos)
- ◆ Prototipo de diseño (FPGAS/CPLD)
- ◆ Emulación
- ◆ Computación de propósito específico.
- ◆ Reconfiguración dinámica (Codiseño)
- ◆ SoC (System on Chip)

# Diseño de Interfaces de alta velocidad

◆ Redimiento de E/S de 840 Mbps

◆ Soporta la siguientes interfaces.

- PCI-X 133 MHz
- RapidI/O™
- POS PHY Level 4 (16 bits/clock @ 832 Mbps), SPI-4
- Lightning Data Transport (LDT)

# Up to 1108 User I/Os

## ◆ Double data rate

- Input, output and 3-state control registers

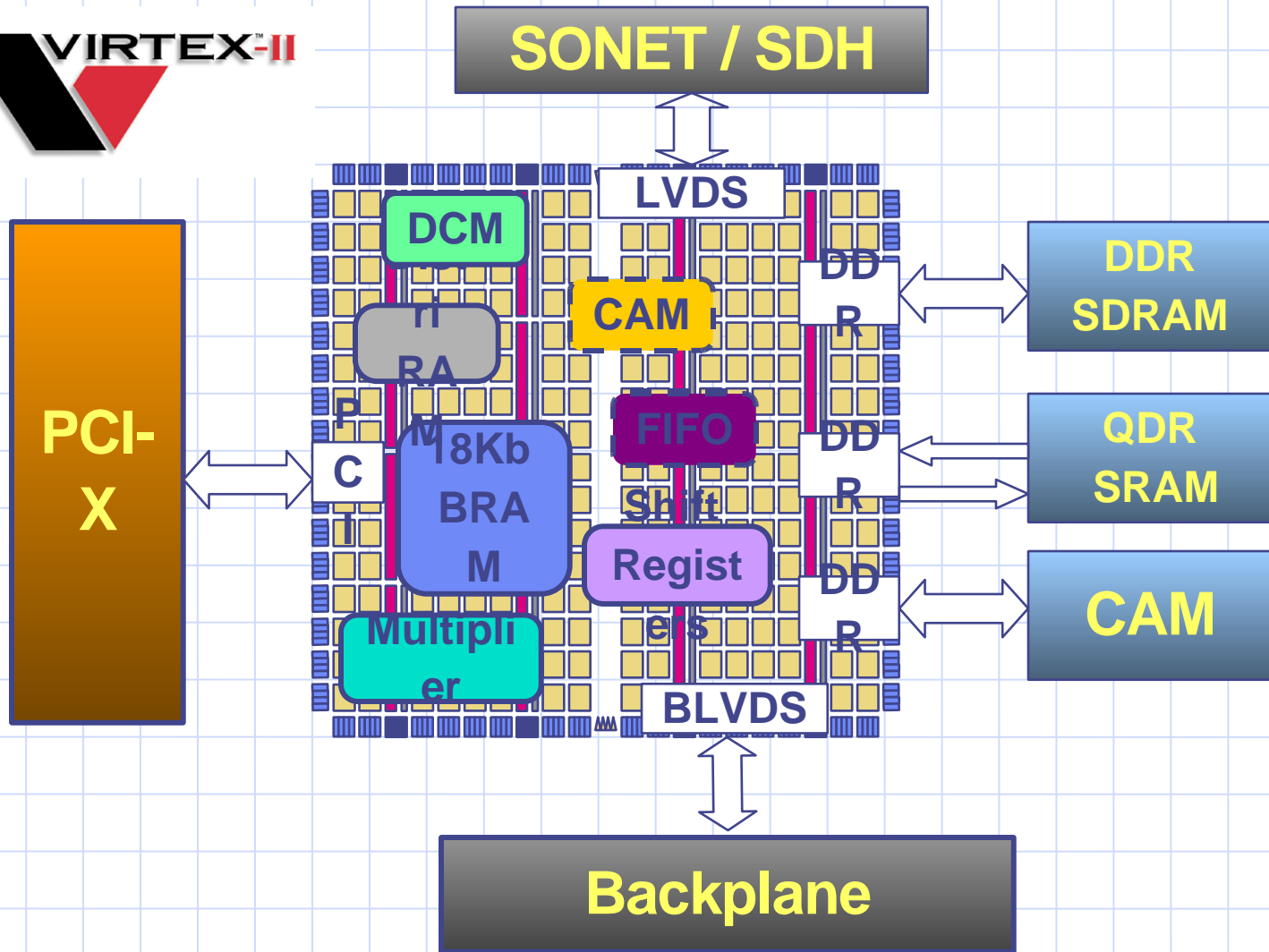
## ◆ Up to 554 differential I/O pairs:

- High-speed LVDS, Bus LVDS, and LVPECL on all I/O pairs

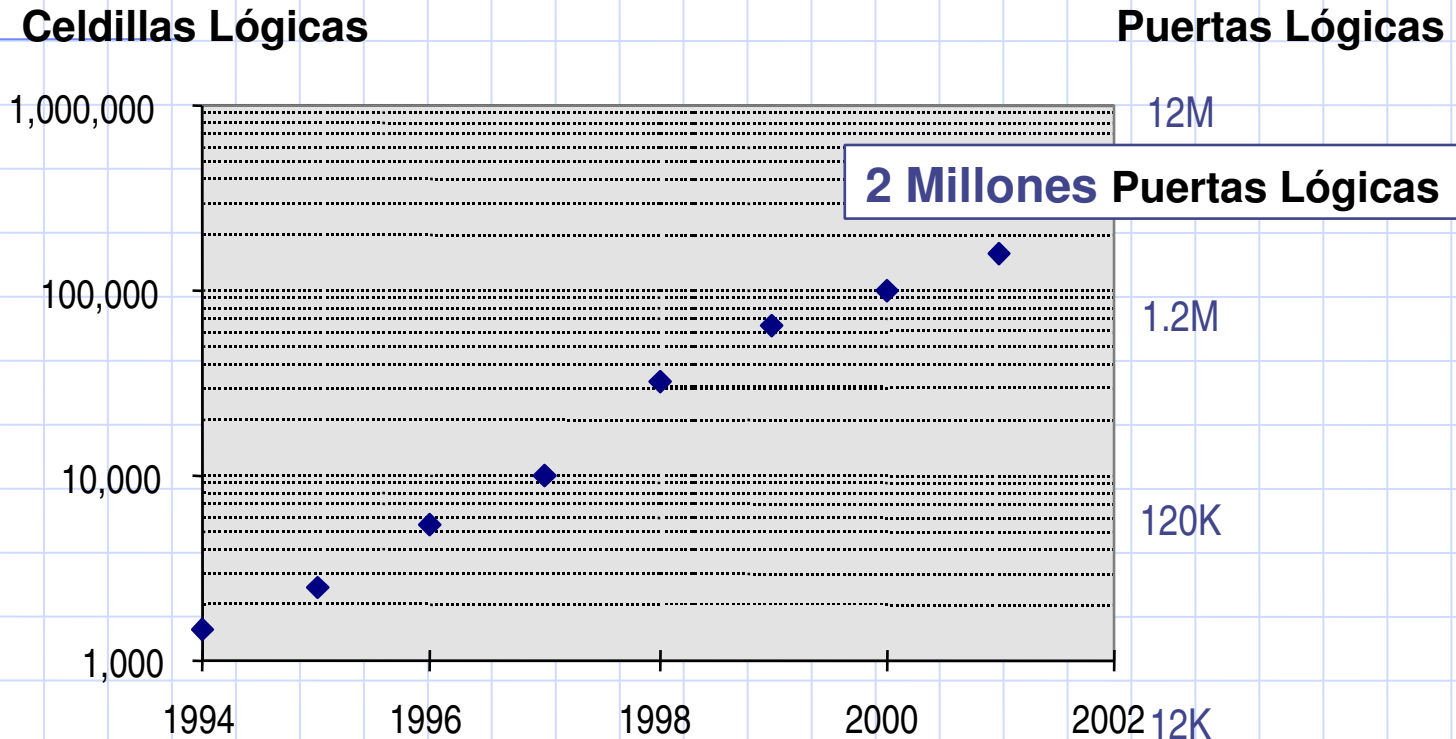
## ◆ Single-ended I/O: 19 standards supported

- PCI @ 33 MHz & 66 MHz compliant
- PCI-X @ 133 MHz compliant
- LVTTTL, LVCMOS, SSTL, HSTL, GTL, AGP

# Virtex-II Architecture

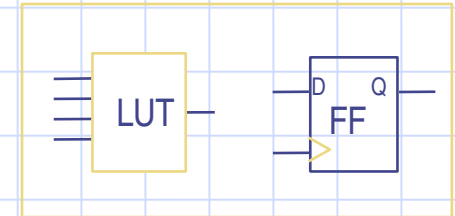


# Crecimiento Exponencial en Densidad



◆ **175,000 celdas Lógicas = 2.0 millones**  
**puertas Lógicas en 2001**

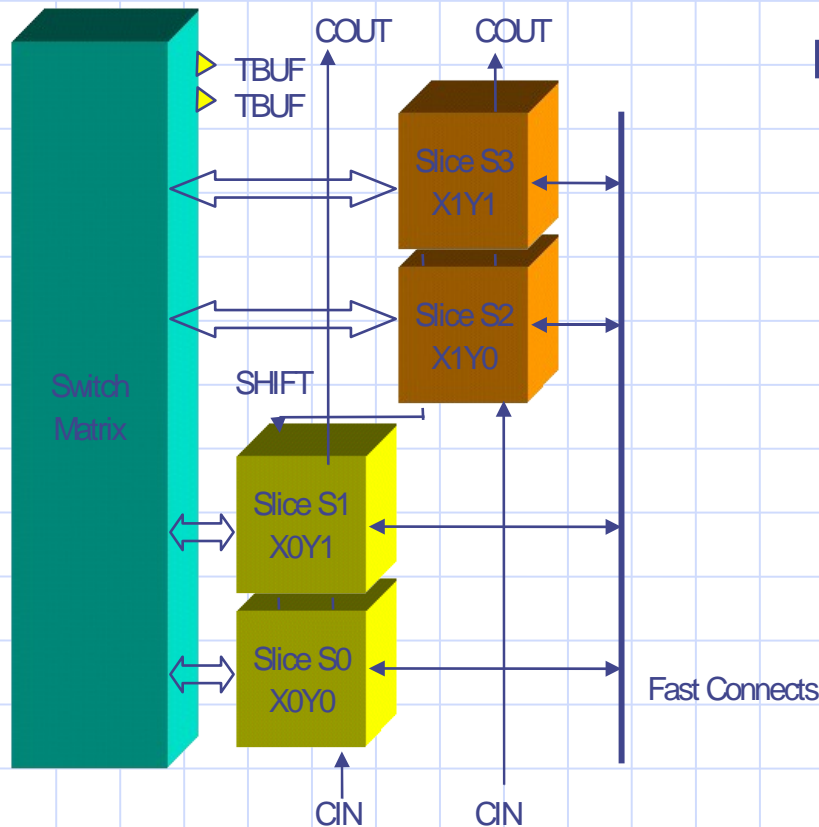
◆ **1 celdilla lógica = LUT 4-E + FF**



# CLB Contains Four Slices

◆ Each CLB is connected to one switch matrix

■ Providing access to general routing resources



High level of logic integration

- ◆ Wide-input functions:
  - 16:1 multiplexer in 1 CLB or any function
  - 32:1 multiplexer in 2 CLBs (1 level of LUT)
- ◆ Fast arithmetic functions
  - 2 look-ahead carry chains per CLB column
- ◆ Addressable shift registers in LUT
  - 16-b shift register in 1 LUT
  - 128-b shift register in 1 CLB (dedicated shift chain)



# SelectI/O-Ultra™ Technology

## ◆ High Bandwidth and XCITE™ on-chip termination

- Support 19 single-ended standards and 6 differential standards

✓	Digitally controlled impedance (XCITE)
✓	Up to 840 Mbps per I/O pair (LVDS)
✓	Built-in DDR registers
✓	LVDS current source drivers
✓	LDT & ULVDS differential signaling
✓	HSTL-II standard support
✓	Up to 1,108 user I/Os
✓	
✓	

# Processes and Tools

## Step 1: Design

## Step 2: Synthesize to create netlist

## Step 3: Implement design

- ◆ Some tools are listed multiple times with different task names

## Step 4: Configuration

