# Arquitectura de Computadoras

(Cód. 5561)
1° Cuatrimestre 2018

Dra. Dana K. Urribarri
DCIC - UNS

# Circuitos integrados

# Circuitos integrados

- Circuito intergrado (IC) o *chip*:

  Estructura de silicio (material semiconductor) que contiene los componentes electrónicos de compuertas digitales y elementos de almacenamiento.

  Se monta en un contenedor de plástico o cerámica.

# Niveles de integración

A medida que mejor la tecnología, se incrementa la cantidad de compuertas que puede haber en un chip.

*Small-scale integration (SSI)*

- *V*arias compuertas independientes en un chip. Menos de 10 compuertas.

*Medium-scale integration (MSI)*

- Realizan alguna función específica elemental (sumar). Entre 10 y 100 compuertas.

*Large-scale integration (LSI)*

- Incluye procesadores pequeños, memorias pequeñas y módulos programables. Entre 100 y 100.000 compuertas.

*Very-large-scale integration (VLSI)*

- Microprocesadores complejos o para procesamiento digital de señales (DSP). Más de 100.000 compuertas (cientos de millones).

# SSI: Bloques funcionales

- Decoder
- Encoder
- Multiplexor

# Decoder

- Un código de *n* bits puede representar $2^n$ elementos distintos.
- *Decodificador de n a m*

  Convierte una entrada *n* bits en una salida *m* bits: $n \leq m \leq 2^n$

  - Genera $2^n$ (o menos) minitérminos a partir de los *n* bits de entrada.
  - La salida $D_i$ es igual a 1 ($D_j = 0\ \forall j \neq i$) si la entrada codifica el valor *i.*

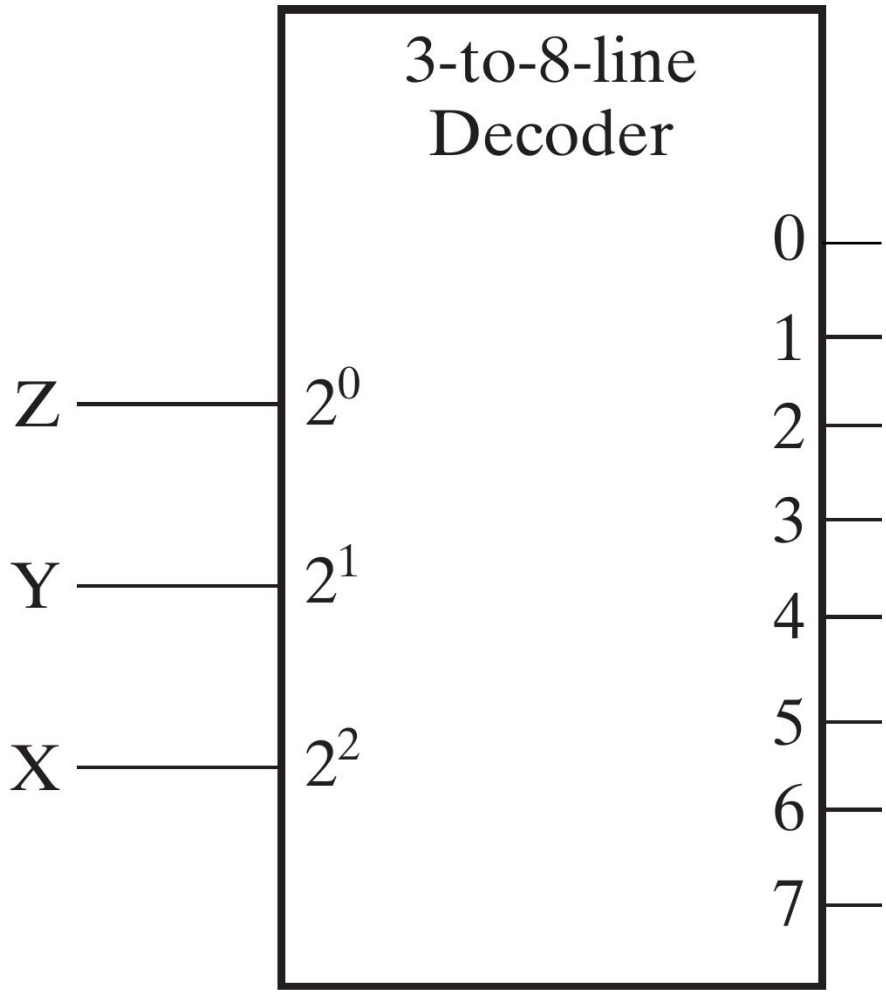
# Decoder

Decoder 1 a 2

| A | $D_0$ | $D_1$ |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 0 | 1 |

Decoder 2 a 4

| $A_1$ | $A_0$ | $D_0$ | $D_1$ | $D_2$ | $D_3$ |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |



# Encoder

- Realiza la función inversa del decoder.
- Tiene $2^n$ entradas (o menos) y *n* salidas.

Encoder 8 a 3

| Inputs | | | | | | | | Outputs | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | $A_2$ | $A_1$ | $A_0$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

# Encoder

- Ambigüedades:
  - Solo una entrada puede tener valor 1.
    ¿Qué pasa cuando hay más de una entrada en 1? La salida dependerá de la implementación.
  - ¿Qué pasa cuando todas las entradas valen 0?

- Se establecen prioridades.

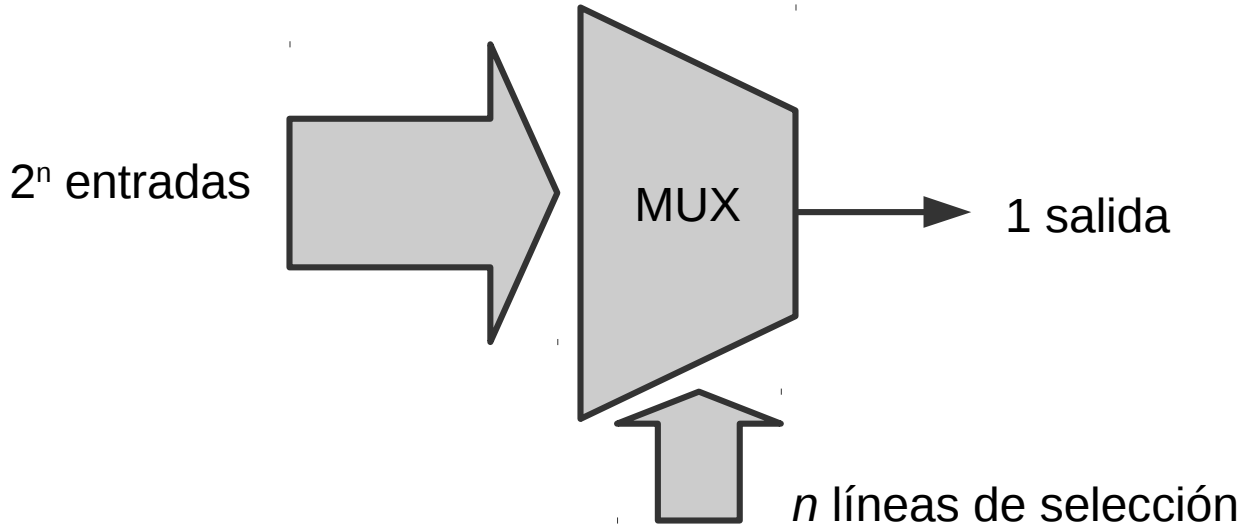
# Priority encoder

- Las entradas tienen diferentes prioridades.
- Agrega una salida que indica validez de la demás salidas.

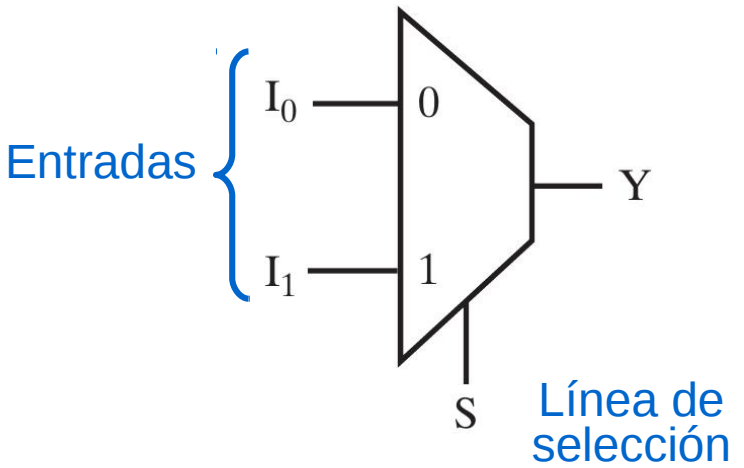
| Inputs | | | | Outputs | | |
|---|---|---|---|---|---|---|
| $D_3$ | $D_2$ | $D_1$ | $D_0$ | $A_1$ | $A_0$ | V |
| 0 | 0 | 0 | 0 | X | X | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | X | 0 | 1 | 1 |
| 0 | 1 | X | X | 1 | 0 | 1 |
| 1 | X | X | X | 1 | 1 | 1 |

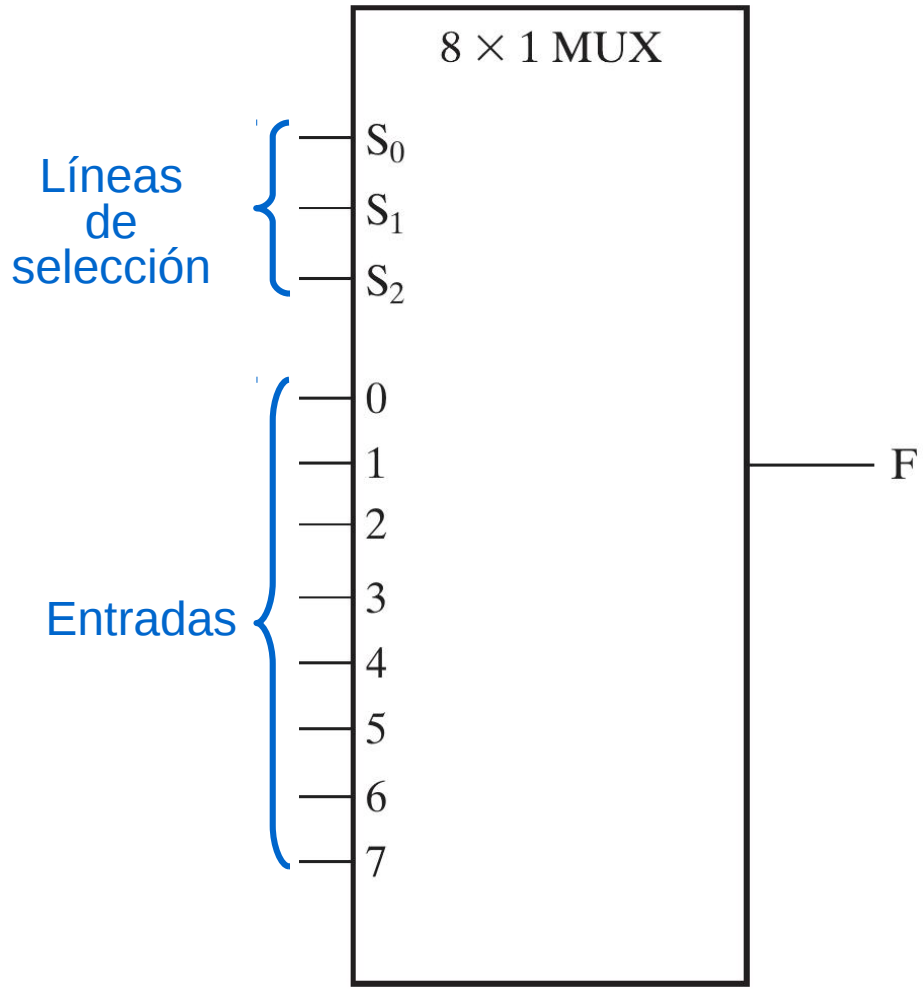
# Multiplexor

- Circuito combinacional que selecciona una de las entradas la mapea a la (única) salida.
- Generalmente, *n* líneas de selección que determinan cuál de las $2^n$ entradas será la salida.

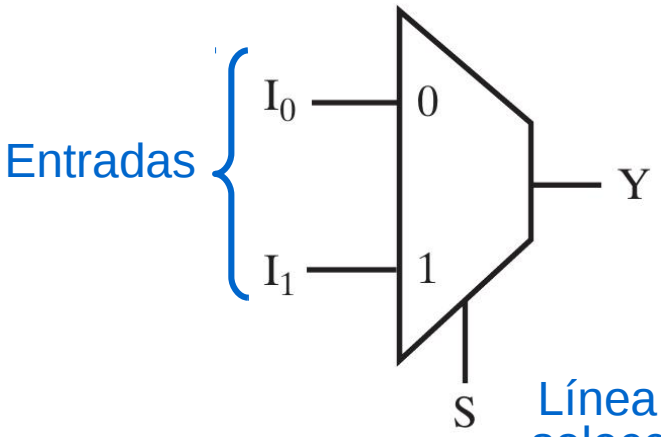


# Multiplexor



# Multiplexor




| S | Y |
|---|---|
| 0 | $I_0$ |
| 1 | $I_1$ |

| S | $I_0$ | $I_1$ | Y |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

# Multiplexor

| $S_1$ | $S_0$ | Y |
|---|---|---|
| 0 | 0 | $I_0$ |
| 0 | 1 | $I_1$ |
| 1 | 0 | $I_2$ |
| 1 | 1 | $I_3$ |

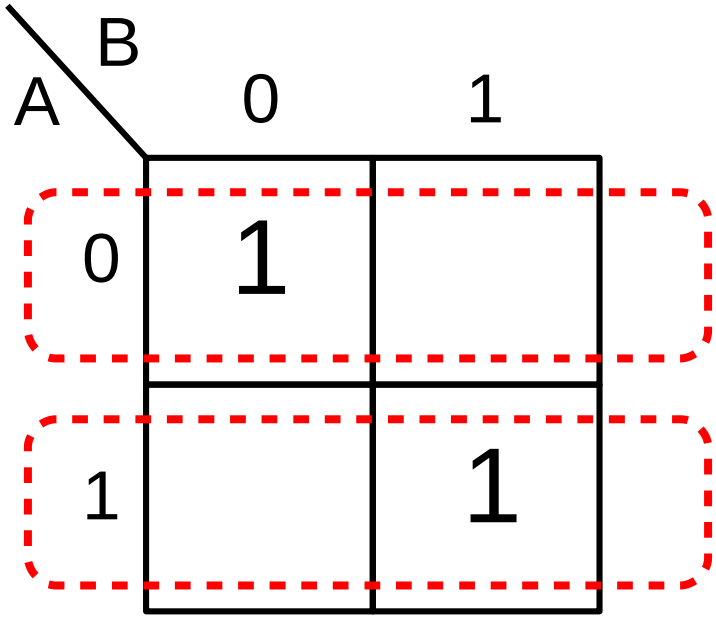
| $S_1$ | $S_2$ | $I_0$ | $I_1$ | $I_3$ | $I_4$ | F |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | **0** |
| 0 | 0 | 0 | 0 | 0 | 1 | **0** |
| 0 | 0 | 0 | 0 | 1 | 0 | **0** |
| 0 | 0 | 0 | 0 | 1 | 1 | **0** |
| 0 | 0 | 0 | 1 | 0 | 0 | **0** |
| 0 | 0 | 0 | 1 | 0 | 1 | **0** |
| 0 | 0 | 0 | 1 | 1 | 0 | **0** |
| 0 | 0 | 0 | 1 | 1 | 1 | **0** |
| 0 | 0 | 1 | 0 | 0 | 0 | **1** |
| 0 | 0 | 1 | 0 | 0 | 1 | **1** |
| 0 | 0 | 1 | 0 | 1 | 0 | **1** |
| 0 | 0 | 1 | 0 | 1 | 1 | **1** |
| 0 | 0 | 1 | 1 | 0 | 0 | **1** |
| 0 | 0 | 1 | 1 | 0 | 1 | **1** |
| 0 | 0 | 1 | 1 | 1 | 0 | **1** |
| 0 | 0 | 1 | 1 | 1 | 1 | **1** |

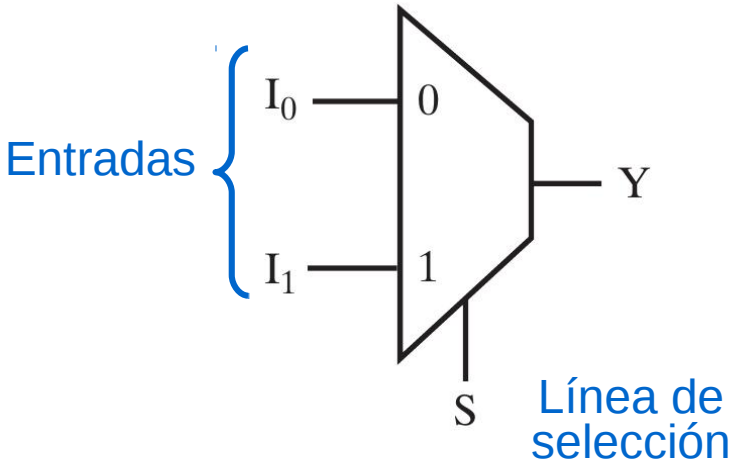
| $S_1$ | $S_2$ | $I_0$ | $I_1$ | $I_3$ | $I_4$ | F |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | * | * | * | **0** |
| 0 | 0 | 1 | * | * | * | **1** |
| 0 | 1 | * | 0 | * | * | **0** |
| 0 | 1 | * | 1 | * | * | **1** |
| 1 | 0 | * | * | 0 | * | **0** |
| 1 | 0 | * | * | 1 | * | **1** |
| 1 | 1 | * | * | * | 0 | **0** |
| 1 | 1 | * | * | * | 1 | **1** |

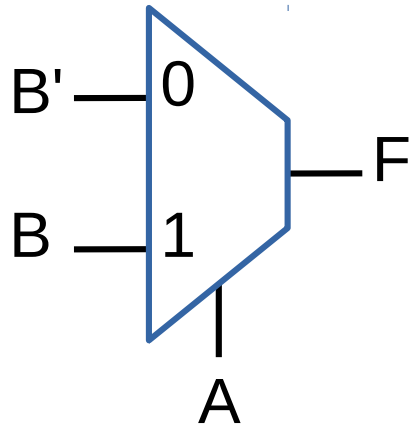
# Ejemplo

Implementar F usando un MUX 2 a 1

| A | B | F |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



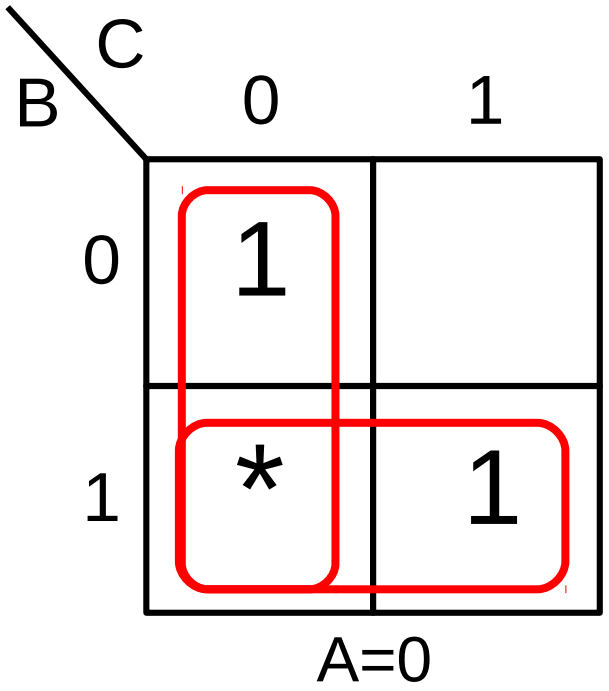



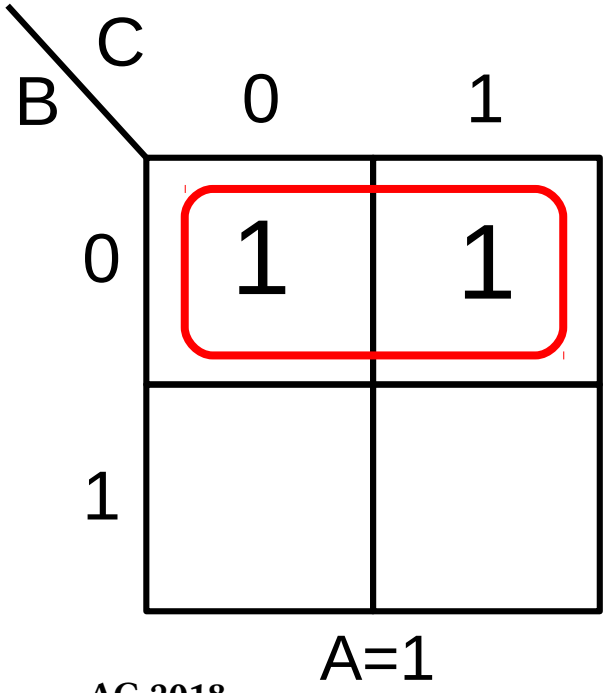



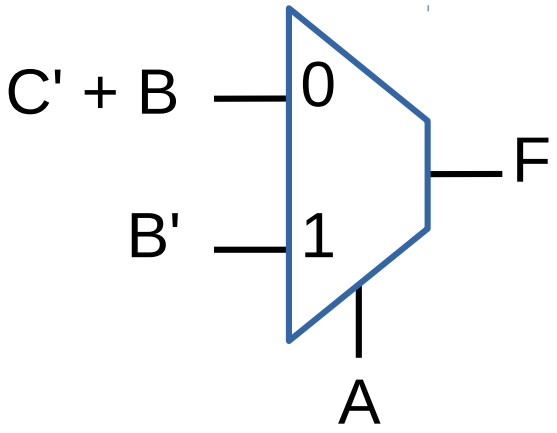
# Ejemplo

Implementar F con un MUX 2 a 1

| A | B | C | F |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | * |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |



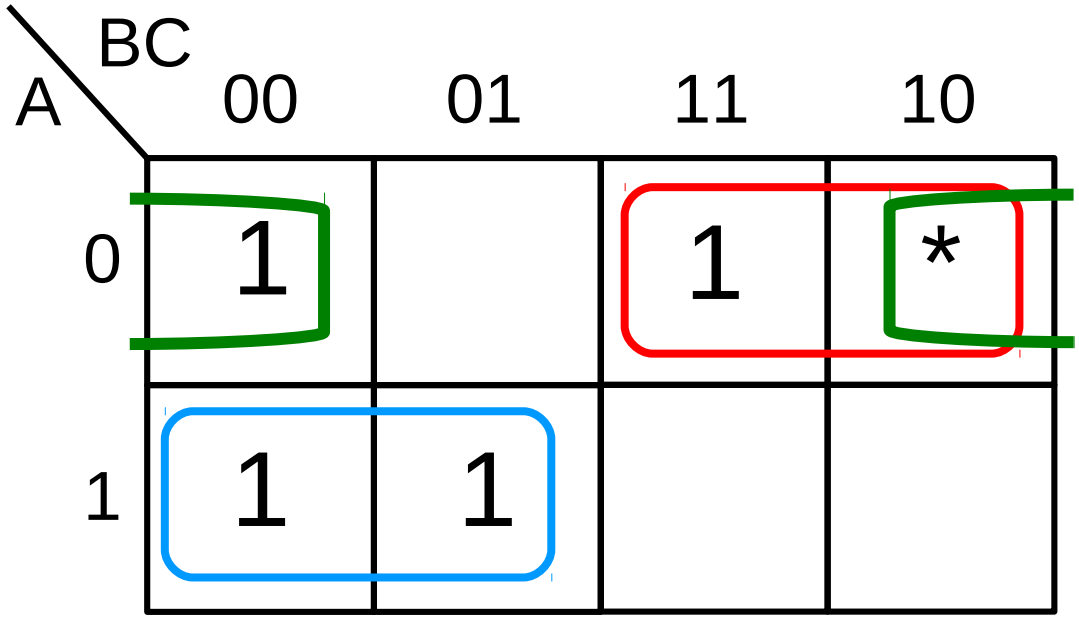



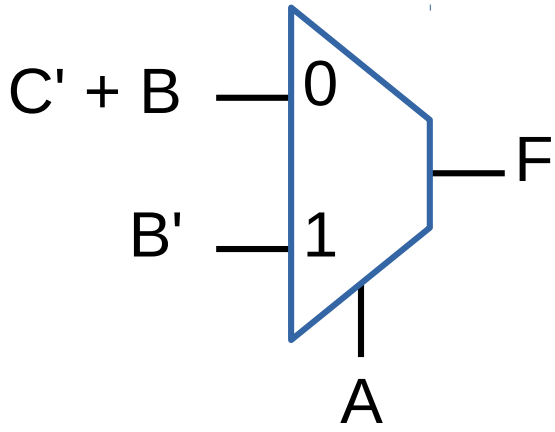



# Ejemplo

Implementar F con un MUX 2 a 1

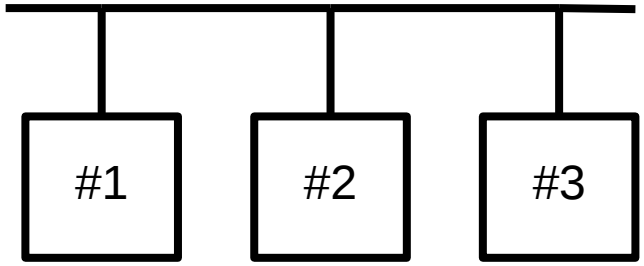
| A | B | C | F |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | * |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |



# Conexión directa de salidas

- Hay casos (implementación de los buses) en la que es deseable poder unir salidas.




- Para esto es necesario que uno de los niveles predomine sobre el otro para evitar:
  - Niveles de voltajes indefinidos
  - Daños en los circuitos

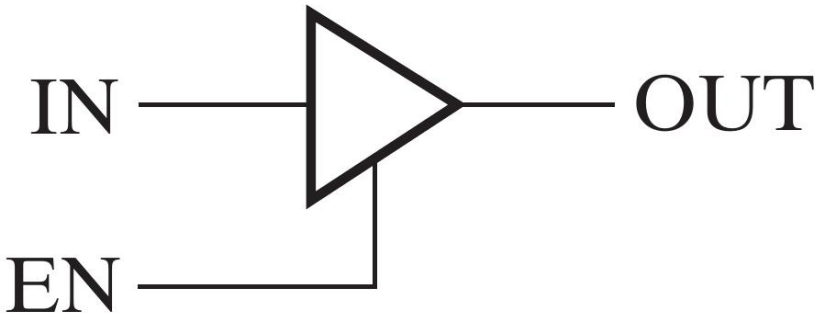
# Conexión directa de salidas

- Lógica Three state, Tristate, 3 state
- Open collector → Tecnología TTL
  - Open drain → Tecnología CMOS

# Three state

- En las compuertas convencionales hay solo dos estados $V_H$ y $V_L$ que se corresponden con el 0 y 1 lógicos.
- El tercer estado es el estado de alta impedancia (Hi-Z) en el que
  - La lógica se comporta como un circuito abierto (desconectada)
  - El circuito conectado a la salida de la compuerta *tristate* no es afectado por la entradas a la compuerta.
- Disponible para cualquier tipo de compuerta. Común en buffers.

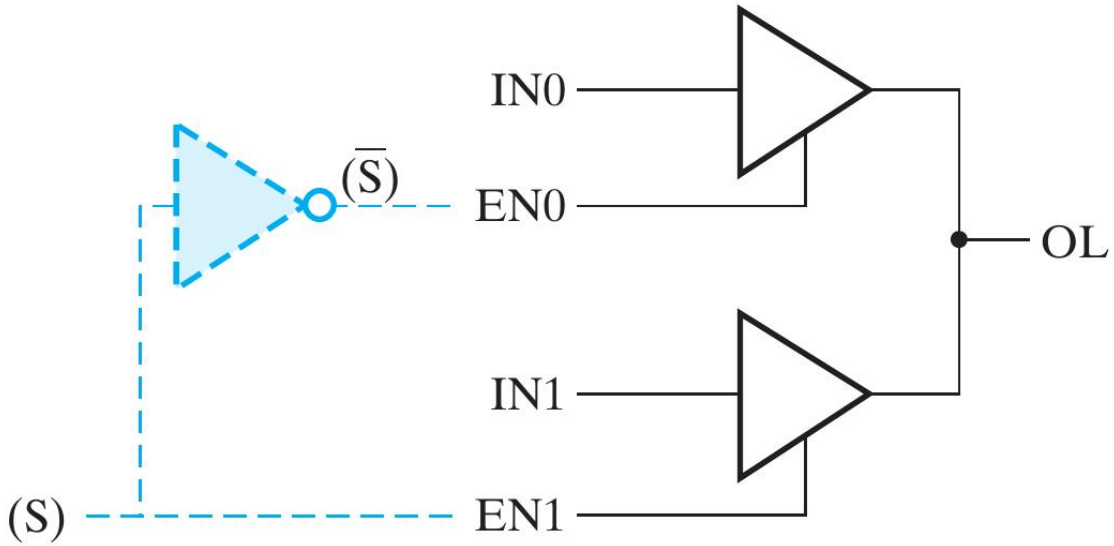
# Three state




| EN | IN | OUT |
|---|---|---|
| 0 | X | Hi-Z |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

# Three state

Los buffers three-state se puede conectar para generar una línea de multiplexado.




| EN1 | EN0 | IN1 | IN0 | OL |
|---|---|---|---|---|
| 0 | 0 | X | X | Hi-Z |
| (S) 0 | $\overline{(S)}$ 1 | X | 0 | 0 |
| 0 | 1 | X | 1 | 1 |
| 1 | 0 | 0 | X | 0 |
| 1 | 0 | 1 | X | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | |
| 1 | 1 | 1 | 0 | |

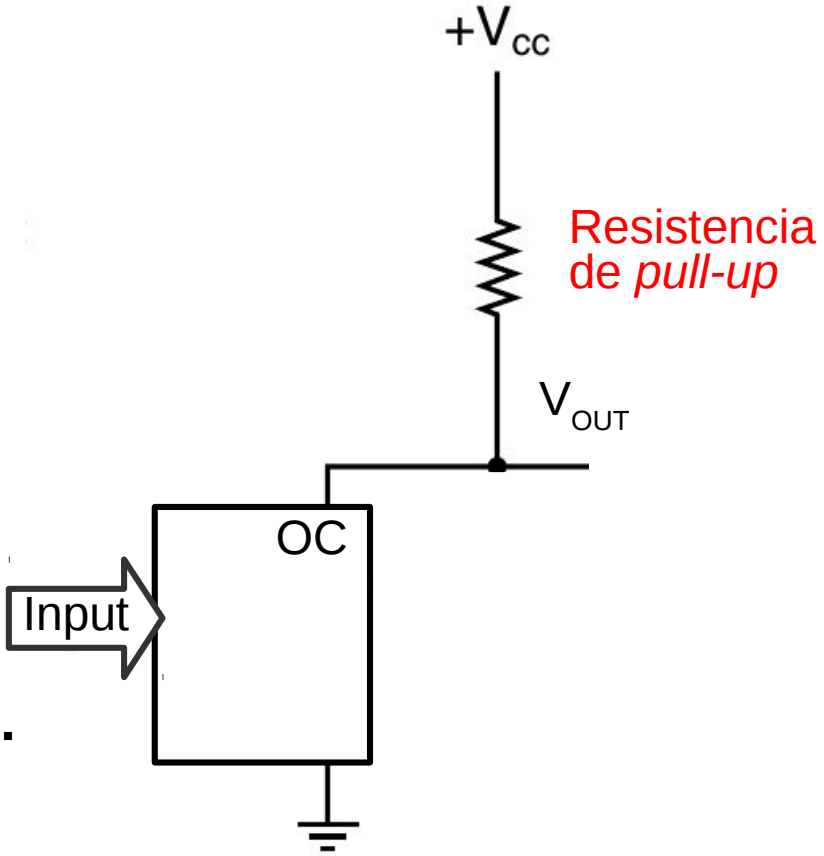
Open collector es más tolerante a fallas.

# Open Collector

- La salida de una compuerta *Open Collector* puede tener dos valores:

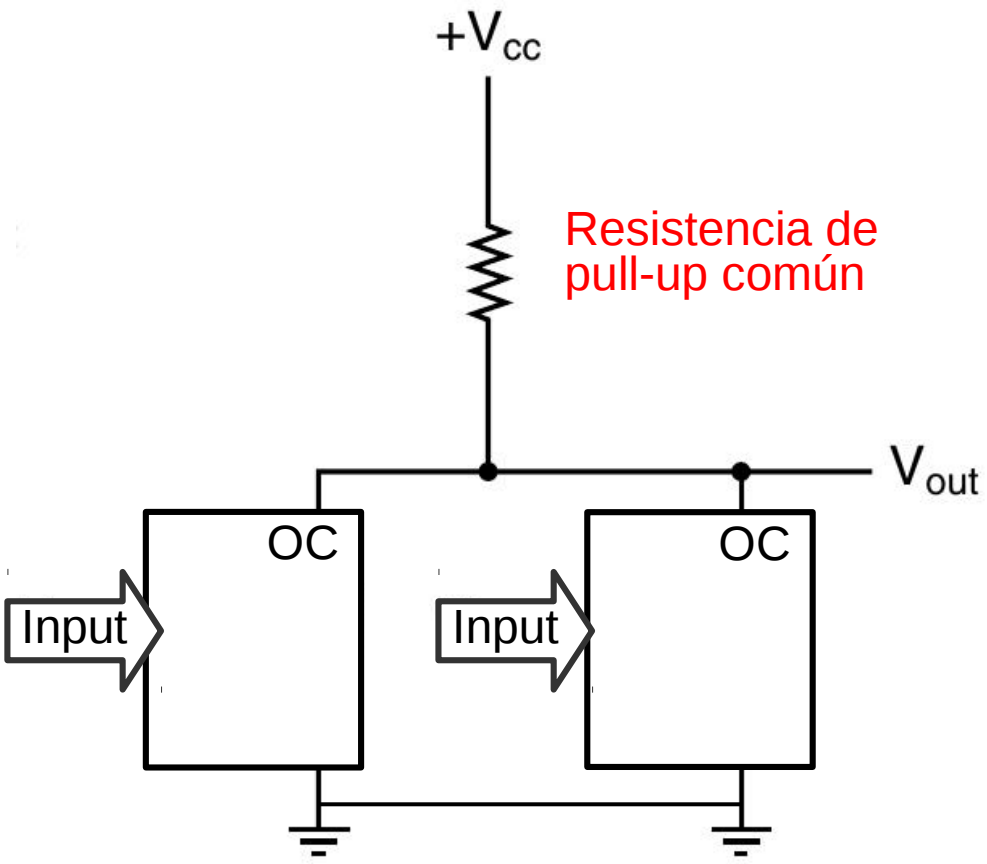
  $V_L$ o Hi-Z.

- La salida de la compuerta OC se une a $V_{cc}$ a través de una resistencia de *pull-up*.
- Si la salida de la compuerta es:
  - $V_L$ , entonces $V_{OUT} = V_L$
  - Hi-Z, entonces $V_{OUT} = V_{CC}$ ($V_H$).



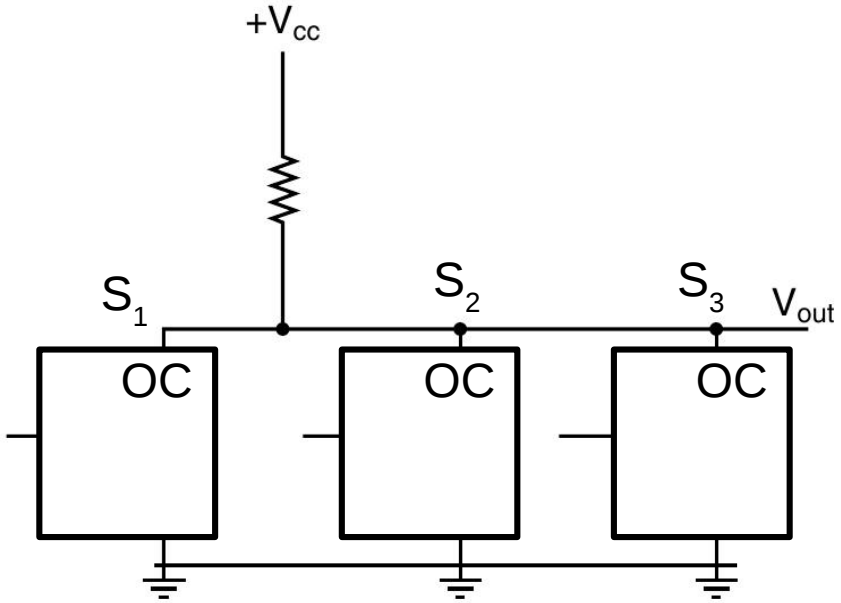
# Open Collector

- Las salidas de varias compuertas *Open Collector* se unen a una misma resistencia de *pull-up*.
- Si la salida de alguna compuerta es $V_L$, entonces $V_{OUT} = V_L$
- Si la salida de ninguna compuerta es $V_L$ (la salida de todas las compuertas es Hi-Z), $V_{OUT} = V_{CC}$ ($V_H$).



# Open Collector

$S_1$, $S_2$ y $S_3$ son las salidas de las compuertas.




| $S_1$ | $S_2$ | $S_3$ | $V_{out}$ |
|---|---|---|---|
| $V_L$ | * | * | $V_L$ |
| * | $V_L$ | * | $V_L$ |
| * | * | $V_L$ | $V_L$ |
| Hi-Z | Hi-Z | Hi-Z | $V_H$ |

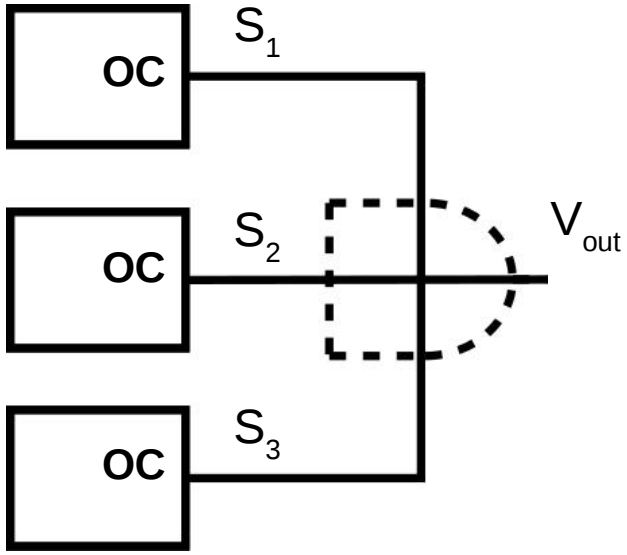
- Si al menos una salida es $V_L$ ($S_1$ o $S_2$ o $S_3$ es $V_L$) entonces $V_{out} = V_L$
- Si ninguna salida es $V_L$ ($S_1$=$S_2$=$S_3$=Hi-Z) entonces $V_{out} = V_H$

# Open Collector

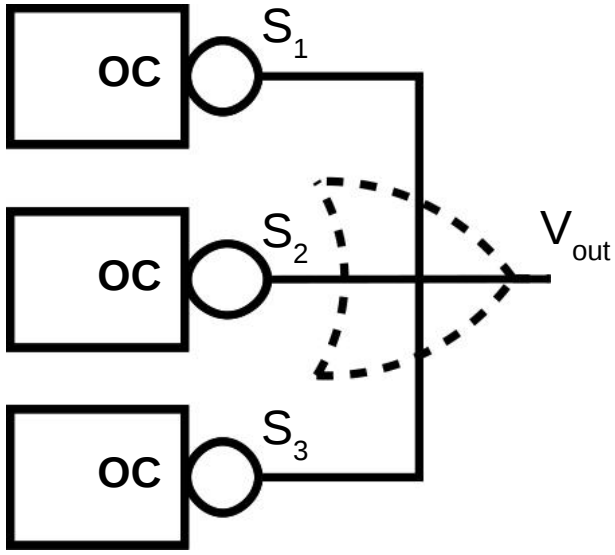
| $S_1$ | $S_2$ | $S_3$ | $V_{out}$ |
|---|---|---|---|
| $V_L$ | * | * | $V_L$ |
| * | $V_L$ | * | $V_L$ |
| * | * | $V_L$ | $V_L$ |
| Hi-Z | Hi-Z | Hi-Z | $V_H$ |

*Implícitamente indican $V_H$*

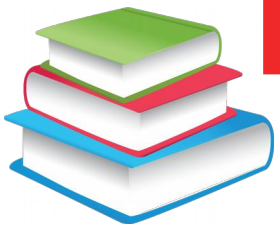
LP → AND




LN → OR



# Bibliografía



- *Capítulo 3* y 5*.* Morris Mano, Kime & Martin. *Logic and computer design fundamentals.* Prentice Hall (5ta Ed. 2015)

*Suplementaria*

- *Apéndice B.* David A. Patterson & John L. Hennessy. *Computer Organization and Design. The Hardware/Software Interface.* Elsevier. (5ta Ed. 2014)

* no completos

# Arquitectura de Computadoras

(Cód. 5561)
1° Cuatrimestre 2018

Dra. Dana K. Urribarri
DCIC - UNS

# Dispositivos Lógicos Programables

# Dispositivos lógicos programables

- Read Only Memory (ROM)
- Programmable Logic Array (PLA)
- Programmable Array Logic (PAL®)
- Field Programmable Gate Array (FPGA)

# Dispositivos lógicos programables

- No tienen una función lógica preestablecida
- Es posible controlar las conexiones o almacenar información para definir la lógica a implementar.
- Para poder ser usados necesitan ser *programados:* un procedimiento de hw que determina la función a implementar.

# Dispositivos lógicos programables

## Tecnologías permanentes

- Fusibles
  - Inicialmente cerrados. Se queman con voltajes superiores a los normales y eso abre la conexión.
- Antifusibles
  - Inicialmente abiertos. Contienen un material no conductor que con voltajes elevados se funde y baja la resistencia cerrando la conexión.
- Programación por máscara
  - La realiza el fabricante durante las últimas fases del proceso de fabricación del chip. Dependiendo de la función a implementar, se realizan o no las conexiones sobre las capas de metal que sirven como conductoras en el chip.

# Dispositivos lógicos programables

## Tecnologías reconfigurables

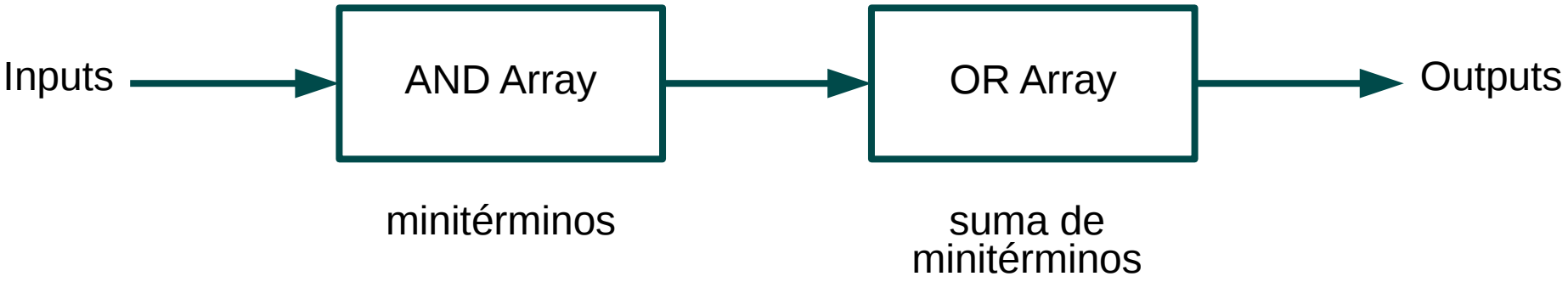
- Un dispositivo de almacenamiento de 1 bit que controla un transistor: si el bit está en 1, el transistor cierra el circuito. Si el bit está en 0, el transistor abre el circuito.

  Es fácilmente reprogramable, pero necesita alimentación.

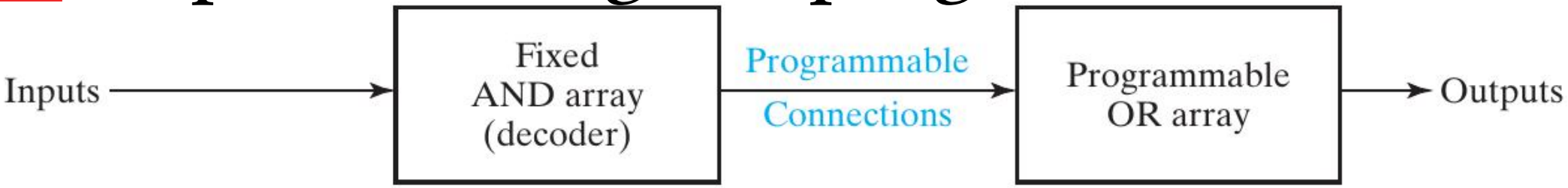
- Basada en transistores *floating-gate* (transistor que tiene una puerta flotante aislada en el interior) conectada de forma capacitiva. Como está aislada permite mantener la carga por largos períodos de tiempo.

# Dispositivos lógicos programables

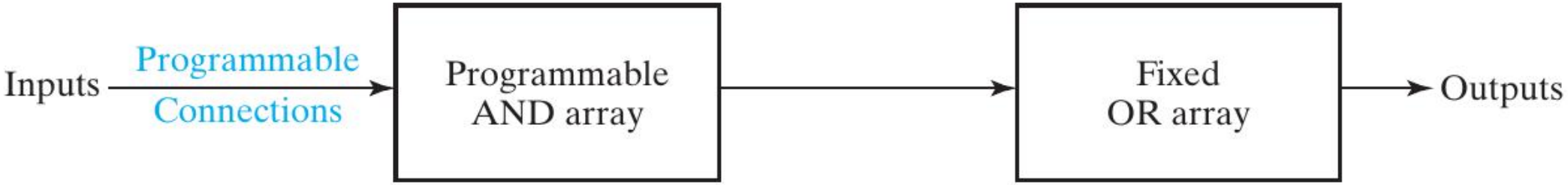
- Definición genérica de una función lógica



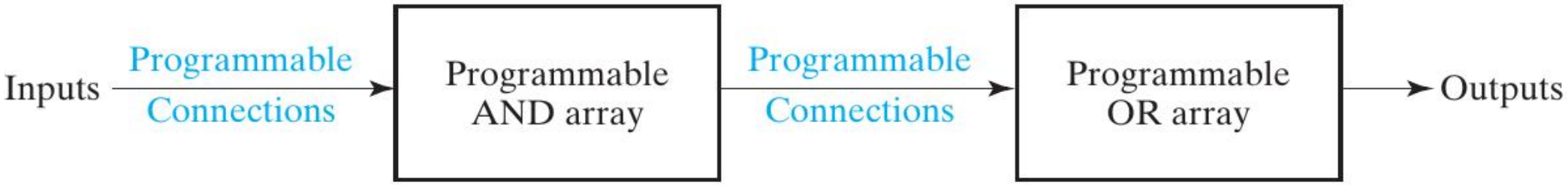
# Dispositivos lógicos programables




(a) Programmable read-only memory (PROM)



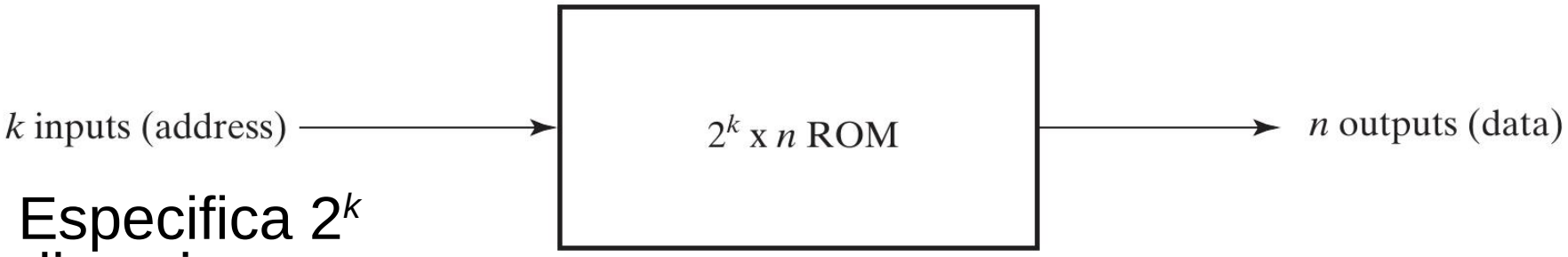

(b) Programmable array logic (PAL) device




(c) Programmable logic array (PLA) device

# ROM

- No volátil y permanente.




Especifica $2^k$ direcciones distintas

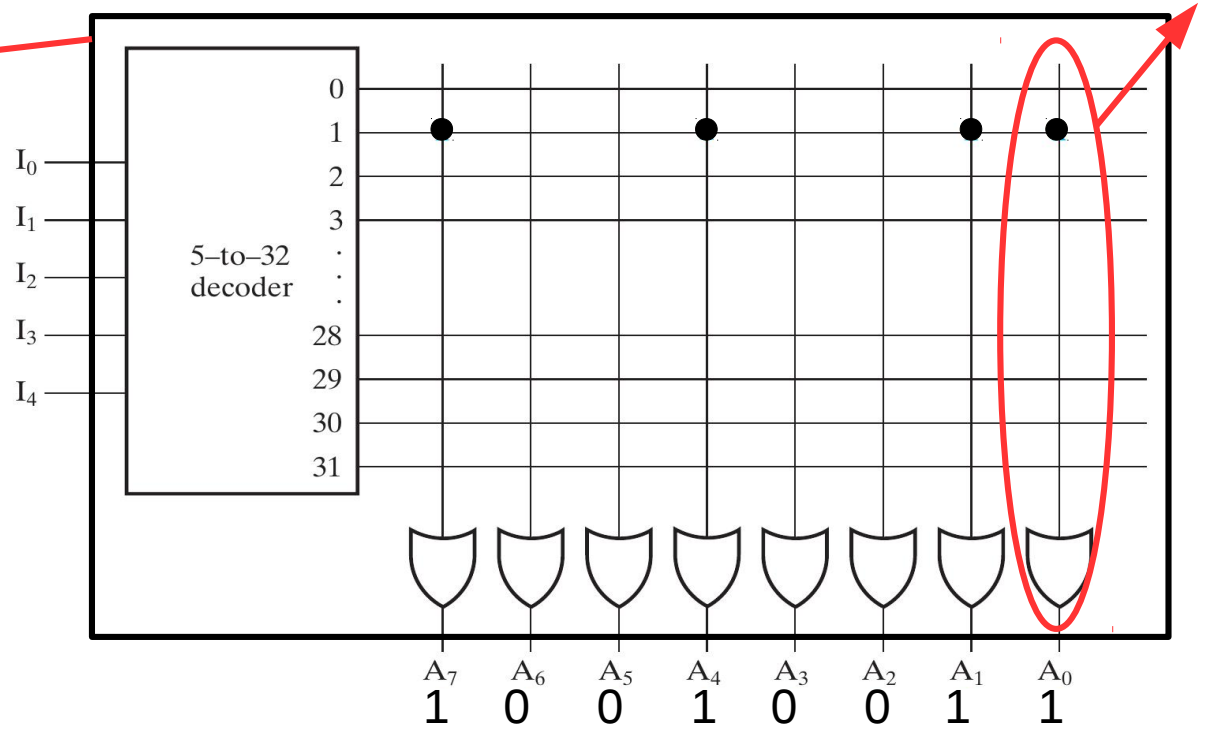
- Permite especificar completamente *n* funciones de *k* entradas.
- Tamaño: $2^k \times n$ bits

# ROM

- Una ROM de $2^k$ x $n$ tendrá internamente un decoder de $k$ a $2^k$ y $n$ compuertas OR.
- Cada OR tendrá $2^k$ entradas conectadas de manera programable a cada salida del decoder.

ROM de 32x8:
32 palabras de 8 bits cada una.

5 líneas de entrada y
8 líneas de salida.



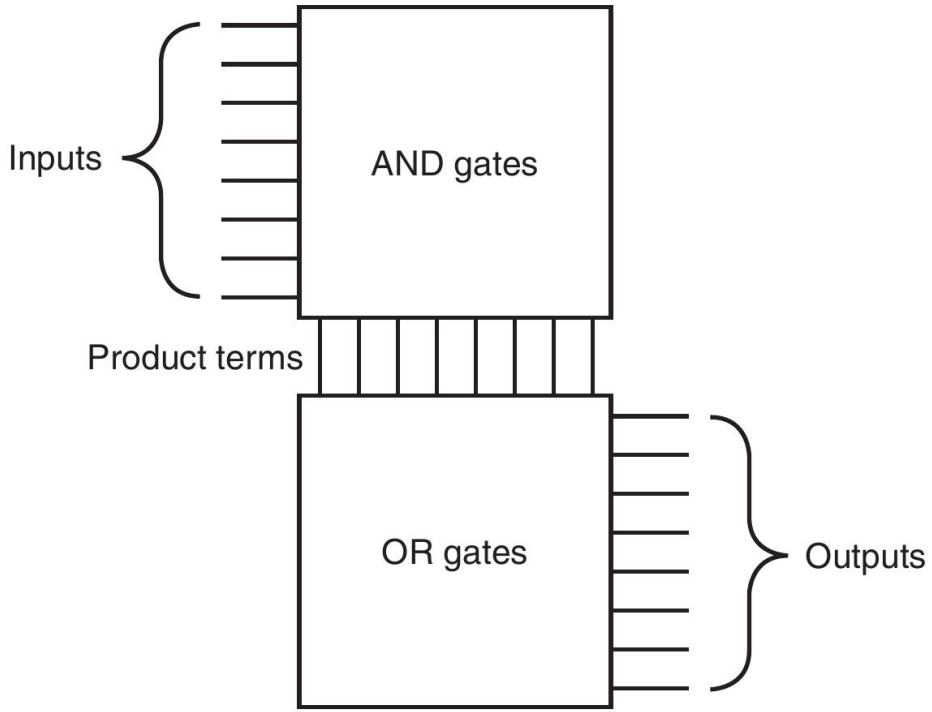
# ROM

- Dependiendo de la tecnología de programación:
  - ROM → programación por máscara
  - PROM → programación por fusible o antifusible
  - EPROM → tecnología que usa transistores *floating-gate*. Programable electrónicamente y borrable con luz ultravioleta.
  - EEPROM → tecnología que usa transistores *floating-gate.* Programable y borrable electrónicamente.
  - Memoria FLASH → Mejora de la EEPROM.

# PLA

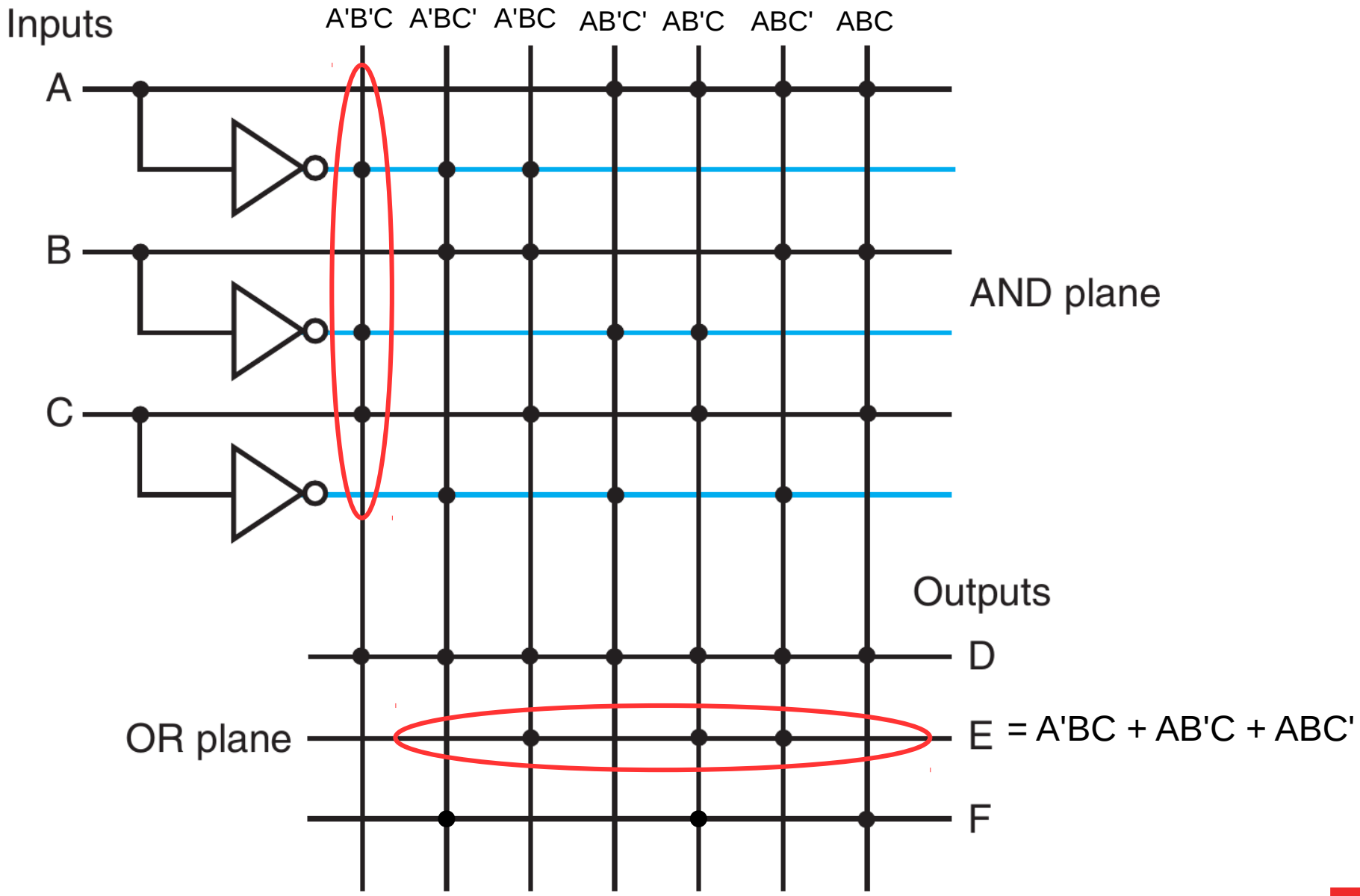
- Similar a la ROM
- No genera todos los posibles minitérminos.
- El decoder se reemplaza por un arreglo de AND que se pueden programar para generar términos productos.
- Los términos productos se conectan selectivamente a compuertas OR para generar la función requerida.
- Tamaño: cantidad de conexiones disponibles

# PLA



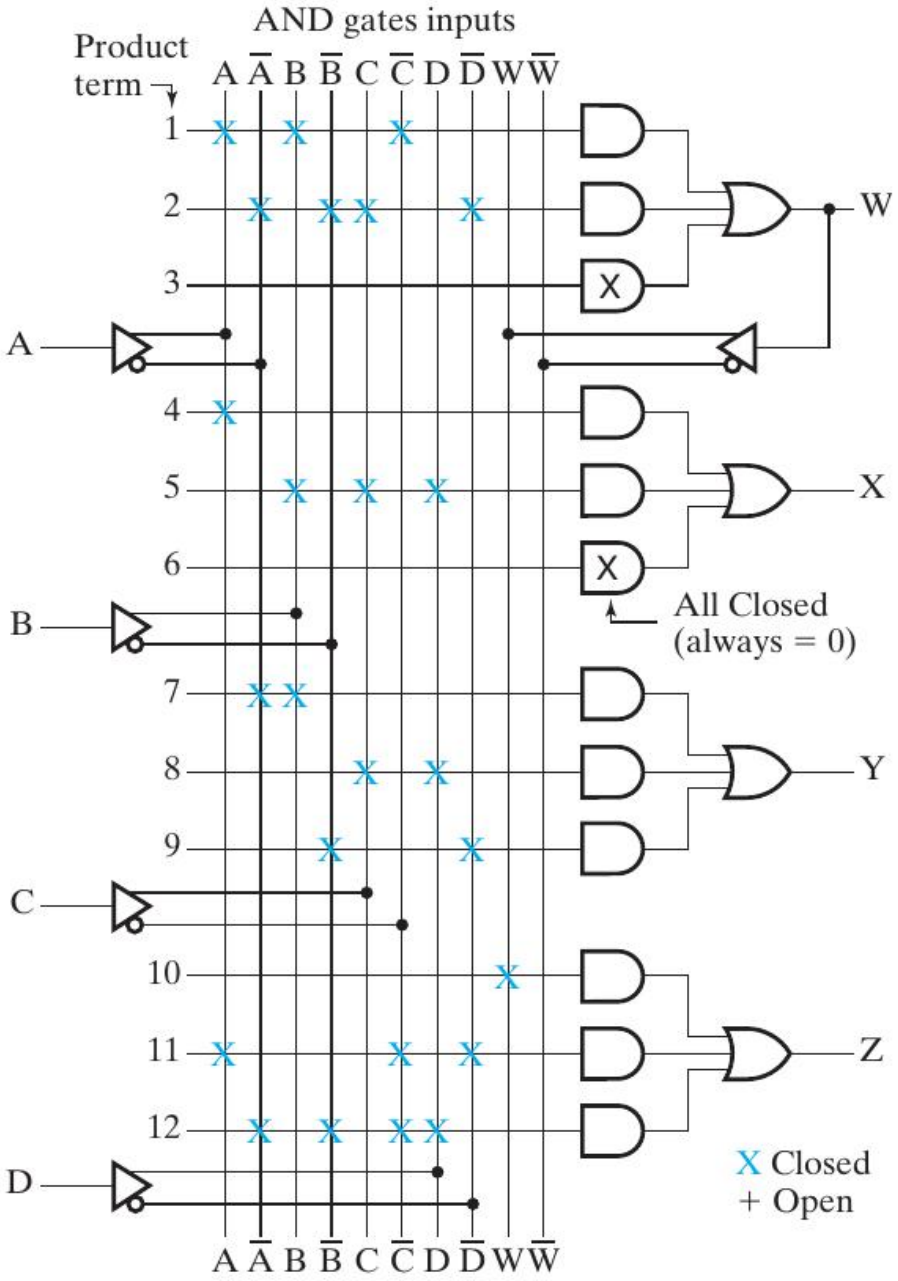

Tamaño: (nro entradas + nro de salidas) × nro términos producto

# PLA


Inputs
A
B
C
A'B'C
A'BC'
A'BC
AB'C'
AB'C
ABC'
ABC
AND plane
Outputs
D
E = A'BC + AB'C + ABC'
F
OR plane

# PAL®

- Similar al PLA pero no tan flexible.
- El plano de AND es programable.
- El plano de OR es fijo.
- Los minitérminos no se pueden compartir entre múltiples salidas.



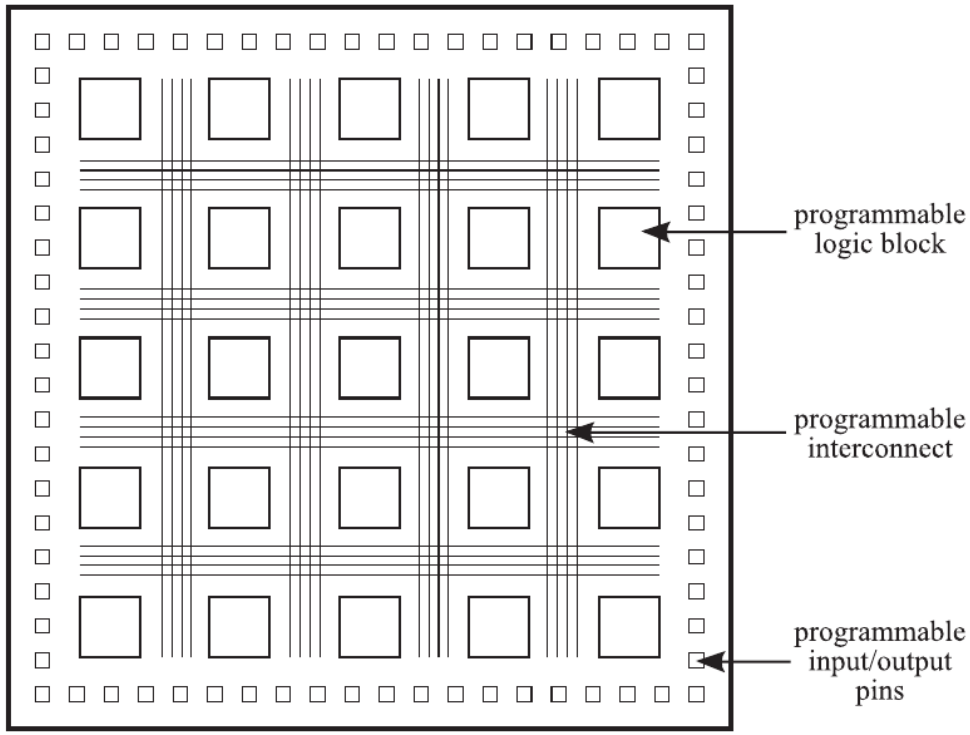
# FPGA

- Interconectan elementos básicos para lograr funcionalidad más avanzada.
  - Bloques de lógica programable
  - Interconexiones programables entre los bloques
  - Pines de entrada/salida programables
- Adicionalmente, bloques de lógica dedicada:
  - Memorias
  - Unidades aritmético/lógicas
  - Microprocesadores

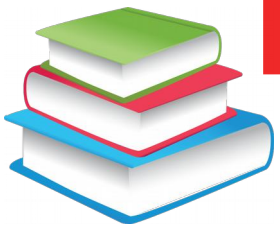
# FPGA

Conceptualmente: un arreglo 2D de celdas con posibles interconexiones entre ellas.

- Las celdas consisten de tablas de lookup (LUT): una pequeña cantidad de lógica y RAM.
- Configurar el FPGA requiere configurar los pines de entrada, los bloques de lógica programable, el conexionado entre los bloques, y los pines de salida.



# Bibliografía



- *Capítulo 3* y 5*.* Morris Mano, Kime & Martin. *Logic and computer design fundamentals.* Prentice Hall (5ta Ed. 2015)

*Suplementaria*

- *Apéndice B.* David A. Patterson & John L. Hennessy. *Computer Organization and Design. The Hardware/Software Interface.* Elsevier. (5ta Ed. 2014)

* no completos