

# Fundamentos de Electrónica

2ª parte : Electrónica Digital

**Teresa Riesgo Alcaide**

[teresa.riesgo@upm.es](mailto:teresa.riesgo@upm.es)

©UNIVERSIDAD POLITÉCNICA DE MADRID



# La Asignatura “Fundamentos de Electrónica”

## ¿Cómo se aprueba?

- Para aprobar la asignatura se ha de obtener una nota igual o superior a 5 en la Calificación Final ( $CF \geq 5$  sobre 10) de la asignatura y haber hecho las tres prácticas de laboratorio obligatorias.
- La asignatura tiene tres partes (Digital, Microprocesadores y Analógica) y en cada parte se obtiene una calificación por evaluación continua (CECdig, CECmic y CECana). La evaluación continua de cada parte puede estar compuesta por una o varias actividades: prueba de evaluación continua (PEC), trabajo, etc. Las actividades que componen la evaluación continua y el peso de cada una se definirá en cada parte de la asignatura. La Calificación por Evaluación Continua de la asignatura CEC será:

$$CEC = 0,3 \cdot CECdig + 0,3 \cdot CECmic + 0,4 \cdot CECana$$

estando puntuada cada parte sobre 10.

- La asignatura tiene una prueba global obligatoria que se ha de realizar en la convocatoria ordinaria o en la extraordinaria, y se denomina CPG a la Calificación obtenida en esta Prueba Global (puntuada sobre 10).
- La calificación final (CF) de la asignatura se calcula como:
  - Si  $CPG \geq 4 \rightarrow CF = CPG + 0,3 \cdot CEC$
  - Si  $CPG < 4 \rightarrow CF = CPG$
- Si el alumno no aprueba en la convocatoria ordinaria, se le conservan las calificaciones por evaluación continua de cada parte para la convocatoria extraordinaria, aplicándose en esta convocatoria el mismo algoritmo del punto anterior.

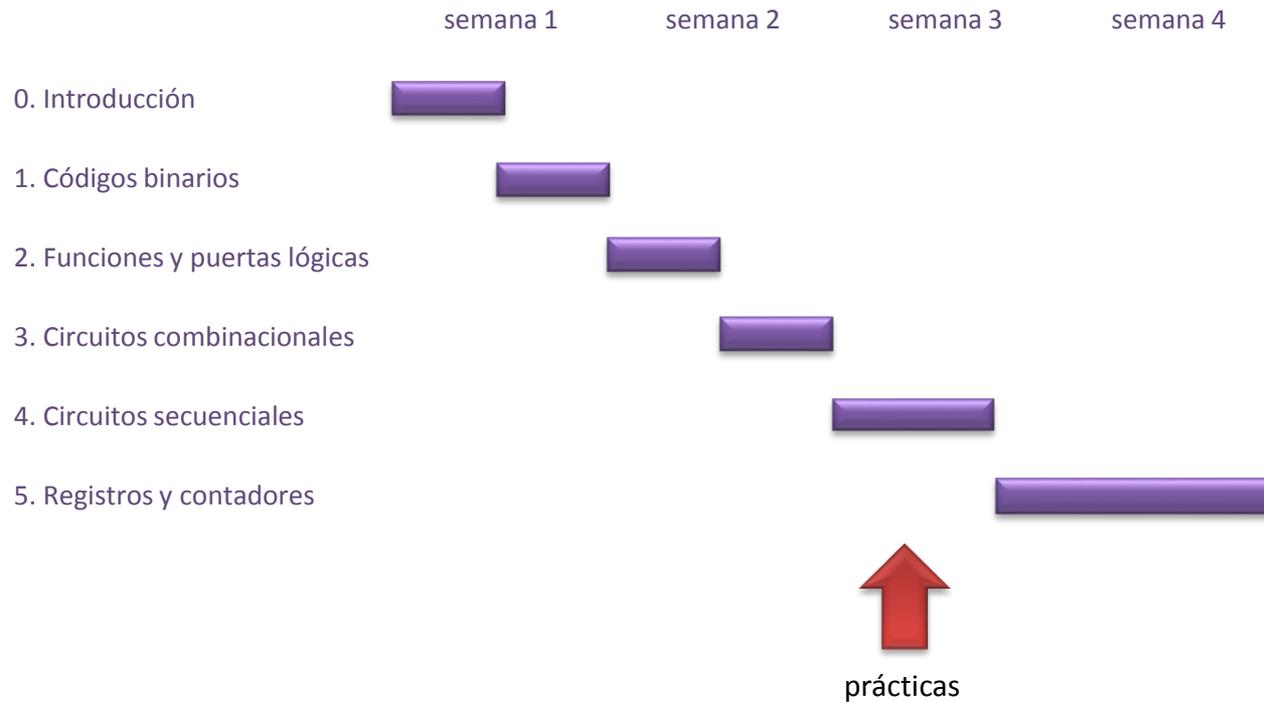
# La Asignatura “Fundamentos de Electrónica”

¿Cómo se aprueba?

- Para la parte de **Electrónica Digital...**
  - Las prácticas son obligatorias (del 23 al 26 de septiembre)
  - 80% de la nota de evaluación continua (CECdig) es de una Prueba de Evaluación Continua (6/10 a las 18h)
  - 20% de la nota de evaluación continua (CECdig) es de un trabajo de diseño (a entregar hasta el 17/10)

# FE: Electrónica Digital – Contenidos y Programación

## Fundamentos de Electrónica - Digital (2015)

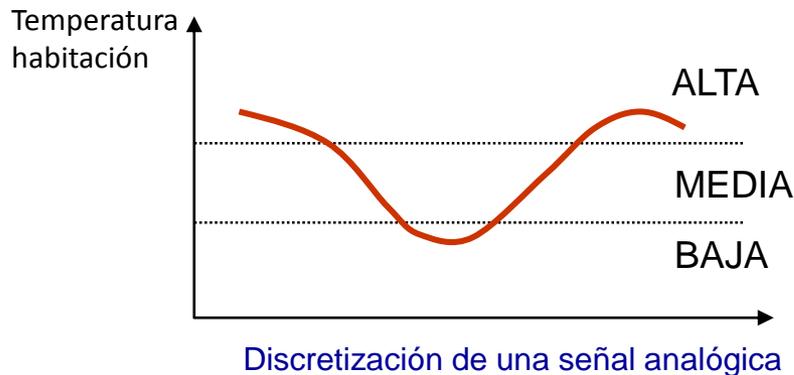


# Bibliografía

- Digital Design and Computer Architecture.  
David Money and Sarah Harris. Elsevier-Morgan Kauffmann (2007)
- Fundamentos de Sistemas Digitales  
Thomas Floyd. Ed. Prentice Hall (2003)
- Problemas de circuitos y sistemas digitales  
Baena, Bellido, Molina, Parra. Ed. McGraw-Hill (1997)
- Colecciones de exámenes, disponibles en la secretaría de Electrónica

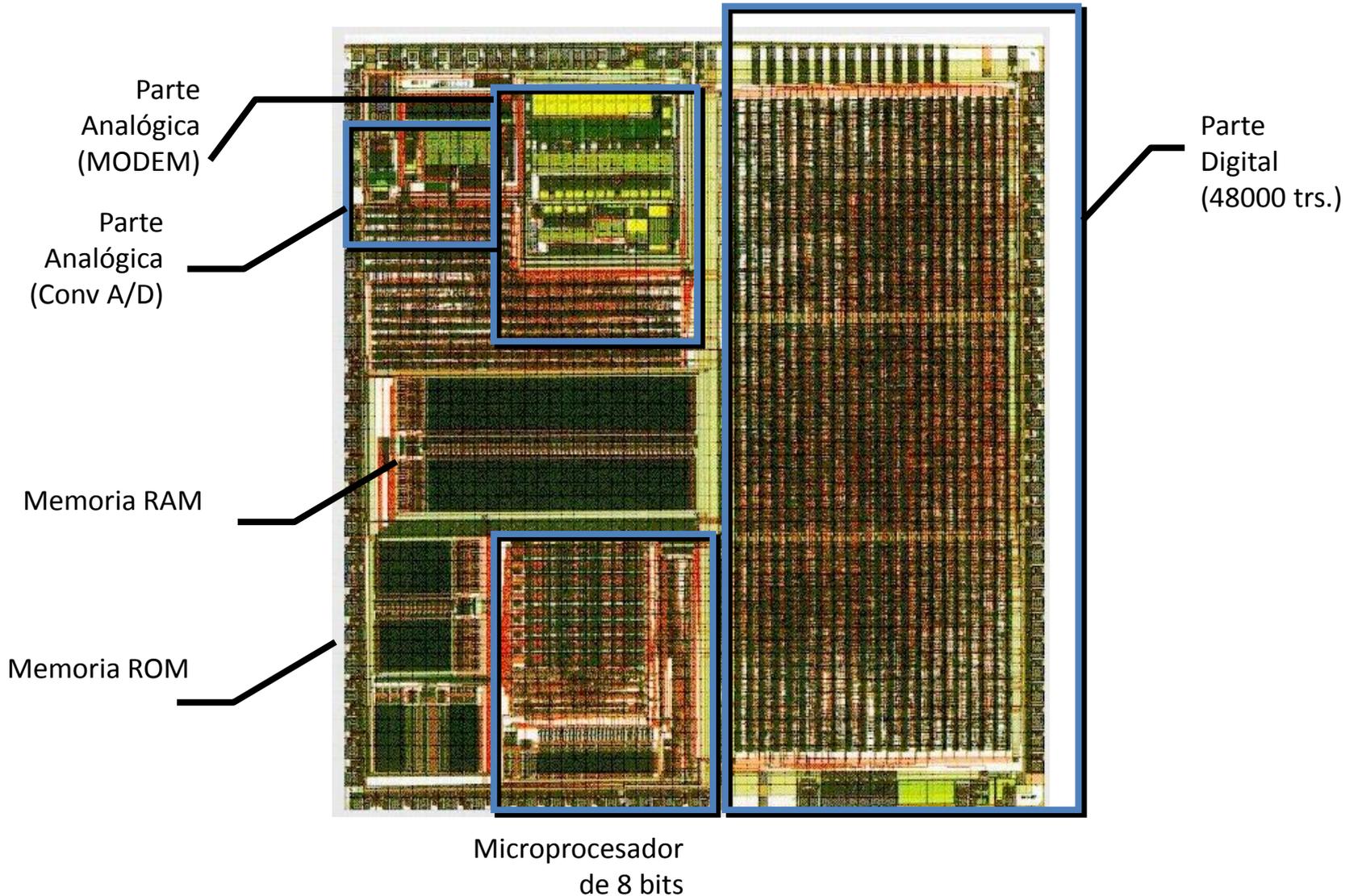
# Análogo y Digital

- Una **señal digital** puede tomar un valor entre un número finito de valores o estados
- Una **señal analógica** puede tomar un número infinito de valores



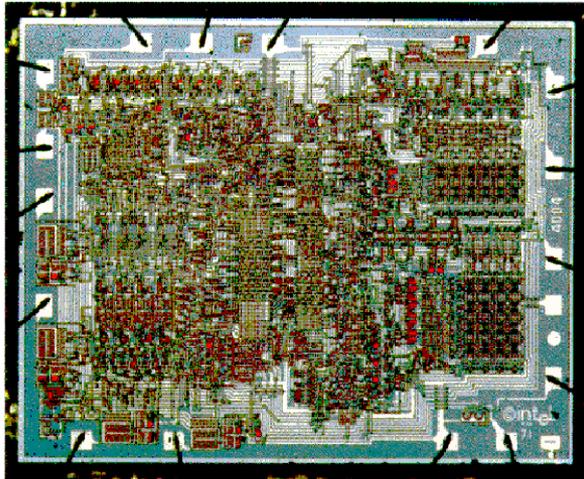
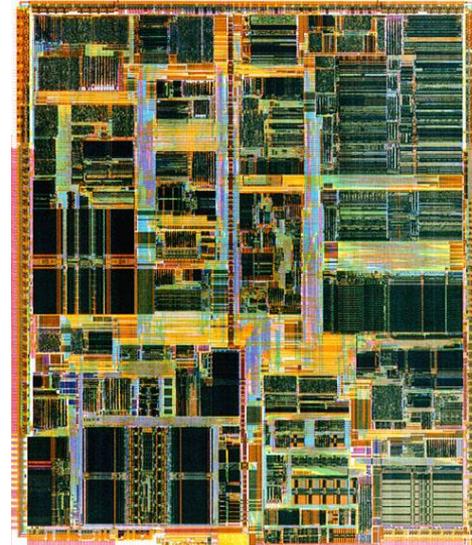
- Mundo real es un mundo analógico...¿Por qué usar SISTEMAS DIGITALES?  
Porque nos dan
  - Capacidad para manejar gran cantidad de información
    - Fácil de transmitir
    - Muy inmune al ruido
  - Gran desarrollo de la tecnología: CI (integran millones de trtr)
  - Microprocesadores: Gran capacidad de cálculo

# Integración de funciones analógicas y digitales



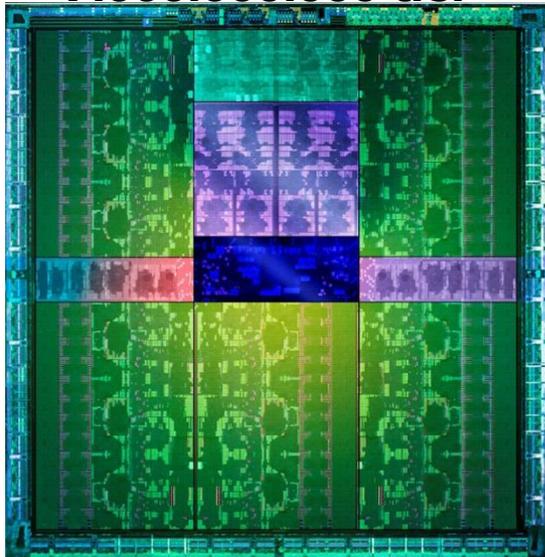
# Ejemplo: evolución de los procesadores

**Pentium II (1996) 10.000.000 trs.**

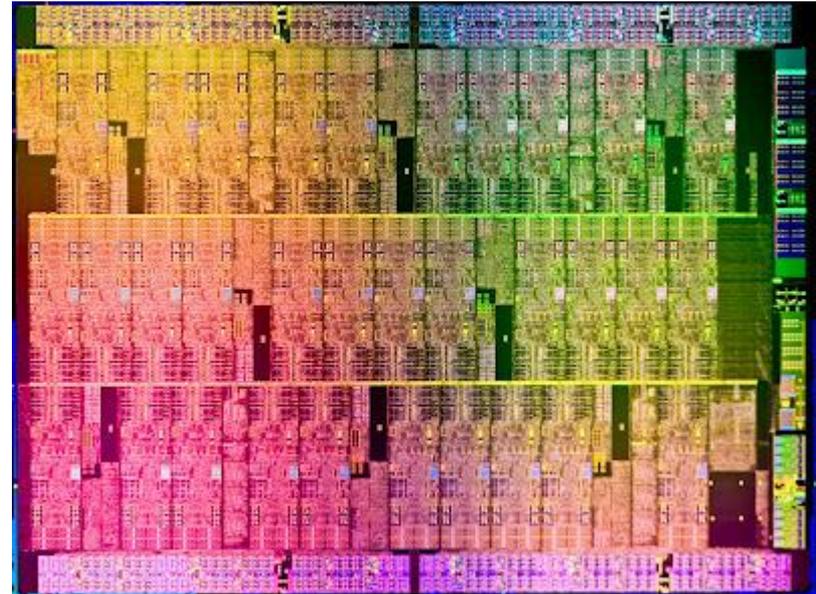


**Intel 4004 (1970) 1.500 trs.**

**nVIDIA GK110 (2012)  
7.000.000.000 trs.**

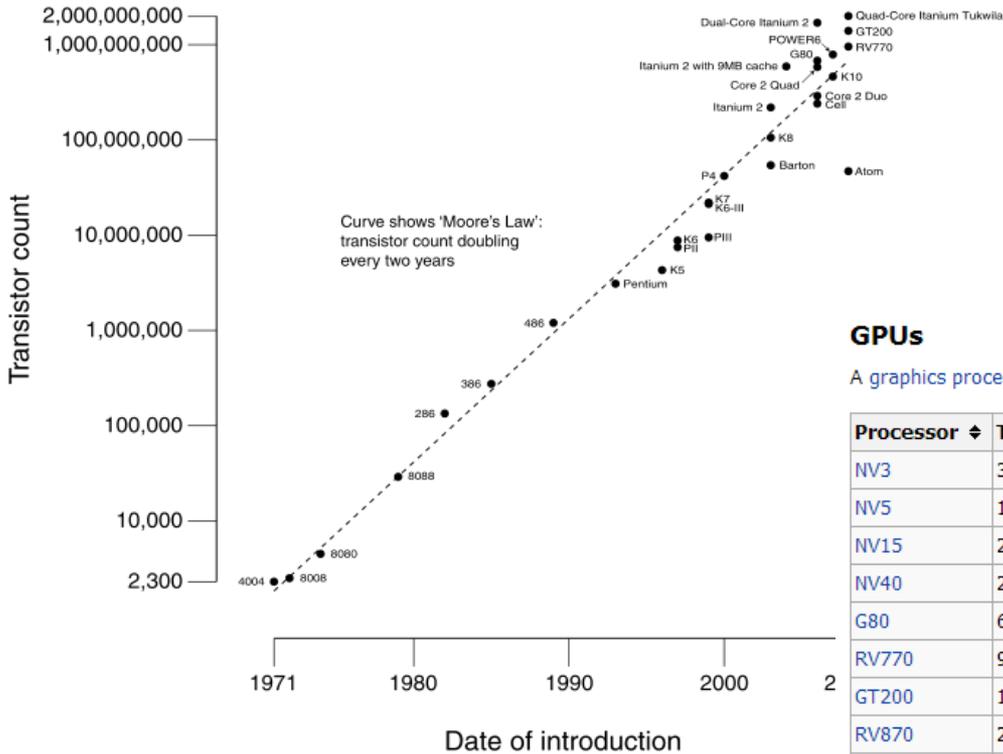


**Intel Xeon-Phi (2012) 5.000.000.000 trs.**



# Evolución...

## CPU Transistor Counts 1971-2008 & Moore's Law



## GPUs

A [graphics processing unit](#) (GPU) is a specialized electronic circuit designed to rapidly manipulate and alter mem

| Processor | Transistor count              | Date of introduction | Manufacturer | Process | Area                |
|-----------|-------------------------------|----------------------|--------------|---------|---------------------|
| NV3       | 3,500,000                     | 1997                 | NVIDIA       | 350 nm  |                     |
| NV5       | 15,000,000                    | 1999                 | NVIDIA       | 250 nm  |                     |
| NV15      | 25,000,000                    | 2000                 | NVIDIA       | 180 nm  |                     |
| NV40      | 222,000,000                   | 2004                 | NVIDIA       | 130 nm  | 305 mm <sup>2</sup> |
| G80       | 681,000,000                   | 2006                 | NVIDIA       | 90 nm   | 480 mm <sup>2</sup> |
| RV770     | 956,000,000 <sup>[10]</sup>   | 2008                 | AMD          | 55 nm   | 260 mm <sup>2</sup> |
| GT200     | 1,400,000,000 <sup>[11]</sup> | 2008                 | NVIDIA       | 55 nm   | 576 mm <sup>2</sup> |
| RV870     | 2,154,000,000 <sup>[12]</sup> | 2009                 | AMD          | 40 nm   | 334 mm <sup>2</sup> |
| Cayman    | 2,640,000,000                 | 2010                 | AMD          | 40 nm   | 389 mm <sup>2</sup> |
| 1st GF100 | 3,200,000,000 <sup>[13]</sup> | Mar 2010             | NVIDIA       | 40 nm   | 526 mm <sup>2</sup> |
| 2nd GF100 | 3,000,000,000 <sup>[14]</sup> | Nov 2010             | NVIDIA       | 40 nm   | 520 mm <sup>2</sup> |
| Tahiti    | 4,310,000,000 <sup>[15]</sup> | 2011                 | AMD          | 28 nm   | 365 mm <sup>2</sup> |
| GK104     | 3,540,000,000 <sup>[16]</sup> | 2012                 | NVIDIA       | 28 nm   | 294 mm <sup>2</sup> |
| GK110     | 7,100,000,000 <sup>[17]</sup> | 2012                 | NVIDIA       | 28 nm   |                     |

# TEMA 1

## Códigos binarios

# Sistemas de numeración

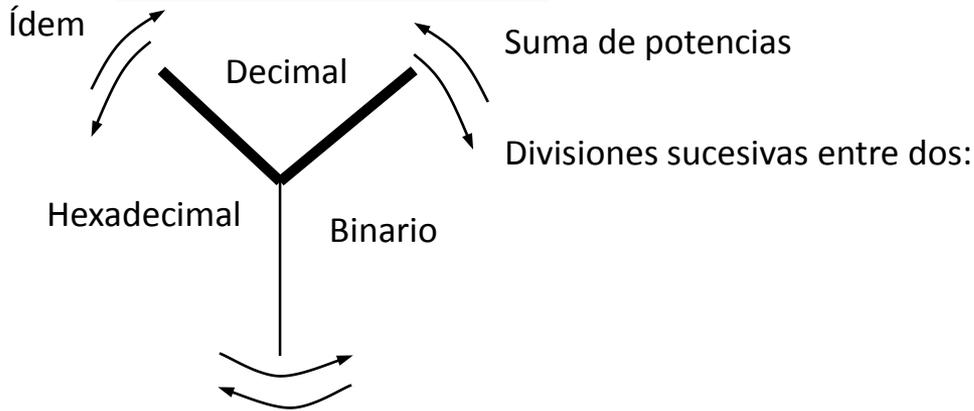
$$1327 = 1 \times 10^3 + 3 \times 10^2 + 2 \times 10^1 + 7 \times 10^0$$

$$N = p_{n-1} \cdot b^{n-1} + p_{n-2} \cdot b^{n-2} + \dots + p_1 \cdot b^1 + p_0 \cdot b^0$$

|  |                     |         |                        |
|--|---------------------|---------|------------------------|
|  $b = 10$ | Sistema decimal     | Dígitos | 0, 1, 2, ..., 9        |
|  $b = 2$  | Sistema binario     | Dígitos | 0, 1 BIT               |
|  $b = 16$ | Sistema hexadecimal | Dígitos | 0, 1, ..., 9, A, ... F |
| $b = 8$  | Sistema octal       | Dígitos | 0, ..., 7              |

# Algunas propiedades de los sistemas de numeración

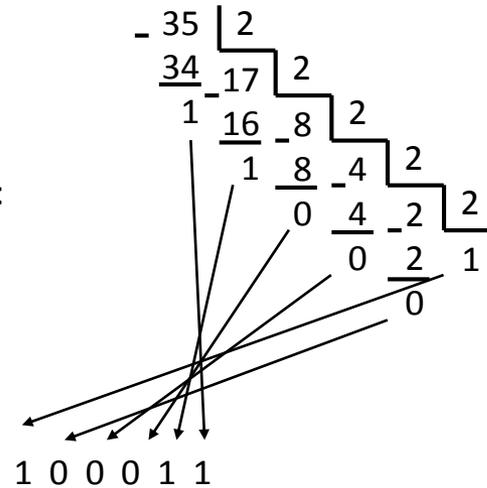
## Cambios de base



Inmediato:

C3A5 = 1100 0011 1010 0110

C
3
A
5



## Multiplicar y dividir

Decimal:

$$9875 \times 100 = 987500$$

Binario:

$$10110 \times 10 = 101100$$

(x2)

$$110011 \times 100 = 11001100 \text{ (x4)}$$

$$101010 / 100 = 1010 \text{ R } 10 \text{ (/4)}$$

Hex:

$$ABCD \times 100 = ABCD00 \text{ (x256)}$$

# Códigos binarios

## Números positivos

Binario natural

## Números negativos

Bit de signo + magnitud

SMMMMM

$S=0 \rightarrow$  positivo;  $S=1 \rightarrow$  negativo

Complemento a 1

Cambiar signo  $\leftrightarrow$  Intercambiar 0s y 1s

Complemento a 2

Cambiar signo  $\leftrightarrow$  Compl. a 1 + 1

## Números reales

Estándares IEEE (Ej. IEEE Std 754, 1985)

|          |                  |                |
|----------|------------------|----------------|
| <b>S</b> | <b>exponente</b> | <b>mantisa</b> |
|----------|------------------|----------------|

# Representaciones de números binarios negativos utilizando 4 bits

| Decimal | Binario signo y magnitud | Binario compl. a 1 | Binario compl. a 2 |
|---------|--------------------------|--------------------|--------------------|
| -8      | --                       | --                 | 1000               |
| -7      | 1 111                    | 1000               | 1001               |
| -6      | 1 110                    | 1001               | 1010               |
| -5      | 1 101                    | 1010               | 1011               |
| -4      | 1 100                    | 1011               | 1100               |
| -3      | 1 011                    | 1100               | 1101               |
| -2      | 1 010                    | 1101               | 1110               |
| -1      | 1 001                    | 1110               | 1111               |
| 0       | 1 000<br>0 000           | 1111<br>0000       | 0000               |
| 1       | 0 001                    | 0001               | 0001               |
| 2       | 0 010                    | 0010               | 0010               |
| 3       | 0 011                    | 0011               | 0011               |
| 4       | 0 100                    | 0100               | 0100               |
| 5       | 0 101                    | 0101               | 0101               |
| 6       | 0 110                    | 0110               | 0110               |
| 7       | 0 111                    | 0111               | 0111               |

Usando compl. a 2,  
las restas son sumas

$$\begin{array}{r} -4 \\ +2 \\ \hline -2 \end{array} \quad \begin{array}{r} 1100 \\ 0010 \\ \hline 1110 \end{array}$$

$$\begin{array}{r} -6 \\ +7 \\ +1 \\ \hline \end{array} \quad \begin{array}{r} 1010 \\ 0111 \\ 10001 \end{array}$$

Se elimina el posible acarreo

# Código BCD (Binary Coded Decimal)

El código BCD representa números decimales, codificados en binario dígito a dígito

|   | 8421 | 2421 | n + 3 |
|---|------|------|-------|
| 0 | 0000 | 0000 | 0011  |
| 1 | 0001 | 0001 | 0100  |
| 2 | 0010 | 0010 | 0101  |
| 3 | 0011 | 0011 | 0110  |
| 4 | 0100 | 0100 | 0111  |
| 5 | 0101 | 1011 | 1000  |
| 6 | 0110 | 1100 | 1001  |
| 7 | 0111 | 1101 | 1010  |
| 8 | 1000 | 1110 | 1011  |
| 9 | 1001 | 1111 | 1100  |

BCD Natural      Aiken      Exceso tres

Autocomplementarios

Ejemplo: BCD natural

37d = 0011 0111

Conversión BCD a decimal inmediata

n y 9-n son uno el inverso del otro  
 → Mismo número de 1s que de 0s

# TEMA 2

## Funciones y puertas lógicas

# Variables, operaciones y funciones lógicas

## Variables lógicas

$$A = \{0,1\}$$

## Operaciones lógicas

SUMA  $\rightarrow$  OR

| A | B | A+B |
|---|---|-----|
| 0 | 0 | 0   |
| 0 | 1 | 1   |
| 1 | 0 | 1   |
| 1 | 1 | 1   |

PRODUCTO  $\rightarrow$  AND

| A | B | A·B |
|---|---|-----|
| 0 | 0 | 0   |
| 0 | 1 | 0   |
| 1 | 0 | 0   |
| 1 | 1 | 1   |

NEGACIÓN  
o COMPLEMENTO  $\rightarrow$  NOT

| A | $\bar{A}$ |
|---|-----------|
| 0 | 1         |
| 1 | 0         |

## Funciones

$$F = f(A, B, C, \dots)$$
$$f(A, B, C) = \bar{A} \cdot \bar{B} \cdot C + A \cdot B \cdot C$$

# Teoremas fundamentales

## Teorema del cierre

El resultado de aplicar cualquier función booleana a variables booleanas tiene como resultado una variable booleana

## Teorema de idempotencia

$$A + A = A$$

$$A \cdot A = A$$

## Teorema de involución

$$\overline{\overline{A}} = A$$

## Propiedad conmutativa

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

## Propiedad asociativa

$$(A + B) + C = A + (B + C)$$

$$(A \cdot B) \cdot C = A \cdot (B \cdot C)$$

## Propiedad distributiva

$$A \cdot (B + C) = A \cdot B + A \cdot C$$

$$A + B \cdot C = (A + B) \cdot (A + C)$$

# Teoremas fundamentales

## Teorema de absorción

$$A + A \cdot B = A$$

$$A \cdot (A + B) = A$$

## Ley de De Morgan

$$\overline{A + B} = \bar{A} \cdot \bar{B}$$

$$\overline{A \cdot B} = \bar{A} + \bar{B}$$

N variables:

$$\overline{A + B + C \dots} = \bar{A} \bar{B} \bar{C} \dots$$

$$\overline{A \cdot B \cdot C \dots} = \bar{A} + \bar{B} + \bar{C} + \dots$$

## Ley de De Morgan generalizada

$$\overline{F(A, B, C, \dots, +, \cdot)} = F(\bar{A}, \bar{B}, \bar{C}, \dots, \cdot, +)$$

Ejemplo:  $f(A, B, C, D, E) = [(A + C)\bar{D} + BE](D + E)$   
 $\overline{f(A, B, C, D, E)} = (\bar{A}\bar{C} + D) \cdot (\bar{B} + \bar{E}) + \bar{D}\bar{E}$

# Formas canónicas

$$F(A,B,C,\dots) = AF(1,B,C) + \bar{A}F(0,B,C)$$

Ejemplo  $f(A,B) = Af(1,B) + \bar{A}f(0,B) =$   
 $= ABf(1,1) + A\bar{B}f(1,0) + \bar{A}Bf(0,1) + \bar{A}\bar{B}f(0,0)$  **1ª forma canónica**

$$F(A,B,C,\dots) = [A + F(0,B,C,\dots)][\bar{A} + F(1,B,C,\dots)]$$

Ejemplo  $f(A,B) = [A + f(0,B)][\bar{A} + f(1,B)] =$   
 $= [A + B + f(0,0)][A + \bar{B} + f(0,1)]$   
 $[A + B + f(1,0)][A + \bar{B} + f(1,1)]$  **2ª forma canónica**

Las formas canónicas son únicas.

Dos funciones son iguales si tienen la misma forma canónica

Primera forma



+ de ·

Segunda forma



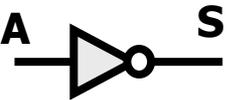
· de +

En cada prod. o suma  
intervienen todas las variables,  
Complementada o sin complementar



# Puertas lógicas

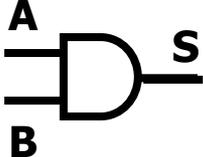
**INVERSOR**



| A | S |
|---|---|
| 0 | 1 |
| 1 | 0 |

7404 (6x1input)

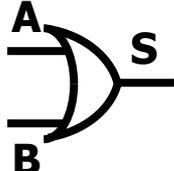
**AND**



| A | B | S |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

7408 (4x2inputs)

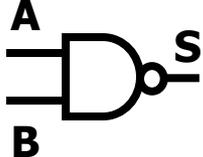
**OR**



| A | B | S |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

7432 (4x2inputs)

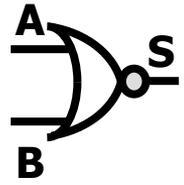
**NAND**



| A | B | S |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

7400 (4x2inputs)

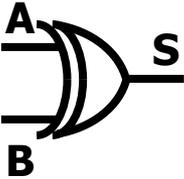
**NOR**



| A | B | S |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

7402 (4x2inputs)

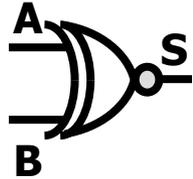
**XOR**



| A | B | S |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

7486 (4x2inputs)

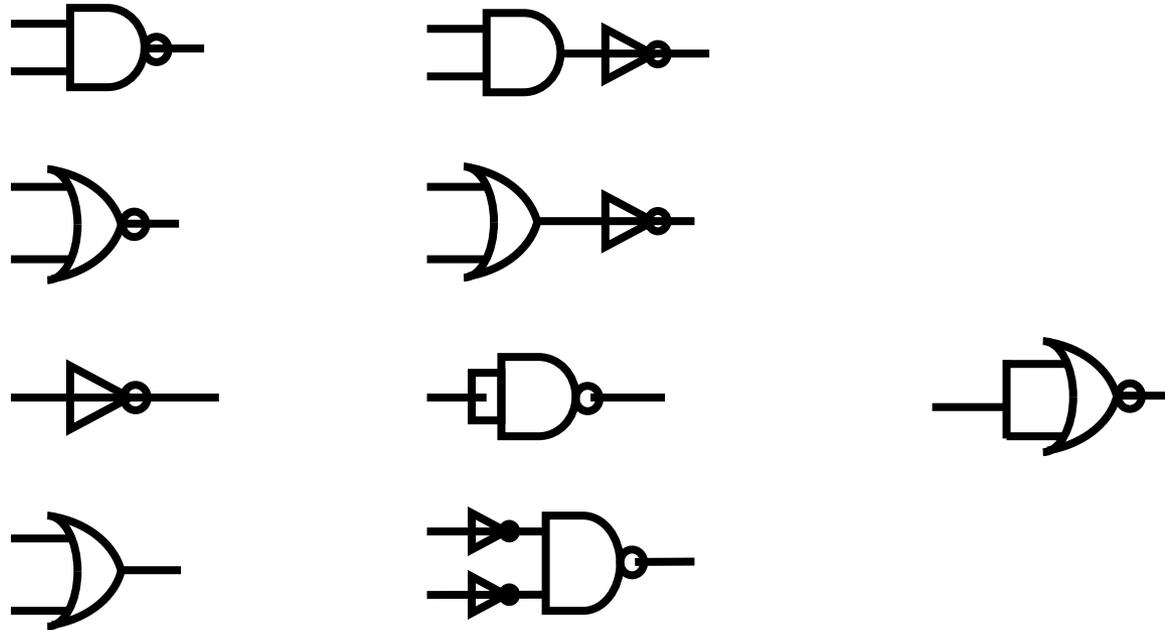
**XNOR**



| A | B | S |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

74266 (4x2inputs)

# Equivalencia entre puertas

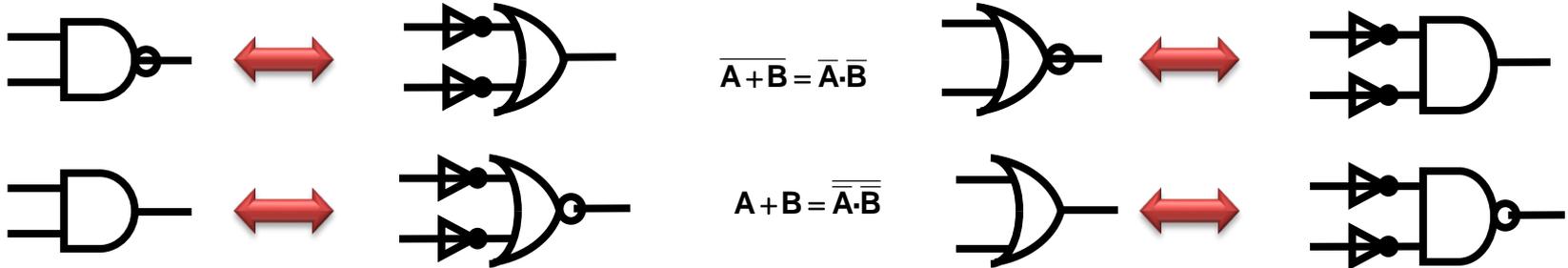


De Morgan



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

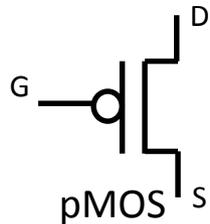
$$A \cdot B = \overline{\overline{A} + \overline{B}}$$



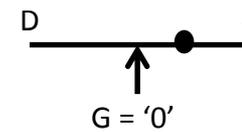
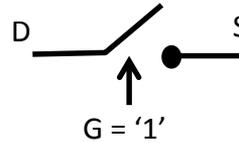
$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

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

# La lógica del transistor (tecnología CMOS)

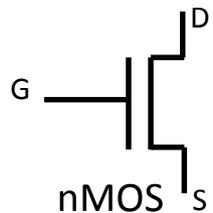


Se activa con un '0' en la G  
Conduce "bien" los '1s'

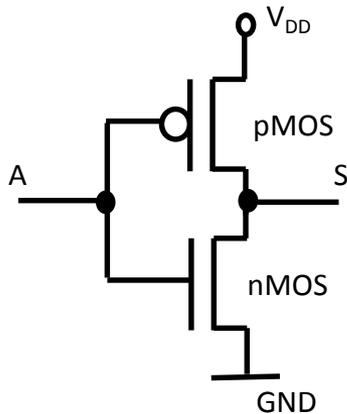
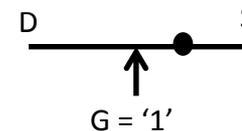
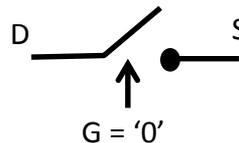


'1' =  $V_{DD}$

'0' = GND



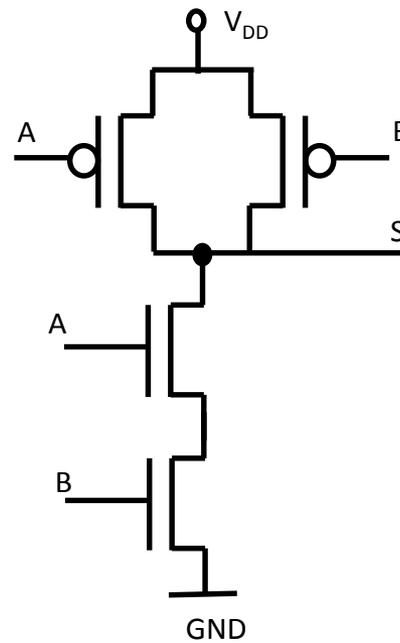
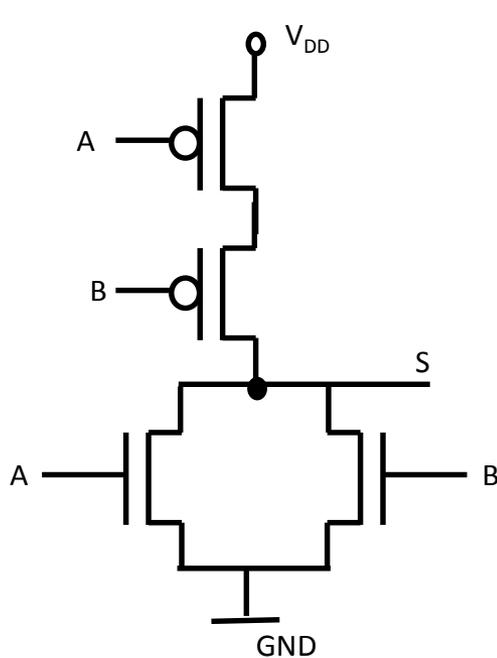
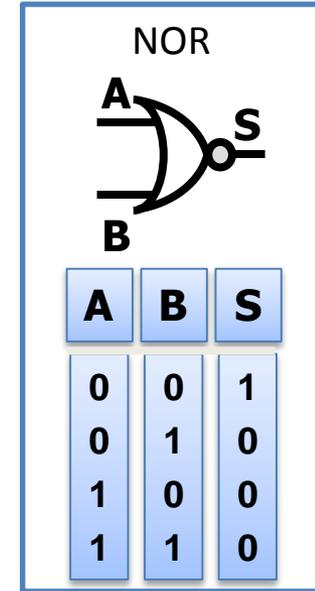
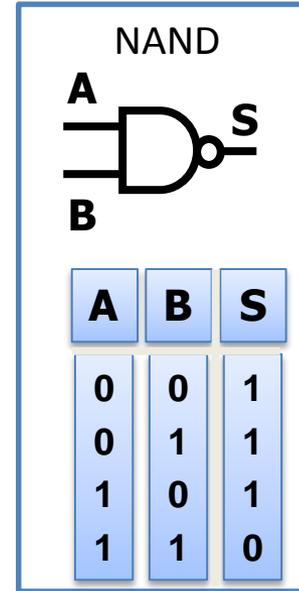
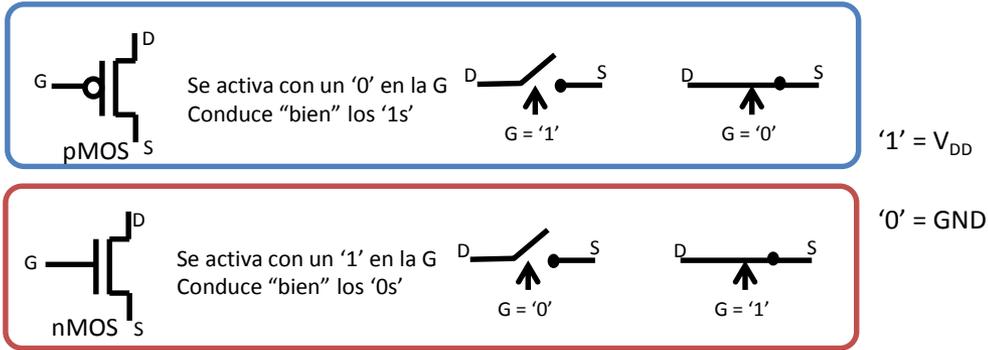
Se activa con un '1' en la G  
Conduce "bien" los '0s'



Si  $A = '1'$  ( $A = V_{DD}$ )  $\rightarrow$  nMOS conduce, pMOS no conduce  $\rightarrow S = '0'$  ( $S = GND$ )

Si  $A = '0'$  ( $A = GND$ )  $\rightarrow$  nMOS no conduce, pMOS conduce  $\rightarrow S = '1'$  ( $S = V_{DD}$ )

# La lógica del transistor (tecnología CMOS)



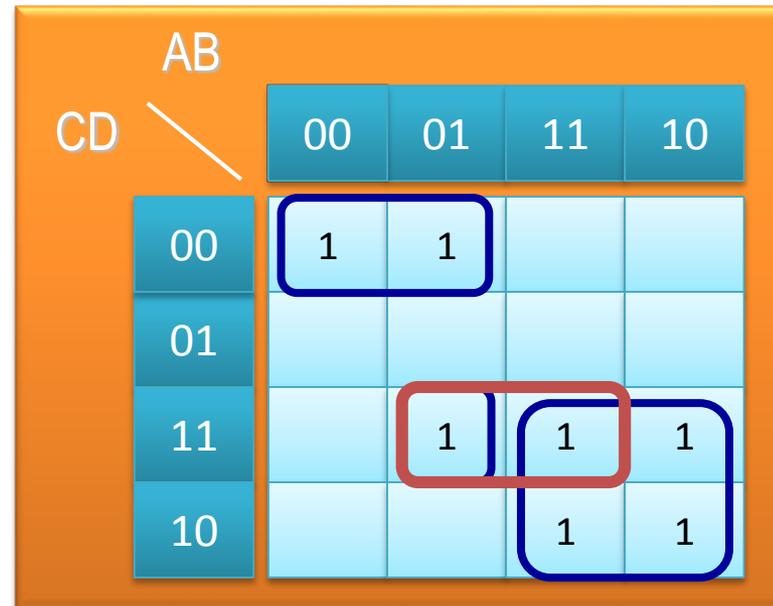
Si quiero hacer una AND tendré que hacer una NAND+INV

Si quiero hacer una OR tendré que hacer una NOR +INV

# Minimización de funciones lógicas

| A | B | C | D | f |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

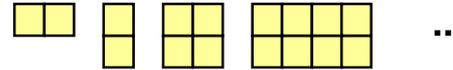
$$\begin{aligned}
 f &= \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD = \\
 &= \overline{A}\overline{C}\overline{D}(B + \overline{B}) + \overline{A}\overline{B}C(D + \overline{D}) + \overline{A}BC(D + \overline{D}) + \overline{A}\overline{B}CD = \\
 &= \overline{A}\overline{C}\overline{D} + \overline{A}\overline{B}C + \overline{A}BC + \overline{A}\overline{B}CD = \\
 &= \overline{A}\overline{C}\overline{D} + AC + \overline{A}\overline{B}CD
 \end{aligned}$$



# Minimización de funciones lógicas

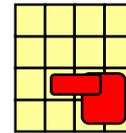
## ¿Qué hemos hecho?

Aplicar la distributiva → Celdas contiguas



T. De idempotencia →

Repetir un producto más de una vez



## ¿Qué hay que hacer, en general?

Agrupar todos los 1s en grupos lo más grandes posibles, sin importar solapes entre ellos.

Se deben considerar las adyacencias de los bordes.

Cada agrupación es un producto.

Las XOR no se deducen, aunque suelen ser



# Minimización de funciones lógicas: ejemplo

Para la siguiente función lógica

$$F = \bar{A} \cdot \bar{B} \cdot \bar{D} + A \cdot B \cdot \bar{C} \cdot \bar{D} + \bar{A} \cdot \bar{B} \cdot C \cdot D + A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D} + A \cdot C \cdot D + A \cdot \bar{B} \cdot C \cdot \bar{D}$$

- \* Obtener la tabla de verdad
- \* Obtener la expresión mínima de la función

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

| AB \ CD | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      | 1  | 0  | 1  | 1  |
| 01      | 0  | 0  | 0  | 0  |
| 11      | 1  | 0  | 1  | 1  |
| 10      | 1  | 0  | 0  | 1  |

$$F = \bar{B} \cdot \bar{D} + \bar{B} \cdot C + A \cdot \bar{C} \cdot \bar{D} + A \cdot C \cdot D$$

$$F = \bar{B} \cdot \bar{D} + \bar{B} \cdot C + A \cdot (\bar{C} \cdot \bar{D} + C \cdot D)$$

$$F = \bar{B} \cdot \bar{D} + \bar{B} \cdot C + A \cdot (\overline{C \oplus D})$$

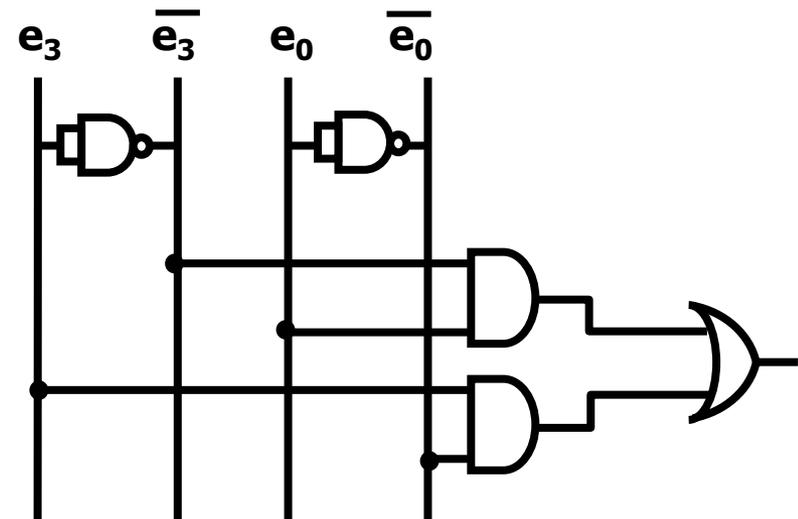
# Minimización de funciones lógicas: ejemplo

Diseñar un circuito con puertas NAND que determine si el mes del año, codificado en binario natural con 4 bits, tiene 31 días (salida a valor 1) o menos de 31 día (salida a 0).

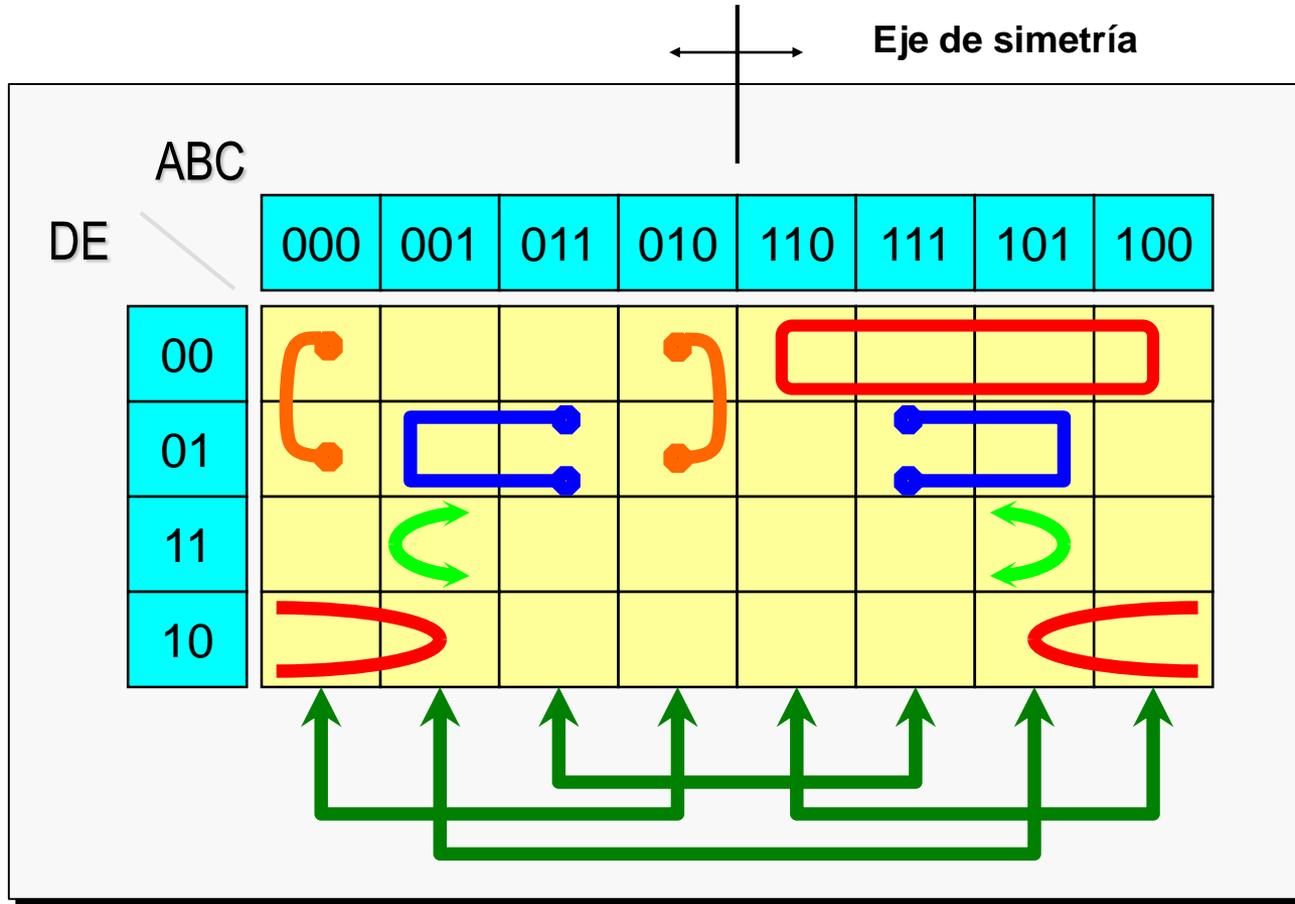
|   | $e_3$ | $e_2$ | $e_1$ | $e_0$ | $f$ |
|---|-------|-------|-------|-------|-----|
| X | 0     | 0     | 0     | 0     | X   |
| E | 0     | 0     | 0     | 1     | 1   |
| F | 0     | 0     | 1     | 0     | 0   |
| M | 0     | 0     | 1     | 1     | 1   |
| A | 0     | 1     | 0     | 0     | 0   |
| M | 0     | 1     | 0     | 1     | 1   |
| J | 0     | 1     | 1     | 0     | 0   |
| J | 0     | 1     | 1     | 1     | 1   |
| A | 1     | 0     | 0     | 0     | 1   |
| S | 1     | 0     | 0     | 1     | 0   |
| O | 1     | 0     | 1     | 0     | 1   |
| N | 1     | 0     | 1     | 1     | 0   |
| D | 1     | 1     | 0     | 0     | 1   |
| X | 1     | 1     | 0     | 1     | X   |
| X | 1     | 1     | 1     | 0     | X   |
| X | 1     | 1     | 1     | 1     | X   |

| $e_3e_2$    | 00 | 01 | 11 | 10 |
|-------------|----|----|----|----|
| $e_1e_0$ 00 | X  | 0  | 1  | 1  |
| 01          | 1  | 1  | X  | 0  |
| 11          | 1  | 1  | X  | 0  |
| 10          | 0  | 0  | X  | 1  |

$$f = \bar{e}_3 e_0 + e_3 \bar{e}_0 = e_3 \oplus e_0$$



# Minimización de funciones lógicas: para 5 variables

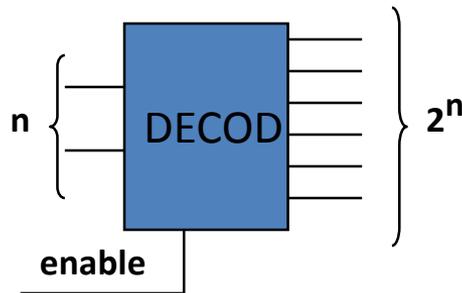


# TEMA 3

## Circuitos combinatoriales

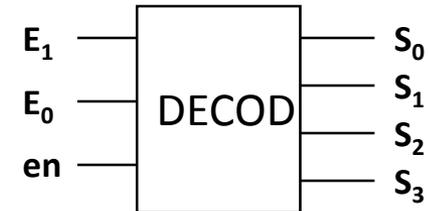
# Decodificadores y codificadores

## Decodificador



Se activa la salida correspondiente al número binario codificado en la entrada

*Ejemplo:*  
Decod 2 entradas  
con enable



| en | E <sub>1</sub> | E <sub>0</sub> | S <sub>0</sub> | S <sub>1</sub> | S <sub>2</sub> | S <sub>3</sub> |
|----|----------------|----------------|----------------|----------------|----------------|----------------|
| 0  | x              | x              | 0              | 0              | 0              | 0              |
| 1  | 0              | 0              | 1              | 0              | 0              | 0              |
| 1  | 0              | 1              | 0              | 1              | 0              | 0              |
| 1  | 1              | 0              | 0              | 0              | 1              | 0              |
| 1  | 1              | 1              | 0              | 0              | 0              | 1              |

## Funciones lógicas

$$S_0 = e_n \cdot \bar{E}_1 \bar{E}_0$$

$$S_1 = e_n \cdot \bar{E}_1 E_0$$

$$S_2 = e_n \cdot E_1 \bar{E}_0$$

$$S_3 = e_n \cdot E_1 E_0$$

## Comerciales

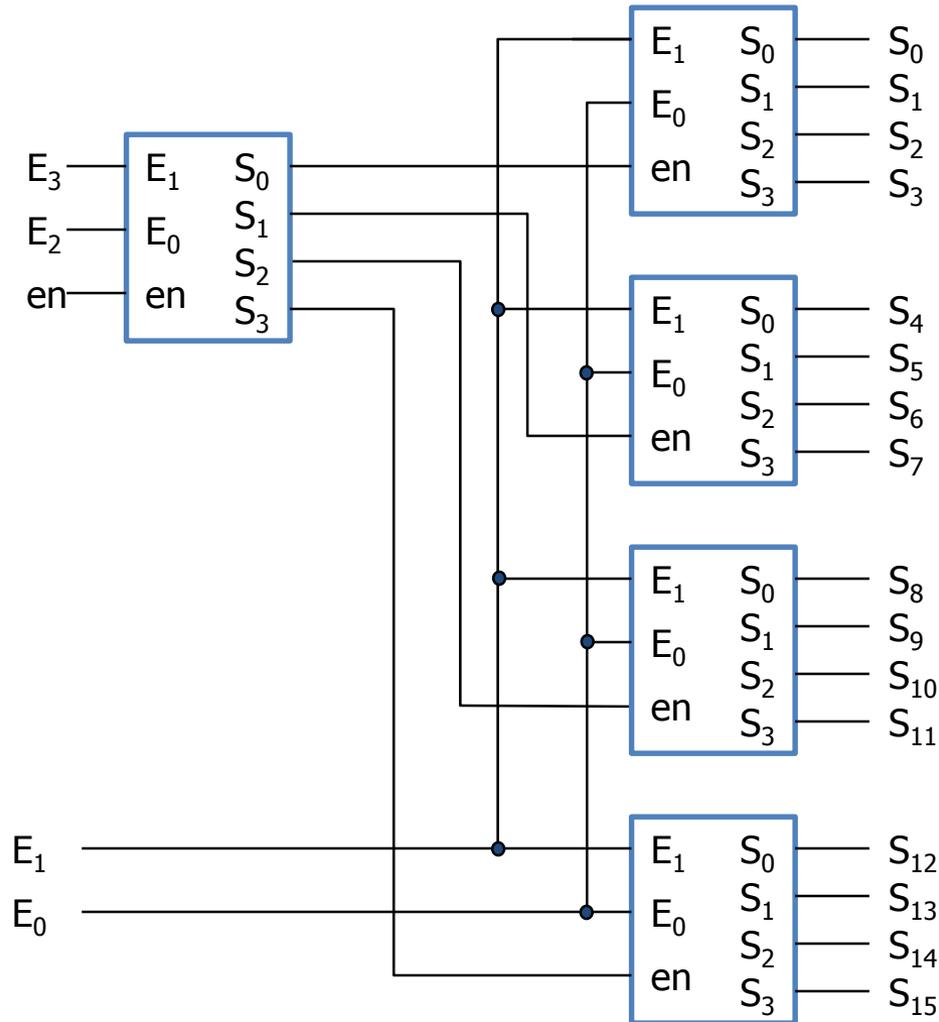
74154 (4 a 16) lógica negativa

74238 (3 a 8) lógica positiva

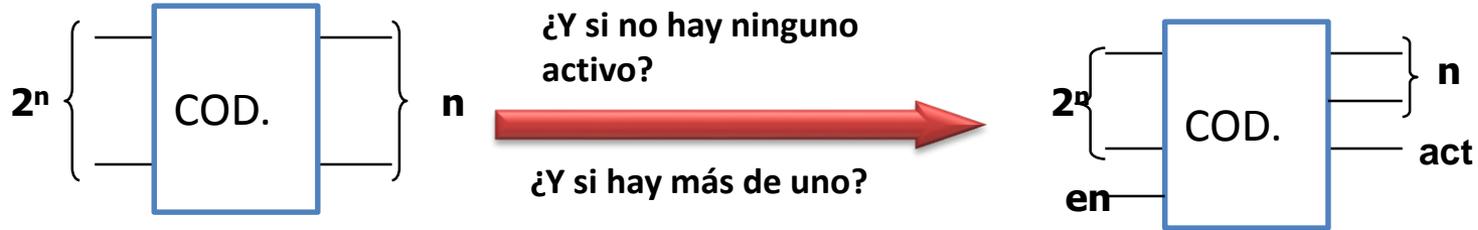
74138 (3 a 8) lógica negativa

# Ejemplo

A partir de decodificadores de 2 entradas, construir un decodificador de 4 entradas



# Codificador



Se codifica en binario sobre la salida el número de entrada que esté activa

## Codificador prioritario al más alto

| en | E <sub>0</sub> | E <sub>1</sub> | E <sub>2</sub> | E <sub>3</sub> | A <sub>1</sub> | A <sub>0</sub> | act |               |
|----|----------------|----------------|----------------|----------------|----------------|----------------|-----|---------------|
| 0  | x              | x              | x              | x              | 0              | 0              | 0   | deshabilitado |
| 1  | 0              | 0              | 0              | 0              | 0              | 0              | 0   | inactivo      |
| 1  | x              | x              | x              | 1              | 1              | 1              | 1   | activo        |
| 1  | x              | x              | 1              | 0              | 1              | 0              | 1   |               |
| 1  | x              | 1              | 0              | 0              | 0              | 1              | 1   |               |
| 1  | 1              | 0              | 0              | 0              | 0              | 0              | 1   |               |

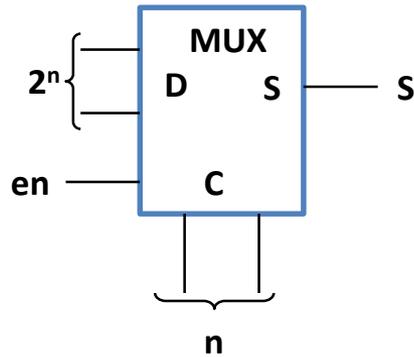
También existe el codificador prioritario al más bajo

## Comerciales

74148 (8 a 3),  
lógica negativa  
en la entrada

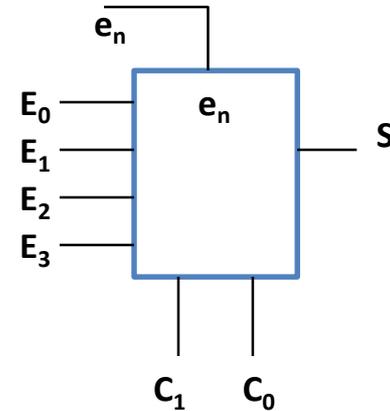
# Multiplexores y demultiplexores

## Multiplexor (MUX)

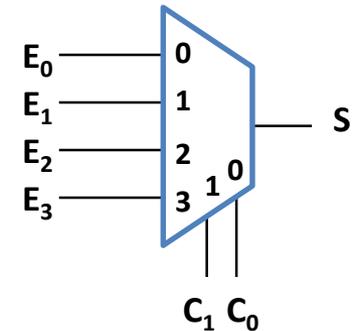


La entrada de datos correspondiente al número codificado en binario en las señales de control se conecta a la salida

| en | E <sub>0</sub> | E <sub>1</sub> | E <sub>2</sub> | E <sub>3</sub> | C <sub>1</sub> | C <sub>0</sub> | S |
|----|----------------|----------------|----------------|----------------|----------------|----------------|---|
| 0  | X              | X              | X              | X              | X              | X              | 0 |
| 1  | D              | X              | X              | X              | 0              | 0              | D |
| 1  | X              | D              | X              | X              | 0              | 1              | D |
| 1  | X              | X              | D              | X              | 1              | 0              | D |
| 1  | X              | X              | X              | D              | 1              | 1              | D |



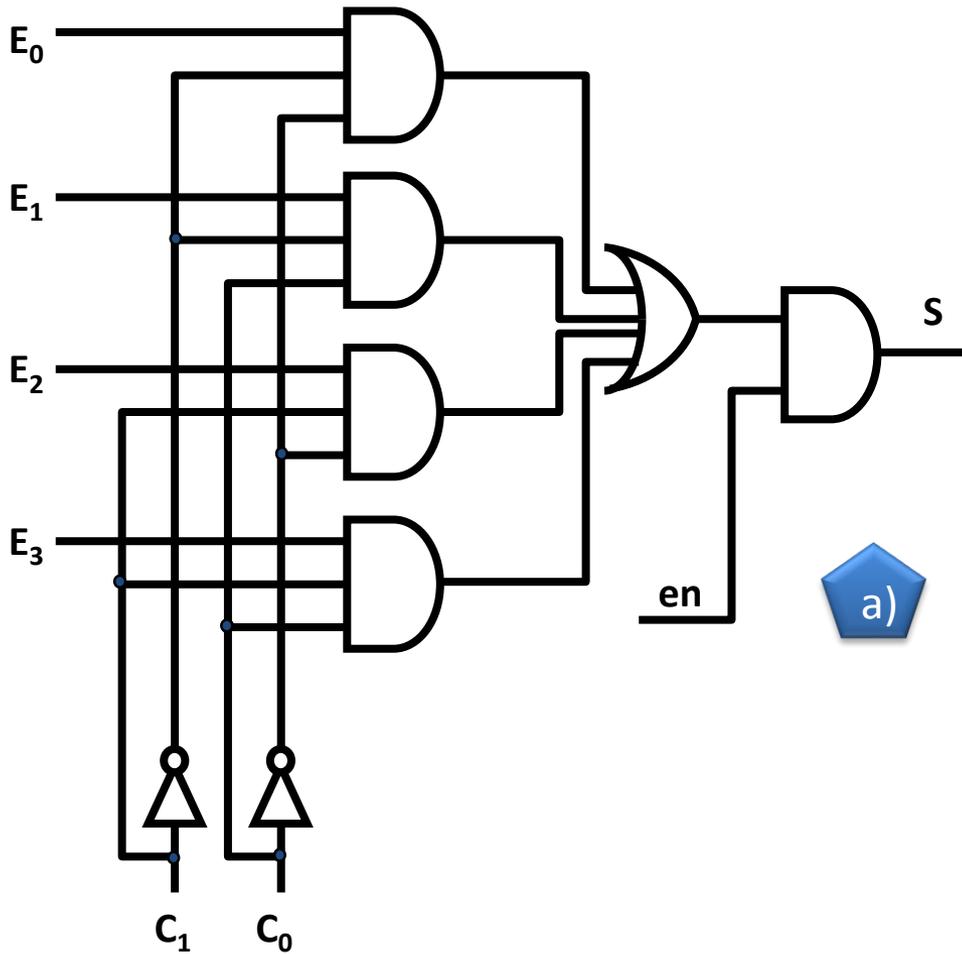
Símbolo propio:



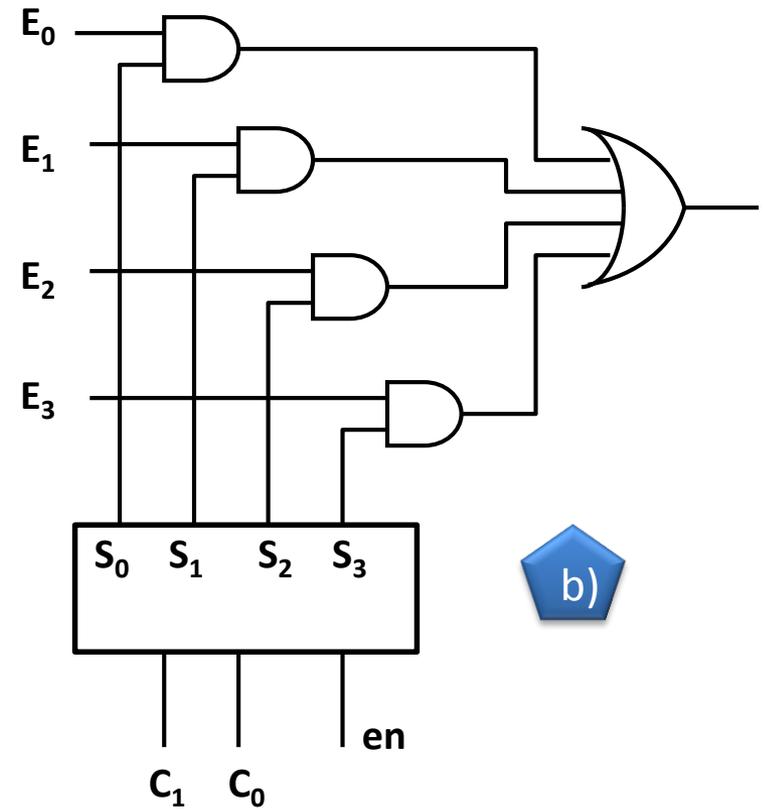
## Comerciales

74157    4x 2 a 1  
 74152    8 a 1  
 74293    2x 4 a 1

# MUX mediante puertas lógicas



a)

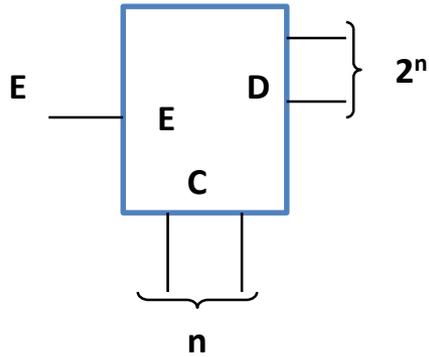


b)

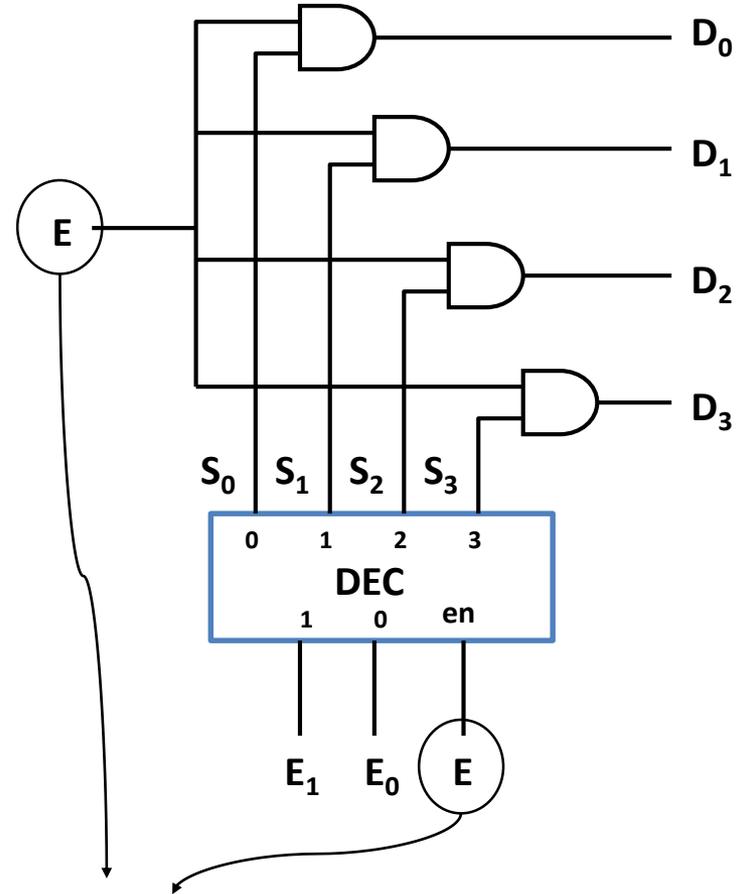
# Construcción de multiplexores grandes

A partir de mux de 2 entradas, construir uno de ocho entradas

# Demultiplexores



Saca la entrada por aquella salida correspondiente al número codificado en las señales de control



Sólo una de las dos opciones

# Funciones lógicas mediante decodificadores/mux

## Ejemplo

Diseñar un circuito que tiene como entrada el mes del año codificado en binario y como salida un '1' si el mes es de 31 días o un '0' si es de menos de 31 días

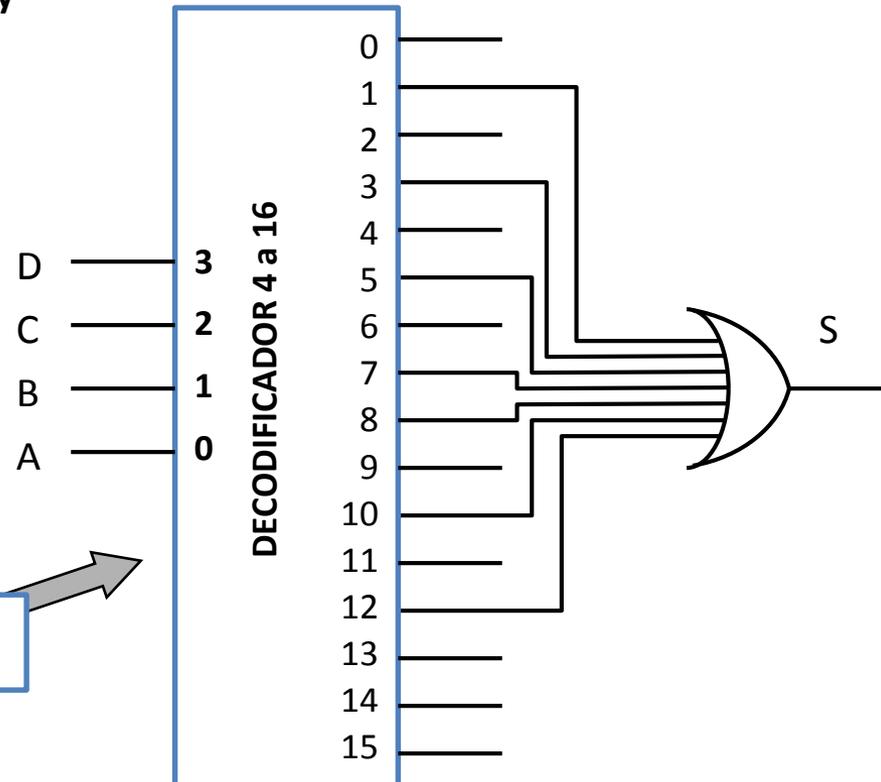
| D | C | B | A | S |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | x |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | x |
| 1 | 1 | 1 | 0 | x |
| 1 | 1 | 1 | 1 | x |

**A** Mediante un decodificador y una puerta OR

**B** Mediante un multiplexor e inversores

**C** Mediante multiplexores 2 a 1 e inversores

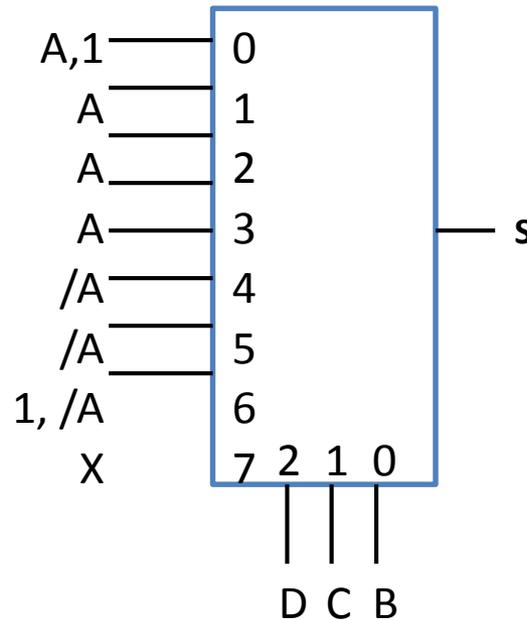
**A** Un decodificador



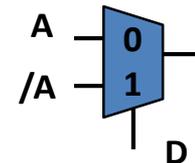
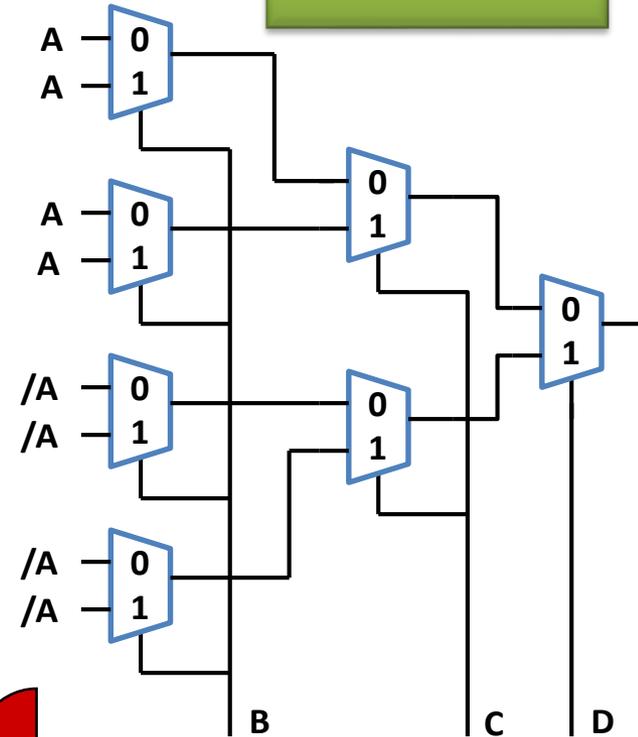
# Funciones lógicas mediante decodificadores/mux

**B**  
Un MUX

| D | C | B | A | S |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | x |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | x |
| 1 | 1 | 1 | 0 | x |
| 1 | 1 | 1 | 1 | x |



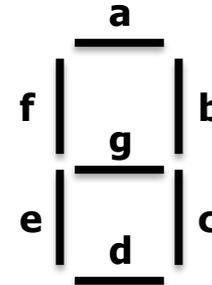
**C**  
MUX 2 a 1



# Convertidores de código

## A) Convertidor BCD a 7 segmentos

| E <sub>3</sub> | E <sub>2</sub> | E <sub>1</sub> | E <sub>0</sub> | a | b | c | d | e | f | g |
|----------------|----------------|----------------|----------------|---|---|---|---|---|---|---|
| 0              | 0              | 0              | 0              | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0              | 0              | 0              | 1              | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0              | 0              | 1              | 0              | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0              | 0              | 1              | 1              | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0              | 1              | 0              | 0              | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0              | 1              | 0              | 1              | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0              | 1              | 1              | 0              | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0              | 1              | 1              | 1              | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1              | 0              | 0              | 0              | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1              | 0              | 0              | 1              | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| resto          |                |                |                | x | x | x | x | x | x | x |

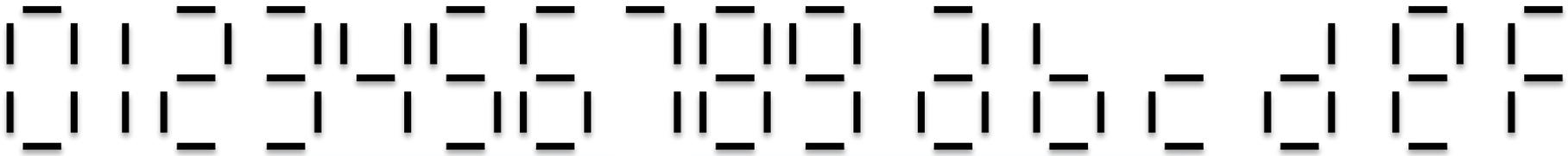


Comerciales

- 7447 (lógica neg.)
- 7448 (lógica pos.)

## B) Hexadecimal a 7 segmentos

- 7446 (lógica pos.)
- 7449 (lógica neg.)

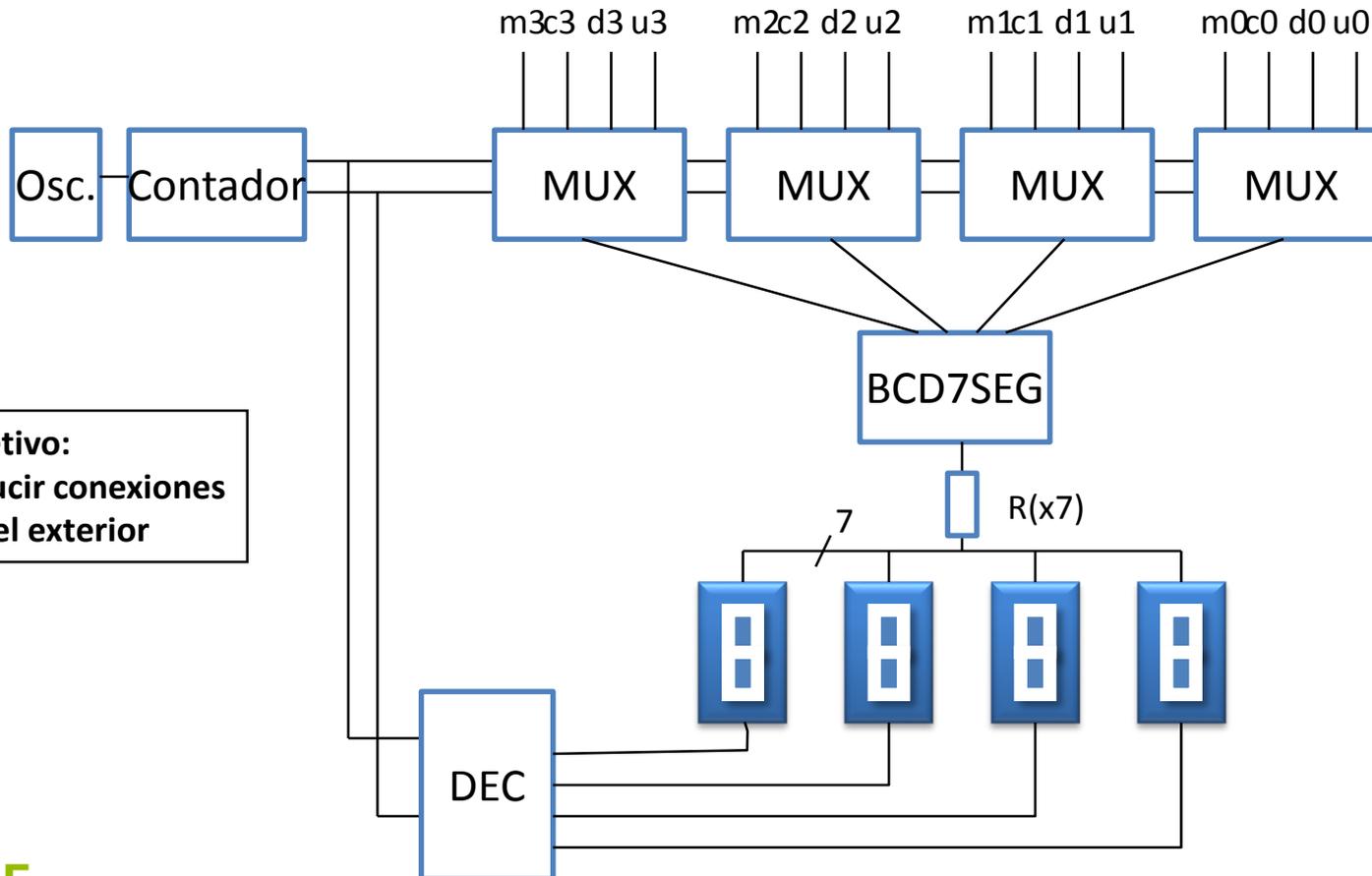


Más complejo

Mayor número de puertas

# Ejemplo: displays de 7 segmentos multiplexados

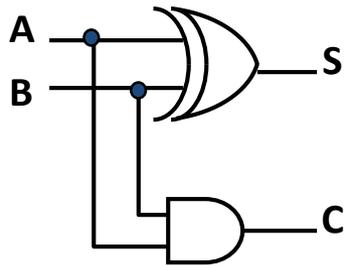
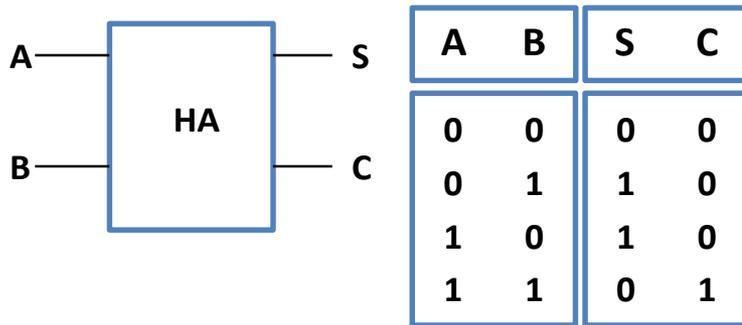
millares      centenas      decenas      unidades  
 m3 m2 m1 m0    c3 c2 c1 c0    d3 d2 d1 d0    u3 u2 u1 u0  
 ↓ ↓ ↓ ↓        ↓ ↓ ↓ ↓        ↓ ↓ ↓ ↓        ↓ ↓ ↓ ↓



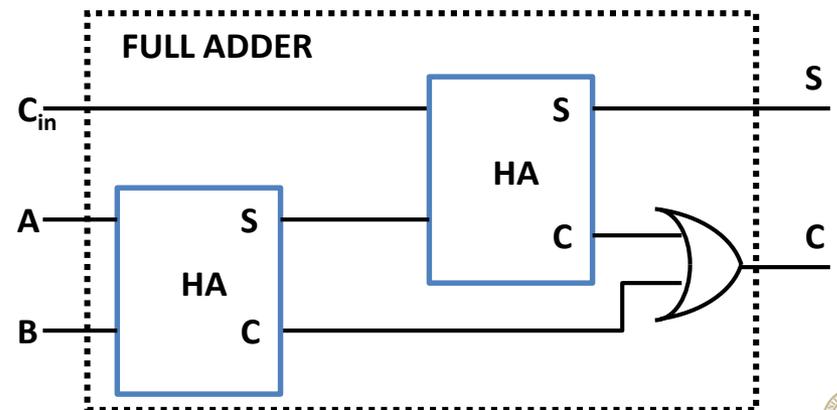
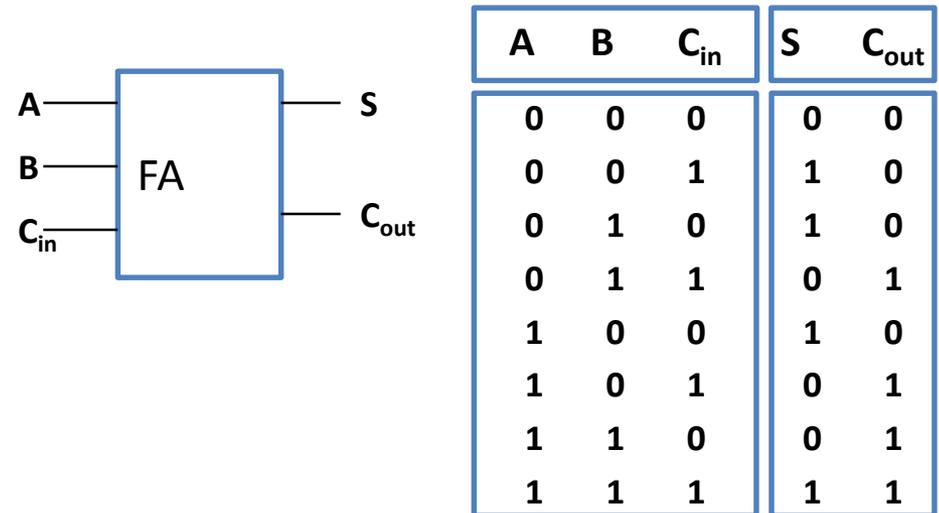
**Objetivo:**  
 Reducir conexiones  
 con el exterior

# Funciones aritméticas: sumador

## Semisumador binario (Half adder)

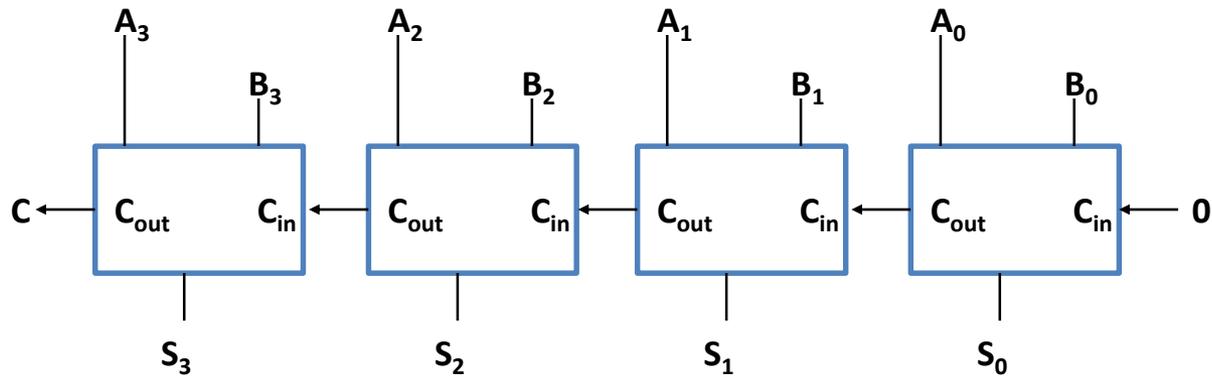


## Sumador binario (Full adder)



# Sumador serie

Suma de 2 números de 4 bits



## Características

- Número de puertas bajo
- Retardo proporcional al número de bits

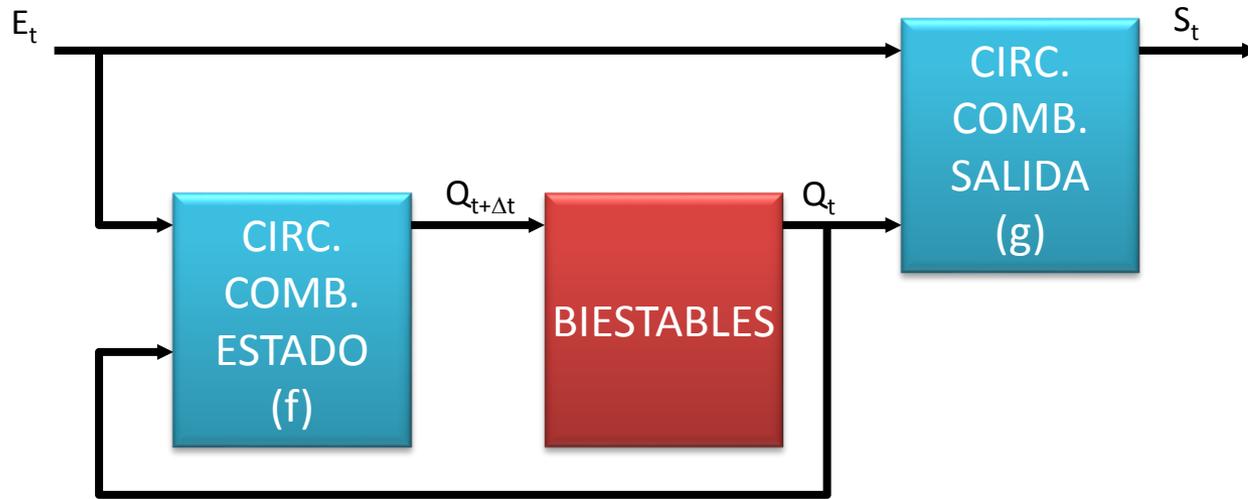
# TEMA 4

## Circuitos secuenciales – Biestables

# Introducción

Circuito secuencial es aquel en el que las salidas en un instante de tiempo dependen de las entradas en ese instante de tiempo y en instantes anteriores.

La evolución pasada está almacenada en unos elementos con capacidad de **memorizar** el estado interno. Cada bit de información de estado se guarda en un **biestable**.



$$Q_{t+\Delta t} = f(E_t, Q_t)$$

Ecuaciones de ESTADO

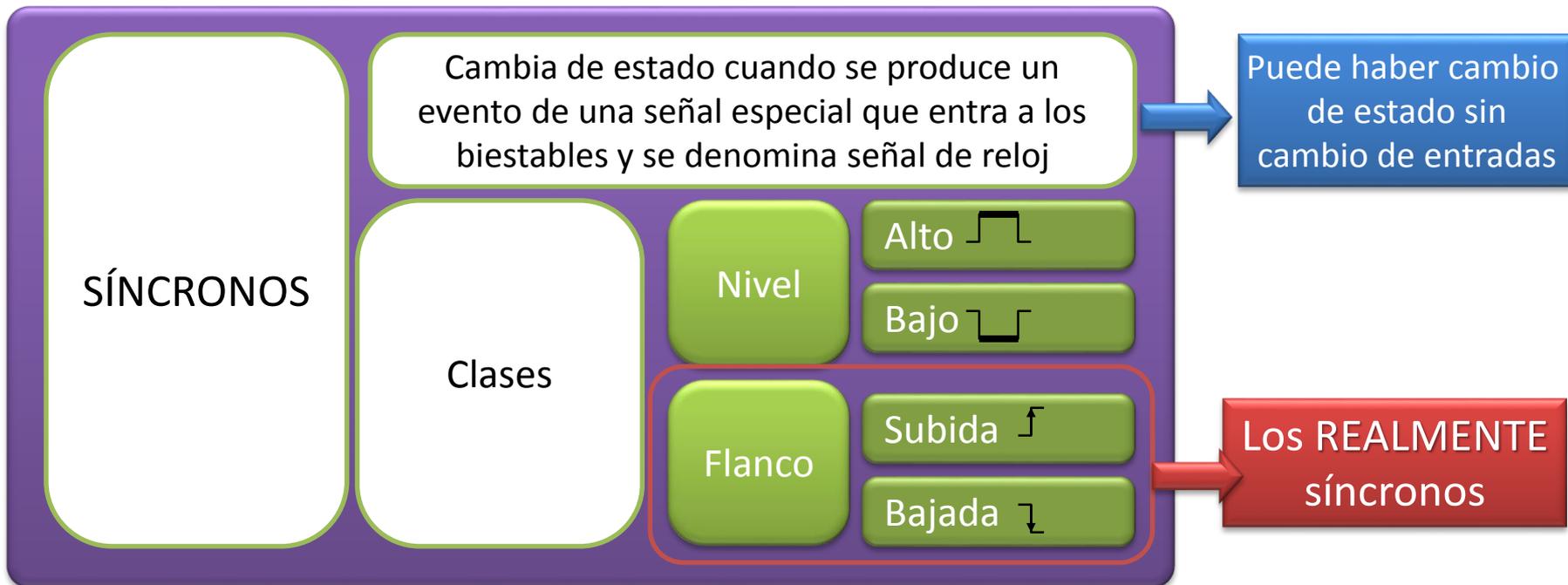
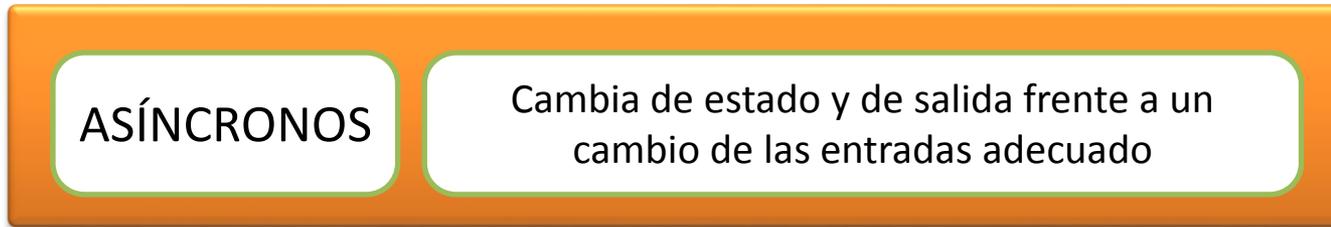
← EVOLUCIÓN TEMPORAL

$$S_t = g(E_t, Q_t)$$

Ecuaciones de SALIDA

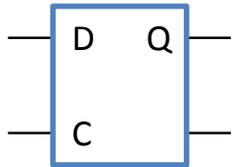
← VALORES INSTANTÁNEOS

# Circuitos asíncronos y síncronos

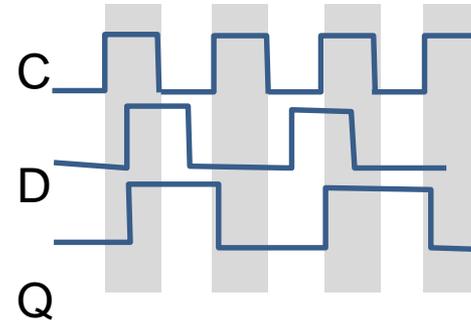


# Biestables asíncronos

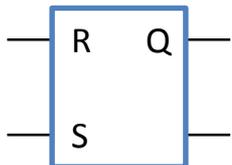
## Biastable latch (asíncr.)



| D | C            | $Q_{t+\Delta t}$ |                     |
|---|--------------|------------------|---------------------|
| X | 0            | $Q_t$            | Cerrado             |
| 0 | 1            | 0                | } Modo transparente |
| 1 | 1            | 1                |                     |
| 0 | $\downarrow$ | 0                | } Captura del dato  |
| 1 | $\downarrow$ | 1                |                     |



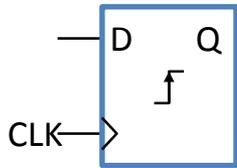
## Biastable RS



| R | S | $Q_{t+\Delta t}$ |         |
|---|---|------------------|---------|
| 0 | 0 | $Q_t$            |         |
| 0 | 1 | 1                | reset   |
| 1 | 0 | 0                | set     |
| 1 | 1 | 0/1              | depende |

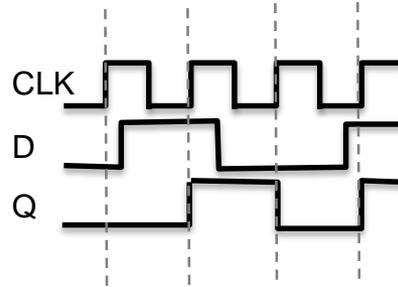
# Biestables síncronos

## Biestable D (*dato*)

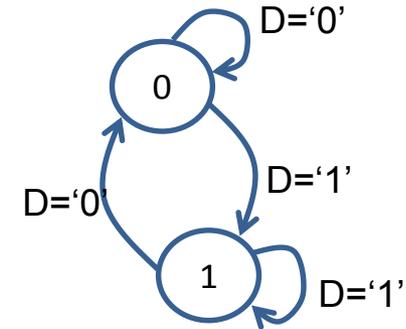


| D | CLK | Qt+Δt |
|---|-----|-------|
| 0 | f   | 0     |
| 1 | f   | 1     |

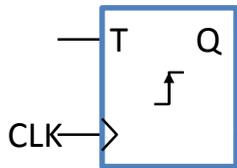
Puede ser activo por flanco  $\uparrow$  de subida  $\downarrow$  o de bajada



La entrada D se muestrea en el flanco activo de CLK  
Q sólo puede cambiar en ese flanco

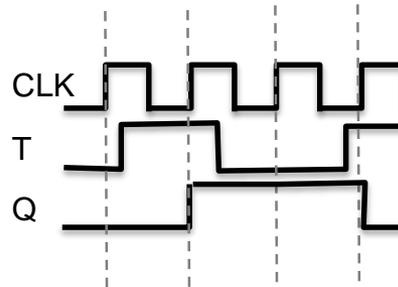


## Biestable T (*toggle*)

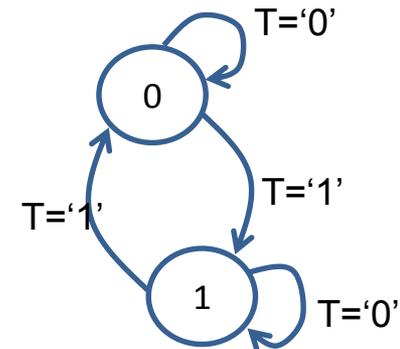


| T | CLK | Qt+Δt |
|---|-----|-------|
| 0 | f   | Qt    |
| 1 | f   | /Qt   |

Puede ser activo por flanco  $\uparrow$  de subida  $\downarrow$  o de bajada

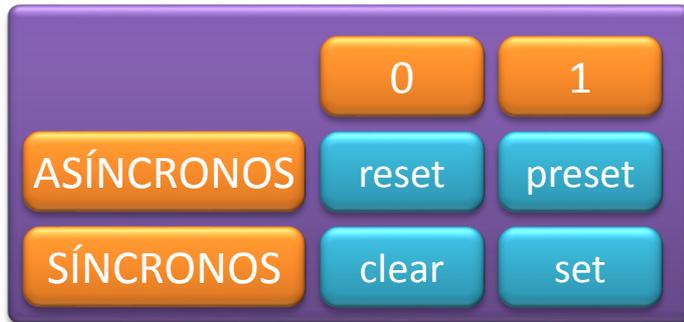


La entrada T se muestrea en el flanco activo de CLK  
Q sólo puede cambiar en ese flanco



# Otras señales de los biestables

## Inicialización de biestables

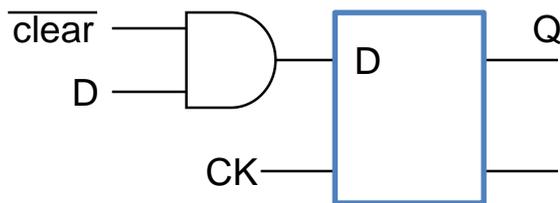


No hay acuerdo en el uso de esta terminología

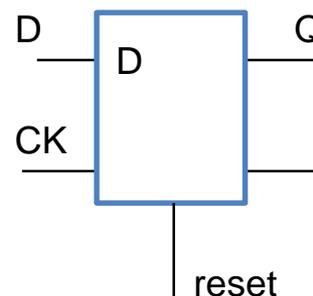
La inic. asíncrona es inmediata, mientras que la síncrona se espera al primer flanco activo de reloj

Las señales de inicialización suelen ser activas por nivel bajo (la acción se produce cuando la señal es 0)

## Ejemplo: Biestable D con clear



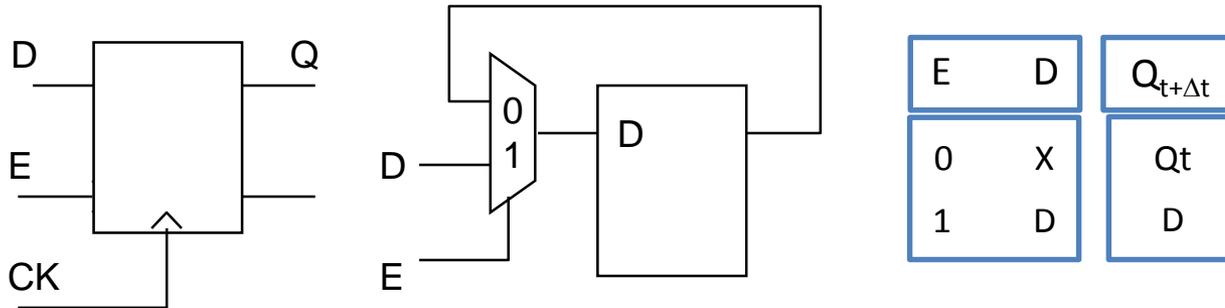
## Ejemplo: Biestable D con reset



La inic. síncrona se puede considerar como parte de la funcionalidad

# Otras señales de los biestables

## Señal de enable (carga) en biestables D



No confundir este enable con el combinacional

Ejemplo: A partir de un biestable D, construir un T con clear, reset y carga

Orden: reset > clear > carga > T

# Características temporales de los biestables

## A Restricciones biestable

→ Duración del reloj

$t_{1,\min}$

$t_{0,\min}$

frec. máxima:  $f_{\max} = 1 / (t_{1,\min} + t_{0,\min})$

→ Duración de las señales de inicialización asíncronas

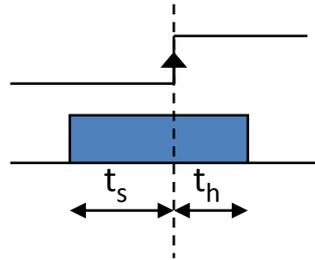
$t_{\text{reset},\min}$

$t_{\text{preset},\min}$

→ Tiempos de inserción de las señales de dato, toggle, etc.

$t_{\text{setup}}$

$t_{\text{hold}}$



## B Retardos

→ Retardos de propagación típicos

$t_{\text{CK},Q}$

$t_{\text{preset},Q}$

## C Restricciones circuitos

→ Compatibilidad propia

$$t_{\text{CK},Q} > t_{\text{hold}}$$

→ frec. máxima circuito

$$f_{\max} = \frac{1}{(t_{\text{CK},Q} + t_{\text{crítico}} + t_{\text{setup}})}$$

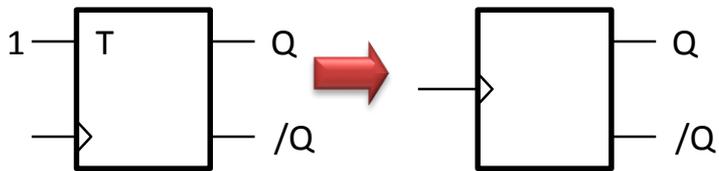
# TEMA 5

## Circuitos secuenciales – Aplicaciones

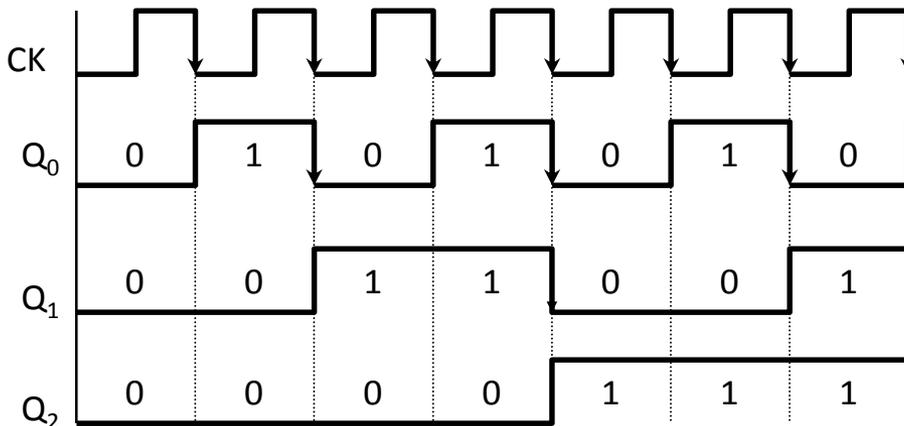
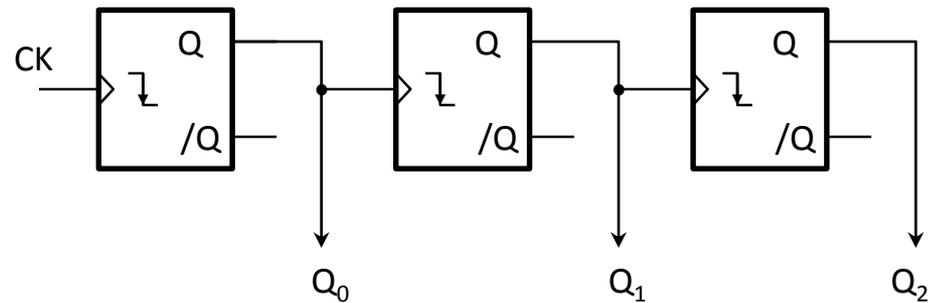
# Contadores

## Contador asíncrono de n bits – Divisor de frecuencia

Notación:



Contador de 3 bits

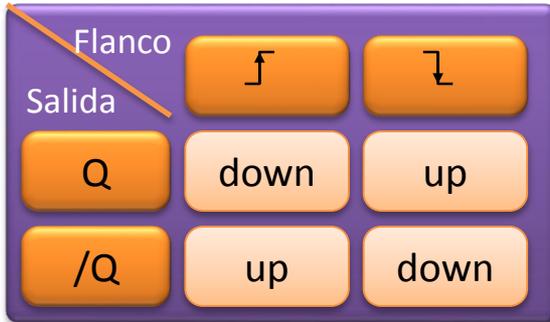


Aunque los biestables son síncronos, el reloj no es común a todos ellos

$$\text{frec}(Q_2) = \frac{\text{frec}(CK)}{2^3}$$

# Contadores

Contadores ascendentes y descendentes



¿Cómo hacer un contador up-down?

No sirve multiplexar las Qs o las /Qs, ya que habría 'saltos' en la cuenta.

**Solución:**

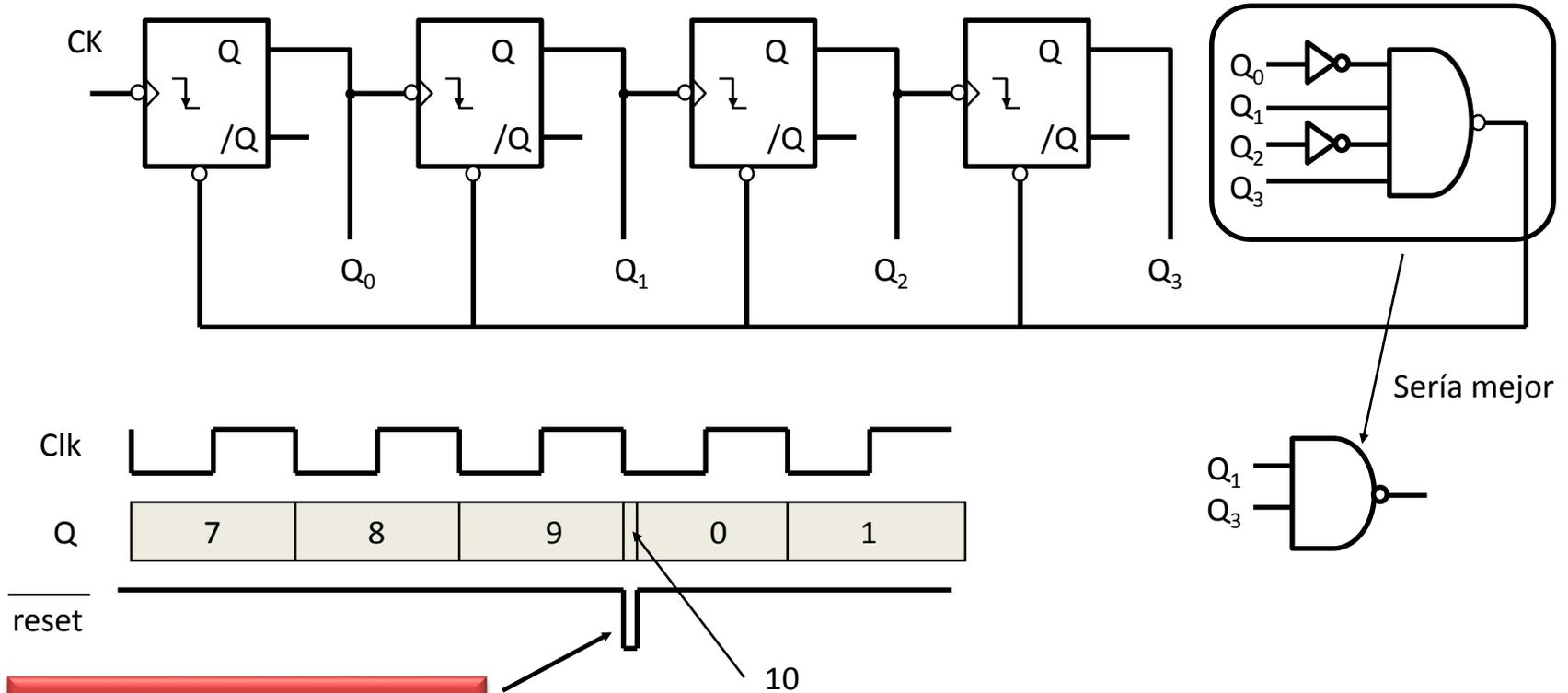
Cambiar el flanco activo

A diagram of an inverter gate (NOT gate) with a red arrow pointing to its input from the left and its output to the right.

|     |     |
|-----|-----|
| 000 | 111 |
| 001 | 110 |
| 010 | 101 |
| 011 | 100 |
| 100 | 011 |
| 101 | 010 |
| 110 | 001 |
| 111 | 000 |

# Contadores

Contadores de 0 a  $n = 2^m$ . Ejemplo: Contador de 0 a 9



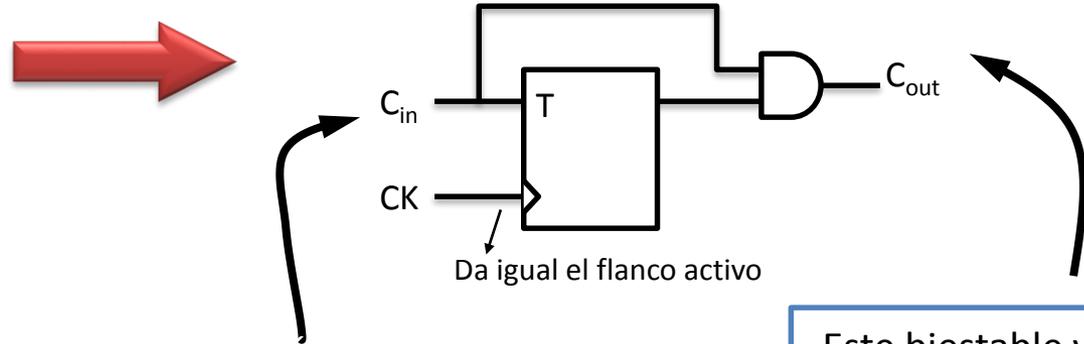
**POSIBLES PROBLEMAS !!**

Se debe codificar el valor siguiente al último valor que se quiera contar.

# Contadores

Contadores síncronos

Contador síncrono de 1 bit



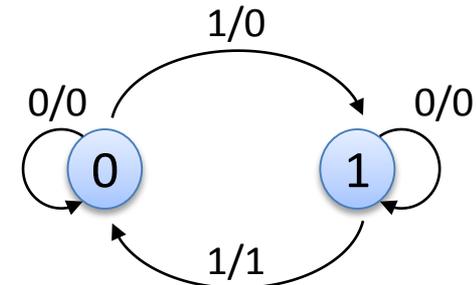
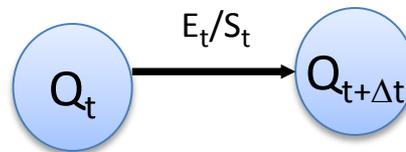
Los biestables de atrás son 1 → cambia

Este biestable y los anteriores son 1

Tabla de estados

| Cin | Qt | Cout | Qt+Δt |
|-----|----|------|-------|
| 0   | 0  | 0    | 0     |
| 0   | 1  | 0    | 1     |
| 1   | 0  | 0    | 1     |
| 1   | 1  | 1    | 0     |

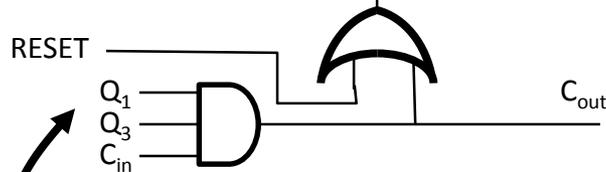
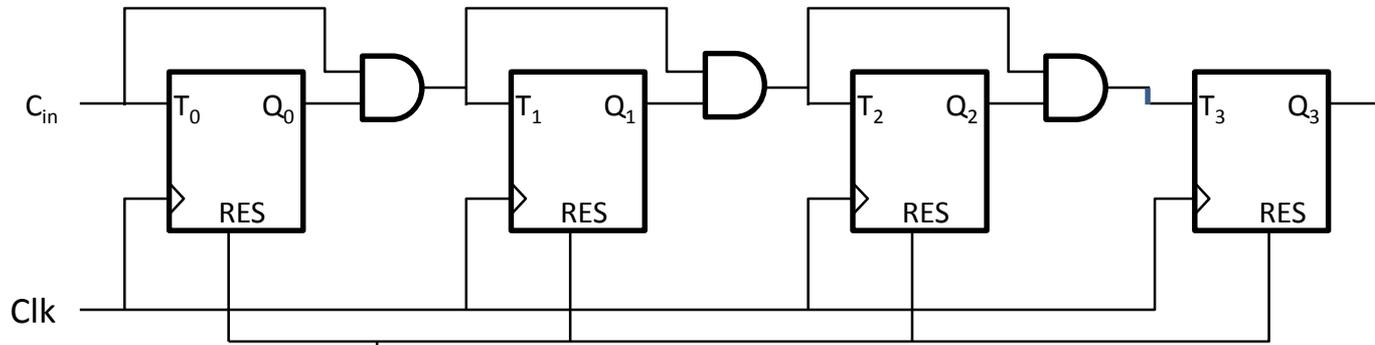
Diagrama de estados:  $C_{in}/C_{out}$



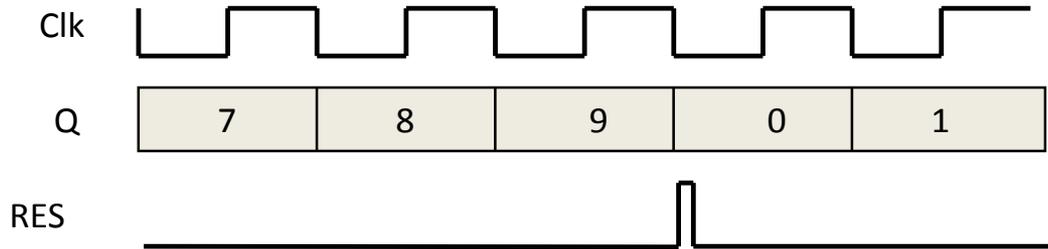


# Contadores

Ejemplo: Contador síncrono de 0 a 9

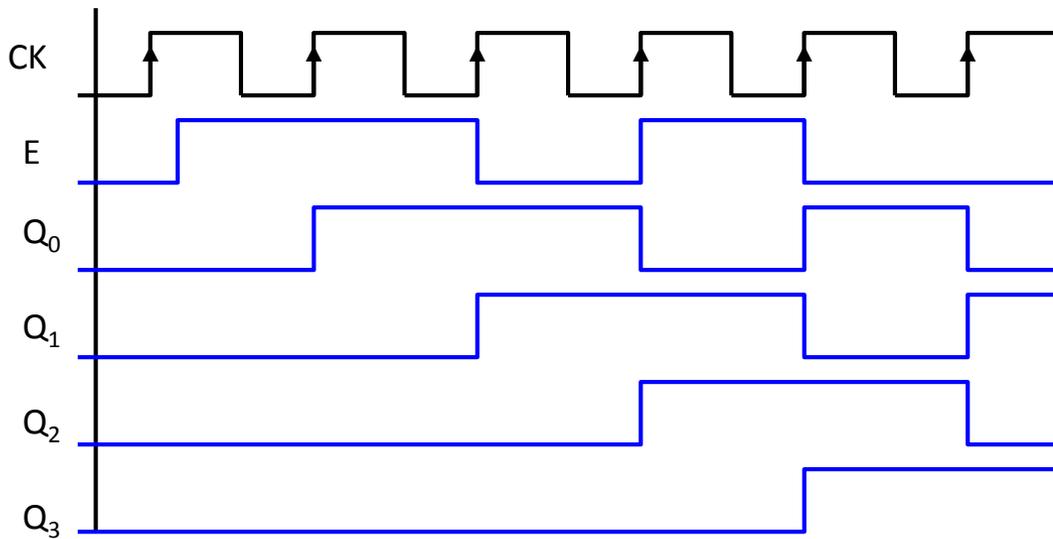
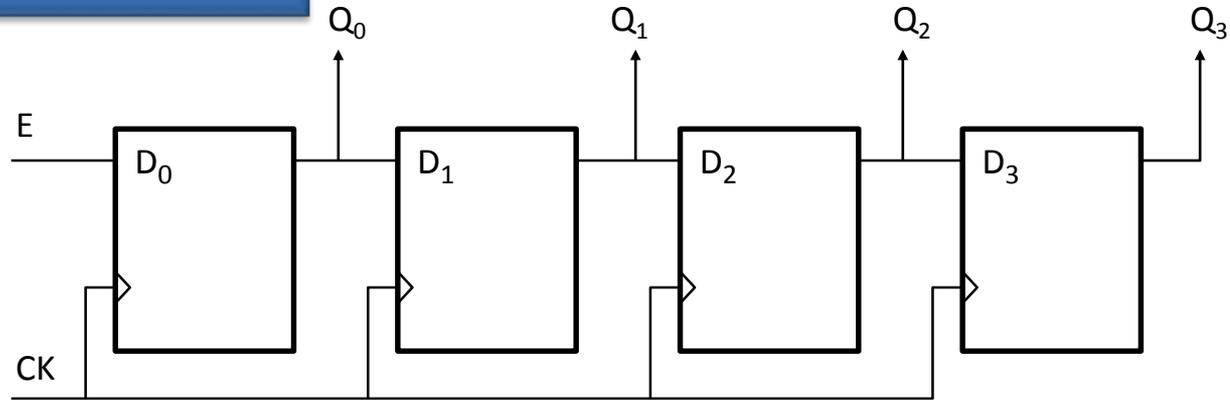


Se codifica el siguiente valor al que se quiere contar



# Registros

## Registros de desplazamiento

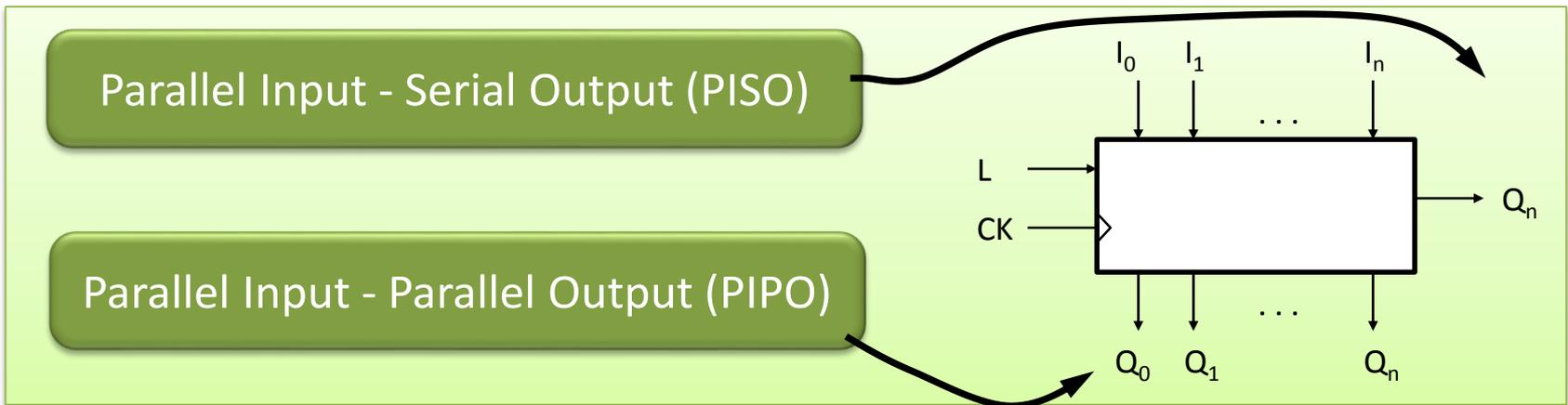
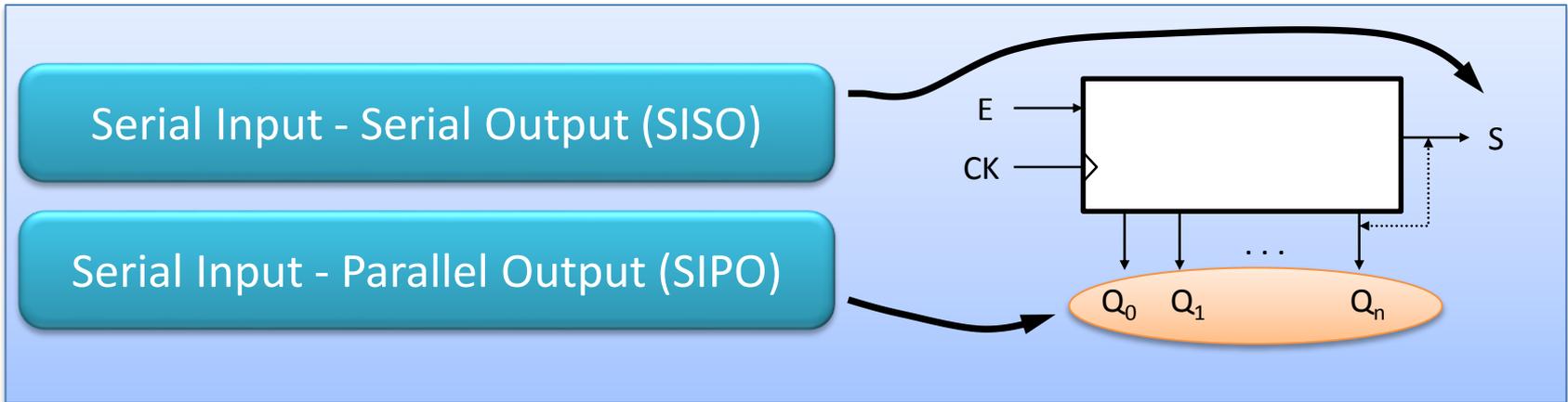


## Interpretación de salidas:



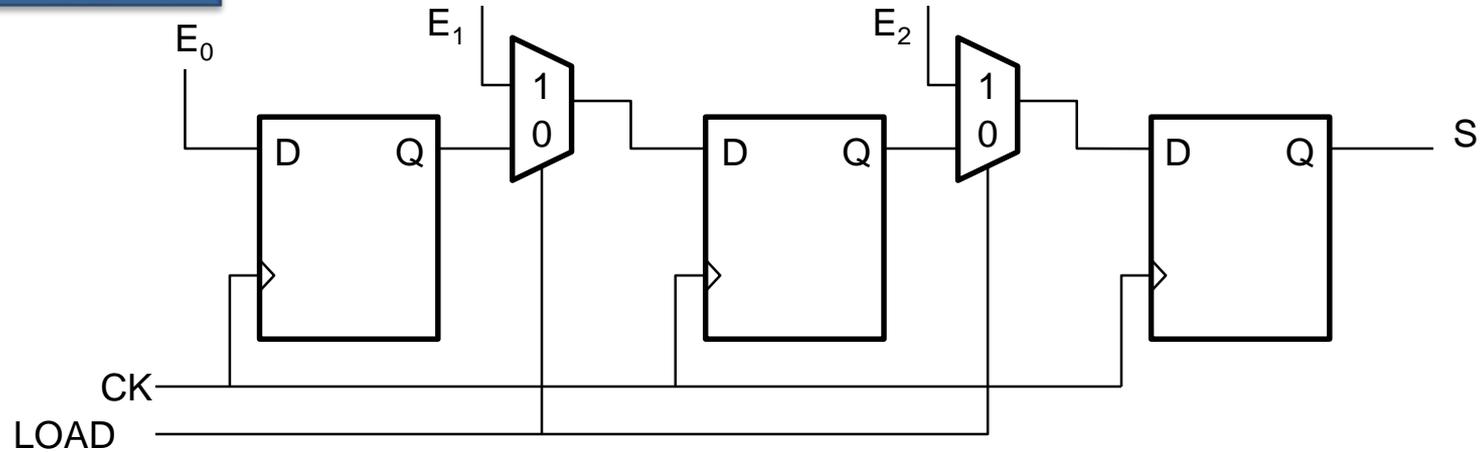
# Registros

## Tipos de registros



# Registros

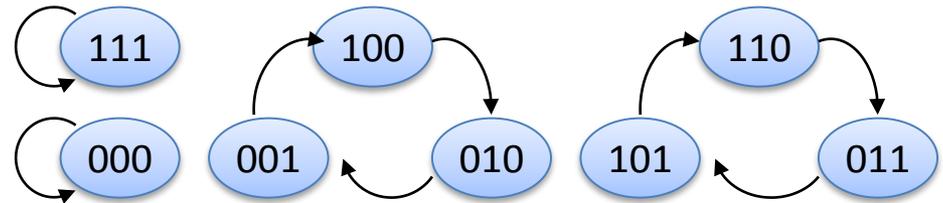
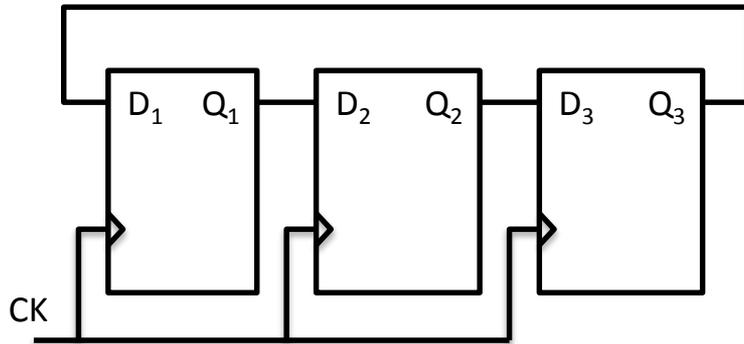
## Registro PISO



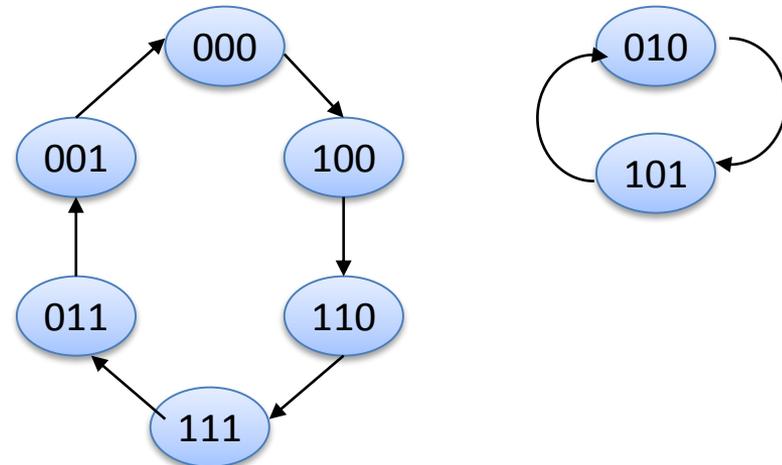
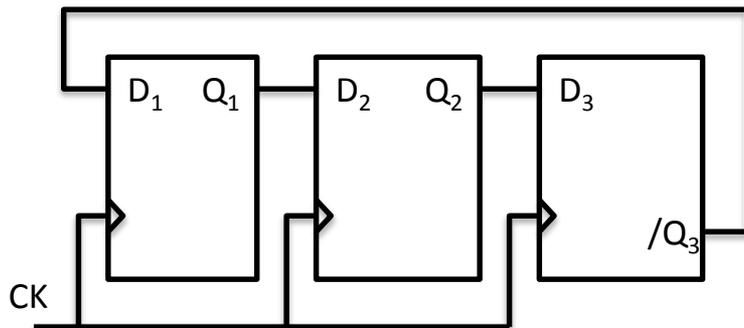
Si  $LOAD = '1'$ , carga en paralelo  $E_2, E_1, E_0$   
Si  $LOAD = '0'$ , desplaza

# Contadores con registros de desplazamiento

## Contador en anillo



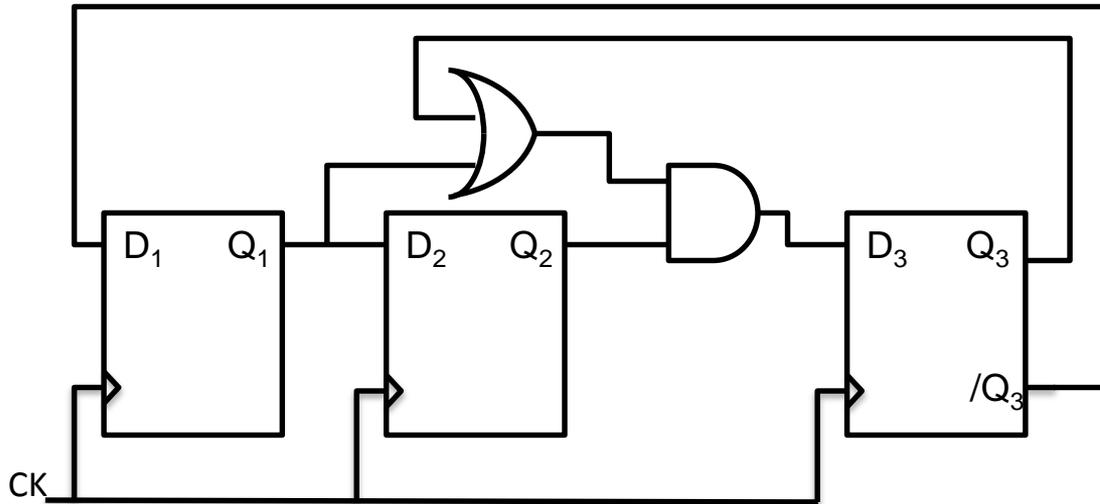
## Contador Johnson



# Diagramas de estado

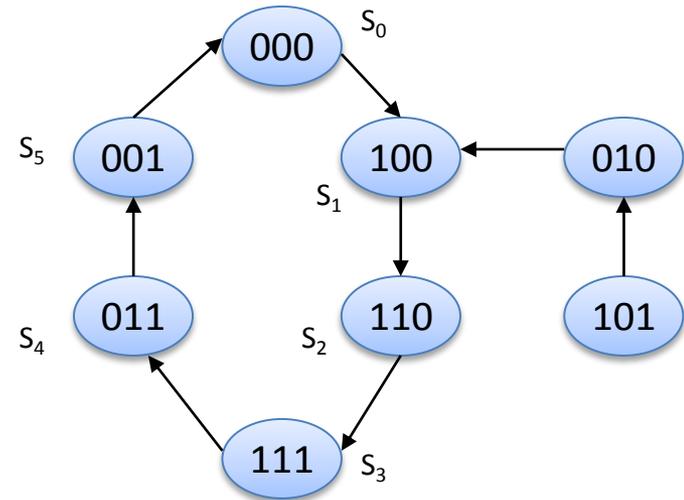
Ejemplo: Para el circuito de la figura, deducir:

- Tabla de estados y diagrama de estados
- Función del circuito
- Decodificar los estados principales del circuito



# Diagramas de estado

| $Q_1$ | $Q_2$ | $Q_3$ | $Q'_1$ | $Q'_2$ | $Q'_3$ |
|-------|-------|-------|--------|--------|--------|
| 0     | 0     | 0     | 1      | 0      | 0      |
| 0     | 0     | 1     | 0      | 0      | 0      |
| 0     | 1     | 0     | 1      | 0      | 0      |
| 0     | 1     | 1     | 0      | 0      | 1      |
| 1     | 0     | 0     | 1      | 1      | 0      |
| 1     | 0     | 1     | 0      | 1      | 0      |
| 1     | 1     | 0     | 1      | 1      | 1      |
| 1     | 1     | 1     | 0      | 1      | 1      |



Es un contador Johnson de 6 estados (3 bits) con autoinicio

| $Q_1$ | $Q_2$ | $Q_3$ | $S_0$ | $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0     |
| 1     | 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     |
| 1     | 1     | 0     | 0     | 0     | 1     | 0     | 0     | 0     |
| 1     | 1     | 1     | 0     | 0     | 0     | 1     | 0     | 0     |
| 0     | 1     | 1     | 0     | 0     | 0     | 0     | 1     | 0     |
| 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0     | 1     |
| 0     | 1     | 0     | x     | x     | x     | x     | x     | x     |
| 1     | 0     | 1     | x     | x     | x     | x     | x     | x     |

$$S_0 = \neg Q_1 \neg Q_3$$

$$S_1 = Q_1 \neg Q_2$$

$$S_2 = Q_2 \neg Q_3$$

$$S_3 = Q_1 Q_3$$

$$S_4 = \neg Q_1 Q_2$$

$$S_5 = \neg Q_2 Q_3$$