Apuntes de 3°GITI

María Ballesteros

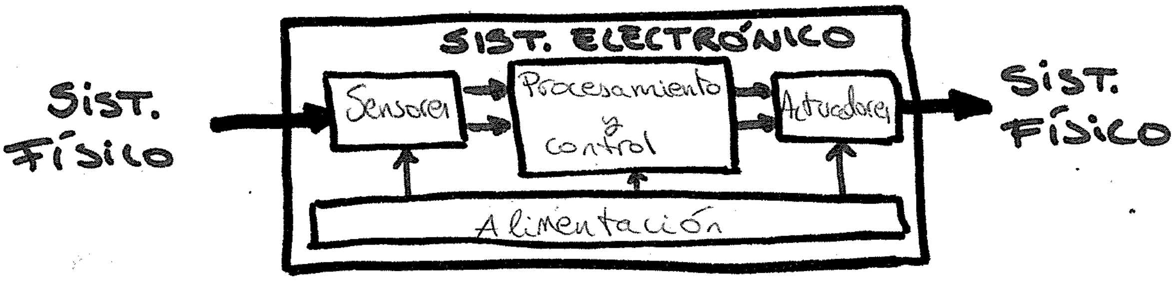
APUNTES ELECTRÓNICA

Si alguna vez estos
apuntes te sirvieron
de ayuda, piensa que
tus apuntes pueden
ayudar a muchas
otras personas.
Comparte tus apuntes
En indusbol.com o
simplyjarod.com

# INTRO: SISTEMAS ELECTRÓNICOS.

## 1 Definición

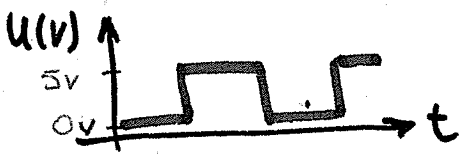
Un sistema electrónico toma información de una magnitud externa de un sistema físico, la transforma en una señal eléctrica (sensores), la procesa y genera otra señal, de la magnitud que corresponda, que actúa sobre el sistema físico.



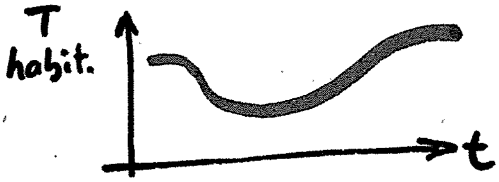

## 2 Tipos

Analógico y digital

- Una señal digital puede tomar un valor entre un nº finito de valores




- Una señal analógica puede tomar un nº infinito de valores.




El mundo real es analógico
¿Por qué usar sistemas digitales?

- Capacidad para manejar gran cantidad de información
  - Fácil de almacenar y transmitir
  - Inmunes al ruido
- Gran desarrollo de la tecnología: Circuitos integrados con millones de trts
- Microprocesadores: Gran capacidad de cálculo

# BINARIO

## 1 Sistemas de numeración

### b=10; SISTEMA DECIMAL

Dígitos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Podemos representar con n dígitos → $10^n$ números

$251 = 2\cdot10^2 + 5\cdot10^1 + 1\cdot10^0$

→ Dígito menos significativo → Dígito más significativo

### b=2; SISTEMA BINARIO

Dígitos: 0, 1 BIT ; Con n bits podemos representar → $2^n$ números

$1011 = 1\cdot2^3 + 0\cdot2^2 + 1\cdot2^1 + 1\cdot2^0$

→ Bit menos significativo least significant bit LSB

→ Bit más significativo. Most significant bit MSB

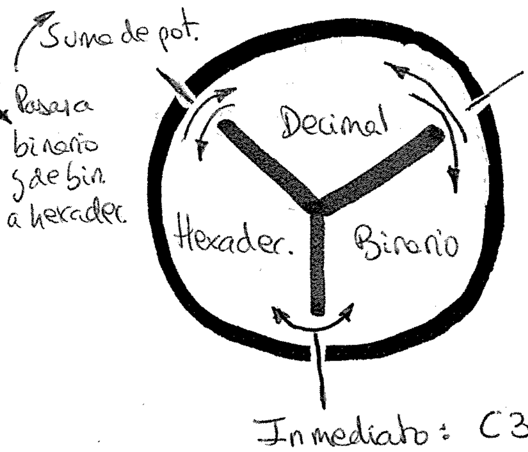
### b=16; SISTEMA HEXADECIMAL

Dígitos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

### b=8 SISTEMA OCTAL

Dígitos: 0, 1, 2, 3, 4, 5, 6, 7

### CAMBIOS DE BASE




Suma de potencias: $1011 = 1\cdot2^3 + 0\cdot2^2 + 1\cdot2^1 + 1\cdot2^0 = 11$

Divisiones sucesivas entre 2:

35 | 2 → 17, resto 1; 17 | 2 → 8, resto 1; 8 | 2 → 4, resto 0; 4 | 2 → 2, resto 0; 2 | 2 → 1, resto 0

$35 = 100011$

Inmediato: $C3A5 = \underbrace{1100}_{C}\ \underbrace{0011}_{3}\ \underbrace{1010}_{A}\ \underbrace{0101}_{5}$

# 2 Binario

Para los números positivos usaremos el código binario natural pero, para los negativos, estudiaremos dos sistemas de codificación:

- Signo magnitud
- Complemento a2

## Signo magnitud

El primer bit es denominado bit de signo → Signo positivo: bit a 'Ø' / Signo negativo: bit a '1'

Ej.: +27 $\boxed{\emptyset}\ \boxed{1\ 1\ 0\ 1\ 1}$

-27 $\boxed{1}\ \boxed{1\ 1\ 0\ 1\ 1}$

## Complemento a 2

Tomando el nº positivo, para hallar su negativo se intercambian los 'Ø' por '1' y viceversa y se suma 1 y añade 1 al principio.

Ej.: -27

$$\begin{array}{rl} 27 \equiv & 1\,1\,0\,1\,1 \\ & 0\,0\,1\,0\,0 \\ + & 1 \\ \hline \boxed{1} & 0\,0\,1\,0\,1 \end{array}$$

$\boxed{1}$ 0 0 1 0 1 representa:

$-1 \cdot 2^5 + 1 \cdot 2^2 + 1 \cdot 2^0 = \underline{\underline{-27}}$

Usando Complemento a2 las restas son sumas:

$$30-27 \begin{cases} 30 \equiv 0\,1\,1\,1\,1\,0 \\ -27 \equiv \begin{cases} 27: 1\,1\,0\,1\,1 \\ \quad\ \ 0\,0\,1\,0\,0 \\ + \qquad\quad\ 1 \\ \hline 1\,0\,0\,1\,0\,1 \equiv -27 \end{cases} \end{cases}$$

$$\begin{array}{rl} & 0\,1\,1\,1\,1\,0 \\ + & 1\,0\,0\,1\,0\,1 \\ \hline & 0\,0\,0\,0\,1\,1 = 3 = 30-27 \end{array}$$

**Ejercicio**: Expresar la resta 27-30 en C.a2:

$27 \longrightarrow \boxed{0}\,11011$

$-30 \longrightarrow \left\{ \begin{array}{r} 30 \quad \boxed{0}\,11110 \\ 00001 \\ +\quad 1 \\ \hline \boxed{1}\,00010 \end{array} \right.$

$$\Rightarrow \begin{array}{r} \boxed{0}\,11011 \\ +\ \boxed{1}\,00010 \\ \hline \boxed{1}\,11101 \end{array} = -1\cdot 2^5 + 2^4 + 2^3 + 2^2 + 2^0 = \underline{\underline{-3}}$$

Podríamos deshacer el cambio de signo

$$\begin{array}{r} \boxed{0}\,00010 \\ +\quad 1 \\ \hline \boxed{0}\,00011 \end{array} = 3$$

**NOTA!!** Otro código que se nos puede presentar en alguna ocasión es el BCD = Binary Code Decimal

Representa los números decimales codificados en binario dígito a dígito. Ej.: $923 \equiv \underbrace{1001}_{9}\ \underbrace{0010}_{2}\ \underbrace{0011}_{3}$

# ÁLGEBRA DE BOOLE

## 1 Variables, operaciones y funciones lógicas

**Variables lógicas:** A {0, 1}

**Operaciones lógicas:**

SUMA → OR

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

PRODUCTO → AND

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

NEGACIÓN O COMPLEMENTO → NOT

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

**Funciones lógicas:** $f(A, B, C, \ldots) = \bar{A}\cdot\bar{B}\cdot C + A\cdot B\cdot C \ldots$

## 2 Teoremas

**Teorema del cierre:** El resultado de aplicar cualquier función booleana a variables booleanas tiene como resultado otra variable booleana.

**Teorema de idempotencia:** $A + A = A$ ; $A \cdot A = A$

**Teorema de involución:** $\bar{\bar{A}} = A$

**Propiedad conmutativa:** $A + B = B + A$ $\quad AB = BA$

Propiedad asociativa: $(A+B)+C = A+(B+C)$ ; $(A\cdot B)\cdot C = A\cdot(B\cdot C)$

Propiedad distributiva: $A\cdot(B+C) = AB + AC$ ; $\underbrace{A + B\cdot C = (A+B)(A+C)}_{!!}$

Parece rara pero se cumple

## Teorema de absorción $A + A \cdot B = A$ ; $A \cdot (A+B) = A$

| A | B | $A \cdot B$ | $A+B$ | $A + A \cdot B$ | $A(A+B)$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |
| | | | | ≡ A | ≡ A |

## Ley de Morgan

$$\boxed{\overline{A+B} = \bar{A} \cdot \bar{B}} \qquad \boxed{\overline{AB} = \bar{A} + \bar{B}}$$

Generalizando: $\overline{\text{Expresión lógica}}$ = Cada una de las subexpresiones negadas cambiando $\cdot \rightleftarrows +$

**NOTA!!** otras propiedades $A \cdot 1 = A$ ; $A \cdot \varnothing = \varnothing$
$A + 1 = 1$ ; $A + \varnothing = A$

- Demo Leyes de Morgan con tablas de verdad

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

| A | B | $A+B$ | $\overline{A+B}$ | $\bar{A}$ | $\bar{B}$ | $\bar{A} \cdot \bar{B}$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 |

=

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

| A | B | $A \cdot B$ | $\overline{AB}$ | $\bar{A}$ | $\bar{B}$ | $\bar{A} + \bar{B}$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 |

=

**NOTA!!** 1ª forma canónica de una función: Suma de productos
2ª forma canónica de una función: Productos de sumas

# 3 Puertas lógicas

INVERSOR | AND | OR | NAND | NOR

| A | $S=\overline{A}$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

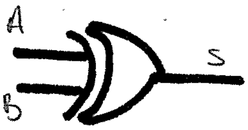
| A | B | $S=A\cdot B$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| A | B | $S=A+B$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

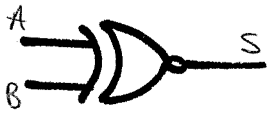
| A | B | $\overline{A\cdot B}=S$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

| A | B | $\overline{A+B}=S$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

XOR



XNOR

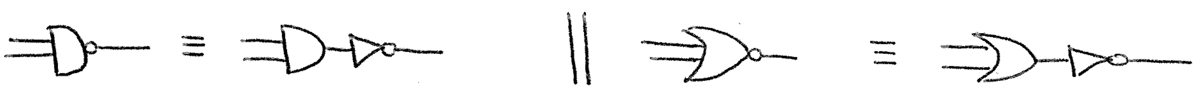


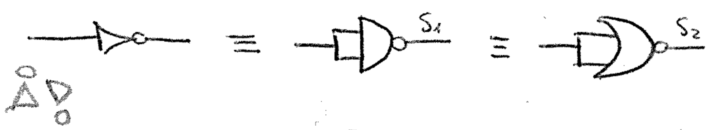
| A | B | $S=A\oplus B$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

| A | B | $=\overline{A\oplus B}$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

## Equivalencia entre puertas:

- Como es lógico, las puertas NAND y NOR pueden conseguirse con puertas AND + inversor u OR + inversor:



- 

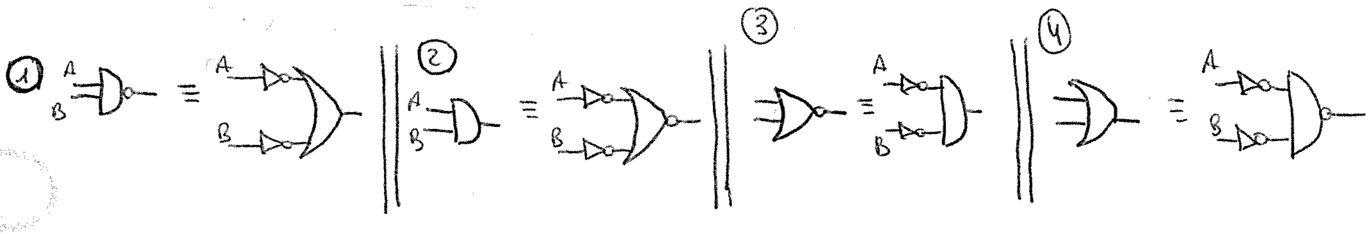
Por las dos entradas entra A

| A | $S_1=\overline{A\cdot A}$ | $S_2=\overline{A+A}$ |
|---|---|---|
| 0 | 1 | 1 |
| 1 | 0 | 0 |

$\equiv \overline{A}$

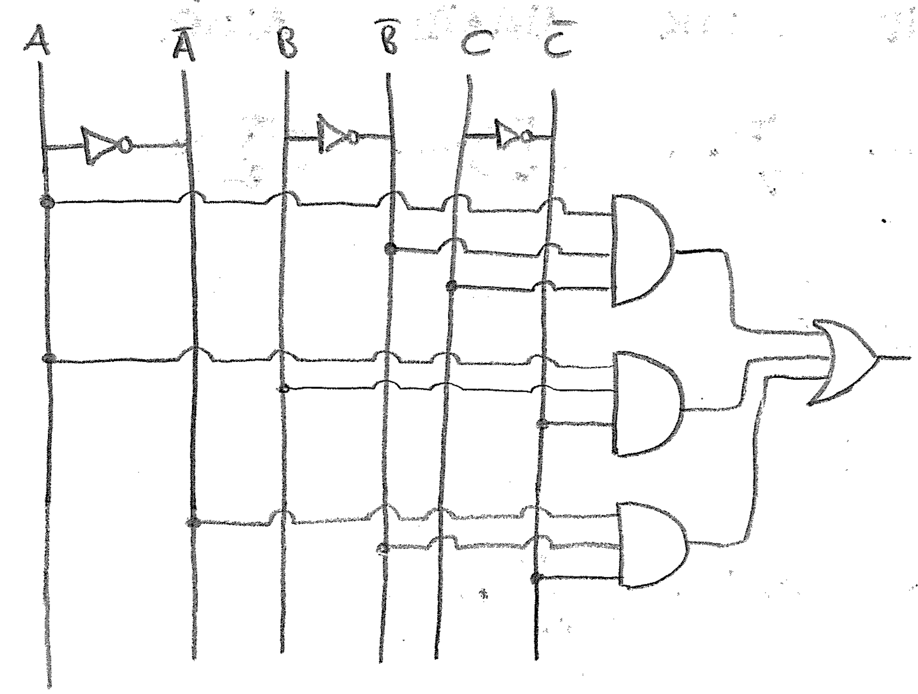
- Leyes de Morgan

$$\overline{A\cdot B}=\overline{A}+\overline{B} \quad (1)$$
$$A\cdot B=\overline{\overline{A}+\overline{B}} \quad (2)$$
$$\overline{A+B}=\overline{A}\cdot\overline{B} \quad (3)$$
$$A+B=\overline{\overline{A}\,\overline{B}} \quad (4)$$

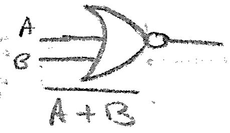


Ejemplo: Representar la función $f = A\bar{B}C + AB\bar{C} + \bar{A}\bar{B}\bar{C}$ mediante puertas NOR

Primero representamos la función como tal:




PUERTA NOR




$A\bar{B}C \overset{\text{Morgan}}{=} \overline{\bar{A}+B+\bar{C}}$

$AB\bar{C} \overset{\text{Morgan}}{=} \overline{\bar{A}+\bar{B}+C}$

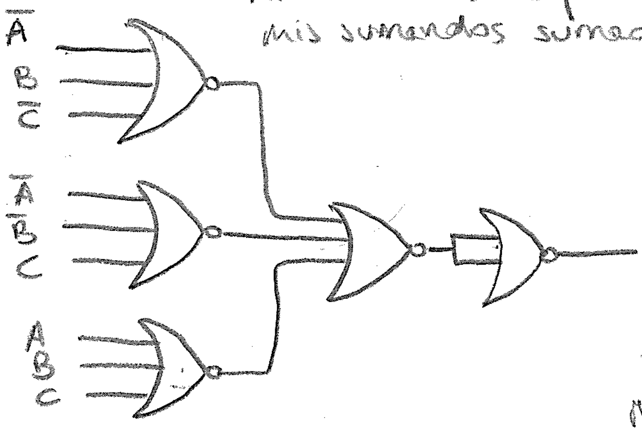
$\bar{A}\bar{B}\bar{C} \overset{\text{Morgan}}{=} \overline{A+B+C}$

Creo puertas NOR DE CADA SUMANDO

Al unir todas las puertas NOR con otra NOR me quedan todos mis sumandos sumados y negados ⟶ Tengo que negarlos otra vez para anular la negación

⇓

Puerta NOR cortocircuitada ≡ Uso inversor o su equivalente




Tenemos inicialmente: $A\bar{B}C + AB\bar{C} + \bar{A}\bar{B}\bar{C} \overset{\text{Morgan a cada sumando}}{=} \left(\overline{\bar{A}+B+\bar{C}}\right) + \left(\overline{\bar{A}+\bar{B}+C}\right) + \left(\overline{A+B+C}\right)$

Hemos obtenido: $\overline{\overline{\left[\left(\overline{\bar{A}+B+\bar{C}}\right) + \left(\overline{\bar{A}+\bar{B}+C}\right) + \left(\overline{A+B+C}\right)\right]}} =$ es equivalente

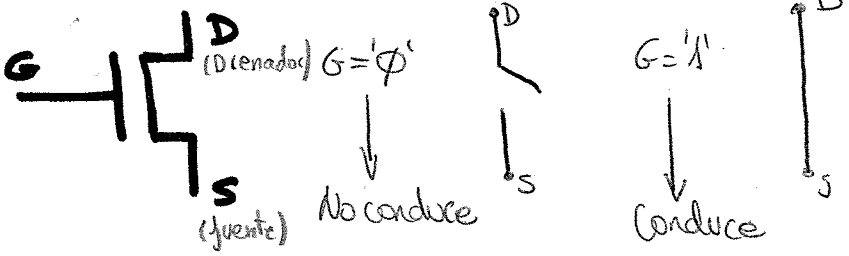
# 4 La lógica del transistor CMOS

Complementary Metal Oxide Semiconductor

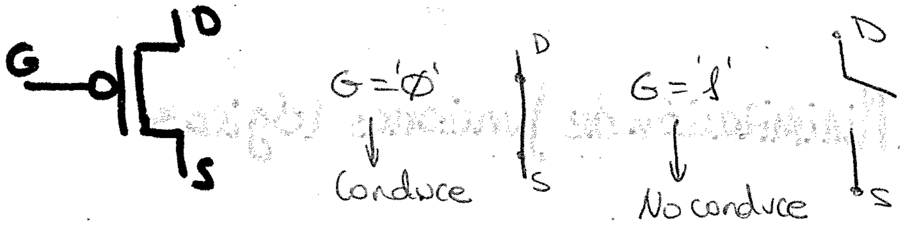
Nos interesa el modo de funcionamiento de los transistores como interruptores controlados para crear puertas lógicas.

Tipos:

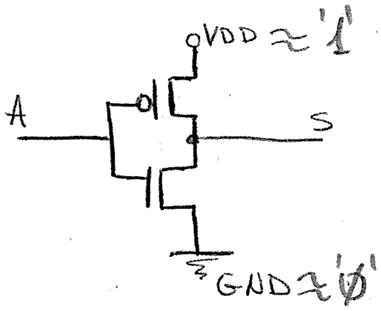
- NMOS




- PMOS




→ INVERSOR CON TRANSISTORES ≡ pMOS + nMOS en serie !!

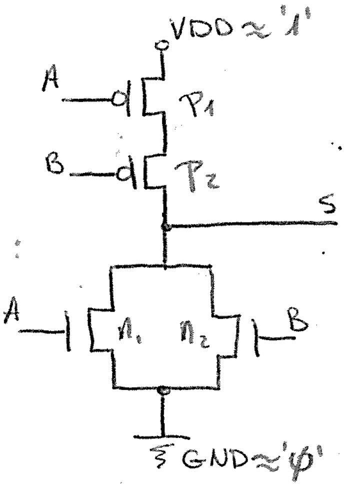



Demo

| A | pMOS | nMOS | S |
|---|---|---|---|
| 0 | Conduce | No cond. | VDD = '1' |
| 1 | No cond | Conduce | GND = '∅' |

$\} \equiv \overline{A}$

→ PUERTA NOR CON TRANSISTORES ≡ 2 pMOS en serie + 2 nMOS en paralelo !!
Todo ello en serie con S entre medias

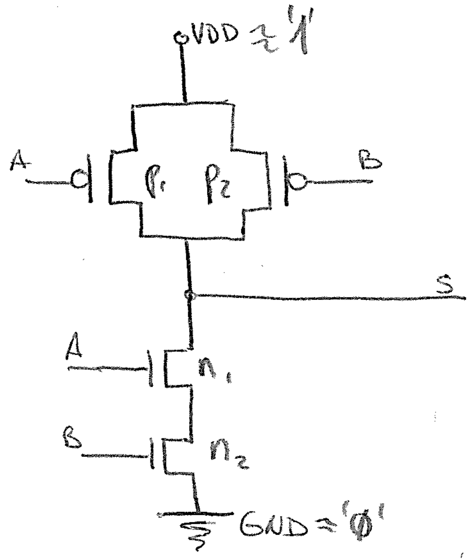



Demo

| A | B | P1 | P2 | n1 | n2 | S |
|---|---|---|---|---|---|---|
| 0 | 0 | Sí | Sí | No | No | VDD ≡ '1' |
| 0 | 1 | Sí | No | No | Sí | GND ≡ '∅' |
| 1 | 0 | No | Sí | Sí | No | GND ≡ '∅' |
| 1 | 1 | No | No | Sí | Sí | GND ≡ '∅' |

$\} \overline{A+B}$ " Puerta NOR

→ PUERTA NAND CON TRANSISTORES ≡ 2 pMOS en paralelo + 2 nMOS en serie !!
Todo ello en serie con S entre medias




Demo

| A | B | $P_1$ | $P_2$ | $n_1$ | $n_2$ | S |
|---|---|---|---|---|---|---|
| 0 | 0 | Sí | Sí | No | No | VDD ≡ '1' |
| 0 | 1 | Sí | No | No | Sí | VDD ≡ '1' |
| 1 | 0 | No | Sí | Sí | No | VDD ≡ '1' |
| 1 | 1 | No | No | Sí | Sí | GND ≡ 'Ø' |

$\overline{A \cdot B}$ ≡ Puerta NAND

# 5 Minimización de funciones lógicas

Usaremos el método de minimización por mapa de Karnaugh. Este método se complica cuando tengamos más de 4 variables pero no será habitual en este curso. Vamos a estudiarlo con el ejemplo dado:

$$f = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}BCD + A\bar{B}C\bar{D} + A\bar{B}CD + ABC\bar{D} + ABCD$$

| 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 |

Solo cambia 1 bit entre vecinos

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

Tenemos que hacer cuadrados o rectángulos lo más grandes posible cuyos lados midan una potencia de 2 incluyendo todos los 1.
Pueden estar varios '1s' en 2 cuadrados a la vez.

La función simplificada queda:

$$\underbrace{\bar A \bar C \bar D}_{\text{Las variables que no cambian en el cuadro verde oscuro}} + \underbrace{AC}_{\text{Idem en verde claro}} + \underbrace{\bar A B C D}_{\text{Idem en cuadro morado}} = f$$

**Ejercicio:** Diseñar un circuito digital que nos devuelva un '1' a la salida cuando la entrada de 4 bits sea un nº primo

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

1ª forma canónica

$$S = \bar A \bar B \bar C D + \bar A \bar B C \bar D + \bar A \bar B C D + \bar A B \bar C D + \bar A B C D + A \bar B C D + A B \bar C D$$

Mapa de Karnaugh

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

También son vecinos!!

$$S = \bar A D + B \bar C D + \bar A \bar B C + \bar B C D$$

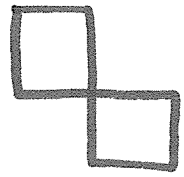
EXTRA: Transformar la función S a funciones de tipo NAND

NAND ≡ Productos negados → Aplico Leyes de Morgan

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

$$S = \overline{(\overline{\bar A D}) \cdot (\overline{A \bar C D}) \cdot (\overline{\bar A \bar B C}) \cdot (\overline{\bar B C D})} \longleftarrow \text{Ya son funciones NAND}$$

**NOTA!!** Las XOR no se deducen aunque suelen aparecer



Por ejemplo: Sea el mapa de Karnaugh

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | Ø | Ø | 1 |
| 01 | Ø | Ø | 1 | Ø |
| 11 | Ø | 1 ? | Ø | Ø |
| 10 | 1 | Ø | Ø | 1 |

Podría ser una XOR!!

$$f = \bar{B}\bar{D} + AB\bar{C}D + \bar{A}BCD$$

$$f = \bar{B}\bar{D} + BD\underbrace{(A\bar{C} + \bar{A}C)}$$

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

$$\parallel$$
$$A \oplus C$$

$$\Longrightarrow \boxed{f = \bar{B}\bar{D} + BD(A \oplus C)}$$

**Ejercicio:** Diseñar un circuito que determine si el mes del año, codificado en la entrada con binario natural 4 bits, tiene 31 días (salida a valor 1) o menos (salida a Ø)

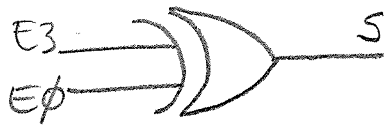
| E3 | E2 | E1 | EØ | S |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | X 4? |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | X |
| 1 | 1 | 1 | 0 | X |
| 1 | 1 | 1 | 1 | X |

Términos indiferentes → Pueden estar dentro o fuera de los cuadrados del mapa si nos ayudan a hacer cuadros más grandes

| E1 EØ \ E3 E2 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | Ø | 1 | 1 |
| 01 | 1 | 1 | X | Ø |
| 11 | 1 | 1 | X | Ø |
| 10 | Ø | Ø | X | 1 |

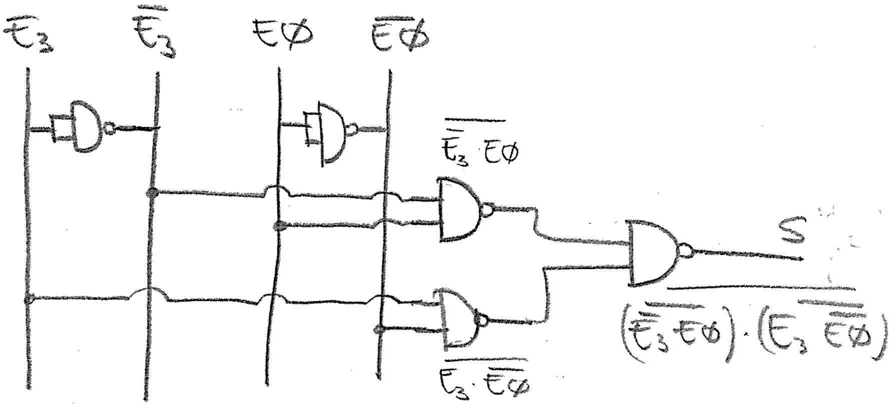
$$S = \overline{E_3}\, E\emptyset + E_3\, \overline{E\emptyset} = E_3 \oplus E\emptyset$$

El circuito podría simplificarse a:




Realizar este circuito sólo con <u>puertas NAND</u> → Productos negados

$$S = \overline{E_3}\, E\emptyset + E_3\, \overline{E\emptyset} \underset{\substack{\uparrow \\ \text{Morgan}}}{=} \overline{\left(\overline{\overline{E_3}\, E\emptyset}\right) \cdot \left(\overline{E_3\, \overline{E\emptyset}}\right)}$$

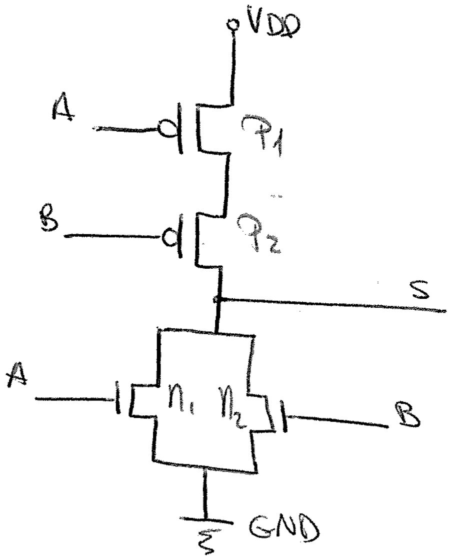



**NOTA!!** Si tuviéramos que elaborar mapas de Karnaugh con 5 variables haríamos → Dejo 1 var fija, por ejemplo, A y elaboro dos mapas de Karnaugh, uno para $A = \emptyset \rightarrow f_1$ y otro para $A = '1' \rightarrow f_2$. La función final sería:

$$f = \overline{A} f_1 + A f_2 \;//$$

# PUERTA NOR

4 TRASISTORES

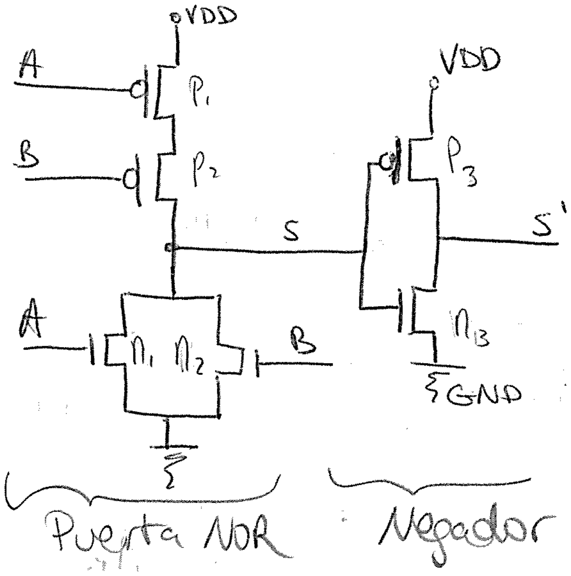



| A | B | P1 | P2 | N1 | N2 | S |
|---|---|---|---|---|---|---|
| 0 | 0 | Sí | Sí | No | No | 1 |
| 0 | 1 | Sí | No | No | Sí | 0 |
| 1 | 0 | No | Sí | Sí | No | 0 |
| 1 | 1 | No | No | Sí | Sí | 0 |

$\rightarrow \overline{A+B}$

# PUERTA OR

6 TRANSISTORES

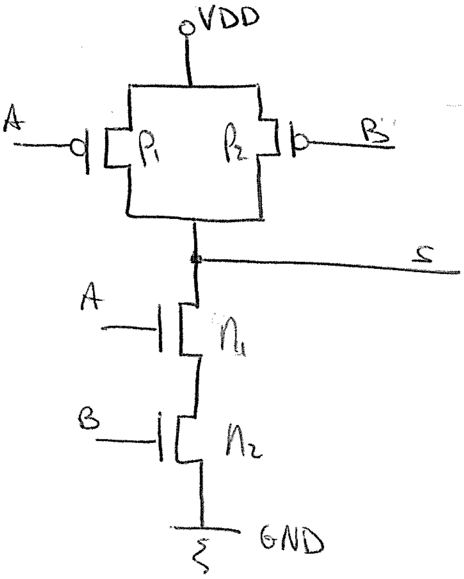



| A | B | P1 | P2 | N1 | N2 | S | P3 | N3 | S' |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | Sí | Sí | No | No | 1 | No | Sí | 0 |
| 0 | 1 | Sí | No | No | Sí | 0 | Sí | No | 1 |
| 1 | 0 | No | Sí | Sí | No | 0 | Sí | No | 1 |
| 1 | 1 | Sí | Sí | No | No | 0 | Sí | No | 1 |

$A+B$

# PUERTA NAND

4 TRANSISTORES

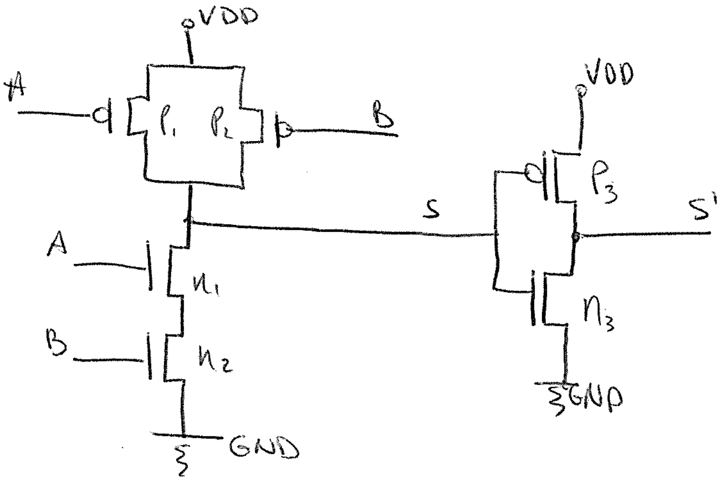



| AB | $P_1$ | $P_2$ | $n_1$ | $n_2$ | S |
|---|---|---|---|---|---|
| 00 | Sí | Sí | No | No | 1 |
| 01 | Sí | No | No | Sí | 1 |
| 10 | No | Sí | Sí | No | 1 |
| 11 | No | No | Sí | Sí | 0 |

$\rightarrow \overline{AB}$

# PUERTA AND

6 TRANSISTORES

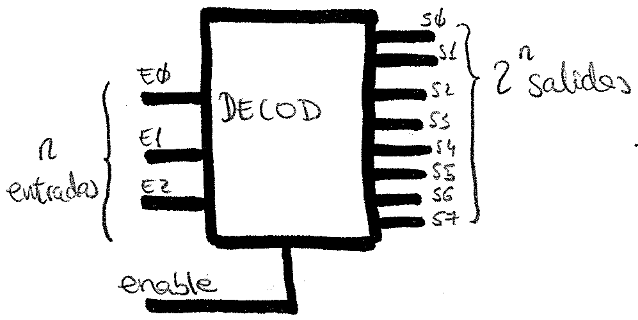



| AB | $P_1$ | $P_2$ | $n_1$ | $n_2$ | S | $P_3$ | $n_3$ | S' |
|---|---|---|---|---|---|---|---|---|
| 00 | Sí | Sí | No | No | 1 | No | Sí | 0 |
| 01 | Sí | No | No | Sí | 1 | No | Sí | 0 |
| 10 | No | Sí | Sí | No | 1 | No | Sí | 0 |
| 11 | No | No | Sí | Sí | 0 | Sí | No | 1 |

AB

# Circ. Combinacionales

## 1 Decodificador




- Cuando el enable está a 'Ø':

  El decodificador no funciona, esto es, todas las salidas y entradas a Ø

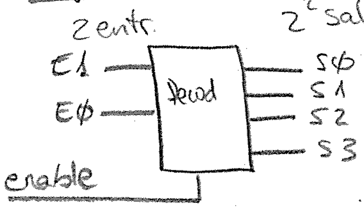
- Cuando el enable está a '1':

  Todas las salidas están a Ø excepto la salida cuyo nº codifica la entrada

  Por ejemplo: E2, E1, EØ = "110" (6) ⇒

  ⇒ SØ, S1, .... S5, S7 = 'Ø'
  S6 = '1'

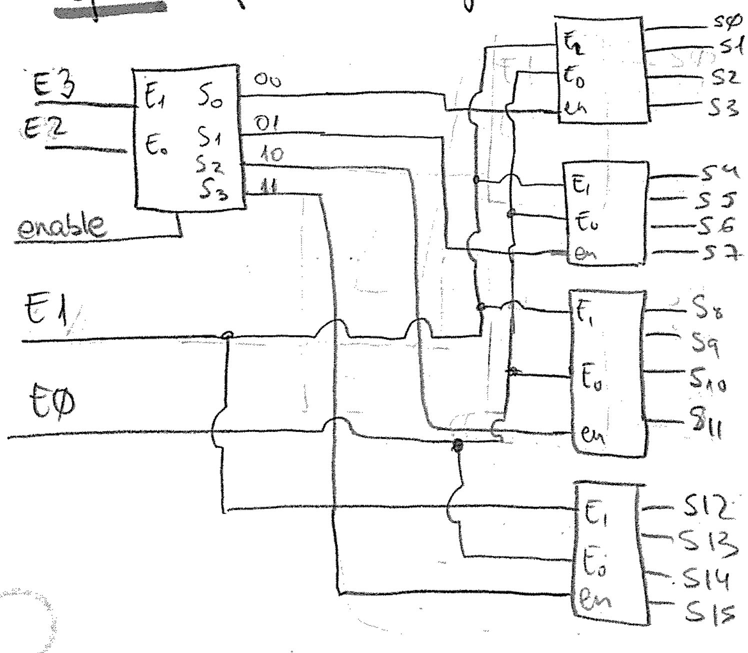
Ejemplo decod. de 2 entradas:




| en | E1 | EØ | SØ | S1 | S2 | S3 |
|---|---|---|---|---|---|---|
| 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 |

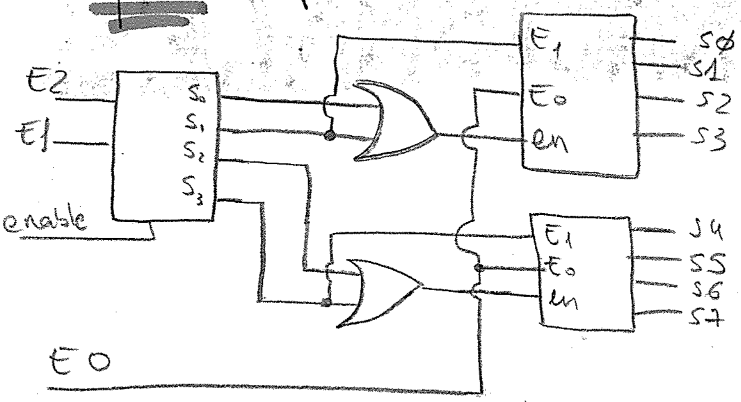
$S\emptyset = en \cdot \overline{E1}\,\overline{E\emptyset}$; $S1 = en \cdot \overline{E1}\,E\emptyset$; $S2 = en\,E1\,\overline{E\emptyset}$, $S3 = en\,E1\,E\emptyset$

Ejercicio: A partir de decodificador de 2 entradas, construir uno de 4: ⟶ $2^4 = 16$ salidas

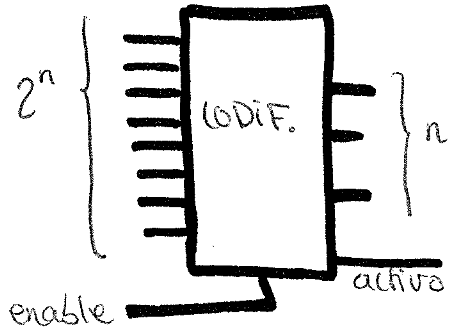



| | | | | |
|---|---|---|---|---|
| 0 0 | 0 0 | | | |
| 0 0 | 0 1 | → en — $S_0$ | | |
| 0 0 | 1 0 | | | |
| 0 0 | 1 1 | | | |
| 0 1 | 0 0 | en — $S_1$ | | |
| 0 1 | 0 1 | | | |
| 0 1 | 1 0 | | | |
| 0 1 | 1 1 | | | |
| 1 0 | 0 0 | en → $S_2$ | | |
| 1 0 | 0 1 | | | |
| 1 0 | 1 0 | | | |
| 1 0 | 1 1 | | | |
| 1 1 | 0 0 | en $S_3$ | | |
| 1 1 | 0 1 | | | |
| 1 1 | 1 0 | | | |
| 1 1 | 1 1 | | | |

Ejercicio: A partir de un decodificador de 2 entradas construir uno de 3




# 2 Codificador




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

¿Qué pasa si hay más de una entrada activa?
Si se trata de un codificador prioritario al más alto se dará prioridad a la entrada más alta.

activo ≡ Si no hay ninguna entrada a 1

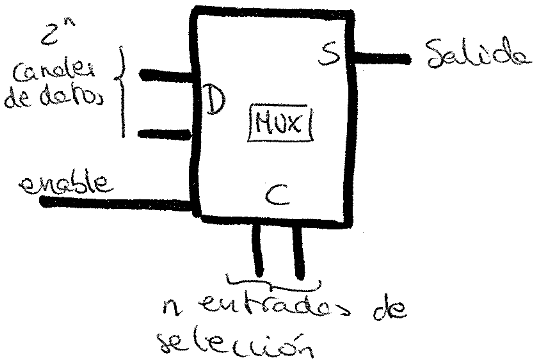
Ejemplo codificador 4 entradas/2 salidas

| en | $E_0$ | $E_1$ | $E_2$ | $E_3$ | $S_1$ | $S_0$ | 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 | Activo |
| 1 | X | 1 | 0 | 0 | 0 | 1 | 1 | Activo |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | Activo |

También existe el codificador prioritario al más bajo

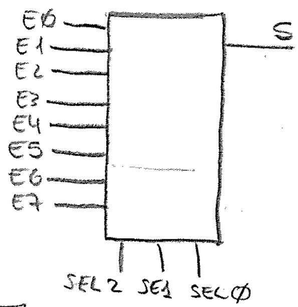
# 3 Multiplexores y Demultiplexores

MULTIPLEXOR ≡ MUX




La salida sigue a la entrada cuyo nº indica la selección

Ejemplo: n=3

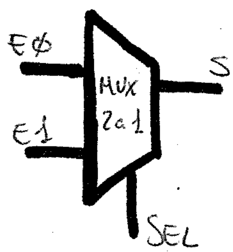



SEL(2:0)='1 0 1' (=5)

S = E5

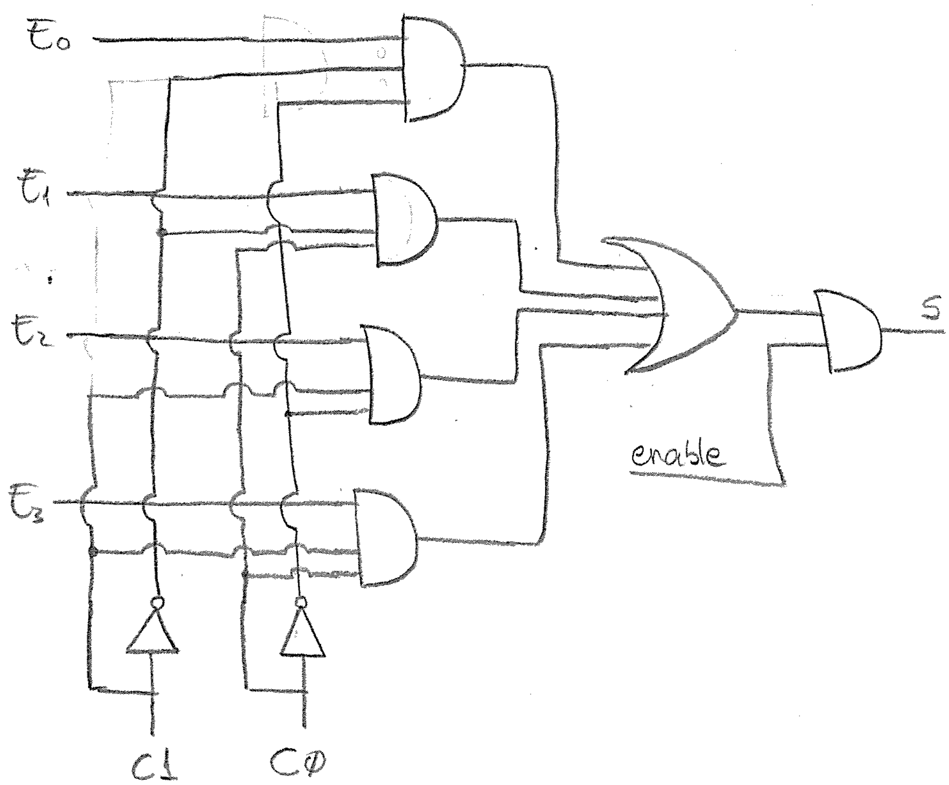
Tienen un símbolo própio

Ejemplo de MUX especial llamado MUX 2 a 1

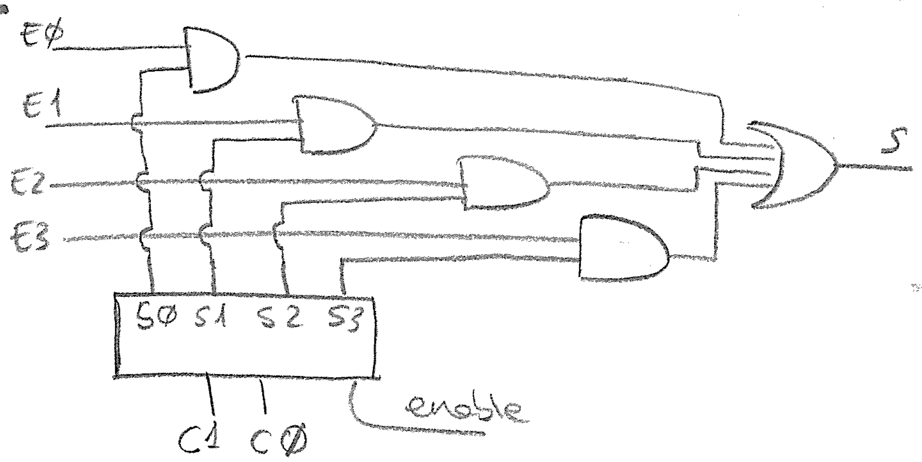



$S = \overline{SEL} \cdot E\emptyset + SEL \cdot E1$

Ejercicio: MUX con puertas lógicas



**Ejercicio:** MUX con decodificador e puertas



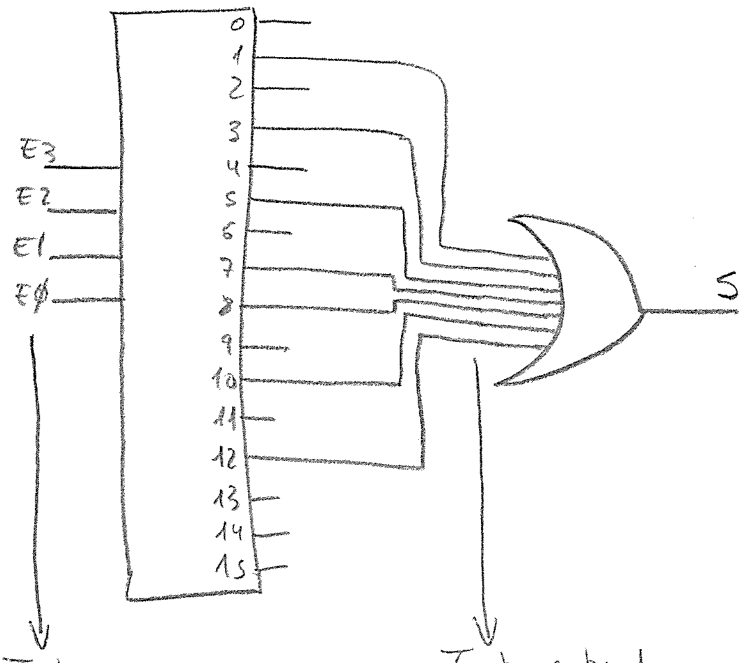

**DEMULTIPLEXORES:** Hacen lo contrario a los MUX. Sacan la entrada por aquella salida correspondiente al nº codificado en las señales de control

**Ejercicio:** Diseñar un circuito que tenga como entrada un mes del año codificado en binario y como salida un '1' si el mes es de 31 días y un '∅' en el caso contrario

a) Mediante un decodificador y una puerta OR
b) Mediante un MUX e inversores
c) Mediante MUXs 2 a 1 e inversores

| E3 | E2 | E1 | E0 | S |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | X |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | ∅ |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | ∅ |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | ∅ |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | ∅ |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | ∅ |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | X |
| 1 | 1 | 1 | 0 | X |
| 1 | 1 | 1 | 1 | X |

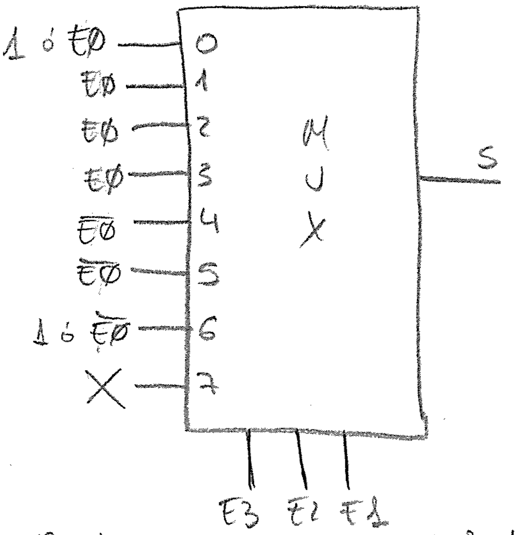
a) Decod + OR




Tantas entradas (al decod) como var. tenga la función lógica

Tantas entradas a la OR como '1's haya en la tabla de verdad

b) MUX + INVERSORES

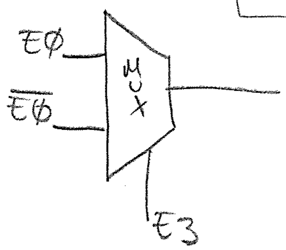



Nº entradas de selección del MUX = nº entradas función − 1

c) MUX 2 a 1

Por Karnaugh se sabe que $S = \overline{E_3}\,E\emptyset + E_3\,\overline{E\emptyset}$ | La salida del MUX es $S = \overline{SEL}\,E\emptyset + SEL\,E1$

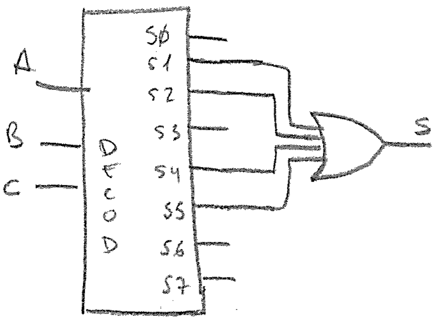
$SEL \equiv E_3$



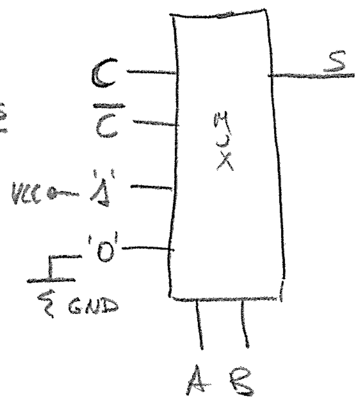

Ejercicio: Dado la tabla de verdad, representarla con 3 diseños diferentes

| A | B | C | S | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | S=C |
| 0 | 0 | 1 | 1 | |
| 0 | 1 | 0 | 1 | $S=\overline{C}$ |
| 0 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 1 | S=1 |
| 1 | 0 | 1 | 1 | |
| 1 | 1 | 0 | 0 | S=0 |
| 1 | 1 | 1 | 0 | |

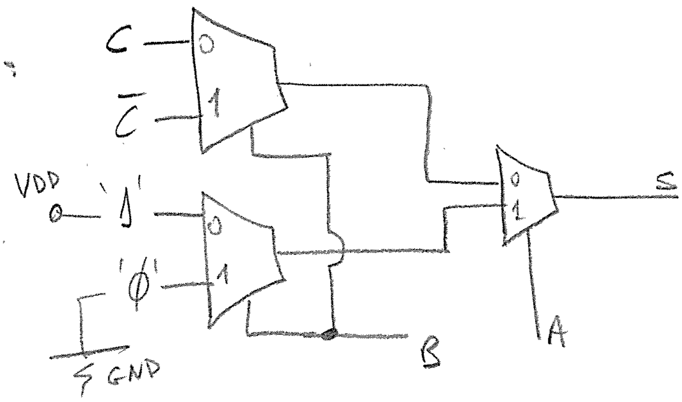
Diseño 1



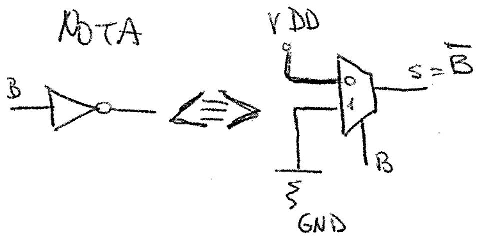

Diseño 2




Diseño 3:




NOTA



# CIRC. SECUENCIALES

## 1 Definición y tipos

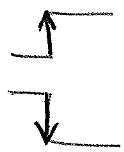
Un circuito secuencial es aquel en el que las salidas en un instante de tiempo dependen de las entradas en ese instante de tiempo y en instantes anteriores.

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.

Circuitos Asíncronos: Cambia de estado y de salida frente a un cambio de las entradas adecuado

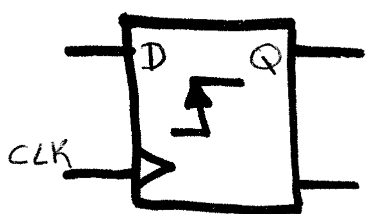
Circuitos Síncronos: Cambia de estado cuando se produce un evento de una señal especial que entra a los biestables y se denomina señal de reloj.

Los cambios se producen en los flancos del reloj

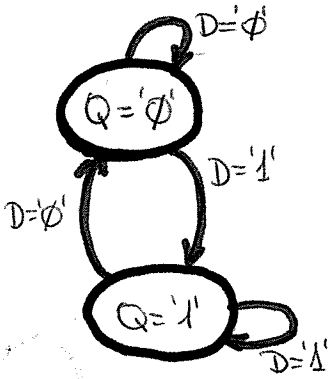


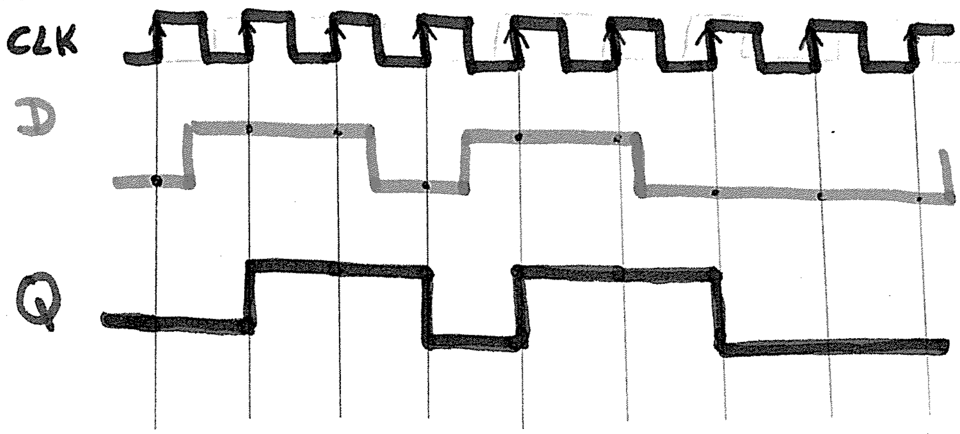

## 2 Biestables síncronos

### BIESTABLE D (dato)

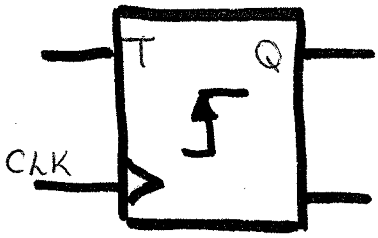



Cuando se produce el flanco activo del reloj de la señal CLK (⌐⇡) Q se actualiza con el valor de D

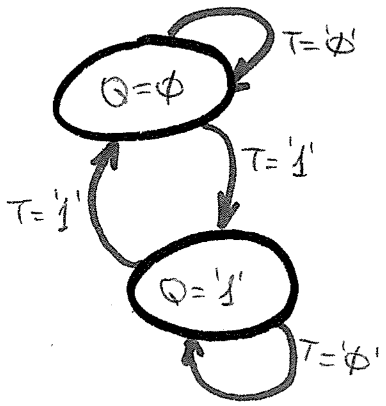


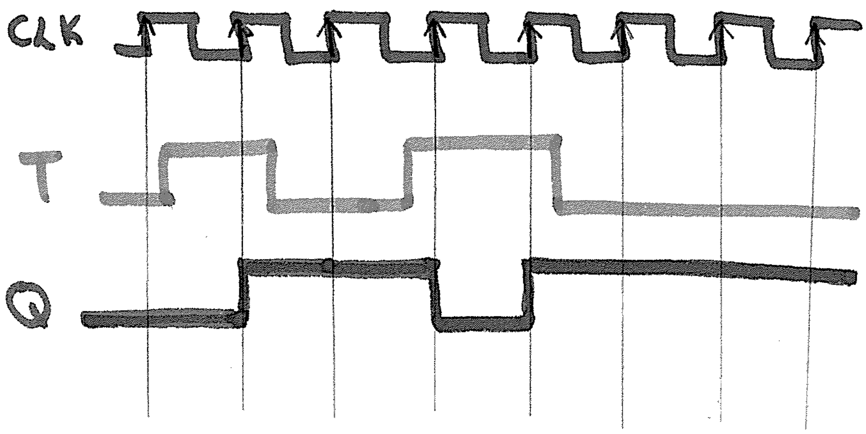



# BIESTABLE T (toggle)

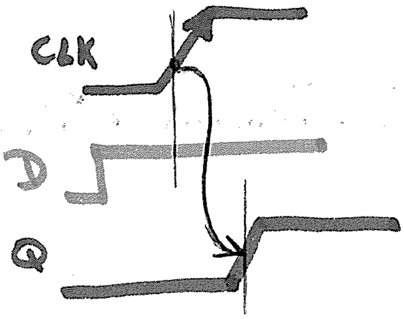



Cuando llega el flanco activo del reloj CLK (↑) Si $T=1 \rightarrow Q$ cambia
Si $T=0 \rightarrow Q$ se mantiene

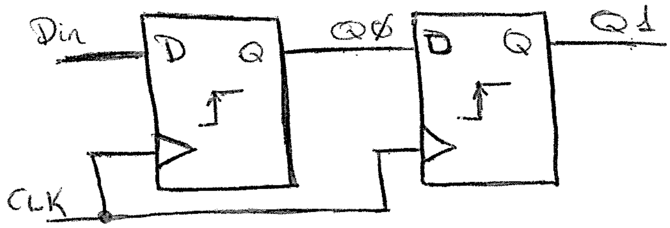


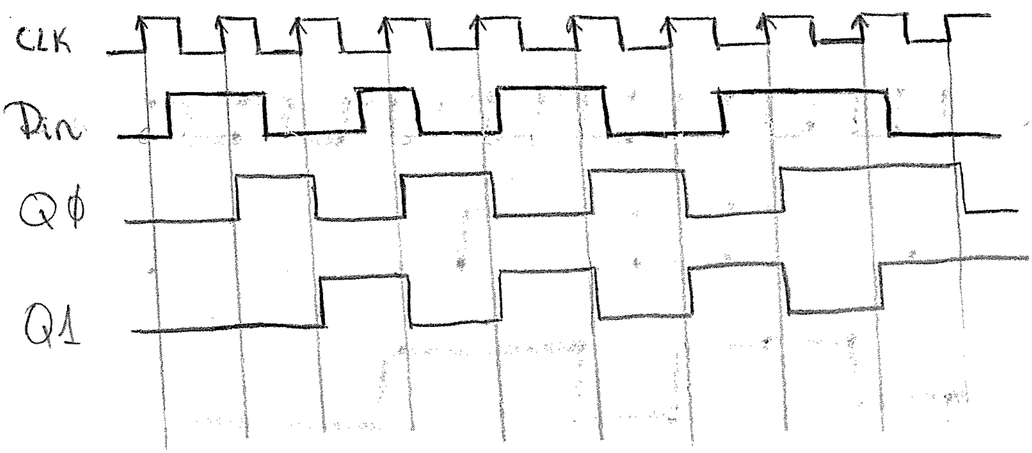




NOTA!! Retardo en señales → Hay que tener muy presente que 1º se produce la subida o bajada del flanco del reloj y después el cambio en Q




Ejercicio: Representar QØ y Q1



# 3 Otras señales de los biestables

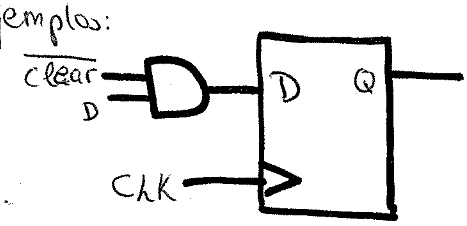
## Inicialización de biestables

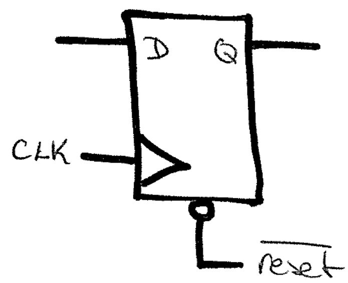
| | 0 | 1 |
|---|---|---|
| Asíncronos | RESET | PRESET |
| Síncronos | CLEAR | SET |

La inicialización síncrona espera al primer flanco activo del reloj mientras que la asíncrona es inmediata.

Las señales de inicialización suelen ser activas por nivel bajo

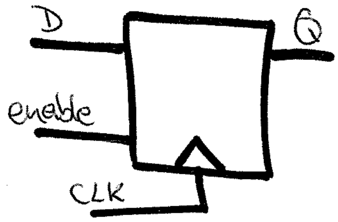
Ejemplos:

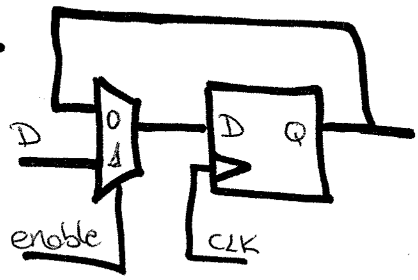



Biestables D con clear y reset.

## Señal enable en biestables D

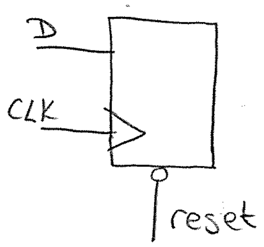



No confundir este enable con el combinacional

| E | D | $Q_{t+\Delta t}$ |
|---|---|---|
| 0 | X | $Q_t$ |
| 1 | D | D |

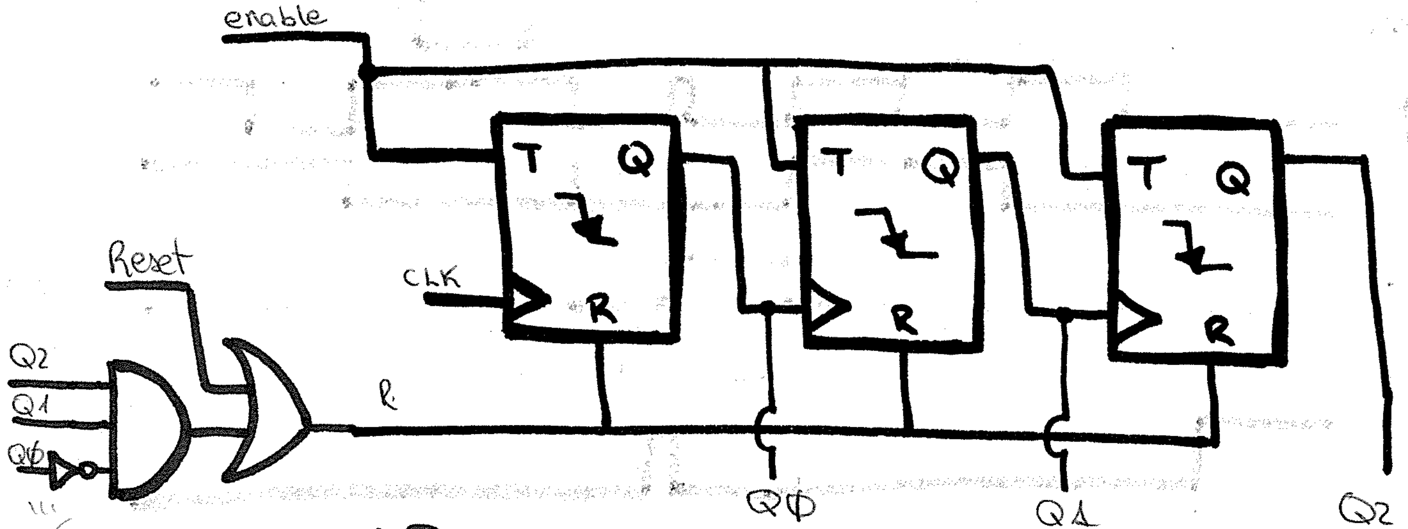
Ejercicio: A partir de un biestable D construir uno T con clear, reset y carga. Orden: reset → clear → carga → T



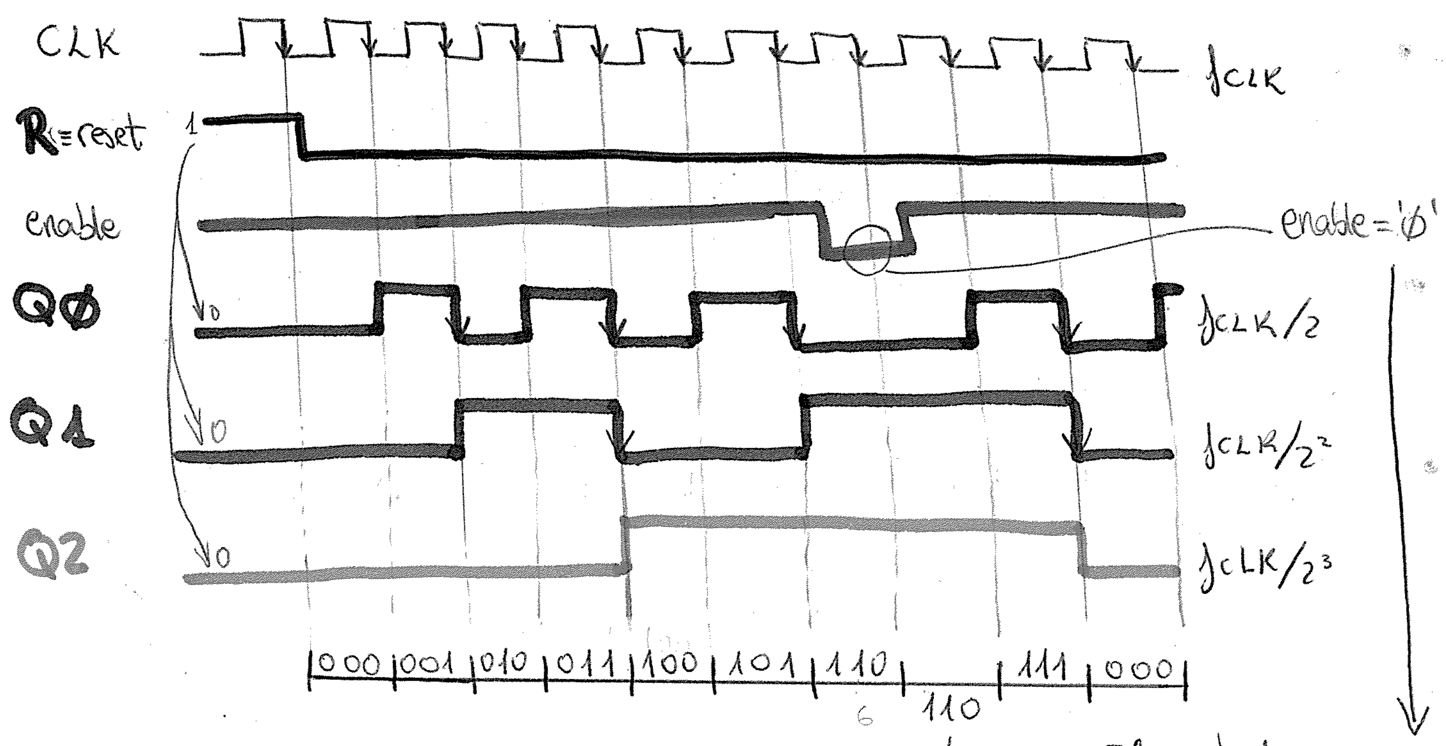
# APLICACIONES -CIRCUÍTOS SECUENCIALES-

## 1 Contadores

Contadores asíncronos: Aunque los biestables son síncronos el reloj no es común a todos ellos



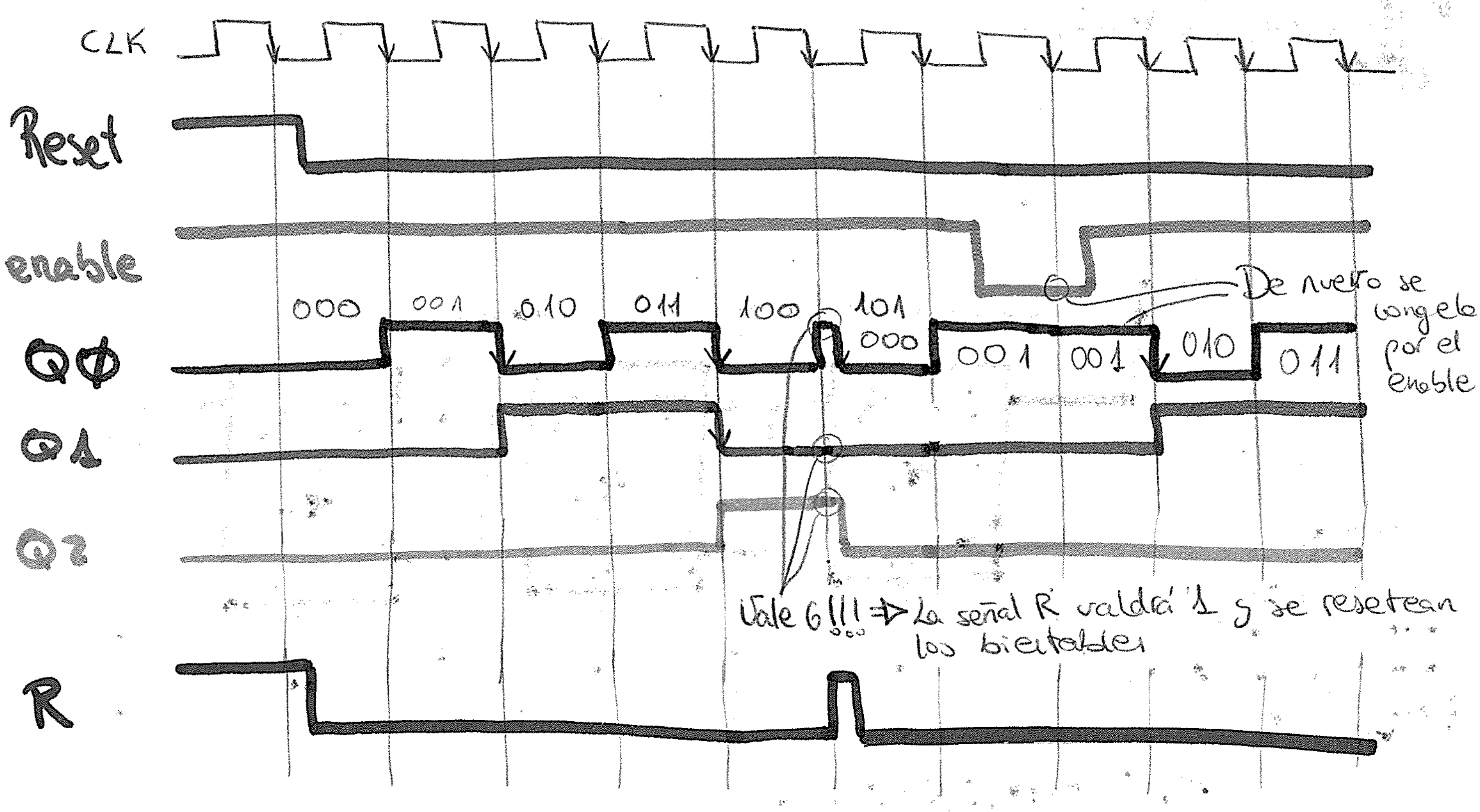

SIN PARTE ROJA




El contador se congela y repite el número

# CON PARTE ROJA

Añadiendo esta parte al contador podremos controlar el valor de la señal R de forma que el contador se inicializará en el valor que queramos. En este caso, el 6 '110'
Q2 Q1 Q0

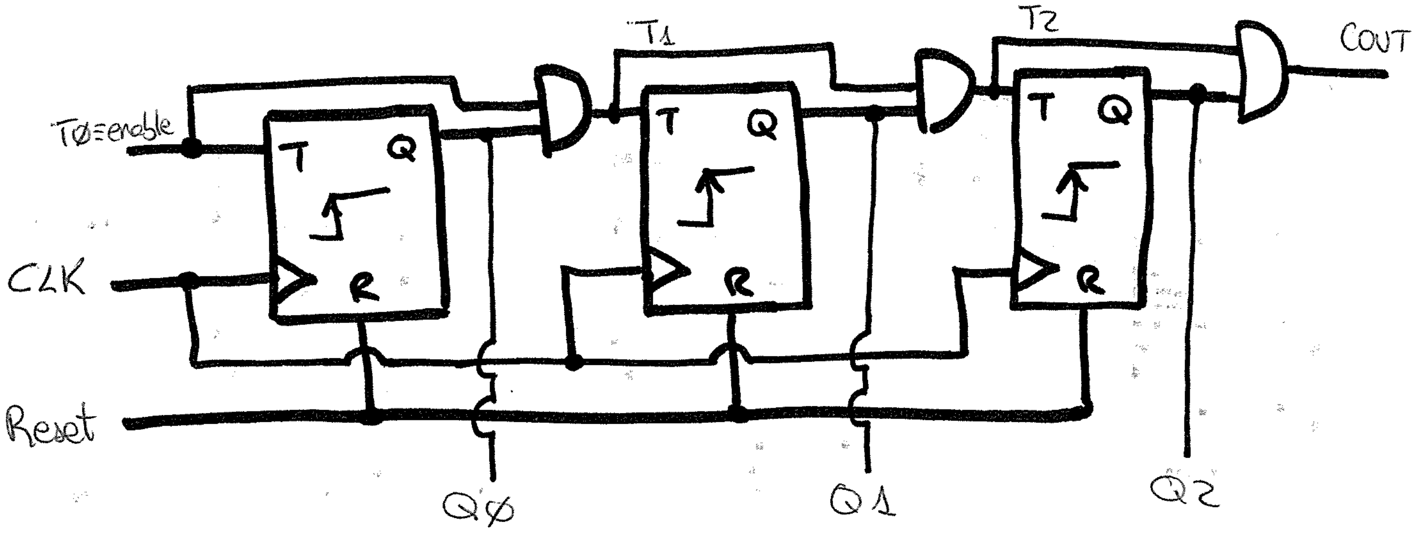


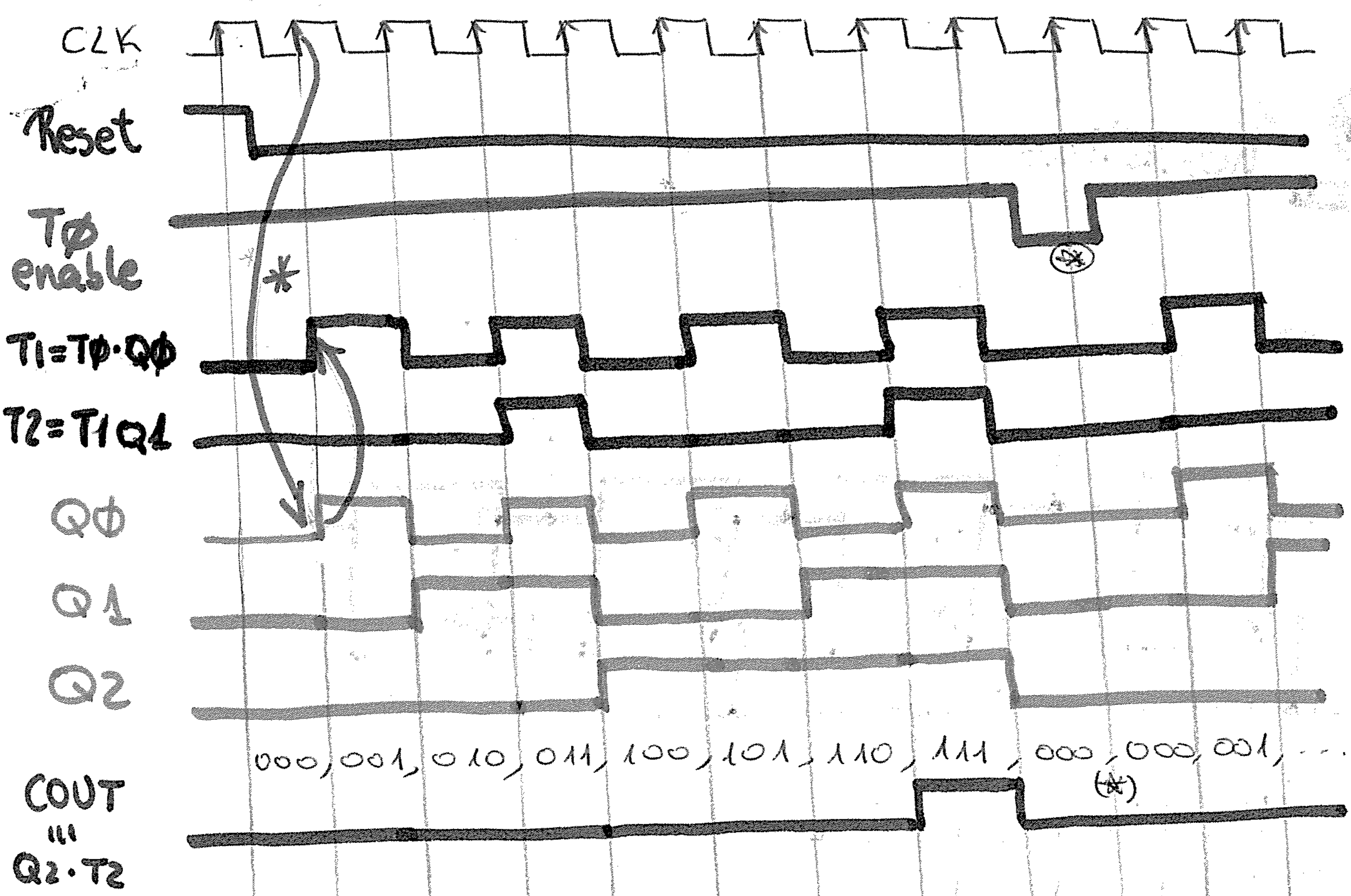

- Se debe codificar en la puerta AND el siguiente valor al valor en que queramos que pare.
  Por eso para de contar en el 5, pues hemos codificado un 6.

- El hecho de que la señal R cambie de valor instantaneamente puede generar problemas

- Si montamos un contador asíncrono con biestables T activos por flanco de subida del reloj, el contador cuenta hacia atrás.

# Contadores síncronos



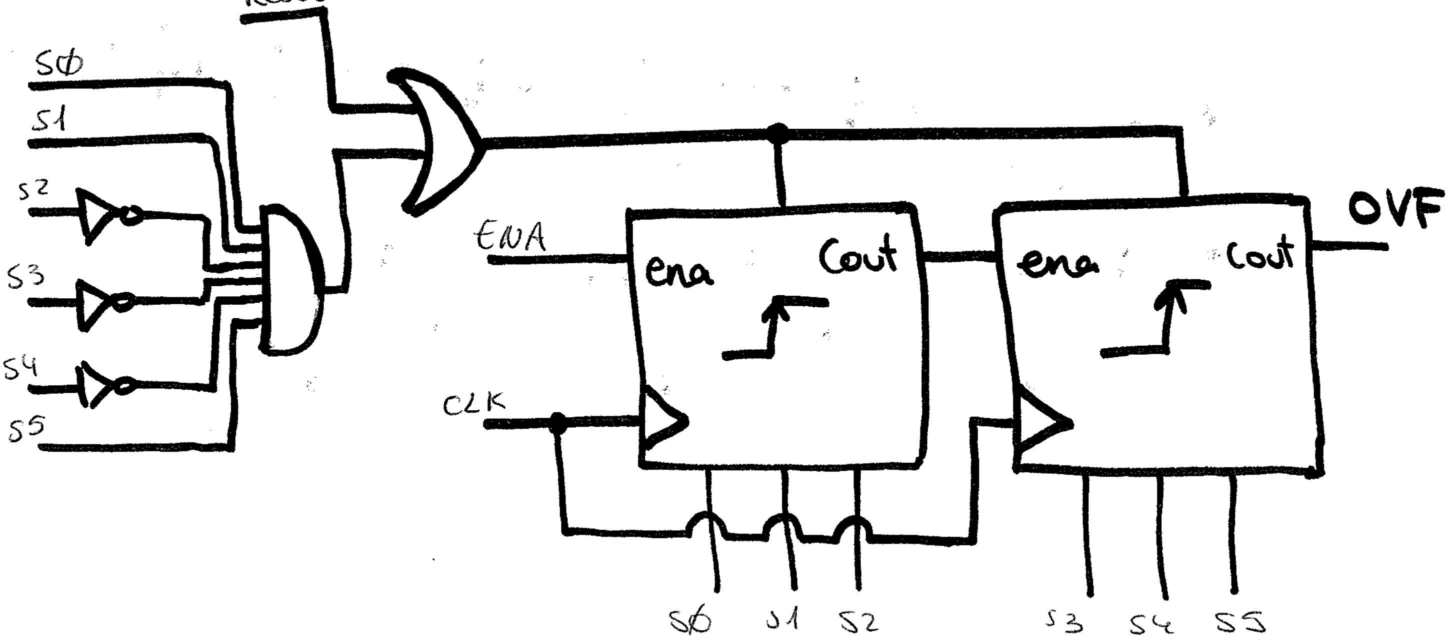




(*) De nuevo el enable = Ø congela el contador

(*) Es importante considerar el orden de los sucesos:

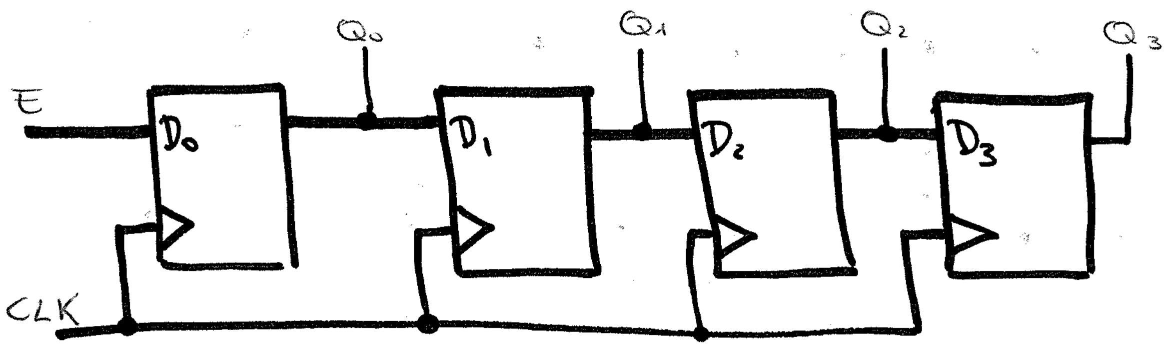
1º sube el CLK ↑; esto hace subir a QØ; que hace subir a T1

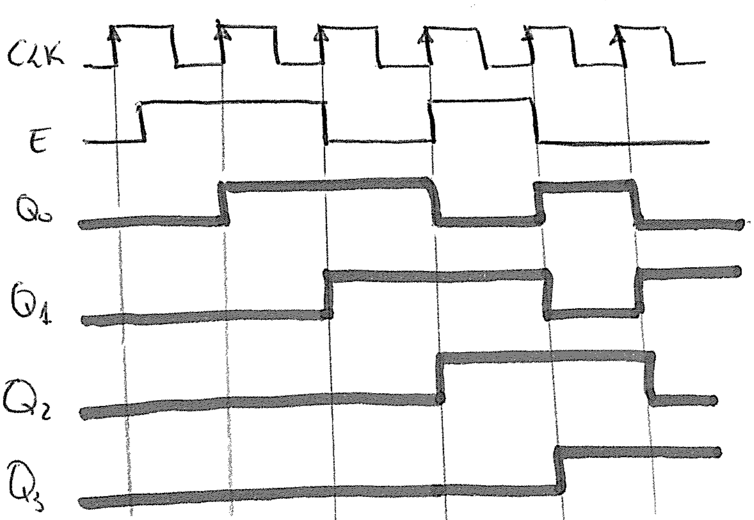
Ejemplo contador síncrono de 6 bits que cuenta hasta 35




# 2 Registros

## REGISTROS DE DESPLAZAMIENTO




Interpretación de las señales:

Por E entra lo que quiero almacenar.
La entrada se desplaza en el tiempo

Indusbol.com

## Tipos de registros

Ver diapositivas !!

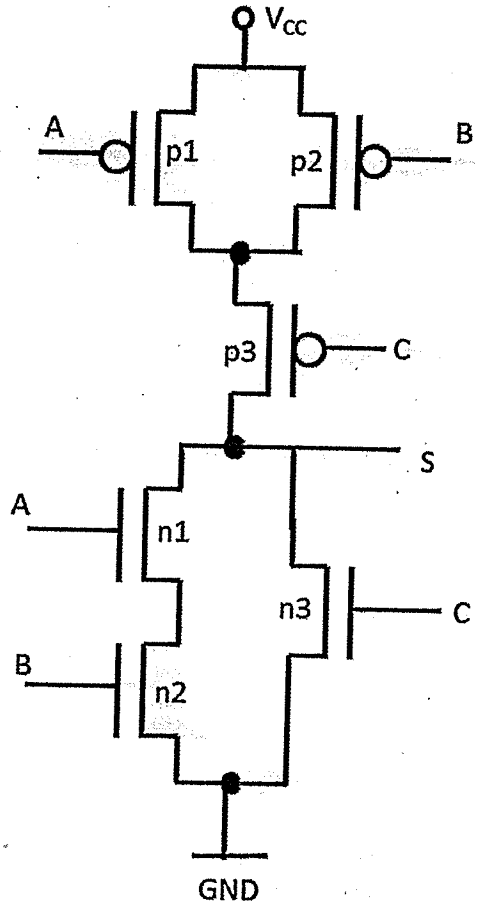
**Asignatura: Fundamentos de Electrónica (55000025)**
**1ª prueba de evaluación continua (Electrónica Digital)**

**Fecha: 15/10/2012**

**A**

**PROBLEMA 1.** **(2 PUNTOS)**

Para el circuito de la figura, indica el estado de los transistores n1, n2, n3 y p1, p2 y p3 (este estado se indicará como "ON" cuando el transistor conduzca y "OFF cuando no lo haga), ante todas las posibles combinaciones de las entradas A, B y C, así como el valor de la salida S. Indique la función lógica de la salida S.




**PROBLEMA 2.** **(4 PUNTOS)**

Para la siguiente función lógica

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

se pide:

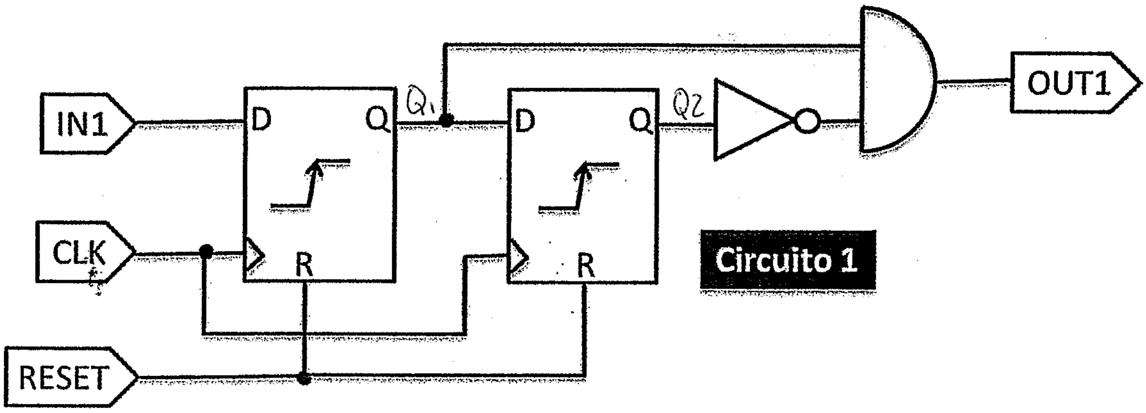
a) Obtener la tabla de verdad
b) Obtener la expresión mínima de la función
c) Implementar la función F:
   1. Minimizada, usando puertas NAND
   2. Con multiplexores (e inversores, si hicieran falta)
   3. Con un decodificador y una puerta OR

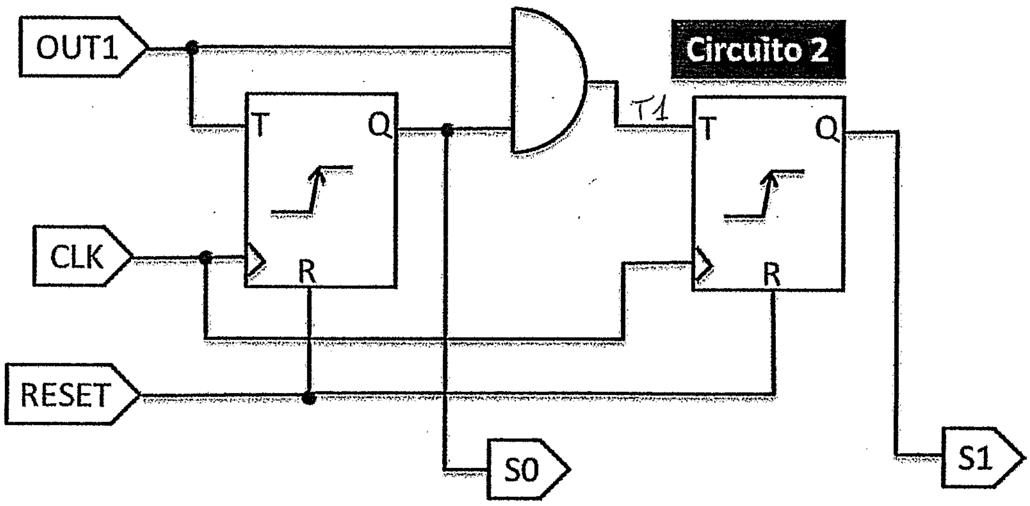
**PROBLEMA 3.** **(4 PUNTOS)**

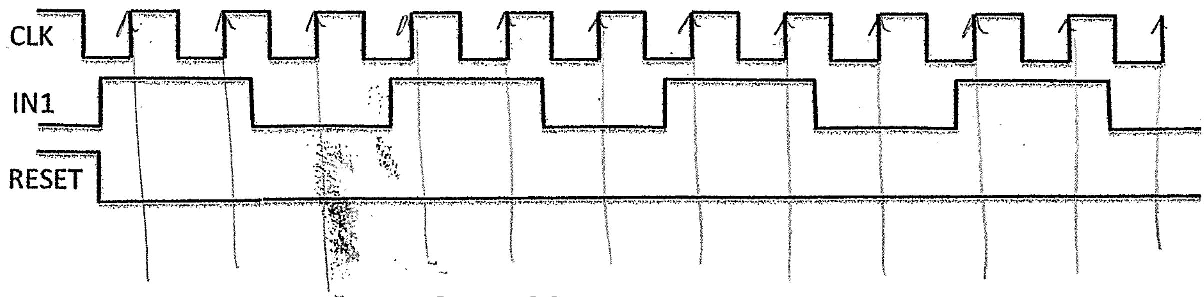
Para el circuito siguiente dibuja la evolución de las señales OUT1 y S1 y S0, para los estímulos de entrada mostrados en la figura. ¿Qué función tiene el "circuito 2"?

Nótese que:

- Las señales CLK y RESET son comunes a ambos "circuito 1" y "circuito 2"
- La señal de salida del "circuito 1" OUT1 es la entrada del "circuito 2"
- La señal RESET es una señal de inicialización asíncrona de los biestables que pone a '0' el estado del biestable cuando la señal de RESET vale '1', de manera asíncrona




Cada problema se entregará en hoja(s) separada(s)

Indica **en todas las hojas entregadas**: nombre, número de matrícula, grupo y tipo de examen (A ó B)

Las calificaciones provisionales se publicarán el 29 de octubre de 2012

Tiempo total de la prueba: **90 minutos**

# PROBLEMAS

# ELECTRÓNICA DIGITAL

## Examen Junio 2012 → Ejercicio 3

Se dispone de dos nº de 2 bits ("A1, AØ" y "B1, BØ"). Se quiere construir un circuito que dé un '1' si A > B y un 'Ø' en caso contrario, siéndole indiferente que A=B. Se pide

a) Tabla de verdad de la salida S:

| A1 | AØ | B1 | BØ | S |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | X |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 ← |
| 0 | 1 | 0 | 1 | X |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 ← |
| 1 | 0 | 0 | 1 | 1 ← |
| 1 | 0 | 1 | 0 | X |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 ← |
| 1 | 1 | 0 | 1 | 1 ← |
| 1 | 1 | 1 | 0 | 1 ← |
| 1 | 1 | 1 | 1 | X |

b) Primera forma canónica

$$S = \overline{A1}\cdot A\emptyset \cdot \overline{B1}\,\overline{B\emptyset} + A1\cdot\overline{A\emptyset}\cdot\overline{B1}\cdot\overline{B\emptyset} + A1\cdot\overline{A\emptyset}\cdot\overline{B1}\cdot B\emptyset +$$
$$+ A1\,A\emptyset\,\overline{B1}\,\overline{B\emptyset} + A1\,A\emptyset\,\overline{B1}\,B\emptyset + A1\,A\emptyset\,B1\,\overline{B\emptyset}$$

c) Dibujar el circuito minimizado con:

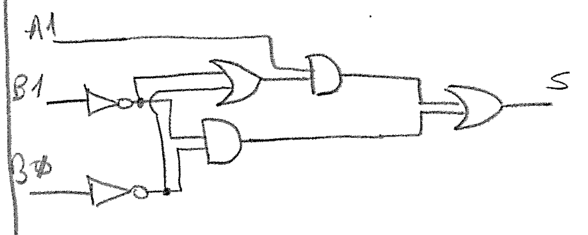
- Puertas NAND
- Puertas NOR
- Cualquier tipo de puerta
- Multiplexores (e inversores)
- Decodificadores y puertas OR

Minimización por Karnaugh

| B1BØ \ A1AØ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | 1 | 1 | 1 |
| 01 | Ø | X | 1 | 1 |
| 11 | Ø | Ø | X | Ø |
| 10 | Ø | Ø | 1 | X |

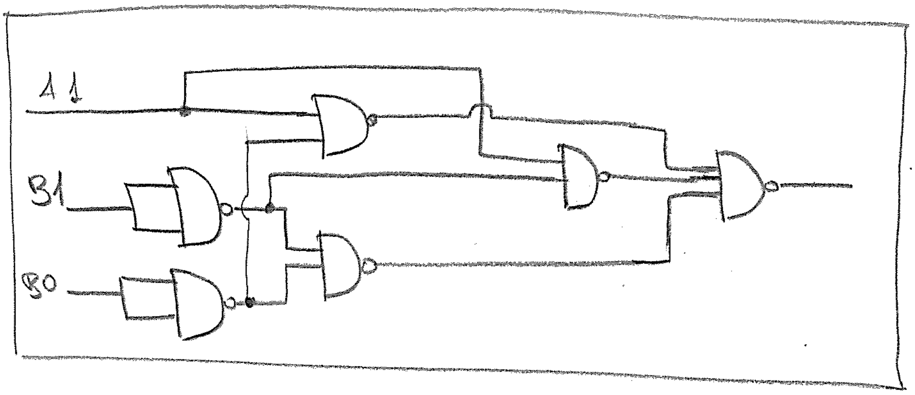
$$S = \overline{B1}\,\overline{B\emptyset} + A1\,\overline{B1} + A1\,\overline{B\emptyset} =$$
$$= \overline{B1}\,\overline{B\emptyset} + A1\,(\overline{B1} + \overline{B\emptyset})$$

*Con cualquier tipo de puerta:



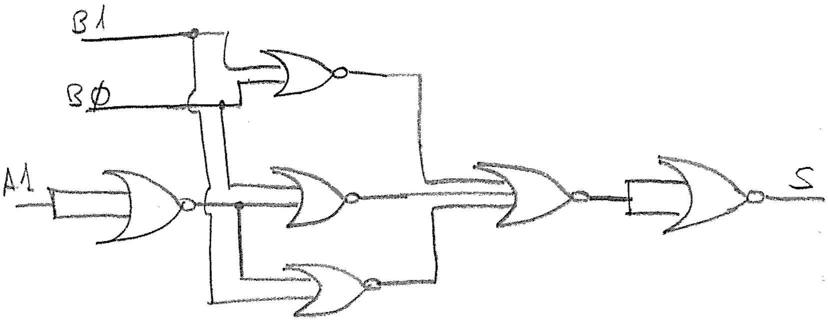
## * Con puertas NAND (productos negados)

$$S = \bar{B1}\,\bar{B0} + A1\,\bar{B0} + A1\,\bar{B1} = \overline{\overline{\bar{B1}\,\bar{B0}} \cdot \overline{A1\,\bar{B0}} \cdot \overline{A1\,\bar{B1}}}$$



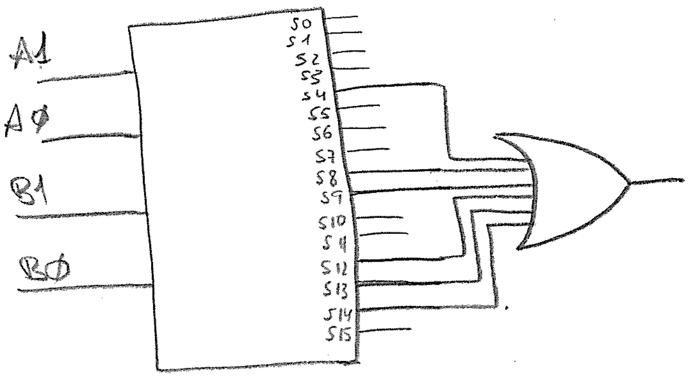

## * Con puertas NOR (sumas negadas)

$$S = \overline{\overline{\overline{(B1+B0)} + \overline{(\bar{A1}+B0)} + \overline{(\bar{A1}+B1)}}}$$

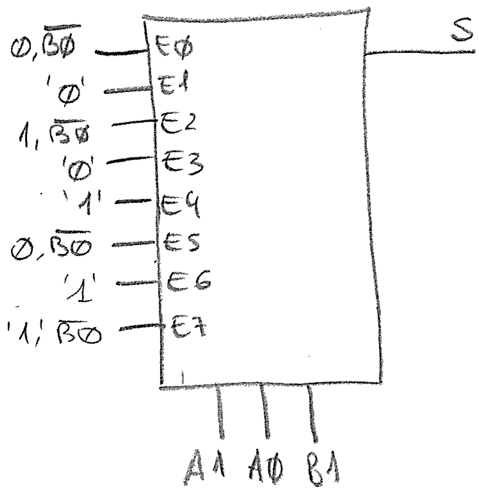



Diferente de la solución propuesta!!

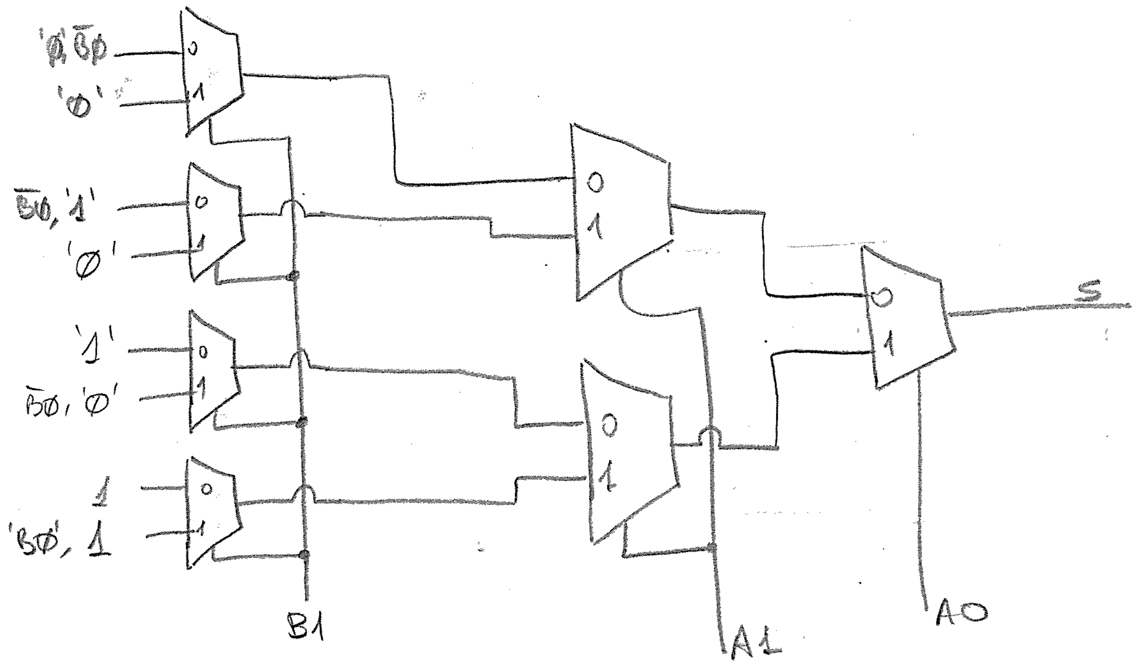
## * Decod. y puerta OR



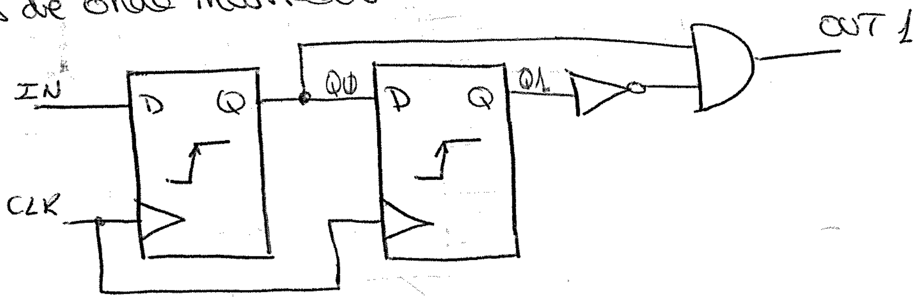

## * Multiplexores e inversores

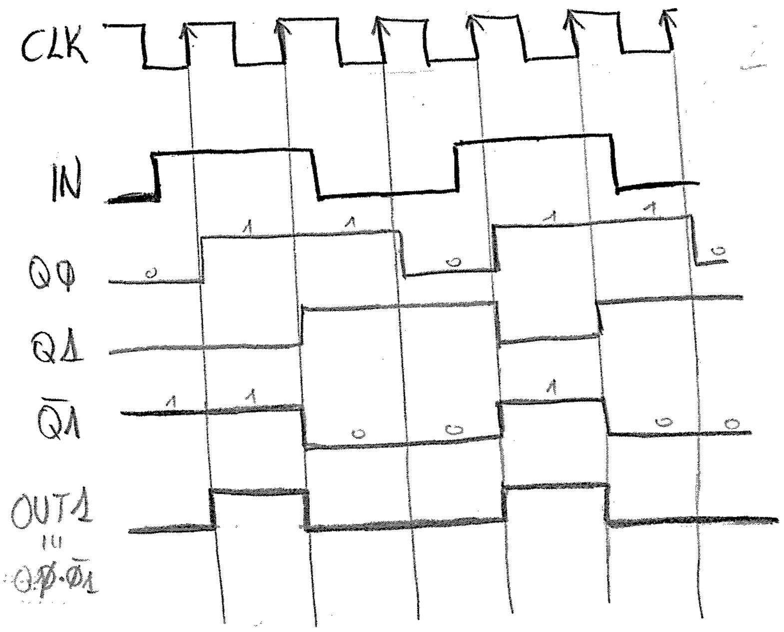


* Mux '2a1'




d) Para el circuito de la figura, dibujar la evolución de la señal OUT 1, para las formas de onda mostradas



# Examen Junio 2010 → Ejercicio 3 (Sólo parte A)

3.A) Para la tabla de verdad dada

a) Expresar la función F según la 1ª forma canónica
b) " " " F " " 2ª " "
c) Función mínima como producto de sumos
d) Implementación de función mínima con NAND
e) " " " " " Decod.
d) " " " " con Mux.

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

a) 1ª forma canónica (suma de productos)

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

b) 2ª forma canónica (producto de sumas)

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

**NOTA!!** Cómo formular la primera y la 2ª forma canónica

**1ª FORMA**: Cogemos los '1' de la tabla de verdad
Escribimos (A·B·C·D) y negamos los que no valgan 1

**2ª FORMA**: Cogemos los '0' de la tabla de verdad
Escribimos (A+B+C+D) y negamos los que no valgan 0.

c) Función mínima como producto de sumas ⇒ Cogemos los ceros!!

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

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

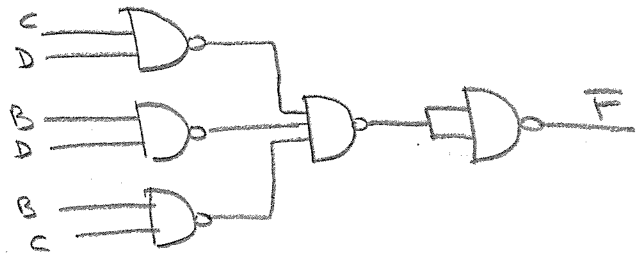
Como producto de sumas

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

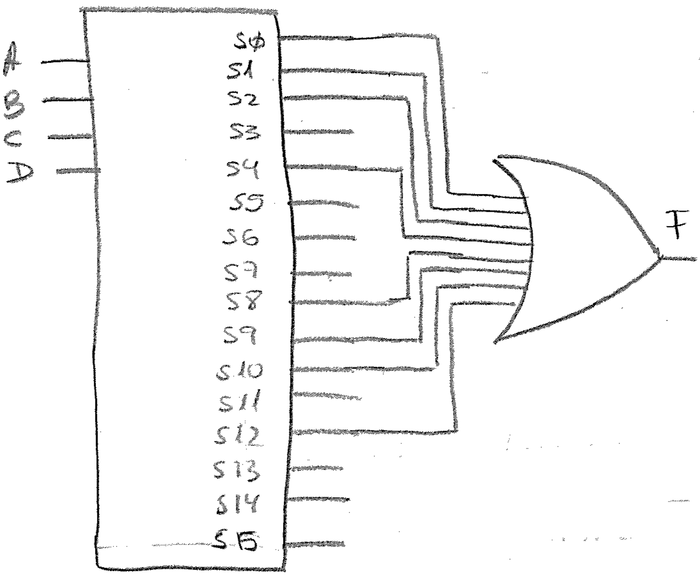
d) Implementación NAND = Productos negados

$\bar{C}+\bar{D}=\overline{CD}$ ; $\bar{B}+\bar{D}=\overline{BD}$ ; $\bar{B}+\bar{C}=\overline{BC}$ $\Rightarrow$ Por tanto:

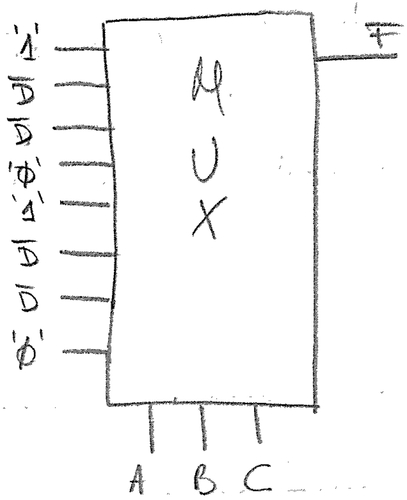
$$F=(\bar{C}+\bar{D})(\bar{B}+\bar{D})(\bar{B}+\bar{C})=\overline{CD}\cdot\overline{BD}\cdot\overline{BC}=\overline{\overline{\overline{CD}\;\overline{BD}\;\overline{BC}}}$$




e) Con decodificador

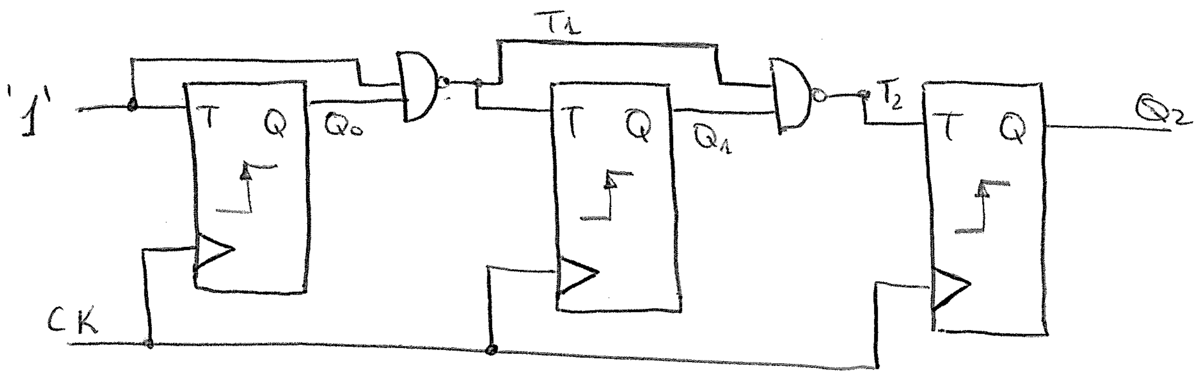



f) Con MUX

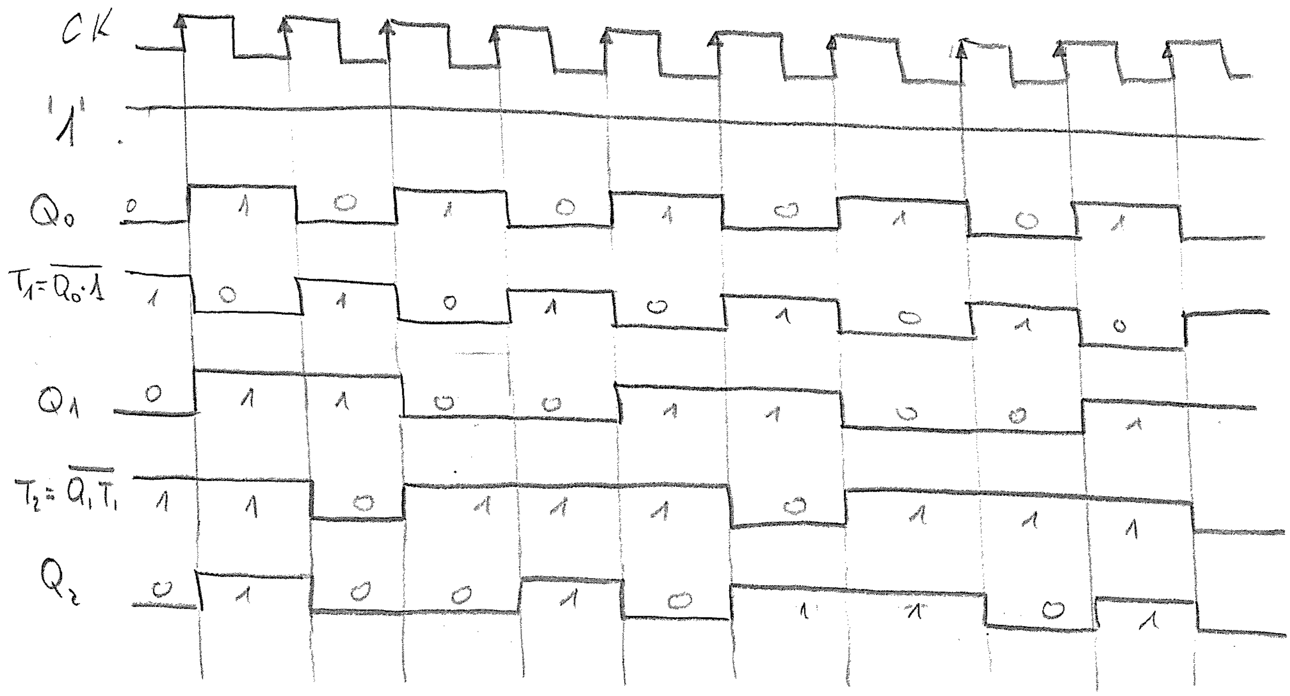


EXAMEN JUNIO 2004 : Problema 3

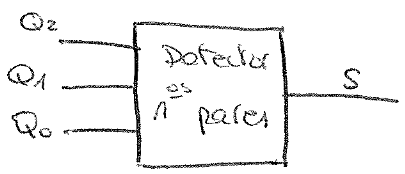
a) Dado el circuito de la figura y suponiendo que todos los biestables se inicializan a 'Ø', se pide dibujar la evolución de las señales $Q_0$, $T_1$, $Q_1$, $T_2$ y $Q_2$ durante 10 ciclos del reloj




Suponemos que los biest. se activan por flanco de subida del reloj (CK)




b) $Q_0$, $Q_1$, $Q_2$ del circuito anterior se consideran entradas de un nuevo circuito combinacional a diseñar. Interpretando dichas señales como un nº de 3 bits ($Q_2$ MSB, $Q_0$ LSB). Su función es detectar los nº pares poniendo a '1' la salida S para nº pares y a 'Ø' para los impares. Se pide la tabla de verdad de dicho circuito, su 1ª forma canónica y la función minimizada.




Nota: Considerese el Ø como cond. indif. de salida (no importa contarlo como par o impar)

b) *Tabla de verdad del circuito:

| $Q_2$ | $Q_1$ | $Q_0$ | S |
|---|---|---|---|
| 0 | 0 | 0 | X |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

* 1ª forma canónica

$$S = \bar{Q}_2 Q_1 \bar{Q}_0 + Q_2 \bar{Q}_1 \bar{Q}_0 + Q_2 Q_1 \bar{Q}_0$$

Mapa de Karnaugh

| $Q_0$ \ $Q_2Q_1$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | X | 1 | 1 | 1 |
| 1 | ∅ | ∅ | ∅ | ∅ |

Función minimizada $\boxed{S = \bar{Q}_0}$

EXAMEN SEPT. 2004: Problema 3 (sólo ap. A)

A) Siendo $A_1A_0$ y $B_1B_0$ dos nº binarios, se quiere obtener la lógica combinacional que pone a '1' la salida 'Y' cuando los dos nºs son iguales.

Se pide: A.1) Representar 'Y' en la 1ª forma canónica

A.2) Obtener la mínima expresión de 'Y' como suma de productos y como producto de sumas

| $A_1$ | $A_0$ | $B_1$ | $B_0$ | Y |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

* 1ª forma canónica:

$$Y = \bar{A}_1\bar{A}_0\bar{B}_1\bar{B}_0 + \bar{A}_1 A_0 \bar{B}_1 B_0 + A_1 \bar{A}_0 B_1 \bar{B}_0 + A_1 A_0 B_1 B_0$$

* Mapa Karnaugh:

| $B_1B_0$ \ $A_1A_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | ∅ | ∅ | ∅ |
| 01 | ∅ | 1 | ∅ | ∅ |
| 11 | ∅ | ∅ | 1 | ∅ |
| 10 | ∅ | ∅ | ∅ | 1 |

* Y como suma de prod.:

$$Y = \bar{A}_1\bar{A}_0\bar{B}_1\bar{B}_0 + \bar{A}_1 A_0 \bar{B}_1 B_0 + A_1 \bar{A}_0 B_1 \bar{B}_0 + A_1 A_0 B_1 B_0$$

* Y como producto de sumas

$$Y = (\bar{A}_1 + B_1)(A_1 + \bar{B}_1)(\bar{A}_0 + \bar{B}_0)$$
$$(A_0 + \bar{B}_0)$$

EXAMEN Febrero 2006: Prob. 3 (sólo ap. a) b))

a) Diseñar un circuito mínimo con puertas NAND que, a partir de 4 bits que representan los dígitos del '0' al '9', genere un '1' para aquellos combinaciones de entrada que deberán encender el display de 7 segmentos

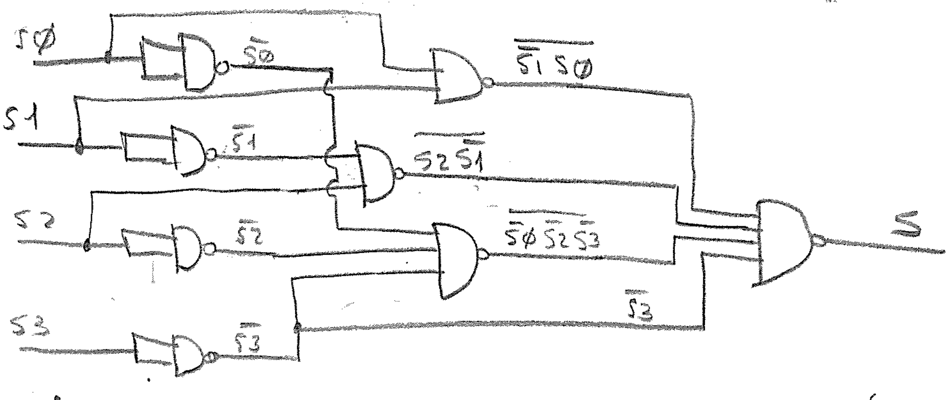
| S3 | S2 | S1 | S0 | S |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Ø |
| 0 | 0 | 0 | 1 | Ø |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | Ø |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | X |
| 1 | 0 | 1 | 1 | X |
| 1 | 1 | 0 | 0 | X |
| 1 | 1 | 0 | 1 | X |
| 1 | 1 | 1 | 0 | X |
| 1 | 1 | 1 | 1 | X |

| S1S0 \ S3S2 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | Ø | 1 | X | 1 |
| 01 | Ø | 1 | X | 1 |
| 11 | 1 | Ø | X | X |
| 01 | 1 | 1 | X | X |

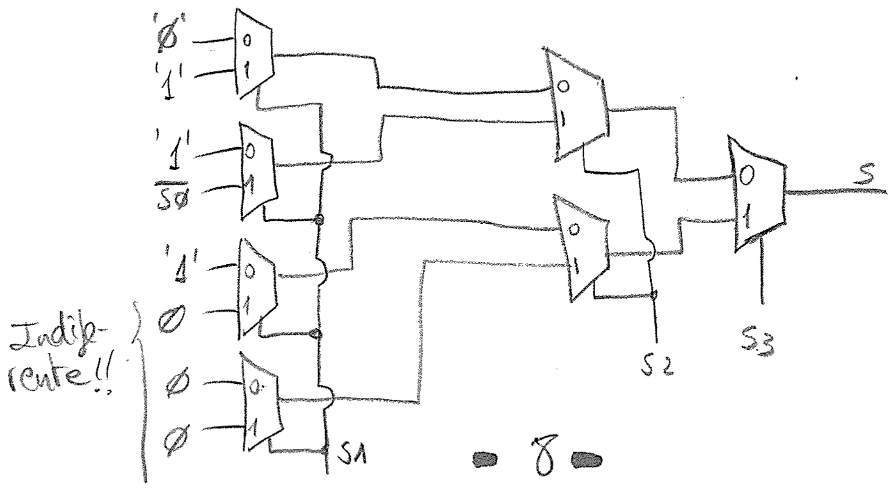
$$S = \underbrace{S3}_{A} + \underbrace{S2\,\overline{S1}}_{B} + \underbrace{\overline{S1}\,S0}_{C} + \underbrace{\overline{S3}\,\overline{S2}\,\overline{S0}}_{D}$$

Puertas NAND = productos negados

$$S = A + B + C + D = \overline{\bar{A}\,\bar{B}\,\bar{C}\,\bar{D}} = \overline{\overline{S3}\cdot\overline{S2\,\overline{S1}}\cdot\overline{\overline{S1}\,S\emptyset}\cdot\overline{\overline{S3}\,\overline{S2}\,\overline{S\emptyset}}}$$




b) Realizar la misma función con MUX '2 a 1'



Prob. 3 (excepto ap. d))

# EXAMEN JUNIO 2006

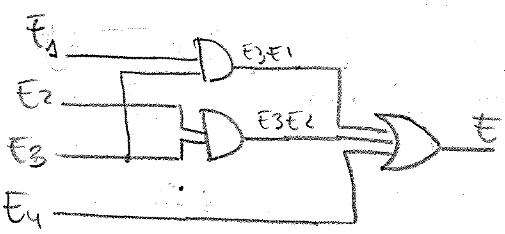
Sistema electrónico de 5 señales $E_1, E_2, E_3, E_4, CLK$. Las señales E codifican la calificación de un examen (0-10 sin decimales). La señal CLK es de un reloj de 1 MHz de frec. Realizar un sistema electrónico que realice:

a) Genere una señal indicando si se ha aprobado (>5)

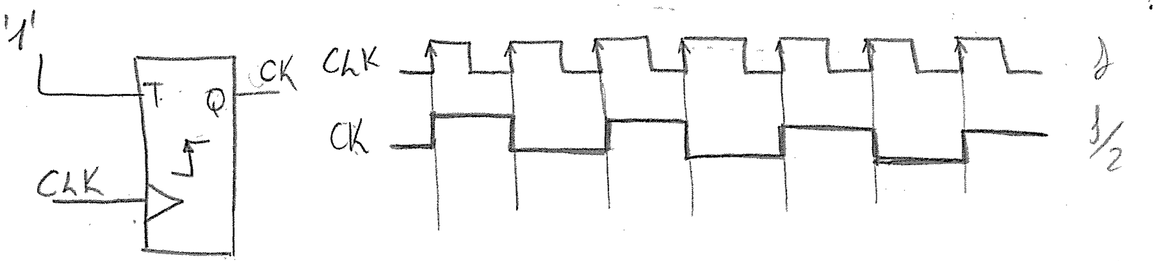
| $E_4$ | $E_3$ | $E_2$ | $E_1$ | $E$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | ∅ |
| 0 | 0 | 0 | 1 | ∅ |
| 0 | 0 | 1 | 0 | ∅ |
| 0 | 0 | 1 | 1 | ∅ |
| 0 | 1 | 0 | 0 | ∅ |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | X |
| 1 | 1 | 0 | 0 | X |
| 1 | 1 | 0 | 1 | X |
| 1 | 1 | 1 | 0 | X |
| 1 | 1 | 1 | 1 | X |

| $E_2E_1$ \ $E_4E_3$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | ∅ | ∅ | X | 1 |
| 01 | ∅ | 1 | X | 1 |
| 11 | ∅ | 1 | X | X |
| 10 | ∅ | 1 | X | 1 |

$$E = E_4 + E_3E_1 + E_3E_2$$




b) A partir de la señal del reloj, se quiere obtener otra con $f = 500\,KHz$




La frec. queda div. entre 2

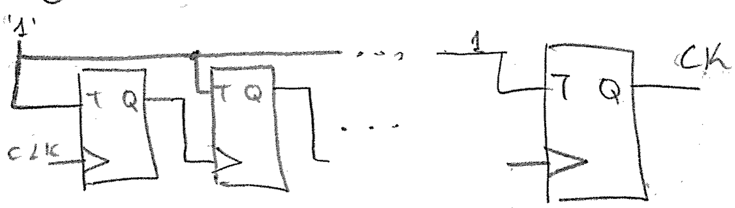
* Si $f_{CLK} = 1\,GHz$

$$f = 500\,KHz = \frac{1000000}{2^n} \longrightarrow 2^n = 2000$$

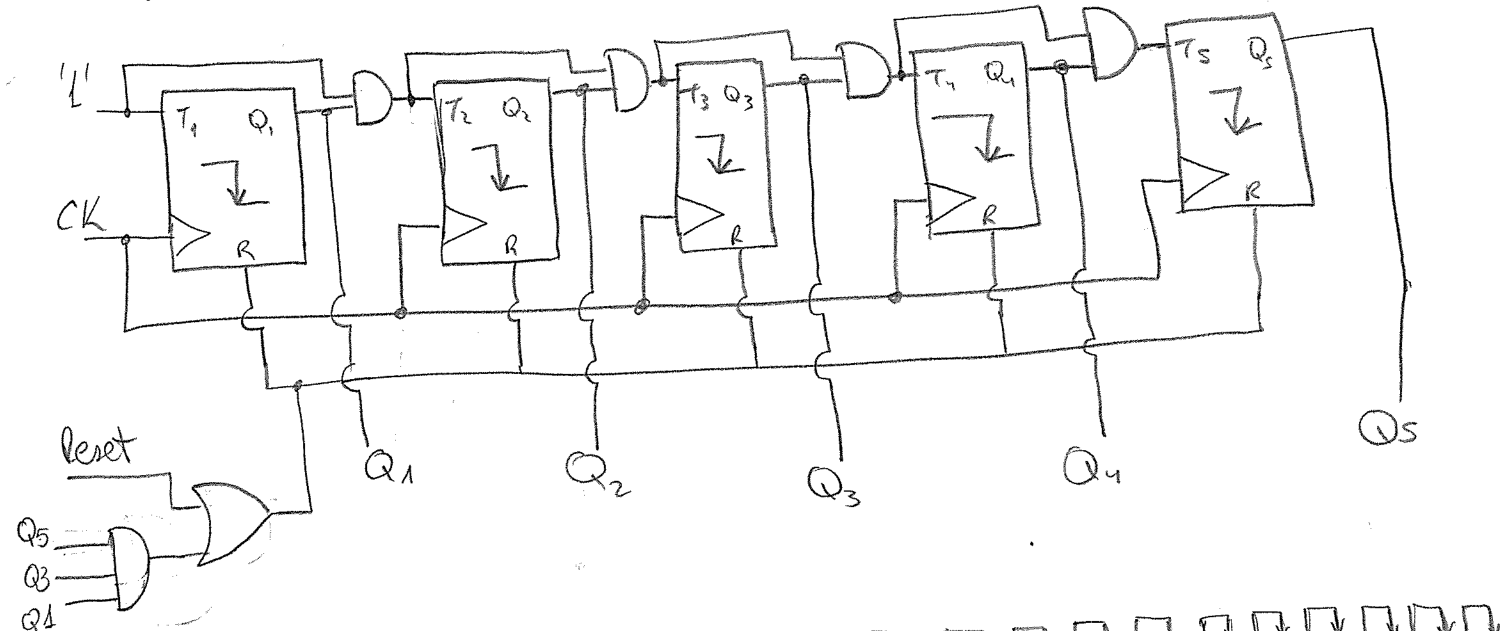
$$n \log 2 = \log 2000, \quad n \approx 11$$

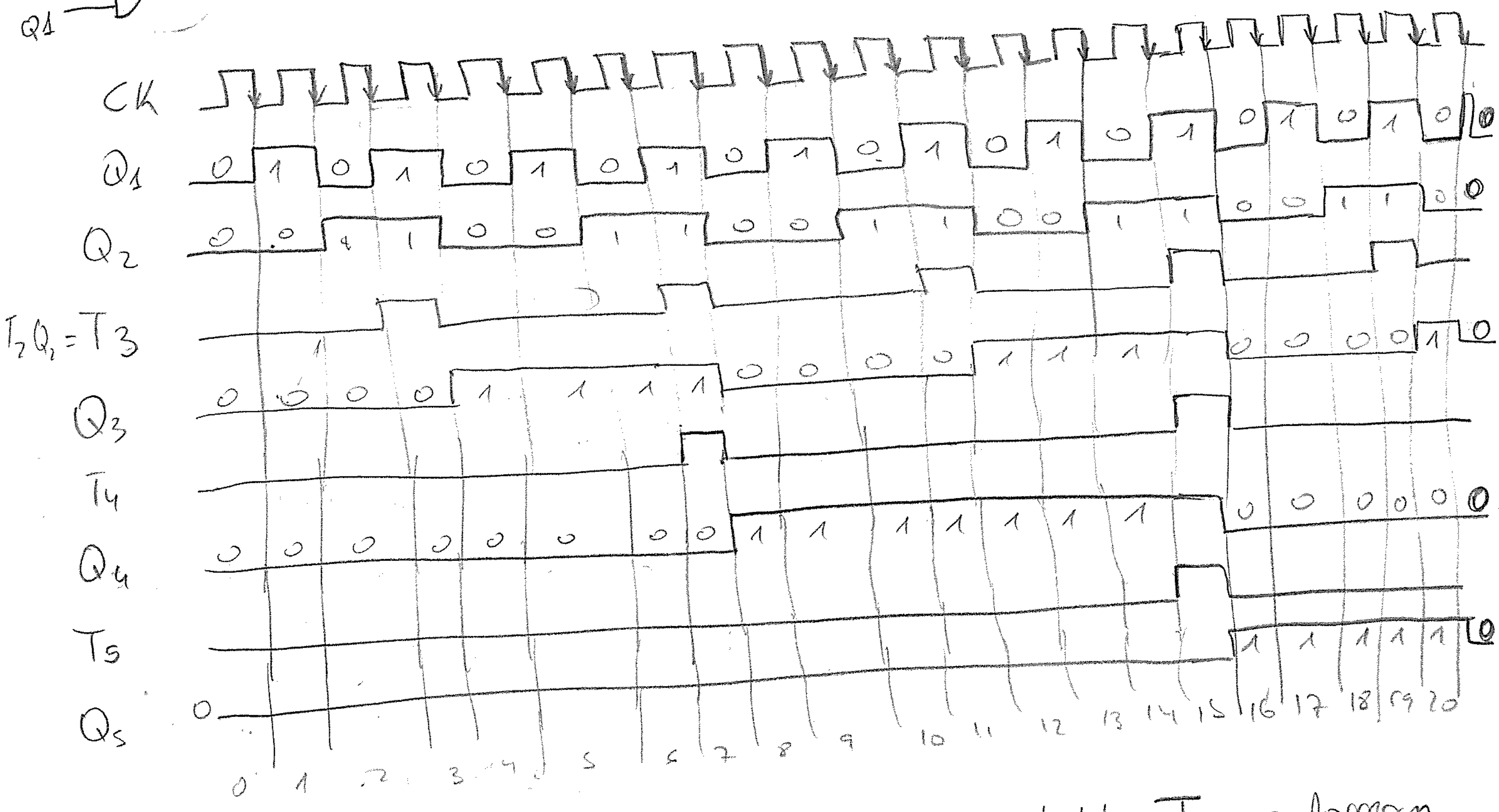
$$\Rightarrow f = 488 \simeq 500\,KHz$$

Habría que montar un circuito de 11 biestables en serie $\Rightarrow$



c) A continuación se quiere contar de forma síncrona ascendente hasta flancos de bajada de la señal generada de 500KHz



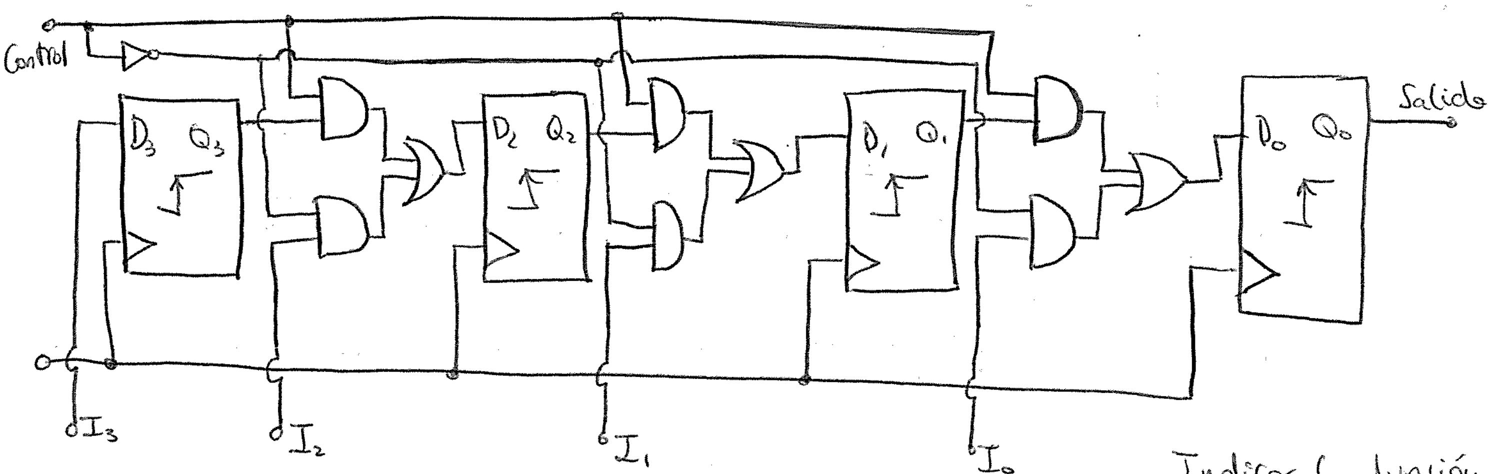

El circuito está formado por un conjunto de 5 biestables T que forman un contador de hasta $2^5$ (32 dígitos)

Como sólo queremos contar hasta 20 ordenamos mediante las puertas AND y OR que los biestables se reseteen al llegar a 21

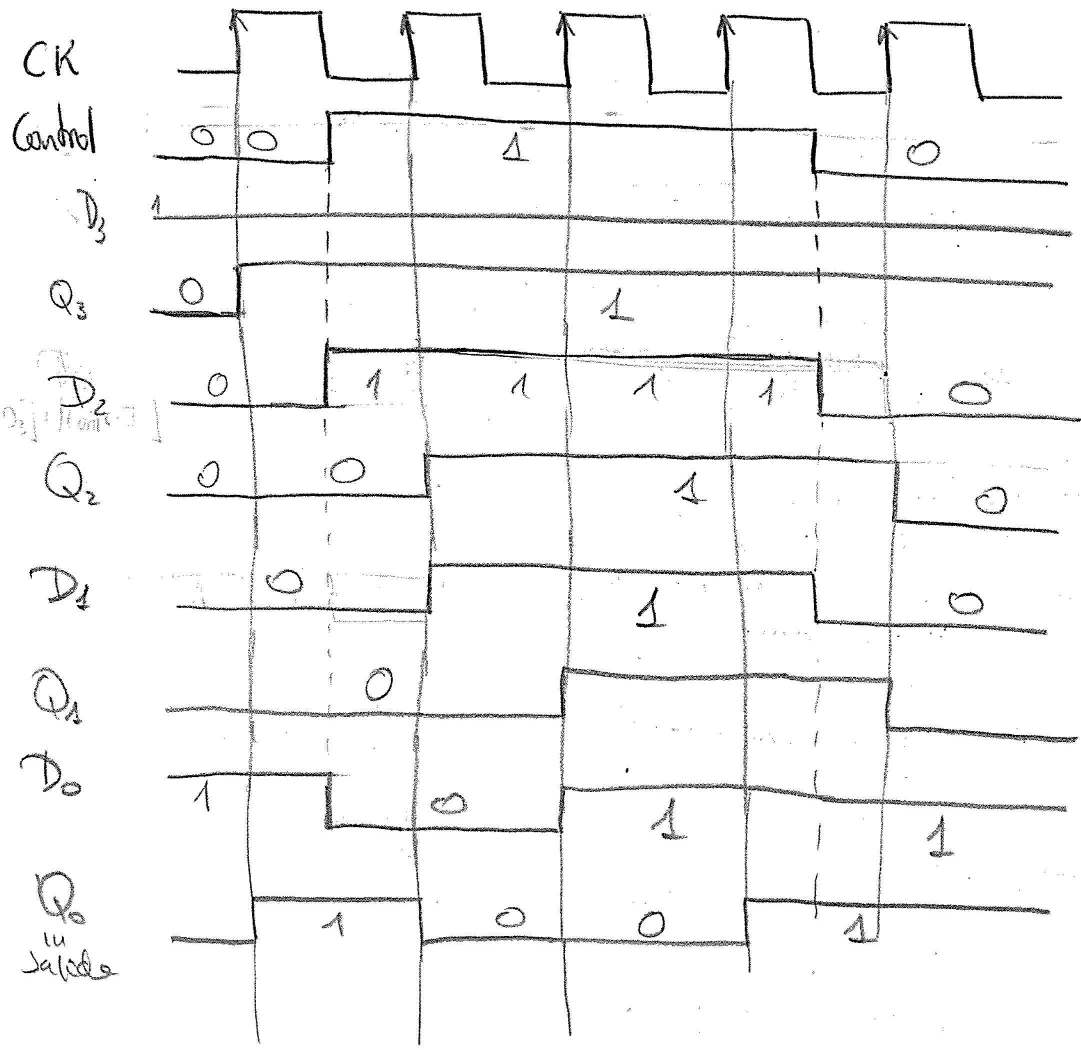
Recordamos que este método puede generar problemas

## Examen SEPT. 2006 → Ejercicio 2

Para el circuito de la figura y para las señales CK y control, obtener el cronograma de $D_0$, $Q_0$, $D_1$, $Q_1$, $D_2$, $Q_2$, $D_3$, $Q_3$, sabiendo que en el instante inicial, cuando se produce el flanco activo del reloj, los valores son: $I_0 = 1$, $I_1 = 0$, $I_2 = 0$, $I_3 = 1$




Indicar la función del circuito y su nombre




$$Q_2 = Control \cdot Q_3 + \underbrace{\overline{Control} \cdot \overset{0}{I_2}}_{0}$$

$$D_1 = Control \cdot Q_2 + \underbrace{\overline{Control} \cdot \overset{0}{I_1}}_{0}$$

$$D_0 = Control \cdot Q_1 + \overline{Control} \cdot \overset{1}{I_0}$$

Es un circuito que convierte un dato que se introduce en paralelo, en una salida seriada (entrada paralelo – salida serie)

# Examen SEPT. 2007 Ejercicio 3

Se desea diseñar un circuito que tiene como entradas dos nº binarios de dos bits cada uno. La salida valdrá 1 cuando dichos num. sean iguales y '0' en caso contrario.

a) Tabla de verdad y 1ª forma canónica  Números $A_1A_0$, $B_1B_0$

| $A_1$ | $A_0$ | $B_1$ | $B_0$ | S |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | Ø |
| 0 | 0 | 1 | 0 | Ø |
| 0 | 0 | 1 | 1 | Ø |
| 0 | 1 | 0 | 0 | Ø |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | Ø |
| 0 | 1 | 1 | 1 | Ø |
| 1 | 0 | 0 | 0 | Ø |
| 1 | 0 | 0 | 1 | Ø |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | Ø |
| 1 | 1 | 0 | 0 | Ø |
| 1 | 1 | 0 | 1 | Ø |
| 1 | 1 | 1 | 0 | Ø |
| 1 | 1 | 1 | 1 | 1 |

1ª forma canónica

$$S = \bar{A}_1\bar{A}_0\bar{B}_1\bar{B}_0 + \bar{A}_1A_0\bar{B}_1B_0 + A_1\bar{A}_0B_1\bar{B}_0 + A_1A_0B_1B_0$$

b) Función lógica minimizada como producto de sumas

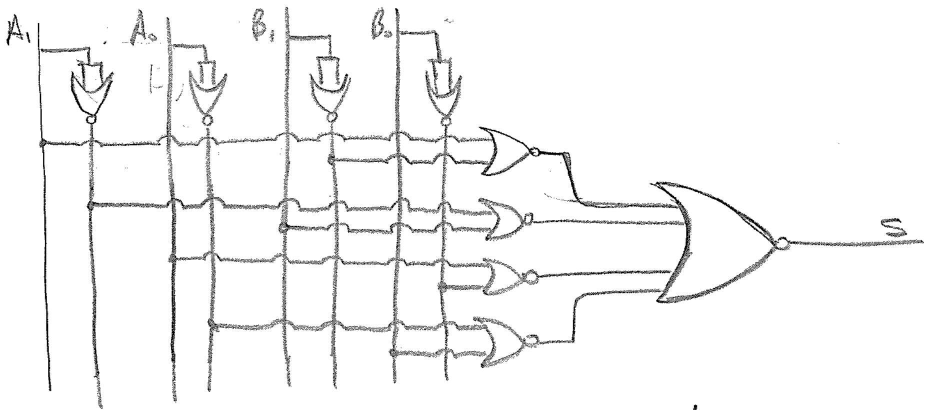
Mapa de Karnaugh

| $B_1B_0$ \ $A_1A_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | Ø | Ø | Ø |
| 01 | Ø | 1 | Ø | Ø |
| 11 | Ø | Ø | 1 | Ø |
| 10 | Ø | Ø | Ø | 1 |

$$S = (\bar{A}_0 + B_0)(\bar{A}_1 + B_1)(A_0 + \bar{B}_0)(A_1 + \bar{B}_1)$$

c) Expresar la función lógica minimizada resultante del ap. b) únicamente usando puertas NOR → sumas negadas

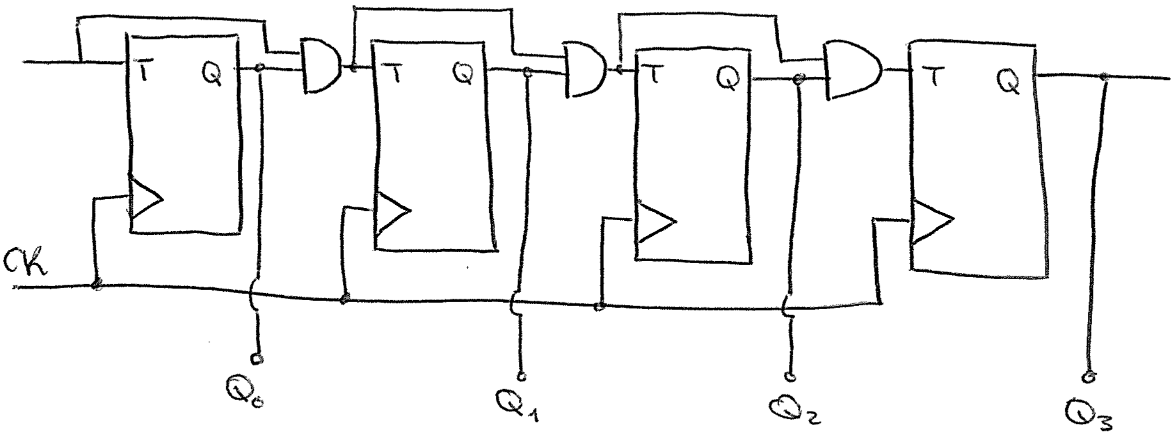
$$S = (\bar{A}_0 + B_0)(A_0 + \bar{B}_0)(\bar{A}_1 + B_1)(A_1 + \bar{B}_1) = \overline{\overline{(\bar{A}_0 + B_0)} + \overline{(A_0 + \bar{B}_0)} + \overline{(\bar{A}_1 + B_1)} + \overline{(A_1 + \bar{B}_1)}}$$



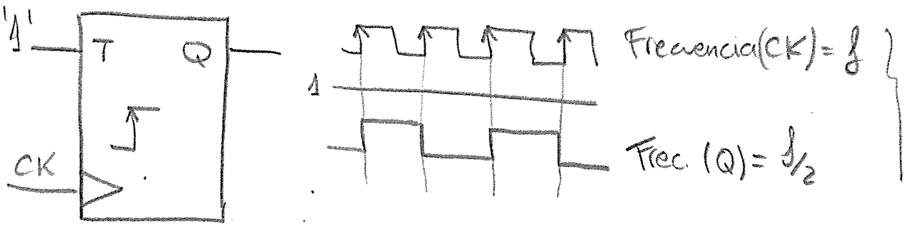

d) Dibujar el circuito resultante

# Examen SEPT 2007 Ejercicio 4

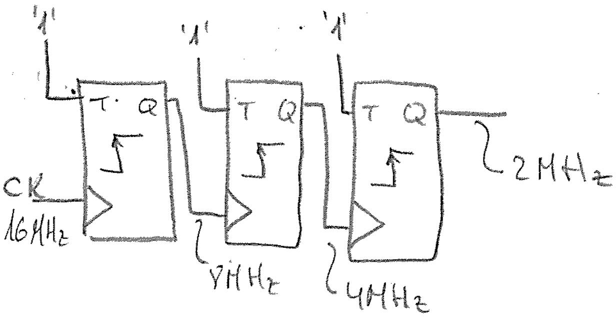
a) Diseñar un circuito binario síncrono capaz de contar de 0 a 15



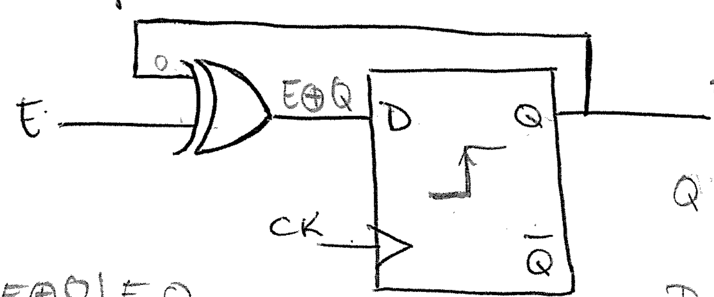

b) Diseñar un circuito con entrada del reloj 16 MHz, que genere una señal de 2 MHz

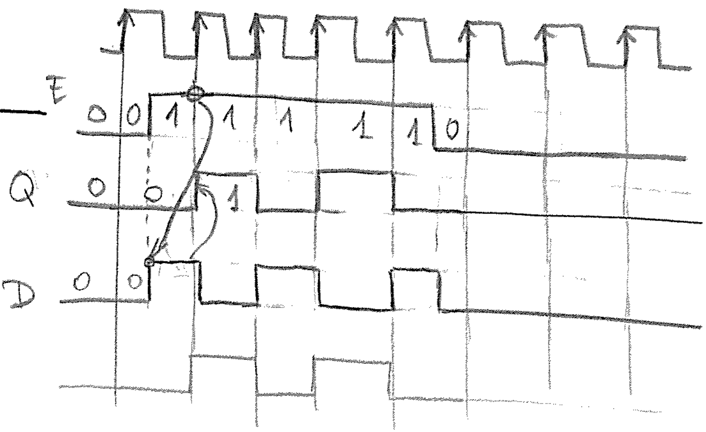



Como vemos, un biestable T divide la frecuencia del reloj entre 2.
Como queremos dividirla entre 8, esto es, $2^3$, necesitamos 3 biestables conectados según se muestra.

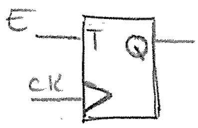



c) Explicar a qué circuito es equivalente el siguiente esquema. Justifíquese la respuesta



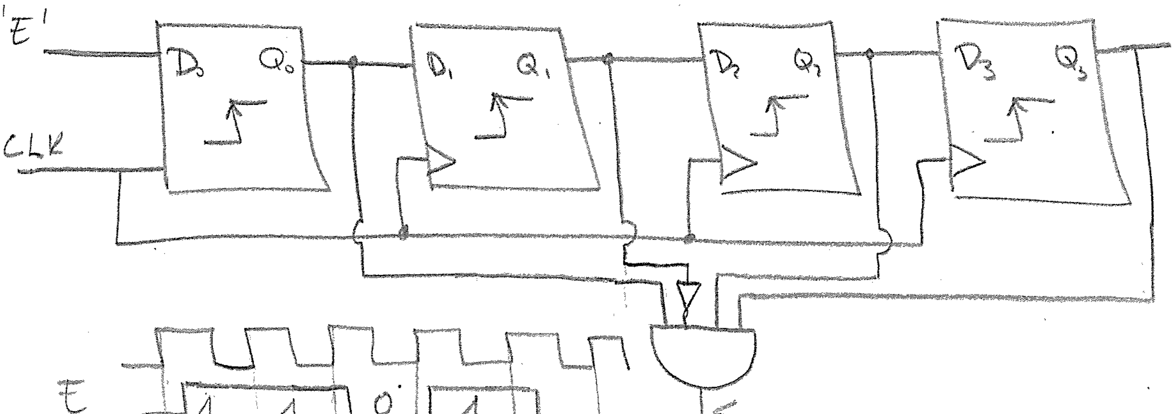

| E⊕Q | E Q |
|---|---|
| 0 | 0 0 |
| 1 | 0 1 |
| 1 | 1 0 |
| 0 | 1 1 |

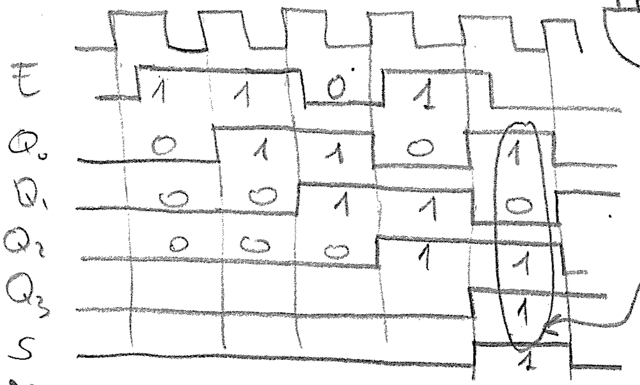


# Examen FEB 2008 Problema 4

a) Diseñar un circuito que tenga como entradas una señal de reloj 'CLK' y una línea de bits serie 'E', y que dé como salida 'S' un pulso de 1 ciclo del reloj de duración al reconocer la secuencia de bits 1101.
$2^3 2^2 \quad 2^0 = 13$

Lo más sencillo es un registro de desplazamiento tipo SIPO.
Nos obligan a que la entrada sea seriada por tanto elegimos un registro SERIAL INPUT, PARALEL OUTPUT o "SIPO".



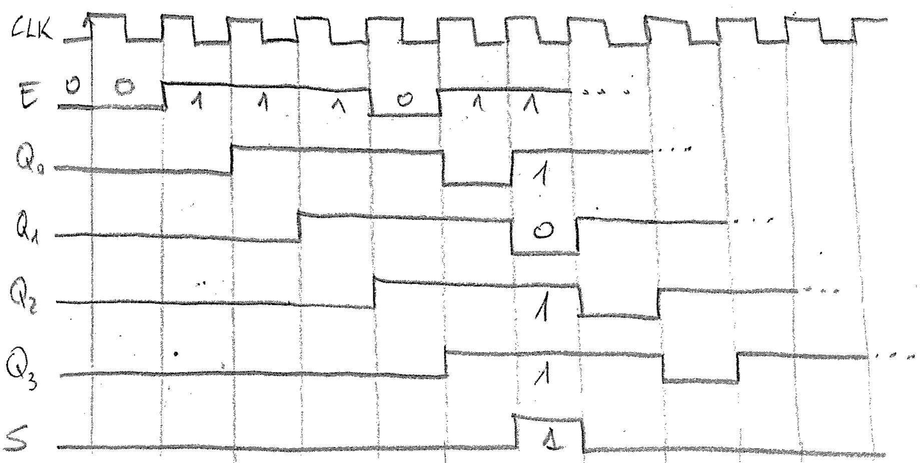




Secuencia detectada ≡ S='1' durante ese ciclo del reloj

b) Dibujar un cronograma para el circuito del apartado a) con la secuencia de entrada 0011101 donde se observen las señales 'E', 'S', 'CLK' y las señales internas que se consideren significativas

Dibujaremos las señales E, S, CLK y además $Q_0$ $Q_1$ $Q_2$ $Q_3$

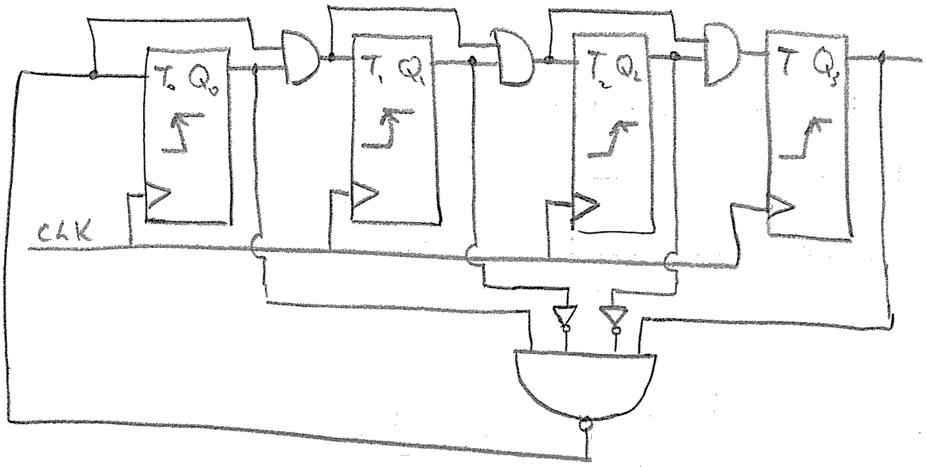
NOTA: se supondrá que E cambia de 0 a 1 en cada flanco del reloj

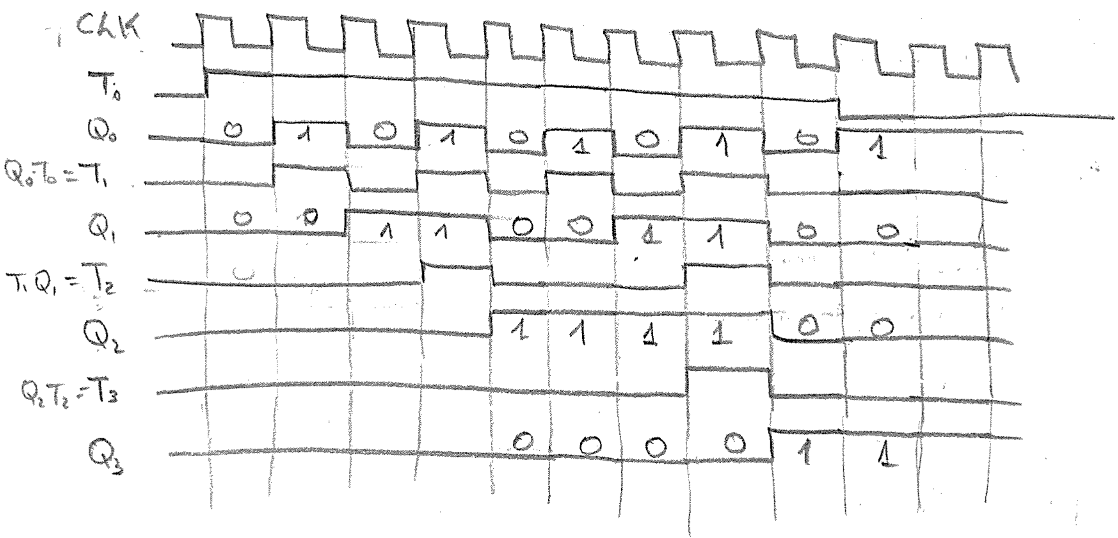


# Examen FEB. 2008: Ejercicio 5

a) Diseñar un circuito síncrono que cuente de 0 a 9 y al llegar a 9 se pare

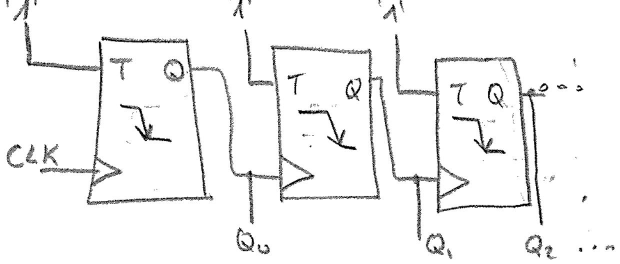
9 = '1001'




b) Explicar la diferencia entre un contador síncrono y uno asíncrono.

En un contador síncrono el reloj es común a todos los biestables, sin embargo, en uno asíncrono la señal del reloj sólo entra en el primer biestable, de la forma que se indica.

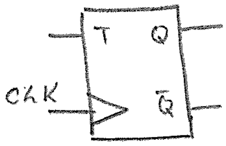



Ha de ser activo por flancos de bajada porque si no, contará hacia atrás

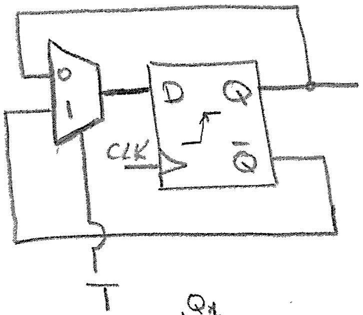
# Examen FEB 2009 Cuestión 3

**a)** Construir un contador ascendente síncrono de 4 bits, usando flip-flops D como elemento básico

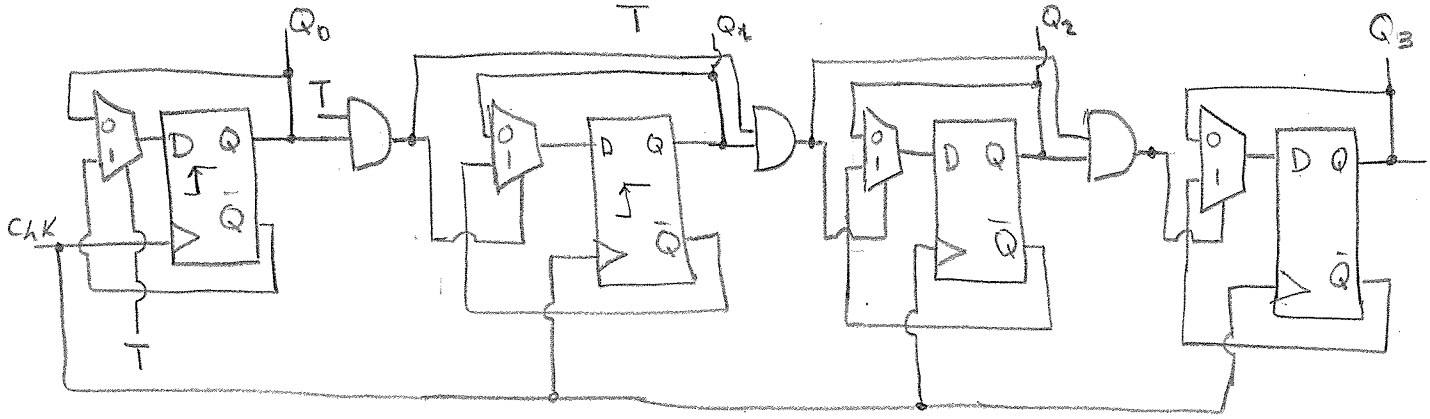
Sabemos que:




$\Longleftrightarrow$




Por tanto el contador de 4 bits con flip-flops D quedaría $\Longrightarrow$




**b)** Las salidas del circuito del apartado a) se conectan a la entrada de un nuevo circuito digital que tiene 2 salidas S1 y S2

- La salida S1 valdrá '1' cuando el nº binario representado por $Q_3Q_2Q_1Q_0$ sea impar y primo, y 'Ø' en caso contrario
- La salida S2 valdrá 1 cuando la entrada valga 0,1,2,8,10 y 15, y 'Ø' en el caso contrario.

Se pide: b.1) Función lógica minimizada para S1 y S2

b.2) Implementar S1 usando un MUX

b.3) Implementar S2 con puertas lógicas y decodificadores.

| $Q_3$ | $Q_2$ | $Q_1$ | $Q_0$ | S1 | S2 |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 |

**b.1)** Mapa de Karnaugh para S1

| $Q_1Q_0$ \ $Q_3Q_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | Ø | Ø | Ø | Ø |
| 01 | 1 | 1 | 1 | Ø |
| 11 | 1 | 1 | Ø | 1 |
| 10 | Ø | Ø | Ø | Ø |

$$\boxed{S1 = \bar{Q}_3 Q_0 + Q_2 \bar{Q}_1 Q_0 + \bar{Q}_2 Q_1 Q_0}$$

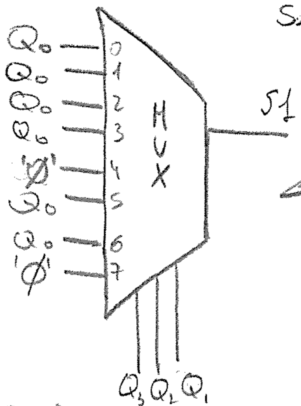
Mapa de Karnaugh para S2

| $Q_1Q_0$ \ $Q_3Q_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | ∅ | ∅ | 1 |
| 01 | 1 | ∅ | ∅ | ∅ |
| 11 | ∅ | ∅ | 1 | ∅ |
| 10 | 1 | ∅ | ∅ | 1 |

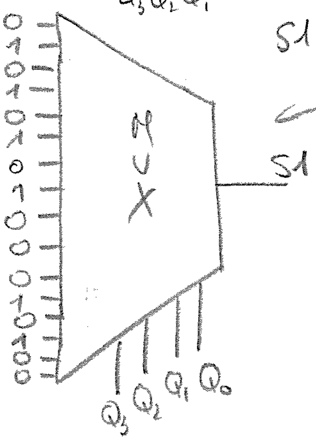
$$\boxed{S2 = \overline{Q_3}\,\overline{Q_2}\,\overline{Q_1} + \overline{Q_2}\,\overline{Q_0} + Q_3 Q_2 Q_1 Q_0}$$

b.2)

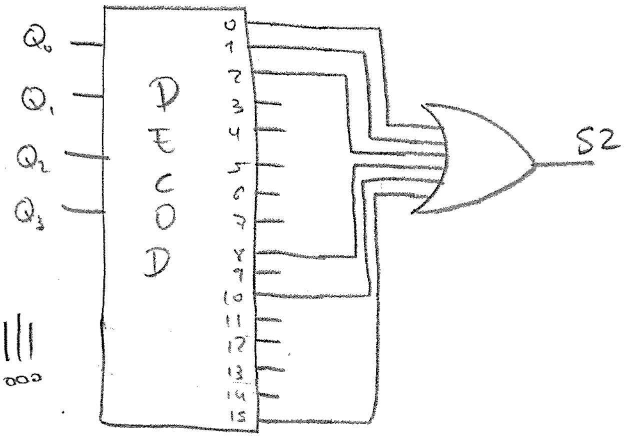
S1 con MUX + inversores




S1 sólo con MUX !!!

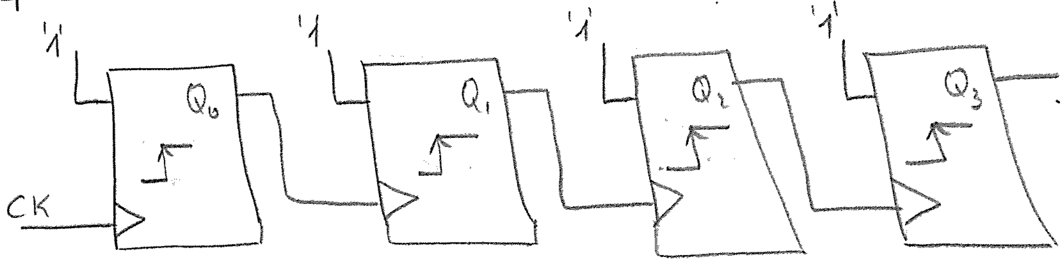


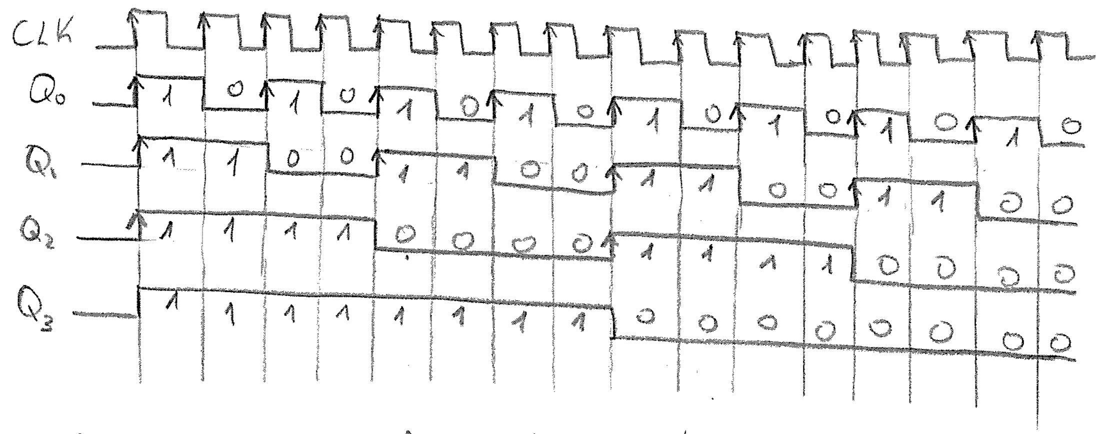

b.3) S2 con decod. y puertas




## Examen JUN. 2008 Ejercicio 3

a) Dibujar el cronograma de las señales $Q_3 Q_2 Q_1 Q_0$ mostrados en el circuito de la figura, para 16 ciclos del reloj, suponiendo que inicialmente las salidas de todos los flip-flops están a '∅'. ¿Qué función desempeña este circuito? ¿Con qué nombre lo denominarías?






Observamos que el circuito cuenta hacia atrás.
Es un contador asíncrono descendiente

**b)** La salida del circuito del ap. ⓐ se conecta a la entrada de un nuevo circuito digital que tiene 2 salidas S1 y S2.
- La salida S1 valdrá '1' cuando el nº binario $Q_3Q_2Q_1Q_0$ sea múltiplo de 2 (sin considerar '0' como múltiplo) y '∅' en caso contrario
- La salida S2 valdrá 1 cuando la entrada valga 0, 1, 2, 8 y 10.

Se pide: b.1) Función lógica minimizada para S1 S2
b.2) Implementar S1 y S2 de 3 formas distintas

| $Q_3Q_2Q_1Q_0$ | S1 | S2 |
|---|---|---|
| 1 1 1 1 | ∅ | ∅ |
| 1 1 1 0 | 1 | ∅ |
| 1 1 0 1 | ∅ | ∅ |
| 1 1 0 0 | 1 | ∅ |
| 1 0 1 1 | ∅ | ∅ |
| 1 0 1 0 | 1 | 1 |
| 1 0 0 1 | ∅ | ∅ |
| 1 0 0 0 | 1 | 1 |
| 0 1 1 1 | ∅ | ∅ |
| 0 1 1 0 | 1 | ∅ |
| 0 1 0 1 | ∅ | ∅ |
| 0 1 0 0 | 1 | ∅ |
| 0 0 1 1 | ∅ | ∅ |
| 0 0 1 0 | 1 | 1 |
| 0 0 0 1 | ∅ | 1 |
| 0 0 0 0 | ∅ | 1 |

b.1) Mapa Karnaugh S1

| $Q_1Q_0$ \ $Q_3Q_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | ∅ | 1 | 1 | 1 |
| 01 | ∅ | ∅ | ∅ | ∅ |
| 11 | ∅ | ∅ | ∅ | ∅ |
| 10 | 1 | 1 | 1 | 1 |

$$\boxed{S1 = Q_1\overline{Q}_0 + Q_2\overline{Q}_0 + Q_3\overline{Q}_0}$$

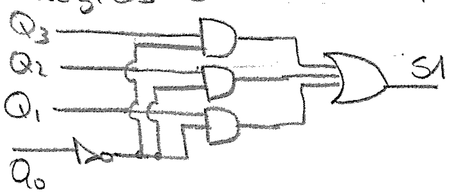
$$S1 = \overline{Q}_0(Q_1 + Q_2 + Q_3)$$

Mapa Karnaugh S2

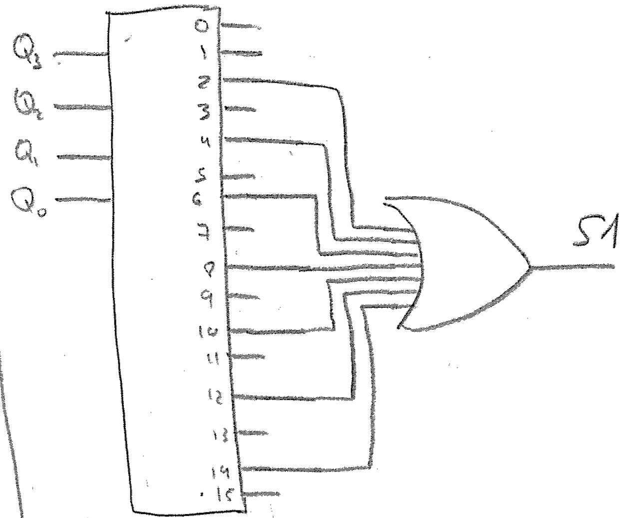
| $Q_1Q_0$ \ $Q_3Q_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | ∅ | ∅ | 1 |
| 01 | 1 | ∅ | ∅ | ∅ |
| 11 | ∅ | ∅ | ∅ | ∅ |
| 10 | 1 | ∅ | ∅ | 1 |

$$\boxed{S2 = \overline{Q}_3\overline{Q}_2\overline{Q}_1 + \overline{Q}_2\overline{Q}_0}$$

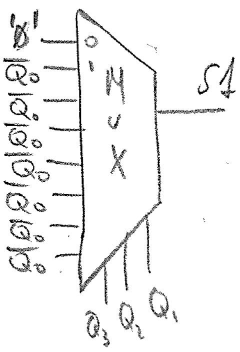
Implementación S1 con puertas lógicas e inversores



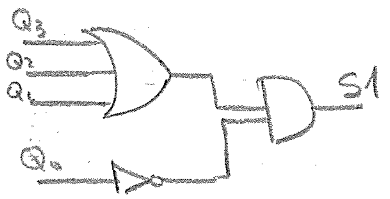

Implementación S1 con decodificador y puertas lógicas




Implementación S1 con MUX e inversores

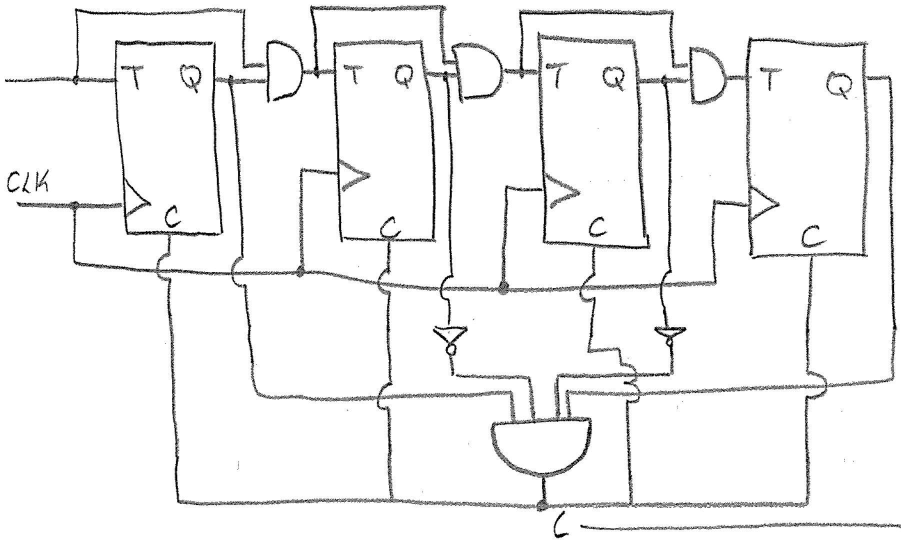



Con otras puertas



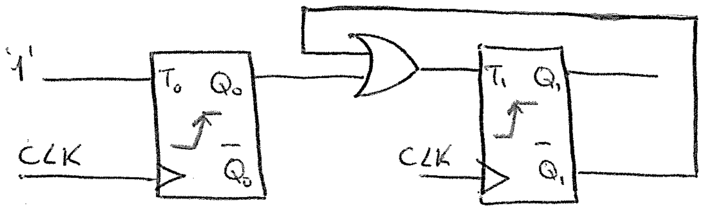

Examen SEPT. 2009 Problema 3 (ap. (a) NO ENTRA)

**b)** Diseña un contador síncrono ascendente que cuente de '0' a '9' y al llegar a 9 vuelva a empezar la cuenta en 0 (suponer que los biestables tienen una señal de entrada de inicialización, clear, que borra el biestable en el siguiente flanco activo del reloj tras ser activada)

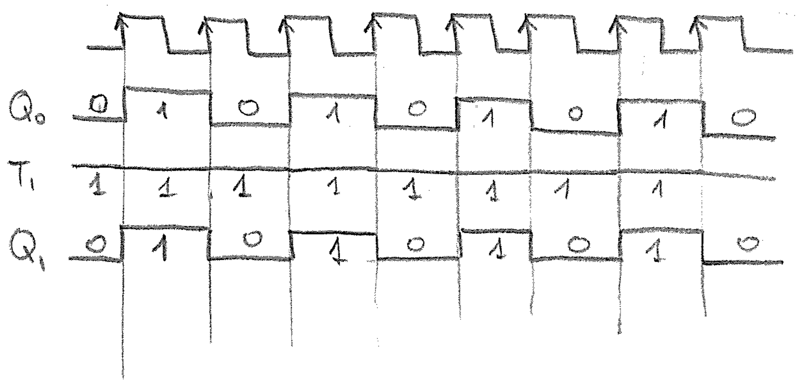



Esta señal se pone a 1 cuando $Q_3Q_2Q_1Q_0 \equiv 1001\,(9)$ y borra los biestables

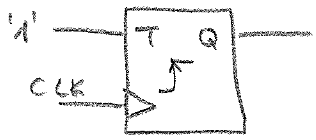
c) Dibujar el cronograma para el siguiente circuito durante 8 ciclos del reloj suponiendo que inicialmente $Q_0$ y $Q_1$ valen '0'




Simplificar el circuito anterior tras analizar el cronograma realizado

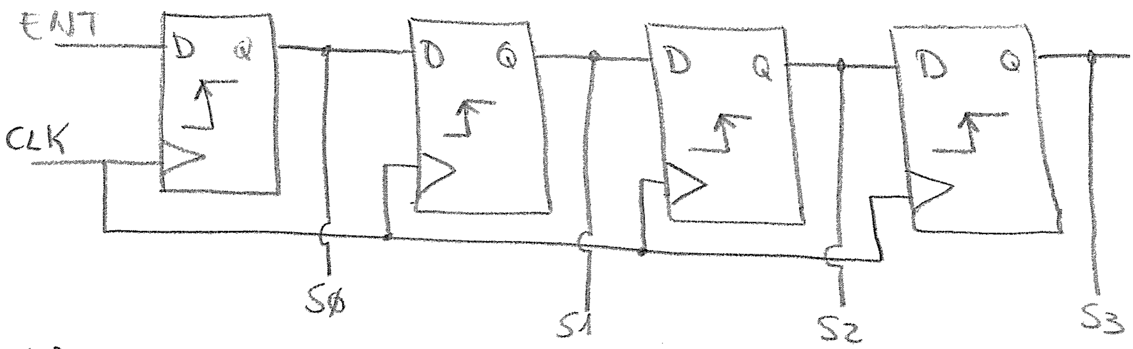



Como vemos $Q_0 \equiv Q_1$ por lo que podemos suprimir parte del circuito y nos quedaría:

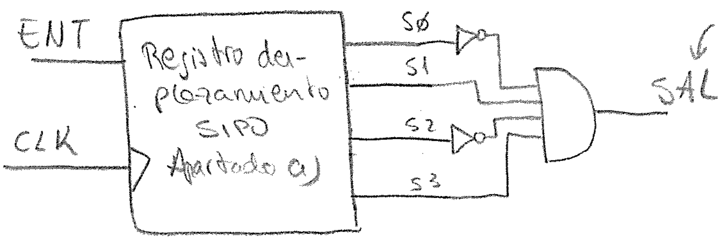



Examen SEPT. 2010 : Cuestión 4

a) Dibujar un esquema interno de un registro de desplazamiento de 4 bits de salida S3 S2 S1 S0 y una señal de entrada en serie ENT, síncrono con la señal de reloj CLK




b) A partir de este registro de desplazamiento, realizar un detector de la secuencia "1010" añadiendo la lógica combinacional necesaria y explicando su funcionamiento. El detector de secuencia deberá poner a '1' una señal de salida SAL cuando detecte la frecuencia '1010' por la entrada ENT



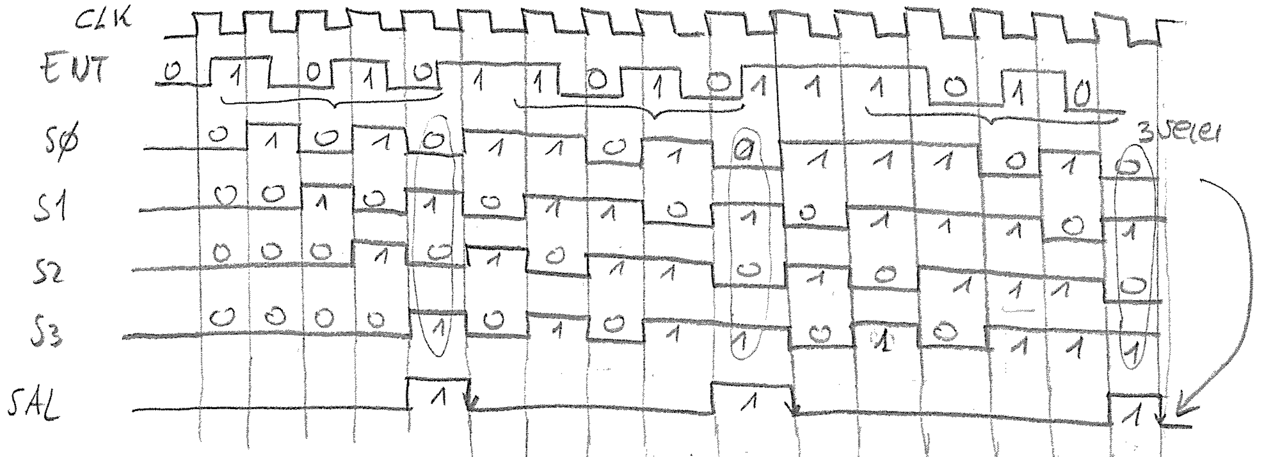

La salida SAL se pondrá a 1 cuando

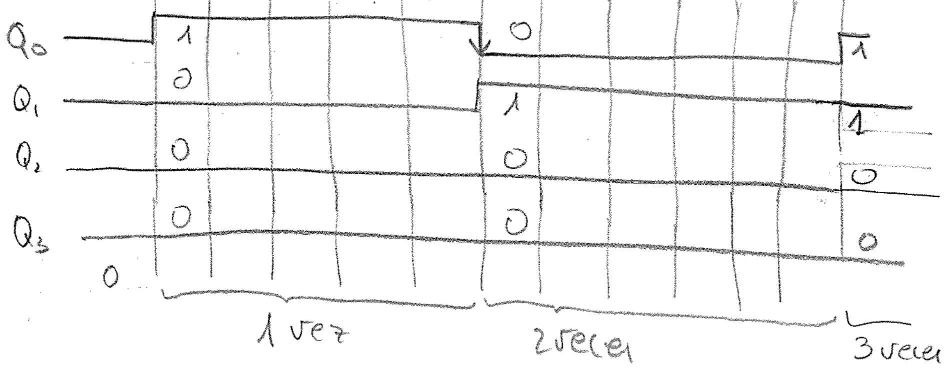
$$\left.\begin{aligned} S3 &= 1 \\ S2 &= 0 \\ S1 &= 1 \\ S\emptyset &= 0 \end{aligned}\right\} SAL = 1 \Rightarrow \text{Secuencia detectada}$$

Permanecerá a 1 durante 1 flanco del reloj

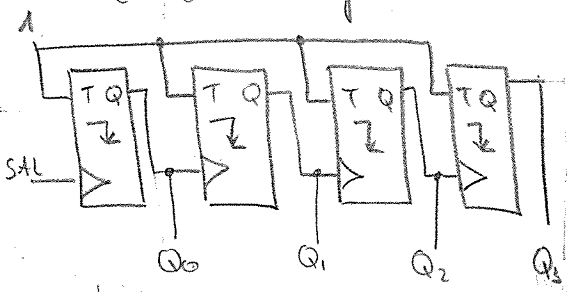
c) Se quiere contar cuántas veces se detectan dichas frecuencias en cada 16 ciclos del reloj. Diseñar un circuito que realice esta finalidad

Suponemos



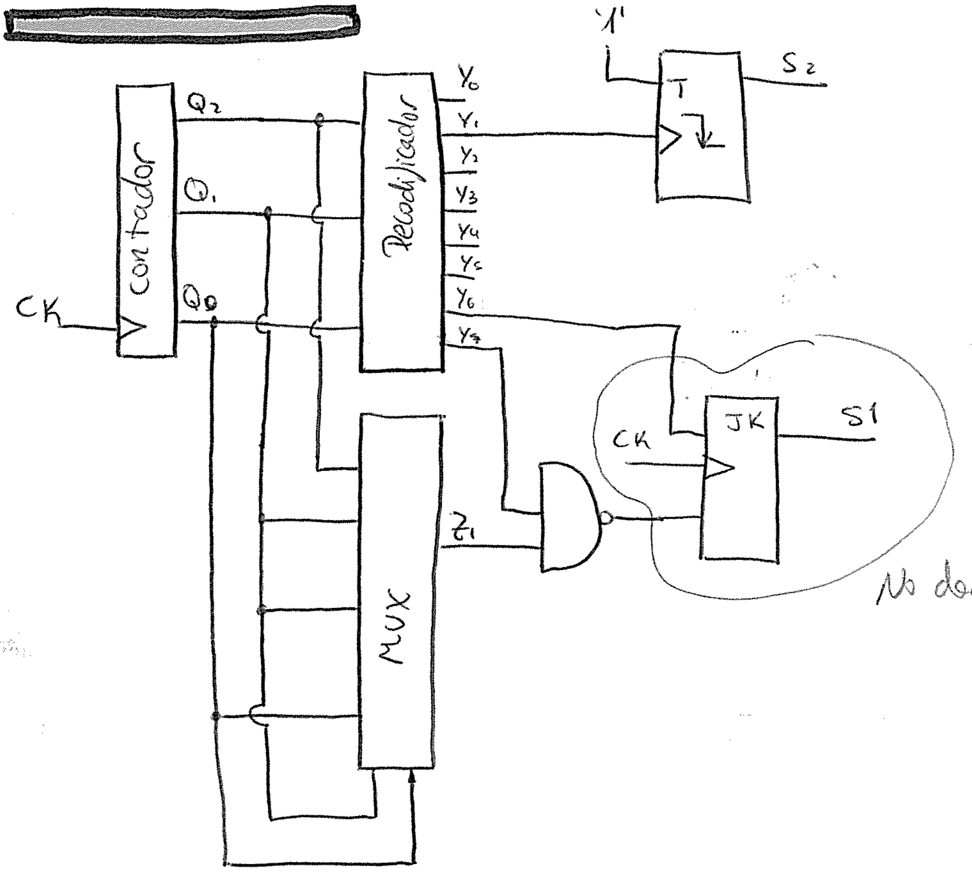




El circuito pedido será



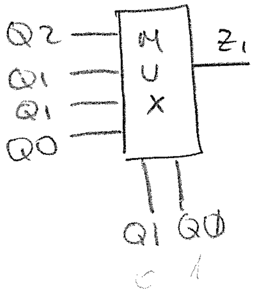

Contador asíncrono 4 bits

## Examen ENERO 2011 Problema 3




No dado NO ENTRA

**a)** Calcular la tabla de verdad de $Z_1$ en función de $Q_2 Q_1 Q_0$

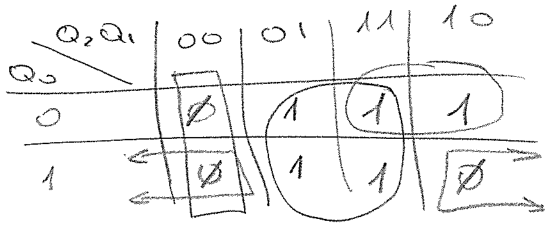



| $Q_2$ | $Q_1$ | $Q_0$ | $Z_1$ |
|---|---|---|---|
| 0 | 0 | 0 | Q2 = 0 |
| 0 | 0 | 1 | Q1 = 0 |
| 0 | 1 | 0 | $Q_1$ = 1 |
| 0 | 1 | 1 | Q0 = 1 |
| 1 | 0 | 0 | Q2 = 1 |
| 1 | 0 | 1 | Q1 = 0 |
| 1 | 1 | 0 | Q1 = 1 |
| 1 | 1 | 1 | Q0 = 1 |

**b)** Expresar Z1 según la 2ª forma canónica

$$Z_1 = (Q_2 + Q_1 + Q_0)(Q_2 + Q_1 + \overline{Q_0})(\overline{Q_2} + Q_1 + \overline{Q_0})$$

**c)** Simplificar $Z_1$ mediante Karnaugh




Como prod. sumas

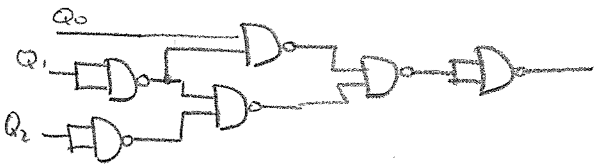
$$Z_1 = (Q_1 + Q_2)(Q_1 + \overline{Q_0})$$

$$Z_1 = Q_1 + Q_2\overline{Q_0}$$

Como suma de prod.

**d)** Implementar Z1 mediante NAND → productos negados

$$Z_1 = (\overline{\overline{Q_1} \cdot \overline{Q_2}})(\overline{\overline{Q_1} \cdot Q_0})$$

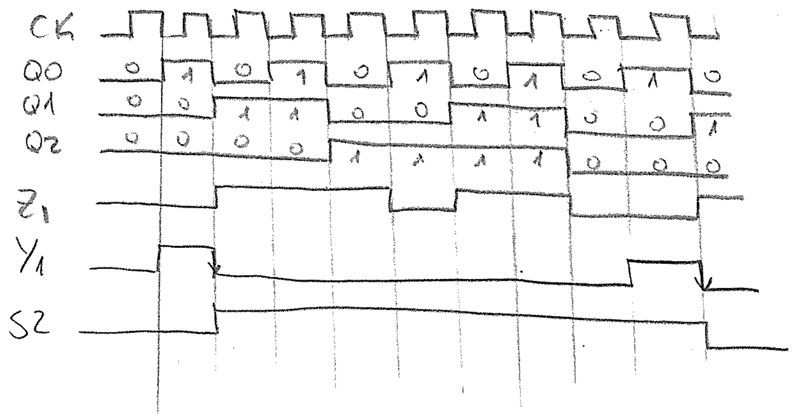



$$Z_1 = \overline{\overline{(\overline{\overline{Q_1} \cdot \overline{Q_2}}) \cdot (\overline{\overline{Q_1} \cdot Q_0})}}$$

**e)** Representar la evolución de $Z_1$, ~~Y1~~, $S_2$ para los 10 primeros ciclos de CK

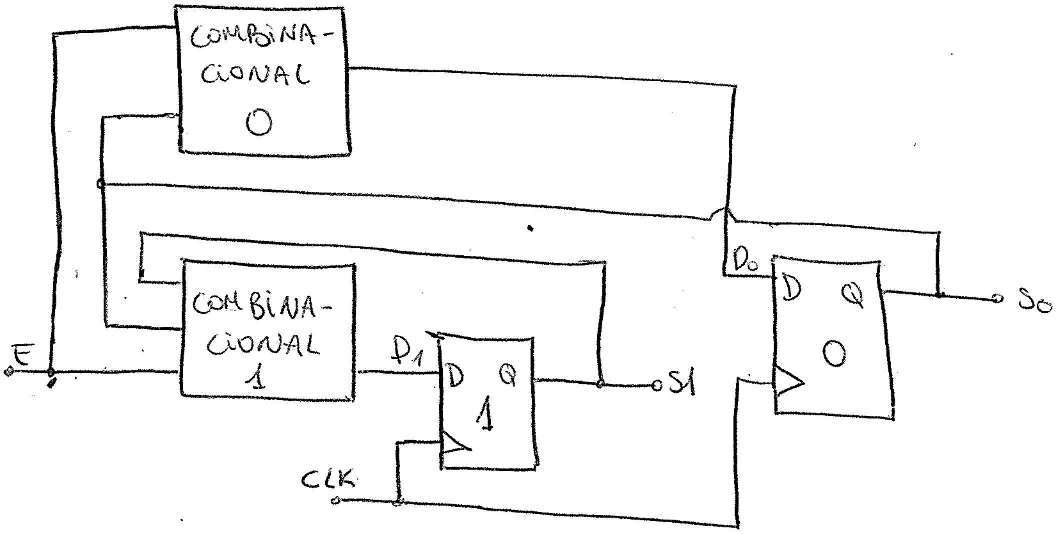
No entra

NOTA: Inicialmente biestables a cero



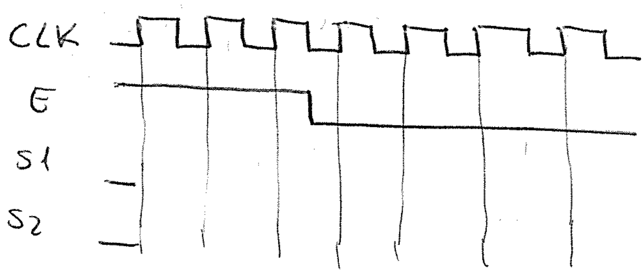
# Examen JUNIO 2011: Problema 3.

El circuito de la figura está compuesto por dos flip-flops y dos circuitos combinacionales




El comportamiento de los bloques combinacionales viene dado por la siguiente tabla de verdad

| $S_1$ | $S_0$ | $E$ | $D_1$ | $D_0$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |




Se pide: a) Diseñar ambos circuitos combinacionales con el menor nº de puertas lógicas

Mapa de Karnaugh para D1:

| E \ S1S0 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | Ø | Ø | 1 | 1 |
| 1 | Ø | 1 | Ø | 1 |

XOR??

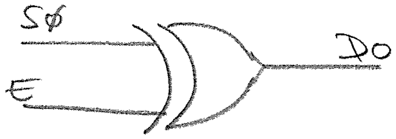
Mapa de Karnaugh para D0:

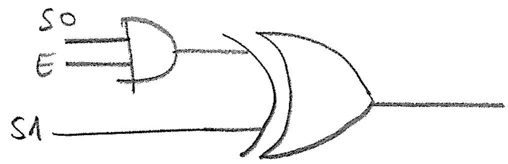
| E \ S1S0 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | Ø | 1 | 1 | Ø |
| 1 | 1 | Ø | Ø | 1 |

$$D1 = S1\overline{E} + \overline{S1}S0E + S1\overline{S0} =$$
$$= S1(\overline{E}+\overline{S0}) + \overline{S1}S0E = S1\,\overline{S0\cdot E} + \overline{S1}\cdot S0E$$

$$\boxed{D1 = S1 \oplus (S0\cdot E)}$$

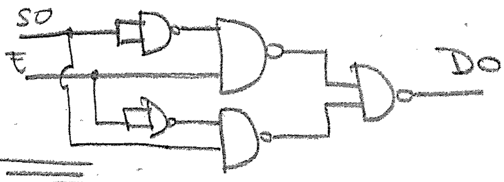
$$\boxed{D_0 = S0\overline{E} + \overline{S0}E = S0 \oplus E}$$



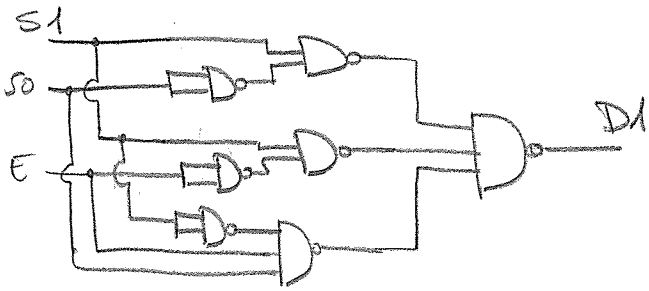




b) Obtener circuitos equivalentes usando sólo NAND

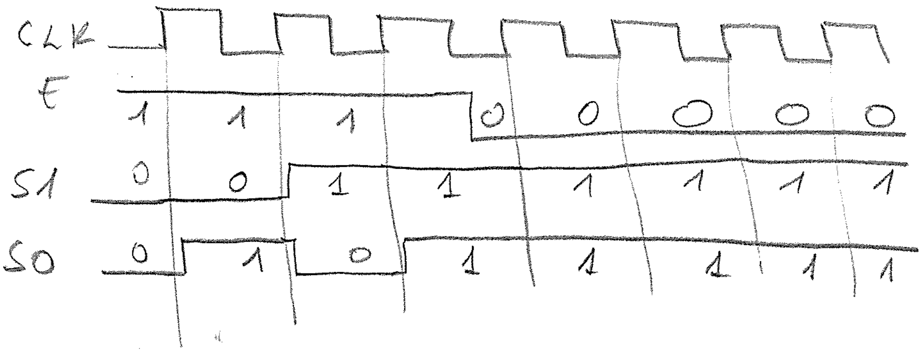
$$D_0 = \underbrace{S0\,\bar{E}}_{A} + \underbrace{\overline{S0}\,E}_{B} = \overline{\bar{A}\,\bar{B}} = \overline{\overline{S0\,\bar{E}}\;\overline{\overline{S0}\,E}}$$




$$D1 = S1\,\bar{E} + \overline{S1}\,S0\,E + S1\,\overline{S\emptyset} = \overline{\overline{S1\,\bar{E}}\cdot\overline{\overline{S1}\,S0\,E}\cdot\overline{S1\,\overline{S\emptyset}}}$$

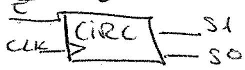


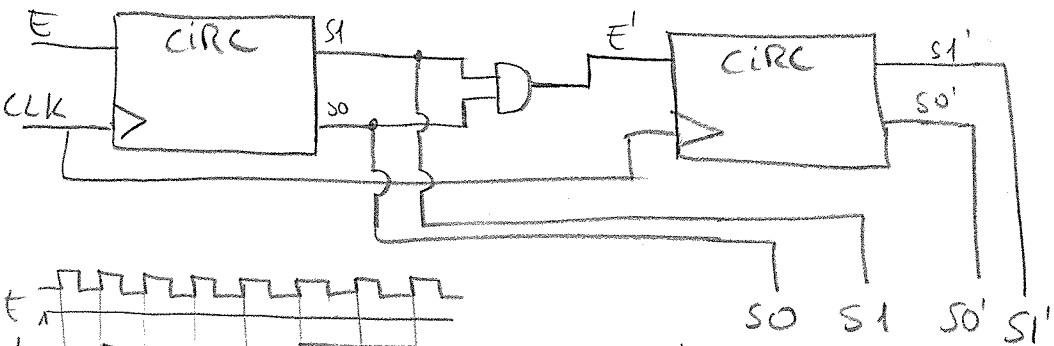

c) Completar el cronograma y decir qué función realiza el circuito

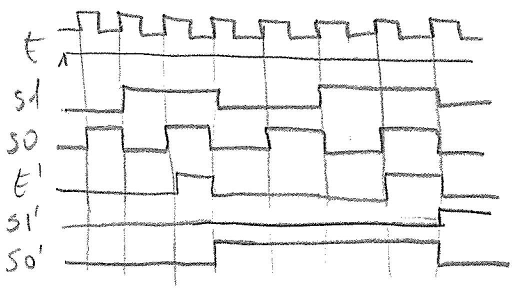



Es un contador binario ascendente de 2 bits con enable (congela el contador cuando está a Ø)

d) Usando el nº de puertas lógicas que necesite y el circuito anterior diseñar un contador binario de 4 bits

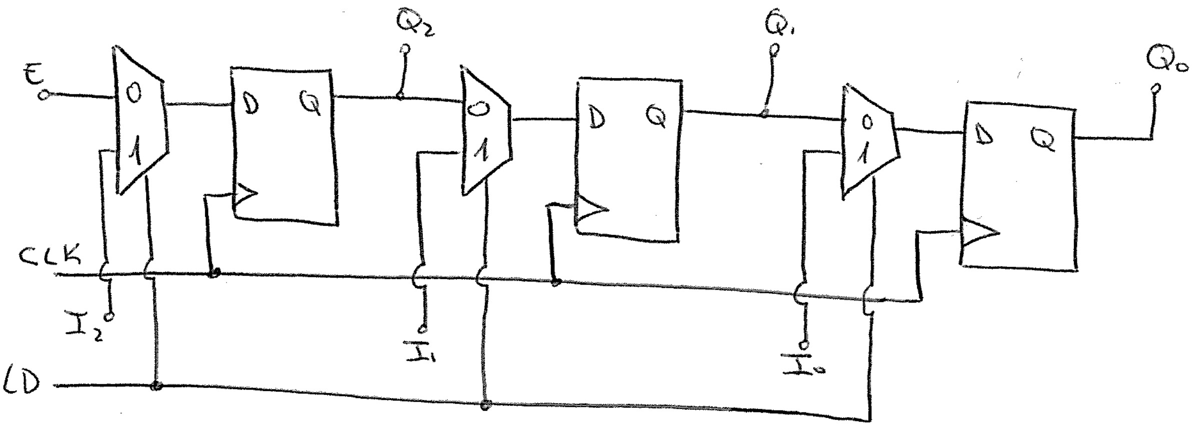



Contador binario ascendente de 4 bits con enable

**Examen Julio 2011 : Problema 3:** En el circuito síncrono de la figura se pide:

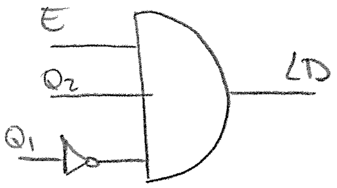



**a)** Explicar brevemente su funcionamiento cuando la señal LD está a '1' y cuando está a '0'.

Cuando LD = 1 se realiza una carga en paralelo de los 3 flip-flops, y, coincidiendo con el flanco activo del reloj, las salidas de los flip-flops se cargan de forma que $Q_2 = I_2$, $Q_1 = I_1$, $Q_0 = I_0$

Cuando LD = 0 el circuito se comporta como un registro de desplazamiento, ya que el MUX selecciona la entrada E y las salidas $Q_2$ y $Q_1$ de sus MUX correspondientes

**b)** Se desea diseñar un circuito combinacional que tenga como entradas las señales E, $Q_2$, $Q_1$, $Q_0$ y que impida que se almacene en los flip-flops el dato 110, anticipándose a esta eventualidad, poniendo su única salida LD a 1.




Se trata de detectar e impedir que las salidas de los flip-flops $Q_2 Q_1 Q_0 = 110$

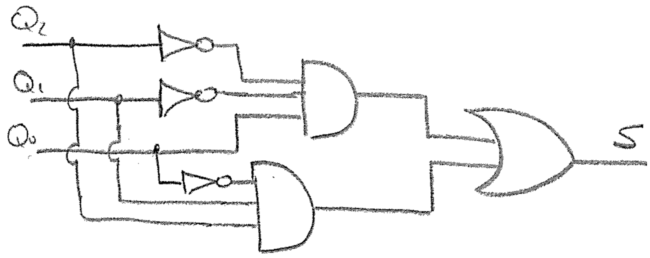
Para ello, cuando LD = 0, se ha de detectar el valor en $E, Q_2, Q_1 = 110$, con ello se consigue la anticipación pedida en el enunciado

c) Tambien se ha de diseñar un circuito con el menor nº de puertas que proporcione un '1' cuando los flip-flops en haya almacenado un 001, o bien, un 110; y un 0 en cualquier otro caso

| $Q_2$ | $Q_1$ | $Q_0$ | S |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | Ø |

| $Q_0$ \ $Q_2Q_1$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | Ø | Ø | 1 | Ø |
| 1 | 1 | Ø | Ø | Ø |

$S = Q_2 Q_1 \bar{Q}_0 + \bar{Q}_2 \bar{Q}_1 Q_0$

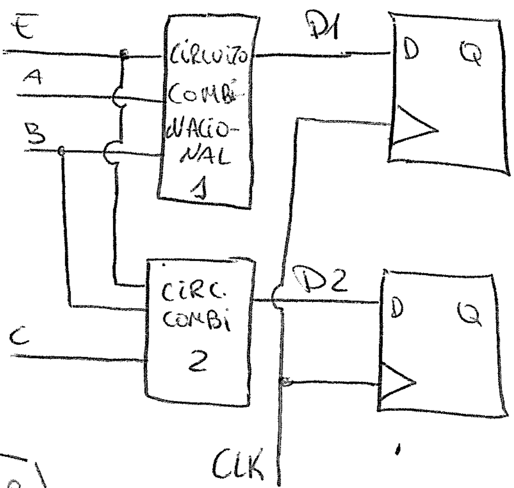



Examen ENERO 2012 Ejercicio 3. En el circuito de la figura se pide:

a) Obtener la ecuación lógica simplificada por Karnaugh y el circuito correspondiente para D1, D2 teniendo en cuenta las tablas de verdad siguientes

| E | A | B | D1 |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

| E | B | C | $D_2$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |




Mapa Karnaugh D1:

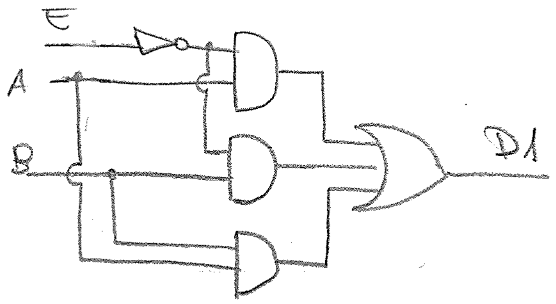
| B \ EA | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |

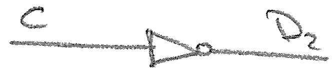
$\boxed{D1 = \bar{E}A + \bar{E}B + AB}$

Mapa Karnaugh D2

| C \ EB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |

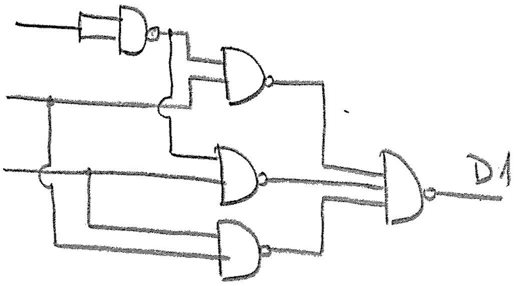
$\boxed{D2 = \bar{C}}$

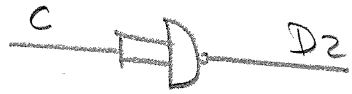


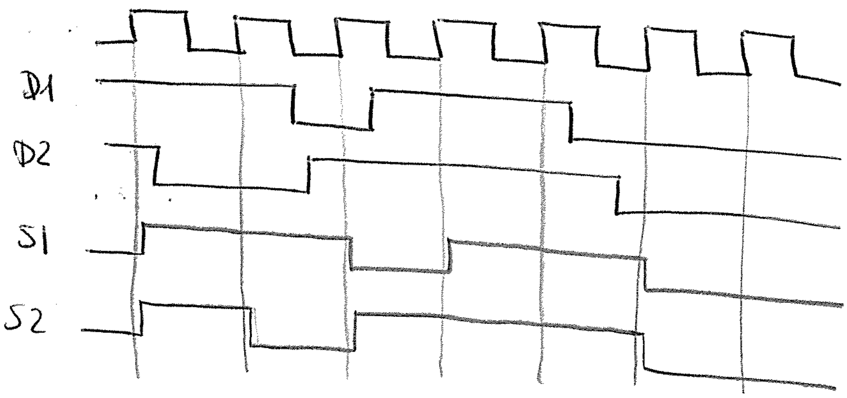

**b)** Dibujar los circuitos combinacionales 1 y 2 utilizando únicamente puertas NAND.

$$D1 = \bar{E}A + \bar{E}B + AB = \overline{\overline{\bar{E}A} \cdot \overline{\bar{E}B} \cdot \overline{AB}}$$



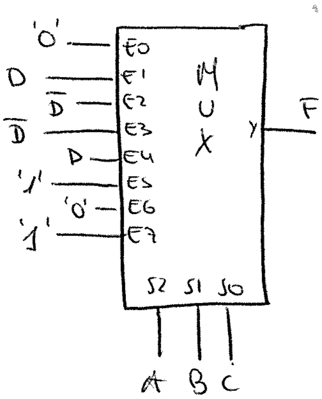




**c)** Complete el cronograma siguiente




**Examen Julio 2012**

Ejercicio 3. El circuito de la figura implementa una función lógica $F = f(A, B, C, D)$ mediante un MUX de tres entradas de selección (8 de datos)




Para este circuito se pide:

a) Tabla de verdad de $F = f(A, B, C, D)$

b) Función minimizada de F

c) Dibujar un circuito mínimo para implementar F utilizando solamente puertas NAND

a) Tabla de verdad

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

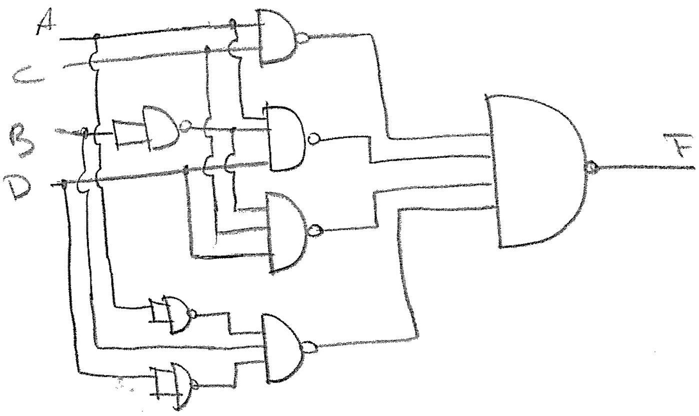
b) Función minimizada

Mapa de Karnaugh

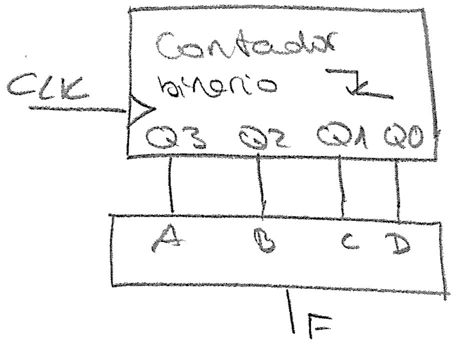
| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 1 | 0 | 0 |
| 01 | 0 | 0 | 0 | 1 |
| 11 | 1 | 0 | 1 | 1 |
| 10 | 0 | 1 | 1 | 1 |

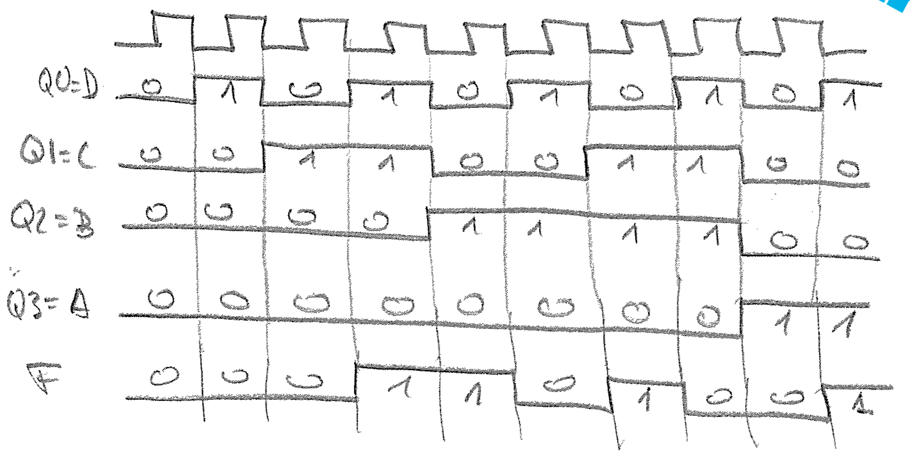
$$\boxed{F = AC + A\bar{B}D + \bar{A}B\bar{D} + \bar{B}CD}$$

c) F con NAND $\longrightarrow$ $F = \overline{\overline{AC}\cdot\overline{A\bar{B}D}\cdot\overline{\bar{A}B\bar{D}}\cdot\overline{\bar{B}CD}}$




d) El circuito anterior se conecta a un contador binario de 4 bits, activo por flanco de bajada de la entrada del reloj CLK de manera que las salidas $Q_3, Q_2, Q_1, Q_0$ del contador se conectan respectivamente con ABCD. Dibuje el cronograma de la señal F después de 9 ciclos de la señal CLK. Considere que el contador está a "0000" en el instante inicial


Contador binario
CLK
Q3 Q2 Q1 Q0
A B C D
F


Q0=D
Q1=C
Q2=B
Q3=A
F

# INTRO Microprocesadores

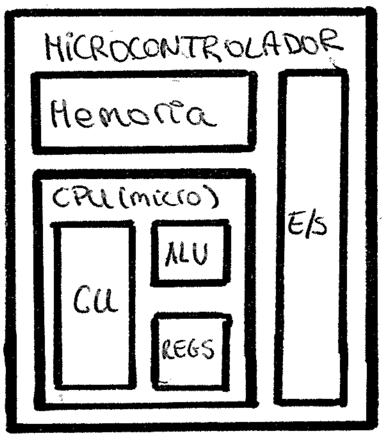
## 1 MICROPROCESADOR ($\mu P$) VS MICROCONTROLADOR ($\mu C$)

MICROCONTROLADOR: TODO UN SISTEMA EN UN CHIP.

- Microprocesador (se verá más detenidamente): Posee
  - Unidad de Control (UC)
  - Unidad Aritmético Lógica (ALU)
  - Banco de registros

  } CPU Central Processing Unit
- Memoria: Para programa y datos (chips adicionales)
- Lógica de interfaz con los dispositivos entrada/salida

Diferencia principal: El micro puede estar él solo en un chip, estando la memoria y la interfaz E/S en otros chips, por ejemplo en un ordenador. Sin embargo, si hablamos de un microcontrolador nos referimos a un circuito digital integrado (todo el sistema en un chip) que puede programarse vía software y que se utiliza generalmente para aplicaciones de bajas prestaciones, mucho más pequeñas que un ordenador.




### PRESTACIONES

Gama baja: 4, 8, 16 bits. Dedicados a tareas de control (electrodomésticos, algunos periféricos de ordenador, ...) Gen. $\mu C$

Gama media: 16, 32 bits. Tareas de control con cierto grado de procesamiento (control en automóviles, móviles, PDA, etc). Suelen ser $\mu C$ o $\mu P$ + periféricos integrados, y mem. externa

Gama alta: 32, 64, 128 bits. Fundamentalmente procesamiento (ordenadores). Casi en su totalidad $\mu P$ + circuitería periférica + memoria

Indusbol.com

TECNOLOGÍA

Alimentación: 5V, 3.3V, 2.5V, 1.8V
Consumo: $\mu$W - decenas W
Frecuencia: KHz - GHz

PRECIO

$\mu$C sencillos: 0.5 - 20 €
$\mu$P gama alta: 100 - 1000 €

# 2 SISTEMAS EMBEBIDOS

Se trata de una combinación de hardware y software con una funcionalidad fija, esto es, para una tarea específica aunque puede ser reprogramable
Su ventaja principal es que cubre los requisitos minimizando costes.

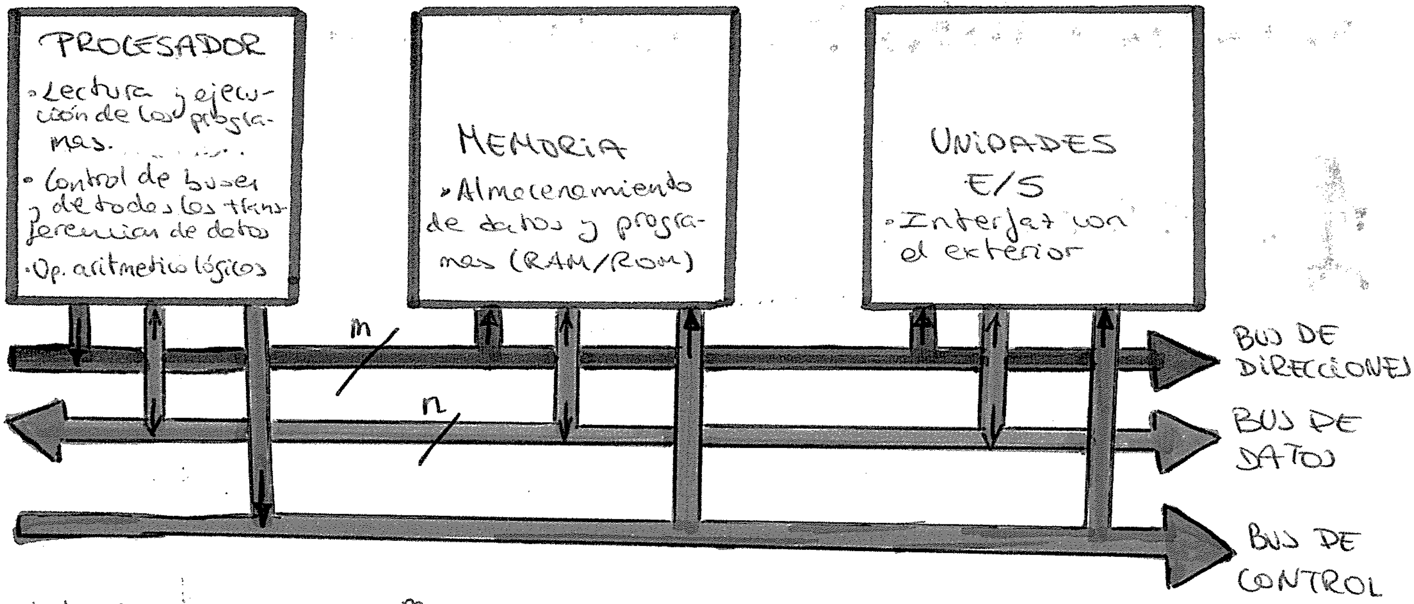
# SIST. MÍNIMO

## MICROPROCESADOR µP

## 1 DEFINICIÓN

**Sistema mínimo:** Lo mínimo que necesita un µP para realizar una tarea partiendo de que el µP es un circuito digital de propósito general que realiza una tarea en función de lo que se programe en él.

**Necesita:**
- Memoria, elemento donde almacenar programa y datos
  - Volátil: Para hacer operaciones, su información se pierde al apagar el sistema.
  - No volátil: ya que, al menos, una pequeña parte del programa debe estar accesible al encender el sistema.
- Elemento que realice cálculos, tome decisiones, ..., es la CPU, en ella se hacen las operaciones con los datos, interactúa con la memoria y la E/S y es el núcleo del sistema mínimo.
- Entradas y salidas
  - Entradas: señales que el sistema es capaz de leer y reaccionar en función de su valor
  - Salidas: señales que controlar de acuerdo al programa y a las entradas
  - Unidades E/S: Circuitería digital que realiza el interfaz entre el µP y las señales E/S.




- m bits de direcciones: $2^m$ posiciones o direcciones de memoria distintas
- n bits de datos: n bits en paralelo (m cajitas de memoria que guarden cada una n datos)

→ Capacidad: $2^m \cdot n$

- Varias señales de control

**BUSES**
- Direcciones: Selecciona dirección origen o destino — BUS UNIDIRECCIONAL
- Datos: Realiza la transferencia del dato — BUS BIDIRECCIONAL
- Control: Reloj, reset, read/write, interrupción, ... — BUS HETEROGENEO.

**Transferencias de datos.**

Escritura UP → MEMORIA
- Bus de control: señal write a memoria
- Bus de direcciones: Escoge una dirección de la memoria
- Bus de datos: Escribe el dato en esa dirección de memoria (UP → memoria)

Lectura MEMORIA → UP
- Bus de control: señal read a memoria
- Bus de direcciones: Escoge una dirección de la memoria
- Bus de datos: Lee el dato de esa dirección de memoria y se lo pasa al micro (memoria → UP)

Entrada E/S → UP

Salida UP → E/S

# ARQ. INTERNA

## 1 LENGUAJE ENSAMBLADOR Y CÓDIGO MÁQUINA

Para transmitir instrucciones dentro de un micro se usa el lenguaje ensamblador que se codifica en un código binario. Según el micro, esta codificación podría hacerse en una o varias palabras por instrucción.

Para realizar la codificación, se usa un programa ensamblador, que se encarga de traducir el lenguaje ensamblador a código máquina. Normalmente se utiliza una codificación por "campos"

Ejemplos:

| | | |
|---|---|---|
| R0 ← M(1000) | LD R0,1000 | En el registro R0 cargo la dirección de memoria 1000 |
| R0 ← 5 | LDI R0,5 | En el registro R0 cargo la constante 5 |
| R0 ← R0+R3 | ADD R0, R3 | Sumamos el registro R0 y R3 y lo guardamos en R0 |
| R4 ← $\overline{R4}$ | NOT, R4 | Negamos R4 y lo guardamos en R4 |
| M(FC00) ← R3 | ST R3,FC00 | Guardamos el contenido del registro R3 en la dirección de memoria FC00 |
| R4 ← R5 | MOV R5, R4 | Guardamos el contenido de un registro R5 en la memoria de otro registro R4 |

Para instrucciones puede:

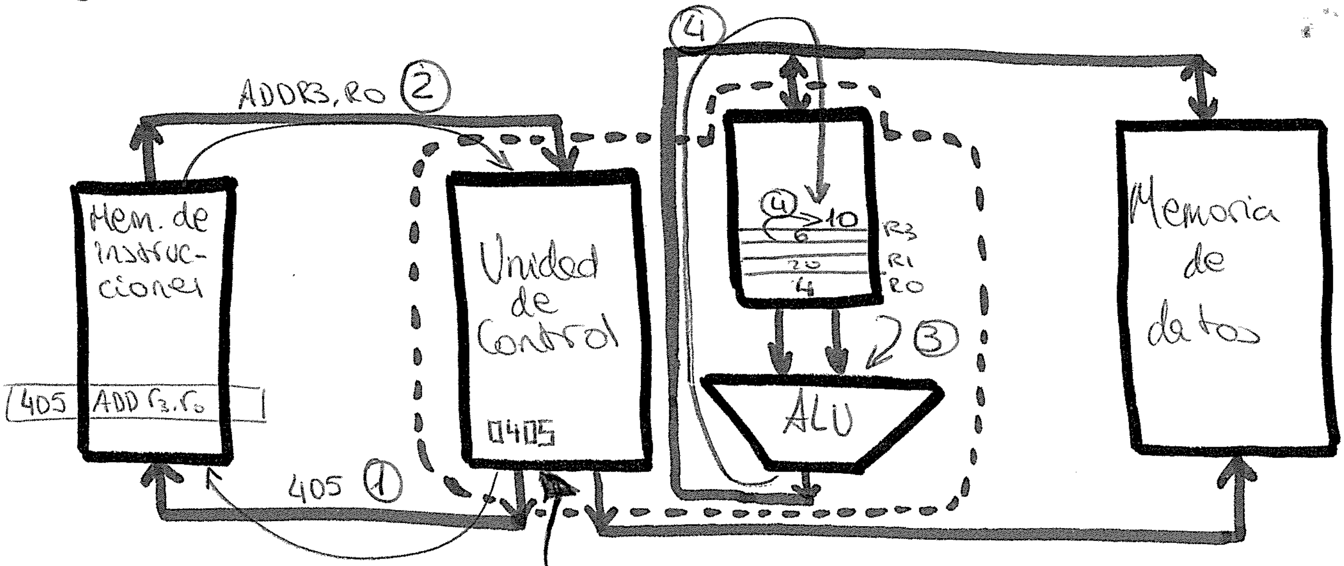
Ejecutar una instrucción detrás de otra

Saltar a una instrucción determinada y continuar la ejecución desde ahí JMP2000 (incondicional)

Salta un nº de instrucciones más adelante si el resultado de la última operación con la ALU fue ∅ (por ejemplo) BREQ 100

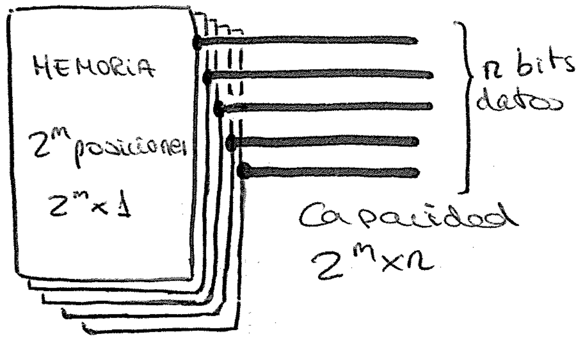
# 2 CICLO BÁSICO DE EJECUCIÓN DE UNA INSTRUCCIÓN

La CPU va leyendo cíclicamente instrucciones de la memoria de programa, las interpreta y la unidad de control activa las señales de control correspondientes para ejecutarlas



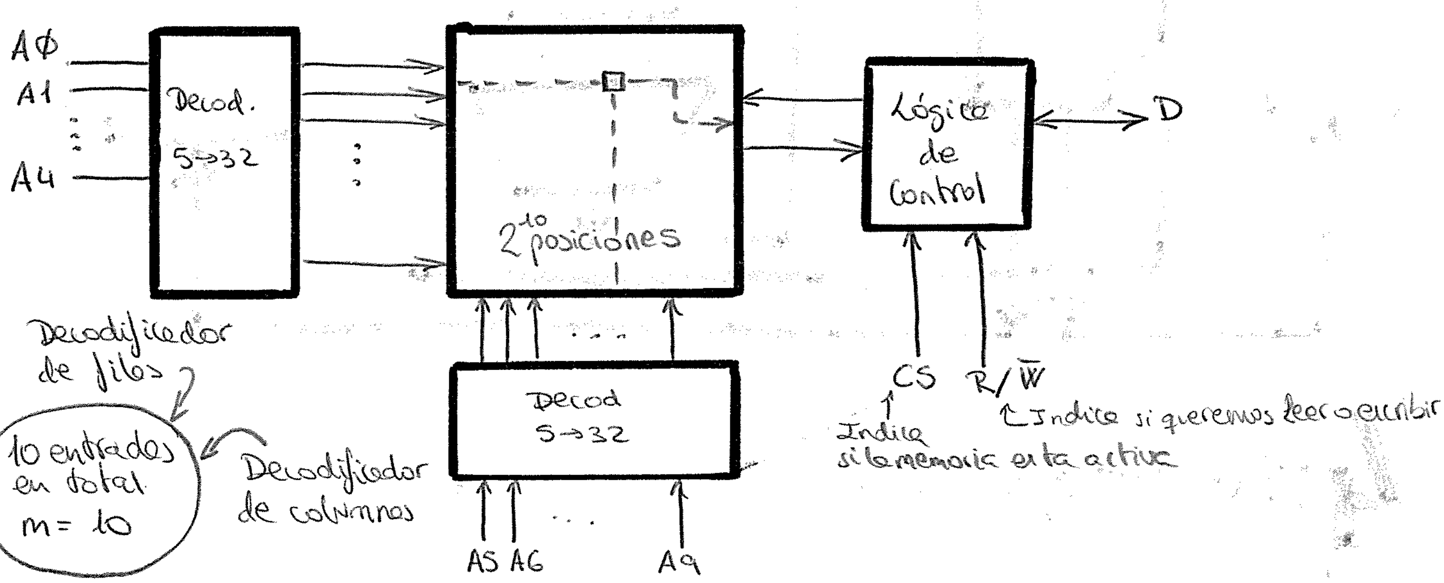

1→ Búsqueda de la instrucción
2→ Decodificación de la instrucción
3→ Operación / ejecución
4→ Carga / Almacenamiento en registros / memoria
5→ Incremento del contador de programa

# 2 MEMORIAS

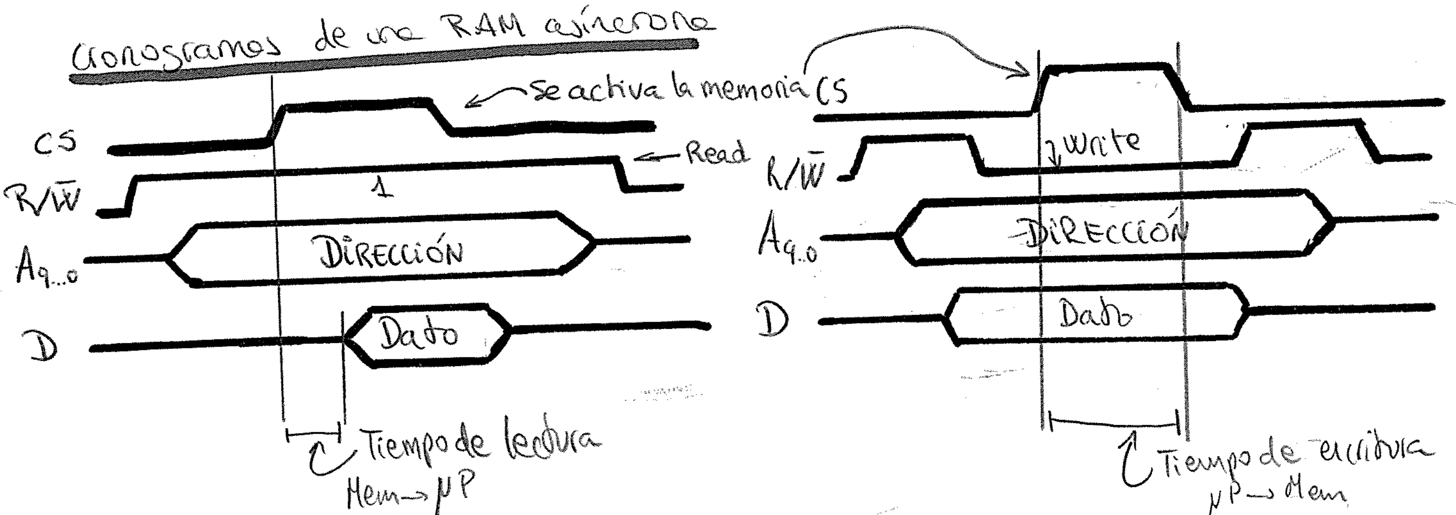



Se estructuran normalmente en $2^m$ ($m$ = ancho del bus de direcciones) posiciones de memoria de $n$ ($n$ = ancho del bus de datos) bits cada una

El diseño interno se optimiza para almacenar muchos bits a bajo coste, por eso son bastante más lentas comparadas con otros circuitos digitales. La estructura interna de cada celda de memoria depende del tipo de memoria y de su tecnología de fabricación.



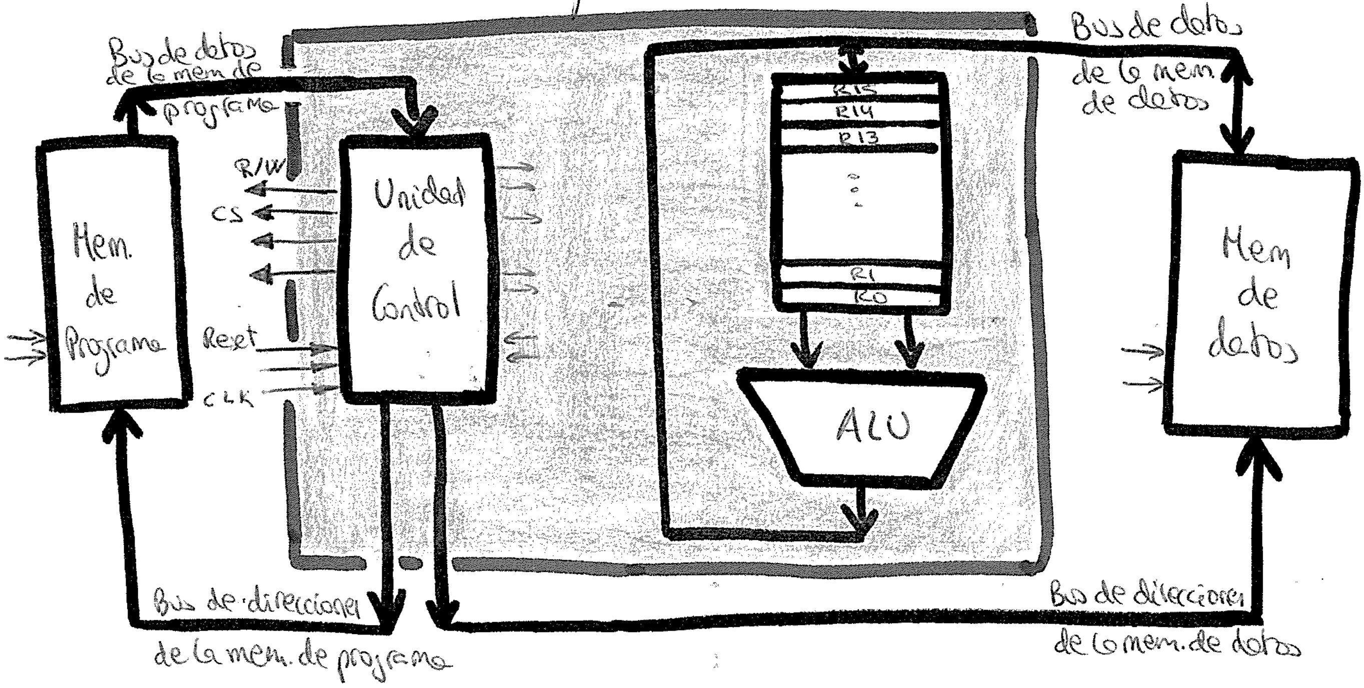

## Cronogramas de una RAM asíncrona



# 3 UNIDAD CENTRAL DE PROCESOS CPU

Encargada de la secuenciación de todas las operaciones y movimiento de datos. Típicamente se divide en:

- Banco de registros: almacena datos y resultados de operaciones con la ALU
- Unidad Aritmético Lógica (ALU): Realiza operaciones entre registros
- Unidad de Control: Secuencia las acciones en función de la instrucción a ejecutar




# 4 DISPOSITIVOS E/S

Algunos ejemplos típicos:

E/S paralelo para control y comunicaciones
E/S serie para comunicaciones
Temporizadores y contadores
Perro guardián (watchdog) que controla la ejecución correcta de un programa
Convertidores A/D a D/A
Controladores específicos como discos duros, disquetes, ...

# E/S Dispositivos

## 1 PUERTOS Y REGISTROS E/S

Puerto E/S : Es un conjunto de señales externas con una funcionalidad común y circuitería que permite leerlas y/o controlarlas
Existen diferentes tipos: E/S paralelo, E/S serie, E/S analógica y, según se configuren pueden tener asociados diferentes funcionalidades

Registros E/S: Cada puerto E/S puede tener asociados uno o varios registros E/S, o un registro puede afectar a varios puertos.
Existen registros asociados } a los datos (las señales a controlar)
} y al control (el modo de operación)
Los registros de E/S se acceden mediante instrucciones de lectura/escritura de forma análoga a cómo se accede a las posiciones de memoria

## 2 PUERTOS E/S PARALELO EN AtMega168

Los puertos E/S paralelo controlan varias señales externas al mismo tiempo y son los más comunes

Registros asociados

- De datos: Conectado a las señales de E/S
- De direcciones de los datos: fija la dirección de las señales E/S
  Habitualmente '0' indica entrada
  '1' indica salida

En AtMega
Puerto B → Genera señales PWM
Puerto C → Convertidor A/D
Puerto D → E/S paralelo + puerto serie
señales PWM

**REGISTRO PINx** — Lectura de un puerto

- Un bit se lee como un '1' si hay un '1' (Vcc) en su pin correspondiente
- Un bit " " " " '0' " " " '0' (Gnd) " " " "

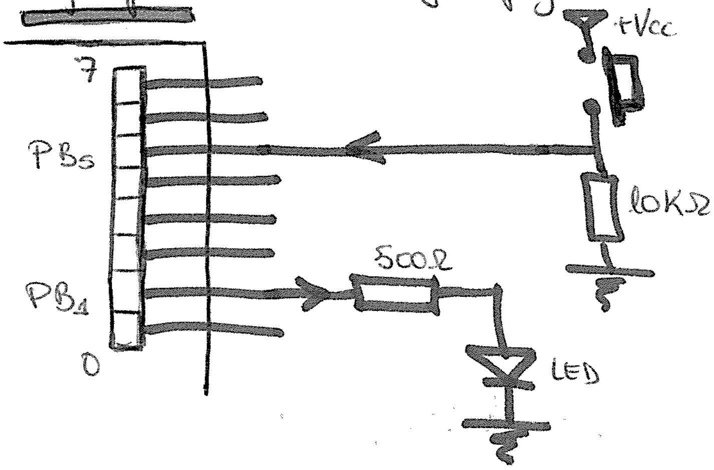
**REGISTRO DDRx** — Data Direction Register

- Configura el sentido de cada bit del puerto
- 'Ø' entradas y '1' salidas

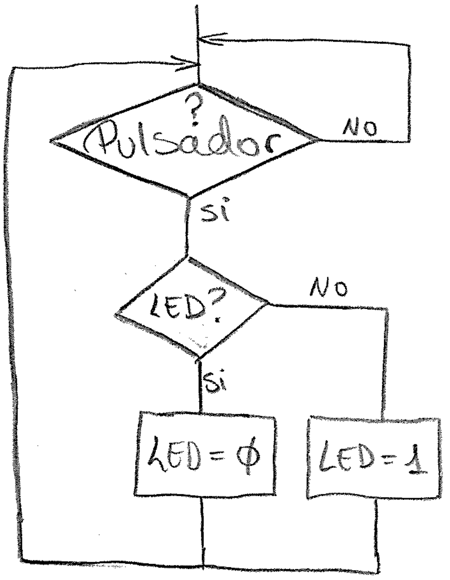
**REGISTRO PORTx** — Si un pin está configurado como salida

- Al escribir un '1' en un bit se pone un '1'(Vcc) en su pin correspondiente
- " " " '0' " " " " " " '0' (Gnd) " " " "

**Ejemplo:** Encender y apagar un LED con un pulsador




```
int boton;
int led=0;

int main(){

 DDRB= 00000010;
       76543210

while(true){

 boton= PINB & 00100000
               76543210
//* cuando pulsemos el interruptor
 PB5 se pone a 1 y boton=1//

if(boton!=0){
   if (led==0){
   PORTB= PORTB | 00000010;
                  76543210
   //Mandamos Vcc por PB1, y encendemos
   el led// led = 1
   }else{
   PORTB= PORTB & 11111101;
   //Mandamos GND por PB1 y apagamos
   el led// led=0;
   }
  }
 }
}
```



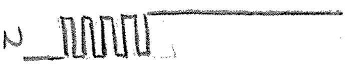
# ENCENDIDO DE LED CON ARDUINO

```
void setup(){
    pinMode(15, OUTPUT);   ← Led
    pinMode(19, INPUT);    ← Boton
}
void loop(){
    int led = 0;    ← Definimos variable que guarda estado del led
    digitalWrite(15, LOW);   Apagamos el led
                             Tambien podríamos haber hecho
                             int led = digitalRead(15);
    int boton = digitalRead(19);
    if (boton != 0){   ← delay(100);
        if (led == 0){
            led = 1;
            digitalWrite(15, HIGH);
        }
        else {
            led = 0;
            digitalWrite(15, LOW);
        }
    }
}
```

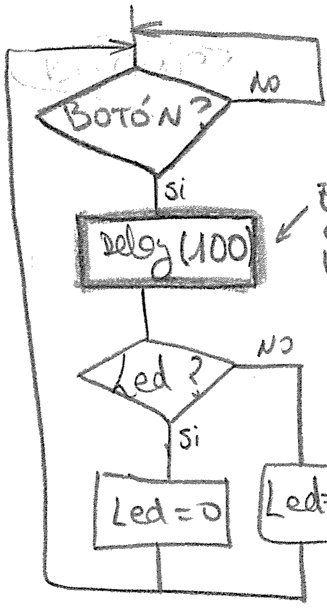
En Arduino
PB5 (entrada por bot) ⟺ 19
PB1 (salida al led) ⟺ 15

## PROBLEMA REBOTES

Mientras pulsamos el boton el led luce levemente

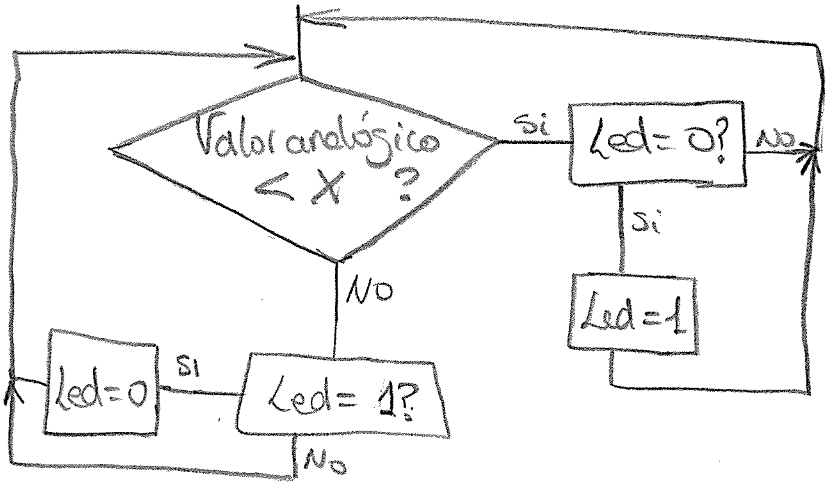
BOTON

Rebotes del interruptor

## SOLUCIÓN




Esperamos a que pasen los rebotes -100 ms-

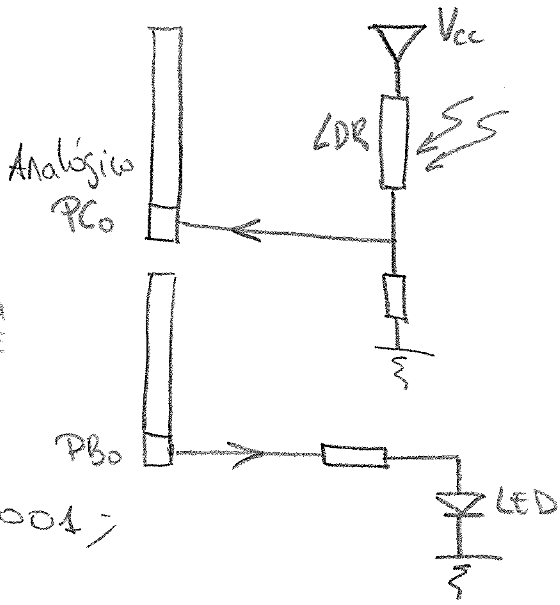
# CONTROLAR LED CON LUZ AMBIENTE




```
int main(){
    int AnalogVal;
    int led=0;

        76543210
DDRB = 00000001

//Configuraríamos ADC para
leer el valor analógico de
la LDR por PC0//

while (true){
    analogVal = valor devuelto por
                el ADC del PC0
                (0 – 1023)

    if (analogVal < 100){      Valor cuando no hay casi luz → NOCHE

        if(led==0){
            led=1;
            PORTB = PORTB|00000001;
        }
    } else {
        if (led==1){
            led=0;
            PORTB= PORTB|11111110;
        }
    }

} //cierre while (true)

} //cierre int main()
```

Si es de noche

Si es de día

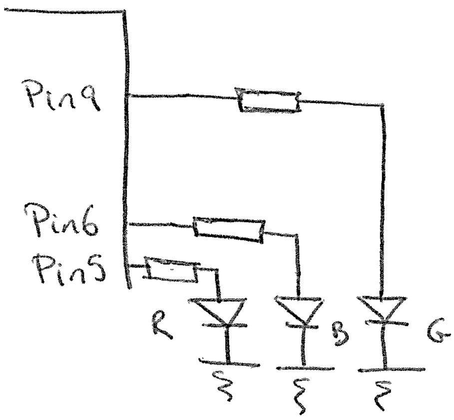


y con Arduino?

Pines:
Analógico A0
Digital Pin 8

```
void setup(){
    pinMode( 8 , OUTPUT);
    pinMode (A0, INPUT);
}

void loop(){
  int analogVal=analogRead(A0);  Guardamos el valor de la LDR
  int led=0;
  if(analogVal < 100){
        if(led==0){
        digitalWrite (8, HIGH);
        led=1;
        }
  }else{
        if(led==1){
        digitalWrite(8, LOW);
        led=0;
        }
  }
}
```

# Controlar intensidad de un led rojo para variar el color




```
void setup(){
    pinMode (5, OUTPUT) ← Rojo
    pinMode (6, OUTPUT) ← Azul
    pinMode (9, OUTPUT) ← Verde
}
void setColor(int r, int b, int g){
    analogWrite (5, r);
    analogWrite (6, b);
    analogWrite (9, g);
}
int r=0;
int inc=1;
void loop(){
setColor (r, 200, 100);
    r = r + inc;
    if (r==255) { inc = -1 ;}
    if (r==0) { inc = 1 ; }
    delay (10);
}
```

Ponemos el rojo a cero

Hay que definirlos fuera para que cada vez que se ejecute el bucle no cambien a los valores iniciales

Valores fijos en verde y azul

Incrementamos en 1 la intensidad del rojo

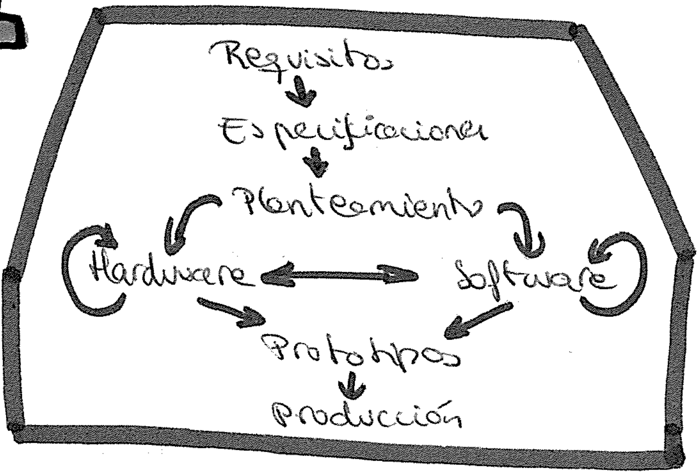
Esperamos 10 ms y volvemos

Si rojo llega a valer 255 empezamos a disminuir hasta que vuelve a llegar a cero

# DESARROLLO

## PROCESO Y HERRAMIENTAS

## 1 ETAPAS PROCESO DE DISEÑO




### Planteamiento

Elección del modo de operación/arquitectura. La dificultad del hardware y el software dependen de esta fase

### Software

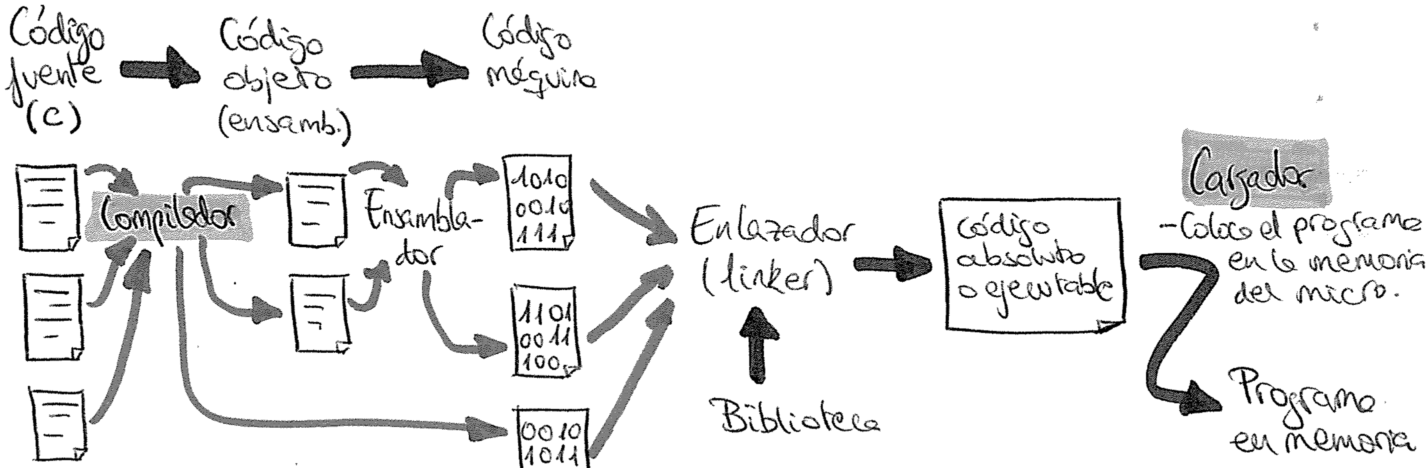
Desarrollo en lenguajes de alto nivel y ensamblador.
Las modificaciones en software son más baratas

### Hardware

Debe desarrollarse un hardware suficientemente flexible como para admitir correcciones por software.

## 2 HERRAMIENTAS DE DESARROLLO

- Proceso de generación y carga de un programa en el micro




El código fuente puede ser ensamblador y compilarse directamente a código máquina

Las herramientas de compilador y cargador (y asociadas), generan el programa. Son necesarias, además, herramientas para depurarlo (probarlo y corregirlo).

**Simulador**

Simula el funcionamiento del micro (µP o µC) respondiendo funcionalmente igual pero no es realista (más lento, no interactúa con E/S, ...)

**Depurador/Programador**

Sistema electrónico que se usa para programar el µP/µC, pero que además dispone de capacidades de depuración

**Kit de desarrollo**

Sistema electrónico de bajo coste que permite interactuar con la aplicación pero con capacidades de depuración reducidas.

# 3 SIST. Y PROCESO DE DESARROLLO CON µC

- Desarrollo del software (edición/compilación)
- Simulación en el pc/depuración (verificación funcional, ejecución, ...)
- Emulación/depuración (verificación preliminar con el hardware)
- Grabación
- Pruebas de campo

— Desarrollo del hardware
— Fabricación

# E/S

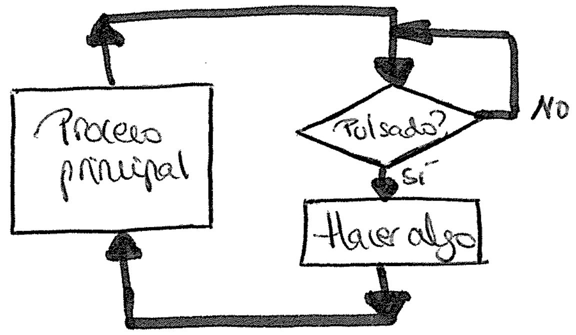
# MÉTODOS

## 1 PROCEDIMIENTOS E/S

Son la forma en que hacemos que nuestro sistema interactúe con las señales E/S

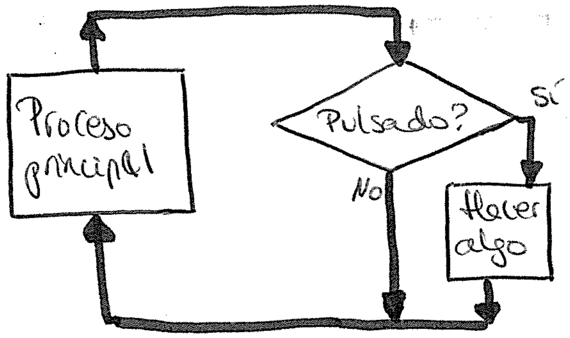
- **BLOQUEO DE PROCESO:**

El $\mu P$ espera a que el periférico le responda



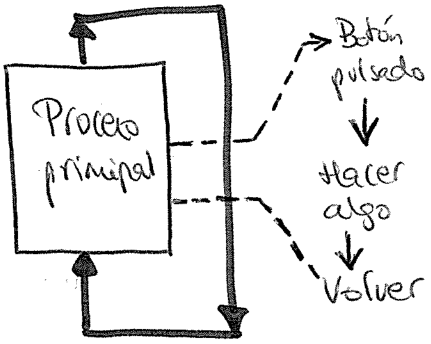

- **CONSULTA PERIÓDICA (polling)**

El $\mu P$ consulta de forma periódica el estado del periférico

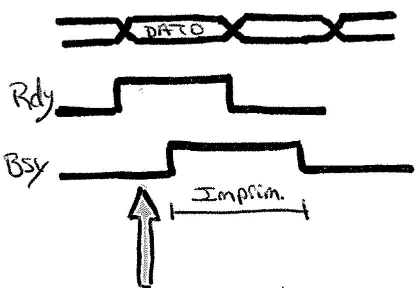



- **POR INTERRUPCIÓN**

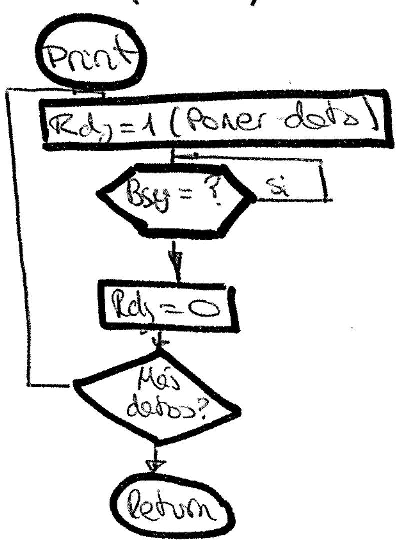
El $\mu P$ responde al periférico cuando éste le interrumpe (se verá)




IMPRESORA (por bloqueo de proceso)




No funciona, la impresora recibe el dato y tarda en ponerse a imprimir.
NO IMPRIMIRÍA !!



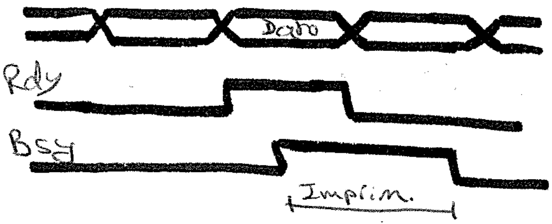

Problemas de bloqueo de proceso y ventajas

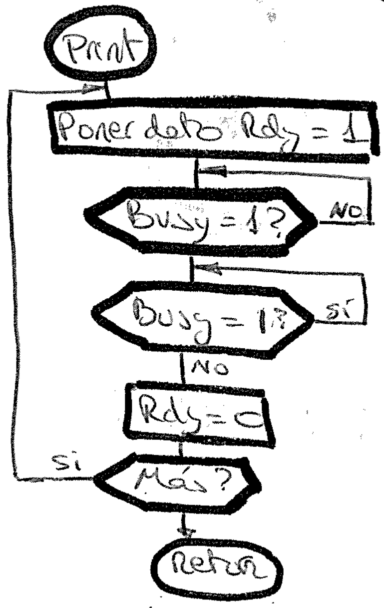
☺ Fácil de implementar

☹ Bloquea el $\mu P$ esperando respuesta. Válido para pocos datos y periféricos rápidos

☹ Problemas derivados de las diferentes velocidades

Solución a problema impresora lenta



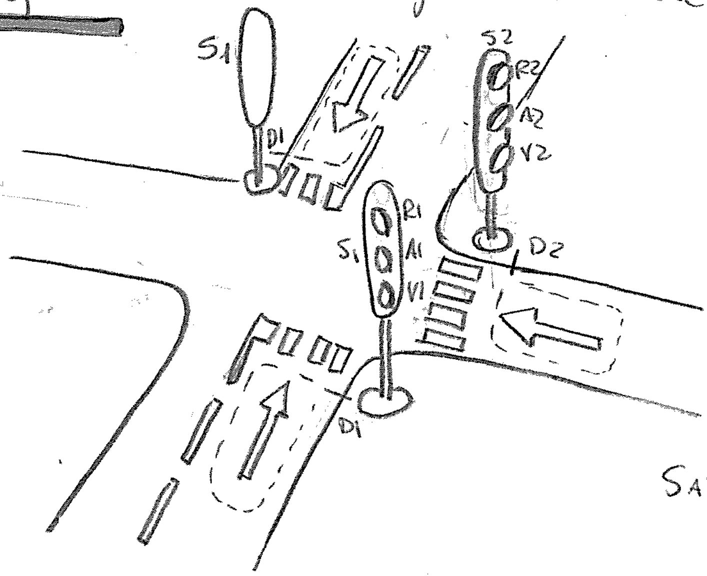



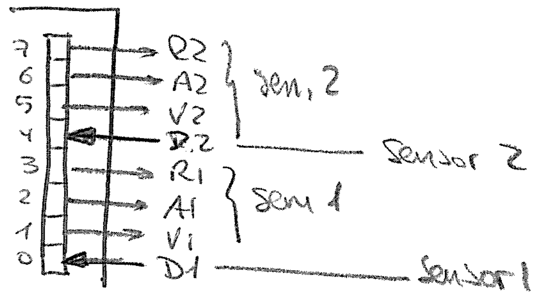

## Ventajas y desventajas consulta periódica

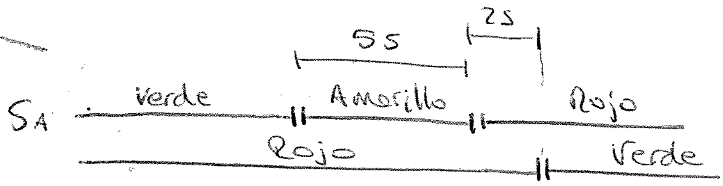
- No detiene la ejecución del resto de la aplic.
- Más complicado especialmente si hay muchos periféricos

Al usar consulta periódica, los estados que se generaban en los bucles usando bloqueo del proceso deben sustituirse por variables que mantengan memoria del estado en el que está el sistema.

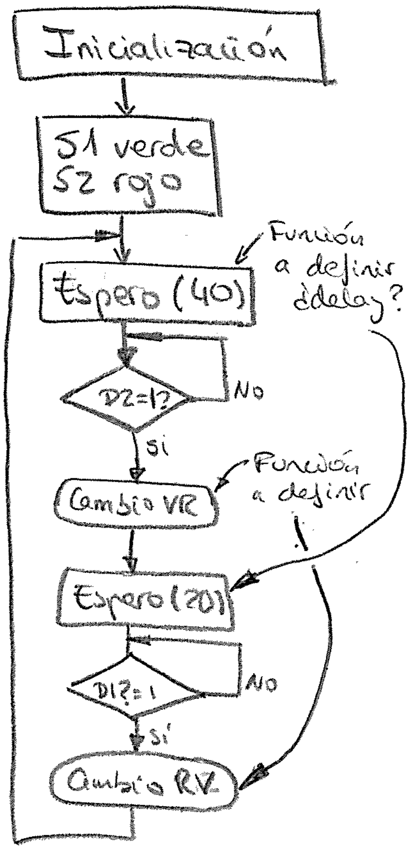
## Ejercicio: Control de semáforos en un cruce

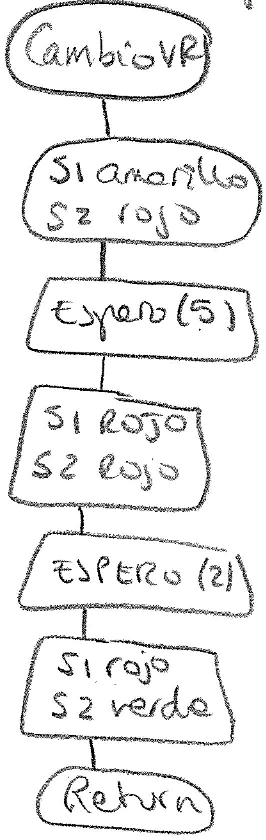


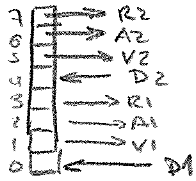

- La vía 1 es la principal que está normalmente abierta
- Cuando llega un coche a 2 (D2 → 1) se abre la vía 2 y se esperan 20 s antes de cerrarla, o, si no hubiese coches en 1, hasta que aparezca un coche en esta vía (D1 → 1)
- Cuando se vuelve a abrir la vía 1, al menos han de pasar 40 seg. antes de volver a abrir 2 (si fuese necesario)
- El paso de verde a rojo en cualquiera de los sem. debe hacerse cómo se indica




S1 verde → S1 rojo
S2 rojo → S2 verde } El cambio contrario es análogo

① Inicializo puertos




Salidas 1
Entr. ∅

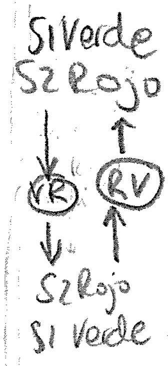
1110 1110

② Inicializo el semáforo
S1 verde escritura
S2 rojo

1000 0010

③ Funciones
④ Void loop()

```
void setup(){
    DDRB = 11101110;  //E/S
    PORTB = 10000010;  // S1 verde, S2 rojo
}

void cambioVR(){
    PORTB = 10|0|0|0|1|0|0;
    delay(5000);
    PORTB = 1000 1000;
    delay(2000);
    PORTB = 0010 1000;
}
```




```
void cambioRV(){
    PORTB = 0100 1000;
    delay(5000);
    PORTB = 1000 1000;
    delay(2000);
    PORTB = 1000 0010;
}
```

```
void loop(){
  Delay (40000);
  int D2;
                       D2
                       ↓
  D2 = PINB & 00010000;
  while (D2 == 0){
     D2 = PINB & 00010000;
  }
 Cambio VR();
 delay(20000);
 int D1;
 D1 = PINB & 00000001;
 while (D1 == 0){
   D1 = PINB & 00000001;
 }
 Cambio RV();
}
```

Cuando leas en PINB un 1 en la 4ª posición, la del sensor 2, D2 valdrá 1

// Se ejecutará este bucle hasta que D2 sea 1
BLOQ PROC.

// Se ejecutará por BLOQ DE PROCESO hasta que D1 valga 1

TEMA 7

Micro

Indusbol .com

UPM DIE
INDUSTRIALES

# 7 - Interrupciones

- Conceptos generales
- Interrupciones externas
- Interrupciones temporales
- Ejemplos



cei@upm.es

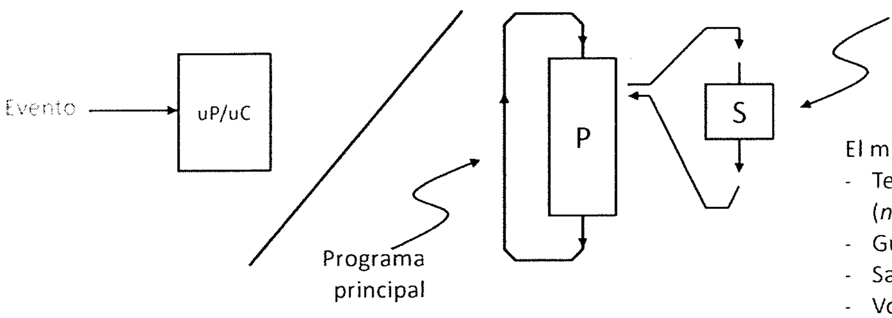
Si se activa ① de una interrup. se ejecuta x la que tiene prioridad

Tipos
RESET: Es la mas priorit. → Ejecuta la rutina de inicializ.
ENMASC. INT Priorid. interm.
NO ENMASC. NMI Priorid. Sup.

## Interrupciones: Conceptos generales (I)

**¿Qué es una interrupción?**

- ❑ Es el desvío de la ejecución normal de un programa a petición de un periférico
- ❑ El origen de la petición es un evento externo al uP, y por tanto, asíncrono respecto al programa (no podemos predecir en qué parte del programa, ni cuándo, ocurrirá)
- ❑ Si el micro la acepta, ejecutará una función asociada para atender al periférico




Subrutina de servicio, o atención a la interrupción

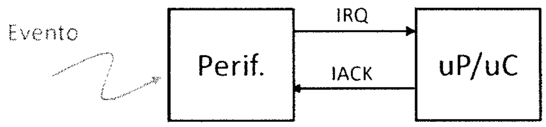
El micro debe:
- Terminar la instrucción en curso (*no en C, sino en código máquina*)
- Guardar el estado del micro
- Saltar a la subrutina y ejecutarla
- Volver (*dejándolo todo igual, en principio*)

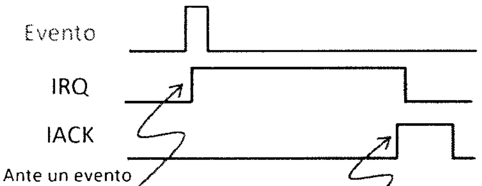
**La importancia de las interrupciones en el control**

- ❑ El control se simplifica, ya que se atiende a los periféricos cuando éstos lo requieren, y no es necesario hacer *polling* o bloqueo
- ❑ En ocasiones, un programa de control está formado por un bucle de espera, en el que no se hace nada, y sólo se atienden las interrupciones que van llegando

# Interrupciones: Conceptos generales (II)

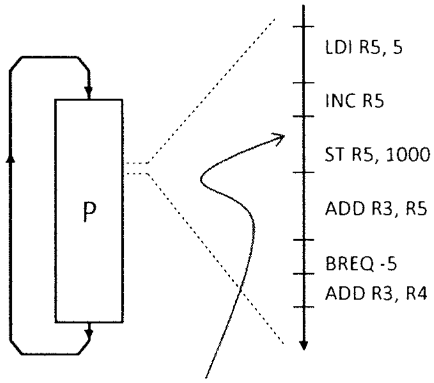
**¿Cómo se produce una interrupción?**




Ante un evento el periférico activa la señal de solicitud de interrupción

Cuando el uP acepta la interrupción, la reconoce para que el periférico retire la petición




La interrupción puede ocurrir en cualquier instante, pero la Unidad de Control del uP sólo consulta la señal de interrupción cuando va a buscar la siguiente instrucción

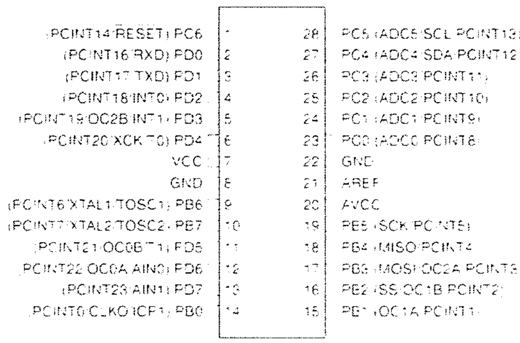
**Enmascaramiento**

- El uP dispone de mecanismos (máscaras) para habilitar y deshabilitar la llegada de interrupciones
- Existen máscaras a nivel global (deshabilitan todas las ints.) y máscaras locales a cada periférico (para deshabilitar o habilitar las interrupciones provenientes del mismo)
- Cuando se atiende una interrupción, lo normal es que las interrupciones se deshabiliten automáticamente, volviéndose a habilitar al retornar al programa principal

# Interrupciones externas o físicos

- Casi todos los uC disponen de señales externas que pueden producir interrupciones en determinadas condiciones (flanco de subida o bajada, nivel...)

- En el caso del Atmega168, todos los pines de E/S permiten producir interrupciones externas, pero sólo dos del puerto D son configurables (por flanco de subida, bajada, ambos...). *Sólo trataremos estos.*




**Funciones de la API de Arduino para trabajar con las $\boxed{\text{interrupciones externas}}$:**

Los pines 2 y 3 de la placa de Arduino están conectados a PD2 y PD3, y por lo tanto tienen la posibilidad de interrupciones externas PD2 ≡ INT∅ → pin 2 * En el attach empiezo poniendo ∅
PD3 ≡ INT1 → pin 3

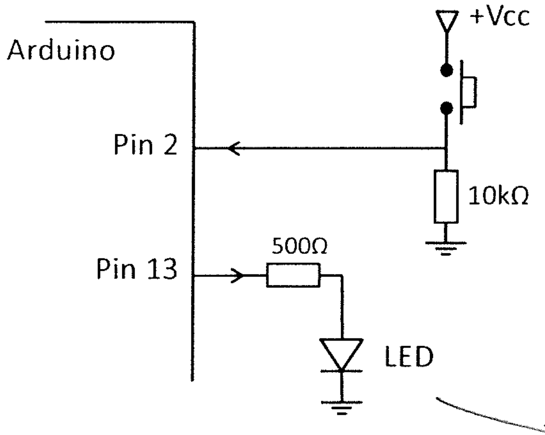
- **attachInterrupt**( int_num, isr_name, condition ); // Asocia una rutina a una interrupción bajo una determinada condición
  - *int_num* : 0 para el pin 2, 1 para el pin 3
  - *isr_name*: nombre de la rutina de servicio (función que no debe tener parámetros)
  - *condition*: condición que produce la interrupción ( FALLING, RISING, BOTH, LOW ) En nivel bajo

  Flanco bajada — Flanco subida — ambos

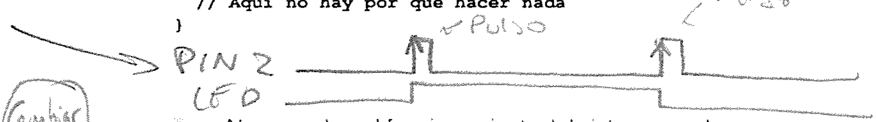
- **dettachInterrupt**( int_num ); // Desactiva la interrupción

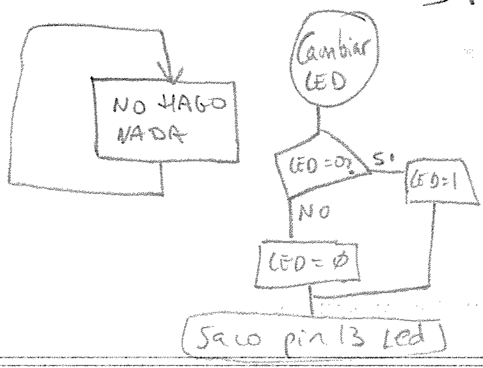
# Ejemplo: Controlar un LED con interrupciones

- Utilizando la API de Arduino ¿Qué habría que hacer para conmutar el LED cada vez que se pulsa el botón?




```
#define LED 13

void setup() {
  pinMode( LED , OUTPUT );
  // Asocio la función toggleLed a la interrupción por PD2
  attachInterrupt( 0, toggleLed, RISING );
}

// Función que se llamará en la interrupción
// No debe tener parámetros
void toggleLed() {
  int ledVal = digitalRead( LED );
  digitalWrite( LED, !ledVal );
}

void loop() {
  // Aquí no hay por qué hacer nada
}
```




Al comprobar el funcionamiento del sistema se observa que al pulsar o soltar, el led cambia de estado de forma no prevista, a veces cambia cuando no debe y otras no cambia cuando debe (debería cambiar de estado sólo cada vez que pulso)
→¿A qué puede ser debido ese comportamiento?
→¿Cómo podría solucionarse?

(A) Ver pag siguiente

setup led = Ø
Definir PIN 13 salida
~~Definir PIN 2 interrup~~
Ejecutar cambiar led en flanco subida
attach interrup(Ø, cambiar LED, RISING)

# Interrupciones temporales y periódicas

Por software

- Los uC disponen de contadores programables que permiten producir interrupciones de forma periódica o pasado un cierto tiempo
- En el caso del AtMega168, dispone de 3 contadores que permiten generar interrupciones periódicas y temporales de diversas formas con periodos desde pocos µseg hasta 32 segs. Sólo vamos a tratar con uno de ellas.

**Funciones de la API de para trabajar con las interrupciones periódicas y temporales**:

- **attachPeriodicInterrupt**( isr_name, period_in_ms ); // Asocia una rutina a una interrupción periódica
  *isr_name*: nombre de la rutina de servicio (función que no debe tener parámetros)
  *period_in_ms*: periodo en milisegundos. Debe ser un entero

- **dettachPeriodicInterrupt**( isr_name ); // Elimina la asociación

- **attachTimedInterrupt** ( isr_name, time_in_ms ); // Asocia una rutina para que se ejecute
  // una sola vez pasado el tiempo especificado

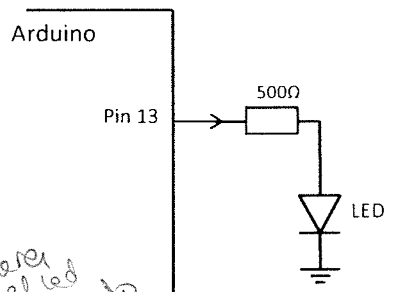
- **dettachTimedInterrupt**( isr_name ); // Elimina la asociación

**Otras funciones asociadas con las interrupciones**:

- **interrupts**(); // Habilita las interrupciones a nivel global
- **noInterrupts**(); // Deshabilita las interrupciones a nivel global

# Ejemplo: Parpadear un LED con interrupciones

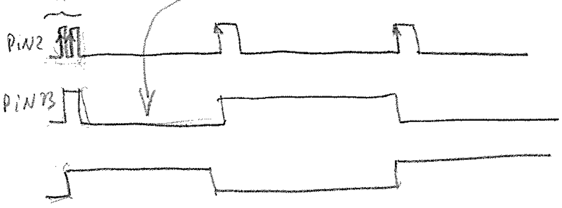
- Utilizando la API de Arduino ¿Qué habría que hacer para conmutar el LED cada 0,5 segs?




```
#define LED 13

void setup() {

  pinMode( LED , OUTPUT );

  // Asocio la función toggleLed a la interrupción periódica
  // que ocurre cada cada 500 milisegundos

  attachPeriodicInterrupt( toggleLed, 500 );
}

// Función que se llamará en la interrupción
// No debe tener parámetros
void toggleLed() {
  int ledVal = digitalRead( LED );
  digitalWrite( LED, !ledVal );
}

void loop() {
  // Aquí se podrían hacer otras cosas, sin que
  // nos interfiera lo que se hace en la interrupción
  ...
  ...
}
```

rebotes para dejar el led apegado

Rebota!!




Introducimos un comando para deshabilitar INT
Espero X ms (delay ...) Vuelvo a habilitar interrup.

# Ejemplo: Comparando soluciones

- Utilizado la API de Arduino ¿Cómo resultaría el control del LED mediante los distintos procedimientos de entrada salida que conocemos (bloqueo, consulta periódica e interrupciones)?

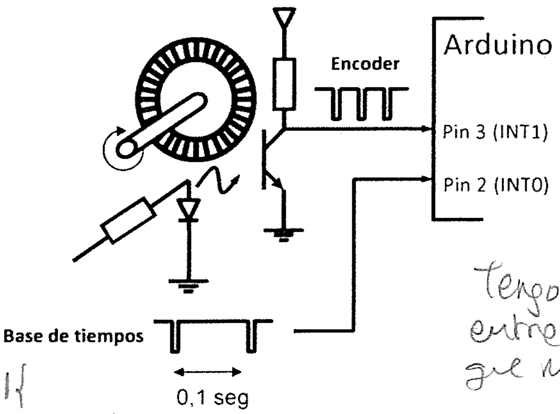
```
// Solución por
//--------------------------
// BLOQUEO DEL PROCCESO
//--------------------------

#define LED 13

void setup() {
  pinMode( LED , OUTPUT );
}

void toggleLed() {
  int ledVal = digitalRead( LED );
  digitalWrite( LED, !ledVal );
}

void loop() {

  // la función sleep espera
  // N ms, de forma bloqueante

  // El código que hay aqui
  // solo se ejecutará cada
  // 0,5 segs
  ...
}
```

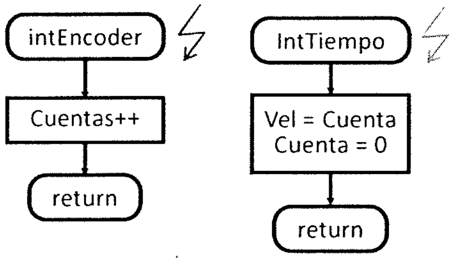
```
// Solución por
//--------------------------
// CONSULTA PERIÓDICA
//--------------------------

#define LED 13

void setup() {
  pinMode( LED , OUTPUT );
}

void toggleLed() {
  int ledVal = digitalRead( LED );
  digitalWrite( LED, !ledVal );
}

unsigned long lastTime = 0;

void loop() {

  // la función millis() devuelve
  // los ms pasados desde el reset

  // Aquí se podrían hacer
  // otras cosas, siempre que no
  // tarden demasiado
  ...
}
```

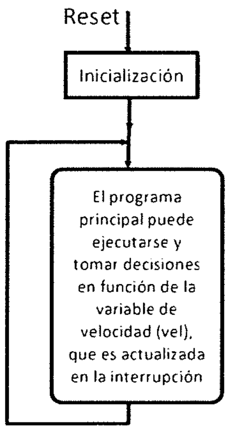
```
// Solución mediante
//--------------------------
// INTERRUPCIONES
//--------------------------

#define LED 13

void setup() {
  pinMode( LED , OUTPUT );

}

void toggleLed() {
  int ledVal = digitalRead( LED );
  digitalWrite( LED, !ledVal );
}

void loop() {

  // Aquí se podrían
  // hacer otras cosas,
  // sin que nos interfiera lo que
  // se hace en la interrupción
  // ni viceversa
  ...
}
```

# Ejemplos con interrupciones: Tacómetro (I)

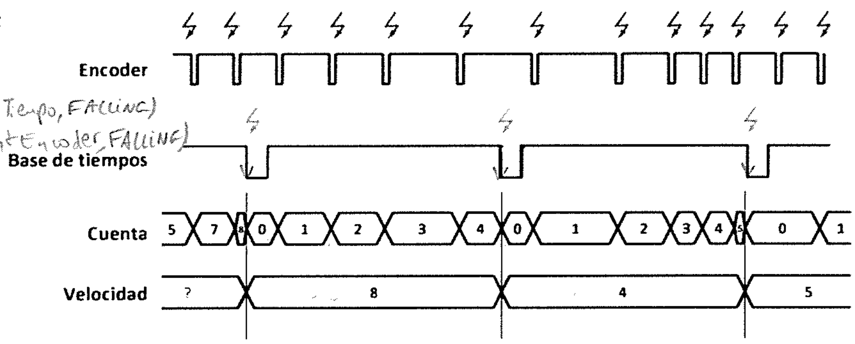
- Utilizando la API y la placa de Arduino, medir la velocidad de giro (en pulsos cada 0.1 segs)




Tengo que contar pulsos entre intervalos en 0,1 s que me marca pin 2




void setup(){
pinMode(3, INPUT);
pinMode(2, INPUT);
attachInterrupt(0, intTiempo, FALLING)
attachInterrupt(1, intEncoder, FALLING)
int cuenta; int vel;
cuenta = 0
vel = 0
}

void intEncoder(){
cuenta ++;
}

void intTiempo(){
vel = cuenta;
cuenta=0;
}

# Ejemplos interrupciones: Tacómetro (II)

ver declaración arriba!!

**Inicialización**

```
void setup() {
  // Encoder
  attachInterrupt( 1, isrEncoder, FALLING );

  // Base de tiempos
  attachInterrupt( 0, isrTimeBase, FALLING );

  // También podríamos utilizar una interrpción
  // periódica y ahorrarnos la circuitría que
  // genera la base de tiempos.
  // En ese caso pondríamos...
  // attachPeriodicInterrupt( isrTimeBase, 100 );

}
```

**ISR de atención a la base de tiempos**

```
// ISR (Interrupt Service Routine)
// de atencion a la base de tiempos

void isrTimeBase() {

  velocidad = cuentas;
  cuentas = 0;

}
```

**ISR de atención al codificador**

```
void isrEncoder() {

  cuentas = cuentas + 1;

}
```

**Programa principal**

```
volatile int cuentas;
volatile int velocidad;

void loop() {

  // Aquí, por ejemplo, haríamos algo en
  // función de la velocidad

  if( velocidad > VALOR_LIMITE) {
    acciones en este caso...
  } else {
    acciones en caso contrario...
  }

  // o podríamos mostrar el valor por un display

  display(velocidad);
  ...

}
```

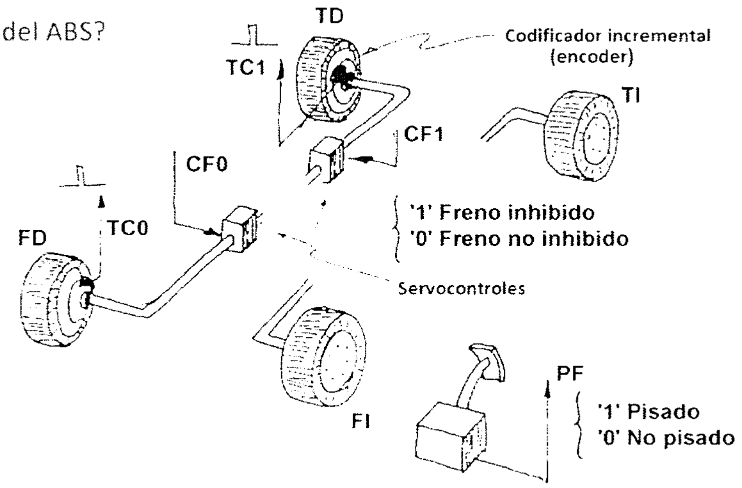
**main**

```
// Esto lo incluye automáticamente el IDE
// de Arduino

int main() {
  setup();  // esto solo se hará una vez

  for(;;) { // el resto en un bucle infinito
    loop();
  }
}
```

# Ejemplos interrupciones: Control ABS (simplificado)

- Se pretende realizar un sistema antibloqueo de frenada en un vehículo. Para ello, se quiere medir la velocidad de giro de las ruedas (sólo dos en el ejemplo) y, en caso de que una patine (gire más despacio que las otras), inhibir el frenado en dicha rueda.
- Se dispone de:
  - Dos encoders (TC0 y TC1) en las ruedas que dan 1 pulso cada 10 ms por cada kilómetro por hora de velocidad del vehículo.
  - Un detector de pedal de freno pisado
  - Dos servocontroles (CF0 y CF1) que pueden inhibir la frenada en su rueda correspondiente
- ¿Cómo podríamos realizar el control del ABS?

Nota: El sistema propuesto está muy simplificado respecto a la realidad (en la realidad se usan más sensores, como acelerómetros, giróscopos, se controlan todas las ruedas, etc..). En esta versión simplificada ¿qué problemas de funcionamiento son predecibles?




# Ejemplos: Control ABS (usando API de Arduino)

**Inicialización**

```
// Pines de E/S
#define CF1 6
#define CF0 5
#define PF  4
#define TC1 3
#define TC0 2

// Variables utilizadas
volatile int cuentasFD;
volatile int cuentasTD;
volatile int velFD,
volatile int velTD;

void setup() {
  // E/S
  pinMode(CF0, OUTPUT);
  pinMode(CF1, OUTPUT);

  // Encoders
  attachInterrupt( 0, IsrTC0, RISING);
  attachInterrupt( 1, IsrTC1, RISING);

  // Base de tiempos.
  attachPeriodicInterrupt( isrTimeBase, 10 );

}
```

**ISRs de atención a los codificadores**

```
void IsrTC0() {
  cuentasFD = cuentasFD + 1;
}

void IsrTC1() {
  cuentasTD = cuentasTD + 1;
}
```

**Programa principal**

```
void loop() {

  if ( digitalRead(PF) != 0 ) { // ¿Freno pisado?
    if( velFD != velTD) { // ¿Alguna rueda patina?
      if (velFD > velTD) { // ¿Patina la trasera?
        digitalWrite(CF1, HIGH); // Inhibo trasera
      } else { // si no, es la delantera
        digitalWrite(CF0, HIGH); // Inhibo delantera
      }
    } else { // no patinan
      digitalWrite(CF0, LOW); // desinhibo frenos
      digitalWrite(CF1, LOW);
    }
  } else {
    // si no frena, por si acaso deshninibo ¿?
    digitalWrite(CF0, LOW);
    digitalWrite(CF1, LOW);
  }

}
```

**ISR de atención a la base de tiempos**

```
// ISR (Interrupt Service Routine)
// de atencion a la base de tiempos

void isrTimeBase() {

  velFD = cuentasFD;
  velTD = cuentasTD;
  cuentasFD = 0;
  cuentasTD = 0;

}
```

# MICRO

- Diferencia entre $\mu P$ y $\mu C$
  - Un $\mu P$ es un circuito de computación integrado en un chip. Contiene la ALU, el banco de registros y la unidad de control
  - Mientras que un $\mu C$ es un circuito digital integrado – Todo el sistema en un chip – que incluye un $\mu P$, memoria y dispositivos periféricos. Puede programarse vía software y se utiliza generalmente para aplicaciones de bajas prestaciones, más pequeñas que por ejemplo un ordenador, para el que se emplea un $\mu P$.
- ¿Cantidad de memoria tiene típicamente un $\mu C$?

  Entre 512 bytes y 8 Kbytes
- Qué se vende más ¿$\mu C$ de 8 o 32 bits?

  55% son de 8 bits y menos del 10% tienen 32 o más.
- ¿Cuál/es son sist. empotrados? Lavadora, PDA, ordenador, máq. tabaco, teléfono móvil.

  Un sistema empotrado es aquel que integra un $\mu P/\mu C$ dedicado a una tarea específica → lavadora, máq. tabaco, telf. móvil (antiguo)
- En un $\mu C$ de 8 bits ¿cuál es el tamaño del bus de datos?

  Obviamente, 8 bits. Es el tamaño de la palabra del $\mu P$
- Señales típicas de un bus de control de un $\mu P$.

  Reset, $R/\overline{W}$ (read/write), clock reloj, interrupciones, ...
- En un $\mu P$ de 16 bits ¿Cuántos bits son necesarios para direccionar 32 KB de memoria?

  Capacidad de $\mu P = 2^M \cdot N$ $\left\{ \begin{array}{l} N = 16 \text{ (tamaño del bus de datos - palabra del } \mu P) \\ Cap = 32 \text{ KB ¿Kbytes o Kbits?} \end{array} \right.$ — Ⓐ Kbytes / Ⓑ Kbits

Ⓐ $Cap = 32 \text{ Kbytes} = 32000 \cdot 8 \text{ bits} = 2^M \cdot 16 \rightarrow \boxed{M = 14 \text{ bits}}$

Ⓑ $Cap = 32 \text{ Kbits} = 32000 \text{ bits} = 2^M \cdot 16 \rightarrow \boxed{M = 11 \text{ bits}}$

- ¿Cuál suele ser el cuello de botella de un $\mu P$?

El bus de datos ya que todas las transf. de información han de pasar por ahí. La cantidad de info viene limitada por el ancho de banda o tamaño del bus (palabra con la que trabaja el $\mu P$)

- ¿Cuál de los buses de un $\mu P$ son bidireccionales?

El bus de datos lo es siempre.
El bus de control puede tener algunos bits bidireccionales
El bus de direcciones sólo es bidireccional en sistemas multiprocesador.

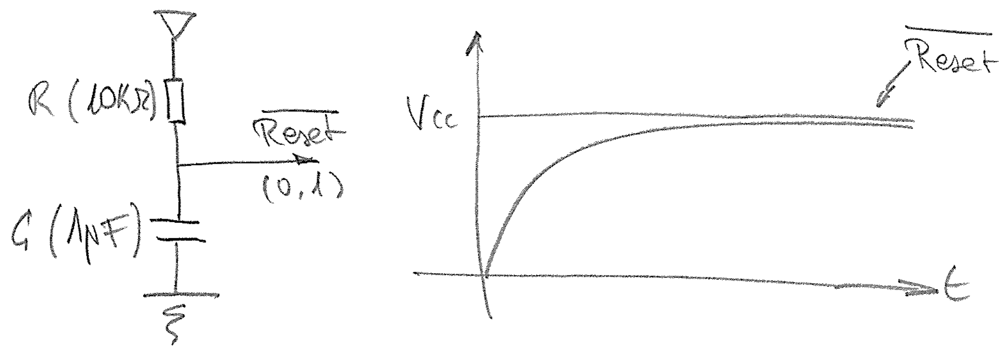
- ¿Qué elementos componen un sistema mínimo $\mu P$?

Un microprocesador, memoria, unidades E/S, bus de datos direcciones y control.

- Tipos de dispositivos típicos en un $\mu C$.

Temporizadores y contadores.
E/S paralelo
E/S serie
Perro guardián
Convertidores A/D D/A
Controladores específicos

- Esquema power-on reset




- En un puerto E/S paralelo, si una entrada no se utiliza, ¿puede dejarse al aire? ¿por qué?

No pueden dejarse al aire. Hay que conectarlas mediante una resistencia a un valor alto (pull-up) o bajo (pulldown) para fijar la entrada a un determinado valor lógico.

- ¿Consumo de un µC de gama baja?

  ≈ milivatios (68HC11, PIC o similares)

- ¿Y uno tipo Pentium, AMD o similar?

  ≈ vatios

- Elementos de un µP

  Unidad Aritmético Lógica ALU, Banco de registros

  Unidad de control UC, Decodificador de instrucciones

- Pasos típicos en la ejecución de una instrucción en un micro?

  Búsqueda de instrucción → Decodificación → Operación con la ALU → Transferencia de registros → Incremento de contador de programa

- Diferencia entre compilador y ensamblador

  El compilador crea código a partir de lenguaje de alto nivel (C, C++, ...) mientras que el ensamblador parte de un código mnemónico. En ocasiones el código de salida del compilador pase al ensamblador para después poder ser leído por la máquina (código máquina)

- Principales procedimientos E/S

  Bloqueo de proceso // Consulta periódica (polling) // Por interrupción

- Dentro del ciclo de ejecución de una instrucción ¿en qué momento se determina si existe alguna interrupción pendiente?

  Cada vez que se termina una instrucción se comprueba si hay alguna interrupción pendiente

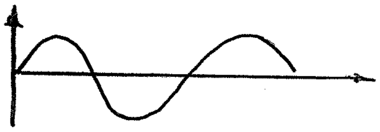
# Dispositivos y Sensores

## 1 Señales de los sistemas electrónicos

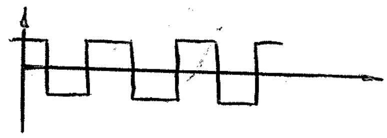
**Intro:**

En un sistema electrónico la información la llevan señales eléctricas en forma de:
- Amplitud (niveles de tensión o corriente)
- Frecuencia

**Señales Continuas o Analóg.**
- Varían de forma ininterrumpida
- Pueden tomar un nº infinito de valores



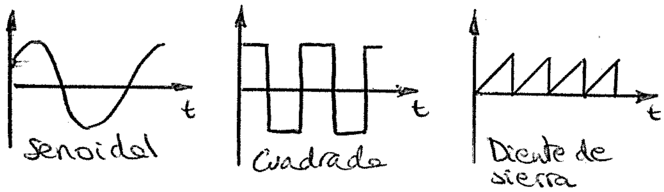
**Señales Discretas o digit.**
- Pueden tomar sólo un nº discreto de valores
- Binarias: El nº de valores posible es 2.

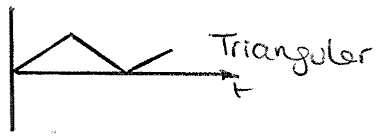


- La magnitud representada en las señales puede ser proporcional a la amplitud de la señal en tensión o corriente, y también en frecuencia
- Las magnitudes físicas varían de forma continua. A veces interesa discretizarlas ya que las señales discretas son más inmunes al ruido y a la distorsión

**Periódicas**: Se repiten en el tiempo, siendo su periodo el tiempo que transcurre entre dos instantes en los que la señal pasa por el mismo punto

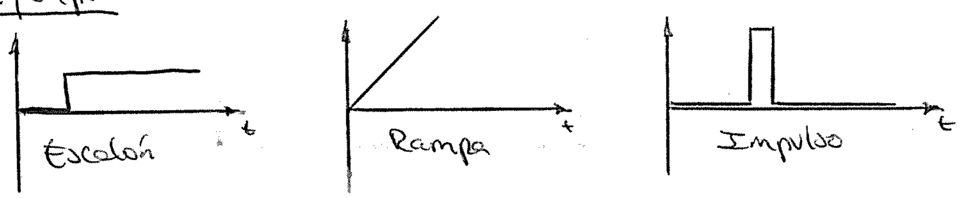
Ejemplos:



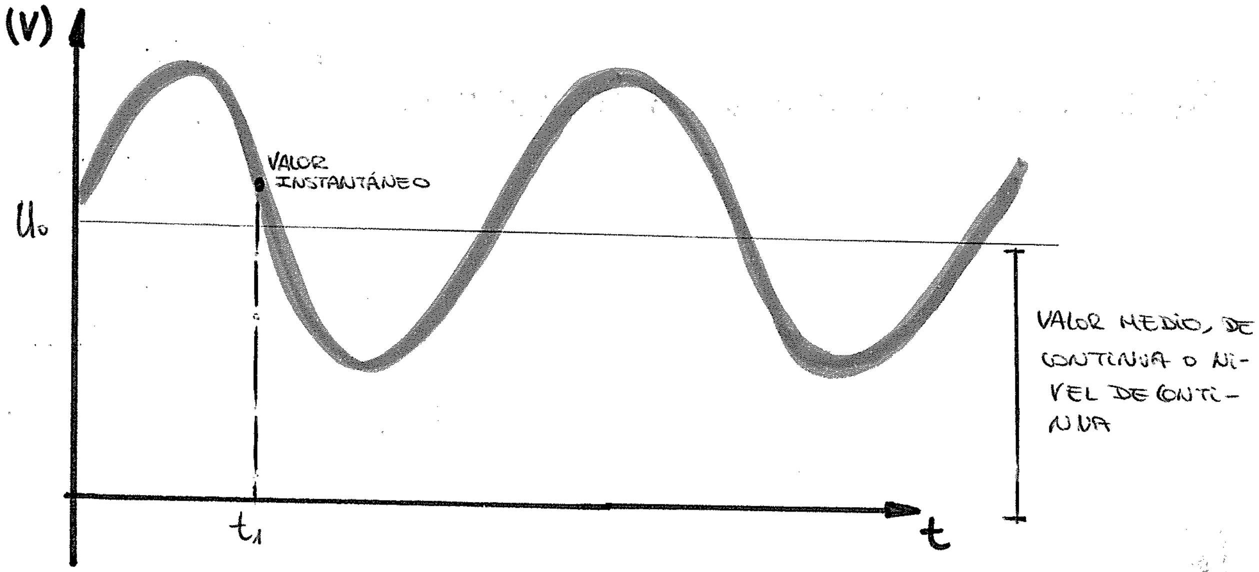




**Aperiódicas**: No tienen repetitividad.

Ejemplos:

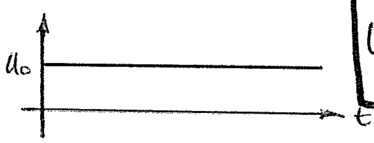


## PARÁMETROS TÍPICOS DE LAS SEÑALES.



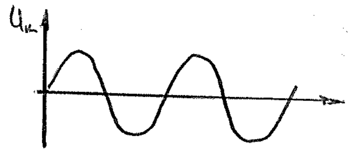

<u>Valor instantáneo</u>: Valor que toma la señal en cada instante

<u>Componente continua</u>: Valor medio de la señal




$$\boxed{U_0 = \frac{1}{T}\int_0^T u(t)\,dt}$$

<u>Componente alterna</u>: Lo que queda de la señal al quitar la componente continua.

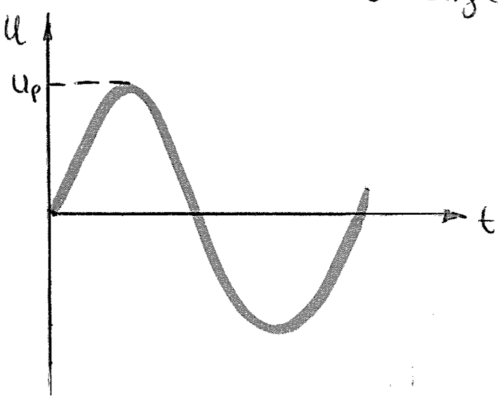



El valor medio de la componente alterna es cero.

Cualquier señal periódica podrá descomponerse en: Parte continua + parte alterna.

Ej: $\boxed{u(t) = U_0 + U_m \,\mathrm{sen}(\omega t) \;;\; \omega = 2\pi f,\; f = \frac{1}{T}}$

<u>Valor eficaz</u>: Valor de tensión continua que haría disipar la misma potencia, sobre una carga resistiva, que la onda estudiada.

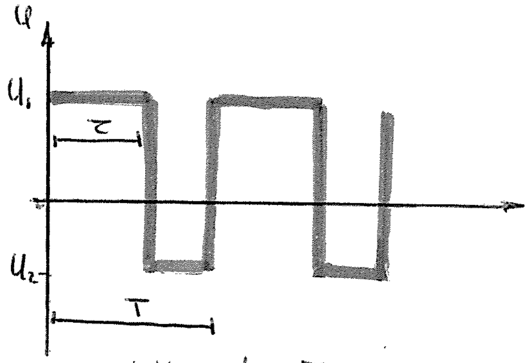



$$\boxed{U_{ef} = \frac{U_p}{\sqrt{2}}} \qquad P = \frac{U_{ef}^2}{R} = \frac{U_p^2}{2R}$$

Se calcula como

$$\boxed{U_{ef} = \sqrt{\frac{1}{T}\int_0^T u^2(t)\,dt}}$$

**Ejemplo:** Calcular el valor medio y el eficaz




$$U_{medio} = \frac{1}{T}\int_0^T u(t)\,dt = \frac{1}{T}\left[\int_0^{\tau} u_1\,dt + \int_{\tau}^{T} u_2\,dt\right] =$$

$$= \frac{1}{T}\left(u_1\tau + u_2(T-\tau)\right)$$

$$\boxed{U_{medio} = u_1\frac{\tau}{T} + u_2\frac{T-\tau}{T}}$$

$$u(t) = \begin{cases} u_1 & t<\tau \\ u_2 & \tau<t<T \end{cases}$$

$$U_{ef}^2 = \frac{1}{T}\int_0^{\tau} u_1^2\,dt + \frac{1}{T}\int_{\tau}^{T} u_2^2\,dt = \frac{1}{T}u_1^2\tau + \frac{1}{T}u_2^2(T-\tau); \quad \boxed{U_{ef} = \sqrt{\frac{1}{T}\left(u_1^2\tau + u_2^2(T-\tau)\right)}}$$

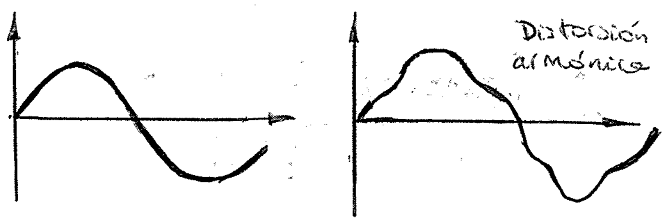
**NOTA TEÓRICA!!** Espectro de Frecuencias: ANÁLISIS DE FOURIER

Una forma de onda periódica podemos descomponerla siempre en una componente continua y otra alterna con → componente fundamental: misma frecuencia que señal en sí (f)
→ componentes armónicos: señales cuya frecuencia son múltiplos de la fundamental (3f, 5f, ...)

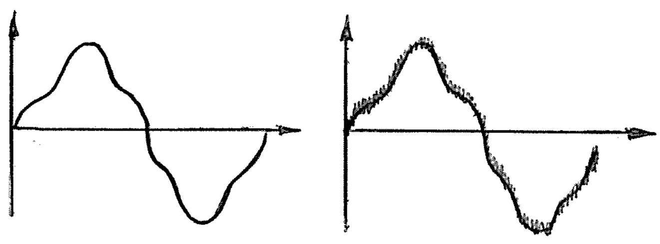
## ● RUIDO Y DISTORSIÓN

Los sistemas electrónicos tienen limitaciones respecto de las amplitudes y frecuencias que pueden pasar a través de ellos.

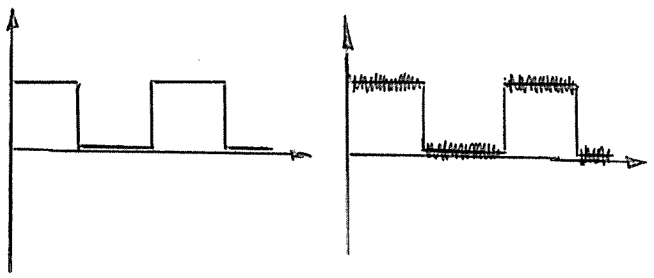
Como consecuencia las señales presentan distorsiones alterando la amplitud, la frecuencia o la fase de una señal.




El ruido también puede afectar a las señales. Es una fluctuación aleatoria de una señal que se produce por variaciones dentro del sistema o por efectos externos del medio.



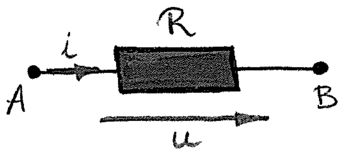
Las señales discretas son más inmunes al ruido. Sólo hay que discriminar entre los niveles alto y bajo que, aunque exista ruido, no se solapan



# 2 COMPONENTES DE LOS SISTEMAS ELECTRÓNICOS

- Componentes pasivos: No son exactamente componentes electrónicos (no son semiconductores) pero se usan en la electrónica. Ejemplos: Resistencias, bobinas, condensadores, ...
- Componentes electrónicos: son componentes basados en semiconductores
  - Discretos: Integran un dispositivo (diodos, transistores, tiristores, ...)
  - Circuitos integrados: Integran gran cantidades de transistores (microprocesadores, memorias, ...)

## ● COMPONENTES PASIVOS – $\boxed{\text{Resistencias}}$



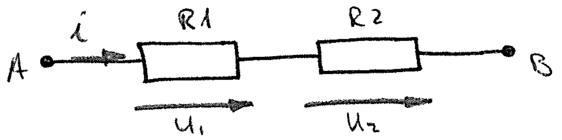

Ley de OHM $\boxed{U = R \cdot i}$

Potencia $\boxed{P = U \cdot i}$

$[R] \equiv \Omega$ Ohmios

Valores típicos ($m\Omega$ – $M\Omega$)

### Divisor de tensión ~ Asociación en serie

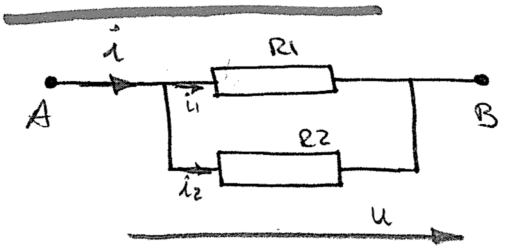



$R_{AB} = R_1 + R_2$

$$\boxed{\begin{aligned} u_1 &= U_{AB} \cdot \frac{R_1}{R_1+R_2} \\ u_2 &= U_{AB} \cdot \frac{R_2}{R_1+R_2} \end{aligned}}$$

$$i = \frac{U_{AB}}{R_1+R_2}$$

### Divisor de corriente ~ Asociación en paralelo

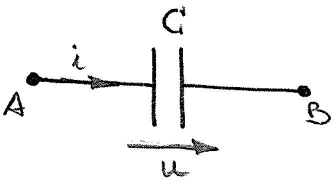



$$R_{AB} = \frac{R_1 R_2}{R_1+R_2}$$

$$\boxed{\begin{aligned} i_1 &= i \cdot \frac{R_2}{R_1+R_2} \\ i_2 &= i \cdot \frac{R_1}{R_1+R_2} \end{aligned}}$$

$$u = i\,\frac{R_1 R_2}{R_1+R_2}$$

## COMPONENTES PASIVOS — Condensadores




$$\boxed{q = C \cdot u} \xrightarrow{\text{Derivando}} \frac{dq}{dt} = i(t) = C\frac{du}{dt} \; ; \; \boxed{i(t) = C\frac{du(t)}{dt}}$$

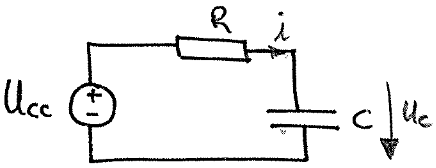
[C] ≡ Faradios

Típicamente pF – μF

$$u(t) = u_0(t) + \frac{1}{C}\int_0^t i(t)\,dt \; ; \; \boxed{E = \frac{1}{2}Cu^2}$$

Energía almacenada

### Análisis circuito R-C

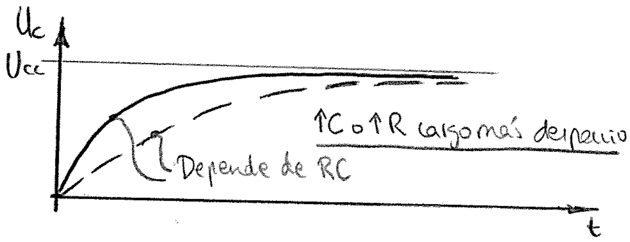


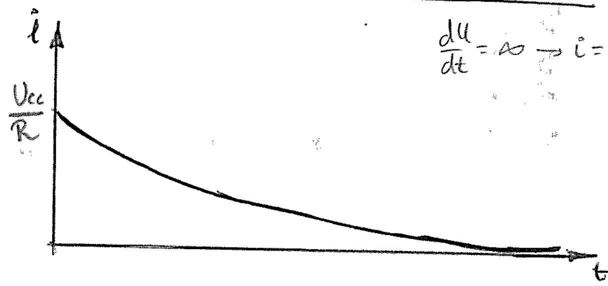

$$u_c(t) = u_c(0) + \left(U_{cc} - u_c(0)\right)\left(1 - e^{-\frac{t}{RC}}\right)$$

$\boxed{\tau = RC}$ constante de tiempo.

La tensión en un condensador no puede variar bruscamente!!

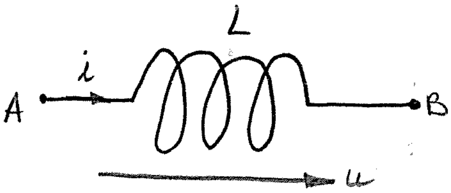
$\frac{du}{dt} = \infty \rightarrow i = \infty$ Quemo el circuito!!




- En régimen permanente los condensadores se comportan como circuitos abiertos
- En el instante inicial el condensador es un corto circuito.

## COMPONENTES PASIVOS — Bobinas




$$u(t) = N\frac{d\phi}{dt} \longrightarrow \boxed{u(t) = L\frac{di(t)}{dt}}$$

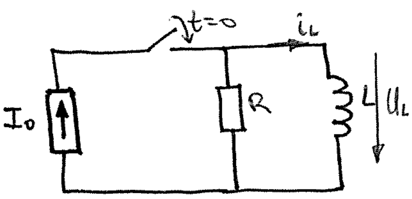
$$i(t) = i(0) + \frac{1}{L}\int_0^t u(t)\,dt \; ; \; \boxed{E = \frac{1}{2}Li^2}$$

Energía almacenada

[L] = Henrios

Típicamente μH – mH

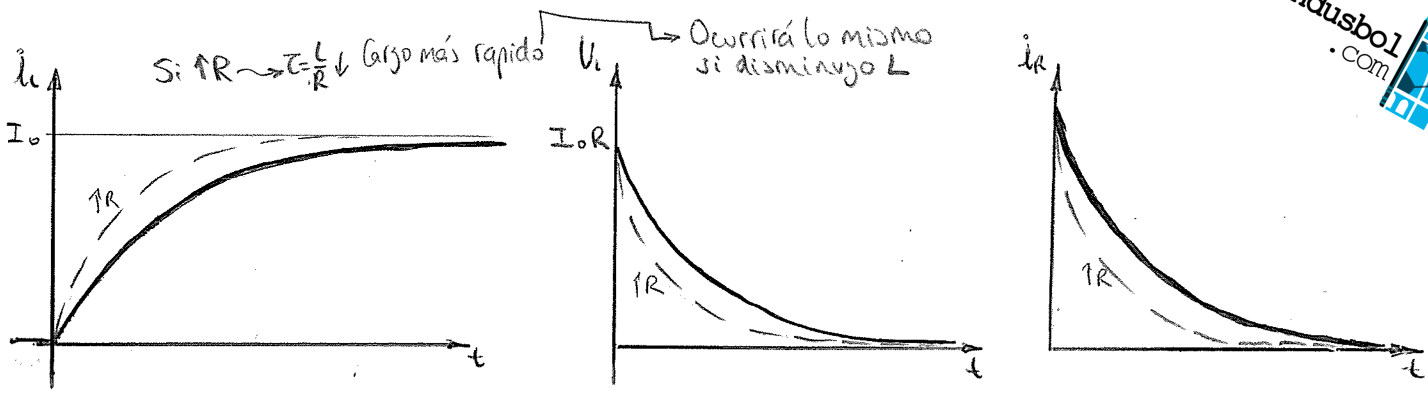
### Análisis circuito R-L




$$i_L(t) = i_L(0) + \left(I_0 - i_L(0)\right)\left(1 - e^{-t/\frac{L}{R}}\right)$$

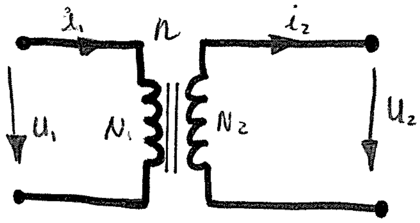
$\boxed{\tau = \frac{L}{R}}$ Constante de tiempo

La intensidad en una bobina no puede variar bruscamente!!




- En t → ∞ la bobina se comporta como un cortocircuito (régimen permanente)
- En el instante inicial la bobina es un circuito abierto

## TRANSFORMADORES



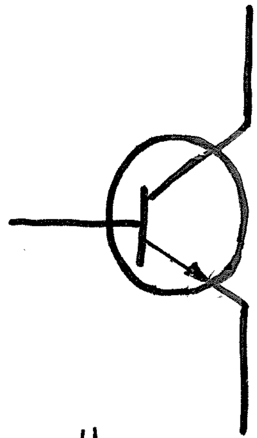

$$\boxed{\begin{aligned} n &= \frac{N_1}{N_2} \\ i_1 N_1 &= i_2 N_2 \\ u_1 &= n\, u_2 \end{aligned}}$$

$\boxed{P_1 = P_2 \text{ (sin pérdidas)}}$

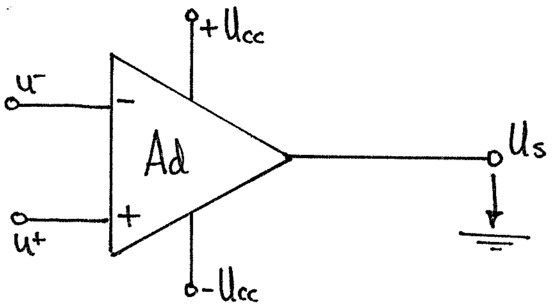
## COMPONENTES DISCRETOS

Diodos

Transistores



Amplificador Operacional




## CIRCUITOS INTEGRADOS

Existe una gran variedad de chips en el mercado, ya que el coste final se reduce a fabricarlos en serie

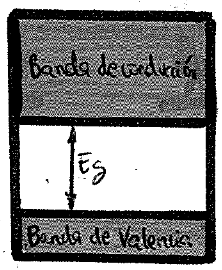
Las ventajas de integrar incluyen, además del coste: Reducción de peso
Aumento de prestaciones
Reducción de consumo
Confidencialidad

Cabe destacar los microprocesadores programables mediante software. Se utilizan tanto en ordenadores como en sistemas embebidos (móviles, lavadoras, ...)

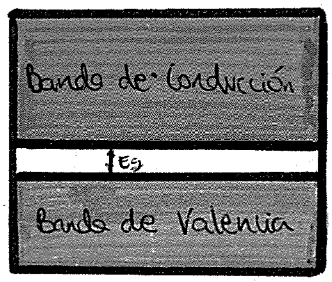
# 3 DIODOS DE SEMICONDUCTOR

## INTRO: Semiconductores

Aislante ($SiO_2$)
$E_g = 9eV$




Semiconductor (Si)
$E_g = 1.1eV$




Conductor (Grafito)
No hay $E_g$

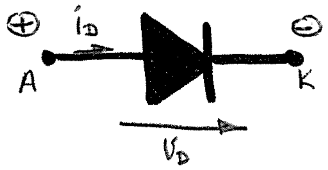

Nivel de energía de un $e^-$ libre: Banda de conducción
Nivel de energía de un $e^-$ en enlace covalente: Banda de valencia

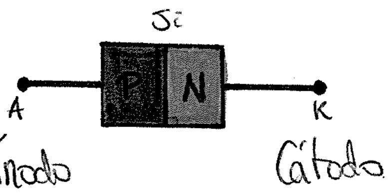
A altas temperaturas algunos $e^-$ de la banda de valencia de los semiconductores alcanzan el nivel de energía de la banda de conducción

Por tanto, los semiconductores pueden comportarse como aislantes o como conductores.

Son elementos del grupo 4, principalmente silicio.

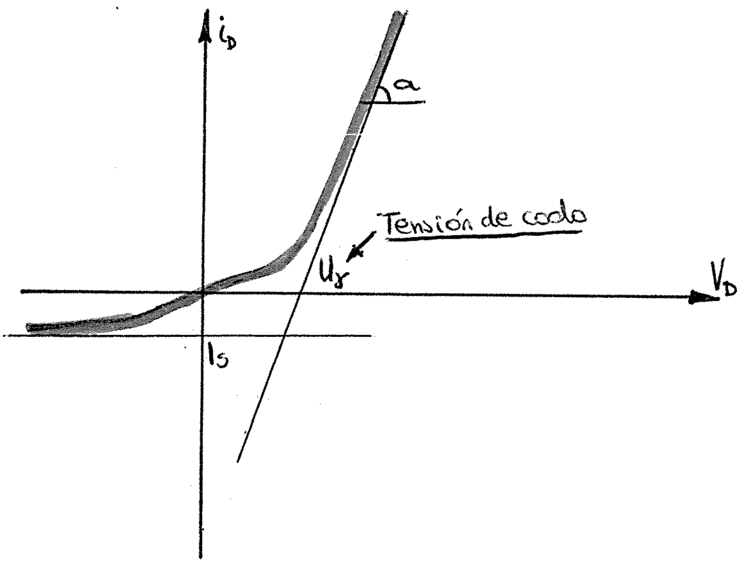
## SÍMBOLO, FUNCIONAMIENTO Y CURVA CARACTERÍSTICA



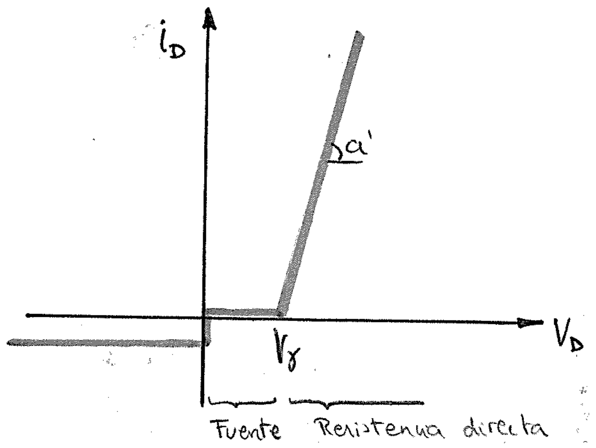




- Cuando $V_D > V_\gamma$ ($\approx 0{,}6V$)
  Circula corriente sentido ánodo cátodo (polarización directa)

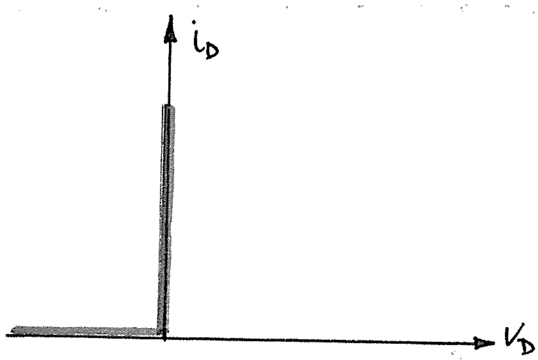
- Cuando $V_D < 0$: $i_D \approx I_s$
  Corriente inversa muy pequeña ya que el diodo es prácticamente unidireccional



## Característica linealizada



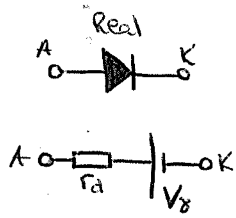

## Característica ideal




$V_D > V_\gamma \rightarrow V_D = V_\gamma + r_D \cdot i_D$

$0 < V_D < V_\gamma \rightarrow i_D = 0$

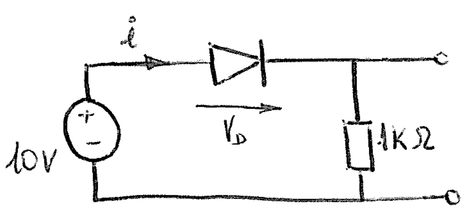
$V_D < 0 \rightarrow i_D = i_S (<0)$

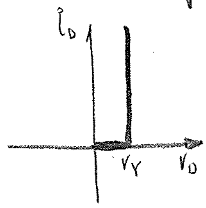


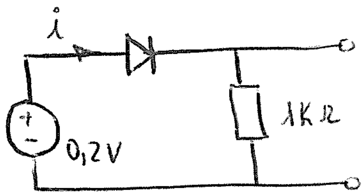

$V_D > 0 \rightarrow i_D > 0$ Cortocircuito


$V_D < 0 \rightarrow i_D = 0$ Circuito abierto


**Ejemplo:** Calcular la intensidad para cada uno de los circuitos



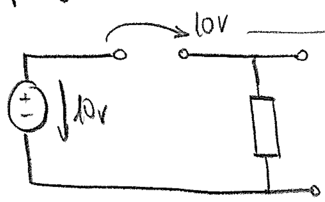




Tensión de codo $V_\gamma = 0{,}6\,V$

Resistencia directa $r_D = 0\,\Omega$

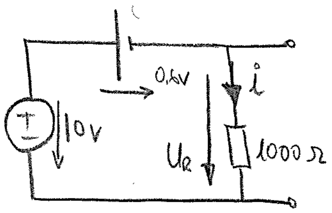
Supongo que el diodo no conduce → circuito abierto




Con 10V el diodo conduce

↓

Lo sustituimos por una fuente de tensión de valor $V_\gamma = 0{,}6V$

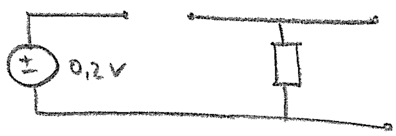



$U_R = 10 - 0{,}6 = i \cdot 1000$

$\boxed{i = 9{,}4 \cdot 10^{-3} = 9{,}4\,mA}$

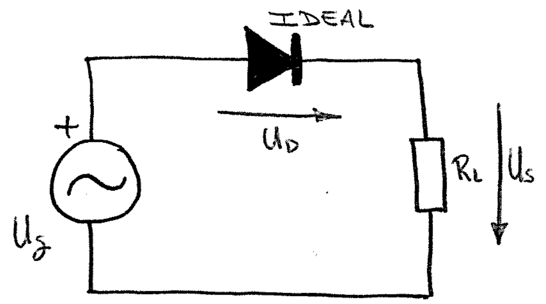
Con 0,2 V el diodo no conduce pues $V_D < V_\gamma$ $(0{,}2 < 0{,}6)$

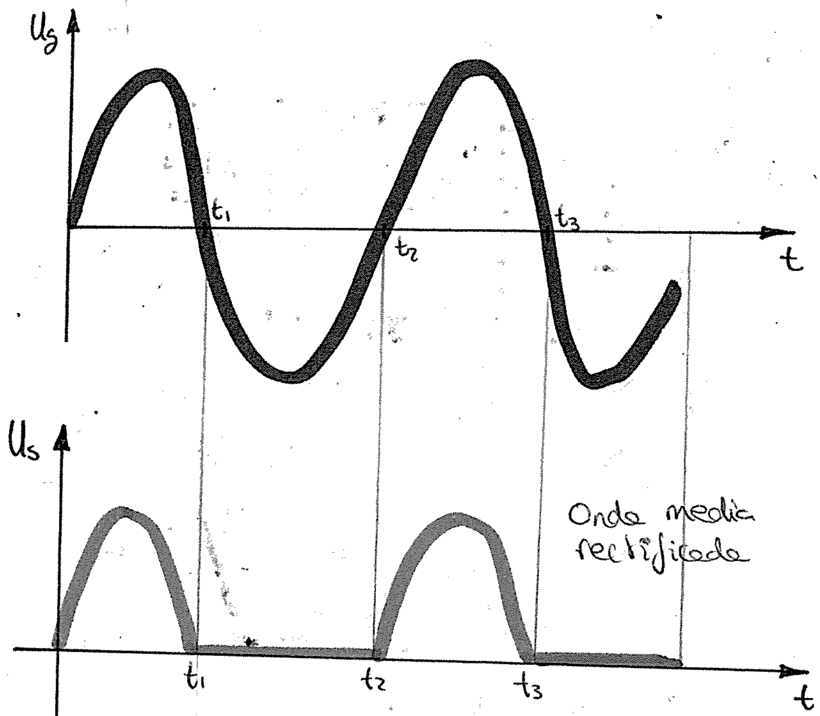
El diodo es por tanto un circuito abierto



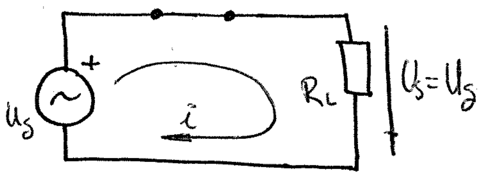

$\boxed{i = 0}$

# Circuitos con diodos: Rectificador de media Onda



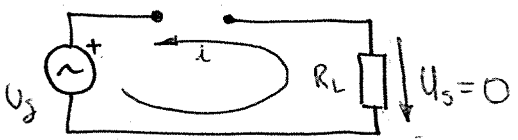




Semiciclo Positivo (Intensidad en sentido horario → D conduce)



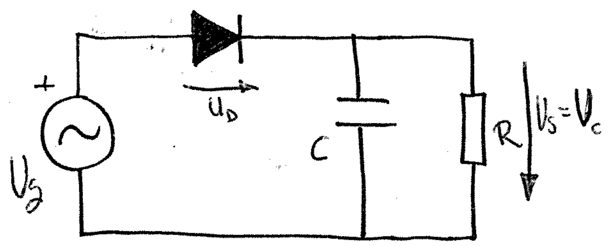

$0 < t < t_1 \longrightarrow U_g > 0 \; ; \; U_D \geq 0$

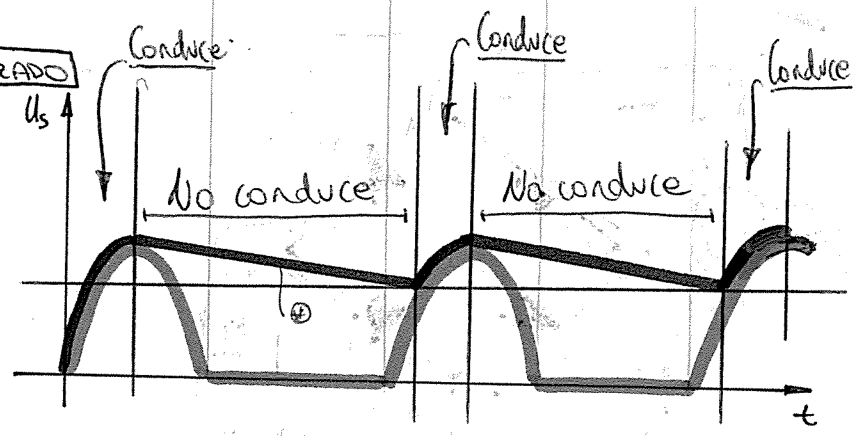
Semiciclo negativo (Intensidad en sentido antihorario → D NO conduce)




$t_1 < t < t_2 \; ; \; U_g < 0 \; ; \; U_D < 0$

Añadimos un condensador → $\boxed{\text{Rizado}}$



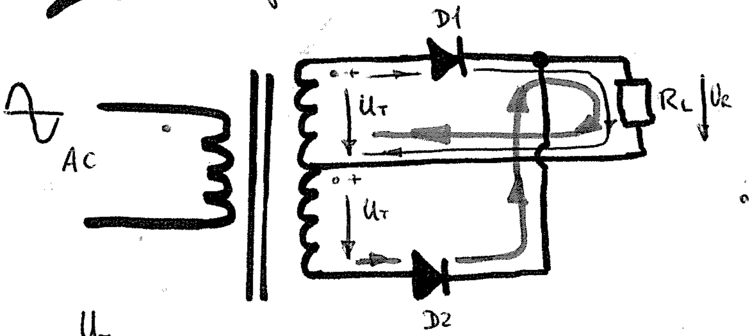

El diodo conduce mientras $U_g > U_c$

(*) En realidad son exponenciales de descarga del condensador.

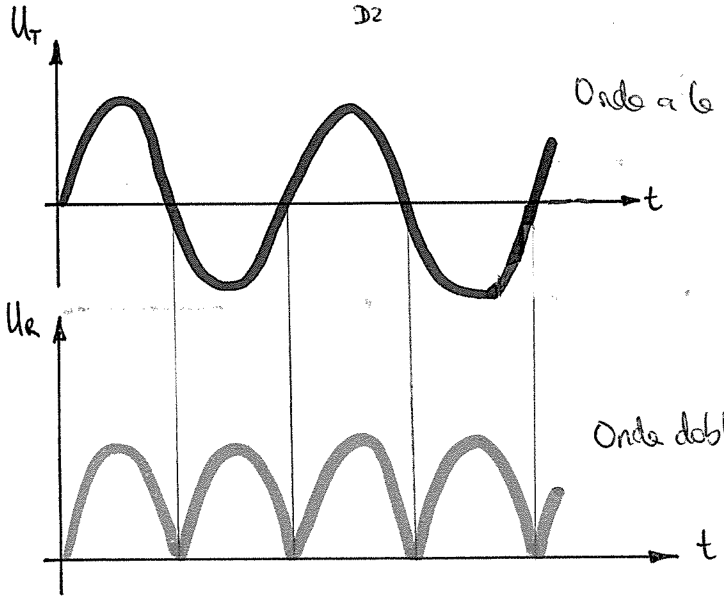
Consigo en la carga una tensión casi continua excepto en el ciclo principal

# Circuitos con diodos: Rectificador de doble onda

## I) Con transformador de toma media



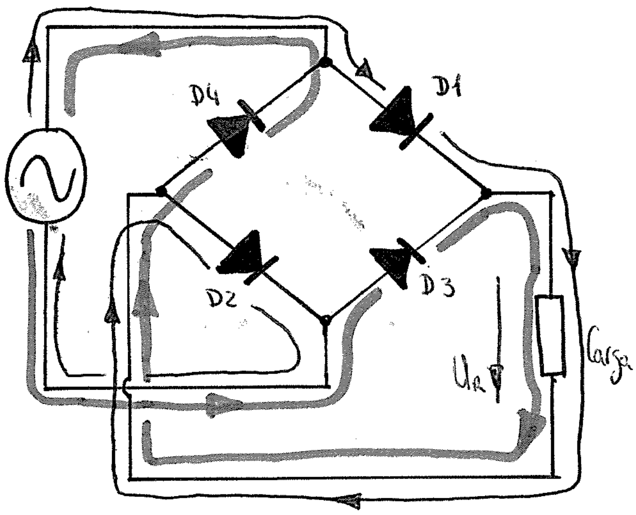

- Semiciclo positivo
Conduce D1, D2 no conduce
- Semiciclo negativo
Conduce D2, D1 no conduce




Onda a la salida del trafo

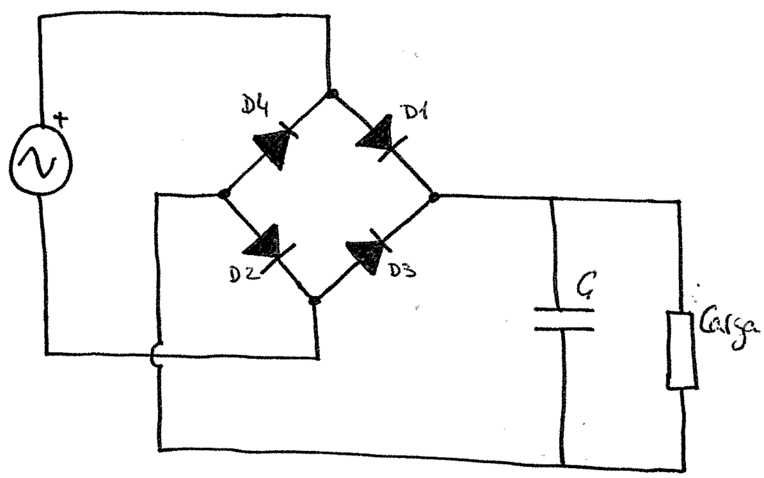
Onda doble rectificada

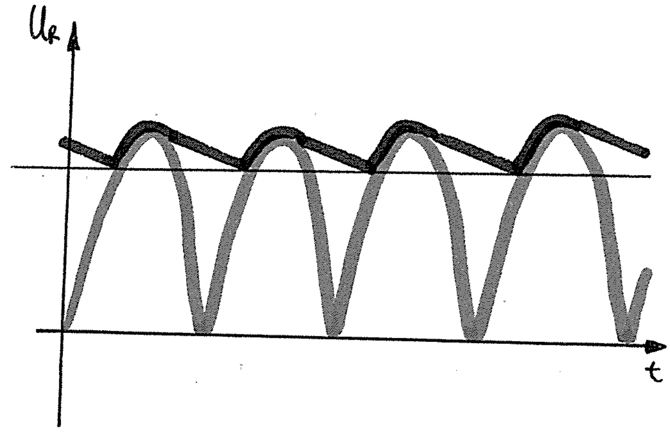
## II) Usando 4 diodos




- Semiciclo positivo: I
Conducen D1 y D2
D3 y D4 bloqueados
- Semiciclo negativo II
Conducen D3 y D4
D1 y D2 bloqueados

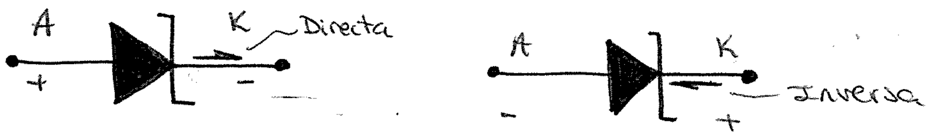
## AÑADIMOS UN CONDENSADOR → $\boxed{\text{Rizado}}$



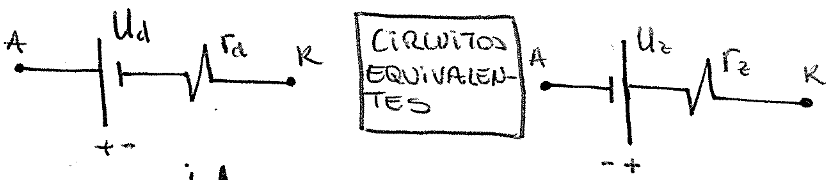




Ahora sí, la tensión en la carga es prácticamente continua

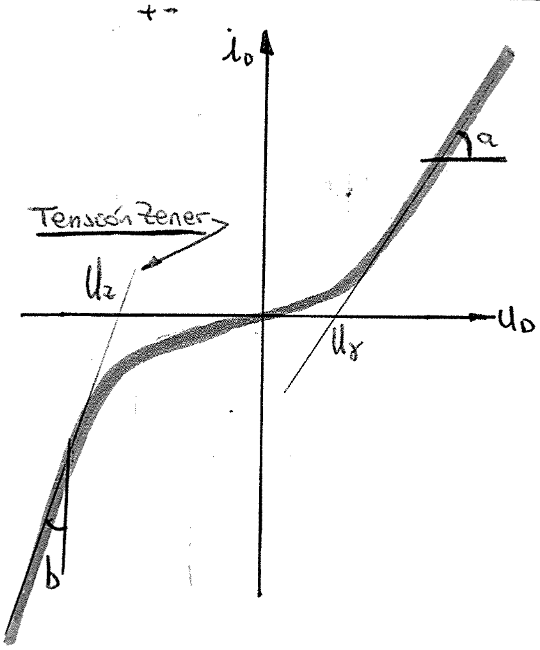
## ● DIODOS ZENER




Se trata de un diodo que conduce tanto en directa como en inversa




Se usan para estabilizar tensiones.

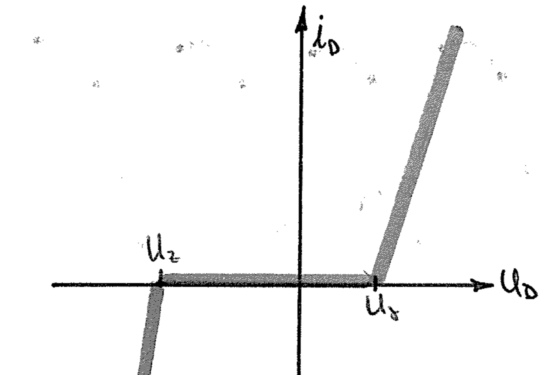



$U_D > U_\gamma$ : Se comporta como un diodo normal

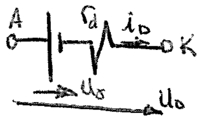
$U_z < U_D < 0$ : Conduce en inversa

## CURVA CARACTERICA ZENER

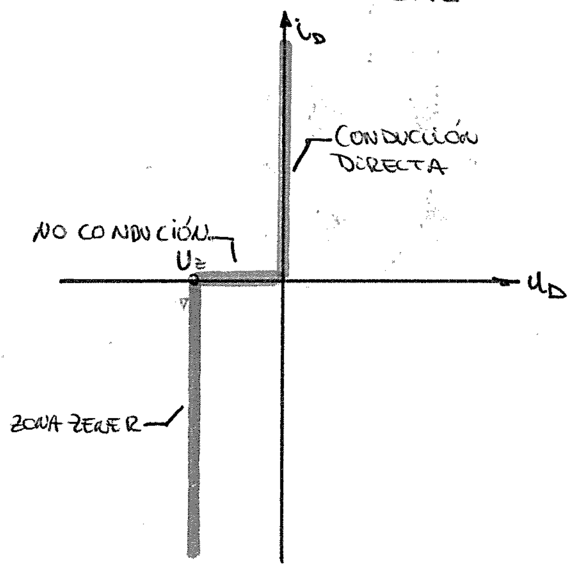
CARACTERÍSTICA LINEALIZADA




- $U_D > U_\gamma$ : $i_D = \dfrac{U_D - U_\gamma}{r_D}$

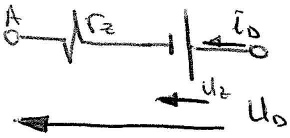



CARACTERÍSTICA IDEAL




- $U_z < |U_D|$

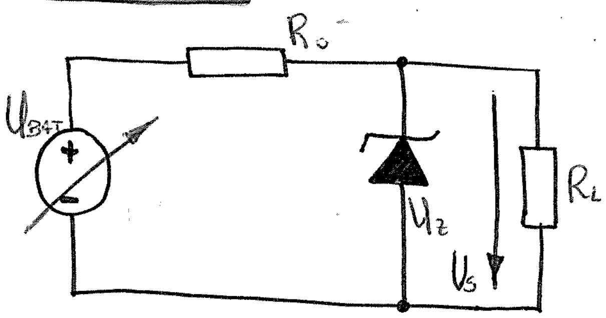
$i_D = \dfrac{|U_D| - U_z}{r_z}$




- $U_z < U_D < U_\gamma$ : No conduce

$i_D = 0$

**Ejercicio: Estabilizador de tensión**




DATOS:

$U_{BAT} = 10 \div 12$ V (Variable entre esos valores)

$R_o = 100\,\Omega$

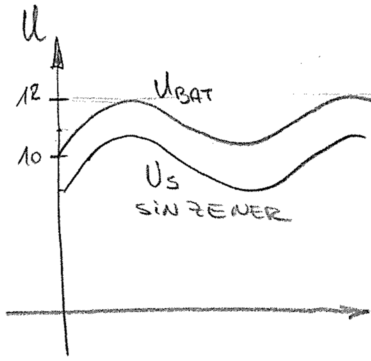
$R_L = 1\,K\Omega$

Zener ideal, $U_z = 7V$

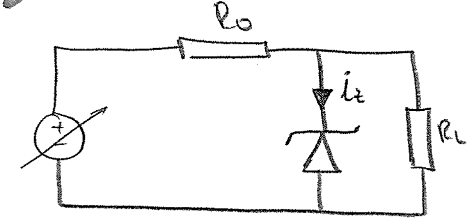
**a)** $U_s$ si no hubiera zener

Div. de tensión

$$U_s = U_{BAT} \cdot \frac{R_L}{R_o + R_L} \ ; \ \boxed{U_s = U_{BAT} \cdot \frac{10}{11} = 9{,}1 \div 10{,}9 \ V}$$

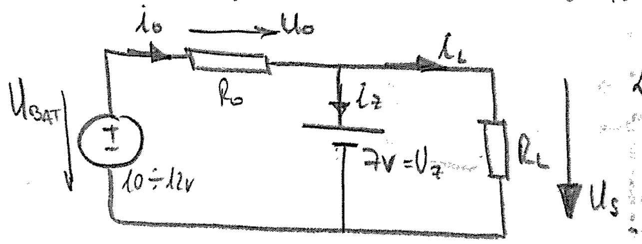


**b)** Conectando el zener, calcular la intensidad que circula por él.




Suponemos que zener conduce en zona zener → condición: $i_z > 0$ en el sentido de la figura

⇓ ~ zener ideal = $U_z = 7V$




$$\left\{ \begin{array}{l} i_L = \dfrac{U_s}{R_L} = \dfrac{7}{1000} = 7\ mA \\ U_s = U_z = 7V \end{array} \right.$$

$$i_o = \frac{U_o}{R_o} = \frac{1}{R_o}\left(U_{BAT} - U_z\right) \quad \left\{ \begin{array}{l} i_{o\,mín} = \dfrac{10-7}{100} = 30\ mA \longrightarrow \boxed{i_{z\,mín} = 23\ mA > 0} \\ i_{o\,máx} = \dfrac{12-7}{100} = 50\ mA \longrightarrow \boxed{i_{z\,máx} = 43\ mA > 0} \end{array} \right.$$

$$i_z = i_o - i_L$$

El zener conduce siempre en zona zener → Correcta la suposición

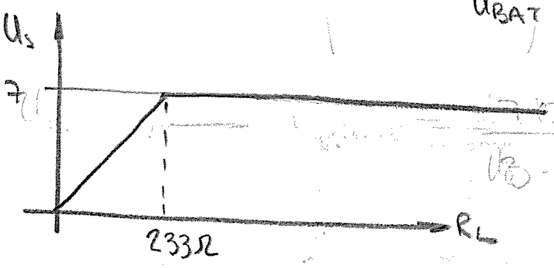
**NOTA!!** El zener ha conseguido que la tensión en carga no varíe y se mantenga en un valor de 7V. ⇒ Estabilizador de tensión

**c)** ¿Hasta qué valor de $R_L$ estabiliza la tensión de este circuito el zener?

Este valor corresponderá con aquel que haga que la intensidad en la rama del diodo circule en sentido contrario polarizando el diodo en directa.

El valor límite corresponderá al que haga $i_z = 0$

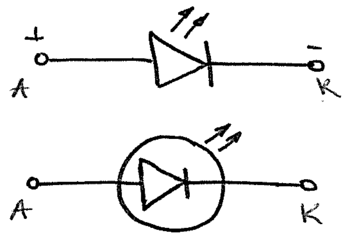
$$i_z = 0 \longrightarrow i_o = i_L : \quad \begin{array}{l} U_{BAT} = 10 \longrightarrow i_o = 30\ mA = 7/R_L \longrightarrow R_{L\,límite} = \dfrac{7}{0{,}03} = 233\ \Omega \\ U_{BAT} = 12 \longrightarrow i_o = 50\ mA = 7/R_L \longrightarrow R_{L\,límite} = \dfrac{7}{0{,}05} = 140\ \Omega \end{array}$$

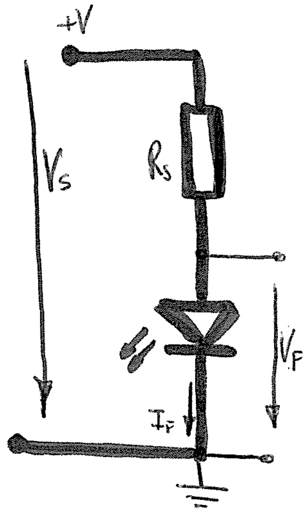


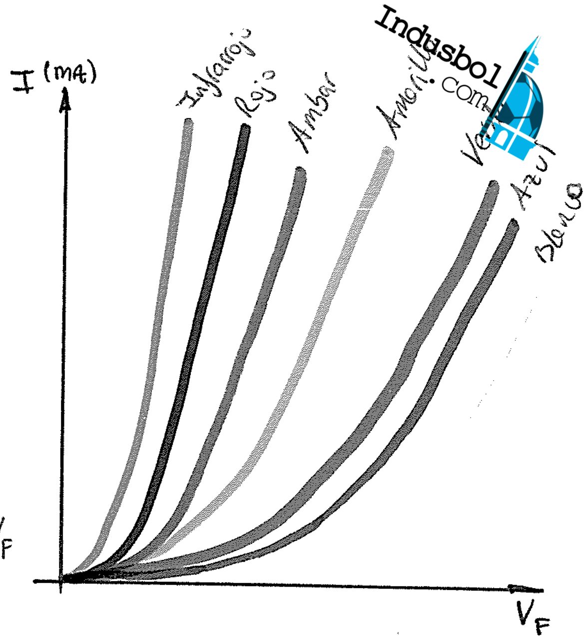

$$\boxed{R_{L\,límite} = 233\ \Omega}$$

## LED: Light Emiting Diode

Símbolos:



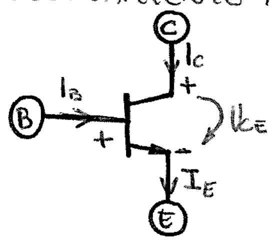



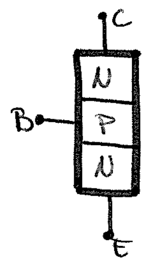

$$R_S = \frac{V_S - V_F}{I_F}$$

# 4 EL TRANSISTOR BIPOLAR

## SÍMBOLO FUNCIONAMIENTO Y CURVA CARACTERÍSTICA

Transistor NPN



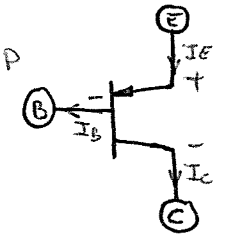

C = Colector

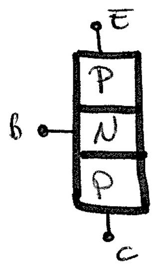
B = Base

E = Emisor

$$\boxed{I_E = I_B + I_C}$$

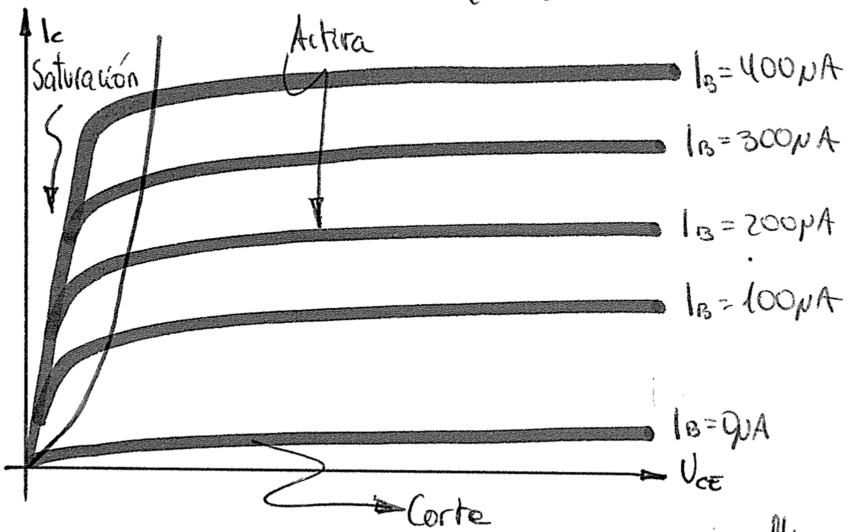
Transistor PNP




$$\boxed{I_C = I_E - I_B}$$

CURVA CARACTERÍSTICA REAL (NPN)

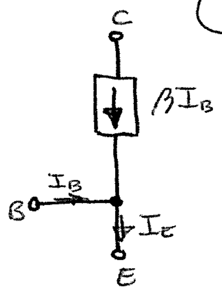



$$\boxed{I_C = \beta I_B}$$

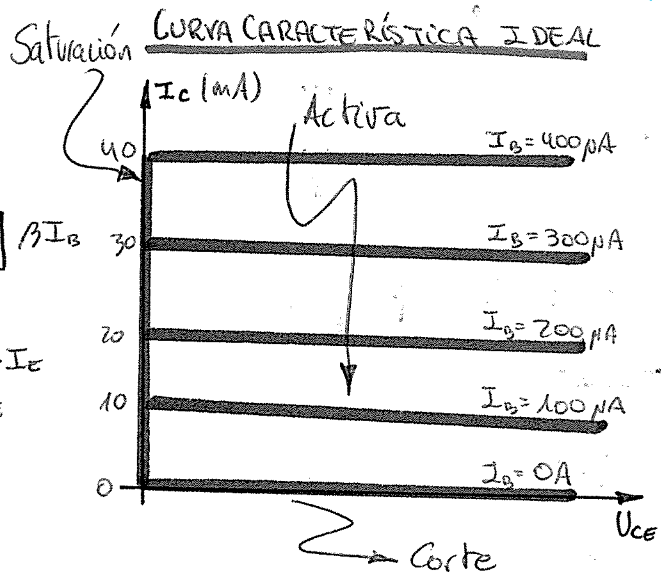
## Zonas de funcionamiento

- **Zona de corte**
  $I_B = 0;\ I_C = 0$


- **Lineal o activa**
  $I_C = \beta I_B$

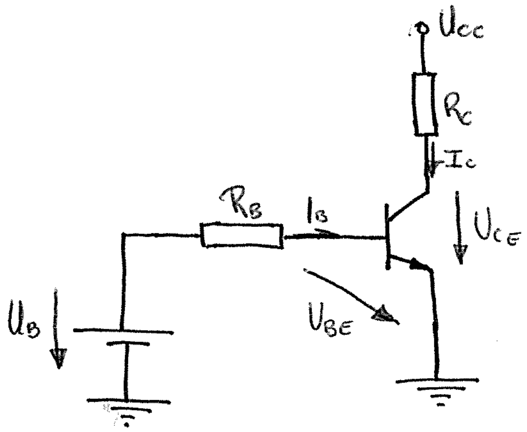



- **Saturación**
  $U_{CE} = \phi\ V$

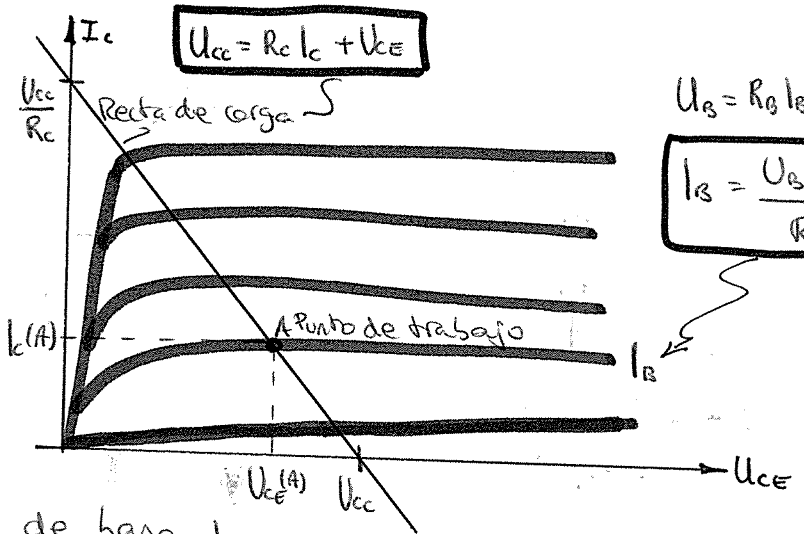

## POLARIZACIÓN Y RECTA DE CARGA

Polarizar un transistor consiste en proporcionarle alimentación, darle un punto de trabajo




Llamamos recta de carga al conjunto de puntos $(U_{CE}, I_C)$ donde el transistor puede trabajar y que viene determinado por el circuito externo.

$$\boxed{U_{CC} = R_C I_C + U_{CE}}$$




$U_B = R_B I_B + U_{BE}$

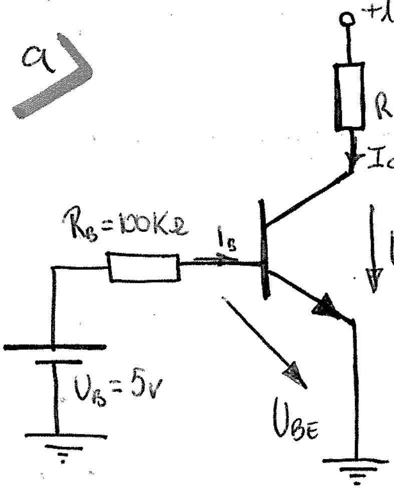
$$\boxed{I_B = \frac{U_B - U_{BE}}{R_B}}$$

$A(U_{CE}(A), I_C(A))$ = PUNTO DE TRABAJO

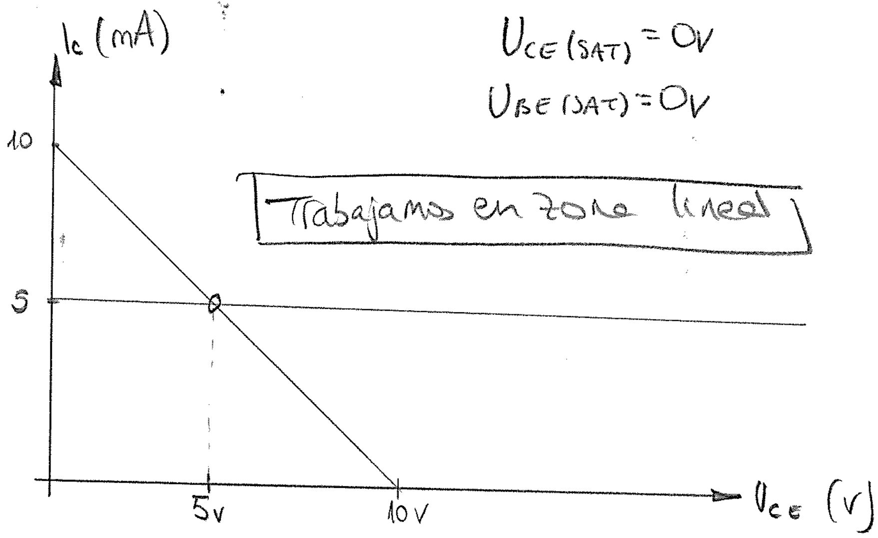
Viene dado por el corte de la recta de carga y la curva característica para la corriente de base $I_B$

Ejemplo: Calcular el punto de trabajo

a)




DATOS: Transistor ideal
Zona activa $\beta = 100$
$U_{BE} = 0V$
Zona saturación
$U_{CE(SAT)} = 0V$
$U_{BE(SAT)} = 0V$




$\boxed{\text{Trabajamos en zona lineal}}$

$$\left\{\begin{aligned} U_{cc} &= I_c R_c + U_{CE} \\ 10 &= 1000 I_c + U_{CE} \\ I_c &= (10 - U_{CE})\ mA \end{aligned}\right.$$

$\longrightarrow$ Recta de carga

Zona activa: $U_{BE} = 0$

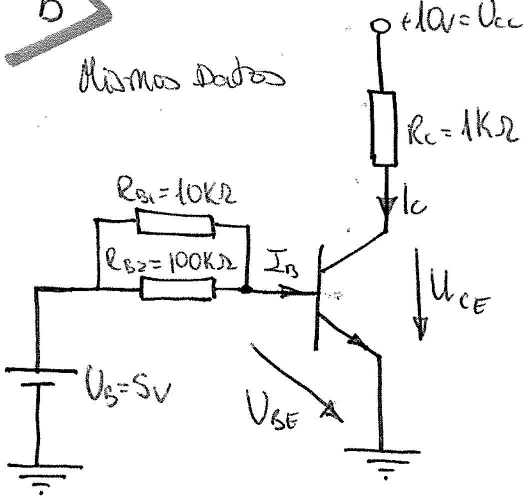
$$U_B = R_B I_B + U_{BE};\quad I_B = \frac{U_B - U_{BE}}{R_B} = \frac{5}{10^5}\ A = 50\ \mu A \longrightarrow I_c = \beta I_B = 5\ mA$$

Saturación $\rightarrow U_{CE} = 0 \longrightarrow \underline{I_c = 10\ mA}$;

$\boxed{\text{Punto de funcionamiento} \left\{ \begin{aligned} I_c &= 5\ mA \\ U_{CE} &= 10 - 5 = 5V \end{aligned}\right.}$

b)

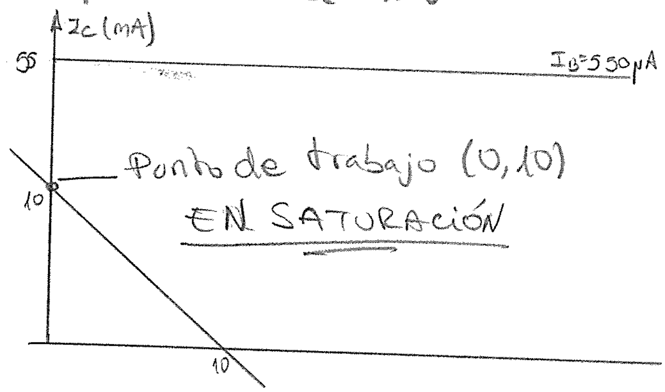
Mismos Datos




• Recta de carga: $U_{cc} = I_c R_c + U_{CE}$; $\boxed{I_c = 10 - U_{CE}\ (mA)}$

En saturación $\{ U_{CE(SAT)} = 0 \longrightarrow I_c = 10\ mA$

En corte $\{ I_c = 0 \longrightarrow U_{CE} = 10V$




Punto de trabajo (0,10)
EN SATURACIÓN

$$R_B = R_{B1} // R_{B2} = \frac{R_{B1} R_{B2}}{R_{B1} + R_{B2}} = \frac{10^3}{110} = 9{,}09\ K\Omega$$

$$U_B = I_B R_B + U_{BE};\quad I_B = \frac{U_B - U_{BE}}{R_B} = \frac{5}{9{,}09} = 550\ \mu A \longrightarrow I_c = 55\ mA$$

zona activa $U_{BE} = \phi$

$550\ \mu A > 100\ \mu A = I_B(sat)$

No puede estar en zona lineal

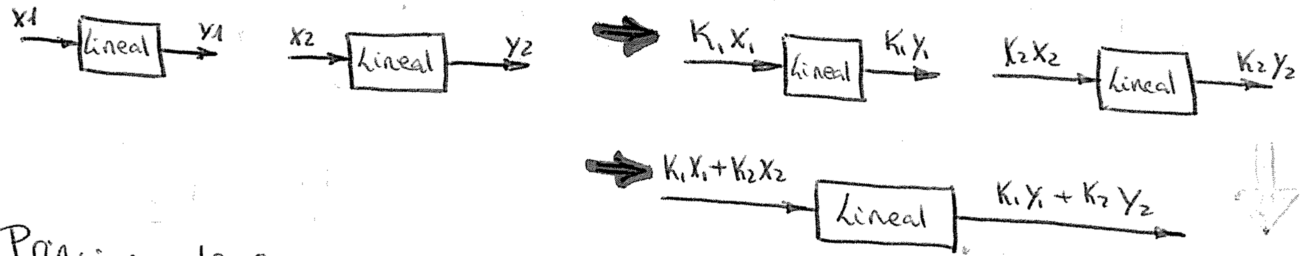
TEMA 2
Analógica

# AMPLIFICACIÓN

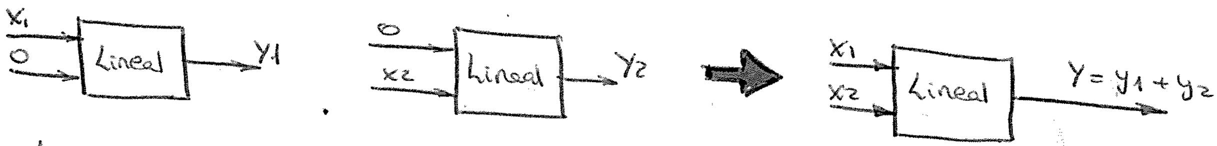
## 1 LINEALIDAD Y SUPERPOSICIÓN

En un sistema lineal se cumple que:

Si



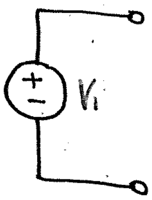

### Principio de superposición:



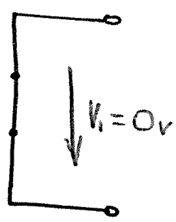

- Analizamos la respuesta del circuito ante cada entrada, anulando el resto
- Sumando todas las respuestas individuales se obtiene la "y" general

NOTA!! Fuentes. Como anularlas

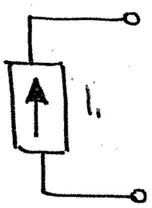
Fuente de tensión (pueden ser dependientes o independientes)



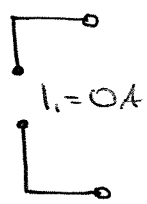

Para anular una fuente de tensión
$\rightarrow V_i=0 \rightarrow$ Cortocircuito




Fuente de intensidad (pueden ser dependientes o independientes)



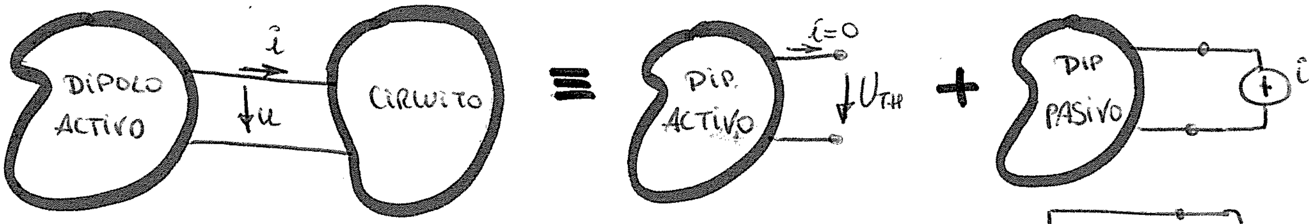

Para anular una fuente de intensidad
$\rightarrow I_i=0 \rightarrow$ Circuito abierto

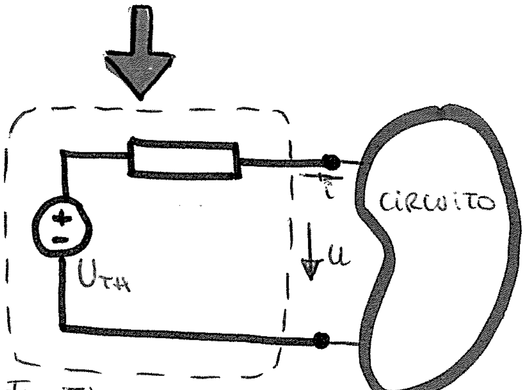


# 2 EQUIVALENTES THEVENIN Y NORTON. IMPEDANCIA DE ENTRADA Y SALIDA

## EQUIVALENTE THEVENIN

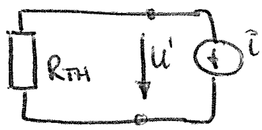
"Dado un dipolo activo es posible encontrar una fuente real de tensión equivalente al dipolo"




Eq. Thevenin

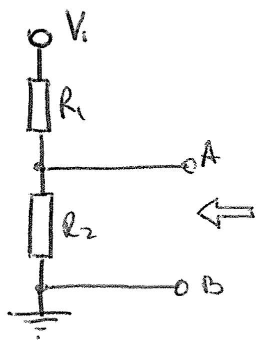
Para determinar $U_{TH}$ dejamos el dipolo conectado a circuito abierto.



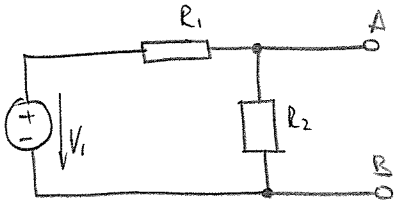

Hacemos el circuito pasivo anulando las fuentes independientes

$$R_{TH} = \frac{u'}{i}$$

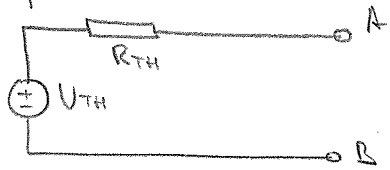
**Ejemplo:** Equivalente Thevenin del circuito dado



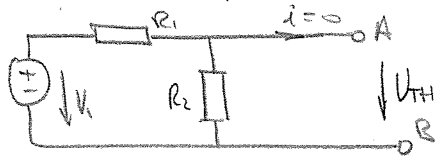

≈




Eq. Thévenin

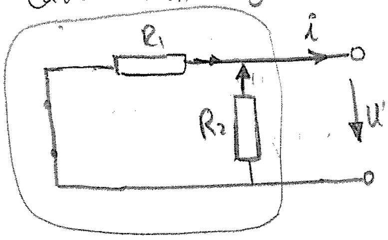



Cálculo $U_{TH}$ = Dejo a circuito abierto




$$\boxed{U_{TH} = \frac{R_2}{R_1+R_2} V_1}$$

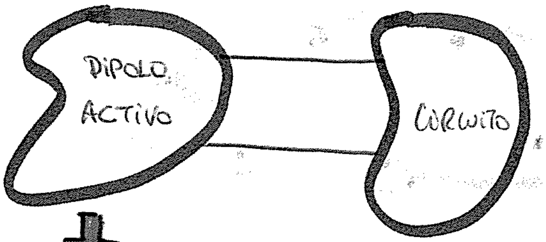
Cálculo $R_{TH}$: Hago circuito pasivo



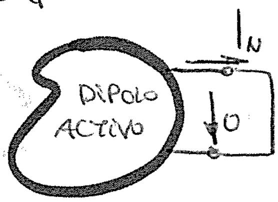

$$\boxed{R_{TH} = R_1 // R_2} = \frac{R_1 R_2}{R_1 + R_2}$$

## ● EQUIVALENTE NORTON

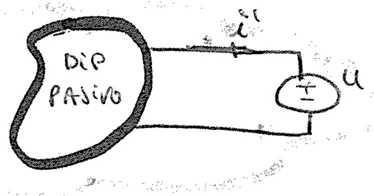
"Dado un dipolo activo es posible encontrar una fuente de intensidad equivalente al dipolo".




≡

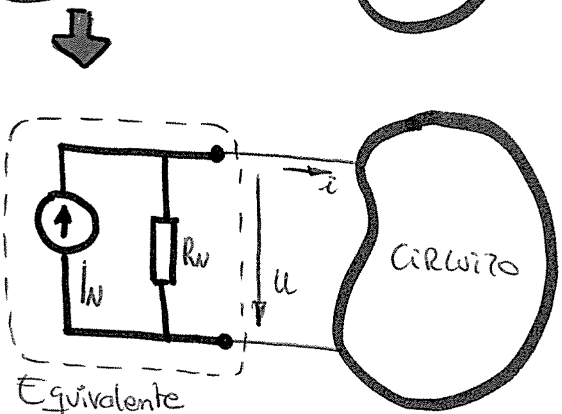



Cortocircuitamos el dipolo para determinar $I_N$



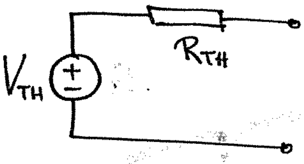

Hacemos el dipolo pasivo anulando las fuentes independientes

$$R_N = \frac{u}{i'}$$

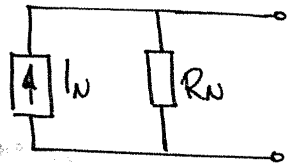



Equivalente Norton

## ● Equivalencia Thevenin / Norton



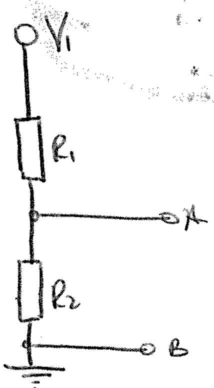

EQUIVALENTE THEVENIN

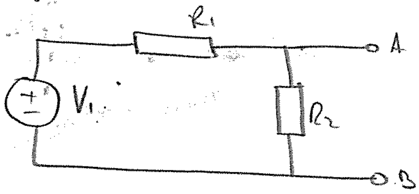



EQUIVALENTE NORTON

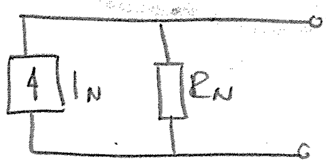
$$\boxed{\begin{array}{l} R_{TH} = R_N \\ V_{TH} = R_N I_N \end{array}}$$

**Ejemplo:** Calcular equivalente Norton del ejemplo anterior

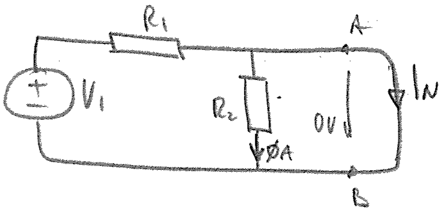



Eq Norton




Calculo $I_N$: Cortocircuito

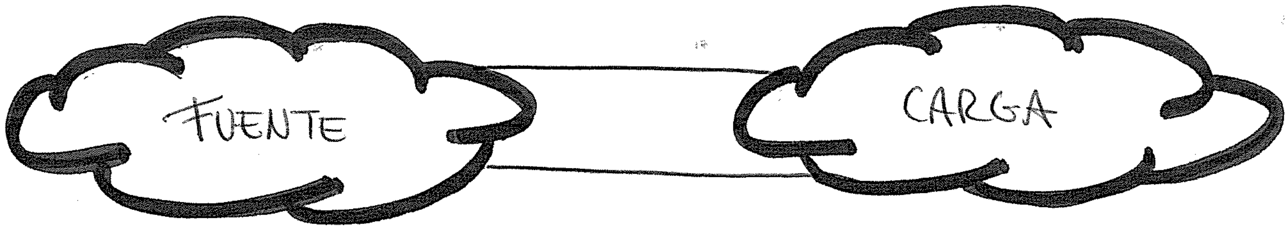



$$\boxed{I_N = \frac{V_1}{R_1}}$$

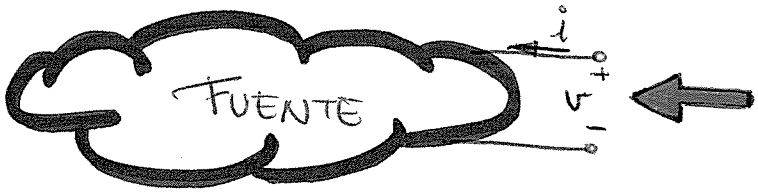
Calculo $R_N$: Hago circuito pasivo → $\boxed{R_N = R_{TH} = R_1 // R_2 = \frac{R_1 R_2}{R_1 + R_2}}$

Se comprueba que $U_{TH} = \frac{R_2}{R_1 + R_2} V_1 = I_N R_N = \frac{V_1}{\cancel{R_1}} \frac{R_2 \cancel{R_1}}{R_2 + R_1}$

## IMPEDANCIA DE ENTRADA Y DE SALIDA

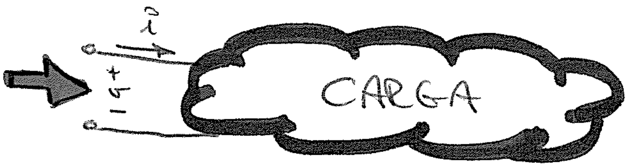



Impedancia de salida

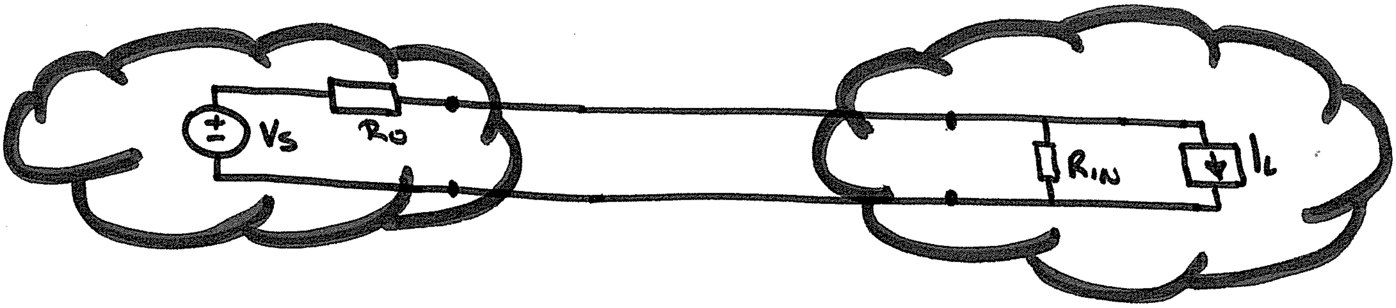



Impedancia de la fuente cuando todos las fuentes independientes son nulas
Es igual a la $R_{TH}$ o $R_N$ vista desde los terminales

Impedancia entrada

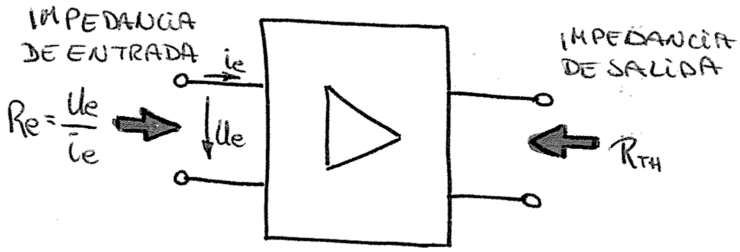



Impedancia de la carga cuando todas las fuentes independientes son cero.
Es igual a la $R_{TH}$ o $R_N$ vista desde los terminales.



# 3 AMPLIFICACIÓN DE SEÑALES: TIPOS DE AMPLIFICADORES

● DEFINICIÓN



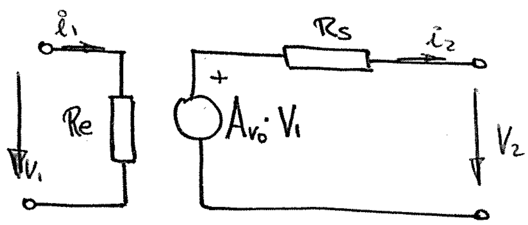

Un amplificador es un dispositivo que magnifica una señal

La potencia de salida puede ser mayor que la de la entrada, lo que implica la existencia de una fuente de energía adicional. Por tanto, un transformador NO es un amplificador

Se define <u>ganancia</u> como la relación entre la magnitud de salida y la de la entrada.

● CIRCUITO EQUIVALENTE

<u>Amplificador de tensión</u>



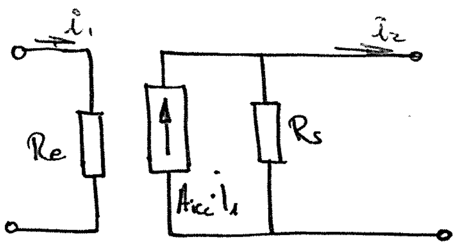

$A_{V_0}$ ≡ Ganancia de tensión en vacío (circuito abierto)

$R_e$ ≡ Resistencia de entrada

$R_s$ ≡ Resistencia de salida

Deseable $R_e \uparrow$, $R_s \downarrow$

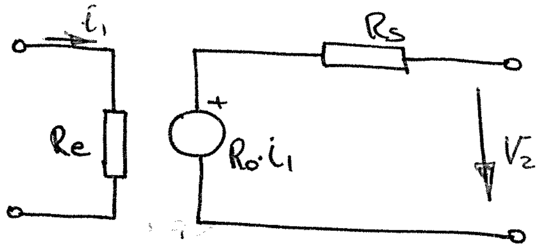
<u>Amplificador de corriente</u>




$A_{icc}$ ≡ Ganancia de intensidad en vacío (circuito abierto)

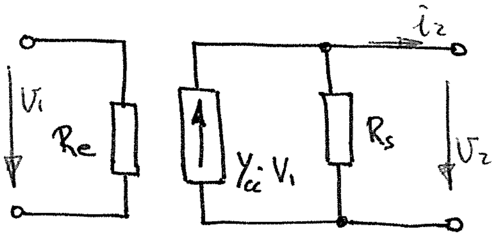
Deseable $R_e \downarrow$, $R_s \uparrow$

## Amplificador de transresistencia



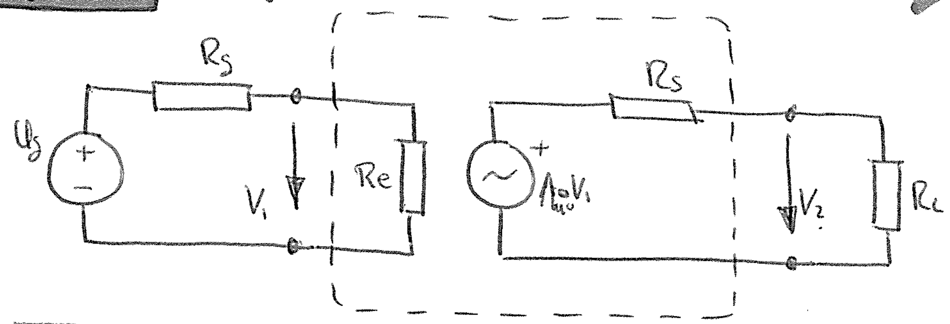

Deseable Re ↓, Rs ↓

## Amplificador de transconductancia




Deseable Re ↑, Rs ↑

## Ejemplo: amplificador de tensión




a) Calcular $V_1$, $V_2$

Datos

$A_{uo} = 10$
$R_e = 1K\Omega$
$R_s = 10\,\Omega$
$U_g = 2\,\text{sen}(wt)\ V$
$R_g = 100\,\Omega$
$R_L = 50\,\Omega$

$$\boxed{V_1 = \frac{R_e}{R_e+R_g} U_g = \frac{1000}{1100} 2\,\text{sen}(wt) = 1{,}82\,\text{sen}(wt)\ V}$$

$$\boxed{V_2 = A_{uo} V_1 \frac{R_L}{R_s+R_L} = 18{,}2\,\text{sen}(wt) \frac{50}{60} = 15{,}15\,\text{sen}(wt)\ V}$$

b) Ganancia en carga

$$\boxed{A_{vL}} = \frac{V_2}{U_g} = \frac{A_{uo} V_1 \frac{R_L}{R_s+R_L}}{U_g} = \frac{A_{uo} \frac{R_e}{R_e+R_g} \cancel{U_g} \frac{R_L}{R_s+R_L}}{\cancel{U_g}} = 10 \cdot \frac{10}{11} \cdot \frac{5}{6} = \boxed{7{,}57}$$

c) Potencia de entrada, salida y pérdidas

$$P_S = \frac{V_2^2}{R_L} = \left(\frac{15{,}15}{\sqrt{2}}\right)^2 \frac{1}{50} = 2{,}3\ W$$

$$P_e = \frac{V_1^2}{R_e} = \left(\frac{1{,}82}{\sqrt{2}}\right)^2 \frac{1}{1000} = 1{,}6\ mW$$

En $R_s$

$$P_{perd} = I_2^2 \cdot R_s = \left(\frac{A_{vo} V_1}{R_s + R_L}\right)^2 \cdot R_s = \left(\frac{10 \frac{1,82}{\sqrt{2}}}{10+50}\right)^2 \cdot 10 = 0,46\ W$$

Comprobamos que $P_1 < P_2 + P_{perd}$

## POTENCIA TRANSFERIDA

Vamos a analizar la expresión de la potencia a la salida del circuito anterior

$$P_2 = \frac{V_2^2}{R_L} ; \quad V_2 = \frac{A_{vo} V_1}{R_s + R_L} R_L \longrightarrow P_2 = A_{vo}^2 V_1^2 \frac{R_L}{(R_s + R_L)^2}$$

$$\frac{dP_2}{dR_L} = 0 \longrightarrow R_L = R_s \left(\begin{array}{c}\text{Máxima potencia} \\ \text{transferida}\end{array}\right)$$

PERO $P_{2max} \not\Rightarrow \eta_{max}$

- Adaptando impedancias para máxima potencia $(R_L = R_s)$

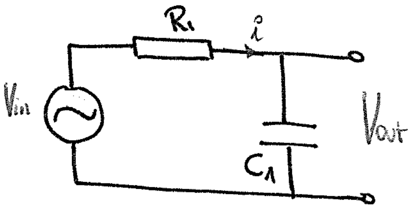
$$\eta = \frac{P_2}{P_2 + P_{Rs}} = 50\%$$

- Transferencia eficiente, sin adaptar impedancias

$R_L > R_s$ Me interesa $R_s$ lo más pequeña posible $\longrightarrow \eta > 50\%$

# 4 Respuesta en frecuencia: Diagramas de Bode, dB

## Conceptos básicos




$\omega \equiv$ Pulsación de excitación; $\omega = 2\pi f$

$f \equiv$ Frecuencia de excitación;

$T \equiv$ Periodo de excitación; $T = \dfrac{1}{f}$

### Función de Transferencia

Variable s (de Laplace) $s = j\omega \Rightarrow$

$$\boxed{G(s) = \frac{1}{1+T_p s} \qquad G(s) = \frac{1}{1+\frac{s}{\omega_p}}}$$

$T_p \equiv$ cte de tiempo del polo

$\omega_p \equiv$ Pulsación del polo

$T_p = \dfrac{1}{\omega_p}$

En el ejemplo

$$G(s) = \frac{V_{out}}{V_{in}} = \frac{1}{1+R_1C_1s}$$

$V_{out} = \dfrac{1}{C_1 s}\, i \longrightarrow i = V_{out} \cdot C_1 s$

$V_{in} = V_{out} + R_1 i \longrightarrow V_{in} = V_{out}(1+R_1C_1s)$

$$\boxed{T_p = RC}$$

Análisis:

## Respuesta en frecuencia

Vamos a ir variando la frecuencia de excitación $\omega$ para:

$RC = 1ms$ (cte. tiempo)

$$G(s) = \frac{1}{1+RCs} = \frac{1}{1+RC\omega j}$$

$\omega_1 = \dfrac{0{,}1}{RC} = 100\ rad/s$

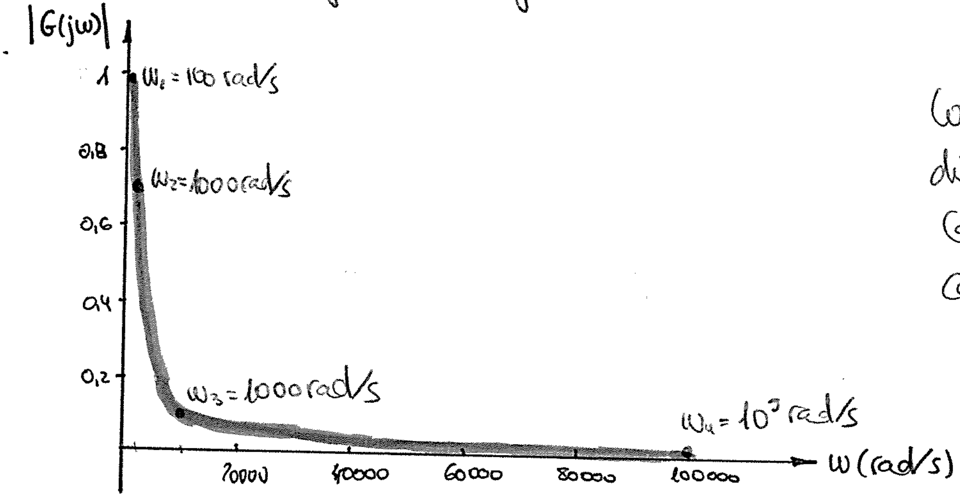
$\omega_2 = \dfrac{1}{RC} = 1000\ rad/s$

$\omega_3 = \dfrac{10}{RC} = 10000\ rad/s$

$\omega_4 = \dfrac{100}{RC} = 10^5\ rad/s$

$$\begin{cases} G(j\omega_1) = \dfrac{1}{1+RC\frac{0{,}1}{RC}j} = \dfrac{1}{1+0{,}1j} = 0{,}995\angle{-5{,}7^\circ} \\ G(j\omega_2) = \dfrac{1}{1+j} = 0{,}707\angle{-45^\circ} \\ G(j\omega_3) = \dfrac{1}{1+10j} = 0{,}0995\angle{-84^\circ} \\ G(j\omega_4) = \dfrac{1}{1+100j} = 0{,}00999\angle{-89^\circ} \end{cases}$$

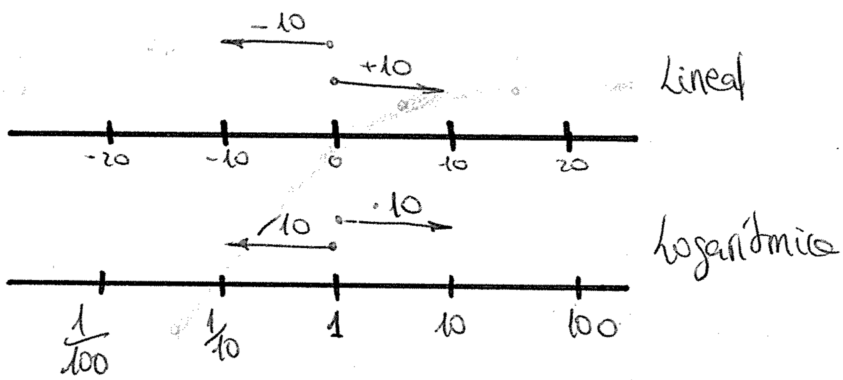
Por tanto, la respuesta en frecuencia en escala lineal será:




Como vemos es muy difícil visualizar la respuesta en escala lineal

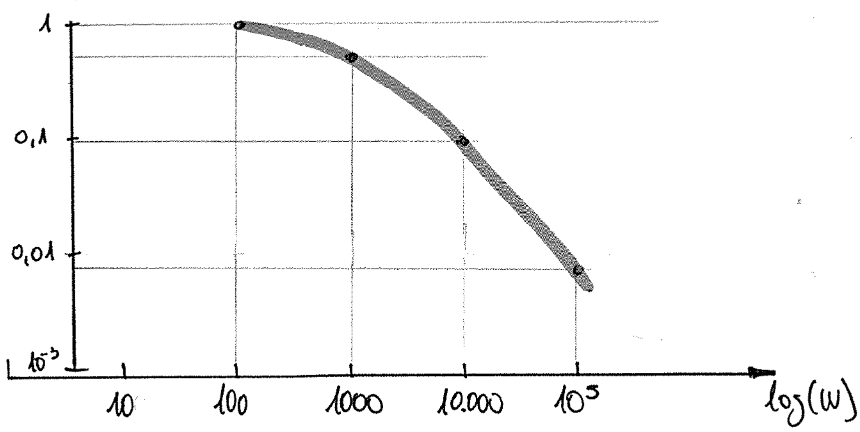
## ● ESCALA LOGARÍTMICA

Son escalas útiles cuando las variables cambian varios órdenes de magnitud.




- Si queremos desplazarnos un factor de 100: $\log(100) = 2$ unidades
- Si queremos desplazarnos un factor de 0,1: $\log(0,1) = -1$ unidades

La representación anterior |G(jω)| en escala logarítmica queda:



## Ganancias en dB

Las ganancias podemos medirlas en dB

$$A_v(dB) = 20 \log_{10} \frac{v_2}{v_1} \;,\; A_i(dB) = 20 \log_{10} \frac{i_2}{i_1} \;,\; A_p(dB) = 10 \log_{10} \frac{P_2}{P_1}$$

Decada: Es un cambio por un factor de 10

N décadas es un cambio por un factor de $10^N$

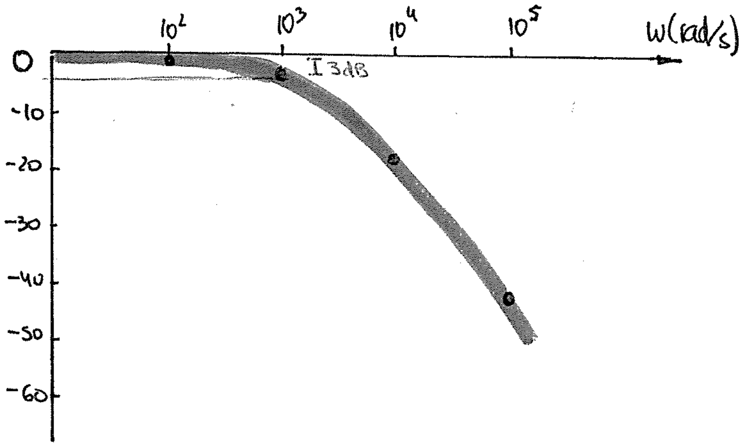
Ejemplo: Calcular décadas entre:

- 1 kHz está una década por encima de 100 Hz
- 100 kHz está 3 décadas por encima de 100 Hz

$$N = \log\left(\frac{f_2}{f_1}\right) \qquad \frac{f_2}{f_1} = 10^N$$

La representación anterior con las ganancias en dB queda

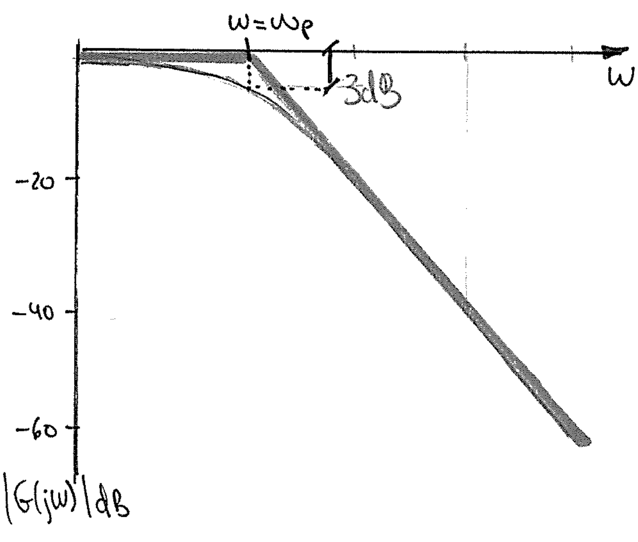
| $\omega$ | $\lvert G(j\omega)\rvert$ | $\lvert G(j\omega)\rvert$ en dB $20\log(\lvert G(j\omega)\rvert)$ |
|---|---|---|
| 100 | 0,995 | −0,043 dB |
| 1000 | 0,7 | −3 dB |
| 10000 | 0,1 | −20 dB |
| 100000 | 0,01 | −40 dB |




## Diagramas de Bode

Consisten en aproximar por rectas las gráficas $(\omega, |G(j\omega)|\,dB)$ así como $(\omega, \angle G(j\omega))$

<u>Diagrama de Bode asintótico polo:</u>



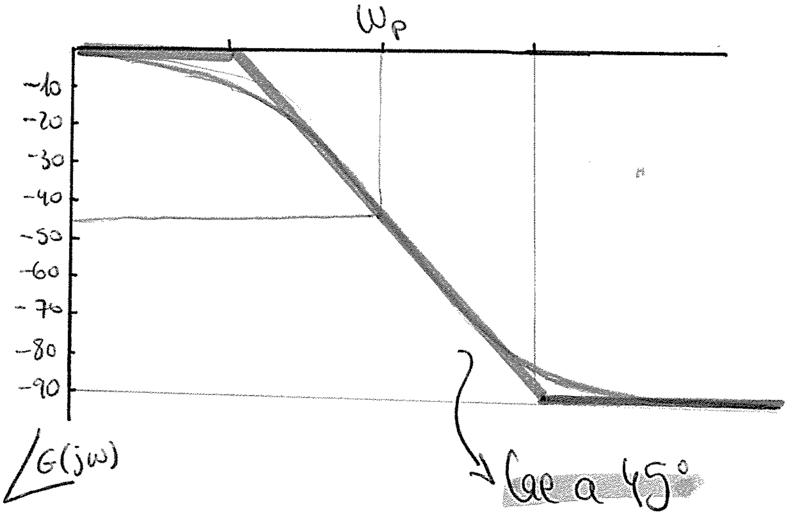

$$G(s) = \frac{1}{1+\frac{s}{\omega_p}} \;;\; |G(j\omega)| = \left|\frac{1}{1+j\frac{\omega}{\omega_p}}\right| = \frac{1}{\sqrt{1+\left(\frac{\omega}{\omega_p}\right)^2}}$$

$$\begin{cases} \omega \ll \omega_p \rightarrow |G(j\omega)| \simeq 1 \xrightarrow{dB} 20\log 1 = 0\,dB \\ \omega \gg \omega_p \rightarrow |G(j\omega)| \simeq \frac{\omega_p}{\omega} \xrightarrow{dB} 20\log\frac{\omega_p}{\omega} = \end{cases}$$

$\boxed{\omega = \omega_p}$ ; $|G(j\omega)| = \frac{1}{\sqrt{2}} \Rightarrow \boxed{3\,dB}$ $\qquad = 20\log(\omega_p) - 20\log(\omega)$

- Empieza, para $\omega$ pequeña → 0dB
- Cae a 20dB por década

También podemos representar la fase:




- 1º hallemos $\angle G(\omega=\omega_p)$

$\angle G(\omega=\omega_p) = -45°$

- Empieza a caer una década antes $\angle G(\omega=\frac{\omega_p}{10})$

$\angle G(\omega=\frac{\omega_p}{10}) = -6° \approx 0$

- Deja de caer una década después $\angle G(\omega=10\omega_p)$

$\angle G(\omega=10\omega_p) = -84° \approx -90°$

## Diagrama Bode asintótico (cero):

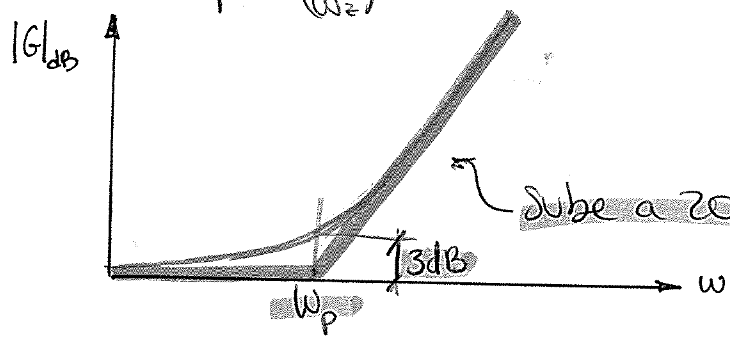
$$G(s) = 1 + \frac{s}{\omega_z} = 1 + \frac{\omega}{\omega_z} j$$

$$|G(s)| = \sqrt{1+\left(\frac{\omega}{\omega_z}\right)^2}$$

$\omega << \omega_z \longrightarrow |G(s)| = 1 \xrightarrow{dB} 0\,dB$

$\omega >>> \omega_z \longrightarrow |G(s)| = \frac{\omega}{\omega_z} \xrightarrow{dB}$

$\xrightarrow{dB} -20\log(\omega_z) + 20\log\omega$




$|G(\omega=\omega_z)| = \sqrt{2} \xrightarrow{dB} 20\log(\sqrt{2}) = 3dB$

Y para la fase:

- 1º hallemos $\angle G(\omega=\omega_z)$

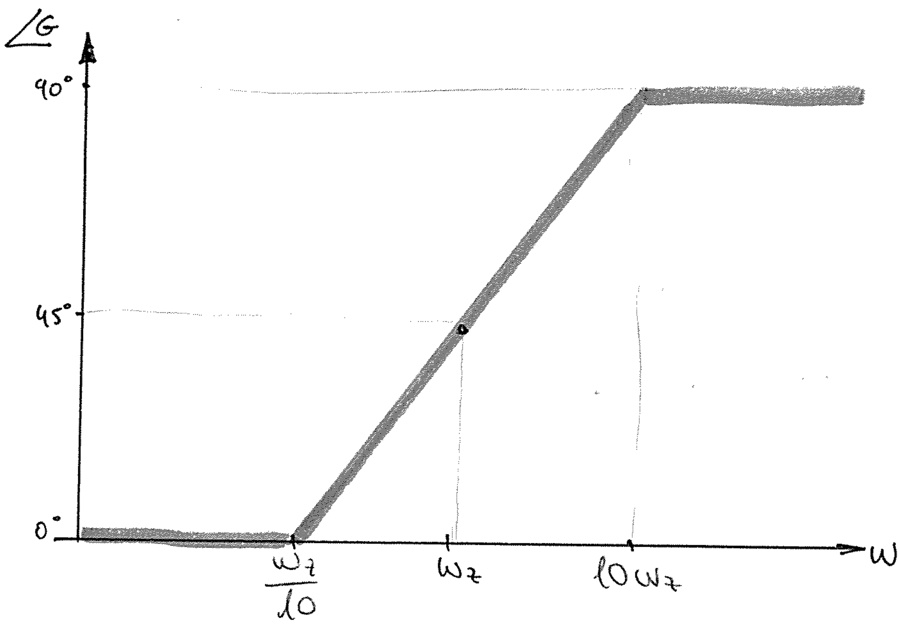
$\angle G(\omega=\omega_z) = 45°$

- Empezamos a subir una década antes

$\angle G(\omega=\frac{\omega_z}{10}) \approx 0°$

- Acabamos de subir una década después

$\angle G(\omega=10\omega_z) \approx 90°$



## Diagrama asintótico polos múltiples

Suponemos un polo múltiple de orden $n$.

El diagrama será idéntico al del polo simple excepto:

$$G(s) = \frac{1}{\left(1+\frac{s}{\omega_p}\right)^n} \qquad |G(\omega=\omega_p)| = \left(\sqrt{1+1^2}\right)^{-n} = \frac{1}{(\sqrt{2})^n} \xrightarrow{dB} -20\cdot n\cdot \log(\sqrt{2})$$

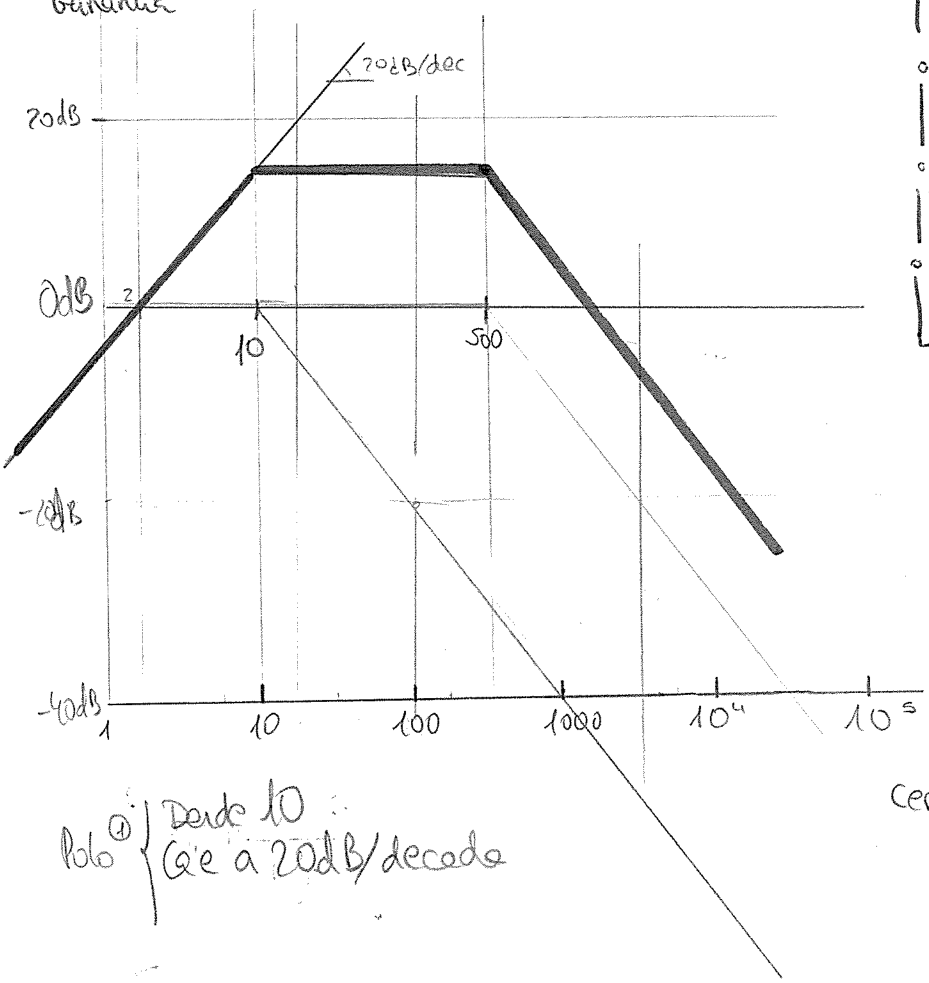
Punto en el que empieza a caer $\boxed{-3\cdot n\, dB}$

La pendiente serán $-n\cdot 20\, dB/década$

**Ejemplo**: Diagrama de Bode, fase y ganancia de:

$$G(s) = \frac{s/2}{\left(1+\frac{s}{10}\right)\left(1+\frac{s}{500}\right)}$$

Ganancia




CEROS Y POLOS EN EL ORIGEN

- **Cero**: $G(s) = s^n$
  - Fase: $90°\cdot n$
  - Crece a $20\cdot n\, dB/década$
- **Polo** $G(s) = \frac{1}{s^n}$
  - Fase: $-90°\cdot n$
  - Cae a $-20\cdot n\, dB/década$

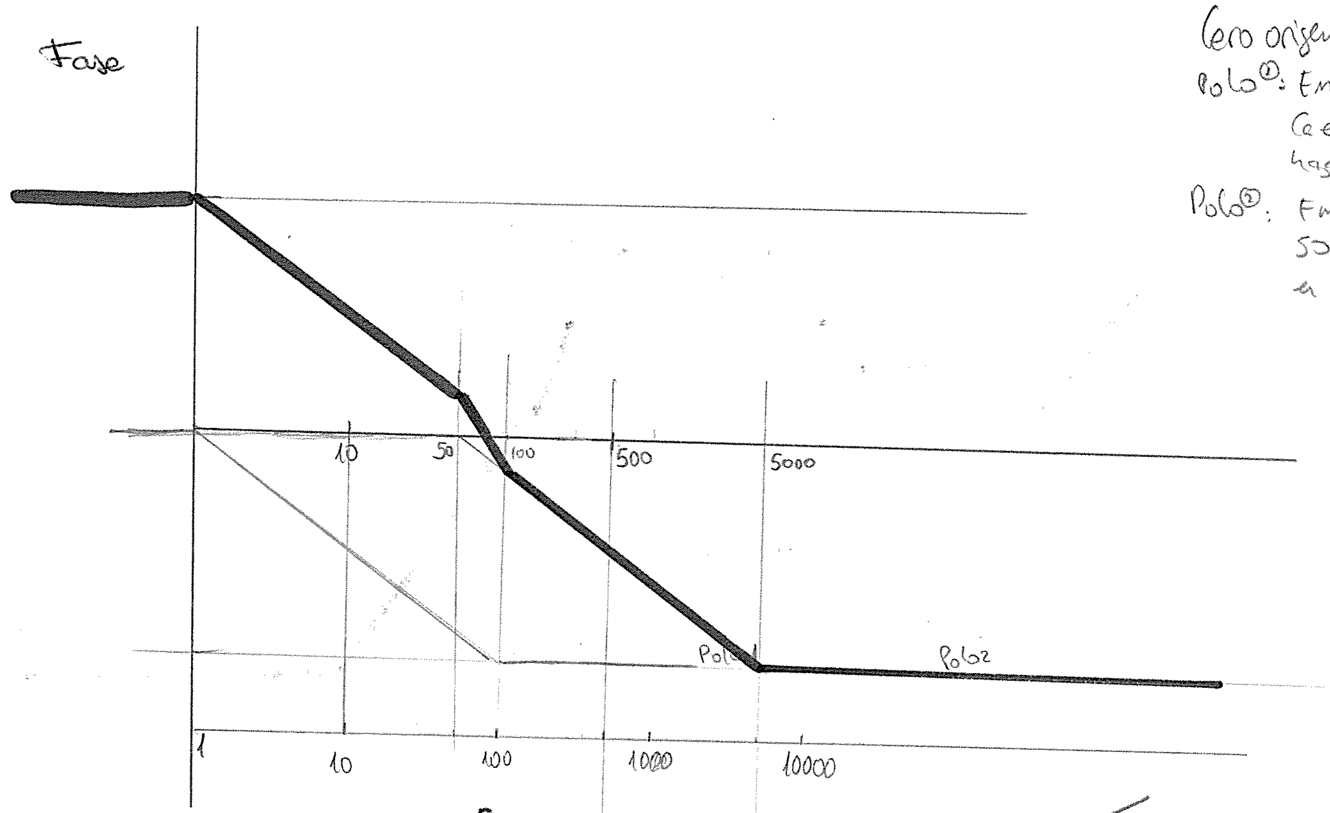
- Cero en el origen
- Polo① $\omega_p^{①} = 10$
- Polo② $\omega_p^{②} = 500$

Cero
$$\begin{cases} G_?(s) = \frac{s}{2} = \frac{\omega}{2} j \\ |G(s)| = \frac{1}{2}\omega, \xrightarrow{\omega=2} |G(s)| = 1 \longrightarrow \\ \xrightarrow{dB} 20\log(1) = 0 \end{cases}$$

Polo① $\begin{cases} \text{Desde } 10: \\ \text{Cae a } 20dB/década \end{cases}$

Cero origen: 90°
Polo ①: Empieza en
Cae a 45°/dec
hasta 100
Polo ②: Empieza en
50 y acaba
a 5000

Fase

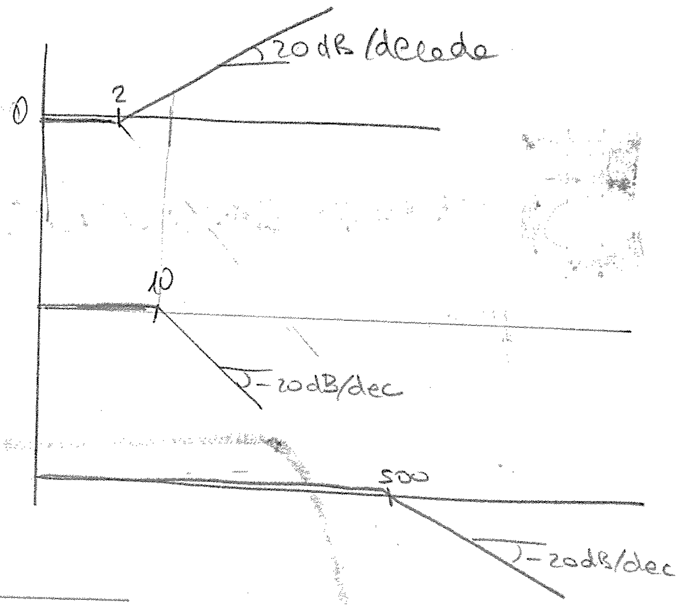



**Ejemplo:** $G(s) = \dfrac{1+\frac{s}{2}}{\left(1+\frac{s}{10}\right)\left(1+\frac{s}{500}\right)}$

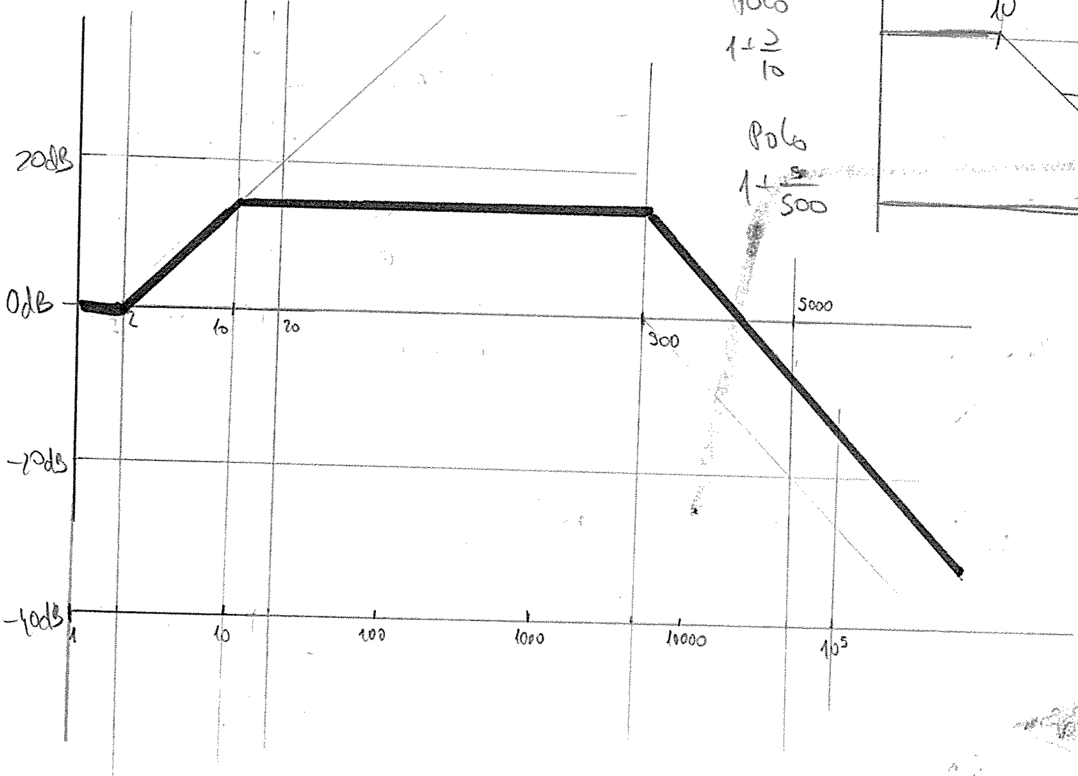
Cero $1+\frac{s}{2}$ $W_z=2$

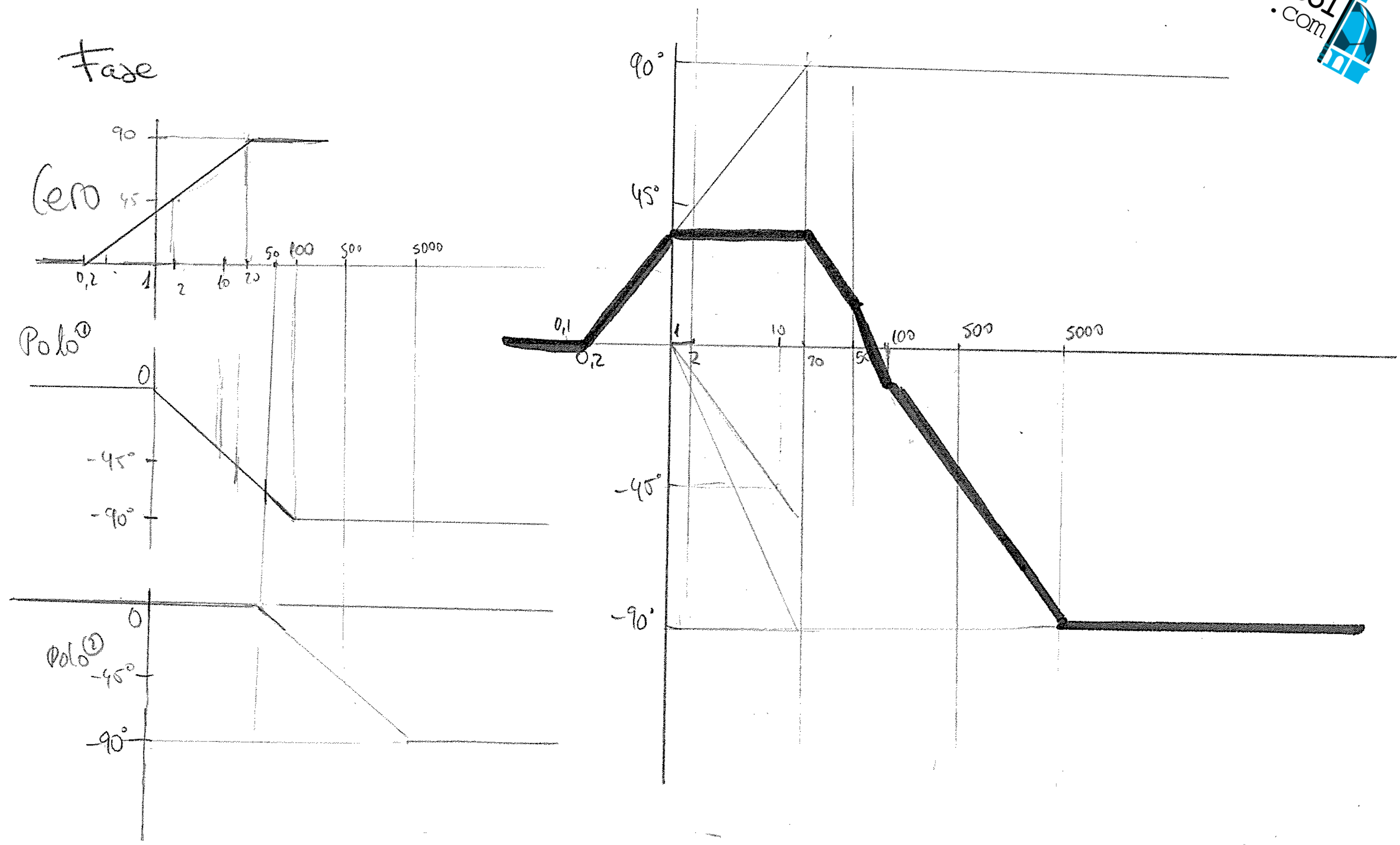
Polo $1+\frac{s}{10}$

Polo $1+\frac{s}{500}$

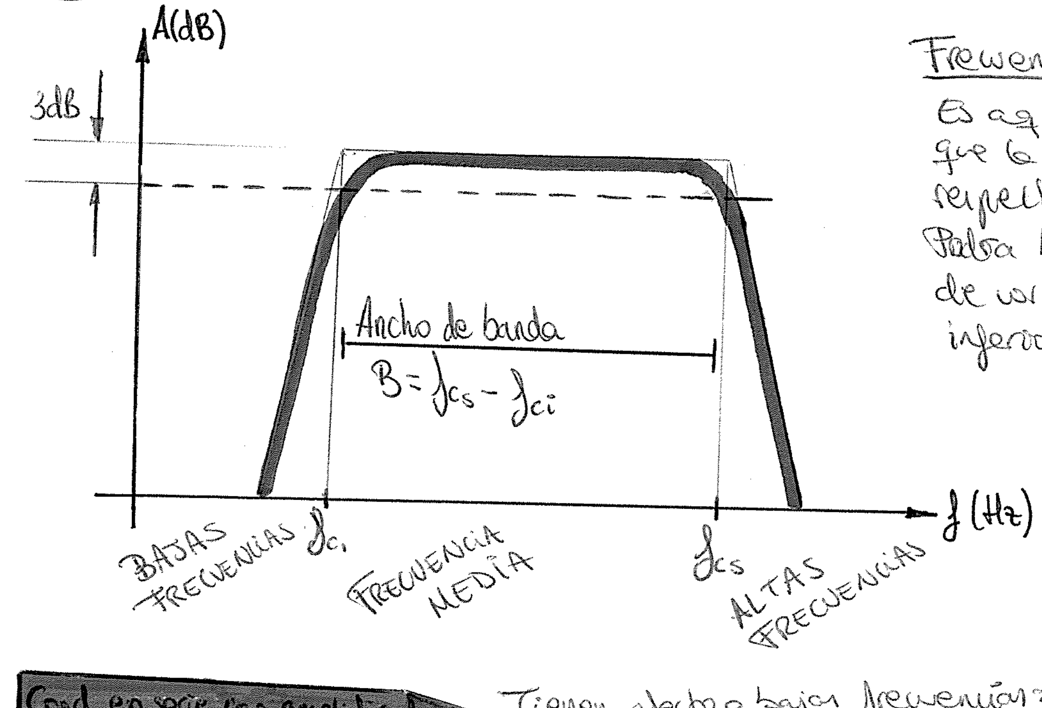



Ganancia






# 5 RESPUESTA EN FRECUENCIA DE AMPLIFICADORES



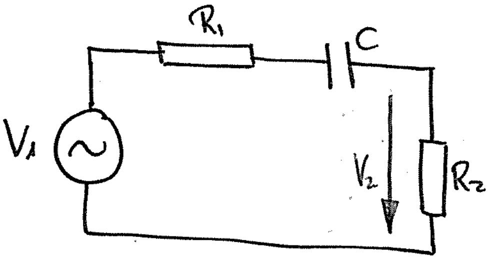

## Frecuencia de corte

Es aquella frecuencia a la que la ganancia cae 3dB respecto de la nominal. Suelen haber frecuencias de corte superiores e inferiores.

**Cond. en serie con amplificador** → Tienen efecto a bajas frecuencias: Meten $f_{ci}$. A altas frecuencias → cortocircuitos.

**Cond. en paralelo con amplific.** → Tienen efecto a altas frecuencias: Meten $f_{cs}$. A bajas frecuencias → circuitos abiertos.

## Efecto condensadores en serie




$$A(s)=\frac{V_2}{V_1}=\frac{R_2}{R_1+R_2+\frac{1}{Cs}}=\frac{R_2Cs}{(R_1+R_2)Cs+1}$$

$$A(s)=\frac{R_2}{R_1+R_2}\;\frac{C(R_1+R_2)s}{1+C(R_1+R_2)s}$$

A frecuencias medias/altas $\longrightarrow$ $A_m=\frac{R_2}{R_1+R_2}$ (cortocircuito en C)

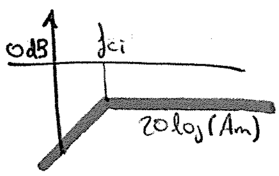
Para que $|A(\omega_L)|=\frac{A_m}{\sqrt{2}}$, esto es, caiga 3dB respecto de la frecuencia media, se tiene que cumplir que:

$$\left|\cancel{\frac{R_2}{R_1+R_2}}\;\frac{C(R_1+R_2)s}{1+C(R_1+R_2)s}\right|=\frac{1}{\sqrt{2}}\cancel{\frac{R_2}{R_1+R_2}}\qquad\left|\frac{1}{1+\frac{1}{(R_1+R_2)Cs}}\right|=\frac{1}{\sqrt{2}}$$

$$\frac{1}{\sqrt{1+\left(\frac{1}{(R_1+R_2)C\omega_L}\right)^2}}=\frac{1}{\sqrt{2}};\quad\boxed{\frac{1}{(R_1+R_2)C\omega_L}=1}$$

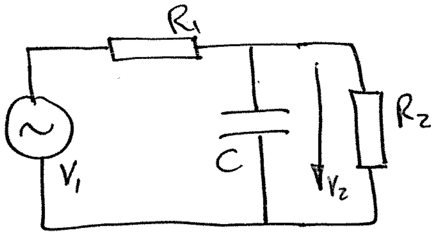
Portanto, $F_{ci}$ debido a C

$$\boxed{f_{ci}=\frac{1}{2\pi C(R_1+R_2)}}$$




## Efecto condensadores en paralelo

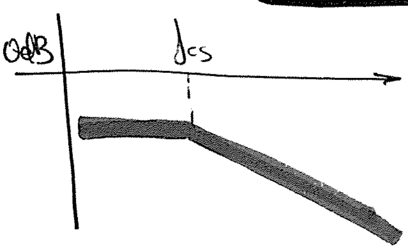
Análisis similar al anterior en circuito



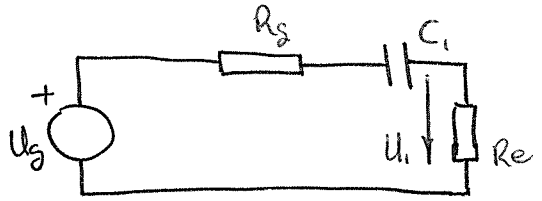

$F_{cs}$ debido a C

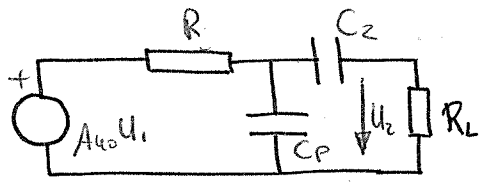
$$\boxed{f_{cs}=\frac{1}{2\pi C_p\frac{R_1R_2}{R_1+R_2}}}$$

A frec. bajas son circuitos abiertos

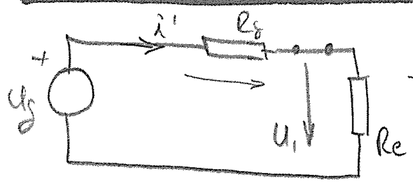


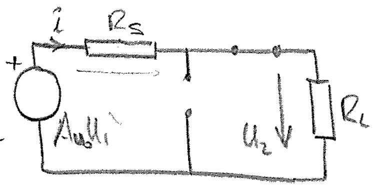
## APLICACIÓN AL AMPLIFICADOR




### Ganancia a frecuencias medias



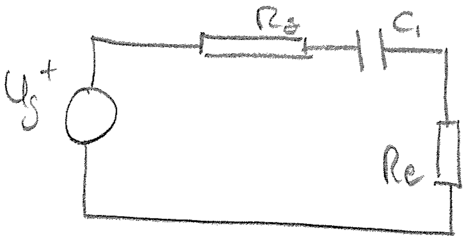



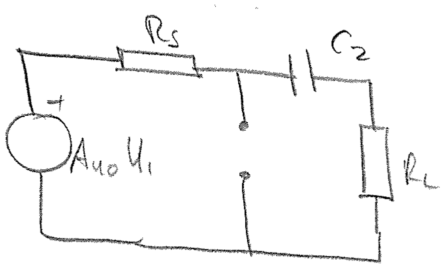

$$A_{u2} = \frac{U_2}{U_g} = A_{u0} \frac{R_e}{R_g + R_e} \cdot \frac{R_L}{R_s + R_L}$$

$$U_2 = \frac{R_L}{R_s + R_L} A_{u0} U_1 = \frac{R_e}{R_g + R_e} \cdot \frac{R_L}{R_s + R_L} \cdot U_g A_{u0}$$

$$U_1 = \frac{R_e}{R_g + R_e} U_g$$

### A baja frecuencia → C paralelo ≡ Circ. abierto

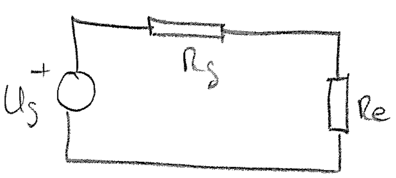


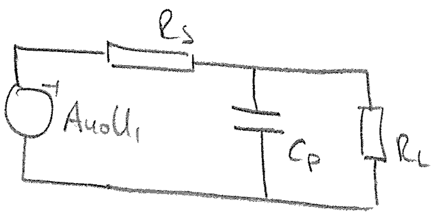




$$f_{ci①} = \frac{1}{2\pi C_1 (R_g + R_e)} \; ; \; f_{ci②} = \frac{1}{2\pi C_2 (R_s + R_L)}$$

Condensadores de desacoplo

### A alta frecuencia → C serie ≡ Cortocircuito




$$f_{cs} = \frac{1}{2\pi C_p \frac{R_s R_L}{R_s + R_L}}$$

Capacidades parásitas

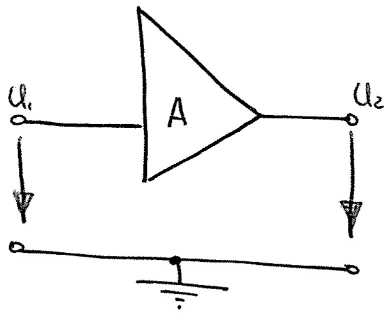
$$f_{csr} = f_{cs} \sqrt{2^{1/n} - 1}$$

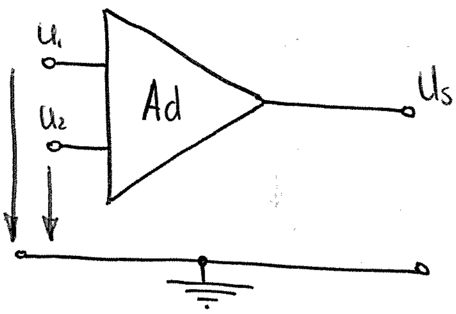
$$f_{cir} = f_{ci} \frac{1}{\sqrt{2^{1/n} - 1}}$$

# A. OPERACIONAL AOs

## 1 AMPLIFICADORES DIFERENCIALES

Los amplificadores, analizados en la lección anterior, tienen como entrada una sola tensión de referencia (masa)




Los AMPLIFICADORES DIFERENCIALES tienen dos entradas y dan una salida proporcional a la diferencia de las tensiones aplicadas a la entrada.

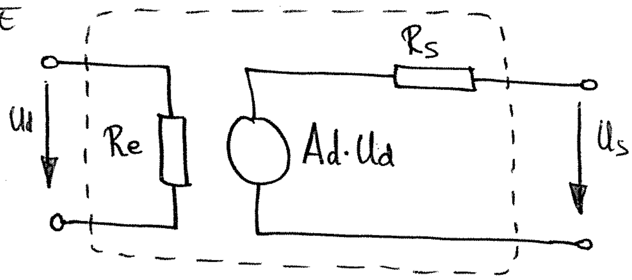
- TENSIÓN EN MODO DIFERENCIAL $\boxed{U_d = U_1 - U_2}$
- TENSIÓN EN MODO COMÚN $\boxed{U_{mc} = \frac{U_1 + U_2}{2}}$
- GANANCIAS $\begin{cases} \text{Modo común } A_{mc} \\ \text{Diferencial } A_d \end{cases}$ $A_{mc} << A_d$

AOs REALES $\boxed{U_s = A_d U_d + A_{mc} U_{mc}}$

AOs IDEALES $\boxed{U_s = A_d U_d}$

Me interesa que $A_{mc}$ sea lo más pequeña posible. Alimentando simétricamente nuestro AO conseguimos $U_{mc} = 0$ $\Rightarrow$ Razón de rechazo en modo común $RRMC = \frac{|A_d|}{|A_{mc}|}$

- CIRCUITO EQUIVALENTE

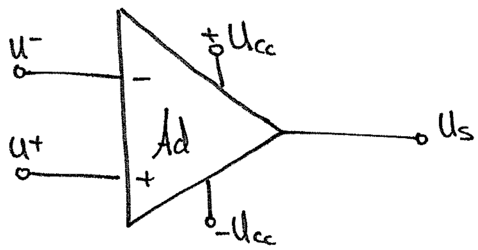


# 2 AMPLIFICADOR OPERACIONAL. EL AO IDEAL

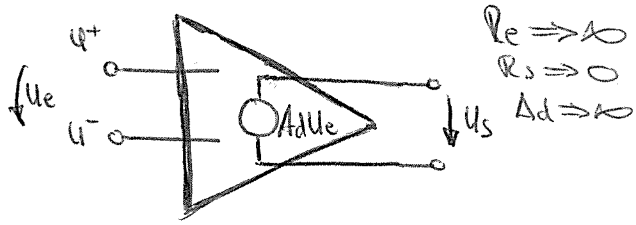
Los AOs son amplificadores diferenciales que se integran en los circuitos, caracterizados por tener:

- Ganancia de tensión muy alta $\longrightarrow$ IDEAL $A_d = \infty$
- Alta impedancia de entrada $\longrightarrow$ IDEAL $R_e = \infty$
- Baja impedancia de salida $\longrightarrow$ IDEAL $R_s = 0$
- Amplifican tensión y potencia

SÍMBOLO



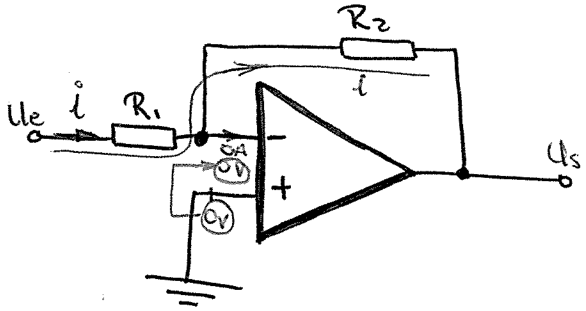

EQUIVALENTE AO IDEAL




# 3 APLICACIONES LINEALES DE LOS AOs

- Se tienen aplicaciones lineales cuando se realimenta negativamente el amplificador
- Consideraremos características ideales $(i = 0)$
- Con realimentación negativa $\longrightarrow u^+ = u^-$

## Amplificador inversor

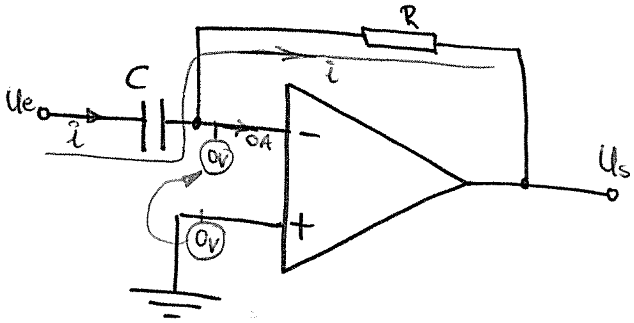



$$i=\frac{U_e-0}{R_1} \qquad i=\frac{0-U_s}{R_2}$$

$$\frac{U_e}{R_1}=\frac{-U_s}{R_2}$$

$$\boxed{U_s=\frac{-R_2}{R_1}U_e}$$

## Amplificador derivador




Análisis temporal

$$i_c=C\frac{dU_c}{dt}=C\frac{dU_e}{dt} \qquad i_R=\frac{0-U_s}{R}=\frac{-U_s}{R}$$

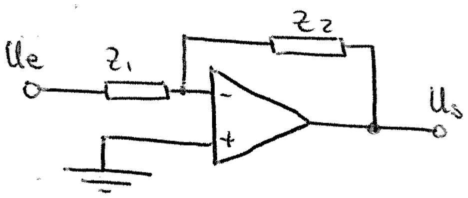
$$\boxed{U_s=-RC\frac{dU_e}{dt}}$$

Usando análisis frecuencial, por paralelismo con inversor

$$\boxed{U_s=\frac{-R}{1/Cs}U_e=-RCsU_e}$$

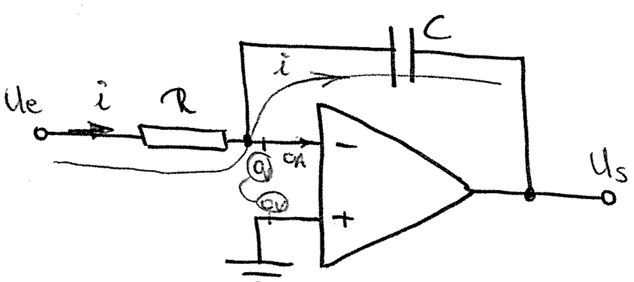
Un cero en el origen es un derivador

**NOTA!!** Generalizando, cualquier estructura del tipo inversor como:




$$U_s=\frac{-Z_2}{Z_1}U_e$$

## Amplificador integrador




Análisis temporal

$$i=\frac{U_e}{R} \qquad i=C\frac{d(-U_s)}{dt}$$

$$\frac{U_e}{R}=-C\frac{dU_s}{dt}\;;\; dU_s=-\frac{U_e}{RC}dt$$

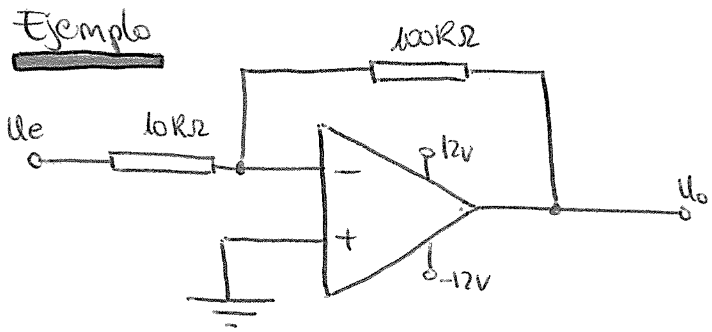
$$\boxed{U_s=\frac{-1}{RC}\int U_e\,dt}$$

Usando análisis frecuencial, por paralelismo con estructura inversora:

$$U_s=\frac{-\frac{1}{Cs}}{R}U_e\;; \qquad \boxed{U_s=\frac{-1}{RCs}U_e}$$

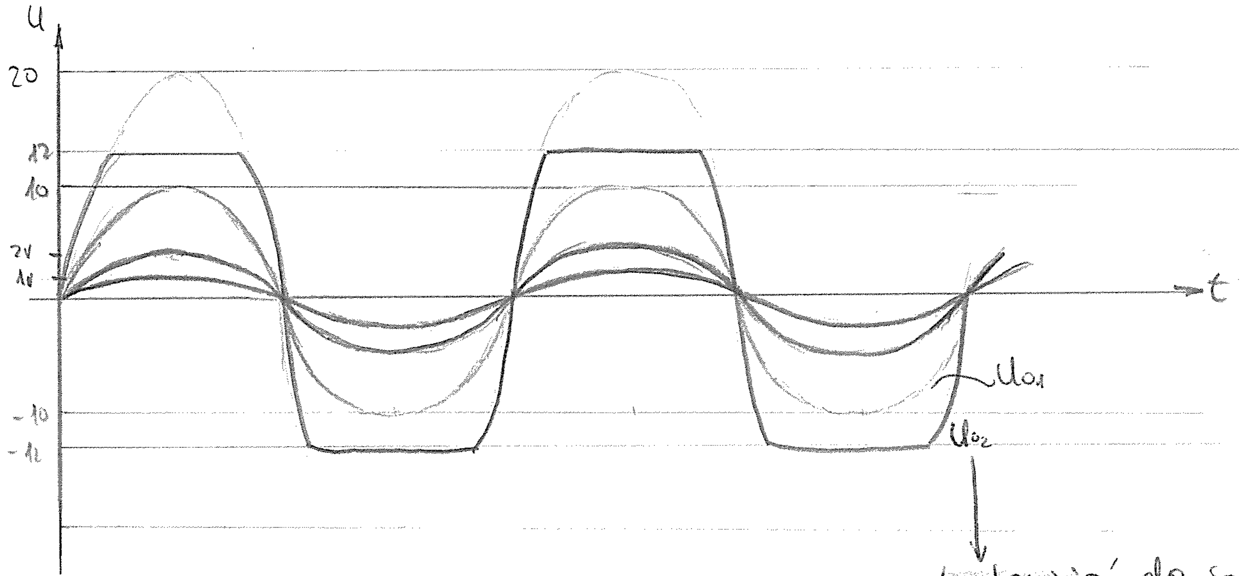
Un polo en el origen es un integrador

## Ejemplo



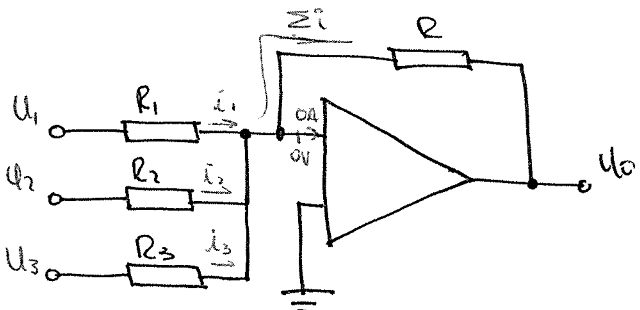

$$\boxed{\frac{U_o}{U_e} = \frac{-100}{10} = -10}$$

- Dibujar salida si $U_{e1} = 1V$ y $U_{e2} = 2V$




La tensión de salida no puede superar los 12 V de la alimentación

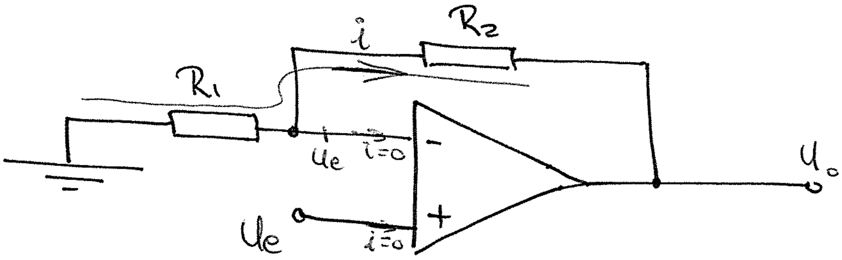
- SUMADOR INVERSOR




$$\frac{U_1}{R_1} + \frac{U_2}{R_2} + \frac{U_3}{R_3} = \frac{-U_o}{R}$$

$$\boxed{U_o = -\left(\frac{R}{R_1} U_1 + \frac{R}{R_2} U_2 + \frac{R}{R_3} U_3\right)}$$

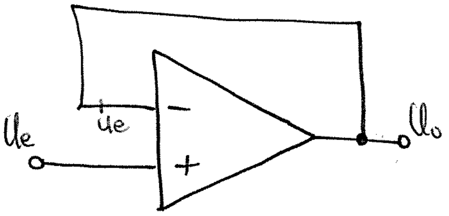
## AMPLIFICADOR DE GANANCIA POSITIVA




Ocurre cuando alimento por la pata ⊕

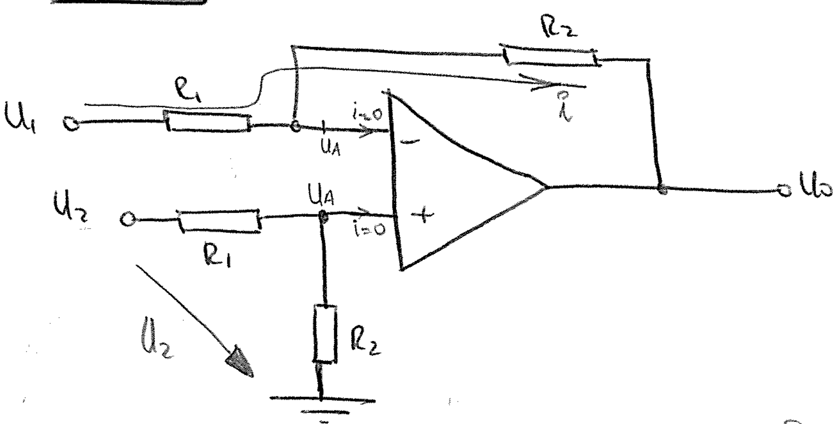
$$\frac{0-U_e}{R_1} = \frac{U_e-U_o}{R_2} ; \quad \frac{U_e}{R_1} = \frac{U_o-U_e}{R_2} ; \quad \boxed{U_o = \frac{R_2}{R_1}U_e + U_e = \left(1+\frac{R_2}{R_1}\right)U_e = \frac{R_2+R_1}{R_1}U_e}$$

## SEGUIDOR DE EMISOR




$$U_o = U_e$$

Ejercicio: Amplif diferencial. Se pide $U_o$



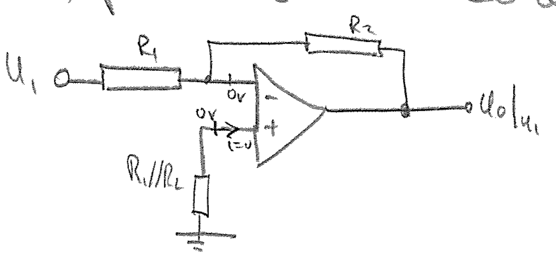

$$U_A = \frac{R_2}{R_1+R_2} U_2$$

$$\left.\begin{array}{l} i = \dfrac{U_1-U_A}{R_1} \\ i = \dfrac{U_A-U_o}{R_2} \end{array}\right\} \quad \frac{U_1-U_A}{R_1} = \frac{U_A-U_o}{R_2}$$

$$\frac{R_2}{R_1}U_1 - \frac{R_2}{R_1}U_A - U_A + U_o = 0 ; \quad \boxed{U_o = U_A\overbrace{\left(1+\frac{R_2}{R_1}\right)}^{\frac{R_1+R_2}{R_1}} - \frac{R_2}{R_1}U_1 = \frac{R_2}{R_1}(U_2-U_1)}$$

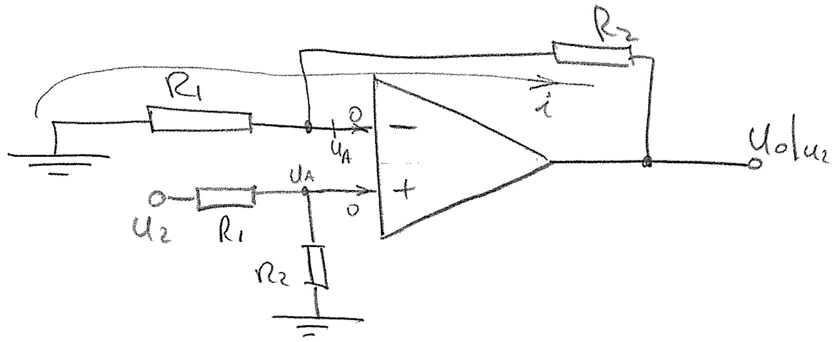
Aplicando superposición, podríamos haber hecho:

$U_o|_{U_1}$ ; $U_2 = \emptyset$ :




$$\frac{U_1}{R_1} = \frac{-U_o|_{U_1}}{R_2} ; \quad U_o|_{U_1} = \frac{-R_2}{R_1}U_1$$

$U_o\big|_{U_2}$ , $U_1 = \emptyset$




$$U_A = \frac{R_2}{R_1+R_2}U_2 \; ; \quad \frac{0-U_A}{R_1} = \frac{U_A - U_o\big|_{U_2}}{R_2} \; ; \quad \frac{U_A}{R_2} + \frac{U_A}{R_1} = \frac{U_o\big|_{U_2}}{R_2}$$

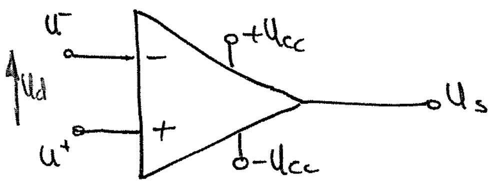
$$U_o\big|_{U_2} = U_A + \frac{R_2}{R_1}U_A = \frac{R_2}{R_1+R_2}U_2\left(1+\frac{R_2}{R_1}\right) = \frac{R_2}{R_1}U_2$$

$$\boxed{U_o = U_o\big|_{U_1} + U_o\big|_{U_2} = \frac{R_2}{R_1}(U_2 - U_1)}$$

# 4 APLICACIONES NO LINEALES DE LOS AOs

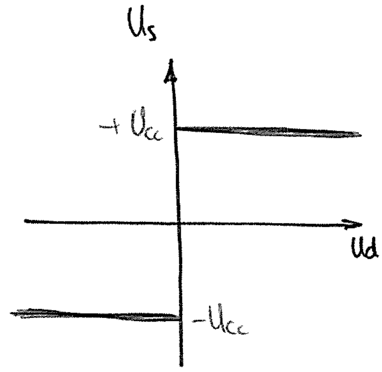
- Implican lazo abierto o realimentación positiva
- La salida solo puede valer $+U_{cc}$ o $-U_{cc}$

● LAZO ABIERTO. COMPARADOR

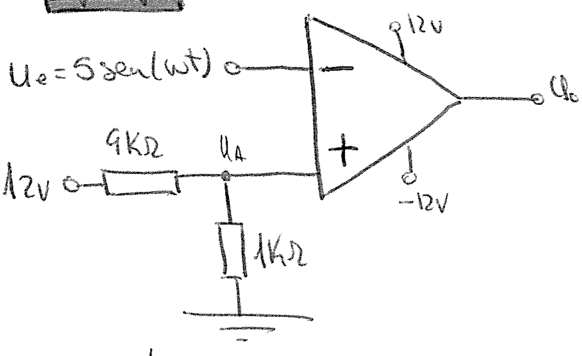


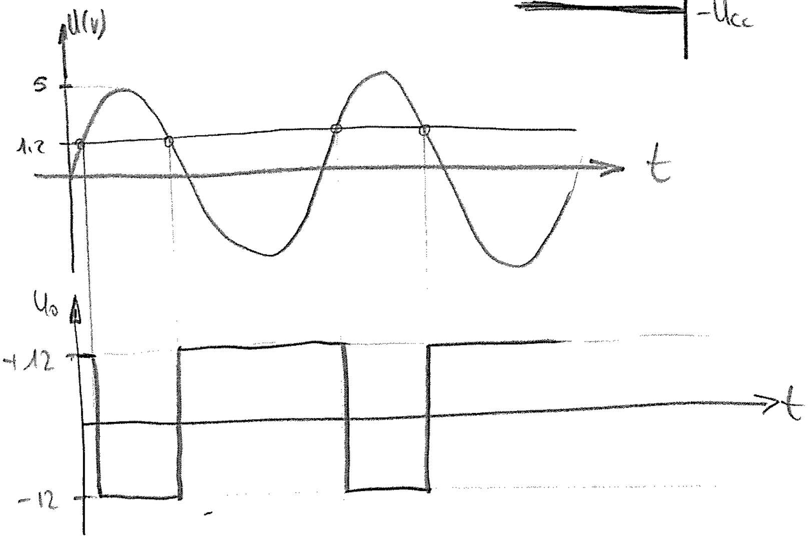

Si $U_- > U_+ \longrightarrow U_s = -U_{cc}$

Si $U_+ > U_- \longrightarrow U_s = +U_{cc}$



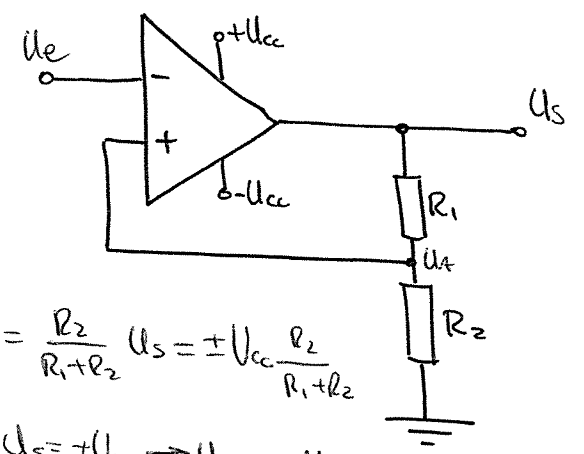

Ejemplo



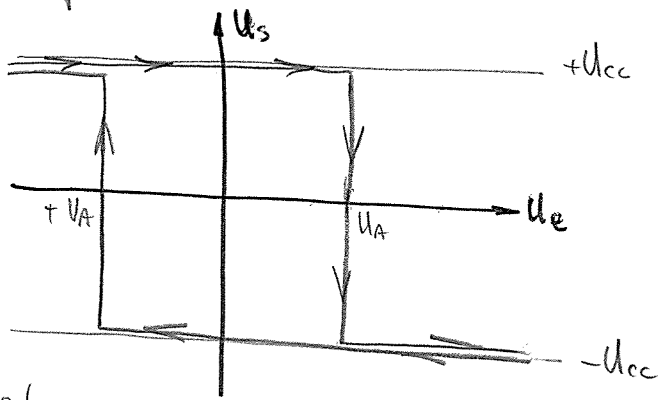




$$U_A = \frac{1}{9+1}\cdot 12 = 1{,}2$$

## Comparador con histéresis




Conviene dibujar 1º la variación Ue respecto a Us

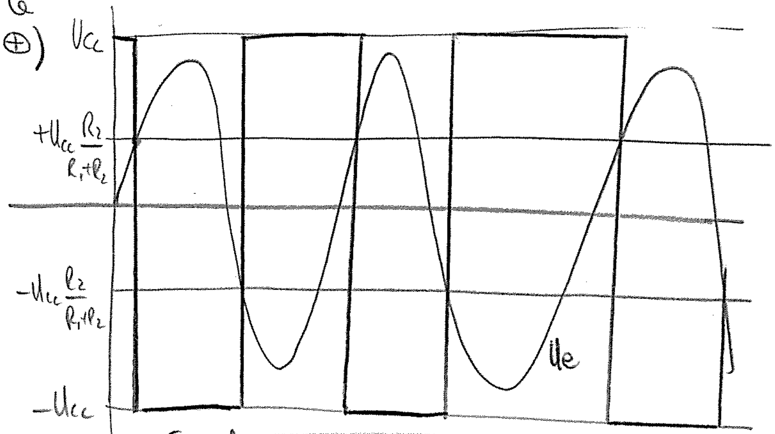



$V_A = \frac{R_2}{R_1+R_2} U_s = \pm V_{cc} \frac{R_2}{R_1+R_2}$

Si $U_s = +U_{cc} \rightarrow U_{comp} = U_A = +U_{cc} \cdot \frac{R_2}{R_1+R_2}$

Si $U_s = -U_{cc} \rightarrow U_{comp} = U_A = -U_{cc} \frac{R_2}{R_1+R_2}$

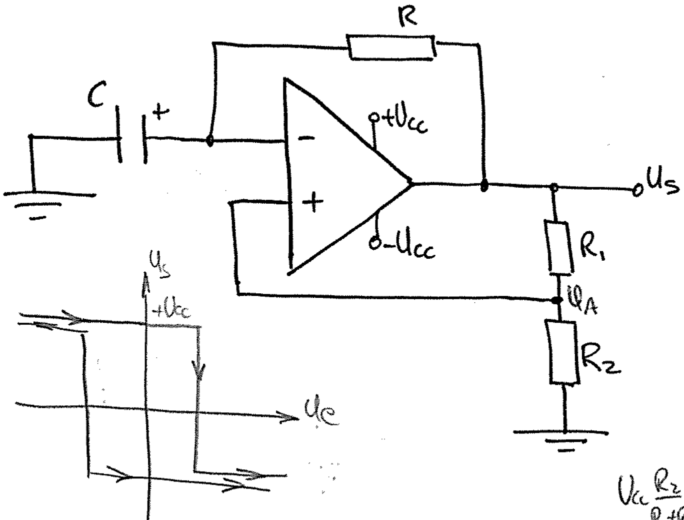
(la de la pata ⊕)




Cuando estoy en $-U_{cc}$ comparo con $-U_{cc} \frac{R_2}{R_1+R_2}$

Al revés cuando estoy en ⊕

## Oscilador aestable




De nuevo $U_A = \pm U_{cc} \frac{R_2}{R_1+R_2}$

- Si $U_s = +U_{cc}$ $U_{comp} = U_{cc} \frac{R_2}{R_1+R_2}$
- Si $U_s = -U_{cc}$ $U_{comp} = -U_{cc} \frac{R_2}{R_1+R_2}$

Cuando $U_s = +U_{cc}$ el condensador se carga

Cuando $U_s = -U_{cc}$ el condensador se descarga